



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Διαδικτυακή πλατφόρμα διαχείρισης αγγελιών εργασίας (JobFinder) Job Management Online Application (JobFinder)
Όνοματεπώνυμο Φοιτητή	Ευάγγελος Κοροπούλης
Πατρώνυμο	Κυριάκος
Αριθμός Μητρώου	ΜΠΠΛ 13041
Επιβλέπων	Μαρία Βίρβου, Καθηγήτρια

Ημερομηνία Παράδοσης **Φεβρουάριος 2017**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Γεώργιος Τσιχριντζής
Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Ευθύμιος Αλέπης
Επίκουρος Καθηγητής

Πρόεδρος Τμήματος
Πληροφορικής

Περιεχόμενα

Περίληψη / Abstract	6
Σύνομη περιγραφή.....	7
Λειτουργίες ανά χρήστη.....	7
Μοντέλο αρχιτεκτονικής λογισμικού	10
Τεχνολογίες ανάπτυξης	11
Οδηγίες εγκατάστασης JobFinder	12
Πηγαίος κώδικας	14
Συμπεράσματα	16
Βιβλιογραφία – Ηλεκτρονικές πηγές.....	17
ΠΑΡΑΡΤΗΜΑ 1 – Εικόνες στιγμιότυπα JobFinder	18
ΠΑΡΑΡΤΗΜΑ 2 – Πηγαίος κώδικας JobFnder.....	30

Περίληψη

Η εφαρμογή **Unipi JobFinder** είναι μια διαδικτυακή εφαρμογή που λειτουργεί ως πλατφόρμα εύρεσης, δημοσίευσης και διαχείρισης θέσεων εργασίας. Η χρήση της εφαρμογής μπορεί να είναι διαφορετική για τον κάθε επισκέπτη αναλόγως με τον τύπο χρήστη και τις ανάγκες του. Ένας εργαζόμενος που αναζητάει θέσεις εργασίας και ένας εργοδότης που προσφέρει θέσεις εργασίας μπορούν να εξυπηρετηθούν από τις λειτουργίες της εφαρμογής με διαφορετικό τρόπο ο καθένας. Παράλληλα, θεωρούνται και οι δύο απλοί χρήστες σε αντίθεση με τον διαχειριστή της πλατφόρμας. Στην εφαρμογή JobFinder, ένας εργαζόμενος μπορεί να αναζητήσει θέσεις εργασίας με browsing, με μηχανή αναζήτησης, ανά κατηγορία ή πόλη, να εντοπίσει τα σημεία των θέσεων εργασίας σε χάρτη καθώς και να ζητήσει τα πλησιέστερα σημεία στον χάρτη σε σχέση με την πόλη που βρίσκεται. Τέλος, μπορεί να κάνει αίτηση για μια θέση εργασίας ή εγγραφή σε μια κατηγορία/πόλη λαμβάνοντας ενημερώσεις απευθείας στην ηλεκτρονική του διεύθυνση. Ο εργοδότης μπορεί να δημοσιεύσει αγγελίες με θέσεις εργασίας και να λαμβάνει τις αιτήσεις στην εταιρική ηλεκτρονική διεύθυνση. Ο διαχειριστής αποτελεί τον μεσολαβητή ανάμεσα στον εργαζόμενο και τον εργοδότη και διαχειρίζεται τις αγγελίες εργασίας, τις αιτήσεις, τις εγγραφές, τις κατηγορίες των θέσεων εργασίας και τις πόλεις. Η εφαρμογή JobFinder μπορεί να αξιοποιηθεί για θέσεις εργασίας που βρίσκονται σε όλη την Ελλάδα. Η ανάπτυξη της εφαρμογής βασίζεται στο μοντέλο αρχιτεκτονικού σχεδιασμού MVC (Model—view-controller). Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε γλώσσα PHP/MySQL και τεχνολογίες/εργαλεία που έπαιξαν καθοριστικό ρόλο στην δημιουργία μιας απλής και εύχρηστης εφαρμογής όπως Slim Microframework, Google Map API, PHPMailer, RedBeanPHP και Bootstrap 3 UI.

Abstract

Unipi JobFinder is an online application for searching jobs, posting jobs and managing jobs. JobFinder can be used in many different ways depending on the type of user and his/her needs. A person that is looking for a job and a person offering a job position can use this same app in different ways. However, both of them are considered simple users, contrary to the administrator of the platform. With JobFinder, an employee or unemployed person has the ability to browse through jobs, choose categories or cities, use the search engine, locate the activated job positions on a map, as well as ask for nearby locations. Moreover, this type of user can apply for a job and/or subscribe to certain job categories in order to receive new job posts directly to his/her email address. On the other hand, an employer can post a job position and receive job applications to the company's email address. The administrator of JobFinder has the ability to manage job posts, job applications, subscribers, job categories and cities. JobFinder is an application for posting job positions all over Greece. The development of JobFinder is based on the MVC software design pattern (Model-view-controller). It was programmed in PHP/MySQL combined with the use of technologies/tools that played a significant role in the creation of a simple and user-friendly online app such as Slim Microframework, Google Map API, PHPMailer, RedBeanPHP and Bootstrap 3 UI.

Σύντομη περιγραφή

Ζητούμενο της παρούσας μεταπτυχιακής διατριβής είναι ο προγραμματισμός μιας διαδικτυακής εφαρμογής που θα λειτουργεί ως πλατφόρμα εύρεσης, δημοσίευσης και διαχείρισης θέσεων εργασίας. Σκοπός της εργασίας είναι η ανάπτυξη μιας ολοκληρωμένης εφαρμογής που θα χαρακτηρίζεται για την λειτουργική της απλότητα και θα διαθέτει ένα εύχρηστο User Interface (UI). Η εφαρμογή αυτή ονομάζεται **Unipi JobFinder**.

Με βάση την χρηστο-κεντρική προσέγγιση, μπορούμε να διακρίνουμε **3 τύπους χρηστών** της εφαρμογής: τον εργαζόμενο, τον εργοδότη και τον διαχειριστή. Ο εργαζόμενος και ο εργοδότης είναι απλοί χρήστες, ενώ ο διαχειριστής αποκτάει παραπάνω δικαιώματα χρήσης από τους άλλους δύο μέσω μιας Login σελίδας.

Η εφαρμογή *JobFinder* εκτελεί τις ακόλουθες **βασικές λειτουργίες**:

- browsing στις αγγελίες εργασίας (browsing jobs)
- αναζήτηση θέσεων εργασίας με λέξεις – κλειδιά (searching jobs)
- εγγραφή για λήψη νέων αγγελιών εργασίας μέσω email (email subscription)
- προβολή αγγελιών εργασίας σε χάρτη (map)
- αίτηση για τις θέσεις εργασίας (apply for jobs)
- δημοσίευση αγγελιών εργασίας (post jobs)
- administration panel για την διαχείριση των αγγελιών, των αιτήσεων, των κατηγοριών εργασιών και των πόλεων (manage jobs, applications, categories, cities)
- δυνατότητα για RSS feed
- κοινοποίηση των θέσεων εργασίας στα μέσα κοινωνικής δικτύωσης (social media sharing)

Λειτουργίες ανά χρήστη

Ο **εργαζόμενος** είναι ο χρήστης όπου θα επισκεφθεί την πλατφόρμα με σκοπό να αναζητήσει διαθέσιμες θέσεις εργασίας. Σε αυτό τον τύπο χρήστη μπορούν να φανούν χρήσιμες οι ακόλουθες λειτουργίες:

- **Browsing**: Στην αρχική σελίδα της εφαρμογής, ο επισκέπτης μπορεί να δει τις ενεργοποιημένες αγγελίες εργασίας ανά κατηγορία (Παράρτημα 1: εικόνα 1). Με κλικ στην επιθυμητή αγγελία εργασίας, εμφανίζεται η σελίδα προβολής της θέσης εργασίας (Παράρτημα 1: εικόνα 2). Επιπλέον, ο χρήστης μπορεί να επιλέξει συγκεκριμένη κατηγορία (“Categories”) ή πόλη (“Cities”) από το κεντρικό μενού (Παράρτημα 1: εικόνες 3 και 4).
- **Αναζήτηση**: Στην αρχική σελίδα, ο χρήστης μπορεί να πληκτρολογήσει λέξεις – κλειδιά στην μηχανή αναζήτησης (Παράρτημα 1: εικόνα 5) και πατώντας enter να του εμφανιστούν τα αποτελέσματα αναζήτησης (Παράρτημα 1: εικόνα 6).
- **Εγγραφή**: Ο χρήστης μπορεί να ενημερώνεται αυτόματα στην ηλεκτρονική του διεύθυνση με τις νέες θέσεις εργασίας που δημοσιεύονται. Αφού επιλέξει από το

κεντρικό μενού την επιθυμητή κατηγορία ή πόλη, συμπληρώνει το email του στο πεδίο που βρίσκεται κάτω από τις αγγελίες και κάνει την εγγραφή (Παράρτημα 1: εικόνα 7).

- RSS feed: Εναλλακτικά, ο χρήστης μπορεί να ενημερώνεται για νέες αγγελίες μέσω RSS feed κάνοντας κλικ στο πορτοκαλί εικονίδιο με την ένδειξη RSS (Παράρτημα 1: εικόνα 8).
- Χάρτης: Ο χάρτης βοηθάει τον χρήστη να αναζητήσει θέσεις εργασίας με κριτήριο την τοποθεσία και συγκεκριμένα την πόλη. Επιλέγοντας “Map” στο κεντρικό μενού, ο χρήστης μπορεί να δει σε έναν χάρτη τις διαθέσιμες θέσεις εργασίας. Οι αγγελίες είναι απλωμένες στον χάρτη ανά πόλη και κατηγορία. Κάθε κατηγορία αντιστοιχεί σε marker με διαφορετικό χρώμα για να τις ξεχωρίζει ο χρήστης. Με κλικ πάνω σε ένα marker, ανοίγει info box με τις σημαντικότερες πληροφορίες για την θέση εργασίας όπως τίτλος, εταιρία, κατηγορία εργασίας, πόλη, ιστότοπος εργοδότη και ηλεκτρονική διεύθυνση εργοδότη (Παράρτημα 1: εικόνα 9). Επιπλέον, ο χρήστης μπορεί να δώσει ως input ένα αρχικό σημείο όπως για παράδειγμα την πόλη που μένει, καθώς και μια χιλιομετρική απόσταση (Παράρτημα 1: εικόνες 10 και 11). Αφού συμπληρώσει τα πεδία αυτά, επιλέγει “Show nearest jobs” και τότε εμφανίζονται στον χάρτη όλες οι αγγελίες εντός της γεωγραφικής ζώνης που όρισε ο ίδιος. Τέλος, με κλικ πάνω στα marker, ανοίγει ένα info box με όλες τις σημαντικότερες πληροφορίες για την θέση εργασίας, καθώς και όλα τα γεωγραφικά δεδομένα όπως γεωγραφικό μήκος και πλάτος της θέσης εργασίας, γεωγραφικό μήκος και πλάτος του αρχικού σημείου και η χιλιομετρική απόσταση μεταξύ των δύο σημείων (Παράρτημα 1: εικόνα 12).
- Αίτηση: Στην σελίδα προβολής κάθε αγγελίας, ο χρήστης μπορεί να κάνει αίτηση επιλέγοντας “Apply now” (Παράρτημα 1: εικόνα 13). Στην σελίδα που εμφανίζεται, συμπληρώνει όλα τα πεδία της φόρμας, επισυνάπτει το βιογραφικό του σημείωμα και επιλέγει “Submit your application” (Παράρτημα 1: εικόνα 14). Η αίτηση για την θέση εργασίας αποστέλλεται αυτόματα στην ηλεκτρονική διεύθυνση του εργοδότη. Σε περίπτωση που δεν εμφανίζεται το κουμπί “Submit your application”, αυτό συμβαίνει επειδή ο εργοδότης έχει δώσει συγκεκριμένες οδηγίες για τον τρόπο αίτησης. Συνεπώς, ο ενδιαφερόμενος ακολουθεί τις οδηγίες του εργοδότη που εμφανίζονται στο πεδίο “How to apply” όπως για παράδειγμα «Αποστολή βιογραφικού στο cv@company3.gr έως 31 Φεβρουαρίου 2017 (Παράρτημα 1: εικόνα 15).
- Social media sharing: Ο χρήστης έχει την δυνατότητα να κοινοποιήσει κάθε σελίδα προβολής αγγελίας στα μέσα κοινωνικής δικτύωσης, να την εκτυπώσει, να την αποστείλει μέσω email ή να την κάνει bookmark (Παράρτημα 1: εικόνα 16).

Ο επόμενος τύπος απλού χρήστη είναι ο **εργοδότης**, ο οποίος κατά κύριο λόγο θα χρησιμοποιήσει την εφαρμογή για να δημοσιεύσει μια αγγελία εργασίας. Για την δημοσίευση αγγελίας ο χρήστης πρέπει να κάνει κλικ στο μπλε κουμπί της αρχικής σελίδας με την ένδειξη “POST A JOB” (Παράρτημα 1: εικόνα 17). Στην συνέχεια μεταφέρεται σε μια νέα σελίδα (Παράρτημα 1: εικόνα 18) όπου συμπληρώνει υποχρεωτικά όλα τα πεδία της φόρμας (step “create”), την ελέγχει (step “review”) και την δημοσιεύει (step “publish”).

Ο τρίτος τύπος χρήστη είναι ο **διαχειριστής** της πλατφόρμας, ο οποίος μέσω ενός control panel μπορεί να διαχειριστεί κυρίως τις αγγελίες, τις αιτήσεις, τις κατηγορίες εργασίας και τις πόλεις. Για να το κάνει αυτό χρειάζεται να αποκτήσει δικαιώματα χρήσης admin μέσω ταυτοποίησης. Πιο συγκεκριμένα, επιλέγει “login” στο κεντρικό μενού και αφού μεταφερθεί στην login σελίδα, συμπληρώνει email και password (Παράρτημα 1: εικόνα 19). Εφόσον ταυτοποιηθεί

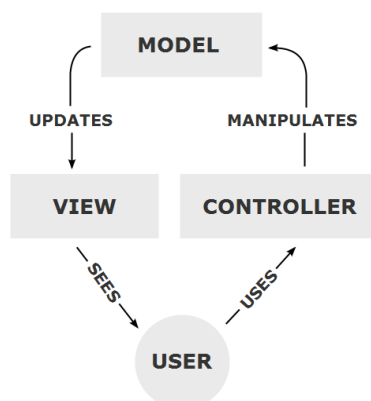
από την εφαρμογή ως διαχειριστής της πλατφόρμας, εισέρχεται στο admin panel όπου εκεί έχει τις ακόλουθες δυνατότητες:

- Διαχείριση αγγελιών (edit / feature on / approve / deactivate / delete): Στην αρχική σελίδα του panel, ο διαχειριστής μπορεί να δει τις αγγελίες που βρίσκονται σε κατάσταση “inactive”. Αυτό σημαίνει ότι δεν εμφανίζονται στις σελίδες που έχει πρόσβαση ο απλός χρήστης. Αυτό συμβαίνει είτε επειδή μια αγγελία έχει δημοσιευθεί από τον εργοδότη αλλά δεν έχει εγκριθεί από τον διαχειριστή, είτε επειδή έχουν λήξει δηλαδή, η επιτρεπόμενη χρονική διάρκεια ισχύος της αγγελίας έχει παρέλθει. Η default τιμή που έχει δοθεί είναι 30 ημέρες και μπορεί να αλλάξει μόνο από τον προγραμματιστή. Ο διαχειριστής μπορεί να το διαπιστώσει αυτό από την ημερομηνία δημοσίευσης που αναγράφεται σε κάθε αγγελία. Όσες αγγελίες ανήκουν στις μη εγκεκριμένες ελέγχονται και εγκρίνονται με κλικ στον πράσινο εικονίδιο (approve) που βρίσκεται στην στήλη “action” του πίνακα (Παράρτημα 1: εικόνα 20). Μετά την έγκρισή τους μπαίνουν σε status “active” και είναι ορατές στους απλούς χρήστες της εφαρμογής μέχρι να περάσουν 30 μέρες. Στην στήλη “action” βρίσκονται και άλλα εικονίδια για τον διαχειριστή όπως το γαλάζιο εικονίδιο για την επεξεργασία του περιεχομένου των αγγελιών (edit), το μπλε εικονίδιο για να κάνει highlight μια αγγελία (feature on), και το κόκκινο εικονίδιο για την διαγραφή των αγγελιών (delete). Προς διευκόλυνση του διαχειριστή, οι ληγμένες αγγελίες συγκεντρώνονται όλες στην σελίδα “Expire Jobs” που βρίσκεται στο drop-down υπο-μενού “Manage” του κεντρικού μενού. Η σελίδα αυτή είναι ακριβώς η ίδια με την αρχική σελίδα του admin panel με την διαφορά ότι σε αυτή εμφανίζονται μόνο οι ληγμένες θέσεις εργασίας ενώ στην αρχική εμφανίζονται τόσο οι ληγμένες όσο και οι μη εγκεκριμένες. Οι ληγμένες αγγελίες από “inactive” μπορούν να ξαναγίνουν “active” από τον διαχειριστή με την επιλογή του πράσινου εικονιδίου (approve) σε περίπτωση που ο εργοδότης ζητήσει παράταση ισχύος. Τέλος, η διαχείριση των αγγελιών μπορεί να πραγματοποιηθεί ανά κατηγορία ή ανά πόλη, αφού ο διαχειριστής επιλέξει από το κεντρικό μενού την κατηγορία ή πόλη που επιθυμεί. Στην περίπτωση αυτή, ο διαχειριστής βλέπει όσες αγγελίες είναι “active” ενώ αντί για το πράσινο εικονίδιο με εντολή “activate”, εμφανίζεται το πορτοκαλί εικονίδιο με εντολή
- “deactivate” (Παράρτημα 1: εικόνα 21). Συνεπώς, ο διαχειριστής έχει την δυνατότητα να απενεργοποιήσει εκτάκτως μια αγγελία χωρίς να περιμένει να λήξει. Συμπερασματικά, αξίζει να σημειωθεί ότι όλος αυτός ο μηχανισμός δίνει στον διαχειριστή τον ρόλο του ελεγκτή και μεσολαβητή ανάμεσα στον εργαζόμενο και τον εργοδότη. Από την μια πλευρά, αποτρέπει την αποστολή αιτήσεων από τον εργαζόμενο προς τον εργοδότη για αγγελίες που δεν ισχύουν, και παράλληλα διασφαλίζει στον εργαζόμενο ότι το περιεχόμενο της εφαρμογής είναι έγκυρο και σε ισχύ.
- Διαχείριση κατηγοριών: Στη σελίδα “Categories” του drop-down μενού “Manage”, ο χρήστης μπορεί να διαχειριστεί τις κατηγορίες των θέσεων εργασίας. Στην στήλη “action” της λίστας των κατηγοριών υπάρχει ένα γαλάζιο εικονίδιο (edit) για την επεξεργασία των κατηγοριών και ένα κόκκινο εικονίδιο (delete) για την διαγραφή των κατηγοριών. Επιπλέον, ο χρήστης μπορεί να προσθέσει νέες κατηγορίες συμπληρώνοντας την φόρμα δεξιά της λίστας (Παράρτημα 1: εικόνα 22).
- Διαχείριση πόλεων: Στη σελίδα “Cities” του drop-down μενού “Manage”, ο χρήστης μπορεί να διαχειριστεί τις πόλεις των θέσεων εργασίας. Στην στήλη “action” της λίστας των κατηγοριών υπάρχει ένα γαλάζιο εικονίδιο (edit) για την επεξεργασία των κατηγοριών και ένα κόκκινο εικονίδιο (delete) για την διαγραφή των κατηγοριών. Επιπλέον, ο χρήστης μπορεί να προσθέσει νέες πόλεις συμπληρώνοντας την φόρμα

- δεξιά της λίστας (Παράρτημα 1: εικόνα 23). Είναι σημαντικό να διευκρινιστεί ότι σε περίπτωση προσθήκης πόλης, χρειάζεται να ενημερωθεί η βάση δεδομένων με τις συντεταγμένες της νέας πόλης για να μπορεί να εμφανιστεί στον χάρτη. Αυτό γίνεται αυτόματα είτε από τον διαχειριστή είτε από τον προγραμματιστή της εφαρμογής μέσω Google API καλώντας το αρχείο ".../map/geocode.php".
- Διαχείριση εγγραφών: Στη σελίδα "Subscribers" του drop-down μενού "Manage", ο χρήστης μπορεί να διαχειριστεί τους εγγεγραμμένους χρήστες. Στην στήλη "action" της λίστας των κατηγοριών υπάρχει ένα πορτοκαλί εικονίδιο (deactivate) για την απενεργοποίηση μιας εγγραφής και ένα κόκκινο εικονίδιο (delete) για την διαγραφή της. Επιπλέον, ο χρήστης μπορεί να προσθέσει νέες εγγραφές συμπληρώνοντας την φόρμα δεξιά της λίστας (Παράρτημα 1: εικόνα 24).
- Δημοσίευση αγγελίας: Ο διαχειριστής μπορεί να δημοσιεύσει μια νέα αγγελία όπως οι απλοί χρήστες, επιλέγοντας "Post a job" από το κεντρικό μενού.
- Αποσύνδεση: Η αποσύνδεση του διαχειριστή από την διαχειριστική πλατφόρμα της εφαρμογής γίνεται με κλικ στην επιλογή "Logout" του κεντρικού μενού.

Μοντέλο αρχιτεκτονικής λογισμικού

Η εφαρμογή στηρίζεται στο μοντέλο **MVC (Model-view-controller)**. Το μοντέλο MVC είναι ένα μοντέλο αρχιτεκτονικής λογισμικού το οποίο χρησιμοποιείται για την δημιουργία περιβαλλόντων αλληλεπίδρασης χρήστη. Στο μοντέλο αυτό η εφαρμογή διαιρείται σε τρία διασυνδεδεμένα μέρη ώστε να διαχωριστεί η παρουσίαση της πληροφορίας στον χρήστη από την μορφή που έχει αποθηκευτεί στο σύστημα. Το κύριο μέρος του μοντέλου είναι το αντικείμενο **Model** το οποίο διαχειρίζεται την ανάκτηση/αποθήκευση των δεδομένων στο σύστημα. Το αντικείμενο **View** χρησιμοποιείται για να παρουσιάζεται η πληροφορία στον χρήστη (π.χ. με γραφικό τρόπο). Το τρίτο μέρος είναι ο **Controller** ο οποίος δέχεται την είσοδο και στέλνει εντολές στο αντικείμενο Model και στο View. Η σχεδίαση model-view-controller ορίζει και τις αλληλεπιδράσεις των μοντέλων: Ο **controller** μπορεί να στέλνει εντολές στο μοντέλο και να ενημερώνει την κατάσταση του μοντέλου. Μπορεί επίσης να στέλνει εντολές ώστε να γίνει η αντίστοιχη αναπαράσταση των δεδομένων του μοντέλου μέσω του View. Το **model** ενημερώνει τις αντίστοιχες αναπαραστάσεις views και τους controllers όταν υπάρχει αλλαγή στα δεδομένα. Αυτή η ενημέρωση επιτρέπει στα views να ενημερώνουν την γραφική απεικόνιση. Το **view** αναπαριστά με γραφικό τρόπο την πληροφορία που περιέχει το model δημιουργώντας γραφική παρουσίαση στο χρήστη.



Τεχνολογίες ανάπτυξης

Για τον προγραμματισμό της διαδικτυακής εφαρμογής JobFinder χρησιμοποιήθηκαν οι παρακάτω τεχνολογίες:

- **Γλώσσες προγραμματισμού:** HTML/CSS, Javascript, PHP, MySQL
- **Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE):** NetBeans IDE 8.2
- **Διακομιστές:** 127.0.0.1 (localhost) via TCP/IP, MySQL/MariaDB, Apache/2.4.23 (Win32) OpenSSL/1.0.2h PHP/5.6.24, phpMyAdmin, xampp-control
- **Slim Microframework:** Πρόκειται για ένα PHP framework που βοηθά στην κατασκευή απλών αλλά ισχυρών διαδικτυακών εφαρμογών και API. Ο τρόπος που λειτουργεί στον πυρήνα του είναι να λαμβάνει ένα αίτημα HTTP (HTTP request), να καλεί την κατάλληλη ρουτίνα επανάκλησης (callback routine) και να επιστρέφει μια απάντηση HTTP (HTTP response). Αρχικά, εκείνο που χρειάζεται είναι ένας Apache (εναλλακτικά Nginx) server. Η εφαρμογή αποτελείται από routes που στέλνουν στον client απαντήσεις σε συγκεκριμένα HTTP requests που λαμβάνει ο web server. Τα βήματα για την λειτουργία της εφαρμογής είναι: 1. Create & configure app, 2. Define app routes (τα routes δέχονται Request και Response αντικείμενα ως arguments εντός των callback routines), 3. Run app. Ακολουθεί ένα απλό παράδειγμα:

```
<?php
// Create and configure Slim app
$config = ['settings' => [
    'addContentLengthHeader' => false,
]];
$app = new \Slim\App($config);
// Define app routes
$app->get('/hello/{name}', function ($request, $response, $args) {
    return $response->write("Hello " . $args['name']);
});
// Run app
$app->run();
```

- **Slim Microframework Add-Ons:** Από τα πρόσθετα εξαρτήματα που διαθέτει το Slim Microframework, γίνεται χρήση του CSRF Protection το οποίο προστατεύει από επιθέσεις τύπου CSRF (cross-site request forgery). Πρόκειται για μια εφαρμογή που λειτουργεί ως middleware και παράγει ένα μοναδικό token με όνομα και τιμή για κάθε http request που πραγματοποιείται. Το όνομα και η τιμή του token χρησιμοποιείται με την μορφή hidden field στην html φόρμα του template έτσι ώστε να γίνεται επικαιροποίηση μέσω του μοναδικού token στο μεταγενέστερο POST request. Ακολουθεί ένα απλό παράδειγμα στο οποίο το GET request και το POST request θα μπορούσαν να βρίσκονται σε έναν controller και τα hidden fields στο view αρχείο .php με το οποίο αλληλεπιδράει ο controller για να εκτελεστεί η λειτουργία:

```
$app->get('/foo', function ($req, $res, $args) {
    // CSRF token name and value
    $nameKey = $this->csrf->getTokenNameKey();
```

```

$valueKey = $this->csrf->getTokenValueKey();
$name = $req->getAttribute($nameKey);
$value = $req->getAttribute($valueKey);

// Render HTML form which POSTs to /bar with two hidden input fields for the
// name and value:
// <input type="hidden" name="<?=$nameKey ?>" value="<?=$name ?>">
// <input type="hidden" name="<?=$valueKey ?>" value="<?=$value ?>">
});

$app->post('/bar', function ($req, $res, $args) {
    // CSRF protection successful if you reached
    // this far.
});

```

- **RedBeanPHP:** Βιβλιοθήκη τύπου ORM (object-relational mapping) για PHP που χρησιμεύει ως εργαλείο προγραμματισμού και τεχνική για την πρόσβαση από τον PHP κώδικα της εφαρμογής στην σχεσιακή βάση δεδομένων. Χρησιμοποιείται για λόγους ευκολίας και ταχύτητας στον προγραμματισμό σε γλώσσα PHP και παρέχει μια NoSQL εμπειρία χωρίς όμως να υποκαθιστά την SQL. Ένα χαρακτηριστικό παράδειγμα της βιβλιοθήκης αυτής είναι οι CRUD λειτουργίες της που δίνουν την δυνατότητα αλληλεπίδρασης με την βάση δεδομένων μέσα από αιτήματα τύπου CREATE, RETRIEVE, UPDATE, DELETE, τα οποία αποτελούν τον πυρήνα λειτουργίας πολλών διαδικτυακών εφαρμογών.
- **Bootstrap 3 UI:** Πρόκειται για ένα Kit που περιέχει vector User Interface στοιχεία. Τα στοιχεία αυτά είναι μορφοποιημένα και κωδικοποιημένα σε HTML/CSS κώδικα προκειμένου να χρησιμοποιηθούν στις ιστοσελίδες της εφαρμογής. Το Bootstrap είναι το πιο δημοφιλές HTML/CSS/Javascript framework για την ανάπτυξη διαδικτυακών εφαρμογών με αποκριτική σχεδίαση (responsive design) .
- **PHPMailer:** Βιβλιοθήκη για την αποστολή email σε PHP εφαρμογές.
- **Google Map API:** Διασύνδεση της εφαρμογής με το Google Map API για geocoding και προβολή MySQL δεδομένων σε χάρτη. Η διασύνδεση αυτή γίνεται μέσω HTTP υπηρεσιών, την χρήση ενός κλειδίου (Google Map API key), καθώς και την online ανταλλαγή πληροφοριών μέσω XML/JSON.

Οδηγίες εγκατάστασης JobFinder

Για την λειτουργία της εφαρμογής JobFinder είναι απαραίτητες οι παρακάτω τεχνικές προδιαγραφές:

- PHP 5.3 και άνω
- MySQL
- mod_rewrite ενεργοποιημένο

Για την εγκατάσταση της εφαρμογής είναι απαραίτητα τα ακόλουθα βήματα:

- Εξαγωγή του αρχείου .sql από τον φάκελο .zip της εφαρμογής
- Ενημέρωση του πίνακα admin με το επιθυμητό username και password (για λόγους ασφάλειας ο κωδικός κρυπτογραφείται με τον αλγόριθμο SHA1)
- Ανέβασμα της βάσης δεδομένων στον database server με import του updated αρχείου .sql
- Ανέβασμα του φακέλου με τα αρχεία της εφαρμογής στον application server
- Ενημέρωση του αρχείου config.php με τις επιθυμητές ρυθμίσεις (αν η εφαρμογή βρίσκεται σε απομακρυσμένο hosting provider τότε θα χρειαστεί ένας ftp εξυπηρετητής)
- Επιβεβαίωση ότι η πρόσβαση στα αρχεία της εφαρμογής στους φακέλους /assets/images και /assets/attachments είναι ενεργοποιημένη δηλαδή, ότι στα .htaccess αρχεία των φακέλων αυτών υπάρχει η ένδειξη “php_flag engine on” στην πρώτη γραμμή
- Επιβεβαίωση ότι έχουν ανέβει τα σωστά .htaccess αρχεία στον server

Σχετικά με τα παραπάνω βήματα εγκατάστασης, είναι σημαντικό να διευκρινιστούν τα εξής:

- Μέσα στον φάκελο της εφαρμογής βρίσκεται το αρχείο unipi.sql που είναι απαραίτητο για την εισαγωγή του σε μια MySQL βάση δεδομένων. Είναι σημαντικό να μην γίνουν αλλαγές στον sql κώδικα του αρχικού αρχείου προκειμένου να «τρέξουν» οι σωστές sql εντολές μετά το import και να δημιουργηθεί η βάση δεδομένων με τους σωστούς πίνακες και τις σωστές τιμές. Σε διαφορετική περίπτωση μπορεί να προκληθούν δυσλειτουργίες στην εφαρμογή λόγω κακής αλληλεπίδρασης των αρχείων .php με την MySQL βάση δεδομένων. Ο sql κώδικας της βάσης δεδομένων είναι διαθέσιμος στο Παράρτημα 2 – ενότητα «MYSQL ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ». Οι default τιμές για το Login του διαχειριστή στην πλατφόρμα είναι: email (username) = angel.korop και password = admin. Στο αρχείο unipi.sql υπάρχει η γραμμή κώδικα:

```
INSERT INTO `admin` (`id`, `email`, `password`) VALUES  
(1, 'angel.korop@gmail.com', 'd033e22ae348aeb5660fc2140aec35850c4da997');
```

Αυτή μπορεί να μεταβληθεί σε:

```
INSERT INTO `admin` (`id`, `email`, `password`) VALUES  
(1, 'your-email@gmail.com', sha1('your-password'));
```

- Οι σημαντικές ρυθμίσεις στο αρχείο config.php για την σωστή λειτουργία της εφαρμογής είναι:

```
// SMTP SETTINGS  
define('SMTP_ENABLED', true);
```

```

define('SMTP_AUTH', false);
define('SMTP_URL', 'localhost');
define('SMTP_USER', 'email@example.com');
define('SMTP_PASS', "");
define('SMTP_PORT', 25);
define('SMTP_SECURE', "");

// APPLICATION URL PATHS
define('BASE_URL', 'http://127.0.0.1:8080/jobfinder/'); // always include the trailing
slash at the end

// DATABASE SETTINGS
define('DB_HOST', 'localhost');
define('DB_NAME', 'unipi');
define('DB_USER', 'root');
define('DB_PASS', "");

```

Οι ρυθμίσεις στο πρωτόκολλο Simple Mail Transfer Protocol (SMTP) για την μετάδοση των μηνυμάτων ηλεκτρονικού ταχυδρομείου στο Διαδίκτυο, ικανοποιούν την περίπτωση που η εφαρμογή φιλοξενείται σε τοπικό server λόγω development ή test των λειτουργιών της εφαρμογής. Για την διαδικτυακή εφαρμογή προτείνεται η χρήση Gmail με τις παρακάτω ενδεικτικές ρυθμίσεις:

```

// SMTP SETTINGS
define('SMTP_ENABLED', true);
define('SMTP_AUTH', true);
define('SMTP_URL', 'smtp.gmail.com');
define('SMTP_USER', 'email@example.com');
define('SMTP_PASS', 'gmail password');
define('SMTP_PORT', 465);
define('SMTP_SECURE', 'ssl');

```

Αν ο διαχειριστής της εφαρμογής επιθυμεί να χρησιμοποιήσει δικό του mail host, τότε θα πρέπει να γίνουν οι κατάλληλες παραμετροποιήσεις έτσι ώστε τα email notifications να δουλεύουν σωστά.

Πηγαίος κώδικας

- **Κατηγορία MODELS:** Στο Παράρτημα 2 – ενότητα “MODELS” βρίσκεται διαθέσιμος ο πηγαίος κώδικας των βασικότερων αρχείων .php που συμπεριλαμβάνονται στον φάκελο models της εφαρμογής. Πρόκειται για τις κλάσεις (classes) που διαχειρίζονται τις αιτήσεις (applications.php), τις κατηγορίες (categories.php), τις πόλεις (cities.php), τις θέσεις εργασίας (jobs.php), την αναζήτηση (search.php) και τις εγγραφές (subscriptions.php).
- **Κατηγορία CONTROLLERS:** Στο Παράρτημα 2 – ενότητα “CONTROLLERS” βρίσκεται διαθέσιμος ο πηγαίος κώδικας των βασικότερων αρχείων .php που συμπεριλαμβάνονται στον φάκελο controllers της εφαρμογής. Πρόκειται για τα routes που ελέγχουν τις ενέργειες του διαχειριστή της πλατφόρμας (admin.php), τις

αιτήσεις (applications.php), τις κατηγορίες εργασιών (categories.php), τις πόλεις (cities.php), τις εργασίες (jobs.php), την αναζήτηση (search.php) και τις εγγραφές (subscriptions.php).

- **Κατηγορία VIEWS:** Στο Παράρτημα 2 – ενότητα “VIEWS” βρίσκεται ο πηγαίος κώδικας των βασικότερων αρχείων .php που συμπεριλαμβάνονται στον φάκελο views της εφαρμογής. Ο φάκελος views αποτελείται από δύο υπο-φακέλους, τον default και τον admin. Πρόκειται για τα αρχεία που ενημερώνουν τις αναπαραστάσεις των δεδομένων των μοντέλων (models) παρουσιάζοντας την πληροφορία στον απλό χρήστη (εργαζόμενος, εργοδότης) μέσω του υπο-φακέλου default και στον διαχειριστή της πλατφόρμας μέσω του υπο-φακέλου admin. Στον φάκελο **DEFAULT** βρίσκεται ο κώδικας των βασικότερων views που χρησιμοποιούνται για την αναπαράσταση της αρχικής σελίδας (home.php), της σελίδας προβολής των αγγελιών ανά κατηγορία (categories.php), της σελίδας προβολής των αγγελιών ανά πόλη (cities.php), της σελίδας με την φόρμα αίτησης (apply.new.php), των σελίδων για την δημοσίευση αγγελίας (post.new.php, post.publish.php, post.review.php, post.show.php) και της σελίδας προβολής του χάρτη (page.php). Στον φάκελο **ADMIN** βρίσκεται ο κώδικας των βασικότερων views που χρησιμοποιούνται για την αναπαράσταση σελίδων της πλατφόρμας διαχείρισης όπως η σελίδα login, η αρχική διαχειριστική σελίδα (home.php), η σελίδα διαχείρισης των θέσεων εργασίας ανά κατηγορία (categories.php), η σελίδα διαχείρισης ανά πόλη (cities.php), η σελίδα προβολής της συνολικής λίστας αιτήσεων σε αγγελίες (applications.php), η σελίδα προβολής της λίστας αιτήσεων σε συγκεκριμένη αγγελία (applications.job.php), η σελίδα διαχείρισης των κατηγοριών (categories.edit.php), η σελίδα διαχείρισης των πόλεων (cities.edt.php) και η σελίδα διαχείρισης των εγγραφών (subscribers.php).
- **Φάκελος MAP:** Στο Παράρτημα 2 – ενότητα “MAP” βρίσκεται ο πηγαίος κώδικας για τα αρχεία .php που αλληλεπιδρούν με το Google Map API, την βάση δεδομένων MySQL και το αρχείο page.php της κατηγορίας views, υπεύθυνο για την προβολή του χάρτη στη σελίδα Map. Όταν ο διαχειριστής της πλατφόρμας προσθέσει μια νέα πόλη, τότε θα χρειαστεί είτε ο ίδιος είτε ο προγραμματιστής της εφαρμογής να καλέσει το αρχείο geocode.php προκειμένου να ενεργοποιηθεί η διαδικασία geocoding μέσω Google Map API και να αποθηκευτούν στην βάση δεδομένων οι συντεταγμένες της πόλης αυτής. Το αρχείο rphpsqlajax_genxml.php επικοινωνεί με την βάση δεδομένων και με τα απαραίτητα sql ερωτήματα, αντλεί τα δεδομένα που χρειάζεται και τα στέλνει στο αρχείο page.php. Έτσι, όταν ο χρήστης καλέσει την σελίδα Map, ανοίγει ο χάρτης με τα markers και τα info box για κάθε marker. Όταν ο
- χρήστης δώσει στην εφαρμογή ως input, το αρχικό σημείο στο οποίο βρίσκεται και την μέγιστη απόσταση των θέσεων εργασίας από αυτό, τότε το αρχείο inputHandler.php επικοινωνεί με το Google Map API για να κάνει geocoding του αρχικού σημείου και στην συνέχεια υπολογίζει τα πλησιέστερα σημεία ενημερώνοντας παράλληλα την βάση δεδομένων. Το αρχείο rphpsqlajax_genxml2.php εκτελεί την ίδια λειτουργία με το αρχείο rphpsqlajax_genxml.php. Η διαφορά είναι ότι κάνει generate xml με τα δεδομένα που προκύπτουν από το input του χρήστη και το φιλτράρισμα του inputHandler.php και όχι με βάση τα default data δηλαδή, όλα τα σημεία θέσεων εργασίας. Το xml node λαμβάνεται από το αρχείο myMap.php όπου χαρτογραφεί τα δεδομένα με αποτέλεσμα να εμφανίζονται πάνω στον χάρτη τα πλησιέστερα σημεία που ζήτησε ο χρήστης.

Συμπεράσματα

Ζητούμενο της παρούσας μεταπτυχιακής εργασίας ήταν ο προγραμματισμός μιας διαδικτυακής εφαρμογής που θα λειτουργεί ως πλατφόρμα εύρεσης, δημοσίευσης και διαχείρισης θέσεων εργασίας.

Αρχικά, πραγματοποιήθηκε έρευνα παρόμοιων εφαρμογών που μπορούσαν να αποτελέσουν πηγή έμπνευσης ως προς τις λειτουργίες και το User Interface του JobFinder. Η γλώσσα προγραμματισμού που επιλέχθηκε είναι η PHP, η οποία είναι αλληλένδετη με την γλώσσα MySQL για την αλληλεπίδραση με την βάση δεδομένων. Αναζητήθηκαν οι κατάλληλες τεχνολογίες που θα ήταν συμβατές με την PHP και θα μπορούσαν να λειτουργήσουν ως καθοδηγητές στον τρόπο ανάπτυξης της εφαρμογής όπως frameworks και βιβλιοθήκες. Καθοριστικό ρόλο έπαιξε η χρήση του Slim Microframework, ένα framework αρκετά χρήσιμο για την ανάπτυξη εύχρηστων διαδικτυακών εφαρμογών σε σύντομο χρονικό διάστημα. Ένα ακόμα δημοφιλές framework για προγραμματισμό σε γλώσσα PHP είναι το Laravel, το οποίο όμως δεν προτιμήθηκε λόγω πολυπλοκότητας. Αξίζει να σημειωθεί, ότι σε θέματα User Interface έπαιξε καθοριστικό ρόλο η χρήση του Bootstrap, που είναι το πιο δημοφιλές HTML/CSS/JS framework για διαδικτυακές εφαρμογές. Επιπλέον, αξιοποιήθηκαν οι δυνατότητες που προσφέρει η Google στους προγραμματιστές μέσω Google Map API με σκοπό η εφαρμογή να προσφέρει στον χρήστη ένα καλύτερο User Experience.

Μελλοντικά υπάρχει η δυνατότητα για εξέλιξη και επέκταση των λειτουργιών και των χαρακτηριστικών της εφαρμογής JobFinder, όπως για παράδειγμα η μετατροπή της εφαρμογής σε πολυγλωσσική (εκτός από αγγλικά), η προσθήκη Javascript φίλτρων στον χάρτη για τον απλό χρήστη (π.χ. προβολή εργασιών ανά κατηγορία/πόλη), διασύνδεση της εφαρμογής με Google Analytics με σκοπό την παρακολούθηση της επισκεψιμότητας, εμπλουτισμός του περιεχομένου με την προσθήκη όλων των πόλεων της Ελλάδας και επιπλέον κατηγοριών για τις θέσεις εργασίας και η διασύνδεση της εφαρμογής με Paypal έτσι ώστε οι αγγελίες να μπαίνουν σε status "activated" μόνο όταν ο εργοδότης έχει πληρώσει συνδρομή για 30 μέρες.

Βιβλιογραφία - Ηλεκτρονικές πηγές

<https://secure.php.net/>

<http://www.w3schools.com/php/>

<https://www.mysql.com/>

<http://www.w3schools.com/sql/>

<https://www.javascript.com/>

http://www.w3schools.com/html/html_css.asp

<https://www.apachefriends.org/index.html>

<https://www.phpmyadmin.net/>

<https://www.slimframework.com/>

<http://www.redbeanphp.com/index.php>

<https://developers.google.com/maps/>

https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

<http://phpmailer.worxware.com/>

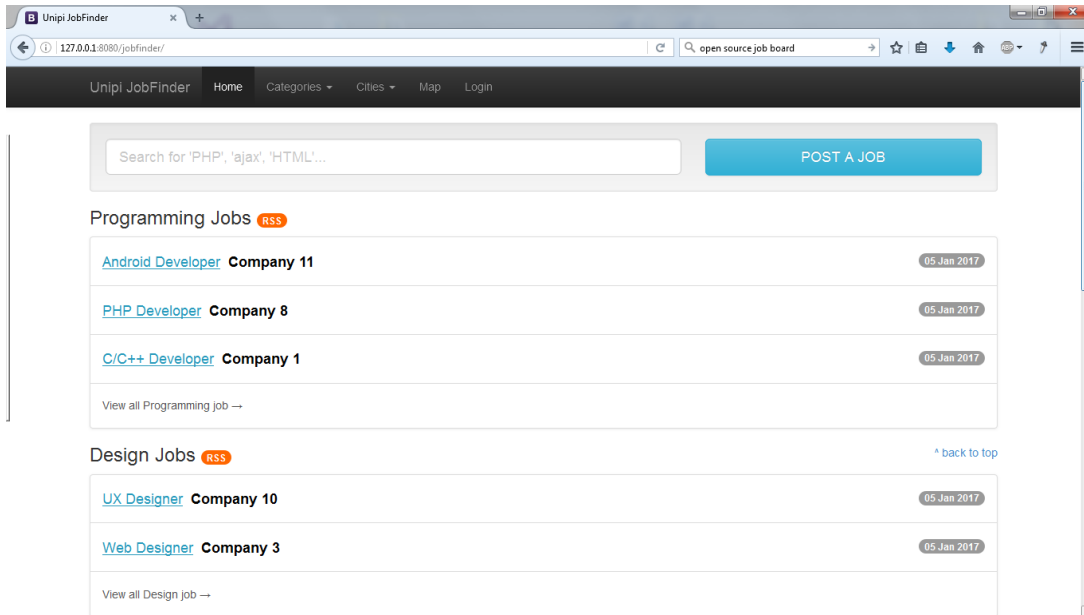
<https://netbeans.org/>

<http://bootstrapuikit.com/>

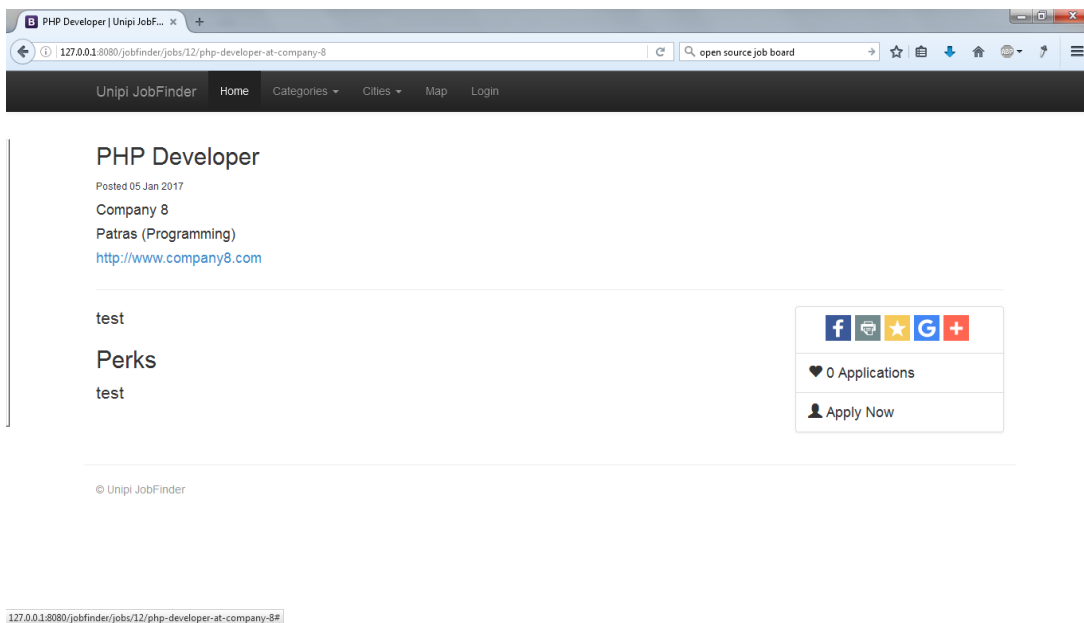
<http://www.w3schools.com/bootstrap/>

<http://getbootstrap.com/>

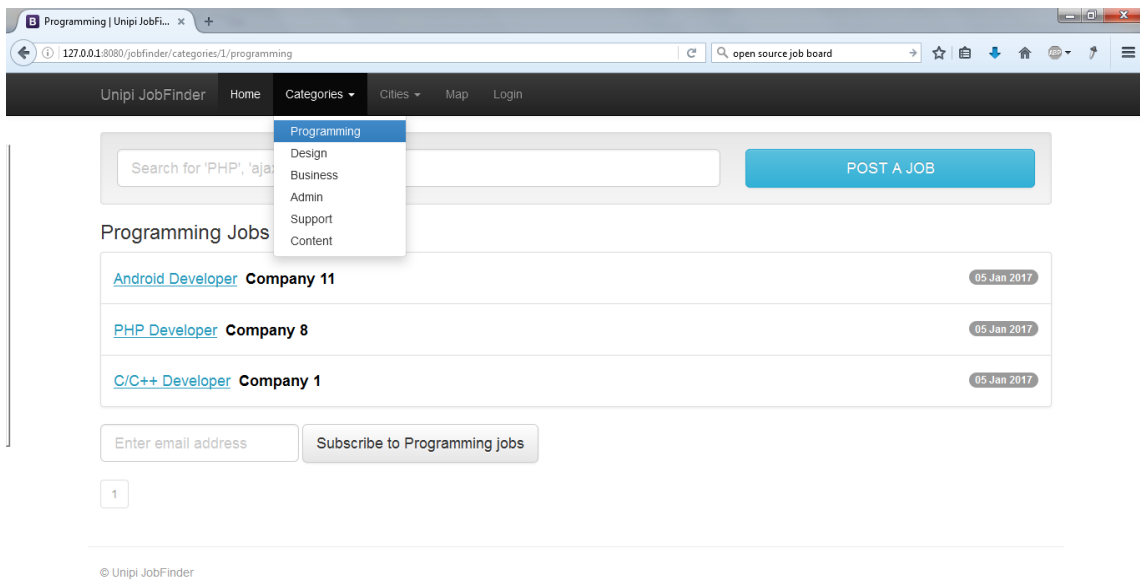
ΠΑΡΑΡΤΗΜΑ 1 – Εικόνες στιγμιότυπα JobFinder



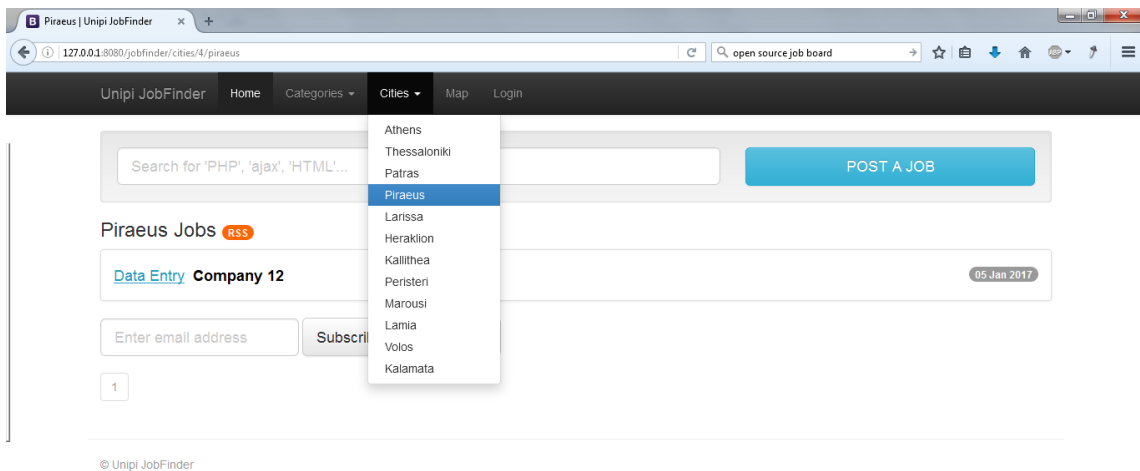
Εικόνα 1: Browsing θέσεων εργασίας – προβολή αγγελιών στην αρχική σελίδα ανά κατηγορία



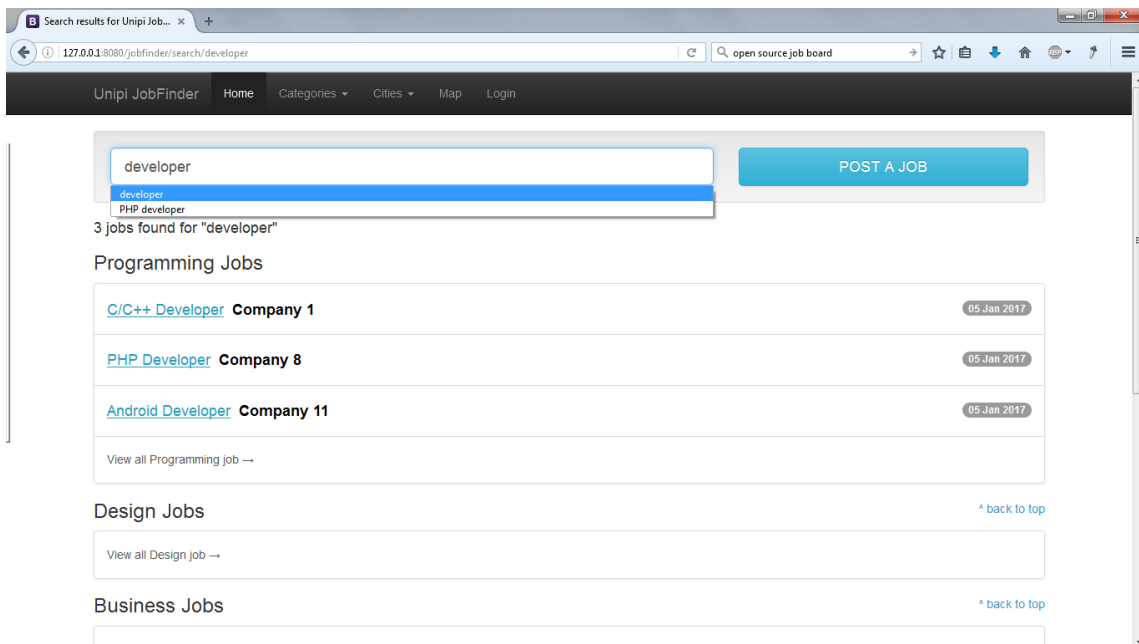
Εικόνα 2: Browsing θέσεων εργασίας – επιλογή και προβολή συγκεκριμένης αγγελίας εργασίας



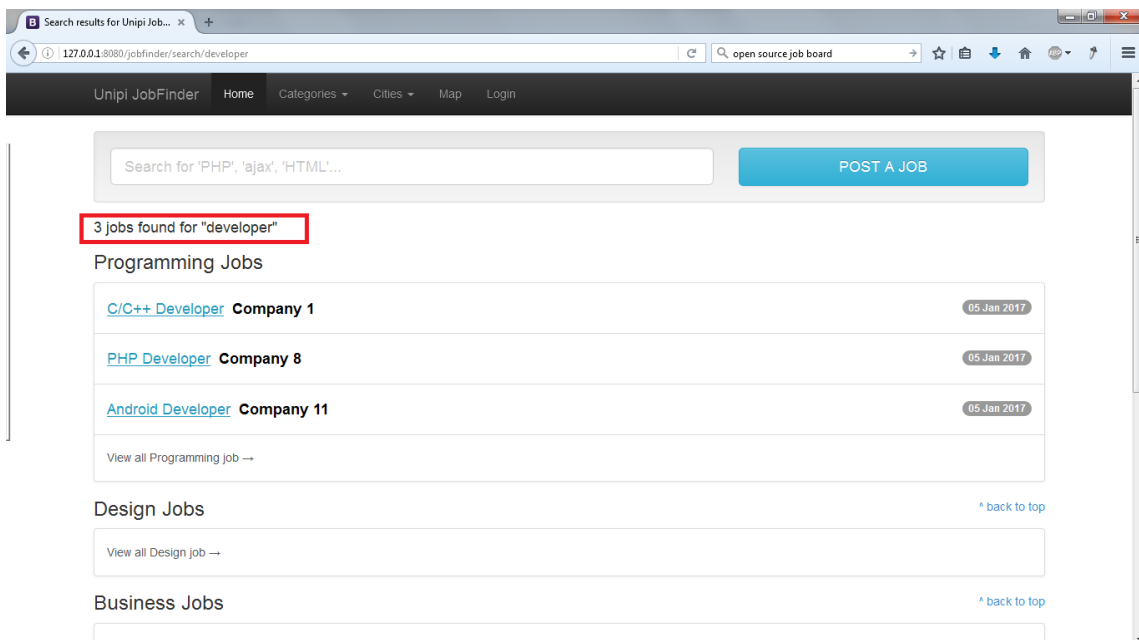
Εικόνα 3: Browsing θέσεων εργασίας - επιλογή κατηγορίας από το κεντρικό μενού



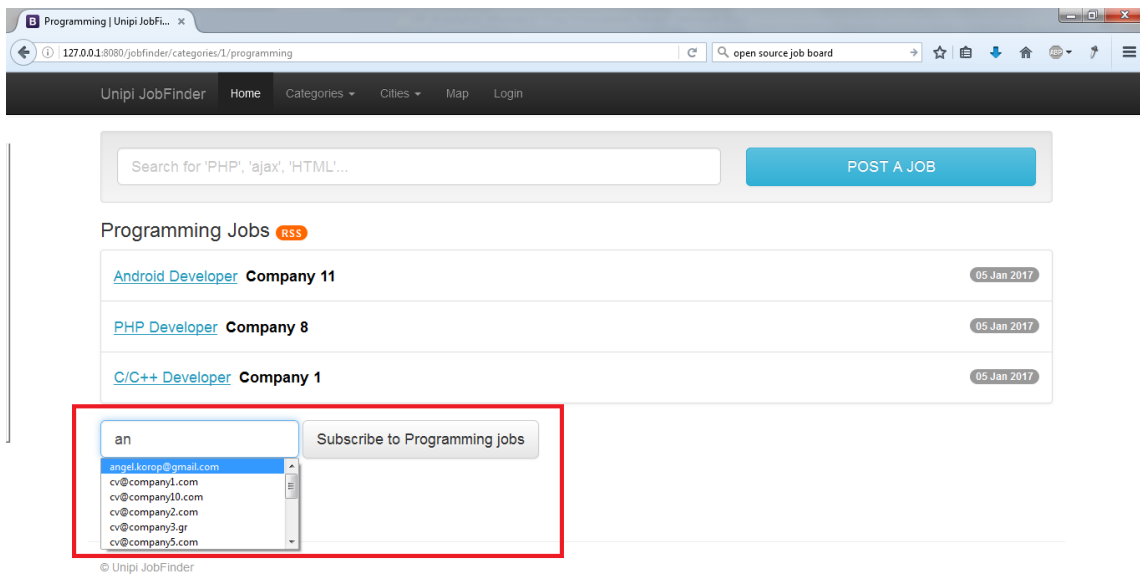
Εικόνα 4: Browsing θέσεων εργασίας - επιλογή πόλης από το κεντρικ



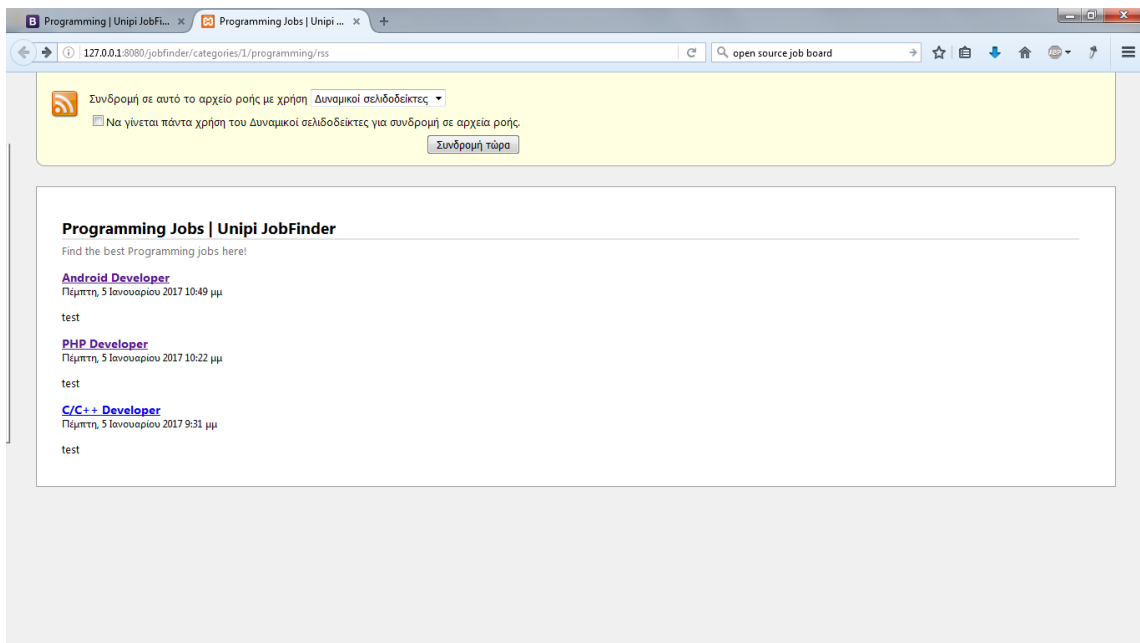
Εικόνα 5: Αναζήτηση θέσεων εργασίας με την λέξη – κλειδί “developer”



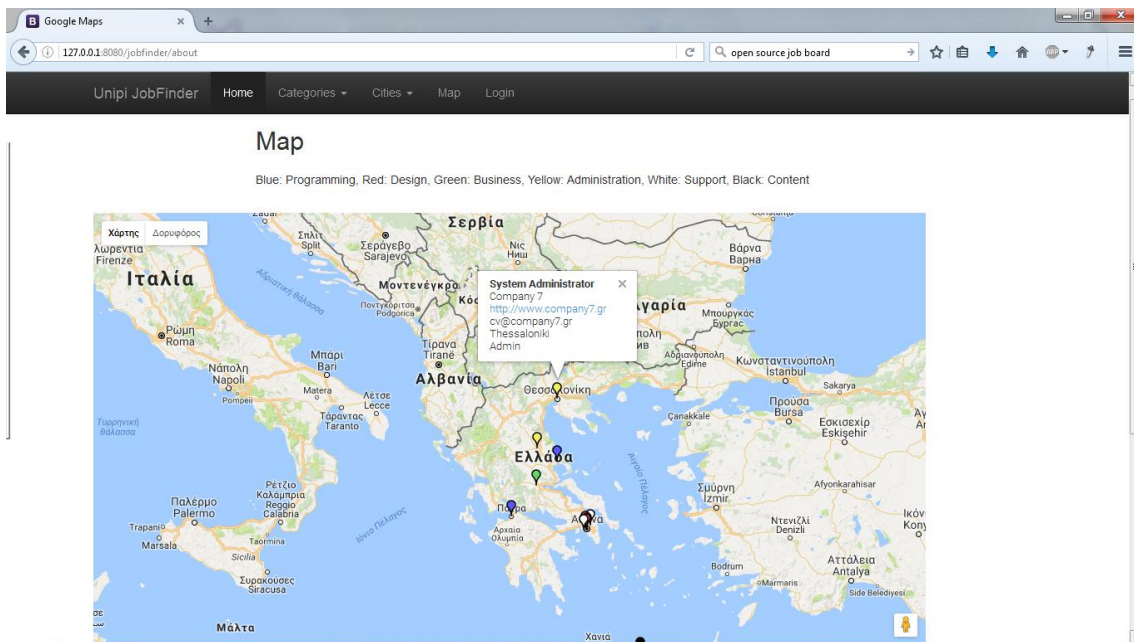
Εικόνα 6: Αποτελέσματα αναζήτησης για θέσεις εργασίας με τίτλο περιγραφής “developer”



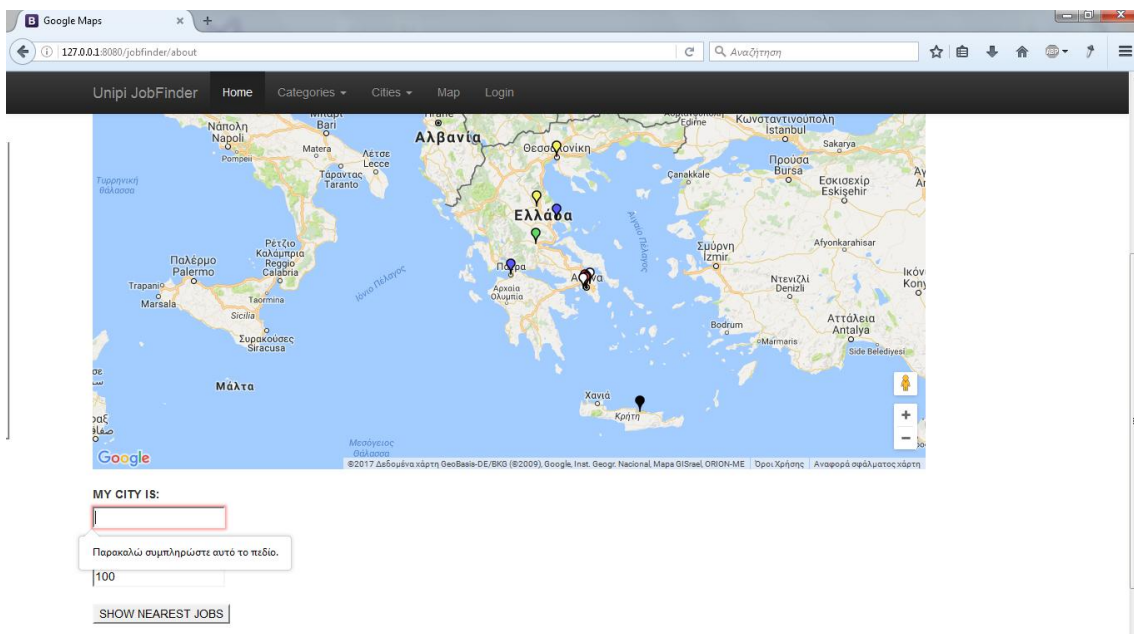
Εικόνα 7: Εγγραφή μέσω email για την κατηγορία αγγελιών “Programming”



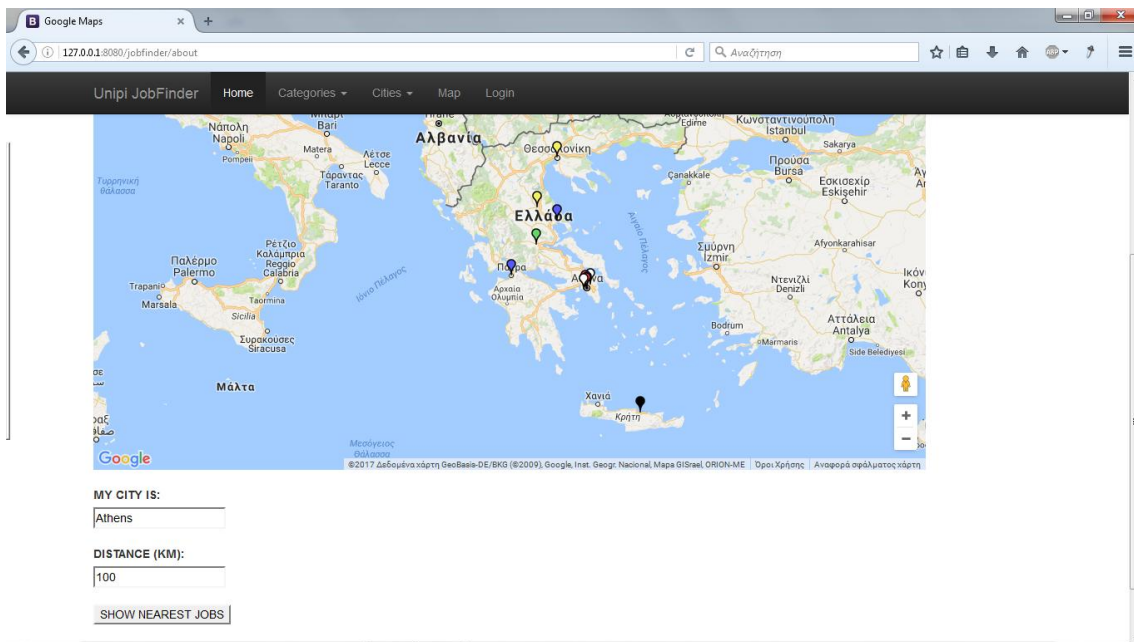
Εικόνα 8: RSS feed για την κατηγορία “Programming”



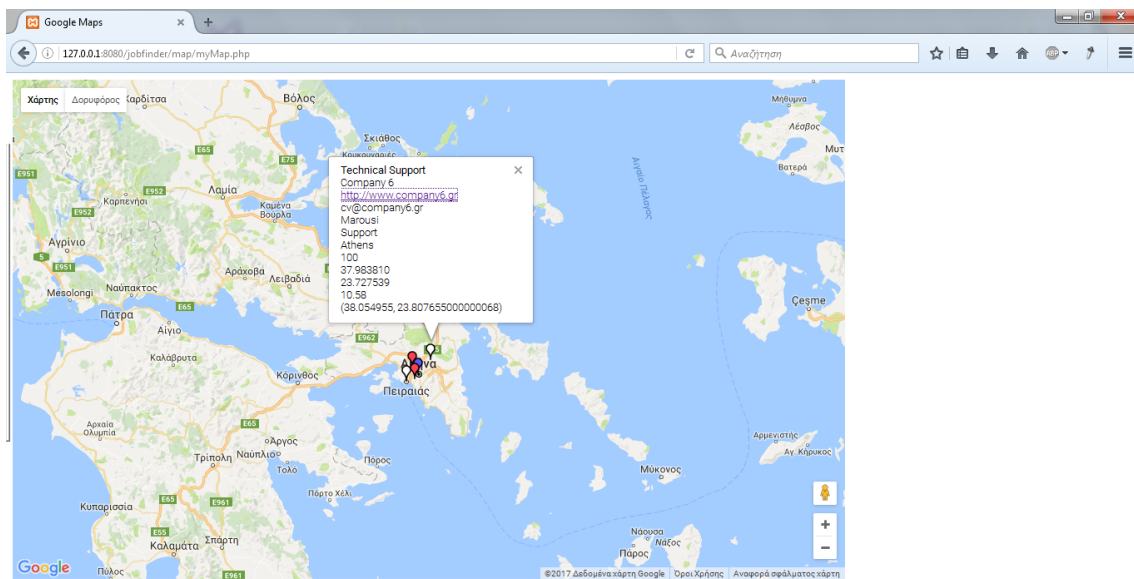
Εικόνα 9: Χάρτης με όλες τις διαθέσιμες θέσεις εργασίας



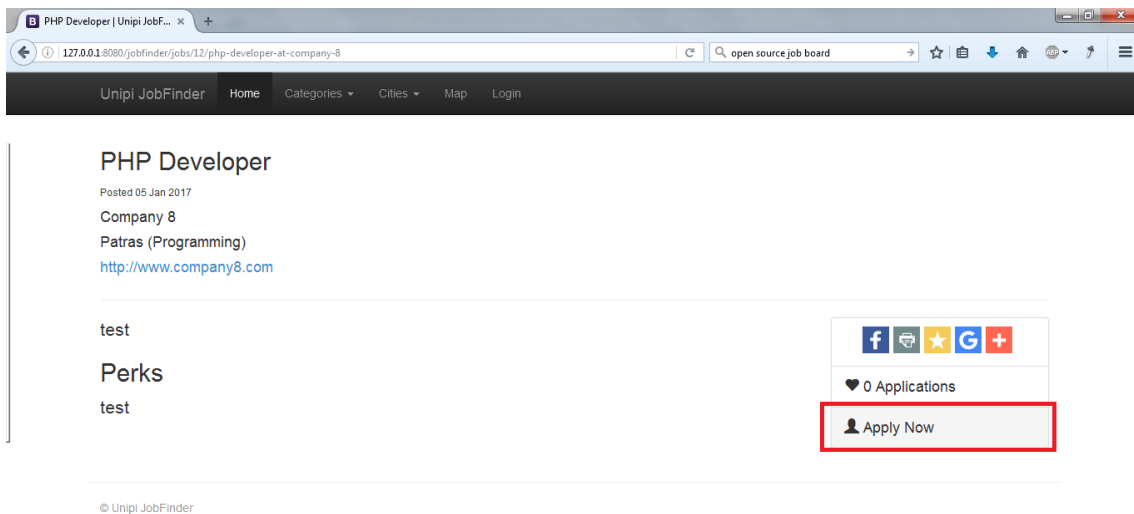
Εικόνα 10: Συμπλήρωση πεδίων στον χάρτη (υποχρεωτικά και τα 2 πεδία)



Εικόνα 11: Συμπλήρωση όλων των πεδίων στον χάρτη και αναζήτηση αποτελεσμάτων

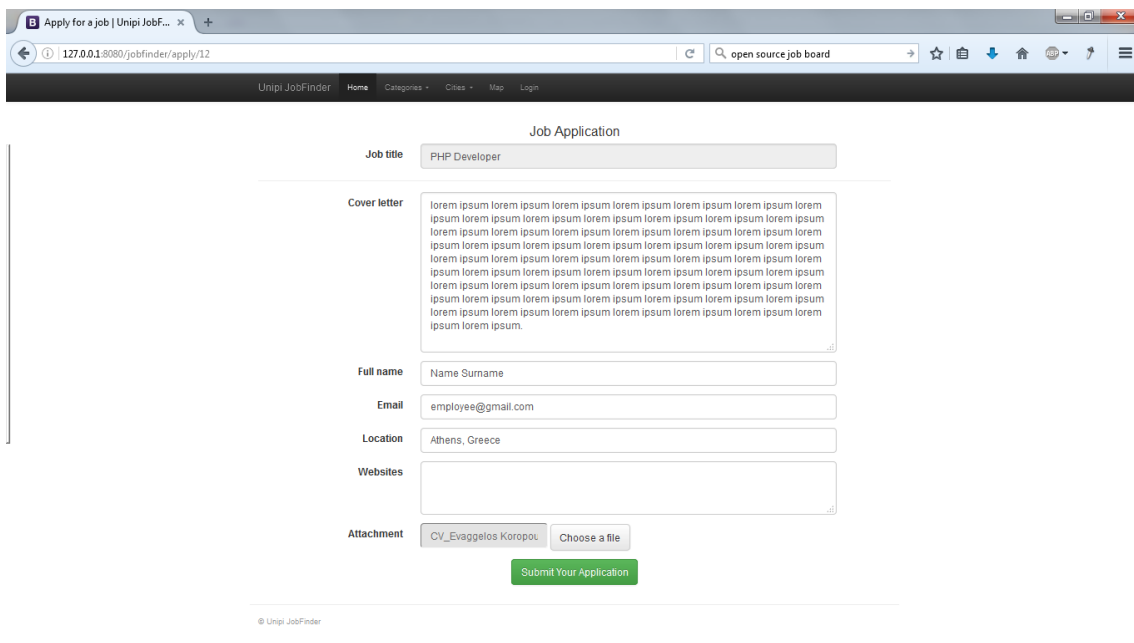


Εικόνα 12: Αποτελέσματα αναζήτησης με τις θέσεις εργασίας σε απόσταση 100χλμ από Αθήνα

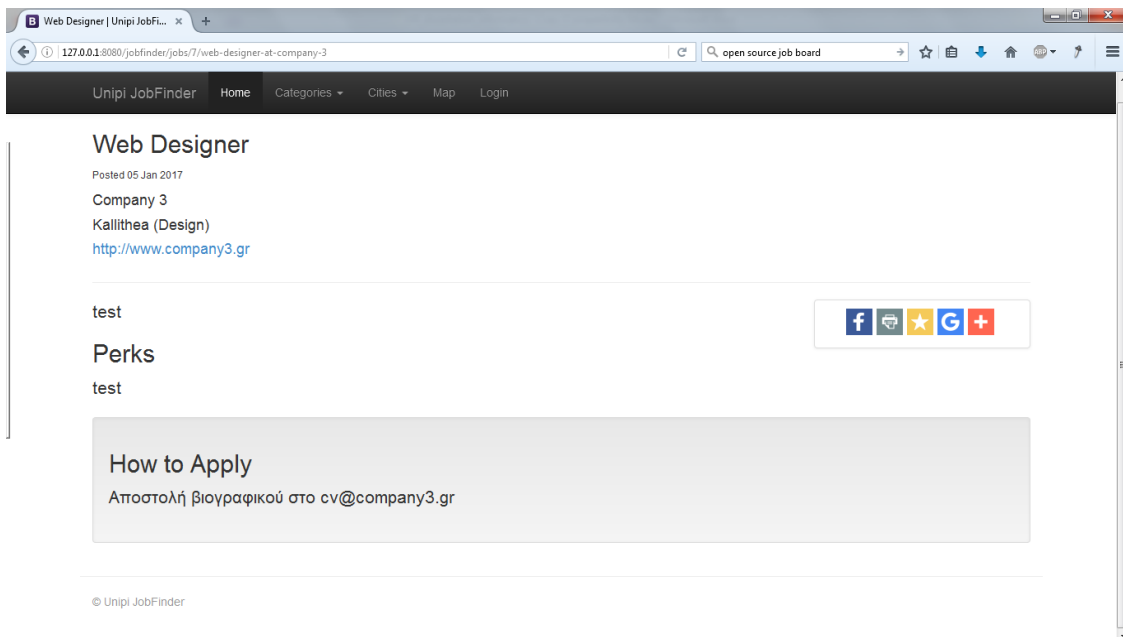


`127.0.0.1:8080/jobfinder/apply/12`

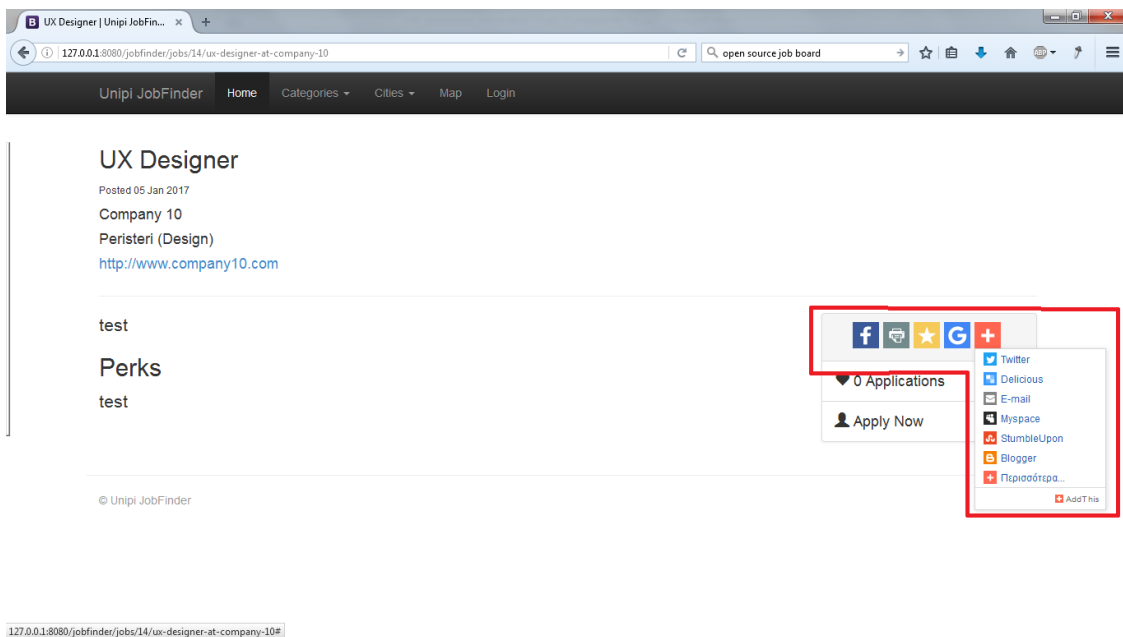
Εικόνα 13: Αίτηση για την θέση εργασίας “PHP Developer” μέσω “Apply Now”



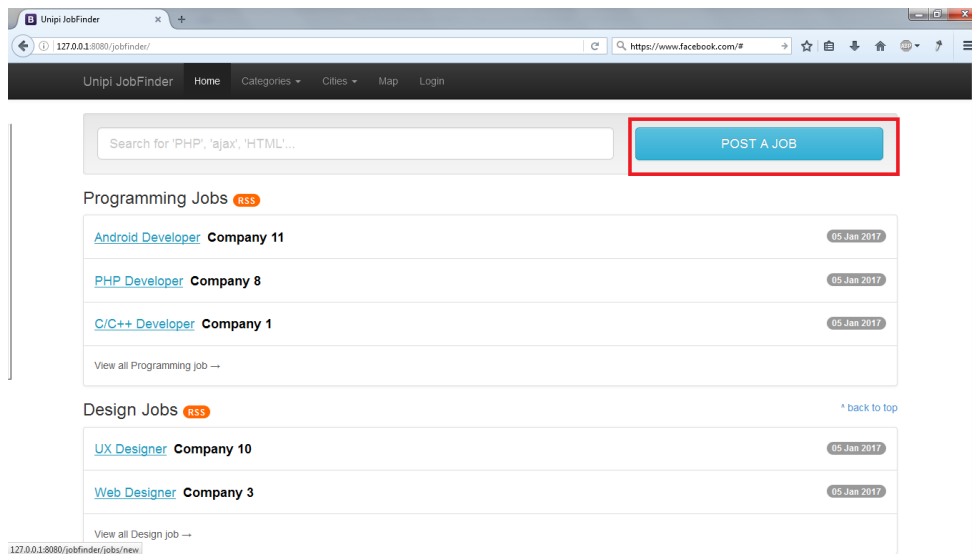
Εικόνα 14: Συμπλήρωση της φόρμας για αίτηση στη αγγελία “PHP Developer”



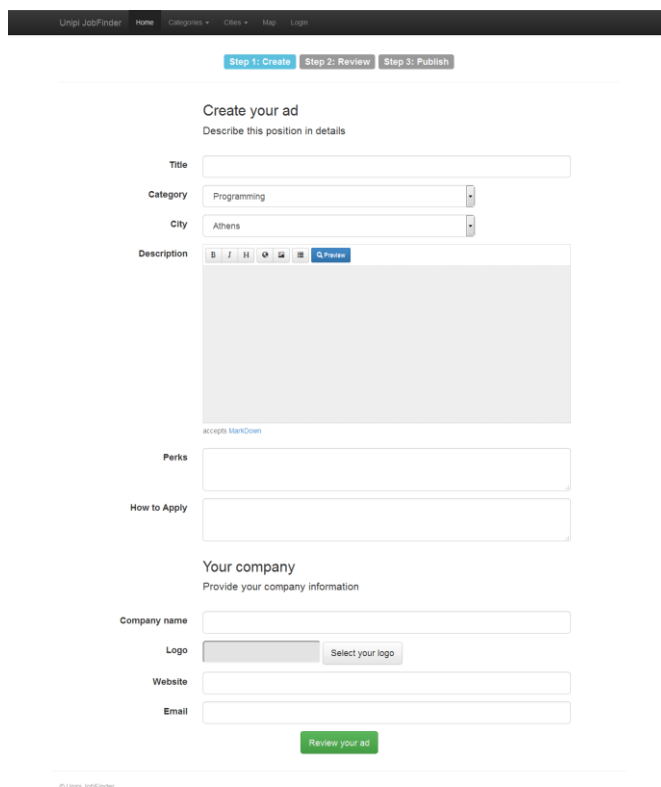
Εικόνα 15: Αίτηση για την θέση εργασίας “Web Designer” μέσω “How to Apply”



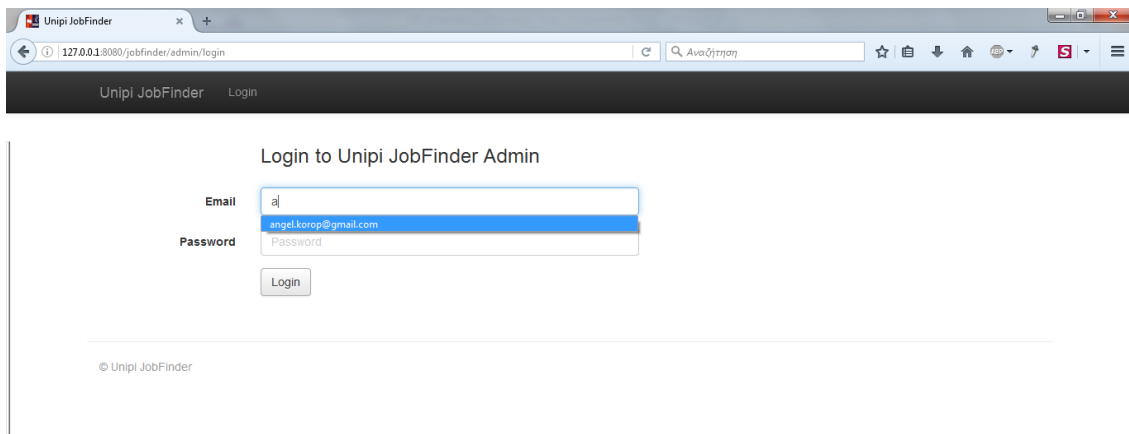
Εικόνα 16: Κοινοποίηση θέσης εργασίας στα μέσα κοινωνικής δικτύωσης



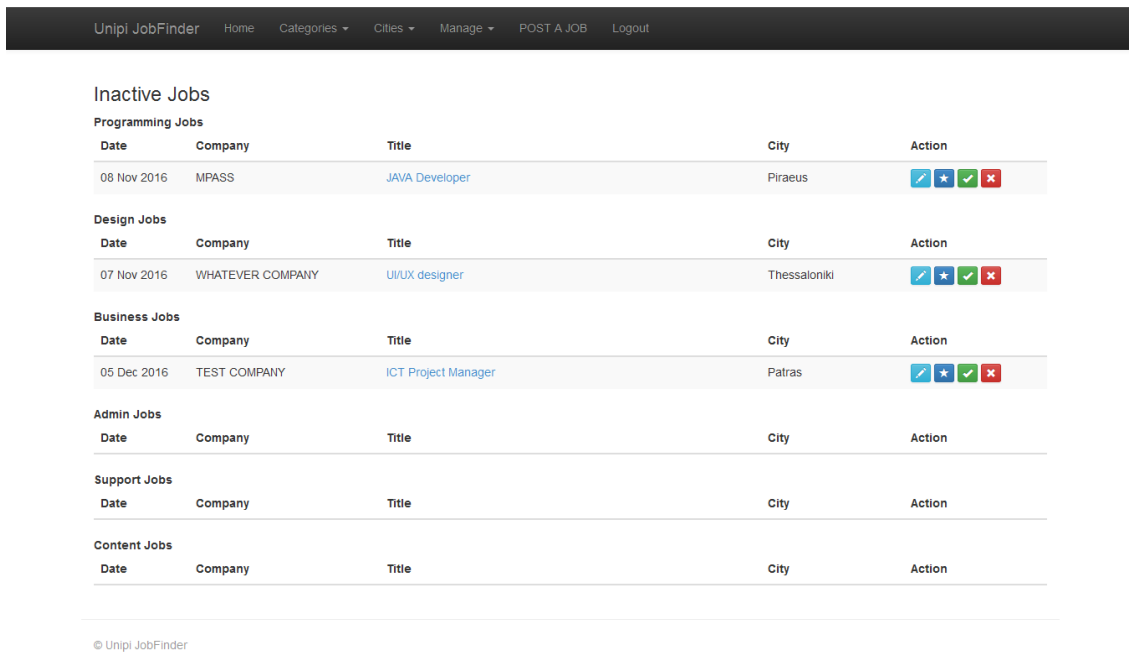
Εικόνα 17: Δημοσίευση αγγελίας



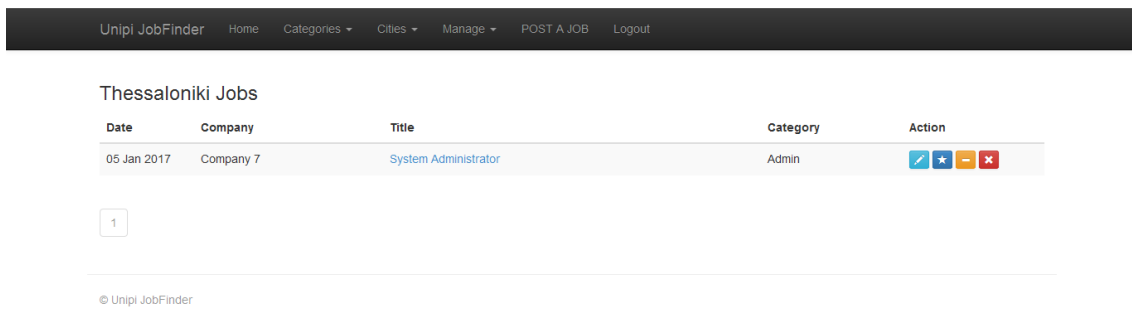
Εικόνα 18: Φόρμα δημοσίευσης αγγελίας



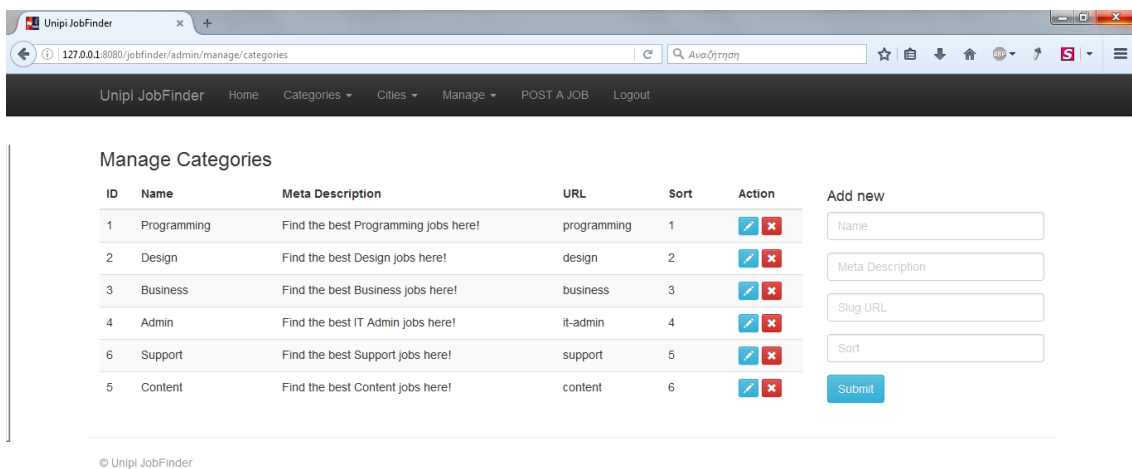
Εικόνα 19: Login σελίδα



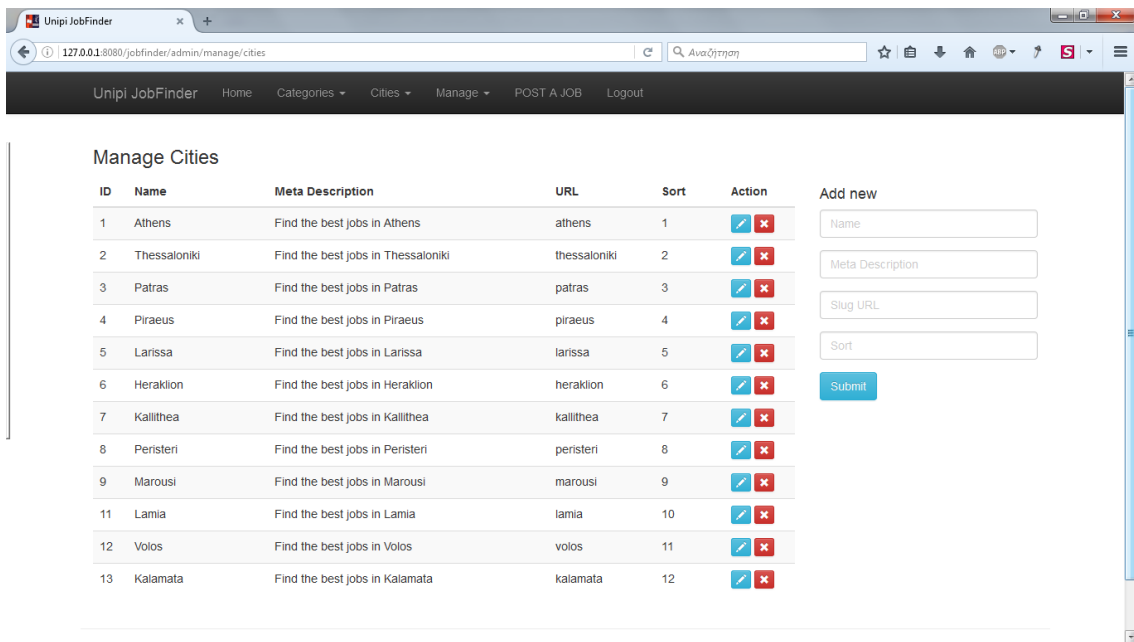
Εικόνα 20: Αρχική σελίδα του admin panel



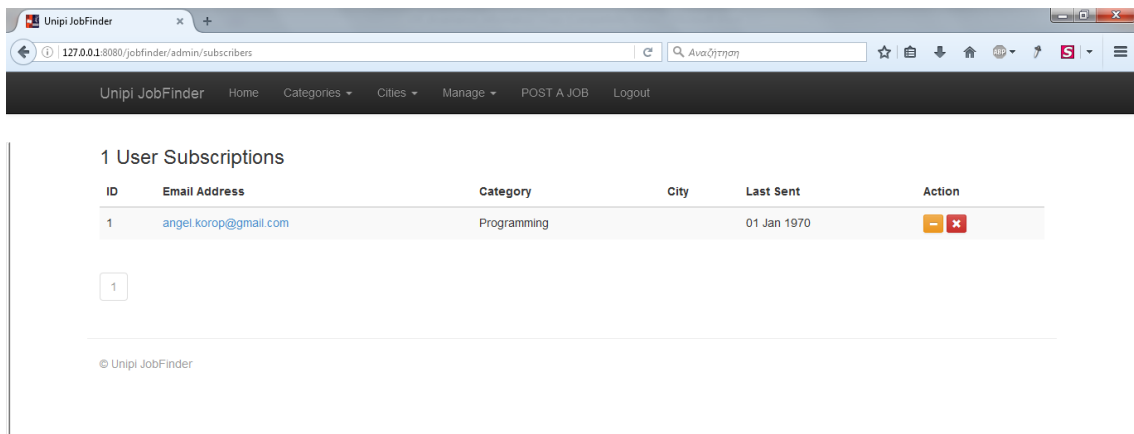
Εικόνα 21: Διαχειριστική σελίδα για τις θέσεις εργασίας με τοποθεσία «Θεσσαλονίκη»



Εικόνα 22: Διαχειριστική σελίδα για τις κατηγορίες των θέσεων εργασίας



Εικόνα 23: Διαχειριστική σελίδα για τις πόλεις των θέσεων εργασίας



Εικόνα 24: Διαχειριστική σελίδα για τις εγγραφές των χρηστών

ΠΑΡΑΡΤΗΜΑ 2 – Πηγαίος κώδικας JobFinder

MYSQL ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ

```

-- phpMyAdmin SQL Dump
-- version 4.5.1
-- http://www.phpmyadmin.net
--
-- Φιλοξενητής: 127.0.0.1
-- Χρόνος δημιουργίας: 02 Φεβ 2017 στις 21:14:19
-- Έκδοση διακομιστή: 10.1.16-MariaDB
-- Έκδοση PHP: 5.6.24

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Βάση δεδομένων: `unipi`
--

-----

--
-- Δομή πίνακα για τον πίνακα `admin`
--

CREATE TABLE `admin` (
  `id` tinyint(4) NOT NULL,
  `email` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  `password` varchar(50) COLLATE utf8_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--
-- Άδειασμα δεδομένων του πίνακα `admin`
--

INSERT INTO `admin` (`id`, `email`, `password`) VALUES
(1, 'angel.korop@gmail.com', 'd033e22ae348aeb5660fc2140aec35850c4da997');

-----

--
-- Δομή πίνακα για τον πίνακα `applications`

```

```

--
CREATE TABLE `applications` (
  `id` int(11) NOT NULL,
  `job_id` int(11) NOT NULL,
  `cover_letter` text COLLATE utf8_unicode_ci,
  `full_name` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `email` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `location` varchar(150) COLLATE utf8_unicode_ci DEFAULT NULL,
  `websites` text COLLATE utf8_unicode_ci,
  `attachment` text COLLATE utf8_unicode_ci,
  `token` varchar(32) COLLATE utf8_unicode_ci NOT NULL,
  `created` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--
-- Άδειασμα δεδομένων του πίνακα `applications`
--

INSERT INTO `applications` (`id`, `job_id`, `cover_letter`, `full_name`, `email`,
`location`, `websites`, `attachment`, `token`, `created`) VALUES
(1, 4, 'test application', 'Evaggelos Koropoulos', 'angel.korop@gmail.com', 'Athens,
Greece', 'www.testapplication.gr', '1480879898_CV_Evaggelos Koropoulos.pdf',
'24996b170c1a93bd83a732204d5f20d5', '2016-12-05 03:31:38'),
(2, 12, 'lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem
lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem ipsum lorem
lorem ipsum lorem ipsum.', 'Name Surname', 'employee@gmail.com', 'Athens,
Greece', 'www.myportfolio.gr', '1485811372_CV_Evaggelos Koropoulos.pdf',
'a44e1838a1974efd6ad957c7715c4742', '2017-01-30 23:22:52');

-----

--
-- Δομή πίνακα για τον πίνακα `banlist`
--

CREATE TABLE `banlist` (
  `id` int(11) NOT NULL,
  `type` varchar(10) COLLATE utf8_unicode_ci NOT NULL,
  `value` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `created` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

-----

--
-- Δομή πίνακα για τον πίνακα `blocks`
--

```

```

CREATE TABLE `blocks` (
  `id` int(11) NOT NULL,
  `name` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  `url` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  `content` text COLLATE utf8_unicode_ci
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--
-- Άδειασμα δεδομένων του πίνακα `blocks`
--

INSERT INTO `blocks` (`id`, `name`, `url`, `content`) VALUES
(1, 'AddThis', 'addthis', '<script type="text/javascript"
src="//s7.addthis.com/js/300/addthis_widget.js#pubid=ra-
541e6f810d78ec0c"></script>\n<div class="addthis_sharing_toolbox"></div>');

-----

--
-- Δομή πίνακα για τον πίνακα `categories`
--

CREATE TABLE `categories` (
  `id` int(11) NOT NULL,
  `name` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  `description` varchar(160) COLLATE utf8_unicode_ci DEFAULT NULL,
  `url` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  `sort` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--
-- Άδειασμα δεδομένων του πίνακα `categories`
--

INSERT INTO `categories` (`id`, `name`, `description`, `url`, `sort`) VALUES
(1, 'Programming', 'Find the best Programming jobs here!', 'programming', 1),
(2, 'Design', 'Find the best Design jobs here!', 'design', 2),
(3, 'Business', 'Find the best Business jobs here!', 'business', 3),
(4, 'Admin', 'Find the best IT Admin jobs here!', 'it-admin', 4),
(5, 'Content', 'Find the best Content jobs here!', 'content', 6),
(6, 'Support', 'Find the best Support jobs here!', 'support', 5);

-----

--
-- Δομή πίνακα για τον πίνακα `cities`
--

CREATE TABLE `cities` (
  `id` int(11) NOT NULL,
  `name` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  `description` varchar(160) COLLATE utf8_unicode_ci DEFAULT NULL,
  `url` varchar(150) COLLATE utf8_unicode_ci NOT NULL,
  `sort` int(11) NOT NULL,
  `lat` float(10,6) NOT NULL,
  `lng` float(10,6) NOT NULL

```



```

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--
-- Άδειασμα δεδομένων του πίνακα `cities`
--

INSERT INTO `cities` (`id`, `name`, `description`, `url`, `sort`, `lat`, `lng`) VALUES
(1, 'Athens', 'Find the best jobs in Athens', 'athens', 1, 37.983810, 23.727539),
(2, 'Thessaloniki', 'Find the best jobs in Thessaloniki', 'thessaloniki', 2, 40.640064,
22.944420),
(3, 'Patras', 'Find the best jobs in Patras', 'patras', 3, 38.246639, 21.734573),
(4, 'Piraeus', 'Find the best jobs in Piraeus', 'piraeus', 4, 37.943012, 23.646955),
(5, 'Larissa', 'Find the best jobs in Larissa', 'larissa', 5, 39.639023, 22.419125),
(6, 'Heraklion', 'Find the best jobs in Heraklion', 'heraklion', 6, 35.338734,
25.144213),
(7, 'Kallithea', 'Find the best jobs in Kallithea', 'kallithea', 7, 37.955894, 23.702099),
(8, 'Peristeri', 'Find the best jobs in Peristeri', 'peristeri', 8, 38.017666, 23.685926),
(9, 'Marousi', 'Find the best jobs in Marousi', 'marousi', 9, 38.054955, 23.807655),
(11, 'Lamia', 'Find the best jobs in Lamia', 'lamia', 10, 38.871044, 22.381462),
(12, 'Volos', 'Find the best jobs in Volos', 'volos', 11, 39.362190, 22.942160),
(13, 'Kalamata', 'Find the best jobs in Kalamata', 'kalamata', 12, 37.042236,
22.114126);

-----

--
-- Δομή πίνακα για τον πίνακα `downloads`
--

CREATE TABLE `downloads` (
  `id` int(11) NOT NULL,
  `email` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `website` text COLLATE utf8_unicode_ci,
  `ip` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `created` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

-----

--
-- Δομή πίνακα για τον πίνακα `filtered_markers`
--

CREATE TABLE `filtered_markers` (
  `title` varchar(255) DEFAULT NULL,
  `company_name` varchar(255) DEFAULT NULL,
  `url` text,
  `email` varchar(255) DEFAULT NULL,
  `city_name` varchar(150) NOT NULL,
  `lat` float(10,6) NOT NULL,
  `lng` float(10,6) NOT NULL,
  `categories_name` varchar(150) NOT NULL,
  `city` varchar(255) NOT NULL,
  `input_distance` int(10) NOT NULL,
  `initial_lat` float(10,6) NOT NULL,
  `initial_lng` float(10,6) NOT NULL,

```

```

`distance` float(10,2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Άδειασμα δεδομένων του πίνακα `filtered_markers`
--

INSERT INTO `filtered_markers` (`title`, `company_name`, `url`, `email`,
`city_name`, `lat`, `lng`, `categories_name`, `city`, `input_distance`, `initial_lat`,
`initial_lng`, `distance`) VALUES
('Android Developer', 'Company 11', 'http://www.company11.gr', 'hr@company11.gr',
'Athens', 37.983810, 23.727539, 'Programming', 'athens', 15, 37.983810, 23.727539,
0.00),
('Web Designer', 'Company 3', 'http://www.company3.gr', 'cv@company3.gr',
'Kallithea', 37.955894, 23.702099, 'Design', 'athens', 15, 37.983810, 23.727539,
3.82),
('UX Designer', 'Company 10', 'http://www.company10.com', 'cv@company10.com',
'Peristeri', 38.017666, 23.685926, 'Design', 'athens', 15, 37.983810, 23.727539,
5.24),
('Data Entry', 'Company 12', 'http://www.company12.com', 'hr@company12.gr',
'Piraeus', 37.943012, 23.646955, 'Support', 'athens', 15, 37.983810, 23.727539,
8.40),
('Technical Support', 'Company 6', 'http://www.company6.gr', 'cv@company6.gr',
'Marousi', 38.054955, 23.807655, 'Support', 'athens', 15, 37.983810, 23.727539,
10.58);

-----

--
-- Δομή πίνακα για τον πίνακα `input`
--

CREATE TABLE `input` (
`id` int(10) NOT NULL,
`city` varchar(255) NOT NULL,
`input_distance` int(10) NOT NULL,
`initial_lat` float(10,6) NOT NULL,
`initial_lng` float(10,6) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Άδειασμα δεδομένων του πίνακα `input`
--

INSERT INTO `input` (`id`, `city`, `input_distance`, `initial_lat`, `initial_lng`) VALUES
(20, 'athens', 15, 37.983810, 23.727539);

-----

--
-- Δομή πίνακα για τον πίνακα `jobs`
--

CREATE TABLE `jobs` (
`id` int(11) UNSIGNED NOT NULL,
`title` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,

```

```

`category` tinyint(4) DEFAULT NULL,
`city` tinyint(4) DEFAULT NULL,
`description` text COLLATE utf8_unicode_ci,
`perks` text COLLATE utf8_unicode_ci,
`how_to_apply` text COLLATE utf8_unicode_ci,
`company_name` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
`logo` text COLLATE utf8_unicode_ci,
`url` text COLLATE utf8_unicode_ci,
`email` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
`is_featured` tinyint(1) DEFAULT NULL,
`token` varchar(32) COLLATE utf8_unicode_ci DEFAULT NULL,
`status` tinyint(3) DEFAULT NULL,
`created` datetime DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--
-- Άδειασμα δεδομένων του πίνακα `jobs`
--

INSERT INTO `jobs` (`id`, `title`, `category`, `city`, `description`, `perks`,
`how_to_apply`, `company_name`, `logo`, `url`, `email`, `is_featured`, `token`,
`status`, `created`) VALUES
(2, 'UI/UX designer', 2, 2, 'whatever', 'whatever', 'whatever', 'WHATEVER
COMPANY', '', 'hr@whatever.com', 0, 'd5b3c425799476f8eaef7a4dc783dafb', 0,
'2016-11-07 01:39:07'),
(3, 'JAVA Developer', 1, 4, 'lorem ipsum lorem ipsum', 'lorem ipsum lorem ipsum',
'lorem ipsum lorem ipsum', 'MPASS', '', 'http://www.mpass.gr',
'angel.korop@gmail.com', 0, 'c8198e30fb07af93ce3cf759b12c8365', 0, '2016-11-08
03:38:47'),
(4, 'ICT Project Manager', 3, 3, 'description test', 'description test', '', 'TEST
COMPANY', '', 'http://www.testcompany.com', 'cv@testcompany.com', 0,
'4958d4707edabbabdaaf926f8ce2d75e', 0, '2016-12-05 03:16:41'),
(5, 'C/C++ Developer', 1, 12, 'test', 'test', '', 'Company 1', '',
'http://www.company1.com', 'cv@company1.com', 0,
'aa6f1f8e027d0142cda8999f4a70be07', 1, '2017-01-05 21:31:09'),
(6, 'Sales Manager', 3, 11, 'test', 'test', '', 'Company 2', '', 'http://www.company2.com',
'cv@company2.com', 0, '10414c182faf9bf906ff25ee02dc5e10', 1, '2017-01-05
21:32:45'),
(7, 'Web Designer', 2, 7, 'test', 'test', 'Αποστολή βιογραφικού στο cv@company3.gr',
'Company 3', '', 'http://www.company3.gr', 'cv@company3.gr', 0,
'6f3da3413c4a532b9564c558f028868a', 1, '2017-01-05 21:34:39'),
(9, 'Graphic Designer', 5, 6, 'test', 'test', '', 'Company 5', '',
'http://www.company5.com', 'cv@company5.com', 0,
'85cb00128bc6e62416473fc6deb383ee', 1, '2017-01-05 21:38:08'),
(10, 'Technical Support', 6, 9, 'test', 'test', '', 'Company 6', '',
'http://www.company6.gr', 'cv@company6.gr', 0,
'1fbbd158e0fce646183abcd0faa4dfd6', 1, '2017-01-05 21:39:40'),
(11, 'System Administrator', 4, 2, 'test', 'test', '', 'Company 7', '',
'http://www.company7.gr', 'cv@company7.gr', 0,
'f5e48b3f5f505135758ad9eca97d290d', 1, '2017-01-05 22:19:19'),
(12, 'PHP Developer', 1, 3, 'test', 'test', '', 'Company 8', '', 'http://www.company8.com',
'cv@company8.com', 0, '179b9789b4eee823c87c72b8d4262ab7', 1, '2017-01-05
22:22:30'),
(13, 'Data Analyst', 4, 5, 'test', 'test', '', 'Company 9', '', 'http://www.company9.com',
'info@company9.gr', 0, 'bac76746cbd4fc9222f401f8a90ff376', 1, '2017-01-05
22:28:15'),

```

```
(14, 'UX Designer', 2, 8, 'test', 'test', '', 'Company 10', '', 'http://www.company10.com',
'cv@company10.com', 0, '63750ab82fe988469a82bd07906a5066', 1, '2017-01-05
22:45:12'),
(15, 'Android Developer', 1, 1, 'test', 'test', '', 'Company 11', '',
'http://www.company11.gr', 'hr@company11.gr', 0,
'6ce3a48328f2e2bf38dc7f14de345b62', 1, '2017-01-05 22:49:43'),
(16, 'Data Entry', 6, 4, 'test', 'test', 'Αποστολή στο hr@company12.gr', 'Company 12',
'', 'http://www.company12.com', 'hr@company12.gr', 0,
'5633bfc2a7d877e9e140956af33b0beb', 1, '2017-01-05 22:54:09');
```

```
-----
```

```
--
```

```
-- Δομή πίνακα για τον πίνακα `markers`
```

```
--
```

```
CREATE TABLE `markers` (
  `title` varchar(255) DEFAULT NULL,
  `company_name` varchar(255) DEFAULT NULL,
  `url` text CHARACTER SET utf8 COLLATE utf8_unicode_ci,
  `email` varchar(255) DEFAULT NULL,
  `city_name` varchar(150) NOT NULL,
  `lat` float(10,6) NOT NULL,
  `lng` float(10,6) NOT NULL,
  `categories_name` varchar(150) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Άδειασμα δεδομένων του πίνακα `markers`
```

```
--
```

```
INSERT INTO `markers` (`title`, `company_name`, `url`, `email`, `city_name`, `lat`,
`lng`, `categories_name`) VALUES
('C/C++ Developer', 'Company 1', 'http://www.company1.com', 'cv@company1.com',
'Volos', 39.362190, 22.942160, 'Programming'),
('PHP Developer', 'Company 8', 'http://www.company8.com', 'cv@company8.com',
'Patras', 38.246639, 21.734573, 'Programming'),
('Android Developer', 'Company 11', 'http://www.company11.gr', 'hr@company11.gr',
'Athens', 37.983810, 23.727539, 'Programming'),
('Web Designer', 'Company 3', 'http://www.company3.gr', 'cv@company3.gr',
'Kallithea', 37.955894, 23.702099, 'Design'),
('UX Designer', 'Company 10', 'http://www.company10.com', 'cv@company10.com',
'Peristeri', 38.017666, 23.685926, 'Design'),
('Sales Manager', 'Company 2', 'http://www.company2.com', 'cv@company2.com',
'Lamia', 38.871044, 22.381462, 'Business'),
('System Administrator', 'Company 7', 'http://www.company7.gr', 'cv@company7.gr',
'Thessaloniki', 40.640064, 22.944420, 'Admin'),
('Data Analyst', 'Company 9', 'http://www.company9.com', 'info@company9.gr',
'Larissa', 39.639023, 22.419125, 'Admin'),
('Graphic Designer', 'Company 5', 'http://www.company5.com', 'cv@company5.com',
'Heraklion', 35.338734, 25.144213, 'Content'),
('Technical Support', 'Company 6', 'http://www.company6.gr', 'cv@company6.gr',
'Marousi', 38.054955, 23.807655, 'Support'),
('Data Entry', 'Company 12', 'http://www.company12.com', 'hr@company12.gr',
'Piraeus', 37.943012, 23.646955, 'Support');
```

```
-----  
--  
-- Δομή πίνακα για τον πίνακα `pages`  
--  
CREATE TABLE `pages` (  
  `id` int(11) NOT NULL,  
  `name` varchar(150) COLLATE utf8_unicode_ci NOT NULL,  
  `description` varchar(160) COLLATE utf8_unicode_ci DEFAULT NULL,  
  `url` varchar(150) COLLATE utf8_unicode_ci NOT NULL,  
  `content` text COLLATE utf8_unicode_ci  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;  
  
--  
-- Άδειασμα δεδομένων του πίνακα `pages`  
--  
INSERT INTO `pages` (`id`, `name`, `description`, `url`, `content`) VALUES  
(1, 'Map', 'Map', 'about', 'Blue: Programming, Red: Design, Green: Business, Yellow:  
Administration, White: Support, Black: Content');  
  
-----  
--  
-- Δομή πίνακα για τον πίνακα `subscriptions`  
--  
CREATE TABLE `subscriptions` (  
  `id` int(10) NOT NULL,  
  `email` varchar(255) COLLATE utf8_unicode_ci NOT NULL,  
  `category_id` tinyint(4) NOT NULL,  
  `city_id` tinyint(4) NOT NULL,  
  `token` varchar(50) COLLATE utf8_unicode_ci NOT NULL,  
  `last_sent` datetime DEFAULT NULL,  
  `is_confirmed` tinyint(4) NOT NULL DEFAULT '0',  
  `created` datetime NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;  
  
--  
-- Άδειασμα δεδομένων του πίνακα `subscriptions`  
--  
INSERT INTO `subscriptions` (`id`, `email`, `category_id`, `city_id`, `token`,  
  `last_sent`, `is_confirmed`, `created`) VALUES  
(1, 'angel.korop@gmail.com', 1, 0, '144b6fa04f2fba6f98993505477320ee', NULL, 1,  
  '2016-11-07 01:41:51');  
  
--  
-- Ευρετήρια για άχρηστους πίνακες  
--  
--  
-- Ευρετήρια για πίνακα `admin`  
--  
ALTER TABLE `admin`
```

```
    ADD PRIMARY KEY (`id`);

--
-- Ευρετήρια για πίνακα `applications`
--
ALTER TABLE `applications`
  ADD PRIMARY KEY (`id`);

--
-- Ευρετήρια για πίνακα `banlist`
--
ALTER TABLE `banlist`
  ADD PRIMARY KEY (`id`);

--
-- Ευρετήρια για πίνακα `blocks`
--
ALTER TABLE `blocks`
  ADD PRIMARY KEY (`id`);

--
-- Ευρετήρια για πίνακα `categories`
--
ALTER TABLE `categories`
  ADD PRIMARY KEY (`id`);

--
-- Ευρετήρια για πίνακα `cities`
--
ALTER TABLE `cities`
  ADD PRIMARY KEY (`id`);

--
-- Ευρετήρια για πίνακα `downloads`
--
ALTER TABLE `downloads`
  ADD PRIMARY KEY (`id`);

--
-- Ευρετήρια για πίνακα `input`
--
ALTER TABLE `input`
  ADD PRIMARY KEY (`id`);

--
-- Ευρετήρια για πίνακα `jobs`
--
ALTER TABLE `jobs`
  ADD PRIMARY KEY (`id`);

--
-- Ευρετήρια για πίνακα `pages`
--
ALTER TABLE `pages`
  ADD PRIMARY KEY (`id`);
```

```
--  
-- Ευρετήρια για πίνακα `subscriptions`  
--  
ALTER TABLE `subscriptions`  
  ADD PRIMARY KEY (`id`);  
  
--  
-- AUTO_INCREMENT για άχρηστους πίνακες  
--  
--  
-- AUTO_INCREMENT για πίνακα `admin`  
--  
ALTER TABLE `admin`  
  MODIFY `id` tinyint(4) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;  
--  
-- AUTO_INCREMENT για πίνακα `applications`  
--  
ALTER TABLE `applications`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=3;  
--  
-- AUTO_INCREMENT για πίνακα `banlist`  
--  
ALTER TABLE `banlist`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;  
--  
-- AUTO_INCREMENT για πίνακα `blocks`  
--  
ALTER TABLE `blocks`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;  
--  
-- AUTO_INCREMENT για πίνακα `categories`  
--  
ALTER TABLE `categories`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;  
--  
-- AUTO_INCREMENT για πίνακα `cities`  
--  
ALTER TABLE `cities`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=14;  
--  
-- AUTO_INCREMENT για πίνακα `downloads`  
--  
ALTER TABLE `downloads`  
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;  
--  
-- AUTO_INCREMENT για πίνακα `input`  
--  
ALTER TABLE `input`  
  MODIFY `id` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;  
--  
-- AUTO_INCREMENT για πίνακα `jobs`  
--  
ALTER TABLE `jobs`  
  MODIFY `id` int(11) UNSIGNED NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=17;
```

```

--
-- AUTO_INCREMENT για πίνακα `pages`
--
ALTER TABLE `pages`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
--
-- AUTO_INCREMENT για πίνακα `subscriptions`
--
ALTER TABLE `subscriptions`
  MODIFY `id` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION
*/;

```

MODELS

Applications.php

```

<?php
/**
 * Application class handles job application sent through JobFinder
 */

class Applications
{
    private $_job_id;

    public function __construct($job_id=null)
    {
        if (isset($job_id)) {
            $this->_job_id = $job_id;
        }
    }

    public function getJobTitle()
    {
        $job = R::load('jobs', $this->_job_id);
        return $job->title;
    }

    public function getJobTitleURL()
    {
        global $lang;
        $job = R::load('jobs', $this->_job_id);
        $job_url = $job->title . " {"$lang->t('jobs|at')} ". $job->company_name;
        return slugify($job_url);
    }

    public function applyForJob($data) {

        $apply = R::dispense('applications');

```



```

$apply->job_id = $this->_job_id;
$apply->cover_letter = $data['cover_letter'];
$apply->full_name = $data['full_name'];
$apply->email = $data['email'];
$apply->location = $data['location'];
$apply->websites = $data['websites'];
$apply->attachment = $data['attachment'];
$apply->token = $data['token'];
$apply->created = R::isoDateTime();
$id = R::store($apply);

$job = R::load('jobs', $this->_job_id);

$data['title'] = $job->title;
$data['recipient'] = $job->email;

$notif = new Notifications();
if ($notif->applyForJobMail($data)) {
    return $id;
}
return false;

}

public function countJobApplications()
{
    $count = R::count('applications', ' job_id=:job_id ', array(':job_id'=>$this->job_id));
    return $count;
}

public function getApplications($start)
{
    if (isset($this->_job_id)) {
        $apps = R::findAll('applications', " job_id=:job_id ORDER BY created DESC LIMIT :start, :limit ", array(':job_id'=>$this->_job_id, ':start'=>$start, ':limit'=>LIMIT));
    } else {
        $apps = R::findAll('applications', " ORDER BY created DESC LIMIT :start, :limit ", array(':start'=>$start, ':limit'=>LIMIT));
    }
    return $apps;
}

public function countApplications()
{
    if (isset($this->_job_id)) {
        $apps = R::count('applications', ' job_id=:job_id ORDER BY created DESC ', array(':job_id'=>$this->_job_id));
    } else {
        $apps = R::count('applications', ' ORDER BY created DESC ');
    }
    return $apps;
}
}

```

Categories.php

```

<?php
/**
 * Categories class handles category display and management
 */
class Categories {

    private $_id;

    public function __construct($id)
    {
        $this->_id = $id;
    }

    public function findCategory()
    {
        $category = R::findOne('categories', ' id=:id ', array(':id'=>$this->_id));
        return $category;
    }

    public function findCategoryJobs($start, $limit)
    {
        $jobs = R::findAll('jobs', " status=1 AND category=:category ORDER BY created DESC
LIMIT :start, :limit ",
            array(':category'=>$this->_id, ':start'=>$start, ':limit'=>$limit));
        return $jobs;
    }

    public function findAllCategoryJobs()
    {
        $jobs = R::findAll('jobs', " status=1 AND category=:category ORDER BY created DESC
LIMIT 0, 100",
            array(':category'=>$this->_id));
        return $jobs;
    }

    public function countCategoryJobs() {
        $count = R::count('jobs', " status=1 AND category=:category ", array(':category'=>$this-
>_id));
        return $count;
    }

    public function addCategory($data)
    {
        if (isset($data['id']) && $data['id'] > 0) {
            $category = R::load('categories', $data['id']);
        } else {
            $category = R::dispense('categories');
        }
        $category->name = $data['name'];
        $category->description = $data['description'];
        $category->url = $data['url'];
        $category->sort = $data['sort'];
        $id = R::store($category);
    }
}

```

```

    return $id;
}

public function deleteCategory()
{
    $count = R::count('jobs', " category=:category ", array(':category'=>$this->_id));
    if (!$count) {
        $category = R::load('categories', $this->_id);
        R::trash($category);
        return true;
    }
    return false;
}

public static function findCategories()
{
    $categories = R::findAll('categories', ' ORDER BY sort ASC ');
    return $categories;
}
}

```

Cities.php

```

<?php
/**
 * Cities class handles city display and management
 */

class Cities {

    private $_id;

    public function __construct($id)
    {
        $this->_id = $id;
    }

    public function findCity()
    {
        $city = R::findOne('cities', ' id=:id ', array(':id'=>$this->_id));
        return $city;
    }

    public function findCityJobs($start, $limit)
    {
        $jobs = R::findAll('jobs', " status=1 AND city=:city ORDER BY created DESC LIMIT :start,
:limit ",
            array(':city'=>$this->_id, ':start'=>$start, ':limit'=>$limit));
        return $jobs;
    }

    public function findAllCityJobs()
    {

```

```

    $jobs = R::findAll('jobs', " status=1 AND city=:city ORDER BY created DESC LIMIT 0, 100",
        array(':city'=>$this->_id));
    return $jobs;
}

public function countCityJobs() {
    $count = R::count('jobs', " status=1 AND city=:city ", array(':city'=>$this->_id));
    return $count;
}

public function addCity($data)
{
    if (isset($data['id']) && $data['id'] > 0) {
        $city = R::load('cities', $data['id']);
    } else {
        $city = R::dispense('cities');
    }
    $city->name = $data['name'];
    $city->description = $data['description'];
    $city->url = $data['url'];
    $city->sort = $data['sort'];
    $id = R::store($city);
    return $id;
}

public function deleteCity()
{
    $count = R::count('jobs', " city=:city ", array(':city'=>$this->_id));
    if (!$count) {
        $city = R::load('cities', $this->_id);
        R::trash($city);
        return true;
    }
    return false;
}

public static function findCities()
{
    $cities = R::findAll('cities', ' ORDER BY sort ASC ');
    return $cities;
}
}

```

Jobs.php

```

<?php
/**
 * Jobs class handles all operations related to jobs
 */

class Jobs
{

```

```

private $_id;
private $_job;

public function __construct($id=0)
{
    $this->_id = $id;
    $this->_job = R::load('jobs', $id);
}

public function countJobApplications()
{
    $count = R::count('applications', ' job_id=:job_id ', array(':job_id'=>$this->_id));
    return $count;
}

public function getJobFromToken($token)
{
    $job = R::findOne('jobs', ' id=:id AND token=:token ', array(':id'=>$this->_id,
'token'=>$token));
    if (isset($job) && $job->id) {
        return $job;
    }
    return false;
}

public function getJobCity($city_id)
{
    $city = R::load('cities', $city_id);
    return $city->name;
}

public function getJobCategory($category_id)
{
    $category = R::load('categories', $category_id);
    return $category->name;
}

public function showJobDetails()
{
    return $this->_job;
}

public function jobCreateUpdate($data, $status=INACTIVE)
{
    if (isset($data['id']) && $data['id'] != '' && $this-> getJobFromToken($data['token'])) {
        $job = R::load('jobs', $data['id']);
        $data['email'] = $job->email;
    } else {
        $job = R::dispense('jobs');
        $job->created = R::isoDateTime();
        $job->email = $data['email'];
        $job->token = $data['token'];
        $job->status = $status;
    }
}

```

```

$job->title = $data['title'];
$job->category = $data['category'];
$job->city = $data['city'];
$job->description = $data['description'];
$job->perks = $data['perks'];
$job->how_to_apply = $data['how_to_apply'];
$job->company_name = $data['company_name'];
$job->logo = $data['logo'];
$job->url = $data['url'];
$job->is_featured = $data['is_featured'];
$id = R::store($job);

$data['access'] = accessToken($id);

// send user an email if job is inactive
if (!$job->status && $data['step'] == 3) {
    $notif = new Notifications();
    if ($notif->jobCreateUpdateMail($data)) {
        return true;
    }
}

if ($id > 0) {
    return $id;
}
return false;
}

public function featureJob($token, $action='on')
{
    $job = $this->getJobFromToken($token);

    if ($job && $job->id) {
        $action = ($action == 'on') ? ACTIVE : INACTIVE;
        $job->is_featured = $action;
        R::store($job);
        return true;
    }
    return false;
}

public function deleteJob($token)
{
    $job = $this->getJobFromToken($token);
    if ($job && ($job->id)) {
        R::trash($job);
        return true;
    }
    return false;
}

public function expireJobs()
{
    $sql = "UPDATE jobs SET status = 0 WHERE created < (NOW() - INTERVAL ".
    EXPIRE_JOBS ." DAY)";

```

```

    if (R::exec($sql)) {
        return true;
    }
    return false;
}

public function activateJob($token)
{
    $job = $this->showJobDetails();
    if ($token == accessToken($job->id)) {
        $job->status = ACTIVE;
        R::store($job);
        return true;
    }
    return false;
}

public function deactivateJob($token)
{
    $job = $this->showJobDetails();
    if ($token == accessToken($job->id)) {
        $job->status = INACTIVE;
        R::store($job);
        return true;
    }
    return false;
}

public function getJobs($status=1, $category=1, $start=null, $limit=null)
{
    if (!is_null($start) && !is_null($limit)) {
        $jobs = R::findAll('jobs',
            " status=:status AND category=:category ORDER BY created DESC LIMIT :start,
:limit",
            array(':status'=>$status, ':category'=>$category, ':start'=>$start, ':limit'=>$limit));
    } else {
        $jobs = R::findAll('jobs',
            " status=:status AND category=:category ORDER BY created DESC",
            array(':status'=>$status, ':category'=>$category));
    }
    return $jobs;
}

public function getStatus()
{
    $job = $this->showJobDetails();
    return $job->status;
}

public function getSlugTitle()
{
    global $lang;
    $job = $this->showJobDetails();
    return slugify($job->title ." {"$lang->t('jobs|at')} " . $job->company_name);
}

public function getSeoTitle()

```

```

{
    global $lang;
    $job = $this->showJobDetails();
    return $job->title . " {"$lang->t('jobs|at')} " . $job->company_name;
}
}

```

Search.php

```

<?php
/**
 * Search class handles job search functionality
 */

class Search
{

    public function searchJobs($terms)
    {
        global $categories;
        $terms = splitTerms($terms);
        $terms = implode('|', $terms);

        try {
            foreach ($categories as $cat) {
                $jobs[$cat->id] = R::findAll('jobs', " status=1 AND category=:category AND (title
REGEXP :title OR description REGEXP :description OR perks REGEXP :perks OR
how_to_apply REGEXP :how_to_apply OR company_name REGEXP :company_name)
ORDER BY created",
                    array(':category'=>$cat->id,
                        ':title'=>$terms,
                        ':description'=>$terms,
                        ':perks'=>$terms,
                        ':how_to_apply'=>$terms,
                        ':company_name'=>$terms));
            }
            return $jobs;
        } catch (Exception $e) {
            return false;
        }
    }

    public function countJobs($terms)
    {
        $terms = splitTerms($terms);
        $terms = implode('|', $terms);

        try {
            $jobs = R::findAll('jobs', " status=1 AND (title REGEXP :title OR description REGEXP
:description OR perks REGEXP :perks OR how_to_apply REGEXP :how_to_apply OR
company_name REGEXP :company_name)",

```



```

        array(':title'=>$terms,
              ':description'=>$terms,
              ':perks'=>$terms,
              ':how_to_apply'=>$terms,
              ':company_name'=>$terms));
    }
    return count($jobs);
} catch (Exception $e) {
    return 0;
}
}
}
}

```

Subscriptions.php

```

<?php
/**
 * Subscriptions class handles user email subscriptions
 */

class Subscriptions
{
    protected $_email;
    protected $_category_id;
    protected $_city_id;

    public function __construct($email, $category_id=0, $city_id=0)
    {
        $this->_email = $email;
        $this->_category_id = $category_id;
        $this->_city_id = $city_id;
    }

    public function createSubscription($subscription_for)
    {
        if (!$this->isUserSubscribed()) {

            $user = R::dispense('subscriptions');
            $user->email = $this->_email;
            $user->category_id = $this->_category_id;
            $user->city_id = $this->_city_id;
            $user->token = token();
            $user->is_confirmed = 0;
            $user->created = R::isoDateTime();
            $id = R::store($user);

            $notif = new Notifications();
            $notif->createSubscriptionMail($id, $user->token, $user->email, $subscription_for);
            return $id;
        }
        return false;
    }
}

```

```

public function updateSubscription($id, $status=INACTIVE)
{
    global $categories, $cities;
    $user = R::load('subscriptions', $id);
    if (isset($user) && $user->id) {
        $user->is_confirmed = $status;
        R::store($user);
        if ($status) {
            $subscription_for = ($user->category_id) ? $categories[$user->category_id]['name'] :
            $cities[$user->city_id]['name'];
            $notif = new Notifications();
            $notif->updateSubscriptionMail($user->email, $id, $user->token, $subscription_for);
        }
        return $id;
    }
    return false;
}

public function sendEmailsToSubscribers($job_id)
{
    $notif = new Notifications();
    $notif->sendEmailsToSubscribersMail($job_id);
}

public function getUserSubscription($id, $token)
{
    $user = R::findOne('subscriptions', ' id=:id AND token=:token', array(':id'=>$id,
    ':token'=>$token));
    if ($user && $user->id) {
        return $user;
    }
    return false;
}

public function isUserSubscribed()
{
    $user = R::findOne('subscriptions', ' email=:email AND category_id=:category_id AND
    city_id=:city_id ',
    array(':email'=>$this->_email, ':category_id'=> $this->_category_id,
    ':city_id'=>$this->_city_id));
    if ($user && $user->id) {
        return $user->id;
    }
    return false;
}

public function getAllSubscriptions($start)
{
    $users = R::findAll('subscriptions', ' ORDER BY created DESC LIMIT :start, :limit ',
    array(':start'=>$start, ':limit'=>LIMIT));
    if (isset($users)) {
        return $users;
    }
    return false;
}

```

```

public function countSubscriptions()
{
    $count = R::count('subscriptions');
    return $count;
}

public function deleteSubscription($id, $token) {
    $user = $this->getUserSubscription($id, $token);
    R::trash($user);
}
}

```

CONTROLLERS

Admin.php

```

<?php
/**
 * Admin
 * JobFinder admin panel
 */

define('ADMIN_MANAGE', ADMIN_URL . 'manage');

/*
 * Admin
 * Show admin options
 */

$app->group('/admin', function () use ($app) {

    // admin default
    $app->get('/', function () use ($app) {
        if (userIsValid()) {
            $app->redirect(ADMIN_MANAGE);
        } else {
            $app->redirect(LOGIN_URL);
        }
    });

    // login admin form
    $app->get('/login', function () use ($app) {

        global $lang;

        $val = array('lang' => $lang,
            'seo_title' => APP_NAME,
            'seo_desc' => APP_DESC,
            'seo_url' => ADMIN_URL);

        if (trim($app->request->get('user')) == 'invalid') {

```

```

        $val['invalid'] = 'yes';
    }

    $app->render(ADMIN_THEME . 'login.php', $val);
});

// authenticate user
$app->post('/authenticate', 'isValidReferrer', function () use ($app) {

    global $lang;

    $data = $app->request->post();
    $admin = R::findOne('admin', ' email=:email AND password=:password ',
array(':email'=>$data['email'], ':password'=>sha1($data['password'])));
    if (isset($admin) && $admin->id) {

        $_SESSION['is_admin'] = true;
        $_SESSION['email'] = $data['email'];

        $app->redirect(ADMIN_MANAGE);
    } else {
        $app->flash('danger', $lang->t('admin|invalid_login'));
        $app->redirect(LOGIN_URL);
    }
});

// logout admin
$app->get('/logout', 'validateUser', function () use ($app) {

    global $lang;

    unset($_SESSION['email']);
    unset($_SESSION['is_admin']);
    $app->flash('success', $lang->t('admin|logout_success'));
    $app->redirect('login');
});

/*
 * Manage group
 * Manage inactive jobs, categories list, cities list
 */
$app->group('/manage', function () use ($app) {

    // manage inactive jobs
    $app->get('/', 'validateUser', function () use ($app) {

        global $categories;
        global $lang;

        $j = new Jobs();
        foreach ($categories as $cat) {
            $jobs[$cat->id] = $j->getJobs(INACTIVE, $cat->id);
        }

        $app->render(ADMIN_THEME . 'home.php',
            array('lang' => $lang,

```

```

        'jobs'=>$jobs));
    });

    /*
    * Manage categories group
    */
    $app->group('/categories', function () use ($app) {

        $app->post('/', 'isValidReferrer', 'validateUser', function () use ($app) {

            global $lang;

            $data = $app->request->post();
            $c = new Categories($data['id']);
            $c->addCategory($data);
            if ($data && $data['id'] != "")
                $message = $lang->t('admin|category_update');
            else {
                $message = $lang->t('admin|category_new');
            }
            $app->flash('success', $message);
            $app->redirect(ADMIN_MANAGE . '/categories');
        });

        $app->get('/:id(/:action)'), 'validateUser', function ($id=null,$action=null) use ($app) {

            global $lang;

            $category = null;
            $c = new Categories($id);
            if ($id && $action == 'edit') {
                $category = $c->findCategory();
            } elseif ($id && $action == 'delete') {
                if ($c->deleteCategory()) {
                    $app->flash('success', $lang->t('admin|category_delete'));
                } else {
                    $app->flash('danger', $lang->t('admin|category_not_delete'));
                }
            }
            $app->redirect(ADMIN_MANAGE . '/categories');
        }
        $categories = Categories::findCategories();

        $app->render(ADMIN_THEME . 'categories.edit.php', array('lang' => $lang,
        'categs'=>$categories, 'category'=>$category));

    });

    });

    /*
    * Manage cities group
    */
    $app->group('/cities', function () use ($app) {

        $app->post('/', 'isValidReferrer', 'validateUser', function () use ($app) {

```

```

    global $lang;

    $data = $app->request->post();
    $c = new Cities($data['id']);
    $c->addCity($data);
    if ($data && $data['id'] != "")
        $message = $lang->t('admin|city_update');
    else {
        $message = $lang->t('admin|city_new');
    }
    $app->flash('success', $message);
    $app->redirect(ADMIN_MANAGE . '/cities');
});

$app->get('/:id(/:action)'), 'validateUser', function ($id=null,$action=null) use ($app) {

    global $lang;

    $city = null;
    $c = new Cities($id);
    if ($id && $action == 'edit') {
        $city = $c->findCity();
    } elseif ($id && $action == 'delete') {
        $c->deleteCity();
        if ($c->deleteCity()) {
            $app->flash('success', $lang->t('admin|city_delete'));
        } else {
            $app->flash('danger', $lang->t('admin|city_not_delete'));
        }
        $app->redirect(ADMIN_MANAGE . '/cities');
    }
    $cities = Cities::findCities();

    $app->render(ADMIN_THEME . 'cities.edit.php', array('lang' => $lang, 'cits'=>$cities,
'city'=>$city));

});

});

});

/*
 * Jobs group
 * Admin jobs routes
 */
$app->group('/jobs', function () use ($app) {

    // upload jobs from csv
    $app->get('/upload', 'validateUser', function () use ($app) {

        global $lang;

        $app->render(ADMIN_THEME . 'upload.php', array('lang' => $lang,
'filestyle'=>ACTIVE));

```

```

});

// process csv file
$app->post('/upload', 'isValidReferrer', 'validateUser', function () use ($app) {

    global $lang;

    $data = array();
    if (isset($_FILES['csv']) && $_FILES['csv']['name'] != "") {
        $file = $_FILES['csv'];
        $path = ATTACHMENT_PATH;
        $data['csv'] = time() . '_' . $file['name'];
        $data['type'] = $file['type'];
        $data['csv_size'] = $file['size'];

        $csv = "{$path}{$data['csv']}";

        if ($data['type'] == 'text/csv' && move_uploaded_file($file['tmp_name'], $csv)) {

            $added = 0;
            $skipped = 0;

            $file = new SplFileObject($csv);
            $file->setFlags(SplFileObject::READ_CSV);
            foreach ($file as $data) {

                // check if the csv file has the right number of fields
                if (count($data) == CSV_FIELDS) {

                    $featured = (isset($data[9]) && $data[9] == 'featured') ? 1 : 0;

                    $job = array('title'=>$data[0],
                        'category'=>$data[1],
                        'city'=>$data[2],
                        'description'=>$data[3],
                        'perks'=>$data[4],
                        'how_to_apply'=>$data[5],
                        'company_name'=>$data[6],
                        'url'=>$data[7],
                        'email'=>$data[8],
                        'logo'=> "",
                        'is_featured'=>$featured,
                        'token'=>token(),
                        'step'=>3);

                    $j = new Jobs();
                    $j->jobCreateUpdate($job, ACTIVE);

                    $added++;
                } else {
                    $skipped++;
                }
            }

            // remove csv file
            if (file_exists($csv)) {

```

```

        unlink($csv);
    }
    $app->flash('success', $lang->t('admin|upload_success', $added));
    $app->redirect(ADMIN_URL . 'jobs/upload');

    } else {
        $app->flash('danger', $lang->t('admin|upload_invalid'));
        $app->redirect(ADMIN_URL . 'jobs/upload');
    }
    } else {
        $app->flash('danger', $lang->t('admin|upload_none'));
        $app->redirect(ADMIN_URL . 'admin/upload');
    }
}

});

// expire jobs after X days
$app->get('/expire', 'validateUser', function () use ($app) {

    global $lang;

    $j = new Jobs();
    $j->expireJobs();

    $app->flash('success', $lang->t('admin|expire_success'));
    $app->redirect(ADMIN_MANAGE);
});

// get job post form
$app->get('/new', 'validateUser', function () use ($app) {

    global $lang;

    $token = token();
    $app->render(ADMIN_THEME . 'job.new.php',
        array('lang' => $lang,
            'token'=>$token,
            'markdown'=>ACTIVE,
            'filestyle'=>ACTIVE));
});

// review job
$app->post('/review', 'isValidReferrer', 'validateUser', function () use ($app) {

    global $lang;

    $data = $app->request->post();
    $data = escape($data);

    if ($data['trap'] != "") {
        $app->redirect(ADMIN_URL . "jobs/new");
    }

    if (isset($_FILES['logo']) && $_FILES['logo']['name'] != "") {
        $file = $_FILES['logo'];
        $path = IMAGE_PATH;
    }
}

```



```

    $data['logo'] = time() .'_.' . $file['name'];
    $data['logo_type'] = $file['type'];
    $data['logo_size'] = $file['size'];

    $ext = strtolower(pathinfo($data['logo'], PATHINFO_EXTENSION));
    if (move_uploaded_file($file['tmp_name'], "{$path}{$data['logo']}") &&
isValidImageExt($ext)) {
        $resize = new ResizeImage("{$path}{$data['logo']}");
        $resize->resizeTo(LOGO_H, LOGO_W);
        $resize->saveImage("{$path}thumb_{$data['logo']}");
    } else {
        $data['logo'] = "";
    }
} else {
    $data['logo'] = "";
}

$data['is_featured'] = (isset($data['is_featured'])) ? ACTIVE : INACTIVE;

$j = new Jobs();
$data['step'] = 2;
$id = $j->jobCreateUpdate($data, ACTIVE);

$app->redirect(ADMIN_URL . "jobs/{$id}/edit/{$data['token']}");
});

// post job publish form
$app->post('/:id/publish/:token', 'isValidReferrer', 'validateUser', function ($id, $token) use
($app) {

    global $lang;

    $data = $app->request->post();
    $data = escape($data);

    $j = new Jobs($id);
    $job = $j->getJobFromToken($token);

    $data['is_featured'] = (isset($data['is_featured'])) ? 1 : 0;

    if ($data['trap'] != "") {
        $app->redirect(ADMIN_URL . "jobs/new");
    }

    if (isset($_FILES['logo']) && $_FILES['logo']['name'] != "") {
        $file = $_FILES['logo'];
        $path = IMAGE_PATH;
        $data['logo'] = time() .'_.' . $file['name'];
        $data['logo_type'] = $file['type'];
        $data['logo_size'] = $file['size'];

        $ext = strtolower(pathinfo($data['logo'], PATHINFO_EXTENSION));
        if (move_uploaded_file($file['tmp_name'], "{$path}{$data['logo']}") &&
isValidImageExt($ext)) {
            $resize = new ResizeImage("{$path}{$data['logo']}");
            $resize->resizeTo(LOGO_H, LOGO_W);

```

```

        $resize->saveImage("{$_path}thumb_{$data['logo']}");
    }
} else {
    $data['logo'] = $job->logo;
}

$data['step'] = 3;
$j->jobCreateUpdate($data, ACTIVE);
$app->redirect(ADMIN_URL . "jobs/{$id}/publish/{$token}");
});

// get publish job details
$app->get('/:id/publish/:token', 'validateUser', function ($id, $token) use ($app) {

    global $lang;

    $j = new Jobs($id);
    $job = $j->getJobFromToken($token);
    $title = $j->getSlugTitle();
    $city = $j->getJobCity($job->city);
    $category = $j->getJobCategory($job->category);

    if (isset($job) && $job->id) {
        $app->render(ADMIN_THEME . 'job.publish.php',
            array('lang' => $lang,
                'job'=>$job,
                'city'=>$city,
                'category'=>$category));
    } else {
        $app->redirect(ADMIN_URL . "jobs/{$id}/{title}");
    }
});

// edit job
$app->get('/:id/edit/:token', 'validateUser', function ($id, $token) use ($app) {

    global $lang;

    $j = new Jobs($id);
    $job = $j->getJobFromToken($token);

    if (isset($job) && $job->id) {
        $app->render(ADMIN_THEME . 'job.review.php',
            array('lang' => $lang,
                'job'=>$job,
                'markdown'=>ACTIVE,
                'filestyle'=>ACTIVE));
    } else {
        $job = $j->showJobDetails();
        $title = $j->getSlugTitle();

        $app->redirect(ADMIN_URL . "jobs/{$job->id}/{title}");
    }
});

// feature job

```

```

$app->get('/:id/feature/:action/:token', 'validateUser', function ($id, $action, $token) use
($app) {

    global $lang;

    $j = new Jobs($id);
    $title = $j->getSlugTitle();
    if ($j->featureJob($token, $action)) {
        $app->flash('success', $lang->t('admin|feature_success', $action));
        $app->redirect(ADMIN_URL . "jobs/{$id}/{$title}");
    } else {
        $app->flash('danger', $lang->t('admin|feature_error'));
        $app->redirect(ADMIN_URL . "jobs/{$id}/{$title}");
    }
});

// delete existing job
$app->get('/:id/delete/:token', 'validateUser', function ($id, $token) use ($app) {

    global $lang;

    $j = new Jobs($id);
    if ($j->deleteJob($token)) {
        $app->flash('success', $lang->t('admin|delete_success', $id));
        $app->redirect(ADMIN_URL);
    } else {
        $app->flash('danger', $lang->t('admin|delete_error', $id));
        $app->redirect(ADMIN_MANAGE);
    }
});

// activate job
$app->get('/:id/activate/:token', 'validateUser', function ($id, $token) use ($app) {

    global $lang;

    $j = new Jobs($id);
    if ($j->activateJob($token)) {
        $job = $j->showJobDetails();
        $title = $j->getSlugTitle();

        $notif = new Notifications();
        $notif->sendEmailsToSubscribersMail($id);

        $app->flash('success', $lang->t('admin|activate_success', $id));
        $app->redirect(ADMIN_URL . "jobs/{$job->id}/{$title}");
    } else {
        $app->flash('danger', $lang->t('admin|activate_error', $id));
        $app->redirect(ADMIN_URL . "jobs/{$id}");
    }
});

// deactivate job
$app->get('/:id/deactivate/:token', 'validateUser', function ($id, $token) use ($app) {

    global $lang;

```

```

    $j = new Jobs($id);
    if ($j->deactivateJob($token)) {
        $job = $j->showJobDetails();
        $title = $j->getSlugTitle();
        $app->flash('success', $lang->t('admin|deactivate_success', $id));
        $app->redirect(ADMIN_URL . "jobs/{$job->id}/{title}");
    } else {
        $app->flash('danger', $lang->t('admin|deactivate_error', $id));
        $app->redirect(ADMIN_URL . "jobs/$id");
    }
});

// show job information
$app->get('/:id(/:title)', 'validateUser', function ($id, $title=null) use ($app) {

    global $lang;

    $j = new Jobs($id);
    $job = $j->showJobDetails();
    $city = $j->getJobCity($job->city);
    $category = $j->getJobCategory($job->category);
    $applications = $j->countJobApplications();

    if (isset($job) && $job->id) {
        $app->render(ADMIN_THEME . 'job.show.php',
            array('lang' => $lang,
                'job'=>$job,
                'id'=>$id,
                'applications'=>$applications,
                'category'=>$category,
                'city'=>$city));
    } else {
        $app->flash('danger', $lang->t('admin|not_found'));
        $app->redirect(ADMIN_MANAGE);
    }
});

});

/*
 * Categories group
 * Admin job categories routes
 */
$app->group('/categories', function () use ($app) {

    $app->get('/', 'validateUser', function () use ($app) {
        $app->redirect(ADMIN_MANAGE);
    });

    // get category jobs
    $app->get('/:id(/:name(/:page))', 'validateUser', function ($id, $name=null, $page=1) use
($app) {

        global $lang;

```

```

        $id = (int)$id;
        $cat = new Categories($id);
        $start = getPaginationStart($page);
        $count = $cat->countCategoryJobs();
        $number_of_pages = ceil($count/LIMIT);

        $categ = $cat->findCategory();
        $jobs = $cat->findCategoryJobs($start, LIMIT);

        if (isset($categ) && $categ) {
            $app->render(ADMIN_THEME . 'categories.php',
                array('lang' => $lang,
                    'categ'=>$categ,
                    'jobs'=>$jobs,
                    'id' => $id,
                    'number_of_pages'=>$number_of_pages,
                    'current_page'=>$page,
                    'page_name'=>'categories'));
        } else {
            $app->redirect(ADMIN_MANAGE);
        }
    });

});

/*
 * Cities group
 * Admin job cities routes
 */
$app->group('/cities', function () use ($app) {

    $app->get('/', 'validateUser', function () use ($app) {
        $app->redirect(ADMIN_MANAGE);
    });

    // get category jobs
    $app->get('/:id(/:name(/:page))', 'validateUser', function ($id, $name=null, $page=1) use
($app) {

        global $lang;

        $id = (int)$id;
        $cit = new Cities($id);

        $start = getPaginationStart($page);
        $count = $cit->countCityJobs();
        $number_of_pages = ceil($count/LIMIT);

        $city = $cit->findCity();
        $jobs = $cit->findCityJobs($start, LIMIT);

        if (isset($city) && $city) {
            $app->render(ADMIN_THEME . 'cities.php',
                array('lang' => $lang,
                    'city'=>$city,
                    'jobs'=>$jobs,

```

```

        'id' => $id,
        'number_of_pages'=>$number_of_pages,
        'current_page'=>$page,
        'page_name'=>'cities'));
    } else {
        $app->redirect(ADMIN_MANAGE);
    }
});

});

/*
 * Pages group
 * Manage pages
 */
$app->group('/pages', function () use ($app) {

    $app->post('/', 'isValidReferrer', 'validateUser', function () use ($app) {

        global $lang;

        $data = $app->request->post();
        $data = escape($data);

        $p = new Pages();
        $p->addToPageList($data);
        $method = (isset($data['id']) && $data['id'] > 0) ? 'edited' : 'added';
        $app->flash('success', $lang->t('admin|page_success', $method));
        $app->redirect(ADMIN_URL . 'pages');

    });

    $app->get('/new', 'validateUser', function () use ($app) {

        global $lang;

        $app->render(ADMIN_THEME . 'pages.new.php', array('lang' => $lang,
'method'=>'new', 'markdown'=>ACTIVE));

    });

    $app->get('/edit/:id', 'validateUser', function ($id) use ($app) {

        global $lang;

        $p = new Pages();
        $page = $p->showPage($id);
        $app->render(ADMIN_THEME . 'pages.edit.php', array('lang' => $lang, 'page'=>$page,
'method'=>'edit', 'markdown'=>ACTIVE));

    });

    $app->get('/delete/:id', 'validateUser', function ($id) use ($app) {

        global $lang;

```

```

    $p = new Pages();
    $p->deleteFromPage($id);
    $app->flash('success', $lang->t('admin|page_delete'));
    $app->redirect(ADMIN_URL . 'pages');
});

$app->get('/:page)', 'validateUser', function ($page=1) use ($app) {

    global $lang;

    $p = new Pages();

    $start = getPaginationStart($page);
    $count = $p->countPageList();
    $number_of_pages = ceil($count/LIMIT);

    $pages = $p->showPageList($start, LIMIT);

    $app->render(ADMIN_THEME . 'pages.php',
        array('lang' => $lang,
            'pages'=>$pages,
            'number_of_pages'=>$number_of_pages,
            'current_page'=>$page,
            'page_name'=>'pages'));

});

});

/*
 * Blocks group
 * Manage block content
 */
$app->group('/blocks', function () use ($app) {

    $app->post('/', 'isValidReferrer', 'validateUser', function () use ($app) {

        global $lang;

        $data = $app->request->post();

        $b = new Blocks();
        $b->addToBlockList($data);
        $method = (isset($data['id']) && $data['id'] > 0) ? 'edited' : 'added';
        $app->flash('success', $lang->t('admin|block_success', $method));
        $app->redirect(ADMIN_URL . 'blocks');

    });

    $app->get('/new', 'validateUser', function () use ($app) {

        global $lang;

        $app->render(ADMIN_THEME . 'blocks.new.php', array('lang' => $lang,
            'method'=>'new'));
    });

```

```

    });

    $app->get('/edit/:id', 'validateUser', function ($id) use ($app) {

        global $lang;

        $b = new Blocks();
        $block = $b->showBlock($id);
        $app->render(ADMIN_THEME . 'blocks.edit.php', array('lang' => $lang, 'block'=>$block,
'method'=>'edit'));

    });

    $app->get('/delete/:id', 'validateUser', function ($id) use ($app) {

        global $lang;

        $b = new Blocks();
        $b->deleteFromBlock($id);
        $app->flash('success', $lang->t('admin|block_delete'));
        $app->redirect(ADMIN_URL . 'blocks');

    });

    $app->get('(/(:page))', 'validateUser', function ($page=1) use ($app) {

        global $lang;

        $b = new Blocks();

        $start = getPaginationStart($page);
        $count = $b->countBlockList();
        $number_of_pages = ceil($count/LIMIT);

        $blocks = $b->showBlockList($start, LIMIT);

        $app->render(ADMIN_THEME . 'blocks.php',
            array('lang' => $lang,
                'blocks'=>$blocks,
                'number_of_pages'=>$number_of_pages,
                'current_page'=>$page,
                'page_name'=>'blocks'));

    });

});

/*
 * Banlist group
 * Manage ban list
 */
$app->group('/ban', function () use ($app) {

    $app->post('/', 'isValidReferrer', 'validateUser', function () use ($app) {

        global $lang;

```



```

    $ban = new Banlist();
    $data = $app->request->post();
    $ban->addToList($data['type'], $data['value']);

    $app->flash('success', $lang->t('admin|ban_add', $data['value']));
    $app->redirect(ADMIN_URL . 'ban');

});

$app->get('/delete/:id', 'validateUser', function ($id) use ($app) {

    global $lang;

    $ban = new Banlist();
    $value = $ban->deleteFromList($id);

    $app->flash('success', $lang->t('admin|ban_remove', $value));
    $app->redirect(ADMIN_URL . 'ban');

});

$app->get('/(:page)', 'validateUser', function ($page=1) use ($app) {

    global $lang;

    $ban = new Banlist();

    $start = getPaginationStart($page);
    $count = $ban->countBanList();
    $number_of_pages = ceil($count/LIMIT);

    $list = $ban->showBanList($start, LIMIT);

    $app->render(ADMIN_THEME . 'banlist.php',
        array('lang' => $lang,
            'list'=>$list,
            'number_of_pages'=>$number_of_pages,
            'current_page'=>$page,
            'page_name'=>'banlist'));

});

});

/*
 * Applications group
 * Admin job applications routes
 */
$app->group('/applications', function () use ($app) {

    // show all job applications
    $app->get('/(:page)', 'validateUser', function ($page=1) use ($app) {

        global $lang;

        $a = new Applications();
        $start = getPaginationStart($page);

```

```

$count = $a->countApplications();
$number_of_pages = ceil($count/LIMIT);

$applications = $a->getApplications($start);

$app->render(ADMIN_THEME . 'applications.php',
    array('lang' => $lang,
        'applications'=>$applications,
        'number_of_pages'=>$number_of_pages,
        'current_page'=>$page,
        'page_name'=>'applications',
        'count'=>$count));
});

// get job applications
$app->get('/jobs:id(/:page)', 'validateUser', function ($id, $page=1) use ($app) {

    global $lang;

    $a = new Applications($id);
    $start = getPaginationStart($page);
    $count = $a->countApplications($id);
    $number_of_pages = ceil($count/LIMIT);

    $j = new Jobs($id);
    $title = $j->getSeoTitle();

    $applications = $a->getApplications($start);

    $app->render(ADMIN_THEME . 'applications.job.php',
        array('lang' => $lang,
            'applications'=>$applications,
            'number_of_pages'=>$number_of_pages,
            'current_page'=>$page,
            'page_name'=>'applications',
            'count'=>$count,
            'title'=>$title,
            'id'=>$id));

});

});

$app->group('/subscribers', function () use ($app) {

    $app->get('/(:page)', 'validateUser', function ($page=1) use ($app) {

        global $lang;

        $s = new Subscriptions("");

        $start = getPaginationStart($page);
        $count = $s->countSubscriptions();
        $number_of_pages = ceil($count/LIMIT);

        $users = $s->getAllSubscriptions($start);

```

```

$app->render(ADMIN_THEME . 'subscribers.php',
    array('lang' => $lang,
        'users'=>$users,
        'number_of_pages'=>$number_of_pages,
        'current_page'=>$page,
        'count'=>$count,
        'page_name'=>'subscribers'));

});

$app->get('/:id/:action/:token', 'validateUser', function ($id, $action, $token) use ($app) {

    global $lang;

    $s = new Subscriptions("");
    $user = $s->getUserSubscription($id, $token);

    if (isset($user)) {
        switch ($action) {
            case 'approve':
                $s->updateSubscription($id, ACTIVE);
                $app->flash('success', $lang->t('admin|subscribe_confirm'));
                break;
            case 'deactivate':
                $s->updateSubscription($id, INACTIVE);
                $app->flash('success', $lang->t('admin|subscribe_deactivate'));
                break;
            case 'delete':
                $s->deleteSubscription($id, $token);
                $app->flash('success', $lang->t('admin|subscribe_delete'));
                break;
        }
    }
    $app->redirect(ADMIN_URL . 'subscribers');

});

});

});

```

Applications.php

```

<?php
/**
 * Applications
 * Job application form submission
 */

$app->group('/apply', function () use ($app) {

    // get job post form

```

```

$app->get('/:job_id(/)', 'isBanned', function ($job_id) use ($app) {

    global $lang;

    $token = token();

    $seo_title = $lang->t('apply|seo_title') . ' | ' . APP_NAME;
    $seo_desc = $lang->t('apply|seo_desc') . ' | ' . APP_NAME;
    $seo_url = BASE_URL . 'apply/new';

    $job = new Applications($job_id);
    $title = $job->getJobTitle();

    $app->render(THEME_PATH . 'apply.new.php',
        array('lang' => $lang,
            'seo_url'=>$seo_url,
            'seo_title'=>$seo_title,
            'seo_desc'=>$seo_desc,
            'token'=>$token,
            'job_id'=>$job_id,
            'job_title'=>$title,
            'filestyle'=>ACTIVE));
});

// submit job application
$app->post('/submit', 'isValidReferrer', 'isBanned', function () use ($app) {

    global $lang;

    $data = $app->request->post();

    if (Banlist::isBanned('email', $data['email'])
        || Banlist::isBanned('ip', $_SERVER['REMOTE_ADDR'])) {
        $app->flash('danger', $lang->t('apply|email_ip_banned'));
        $app->redirect(BASE_URL . "apply/{$data['job_id']}");
    }

    $data = escape($data);

    if ($data['trap'] != "") {
        $app->redirect(BASE_URL . "apply/{$data['job_id']}");
    }

    if (isset($_FILES['attachment']) && $_FILES['attachment']['name'] != "") {
        $file = $_FILES['attachment'];
        $path = ATTACHMENT_PATH;
        $attachment = time() . '_' . $file['name'];
        $data['attachment_type'] = $file['type'];
        $data['attachment_size'] = $file['size'];

        if (move_uploaded_file($file['tmp_name'], "{$path}{$attachment}")) {
            $data['attachment'] = $attachment;
        }
    } else {
        $data['attachment'] = "";
    }
}

```

```

    $apply = new Applications($data['job_id']);
    if ($apply->applyForJob($data)) {
        $app->flash('success', $lang->t('apply|msg_success'));
    } else {
        $app->flash('danger', $lang->t('apply|msg_fail'));
    }
    $title = $apply->getJobTitleURL();
    $app->redirect(BASE_URL . "jobs/{$data['job_id']}/{$title}");
});
});

```

Categories.php

```

<?php
/**
 * Categories
 * Shows jobs per category
 */

$app->group('/categories', function () use ($app) {

    // get categories index
    $app->get('/', function () use ($app) {
        $app->redirect(BASE_URL);
    });

    // rss category jobs
    $app->get('/:id/:name/rss', function ($id, $name) use ($app) {

        global $lang;

        $id = (int)$id;
        $cat = new Categories($id);
        $info = $cat->findCategory();

        $jobs = $cat->findAllCategoryJobs();

        $app->response->headers->set('Content-Type', 'application/rss+xml;charset=utf-8');

        $xml = new SimpleXMLElement('<rss version="2.0"></rss>');
        $xml->addChild('channel');
        $xml->channel->addChild('title', htmlentities(escapeXML($info->name)) . " " . $lang->t('jobs|jobs') . ' | ' . APP_NAME);
        $xml->channel->addChild('link', BASE_URL . "categories/{$info->id}/{$info->url}");
        $xml->channel->addChild('description', htmlentities(escapeXML($info->description)));
        foreach ($jobs as $job) {
            $item = $xml->channel->addChild('item');
            $item->addChild('title', htmlentities(escapeXML($job->title)));
            $item->addChild('link', BASE_URL . "jobs/{$job->id}/" . slugify($job->title . " {" . $lang->t('jobs|at') . " " . $job->company_name));
            $item->addChild('description', htmlentities(escapeXML($job->description)));
        }
    });
});

```

```

        $guid = $item->addChild('guid', $job->id . '@' . BASE_URL);
        $guid->addAttribute('isPermaLink', "false");
        $item->addChild('pubDate', date(DATE_RSS, strtotime($job->created)));
    }
    $dom = new DOMDocument();
    $dom->preserveWhiteSpace = false;
    $dom->formatOutput = true;
    $dom->loadXML(html_entity_decode($xml->asXML()));
    echo $dom->saveXML();

});

// get category jobs
$app->get('/:id(/:name(/:page))', function ($id, $name=null, $page=1) use ($app) {

    global $lang;

    $id = (int)$id;
    $cat = new Categories($id);
    $categ = $cat->findCategory();

    if (isset($categ) && $categ) {

        $start = getPaginationStart($page);
        $count = $cat->countCategoryJobs();
        $number_of_pages = ceil($count/LIMIT);
        $jobs = $cat->findCategoryJobs($start, LIMIT);

        $seo_title = $categ->name . ' | ' . APP_NAME;
        $seo_desc = excerpt($categ->description);
        $seo_url = BASE_URL . "categories/{$id}/{$name}";

        $app->render(THEME_PATH . 'categories.php',
            array('lang' => $lang,
                'seo_url'=>$seo_url,
                'seo_title'=>$seo_title,
                'seo_desc'=>$seo_desc,
                'categ'=>$categ,
                'jobs'=>$jobs,
                'id' => $id,
                'number_of_pages'=>$number_of_pages,
                'current_page'=>$page,
                'page_name'=>'categories'));
    } else {
        $app->flash('danger', $lang->t('alert|page_not_found'));
        $app->redirect(BASE_URL, 404);
    }

});
});

```

Cities.php

```

<?php
/**
 * Cities
 * Shows jobs per city
 */

$app->group('/cities', function () use ($app) {

    // get cities index
    $app->get('/', function () use ($app) {
        $app->redirect(BASE_URL);
    });

    // rss city jobs
    $app->get('/:id:/name/rss', function ($id, $name) use ($app) {

        global $lang;

        $id = (int)$id;
        $city = new Cities($id);
        $info = $city->findCity();

        $jobs = $city->findAllCityJobs();

        $app->response->headers->set('Content-Type', 'application/rss+xml;charset=utf-8');

        $xml = new SimpleXMLElement('<rss version="2.0"></rss>');
        $xml->addChild('channel');
        $xml->channel->addChild('title', htmlentities(escapeXML($info->name)) . " " . $lang->
>t('jobs|jobs') . ' | ' . APP_NAME);
        $xml->channel->addChild('link', BASE_URL . "cities/{$info->id}/{$info->url}");
        $xml->channel->addChild('description', htmlentities(escapeXML($info->description)));
        foreach ($jobs as $job) {
            $item = $xml->channel->addChild('item');
            $item->addChild('title', htmlentities(escapeXML($job->title)));
            $item->addChild('link', BASE_URL . "jobs/{$job->id}/". slugify($job->title . " {$lang-
>t('jobs|at')} " . $job->company_name));
            $item->addChild('description', htmlentities(escapeXML($job->description)));
            $guid = $item->addChild('guid', $job->id . '@' . BASE_URL);
            $guid->addAttribute('isPermaLink', "false");
            $item->addChild('pubDate', date(DATE_RSS, strtotime($job->created)));
        }
        $dom = new DOMDocument();
        $dom->preserveWhiteSpace = false;

        $dom->formatOutput = true;
        $dom->loadXML(html_entity_decode($xml->asXML()));
        echo $dom->saveXML();

    });

    // get city jobs

```

```

$app->get('/:id(/:name(/:page))', function ($id, $name=null, $page=1) use ($app) {

    global $lang;

    $id = (int)$id;
    $cit = new Cities($id);
    $city = $cit->findCity();

    if (isset($city) && $city) {

        $start = getPaginationStart($page);
        $count = $cit->countCityJobs();
        $number_of_pages = ceil($count/LIMIT);
        $jobs = $cit->findCityJobs($start, LIMIT);

        $seo_title = $city->name . ' | ' . APP_NAME;
        $seo_desc = excerpt($city->description);
        $seo_url = BASE_URL . "cities/{$id}/{ $name}";

        $app->render(THEME_PATH . 'cities.php',
            array('lang' => $lang,
                'seo_url'=>$seo_url,
                'seo_title'=>$seo_title,
                'seo_desc'=>$seo_desc,
                'city'=>$city,
                'jobs'=>$jobs,
                'id' => $id,
                'number_of_pages'=>$number_of_pages,
                'current_page'=>$page,
                'page_name'=>'cities'));
    } else {
        $app->flash('danger', $lang->t('alert|page_not_found'));
        $app->redirect(BASE_URL, 404);
    }

});
});

```

Jobs.php

```

<?php
/**
 * Jobs
 * Shows job information and job posting option
 */

$app->group('/jobs', function () use ($app) {

    // get job post form
    $app->get('/new', 'isJobPostAllowed', 'isBanned', function () use ($app) {

        global $lang;
        $token = token();

        $seo_title = 'Post new job | ' . APP_NAME;

```



```

$seo_desc = 'Post a new job at '. APP_NAME;
$seo_url = BASE_URL .'jobs/new';

$app->render(THEME_PATH . 'job.new.php',
    array('lang' => $lang,
        'seo_url'=>$seo_url,
        'seo_title'=>$seo_title,
        'seo_desc'=>$seo_desc,
        'token'=>$token,
        'markdown'=>ACTIVE,
        'filestyle'=>ACTIVE));
});

// review job
$app->post('/review', 'isJobPostAllowed', 'isBanned', 'isValidReferrer', function () use ($app)
{
    global $lang;

    $data = $app->request->post();

    if (Banlist::isBanned('email', $data['email'])
        || Banlist::isBanned('ip', $_SERVER['REMOTE_ADDR'])) {
        $app->flash('danger', $lang->t('alert|ip_banned'));
        $app->redirect(BASE_URL . "jobs/new");
    }

    $data = escape($data);

    if ($data['trap'] != "") {
        $app->redirect(BASE_URL . "jobs/new");
    }

    if (isset($_FILES['logo']) && $_FILES['logo']['name'] != "") {
        $file = $_FILES['logo'];
        $path = IMAGE_PATH;
        $data['logo'] = time() . '_' . $file['name'];
        $data['logo_type'] = $file['type'];
        $data['logo_size'] = $file['size'];
        $ext = strtolower(pathinfo($data['logo'], PATHINFO_EXTENSION));

        if (move_uploaded_file($file['tmp_name'], "{$path}{$data['logo']}") &&
            isValidImageExt($ext)) {
            $resize = new ResizeImage("{$path}{$data['logo']}");
            $resize->resizeTo(LOGO_H, LOGO_W);
            $resize->saveImage("{$path}thumb_{$data['logo']}");
        } else {
            $data['logo'] = "";
        }
    } else {
        $data['logo'] = "";
    }

    $data['is_featured'] = (isset($data['is_featured'])) ? 1 : 0;

    $j = new Jobs();

```

```

$data['step'] = 2;

if ($data['email'] != "" && filter_var($data['email'], FILTER_VALIDATE_EMAIL)) {
    $id = $j->jobCreateUpdate($data);
}

if (!isset($id)) {
    $app->flash('danger', $lang->t('alert|error_encountered'));
    $app->redirect(BASE_URL . "jobs/new");
} else {
    $app->redirect(BASE_URL . "jobs/{$id}/edit/{$data['token']}");
}

});

// post job publish form
$app->post('/:id/publish/:token', 'isJobPostAllowed', 'isBanned', 'isValidReferrer', function
($id, $token) use ($app) {

    global $lang;

    $data = $app->request->post();
    $data = escape($data);

    $j = new Jobs($id);
    $job = $j->getJobFromToken($token);

    $data['is_featured'] = (isset($data['is_featured'])) ? 1 : 0;

    if ($data['trap'] != "" || !$job) {
        $app->redirect(BASE_URL . "jobs/new");
    }
    if (isset($_FILES['logo']) && $_FILES['logo']['name'] != "") {
        $file = $_FILES['logo'];
        $path = IMAGE_PATH;
        $data['logo'] = time() . '_' . $file['name'];
        $data['logo_type'] = $file['type'];
        $data['logo_size'] = $file['size'];

        $ext = strtolower(pathinfo($data['logo'], PATHINFO_EXTENSION));
        if (move_uploaded_file($file['tmp_name'], "{$path}{$data['logo']}") &&
isValidImageExt($ext)) {
            $resize = new ResizeImage("{$path}{$data['logo']}");
            $resize->resizeTo(LOGO_H, LOGO_W);
            $resize->saveImage("{$path}thumb_{$data['logo']}");
        } else {
            $data['logo'] = "";
        }
    } else {
        $data['logo'] = $job->logo;
    }

    $data['step'] = 3;

    $j->jobCreateUpdate($data);
    if (!$j->getStatus()) {

```

```

        $app->flash('success', $lang->t('alert|activation_email', $job->email));
    } else {
        $app->flash('success', $lang->t('alert|edit_successful'));
    }

    $app->redirect(BASE_URL . "jobs/{$id}/publish/{$token}");
});

// get publish job details
$app->get('/:id/publish/:token', 'isJobPostAllowed', 'isBanned', 'isValidReferrer', function ($id,
$token) use ($app) {

    global $lang;

    $j = new Jobs($id);
    $job = $j->getJobFromToken($token);

    $title = $j->getSlugTitle();
    $city = $j->getJobCity($job->city);
    $category = $j->getJobCategory($job->category);

    if (isset($job) && $job->id) {

        $seo_title = clean($job->title) . ' | '. APP_NAME;
        $seo_desc = excerpt($job->description);
        $seo_url = BASE_URL . "jobs/{$id}/{title}";

        $app->render(THEME_PATH . 'job.publish.php',
            array('lang' => $lang,
                'seo_url'=>$seo_url,
                'seo_title'=>$seo_title,
                'seo_desc'=>$seo_desc,
                'job'=>$job,
                'city'=>$city,
                'category'=>$category));
    } else {
        $app->flash('danger', $lang->t('alert|error_encountered'));
        $app->redirect(BASE_URL . "jobs/{$id}/{title}");
    }
});

// edit job
$app->get('/:id/edit/:token', 'isJobPostAllowed', 'isBanned', function ($id, $token) use ($app)
{

    global $lang;

    $j = new Jobs($id);
    $job = $j->getJobFromToken($token);
    $title = $j->getSlugTitle();

    if (isset($job) && $job) {
        $seo_title = 'Edit job | '. APP_NAME;
        $seo_desc = APP_DESC;
        $seo_url = BASE_URL;
    }
}

```

```

    $app->render(THEME_PATH . 'job.review.php',
        array('lang' => $lang,
            'seo_url'=>$seo_url,
            'seo_title'=>$seo_title,
            'seo_desc'=>$seo_desc,
            'job'=>$job,
            'markdown'=>ACTIVE,
            'filestyle'=>ACTIVE));
    } else {
        $app->flash('danger', $lang->t('alert|error_encountered'));
        $app->redirect(BASE_URL . "jobs/{$id}/{title}");
    }
});

// delete existing job
$app->get('/:id/delete/:token', 'isJobPostAllowed', 'isBanned', function ($id, $token) use
($app) {

    global $lang;

    $j = new Jobs($id);
    if ($j->deleteJob($token)) {
        $app->flash('success', $lang->t('admin|delete_success', $id));
        $app->redirect(BASE_URL);
    } else {
        $app->flash('danger', $lang->t('admin|delete_error', $id));
        $app->redirect(BASE_URL);
    }

});

// activate job
$app->get('/:id/activate/:token', 'isJobPostAllowed', 'isBanned', function ($id, $token) use
($app) {

    global $lang;

    $j = new Jobs($id);
    $title = $j->getSlugTitle();

    if ($j->activateJob($token)) {

        $notif = new Notifications();
        $notif->sendEmailsToSubscribersMail($id);

        $job = $j->showJobDetails();
        $app->flash('success', $lang->t('admin|activate_success', $id));
        $app->redirect(BASE_URL . "jobs/{$id}/{title}");
    } else {
        $app->flash('danger', $lang->t('admin|activate_error', $id));
        $app->redirect(BASE_URL);
    }
});

// deactivate job

```

```

$app->get('/:id/deactivate/:token', 'isJobPostAllowed', 'isBanned', function ($id, $token) use
($app) {

    global $lang;

    $j = new Jobs($id);
    var_dump($j);
    if ($j->deactivateJob($token)) {
        $job = $j->showJobDetails();
        $title = $j->getSlugTitle();
        $app->flash('success', $lang->t('admin|deactivate_success', $id));
        $app->redirect(BASE_URL);
    } else {
        $app->flash('danger', $lang->t('admin|deactivate_error', $id));
        $app->redirect(BASE_URL);
    }
});

// show job information
$app->get('/:id(/:title)', function ($id, $title=null) use ($app) {

    global $lang;

    $j = new Jobs($id);
    $job = $j->showJobDetails();
    $city = $j->getJobCity($job->city);
    $category = $j->getJobCategory($job->category);
    $applications = $j->countJobApplications();

    if (isset($job) && $job->id && $job->status==ACTIVE) {

        $seo_title = clean($job->title) . ' | ' . APP_NAME;
        $seo_desc = excerpt($job->description);
        $seo_url = BASE_URL . "jobs/{$id}/{$title}";

        $app->render(THEME_PATH . 'job.show.php',
            array('lang' => $lang,
                'seo_url' => $seo_url,
                'seo_title' => $seo_title,
                'seo_desc' => $seo_desc,
                'job' => $job,
                'id' => $id,
                'applications' => $applications,
                'category' => $category,
                'city' => $city));
    } else {
        $app->redirect(BASE_URL);
    }
});
});

```

Search.php

```

<?php
/**
 * Search
 * Search for jobs
 */

$app->group('/search', function () use ($app) {

    $app->post('/', function () use ($app) {

        $data = $app->request->post('terms');

        $terms = escape($data);
        $terms = urlencode($terms);

        $app->redirect(BASE_URL . "search/{$terms}");
    });

    $app->get('/:terms', 'isBanned', function ($terms="") use ($app) {

        global $lang;
        $s = new Search();

        $jobs = $s->searchJobs($terms);
        $count = $s->countJobs($terms);

        $app->render(THEME_PATH . 'search.php',
            array('lang' => $lang,
                'terms'=>$terms,
                'count'=>$count,
                'seo_url'=>BASE_URL . 'search',
                'seo_title'=>$lang->t('search|search_result') . ' . APP_NAME,
                'seo_desc'=>$lang->t('search|search_result') . ' . APP_NAME,
                'jobs'=>$jobs));

    });

});

```

Subscriptions.php

```

<?php
/**
 * Subscriptions
 * Create and manage email subscriptions
 */

$app->group('/subscribe', function () use ($app) {

    $app->post('/new', 'isBanned', function () use ($app) {

```

```

global $categories, $cities;
global $lang;

$data = $app->request->post();

$redirect = ($data['category_id'] > 0) ? 'categories' : 'cities';
$id = ($data['category_id'] > 0) ? $data['category_id'] : $data['city_id'];
if ($data['category_id'] > 0) {
    $subscription_for = $categories[$data['category_id']]['name'];
} else {
    $subscription_for = $cities[$data['city_id']]['name'];
}

if ($data['trap'] == "") {
    $subscribe = new Subscriptions($data['email'], $data['category_id'], $data['city_id']);
    if ($subscribe->createSubscription($subscription_for)) {
        $app->flash('success', $lang->t('subscribe|confirm_email'));
    } else {
        $app->flash('danger', $lang->t('subscribe|existing'));
    }
    $app->redirect(BASE_URL . "{$redirect}/{id}");
} else {
    $app->flash('danger', $lang->t('subscribe|not_allowed'));
    $app->redirect(BASE_URL . "{$redirect}/{id}");
}
});

$app->get('/:id/:action/:token', 'isBanned', function ($id, $action, $token) use ($app) {

    global $lang;

    $status = ($action == 'confirm') ? ACTIVE : INACTIVE;
    $s = new Subscriptions("");
    $user = $s->getUserSubscription($id, $token);
    if ($user) {
        $s->updateSubscription($id, $status);
        if ($status == ACTIVE) {
            $app->flash('success', $lang->t('subscribe|confirmed'));
        } else {
            $app->flash('success', $lang->t('subscribe|cancel'));
        }
        $app->redirect(BASE_URL);
    } else {
        $app->flash('danger', $lang->t('subscribe|confirm_error'));
        $app->redirect(BASE_URL);
    }

});
});

```

VIEWS

VIEWS - DEFAULT

Home.php

```

<?php include 'header.php'; ?>
<?php include 'search-form.php'; ?>

<?php include 'flash.php'; ?>
<?php foreach($categories as $category): ?>
<a name="<?php _e($category->url); ?>"></a>
<h3><?php _e($category->name); ?> <?php echo $lang->t('jobs|jobs'); ?> <a href="<?php
_e(BASE_URL ."categories/{<?php _e($category->id)}/{<?php _e($category->url)/rss"); ?>" target="_blank"
class="badge job-rss">RSS</a></h3>
<div class="list-group">
<?php foreach($jobs[$category->id] as $job): ?>
  <a class="list-group-item <?php if ($job->is_featured): ?>job-highlight<?php endif; ?>"
href="<?php _e(BASE_URL ."jobs/{<?php _e($job->id)}/". slugify($job->title ." {$lang->t('jobs|at')} "). $job-
>company_name)); ?>">
  <h4>
    <span class="job-title"><?php _e($job->title); ?></span>&nbsp;
    <span class="job-company"><?php _e($job->company_name); ?></span>
    <span class="badge pull-right"><?php niceDate($job->created); ?></span>
  </h4>
</a>
<?php endforeach; ?>
<a class="list-group-item" href="<?php _e(BASE_URL ."categories/{<?php _e($category->id)}/{<?php _e($category->url)"); ?>">
  <h5><?php echo $lang->t('jobs|view_all', $category->name); ?></h5>
</a>
</div>
<p class="pull-right"><a href="#top"><?php echo $lang->t('jobs|back_to_top'); ?></a></p>

<?php endforeach; ?>

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

```

Categories.php

```

<?php include 'header.php'; ?>
<?php include 'search-form.php'; ?>

<?php include 'flash.php'; ?>
<h3><?php _e($categ->name); ?> <?php echo $lang->t('jobs|jobs'); ?> <a href="<?php
_e(BASE_URL ."categories/{<?php _e($categ->id)}/{<?php _e($categ->url)/rss"); ?>" target="_blank" class="badge
job-rss">RSS</a></h3>
<div class="list-group">

```



```

<?php foreach($jobs as $job): ?>
  <a class="list-group-item <?php if ($job->is_featured): ?>job-highlight<?php endif; ?>"
  href="<?php _e(BASE_URL ."jobs/{<?php echo $job->id}&#125;/". slugify($job->title ." {<?php echo $lang->t('jobs|at')} "). $job-
  >company_name)); ?>">
    <h4>
      <span class="job-title"><?php _e($job->title); ?>&#125;&#125;</span>&#125;
      <span class="job-company"><?php _e($job->company_name); ?></span>
      <span class="badge pull-right"><?php niceDate($job->created); ?></span>
    </h4>
  </a>
<?php endforeach; ?>
</div>

<form class="form-inline" role="form" method="post" action="<?php _e(BASE_URL .
"subscribe/new"); ?>">
  <div class="form-group">
    <input type="email" class="form-control input-lg" id="email" name="email"
    placeholder="<?php echo $lang->t('jobs|enter_email'); ?>" required />
    <input type="hidden" name="category_id" value="<?php echo $categ->id; ?>" />
    <input type="hidden" name="city_id" value="0" />
    <input type="hidden" name="trap" value="" />
    <input type="hidden" name="<?php _e($csrf_key); ?>" value="<?php _e($csrf_token); ?>">
  </div>
  <button type="submit" class="btn btn-default btn-lg"><?php echo $lang-
  >t('jobs|subscribe_to', $categ->name); ?></button>
</form>

<ul class="pagination">
  <?php for($i=1;$i<=$number_of_pages;$i++): ?>
  <li <?php if($current_page == $i) { _e(" class='disabled'"); } ?>>
    <?php if ($page_name == 'home'): ?>
      <a href="<?php _e(BASE_URL ."categories/{<?php echo $categ->id}&#125;/{<?php echo $categ->url}&#125;"); ?>"><?php
      _e($i); ?></a>
    <?php else: ?>
      <a href="<?php _e(BASE_URL ."categories/{<?php echo $categ->id}&#125;/{<?php echo $categ->url}&#125;/{<?php echo $i}&#125;"); ?>"><?php
      _e($i); ?></a>
    <?php endif; ?>
  </li>
  <?php endfor; ?>
</ul>

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

```

Cities.php

```

<?php include 'header.php'; ?>
<?php include 'search-form.php'; ?>

<?php include 'flash.php'; ?>
<h3><?php _e($city->name); ?> <?php echo $lang->t('jobs|jobs'); ?> <a href="<?php
_e(BASE_URL ."cities/{<?php echo $city->id}&#125;/{<?php echo $city->url}&#125;/rss"); ?>" target="_blank" class="badge job-
rss">RSS</a></h3>
<div class="list-group">

```

```

<?php foreach($jobs as $job): ?>
  <a class="list-group-item <?php if ($job->is_featured): ?>job-highlight<?php endif; ?>"
  href="<?php _e(BASE_URL ."jobs/{<?php echo $job->id}&#x27;}/". slugify($job->title ." {<?php echo $lang->t('jobs|at')} "). $job->company_name)); ?>">
    <h4>
      <span class="job-title"><?php _e($job->title); ?></span>&nbsp;
      <span class="job-company"><?php _e($job->company_name); ?></span>
      <span class="badge pull-right"><?php niceDate($job->created); ?></span>
    </h4>
  </a>
<?php endforeach; ?>
</div>

<form class="form-inline" role="form" method="post" action="<?php _e(BASE_URL ."subscribe/new"); ?>">
  <div class="form-group">
    <input type="email" class="form-control input-lg" id="email" name="email"
    placeholder="<?php echo $lang->t('jobs|enter_email'); ?>" required />
    <input type="hidden" name="category_id" value="0" />
    <input type="hidden" name="city_id" value="<?php echo $city->id; ?>" />
    <input type="hidden" name="trap" value="" />
    <input type="hidden" name="<?php _e($csrf_key); ?>" value="<?php _e($csrf_token); ?>">
  </div>
  <button type="submit" class="btn btn-default btn-lg"><?php echo $lang->t('jobs|subscribe_to', $city->name); ?></button>
</form>

<ul class="pagination">
  <?php for($i=1;$i<=$number_of_pages;$i++): ?>
  <li <?php if($current_page == $i) { _e(" class='disabled'"); } ?>>
    <?php if ($page_name == 'home'): ?>
      <a href="<?php _e(BASE_URL ."cities/{<?php echo $city->id}&#x27;}/{<?php echo $city->url}&#x27;"); ?>"><?php _e($i); ?></a>
    <?php else: ?>
      <a href="<?php _e(BASE_URL ."cities/{<?php echo $city->id}&#x27;}/{<?php echo $city->url}&#x27;/{<?php echo $i}&#x27;"); ?>"><?php _e($i); ?></a>
    <?php endif; ?>
  </li>
<?php endfor; ?>
</ul>

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

```

Apply.new.php

```

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

<div class="row">
  <div class="col-md-12">
    <?php include 'flash.php'; ?>
    <form id="apply" class="form-horizontal" role="form" action="<?php echo BASE_URL; ?>apply/submit" method="post" enctype="multipart/form-data">

      <h3 class="text-center"><?php echo $lang->t('apply|header'); ?></h3>

```

```

    <div class="form-group">
      <label for="title" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('apply|job_title'); ?></label>
      <div class="col-sm-8">
        <input type="text" class="form-control input-lg" id="job_title" name="job_title"
value="<?php _e($job_title); ?>" disabled />
      </div>
    </div>
    <hr />
    <div class="form-group">
      <label for="cover_letter" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('apply|cover_letter'); ?></label>
      <div class="col-sm-8">
        <textarea class="col-sm-6 form-control input-lg" id="cover_letter"
name="cover_letter" rows="10" required ></textarea>
      </div>
    </div>

    <div class="form-group">
      <label for="company_name" class="col-sm-3 control-label input-lg"><?php echo
$lang->t('apply|full_name'); ?></label>
      <div class="col-sm-8">
        <input type="text" class="form-control input-lg" id="full_name" name="full_name"
required />
      </div>
    </div>

    <div class="form-group">
      <label for="email" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('apply|email'); ?></label>
      <div class="col-sm-8">
        <input type="email" class="form-control input-lg" id="email" name="email"
required />
      </div>
    </div>

    <div class="form-group">
      <label for="location" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('apply|location'); ?></label>
      <div class="col-sm-8">
        <input type="text" class="form-control input-lg" id="location" name="location" />
      </div>
    </div>

    <div class="form-group">
      <label for="websites" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('apply|websites'); ?></label>
      <div class="col-sm-8">
        <textarea id="websites" name="websites" class="form-control input-lg" rows="2"
required ></textarea>
      </div>
    </div>

```

```

    <div class="form-group">
      <label for="attachment" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('apply|attachment'); ?></label>
      <div class="col-sm-8">
        <input type="file" class="filestyle" data-classButton="btn btn-default btn-lg"
id="attachment" name="attachment" data-buttonText="<?php echo $lang->t('jobs|btn_file');
?>" />
      </div>
    </div>

    <div class="form-group">
      <div class="text-center">
        <input type="hidden" id="job_id" name="job_id" value="<?php _e($job_id); ?>" />
        <input type="hidden" id="token" name="token" value="<?php _e($token); ?>" />
        <input type="hidden" id="trap" name="trap" value="" />
        <input type="hidden" name="<?php _e($csrf_key); ?>" value="<?php
_e($csrf_token); ?>" />
        <input type="submit" class="btn btn-success btn-lg" value="<?php echo $lang-
>t('apply|btn_submit'); ?>" />
      </div>
    </div>
  </form>
</div>
</div>

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

```

Job.new.php

```

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

<div class="row">

  <div class="col-md-12">
    <?php include 'flash.php'; ?>
    <form class="form-horizontal" role="form" action="<?php _e(BASE_URL);
?>jobs/review" method="post" enctype="multipart/form-data">
      <div class="form-group">
        <h3 class="text-center">
          <span class="label label-info"><?php echo $lang->t('jobs|step1'); ?></span>
          <span class="label label-default"><?php echo $lang->t('jobs|step2'); ?></span>
          <span class="label label-default"><?php echo $lang->t('jobs|step3'); ?></span>
        </h3>
      </div>
      <hr />
      <div class="form-group">
        <div class="col-sm-3"></div>
        <div class="col-sm-8">
          <h2><?php echo $lang->t('jobs|create_ad'); ?></h2>
          <p class="lead"><?php echo $lang->t('jobs|describe_position'); ?></p>
        </div>
      </div>
    </form>
  </div>

```

```

<div class="form-group">
  <label for="title" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|title'); ?></label>
  <div class="col-sm-8">
    <input type="text" class="form-control input-lg" id="title" name="title" required />
  </div>
</div>

<div class="form-group">
  <label for="category" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|category'); ?></label>
  <div class="col-sm-6">
    <select name="category" id="category" class="form-control input-lg">
      <?php foreach($categories as $category): ?>
        <option value="<?php _e($category->id); ?>"><?php _e($category->name);
?></option>
      <?php endforeach; ?>
    </select>
  </div>
</div>

<div class="form-group">
  <label for="city" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|city'); ?></label>
  <div class="col-sm-6">
    <select name="city" id="city" class="form-control input-lg">
      <?php foreach($cities as $city): ?>
        <option value="<?php _e($city->id); ?>"><?php _e($city->name); ?></option>
      <?php endforeach; ?>
    </select>
  </div>
</div>

<div class="form-group">
  <label for="description" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|description'); ?></label>
  <div class="col-sm-8">
    <textarea id="description" data-provide="markdown" name="description"
rows="15" required ></textarea>
    <p class="help-block"><?php echo $lang->t('jobs|accepts'); ?> <a
href="http://daringfireball.net/projects/markdown/syntax"
target="_blank">Markdown</a></p>
  </div>
</div>

<div class="form-group">
  <label for="perks" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|perks'); ?></label>
  <div class="col-sm-8">
    <textarea id="perks" name="perks" class="form-control input-lg"
rows="2"></textarea>
  </div>
</div>
<div class="form-group">
  <label for="how_to_apply" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|how_to_apply'); ?></label>

```

```

        <div class="col-sm-8">
            <textarea id="how_to_apply" name="how_to_apply" class="form-control input-lg"
rows="2"></textarea>
        </div>
    </div>

```

```

<div class="form-group">
    <div class="col-sm-3"></div>
    <div class="col-sm-8">
        <h2><?php echo $lang->t('jobs|company'); ?></h2>
        <p class="lead"><?php echo $lang->t('jobs|company_info'); ?></p>
    </div>
</div>

```

```

<div class="form-group">
    <label for="company_name" class="col-sm-3 control-label input-lg"><?php echo
$lang->t('jobs|company_name'); ?></label>
    <div class="col-sm-8">
        <input type="text" class="form-control input-lg" id="company_name"
name="company_name" required />
    </div>
</div>

```

```

<div class="form-group">
    <label for="logo" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|logo'); ?></label>
    <div class="col-sm-8">
        <input type="file" class="filestyle" data-classButton="btn btn-default btn-lg"
id="logo" name="logo" accept="image/*" data-buttonText="<?php echo $lang-
>t('jobs|btn_logo'); ?>" />
    </div>
</div>

```

```

<div class="form-group">
    <label for="url" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|website'); ?></label>
    <div class="col-sm-8">
        <input type="url" class="form-control input-lg" id="url" name="url" />
    </div>
</div>
<div class="form-group">
    <label for="email" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|email'); ?></label>
    <div class="col-sm-8">
        <input type="email" class="form-control input-lg" id="email" name="email"
required />
    </div>
</div>

```

```

<div class="form-group">
    <div class="text-center">
        <input type="hidden" id="token" name="token" value="<?php _e($token); ?>" />
        <input type="hidden" id="trap" name="trap" value="" />
        <input type="hidden" name="<?php _e($csrf_key); ?>" value="<?php
_e($csrf_token); ?>" />
    </div>

```

```

        <input type="submit" class="btn btn-success btn-lg" value="<?php echo $lang-
>t('jobs|btn_review'); ?>" />
    </div>
</div>
</form>

</div>

</div>

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

```

Job.review.php

```

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

<div class="row">
    <div class="col-md-12">

        <form class="form-horizontal" role="form" action="<?php _e(BASE_URL . "jobs/{$job-
>id}/publish/{$job->token}"); ?>" method="post" enctype="multipart/form-data">
            <div class="form-group">
                <h3 class="text-center">
                    <span class="label label-default"><?php echo $lang->t('jobs|step1'); ?></span>
                    <span class="label label-info"><?php echo $lang->t('jobs|step2'); ?></span>
                    <span class="label label-default"><?php echo $lang->t('jobs|step3'); ?></span>
                </h3>
            </div>
            <hr />
            <div class="form-group">
                <div class="col-sm-3"></div>
                <div class="col-sm-8">
                    <h2><?php echo $lang->t('jobs|review_ad'); ?></h2>
                    <p class="lead"><?php echo $lang->t('jobs|describe_position'); ?></p>
                </div>
            </div>

            <div class="form-group">
                <label for="title" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|title'); ?></label>
                <div class="col-sm-8">
                    <input type="text" class="form-control input-lg" id="title" name="title"
value="<?php _e($job->title); ?>" required />
                </div>
            </div>

            <div class="form-group">
                <label for="category" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|category'); ?></label>
                <div class="col-sm-6">
                    <select name="category" id="category" class="form-control input-lg">
                        <?php foreach($categories as $category): ?>

```

```

        <option value="<?php _e($category->id); ?>" <?php if($category->id == $job-
>category) { _e("selected"); } ?>><?php _e($category->name); ?></option>
        <?php endforeach; ?>
    </select>
</div>
</div>

<div class="form-group">
    <label for="city" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|city'); ?></label>
    <div class="col-sm-6">
        <select name="city" id="city" class="form-control input-lg">
            <?php foreach($cities as $city): ?>
                <option value="<?php _e($city->id); ?>" <?php if($city->id == $job->city) {
_e("selected"); } ?>><?php _e($city->name); ?></option>
            <?php endforeach; ?>
        </select>
    </div>
</div>

<div class="form-group">
    <label for="description" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|description'); ?></label>
    <div class="col-sm-8">
        <textarea id="description" data-provide="markdown" name="description"
rows="15" required ><?php _e($job->description, 'input'); ?></textarea>
    </div>
</div>

<div class="form-group">
    <label for="perks" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|perks'); ?></label>
    <div class="col-sm-8">
        <textarea id="perks" name="perks" class="form-control input-lg" rows="2"><?php
_e($job->perks, 'input'); ?></textarea>
    </div>
</div>

<div class="form-group">
    <label for="how_to_apply" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|how_to_apply'); ?></label>
    <div class="col-sm-8">
        <textarea id="how_to_apply" name="how_to_apply" class="form-control input-lg"
rows="2"><?php _e($job->how_to_apply, 'input'); ?></textarea>
    </div>
</div>

<div class="form-group">
    <div class="col-sm-3"></div>
    <div class="col-sm-8">
        <h2><?php echo $lang->t('jobs|company'); ?></h2>
        <p class="lead"><?php echo $lang->t('jobs|company_info'); ?></p>
    </div>
</div>

<div class="form-group">

```



```

    <label for="company_name" class="col-sm-3 control-label input-lg"><?php echo
$lang->t('jobs|company_name'); ?></label>
    <div class="col-sm-8">
        <input type="text" class="form-control input-lg" id="company_name"
name="company_name" value="<?php _e($job->company_name); ?>" required />
    </div>
</div>

<?php if ($job->logo != ""): ?>
<div class="form-group">
<label for="current_logo" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|logo'); ?></label>
    <div class="col-sm-8">
        logo}"; ?>" alt=""
class="img-thumbnail">
    </div>
</div>
<?php endif; ?>

<div class="form-group">
<label for="logo" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|logo'); ?></label>
    <div class="col-sm-8">
        <input type="file" class="filestyle" data-classButton="btn btn-default btn-lg"
id="logo" name="logo" accept="image/*" data-buttonText="<?php echo $lang-
>t('jobs|btn_logo'); ?>" />
    </div>
</div>

<div class="form-group">
<label for="url" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|website'); ?></label>
    <div class="col-sm-8">
        <input type="url" class="form-control input-lg" id="url" name="url" value="<?php
_e($job->url); ?>" />
    </div>
</div>
<div class="form-group">
<label for="email" class="col-sm-3 control-label input-lg"><?php echo $lang-
>t('jobs|email'); ?></label>
    <div class="col-sm-8">
        <input type="text" class="form-control input-lg" id="email_muted"
name="email_muted" value="<?php _e($job->email); ?>" disabled />
    </div>
</div>

<div class="form-group">
<div class="text-center">
    <input type="hidden" name="id" value="<?php _e($job->id); ?>" />
    <input type="hidden" name="token" value="<?php _e($job->token); ?>" />
    <input type="hidden" name="trap" value="" />
    <input type="hidden" name="<?php _e($csrf_key); ?>" value="<?php
_e($csrf_token); ?>" />
    <input type="submit" class="btn btn-success btn-lg" value="<?php echo $lang-
>t('jobs|btn_submit'); ?>" />
</div>
</div>

```

```

    </div>
  </form>

</div>

</div>

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

```

Job.publish.php

```

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

<div class="row">
  <div class="col-md-12">
    <div class="form-group">
      <h3 class="text-center">
        <span class="label label-default"><?php echo $lang->t('jobs|step1'); ?></span>
        <span class="label label-default"><?php echo $lang->t('jobs|step2'); ?></span>
        <span class="label label-info"><?php echo $lang->t('jobs|step3'); ?></span>
      </h3>
      <hr />
    </div>
  </div>
  <div class="col-md-2"></div>
</div>
<?php include 'flash.php'; ?>
<div class="row">
  <div class="col-md-9">
    <h2><?php _e($job->title); ?></h2>
    <small class="muted"><?php echo $lang->t('jobs|posted'); ?> <?php niceDate($job->created); ?></small>
    <h4><?php _e($job->company_name); ?></h4>
    <h4><?php _e($city); ?> (<?php _e($category); ?>)</h4>
    <h4><a href="<?php _e($job->url); ?>"><?php _e($job->url); ?></a></h4>
  </div>
  <div class="col-md-3">
    <?php if ($job->logo != "") ?>
    " alt=""
class="img-thumbnail">
    <?php endif; ?>
  </div>
</div>
<div class="row">
  <div class="col-md-12">
    <hr />
  </div>
</div>
<div class="row">
  <div class="col-md-9">
    <div class="lead">
      <?php echo Parsedown::instance()->parse($job->description); ?>
    </div>
  </div>
</div>

```

```

    <?php if ($job->perks != ""): ?>
        <h2><?php echo $lang->t('jobs|perks'); ?></h2>
        <p class="lead">
            <?php _e($job->perks); ?>
        </p>
    <?php endif; ?>
</div>
<div class="col-md-3"></div>
</div>
<?php if ($job->how_to_apply != ""): ?>
<div class="well">
    <h2><?php echo $lang->t('jobs|how_to_apply'); ?></h2>
    <p class="lead"><?php _e($job->how_to_apply); ?></p>
</div>
<?php endif; ?>
<?php include 'footer.php'; ?>

```

Job.show.php

```

<?php include 'header.php'; ?>
<div class="row">
    <div class="col-md-9">
        <?php include 'flash.php'; ?>
        <h2><?php _e($job->title); ?></h2>
        <small class="muted"><?php echo $lang->t('jobs|posted'); ?> <?php niceDate($job->created); ?></small>
        <h4><?php _e($job->company_name); ?></h4>
        <h4><?php _e($city); ?> (<?php _e($category); ?>)</h4>
        <h4><a href="<?php _e($job->url); ?>" target="_blank"><?php _e(excerpt($job->url, 50)); ?></a></h4>
    </div>
    <div class="col-md-3">
        <?php if ($job->logo != ""): ?>
            logo); ?>" alt="" class="img-thumbnail">
                <?php endif; ?>
        </div>
    </div>
</div>
<div class="row">
<div class="col-md-12">
    <hr />
</div>
</div>
<div class="row">
    <div class="col-md-9">
        <div class="lead">
            <?php echo Parsedown::instance()->parse($job->description); ?>
        </div>
        <?php if ($job->perks != ""): ?>
            <h2><?php echo $lang->t('jobs|perks'); ?></h2>
            <p class="lead">
                <?php _e($job->perks, 'r'); ?>
            </p>
        </div>
    </div>
</div>

```

```

    <?php endif; ?>
</div>
<div class="col-md-3">
    <div class="list-group">
        <a class="list-group-item text-center">
            <?php Blocks::showBlockByID(1); ?>
        </a>
        <?php if ($job->how_to_apply == ""): ?>
        <a class="list-group-item" />
            <h4 class="list-group-item-heading"><span class="glyphicon glyphicon-
heart"></span> <?php _e($applications); ?> <?php echo $lang->t('apply|applications');
?></h4>
            </a>
            <a href="<?php _e(BASE_URL . "apply/{$job->id}"); ?>" class="list-group-item" />
                <h4 class="list-group-item-heading"><span class="glyphicon glyphicon-
user"></span> <?php echo $lang->t('apply|apply_now'); ?></h4>
            </a>
        <?php endif; ?>
    </div>
</div>
<?php if ($job->how_to_apply != ""): ?>
<div class="well">
    <h2><?php echo $lang->t('jobs|how_to_apply'); ?></h2>
    <p class="lead"><?php _e($job->how_to_apply, 'r'); ?></p>
</div>
<?php endif; ?>
<?php include 'footer.php'; ?>

```

Page.php

```

<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
    <title>Google Maps</title>
    <script
src="https://maps.googleapis.com/maps/api/js?key=AlzaSyCCB5rlzgCswG1npCWGb0Wg
6rIkIW9Der0"
    type="text/javascript"></script>
    <script type="text/javascript">

    var customIcons = {
        Programming: {
            icon: 'http://labs.google.com/ridefinder/images/mm_20_blue.png'
        },
        Design: {
            icon: 'http://labs.google.com/ridefinder/images/mm_20_red.png'
        },
        Business: {
            icon: 'http://labs.google.com/ridefinder/images/mm_20_green.png'
        },
    },

```

```

Admin: {
  icon: 'http://labs.google.com/ridefinder/images/mm_20_yellow.png'
},
Support: {
  icon: 'http://labs.google.com/ridefinder/images/mm_20_white.png'
},
Content: {
  icon: 'http://labs.google.com/ridefinder/images/mm_20_black.png'
}
};

function load() {
  var map = new google.maps.Map(document.getElementById("map"), {
    center: new google.maps.LatLng(39.37534, 21.67789),
    zoom: 6,
    mapTypeId: 'roadmap'
  });

  var infoWindow = new google.maps.InfoWindow;

  downloadUrl("map/phpsqlajax_genxml.php", function(data) {
    var xml = data.responseXML;
    var markers = xml.documentElement.getElementsByTagName("marker");
    for (var i = 0; i < markers.length; i++) {
      var title = markers[i].getAttribute("title");
      var company_name = markers[i].getAttribute("company_name");
      var url = markers[i].getAttribute("url");
      var email = markers[i].getAttribute("email");
      var city_name = markers[i].getAttribute("city_name");
      var categories_name = markers[i].getAttribute("categories_name");
      var point = new google.maps.LatLng(
        parseFloat(markers[i].getAttribute("lat")),
        parseFloat(markers[i].getAttribute("lng")));
      var html = "<b>" + title + "</b> <br/>" + company_name + "</b> <br/>" + "<a href="
+ url + "</a>" + "</b> <br/>" + email + "</b> <br/>" + city_name + "</b> <br/>" +
categories_name;
      var icon = customIcons[categories_name] || {};
      var marker = new google.maps.Marker({
        map: map,
        position: point,
        icon: icon.icon
      });
      bindInfoWindow(marker, map, infoWindow, html);
    }
  });
}

function bindInfoWindow(marker, map, infoWindow, html) {
  google.maps.event.addListener(marker, 'click', function() {
    infoWindow.setContent(html);
    infoWindow.open(map, marker);
  });
}

function downloadUrl(url, callback) {
  var request = window.ActiveXObject ?

```

```

    new XMLHttpRequest('Microsoft.XMLHTTP') :
    new XMLHttpRequest;

    request.onreadystatechange = function() {
        if (request.readyState === 4) {
            request.onreadystatechange = doNothing;
            callback(request, request.status);
        }
    };

    request.open('GET', url, true);
    request.send(null);
}

function doNothing() {}

//]]>

</script>
</head>
<body onload="load()">
  <?php include 'header.php'; ?>
  <div class="row">
    <div class="col-md-2"></div>
    <div class="col-md-8">
      <h2><?php _e($content->name); ?></h2>
      <p class="lead"><?php echo Parsedown::instance()->parse($content->content);
?></p>
    </div>
  </div><br/>
  <div id="map" style="width: 1000px; height: 600px"></div><br/>
  <form action="map/myMap.php" method="post">
    <label id="first">MY CITY IS:</label><br/>
    <input type="text" name="city" required=""><br/>
    <label id="first">DISTANCE (KM):</label><br/>
    <input type="text" name="distance" required=""><br/>
    <button type="submit" name="save">SHOW NEAREST JOBS</button>
  </form>
  <?php include 'footer.php'; ?>
</body>
</html>

```

VIEWS - ADMIN

Home.php

```

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

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

<div class="row">
  <div class="col-md-6">

```

```

    <a class="twitter-timeline" href="..." data-widget-id="..."></a>
</div>
<div class="col-md-6">
    <a class="twitter-timeline" href="..." data-widget-id="..."></a>
</div>
<script>!function(d,s,id){var
js,fjs=d.getElementsByTagName(s)[0],p=/^http:/.test(d.location)?'http':'https';if(!d.getElementById(
ld(id)){fjs=d.createElement(s);js.id=id;js.src=p+"://platform.twitter.com/widgets.js";fjs.parentNode
.insertBefore(fjs,fjs);}}(document,"script","twitter-wjs");</script>
</div>

<h3><?php echo $lang->t('admin|inactive'); ?> <?php echo $lang->t('jobs|jobs'); ?></h3>
<?php foreach($categories as $category): ?>
    <strong><?php _e($category->name); ?> <?php echo $lang->t('jobs|jobs'); ?></strong>
    <div class="table-responsive">
        <table class="table table-striped">
            <thead>
                <tr>
                    <th width="10%"><?php echo $lang->t('admin|tbl_date'); ?></th>
                    <th width="20%"><?php echo $lang->t('admin|tbl_company'); ?></th>
                    <th width="40%"><?php echo $lang->t('admin|tbl_title'); ?></th>
                    <th width="15%"><?php echo $lang->t('admin|tbl_city'); ?></th>
                    <th width="15%"><?php echo $lang->t('admin|tbl_action'); ?></th>
                </tr>
            </thead>
            <tbody>

                <?php foreach($jobs[$category->id] as $job): ?>
                <tr>
                    <td><?php _e(niceDate($job->created)); ?></td>
                    <td><?php _e($job->company_name); ?></td>
                    <td><a href="<?php _e(ADMIN_URL . "jobs/{$job->id}"); ?>"><?php _e($job->title);
?></a></td>
                    <td><?php _e($cities[$job->city]['name']); ?></td>
                    <td>
                        <a type="button" class="btn btn-info btn-xs" title="<?php echo $lang-
>t('admin|btn_edit'); ?>" href="<?php _e(ADMIN_URL . "jobs/{$job->id}/edit/{$job->token}");
?>">
                            <span class="glyphicon glyphicon-pencil"></span>
                        </a>
                        <?php if (!$job->is_featured): ?>
                            <a type="button" class="btn btn-primary btn-xs" title="<?php echo $lang-
>t('admin|btn_feature_on'); ?>" href="<?php _e(ADMIN_URL . "jobs/{$job->id}/feature/on/{$job-
>token}"); ?>">
                                <span class="glyphicon glyphicon-star"></span>
                            </a>
                        <?php else: ?>
                            <a type="button" class="btn btn-default btn-xs" title="<?php echo $lang-
>t('admin|btn_feature_off'); ?>" href="<?php _e(ADMIN_URL . "jobs/{$job->id}/feature/off/{$job-
>token}"); ?>">
                                <span class="glyphicon glyphicon-star"></span>
                            </a>
                        <?php endif; ?>
                            <a type="button" class="btn btn-success btn-xs" title="<?php echo $lang-
>t('admin|btn_approve'); ?>" href="<?php _e(ADMIN_URL . "jobs/{$job->id}/activate/" .
accessToken($job->id)); ?>">

```

```

        <span class="glyphicon glyphicon-ok"></span>
    </a>
    <a type="button" class="btn btn-danger btn-xs" title="<?php echo $lang-
>t('admin|btn_delete'); ?>" href="<?php _e(ADMIN_URL . "jobs/{\$job->id}/delete/{\$job-
>token}"); ?>">
        <span class="glyphicon glyphicon-remove"></span>
    </a>
</td>
</tr>
<?php endforeach; ?>

</tbody>
</table>
</div>
<?php endforeach; ?>

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

```

Categories.php

```

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

<h3><?php _e(\$categ->name); ?> <?php echo $lang->t('jobs|jobs'); ?></h3>

<div class="table-responsive">
    <table class="table table-striped">
        <thead>
            <tr>
                <th width="10%"><?php echo $lang->t('admin|tbl_date'); ?></th>
                <th width="20%"><?php echo $lang->t('admin|tbl_company'); ?></th>
                <th width="40%"><?php echo $lang->t('admin|tbl_title'); ?></th>
                <th width="15%"><?php echo $lang->t('admin|tbl_city'); ?></th>
                <th width="15%"><?php echo $lang->t('admin|tbl_action'); ?></th>
            </tr>
        </thead>
        <tbody>
            <?php foreach(\$jobs as \$job): ?>
                <tr>
                    <td><?php _e(niceDate(\$job->created)); ?></td>
                    <td><?php _e(\$job->company_name); ?></td>
                    <td><a href="<?php _e(ADMIN_URL . "jobs/{\$job->id}"); ?>"><?php _e(\$job->title);
?></a></td>
                    <td><?php _e(\$cities[\$job->city]['name']); ?></td>
                    <td>
                        <a type="button" class="btn btn-info btn-xs" title="<?php echo $lang-
>t('admin|btn_edit'); ?>" href="<?php _e(ADMIN_URL . "jobs/{\$job->id}/edit/{\$job->token}");
?>">
                            <span class="glyphicon glyphicon-pencil"></span>
                        </a>
                        <?php if (!\$job->is_featured): ?>

```



```

        <a type="button" class="btn btn-primary btn-xs" title="<?php echo $lang-
>t('admin|btn_feature_on'); ?>" href="<?php _e(ADMIN_URL . "jobs/{<?php echo $job->id}/feature/on/{<?php echo $job->token}"); ?>">
            <span class="glyphicon glyphicon-star"></span>
        </a>
        <?php else: ?>
        <a type="button" class="btn btn-default btn-xs" title="<?php echo $lang-
>t('admin|btn_feature_off'); ?>" href="<?php _e(ADMIN_URL . "jobs/{<?php echo $job->id}/feature/off/{<?php echo $job->token}"); ?>">
            <span class="glyphicon glyphicon-star"></span>
        </a>
        <?php endif; ?>
        <a type="button" class="btn btn-warning btn-xs" title="<?php echo $lang-
>t('admin|btn_deactivate'); ?>" href="<?php _e(ADMIN_URL . "jobs/{<?php echo $job->id}/deactivate/" .
accessToken(<?php echo $job->id}); ?>">
            <span class="glyphicon glyphicon-minus"></span>
        </a>
        <a type="button" class="btn btn-danger btn-xs" title="<?php echo $lang-
>t('admin|btn_delete'); ?>" href="<?php _e(ADMIN_URL . "jobs/{<?php echo $job->id}/delete/{<?php echo $job->token}"); ?>">
            <span class="glyphicon glyphicon-remove"></span>
        </a>
    </td>
</tr>
<?php endforeach; ?>

</tbody>
</table>
</div>

<ul class="pagination">
    <?php for($i=1;$i<=$number_of_pages;$i++): ?>
    <li <?php if($current_page == $i) { _e(" class='disabled'"); } ?>>
        <?php if ($page_name == 'home'): ?>
            <a href="<?php _e(ADMIN_URL . "categories/{<?php echo $categ->id}/{<?php echo $categ->url}"); ?>"><?php
            _e($i); ?></a>
        <?php else: ?>
            <a href="<?php _e(ADMIN_URL . "categories/{<?php echo $categ->id}/{<?php echo $categ->url}/{<?php echo $i}"); ?>"><?php
            _e($i); ?></a>
        <?php endif; ?>
    </li>
<?php endfor; ?>
</ul>

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

```

Cities.php

```

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

<h3><?php _e($city->name); ?> <?php echo $lang->t('jobs|jobs'); ?></h3>

<div class="table-responsive">

```

```

<table class="table table-striped">
<thead>
<tr>
<th width="10%"><?php echo $lang->t('admin|tbl_date'); ?></th>
<th width="20%"><?php echo $lang->t('admin|tbl_company'); ?></th>
<th width="40%"><?php echo $lang->t('admin|tbl_title'); ?></th>
<th width="15%"><?php echo $lang->t('admin|tbl_category'); ?></th>
<th width="15%"><?php echo $lang->t('admin|tbl_action'); ?></th>
</tr>
</thead>
<tbody>

<?php foreach($jobs as $job): ?>
<tr>
<td><?php _e(niceDate($job->created)); ?></td>
<td><?php _e($job->company_name); ?></td>
<td><a href="<?php _e(ADMIN_URL . "jobs/{$job->id}"); ?>"><?php _e($job->title);
?></a></td>
<td><?php _e($categories[$job->category]['name']); ?></td>
<td>
<a type="button" class="btn btn-info btn-xs" title="<?php echo $lang-
>t('admin|btn_edit'); ?>" href="<?php _e(ADMIN_URL . "jobs/{$job->id}/edit/{$job->token}");
?>">
<span class="glyphicon glyphicon-pencil"></span>
</a>
<?php if (!$job->is_featured): ?>
<a type="button" class="btn btn-primary btn-xs" title="<?php echo $lang-
>t('admin|btn_feature_on'); ?>" href="<?php _e(ADMIN_URL . "jobs/{$job->id}/feature/on/{$job-
>token}"); ?>">
<span class="glyphicon glyphicon-star"></span>
</a>
<?php else: ?>
<a type="button" class="btn btn-default btn-xs" title="<?php echo $lang-
>t('admin|btn_feature_off'); ?>" href="<?php _e(ADMIN_URL . "jobs/{$job->id}/feature/off/{$job-
>token}"); ?>">
<span class="glyphicon glyphicon-star"></span>
</a>
<?php endif; ?>
<a type="button" class="btn btn-warning btn-xs" title="<?php echo $lang-
>t('admin|btn_deactivate'); ?>" href="<?php _e(ADMIN_URL . "jobs/{$job->id}/deactivate/" .
accessToken($job->id)); ?>">
<span class="glyphicon glyphicon-minus"></span>
</a>
<a type="button" class="btn btn-danger btn-xs" title="<?php echo $lang-
>t('admin|btn_delete'); ?>" href="<?php _e(ADMIN_URL . "jobs/{$job->id}/delete/{$job-
>token}"); ?>">
<span class="glyphicon glyphicon-remove"></span>
</a>
</td>
</tr>
<?php endforeach; ?>

</tbody>
</table>
</div>

```

```

<ul class="pagination">
  <?php for($i=1;$i<=$number_of_pages;$i++): ?>
  <li <?php if($current_page == $i) { _e(" class='disabled'"); } ?>>
    <?php if ($page_name == 'home'): ?>
      <a href="<?php _e(ADMIN_URL . "cities/{$city->id}/{ $city->url}"); ?>"><?php _e($i); ?></a>
    <?php else: ?>
      <a href="<?php _e(ADMIN_URL . "cities/{$city->id}/{ $city->url}/{ $i}"); ?>"><?php _e($i);
    ?></a>
    <?php endif; ?>
  </li>
<?php endfor; ?>
</ul>

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

```

Applications.php

```

<?php include 'header.php'; ?>
<h3><?php _e($count); ?> <?php echo $lang->t('admin|job_applications'); ?></h3>
<?php if (isset($title) && $title != " ") : ?>
  <h4><?php echo $lang->t('apply|job_title'); ?>: <?php _e($title); ?></h4>
<?php endif; ?>

<div class="table-responsive">
  <table class="table table-striped">
    <thead>
      <tr>
        <th><?php echo $lang->t('admin|tbl_id'); ?></th>
        <th><?php echo $lang->t('admin|tbl_name'); ?></th>
        <th><?php echo $lang->t('admin|tbl_location'); ?></th>
        <th><?php echo $lang->t('admin|tbl_email'); ?></th>
        <th><?php echo $lang->t('admin|tbl_attachment'); ?></th>
        <th><?php echo $lang->t('admin|tbl_sent'); ?></th>
      </tr>
    </thead>
    <tbody>
      <?php foreach ($applications as $apps): ?>
      <tr>
        <td><a href="<?php _e(ADMIN_URL . "jobs/{ $apps->job_id}"); ?>"><?php _e($apps->
job_id); ?></a></td>
        <td><?php _e($apps->full_name); ?></td>
        <td><?php _e($apps->location); ?></td>
        <td><?php _e($apps->email); ?></td>
        <td>
          <?php if ($apps->attachment != " "): ?>
            <a href="<?php _e(ASSET_URL . "attachments/{ $apps->attachment}");
?>">Download</a>
          <?php endif; ?>
        </td>
        <td><?php _e(niceDate($apps->created)); ?></td>
      </tr>
      <tr>
        <td colspan="6">

```

```

        <p><?php _e($apps->cover_letter,'r'); ?></p>
    </td>
</tr>
<?php endforeach; ?>
</tbody>
</table>
</div>

<ul class="pagination">
<?php for($i=1;$i<=$number_of_pages;$i++): ?>
<li <?php if($current_page == $i) { _e(" class='disabled'"); } ?>>
    <a href="<?php _e(ADMIN_URL ."applications/{$i}"); ?>"><?php _e($i); ?></a>
</li>
<?php endfor; ?>
</ul>

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

```

Applications.job.php

```

<?php include 'header.php'; ?>
<h3><?php _e($count); ?> <?php echo $lang->t('admin|job_applications'); ?></h3>
<?php if (isset($title) && $title != " ") :?>
    <h4><?php echo $lang->t('apply|job_title'); ?>: <a href="<?php _e(ADMIN_URL .
"jobs/{$id}"); ?>"><?php _e($title); ?></a></h4>
<?php endif; ?>

<div class="table-responsive">
    <table class="table table-striped">
        <thead>
            <tr>
                <th><?php echo $lang->t('admin|tbl_name'); ?></th>
                <th><?php echo $lang->t('admin|tbl_location'); ?></th>
                <th><?php echo $lang->t('admin|tbl_email'); ?></th>
                <th><?php echo $lang->t('admin|tbl_attachment'); ?></th>
                <th><?php echo $lang->t('admin|tbl_sent'); ?></th>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($applications as $apps): ?>
                <tr>
                    <td><?php _e($apps->full_name); ?></td>
                    <td><?php _e($apps->location); ?></td>
                    <td><?php _e($apps->email); ?></td>
                    <td>
                        <?php if ($apps->attachment != ""): ?>
                            <a href="<?php _e(ASSET_URL . "attachments/{$apps->attachment}"); ?>"><?php
echo $lang->t('admin|download'); ?></a>
                        <?php endif; ?>
                    </td>
                    <td><?php _e(niceDate($apps->created)); ?></td>
                </tr>
            </tr>
        </tbody>
    </table>

```

```

        <td colspan="6">
            <p><?php _e($apps->cover_letter,'r'); ?></p>
        </td>
    </tr>
</tbody>
</table>
</div>

<ul class="pagination">
    <?php for($i=1;$i<=$number_of_pages;$i++): ?>
    <li <?php if($current_page == $i) { _e(" class='disabled'"); } ?>>
    <a href="<?php _e(ADMIN_URL ."applications/jobs/{$apps->job_id}/{ $i}"); ?>"><?php _e($i);
    ?></a>
    </li>
    <?php endfor; ?>
</ul>

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

```

Categories.edit.php

```

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

<h3><?php echo $lang->t('admin|manage'); ?> <?php echo $lang->t('link|categories'); ?></h3>
<?php include 'flash.php'; ?>

<div class="row">
    <div class="col-md-9">

        <div class="table-responsive">
            <table class="table table-striped">
                <thead>
                    <tr>
                        <th width="5%"><?php echo $lang->t('admin|tbl_id'); ?></th>
                        <th width="20%"><?php echo $lang->t('admin|tbl_name'); ?></th>
                        <th width="40%"><?php echo $lang->t('admin|tbl_meta_desc'); ?></th>
                        <th width="15%"><?php echo $lang->t('admin|tbl_url'); ?></th>
                        <th width="10%"><?php echo $lang->t('admin|tbl_sort'); ?></th>
                        <th width="10%"><?php echo $lang->t('admin|tbl_action'); ?></th>
                    </tr>
                </thead>
                <tbody>

                    <?php foreach($categs as $cat): ?>
                    <tr>
                        <td><?php _e($cat->id); ?></td>
                        <td><?php _e($cat->name); ?></td>
                        <td><?php _e($cat->description); ?></td>
                        <td><?php _e($cat->url); ?></td>
                        <td><?php _e($cat->sort); ?></td>
                        <td>

```

```

        <a type="button" class="btn btn-info btn-xs" title="<?php echo $lang-
>t('admin|btn_edit'); ?>" href="<?php _e(ADMIN_MANAGE . "/categories/{$cat->id}/edit"); ?>">
        <span class="glyphicon glyphicon-pencil"></span>
    </a>
    <a type="button" class="btn btn-danger btn-xs" title="<?php echo $lang-
>t('admin|btn_delete'); ?>" href="<?php _e(ADMIN_MANAGE . "/categories/{$cat->id}/delete");
?>">
        <span class="glyphicon glyphicon-remove"></span>
    </a>
</td>
</tr>
<?php endforeach; ?>

</tbody>
</table>
</div>

</div>
<div class="col-md-3">
    <h4><?php if ($category && $category->name) { echo $lang->t('admin|btn_edit') . "<span
class='pull-right'><a href='". ADMIN_MANAGE . '/categories' . "'>". $lang-
>t('admin|btn_add_new') . "</a></span>"; } else { echo $lang->t('admin|btn_add_new'); }
?></span></h4>
    <form class="form" role="form" method="post" action="<?php _e(ADMIN_MANAGE .
'/categories'); ?>">
        <div class="form-group">
            <input type="text" class="form-control" name="name" placeholder="<?php echo $lang-
>t('admin|tbl_name'); ?>" value="<?php if ($category && $category->name) { _e($category-
>name); } ?>" required />
        </div>
        <div class="form-group">
            <input type="text" class="form-control" name="description" placeholder="<?php echo
$lang->t('admin|tbl_meta_desc'); ?>" maxlength="150" value="<?php if ($category &&
$category->description) { _e($category->description); } ?>" required />
        </div>
        <div class="form-group">
            <input type="text" class="form-control" name="url" placeholder="<?php echo $lang-
>t('admin|slug_url'); ?>" value="<?php if ($category && $category->url) { _e($category->url); }
?>" required />
        </div>
        <div class="form-group">
            <input type="text" class="form-control" name="sort" placeholder="<?php echo $lang-
>t('admin|tbl_sort'); ?>" value="<?php if ($category && $category->sort) { _e($category->sort);
} ?>" required />
        </div>
        <input type="hidden" name="id" value="<?php if ($category && $category->id) {
_e($category->id); } ?>">
        <input type="hidden" name="<?php _e($csrf_key); ?>" value="<?php _e($csrf_token);
?>">
        <button type="submit" class="btn btn-info"><?php echo $lang->t('admin|btn_submit');
?></button>
    </form>
</div>
</div>

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

```

Cities.edit.php

```

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

<h3><?php echo $lang->t('admin|manage'); ?> <?php echo $lang->t('link|cities'); ?></h3>
<?php include 'flash.php'; ?>

<div class="row">
  <div class="col-md-9">

    <div class="table-responsive">
      <table class="table table-striped">
        <thead>
          <tr>
            <th width="5%"><?php echo $lang->t('admin|tbl_id'); ?></th>
            <th width="20%"><?php echo $lang->t('admin|tbl_name'); ?></th>
            <th width="40%"><?php echo $lang->t('admin|tbl_meta_desc'); ?></th>
            <th width="15%"><?php echo $lang->t('admin|tbl_url'); ?></th>
            <th width="10%"><?php echo $lang->t('admin|tbl_sort'); ?></th>
            <th width="10%"><?php echo $lang->t('admin|tbl_action'); ?></th>
          </tr>
        </thead>
        <tbody>

          <?php foreach($cits as $cit): ?>
            <tr>
              <td><?php _e($cit->id); ?></td>
              <td><?php _e($cit->name); ?></td>
              <td><?php _e($cit->description); ?></td>
              <td><?php _e($cit->url); ?></td>
              <td><?php _e($cit->sort); ?></td>
              <td>
                <a type="button" class="btn btn-info btn-xs" title="<?php echo $lang->t('admin|btn_edit'); ?>" href="<?php _e(ADMIN_MANAGE . "/cities/{\$cit->id}/edit"); ?>">
                  <span class="glyphicon glyphicon-pencil"></span>
                </a>
                <a type="button" class="btn btn-danger btn-xs" title="<?php echo $lang->t('admin|btn_delete'); ?>" href="<?php _e(ADMIN_MANAGE . "/cities/{\$cit->id}/delete"); ?>">
                  <span class="glyphicon glyphicon-remove"></span>
                </a>
              </td>
            </tr>
          <?php endforeach; ?>

        </tbody>
      </table>
    </div>

    <div class="col-md-3">
      <h4><?php if ($city && $city->name) { echo $lang->t('admin|btn_edit') . " <span class='pull-right'><a href='". ADMIN_MANAGE . "/cities' .">. $lang->t('admin|btn_add_new') . "</a></span>"; } else { echo $lang->t('admin|btn_add_new'); } ?></h4>

```

```

<form class="form" role="form" method="post" action="<?php _e(ADMIN_MANAGE .
/cities); ?>">
  <div class="form-group">
    <input type="text" class="form-control" name="name" placeholder="<?php echo $lang-
>t('admin|tbl_name'); ?>" value="<?php if ($city && $city->name) { _e($city->name); } ?>"
required />
  </div>
  <div class="form-group">
    <input type="text" class="form-control" name="description" placeholder="<?php echo
$lang->t('admin|tbl_meta_desc'); ?>" maxlength="150" value="<?php if ($city && $city-
>description) { _e($city->description); } ?>" required />
  </div>
  <div class="form-group">
    <input type="text" class="form-control" name="url" placeholder="<?php echo $lang-
>t('admin|slug_url'); ?>" value="<?php if ($city && $city->url) { _e($city->url); } ?>" required />
  </div>
  <div class="form-group">
    <input type="text" class="form-control" name="sort" placeholder="<?php echo $lang-
>t('admin|tbl_sort'); ?>" value="<?php if ($city && $city->sort) { _e($city->sort); } ?>" required
/>
  </div>
  <input type="hidden" name="id" value="<?php if ($city && $city->id) { _e($city->id); } ?>">
  <input type="hidden" name="<?php _e($csrf_key); ?>" value="<?php _e($csrf_token);
?>">
  <button type="submit" class="btn btn-info"><?php echo $lang->t('admin|btn_submit');
?></button>
</form>
</div>
</div>
<?php include 'footer.php'; ?>

```

Subscribers.php

```

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

<h3><?php _e($count); ?> <?php echo $lang->t('admin|user_subscriptions'); ?></h3>
<?php include 'flash.php'; ?>

<div class="table-responsive">
  <table class="table table-striped">
    <thead>
      <tr>
        <th><?php echo $lang->t('admin|tbl_id'); ?></th>
        <th><?php echo $lang->t('admin|tbl_email'); ?></th>
        <th><?php echo $lang->t('admin|tbl_category'); ?></th>
        <th><?php echo $lang->t('admin|tbl_city'); ?></th>
        <th><?php echo $lang->t('admin|tbl_last_sent'); ?></th>
        <th><?php echo $lang->t('admin|tbl_action'); ?></th>
      </tr>
    </thead>
    <tbody>
      <?php foreach ($users as $user): ?>
        <tr>

```



```

<td><?php _e($user->id); ?></td>
<td><a href="mailto:<?php _e($user->email); ?>"><?php _e($user->email); ?></a></td>
<td><?php if ($user->category_id > 0) { _e($categories[$user->category_id]->name); }
?></td>
<td><?php if ($user->city_id > 0) { _e($cities[$user->city_id]->name); } ?></td>
<td><?php _e(niceDate($user->last_sent)); ?></td>
<td>
  <?php if (!$user->is_confirmed): ?>
    <a type="button" class="btn btn-success btn-xs" title="<?php echo $lang-
>t('admin|btn_approve'); ?>" href="<?php _e(ADMIN_URL . "subscribers/{$user-
>id}/approve/{$user->token}"); ?>">
      <span class="glyphicon glyphicon-ok"></span>
    </a>
  <?php else: ?>
    <a type="button" class="btn btn-warning btn-xs" title="<?php echo $lang-
>t('admin|btn_deactivate'); ?>" href="<?php _e(ADMIN_URL . "subscribers/{$user-
>id}/deactivate/{$user->token}"); ?>">
      <span class="glyphicon glyphicon-minus"></span>
    </a>
  <?php endif; ?>
  <a type="button" class="btn btn-danger btn-xs" title="<?php echo $lang-
>t('admin|btn_delete'); ?>" href="<?php _e(ADMIN_URL . "subscribers/{$user-
>id}/delete/{$user->token}"); ?>">
    <span class="glyphicon glyphicon-remove"></span>
  </a>
</td>
</tr>

<?php endforeach; ?>
</tbody>
</table>
</div>

<ul class="pagination">
  <?php for($i=1;$i<=$number_of_pages;$i++): ?>
  <li <?php if($current_page == $i) { _e(" class='disabled'"); } ?>>
    <a href="<?php _e(ADMIN_URL . "subscribers/{$i}"); ?>"><?php _e($i); ?></a>
  </li>
  <?php endfor; ?>
</ul>

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

```

MAP

Geocode.php

```

<?php
require("phpsqlajax_dbinfo.php");

define("MAPS_HOST", "maps.googleapis.com");
define("KEY", "AlzaSyCCB5rlzgCswG1npCWGb0Wg6rIkIW9Der0");

```

```

// Opens a connection to a MySQL server
$connection = mysql_connect("localhost", $username, $password);
if (!$connection) {
    die("Not connected : " . mysql_error());
}

// Set the active MySQL database
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {

    die("Can't use db : " . mysql_error());
}

$query = "SELECT id, name FROM cities order by id asc limit 0,1000";
$result = mysql_query($query);
if (!$result) {
    die("Invalid query: " . mysql_error());
}

//Initialize delay in geocode speed

$delay=0;
$base_url = "http://" . MAPS_HOST . "/maps/api/geocode/json?address=";

while ($row = @mysql_fetch_assoc($result)) {
    $geocode_pending = true;

    while ($geocode_pending){

        $address = $row["name"];
        $id = $row["id"];

        $fullAddress = $address . ", GREECE";

        $request_url = $base_url . "" . urlencode($fullAddress) . "&sensor=false";

        sleep(2);

        $json = file_get_contents($request_url);
        $json_decoded = json_decode($json);

        $status = $json_decoded->status;

        if (strcmp($json_decoded->status, "OK") == 0) {

            $geocode_pending = false;

            $lat = $json_decoded->results[0]->geometry->location->lat;
            $lng = $json_decoded->results[0]->geometry->location->lng;

            echo 'here';

            $query = sprintf("UPDATE cities
                SET lat = '%s', lng = '%s'
                WHERE id = '%s' LIMIT 1;");

```

```

        mysql_real_escape_string($lat),
        mysql_real_escape_string($lng),
        mysql_real_escape_string($id));
$update_result = mysql_query($query);

echo $id;

if (!$update_result) {
    die("Invalid query: " . mysql_error());
}
}

else {
// failure to geocode
$geocode_pending = false;
echo "Address " . $fullAddress . " failed to geocode. ";
echo "Received status " . $status . "\n";
}

// usleep($delay);
}
}
?>

```

Phpsqlajax_genxml.php

```

<?php
require("phpsqlajax_dbinfo.php");

function parseToXML($htmlStr)
{
$xmlStr=str_replace('<','&lt;',$htmlStr);
$xmlStr=str_replace('>','&gt;',$xmlStr);
$xmlStr=str_replace('"','&quot;',$xmlStr);
$xmlStr=str_replace("'",'&#39;',$xmlStr);
$xmlStr=str_replace("&","&amp;",$xmlStr);
return $xmlStr;
}

// Opens a connection to a MySQL server
$connection=mysql_connect ('localhost', $username, $password);
if (!$connection) {
    die('Not connected : ' . mysql_error());
}

// Set the active MySQL database
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
    die ("Can't use db : " . mysql_error());
}

```

```

// delete previous data from table markers

$preresult = mysql_query("DELETE FROM markers WHERE 1;");

if (!$preresult) {
    die('Invalid query: ' . mysql_error());
}

// Inner join between table jobs, cities, categories - insert mysql query result into table markers

$result = mysql_query("INSERT INTO markers
    SELECT jobs.title, jobs.company_name, jobs.url,jobs.email,cities.name,
    cities.lat, cities.lng, categories.name
    FROM jobs
    INNER JOIN cities
    ON jobs.city=cities.id
    INNER JOIN categories
    ON jobs.category=categories.id
    WHERE jobs.status=1;");

if (!$result) {
    die('Invalid query: ' . mysql_error());
}

// Select all the rows in the markers table

$result = mysql_query("SELECT * FROM markers WHERE 1");
if (!$result) {
    die('Invalid query: ' . mysql_error());
}

header("Content-type: text/xml");

// Start XML file, echo parent node
echo '<markers>';

// Iterate through the rows, printing XML nodes for each
while ($row = mysql_fetch_assoc($result)){
    // ADD TO XML DOCUMENT NODE
    echo '<marker ';
    echo 'title="' . parseToXML($row['title']) . "' ";
    echo 'company_name="' . parseToXML($row['company_name']) . "' ";
    echo 'url="' . parseToXML($row['url']) . "' ";
    echo 'email="' . parseToXML($row['email']) . "' ";
    echo 'city_name="' . parseToXML($row['city_name']) . "' ";
    echo 'lat="' . $row['lat'] . "' ";
    echo 'lng="' . $row['lng'] . "' ";
    echo 'categories_name="' . $row['categories_name'] . "' ";
    echo '/>';
}
// End XML file
echo '</markers>';

?>

```

InputHandler.php

```

<?php
require("phpsqlajax_dbinfo.php");

define("MAPS_HOST", "maps.googleapis.com");
define("KEY", "AIzaSyCCB5rlzgCswG1npCWGb0Wg6rIkIW9Der0");

// Opens a connection to a MySQL server
$connection=mysql_connect ('localhost', $username, $password);
if (!$connection) {
    die('Not connected : ' . mysql_error());
}

// Set the active MySQL database
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
    die ('Can\'t use db : ' . mysql_error());
}

$city = $_POST['city'];
$distance = $_POST['distance'];

if (isset($_POST['save'])) {

$query = mysql_query("UPDATE input
                    SET city = '$city', input_distance = '$distance'
                    LIMIT 1");
//echo "Form Submitted succesfully";

if (!$query) {
    die('Invalid query: ' . mysql_error());
}

}

$initial_city = mysql_query (
    "SELECT id, city
    FROM input
    ORDER BY city
    DESC LIMIT 1"
    );

if (!$initial_city) {
    die('Invalid query: ' . mysql_error());
}

//Initialize delay in geocode speed

$delay=0;
$base_url = "http://" . MAPS_HOST . "/maps/api/geocode/json?address=";

while ($row = mysql_fetch_assoc($initial_city)) {
    $geocode_pending = true;

```

```

while ($geocode_pending){

    $address = $row["city"];
    $id = $row["id"];

    $fullAddress = $address . ", GREECE";

    $request_url = $base_url . "" . urlencode($fullAddress) . "&sensor=false";

    sleep(2);

    $json = file_get_contents($request_url);
    $json_decoded = json_decode($json);

    $status = $json_decoded->status;

    if (strcmp($json_decoded->status, "OK") == 0) {

    $geocode_pending = false;

        $lat = $json_decoded->results[0]->geometry->location->lat;
        $lng = $json_decoded->results[0]->geometry->location->lng;

//echo 'here';

    $getcoords = sprintf("UPDATE input
        SET initial_lat = '%s', initial_lng = '%s'
        WHERE id = '%s' LIMIT 1;",
        mysql_real_escape_string($lat),
        mysql_real_escape_string($lng),
        mysql_real_escape_string($id));
    $update_result = mysql_query($getcoords);

//echo $id;

    if (!$update_result) {
        die("Invalid query: " . mysql_error());
    }
    } else {
        // failure to geocode
        $geocode_pending = false;
        echo "Address " . $fullAddress . " failed to geocode. ";
        echo "Received status " . $status . "\n";
    }

// usleep($delay);

}

}

// empty table filtered_markers before inserting new data

$preresult = mysql_query("DELETE FROM filtered_markers WHERE 1;");

```

```

if (!$preresult) {
    die("Invalid query: ' . mysql_error());
}

// calculate the distance between user's latitude, longitude and database latitudes, longitudes

$calculation = mysql_query(
    "INSERT INTO filtered_markers
    SELECT markers.*, input.city, input.input_distance, input.initial_lat, input.initial_lng,
    ( 6371 * acos ( cos ( radians(input.initial_lat) )
    * cos( radians( markers.lat ) )
    * cos( radians( markers.lng ) - radians(input.initial_lng) )
    + sin ( radians(input.initial_lat) )
    * sin( radians( markers.lat ) ) ) )
    AS distance
    FROM markers, input
    HAVING distance < input.input_distance
    ORDER BY distance
    LIMIT 0 , 100");

if (!$calculation) {
    die("Invalid query: ' . mysql_error());
}

?>

```

Phpsqlajax_genxml2.php

```

<?php
require("phpsqlajax_dbinfo.php");

function parseToXML($htmlStr)
{
    $xmlStr=str_replace('<','&lt;',$htmlStr);
    $xmlStr=str_replace('>','&gt;',$xmlStr);
    $xmlStr=str_replace('"','&quot;',$xmlStr);
    $xmlStr=str_replace("'",'&#39;',$xmlStr);
    $xmlStr=str_replace("&","&amp;",$xmlStr);
    return $xmlStr;
}

// Opens a connection to a MySQL server
$connection=mysql_connect ('localhost', $username, $password);
if (!$connection) {
    die("Not connected : ' . mysql_error());
}

// Set the active MySQL database
$db_selected = mysql_select_db($database, $connection);
if (!$db_selected) {
    die ("Can't use db : ' . mysql_error());
}

```

```

// Select all rows in the filtered_markers table

$result = mysql_query("SELECT * FROM filtered_markers WHERE 1");
if (!$result) {
    die('Invalid query: ' . mysql_error());
}

header("Content-type: text/xml");

// Start XML file, echo parent node
echo '<markers>';

// Iterate through the rows, printing XML nodes for each
while ($row = mysql_fetch_assoc($result)){
    // ADD TO XML DOCUMENT NODE
    echo '<marker ';
    echo 'title="' . parseToXML($row['title']) . "' ";
    echo 'company_name="' . parseToXML($row['company_name']) . "' ";
    echo 'url="' . parseToXML($row['url']) . "' ";
    echo 'email="' . parseToXML($row['email']) . "' ";
    echo 'city_name="' . parseToXML($row['city_name']) . "' ";
    echo 'lat="' . $row['lat'] . "' ";
    echo 'lng="' . $row['lng'] . "' ";
    echo 'categories_name="' . $row['categories_name'] . "' ";
    echo 'city="' . parseToXML($row['city']) . "' ";
    echo 'input_distance="' . $row['input_distance'] . "' ";
    echo 'initial_lat="' . $row['initial_lat'] . "' ";
    echo 'initial_lng="' . $row['initial_lng'] . "' ";
    echo 'distance="' . $row['distance'] . "' ";
    echo '/>';
}
// End XML file
echo '</markers>';

?>

```

MyMap.php

```

<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="initial-scale=1.0, user-scalable=no" />
    <meta http-equiv="content-type" content="text/html; charset=UTF-8"/>
    <title>Google Maps</title>
    <script
src="https://maps.googleapis.com/maps/api/js?key=AlzaSyCCB5rlzgCswG1npCWGb0Wg
6rIkIW9Der0"
        type="text/javascript"></script>
<script type="text/javascript">

    //<![CDATA]

    var customIcons = {
        Programming: {

```



```

    icon: 'http://labs.google.com/ridefinder/images/mm_20_blue.png'
  },
  Design: {
    icon: 'http://labs.google.com/ridefinder/images/mm_20_red.png'
  },
  Business: {
    icon: 'http://labs.google.com/ridefinder/images/mm_20_green.png'
  },
  Admin: {
    icon: 'http://labs.google.com/ridefinder/images/mm_20_yellow.png'
  },
  Support: {
    icon: 'http://labs.google.com/ridefinder/images/mm_20_white.png'
  },
  Content: {
    icon: 'http://labs.google.com/ridefinder/images/mm_20_black.png'
  }
};

```

```

function load() {

```

```

    var map = new google.maps.Map(document.getElementById("map"), {
      center: new google.maps.LatLng(39.37534, 21.67789),
      zoom: 6,
      mapTypeId: 'roadmap'
    });

```

```

    var infoWindow = new google.maps.InfoWindow;

```

```

    downloadUrl("phpsqlajax_genxml2.php", function(data) {
      var xml = data.responseXML;
      var markers = xml.documentElement.getElementsByTagName("marker");
      for (var i = 0; i < markers.length; i++) {
        var title = markers[i].getAttribute("title");
        var company_name = markers[i].getAttribute("company_name");
        var url = markers[i].getAttribute("url");
        var email = markers[i].getAttribute("email");
        var city_name = markers[i].getAttribute("city_name");
        var categories_name = markers[i].getAttribute("categories_name");
        var incity = markers[i].getAttribute("city");
        var inidist = markers[i].getAttribute("input_distance");
        var lat = markers[i].getAttribute("initial_lat");
        var lng = markers[i].getAttribute("initial_lng");
        var finaldist = markers[i].getAttribute("distance");
        var point = new google.maps.LatLng(
          parseFloat(markers[i].getAttribute("lat")),
          parseFloat(markers[i].getAttribute("lng")));
        var html = "<b>" + title + "</b> <br/>" + company_name + "</b> <br/>" + "<a href="
+ url + "</a>" + "</b> <br/>"
          + email + "</b> <br/>" + city_name + "</b> <br/>" + categories_name + "</b>
<br/>" + incity + "</b> <br/>"
          + inidist + "</b> <br/>" + lat + "</b> <br/>" + lng + "</b> <br/>" + finaldist +
"</b> <br/>" + point;
        var icon = customIcons[categories_name] || {};
        var marker = new google.maps.Marker({
          map: map,

```

```

        position: point,
        icon: icon.icon
    });
    bindInfoWindow(marker, map, infoWindow, html);
}
});
};

function bindInfoWindow(marker, map, infoWindow, html) {
    google.maps.event.addListener(marker, 'click', function() {
        infoWindow.setContent(html);
        infoWindow.open(map, marker);
    });
}

function downloadUrl(url, callback) {
    var request = window.ActiveXObject ?
        new ActiveXObject('Microsoft.XMLHTTP') :
        new XMLHttpRequest;

    request.onreadystatechange = function() {
        if (request.readyState === 4) {
            request.onreadystatechange = doNothing;
            callback(request, request.status);
        }
    };

    request.open('GET', url, true);
    request.send(null);
}

function doNothing() {}

//]]>

</script>
</head>
<body onload="load()">
    <?php include 'inputHandler.php'?>
    <div id="map" style="width: 1000px; height: 600px"></div><br/>
</body>
</html>

```