

ΤΕΧΝΟΛΟΓΙΕΣ ΕΦΑΡΜΟΓΩΝ ΔΙΑΔΙΚΤΥΟΥ

ΕΦΑΡΜΟΓΗ
ΕΠΑΓΓΕΛΜΑΤΙΚΗΣ
ΔΙΚΤΥΩΣΗΣ

ΒΑΣΙΛΕΙΟΣ ΠΑΠΑΒΑΣΙΛΕΙΟΥ - 1115201400139
ΒΑΣΙΛΕΙΟΣ ΣΑΚΚΑΣ - 1115201400175



Περιεχόμενα

Εισαγωγή	3
Υλοποίηση	4
Βάση Δεδομένων	5
Εφαρμογή	7
Nearest Neighbor	19

Εισαγωγή

Στα πλαίσια αυτής της εργασίας, κατασκευάσαμε μια εφαρμογή επαγγελματικής δικτύωσης, το WorkConnect. Το WorkConnect επιτρέπει σε χρήστες να φτιάξουν έναν προσωπικό λογαριασμό, να εισάγουν πληροφορίες σχετικά με την επαγγελματική εμπειρία που διαθέτουν, καθώς και τις ικανότητες και ενδιαφέροντας που έχουν. Μπορούν να αναρτήσουν άρθρα που περιέχουν κείμενο, καθώς και εικόνες, βίντεο και αρχεία ήχου και μπορούν να δουν τις αναρτήσεις που έχουν κάνει άλλοι χρήστες, να κάνουν like και να αφήσουν σχόλια στα άρθρα αυτά.

Έχουν την δυνατότητα να δουν το προφίλ άλλων, χρηστών, να κάνουν αίτηση σύνδεσης μαζί τους και να ανταλλάξουν μηνύματα με αυτούς, εφόσον έχουν συνδεθεί με αυτούς. Μπορούν να δουν τα σχόλια και τα likes των συνδεδεμένων με αυτούς χρήστες, να ανταποκριθούν στα εισερχόμενα αιτήματα σύνδεσης.

Επιπλέον, οι συνδεδεμένοι χρήστες μπορούν να αναρτήσουν τις δικές τους αγγελίες ή να δουν τις αγγελίες που έχουν αναρτήσει άλλοι χρήστες και να κάνουν αιτήσεις σε αυτές. Οι αγγελίες, αλλά και οι αναρτήσεις στο χρονολόγιο του χρήστη επιλέγονται με τέτοιο τρόπο ώστε να εμφανίζονται οι πιο ενδιαφέρουσες και σχετικές για τον χρήστη με βάση το προφίλ, τα ενδιαφέροντα και τις επαφές του χρήστη.

Όσον, αφορά τον διαχειριστή της σελίδας, υπάρχει η δυνατότητα να δει το προφίλ όλων των χρηστών της εφαρμογής και να τροποποιήσει τα στοιχεία του χρήστη. Μπορεί, επίσης, να κατεβάσει ένα αρχείο XML το οποίο περιέχει όλες τις πληροφορίες για έναν ή και περισσότερους χρήστες της εφαρμογής.

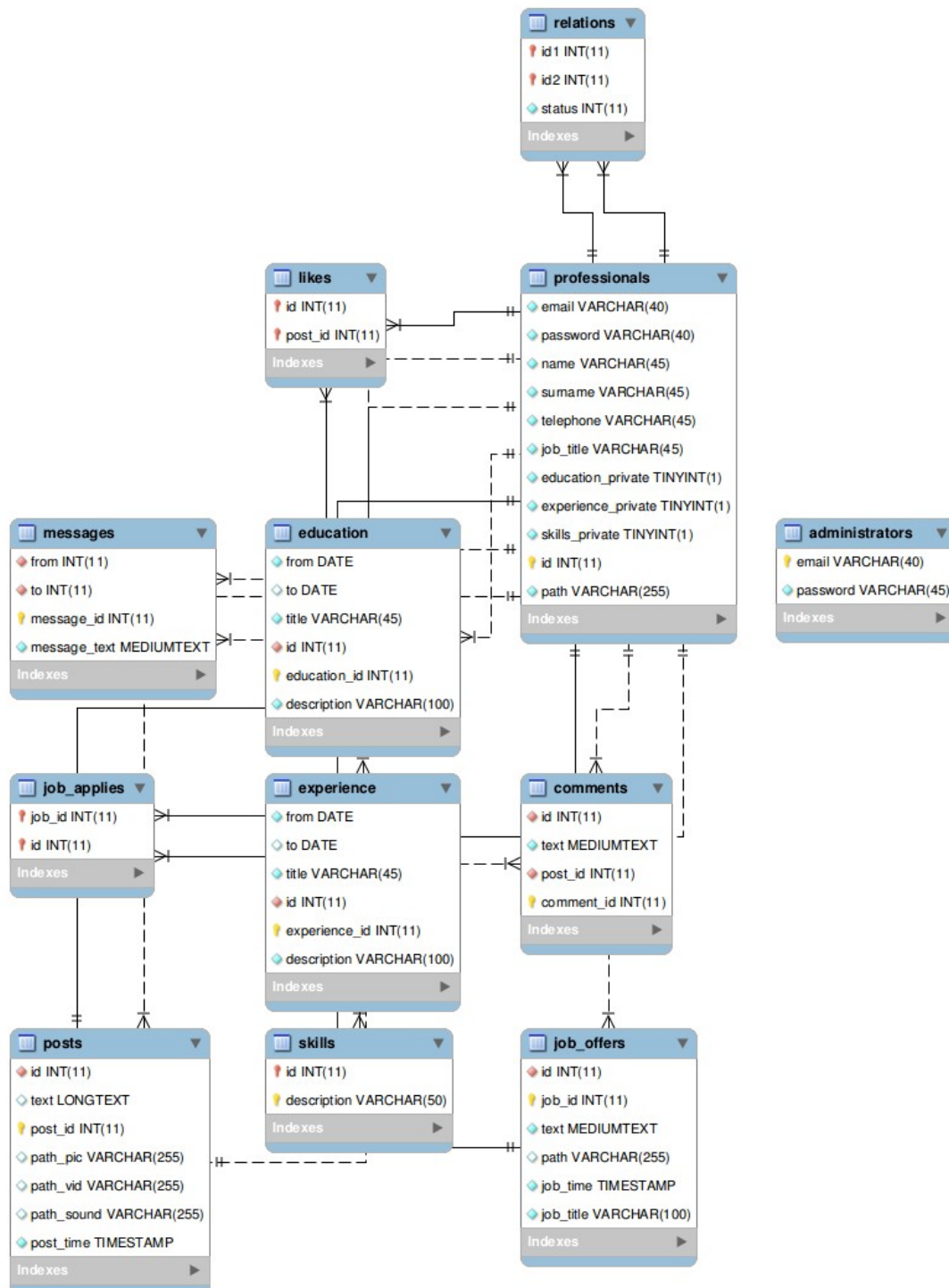
Υλοποίηση

Η εφαρμογή κατασκευάστηκε με βάση το πρότυπο MVC (Model-View-Controller). Έτσι, η εφαρμογή διαχωρίζεται σε τρία λειτουργικά μέρη που αλληλεπιδρούν μεταξύ τους και είναι όσον το δυνατόν πιο ανεξάρτητα. Πιο συγκεκριμένα, έχουμε:

- **Model:** Το model αποτελεί το κομμάτι της εφαρμογής που διαχειρίζεται την βάση δεδομένων και τη λογική της εφαρμογής. Συγκεκριμένα σε αυτό το κομμάτι ανήκουν τα αρχεία του package model που είναι η αναπαράσταση των δεδομένων σε data objects, τα αρχεία του package dao που αποτελούν την αλληλεπίδραση μεταξύ του Controller και του Model και τέλος το package helper που έχει μέσα λογική για κάποιες από τις λειτουργίες. Γενικά για την ανάκτηση δεδομένων όπου είναι δυνατόν αντί για τις συναρτήσεις των entities χρησιμοποιούνται JPQL queries. Το fetch των entities του model με την χρήση των συναρτήσεων γίνεται με lazy τρόπο. Επίσης πρέπει να αναφερθεί ότι ο Entity Manager ακολουθεί το σχεδιαστικό πρότυπο transaction-per-request για λόγους ευχρηστίας και απόδοσης.
- **View:** Αποτελείται από τις JSP σελίδες που είναι το κομμάτι της εφαρμογής με την αλληλεπιδρά ο χρήστης. Χρησιμοποιείται HTML και CSS για την μορφοποίηση των σελίδων. Πιο συγκεκριμένα, χρησιμοποιείται το W3.CSS προκειμένου να παρέχουμε μια καλά σχεδιασμένη εφαρμογή που μπορεί να προσαρμοστεί σε διάφορα είδη συσκευών με διαφορετικό μέγεθος οθόνης και/ή ανάλυσης. Επιπλέον, χρησιμοποιείται το Font Awesome έτσι ώστε να παρέχουμε μια ποικιλία εικονιδίων τα οποία χρησιμοποιούνται σε διάφορες σελίδες της εφαρμογής για να διευκολύνεται η σημασία των διάφορων κουμπιών και συνδέσμων που υπάρχουν. Τέλος, κάθε σελίδα της εφαρμογής περιέχει σε κάποιο βαθμό κώδικα JavaScript (JS) προκειμένου είτε να βελτιωθεί η εμπειρία χρήσης της εφαρμογής, όπως, π.χ. με τον έλεγχο των διαφόρων πλαισίων κειμένου που υπάρχουν σε πραγματικό χρόνο ή ώστε να γίνεται εύκολα η παραγωγή δυναμικού περιεχομένου σε περιπτώσεις όπου η επικοινωνία με τον Web Server δεν είναι αναγκαία. Η χρήση της Javascript έχει περιοριστεί αρκετά έτσι ώστε όσο το δυνατόν περισσότερες σελίδες να μπορούν να λειτουργούν χωρίς να είναι αναγκαία η ύπαρξη του εν λόγω κώδικα JavaScript. Κύριος στόχος δηλαδή, είναι να βελτιώνει την εμπειρία χρήσης προσθέτοντας μηδαμινό κόστος στον χρόνο φόρτωσης των σελίδων και χωρίς να είναι πάντα αναγκαία η χρήση της.
- **Controller:** Το κομμάτι του controller αποτελείται από τα servlets που βρίσκονται στο package servlets και αναλαμβάνουν την διασύνδεση μεταξύ του View και του Model. Γενικά ο χρήστης δεν έρχεται σε απευθείας επαφή με τα jsp αρχεία δηλαδή ο controller κάνει forward στα αντίστοιχα jsp με αποτέλεσμα να υπάρχει καλύτερος έλεγχος στο τι βλέπει ο χρήστης. Επίσης στο package filters υπάρχει φίλτρο για να μην κράταει ο browser το site στην cache (για λόγους ασφαλείας) και τέλος ένα φίλτρο το οποίο βοηθάει στην πραγματοποίηση του threadlocal transaction προτύπου (για να είναι δυνατό το transaction-per-request του Entity Manager).

Βάση Δεδομένων

Παρακάτω παρατίθεται ένα σχεδιάγραμμα της βάσης το οποίο βοηθά στην κατανόηση της δομής της βάσης. Γενικά έχει δοθεί μέριμνα ώστε η βάση να είναι σε κανονικοποιημένη μορφή προκειμένου οι εισαγωγές και εξαγωγές δεδομένων να είναι απλές.



Λεπτομέρειες σχετικά με την βάση:

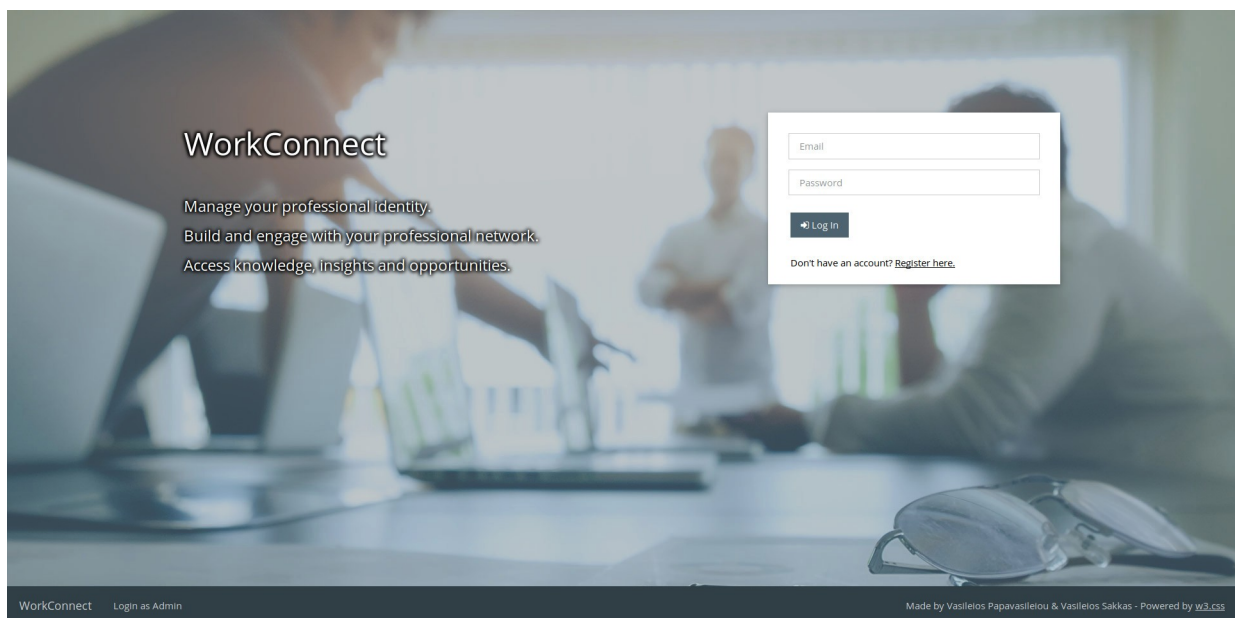
Στο table relations υπάρχει η μεταβλητή status η οποία παίρνει τιμές 0 άμα οι χρήστες είναι συνδεδεμένοι, 1 άμα υπάρχει κάποια αίτηση σύνδεσης από έναν χρήστη σε κάποιον άλλο χρήστη το οποίο όμως δεν έχει απαντηθεί ακόμη, και τέλος παίρνει τιμή 2 αν ο χρήστης απορρίψει το αίτημα σύνδεσης του άλλου χρήστη.

Εφαρμογή

Στην συνέχεια θα παρουσιάσουμε και θα εξηγήσουμε τις σελίδες της εφαρμογής:

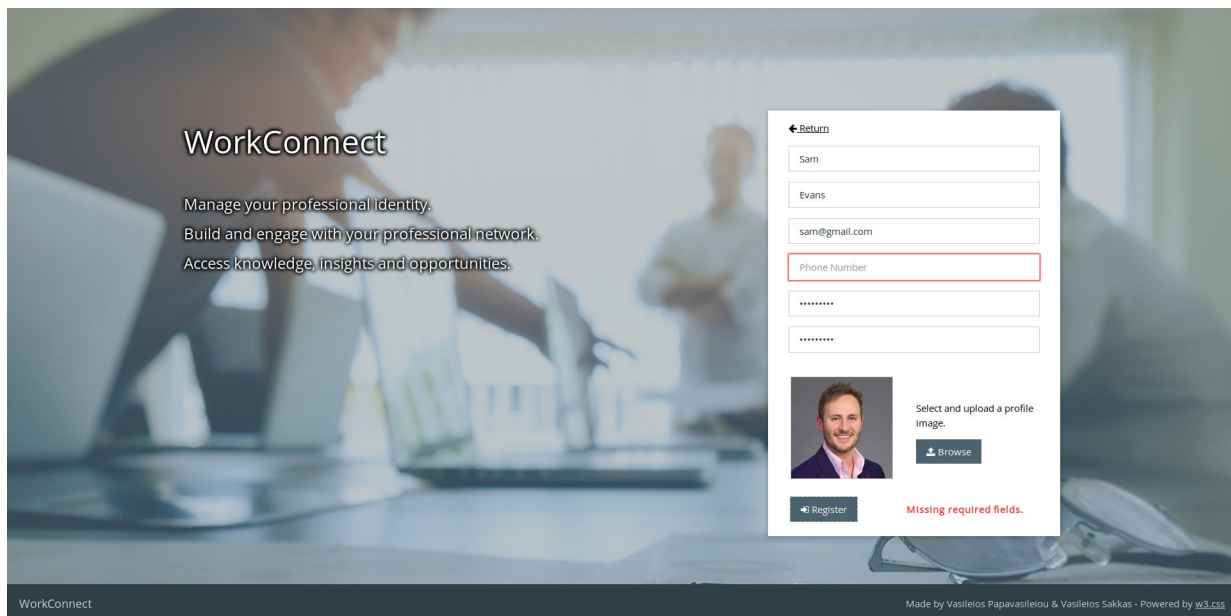
- **Σύνδεση Χρήστη (index.jsp)**

Πρόκειται για την αρχική σελίδα που βλέπει ένας χρήστης όταν επισκέπτεται τον ιστοχώρο και σκοπός της είναι να παρέχει ο επισκέπτης τα στοιχεία του ώστε να συνδεθεί στην εφαρμογή. Εάν δεν διαθέτει ήδη έναν λογαριασμό, υπάρχει σύνδεσμος που τον οδηγεί στην σελίδα εγγραφής. Επιπλέον, υπάρχει στο κάτω μέρος της σελίδας σύνδεσμος ώστε ο διαχειριστής της εφαρμογής να μπορέσει να συνδεθεί (μέσω μιας διαφορετικής φόρμας σύνδεσης). Η σελίδα χρησιμοποιεί κώδικα JS προκειμένου να ελέγχει την ορθότητα των στοιχείων που εισάγει ο χρήστης και να τον ενημερώνει για τυχόν ελλείψεις ή λάθη καθώς εισάγει τα στοιχεία του.



- **Εγγραφή Χρήστη (register.jsp)**

Διαθέτει μια φόρμα συμπλήρωσης στοιχείων, όπως το email, το όνομα, το επώνυμο, μια φωτογραφία προφίλ (με δυνατότητα να δει ο χρήστης την εικόνα που έχει επιλέξει πριν πατήσει να γίνει εγγραφή), κτλ. Όπως και με την σελίδα σύνδεσης, η σελίδα εγγραφής χρησιμοποιεί κώδικα JS προκειμένου να ελέγχει την ορθότητα των στοιχείων που εισάγει ο χρήστης και να τον ενημερώνει για τυχόν ελλείψεις ή λάθη καθώς εισάγει τα στοιχεία του.

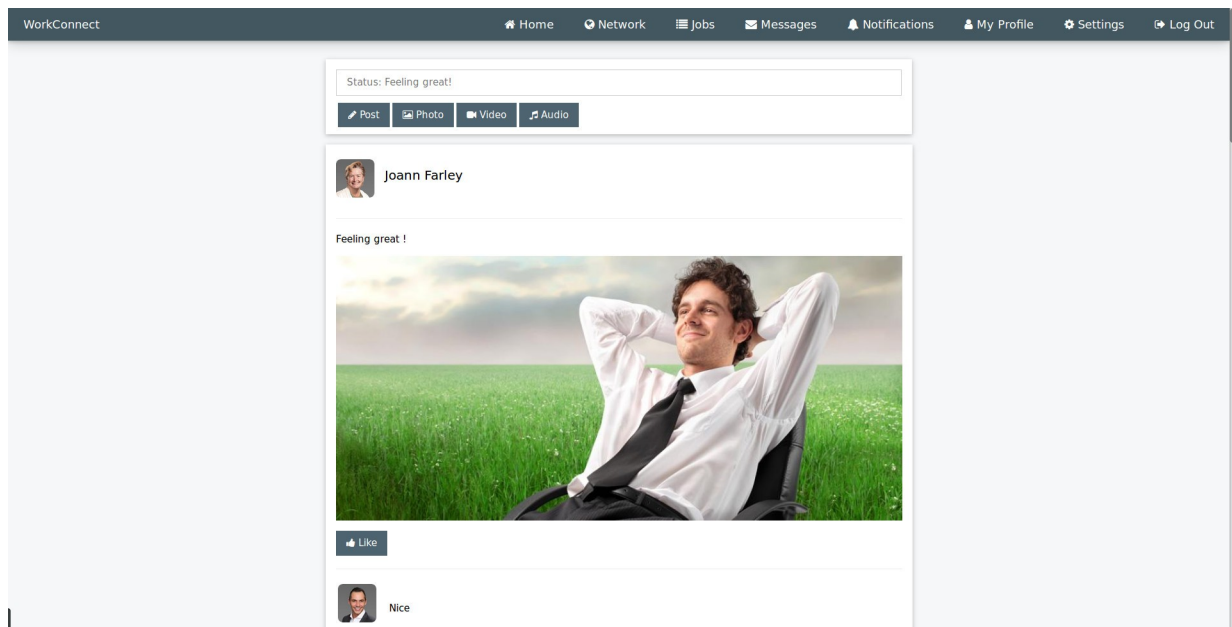


- **Αρχική Σελίδα/Χρονολόγιο (home.jsp)**

Όταν ο χρήστης συνδέεται ή κάνει εγγραφή, καθοδηγείται σε αυτήν την σελίδα. Εδώ ο χρήστης μπορεί να δει στο πάνω μέρος της σελίδας μια κεντρική μπάρα η οποία είναι διαθέσιμη σε κάθε σελίδα της εφαρμογής και παρέχει συνδέσμους προς όλες τις κεντρικές σελίδες της εφαρμογής (Home, Network, Jobs, Messages, Notifications, Profile, Settings και Log Out).

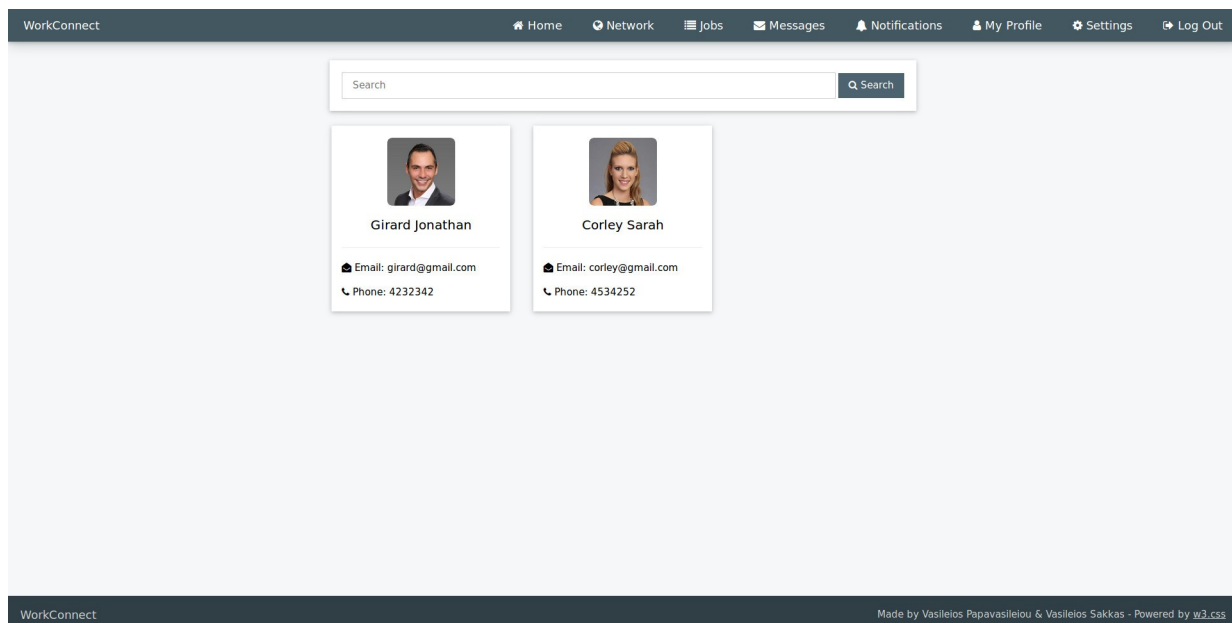
Επιπλέον, μπορεί ο χρήστης να δει τις αναρτήσεις που έχουν κάνει άλλοι χρήστες, όπως π.χ. οι χρήστες με του οποίους έχει συνδεθεί. Υπάρχει η δυνατότητα να κάνει like στις αναρτήσεις άλλων χρηστών ή και να αφήσει ένα ή περισσότερα σχόλια σε αυτές, ενώ μπορεί να δει και τα σχόλια που έχουν αφήσει άλλοι χρήστες. Οι αναρτήσεις που βλέπει ο χρήστης ταξινομούνται με βάση την χρήση του αλγορίθμου Nearest Neighbors που περιγράφεται παρακάτω.

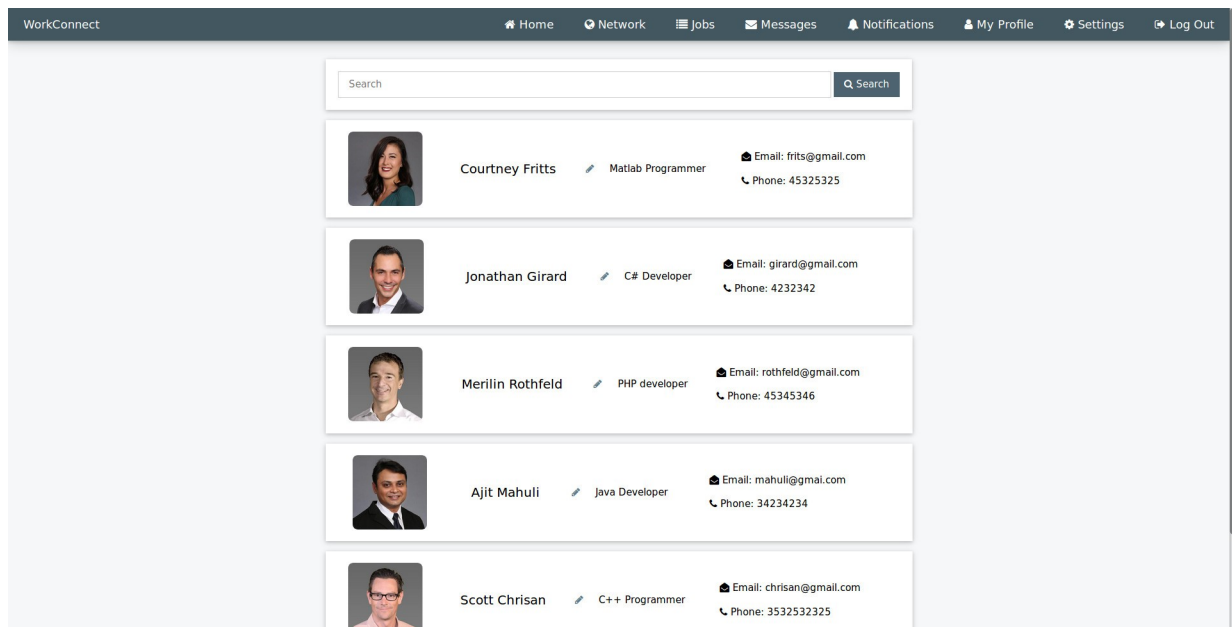
Τέλος, ο χρήστης μπορεί να κάνει τις δικές του αναρτήσεις μέσω μιας φόρμας που υπάρχει στο πάνω μέρος της σελίδας μέσω της οποίας μπορεί να προσθέσει στην ανάρτησή του κείμενο, εικόνες, βίντεο και αρχεία ήχου. Μια ανάρτηση μπορεί να περιέχει το πολύ μία εικόνα, το πολύ ένα βίντεο και το πολύ ένα αρχείο ήχου. Ακόμη, η φόρμα δημιουργίας αναρτήσεων εμφανίζει στον χρήστη το όνομα του τελευταίου αρχείου που έχει επιλέξει να αναρτήσει ώστε να διευκολύνεται στο να θυμάται πιο αρχείο επέλεξε. Εάν έχει ήδη επιλέξει ένα αρχείο ενός συγκεκριμένου τύπου (π.χ. εικόνα) και επιλέξει να αναρτήσει ένα διαφορετικό αρχείο του ίδιου τύπου (εικόνα), τότε η ανάρτηση που θα κάνει ο χρήστης θα περιέχει μόνο την δεύτερη εικόνα λόγω του περιορισμού που αναφέρθηκε νωρίτερα.



- **Δίκτυο και Αναζήτηση Χρηστών (network.jsp)**

Σε αυτήν την σελίδα μπορεί ο χρήστης να δει σε μορφή πλέγματος όλες τις επαφές του, κάποια βασικά στοιχεία για αυτούς τους χρήστες και μπορεί να οδηγηθεί στο προφίλ του κάθε χρήστη που εμφανίζεται στο πλέγμα. Υπάρχει επίσης, στο πάνω μέρος της σελίδας, μια μπάρα αναζήτησης μέσω της οποίας μπορεί ο χρήστης να εισάγει ένα όνομα και να βρει όλους τους χρήστες της εφαρμογής που έχουν αυτό το όνομα. Κατά την αναζήτηση ο χρήστης οδηγείται σε μια νέα σελίδα στην οποία εμφανίζονται οι χρήστες σε μορφή λίστας, μαζί με κάποιες πληροφορίες για αυτούς και την δυνατότητα να πλοηγηθεί ο χρήστης στο προφίλ τους.

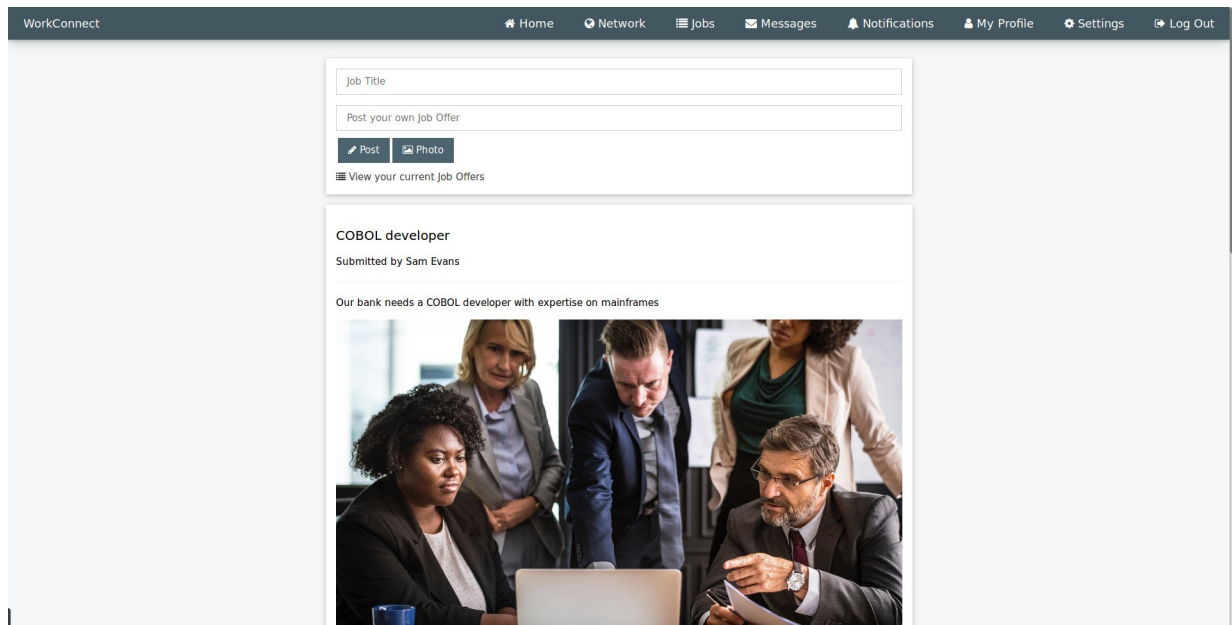




- **Αγγελίες (jobs.jsp)**

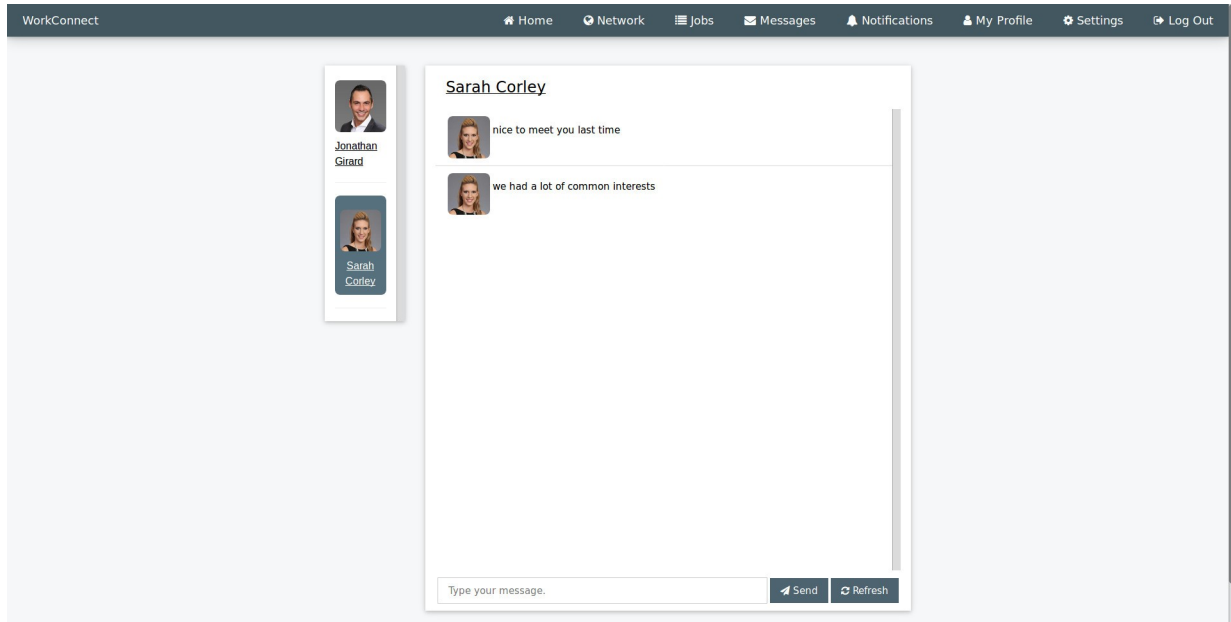
Στην σελίδα αυτή μπορεί ο χρήστης να βρει αγγελίες που έχουν αναρτήσει άλλοι χρήστες και να στείλει μια αίτηση για τις αγγελίες που τον ενδιαφέρουν. Οι αγγελίες που εμφανίζονται στην σελίδα ταξινομούνται με τον αλγόριθμο Nearest Neighbors που περιγράφονται παρακάτω.

Ακόμη, μπορεί ο χρήστης να αναρτήσει τις δικές του αγγελίες μέσω μιας φόρμας που βρίσκεται στο πάνω μέρος της σελίδας, οι οποίες περιέχουν κείμενο και προαιρετικά μία εικόνα. Παρόμοια με τις αναρτήσεις, όταν ο χρήστης επιλέξει μια εικόνα, μπορεί να δει το όνομα του αρχείου πριν αναρτήσει την αγγελία. Τέλος, κάτω από την φόρμα υπάρχει σύνδεσμος που οδηγεί τον χρήστη σε μια σελίδα που περιέχει τις δικές του αγγελίες, ή οποία αναλύεται παρακάτω.



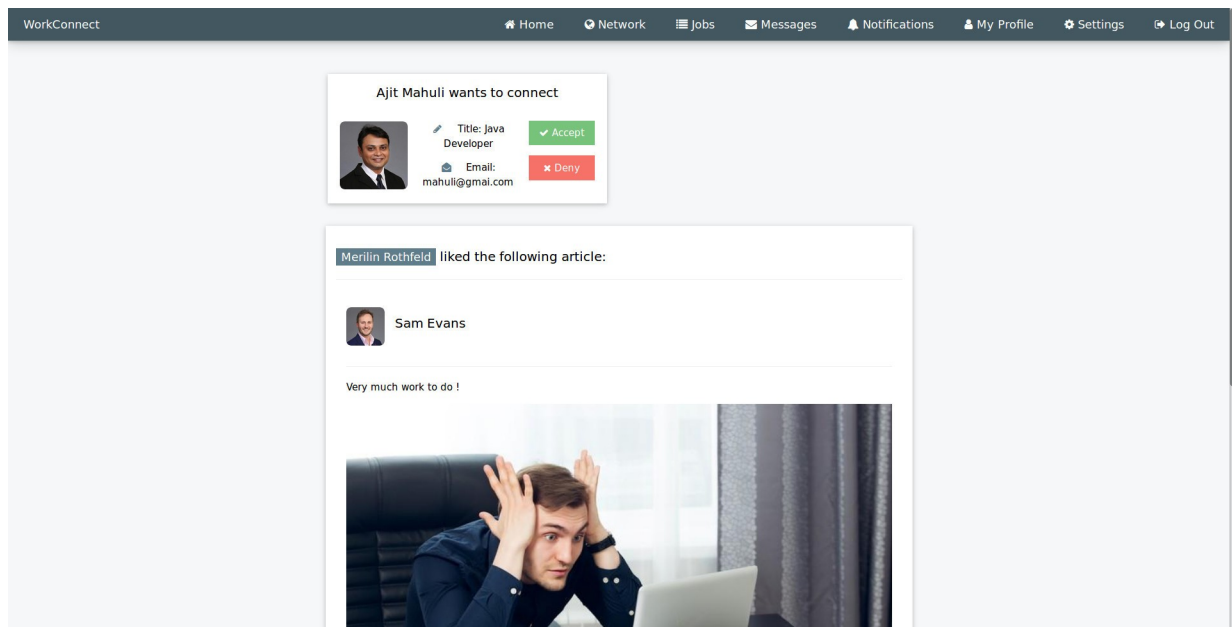
- **Συζητήσεις (messages.jsp)**

Στην σελίδα αυτή, ο χρήστης μπορεί να στείλει μηνύματα σε άλλους χρήστες. Αριστερά, υπάρχει μια λίστα με τις επαφές του χρήστη και όταν κάνει κλικ σε έναν από αυτούς μπορεί να δει τις τα μηνύματα που έχει ανταλλάξει μαζί του, καθώς και να στείλει ένα μήνυμα. Τα μηνύματα μπορούν να περιέχουν μόνο κείμενο. Επειδή δεν υπάρχει η δυνατότητα να εμφανίζονται νέα μηνύματα σε πραγματικό χρόνο, υπάρχει ένα επιπλέον κουμπί (Refresh) που επιτρέπει στον χρήστη να ξαναφορτώσει τα μηνύματα και έτσι, εάν υπάρχουν νέα μηνύματα να εμφανιστούν.



- **Ειδοποιήσεις (notifications.jsp)**

Η σελίδα αυτή αποτελείται από δύο μέρη. Στο πάνω μέρος της σελίδας εμφανίζεται ένα grid από αιτήσεις σύνδεσης από διάφορους χρήστες της εφαρμογής προς τον συνδεδεμένο χρήστη. Από εκεί, μπορεί ο χρήστης να δει κάποιες βασικές πληροφορίες για τους χρήστες που έχουν στείλει αίτηση και να δεχτεί ή να αρνηθεί την αίτηση σύνδεσης. Έπειτα, κάτω από τις ειδοποιήσεις, μπορεί ο χρήστης να δει ποιοι χρήστες έχουν κάνει like ή έχουν αφήσει κάποιο σχόλιο σε οποιοδήποτε από τα άρθρα του. Μαζί με την ειδοποίηση ενσωματώνεται και η ίδια η ανάρτηση που χρήστη μαζί με όλα τα σχόλια που έχουν γίνει.

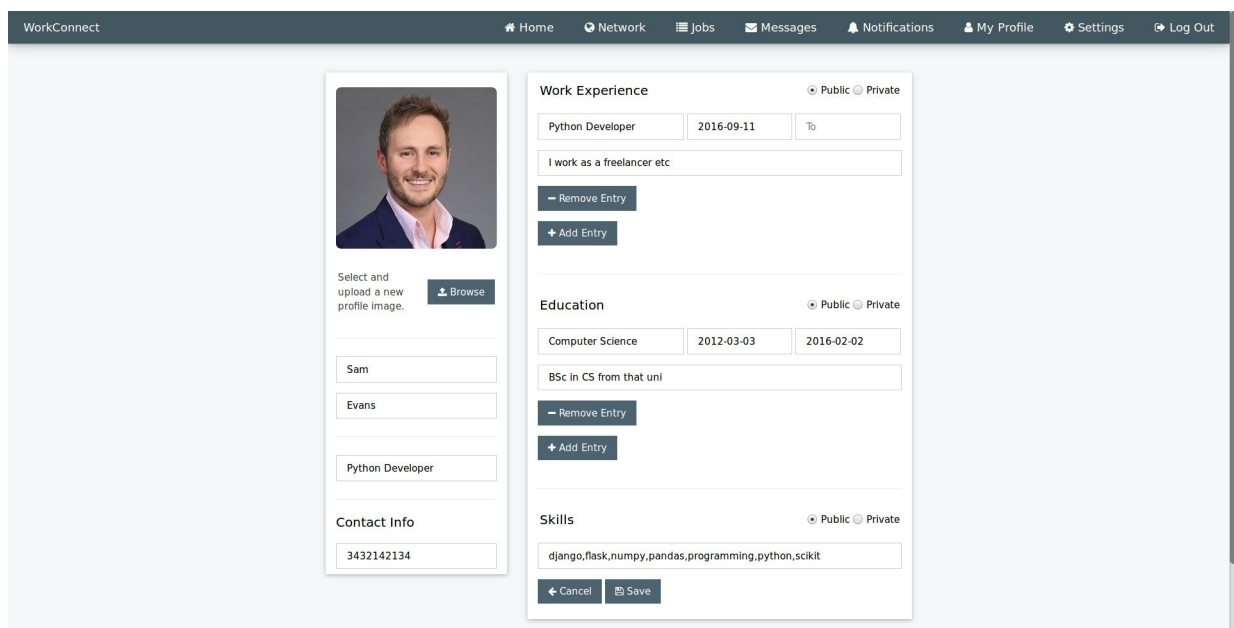
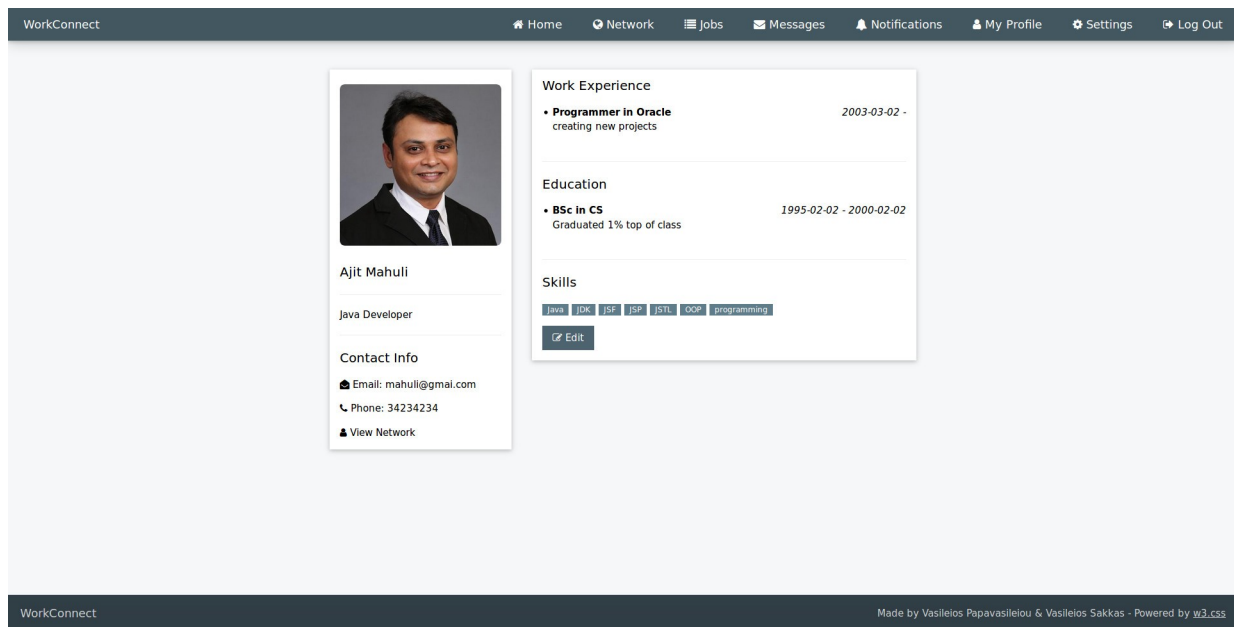


- **Προσωπικά Στοιχεία (profile.jsp & profile_edit.jsp)**

Σε αυτήν την σελίδα μπορεί ο χρήστης να δει το προφίλ του που περιέχει όλα τα στοιχεία που βλέπουν άλλοι χρήστες της εφαρμογής. Στο αριστερό τμήμα της σελίδας αυτής, υπάρχει η εικόνα προφίλ του χρήστη, καθώς και ο όνομα του, ο τίτλος εργασίας που έχει αυτήν την στιγμή, τηλέφωνο επικοινωνίας και ένας σύνδεσμος που οδηγεί στην σελίδα Δικτύου που περιγράψαμε νωρίτερα.

Στην δεξιά μεριά, υπάρχει μια λίστα με καταχωρήσεις που έχει κάνει ο χρήστης όσον αφορά την επαγγελματική εμπειρία που διαθέτει και παρακάτω υπάρχει μια αντίστοιχη λίστα για την εκπαίδευση που έχει. Παρακάτω, ακολουθούν οι ικανότητες και ενδιαφέροντα που έχει επιλέξει να αναρτήσει ο χρήστης και τέλος υπάρχει ένα κουμπί που επιτρέπει στον χρήστη να ανανεώσει το προφίλ του.

Η σελίδα ανανέωσης του προφίλ του χρήστη αποτελείται από πλαίσια αλλαγής και εισαγωγής στοιχείων του χρήστη. Όταν φορτώσει η σελίδα αυτή, τα πλαίσια είναι ήδη συμπληρωμένα με τα στοιχεία του χρήστη και έχει την δυνατότητα να αλλάξει τις τιμές τους, να αφαιρέσει κάποιες από αυτές ή και να προσθέσει καινούργιες. Πιο συγκεκριμένα, ο χρήστης μπορεί να ανεβάσει μια νέα φωτογραφία προφίλ, να αλλάξει το όνομα που προβάλλεται στο προφίλ του, να αλλάξει τον τίτλο εργασίας, να προσθέσει και να αφαιρέσει καταχωρήσεις από τις λίστες επαγγελματικής εμπειρίας, εκπαίδευσης και ικανοτήτων. Συγκεκριμένα, για την προσθήκη και την αφαίρεση καταχωρήσεων από τις λίστες επαγγελματικής εμπειρίας και εκπαίδευσης, είναι αναγκαία η χρήση της JS ώστε να προσθέτονται και να αφαιρούνται δυναμικά τα πλαίσια κειμένου. Όσον αφορά τις λίστες επαγγελματικής εμπειρίας, εκπαίδευσης και ικανοτήτων, ο χρήστης έχει την ικανότητα να κάνει την καθεμία από αυτές τις λίστες ιδιωτική, ώστε οι χρήστες που επισκέπτονται το προφίλ του να μην μπορούν να τις δουν. Τέλος, υπάρχει το κουμπί επιβεβαίωσης των αλλαγών και ένα κουμπί ακύρωσης των αλλαγών, δηλαδή, δεν στέλνεται κάποιο αίτημα στον Web Server.



- **Ρυθμίσεις (settings.jsp)**

Πρόκειται για μια απλή σελίδα που παρέχει στον χρήστη την δυνατότητα να αλλάξει το email και/ή τον κωδικό μέσα από δυο ξεχωριστές φόρμες συμπλήρωσης στοιχείων. Και οι δύο φόρμες χρησιμοποιούν κώδικα JS προκειμένου να ελέγχουν την ορθότητα των στοιχείων που εισάγει ο χρήστης και να τον ενημερώνουν για τυχόν ελλείψεις ή λάθη καθώς εισάγει τα στοιχεία του.

WorkConnect

Home Network Jobs Messages Notifications My Profile Settings Log Out

Email

not_an_email

Change invalid email.

Password

Change Passwords not matching.

WorkConnect

Made by Vasileios Papavasileiou & Vasileios Sakkas - Powered by w3.css

- **Σύνδεση Διαχειριστή (login_admin.jsp)**

Η σελίδα αυτή επιτρέπει στον Διαχειριστή της εφαρμογής να συνδεθεί με τον λογαριασμό του. Πρόκειται για μια απλή σελίδα που έχει μια φόρμα συμπλήρωσης στοιχείων και αρκεί ο διαχειριστής να εισάγει το email και τον κωδικό του. Η σελίδα χρησιμοποιεί κώδικα JS προκειμένου να ελέγχει την ορθότητα των στοιχείων που εισάγει ο διαχειριστής και να τον ενημερώνει για τυχόν ελλείψεις ή λάθη καθώς εισάγει τα στοιχεία του.

WorkConnect

Login as Admin

Email

Password

Log In

Not an Administrator? Go back to the Login page.

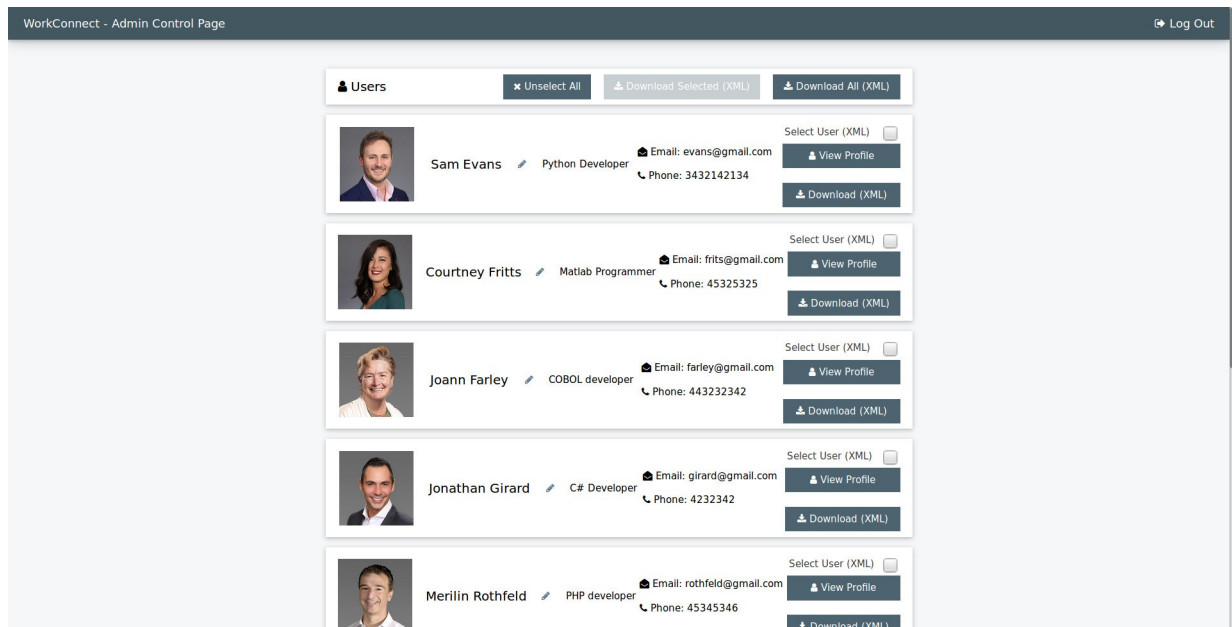
WorkConnect

Made by Vasileios Papavasileiou & Vasileios Sakkas - Powered by w3.css

- **Κεντρική Σελίδα Διαχειριστή (admin.jsp)**

Πρόκειται για μια σελίδα που χρησιμοποιεί μόνο ο διαχειριστής αφού συνδεθεί μέσω της ξεχωριστής για αυτόν σελίδας σύνδεσης. Εδώ ο διαχειριστής έχει πρόσβαση σε μια λίστα με όλους τους χρήστες της εφαρμογής. Μπορεί αν δει την φωτογραφία προφίλ του χρήστη, καθώς και μερικά στοιχεία του. Επιπλέον, έχει την δυνατότητα να μεταφερθεί στο προφίλ του χρήστη, το οποίο αναλύεται παρακάτω.


Ακόμη, ο διαχειριστής έχει την δυνατότητα να κατεβάσει ένα ή πολλά XML αρχεία τα οποία περιέχουν όλες τις χρήσιμες πληροφορίες για τον χρήστη. Ο διαχειριστής έχει στην διάθεσή του ορισμένα κουμπιά και checkboxes για να τον διευκολύνουν σε αυτό. Μπορεί με τα checkboxes να επιλέξει τους χρήστες που τον ενδιαφέρουν και με το κουμπί “Download Selected” να κατεβάσει ένα zip αρχείο που περιέχει όλα τα XML αρχεία που αναφέρονται στους χρήστες που είχε επιλέξει. Μπορεί ακόμη να πατήσει το κουμπί “Download All” προκειμένου να κατεβάσει ένα zip αρχείο που περιέχει όλα τα XML αρχεία, ένα για κάθε χρήστη της εφαρμογής. Τέλος, υπάρχει το κουμπί “Unselect All” ώστε να μπορεί εύκολα ο διαχειριστής να “ξεμαρκάρει” τους επιλεγμένους χρήστες. Επίσης πρέπει να αναφερθεί ότι χρησιμοποιείται η βιβλιοθήκη Jackson για την παραγωγή των αρχείων XML (με την βοήθεια annotations).



- **Προσωπικά Στοιχεία Χρηστών (profile_admin.jsp & profile_admin_edit.jsp)**
Πρόκειται για δύο σελίδες που είναι παρόμοιες με την σελίδα προφίλ που βλέπει ένας χρήστης όταν επισκέπτεται το δικό του προφίλ. Δηλαδή, ο διαχειριστής βλέπει όλα τα στοιχεία ενός χρήστη όπως θα έβλεπε τα δικά του, ενώ έχει την δυνατότητα να τροποποιήσει όλα τα στοιχεία σαν να ήταν το δικό του προφίλ. Ακόμη, η σελίδα προφίλ διαθέτει ένα επιπλέον κουμπί που επιτρέπει στον διαχειριστή να κατεβάσει ένα αρχείο XML που περιέχει τα στοιχεία του συγκεκριμένου χρήστη.

WorkConnect - Admin Control Page
Log Out

Return
Edit
Download (XML)



Sam Evans

Python Developer

Contact Info

Email: evans@gmail.com

Phone: 3432142134

View Network

Work Experience

- Python Developer**
2016-09-11 -
 I work as a freelancer etc

Education

- Computer Science**
2012-03-03 - 2016-02-02
 BSc in CS from that uni

Skills

django flask numpy pandas programming python scikit


WorkConnect
Made by Vasileios Papavasiliou & Vasileios Sakkas - Powered by w3.css

Πέρα από τις παραπάνω βασικές σελίδες, υπάρχουν και μερικές επιπλέον σελίδες που παρέχουν χρήσιμες λειτουργίες στην εφαρμογή και δεν είναι άμεσα διαθέσιμες μέσω της κεντρικής μπάρας της εφαρμογής:

- Προφίλ Χρήστη**

Κατά την πλοήγηση στον ιστοχώρο, υπάρχουν διάφοροι σύνδεσμοι που μπορούν να οδηγήσουν στο προφίλ ενός χρήστη. Η σελίδα αυτή είναι παρόμοια με την σελίδα που βλέπει ο χρήστης τα δικά του προσωπικά στοιχεία, με κάποιες διαφορές όμως. Εάν ο χρήστης που επισκεπτόμαστε έχει επιλέξει κάποια από τα στοιχεία του να είναι ιδιωτικά, τότε αυτά δεν εμφανίζονται στην σελίδα αυτή. Ακόμη, εάν δεν είμαστε συνδεδεμένοι με τον χρήστη αυτό, τότε υπάρχει η δυνατότητα να στείλουμε αίτημα σύνδεσης προς τον χρήστη αυτό, ενώ μέχρι είτε να δεχτεί, είτε να αρνηθεί το αίτημα σύνδεσης, δεν μπορούμε να ξαναστείλουμε άλλο αίτημα προς αυτόν τον χρήστη. Εάν είμαστε ήδη συνδεδεμένοι με τον χρήστη, τότε αντί για κουμπί σύνδεσης, υπάρχει ένα κουμπί που μας στέλνει απευθείας στα προσωπικά μηνύματα που έχουμε ανταλλάξει με τον χρήστη αυτό. Επιπλέον, εάν είμαστε συνδεδεμένοι με αυτόν τον χρήστη, είναι διαθέσιμος και ένας σύνδεσμος που μας οδηγεί σε μια λίστα με όλες τις επαφές του χρήστη αυτού.

WorkConnect
Home
Network
Jobs
Messages
Notifications
My Profile
Settings
Log Out



Joann Farley [Connect](#)

COBOL developer

Contact Info
Email: farley@gmail.com
Phone: 443232342

Work Experience

- IBM** 1970-03-23 -
Working in their systems

Education


- BSc in Physics** 1969-03-03 - 1974-02-03
Took electronics major

Skills

assembly cobol mainframes programming servers

WorkConnect
Made by Vasileios Papavasileiou & Vasileios Sakkas - Powered by w3.css

WorkConnect
Home
Network
Jobs
Messages
Notifications
My Profile
Settings
Log Out



Jonathan Girard [Message](#)

C# Developer

Contact Info
Email: girard@gmail.com
Phone: 4232342

[View Network](#)

Work Experience

- Startup DS** 2015-02-02 - 2016-02-02
senior advisor
- Microsoft** 2002-02-02 -
.NET Team

Education

- BSc in CS** 1998-02-03 - 2002-03-03
major in networking

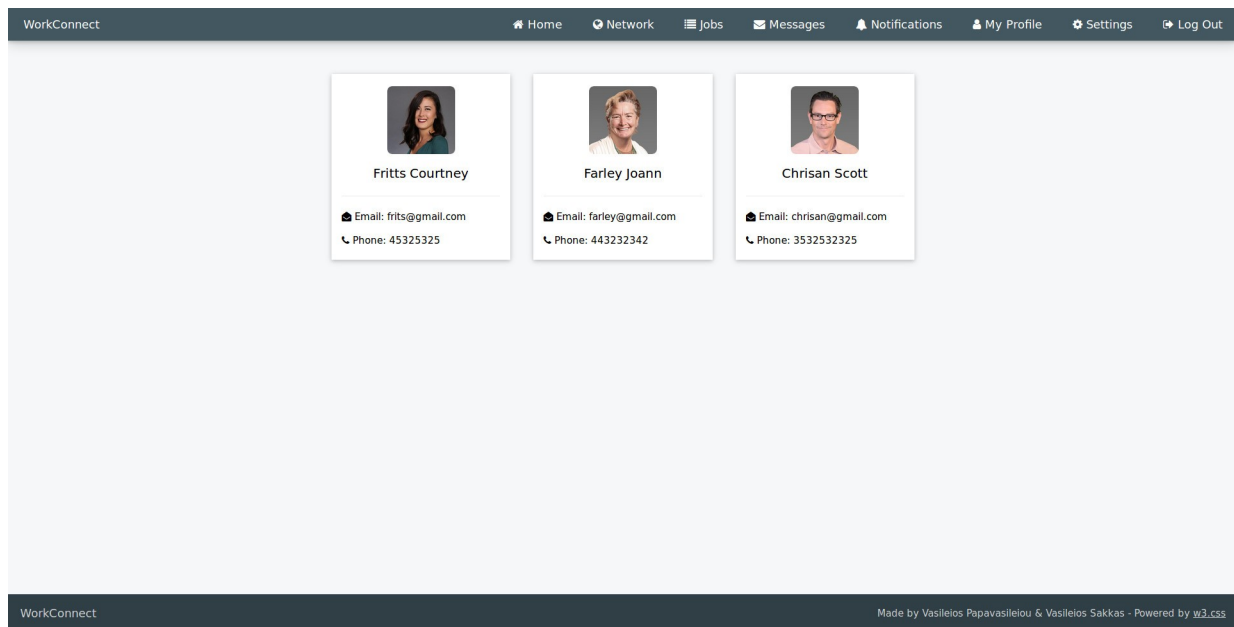
Skills

.NET Azure C# Microsoft programming Visual Studio

WorkConnect
Made by Vasileios Papavasileiou & Vasileios Sakkas - Powered by w3.css

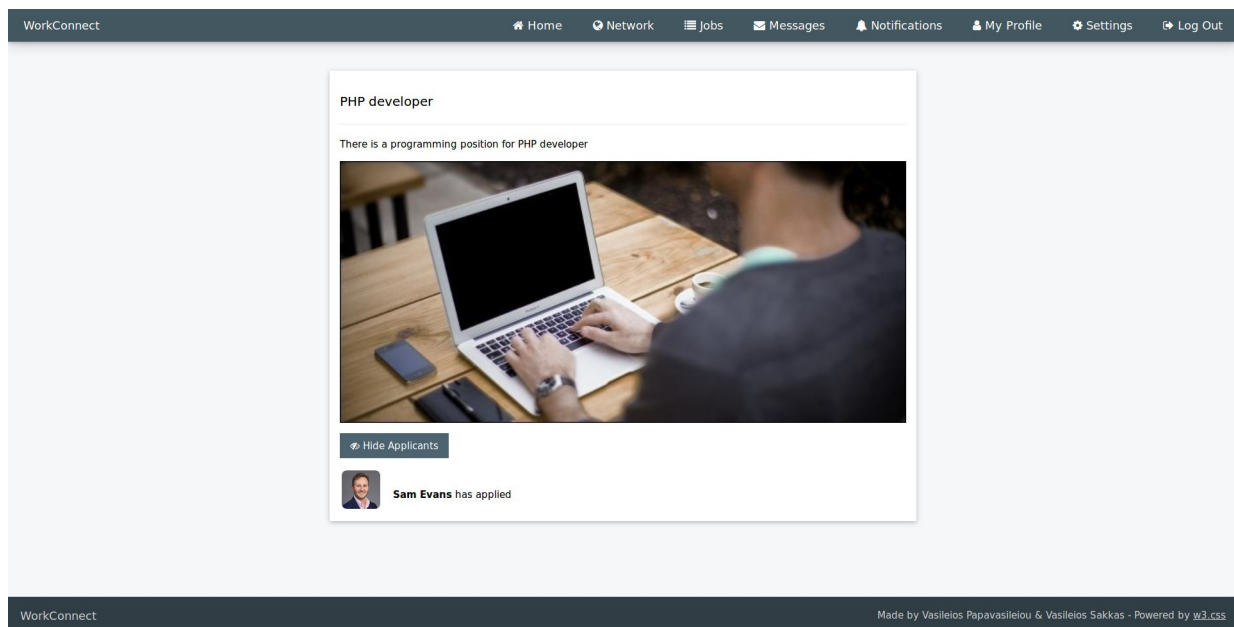
• Επαφές Χρήστη

Όπως αναφέρθηκε παραπάνω, όταν επισκεπτόμαστε το προφίλ ενός χρήστη με τον οποίο έχουμε συνδεθεί, μπορούμε να δούμε τις επαφές του. Ακόμη, όταν ο διαχειριστής επισκέπτεται το προφίλ οποιουδήποτε χρήστη, μπορεί επίσης να δει τις επαφές του. Η σελίδα αυτή περιέχει ένα πλέγμα από χρήστες και μας δείχνει κάποια στοιχεία αυτών των χρηστών μαζί με την φωτογραφία προφίλ τους. Κάθε στοιχείο αυτού του πλέγματος αποτελεί έναν σύνδεσμο που μας οδηγεί στην αντίστοιχη σελίδα προφίλ του χρήστη αυτού.



- **Ιστορικό Αγγελιών**

Όπως αναφέρθηκε, η σελίδα Αγγελιών διαθέτει έναν σύνδεσμο που οδηγεί τον χρήστη στην σελίδα με όλες τις αγγελίες που έχει αναρτήσει καθώς και μια λίστα κάτω από κάθε αγγελία με κάθε χρήστη που έχει κάνει αίτηση προς αυτήν την αγγελία. Για λόγους ευχρηστίας, υπάρχει επίσης και ένα κουμπί πάνω από κάθε λίστα με την οποία μπορεί ο χρήστης να κρύψει (και έπειτα να ξαναεμφανίσει εάν θέλει) την λίστα με τους χρήστες, ώστε να γίνεται πιο εύκολα και γρήγορα η πλοήγηση στις υπόλοιπες αγγελίες που έχει αναρτήσει.



Nearest Neighbor

Για την προβολή των αναρτήσεων στο χρονολόγιο του χρήστη καθώς και τις αγγελίες, χρησιμοποιείται ο αλγόριθμος Nearest Neighbor προκειμένου να ταξινομηθούν τα άρθρα και οι αγγελίες με τέτοιο τρόπο ώστε να εμφανίζονται στον χρήστη αναρτήσεις που τον ενδιαφέρουν. Αυτό γίνεται με βάση διάφορες πληροφορίες που διαθέτουμε για τον χρήστη όπως οι επαφές του, οι ικανότητες και τα ενδιαφέροντα του, τα άρθρα που στα οποία έχει κάνει like, κτλ. Αυτές οι πληροφορίες χρησιμοποιούνται ώστε να κατασκευάσουμε διανύσματα για τον χρήστη αλλά και για άλλους χρήστες όπως, π.χ. για τις επαφές του. Μετά την κατασκευή των διανυσμάτων χρησιμοποιείται η Ευκλείδεια απόσταση ώστε να μπορέσουμε να υπολογίσουμε την “απόσταση” μεταξύ των χρηστών, όπου μικρή απόσταση μεταξύ δύο χρηστών σημαίνει ότι οι χρήστες αυτοί έχουν αρκετά κοινά χαρακτηριστικά μεταξύ τους. Έπειτα, με βάση τις αποστάσεις που έχουν υπολογιστεί, γίνεται η ταξινόμηση των άρθρων/αγγελιών και η εκτέλεση του αλγορίθμου ολοκληρώνεται. Αν και η παραπάνω λογική εφαρμόζεται τόσο στον εύρεση σχετικών με τον χρήστη αναρτήσεων όσο και με τις αγγελίες, υπάρχουν κάποιες διαφορές στην πρακτική εφαρμογή του αλγορίθμου. Για αυτόν τον λόγο, υπάρχουν δύο ξεχωριστές συναρτήσεις που υλοποιούν τον αλγόριθμο, μία για την ταξινόμηση των αναρτήσεων και μία για την ταξινόμηση των αγγελιών.

- **Άρθρα/Αναρτήσεις**

Για την ταξινόμηση των άρθρων που ανεβάζουν οι χρήστες χρησιμοποιείται η συνάρτηση *NearestNeighborPosts*. Αυτή η συνάρτηση δέχεται ως παραμέτρους τον χρήστη που έχει συνδεθεί, μία λίστα από χρήστες και μία λίστα από άρθρα/αναρτήσεις. Όσον αφορά την λίστα από χρήστες, περιέχει όλες τις επαφές του συνδεδεμένου χρήστη. Επιπλέον, εάν κάποιος από τις επαφές του χρήστη έχει κάνει like ή comment σε άρθρο κάποιου χρήστη που δεν βρίσκεται ήδη σε αυτήν την λίστα, τότε προσθέτουμε τον χρήστη που είχε κάνει την ανάρτηση αυτή. Αυτό γίνεται ώστε να μπορούμε να προτείνουμε στον χρήστη άρθρα από χρήστες που θεωρούμε ότι ο συνδεδεμένος χρήστης θα έχει αρκετά κοινά χαρακτηριστικά, φροντίζοντας βέβαια τα άρθρα των επαφών του να εμφανίζονται πάντα, άσχετα από το με τι σειρά θα ταξινομηθούν. Η λίστα από αναρτήσεις περιέχει όλες τις αναρτήσεις που έγιναν τους τελευταίους δύο μήνες, ώστε να μην εμφανίζονται πολύ παλιά άρθρα στον χρήστη που δεν τον ενδιαφέρουν.

Όσον αφορά την ίδια την εκτέλεση του αλγορίθμου, αρχικά κατασκευάζουμε διανύσματα για κάθε χρήστη, δηλαδή για το συνδεδεμένο χρήστη, αλλά και για κάθε χρήστη από την λίστα που παρέχουμε στην συνάρτηση του αλγορίθμου. Τα διανύσματα αυτά είναι:

1) Ελέγχουμε εάν ο χρήστης είναι συνδεδεμένος ή όχι με κάθε χρήστη από την λίστα που περάσαμε στην συνάρτηση. Θεωρούμε σαν ειδική περίπτωση ότι ένας χρήστης είναι συνδεδεμένος με τον εαυτό του κατά τον υπολογισμό του διανύσματος. Αυτό το διάνυσμα μας επιτρέπει να δώσουμε προτεραιότητα σε χρήστες που είναι συνδεδεμένοι με τον χρήστη.

2) Ελέγχουμε εάν ο χρήστης έχει κάνει like ή όχι για κάθε άρθρο από την λίστα που παρέχουμε στην συνάρτηση. Αυτό το διάνυσμα μας βοηθάει να βρούμε κοινά ενδιαφέροντα μεταξύ των χρηστών μιας και δύο χρήστες που έχουν κάνει like σε πολλά κοινά άρθρα ενδεχομένως να μοιράζονται κοινές απόψεις και προτιμήσεις.

3) Ελέγχουμε εάν ο χρήστης έχει κάνει comment ή όχι για κάθε άρθρο από την λίστα που παρέχουμε στην συνάρτηση. Αυτό το διάνυσμα μας βοηθάει να βρούμε κοινά ενδιαφέροντα μεταξύ των χρηστών μιας και δύο χρήστες που έχουν κάνει comment σε πολλά κοινά άρθρα ενδεχομένως να μοιράζονται κοινές απόψεις και προτιμήσεις.

Αφού γίνει ο υπολογισμός των διανυσμάτων για όλους τους χρήστες που παρέχουμε στον αλγόριθμο, χρησιμοποιούμε αυτά τα διανύσματα για να υπολογίσουμε την Ευκλείδεια απόσταση του κάθε χρήστη από την λίστα από τον συνδεδεμένο χρήστη. Έπειτα,

ταξινομούμε αυτούς του χρήστες με βάση την απόσταση σε αύξουσα σειρά, αφού χρήστες με μικρότερη απόσταση έχουν πιο πολλά κοινά χαρακτηριστικά με τον συνδεδεμένο χρήστη. Στην συνέχεια, αφού ταξινομήσουμε τους χρήστες, “φιλτράρουμε” την λίστα των αναρτήσεων που περάσαμε στην συνάρτηση ώστε να περιέχει άρθρα από τους χρήστες τις λίστες που μόλις ταξινομήσαμε έτσι ώστε να αφαιρέσουμε άρθρα που μάλλον δεν αφορούν τον χρήστη. Τέλος ταξινομούμε τα άρθρα έτσι ώστε να είναι ταξινομημένα με βάση την ταξινόμηση των χρηστών που κάναμε νωρίτερα, δηλαδή, τα άρθρα από τους χρήστες που έχουν μικρή απόσταση από τον συνδεδεμένο χρήστη έχουν μεγαλύτερη προτεραιότητα. Τα άρθρα που επιστρέφει ο αλγόριθμος είναι τα άρθρα που θα δει ο χρήστης στην αρχική σελίδα, με την σειρά που τα επιστρέφει.

- **Αγγελίες**

Για την ταξινόμηση των αγγελιών που θα δει ο χρήστης χρησιμοποιείται η συνάρτηση *NearestNeighborJobOffers*. Η συνάρτηση αυτή δέχεται ως παραμέτρους τον χρήστη που έχει συνδεθεί και μια λίστα από αγγελίες. Όσον αφορά την λίστα, περιέχει όλες τις αγγελίες που αναρτήθηκαν τους τελευταίους δύο μήνες, ώστε να μην εμφανίζονται στον χρήστη παλιές αγγελίες που δεν τον ενδιαφέρουν.

Όσον αφορά την ίδια την εκτέλεση του αλγορίθμου, αρχικά κατασκευάζουμε διανύσματα για κάθε αγγελία για να βρούμε την απόσταση, δηλαδή την σχέση του χρήστη με την αγγελία. Τα διανύσματα αυτά είναι:

1) Ελέγχουμε εάν ο χρήστης είναι συνδεδεμένος με τον χρήστη που ανάρτησε την αγγελία. Αγγελίες από τις επαφές του χρήστη πρέπει να έχουν υψηλότερη προτεραιότητα σε σχέση με αγγελίες μη συνδεδεμένων χρηστών.

2) Ελέγχουμε πόσες κοινές επαφές έχουν ο συνδεδεμένος χρήστης με τον χρήστη που ανάρτησε την αγγελία. Εάν ο χρήστης που ανάρτησε την αγγελία έχει πολλές κοινές επαφές με τον χρήστη, τότε είναι πιθανό η αγγελία να ενδιαφέρει τον συνδεδεμένο χρήστη.

3) Ελέγχουμε το τίτλο και το περιεχόμενο της κάθε αγγελίας και τα συγκρίνουμε τις ικανότητες (skills) που έχει ο συνδεδεμένος χρήστης στο προφίλ του. Αυτό αποτελεί την πιο σημαντική παράμετρο για να βρούμε στις χρήσιμες για τον χρήστη αγγελίες, αφού ο ίδιος αναζητεί θέσεις εργασίας που ζητούν ικανότητες παρόμοιες με τις δικές του.

Αφού γίνει ο υπολογισμός των διανυσμάτων, χρησιμοποιούμε την Ευκλείδεια απόσταση για να βρούμε τις αποστάσεις των αγγελιών από τον χρήστη. Έπειτα, ταξινομούμε τις αγγελίες με βάση τις αποστάσεις που υπολογίσαμε σε αύξουσα σειρά, έτσι ώστε οι αγγελίες που έχουν μικρή απόσταση από τον χρήστη να εμφανιστούν υψηλότερα.

Σε αντίθεση με την συνάρτηση που χρησιμοποιήσαμε για την ταξινόμηση των άρθρων, εδώ δεν γίνεται κάποιο φιλτράρισμα αφού δεν μας ενδιαφέρει ποιος ανέβασε την αγγελία. Θέλουμε απλώς να βρούμε ποιες είναι πιο χρήσιμες για τον χρήστη.

Οι αγγελίες που επιστρέφει ο αλγόριθμος είναι οι αγγελίες που θα δει ο χρήστης στην σελίδα αγγελιών, με την σειρά που τις επιστρέφει.

Έτσι, λοιπόν, με την χρήση των δύο παραπάνω συναρτήσεων που στηρίζονται πάνω στον αλγόριθμο *Nearest Neighbor* μπορούμε να παρέχουμε στον κάθε χρήστη περιεχόμενο που τον ενδιαφέρει με βάση τις αλληλεπιδράσεις τους με άλλους χρήστες της εφαρμογής.