



ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

**ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΠΛΗΡΟΦΟΡΙΑΣ
ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Διαμοιρασμός χρήσιμων και αξιόπιστων πληροφοριών φιλτραρισμένων ως προς θέση
και χρόνο
(Sharing useful and trusted information filtered by location and time)



Ιχτσής Απόστολος

Επιβλέπων: Ιωάννης Ρεφανίδης, Καθηγητής

ΘΕΣΣΑΛΟΝΙΚΗ

ΣΕΠΤΕΜΒΡΙΟΣ 2017

Φύλλο Εξέτασης

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Τόπος:..... Ημερομηνία:/...../.....

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

Ιωάννης Ρεφανίδης, Καθηγητής

Υπογραφή:.....

Ηλίας Σακελλαρίου, Επίκουρος Καθηγητής

Υπογραφή:.....

Νικόλαος Σαμαράς, Καθηγητής

Υπογραφή:.....

«Ο συγγραφέας βεβαιώνει ότι το περιεχόμενο του παρόντος έργου είναι αποτέλεσμα προσωπικής εργασίας και ότι έχει γίνει η κατάλληλη αναφορά στην εργασία τρίτων, όπου κάτι τέτοιο ήταν απαραίτητο σύμφωνα με τους κανόνες της ακαδημαϊκής δεοντολογίας»

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να προσφέρω τις μεγαλύτερες ευχαριστίες μου στα άτομα τα οποία αποτέλεσαν τους αρωγούς για την υλοποίηση αυτού του έργου. Πρώτα, λοιπόν, θέλω να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Ιωάννη Ρεφανίδη ο οποίος μου το εμπιστεύθηκε και μου έδωσε τη δυνατότητα να ασχοληθώ με κάτι που με ενδιαφέρει πολύ και αποτέλεσε το μέσο για να μάθω πολλά καινούρια πράγματα.

Επίσης θα ήθελα να ευχαριστήσω τους φίλους και την οικογένειά μου οι οποίοι ήταν μαζί μου από την αρχή, πίστεψαν σε εμένα και με στήριξαν σε όλη την διάρκεια δημιουργίας αυτού του έργου. Ιδιαίτερες ευχαριστίες θέλω να δώσω στην αδερφή μου η οποία με βοήθησε εμπράκτως με τον σχεδιασμό της ιστοσελίδας ώστε να αποκτήσει την τελική της εμφάνιση.

Περίληψη

Καθημερινά, οι χρήστες του διαδικτύου πλημμυρίζονται από πολλές πληροφορίες τις περισσότερες από τις οποίες απλώς τις προσπερνούν γιατί δεν τους ενδιαφέρουν. Έτσι καταλήγει αυτός ο καταγισμός άσκοπος για τον χρήστη αφού δεν βλέπει απευθείας αυτό που πραγματικά θέλει. Ενδεχόμενη λύση σε αυτό το πρόβλημα μπορεί να αποτελέσει το «προϊόν» αυτής της εργασίας. Το Infoging αποτελεί ένα μέσο κοινωνικής δικτύωσης, το οποίο δίνει τη δυνατότητα στους χρήστες του να προβάλουν τις πληροφορίες που αυτοί επιθυμούν, από τα άτομα που αυτοί επιθυμούν, με αποτέλεσμα να λαμβάνουν πιο στοχευμένες πληροφορίες και να αυξάνεται η ικανοποίησή τους. Στην εργασία αυτή αναλύεται ο τρόπος με τον οποίο κατασκευάστηκε αυτό το έργο, τα προαπαιτούμενα που χρειάζονται για την υλοποίησή του, καθώς και προτάσεις για μελλοντική μελέτη και περαιτέρω ανάπτυξή του.

Λέξεις Κλειδιά: Αξιοπιστία, Social Media, ASP.NET, Azure, Google Maps

Abstract

Every day, Internet users are flooded with a lot of information, most of which simply overtake them because they are irrelevant with their needs. This makes this overwhelming quantity of information unnecessary for the user as he does not directly see what he really wants. A possible solution to this problem can be the product of this work. Infoging is a means of social networking that enables its users to display the information they want from the people they want, thus delivering more targeted results and increasing their satisfaction. In this paper we analyze the way in which this project was constructed, the prerequisites needed for its implementation, as well as proposals for future study and further development.

Keywords: Trust, Social Media, ASP.NET, Azure, Google Maps

Περιεχόμενα

| | |
|---|----|
| 1. Εισαγωγή..... | 1 |
| 1.1 Σκοπός της εργασίας | 1 |
| 1.2 Δομή της εργασίας | 2 |
| 2. Τεχνολογίες..... | 3 |
| 2.1 Γλώσσες προγραμματισμού..... | 3 |
| 2.1.1 HTML | 3 |
| 2.1.2 CSS..... | 4 |
| 2.1.3 Javascript..... | 5 |
| 2.1.4 jQuery..... | 6 |
| 2.1.5 C#..... | 7 |
| 2.1.6 SQL..... | 8 |
| 2.2 Περιβάλλον Προγραμματισμού..... | 9 |
| 3. Σχεδίαση και Υλοποίηση | 12 |
| 3.1 Ανάλυση απαιτήσεων | 12 |
| 3.2 Περιπτώσεις Χρήσης | 13 |
| 3.3 Διάγραμμα αρχιτεκτονικής συστήματος..... | 16 |
| 3.4 Δημιουργία καινούριου project | 16 |
| 3.5 Δημιουργία λογαριασμού στο Azure | 17 |
| 3.6 Κατασκευή server και βάσης δεδομένων | 20 |
| 3.7 Σύνδεση της βάσης με το project..... | 23 |
| 3.7.1 Εναλλακτικός τρόπος (Δημιουργία τοπικής βάσης) | 24 |
| 3.8 Κατασκευή πινάκων | 26 |
| 3.9 Google Maps JavaScript API | 29 |
| 3.10 Δημιουργία ιστοτόπου | 29 |
| 3.10.1 Σελίδα εγγραφής και σύνδεσης..... | 31 |
| 3.10.2 Σελίδα ενεργοποίησης λογαριασμού | 33 |
| 3.10.3 Σελίδα έναρξης διαδικασίας αλλαγής κωδικού..... | 34 |
| 3.10.4 Σελίδα δημιουργίας νέου κωδικού | 35 |
| 3.10.5 Σελίδα εύρεσης πληροφοριών | 36 |
| 3.10.6 Σελίδα προφίλ του χρήστη..... | 42 |
| 3.10.7 Σελίδα ομάδων | 47 |
| 3.10.8 Σελίδα προφίλ άλλων χρηστών | 49 |
| 4. Παρουσίαση..... | 50 |
| 4.1. Εγγραφή, ενεργοποίηση λογαριασμού και αλλαγή συνθηματικού..... | 50 |
| 4.1.1 Εγγραφή..... | 50 |

| | |
|--|----|
| 4.1.2 Ενεργοποίηση λογαριασμού | 51 |
| 4.1.3 Αλλαγή συνθηματικού | 51 |
| 4.2 Προφίλ του χρήστη..... | 54 |
| 4.3 Ομάδες | 60 |
| 4.4 Τέταρτη σελίδα..... | 62 |
| 4.5 Εύρεση πληροφοριών | 64 |
| 5. Συμπεράσματα | 66 |
| 6. Προτάσεις για μελλοντική μελέτη | 67 |
| Βιβλιογραφία..... | 68 |

Κατάλογος Εικόνων

| | |
|---|----|
| Εικόνα 1: Λογότυπο HTML | 3 |
| Εικόνα 2: Δομή της HTML..... | 4 |
| Εικόνα 3: Λογότυπο CSS | 4 |
| Εικόνα 4: Δομή της CSS | 5 |
| Εικόνα 5: Λογότυπο της JavaScript | 5 |
| Εικόνα 6: Δομή της JavaScript | 6 |
| Εικόνα 7: Λογότυπο της jQuery..... | 6 |
| Εικόνα 8: Δομή της jQuery | 6 |
| Εικόνα 9: Λογοτυπο της C#..... | 7 |
| Εικόνα 10 Ποσοστό χρήστης διαφόρων γλωσσών προγραμματισμού από τη μεριά του server..... | 7 |
| Εικόνα 11: Λογότυπο του Visual Studio\..... | 9 |
| Εικόνα 12: Διαθέσιμοι τύποι του Visual Studio 2017 | 9 |
| Εικόνα 13: Λογότυπο του Microsoft Azure | 10 |
| Εικόνα 14: Λογότυπο του Microsoft SQL Server | 11 |
| Εικόνα 15: Διάγραμμα Περιπτώσεων Χρήσης | 13 |
| Εικόνα 16: Διάγραμμα Αρχιτεκτονικής Συστήματος | 16 |
| Εικόνα 17: Δημιουργία νέου website project | 17 |
| Εικόνα 18: Δημιουργία λογαριασμού στο Azure | 18 |
| Εικόνα 19: Δημιουργία λογαριασμού Imagine | 18 |
| Εικόνα 20: Επικύρωση λογαριασμού μέσω πανεπιστημιακού email..... | 19 |
| Εικόνα 21: Ολοκλήρωση διαδικασίας ενεργοποίησης | 20 |
| Εικόνα 22 Αρχική σελίδα του Azure | 20 |
| Εικόνα 23: Δημιουργία Resource Group..... | 21 |
| Εικόνα 24: Συμπλήρωση στοιχείων στο Resource Group | 21 |
| Εικόνα 25: Δημιουργία βάσης δεδομένων | 22 |
| Εικόνα 26: Συμπλήρωση στοιχείων για τη δημιουργία της βάσης..... | 22 |
| Εικόνα 27: Συμπλήρωση στοιχείων για τον server | 23 |
| Εικόνα 28: Επιλογή Data Source | 24 |
| Εικόνα 29: Συμπλήρωση στοιχείων στον server | 24 |
| Εικόνα 30: Δημιουργία τοπικής βάσης | 25 |
| Εικόνα 31: Βάση έτοιμη για χρήση | 25 |
| Εικόνα 32: Απαραίτητη Πίνακες..... | 26 |
| Εικόνα 33: Οι μέθοδοι της πρώτης σελίδας με τις απαραίτητες βιβλιοθήκες | 31 |
| Εικόνα 34: Οι μέθοδοι της δεύτερης σελίδας με τις απαραίτητες βιβλιοθήκες | 33 |
| Εικόνα 35: Οι μέθοδοι της τρίτης σελίδας με τις απαραίτητες βιβλιοθήκες | 34 |
| Εικόνα 36: Οι μέθοδοι της τέταρτης σελίδας με τις απαραίτητες βιβλιοθήκες | 35 |
| Εικόνα 37: Οι μέθοδοι του MasterPage μαζί με τις απαραίτητες βιβλιοθήκες | 36 |
| Εικόνα 38: Οι μέθοδοι της πέμπτης σελίδας με τις απαραίτητες βιβλιοθήκες | 38 |
| Εικόνα 39: Οι μέθοδοι της έκτης σελίδας | 42 |
| Εικόνα 40: Οι μέθοδοι της έβδομης σελίδας..... | 47 |
| Εικόνα 41: Οι μέθοδοι της όγδοης σελίδας | 49 |
| Εικόνα 42: Παρουσία της καρτέλας εγγραφής χρήστη..... | 50 |
| Εικόνα 43: Παρουσίαση σελίδας ενεργοποίησης λογαριασμού | 51 |

| | |
|--|----|
| Εικόνα 44: Μετάβαση στην σελίδα αλλαγής συνθηματικού..... | 51 |
| Εικόνα 45: Παρουσίαση σελίδας αποστολής e-mail για αλλαγή συνθηματικού | 52 |
| Εικόνα 46: Παρουσίαση σελίδας αλλαγής συνθηματικού | 53 |
| Εικόνα 47: Παρουσίαση καρτέλας σύνδεσης | 53 |
| Εικόνα 48: Παρουσίαση σελίδας προφίλ του χρήστη | 54 |
| Εικόνα 49: Επιλογές της κεφαλίδας | 55 |
| Εικόνα 50: Λίστα σε πεδίο για αυτόματα αποτελέσματα | 55 |
| Εικόνα 51:Αλλαγή στοιχείων και εικόνας προφίλ του χρήστη | 55 |
| Εικόνα 52: Στατιστικά του χρήστη και των ομάδων του | 56 |
| Εικόνα 53: Δημοσιεύσεις του χρήστη | 56 |
| Εικόνα 54: Διαδικασία δημοσίευση πληροφορίας..... | 57 |
| Εικόνα 55: Ημερολόγια για την επιλογή ημερομηνίας για τη δημοσίευση | 58 |
| Εικόνα 56: Σελίδα ομάδων πριν τη δημιουργία κάποιας: | 60 |
| Εικόνα 57: Σελίδα ομάδων μετά τη δημιουργία κάποιας..... | 60 |
| Εικόνα 58: Παρουσίαση σελίδας προφίλ άλλων χρηστών | 62 |
| Εικόνα 59: Παρουσίαση σελίδας εύρεσης πληροφοριών | 64 |
| Εικόνα 60: Εμφάνιση σελίδας μετά την αναζήτηση με κρυμμένα τα πεδία αναζήτησης | 65 |

1. Εισαγωγή

Τα μέσα κοινωνικής δικτύωσης αποκτούν ολοένα και μεγαλύτερο κομμάτι στη ζωή των ανθρώπων. Με την ραγδαίως αυξανόμενη χρήση των smartphone, τα οποία επιτρέπουν στους χρήστες να έχουν ένα «κινητό» γραφείο στην τσέπη τους, τα πράγματα είναι πλέον πολύ απλά. Οι επιλογές μεταξύ των μέσων κοινωνικής δικτύωσης είναι πάρα πολλές. Είτε αυτό θα είναι το Facebook, είτε το Twitter είτε οποιοδήποτε άλλο αντίστοιχο μέσο, ο χρήστης μπορεί να έρθει σε επαφή με όλον τον πλανήτη απευθείας από την παλάμη του χεριού του.

Αυτό που κάνει μοναδικό το κάθε μέσο και προσελκύει χρήστες σε αυτό, είναι η ιδιαιτερότητα του ως προς τις παροχές του σε αυτούς. Ανάλογα με το τι ζητάει ένας χρήστης, έχει τη δυνατότητα να στραφεί σε συγκεκριμένο/-α μέσο/-α. Το Facebook για παράδειγμα είναι μέσο προβολής και διαφήμισης, κάτι που γίνεται ιδιαίτερα εμφανές, ειδικά τα τελευταία χρόνια, από την έκρηξη διαφημίσεων ιδίως νεοσυσταθέντων επιχειρήσεων των οποίων ο κύριος τομέας δράσης είναι το ίδιο το Facebook αφού αυτές πουλάνε τα προϊόντα τους μόνο μέσω αυτού, λόγω της απευθείας σύνδεσης που τους παρέχει με υποψήφιους καταναλωτές από όλον τον πλανήτη.

Το Infoging, αποτελεί ένα νέο μέσο κοινωνικής δικτύωσης. Ένα μέσο με το οποίο οι χρήστες θα μπορούν να μοιράζονται στοχευμένες πληροφορίες, οι οποίες θα είναι διαθέσιμες μόνο σε όποιον τις ζητάει. Δηλαδή οι πληροφορίες αυτές θα οριοθετούνται τοπικά και χρονικά και ο κάθε χρήστης θα μπορεί να τις φιλτράρει ώστε να λαμβάνει μόνο αυτές που καλύπτουν τα όρια που ο ίδιος θα θέτει. Το γεγονός αυτό σε συνδυασμό με τη δημιουργία έμπιστων ομάδων για τη διαμοίραση και λήψη πληροφοριών, αποτελεί κάτι το καινοτόμο και θα προσφέρει στον χρήστη δυνατότητες τις οποίες κανένα άλλο μέσο δεν τις παρείχε μέχρι τώρα.

1.1 Σκοπός της εργασίας

Όπως προαναφέρθηκε, οι επιλογές των social media οι οποίες υπάρχουν διαθέσιμες στις μέρες μας είναι πολλές. Τα χαρακτηριστικά τους είναι ποικίλα και ο καθένας από εμάς ελκύεται από αυτό που θα πληρεί περισσότερο τις ανάγκες του. Ανάγκες οι οποίες διαφέρουν από άτομο σε άτομο. Μία από αυτές είναι κοινή για όλους τους ανθρώπους, η ανάγκη της πληροφόρησης. Με τον έναν ή με τον άλλο τρόπο όλοι οι χρήστες θέλουν να λαμβάνουν πληροφορίες για τους υπόλοιπους, για αυτά που συμβαίνουν δίπλα τους, γύρω τους.

Το ερώτημα είναι όμως αν οι πληροφορίες που λαμβάνουν οι άνθρωποι είναι πραγματικά χρήσιμες. Αν είναι όντως πληροφορίες που θέλουν και ζητάνε ή αν είναι απλώς πληροφορίες τις οποίες οι υπόλοιποι προσφέρουν και αυτοί τις δέχονται ανεξάρτητα από τα δικά τους προσωπικά κριτήρια και θέλω.

Σκοπός της εργασίας αυτής, λοιπόν, είναι να αναφέρει και να περιγράψει όλα τα μέσα και τις τεχνολογίες που είναι απαραίτητο να χρησιμοποιηθούν ώστε να υλοποιηθεί ένα τέτοιο έργο.

1.2 Δομή της εργασίας

Το πρώτο θέμα το οποίο παρουσιάζεται στο Κεφάλαιο 2, είναι οι τεχνολογίες και τα μέσα τα οποία χρησιμοποιήθηκαν έτσι ώστε να έρθει σε πέρας η υλοποίηση του έργου αυτού. Οι επιλογές ήταν αρκετές αλλά χρησιμοποιήθηκαν ίσως οι πιο διαδεδομένες ανάμεσα σε αυτές που υπήρχαν διαθέσιμες. Στη συνέχεια, στο Κεφάλαιο 3, αναλύονται τα βήματα σχεδίασης και υλοποίησης της εφαρμογής ενώ στο Κεφάλαιο 4 γίνεται μία παρουσίαση του τρόπου χρήσης του ιστοτόπου. Στο Κεφάλαιο 5, παρουσιάζονται τα συμπεράσματα που εξάχθηκαν από όλη αυτή τη διαδικασία και τέλος στο Κεφάλαιο 6, γίνονται ορισμένες προτάσεις για περαιτέρω μελέτη.

2. Τεχνολογίες

2.1 Γλώσσες προγραμματισμού

Με την πάροδο των χρόνων, το web development γίνεται όλο και περισσότερο δημοφιλές. Οι γλώσσες προγραμματισμού οι οποίες δίνουν σε κάποιον την δυνατότητα να δημιουργήσει ένα website πληθαίνουν συνεχώς ενώ παράλληλα οι πιο παλιές αναβαθμίζονται ανά τακτά χρονικά διαστήματα ώστε να μπορέσουν να συμβαδίσουν με τις αυξανόμενες απαιτήσεις των χρηστών. Οι επιλογές λοιπόν ήταν αρκετές όσον αφορά τις γλώσσες οι οποίες θα αποτελούσαν το μέσο για την υλοποίηση αυτού του έργου. Τελικά επιλέχθηκαν κατά κύριο λόγο αυτές οι οποίες προτείνονται σε έναν αρχάριο όταν ξεκινά τα πρώτα του βήματα στον τομέα αυτό.

2.1.1 HTML

Η HTML (*HyperText Markup Language*), είναι μία γλώσσα σήμανσης και με τα συστατικά της δημιουργούμε τη δομή μιας ιστοσελίδας. Τα έγγραφα HTML, αποτελούνται από στοιχεία HTML τα οποία στην πιο γενική μορφή τους έχουν τρία συστατικά: ένα ζεύγος από ετικέτες, την «ετικέτα εκκίνησης» και την «ετικέτα τερματισμού», μερικές ιδιότητες μέσα στην ετικέτα εκκίνησης, και τέλος το κείμενο ή το γραφικό περιεχόμενο μεταξύ των ετικετών, το οποίο μπορεί να περιλαμβάνει και άλλα στοιχεία εμφωλευμένα μέσα του. Το στοιχείο HTML, μπορεί να είναι οτιδήποτε ανάμεσα στις ετικέτες εκκίνησης και τερματισμού. Τέλος, κάθε ετικέτα περικλείεται από σύμβολα «μεγαλύτερο από» και «μικρότερο από», δηλαδή < και >. (Wikipedia)



Εικόνα 1: Λογότυπο HTML

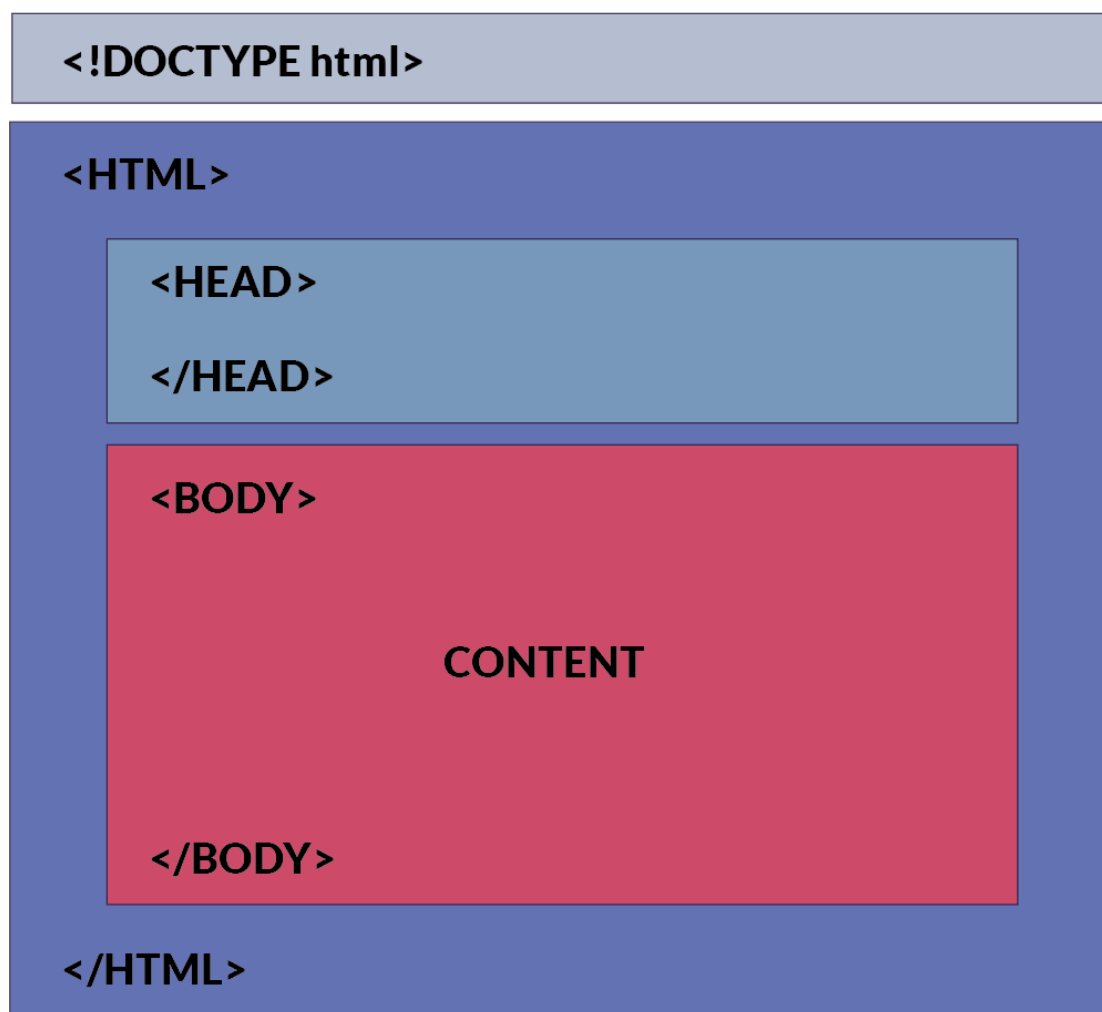
Επομένως, η γενική μορφή ενός στοιχείου HTML είναι:

```
<ετικέτα ιδιότητα1="τιμή1" ιδιότητα2="τιμή2">περιεχόμενο</ετικέτα>.
```

Το όνομα κάθε στοιχείου HTML είναι το ίδιο όνομα που χρησιμοποιείται στις αντίστοιχες ετικέτες. Το όνομα της ετικέτας τερματισμού ξεκινά με μια κάθετο «/», η οποία παραλείπεται στα άδεια στοιχεία. Τέλος, αν δεν αναφέρονται ρητά οι ιδιότητες ενός στοιχείου, τότε χρησιμοποιούνται οι προεπιλογές σε κάθε περίπτωση. (Wikipedia)

Ένα παράδειγμα ετικέτας στην HTML είναι η ετικέτα <header></header>, η οποία είναι η ετικέτα κεφαλίδας και ανάμεσα στις ετικέτες βάζουμε οποιαδήποτε πληροφορία πρέπει να συμπεριληφθεί σε αυτή.

Η βασική δομή κάθε εγγράφου HTML είναι η ακόλουθη:



Εικόνα 2: Δομή της HTML

2.1.2 CSS

Η γλώσσα CSS (Cascading Style Sheet) αποτελεί αναπόσπαστο κομμάτι της κατασκευή ιστοσελίδων. Σκοπός της είναι ο έλεγχος της εμφάνισης ενός εγγράφου το οποίο έχει γραφεί με μία γλώσσα σήμανσης όπως η HTML.

Η CSS σχεδιάστηκε κυρίως για να επιτρέψει τον διαχωρισμό της παρουσίασης και του περιεχομένου, συμπεριλαμβανομένων πτυχών όπως η διάταξη, τα χρώματα και οι γραμματοσειρές. Αυτός ο διαχωρισμός μπορεί να βελτιώσει την προσβασιμότητα του περιεχομένου, να δώσει μεγαλύτερη ευελιξία και έλεγχο στις προδιαγραφές των χαρακτηριστικών παρουσίασης, να επιτρέψει σε πολλαπλές σελίδες HTML να μοιράζονται μορφοποίηση καθορίζοντας το σχετικό CSS σε ξεχωριστό αρχείο .css και μειώνοντας την πολυπλοκότητα και την επανάληψη στο δομικό περιεχόμενο. (Wikipedia)



Εικόνα 3: Λογότυπο CSS

Οι εντολές της είναι ιδιαίτερα σαφείς, δηλαδή στην πλειοψηφία τους μπορεί κανείς να αντιληφθεί τι είδους αλλαγή θα επιφέρει κάθε μια από αυτές στο site απλά και μόνο διαβάζοντας το όνομά της. Παρακάτω φαίνεται ένα κομμάτι κώδικα σε CSS.

Είναι εμφανείς, όπως αναφέρθηκε πριν, οι αλλαγές που πρόκειται να γίνουν στο site με τη χρήση των εντολών αυτών.

```
1  body {
2      background-color: #D6D6F5;
3  }
4
5  h1 {
6      color: #3333CC;
7      text-align: center;
8  }
9
10 .circle {
11     height: 100px;
12     width: 100px;
13     background-color: purple;
14     border-radius: 100px;
15 }
16 };
17
18
```

Εικόνα 4: Δομή της CSS

2.1.3 Javascript

Η γλώσσα αυτή σε συνδυασμό με τις δύο γλώσσες που αναφέρθηκαν προηγουμένως αποτελούν τους βασικούς πυρήνες κατασκευής ιστοσελίδων. Η JavaScript χρησιμοποιείται ώστε να γίνει μία ιστοσελίδα περισσότερο διαδραστική και να προσφέρει στον χρήστη ένα πιο εύχρηστο περιβάλλον συνεπώς αποτελεί αναπόσπαστο τμήμα στην πλειοψηφία των website.



Εικόνα 5: Λογότυπο της JavaScript

Η JavaScript αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται. (Flannagan, 2006)

Πλέον υπάρχουν περιβάλλοντα που επιτρέπουν στους προγραμματιστές να χρησιμοποιούν τη JavaScript για την υλοποίηση desktop και mobile εφαρμογών, όπως για παράδειγμα οι επεξεργαστές κειμένου και τα αρχεία PDF. Σημαντικό είναι επίσης να τονισθεί ότι παρόλο που η JavaScript θυμίζει πολύ την Java όσον αφορά το όνομα αλλά και τα διάφορα χαρακτηριστικά που έχουν, στην ουσία είναι δύο διαφορετικές μεταξύ τους γλώσσες.

```

<!--JavaScript code goes here.-->
<script language="javascript">
    function MouseRollover(MyImage) {
        MyImage.src = "MyPicture2.jpg";
    }

    function MouseOut(MyImage) {
        MyImage.src = "MyPicture1.jpg";
    }
</script>

```

Εικόνα 6: Δομή της JavaScript

2.1.4 jQuery

Η jQuery είναι μία cross-platform βιβλιοθήκη της JavaScript η οποία έχει σχεδιαστεί για να απλοποιήσει την υλοποίηση σεναρίων στην πλευρά του πελάτη (client-side) της HTML. Αποτελεί ελεύθερο λογισμικό και κυκλοφορεί με την άδεια του MIT. Η jQuery παρέχει στον προγραμματιστή πολλές δυνατότητες, μεταξύ των οποίων συγκαταλέγονται η επιλογή και διαμόρφωση στοιχείων HTML, η διαμόρφωση στοιχείων CSS, διεργασίες γεγονότων (event-handlers), εφέ και animations με JavaScript, διαμόρφωση του HTML DOM [Document Object Module], χρήση AJAX και πληθώρα άλλων εφαρμογών. (PCMAG, 2011)



Εικόνα 7: Λογότυπο της jQuery

Είναι τόσο δημοφιλής που χρησιμοποιείται σε πάνω από το 65% των 10.000 ιστοτόπων με τη μεγαλύτερη επισκεψιμότητα όπως το Google adWords, το YouTube αλλά και από κολοσσούς της πληροφορικής όπως είναι η Microsoft και η IBM.

```

$(document).ready(function() {
    changeColor();
});

function changeColor() {
    var rndColors = ["#00FF00", "#CCCCCC", "#995499", "#FFFFFF", "#FF9900"];
    var selColor = Math.floor(Math.random() * rndColors.length);
    $('div').css("background-color", rndColors[selColor]);
    setTimeout(changeColor, 1000);
}

```

Εικόνα 8: Δομή της jQuery

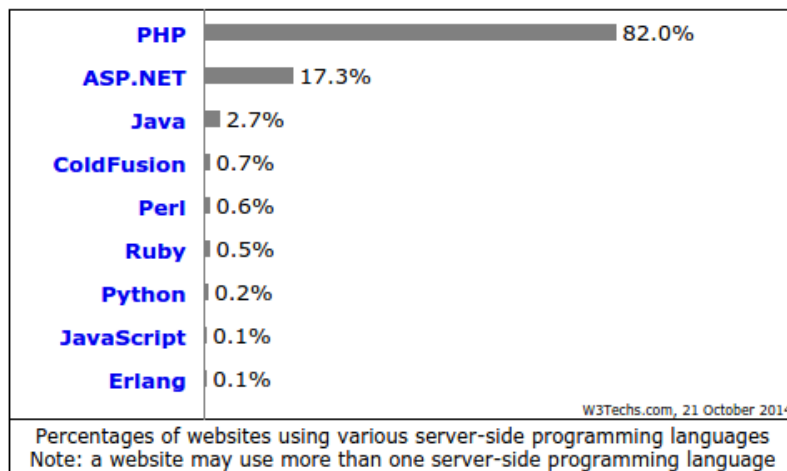
2.1.5 C#

Οι γλώσσες που αναφέρθηκαν μέχρι στιγμής, αφορούν το front-end κομμάτι του ιστοτόπου. Δηλαδή όλα αυτά τα οποία συμβαίνουν στον περιηγητή του χρήστη όταν μπαίνει στη σελίδα καθώς και τα στοιχεία που παρουσιάζονται σε αυτόν και αλληλοεπιδρά μαζί τους. Εξίσου σημαντικό κομμάτι για την σωστή υλοποίηση ενός τέτοιου έργου είναι και το back-end κομμάτι, δηλαδή η δημιουργία των συμβάντων που εκτελούνται όταν ο χρήστης στέλνει πληροφορίες στον server, όπως για παράδειγμα συμβαίνει με την αποστολή των δεδομένων μιας φόρμας.



Εικόνα 9: Λογότυπο της C#

Και σε αυτόν τον τομέα οι επιλογές μεταξύ των γλωσσών είναι πάρα πολλές. Παρακάτω φαίνεται μια εικόνα με τις διαθέσιμες επιλογές που υπάρχουν. Είναι εμφανής η τεράστια υπεροχή της PHP συγκριτικά με τις υπόλοιπες γλώσσες, με το ASP.NET να έρχεται δεύτερο στην κατάταξη.



Εικόνα 10 Ποσοστό χρήσης διαφόρων γλωσσών προγραμματισμού από τη μεριά του server

Παρά την υπεροχή της PHP, επιλέχθηκε για την πραγματοποίηση αυτής της εφαρμογής η γλώσσα C#. Η επιλογή αυτή έγινε λόγω προηγούμενης επαφής, στα πλαίσια ενός ακαδημαϊκού μαθήματος, με τη C# και το Visual Studio το οποίο είναι ένα λογισμικό κατασκευασμένο από τη Microsoft και υποστηρίζει αυτή τη γλώσσα. Επίσης μία ακόμη δυνατότητα που παρείχε αυτή η επιλογή είναι η ευκολία μεταφόρτωσης του ιστοτόπου στο Azure το οποίο είναι ένα ακόμη προϊόν της Microsoft και προσφέρει Cloud δυνατότητες για την φιλοξενία server, δημιουργία βάσης δεδομένων και διάφορα άλλα. Περισσότερες λεπτομέρειες για τα δύο αυτά λογισμικά υπάρχουν στις επόμενες ενότητες.

Σχετικά με την C#, πρόκειται για μία κομψή αντικειμενοστραφή γλώσσα προγραμματισμού, που επιτρέπει στους προγραμματιστές να δημιουργήσουν μια ποικιλία ασφαλών και ισχυρών εφαρμογών που εκτελούνται στο .NET Framework.

Μπορεί να χρησιμοποιηθεί για τη δημιουργία Windows Client Applications, υπηρεσίες Web XML, κατακευμασμένα συστήματα, εφαρμογές πελάτη-διακομιστή, βάσεις δεδομένων και πολλά άλλα. (Wagner, Wenzel, tompratt-AQ, & Latham, 2017)

Η γλώσσα αυτή αποτελεί επίσης μία ισχυρή γλώσσα. Κάποια παραδείγματα μεγάλων ιστοτόπων οι οποίοι έχουν υλοποιηθεί είναι οι ακόλουθοι:

- **Diply.com**
Μια κοινότητα που παρέχει ειδήσεις και ψυχαγωγία και είναι αφιερωμένη στη σύνδεση των θεατών με το περιεχόμενο που τους αρέσει να καταναλώνουν. (Diply.com)
- **W3Schools.com**
Είναι ένας δημοφιλής ιστότοπος για την εκμάθηση ηλεκτρονικών τεχνολογιών στο διαδίκτυο. Το περιεχόμενο περιλαμβάνει μαθήματα και αναφορές σχετικά με HTML, CSS, JavaScript, PHP, AngularJS, SQL, Bootstrap και jQuery. Δέχεται περισσότερους από 10 εκατομμύρια μοναδικούς επισκέπτες κάθε μήνα και μετράει 45 εκατομμύρια επισκέπτες μηνιαίως. (W3Schools)
- **Stackoverflow.com**
Ίσως ο πιο δημοφιλής και αγαπημένος ιστότοπος των προγραμματιστών. Είναι ένα φόρουμ, στο οποίο συμμετέχουν εκατομμύρια προγραμματιστές από όλο τον πλανήτη. Μπορεί κανείς εύκολα να βρει μία ολοκληρωμένη απάντηση ή έστω κάποιο στοιχείο για να οδηγηθεί στη λύση του προβλήματός του.

Αυτά είναι ίσως τα μεγαλύτερα παραδείγματα αλλά φυσικά υπάρχουν και άλλα πολλά. Επίσης εννοείται ότι όλα τα site της Microsoft και οι περισσότερες εφαρμογές που παράγει είναι κατασκευασμένες με τη χρήση της C#. Γενικά η C# χρησιμοποιείται κατά κύριο λόγο από επιχειρήσεις για την παραγωγή προγραμμάτων που θα χρησιμοποιηθούν από την ίδια γι' αυτό και τα παραδείγματα δεν είναι τόσο πολλά συγκριτικά με την PHP.

2.1.6 SQL

Η SQL (Structured Query Language), είναι μία γλώσσα υπολογιστών στις βάσεις δεδομένων που σχεδιάστηκε για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System, RDBMS) και η οποία, αρχικά, βασίστηκε στη σχεσιακή άλγεβρα. Η γλώσσα περιλαμβάνει δυνατότητες ανάκτησης και ενημέρωσης δεδομένων, δημιουργίας και τροποποίησης σχημάτων και σχεσιακών πινάκων, αλλά και ελέγχου πρόσβασης στα δεδομένα. (Wikipedia)

Το Visual Studio, υποστηρίζει τον SQL Server μέσα από ένα εύχρηστο γραφικό περιβάλλον. Για το λόγο αυτό η γλώσσα SQL, δεν χρειάζεται να χρησιμοποιηθεί για τη δημιουργία, την ανανέωση και τη διαγραφή πινάκων καθώς αυτά είναι δυνατόν να γίνουν εύκολα μέσω του περιβάλλοντος αυτού. Επομένως, η SQL, θα χρησιμοποιηθεί μόνο σε ότι αφορά την εισαγωγή, διαγραφή και ανανέωση των δεδομένων στη βάση.

2.2 Περιβάλλον Προγραμματισμού

Παράμετροι οι οποίοι έπαιξαν σημαντικό ρόλο στην υλοποίηση της εργασίας αυτής είναι το περιβάλλον Visual Studio και το cloud σύστημα Azure. Αρχικά θα γίνει αναφορά σε κάποια πράγματα για το Visual Studio όπως η διαδικασία που πρέπει να ακολουθηθεί για την εγκατάστασή του καθώς και μερικά πράγματα για τον τρόπο λειτουργίας του και στη συνέχεια ακολουθεί ένα τμήμα σχετικά με το Azure και τις δυνατότητες που προσφέρει.



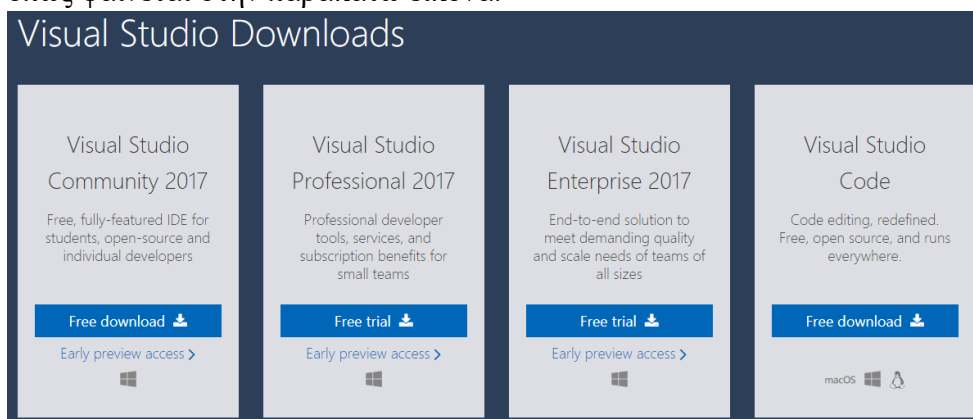
Εικόνα 11: Λογότυπο του Visual Studio\

Το Microsoft Visual Studio λοιπόν είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) από τη Microsoft. Χρησιμοποιείται για την ανάπτυξη προγραμμάτων ηλεκτρονικών υπολογιστών για τα Windows, καθώς και για ιστότοπους, εφαρμογές ιστού, υπηρεσίες ιστού και εφαρμογές για κινητά. Το Visual Studio χρησιμοποιεί πλατφόρμες ανάπτυξης λογισμικού της Microsoft, όπως τα Windows API, τα Windows Forms, το Windows Presentation Foundation, το Windows Store και το Microsoft Silverlight. (Wikipedia). Επίσης είναι σημαντικό να τονίσουμε ότι το Visual Studio υποστηρίζει 36 διαφορετικές γλώσσες προγραμματισμού και επιτρέπει στον code editor και στον debugger να υποστηρίζει (σε διαφορετικούς βαθμούς) σχεδόν οποιαδήποτε γλώσσα προγραμματισμού, δοθέντος ότι υπάρχει ειδική υπηρεσία για κάθε γλώσσα. (Wikipedia)

Είναι λοιπόν προφανές ότι πρόκειται για ένα αρκετά ευέλικτο περιβάλλον εργασίας, το οποίο προσφέρει πολλές δυνατότητες στους προγραμματιστές και ο καθένας μπορεί να το προσαρμόσει στις ανάγκες του και να υλοποιήσει τον σκοπό του.

Παρακάτω παρέχονται οδηγίες, σχετικά με τον τρόπο εγκατάστασης της τελευταίας έκδοσης του περιβάλλοντος που είναι η Visual Studio 2017. (Microsoft) Η έκδοση αυτή ξεκίνησε να διανέμεται στη σταθερή της μορφή τον Μάρτιο του 2017. Επίσης προσφέρεται και ο τύπος Community ο οποίος παρέχει τις περισσότερες δυνατότητες του Visual Studio και έρχεται χωρίς κόστος.

1. Αρχικά ο χρήστης πρέπει να επισκεφθεί την ιστοσελίδα της Microsoft [εδώ](#), όπου μπορεί να επιλέξει ποια μορφή του περιβάλλοντος επιθυμεί να κατεβάσει όπως φαίνεται στην παρακάτω εικόνα.



- 2.

Εικόνα 12: Διαθέσιμοι τύποι του Visual Studio 2017

3. Αφού ο χρήστης επιλέξει έναν από τους συνδέσμους, θα κατέβει στον υπολογιστή του ένα εκτελέσιμο αρχείο.
4. Με διπλό κλικ πάνω στο αρχείο, θα ξεκινήσει η διαδικασία εγκατάστασης του installer, μέσω του οποίου στη συνέχεια θα εγκατασταθεί το Visual Studio στον υπολογιστή.
5. Με το πέρας της προηγούμενης διαδικασίας θα δημιουργηθεί ο installer και με διπλό κλικ πάνω σε αυτόν ξεκινάει η εγκατάσταση του Visual Studio.
6. Ξεκινώντας από αυτή την έκδοση, η Microsoft παρέχει έναν νέο τρόπο εγκατάστασης. Επιτρέπει στον χρήστη να επιλέξει ο ίδιος τα κομμάτια που θέλει να εγκαταστήσει ανάλογα με τη δουλειά που θέλει να κάνει. Έτσι αφού ο χρήστης κάνει τις δικές του μοναδικές επιλογές (για την εργασία αυτή χρησιμοποιήθηκαν τα ASP.NET and web development, Azure development και Data Storage and Processing) συνεχίζει αφήνοντας το πρόγραμμα να εγκατασταθεί μόνο του έως ότου είναι έτοιμο για χρήση.

Αφού έχει ολοκληρωθεί η εγκατάσταση, ο χρήστης είναι έτοιμος να ξεκινήσει με την υλοποίηση του έργου. Λόγω της εγκατάστασης του Azure development, είναι πολύ εύκολο να συνδεθεί η εφαρμογή που θα υλοποιηθεί με τον server στο Azure. Η χρήση του Azure κανονικά γίνεται επί πληρωμή αλλά όλοι οι φοιτητές του τμήματος Εφαρμοσμένης Πληροφορικής όπως και φοιτητές άλλων τμημάτων πανεπιστημίων δικαιούνται κάποιες παροχές δωρεάν μέσω του προγράμματος Microsoft Imagine το οποίο συνδέεται με τους ακαδημαϊκούς λογαριασμούς των φοιτητών.

Το Microsoft Azure, (πρώην Windows Azure), είναι μια υπηρεσία cloud computing που δημιουργήθηκε από τη Microsoft για την κατασκευή, τη δοκιμή, την ανάπτυξη και τη διαχείριση εφαρμογών και υπηρεσιών μέσω ενός παγκόσμιου δικτύου data centers που διαχειρίζεται η Microsoft. (Wikipedia)



Εικόνα 13: Λογότυπο του Microsoft Azure

Το Azure κυκλοφόρησε τον Φεβρουάριο του 2010. Από τότε αναπτύσσεται και μεγαλώνει συνεχώς με αποτέλεσμα να παρέχει στους χρήστες του ολοένα και περισσότερες δυνατότητες, μετρώντας πλέον πάνω από 600 από αυτές. Προσφέρει στους πελάτες του ένα ολοκληρωμένο πακέτο παροχών, λ όπου μπορούν να συγκεντρωθούν τα διαφορετικά μέρη για παράδειγμα μιας επιχείρησης. Έχει κριθεί ως αρκετά αποδοτικό και υιοθετείται από συνεχώς μεγαλύτερο αριθμό ατόμων.

Κάποια από τα χαρακτηριστικά που το κάνουν τόσο δελεαστικό είναι για παράδειγμα, η δυνατότητα φιλοξενίας ιστοτόπων οι οποίοι λειτουργούν και με κάποια άλλη γλώσσα εκτός από την C#, η δυνατότητα δημιουργίας βάσης δεδομένων στον SQL Server ο οποίος τρέχει στο Cloud, η δυνατότητα να τρέχει ο ιστότοπος από ένα δικό του domain σε συνδυασμό με τη χρήση εργαλείων που ελέγχουν για παράδειγμα την επισκεψιμότητα κτλ καθώς και πολλά άλλα.

Τέλος, το τελευταίο κομμάτι που απαιτείται για την εφαρμογή είναι ο SQL Server. Με την εγκατάσταση του Visual Studio με τον τρόπο που αναφέρθηκε πριν, γίνεται η εγκατάσταση των απαιτούμενων πακέτων ώστε να μπορεί να χρησιμοποιηθεί ο SQL Server για τη δημιουργία και διαχείριση των βάσεων δεδομένων που χρειάζονται για την αποθήκευση των δεδομένων που θα παράγονται από τον χρήστη κατά τη διάρκεια χρήσης του site.



Εικόνα 14: Λογότυπο του Microsoft SQL Server

Το Visual Studio περιλαμβάνει έναν ιδιαίτερα εύκολο τρόπο πρόσβασης στον SQL Server ο οποίος συνδέεται στην εφαρμογή. Εύκολα και γρήγορα μπορεί ο χρήστης να δημιουργήσει καινούριους πίνακες, να αλλάξει υπάρχοντες, να προβάλλει τα δεδομένα τους και πολλά άλλα.

Αν ο ιστότοπος έχει συνδεθεί με κάποια βάση στο Azure, τότε η βάση αυτή εμφανίζεται στην καρτέλα του Server Explorer στο Visual Studio από όπου μπορούμε να κάνουμε τις παραπάνω αλλαγές άμεσα και εύκολα.

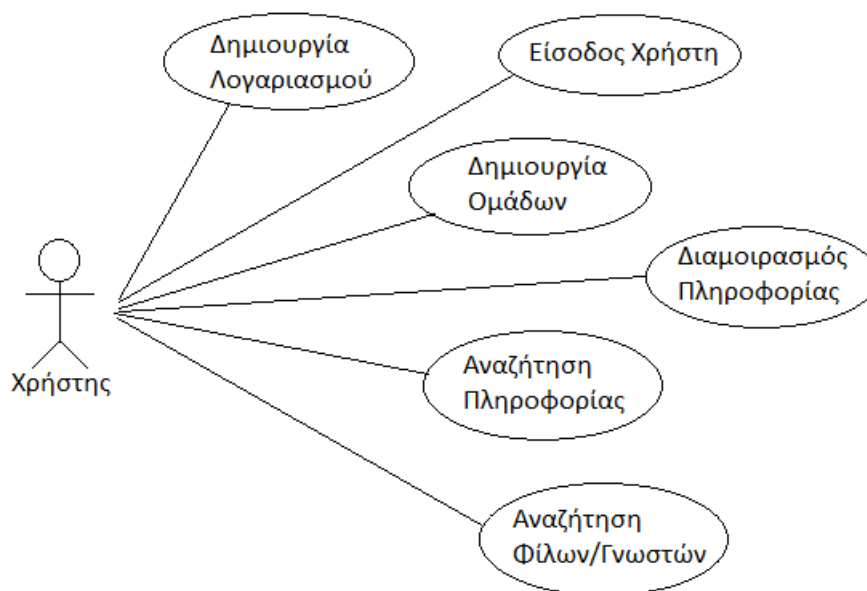
3. Σχεδίαση και Υλοποίηση

3.1 Ανάλυση απαιτήσεων

Όπως έχει προαναφερθεί, η συγκεκριμένη εφαρμογή βασίζεται στον διαμοιρασμό και την εύρεση στοχευμένων πληροφοριών, οριοθετημένων τοπικά και χρονικά. Απαραίτητη λοιπόν, αρχικά, είναι η εγγραφή του χρήστη στο σύστημα και η σύνδεση αυτού με τα στοιχεία που παρείχε, ώστε να γίνει η αναγνώρισή του. Για να δημοσιεύσει ο χρήστης μια πληροφορία, μεταβαίνει στο προφίλ του, συμπληρώνει τα πεδία που απαιτούνται και ολοκληρώνει τη διαδικασία. Η πληροφορία αυτή στη συνέχεια, είναι διαθέσιμη είτε σε όλους τους χρήστες του συστήματος είτε σε αυτούς που ενδεχομένως έχει επιλέξει ο χρήστης κατά τη δημοσίευσή της. Για να μοιράζεται και να λαμβάνει πληροφορίες από συγκείμενους χρήστες, του δίνεται η δυνατότητα κατασκευής ομάδων, αποτελούμενων από τα άτομα που αυτός επιθυμεί. Μετά τη δημιουργία ομάδων, μπορεί να επιλέξει να μοιράζεται ή να προβάλλει πληροφορίες, μόνο με τα άτομα αυτών. Για την προβολή πληροφοριών, ο χρήστης μπορεί να θέσει κάποια όρια, σχετικά με τον τόπο και τον χρόνο, καθώς και σχετικά με τα άτομα που δημοσίευσαν τις πληροφορίες που πρόκειται να προβάλλει. Αφού προσθέσει όσα κριτήρια επιθυμεί, εμφανίζει τα διαθέσιμα αποτελέσματα. Τα αποτελέσματα αυτά, μπορεί στη συνέχεια να τα βαθμολογήσει ως χρήσιμα ή ως αδιάφορα. Για να γίνει ακόμα πιο συγκεκριμένη η δημοσίευση και η αναζήτηση πληροφοριών, μπορούν να προστεθούν σε αυτές ετικέτες. Έτσι κατά την αναζήτηση, αν επιλεγούν κάποιες ετικέτες, θα εμφανιστούν μόνο οι πληροφορίες που περιέχουν όλες ή έστω κάποιες από αυτές. Επίσης, ο χρήστης μπορεί να αναζητήσει τα προφίλ άλλων χρηστών, με σκοπό να βρει γνωστούς και φίλους ή γενικά αξιόπιστα άτομα. Όταν προβάλλει το προφίλ ενός άλλου χρήστη, εμφανίζονται κάποια στατιστικά αυτού, όπως το πλήθος των δημοσιεύσεών του και πόσες από τις δημοσιεύσεις αυτές έχουν φανεί χρήσιμες σε άλλους χρήστες. Αυτό αποτελεί και ένα κριτήριο αξιοπιστίας του χρήστη. Ένα πιο δυνατό κριτήριο όμως, είναι η δυνατότητα φιλτραρίσματος των στατιστικών αυτών, βάσει των χρηστών μια ομάδας του χρήστη που προβάλλει το προφίλ. Τα παραπάνω δεδομένα φιλτράρονται και προσδιορίζονται σχετικά με το αν οι χρήστες της ομάδας που επιλέχθηκε, έχουν βαθμολογήσει θετικά τις δημοσιεύσεις αυτού του προφίλ.

3.2 Περιπτώσεις Χρήσης

Παρακάτω φαίνεται το διάγραμμα περιπτώσεων χρήσης της εφαρμογής και στη συνέχεια αναλύεται κάθε μια από αυτές τις περιπτώσεις.



Εικόνα 15: Διάγραμμα Περιπτώσεων Χρήσης

Δημιουργία Λογαριασμού

Βασική Ροή

1. Ο Χρήστης επιλέγει την καρτέλα «Εγγραφή».
2. Το Σύστημα μεταβαίνει στη φόρμα εγγραφής.
3. Ο Χρήστης εισάγει τα απαραίτητα στοιχεία.
4. Ο Χρήστης πατάει το κουμπί «Εγγραφή».
5. Το Σύστημα επεξεργάζεται τα στοιχεία.
6. Τα στοιχεία αποθηκεύονται στη βάση δεδομένων.
7. Το Σύστημα εμφανίζει μήνυμα επιτυχίας και κατευθύνει τον χρήστη στην καρτέλα σύνδεσης.

Εναλλακτική Ροή

- 5.α.1 Το όνομα χρήστη ή το e-mail υπάρχουν ήδη.
- 5.α.2 Το σύστημα εμφανίζει το κατάλληλο προειδοποιητικό μήνυμα.
- 5.α.3 Η περίπτωση χρήσης επιστρέφει στο βήμα 3 της βασικής ροής.

Είσοδος Χρήστη

Βασική Ροή

1. Ο Χρήστης επισκέπτεται τη σελίδα σύνδεσης.
2. Το Σύστημα εμφανίζει τη φόρμα σύνδεσης.
3. Ο Χρήστης εισάγει τα στοιχεία εισόδου του.
4. Το Σύστημα πιστοποιεί τον χρήστη και τον μεταφέρει στη σελίδα του προφίλ του.

Εναλλακτική Ροή

- 4.α.1 Ο Χρήστης εισήγαγε λάθος όνομα χρήστη ή συνθηματικό.
- 4.α.2 Το Σύστημα εμφανίζει μήνυμα λάθους στη φόρμα.
- 4.α.3 Η περίπτωση χρήσης επιστρέφει στο βήμα 3 της βασική ροής.

Δημιουργία Ομάδων

Βασική Ροή

1. Ο Χρήστης πατάει το κουμπί επεξεργασίας ομάδων.
2. Το Σύστημα μεταφέρει τον χρήστη στη σελίδα διαχείρισης ομάδων.
3. Ο Χρήστης εισάγει ένα όνομα για την ομάδα.
4. Ο Χρήστης επιλέγει το κουμπί «Δημιουργία».
5. Το Σύστημα αποθηκεύει τα δεδομένα στη βάση δεδομένων και δημιουργεί την ομάδα.
6. Ο Χρήστης πληκτρολογεί το όνομα του χρήστη που θέλει να προσθέσει στην ομάδα.
7. Ο Χρήστης επιλέγει το κουμπί προσθήκης.
8. Το Σύστημα αποθηκεύει τα δεδομένα στη βάση δεδομένων.
9. Το Σύστημα εμφανίζει μήνυμα επιτυχίας.

Αναζήτηση φίλων/γνωστών

Βασική ροή

1. Ο Χρήστης πληκτρολογεί τα γράμματα του ονόματος του χρήστη που ψάχνει.
2. Το Σύστημα ψάχνει στη βάση χρήστες, των οποίων το όνομα περιέχει τα γράμματα που πληκτρολογεί ο χρήστης.
3. Το Σύστημα επιστρέφει μία λίστα με τα ονόματα των χρηστών που βρήκε.
4. Ο Χρήστης βρίσκει το όνομα στη λίστα και το επιλέγει.
5. Το Σύστημα μεταφέρει τον χρήστη στη σελίδα του χρήστη.

Διαμοιρασμός Πληροφορίας

Βασική Ροή

1. Ο Χρήστης πηγαίνει στη σελίδα προφίλ.
2. Ο Χρήστης εισάγει υποχρεωτικά τοποθεσία, χρονική περίοδο και περιεχόμενο, ενώ προαιρετικά, μπορεί να εισάγει ομάδες και ετικέτες.
3. Το Σύστημα ελέγχει τα δεδομένα που εισήχθησαν.
4. Το Σύστημα αποθηκεύει τα δεδομένα αυτά.
5. Το Σύστημα αδειάζει τη φόρμα των δεδομένων.

Εναλλακτική Ροή

- 3.α.1 Ο Χρήστης δεν εισήγαγε κάποιο από τα υποχρεωτικά πεδία.
- 3.α.2 Το Σύστημα εμφανίζει το κατάλληλο μήνυμα σφάλματος
- 3.α.3 Η περίπτωση χρήσης επιστρέφει στο βήμα 2 της βασικής ροής.

Αναζήτηση Πληροφορίας

Βασική Ροή

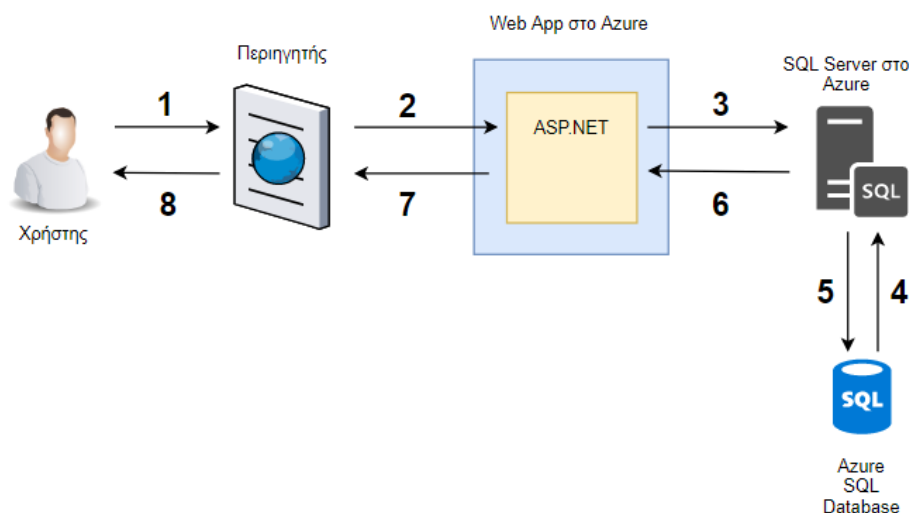
1. Ο Χρήστης πατάει το κουμπί «Αρχική».
2. Το Σύστημα μεταφέρει τον χρήστη στη σελίδα εύρεσης πληροφοριών.
3. Ο Χρήστης εισάγει υποχρεωτικά μια χρονική περίοδο και μια τοποθεσία, ενώ προαιρετικά μπορεί να εισάγει ομάδες ή/και ετικέτες.
4. Ο Χρήστης πατάει το κουμπί «Αναζήτηση».
5. Το Σύστημα συγκρίνει τα δεδομένα με αυτά που υπάρχουν στη βάση και επιστρέφει τις κατάλληλες πληροφορίες.

Εναλλακτική Ροή

- 4.α.1 Ο Χρήστης δεν εισήγαγε κάποιο από τα υποχρεωτικά πεδία.
- 4.α.2 Το Σύστημα εμφανίζει το κατάλληλο μήνυμα λάθους.
- 4.α.3 Η περίπτωση χρήσης επιστρέφει στο βήμα 3 της βασικής ροής.

3.3 Διάγραμμα αρχιτεκτονικής συστήματος

Στην παρακάτω εικόνα φαίνεται το διάγραμμα αρχιτεκτονικής του συστήματος. Ο χρήστης, αρχικά, μέσω του περιηγητή του, έρχεται σε επαφή με το web app, το οποίο φιλοξενείται στο Azure. Η εφαρμογή αυτή, επικοινωνεί με έναν SQL Server ο οποίος επίσης φιλοξενείται στο Azure. Αυτός με τη σειρά του, επικοινωνεί με τη βάση δεδομένων που υπάρχει στο ίδιο σύστημα, για να ανακτήσει τα απαιτούμενα δεδομένα. Ακολουθώντας την αντίστροφη διαδρομή, τα αποτελέσματα επιστρέφουν τελικά στον χρήστη.



Εικόνα 16: Διάγραμμα Αρχιτεκτονικής Συστήματος

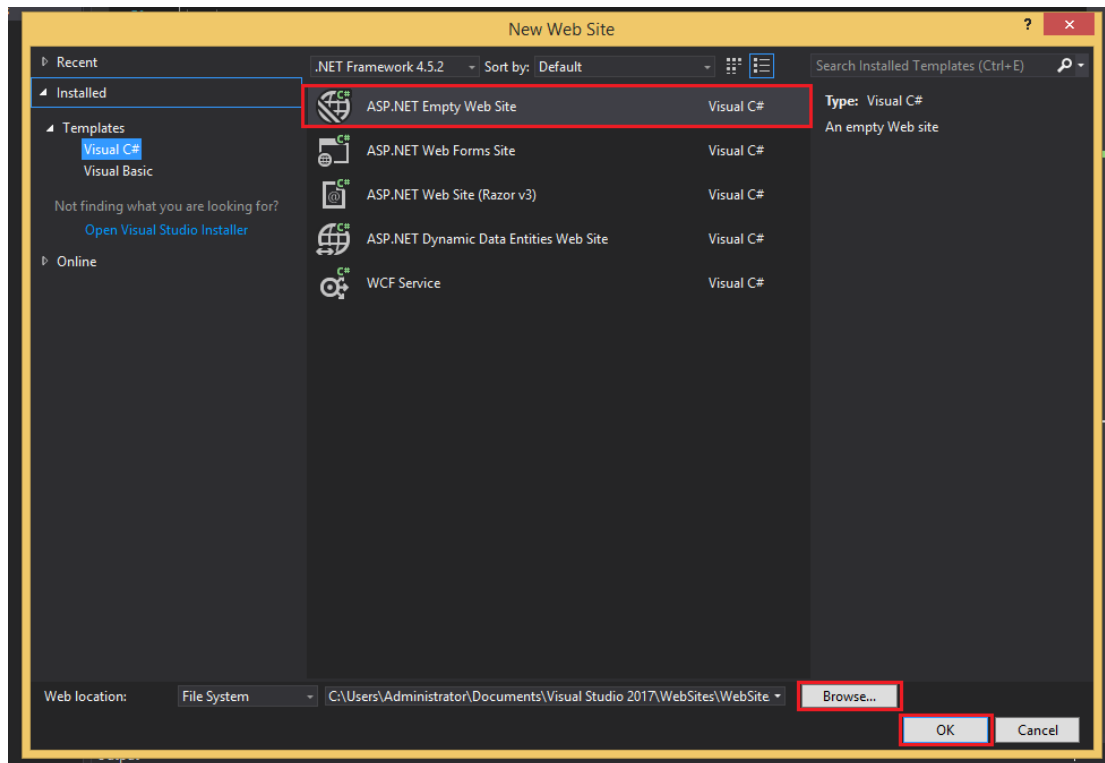
3.4 Δημιουργία καινούριου project

Αφού έχει ολοκληρωθεί η εγκατάσταση του απαραίτητου λογισμικού με τα βήματα που αναφέρθηκαν προηγουμένως, μπορεί πλέον να ξεκινήσει η υλοποίηση του ιστοτόπου. Το πρώτο βήμα που πρέπει να γίνει για τη δημιουργία ενός τέτοιου έργου είναι η δημιουργία ενός καινούριου project. Για να γίνει αυτό πρέπει να γίνουν οι ακόλουθες επιλογές:

File → New → Website ή με τον συνδυασμό των πλήκτρων Shift + Alt + N

(Η γλώσσα στην οποία είναι εγκατεστημένο το πρόγραμμα είναι τα αγγλικά, επομένως όλα τα βήματα που θα αναφέρονται θα είναι γραμμένα σε αυτή.)

Ακολουθώντας αυτά τα βήματα ο χρήστης, εμφανίζεται στην οθόνη του το παράθυρο που φαίνεται στην παρακάτω εικόνα. Από το παράθυρο αυτό επιλέγει ASP.NET Empty Web Site και μετά πατάει το κουμπί OK. Αν το επιθυμεί, μπορεί να αλλάξει το μονοπάτι όπου θα αποθηκευτεί το project καθώς επίσης και το όνομά του κάνοντας κλικ στο κουμπί Browse.



Εικόνα 17: Δημιουργία νέου website project

Σε περίπτωση που δεν υπάρχουν οι επιλογές αυτές προτεινόμενες, τότε ο χρήστης πρέπει να πάει στην κατηγορία Templates (αριστερή στήλη του παραθύρου) και να επιλέξει ως γλώσσα την C#. Άλλωστε αυτή είναι και η γλώσσα που θα χρησιμοποιηθεί για την δημιουργία του ιστοτόπου.




Τελειώνοντας με τα προηγούμενα βήματα και πατώντας OK ξεκινάει η διαδικασία δημιουργίας ενός καινούριου site project. Αμέσως μόλις ολοκληρωθεί αυτή η διαδικασία και ετοιμαστεί το κενό site, μπορεί να ξεκινήσει η διαδικασία υλοποίησης του έργου.

Πριν την κατασκευή του ιστοτόπου, χρήσιμο είναι να δημιουργηθεί στο Azure η βάση δεδομένων η οποία θα φιλοξενεί τα δεδομένα του site, να συνδεθεί η βάση αυτή με το project στο Visual Studio και τέλος να δημιουργηθούν οι πίνακες μέσα στους οποίους θα αποθηκεύονται τα δεδομένα.

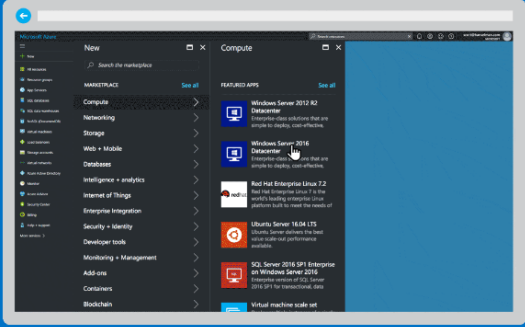
3.5 Δημιουργία λογαριασμού στο Azure

Για να ξεκινήσουμε να χρησιμοποιούμε το Azure πρέπει να δημιουργήσουμε λογαριασμό στην σελίδα της Microsoft. Για όσους δεν είναι φοιτητές απαιτείται η δημιουργία λογαριασμού επί πληρωμή, ενώ παρέχεται και ένα Free Trial το οποίο πιστώνει στον λογαριασμό του χρήστη 170€, ώστε να δοκιμάσει παροχές που αντιστοιχούν μέχρι και το ποσό αυτό. Για τη δημιουργία τέτοιου λογαριασμού επισκεπτόμαστε αυτή τη [σελίδα](#). Αφού μεταφερθούμε στη σελίδα, κάνουμε κλικ στο κουμπί Start free όπως φαίνεται στην εικόνα:

Create your free Azure account today

-  **Get \$200 free credit**
Start free with \$200 in credit, and keep going with free options.
-  **Try any Azure services**
Explore our cloud by trying out any combination of Azure services for 30 days.
-  **Pay nothing at the end**
We use your credit card information for identity verification, but you'll never be charged unless you choose to subscribe.

[Start free >](#)



Εικόνα 18: Δημιουργία λογαριασμού στο Azure

Ανακατευθυνόμαστε σε μία νέα σελίδα όπου μας ζητάει να κάνουμε log in με κάποιον λογαριασμό Microsoft. Μετά το log in μεταφερόμαστε στη σελίδα του Azure με όλες τις παροχές του. Αν προσπαθήσουμε να χρησιμοποιήσουμε κάποια από αυτές, μας ζητάει πρώτα να πιστοποιήσουμε τον λογαριασμό μας μεταφέροντάς μας σε μία νέα σελίδα όπου μας ζητάει να εισάγουμε τηλέφωνο, πιστωτική κάρτα και άλλα στοιχεία. Αφού τα συμπληρώσουμε όλα είμαστε έτοιμοι να ξεκινήσουμε.

Η παραπάνω διαδικασία, αφορά λογαριασμό επί πληρωμή για άτομα που δεν έχουν δικαίωμα δωρεάν λογαριασμού και δεν αναλύθηκε λεπτομερώς. Για την απόκτηση δωρεάν λογαριασμού για φοιτητές που το δικαιούνται ακολουθείται ένας διαφορετικός τρόπος και επειδή είναι και ο τρόπος που χρησιμοποιήθηκε για την εργασία αυτή, θα αναλυθεί περισσότερο.

Ο τρόπος που χρησιμοποιείται εδώ είναι το Microsoft Imagine το οποίο είναι μία υπηρεσία η οποία συνεργάζεται με πανεπιστήμια και προσφέρει στους φοιτητές τους δωρεάν παροχές. Μία από αυτές είναι και το Azure σε κάποιο βαθμό. Το Microsoft Imagine είναι συνδεδεμένο με τον ακαδημαϊκό λογαριασμό του φοιτητή, επομένως πρέπει αρχικά να συνδεθεί με ένα λογαριασμό Hotmail. Για να γίνει αυτό πηγαίνουμε σε αυτή τη [σελίδα](#) και επιλέγουμε Create a Microsoft Imagine profile όπως φαίνεται στην εικόνα:

Student Account

By creating a Microsoft Imagine student profile, you can access developer tools, resources, contests and more.

[Create a Microsoft Imagine profile >](#)

Why sign up for Microsoft Imagine?

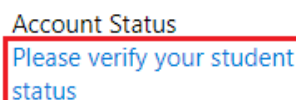
Create your student account and verify your student status to download the same tools and products used by professional developers.

Why verify as a student?

Microsoft Imagine software benefits are only available for current full-time students. Please verify with your school-issued email address, a special activation code, or with your ISIC card.

Εικόνα 19: Δημιουργία λογαριασμού Imagine

Αφού πατήσουμε σε αυτό το κουμπί, οδηγούμαστε σε μία καινούρια σελίδα όπου μας ζητάει να κάνουμε log in με κάποιο λογαριασμό της Microsoft. Μετά το log in ανακατευθυνόμαστε στη σελίδα του λογαριασμού μας στο Imagine. Από εκεί επιλέγουμε Please verify your student status όπως φαίνεται στην εικόνα ώστε να συνδέσουμε τον ακαδημαϊκό λογαριασμό μας, ο οποίος είναι συμβεβλημένος με το Microsoft Imagine, με το λογαριασμό Hotmail που μόλις γράψαμε.



Account Status
Please verify your student status

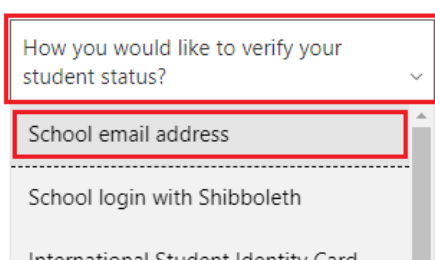
Εικόνα 17: Επικύρωση λογαριασμού

Στην επόμενη σελίδα μας ρωτάει με ποιον τρόπο θέλουμε να ενεργοποιήσουμε τον λογαριασμό Imagine. Εμείς επιλέγουμε ότι θέλουμε να γίνει ενεργοποίηση μέσω ακαδημαϊκού email και συνεχίζουμε.

Student Verification

Verify your student status

Verified students get full access to all our software and services at no cost. You can re-verify your status every year as long as you're a student. See [benefits](#) for more information.



How you would like to verify your student status? ▾

- School email address
- School login with Shibboleth
- International Student Identity Card

Εικόνα 20: Επικύρωση λογαριασμού μέσω πανεπιστημιακού email

Τέλος, βάζουμε το ακαδημαϊκό e-mail μας στα δύο πεδία που εμφανίζονται παρακάτω, πατάμε το κουμπί submit και είμαστε έτοιμοι να χρησιμοποιήσουμε τον Azure λογαριασμό μας.

Enter your school email address. If your school is in our database, we will email you a verification link.

*Indicates required information

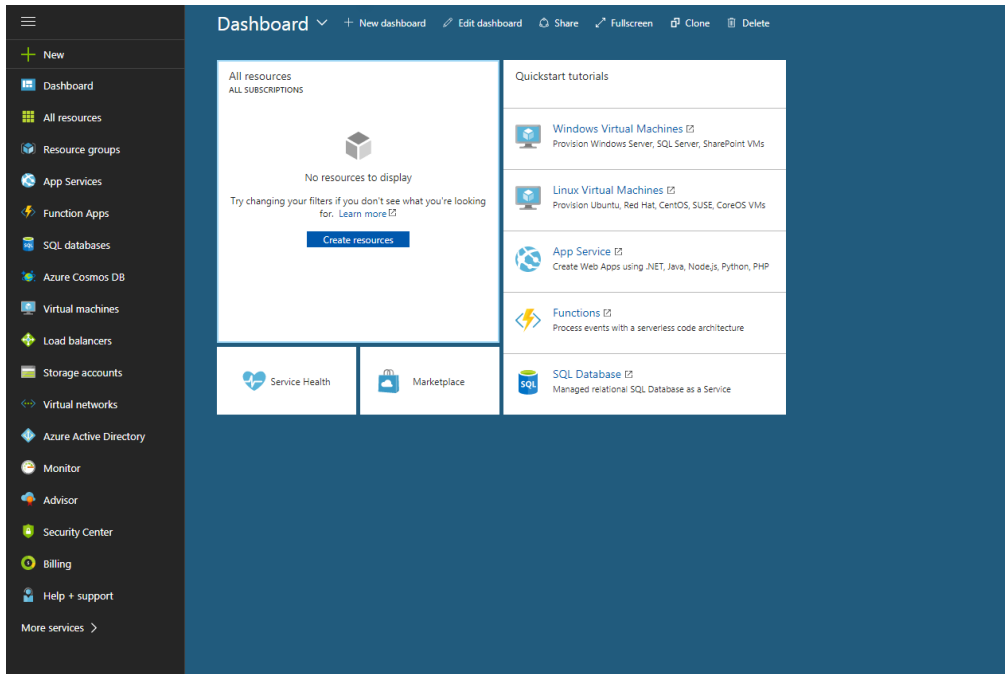
School email address*

Re-enter school email address*

Submit

Εικόνα 21: Ολοκλήρωση διαδικασίας ενεργοποίησης

Έχοντας πλέον συνδέσει έναν Hotmail λογαριασμό με το Microsoft Imagine μπορούμε να ξεκινήσουμε να χρησιμοποιούμε το Azure. Για να συνδεθούμε επισκεπτόμαστε τη σελίδα [εδώ](#) και χρησιμοποιούμε το e-mail του προηγούμενου βήματος. Αφού βάλουμε τα στοιχεία, γίνεται αυθεντικοποίηση αυτών και μεταφερόμαστε στην κεντρική σελίδα του Azure.

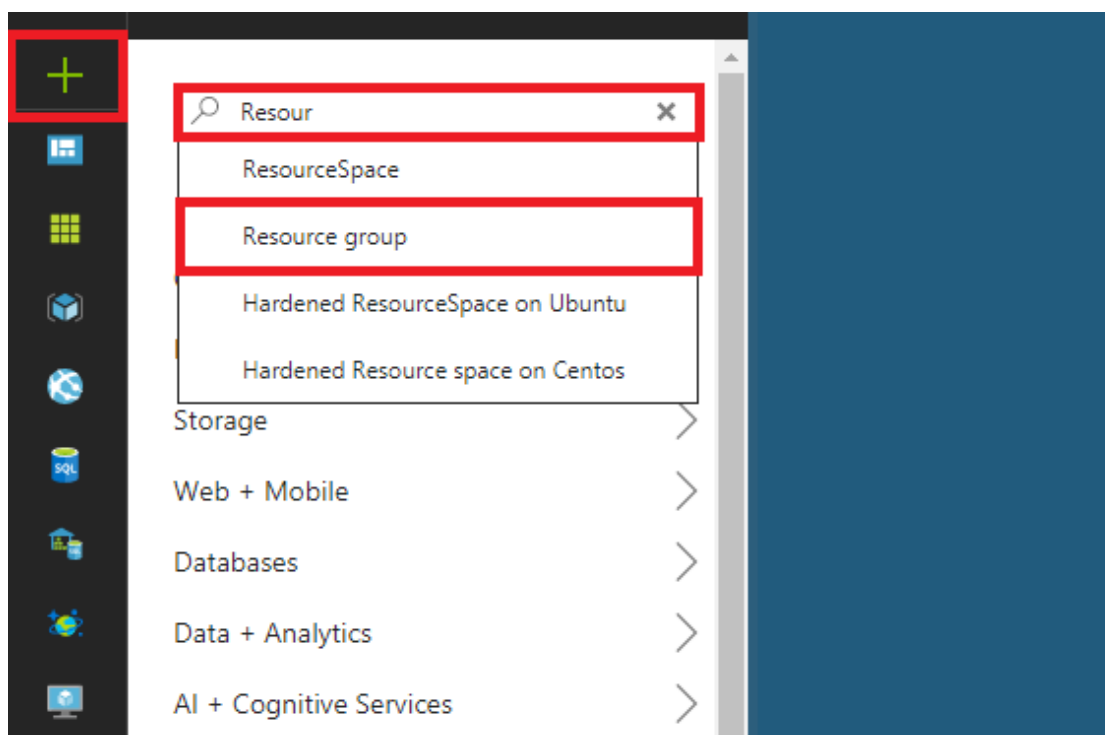


Εικόνα 22 Αρχική σελίδα του Azure

3.6 Κατασκευή server και βάσης δεδομένων

Το πρώτο βήμα που πρέπει να κάνουμε πριν ξεκινήσουμε με την δημιουργία της βάσης, είναι να δημιουργήσουμε ένα Resource Group το οποίο χρειάζεται σε όλες τις υπηρεσίες του Azure. Για να το κάνουμε αυτό πατάμε New και μετά το ψάχνουμε από

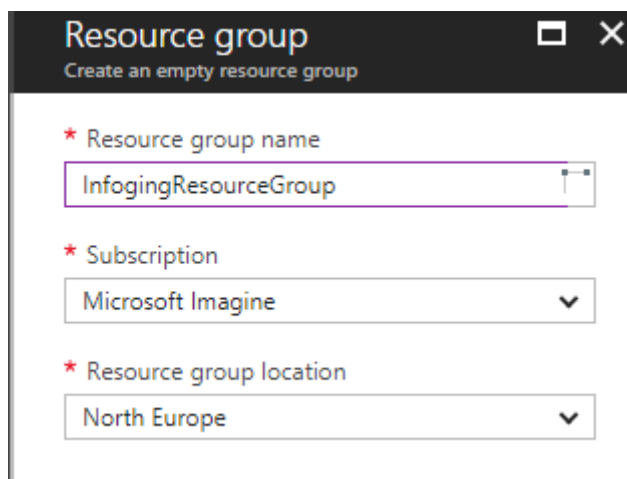
το αντίστοιχο πεδίο αναζήτησης, μόλις το βρούμε το επιλέγουμε και στο καινούριο παράθυρο που εμφανίζεται πατάμε το κουμπί Create.



Εικόνα 23: Δημιουργία Resource Group

Αφού πατήσουμε Create εμφανίζεται ένα νέο παράθυρο το οποίο ζητάει να συμπληρώσουμε το όνομα, τη συνδρομή και την τοποθεσία. Μπορούμε να δώσουμε όποιο όνομα επιθυμούμε. Στη συνδρομή βάζουμε το Microsoft Imagine ενώ στην τοποθεσία βάζουμε την πιο κοντινή σε εμάς για πιο σταθερές υπηρεσίες, δηλαδή το North Europe.

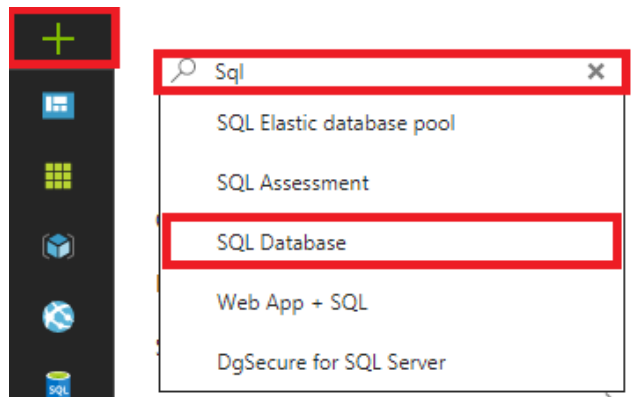
Τελειώνοντας με τη συμπλήρωση των στοιχείων πατάμε το κουμπί Create που βρίσκεται στο κάτω μέρος της σελίδας και περιμένουμε να ετοιμαστεί το Resource group ώστε να είναι έτοιμο για χρήση.



Εικόνα 24: Συμπλήρωση στοιχείων στο Resource Group

Έχοντας τελειώσει και από εδώ μπορούμε να προχωρήσουμε με τη δημιουργία της βάσης και του server αυτής.

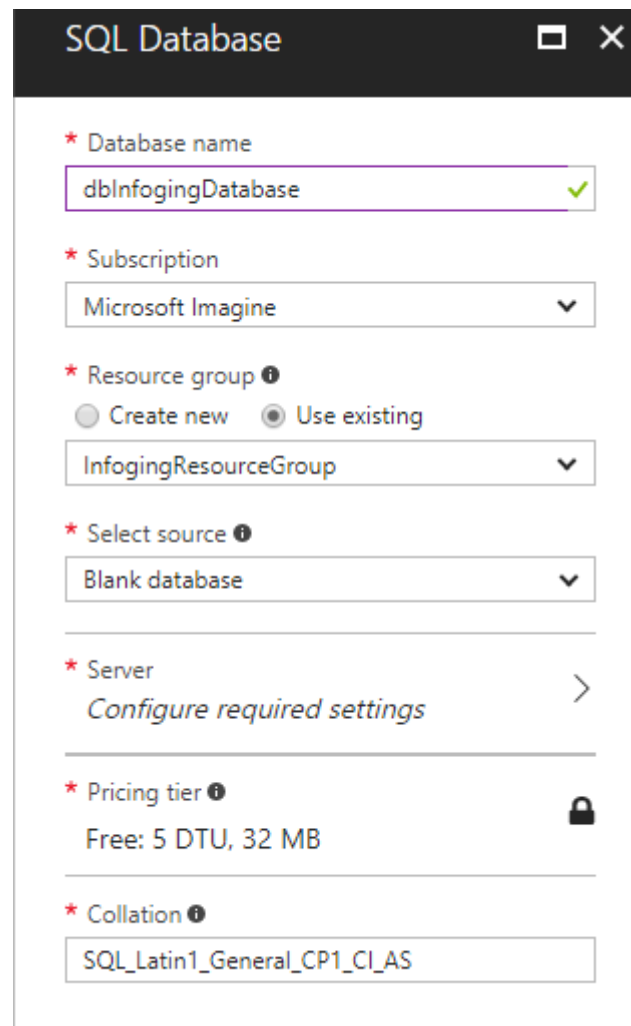
Για να δημιουργήσουμε τη βάση δεδομένων του site ακολουθούμε τη ίδια διαδικασία με πριν. Επιλέγουμε το New και μετά πληκτρολογούμε SQL Database μέσα στο πεδία της αναζήτησης ώστε να βρούμε και να επιλέξουμε την επιλογή SQL Database. Αφού την επιλέξουμε ανοίγει ένα νέο παράθυρο ενημερώνοντας μας σχετικά με τη βάση. Μέσα από αυτό επιλέγουμε Create (στο κάτω μέρος της οθόνης).



Εικόνα 25: Δημιουργία βάσης δεδομένων

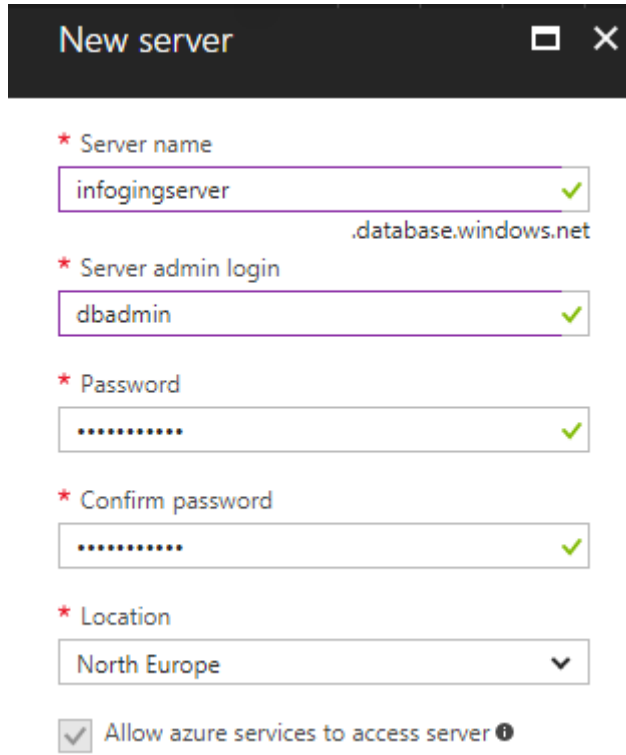
Αφού πατήσουμε το κουμπί εμφανίζεται μενού όπως αυτό που φαίνεται δεξιά. Στο πεδίο του ονόματος δίνουμε ένα όνομα της αρεσκείας μας, αρκεί να είναι μοναδικό, κάτι για το οποίο μας ενημερώνει το σύστημα. Στη συνδρομή επιλέγουμε Microsoft Imagine, στο Resource Group βάζουμε αυτό που δημιουργήσαμε πριν, ενώ την πηγή την αφήνουμε στην προεπιλεγμένη επιλογή που είναι η Blank database αφού τώρα θα αρχίσουμε να γεμίζουμε τη βάση μας. Στο Pricing Tier επιλέγουμε αυτό που μας προσφέρει η συνδρομή Imagine ενώ το Collation το αφήνουμε όπως είναι.

Το επόμενο βήμα που πρέπει να γίνει είναι η δημιουργία του Server για τη βάση. Πατώντας στην επιλογή Server που φαίνεται στην εικόνα δεξιά εμφανίζεται ένα καινούριο μενού από το οποίο επιλέγουμε Create new server.



Εικόνα 26: Συμπλήρωση στοιχείων για τη δημιουργία της βάσης

Μετά την επιλογή Create new server εμφανίζεται το μενού που φαίνεται στην εικόνα δεξιά. Και εδώ πρέπει να συμπληρώσουμε τα απαραίτητα στοιχεία. Προσθέτουμε το όνομα που επιθυμούμε στον server καθώς και ένα username και κάποιο password τα οποία θα τα χρησιμοποιήσουμε στη συνέχεια για να συνδεθούμε με τον server και κατ' επέκταση με τη βάση από το project που δημιουργήσαμε στο Visual Studio. Το τελευταίο πεδίο που πρέπει να συμπληρώσουμε είναι η τοποθεσία, όπου επιλέγουμε ξανά North Europe. Το Azure μας ενημερώνει εάν τα στοιχεία που έχουμε εισάγει είναι σωστά και εφόσον έχουμε ολοκληρώσει πατάμε Select (βρίσκεται στο κάτω μέρος του μενού) ώστε να επιλεγεί ο server.



Εικόνα 27: Συμπλήρωση στοιχείων για τον server

Τέλος πατάμε Create για να δημιουργηθούν η βάση και ο server. Μόλις είναι έτοιμα, το Azure μας ενημερώνει σχετικά και πλέον μπορούμε να τα χρησιμοποιήσουμε. Είναι πολύ σημαντικό να θυμόμαστε τα στοιχεία αυτά γιατί θα μας χρειαστούν μετά για τη σύνδεση του project στο Visual Studio με τη βάση.

3.7 Σύνδεση της βάσης με το project

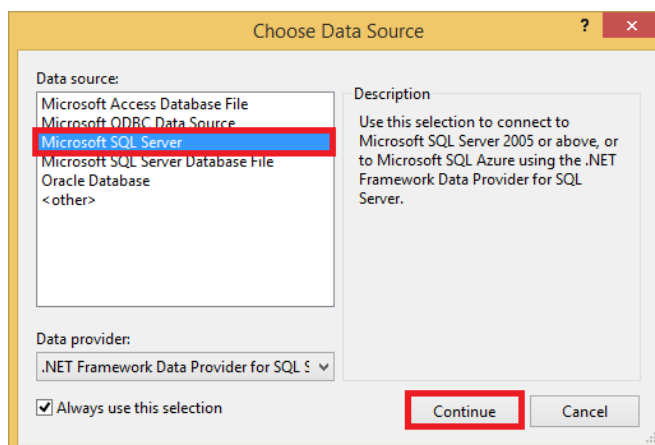
Για να τα χρησιμοποιήσουμε λοιπόν, επιστρέφουμε στο project μας στο Visual Studio. Σε περίπτωση που το έχουμε κλείσει μπορούμε να το επαναφέρουμε ακολουθώντας τα παρακάτω:

File → Open → Project/Solution...

Αφού επιλέξουμε τα παραπάνω, μας εμφανίζει ένα παράθυρο περιήγησης με τα project που έχουμε δημιουργήσει. Κάνουμε διπλό κλικ στο δικό μας, επιλέγουμε το αρχείο με επέκταση .sln και πατάμε άνοιγμα.

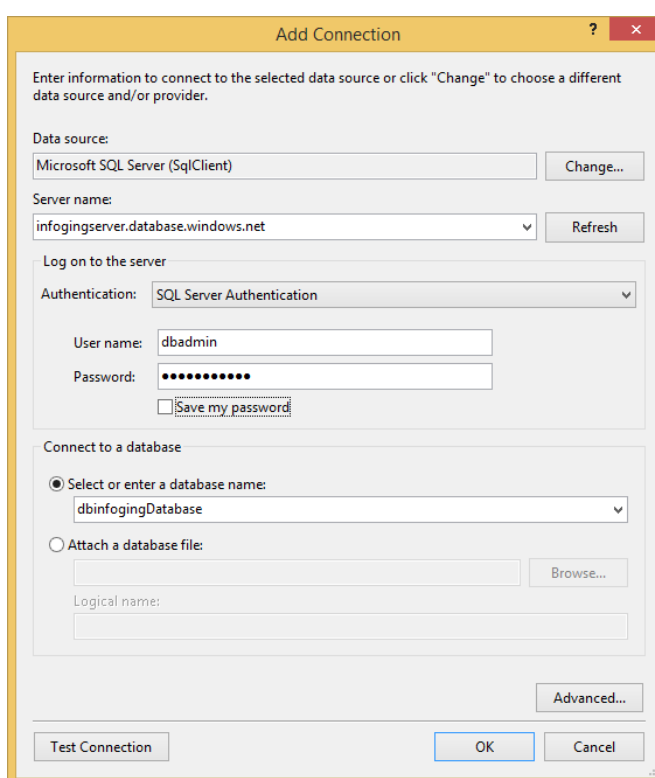
Έχοντας το project μας ενεργό πηγαίνουμε αριστερά και βρίσκουμε την επιλογή Server Explorer. Εδώ υπάρχουν 3 πιθανές περιπτώσεις. Μία περίπτωση είναι να είναι ήδη ενεργό σε μία στήλη στα αριστερά του περιβάλλοντος οπότε και έχουμε άμεση πρόσβαση. Η δεύτερη περίπτωση είναι να είναι ενεργό, οπότε και θα είναι γραμμένο κάθετα στα αριστερά του περιβάλλοντος. Η Τρίτη και τελευταία περίπτωση είναι να μην είναι ενεργοποιημένο. Για να το ενεργοποιήσουμε, επιλέγουμε από το οριζόντιο μενού στο πάνω μέρος του περιβάλλοντος View → Server Explorer ή πατώντας τον συνδυασμό των πλήκτρων Ctrl + Alt + S.

Πλέον, αφού έχουμε μπροστά μας τον Server Explorer κάνουμε δεξί κλικ στο Data Connections και επιλέγουμε Add Connection. Έπειτα εμφανίζεται στην οθόνη το παράθυρο που φαίνεται στην εικόνα, το οποίο μας ζητάει να επιλέξουμε την πηγή των δεδομένων της σύνδεσης. Επιλέγουμε Microsoft SQL Server και πατάμε Continue.



Εικόνα 28: Επιλογή Data Source

Πατώντας Continue εμφανίζεται στην οθόνη ένα νέο παράθυρο μέσα στο οποίο καλούμαστε να συμπληρώσουμε τα στοιχεία του server. Τα στοιχεία αυτά είναι εκείνα που εισάγαμε όταν κατασκευάσαμε τον server και τη βάση στο Azure. Στο Authentication έχει στην αρχή Windows Authentication ως επιλογή οπότε εμείς το αλλάζουμε σε SQL Server Authentication. Αφού δώσουμε όλα τα απαραίτητα δεδομένα πατάμε OK και περιμένουμε έως ότου πραγματοποιηθεί η σύνδεση με τη βάση. Μόλις γίνει αυτό κάνουμε ξανά δεξί κλικ στο Data Connections και πατάμε Refresh.



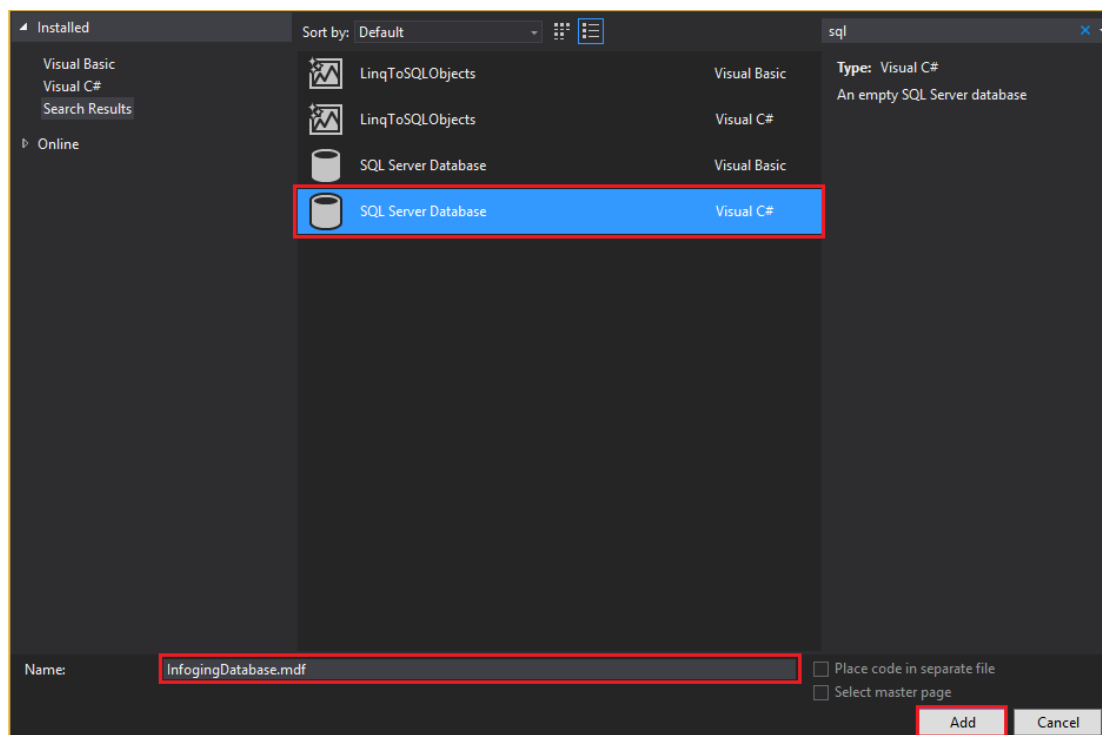
Εικόνα 29: Συμπλήρωση στοιχείων στον server

Αν έχουμε ολοκληρώσει με επιτυχία όλα τα προηγούμενα βήματα, έχει συνδεθεί επιτυχώς η βάση και είμαστε έτοιμοι να δημιουργήσουμε τους πίνακες μας.

3.7.1 Εναλλακτικός τρόπος (Δημιουργία τοπικής βάσης)

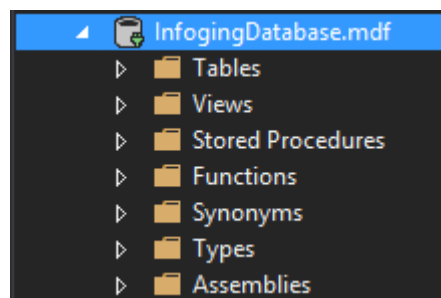
Η παραπάνω μέθοδος ακολουθήθηκε ώστε να δημιουργηθεί και να χρησιμοποιηθεί μία βάση δεδομένων μέσω του Azure καθώς το site θα δημοσιευτεί εκεί για να είναι προσβάσιμο από όλους. Μία εναλλακτική λύση όμως που υπάρχει σε περίπτωση που κάποιος δεν επιθυμεί να δημοσιεύσει το site, είναι η δημιουργία τοπικής βάσης δεδομένων. Παρακάτω εξηγείται ο τρόπος με τον οποίο μπορεί να κατασκευαστεί μία τέτοια βάση. Η διαδικασία είναι πολύ πιο απλή και εύκολη.

Στα δεξιά του περιβάλλοντος υπάρχει η στήλη Solution Explorer όπου βρίσκονται τα αρχεία του project. Στην κορυφή υπάρχει το όνομα του project και από κάτω εμφωλευμένα όλα τα αρχεία. Κάνουμε δεξί κλικ στο όνομα του project και επιλέγουμε Add → Add New Item... ή πατώντας τα πλήκτρα Ctrl + Alt + A. Αφού ακολουθήσουμε τη διαδικασία αυτή, εμφανίζεται ένα νέο παράθυρο με όλα τα διαθέσιμα συστατικά τα οποία μπορούμε να χρησιμοποιήσουμε στον ιστότοπο. Πάνω δεξιά, στο πεδίο της αναζήτησης, πληκτρολογούμε sql για να φιλτράρουμε τα αποτελέσματα. Από διαθέσιμα αποτελέσματα επιλέγουμε SQL Server Database με γλώσσα την Visual C#, δίνουμε ένα όνομα της αρεσκείας μας στη βάση και πατάμε Add.



Εικόνα 30: Δημιουργία τοπικής βάσης

Έχοντας κάνει τα παραπάνω βήματα, εμφανίζεται στην οθόνη ένα προειδοποιητικό μήνυμα σχετικά με την τοποθεσία αποθήκευσης της βάσης. Πατάμε Ναι και περιμένουμε να δημιουργηθεί η βάση. Μόλις είναι έτοιμη, το Visual Studio θα ανανεώσει το Solution Explorer και πλέον μπορούμε να δούμε μέσα την καινούρια μας βάση και να τη χρησιμοποιήσουμε.

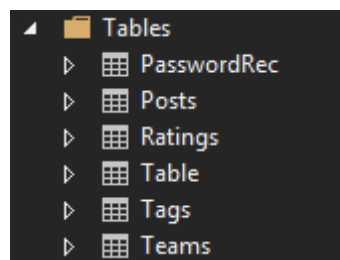


Εικόνα 31: Βάση έτοιμη για χρήση

Αφού δημιουργήσουμε, με τον ένα ή με τον άλλο τρόπο τη βάση δεδομένων, είμαστε έτοιμοι να ξεκινήσουμε την κατασκευή των πινάκων που θα χρησιμοποιήσουμε στον ιστότοπο. Το Visual Studio, μέσω του Solution Explorer, παρέχει ένα εύχρηστο γραφικό περιβάλλον για να πετύχουμε τον σκοπό μας. Στην επόμενη ενότητα εξηγείται λεπτομερώς πως θα το κάνουμε αυτό.

3.8 Κατασκευή πινάκων

Οι πίνακες οι οποίοι πρέπει να δημιουργηθούν είναι έξι όπως φαίνεται και στην *Εικόνα 30*. Για να δημιουργήσουμε έναν καινούριο πίνακα κάνουμε δεξί κλικ στον φάκελο Tables κάτω από τη βάση μέσα στο Server Explorer και επιλέγουμε Add New Table. Το Visual Studio ανοίγει μία καρτέλα για τη δημιουργία των απαραίτητων πεδίων και την επιλογή του ονόματος για τον πίνακα. Οι ενέργειες αυτές μπορούν να γίνουν είτε



Εικόνα 32: Απαραίτητη Πίνακες

γραφικά (πάνω μέρος) όπου πληκτρολογούμε τα ονόματα στα αντίστοιχα κελιά και επιλέγουμε τον τύπο τους, είτε προγραμματιστικά (κάτω μέρος) όπου δίνουμε προγραμματιστικά τα ονόματα των κελιών και τους τύπους τους σε γλώσσα T-SQL. Αφού τελειώσουμε με τη συμπλήρωση των πεδίων, με οποιονδήποτε από τους δύο τρόπους, πατάμε το κουμπί Update που βρίσκεται πάνω από τα κελιά για να ενημερωθεί η βάση στον server σχετικά με την αλλαγή. Μόλις πατήσουμε Update, εμφανίζεται ένα νέο παράθυρο το οποίο μας ενημερώνει σχετικά με τις συνέπειες στη βάση από τις αλλαγές που έχουμε κάνει. Στο παράθυρο αυτό πατάμε Update Database και περιμένουμε έως ότου καταχωρηθούν οι αλλαγές μας.

Σε περίπτωση που θέλουμε για κάποιο λόγο να αλλάξουμε κάτι σε κάποιον πίνακα που έχουμε δημιουργήσει πηγαίνουμε ξανά στον φάκελο Tables, εμφανίζουμε όλους τους πίνακες, κάνουμε δεξί κλικ σε αυτόν που θέλουμε να τροποποιήσουμε και επιλέγουμε Open Table Definition. Ακολουθώντας την ίδια διαδικασία με πριν, κάνουμε τις αλλαγές που χρειαζόμαστε και πατάμε Update.

Παρακάτω αναλύεται ο κάθε πίνακας που πρέπει να δημιουργηθεί καθώς και ο ρόλος του:

- *PasswordRec*: Ο πίνακας, χρησιμεύει για την αλλαγή του κωδικού ενός χρήστη όταν ο τελευταίος τον έχει ξεχάσει και πραγματοποιήσει αίτημα αλλαγής αλλά και για την αυθεντικοποίηση του χρήστη μετά την εγγραφή του.
 - 1) *UserGUID – nvarchar(250)*: Αποθηκεύει το μοναδικό ID του χρήστη.
 - 2) *Codes – nvarchar(250)*: Αποθηκεύει έναν μοναδικό κωδικό αλλαγής συνθηματικού.
- *Posts*: Στον πίνακα αυτό αποθηκεύονται τα στοιχεία για κάθε δημοσίευση που κάνουν οι χρήστες.
 - 1) *id – int*: Χαρακτηρίζει μοναδικά ένα ποστ αλλά δεν χρησιμοποιείται πολύ. Μπορεί να τεθεί σε AUTO INCREMENT ώστε να αυξάνει κατά μία μονάδα με κάθε νέα εγγραφή.
 - 2) *username – nvarchar(50)*: Εδώ αποθηκεύεται το όνομα του χρήστη που έκανε τη δημοσίευση
 - 3) *text – nvarchar(250)*: Δέχεται το περιεχόμενο της δημοσίευσης.
 - 4) *fromDate – date*: Αποθηκεύει την ημερομηνία έναρξης του συμβάντος.

- 5) toDate – date: Αποθηκεύει την ημερομηνία λήξης του συμβάντος.
 - 6) location – nvarchar(250): Δέχεται την τοποθεσία που λαμβάνει χώρα η συγκεκριμένη δημοσίευση.
 - 7) latitude – float: Γεωγραφικό πλάτος της παραπάνω τοποθεσίας.
 - 8) longtitude – float: Γεωγραφικό μήκος της παραπάνω τοποθεσίας.
 - 9) Geoloc – sys.geography: Ειδικός τύπος στον SQL Server ο οποίος μπορεί να αποθηκεύσει ένα στίγμα το οποίο δημιουργείται από ένα γεωγραφικό πλάτος και μήκος και μπορεί να συγκριθεί με άλλα τέτοια στίγματα για να βρεθεί η γεωγραφική απόσταση τους.
 - 10) limit – nvarchar(250): Αποθηκεύει τον περιορισμό ως προς τις ομάδες ατόμων οι οποίες θα μπορούν να προβάλουν την συγκεκριμένη δημοσίευση.
 - 11) usefull – int: Κρατάει το σύνολο των θετικών βαθμολογιών.
 - 12) useless – int: Κρατάει το σύνολο των αρνητικών βαθμολογιών.
 - 13) tags – nvarchar(MAX): Δέχεται τις ετικέτες οποίες έχουν δηλωθεί για τη συγκεκριμένη δημοσίευση.
 - 14) userPicture – nvarchar(MAX): Αποθηκεύει τη διεύθυνση της φωτογραφίας του χρήστη που έκανε τη δημοσίευση.
- *Ratings*: Στον πίνακα αυτό αποθηκεύονται όλα τα δεδομένα σχετικά τις βαθμολογίες που αφήνουν οι χρήστες στις δημοσιεύσεις. Τα απαραίτητα πεδία είναι:
 - 1) Id – int: Αποθηκεύει έναν αύξοντα αριθμό. Με κάθε νέα βαθμολογία η τιμή αυτού του πεδίου ανεβαίνει αυτόματα κατά μία μονάδα. Για να γίνει αυτό θέτουμε το πεδίο INT και από δίπλα στην T-SQL γράφουμε IDENTITY(1,1).
 - 2) postID – int: Δέχεται το μοναδικό ID του post στο οποίο έγινε η βαθμολογία.
 - 3) username – nvarchar(250): Κρατάει το όνομα χρήστη αυτού που έκανε τη δημοσίευση.
 - 4) usefull – int: Boolean τιμή η οποία παίρνει τιμή 1 αν ήταν θετική η βαθμολογία ενώ παίρνει 0 διαφορετικά.
 - 5) useless – int: Boolean τιμή η οποία παίρνει τιμή 1 αν ήταν αρνητική η βαθμολογία ενώ παίρνει 0 διαφορετικά.
 - 6) timestamp – date: Δέχεται την ημερομηνία που έγινε η βαθμολογία
 - 7) deleted – int: Boolean τιμή η οποία παίρνει τιμή 1 αν έχει διαγραφεί η βαθμολογία ενώ παίρνει 0 διαφορετικά.

- *Table*: Εδώ αποθηκεύονται τα στοιχεία των χρηστών οι οποίοι εγγράφονται στον ιστότοπο. Η χρήση κάποιων πεδίων είναι εμφανής επομένως δεν έχει γίνει κάποια επεξήγηση για τα πεδία αυτά.
 - 1) *ID – nvarchar(250)*: Αλφαριθμητικό το οποίο προσδιορίζει μονοσήμαντα έναν χρήστη.
 - 2) *name – nvarchar(50)*
 - 3) *surname – nvarchar(50)*
 - 4) *e-mail – nvarchar(50)*
 - 5) *username – nvarchar(50)*
 - 6) *password – nvarchar(50)*
 - 7) *location – nvarchar(250)*: Αποθηκεύει τον τόπο διαμονής του χρήστη.
 - 8) *education – nvarchar(250)*: Δέχεται το επίπεδο εκπαίδευσης του χρήστη.
 - 9) *activated – int*: Boolean πεδίο το οποίο παίρνει τιμή 1 αν ο συγκεκριμένος λογαριασμός ενεργοποιηθεί και 0 διαφορετικά.
 - 10) *profilePic – varbinary(MAX)*: Ειδικό πεδίο αποθήκευσης φωτογραφιών το οποίο μεταφράζει την εικόνα σε byte.
 - 11) *uploadedPicture – int*: Boolean πεδίο το οποίο παίρνει τιμή 0 αν ο χρήστης έχει ανεβάσει δικιά του φωτογραφία και 0 διαφορετικά.
- *Tags*: Στον πίνακα αυτό αποθηκεύονται οι ετικέτες που προστίθενται στις διάφορες δημοσιεύσεις. Τα πεδία του είναι:
 - 1) *TagID – int*: Αποθηκεύει έναν ακέραιο αριθμό ο οποίος προσδιορίζει το συγκεκριμένο tag.
 - 2) *TagName – nvarchar(MAX)*: Δέχεται το όνομα της ετικέτας.
 - 3) *TimesUsed – int*: Αποθηκεύει τον αριθμό των φορών που έχει χρησιμοποιηθεί η ετικέτα.
- *Teams*: Εδώ αποθηκεύονται οι ομάδες που δημιουργούν οι χρήστες. Τα πεδία που έχει αυτός ο πίνακας είναι:
 - 1) *id – int*: Ακέραιος αριθμός που προσδιορίζει την ομάδα.
 - 2) *teamName – nvarchar(250)*: Το όνομα της ομάδας
 - 3) *username – nvarchar(250)*: Το όνομα του χρήστη που προστίθεται στην ομάδα.
 - 4) *owner – nvarchar(250)*: Το όνομα του δημιουργού της ομάδας.
 - 5) *userPicture – nvarchar(250)*: Η φωτογραφία του χρήστη που προστέθηκε στην ομάδα.

Αφού ολοκληρωθεί η διαδικασία κατασκευής των πινάκων που θα χρησιμοποιηθούν, μπορεί πλέον να ξεκινήσει η κατασκευή του ιστοτόπου.

3.9 Google Maps JavaScript API

Οι πληροφορίες που θα προβάλλει ο χρήστης φιλτράρονται και τοπικά. Αυτό γίνεται μέσω ενός χάρτη της Google. Για να αποκτήσει κανείς πρόσβαση στους χάρτες αυτούς είναι απαραίτητο να δημιουργήσει ένα API Key, το οποίο είναι ένα κλειδί που αναγνωρίζει ποιος καλεί την υπηρεσία και παρέχει στον χρήστη πρόσβαση στο χάρτη και τα διάφορα χαρακτηριστικά του.

Για τη δημιουργία του κλειδιού πρέπει να επισκεφθούμε αρχικά το Google Cloud Platform (παλαιότερα ονομαζόταν Google API Console) από τη διεύθυνση [εδώ](#) και να εισέλθουμε στο σύστημα χρησιμοποιώντας έναν λογαριασμό Gmail.

Στο πλαίσιο της αναζήτησης στο πάνω μέρος της οθόνης, πληκτρολογούμε Google Maps JavaScript API και επιλέγουμε την επιλογή που μας βγάζει. Στην συνέχεια το σύστημα μας ειδοποιεί ότι απαιτείται η δημιουργία ενός project για να προβάλλουμε το περιεχόμενο και για το λόγο αυτό πατάμε το κουμπί Create. Στη σελίδα που μεταφερόμαστε δίνουμε ένα όνομα στο Project και πατάμε Create.

Αφού δημιουργήσαμε το project, επιλέγουμε Create Credentials και στη λίστα που μας εμφανίζει πατάμε API Key. Εμφανίζεται ένα μήνυμα που μας ενημερώνει για τη δημιουργία του κλειδιού και μας προτρέπει να το περιορίσουμε. Πατάμε CLOSE και το κλειδί είναι έτοιμο για χρήση.

Αργότερα, μέσω αυτής της πλατφόρμας, μπορούμε να ελέγχουμε την χρήση του κλειδιού, δηλαδή πόσες κλήσεις έχουν γίνει συνολικά ή σε συγκεκριμένα χρονικά διαστήματα και πολλά άλλα.

3.10 Δημιουργία ιστοτόπου

Το Visual Studio, προσφέρει πολλά χαρακτηριστικά ώστε να βοηθήσει στην κατασκευή ενός ιστοτόπου. Περιέχει μια εργαλειοθήκη (Toolbox), η οποία βρίσκεται στα αριστερό μέρος του περιβάλλοντος και περιέχει εργαλεία τα οποία βοηθούν σε αυτή την προσπάθεια. Η εργαλειοθήκη αυτή, όπως συνέβαινε και με τον Server Explorer είτε θα βρίσκεται ήδη ενεργή δίπλα στην καρτέλα του Server Explorer είτε θα υπάρχει αλλά δεν θα είναι ενεργή είτε δεν θα υπάρχει καθόλου. Σε αυτή την περίπτωση, για να την εμφανίσουμε, πατάμε από το μενού εργαλείων View → Toolbox ή τον συνδυασμό των πλήκτρων Ctrl + Alt + X.

Τα εργαλεία που παρέχονται, μπορούν να αντικαταστήσουν τις ετικέτες της HTML αλλά υπάρχουν και εργαλεία με διαφορετικές δυνατότητες τα οποία είναι πολύ χρήσιμα. Η εισαγωγή τους στη σελίδα γίνεται πολύ εύκολα με drag 'n' drop στο σημείο που τα θέλουμε μέσα σε αυτή. Η χρήση αυτών είναι σημαντική έναντι των ετικετών της HTML σε μεγάλο βαθμό γιατί αυτά μπορούμε ευκολότερα να τα ελέγξουμε πίσω στον server σε κάθε κλήση συγκριτικά με την HTML.

Ένα ακόμη πλεονέκτημα που παρέχει το ASP.NET είναι η χρήση των MasterPage. Ένα MasterPage μπορεί να γίνει ο πατέρας πολλών σελίδων. Δηλαδή μπορούμε να επιλέξουμε κάποιες ιστοσελίδες, οι οποίες θα κληρονομήσουν τη σελίδα αυτή. Μπορούν δηλαδή να κληρονομήσουν αρχεία CSS, μεθόδους και άλλα. Ακόμη ένα σημαντικό χαρακτηριστικό είναι ότι μπορούμε να διαμορφώσουμε την MasterPage όπως θέλουμε και να εμφωλεύσουμε σελίδες μέσα σε αυτή. Δηλαδή γενικά η δομή του

MasterPage θα μένει ίδια σε όλες τις σελίδες που την κληρονομούν και το μόνο που θα αλλάζει θα είναι το εσωτερικό κομμάτι της κάθε σελίδας.

Γενικά, η διαδικασία για να δημιουργήσουμε μια MasterPage αλλά και ιστοσελίδες είναι η ίδια. Ακολουθούμε το εξής μονοπάτι:

Δεξί κλικ στο Solution → Add.. → Add New Item...

Στα αριστερά του παραθύρου που εμφανίζεται, ελέγχουμε αν η επιλεγμένη γλώσσα είναι η Visual C#. Αν δεν είναι, την επιλέγουμε και ψάχνουμε να βρούμε αυτό που χρειαζόμαστε. Για τη δημιουργία MasterPage, επιλέγουμε MasterPage ενώ για τη δημιουργία ιστοσελίδας επιλέγουμε Web Form. Αν θέλουμε να τους αλλάξουμε όνομα, μπορούμε να το κάνουμε στο αντίστοιχο πεδίο στο κάτω μέρος του παραθύρου. Στην περίπτωση που θέλουμε να συνδέσουμε τη σελίδα με κάποιο MasterPage, τσεκάρουμε την επιλογή Select Master Page στα δεξιά και πατάμε Add, όπου και θα μας προτρέψει να επιλέξουμε την MasterPage που επιθυμούμε.

Κάποιος που είναι εξοικειωμένος με την HTML και θέλει να χρησιμοποιήσει την HTML, μπορεί φυσικά να το κάνει αφού υποστηρίζεται πλήρως, αλλά θα πρέπει σε κάθε ετικέτα να προσθέτει την ιδιότητα `runat="server"` για να μπορεί να ελέγχει τις ετικέτες αυτές από την πλευρά του server με τη C#.

Οι ιστοσελίδες οι οποίες κατασκευάστηκαν στα πλαίσια αυτής της πτυχιακής είναι 10 στο σύνολο. Οι δύο από αυτές είναι Master Pages ενώ οι υπόλοιπες οκτώ είναι τύπου Web Form. Παρακάτω εξηγούνται οι μέθοδοι που χρησιμοποιούνται σε κάθε σελίδα καθώς και η λειτουργία τους.

3.10.1 Σελίδα εγγραφής και σύνδεσης

Για τη σελίδα αυτή, καθώς και για τις επόμενες 3, πρέπει να δημιουργηθεί ένα Master Page το οποίο θα κληρονομούν. Στο MasterPage αυτό κατασκευάζεται η κεφαλίδα της σελίδας η οποία θα είναι ίδια και στις 4 σελίδες. Οι λειτουργίες που θα έχει η πρώτη σελίδα είναι αυτές που φαίνονται στη εικόνα:

```
1 using System;
2 using System.Web.UI;
3 using System.Web.UI.WebControls;
4 using System.Data.SqlClient;
5 using System.Configuration;
6 using System.Net.Mail;
7 using System.Net;
8
9 public partial class _Default : System.Web.UI.Page
10 {
11
12     protected void Page_Load(object sender, EventArgs e)...
13
14
15
16
17
18
19
20
21
22
23
24
25     protected void registerButton_Click(object sender, EventArgs e)...
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45     private void SendActivationEmail()...
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82     protected void loginButton_Click(object sender, EventArgs e)...
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185 }
```

Εικόνα 33: Οι μέθοδοι της πρώτης σελίδας με τις απαραίτητες βιβλιοθήκες

- **registerButton_Click(object sender, EventArgs e):** είναι μία μέθοδος η οποία πυροδοτείται από το πάτημα του κουμπιού εγγραφής. Σκοπός της είναι να εγγράψει τον χρήστη στο σύστημα. Αρχικά δημιουργεί μία σύνδεση με τη βάση δεδομένων για να αποκτήσει πρόσβαση στους πίνακες της και ελέγχει αν το όνομα χρήστη ή το e-mail που εισήγαγε ένας χρήστης ανήκουν σε κάποιον άλλο χρήστη και εμφανίζει το αντίστοιχο προειδοποιητικό μήνυμα. Αν δεν υπάρξει τέτοιο ζήτημα τότε δημιουργεί ένα μοναδικό ID για τον χρήστη και εισάγει αυτό καθώς και όλα τα στοιχεία που χρησιμοποίησε αυτός στη βάση δεδομένων. Αφού γίνει η εγγραφή του, εμφανίζεται μήνυμα επιτυχίας και ανακατευθύνεται ο χρήστης στην καρτέλα σύνδεσης.
- **SendActivationEmail():** καλείται μέσω της registerButton() αφού η τελευταία εισάγει τα στοιχεία του χρήστη στη βάση. Η SendActivationEmail() είναι υπεύθυνη για την αποστολή ενός e-mail ενεργοποίησης λογαριασμού στον χρήστη που μόλις έκανε την εγγραφή του. Στη μέθοδο αυτή κατασκευάζεται ένα e-mail, δηλαδή το θέμα του, το σώμα του, ο SMTP Client που θα χρησιμοποιηθεί κτλ και στη συνέχεια το e-mail αυτό αποστέλλεται στον χρήστη. Το e-mail, περιέχει έναν σύνδεσμο, ο οποίος στέλνει τον χρήστη σε μία διεύθυνση URL η οποία έχει ως παράμετρο ένα μοναδικό Activation Code, το οποίο θα χρησιμοποιηθεί για την πιστοποίηση του χρήστη και την αυθεντικοποίηση του. Μέσα στην SendActivationEmail(), εκτός από την δημιουργία του e-mail, συνδέεται και ο χρήστης στον οποίο στάλθηκε το e-mail μαζί με αυτό το Activation Code στη βάση δεδομένων PasswordRec.

- **loginButton_Click(object sender, EventArgs e):** Η μέθοδος αυτή καλείται με το πάτημα του κουμπιού σύνδεσης. Αρχικά, δημιουργεί μια σύνδεση με τη βάση δεδομένων για να αποκτήσει πρόσβαση στους πίνακες της και να πραγματοποιήσει κάποιους απαραίτητους ελέγχους. Το πρώτο πράγμα που ελέγχεται είναι αν το όνομα χρήστη υπάρχει στη βάση, δηλαδή αν ο χρήστης είναι εγγεγραμμένος. Αν δεν υπάρχει εμφανίζει κατάλληλο προειδοποιητικό μήνυμα. Από την άλλη, αν υπάρχει, ελέγχει αν το συνθηματικό που εισήγαγε είναι το σωστό. Σε περίπτωση λανθασμένου κωδικού, εμφανίζει προειδοποιητικό μήνυμα. Μετά τον έλεγχο του κωδικού ελέγχεται αν έχει γίνει ενεργοποίηση του λογαριασμού. Αν αυτή αν είναι αρνητική, τότε έχουμε και εδώ την εμφάνιση του κατάλληλου μηνύματος. Τέλος, εφόσον έχει περάσει όλους τους ελέγχους με επιτυχία, δημιουργεί μία νέα συνεδρία στην οποία αποθηκεύει το όνομα χρήστη και ανακατευθύνει τον χρήστη στην αρχική του σελίδα.

Η δημιουργία και η ύπαρξη του της συνεδρίας με το όνομα του χρήστη αποθηκευμένο, είναι πολύ σημαντική γιατί χωρίς αυτή δεν λειτουργεί ο ιστότοπος. Για τον λόγο αυτό, σε περίπτωση που ο χρήστης ανοίξει κάποια σελίδα μέσω του browser χωρίς να έχει κάνει πρώτα σύνδεση, ανακατευθύνεται αμέσως στην αρχική σελίδα ώστε να γράψει πρώτα τα στοιχεία εισόδου του και στη συνέχεια να συνδεθεί ώστε να αποθηκευτεί το όνομα χρήστη του στη συνεδρία.

Στην περίπτωση αυτή, στο URL της πρώτης αυτής ιστοσελίδας υπάρχει η παράμετρος returnUrl. Έτσι, αφού περάσουν τα στοιχεία που εισήγαγε από όλους του ελέγχους που αναφέραμε πριν, δημιουργείται η συνεδρία και ο χρήστης μεταφέρεται στην σελίδα την οποία επισκέφθηκε αρχικά και όχι στη δικιά του αρχική σελίδα όπως έγινε πριν.

3.10.2 Σελίδα ενεργοποίησης λογαριασμού

Η σελίδα αυτή αφορά τη σελίδα ενεργοποίησης του λογαριασμού ενός χρήστη. Η μέθοδος μαζί με τις βιβλιοθήκες που χρησιμοποιούνται φαίνονται στην παρακάτω εικόνα:

```
1 using System;
2     using System.Collections.Generic;
3     using System.Configuration;
4     using System.Data.SqlClient;
5     using System.Linq;
6     using System.Web;
7     using System.Web.UI;
8     using System.Web.UI.WebControls;
9
10    public partial class account_activation : System.Web.UI.Page
11    {
12        protected void Page_Load(object sender, EventArgs e)
13    {
14    }
```

Εικόνα 34: Οι μέθοδοι της δεύτερης σελίδας με τις απαραίτητες βιβλιοθήκες

Ουσιαστικά δεν χρησιμοποιείται κάποια νέα μέθοδος. Η Page_Load() υπάρχει σε κάθε νέα σελίδα που δημιουργούμε και περιέχει τον κώδικα που εκτελείται κατά τη φόρτωση της σελίδας. Επειδή απαιτείται μόνο μία λειτουργία, δεν χρειάζεται η κατασκευή ξεχωριστής μεθόδου. Ο κώδικας ο οποίος εκτελείται μέσα στην Page_Load() αποθηκεύει το ActivationCode το οποίο υπάρχει ως παράμετρος στο URL και ελέγχει αν υπάρχει κάποια εγγραφή με αυτό το ActivationCode στον πίνακα PasswordRec. Αν δεν υπάρχει, σημαίνει ότι ο λογαριασμός αυτός έχει ήδη ενεργοποιηθεί. Διαφορετικά βρίσκει το μοναδικό ID του χρήστη, ενημερώνει τον πίνακα των χρηστών ότι ο χρήστης αυθεντικοποίησε τον λογαριασμό του και διαγράφει την εγγραφή του κωδικού και του χρήστη από τον πίνακα PasswordRec.

3.10.3 Σελίδα έναρξης διαδικασίας αλλαγής κωδικού

Η σελίδα αυτή αφορά την αλλαγή του κωδικού πρόσβασης σε περίπτωση που κάποιος χρήστης τον έχει ξεχάσει. Εδώ έχουμε δύο μεθόδους όπως φαίνεται παρακάτω στην εικόνα. Ο χρήστης καλείται να συμπληρώσει ένα πεδίο με το e-mail του και στη συνέχεια καλείται η μέθοδος *passwordResetButton_Click(object sender, EventArgs e)*.

```
4 using System.Net.Mail;
5 using System.Net;
6
7 public partial class password_recovery : System.Web.UI.Page
8 {
9     protected void Page_Load(object sender, EventArgs e)
10    {
11        ...
12    }
13
14    protected void passwordResetButton_Click(object sender, EventArgs e) ...
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55    private void SendPasswordChangeEmail() ...
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94 }
```

Εικόνα 35: Οι μέθοδοι της τρίτης σελίδας με τις απαραίτητες βιβλιοθήκες

Αρχικά, δημιουργείται μία σύνδεση με τη βάση δεδομένων και μέσω αυτής ελέγχεται αν το e-mail που εισήγαγε ο χρήστης υπάρχει στη βάση δεδομένων ώστε να διαπιστωθεί αν ο χρήστης είναι εγγεγραμμένος. Αν δεν βρεθεί, η διαδικασία σταματά και εμφανίζεται αντίστοιχο προειδοποιητικό μήνυμα. Αν ο χρήστης είναι εγγεγραμμένος, πραγματοποιείται ο επόμενος έλεγχος ο οποίος πιστοποιεί αν ο χρήστης έχει ενεργοποιήσει τον λογαριασμό του ή όχι. Εφόσον ο λογαριασμός είναι ενεργοποιημένος, καλείται η συνάρτηση *SendPasswordChangeEmail()* και εμφανίζεται έναν μήνυμα επιτυχίας της διαδικασίας στον χρήστη. Αντιθέτως, αν δεν είναι ενεργοποιημένος εμφανίζεται και εδώ το κατάλληλο μήνυμα.

Σχετικά με τη μέθοδο *SendPasswordChangeEmail()*, η λειτουργία της είναι η ίδια με τη *SendActivationEmail()*. Και σε αυτή την περίπτωση δημιουργούμε μια σύνδεση με τη βάση, κατασκευάζουμε ένα νέο μοναδικό ID για τον κωδικό και στέλνουμε ένα email στον χρήστη με έναν σύνδεσμο. Ο σύνδεσμος αυτός τον κατευθύνει σε μια διαφορετική σελίδα έχοντας ως παράμετρο στο URL και εδώ αυτό το μοναδικό ID. Και σε αυτή την περίπτωση ο πίνακας *PasswordRec* ανανεώνεται με μία καινούρια εγγραφή η οποία περιλαμβάνει το μοναδικό ID του χρήστη ο οποίος έκανε την αίτηση αλλαγής και του ID που δημιουργήθηκε για την συγκεκριμένη αλλαγή

3.10.4 Σελίδα δημιουργίας νέου κωδικού

Εδώ πραγματοποιείται το τελικό βήμα για να αλλάξει κωδικό πρόσβασης ένας χρήστης. Απαιτείται η εισαγωγή πληροφοριών σε τρία πεδία. Του e-mail, του κωδικού πρόσβασης και μία επανάληψη του κωδικού πρόσβασης. Η μέθοδος που καλείται σε αυτή τη σελίδα είναι μία όπως φαίνεται στην παρακάτω εικόνα.

```
1  using System;
2  using System.Configuration;
3  using System.Data.SqlClient;
4
5
6  public partial class password_change : System.Web.UI.Page
7  {
8      protected void Page_Load(object sender, EventArgs e)
9      {
10         ...
11     }
12
13     protected void confirmPassButton_Click(object sender, EventArgs e) ...
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60 }
```

Εικόνα 36: Οι μέθοδοι της τέταρτης σελίδας με τις απαραίτητες βιβλιοθήκες

- **confirmPassButton_Click():** αποθηκεύει το ID αλλαγής κωδικού που υπάρχει στο URL της σελίδας ως παράμετρος και έπειτα ανοίγει μια σύνδεση με τη βάση δεδομένων για να αποκτήσει πρόσβαση στους πίνακες της. Αρχικά ελέγχει αν υπάρχει εγγραφή στη βάση με αυτόν τον κωδικό που αποθήκευσε. Αν δεν υπάρχει, τότε σημαίνει ότι ο κωδικός έχει χρησιμοποιηθεί, οπότε πρέπει να ξαναγίνει αίτηση για αλλαγή κωδικού. Αντιθέτως, αν υπάρχει εγγραφή, αποθηκεύει το ID του χρήστη στον οποίο αντιστοιχεί ο κωδικός αυτός. Έπειτα, αλλάζει τον κωδικό του χρήστη με τον καινούριο που επέλεξε, εμφανίζει το κατάλληλο μήνυμα και διαγράφει από την βάση την αντιστοιχία του μοναδικού κωδικού αλλαγής και του μοναδικού κωδικού χρήστη.

3.10.5 Σελίδα εύρεσης πληροφοριών

Οι σελίδες οι οποίες περιγράφονται από αυτή τη σελίδα και κάτω είναι «παιδιά» της ίδιας MasterPage. Η τελευταία, έχει κάποιες λειτουργίες οι οποίες μοιράζονται μεταξύ των σελίδων αυτών, επομένως θα αναλυθούν πρώτα οι λειτουργίες αυτές και στη συνέχεια θα περιγραφούν οι μέθοδοι της σελίδας αυτής της ενότητας. Οι μέθοδοι που χρησιμοποιούνται φαίνονται στην *Εικόνα 35*. Μέσα στην μέθοδο *Page_Load(object sender, EventArgs e)* τίθεται το κείμενο σε ένα σύνδεσμο ο οποίος οδηγεί στο προφίλ του χρήστη που έχει συνδεθεί ως το όνομα του χρήστη αυτού, το οποίο το βρίσκουμε χρησιμοποιώντας τη συνεδρία που δημιουργήθηκε κατά τη σύνδεση του χρήστη και έχει αποθηκευμένο το όνομα του τελευταίου. Στη δεύτερη μέθοδο, την *LogoutButton_Click(object sender EventArgs e)*, με την οποία γίνεται η αποσύνδεση του χρήστη, διαγράφουμε τη συνεδρία και ανακατευθύνουμε τον χρήστη στην σελίδα σύνδεση.

```
1  using System;
2      using System.Collections.Generic;
3      using System.Configuration;
4      using System.Data.SqlClient;
5      using System.Linq;
6      using System.Web;
7      using System.Web.UI;
8      using System.Web.UI.WebControls;
9
10     public partial class _Default : System.Web.UI.MasterPage
11     {
12         protected void Page_Load(object sender, EventArgs e) ...
16
17         protected void LogoutButton_Click(object sender, EventArgs e) ...
22
23     }
24
```

Εικόνα 37: Οι μέθοδοι του MasterPage μαζί με τις απαραίτητες βιβλιοθήκες

Έχοντας τελειώσει με τις μεθόδους του MasterPage, μπορούμε να συνεχίσουμε με την ανάλυση των μεθόδων της κύριας σελίδας. Για την ομαλή λειτουργία της σελίδας αυτής, εκτός από τον κώδικα που πρέπει να γραφεί στο back-end κομμάτι, απαιτείται και η δημιουργία κάποιων μεθόδων στο front-end κομμάτι, δηλαδή αυτό που αφορά τον client, οι οποίες είναι γραμμένες σε JavaScript και σε jQuery. Τέτοιες μέθοδοι χρησιμοποιούνται κατά κύριο λόγο για να προσθέσουν στο site μεγαλύτερο ποσοστό αλληλεπίδρασης με τον χρήστη και να μην είναι όλα τελείως στατικά και ο χρήστης απλώς να πατάει ένα κουμπί και να περιμένει να δει αποτελέσματα.

- **pageLoad() (jQuery):** αφορά το jQuery datepicker, το οποίο είναι ένα έτοιμο ημερολόγιο που μπορεί να χρησιμοποιηθεί είτε μόνο του είτε να εμφανίζεται πατώντας σε κάποιο textbox και επιτρέπει τον χρήστη να επιλέξει μια ημερομηνία με γραφικό τρόπο. Ο τρόπος χρήσης παρουσιάζεται στο επόμενο κεφάλαιο. Στη μέθοδο αυτή θέτουμε την ελάχιστη ημερομηνία να είναι η σημερινή, προσδιορίζουμε τον τρόπο εμφάνισης της ημερομηνίας ο οποίος

στην εργασία αυτή είναι μέρα/μήνας/έτος και κάθε ένα από αυτά περιγράφεται με δύο αριθμούς. Επίσης μόλις ο χρήστης θέσει μία ημερομηνία στο πρώτο datapicker, τότε η ελάχιστη ημερομηνία του δεύτερου τίθεται μία μέρα μετά.

- **OnTeamSelected(source, EventArgs) (JavaScript):** Η μέθοδος αυτή γεμίζει ένα πεδίο κειμένου με τα ονόματα των ομάδων τις οποίες επιλέγει ο χρήστης ώστε να του εμφανιστούν πληροφορίες μόνο από αυτές. Η ομάδα αυτές παρέχονται από ένα άλλο πεδίο κειμένου το οποίο τις τραβάει από τη βάση δεδομένων. Η μέθοδος αυτή ουσιαστικά ελέγχει ποια ομάδα επιλέγει ο χρήστης και την προσθέτει στο πεδίο μαζί με τις υπόλοιπες.
- **OnTagsSelected(source, EventArgs) (JavaScript):** Η λειτουργία της μεθόδου αυτής είναι ίδια με την προηγούμενη, μόνο που αυτή τη φορά τα πεδία κειμένου που χρησιμοποιούνται αφορούν ετικέτες. Υπάρχει ένα πεδίο το οποίο τραβάει τις ετικέτες που υπάρχουν στη βάση δεδομένων, ο χρήστης επιλέγει κάποια και η μέθοδος αυτή κρατάει το όνομά της και το προσθέτει σε ένα δεύτερο πεδίο.
- **clearTarget(this) (JavaScript):** Μία απλή μέθοδος η οποία εφαρμόζεται στα πεδία που γεμίζει ο χρήστης με ομάδες και ετικέτες. Μόλις ο χρήστης κάνει κλικ στα πεδία αυτά, τότε καλείται αυτή η μέθοδος και καθαρίζει το πεδίο αφαιρώντας τις ομάδες ή τις ετικέτες που είχαν προστεθεί σε αυτό.
- **ShowHideDivOnButtonClick() (jQuery):** Χρησιμοποιείται για να εμφανίζει ή να κρύβει ένα κομμάτι τις σελίδας με το πάτημα ενός κουμπιού. Μόλις πατηθεί το κουμπί ελέγχεται αν το κομμάτι αυτό είναι κρυμμένο ή όχι και αναλόγως συμβαίνει η κατάλληλη λειτουργία.
- **initAutocomplete():** Η σελίδα αυτή χρησιμοποιείται παράλληλα με τον κώδικα της σελίδας. Γενικά ο σκοπός της είναι η εμφάνιση του χάρτη και η δημιουργία λίστας επιλογών από τοποθεσίες κατά την αναζήτηση τοποθεσίας από τον χρήστη. Αυτός ο σκοπός επιτυγχάνεται με πολύ πιο γρήγορο τρόπο όταν η μέθοδος καλείται αμέσως μετά το κομμάτι στο οποίο θα εμφανιστεί ο χάρτης. Δημιουργεί αρχικά έναν χάρτη με κέντρο την Ελλάδα. Στη συνέχεια «ακούει» για αλλαγές που γίνονται στο πεδίο αναζήτησης τοποθεσίας. Κάθε φορά που ο χρήστης επιλέγει μία τοποθεσία, αποθηκεύονται το γεωγραφικό μήκος και πλάτος της τοποθεσίας σε δύο κρυφά πεδία, η κάμερα κεντράρει στο σημείο αυτό και εμφανίζεται εκεί ένα στίγμα μαζί με έναν κύκλο γύρω από αυτό με ακτίνα τόση όση η απόσταση που έδωσε ο χρήστης από το σημείο που επέλεξε. Επίσης μετά από αυτή τη μέθοδο εισάγουμε μία διεύθυνση URL, μέσω της οποίας καλείται η υπηρεσία και στο τέλος αυτής εισάγουμε την παράμετρο key=«το κλειδί που δημιουργήσαμε πριν».

Αυτές ήταν οι μέθοδοι που απαιτούνται για την ομαλή λειτουργία του ιστοτόπου όσον αφορά το front-end κομμάτι. Παρακάτω περιγράφονται οι μέθοδοι οι οποίες έχουν γραφεί για την ίδια σελίδα αλλά στο back-end κομμάτι για την επικοινωνία της σελίδας με τον server. Οι μέθοδοι αυτές φαίνονται στην παρακάτω εικόνα.

```
4 using System.Data;
5 using System.Data.SqlClient;
6 using System.Linq;
7 using System.Web;
8 using System.Web.UI;
9 using System.Web.UI.WebControls;
10 using System.Globalization;
11
12 public partial class index : System.Web.UI.Page
13 {
14     protected void Page_Load(object sender, EventArgs e)
23
24     protected void searchButton_Click(object sender, EventArgs e)
271
272     protected void insertRating(object source, DataListCommandEventArgs e)
488
489     [System.Web.Script.Services.ScriptMethod()]
490     [System.Web.Services.WebMethod]
491     public static List<string> searchUsernames(string prefixText, int count)
509
510     [System.Web.Script.Services.ScriptMethod()]
511     [System.Web.Services.WebMethod]
512     public static List<string> searchTeams(string prefixText, int count)
530
531     [System.Web.Script.Services.ScriptMethod()]
532     [System.Web.Services.WebMethod]
533     public static List<string> searchTags(string prefixText, int count)
550 }
```

Εικόνα 38: Οι μέθοδοι της πέμπτης σελίδας με τις απαραίτητες βιβλιοθήκες

- **Page_Load(object sender, EventArgs e):** Όπως αναφέρθηκε και προηγουμένως, η μέθοδος αυτή καλείται όταν φορτώνεται η σελίδα. Εδώ, οι λειτουργίες που γίνονται είναι δύο. Η πρώτη λειτουργία είναι να ελέγχεται αν υπάρχει συνεδρία, η οποία να έχει αποθηκευμένο το όνομα χρήστη, δηλαδή αν ο χρήστης που την επισκέφθηκε έχει συνδεθεί στο σύστημα. Αν ο έλεγχος αποτύχει, σημαίνει ότι ο χρήστης δεν έχει συνδεθεί και ανακατευθύνεται στην σελίδα σύνδεσης, έχοντας στο URL μία παράμετρο returnUrl ώστε μετά την σύνδεση να επιστρέψει σε αυτή τη σελίδα από την οποία ξεκίνησε. Η δεύτερη λειτουργία που εκτελείται είναι ο ορισμός του τίτλου της σελίδας.
- **searchButton_Click(object sender, EventArgs e):** Η λειτουργία αυτής της μεθόδου είναι ελαφρώς εκτεταμένη και καλύπτει ένα μεγάλο εύρος ελέγχων, οι οποίοι πρέπει να γίνουν ώστε το τελικό αποτέλεσμα να είναι το σωστό. Ο πρώτος έλεγχος που γίνεται είναι αν έχει προσθέσει ο χρήστης απόσταση από την επιλεγμένη πόλη. Αν ο έλεγχος αποτύχει, εμφανίζεται προειδοποιητικό μήνυμα. Διαφορετικά, δημιουργείται μια σύνδεση με τη βάση δεδομένων και

ξεκινάμε με την κατασκευή ενός αιτήματος προς τη βάση για να εμφανίσουμε τα κατάλληλα μηνύματα. Έπειτα γίνεται έλεγχος αν ο χρήστης έχει εισάγει ομάδες από τις οποίες θέλει να πάρει πληροφορίες. Εδώ έχουμε δύο περιπτώσεις, όμως υπάρχει ένας βασικός κορμός του μηνύματος ο οποίος είναι ο ίδιος και στις δύο περιπτώσεις. Αυτός ορίζει ότι, ανεξάρτητα από τις υπόλοιπες επιλογές, τα μηνύματα θα πρέπει να καλύπτουν το εύρος ενός χρονικού διαστήματος που ορίζει ο χρήστης και η απόσταση των μηνυμάτων από μία τοποθεσία να είναι ίση με αυτή που όρισε ο χρήστης.

- 1. Ο χρήστης έχει επιλέξει ομάδες:** Στην περίπτωση αυτή αποθηκεύονται τα ονόματα των ομάδων που εισήγαγε ο χρήστης και χρησιμοποιούνται στο αίτημα. Η χρήση τους, περιορίζει τα μηνύματα κυρίως σε αυτά τα οποία έχουν γραφεί από άτομα τα οποία ανήκουν στις ομάδες αυτές.
- 2. Ο χρήστης δεν έχει επιλέξει ομάδες:** Στην περίπτωση αυτή δεν έχουμε κάποιον περιορισμό και τα μηνύματα μπορούν να προέρχονται από διάφορες ομάδες μέχρι ενός σημείου, όπως φαίνεται παρακάτω.

Τελειώνοντας με τις ομάδες, έχουμε και άλλα κομμάτια του αιτήματος τα οποία είναι κατά μεγάλο βαθμό ίδια και στις δύο αυτές περιπτώσεις. Κάθε φορά που δημοσιεύεται ένα μήνυμα, προσδιορίζεται το εύρος των ατόμων που μπορεί να το διαβάσουν. Το εύρος αυτό μπορεί να είναι δημόσιο, να αφορά όλες τις ομάδες του χρήστη που έκανε τη δημοσίευση ή να αφορά συγκεκριμένες ομάδες αυτού.

Ανεξάρτητα από την επιλογή ή όχι ομάδων, οι πληροφορίες που έχουν οριστεί ως δημόσιες εμφανίζονται σε όλες τις αναζητήσεις. Επίσης, όταν το μήνυμα είναι περιορισμένο σε κάποια ομάδα, τότε για να εμφανιστεί σε μία αναζήτηση, πρέπει αυτός που την έγραψε, να έχει προσθέσει σε μία από τις ομάδες του αυτόν που κάνει την αναζήτηση και αυτός που κάνει την αναζήτηση πρέπει να έχει προσθέσει σε κάποια από τις ομάδες του αυτόν ο οποίος έχει γράψει το μήνυμα. Το γεγονός αυτό εξασφαλίζει ότι οι δύο χρήστες εμπιστεύονται ο ένας τον άλλο και θέλουν να μοιραστούν και να λαμβάνουν πληροφορίες μεταξύ τους.

Στη συνέχεια γίνεται έλεγχος ώστε να διαπιστωθεί αν ο χρήστης έχει επιλέξει ή όχι ετικέτες. Οι ετικέτες είναι ουσιαστικά ένα έξτρα φίλτρο το οποίο εφαρμόζεται στους προηγούμενους περιορισμούς και φιλτράρει τα μηνύματα για να εμφανιστούν μόνο αυτά που περιέχουν τις συγκεκριμένες ετικέτες. Αν ο χρήστης δεν έχει εισάγει κάποιες ετικέτες, τότε η διαδικασία συνεχίζεται. Από την άλλη, αν έχει εισάγει κάποιες, τότε ελέγχονται αυτές, εισάγονται στο αίτημα και ανάλογα με το αν ο χρήστης επέλεξε να εμφανίζονται όλες όσες εισήγαγε ή έστω μία από αυτές, προστίθενται αυτές στο αίτημα με τον κατάλληλο τρόπο.

Αφού ετοιμαστεί το αίτημα, γίνεται αναζήτηση στη βάση και αποθηκεύονται σε έναν πίνακα όλες οι εγγραφές οι οποίες πληρούσαν τα κριτήρια που τέθηκαν. Ο χρήστης ο οποίος προβάλλει μια πληροφορία, μπορεί να την αξιολογήσει ως

χρήσιμη ή αδιάφορη. Έτσι μαζεύονται διάφορες αξιολογήσεις για κάθε μια πληροφορία. Για το λόγο αυτό, μετά τη δημιουργία του παραπάνω πίνακα, γίνεται ένα άλλο αίτημα προς τη βάση για να βρεθούν οι αξιολογήσεις και μετά εισάγονται αυτές στις αντίστοιχες πληροφορίες.

- **inserRating(object source, DataListCommandEventArgs e):** Η μέθοδος αυτή, χρησιμοποιείται στον πίνακα ο οποίος περιέχει τα μηνύματα που εμφανίζονται στον χρήστη κατά την αναζήτηση. Έχει δύο λειτουργίες. Η πρώτη λειτουργία είναι να ανακατευθύνει τον χρήστη στο προφίλ εκείνου του χρήστη ο οποίος έγραψε την συγκεκριμένη πληροφορία, αν πατήσει πάνω στο όνομά του. Η δεύτερη λειτουργία είναι η αξιολόγηση της πληροφορίας ως χρήσιμη ή αδιάφορη. Όταν ο χρήστης επιλέξει μία από τις δύο βαθμολογίες, η μέθοδος ελέγχει αν είναι η πρώτη αξιολόγηση του χρήστη σε αυτή την πληροφορία. Αν είναι η πρώτη, τότε ανάλογα με το αν ο χρήστης βαθμολόγησε την πληροφορία ως θετική ή αρνητική, θα περάσει στη βάση η θετική ή αρνητική αυτή βαθμολογία και θα εμφανιστεί μία εικόνα ανάλογη με τη βαθμολογία αυτή. Διαφορετικά, αν ο χρήστης είχε βαθμολογήσει ξανά αυτή την πληροφορία τότε γίνεται μια διαφορετική διαδικασία. Αρχικά ελέγχεται τι είδους αξιολόγηση είχε αφήσει ο χρήστης την τελευταία φορά και αναλόγως με το τι θέλει να κάνει αυτή τη φορά θα προκύψει και το αντίστοιχο αποτέλεσμα. Για παράδειγμα, αν ο χρήστης είχε αφήσει θετική βαθμολογία και τώρα θέλει να αφήσει αρνητική, θα γίνει η αλλαγή αυτή στη βάση και θα αλλάξει ανάλογα και η εικόνα. Ενώ αν ο χρήστης είχε αφήσει για παράδειγμα αρνητική βαθμολογία και τώρα επιλέξει ξανά την αρνητική, ουσιαστικά θα πάρει πίσω τη βαθμολογία του και θα είναι σαν να μην άφησε βαθμολογία. Είναι σημαντικό όμως να τονιστεί ότι τίποτα δεν διαγράφεται πραγματικά. Ό,τι αξιολογήσεις έχει κάνει κάποιος χρήστης σε πληροφορίες, παραμένουν μέσα στη βάση. Απλώς, αν τις αλλάξει ή τις ανακαλέσει όπως στο παράδειγμα παραπάνω, σημαδεύονται ως διαγραμμένες και δεν λαμβάνονται υπόψιν.

Παρακάτω ακολουθούν οι τρεις τελευταίες μέθοδοι της σελίδας αυτής. Οι μέθοδοι αυτές είναι Υπηρεσίες Ιστού (Web Services). Με απλά λόγια μία υπηρεσία ιστού, είναι μια συνάρτηση, η οποία μπορεί να προσπελαστεί από άλλα προγράμματα μέσω του διαδικτύου (HTTP) (Ramdhanie, 2017). Αυτές οι υπηρεσίες διαφέρουν ελαφρώς γιατί δημιουργούνται και προσπελούνται απευθείας μέσω αυτής της ιστοσελίδας χρησιμοποιώντας το AJAX (Asynchronous JavaScript and XML), το οποίο είναι μια τεχνολογία η οποία επιτρέπει την αποστολή και τη λήψη δεδομένων από τον server ασύγχρονα, χωρίς να επεμβαίνει στην απεικόνιση και τη συμπεριφορά της υπάρχουσας σελίδας (Ajax (programming)). Η χρήση του στον ιστότοπο αυτό είναι να συνδέει τις παρακάτω υπηρεσίες με ένα TextBox από το Toolbox, το οποίο το «γεμίζει» με τα δεδομένα που λαμβάνει από τις υπηρεσίες

- **searchUsernames(string prefixText, int count):** Στη μέθοδο αυτή, όπως φαίνεται και στο όνομά της, γίνεται αναζήτηση για ονόματα χρηστών. Δημιουργείται μία σύνδεση με τη βάση δεδομένων και στη συνέχεια γίνεται ένα ερώτημα σε αυτή ώστε να επιστραφούν ονόματα χρηστών. Όπως αναφέρθηκε πριν, το AJAX προσφέρει ασύγχρονη επικοινωνία με τον server. Για το λόγο αυτό, όσο πληκτρολογεί ο χρήστης κάποια γράμματα για να βρει τον χρήστη

που επιθυμεί, η υπηρεσία δέχεται αυτόματα την είσοδο αυτή και εμφανίζει αποτελέσματα χρηστών ανάλογα με το κείμενο που έχει γραφεί μέσα στο πεδίο κειμένου. Τα αποτελέσματα εξάγονται και επιστρέφονται αυτόματα πίσω στο πεδίο. Η παράμετρος `prefixText` είναι η είσοδος του χρήστη, ενώ το `count` είναι ο ελάχιστος αριθμός γραμμάτων που πρέπει να εισάγει ο χρήστης στο πεδίο εισόδου ώστε να αρχίσει να εμφανίζεται η λίστα των αποτελεσμάτων.

- **`searchTeams(string prefixText, int count)`:** Η μέθοδος αυτή, ακολουθεί την ίδια φιλοσοφία με την προηγούμενη και οι παράμετροι `prefixText` και `count` έχουν την ίδια σημασία που είχαν και πριν. Η διαφορά είναι ότι εδώ αντί για ονόματα χρηστών, η αναζήτηση αφορά την εύρεση ονομάτων από ομάδες. Όσο πληκτρολογεί ο χρήστης γράμματα ψάχνοντας μια ομάδα, πραγματοποιείται ασύγχρονη επικοινωνία με τον server, μέσω της οποίας γίνεται αναζήτηση στη βάση δεδομένων για την εύρεση των αντίστοιχων ονομάτων από ομάδες οι οποίες στη συνέχεια επιστρέφονται πίσω στο πεδίο κειμένου όπου ο χρήστης πληκτρολογεί την είσοδο.
- **`searchTags(string prefixText, int count)`:** Και σε αυτή τη μέθοδο ακολουθείται η ίδια διαδικασία η οποία ακολουθείται και στις δύο προηγούμενες μεθόδους. Στόχος αυτής της μεθόδου όμως είναι η εύρεση ετικετών. Όσο ο χρήστης πληκτρολογεί μέσα στο `TextBox`, περνάει η είσοδος μέσω του AJAX στη μέθοδο και αυτή με τη σειρά της επιστρέφει στο `TextBox` μία λίστα με όλες τις ετικέτες οι οποίες αντιστοιχούν στην είσοδο που έχει γραφεί.

3.10.6 Σελίδα προφίλ του χρήστη

Η σελίδα αυτή είναι η κύρια σελίδα του site και γι' αυτό περιλαμβάνει τις περισσότερες μεθόδους συγκριτικά με τις υπόλοιπες σελίδες. Μερικές από αυτές είναι ίδιες με αυτές που αναλύθηκαν σε άλλες σελίδες και για τον λόγο αυτό δεν θα αναλυθούν στον ίδιο βαθμό ξανά. Στην παρακάτω εικόνα φαίνονται συγκεντρωμένες όλες οι μέθοδοι.

```
protected void Page_Load(object sender, EventArgs e)...

protected void ChangeUserInfo(object sender, EventArgs e)...

protected void saveButton_Click(object sender, EventArgs e)...
protected void cancelButton_Click(object sender, EventArgs e)...

protected void groupsButton(object sender, EventArgs e)...

protected void Upload(object sender, EventArgs e)...

private void pageConditionInitialize()...

private void UserInfoGenerate()...

protected void shareButton_Click(object sender, EventArgs e)...
protected void addTagButton_Click(object sender, EventArgs e)...

protected void statusUpdates()...

protected void dataListItemCommand(object source, DataListCommandEventArgs e)...

private void fetchStatistics()...

protected void fetchTeamStatistics(object sender, EventArgs e)...

private byte[] ConvertImageToByteArray(System.Drawing.Image imageToConvert,
                                        System.Drawing.Imaging.ImageFormat formatOfImage)...

[System.Web.Script.Services.ScriptMethod()]
[System.Web.Services.WebMethod]
public static List<string> searchUsernames(string prefixText, int count)...

[System.Web.Script.Services.ScriptMethod()]
[System.Web.Services.WebMethod]
public static List<string> searchTeams(string prefixText, int count)...

[System.Web.Script.Services.ScriptMethod()]
[System.Web.Services.WebMethod]
public static List<string> searchTags(string prefixText, int count)...
```

Εικόνα 39: Οι μέθοδοι της έκτης σελίδας

Για καλύτερη κατανόηση της λειτουργίας των παρακάτω μεθόδων είναι προτιμότερο να διαβάσει ο αναγνώστης πρώτα το επόμενο κεφάλαιο όπου γίνεται η παρουσίαση της σελίδας και μετά να διαβάσει το παρακάτω τμήμα.

- **Page_Load(object sender, EventArgs e):** Όπως έχει αναφερθεί ξανά η μέθοδος αυτή καλείται όταν φορτώνει η σελίδα. Γίνεται αρχικά ένας έλεγχος αν έχει δημιουργηθεί συνεδρία με αποθηκευμένο το όνομα χρήστη, δηλαδή αν ο χρήστης έχει συνδεθεί στο σύστημα και αν δεν το έχει κάνει ανακατευθύνεται στην αρχική σελίδα για να συνδεθεί. Έπειτα καλούνται μέθοδοι, οι οποίες θα αναλυθούν παρακάτω, και αφορούν την αρχικοποίηση της σελίδας και τη φόρτωση δεδομένων από τη βάση για να προβληθούν στον χρήστη.
- **ChangeUserInfo(object sender, EventArgs e):** Η μέθοδος αυτή καλείται όταν ο χρήστης θέλει να αλλάξει τα στοιχεία του. Αρχικά κρύβει τα προηγούμενα

κουμπιά και εμφανίζει στη θέση τους δύο καινούρια. Έπειτα κρύβει τις πληροφορίες και εμφανίζει στη θέση τους πεδία κειμένου στα οποία τίθεται η ιδιότητα placeholder ίση με τις αντίστοιχες προηγούμενες πληροφορίες. Επίσης καλείται η συνάρτηση `initAutocomplete()`, που αναφέρθηκε στην προηγούμενη σελίδα γιατί διαφορετικά ο χάρτης χάνεται.

- **saveButton_Click(object sender, EventArgs e):** Η μέθοδος αυτή καλείται, όταν ο χρήστης πατήσει το κουμπί της αποθήκευσης που εμφανίστηκε από την παραπάνω μέθοδο. Γενικά η μέθοδος αυτή, αποθηκεύει τα στοιχεία που έγραψε ο χρήστης στα πεδία κειμένου τα οποία εμφανίστηκαν με την προηγούμενη. Ξεκινάει με την δημιουργία μιας σύνδεσης με τη βάση δεδομένων και την κατασκευή ενός αιτήματος για την αποθήκευση. Ένας βασικός κορμός είναι έτοιμος και ο υπόλοιπος κατασκευάζεται σταδιακά μέσα από ελέγχους. Ο κάθε έλεγχος τσεκάρει ένα από τα πεδία και αναλόγως προσθέτει κομμάτια στο αίτημα. Μόλις πραγματοποιηθούν όλοι οι απαραίτητοι έλεγχοι, καλείται το αίτημα και ανανεώνεται η βάση. Τέλος, εκτελούνται οι αντίστροφες εντολές της προηγούμενης μεθόδου για να επανέλθει η σελίδα στην προηγούμενη της κατάσταση και καλείται η συνάρτηση `initAutocomplete()` για την εμφάνιση καθώς και μία ακόμα μέθοδος η οποία θα αναλυθεί παρακάτω και είναι υπεύθυνη για τη φόρτωση των πληροφοριών του χρήστη από τη βάση.
- **cancelButton_Click(object sender, EventArgs e):** Καλείται όταν ο χρήστης πατήσει το δεύτερο κουμπί από αυτά που εμφάνισε η μέθοδος `ChangeUserInfo()`. Στην περίπτωση αυτή ο χρήστης ακυρώνει την αλλαγή των στοιχείων του. Οι λειτουργία αυτής της μεθόδου περιλαμβάνει την αντίστροφη διαδικασία της `ChangeUserInfo()`. Κρύβει τα τρέχοντα κουμπιά και εμφανίζει τα προηγούμενα, κρύβει τα πεδία κειμένου για την εισαγωγή δεδομένων και εμφανίζει τα στοιχεία του χρήστη και τέλος καλεί και αυτή την συνάρτηση `initAutocomplete()` για να μην χαθεί ο χάρτης.
- **groupsButton_Click(object sender, EventArgs e):** Η μέθοδος αυτή, καλείται όταν ο χρήστης πατάει το κουμπί των ομάδων και η λειτουργία της είναι να κατευθύνει τον χρήστη στη σελίδα με τις ομάδες.
- **Upload(object sender, EventArgs e):** Αυτή η μέθοδος, καλείται για το ανέβασμα μιας φωτογραφίας του χρήστη στη σελίδα. Αρχικά ελέγχει αν ο χρήστης έχει επιλέξει κάποια εικόνα στο κατάλληλο πεδίο. Αν δεν το έχει κάνει τότε εμφανίζεται το κατάλληλο μήνυμα στον χρήστη. Διαφορετικά αποθηκεύει την εικόνα και δημιουργεί μια σύνδεση με τη βάση δεδομένων. Στη συνέχεια προσθέτει την εικόνα αυτή στη βάση στον πίνακα των χρηστών αφού πρώτα τη μετατρέψει σε μία ακολουθία από byte. Σε περίπτωση που προκύψει κάποιο σφάλμα κατά τη διαδικασία αυτή, τότε η διαδικασία σταματά και εμφανίζεται στον χρήστη αντίστοιχο μήνυμα.
- **pageConditionInitialize():** Η λειτουργία αυτής της μεθόδου αφορά την αρχικοποίηση της σελίδας για τη σωστή προβολή της στον χρήστη. Αρχικά τίθεται ο τίτλος της σελίδας, ο οποίος παίρνει το όνομα του χρήστη το οποίο

είναι αποθηκευμένο στη συνεδρία. Στη συνέχεια θέτει μία ιδιότητα στο κουμπί επιλογής φωτογραφίας, ώστε όταν ο χρήστης το πατήσει να του εμφανιστεί ένα παράθυρο με τα αρχεία του για να επιλέξει τη φωτογραφία του από εκεί και δημιουργεί μια σύνδεση με τη βάση δεδομένων. Έπειτα, χρησιμοποιεί τη σύνδεση αυτή για να ελέγξει αν ο χρήστης έχει ανεβάσει κάποια φωτογραφία ή όχι. Αν δεν έχει ανεβάσει, τότε η φωτογραφία προφίλ του θα είναι η προκαθορισμένη ενώ αν έχει ανεβάσει τότε θα του εμφανιστεί η δική του.

- **UserInfoGenerate():** Η μέθοδος αυτή φέρνει τα στοιχεία του χρήστη από τη βάση. Συνδέεται με τη βάση δεδομένων και ζητάει από τον πίνακα των χρηστών το όνομα του χρήστη που έχει συνδεθεί στο site, το επώνυμο του, τον τόπο διαμονής του και την εκπαίδευσή του. Τα πεδία, όνομα και επώνυμο, είναι υποχρεωτικά κατά την εγγραφή, επομένως θα φέρει σίγουρα αποτέλεσμα από τη βάση. Τα πεδία της διαμονής και της εκπαίδευσης όμως δεν είναι υποχρεωτικά και ίσως ο χρήστης να μην έχει εισάγει κάτι ακόμα. Στην περίπτωση αυτή επιστρέφεται το κείμενο «Δεν έχει οριστεί».
- **shareButton_Click(object sender, EventArgs e):** Καλείται όταν ο χρήστης θέλει να δημοσιεύσει κάποια πληροφορία. Η πρώτη εντολή της είναι, όπως και στις περισσότερες μεθόδους που είδαμε μέχρι τώρα, η εμφάνιση του χάρτη. Έπειτα ακολουθεί μία σειρά ελέγχων για να εξασφαλιστεί ότι πληρούνται ορισμένες προϋποθέσεις. Αρχικά, ελέγχεται αν ο χρήστης έχει επιλέξει κάποια τοποθεσία. Αν όχι τότε του εμφανίζεται προειδοποιητικό μήνυμα. Έπειτα ελέγχεται αν έχει εισάγει το εύρος των ατόμων που θα μπορούν να προβάλλουν την πληροφορία. Αν δεν το έχει κάνει τότε ενημερώνεται αντίστοιχα. Μετά ελέγχεται αν έχουν εισαχθεί με τον κατάλληλο τρόπο οι ομάδες και οι ετικέτες και αν δεν έχουν εισαχθεί σωστά τότε ξανά εμφανίζεται προειδοποιητικό μήνυμα. Εφόσον έχουν ελεγχθεί όλοι οι παραπάνω περιορισμοί η διαδικασία προχωράει στο κυρίως κομμάτι. Όπως και στις υπόλοιπες μεθόδους, η πρώτη δουλειά είναι η σύνδεση με τη βάση δεδομένων. Στη συνέχεια ελέγχει αν ο χρήστης έχει επιλέξει ετικέτες για τη δημοσίευση. Αν έχει επιλέξει τότε υπάρχουν δύο περιπτώσεις. Η πρώτη περίπτωση είναι οι ετικέτες να έχουν γραφεί για πρώτη φορά, επομένως απλώς εισάγονται στη βάση, ενώ η δεύτερη περίπτωση είναι να υπάρχουν ήδη οι ετικέτες, επομένως εδώ αυξάνεται ο αριθμός των φορών που έχουν χρησιμοποιηθεί. Στη συνέχεια δημιουργείται το κύριο αίτημα, το οποίο αφορά την εισαγωγή της δημοσίευσης στη βάση. Όλες οι απαραίτητες πληροφορίες συγκεντρώνονται και εισάγονται στο αίτημα αυτό. Οι πληροφορίες είναι: id, όνομα χρήστη, περιεχόμενο, ημερομηνία έναρξης και λήξης, τοποθεσία, γεωγραφικό μήκος και πλάτος, ετικέτες, η εικόνα του χρήστη που δημοσίευσε την πληροφορία, οι θετικές και αδιάφορες βαθμολογίες οι οποίες εδώ είναι ίσες με 0 και το εύρος των ατόμων που θα βλέπουν τη δημοσίευση. Μετά την εκτέλεση του αιτήματος αδειάζουν όλα τα πεδία από τις πληροφορίες που εισήχθησαν και καλείται η μέθοδος *statusUpdates()* για να ανανεωθούν οι δημοσιεύσεις του χρήστη με αυτή που μόλις έκανε.

- **addTagButton_Click(object sender, EventArgs e):** Αυτή η μέθοδος καλείται όταν ο χρήστης εισάγει μια καινούρια ετικέτα και πατάει το κουμπί εισαγωγής. Όλες οι ετικέτες μαζεύονται σε ένα πεδίο και κάθε φορά που ο χρήστης γράφει μια καινούρια ετικέτα και πατάει το κουμπί εισαγωγής, εισάγεται το όνομα αυτής στο πεδίο. Και σε αυτή τη μέθοδο, καλείται η μέθοδος `initAutocomplete()` για να εμφανιστεί ο χάρτης.
- **statusUpdates():** Η λειτουργία αυτής της μεθόδου, είναι να εμφανίζει τις δημοσιεύσεις που έχει κάνει ο χρήστης. Συνδέεται με τη βάση και δημιουργεί ένα αίτημα για να φέρει τις δημοσιεύσεις ενός συνδεδεμένου χρήστη χρησιμοποιώντας το όνομα χρήστη το οποίο είναι αποθηκευμένο στη συνεδρία που δημιουργήθηκε κατά την εγγραφή του. Αν δεν υπάρχει καμία δημοσίευση τότε εμφανίζεται ένα ενημερωτικό μήνυμα. Διαφορετικά εμφανίζονται οι δημοσιεύσεις ξεκινώντας από εκείνες που έγιναν πιο πρόσφατα.
- **dataListItemCommand(object source, DataListCommandEventArgs e):** Αυτή η μέθοδος χρησιμοποιείται για τις δημοσιεύσεις. Αναγνωρίζει ποια από τις δημοσιεύσεις έχει επιλέξει ο χρήστης και έχει 4 λειτουργίες.
 1. Τροποποίηση: Αν ο χρήστης επιλέξει να τροποποιήσει τη δημοσίευση, τότε αλλάζει η εμφάνισή της ώστε να μπορεί ο χρήστης να αλλάξει τα στοιχεία της.
 2. Ακύρωση: Αν ο χρήστης επιλέξει να ακυρώσει τις αλλαγές, τότε η εμφάνιση της δημοσίευσης επιστρέφει στην αρχική της μορφή.
 3. Διαγραφή: Αν επιλέξει να διαγράψει μία δημοσίευση, τότε δημιουργείται μία σύνδεση με τη βάση και χρησιμοποιείται το `id` της δημοσίευσης για να διαγραφεί από τον πίνακα. Μετά καλείται η μέθοδος `statusUpdates()` για να ανανεωθούν οι δημοσιεύσεις.
 4. Ανανέωση: Αφού ο χρήστης επιλέξει να τροποποιήσει τη δημοσίευση και αλλάξει όσες πληροφορίες θέλει, μπορεί να επιλέξει ανανέωση. Με τη μέθοδο αυτή δημιουργείται μία σύνδεση στη βάση, αποθηκεύονται όλα τα στοιχεία που εισήγαγε ο χρήστης και ανανεώνεται ο πίνακας με της δημοσιεύσεις. Στο τέλος καλείται η μέθοδος `statusUpdates()`, για να ανανεωθούν οι καταχωρήσεις.
- **fetchStatistics():** Είναι μία από τις μεθόδους που καλούνται όταν φορτώνει η ιστοσελίδα. Συνδέεται με τη βάση δεδομένων και συγκεκριμένα με τον πίνακα των δημοσιεύσεων. Αρχικά δημιουργεί ένα αίτημα προς τη βάση για να βρει πόσες από τις δημοσιεύσεις του χρήστη έχουν βαθμολογηθεί έστω και μία φορά ως χρήσιμες και εμφανίζει τον αριθμό. Στη συνέχεια δημιουργεί ένα δεύτερο αίτημα με το οποίο βρίσκει τον αριθμό των θετικών βαθμολογιών του χρήστη και τους εμφανίζει.
- **fetchTeamStatistics():** Η λειτουργία της μεθόδου αυτής είναι παρόμοια με την προηγούμενη, με τη διαφορά ότι εδώ τα στατιστικά που εμφανίζονται αφορούν ομάδες. Η μέθοδος συνδέεται με τη βάση δεδομένων και παίρνει τα ονόματα των ομάδων που εισήγαγε ο χρήστης για να βρει πόσες από τις δημοσιεύσεις

των μελών τους έχουν βαθμολογηθεί ως χρήσιμες και πόσες είναι οι συνολικές θετικές βαθμολογίες που έχουν λάβει.

- **ConvertImageToByteArray(System.Drawing.Image imageToConvert, System.Drawing.Imaging.ImageFormat formatOfImage):**
Χρησιμοποιείται στην μεθοδο Upload(), για να μετατρέψει μία εικόνα σε μία ακολουθία από byte. Οι παράμετροι που δέχεται, είναι μια εικόνα και ο τύπος αυτής. Αποθηκεύει την εικόνα με τον τύπο της και στη συνέχεια τα μετατρέπει σε μία ακολουθία από byte και επιστρέφει αυτή την ακολουθία.

Οι τελευταίες τρεις μέθοδοι, είναι οι ίδιες με αυτές που είχαν αναλυθεί στην προηγούμενη σελίδα. Προς υπενθύμιση τους, είναι Υπηρεσίες Ιστού οι οποίες δημιουργούνται και καλούνται εσωτερικά στον ιστότοπο. Οι μέθοδοι αυτές είναι τρεις:

1. searchUsernames(string prefixText, int count)
2. searchTeams(string prefixText, int count)
3. searchTags(string prefixText, int count)

Η πρώτη βρίσκει ονόματα χρηστών όσο ο χρήστης πληκτρολογεί σε ένα πεδίο κειμένου και επιστρέφει μία λίστα με τα ονόματα αυτά στο πεδίο. Η δεύτερη και η τρίτη μέθοδος λειτουργούν με τον ίδιο τρόπο όμως αυτές ψάχνουν για ονόματα ομάδων και ετικετών αντίστοιχα.

3.10.7 Σελίδα ομάδων

Η σελίδα αυτή είναι η σελίδα διαχείρισης ομάδων. Ο χρήστης μπορεί να προβάλει μέλη ομάδων, να τα διαγράψει, να προσθέσει άλλα και να δημιουργήσει νέες ομάδες.

```
protected void Page_Load(object sender, EventArgs e)...  
  
protected void createTeamButton_Click(object sender, EventArgs e)...  
  
protected void checkIfUserHasCreatedTeams()...  
  
protected void fetchTeams()...  
  
protected void fetchTeamMembers()...  
  
protected void searchForMember(object sender, EventArgs e)...  
  
protected void onTeamClick(object sender, DataListCommandEventArgs e)...  
  
protected void AddUserInTeam(object sender, EventArgs e)...  
  
[System.Web.Script.Services.ScriptMethod()]  
[System.Web.Services.WebMethod]  
public static List<string> searchUsernames(string prefixText, int count)...  
  
[System.Web.Script.Services.ScriptMethod()]  
[System.Web.Services.WebMethod]  
public static List<string> searchTeamMembers(string prefixText, int count)...  
  
[System.Web.Script.Services.ScriptMethod()]  
[System.Web.Services.WebMethod]  
public static List<string> searchTeams(string prefixText, int count)...
```

Εικόνα 40: Οι μέθοδοι της έβδομης σελίδας

- **Page_Load(object sender, EventArgs e):** Η κλασική μέθοδος η οποία καλείται με τη φόρτωση της σελίδας. Ελέγχει εδώ αν υπάρχει αποθηκευμένη συνεδρία με το όνομα χρήστη, διαφορετικά ανακατευθύνει τον χρήστη στην αρχική για να συνδεθεί. Έπειτα θέτει τον τίτλο της σελίδας σε «Ομάδες» και καλεί μία μέθοδο (αναλύεται παρακάτω) η οποία ελέγχει αν ο χρήστης έχει δημιουργήσει ομάδες.
- **createTeamButton_Click(object sender, EventArgs e):** Είναι μια μέθοδος για τη δημιουργία νέας ομάδας. Δημιουργεί μία σύνδεση με τη βάση δεδομένων για να αντλεί και να εισάγει δεδομένα και αρχικά ελέγχει αν το όνομα της ομάδας που έβαλε ο χρήστης υπάρχει ήδη. Αν υπάρχει εμφανίζεται προειδοποιητικό μήνυμα. Διαφορετικά μαζεύει τα απαραίτητα στοιχεία τα οποία είναι το id της ομάδας, το όνομά της, το όνομα του χρήστη και του δημιουργού που εδώ είναι το ίδιο καθώς και την εικόνα του χρήστη και εκτελεί ένα αίτημα δημιουργίας νέας εγγραφής στον πίνακα Teams.
- **fetchTeams():** Πρόκειται για μια μέθοδο, η οποία φέρνει τα ονόματα των ομάδων του χρήστη από τη βάση δεδομένων. Ξεκινάει με μια σύνδεση με τη βάση, φέρνει όλα τα ονόματα των ομάδων τις οποίες έχει δημιουργήσει ο χρήστης και τα αποθηκεύει σε έναν πίνακα. Αν ο πίνακας αυτός είναι κενός, δηλαδή δεν υπάρχουν ονόματα, τότε απλώς εμφανίζει το κατάλληλο μήνυμα.

- **fetchTeamMembers():** Όπως δηλώνει και το όνομα της, χρησιμοποιείται για να φέρει τα μέλη μιας ομάδας από τη βάση δεδομένων. Συνδέεται με τη βάση και δημιουργεί ένα αίτημα με το οποίο ζητάει το όνομα και την εικόνα των μελών της ομάδας και τα αποθηκεύει σε έναν πίνακα. Αν ο πίνακας είναι κενός εμφανίζεται το κατάλληλο μήνυμα.
- **searchForMember(object sender, EventArgs e):** καλείται όταν ο χρήστης θέλει να βρει κάποιος μέλος μιας ομάδας. Συνδέεται με τη βάση και ψάχνει στον πίνακα των ομάδων να βρει αν υπάρχει μέλος με το όνομα που έδωσε ο χρήστης και αν υπάρχει το εμφανίζει.
- **onTeamClick(object sender, DataListCommandEventArgs e):** Συνδέεται με τον πίνακα που έχει τα ονόματα των ομάδων καθώς και με τον πίνακα που έχει τα μέλη αυτών. Έχει τέσσερις λειτουργίες οι οποίες εκτελούνται ανάλογα με τι επέλεξε ο χρήστης.
 1. Εμφάνιση μελών: Αν ο χρήστης πατήσει το όνομα κάποιας ομάδας τότε δημιουργείται μία σύνδεση με τη βάση και ένα αίτημα προς αυτή ώστε να φέρει τα ονόματα των μελών και τις φωτογραφίες τους τα οποία αποθηκεύει σε έναν πίνακα και τα εμφανίζει.
 2. Διαγραφή ομάδας: Αν ο χρήστης επιλέξει να διαγράψει μια ομάδα τότε συνδέεται η μέθοδος με τη βάση και δημιουργεί ένα αίτημα διαγραφής ώστε να διαγράψει από τον πίνακα την ομάδα με το αντίστοιχο όνομα.
 3. Προφίλ χρήστη: Αν ο χρήστης πατήσει στο όνομα ενός χρήστη ανακατευθύνεται στο προφίλ του.
 4. Διαγραφή μέλους: Αν ο χρήστης πατήσει το κουμπί διαγραφής ενός χρήστη τότε συνδέεται και πάλι η μέθοδος με τη βάση και διαγράφει το συγκεκριμένο μέλος από την ομάδα.
- **AddUserInTeam(object sender, EventArgs e):** Καλείται για να προσθέσει έναν χρήστη ως μέλος σε μια ομάδα. Όπως και στις προηγούμενες μεθόδους, το πρώτο βήμα είναι η δημιουργία σύνδεσης με τη βάση δεδομένων. Έπειτα ελέγχει αν ο χρήστης υπάρχει ήδη στην ομάδα. Αν υπάρχει εμφανίζει μήνυμα σφάλματος. Διαφορετικά, συγκεντρώνει όλα τα απαραίτητα στοιχεία (id, όνομα ομάδας, όνομα χρήστη και δημιουργού και την εικόνα του χρήστη) και δημιουργεί ένα αίτημα με όλα αυτά.

Οι τελευταίες τρεις μέθοδοι είναι οι ίδιες με τις προηγούμενες σελίδες γι' αυτό δεν θα αναλυθούν ξανά εδώ αφού αυτό έχει ήδη γίνει πριν. Παρακάτω φαίνονται τα ονόματα των μεθόδων για απλή αναφορά.

- searchUsernames(string prefixText, int count)
- searchTeamMembers (string prefixText, int count)
- searchTeams(string prefixText, int count)

3.10.8 Σελίδα προφίλ άλλων χρηστών

Η τελευταία σελίδα αυτού του έργου. Αφορά τα προφίλ των χρηστών όταν τα επισκέπτεται κάποιος χρήστης. Οι μέθοδοι της σελίδας φαίνονται παρακάτω.

```
protected void Page_Load(object sender, EventArgs e)...

private void initiatePage()...

private void UserInfoGenerate()...

private void fetchPosts()...

private void fetchStatistics()...

protected void fetchTeamStatistics(object sender, EventArgs e)...

protected void AddUserInTeam(object sender, EventArgs e)...

[System.Web.Script.Services.ScriptMethod()]
[System.Web.Services.WebMethod]
public static List<string> searchUsernames(string prefixText, int count)...

[System.Web.Script.Services.ScriptMethod()]
[System.Web.Services.WebMethod]
public static List<string> searchTeams(string prefixText, int count)...
```

Εικόνα 41: Οι μέθοδοι της όγδοης σελίδας

- **Page_Load(object sender, EventArgs e):** Στη σελίδα αυτή, μετά τον έλεγχο για τη σύνδεση του χρήστη, καλούνται τέσσερις μέθοδοι και αφορούν την αρχικοποίηση της σελίδας, τη φόρτωση πληροφοριών του προφίλ, τη φόρτωση των δημοσιεύσεων του χρήστη στον οποίο ανήκει το προφίλ καθώς και τα στατιστικά του.
- **initiatePage():** Είναι η μέθοδος που καλείται για την αρχικοποίηση της σελίδας. Θέτει τον τίτλο της ως το όνομα του χρήστη στον οποίο ανήκει το προφίλ και συνδέεται με τη βάση για να βρει την φωτογραφία του χρήστη. Αν ο χρήστης έχει ανεβάσει φωτογραφία φέρνει αυτή, διαφορετικά προσθέτει την προκαθορισμένη.
- **fetchPosts():** Η μέθοδος αυτή χρησιμοποιείται για να φέρει από τη βάση και να εμφανίσει τις δημοσιεύσεις του χρήστη. Δημιουργεί μία σύνδεση με τη βάση δεδομένων και ελέγχει αν ο χρήστης έχει κάνει κάποιες δημοσιεύσεις. Αν δεν έχει κάνει εμφανίζεται το κατάλληλο μήνυμα. Διαφορετικά δημιουργεί ένα αίτημα και φέρνει από τη βάση όλα τα απαραίτητα στοιχεία για την πληροφορία.

Όλες οι υπόλοιπες μέθοδοι οι οποίες καλούνται στην σελίδα αυτή είναι ίδιες με αυτές τις οποίες περιγράφηκαν στην προηγούμενη σελίδα και για τον λόγο αυτό δεν θα αναλυθούν και στην ενότητα αυτή.

4. Παρουσίαση

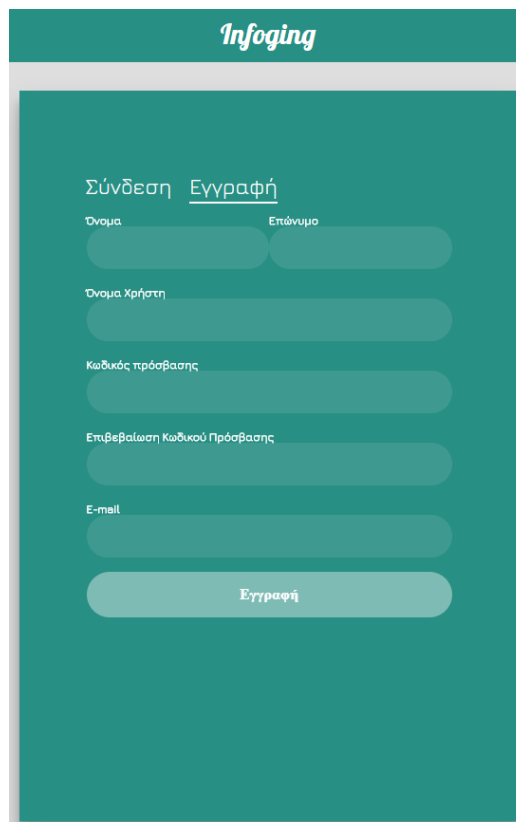
Στην ενότητα αυτή γίνεται η παρουσίαση των σελίδων του ιστοτόπου η οποία συνοδεύεται με εικόνες από κάθε μία από τις σελίδες ώστε να είναι πιο ξεκάθαρος στον αναγνώστη ο τρόπος λειτουργίας τους.

4.1. Εγγραφή, ενεργοποίηση λογαριασμού και αλλαγή συνθηματικού

4.1.1 Εγγραφή

Η σελίδα αυτή είναι η πρώτη σελίδα που συναντά ο χρήστης όταν επισκεφθεί τον ιστότοπο. Μέσω της σελίδας αυτής μπορεί είτε να συνδεθεί εάν έχει ήδη δημιουργήσει λογαριασμό, είτε να κάνει την εγγραφή του.

Το παράθυρο που φαίνεται στην εικόνα δεξιά είναι αυτό της εγγραφής. Αποτελεί το πρώτο βήμα που πρέπει να κάνει ο χρήστης ώστε να αποκτήσει πρόσβαση στον ιστότοπο. Για να κάνει την εγγραφή του ο χρήστης πρέπει να συμπληρώσει τα στοιχεία που απαιτούνται. Όλα είναι υποχρεωτικά επομένως θα πρέπει να δώσει τις αντίστοιχες πληροφορίες σε καθένα από αυτά και να μην αφήσει τίποτα κενό. Σε περίπτωση που το κάνει, θα του εμφανιστεί προειδοποιητικό μήνυμα κάτω από το κουμπί εγγραφή το οποίο θα τον ενημερώνει κατάλληλα. Σημαντικό είναι επίσης ότι επειδή το όνομα χρήστη και το e-mail είναι μοναδικά για κάθε χρήστη, θα πρέπει αυτός να εισάγει το οποίο δεν έχει χρησιμοποιηθεί ξανά. Αν επιλέξει κάποιο όνομα χρήστη που έχει χρησιμοποιηθεί ξανά θα ενημερωθεί μόλις πατήσει το κουμπί «Εγγραφή». Αν εισαχθούν όλα τα στοιχεία σωστά τότε με το πάτημα του κουμπιού Εγγραφή ο server αποθηκεύει τα στοιχεία αυτά, εγγράφει τον χρήστη στο σύστημα και εμφανίζεται στον χρήστη μήνυμα επιτυχίας και κατευθύνεται ο χρήστης στην καρτέλα της σύνδεσης.

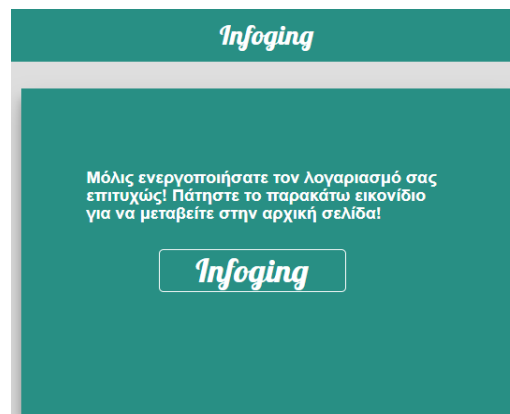


Εικόνα 42: Παρουσία της καρτέλας εγγραφής χρήστη

4.1.2 Ενεργοποίηση λογαριασμού

Η διαδικασία της εγγραφής όμως δεν τελειώνει εδώ. Αν ο χρήστης προσπαθήσει να συνδεθεί θα του εμφανιστεί μήνυμα ότι δεν έχει ενεργοποιήσει ακόμη τον λογαριασμό του. Με την ολοκλήρωση της εγγραφής αποστέλλεται στο e-mail που χρησιμοποιήθηκε μήνυμα ενεργοποίησης. Στο μήνυμα αυτό υπάρχει ένας σύνδεσμος τον οποίο πρέπει να ακολουθήσει ο χρήστης για να ενεργοποιήσει τον λογαριασμό του.

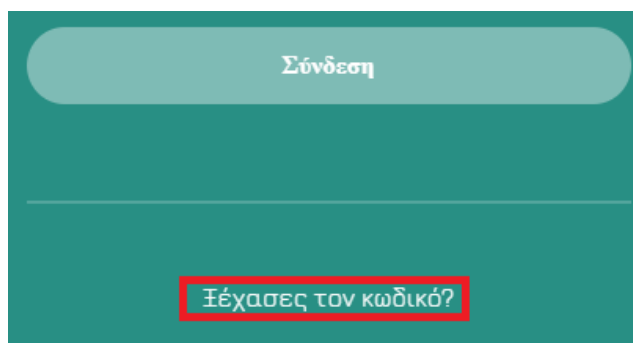
Ο σύνδεσμος αυτός του e-mail οδηγεί τον χρήστη στη σελίδα που φαίνεται επάνω. Μόλις ανοίξει αυτή η σελίδα, ενεργοποιείται ο λογαριασμός του χρήστη και αν πατήσει στο εικονίδιο που φαίνεται κάτω από το κείμενο κατευθύνεται στην αρχική σελίδα έτοιμος να πραγματοποιήσει την σύνδεση του και να ξεκινήσει να χρησιμοποιεί το site.



Εικόνα 43: Παρουσίαση σελίδας ενεργοποίησης λογαριασμού

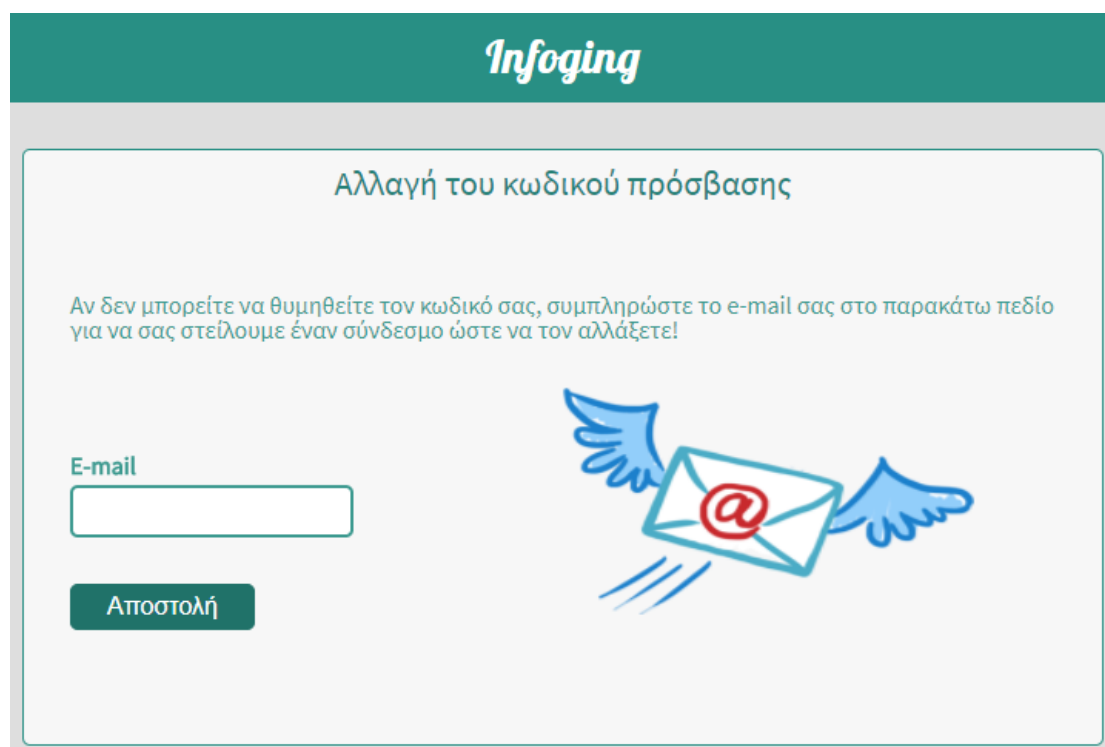
4.1.3 Αλλαγή συνθηματικού

Πριν τη σύνδεση όμως, παρέχεται και μία ακόμα δυνατότητα στην αρχική σελίδα. Αν ο χρήστης έχει ξεχάσει τον κωδικό του μπορεί να ξεκινήσει την διαδικασία αλλαγής του πατώντας στο κουμπί «Ξέχασες τον κωδικό?» κάτω από το κουμπί «Σύνδεση» όπως φαίνεται στην εικόνα δεξιά.



Εικόνα 44: Μετάβαση στην σελίδα αλλαγής συνθηματικού

Πατώντας σε αυτό το κουμπί ο χρήστης μεταφέρεται στη σελίδα που φαίνεται κάτω. Στη σελίδα αυτή του ζητείται να εισάγει το e-mail του προκειμένου να του σταλεί ένα mail το οποίο θα περιέχει έναν σύνδεσμο με τον οποίο θα μπορέσει να αλλάξει τον κωδικό του.



Infoging

Αλλαγή του κωδικού πρόσβασης

Αν δεν μπορείτε να θυμηθείτε τον κωδικό σας, συμπληρώστε το e-mail σας στο παρακάτω πεδίο για να σας στείλουμε έναν σύνδεσμο ώστε να τον αλλάξετε!

E-mail

Αποστολή

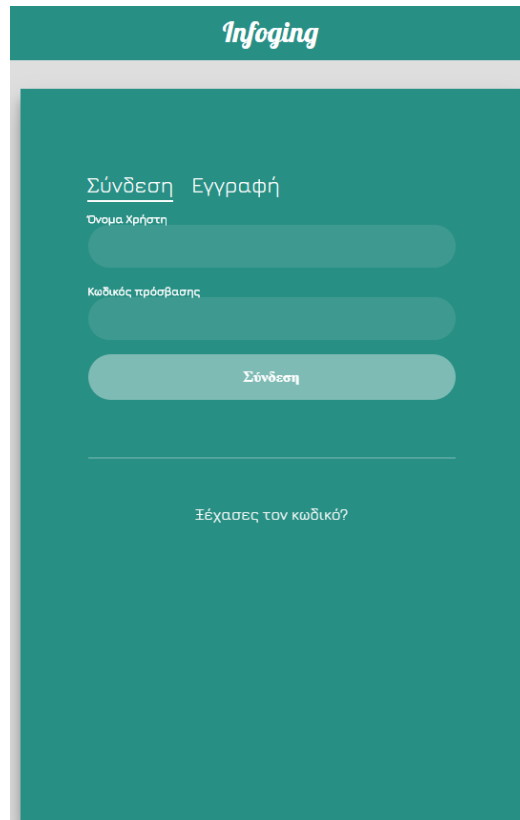
Εικόνα 45: Παρουσίαση σελίδας αποστολής e-mail για αλλαγή συνθηματικού

Στη συνέχεια, αφού ο χρήστης ακολουθήσει τον σύνδεσμο αυτό μεταφέρεται στη σελίδα που φαίνεται στην εικόνα παρακάτω. Για να αποκτήσει έναν καινούριο λογαριασμό ο χρήστης, πρέπει να εισάγει δύο φορές τον καινούριο συνθηματικό της επιλογής του στα αντίστοιχα πεδία και να πατήσει το κουμπί «Αλλαγή Κωδικού». Αφού κάνει τα παραπάνω εμφανίζεται ένα μήνυμα που τον ενημερώνει ότι ο κωδικός του έχει αλλάξει επιτυχώς και μπορεί πλέον να χρησιμοποιήσει τον νέο του κωδικό για να συνδεθεί στο σύστημα.



Εικόνα 46: Παρουσίαση σελίδας αλλαγής συνθηματικού

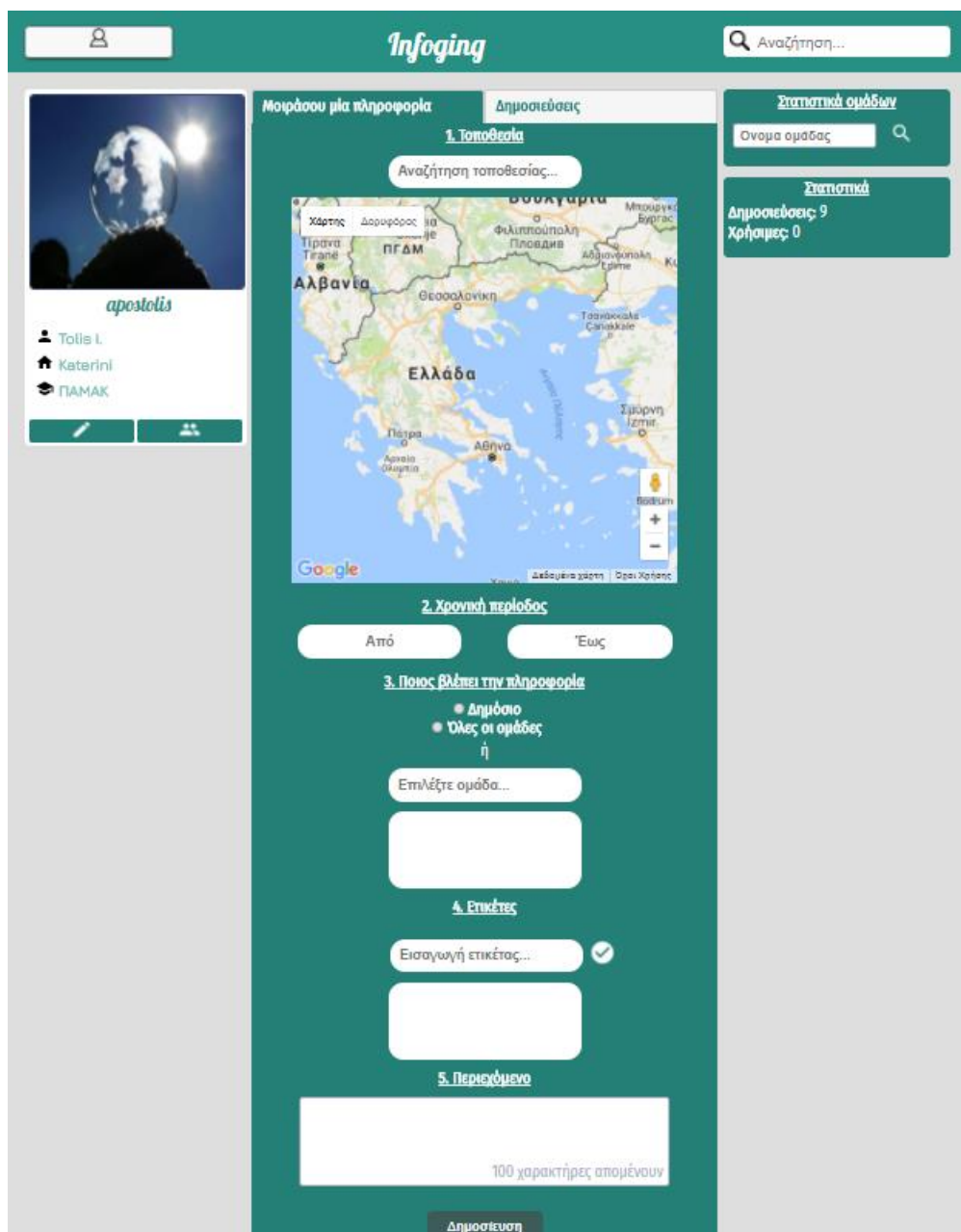
Έχοντας τελειώσει με τα παραπάνω, ο χρήστης είναι έτοιμος για να πραγματοποιήσει την σύνδεση του στο σύστημα. Η καρτέλα της σύνδεσης στην αρχική σελίδα φαίνεται στην εικόνα. Χρησιμοποιώντας το όνομα χρήστη και τον κωδικό του, ο χρήστης εισέρχεται στο σύστημα. Διαφορετικά, σε περίπτωση που είτε το όνομα χρήστη που εισήγαγε είτε ο κωδικός πρόσβασης είναι εσφαλμένα, το πλαίσιο τους γίνεται κόκκινο και εμφανίζεται στο παράθυρο το κατάλληλο μήνυμα για να τον ενημερώσει.



Εικόνα 47: Παρουσίαση καρτέλας σύνδεσης

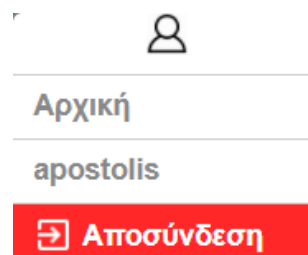
4.2 Προφίλ του χρήστη

Στην παραπάνω εικόνα φαίνεται η σελίδα προφίλ του χρήστη. Εδώ κατευθύνεται ο χρήστης μετά την είσοδο του στο σύστημα. Οι επιλογές του χρήστη μέσω της σελίδας αυτής είναι πολλές. Αρχικά όμως θα αναλυθεί η επικεφαλίδα της σελίδας η οποία, όπως αναφέρθηκε στην προηγούμενη ενότητα, αποτελεί ένα MasterPage το οποίο αποτελεί τον πατέρα αυτής και των επόμενων τριών σελίδων που θα αναλυθούν παρακάτω.



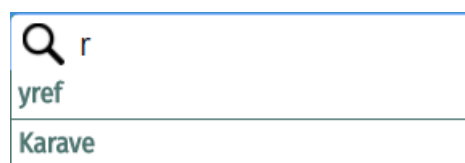
Εικόνα 48: Παρουσίαση σελίδας προφίλ του χρήστη

Στα αριστερά του MasterPage αυτού λοιπόν υπάρχει ένα κουμπί το οποίο όταν το πατήσει εμφανίζει ένα μενού με τρεις επιλογές όπως φαίνεται και στην εικόνα. Η πρώτη επιλογή του χρήστη είναι το κουμπί «Αρχική» το οποίο θα τον μεταφέρει στην αρχική σελίδα της εφαρμογής. Η δεύτερη επιλογή είναι το κουμπί του προφίλ το οποίο έχει το username του χρήστη ο οποίος έχει συνδεθεί στο σύστημα. Η τρίτη και τελευταία του επιλογή είναι το κουμπί «Αποσύνδεση» το οποίο αποσυνδέει τον χρήστη από το σύστημα και τον κατευθύνει στην αρχική του σελίδα.



Εικόνα 49: Επιλογές της κεφαλίδας

Στα δεξιά του MasterPage υπάρχει ένα πεδίο αναζήτησης. Στο πεδίο αυτό μπορεί ο χρήστης να αναζητήσει φίλους, γνωστούς και γενικά οποιοδήποτε άτομο για να εμφανίσει το προφίλ του. Όσο ο χρήστης πληκτρολογεί το όνομα του χρήστη που θέλει να βρει εμφανίζεται κάτω από το πεδίο αυτό μια λίστα με άτομα τα οποία περιέχουν την είσοδο που έγραψε ο χρήστης. Μόλις ο χρήστης βρει το άτομο που ψάχνει μπορεί να το επιλέξει από τη λίστα και να κατευθυνθεί στο προφίλ του.



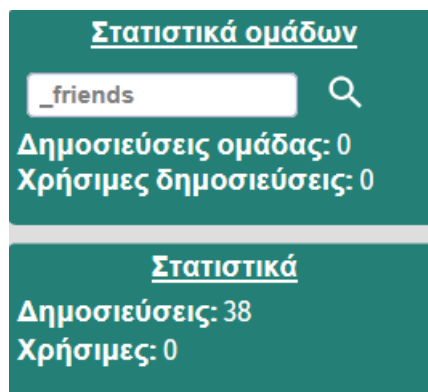
Εικόνα 50: Λίστα σε πεδίο για αυτόματα αποτελέσματα

Για το κυρίως μέρος της σελίδας, ξεκινώντας από τα αριστερά, υπάρχει ένα πεδίο στο οποίο εμφανίζεται η εικόνα του χρήστη, το όνομα χρήστη που χρησιμοποιεί καθώς και τα στοιχεία του. Επίσης όταν ο χρήστης σέρνει το ποντίκι του μέσα στο πεδίο αυτό εμφανίζονται δύο εικονίδια τα οποία του επιτρέπουν να αλλάξει τη φωτογραφία του αν το επιθυμεί. Επίσης υπάρχουν δύο κουμπιά στο κάτω μέρος. Το πρώτο του επιτρέπει να αλλάξει τα στοιχεία του. Όταν το πατάει τα στοιχεία αυτά μετατρέπονται σε πεδία κειμένου μέσα στα οποία ο χρήστης μπορεί να γράψει τα νέα του στοιχεία. Επίσης τα δύο κουμπιά αντικαθίστανται από καινούρια. Ένα κουμπί αποθήκευσης και ένα κουμπί ακύρωσης. Από τα αρχικά κουμπιά, το δεύτερο των μεταφέρει στη σελίδα των ομάδων του. Όλα αυτά φαίνονται στην εικόνα δεξιά.



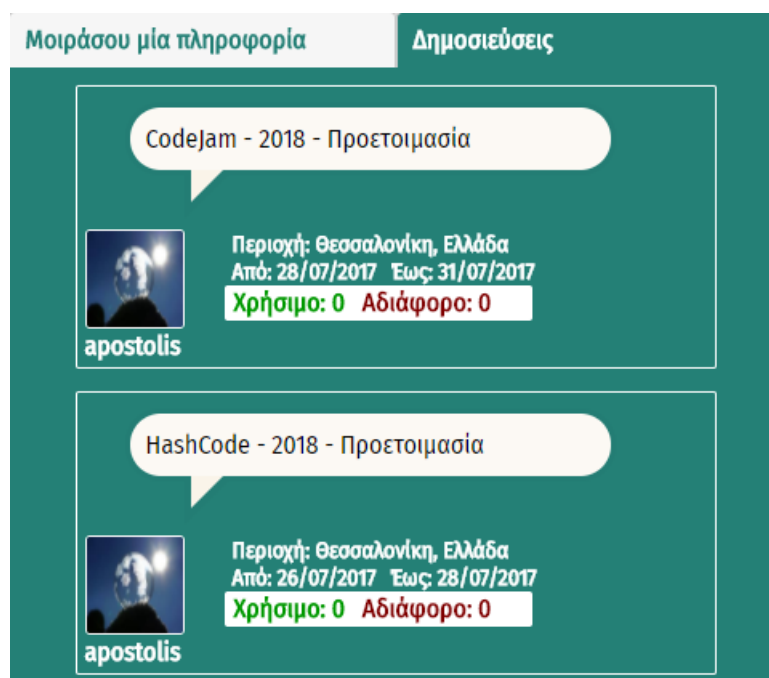
Εικόνα 51: Αλλαγή στοιχείων και εικόνας προφίλ του χρήστη

Συνεχίζοντας, στην δεξιά μεριά της σελίδας, υπάρχουν τα στατιστικά των ομάδων στο πάνω μέρος και τα στατιστικά του χρήστη στο κάτω μέρος. Τα στατιστικά τα οποία εμφανίζονται είναι οι συνολικές δημοσιεύσεις που έχει κάνει ο χρήστης ή η ομάδα καθώς και τον αριθμό αυτών οι οποίες έχουν βαθμολογηθεί ως χρήσιμες. Για να εμφανιστούν τα στατιστικά κάποιες ομάδες, ο χρήστης πρέπει να συμπληρώσει το όνομα μιας ομάδας και να πατήσει το κουμπί της αναζήτησης. Όσο ο χρήστης πληκτρολογεί το όνομα μιας ομάδας εμφανίζεται στο αντίστοιχο πεδίο κειμένου μία λίστα με όλα τα διαθέσιμα ονόματα ομάδων τα οποία περιέχουν την είσοδο του χρήστη. Όταν βρει ο χρήστης αυτή που επιθυμεί μπορεί να την επιλέξει και το όνομά της εμφανίζεται στο πεδίο. Τα παραπάνω φαίνονται στην εικόνα δεξιά.



Εικόνα 52: Στατιστικά του χρήστη και των ομάδων του

Η κύρια λειτουργία αυτής της σελίδας είναι η διαμοίραση μιας πληροφορίας. Αυτό γίνεται στο μεσαίο τμήμα της σελίδας. Υπάρχουν δύο καρτέλες. Στη μία από αυτές περιλαμβάνονται όλα τα απαραίτητα πεδία τα οποία πρέπει να συμπληρώσει ο χρήστης ώστε να μοιραστεί μια πληροφορία. Στην άλλη υπάρχουν όλες οι δημοσιεύσεις οι οποίες έχει κάνει ο χρήστης ώστε να τις προβάλλει και είναι ταξινομημένες κατά φθίνουσα σειρά με βάση τη χρονική στιγμή που δημιουργήθηκαν. Ο χρήστης έχει τη δυνατότητα να διαγράψει ή να αλλάξει αυτές τις πληροφορίες.



Εικόνα 53: Δημοσιεύσεις του χρήστη

Στην εικόνα φαίνεται και ο τρόπος με τον οποίο προβάλλονται οι δημοσιεύσεις. Σε κάθε δημοσίευση υπάρχει η εικόνα του χρήστη, το όνομα του, το περιεχόμενο της, η

τοποθεσία στην οποία λαμβάνει χώρα το θέμα της δημοσίευσης, το χρονικό της διάστημα καθώς και οι χρήσιμες και αδιάφορες βαθμολογίες που έχει λάβει.

Στην εικόνα δεξιά φαίνονται όλα τα πεδία τα οποία μπορούν να συμπληρωθούν, άλλα απαιτούνται άλλα όχι, για να δημοσιεύσει ο χρήστης κάποια πληροφορία. Τα πρώτα δύο στοιχεία τα οποία καλείται να συμπληρώσει ο χρήστης και είναι υποχρεωτικά, είναι η τοποθεσία και το χρονικό διάστημα. Για την προσθήκη τοποθεσίας χρησιμοποιείται ένα πεδίο μέσα στο οποίο ο χρήστης μπορεί να γράψει το όνομα κάποιας τοποθεσίας και ένας χάρτης. Το πεδίο αυτό χρησιμοποιεί μια τεχνολογία της Google και όταν ο χρήστης πληκτρολογεί το όνομα μιας περιοχής τότε ανακτώνται αυτόματα τοποθεσίες από την Google σε μορφή λίστας κάτω από το πεδίο. Μόλις βρεθεί η επιθυμητή τοποθεσία και την επιλέξει ο χρήστης από τη λίστα, αποθηκεύονται το γεωγραφικό της μήκος και πλάτος και εμφανίζεται ένας δείκτης στον χάρτη ο οποίος σημαδεύει στην συγκεκριμένη περιοχή.

Για την χρονική περίοδο, ο χρήστης καλείται να συμπληρώσει δύο πεδία. Την ημερομηνία έναρξης και την ημερομηνία περάτωσης της πληροφορίας. Η επιλογή της ημερομηνίας γίνεται γραφικά. Όταν ο χρήστης πατήσει μέσα σε κάποιο από τα πεδία ημερομηνίας εμφανίζεται το παράθυρο που φαίνεται στην *Εικόνα 53*. Από εκεί ο χρήστης μπορεί να βρει τον μήνα και την ημέρα που επιθυμεί και να την επιλέξει. Μόλις το κάνει η ημερομηνία αυτή εμφανίζεται μέσα στο πεδίο της ημερομηνίας. Επίσης αφού ο χρήστης επιλέξει μια ημερομηνία ως αρχή της χρονικής περιόδου της δημοσίευσης τότε η ημερομηνία τέλους της χρονικής περιόδου τίθεται αυτόματα μία μέρα μετά εκτός και αν ο χρήστης επιλέξει διαφορετικά.

Μοιράσου μια πληροφορία Δημοσιεύσεις

1. Τοποθεσία

Αναζήτηση τοποθεσίας...

2. Χρονική περίοδος

Από Έως

3. Ποιος βλέπει την πληροφορία

• Δημόσιο
• Όλες οι ομάδες
ή
Επιλέξτε ομάδα...

4. Ετικέτες

Εισαγωγή ετικέτας... ✓

5. Περιεχόμενο

100 χαρακτήρες απομένουν

Δημοσίευση

Εικόνα 54: Διαδικασία δημοσίευση πληροφορίας

Το επόμενο στοιχείο το οποίο καλείται να συμπληρώσει ο χρήστης είναι το εύρος της πληροφορίας. Το πεδίο αυτό δεν είναι υποχρεωτικό. Ο χρήστης έχει τρεις επιλογές. Μπορεί να επιλέξει μεταξύ δημόσιας, περιορισμένης στις ομάδες του ή περιορισμένη σε συγκεκριμένες ομάδες τις οποίες ορίζει ο ίδιος. Αν ο χρήστης δεν επιλέξει κάποια από αυτές τις επιλογές τότε το εύρος της πληροφορίας τίθεται αυτόματα σε δημόσιο. Οι δημόσιες πληροφορίες είναι διαθέσιμες σε όλους τους χρήστες του ιστοτόπου ανεξάρτητα από το αν αυτοί ανήκουν σε κάποια από τις ομάδες του χρήστη. Οι πληροφορίες οι οποίες είναι περιορισμένες στις ομάδες του χρήστη είναι διαθέσιμες μόνο στους χρήστες οι οποίοι

| July 2017 | | | | | | |
|-----------|----|----|----|----|----|----|
| SU | MO | TU | WE | TH | FR | SA |
| | | | | | | 1 |
| 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| 23 | 24 | 25 | 26 | 27 | 28 | 29 |
| 30 | 31 | | | | | |

Εικόνα 55: Ημερολόγια για την επιλογή ημερομηνίας για τη δημοσίευση

ανήκουν τουλάχιστον σε μία από τις ομάδες του χρήστη ενώ οι πληροφορίες τις οποίες ο χρήστης περιόρισε μόνο σε κάποιες ομάδες του, είναι διαθέσιμες μόνο στους χρήστες των ομάδων αυτών. Σημαντικό είναι να τονιστεί ότι για να προβάλει κάποιος χρήστης μία πληροφορία, πρέπει αυτός να είναι μέλος σε μία από τις ομάδες του χρήστη που μοιράστηκε την πληροφορία αλλά να έχει και ο ίδιος τον χρήστη αυτό σε μία από τις ομάδες του. Έτσι επιτυγχάνεται το γεγονός ότι, εκτός των δημοσίων πληροφοριών, ο κάθε χρήστης θα προβάλει τις πληροφορίες από τους χρήστες που αυτός θέλει και εμπιστεύεται. Όταν ο χρήστης θέλει να επιλέξει κάποια από τις ομάδες του για να περιορίσει την πληροφορία, γράφει το όνομα της στο πεδίο και αυτόματα εμφανίζεται μια λίστα με ομάδες οι οποίες αντιστοιχούν στην είσοδο που έχει γράψει ο χρήστης. Μόλις ο χρήστης βρει και επιλέξει μια ομάδα, το όνομα αυτής γράφεται στο πεδίο κάτω από το πεδίο αναζήτησης ομάδων. Αν ο χρήστης μετανιώσει για την εισαγωγή του μπορεί να κάνει ένα κλικ μέσα στο πεδίο αυτό και οι ομάδες που έχει γράψει θα διαγραφούν.

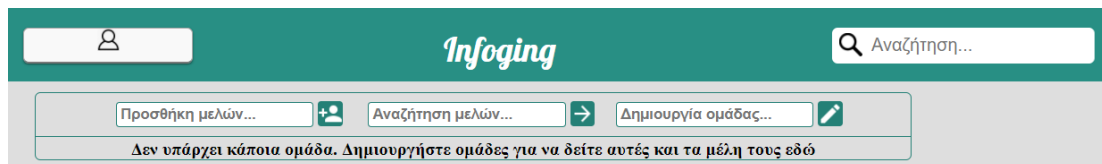
Το επόμενο στοιχείο που μπορεί να συμπεριλάβει στην δημοσίευση ο χρήστης είναι οι ετικέτες. Αυτές μπορεί να υπάρχουν ήδη ή να τις δημιουργήσει ο χρήστης εκείνη τη στιγμή. Όταν ο χρήστης αρχίσει να γράφει το όνομα μιας ετικέτας στο πεδίο εισαγωγής, εμφανίζεται σε αυτό μια λίστα με τα ονόματα ετικετών τα οποία περιέχουν το κομμάτι που εισήγαγε αυτός. Αν ο χρήστης βρει σε αυτή τη λίστα την ετικέτα που θέλει να χρησιμοποιήσει μπορεί να την επιλέξει και αυτή θα εμφανιστεί στο κενό πεδίο κάτω από το πεδίο εισαγωγής. Αν όμως ο χρήστης δεν βρει την ετικέτα και θέλει να δημιουργήσει μία καινούρια, γράφει το όνομα της και πατάει το κουμπί δεξιά του πεδίου εισαγωγής. Όταν το πατήσει η ετικέτα προστίθεται στο πεδίο κάτω από το πεδίο εισαγωγής. Αν ο χρήστης μετανιώσει για την εισαγωγή του μπορεί να κάνει ένα κλικ μέσα στο πεδίο και οι ετικέτες που έχει γράψει θα διαγραφούν.

Τέλος ο χρήστης προσθέτει το περιεχόμενο της δημοσίευσης. Το περιεχόμενο αυτό είναι περιορισμένο σε 100 χαρακτήρες και όσο ο χρήστης πληκτρολογεί την πληροφορία του υπάρχει μέσα στο πλαίσιο ένας αριθμός ο οποίος φθίνει και τον ενημερώνει σχετικά με τον εναπομείνοντα αριθμό χαρακτήρων.

Αφού έχει τελειώσει με την εισαγωγή στοιχείων, πατάει το κουμπί «Δημοσίευση». Ο server επεξεργάζεται το αίτημα με τον τρόπο που περιγράφηκε στο προηγούμενο κεφάλαιο, η δημοσίευση αποθηκεύεται και εμφανίζεται στην καρτέλα των δημοσιεύσεων. Επίσης όλα τα πεδία τα οποία συμπλήρωσε ο χρήστης γίνονται και πάλι κενά και είναι έτοιμα να δεχθούν την επόμενη δημοσίευση του.

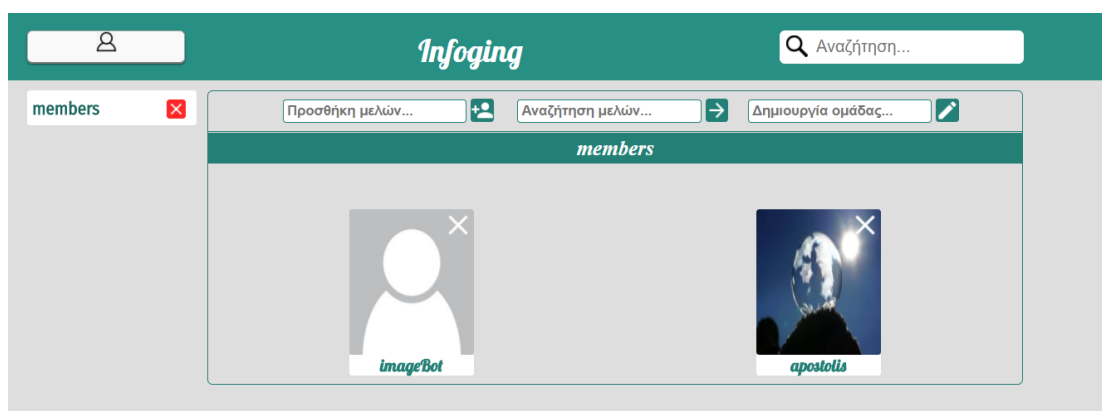
4.3 Ομάδες

Η σελίδα αυτή αφορά τις ομάδες που έχει ο χρήστης. Όπως φαίνεται και στην παρακάτω εικόνα, στη σελίδα αυτή, ο χρήστης μπορεί να δει όλες τις ομάδες του, να δημιουργήσει καινούριες, να προσθέσει μέλη στις ήδη υπάρχουσες και να αναζητήσει μέλη μέσα σε αυτές.



Εικόνα 56: Σελίδα ομάδων πριν τη δημιουργία κάποιας

Στην παραπάνω εικόνα φαίνεται η σελίδα των ομάδων την πρώτη φορά που την επισκέπτεται ο χρήστης και δεν υπάρχει ακόμη κάποια δημιουργημένη ομάδα. Στη συνέχεια όταν ο χρήστης δημιουργήσει μία, τότε η εμφάνιση της σελίδας αυτής αλλάζει και παίρνει τη μορφή που φαίνεται στην *Εικόνα 55*.



Εικόνα 57: Σελίδα ομάδων μετά τη δημιουργία κάποιας

Οι ομάδες που έχει δημιουργήσει ο χρήστης φαίνονται στην αριστερή πλευρά της σελίδας. Αν ο χρήστης πατήσει στο όνομα κάποιας από αυτές τότε στο δεξιό μέρος φορτώνονται τα μέλη της. Επίσης ο χρήστης μπορεί να πατήσει το κόκκινο κουμπί το οποίο υπάρχει στο τέλος του ονόματος κάθε ομάδας για να την διαγράψει.

Στο δεξιό μέρος υπάρχουν τρεις επιλογές, η προσθήκη μελών, η αναζήτηση μελών και η δημιουργία ομάδας. Στο πεδίο «Δημιουργία ομάδας», ο χρήστης μπορεί να πληκτρολογήσει το όνομα της ομάδας που θέλει να δημιουργήσει και να πατήσει το κουμπί δεξιά του πεδίου για να τη δημιουργήσει. Αν το όνομα που έδωσε στην ομάδα υπάρχει ήδη θα του εμφανιστεί προειδοποιητικό μήνυμα. Διαφορετικά η ομάδα δημιουργείται και η στήλη των ομάδων στα αριστερά ανανεώνεται ώστε να περιλαμβάνει και την νέα ομάδα. Με το πεδίο προσθήκη μελών μπορεί ο χρήστης να προσθέσει νέα μέλη στις ομάδες του. Όταν αρχίσει να πληκτρολογεί το όνομα κάποιου χρήστη, όπως συνέβαινε και με τα άλλα πεδία, εμφανίζεται μία λίστα με τα ονόματα των χρηστών τα οποία περιέχουν τη είσοδο του χρήστη κάτω από το πεδίο. Όταν βρει αυτό που θέλει, το επιλέγει και πατάει το κουμπί προσθήκης ώστε να προστεθεί το νέο μέλος στην αντίστοιχη ομάδα. Τέλος, με την αναζήτηση μελών, μπορεί ο χρήστης άμεσα να βρει ένα μέλος της ομάδας του. Αυτό δεν είναι ιδιαίτερα χρήσιμο όταν οι

ομάδες που έχει ο χρήστης είναι μικρές αλλά όταν αυτές αρχίσουν να μεγαλώνουν είναι πολύ χρήσιμο γιατί μπορεί άμεσα να προβάλει το άτομο που θέλει ή να χρησιμοποιήσει το πεδίο για να βρει αν ένα συγκεκριμένο άτομο είναι μέλος της ομάδας ή όχι.

4.4 Τέταρτη σελίδα

Η σελίδα αυτή αφορά τα προφίλ των χρηστών που επισκέπτεται αυτός που έχει συνδεθεί στο σύστημα. Η μορφή της σελίδας είναι αυτή που φαίνεται παρακάτω. Στη σελίδα αυτή προβάλλονται όλα τα στοιχεία του χρήστη όπως το ονοματεπώνυμο του, η εικόνα του κτλ και εμφανίζονται κάποιες από τις δημοσιεύσεις του. Επίσης εδώ μπορεί ο χρήστης να βρει τα στατιστικά αυτού του οποίου τη σελίδα επισκέφθηκε, τα στατιστικά από τις δικές του ομάδες σχετικά με την αξιοπιστία του ατόμου αυτού αλλά και να τον προσθέσει σε κάποια από τις ομάδες του.

The screenshot shows the user profile page in the Infoging system. The header is green and contains the 'Infoging' logo and a search bar. The main content area is divided into several sections:

- Profile Picture:** A globe image with the name 'apostolis' below it.
- User List:** A list of other users: 'Tolis I.', 'Katerini', and 'ΠΑΜΑΚ'.
- Statistics (Στατιστικά):** A green box showing: Δημοσιεύσεις: 40, Χρήσιμες: 0, Θετικές βαθμολογίες: 1.
- Filtering Statistics (Φιλτράρισμα στατιστικών):** A green box with a search box labeled 'Όνομα ομάδας' and a filter icon.
- Add (Προσθήκη):** A green box with a search box labeled 'Όνομα ομάδας' and a plus icon.
- Event Cards:** Two cards for 'CodeJam - 2018 - Προετοιμασία' and 'HashCode - 2018 - Προετοιμασία'. Each card shows a location (Thessaloniki, Greece), dates, and usage statistics (Χρήσιμο: 0, Αδιάφορο: 0).

Εικόνα 58: Παρουσίαση σελίδας προφίλ άλλων χρηστών

Το φιλτράρισμα των στατιστικών αφορά φιλτράρισμα ως προς τις ομάδες. Ουσιαστικά ο χρήστης μπορεί να επιλέξει μία ή και περισσότερες από τις ομάδες του και να φιλτράρει τα στατιστικά ώστε να του εμφανίσει πόσες από τις δημοσιεύσεις του χρήστη στον οποίο ανήκει το προβαλλόμενο προφίλ φάνηκαν χρήσιμες σε άτομα των ομάδων αυτών. Η διαδικασία αυτή αποτελεί ένα κριτήριο αξιοπιστίας για το συγκεκριμένο άτομο. Από τη στιγμή που κάποιος χρήστης έχει κάποια άτομα στις ομάδες του, σημαίνει ότι εμπιστεύεται τα άτομα αυτά επομένως θα εμπιστεύεται και τα άτομα που αυτοί θεωρούν έμπιστα. Έτσι αν αυτοί θεωρούν πολλές από τις δημοσιεύσεις του χρήστη αυτού χρήσιμες τότε αυτό σημαίνει ότι αυτός είναι έμπιστος άρα οι πληροφορίες που θα μας δίνει θα είναι αξιόπιστες και χρήσιμες.

Μετά από αυτό λοιπόν, σε περίπτωση που δεν ξέραμε τον χρήστη και θέλαμε να ελέγξουμε την αξιοπιστία του, μπορούμε να τον προσθέσουμε σε κάποια από τις ομάδες μας. Αυτό μπορεί να γίνει πολύ εύκολα από την τελευταία στήλη. Στο πεδίο που υπάρχει μπορεί ο χρήστης να γράψει το όνομα της ομάδας στην οποία θέλει να προσθέσει τον χρήστη και να πατήσει το κουμπί της προσθήκης. Και στο πεδίο αυτό εφαρμόζεται η μέθοδος με τη λίστα η οποία προτείνει αυτόματα ομάδες. Σε περίπτωση

που ο χρήστης υπάρχει ήδη στην ομάδα που επέλεξε ο χρήστης τότε του εμφανίζει κατάλληλο μήνυμα. Διαφορετικά ο χρήστης προστίθεται και εμφανίζεται αντίστοιχο μήνυμα.

4.5 Εύρεση πληροφοριών

Εδώ πραγματοποιείται η αναζήτηση για να βρει κάποιος χρήστης πληροφορίες. Υπάρχουν κάποια πεδία, απαραίτητα και μη, τα οποία ο χρήστης πρέπει να συμπληρώσει για να του εμφανιστούν τα αποτελέσματα. Η δομή της σελίδας φαίνεται στην παρακάτω εικόνα.

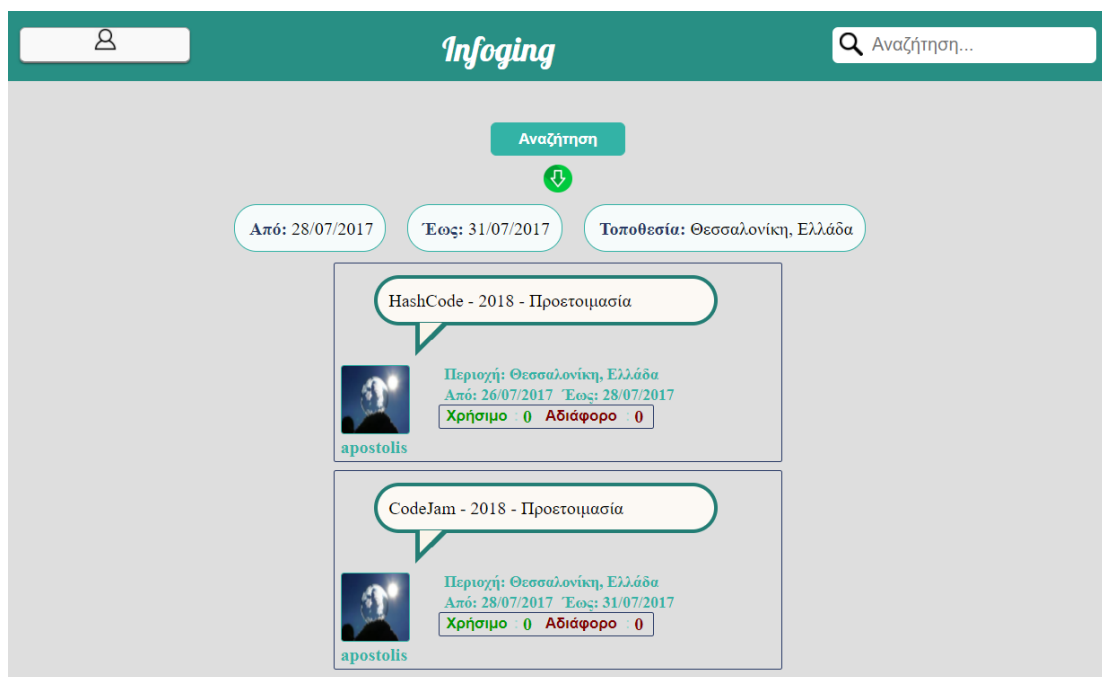
The screenshot shows the 'Infoqing' search interface. It features a green header with the 'Infoqing' logo and a search bar. Below the header, there are three filter sections: 'Ημερομηνία' (Date) with 'Από' and 'Έως' fields; 'Ετικέτες' (Tags) with a search field and radio buttons for 'Έστω ένα' and 'Όλα'; and 'Ομάδες' (Groups) with a search field. To the right is a map of Greece and Albania. Below the map is a 'Τοποθεσία' (Location) section with 'Απόσταση(σε χλμ)..' and 'Περιοχή...' fields. At the bottom center is a green 'Αναζήτηση' (Search) button with a magnifying glass icon.

Εικόνα 59: Παρουσίαση σελίδας εύρεσης πληροφοριών

Όπως φαίνεται παραπάνω, η σελίδα αποτελείται από δύο στήλες. Στα αριστερά έχουμε τρία στοιχεία τα οποία μπορούν να συμπληρωθούν. Πρώτο πεδίο το οποίο είναι και υποχρεωτικό είναι αυτό της ημερομηνίας. Ο χρήστης πατάει μέσα στα πεδία «Από» και «Έως» και εμφανίζεται ένα παράθυρο σαν αυτό που εμφανιζόταν στο προφίλ του από το οποίο μπορεί να επιλέξει την ημερομηνία για την οποία θέλει να βρει τις διαθέσιμες πληροφορίες. Στη συνέχεια μπορεί να επιλέξει ετικέτες για να περιορίσει την πληροφορίες μόνο σε αυτές τις οποίες περιέχουν αυτές τις ετικέτες. Το πεδίο των ετικετών δεν είναι υποχρεωτικό. Ο χρήστης μπορεί να προσθέσει όσες από τις υπάρχουσες ετικέτες θέλει και μπορεί να επιλέξει αν οι πληροφορίες θα πρέπει να περιέχουν όλες όσες επέλεξε ή έστω μία από αυτές. Το τελευταίο πεδίο της αριστερής στήλης είναι αυτό των ομάδων. Ο χρήστης μπορεί να επιλέξει από τις ομάδες του, όσες αυτός θέλει, για να προβάλλει πληροφορίες μόνο από τα άτομα που ανήκουν σε αυτές. Όπως έχει αναφερθεί ξανά οι πληροφορίες που τίθενται ως δημόσιες και καλύπτουν τα χρονικά και τοπικά διαστήματα καθώς και αυτά που αφορούν τις ετικέτες, αν υπάρχουν, θα εμφανίζονται ανεξάρτητα από το αν ο χρήστης έχει επιλέξει συγκεκριμένες ομάδες για να δει πληροφορίες μόνο από αυτές.

Στην δεξιά στήλη της σελίδας έχουμε την επιλογή της τοποθεσίας. Στα πεδία κάτω από τον χάρτη, ο χρήστης μπορεί να επιλέξει μια τοποθεσία και μία απόσταση γύρω από αυτή. Όταν το κάνει εμφανίζεται στο χάρτη ένα σημείο με έναν κύκλο γύρω από αυτό

το οποίο συμβολίζει την απόσταση αυτή. Οι πληροφορίες πρέπει να έχουν ως δηλωμένη τοποθεσία το σημείο ή κάποιο σημείο σε απόσταση ίση με αυτή που έθεσε ο χρήστης γύρω από αυτό. Πατώντας το κουμπί αναζήτησης εμφανίζονται στον χρήστη όλες οι πληροφορίες οι οποίες πληρούν τα κριτήρια αυτά. Ο χρήστης μπορεί να πατήσει το πράσινο βελάκι που φαίνεται κάτω από το κουμπί «Αναζήτηση» για να κρύψει τα παραπάνω πεδία και να προβάλλει μόνο τις δημοσιεύσεις, όπως φαίνεται παρακάτω.



Εικόνα 60: Εμφάνιση σελίδας μετά την αναζήτηση με κρυμμένα τα πεδία αναζήτησης

Αφού ο χρήστης προβάλλει τις δημοσιεύσεις, μπορεί να τις βαθμολογήσει. Υπάρχουν δύο κουμπιά, «Χρήσιμο» και «Αδιάφορο». Πατώντας το αντίστοιχο κουμπί αυξάνεται ο αριθμός των χρήσιμων ή αδιάφορων βαθμολογιών. Στις δημοσιεύσεις τις οποίες έχει αξιολογήσει εμφανίζεται μία εικόνα ανάλογη με τον βαθμό που έχει αφήσει.

5. Συμπεράσματα

Πλέον, στον κόσμο του διαδικτύου τα social media έχουν αποκτήσει έναν πολύ σημαντικό ρόλο. Εκατομμύρια ανθρώπων τα χρησιμοποιούν σε καθημερινή βάση για να καλύψουν διάφορες ανάγκες τους. Η πιο συχνή όμως ανάγκη που έχουν είναι αυτή της πληροφόρησης. Όλοι θέλουν να λαμβάνουν πληροφορίες για διάφορα πράγματα και καταστάσεις. Φυσικά, τα μέσα που τους παρέχουν την πληροφορία αυτή είναι πολλά και το κάνουν αυτό με έναν πολύ εύχρηστο τρόπο. Το πρόβλημα όμως είναι ότι η πληροφορία που παρέχουν είναι απροσδιόριστη και λαμβάνεται αφιλτράριστη.

Σκοπός της εργασίας αυτής ήταν να δώσει μια λύση σε αυτό το ζήτημα. Μέσου του έργου που αναπτύχθηκε, δίνεται η δυνατότητα στους χρήστες να προσδιορίζουν την πληροφορία που πρόκειται να προσφέρουν αλλά και να φιλτράρουν την πληροφορία που πρόκειται να καταναλώσουν. Αυτό επιτρέπει στον καθένα να βρει πιο εύκολα την πληροφορία που τον ενδιαφέρει.

Η διαδικασία για να επιτευχθεί ένα τέτοιο αποτέλεσμα δεν είναι εύκολη. Έχει αρκετές απαιτήσεις και προϋποθέσεις οι οποίες πρέπει να πληρούνται. Καθ' όλη τη διάρκεια όμως αποκτώνται πολλές γνώσεις και είναι μια εμπειρία η οποία μόνο να προσφέρει έχει.

6. Προτάσεις για μελλοντική μελέτη

Τέτοιου είδους έργα δεν έχουν εύκολα τέλος. Χρειάζονται πολύ όρεξη και ιδέες και μπορεί κανείς συνεχώς να τα επεκτείνει. Φυσικά το συγκεκριμένο έργο δεν αποτελεί εξαίρεση. Υπάρχουν πολλές βελτιώσεις τις οποίες μπορεί να κάνει κάποιος σε αυτό για να γίνει πιο ολοκληρωμένο. Αρχικά θα ήταν πολύ ενδιαφέρον να γίνει δυνατή η είσοδος χρηστών μέσω άλλων πηγών όπως για παράδειγμα με κάποιο λογαριασμό στο Facebook, στο Google, στο GitHub κτλ. Επίσης ιδιαίτερα χρήσιμο θα ήταν να γίνουν προτάσεις φίλων. Δηλαδή αφού κάποιος γίνει μέλος, να εμφανίζονται με τον καιρό στο προφίλ του προτάσεις ατόμων με κοινούς φίλους, οι οποίοι θα προσδιορίζονται από τα κοινά άτομα στις ομάδες τους, ή κοινά ενδιαφέροντα τα οποία θα μπορούσαν για παράδειγμα να προσδιορίζονται από τυχόν κοινές ετικέτες που χρησιμοποιεί ο χρήστης με άλλους χρήστες. Ακόμη, τα στατιστικά θα μπορούσαν να διευρυνθούν περισσότερο και να εμφανίζουν περισσότερες πληροφορίες από αυτές που εμφανίζουν τώρα. Επίσης, θα μπορούσε ο κάθε χρήστης να έχει ένα έμβλημα ανάλογα με τη βαθμολογία των δημοσιεύσεων του. Το έμβλημα αυτό θα μπορεί ίσως να φιλτράρεται μαζί με τα στατιστικά των ομάδων όταν επισκέπτεται κάποιος το προφίλ του χρήστη.

Βιβλιογραφία

- (n.d.). Ανάκτηση από W3Schools: <https://www.w3schools.com/about/default.asp>
- Ajax (programming)*. (n.d.). Ανάκτηση από Wikipedia:
[https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming))
- AndyPi*. (n.d.). Ανάκτηση από Accessing a Web API using JavaScript in a web page:
<https://andypi.co.uk/2016/08/06/accessing-a-web-api-using-javascript-in-a-web-page/>
- Azure*. (n.d.). Ανάκτηση από [12https://azure.microsoft.com/en-us/](https://azure.microsoft.com/en-us/)
- CodingBlonde*. (2015, 9 10). Ανάκτηση από CSS – Blonde Dictionary:
<http://codingblonde.com/2015/09/css-blonde-dictionary/>
- Diply.com*. (n.d.). Ανάκτηση από <http://diply.com/static/about>
- Flanagan. (2006). *Wikipedia*. Ανάκτηση από JavaScript:
<https://el.wikipedia.org/wiki/JavaScript>
- HTML*. (n.d.). Ανάκτηση από Wikipedia: <https://en.wikipedia.org/wiki/HTML>
- JQUERYHOUSE*. (n.d.). Ανάκτηση από jQuery Code Snippets for your Future Project:
<http://jqueryhouse.com/jquery-code-snippets-for-your-future-project/>
- Lizzie's Website*. (n.d.). Ανάκτηση από HTML: <http://estmary.2020.madeateps.org/html-structure/>
- Matt's Debates*. (n.d.). Ανάκτηση από sql server: <http://www.matt-thornton.net/tag/sql-server>
- Microsoft*. (n.d.). Ανάκτηση από Install Visual Studio 2017: <https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio>
- Microsoft Visual Academy*. (n.d.). Ανάκτηση από Visual Studio Courses:
<https://mva.microsoft.com/product-training/visual-studio-courses#!lang=1033>
- PCMAG*. (2011, 11 21). Ανάκτηση από <http://gr.pcmag.com/web-design/14300/help/ksekinontas-ue-ten-jquery>
- quora*. (n.d.). Ανάκτηση από What programming language is the most popular for web development?: [10https://www.quora.com/What-programming-language-is-the-most-popular-for-web-development](https://www.quora.com/What-programming-language-is-the-most-popular-for-web-development)
- Ramdhani, V. (2017, 08 22). *Stackoverflow*. Ανάκτηση από <https://stackoverflow.com/questions/226108/what-is-a-web-service-in-plain-english>
- Rick's Code*. (n.d.). Ανάκτηση από How to style a select dropdown with CSS:
<https://php.quicoto.com/how-to-style-a-select-dropdown-with-css/>
- udemy*. (n.d.). Ανάκτηση από Programming for Beginners in C# Windows Forms:
<https://www.udemy.com/programming-windows-applications-for-desktop-in-c-sharp/>

Wagner, B., Wenzel, M., tompratt-AQ, & Latham, L. (2017, 7 12). *Microsoft*. Ανάκτηση από Introduction to the C# Language and the .NET Framework:
<https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>

Wikipedia. (n.d.). Ανάκτηση από How to Make a JavaScript Image Rollover:
<http://www.wikihow.com/Make-a-JavaScript-Image-Rollover>

Wikipedia. (n.d.). Ανάκτηση από jQuery logo:
https://en.wikipedia.org/wiki/File:jQuery_logo.svg

Wikipedia. (n.d.). Ανάκτηση από <https://el.wikipedia.org/wiki/HTML>

Wikipedia. (n.d.). Ανάκτηση από CSS: https://en.wikipedia.org/wiki/Cascading_Style_Sheets

Wikipedia. (n.d.). Ανάκτηση από Microsoft Visual Studio:
https://en.wikipedia.org/wiki/Microsoft_Visual_Studio

Wikipedia. (n.d.). Ανάκτηση από Microsoft Azure:
https://en.wikipedia.org/wiki/Microsoft_Azure

Wikipedia. (n.d.). Ανάκτηση από SQL: <https://el.wikipedia.org/wiki/SQL>