

‘BotVoyage Platform’

**Ανάπτυξη διαδικτυακής πλατφόρμας που
εξειδικεύεται στην παροχή ταξιδιωτικών
πληροφοριών και επέκταση του
αυτοματοποιημένου λογισμικού BotVoyage.**

Η Διπλωματική Εργασία
παρουσιάστηκε ενώπιον
του Διδακτικού Προσωπικού του
Πανεπιστημίου Αιγαίου

Σε Μερική Εκπλήρωση
των Απαιτήσεων για το Δίπλωμα του
Μηχανικού Πληροφοριακών και Επικοινωνιακών Συστημάτων

των

ΧΡΙΣΤΟΦΟΡΟΥ ΚΟΝΑΚΑ ΚΑΙ ΖΑΧΑΡΙΑ ΣΙΑΛΙΑΚΗ
ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2018-2019

ΠΕΡΙΛΗΨΗ

Η διπλωματική αυτή εργασία, περιλαμβάνει την ανάπτυξη μιας πλατφόρμας στην οποία στεγάζεται ένα αυτοματοποιημένο λογισμικό (bot) και βασίζεται στην ανάλυση βασικών εννοιών της επιστήμης των υπολογιστών, όπως η τεχνητή νοημοσύνη, καθώς και βασικές αρχές βάσεων δεδομένων και προγραμματισμού στο διαδίκτυο. Επίσης, περιλαμβάνει μια εκτενής έρευνα των κυριότερων ταξιδιωτικών πλατφορμών, οι οποίες τείνουν σήμερα να κατακτήσουν το διαδίκτυο με την χρησιμοποίηση διαφόρων μεθόδων προσέλκυσης επισκεπτών στους διαδικτυακούς χώρους τους. Στην συνέχεια, παρουσιάζεται η σύλληψη της ιδέας για την πραγματοποίηση μιας τέτοιας εφαρμογής στο διαδίκτυο, καθώς και διάφορα εργαλεία που χρησιμοποιήθηκαν έτσι ώστε να επιτύχουμε την ανάπτυξη μιας τέτοιας εφαρμογής, η οποία έχει προγραμματιστεί έτσι ώστε να προσφέρει πληροφορίες ταξιδιωτικού περιεχομένου στους χρήστες μέσα από ένα γρήγορο και ευπαρουσίαστο περιβάλλον.

© 2018

ΤΩΝ

ΧΡΙΣΤΟΦΟΡΟΥ ΚΟΝΑΚΑ ΚΑΙ ΖΑΧΑΡΙΑ ΣΙΑΛΙΑΚΗ

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ABSTRACT

This diploma thesis includes the development of a platform that hosts an automated bot software and the analysis of basic concepts of computer science, such as artificial intelligence, as well as basic database and programming principles on the internet. It also includes an extensive survey of the major travel platforms that today tend to conquer the internet by using various methods of attracting visitors to their web sites. Below is the concept of implementing such an application on the web, as well as the various tools used to achieve the development of such an application, that is programmed to provide travel information to users through a fast and presentable environment.

© 2018

CHRISTOFOROS KONAKAS AND ZAHARIAS SIALIAKIS

Department of Information and Communication Systems Engineering

UNIVERSITY OF THE AEGEAN

ΕΥΧΑΡΙΣΤΙΕΣ - ΑΦΙΕΡΩΣΕΙΣ

Πρώτα απ' όλα θα θέλαμε να πούμε ένα τεράστιο ευχαριστώ στον κ . Ιωάννη Χαραλαμπίδη, αναπληρωτή καθηγητή του τμήματος Μ.Π.Ε.Σ του πανεπιστημίου Αιγαίου για την συνεισφορά του και το ενδιαφέρον του σε προβλήματα που αφορούσαν την ανάπτυξη της εφαρμογής μας. Επίσης η υπόληψή μας προς αυτόν σαν καθηγητή καθώς και η τεράστια αναγνώρισή του στο περιβάλλον της Ηλεκτρονικής Διακυβέρνησης αποτέλεσαν ισχυρά κίνητρα για εμάς στο να προσδοκούμε μια υλοποίηση διπλωματικής εργασίας μαζί του.

Ακόμα, είναι αξιοσημείωτη η βοήθεια που λάβαμε από την κ. Ζωή Λαχανά, για συστηματική επίλυση αποριών που αφορούσαν την διπλωματική μας εργασία, και χωρίς αυτήν την βοήθεια η εργασία δεν θα είχε αυτό το τελικό αποτέλεσμα.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ	11
1.1 Σκοπός της εργασίας	11
1.2 Αντικείμενο της εργασίας	11
1.3 Δομή εργασίας	12
ΚΕΦΑΛΑΙΟ 2 - ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ	13
2.1 Τεχνητή Νοημοσύνη	13
2.1.1 Ορισμός	13
2.1.2 Η τεχνητή νοημοσύνη στη ζωή μας	13
2.1.2.1 Κλάδοι συσχέτισης	13
2.2 Διαδικτυακό Robot	14
2.2.1 Ορισμός	14
2.2.2 Τα bot στη ζωή των ανθρώπων	14
2.2.2.1 Χρήση	14
2.2.3 Bot και τουρισμός	15
2.2.4 Απλότητα στα τουριστικά bot	15
2.2.5 Τουριστικά bots και ευφυΐα	16
ΚΕΦΑΛΑΙΟ 3 - ΑΝΑΛΥΣΗ ΚΥΡΙΟΤΕΡΩΝ ΤΑΞΙΔΙΩΤΙΚΩΝ ΠΛΑΤΦΟΡΜΩΝ	17
3.1 Ανάλυση των κυριότερων πλατφορμών στον ταξιδιωτικό χώρο	17
3.1.1 trivago.com	17
3.1.2 tripadvisor.com	19
3.1.3 yallou.com	21
3.1.4 tripexpert.com	22
3.1.5 yelp.com	23
3.1.6 trip.skyscanner.com	25
3.2 Πίνακας σύγκρισης των χαρακτηριστικών	27
3.2.1 Υπόμνημα Πίνακα – Επεξήγηση	27
3.3 Συμπεράσματα ανάλυσης	28
ΚΕΦΑΛΑΙΟ 4 - ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΙ ΕΡΓΑΛΕΙΑ	29
4.1 Σύλληψη και διαμόρφωση ιδέας	29
4.2 Πλατφόρμες και εργαλεία	30
4.3 JavaScript	30
4.3.1 Ιστορική αναδρομή	30
4.3.2 Τι είναι	31
4.4 Javascript Frameworks	31
4.4.1 ReactJS	31
4.4.1.1 Ιστορική Αναδρομή	31
4.4.1.2 Χρήση	32
4.4.1.3 ReactJS - Χαρακτηριστικά στοιχεία	32

4.4.2	VueJS	34
4.4.3	AngularJS	34
4.4.4	Ζήτηση των τριών Frameworks	35
4.4.5	Πλεονεκτήματα και μειονεκτήματα των Javascript Frameworks	36
4.4.5.1	ReactJs	36
4.4.5.2	AngularJS	37
4.4.5.3	VueJS	38
4.5	Node.js	39
4.5.1	Τι είναι	39
4.6	Firebase (API)	40
4.6.1	Τι είναι	40
4.6.2	Ιστορική Αναδρομή	41
4.6.3	Υπηρεσίες Firebase	42
4.6.3.1	Firebase Analytics η αλλιώς Google Analytics for Firebase	42
	Εικόνα 23 - Firebase Analytics - https://firebase.google.com/docs/analytics/	42
4.6.3.1.1	Τι είναι	42
4.6.3.1.2	Πως δουλεύει	42
4.6.3.2	Firebase Cloud Messaging	43
4.6.3.2.1	Τι είναι	43
4.6.3.2.2	Πως δουλεύει	43
4.6.3.3	Firebase Authentication	44
4.6.3.3.1	Τι είναι	44
4.6.3.3.2	Πως δουλεύει	45
4.6.3.4	Firebase Realtime Database	46
4.6.3.4.1	Τι είναι	46
4.6.3.4.2	Πως δουλεύει	46
4.6.3.5	Firebase Cloud Storage	48
4.6.3.5.1	Τι είναι	48
4.6.3.5.2	Πως δουλεύει	48
4.6.3.6	Firebase Hosting	49
4.6.3.6.1	Τι είναι	50
4.6.3.6.2	Πως δουλεύει	50
4.6.3.7	Firebase Cloud Functions	51
4.6.3.7.1	Τι είναι	51
4.6.3.7.2	Πως δουλεύει	51
4.6.4	Σύνδεση εφαρμογής με Firebase	53
4.6.4.1	Firebase Facebook και Google Login	55
4.6.4.2	Αποθήκευση χρηστών στην βάση	55
4.6.4.3	Αναπαράσταση των logins με γραφήματα	56
4.7	Webpack	56

4.7.1 Τι είναι	56
4.7.2 Πως δουλεύει	57
4.7.3 Διαδικασία Επίλυσης	58
4.7.4 Διαδικασία αξιολόγησης	59
4.7.5 Το webpack με βάση το configuration	59
4.7.6 Hot Module Replacement	60
4.7.7 Διαχωρισμός κώδικα	60
4.7.8 Συμπεράσματα	61
4.7.9 Σύνοψη	61
4.7.10 Styling	62
4.7.11 Φόρτωση css	62
4.7.12 Εγκατάσταση file-loader για φόρτωση εικόνων	63
4.7.13 Εγκατάσταση των webpack plugins και ορισμός της πόρτας	64
4.8 Babel	67
4.8.1 Τι είναι	67
4.8.2 ES2015 και μετέπειτα	67
4.8.3 JSX και React	67
4.8.4 Λογική	68
4.8.5 React JSX	68
4.8.6 Καθορισμός του τύπου React Element	69
4.9 Wit.ai	70
4.9.1 Τι είναι NLP/NLU	70
4.9.2 Τι είναι το Wit.ai	71
4.10 Git (λογισμικό)	72
4.10.1 Τι είναι και που μας χρησιμεύει	72
4.11 Npm	73
4.11.1 Τι είναι	73
4.11.2 Πώς χρησιμοποιείται με το Visual Studio Code	73
4.11.3 Τα Npm πακέτα της πλατφόρμας μας	77
4.11.3.1 Dependencies	78
4.11.3.2 DevDependencies	78
4.11.3.3 Κατηγορία NPM Πακέτων Dependencies	79
4.11.3.3.1 Material-UI	79
4.11.3.3.2 Axios	79
4.11.3.3.3 Botframework - Webchat (Channel)	80
4.11.3.3.4 Firebase	80
4.11.3.3.5 firestore-react	81
4.11.3.3.6 React-alice-carousel	81
4.11.3.3.7 react-admin	82
4.11.3.3.8 React-admin-firebase	82
4.11.3.3.9 react-containers	82

4.11.3.3.10 React-dom	82
4.11.3.3.11 React-firebase	82
4.11.3.3.12 React-firebaseui	83
4.11.3.3.13 React-redux	83
4.11.3.3.14 React-router-dom	83
4.11.3.3.15 Redux-saga	85
4.11.3.4 Κατηγορία NPM Πακέτων DevDependencies	85
4.11.3.4.1 Babel-core	85
4.11.3.4.1 Eslint	86
4.11.3.4.2 File-loader	86
4.11.3.4.3 Html-webpack-plugin	86
4.11.3.4.4 Image-loader-webpack	87
4.11.3.4.5 Node-sass	87
4.11.3.4.6 Prop-types	87
4.11.3.4.7 Sass-loader	87
4.11.3.4.8 Style-loader	87
4.11.3.4.9 Superagent	87
4.11.3.4.10 Webpack	88
4.11.3.4.11 Webpack-dev-server	88
4.12 HTML	88
4.12.1 Τι είναι	88
4.12.2 Εκδόσεις HTML	89
4.12.3 Δημιουργία αρχείων HTML	89
4.12.4 CSS	89
4.12.5 SCSS	90
4.13 Foursquare (API)	90
4.13.1 Τι είναι	90
4.13.2 Πώς λάβαμε δεδομένα από το Foursquare API	91
4.14 OpenWeatherMap (API)	93
4.14.1 Τι είναι	93
4.14.2 Πως χρησιμοποιείται	93
4.15 BotVoyage Platform Block Diagram	94
4.16 Προβλήματα και περιορισμοί εργαλείων	95
ΚΕΦΑΛΑΙΟ 5 - ΑΠΑΙΤΗΣΕΙΣ BOTVOYAGE - PLATFORM	97
5.1 Διαγράμματα	97
5.1.1 Use Case Diagrams	97
5.1.1.1 Μελέτη Περίπτωσης Επισκέπτη Πλατφόρμας	97
5.1.1.2 Μελέτη Περίπτωσης Χρήστη Πλατφόρμας	98
5.2 Απαιτήσεις συστήματος	99
5.2.1 Σύνολο απαιτήσεων πλατφόρμας	99

ΚΕΦΑΛΑΙΟ 6 - ΕΝΣΩΜΑΤΩΣΗ ΤΟΥ ΒΟΤ ΣΤΗΝ ΠΛΑΤΦΟΡΜΑ ΚΑΝΟΝΤΑΣ ΧΡΗΣΗ ΤΟΥ ΚΑΝΑΛΙΟΥ WEBCHAT	105
6.1 Τι είναι το Bot Framework	105
6.2 Bot Connector	106
6.3 Channels	107
6.3.1 Web Chat	108
6.4 Μοντέλο διαλόγου (Dialog)	110
ΚΕΦΑΛΑΙΟ 7 - ΠΑΡΟΥΣΙΑΣΗ ΛΕΙΤΟΥΡΓΙΩΝ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ	111
7.1 Αρχική σελίδα	112
7.2 Σελίδα AboutUs	114
7.3 Σελίδα sign_in	114
7.4 Σελίδα SIGN UP	116
7.5 Σελίδα GOOGLE LOGIN	116
7.6 Σελίδα Search	117
7.7 Είσοδος του χρήστη στην πλατφόρμα	118
7.8 Σελίδα Account- προφίλ του συνδεδεμένου χρήστη	119
7.9 Επιλογές του χρήστη	120
7.9.1 Αλλαγή Password.	120
7.9.2 Αλλαγή του ονόματος προφίλ	121
7.9.3 Αλλαγή της εικόνας προφίλ του χρήστη.	122
7.9.4 Επιλογή PasswordForgot στην σελίδα SIGN IN	124
7.10 Σελίδα του Bot	125
7.10.1 Αποτελέσματα BOT - Παρουσίαση κατηγοριών(επεκτάσεων)	126
7.10.1.1 Μήνυμα Καλωσορίσματος	126
7.10.1.2 Πληροφορίες αναζητήσεων	127
7.10.2 Κατηγορίες	127
7.10.2.1 DrinkType (coffee, beer, coctail, wine)	127
7.10.2.2. FoodType (sushi, souvlaki, pizza, mexican, burger, restaurant)	129
7.10.2.3. Restaurants	131
7.10.2.4. Μήνυμα Ευχαρίστησης (English)	132
7.10.2.5 Επιλογή Ροφήματος	132
7.10.2.6 Επιλογή φαγητού	133
7.10.2.7 Pharmacy	133
7.10.2.8. Rent a car	134
7.10.2.9. Cinema	134
7.10.2.10. Casino	135
7.10.2.11. Gas Station	135
7.10.2.12. Beach	136
7.10.2.13. University	136
7.10.2.14. Gym	137
7.10.2.15. Paintball	137

7.10.2.16. Stadium	138
7.10.2.17. Golf	138
7.10.2.18. Billiards	139
7.10.2.19. Shop	139
7.10.2.20. Market	140
7.10.2.21. Hotel	140
7.10.2.22. Zoo	141
7.10.2.23. Ski	141
7.10.2.24. Train	142
7.10.2.25 Bus	142
7.10.2.26. Port (Boat-Ship)	143
7.10.2.27. Club	143
7.10.2.28. Bakery	144
7.10.2.29. Museum	144
7.10.2.30. Theatre	145
7.10.2.31. Music Stores	145
7.10.2.32. Weather	146
7.11 Σελίδα Recommendations	146
7.12 Κουμπί SIGN OUT	147
7.13 Βάση δεδομένων firebase	148
7.13.1 Δημιουργία χρήστη στην βάση μας με την μέθοδο sign up.	148
7.14 Λειτουργίες admin dashboard	149
7.14.1 Σύνδεση του χρήστη στο admin dashboard	149
7.14.2 Δημιουργία καινούργιου post	152
7.14.3 Τροποποίηση και διαγραφή του post	153
7.14.4 Σύνοψη	154
7.15 Απεικόνιση και ανάλυση διαχειριστικών περιβαλλόντων	155
7.15.1 Azure Portal	155
7.15.2 Bot Framework Emulator	156
7.15.3 Wit.ai	156
7.15.4 Facebook for Developers Platform	159
ΚΕΦΑΛΑΙΟ 8 - ΕΠΕΚΤΑΣΗ - ΣΥΜΠΕΡΑΣΜΑΤΑ - ΕΠΟΜΕΝΑ ΒΗΜΑΤΑ	163
8.1 Επέκταση BotVoyage	163
8.2 Συμπεράσματα επί των εργαλείων	164
8.2 Συμπεράσματα επί της εφαρμογής	166
8.2.1 Λειτουργικές απαιτήσεις πλατφόρμας	168
8.2.2 Μη λειτουργικές απαιτήσεις πλατφόρμας	169
8.3 Επόμενα βήματα	170

ΚΕΦΑΛΑΙΟ 1 - ΕΙΣΑΓΩΓΗ

1.1 Σκοπός της εργασίας

Σήμερα, το διαδίκτυο αποτελεί την νούμερο ένα μηχανή αναζήτησης πληροφοριών που αφορούν τον τουριστικό τομέα. Γι' αυτόν τον λόγο, οι χρήστες αναζητούν ιστοχώρους στους οποίους η ανταλλαγή πληροφοριών θα έχει μια κατανοητή και εύκολη δομή. Στόχος της διπλωματικής μας εργασίας είναι η εύρεση ταξιδιωτικών πληροφοριών προς τους χρήστες της εφαρμογής μας, μέσα από μια μεγάλη γκάμα κατηγοριών όπως η διασκέδαση, η ψυχαγωγία, η διαμονή, το φαγητό, ο καιρός, κ.ά. για διάφορες περιοχές προτίμησης του χρήστη. Η εργασία μας απευθύνεται σε όλους τους χρήστες παγκοσμίως και δεν περιορίζεται σε συγκεκριμένα γεωγραφικά σύνορα. Το κύριο μέλημά μας είναι η επιλογή της εφαρμογής μας από τον χρήστη που αναζητά ταξιδιωτικές πληροφορίες ανάμεσα σε ένα μεγάλο πλήθος τουριστικών ιστοτόπων, που εξειδικεύονται στην παροχή πληροφοριών σχετικά με τα ταξίδια.

1.2 Αντικείμενο της εργασίας

Το αντικείμενο της εργασίας εδραιώνεται στην επέκταση της υπάρχον εφαρμογής botvoyage, η οποία στεγάζεται στην πλατφόρμα του messenger μέσω της επανεκπαίδευσης της διεπαφής της φυσικής γλώσσας, στην ενίσχυση των αποτελεσμάτων αναζήτησης από την προσθήκη περισσότερων API, http πακέτων και επιλογών προς τους χρήστες, περισσότερης εκπαίδευσης στο bot, αλλά κυρίως στην δημιουργία πλατφόρμας για την πλαισίωση του bot έτσι ώστε το τελικό αποτέλεσμα να είναι μια ολοκληρωμένη εφαρμογή με μια ωραία διεπαφή ανάμεσα στον χρήστη και την εφαρμογή που προσδίδει αμεσότητα και ευκολία. Με λίγα λόγια αυτό που θέλουμε να πετύχουμε είναι η ακριβής πληροφόρηση του χρήστη μέσα σε ένα ευχάριστο διαδικτυακό περιβάλλον, η δυνατότητα επικοινωνίας των επισκεπτών μας με έναν ΑΙ πράκτορα ο οποίος θα πληροί ακριβώς τα χαρακτηριστικά που αναζητούν οι χρήστες, καθώς και η καταγραφή στοιχείων των χρηστών σε βάσεις δεδομένων για εξαγωγή πολλαπλών συμπερασμάτων.

1.3 Δομή εργασίας

Στο κεφάλαιο 2 αναλύονται κάποιοι σημαντικοί ορισμοί, οι οποίοι είναι άρρητα συνδεδεμένοι με την διπλωματική μας εργασία και την ανάπτυξη της εφαρμογής, την οποία πραγματοποιήσαμε. Αυτοί οι ορισμοί αφορούν κλάδους της πληροφορικής, όπως είναι η τεχνητή νοημοσύνη και τα bots καθώς και τη σημασία αυτών στο διαδικτυακό τουριστικό περιβάλλον.

Στο κεφάλαιο 3 επεξηγούνται και αναλύονται οι λειτουργίες των σημαντικότερων ταξιδιωτικών ιστοχώρων, με βάση τις οποίες συνθέσαμε και αναπτύξαμε την δική μας εφαρμογή.

Στο κεφάλαιο 4 παρουσιάζονται οι μεθοδολογίες καθώς και τα εργαλεία τα οποία χρησιμοποιήσαμε και στα οποία βασιστήκαμε, έτσι ώστε να επιτύχουμε την ανάπτυξη της διαδικτυακής εφαρμογής μας, καθώς και μια σειρά από προβλήματα και περιορισμούς τα οποία στάθηκαν εμπόδιο στην διαδρομή μας και μας γέμισαν προβληματισμούς όσον αφορά το σκεπτικό μας.

Στο κεφάλαιο 5 γίνεται η ανάλυση της εφαρμογής μας μέσω διαγραμμάτων περιπτώσεων χρήσης (use case diagrams), έτσι ώστε να παρέχουμε στον αναγνώστη μια πιο κατανοητή επεξήγηση των λειτουργιών που προσφέρουμε σαν developers. Επίσης, παρουσιάζονται οι απαιτήσεις του συστήματος(της εφαρμογής μας) αριθμημένες, είτε αυτές αφορούν την πλατφόρμα μας, είτε το bot.

Στο κεφάλαιο 6 ορίζονται οι έννοιες των εργαλείων και των καναλιών που χρησιμοποιήθηκαν, προκειμένου να γίνει η σύνδεση του bot στην διαδικτυακή πλατφόρμα μας, για αλληλεπίδραση με τους χρήστες της.

Στο κεφάλαιο 7 απεικονίζονται όλες οι λειτουργίες της πλατφόρμας (και κατά συνέπεια του BotVoyage), είτε πρόκειται για επισκέπτη, είτε για χρήστη που πραγματοποιεί είσοδο και επιθυμεί να συνομιλήσει με το αυτοματοποιημένο λογισμικό.

Στο κεφάλαιο 8 περιλαμβάνονται όλα τα συμπεράσματα επί της διπλωματικής εργασίας, είτε αυτά αφορούν την εφαρμογή είτε τα εργαλεία που χρησιμοποιήθηκαν, οι λειτουργικές και οι μη λειτουργικές απαιτήσεις της εφαρμογής μας, έτσι ώστε να πραγματοποιηθεί μια εκτενής σύνθεση των μελλοντικών βημάτων και επεκτάσεων.

Τέλος αναγράφεται η βιβλιογραφία που χρησιμοποιήθηκε προκειμένου να εκπονηθεί αυτή η διπλωματική εργασία.

ΚΕΦΑΛΑΙΟ 2 - ΕΙΣΑΓΩΓΙΚΕΣ ΕΝΝΟΙΕΣ

2.1 Τεχνητή Νοημοσύνη

2.1.1 Ορισμός

Η τεχνητή νοημοσύνη (Artificial Intelligence/ AI) αναφέρεται στον κλάδο της πληροφορικής και εστιάζει στη δημιουργία έξυπνων μηχανών, που λειτουργούν και συμπεριφέρονται όπως οι άνθρωποι. Μερικές από τις δραστηριότητες στο χώρο των υπολογιστών με τη δράση της τεχνητής νοημοσύνης έχουν σχεδιαστεί για να περιλαμβάνουν:

1. Κατανόηση ομιλίας
2. Μάθηση
3. Σχεδιασμό
4. Επίλυση προβλημάτων

2.1.2 Η τεχνητή νοημοσύνη στη ζωή μας

2.1.2.1 Κλάδοι συσχέτισης

Η τεχνολογία γνώσης (Knowledge engineering) είναι ένα σημαντικό κομμάτι στην έρευνα της τεχνητής νοημοσύνης. Τα εξαρτήματα μπορούν να δρουν και να συμπεριφέρονται συχνά όπως οι άνθρωποι, μόνο όταν έχουν στην κατοχή τους άφθονες πληροφορίες που σχετίζονται με τον έξω κόσμο. Η τεχνητή νοημοσύνη πρέπει να έχει σύνδεση σε αντικείμενα, σε κατηγορίες, σε ιδιότητες και σχέσεις μεταξύ όλων αυτών για να μπορέσουν να εφαρμόσουν την τεχνολογία της γνώσης.

Η εκμάθηση μηχανών (Machine learning) είναι ένας επίσης μεγάλο κομμάτι της τεχνητής νοημοσύνης. Η μάθηση χωρίς κάποιου είδους εποπτεία, απαιτεί την δυνατότητα να αναγνωρίζει μονοπάτια εισαγωγικών ροών, ενώ η μάθηση με ύπαρξη επίβλεψης, συμπεριλαμβάνει τη σωστή ταξινόμηση και παλινδρόμηση των ροών αυτών. Η σωστή ταξινόμηση καθορίζει την κατηγορία που ανήκει ένα αντικείμενο εισροής, ενώ η παλινδρόμηση ασχολείται με την απόκτηση ενός συνόλου X αριθμητικών παραδειγμάτων

εισόδων ή εξόδων των ροών, με αποτέλεσμα να υπάρχουν οι κατάλληλες λειτουργίες που επιτρέπουν την έγκυρη παραγωγή εξόδων από τις αντίστοιχες εισόδους που δέχθηκαν.

Η ρομποτική τέλος εξαρτάται σε μεγάλο βαθμό από την τεχνητή νοημοσύνη. Τα ρομπότ χρειάζονται την κατάλληλη ευφυΐα, για να διαχειριστούν εργασίες όπως είναι ο χειρισμός των αντικειμένων και της πλοήγησης, μαζί με υπό-προβλήματα εντοπισμού, σχεδιασμού κινήσεων και χαρτογράφησης.

2.2 Διαδικτυακό Robot

2.2.1 Ορισμός

Το bot διαδικτύου, το οποίο είναι γνωστό και ως www robot ή απλά bot, είναι ένα πρόγραμμα λογισμικού, το οποίο εκτελεί αυτοματοποιημένες εργασίες μέσω του διαδικτύου.

2.2.2 Τα bot στη ζωή των ανθρώπων

2.2.2.1 Χρήση

Στις περισσότερες περιπτώσεις τα bots εκτελούν σχετικά απλές λειτουργίες, που θα πρέπει να επαναληφθούν εκατοντάδες ή χιλιάδες φορές. Μία κλασική εφαρμογή των bots, είναι οι αράχνες του διαδικτύου (Web spiders), οι οποίες περιφέρονται από ιστοσελίδα σε ιστοσελίδα και χρησιμοποιούνται για την ανάλυση της, σε ρυθμό πολλαπλάσιο από ότι θα μπορούσε ένας άνθρωπος. Για παράδειγμα, ακόμη και οι μεγαλύτερες μηχανές αναζήτησης (Google , Yahoo, κοκ) χρησιμοποιούν αυτήν την τεχνική για την ανάλυση και την ταξινόμηση των ιστοσελίδων, σύμφωνα με διάφορες λέξεις – κλειδιά, ούτως ώστε στη συνέχεια να μπορούν να παρουσιάσουν στο χρήστη τα αποτελέσματα της αναζήτησης σε πολύ μικρό χρονικό διάστημα. Μετέπειτα, οι διάφοροι web servers μπορούν να δημιουργήσουν ένα απλό κείμενο με το όνομα «robots.txt», το οποίο θα περιέχει κανόνες που θα πρέπει να τηρήσουν τα bots που επισκέπτονται την σελίδα. Παρόλα αυτά όμως , μπορεί κάποιο bot να αγνοήσει αυτούς του κανόνες.

2.2.3 Bot και τουρισμός

Τα online chatbots, έχουν ήδη αποδείξει τα πλεονεκτήματά τους όσον αφορά την προσέλκυση επιχειρήσεων και την παροχή υποστήριξης στους πελάτες. Αυτοί οι δύο παράγοντες, σχετίζονται σε μεγάλο βαθμό με τον τουριστικό κλάδο. Ως συμμετέχοντες σε ένα ανταγωνιστικό περιβάλλον, τα ταξιδιωτικά γραφεία, οι ξενοδοχειακές κρατήσεις και άλλες επιχειρήσεις που έχουν οικοδομηθεί γύρω από τα ταξίδια και τον τουρισμό, πρέπει να συνεχίσουν να αξιολογούν και να προωθούν τις προσφορές τους και τις υπηρεσίες τους, για να εξασφαλίσουν ότι θα αντλήσουν και θα διατηρήσουν αρκετούς πελάτες. Με την ανάπτυξη των chatbots ταξιδιού, μπορούν να πετύχουν αυτό, καθώς και πολλά άλλα.

2.2.4 Απλότητα στα τουριστικά bot

Όταν πρόκειται για αναζήτηση πληροφοριών ή βοήθειας, το πιο απλό και χρήσιμο χαρακτηριστικό είναι να ρωτάς μια ερώτηση και να παίρνεις την ακριβή απάντηση που αναζητάς. Αυτό είναι και το χαρακτηριστικό στο οποίο οι καλύτεροι ιστοχώροι ταξιδιών και εφαρμογές για smartphones υστερούν. Μπορούν να προσφέρουν ελκυστικά πακέτα διακοπών και τιμών δωματίων ή εξαιρετικά λεπτομερείς πληροφορίες σχετικά με προορισμούς αξιοθέατων και υπηρεσίες πόλεων, αλλά αν οι χρήστες πρέπει να περιηγηθούν σε πολλαπλά επίπεδα πληροφοριών προκειμένου να βρουν αυτό που αναζητούν, ίσως καταλήξουν να εγκαταλείψουν την διαδικασία αναζήτησης. Αντιθέτως, κάποιος που ενδιαφέρεται για μια τοποθεσία μπορεί απλά να κατευθυνθεί στο παράθυρο συνομιλίας και να θέσει μια απλή ερώτηση όπως “Ποια είναι τα δέκα καλύτερα πράγματα που μπορώ να κάνω στην συγκεκριμένη τοποθεσία;”. Ακόμα και η πληκτρολόγηση λέξεων κλειδιών όπως “εστιατόρια”, “μουσεία”, “ψώνια” ή ακόμα και “πεζοπορία”, θα παράγει χρήσιμες και σχετικές πληροφορίες σχετικά με τα μέρη στα οποία μπορούν να πάνε. Τα καλύτερα chatbots για τις τουριστικές υπηρεσίες, ενσωματώνουν χαρακτηριστικά όπως οδηγίες και σχόλια επισκεπτών με τις προτάσεις τους, βοηθώντας τους τουρίστες να κάνουν τις σωστές επιλογές με βάση το χρονοδιάγραμμα, τον προϋπολογισμό τους και άλλες προτιμήσεις.

2.2.5 Τουριστικά bots και ευφυΐα

Οι ιστοσελίδες και τα μενού εφαρμογών απαιτούν κάποια συνήθεια από τους χρήστες. Οι χρήστες πρέπει να εξοικειωθούν με τις διαθέσιμες επιλογές και μετά από μερικές επισκέψεις, θα μπορούν να ολοκληρώσουν τις αναζητήσεις ξενοδοχείων, πτήσεων και υπηρεσιών τοποθεσιών πιο γρήγορα. Τα τουριστικά chatbots, κάνουν αυτήν την διαδικασία ξεχωριστή και την περνάνε σε άλλο επίπεδο. Αντί να ζητούν από τους χρήστες να γνωρίσουν την ταξιδιωτική υπηρεσία καλύτερα, τα bots τεχνητής νοημοσύνης, λειτουργούν αντίθετα. Χρησιμοποιούν εκμάθηση μηχανών για τον εντοπισμό της κοινώς χρησιμοποιούμενης φρασεολογίας και των όρων αναζήτησης, για την καλύτερη προσαρμογή τους στις προτιμήσεις των ταξιδιωτών με την πάροδο του χρόνου. Εκτός αυτού, τα σενάρια συζήτησης, μπορούν πάντα να ενημερώνονται με βάση τα στοιχεία των προηγούμενων συνομιλιών με τους χρήστες. Η ανάπτυξη του chatbot είναι έτσι μια δυναμική διαδικασία. Κατά τη διάρκεια των αλληλοεπιδράσεων τους με τους ταξιδιώτες, τα bots γίνονται πιο έξυπνα και επιτρέπουν στην τουριστική επιχείρηση να προσφέρει βελτιωμένες υπηρεσίες και εμπειρίες χρηστών.

ΚΕΦΑΛΑΙΟ 3 - ΑΝΑΛΥΣΗ ΚΥΡΙΟΤΕΡΩΝ ΤΑΞΙΔΙΩΤΙΚΩΝ ΠΛΑΤΦΟΡΜΩΝ

3.1 Ανάλυση των κυριότερων πλατφορμών στον ταξιδιωτικό χώρο

3.1.1 trivago.com

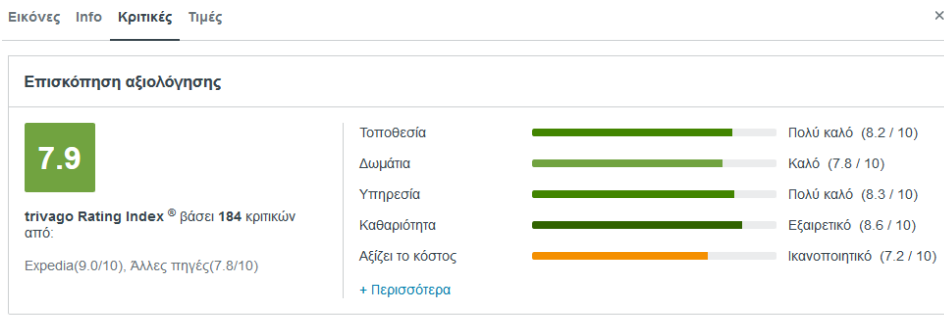
Η συγκεκριμένη πλατφόρμα, βρίσκει ξενοδοχεία συγκρίνοντας τιμές από διαφορετικές ιστοσελίδες. Η ροή των αναζητήσεων γίνεται από την στιγμή που ο χρήστης θα αναζητήσει μια τοποθεσία στο αρχικό search bar. Π.χ. ξενοδοχεία στην περιοχή Χαλκιδική.

The screenshot shows the Trivago search interface. At the top, the search bar contains 'Χαλκιδική' and the currency is set to EUR. Below the search bar, there are filters for 'Αναχώρηση', 'Δίκλινο', and 'Τιμή' (500€+). The results are sorted by 'Ταξινόμηση' (Οι προτάσεις μας). Two hotels are displayed:

Hotel Name	Rating	Location	Price (Trivago)	Price (Other Sites)
Λεάνδρος	3★ Ξενοδοχείο	Νέα Ρόδα, Κεντρική Μακεδονία, Ελλάδα	7.9 Καλό (184 κριτικές)	Expedia 174€, Hotels.com 175€
Villa Maria	Κατοικία / Διαμέρισμα	Πολύχρονο, Κεντρική Μακεδονία, Ελλάδα	8.9 Εξαιρετικό (42 κριτικές)	Booking.com 40€, Expedia 45€, Hotels.com 45€

Εικόνα 1 - Trivago - <https://www.trivago.gr/>

Έπειτα, μετά την αναζήτηση ο επισκέπτης μπορεί να δει πληροφορίες για το ξενοδοχείο που ενδιαφέρεται όπως εικόνες, πληροφορίες, κριτικές, τιμές, όπως φαίνεται παρακάτω:



Εικόνα 2 - Trivago - <https://www.trivago.gr/>

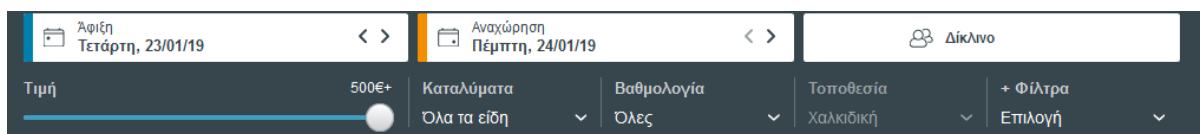
Δες προσφορές με:

Δωρεάν πρωινό Πληρωμή στο ξενοδοχείο Express Booking

Expedia	Junior Σουίτα, 1 Υπνοδωμάτιο, Θέα στη Θάλασσα - Express Booking >>> Δωρεάν πρωινό Δες 1 ακόμη προσφορά (περιλαμβάνει Δωρεάν πρωινό, Πληρωμή στο ξενοδοχείο)	174€ >>>
Hotels.com	Junior Σουίτα, 1 Υπνοδωμάτιο, Θέα στη Θάλασσα Δωρεάν πρωινό Δες 1 ακόμη προσφορά (περιλαμβάνει Δωρεάν πρωινό, Πληρωμή στο ξενοδοχείο)	175€ >
AMOMA.com	Προσωρινά το συγκεκριμένο site δεν παρέχει προσφορές	>
lol.travel	Προσωρινά το συγκεκριμένο site δεν παρέχει προσφορές	>

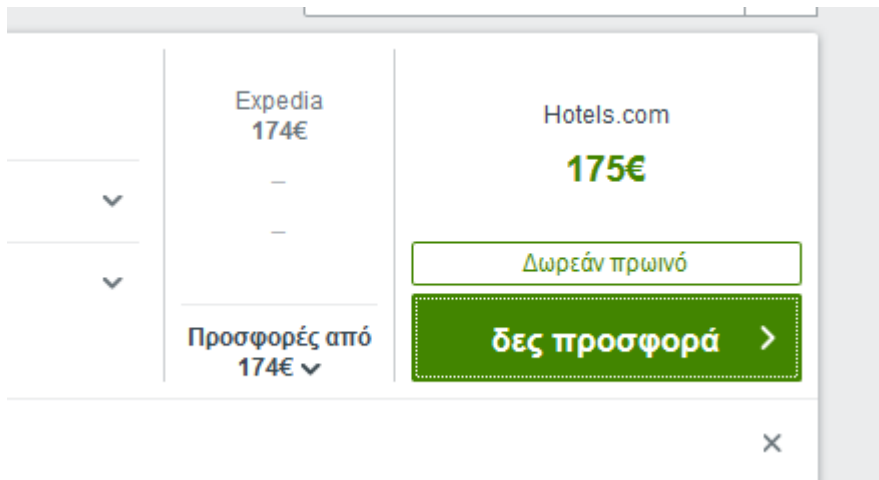
Εικόνα 3 - Trivago - <https://www.trivago.gr/>

Ακόμα, μπορεί να διαλέξει ημερομηνίες για διανυκτέρευση, να παραμετροποιήσει τον κέρσορα τιμών ανάλογα με τα ενδιαφέροντά του, να επιλέξει το είδος των καταλυμάτων ανάλογα με τα αστέρια που επιθυμεί να έχει το ξενοδοχείο, να επιλέξει βαθμολογία ξενοδοχείων που τον ενδιαφέρουν, καθώς και φίλτρα τα οποία πρόκειται για υπηρεσίες που παρέχει το ξενοδοχείο στους επισκέπτες του, όπως π.χ. δωρεάν wifi.



Εικόνα 4 - Trivago - <https://www.trivago.gr/>

Επιπρόσθετα, έχει την επιλογή με το κουμπί προσφορά να δει προσφορές για το ξενοδοχείο που επέλεξε.

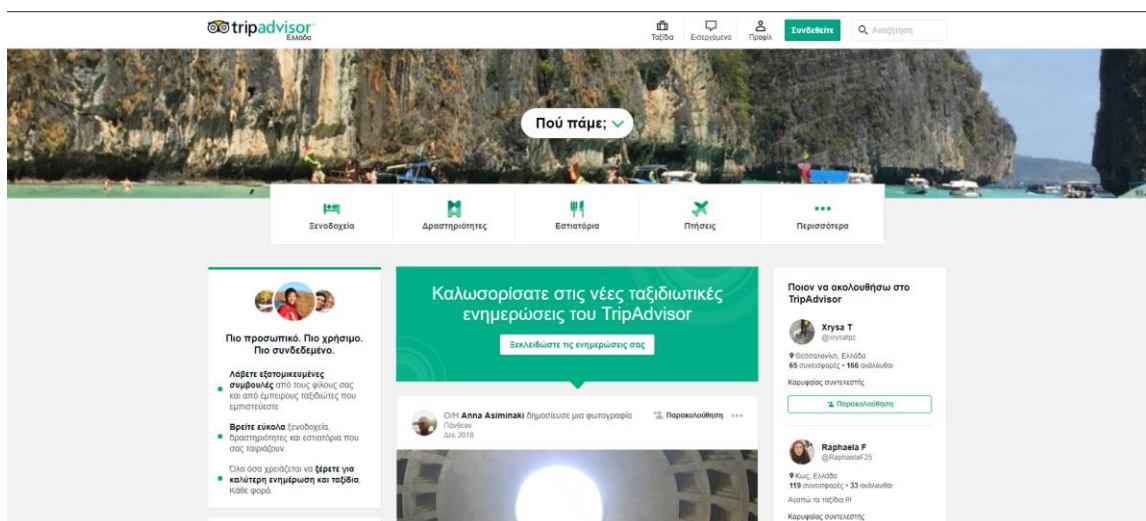


Εικόνα 5 - Trivago - <https://www.trivago.gr/>

Αυτό θα τον ανακατευθύνει στην πλατφόρμα Hotels.com, στην οποία μπορεί να δει περισσότερες πληροφορίες για το ξενοδοχείο που επέλεξε και να πραγματοποιήσει κράτηση ανάλογα τις ημερομηνίες που επιθυμεί.

3.1.2 tripadvisor.com

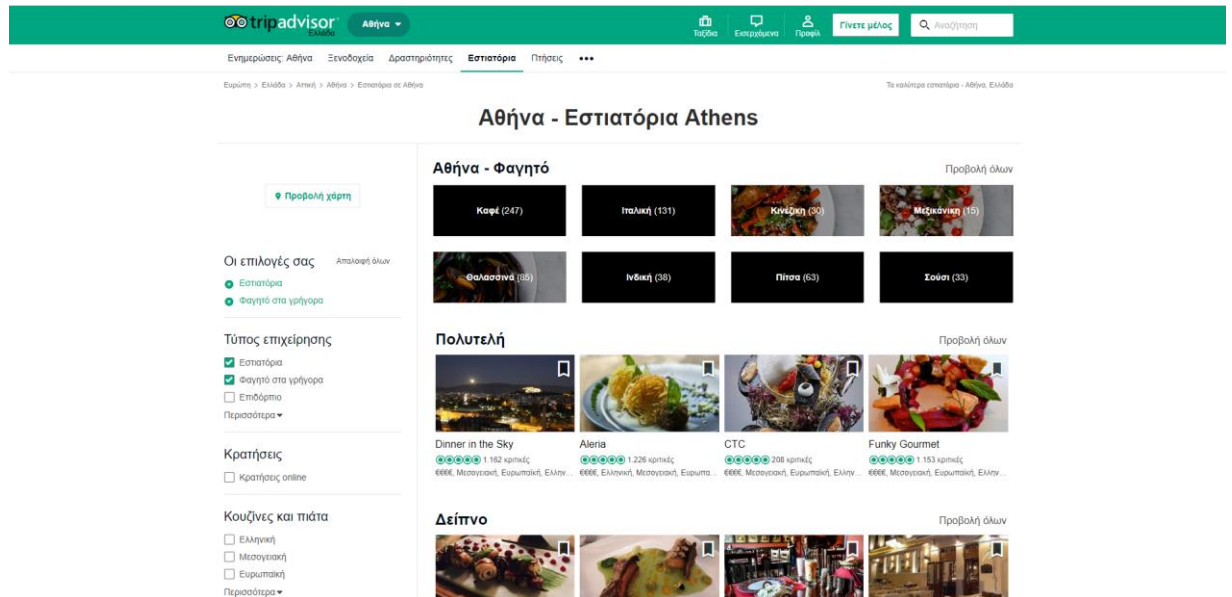
Σ’ αυτήν την ταξιδιωτική πλατφόρμα, ο επισκέπτης, μπορεί να επιλέξει ανάμεσα σε ξενοδοχεία, δραστηριότητες, εστιατόρια, πτήσεις κ.ά.



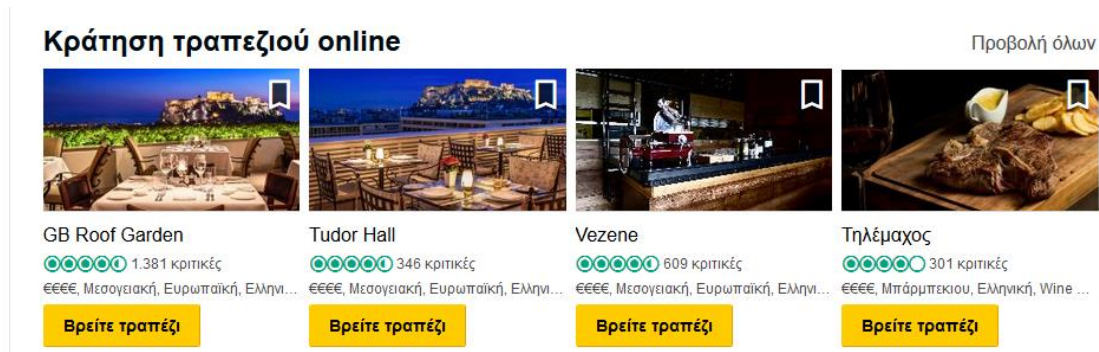
Εικόνα 6 - Tripadvisor - <https://www.tripadvisor.com.gr/>

Στην κατηγορία εστιατόρια, για την τοποθεσία που επέλεξε, μπορεί να δει εικόνες για φαγητό και να επιλέξει ανάμεσα σε υποκατηγορίες κουζινών όπως ιταλική, κινέζικη,

θαλασσινά κ.ά καθώς και μια γκάμα άλλων υποκατηγοριών φαγητού όπως πολυτελή, δείπνο, τοπική κουζίνα, μέσες τιμές, οικονομικά, καθώς και να πραγματοποιήσει online κράτηση τραπέζιού για το μαγαζί που τον ενδιαφέρει. Μπορεί ακόμα, να δει κριτικές και σχόλια από χρήστες που έχουν επισκεφτεί αυτά τα μέρη.



Εικόνα 7 - Tripadvisor - <https://www.tripadvisor.com.gr/>



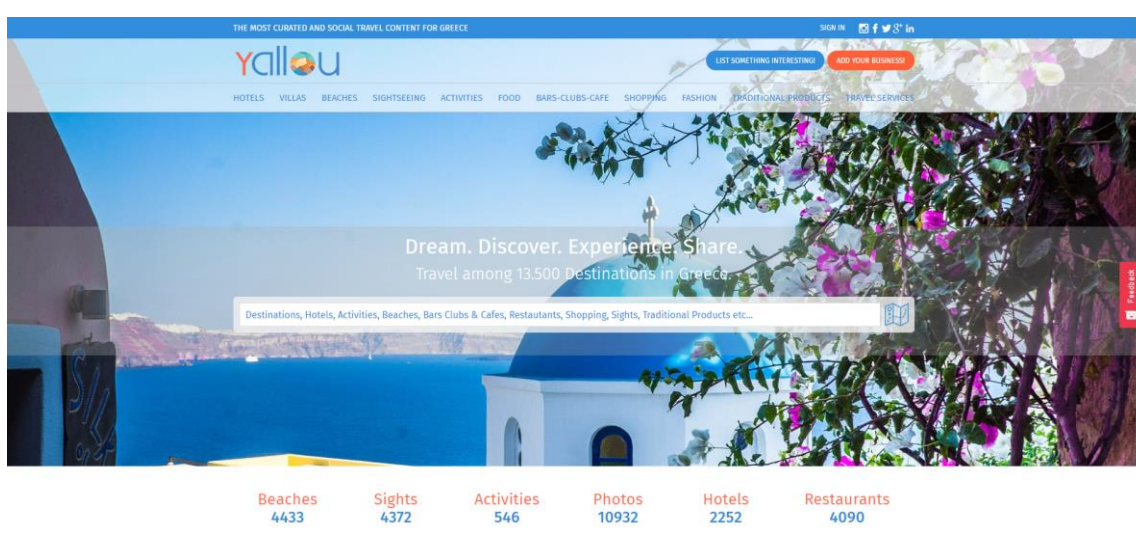
Εικόνα 8 - Tripadvisor - <https://www.tripadvisor.com.gr/>

Στην κατηγορία ξενοδοχεία, έχει περίπου τις ίδιες επιλογές με το trivago που αναφέρθηκε, με τον χρήστη σε περίπτωση που θέλει να πραγματοποιήσει κράτηση να ανακατευθύνεται στην πλατφόρμα booking, σε αντίθεση με το trivago που ο χρήστης πήγαινε στο Hotels.com .

Στην κατηγορία Δραστηριότητες είναι διαθέσιμες δραστηριότητες τύπου φύση και πάρκα, αρχαία ερείπια, μουσεία, μνημεία και αξιοθέατα, ιστορικές τοποθεσίες κ.ά, στα οποία πάλι μπορεί να δει κριτικές άλλων χρηστών και σχόλια.

3.1.3 yallou.com

Πρόκειται για μια ταξιδιωτική πλατφόρμα, η οποία περιλαμβάνει περιεχόμενο το οποίο μπορεί να αναζητήσει ένας επισκέπτης και αναφέρεται στην Ελλάδα. Τέτοιο περιεχόμενο είναι προορισμοί, ξενοδοχεία, δραστηριότητες, παραλίες, μπαρ, νυχτερινά κλαμπ, καφετέριες, εστιατόρια, ψώνια, αξιοθέατα, όπως επίσης παραδοσιακά προϊόντα της Ελλάδας.



Εικόνα 9 - Yallou - <https://yallou.com/>

Ακόμα, προσφέρει στον χρήστη μέσω γραφικών στοιχείων, προτεινόμενα προς τους χρήστες μέρη, από τα οποία μπορεί να δει φωτογραφίες και την τοποθεσία τους στον χάρτη, καθώς και να συμπληρώσει διάφορες φόρμες για επικοινωνία με τους διαχειριστές της πλατφόρμας και ένταξη της επιχείρησης των επισκεπτών στην πλατφόρμα για μελλοντική συνεργασία.

Recommended by Yallou



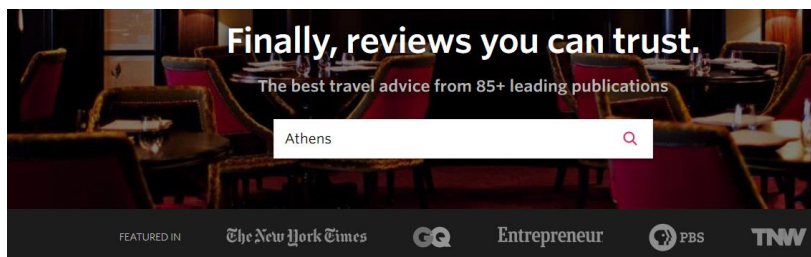
Εικόνα 10 - Yallou - <https://yallou.com/>



Εικόνα 11 - Yallou - <https://yallou.com/>


3.1.4 tripexpert.com

Η συγκεκριμένη πλατφόρμα αναζητεί και παρουσιάζει βαθμολογίες και σχόλια για ξενοδοχεία, εστιατόρια και αρχαιολογικά μουσεία για την τοποθεσία που επέλεξε ο χρήστης. Παρέχει την δυνατότητα στον χρήστη να βλέπει τις επιλογές του με ιστορικό από προηγούμενες αναζητήσεις του.



Εικόνα 12 - Tripexpert - <https://www.tripexpert.com/>





167 expert recommended hotels



Hotel Grande Bretagne, A Luxury Collection Hotel 96

Syntagma Square and Kolonaki

Originally built in 1842 for King Otto, the Grande Bretagne has always been Athens' most exclusive hotel. — *DK Eyewitness*





[All 17 reviews →](#)

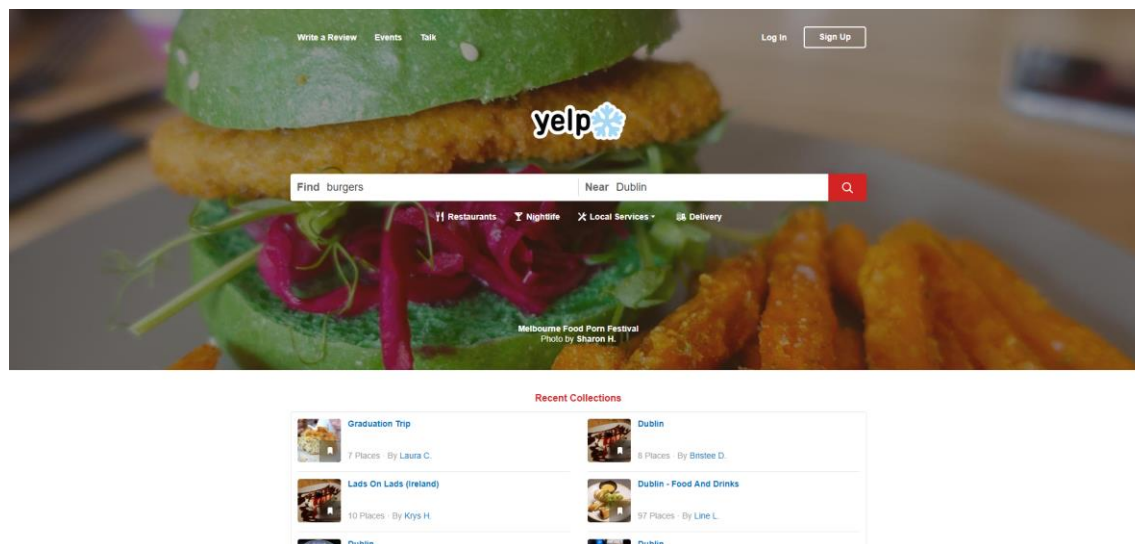
[Book now](#)

FROM \$353

Εικόνα 13 - Tripexpert - <https://www.tripexpert.com/>

3.1.5 yelp.com

Είναι αξιοσημείωτη η αναφορά στην συγκεκριμένη ταξιδιωτική πλατφόρμα, καθώς πρόκειται για ένα πολύ χρήσιμο εργαλείο, για τα άτομα που ενδιαφέρονται να μάθουν λεπτομέρειες για μέρη όπως εστιατόρια, μπαρ, φαγητό, delivery, φαγητό σε πακέτο, νυχτερινή ζωή καθώς και για μαγαζιά τα οποία δέχονται κρατήσεις.



The screenshot shows the Yelp website interface. At the top, there are links for "Write a Review", "Events", and "Talk". On the right, there are "Log In" and "Sign Up" buttons. The main content area features a search bar with "Find burgers" and "Near Dublin" entered. Below the search bar, there are navigation options: "Restaurants", "Nightlife", "Local Services", and "Delivery". A photo of a burger is visible in the background. At the bottom, there is a "Recent Collections" section with several items:

- Graduation Trip: 7 Places - By Laura C.
- Lads On Lads (Ireland): 10 Places - By Kays H.
- Dublin
- Dublin: 4 Places - By Bristee D.
- Dublin - Food And Drinks: 17 Places - By Line L.
- Dublin

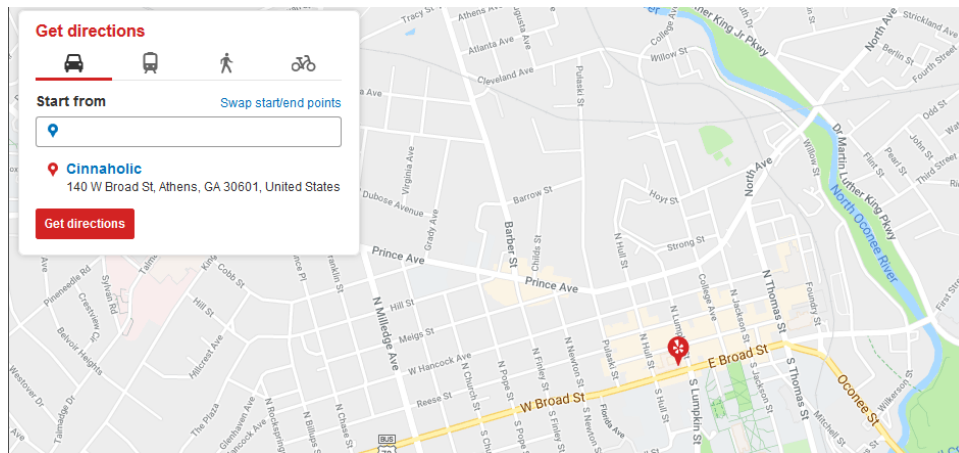
Εικόνα 14 - Tripexpert - <https://www.yelp.ie>

Μόλις πραγματοποιηθεί μια αναζήτηση από έναν επισκέπτη της πλατφόρμας π.χ. στην περίπτωση μας Coffee & Tea Shops κοντά στην περιοχή Αθήνα, του εμφανίζει διάφορα μέρη στην περιοχή που επέλεξε με βάση την κατηγορία επιλογής. Μπορεί να δει εικόνες, βαθμολογίες χρηστών καθώς και άλλες χρήσιμες πληροφορίες όπως οδός, τηλέφωνο, εικόνες και τοποθεσία του μαγαζιού στον χάρτη της google.

Εικόνα 15 - Tripexpert - <https://www.yelp.ie>

Εικόνα 16 - Tripexpert - <https://www.yelp.ie>

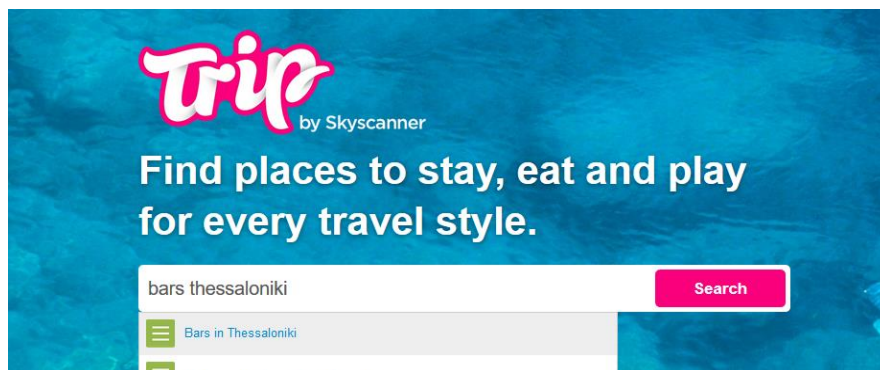
Επίσης οι χρήστες μπορούν να γράψουν την γνώμη τους σχετικά με το συγκεκριμένο μαγαζί, να προσθέσουν φωτογραφίες, καθώς και να επιλέξουν να πλοηγηθούν από την τοποθεσία που βρίσκονται στην τοποθεσία που στεγάζεται το μαγαζί με την βοήθεια του google maps.



Εικόνα 17 - Tripexpert - <https://www.yelp.ie>

3.1.6 trip.skyscanner.com


Σε αντίθεση με την επίσημη σελίδα της skyscanner, η οποία είναι για αναζητήσεις πτήσεων ξενοδοχείων και ενοικιάσεων αυτοκινήτου η πλατφόρμα trip.skyscanner.com, βασίζεται στις αναζητήσεις ενδιαφέροντος των χρηστών παγκοσμίως, όπως και κάποιες προηγούμενες πλατφόρμες που περιγράψαμε, προσθέτοντας περαιτέρω επιλογές. Για παράδειγμα, αναζητήσαμε bars στην περιοχή Θεσσαλονίκη.



Εικόνα 18 - Trip Skyscanner - <https://www.trip.skyscanner.com/>

Έπειτα εφόσον η αναζήτηση πραγματοποιήθηκε, ο χρήστης μπορεί να ενημερωθεί σχετικά με reviews, ratings, τηλέφωνο της επιχείρησης, ωράριο λειτουργίας, url της σελίδας στο facebook κ.ά.

3



FINOKIO Meze Bar

Katouni ke Agiou Mina, Thessaloniki 546 25, Greece


★★★★★ 2 reviews

Greek, Bar

Popular with Foodies

“ The food here is really delicious and there is also a nice selection of wines. Service is friendly. Finokio is also a great place to visit during the night as ... ”

4



Sherlock Speak Easy Bar

Leof. Nikis 25, Thessaloniki 546 23, Greece

★★★★★ 2 reviews

Bar

Popular with Foodies Nightlife

“ Sherlock is one of the most beautiful, creative and trendiest bars in Thessaloniki. It has a wide selection of great cocktails, perfect ambience for a cool night out and comfortable ... ”

Εικόνα 19 - Trip Skyscanner - <https://www.trip.skyscanner.com/>

FINOKIO Meze Bar

★★★★★ 2 reviews

Greek, Bar

Ranked #8 in Thessaloniki restaurants


Popular with:

9.8

Foodies

Write Review
Bookmark

Katouni ke Agiou Mina, Thessaloniki 546 25, Greece
 +30 231 038 5850
 Today, 12:00 PM - 5:00 AM Open Now
<https://www.facebook.com/FinokioMezeBar/>



Add Photo

Εικόνα 20 - Trip Skyscanner - <https://www.trip.skyscanner.com/>

Το ξεχωριστό με αυτήν την πλατφόρμα είναι ότι περιέχει διάφορες κατηγορίες για τους επισκέπτες της. Με λίγα λόγια, διαθέτει μια γκάμα κατηγοριών για παρουσίαση των χώρων οι οποίοι σχετίζονται με την συγκεκριμένη κατηγορία. Τέτοιες είναι: Adventure

Travelers, Art & Design Lovers, Backpackers, Foodies, Business Travelers και πολλές άλλες.

3.2 Πίνακας σύγκρισης των χαρακτηριστικών

	BotVoyage						
	1	2	3	4	5	6	Platform
Bot/Webchat	-	-	-	-	-	-	+
Google Login	+	+	-	+	+	-	+
Search Variety	-	+	-	-	-	+	+
Multilingual	-	+	-	-	+	-	+
Location	+	+	-	+	-	+	+
NLP	-	-	-	-	-	-	+
User Profile	+	+	+	-	+	+	+
Database	+	+	+	+	+	+	+
Search History	-	-	-	+	+	-	+
Recommendations	-	+	+	+	-	+	+
Friendly UI	+	+	-	+	+	+	+

3.2.1 Υπόμνημα Πίνακα – Επεξήγηση

- Bot/Webchat: Ύπαρξη bot (webchat), για την αναζήτηση κατηγοριών και αλληλεπίδραση με τον χρήστη.
- Google Login: Σύνδεση με google account.
- Search Variety: Πλήρης και ποικίλη αναζήτηση κατηγοριών σε κάθε επιθυμητή τοποθεσία.
- Multilingual: Να υποστηρίζει πλήθος γλωσσών.
- Location: Επιλογή τοποθεσίας.
- NLP: Δυνατότητα κατανόησης με τεχνητή νοημοσύνη φυσικής γλώσσας (Bot).
- User Profile: Διαμόρφωση προφίλ χρήστη.

- Database: Ύπαρξη βάσης δεδομένων για τους χρήστες.
- Search History: Ιστορικό αναζήτησης.
- Recommendations: Παρουσίαση των επικρατέστερων προορισμών.
- Friendly UI: Ευπαρουσίαστο και φιλικό περιβάλλον προς τον χρήστη.

3.3 Συμπεράσματα ανάλυσης

Μέσα από αυτήν την ανάλυση των ιστοτόπων που αναφέρθηκαν, καταλαβαίνει κανείς και την παρόμοια ροή που εφαρμόζεται σε αυτούς, αφού ο χρήστης πρέπει πρώτα να πληκτρολογήσει την τοποθεσία του προκειμένου να αναζητήσει τα μέρη που επιθυμεί. Αυτά τα μέρη σε κάποιες πλατφόρμες χωρίζονται και σε κατηγορίες για την καλύτερη πληροφόρηση των ενδιαφερόμενων. Το πιο κοινό στοιχείο είναι οι λεπτομερείς πληροφορίες που παρέχονται για τοποθεσίες, που σε κάποιες περιπτώσεις αυτό τείνει να είναι λίγο κουραστικό και χαοτικό για το ανθρώπινο μάτι. Επίσης, οι περισσότερες πλατφόρμες συνεργάζονται με ιστοχώρους οι οποίοι εξειδικεύονται στην κράτηση ή αλλιώς booking διαφόρων κατηγοριών όπως κράτηση τραπεζιού, κράτηση αυτοκινήτου, κράτηση πτήσεων, κράτηση εισιτηρίων για συγκεκριμένα μέρη όπως αρχαιολογικά μουσεία. Αξίζει να σημειωθεί, ότι η παρουσία ενός διαδικτυακού ρομπότ είναι ελλιπής και αυτό είναι ένα μεγάλο μειονέκτημα γι' αυτούς τους ιστοχώρους καθώς θα μπορούσαν να προσελκύσουν περισσότερους επισκέπτες με την ύπαρξη αυτής της λειτουργίας στις υπηρεσίες παροχής τους. Δεν αναφερόμαστε σε μπουτς τα οποία δίνουν πληροφορίες για το πως να πραγματοποιήσει μια αναζήτηση ο χρήστης στους ιστοχώρους τους, αλλά σε μπουτς τα οποία είναι ικανά έτσι ώστε να πραγματοποιήσουν από μόνα τους αυτήν την αναζήτηση. Τα περισσότερα sites που παρέχουν μπουτ τέτοιων λειτουργιών, φροντίζουν στην πλαισίωσή τους στην πλατφόρμα messenger αντί των ιστοτόπων τους.

ΚΕΦΑΛΑΙΟ 4 - ΜΕΘΟΔΟΛΟΓΙΑ ΚΑΙ ΕΡΓΑΛΕΙΑ

4.1 Σύλληψη και διαμόρφωση ιδέας

Αρχικά, η έμπνευση της ιδέας προήλθε από την διπλωματική που υλοποίησαν οι φοιτητές Κωνσταντίνος Μπολότης και Αλέξανδρος Φαΐτασιδης με ονομασία BotVoyage, τον Ιούνιο του 2018. Η συγκεκριμένη εφαρμογή στεγαζόταν αποκλειστικά στην πλατφόρμα του messenger έτσι κάποιος χρήστης για να την χρησιμοποιήσει θα έπρεπε και εξ' ολοκλήρου να είναι χρήστης της εφαρμογής messenger είτε στο κινητό του είτε από τον υπολογιστή του. Η συγκεκριμένη διπλωματική εργασία μας κέντρισε ιδιαίτερα το ενδιαφέρον, καθώς ως φοιτητές του τμήματος ΜΠΕΣ και χρήστες του διαδικτύου χρησιμοποιούμε και εμείς τουριστικές πλατφόρμες για την πληροφόρησή μας πάνω σε θέματα ταξιδιών. Μετά την παρακολούθηση της εν λόγω διπλωματικής και μιας εκτενής έρευνας πάνω στις τουριστικές πλατφόρμες, η πρώτη μας ερώτηση ήταν γιατί στο messenger και όχι αυτόνομο λογισμικό; Η ίδια ερώτηση ήρθε να δώσει απάντηση και στα διλήμματα μας περί απουσίας αυτοματοποιημένου λογισμικού και στις ταξιδιωτικές πλατφόρμες που χρησιμοποιούσαμε, καθώς πρόκειται για μια εξαιρετική υπηρεσία παροχής πληροφοριών. Έπειτα, όπως παρατηρήσαμε, η εφαρμογή BotVoyage περιείχε λίγες κατηγορίες τις οποίες μπορούσε να αναζητήσει κάποιος χρήστης και αυτές προοριζόντουσαν για μαγαζιά, νοσοκομεία, καθώς και πτήσεις. Ένας χρήστης ο οποίος θα ενδιαφερόταν για ασχολίες προκειμένου να περάσει τον ελεύθερό του χρόνο σε κάποια περιοχή, δεν είχε και πολλές επιλογές. Έτσι, αποφασίσαμε την επέκταση της συγκεκριμένης εφαρμογής σε πλήθος κατηγοριών οι οποίες θα άφηναν έναν μέσο χρήστη ευχαριστημένο. Αυτό το καταφέραμε κυρίως αναλύοντας και ερευνώντας το documentation από το Botframework καθώς και των κατηγοριών τις οποίες προσφέρει το Foursquare API. Ένα ακόμη δίλημμα, ήταν η παρουσίαση φωτογραφιών, βαθμολογιών, οδών, τηλεφώνων για τις συγκεκριμένες κατηγορίες τις οποίες θα αναζητούσε κάποιος επισκέπτης της εφαρμογής αλλά και σχολίων άλλων χρηστών οι οποίοι έχουν επισκεφτεί τα συγκεκριμένα μέρη. Αυτό επιτεύχθηκε μέσω διαφόρων requests προς το Foursquare, τα οποία στάλθηκαν μέσω από ένα πλήθος πακέτων που μας παρέχει το ReactJS. Επίσης, η δημιουργία μιας βάσης δεδομένων για τους χρήστες οι οποίοι χρησιμοποιούν την εφαρμογή και η διαδικασία αυθεντικοποίησης και εξουσιοδότησης των χρηστών, προκειμένου να έχουν πρόσβαση στην πλατφόρμα στην οποία θα στεγάζεται το μποτ, αποτελούσαν σημαντικές λύσεις για μια καλύτερη και λειτουργικότερη εφαρμογή. Η

συγκεκριμένη λειτουργία υλοποιήθηκε μέσω της βάσης δεδομένων Realtime Database, μετά από μια εκτενής έρευνα στο documentation του firebase και επικοινωνώντας μαζί της μέσω διαφόρων πακέτων που αναφέρθηκαν και παραπάνω τα οποία μας προσφέρει το ReactJS.

4.2 Πλατφόρμες και εργαλεία

Παρακάτω, θα προχωρήσουμε στην ανάλυση των γλωσσών προγραμματισμού, των εργαλείων και πλατφορμών με μία σχετική αναδρομή, επεξήγηση και γενικότερη χρήση τους. Σε πρώτο επίπεδο, θα αναλυθούν τα βασικά τμήματα που θεμελιώνουν την εφαρμογή μας, οι γλώσσες δηλαδή που χρησιμοποιήθηκαν, στη συνέχεια σε ποιες πλατφόρμες, frameworks που αναπτύχθηκαν και τέλος με ποια εργαλεία παραμετροποιήθηκαν.

4.3 JavaScript

4.3.1 Ιστορική αναδρομή

Το 1995, η Netscape ανακοίνωσε την JavaScript, παρουσιάζοντάς την, ως μία πολύ εύκολη λύση όσον αφορά τη σχεδίαση live online εφαρμογών, οι οποίες διασυνδέουν αντικείμενα και πόρους και στους clients και στους servers. Ως αποτέλεσμα, αποτέλεσε την πρώτη επιλογή των προγραμματιστών για την ανάπτυξη client-side εφαρμογών μες στο πλαίσιο των προγραμμάτων περιήγησης στο διαδίκτυο.

Ωστόσο, το 1995 επίσης με την κυκλοφορία του Netscape Enterprise Server, η JavaScript άρχισε να εφαρμόζεται και στη διαμόρφωση της πλευράς των διακομιστών (server-side). Στη συνέχεια, το 1996 συμπεριλήφθηκε στον IIS(διακομιστής διαδικτύου που υποστηρίζει τα HTTP και FTP πρωτόκολλα) και παρουσιάστηκε με το όνομα Jscript σε ASP και .NET σελίδες.

Τέλος, στα μέσα της δεκαετίας του 2000, παρουσιάστηκαν διάφορες καινούριες server-side εφαρμογές της JavaScript, με διασημότερη πλέον το Node.js, το οποίο και θα αναλυθεί σε επόμενη επικεφαλίδα.

4.3.2 Τι είναι

Η JavaScript, η οποία συνήθως συναντάται με τη συντομογραφία JS, είναι μία αντικειμενοστραφής, ελαφριά γλώσσα προγραμματισμού, η οποία υποστηρίζει την ύπαρξη συναρτήσεων πρώτης κλάσης(δυνατότητα περάσματος συναρτήσεων ως όρισμα σε άλλη συνάρτηση) και είναι κυρίως γνωστή ως scripting γλώσσα για ιστοσελίδες. Ωστόσο, χρησιμοποιείται σε πολλά περιβάλλοντα, τα οποία δε βασίζονται στην ύπαρξη προγράμματος περιήγησης (browser).

Πρόκειται για μια βασισμένη σε πρότυπα(αξιοποιεί την έννοια της κληρονομικότητας) πολλαπλών παραδειγμάτων scripting γλώσσα προγραμματισμού, που είναι δυναμική, και υποστηρίζει αντικειμενοστραφή, επιτακτικά(χρήση εντολών για αξιοποίηση από τον υπολογιστή) και λειτουργικά στυλ προγραμματισμού. Δηλαδή, functional programming κώδικα με σειριακή εκτέλεση.

4.4 Javascript Frameworks

Τα τρία Frameworks της Javascript στα οποία εστίασαμε και συγκρίναμε προκειμένου να διαλέξουμε, έτσι ώστε να δημιουργήσουμε το project, είναι τα: ReactJS, AngularJS και Vue.JS

4.4.1 ReactJS

4.4.1.1 Ιστορική Αναδρομή

Η βιβλιοθήκη React, δημιουργήθηκε από τον Jordan Walke, μηχανικό λογισμικού του Facebook. Επηρεασμένος από το XHP, την HTML και την PHP. Χρησιμοποιήθηκε για πρώτη φορά στο newsfeed του Facebook το 2011 και αργότερα στο Instagram το 2012. Το 2013 έγινε open-sourced στην JSConf στις ΗΠΑ. Η React Native, η οποία επιτρέπει την ανάπτυξη native εφαρμογών σε Android, iOS και UWP, δημοσιεύθηκε στο React.js Conf της Facebook τον Φεβρουάριο του 2015 και έγινε open-sourced τον Μάρτιο της ίδιας χρονιάς. Στις 18 Απριλίου του 2017 η Facebook ανακοίνωσε το νέο βασικό Framework Library React Fiber, για την ανάπτυξη User Interfaces. Το Fiber θα αποτελέσει το θεμέλιο για μελλοντικές βελτιώσεις και ανάπτυξη των χαρακτηριστικών του React Framework.

4.4.1.2 Χρήση

Η React (επίσης γνωστή ως React.js ή ReactJS) είναι μια βιβλιοθήκη της γλώσσας προγραμματισμού Javascript, για την δημιουργία User Interfaces(διεπαφών για χρήστες). Διατηρείται από το Facebook και από μια κοινότητα μεμονωμένων προγραμματιστών και εταιρειών. Η React, μπορεί να χρησιμοποιηθεί ως βάση, για την ανάπτυξη εφαρμογών μιας σελίδας ή κινητών. Οι εφαρμογές Complex React, συνήθως απαιτούν την χρήση πρόσθετων βιβλιοθηκών για state management, routing και αλληλεπίδραση με κάποιο API.

4.4.1.3 ReactJS - Χαρακτηριστικά στοιχεία

- One-Way data binding with props

Τα properties (props), περνάνε στα Child Components, από τα Parent Components. Τα Components, δέχονται τα props ως ένα ενιαίο σύνολο immutable (αμετάβλητων) τιμών (π.χ. ένα javascript αντικείμενο).

- Stateful Components

Τα States, διατηρούν τις τιμές μέσω των Components και μπορούν να ενσωματωθούν στα child components μέσω των props.

- Virtual DOM

Ένα άλλο αξιοσημείωτο χαρακτηριστικό, είναι η χρήση του Virtual Document Object Model ή αλλιώς Virtual DOM. Η React, δημιουργεί μια προσωρινή μνήμη δομής δεδομένων μέσα στην cache, υπολογίζοντας τις προκύπτουσες διαφορές και στην συνέχεια ενημερώνει αποτελεσματικά το DOM, που εμφανίζεται στο πρόγραμμα περιήγησης (browser). Αυτό, επιτρέπει στον προγραμματιστή να γράφει κώδικα σαν να επαναφορτίζεται ολόκληρη η σελίδα σε κάθε αλλαγή, ενώ οι βιβλιοθήκες της React φορτώνουν μόνο τα υποδεέστερα components που πραγματικά αλλάζουν.

- Lifecycle-Methods

Τα Lifecycle Methods είναι hooks, τα οποία επιτρέπουν την εκτέλεση κώδικα σε καθορισμένα σημεία κατά την διάρκεια ζωής ενός component.

`shouldComponentUpdate()`: Επιτρέπει στον προγραμματιστή να αποτρέψει την περιττή μεταφόρτωση ενός component επιστρέφοντας `false` αν μια φόρτωση δεν απαιτείται.

`componentDidMount()`: Καλείται όταν ένα component γίνει mount (το component δημιουργείται στο User interface, συχνά συνδέοντάς το με ένα DOM node). Συνήθως χρησιμοποιείται για να ενεργοποιήσει την φόρτωση δεδομένων από μια απομακρυσμένη πηγή μέσω ενός API.

`render()`: Η `render` είναι η πιο σημαντική μέθοδος lifecycle και είναι η μόνη που απαιτείται στα components. Συνήθως, καλείται κάθε φορά που ενημερώνονται τα state των component, αντικατοπτρίζοντας τις αλλαγές στο περιβάλλον εργασίας του χρήστη (User Interface).

- JSX

Η JSX(JavaScript XML) είναι μια επέκταση της σύνταξης της JavaScript. Παρόμοια σε εμφάνιση με την HTML, παρέχει έναν τρόπο για την φόρτωση των component, χρησιμοποιώντας αυτήν την σύνταξη που είναι γνωστή σε πολλούς προγραμματιστές. Τα React components, μπορούν να γραφτούν σε JSX, παρόλο που δεν είναι αναγκαίο. Μπορούν να γραφούν και σε καθαρή Javascript. Η JSX, είναι παρόμοια με μια άλλη σύνταξη-επέκταση που δημιουργήθηκε από την Facebook για τις PHP και XHP.

- Nested Elements

Πολλαπλά στοιχεία που βρίσκονται στο ίδιο επίπεδο πρέπει να ενσωματωθούν σε ένα στοιχείο container (πλαίσιο), όπως είναι το `<div>` στοιχείο (element).

- Attributes

Η JSX παρέχει μια σειρά χαρακτηριστικών στοιχείων που έχουν σχεδιαστεί για να αντικατοπτρίζουν αυτά που παρέχονται από την HTML. Επίσης, σε ένα component μπορούν να περάσουν και προσαρμοσμένα attributes. Όλα τα attributes θα ληφθούν από το component ως props.

- JavaScript expressions

Τα JavaScript expressions (αλλά όχι τα statements) μπορούν να χρησιμοποιηθούν μέσα σε JSX με ‘καμπυλωτές’ αγκύλες ({}).

- Conditional statements

Οι εντολές if-else δεν μπορούν να χρησιμοποιηθούν μέσα σε JSX κώδικα, αλλά μπορούν να χρησιμοποιηθούν σε εναλλακτικές εκφράσεις.

Οι μέθοδοι και η JSX μπορούν να χρησιμοποιηθούν σε όρους. Ο κώδικας που γράφεται σε JSX, απαιτεί μετατροπή σε ένα εργαλείο όπως η Babel πριν να καταστεί κατανοητός από τους web browsers. Αυτή η επεξεργασία, πραγματοποιείται κατά την διάρκεια μιας διαδικασίας ενός λογισμικού, πριν την ανάπτυξη μιας εφαρμογής.

- Architecture beyond HTML

Η βασική αρχιτεκτονική της React βιβλιοθήκης ισχύει και πέρα από την φόρτωση HTML στους browsers. Για παράδειγμα, το Facebook έχει δυναμικά διαγράμματα τα οποία αποδίδονται σε <canvas> ετικέτες (tags). Ακόμα, το Netflix και το PayPal χρησιμοποιούν καθολική φόρτιση για να κάνουν render ταυτόσημη HTML, τόσο στον server, όσο και στον client.

4.4.2 VueJS

Η Vue.js διαθέτει μια προοδευτικά υιοθετήσιμη αρχιτεκτονική που επικεντρώνεται στο αποδοτικό rendering και στην σύνθεση των component. Διαθέτει λειτουργίες οι οποίες απαιτούνται για σύνθετες λειτουργίες όπως το routing (δρομολόγηση), το state management, και την κατασκευή εργαλείων. Όλα τα παραπάνω προσφέρονται μέσω επίσημα διατηρουμένων βιβλιοθηκών και πακέτων.

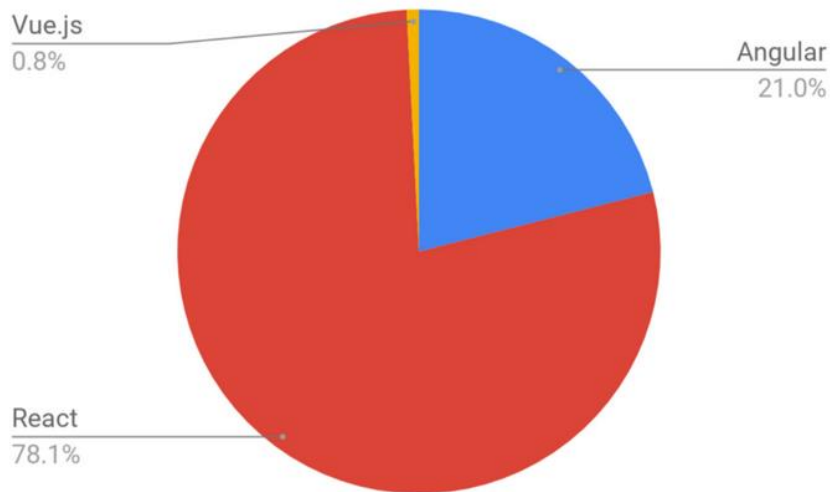
4.4.3 AngularJS

Το AngularJS βασίζεται στην πεποίθηση ότι ο δηλωτικός προγραμματισμός πρέπει να χρησιμοποιείται για τη δημιουργία διεπαφών χρήστη και τη σύνδεση στοιχείων λογισμικού, ενώ ο επιτακτικός προγραμματισμός είναι περισσότερο κατάλληλος για

τον ορισμό της επιχειρησιακής λογικής μιας εφαρμογής. Το πλαίσιο προσαρμόζει και επεκτείνει την παραδοσιακή HTML για να παρουσιάσει το δυναμικό περιεχόμενο μέσω αμφίδρομης δέσμησης δεδομένων που επιτρέπει τον αυτόματο συγχρονισμό μοντέλων και προβολών. Ως αποτέλεσμα, το AngularJS υπογραμμίζει τη σαφή χειραγώγηση του μοντέλου αντικειμένων εγγράφου (DOM Documentation Object Model Modeling), με στόχο τη βελτίωση της δοκιμής και της απόδοσης.

4.4.4 Ζήτηση των τριών Frameworks

Τα Javascript Frameworks αναπτύσσονται με εξαιρετικά γρήγορο ρυθμό, γεγονός που σημαίνει ότι σήμερα έχουμε συχνά ενημερωμένες εκδόσεις των Angular, ReactJS και Vue.js. Σύμφωνα με την πηγή Indeed.com όσων αφορά τον αριθμό ανοικτών θέσεων εργασίας που απαιτούν συγκεκριμένη γνώση ενός εκ των τριών Javascript Frameworks τα ποσοστά ζήτησης έχουν ως εξής:



(**Σημείωση:** Το ακόλουθο διάγραμμα αναφέρεται σε περισσότερες από 60.000 προσφορές εργασίας.)

Εικόνα 21 - ReactJS VueJS AngularJS
<https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d>

4.4.5 Πλεονεκτήματα και μειονεκτήματα των Javascript Frameworks

4.4.5.1 ReactJs

Πλεονεκτήματα

- Εύκολο στην εκμάθηση. Αυτό συμβαίνει λόγω της απλότητάς του από πλευράς σύνταξης. Οι προγραμματιστές, πρέπει μόνο να ανακαλέσουν τις δεξιότητες γραφής της HTML και τίποτα άλλο. Δεν χρειάζεται βαθιά γνώση του TypeScript όπως στο Angular.
- Υψηλό επίπεδο ευελιξίας και βέλτιστη ανταπόκριση.
- Το Virtual DOM επιτρέπει την ταξινόμηση εγγράφων σε μορφές HTML, XHTML ή XML σε ένα tree, το οποίο είναι καλύτερα αποδεκτό από προγράμματα περιήγησης ιστού κατά το πέρασμα διαφορετικών στοιχείων των Web Applications.
- Σε συνδυασμό με τα ES6-ES7, το ReactJS, μπορεί να λειτουργήσει με εύκολο τρόπο παρόλο τον υψηλό φόρτο εργασιών.
- Downward Data Binding, που σημαίνει ότι με αυτό το είδος ροής δεδομένων τα child elements δεν μπορούν να επηρεάσουν τα parent elements.
- 100% ανοιχτού κώδικα JavaScript βιβλιοθήκη που παίρνει πολλές καθημερινές ενημερώσεις και βελτιώσεις σύμφωνα με τη συμβολή των προγραμματιστών από όλο τον κόσμο.
- Είναι απολύτως ελαφρύ, επειδή τα δεδομένα που εκτελούνται από την πλευρά του User μπορούν να εκπροσωπηθούν εύκολα από την πλευρά του Server ταυτόχρονα.
- Η μετεγκατάσταση μεταξύ των εκδόσεων είναι γενικά πολύ εύκολη, με το Facebook να παρέχει "codemods" για να αυτοματοποιήσει ένα μεγάλο μέρος της διαδικασίας.

Μειονεκτήματα

- Έλλειψη επίσημου Documentation - η γρήγορη ανάπτυξη του ReactJS δεν αφήνει χώρο για το κατάλληλο Documentation, το οποίο είναι λίγο χαοτικό τώρα που πολλοί προγραμματιστές συμβάλλουν στην ανάπτυξή του μεμονωμένα χωρίς συστηματική προσέγγιση.
- Το ReactJS απαιτεί βαθιά γνώση του τρόπου ενσωμάτωσης της διεπαφής χρήστη στο MVC Framework.

4.4.5.2 AngularJS

Πλεονεκτήματα

- Νέες λειτουργίες όπως το ενισχυμένο RXJS, ταχύτερη μεταγλώττιση (σε λιγότερο από 3 δευτερόλεπτα), νέα κυκλοφορία του HttpClient.
- Λεπτομερής Documentation, που επιτρέπει τη λήψη όλων των απαραίτητων πληροφοριών για τον προγραμματιστή που δουλεύει μόνος του, χωρίς να ρωτάει τους συναδέλφους του. Ωστόσο, αυτό απαιτεί περισσότερο χρόνο για εκπαίδευση.
- Αμφίδρομη σύνδεση δεδομένων που επιτρέπει μοναδική συμπεριφορά για την εφαρμογή, η οποία ελαχιστοποιεί τους κινδύνους πιθανών σφαλμάτων.
- MVVM (Model-View-ViewModel), που επιτρέπει στους προγραμματιστές να δουλεύουν ξεχωριστά στην ίδια ενότητα εφαρμογών χρησιμοποιώντας το ίδιο σύνολο δεδομένων.

Μειονεκτήματα

- Η σύνθετη σύνταξη που προέρχεται από την πρώτη έκδοση του Angular. Παρ'όλα αυτά, το Angular 5 χρησιμοποιεί το TypeScript 2.4 το οποίο είναι το λιγότερο δύσκολο.
- Θέματα μετεγκατάστασης που μπορούν να εμφανιστούν κατά τη μετάβαση από την παλαιότερη έκδοση στις πιο πρόσφατες.

4.4.5.3 VueJS

Πλεονεκτήματα

- Empowered HTML. Αυτό σημαίνει ότι το Vue.js έχει πολλά παρόμοια χαρακτηριστικά με το Angular και αυτό μπορεί να βοηθήσει στη βελτιστοποίηση του χειρισμού των μπλοκ HTML, με τη χρήση διαφορετικών Components.
- Λεπτομερές Documentation. Το Vue.js έχει πολύ περιστασιακό Documentation, που μπορεί να προσαρμόσει την εκμάθηση για τους προγραμματιστές και να σώσει πολύ χρόνο για την ανάπτυξη μιας εφαρμογής, χρησιμοποιώντας μόνο τις βασικές γνώσεις HTML και JavaScript.
- Ικανότητα προσαρμογής. Παρέχει μια γρήγορη περίοδο αλλαγής από άλλα Frameworks σε Vue.js, λόγω της ομοιότητας με το Angular and React, από άποψη σχεδιασμού και αρχιτεκτονικής.
- Εξαιρετική ενσωμάτωση. Το Vue.js μπορεί να χρησιμοποιηθεί τόσο για την κατασκευή εφαρμογών μιας σελίδας, όσο και για πιο δύσκολες διεπαφές ιστού εφαρμογών. Το κυριότερο είναι ότι τα μικρότερα διαδραστικά μέρη μπορούν εύκολα να ενσωματωθούν στην υπάρχουσα υποδομή χωρίς αρνητικές επιπτώσεις σε ολόκληρο το σύστημα.

- Μικρό μέγεθος. Το Vue.js μπορεί να έχει βάρος περίπου 20KB, διατηρώντας την ταχύτητα και την ευελιξία του, κάτι το οποίο επιτρέπει την επίτευξη πολύ καλύτερων επιδόσεων σε σύγκριση με άλλα Frameworks.

Μειονεκτήματα

- Έλλειψη πηγών. Το Vue.js εξακολουθεί να έχει ένα αρκετά μικρό μερίδιο αγοράς σε σύγκριση με το React ή το Angular, πράγμα που σημαίνει ότι η ανταλλαγή γνώσεων σε αυτό το Framework βρίσκεται ακόμα στην αρχή.
- Κίνδυνος υπερβολικής ελαστικότητας. Μερικές φορές, το Vue.js μπορεί να έχει προβλήματα ενώ ενσωματώνει τεράστια έργα και δεν υπάρχει ακόμη εμπειρία με πιθανές λύσεις, αλλά σίγουρα θα έρθουν σύντομα.
- Έλλειψη πλήρους αγγλικής τεκμηρίωσης. Αυτό οδηγεί σε μερική πολυπλοκότητα σε ορισμένα στάδια ανάπτυξης, ωστόσο όλο και περισσότερα υλικά μεταφράζονται στα αγγλικά.

4.5 Node.js

4.5.1 Τι είναι

Το Node.js είναι με απλά λόγια μία πλατφόρμα, η οποία χρησιμοποιείται για ανάπτυξη server-side JavaScript. Ειδικότερα, είναι μία μηχανή JavaScript, η οποία είναι θεμελιωμένη πάνω στην V8 JavaScript Engine του Chrome Browser. Για την καλύτερη κατανόηση, ας εξηγήσουμε τι είναι η συγκεκριμένη μηχανή.

Η εν λόγω μηχανή, όπως κάθε JavaScript engine, έχει ως στόχο την εκτέλεση του κώδικα που είναι γραμμένος σε JavaScript και η μετατροπή του στην πιο βελτιστοποιημένη μορφή, ώστε είτε να μπορεί να ερμηνευτεί από το πρόγραμμα περιήγησης, είτε ακόμα να ενσωματώνεται σε μία εφαρμογή. Το παράδοξο με την V8, είναι το γεγονός ότι ο πηγαίος κώδικάς της, έχει γραφτεί σε C++. Η χρήση της έγκειται στον Google Chrome και φυσικά στο Node.js, αλλά και σε άλλα περιβάλλοντα.

Η πλατφόρμα του Node.js, χρησιμοποιεί ένα non-blocking I/O μοντέλο, το οποίο επιτρέπει στον κώδικα να έχει ευελιξία και ταχύτητα, διότι δεν υπάρχει απλώς ένα thread

το οποίο να τρέχει σειριακά και να ελέγχει όλες τις λειτουργίες. Στο υπόβαθρο του Node.js, συναντάται μία πολυνηματική αρχιτεκτονική, μέσω της οποίας δεν υπάρχει η πιθανότητα το πρόγραμμα όπως γίνεται κατανοητό κι από την παραπάνω λέξη(non-blocking) να μπλοκαριστεί, είτε λόγω ενός λάθους κατά την εκτέλεση, είτε λόγω μιας γενικής καθυστέρησης κατά την εκτέλεση ενός request, διότι απλώς θα απασχολείται το συγκεκριμένο τμήμα κι όχι όλο το υπόλοιπο κομμάτι του λογισμικού, μιας και σε κάθε thread έχει ανατεθεί μία λειτουργία, διεργασία. Το εν λόγω μοντέλο, εύλογα προσδίδει στην πλατφόρμα τον χαρακτηρισμό ασύγχρονη.

Μέσω του Node.js, διατίθεται το μεγαλύτερο οικοσύστημα πακέτων ανοιχτού λογισμικού, το οποίο όπως αναφέραμε παραπάνω, προσφέρει βιβλιοθήκες κώδικα προς παραμετροποίηση και εκ νέου επέκταση.

Συμπερασματικά, το Node.js αποτελεί μία επέκταση της V8 JavaScript Engine, συνιστώντας ένα περιβάλλον διακομιστή, προσδίδοντας στον developer τη δυνατότητα να αναπτύξει server-side JavaScript λογισμικό με ταχύτητα, ευελιξία και ένα υπέρογκο πλήθος πακέτων, μέσω του npm.

4.6 Firebase (API)



Firestore

Εικόνα 22 - Firebase - <https://firebase.google.com/>

4.6.1 Τι είναι

Το Firebase είναι μια πλατφόρμα ανάπτυξης εφαρμογών για κινητά και web, που αναπτύχθηκε από την Firebase το 2011 και στη συνέχεια εξαγοράστηκε από την Google το 2014. Από τον Οκτώβριο του 2018, η πλατφόρμα Firebase διαθέτει 18 υπηρεσίες που χρησιμοποιούνται από 1,5 εκατομμύρια εφαρμογές.

4.6.2 Ιστορική Αναδρομή

Το Firebase εξελίχθηκε από το Envolv, που ίδρυσαν ο James Tamplin και ο Andrew Lee το 2011. Το Envolv παρέχει στους προγραμματιστές ένα API, που επιτρέπει την ενσωμάτωση της δυνατότητας online συνομιλίας στις ιστοσελίδες τους. Αργότερα, οι Tamplin και Lee διαπίστωσαν ότι το API αυτό, χρησιμοποιείται για να διαβιβάζει δεδομένα εφαρμογών που δεν ήταν μηνύματα συνομιλίας. Οι προγραμματιστές χρησιμοποιούν το Envolv για να συγχρονίσουν δεδομένα εφαρμογών, όπως η κατάσταση των παιχνιδιών σε πραγματικό χρόνο στους χρήστες τους. Έπειτα, οι δυο παραπάνω ιδρυτές αποφάσισαν να χωρίσουν το σύστημα συνομιλίας από την αρχιτεκτονική του API. Ίδρυσαν την Firebase, ως ξεχωριστή εταιρεία τον Απρίλιο του 2012.

Η Firebase αύξησε την χρηματοδότηση των πόρων της τον Μάιο του 2012. Η εταιρεία αύξησε περαιτέρω την χρηματοδότηση της Σειράς A τον Ιούνιο του 2013. Τον Οκτώβριο του 2014, η Firebase εξαγοράστηκε από την Google. Τον Οκτώβριο του 2015 η Google αγόρασε το Divshot για να το συγχωνεύσει με το Firebase. Από την εξαγορά, το Firebase έχει αναπτυχθεί μέσα στο Google και έχει επεκτείνει τις υπηρεσίες του, έτσι ώστε να γίνει μια ενοποιημένη πλατφόρμα για κινητά. Το Firebase ενσωματώνεται τώρα με διάφορες άλλες υπηρεσίες της Google, για να προσφέρει ευρύτερα προϊόντα στους προγραμματιστές. Τον Ιανουάριο του 2017, η Google εξαγόρασε τα Fabric και Crashlytics από το Twitter, για να ενσωματώσει αυτές τις υπηρεσίες στο Firebase Api. Το Firebase ξεκίνησε το Cloud Firestore μια βάση δεδομένων εγγράφων, τον Οκτώβριο του 2017.

4.6.3 Υπηρεσίες Firebase

4.6.3.1 Firebase Analytics η αλλιώς Google Analytics for Firebase



Εικόνα 23 - Firebase Analytics - <https://firebase.google.com/docs/analytics/>

4.6.3.1.1 Τι είναι

Στην καρδιά του Firebase βρίσκεται το Google Analytics for Firebase, μια δωρεάν και απεριόριστη λύση ανάλυσης. Το Analytics ενσωματώνει τις λειτουργίες του Firebase και σας παρέχει απεριόριστες αναφορές μέχρι και 500 ξεχωριστά συμβάντα που μπορείτε να ορίσετε χρησιμοποιώντας το Firebase SDK. Οι αναφορές του Analytics σας βοηθούν να καταλάβετε με σαφήνεια τον τρόπο συμπεριφοράς των χρηστών σας, που σας δίνει τη δυνατότητα να λαμβάνετε ενημερωμένες αποφάσεις σχετικά με το μάρκετινγκ εφαρμογών και τις βελτιστοποιήσεις απόδοσης.

4.6.3.1.2 Πως δουλεύει

Το Google Analytics for Firebase σας βοηθά να κατανοήσετε τον τρόπο με τον οποίο οι χρήστες χρησιμοποιούν την εφαρμογή σας iOS ή Android. Το SDK συλλαμβάνει αυτόματα μια σειρά από συμβάντα και ιδιότητες χρηστών και επίσης σας επιτρέπει να ορίσετε τα δικά σας προσαρμοσμένα συμβάντα για να μετρήσετε τα πράγματα που έχουν μοναδική σημασία για την επιχείρησή σας. Μόλις ληφθούν τα δεδομένα, είναι διαθέσιμα σε ένα ταμπλό μέσω της κονσόλας Firebase. Αυτός ο πίνακας ελέγχου παρέχει λεπτομερείς πληροφορίες σχετικά με τα δεδομένα σας - από συνοπτικά δεδομένα, όπως

ενεργούς χρήστες και δημογραφικά στοιχεία, έως πιο λεπτομερή στοιχεία, όπως τον εντοπισμό των πιο αγορασμένων προϊόντων και υπηρεσιών σας.

4.6.3.2 Firebase Cloud Messaging



Εικόνα 24 - Firebase Cloud Messaging - <https://firebase.google.com/docs/cloud-messaging/>

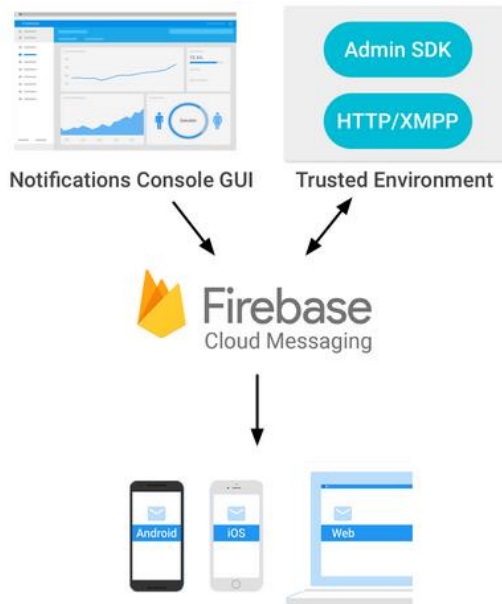
4.6.3.2.1 Τι είναι

Παλαιότερα γνωστό ως Google Cloud Messaging (GCM), το Firebase Cloud Messaging (FCM) είναι μια λύση μεταξύ των πλατφορμών για μηνύματα και ειδοποιήσεις για εφαρμογές Android, iOS και web, οι οποίες από το 2016 μπορούν να χρησιμοποιηθούν χωρίς κόστος.

4.6.3.2.2 Πως δουλεύει

Μια εφαρμογή που χρησιμοποιεί FCM περιλαμβάνει δύο κύρια στοιχεία για την αποστολή και λήψη:

1) Ένα αξιόπιστο περιβάλλον, όπως Cloud Functions για Firebase ή έναν Server εφαρμογών, στα οποία μπορείτε να δημιουργήσετε, να στοχεύσετε και να στείλετε μηνύματα. Μια εφαρμογή iOS, Android ή Web (JavaScript) που λαμβάνει μηνύματα.



Εικόνα 25 - Firebase Cloud Messaging - <https://firebase.google.com/docs/cloud-messaging/>

2) Μπορείτε να στείλετε μηνύματα μέσω του διαχειριστή SDK ή των API HTTP και XMPP. Για τη δοκιμή ή την αποστολή μηνυμάτων μάρκετινγκ με ισχυρές ενσωματωμένες στοχευμένες και αναλυτικές υπηρεσίες, μπορείτε επίσης να χρησιμοποιήσετε τον Notification Composer.

4.6.3.3 Firebase Authentication



Εικόνα 26 - Firebase Authentication - <https://firebase.google.com/docs/auth/>

4.6.3.3.1 Τι είναι

Οι περισσότερες εφαρμογές πρέπει να γνωρίζουν την ταυτότητα ενός χρήστη. Η γνώση της ταυτότητας ενός χρήστη, επιτρέπει σε μια εφαρμογή να αποθηκεύει με ασφάλεια τα δεδομένα χρήστη στο Cloud και να παρέχει την ίδια εξατομικευμένη εμπειρία σε όλες τις συσκευές του χρήστη. Η υπηρεσία Authentication Firebase παρέχει υπηρεσίες back-end, εύχρηστα SDK και έτοιμες βιβλιοθήκες UI, για τον έλεγχο της ταυτότητας των χρηστών στην εφαρμογή σας. Υποστηρίζει έλεγχο ταυτότητας, χρησιμοποιώντας κωδικούς πρόσβασης, αριθμούς τηλεφώνου, δημοφιλείς παρόχους ομοσπονδιακής ταυτότητας όπως το Google, Facebook και Twitter και πολλά άλλα. Ο έλεγχος ταυτότητας Firebase, ενσωματώνεται με άλλες υπηρεσίες Firebase και αξιοποιεί τα πρότυπα της βιομηχανίας όπως το OAuth 2.0 και το OpenID Connect, ώστε να μπορεί εύκολα να ενσωματωθεί με το δικό σας backend.

4.6.3.3.2 Πως δουλεύει

Για να εγγράψετε έναν χρήστη στην εφαρμογή σας, λαμβάνετε πρώτα τα διαπιστευτήρια ελέγχου ταυτότητας από το χρήστη. Αυτά τα διαπιστευτήρια μπορούν να είναι η διεύθυνση ηλεκτρονικού ταχυδρομείου και ο κωδικός πρόσβασης του χρήστη ή ένα διακριτικό OAuth από έναν πάροχο ταυτότητας. Στη συνέχεια, αυτά, μεταβιβάζονται τα διαπιστευτήρια στο SDK Authentication Firebase. Οι υπηρεσίες backend θα επαληθεύσουν τα διαπιστευτήρια αυτά και θα επιστρέψουν μια απάντηση στον πελάτη.



Εικόνα 27 - Firebase Authentication - <https://firebase.google.com/products/auth/>

Μετά την επιτυχή είσοδο, μπορείτε να έχετε πρόσβαση στις βασικές πληροφορίες του προφίλ του χρήστη και μπορείτε να ελέγχετε την πρόσβαση του χρήστη σε δεδομένα που είναι αποθηκευμένα σε άλλα προϊόντα της Firebase. Μπορείτε επίσης, να

χρησιμοποιήσετε το παρεχόμενο διακριτικό ελέγχου ταυτότητας για να επαληθεύσετε την ταυτότητα των χρηστών στις δικές σας υπηρεσίες back-end.

4.6.3.4 Firebase Realtime Database



Εικόνα 28 - Firebase Realtime Database -

<https://firebase.google.com/docs/database/>

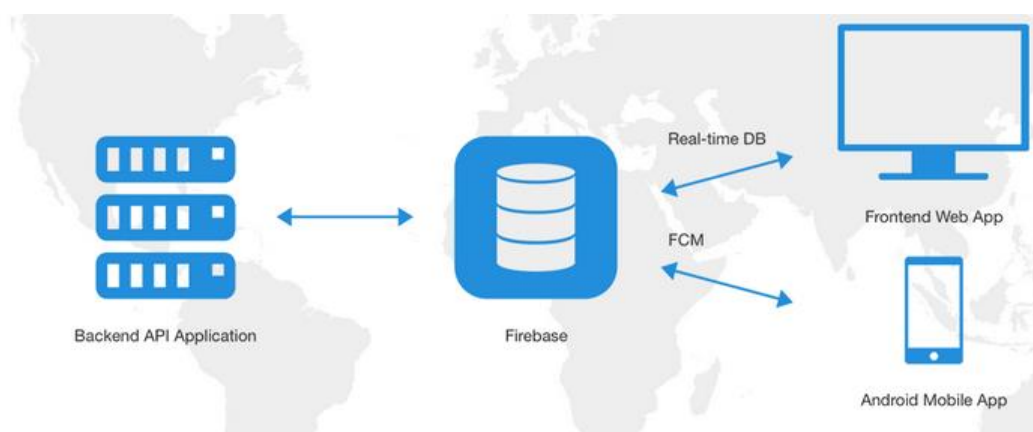
4.6.3.4.1 Τι είναι

Η βάση δεδομένων Firebase Realtime είναι μια βάση δεδομένων που φιλοξενείται σε Cloud. Τα δεδομένα, αποθηκεύονται ως JSON και συγχρονίζονται σε πραγματικό χρόνο σε κάθε συνδεδεμένο πελάτη. Όταν δημιουργείτε εφαρμογές πολλαπλών πλατφορμών με τα SDK iOS, Android και JavaScript, όλοι οι πελάτες σας μοιράζονται ένα στιγμιότυπο βάσης δεδομένων πραγματικού χρόνου και λαμβάνουν αυτόματα ενημερώσεις με τα νεότερα δεδομένα.

4.6.3.4.2 Πως δουλεύει

Η βάση δεδομένων Firebase Realtime, σας επιτρέπει να δημιουργείτε πλούσιες εφαρμογές συνεργασίας, επιτρέποντας ασφαλή πρόσβαση στην βάση δεδομένων απευθείας από τον κωδικό του user. Τα δεδομένα, διατηρούνται τοπικά και ακόμα και όταν είναι εκτός σύνδεσης, τα γεγονότα σε πραγματικό χρόνο συνεχίζουν να υπάρχουν, δίνοντας στον τελικό χρήστη μια εμπειρία απόκρισης. Όταν η συσκευή ανακτήσει τη σύνδεση, η βάση δεδομένων Realtime συγχρονίζει τις τοπικές αλλαγές δεδομένων με τις

απομακρυσμένες ενημερώσεις που συνέβησαν ενώ ο client ήταν εκτός σύνδεσης, συγχωνεύοντας οποιεσδήποτε συγκρούσεις αυτόματα. Η βάση δεδομένων Realtime, παρέχει μια ευέλικτη γλώσσα κανόνων βασισμένη στην έκφραση, που ονομάζεται Κανόνες Ασφαλείας Βάσεων Δεδομένων Firebase Realtime, για να καθορίσετε την δομή των δεδομένων σας και πότε μπορούν να διαβαστούν ή να γραφτούν δεδομένα. Όταν ενσωματωθεί με τον έλεγχο ταυτότητας Firebase, οι προγραμματιστές μπορούν να καθορίσουν ποιος έχει πρόσβαση σε ποια δεδομένα και πώς μπορούν να έχουν πρόσβαση σε αυτά. Η βάση δεδομένων Realtime είναι μια βάση δεδομένων NoSQL και ως εκ τούτου έχει διαφορετικές βελτιστοποιήσεις και λειτουργικότητα σε σχέση με μια σχεσιακή βάση δεδομένων.



Εικόνα 29 - Firebase Realtime Database -

<https://firebase.google.com/products/realtime-database/>

Το API βάσης δεδομένων Realtime, έχει σχεδιαστεί ώστε να επιτρέπει μόνο εργασίες που μπορούν να εκτελεστούν γρήγορα. Αυτό, σας δίνει τη δυνατότητα να δημιουργήσετε μια μεγάλη εμπειρία σε πραγματικό χρόνο, που μπορεί να εξυπηρετήσει εκατομμύρια χρήστες χωρίς συμβιβασμούς στην απόκριση. Εξαιτίας αυτού, είναι σημαντικό να σκεφτείτε πώς οι χρήστες πρέπει να έχουν πρόσβαση στα δεδομένα σας και στη συνέχεια να τα διαμορφώνουν ανάλογα.

4.6.3.5 Firebase Cloud Storage



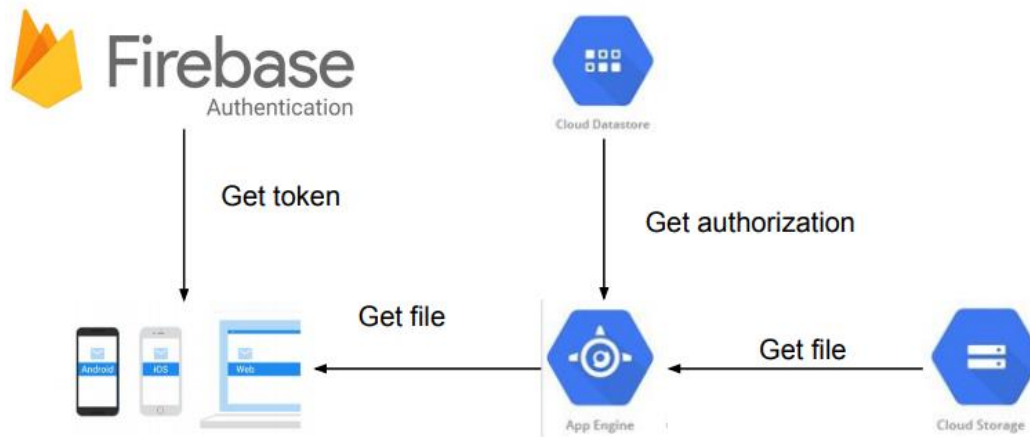
Εικόνα 30 - Firebase Cloud Storage - <https://firebase.google.com/docs/storage/>

4.6.3.5.1 Τι είναι

Το Cloud Storage for Firebase είναι μια ισχυρή, απλή και οικονομικά αποδοτική υπηρεσία αποθήκευσης αντικειμένων, που έχει σχεδιαστεί για την Google. Τα SDK Firebase για Cloud Storage προσθέτουν Google Security για μεταφορτώσεις και λήψεις αρχείων για τις εφαρμογές Firebase σας, ανεξάρτητα από την ποιότητα του δικτύου. Μπορείτε να χρησιμοποιήσετε τα SDK για την αποθήκευση εικόνων, ήχου, βίντεο ή άλλου περιεχομένου που δημιουργεί ο χρήστης. Στον Server, μπορείτε να χρησιμοποιήσετε το Google Cloud Storage, για να έχετε πρόσβαση στα ίδια αρχεία.

4.6.3.5.2 Πως δουλεύει

Οι προγραμματιστές, χρησιμοποιούν το SDK του Firebase για Cloud Storage για να φορτώσουν και να κατεβάσουν αρχεία απευθείας από πελάτες. Εάν η σύνδεση δικτύου είναι κακή, ο πελάτης είναι σε θέση να ξαναδοκιμάσει τη λειτουργία εκεί όπου διακόπηκε, εξοικονομώντας έτσι στους χρήστες χρόνο και bandwidth. Το Cloud Storage, αποθηκεύει τα αρχεία σας σε ένα bucket αποθήκευσης του Google Cloud, καθιστώντας τα προσβάσιμα μέσω του Firebase και του Google Cloud. Αυτό σας επιτρέπει την ευελιξία να ανεβάζετε και να κατεβάζετε αρχεία από πελάτες κινητής τηλεφωνίας, μέσω των SDK Firebase και να επεξεργάζεστε την πλευρά του διακομιστή, όπως το φιλτράρισμα εικόνων ή την κρυπτογράφηση βίντεο χρησιμοποιώντας την πλατφόρμα Google Cloud Platform. Έτσι λοιπόν, δεν υπάρχει καμία ανάγκη να μεταναστεύσουν σε οποιοδήποτε άλλο πάροχο.



Εικόνα 31 - Firebase Cloud Storage - <https://geovation.github.io/downloading-protected-files-in-google-cloud-storage>

Το SDK Firebase για Cloud Storage ενσωματώνεται άψογα με την πιστοποίηση Firebase Authentication για τον προσδιορισμό των χρηστών και παρέχει μια δηλωτική γλώσσα ασφαλείας, που σας επιτρέπει να ορίζετε στοιχεία ελέγχου πρόσβασης σε μεμονωμένα αρχεία ή ομάδες αρχείων, ώστε να μπορείτε να δημιουργείτε αρχεία ως δημόσια ή ιδιωτικά όπως θέλετε.

4.6.3.6 Firebase Hosting



Εικόνα 32 - Firebase Hosting - <https://medium.com/evenbit/gateway-to-firebase-hosting-dda81e83fca>

4.6.3.6.1 Τι είναι

Το Firebase Hosting είναι φιλοξενία ιστοσελίδων για την παραγωγή από προγραμματιστές. Με μία μόνο εντολή, μπορείτε να αναπτύξετε γρήγορα εφαρμογές ιστού και να εκτελείτε τόσο στατικό όσο και δυναμικό περιεχόμενο σε ένα παγκόσμιο δίκτυο παροχής περιεχομένου CDN(content delivery network). Μπορείτε επίσης, να αντιστοιχίσετε το Firebase Hosting με Cloud Functions, για να δημιουργήσετε και να φιλοξενήσετε μικρές επιχειρήσεις στη Firebase.

4.6.3.6.2 Πως δουλεύει

Το Firebase Hosting είναι χτισμένο για το σύγχρονο web developer. Οι ιστότοποι και οι εφαρμογές, είναι πιο ισχυροί από ποτέ με την άνοδο των Javascript Frameworks front-end, όπως τα React και Angular και στατικά εργαλεία όπως το Jekyll. Είτε αναπτύσσεται μια απλή σελίδα καλωσορίσματος του χρήστη, είτε μια σύνθετη και πολύπλοκη εφαρμογή, το Hosting παρέχει την υποδομή, τις λειτουργίες και τα εργαλεία που είναι προσαρμοσμένα στην ανάπτυξη και τη διαχείριση ιστοτόπων και εφαρμογών. Το Hosting, δίνει το έργο σας στον τομέα firebaseapp.com. Χρησιμοποιώντας το Firebase CLI, μπορείτε να αναπτύξετε αρχεία από τοπικούς καταλόγους στον υπολογιστή σας στον browser φιλοξενίας σας. Πέρα από την εξυπηρέτηση του στατικού περιεχομένου, μπορείτε να χρησιμοποιήσετε τις λειτουργίες Cloud για Firebase για να προβάλετε δυναμικό περιεχόμενο και να φιλοξενήσετε μικρές επιχειρήσεις στον ιστότοπό σας. Όλο το περιεχόμενο προβάλλεται μέσω σύνδεσης SSL από τον κοντινότερο edge Server στο παγκόσμιο CDN. Εκτός από τη φιλοξενία περιεχομένου, το Firebase Hosting προσφέρει ελαφρές επιλογές διαμόρφωσης φιλοξενίας για να δημιουργήσετε εξελιγμένα PWA(Progressive Web Apps). Μπορείτε εύκολα να ξαναγράψετε διευθύνσεις URL για δρομολόγηση από πλευράς πελάτη ή να δημιουργήσετε προσαρμοσμένες κεφαλίδες. Μόλις είστε έτοιμοι να πάρετε έναν ιστότοπο στην παραγωγή, μπορείτε να συνδέσετε το δικό σας όνομα τομέα στο Firebase Hosting. Προσφέρεται αυτόματα ένα πιστοποιητικό SSL για τον τομέα σας, έτσι ώστε όλο το περιεχόμενό σας να εμφανίζεται με ασφάλεια.

4.6.3.7 Firebase Cloud Functions



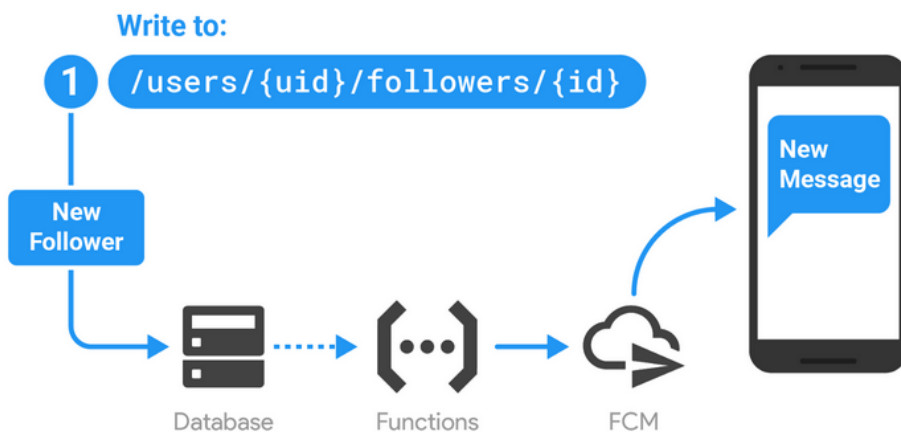
Εικόνα 33 - Firebase Cloud Functions - <https://medium.com/oceanize-geeks/android-firebase-clould-function-take-control-of-your-firebase-backend-a9a996e8ad51>

4.6.3.7.1 Τι είναι

Οι Cloud Functions του Firebase, επιτρέπουν στους προγραμματιστές να εκτελούν αυτόματα τον κώδικα backend ως απόκριση στα γεγονότα που ενεργοποιούνται από τις λειτουργίες Firebase και τις αιτήσεις HTTPS. Ο κώδικας τους αποθηκεύεται στο σύννεφο της Google και εκτελείται σε ένα διαχειριζόμενο περιβάλλον. Έτσι, δεν χρειάζεται να διαχειριστούν τους δικούς τους Servers.

4.6.3.7.2 Πως δουλεύει

Αφού γραφτεί και αναπτυχθεί μια μέθοδος, οι servers της Google αρχίζουν να διαχειρίζονται τη μέθοδο αμέσως. Οι προγραμματιστές μπορούν να πυροδοτήσουν τη λειτουργία απευθείας με ένα αίτημα HTTP, ή στην περίπτωση λειτουργιών background, οι servers της Google θα ακούσουν τα συμβάντα και θα εκτελέσουν τη λειτουργία όταν ενεργοποιηθεί η κληθεί από κάποια άλλη μέθοδο.

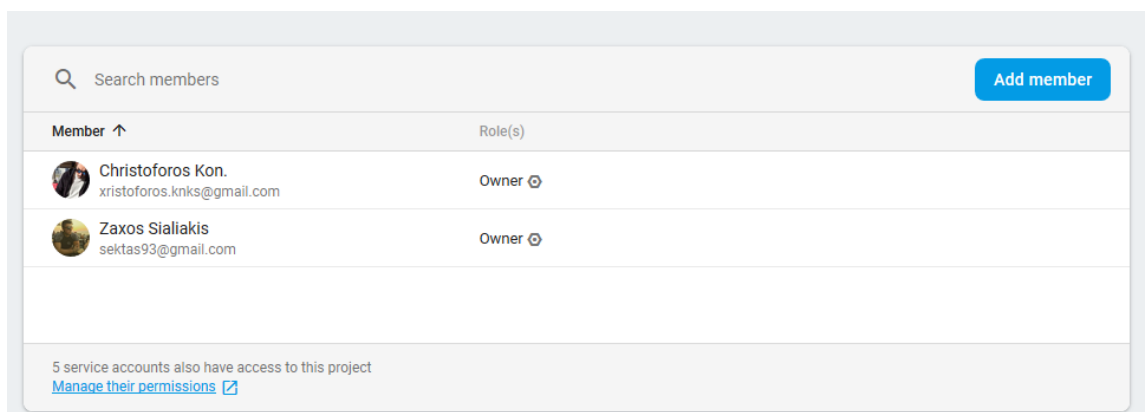


Εικόνα 34 - Firebase Cloud Functions - <https://arpentechologies.com/en/blog/programming-and-web-design/what-are-the-cloud-functions-de-firebase/>

- 1) Ο προγραμματιστής γράφει τον κώδικα για μια νέα λειτουργία, επιλέγοντας έναν πάροχο (όπως η Realtime Database) και ορίζοντας τους όρους υπό τους οποίους θα εκτελεστεί η μέθοδος. Ο προγραμματιστής αναπτύσσει τη μέθοδο και το Firebase τη συνδέει με τον επιλεγμένο πάροχο συμβάντων.
- 2) Όταν ο πάροχος δημιουργεί ένα συμβάν που ταιριάζει με τις συνθήκες της λειτουργίας, ο κώδικας καλείται.
- 3) Εάν η λειτουργία είναι απασχολημένη με το χειρισμό πολλών συμβάντων, η Google δημιουργεί περισσότερες περιπτώσεις για να χειριστεί το έργο πιο γρήγορα. Εάν η λειτουργία είναι αδρανής, τα περιστατικά καθαρίζονται.
- 4) Όταν ο προγραμματιστής ενημερώνει τη μέθοδο αναπτύσσοντας τον ενημερωμένο κώδικα, όλες οι παρουσίες της παλιάς έκδοσης καθαρίζονται και αντικαθίστανται από νέα versions.
- 5) Όταν ο προγραμματιστής διαγράψει τη μέθοδο, όλες οι εμφανίσεις καθαρίζονται και αφαιρείται η σύνδεση μεταξύ της μεθόδου και του παρόχου.

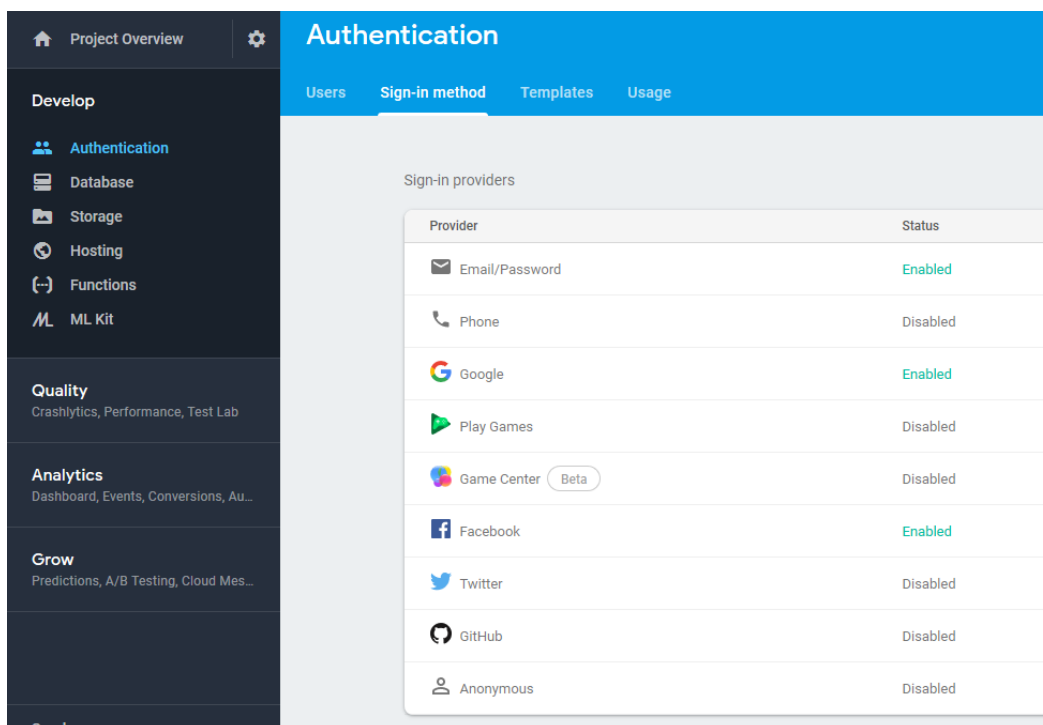
4.6.4 Σύνδεση εφαρμογής με Firebase

Αρχικά, δημιουργήσαμε ένα καινούργιο project στο firebase API και θέσαμε ως owners τους διαχειριστές της εφαρμογής όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 35 - Firebase Console - (δικό μας)

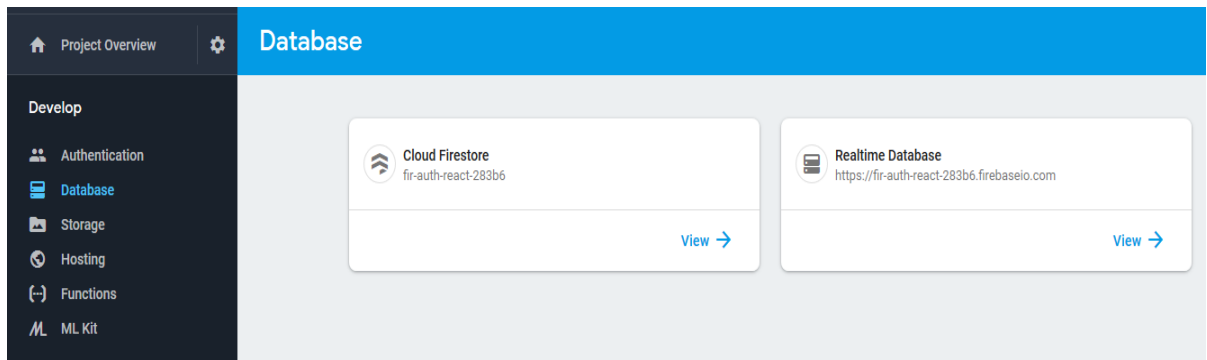
Έπειτα, πήγαμε στην καρτέλα Authentication και ρυθμίσαμε τις επεκτάσεις που θέλουμε να έχουμε στην πλατφόρμα μας, όσων αφορά το sign-in method που μας παρέχει το firebase. Ορίσαμε τις επιλογές Email/Password, Google και Facebook ως enabled καθώς ήταν και αυτά που επιθυμούσαμε να παρέχουμε στους επισκέπτες.



Εικόνα 36 - Firebase Console - (δικό μας)

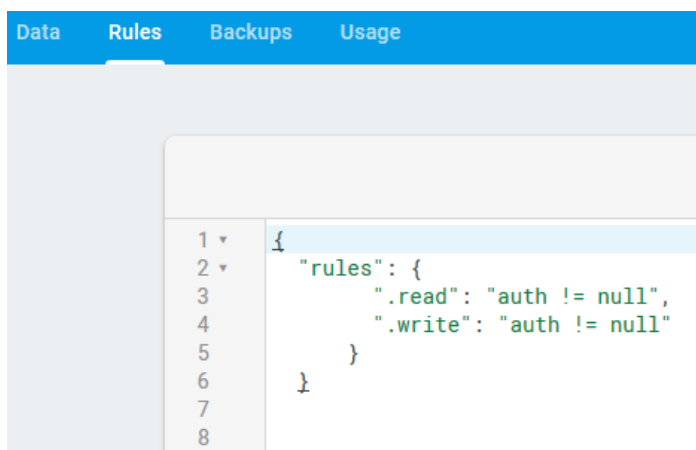
Στη συνέχεια, ορίσαμε δύο Cloud Databases, που μας παρέχει το Firebase ως υπηρεσίες οι οποίες αναφέρθηκαν παραπάνω. Αυτές είναι η Google Firestore, την

οποία ορίσαμε για τους admins της πλατφόρμας BotVoyage και την Firebase Realtime Database για τους χρήστες της πλατφόρμας μας.



Εικόνα 37 - Firebase Console - (δικό μας)

Έπειτα παραμετροποιήσαμε τα rules των βάσεων έτσι ώστε να μπορούμε να ανακτούμε αλλά και να δημιουργούμε περιεχόμενο σε αυτήν μετά την εκτέλεση διαφόρων ενεργειών στην πλατφόρμας π.χ. κατά το sign-up ενός χρήστη αλλά και κατά την πραγματοποίηση ενεργειών των admins στο dashboard.



Εικόνα 38 - Firebase Database Rules - (δικό μας)

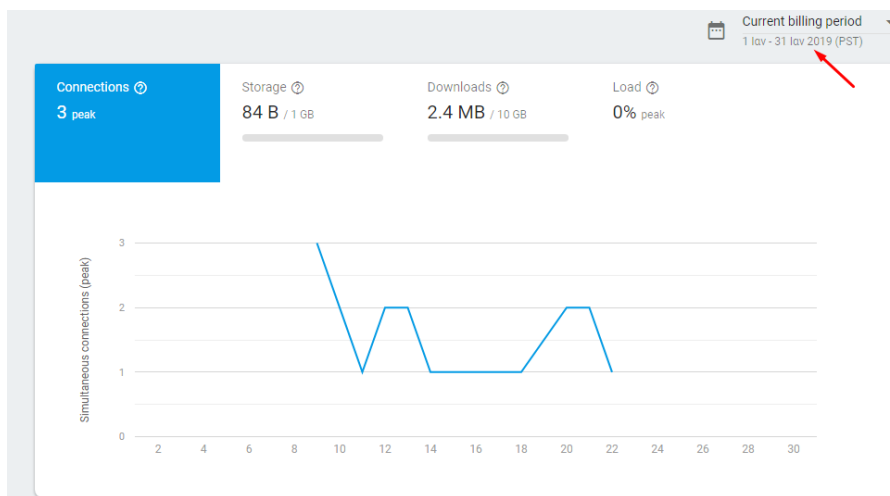
Τέλος, έπρεπε να συνδέσουμε την εφαρμογή μας με το firebase. Αυτό το παρέχει αυτόματα το firebase, αφού εξάγει έτοιμο κώδικα στα settings του project μας. Στο firebase console πήραμε τον κώδικα και τον τοποθετήσαμε στην εφαρμογή μας. Για λόγους ασφάλειας, χρειάστηκε να κρύψουμε τα στοιχεία της βάσης μας στο παρακάτω screenshot.


```
axios.post('/users.json', users)
  .then(response => console.log(response))
  .catch(error => console.log(error));
```

Εικόνα 42 - Firebase Create Users - Notepad++ - (δικό μας)

4.6.4.3 Αναπαράσταση των logins με γραφήματα

Αξίζει επίσης να σημειωθεί, πως το firebase αναπαριστά τις συνδέσεις στην πλατφόρμας μας σε γραφήματα με βάση το πλήθος των εισόδων για συγκεκριμένα χρονικά διαστήματα όπως φαίνεται παρακάτω.



Εικόνα 43 - Firebase Users Logins Graphs - (δικό μας)

4.7 Webpack

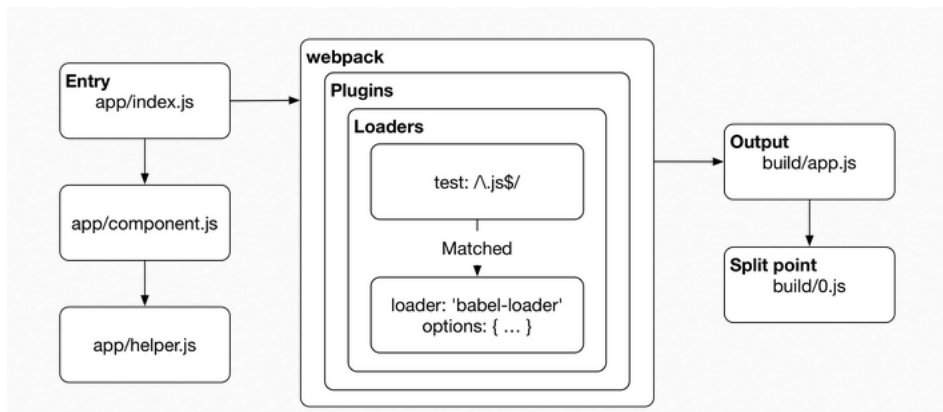
4.7.1 Τι είναι

Το webpack είναι ένα module bundler για εφαρμογές Javascript. Με τον όρο module Bundling, εννοούμε την διαδικασία περισυλλογής πακέτων και στοιχείων και των dependencies τους σε ένα ενιαίο αρχείο με την σωστή σειρά. Το webpack μπορεί να φροντίσει για την μαζική ομαδοποίηση σε έναν ξεχωριστό δρομέα εργασιών. Σήμερα, ολοένα και περισσότερα webpack plugins αναπτύσσονται από την μεγάλη κοινότητα των προγραμματιστών. Μερικές φορές, αυτές οι προσθήκες χρησιμοποιούνται για την εκτέλεση εργασιών οι οποίες γίνονται έξω από το webpack,

παραδείγματος χάρη το καθάρισμα ενός build μιας εφαρμογής. Το React βοήθησε στην διάδοση του webpack και σε άλλα προγραμματιστικά περιβάλλοντα.

4.7.2 Πως δουλεύει

Το webpack βασίζεται σε modules. Όλα τα projects που συνδέονται με το webpack αποτελούνται από μια είσοδο και μια έξοδο. Η διαδικασία ομαδοποίησης ξεκινά από καταχωρήσεις καθορισμένες από το χρήστη. Οι ίδιες οι εγγραφές είναι μονάδες και μπορούν να υποδεικνύουν άλλες ενότητες μέσω των εισαγωγών. Όταν χρησιμοποιούμε webpack σε ένα project τότε αυτό διασχίζει τις εισαγωγές, δηλαδή τα imports του React και τα require της Javascript, κατασκευάζοντας ένα dependency graph του project και παράγει την έξοδο με βάση το configuration. Dependency graph, ονομάζεται ένα κατευθυνόμενο γράφημα το οποίο περιγράφει τον τρόπο με τον οποίο τα αρχεία της εφαρμογής συνδέονται μεταξύ τους. Σε αυτήν την περίπτωση το γράφημα ορίζεται από τις αναφορές(import, require), που γίνονται μεταξύ των αρχείων.



Εικόνα 44 - Webpack Execution Process - <https://survivejs.com/webpack/what-is-webpack/>

Το webpack ξεκινά την δουλειά του από καταχωρήσεις-entries. Κατά την διάρκεια αυτής της διαδικασίας, το webpack αξιολογεί τις αντιστοιχίες εισαγωγής με τις

διαμορφώσεις που είναι προκαθορισμένες από τους φορτωτές-loader, όπως φαίνεται στην εικόνα, οι οποίοι λένε στο webpack πως να μετατρέψει το κάθε πέρασμα.

4.7.3 Διαδικασία Επίλυσης

Μια καταχώρηση είναι από μόνη της ένα module. Όταν το webpack εντοπίσει μια, τότε προσπαθεί να ταιριάξει την καταχώρηση χρησιμοποιώντας το resolve configuration της καταχώρησης αυτής. Μπορούμε να τροποποιήσουμε το webpack, έτσι ώστε να ψάξει μέσα σε συγκεκριμένα directories και συγκεκριμένα στον φάκελο node_modules στον οποίο στεγάζονται τα πακέτα npm που κάνουμε εγκατάσταση και χρησιμοποιούμε στην εφαρμογή μας. Επίσης, μπορούμε να παραμετροποιήσουμε τον τρόπο με τον οποίο το webpack, πραγματοποιεί το ταίριασμα των καταλήξεων των αρχείων που χρησιμοποιεί η εφαρμογή μας. Αν η διαδικασία της επίλυσης αποτύχει, τότε το webpack δημιουργεί ένα σφάλμα χρόνου εκτέλεσης. Από την άλλη πλευρά αν το webpack χρησιμοποίησε την διαδικασία της επίλυσης με επιτυχία, τότε επεξεργάζεται το αρχείο που έγινε το ταίριασμα με βάση τον φορτωτή-loader. Η ευελιξία του webpack επίσης, μας επιτρέπει ακόμα και να εφαρμόσουμε μια συγκεκριμένη μετατροπή σε ένα αρχείο με βάση την τοποθεσία στην οποία έγινε αυτό import σχετικά με το React στο Project μας.

Το webpack, θα επιλύσει κάθε module που συναντά κατά την κατασκευή του dependency graph που αναφέρθηκε παραπάνω. Αν μια καταχώρηση περιέχει dependencies τότε η διαδικασία θα εκτελείται αναδρομικά σε κάθε dependency μέχρι την επιτυχή μετάβαση σε άλλη καταχώρηση. Το Webpack μπορεί να εκτελέσει αυτή τη διαδικασία έναντι οποιουδήποτε τύπου αρχείου, σε αντίθεση με εξειδικευμένα εργαλεία όπως ο μεταγλωττιστής Babel ή Sass. Το webpack μας δίνει την δυνατότητα να αποκτούμε τον έλεγχο των ενεργειών που πρόκειται να πραγματοποιήσει με βάση τα στοιχεία που εντοπίζει. Για παράδειγμα, μας δίνει την δυνατότητα της απόφασης για ενσωμάτωση στοιχείων ενεργητικού τύπου στην Javascript. Επίσης, μας δίνει την δυνατότητα χρησιμοποίησης τεχνικών css, έτσι ώστε να κάνουμε styling των κώδικα της εφαρμογής μας και να αποφύγουμε ζητήματα τυποποιημένου styling. Τέτοιου είδους ευελιξίες είναι και αυτές που κάνουν το

webpack τόσο πολύτιμο. Ακόμα, το webpack μπορεί να συλλάβει στοιχεία όπως εικόνες ή γραμματοσειρές και να εξάγει ξεχωριστά αρχεία για αυτά.

4.7.4 Διαδικασία αξιολόγησης

Υποθέτοντας ότι το webpack έχει βρει όλους τους φορτωτές-loaders, ξεκινά να τους αξιολογεί από κάτω προς τα πάνω και από δεξιά προς τα αριστερά(styleLoader(cssLoader('./main.css'))), ενώ εκτελεί κάθε module με τον κάθε φορτωτή που έχει οριστεί ανάλογα με την σειρά που έχει. Ως αποτέλεσμα, θα υπάρχει μια έξοδος η οποία θα αντιστοιχείται με το πακέτο το οποίο έχει οριστεί γι' αυτήν την έξοδο. Εάν ολοκληρωθεί η αξιολόγηση όλων των φορτωτών χωρίς κανένα σφάλμα, το webpack θα αντιστοιχίσει την πηγή στο τελευταίο bundle. Τα plugins μας επιτρέπουν να παρακολουθήσουμε συμβάντα χρόνου εκτέλεσης σε διαφορετικά στάδια της διαδικασίας bundling όπως αναφέρθηκε παραπάνω. Παρόλο που οι φορτωτές μπορούν να κάνουν πολλά, δεν παρέχουν αρκετή ισχύ για προηγμένες διεργασίες. Τα plugins μπορούν να παρακολουθήσουν τα συμβάντα χρόνου εκτέλεσης που παρέχονται από το webpack. Ένα καλό παράδειγμα είναι η εξαγωγή ενός bundle που εκτελείται από το MiniCssExtractPlugin, το οποίο όταν χρησιμοποιείται με έναν φορτωτή, εξάγει τα αρχεία CSS από το bundle σε ένα ξεχωριστό αρχείο. Χωρίς αυτό το βήμα, το CSS δεν θα αναγνωρίζεται από την προκύπτουσα JavaScript, καθώς το webpack αντιμετωπίζει όλο τον κώδικα ως JavaScript από προεπιλογή.

4.7.5 Το webpack με βάση το configuration

Το webpack βασίζεται εξ ολοκλήρου στο configuration της εφαρμογής την οποία αναπτύσσουμε. Παράδειγμα αρχείου webpack.config.js της εφαρμογής μας:


```

1  const HtmlWebpackPlugin = require("html-webpack-plugin");
2  const HtmlWebpackPluginConfig = new HtmlWebpackPlugin({
3    template: `_${dirname}/src/index.html`,
4    filename: "index.html",
5    inject: "body"
6  });
7
8  const path = require("path");
9
10 module.exports = {
11   entry: [
12     "./src/index.js"
13   ],
14   output: {
15     path: path.join(__dirname, "dist"),
16     filename: "index_bundle.js"
17   },
18   module: {
19     loaders: [
20       {
21         test: /\.js$/,
22         exclude: /node_modules/,
23         loader: "babel-loader",
24         query: {
25           presets: ["es2015", "react", "stage-1"]
26         }
27       },
28       {
29         test: /\.scss$/,
30         loader: "style-loader!css-loader!sass-loader"
31       },
32       {
33         test: /\.(gif|png|jpe?g|svg)$/i,
34         use: [
35           'file-loader',
36           {
37             loader: 'image-webpack-loader',
38             options: {
39               bypassOnDebug: true, // webpack@1.x
40               disable: true, // webpack@2.x and newer
41             }
42           }
43         ]
44       }
45     ]
46   }
47 }

```

Εικόνα 45 - webpack.config.js - Visual Studio Code - (δικό μας)

Το μοντέλο Webpack Configuration μπορεί να φαίνεται λίγο δύσκολο στην κατανόηση κατά περιόδους, καθώς το αρχείο webpack.config.js μπορεί να εμφανίζεται απομονωμένο από τα υπόλοιπα χαρακτηριστικά της εφαρμογής.

4.7.6 Hot Module Replacement

Εργαλεία όπως τα LiveReload και BrowserSync, χρησιμοποιούνται για να ανανεώνουν αυτόματα τα προγράμματα περιήγησης σε περίπτωση αλλαγών στον κώδικα της εφαρμογής μας. Με το Hot Module Replacement, τα πράγματα προχωράνε ένα βήμα παραπέρα. Στην περίπτωση του React, επιτρέπεται στην εφαρμογή να διατηρήσει την κατάσταση χωρίς να προβεί σε ανανέωση. Αν και αυτό δεν ακούγεται πολύ χρήσιμο, μπορεί να κάνει την διαφορά στην πράξη.

4.7.7 Διαχωρισμός κώδικα

Εκτός από το HMR που αναφέρθηκε στην προηγούμενη παράγραφο, οι δυνατότητες του webpack είναι εκτεταμένες. Το webpack μας επιτρέπει να χωρίζουμε τον κώδικα

με πολλούς τρόπους. Μπορούμε ακόμη και να φορτώσουμε τον κώδικα δυναμικά χωρίς να εκτελείται η εφαρμογή μας. Ακόμη και μικρές εφαρμογές μπορούν να επωφεληθούν από αυτήν την διάσπαση κώδικα, καθώς επιτρέπει στους χρήστες να παίρνουν κάτι πιο εύχρηστο στα χέρια τους σε γρηγορότερο χρονικό διάστημα. Η απόδοση του χωρισμού κώδικα είναι ένα ισχυρό χαρακτηριστικό του webpack.

4.7.8 Συμπεράσματα

Το webpack είναι ένα εργαλείο το οποίο αξίζει τον χρόνο εκμάθησης, δεδομένου του χρόνου και της προσπάθειας που μπορεί να εξοικονομήσει μακροπρόθεσμα. Το webpack δεν θα λύσει όλα τα προβλήματα που προκύπτουν στον κώδικα της εφαρμογής μας, μπορεί να λύσει όμως το θέμα της ομαδοποίησης. Ένα πρόβλημα λιγότερο κατά της διαδικασία της ανάπτυξης της εφαρμογής. Η χρήση του αρχείου `package.json` και του webpack μπορεί να οδηγήσει την εφαρμογή μας σε άλλο επίπεδο.

4.7.9 Σύνοψη

- Το webpack είναι ένα εργαλείο που χρησιμοποιείται για την διαδικασία `module bundling` που αναφέρθηκε αρχικά.
- Το webpack βασίζεται σε ένα `dependency graph`.
- Το webpack βασίζεται επίσης σε φορτωτές και `plugins`.
- Το `webpack configuration` περιγράφει τον μετασχηματισμό των στοιχείων του κώδικα της εφαρμογής και της εξόδου που θα παράγει η εφαρμογή μας.
- Το `Hot Module Replacement` του webpack, μπορεί να βελτιώσει την εμπειρία ανάπτυξης της εφαρμογής με την ενημέρωση κώδικα στο πρόγραμμα περιήγησης που χρησιμοποιούν οι χρήστες, χωρίς να χρειάζεται η ανανέωση πλήρους σελίδας.

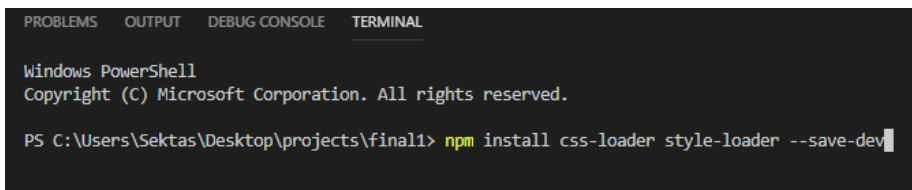
4.7.10 Styling

Το webpack δεν χειρίζεται το styling των αρχείων του κώδικα της εφαρμογής μας, ως εκ τούτου χρησιμοποιήσαμε φορτωτές για να επιτρέψουμε την προσθήκη styling στα αρχεία μας. Οπότε ρυθμίσαμε το css styling με το project μας, έτσι ώστε να λειτουργεί με την αυτόματη ανανέωση του προγράμματος περιήγησης που χρησιμοποιούμε για την αναπαράσταση της εφαρμογής μας.

4.7.11 Φόρτωση css

Για την φόρτωση css χρησιμοποιούμε την κατάληξη scss στα αρχεία του κώδικά μας και αφού το περάσαμε σαν ιδιότητα στο αρχείο webpack.config.js. Χρησιμοποιήσαμε δύο loaders τον css-loader και τον style-loader. Ο πρώτος αναζητά imports και url() στα αντίστοιχα αρχεία και τα αντιμετωπίζει ως κανονική εισαγωγή ES2015 (Next Generation Javascript). Αν ένα import παραπέμπει σε έναν εξωτερικό πόρο, ο css-loader το παρακάμπει καθώς μόνο οι εσωτερικοί πόροι παίρνουν περαιτέρω επεξεργασία από το webpack.

Ο style-loader εισάγει το styling μέσω στοιχείων. Ο τρόπος με τον οποίο γίνεται αυτό είναι προσαρμόσιμος. Επίσης, χρησιμοποιεί το Hot Module Replacement για μια καλύτερη εμπειρία της διαδικασίας ανάπτυξης της εφαρμογής. Τα αντιστοιχισμένα αρχεία μπορούν να υποβληθούν επεξεργασία μέσω των loader όπως file-loader ή url-loader.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Sektas\Desktop\projects\final1> npm install css-loader style-loader --save-dev
```

Εικόνα 46 - Installation of npm loaders - Visual Studio Code - (δικό μας)

Έπειτα, φροντίζουμε έτσι ώστε το webpack να επιτρέπει την φόρτωση scss αρχείων και μόνο για το styling της εφαρμογής μας καθώς και να αναγνωρίζει τους loaders που έγιναν εγκατάσταση στην εικόνα παραπάνω. Αυτές οι παραμετροποιήσεις έγιναν

στο αρχείο webpack.config.js. Επίσης, έχουμε ορίσει την ανάγνωση και το compiling αρχείων Javascript, με κατάληξη js και όχι jsx.

```
module: {
  loaders: [
    {
      test: /\.js$/,
      exclude: /node_modules/,
      loader: "babel-loader",
      query: {
        presets: ["es2015", "react", "stage-1"]
      }
    },
    {
      test: /\.scss$/,
      loader: "style-loader!css-loader!sass-loader"
    },
    {
      test: /\.(gif|png|jpe?g|svg)$/i,
      use: [
        'file-loader',
        {
          loader: 'image-webpack-loader',
          options: {
            bypassOnDebug: true, // webpack@1.x
            disable: true, // webpack@2.x and newer
          }
        }
      ]
    }
  ]
}
```

Εικόνα 47 - webpack.config.js - Visual Studio Code - (δικό μας)

4.7.12 Εγκατάσταση file-loader για φόρτωμα εικόνων

Αφού έχουμε κάνει εγκατάσταση τα npm πακέτα file loader και image-webpack-loader με τις εντολές npm install file-loader --save και npm install image-webpack-loader --save, ανοίξαμε το αρχείο webpack.config.js και θέσαμε στις εντολές use και loader τα αρχεία έτσι ώστε να αναγνωρίζονται και να χρησιμοποιούνται από το webpack για το φόρτωμα των εικόνων. Από προεπιλογή, ο file loader θα επιστρέψει ένα hash τύπου MD5 των περιεχομένων του αρχείου που έχει οριστεί στο test στο webpack.

```
test: /\.(gif|png|jpe?g|svg)$/i,
use: [
  'file-loader',
  {
    loader: 'image-webpack-loader',
    options: {
      bypassOnDebug: true, // webpack@1.x
      disable: true, // webpack@2.x and newer
    }
  }
]
```

Εικόνα 48 - webpack.config.js - Visual Studio Code - (δικό μας)

4.7.13 Εγκατάσταση των webpack plugins και ορισμός της πόρτας

Σε αυτήν την περίπτωση, κάναμε install το πακέτο html-webpack-plugin με την παρακάτω εντολή και σαν dev dependency στο αρχείο package.json.

```
npm install --save-dev html-webpack-plugin
```

Εικόνα 49 - Installation of webpack plugin - Visual Studio Code - (δικό μας)

Το HtmlWebpackPlugin απλοποιεί την δημιουργία αρχείων HTML για την εξυπηρέτηση των webpack bundles. Αυτό είναι ιδιαίτερα χρήσιμο για αυτά τα bundles, τα οποία περιλαμβάνουν ένα hash στο όνομα του αρχείου το οποίο αλλάζει σε κάθε compile. Στην περίπτωσή μας, χρησιμοποιήσαμε το αρχείο index.html, το οποίο περιλαμβάνει στοιχεία όπως ο τίτλος της εφαρμογής μας, την ώρα που αυτή είναι live στο πρόγραμμα περιήγησης.

Αρχείο webpack.config.js

```
const HtmlWebpackplugin = require("html-webpack-plugin");
const HtmlWebpackpluginConfig = new HtmlWebpackplugin({
  template: `${__dirname}/src/index.html`,
  filename: "index.html",
  inject: "body"
});
```

που βλέπει

Εικόνα 50 - webpack.config.js - Visual Studio Code - (δικό μας)

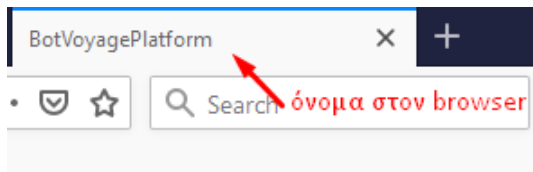
Αρχείο index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   <title>BotVoyagePlatform</title>
8 </head>
9 <body>
10  <div id="root"></div>
11 </body>
12 </html>
13
```

Τίτλος εφαρμογής όταν τρέχει στον browser

Εικόνα 51 - index.html - Visual Studio Code - (δικό μας)

Αναπαράσταση στον browser



Εικόνα 52 - BotVoyagePlatform on Mozilla browser - (δικό μας)

Αποτέλεσμα μετά από compile με την εντολή npm start στο VS CODE

```
Project is running at http://localhost:8000/ ← ρυθμισμένη πόρτα στο webpack
webpack output is served from /
Content not from webpack is served from C:\Users\Sektas\Desktop\projects\final1\dist
404s will fallback to /index.html ← αρχείο προσδιορισμένο στο webpack plugin
Hash: 20fa0d8b7908fb05f745
Version: webpack 3.3.0
Time: 76603ms
```

Εικόνα 53 - npm start run command - Visual Studio Code - (δικό μας)

Asset	Size	Chunks	Chunk Names
e4fbf75202872fc47aa11e6ca322280c.png	189 kB	[emitted]	
c109afcd35e66017f072f063a4d0e42a.png	237 bytes	[emitted]	
46c9588f841037bf55262c1f8cb35a6d.jpeg	80.4 kB	[emitted]	
42449323fcc17bf8f2b4b3751935126c.png	5.22 kB	[emitted]	
2725b2c7714cb29d3af4043767c1db0c.png	245 bytes	[emitted]	
788b1fcd0d8a4466a904635124e034f9.png	249 bytes	[emitted]	
de8b700aaa1d2522cc8dea6351000a23.png	2.04 kB	[emitted]	
b4b8e87e0e00f222b6e273c01810b13b.png	9.47 kB	[emitted]	
9868f55be9fd4ecdca719210275132c7.png	235 bytes	[emitted]	
9760db9d2869ed47ce2fce6be8f260e5.png	34.7 kB	[emitted]	
5fdaaf7d2b568e6aa6a0cc342748122c.png	188 kB	[emitted]	
84e4dd20d48716a186ad2593d8db5a8d.png	186 kB	[emitted]	
3e774d0558fc48016e72fbd2a2c69d46.png	168 kB	[emitted]	
49aba049c205d1be567d5b4c132424b8.png	194 kB	[emitted]	

Εικόνα 54 - npm start run command - Visual Studio Code - (δικό μας)

Επίσης, μας εμφανίζει μετά το compile τα αρχεία τα οποία χρησιμοποιεί η εφαρμογή για την αναπαράσταση στον browser τα οποία έχουν οριστεί από το αρχείο index.js στο webpack, όπως η αναπαράσταση του high order component του react(UserDashboard.js), το οποίο έχει δηλωθεί στο αρχείο index.js στο DOM του react, καθώς και όλων των npm πακέτων και των webpack bundles.

Στο τέλος του compile, εμφανίζεται το μήνυμα webpack:

Compiled successfully σε περίπτωση επιτυχίας.

```
index.html 358 bytes [emitted]
[0] ./node_modules/react/index.js 190 bytes {0} [built]
[50] ./node_modules/react-dom/index.js 1.36 kB {0} [built]
[64] ./node_modules/react-router-dom/es/index.js 1.01 kB {0} [built]
[452] ./node_modules/url/url.js 23.3 kB {0} [built]
[896] multi (webpack)-dev-server/client?http://localhost:8000 ./src/index.js 40 bytes {0} [built]
[897] (webpack)-dev-server/client?http://localhost:8000 5.59 kB {0} [built]
[902] ./node_modules/strip-ansi/index.js 161 bytes {0} [built]
[904] (webpack)-dev-server/client/socket.js 856 bytes {0} [built]
[936] (webpack)-dev-server/client/overlay.js 3.6 kB {0} [built]
[941] (webpack)/hot/emitter.js 77 bytes {0} [built]
[942] ./src/index.js 706 bytes {0} [built]
[951] ./src/index.scss 1.13 kB {0} [built]
[957] ./src/component/main/Main.js 2.47 kB {0} [built]
[970] ./node_modules/react-router-dom/es/Prompt.js 131 bytes {0} [built]
[982] ./src/component/App/UserDashboard.js 4.91 kB {0} [built]
+ 2270 hidden modules
Child html-webpack-plugin for "index.html":
[0] ./node_modules/html-webpack-plugin/lib/loader.js!./src/index.html 676 bytes {0} [built]
[1] ./node_modules/lodash/lodash.js 540 kB {0} [built]
[2] (webpack)/buildin/global.js 509 bytes {0} [built]
[3] (webpack)/buildin/module.js 517 bytes {0} [built]
webpack: Compiled successfully.
```

Εικόνα 55 - npm start run command - Visual Studio Code - (δικό μας)

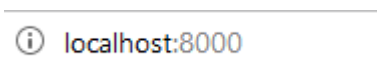
Αντιθέτως σε περίπτωση αποτυχίας, θα μας εμφανίζει που βρίσκει error και το μήνυμα webpack: Failed to compile

Όσον αφορά την ρύθμιση της πόρτας στην οποία θα τρέχει η εφαρμογή μας στον browser, δηλώνεται στο αρχείο webpack.config.js όπως παρακάτω.

```
devServer: {
  port: 8000,
  contentBase: path.join(__dirname, "dist"),
  historyApiFallback: true
},
```

Εικόνα 56 - webpack.config.js - Visual Studio Code - (δικό μας)

Το αποτέλεσμα είναι η αναπαράσταση στον browser



Εικόνα 57 - localhost and port on browser - (δικό μας)

Καθώς και μια ενημέρωση κατά την διάρκεια του compile με την εντολή npm start.

```
Project is running at http://localhost:8000/
```

Εικόνα 58 - npm start run command - Visual Studio Code - (δικό μας)

4.8 Babel

4.8.1 Τι είναι

Η Babel είναι ένας Javascript compiler. Πρόκειται για μια εργαλειομηχανή η οποία χρησιμοποιείται για την μετατροπή ECMAScript 2015+ κώδικα, δηλαδή Next Generation Javascript κώδικα σε μια συμβατή με το παρελθόν έκδοση της Javascript σε τρέχοντα και παλαιότερα προγράμματα περιήγησης. Κάποια από τα κύρια χαρακτηριστικά της είναι:

- Μετατροπή Σύνταξης
- Παροχή χαρακτηριστικών Polyfill που λείπουν από το περιβάλλον ανάπτυξης της εφαρμογής μας (npm πακέτο @babel/polyfill)
- Μετασχηματισμοί πηγαίου κώδικα (codemods)

4.8.2 ES2015 και μετέπειτα

Η Babel κατέχει υποστήριξη για την τελευταία έκδοση της Javascript μέσω διαφόρων μετασχηματιστών σύνταξης. Αυτά τα plugins, μας επιτρέπουν να χρησιμοποιήσουμε νέα σύνταξη χωρίς να περιμένουμε κάποια υποστήριξη από κάποιο πρόγραμμα περιήγησης

4.8.3 JSX και React

Η JSX είναι μια επέκταση σύνταξης που μοιάζει με XML σε ECMAScript χωρίς κάποια καθορισμένη σημασιολογία. Δεν προορίζεται για εφαρμογή από μηχανές ή προγράμματα περιήγησης. Αντιθέτως, προορίζεται για εφαρμογή από διάφορους προεπεξεργαστές (transpilers) για την μετατροπή αντικειμένων σύνταξης σε ECMAScript.

Παράδειγμα χρήσης εφαρμογής JSX, για την αναπαράσταση components γραφικών:

```
// Using JSX to express UI components.
var dropdown =
  <Dropdown>
    A dropdown list
    <Menu>
      <MenuItem>Do Something</MenuItem>
      <MenuItem>Do Something Fun!</MenuItem>
      <MenuItem>Do Something Else</MenuItem>
    </Menu>
  </Dropdown>;

render(dropdown);
```

Εικόνα 59 - Παράδειγμα χρήσης εφαρμογής JSX - <https://facebook.github.io/jsx/>

4.8.4 Λογική

Ο σκοπός αυτής της προδιαγραφής είναι να ορίσει μια σύντομη και οικεία σύνταξη για τον ορισμό δομών δέντρου με χαρακτηριστικά. Μια γενική αλλά σαφώς καθορισμένη σύνταξη, επιτρέπει σε μια κοινότητα ανεξάρτητων συσκευών ανάγνωσης και σύνταξης εγγράφων να συμμορφώνονται με μια ενιαία προδιαγραφή. Η ενσωμάτωση μιας νέας σύνταξης σε μια υπάρχουσα γλώσσα είναι μια επικίνδυνη ενέργεια. Άλλοι εκτελεστές σύνταξης ή η υπάρχουσα γλώσσα, ενδέχεται να εισάγουν άλλη μη συμβατή επέκταση σύνταξης. Μέσα από μια αυτόνομη προδιαγραφή, διευκολύνουμε τους υλοποιητές άλλων επεκτάσεων σύνταξης να εξετάσουν το JSX κατά το σχεδιασμό της δικής τους σύνταξης. Αυτή η προδιαγραφή δεν επιχειρεί να συμμορφωθεί με προδιαγραφές XML ή HTML. Το JSX έχει σχεδιαστεί ως χαρακτηριστικό ECMAScript και η ομοιότητα με την XML είναι μόνο για εξοικείωση.

4.8.5 React JSX

Βασικά η JSX παρέχει μόνο προσθήκες σύνταξης για την μέθοδο `React.createElement(component, props, ...children)`. Ο κώδικας JSX:

```
<MyButton color="blue" shadowSize={2}>
  Click Me
</MyButton>
```

Εικόνα 60 - Παράδειγμα React - <https://reactjs.org/docs/jsx-in-depth.html>

Γίνεται compile σε:

```
React.createElement(
  MyButton,
  {color: 'blue', shadowSize: 2},
  'Click Me'
)
```

Εικόνα 61 - Παράδειγμα React - <https://reactjs.org/docs/jsx-in-depth.html>

Μπορούμε επίσης να χρησιμοποιήσουμε self-closing ετικέτα αν δεν υπάρχουν παιδιά (children component).

```
<div className="sidebar" />
```

Εικόνα 62 - Παράδειγμα React - <https://reactjs.org/docs/jsx-in-depth.html>

Το οποίο γίνεται compile σε

```
React.createElement(
  'div',
  {className: 'sidebar'},
  null
)
```

Εικόνα 63 - Παράδειγμα React - <https://reactjs.org/docs/jsx-in-depth.html>

4.8.6 Καθορισμός του τύπου React Element

Το πρώτο μέρος μιας ετικέτας JSX καθορίζει τον τύπο του στοιχείου React element. Οι κεφαλαίοι τύποι υποδεικνύουν ότι η ετικέτα JSX αναφέρεται σε ένα στοιχείο React. Αυτές οι ετικέτες, μεταγλωττίζονται σε μια άμεση αναφορά στην ονομαζόμενη μεταβλητή, οπότε αν χρησιμοποιήσουμε την έκφραση JSX `<Foo/>`, το οποίο πρόκειται για ένα react component, αυτό πρέπει να είναι στο πεδίο εφαρμογής. Από

τη στιγμή που η JSX μεταγλωττίζεται σε κλήσεις προς το `React.createElement`, η βιβλιοθήκη React πρέπει επίσης να είναι πάντοτε σε πεδίο εφαρμογής από τον κώδικα JSX. Για παράδειγμα, και οι δύο εισαγωγές (imports) είναι απαραίτητες σε αυτόν τον κώδικα, παρόλο που το React και το `CustomButton` δεν αναφέρονται απευθείας από την JavaScript:

```
import React from 'react';
import CustomButton from './CustomButton';

function WarningButton() {
  // return React.createElement(CustomButton, {color: 'red'}, null);
  return <CustomButton color="red" />;
}
```

Εικόνα 64 - Παράδειγμα React - <https://reactjs.org/docs/jsx-in-depth.html>

Εάν δεν χρησιμοποιήσουμε έναν JavaScript bundler και φορτώσουμε το React από μια ετικέτα `<script>`, είναι ήδη στο πεδίο εφαρμογής ως `React global`.

4.9 Wit.ai

4.9.1 Τι είναι NLP/NLU

Το NLP (Natural Language Processing) συνιστά έναν τομέα της επιστήμης των υπολογιστών, ο οποίος ασχολείται με την αλληλεπίδραση των υπολογιστών με τις φυσικές (ανθρώπινες) γλώσσες και ειδικότερα με το πως θα προγραμματίσει τους υπολογιστές να επεξεργάζονται αποτελεσματικά μεγάλα ποσά από δεδομένα φυσικής γλώσσας. Με άλλα λόγια, με τη βοήθεια του NLP, άνθρωποι και μηχανές μπορούν πλέον να μιλάνε φυσικά. Επιπρόσθετα, είναι σημαντικό να κατανοηθεί, ότι είναι αναπόσπαστο τμήμα σε οποιοδήποτε εγχείρημα ανάπτυξης μιας τεχνητής νοημοσύνης που να αποσκοπεί στην προσομοίωση του ανθρώπινου είδους.

Ένα NLP σύστημα είναι αποτελεσματικό, όταν μπορεί να συλλέξει οτιδήποτε έχει ειπωθεί προς αυτό, να το τμηματοποιήσει, να κατανοήσει τη σημασία και να

ανταποκριθεί με τη σωστή επιλογή πράξης και φυσικά αυτή η πράξη να συνιστά φυσική και αντιληπτή από τον χρήστη.

Τις περισσότερες φορές προκύπτει παρανόηση μεταξύ της παραπάνω έννοιας και του NLU (Natural Language Understanding). Το τελευταίο, αποτελεί απλώς ένα υποσύνολο του ευρύτερου κόσμου του NLP. Για να γίνει κατανοητή η διαφοροποίηση ας προχωρήσουμε στην επεξήγηση της συγκεκριμένης έννοιας. Η έννοια του NLU, πραγματεύεται την απόσπαση μιας δομημένης πληροφορίας από την αδόμητη μορφή της φυσικής γλώσσας, μέσω δηλαδή των μηνυμάτων σε μία πλατφόρμα chatting.

Στη συνέχεια, ας αναλύσουμε δύο σημαντικές ορολογίες για να γίνει κατανοητή η λειτουργία. Σε πρώτο επίπεδο συναντάται η έννοια intent (πρόθεση, σκοπός), η οποία ουσιαστικά είναι το θέμα για το οποίο μιλά ο χρήστης σε φυσική γλώσσα (π. χ. ‘Θα βρέξει αύριο;’, εννοώντας τον καιρό). Σε δεύτερο επίπεδο, υπάρχει η έννοια της entity(οντότητα), η οποία συνιστά ένα τμήμα πληροφορίας προς ανίχνευση από την είσοδο του χρήστη. Στο παραπάνω παράδειγμα, θα πρέπει η οντότητα να είναι ο καιρός με κάποιες τιμές όπως ηλιόλουστος η βροχερός.

Υπάρχουν διάφορα είδη intent, τα οποία ας αναφέρουμε. Αρχικά, υπάρχουν trait entities (χαρακτηριστικών), μέσω του οποίου ανιχνεύεται μια πληροφορία με βάση το γενικότερο νόημα της πρότασης. Δεύτερον, με βάση κάποιες λέξεις κλειδιά (keywords). Επίσης, υπάρχει συνδυασμός χαρακτηριστικών με λέξεις κλειδιά και τέλος υπάρχουν έτοιμες ανάλογα την υπηρεσία. Για παράδειγμα στο Wit.ai, υπάρχει έτοιμη οντότητα η οποία κατανοεί την τοποθεσία.

Συμπερασματικά, έχουμε την ευρύτερη κατηγορία του NLP, που στοχεύει στη φυσική επικοινωνία μεταξύ μηχανής και ανθρώπου και το NLU, το οποίο όντας υποσύνολο του παραπάνω, αποσκοπεί στη μετατροπή της αδόμητης πληροφορία (φυσική γλώσσα) σε δομημένη. Για αυτόν τον σκοπό, υπάρχουν υπηρεσίες που με τη λογική των intents και entities, τον εξυπηρετούν σε θεμιτό βαθμό.

4.9.2 Τι είναι το Wit.ai

Το Wit.ai είναι μία υπηρεσία NLU, το οποίο απλούστατα κατανοεί τι λένε οι χρήστες στην εφαρμογή σου χρησιμοποιώντας τις κατηγορίες οντοτήτων για κατανόηση του περιεχομένου της φυσικής γλώσσας όπως αναλύθηκε παραπάνω. Για να επέλθει

καλύτερο αποτέλεσμα, απαραίτητη είναι η εκπαίδευση με ένα εύλογο πλήθος διαφορετικών προτάσεων για το training μιας συγκεκριμένης οντότητας. Τέλος, αλλά λιγότερο σημαντικό, μια παράμετρος της υπηρεσίας που πρέπει να παρατηρείται, είναι ο δείκτης αυτοπεποίθησης, ο οποίος κυμαίνεται από το 0 έως το 1 και προειδοποιεί τον διαχειριστή πόσο αποτελεσματικά έχει γίνει η εκπαίδευση.

4.10 Git (λογισμικό)

4.10.1 Τι είναι και που μας χρησιμεύει

Το Git, είναι ένα σύστημα ελέγχου εκδόσεων (λέγεται και σύστημα ελέγχου αναθεωρήσεων ή σύστημα ελέγχου πηγαίου κώδικα) με έμφαση στην ταχύτητα, την ακεραιότητα των δεδομένων και την υποστήριξη για κατανεμημένες μη γραμμικές ροές εργασίας. Το Git σχεδιάστηκε και αναπτύχθηκε αρχικά από τον Λίνους Τόρβαλντς για την ανάπτυξη του πυρήνα Linux το 2005 και έχει γίνει από τότε το πιο διαδεδομένο σύστημα ελέγχου εκδόσεων για ανάπτυξη λογισμικού.

Όπως τα περισσότερα άλλα κατανεμημένα συστήματα ελέγχου εκδόσεων/αναθεωρήσεων και αντίθετα με τα περισσότερα συστήματα client-server, κάθε κατάλογος εργασίας του Git είναι ένα ολοκληρωμένο αποθετήριο λογισμικού με πλήρες ιστορικό και δυνατότητες πλήρους παρακολούθησης της έκδοσης, ανεξάρτητα από την πρόσβαση δικτύου ή ενός κεντρικού server. Το Git ήταν ένας σημαντικός παράγοντας στην άμεση συνεργασία μεταξύ μας, σε προγραμματιστικό επίπεδο. Έχοντας το origin αποθηκευμένο στο ίντερνετ, μπορέσαμε ανά πάσα στιγμή να κάνουμε αλλαγές στον κώδικα, χωρίς να μας απασχολεί κάποια αλλαγή του συναδέλφου που προκύπτει την ίδια στιγμή με κάποια αλλαγή του άλλου.

4.11 Npm

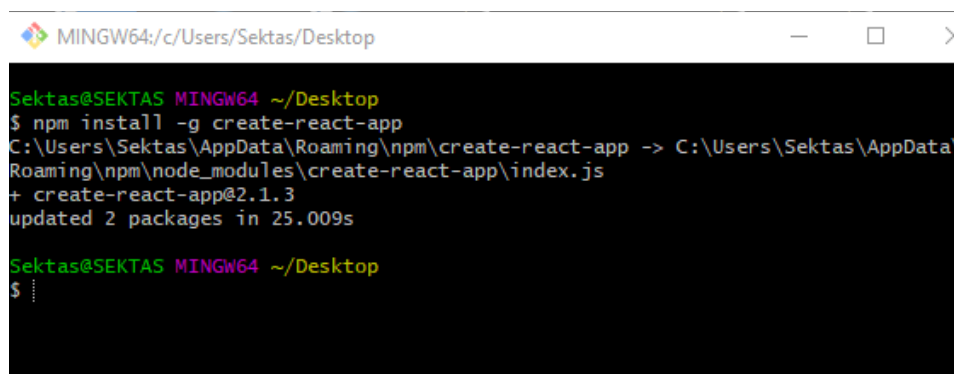
4.11.1 Τι είναι

Αποτελώντας τον διαχειριστή πακέτων για την JavaScript, διαμορφώνει ένα τεράστιο σύνολο από πακέτα τα οποία προσεγγίζουν τα εξακόσιες χιλιάδες με περίπου τρία δισεκατομμύρια downloads ανά εβδομάδα. Χρησιμοποιείται από προγραμματιστές που αναπτύσσουν ανοιχτό λογισμικό για να δανειστούν κάποιο αλλά και να κοινοποιήσουν ένα δικό τους πακέτο, με αποτέλεσμα όλη η κοινότητα και το σύνολο των πακέτων να αυξάνεται.

Κατανέμεται σε τρία τμήματα και συγκεκριμένα στην ιστοσελίδα, όπου κανείς μπορεί να εξερευνήσει πακέτα, και να διαμορφώσει κάποιο λογαριασμό, στο CLI(Command Line Interface), τον τρόπο δηλαδή που θα αλληλοεπιδράσει ο χρήστης με το npm, και τέλος στο registry, το οποίο ουσιαστικά συνιστά μία δημόσιας εμβέλειας βάση δεδομένων λογισμικού JavaScript, μαζί με όλα τα απαραίτητα μεταδεδομένα .

4.11.2 Πώς χρησιμοποιείται με το Visual Studio Code

Αρχικά τρέξαμε την εντολή `npm install -g create-react-app` για να εγκαταστήσουμε τα πακέτα που μας παρέχει το react.



```
MINGW64:/c/Users/Sektas/Desktop
Sektas@SEKTAS MINGW64 ~/Desktop
$ npm install -g create-react-app
C:\Users\Sektas\AppData\Roaming\npm\create-react-app -> C:\Users\Sektas\AppData\Roaming\npm\node_modules\create-react-app\index.js
+ create-react-app@2.1.3
updated 2 packages in 25.009s

Sektas@SEKTAS MINGW64 ~/Desktop
$ .....
```

Εικόνα 65 - Εγκατάσταση του πακέτου `create-react-app` - (δικό μας)

Αφού έγινε η εγκατάσταση, πλέον μπορούμε να χρησιμοποιήσουμε το πακέτο για να δημιουργήσουμε μια εφαρμογή React με την εντολή `create-react-app my-app` όπως παρακάτω:

```
MINGW64:/c/Users/Sektas/Desktop
Sektas@SEKTAS MINGW64 ~/Desktop
$ create-react-app my-app

Creating a new React app in C:\Users\Sektas\Desktop\my-app.

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts...

+ react@16.7.0
+ react-dom@16.7.0
+ react-scripts@2.1.3
added 1725 packages in 697.096s

Initialized a git repository.

Success! Created my-app at C:\Users\Sektas\Desktop\my-app
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

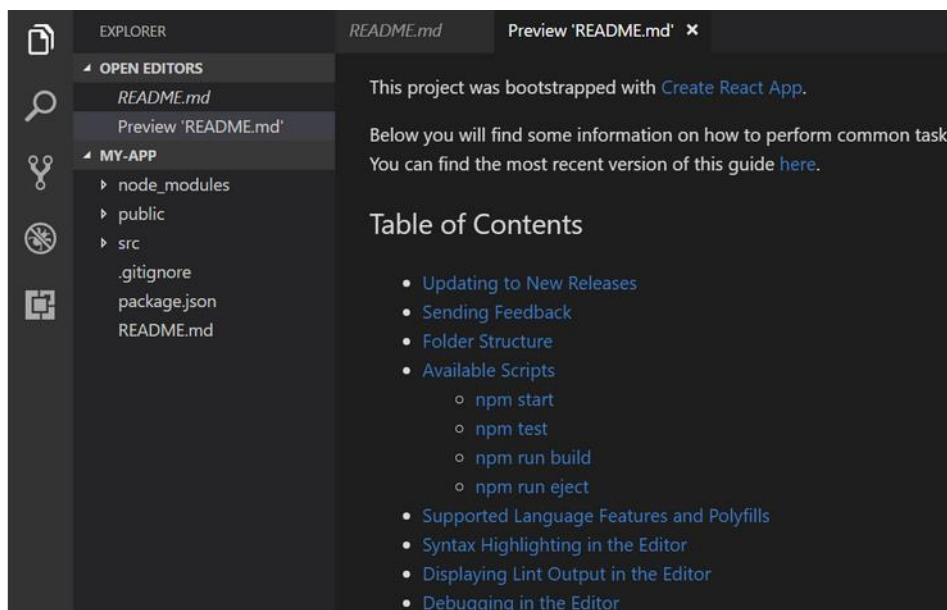
  cd my-app
  npm start

Happy hacking!
```

Εικόνα 66 - Εγκατάσταση του πακέτου create-react-app - (δικό μας)

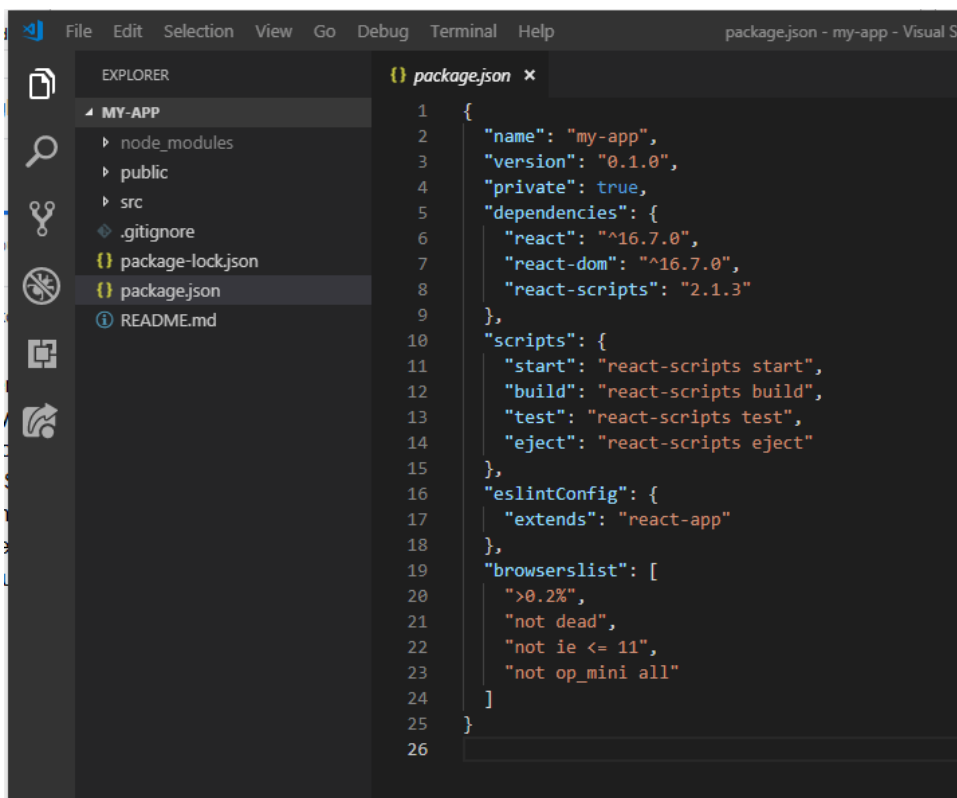
Έπειτα ανοίξαμε την εφαρμογή στο Visual Studio Code με την εντολή `code`.

Στον File Explorer του editor στο αρχείο README.md, μπορεί κάποιος χρήστης να δει σημαντικές πληροφορίες για την εφαρμογή που μόλις έκανε εγκατάσταση, αλλά και γενικά για το React.



Εικόνα 67 - Αρχείο README.md - Visual Studio Code - (δικό μας)

Επίσης, μετά την εγκατάσταση της εφαρμογής, δημιουργείται αυτόματα και ένα αρχείο με ονομασία `package.json` το οποίο περιέχει όλα τα `versions` και τις ονομασίες των `npm` πακέτων, τα οποία γίνονται εγκατάσταση κατά την αρχή, την εξέλιξη και στο τέλος της εφαρμογής από τον προγραμματιστή-χρήστη. Αυτό το αρχείο, ταξινομεί και κατατάσσει αυτά τα `npm` πακέτα ως `dependencies` όπως παρακάτω. Να σημειωθεί πως ο προγραμματιστής της εφαρμογής, μπορεί να πειράζει τα `dependencies` του αρχείου αυτού (π.χ. να θέλει να αλλάξει τα `versions` αλλά και να θέλει να προσθέσει καινούργια πακέτα) και έπειτα με την εντολή `npm install` να κάνει εγκατάσταση τα πακέτα που έχει παραμετροποιήσει.



```
1  {
2    "name": "my-app",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "react": "^16.7.0",
7      "react-dom": "^16.7.0",
8      "react-scripts": "2.1.3"
9    },
10   "scripts": {
11     "start": "react-scripts start",
12     "build": "react-scripts build",
13     "test": "react-scripts test",
14     "eject": "react-scripts eject"
15   },
16   "eslintConfig": {
17     "extends": "react-app"
18   },
19   "browserslist": [
20     ">0.2%",
21     "not dead",
22     "not ie <= 11",
23     "not op_mini all"
24   ]
25 }
26
```

Εικόνα 68 - Αρχείο `package.json` - Visual Studio Code - (δικό μας)

Το πακέτο `react-scripts` που ενεργοποιείται με τις εντολές `npm start`, `npm build`, `npm test` και `npm eject` εγκαθίσταται αυτόματα μαζί με την εφαρμογή.

Στο αρχείο `index.js`, στον φάκελο `src`, μπορεί κανείς να δει ποιο αρχείο χρησιμοποιεί το `Virtual DOM` του `React` για να τρέξει η εφαρμογή και να εμφανιστεί το περιεχόμενο στο

localhost:3000, αφού το πακέτο react-scripts που χρησιμοποιεί η εφαρμογή, έχει ορίσει ως default πόρτα την 3000.

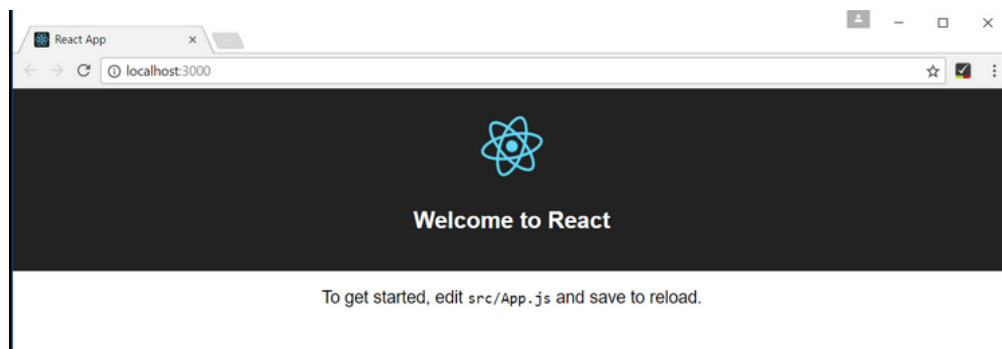
Το React δημιουργεί ως default αρχείο το App.js, όπως φαίνεται κάθε φορά που δημιουργείται μια καινούργια εφαρμογή και το θέτει σαν root στο Virtual DOM.

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import App from './App';

App.js src
1 import React, { Component } from 'react';
2 import logo from './logo.svg';
3 import './App.css';
4
5 class App extends Component {
6   render() {
7     return (
8       <div className="App">
9         <div className="App-header">
10          <img src={logo} className="App-logo" alt="logo" />
11          <h2>Welcome to React</h2>
12        </div>
13        <p className="App-intro">
14          To get started, edit <code>src/App.js</code> and save to reload.
15        </p>
16      </div>
17    );
18  }
19 }
20
21 import registerServiceWorker from './registerServiceWorker';
22 import './index.css';
```

Εικόνα 69 - React Default App.js - Visual Studio Code - (δικό μας)

Έπειτα, με την εντολή npm start, η εφαρμογή τρέχει και ανοίγει ένα καινούργιο παράθυρο στον default browser στην πόρτα 3000 ως εξής.



Εικόνα 70 - Αναπαράσταση App.js στον browser - (δικό μας)

Επίσης για να κάνουμε debug στον localhost, χρησιμοποιούμε τον Debugger for Chrome που διατίθεται σαν extension του VS Code



Εικόνα 71 - Debugger for Chrome Extension - Visual Studio Code - (δικό μας)

Έπειτα, προσθέσαμε Linters οι οποίες αναλύουν τον κώδικά μας και μπορούν να μας προειδοποιήσουν για πιθανά σφάλματα προτού τρέξουμε την εφαρμογή μας. Να σημειωθεί, ότι οι υπηρεσίες γλώσσας Javascript που περιλαμβάνονται με το Visual Studio Code, έχουν προεπιλεγμένη υποστήριξη ελέγχου σφαλμάτων.

Ένας δημοφιλής Javascript Linter, είναι ο ESLint. Η ESLint όταν συνδυάζεται με την επέκταση του VS Code, παρέχει μεγάλη εμπειρία στην διόρθωση κώδικα.

Πρώτα, κάναμε εγκατάσταση του πακέτου με την εντολή `npm install --g eslint` και έπειτα εγκαταστήσαμε το extension που παρέχει το VS Code όπως παρακάτω.



Εικόνα 72 - ESLint Extension - Visual Studio Code - (δικό μας)

4.11.3 Τα Npm πακέτα της πλατφόρμας μας

Ξεκινήσαμε όπως προαναφέραμε με την εγκατάσταση του πακέτου `create-react-app` με το οποίο φτιάξαμε μια εφαρμογή React στο Visual Studio Code. Έπειτα, με το σκεπτικό που είχαμε στο μυαλό μας για την εξέλιξη της εφαρμογής και τις λειτουργικές απαιτήσεις που με αυτήν θα έπρεπε να ικανοποιεί, προσθέσαμε

καινούργια πακέτα ημ τα οποία κατά την εγκατάσταση ενημερώνονταν αυτόματα μαζί με τα versions τους στο αρχείο package.json. Τα πακέτα που χρησιμοποιήσαμε αρχειοθετούνται σε δύο ξεχωριστές κατηγορίες στο αρχείο package.json. Αυτές οι κατηγορίες είναι τα απλά dependencies και τα devdependencies.

4.11.3.1 Dependencies

Τα απλά Dependencies περιέχουν βιβλιοθήκες και Frameworks τα οποία είναι ενσωματωμένα στην εφαρμογή μας, τέτοια Frameworks είναι αυτά που έχουμε αναφέρει όπως React, Vue, Angular, Express, JQuery κ.ά. Χωρίς αυτά τα dependencies, η εφαρμογή μας δεν θα έτρεχε.

4.11.3.2 DevDependencies

Τα Development Dependencies είναι πακέτα που προορίζονται μόνο για την ανάπτυξη της εφαρμογής, όχι όμως και για την παραγωγή της. Τέτοια πακέτα είναι π.χ. τα Webpack και Babel τα οποία αναλύονται στην συνέχεια.

Τέλος, αφού πήραμε μια πρώτη εικόνα για το πως ήταν το αρχείο package.json κατά την δημιουργία της εφαρμογής μας, ας δούμε πως διαμορφώθηκε κατά το πέρας της, καθώς και να αναλύσουμε το κάθε πακέτο που χρησιμοποιήσαμε ξεχωριστά.

4.11.3.3 Κατηγορία NPM Πακέτων Dependencies

```
"dependencies": {
  "@material-ui/core": "^3.6.2",
  "@material-ui/icons": "^3.0.1",
  "axios": "^0.18.0",
  "bootstrap": "^4.2.1",
  "botframework-webchat": "^4.1.0",
  "firebase": "^5.7.0",
  "firestore-react": "^0.1.5",
  "material-ui": "^0.20.2",
  "material-ui-icons": "^1.0.0-beta.36",
  "nuka-carousel": "^4.4.5",
  "path": "^0.12.7",
  "ra-input-rich-text": "^2.3.0",
  "react": "^16.4.2",
  "react-admin": "^2.3.1",
  "react-admin-firebase": "^0.2.8",
  "react-containers": "^16.0.5",
  "react-coverflow": "^0.2.17",
  "react-dom": "^16.6.3",
  "react-firebase": "^2.2.8",
  "react-firebase-ui": "^1.0.0",
  "react-firebaseui": "^3.1.2",
  "react-redux": "^5.1.1",
  "react-router": "^4.3.1",
  "react-router-dom": "^4.3.1",
  "reactstrap": "^7.0.0",
  "recompose": "^0.30.0",
  "redux-saga": "^0.16.0",
  "semantic-ui-react": "^0.84.0"
```

Εικόνα 73 - Αρχείο package.json - Visual Studio Code - (δικό μας)

4.11.3.3.1 Material-UI

Το Material-UI είναι ένα πακέτο npm ανοικτού κώδικα που προσδίδει styling σε Javascript αντικείμενα όπως Buttons, Dialogs κ.ά. Έχει αδειοδότηση MIT, με συνεχή εξέλιξη.

4.11.3.3.2 Axios

Το axios είναι ένα πακέτο npm βασισμένο σε promises. Τα promises είναι αντικείμενα τα οποία αντιπροσωπεύουν μια ενέργεια η οποία δεν έχει ακόμα ολοκληρωθεί, αλλά θα πραγματοποιηθεί κάποια στιγμή στο μέλλον. Είναι κατά κάποιο τρόπο ένα αντικείμενο δέσμευσης θέσης για το αποτέλεσμα ενός είδους ασύγχρονης λειτουργίας, όπως παραδείγματος χάρη ενός αιτήματος HTTP. Ένα τέτοιο αντικείμενο αντιπροσωπεύει το τελικό αποτέλεσμα μιας ασύγχρονης λειτουργίας. Με λίγα λόγια το axios βοηθάει τον προγραμματιστή να παρακολουθεί και να ακυρώνει αιτήματα, καθώς και να διατηρεί μια αξιολογη προστασία από την παραχάραξη αιτημάτων μεταξύ

διαφορετικών ιστοτόπων από την μεριά του client. Κάποια από τα χαρακτηριστικά του είναι:

- Πραγματοποίηση XML-Http Requests
- Πραγματοποίηση HTTP Requests από το node.js
- Υποβολή αιτημάτων και απαντήσεων
- Μετασχηματισμός δεδομένων αιτήματος και απόκρισης
- Ακύρωση αιτημάτων
- Αυτόματη μετατροπή δεδομένων JSON
- Υποστήριξη από πλευρά client έναντι XSRF(Cross-site Request Forgery) επιθέσεων.

4.11.3.3.3 Botframework - Webchat (Channel)

Το WebChat είναι διαθέσιμο τόσο ως Component της βιβλιοθήκης React, όσο και αυτοτελής οντότητα η οποία μπορεί εύκολα να χρησιμοποιηθεί σε οποιοδήποτε ιστότοπο μη-React. Αυτή την χρονική περίοδο αναπτύσσεται με TypeScript, χρησιμοποιώντας το Redux για το state management και το RxJS για async μεθόδους (ασύγχρονος προγραμματισμός). Μπορούμε να το ενσωματώσουμε στον ιστότοπό μας μέσω του DirectLine API το οποίο είναι εύκολα προσβάσιμο μέσω του καναλιού DirectLine από το Azure Web App χρησιμοποιώντας μόνο το κλειδί.

Χρησιμοποιείται από το portal του Bot Framework, και στο Azure Bot Service μέσω του καναλιού Web Chat.

4.11.3.3.4 Firebase

Το πακέτο npm firebase, παρέχει τα εργαλεία και την υποδομή που χρειάζεται ένας προγραμματιστής προκειμένου να αναπτύξει μια εφαρμογή, να αυξήσει την αποδοτικότητά της και να κερδίσει χρήματα από αυτήν. Αυτό το πακέτο

υποστηρίζει web (browser) clients, mobile-web clients καθώς και server (Node.js) clients. Κάποια από τα χαρακτηριστικά του είναι:

- **Firestore Realtime Database** - Η βάση δεδομένων του Firebase επιτρέπει στους προγραμματιστές να αποθηκεύουν και να αναζητούν δεδομένα χρηστών καθώς και να τα καθιστούν διαθέσιμα μεταξύ των χρηστών
- **Cloud Firestore** - Το Cloud Firestore του Firebase είναι μια ευέλικτη βάση δεδομένων για την ανάπτυξη mobile, web και server applications από το Google Cloud Platform.
- **Firestore Storage** - Το Firestore Storage επιτρέπει την αποθήκευση και το upload περιεχομένου από χρήστες, όπως αρχεία και εικόνες.
- **Firestore Cloud Messaging** - Πρόκειται για μια λύση ανταλλαγής μηνυμάτων μεταξύ διαφορετικών πλατφορμών με αξιοπιστία και χωρίς κόστος.
- **Firestore Authentication** - Βοηθά τους προγραμματιστές στην πιστοποίηση και διαχείριση των χρηστών που έχουν πρόσβαση στην εφαρμογή τους.

4.11.3.3.5 firestore-react

Παρέχει δύο λειτουργίες:

- Λαμβάνει δεδομένα και τα περνάει σε React Components
- Προσθέτει έναν συνδρομητή για να ακούσει τυχόν ενημερώσεις στιγμιότυπων στην βάση, και επίσης αφαιρεί τον συνδρομητή όταν κάποιο component είναι unmounted, το οποίο επιστρέφεται από την lifecycle method που αναφέραμε σχετικά με το React, `componentDidMount()`.

4.11.3.3.6 React-alice-carousel

Πρόκειται για ένα πακέτο npm το οποίο το χρησιμοποιήσαμε προκειμένου να δημιουργήσουμε ένα Carousel Component, το οποίο θα περιείχε εικόνες από τα μέρη τα οποία έχουν αναζητήσει περισσότερο οι χρήστες στην πλατφόρμα μας.

4.11.3.3.7 react-admin

Πρόκειται για ένα frontend Framework το οποίο χρησιμοποιείται για την ανάπτυξη εφαρμογών που περιέχουν admin panels και τρέχουν σε κάποιον browser βασιζόμενο σε υπηρεσίες REST/GraphQL, χρησιμοποιώντας ES6, React και Material Design. Πρόκειται για ανοιχτό λογισμικό και διατηρείται από την marmelab.

4.11.3.3.8 React-admin-firebase

Πρόκειται για ένα πάροχο δεδομένων (Data Provider), ο οποίος συλλέγει δεδομένα από documents στην βάση δεδομένων του Firebase στην εφαρμογή react-admin που χρησιμοποιείται ως dashboard για τους χρήστες που έχουν ρόλο admin, έτσι ώστε να ελέγχουν τις εφαρμογές τους.

4.11.3.3.9 react-containers

Το παραπάνω πακέτο χρησιμοποιήθηκε για τον κώδικά μας, καθώς δίνει την δυνατότητα στους προγραμματιστές να δηλώνουν components στο react και να τα χρησιμοποιούν σαν wrappers έτσι ώστε να είναι πιο εύκολη η εφαρμογή μας στο styling αλλά και στην καλύτερη ανάγνωση του πηγαίου κώδικα.

4.11.3.3.10 React-dom

Αυτό το πακέτο χρησιμοποιείται ως σημείο εισόδου στο Virtual DOM του React αλλά και στην μέθοδο render που αποτελεί την πιο σημαντική μέθοδο για πέρασμα στοιχείων στα react components.

4.11.3.3.11 React-firebase

Συνδέει ένα React Component με μια αναφορά-document του Firebase API. Δεν τροποποιεί την κλάση Component στην οποία γίνεται αναφορά. Αντιθέτως, επιστρέφει μια συνδεδεμένη κλάση Component την οποία μπορεί να χρησιμοποιήσει ο προγραμματιστής.

4.11.3.3.12 React-firebaseui

Αυτό το πακέτο χρησιμοποιείται για να δηλώσει κάποια react components ονομαζόμενα και ως Firebase UI React Components τα οποία παρέχουν React Wrappers στο Web UI του Firebase και κυρίως του Firebase UI Auth που χρησιμοποιούμε για να αυθεντικοποιήσουμε τους χρήστες της πλατφόρμας.

4.11.3.3.13 React-redux

Το React Redux διατηρείται από την ομάδα Redux και ενημερώνεται συνεχώς με τα πιο πρόσφατα API από την Redux και την React. Είναι σχεδιασμένο για να λειτουργεί με το μοντέλο components του React. Ο προγραμματιστής ορίζει τον τρόπο εξαγωγής των τιμών που χρειάζονται τα component από το Redux και το component τις λαμβάνει ως props. Υλοποιεί αυτόματα συναρτησιακά components (functional components), έτσι ώστε αυτά να αναπαράγονται με την μέθοδο render όταν τα δεδομένα στα οποία αναφέρονται έχουν αλλάξει και όχι όταν παραμένουν στάσιμα.

4.11.3.3.14 React-router-dom

Το React-router συγχρονίζει την εφαρμογή μας κατά την εκτέλεσή της με τις διευθύνσεις url που εμείς αναθέτουμε στον κώδικα. Διαθέτει ένα απλό API με ισχυρά χαρακτηριστικά όπως το lazy code loading, το dynamic route matching και το location transition handling.

1. Βασικό Routing

Στην περίπτωση μας χρησιμοποιήθηκε στο UserDashboard Component καθώς ως βασικό route με την κατάληξη / χρησιμοποιήθηκε το SearchPage και έπειτα ορίσαμε το Navigation Component να λειτουργεί σαν wrapper στα routes-urls που βλέπει ο χρήστης.

```
render((  
  <BrowserRouter>  
    <UserDashboard />  
  </BrowserRouter>  
) , document.getElementById('root'));
```


Εικόνα 74 - Βασικό Routing - Notepad++ - (δικό μας)

2. 404 Route

Σε περίπτωση που κάποιο url που θα πληκτρολογούσε ο χρήστης της εφαρμογής δεν υπήρχε χρησιμοποιήσαμε ένα εναλλακτικό route το NoMatch.

```
<Route component={() => <NoMatch />} />
```

Εικόνα 75 - Routing Error - Notepad++ - (δικό μας)

3. Link Routes

Όσον αφορά αυτήν την κατηγορία των routes στο react, χρησιμοποιήσαμε σαν route links δύο react components. Το ένα είναι αυτό το οποίο είναι διαθέσιμο προς τους χρήστες ακόμα και αν δεν έχουν κάνει κάποιο login στην πλατφόρμα μας και ονομάζεται AppBarNonAuth στο component Navigation και το άλλο είναι το περιεχόμενο το οποίο μπορούν να δουν οι χρήστες μετά το authentication τους στην πλατφόρμα μας μέσω του Firebase, το οποίο ονομάζεται AppBar και βρίσκεται επίσης στο react component Navigation. Και στις δύο περιπτώσεις χρησιμοποιήθηκαν Javascript αντικείμενα buttons από το πακέτο material-ui.

```
<Button component={Link} to={routes.ABOUTUS}>About Us</Button>  
<Button component={Link} to={routes.HOME}>DETAILS</Button>  
<Button component={Link} to={routes.ACCOUNT}>ACCOUNT</Button>  
<Button component={Link} to={routes.SEARCHPAGE}>SEARCH</Button>  
<Button component={Link} to={routes.BOT}>BOT</Button>  
<Button component={Link} to={routes.RECOMEND}>RECOMENDATIONS</Button>
```

Εικόνα 76 - Routing Links - Notepad++ - (δικό μας)

```
<Button component={Link} to={routes.ABOUTUS}>About Us</Button>  
<Button component={Link} to={routes.SIGN_IN}>SIGN IN</Button>  
<Button component={Link} to={routes.SIGN_UP}>SIGN UP</Button>  
<Button component={Link} to={routes.FACEBOOKLOGIN}>GOOGLE LOGIN</Button>  
<Button component={Link} to={routes.SEARCHPAGE}>SEARCH</Button>
```

Εικόνα 77 - Routing Links - Notepad++ - (δικό μας)

4.11.3.3.15 Redux-saga

Το Redux Saga είναι μια βιβλιοθήκη που χρησιμοποιείται για την αντιμετώπιση των προβλημάτων στο Redux. Όταν πυροδοτείται μια ενέργεια αλλάζει κάτι στην κατάσταση της εφαρμογής και ίσως χρειαστεί να κάνουμε κάτι που προέρχεται από αυτήν την αλλαγή στο state της κλάσης του component που δουλεύουμε. Χρησιμοποιείται σε περιπτώσεις που επιθυμούμε να:

- Να στείλουμε ένα event είδους websocket
- Να πάρουμε κάποια δεδομένα από έναν server τύπου GraphQL
- Να κάνουμε αποθήκευση δεδομένων στην κρυφή μνήμη ή στον αποθηκευτικό χώρο του browser που χρησιμοποιούμε

4.11.3.4 Κατηγορία NPM Πακέτων DevDependencies

```
"devDependencies": {
  "babel-core": "^6.25.0",
  "babel-eslint": "^7.2.3",
  "babel-loader": "^7.1.1",
  "babel-plugin-transform-es2015-arrow-functions": "^6.22.0",
  "babel-preset-es2015": "^6.24.1",
  "babel-preset-react": "^6.24.1",
  "babel-preset-stage-1": "^6.24.1",
  "css-loader": "^0.28.4",
  "eslint": "^4.2.0",
  "eslint-plugin-babel": "^4.1.2",
  "eslint-plugin-react": "^7.1.0",
  "file-loader": "^0.11.2",
  "html-webpack-plugin": "^2.30.1",
  "image-webpack-loader": "^4.5.0",
  "node-sass": "^4.5.3",
  "prop-types": "^15.6.2",
  "sass-loader": "^6.0.6",
  "style-loader": "^0.18.2",
  "superagent": "^3.8.3",
  "webpack": "^3.3.0",
  "webpack-dev-server": "^2.5.1"
}
```

Εικόνα 78 - Αρχείο package.json - Visual Studio Code - (δικό μας)

4.11.3.4.1 Babel-core

Το πακέτο npm Babel είναι μια εργαλειομηχανή που χρησιμοποιείται για την μετατροπή κώδικα τύπου Next Generation Javascript, σε μια συμβατή με το παρελθόν

έκδοση της Javascript σε τρέχοντα και παλαιότερα προγράμματα περιήγησης. Τα κύρια χαρακτηριστικά που μας προσφέρει αυτό το πακέτο είναι:

- Μετατροπή της σύνταξης
- Χαρακτηριστικά Polyfill (κώδικας που υλοποιεί μια λειτουργία σε προγράμματα περιήγησης που δεν υποστηρίζουν την λειτουργία) που λείπουν από το περιβάλλον ανάπτυξης της εφαρμογής μας
- Μετασχηματισμοί πηγαίου κώδικα (codemods)

4.11.3.4.1 Eslint

Το npm eslint, είναι ένα εργαλείο για τον εντοπισμό προτύπων που βρίσκονται στον κώδικα Next Generation Javascript/Javascript. Κάποια από τα χαρακτηριστικά του πακέτου αυτού είναι:

- Το eslint χρησιμοποιεί το Espree (ένας Javascript parser) για την ανάλυση Javascript κώδικα
- Το ESLint χρησιμοποιεί ένα AST (ABSTRACT SYNTAX TREE) για να αξιολογήσει τα πρότυπα στον κώδικα.
- Το ESLint είναι εντελώς pluginable, κάθε κανόνας είναι ένα plugin και μπορείτε να προσθέσετε περισσότερα κατά το χρόνο εκτέλεσης.

4.11.3.4.2 File-loader

Το πακέτο npm file-loader επιλύει την διαδικασία εισαγωγής αρχείων στον κώδικά μας. Παραδείγματος χάρη για το React, επιλύει εντολές τύπου import στην αρχή του κώδικα ενώ για την Javascript εντολές τύπου require(). Το πακέτο αυτό χρησιμοποιείται για την επίλυση εντολών τύπου @import και url(), ερμηνεύοντας τα σαν import/require().

4.11.3.4.3 Html-webpack-plugin

Plugin το οποίο απλοποιεί την δημιουργία αρχείων html για την εκτέλεση του κώδικα της εφαρμογής μας.

4.11.3.4.4 Image-loader-webpack

Πρόκειται για έναν φορτωτή εικόνων για το πακέτο webpack τύπου PNG, JPEG, GIF, SVG, WEBP.

4.11.3.4.5 Node-sass

Πρόκειται για μια βιβλιοθήκη που παρέχει binding και συγκεκριμένα την μεταγλώττιση αρχείων τύπου scss σε css.

4.11.3.4.6 Prop-types

Πρόκειται για ένα πακέτο που παρέχει έλεγχο για τα props των components του React. Συγκεκριμένα, η μέθοδος που χρησιμοποιεί είναι η `checkPropTypes()` η οποία ελέγχει τα props(αντικείμενα) που περνιούνται στα React Components και προειδοποιεί την εφαρμογή μας αν αυτά δεν ταιριάζουν με την αναφορά που έχει γίνει σε αυτά.

4.11.3.4.7 Sass-loader

Παρόμοιος ορισμός με αυτό που δόθηκε για το πακέτο node-sass, δηλαδή μεταγλωττίζει ένα scss αρχείο σε css.

4.11.3.4.8 Style-loader

Προσθέτει css στο DOM του React, εισάγοντας μια ετικέτα `<style>`.

4.11.3.4.9 Superagent

Πρόκειται για ένα προοδευτικό AJAX API, το οποίο χρησιμοποιείται για την αποστολή αιτήσεων HTTP σε διάφορες εφαρμογές. Είναι παρόμοιο με το πακέτο axios που αναφέρθηκε παραπάνω.

4.11.3.4.10 Webpack

Πρόκειται για ένα από τα πιο σημαντικά πακέτα, γι' αυτό και έγινε μια πιο εκτενής αναφορά σε αυτό. Ο κύριος σκοπός του είναι η σύνδεση των αρχείων Javascript που χρησιμοποιούνται για την εκτέλεση σε ένα πρόγραμμα περιήγησης.

4.11.3.4.11 Webpack-dev-server

Χρησιμοποιεί το webpack-dev-middleware, το οποίο παρέχει γρήγορη πρόσβαση στα στοιχεία του webpack

4.12 HTML

4.12.1 Τι είναι

Η *HTML* είναι το ακρωνύμιο των λέξεων *HyperText Markup Language* (γλώσσα μορφοποίηση υπερκειμένου) και είναι η βασική γλώσσα δόμησης σελίδων του *World Wide Web* (ή απλά ιστού: Web). Ορίζεται ως μια γλώσσα σήμανσης. Χρησιμοποιεί ένα αριθμό από *tags* για την μορφοποίηση κειμένου, για την δημιουργία συνδέσμων (*links*) μετάβασης ανάμεσα των σελίδα, για την εισαγωγή εικόνων, ήχου κ.ά. Όταν ένας Web Browser ανοίγει ένα αρχείο HTML τα στοιχεία (*tags*) μεταφράζονται σε κατάλληλα χαρακτηριστικά με αποτελέσματα στην εμφάνιση και στην λειτουργικότητα της συγκεκριμένης σελίδας. Αρχικά, είχε κατασκευαστεί με σκοπό μόνο την μορφοποίηση κειμένου, αλλά μεγάλωσε και ενσωμάτωσε σχεδιαστικές τεχνικές κ.ά.

Η HTML ορίζει την δομή και την διάταξη ενός web εγγράφου χρησιμοποιώντας μια ποικιλία από tags και attributes. Η σωστή δομή για ένα έγγραφο HTML ξεκινάει με

<HTML><HEAD>(πληκτρολογήστε εδώ το έγγραφο) <BODY> και τελειώνει με </BODY></HTML>. Όλες οι πληροφορίες που θέλουμε να συμπεριλάβουμε στην ιστοσελίδα μας είναι τοποθετημένες μεταξύ των ετικετών <BODY> και </BODY>.

4.12.2 Εκδόσεις HTML

Το 1990 ο Tim Berners-Lee από το Cern, το εργαστήριο φυσικής της Γενεύης, δημιούργησε ένα νέο πρωτόκολλο με το οποίο θα μπορούσαν να μεταφέρονται κάθε είδος αρχείων και αντικειμένων μέσα από το Internet. Το πρωτόκολλο αυτό ονομάστηκε HTTP (HyperText Transfer Protocol) και σηματοδότησε την αρχή του WWW όπως το ξέρουμε σήμερα. Οι σελίδες που ήταν η βάση του WWW ήταν γραμμένες στην πρώτη έκδοση της γλώσσα HTML.

Το 1994 αναπτύσσεται το πρότυπο HTML 2.0 από ένα διεθνή οργανισμό (Internet Engineering Task Force). Η επόμενη έκδοση η 3.0 δεν έγινε αποδεκτή από τις Microsoft και Netscape οπότε γρήγορα αντικαταστάθηκε από την έκδοση 3.2 (1996). Η τελευταία περιελάμβανε πολλές από τις σημάνσεις (tags) που είχαν εισάγει οι δύο εταιρείες. Η έκδοση 4.0 παρουσιάστηκε τον Ιούνιο του 1997.

4.12.3 Δημιουργία αρχείων HTML

Η δημιουργία αρχείων HTML είναι πολύ απλή. Αρκεί να “τρέξουμε” έναν οποιοδήποτε διορθωτή *κειμένου text*, όπως το *Notepad* των Windows. Να γράψουμε τον κώδικα HTML που επιθυμούμε και να το αποθηκεύσουμε (σώσουμε) σε ένα αρχείο με κατάληξη **.htm** ή **.html**. Συνήθως αποθηκεύουμε με επέκταση **.html** όταν θα δημοσιεύσουμε τα αρχεία μας σε διακομιστή Unix. Αντίθετα ο εξυπηρετητής Windows NT καταλαβαίνει τα αρχεία και σαν **.htm** και σαν **.html**.

4.12.4 CSS

Το CSS ορίζεται ως Cascading Style Sheets με έμφαση στο “Style”. Σε αντίθεση με την HTML, η οποία χρησιμοποιείται για την δομή ενός web εγγράφου, το CSS καθορίζει το στυλ της σελίδας, τα χρώματα τις γραμματοσειρές κ.ά. Με λίγα λόγια, το HTML λειτουργεί ως θεμέλιο (όπως κάθε σπίτι έχει) και το CSS ως τις αισθητικές επιλογές.

4.12.5 SCSS

Το SCSS είναι ένας ειδικός τύπος αρχείου για το SASS, ένα πρόγραμμα γραμμένο σε Ruby που συγκεντρώνει CSS style sheets για ένα πρόγραμμα περιήγησης και για πληροφορίες. Πληροφοριακά, το SASS προσθέτει πολλές πρόσθετες λειτουργίες σε CSS όπως variables (μεταβλητές), nesting, και πολλά άλλα τα οποία καθιστούν την γραφή σε CSS ευκολότερη και γρηγορότερη. Τα αρχεία SCSS, υποβάλλονται σε επεξεργασία από τον server (διακομιστή) που εκτελεί μια εφαρμογή ιστού για την εξαγωγή ενός κλασσικού CSS το οποίο μπορεί να κατανοήσει το πρόγραμμα περιήγησης.

4.13 Foursquare (API)

4.13.1 Τι είναι

Ας προχωρήσουμε στον διαχωρισμό ανάμεσα στην ίδια την εταιρεία και το API της. Το Foursquare είναι με εταιρεία τεχνολογίας, η οποία χρησιμοποιεί την τοποθεσία ώστε να θεμελιώσει εμπειρίες γεμάτες νόημα για τους πελάτες και επίσης σε δεύτερο επίπεδο να προσφέρει επιχειρηματικές λύσεις.

Για τους καταναλωτές(πελάτες/consumers), διαθέτει δύο εφαρμογές. Το Foursquare City Guide και το Foursquare Swarm. Η πρώτη δίνει τη δυνατότητα εξερεύνησης νέων τοποθεσιών και προσφέρει προτάσεις από μία αξιόπιστη κοινότητα. Ιδιαίτερο ενδιαφέρον, παρουσιάζει η εφαρμογή Swarm, η οποία διατηρεί τα ίχνη από όλα τα μέρη που έχεις επισκεφθεί και όντας συνεχώς μαζί σου, σε κινητοποιεί για νέες και πιο ενδιαφέρουσες εμπειρίες μέσα από την πραγματοποίηση ταξιδιών.

Όσον αφορά τις επιχειρήσεις, τους δίνεται η δυνατότητα να επικοινωνήσουν με τους δικούς τους πελάτες και να συνειδητοποιήσουν το μέγεθος της απήχησης που διαθέτουν προς αυτούς. Διατίθεται πλήθος λύσεων, οι οποίες περιλαμβάνουν το Pinpoint by Foursquare, Attribution by Foursquare, το Place Insights by Foursquare και το Pilgrim SDK by Foursquare, μέσω των οποίων οι ίδιες οι επιχειρήσεις μπορούν να κατανοήσουν και να συνδεθούν με τα κοινά-στόχους (targeted audiences), να μετρήσουν την κίνηση και την επιτυχία που ανακύπτει από τις διαφημίσεις.

4.13.2 Πώς λάβαμε δεδομένα από το Foursquare API

Η διαδικασία αυτήν πραγματοποιήθηκε μέσω του Node.js, το οποίο μπορεί να γίνει με δύο τρόπους 1) με κανονικό request, 2) με κάποια βιβλιοθήκη/ πακέτο από το npm (όπως το axios ή το superagent).

Τα αποτελέσματα τα οποία πήραμε ήταν

- Πληροφορίες σχετικά με ένα Foursquare Venue

Η μορφή του request που στείλαμε στο foursquare ήταν `GET https://api.foursquare.com/v2/venues/VENUE_ID` το οποίο επιστρέφει πληροφορίες σχετικά με το συγκεκριμένο venue όπως τοποθεσία, tips, κατηγορίες. Το response που πήραμε από το Foursquare και περιείχε τις συγκεκριμένες πληροφορίες ήταν της μορφής

```
"response": {
  "venue": {
    "id": "412d2800f964a520df0c1fe3",
    "name": "Central Park",
    "contact": {
      "phone": "2123106600",
      "formattedPhone": "(212) 310-6600",
      "twitter": "centralparknyc",
      "instagram": "centralparknyc",
      "facebook": "37965424481",
      "facebookUsername": "centralparknyc",
      "facebookName": "Central Park"
    },
    "location": {
      "address": "59th St to 110th St",
```

Εικόνα 79 - Foursquare Venue details

<https://developer.foursquare.com/docs/api/venues/details>

- Venues τα οποία βρίσκονται κοντά σε μια συγκεκριμένη τοποθεσία

Η μορφή του request που στείλαμε στο foursquare ήταν `GET https://api.foursquare.com/v2/venues/search`

.

Το response που μας επέστρεψε το api ήταν της μορφής


```

"response": {
  "venues": [
    {
      "id": "5642aef9498e51025cf4a7a5",
      "name": "Mr. Purple",
      "location": {
        "address": "180 Orchard St",
        "crossStreet": "btwn Houston & Stanton St",
        "lat": 40.72173744277209,
        "lng": -73.98800687282996,
        "labeledLatLngs": [

```

Εικόνα 80 - Foursquare Venue details -
<https://developer.foursquare.com/docs/api/venues/details>

- Επιστροφή φωτογραφιών για συγκεκριμένα venues

Η μορφή του request που στείλαμε στο foursquare ήταν

GET https://api.foursquare.com/v2/venues/VENUE_ID/photos

στο οποίο το api ανταποκρίθηκε με ένα response της μορφής

```

"response": {
  "photos": {
    "count": 1,
    "items": [
      {
        "id": "51e4151c498e60b5d17bc721",
        "createdAt": 1373902108,
        "source": {
          "name": "Instagram",
          "url": "http://instagram.com"
        },
        "prefix": "https://igx.4sqi.net/img/general/",
        "suffix": "/1022386_wEd7VXnWzp51ajvRLtAPunoifDSIq8PMEtEuIH2Tzg.jpg",
        "width": 612,
        "height": 612,

```

Εικόνα 81 - Foursquare Photo of Venue -
<https://developer.foursquare.com/docs/api/photos/details>

- Επιστροφή σχολίων χρηστών που έχουν επισκεφτεί τα συγκεκριμένα venues

Η μορφή του request που στείλαμε στο foursquare ήταν

GET

https://api.foursquare.com/v2/venues/VENUE_ID/tips

και το αναμενόμενο ήταν να λάβουμε ένα response σε μορφή JSON όπως και τα προηγούμενα requests της μορφής

```
"response": {
  "tips": {
    "count": 231,
    "items": [
      {
        "id": "53bc46b7498e355aed38c696",
        "createdAt": 1404847799,
        "text": "A true lower east side spectacle with a retractable rooftop and dance floor in the basement. Come here for the nights you won't remember with the friends you won't forget!",
        "author": "-----"
```

Εικόνα 82 - Foursquare Tips of Venue - <https://developer.foursquare.com/docs/api/tips/details>

4.14 OpenWeatherMap (API)

4.14.1 Τι είναι

Το OpenWeatherMap είναι μια ηλεκτρονική υπηρεσία που παρέχει δεδομένα σχετικά με τον καιρό, συμπεριλαμβανομένων των τρεχόντων δεδομένων για καιρικές συνθήκες, των προβλέψεων και δεδομένα του παρελθόντος. Χρησιμοποιεί δεδομένα από υπηρεσίες μετεωρολογικής μετάδοσης τα οποία προέρχονται από ραντάρ και μετεωρολογικούς σταθμούς.

4.14.2 Πως χρησιμοποιείται

Το OpenWeatherMap παρέχει ένα API σε μορφή JSON, XML και HTML με χρήση χωρίς χρέωση αλλά σε έναν περιορισμένο βαθμό. Κάνοντας χρήση επί πληρωμή, ο προγραμματιστής μπορεί να κάνει κλήσεις (request) σε ποσότητα μεγαλύτερη των 60/λεπτό.

Οι χρήστες, μπορούν λοιπόν να ζητήσουν τρέχουσες καιρικές προβλέψεις, όπως για παράδειγμα στην εφαρμογή μας, την τρέχουσα θερμοκρασία μιας τοποθεσίας. Παρακάτω φαίνεται και το αντίστοιχο μπλοκ κώδικα.

```

function (session, results) {
  session.userData.location = results.response;
  const uri = "http://api.openweathermap.org/data/2.5/weather?q=" + results.response + "&units=metric&appid=" + "API ID";

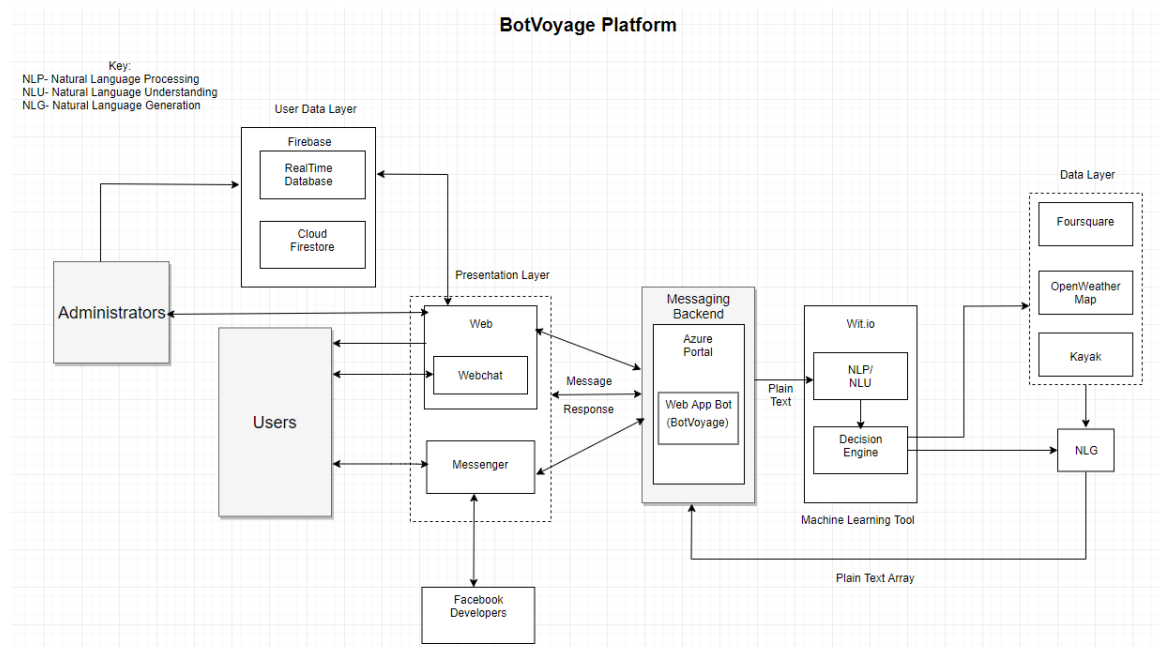
  // Use request to find the location through the OpenWeatherMap API
  request(uri, (error, response, body) => {
    if (!error && response.statusCode === 200) {
      const res = JSON.parse(body);
      if (session.userData.lingua == 'greek') {
        // Prompt for the users location
        session.send("Αυτήν την στιγμή, η θερμοκρασία στην τοποθεσία " + res.name + " είναι %d°C", res.main.temp);
        var iconData = res.weather[0].icon;

        var msg = new builder.Message(session);
        var msg = new builder.Message(session).addAttachment({contentType: 'image/png', contentUrl: "http://openweathermap.org/img/w/" + iconData + ".png"});
        builder.Prompts.text(session, msg);
      }
    }
  });
}

```

Εικόνα 83 - OpenWeatherMap Request block - Azure Portal - (δικό μας)

4.15 BotVoyage Platform Block Diagram



Στο παραπάνω διάγραμμα(Block Diagram), παρουσιάζονται τα στοιχεία από τα οποία αποτελείται η πλατφόρμα BotVoyage. Όλα τα δεδομένα και τα αποτελέσματα που εμφανίζονται στους χρήστες της πλατφόρμας προέρχονται από τα υποφαινόμενα Data Layer. Το User Data Layer αφορά τα στοιχεία χρηστών και διαχειριστών τα οποία ενσωματώνονται σε δύο βάσεις δεδομένων αντίστοιχα, που μας παρέχει το Firebase (Real time Database, Cloud Firestore). Εκείνα τα οποία αφορούν κατηγορίες και μέρη, πηγάζουν από τα Foursquare, OpenWeatherMap και Kayak APIs (Data Layer). Αυτά, παρουσιάζονται στους χρήστες μέσω των διεπαφών του Presentation Layer, δηλαδή στο Web μέσω Webchat ή SearchBar και σε Messenger. Η εφαρμογή BotVoyage (bot),

δημιουργήθηκε και συντηρείται στο Facebook for Developers προκειμένου να λειτουργεί ομαλά στο Messenger, αλλά όλες οι υπόλοιπες (βασικές και μη) διαδικασίες, προέρχονται από το Web App Bot, το οποίο στεγάζεται στο Azure Portal. Η βασική διαδικασία εξόρυξης δεδομένων πραγματοποιείται μέσω του Messaging Backend, το οποίο στέλνει τις επιλογές του χρήστη στο Wit.io (Machine Learning Tool) για να αναγνωριστούν μέσα από τις διεπαφές του (NLP/ NLU). Ανάλογα με το είδος της οντότητας επιλέγεται το αντίστοιχο API. Τέλος, για να αναπαρασταθούν τα δεδομένα στον χρήστη, χρησιμοποιείται ξανά το Messaging Backend, αυτή τη φορά όμως μέσω της διεπαφής NLG.

4.16 Προβλήματα και περιορισμοί εργαλείων

Αρχικά, το μεγαλύτερο πρόβλημα που παρουσιάστηκε ήταν η τοποθεσία με την χρήση Google Maps από το Geocoding API, που χρησιμοποιεί το Facebook με την μέθοδο quick replies έτσι ώστε να παίρνει τις τοποθεσίες χρηστών και να τις εμφανίζει στον χάρτη. Εμείς αυτό που θέλαμε από την πρώτη στιγμή, ήταν να φορτώσουμε το bot μέσα στην πλατφόρμα. Όταν το καταφέραμε με την χρήση του Webchat, στείλαμε το πρώτο μήνυμα στο bot, αυτό μας ζήτησε να στείλουμε την τοποθεσία μας με ένα text. Αυτό μας προκάλεσε μεγάλη περιέργεια. Ύστερα από πολλές μέρες έρευνας καταλάβαμε πως η μέθοδος quick replies σε παροχές τοποθεσίας, χρησιμοποιείται αποκλειστικά από την πλατφόρμα messenger και δεν παρέχει λειτουργίες σε προγραμματιστές έτσι ώστε να μπορούν να παίρνουν την τοποθεσία των χρηστών στους δικούς τους ιστοτόπους. Τελικά, μετά από πολλές προσπάθειες, το πρόβλημα λύθηκε παίρνοντας την τοποθεσία του χρήστη σαν κείμενο και από εκεί περνώντας το στα API όπως π.χ. το Foursquare βρίσκοντας έτσι χωρίς συντεταγμένες την πόλη επιλογής (π.χ. 'Λονδίνο'). Αυτό είναι πολύ πιο χρήσιμο, από το να παίρνει την τοποθεσία του χρήστη σε έναν χάρτη και να του εμφανίζει κατηγορίες για την συγκεκριμένη τοποθεσία, γιατί παραδείγματος χάρη αν ένας χρήστης είναι στην Αθήνα μπορεί να αναζητήσει μόνο για την Αθήνα και όχι για κάποιο άλλο μέρος.

Έχοντας αναφέρει για το messenger και την χρήση του bot από Facebook accounts-χρήστες, μπορούμε να πούμε ότι η Facebook platform για προγραμματιστές, στην οποία στεγάζεται το bot γι' αυτήν την μέθοδο-χρήσης, μας φάνηκε κάπως παράξενη και δυσνόητη. Με λίγα λόγια, η συγκεκριμένη πλατφόρμα έχει 2-3 αρνητικά για μας στοιχεία και ένα από αυτά είναι η συνεχής ανακατασκευή όλης της πλατφόρμας αλλά και

πιο συγκεκριμένα στην διαδικασία δημιουργίας μιας εφαρμογής bot για Messenger. Όταν παραλάβαμε το bot για να το επεκτείνουμε, η πλατφόρμα του Facebook έκανε μια ανακατασκευή όπως προαναφέρθηκε, κάτι το οποίο μας αφαίρεσε πολύ χρόνο ώστε να επαναφέρουμε το bot στην λειτουργική κατάσταση στην οποία βρισκόταν πριν. Παρ' όλ' αυτά, το Messenger της Facebook είναι ένα από τα επικρατέστερα μέσα κοινωνικής δικτύωσης αυτήν την εποχή και θα ήταν αδύνατο να μην υπάρχει το bot μας εκεί.

Ένα άλλο πρόβλημα που παρουσιάστηκε ήταν το ότι δεν μπορούσαμε να κάνουμε fetch τα αποτελέσματα που είχαμε στην βάση Cloud Firestore του Firebase στην πλατφόρμα μας. Η βάση αυτή, χρησιμοποιήθηκε για την διαχείριση των λειτουργιών του admin dashboard έτσι ώστε οι χρήστες που έχουν ρόλο admin στην πλατφόρμα, να μπορούν να στέλνουν μηνύματα και να επικοινωνούν με χρήστες μέσω των υπηρεσιών firebase Cloud Messaging και firebase Cloud Functions. Παρόλο που αυτό υλοποιήθηκε για την βάση Realtime Database, η οποία περιέχει τα στοιχεία των χρηστών της πλατφόρμας και την δηλώνουμε στο αρχείο firebase ως `const db = firebase.database()`, όταν πηγαίναμε να δηλώσουμε την βάση firestore ως `const fir = firebase.firestore()` για να μπορέσουμε να εξάγουμε δεδομένα από την βάση ο compiler, μας εμφάνιζε το μήνυμα 'firebase.firestore is undefined', ενώ είχαμε δηλώσει και το πακέτο που χρειαζόταν ως `import firebase/firestore`. Επίσης, το facebook login υλοποιήθηκε όπως και το google login, αλλά σύμφωνα με το facebook developers έπρεπε να παρέχουμε ένα url το οποίο να έχει την κατάληξη https και όχι http γι' αυτό και το κρύψαμε από τις επιλογές και εμφανίζεται μόνο το google login, καθώς για να τρέξει όλο το project θα έπρεπε να δημιουργούμε καινούργιο κανάλι κάθε φορά με το ngrok ο οποίος είναι πάροχος καναλιών έτσι ώστε το localhost:8000 να εμφανίζεται σαν `https://ngrok.io//dgj34382hbd`. Δηλαδή, να μας παρέχει ένα διαφορετικό url κάθε φορά, το οποίο να γίνεται δεκτό από το facebook developers. Ακόμη, για να παίρνουμε περισσότερα αποτελέσματα από το foursquare όπως φωτογραφίες, σχόλια χρηστών, φωτογραφίες στα venue cards στην αρχική σελίδα θα έπρεπε να είμαστε premium χρήστες στο foursquare, έτσι αυτό μας περιόρισε κάπως καθώς η εφαρμογή θα μπορούσε να είναι πολύ πιο εμφανίσιμη και χρηστική. Το ίδιο συμβαίνει και με το OpenWeatherMap API.

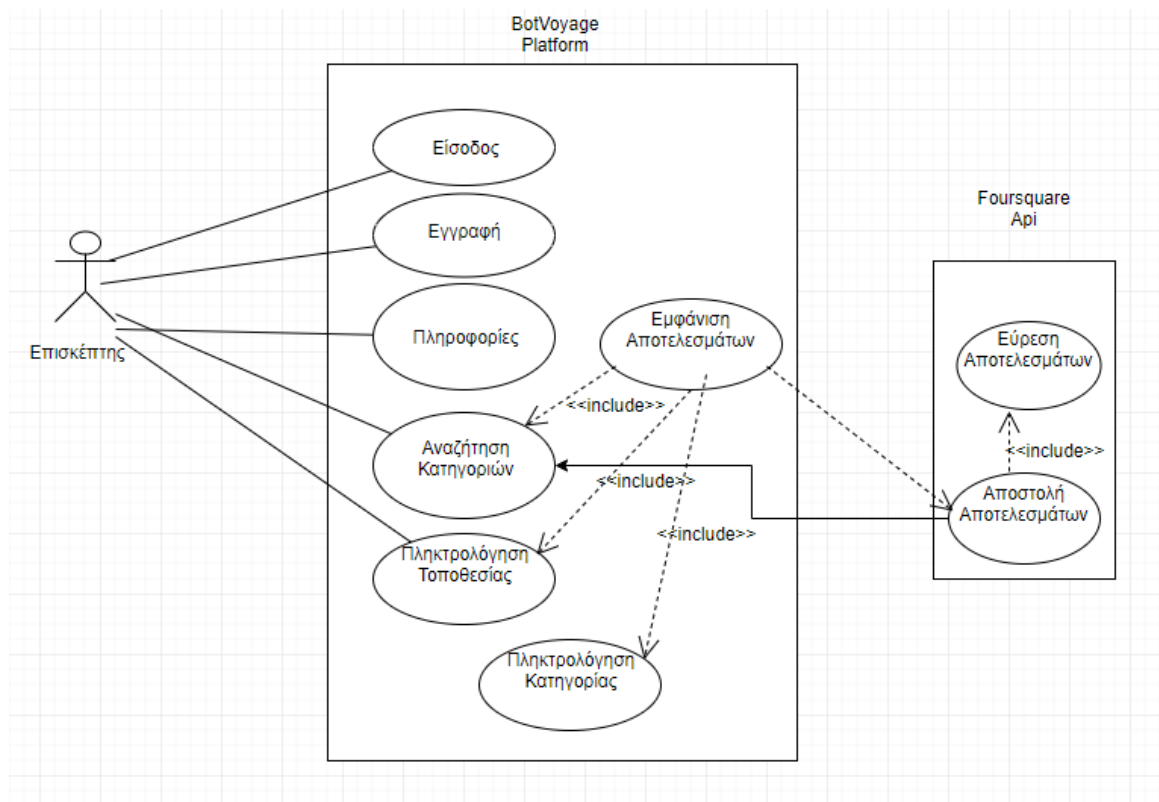
ΚΕΦΑΛΑΙΟ 5 - ΑΠΑΙΤΗΣΕΙΣ ΒΟΤVOYAGE - PLATFORM

5.1 Διαγράμματα

Τα παρακάτω διαγράμματα δημιουργήθηκαν με το λογισμικό draw.io, το οποίο είναι ελεύθερο για χρήση στην ομώνυμη σελίδα του.

5.1.1 Use Case Diagrams

5.1.1.1 Μελέτη Περίπτωσης Επισκέπτη Πλατφόρμας



Αρχικά, ο επισκέπτης έχει την δυνατότητα να δει τις σελίδες της εφαρμογής μας οι οποίες είναι η SIGNIN(είσοδος), η SIGNUP(εγγραφή), η ABOUTUS(πληροφορίες) και η SEARCH(αναζήτηση κατηγοριών). Έπειτα, πηγαίνοντας στην σελίδα search μπορεί να επιλέξει μέσω searchbar την τοποθεσία και την κατηγορία που επιθυμεί να αναζητήσει. Στη συνέχεια, οι προτιμήσεις κατηγορίας και τοποθεσίας στέλνονται σε μορφή text μέσω ενός request στο foursquare το οποίο θα επιστρέψει ένα response με τα αποτελέσματα στην σελίδα SEARCH η οποία είναι ορατή στον επισκέπτη της πλατφόρμας μας.

σελίδες ACCOUNT(προφίλ) και DETAILS(διαθέσιμες κατηγορίες) περιέχουν στοιχεία των χρηστών από την βάση δεδομένων. Στην πρώτη περίπτωση τα στοιχεία με τα οποία ο χρήστης έκανε LOGIN(είσοδος), έτσι ώστε να δημιουργηθεί το προφίλ του και στην δεύτερη περίπτωση τον τελευταίο χρήστη που πραγματοποίησε εγγραφή στην πλατφόρμας μας. Ακόμα, ο χρήστης μπορεί να επισκεφτεί την σελίδα BOT στην οποία στεγάζεται το webchat. Ο χρήστης κάνει αποστολή αρχικού μηνύματος μέσω του καναλιού WebChat. Στην συνέχεια, το bot ζητάει την τοποθεσία για την οποία επιθυμεί αποτελέσματα ο χρήστης μέσω ενός text message. Ωστόσο, για να γίνει η αποστολή αποτελεσμάτων μέσω συσχέτισης include, πρέπει πρώτα να γίνει η κατανόηση οντότητας μέσω της πλατφόρμας Wit.ai. Για παράδειγμα, αν ο χρήστης πληκτρολογήσει μια έκφραση τύπου “Άκουσα για ακραία καιρικά φαινόμενα”, θα μεταφραστεί σε καιρός και θα αποσταλεί. Οπότε θα είναι έτοιμο το bot να στείλει την προτροπή για πληκτρολόγηση τοποθεσίας. Επιπλέον, το bot για να πραγματοποιήσει την εμφάνιση των αποτελεσμάτων μέσω του WebChat στην σελίδα BOT της εφαρμογής, πρέπει να στείλει την τοποθεσία που πληκτρολόγησε ο χρήστης σε μορφή text στο foursquare καθώς και την κατηγορία αποτελεσμάτων που επιθυμεί. Αντίστοιχα για το OpenWeatherMap, πρέπει να στείλει ένα request για την τοποθεσία στην οποία επιθυμεί να δει τον καιρό. Σε επόμενο στάδιο, γίνεται μέσω των δύο αυτών APIs, η αποστολή των αποτελεσμάτων η οποία για να πραγματοποιηθεί, χρειάζεται πρώτα την εύρεση των αποτελεσμάτων όπως παρατηρούμε μέσω των συσχετίσεων include και στις δύο περιπτώσεις. Τέλος, συνδέεται με την αποστολή αποτελεσμάτων και με την εμφάνιση, οπότε και μέσω του καναλιού WebChat γίνεται η αναπαράσταση στην σελίδα BOT της πλατφόρμας μας.

5.2 Απαιτήσεις συστήματος

5.2.1 Σύνολο απαιτήσεων πλατφόρμας

1. Ο χρήστης πρέπει να είναι συνδεδεμένος στο διαδίκτυο.
2. Το σύστημα θα πρέπει να επιτρέπει στον επισκέπτη να αναζητήσει την πληροφορία που επιθυμεί.
3. Το σύστημα θα πρέπει να επιτρέπει την είσοδο στον χρήστη, μετά από αυθεντικοποίηση.

4. Το σύστημα θα πρέπει να επιτρέπει την είσοδο στον χρήστη, μετά από εξουσιοδότηση.
5. Ο χρήστης θα πρέπει να είναι συνδεδεμένος στην πλατφόρμα για να χρησιμοποιήσει το bot.
6. Το σύστημα θα πρέπει να προσφέρει την λειτουργία αλλαγής κωδικού πρόσβασης.
7. Το σύστημα θα πρέπει να συνδέεται στο Foursquare, έτσι ώστε να παρουσιάζει δεδομένα αναζήτησης.
8. Το σύστημα θα πρέπει να συνδέεται στο OpenWeatherMap, έτσι ώστε να παρουσιάζει δεδομένα αναζήτησης.
9. Το σύστημα θα πρέπει να επιτρέπει την εναλλαγή εικόνων στα γραφικά.
10. Το σύστημα θα πρέπει να διατηρεί ιστορικό αναζητήσεων.
11. Το σύστημα θα πρέπει να διαθέτει βάση δεδομένων στο firebase.
12. Το σύστημα θα πρέπει να επιτρέπει την είσοδο μέσω social login (google login) στον χρήστη.
13. Το σύστημα θα πρέπει να παρουσιάζει κατηγορίες αναζήτησης στον χρήστη.
14. Το σύστημα θα πρέπει να αποθηκεύει δεδομένα στο firebase.
15. Το σύστημα θα πρέπει να συνδέεται στο firebase και να παρουσιάζει δεδομένα καταχώρησης.
16. Το σύστημα θα πρέπει να παρουσιάζει δεδομένα των διαχειριστών.
17. Το σύστημα θα πρέπει να παρέχει ανακατεύθυνση σελίδων.
18. Το σύστημα θα πρέπει να παρέχει εκτέλεση ενεργειών μέσω γραφικών
19. Το σύστημα θα πρέπει να διατηρεί λογαριασμό για τον συνδεδεμένο χρήστη
20. Το σύστημα θα πρέπει να παρέχει στον χρήστη την δυνατότητα αλλαγής του ονόματός του.
21. Το σύστημα θα πρέπει να παρουσιάζει τα δεδομένα σύνδεσης στον χρήστη εκτός από το password του.
22. Το σύστημα θα πρέπει να παρέχει εναλλαγές εικόνων.

23. Το σύστημα θα πρέπει να παρουσιάζει τις επικρατέστερες αναζητήσεις σε carousel.
24. Το σύστημα θα πρέπει να παρέχει μέθοδο αποσύνδεσης.
25. Ο χρήστης θα πρέπει να χρησιμοποιεί τα σωστά στοιχεία σύνδεσης.
26. Ο χρήστης θα πρέπει να είναι εξουσιοδοτημένος προκειμένου να δει τις επικρατέστερες αναζητήσεις.
27. Ο χρήστης θα πρέπει να είναι εξουσιοδοτημένος προκειμένου να επισκεφτεί το προφίλ του.
28. Το σύστημα θα πρέπει να ανανεώνει στο firebase τους χρήστες.
29. Το σύστημα θα πρέπει να επιτρέπει την αλλαγή εικόνας προφίλ στον χρήστη.
30. Το σύστημα θα πρέπει να ανανεώνει την εικόνα προφίλ του χρήστη σε περίπτωση σύνδεση με google login.
31. Ο χρήστης θα πρέπει να διαθέτει λογαριασμό google αν επιθυμεί την γρηγορότερη πρόσβασή του στην πλατφόρμα.
32. Το σύστημα θα πρέπει να παίρνει δεδομένα από το google όπως όνομα, επώνυμο.
33. Το σύστημα θα πρέπει να διαθέτει ιστορικό εγγραφών και συνδέσεων των χρηστών.
34. Το σύστημα θα πρέπει να διαθέτει ιστορικό μεθόδου σύνδεσης.
35. Το σύστημα θα πρέπει να περιέχει περιγραφή των σελίδων που εναλλάσσει ο χρήστης.
36. Το σύστημα θα πρέπει να ενημερώνει με μηνύματα για επιτυχείς ενέργειες.
37. Το σύστημα θα πρέπει να αλλάζει τα url στον browser σε περίπτωση αλλαγής της σελίδας.
38. Το σύστημα θα πρέπει να επιτρέπει την είσοδο χρηστών που έχουν δημιουργηθεί από την βάση.
39. Το σύστημα θα πρέπει να ενημερώνεται αυτόματα από την βάση.
40. Το σύστημα θα πρέπει να επιτρέπει την ανταλλαγή μηνυμάτων μεταξύ μπουτ και χρήστη.

41. Το σύστημα θα πρέπει παρουσιάζει το μποτ στον συνδεδεμένο χρήστη.
42. Ο χρήστης θα πρέπει να επιλέγει μια ενέργεια κάθε φορά.
43. Το σύστημα θα πρέπει να επικοινωνεί με το webchat στο azure για την αναπαράσταση του bot.
44. Το σύστημα θα πρέπει να ενημερώνει τυχόν αλλαγές στα αρχεία json της βάσης, από τους διαχειριστές.
45. Το σύστημα θα πρέπει να ενημερώνει τον χρήστη σε περίπτωση λάθους ενέργειας.
46. Το σύστημα θα πρέπει να αναγνωρίζει την τοποθεσία που επιλέγει ο χρήστης.
47. Το σύστημα θα πρέπει να παρέχει στον χρήστη μία κατανοητή ροή, μέσα από το γραφικό του περιβάλλον.
48. Το σύστημα θα πρέπει να ενημερώνει την γλώσσα, εάν ο χρήστης το ζητήσει.
49. Ο χρήστης θα πρέπει να έχει τη δυνατότητα να δει εξ αρχής όλες τις λειτουργικές επιλογές της εφαρμογής.
50. Ο χρήστης θα πρέπει να επιλέγει μία δυνατότητα κάθε φορά.
51. Το σύστημα θα πρέπει να στέλνει την τοποθεσία που επέλεξε ο χρήστης στο Foursquare.
52. Το σύστημα θα πρέπει να στέλνει την κατηγορία αναζήτησης του χρήστη στο Foursquare.
53. Το σύστημα θα πρέπει να παρέχει τα δεδομένα σε μορφή json στην βάση.
54. Το σύστημα θα πρέπει να διατηρεί collections.
55. Το σύστημα θα πρέπει να διατηρεί διαφορετική βάση για τους διαχειριστές.
56. Οι διαχειριστές θα πρέπει να μπορούν να δημιουργήσουν χρήστες από το admin dashboard.
57. Οι διαχειριστές θα πρέπει να μπορούν να διαγράψουν κάποιον χρήστη από το admin dashboard.
58. Οι διαχειριστές θα πρέπει να μπορούν να εξάγουν τα δεδομένα σε αρχεία από το admin dashboard.

59. Το σύστημα θα πρέπει να επιτρέπει την δημιουργία posts για τους users.
60. Το σύστημα θα πρέπει να ανακατευθύνει τον χρήστη στην προηγούμενη σελίδα στον browser.
61. Το σύστημα θα πρέπει να ανακατευθύνει τον χρήστη στην επόμενη σελίδα στον browser.
62. Το σύστημα θα πρέπει να πλαισιώνει τους χρήστες στο admin dashboard.
63. Το σύστημα θα πρέπει να παράγει σε μορφή hash το id του χρήστη στην βάση.
64. Το σύστημα θα πρέπει να παράγει σε μορφή hash το id του χρήστη στο admin dashboard.
65. Το σύστημα θα πρέπει να παρέχει ασφάλεια στα στοιχεία των χρηστών στην βάση.
66. Θα πρέπει να υπάρχει δυνατότητα στον χρήστη, για επανεκκίνηση του διαλόγου και των στοιχείων του σε περίπτωση λάθους.
67. Το σύστημα θα πρέπει να ανακατευθύνει τον χρήστη στους ιστοτόπους για πραγματοποίηση συναλλαγών και ενημέρωσης.
68. Το σύστημα θα πρέπει να διαμορφώνει ένα profile για τον κάθε χρήστη για τη διευκόλυνση της αλληλεπίδρασής του.
69. Το σύστημα θα πρέπει να παρέχει στον χρήστη μία κατανοητή ροή, μέσα από το γραφικό του περιβάλλον.
70. Το σύστημα θα πρέπει να κατανοεί τη φυσική γλώσσα του χρήστη.
71. Το σύστημα θα πρέπει να συνδέεται σε ένα από τα api, και να επιστρέφει τα δεδομένα που ζητήθηκαν από τον χρήστη.
72. Το σύστημα θα πρέπει ανά διαστήματα να ενημερώνει τον χρήστη για γεγονότα με βάση το ιστορικό του profile του.
73. Το σύστημά μας θα πρέπει να αποκρίνεται το συντομότερο δυνατό, γεγονός που κρίνεται από την επιλογή διακομιστή.
74. Ο διακομιστής (server) μας, θα ανήκει στο Azure της Microsoft.
75. Η διασύνδεση χρήστη θα πρέπει να υλοποιηθεί μέσω της JavaScript, σε Node.js περιβάλλον με χρήση της βιβλιοθήκης ReactJS.

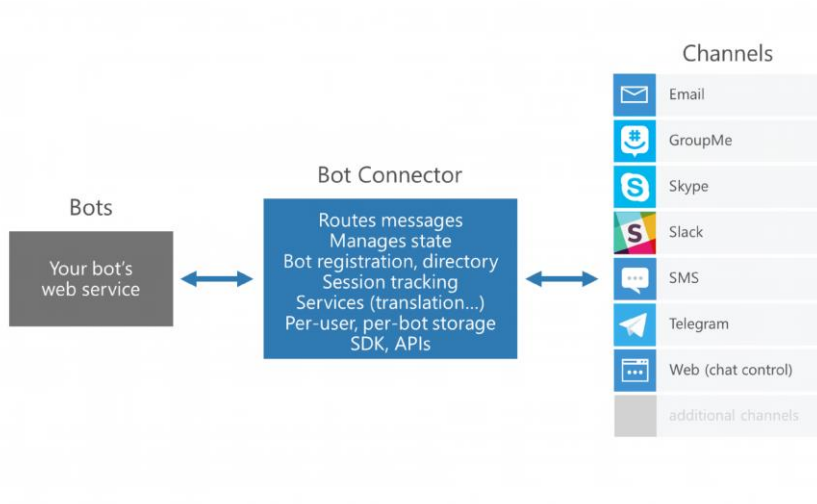
76. Το ιστορικό του διαλόγου του χρήστη θα πρέπει να αποθηκεύεται με ασφάλεια, στη βάση δεδομένων μας.
77. Τα δεδομένα που θα παραχωρούνται στον χρήστη, θα πρέπει να έχουν προέλευση από αξιόπιστες πηγές.
78. Το σύστημα θα πρέπει να ενημερώνει τον χρήστη σε περιπτώσεις λάθους, ώστε να υπάρξει η κατάλληλη χρηστικότητα.
79. Το σύστημα δε θα πρέπει να παραβιάζει το privacy policy των api, όσον αφορά την κατακράτηση δεδομένων.
80. Το σύστημα θα πρέπει να ακολουθεί τη λογική ροή ήδη υπαρχόντων bot, ώστε να είναι πιο εύκολη η εμπειρία χρήσης.
81. Το σύστημα θα δίνει τη δυνατότητα στον χρήστη, να παρακολουθήσει μία παραδειγματική ροή της εφαρμογής.
82. Το σύστημα δε θα περιλαμβάνει κάποιο χρονικό περιορισμό όσον αφορά την είσοδο του χρήστη.

ΚΕΦΑΛΑΙΟ 6 - ΕΝΣΩΜΑΤΩΣΗ ΤΟΥ ΒΟΤ ΣΤΗΝ ΠΛΑΤΦΟΡΜΑ ΚΑΝΟΝΤΑΣ ΧΡΗΣΗ ΤΟΥ ΚΑΝΑΛΙΟΥ WEBCHAT

6.1 Τι είναι το Bot Framework

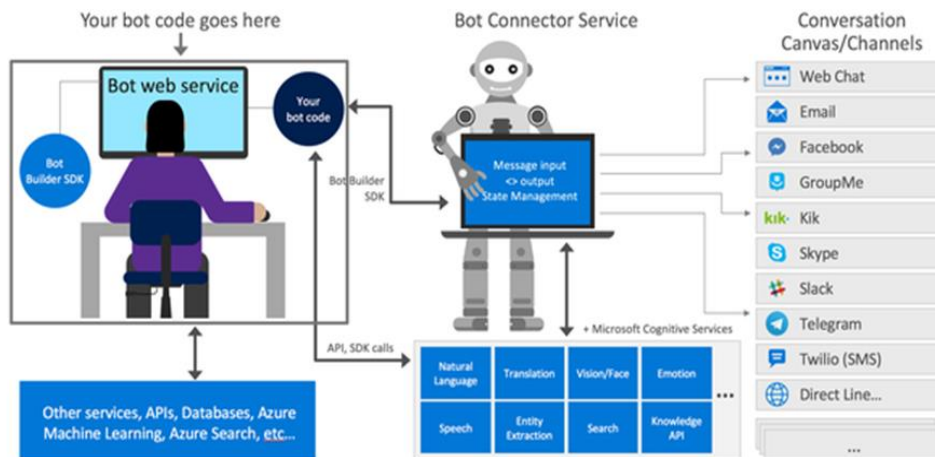
Το Microsoft Bot Framework παρέχει ακριβώς αυτό που χρειαζόμαστε για να δημιουργήσουμε και να συνδέσουμε έξυπνα bots που αλληλεπιδρούν φυσικά με την ύπαρξη ομιλίας χρηστών, από απλά μηνύματα /sms μέχρι τις πιο διάσημες διαδικτυακές εφαρμογές ομιλίας (Skype, Messenger, Viber κλπ). Για τη δημιουργία ενός ολοκληρωμένου συστήματος υπάρχουν τρία κομμάτια που συμπληρώνουν ουσιαστικά το παζλ. Το bot , ο bot connector , και τα κανάλια. Τα bots επιτρέπουν στους χρήστες να αλληλεπιδρούν με έξυπνες λύσεις σαν να συνομιλούν με κάποιο άλλο φυσικό πρόσωπο και οι αλληλεπιδράσεις αυτές μπορούν να πάρουν πολλές μορφές. Το Bot Framework παρέχει στους προγραμματιστές μία «πύλη» και ένα σύστημα ανάπτυξης λογισμικού (SDK) για να δημιουργήσει bots και έναν bot connector για να μπορέσει να συνδεθεί σε διάφορα κανάλια όπως είναι το Twitter και το Slack, όπως επίσης και ένα σημειωματάριο Bot για να μπορέσει να δει τι bot υπάρχουν και ποια από αυτά μπορεί να δοκιμάσει.

6.2 Bot Connector



Εικόνα 84 - Bot Connector - Bot Framework -
https://blogs.msdn.microsoft.com/uk_faculty_connection/2016/04/05/what-is-microsoft-bot-framework-overview/

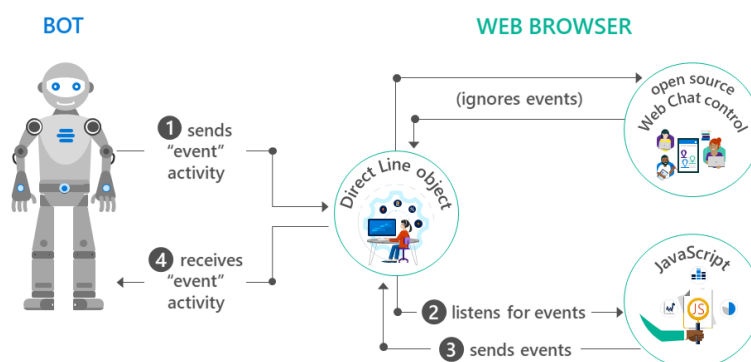
Στην παραπάνω απεικόνιση βλέπουμε τη διαδικασία που μας παρέχει το Bot Framework. Ο Bot Connector μας επιτρέπει να συνδέσουμε τα bot μας χωρίς προβλήματα στα διαθέσιμα κανάλια που μπορούμε να επιλέξουμε. Εμείς σε αυτήν την περίπτωση επιλέγουμε το Web Chat και το DirectLine όπως φαίνεται στην παραπάνω εικόνα καθώς είναι και τα δύο κανάλια που χρησιμοποιήθηκαν στην πλατφόρμα μας για την σύνδεση με το μποτ.



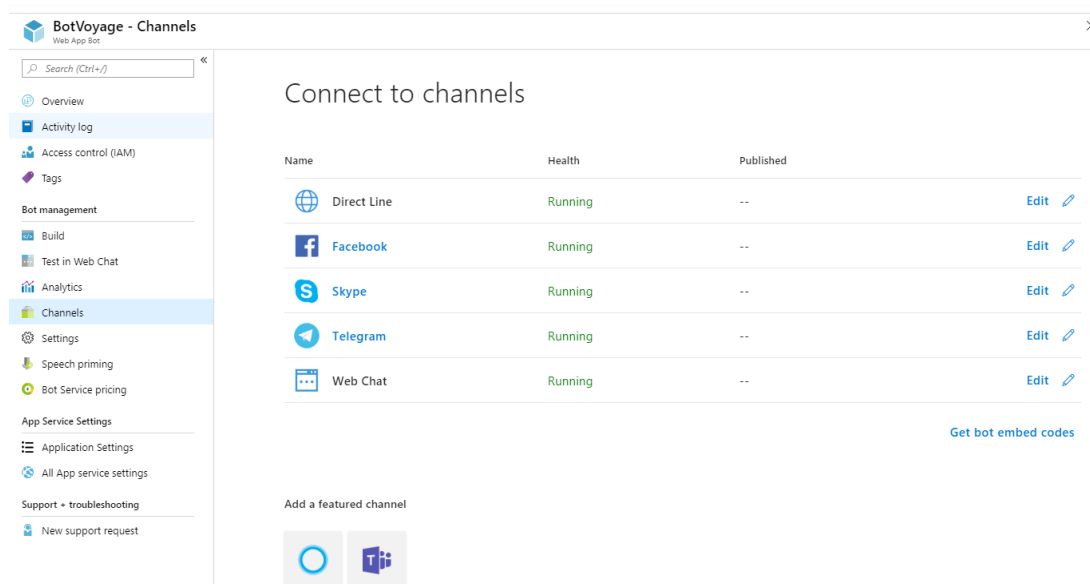
Εικόνα 85 - Bot Connector Service - Bot Framework - https://medium.com/@ashish_fagna/introduction-to-microsoft-azure-bot-service-luis-language-understanding-8826d29d013e

6.3 Channels

Παρακάτω, βλέπουμε τα channels (κανάλια) τα οποία προστέθηκαν στο Web App Bot BotVoyage στο Azure portal. Κάποια από αυτά βρίσκονται υπό δοκιμή, όπως για παράδειγμα το Telegram channel. Για να χρησιμοποιήσουμε κάποιο αυτά τα κανάλια σε ένα dev περιβάλλον, απλά προσθέτουμε το 'key' του καναλιού στην εν λόγω μέθοδο ή λειτουργία που θα χρησιμοποιηθεί. Για να δώσουμε ένα παράδειγμα, το Direct Line κανάλι χρησιμοποιείται για να ενεργοποιήσει την επικοινωνία μεταξύ ενός bot και μιας εφαρμογής (ή αλλιώς client).



Εικόνα 86 - Direct Line - Bot Framework - <https://docs.microsoft.com/en-us/azure/bot-service/nodejs/bot-builder-nodejs-backchannel?view=azure-bot-service-3.0>

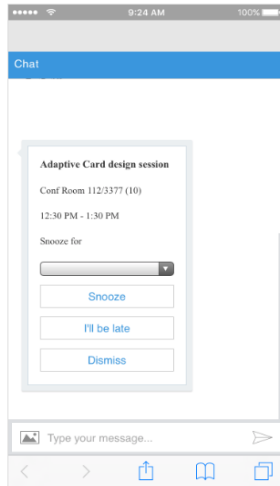


Εικόνα 87 - BotVoyage Channels - Azure Portal - (δικό μας)

6.3.1 Web Chat

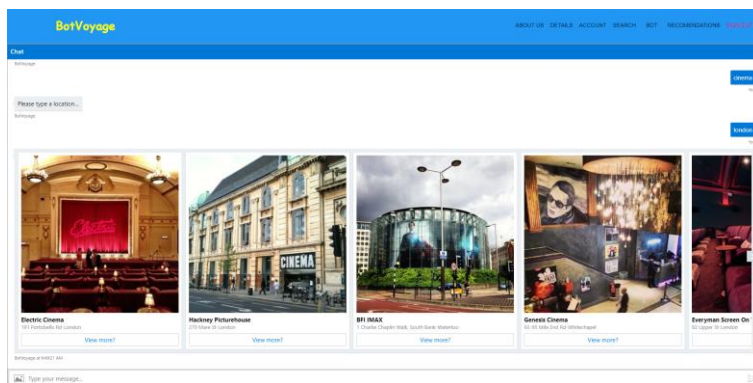
Το Webchat χρησιμοποιείται από τους προγραμματιστές, ώστε να μπορέσουν να προσθέσουν ένα διαδικτυακό chat στην ιστοσελίδα τους. Το web chat που παρέχεται από το bot framework δίνει την δυνατότητα στον χρήστη να συνομιλήσει με ένα bot. Κατά την δημιουργία ενός bot, το οποίο έχει Bot Service, μας δημιουργείται αυτόματα ένα Web Chat κανάλι. Αυτό το κανάλι, μας παρέχει web chat control, στοιχείο το οποίο δίνει την δυνατότητα στους χρήστες, να αλληλεπιδρούν με το bot απευθείας μέσα από μια ιστοσελίδα. Πιο συγκεκριμένα, ένας χρήστης χρειάζεται το secret key του Web Chat καναλιού, το οποίο πρέπει να είναι προστατευμένο, έπειτα το ανταλλάσσει με κάποιο token και έτσι μπορεί να επικοινωνεί με το bot σε ένα web application.

Παράδειγμα εμφώλευσης Web Chat σε mobile περιβάλλον:



Εικόνα 88 - Web Chat - Bot framework - <https://docs.microsoft.com/en-us/azure/bot-service/?view=azure-bot-service-4.0>

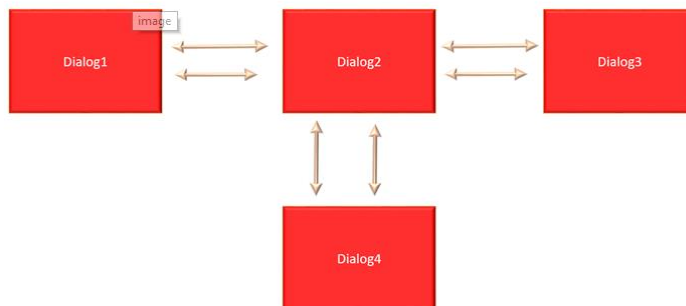
Χρήση Web Chat στο BotVoyagePlatform



Εικόνα 89 - Web Chat - BotVoyage Platform - (δικό μας)

- 1) Ο χρήστης ξεκινά την εφαρμογή.
- 2) Ο Azure Active Directory επαληθεύει την ταυτότητα του χρήστη.
- 3) Ο χρήστης μπορεί να ρωτήσει τι είδους ερωτήματα μπορούν να απαντηθούν.
- 4) Η υπηρεσία επιστρέφει ένα FAQ το οποίο έχει δημιουργηθεί με QnA.
- 5) Ο χρήστης ρωτά έγκυρο ερώτημα.
- 6) Το bot στέλνει ερώτημα στο Azure Search, το οποίο επιστρέφει πληροφορίες σχετικά με τα δεδομένα της εφαρμογής.
- 7) Οι πληροφορίες σχετικά με την εφαρμογή συγκεντρώνονται με χρόνου εκτέλεσης για να βοηθήσουν στην ανάπτυξη για την καλύτερη απόδοση και τη χρήση του bot.

Η έννοια ενός Dialog, είναι πολύ απλά το σύνολο των ερωτήσεων που παρέχει και των απαντήσεων που παρέχονται από και προς τον χρήστη αντίστοιχα. Παρακάτω φαίνεται η αλληλεπίδραση μεταξύ διαφόρων dialog.



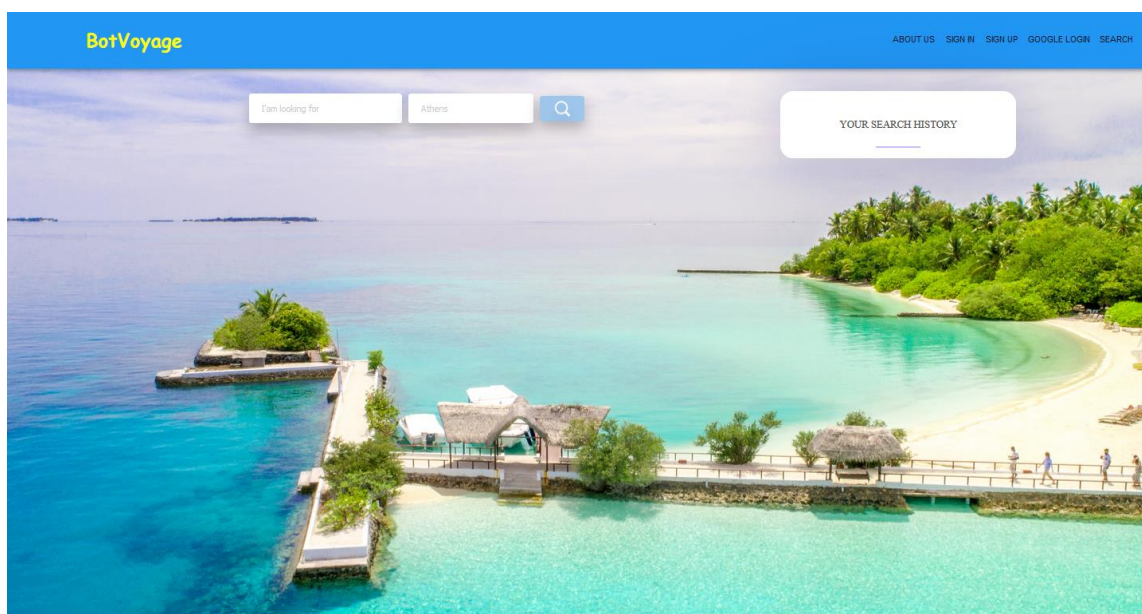
Εικόνα 92 - Dialogs - Bot Framework -

https://blogs.msdn.microsoft.com/uk_faculty_connection/2016/04/05/what-is-microsoft-bot-framework-overview/

ΚΕΦΑΛΑΙΟ 7 - ΠΑΡΟΥΣΙΑΣΗ ΛΕΙΤΟΥΡΓΙΩΝ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ

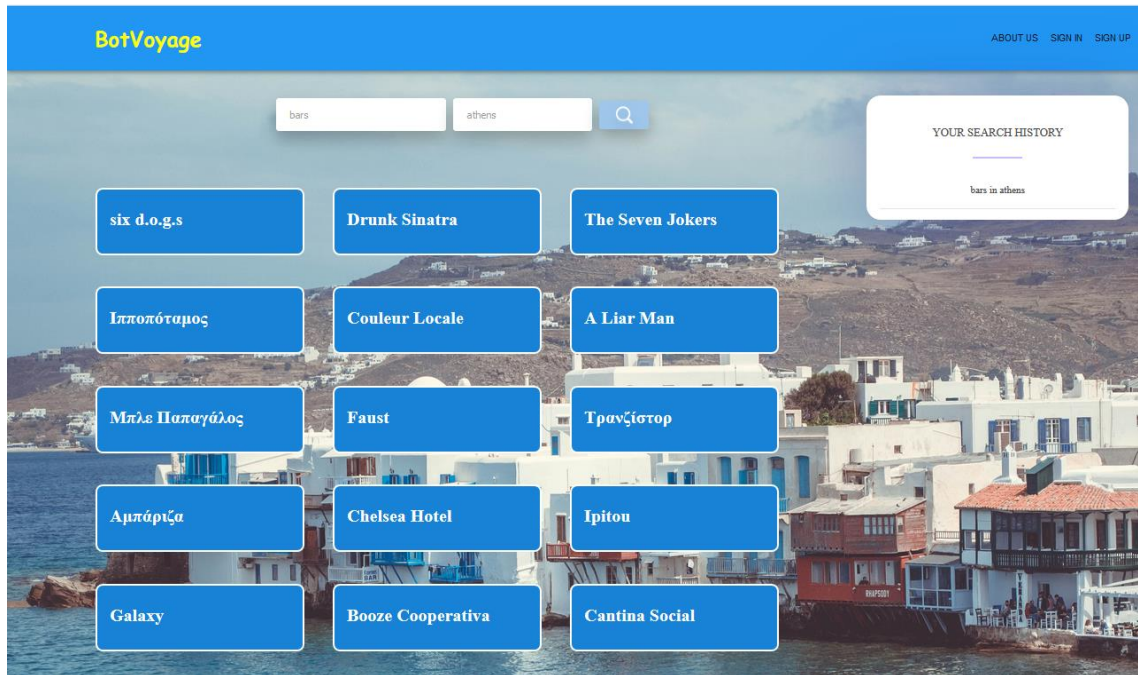
7.1 Αρχική σελίδα

Αρχικά ο χρήστης εισέρχεται στην πλατφόρμα και το πρώτο πράγμα που βλέπει, είναι η σελίδα που έχουμε ορίσει ως αρχική σελίδα στην εφαρμογή μας και χωρίς να έχει προηγηθεί η διαδικασία της αυθεντικοποίησης. Δηλαδή, η σελίδα αυτή είναι ορατή για όλους τους χρήστες που δεν έχουν κάνει ακόμα για την συγκεκριμένη στιγμή που μιλάμε κάποιο sign in. Στην σελίδα αυτή επίσης, οι χρήστες μπορούν να δουν το ιστορικό των αναζητήσεων που αυτοί πραγματοποιούν.



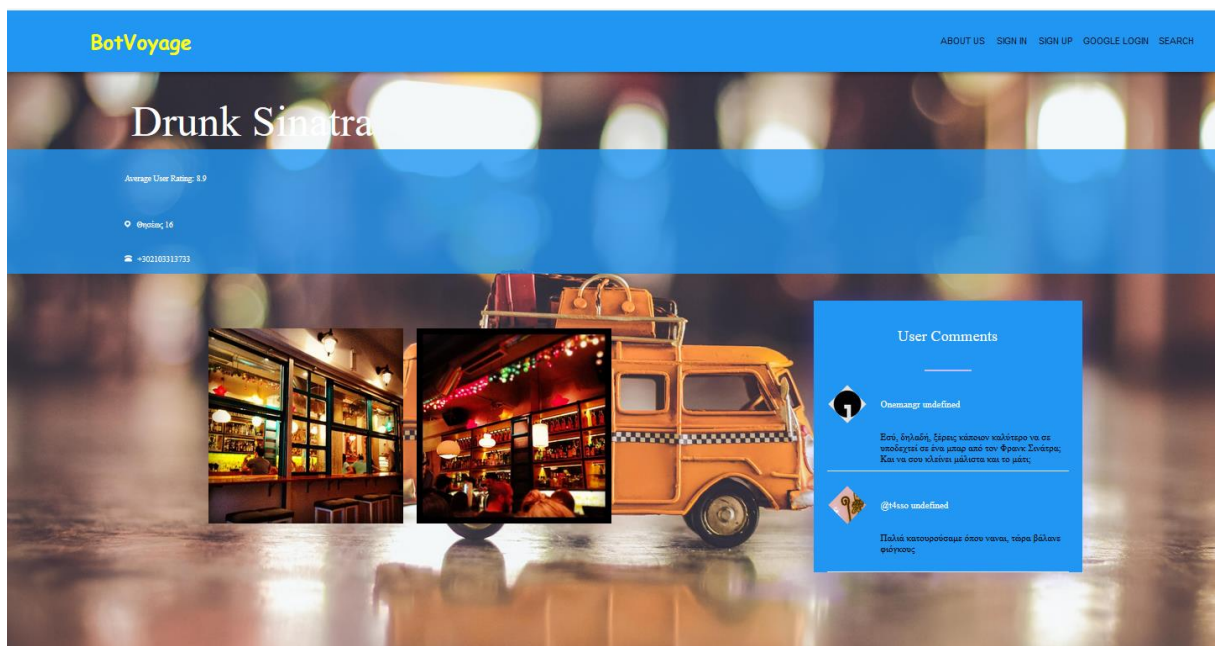
Εικόνα 93 - Αρχική Σελίδα - BotVoyage Platform - (δικό μας)

Έπειτα ας πούμε ότι ο χρήστης επιθυμεί να αναζητήσει την κατηγορία bars στην Αθήνα, οπότε και τα πληκτρολογεί στο searchbar όπως φαίνεται και στην εικόνα. Το αποτέλεσμα είναι να του εμφανίσει μέρη τα οποία έψαξε στο foursquare μέσω ενός request και ανάκτησε μέσω ενός response.



Εικόνα 94 - Αναζήτηση Bars στην Αθήνα - BotVoyage Platform - (δικό μας)

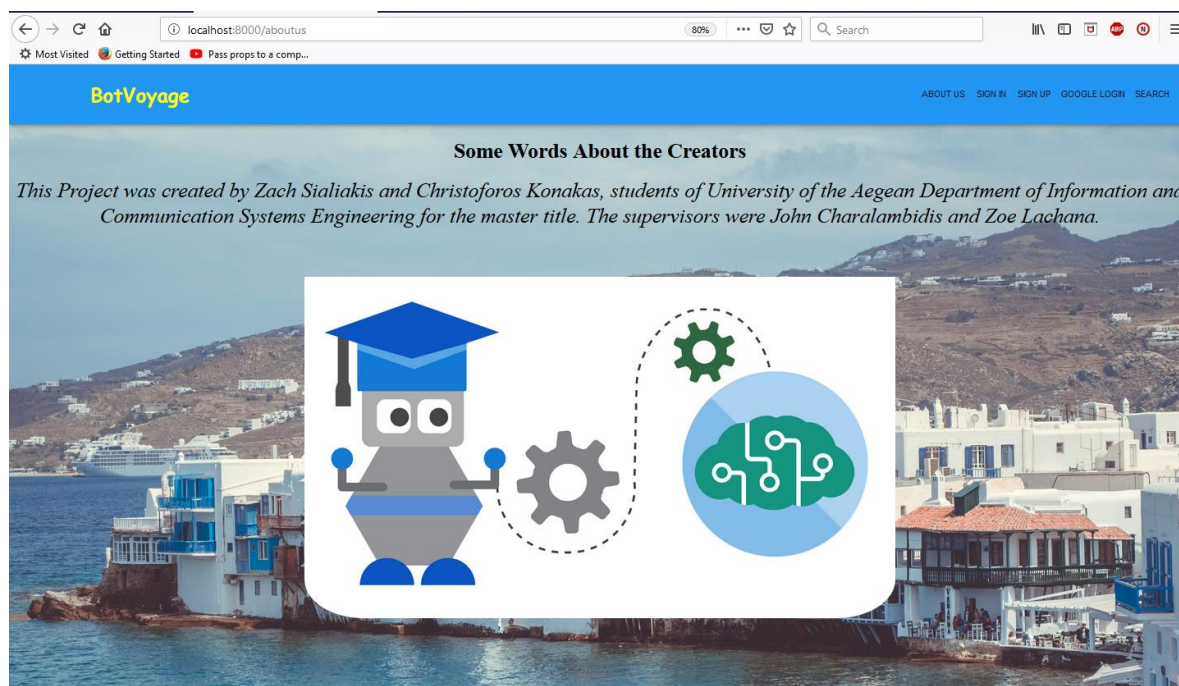
Κάνοντας κλικ στο μέρος που επιθυμεί ή του κέντρισε το ενδιαφέρον η σελίδα αλλάζει και του παρουσιάζεται μια καινούργια σελίδα με σχόλια από χρήστες του Foursquare, βαθμολογία για το εν λόγω μαγαζί από χρήστες οι οποίοι το έχουν επισκεφτεί, οδός που βρίσκεται και τηλέφωνο επιχείρησης όπως φαίνεται παρακάτω.



Εικόνα 95 - Drunk Sinatra - BotVoyage Platform - (δικό μας)

7.2 Σελίδα AboutUs

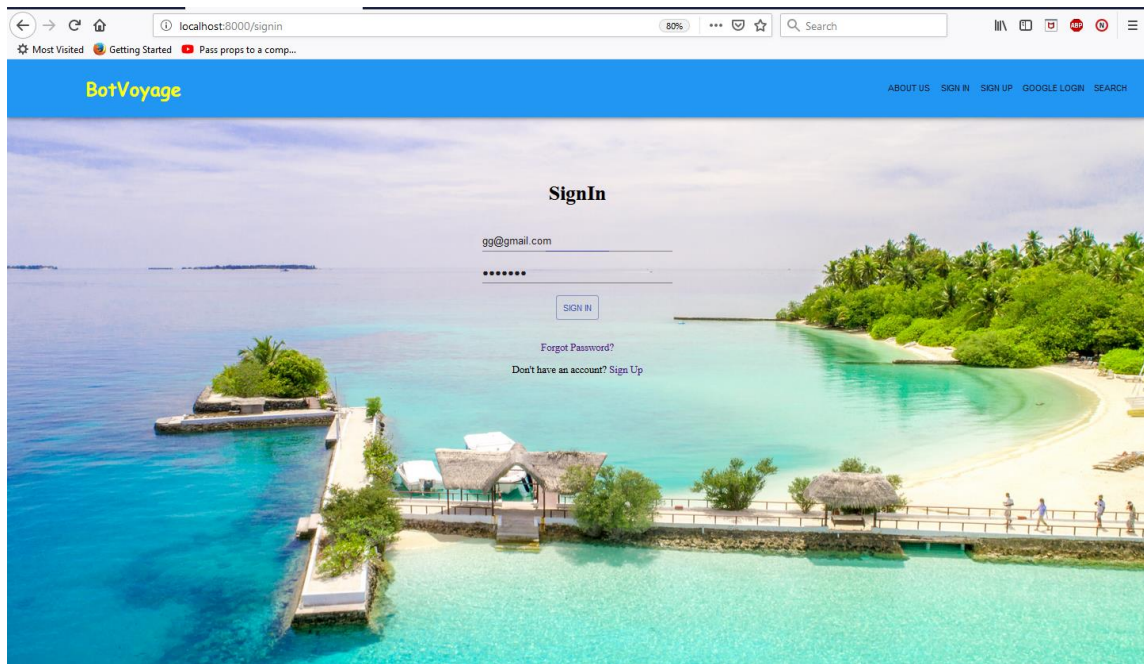
Όταν ο χρήστης επιλέξει την σελίδα About Us (στο Menubar) μεταφέρεται σε μια καινούργια σελίδα η οποία περιέχει πληροφορίες για τους δημιουργούς της εφαρμογής και τους επιβλέποντες των φοιτητών που την πραγματοποίησαν, καθώς και μια εικόνα με ένα bot για να παρέχουμε ένα πιο ευχάριστο περιβάλλον στον επισκέπτη της πλατφόρμας μας.



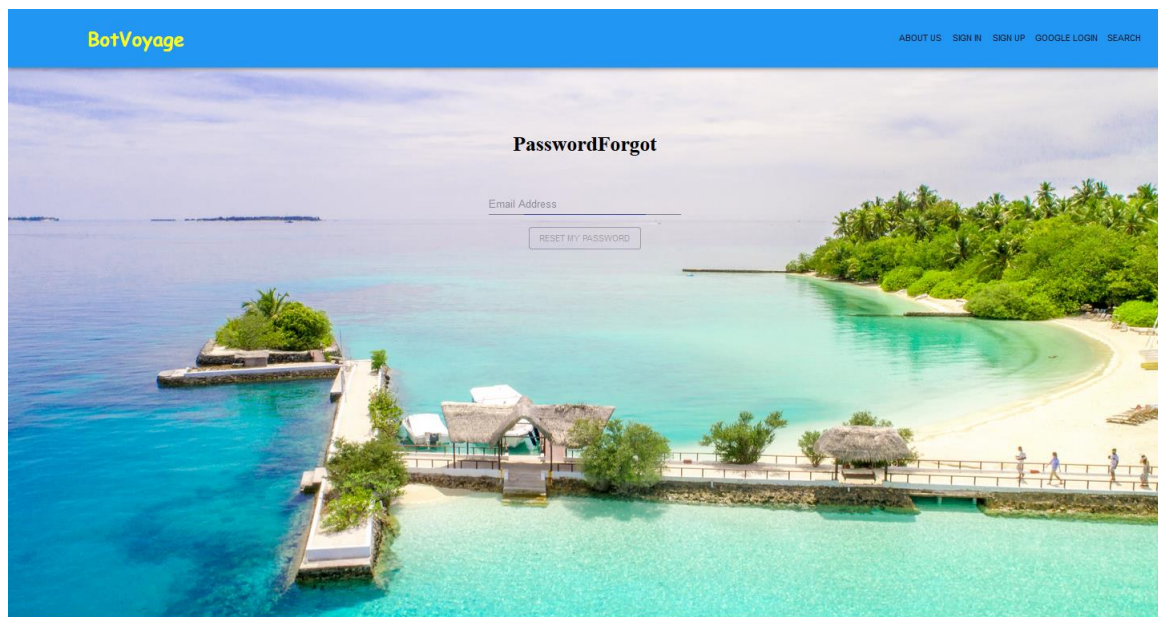
Εικόνα 96 - Σελίδα AboutUs - BotVoyage Platform - (δικό μας)

7.3 Σελίδα sign_in

Πατώντας το κουμπί SIGN IN ο χρήστης μεταφέρεται στην σελίδα της πλατφόρμας μας στην οποία μπορεί να συμπληρώσει τα στοιχεία του έτσι ώστε να εισέλθει αυθεντικοποιημένος στην πλατφόρμα μας. Η σελίδα αυτή επίσης, παρέχει ένα κουμπί forgot password το οποίο σε περίπτωση ενεργοποίησης θα μεταφέρει τον χρήστη σε μια φόρμα συμπλήρωσης email έτσι ώστε να πληκτρολογήσει το email του και να του σταλεί ένα αυτοματοποιημένο μήνυμα για αλλαγή κωδικού πρόσβασης στο email του.



Εικόνα 97 - Σελίδα Sign In - BotVoyage Platform - (δικό μας)

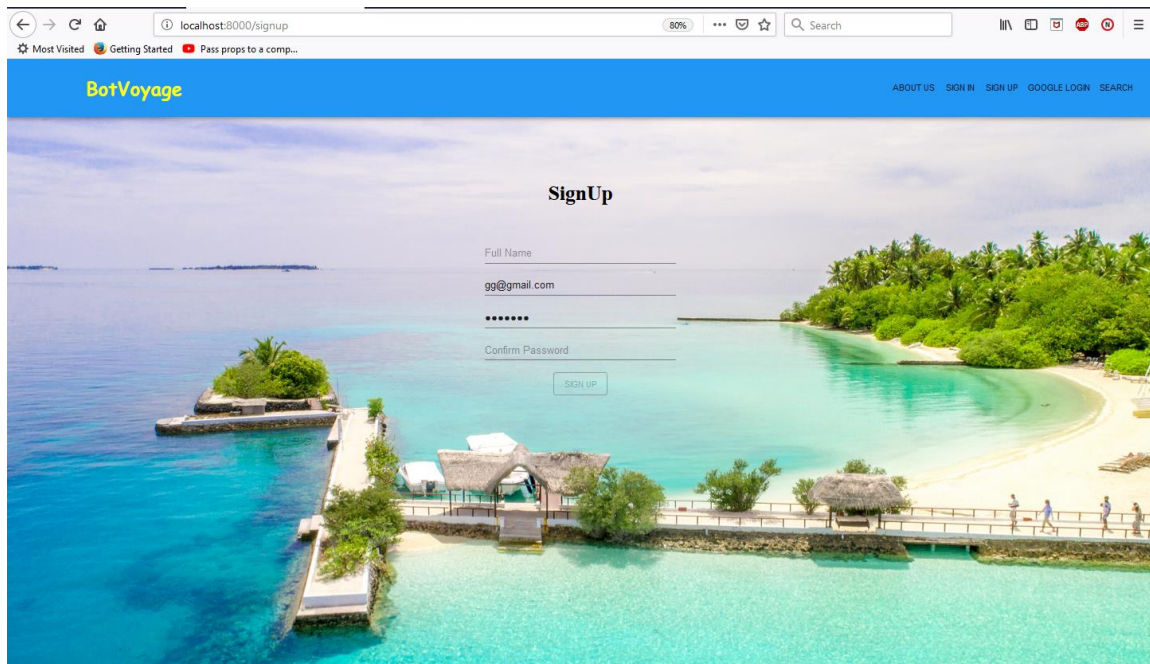


Εικόνα 98 - Σελίδα PasswordForgot - BotVoyage Platform - (δικό μας)

Επίσης υπάρχει το κουμπί sign up με την περιγραφή 'Don't have an account? Sign up' το οποίο όταν πατηθεί από τον χρήστη αυτός θα μεταφερθεί στην σελίδα sign up, για εγγραφή στην πλατφόρμα μας.

7.4 Σελίδα SIGN UP

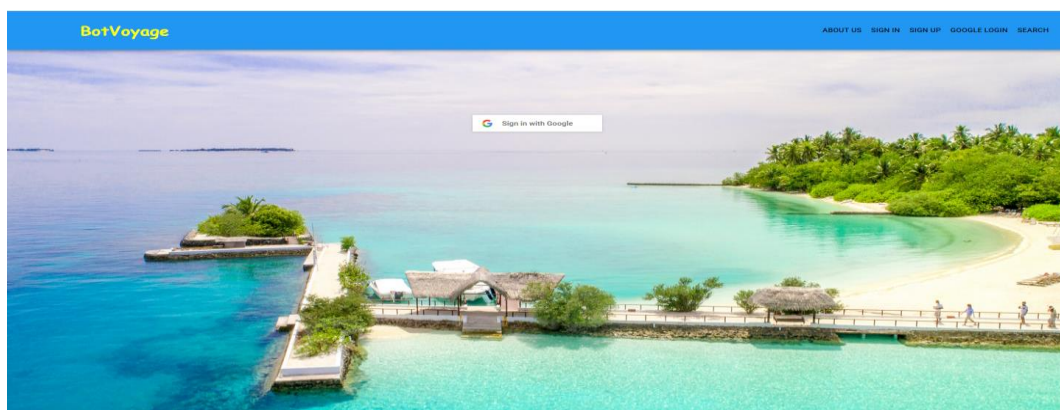
Στη συνέχεια, για την μεταφορά του χρήστη στην σελίδα εγγραφής, υπάρχει κουμπί και στο κεντρικό menu της εφαρμογής μας. Πατώντας το, ο χρήστης, όπως αναφέραμε πηγαίνει στην σελίδα sign_up, στην οποία μπορεί να συμπληρώσει τα στοιχεία για εγγραφή στην σελίδα μας.



Εικόνα 99 - Σελίδα SignUp - BotVoyage Platform - (δικό μας)

7.5 Σελίδα GOOGLE LOGIN

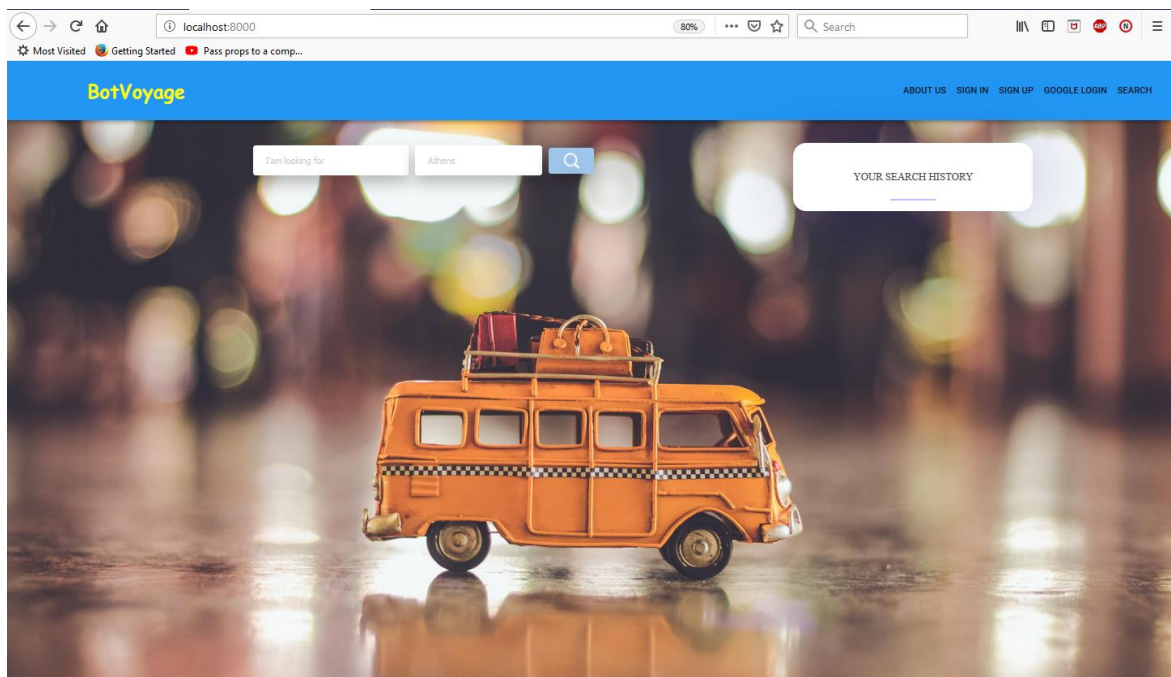
Σε περίπτωση που κάποιος επισκέπτης δεν επιθυμεί να χάσει χρόνο για εγγραφή και είσοδο με την συμπλήρωση στοιχείων υπάρχει η δυνατότητα της εισόδου του στην πλατφόρμα με google login.



Εικόνα 100 - Σελίδα GoogleLogin - BotVoyage Platform - (δικό μας)

7.6 Σελίδα Search

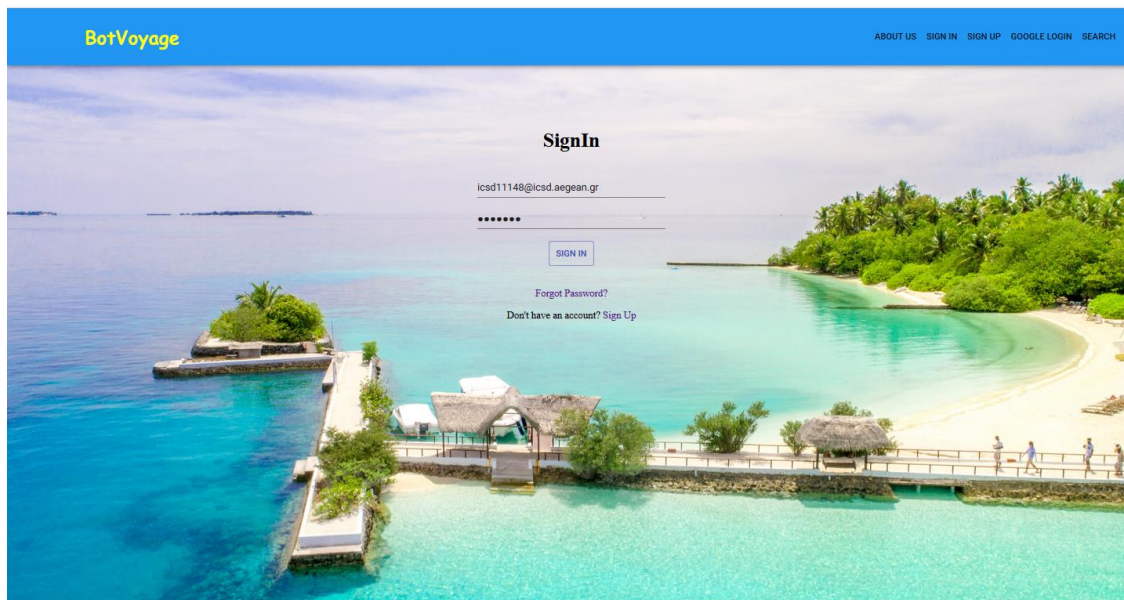
Το κουμπί SEARCH στο μενού της εφαρμογής μας έχει προστεθεί απλά για μεταφορά του χρήστη στην αρχική σελίδα της πλατφόρμας μας, όπως κάνουν οι περισσότερες σελίδες απλά με το κλικ στο logo τους. Όπως παρατηρείτε έχει προστεθεί η λειτουργία της εναλλαγής εικόνων μετά από 10 δευτερόλεπτα έτσι ώστε να προσφέρουμε ένα πιο ευχάριστο περιβάλλον γραφικό στον επισκέπτη μας. Εξάλλου η ικανοποίηση του χρήστη από την χρήση της πλατφόρμας είναι και το κύριο μέλημά μας.



Εικόνα 101 - Σελίδα Search - BotVoyage Platform - (δικό μας)

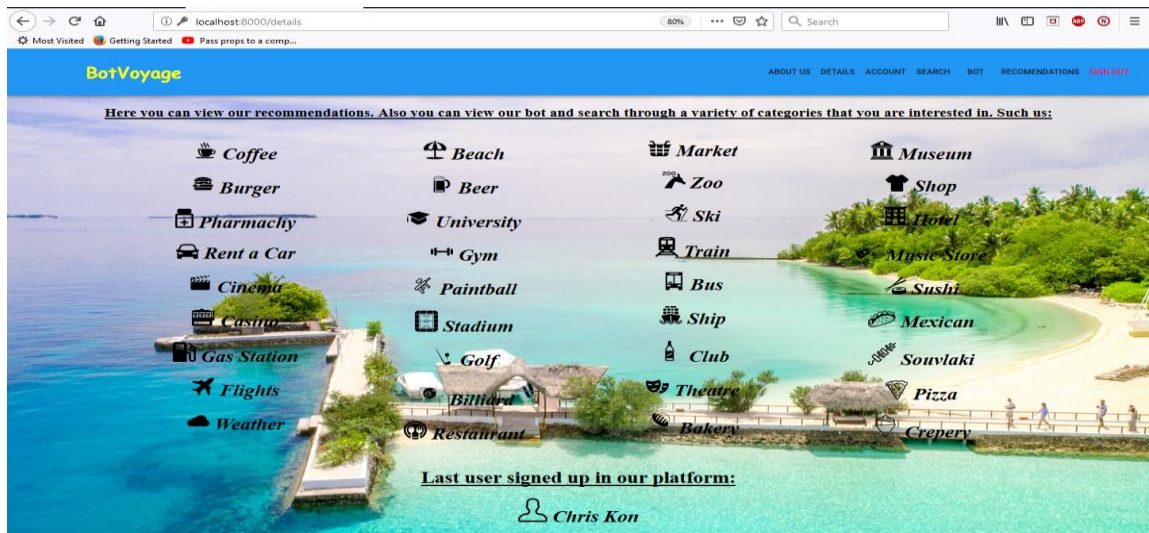
7.7 Είσοδος του χρήστη στην πλατφόρμα

Αφότου ο χρήστης πάει στην σελίδα sign_in και συμπληρώσει τα στοιχεία για την είσοδό του όπως παρακάτω,



Εικόνα 102 - Σελίδα Sign In - BotVoyage Platform - (δικό μας)

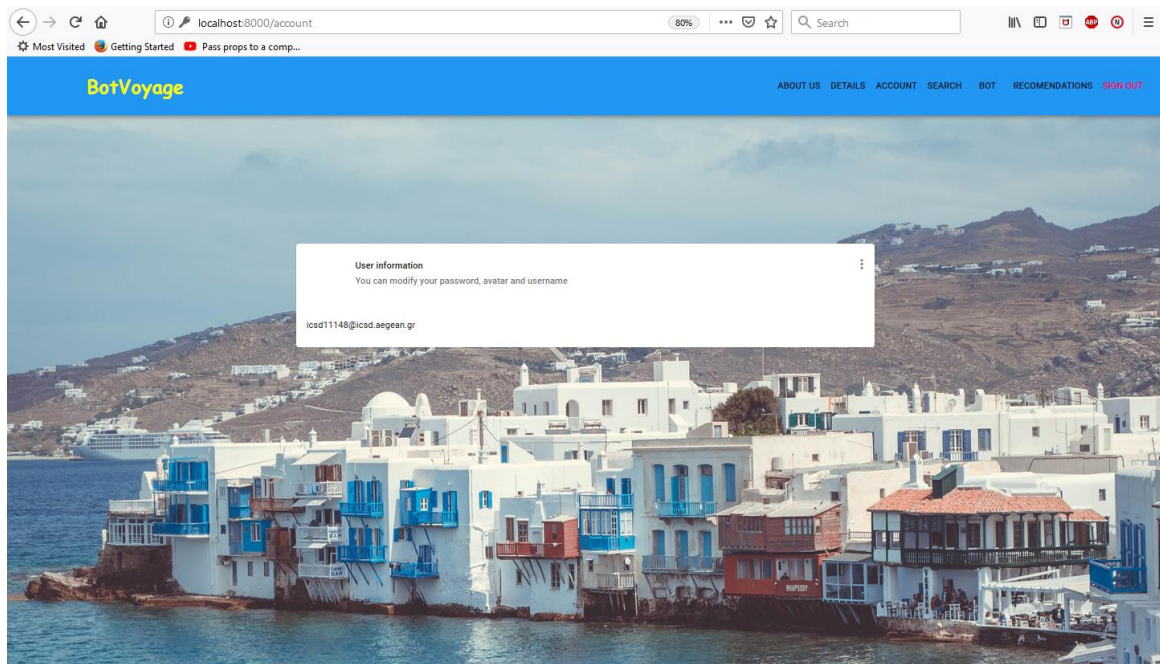
η πρώτη σελίδα που του εμφανίζεται μετά την είσοδό του, είναι η σελίδα details, στην οποία του παρέχουμε μια περιγραφή σχετικά με τις επιλογές που έχει μέσω κατηγοριών αναζήτησης στο bot που είναι ενσωματωμένο στην πλατφόρμα μας, αλλά και μια λίστα χώρων που του παρέχουμε, σαν προτεινόμενα με βάση τις περισσότερες αναζητήσεις που έχει γίνει από την μάζα που χρησιμοποίησε την πλατφόρμα μας. Οι κατηγορίες τις οποίες περιέχει το μποτ είναι καφέ, μπεργκερ, φαρμακεία, ενοικιάσεις αυτοκινήτων, σινεμά, καζίνο, βενζινάδικα, πτήσεις, καιρός, παραλίες, μύρα, πανεπιστήμια, γυμναστήρια, paintball, στάδια, γκολφ, μπιλιάρδο, εστιατόρια, αγορές, ζωολογικοί κήποι, σκι, σιδηροδρομικοί σταθμοί, σταθμοί λεωφορείων, λιμάνια, κλαμπ, θέατρα, φούρνοι, μουσεία, ψώνια, ξενοδοχεία, music stores, sushi, μεξικάνικο, σουβλάκι, pizza, crepery. Επιπρόσθετα, μέσω αλληλεπίδρασης με την βάση RealTime Database του Firebase εμφανίζεται ο τελευταίος χρήστης που έκανε εγγραφή στην πλατφόρμα μας. Αυτό ανανεώνεται κάθε φορά που αποκτούμε έναν καινούργιο χρήστη μέσω της εγγραφής.



Εικόνα 103 - Σελίδα Details - BotVoyage Platform - (δικό μας)

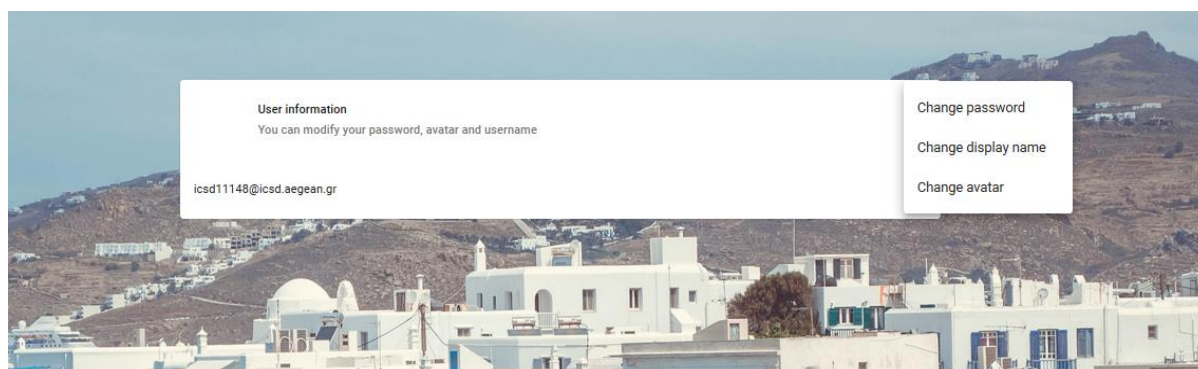
7.8 Σελίδα Account- προφίλ του συνδεδεμένου χρήστη

Σε αυτή τη σελίδα, παρουσιάζονται τα στοιχεία του συνδεδεμένου χρήστη με τα οποία έχει πραγματοποιήσει την σύνδεση στην πλατφόρμα, όπως το email, username.



Εικόνα 104 - Σελίδα Account - BotVoyage Platform - (δικό μας)

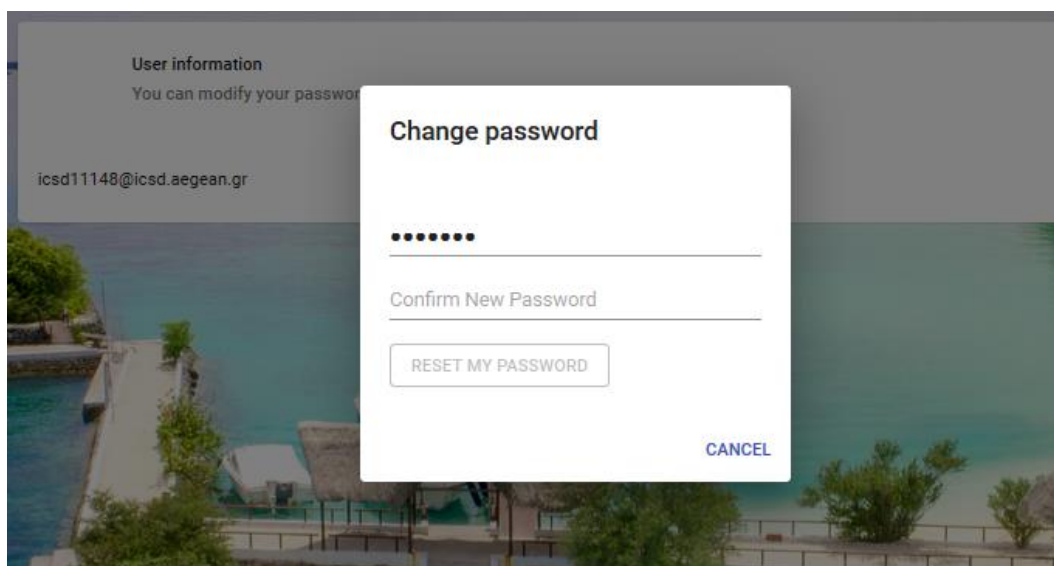
7.9 Επιλογές του χρήστη



Εικόνα 105 - Σελίδα Account - BotVoyage Platform - (δικό μας)

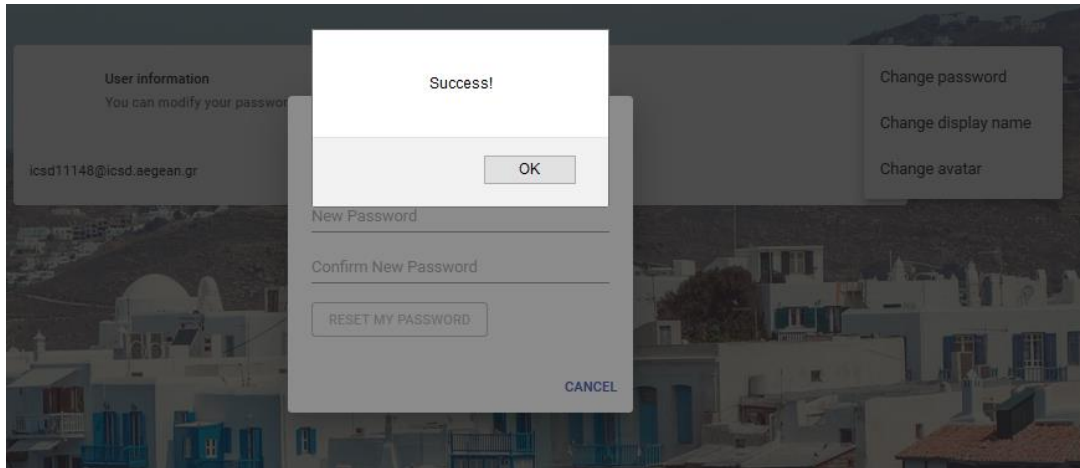
7.9.1 Αλλαγή Password.

Ο χρήστης έχει την δυνατότητα να αλλάξει το password του, με την πληκτρολόγηση ενός καινούργιου κωδικού, το οποίο θα ενημερωθεί αυτόματα στην βάση μας και την επόμενη φορά που θα θελήσει να κάνει σύνδεση θα χρησιμοποιήσει αυτό το καινούργιο password.



Εικόνα 106 - Αλλαγή Password - BotVoyage Platform - (δικό μας)

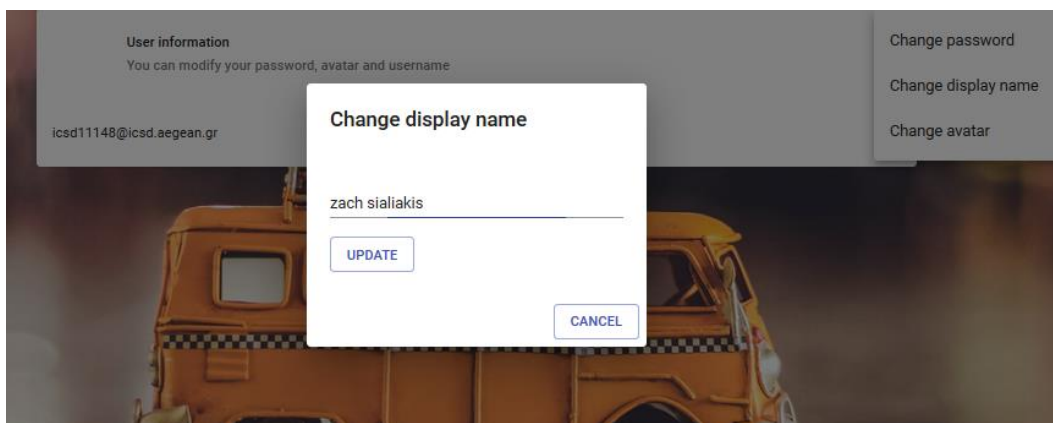
Αν η διαδικασία είναι επιτυχής ένα καινούργιο παράθυρο θα ανοίξει προς ενημέρωσή του.



Εικόνα 107 - Μήνυμα Success - BotVoyage Platform - (δικό μας)

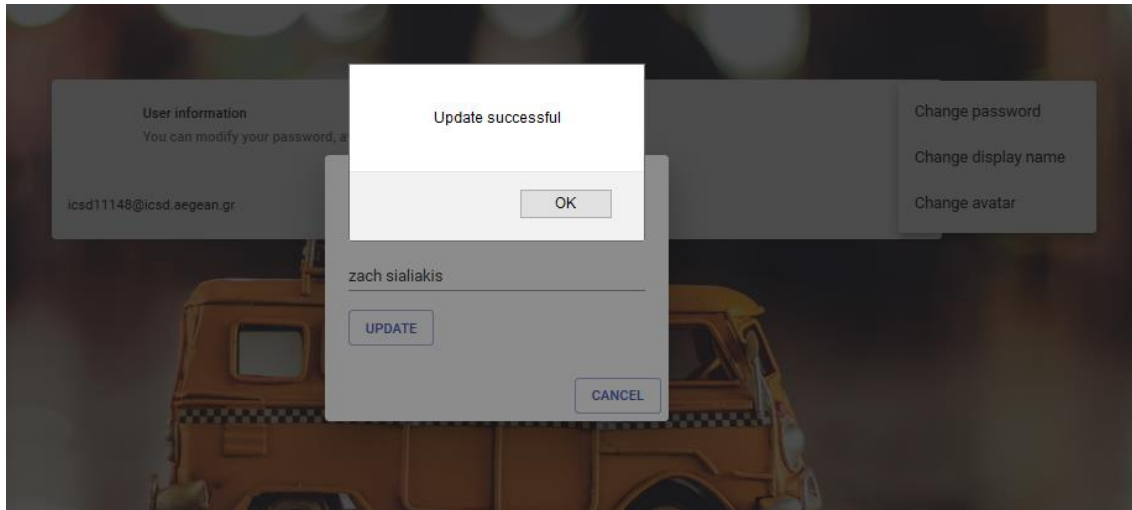
7.9.2 Αλλαγή του ονόματος προφίλ

Ο χρήστης έχει την δυνατότητα αλλαγής του ονόματός του που φαίνεται κάτω στις πληροφορίες του.



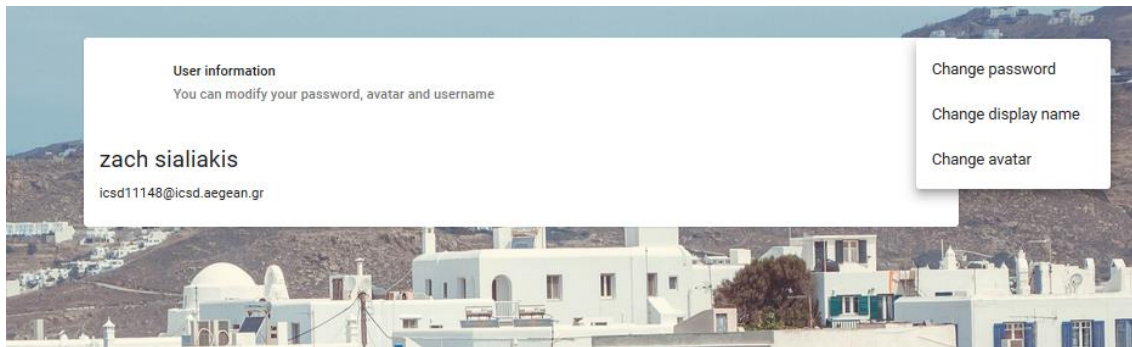
Εικόνα 108 - Αλλαγή ονόματος προφίλ - BotVoyage Platform - (δικό μας)

Όπως και πριν ένα καινούργιο παράθυρο θα εμφανιστεί με την ενημέρωση του χρήστη για την ανανέωση του ονόματός του.



Εικόνα 109 - Μήνυμα Update Successful - BotVoyage Platform - (δικό μας)

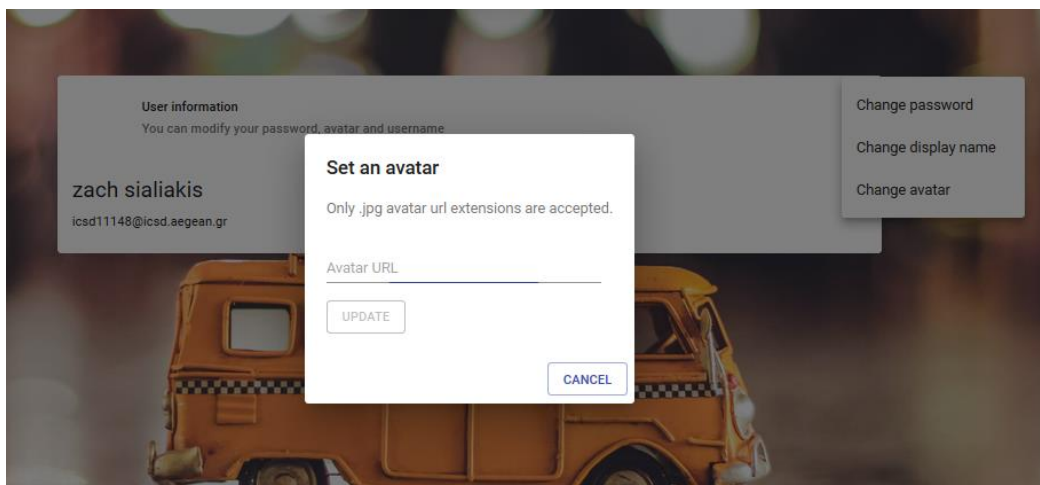
Εμφάνιση του καινούργιου ονόματος στις πληροφορίες του προφίλ του.



Εικόνα 110 - Ενημέρωση Ονόματος Προφίλ - BotVoyage Platform - (δικό μας)

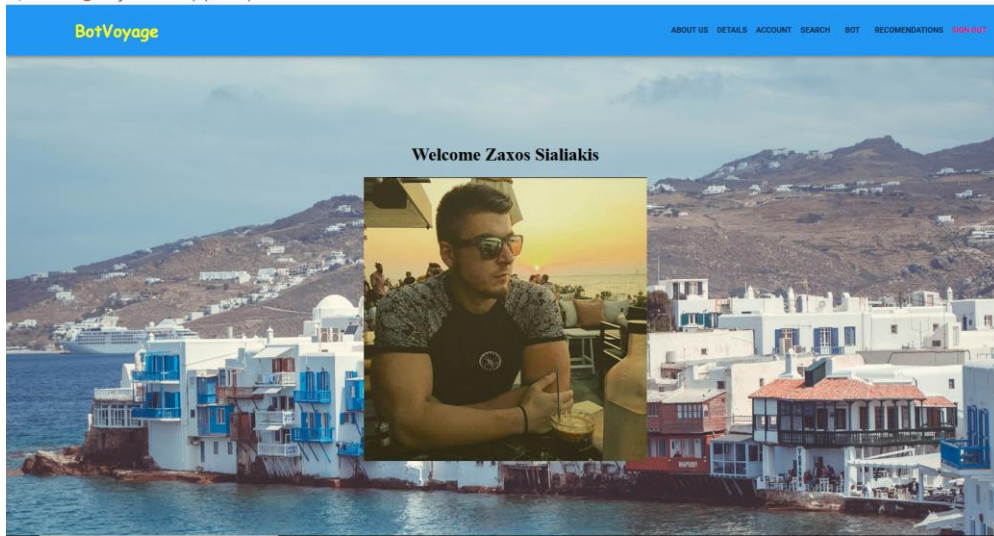
7.9.3 Αλλαγή της εικόνας προφίλ του χρήστη.

Ο χρήστης έχει την δυνατότητα να ορίσει και να αλλάξει την εικόνα προφίλ του αρκεί η μορφή της εικόνας να είναι σε .jpg url extension.

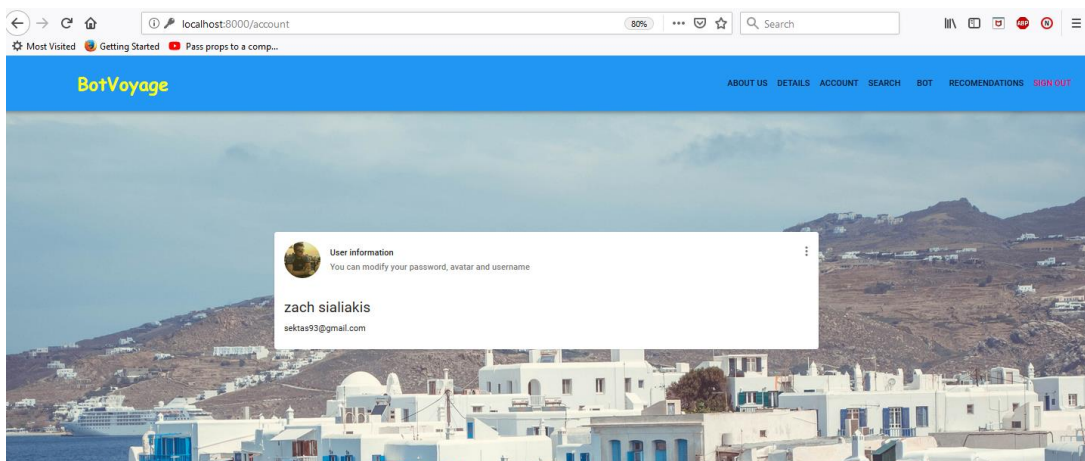


Εικόνα 111 - Αλλαγή Εικόνας Προφίλ - BotVoyage Platform - (δικό μας)

Σε περίπτωση που κάποιος χρήστης πραγματοποιήσει σύνδεση μέσω google, τότε η εικόνα προφίλ του στον λογαριασμό του, ενημερώνεται αυτόματα από την εικόνα προφίλ του στον λογαριασμό google που διαθέτει.



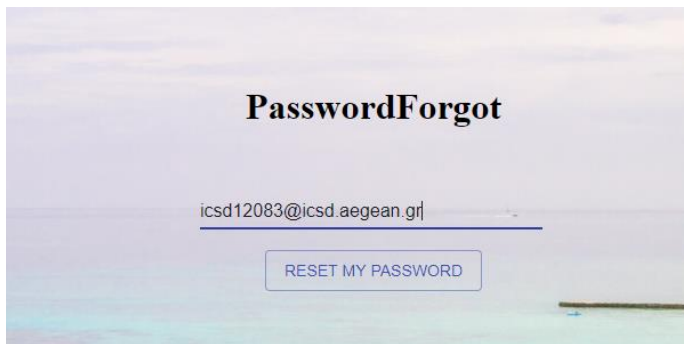
Εικόνα 112 - Μήνυμα Καλωσορίσματος με Google Login - BotVoyage Platform - (δικό μας)



Εικόνα 113 - Ενημέρωση Εικόνας Προφίλ από Google - BotVoyage Platform - (δικό μας)

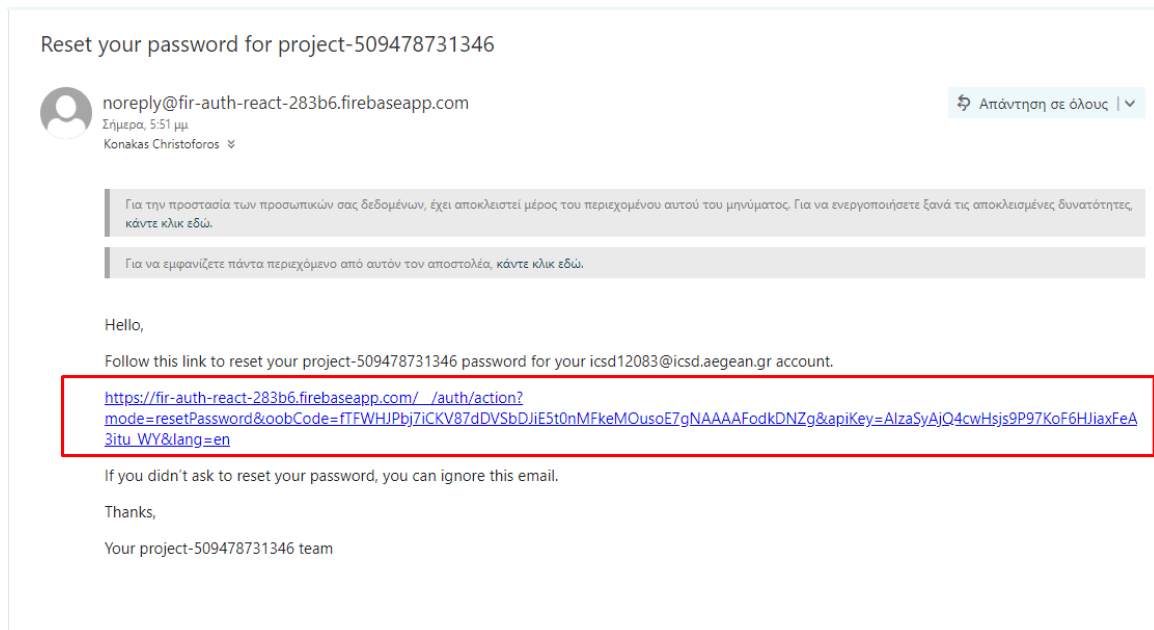
7.9.4 Επιλογή PasswordForgot στην σελίδα SIGN IN

Ο χρήστης επιλέγει το email του και το κουμπί RESET MY PASSWORD



Εικόνα 114 -Reset Password - BotVoyage Platform - (δικό μας)

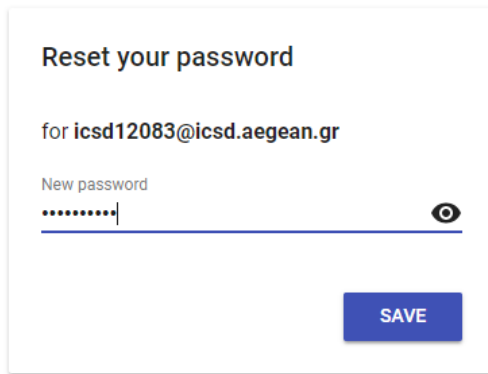
Έπειτα ένα αυτοματοποιημένο email θα εμφανιστεί στην ηλεκτρονική διεύθυνση που συμπλήρωσε



Εικόνα 115 - Μήνυμα email για αλλαγή Password - BotVoyage Platform - (δικό μας)


Κάνοντας κλικ στο url που του εμφανίστηκε προερχόμενο από την εφαρμογή μας στο firebase

μπορεί να αλλάξει το password του.



Reset your password

for **icsd12083@icsd.aegean.gr**

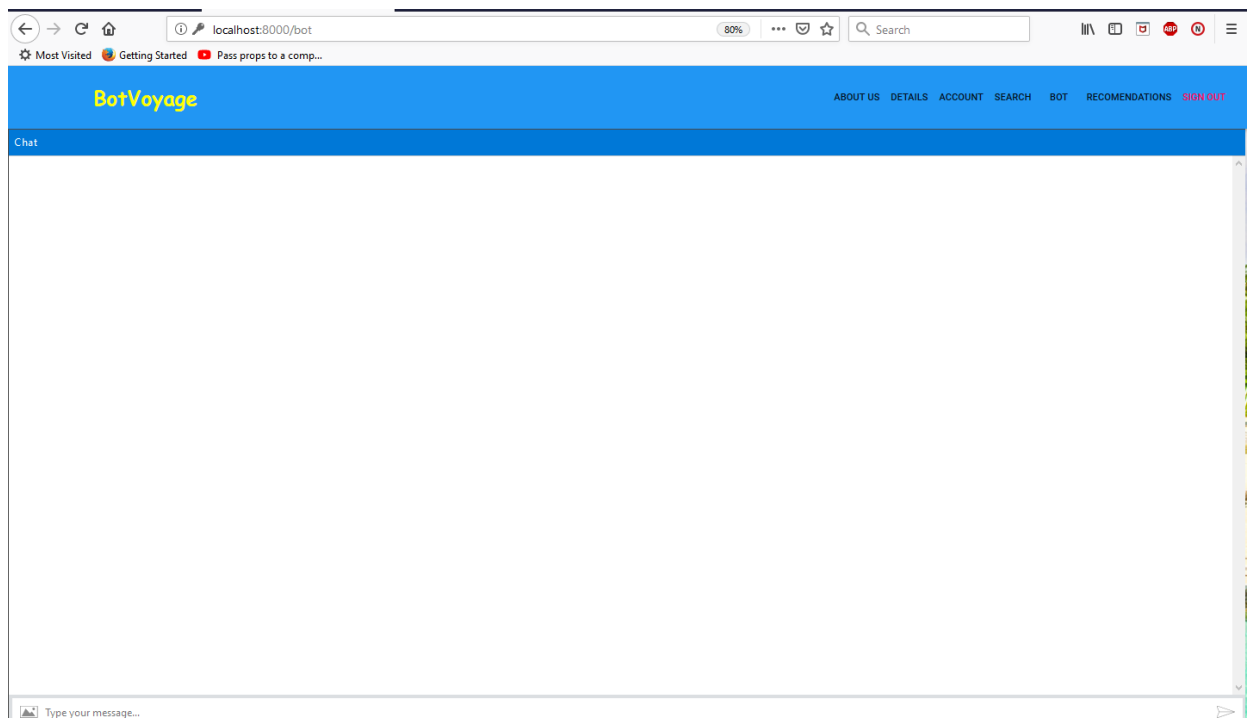
New password
.....| 

SAVE

Εικόνα 116 - Reset your password - BotVoyage Platform - (δικό μας)

7.10 Σελίδα του Bot

Σε περίπτωση που κάποιος χρήστης επιθυμεί να χρησιμοποιήσει το μποτ της πλατφόρμας αυτό μπορεί να γίνει πατώντας το κουμπί BOT στο menu bar της πλατφόρμας οδηγώντας τον σε μια καινούργια σελίδα <http://localhost:8000/bot> όπου στεγάζεται το μποτ όπως φαίνεται παρακάτω

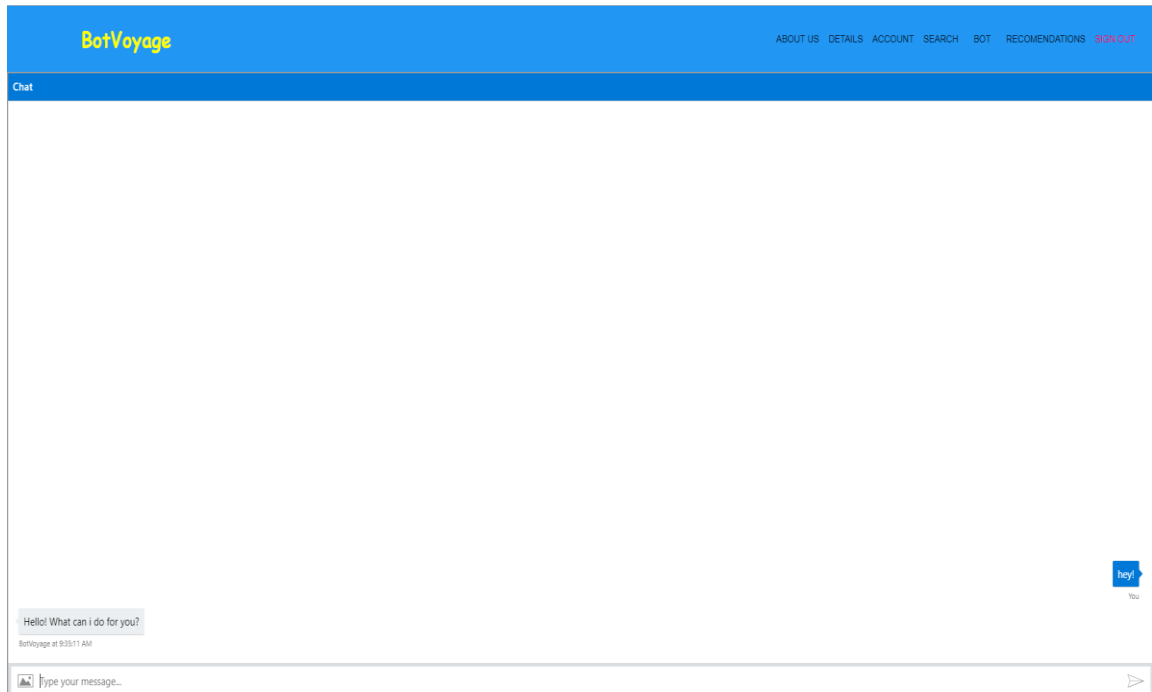


Εικόνα 117 - Το bot στην πλατφόρμα - BotVoyage Platform - (δικό μας)

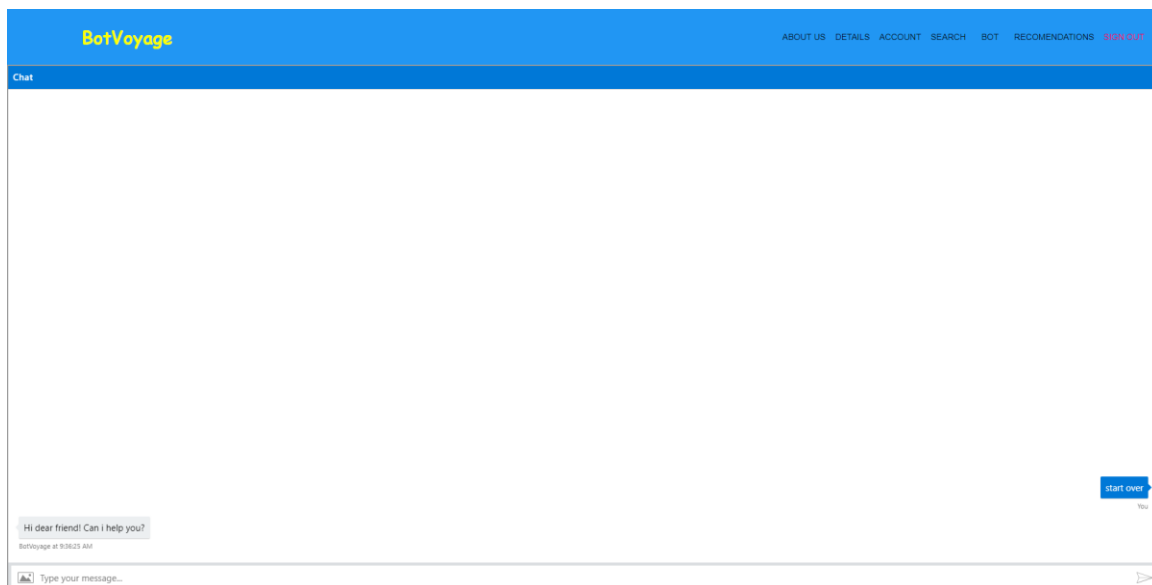
Στην οποία μπορεί να αναζητήσει όλες τις κατηγορίες που του αναφέρθηκαν στην σελίδα detail όπως φαίνεται παρακάτω.

7.10.1 Αποτελέσματα BOT - Παρουσίαση κατηγοριών(επεκτάσεων)

7.10.1.1 Μήνυμα Καλωσορίσματος

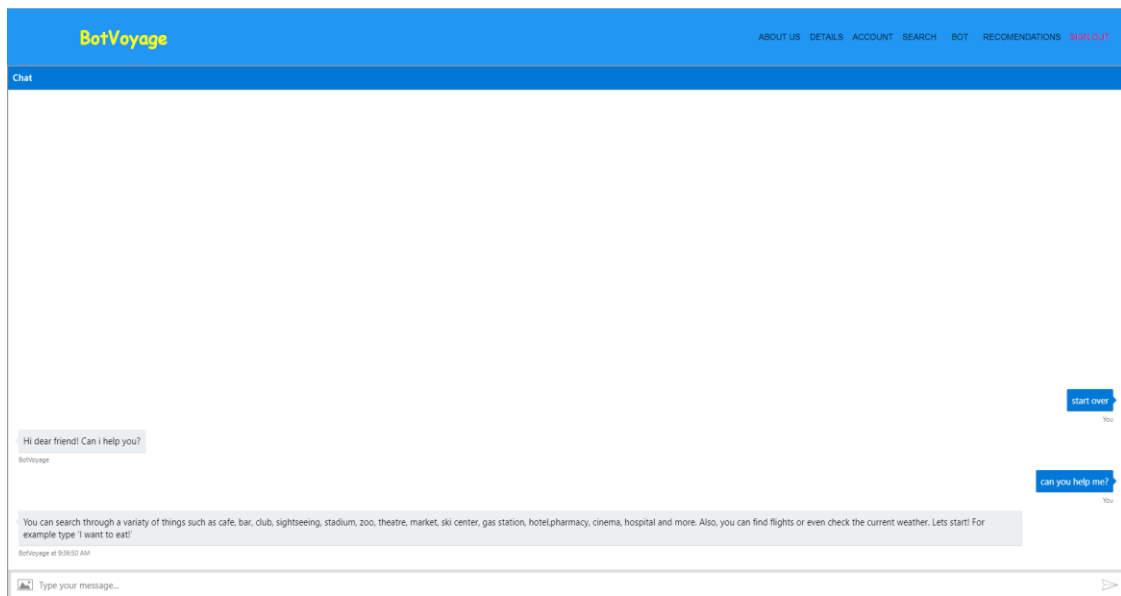


Εικόνα 118 - Μήνυμα Καλωσορίσματος - BotVoyage Platform - (δικό μας)



Εικόνα 119 - Μήνυμα προτροπής επανεκκίνησης - BotVoyage Platform - (δικό μας)

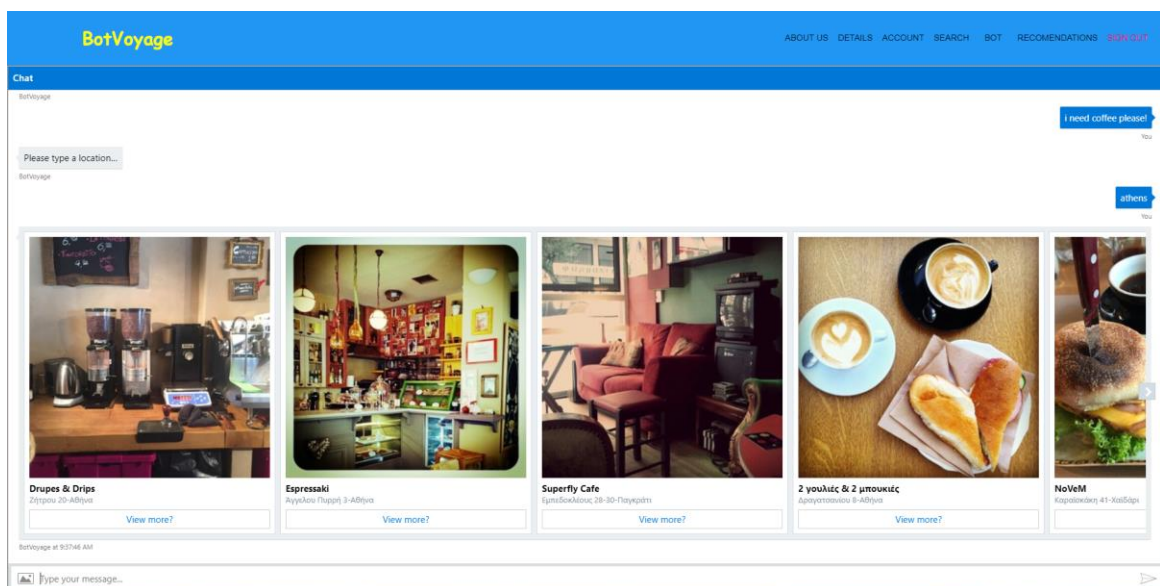
7.10.1.2 Πληροφορίες αναζητήσεων



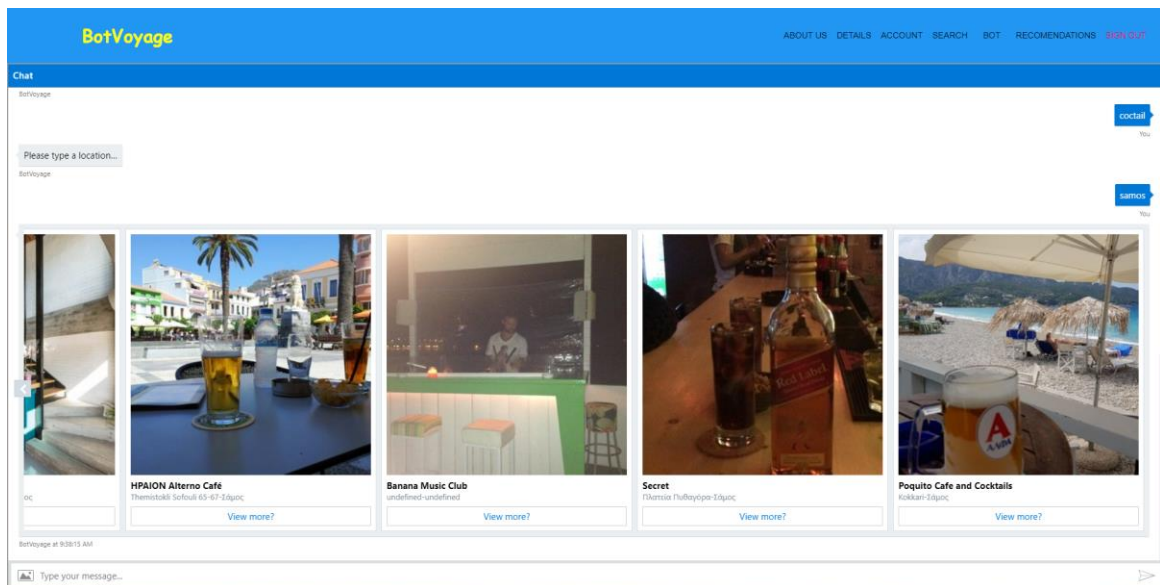
Εικόνα 120 - Πληροφορίες για τις κατηγορίες - BotVoyage Platform - (δικό μας)

7.10.2 Κατηγορίες

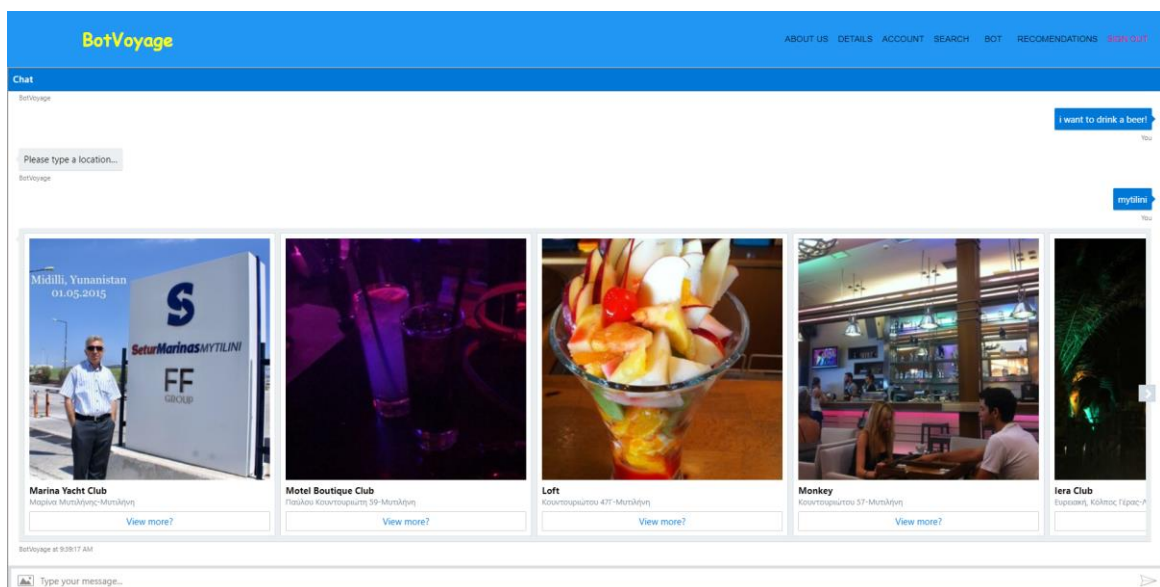
7.10.2.1 DrinkType (coffee, beer, cocktail, wine)



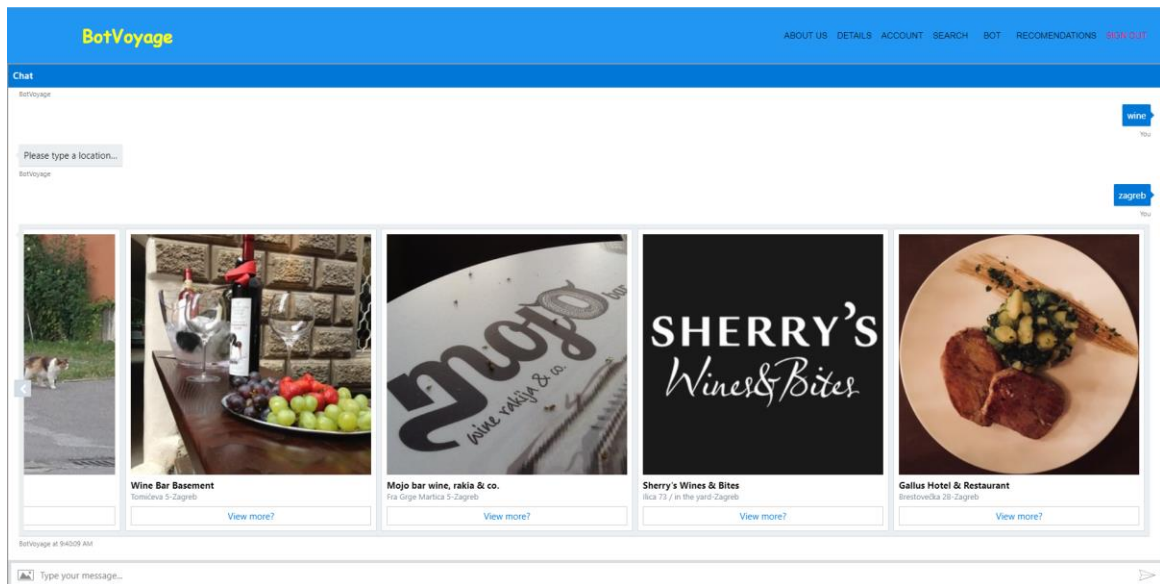
Εικόνα 121 - Αποτελέσματα Coffee στην Αθήνα - BotVoyage Platform - (δικό μας)



Εικόνα 122 - Αποτελέσματα Cocktail στην Σάμο - BotVoyage Platform - (δικό μας)

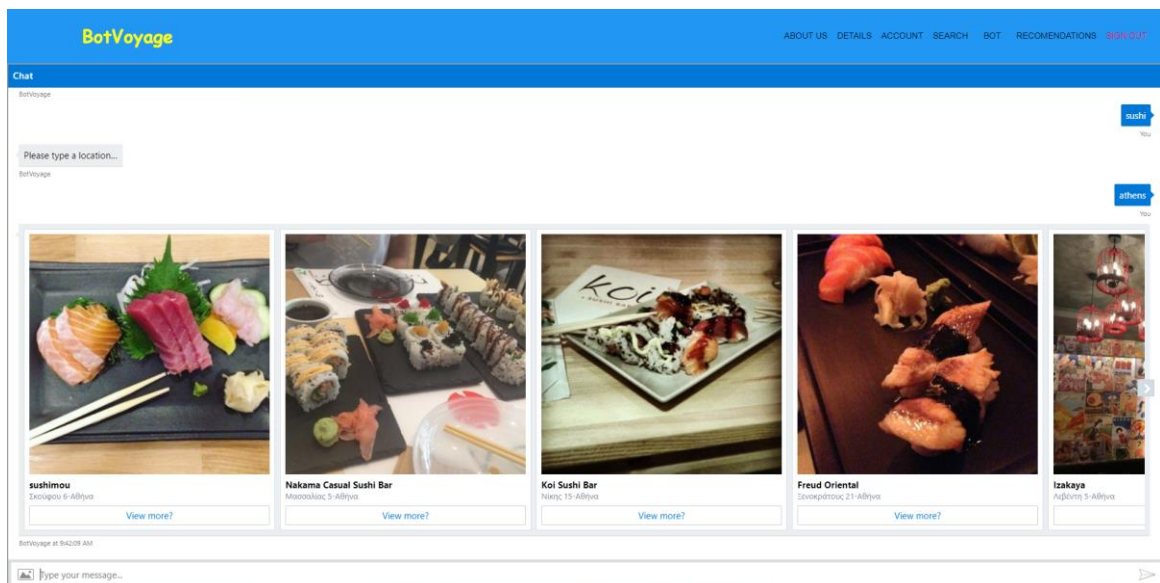


Εικόνα 123 - Αποτελέσματα beer στην Μυτιλήνη - BotVoyage Platform - (δικό μας)

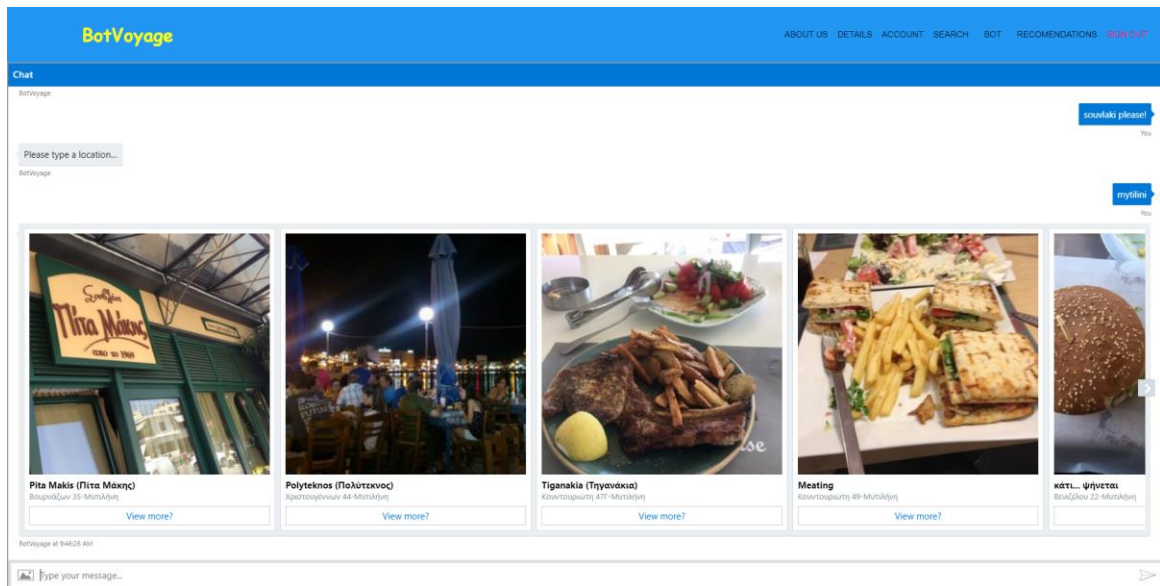


Εικόνα 124 - Αποτελέσματα wine στο Ζάγκρεπ - BotVoyage Platform - (δικό μας)

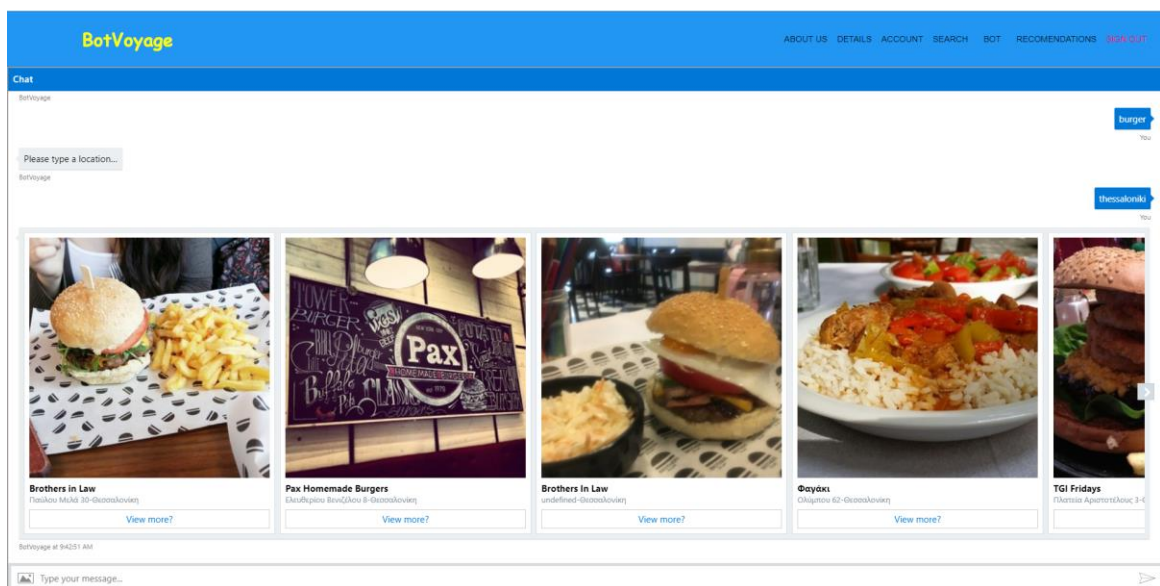
7.10.2.2. FoodType (sushi, souvlaki, pizza, mexican, burger, restaurant)



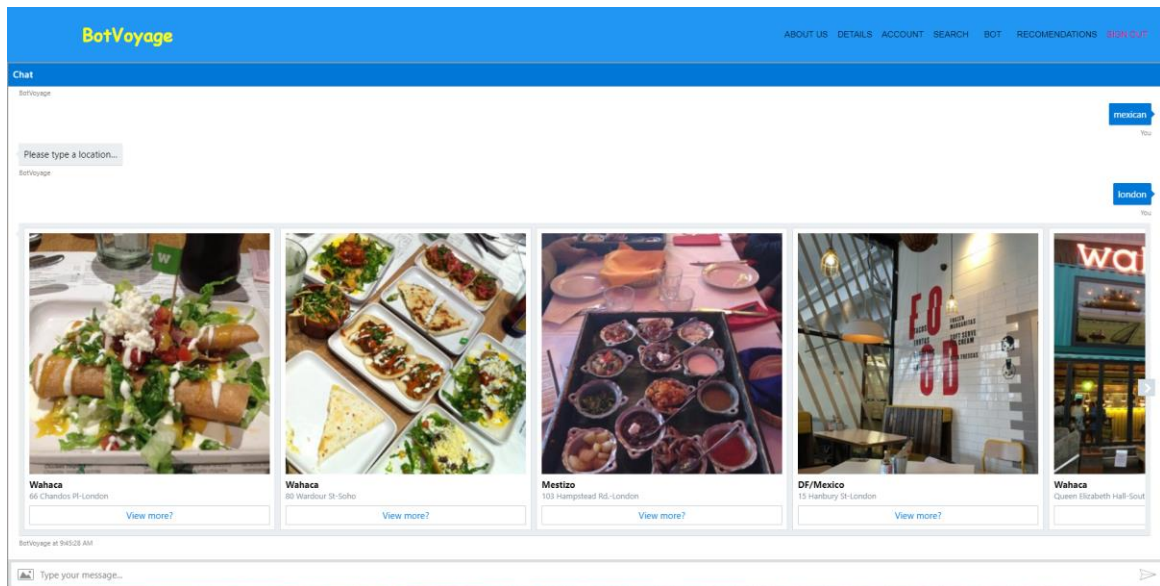
Εικόνα 125 - Αποτελέσματα Sushi στην Αθήνα - BotVoyage Platform - (δικό μας)



Εικόνα 126 - Αποτελέσματα Souvlaki στην Μυτιλήνη - BotVoyage Platform - (δικό μας)

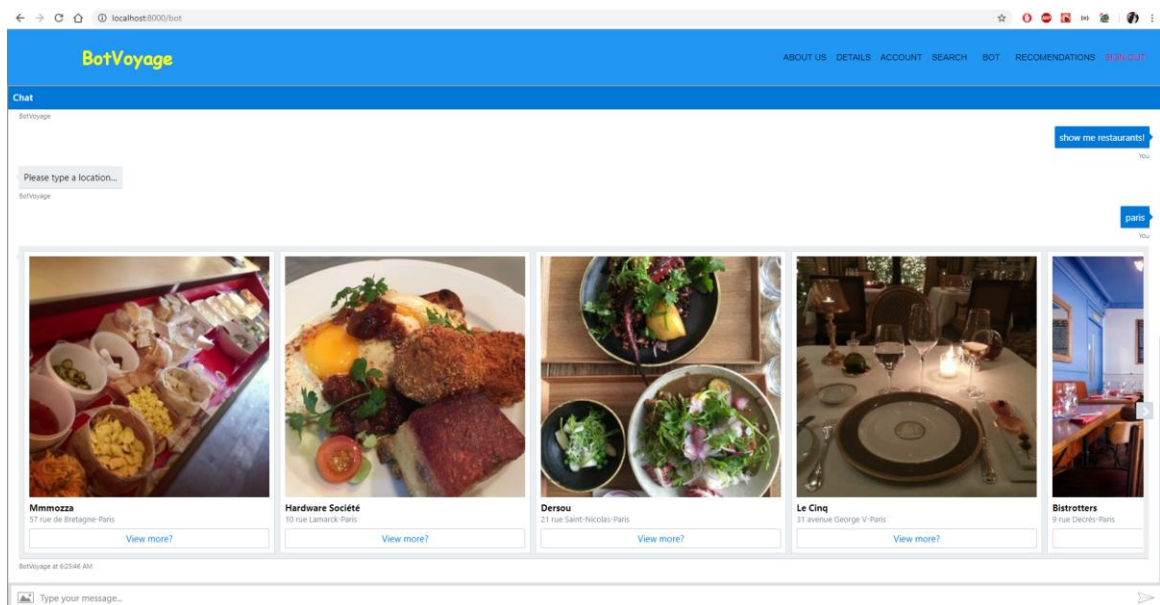


Εικόνα 127 - Αποτελέσματα Burger στην Θεσσαλονίκη - BotVoyage Platform - (δικό μας)



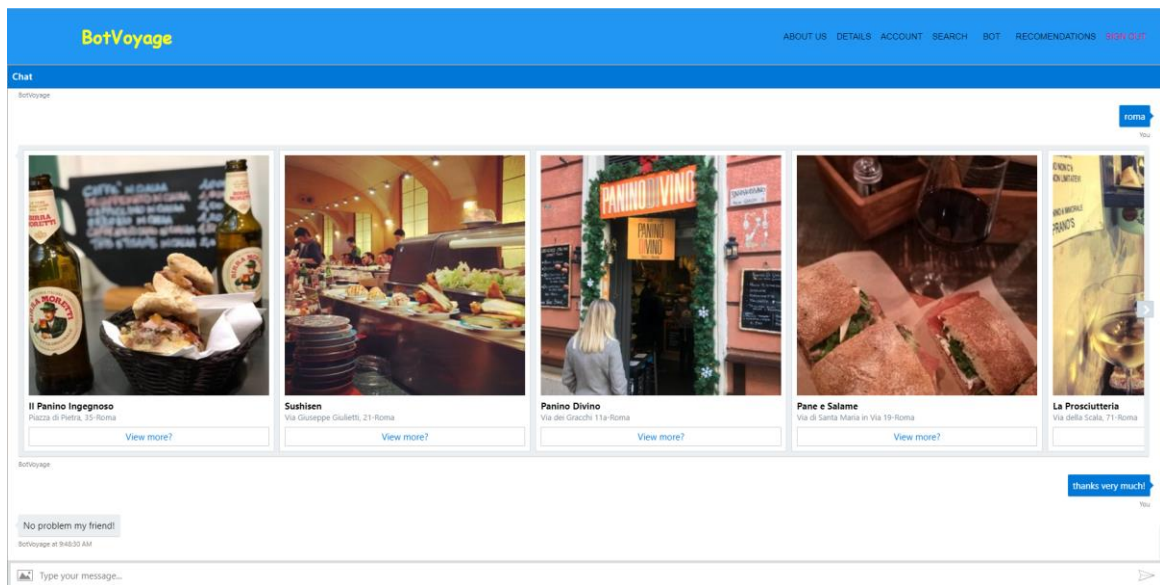
Εικόνα 128 - Αποτελέσματα Mexican στο Λονδίνο - BotVoyage Platform - (δικό μας)

7.10.2.3. Restaurants



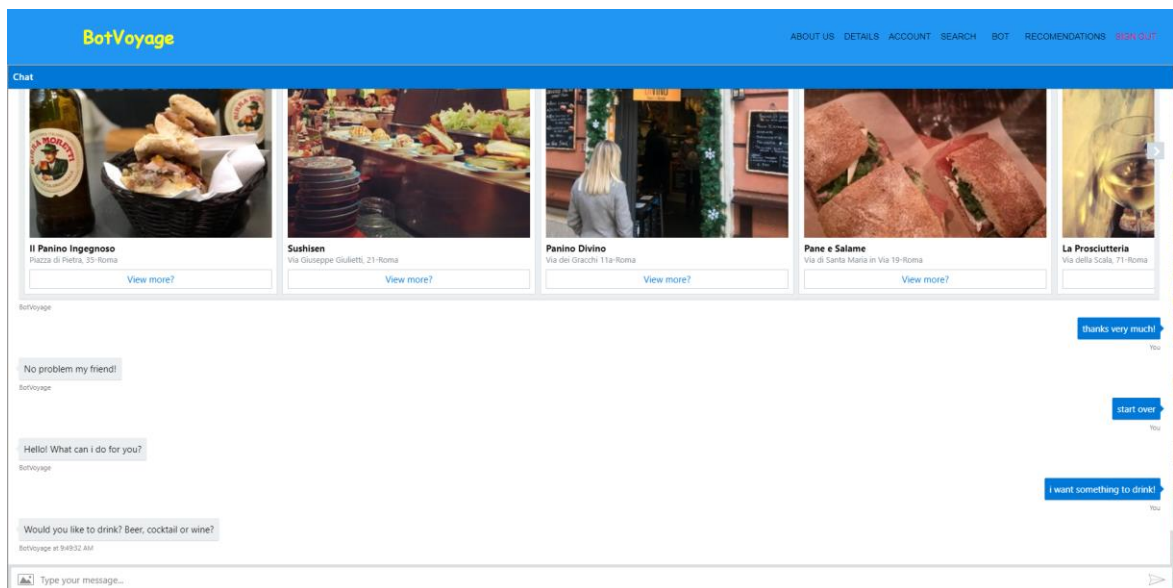
Εικόνα 129 - Αποτελέσματα Restaurants στο Παρίσι - BotVoyage Platform - (δικό μας)

7.10.2.4. Μήνυμα Ευχαρίστησης (English)



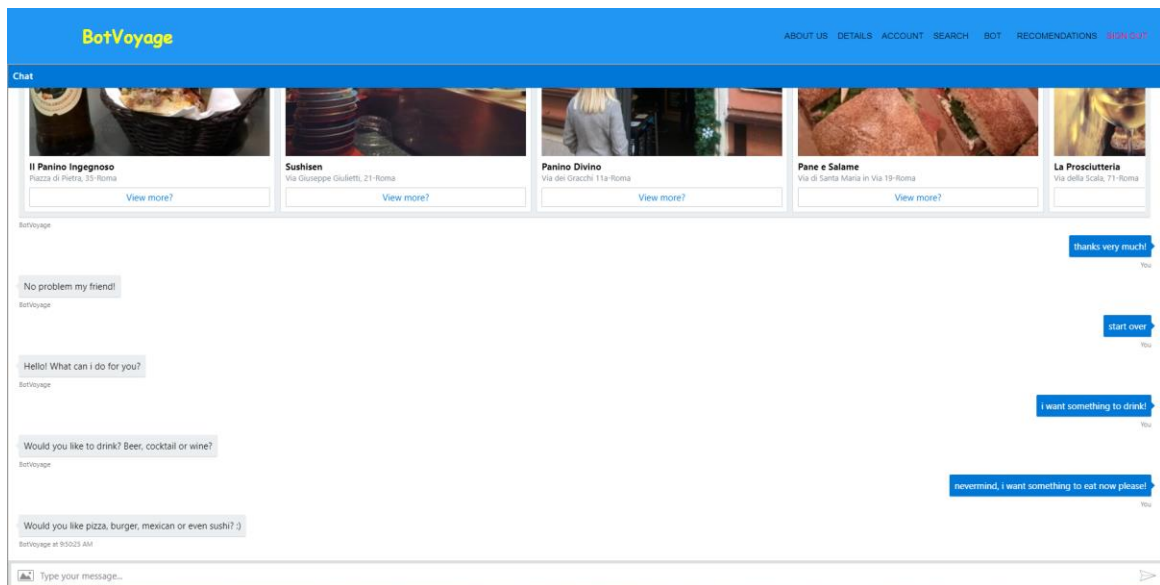
Εικόνα 130 - Μήνυμα Ευχαρίστησης - BotVoyage Platform - (δικό μας)

7.10.2.5 Επιλογή Ροφήματος



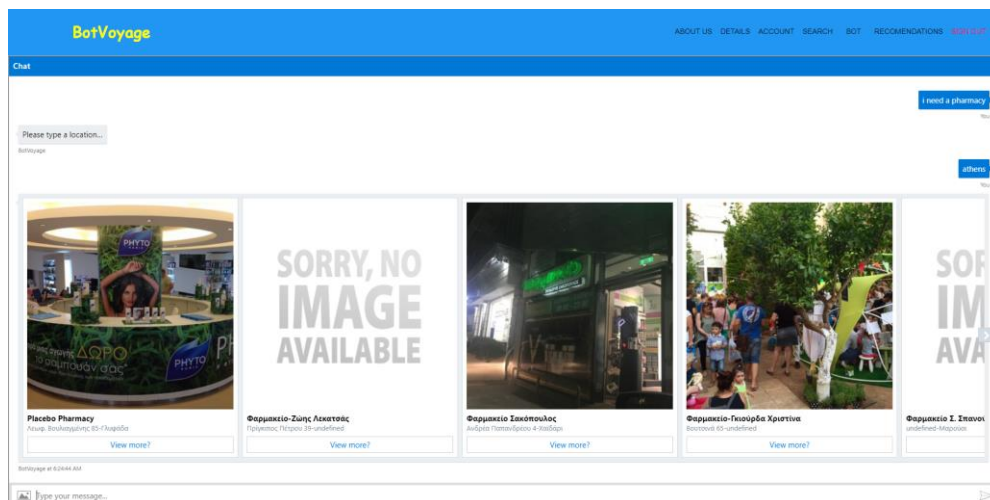
Εικόνα 131 - Επιλογή Ροφήματος - BotVoyage Platform - (δικό μας)

7.10.2.6 Επιλογή φαγητού



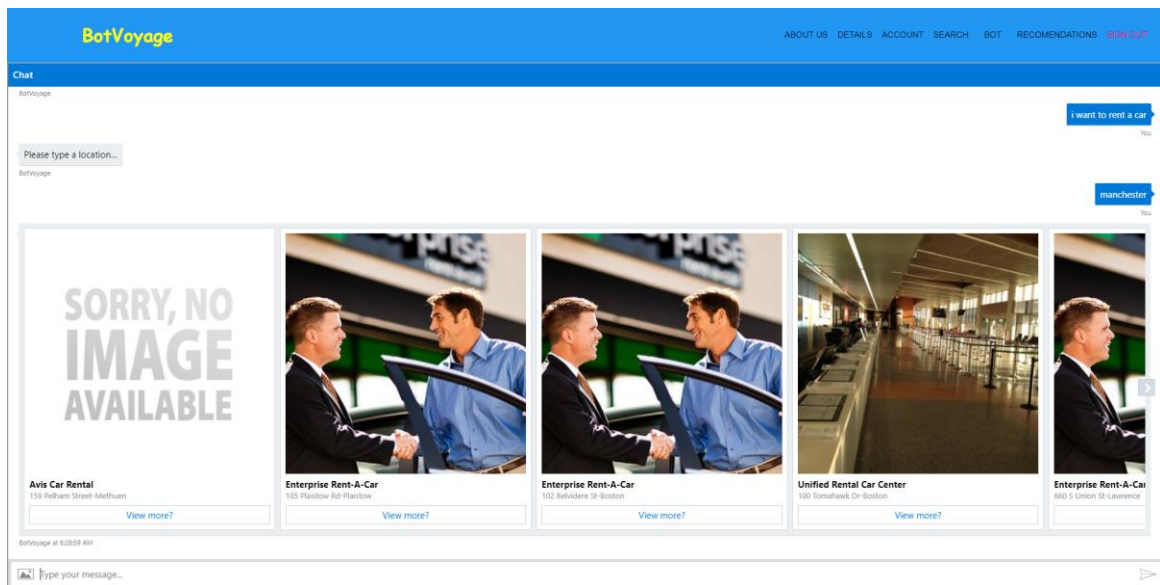
Εικόνα 132 - Επιλογή Φαγητού - BotVoyage Platform - (δικό μας)

7.10.2.7 Pharmacy



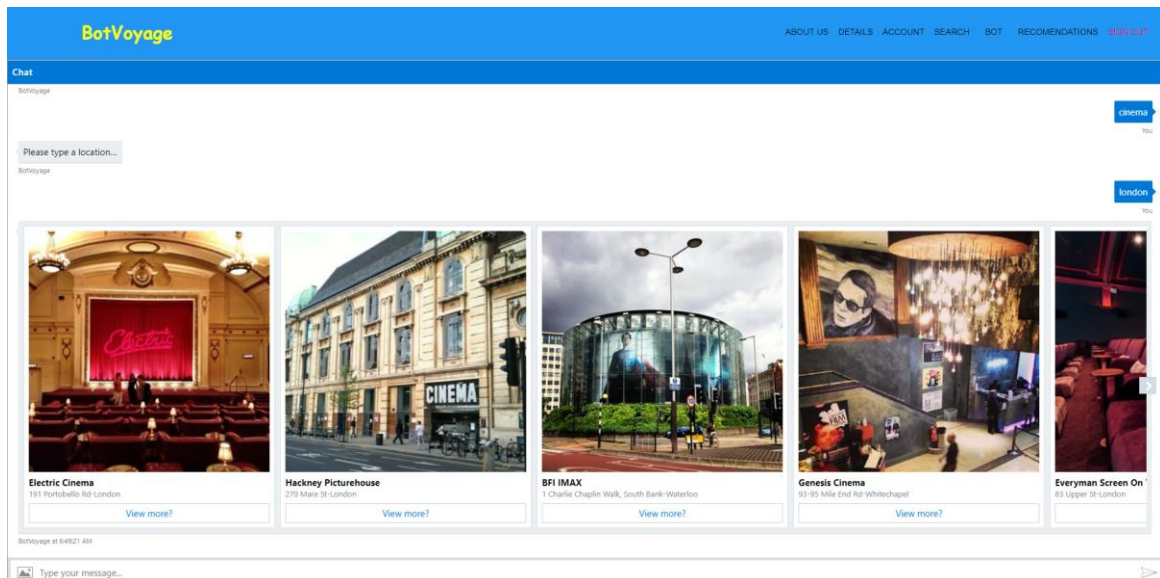
Εικόνα 133 - Αποτελέσματα Pharmacy στην Αθήνα - BotVoyage Platform - (δικό μας)

7.10.2.8. Rent a car



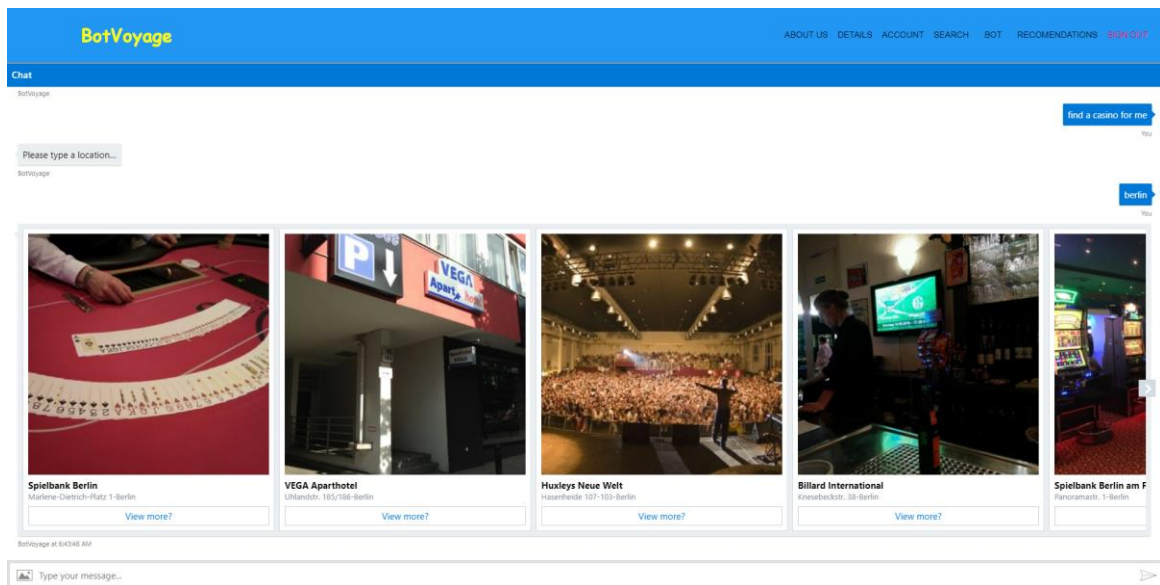
Εικόνα 134 - Αποτελέσματα Rent a car στο Manchester - BotVoyage Platform - (δικό μας)

7.10.2.9. Cinema



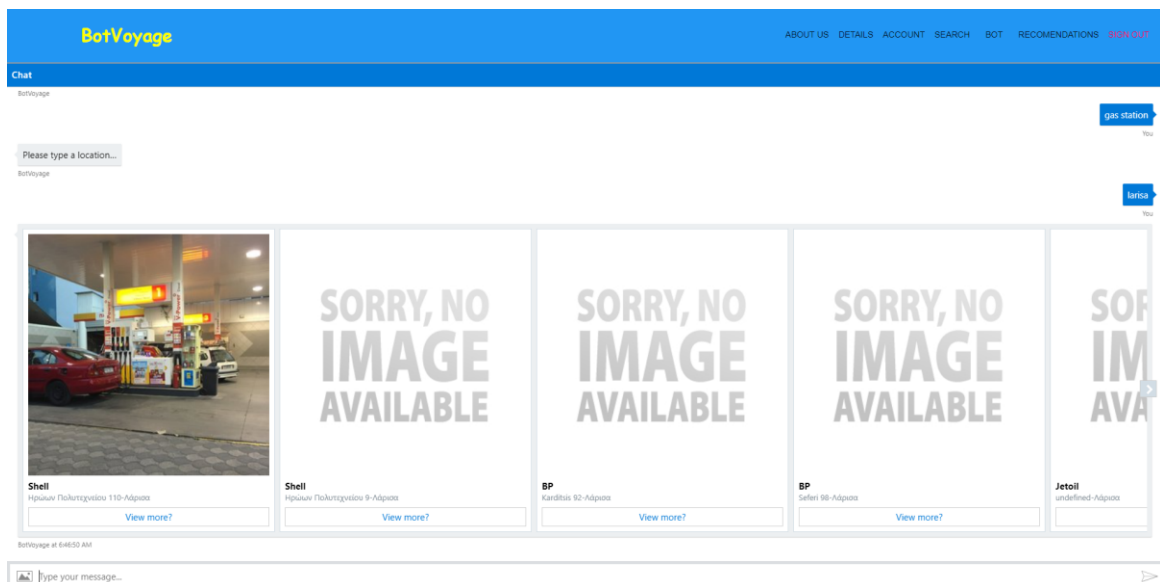
Εικόνα 135 - Αποτελέσματα Cinema στο Λονδίνο - BotVoyage Platform - (δικό μας)

7.10.2.10. Casino



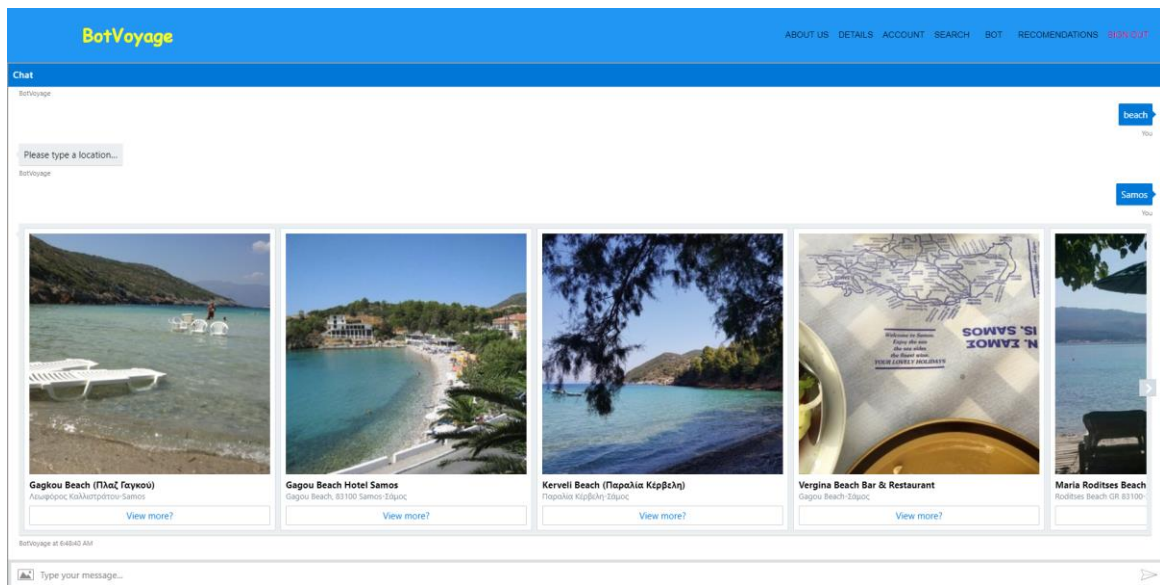
Εικόνα 136 - Αποτελέσματα Casino στο Βερολίνο - BotVoyage Platform - (δικό μας)

7.10.2.11. Gas Station



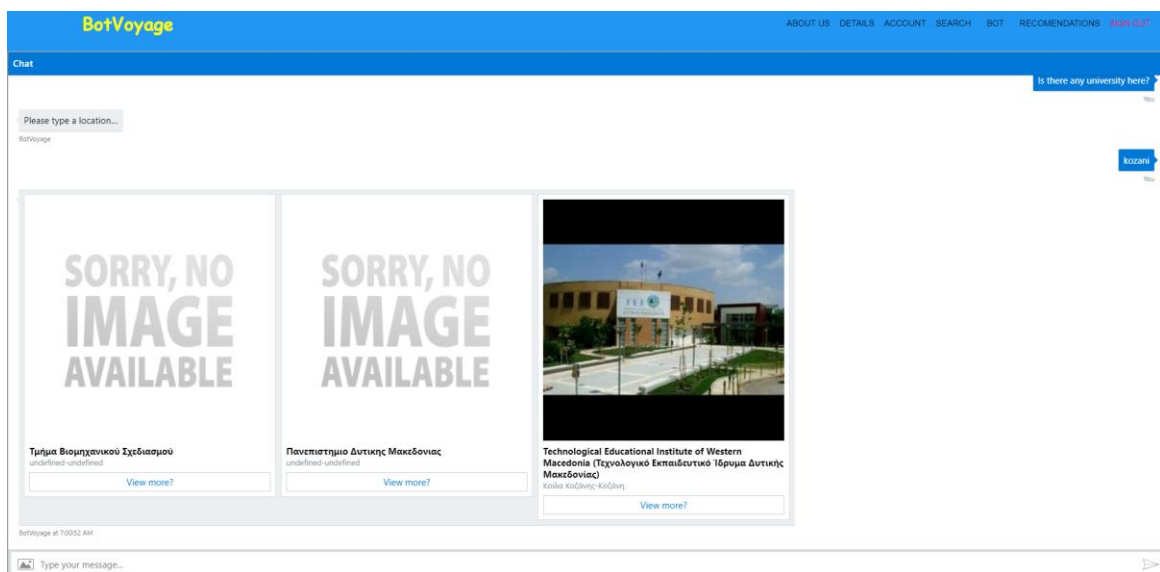
Εικόνα 137 - Αποτελέσματα Gas Station στην Λάρισα - BotVoyage Platform - (δικό μας)

7.10.2.12. Beach



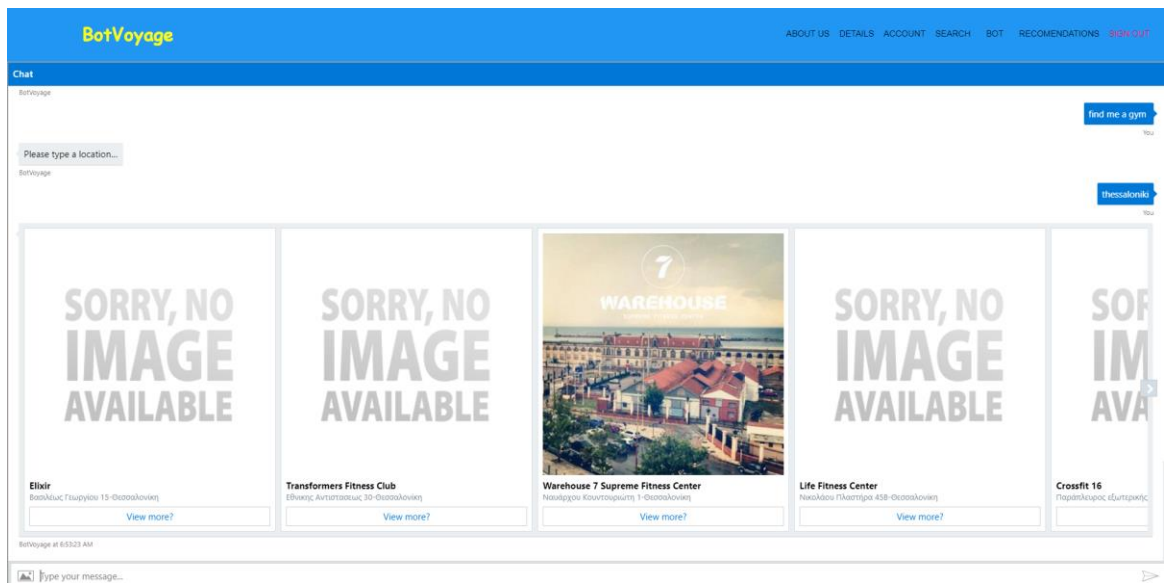
Εικόνα 138 - Αποτελέσματα Beach στην Σάμο - BotVoyage Platform - (δικό μας)

7.10.2.13. University



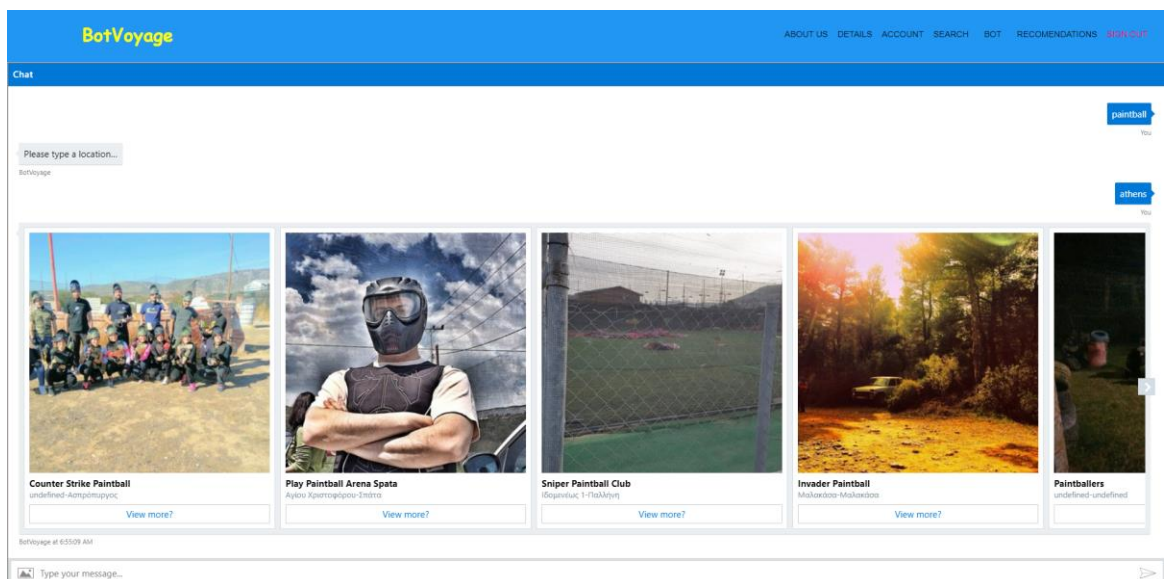
Εικόνα 139 - Αποτελέσματα University στην Κοζάνη - BotVoyage Platform - (δικό μας)

7.10.2.14. Gym



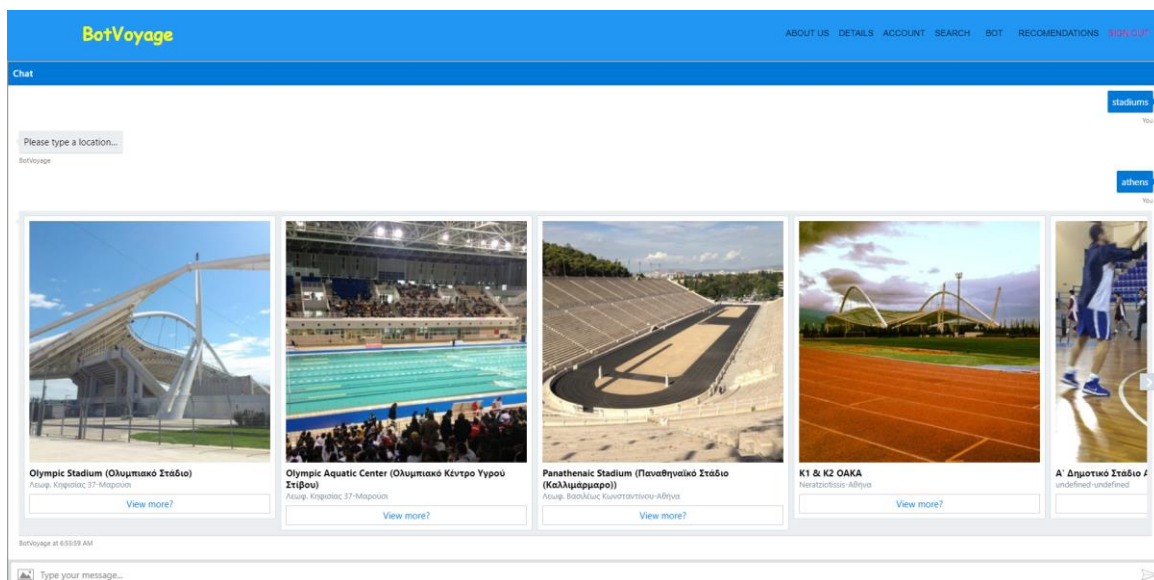
Εικόνα 140 - Αποτελέσματα Gym στην Θεσσαλονίκη - BotVoyage Platform - (δικό μας)

7.10.2.15. Paintball



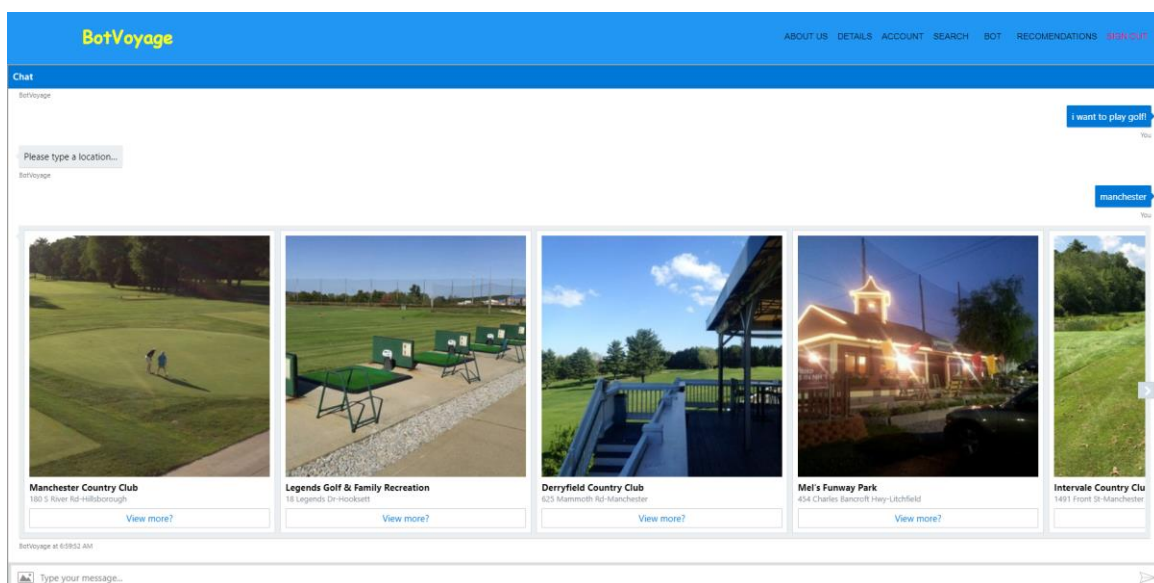
Εικόνα 141 - Αποτελέσματα Paintball στην Αθήνα - BotVoyage Platform - (δικό μας)

7.10.2.16. Stadium



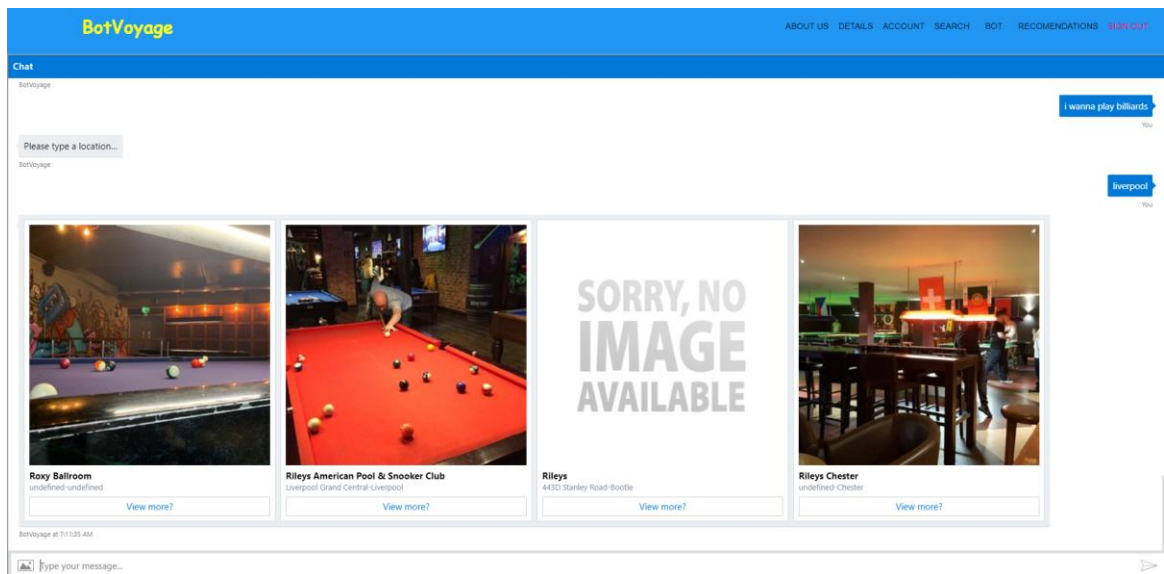
Εικόνα 142 - Αποτελέσματα Stadium στην Αθήνα - BotVoyage Platform - (δικό μας)

7.10.2.17. Golf



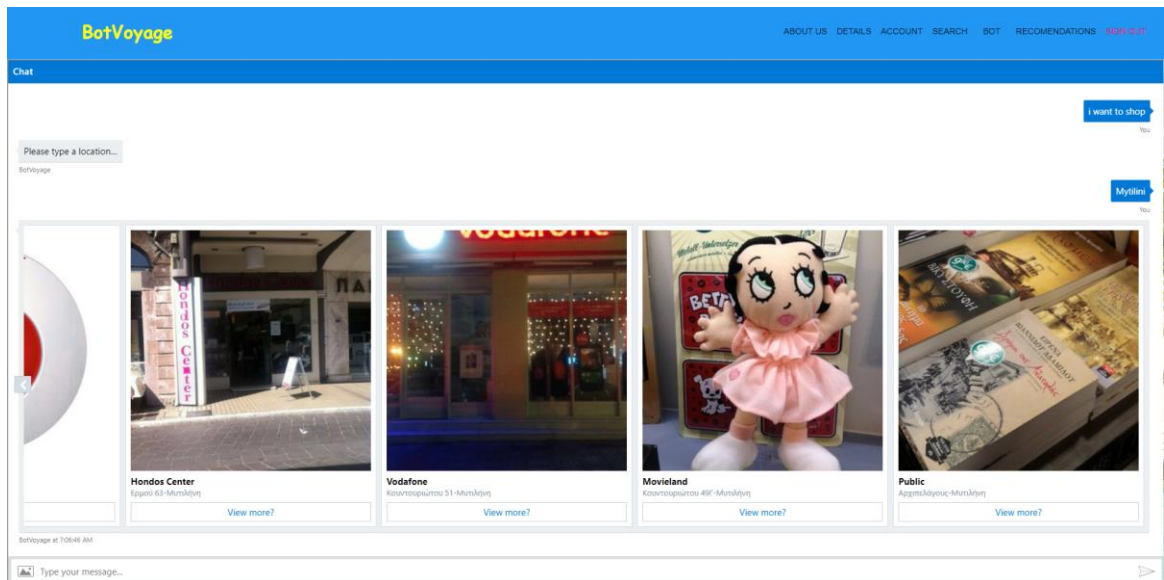
Εικόνα 143 - Αποτελέσματα Golf στο Manchester - BotVoyage Platform - (δικό μας)

7.10.2.18. Billiards



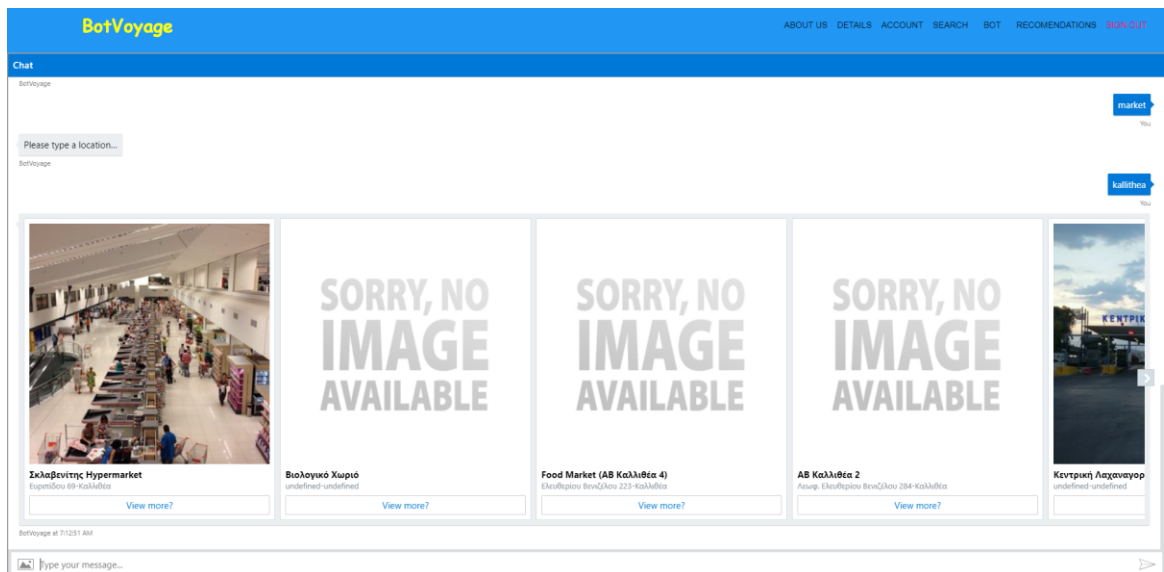
Εικόνα 144 - Αποτελέσματα Billiards στο Λίβερπουλ - BotVoyage Platform - (δικό μας)

7.10.2.19. Shop



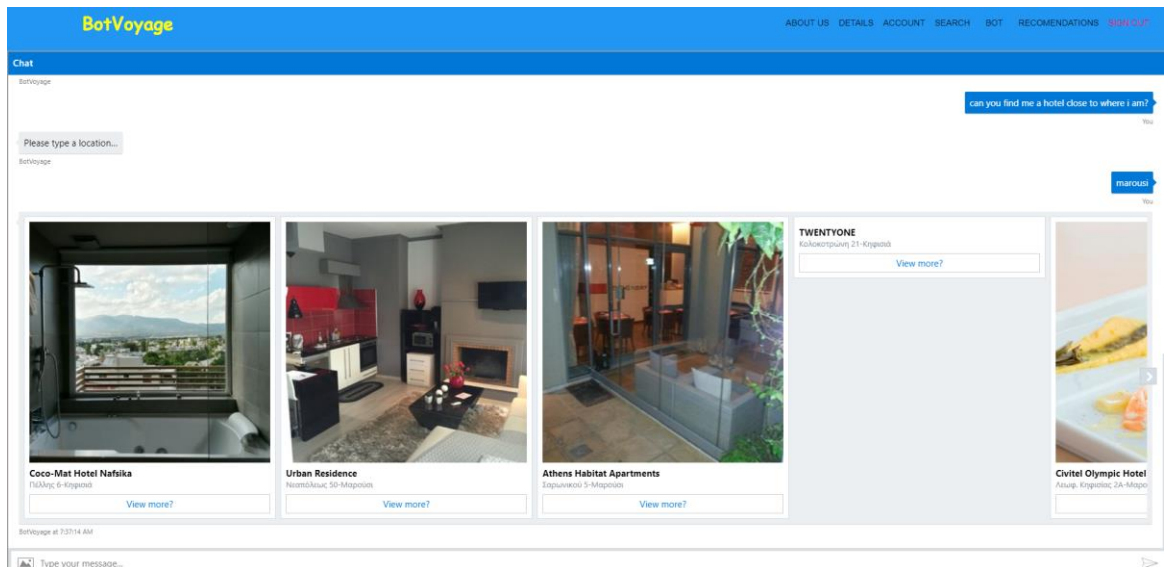
Εικόνα 145 - Αποτελέσματα Shop στην Μυτιλήνη - BotVoyage Platform - (δικό μας)

7.10.2.20. Market



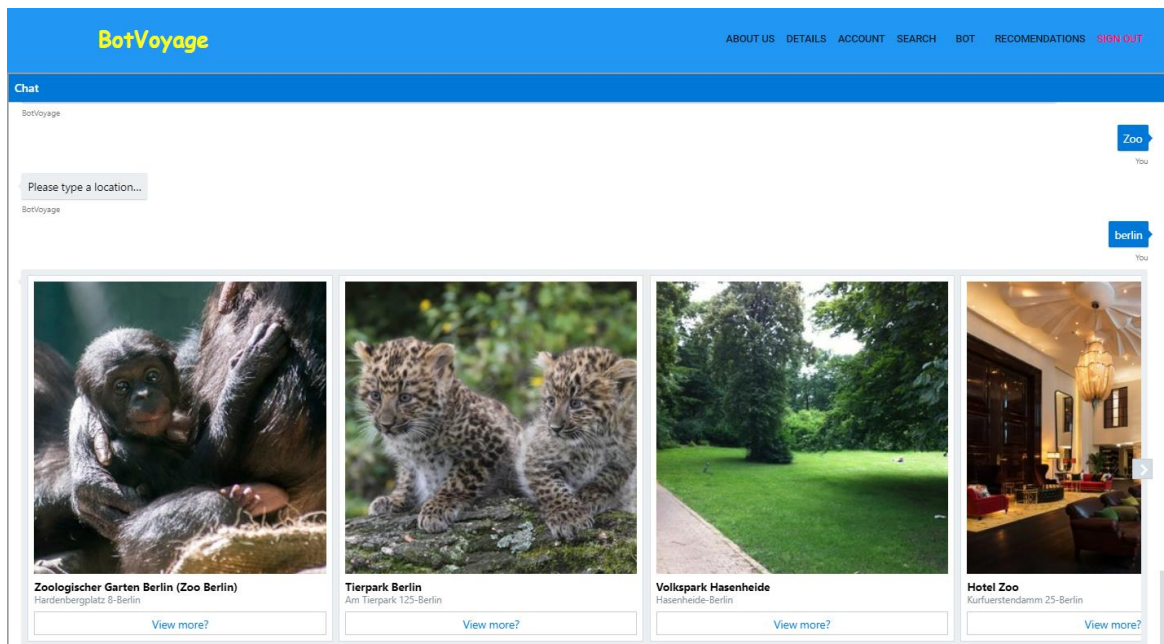
Εικόνα 146 - Αποτελέσματα Market στην Καλλιθέα - BotVoyage Platform - (δικό μας)

7.10.2.21. Hotel



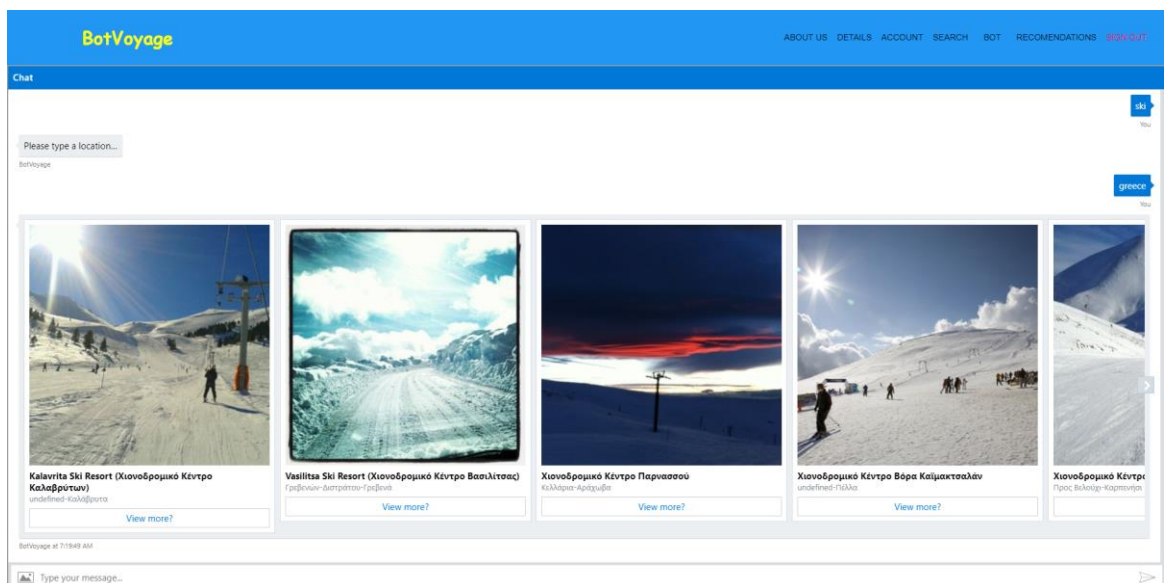
Εικόνα 147 - Αποτελέσματα Hotel στο Μαρούσι - BotVoyage Platform - (δικό μας)

7.10.2.22. Zoo



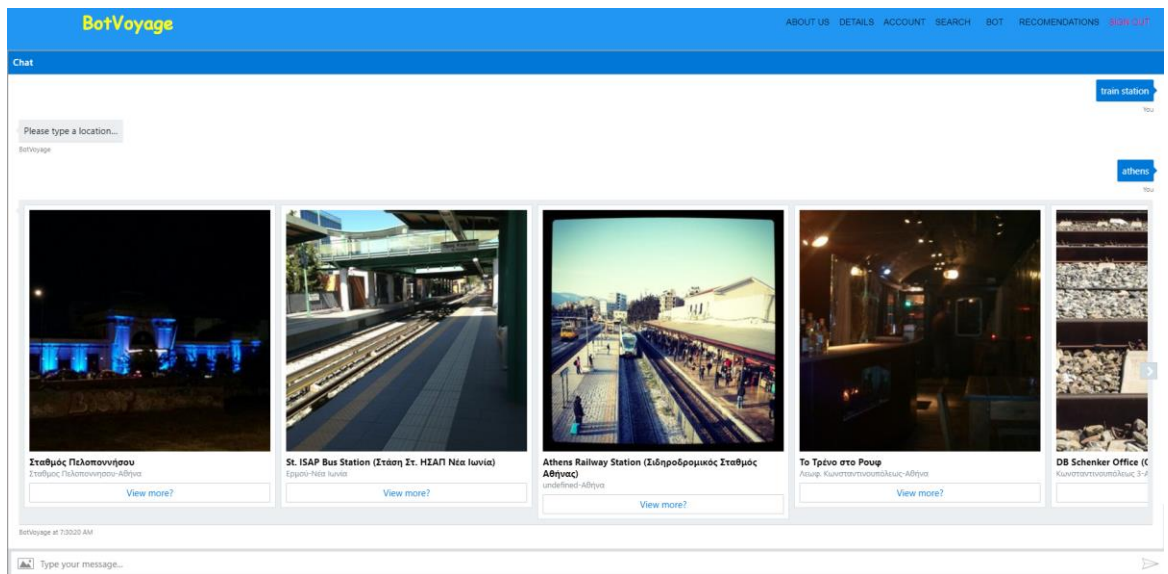
Εικόνα 148 - Αποτελέσματα Zoo στο Βερολίνο - BotVoyage Platform - (δικό μας)

7.10.2.23. Ski



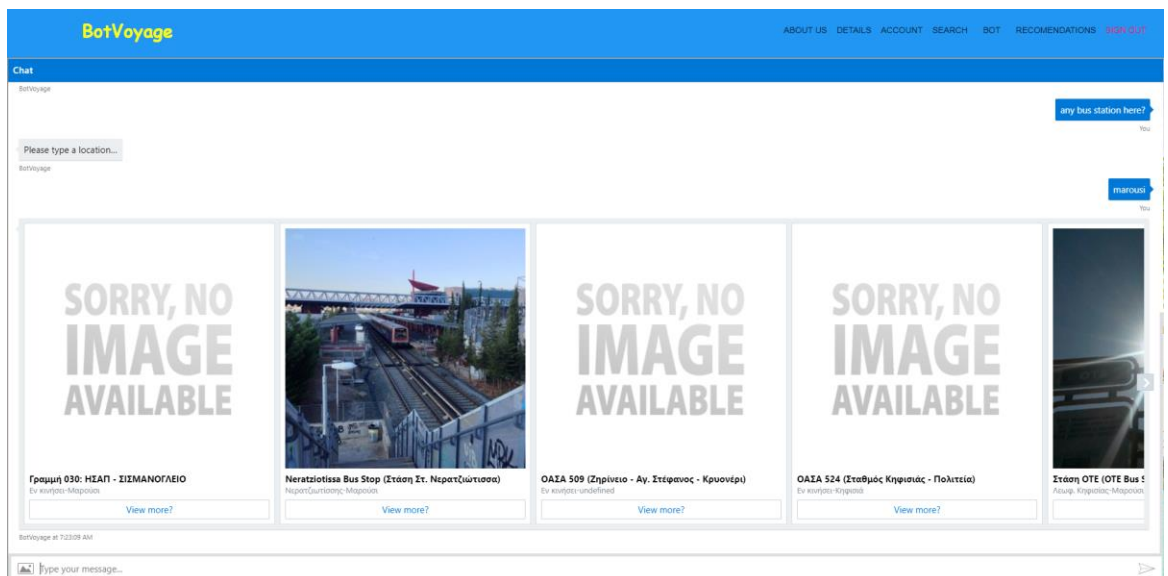
Εικόνα 149 - Αποτελέσματα Ski στην Ελλάδα - BotVoyage Platform - (δικό μας)

7.10.2.24. Train



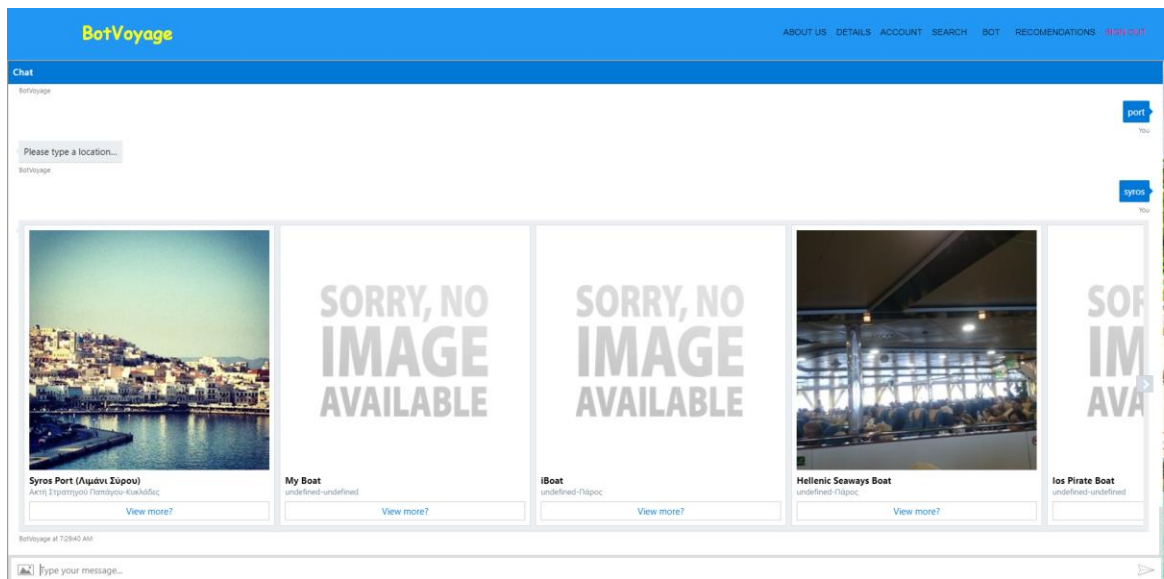
Εικόνα 150 - Αποτελέσματα Train στην Αθήνα - BotVoyage Platform - (δικό μας)

7.10.2.25 Bus



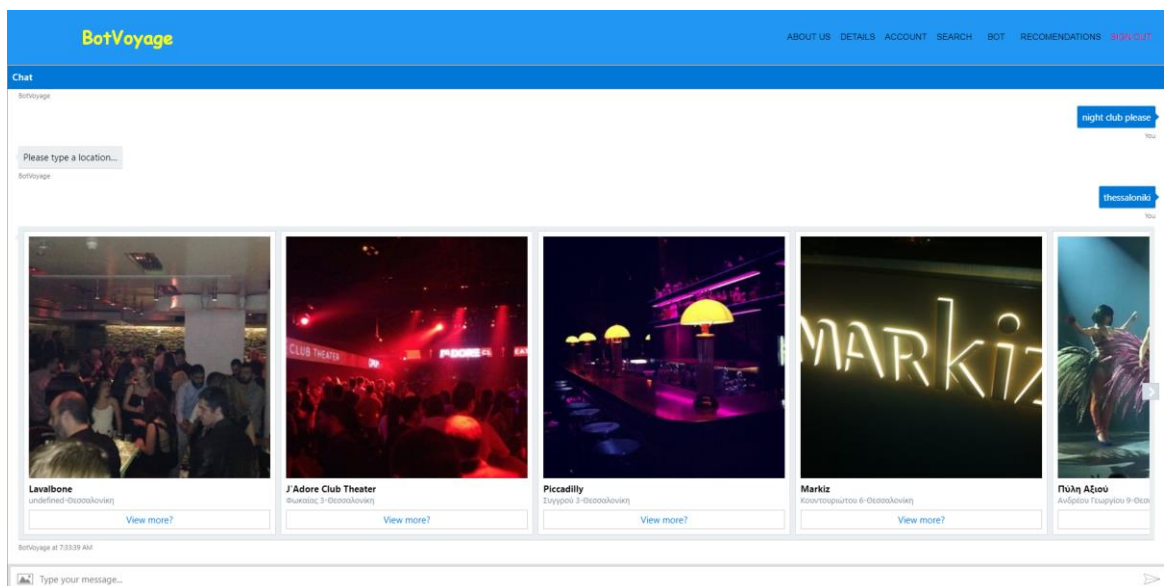
Εικόνα 151 - Αποτελέσματα Bus στο Μαρούσι - BotVoyage Platform - (δικό μας)

7.10.2.26. Port (Boat-Ship)



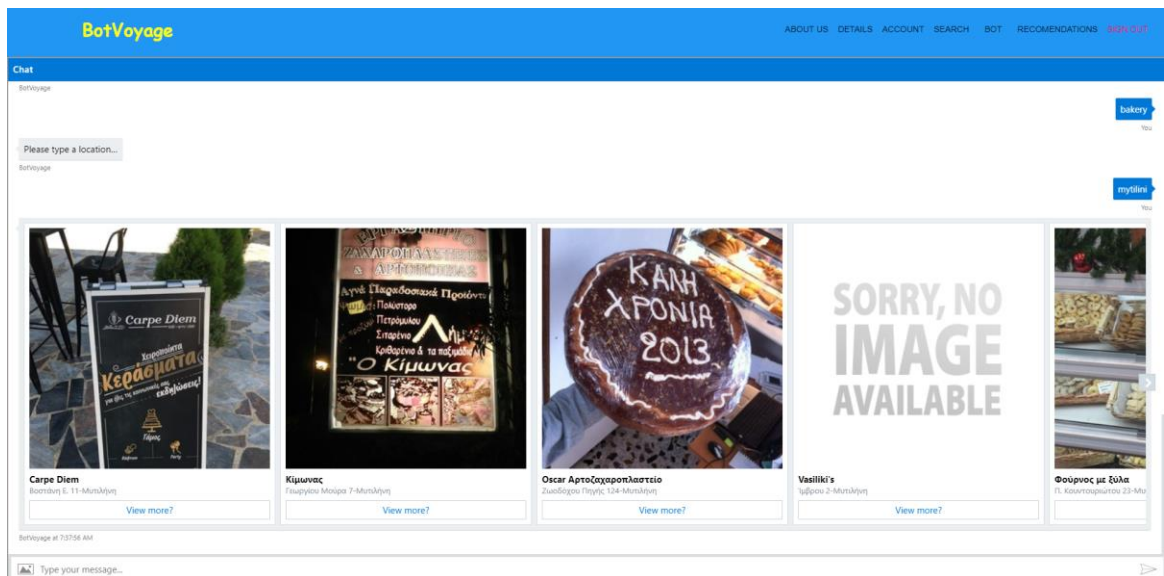
Εικόνα 152 - Αποτελέσματα Port στην Σύρο - BotVoyage Platform - (δικό μας)

7.10.2.27. Club



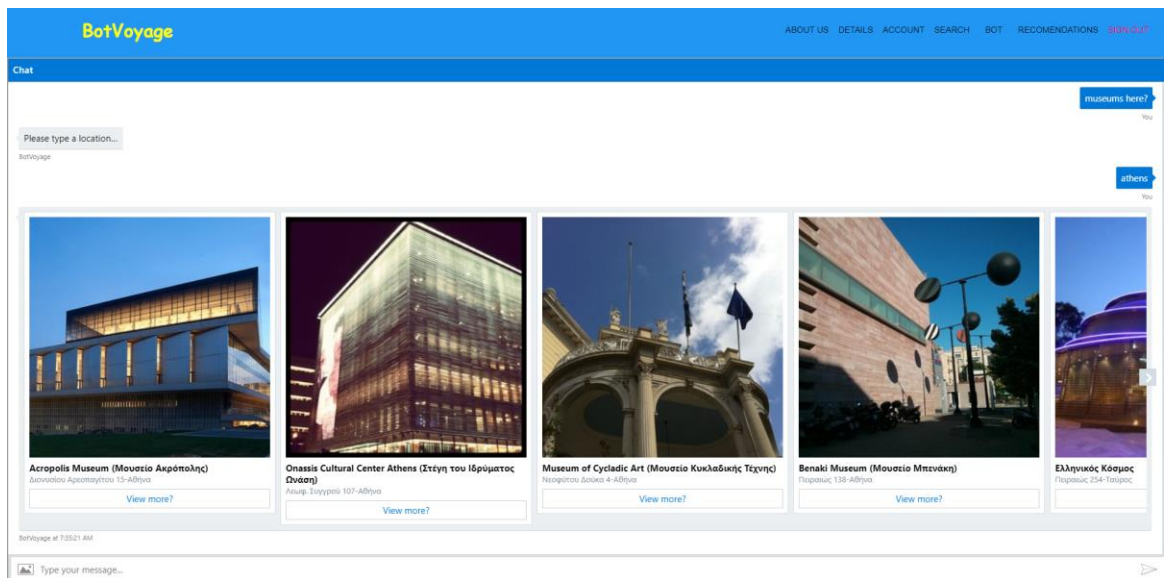
Εικόνα 153 - Αποτελέσματα Club στην Θεσσαλονίκη - BotVoyage Platform - (δικό μας)

7.10.2.28. Bakery



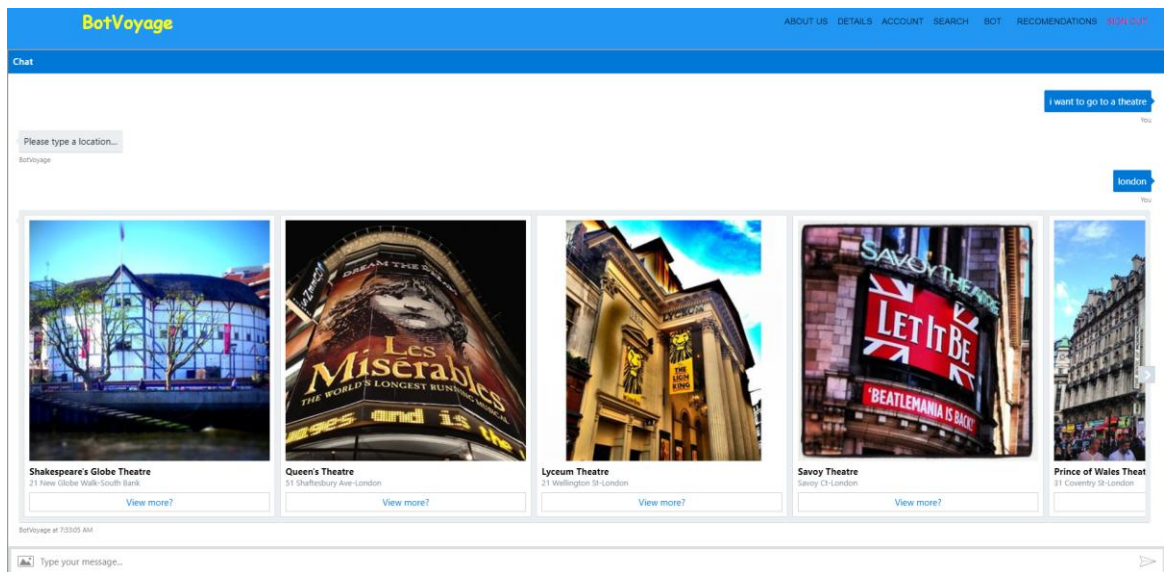
Εικόνα 154 - Αποτελέσματα Bakery στην Μυτιλήνη - BotVoyage Platform - (δικό μας)

7.10.2.29. Museum



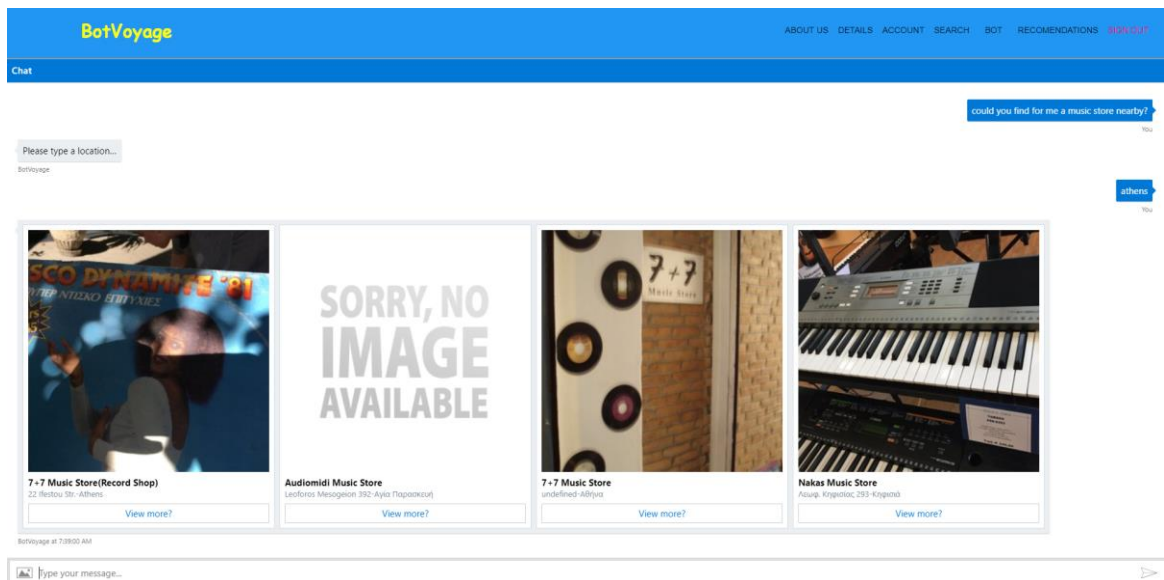
Εικόνα 155 - Αποτελέσματα Museum στην Αθήνα - BotVoyage Platform - (δικό μας)

7.10.2.30. Theatre



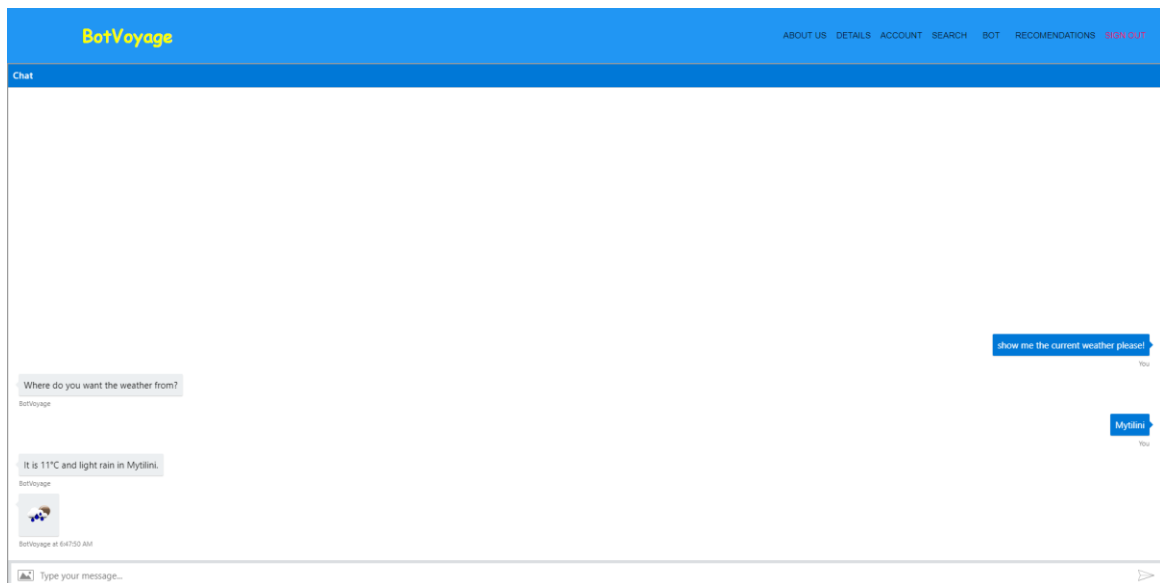
Εικόνα 156 - Αποτελέσματα Theatre στο Λονδίνο - BotVoyage Platform - (δικό μας)

7.10.2.31. Music Stores



Εικόνα 157 - Αποτελέσματα Music Stores στην Αθήνα - BotVoyage Platform - (δικό μας)

7.10.2.32. Weather



Εικόνα 158 - Αποτελέσματα Weather στην Μυτιλήνη - BotVoyage Platform - (δικό μας)

Σημείωση: Αξίζει να σημειωθεί, πως όλες οι κατηγορίες που αναφέρθηκαν παρέχονται και μέσω της ελληνικής γλώσσας. Το μποτ έχει εκπαιδευτεί στο να παρέχει αποτελέσματα με την κατανόηση μιας ελληνικής λέξης αλλά και φράσης.

7.11 Σελίδα Recommendations

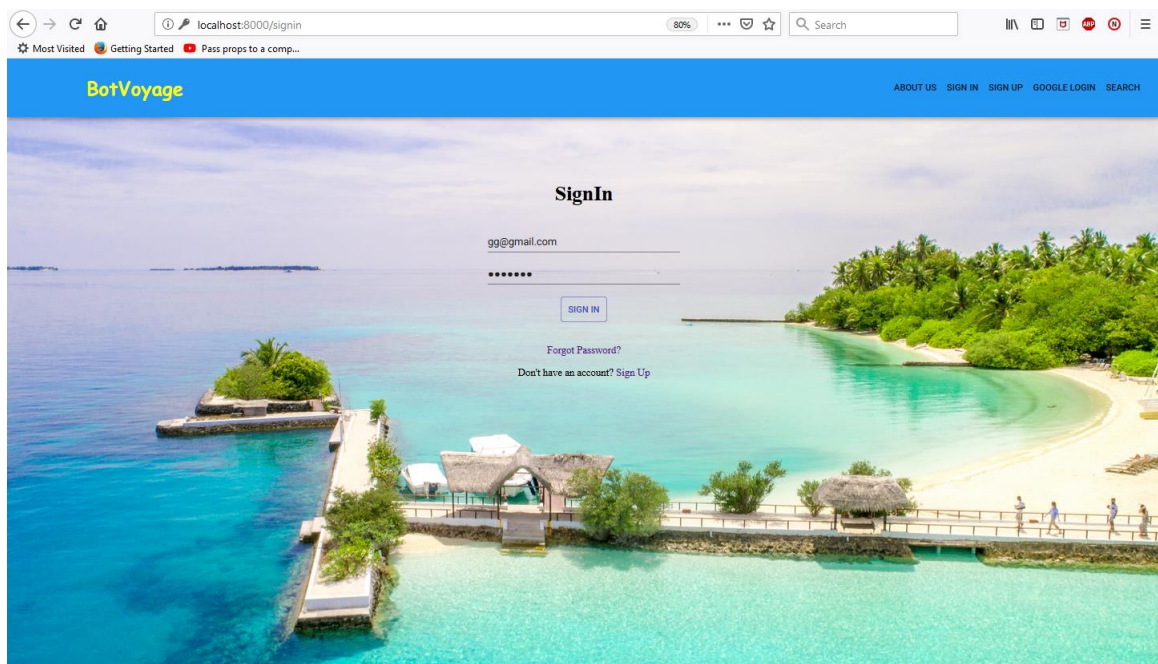
Πατώντας το κουμπί Recommendations (στο Menubar) της εφαρμογής, μας ο χρήστης μπορεί να δει σε καρουζέλ, τις επικρατέστερες έξι αναζητήσεις στην πλατφόρμα μας από άλλους χρήστες. Αυτό μπορεί να χρησιμοποιηθεί και για μελλοντική επέκταση σε περίπτωση συνεργασίας με επιχειρήσεις για διαφήμιση στον ιστότοπό μας.



Εικόνα 159 - Σελίδα Recommendations - BotVoyage Platform - (δικό μας)

7.12 Κουμπί SIGN OUT

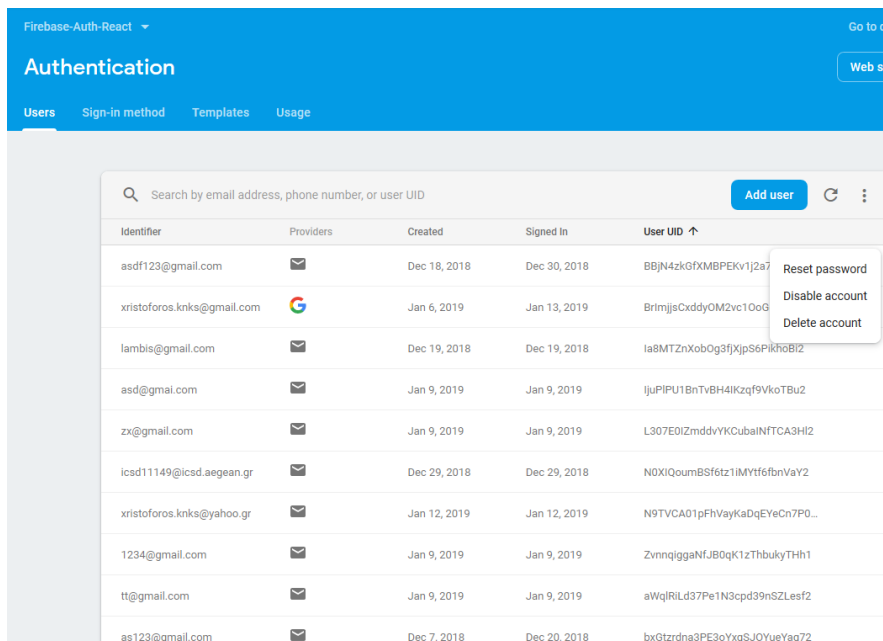
Σε περίπτωση ενεργοποίησης του κουμπιού SIGN OUT από τον χρήστη, αυτός θα μεταφερθεί στην σελίδα SIGN IN της πλατφόρμας μας όπως είναι φυσικό.



Εικόνα 160 - Σελίδα SignIn - BotVoyage Platform - (δικό μας)

7.13 Βάση δεδομένων firebase

Με την χρήση αυτής της βάσης έχουμε τον έλεγχο των logins της πλατφόρμας μας σε ένα γραφικό περιβάλλον που μας παρέχεται από τις υπηρεσίες του firebase. Συγκεκριμένα, μπορούμε να δούμε όλα τα logins που πραγματοποιήθηκαν στην πλατφόρμας μας είτε αυτά ήταν μέσω την μεθόδου sign in, είτε μέσω της αυθεντικοποίησης google. Το firebase μας παρέχει ιστορικό με τα logins χρησιμοποιώντας τις ημερομηνίες που οι χρήστες έκαναν sign up, sign in καθώς και της εμφάνισης του userid του κάθε χρήστη σε ένα hash όπως φαίνεται παρακάτω. Επίσης, στην στήλη providers μπορούμε να δούμε την μέθοδο με την οποία συνδέθηκε ο χρήστης. Ως διαχειριστές της πλατφόρμας, το firebase, μας δίνει την δυνατότητα να αλλάξουμε τον κωδικό κάποιου χρήστη, να απενεργοποιήσουμε τον λογαριασμό του, ή ακόμα και να τον διαγράψουμε.



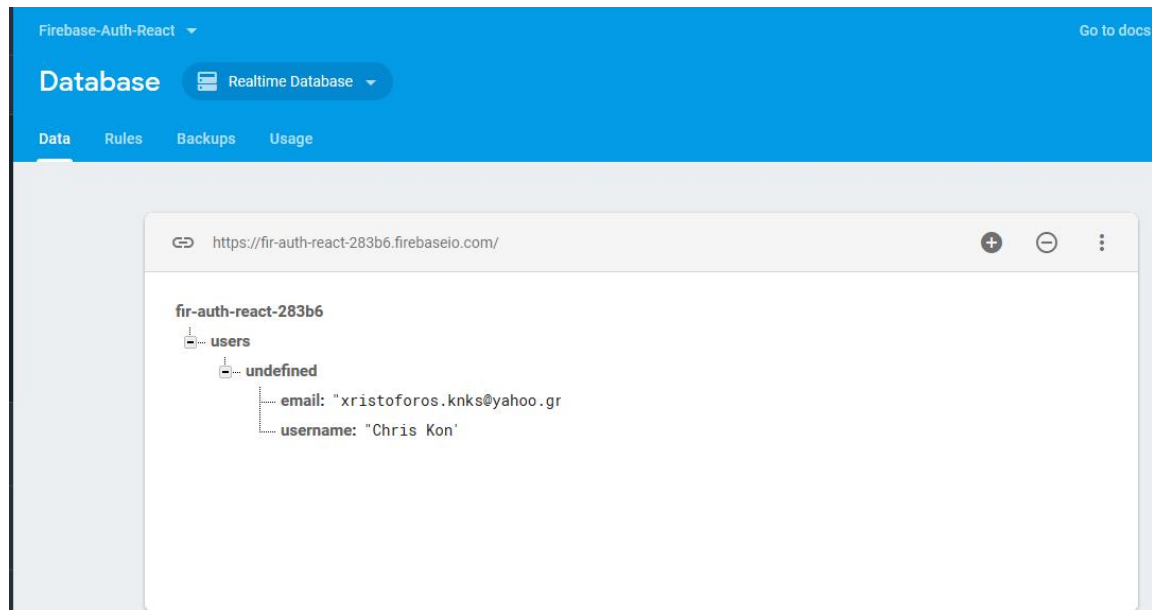
Identifier	Providers	Created	Signed in	User UID
asdf123@gmail.com	✉	Dec 18, 2018	Dec 30, 2018	BBjN4zkGfXMBPEKv1J2a7
xristoforos.knks@gmail.com	🌐	Jan 6, 2019	Jan 13, 2019	BrimjjsCxddyOM2vc1OoG
lambis@gmail.com	✉	Dec 19, 2018	Dec 19, 2018	la8MTZnxXobOg3fXjpS6Pikf0BIZ
asd@gmail.com	✉	Jan 9, 2019	Jan 9, 2019	IjuPIFU1BnTVBH4IKzqf9VkoTBu2
zx@gmail.com	✉	Jan 9, 2019	Jan 9, 2019	L307E0IZmddvYKCubaINFTCA3HI2
icsd11149@icsd.aegean.gr	✉	Dec 29, 2018	Dec 29, 2018	N0XiQoumBSf6tz1IMYtf6bnVaY2
xristoforos.knks@yahoo.gr	✉	Jan 12, 2019	Jan 12, 2019	N9TVCA01pFhVayKaDqEYeCn7P0...
1234@gmail.com	✉	Jan 9, 2019	Jan 9, 2019	ZvnnqiggaNFJB0qK1zThbukyTHh1
th@gmail.com	✉	Jan 9, 2019	Jan 9, 2019	aWqjRiLd37Pe1N3cpd39nSZLesf2
as123@gmail.com	✉	Dec 7, 2018	Dec 20, 2018	bxGtzrdna3PE3oYxgSJOYueYag72

Εικόνα 161 - Βάση firebase με τα logins - (δικό μας)

7.13.1 Δημιουργία χρήστη στην βάση μας με την μέθοδο sign up.

Όταν ένας χρήστης κάνει εγγραφή στην πλατφόρμα μας τα στοιχεία του αποθηκεύονται στην Realtime Database του firebase στο collection users σε μορφή json και έπειτα παίρνουμε το username του μέσα από αυτό το collection και το εμφανίζουμε στην σελίδα details της πλατφόρμας μας, ως τον τελευταίο χρήστη που έκανε εγγραφή.

Όπως αναφέρθηκε και προηγουμένως, αυτό το collection της βάσης μας, ανανεώνεται κάθε φορά που ένας καινούργιος χρήστης κάνει εγγραφή.



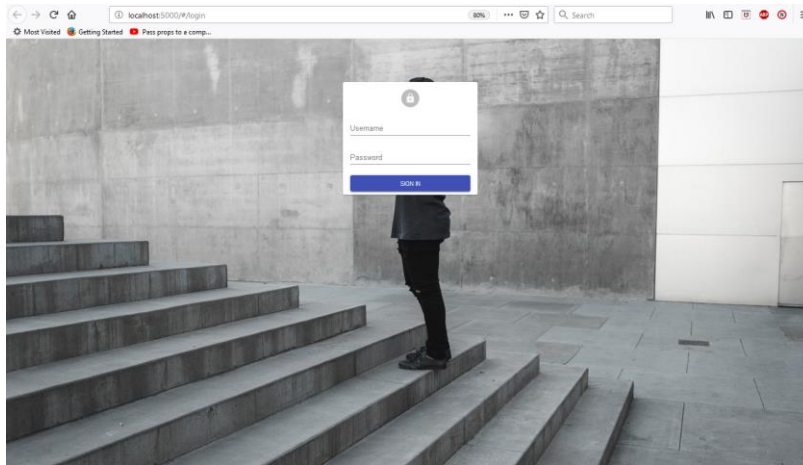
Εικόνα 162 - Users Collection στο Firebase - (δικό μας)

7.14 Λειτουργίες admin dashboard

Ακόμα, πέρα από την πλατφόρμα μας η οποία περιλαμβάνει το γενικό user dashboard κάποιου χρήστη δημιουργήσαμε και ένα admin dashboard το οποίο είναι συνδεδεμένο με την βάση μας και μπορεί κάποιος admin να διαχειρίζεται τους χρήστες της πλατφόρμας από αυτό, αλλά δεν το ενώσαμε με την πλατφόρμα μας καθώς το υλοποιήσαμε για τυχόν μελλοντικές επεκτάσεις στην πλατφόρμα μας, όπως η ανάρτηση posts των users οι οποίοι έχουν δικαιώματα admin στην πλατφόρμας. Παρόλα αυτά, το παρουσιάζουμε εδώ, καθώς υλοποιήθηκε και θα ήτανε μια πολύ καλή λύση όσων αφορά τις μελλοντικές επεκτάσεις.

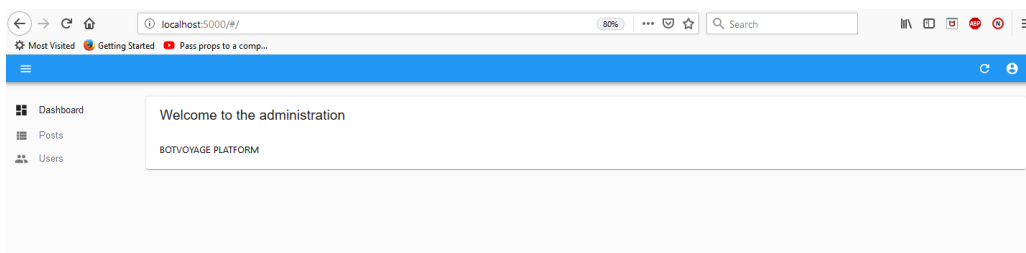
7.14.1 Σύνδεση του χρήστη στο admin dashboard

Αρχικά, ο χρήστης συνδέεται σαν admin στο admin dashboard της πλατφόρμας με τα στοιχεία του,



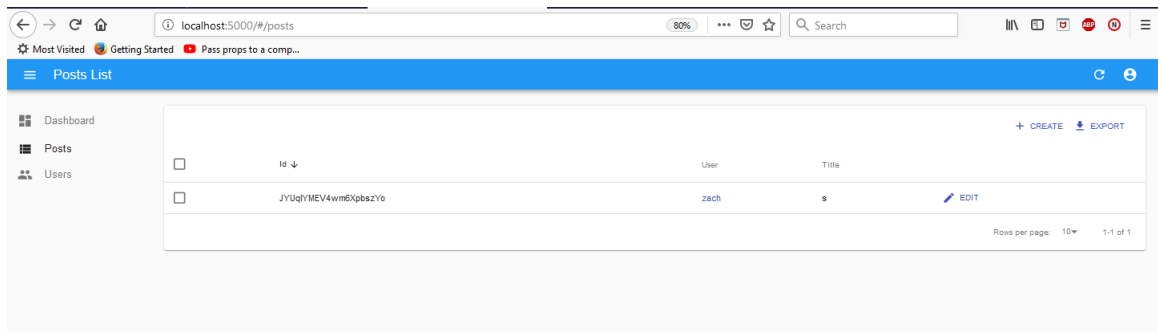
Εικόνα 163 - Σελίδα SignIn - AdminDashboard - (δικό μας)

Έπειτα, εμφανίζεται ένα μήνυμα καλωσορίσματος στον χρήστη που έκανε sign in.



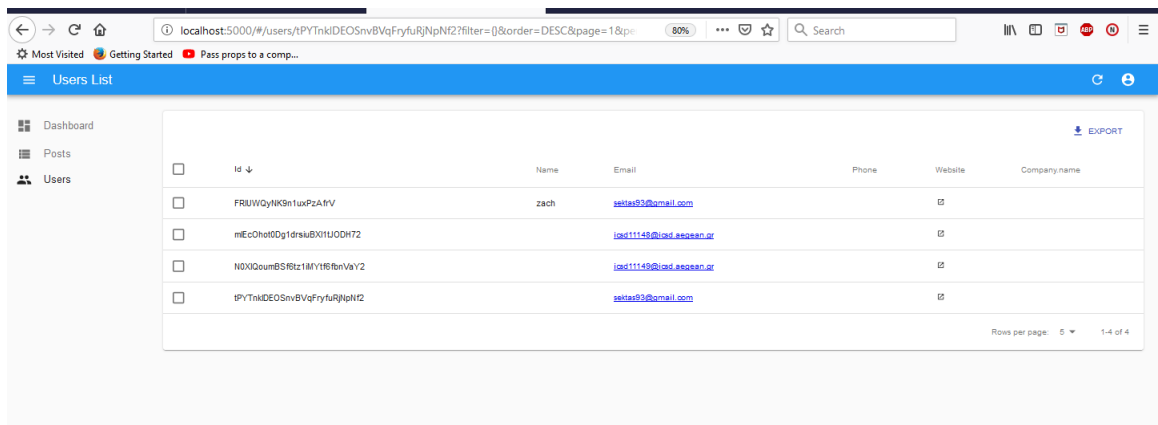
Εικόνα 164 - Είσοδος στο AdminDashboard - (δικό μας)

Για την υλοποίηση του admin dashboard, συνδέσαμε την εφαρμογή μας με την βάση Cloud Firestore του Firebase καθώς επιθυμούσαμε να έχουμε μια βάση για το user dashboard και μια βάση για το admin dashboard, έτσι ώστε να μην υπάρχει σύγχυση μεταξύ των δεδομένων των 2 βάσεων. Πηγαίνοντας λοιπόν στην επιλογή posts ο admin, μπορεί να δει posts που έχουν γίνει από άλλους χρήστες admin σε παλαιότερο χρονικό διάστημα. Η συλλογή posts εμφανίζει ένα id από ένα έγγραφο το οποίο είναι αποθηκευμένο σε ένα collection στο firebase.



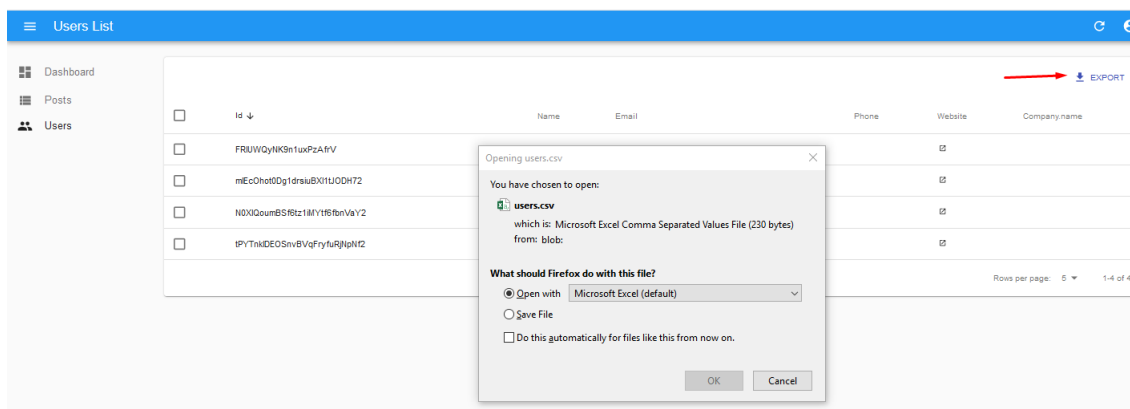
Εικόνα 165 - Posts στο AdminDashboard - (δικό μας)

Στην επιλογή users, ο χρήστης admin μπορεί να δει άλλους χρήστες admin που είναι αποθηκευμένοι στο firebase. Η διαδικασία με τα id είναι η ίδια με αυτή των posts που περιγράφηκε παραπάνω.



Εικόνα 166 - Users στο AdminDashboard - (δικό μας)

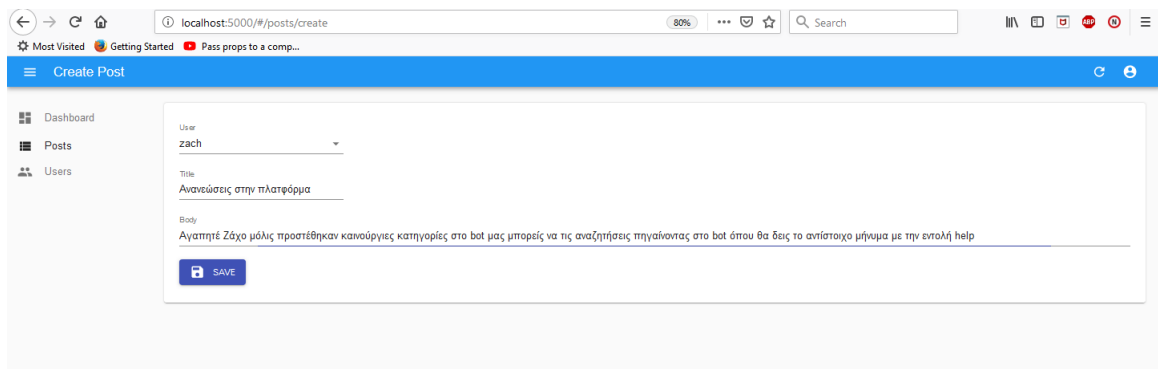
Οι χρήστες admin έχουν την δυνατότητα να κατεβάσουν σε μορφή αρχείου excel τα στοιχεία που είναι αποθηκευμένα στο admin dashboard και ταυτόχρονα στο cloud firestore μέσω του κουμπιού EXPORT πάνω δεξιά.



Εικόνα 167 - Προτροπή για download σε Excel - AdminDashboard - (δικό μας)

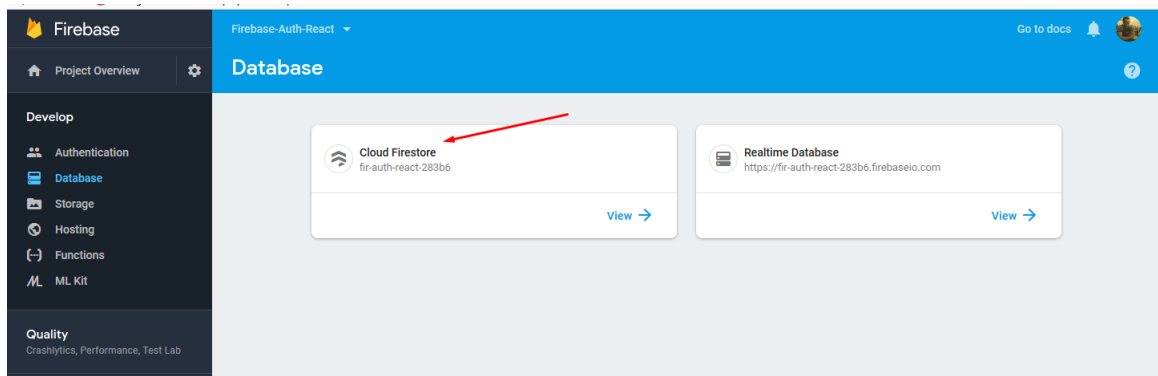
7.14.2 Δημιουργία καινούργιου post

Πατώντας το κουμπί create ο admin μπορεί να δημιουργήσει κάποιο καινούργιο post για έναν συγκεκριμένο χρήστη της πλατφόρμας. Για παράδειγμα, ας πούμε ότι στο μέλλον προσθέταμε καινούργιες κατηγορίες στο bot μας και θέλαμε να ενημερώσουμε τον χρήστη Zach γι' αυτήν την αλλαγή μέσω ενός μηνύματος.

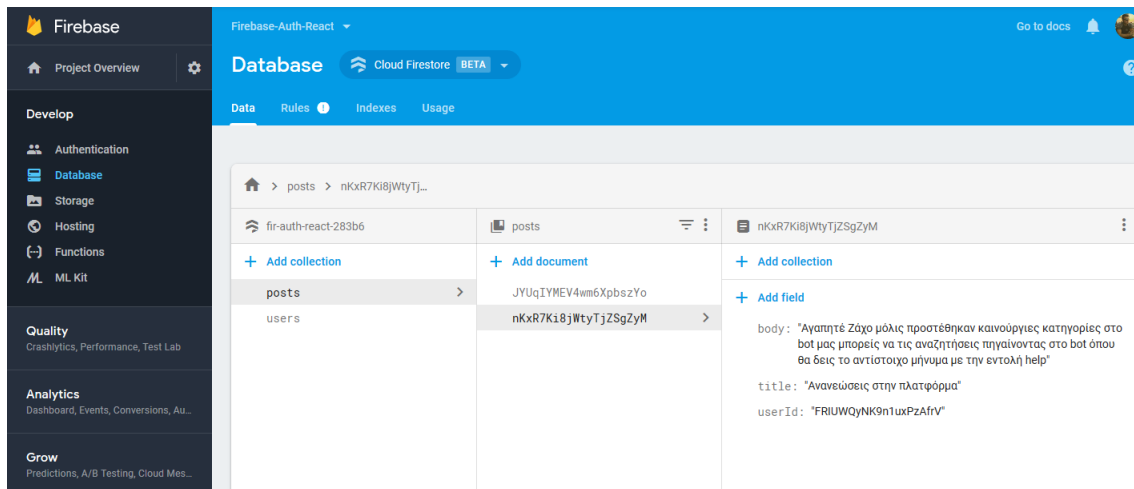


Εικόνα 168 - Δημιουργία καινούργιου Post - AdminDashboard - (δικό μας)

Όταν ο admin πατήσει το κουμπί save αυτό θα αποθηκευτεί στην βάση Cloud Firestore στο Firebase σε μορφή JSON όπως φαίνεται παρακάτω

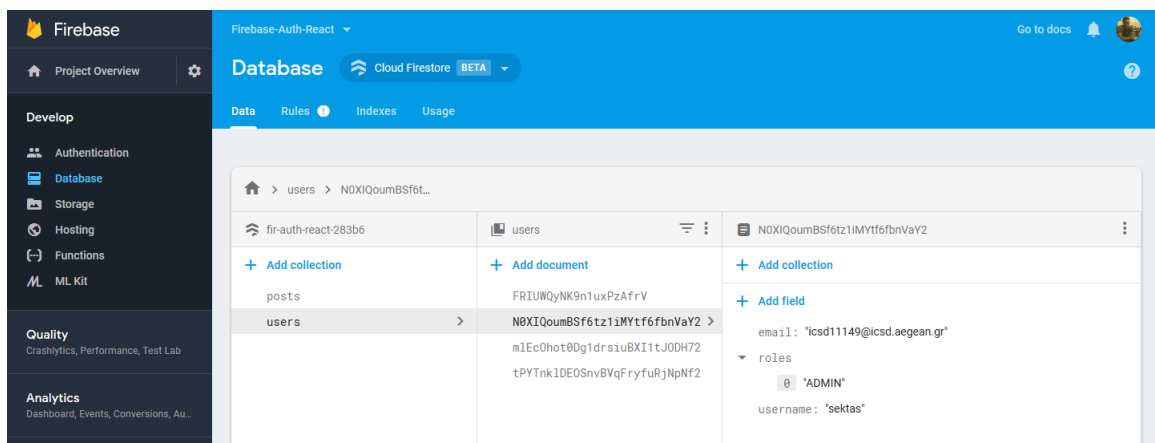


Εικόνα 169 - Βάση Cloud Firestore στο Firebase - (δικό μας)



Εικόνα 170 - Posts Collection στο Cloud Firestore - (δικό μας)

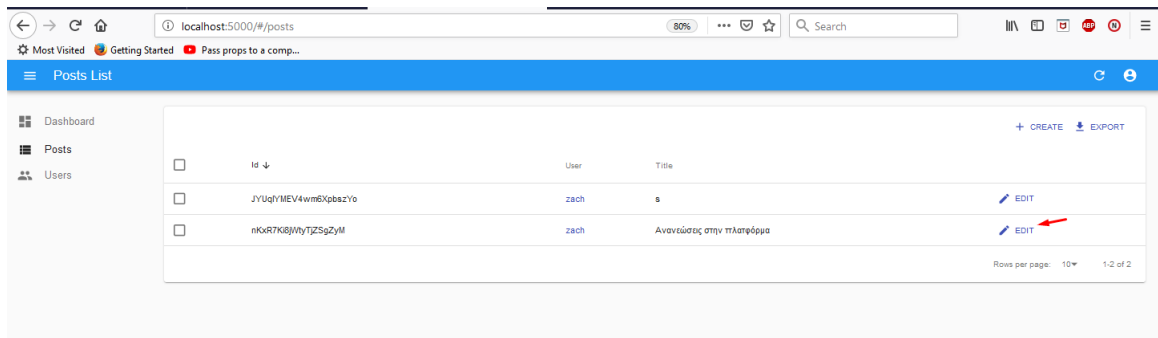
Επίσης, στο collection users μπορούμε να δούμε τους admin χρήστες που έχουν αποθηκευτεί στο firebase.



Εικόνα 171 - Users Collection στο Cloud Firestore - (δικό μας)

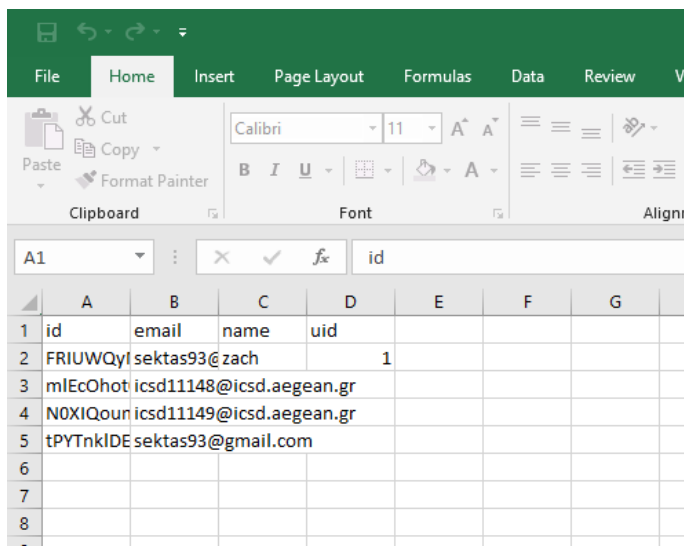
7.14.3 Τροποποίηση και διαγραφή του post

Εφόσον κάποιος χρήστης μπορεί να δημιουργήσει ένα post τα επόμενα βήματα θα ήταν να μπορεί να τροποποιήσει κάποια posts και επίσης να τα διαγράψει. Αυτές οι δύο λειτουργίες προστέθηκαν επίσης στο admin dashboard.



Εικόνα 172 - Λίστα Posts - AdminDashboard - (δικό μας)

Τέλος, όπως και στο collection users προστέθηκε η επιλογή εξαγωγή των posts που υπάρχουν στην βάση σε excel αρχείο το οποίο αν το ανοίξουμε έχει την παρακάτω μορφή.



Εικόνα 173 - Αρχείο Excel με τα Collections από το AdminDashboard - (δικό μας)

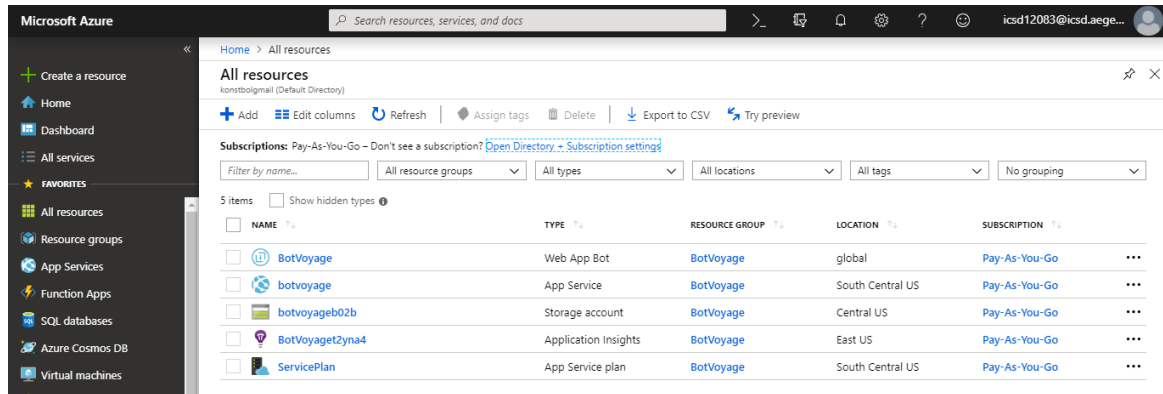
7.14.4 Σύνοψη

Η πλατφόρμα admin dashboard, υλοποιήθηκε όπως αναφέραμε και πριν για μελλοντική χρήση, καθώς παρουσιάστηκαν προβλήματα μετά την υλοποίησή της όσον αφορά την λήψη δεδομένων από το cloud firestore στην πλατφόρμα BotVoyage καθώς δεν ήταν το ίδιο εύκολο με την λήψη δεδομένων από την βάση realtime database στην πλατφόρμα BotVoyage το οποίο και υλοποιήθηκε.

7.15 Απεικόνιση και ανάλυση διαχειριστικών περιβαλλόντων

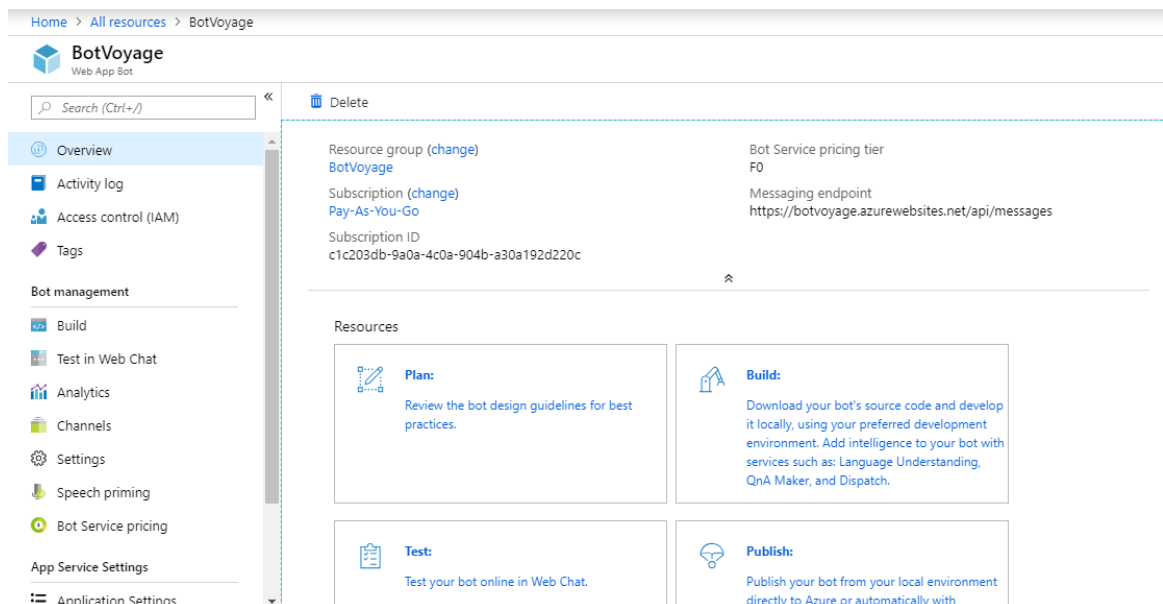
7.15.1 Azure Portal

All Resources Tab



Εικόνα 174 - Azure Portal - All resources - (δικό μας)

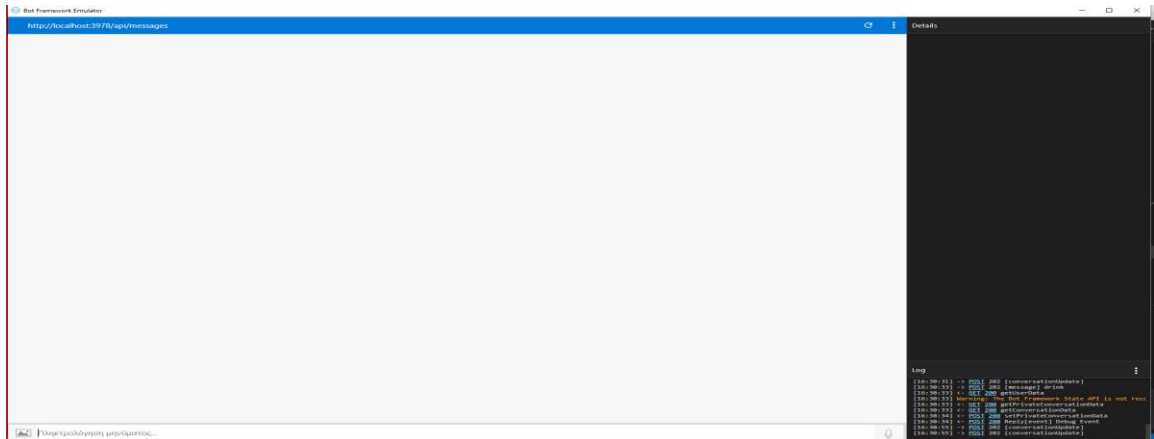
Παρατηρείται στην κάτω δεξιά γωνία το resource botvoyage. Ας το ανοίξουμε για να δούμε το περιβάλλον του συγκεκριμένου bot web app. Ειδικότερα, βλέπουμε το πλήθος των επιλογών που έχουμε ως διαχειριστές, όπως τον online editor και η διασύνδεση με διάφορα κανάλια, επιλογές τις οποίες τις έχουμε αναλύσει στο κεφάλαιο 6.



Εικόνα 175 - Azure Portal - Overview - BotVoyage - (δικό μας)

7.15.2 Bot Framework Emulator

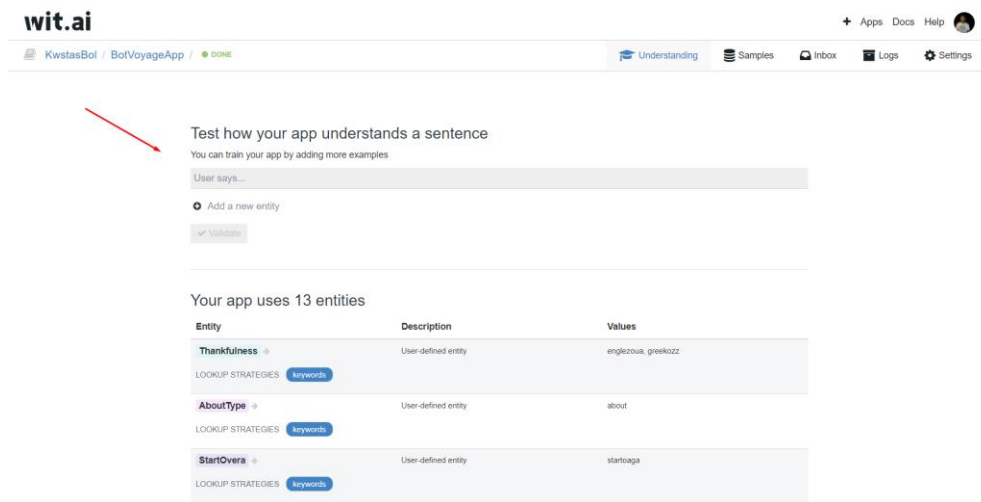
Στην είσοδο που βάζουμε το url, διαμορφώνουμε τη σύνδεση με το node.js, ώστε να μπορούμε να τεστάρουμε την εφαρμογή μας πριν δημοσιευτεί στο διαδίκτυο. Κάτω δεξιά παρατηρούμε τα logs, που μας διευκρινίζουν αν κάτι απέκλινε από τη φυσιολογική λειτουργία του.



Εικόνα 176 - Bot Framework Emulator - BotVoyage - (δικό μας)

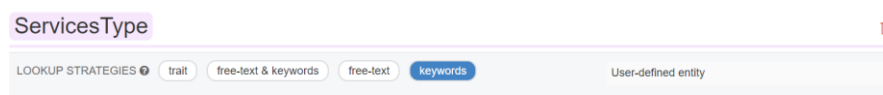
7.15.3 Wit.ai

Το Wit.ai είναι μια διεπαφή φυσικής γλώσσας για εφαρμογές ικανές να μετατρέπουν προτάσεις σε δομημένα δεδομένα. Με αυτόν τον τρόπο, μας είναι εύκολο να δημιουργούμε bot που μπορούν να αλληλεπιδρούν με τους ανθρώπους σε πλατφόρμες ανταλλαγής μηνυμάτων. Φτιάχνοντας αυτά τα bot, οι χρήστες έχουν την δυνατότητα να γράφουν ή ακόμα και να μιλάνε σε αυτά. Το σημαντικότερο κομμάτι του Wit και το πιο ενδιαφέρον είναι ότι πως με κάθε αλληλεπίδραση τα bot γίνονται εξυπνότερα. Παρακάτω φαίνεται το κεντρικό μενού της εφαρμογής BotVoyageApp.



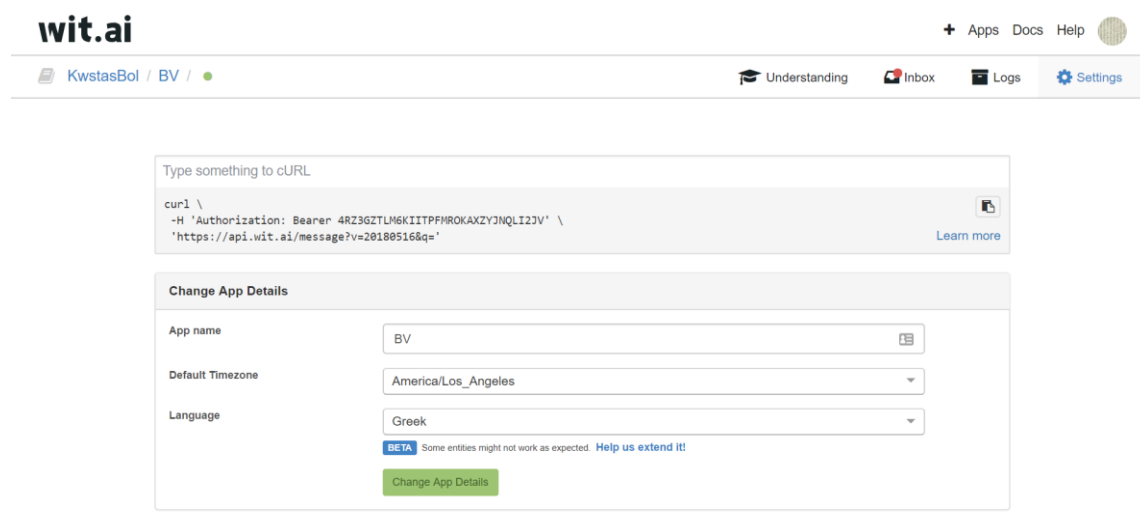
Εικόνα 177 - Wit.ai - BotVoyage - (δικό μας)

Στο textfield (User says...) γράφουμε ως διαχειριστές μία πρόταση για επικύρωση σε σχέση με μία οντότητα (entity). Στο κάτω μέρος βλέπουμε τις οντότητες που χρησιμοποιεί η εφαρμογή μας. Ας δούμε και το γραφικό περιβάλλον μιας οντότητας στην παρακάτω εικόνα.



Εικόνα 178 - Wit.ai - ServicesType - BotVoyage - (δικό μας)

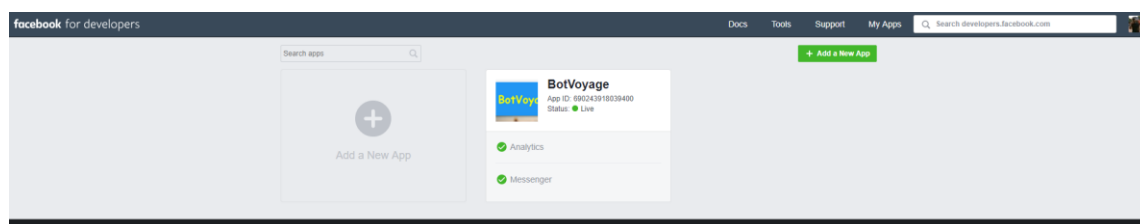
Στην παρακάτω εικόνα, φαίνεται η καρτέλα ρυθμίσεις, στην οποία μπορούμε να ρυθμίσουμε διάφορες παραμέτρους, όπως για παράδειγμα τη γλώσσα της εφαρμογής, βάση της οποίας πραγματοποιείται το NLU (Greek).



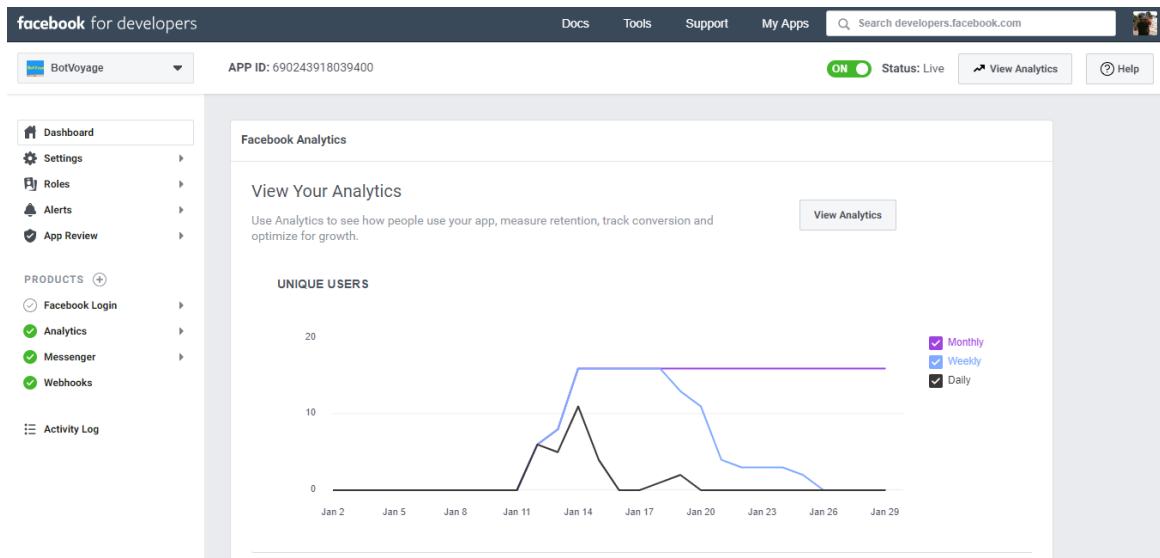
Εικόνα 181 - Wit.ai - Settings - BotVoyage - (δικό μας)

7.15.4 Facebook for Developers Platform

Σε πρώτο πλαίσιο, ας διακρίνουμε το γενικότερο dashboard ενός developer, κι ειδικότερα στο συγκεκριμένο στιγμιότυπο φαίνονται οι εφαρμογές που μπορείς να διαχειριστείς.



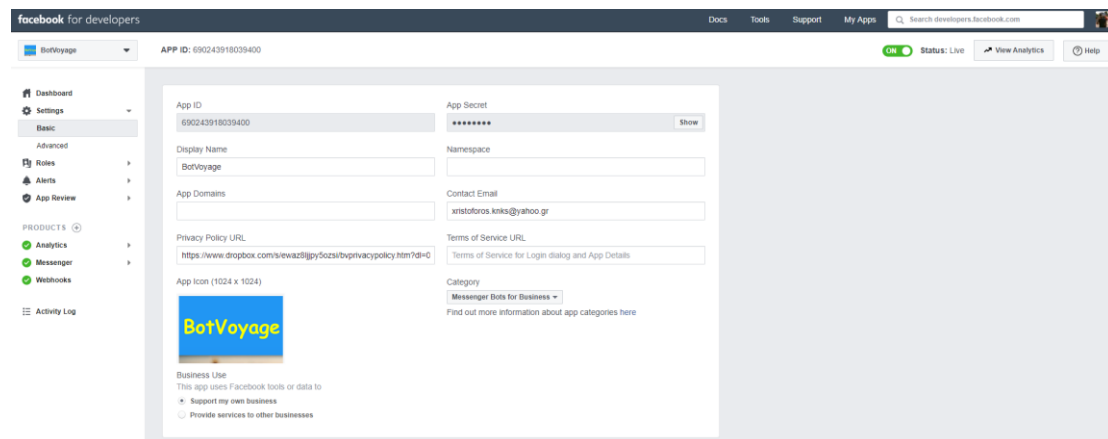
Εικόνα 183 - Facebook For Developers - BotVoyage - (δικό μας)



Εικόνα 183 - Facebook for Developers - Analytics - BotVoyage - (δικό μας)

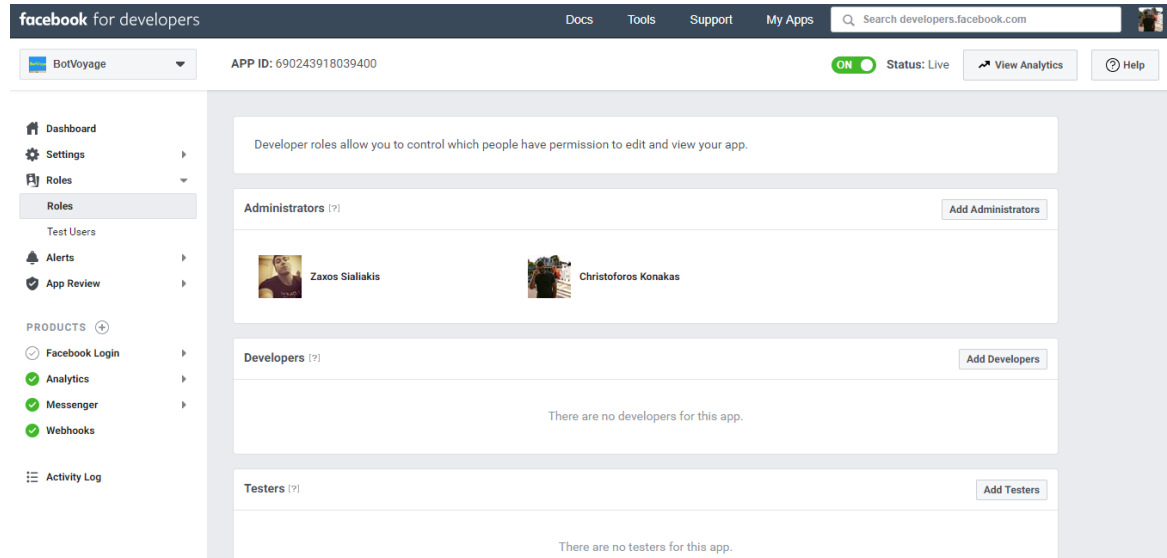
Στην παραπάνω εικόνα βλέπουμε το dashboard της εφαρμογής μας, που όπως φαίνεται αριστερά έχουμε διάφορες επιλογές. Ας δούμε κάποιες επιλογές ενδεικτικά.

Παρακάτω μπορούμε να επιλέξουμε το λογότυπο, το όνομα, τον κωδικό, ακόμη και το έγγραφο που περιλαμβάνει το privacy policy



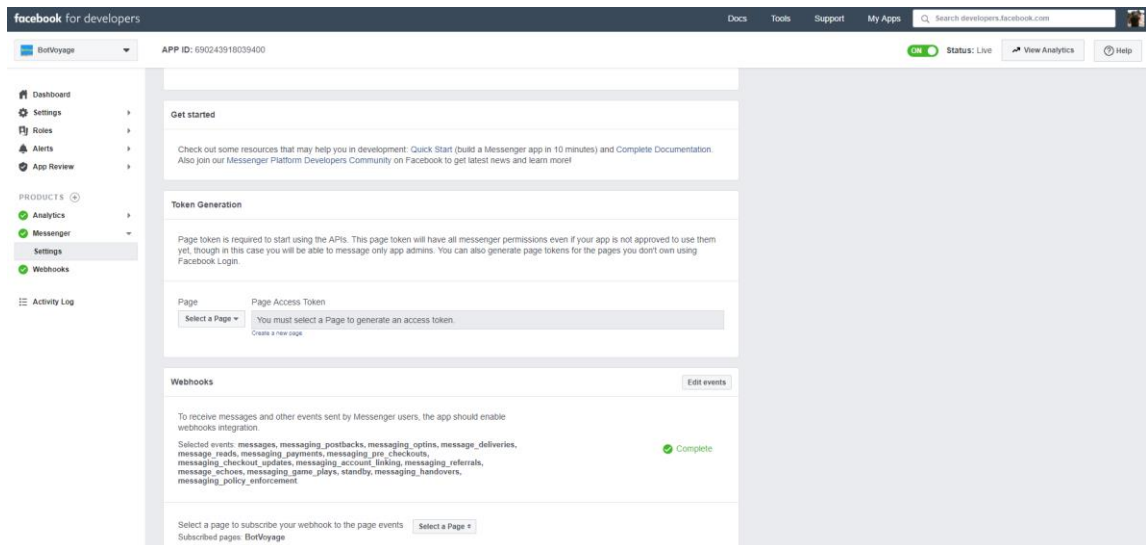
Εικόνα 184 - Facebook for Developers - Settings - BotVoyage - (δικό μας)

Στην παρακάτω εικόνα βλέπουμε το γραφικό περιβάλλον, μέσω του οποίου μπορούμε να ορίσουμε τα άτομα που θα ασχολούνται με την εφαρμογή είτε αυτά είναι διαχειριστές, προγραμματιστές και ούτω καθεξής.



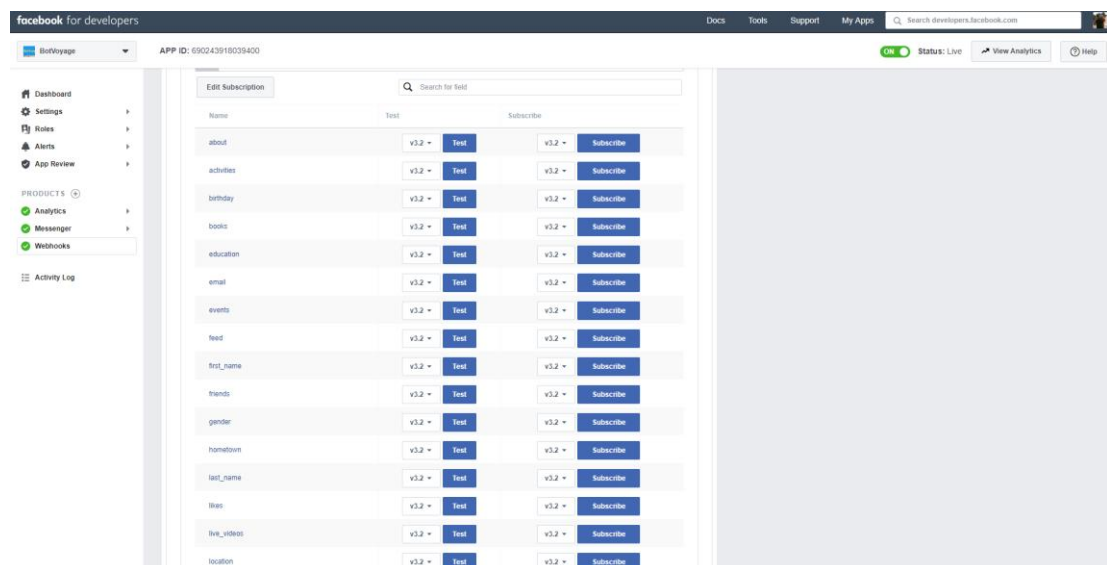
Εικόνα 185 - Facebook for Developers - Roles - BotVoyage - (δικό μας)

Αξίζει επίσης να σημειωθεί η καρτέλα products και συγκεκριμένα η Messenger -> Settings, στην οποία ουσιαστικά μπορείς να επιτελέσεις διάφορες λειτουργίες. Αρχικά, απαραίτητο βήμα είναι στο Token Generation, να επιλέξεις τη σελίδα που θες, για να πάρεις το απαραίτητο token και να το συνδέσεις με την πλευρά του azure portal. Επιπρόσθετα, στο σημείο που διαφαίνεται η έννοια Webhooks, μπορείς να δεις τι ακριβώς μπορεί να κάνει. Για παράδειγμα με το Selected events: messages μπορεί πλέον ο χρήστης να στείλει μήνυμα στη σελίδα.



Εικόνα 186 - Facebook for Developers - Messenger - Settings - BotVoyage - (δικό μας)

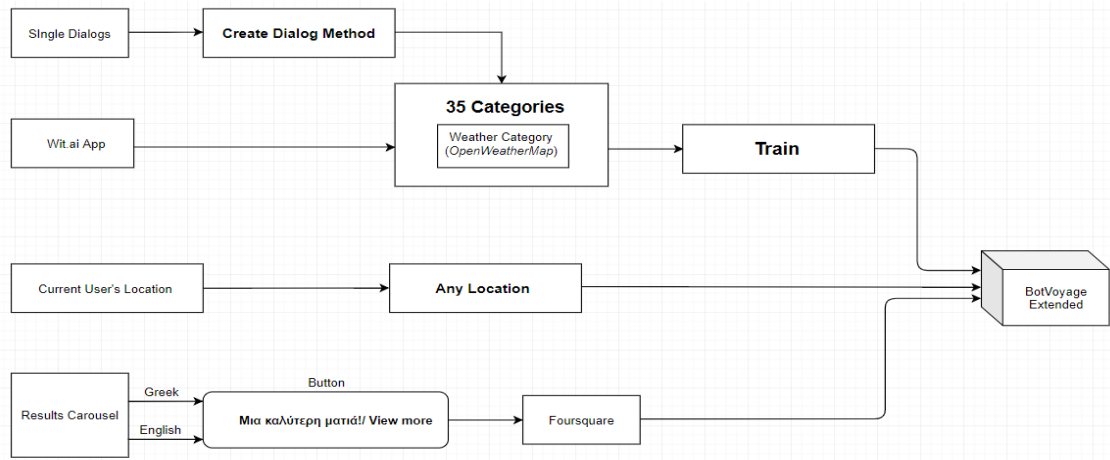
Τέλος, αξίζει να δούμε και την σελίδα των Webhooks που εκεί του ορίζει ο διαχειριστής τι μπορεί να κάνει ακριβώς η εφαρμογή διαμέσου της πλατφόρμας του Messenger (δηλαδή ότι βλέπει στα Settings, αλλάζει από την παρακάτω σελίδα)



Εικόνα 187 - Facebook for Developers - Webhooks - BotVoyage - (δικό μας)

ΚΕΦΑΛΑΙΟ 8 - ΣΥΜΠΕΡΑΣΜΑΤΑ - ΕΠΟΜΕΝΑ ΒΗΜΑΤΑ

8.1 Επέκταση BotVoyage



Αν μπορούμε να αναλογιστούμε ποια είναι τα κυριότερα στοιχεία τα οποία κρατήσαμε και επεκτείναμε από την προηγούμενη έκδοση του αυτοματοποιημένου λογισμικού BoVoyage, αυτά είναι η δομή ενός διαλόγου με την μέθοδο καταρράκτη, η εφαρμογή Wit.ai BotVoyageApp με αρκετά δεδομένα μέσα από εκπαίδευση, η ιδέα της αναζήτησης συγκεκριμένης τοποθεσίας από τους χρήστες και η παρουσίαση των δεδομένων, όπως για παράδειγμα η μορφή Carousel. Επίσης, αξίζει να σημειωθεί μια λειτουργία του Bot η οποία κρατήθηκε αλλά δεν επεκτάθηκε (επισημάνουμε στην επόμενη παραγράφους το γιατί) και είναι αυτή της αναζήτησης αεροπορικών δρομολογίων μέσω του Kayak API. Αναλυτικότερα, όπως διακρίνουμε στο παραπάνω σχήμα, μέσα από την δομή των διαλόγων, αναπτύχθηκε μέθοδος δημιουργίας διαλόγων ώστε μια κατηγορία (στην οποία εκπαιδεύεται το bot) να συνδέεται εύκολα με όλες τις υπόλοιπες με την χρήση κυριολεκτικά μιας γραμμής στον κώδικα. Στην εφαρμογή BotVoyageApp στο Wit.ai, εκπαιδεύσαμε το bot σε 27 νέες κατηγορίες συμπεριλαμβανομένης και της κατηγορίας του καιρού (μέσω του OpenWeatherMap API). Η εκπαίδευση έγινε με πολλές νέες λέξεις-κλειδιά, εκφράσεις, ερωτήσεις, ακόμα και με λανθασμένες πληκτρολογήσεις, σύμφωνα φυσικά με τις πιθανές επιθυμητές αναζητήσεις των χρηστών. Ακόμα, η ιδέα της αναζήτησης τοποθεσίας από τους χρήστες, αναπτύχθηκε για όλες τις πιθανές τοποθεσίες ανά τον κόσμο και όχι μόνο για την τωρινή τοποθεσία ενός χρήστη. Ένα τελευταίο κύριο στοιχείο το οποίο όπως προαναφέραμε αναπτύχθηκε, είναι η παρουσίαση των αποτελεσμάτων αλλά και η ενσωμάτωση και άλλων επιλογών σε εκείνα όπως π.χ. η μετάβαση του χρήστη στο περιβάλλον του Foursquare για περαιτέρω πληροφορίες εάν το επιθυμεί, με την χρήση του κουμπιού “View more!” κάτω από τα αποτελέσματα.

8.2 Συμπεράσματα επί των εργαλείων

Αρχικά, η γλώσσα προγραμματισμού που επιλέξαμε ήταν η **Javascript**. Η επιλογή αυτή δεν ήταν τυχαία, καθώς εκτός από τα θετικά χαρακτηριστικά που μας προσδίδει τα οποία αναλύθηκαν στο κεφάλαιο 4, έχουμε ξανασχοληθεί με αυτήν την γλώσσα προγραμματισμού στο παρελθόν. Κάποιοι άλλοι λόγοι στους οποίους στηρίξαμε την επιλογή μας, ήταν ότι είναι πολύ εύκολο εργαλείο στην εκμάθηση, δεν χρειάζεται setup καθώς η εφαρμογή μας προορίζεται για διαδικτυακή χρήση και η Javascript είναι ενσωματωμένη στον web browser. Επίσης, η Javascript χρησιμοποιείται παντού (React, Angular), εφαρμογές για κινητά, Internet of Things, ρομποτική, εικονική πραγματικότητα κ.ά. Ένα άλλο χαρακτηριστικό επιλογής, ήταν ότι η Javascript μας παρέχει το εργαλείο NodeJS που χρησιμοποιήσαμε, το οποίο διαθέτει πάνω 30.000 NPM πακέτα για ανάπτυξη διαδικτυακών εφαρμογών. Η δημιουργία δυναμικών ιστοσελίδων γίνεται ευκολότερη αν όπως αναφέραμε παραπάνω κάνουμε χρήση της βιβλιοθήκης **React**. Με την συγκεκριμένη βιβλιοθήκη, δεν χρειάστηκε να επαναλαμβάνουμε μπλοκ κώδικα ξανά και ξανά. Ήταν εύκολη στην εκμάθησή της και βελτίωσε σημαντικά την επιδοση της ιστοσελίδας-πλατφόρμας που δημιουργήσαμε. Κι αυτό, λόγω του Virtual DOM, το οποίο λειτουργεί σαν δομή δέντρου και αναπαρηστά κάθε κόμβο σαν ένα μέρος του αρχείου. Επίσης, ένα άλλο θετικό στοιχείο είναι πως το community της βιβλιοθήκης React αυξάνεται ραγδαία, πράγμα το οποίο μαζί με την δημιουργία ενός πλαισίου εφαρμογής στο οποίο οι προγραμματιστές μπορούν να τεστάρουν και να ελέγχουν τον κώδικά τους, βοηθάει εξαιρετικά. Ακόμη, η ύπαρξη των εργαλείων React Developer Tools δίνει μεγάλη ώθηση για προγραμματισμό native εφαρμογών στους developers. Η βιβλιοθήκη της React είναι πλούσια και έτσι η δημιουργία SEO friendly ιστοσελίδων, όπως η δική μας, έγινε χωρίς πολλές δυσκολίες και σε αξιόλογο χρονικό περιθώριο. Όσον αφορά τα δεδομένα της εφαρμογής μας, το **Foursquare** API μας έδωσε τη δυνατότητα να έχουμε άμεση πρόσβαση στα δεδομένα που επιθυμούσαμε χωρίς κάποιο περιορισμό στη χρήση του που θα μας εμπόδιζε. Επίσης, μιας και είναι τόσο εδραιωμένη η χρήση του Foursquare, τα δεδομένα ήταν αξιόπιστα και αφορούσαν όλη την υδρόγειο, οπότε και η εφαρμογή μας μπορούσε να επεκταθεί, κλιμακωθεί ευκολότερα. Τέλος, προσφέρει τεράστια ποικιλία από διαφορετικά δεδομένα, με αποτέλεσμα να έχουμε λειτουργίες από μουσεία μέχρι και ζωολογικούς κήπους. Η προσθήκη της πρόβλεψης του καιρού στο Bot μας, έγινε μέσω του API **OpenWeatherMap**, το οποίο διαθέτει ένα πολύ αναλυτικό documentation σε σχέση με άλλα παρόμοιά του κι έτσι ήταν ιδιαίτερα ευκολόχρηστο.

Κυριολεκτικά, με 2-3 γραμμές κώδικα μπορούμε να δίνουμε στον χρήστη την τρέχουσα θερμοκρασία και το είδος των καιρικών φαινομένων (π.χ. ηλιοφάνεια, συννεφιά, καταιγίδες) για την τοποθεσία που επιλέγει, ακόμη και σε εικονίδια. Το κανάλι **WebChat** ήρθε να αντικαταστήσει την πλατφόρμα του messenger ως προς το ερώτημα για το που στεγάζεται το bot. Στην ουσία, είναι ένα κανάλι το οποίο επικοινωνεί απευθείας με το microsoft azure όπου έχει γίνει deploy το bot μας και το αναπαριστά σε ένα διαδικτυακό περιβάλλον. Ο χρόνος απόκρισης είναι σχεδόν μηδαμινός και γι'αυτό επιλέξαμε την πλατφόρμα στην πλατφόρμα μας. Επίσης, παρέχει επιλογές της μορφής couldn't sent για ενημέρωση του χρήστη σε περίπτωση που δεν παραδοθεί ένα μήνυμα στο bot, καθώς και το κουμπί retry έτσι ώστε το μήνυμα να σταλεί ξανά σε περίπτωση αποτυχίας. Η πλατφόρμα NLP που χρησιμοποιήσαμε ήταν το **Wit.ai**. Το περιβάλλον χρήσης ήταν αρκετά εύκολο, καθώς στην αρχή έπρεπε να μελετήσουμε συστηματικά το documentation προκειμένου να επεκτείνουμε το bot. Επίσης, ήταν αρκετά εύκολο να κατανοήσει κανείς τους διαφορετικούς τύπους οντοτήτων, διότι χωρίς αυτή τη γνώση δεν μπορεί να αναπτυχθεί ένα λειτουργικό σύστημα που να περιλαμβάνει NLP και να εξάγει τα σωστά αποτελέσματα. Ωστόσο, ως εργαλείο περιείχε τους περιορισμούς, όπως ήταν η αδυναμία πλήρους κατανόησης της ελληνικής γλώσσας σε τυχούσες αλλαγές που συνέβησαν μεταξύ εφάμιλλων διαφορετικών προτάσεων. Επίσης, εκτός από λέξεις κλειδιά τα οποία παρέχει προκειμένου να αναγνωρίζονται από το bot, παρέχει και εκφράσεις έτσι ώστε ένας χρήστης να μπορεί να εκτελέσει μια κανονική συνομιλία σαν να μιλάει με κάποιο ανθρώπινο ον και όχι με μια οντότητα τεχνητής νοημοσύνης. Ένας από τους βασικότερους λόγους για τους οποίους επιλέξαμε το **firebase**, είναι ότι επιθυμούσαμε να παίρνουμε τα δεδομένα της εφαρμογής μας σε πραγματικό χρόνο. Ο συγχρονισμός των δεδομένων στην ανάπτυξη εφαρμογών, συνήθως αναγκάζει τους developers να αποσύρουν την προσοχή τους από την εφαρμογή, με αποτέλεσμα η ανάπτυξή της να επιβραδύνεται. Η απεικόνιση των εισόδων και των εγγραφών των χρηστών με γραφήματα και ημερομηνίες, μας βοηθάει στο να εξάγουμε συμπεράσματα για το ποιες εποχές και ημερομηνίες μέσα στον χρόνο, έχει απήχηση η εφαρμογή μας. Ένα ακόμη χαρακτηριστικό, ήταν η αποθήκευση των στοιχείων των χρηστών σε JSON αρχεία με ονομασία collections, τα οποία ταξινομούσε με id σαν documents κι έτσι καθιστούσε την μεθοδο της αναζήτησης ευκολότερη. Το documentation που μας παρέχει για σύνδεση της βάσης με την εφαρμογή μας, εξαγωγή και εμφάνιση στοιχείων από την βάση στην εφαρμογή, αλλά και η διαδικασία της αυθεντικοποίησης των χρηστών, είναι πολύ κατανοητό και εύκολο.

8.2 Συμπεράσματα επί της εφαρμογής

Η εφαρμογή μας χωρίζεται σε δύο σκέλη. Το ένα σκέλος είναι η επέκταση του bot που προϋπήρχε σαν BotVoyage και το άλλο σκέλος αφορά μια καινούργια πλατφόρμα για την ενσωμάτωσή του έτσι ώστε να αποτελεί ένα αυτόνομο λογισμικό το οποίο δεν θα εξαρτάται αποκλειστικά και μόνο από την εφαρμογή messenger. Όσον αφορά το πρώτο σκέλος της εφαρμογής μας (bot), αυτό που επιθυμούσαμε από την αρχή το καταφέραμε στο βέλτιστο επίπεδο του. Αρχικά, έγινε μια μεγάλη επέκταση στο bot προσθέτοντας 26 επιπλέον κατηγορίες, καθώς το παλιό λογισμικό περιείχε μόνο κάποιες βασικές. Εκτός από το Foursquare API το οποίο προϋπήρχε, προσθέσαμε και το OpenWeatherMap API, μέσω του οποίου παίρναμε αποτελέσματα για τον καιρό και τα εμφανίζαμε στην συζήτηση μέσω κατάλληλων πληροφοριών και εικονιδίων. Σχετικά με την φυσική γλώσσα επικοινωνίας (ελληνικά-αγγλικά), προσθέσαμε καινούργια χαρακτηριστικά όπως εκφράσεις και απαντήσεις ή ερωτήσεις προς και από τους χρήστες αντίστοιχα και διορθώσαμε κάποια λάθη που εντοπίσαμε. Επίσης, έγιναν αλλαγές στην μέθοδο που ζητάει και παίρνει την τοποθεσία το bot, στέλνοντάς την στα διάφορα APIs που επικοινωνεί, με αποτέλεσμα οι χρήστες να μπορούν να αναζητήσουν κατηγορίες για τις τοποθεσίες που επιθυμούν ακόμα και αν βρίσκονται στην άλλη μεριά του κόσμου. Έχοντας αναφερθεί στις καινούργιες λειτουργίες που υλοποιήθηκαν εξ ολοκλήρου, ας δούμε και αυτές που δεν υλοποιήθηκαν παρόλη την προσπάθεια και τον χρόνο που διαθέσαμε. Ξεκινώντας από την καινούργια έκδοση του bot framework v. 4 την οποία θελήσαμε να εντάξουμε στο bot, μπορούμε να πούμε ότι λόγω της χρονικής περιόδου της οποίας έγινε η διάθεσή της, η οποία συνέπεσε με την ήδη δημιουργία όλων των λειτουργιών που υλοποιήθηκαν, δεν ήταν δυνατή η ολοκλήρωσή του. Επίσης, το documentation της συγκεκριμένης έκδοσης ήταν ελλιπής. Στην συνέχεια, στοχεύαμε στην χρήση διαφορετικών API ανάλογα με την τοποθεσία του χρήστη (π.χ. Yelp API για Αμερική). Αυτή η προσθήκη δεν ολοκληρώθηκε καθώς ήταν δευτερεύουσα προτεραιότητα σχετικά με τις λειτουργίες τις οποίες είχαμε να υλοποιήσουμε. Ακόμα μια λειτουργία η οποία δεν υλοποιήθηκε, ήταν αυτή του GPS Navigation με την οποία ο χρήστης θα μπορούσε να βρίσκει διαδρομές για μια

τοποθεσία που θα αναζητούσε. Κι αυτό, διότι η αναζήτηση της τοποθεσίας εμφάνισε προβλήματα στα smartphones στα οποία την τεστάρουμε.

Στο δεύτερο σκέλος όπως προαναφέρθηκε, στοχεύαμε την υλοποίηση μιας διαδικτυακής πλατφόρμας στην οποία θα στεγάζεται το bot, διαφορετική από αυτήν του messenger. Έτσι αποφασίσαμε την ένταξη του bot στην ιστοσελίδα μέσω του καναλιού webchat, το οποίο διατίθεται σαν react component μέσω npm πακέτου. Η ιδέα της πλατφόρμας αποφασίστηκε ώστε να δώσουμε στις λειτουργίες του bot μια καινούργια μορφή, μέσω ενός SEO Friendly περιβάλλοντος με διάφορες λειτουργίες οι οποίες παρέχονται και από άλλους επικρατέστερους και ολοκληρωμένους ιστοχώρους και εξειδικεύονται όπως και το BotVoyage, στην διανομή ταξιδιωτικών πληροφοριών. Η υλοποίηση της πλατφόρμας ήταν επιτυχής, προσδίδοντας στον χρήστη σημαντικές πληροφορίες όπως σχόλια και βαθμολογίες χρηστών, οδούς, τηλέφωνα και εικόνες των τοποθεσιών αναζήτησης. Επίσης, υλοποιήθηκε η δημιουργία μιας βάσης δεδομένων η οποία εξειδικεύεται στην αυθεντικοποίηση των χρηστών που πραγματοποιούν είσοδο στην πλατφόρμα μας και στην αποθήκευση των στοιχείων τους. Αυτό συνέβη καθώς θέλαμε να υπάρχει ένα είδος ασφάλειας στο bot και να μην χρησιμοποιείται από χρήστες οι οποίοι δεν είναι εξουσιοδοτημένοι, καθώς και άλλα λογισμικά. Η πλατφόρμα μας επιπρόσθετα, διατηρεί ιστορικό αναζητήσεων καθώς και διαμόρφωση προφίλ του χρήστη. Ο χρήστης επίσης μπορεί να αλλάξει το όνομα προφίλ του, τον κωδικό εισόδου του και να πλοηγηθεί ανάμεσα σε σελίδες οι οποίες αφορούν τον τουριστικό τομέα καθώς και το bot. Κάποιες από τις λειτουργίες που δεν υλοποιήθηκαν, ήταν κατά κύριο λόγο η επικοινωνία του bot με την πλατφόρμα έτσι ώστε το bot να λειτουργεί σαν το κύριο navigation στην σελίδα αντί του search το οποίο συνηθίζεται. Μπορούμε εδώ να πούμε ότι τα προβλήματα της επίτευξης αυτής της λειτουργίας ήταν και τα μεγαλύτερα τα οποία αντιμετωπίσαμε σε όλη την εργασία. Ο χρόνος που απαιτήθηκε για την υλοποίηση αυτής της λειτουργίας δεν ήταν στα χρονικά πλαίσια της εργασίας και γι' αυτό δεν κατάφερε να καταστεί δυνατό. Επίσης, μια ακόμα τέτοια λειτουργία που θα επιθυμούσαμε να εντάξουμε στην πλατφόρμα, ήταν αυτή του Speech Recognition, η οποία ήταν χαμηλότερης προτεραιότητας και δεν ολοκληρώθηκε. Συμπερασματικά, το BotVoyage σε Messenger platform και η καινούργια μας πλατφόρμα με το Bot, αποτελούν μία εύκολη μορφή συνομιλίας σε φυσική γλώσσα ανάμεσα σε χρήστη και bot, όπως ακριβώς σχεδιάζαμε εξ' αρχής, μη έχοντας όμως ενσωματωμένες κάποιες λειτουργίες που θέλαμε να εμπεριέχει, οι οποίες όμως κατείχαν χαμηλότερη προτεραιότητα σε σχέση με τις

βασικότερες, οι οποίες και επιτεύχθηκαν. Παρακάτω παρατίθεται ο πίνακας με τις λειτουργικές και της μη λειτουργικές απαιτήσεις με τον κωδικό, τον τίτλο τους και με μία βαθμίδα που κυμαίνεται ανάμεσα σε τρεις τιμές(Μερικώς, Πλήρως, Καθόλου). Βάση αυτών που υλοποιήθηκαν μερικώς και αυτών που δεν υλοποιήθηκαν καθόλου ,θα συνεχίσουμε στην επόμενη επικεφαλίδα, για να διευκρινίσουμε τα επόμενα βήματα της εφαρμογής μας.

8.2.1 Λειτουργικές απαιτήσεις πλατφόρμας

1	Σύνδεση στο Διαδίκτυο	Πλήρως
2	Ύπαρξη βάσης δεδομένων	Πλήρως
3	Αυθεντικοποίηση Χρηστών	Πλήρως
4	Αποθήκευση Χρηστών	Πλήρως
5	Δημιουργία Προφίλ του Χρήστη	Πλήρως
6	Ενημέρωση του Χρήστη	Πλήρως
7	Αλλαγή Κωδικών Πρόσβασης	Πλήρως
8	Τροποποίηση Προφίλ Χρήστη	Πλήρως
9	Ιστορικό αναζητήσεων	Πλήρως
10	Αλλαγή κωδικού πρόσβασης μέσω email	Πλήρως
11	Επικοινωνία με βάση Real Time Database	Πλήρως
12	Επικοινωνία με βάση Cloud Storage	Καθόλου
13	Εναλλαγή Σελίδων	Πλήρως
14	Δυνατότητα Αποσύνδεσης	Πλήρως
15	Σύνδεση σε Api	Πλήρως
16	Σύνδεση μέσω google login	Πλήρως
17	Σύνδεση μέσω Facebook	Μερικώς

18	Επικοινωνία bot με πλατφόρμα	Καθόλου
19	NewsFeed	Καθόλου
20	Προσφορές	Καθόλου
21	Εναλλαγή URLS	Πλήρως
22	Σύνδεση bot με Πλατφόρμα	Πλήρως
23	Δημιουργία admin dashboard	Μερικώς
24	Ιστορικό μεθόδου σύνδεσης	Πλήρως
25	Ιστορικό logins και register	Πλήρως
26	Ύπαρξη web browser	Πλήρως

8.2.2 Μη λειτουργικές απαιτήσεις πλατφόρμας

1	Βέλτιστη ανταπόκριση	Πλήρως
2	Επιλογή διακομιστή	Πλήρως
3	Δυνατότητα Επαναχρησιμοποίησης	Πλήρως
4	Ασφάλεια στοιχείων χρήστη	Πλήρως
5	Αξιόπιστα δεδομένα	Πλήρως
6	Ενημέρωση για σφάλματα	Μερικώς
7	Νόμιμη χρήση api	Πλήρως
8	Ιδιωτικότητα στοιχείων χρηστών	Πλήρως
9	Ύπαρξη demo	Καθόλου
10	Απουσία χρονικού περιορισμού	Μερικώς

11	Χρήση Πόρων Συστήματος	Μερικώς
12	Ευχρηστία Συστήματος	Πλήρως
13	Εκτέλεση σε διαφορετικό περιβάλλον	Μερικώς

8.3 Επόμενα βήματα

Στο στάδιο που βρίσκεται αυτήν την στιγμή η εφαρμογή μας, μπορούμε να πούμε ότι πρόκειται για μια αυτόνομη εφαρμογή την οποία οραματιζόμασταν εξαρχής. Ωστόσο, θα μπορούσαν να πραγματοποιηθούν διάφορες επεκτάσεις οι οποίες θα εξέλισαν την εφαρμογή και θα την έφταναν ένα επίπεδο πιο πέρα. Αρχικά, όπως είδαμε στο κεφάλαιο 3 κάθε μια ταξιδιωτική πλατφόρμα την οποία και αναλύσαμε, διατηρεί και αναπτύσσει συνεργασίες με άλλους ιστοχώρους οι οποίοι εξειδικεύονται στις κρατήσεις για συγκεκριμένες κατηγορίες αναζητήσεων, όπως είναι τα αεροπορικά εισιτήρια, τραπέζια για εστιατόρια, αυτοκίνητα ή άλλου είδους οχήματα για ενοικίαση και πολλά άλλα. Μια μελλοντική συνεργασία με ένα άλλο site το οποίο θα πραγματοποιούσε κρατήσεις όπως για παράδειγμα το booking.com, θα προσέλκυε σίγουρα χρήστες στην πλατφόρμα μας. Αναφερόμενοι στον τομέα της συνεργασίας και του marketing, θα μπορούσε επίσης να υπάρχουν συνεργασίες με επιχειρήσεις έτσι ώστε να προβάλλουν τις υπηρεσίες τους και τα προϊόντα τους στην πλατφόρμα μας μέσω διαφημίσεων. Ακόμα η επικοινωνία των χρηστών μέσα στην πλατφόρμα μέσω του firebase cloud messaging είναι μια επέκταση που καθιστά λειτουργικότερη την εφαρμογή μας. Μια άλλη λειτουργία η οποία σκεφτήκαμε είναι η δυνατότητα των χρηστών να πραγματοποιούν check in στην πλατφόρμα μας, με την ένδειξη meet me there έτσι ώστε να μπορούν να συναντηθούν με άλλους χρήστες σε μια συγκεκριμένη τοποθεσία αναζήτησης. Ακόμα, ένα άλλο χαρακτηριστικό το οποίο υλοποιήθηκε και συνδέθηκε με την πλατφόρμα μας όμως δεν μπορούσαμε να εξάγουμε δεδομένα μέσω αυτού στην πλατφόρμα είναι το admin dashboard. Στην ουσία οι admins θα μπορούσαν να στείλουν μηνύματα ή ακόμα και να δημιουργήσουν posts στο προφίλ των χρηστών σχετικά με καινούργιες λειτουργίες οι

οποίες προστέθηκαν στο bot ή στην πλατφόρμα μας. Όπως είπαμε και στο κεφάλαιο 4 σχετικά με τα προβλήματα και τους περιορισμούς των εργαλείων τα posts δημιουργούνται και αποθηκεύονται κανονικά στην βάση δεδομένων cloud storage του firebase, αλλά δεν μπορούμε να τα εμφανίσουμε στην πλατφόρμα μας σε αντίθεση με την βάση real time database. Ο κύριος οραματισμός μας, όπως αναφέρθηκε και στα συμπεράσματα επί της εφαρμογής είναι η ενσωμάτωση του bot στην πλατφόρμα, όχι όμως μόνο σαν ξεχωριστή οντότητα, αλλά με την μορφή του κύριου navigation πια, το οποίο είναι κάτι πρωτοποριακό σε σχέση με παρόμοιες ιστοσελίδες. Με αυτό, οι χρήστες θα μπορούν να μιλάνε στο bot και από κει οι πληροφορίες να εμφανίζονται στην πλατφόρμα δυναμικά. Έτσι, η αλληλεπίδραση του χρήστη με την πλατφόρμα χωρίς πια search bar όπως συνηθίζεται, θα ξεχωρίζει χωρίς φυσικά να είμαστε απόλυτα σίγουροι για την λειτουργικότητα και ευκολία στην χρήση του. Τέλος, θα ήταν ιδανικό να περιλαμβάνουμε στην εφαρμογή μας ως επόμενο βήμα, η ύπαρξη διάφορων API, έτσι ώστε να υπάρχει μία ενδιαφέρουσα ποικιλία δεδομένων για να βοηθούν τον χρήστη να έχει μία πολυεπίπεδη όψη διαφορετικών μαγαζιών, υπηρεσιών.

BIBΛΙΟΓΡΑΦΙΑ

Bayan Abu, S. & Atwell E. 2005. Different measurements metrics to evaluate a chatbot system. In bridging the Gap: Academic and Industrial Research in Dialog Technologies Workshop Proceedings. pg. 89-96

Jiyoun, J. 2004. Atlanta, GA, USA. The Study of the Application of a Keywords-based Chatbot System on the Teaching of Foreign Languages.

Berihan Ravandi. Let's Make a Chatbot - Microsoft Bot Framework + Node.js, <https://tutorials.botsfloor.com/lets-make-a-chatbot-microsoft-bot-framework-node-js-7da211149c2f>, 2017

Robert Standefer, Azure Bot Service, <https://docs.microsoft.com/en-us/azure/bot-service/bot-service-design-pattern-embed-web-site?view=azure-bot-service-4.0>

Kim Brandl & Robert Standefer, Azure Bot Service, <https://docs.microsoft.com/en-us/azure/bot-service/rest-api/bot-framework-rest-direct-line-3-0-start-conversation?view=azure-bot-service-4.0>

John Taylor, Azure Bot Service, <https://docs.microsoft.com/en-us/azure/bot-service/bot-builder-concept-dialog?view=azure-bot-service-4.0>

Robert Stafender & Denise Mak, Azure Bot Service, <https://docs.microsoft.com/en-us/azure/bot-service/nodejs/bot-builder-nodejs-dialog-overview?view=azure-bot-service-3.0>

Den Delimarsky & Robert Stafender & Denise Mak, Azure Bot Service, <https://docs.microsoft.com/en-us/azure/bot-service/nodejs/bot-builder-nodejs-dialog-waterfall?view=azure-bot-service-3.0>

Den Delimarsky & Robert Stafender & Kim Brandl, Azure Bot Service, <https://docs.microsoft.com/en-us/azure/bot-service/nodejs/bot-builder-nodejs-backchannel?view=azure-bot-service-3.0>

The Bot Framework Team. Microsoft, Customize Web Chat for websites, <https://blog.botframework.com/2017/10/11/customize-web-chat-websites/>, 2017

Stoll, M. Portland, (US) Aug 2, 2007. Collective Intelligence Recommender System For Travel Information and Travel Industry Marketing Platform.

Satheesh, M. & D'mello, B. J. & Krol, J., 2015. Web Development with MongoDB and NodeJS, Birmingham, United Kingdom: Packt Publishing Ltd.

Jiaqi Pan, Website Chatbots: The New Alternative to Facebook Messenger Bots, <https://medium.com/landbot-io/website-chatbots-the-new-sheriff-in-town-274ecfece467>, 2017

Ron Levinson, The Tools Every Bot Creator Must Know, <https://chatbotmagazine.com/the-tools-every-bot-creator-must-know-c0e9dd685094>, 2016

Jiaqi Pan, Conversational Interfaces: The Future of Chatbots, <https://chatbotmagazine.com/conversational-interfaces-the-future-of-chatbots-18975a91fe5a>, 2017

Wikipedia, npm(software), [https://en.wikipedia.org/wiki/Npm_\(software\)](https://en.wikipedia.org/wiki/Npm_(software))

Npm Docs, What is npm, <https://docs.npmjs.com/getting-started/what-is-np>

Facebook for Developers, Messenger Bot Analytics,

<https://developers.facebook.com/docs/messenger-platform/analytics>

Facebook for Developers, Webhooks, <https://developers.facebook.com/docs/messenger-platform/webhook>

Facebook for Developers, IDs & Profile Information, <https://developers.facebook.com/docs/messenger-platform/identity>

TechMagic, ReactJS vs Angular5 vs Vue.js - What to choose in 2018, <https://medium.com/@TechMagic/reactjs-vs-angular5-vs-vue-js-what-to-choose-in-2018-b91e028fa91d>, 2018

Coder Academy, You Can Build an FB Messenger Style Chat App with ReactJs - Here's How! (Intermediate), <https://medium.com/@coderacademy/you-can-build-an-fb-messenger-style-chat-app-with-reactjs-heres-how-intermediate-211b523838ad>, 2016

Russell & Norvig (2003) Russell, Stuart J.; Norvig, Peter (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall,

What is Artificial Intelligence (AI), Technopedia, <https://www.techopedia.com/definition/190/artificial-intelligence-ai>

MDN web docs, About Javascript, https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript

Wikipedia , History , <https://en.wikipedia.org/wiki/JavaScript#History>

Tom Peterson, Artificial Intelligence in Travel and Tourism, <https://www.webcredible.com/blog/artificial-intelligence-tourism-travel/>, 2017

Github, <https://github.com/>

OpenWeatherMap, <https://openweathermap.org/>

Brad Traversy, Node.js, Express & MongoDB Dev to Deployment, <https://www.udemy.com/nodejs-express-mongodb-dev-to-deployment/>

Maximillian Schwarzmuller, React 16.6 - The Complete Guide (incl.React Router & Redux),

<https://www.udemy.com/react-the-complete-guide-incl-redux/>

Jonas Schmedtmann The Complete Javascript Course 2019: Build Real Projects!,

<https://www.udemy.com/the-complete-javascript-course/>

Jonas Schmedtmann, Build Responsive Real World Websites with HTML5 and CSS3, <https://www.udemy.com/design-and-develop-a-killer-website-with-html5-and-css3/>

Visual Studio Code, Using React in Visual Studio Code, <https://code.visualstudio.com/docs/nodejs/reactjs-tutorial>

Steve Brown, Create a Local Venue App Using React/Redux with the Foursquare API, Part I,

<http://stevebrown.co/journal/creating-a-local-venue-app-using-reactredux-with-the-foursquare-api-part-i>, 2017

Foursquare, <https://developer.foursquare.com/>

Duc Cash Vo, Kamran Iqbal, Robert Standefer, Create a bot with Bot Service, <https://docs.microsoft.com/en-us/azure/bot-service/bot-service-quickstart?view=azure-bot-service-3.0>

Wikipedia, Natural-language processing, https://en.wikipedia.org/wiki/Natural_language_processing

Wit.ai, <https://wit.ai/>

Robin Orheden, Build a React.js Application with User Login and Authentication, <https://www.sitepoint.com/tutorial-build-a-react-js-application-with-user-login-and-authentication/>, 2017

Marmelab, React-Admin, <https://marmelab.com/react-admin/index.html>

Calling The Microsoft Bot Framework Using The Direct Line API, <http://aihelpwebsite.com/Blog/EntryId/16/Calling-The-Microsoft-Bot-Framework-Using-The-Direct-Line-API>

Andy Lin, Getting Current Location with Javascript, <https://coderwall.com/p/ouofjg/getting-current-location-with-javascript>, 2016

Wes Chen, Node.js - Bot Framework - Direct Line, <https://tw.saowen.com/a/14c61bb08cff2d509da8ca8df93ff122e67946d05cc539b0e16dc24a8f69ca8e>, 2017

Mikkonen, T., & Taivalsaari, A. (2007). Using JavaScript as a real programming language.

Node.js, About, <https://nodejs.org/en/>

Lee Stott, What is Microsoft Bot Framework Overview, https://blogs.msdn.microsoft.com/uk_faculty_connection/2016/04/05/what-is-microsoft-bot-framework-overview/

React.parts, <https://react.parts/>

Google, Firebase Documentation, <https://firebase.google.com/docs/database/>

Google, Firebase Functions, <https://firebase.google.com/docs/functions/>

Google, Firebase Cloud Messaging, <https://firebase.google.com/docs/cloud-messaging/>

Google, Firebase Cloud Storage, <https://firebase.google.com/docs/storage/>

Google, Firebase Hosting, <https://firebase.google.com/docs/hosting/>

Google, Firebase Authentication, <https://firebase.google.com/docs/auth/>

Google, Google Analytics for Firebase, <https://firebase.google.com/docs/analytics/>

What is Webpack, Survivejs, <https://survivejs.com/webpack/what-is-webpack/>

Webpack Dev-Server, Survivejs, <https://survivejs.com/webpack/developing/webpack-dev-server/>

Webpack Loading Styles, Survivejs, <https://survivejs.com/webpack/styling/loading/>

Webpack Separating CSS, Survivejs, <https://survivejs.com/webpack/styling/separating-css/>

Webpack Code Splitting, Survivejs, <https://survivejs.com/webpack/building/code-splitting/>

Webpack Loading Images, Survivejs, <https://survivejs.com/webpack/loading/images/>

Webpack Loading Fonts, Survivejs, <https://survivejs.com/webpack/loading/fonts/>

Webpack Loading Javascript, Survivejs, <https://survivejs.com/webpack/loading/javascript/>

What is Babel? Documentation, Babel, <https://babeljs.io/docs/en/>

React Documentation, React, <https://reactjs.org/docs/>

React Components and Props, React, <https://reactjs.org/docs/components-and-props.html>