

ΤΕΧΝΟΛΟΓΙΚΟ  
ΕΚΠΑΙΔΕΥΤΙΚΟ  
Ι Δ Ρ Υ Μ Α



ΠΕΛΟΠΟΝΝΗΣΟΥ

Τ.Ε.Ι. Πελοποννησου

Σχολή Τεχνολογικών  
Εφαρμογών

Τμήμα Μηχανικών  
Πληροφορικής Τ.Ε.

**Σύστημα αναζήτησης και εύρεσης εργασίας και  
εργαζομένων με δυνατότητα προτάσεων συναφών  
θέσεων ή εργαζομένων.**

Αλεξανδρόπουλος Αθανάσιος ΑΜ: 2012068

ΣΕΠΤΕΜΒΡΙΟΣ 2018

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ

ΒΑΣΙΛΗΣ ΠΟΥΛΟΠΟΥΛΟΣ

## Περίληψη

Στόχος της εργασίας είναι η δημιουργία ενός συστήματος που θα αντιστοιχεί υποψήφιους εργαζόμενους σε θέσεις εργασίας. Οι θέσεις θα υποβάλλονται από εταιρείες, οι οποίες θα εγγράφονται στο σύστημα συμπληρώνοντας ένα προφίλ και οι υποψήφιοι εργαζόμενοι θα δημιουργούν ένα βιογραφικό, ώστε το σύστημα να κάνει αντιστοίχιση και να δείχνει προτάσεις στις εταιρείες για υποψήφιους που ταιριάζουν με τη περιγραφή της θέσης εργασίας και στους υποψήφιους εργαζόμενους για θέσεις εργασίας που ταιριάζουν στο προφίλ τους. Έπειτα η εταιρεία θα μπορεί να ειδοποιήσει τον υποψήφιο για να κανονίσουν κάποια συνέντευξη και ο υποψήφιος αντίστοιχα θα μπορεί να επικοινωνήσει με την εταιρεία για να δείξει ένα παραπάνω ενδιαφέρον για την θέση εργασίας. Για την υλοποίηση της εργασίας χρησιμοποιήθηκαν διάφορες τεχνολογίες διαδικτύου και αναπτύχθηκε εφαρμογή διαδικτύου.

## Δήλωση μη λογοκλοπής και ανάληψης προσωπικής ευθύνης

"Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάση επιστημονικής παράφρασης

Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων.

Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δε μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας."

Όνομα και Επώνυμο Συγγραφέα (Με Κεφαλαία): .....

Υπογραφή (Ολογράφως χωρίς μονογραφή):.....

Ημερομηνία (Ημέρα - Μήνας - Έτος):.....

# Πίνακας περιεχομένων

<b>1. Εισαγωγή</b>	<b>6</b>
<b>2. Περιγραφή Συστήματος</b>	<b>7</b>
2.1. Προδιαγραφές - Απαιτήσεις	8
2.2. Εργαλεία και τεχνολογίες ανάπτυξης λογισμικού	9
2.2.1. Ανάλυση απαιτήσεων και λογισμικού	9
2.2.2. XAMPP	9
2.2.2.1. Apache Web Server	10
2.2.2.1. PHPmyAdmin	10
2.2.3. Γλώσσες Προγραμματισμού	10
2.2.3.1. HTML	10
2.2.3.2. CSS	11
2.2.3.3. PHP	11
2.2.3.4. JavaScript	11
2.2.3.5. MySQL	12
2.2.4. Bootstrap	12
2.3. Λειτουργία αντιστοίχισης υποψήφιων εργαζόμενων και θέσεων εργασίας	12
<b>3. Ανάλυση απαιτήσεων και σχεδίαση συστήματος και λογισμικού</b>	<b>13</b>
3.1. Το μοντέλο του καταρράκτη	13
3.1.1. Πλεονεκτήματα	14
3.1.2. Μειονεκτήματα	14
3.2. Κατανόηση απαιτήσεων	15
3.3. Περιπτώσεις χρήσης	15
3.3.1. Περιπτώσεις χρήσης για τον υποψήφιο εργαζόμενο	15
3.3.1.1. 1η έκδοση	15
3.3.1.2. 2η έκδοση	16
3.3.2. Περιπτώσεις χρήσης για τον εργοδότη	17
3.3.2.1. 1η έκδοση	17
3.3.2.2. 2η Έκδοση	17
3.4. Διαγράμματα ροής	18
3.4.1. Ροή γεγονότων εργαζομένου	18
3.4.1.1. Κύρια ροή	18
3.4.1.2. Εναλλακτική ροή	19
3.4.1.3. Μετά-συνθήκες:	19
3.4.2. Ροή γεγονότων εργοδότη/εταιρείας	20
3.4.2.1. Κύρια ροή	20
3.4.2.2. Εναλλακτική ροή	21
3.4.2.3. Μετά-συνθήκες	21

3.5. Διάγραμμα Κλάσεων	22
3.6. Διάγραμμα Αντικειμένων	23
3.7. Διαγράμματα Συνεργασίας	24
3.8. Διάγραμμα σειράς	25
3.9. Διαγράμματα Δραστηριοτήτων	27
3.10. Διαγράμματα Καταστάσεων	28
3.10.1. Διάγραμμα για τη κλάση Job Offer	28
3.10.2. Διάγραμμα για τις κλάσεις Worker και Company	29
3.11. Ανάπτυξη Εφαρμογής	30
<b>4. Υλοποίηση και Έλεγχος</b>	<b>31</b>
4.1. Βάση δεδομένων	31
4.2. Σύνδεση βάσης με κώδικα	32
4.3. Η επικεφαλίδα, το στυλ και το υποσέλιδο	33
4.3.1. Η επικεφαλίδα (head.php)	33
4.3.1.1. Ταυτοποίηση	33
4.3.2. Το αρχείο css (style.css)	34
4.3.3. Υποσέλιδο (footer.php)	34
4.4. Αρχική Σελίδα	34
4.5. Μηχανισμός εισόδου/εξόδου (login/logout)	34
4.5.1. Είσοδος - login	34
4.5.2. Έξοδος - logout	35
4.5.3. Εγγραφή - Register	35
4.5.4. Διαγραφή χρήστη	35
4.6. Σελίδα Προφίλ	35
4.6.1. Υποψήφιος Εργαζόμενος	36
4.6.2. Εταιρεία	36
4.6.3. Φόρμα επικοινωνίας	36
4.6.4. Επεξεργασία προφίλ	36
4.6.4.1. Συλλογή δεδομένων	37
4.6.4.1.1. Υποψήφιος εργαζόμενος	37
4.6.4.1.2. Εταιρεία	37
4.6.4.2. Προβολή δεδομένων - Φόρμα	37
4.6.4.2.1. Υποψήφιοι εργαζόμενοι	37
4.6.4.2.2. Εταιρεία	37
4.6.4.3. Υποβολή στοιχείων στη βάση δεδομένων	38
4.6.4.3.1. Υποψήφιοι εργαζόμενοι	38
4.6.4.3.2. Εταιρεία	38
4.7. Θέση εργασίας	39
4.7.1. Δημιουργία νέας αγγελίας	39
4.7.2. Προβολή αγγελίας εργασίας	39
4.7.3. Επεξεργασία αγγελίας εργασίας	39
4.8. Αντιστοίχιση υποψήφιων εργαζομένων και θέσεων εργασίας	40

<b>5. Λειτουργία Συστήματος / Οδηγός Χρήσης</b>	<b>42</b>
5.1. Υποψήφιος Εργαζόμενος	44
5.2. Επιχείρηση	50
<b>Επίλογος</b>	<b>57</b>
Μελλοντική εξέλιξη	58
<b>Βιβλιογραφία</b>	<b>59</b>
<b>Παράρτημα Ι - Κώδικας εφαρμογής</b>	<b>61</b>

# 1. Εισαγωγή

Πολλές έρευνες σχετικά με τον τρόπο που ένα άτομο μπορεί να βρει εργασία μέσω του διαδικτύου έδειξαν ότι μέχρι πρότινος όλες οι διαθέσιμες πλατφόρμες πρόσφεραν ανοιχτές θέσεις εργασίας, τις οποίες έπρεπε να αναζητήσει ένα άτομο για να στείλει το βιογραφικό του. Από την άλλη, οι επιχειρήσεις μέχρι σήμερα προσλαμβάνουν μεσάζοντες - άλλες εταιρείες - οι οποίες φροντίζουν να συλλέγουν βιογραφικά, να πραγματοποιούν την πρώτη αξιολόγηση των υποψηφίων (με βάση τα κριτήρια που τους έχουν δοθεί) και σε δεύτερη φάση να προωθούν τους κατάλληλους υποψηφίους στους πελάτες τους (επιχειρήσεις). Αυτό το μοντέλο, αν και αποδοτικό αποτελεί αναπόφευκτα μια χρονοβόρα διαδικασία, υψηλού κόστους, μη δίνοντας στις επιχειρήσεις μια σαφή εικόνα των προσόντων των υποψηφίων εργαζομένων. Η ανάγκη αυτοματοποίησης της διαδικασίας εύρεσης εργαζομένων που να καλύπτουν τις ανάγκες των επιχειρήσεων μέσα από ένα περιβάλλον που οι ίδιες οι επιχειρήσεις θα μπορούσαν να ενημερώνονται για τα διαθέσιμα βιογραφικά, όπως έχουν ακολουθήσει νέες εφαρμογές όπως το [Linkedin.com](https://www.linkedin.com) και το [indeed.com](https://www.indeed.com), αποτέλεσαν το έναυσμα της δημιουργία της παρακάτω εφαρμογής μέσω αυτής της εργασίας. Σκοπός της εφαρμογής αποτελεί η άμεση διασύνδεση των επιχειρήσεων με τους υποψήφιους εργαζόμενους, ώστε οι πρώτες να θέτουν κριτήρια πρόσληψης και να λαμβάνουν ως αποτέλεσμα της αναζήτησής τους μέσω της εφαρμογής τα βιογραφικά που ανταποκρίνονται σε αυτές, και οι εργαζόμενοι να μπορούν να ακολουθήσουν την ίδια διαδικασία για την εύρεση μιας ικανοποιητικής εργασίας.

Η εργασία αποτελείται από τέσσερα κεφάλαια εκτός της εισαγωγής και του επιλόγου κάθε ένα από τα οποία αποτελεί συνέχεια του άλλου.

Συγκεκριμένα στο πρώτο κεφάλαιο περιγράφουμε το σύστημα μας, δηλαδή γίνεται η αναφορά των απαιτήσεων και των προδιαγραφών που προκύπτουν για την δημιουργία της εφαρμογής, την αναφορά στις τεχνολογίες που χρησιμοποιούνται για την υλοποίηση του κώδικα της εφαρμογής και μια εμβάθυνση στη θεωρία του τρόπου με τον οποίο θα επιτευχθεί η αυτόματη διασύνδεση θέσεων εργασίας με τους κατάλληλους εργαζόμενους και το αντίθετο.

Στο δεύτερο κεφάλαιο γίνεται η ανάλυση του συστήματος της εφαρμογής και συγκεκριμένα θα αναφερθούμε στη τρόπο ανάπτυξης λογισμικού που χρησιμοποιούμε και στη συνέχεια θα πραγματοποιηθεί η ανάλυση των απαιτήσεων που έχουν προκύψει.

Το τρίτο κεφάλαιο περιλαμβάνει την επεξήγηση του κώδικα που χρησιμοποιήθηκε για τη δημιουργία της εφαρμογής.

Και τέλος στο τέταρτο κεφάλαιο, δημιουργούμε ένα εγχειρίδιο χρήσης της εφαρμογής, όπου γίνεται μια παρουσίαση του τρόπου χειρισμού της εφαρμογής και από τους δυο τύπους χρήστη που περιλαμβάνει η εφαρμογή με απλό τρόπο. Κάθε κίνηση περιγράφεται γραπτώς και συνοδεύεται με στιγμιότυπα από την εφαρμογή, τα οποία δίνουν μια ολοκληρωμένη εικόνα αυτής.

Η παρούσα εργασία ολοκληρώνεται με τον επίλογο ο οποίος περιλαμβάνει μια σύνοψη των διαδικασιών που χρειάστηκαν να γίνουν για την ολοκλήρωση της εφαρμογής αλλά και την μελλοντική εξέλιξη που θα μπορούσε να έχει η εφαρμογή για να γίνει βιώσιμη.

## 2. Περιγραφή Συστήματος

Η εφαρμογή θα ακολουθήσει το πρότυπο του LinkedIn αλλά σε ένα πρώιμο στάδιο, δηλαδή για τη δημιουργία λογαριασμού, θέσης εργασίας και προτάσεων εργαζόμενων σε επιχειρήσεις και το αντίστροφο. Το LinkedIn με λίγα λόγια είναι ένας δικτυακός τόπος επαγγελματικής δικτύωσης, δηλαδή όπως το Facebook, αλλά για επαγγελματική χρήση. Χρησιμοποιείτε πολύ έντονα στο εξωτερικό, αλλά έχει αρχίσει να χρησιμοποιείται από το μεγαλύτερο μέρος των εταιρειών και στην Ελλάδα. Μπορεί να χρησιμοποιηθεί για την εύρεση υπαλλήλων, εργοδοτών, συγκεκριμένων θέσεων εργασίας με ανάλογες απαιτήσεις και προσόντα και τέλος φυσικά την επαγγελματική δικτύωση. Όπως μας μεταφέρει ο Chaidaroglou από το e-biografiko.gr “Υψηλόβαθμο στέλεχος του τμήματος επικοινωνιών της εταιρείας LinkedIn μας εξηγεί: Ο κύριος λόγος που οι εταιρείες χρησιμοποιούν το LinkedIn για να βρουν υπαλλήλους, είναι διότι είναι άμεσο και γρήγορο. Με μια αναζήτηση μπορούν να βρουν τους κατάλληλους ανθρώπους. Ένας άλλος λόγος που οι επιχειρήσεις χρησιμοποιούν το LinkedIn, είναι επειδή οι συστάσεις των υπαλλήλων τους έχουν μεγαλύτερη βαρύτητα, μιας και έχουν συνήθως μεγαλύτερα ποσοστά επιτυχίας. Επίσης, το LinkedIn βοηθά τις επιχειρήσεις να διευρύνουν και να αξιοποιήσουν πιο εύκολα το επαγγελματικό δίκτυο των υπαλλήλων τους.” Η δική μας εφαρμογή δεν ακολουθεί το μοτίβο του LinkedIn ως προς το δίκτυο των επαγγελματικών επαφών που μπορεί ο χρήστης να δημιουργήσει, αλλά το τρόπο που σου προτείνει κάποιες διαθέσιμες θέσεις εργασίας βάση των χαρακτηριστικών του προφίλ σου όπως φαίνεται στην εικόνα 2.1.

Based on your Profile and Career interests

2 job titles · Any location · Any industry · 0 to 1,000 employees ... [Update Career interests](#)

The screenshot displays a grid of job recommendations from LinkedIn. At the top, it states 'Based on your Profile and Career interests' and provides filters: '2 job titles · Any location · Any industry · 0 to 1,000 employees ... Update Career interests'. Below this, there are six job cards and a feedback panel. The first row contains three cards: 1. 'Wordpress Developer - RELOCATION TO MALTA' by Archer IT Recruitment, Athens, Attiki, Greece, posted 2 days ago. 2. 'Web Developer' by DRAXIS Environmental S.A., Thessaloniki, Central Macedonia, Greece, posted 1 week ago, with a 'Be an early applicant' badge. 3. 'PHP/WEB Developer' by Addit Digital Marketing, Greece, posted 1 week ago. The second row contains two more job cards: 4. 'Junior Java Software Developer' by blueground co, Athens, Attica, Greece, posted 7 days ago. 5. 'Software Engineer' by Edenred Greece, Greece, posted 7 days ago. The final panel on the right asks 'Are these jobs right for you?' and features two thumbs-up/down icons, with the text 'We'll use your feedback to improve future recommendations'.

Εικόνα 2.1 Τρόπος παρουσίασης θέσεων εργασίας από το LinkedIn  
(πηγή: <https://www.linkedin.com/jobs/>).



Η ανάπτυξη της εφαρμογής θα γίνει σε περιβάλλον HTML με χρήση php, apache server με το phpMyAdmin και MySQLi για τη διαχείρισή της βάσης, επίσης θα γίνει χρήση της Javascript για δημιουργία διαφόρων JS Queries, παρακάτω ακολουθούν οι προδιαγραφές και απαιτήσεις της.

## 2.1. Προδιαγραφές - Απαιτήσεις

Η ακριβής και σωστή καταγραφή των απαιτήσεων είναι πολύ σημαντική για την περαιτέρω ανάπτυξη της εφαρμογής. Όπως αναφέραμε και στην εισαγωγή υπάρχουν δύο είδη χρηστών και συγκεκριμένα υπάρχει ο υποψήφιος εργαζόμενος και η εταιρεία/επιχείρηση. Παρακάτω θα δούμε τις απαιτήσεις που αφορούν τον κάθε χρήστη αλλά και το σύνολο της εφαρμογής.

1. Θα υπάρχει δυνατότητα εγγραφής χρηστών και διαχείρισής τους
2. Εγγραφές θα γίνονται από:
  - 2.1. Εταιρείες/Επιχειρήσεις
    - 2.1.1. Οι εταιρείες θα εισάγουν και θα επεξεργάζονται στοιχεία τους όπως:επωνυμία, τύπο, έτος ίδρυσης, αριθμό υπαλλήλων, email και κωδικό εισόδου στο σύστημα.
    - 2.1.2. Καθώς και μια φόρμα με τη θέση εργασίας που παρέχουν και κάποια από τα χαρακτηριστικά που θέλουν να έχουν οι υποψήφιοι. Η κάθε εταιρεία θα μπορεί να υποβάλει πάνω από μια θέση εργασίας.
  - 2.2. Υποψήφιος/Φυσικά Πρόσωπα
    - 2.2.1. Οι υποψήφιοι θα εισάγουν και θα επεξεργάζονται στοιχεία όπως: Όνομα, επώνυμο, email για την υπηρεσία και κωδικό εισόδου.
    - 2.2.2. Επίσης θα μπορούν να βάλουν και περαιτέρω στοιχεία μετά την εγγραφή τους που θα δρουν ως το βιογραφικό τους, κάποια θα είναι: έτος γέννησης,τύπος πτυχίου,τύπο εργασίας που επιθυμούν(πχ μερικής απασχόλησης,κανονικής απασχόλησης),τυχόν συστατικές επιστολές/βεβαιώσεις(ονομαστικά),χρόνο προϋπηρεσίας κλπ.

Θα υλοποιηθεί ένα απλό περιβάλλον με χρήση php και html του οποίου οι σελίδες συσχετίζονται μεταξύ τους με ένα απλό σύστημα workflow. Το περιβάλλον αυτό αποτελεί ουσιαστικά ένα επίπεδο διασύνδεσης του περιεχομένου που δημιουργείται στην εφαρμογή από τις εταιρείες και τους υποψηφίους. Ειδικότερα, η εφαρμογή θα μπορούσε να χωριστεί σε τρία κομμάτια:

1. Το κομμάτι στο οποίο ο χρήστης παρέχει πληροφορίες συμπληρώνοντας φόρμες, με σκοπό την αποθήκευσή τους στη βάση.
2. Στο κομμάτι το οποίο ανακαλεί και προβάλλει στο χρήστη την αποθηκευμένη πληροφορία με σκοπό την αξιολόγηση της ή την εκτέλεση κάποιας άλλης ενέργειας.
3. Και στο κομμάτι στο οποίο γίνεται η αντιστοίχιση του βιογραφικού του υποψηφίου με τη κατάλληλη θέση εργασίας και το αντίθετο.

Η εφαρμογή θα στηθεί σε τοπικό περιβάλλον,σε μελλοντική φάση θα μπορούσε να γίνει μεταφορά της σε διαδικτυακό περιβάλλον,ώστε να είναι διαθέσιμη στο ευρύ κοινό.

## 2.2. Εργαλεία και τεχνολογίες ανάπτυξης λογισμικού

Τα εργαλεία ανάπτυξης λογισμικού είναι προγράμματα ή εφαρμογές τα οποία βοηθούν τους προγραμματιστές ή τους χρήστες να αναπτύσσουν, να διορθώνουν ή να βελτιώνουν τις εφαρμογές τους. Κάποια χαρακτηριστικά των εργαλείων αυτών είναι πως καταλαμβάνουν μικρό αποθηκευτικό χώρο στη μνήμη και μπορούν να ενσωματώνονται ή να συνδυάζονται με άλλα. Επίσης, πλέον υπάρχουν εργαλεία λογισμικού που καλύπτουν οποιαδήποτε ανάγκη ανάπτυξης λογισμικού. Όπως αναφέραμε και πιο πάνω στην παρούσα εφαρμογή γίνεται χρήση της γλώσσας προγραμματισμού PHP, της MySQL για την διαχείριση της βάσης δεδομένων, των CSS και HTML, Javascript καθώς και της πλατφόρμας XAMPP που δρα ως Apache web server.

### 2.2.1. Ανάλυση απαιτήσεων και λογισμικού

Για την ανάλυση των απαιτήσεων της εφαρμογής θα χρησιμοποιήσουμε τη UML. Η UML (Unified Modeling Language) είναι μια γενικής χρήσης, γλώσσα μοντελοποίησης στο κλάδο της ανάπτυξης λογισμικού, και παρέχει ένα τυποποιημένο τρόπο για το σχεδιασμό ενός συστήματος. Χρησιμοποιείται για τη γραφική απεικόνιση, προσδιορισμό, κατασκευή και τεκμηρίωση των στοιχείων ενός συστήματος λογισμικού. Η διαδικασία συλλογής και ανάλυσης των απαιτήσεων μιας εφαρμογής και η ενσωμάτωσή τους στο σχεδιασμό προγραμμάτων είναι πολύπλοκη και ο κλάδος υποστηρίζει πολλές μεθοδολογίες που καθορίζουν το τρόπο που μπορούμε να ακολουθήσουμε για την ανάλυση ενός προγράμματος. Ένα χαρακτηριστικό της UML είναι ότι δε βασίζεται σε κάποια συγκεκριμένη μεθοδολογία. Οπότε ανεξάρτητα της μεθοδολογίας που θα ακολουθηθεί για να πραγματοποιηθεί η ανάλυση και ο σχεδιασμός, η UML μπορεί να εκφράσει τα αποτελέσματα. Η UML αποτελείται από δεκατρείς τύπους διαγραμμάτων, τα οποία χωρίζονται σε τρεις κατηγορίες. Έξι για να απεικονίσουν τη στατική δομή της εφαρμογής κάποια εκ των οποίων είναι τα, διαγράμματα κλάσεων και τα διαγράμματα αντικειμένων. Τρία για να απεικονίζουν τους γενικούς τύπους συμπεριφοράς αυτά είναι τα διαγράμματα περιπτώσεων χρήσης, τα διαγράμματα δραστηριοτήτων και τα διαγράμματα καταστάσεων. Και τέλος τέσσερα διαγράμματα για την απεικόνιση διαφόρων πτυχών αλληλεπίδρασης αυτά είναι τα διαγράμματα επικοινωνίας, σειράς, συνεργασίας και τα χρονοδιαγράμματα.

### 2.2.2. XAMPP

Το XAMPP είναι μια πλατφόρμα ελεύθερου λογισμικού που δημιουργήθηκε από το Apache Friends για την περαιτέρω προώθηση και ανάπτυξη του Apache Web Server. Δημιουργήθηκε για να διευκολύνει τους προγραμματιστές, στην εγκατάσταση ενός Apache web server. Για να είναι βολικό για τους προγραμματιστές, το XAMPP έχει ρυθμιστεί με όλες τις λειτουργίες ενεργοποιημένες. Τα πακέτα που υπάρχουν διαθέσιμα είναι για Windows,

Linux αλλά και OS X. Με την εγκατάσταση του παρέχει όλα τα εργαλεία που χρειαζόμαστε όπως: μία έκδοση του Apache, της MySQL, της PHP, τη πλατφόρμα phpMyAdmin, το FileZilla FTP Server και το XAMPP Control Panel.

### 2.2.2.1. Apache Web Server

Ο Apache είναι ο δημοφιλέστερος εξυπηρετητής διαδικτύου (web server) που διατίθεται. Αυτό ισχύει γιατί διατίθεται δωρεάν, ο πηγαίος κώδικας του είναι ανοιχτός προς όλους (open source), μπορεί να υποστηρίξει και μικρές αλλά και μεγάλες ιστοσελίδες, υπάρχει ενσωματωμένη υποστήριξη για ένα ευρύ φάσμα γλωσσών προγραμματισμού όπως οι Perl, Python και PHP, περιλαμβάνει επίσης υποστήριξη SSL και TLS και υποστηρίζεται από μεγάλη ποικιλία λειτουργικών συστημάτων όπως Linux, Windows και Mac OS.

### 2.2.2.1. PHPmyAdmin

Το PhpMyAdmin αποτελεί ένα εργαλείο συγγραφής λογισμικού. Πιο συγκεκριμένα είναι ένα δωρεάν εργαλείο λογισμικού, το οποίο είναι γραμμένο σε γλώσσα php και το οποίο χρησιμοποιείται στη διαχείριση της MySQL μέσω του Παγκόσμιου Ιστού. Διαθέτει ένα πολύ φιλικό περιβάλλον χρήστη το οποίο επιτρέπει στον χρήστη να εκτελεί με ευκολία τις λειτουργίες διαχείρισης της βάσης δεδομένων. Κάποια χαρακτηριστικά του είναι ότι, έχει πολύ φιλικό περιβάλλον για το χρήστη, υποστηρίζει σχεδόν όλες τις λειτουργίες της MySQL, επιτρέπει την εισαγωγή δεδομένων τύπου CSV και SQL καθώς και την εξαγωγή δεδομένων υπό μορφές CSV, SQL, PDF, TEXT διαφόρων τύπων φύλλων λογισμικού ή κειμενογράφων όπως το Word.

## 2.2.3. Γλώσσες Προγραμματισμού

### 2.2.3.1. HTML

Η HTML είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Ο σκοπός ενός web browser είναι να διαβάσει τα έγγραφα HTML και να τα συνθέσει σε σελίδες που μπορεί κανείς να διαβάσει ή να ακούσει. Οι ετικέτες(tags) της HTML δεν εμφανίζονται από το περιηγητή, αλλά χρησιμοποιούνται για να ερμηνεύσουν το περιεχόμενο της σελίδας.

Στοιχεία της HTML χρησιμοποιούνται για να κτίσουν όλους τους ιστότοπους. Η HTML επιτρέπει την ενσωμάτωση εικόνων και άλλων αντικειμένων μέσα στη σελίδα, και μπορεί να χρησιμοποιηθεί για να εμφανίσει διαδραστικές φόρμες. Παρέχει τις μεθόδους δημιουργίας δομημένων εγγράφων, καθορίζοντας δομικά σημαντικά στοιχεία για το κείμενο, όπως κεφαλίδες, παραγράφους, λίστες, συνδέσμους, παραθέσεις και άλλα. Μπορούν επίσης να ενσωματώνονται σε αυτή σενάρια εντολών σε γλώσσες όπως η JavaScript, τα οποία επηρεάζουν τη συμπεριφορά των ιστοσελίδων HTML. Οι περιηγητές μπορούν επίσης να αναφέρονται σε στυλ μορφοποίησης CSS για να ορίζουν την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου υλικού.

### 2.2.3.2. CSS

Η CSS (Cascading Style Sheets - Διαδοχικά Φύλλα Ύφους) είναι μια γλώσσα προγραμματισμού που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης όπως η HTML. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στυλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

### 2.2.3.3. PHP

Η PHP είναι μια γλώσσα προγραμματισμού για τη δημιουργία σελίδων web με δυναμικό περιεχόμενο. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που είτε θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML ή θα επεξεργαστεί τις εισόδους δίχως να προβάλλει την έξοδο στο χρήστη, αλλά θα τις μεταβιβάσει σε κάποιο άλλο PHP script. Η PHP αποτελεί μια από τις πιο διαδεδομένες τεχνολογίες στο Παγκόσμιο Ιστό, καθώς χρησιμοποιείται από πληθώρα εφαρμογών και ιστότοπων. Η ευρύτητα στη χρήση της είναι απόρροια της ευκολίας που παρουσιάζει ο προγραμματισμός με αυτή αλλά και στο γεγονός πως είναι μια γλώσσα η οποία βρίσκεται σχεδόν σε κάθε διακομιστή. Διάσημες εφαρμογές που κάνουν εκτενή χρήση της PHP είναι το γνωστό Σύστημα Διαχείρισης Περιεχομένου (Content Management System, WordPress και το Drupal).

### 2.2.3.4. JavaScript

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης, η σύνταξή της είναι επηρεασμένη από τη C και αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστραφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού. Χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων όπως, έγγραφα PDF, εξειδικευμένοι φυλλομετρητές (site-specific browsers) και μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side). Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript

### 2.2.3.5. MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Το πρόγραμμα τρέχει έναν εξυπηρετητή (server) παρέχοντας πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων. Ο κωδικός του εγχειρήματος είναι διαθέσιμος μέσω της GNU General Public License, καθώς και μέσω ορισμένων ιδιοκτητών συμφωνιών. Ανήκει και χρηματοδοτείται από μία και μοναδική κερδοσκοπική εταιρία, τη σουηδική MySQL AB, η οποία σήμερα ανήκει στην Oracle. Η MySQL είναι από τα πιο δημοφιλή συστήματα διαχείρισης βάσης δεδομένων για διαδικτυακά προγράμματα και ιστοσελίδες. Χρησιμοποιείται σε κάποιες από τις πιο διαδεδομένες διαδικτυακές υπηρεσίες, όπως το Flickr, το YouTube, η Wikipedia, το Google, το Facebook και το Twitter.

### 2.2.4. Bootstrap

Το Bootstrap είναι μια συλλογή εργαλείων ανοιχτού κώδικα για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών, περιέχει στοιχεία HTML και CSS για τις μορφές τυπογραφίας, κουμπιά πλοήγησης και άλλων στοιχείων του περιβάλλοντος, καθώς και προαιρετικές επεκτάσεις JavaScript. Είναι από τα πιο δημοφιλή προγράμματα στο GitHub και έχει χρησιμοποιηθεί από τη NASA και το MSNBC, μεταξύ άλλων.

## 2.3. Λειτουργία αντιστοίχισης υποψήφιων εργαζόμενων και θέσεων εργασίας

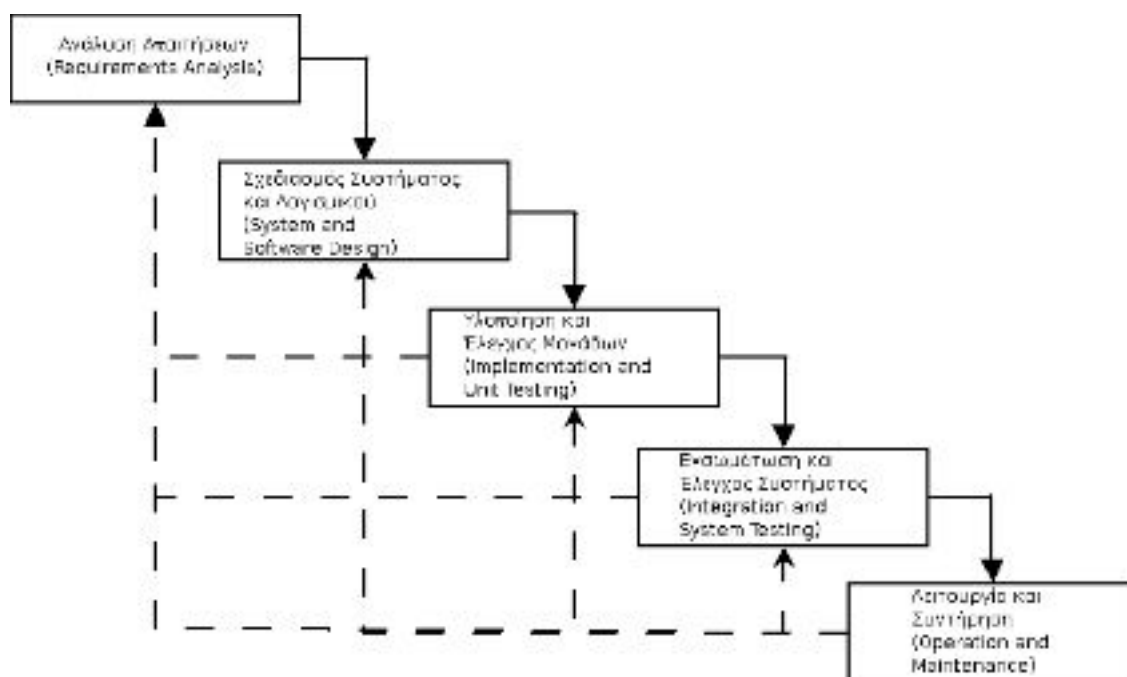
Η λειτουργία αυτή της εφαρμογής, θα είναι και η πιο σημαντική, είναι αυτή που συμβάλλει στη διευκόλυνση των υποψήφιων εργαζομένων για την εύρεση εργασίας και των εργοδοτών φτιάχνοντας μια αγγελία για κάποια θέση εργασίας να μπορέσει να βρει εύκολα και γρήγορα έναν εργαζόμενο με τα απαραίτητα κριτήρια. Στη πρώιμη αυτή φάση της εργασίας η διαδικασία της σύγκρισης και εύρεσης των αποτελεσμάτων θα γίνει πολύ απλά, συγκρίνοντας ένα περιορισμένο αριθμό στοιχείων που θα έχει δώσει ο υποψήφιος εργαζόμενος και η εταιρεία στην αγγελία εργασίας που έχει υποβάλει. Κάποια από τα στοιχεία αυτά θα είναι οι ικανότητες που έχει ο υποψήφιος και ζητείται από την εταιρεία στην αγγελία, το επίπεδο του πτυχίου του, της γλώσσες που ενδεχομένως γνωρίζει και ζητούνται, τη περιοχή που βρίσκεται ο εργαζόμενος και η αντίστοιχη εργασία κ.α.. Συγκρίνοντας τα αποτελέσματα αυτά, θα δημιουργείται ένα σύνολο πόντων, το οποίο θα χρησιμοποιείται για να βάλουμε σε σειρά, αυτοί που έχουν συλλέξει το μεγαλύτερο σύνολο πόντων θα είναι πρώτοι, τους εργαζόμενους που προτείνουμε για μια συγκεκριμένη θέση εργασίας σε μια επιχείρηση ή ακόμα και για το σύνολο των αγγελιών που παρέχει η επιχείρηση αυτή. Επίσης το αντίστοιχο θα γίνεται και στη περίπτωση των υποψήφιων εργαζομένων, με τη διαφορά ότι τώρα η κάθε αγγελία εργασίας θα μαζεύει ένα σύνολο πόντων και βάση αυτού θα ταξινομούνται από τη καλύτερη προς τον εργαζόμενο δηλαδή αυτή που καλύπτει τα χαρακτηριστικά του προς τη χειρότερη.

# 3. Ανάλυση απαιτήσεων και σχεδίαση συστήματος και λογισμικού

Σε αυτό το κεφάλαιο θα αναλύσουμε τις βασικές λειτουργίες του προγράμματος και τους περιορισμούς γύρω από αυτές. Το μοντέλο ανάλυσης και υλοποίησης του προγράμματος που θα χρησιμοποιήσουμε θα είναι το μοντέλο του καταρράκτη.

## 3.1. Το μοντέλο του καταρράκτη

Το μοντέλο του καταρράκτη είναι μια διαδοχική διαδικασία ανάπτυξης, στην οποία η ανάπτυξη θεωρείται πως διατρέχει σταθερά προς τα κάτω (όπως σε έναν καταρράκτη) της φάσεις της ανάλυσης των απαιτήσεων (Requirements) , του σχεδίου (Design), της εφαρμογής (Implementation), της επικύρωσης (Verification), της ολοκλήρωσης, και της συντήρησης (Maintenance). Στην Εικόνα 3.1 φαίνεται το μοντέλο του καταρράκτη και οι επιμέρους φάσεις του. Η πρώτη επίσημη περιγραφή του προτύπου καταρρακτών θεωρείται ότι είναι ένα άρθρο που δημοσιεύτηκε από τον Winston W. Royce το 1970 αν και ο ίδιος ο Royce (1970) δεν χρησιμοποίησε τον όρο "καταρράκτης" σε αυτό το άρθρο.



Εικόνα 3.1 Μοντέλο καταρράκτη

(πηγή: <http://www0.dmst.aueb.gr/louridas/lectures/dais/process/ar01s04.html> )

Για να ακολουθήσει κάποιος το πρότυπο του καταρράκτη, προχωρά από τη μια φάση στην επόμενη κατά τρόπο καθαρά διαδοχικό. Παραδείγματος χάριν, πρώτα ολοκληρώνονται οι προδιαγραφές των απαιτήσεων οι οποίες δεν τίθενται σε περαιτέρω διαπραγματεύσεις Όταν οι απαιτήσεις ολοκληρώνονται πλήρως, προχωρούμε με την σχεδίαση του συστήματος . Το εν λόγω λογισμικό σχεδιάζεται και ένα προσχέδιο του δίνεται στους προγραμματιστές για να το ακολουθούν - αυτό το προσχέδιο πρέπει να είναι ένα σχέδιο για την εφαρμογή των απαιτήσεων που έχουν ήδη παγιωθεί. Όταν το σχέδιο ολοκληρώνεται πλήρως, μια εφαρμογή του σχεδίου γίνεται από τους προγραμματιστές Προς τα τελευταία στάδια της φάσης εφαρμογής, τα χωριστά τμήματα λογισμικού συνδυάζονται για να ολοκληρωθεί η νέα λειτουργία. Το πρότυπο του καταρράκτη υποστηρίζει ότι κάποιος μπορεί να κινηθεί προς μια φάση μόνο όταν ολοκληρώνεται η προηγούμενη φάση της.

Σύμφωνα με τις βασικές αρχές του μοντέλου κάθε έργο διαιρείται σε διαδοχικές φάσεις, με κάποια επικάλυψη και κάποιες αναθεωρήσεις να είναι αποδεκτές μεταξύ των φάσεων. Έμφαση δίνεται στον προγραμματισμό , στα χρονικά προγράμματα, τις προβλεπόμενες ημερομηνίες, τους προϋπολογισμούς και την δημιουργία ενός ολόκληρου συστήματος συγχρόνως. Ο αυστηρός έλεγχος διατηρείται κατά τη διάρκεια της ζωής του προγράμματος μέσω της χρήσης εκτενούς γραπτής τεκμηρίωσης, καθώς επίσης και μέσω των επίσημων αναθεωρήσεων που έχουν την έγκριση του πελάτη.

### 3.1.1. Πλεονεκτήματα

Το μοντέλο αυτό είναι ιδανικό για να υποστηρίξει λιγότερο πεπειραμένες ομάδες και διευθυντές, ή ομάδες των οποίων σύνθεση κυμαίνεται. Η τακτική σειρά των βημάτων ανάπτυξης, οι αυστηροί έλεγχοι για την εξασφάλιση της επαρκούς τεκμηρίωσης και οι αναθεωρήσεις του σχεδίου εξασφαλίζουν την ποιότητα, την αξιοπιστία, και τη δυνατότητα συντήρησης του λογισμικού. Ένα επιπλέον πλεονέκτημα του μοντέλου είναι ότι η πρόοδος της ανάπτυξης των συστημάτων είναι μετρήσιμη. Τέλος ο εκτενής προγραμματισμός του υπό ανάπτυξη λογισμικού βοηθά στην βέλτιστη χρήση των πόρων .

### 3.1.2. Μειονεκτήματα

Αντίθετα το μοντέλο του καταρράκτη χαρακτηρίζεται ως άκαμπτο, αργό, δαπανηρό και δυσκίνητο λόγω της σημαντικής δομής και των αυστηρών ελέγχων. Το πρόγραμμα προχωρά προς τα εμπρός, με πιθανή μόνο τη μικρή μετακίνηση προς τα πίσω. Στο εν λόγω πρότυπο υπάρχει μικρό περιθώριο για επαναληπτική χρήση και επαναχρησιμοποίηση του λογισμικού. Ακόμη πολλά εξαρτώνται από τον πρόωρο προσδιορισμό και καταγραφή των απαιτήσεων, όμως οι χρήστες μπορεί να μην είναι σε θέση να καθορίσουν σαφώς τι χρειάζονται τόσο νωρίς στην εξέλιξη του έργου. Οι ασάφειες των προδιαγραφών, τυχόν ελλείψεις σε τμήματα των συστημάτων, και οι απροσδόκητες ανάγκες σε υποδομές ανακαλύπτονται συχνά κατά τη διάρκεια του της κωδικοποίηση Τα προβλήματα συχνά δεν ανακαλύπτονται μέχρι τη δοκιμή των συστημάτων Η απόδοση του συστήματος δεν μπορεί να εξεταστεί έως ότου το σύστημα κωδικοποιηθεί σχεδόν πλήρως, οπότε και είναι δύσκολο να διορθωθεί. Επιπλέον το μοντέλο του καταρράκτη είναι δύσκολο να ανταποκριθεί στις αλλαγές. Οι αλλαγές που εμφανίζονται αργά στον κύκλο ζωής του προγράμματος είναι δαπανηρές και η ενσωμάτωση τους είναι αποθαρρυντική . Το πρότυπο δε, παράγει

υπερβολική τεκμηρίωση και η συνεχής ενημέρωσή της είναι χρονοβόρα. Τέλος πολλές φορές οι γραπτές προδιαγραφές είναι δύσκολο να κατανοηθούν και να αξιολογηθούν από τους χρήστες του προγράμματος , γεγονός που πολλές δημιουργεί ένα χάσμα μεταξύ των υπευθύνων ανάπτυξης και των τελικών χρηστών.

## 3.2. Κατανόηση απαιτήσεων

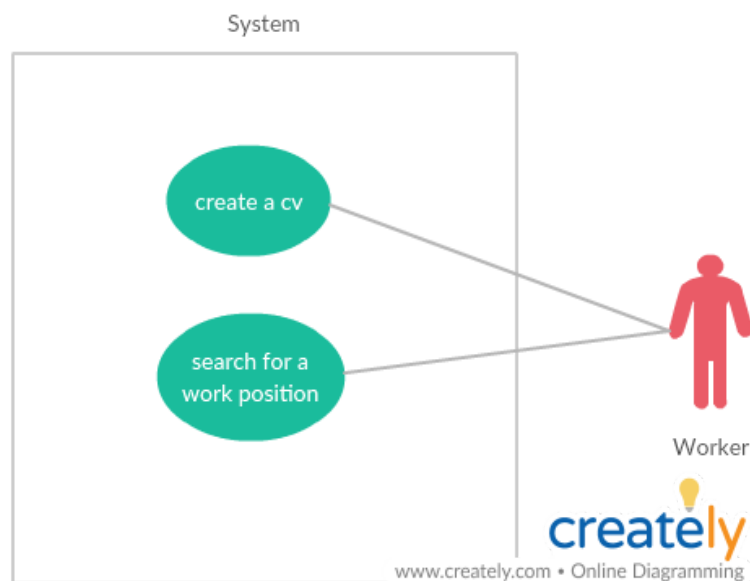
Το λογισμικό απευθύνεται σε επιχειρήσεις και οργανισμούς που θέλουν να προσλάβουν άτομα με συγκεκριμένα χαρακτηριστικά για τη κάλυψη θέσεων εργασίας στις επιχειρήσεις τους, αλλά και σε υποψήφιους που ψάχνουν μια ικανοποιητική θέση εργασίας. Μέσα από την εφαρμογή οι εταιρείες ακολουθώντας μια απλή διαδικασία θα μπορούν να επιλέξουν τον ιδανικό για αυτούς εργαζόμενο αποφεύγοντας τους μεσάζοντες και γλιτώνοντας πολύτιμο χρόνο. Οι υποψήφιοι εισάγοντας τα απαραίτητα στοιχεία του βιογραφικού τους, θα βλέπουν τις καταλληλότερες για αυτούς θέσεις εργασίας. Το σύστημα που στεγάζει το παραπάνω λογισμικό αποτελείται από έναν web-server που θα παρέχει τις απαραίτητες υπηρεσίες, όπως διαχείριση βάσης δεδομένων, για τη καλή λειτουργία του λογισμικού, στη προκειμένη περίπτωση θα υλοποιηθεί ο web-server σε τοπικό περιβάλλον με τη χρήση του πακέτουxampp, το οποίο παρέχει όλα τα προγράμματα που θα χρειαστούμε για την ανάπτυξη της εφαρμογής μας.

## 3.3. Περιπτώσεις χρήσης

Παρακάτω θα εξετάσουμε και θα αναλύσουμε με τη βοήθεια διαγραμμάτων περιπτώσεων χρήσης( use-case diagrams) τις περιπτώσεις χρήσης της εφαρμογής, από υποψήφιους εργαζόμενους και από εταιρείες - εργοδότες.

### 3.3.1. Περιπτώσεις χρήσης για τον υποψήφιο εργαζόμενο

#### 3.3.1.1. 1η έκδοση



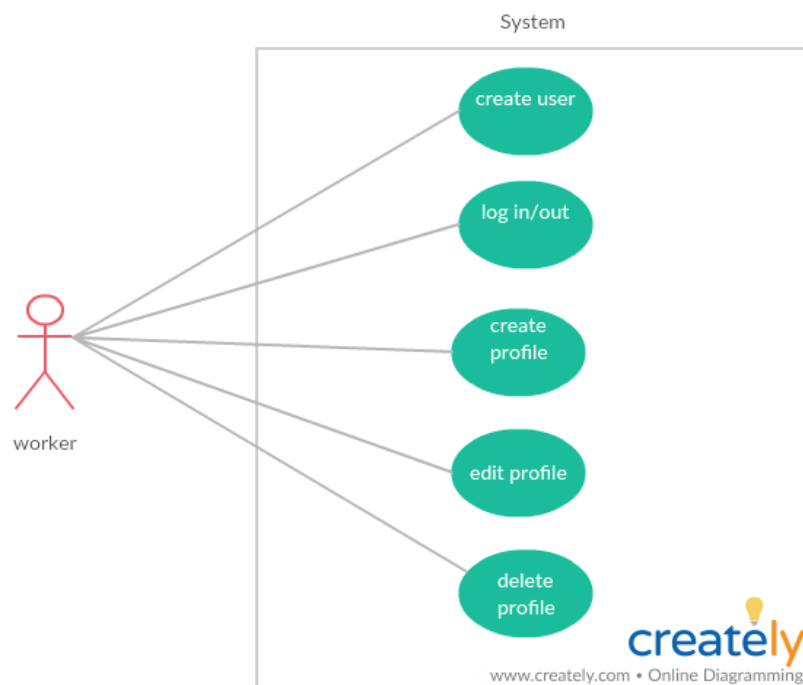


Εικόνα 3.2 1η έκδοση Διαγράμματος περιπτώσεων χρήσης υποψήφιου εργαζόμενου.

Στην εικόνα 3.2 βλέπουμε τις περιπτώσεις για τις οποίες ο υποψήφιος εργαζόμενος θα χρησιμοποιήσει την εφαρμογή. Οι κύριες λειτουργίες που θα μπορεί να κάνει θα είναι να δημιουργήσει ένα βιογραφικό και να αναζητήσει μια θέση εργασίας.

### 3.3.1.2. 2η έκδοση

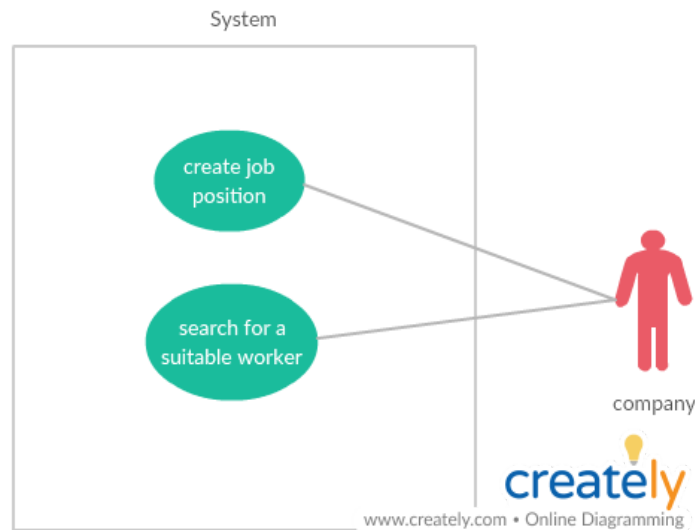
Από τη πρώτη έκδοση του διαγράμματος περιπτώσεων χρήσης προκύπτει ότι ο υποψήφιος εργαζόμενος θα πρέπει να δημιουργήσει αρχικά ένα προσωπικό προφίλ, το οποίο θα αποτελεί το βιογραφικό του, το οποίο θα αποθηκεύεται στην βάση δεδομένων της εφαρμογής και θα πρέπει να είναι προστατευμένο με κάποιο κωδικό για να μη μπορεί ο οποιοσδήποτε να δει τα στοιχεία του εκτός από τους εγγεγραμμένους χρήστες, οπότε θα πρέπει να δημιουργήσει ένα λογαριασμό στην εφαρμογή και θα πρέπει να υπάρχει μηχανισμός εισόδου/εξόδου από την εφαρμογή. Επίσης θα πρέπει να μπορεί να αλλάξει τα στοιχεία του βιογραφικού του ανά πάσα στιγμή αυτός επιλέξει ή ακόμα και να διαγράψει το λογαριασμό του και μέσω του μηχανισμού αυτόματης εύρεσης καταλληλότερης εργασίας βάση των στοιχείων του, να βλέπει τις διαθέσιμες θέσεις εργασίας για αυτόν. Οπότε προκύπτει το παρακάτω διάγραμμα της εικόνας 3.3 .



Εικόνα 3.3 2η έκδοση Διαγράμματος περιπτώσεων χρήσης υποψήφιου εργαζόμενου.

## 3.3.2. Περιπτώσεις χρήσης για τον εργοδότη

### 3.3.2.1. 1η έκδοση

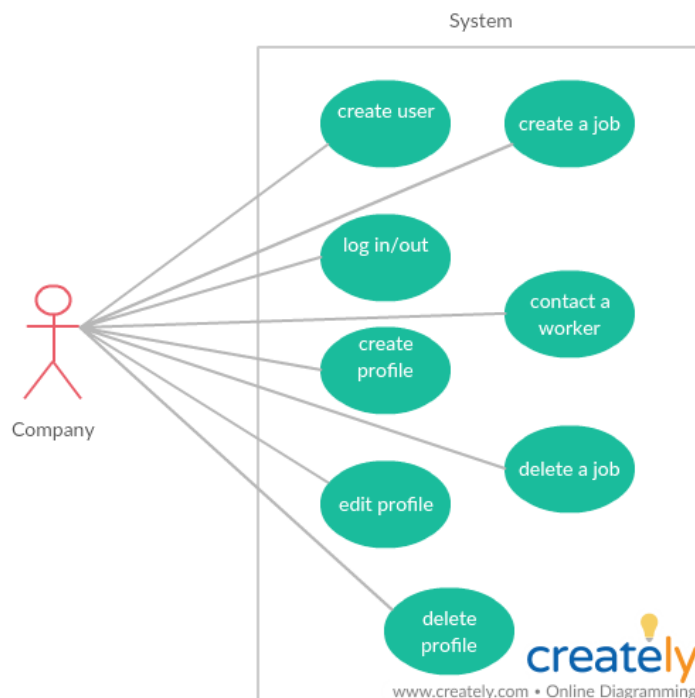


Εικόνα 3.4 1η έκδοση Διαγράμματος περιπτώσεων χρήσης εταιρείας.

Στην εικόνα 3.4 βλέπουμε το διάγραμμα περιπτώσεων χρήσης της εφαρμογής από μια εταιρεία. Οι κύριες δυνατότητες θα είναι να δημιουργήσει μια ή περισσότερες θέσεις εργασίας και να ψάξει για κάποιον υποψήφιο εργαζόμενο.

### 3.3.2.2. 2η Έκδοση

Από τη 1η έκδοση του διαγράμματος προκύπτει όπως και στη περίπτωση του υποψήφιου εργαζόμενου έτσι και στη περίπτωση της εταιρείας προκύπτει ότι, η εταιρεία θα πρέπει να δημιουργήσει αρχικά ένα εταιρικό προφίλ, το οποίο θα έχει κάποιες γενικές πληροφορίες για την εταιρεία, και θα αποθηκεύεται στην βάση δεδομένων της εφαρμογής και θα πρέπει να είναι προστατευμένο με κάποιο κωδικό για να μη μπορεί ο οποιοσδήποτε να δει τα στοιχεία αυτά εκτός από τους εγγεγραμμένους χρήστες, οπότε θα πρέπει να δημιουργήσει ένα λογαριασμό στην εφαρμογή και θα πρέπει να υπάρχει μηχανισμός εισόδου/εξόδου από την εφαρμογή. Επίσης θα πρέπει να μπορεί να αλλάξει τα στοιχεία αυτά ανά πάσα στιγμή ή ακόμα και να διαγράψει το λογαριασμό, επίσης στη περίπτωση της εταιρείας έχουμε και τη δημιουργία των θέσεων εργασίας, οπότε μια εταιρεία θα πρέπει να μπορεί να δημιουργήσει, να επεξεργαστεί αλλά και να διαγράψει μια θέση εργασίας και τέλος θα πρέπει να μπορεί να δει τους καταλληλότερους υποψήφιους για τη κάθε θέση εργασίας και να δίνεται πιθανός τρόπος επικοινωνίας. Οπότε προκύπτει το παρακάτω διάγραμμα της εικόνας 3.5.



Εικόνα 3.5 2η έκδοση Διαγράμματος περιπτώσεων χρήσης εταιρείας

## 3.4. Διαγράμματα ροής

Βάση των παραπάνω διαγραμμάτων περιπτώσεων προκύπτουν οι αντίστοιχες ροές γεγονότων.

### 3.4.1. Ροή γεγονότων εργαζομένου

#### 3.4.1.1. Κύρια ροή

Εξετάζουμε τη περίπτωση που ο υποψήφιος δεν έχει λογαριασμό στην εφαρμογή.

1. Ο υποψήφιος εργαζόμενος επισκέπτεται την ιστοσελίδα και βλέπει την σελίδα login.
2. Ο υποψήφιος εργαζόμενος επιλέγει το κουμπί «Register» από το μενού αν δεν έχει ήδη λογαριασμό.
3. Ο υποψήφιος εργαζόμενος μεταφέρεται στη σελίδα δημιουργίας λογαριασμού.
4. Εισαγωγή username, κωδικού πρόσβασης και email και ολοκλήρωση της λειτουργίας της εγγραφής επιλέγοντας το κουμπί «Submit»
5. Έπειτα ο υποψήφιος εργαζόμενος μεταφέρεται στο προφίλ του για να ολοκληρώσει το βιογραφικό του.
6. Πατώντας στο κουμπί «Edit» μεταφέρεται στη σελίδα επεξεργασίας προφίλ.
7. Ο υποψήφιος εργαζόμενος συμπληρώνει τα πεδία μιας φόρμας και ύστερα την υποβάλλει. (Ολοκλήρωση δημιουργίας προφίλ και αποθήκευση του ολοκληρωμένου προφίλ στη βάση δεδομένων)
8. Στη συνέχεια μεταφέρεται πάλι στο προφίλ του και πατάει το κουμπί « Search for a job».

9. Εμφανίζονται σε ένα πίνακα κάτω από το κουμπί οι καταλληλότερες θέσεις για αυτόν.

#### 3.4.1.2. Εναλλακτική ροή

Σε περίπτωση που ο υποψήφιος εργαζόμενος δεν θέλει να ολοκληρώσει την λειτουργία της εγγραφής και σύνδεσης μπορεί να επιλέξει το κουμπί «Cancel».

Σε περίπτωση που ο υποψήφιος εργαζόμενος δεν επιθυμεί να ολοκληρώσει τη συμπλήρωση της φόρμας του βιογραφικού μπορεί να επιλέξει το κουμπί «Cancel» και να τερματίσει τη διαδικασία.

Σε περίπτωση που ο χρήστης «υποψήφιος εργαζόμενος» επιθυμεί να επεξεργαστεί το βιογραφικό του, κάνει είσοδο στο σύστημα και έχει τη δυνατότητα επεξεργασίας της φόρμας του βιογραφικού πατώντας το κουμπί «Edit».

Σε περίπτωση που ο χρήστης «υποψήφιος εργαζόμενος» επιθυμεί να διαγράψει το βιογραφικό του, κάνει είσοδο στο σύστημα και έχει τη δυνατότητα διαγραφής του λογαριασμού πατώντας το κουμπί «Delete».

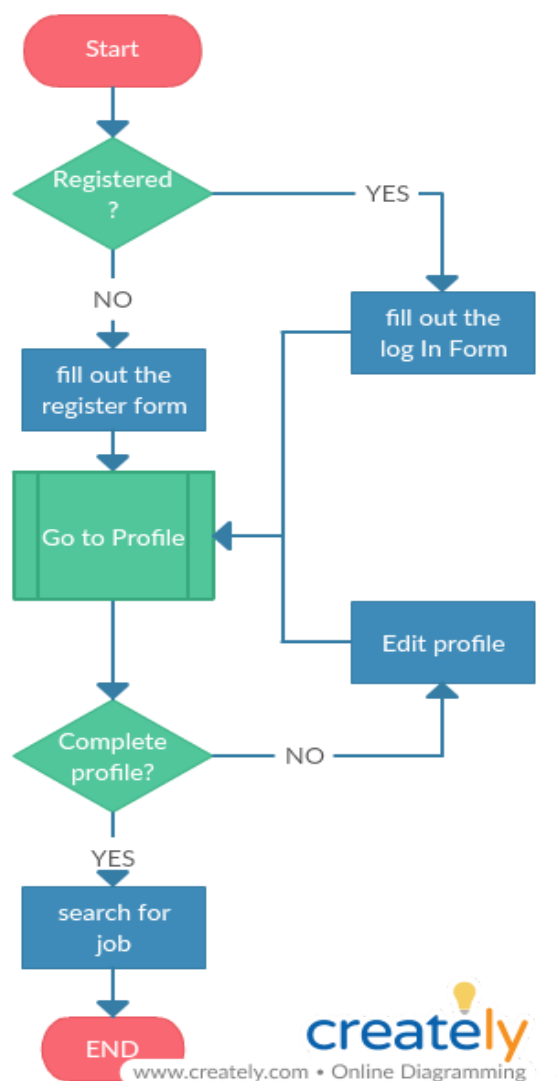
Σε περίπτωση που ο χρήστης “υποψήφιος εργαζόμενος” δεν επιθυμεί να συνεχίσει τις διαδικασίες επικαιροποίησης του βιογραφικού του, μπορεί να τερματίσει τη διαδικασία πατώντας το κουμπί «Cancel».

#### 3.4.1.3. Μετά-συνθήκες:

Όταν ο υποψήφιος εργαζόμενος κάνει εγγραφή στο σύστημα δημιουργείται μία καταχώρηση στη βάση δεδομένων.

Αφού ο υποψήφιος εργαζόμενος ολοκληρώσει τη δημιουργία του βιογραφικού του, το βιογραφικό αποθηκεύεται στη βάση δεδομένων του συστήματος.

Από τα παραπάνω προκύπτει η εικόνα 3.6 .



Εικόνα 3.6 Διάγραμμα ροής υποψήφιου εργαζομένου.

## 3.4.2. Ροή γεγονότων εργοδότη/εταιρείας

### 3.4.2.1. Κύρια ροή

Εξετάζουμε τη περίπτωση που η εταιρεία δεν έχει λογαριασμό στην εφαρμογή.

1. Η εταιρεία επισκέπτεται την ιστοσελίδα και βλέπει την σελίδα login.
2. Η εταιρεία επιλέγει το κουμπί «Register» από το μενού αν δεν έχει ήδη λογαριασμό.
3. Η εταιρεία μεταφέρεται στη σελίδα δημιουργίας λογαριασμού.
4. Εισαγωγή username, κωδικού πρόσβασης και email και ολοκλήρωση της λειτουργίας της εγγραφής επιλέγοντας το κουμπί «Submit»
5. Έπειτα η εταιρεία μεταφέρεται στο προφίλ για να ολοκληρώσει την εισαγωγή πληροφοριών.
6. Πατώντας στο κουμπί «Edit» μεταφέρεται στη σελίδα επεξεργασίας προφίλ.

<sup>1</sup> Σε κάθε περίπτωση ο χρήστης μπορεί να τερματίσει την εφαρμογή, κλείνοντας τη.

7. Η εταιρεία συμπληρώνει τα πεδία της φόρμας και ύστερα την υποβάλει.(Ολοκλήρωση δημιουργίας προφίλ και αποθήκευση του ολοκληρωμένου προφίλ στη βάση δεδομένων)
8. Στη συνέχεια μεταφέρεται πάλι στο προφίλ και πατώντας στο κουμπί «Create job» και μεταφέρετε στη σελίδα δημιουργίας θέσης εργασίας.
9. Συμπληρώνει τη αντίστοιχη φόρμα και πατάει «Submit».
10. Μεταφέρεται στη σελίδα της θέσης εργασίας που μόλις δημιούργησε.
11. Εκεί μπορεί να δει τους καταλληλότερους υποψήφιους και τα στοιχεία επικοινωνίας τους.

### 3.4.2.2. Εναλλακτική ροή

Σε περίπτωση που η εταιρεία δεν θέλει να ολοκληρώσει την λειτουργία της εγγραφής ή της σύνδεσης μπορεί να επιλέξει το κουμπί «Cancel».

Σε περίπτωση που η εταιρεία δεν επιθυμεί να ολοκληρώσει τη συμπλήρωση της φόρμας του προφίλ μπορεί να επιλέξει το κουμπί «Cancel» και να τερματίσει τη διαδικασία.

Σε περίπτωση που η εταιρεία επιθυμεί να επεξεργαστεί το προφίλ της, κάνει είσοδο στο σύστημα και έχει τη δυνατότητα επεξεργασίας της φόρμας του προφίλ πατώντας το κουμπί «Edit».

Σε περίπτωση που η εταιρεία επιθυμεί να διαγράψει το προφίλ της, κάνει είσοδο στο σύστημα και έχει τη δυνατότητα διαγραφής του λογαριασμού πατώντας το κουμπί «Delete».

Σε περίπτωση που η εταιρεία δεν επιθυμεί να συνεχίσει τις διαδικασίες επικαιροποίησης του προφίλ, μπορεί να τερματίσει τη διαδικασία πατώντας το κουμπί «Cancel».

Σε περίπτωση που η εταιρεία επιθυμεί να επεξεργαστεί μια θέση εργασίας, κάνει είσοδο στο σύστημα και έχει τη δυνατότητα επεξεργασίας της φόρμας της θέσης εργασίας πατώντας το κουμπί «Edit» δίπλα από την αντίστοιχη θέση.

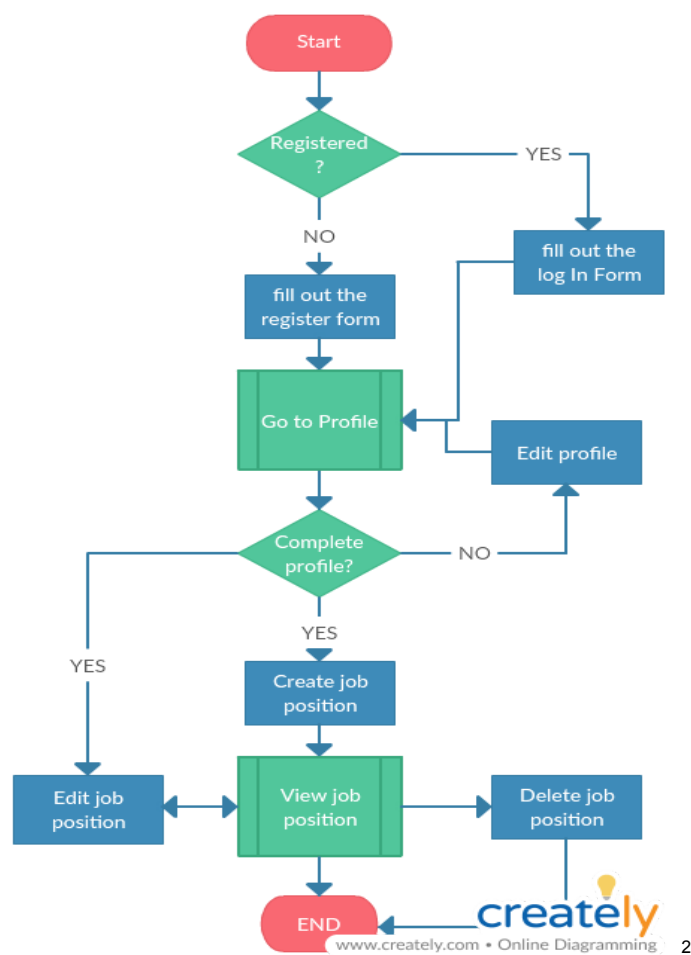
Σε περίπτωση που η εταιρεία επιθυμεί να διαγράψει μια θέση εργασίας, κάνει είσοδο στο σύστημα και έχει τη δυνατότητα διαγραφής του λογαριασμού πατώντας το κουμπί «Delete» δίπλα από τη κάθε θέση εργασίας.

Σε περίπτωση που η εταιρεία δεν επιθυμεί να συνεχίσει τις διαδικασίες επικαιροποίησης θέσης εργασίας, μπορεί να τερματίσει τη διαδικασία πατώντας το κουμπί «Cancel».

### 3.4.2.3. Μετά-συνθήκες

Όταν η εταιρεία κάνει εγγραφή στο σύστημα δημιουργείται μία καταχώρηση στη βάση δεδομένων. Αφού η εταιρεία ολοκληρώσει τη δημιουργία του προφίλ της, αυτό αποθηκεύεται στη βάση δεδομένων του συστήματος. Όταν η εταιρεία ολοκληρώσει τη δημιουργία μιας θέσης εργασίας, αυτή αποθηκεύεται στη βάση δεδομένων.

Από τα παραπάνω προκύπτει η εικόνα 3.7 .



Εικόνα 3.7 Διάγραμμα ροής εταιρείας.

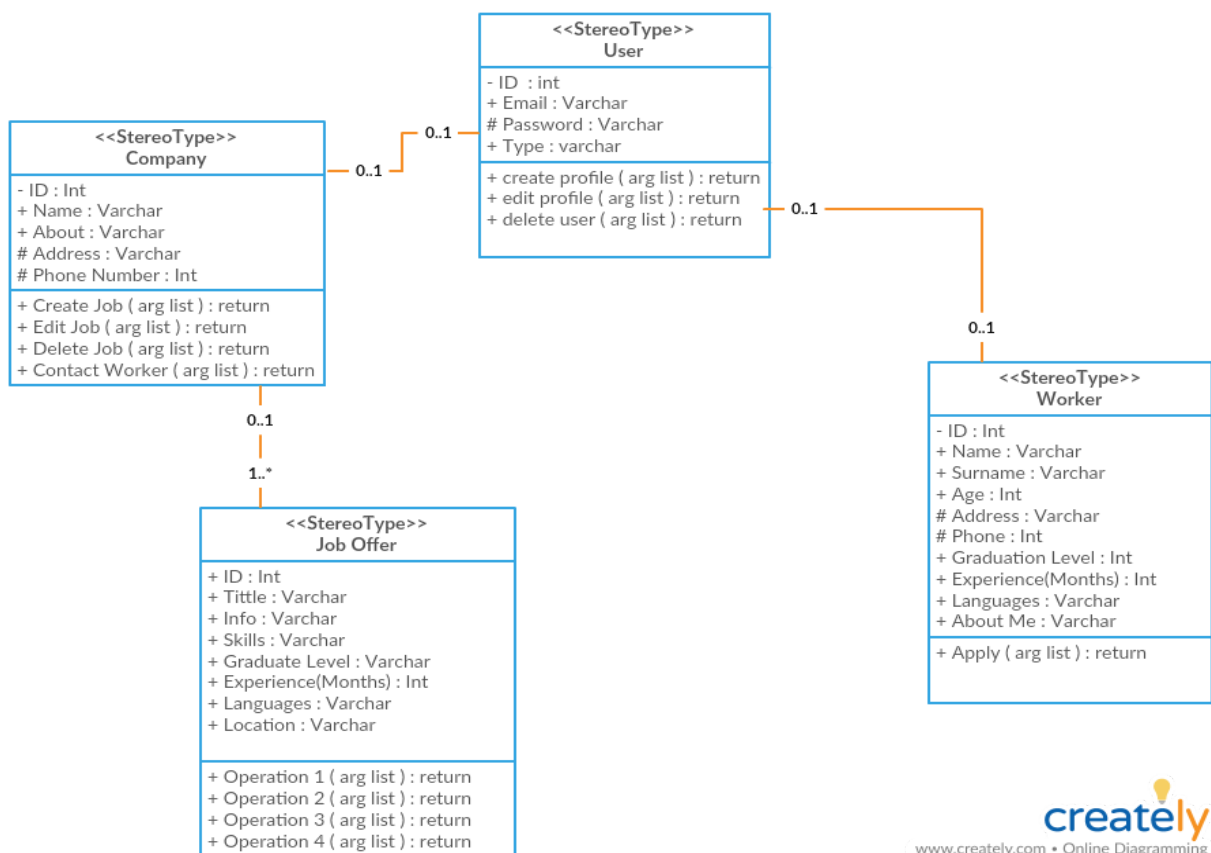
### 3.5. Διάγραμμα Κλάσεων

Για την εφαρμογή μας έχουμε δύο κύριες κλάσεις τη «User» και τη κλάση «Job Offer». Η κλάση «User» έχει δύο υποκλάσεις, την κλάση «Company» και την κλάση «Worker». Η κλάση «Worker» επειδή είναι υποκλάση της κλάσης «User» κληρονομεί τα attributes: id, email και password, και τις λειτουργίες: «Create profile», «Log in», «Edit profile», «Delete profile». Επιπλέον έχει τα attributes: id, name, surname, age, address, phone, graduation, experience, languages και about me. Αντίστοιχα η κλάση «Company» κληρονομεί πάλι τα στοιχεία της κλάσης «User» και έχει επιπλέον τα attributes: id, name, about, address, phone και τις λειτουργίες: «Create Job», «Edit Job», «Delete Job» και «Contact Worker». Η κλάση «Job Offer» έχει τα attributes: id, title, info, skills, graduate level, experience, languages, location.

Η κλάση «Company» συνδέεται με την κλάση «Job offer».

Ένα αντικείμενο της κλάσης «Company» μπορεί να συσχετιστεί με πολλά αντικείμενα της κλάσης «Job offer», ενώ ένα αντικείμενο της κλάσης «Worker» ή «Company» αντιστοιχεί μόνο σε ένα αντικείμενο της κλάσης «User». Επίσης η κλάσεις «Worker» μπορεί να συσχετιστεί με πολλά αντικείμενα της κλάσης «Job offer» και το αντίθετο.

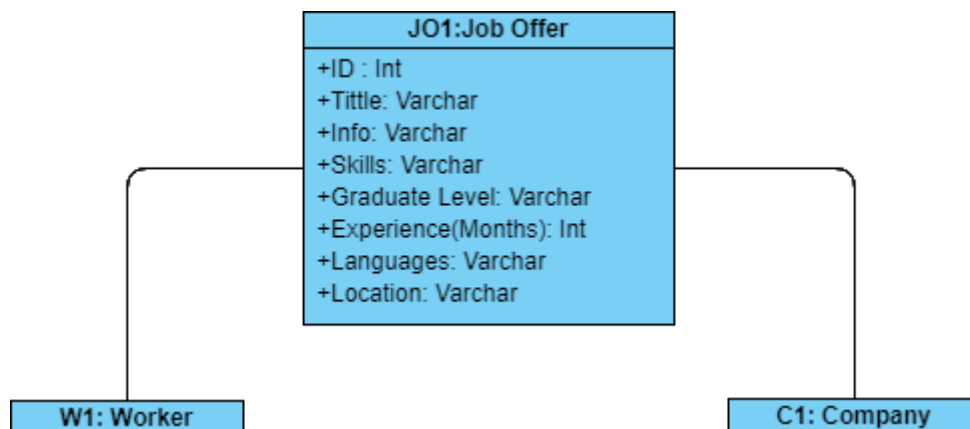
<sup>2</sup> Σε κάθε περίπτωση ο χρήστης μπορεί να τερματίσει την εφαρμογή, κλείνοντας τη.



Εικόνα 3.8 Διάγραμμα κλάσεων.

### 3.6. Διάγραμμα Αντικειμένων

Το αντικείμενο με όνομα «JO1», ανήκει στην κλάση «Job Offer» και αντιστοιχεί στο αντικείμενο της κλάσης «Company» με όνομα «C1». Επίσης σχετίζεται με το αντικείμενο της κλάσης «Worker» με όνομα «W1», γεγονός που δείχνει το αντικείμενο/ θέση εργασίας «JO1» πληροί τα κριτήρια της αναζήτησης που πραγματοποίησε το αντικείμενο «W1».

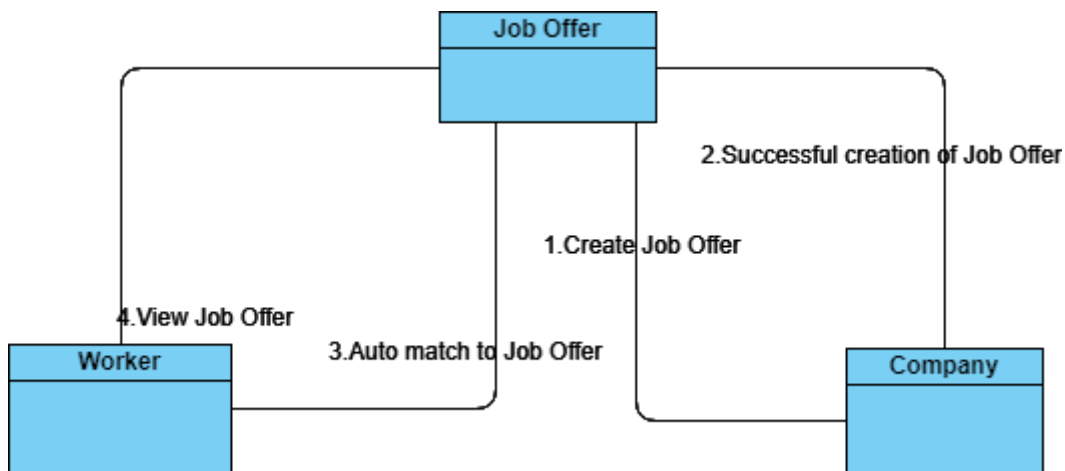


Εικόνα 3.9 Διάγραμμα αντικειμένων αγγελίας εργασίας.



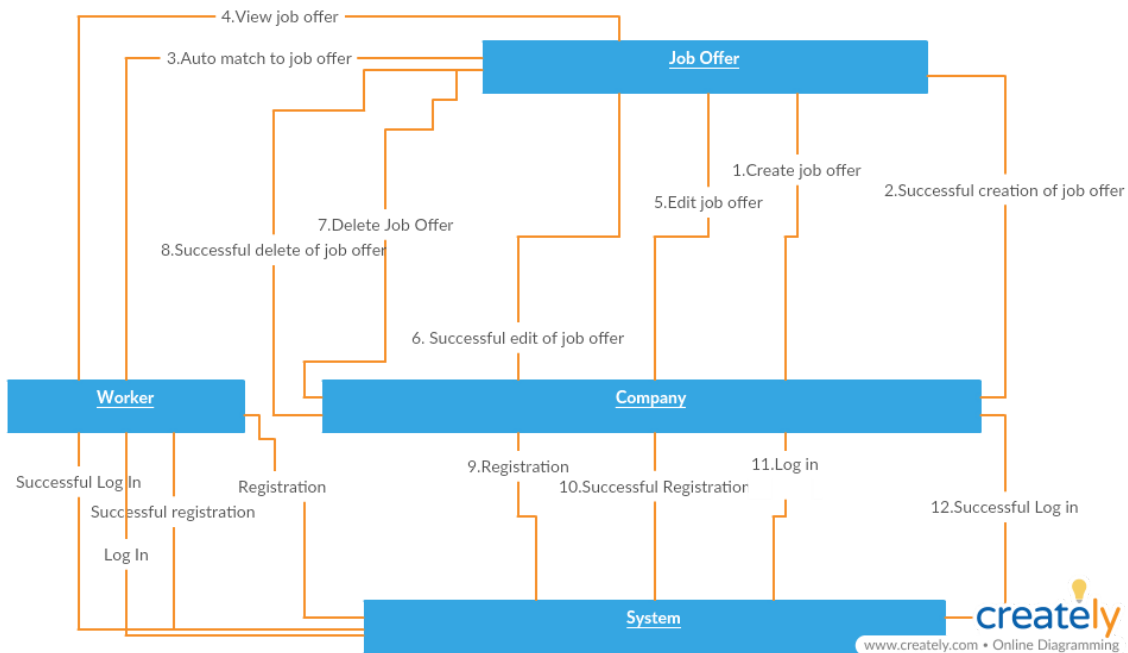
### 3.7. Διαγράμματα Συνεργασίας

Τα αντικείμενα της κλάσης «Company» συνδέονται με τα αντικείμενα της κλάσης «Job Offer», δηλαδή μια εταιρεία μπορεί να δημιουργήσει κάποια θέση εργασίας. Τα αντικείμενα της κλάσης «Job Offer» συνδέονται με τα αντικείμενα της κλάσης «Worker», δηλαδή οι θέσεις εργασίας διατίθενται για αυτόματη αντιστοίχιση από τους υποψήφιους εργαζόμενους.



Εικόνα 3.10 Διάγραμμα συνεργασίας.

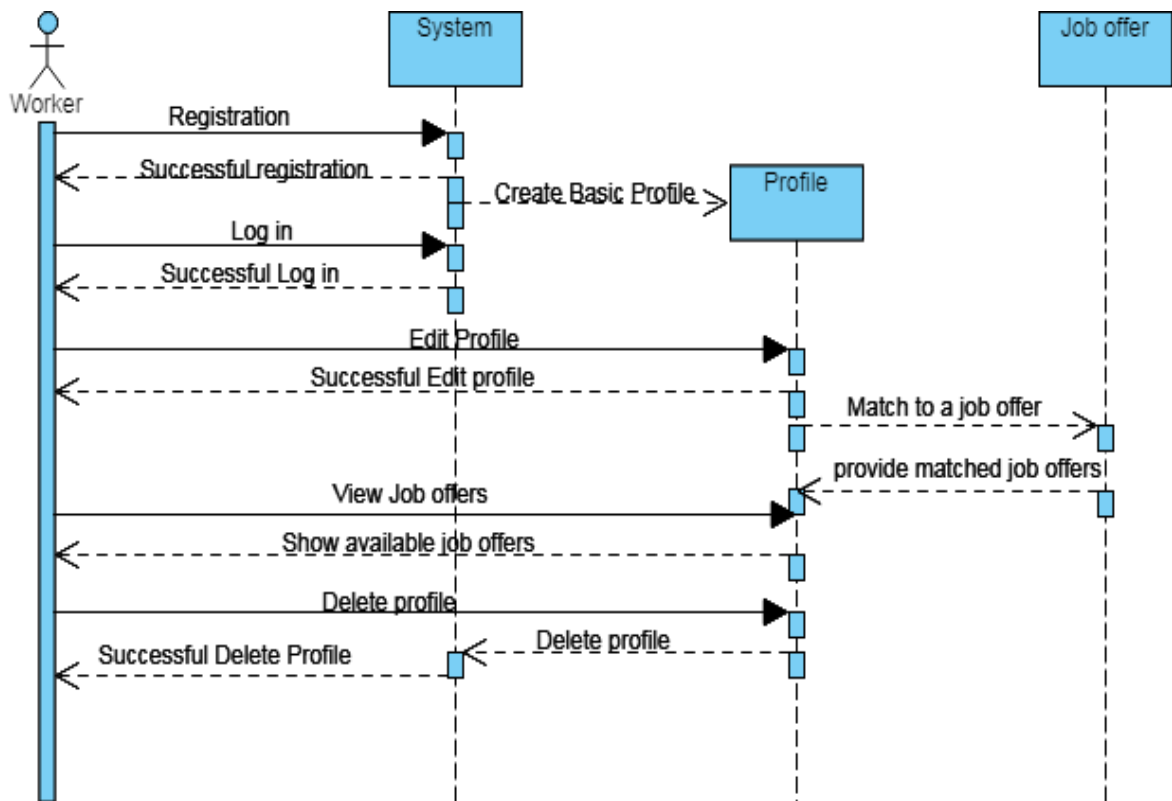
Σε δεύτερη ανάλυση προκύπτει ότι, τα αντικείμενα της κλάσης «Worker» και «Company» συνδέονται με το σύστημα για να εκτελέσουν τις λειτουργίες της εγγραφής και της εισόδου. Στη συνέχεια οι εταιρείες δημιουργούν τα αντικείμενα της κλάσης «Job Offer», μέσω της διαδικασίας δημιουργίας θέσης εργασίας. Σε δεύτερο στάδιο μπορούν να επεξεργαστούν ή να διαγράψουν τα αντικείμενα της κλάσης «Job Offer» που δημιούργησαν. Τέλος, τα αντικείμενα της κλάσης «Job Offer» διατίθενται για να αντιστοιχιστούν με τα αντικείμενα της κλάσης «Worker» μέσω της αυτοματοποιημένης λειτουργίας αντιστοίχισης. Οπότε προκύπτει η εικόνα 3.11 :



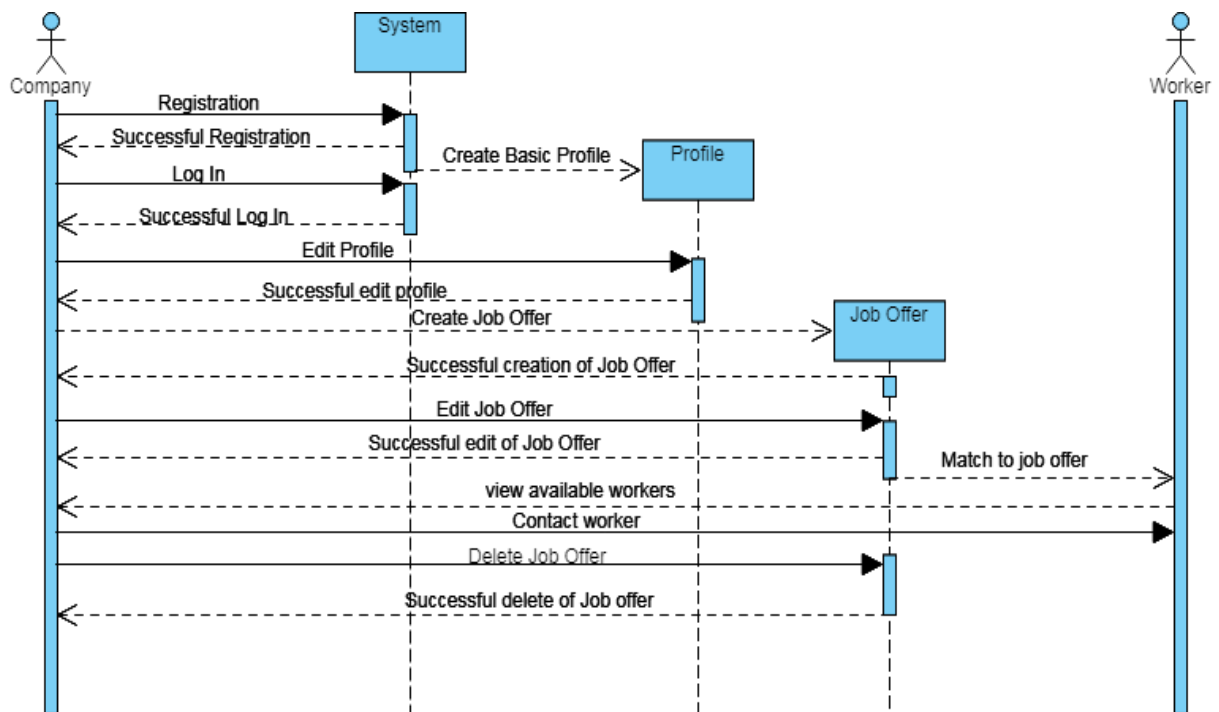
Εικόνα 3.11 2η έκδοση διαγράμματος συνεργασίας.

### 3.8. Διάγραμμα σειράς

Στο διάγραμμα σειράς, η κάθετη διάσταση, η οποία αντιστοιχεί στην κλίμακα του χρόνου μας δείχνει πως για να έχει τη δυνατότητα ο υποψήφιος εργαζόμενος να δημιουργήσει το τελικό προφίλ του και η εταιρεία να δημιουργήσει τη θέση εργασίας, πρέπει πρώτα να κάνει εγγραφή για να δημιουργηθεί ένα βασικό προφίλ και στη συνέχεια να κάνουν είσοδο στο σύστημα. Αφού προηγηθεί η λειτουργία της δημιουργίας του προφίλ από τον υποψήφιο εργαζόμενο και της θέσης εργασίας της εταιρείας, μπορούν να γίνουν οι λειτουργίες της επεξεργασίας και της διαγραφής της θέσης εργασίας και των προφίλ. Για να είναι δυνατή η λειτουργία της αντιστοίχισης εργαζομένων με τις θέσεις εργασίας, πρέπει να προηγηθεί η δημιουργία ολοκληρωμένων προφίλ από υποψήφιους εργαζόμενους και θέσεων εργασίας από τις εταιρείες.



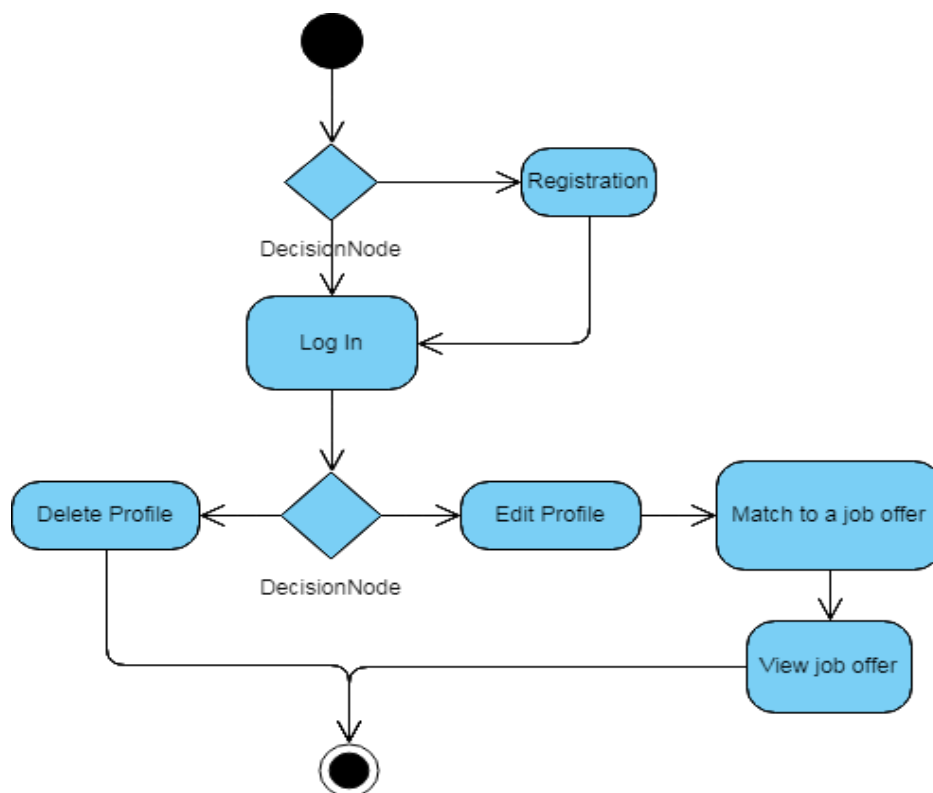
Εικόνα 3.12 Διάγραμμα ακολουθίας υποψήφιου εργαζομένου.



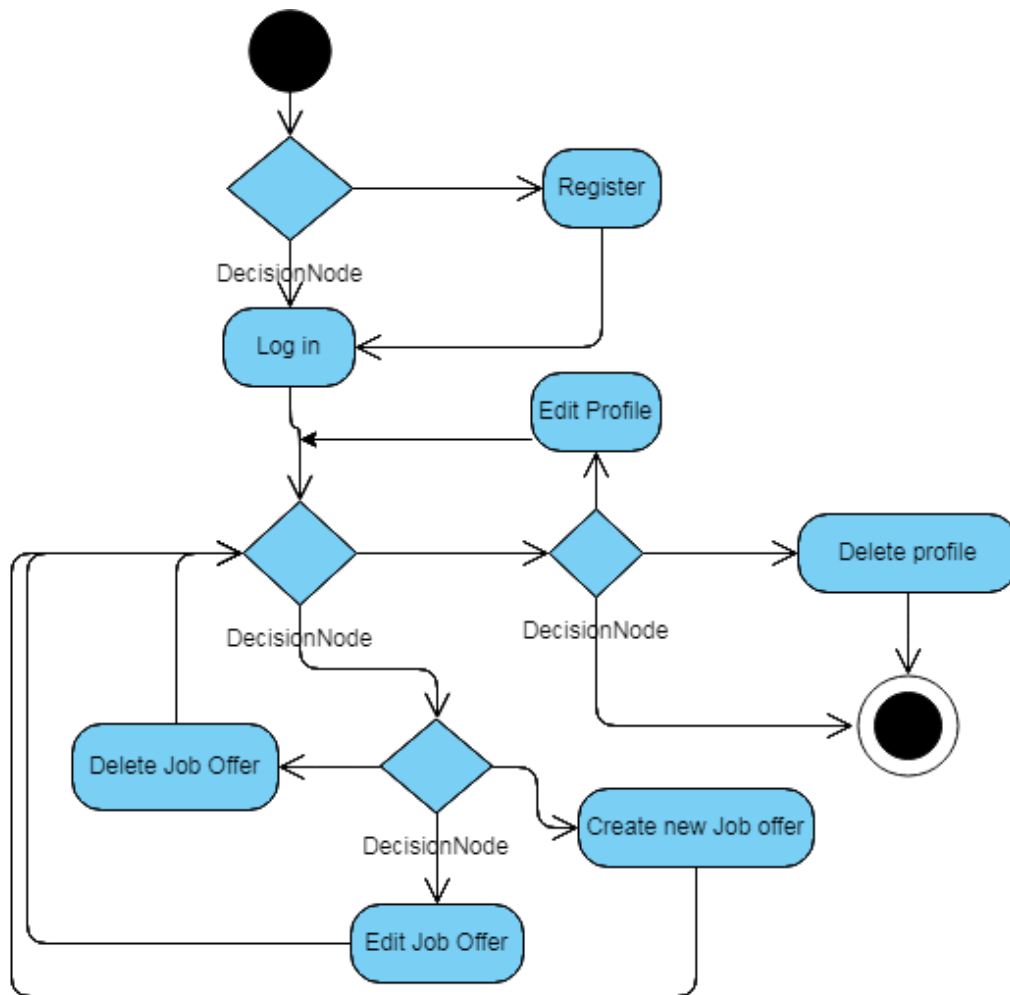
Εικόνα 3.12 Διάγραμμα ακολουθίας εταιρείας.

### 3.9. Διαγράμματα Δραστηριοτήτων

Το διάγραμμα δραστηριοτήτων μας δείχνει τις επιλογές που έχουν οι χρήστες στο σύστημα. Οι χρήστες υποψήφιοι εργαζόμενοι όταν εισέρχονται στην ιστοσελίδα μπορούν να εγγραφούν και στη συνέχεια να πραγματοποιήσουν είσοδο στο σύστημα και στη συνέχεια να επεξεργαστούν το προφίλ τους ώστε να μπορούν να αντιστοιχηθούν αυτόματα με κάποια θέση εργασίας ή να το διαγράψουν. Οι χρήστες εταιρείες μπορούν να εισέλθουν στην ιστοσελίδα, να εγγραφούν και στη συνέχεια να πραγματοποιήσουν είσοδο και να δημιουργήσουν κάποια θέση εργασίας και όταν βρεθεί κάποιος κατάλληλος υποψήφιος να επικοινωνήσουν μαζί του.



Εικόνα 3.13 Διάγραμμα δραστηριοτήτων υποψήφιου εργαζομένου.

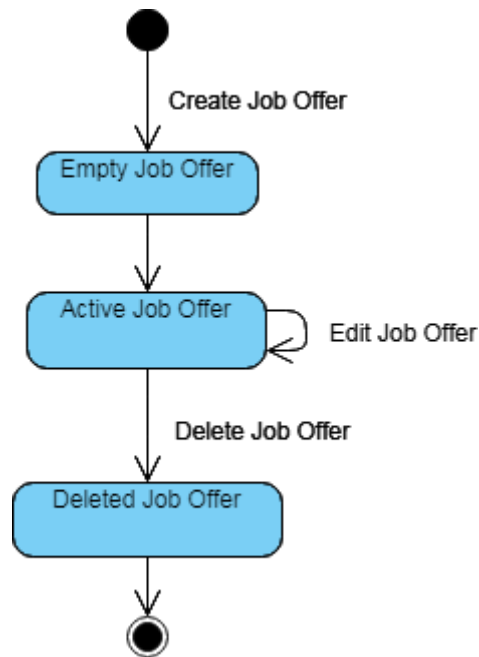


Εικόνα 3.14 Διάγραμμα δραστηριοτήτων εταιρείας.

## 3.10. Διαγράμματα Καταστάσεων

### 3.10.1. Διάγραμμα για τη κλάση Job Offer

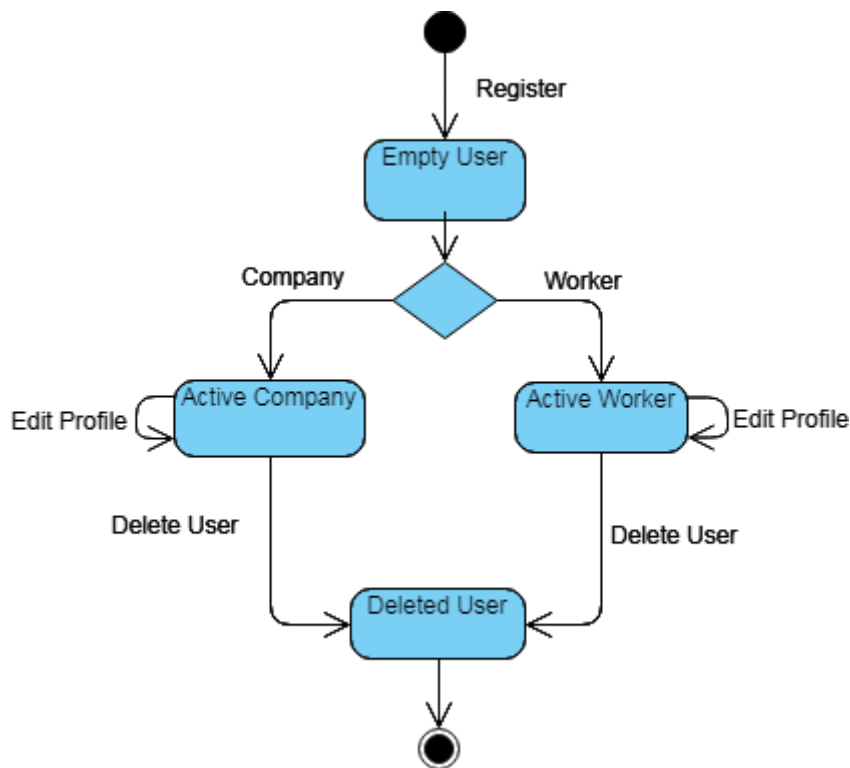
Το διάγραμμα καταστάσεων της εικόνας 3.15, απεικονίζει τις καταστάσεις που μπορεί να βρεθεί ένα αντικείμενο της κλάσης «Job offer». Αρχικά το αντικείμενο δημιουργείται και είναι «Empty Job offer», στη συνέχεια συμπληρώνεται από την εταιρεία και βρίσκεται στην κατάσταση «Active Job offer» και στη συνέχεια είναι ορατό και διατίθεται προς αυτόματη αντιστοίχιση. Αφού βρεθεί στην κατάσταση ενεργό μπορεί να επεξεργαστεί ανά πάσα στιγμή χωρίς να αλλάζει όμως κατάσταση. Στη συνέχεια αν πραγματοποιηθεί η λειτουργία της διαγραφής της θέσης εργασίας η θέση βρίσκεται στην κατάσταση «Deleted Job offer».



Εικόνα 3.15 Διάγραμμα καταστάσεων θέσης εργασίας.

### 3.10.2. Διάγραμμα για τις κλάσεις Worker και Company

Το διάγραμμα της εικόνας 3.16, απεικονίζει τις καταστάσεις που μπορεί να βρεθεί ένα αντικείμενο της κλάσης «User». Αρχικά το αντικείμενο δημιουργείται και είναι «Empty User», στη συνέχεια συμπληρώνεται από τον υποψήφιο εργαζόμενο ή εταιρεία και βρίσκεται στην κατάσταση «Active Worker» ή «Active Company» και στη συνέχεια είναι ορατό και διατίθεται προς αυτόματη αντιστοίχιση ο εργαζόμενος και το χρήστης τύπου εταιρεία μπορεί να δημιουργήσει μια θέση εργασίας . Αφού βρεθεί στην κατάσταση ενεργό μπορεί να επεξεργαστεί ανά πάσα στιγμή χωρίς να αλλάζει όμως κατάσταση. Στη συνέχεια αν πραγματοποιηθεί η λειτουργία της διαγραφής του χρήστη τότε μεταβαίνει στη κατάσταση «Deleted User».



Εικόνα 3.16 Διάγραμμα καταστάσεων χρήστη.

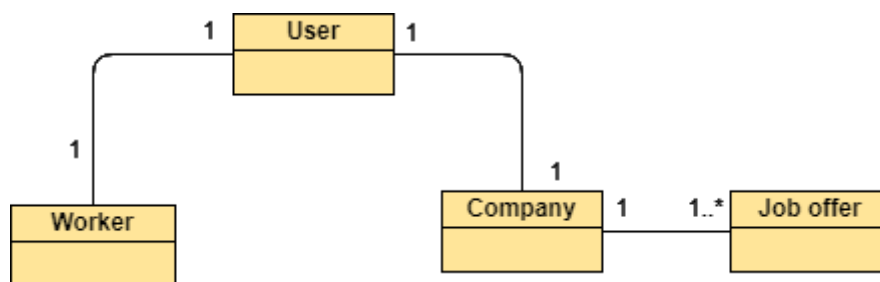
### 3.11. Ανάπτυξη Εφαρμογής

Η υλοποίηση και ανάπτυξη της παρούσας εφαρμογής θα γίνει σε περιβάλλον διαδικτύου και ως εκ τούτου θα δημιουργηθεί ένα web application. Οπότε καταλήγουμε ότι το σύστημα θα χρειάζεται να περιέχει στα συστατικά του: έναν Apache server, κάποια αρχεία κώδικα και μία Βάση Δεδομένων. Η διασύνδεση σε φυσικό επίπεδο προσδιορίζει πως ο server συνδέεται με τη βάση δεδομένων και τα αρχεία κώδικα ώστε να λειτουργεί το σύστημα. Για την οργάνωση των επεξεργαστικών πόρων του συστήματός έχουμε ότι, ο χρήστης χρησιμοποιεί σαν υπολογιστικό πόρο, ένα φυσικό αντικείμενο το οποίο είναι ένας υπολογιστής, ο οποίος επικοινωνεί με έναν server. Ο server αυτός είναι συνδεδεμένος με τη βάση δεδομένων που έχουμε δημιουργήσει.

## 4. Υλοποίηση και Έλεγχος

### 4.1. Βάση δεδομένων

Αρχική φάση της υλοποίησης του έργου, είναι η δημιουργία της βάσης δεδομένων. Από το δεύτερο κεφάλαιο Ανάλυση απαιτήσεων και σχεδίαση συστήματος και λογισμικού και το διάγραμμα κλάσεων<sup>3</sup> προκύπτει ότι θέλουμε τέσσερις πίνακες, ένας γενικός πίνακας για τους χρήστες (ονομασία: «usertab») ο οποίος περιέχει όλους τους χρήστες της εφαρμογής είτε είναι υποψήφιοι εργαζόμενοι, είτε είναι εταιρείες. Αυτός ο πίνακας θα συνδέεται με τους πίνακες εταιρειών (ονομασία: «comptab») και υποψήφιων εργαζόμενων (ονομασία: «worker») με σχέση ένα προς ένα, αυτό γιατί κάθε χρήστης μπορεί να είναι ένας εργαζόμενος ή μια εταιρεία. Επίσης κάθε εταιρεία θα μπορεί να δημιουργήσει μια ή παραπάνω θέσεις εργασίας οπότε έχουμε και το πίνακα θέσεων εργασίας (ονομασία: «joboffer») και η σχέση του με την εταιρεία είναι ένα προς πολλά. Οπότε προκύπτει η παρακάτω βάση σε ένα αρχικό στάδιο.



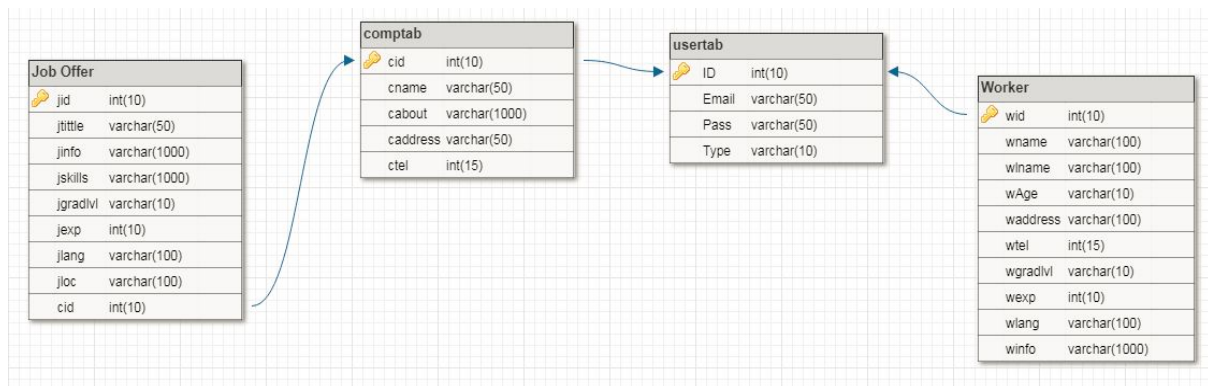
Εικόνα 4.1 1η έκδοση βάσης δεδομένων

Σε περαιτέρω ανάλυση της βάσης δεδομένων προκύπτουν τα πεδία και τα δευτερεύοντα κλειδιά και επομένως η ολοκληρωμένη βάση δεδομένων. Στο πίνακα usertab έχουμε τα πεδία ID που δρα ως πρωτεύον κλειδί, το 'Email', το 'Pass', για να δώσει ο οποιοσδήποτε χρήστης το email και έναν κωδικό ασφαλείας, και το 'Type', το πεδίο 'Type' θα χρησιμοποιηθεί για να ξεχωρίσει τους χρήστες σε εταιρείες και υποψήφιους εργαζόμενους. Ο πίνακας 'comptab' είναι ο πίνακας των εταιρειών και τα πεδία που τον αποτελούν είναι, το 'cid' που είναι το πρωτεύον κλειδί και ταυτόχρονα και το δευτερεύον, συνδέεται με το πρωτεύον κλειδί του πίνακα usertab γιατί έχουν σχέση ένα προς ένα, το πεδίο cname για την επωνυμία της εταιρείας, το 'cabout' για να παραθέσει η εταιρεία λίγα λόγια πχ με τι ασχολείται κάποιο βραβείο κλπ, 'caddress' και 'ctel' που χρησιμεύουν για την διεύθυνση και το τηλέφωνο αντίστοιχα. Ο πίνακας 'joboffer' έχει το πεδίο 'jid' ως πρωτεύον κλειδί και το πεδίο 'cid' ως δευτερεύον, τα κλειδιά αυτά είναι διαφορετικά σε αυτή τη περίπτωση γιατί κάθε εταιρεία μπορεί να δημιουργήσει μία ή πολλές θέσεις εργασίας, το πεδίο 'jtittle' για την επωνυμία της θέσης, το 'jinfo' θα περιέχει κάποιες πληροφορίες για τη θέση, το 'jskills' με κάποια σημαντικά χαρακτηριστικά που θα πρέπει να έχει ο εργαζόμενος για τη θέση αυτή, το 'jgradlvl' που θα έχει το επιθυμητό επίπεδο γνώσης του εργαζομένου, το 'jexp' που θα

<sup>3</sup> Βλέπε εικόνα 3.8



περιέχει το χρόνο προεργασίας του εργαζομένου σε μήνες, το 'jlang', τις γλώσσες που απαιτούνται για τη θέση αυτή, jloc που περιέχει τη διεύθυνση της θέσης εργασίας. Στη περίπτωση του πίνακα worker έχουμε ένα συνδυασμό των πινάκων compdata και 'joboffer' γιατί το προφίλ του εργαζομένου θα αποτελεί και το βιογραφικό του. Το πρωτεύον κλειδί είναι το 'wid' που είναι και το δευτερεύον και συσχετίζει το πίνακα worker με το πίνακα usertab με σχέση, ένα προς ένα.



Εικόνα 4.2 Ολοκληρωμένη βάση δεδομένων<sup>4</sup>

Η βάση θα δημιουργηθεί με τη χρήση του phpMyAdmin, εισάγοντας το κώδικα του παραρτήματος Ι.1 στη σελίδα 61, στην επιλογή SQL.

Αρχικά, όπως φαίνεται και στο παράρτημα Ι 1, φτιάχνουμε τους πίνακες με τα στοιχεία που επιθυμούμε να έχουν και στη συνέχεια θέτουμε τα πρωτεύοντα κλειδιά. Εξαιτίας κάποιου σφάλματος δε θα χρησιμοποιήσουμε δευτερεύοντα κλειδιά στη βάση, αλλά θα διαχειριζόμαστε τις εγγραφές και διαγραφές δεδομένων μέσω της php.

## 4.2. Σύνδεση βάσης με κώδικα

Για να συνδέσουμε τη βάση με το κώδικα θα χρησιμοποιήσουμε τη συνάρτηση `mysqli_connect()`, όπως φαίνεται και στο παράρτημα Ι 2 στη σελίδα 63. Αρχικά εισάγουμε τα στοιχεία της βάσης που μας έχουν δοθεί από το πάροχο του server στις κατάλληλες μεταβλητές για να δημιουργήσουμε τη σύνδεση, η συνάρτηση αυτή θα μπει στο αρχείο 'head.php' για να καλείτε στην αρχή της εφαρμογής.

<sup>4</sup> Τα μεγέθη των κελιών είναι εμπειρικά και θα μπορούσαν να δεχτούν αλλαγές στο στάδιο ελέγχου.

## 4.3. Η επικεφαλίδα, το στυλ και το υποσέλιδο

### 4.3.1. Η επικεφαλίδα (head.php)

Η επικεφαλίδα της εφαρμογής αποτελείται από ένα αρχείο, το οποίο καλείται σε κάθε σελίδα της. Περιέχει κάποιες απολύτως απαραίτητες συναρτήσεις, δηλαδή τη συνάρτηση έναρξης της συνεδρίας, το αρχείο που είναι υπεύθυνο για τη πραγματοποίηση της επικοινωνίας με τη βάση δεδομένων. Κάποια scripts, όπως το script που είναι υπεύθυνο για τη συμπερίληψη της γραμματοσειράς που χρησιμοποιούμε, το script για τη βιβλιοθήκη bootstrap και άλλα, όπως και το αρχικό μέρος του HTML μέχρι και το σημείο που εμφανίζεται το μενού της εφαρμογής.

Όπως φαίνεται και στο παράρτημα I 3 στη σελίδα 63, στις πρώτες γραμμές βλέπουμε τη συνάρτηση `'session_start()'` αυτή η συνάρτηση θα χρησιμοποιηθεί για το μηχανισμό εισόδου και εξόδου στη συνέχεια, και χρησιμοποιείται για να μας βοηθήσει να αλλάξουμε το μενού για κάποιον που είναι συνδεδεμένος στην εφαρμογή και για κάποιον που δεν είναι με τη χρήση του αρχείου `'auth.php'`. Στη συνέχεια καλούμε το αρχείο `'config.php'` που περιέχει τη συνάρτηση `'mysqli_connect()'`, συνδέουμε το αρχείο `'style.css'` στο οποίο θα γίνει η ωραιοποίηση της εφαρμογής, δηλαδή θα βάλουμε κάποια χρώματα, θα επιλέξουμε τις γραμματοσειρές και το μέγεθος τους, θα δημιουργηθούν κάποια κουμπιά και γενικότερα είναι το μέρος που θα φτιαχτεί το οπτικό τμήμα της εφαρμογής. Στα script tags που ακολουθούν εισάγουμε μια γραμματοσειρά της google, βιβλιοθήκη jquery για να μπορούμε να φτιάξουμε κάποια scripts που θα βοηθήσουν στις φόρμες μας, τη βιβλιοθήκη της fontawesome για να μπορούμε να χρησιμοποιήσουμε κάποια εικονίδια και τη bootstrap για να κάνουμε την εφαρμογή όσο γίνεται πιο φιλική για τις κινητές συσκευές. Στη συνέχεια ακολουθεί ένα script για να μπορούμε να κρύβουμε τις φόρμες όταν υποβάλλονται. Στη συνέχεια έχουμε το μενού της εφαρμογής και βλέπουμε ότι με τη χρήση της μεταβλητής `$_SESSION["email"]` και μιας `if{}`, εξετάζουμε αν έχει κάνει είσοδο ο χρήστης, για να του δείξουμε διαφορετικό μενού από το χρήστη που δεν έχει κάνει είσοδο. Παρατηρούμε ότι το αρχείο `'head.php'` δε κλείνει όλα αντικείμενα που ανοίγει, αυτό γίνεται επειδή τα αντικείμενα αυτά κλείνουν στα υπόλοιπα αρχεία που το χρησιμοποιούν, σε άλλη περίπτωση ο κώδικας παραπάνω δε θα δουλεύει σωστά και είχαμε πρόβλημα με τη μνήμη, αυτή η μέθοδος θα συνεχιστεί και σε άλλα αρχεία για λόγους ευχρηστίας και επειδή όλα τα αρχεία συσχετίζονται και το ένα είναι μέρος του άλλου.

#### 4.3.1.1. Ταυτοποίηση

Το αρχείο `'auth.php'` όπως φαίνεται στο παράρτημα I 4 στη σελίδα 65, εξετάζει αν ο χρήστης είναι εγγεγραμμένος ή όχι, αν δεν είναι τον μεταφέρει στην αρχική σελίδα της εφαρμογής.

### 4.3.2. Το αρχείο css (style.css)

Στο παράρτημα 5 στη σελίδα 65, βλέπουμε ένα κομμάτι του αρχείου 'style.css', στο οποίο αλλάζουμε λίγα πράγματα, προσθέτουμε αποστάσεις, θέτουμε τη γραμματοσειρά και τη στοίχιση του κειμένου και όλα τα υπόλοιπα γίνονται από τη bootstrap.

### 4.3.3. Υποσέλιδο (footer.php)

Στο υποσέλιδο έχουμε ένα jQuery script το οποίο χρησιμοποιούμε στις φόρμες μας για να εισάγει και να αφαιρεί πεδία, προσθέτουμε κάποιες βιβλιοθήκες που είναι απαραίτητες για να δουλέψει η bootstrap, τερματίζουμε τη σύνδεση μας με τη βάση δεδομένων και τέλος κλείνουμε όλα τα αντικείμενα που έχουν παραμείνει ανοιχτά από τα υπόλοιπα αρχεία. Ο κώδικας που περιέχει το αρχείο 'footer.php' υπάρχει στο παράρτημα I 6 στη σελίδα 67.

## 4.4. Αρχική Σελίδα

Στην αρχική σελίδα περιλαμβάνουμε κάποια λεκτικά που λέμε στο μελλοντικό χρήστη το λόγο που φτιάχτηκε η εφαρμογή και του παρουσιάζουμε με τις επιλογές να κάνει είσοδο ή δημιουργία νέου χρήστη, χρησιμοποιώντας απλή html και τη τεχνολογία 'flex-box' μέσω της bootstrap. Ο κώδικας του αρχείου της αρχικής σελίδας βρίσκεται στο παράρτημα I 7 στη σελίδα 68.

## 4.5. Μηχανισμός εισόδου/εξόδου (login/logout)

### 4.5.1. Είσοδος - login

Στο κώδικα του παραρτήματος I 8 σελίδα 69, βλέπουμε το μηχανισμό εισόδου, στον οποίο ευθύς αμέσως παρατηρούμε ότι περιέχει το 'head.php', αυτό γίνεται όπως θα δούμε στη συνέχεια γιατί χρειάζεται το αρχείο 'config.php' όπως οι περισσότερες σελίδες. Στη συνέχεια δημιουργούμε τη φόρμα για να μπορέσουμε να βάλουμε τα στοιχεία μας και ακολουθεί ο μηχανισμός επαλήθευσης αυτών, αρχικά χρησιμοποιούμε την συνάρτηση `stripslashes()` για να αποφύγουμε επιθέσεις (sql injections) στη βάση δεδομένων, η τεχνική αυτή χρησιμοποιείται σε όλες τις φόρμες της εφαρμογής μας, έπειτα εξετάζουμε αν κάποιο από τα πεδία μετά την υποβολή είναι άδειο και αν ισχύει τότε προβάλλουμε ένα μήνυμα στον χρήστη και του λέμε να ξανά προσπαθήσει, στη συνέχεια με τη `die(mysql_error())` εξετάζουμε το ενδεχόμενο αν υπάρχει ο χρήστης ή όχι και τέλος αν έχει περάσει όλα τα προηγούμενα στάδια κοιτάμε αν ο κωδικός και το 'email' αντιστοιχίζονται μεταξύ τους, αν περάσει όλους τους ελέγχους τότε ο χρήστης ανακατευθύνετε στη σελίδα του προφίλ του.

## 4.5.2. Έξοδος - logout

Στη περίπτωση της εξόδου ο χρήστης όταν πατήσει το κουμπί 'logout' ενεργοποιεί το κώδικα του παραρτήματος I 9 στη σελίδα 71, ο οποίος χρησιμοποιεί τη συνάρτηση `session_unset()`; για να τερματίσει το session που έχει δημιουργηθεί και στη συνέχεια γίνεται ανακατεύθυνση στη σελίδα 'login.php'.

## 4.5.3. Εγγραφή - Register

Βέβαια δε γίνεται να υπάρξει μηχανισμός εισόδου και εξόδου, χωρίς εγγραφή χρήστη. Και σε αυτή τη περίπτωση βλέπουμε από το παράρτημα I 10 σελίδα 71 για το κώδικα δημιουργίας χρήστη, ότι χρησιμοποιούμε μια φόρμα με τα απαραίτητα κελιά και ότι μετά την υποβολή της γίνεται η χρήση της `stripslashes()`; και της `mysqli_real_escape_string()`; για αποφυγή επιθέσεων στη βάση μας. Στη συνέχεια ακολουθούνται τα ίδια βήματα που ακολουθήσαμε και στη σελίδα της εισόδου, ελέγχουμε αν υπάρχουν κενά κελιά και αν υπάρχει το ίδιο 'email' ξανά στη βάση μας, βέβαια θα πρέπει σε δεύτερο χρόνο να εξεταστεί η περίπτωση που ένας υποψήφιος εργαζόμενος θελήσει να εισάγει μια θέση εργασίας, τέλος αν περάσει όλους του ελέγχους ο νέος χρήστης μπορεί να κάνει είσοδο στο προφίλ του.

## 4.5.4. Διαγραφή χρήστη

Όταν ο χρήστης θελήσει να διαγράψει το προφίλ του, τότε στη σελίδα του προφίλ του πατώντας την επιλογή διαγραφή, χρησιμοποιείται ο κώδικας που φαίνεται στο παράρτημα I 11 στη σελίδα 74. Χρησιμοποιώντας το 'id' του χρήστη που είναι μοναδικό, τον διαγράφουμε με τη χρήση της συνάρτησης `mysqli_multi_query()`; και της εντολής 'delete' της Mysql από όλους τους πίνακες της βάσης μας και αν είναι εταιρεία διαγράφουμε και τις θέσεις εργασίας που μπορεί να είχε καταχωρήσει, στη συνέχεια ανακατευθύνεται στη σελίδα εισόδου της εφαρμογής.

## 4.6. Σελίδα Προφίλ

Παρακάτω θα εξετάσουμε το κώδικα της σελίδας προφίλ του χρήστη. Αρχικά, στο παράρτημα I 12 στη σελίδα 74, έχουμε την εύρεση του χρήστη μέσω της σελίδας 'auth.php' και στη συνέχεια με τη χρήση του 'email' συλλέγουμε τα δεδομένα του από τη βάση δεδομένων με τη συνάρτηση `mysqli_query()`. Και προβάλουμε τα στοιχεία που είναι κοινά για τις εταιρείες και για τους εργαζόμενους. Επίσης γίνεται και ένας έλεγχος αν ο χρήστης που βλέπει τη σελίδα είναι ο ιδιοκτήτης του προφίλ ή απλά κάποιος επισκέπτης, για να του αποκρύψουμε στη συνέχεια κάποιες λειτουργίες όπως να επεξεργαστεί το προφίλ, που δεν πρέπει να έχει εξουσιοδότηση να εφαρμόσει. Στη συνέχεια διαχωρίζουμε το χρήστη αν είναι υποψήφιος εργαζόμενος ή εταιρεία.

### 4.6.1. Υποψήφιος Εργαζόμενος

Στη περίπτωση που ο χρήστης είναι υποψήφιος εργαζόμενος, τότε χρησιμοποιείται από την εφαρμογή ο κώδικας που βρίσκεται στο παράρτημα I 13 στη σελίδα 76, με τη χρήση του οποίου παίρνουμε το 'id' του υποψήφιου εργαζόμενου από το πίνακα 'usertab', κάνουμε αναζήτηση στο πίνακα 'worker' για το ολοκληρωμένο προφίλ του και αν αυτό δεν υπάρχει τότε στη σελίδα φαίνονται μόνο τα στοιχεία του από το πίνακα 'usertab'.

### 4.6.2. Εταιρεία

Στη περίπτωση που ο χρήστης είναι εταιρεία ακολουθείται η ίδια διαδικασία, όπως φαίνεται και στο παράρτημα I 14 στη σελίδα 77, με τη διαφορά ότι αυτή τη φορά χρησιμοποιώντας το id του από το πίνακα 'usertab' κάνουμε αναζήτηση στο πίνακα 'company' για το ολοκληρωμένο προφίλ, αν αυτό δεν υπάρχει τότε στη σελίδα φαίνονται μόνο τα στοιχεία από το πίνακα 'usertab'.

Στην εταιρεία επίσης εμφανίζονται και οι θέσεις εργασίας που έχει καταχωρήσει, δεδομένου του 'id' της εταιρείας γίνεται μια αναζήτηση στο πίνακα 'jobOffer' και από εκεί μπορεί να δει κάθε αγγελία ξεχωριστά πατώντας το κουμπί 'view'. Παρατηρούμε στο τέλος ότι ο χρήστης αν είναι ο ιδιοκτήτης του προφίλ μπορεί να βρει τις καλύτερες προσφορές εργασίας ή εργαζόμενους για αυτόν.

### 4.6.3. Φόρμα επικοινωνίας

Στο τέλος του προφίλ αν ο χρήστης είναι επισκέπτης τότε του προβάλεται ένα κουμπί 'contact' το οποίο όταν το πατήσει του αποκαλύπτει μια απλή φόρμα επικοινωνίας, για να στείλει ένα email ενδιαφέροντος στον χρήστη που ανήκει το προφίλ. Η φόρμα αυτή φαίνεται και στη σελίδα προβολής της θέσης εργασίας και απευθύνεται στην εταιρεία που ανήκει η θέση αυτή. Το παράρτημα I 15 στη σελίδα 79 περιλαμβάνει το κώδικα δημιουργίας και διαχείρισης της φόρμας.

### 4.6.4. Επεξεργασία προφίλ

Όπως φαίνεται στο παράρτημα I 16 σελίδα 82, αρχικά συλλέγουμε το 'id' του χρήστη από το 'url' χρησιμοποιώντας τη συνάρτηση `$_GET['id']`, το 'id' το δώσαμε στο 'url' μέσω του κουμπιού 'edit' στο προφίλ του χρήστη. Στη συνέχεια ψάχνουμε το πίνακα 'usertab' για τις βασικές πληροφορίες του και τις αποθηκεύουμε στις αντίστοιχες μεταβλητές.

Στη συνέχεια με τη χρήση της μεταβλητής `$type` ελέγχουμε αν ο χρήστης είναι υποψήφιος εργαζόμενος ή εταιρεία.

#### 4.6.4.1. Συλλογή δεδομένων

##### 4.6.4.1.1. Υποψήφιος εργαζόμενος

Στο παράρτημα I 17 σελίδα 82, βλέπουμε ότι, αν ο χρήστης είναι υποψήφιος εργαζόμενος τότε ψάχνουμε στο πίνακα 'worker' για τα λοιπά στοιχεία του, αν υπάρχουν τότε τα συλλέγουμε και αυτά σε απλές μεταβλητές με εξαίρεση των 'skills' και 'languages' τα οποία με τη χρήση της συνάρτησης `explode("*", $string);` ,στη συνάρτηση `explode("*", $string)` ο αστερίσκος παίζει το ρόλο του διαχωριστή ανάμεσα στα στοιχεία και θα το δούμε πως μπαίνει στο πίνακα της βάσης με τη χρήση της `explode()` στη συνέχεια. Αν δεν υπάρχουν του δημιουργούμε μια κενή εγγραφή, εισάγοντας το αντίστοιχο id από το πίνακα 'usertab'.

##### 4.6.4.1.2. Εταιρεία

Στη συνέχεια όπως φαίνεται και από το κώδικα του παραρτήματος I 18 στη σελίδα 83, ακολουθούμε την ίδια τακτική αν ο χρήστης είναι εταιρεία, παρατηρούμε ότι στη εταιρεία δεν έχουμε τόσα στοιχεία όσα στον υποψήφιο εργαζόμενο, διότι το προφίλ του εργαζομένου έχει και την ιδιότητα του βιογραφικού του.

#### 4.6.4.2. Προβολή δεδομένων - Φόρμα

Στο παράρτημα I 19 στη σελίδα 84 φαίνεται ο τρόπος που παρουσιάζονται τα στοιχεία που θα φαίνονται ανεξαρτήτως τύπου χρήστη. Παρατηρούμε ότι είναι μια απλή φόρμα η οποία κατά την υποβολή της χρησιμοποιεί τη διαδικασία 'POST' για να αποθηκεύσει τα δεδομένα στη βάση δεδομένων.

##### 4.6.4.2.1. Υποψήφιοι εργαζόμενοι

Στο παράρτημα I 20 στη σελίδα 85, αρχικά βλέπουμε ότι μέσω μιας συνάρτησης 'if' ο κώδικας διαχωρίζεται σε δύο μέρη, το ένα για τον υποψήφιο εργαζόμενο και το άλλο για την εταιρεία, επίσης φαίνεται και η συνέχεια της φόρμας αν ο χρήστης είναι υποψήφιος εργαζόμενος.

Παρατηρούμε για το πεδίο των γλωσσών και τον ικανοτήτων χρησιμοποιούνται οι πίνακες που δημιουργήσαμε με τη συνάρτηση `explode()`, και για τη καλύτερη προβολή και συμπλήρωση τους θα χρησιμοποιήσουμε το jQuery script το οποίο έχει υλοποιηθεί στο αρχείο 'footer.php'.

##### 4.6.4.2.2. Εταιρεία

Για τις εταιρείες ισχύει το ίδιο χωρίς τους πίνακες και το script. Όπως φαίνεται στο παράρτημα I 21 στη σελίδα 87. Στο τέλος παρατηρούμε ότι υπάρχει ένα 'input' πεδίο το

οποίο έχει τύπο 'hidden', αυτό χρησιμοποιείται για να κρατάμε το 'id' του χρήστη αφότου υποβάλει τη φόρμα.

#### 4.6.4.3. Υποβολή στοιχείων στη βάση δεδομένων

Στο κώδικα του παραρτήματος I 22 στη σελίδα 88 βλέπουμε τα βήματα που ακολουθούνται όταν ο χρήστης υποβάλει τη φόρμα επεξεργασίας προφίλ. Παρατηρούμε ότι χρησιμοποιούμε συναρτήσεις όπως η `stripslashes($_POST['email']);` αυτό γίνεται για να αποφύγουμε τυχόν επιθέσεις στη βάση δεδομένων και με τη χρήση της συνάρτησης `mysqli_real_escape_string($mysqli,$email)` δημιουργούμε ένα SQL string που να μπορεί να χρησιμοποιηθεί σε μια SQL πρόταση. Έπειτα κοιτάμε για τυχόν κενά κελιά και αν υπάρχουν τότε έχουμε μήνυμα λάθους για το χρήστη και στο τέλος εισάγουμε τα κοινά στοιχεία που ισχύουν για όλους τους χρήστες. Δεδομένου ότι σε αυτό το κομμάτι του κώδικα επεξεργαζόμαστε τα κοινά στοιχεία των χρηστών, αυτό γίνεται για να αποφύγουμε επανάληψη του ίδιου κώδικα, δηλαδή για να μπορέσουμε να χρησιμοποιήσουμε τις λιγότερες δυνατές υπολογιστικές δυνάμεις που μας είναι εφικτό.

##### 4.6.4.3.1. Υποψήφιοι εργαζόμενοι

Όπως και στη περίπτωση της προβολής της φόρμας κάναμε το διαχωρισμό αν ο χρήστης είναι εργαζόμενος είτε εταιρεία, έτσι και εδώ κάνουμε με τις ίδιες διαδικασίες δηλαδή με τη χρήση μιας 'if', το διαχωρισμό του χρήστη σε εταιρεία ή υποψήφιο εργαζόμενος. Στο κώδικα του παραρτήματος I 23 στη σελίδα 89 έχουμε τη περίπτωση ότι ο χρήστης είναι υποψήφιος εργαζόμενος, και ακολουθείται η παρακάτω διαδικασία, ακολουθούμε την ίδια διαδικασία όπως με τα κοινά στοιχεία και χρησιμοποιούμε τις συναρτήσεις `stripslashes($_POST['wfname'])` και `mysqli_real_escape_string($mysqli,$wfname)`, με τη διαφορά ότι επειδή πρέπει να εισάγουμε και δύο πίνακες χρησιμοποιούμε τη συνάρτηση `implode(" ", $skill);` για να μετατρέψουμε το πίνακα που έχουμε δημιουργήσει στο πεδίο για τις ικανότητες και τις γλώσσες σε 'string' με τον αστερίσκο(\*) να διαχωρίζει τα στοιχεία και να μας διευκολύνει όταν χρησιμοποιούμε τη συνάρτηση `explode()`, στη συνέχεια περνάμε τα δεδομένα στη βάση δεδομένων και προβάλλουμε μήνυμα επιτυχίας αν έχουν περάσει με επιτυχία και τα γενικά και τα εξειδικευμένα δεδομένα, δηλαδή αν αληθεύει και το `$resultwork` και το `$result` ή αν έχουμε κάποιο σφάλμα προβάλλουμε ένα μήνυμα σφάλματος, στο τέλος κάθε if χρησιμοποιούμε τη συνάρτηση `removeJunk(element)` δίνοντας τη 'class' του 'div' της φόρμας που υλοποιείται στο 'header.php' για να μην υπάρξει κάποιο λογικό σφάλμα με την υποβολή της φόρμας.

##### 4.6.4.3.2. Εταιρεία

Όπως βλέπουμε στο παράρτημα I 24 στη σελίδα 91, για την περίπτωση που ο χρήστης είναι εταιρεία ισχύουν τα ίδια όπως και στους εργαζόμενους και σε σημεία ο κώδικας είναι πιο απλός διότι δεν έχει τον ίδιο φόρτο δεδομένων.



## 4.7. Θέση εργασίας

### 4.7.1. Δημιουργία νέας αγγελίας

Στο παράρτημα I 25 σελίδα 93 έχουμε το κώδικα για τη φόρμα που πρέπει να συμπληρωθεί για τη δημιουργία μιας νέας αγγελίας, παρατηρούμε ότι χρησιμοποιούμε παρόμοιο τρόπο με τη δημιουργία νέου χρήστη. Αρχικά από το url της σελίδας κρατάμε το id της εταιρείας, το οποίο έχουμε δώσει στο url από τη σελίδα του προφίλ με αυτό το τρόπο: `<?php echo "<td>". '<a href= \"createjob.php?id=$id\"> New Job </a>'."</td>"; ?>`.

Για τα πεδία των ικανοτήτων και των γλωσσών, χρησιμοποιείται ο ίδιος τρόπος που χρησιμοποιήθηκε και για την επεξεργασία του προφίλ του υποψήφιου εργαζομένου. Στη συνέχεια δημιουργούμε τις μεταβλητές που θα εισάγουμε στη βάση δεδομένων, τις εισάγουμε και προτείνουμε στο χρήστη(εταιρεία) να πάει πίσω στο προφίλ του για να δει τις αγγελίες του, αν υπάρξει κάποιο σφάλμα του το αναφέρουμε για να ξαναπροσπαθήσει. Στο τέλος υπάρχει το script για τα πεδία των γλωσσών και των ικανοτήτων και το footer της σελίδας, Όπως και το script για την απόκρυψη της φόρμας.

### 4.7.2. Προβολή αγγελίας εργασίας

Για τη προβολή της θέσης εργασίας θα εφαρμόσουμε, όπως φαίνεται και στο παράρτημα I 26 στη σελίδα 97, παρόμοιες λειτουργίες με το προφίλ του χρήστη, τις οποίες θα αλλάξουμε για να ικανοποιούν τις απαιτήσεις μας. Αρχικά συλλέγουμε και εδώ τα απαραίτητα στοιχεία, στη προκειμένη περίπτωση τα 'id' του χρήστη και της θέσης εργασίας, τα οποία έχουμε δώσει στο 'url' της σελίδας όταν ανακατευθύνετε ο χρήστης. Και "τραβάμε" από τη βάση δεδομένων τα στοιχεία του χρήστη για να τον διαχωρίσουμε σε εταιρεία ή εργαζόμενο.

Αφού συλλέξουμε τα στοιχεία του χρήστη προβάλλουμε τα στοιχεία της θέσης εργασίας και μετά τα στοιχεία της εταιρείας που τη δημιούργησε, επειδή το 'email' της εταιρείας βρίσκεται στο πίνακα 'usertab' και όχι στο πίνακα 'comptab' πρέπει χρησιμοποιώντας το 'id' της εταιρείας να κάνουμε μια αναζήτηση στο πίνακα 'usertab' για το 'email'.

Στο τέλος ελέγχουμε αν ο χρήστης είναι η εταιρεία που τη δημιούργησε, τότε μπορεί να την επεξεργαστεί και να τη διαγράψει και αν δεν είναι ο δημιουργός της θέσης αυτής τότε μπορεί να επικοινωνήσει και να δείξει ενδιαφέρων για τη θέση εργασίας, με τη φόρμα επικοινωνίας όπως και στο προφίλ.

### 4.7.3. Επεξεργασία αγγελίας εργασίας

Στο παράρτημα I 27 στη σελίδα 102 βλέπουμε τη, σελίδα επεξεργασίας της θέσης εργασίας όπου δρούμε όπως και στη σελίδα επεξεργασίας του προφίλ, αρχικά από το 'url' της σελίδας παίρνουμε το 'id' της θέσης εργασίας και το χρησιμοποιούμε για να συλλέξουμε όποια δεδομένα υπάρχουν στη βάση δεδομένων για τη θέση αυτή, στη συνέχεια θέτουμε μεταβλητές στα δεδομένα αυτά για ευκολία χρήσης. Παρατηρούμε ότι για τα πεδία των



ικανοτήτων και των γλωσσών, χρησιμοποιείται ο ίδιος τρόπος με το προφίλ του εργαζόμενου.

Ακολουθεί η δημιουργία της φόρμας, με τη προβολή των συμπληρωμένων πεδίων, στα πεδία χρησιμοποιούμε την επιλογή 'required' για να δείξουμε αυτή τη φορά με τη χρήση της HTML και όχι της php όπως στην σελίδα επεξεργασίας του προφίλ του, ότι αυτά τα πεδία είναι υποχρεωτικά, στη περίπτωση που διαγράψει το περιεχόμενο τους και δε συμπληρώσει κάτι νέο.

Όταν ο χρήστης κάνει υποβολή της φόρμας πατώντας το κουμπί "update" τότε συλλέγονται τα στοιχεία από τη φόρμα και ετοιμάζονται, για να τα εισάγουμε στο πίνακα της βάσης δεδομένων ακολουθώντας τις ίδιες διαδικασίες με τη φόρμα επεξεργασίας προφίλ του χρήστη.

Αν η εισαγωγή των δεδομένων είναι επιτυχής ο χρήστης βλέπει ένα μήνυμα επιτυχίας και μπορεί να δει την επεξεργασμένη θέση εργασίας, αλλιώς του εμφανίζεται ένα μήνυμα σφάλματος. Στο τέλος του αρχείου έχουμε το footer και το script που καλεί τη συνάρτηση για την εισαγωγή των επιπλέον στοιχείων των πεδίων των ικανοτήτων και των γλωσσών και το script που καλεί τη συνάρτηση που είναι υπεύθυνη για την απόκρυψη της φόρμας μετά την υποβολή της.

## 4.8. Αντιστοίχιση υποψήφιων εργαζομένων και θέσεων εργασίας

Από το παράρτημα I 28 στη σελίδα 107 που περιέχει το κώδικα για μηχανισμό αντιστοίχισης των υποψηφίων εργαζομένων με αγγελίες, βλέπουμε ότι ο χρήστης πατώντας τα κουμπιά προβολής εργαζομένων ή θέσεων εργασίας από το προφίλ του η τη σελίδα της αγγελίας εργασίας, όπως για παράδειγμα αυτό: `<a class="btn costbtn btn-outline-primary" href="automach.php?id=<?php echo $id; ?>">Show Me Job Offers</a>`, μεταβαίνει στη σελίδα εύρεσης εργασίας που μας δείχνει τους καταλληλότερους εργαζόμενους για την αγγελία μας ή τη καλύτερη εργασία για το βιογραφικό μας.

Αρχικά συλλέγουμε από το URL το id του χρήστη ή της αγγελίας που παρέχεται και ορίζουμε ένα μετρητή που θα κρατάει τους πόντους κάθε εργαζόμενου για μια αγγελία, ή κάθε αγγελίας για ένα εργαζόμενο.

Στη συνέχεια ελέγχουμε αν έχουμε το 'jid' ή το 'id'. Αν έχουμε το 'jid' τότε ψάχνουμε τη συγκεκριμένη αγγελία και αποθηκεύουμε τα στοιχεία της σε μεταβλητές. Μετά συλλέγουμε τα στοιχεία όλων των υποψήφιων εργαζομένων και τα τοποθετούμε σε πίνακες, προχωράμε στη σύγκριση των μεταβλητών της αγγελίας με τα στοιχεία όλων των υποψηφίων εργαζομένων χρησιμοποιώντας τη συνάρτηση `in_array($needle, array $haystack[, bool $strict = FALSE])` η οποία δέχεται μια μεταβλητή η οποία μπορεί να είναι και πίνακας και ένα σύνολο τιμών, τα συγκρίνει και επιστρέφει 'true' αν βρει κάποια ίδια τιμή. Τοποθετώντας αυτή τη συνάρτηση σε μια επανάληψη `foreach()` για κάθε στοιχείο των

υποψηφίων εργαζομένων, μπορούμε να τα συγκρίνουμε με τα στοιχεία της μιας συγκεκριμένης αγγελίας εργασίας και μετράμε πόσες φορές επιστρέφει 'true' αυξάνοντας το πίνακα `$points[]`, όταν ολοκληρωθούν όλες οι συγκρίσεις κατανέμουμε τους πίνακες με τα στοιχεία των εργαζομένων ως προς το πίνακα `$points[]` σε φθίνουσα σειρά με τη χρήση της συνάρτησης `array_multisort($keyTable, SORT_DESC, $table1, $table2, ..., $tableN)`;, προβάλλουμε τα αποτελέσματα και παρέχουμε έναν υπερσύνδεσμο για να μπορέσει να επισκεφτεί ο χρήστης στη προκειμένη περίπτωση τη σελίδα της αγγελίας. Η ίδια διαδικασία ακολουθείται και στη περίπτωση που μας έχει δοθεί 'id' χρήστη και είναι υποψήφιος εργαζόμενος, με τη μόνη διαφορά ότι τώρα ψάχνουμε για τον υποψήφιο ποιό είναι η καταλληλή αγγελία, οπότε σε μεταβλητές εισάγονται τα στοιχεία του εργαζόμενου και σε πίνακα τα στοιχεία όλων των διαθέσιμων αγγελιών και η σύγκριση γίνεται ως προς τον εργαζόμενο και όχι ως προς την αγγελία.

Στη περίπτωση που μια εταιρεία έχει πατήσει από το προφίλ της να δει τους καλύτερους εργαζόμενους για τις αγγελίες της, τότε ακολουθείτε πάλι η ίδια διαδικασία με τη διαφορά ότι τώρα συλλέγουμε τα δεδομένα από όλες τις αγγελίες της, τα τοποθετούμε σε πίνακες και μετά συλλέγουμε τα στοιχεία όλων των εργαζομένων που υπάρχουν στο σύστημα μας και τα τοποθετούμε και αυτά σε πίνακες. Έπειτα με τη χρήση μιας διπλής επανάληψης συγκρίνουμε για κάθε αγγελία όλους τους εργαζομένους και παρουσιάζουμε τα αποτελέσματα, με το πρώτο εργαζόμενο να είναι αυτός που η εταιρεία θα μπορέσει να χρησιμοποιήσει καλύτερα.

Στο τέλος απελευθερώνουμε όλες τις μεταβλητές που χρησιμοποιήθηκαν για την mysql για αποφυγή κάποιου σφάλματος.

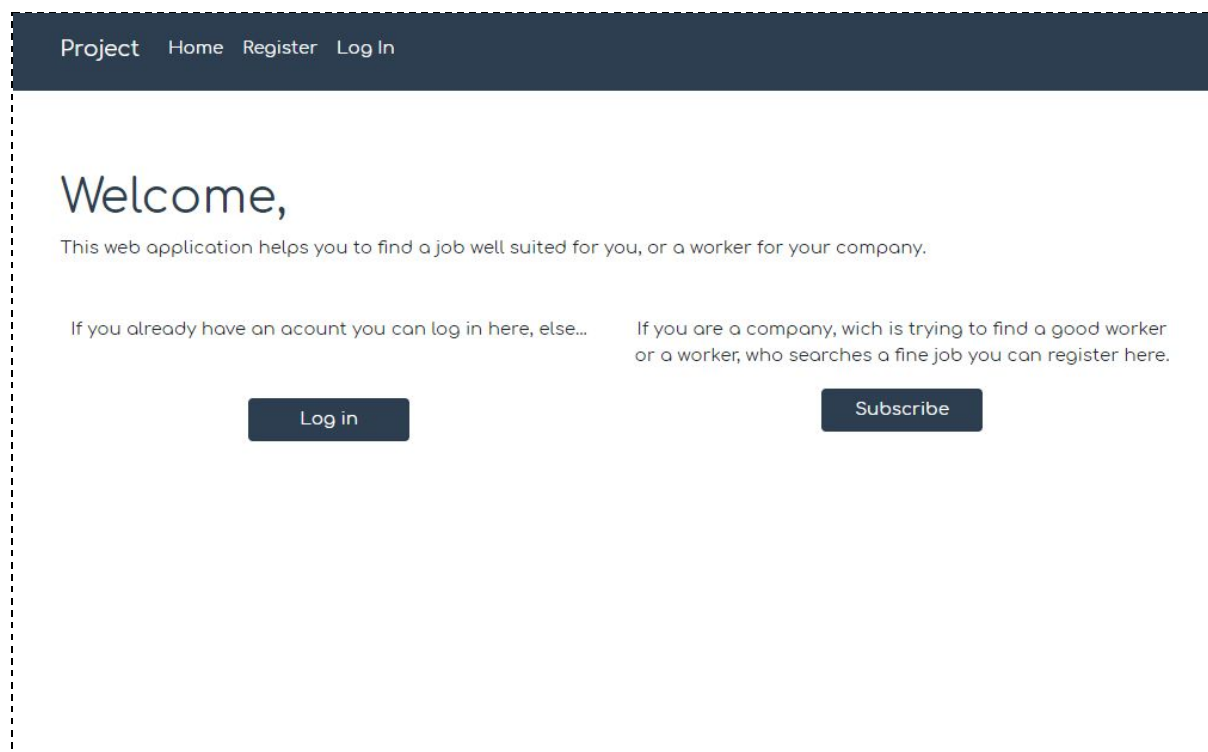
## 5. Λειτουργία Συστήματος / Οδηγός Χρήσης

Η εφαρμογή μας έχει ως σκοπό την δημιουργία προσωποποιημένων προφίλ από υποψήφιους εργαζόμενους και από εταιρείες και τη δημιουργία αγγελιών θέσεων εργασίας από τις εταιρείες. Οι χρήστες που εξυπηρετεί η εφαρμογή μας είναι οι υποψήφιοι εργαζόμενοι και επιχειρήσεις.

Επιπρόσθετα, οι οντότητες που καλούνται να διαχειριστούν οι χρήστες είναι αυτή του προσωποποιημένου προφίλ τους και των αγγελιών εργασίας. Ειδικότερα, το προφίλ δημιουργείται από τους υποψήφιους εργαζόμενους και τις επιχειρήσεις, ενώ τις αγγελίες εργασίας τις δημιουργεί κάποια επιχείρηση.

Στις επόμενες υποενότητες, καταγράφονται ένα προς ένα τα βασικά βήματα που εκτελεί ο εκάστοτε χρήστης προκειμένου να χρησιμοποιήσει την εφαρμογή.

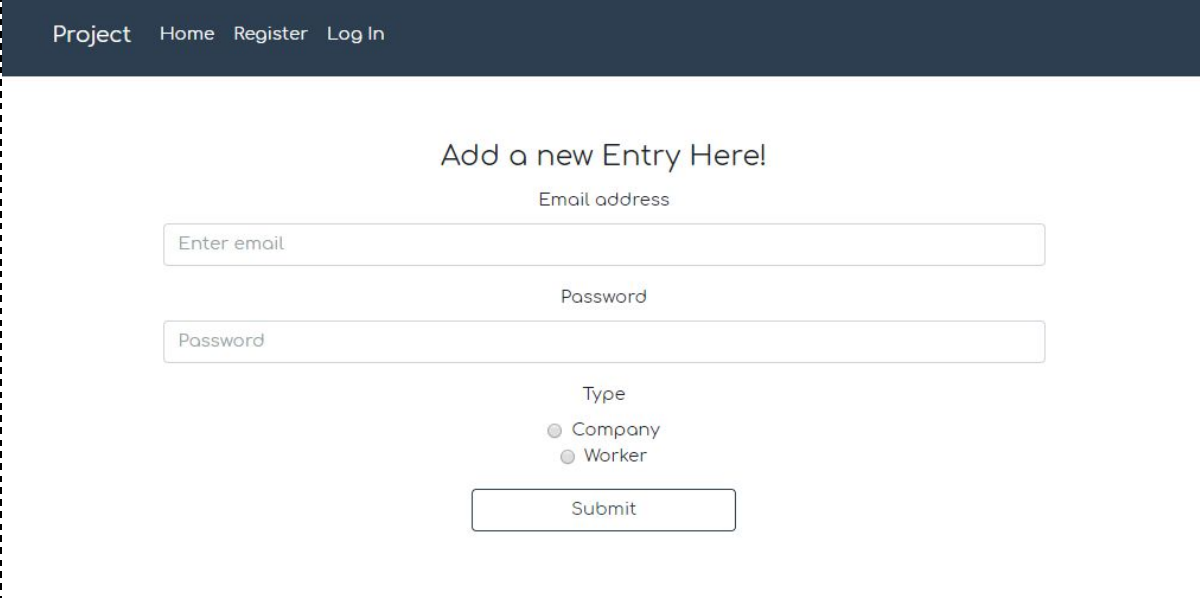
Αρχικά, ο χρήστης εισέρχεται στην αρχική σελίδα της εφαρμογής, όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 5.1 Αρχική σελίδα εφαρμογής

Στην οποία του λέμε τη χρηστικότητα της εφαρμογής και μετά τον καθοδηγούμε να κάνει είσοδο αν έχει δημιουργήσει ήδη χρήστη ή εγγραφή αν δεν έχει δημιουργήσει. Έστω ότι στη περίπτωση μας ο χρήστης δεν έχει εγγραφεί στην εφαρμογή. Τότε επιλέγει το κουμπί

“Register” και μεταφέρεται στη σελίδα εγγραφής όπου καλείται να συμπληρώσει μια φόρμα εγγραφής όπως φαίνεται παρακάτω.



The screenshot shows a registration form titled "Add a new Entry Here!". At the top, there is a dark blue navigation bar with the links "Project", "Home", "Register", and "Log In". The form itself is centered and contains the following elements: a heading "Add a new Entry Here!", a label "Email address" above a text input field with the placeholder "Enter email", a label "Password" above another text input field with the placeholder "Password", a "Type" section with two radio button options: "Company" and "Worker", and a "Submit" button at the bottom.

Εικόνα 5.2 Σελίδα δημιουργίας χρήστη εφαρμογής

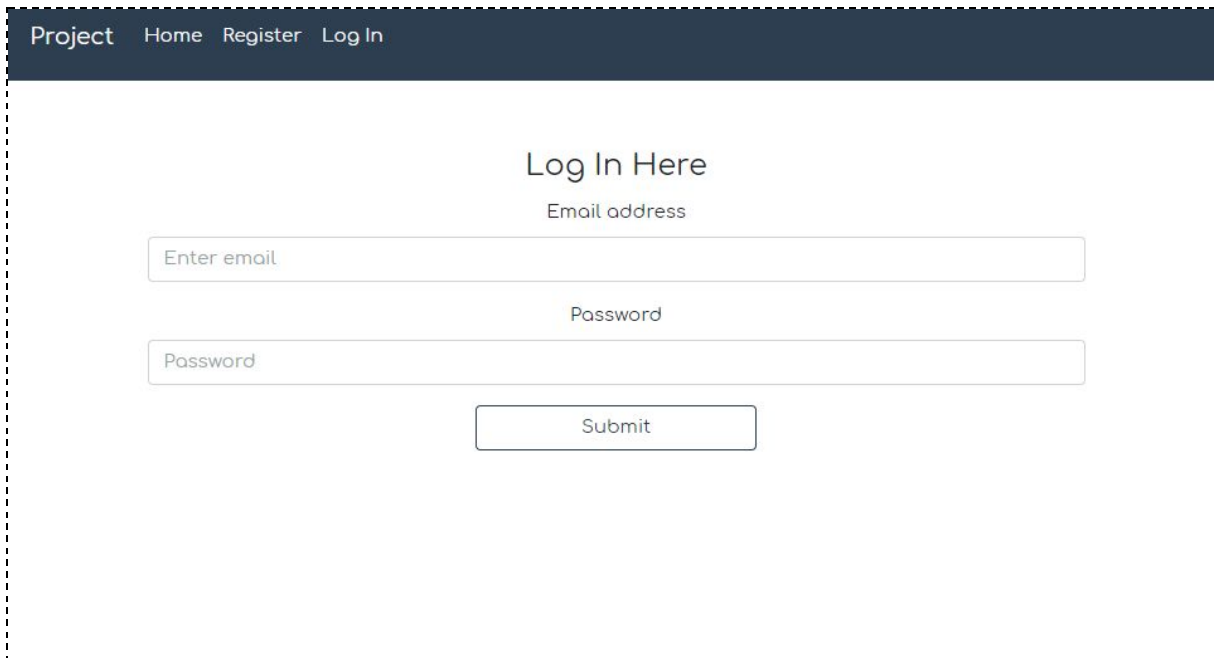
Μόλις συμπληρώσει τη φόρμα εγγραφής και η εγγραφή του είναι επιτυχής του εμφανίζεται ένα μήνυμα επιτυχίας, που του λέει να κάνει είσοδο στην εφαρμογή, όπως φαίνεται παρακάτω.



The screenshot shows the same navigation bar as in the previous image. In the center of the page, there is a green notification box with a white border and a close button (an 'x' icon) on the right. The text inside the box reads: "Your profile is created! Log in to see it."

Εικόνα 5.3 Σελίδα επιτυχής εγγραφής χρήστη

Επιλέγοντας την επιλογή “Log In” από τη κεφαλίδα της εφαρμογής μεταβαίνει στη σελίδα εισόδου.



Log In Here

Email address

Enter email

Password

Password

Submit

Εικόνα 5.4 Σελίδα εισόδου εφαρμογής

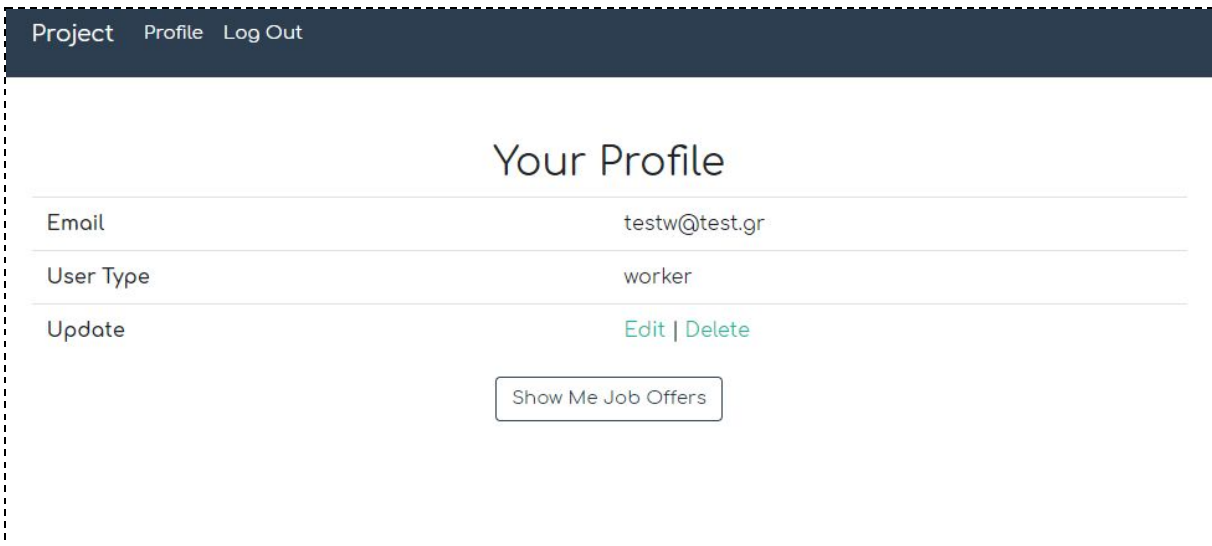
Μετά την επιτυχή είσοδο του ο χρήστης μεταφέρεται στο προφίλ του όπου και του παρουσιάζονται διάφορες ενέργειες που μπορεί να κάνει ανάλογα αν είναι υποψήφιος εργαζόμενος ή επιχείρηση.

## 5.1. Υποψήφιος Εργαζόμενος

Στο προφίλ του, ο εργαζόμενος βλέπει τα στοιχεία τους, τα οποία αν δεν έχει ολοκληρώσει και δεν έχει βάλει όλες τις πληροφορίες του είναι το email του και ο τύπος χρήστη που έχει φτιάξει, επίσης μπορεί να εκτελέσει τις ακόλουθες ενέργειες:

- Να επεξεργαστεί το προφίλ του ,
- Να το διαγράψει,
- Και να δει τις διαθέσιμες αγγελίες εργασίας.

Οι διαθέσιμες θέσεις εργασίας δε θα φέρουν αποτέλεσμα αν το προφίλ του χρήστη δεν είναι ολοκληρωμένο



*Εικόνα 5.5 Σελίδα προφίλ χρήστη υποψήφιου εργαζόμενου*

Ο χρήστης υποψήφιος εργαζόμενος με το πάτημα του κουμπιού “Edit” μεταφέρεται στη σελίδα επεξεργασίας του προφίλ του, όπου καλείται να συμπληρώσει μία φόρμα με τα στοιχεία του.

## Here You can edit Your Profile!

Email address  
testw@test.gr

Password  
\*\*\*\*

Type  
 Company  
 Worker

First Name  
Enter the title

Last Name  
Enter the title

Graduation Level  
 Undergraduate  
 Bachelor  
 Master

Basic info

Age  
31-Dec-1969

Address  
Enter your address

Phone  
telephone

Your work Experience  
Months

Your Skills  
Enter a skill

Known Languages  
Enter a language

Submit

Εικόνα 5.6 Σελίδα επεξεργασίας προφίλ υποψηφίου εργαζόμενου

Μόλις ο χρήστης συμπληρώσει την φόρμα, πατάει το κουμπί “Submit” προκειμένου να αποθηκευτεί το βιογραφικό του στο σύστημα. Τέλος, με την επιτυχή καταχώρηση του βιογραφικού του στη βάση δεδομένων της εφαρμογής, του εμφανίζεται ένα μήνυμα επιτυχίας. Σε περίπτωση που επιθυμεί, μπορεί να επιστρέψει στην σελίδα του προφίλ του πατώντας στο μήνυμα επιτυχίας που εμφανίζεται.

## Your Profile

Email	testw@test.gr
User Type	worker
First Name	Thanasis
Last Name	Alex
Age	1994-12-31
Address	Athens
Phone	0
Graduation Level	under
Your work Experience	12 Months
Known Languages	Greek*English
Your work skills	PHP
Basic info	Something, Something, Something, Something
Update	<a href="#">Edit</a>   <a href="#">Delete</a>

Show Me Job Offers

Εικόνα 5.7 Σελίδα ολοκληρωμένου προφίλ υποψηφίου εργαζόμενου ως ιδιοκτήτης του προφίλ

Παραπάνω φαίνεται το ολοκληρωμένο προφίλ του υποψηφίου εργαζόμενου, τώρα αν αυτός πατήσει το κουμπί “Show Me Job Offers” θα του εμφανίσει τις αγγελίες εργασίας με πρώτη τη καταλληλότερη για αυτόν, όπως φαίνεται παρακάτω.

## Best Job for You

Junior Front-end developer	Athens	Score = 6	<a href="#">View</a>
Web Developer	Sparta	Score = 4	<a href="#">View</a>

Εικόνα 5.8 Σελίδα αντιστοίχισης υποψηφίου εργαζόμενου με αγγελία εργασίας



Από εδώ ο υποψήφιος εργαζόμενος μπορεί να δει το τίτλο της εργασίας που προσφέρεται, τη περιοχή στην οποία βρίσκεται και μπορεί πατώντας το κουμπί “View” να δει όλα τα στοιχεία της θέσεις αυτής.

Project Profile Log Out	
<h2>Job Offer</h2>	
Job Tittle	Junior Front-end developer
Basic info	learn and utilize CSS
Job Location	Athens
Graduation Level Needed	under
Work Experience Needed	0 Months
Required Languages	English*Greek
Skills Required	CSS
Company Name	GWII
About the Company	Somthing
Address	Athens
Phone	2314122
Email	company@company.com
<a href="#">Visit The Company</a>	
<a href="#">Contact</a>	

Εικόνα 5.9 Σελίδα αγγελίας εργασίας ως υποψήφιος εργαζόμενος

Εδώ ο εργαζόμενος μπορεί να δει όλα τα στοιχεία της αγγελίας, να επισκεφτεί την εταιρεία που την έχει εισάγει και να επικοινωνήσει με την εταιρεία μέσω μιας φόρμας επικοινωνίας πατώντας το κουμπί “Contact”.

Project	Profile	Log Out
Skills Required	CSS	
Company Name	GWII	
About the Company	Something	
Address	Athens	
Phone	2314122	
Email	company@company.com	
<a href="#">Visit The Company</a>		
<b>Contact</b>		
Name		
<input type="text"/>		
Email		
<input type="text"/>		
Message		
<input type="text"/>		
<input type="submit" value="Submit"/>		

Εικόνα 5.10 Φόρμα επικοινωνίας στη σελίδα εργασίας

Πατώντας το κουμπί “Contact” ανοίγει ένα πανελ με τη φόρμα επικοινωνίας που ο υποψήφιος εργαζόμενος συμπληρώνει και στέλνει στην εταιρεία.

Από την ίδια σελίδα μπορεί και να επισκεφτεί τη σελίδα της εταιρείας πατώντας το κουμπί “Visit The Company” όπου τον οδηγεί στη σελίδα του προφίλ της εταιρείας. Εδώ μπορεί να δει τα στοιχεία της εταιρείας, άλλες αγγελίες εργασίας που έχει καταχωρήσει η εταιρεία και να επικοινωνήσει με την εταιρεία μέσω του ίδιου μηχανισμού όπως και στην αγγελία εργασίας.

## Your Profile

Email	company@company.com
User Type	company
Company Name	GWII
About the Company	Something
Address	Athens
Phone	2314122

## Job Offers

Junior Front-end developer Athens [View](#)

Contact

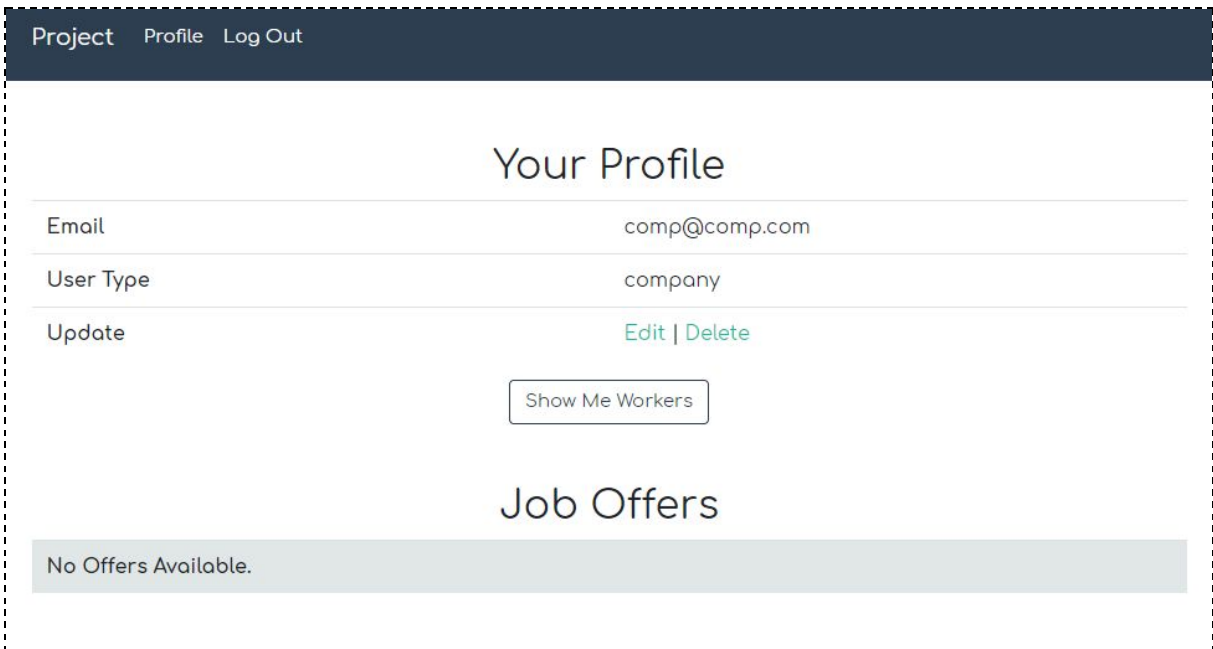
Εικόνα 5.11 Σελίδα προφίλ επιχείρησης ως υποψήφιος εργαζόμενος

## 5.2. Επιχείρηση

Στο προφίλ της η επιχείρηση βλέπει τα στοιχεία της, τα οποία αν δεν είναι ολοκληρωμένα γιατί δεν έχουν εισαχθεί όλες οι πληροφορίες εμφανίζεται μόνο το email και ο τύπος χρήστη που έχει υποβάλλει, επίσης μπορεί να εκτελέσει τις ακόλουθες ενέργειες:

- Να επεξεργαστεί το προφίλ,
- Να το διαγράψει,
- Να δει τις θέσεις εργασίας που έχει καταχωρήσει,
- Και να δει τους διαθέσιμους υποψηφίους.

Οι θέσεις εργασίας θα είναι κενές αφού δε μπορεί να εισάγει νέα αν δεν έχει δημιουργήσει ολοκληρωμένο προφίλ, και το κουμπί “Show me workers” δε θα φέρει αποτέλεσμα επειδή δεν έχει καταχωρηθεί κάποια θέση εργασίας.



Εικόνα 5.12 Σελίδα προφίλ επιχείρησης

Ο χρήστης με το πάτημα του κουμπιού “Edit” μεταφέρεται στη σελίδα επεξεργασίας του προφίλ του, όπου καλείται να συμπληρώσει μία φόρμα με τα απαραίτητα στοιχεία.

## Here You can edit Your Profile!

Email address

comp@comp.com

Password

....

Type

- Company  
 Worker

Company Name

Enter the Name of the Company

About the Company

Address

Enter the Address

Phone Number

telephone

Submit

Εικόνα 5.13 Σελίδα επεξεργασίας προφίλ επιχείρησης

Μόλις ο χρήστης συμπληρώσει την φόρμα, πατάει το κουμπί “Submit” προκειμένου να αποθηκευτούν τα στοιχεία του στο σύστημα. Τέλος, με την επιτυχή καταχώρηση στη βάση δεδομένων της εφαρμογής, εμφανίζεται ένα μήνυμα επιτυχίας. Σε περίπτωση που επιθυμεί, μπορεί να επιστρέψει στην σελίδα του προφίλ του πατώντας στο μήνυμα επιτυχίας που εμφανίζεται.

## Your Profile

Email comp@comp.com

User Type company

Company Name NWO

About the Company Somethign

Address Athens

Phone 1234567890

Add Job Offer [New Job Offer](#)

Update [Edit](#) | [Delete](#)

[Show Me Workers](#)

## Job Offers

No Offers Available.

Εικόνα 5.14 Σελίδα ολοκληρωμένου προφίλ επιχείρησης ως ιδιοκτήτης του προφίλ

Αφού ολοκληρώσει το προφίλ του, ξεκλειδώνεται η επιλογή να δημιουργήσει μια θέση εργασίας.

Πατώντας το κουμπί “New Job Offer” μεταφέρετε στη φόρμα δημιουργίας νέας θέσης εργασίας.

Project Profile Log Out

### Add a new job Here!

Job title

Job information

Graduation Level Needed

- Undergraduate
- Bachelor
- Master

Needed job Experience

Job Skills

+ -

Known Languages

+ -

Job location

Submit

Εικόνα 5.15 Σελίδα εισαγωγής νέας αγγελίας εργασίας

Μόλις ο χρήστης συμπληρώσει την φόρμα, πατάει το κουμπί “Submit” προκειμένου να αποθηκευτεί η αγγελία στο σύστημα. Τέλος, με την επιτυχή καταχώρηση στη βάση δεδομένων της εφαρμογής, εμφανίζεται ένα μήνυμα επιτυχίας. Σε περίπτωση που επιθυμεί, μπορεί να επιστρέψει στην σελίδα του προφίλ του πατώντας στο μήνυμα επιτυχίας που εμφανίζεται την επιλογή “Go to your profile”.

Τώρα στη σελίδα του προφίλ κάτω από το “Job Offers” βλέπει την αγγελία του.

Update Edit | Delete

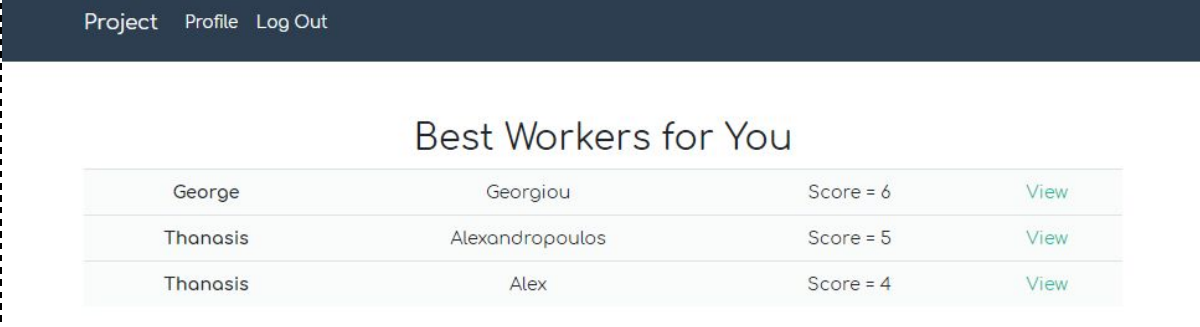
Show Me Workers

### Job Offers

Web developer	Thessalonikh	View
---------------	--------------	------

Εικόνα 5.16 Προβολή αγγελιών εργασίας στο προφίλ της επιχείρησης

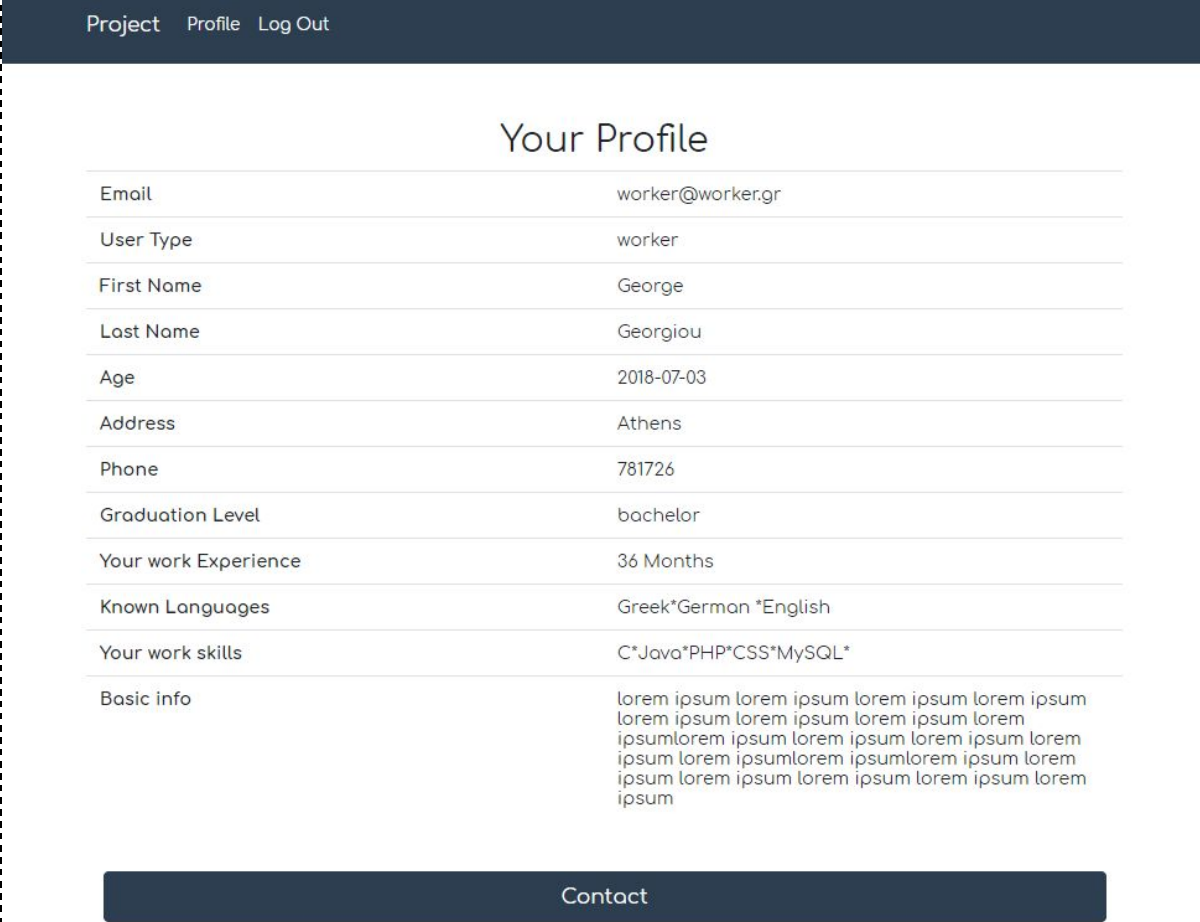
Αφού έχει εισάγει κάποιες αγγελίες εργασίας, πατώντας το κουμπί “Show Me Workers” μπορεί να δει τους διαθέσιμους εργαζόμενους για όλες τις αγγελίες που έχει εισάγει.



Name	Last Name	Score	Action
George	Georgiou	Score = 6	<a href="#">View</a>
Thanasis	Alexandropoulos	Score = 5	<a href="#">View</a>
Thanasis	Alex	Score = 4	<a href="#">View</a>

Εικόνα 5.17 Σελίδα αντιστοίχισης εταιρείας με υποψήφιους εργαζόμενους

Και πατώντας στην επιλογή “View” μεταβαίνει στο προφίλ τους για να δει όλες τις πληροφορίες που είναι διαθέσιμες.



Field	Value
Email	worker@worker.gr
User Type	worker
First Name	George
Last Name	Georgiou
Age	2018-07-03
Address	Athens
Phone	781726
Graduation Level	bachelor
Your work Experience	36 Months
Known Languages	Greek*German *English
Your work skills	C*Java*PHP*CSS*MySQL*
Basic info	lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsumlorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsumlorem ipsumlorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum

Contact

Εικόνα 5.18 Σελίδα προφίλ υποψηφίου εργαζομένου ως εταιρεία

Και πατώντας το κουμπί έχει πρόσβαση στη φόρμα επικοινωνίας, όπως δείξαμε και πιο πάνω στη περίπτωση των υποψηφίων εργαζομένων.



Επίσης από το προφίλ του μπορεί να επισκεφτεί τις αγγελίες εργασίας που έχει εισάγει πατώντας την επιλογή “View” που βρίσκεται δίπλα από κάθε μια.

Job Offer	
Job Tittle	Web developer
Basic info	Code happily
Job Location	Thessalonikh
Graduation Level Needed	bachelor
Work Experience Needed	24 Months
Required Languages	Greek*English
Skills Required	PHP*CSS*HTML5
Company Name	NWO
About the Company	Somethign
Address	Athens
Phone	1234567890
Email	comp@comp.com
Edit the offer	<a href="#">Edit</a>   <a href="#">Delete</a>

[Show Me Workers](#)

Εικόνα 5.19 Σελίδα αγγελίας εργασίας ως εταιρεία που την έχει υποβάλει

Εδώ βλέπει όλες τις πληροφορίες τις αγγελίας, μπορεί να την επεξεργαστεί αλλά και να τη διαγράψει. Επίσης μπορεί να δει τους διαθέσιμους υποψήφιους εργαζόμενους για τη συγκεκριμένη θέση πατώντας το κουμπί “Show Me Workers” και μεταβαίνει στη σελίδα με τους διαθέσιμους υποψηφίους, όπως και στη προηγούμενη περίπτωση, αλλά αυτή τη φορά βλέπει τα αποτελέσματα για τη συγκεκριμένη αγγελία εργασίας και όχι για όλες όσες έχει καταχωρήσει.

Τέλος όλοι οι χρήστες μπορούν να αποσυνδεθούν από την εφαρμογή πατώντας την επιλογή “Log Out” από το μενού.

# Επίλογος

Στο πλαίσιο της εργασίας αναπτύχθηκε εφαρμογή με την οποία είναι εφικτό ένας χρήστης να δημιουργήσει προφίλ υπαλλήλου ή εταιρείας, να το εμπλουτίσει με τα στοιχεία του, να υποβάλει αγγελίες για θέσεις εργασίας στη περίπτωση της επιχείρησης, τέλος στη περίπτωση του υποψηφίου υπαλλήλου να βρει μια θέση εργασίας, βασισμένη στα προσόντα που έχει δώσει στην εφαρμογή και στη περίπτωση που ο χρήστης αντιπροσωπεύει μια επιχείρηση να βρει ένας υπάλληλο βάση των κριτηρίων που έχει θέσει στην αγγελία που έχει υποβάλει. Για να αναπτυχθεί η παρούσα εργασία δημιουργήθηκε μια εφαρμογή διαδικτύου βασισμένη στις γλώσσες προγραμματισμού HTML, PHP, MySQL και Javascript, έγινε χρήση Apache web server και MySQL server μέσω της πλατφόρμας XAMPP. Συγκεκριμένα κατά τη τελειοποίηση της εφαρμογής, βλέπουμε ότι υλοποιήθηκε:

- Η αρχική σελίδα της εφαρμογής,
- Ο μηχανισμός δημιουργίας χρήστη,
- Ο μηχανισμός εισόδου εξόδου από την εφαρμογή,
- Για το προφίλ:
  - Η φόρμα ολοκλήρωσης και επεξεργασίας του προφίλ του χρήστη, για εταιρείες και εργαζόμενους,
  - Η προβολή των άλλων χρηστών από κάποιο χρήστη,
  - Η φόρμα επικοινωνίας που μπορεί να συμπληρώσει ένας ενδιαφερόμενος χρήστης,
  - Η διαγραφή του χρήστη.
- Για τις θέσεις εργασίας:
  - Η εισαγωγή θέσεων εργασίας από τις εταιρείες,
  - Η διαγραφή των θέσεων εργασίας από τις εταιρείες,
  - Η επεξεργασία των θέσεων εργασίας από τους χρήστες που τις εφτιαξαν,
  - Η προβολή των θέσεων εργασίας από όλους τους χρήστες,
  - Η φόρμα επικοινωνίας που μπορούν να συμπληρώσουν οι ενδιαφερόμενοι για κάποια αγγελία εργασίας.
- Ο μηχανισμός εύρεσης καταλληλότερου υποψηφίου εργαζόμενου ή καταλληλότερης αγγελίας εργασίας,
- Το γραφικό περιβάλλον της εφαρμογής.

Ένα από τα πιο σημαντικά κομμάτια της εργασίας είναι ο αλγόριθμος που συμβάλλει στην εύρεση του καταλληλότερου υπαλλήλου για μια θέση εργασίας και το αντίθετο, αυτό συμβάλλει στη διευκόλυνση του χρήστη να βρει αυτό που ψάχνει εύκολα και γρήγορα παραθέτοντας του βάση των στοιχείων που έχει δώσει στην εφαρμογή με τα καλύτερα δυνατά αποτελέσματα είτε πρόκειται για εργαζόμενο που αναζητά μια θέση εργασίας, είτε για επιχείρηση που αναζητά ένα νέο υπάλληλο.

Επιπρόσθετα για τις ανάγκες της παρούσας εργασίας δημιουργήθηκαν φόρμες για την δημιουργία προφίλ υποψηφίων εργαζομένων και εταιρειών, που δρουν και ως βιογραφικό στη περίπτωση του υποψηφίου εργαζομένου, και για τη δημιουργία αγγελιών θέσεων εργασίας, οι οποίες αποτελούνται από ένα σύνολο ενδεικτικών πεδίων και χρήζουν

επέκτασης προκειμένου να καλύψουν τις ανάγκες αποθήκευσης της απαιτούμενης πληροφορίας. Για τις ανάγκες της παρούσας εργασίας αποφασίστηκε η χρήση μιας διαδικτυακής υποδομής περιορισμένων δυνατοτήτων.

## Μελλοντική εξέλιξη

Ξεκινώντας από το παρών διαδικτυακό σύστημα το οποίο είναι σε θέση να δέχεται νέους χρήστες οι οποίοι να μπορούν να πραγματοποιήσουν τις απαραίτητες ενέργειες που προσφέρονται, δημιουργία προφίλ, δημιουργία αγγελίας για θέση εργασίας, φτάνουν στο σημείο να αντιστοιχηθούν με τη κατάλληλη αγγελία ή μια αγγελία να μπορεί να αντιστοιχηθεί με τον κατάλληλο υποψήφιο και δεδομένου ότι στην εποχή μας τα πάντα συνεχίζουν να εξελίσσονται, παρατηρούμε ότι υπάρχει μια σειρά από σημεία στα οποία θα μπορούσε η εφαρμογή να βελτιστοποιηθεί αλλά και να βελτιστοποιείται ώστε να μπορεί να γίνει καλύτερη και γιατί όχι να μπορεί να ανταγωνιστεί παρόμοια εμπορικά συστήματα.

Συγκεκριμένα στο τομέα της ασφάλειας θα μπορούσαν να γίνουν κάποιες αλλαγές όπως, τον εμπλουτισμός του κώδικα με κάποιες συναρτήσεις φίλτρων της php ώστε να αποφύγουμε κακόβουλες επιθέσεις με στόχο τη βάση δεδομένων, την εφαρμογή των απαραίτητων δράσεων σε επίπεδο server ώστε να διασφαλιστεί ότι η εφαρμογή μας δε θα κινδυνεύσει από μια DDos Attack και άλλες παρόμοιες ενέργειες.

Στον τομέα των στοιχείων που αφορούν τον υποψήφιο κάποιες από τις αλλαγές που θα μπορούσαν να γίνουν θα ήταν, να μπορεί να προσθέσει και άλλες πληροφορίες για αυτόν στο προφίλ του, όπως φωτογραφία, προηγούμενες θέσεις εργασίας, τίτλο σπουδών να μπορεί να προσθέσει άτομα που γνωρίζει η τον ενδιαφέρουν σε ένα σύνολο παραδείγματος χάρη όπως το LinkedIn έχει τα “connections”, επίσης θα μπορούσε να υλοποιηθεί ένα σύστημα συστάσεων τόσο για τους υπαλλήλους όσο και για της εταιρείες που θα μπορούσε ίσως να γίνει και ανώνυμο.

Στον τομέα που αφορά τις επιχειρήσεις κάποιες εξελίξεις που θα μπορούσαν να εφαρμοστούν θα ήταν παρόμοιες με των υποψηφίων, αλλά θα μπορούσαν να τους προβάλλονται και διάφορα γραφήματα για τη πρόοδο της αγγελίας τους, για την ανταγωνιστικότητα τους για παράδειγμα στο επίπεδο μισθών, συνθηκών εργασίας κ.α..

Τεχνικά το σύστημα μπορεί να αλλάξει βρίσκοντας τους κατάλληλους υπολογιστικούς πόρους για να μπορεί να είναι διαθέσιμο στο ευρύ κοινό, και επίσης θα μπορούσε να βελτιστοποιηθεί σε μεγαλύτερο βαθμό ο κώδικας για την εξομάλυνση της εμπειρίας του χρήστη αλλά και τη καλύτερη διαχείριση των υπολογιστικών πόρων που παρέχονται στην εφαρμογή, ακόμα θα μπορούσε να βασιστεί σε κάποιο framework για να παρέχει μεγαλύτερη ταχύτητα και ευκολία στην ανάπτυξη και περαιτέρω εξέλιξη της.

# Βιβλιογραφία

1. Adler L. (2016). 'New Survey Reveals 85% of All Jobs are Filled Via Networking' Available at: <https://www.linkedin.com/pulse/new-survey-reveals-85-all-jobs-filled-via-networking-lou-adler/>
2. Andrew S. Tanenbaum, (2009). 'Συγχρονα Λειτουργικά Συστήματα', Εκδόσεις Κλειδαριθμός Επτε
3. Chaidaroglou A., 'LinkedIn και αναζήτηση εργασίας'. Available at: <https://www.e-biografiko.gr/linkedin-%CE%BA%CE%B1%CE%B9-%CE%B1%CE%BD%CE%B1%CE%B6%CE%AE%CF%84%CE%B7%CF%83%CE%B7-%CE%B5%CF%81%CE%B3%CE%B1%CF%83%CE%AF%CE%B1%CF%82/>
4. Cochran, David (November 12, 2012). Twitter Bootstrap Web Development (1st έκδοση). Packt Publishing
5. Duermyer R. (2018). 'Learn What LinkedIn Does and How It Works'. Available at: <https://www.thebalancesmb.com/introduction-to-linkedin-1794572>
6. Hempel J. (2013), 'LinkedIn: How it's changing business (and how to make it work for you)'. Available at: <http://fortune.com/2013/06/13/linkedin-how-its-changing-business-and-how-to-make-it-work-for-you/>
7. Jacobs D. (2013). 'The Six Best Ways To Find Your Next Job' Available at: <https://www.forbes.com/sites/deborahjacobs/2013/03/22/the-six-best-ways-to-find-your-next-job/#2e9560f51369>
8. Miller-Merrell J. (2016). '6 Shocking Job Search Statistics You Need to Know' Available at: <https://workology.com/6-shocking-job-search-statistics-im-discussing-at-sxswi/>
9. Ramakrishnan R., Gehrke J., (2012). 'Συστήματα Διαχείρισης Βάσεων Δεδομένων', Εκδόσεις Α. ΤΖΙΟΛΑ & ΥΙΟΙ Α.Ε.
10. Royce , W. (1970). 'Managing the development of large software systems'. IEEE WESCON, Proceeding (August):1-9
11. Smith A. (2015). 'Searching for Work in the Digital Era' διαθέσιμο από: <http://www.pewinternet.org/2015/11/19/searching-for-work-in-the-digital-era/>
12. Sommerville I. (2011). 'Βασικές Αρχές Τεχνολογίας Λογισμικού', μετάφραση Τσιλογιάννης Δ., Εκδόσεις Κλειδαριθμός

13. Software Engineering GR (2017). 'Μοντέλο Καταρράκτη' διαθέσιμο από:  
<http://softwareengineering.gr/tecnologia-logismikou/sxediasi-logismikou/montelo-katarakth/>
14. Welling L., Thomson L. (2011). 'Ανάπτυξη Web Εφαρμογών με PHP και MySQL, 4η Έκδοση', Εκδόσεις Α. Γκιούρδα & ΣΙΑ ΟΕ
15. Think Tech (2015). 'Τι είναι το LinkedIn, και πως μπορώ να φτιάξω ένα επαγγελματικό profile.' διαθέσιμο από:  
<http://thinktech.gr/%CF%84%CE%B9-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9-%CF%84%CE%BF-linkedln/>
16. Κιουντούζης Ευ. (2009). 'Μεθοδολογίες Ανάλυσης & Σχεδιασμού Πληροφοριακών Συστημάτων', Εκδόσεις Μπένου Ε.
17. Σκουρλάς Χ. (2001). 'Υλοποίηση Εφαρμογών με Γλώσσα SQL' Εκδόσεις: Εκδόσεις Νέων Τεχνολογιών.
18. Ζιανικα Μ. (2009). 'Κρίσιμοι παράγοντες βελτίωσης της διαδικασίας ανάπτυξης λογισμικού: σύγχρονες τάσεις και προκλήσεις', προπτυχιακή διπλωματική, Θεσσαλονίκη: Πανεπιστήμιο Μακεδονίας, διαθέσιμο από:  
<https://dspace.lib.uom.gr/bitstream/2159/13533/1/ZianikaMsc2009.pdf>
19. <http://php.net/manual/en/index.php>
20. <https://el.wikipedia.org/wiki/PHP>
21. <https://el.wikipedia.org/wiki/HTML>
22. <https://el.wikipedia.org/wiki/CSS>
23. <https://el.wikipedia.org/wiki/JavaScript>
24. <https://el.wikipedia.org/wiki/MySQL>
25. <https://www.w3schools.com/>
26. <https://getbootstrap.com/>
27. <https://www.apachefriends.org/index.html>
28. <https://www.phpmyadmin.net/>
29. <http://www.uml.org/what-is-uml.htm>

# Παράρτημα Ι - Κώδικας εφαρμογής

## 1. Κώδικας SQL για τη δημιουργία της βάσης δεδομένων

```
CREATE TABLE `comptab` (  
  `cid` int(10) NOT NULL,  
  `cname` varchar(50) DEFAULT NULL,  
  `cabout` varchar(1000) DEFAULT NULL,  
  `caddress` varchar(50) NOT NULL,  
  `ctel` int(15) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
-----  
  
--  
-- Table structure for table `jobOffer`  
--  
  
CREATE TABLE `jobOffer` (  
  `jid` int(10) NOT NULL,  
  `jtittle` varchar(50) NOT NULL,  
  `jinfo` varchar(1000) DEFAULT NULL,  
  `jskills` varchar(1000) NOT NULL,  
  `jgradlvl` varchar(10) NOT NULL,  
  `jexp` int(10) NOT NULL,  
  `jlang` varchar(100) NOT NULL,  
  `jloc` varchar(100) NOT NULL,  
  `cid` int(10) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
-----  
  
--  
-- Table structure for table `usertab`  
--  
  
CREATE TABLE `usertab` (  
  `id` int(10) NOT NULL,  
  `email` varchar(50) NOT NULL,  
  `pass` varchar(50) NOT NULL,  
  `type` varchar(10) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
-----
```

```

--
-- Table structure for table `worker`
--

CREATE TABLE `worker` (
  `wid` int(10) NOT NULL,
  `wname` varchar(100) NOT NULL,
  `wlname` varchar(100) NOT NULL,
  `wAge` date DEFAULT NULL,
  `waddress` varchar(100) DEFAULT NULL,
  `wtel` int(15) DEFAULT NULL,
  `wgradlvl` varchar(10) NOT NULL,
  `wexp` int(10) NOT NULL,
  `wlang` varchar(100) NOT NULL,
  `winfo` varchar(1000) NOT NULL,
  `wskills` varchar(1000) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Indexes for dumped tables
--

--
-- Indexes for table `comptab`
--
ALTER TABLE `comptab`
  ADD PRIMARY KEY (`cid`);

--
-- Indexes for table `jobOffer`
--
ALTER TABLE `jobOffer`
  ADD PRIMARY KEY (`jid`),
  ADD KEY `jobOffer_fk0` (`cid`);

--
-- Indexes for table `usertab`
--
ALTER TABLE `usertab`
  ADD PRIMARY KEY (`id`);

--
-- Indexes for table `worker`
--
ALTER TABLE `worker`
  ADD PRIMARY KEY (`wid`);

--

```

```

-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `comptab`
--
ALTER TABLE `comptab`
  MODIFY `cid` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `jobOffer`
--
ALTER TABLE `jobOffer`
  MODIFY `jid` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

--
-- AUTO_INCREMENT for table `usertab`
--
ALTER TABLE `usertab`
  MODIFY `id` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `worker`
--
ALTER TABLE `worker`
  MODIFY `wid` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;
COMMIT;

```

## 2. Κώδικας σύνδεσης με βάση δεδομένων

```

<?php
/**
 * using mysqli_connect to establish connection with the db
 */

$databaseHost = 'dbhost.gr';
$databaseName = 'name';
$databaseUsername = 'user';
$databasePassword = 'pass';

$mysqli = mysqli_connect($databaseHost, $databaseUsername, $databasePassword,
$databaseName);
?>

```

## 3. Κώδικας header.php

```

<?php

```



```

//establishing a session
session_start();
//including the connection file
include_once "config.php";

?>
<html>
    <head>
        <?php //including css, a google font, fontawesome icons, jquery and
bootstrap ?>
            <link rel="stylesheet" type="text/css" href="style.css">
            <script src=
"https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js" ></script>
            <link href=
"https://stackpath.bootstrapcdn.com/bootswatch/4.1.1/flatly/bootstrap.min.css"
rel= "stylesheet" integrity=
"sha384-WuViQmTamrPyvMFZjf8te7HpKtdxuzV/HX1yG26a0d8yieIBr+beDf1ME99iX1cM"
crossorigin= "anonymous">
            <link href=
"https://fonts.googleapis.com/css?family=Comfortaa:300,400,700&subset=greek"
rel= "stylesheet">
            <script src="https://use.fontawesome.com/1b697ddbac.js"></script>
            <title>Π τ υ χ ι α κ ή</title>
            <link rel="shortcut icon" href="favicon1.ico">
        </head>
        <body style="padding-top : 120px;">

            <!-- Script to remove divs -->
            <script type="text/javascript">

                function removeJunk(div){
                    div.remove();
                }
            </script>

            <!-- Navigation Bar -->
            <nav class="navbar navbar-expand-lg navbar-dark bg-primary
fixed-top">
                <div class="container">
                    <a class="navbar-brand" href="#">Project</a>
                    <button class="navbar-toggler" type="button" data-toggle=
"collapse" data-target="#navbarColor01" aria-controls="navbarColor01"
aria-expanded="false" aria-label="Toggle navigation">
                        <span class="navbar-toggler-icon"></span>
                    </button>

                    <div class="collapse navbar-collapse" id="navbarColor01">
                        <ul class="navbar-nav mr-auto">
                            <?php if(!isset($_SESSION["email"])){ ?>
                                <li class="nav-item">
                                    <a class="nav-link" href="home.php">

```

```

Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="add.php">
Register</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="login.php">Log
In</a>
        </li>
        <?php }else{ ?>
        <li class="nav-item">
            <a class="nav-link" href= "profile.php"
> Profile</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href= "logout.php"
>Log Out</a>
        </li>
        <?php } ?>
    </ul>
</div>
</div><!-- Closing container div -->
</nav><!-- End Of navigation bar-->

```

#### 4. Κώδικας ταυτοποίησης

```

<?php
//Checking if user is logged
if(!isset($_SESSION["email"])){
    echo("<script>location.href = '". "home.php";</script>");

    exit;
}
?>

```

#### 5. Κώδικας αρχείου css

```

body{
    font-family: 'Comfortaa', cursive!important;
}
/* Home */
.home{
    text-align: center;
}

```

```

.welcome{
    margin: 1em auto;
}
.welcome h2{
    color:#2C3E50;
}
.logincol{
    margin: 1em auto;
}
.logincol a{
    margin-top: 2em;
}
.subcol{
    margin: 1em auto;
}
/* Profile */
.profile{
    text-align: center;
}
.cstmtable tr th{
    width: 50%;
}

/* Profile */
.profile{
    text-align: center;
}
.cstmtable tr th{
    width: 50%;
}

/* Skills and languages in form */
input#skills, input#lang {
    margin: 0.3em;
    flex-basis: 200px;
}
div#inputs, div#inputslang {
    display: flex;
    flex-flow: row wrap;
    justify-content: center;
    align-items: center;
}

.customform{
    margin: auto;
    width: 80%;
    text-align: center;
}

.contForm{
    margin: 3em auto;
}

```

```

}

/*buttons and alerts*/
.custombtn{
    width: 30%;
    margin: auto;
}
.customalert{
    width: 50%;
    margin: 1em auto;
}

```

## 6. Κώδικας αρχείου footer.php

```

<!-- Script to add or remove input fields -->
<script type="text/javascript">
    // Input adding function
    function addInput() {
        $('#inputs').append('<input type="text" name="skills[]"
class="form-control" id="skills" placeholder="Enter a skill">');
    }
    function addInputlang() {
        $('#inputslang').append('<input type="text" name="languages[]"
class="form-control" id="lang" placeholder="Enter a language">');
    }
    function removeskillInput() {
        $('#inputs input').remove('input:last-child');
    }
    function removelangInput() {
        $('#inputslang input').remove('input:last-child');
    }

    // Event handler and the first input
$(document).ready(function () {
    $('#add').click(addInput);
    //addInput();
    $('#addlang').click(addInputlang);
    //addInputlang();
    $('#remove').click(removeskillInput);
    $('#removelang').click(removelangInput);

    });
</script>
<!-- Including the js for bootstrap -->
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"

```

```

integrity="sha384-q8i/X+965Dz00rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abt
TE1Pi6jizo" crossorigin="anonymous"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.
min.js"
integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/18
WvCWPIpM49" crossorigin="anonymous"></script>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min
.js"
integrity="sha384-smHYKdLADwkXOn1EmN1qk/HfnUcbVRZyYmZ4qpPea6sjB/pTJ0euyQ
p0Mk8ck+5T" crossorigin="anonymous"></script>
    </body>
</html>
<?php mysqli_close($mysqli); //Closing the connection with the DB ?>

```

## 7. Κώδικας αρχικής σελίδας εφαρμογής

```

<?php
//including the header file
include 'head.php';
?>

<title>Home Page</title>

<div class=" container home">

    <div class="row">
        <div class="col-sm welcome">
            <h2>Welcome,</h2>
            <p>This web application helps you to find a job well
suited for you, or a worker for your company.</p>
        </div>
    </div>

    <div class="row" style="text-align:center;">
        <div class="col-sm logincol">
            <p>If you already have an account you can log in here,
else...</p>
            <a href="login.php" class="btn btn-primary custombtn"> Log
in </a>
        </div>
        <div class="col-sm subcol">
            <p>If you are a company, which is trying to find a good
worker or a worker, who searches a fine job you can register here.</p>

```

```

        <a href="add.php" class="btn btn-primary
custombtn">Register</a>
    </div>
</div>

</div>

<?php include 'footer.php';?>

```

## 8. Κώδικας μηχανισμού εισόδου

```

<?php include 'head.php'; ?>

<title>Log in</title>

<div class="login container">

    <div class="customform">

        <form action="login.php" method="post" name="form1">

            <fieldset>
                <legend class="text-center">Log In Here</legend>

                <div class="form-group">
                    <label for="Email">Email address</label>
                    <input type="text" name="email" class="form-control"
id="Email" aria-describedby="emailHelp" placeholder="Enter email">
                </div>

                <div class="form-group">
                    <label for="Password">Password</label>
                    <input type="password" name="pass" class="form-control"
id="Password" placeholder="Password">
                </div>

                <button type="submit" name="Submit" class="btn
btn-outline-primary custombtn">Submit</button>

            </fieldset>

        </form><!-- End of Form -->

    </div><!-- Closing customform div -->

<?php
//Form Handler
if(isset($_POST['Submit'])) {

```

```

to the db //Stripping hazardous symbols for the db and preparing to insert

$email = stripslashes($_POST['email']);
$email = mysqli_real_escape_string($mysqli,$email);

$pass = stripslashes($_POST['pass']);
$pass = mysqli_real_escape_string($mysqli,$pass);

// checking empty fields
if(empty($email) || empty($pass)) {

    if(empty($pass)) { ?>

        <div class="alert alert-dismissible alert-danger
customalert" >
            <button type="button" class="close"
data-dismiss="alert">&times;</button>
            <strong>Password field is empty.</strong> <a
href="javascript:self.history.back();" class="alert-link"> try submitting
again.</a>
        </div>
    <?php }

    if(empty($email)) { ?>
        <div class="alert alert-dismissible alert-danger
customalert" >
            <button type="button" class="close"
data-dismiss="alert">&times;</button>
            <strong>Email field is empty.</strong> <a
href="javascript:self.history.back();" class="alert-link"> try submitting
again.</a>
        </div>
    <?php
}
} else {
    // if all the fields are filled (not empty) inserting the data
$result = mysqli_query($mysqli, "SELECT * FROM usertab WHERE
email='$email' and pass='$pass'") or die(mysql_error());

$rows = mysqli_num_rows($result);
if($rows == 1) {

    $_SESSION['email'] = $email;
    echo("<script>location.href = '". "profile.php";</script>");

}else { ?>

    <div class="alert alert-dismissible alert-danger
customalert">
        <button type="button" class="close"

```

```

data-dismiss="alert">&times;</button>
    <strong>Username/Password is incorrect.</strong><a
href='login.php'>Try Again Later</a>
</div>
<!-- Script to remove the form after it is Submitted -->
<script>
    var element = document.querySelector('.customform');
    removeJunk(element);
</script>

    <?php }
    }
} //End of form Handler
mysqli_free_result($result);
?>

</div><!-- Closing Container div -->
<?php include 'footer.php';?>

```

## 9. Κώδικας μηχανισμού εξόδου

```

<?php
include 'head.php';
// Destroying All Sessions
session_unset();
if(session_destroy()){
    // Redirecting to login page
    echo("<script>location.href = '". "login.php";</script>");
}
?>

```

## 10. Κώδικας δημιουργίας χρήστη

```

<?php include 'head.php'; ?>
    <title>Register</title>

    <div class="add container">

        <div class="customform">

            <form action="add.php" method="post" name="form1">
                <fieldset>
                    <legend class="text-center">Add a new Entry Here!</legend>

                    <div class="form-group">
                        <label for="Email">Email address</label>
                        <input type="email" name="email" class="form-control"

```



```

id="Email" placeholder="Enter email">
    </div>

    <div class="form-group">
        <label for="Password">Password</label>
        <input type="password" name="pass" class="form-control"
id="Password" placeholder="Password">
    </div>

    <label for="radio">Type</label>
    <fieldset class="form-group" id="radio">
        <div class="form-check">
            <label class="form-check-label">
                <input type="radio" class="form-check-input"
name="type" id="option1" value="company">Company
            </label>
        </div>
        <div class="form-check">
            <label class="form-check-label">
                <input type="radio" class="form-check-input"
name="type" id="option2" value="worker">Worker
            </label>
        </div>
    </fieldset>
    <button type="submit" name="Submit" class="btn
btn-outline-primary custombtn">Submit</button>
    </fieldset>
</form>
</div><!-- Closing the Form -->

<?php
// Checking if the Form is Submitted
if(isset($_POST['Submit'])) {

    //Stripping hazardous symbols for the db and preparing to insert
to the db

    $email = stripslashes($_POST['email']);
    $email = mysqli_real_escape_string($mysqli,$email);

    $pass = stripslashes($_POST['pass']);
    $pass = mysqli_real_escape_string($mysqli,$pass);

    $type = $_POST["type"];

    // checking for empty fields
    if(empty($email) || empty($pass)|| empty($type)) {

        if(empty($pass)) { ?>

            <div class="alert alert-dismissable alert-danger
customalert">

```

```

        <button type="button" class="close"
data-dismiss="alert">&times;</button>
        <strong>Password field is empty.</strong> <a
href="javascript:self.history.back();" class="alert-link"> try submitting
again.</a>

    </div>
<?php }

if(empty($email)) { ?>

    <div class="alert alert-dismissable alert-danger
customalert">

        <button type="button" class="close"
data-dismiss="alert">&times;</button>
        <strong>Email field is empty.</strong> <a
href="javascript:self.history.back();" class="alert-link">try submitting
again.</a>

    </div>
<?php }

if(empty($type)) { ?>

    <div class="alert alert-dismissable alert-danger
customalert">

        <button type="button" class="close"
data-dismiss="alert">&times;</button>
        <strong>Type field is empty.</strong> <a
href="javascript:self.history.back();" class="alert-link">try submitting
again.</a>

    </div>
<?php }
} else {

    // if all the fields are filled (not empty)
    //insert data to database
    $result = mysqli_query($mysqli, "INSERT INTO
usertab(email,pass,type) VALUES('$email','$pass','$type')");

    //display success message if the data were inserted
if($result){?>
    <div class="alert alert-dismissable alert-success
customalert">

        <button type="button" class="close"
data-dismiss="alert">&times;</button>
        <strong>Your profile is created!Log in to see it.</strong>
    </div>

    <!-- Script to remove the form after it is Submitted -->
    <script>
        var element = document.querySelector('.customform');
        removeJunk(element);

```

```

        </script>

        <?php }else{ ?>
            <div class="alert alert-dismissible alert-danger
customalert">
                <button type="button" class="close"
data-dismiss="alert">&times;</button>
                <strong>SQL Error!</strong><p>Reload your page and try
again!</p>
            </div>
        <?php }

    }
} //Closing the form Handler
mysqli_free_result($result); ?>

</div><!-- Closing Container -->
<?php include 'footer.php';?>

```

## 11. Κώδικας διαγραφή χρήστη

```

<?php
//including the database connection file
include_once("config.php");?>
<?php

//getting id of the data from url
$id = $_GET['id'];

//deleting the row from table without secondary keys in db
$sql = "DELETE FROM usertab WHERE id=$id; ";
$sql.= "DELETE FROM comptab WHERE cid=$id; ";
$sql.= "DELETE FROM worker WHERE wid=$id; ";
$sql.= "DELETE FROM jobOffer WHERE cid=$id; ";
$result = mysqli_multi_query($mysqli, $sql);

//redirecting to the display page (home.php in our case)
echo("<script>location.href = '". "login.php";</script>");
?>

```

## 12. Κώδικας σελίδας προφίλ πριν το διαχωρισμό του χρήστη

```

<?php
//including the database connection and authentication files
include 'head.php';
include("auth.php");

```

```

//fetching data from the URL
$linkid = $_GET['id'];

//checking if the viewer is a visitor or the owner
if (isset($linkid)) {

    $sec=0;
    $result = mysqli_query($mysqli, "SELECT * FROM usertab WHERE
id='$linkid'") or die(mysqli_error());

}else{

    $sec = 1;
    $email = $_SESSION["email"];
    $result = mysqli_query($mysqli, "SELECT * FROM usertab WHERE
email='$email'") or die(mysqli_error());
}

?>
<title>Profile</title>

<div class="profile container">

    <div class="title"><h3>Your Profile</h3></div>

    <table class="table table-hover csthtable">

        <?php
while($res = mysqli_fetch_array($result)) {
    $id = $res['id'];
    $email = $res['email'];
    $pass = $res['pass'];
    $type = $res['type'];
}
mysqli_free_result($result);?>

        <tr>
            <th>Email</th>
            <?php echo "<td>".$email."</td>"; ?>
        </tr>

        <tr>
            <th>User Type</th>
            <?php echo "<td>".$type."</td>"; ?>

```

```
</tr>
```

### 13. Κώδικας σελίδας προφίλ υποψηφίου εργαζομένου

```
<?php
    //checking if the user is Worker or Company
    if( $type == 'worker'){
        $resultwork = mysqli_query($mysqli, "SELECT * FROM worker WHERE wid
= '$id'" ) or die(mysqli_error());

        while($reswork = mysqli_fetch_array($resultwork)){ ?>

            <tr>
                <th>First Name</th>
                <?php echo "<td>".$reswork['wname']."</td>"; ?>
            </tr>

            <tr>
                <th>Last Name</th>
                <?php echo "<td>".$reswork['wlname']."</td>"; ?>
            </tr>

            <tr>
                <th>Age</th>
                <?php echo "<td>".$reswork['wAge']."</td>"; ?>
            </tr>

            <tr>
                <th>Address</th>
                <?php echo "<td>".$reswork['waddress']."</td>"; ?>
            </tr>

            <tr>
                <th>Phone</th>
                <?php echo "<td>".$reswork['wtel']."</td>"; ?>
            </tr>

            <tr>
                <th>Graduation Level</th>
                <?php echo "<td>".$reswork['wgradlvl']."</td>"; ?>
            </tr>

            <tr>
                <th>Your work Experience</th>
                <?php echo "<td>".$reswork['wexp']." Months</td>"; ?>
            </tr>

            <tr>
                <th>Known Languages</th>
```

```

        <?php echo "<td>".$reswork['wlang']."</td>"; ?>
    </tr>

    <tr>
        <th>Your work skills</th>
        <?php echo "<td>".$reswork['wskills']."</td>"; ?>
    </tr>

    <tr>
        <th>Basic info</th>
        <?php echo "<td>".$reswork['winfo']."</td>"; ?>
    </tr>

<?php }
mysqli_free_result($resultwork);

//checking if the user is the owner or a visitor
if (!isset($linkid)) { ?>

    <tr>
        <th>Update</th>
        <?php echo "<td><a href=\"edit.php?id=$id\">Edit</a> | <a
href=\"delete.php?id=$id\" onClick=\"return confirm('Are you sure you want to
delete this?')\">Delete</a></td>"; ?>
    </tr>

    <a class="btn costbtn btn-outline-primary"
href="automach.php?id=<?php echo $id; ?>">Show Me Job Offers</a>

<?php } ?>

<?php }elseif( $type == 'company'){

```

#### 14. Κώδικας σελίδας προφίλ για την εταιρείας

```

<?php }elseif( $type == 'company'){

    $resultcomp = mysqli_query($mysqli, "SELECT * FROM comptab WHERE cid =
'$id'" ) or die(mysqli_error());

    while($rescomp = mysqli_fetch_array($resultcomp)){ ?>

    <tr>
        <th>Company Name</th>
        <?php echo "<td>".$rescomp['cname']."</td>"; ?>
    </tr>

    <tr>

```

```

        <th>About the Company</th>
        <?php echo "<td>".$rescomp['cabout'].</td>"; ?>
    </tr>

    <tr>
        <th>Address</th>
        <?php echo "<td>".$rescomp['caddress'].</td>"; ?>
    </tr>

    <tr>
        <th>Phone</th>
        <?php echo "<td>".$rescomp['ctel'].</td>"; ?>
    </tr>

    <?php if(!isset($linkid)) { ?>

        <tr>
            <th>Add Job Offer</th>
            <?php echo "<td><a href=\"createjob.php?id=$id\">New Job
Offer</a></td>"; ?>
        </tr>

    <?php }

}
mysqli_free_result($resultcomp); ?>

<?php
//checking if the user is the owner or a visitor
if(!isset($linkid)) { ?>

    <tr>
        <th>Update</th>
        <?php echo "<td><a href=\"edit.php?id=$id\">Edit</a> | <a
href=\"delete.php?id=$id\" onClick=\"return confirm('Are you sure you want to
delete this?')\">Delete</a></td>"; ?>
    </tr>
    <?php } ?>

</table><!-- Closing Profile table if user is Company-->

<?php
//checking if the user is the owner or a visitor
if (!isset($linkid)) {?>
    <a class="btn costbtn btn-outline-primary"
href="automach.php?id=<?php echo $id; ?>">Show Me Workers</a>
    <?php }

//if the user is the owner, presenting their Offers
$offers = mysqli_query($mysqli, "SELECT * FROM jobOffer WHERE cid =

```

```

'$id' ORDER BY jid DESC");

$rowcount = mysqli_num_rows($offers); ?>

<table class="oftab table table-secondary table-hover">

  <div class="title"style="margin-top: 5%;"><h3>Job Offers</h3></div>

  <?php
  //setting the correct id
  $uid = (!isset($linkid)) ? $id : $linkid ;

  if($rowcount > 0){
    //Showing the db records
    while($offer = mysqli_fetch_array($offers)) {
      echo "<tr>";
      echo "<th>".$offer['jtittle']."</th>";
      echo "<th>".$offer['jloc']."</th>";
      echo "<td><a
href=\"joboffer.php?jid=$offer[jid]&id=$uid&sec=$sec\">View</a></td>";
    }
    mysqli_free_result($offers);

  }else{echo"<th>No Offers Available.</th>";} ?>
</table><!-- End offers table -->

<?php }//End worker/company if ?>

</table><!-- Closing Profile Table if it didn't close at Company -->

```

## 15. Κώδικας φόρμας επικοινωνίας

```

<?php //If the user is a visitor showing the contact form
if (isset($linkid)) {?>

  <div class="container contForm">

    <div class="panel-group">

      <div class="panel panel-default">

        <div class="panel-heading">
          <h4 class="panel-title">
            <a data-toggle="collapse" class="btn btn-primary btn-lg
btn-block" href="#collapse1">Contact</a>
          </h4>
        </div>

        <div id="collapse1" class="panel-collapse collapse">

```



```

        <div class="panel-body">

            <?php if($msg != ''): ?>
                <div class="alert <?php echo $msgClass; ?>"><?php echo $msg;
?></div>

                <?php endif; ?>

                <form method="post" class="contForm" action="<?php echo
$_SERVER['PHP_SELF']; ?>">

                    <div class="form-group">
                        <label>Name</label>
                        <input type="text" name="name" class="form-control"
value="<?php echo isset($_POST['name']) ? $name : ''; ?>">
                    </div>

                    <div class="form-group">
                        <label>Email</label>
                        <input type="email" name="email" class="form-control"
value="<?php echo isset($_POST['email']) ? $senderEmail : ''; ?>">
                    </div>

                    <div class="form-group">
                        <label>Message</label>
                        <textarea name="message" class="form-control"><?php echo
isset($_POST['message']) ? $message : ''; ?></textarea>
                    </div>

                    <button type="submit" name="submit" class="btn
btn-primary">Submit</button>

                </form><!-- End of Contact Form -->

            </div><!-- Closing the panel-body div -->

        </div><!-- Closing the panel-collapse div -->

    </div><!-- Closing the secondary panel div -->

</div><!-- End of Panel -->

</div><!-- End of contact form -->

<?php }//End of if visitor ?>

</div><!-- End of container -->

<?php
//Contact form Handler

```

```

// Message Vars
$msg = '';
$msgClass = '';

// Check For Submit
if(isset($_POST['submit'])){
    // Get Form Data
    $name = htmlspecialchars($_POST['name']);
    $senderEmail = htmlspecialchars($_POST['email']);
    $message = htmlspecialchars($_POST['message']);

    // Check Required Fields
    if(!empty($senderEmail) && !empty($name) && !empty($message)){

        $toEmail = $email;
        $subject = 'Contact Request From '.$name;
        $body = '<h2>Contact Request</h2>
<h4>Name</h4><p>'.$name.'</p>
<h4>Email</h4><p>'.$senderEmail.'</p>
<h4>Message</h4><p>'.$message.'</p>
';
        // Email Headers
        $headers = "MIME-Version: 1.0" ."\r\n";
        $headers .= "Content-Type:text/html;charset=UTF-8" . "\r\n";

        // Additional Headers
        $headers .= "From: " . $name. "<".$senderEmail.">". "\r\n";

        /*
        * It Should be like that:
        * if(mail($toEmail, $subject, $body, $headers)){
        *     $msg = 'Your email has been sent';
        *     $msgClass = 'alert-success';
        * }
        * But because we don't have a mail service in place, we are bypassing
this step
        */

        if($toEmail == $email ){

            $msg = 'Your email has been sent';
            $msgClass = 'alert-success';

        } else {

            $msg = 'Your email was not sent';
            $msgClass = 'alert-danger';

        }

    } else {

```

```

    $msg = 'Please fill in all fields';
    $msgClass = 'alert-danger';
}

} //End of Contact form Handler

include 'footer.php';?>

```

16. Κώδικας για τη συλλογή δεδομένων της σελίδας επεξεργασίας προφίλ για τους χρήστες

```

<?php
include 'head.php';
include("auth.php");
?>

<?php
//getting id from url
$id = $_GET['id'];
//selecting data associated with this id
$result = mysqli_query($mysqli, "SELECT * FROM usertab WHERE
ID=$id");

while($res = mysqli_fetch_array($result)){
    $pass = $res['pass'];
    $email = $res['email'];
    $type = $res['type'];
}
//freeing the var
mysqli_free_result($result);

```

17. Κώδικας για τη συλλογή δεδομένων της σελίδας επεξεργασίας προφίλ για τον υποψήφιο εργαζόμενο

```

//if there is no record in the DB for this id and the type is
worker/company we create one
if( $type == 'worker'){
    $resultwork = mysqli_query($mysqli, "SELECT * FROM worker WHERE
wid=$id") or die(mysqli_error());
    $rowcount = mysqli_num_rows($resultwork);

    if($rowcount > 0){

        while($reswork = mysqli_fetch_array($resultwork)){
            $wfname = $reswork['wname'];

```

```

        $wlname = $reswork['wlname'];
        $wage = $reswork['wAge'];
        $waddress = $reswork['waddress'];
        $wtel = $reswork['wtel'];
        $wgradlvl = $reswork['wgradlvl'];
        $wexp = $reswork['wexp'];
        $winfo = $reswork['winfo'];
        //making the strings to arrays using "*" as a delimiter using
explode
        $skills = $reswork['wskills'];
        $skill = explode("*", $skills);
        $langs = $reswork['wlang'];
        $lang = explode("*", $langs);
    }
    mysqli_free_result($resultwork);

}

    $resultworkin = mysqli_query($mysqli, "INSERT INTO worker(wid)
VALUES('$id')");
    mysqli_free_result($resultworkin);

}

}elseif( $type == 'company'){

```

18. Κώδικας για τη συλλογή δεδομένων της σελίδας επεξεργασίας προφίλ για την επιχείρηση

```

}elseif( $type == 'company'){

    $resultcomp = mysqli_query($mysqli, "SELECT * FROM comptab WHERE
comptab.cid = " . $id);
    $rowcount = mysqli_num_rows($resultcomp);

    if($rowcount > 0){

        while($rescomp = mysqli_fetch_array($resultcomp)){
            $cname = $rescomp['cname'];
            $cabout = $rescomp['cabout'];
            $caddress = $rescomp['caddress'];
            $ctel = $rescomp['ctel'];
        }
        mysqli_free_result($resultcomp);

    }

}elseif

```

```

        $resultcompin = mysqli_query($mysqli, "INSERT INTO comptab(cid)
VALUES ('$id')");
        mysqli_free_result($resultcompin);

    }
} //End user finder
?>

```

## 19. Κώδικας φόρμας επεξεργασίας προφίλ χρήστη

```

<title>Edit Data</title>

<div class="edit container">

    <div class="customform">

        <form name="form1" method="POST" action="edit.php">

            <fieldset>

                <legend class="text-center">Here You can edit Your Profile!</legend>
                <div class="form-group">
                    <label for="Email">Email address</label>
                    <input type="text" name="email" class="form-control" id="Email"
placeholder="Enter email" value="<?php echo $email; ?>" required>
                </div>

                <div class="form-group">
                    <label for="Password">Password</label>
                    <input type="password" name="pass" class="form-control"
id="Password" placeholder="Password" value="<?php echo $pass;?>" required>
                </div>

                <label for="radio">Type</label>
                <fieldset class="form-group" id="radio">
                    <div class="form-check">
                        <label class="form-check-label radio-inline">
                            <input type="radio" class="form-check-input" name="type"
id="option1" value="company" <?php if (isset($type) && $type=="company") echo
"checked";?>>Company
                        </label>
                    </div>

                    <div class="form-check">
                        <label class="form-check-label radio-inline">
                            <input type="radio" class="form-check-input" name="type"
id="option2" value="worker" <?php if (isset($type) && $type=="worker") echo
"checked";?>>Worker
                        </label>
                    </div>
                </fieldset>
            </fieldset>
        </form>
    </div>
</div>

```

```
</fieldset>
```

## 20. Κώδικας φόρμας επεξεργασίας προφίλ όταν ο χρήστης είναι υποψήφιος εργαζόμενος

```
<?php
    if($type == "worker"){ ?>

        <div class="form-group">
            <label for="name">First Name</label>
            <input type="text" name="wname" class="form-control" id="name"
placeholder="Enter your First Name" value="<?php echo $wfname; ?>" required>
        </div>

        <div class="form-group">
            <label for="lname">Last Name</label>
            <input type="text" name="wlname" class="form-control" id="lname"
placeholder="Enter your Last Name" value="<?php echo $wlname; ?>" required>
        </div>

        <label for="radio">Graduation Level</label>

        <fieldset class="form-group" id="radio">

            <div class="form-check">
                <label class="form-check-label">
                    <input type="radio" class="form-check-input" name="wgradlvl"
id="option1" value="under" <?php if (isset($wgradlvl) && $wgradlvl== 'under')
echo "checked";?>>Undergraduate
                </label>
            </div>

            <div class="form-check">
                <label class="form-check-label">
                    <input type="radio" class="form-check-input" name="wgradlvl"
id="option2" value="bachelor" <?php if (isset($wgradlvl) && $wgradlvl==
'bachelor') echo "checked";?>>Bachelor
                </label>
            </div>

            <div class="form-check">
                <label class="form-check-label">
                    <input type="radio" class="form-check-input" name="wgradlvl"
id="option2" value="master" <?php if (isset($wgradlvl) && $wgradlvl== 'master')
echo "checked";?>>Master
                </label>
            </div>

        </fieldset>
    }
}
```

```

<div class="form-group">
  <label for="Textarea">Basic info</label>
  <textarea name="winfo" class="form-control" id="Textarea"
rows="3"><?php echo $winfo; ?></textarea>
</div>

<div class="form-group">

  <label for="wAge">Age</label>

  <?php //fixing the date format to be able to be shown
  $fwage = date("Y-m-d", strtotime($wage));
  $wage = $fwage; ?>

  <input type="date" name="wAge" class="form-control" id="wAge"
placeholder="Age" value="<?php echo $wage;?>" required>

</div>

<div class="form-group">
  <label for="waddress">Address</label>
  <input type="text" name="waddress" class="form-control"
id="waddress" placeholder="Enter your address" value="<?php echo $waddress;?>"
required>
</div>

<div class="form-group">
  <label for="wtel">Phone</label>
  <input type="number" name="wtel" class="form-control" id="wtel"
placeholder="telephone" value="<?php echo $wtel;?>" required>
</div>

<div class="form-group">
  <label for="wexp">Your work Experience</label>
  <input type="number" name="wexp" class="form-control" id="wexp"
placeholder="Months" value="<?php echo $wexp;?>" required>
</div>

<label for="wskills">Your Skills</label>
<div class="form-group">

  <div id="inputs">

    <?php foreach ($skill as $sk): ?>
      <input type="text" name="skills[]" class="form-control"
id="skills" placeholder="Enter a skill" value="<?php echo $sk; ?>">
      <?php endforeach; ?>

    </div>
  </div>

```

```

        <div class="plusminus">
            <a class="btn" id="adderr" href="#"><i class="fa fa-plus-circle
fa-2x" aria-hidden="true" style="vertical-align: middle;color:#0000a5;"></i></a>
            <a class="btn" id="remover" href="#"><i class="fa
fa-minus-circle fa-2x" aria-hidden="true" style="vertical-align:
middle;color:#c30000;"></i></a>
        </div>

</div>

<label for="lang">Known Languages</label>
<div class="form-group">

    <div id="inputslang">
        <?php foreach ($lang as $la): ?>
            <input type="text" name="languages[]" class="form-control"
id="lang" placeholder="Enter a language" value="<?php echo $la;?>">
        <?php endforeach; ?>
    </div>

    <div class="plusminus">
        <a class="btn" id="adderrlang" href="#"><i class="fa
fa-plus-circle fa-2x" aria-hidden="true" style="vertical-align:
middle;color:#0000a5;"></i></a>
        <a class="btn" id="removelang" href="#"><i class="fa
fa-minus-circle fa-2x" aria-hidden="true" style="vertical-align:
middle;color:#c30000;"></i></a>
    </div>

</div>

<?php }elseif($type == 'company'){ ?>

```

## 21. Κώδικας φόρμας επεξεργασίας προφίλ όταν ο χρήστης είναι εταιρεία

```

<?php }elseif($type == 'company'){ ?>

    <div class="form-group">
        <label for="cname">Company Name</label>
        <input type="text" name="cname" class="form-control" id="cname"
placeholder="Enter the Name of the Company" value="<?php echo $cname; ?>"
required>
    </div>

    <div class="form-group">
        <label for="cabout">About the Company</label>
        <textarea name="cabout" class="form-control" id="cabout"
rows="3"><?php echo $cabout; ?></textarea>
    </div>

```



```

        </div>

        <div class="form-group">
            <label for="address">Address</label>
            <input type="text" name="address" class="form-control"
id="address" placeholder="Enter the Address" value="<?php echo $address;?"
required>
        </div>

        <div class="form-group">
            <label for="ctel">Phone Number</label>
            <input type="text" name="ctel" class="form-control" id="ctel"
placeholder="telephone" value="<?php echo $ctel;?" required>
        </div>

        <?php } ?>
        <!-- Saving the id for after the submission-->
        <input type="hidden" name="id" value="<?php echo $_GET['id'];?>>

        <button type="update" name="update" class="btn btn-outline-primary
custombtn">Submit</button>

    </fieldset>
</form><!-- End of Form -->

</div><!-- End of customform -->

```

## 22. Κώδικας υποβολής στοιχείων επεξεργασίας προφίλ χρήστη στη βάση δεδομένων

```

<?php
//Form Handler
if(isset($_POST['update'])) {

    $id = $_POST['id'];

    //making sure for the content,by removing slashes,preventing sql
injections and preparing the statements
    $email = stripslashes($_POST['email']);
    $email = mysqli_real_escape_string($mysqli,$email);

    //Not hashing the password for convenience
    $pass = stripslashes($_POST['pass']);
    $pass = mysqli_real_escape_string($mysqli,$pass);

    $type = $_POST['type'];

    // checking empty fields
    if(empty($pass) || empty($email) || empty($type)) {

```

```

        if(empty($pass)) { ?>
            <div class="alert alert-dismissible alert-danger customalert">
                <button type="button" class="close"
data-dismiss="alert">&times;</button>
                <strong>Password field is empty.</strong> <a
href="javascript:self.history.back();" class="alert-link"> try submitting
again.</a>
            </div>
        <?php }

        if(empty($email)) { ?>

            <div class="alert alert-dismissible alert-danger customalert">
                <button type="button" class="close"
data-dismiss="alert">&times;</button>
                <strong>Email field is empty.</strong> <a
href="javascript:self.history.back();" class="alert-link">try submitting
again.</a>
            </div>
        <?php }

        if(empty($type)) { ?>
            <div class="alert alert-dismissible alert-danger customalert">
                <button type="button" class="close"
data-dismiss="alert">&times;</button>
                <strong>Type field is empty.</strong> <a
href="javascript:self.history.back();" class="alert-link">try submitting
again.</a>
            </div>
        <?php }

    }else{

        //updating the main user table
        $result = mysqli_query($mysqli, "UPDATE usertab SET
email='$email',pass='$pass',type='$type' WHERE ID=$id");

    }

```

23. Κώδικας υποβολής στοιχείων επεξεργασίας προφίλ για τους υποψήφιους εργαζόμενου στη βάση δεδομένων

```

if($type == 'worker') {

    //making sure of the content,by removing slashes
    $wfname = stripslashes($_POST['wfname']);
    $wfname = mysqli_real_escape_string($mysqli,$wfname);

    $wlname = stripslashes($_POST['wlname']);

```

```

$wlname = mysqli_real_escape_string($mysqli,$wlname);

$wgradlvl = stripslashes($_POST['wgradlvl']);
$wgradlvl = mysqli_real_escape_string($mysqli,$wgradlvl);

$winfo = stripslashes($_POST['winfo']);
$winfo = mysqli_real_escape_string($mysqli,$winfo);

$wage = stripslashes($_POST['wAge']);
$wage = mysqli_real_escape_string($mysqli,$wage);

$waddress = stripslashes($_POST['waddress']);
$waddress = mysqli_real_escape_string($mysqli,$waddress);

$wtel = stripslashes($_POST['wtel']);
$wtel = mysqli_real_escape_string($mysqli,$wtel);

$wexp = stripslashes($_POST['wexp']);
$wexp = mysqli_real_escape_string($mysqli,$wexp);

//making the arrays to a string separated by "*" using implode
$skills = implode(" ", $_POST['skills']);
$skills = stripslashes($skills);
$skills = mysqli_real_escape_string($mysqli,$skills);

$langs = implode(" ", $_POST['languages']);
$langs = stripslashes($langs);
$langs = mysqli_real_escape_string($mysqli,$langs);

//updating the worker table
$resultwork = mysqli_query($mysqli,"UPDATE `worker` SET
`wname`=`$wfname`,`wlname`=`$wlname`,`wgradlvl`=`$wgradlvl`,`winfo`=`$winfo`,`wA
ge`=`$wage`,

`waddress`=`$waddress`,`wtel`=`$wtel`,`wexp`=`$wexp`,`wskills`=`$skills`,`wlang`
=`$langs` WHERE `wid` = '$id'");

if($resultwork && $result){ ?>
    <div class="alert alert-dismissable alert-success customalert">
        <button type="button" class="close"
data-dismiss="alert">&times;</button>
        <strong>Success, Worker profile completed</strong><a
href="profile.php" class="alert-link">Go to your profile</a>
    </div>

    <!--Script to remove the form after submission -->
    <script>
        var element = document.querySelector('.customform');

```

```

        removeJunk(element);
    </script>
<?php }else{ ?>

    <div class="alert alert-dismissable alert-success customalert">
        <button type="button" class="close"
data-dismiss="alert">&times;</button>
        <strong>DataBase Error!Try again later.</strong> <a
href="profile.php" class="alert-link">Go to your profile</a>
    </div>

    <!--Script to remove the form after submission -->
    <script>
        var element = document.querySelector('.customform');
removeJunk(element);
    </script>

    <?php }
    mysqli_free_result($resultwork);

}elseif ($type == 'company') {

```

#### 24. Κώδικας υποβολής στοιχείων επεξεργασίας προφίλ για τις εταιρείες στη βάση δεδομένων

```

}elseif ($type == 'company') {

    $cname = stripslashes($_POST['cname']);
    $cname = mysqli_real_escape_string($mysqli,$cname);

    $cabout = stripslashes($_POST['cabout']);
    $cabout = mysqli_real_escape_string($mysqli,$cabout);

    $caddress = stripslashes($_POST['caddress']);
    $caddress = mysqli_real_escape_string($mysqli,$caddress);

    $ctel = stripslashes($_POST['ctel']);
    $ctel = mysqli_real_escape_string($mysqli,$ctel);

    $resultcomp = $mysqli->query("UPDATE comptab SET
cname='$cname',cabout='$cabout',caddress='$caddress',ctel='$ctel' WHERE
cid=$id");

    if($resultcomp && $result ){ ?>

        <div class="alert alert-dismissable alert-success customalert">
            <button type="button" class="close"
data-dismiss="alert">&times;</button>
            <strong>Success, Company profile completed</strong> <a

```

```

href="profile.php" class="alert-link">Go to your profile</a>
    </div>

    <!--Script to remove the form after submission -->
    <script>
    var element = document.querySelector('.customform');
        removeJunk(element);
    </script>

    <?php }else{ ?>

        <div class="alert alert-dismissable alert-success
customalert">
            <button type="button" class="close"
data-dismiss="alert">&times;</button>
            <strong>DataBase Error!Try again later.</strong> <a
href="profile.php" class="alert-link">Go to your profile</a>
        </div>

        <!--Script to remove the form after submission -->
        <script>
        var element = document.querySelector('.customform');
            removeJunk(element);
        </script>

        <?php }
        mysqli_free_result($resultcomp);

    }else{

        if($result){ ?>

            <div class="alert alert-dismissable alert-success
customalert">
                <button type="button" class="close"
data-dismiss="alert">&times;</button>
                <strong>Success, but your Profile is not Full yet</strong>
                <a href="profile.php" class="alert-link">Go to your profile</a>
            </div>

            <!--Script to remove the form after submission -->
            <script>
            var element = document.querySelector('.customform');
                removeJunk(element);
            </script>

            <?php }else{ ?>

                <div class="alert alert-dismissable alert-success
customalert">
                    <button type="button" class="close"

```

```

data-dismiss="alert">&times;</button>
    <strong>DataBase Error!Try again later.</strong> <a
href="profile.php" class="alert-link">Go to your profile</a>
</div>

<!--Script to remove the form after submission -->
<script>
    var element = document.querySelector('.customform');
    removeJunk(element);
</script>

<?php }

} //End Company/Worker if
mysqli_free_result($result);
} //End Form Handler ?>
</div><!-- Closing Container -->
<?php include 'footer.php';?>

```

## 25. Κώδικας φόρμας δημιουργίας νέας αγγελίας για θέση εργασίας

```

<?php
//including the header and the authenticator files
include 'head.php';
include("auth.php");
?>

<?php
//Getting the id from the URL
$cid = $_GET['id']; ?>

<title>Create New Job Offer</title>

<div class="add container">

<div class="customform">

<form action="createjob.php" method="post" name="form1">

<fieldset>
<legend class="text-center">Add a new job Here!</legend>

<div class="form-group">
<label for="Jobtitle">Job title</label>
<input type="text" name="jtittle" class="form-control"
id="Jobtitle" placeholder="Enter the title" required>
</div>

<div class="form-group">

```

```

        <label for="Textarea">Job information</label>
        <textarea name="jinfo" class="form-control" id="Textarea"
rows="3"></textarea>
    </div>

    <label for="radio">Graduation Level Needed</label>
    <fieldset class="form-group" id="radio">
        <div class="form-check">
            <label class="form-check-label">
                <input type="radio" class="form-check-input"
name="jgradlvl" id="option1" value="under">Undergraduate
            </label>
        </div>
        <div class="form-check">
            <label class="form-check-label">
                <input type="radio" class="form-check-input"
name="jgradlvl" id="option2" value="bachelor">Bachelor
            </label>
        </div>

        <div class="form-check">
            <label class="form-check-label">
                <input type="radio" class="form-check-input"
name="jgradlvl" id="option2" value="master">Master
            </label>
        </div>

    </fieldset>

    <div class="form-group">
        <label for="Jobexp">Needed job Experience</label>
        <input type="number" name="jexp" class="form-control"
id="Jobexp" placeholder="Months" required>
    </div>

    <label for="Jobskills">Job Skills</label>
    <div class="form-group">

        <div id="inputs">
            <input type="text" name="skills[]" class="form-control"
id="Jobskills" placeholder="Enter a skill" required>
            <?php foreach ($skills as $sk): ?>
                <input type="text" name="skills[]" class="form-control"
id="Jobskills" placeholder="Enter a skill">
            <?php endforeach; ?>

        </div>

        <div class="plusminus">
            <a class="btn" id="add" href="#"><i class="fa
fa-plus-circle fa-2x" aria-hidden="true" style="vertical-align:

```

```

middle;color:#0000a5;"></i></a>
        <a class="btn" id="remove" href="#"><i class="fa
fa-minus-circle fa-2x" aria-hidden="true" style="vertical-align:
middle;color:#c30000;"></i></a>
    </div>

</div>

<label for="Joblang">Known Languages</label>
<div class="form-group">

    <div id="inputslang">
        <input type="text" name="languages[]" class="form-control"
id="Joblang" placeholder="Enter a language" required>
        <?php foreach ($lang as $la): ?>
            <input type="text" name="languages[]" class="form-control"
id="Joblang" placeholder="Enter a language">
        <?php endforeach; ?>
    </div>

    <div class="plusminus">
        <a class="btn" id="addlang" href="#"><i class="fa
fa-plus-circle fa-2x" aria-hidden="true" style="vertical-align:
middle;color:#0000a5;"></i></a>
        <a class="btn" id="removelang" href="#"><i class="fa
fa-minus-circle fa-2x" aria-hidden="true" style="vertical-align:
middle;color:#c30000;"></i></a>
    </div>

</div>

<div class="form-group">
    <label for="Jobloc">Job location</label>
    <input type="text" name="jloc" class="form-control"
id="Jobloc" placeholder="Enter the location" required>
</div>

<!-- Store cid in form -->
<input type="hidden" name="cid" value="<?php echo $cid ?>">

<button type="submit" name="Submit" class="btn
btn-outline-primary custombtn">Submit</button>

</fieldset>

</form><!-- Closing the Form -->

</div><!-- Closing the customform div -->

```



```

// Checking if the Form is Submitted
    if(isset($_POST['Submit'])) {

        //Escaping hazardous symbols for the db and preparing to insert to the
db
        $jtittle = stripslashes($_POST['jtittle']);
        $jtittle = mysqli_real_escape_string($mysqli,$jtittle);

        $jgradlvl = stripslashes($_POST['jgradlvl']);
        $jgradlvl = mysqli_real_escape_string($mysqli,$jgradlvl);

        $jinfo = stripslashes($_POST['jinfo']);
        $jinfo = mysqli_real_escape_string($mysqli,$jinfo);

        $jloc = stripslashes($_POST['jloc']);
        $jloc = mysqli_real_escape_string($mysqli,$jloc);

        $jexp = stripslashes($_POST['jexp']);
        $jexp = mysqli_real_escape_string($mysqli,$jexp);

        $cid = stripslashes($_POST['cid']);
        $cid = mysqli_real_escape_string($mysqli,$cid);

        $skills = implode(" ", $_POST['skills']);
        $skills = stripslashes($skills);
        $skills = mysqli_real_escape_string($mysqli,$skills);

        $langs = implode(" ", $_POST['languages']);
        $langs = stripslashes($langs);
        $langs = mysqli_real_escape_string($mysqli,$langs);

        //Inserting the data to the db
        $result = $mysqli->query("INSERT INTO jobOffer (jtittle, jinfo,
jskills, jgradlvl, jexp, jlang, jloc, cid) VALUES
('$jtittle','$jinfo','$skills','$jgradlvl','$jexp','$langs','$jloc','$cid')");

        //Checking if the data were inserted and displaying success message
        if($result){?>

            <div class="alert alert-dismissable alert-success customalert">
                <button type="button" class="close"
data-dismiss="alert">&times;</button>
                <strong>Offer created successfully.</strong> <a href="profile.php"
class="alert-link">Go to your profile</a>
            </div>

            <!-- Script to remove the form after it is Submitted -->
            <script>
                var element = document.querySelector('.customform');
                removeJunk(element);
            </script>

```

```

        <?php }else{ ?>

            <!-- //Checking if the data were not inserted and displaying warning
message -->
            <div class="alert alert-dismissible alert-success customalert">
                <button type="button" class="close"
data-dismiss="alert">&times;</button>
                <strong>DataBase Error!Try again later.</strong> <a
href="profile.php" class="alert-link">Go to your profile</a>
            </div>

            <!-- Script to remove the form after it is Submitted -->
            <script>
                var element = document.querySelector('.customform');
                removeJunk(element);
            </script>

        <?php }

    }//Closing the form Handler
    mysqli_free_result($result);
    ?>

</div><!-- Closing the Container -->
<?php include 'footer.php';?>

```

## 26. Κώδικας προβολής αγγελίας εργασίας

```

<?php
//including the database connection and authentication files
include 'head.php';
include("auth.php");

//collecting data from the URL
$jid = $_GET['jid'];
$id = $_GET['id'];
$sec = $_GET['sec'];

//fetching the data from th db for the id
$result = mysqli_query($mysqli, "SELECT * FROM usertab WHERE id='$id'") or
die(mysqli_error());
?>

<title>Job Offer</title>

    <div class="profile container">

        <div class="title"><h3>Job Offer</h3></div>

```

```

<table class="table table-hover csthtable">

    <?php

    while($res = mysqli_fetch_array($result)) {
        $email = $res['email'];
        $pass = $res['pass'];
        $type = $res['type'];
    } ?>

    <?php $resultoffer = mysqli_query($mysqli, "SELECT * FROM jobOffer
WHERE jid = '$jid'") or die(mysqli_error());

    while($resoffer = mysqli_fetch_array($resultoffer)){ ?>

        <tr>
            <th>Job Title</th>
            <?php echo "<td>".$resoffer['jtittle']."</td>"; ?>
        </tr>

        <tr>
            <th>Basic info</th>
            <?php echo "<td>".$resoffer['jinfo']."</td>"; ?>
        </tr>

        <tr>
            <th>Job Location</th>
            <?php echo "<td>".$resoffer['jloc']."</td>"; ?>
        </tr>

        <tr>
            <th>Graduation Level Needed</th>
            <?php echo "<td>".$resoffer['jgradlvl']."</td>"; ?>
        </tr>

        <tr>
            <th>Work Experience Needed</th>
            <?php echo "<td>".$resoffer['jexp']." Months</td>"; ?>
        </tr>

        <tr>
            <th>Required Languages</th>
            <?php echo "<td>".$resoffer['jlang']."</td>"; ?>
        </tr>

        <tr>
            <th>Skills Required</th>
            <?php echo "<td>".$resoffer['jskills']."</td>"; ?>
        </tr>

```

```

        <?php $cid = $resoffer['cid'];
    } ?>

    <?php
        $companymail = mysqli_query($mysqli, "SELECT email FROM
usertab WHERE ID = '$cid'") or die(mysqli_error());
        while($cmail = mysqli_fetch_array($companymail)){
            $mail = $cmail['email'];
        }
        $resultcomp = mysqli_query($mysqli, "SELECT * FROM comptab
WHERE cid = '$cid'") or die(mysqli_error());
        while($rescomp = mysqli_fetch_array($resultcomp)){ ?>

            <tr>
                <th>Company Name</th>
                <?php echo "<td>".$rescomp['cname']."</td>"; ?>
            </tr>

            <tr>
                <th>About the Company</th>
                <?php echo "<td>".$rescomp['cabout']."</td>"; ?>
            </tr>

            <tr>
                <th>Address</th>
                <?php echo "<td>".$rescomp['caddress']."</td>"; ?>
            </tr>

            <tr>
                <th>Phone</th>
                <?php echo "<td>".$rescomp['ctel']."</td>"; ?>
            </tr>

            <tr>
                <th>Email</th>
                <?php echo "<td>".$mail."</td>"; ?>
            </tr>

        <?php
    }
    if( ($type === 'company') && ($cid === $id) && ($sec != 0)){?>

        <tr>
            <th>Edit the offer</th>
            <?php echo "<td><a
href=\"editoffer.php?jid=$jid\">Edit</a> | <a href=\"delete.php?jid=$jid\"
onClick=\"return confirm('Are you sure you want to delete
this?')\">Delete</a></td>"; ?>
        </tr>

```

```

        <?php }else {
            echo "<td> </td>";
            echo "<td><a href=\"profile.php?id=$cid\">Visit The
Company</a></td>";
        }
        mysqli_free_result($result);
        mysqli_free_result($resultcomp);
        mysqli_free_result($companymail);

        ?>

</table>

<?php
//Checking if the user is owner or visitor
if( ($type === 'company') && ($cid === $id) && ($sec != 0)){?>
    <a class="btn costbtn btn-outline-primary"
href="automach.php?jid=<?php echo $jid; ?>">Show Me Workers</a>

<?php }else{ ?>

    <div class="container contForm" id="contForm">

        <div class="panel-group">

            <div class="panel panel-default">

                <div class="panel-heading">
                    <h4 class="panel-title">
                        <a data-toggle="collapse" class="btn btn-primary
btn-lg btn-block" href="#collapse1">Contact</a>
                    </h4>
                </div>

                <div id="collapse1" class="panel-collapse collapse">
                    <div class="panel-body">

                        <?php if($msg != ''): ?>
                            <div class="alert <?php echo $msgClass; ?>"><?php
echo $msg; ?></div>
                        <?php endif; ?>

                        <form method="post" class="contForm" action="<?php
echo $_SERVER['PHP_SELF']; ?>">

                            <div class="form-group">
                                <label>Name</label>
                                <input type="text" name="name"
class="form-control" value="<?php echo isset($_POST['name']) ? $name : ''; ?>">
                            </div>

```

```

        <div class="form-group">
            <label>Email</label>
            <input type="email" name="email"
class="form-control" value="<?php echo isset($_POST['email']) ? $senderEmail :
''; ?>">

        </div>

        <div class="form-group">
            <label>Message</label>
            <textarea name="message"
class="form-control"><?php echo isset($_POST['message']) ? $message : '';
?></textarea>

        </div>

        <button type="submit" name="submit" class="btn
btn-primary">Submit</button>

    </form><!-- End of contact form -->

</div><!-- Closing the panel body div -->

</div><!-- Closing the panel-collapse div -->

</div><!-- Closing the panel -->

</div><!-- End of Panel -->

</div><!-- End of contact form -->

<?php }//End if visitor//owner ?>

</div><!-- End of container div -->

<?php
//Contact form Handler

// Message Vars
$msg = '';
$msgClass = '';

// Check For Submit
if(isset($_POST['submit'])){
    // Get Form Data
    $name = htmlspecialchars($_POST['name']);
    $senderEmail = htmlspecialchars($_POST['email']);
    $message = htmlspecialchars($_POST['message']);

    // Check Required Fields
    if(!empty($senderEmail) && !empty($name) &&
!empty($message)){

```

```

    $toEmail = $email;
    $subject = 'Contact Request From '.$name;
    $body = '<h2>Contact Request</h2>
        <h4>Name</h4><p>'.$name.'</p>
        <h4>Email</h4><p>'.$senderEmail.'</p>
        <h4>Message</h4><p>'.$message.'</p>
    ';

    // Email Headers
    $headers = "MIME-Version: 1.0" ."\r\n";
    $headers .= "Content-Type:text/html;charset=UTF-8" . "\r\n";

    // Additional Headers
    $headers .= "From: " . $name. "<".$senderEmail.">". "\r\n";

    /*
    * It Should be like that:
    * if(mail($toEmail, $subject, $body, $headers)){
    *     $msg = 'Your email has been sent';
    *     $msgClass = 'alert-success';
    * }
    * But because we don't have a mail service in place, we are
    bypassing this step
    */

    if($toEmail == $email){

        $msg = 'Your email has been sent';
        $msgClass = 'alert-success';

    } else {

        $msg = 'Your email was not sent';
        $msgClass = 'alert-danger';
    }
    } else {
        $msg = 'Please fill in all fields';
        $msgClass = 'alert-danger';
    }
} //End of Contact From Handler

include 'footer.php';?>

```

## 27. Κώδικας σελίδας επεξεργασίας αγγελίας εργασίας

```

<?php
include 'head.php';
include("auth.php");
?>

```

```

<?php
//getting id from url
$jid = $_GET['jid'];

//selecting data associated with this id
$resultoffer = mysqli_query($mysqli, "SELECT * FROM jobOffer WHERE
jid=$jid");

while($resoffer = mysqli_fetch_array($resultoffer)){
    $jtittle = $resoffer['jtittle'];
    $jloc = $resoffer['jloc'];
    $jgradlvl = $resoffer['jgradlvl'];
    $jexp = $resoffer['jexp'];
    $jinfo = $resoffer['jinfo'];
    //making the strings to arrays using "*" as a delimiter using explode
    $skills = $resoffer['jskills'];
    $skill = explode("*", $skills);
    $langs = $resoffer['jlang'];
    $lang = explode("*", $langs);
    $cid = $resoffer['cid'];
}
//freeing the var
mysqli_free_result($resultoffer);
?>

<title>Edit Work Data</title>

<div class="edit container">

    <div class="customform">

        <form name="form1" method="post" action="editoffer.php">
            <fieldset>
                <legend class="text-center">Here You can edit Your Job
Offer!</legend>

                <div class="form-group">
                    <label for="Jobtittle">Job title</label>
                    <input type="text" name="jtittle" class="form-control"
id="Jobtittle" placeholder="Enter the title" value="<?php echo $jtittle; ?>"
required>
                </div>

                <div class="form-group">
                    <label for="Textarea">Job information</label>
                    <textarea name="jinfo" class="form-control" id="Textarea"
rows="3"><?php echo $jinfo; ?></textarea>
                </div>

                <label for="radio">Graduation Level Needed</label>
                <fieldset class="form-group" id="radio">

```



```

        <div class="form-check">
            <label class="form-check-label">
                <input type="radio" class="form-check-input" name="jgradlvl"
id="option1" value="under" <?php if (isset($jgradlvl) && $jgradlvl== 'under')
echo "checked";?>>Undergraduate
            </label>
        </div>

        <div class="form-check">
            <label class="form-check-label">
                <input type="radio" class="form-check-input" name="jgradlvl"
id="option2" value="bachelor" <?php if (isset($jgradlvl) && $jgradlvl==
'bachelor') echo "checked";?>>Bachelor
            </label>
        </div>

        <div class="form-check">
            <label class="form-check-label">
                <input type="radio" class="form-check-input" name="jgradlvl"
id="option2" value="master" <?php if (isset($jgradlvl) && $jgradlvl== 'master')
echo "checked";?>>Master
            </label>
        </div>

    </fieldset>

    <div class="form-group">
        <label for="Jobexp">Needed job Experience</label>
        <input type="number" name="jexp" class="form-control" id="Jobexp"
placeholder="Months" value="<?php echo $jexp;?>" required>
    </div>

    <label for="Jobskills">Job Skills</label>
    <div class="form-group">

        <div id="inputs">

            <?php foreach ($skill as $sk): ?>
                <input type="text" name="skills[]" class="form-control"
id="skills" placeholder="Enter a skill" value="<?php echo $sk; ?>">
            <?php endforeach; ?>

        </div>

        <div class="plusminus">
            <a class="btn" id="adder" href="#"><i class="fa fa-plus-circle
fa-2x" aria-hidden="true" style="vertical-align: middle;color:#0000a5;"></i></a>
            <a class="btn" id="remove" href="#"><i class="fa fa-minus-circle
fa-2x" aria-hidden="true" style="vertical-align: middle;color:#c30000;"></i></a>
        </div>

```

```

</div>

<label for="lang">Known Languages</label>
<div class="form-group">

    <div id="inputslang">

        <?php foreach ($lang as $la): ?>
            <input type="text" name="languages[]" class="form-control"
id="lang" placeholder="Enter a language" value="<?php echo $la;?>">
        <?php endforeach; ?>

    </div>

    <div class="plusminus">
        <a class="btn" id="addlang" href="#"><i class="fa
fa-plus-circle fa-2x" aria-hidden="true" style="vertical-align:
middle;color:#0000a5;"></i></a>
        <a class="btn" id="removelang" href="#"><i class="fa
fa-minus-circle fa-2x" aria-hidden="true" style="vertical-align:
middle;color:#c30000;"></i></a>
    </div>

</div>

<div class="form-group">
    <label for="Jobloc">Job location</label>
    <input type="text" name="jloc" class="form-control" id="Jobloc"
placeholder="Enter the location" value="<?php echo $jloc;?>" required>
</div>

<!-- Store cid and jid in form -->
<input type="hidden" name="cid" value="<?php echo $cid ?>">
<input type="hidden" name="jid" value="<?php echo $_GET['jid'];?>">

<button type="update" name="update" class="btn btn-outline-primary
custombtn">Submit</button>

</fieldset>

</form><!-- End of Form -->

</div><!-- Closing customform div -->

<?php
//Form Handler
if(isset($_POST['update'])) {

    //making sure for the content,by removing slashes,preventing sql

```

```

injections and preparing the statements
    $jid = $_POST['jid'];

    $jtittle = stripslashes($_POST['jtittle']);
    $jtittle = mysqli_real_escape_string($mysqli,$jtittle);

    $jgradlvl = stripslashes($_POST['jgradlvl']);
    $jgradlvl = mysqli_real_escape_string($mysqli,$jgradlvl);

    $jinfo = stripslashes($_POST['jinfo']);
    $jinfo = mysqli_real_escape_string($mysqli,$jinfo);

    $jloc = stripslashes($_POST['jloc']);
    $jloc = mysqli_real_escape_string($mysqli,$jloc);

    $jexp = stripslashes($_POST['jexp']);
    $jexp = mysqli_real_escape_string($mysqli,$jexp);

    $cid = stripslashes($_POST['cid']);
    $cid = mysqli_real_escape_string($mysqli,$cid);

    //making the arrays to a string separated by "*" using implode
    $skills = implode("*", $_POST['skills']);
    $skills = stripslashes($skills);
    $skills = mysqli_real_escape_string($mysqli,$skills);

    $langs = implode("*", $_POST['languages']);
    $langs = stripslashes($langs);
    $langs = mysqli_real_escape_string($mysqli,$langs);

    $resultoffer = mysqli_query($mysqli,"UPDATE `jobOffer` SET
`jtittle`=`$jtittle`,`jinfo`=`$jinfo`,`jskills`=`$skills`,`jgradlvl`=`$jgradlvl`
,`jexp`=`$jexp`,
    `jlang`=`$langs`,`jloc`=`$jloc` WHERE `jid` = '$jid'");

    if($resultoffer){
        //if the data were inserted to the db display success message ?>
        <div class="alert alert-dismissable alert-success customalert">
            <button type="button" class="close"
data-dismiss="alert">&times;</button>
            <strong>Offer updated successfully.</strong> <a
href="joboffer.php?jid=<?php echo $jid;?>&id=<?php echo $cid;?>"
class="alert-link">View job Offer</a>
        </div>

        <!--Script to remove the form after submission -->
        <script>
            var element = document.querySelector('.customform');
            removeJunk(element);
        </script>
    <?php }else { ?>

```

```

        <div class="alert alert-dismissible alert-success customalert">
            <button type="button" class="close"
data-dismiss="alert">&times;</button>
            <strong>DataBaseError!</strong> <a
href="javascript:self.history.back();" class="alert-link">Try again.</a>
        </div>

        <!--Script to remove the form after submission -->
        <script>
            var element = document.querySelector('.customform');
            removeJunk(element);
        </script>
        <?php }
    }//End of form Handler
    mysqli_free_result($resultoffer);
    ?>

</div><!-- Closing the Container -->
<?php include 'footer.php';?>

```

## 28. Κώδικας αυτόματης αντιστοίχισης υποψηφίων εργαζομένων και θέσεων εργασίας

```

<?php
include 'head.php';

//Getting the user or job id from the URL
$id = $_GET['id'];
$jid = $_GET['jid'];

?>

<div class="container">

    <?php
    $points[] = 0;

    //Checking what was inserted user or job id
    if(isset($jid)){

        //Collecting the necessary data from the db
        $resultoffer = mysqli_query($mysqli, "SELECT * FROM jobOffer WHERE jid =
'$jid'");
        while ($resoffer = mysqli_fetch_array($resultoffer)) {
            $jskills = str_replace(" ", "", $resoffer['jskills']);
            $jskills = strtolower($jskills);
            $jskills = explode(" ", $jskills);

            $jlangs = str_replace(" ", "", $resoffer['jlang']);

```

```

$jlangs = strtolower($jlangs);
$jlangs = explode(" ", $jlangs);

$jloc = str_replace(" ", "", $resoffer['jloc']);
$jloc = strtolower($jloc);

$jgradlvl = str_replace(" ", "", $resoffer['jgradlvl']);
$jgradlvl = strtolower($jgradlvl);

$jexp = str_replace(" ", "", $resoffer['jexp']);
$jexp = strtolower($jexp);

$cid = $resoffer['cid'];
}

//Collecting the necessary data from the db and putting the in a table
$j=0;
$result = mysqli_query($mysqli, "SELECT * FROM worker");
while($res = mysqli_fetch_array($result)){
    $wid[$j] = $res['wid'];

    //Removing the Spaces and making alla the letters lowercase
    $wskills[$j] = $res['wskills'];
    $wskills[$j] = str_replace(" ", "", $wskills[$j]);
    $wskills[$j] = strtolower($wskills[$j]);

    $wlangs[$j] = $res['wlang'];
    $wlangs[$j] = str_replace(" ", "", $wlangs[$j]);
    $wlangs[$j] = strtolower($wlangs[$j]);

    $waddress[$j] = $res['waddress'];
    $waddress[$j] = str_replace(" ", "", $waddress[$j]);
    $waddress[$j] = strtolower($waddress[$j]);

    $wgradlvl[$j] = $res['wgradlvl'];
    $wgradlvl[$j] = str_replace(" ", "", $wgradlvl[$j]);
    $wgradlvl[$j] = strtolower($wgradlvl[$j]);

    $wexp[$j] = $res['wexp'];
    $wexp[$j] = str_replace(" ", "", $wexp[$j]);
    $wexp[$j] = strtolower($wexp[$j]);

    $wlname[$j] = $res['wlname'];
    $wname[$j] = $res['wname'];

    $j++;
}

//Making the comparison and and sorting the other tables based on the table
$points[]

```

```

$n = 0;
while ($n < $j) {

    $wlangs[$n] = explode(";", $wlangs[$n]);
    foreach ($wlangs[$n] as $wlang) {

        if( in_array($wlang , $jlangs)){
            $points[$n]++;
        }
    }

    $wskills[$n] = explode(";", $wskills[$n]);
    foreach ($wskills[$n] as $wskill) {

        if( in_array($wskill , $jskills)){
            $points[$n]++;
        }
    }

    if($waddress[$n] == $jloc){
        $points[$n]++;
    }

    if ($wgradlvl[$n] == $jgradlvl) {
        $points[$n]++;
    }

    if ($wexp[$n] >= $jexp) {
        $points[$n]++;
    }

    $n = $n + 1;
}

array_multisort($points, SORT_DESC, $wid, $waddress, $wgradlvl, $wname,
$wlname);

?>

<!-- Presenting the results -->
<table class="table table-light table-hover" style="text-align:center;">
    <div class="title" style="text-align:center;"> <h3>Best Workers for
You</h3> </div>
    <?php
        $i = 0;
        while ($i < count($points)) {

```

```

        if ($points[$i] > 0) {
            echo "<tr>";
            echo "<th>".$wname[$i]."</th>";
            echo "<td>".$wlname[$i]."</td>";
            echo "<td> Score = ".$points[$i]."</td>";
            echo "<td><a href=\"profile.php?id=$wid[$i]\">View</a></td>";
            echo "</tr>";

        }else {

            echo "<tr>";
            echo "<th> None Available </th>";
            echo "</tr>";
        }

        $i++;
    }
?>
</table>

<?php
//End if for the job
//Checking if user id exists
}elseif(isset($id)){

    //Collecting the necessary data from the db
$result = mysqli_query($mysqli, "SELECT * FROM usertab WHERE id = '$id'");
while ($res = mysqli_fetch_array($result)) {
    $email = $res['email'];
    $type = $res['type'];
}

//Collecting the necessary data from the db if the user is a worker
if ($type == 'worker') {

    $resultwork = mysqli_query($mysqli, "SELECT * FROM worker WHERE wid = '$id'");
    while ($reswork = mysqli_fetch_array($resultwork)) {

        //Removing the Spaces and making all the letters lowercase
        $wskills = str_replace(" ", "", $reswork['wskills']);
        $wskills = strtolower($wskills);
        $wskills = explode("*", $wskills);

        $wlangs = str_replace(" ", "", $reswork['wlang']);
        $wlangs = strtolower($wlangs);
        $wlangs = explode("*", $wlangs);

        $waddress = str_replace(" ", "", $reswork['waddress']);
        $waddress = strtolower($waddress);

```

```

$wgradlvl = str_replace(" ", "", $reswork['wgradlvl']);
$wgradlvl = strtolower($wgradlvl);

$wexp = str_replace(" ", "", $reswork['wexp']);
$wexp = strtolower($wexp);
}

//Collecting the necessary data from the db and putting the in a table
$j=0;
$resultoffer = mysqli_query($mysqli, "SELECT * FROM jobOffer");
while($res = mysqli_fetch_array($resultoffer)){
    $jid[$j] = $res['jid'];

    $jskills[$j] = $res['jskills'];
    $jskills[$j] = str_replace(" ", "", $jskills[$j]);
    $jskills[$j] = strtolower($jskills[$j]);

    $jlangs[$j] = $res['jlang'];
    $jlangs[$j] = str_replace(" ", "", $jlangs[$j]);
    $jlangs[$j] = strtolower($jlangs[$j]);

    $jloc[$j] = $res['jloc'];
    $jloc[$j] = str_replace(" ", "", $jloc[$j]);
    $jloc[$j] = strtolower($jloc[$j]);

    $jgradlvl[$j] = $res['jgradlvl'];
    $jgradlvl[$j] = str_replace(" ", "", $jgradlvl[$j]);
    $jgradlvl[$j] = strtolower($jgradlvl[$j]);

    $jexp[$j] = $res['jexp'];
    $jexp[$j] = str_replace(" ", "", $jexp[$j]);
    $jexp[$j] = strtolower($jexp[$j]);

    $jtittle[$j] = $res['jtittle'];

    $cid[$j] = $resoffer['cid'];

    $j++;
}

//Making the comparison and and sorting the other tables based on the
table $points[]
$n = 0;
while ($n < $j) {

```



```

$jlangs[$n] = explode(" ", $jlangs[$n]);
foreach ($jlangs[$n] as $jlang) {

    if( in_array($jlang , $wlangs)){
        $points[$n]++;
    }
}

$jskills[$n] = explode(" ", $jskills[$n]);
foreach ($jskills[$n] as $jskill) {

    if( in_array($jskill , $wskills)){
        $points[$n]++;
    }
}

if($jloc[$n] == $waddress){
    $points[$n]++;
}

if ($jgradlvl[$n] == $wgradlvl) {
    $points[$n]++;
}

if ($jexp[$n] <= $wexp) {
    $points[$n]++;
}

    $n = $n + 1;
}

array_multisort($points, SORT_DESC, $jid, $jskills, $jlangs, $jloc,
$jgradlvl, $jexp, $cid);

?>

<!-- Presenting the results -->
<table class="table table-light table-hover" style="text-align:center;">
    <div class="title" style="text-align:center;"> <h3>Best Job for You</h3>
</div>
<?php
    $i = 0;
    while ($i < count($points)) {

        if ($points[$i] > 0) {

```

```

        echo "<tr>";
        echo "<th>".$jtittle[$i]."</th>";
        $jloc[$i] = ucfirst ( $jloc[$i] );
        echo "<td>".$jloc[$i]."</td>";
        echo "<td> Score = ".$points[$i]."</td>";
        echo "<td><a
href=\"joboffer.php?jid=$jid[$i]&id=$id\">View</a></td>";
        echo "</tr>";
    }else {
        echo "<tr>";
        echo "<th> None Available </th>";
        echo "</tr>";
    }

    $i++;
}
?>

</table>

<?php
//Collecting the necessary data from the db if the user is a Company
}
}else {

//searching the all the offers for the current company
$j=0;
$resultoffer = mysqli_query($mysqli, "SELECT * FROM jobOffer WHERE cid =
'$id'");
while($res = mysqli_fetch_array($resultoffer)){
    $jid[$j] = $res['jid'];

//Removing the Spaces and making all the letters lowercase
$jskills[$j] = $res['jskills'];
$jskills[$j] = str_replace(" ", "", $jskills[$j]);
$jskills[$j] = strtolower($jskills[$j]);

$jlangs[$j] = $res['jlang'];
$jlangs[$j] = str_replace(" ", "", $jlangs[$j]);
$jlangs[$j] = strtolower($jlangs[$j]);

$jloc[$j] = $res['jloc'];
$jloc[$j] = str_replace(" ", "", $jloc[$j]);
$jloc[$j] = strtolower($jloc[$j]);

$jgradlvl[$j] = $res['jgradlvl'];
$jgradlvl[$j] = str_replace(" ", "", $jgradlvl[$j]);
$jgradlvl[$j] = strtolower($jgradlvl[$j]);

$jexp[$j] = $res['jexp'];

```

```

    $jexp[$j] = str_replace(" ", "", $jexp[$j]);
    $jexp[$j] = strtolower($jexp[$j]);

    $j++;
}

// Searching all the workers
$w = 0;
$result = mysqli_query($mysqli, "SELECT * FROM worker");
while($res = mysqli_fetch_array($result)){
    $wid[$w] = $res['wid'];

    //Removing the Spaces and making all the letters lowercase
    $wskills[$w] = $res['wskills'];
    $wskills[$w] = str_replace(" ", "", $wskills[$w]);
    $wskills[$w] = strtolower($wskills[$w]);

    $wlangs[$w] = $res['wlang'];
    $wlangs[$w] = str_replace(" ", "", $wlangs[$w]);
    $wlangs[$w] = strtolower($wlangs[$w]);

    $waddress[$w] = $res['waddress'];
    $waddress[$w] = str_replace(" ", "", $waddress[$w]);
    $waddress[$w] = strtolower($waddress[$w]);

    $wgradlvl[$w] = $res['wgradlvl'];
    $wgradlvl[$w] = str_replace(" ", "", $wgradlvl[$w]);
    $wgradlvl[$w] = strtolower($wgradlvl[$w]);

    $wexp[$w] = $res['wexp'];
    $wexp[$w] = str_replace(" ", "", $wexp[$w]);
    $wexp[$w] = strtolower($wexp[$w]);

    $wlname[$w] = $res['wlname'];
    $wname[$w] = $res['wname'];

    $w++;
}

//Making the comparison and sorting the other tables based on the
table $points[]
$i=0;
while($i < $j){

    $jlangs[$i] = explode(" ", $jlangs[$i]);
    $jskills[$i] = explode(" ", $jskills[$i]);

    $n = 0;
    while ($n < $w) {

```

```

$wlangs[$n] = explode("*", $wlangs[$n]);
foreach ($wlangs[$n] as $wlang) {

    if( in_array($wlang , $jlangs[$i])){
        $points[$n]++;
    }
}

$wskills[$n] = explode("*", $wskills[$n]);
foreach ($wskills[$n] as $wskill) {

    if( in_array($wskill , $jskills[$i])){
        $points[$n]++;
    }
}

if($waddress[$n] == $jloc[$i]){
    $points[$n]++;
}

if ($wgradlvl[$n] == $jgradlvl[$i]) {
    $points[$n]++;
}

if ($wexp[$n] >= $jexp[$i]) {
    $points[$n]++;
}

    $n++;
}

    $i++;
}

array_multisort($points, SORT_DESC, $wid, $wname, $wlname);

?>

<!-- Presenting the results -->
<table class="table table-light table-hover" style="text-align:center;">
    <div class="title" style="text-align:center;"> <h3>Best Workers for
You</h3> </div>
    <?php
        $i = 0;
        while ($i < count($points)) {

            if ($points[$i] > 0) {
                echo "<tr>";
                echo "<th>".$wname[$i]."</th>";
                echo "<td>".$wlname[$i]."</td>";
            }
        }
    }

```

```

        echo "<td> Score = ".$points[$i]."</td>";
        echo "<td><a href=\"profile.php?id=$wid[$i]\">View</a></td>";
        echo "</tr>";
    }else {
        echo "<tr>";
        echo "<th> None Available </th>";
        echo "</tr>";
    }

    $i++;
}
?>

</table>

<?php }//end Company if

}//End the job/user if ?>

</div><!--closing Container-->
<?php
//freeing all the vars
mysqli_free_result($resultwork);
mysqli_free_result($resultcomp);
mysqli_free_result($resultoffer);
mysqli_free_result($result);
Include 'footer.php';
?>

```