



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Διαδικτυακή εφαρμογή αντιστοίχισης αγγελιών με
βιογραφικά

Του φοιτητή
Ελληνίδα Αλέξιου
Αρ. Μητρώου: 093447

Επιβλέπων
Αντωνίου Ευστάθιος
Καθηγητής

Μάιος 2024

Τίτλος Π.Ε. Δημιουργία web εφαρμογής αντιστοίχισης αγγελιών με βιογραφικά

Κωδικός Π.Ε. 23310

Όνοματεπώνυμο φοιτητή: Ελληνίδης Αλέξιος

Όνοματεπώνυμο εισηγητή: Αντωνίου Ευστάθιος

Ημερομηνία ανάληψης Δ.Ε.: 03-11-2023

Ημερομηνία περάτωσης Δ.Ε. ...

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Ελληνίδη Αλέξιου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιέρωση»

Πρόλογος

Το παρόν κείμενο αφορά την πτυχιακή εργασία του φοιτητή Αλέξιου Ελληνίδη. Επικεντρώνεται στην υλοποίηση και παρουσίαση μιας ολοκληρωμένης διαδικτυακής εφαρμογής, η οποία αντιστοιχίζει αγγελίες εργασίας με βιογραφικά. Πρακτικά, πρόκειται για μια δυναμική ιστοσελίδα, η οποία παρέχει τη δυνατότητα στους χρήστες, είτε εργοδότες είτε υποψήφιους εργαζόμενους, να εγγραφούν και να δημοσιεύσουν τις αγγελίες και τα βιογραφικά τους αντίστοιχα. Το σύστημα επεξεργάζεται τα παραπάνω στοιχεία και τα ταιριάζει με βάση προκαθορισμένα κριτήρια, διευκολύνοντας έτσι την αναζήτηση εργασίας για τους υποψήφιους και την εύρεση κατάλληλων υποψηφίων για τους εργοδότες. Ο πρωταρχικός στόχος της εργασίας είναι να παρουσιάσει έναν δομημένο τρόπο ανάπτυξης μιας δυναμικής εφαρμογής web. Αυτό περιλαμβάνει την ανάλυση της εφαρμογής, την ανάπτυξή της, καθώς και την παρουσίαση της τελικής εφαρμογής.

Περίληψη

Σκοπός της εργασίας είναι η σχεδίαση και υλοποίηση μίας web εφαρμογής για την ανάρτηση, αναζήτηση και αντιστοίχιση αγγελιών εργασίας από επιχειρήσεις που αναζητούν προσωπικό. Παράλληλα οι ενδιαφερόμενοι υποψήφιοι θα μπορούν να μπουν στην εφαρμογή και να συμπληρώσουν μία φόρμα ως βιογραφικό.

Η κεντρική ιδέα πίσω από την εφαρμογή είναι η αντιστοίχιση των αγγελιών με τα βιογραφικά που ταιριάζουν, δημιουργώντας έτσι μια αποτελεσματική γέφυρα μεταξύ των εργοδοτών και των ενδιαφερόμενων εργαζομένων. Στο σημείο αυτό, είναι σημαντικό να σημειωθεί ότι η εφαρμογή δεν περιορίζεται μόνο στην αναζήτηση και προβολή των αγγελιών, αλλά περιλαμβάνει και έναν μηχανισμό που αντιστοιχίζει τις αγγελίες με τα βιογραφικά που ταιριάζουν.

Τα κριτήρια για την αντιστοίχιση θα είναι προκαθορισμένα από το σύστημα. Οι χρήστες (επιχειρήσεις, ενδιαφερόμενοι) θα συμπληρώνουν φόρμες (αγγελίες, βιογραφικά), που θα περιλαμβάνουν πεδία για αυτά τα κριτήρια. Πιο συγκεκριμένα, τα κριτήρια είναι τα παρακάτω:

- Μορφή απασχόλησης (Πλήρης, Μερική)
- Τομέας απασχόλησης (Πληροφορική, Marketing, Λογιστική, Τουρισμός...)
- Τόπος απασχόλησης
- Επίπεδο εκπαίδευσης (Βασική, Πτυχίο, Μεταπτυχιακό...)
- Γλώσσες (Αγγλικά, Γερμανικά, Γαλλικά...)

Web application for matching job ads with resumes

Ellinidis Alexios

Abstract

The purpose of the thesis is the design and implementation of a web application for the posting, search and matching of job advertisements from companies looking for staff. At the same time, interested candidates will be able to enter the application and fill in a form as a resume.

The central idea behind the app is to match advertisements with suitable resumes, thereby creating an effective bridge between employers and interested employees. At this point, it is important to note that the application is not only limited to searching and viewing ads, but also includes a mechanism that matches ads with matching resumes.

The criteria for matching will be predefined by the system. Users (businesses, interested parties) will fill in forms (advertisements, resumes), which will include fields for these criteria. More specifically, the criteria are the following:

- Form of employment (Full-time, Part-time)
- Sector of employment (Information, Marketing, Accounting, Tourism...)
- Place of employment
- Education level (Bachelor, Bachelor, Master...)
- Languages (English, German, French....)

Περιεχόμενα

Πρόλογος.....	iv
Περίληψη.....	v
Abstract	vi
Περιεχόμενα	vii
Κατάλογος Σχημάτων	x
Κατάλογος Πινάκων.....	xi
Συνομογραφίες.....	xii
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Διαδικτυακές εφαρμογές δημοσίευσης αγγελιών.....	1
1.2 Δομή της εργασίας	2
Κεφάλαιο 2ο: Τεχνολογίες και εργαλεία.....	3
2.1 Εισαγωγή.....	3
2.2 Γλώσσα προγραμματισμού PHP	3
2.3 HTML – CSS – Javascript.....	4
2.3.1 HTML.....	4
2.3.2 CSS – Bootstrap	7
2.3.3 JavaScript	8
2.4 MySQL.....	9
2.5 VSCode	10
2.6 Git.....	12
2.7 XAMPP	12
2.8 Composer	14
2.9 MVC.....	15
2.9.1 Εισαγωγή.....	15
2.9.2 Ιστορία.....	16
2.9.3 Τα στοιχεία του MVC	16
2.9.4 Πλεονεκτήματα – Μειονεκτήματα.....	17
2.9.5 Εναλλακτικές αρχιτεκτονικές.....	18
2.10 Επίλογος.....	19
Κεφάλαιο 3ο: Frameworks – Codeigniter	20
3.1 Εισαγωγή.....	20
3.2 Γνωστά PHP Frameworks	21

3.3	Codeigniter	22
3.3.1	Εισαγωγή	22
3.3.2	Ιστορικό	22
3.3.3	Δομή φακέλων	22
3.3.4	Μηνύματα HTTP	24
3.3.5	Ροή εφαρμογής Codeigniter	27
3.3.6	Εγκατάσταση	28
3.3.7	Ρύθμιση	28
3.3.8	Ασφάλεια	29
3.3.9	Διαχείριση σφαλμάτων	29
3.3.10	Καταγραφή συμβάντων	29
3.3.11	Κλάση QueryBuilder	30
3.3.12	Διαχείριση βάσης δεδομένων	31
3.3.13	MVC Framework	32
3.3.14	Controllers και δρομολόγηση	32
3.3.15	Δημιουργία αποκρίσεων	32
3.3.16	Models	33
3.3.17	Entities	34
3.4	Επίλογος	34
Κεφάλαιο 4ο: Ανάλυση της εφαρμογής		35
4.1	Εισαγωγή	35
4.2	Ιστορίες χρήσης	35
4.3	Κριτήρια αντιστοίχισης	36
4.4	Διαχείριση χρηστών	38
4.4.1	Εισαγωγή	38
4.4.2	Χρήση του Shield στην εφαρμογή	39
4.5	Σχήμα βάσης δεδομένων	39
4.6	Κλάσεις Model	45
4.7	Controllers	46
4.8	Views	46
4.9	Μηχανισμός αντιστοίχισης	48
4.10	Δρομολόγηση	50
Κεφάλαιο 5ο: Παρουσίαση της εφαρμογής		53
5.1	Εισαγωγή	53
5.2	Λειτουργικότητες χρηστών	55

5.2.1	Εισαγωγή.....	55
5.2.2	Χρήστης ‘Επιχείρηση’.....	55
5.2.3	Χρήστης ‘Υποψήφιος’	58
5.2.4	Χρήστης ‘Διαχειριστής’	62
Κεφάλαιο 6ο:	Συμπεράσματα – Προτάσεις βελτίωσης.....	63
6.1	Συμπεράσματα.....	63
6.2	Προτάσεις βελτίωσης	63
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		65

Κατάλογος Σχημάτων

Εικόνα 2.1 Δομή στοιχείου HTML [3]	5
Εικόνα 2.2 Δομή εγγράφου HTML.....	6
Εικόνα 2.3 Ανατομία κανόνα CSS [4]	7
Εικόνα 2.4 VSCode [7]	10
Εικόνα 2.5 Συνεργασία στο Git [8].....	12
Εικόνα 2.6 Πίνακας ελέγχου XAMPP	13
Εικόνα 2.7 Διάγραμμα αρχιτεκτονικής MVC [21]	15
Εικόνα 3.1 PHP Frameworks	21
Εικόνα 3.2 Δομή αιτήματος HTTP [15].....	24
Εικόνα 3.3 Δομή απόκρισης HTTP [15].....	25
Εικόνα 3.4 Μέθοδοι της κλάσης IncomingRequest [16].....	26
Εικόνα 3.5 Μέθοδοι της κλάσης Response [16]	26
Εικόνα 3.6 Διάγραμμα ροής Codeigniter [17]	27
Εικόνα 3.7 Παράδειγμα χρήσης QueryBuilder	31
Εικόνα 4.1 Παράδειγμα Migration κλάσης.....	40
Εικόνα 4.2 Η κλάση Deploy seeder	41
Εικόνα 4.3 Διάγραμμα βασικών οντοτήτων συσχετίσεων	42
Εικόνα 4.4 Δομή πίνακα company	43
Εικόνα 4.5 Δομή πίνακα job_ad.....	43
Εικόνα 4.6 Δομή πίνακα candidate	44
Εικόνα 4.7 Δομή πίνακα matching.....	44
Εικόνα 4.8 Δομή πίνακα education_level (ένα από τα κριτήρια αντιστοίχισης)	44
Εικόνα 4.9 Κλάση Company Model	45
Εικόνα 4.10 Αρχεία Controllers	46
Εικόνα 4.11 Κεντρικό layout view.....	47
Εικόνα 4.12 Επέκταση του layout view	47
Εικόνα 4.13 Η μέθοδος του Candidate Model που εκκινεί τη διαδικασία αντιστοίχισης.....	48
Εικόνα 4.14 Οι μέθοδοι της Matching κλάσης για την αντιστοίχιση υποψήφιου με βιογραφικά.....	49
Εικόνα 4.15 Αρχείο Routes.php	51
Εικόνα 5.1 Αρχική σελίδα εφαρμογής	53
Εικόνα 5.2 Προτροπή για εγγραφή επιχείρησης	54
Εικόνα 5.3 Προτροπή για εγγραφή υποψήφιου	54
Εικόνα 5.4 Λειτουργικότητες χρήστη 'Επιχείρηση'	55
Εικόνα 5.5 Φόρμα στοιχείων επιχείρησης	56
Εικόνα 5.6 Αγγελίες της επιχείρησης.....	57
Εικόνα 5.7 Φόρμα προσθήκης αγγελίας.....	57
Εικόνα 5.8 Αντιστοιχίσεις αγγελιών με βιογραφικά.....	58
Εικόνα 5.9 Λειτουργικότητες χρήστη 'Υποψήφιος'	59
Εικόνα 5.10 Φόρμα βιογραφικού	59
Εικόνα 5.11 Λίστα αγγελιών - φόρμα φίλτρων.....	60
Εικόνα 5.12 Προβολή λεπτομερειών αγγελίας	61
Εικόνα 5.13 Λίστα αντιστοιχίσεων με αγγελίες.....	61
Εικόνα 5.14 Δυνατότητα διαχείρισης εγγεγραμμένων χρηστών.....	62

Κατάλογος Πινάκων

Πίνακας 1 Κριτήρια αντιστοίχισης αγγελίας - βιογραφικού	37
---	----

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

Κεφάλαιο 1ο: Εισαγωγή

1.1 Διαδικτυακές εφαρμογές δημοσίευσης αγγελιών

Στη σύγχρονη εποχή, την εποχή της πληροφορίας, η ικανότητα να αναζητάμε και να βρίσκουμε την κατάλληλη θέση εργασίας μέσω του διαδικτύου, με έναν οργανωμένο και στοχευμένο τρόπο, είναι πλέον ζωτικής σημασίας. Το αντίστοιχο ισχύει και από την μεριά του εργοδότη, ο οποίος θέλει να απευθυνθεί σε ένα ευρύ πλήθος υποψηφίων. Το απέραντο αποθετήριο πληροφοριών που λέγεται διαδίκτυο, προσφέρει τόσο στους εργαζόμενους όσο και στους εργοδότες μια τεράστια πλατφόρμα για να συνδεθούν και να ανταποκριθούν στις ανάγκες της αγοράς εργασίας. Η παραπάνω διασύνδεση θα πρέπει να επιτευχθεί με τρόπο οργανωμένο και βάση κάποιον κριτηρίων από τη μεριά των ενδιαφερόμενων.

Η ταχεία εξέλιξη της τεχνολογίας, καθώς και οι προηγμένες υποδομές για γρήγορη πρόσβαση στο διαδίκτυο και οι φορητές συσκευές (όπως smartphones, tablets, laptops) που είναι πλέον προσιτές στο μεγαλύτερο ποσοστό του πληθυσμού, δίνουν το πάτημα για την ανάπτυξη εφαρμογών και λύσεων για την αντιμετώπιση του προβλήματος της ανεύρεσης εργασίας, αυξάνοντας την αποτελεσματικότητα και την αποδοτικότητα της διαδικασίας.

Στην Ελλάδα υπάρχουν διαδικτυακές εφαρμογές αγγελιών εργασίας, με κάποιες από τις πιο δημοφιλείς να είναι:

- **JobFind:** <https://www.jobfind.gr/> - Μια πλατφόρμα που σας επιτρέπει να αναζητήσετε αγγελίες εργασίας, να φτιάξετε το βιογραφικό σας και να κλείσετε ραντεβού με εργοδότες.
- **Kariera:** <https://www.kariera.gr/> - Μια δημοφιλής ελληνική ιστοσελίδα αγγελιών εργασίας με αγγελίες από διάφορους κλάδους.
- **Skywalker:** <https://www.skywalker.gr/el> Προσφέρει στους χρήστες τη δυνατότητα να αναζητήσουν θέσεις εργασίας, να καταχωρήσουν το βιογραφικό τους και να λάβουν συμβουλές για την αναζήτηση εργασίας.
- **Indeed:** <https://gr.indeed.com/> - Μια παγκόσμια μηχανή αναζήτησης εργασίας με μεγάλη ποικιλία αγγελιών στην Ελλάδα.
- **LinkedIn:** <https://www.linkedin.com/> - Ένα δίκτυο κοινωνικής δικτύωσης για επαγγελματίες όπου μπορείτε να βρείτε αγγελίες εργασίας, να δικτυωθείτε με άλλους επαγγελματίες και να προωθήσετε την καριέρα σας.
- **Χρυσή Ευκαιρία:** <https://www.xe.gr/> - Μια ιστοσελίδα με αγγελίες εργασίας από διάφορες πηγές, συμπεριλαμβανομένων εφημερίδων και ιστοσελίδων εταιρειών.
- **Αγγελίες Εργασίας:** <https://aggelies-ergasia.gr/> - Μια ιστοσελίδα με αγγελίες εργασίας από διάφορους κλάδους.

1.2 Δομή της εργασίας

Στο Κεφάλαιο 2 της παρούσας εργασίας, θα γίνει μια λεπτομερής αναφορά στις τεχνολογίες και τα χρήσιμα εργαλεία που χρησιμοποιήθηκαν κατά τη διάρκεια της ανάπτυξης της εφαρμογής. Αυτό περιλαμβάνει την παρουσίαση και την ανάλυση των χαρακτηριστικών και των λειτουργιών που παρέχουν αυτές οι τεχνολογίες και τα εργαλεία. Επίσης θα γίνει μια εκτενής επεξήγηση της αρχιτεκτονικής MVC, μιας αρχιτεκτονικής που είναι ευρέως διαδεδομένη στην ανάπτυξη web εφαρμογών και στην οποία βασίζεται και το Codeigniter.

Στη συνέχεια, στο Κεφάλαιο 3, θα εξεταστεί λεπτομερώς η χρήση ενός PHP framework, και πιο συγκεκριμένα του Codeigniter 4, το οποίο χρησιμοποιήθηκε προκειμένου να υλοποιηθεί η εφαρμογή για την εργασία αυτή.

Στο επόμενο Κεφάλαιο 4, θα γίνει μια εμπειριστατωμένη ανάλυση της εφαρμογής, παρουσιάζοντας τα κύρια στοιχεία που την αποτελούν, καθώς και τη διαδικασία υλοποίησής της.

Το Κεφάλαιο 5 αποτελεί ουσιαστικά ένα εγχειρίδιο χρήσης. Γίνεται μία παρουσίαση ανά ομάδα χρήστη

Τέλος, στο Κεφάλαιο 6, θα αναφερθούν πιθανές μελλοντικές βελτιώσεις, τόσο στη λειτουργικότητα της εφαρμογής όσο και στη διαδικασία ανάπτυξης και συντήρησης της, παρέχοντας έτσι μια πλήρη εικόνα των πιθανών εξελίξεων και βελτιώσεων που μπορούν να γίνουν στο μέλλον.

Κεφάλαιο 2ο: Τεχνολογίες και εργαλεία

2.1 Εισαγωγή

Στο παρόν κεφάλαιο, θα πραγματοποιηθεί μια λεπτομερής παρουσίαση των τεχνολογιών και των εργαλείων που επιλέχθηκαν για την υλοποίηση της εφαρμογής. Θα αναφερθούμε στις λειτουργίες, τις δυνατότητες και τους λόγους που οδήγησαν στην επιλογή αυτών των τεχνολογιών και εργαλείων, ενώ θα αναλύσουμε και τον τρόπο με τον οποίο αυτά συνέβαλαν στην ανάπτυξη και στην τελική υλοποίηση της εφαρμογής.

2.2 Γλώσσα προγραμματισμού PHP

Για την επιτυχή υλοποίηση της παρούσας εργασίας, έχει γίνει εκτεταμένη χρήση της γλώσσας προγραμματισμού PHP. Αυτή η γλώσσα προγραμματισμού, ευρέως γνωστή για την ευελιξία και την ισχύ της, ήταν ο κύριος πυρήνας του κώδικα που δημιουργήθηκε.

Η PHP είναι μία scripting (γλώσσα σεναρίων) γλώσσα προγραμματισμού, γενικού σκοπού, με την πιο δημοφιλή χρήση της να είναι για την ανάπτυξη ιστοσελίδων και δυναμικών web εφαρμογών. Είναι γνωστή για την ευκολία χρήσης και την ευελιξία της.

Χρησιμοποιείται συνήθως στον server για τη δημιουργία δυναμικών αποκρίσεων σε HTTP αιτήματα. Μπορεί, για παράδειγμα, να επεξεργάζεται μια HTML σελίδα πριν αυτή εμφανιστεί στον client, ή να δημιουργεί δυναμικά μια εικόνα.

Η PHP είναι λογισμικό ανοιχτού κώδικα, διαθέσιμη δωρεάν για όλους τους χρήστες και έχει ευρεία υποστήριξη από μια μεγάλη κοινότητα χρηστών. Έχει επίσης πληθώρα πόρων και ένα πλούσιο οικοσύστημα βιβλιοθηκών, εργαλείων και πλαισίων ανάπτυξης (frameworks).

Η PHP δημιουργήθηκε αρχικά από τον Rasmus Lerdorf το 1994 [1]. Το ακρωνύμιο PHP αρχικά σήμαινε "Personal Home Page", καθώς ο Lerdorf δημιούργησε την PHP για να διαχειριστεί την προσωπική του ιστοσελίδα. Η γλώσσα εξελίχθηκε σταδιακά και γρήγορα άρχισε να χρησιμοποιείται σε ολοένα και περισσότερες εφαρμογές ιστού.

Με την κυκλοφορία της PHP 7.4 το 2019 [2], εισήχθησαν νέα χαρακτηριστικά όπως η υποστήριξη για τον τελεστή null coalescing, την συντομευμένη σύνταξη για τα arrays και την προ-φόρτωση, που επιτρέπει στον PHP interpreter να φορτώνει τις πιο συχνά χρησιμοποιούμενες βιβλιοθήκες στη μνήμη, βελτιώνοντας την απόδοση.

Μερικές από τις κύριες δυνατότητες και χρήσεις της PHP περιλαμβάνουν την επικοινωνία με βάσεις δεδομένων, τη δημιουργία δυναμικών ιστοσελίδων, την υποστήριξη συστημάτων διαχείρισης περιεχομένου (CMS), την επεξεργασία δεδομένων, τον χειρισμό φορμών, την υλοποίηση διαδραστικών λειτουργιών, την ανάπτυξη web services και τη σύνδεση με εξωτερικές υπηρεσίες και APIs.

Όλα αυτά τα χαρακτηριστικά, σε συνδυασμό με την ευκολία εκμάθησης της γλώσσας, καθιστούν την PHP μια από τις πιο δημοφιλείς και ευρέως χρησιμοποιούμενες γλώσσες προγραμματισμού για την ανάπτυξη web εφαρμογών. Εκτός από τα παραπάνω, η PHP έχει και άλλα πλεονεκτήματα που την κάνουν δημοφιλή.

Η PHP έχει αντλήσει το μεγαλύτερο μέρος της σύνταξής της από τις γλώσσες C, Java και Perl, ενώ παράλληλα έχει προσθέσει και μερικά δικά της μοναδικά στοιχεία στην επικρατούσα σύνταξη. Αυτό την καθιστά μοναδική και ευέλικτη, ιδιαίτερα δε για τη δημιουργία δυναμικού περιεχομένου. Ο βασικός στόχος της γλώσσας είναι να επιτρέπει στους προγραμματιστές να δημιουργούν δυναμικές ιστοσελίδες, επιταχύνοντας την εφαρμογή των αλλαγών και την ανανέωση του περιεχομένου.

Συνεχίζοντας, είναι σημαντικό να σημειωθεί ότι η PHP είναι μια δυναμική γλώσσα προγραμματισμού, που εκτελείται στον server και όχι στη συσκευή του χρήστη (π.χ. στον υπολογιστή ή στο κινητό τηλέφωνο του χρήστη). Αυτό παρέχει μεγάλη ευελιξία, καθώς επιτρέπει στους προγραμματιστές να δημιουργούν δυναμικές ιστοσελίδες που μπορούν να προσαρμόζονται σε διαφορετικούς χρήστες και σε διαφορετικές καταστάσεις.

Επίσης, η PHP μπορεί να ενσωματωθεί απευθείας σε HTML κώδικα, κάτι που διευκολύνει τη διαδικασία ανάπτυξης ιστοσελίδων. Οι προγραμματιστές μπορούν να γράψουν κώδικα PHP μέσα σε HTML, δημιουργώντας έτσι δυναμικά στοιχεία σε στατικές HTML σελίδες.

Τέλος, η PHP υποστηρίζει πολλές βάσεις δεδομένων, συμπεριλαμβανομένων MySQL, PostgreSQL και Oracle, καθώς και πολλά διαφορετικά πρότυπα ασφαλείας. Αυτό την καθιστά ιδανική για την ανάπτυξη εφαρμογών που απαιτούν επεξεργασία και αποθήκευση μεγάλων ποσοτήτων δεδομένων.

Αξίζει να σημειωθεί ότι παρά τα πολλά πλεονεκτήματα της PHP, υπάρχουν και κάποιες προκλήσεις που πρέπει να αντιμετωπίσουν οι προγραμματιστές όταν την χρησιμοποιούν. Για παράδειγμα, η PHP είναι μια γλώσσα που είναι ευάλωτη σε διάφορους τύπους επιθέσεων, όπως SQL Injection και Cross-Site Scripting (XSS). Ωστόσο, με τη σωστή χρήση των πρακτικών ασφαλείας και των εργαλείων που παρέχει η PHP, οι προγραμματιστές μπορούν να μειώσουν σημαντικά αυτούς τους κινδύνους.

Επιπλέον, παρόλο που η PHP είναι ευέλικτη και ευρέως χρησιμοποιούμενη, η απόδοσή της μπορεί να μην είναι τόσο καλή όσο άλλες γλώσσες προγραμματισμού, ιδίως για πολύπλοκες ή απαιτητικές εφαρμογές. Ωστόσο, υπάρχουν πολλές τεχνικές και εργαλεία προσωποποίησης που μπορούν να βελτιώσουν την απόδοση του κώδικα PHP.

Τέλος, ενώ η PHP είναι ευρέως διαδεδομένη και υποστηρίζεται από μια μεγάλη κοινότητα, η εκμάθησή της και η εξοικείωσή με τη γλώσσα μπορεί να απαιτήσει χρόνο και προσπάθεια, ιδιαίτερα για τους νέους προγραμματιστές. Παρ' όλα αυτά, η PHP θεωρείται μια από τις πιο προσβάσιμες γλώσσες προγραμματισμού, καθιστώντας την μια εξαιρετική επιλογή για τους ανθρώπους που θέλουν να εισέλθουν στον κόσμο του web development.

2.3 HTML – CSS – Javascript

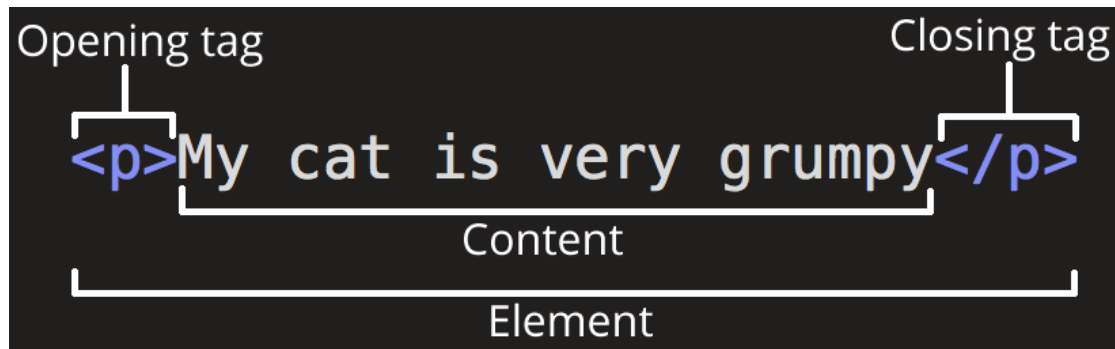
Παρακάτω θα δούμε τις βασικές τεχνολογίες – γλώσσες προγραμματισμού, οι οποίες χρησιμοποιούνται για την παρουσίαση περιεχομένου στην πλευρά του χρήστη (frontend).

2.3.1 HTML

Η HyperText Markup Language (HTML) είναι η καθιερωμένη γλώσσα σήμανσης για έγγραφα που εμφανίζονται σε έναν πρόγραμμα περιήγησης ιστού. Αποτελεί το βασικό εργαλείο δόμησης και οργάνωσης του περιεχομένου που εμφανίζεται στον παγκόσμιο ιστό. Είτε πρόκειται για απλές ιστοσελίδες με κείμενο και εικόνες, είτε για σύνθετες διαδραστικές εφαρμογές που συνδυάζουν εικόνες,

βίντεο, ήχο και αλληλεπίδραση χρήστη. Συχνά υποστηρίζεται και από άλλες τεχνολογίες όπως τα Cascading Style Sheets (CSS) για την εμφάνιση και την JavaScript για προσθήκη λειτουργικότητας.

Τα έγγραφα HTML, που τα προγράμματα περιήγησης ιστού (browsers) λαμβάνουν από έναν διακομιστή ιστού, απεικονίζονται ως πολυμεσικές ιστοσελίδες. Η HTML περιγράφει τη δομή μιας ιστοσελίδας σημασιολογικά και το πρόγραμμα περιήγησης εμφανίζει τη γραφική απεικόνιση αυτής. Ο όρος “Hypertext” ή “Υπερκείμενο” αφορά στους συνδέσμους που περιλαμβάνει ένα τέτοιο έγγραφο HTML, προς άλλα έγγραφα ή ιστοσελίδες.



Εικόνα 2.1 Δομή στοιχείου HTML [3]

Η βασική έννοια για τη δόμηση ενός εγγράφου HTML είναι τα στοιχεία ή “elements”. Με αυτά η HTML σηματοδοτεί κείμενο, εικόνες και άλλα αντικείμενα όπως διαδραστικές φόρμες μπορούν να ενσωματωθούν στην απεικονιζόμενη σελίδα. Για κάθε σημασιολογικό στοιχείο υπάρχει η αντίστοιχη ετικέτα ή “tag”. Η ονομασία της εκάστοτε ετικέτας περικλείεται από τους χαρακτήρες “<” και “>”. Ενώ τα στοιχεία περικλείονται από ετικέτα έναρξης και ετικέτα λήξης. Να σημειωθεί εδώ ότι η ετικέτα λήξης είναι ίδια με αυτήν της έναρξης, αλλά έχει επιπλέον το χαρακτήρα “/” αμέσως μετά το “<”. Σε ορισμένες περιπτώσεις, χρησιμοποιείται μία μόνο ετικέτα, η οποία ανοίγει και κλείνει μόνη της (self-closing tag). Τα προγράμματα περιήγησης δεν εμφανίζουν τα tags HTML αλλά τα χρησιμοποιούν για να ερμηνεύσουν το περιεχόμενο της σελίδας.

Η δομή μίας HTML ιστοσελίδας βασίζεται σε ιεραρχική διάταξη, με κάθε στοιχείο να ενσωματώνεται σε ένα ή περισσότερα ανώτερα στοιχεία. Η βασική ιεραρχία μίας HTML σελίδας έχει ως εξής:

- **<!DOCTYPE html>**: Η δήλωση τύπου εγγράφου, η οποία ενημερώνει το πρόγραμμα περιήγησης για την έκδοση HTML που χρησιμοποιείται.
- **<html>**: Το ριζικό στοιχείο του εγγράφου, το οποίο περιλαμβάνει όλο το περιεχόμενο της ιστοσελίδας.
- **<head>**: Το στοιχείο “κεφάλι”, το οποίο περιέχει μεταδεδομένα και πληροφορίες για την ιστοσελίδα, όπως τίτλος, κωδικοποίηση χαρακτήρων, συνδέσμους σε αρχεία CSS και JavaScript.
- **<body>**: Το στοιχείο “σώμα”, στο οποίο υπάρχει το ορατό περιεχόμενο της ιστοσελίδας, όπως κείμενο, εικόνες, συνδέσμους και άλλα στοιχεία.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-w
6     idth, initial-scale=1.0">
7     <title>Document</title>
8 </head>
9 <body>
10 </body>
11 </html>
```

Εικόνα 2.2 Δομή εγγράφου HTML

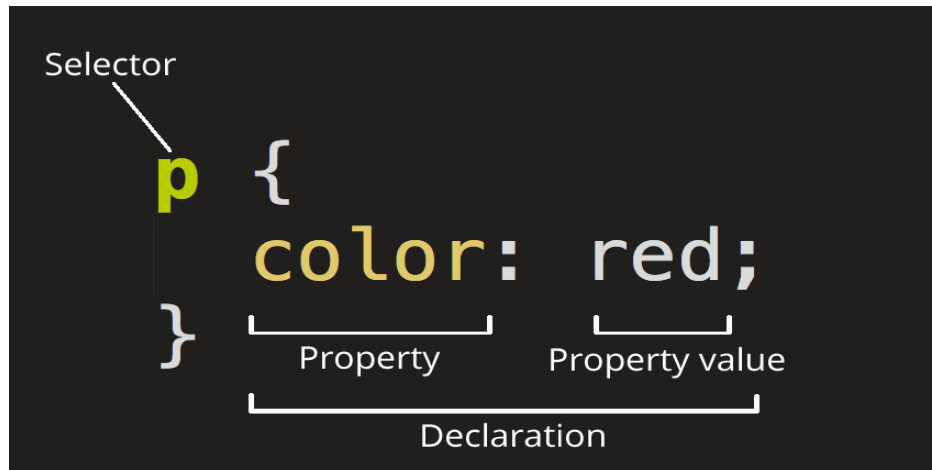
Η HTML προσφέρει πλήθος στοιχείων για τον ορισμό και την οργάνωση του περιεχομένου μίας ιστοσελίδας. Ακολουθούν κάποια βασικά:

- **<p>**: Δημιουργεί παράγραφο κειμένου.
- **<h1>** έως **<h6>**: Δημιουργούν τίτλους διαφορετικών επιπέδων (από μεγαλύτερο έως μικρότερο).
- ****: Δημιουργεί μη ταξινομημένη λίστα.
- ****: Δημιουργεί ταξινομημένη λίστα.
- ****: Δημιουργεί στοιχείο μίας λίστας.
- ****: Ενσωματώνει εικόνα στην ιστοσελίδα.
- **<a>**: Δημιουργεί σύνδεσμο προς άλλη ιστοσελίδα ή τμήμα της ίδιας σελίδας.
- **<table>**: Δημιουργεί πίνακα.
- **<tr>**: Δημιουργεί σειρά σε πίνακα.
- **<td>**: Δημιουργεί κελί σε πίνακα.

Συνοψίζοντας, η HTML δεν είναι γλώσσα προγραμματισμού, αλλά μία γλώσσα σήμανσης. Περιγράφει δηλαδή τη δομή και τις ιδιότητες μίας ιστοσελίδας και επιτρέπει την ενσωμάτωση εικόνων, βίντεο και άλλων πολυμεσικών στοιχείων σε αυτήν, καθώς και την δημιουργία συνδέσμων που επιτρέπουν την πλοήγηση στο διαδίκτυο. Είναι μια απλή αλλά ισχυρή γλώσσα που είναι ζωτικής σημασίας για οποιονδήποτε θέλει να ασχοληθεί με τον σχεδιασμό και την ανάπτυξη ιστοσελίδων.

2.3.2 CSS – Bootstrap

Τα Cascading Style Sheets (CSS) είναι μια γλώσσα στυλ που χρησιμοποιείται για να περιγράψει την εμφάνιση ενός εγγράφου γραμμένου σε HTML. Με άλλα λόγια, μέσω του CSS, οι web developers έχουν την ευελιξία να διαμορφώνουν και να προσαρμόζουν την εμφάνιση και τη διάταξη των HTML στοιχείων, καθιστώντας την ιστοσελίδα πιο ευχάριστη και λειτουργική για τον χρήστη. την εμφάνιση των στοιχείων στην οθόνη, την εμφάνιση κατά την εκτύπωση, αλλά και την εμφάνιση σε άλλες μορφές παρουσίασης. Δημιουργήθηκε για να γίνει εφικτός ο διαχωρισμός μεταξύ του περιεχομένου και του τρόπου εμφάνισης αυτού.



Εικόνα 2.3 Ανατομία κανόνα CSS [4]

Το CSS καθορίζει την εμφάνιση των HTML στοιχείων μέσω των επιλογέων, των ιδιοτήτων και των τιμών. Οι επιλογείς είναι τρόποι που μπορούμε να επιλέξουμε και να διαμορφώσουμε συγκεκριμένα HTML στοιχεία μέσα σε ένα έγγραφο. Οι ιδιότητες, αφορούν το επιλεγμένο HTML στοιχείο, και είναι ο τρόπος που μπορούμε να προσαρμόσουμε την εμφάνισή του. Τέλος οι τιμές είναι ο τρόπος για να προσδιορίσουμε μία ιδιότητα.

Το CSS προσφέρει πολλά πλεονεκτήματα, όπως την ευελιξία, τη δυνατότητα επαναχρησιμοποίησης και τη δυνατότητα ακριβούς ελέγχου της διάταξης σε πολλαπλές σελίδες ταυτόχρονα. Επιπλέον, το CSS επιτρέπει την προσαρμογή της εμφάνισης της ιστοσελίδας σε διάφορες συσκευές και μεγέθη οθόνης.

Το CSS είναι γνωστό και για τη δυνατότητά του να δημιουργεί πολύπλοκες διατάξεις και εφέ στοιχείων που αυξάνουν την αισθητική και την αλληλεπίδραση μιας ιστοσελίδας. Παραδείγματα αυτών των δυνατοτήτων περιλαμβάνουν την δημιουργία πολυεπίπεδων μενού, την αλλαγή των στοιχείων κατά την κίνηση του ποντικιού, και την εφαρμογή εφέ μεταβάσεων και μετασχηματισμών.

Επιπλέον, το CSS επιτρέπει τη χρήση των media queries, τα οποία επιτρέπουν στους σχεδιαστές να εφαρμόζουν διαφορετικά στυλ ανάλογα με τα χαρακτηριστικά της συσκευής που χρησιμοποιείται (όπως το πλάτος της οθόνης, το είδος της συσκευής, η προσανατολισμός κ.ά.). Αυτό επιτρέπει την πραγματοποίηση του αποκαλούμενου "Responsive Web Design", το οποίο εξασφαλίζει ότι οι ιστοσελίδες θα εμφανίζονται σωστά και θα είναι λειτουργικές σε οποιοδήποτε μέγεθος οθόνης και συσκευή.

Τέλος, το CSS είναι απαραίτητο για την προσβασιμότητα στον ιστό. Η προσβασιμότητα είναι η πρακτική της κατάρτισης ιστοσελίδων ώστε να είναι προσβάσιμες και χρησιμοποιήσιμες από όλους τους χρήστες, ανεξάρτητα από τις ικανότητες ή τις τεχνολογίες που χρησιμοποιούν [5].

Στη διαδικασία υλοποίησης μιας νέας ιστοσελίδας, συχνά επιλέγεται ένα CSS framework ως εργαλείο για την ανάπτυξη του σχεδιασμού και της διάταξης της σελίδας. Ένα CSS framework είναι μία συλλογή από προκαθορισμένους κανόνες CSS, οι οποίοι έχουν σκοπό να διευκολύνουν τη διαδικασία προγραμματισμού των web εφαρμογών.

Αυτά τα πλαίσια περιλαμβάνουν μια σειρά από κανόνες, οι οποίοι έχουν ήδη ορισμένες ιδιότητες και τιμές, και ανατίθενται σε προκαθορισμένους επιλογείς (classes). Αυτό σημαίνει ότι ο προγραμματιστής μπορεί να εφαρμόσει έναν έτοιμο κανόνα που παρέχεται από το framework, χρησιμοποιώντας απλά το όνομα της κλάσης (class name).

Τα CSS frameworks δημιουργήθηκαν για να αυξήσουν την αποτελεσματικότητα και την ταχύτητα της ανάπτυξης ιστοσελίδων, καθώς μειώνουν την ανάγκη για επανειλημμένη εισαγωγή κοινών κανόνων CSS. Με αυτόν τον τρόπο, τα CSS frameworks εξοικονομούν χρόνο και προσφέρουν σταθερότητα και συνέπεια στον σχεδιασμό των ιστοσελίδων.

Το Bootstrap, που είναι ένα από τα πιο διάσημα CSS frameworks [6], αποτελεί έναν εξαιρετικό εργαλείο για τους web developers. Αυτό το framework, που είναι ανοιχτού κώδικα, δίνει πρόσβαση σε μια πληθώρα ετοιμών λύσεων, επιταχύνοντας την διαδικασία ανάπτυξης ιστοσελίδων. Το Bootstrap υποστηρίζει εξ ορισμού το "Responsive Web Design", δηλαδή την αυτόματη προσαρμογή της εμφάνισης μιας ιστοσελίδας ανάλογα με το μέγεθος της οθόνης της συσκευής που χρησιμοποιείται για την προβολή της. Εκτός από τους έτοιμους κανόνες CSS που παρέχει, το Bootstrap μπορεί, προαιρετικά, να περιλαμβάνει και κώδικα JavaScript, για την προσθήκη δυναμικών λειτουργικοτήτων στις ιστοσελίδες.

2.3.3 JavaScript

Η JavaScript είναι μια δυναμική, αντικειμενοστραφής, υψηλού επιπέδου γλώσσα προγραμματισμού ιστού. Αν και είναι κυρίως γνωστή ως γλώσσα σεναρίων για ιστοσελίδες, χρησιμοποιείται επίσης σε περιβάλλοντα εκτός των προγραμμάτων περιήγησης, όπως το Node.js.

Ο πρωταρχικός σκοπός της JavaScript είναι να κάνει τις σελίδες διαδραστικές και άμεσες. Αυτό επιτυγχάνεται με την εκτέλεση στο πρόγραμμα περιήγησης του πελάτη, το οποίο σημαίνει ότι ο κώδικας JavaScript εκτελείται στον υπολογιστή του χρήστη και όχι στον διακομιστή. Αυτό σημαίνει ότι η JavaScript μπορεί να εκτελεί τις λειτουργίες της αμέσως, χωρίς να χρειάζεται να περιμένει για αποκρίσεις από τον διακομιστή.

Τα προγράμματα περιήγησης στο Internet χρησιμοποιούν την JavaScript για να βελτιώσουν την εμπειρία του χρήστη, προσθέτοντας διαδραστικότητα στις ιστοσελίδες. Αυτό περιλαμβάνει δυνατότητες όπως: η δυνατότητα να αλλάζει εικόνες όταν περνάτε το ποντίκι πάνω τους, να εμφανίζει πληροφορίες όταν κάνετε κλικ σε ένα στοιχείο, ή να εκτελεί υπολογισμούς για φόρμες χωρίς να χρειάζεται να ανανεώσει τη σελίδα.

Η JavaScript είναι αντικειμενοστραφής, πράγμα που σημαίνει ότι τα δεδομένα οργανώνονται σε αντικείμενα τα οποία είναι συλλογές από δεδομένα και λειτουργίες που έχουν σχέση με αυτά τα δεδομένα. Είναι επίσης γλώσσα υψηλού επιπέδου, επειδή είναι ανεξάρτητη από την ειδική

υλικολογισμική αρχιτεκτονική του υπολογιστή. Αυτό σημαίνει ότι οι προγραμματιστές μπορούν να δημιουργήσουν λειτουργικό κώδικα χωρίς να γνωρίζουν τις λεπτομέρειες της λειτουργίας του υπολογιστή στον οποίο θα τρέξει ο κώδικας.

Παρόλο που η JavaScript περιέχει τη λέξη "Java" στο όνομά της, είναι σημαντικό να σημειωθεί ότι δεν έχει καμία σχέση με τη γλώσσα προγραμματισμού Java. Αυτή η ονομασία ήταν στην πραγματικότητα αποτέλεσμα μιας στρατηγικής μάρκετινγκ, που επιχειρούσε να εκμεταλλευτεί την τότε αυξανόμενη δημοτικότητα της Java. Αυτή η σύγχυση όσον αφορά την ονομασία των δύο γλωσσών προγραμματισμού είναι κάτι που συνεχίζει να προκαλεί παρανοήσεις μέχρι και σήμερα.

2.4 MySQL

Η MySQL είναι ένα εξαιρετικά προηγμένο, και δημοφιλές, open-source σύστημα διαχείρισης βάσεων δεδομένων (RDBMS), το οποίο βασίζεται στη γλώσσα προγραμματισμού SQL. Η αξιοπιστία της και η ταχύτητα με την οποία μπορεί να επεξεργαστεί δεδομένα την καθιστούν ιδιαίτερα ελκυστική για εφαρμογές που λειτουργούν στο διαδίκτυο.

Η MySQL είναι ευρέως αποδεκτή και χρησιμοποιείται σε πολλές εφαρμογές, όπως στην ανάπτυξη ιστοσελίδων, στην αποθήκευση και τη διαχείριση δεδομένων και στη δημιουργία αναφορών. Συχνά βλέπουμε την MySQL ως μέλος της στοίβας LAMP (Linux, Apache, MySQL, Perl/PHP/Python), καθώς αυτή η πλατφόρμα ανάπτυξης ιστού είναι εξαιρετικά δημοφιλής.

Η εταιρεία που δημιούργησε τη MySQL, MySQL AB, ιδρύθηκε από τους Σουηδούς David Axmark, Allan Larsson και τον Φινλανδό Michael Widenius. Το 2008, η Oracle Corporation εξαγόρασε την MySQL AB, κάνοντας τη MySQL να γίνει μέρος της οικογένειας προϊόντων της Oracle. Πολλές μεγάλες εταιρείες, όπως Google, Facebook, Twitter και YouTube, χρησιμοποιούν τη MySQL για τη διαχείριση των πολύτιμων δεδομένων τους.

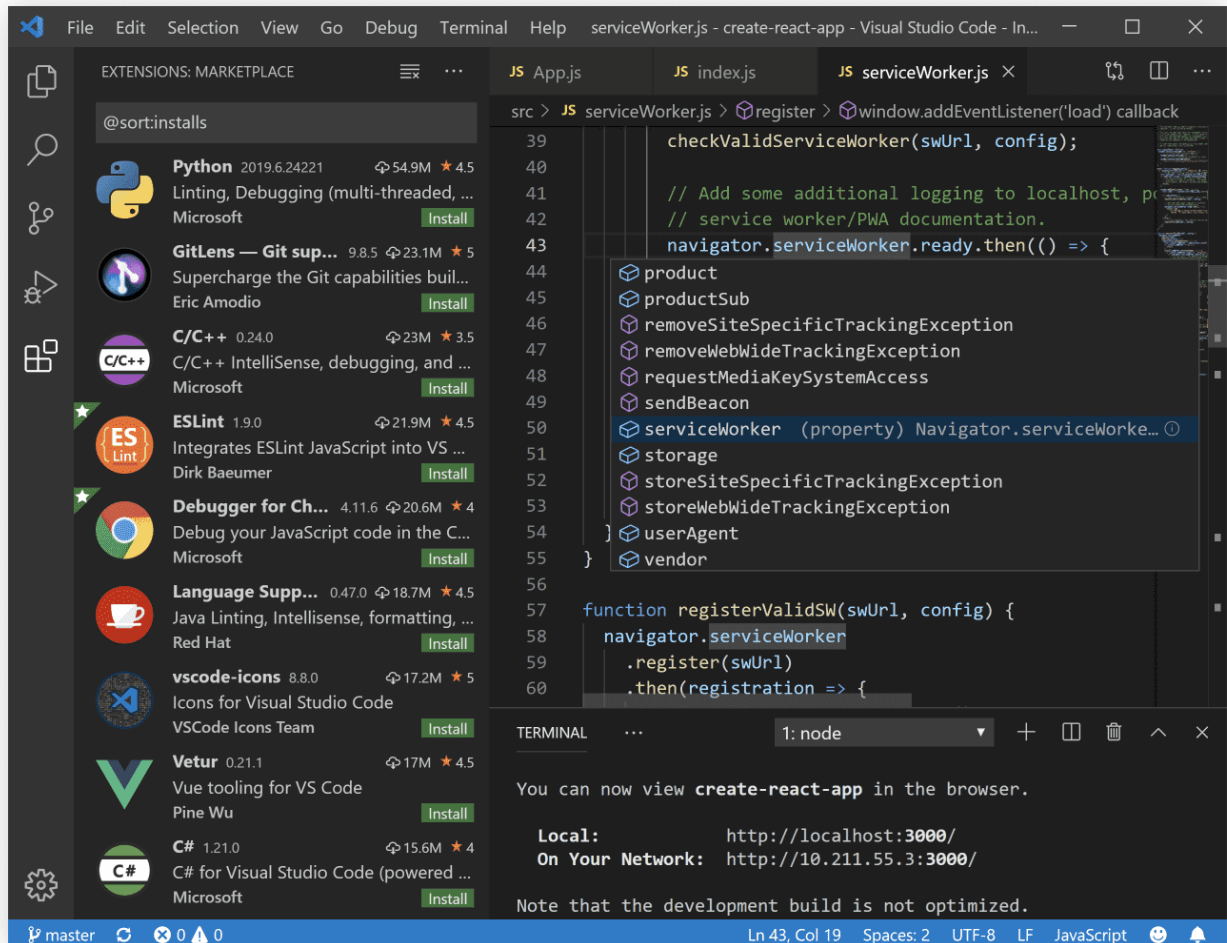
Η MySQL είναι μία πλατφόρμα διαχείρισης βάσεων δεδομένων, η οποία διακρίνεται για την εξαιρετική της απόδοση. Επιτρέπει στον χρήστη να αποθηκεύει, να αναζητά, να ταξινομεί και να βρίσκει τα δεδομένα του, με αποτελεσματικό τρόπο. Αυτό επιτυγχάνεται καθώς υποστηρίζει τη γλώσσα SQL, που είναι η καθιερωμένη γλώσσα ερωτημάτων, επιτρέποντας την απρόσκοπτη ενσωμάτωση και εκτέλεση διαφόρων ερωτημάτων. Άλλο ένα χαρακτηριστικό όπως το ότι είναι πολυνηματική, δηλαδή μπορεί να εξυπηρετήσει πολλούς πελάτες για πολλαπλές βάσεις ταυτόχρονα. Επίσης είναι προσβάσιμη μέσω δικτύου, οπότε μπορεί να διαμοιράσει δεδομένα και στο διαδίκτυο. Πέραν αυτού, παρέχει πολλές επεκτάσεις, οι οποίες συμβάλλουν σημαντικά στη βελτίωση της ευκολίας χρήσης και της απόδοσης του συστήματος. Η MySQL είναι επίσης γνωστή για την ισχυρή ασφάλεια που προσφέρει. Κάτι τέτοιο επιτρέπει στους προγραμματιστές να καθορίζουν ευέλικτα επίπεδα πρόσβασης στις βάσεις δεδομένων, δίνοντάς τους τη δυνατότητα να διαμορφώνουν προσαρμοσμένα δικαιώματα για τους χρήστες, έτσι ώστε να διασφαλίζεται η επιθυμητή προστασία των δεδομένων.

Λόγω της συμβατότητάς της με όλα τα μεγάλα λειτουργικά συστήματα - συμπεριλαμβανομένων των Linux, UNIX, Windows και macOS - η MySQL λειτουργεί εξαιρετικά καλά σε κάθε περιβάλλον. Επιπλέον, είναι επεκτάσιμη και μπορεί να χειριστεί τόσο μικρές όσο και μεγάλες βάσεις δεδομένων, υποστηρίζοντας εκατομμύρια γραμμές.

Τέλος, η MySQL προσφέρει μια σειρά από εργαλεία για την ανάλυση και την επεξεργασία δεδομένων, καθιστώντας την μια πολύτιμη πλατφόρμα για τη διαχείριση δεδομένων. Παρά την πολυπλοκότητά της,

Κεφάλαιο 2

η MySQL διατηρεί μια φιλική προς τον χρήστη διεπαφή, η οποία καθιστά εύκολη την εκμάθηση και τη χρήση του συστήματος.



Εικόνα 2.4 VSCode [7]

2.5 VSCode

Το Visual Studio Code (VS Code) είναι ένας ελεύθερος, ανοιχτού κώδικα, επεξεργαστής κώδικα ο οποίος αναπτύσσεται και συντηρείται από τη Microsoft. Είναι ιδιαίτερα δημοφιλές μεταξύ των προγραμματιστών web, λόγω των πολλαπλών χαρακτηριστικών που προσφέρει. Υποστηρίζει πολλές γλώσσες προγραμματισμού, συμπεριλαμβανομένης της PHP, που είναι κοινή στην ανάπτυξη web εφαρμογών.

Το VS Code απλοποιεί την ανάπτυξη web εφαρμογών με PHP μέσα από ποικίλα χαρακτηριστικά. Πρώτον, προσφέρει συμπλήρωση κώδικα σε πραγματικό χρόνο, κάτι που επιταχύνει την ανάπτυξη και μειώνει τις πιθανότητες λάθους. Δεύτερον, το VS Code παρέχει ενσωματωμένη υποστήριξη για debugging, κάτι που επιτρέπει στους developers να εντοπίζουν και να διορθώνουν τα λάθη

αποτελεσματικά. Τρίτον, υποστηρίζει την οργάνωση κώδικα μέσα από τη χρήση snippets, που μπορούν να αυξήσουν την παραγωγικότητα.

Όσον αφορά τα σχετικά πρόσθετα(extensions/plugins), το VSCode διαθέτει ένα τεράστιο αποθετήριο από δωρεάν επιλογές. Μία από αυτές, το PHP Inteliphense, είναι ένα εξαιρετικά χρήσιμο πρόσθετο για την ανάπτυξη PHP στο VS Code. Προσφέρει εξαιρετικά γρήγορη συμπλήρωση κώδικα, βελτιωμένη απόδοση, μειωμένη κατανάλωση μνήμης και υποστήριξη για σύγχρονες PHP λειτουργίες. Το PHP Inteliphense επίσης περιλαμβάνει βελτιωμένη ανάλυση τύπων, που μπορεί να βοηθήσει στην εύρεση πιθανών λαθών κατά τη διάρκεια της ανάπτυξης.

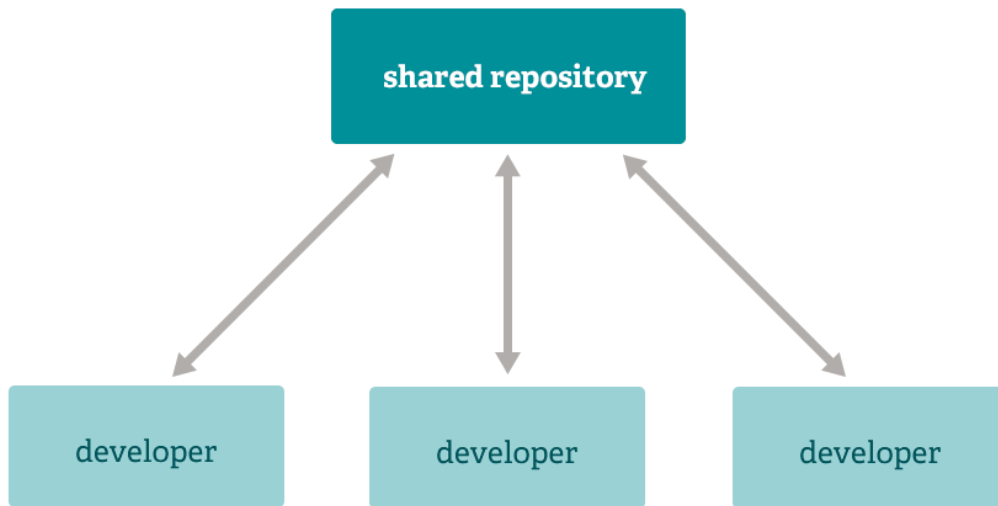
Το PHP Debug είναι ένα άλλο σημαντικό πρόσθετο που προσφέρει το VS Code. Επιτρέπει στους προγραμματιστές να εκτελούν αποσφαλμάτωση (debugging) του κώδικά τους απευθείας από τον επεξεργαστή, προσφέροντας δυνατότητες όπως τα breakpoints, την παρακολούθηση τιμών μεταβλητών και την εκτέλεση βήμα-προς-βήμα.

Επιπλέον, το VS Code ενσωματώνει εργαλεία ελέγχου έκδοσης, όπως το Git. Αυτό επιτρέπει στους προγραμματιστές να παρακολουθούν και να διαχειρίζονται τις αλλαγές στον κώδικα, να δημιουργούν νέες εκδόσεις και να συνεργάζονται αποδοτικά με άλλους προγραμματιστές. Το VS Code επίσης υποστηρίζει τη συνεργασία σε πραγματικό χρόνο μέσω του Live Share, επιτρέποντας στους προγραμματιστές να μοιράζονται και να επεξεργάζονται τον κώδικα με άλλους σε πραγματικό χρόνο.

Όλα αυτά τα χαρακτηριστικά και εργαλεία καθιστούν το VS Code έναν ισχυρό επεξεργαστή κώδικα για την ανάπτυξη web εφαρμογών με PHP. Με την ευελιξία, την ευκολία χρήσης και την υψηλή προσαρμοστικότητά του, το VS Code μπορεί να βοηθήσει τους προγραμματιστές να αυξήσουν την παραγωγικότητά τους και να παράγουν κώδικα υψηλής ποιότητας.

Πέραν των παραπάνω, το VS Code παρέχει και άλλες λειτουργίες που το καθιστούν ιδιαίτερα δημοφιλές. Για παράδειγμα, διαθέτει ενσωματωμένο τερματικό, το οποίο επιτρέπει στους προγραμματιστές να εκτελούν εντολές και να διαχειρίζονται τις εφαρμογές τους απευθείας μέσα από το περιβάλλον του επεξεργαστή. Επίσης, το VS Code υποστηρίζει την προσαρμογή του περιβάλλοντος εργασίας μέσω της εγκατάστασης γραφικών θεμάτων και πρόσθετων(όπως προαναφέρθηκε), επιτρέποντας στους προγραμματιστές να προσαρμόσουν το εργαλείο στις προτιμήσεις τους.

Σε γενικές γραμμές, το Visual Studio Code είναι μια ισχυρή, ευέλικτη και προσαρμοστική λύση για την ανάπτυξη εφαρμογών λογισμικού, και ειδικά σε PHP. Ανεξάρτητα από το επίπεδο εμπειρίας ή τις ανάγκες του προγραμματιστή, το VS Code μπορεί να βοηθήσει στην αύξηση της παραγωγικότητας, στην αποτελεσματικότητα του debugging και στην παραγωγή κώδικα υψηλής ποιότητας.



Εικόνα 2.5 Συνεργασία στο Git [8]

2.6 Git

Το git είναι ένα λογισμικό που χρησιμοποιείται για την παρακολούθηση αλλαγών σε αρχεία και τη συνεργασία μεταξύ πολλαπλών προγραμματιστών. Είναι ανοιχτού κώδικα και διατίθεται δωρεάν στους χρήστες. Το Git είναι ιδιαίτερα χρήσιμο σε μεγάλες ομάδες προγραμματιστών, όπου πολλοί άνθρωποι μπορεί να εργάζονται παράλληλα στο ίδιο έργο λογισμικού. Το σύστημα ελέγχου εκδόσεων του Git βοηθά στη διαχείριση και την επίλυση συγκρούσεων κώδικα, επιτρέποντας την αποτελεσματική συνεργασία και την ομαλή ανάπτυξη του λογισμικού.

Κύριες δυνατότητες που έχει το Git, και το καθιστούν χρήσιμο έως απαραίτητο για τη διαδικασία ανάπτυξης λογισμικού, είναι ότι επιτρέπει στους προγραμματιστές να δημιουργούν πολλαπλές ανεξάρτητες γραμμές ανάπτυξης (branches), να μεταβούν ανάμεσα σε αυτές και να συνενώσουν τις αλλαγές τους όταν είναι έτοιμες.

2.7 XAMPP

Το XAMPP είναι ένα πακέτο λογισμικού ανοιχτού κώδικα, το οποίο διατίθεται δωρεάν και είναι εξαιρετικά απλό να εγκατασταθεί και να χρησιμοποιηθεί. Το XAMPP συμπεριλαμβάνει τα Apache, MariaDB (MySQL), PHP και Perl, που παρέχουν όλα τα απαραίτητα εργαλεία για την ανάπτυξη και τη δοκιμή ιστοσελίδων.

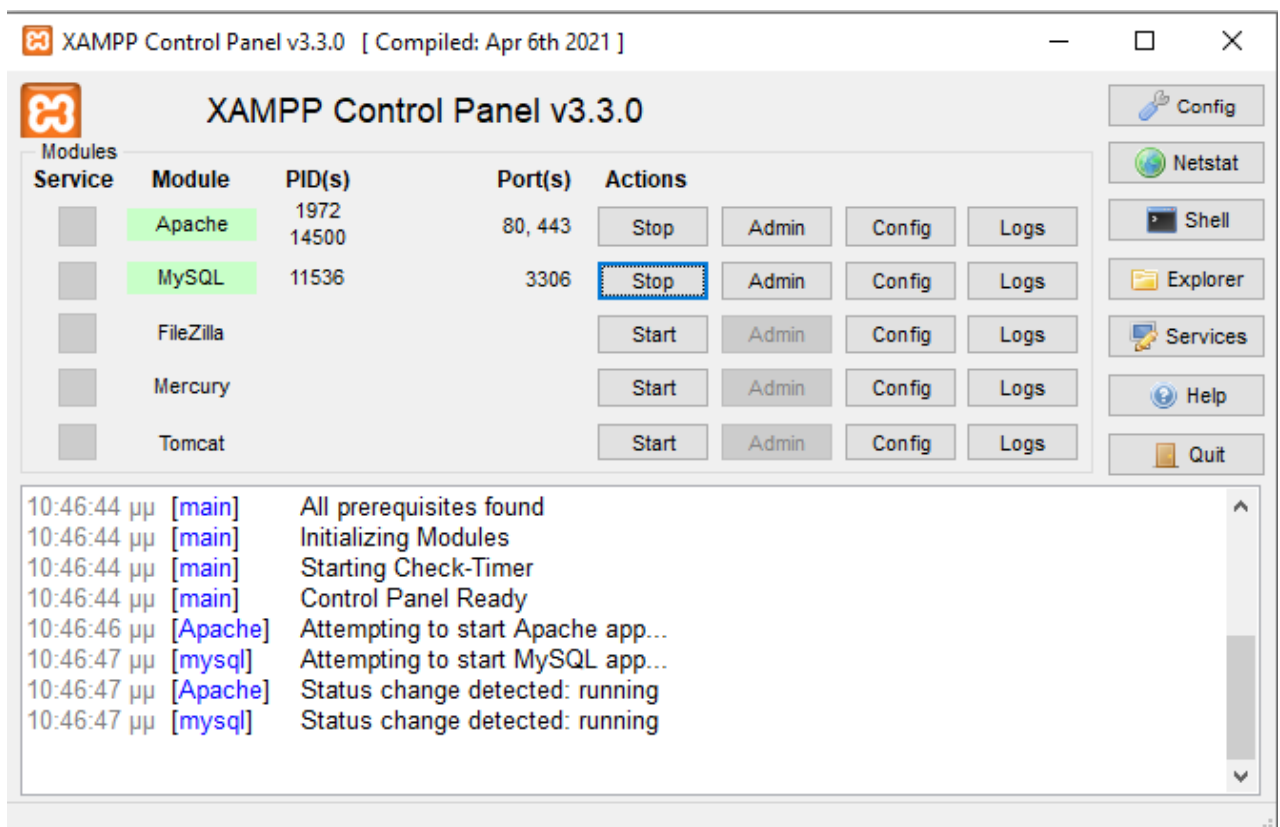
Η εγκατάσταση του XAMPP είναι ευθεία και απλή. Μετά την εγκατάσταση, μπορείτε να ξεκινήσετε τον Web Server με ένα απλό κλικ, και να αρχίσετε να αναπτύσσετε τις ιστοσελίδες σας στο localhost.

Στην ουσία το XAMPP παρέχει στους προγραμματιστές ένα τοπικό περιβάλλον για την ανάπτυξη διαδικτυακών εφαρμογών, έτσι το XAMPP είναι ιδιαίτερα χρήσιμο και δημοφιλές στους προγραμματιστές ιστοσελίδων, καθώς παρέχει ένα περιβάλλον ανάπτυξης που είναι εύκολο να ρυθμιστεί και να χρησιμοποιηθεί. Μπορείτε να τρέξετε τις εφαρμογές σας σε ένα τοπικό περιβάλλον ανάπτυξης πριν τις δημοσιεύσετε σε έναν παραγωγικό διακομιστή.

Το XAMPP απλοποιεί τη διαδικασία δημιουργίας και ελέγχου ιστοσελίδων. Μπορείτε να χρησιμοποιήσετε το XAMPP για να εγκαταστήσετε και να ρυθμίσετε το Apache, την MySQL και την PHP στον υπολογιστή σας. Επιπλέον, το XAMPP περιλαμβάνει έναν ενσωματωμένο FTP διακομιστή, ώστε μπορείτε να ανεβάσετε και να κατεβάσετε αρχεία στον τοπικό σας διακομιστή. Το XAMPP περιλαμβάνει επίσης το phpMyAdmin, ένα εργαλείο διαχείρισης βάσεων δεδομένων που σας επιτρέπει να δημιουργείτε, να επεξεργάζεστε και να διαγράφετε βάσεις δεδομένων MySQL. Το XAMPP είναι επίσης εξοπλισμένο με μια σειρά από επιπλέον ενότητες, όπως το OpenSSL και το phpMyAdmin. Αυτές οι δυνατότητες κάνουν το XAMPP έναν ισχυρό εργαλείο για την ανάπτυξη και τη δοκιμή ιστοσελίδων.

Εκτός από όλες τις προαναφερθείσες ιδιότητες, ένα ακόμα χαρακτηριστικό που το κάνει τόσο δημοφιλές είναι ότι υπάρχει μεγάλη κοινότητα προγραμματιστών που το χρησιμοποιούν. Αυτό έχει ως αποτέλεσμα να μπορεί κάποιος προγραμματιστής να αναζητήσει υποστήριξη σε κοινά προβλήματα, τα οποία έχουν ήδη επιλυθεί.

Παρόλο που το XAMPP είναι εξαιρετικά χρήσιμο για την ανάπτυξη και τη δοκιμή ιστοσελίδων, δεν προτείνεται για χρήση σε παραγωγικά περιβάλλοντα λόγω θεμάτων ασφάλειας. Προεπιλεγμένα, το XAMPP εγκαθίσταται με απενεργοποιημένη ασφάλεια, καθιστώντας το ευάλωτο σε επιθέσεις. Ωστόσο, οι χρήστες μπορούν να ρυθμίσουν τις επιλογές ασφαλείας για να καταστήσουν το XAMPP πιο ασφαλές.



Εικόνα 2.6 Πίνακας ελέγχου XAMPP

Η εγκατάσταση του XAMPP είναι απλή και γρήγορη. Αρχικά, κατεβάστε το XAMPP από την επίσημη ιστοσελίδα. Εκτελέστε τον οδηγό εγκατάστασης και ακολουθήστε τις οδηγίες που εμφανίζονται στην οθόνη. Μόλις ολοκληρώσετε την εγκατάσταση, θα μπορείτε να ξεκινήσετε το XAMPP από την επιφάνεια εργασίας ή το μενού 'Εκκίνηση'. Μετά την εγκατάσταση, μπορείτε να διαχειριστείτε τις υπηρεσίες του XAMPP, όπως το Apache ή το MySQL, μέσω του πίνακα ελέγχου του XAMPP. Εδώ, μπορείτε να ξεκινήσετε, να σταματήσετε, να επανεκκινήσετε και να ελέγξετε την κατάσταση των υπηρεσιών σας.

Σε σύνοψη, το XAMPP είναι ένα ισχυρό εργαλείο για τους προγραμματιστές ιστοσελίδων. Προσφέρει ένα πλήρες περιβάλλον ανάπτυξης που είναι εύκολο να εγκατασταθεί και να ρυθμιστεί, επιτρέποντάς σας να αναπτύξετε και να δοκιμάζετε ιστοσελίδες σε διάφορες γλώσσες προγραμματισμού. Ωστόσο, λόγω θεμάτων ασφάλειας, δεν προτείνεται για χρήση σε παραγωγικά περιβάλλοντα.

2.8 Composer

Το Composer [9] είναι ένα εργαλείο διαχείρισης εξαρτήσεων για τη γλώσσα προγραμματισμού PHP. Παρέχει έναν τυποποιημένο τρόπο για τη διαχείριση των βιβλιοθηκών που χρησιμοποιούνται σε μία εφαρμογή PHP.

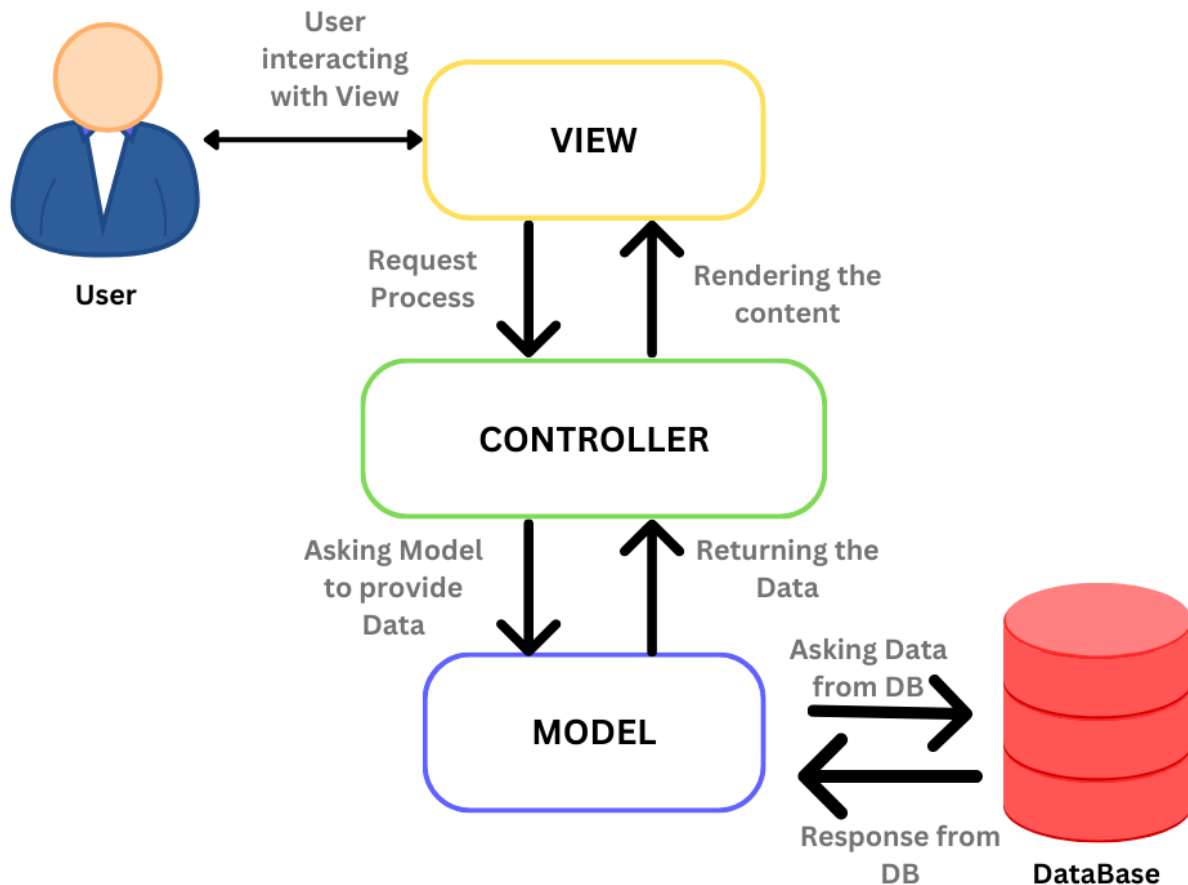
Το Composer επιτρέπει στους προγραμματιστές να δηλώσουν και να εγκαταστήσουν τις βιβλιοθήκες που απαιτούνται για την ανάπτυξη μίας συγκεκριμένης εφαρμογής και αναλαμβάνει τη διαχείριση (τις εγκαθιστά/αναβαθμίζει/απεγκαθιστά) για αυτούς. Αυτό σημαίνει ότι το Composer χειρίζεται τα πακέτα (ή βιβλιοθήκες) και τις εξαρτήσεις τους, επιτρέποντας στον προγραμματιστή να επικεντρωθεί στην κύρια δουλειά του, τον προγραμματισμό.

Το Composer παρέχει επίσης τη δυνατότητα αυτόματης φόρτωσης κλάσεων στα PHP scripts, καθιστώντας την διαχείριση εξαρτήσεων πολύ πιο απλή και ομαλή. Επιπρόσθετα παρέχει τη δυνατότητα για εκτέλεση εξατομικευμένων script.

Κάτι άλλο που καθιστά το Composer ιδιαίτερα χρήσιμο είναι η ικανότητά του να διαχειρίζεται εκδόσεις. Καθώς ο κώδικας γίνεται πιο πολύπλοκος και χρησιμοποιεί περισσότερες βιβλιοθήκες, είναι σημαντικό να μπορείς να ελέγχεις ποιες εκδόσεις των βιβλιοθηκών χρησιμοποιούνται. Το Composer κάνει αυτό το έργο ευκολότερο, επιτρέποντας στους προγραμματιστές να καθορίσουν συγκεκριμένες εκδόσεις για κάθε βιβλιοθήκη.

Τέλος, το Composer είναι συμβατό με πολλά δημοφιλή πλαίσια ανάπτυξης PHP, όπως το Codeigniter, το Laravel και το Symfony, πράγμα που το καθιστά ακόμη πιο πολύτιμο εργαλείο για τους προγραμματιστές PHP.

2.9 MVC



Εικόνα 2.7 Διάγραμμα αρχιτεκτονικής MVC [21]

2.9.1 Εισαγωγή

Το MVC (Model-View-Controller) είναι ένα δημοφιλές και διαδεδομένο σχεδιαστικό μοντέλο που χρησιμοποιείται συχνά στην ανάπτυξη λογισμικού. Αυτή η προσέγγιση διαχωρίζει με πρακτικό τρόπο την εφαρμογή σε τρία κύρια στοιχεία, τα οποία συνεργάζονται αποτελεσματικά για την υλοποίηση των στόχων της εφαρμογής. Το πρώτο στοιχείο είναι το Μοντέλο, το οποίο ασχολείται με τη διαχείριση των δεδομένων και την επικοινωνία με τη βάση δεδομένων. Το δεύτερο στοιχείο είναι η Προβολή, που είναι αυτό που βλέπει ο χρήστης και παρουσιάζει τα δεδομένα που παρέχει το μοντέλο. Το τρίτο και τελευταίο στοιχείο είναι ο Ελεγκτής, που ελέγχει την είσοδο του χρήστη και συνδέει το μοντέλο με την προβολή.

Το πρότυπο MVC, γνωστό και ως Model-View-Controller, είναι ένας αρχιτεκτονικός σχεδιασμός που εφαρμόζεται ευρέως σε λογισμικό που απαιτεί μια γραφική διεπαφή χρήστη. Αυτός ο σχεδιασμός είναι ιδιαίτερα δημοφιλής σε web εφαρμογές, καθώς το MVC μοντέλο παρέχει μια ισχυρή και ευέλικτη δομή για την ανάπτυξη εφαρμογών, διευκολύνοντας τη διαχείριση των δεδομένων, της λογικής και της διεπαφής χρήστη με έναν καθορισμένο και οργανωμένο τρόπο.

2.9.2 Ιστορία

Το Model-View-Controller (MVC), ένα σχέδιο αρχιτεκτονικής που ξεκίνησε να αναπτύσσεται στα τέλη της δεκαετίας του '70 από την Smalltalk [10], παρουσιάζει έναν ορόσημο στην εξέλιξη της πληροφορικής. Η Smalltalk αποτελεί μια πρωτοποριακή γλώσσα προγραμματισμού που έχει συμβάλει σημαντικά στην εξέλιξη των αντικειμενοστραφών γλωσσών προγραμματισμού, όπως η Java και η C++. Ο σχεδιασμός MVC, που έχει ενσωματωθεί πλέον ως πρότυπο στην ανάπτυξη λογισμικού, επιτρέπει μια πιο ολοκληρωμένη οργάνωση και διαχείριση του κώδικα.

Η κύρια φιλοσοφία πίσω από την επιτυχία της μεθόδου MVC είναι η διάκριση της εφαρμογής σε τρία διαφορετικά επίπεδα: το μοντέλο, την προβολή και τον ελεγκτή. Κάθε ένα από αυτά τα επίπεδα έχει τις δικές του ευθύνες, δημιουργώντας έτσι μια καθαρή διαχωριστική γραμμή μεταξύ της λογικής της εφαρμογής και της παρουσίασης των δεδομένων. Η εφαρμογή της MVC παρέχει ευελιξία και προσαρμοστικότητα στην ανάπτυξη λογισμικού, επιτυγχάνοντας την αποτελεσματική διαχείριση των πόρων και την παραγωγικότητα των προγραμματιστών.

Το πρότυπο MVC εξελίχθηκε στην πορεία του χρόνου, δίνοντας έτσι την αφετηρία για τη δημιουργία διαφόρων παραλλαγών του, όπως το ιεραρχικό μοντέλο-προβολή-ελεγκτή (HMVC), το μοντέλο-προβολή-προσαρμογέα (MVA), το μοντέλο-προβολή-παρουσιαστή (MVP), το μοντέλο-προβολή-εμφάνιση μοντέλου (MVVM) και άλλα που προσαρμόζουν το MVC σε διάφορα περιβάλλοντα.

Αυτές οι παραλλαγές του MVC ενσωματώνουν και επεκτείνουν τις αρχές του αρχικού σχεδιαστικού μοντέλου, προσαρμόζοντας τα στοιχεία και τη λειτουργία τους για να ανταποκριθούν σε ειδικές ανάγκες και προκλήσεις.

2.9.3 Τα στοιχεία του MVC

Το Model-View-Controller (MVC), είναι ένα αρχιτεκτονικό πρότυπο που χρησιμοποιείται στην ανάπτυξη λογισμικού για τη διάσπαση μιας εφαρμογής σε τρία κύρια στοιχεία: το Model, το View και τον Controller.

1. **To Model** αποτελεί τη βάση και τον κορμό της εφαρμογής, όπου διαχειρίζεται τα δεδομένα και την επιχειρηματική λογική. Αυτό περιλαμβάνει την αναπαράσταση των δεδομένων, την επεξεργασία τους, καθώς και τους επιχειρηματικούς κανόνες που πρέπει να ακολουθούνται για την εύρυθμη λειτουργία της εφαρμογής. Το Model είναι υπεύθυνο για την αλληλεπίδραση με τη βάση δεδομένων, την αποθήκευση και την ανάκτηση των δεδομένων, καθιστώντας το ζωτικής σημασίας για την οργάνωση και τη διαχείριση των πληροφοριών.
2. **To View** αποτελεί το επίπεδο παρουσίασης και αλληλεπίδρασης με τον χρήστη. Είναι υπεύθυνο για την εμφάνιση των δεδομένων που παρέχει το Model στον χρήστη και την παροχή μιας διεπαφής για την εισαγωγή δεδομένων από τον χρήστη. Το View πρέπει να είναι όσο το δυνατόν πιο απλό και να μην περιέχει επιχειρηματική λογική, παρά μόνο την αναγκαία λογική για την

παρουσίαση των δεδομένων, επιτρέποντας έτσι στον χρήστη να επικοινωνήσει αποτελεσματικά με την εφαρμογή.

3. **O Controller** λειτουργεί ως διαμεσολαβητής μεταξύ του Model και του View, διασφαλίζοντας την ομαλή επικοινωνία και αλληλεπίδραση μεταξύ τους. Λαμβάνει την είσοδο από τον χρήστη μέσω του View, ενημερώνει το Model με βάση αυτή την είσοδο, και στη συνέχεια επιστρέφει τα αποτελέσματα στο View για να παρουσιαστούν στον χρήστη. Ο Controller είναι υπεύθυνος για τη διαχείριση της ροής δεδομένων και την εκτέλεση της λογικής της εφαρμογής, διασφαλίζοντας την ομαλή και άρτια λειτουργία της.

2.9.4 Πλεονεκτήματα – Μειονεκτήματα

Τα υπέρ της προσέγγισης MVC περιλαμβάνουν την αποτελεσματική διαχείριση των δεδομένων, την ευκολία στην ενημέρωση και τη συντήρηση της εφαρμογής και τη δυνατότητα για πιο αποδοτική δοκιμή. Ωστόσο, η προσέγγιση MVC μπορεί να είναι περίπλοκη για μικρές εφαρμογές και μπορεί να οδηγήσει σε αύξηση του χρόνου ανάπτυξης.

Το MVC μπορεί να βελτιώσει μια web εφαρμογή από την άποψη της δομής και της οργάνωσης. Για παράδειγμα, μια web εφαρμογή που χρησιμοποιεί το MVC μπορεί να διαχειριστεί πιο αποτελεσματικά τα δεδομένα των χρηστών, να παρέχει μια καθαρή και ευανάγνωστη διεπαφή και να είναι πιο εύκολη στη συντήρηση και την ενημέρωση.

Το CodeIgniter είναι ένα PHP framework που χρησιμοποιεί το πρότυπο MVC. Με τη χρήση του MVC, το CodeIgniter επιτρέπει στους προγραμματιστές να οργανώνουν τον κώδικά τους με έναν καθαρό και ευανάγνωστο τρόπο, καθιστώντας το πιο εύχρηστο και ανθεκτικό σε σφάλματα.

Συνοπτικά τα κύρια οφέλη της χρήσης MVC αρχιτεκτονικής:

- Προσφέρει σαφή διαχωρισμό των ευθυνών (SoC).
- Εύκολη συντήρηση του κώδικα που είναι εύκολο να επεκταθεί και να αναπτυχθεί.
- Κάθε κλάση και αντικείμενο είναι αυτόνομο, πράγμα που επιτρέπει τον ευκολότερο έλεγχο του κώδικα.
- Δημιουργεί το πλαίσιο για επαναχρησιμοποίηση κώδικα, για παράδειγμα πολλοί Controllers μπορούν να χρησιμοποιήσουν το ίδιο Model.
- Η ανάπτυξη των διαφόρων συστατικών μπορεί να γίνει παράλληλα.
- Βοηθά την αποφυγή της πολυπλοκότητας διαιρώντας μια εφαρμογή στις τρεις ενότητες: Μοντέλο, Προβολή και Ελεγκτής.
- Είναι αποτελεσματικό για τις εφαρμογές Ιστού που υποστηρίζονται από εκτεταμένες ομάδες σχεδιαστών και προγραμματιστών ιστού.
- Επιτρέπει τη χρήση διαφορετικών τεχνολογιών ανά επίπεδο.

Κάποια μειονεκτήματα που μπορεί να παρατηρηθούν είναι:

- Το MVC νέα επίπεδα αφαίρεσης που απαιτούν από τους χρήστες να προσαρμοστούν στα κριτήρια αποσύνθεσης του MVC.
- Υπάρχει ανάγκη για πολλούς προγραμματιστές να διεξάγουν παράλληλο προγραμματισμό.
- Απαιτείται γνώση πολλαπλών τεχνολογιών.

2.9.5 Εναλλακτικές αρχιτεκτονικές

Υπάρχουν πολλές εναλλακτικές για το MVC στην ανάπτυξη ιστοσελίδων. Κάποιες από αυτές περιλαμβάνουν το MVP (Model-View-Presenter), το MVVM (Model-View-ViewModel) και το MVA (Model-View-Adapter).

Το MVP μοιάζει με το MVC, αλλά ο Presenter αναλαμβάνει τη θέση του Controller και επικοινωνεί απευθείας με το View. Αυτό καθιστά τον Presenter πιο προσαρμόσιμο, καθώς μπορεί να χειριστεί πολλά Views ταυτόχρονα.

Το MVVM, από την άλλη πλευρά, επιτρέπει στο View να επικοινωνεί απευθείας με το Model. Αυτό επιτρέπει τη δημιουργία πιο δυναμικών και αποκρίσιμων διεπαφών χρήστη.

Τέλος, το MVA χρησιμοποιεί έναν Adapter για την επικοινωνία μεταξύ του View και του Model. Αυτό του επιτρέπει να διαχειριστεί πιο πολύπλοκες καταστάσεις και μπορεί να κάνει τον κώδικα πιο οργανωμένο και ευανάγνωστο.

Κάθε μία από αυτές τις αρχιτεκτονικές έχει τα δικά της πλεονεκτήματα και μειονεκτήματα, και η επιλογή της κατάλληλης εξαρτάται από την πολυπλοκότητα της εφαρμογής, τις απαιτήσεις των χρηστών, την εμπειρία της ομάδας ανάπτυξης και τις προτιμήσεις της.

Έχοντας υπόψη τα παραπάνω, είναι σημαντικό για τους προγραμματιστές να επιλέγουν το κατάλληλο σχεδιαστικό μοντέλο βάσει των αναγκών και των προδιαγραφών της κάθε εφαρμογής. Το MVC αποτελεί μια πολύ καλή επιλογή για ορισμένους τύπους εφαρμογών, αλλά ίσως να μην είναι ο πιο κατάλληλος για άλλους.

Όπως και με κάθε εργαλείο, είναι σημαντικό να γνωρίζουμε πώς και πότε να το χρησιμοποιήσουμε κατάλληλα, ώστε να μπορούμε να αξιοποιήσουμε πλήρως τα πλεονεκτήματα που προσφέρει. Είναι επίσης σημαντικό να γνωρίζουμε ότι τα σχεδιαστικά μοντέλα, όπως το MVC, δεν πρέπει να θεωρούνται αυστηροί κανόνες, αλλά μάλλον γενικές οδηγίες. Ανάλογα με τις απαιτήσεις της εφαρμογής, μπορεί να είναι αποδεκτό ή ακόμα και επιθυμητό να παρεκκλίνουμε από το πρότυπο MVC και να προσαρμόσουμε τον κώδικά μας ανάλογα.

Η εκμάθηση και η κατανόηση των βασικών αρχών του MVC, καθώς και των πιθανών προκλήσεων που μπορεί να προκύψουν, μπορούν να βοηθήσουν τους προγραμματιστές να χρησιμοποιήσουν αυτό το μοντέλο σχεδίασης με τον πιο αποτελεσματικό τρόπο. Παράλληλα, η διαρκής ενημέρωση σχετικά με τις νέες πρακτικές και τις προσεγγίσεις στον τομέα της ανάπτυξης λογισμικού μπορεί να βοηθήσει στην προσαρμογή και την εξέλιξη της χρήσης του MVC, ανάλογα με τις ανάγκες κάθε προγραμματιστή και κάθε εφαρμογής.

Εν κατακλείδι, το MVC είναι ένα ισχυρό εργαλείο στην ανάπτυξη λογισμικού, αλλά όπως και όλα τα εργαλεία, πρέπει να χρησιμοποιείται με σύνεση και κριτική σκέψη.

2.10 Επίλογος

Σε αυτό το κεφάλαιο, παρουσιάστηκαν συνοπτικά τα κύρια εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής σε αυτή την εργασία. Αυτά περιλαμβάνουν τη γλώσσα προγραμματισμού PHP, τη βάση δεδομένων MySQL, καθώς και διάφορες βιβλιοθήκες και εργαλεία ανάπτυξης εφαρμογών. Έγινε επίσης μία παρουσίαση του μοντέλου MVC.

Ωστόσο, υπάρχει ένα σημαντικό εργαλείο που παραλήφθηκε σκόπιμα στο παρόν κεφάλαιο. Αυτό είναι τα PHP frameworks, η χρησιμότητα και η σημασία των οποίων θα εξεταστούν λεπτομερώς στο επακόλουθο κεφάλαιο. Συγκεκριμένα, θα επικεντρωθούμε στο Codeigniter, το οποίο επιλέξαμε ως το κύριο framework για την ανάπτυξη της εφαρμογής μας.

Τα PHP frameworks παρέχουν μια βασική δομή για την ανάπτυξη αποδοτικών, ασφαλών και βιώσιμων εφαρμογών ιστού, επιτρέποντας στους προγραμματιστές να αποφύγουν την επανειλημμένη συγγραφή του ίδιου κώδικα και να επικεντρωθούν στη λειτουργικότητα της εφαρμογής.

Συνεχίζοντας στο επόμενο κεφάλαιο, θα διερευνηθεί περαιτέρω η επιλογή για το Codeigniter, καθώς και τους λόγους που το καθιστούν ένα εξαιρετικό εργαλείο για την ανάπτυξη της εφαρμογής της παρούσας εργασίας.

Κεφάλαιο 3ο: Frameworks – Codeigniter

3.1 Εισαγωγή

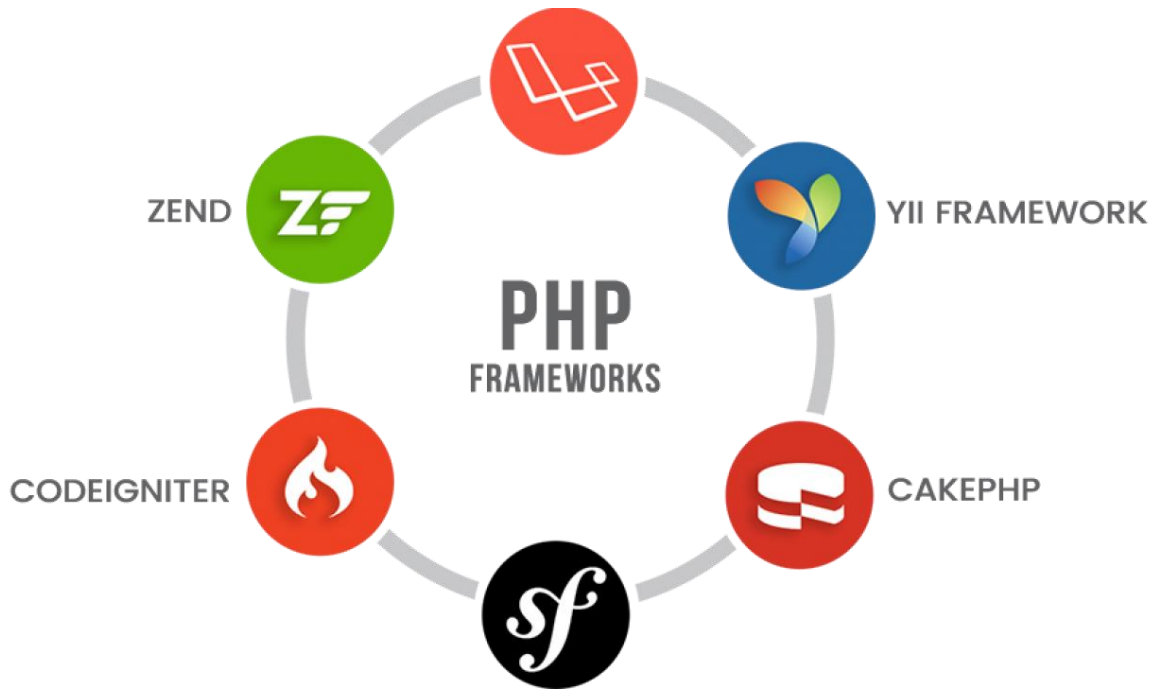
Ένα PHP framework αποτελεί μια πλατφόρμα, σχεδιασμένη με σκοπό τη δημιουργία και την ανάπτυξη εφαρμογών ιστού που χρησιμοποιούν τη γλώσσα προγραμματισμού PHP. Οι πλατφόρμες αυτές, εκτός από την υποστήριξη υψηλού επιπέδου, προσφέρουν επίσης βιβλιοθήκες κώδικα για κοινές λειτουργίες, καθιστώντας την επαναχρησιμοποίηση κώδικα ευκολότερη και πιο αποτελεσματική. Ο στόχος τους είναι να ελαχιστοποιήσουν την ποσότητα του αρχικού κώδικα που θα πρέπει να γράψει ο προγραμματιστής, μειώνοντας έτσι το χρόνο που απαιτείται για την ανάπτυξη της εφαρμογής. Αυτό συμβάλλει στην αυξημένη αποτελεσματικότητα της διαδικασίας της σχεδίασης της εφαρμογής και των λειτουργικών απαιτήσεών της, καθώς και του προγραμματισμού αυτών. Αυτό καθιστά την όλη διαδικασία λιγότερο χρονοβόρα και πιο οικονομικά αποδοτική. Επιπλέον, παρέχει ένα σταθερό και αξιόπιστο πλαίσιο για τη δημιουργία και τη υλοποίηση των εφαρμογών, καθιστώντας την ολόκληρη διαδικασία πιο ομαλή και διαχειρίσιμη.

Συνοψίζοντας, παρακάτω είναι οι βασικοί λόγοι, για τη χρήση PHP framework:

- Ταχύτητα υλοποίησης
- Ελαχιστοποίηση κώδικα που πρέπει να γραφτεί και να συντηρηθεί
- Έτοιμες λύσεις για κοινές λειτουργικότητες
- Καλύτερη δόμηση των αρχείων του κώδικα
- Ασφάλεια
- Έλεγχος κώδικα (testing)
- Ευκολότερη συνεργασία σε μία ομάδα ανάπτυξης

Ένα από τα βασικά αρνητικά στοιχεία που πρέπει να ληφθούν υπόψη όταν αποφασίζεται η χρήση ενός framework, είναι η καμπύλη εκμάθησης που συνεπάγεται. Σε ορισμένες περιπτώσεις, αυτό μπορεί να είναι σαν να μαθαίνεις μία νέα γλώσσα προγραμματισμού από την αρχή, κάτι που απαιτεί χρόνο και προσπάθεια. Επίσης, κάτι άλλο που μπορεί να θεωρηθεί αρνητικό είναι ότι ένα framework, παρέχοντας έτοιμες λύσεις και μια δομή που πρέπει να ακολουθηθεί, μπορεί να περιορίσει τον προγραμματιστή στις δημιουργικές του επιλογές. Αυτό μπορεί να είναι προβληματικό εάν ο προγραμματιστής επιθυμεί να εξερευνήσει νέες προσεγγίσεις ή να αναπτύξει πρωτότυπες λειτουργίες.

Στην παρακάτω ενότητα, παρουσιάζονται συνοπτικά κάποια από τα πιο δημοφιλή PHP frameworks που χρησιμοποιούνται ευρέως σήμερα στον κόσμο της ανάπτυξης web εφαρμογών λογισμικού.



Εικόνα 3.1 PHP Frameworks

3.2 Γνωστά PHP Frameworks

Η ανάπτυξη εφαρμογών ιστού είναι μια διαδικασία που απαιτεί προσεκτικό σχεδιασμό, προγραμματισμό και έλεγχο κώδικα. Για να διευκολύνουν αυτήν τη διαδικασία, οι προγραμματιστές συχνά επιλέγουν ένα από τα δωρεάν διαθέσιμα frameworks. Αυτά τα εργαλεία προσφέρουν μια σειρά από λειτουργίες και βιβλιοθήκες που διευκολύνουν τη δημιουργία, τη δοκιμή και τη συντήρηση των εφαρμογών. Στην παρακάτω λίστα, θα παρουσιαστούν κάποια από τα πιο γνωστά frameworks της PHP.

- Το Laravel είναι ένα δημοφιλές, δωρεάν PHP framework με εκλεπτυσμένη σύνταξη και ισχυρά εργαλεία, ιδανικό για την ανάπτυξη ισχυρών εφαρμογών ιστού. Διαθέτει μια ενεργή κοινότητα που παρέχει πολυάριθμους πόρους και εκπαιδευτικό υλικό.
- Το CakePHP είναι ένα ανοιχτού κώδικα PHP framework που ακολουθεί την αρχιτεκτονική MVC. Προσφέρει πολλά χαρακτηριστικά και είναι ιδανικό για την γρήγορη ανάπτυξη εφαρμογών στην γλώσσα προγραμματισμού PHP.
- Το Symfony είναι ένα ευέλικτο framework ανάπτυξης εφαρμογών ιστού, ιδανικό για μεγάλης κλίμακας επιχειρηματικά έργα. Η ενεργή κοινότητα του Symfony παρέχει πόρους και υποστήριξη προγραμματιστών.
- Το Yii είναι ένα PHP framework υψηλής απόδοσης, σχεδιασμένο ειδικά για τη δημιουργία σύγχρονων, ευέλικτων και λειτουργικών εφαρμογών ιστού. Είναι εξοπλισμένο με πληθώρα εργαλείων και βιβλιοθηκών που καθιστούν την ανάπτυξη σύνθετων εφαρμογών πολύ πιο εύκολη και αποτελεσματική.
- Το Laminas, μία επέκταση του Zend, είναι ένα MVC framework με ανεξάρτητα συστατικά. Παρέχει εξαιρετικές λύσεις για middleware, βελτιώνοντας την απόδοση και την ασφάλεια των εφαρμογών.

3.3 Codeigniter

3.3.1 Εισαγωγή

Σε αυτή την ενότητα, θα γίνει μια πιο αναλυτική και εκτεταμένη περιγραφή του framework Codeigniter. Αυτό έχει επιλεγεί ως το κύριο πλαίσιο για την ανάπτυξη της εφαρμογής που αναλύεται στην παρούσα εργασία. Το Codeigniter είναι ένα από τα πιο δημοφιλή frameworks και χρησιμοποιείται ευρέως για την ανάπτυξη web εφαρμογών. Είναι γνωστό για την απλότητα του, την ευελιξία του και τη δυνατότητα που προσφέρει στους προγραμματιστές να δημιουργήσουν γρήγορα και αποτελεσματικά πολύπλοκες εφαρμογές.

Το Codeigniter είναι ένα ελαφρύ PHP framework, βασισμένο στην αρχιτεκτονική MVC, που προσφέρει μια απλούστευση στην ανάπτυξη δυναμικών εφαρμογών ιστού. Με ένα σχετικά μικρό αποτύπωμα, δηλαδή, περίπου 2MB συμπεριλαμβανομένης της τεκμηρίωσης, είναι πολύ εύκολο στην εγκατάσταση και τη ρύθμιση. Αυτό το καθιστά μια εξαιρετική επιλογή για μικρά έως μεσαία έργα, καθώς μπορεί να χειριστεί την ανάπτυξη εφαρμογών με λιγότερο κώδικα και μικρότερο χρόνο. Το έργο του Codeigniter υποστηρίζεται από μια ενεργή και αφοσιωμένη κοινότητα προγραμματιστών που διαρκώς αναπτύσσει και ενημερώνει την πλατφόρμα. Επιπρόσθετα, η κοινότητα, προσφέρει πολλούς πόρους εκπαιδευτικών υλικών, παραδειγμάτων και οδηγιών, τα οποία είναι διαθέσιμα για να βοηθήσουν τους νέους χρήστες να ξεκινήσουν και να εξοικειωθούν με την πλατφόρμα.

Το Codeigniter είναι ένα αποτελεσματικό πλαίσιο που παρέχει ένα φιλικό προς τον χρήστη περιβάλλον, το οποίο είναι σχεδιασμένο για να κάνει την ανάπτυξη εφαρμογών ιστού ένα απλό και γρήγορο εγχείρημα. Εκτός από αυτό, προσφέρει επίσης ένα σετ εργαλείων και βιβλιοθηκών που έχουν σχεδιαστεί για να αυξήσουν την αποτελεσματικότητα και την παραγωγικότητα των προγραμματιστών. Το Codeigniter είναι επίσης γνωστό για τη σταθερότητα και την ασφάλεια του. Έχει ενσωματωμένη προστασία από διάφορες επιθέσεις, όπως η εισαγωγή SQL, η επίθεση XSS και η επίθεση CSRF, καθιστώντας το ένα ασφαλές πλαίσιο για την ανάπτυξη εφαρμογών.

3.3.2 Ιστορικό

Το framework δημοσιεύτηκε για πρώτη φορά στην αρχική του έκδοση το 2006, από την εταιρεία που το δημιούργησε, την EllisLab [11]. Αυτή ήταν η αρχή ενός εργαλείου που θα επηρεάσει σημαντικά τον τομέα της ανάπτυξης λογισμικού. Από τότε, ο κώδικάς του έχει δημοσιευτεί στο github, γνωστή πλατφόρμα για την ανάπτυξη λογισμικού, κάτω από την άδεια χρήσης MIT License, επιτρέποντας έτσι στην κοινότητα των προγραμματιστών να συνεισφέρει στην εξέλιξή του. Στο ενδιάμεσο, το framework έχει δει πολλές αλλαγές και εξελίξεις, με την τέταρτη μεγάλη έκδοση του να κυκλοφορεί το 2020 [12]. Αυτή η έκδοση, με τις πλέον εξελιγμένες της δυνατότητες, ήταν αυτή που χρησιμοποιήθηκε κατά τη διάρκεια της παρούσας εργασίας.

3.3.3 Δομή φακέλων

Για την ανάπτυξη μιας εφαρμογής με τη χρήση του Codeigniter 4, ένα από τα πιο σημαντικά στοιχεία είναι η πλήρης κατανόηση της δομής των αρχείων και των φακέλων της εφαρμογής. Αυτό σημαίνει ότι ο προγραμματιστής θα πρέπει να γνωρίζει πώς είναι οργανωμένη η εφαρμογή, ποια αρχεία και φάκελοι

υπάρχουν και πώς συνδέονται μεταξύ τους. Αυτή η κατανόηση θα επιτρέψει στον προγραμματιστή να λάβει αποφάσεις σχετικά με το πού και πώς θα κάνει αλλαγές, πού θα προσθέσει νέο κώδικα και πού θα ενσωματώσει την επιχειρησιακή λογική. Κατ' αυτόν τον τρόπο, ο προγραμματιστής θα μπορεί να αξιοποιήσει πλήρως το πλαίσιο εργασίας Codeigniter 4 και να δημιουργήσει πιο αποτελεσματικά και λειτουργικά προγράμματα.

Η βασική δομή περιλαμβάνει τους παρακάτω φακέλους:

- app
- public
- writable
- test
- system

Από τους παραπάνω, αυτός που ενδιαφέρει περισσότερο τον προγραμματιστή, από σκοπιάς συγγραφής κώδικα, είναι ο φάκελος app. Στον εν λόγω φάκελο βρίσκεται όλος ο κώδικας που αποτελεί την εξατομικευμένη λειτουργικότητα της εκάστοτε εφαρμογής. Αυτό σημαίνει ότι, οποιαδήποτε τροποποίηση, προσθήκη ή αφαίρεση λειτουργιών που θέλει να κάνει ο προγραμματιστής θα πρέπει να γίνει εντός αυτού του φακέλου. Επίσης, είναι ο φάκελος που περιέχει τα αρχεία που ορίζουν την παρουσίαση και τη λειτουργία της εφαρμογής στον τελικό χρήστη.

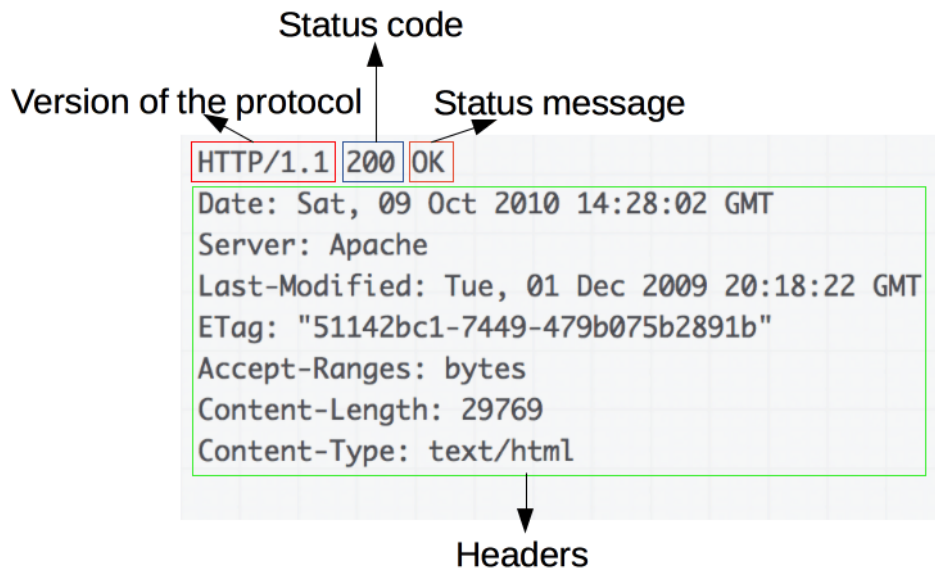
Ο φάκελος system περιλαμβάνει όλον τον κώδικα που αποτελεί το ίδιο το framework. Αυτό που βρίσκεται εντός αυτού του φακέλου δεν έχει σχεδιαστεί για να τροποποιηθεί από τον προγραμματιστή, καθώς ο στόχος είναι να παραμείνει αμετάβλητο. Αντ' αυτού, ισχύει η αρχή σχεδίασης Open-Closed [13], μια αρχή που υποστηρίζει ότι οι κλάσεις που βρίσκονται στο φάκελο system μπορούν να επεκταθούν σε μία νέα υλοποίηση (π.χ., μέσα στο φάκελο app), αλλά παραμένουν κλειστές σε τροποποιήσεις. Αυτό σημαίνει ότι το framework καθαυτό μπορεί να προσαρμοστεί στις ανάγκες του προγράμματος, ενώ παράλληλα διατηρεί την ακεραιότητα και τη σταθερότητα του.

Το σημείο δημόσιας πρόσβασης για έναν περιηγητή (browser), βρίσκεται συγκεντρωμένο στον φάκελο public. Αυτός ο φάκελος αποτελεί τη ρίζα του ιστότοπου, δηλαδή το σημείο στο οποίο πρέπει να δείχνει ο εξυπηρέτης για να επιτευχθεί η σωστή λειτουργία του ιστότοπου. Εντός αυτού του φακέλου, φιλοξενούνται διάφορα σημαντικά στοιχεία της εφαρμογής, όπως βιβλιοθήκες javascript και css, που είναι απαραίτητες για την εμφάνιση και τη λειτουργία του ιστότοπου. Επίσης, περιλαμβάνει το αρχείο index.php, το οποίο υλοποιεί το πρότυπο σχεδίασης Front Controller [14], το οποίο είναι θεμελιώδες για την οργάνωση και την ασφάλεια της εφαρμογής.

Κατά τη διάρκεια του κύκλου εκτέλεσης της εφαρμογής, είναι ιδιαίτερα επιθυμητό να υπάρχει ένα συγκεκριμένο σημείο που θα επιτρέψει την καταγραφή διαφόρων συμβάντων. Είναι απαραίτητο να έχουμε ένα τέτοιο σημείο για λόγους οργάνωσης και ευκολίας. Επιπλέον, στο ίδιο ακριβώς σημείο θα πραγματοποιείται η εγγραφή προσωρινών αρχείων ή ακόμη και αρχείων που θα ανεβάσει ο τελικός χρήστης. Αυτό είναι πολύ σημαντικό για την αποτελεσματική λειτουργία της εφαρμογής. Γενικά, αυτή είναι η μοναδική τοποθεσία όπου επιτρέπεται η εγγραφή κατά τη διάρκεια της εκτέλεσης της εφαρμογής. Αυτό το σημείο αποτελείται από τον φάκελο που ονομάζεται writable.

Τέλος, έχουμε τον φάκελο tests, ο οποίος είναι ζωτικής σημασίας κατά τις φάσεις ανάπτυξης και συντήρησης της εφαρμογής. Αυτός ο φάκελος δεν πρόκειται να συμπεριληφθεί στην τελική δημοσίευση

μήνυμα απόκρισης πίσω στον πελάτη. Το μήνυμα απόκρισης έχει μια δομή που είναι πολύ παρόμοια με τη δομή του αρχικό αίτημα, όπως μπορεί να φανεί καθαρά στην *Εικόνα 3.3*.



Εικόνα 3.3 Δομή απόκρισης HTTP [15]

Η απόκριση περιλαμβάνει την έκδοση του πρωτοκόλλου HTTP που χρησιμοποιείται, έναν κωδικό κατάστασης που δηλώνει αν το αίτημα ήταν επιτυχές ή όχι, και ένα κείμενο κατάστασης που παρέχει περισσότερες λεπτομέρειες σχετικά με την κατάσταση της απόκρισης. Επιπλέον, οι κεφαλίδες παρέχουν συνοδευτικές πληροφορίες, όπως τον τύπο των περιεχομένων και την κωδικοποίηση της απόκρισης, που είναι ζωτικής σημασίας για την κατανόηση και την ερμηνεία των δεδομένων. Τέλος, εάν ο πελάτης έχει αιτηθεί συγκεκριμένα δεδομένα, τα δεδομένα αυτά θα περιλαμβάνονται επίσης στο σώμα της απόκρισης, ανάλογα με το αίτημα.

Αν και η γλώσσα προγραμματισμού PHP διαθέτει ενσωματωμένους μηχανισμούς για τη διαχείριση μηνυμάτων HTTP, το Codeigniter, ως ένα πλαίσιο λογισμικού (framework), παρέχει μια αφαιρετική υλοποίηση για τη διαχείριση των μηνυμάτων αυτών. Αυτό συμβάλλει στην παροχή μιας απλοποιημένης και ενιαίας διεπαφής για τους προγραμματιστές, καθιστώντας τη διαδικασία διαχείρισης των HTTP μηνυμάτων πιο οργανωμένη και λιγότερο πολύπλοκη.

Η κλάση `IncomingRequest` αποτελεί μία αντικειμενοστραφή αποτύπωση του αιτήματος HTTP. Παρέχει στον προγραμματιστή όλες τις απαραίτητες μεθόδους για να αντλήσει τις πληροφορίες που χρειάζεται από το αίτημα, όπως φαίνεται στο παρακάτω παράδειγμα κώδικα *Εικόνα 3.4*.

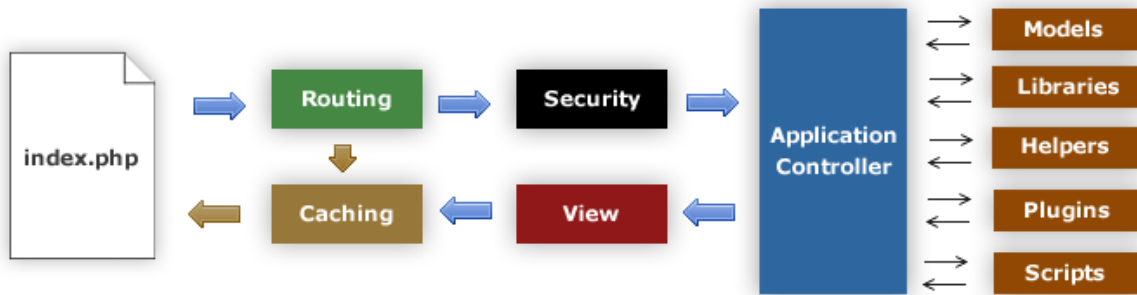
```
1  <?php
2
3  use CodeIgniter\HTTP\IncomingRequest;
4
5  $request = request();
6
7  // the URI path being requested (i.e., /about)
8  $request->getUri()->getPath();
9
10 // Retrieve $_GET and $_POST variables
11 $request->getGet('foo');
```

Εικόνα 3.4 Μέθοδοι της κλάσης IncomingRequest [16]

Αντίστοιχα για το μήνυμα απόκρισης προς τον πελάτη, το Codeigniter, παρέχει μια αντικειμενοστραφή αποτύπωση της δομής του μηνύματος απόκρισης, μέσω της κλάσης Response. Αυτό προσφέρει μια συνεκτική και απλή διεπαφή, επομένως επιτρέπει στον προγραμματιστή να χτίζει δυναμικά την απάντηση στο αίτημα του πελάτη, προσφέροντας έτσι μεγάλη ευελιξία και ευκολία στη διαχείριση των αιτήσεων, όπως φαίνεται στην *Εικόνα 3.5*.

```
1  <?php
2
3  use CodeIgniter\HTTP\Response;
4
5  $response = response();
6
7  $response->setStatusCode(Response::HTTP_OK);
8  $response->setBody($output);
9  $response->setHeader('Content-Type', 'text/html');
10 $response->noCache();
11
12 // Sends the output to the browser
13 // This is typically handled by the framework
14 $response->send();
```

Εικόνα 3.5 Μέθοδοι της κλάσης Response [16]



Εικόνα 3.6 Διάγραμμα ροής Codeigniter [17]

3.3.5 Ροή εφαρμογής Codeigniter

Η ροή εκτέλεσης της εφαρμογής περιγράφεται στα παρακάτω βήματα:

1. **Αίτημα Χρήστη (HTTP Request):** Ο χρήστης ξεκινά ένα αίτημα HTTP, συνήθως υποβάλλοντας δεδομένα μέσω μιας φόρμας ή κάνοντας κλικ σε έναν σύνδεσμο.
2. **Front Controller (index.php):** Το αρχείο index.php λειτουργεί ως το σημείο εισόδου για όλα τα εισερχόμενα αιτήματα. Αρχικοποιεί το πλαίσιο, θέτει τους κανόνες δρομολόγησης και φορτώνει τον κατάλληλο ελεγκτή.
3. **Δρομολόγηση:** Η δρομολόγηση αναλύει το αίτημα URL και το ταιριάζει με τις ορισμένες διαδρομές. Καθορίζει ποια μέθοδος ελεγκτή πρέπει να εκτελεστεί βάσει της ζητούμενης URI.
4. **Middleware (προαιρετικό):** Το Middleware μπορεί να εφαρμοστεί για να διακόψει αιτήματα στον ελεγκτή και να εκτελέσει πρόσθετες εργασίες, όπως η ταυτοποίηση, η εξουσιοδότηση ή η καταγραφή.
5. **Μέθοδος Ελεγκτή:** Καλείται η μέθοδος του ελεγκτή, η οποία είναι υπεύθυνη για τη διαχείριση του συγκεκριμένου αιτήματος. Μπορεί να περιλαμβάνει την αλληλεπίδραση με τα μοντέλα, την εκτέλεση της επιχειρησιακής λογικής και την ανάκτηση ή την επεξεργασία δεδομένων.
6. **Μοντέλο (προαιρετικό):** Εάν απαιτείται πρόσβαση σε δεδομένα, ο ελεγκτής μπορεί να φορτώσει και να εκτελέσει μεθόδους μοντέλου για να αλληλοεπιδράσει με τη βάση δεδομένων. Τα μοντέλα χειρίζονται την διατήρηση και την ανάκτηση δεδομένων.
7. **Επεξεργασία και Επικύρωση Δεδομένων:** Ο ελεγκτής επεξεργάζεται και επικυρώνει τα δεδομένα που λαμβάνει από το αίτημα, εξασφαλίζοντας την ακρίβειά τους και τη συνέπειά τους.
8. **Προβολή (προαιρετική):** Ο ελεγκτής μπορεί να χρησιμοποιήσει τις προβολές για την απεικόνιση του επιπέδου παρουσίας της εφαρμογής. Οι προβολές περιέχουν πρότυπα HTML και δυναμικά δεδομένα που περνούν από τον ελεγκτή.
9. **Απεικόνιση και Απάντηση:** Ο ελεγκτής συναρμολογεί την τελική απάντηση, συμπεριλαμβανομένης της απεικονισμένης προβολής (εάν εφαρμόζεται) και οποιωνδήποτε απαραίτητων κεφαλίδων ή οδηγιών ανακατεύθυνσης.
10. **Αποστολή Απάντησης στον Χρήστη:** Η απάντηση αποστέλλεται πίσω στον περιηγητή του χρήστη, εμφανίζοντας το ζητούμενο περιεχόμενο και ολοκληρώνοντας την ροή της εφαρμογής.

3.3.6 Εγκατάσταση

Για την εγκατάσταση του Codeigniter 4 υπάρχουν δύο βασικοί τρόποι που μπορεί κάποιος να ακολουθήσει. Αυτοί οι τρόποι είναι σχεδιασμένοι έτσι ώστε να καλύπτουν τις διάφορες ανάγκες και προτιμήσεις των χρηστών, ανάλογα με το περιβάλλον ανάπτυξης και την εμπειρία τους στη χρήση τέτοιων εργαλείων.

1. **Εγκατάσταση μέσω Composer (αναφέρθηκε σε προηγούμενο κεφάλαιο):** Αυτή είναι και η προτεινόμενη μέθοδος εγκατάστασης. Πρώτα, θα πρέπει να εγκαταστήσετε το Composer, το οποίο είναι ένα εργαλείο διαχείρισης εξαρτήσεων για την PHP. Μετά την εγκατάσταση του Composer, μπορείτε να εγκαταστήσετε το Codeigniter 4 εκτελώντας την παρακάτω εντολή στο τερματικό: `composer create-project codeigniter4/appstarter project-root`
2. **Μη αυτόματη εγκατάσταση:** Αν δεν θέλετε ή δεν μπορείτε να χρησιμοποιήσετε το Composer, μπορείτε να κατεβάσετε το Codeigniter 4 απευθείας από την ιστοσελίδα του. Απλώς κατεβάστε το πακέτο, αποσυμπιέστε το και αντιγράψτε τα αρχεία στο ριζικό φάκελο του έργου σας.

Μετά την εγκατάσταση του Codeigniter 4, μπορεί να ξεκινήσει άμεσα η ανάπτυξη της εφαρμογής. Το περιβάλλον ανάπτυξης του Codeigniter είναι έτοιμο για χρήση εξ' ορισμού και μπορείτε να ξεκινήσετε να δημιουργείτε τους controllers, τα models και τις views της εφαρμογής σας.

Είναι σημαντικό να θυμάστε ότι το Codeigniter είναι ένα MVC (Model-View-Controller) framework. Αυτό σημαίνει ότι η λογική της εφαρμογής σας θα πρέπει να χωρίζεται σε τρεις βασικές ενότητες: τα Models, τα Views και τα Controllers. Οι ενότητες αυτές εργάζονται από κοινού για να επιτρέπουν την εκτέλεση των ενεργειών της εφαρμογής σας και την παρουσίαση των δεδομένων στο χρήστη.

Αν παρουσιαστεί οποιαδήποτε δυσκολία κατά την εγκατάσταση ή την ανάπτυξη της εφαρμογής με το Codeigniter 4, μπορεί κάποιος να ανατρέξει στον επίσημο οδηγό χρήσης του Codeigniter ή σε διάφορες κοινότητες και φόρουμ στο διαδίκτυο που μπορούν να παράσχουν βοήθεια και υποστήριξη.

3.3.7 Ρύθμιση

Μετά την εγκατάσταση του Codeigniter 4, πρέπει να γίνουν ορισμένες βασικές ρυθμίσεις για να εξασφαλιστεί η σωστή λειτουργία της εφαρμογής. Αυτές οι ρυθμίσεις περιλαμβάνουν:

- **Ρύθμιση της Βάσης Δεδομένων:** Καθορισμό των λεπτομερειών σύνδεσης της βάσης δεδομένων στο αρχείο ρυθμίσεων, περιλαμβάνοντας το hostname, το username, το password, και το όνομα της βάσης δεδομένων.
- **Ρύθμιση του βασικού URL:** Θέτετε το βασικό URL της εφαρμογής σας. Αυτό είναι σημαντικό για την διαχείριση των συνδέσεων και της πλοήγησης στην εφαρμογή.
- **Ρύθμιση της Ζώνης Ώρας:** Ορίζετε την προεπιλεγμένη ζώνη ώρας για την εφαρμογή σας. Αυτό είναι σημαντικό για την επεξεργασία και την αποθήκευση των χρονικών δεδομένων.
- **Ρύθμιση της Αποσφαλμάτωσης:** Ενεργοποιείτε ή απενεργοποιείτε την αποσφαλμάτωση. Όταν η αποσφαλμάτωση είναι ενεργοποιημένη, τα λάθη και οι προειδοποιήσεις θα

εμφανίζονται στην οθόνη. Αυτό είναι χρήσιμο κατά την ανάπτυξη, αλλά πρέπει να απενεργοποιηθεί στο περιβάλλον παραγωγής.

Προτείνεται να ελέγξετε τον επίσημο οδηγό χρήσης του Codeigniter για αναλυτικές οδηγίες σχετικά με αυτές και άλλες ρυθμίσεις [18].

3.3.8 Ασφάλεια

Το Codeigniter, στο πλαίσιο της ανάπτυξης εφαρμογών για την εξασφάλιση της ασφάλειας, ακολουθεί τις συστάσεις του οργανισμού Open Web Application Security Project (OWASP) [19], που είναι κορυφαίος στον τομέα της ασφάλειας των εφαρμογών. Πιο συγκεκριμένα, το Codeigniter παρέχει μια σειρά από μεθόδους και τεχνικές για να επιβάλει καλές πρακτικές ασφάλειας [20], ή να επιτρέψει την εφαρμογή των βέλτιστων πρακτικών ασφάλειας από τον προγραμματιστή με τον πιο απλό τρόπο. Αυτό περιλαμβάνει την παροχή κλάσεων και βιβλιοθηκών που παρέχουν προστασία από επιθέσεις Cross-Site Request Forgery (CSRF), καθώς και προστασία από SQL injection, που είναι δύο από τις πιο συνηθισμένες επιθέσεις στην ασφάλεια των εφαρμογών.

3.3.9 Διαχείριση σφαλμάτων

Ο χειρισμός σφαλμάτων στο Codeigniter 4 είναι μια σημαντική διαδικασία κατά την ανάπτυξη εφαρμογών. Οι εξαιρέσεις είναι γεγονότα που συμβαίνουν κατά την εκτέλεση ενός προγράμματος και επηρεάζουν την κανονική ροή του προγράμματος.

Στο Codeigniter 4, οι εξαιρέσεις χρησιμοποιούνται για να ειδοποιήσουν τον προγραμματιστή για σφάλματα που μπορεί να προκύψουν κατά τη διάρκεια της εκτέλεσης ενός προγράμματος. Όταν προκύπτει μια εξαίρεση, η εκτέλεση του προγράμματος σταματά και το σύστημα προωθεί το σφάλμα στον κώδικα διαχείρισης. Ανάλογα με το περιβάλλον, η διαχείριση είναι διαφορετική. Στο παραγωγικό για παράδειγμα θα επιστραφεί ένα γενικό μήνυμα σφάλματος προς τον τελικό χρήστη.

Οι εξαιρέσεις στο Codeigniter 4 χειρίζονται με τη χρήση των δηλώσεων try, catch και finally. Κατά τη διάρκεια της εκτέλεσης του κώδικα μέσα στο μπλοκ try, αν προκύψει μια εξαίρεση, η εκτέλεση του κώδικα σταματά και το σύστημα αναζητά ένα αντίστοιχο μπλοκ catch για να χειριστεί την εξαίρεση. Αν μια εξαίρεση δεν είναι χειρισμένη από κάποιο μπλοκ catch, τότε το σύστημα θα ενεργοποιήσει την προεπιλεγμένη ρουτίνα χειρισμού εξαιρέσεων.

Με την κατάλληλη χρήση των εξαιρέσεων, ο προγραμματιστής μπορεί να δημιουργήσει εφαρμογές που είναι πιο ανθεκτικές στα σφάλματα και πιο εύκολες στην αποσφαλμάτωση.

3.3.10 Καταγραφή συμβάντων

Το Codeigniter 4, όπως και πολλά άλλα frameworks, παρέχει έναν μηχανισμό καταγραφής συμβάντων. Αυτός ο μηχανισμός είναι εξαιρετικά χρήσιμος για την ανίχνευση και την αποσφαλμάτωση προβλημάτων στην εφαρμογή, καθώς καταγράφει όλα τα σημαντικά γεγονότα και τα λάθη που συμβαίνουν κατά τη διάρκεια της εκτέλεσης του κώδικα.

Οι καταγραφές του Codeigniter 4 αποθηκεύονται στον φάκελο 'writable/logs' της εφαρμογής σας. Κάθε μέρα, το Codeigniter δημιουργεί ένα νέο αρχείο καταγραφής με όνομα την ημερομηνία, π.χ., 'log-2022-02-21.php'. Ένας ακόμα ενδιαφέρον τρόπος, που είναι και χρήσιμος στο περιβάλλον ανάπτυξης, είναι ένα πρόσθετο για την εμφάνιση συμβάντων στην κονσόλα του Google Chrome browser [21].

Η διαμόρφωση του μηχανισμού καταγραφής πραγματοποιείται στο αρχείο 'app/Config/Logger.php'. Εκεί, μπορείτε να ορίσετε το επίπεδο της καταγραφής 'logThreshold'. Το επίπεδο της καταγραφής μπορεί να είναι από 0 (χωρίς καταγραφή) μέχρι 9 (καταγραφή όλων των συμβάντων και των λεπτομερειών).

Τέλος, είναι σημαντικό να γνωρίζετε ότι οι καταγραφές συμβάντων πρέπει να διαχειρίζονται με προσοχή για να μην επηρεάσουν την απόδοση και την ασφάλεια της εφαρμογής. Για παράδειγμα, δεν πρέπει να περιλαμβάνουν ευαίσθητες πληροφορίες, όπως κωδικοί πρόσβασης ή προσωπικά δεδομένα των χρηστών, και πρέπει να προστατεύονται από μη εξουσιοδοτημένη πρόσβαση. Επίσης είναι απαραίτητο να την περιορίζετε όταν είστε σε περιβάλλον παραγωγής.

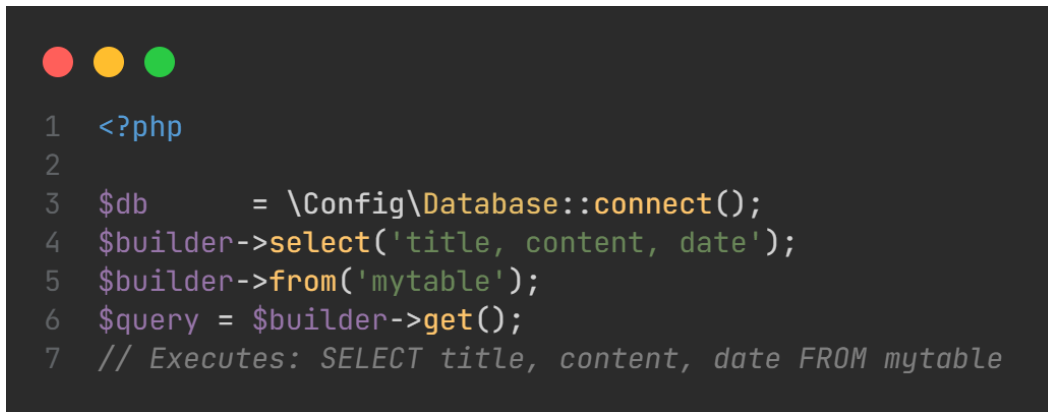
3.3.11 Κλάση QueryBuilder

Η κλάση QueryBuilder του Codeigniter 4 είναι ένα ισχυρό εργαλείο που παρέχει έναν ευέλικτο και ασφαλή τρόπο για τη δημιουργία και την εκτέλεση ερωτημάτων βάσης δεδομένων. Η χρήση της κλάσης QueryBuilder μας επιτρέπει να γράφουμε κώδικα βάσης δεδομένων που είναι ανεξάρτητος από την βάση δεδομένων, να προστατεύει αυτόματα από επιθέσεις εισαγωγής SQL και να διαχειρίζεται αυτόματα τις διαφορές μεταξύ των διάφορων βάσεων δεδομένων.

Με την κλάση QueryBuilder, μπορούμε να δημιουργήσουμε ερωτήματα SQL χρησιμοποιώντας μεθόδους PHP. Αυτές οι μέθοδοι μας επιτρέπουν να γράψουμε ερωτήματα με έναν τρόπο που είναι πιο φυσικός και πιο κατανοητός για τους προγραμματιστές PHP. Για παράδειγμα, μπορούμε να χρησιμοποιήσουμε την μέθοδο 'select()' για να ορίσουμε τα πεδία που θέλουμε να επιστρέψουν, την μέθοδο 'from()' για να ορίσουμε τον πίνακα από τον οποίο θέλουμε να λάβουμε τα δεδομένα, και την μέθοδο 'where()' για να ορίσουμε τις συνθήκες που πρέπει να πληρούνται.

Επιπλέον, η κλάση QueryBuilder περιέχει μεθόδους για την εκτέλεση ερωτημάτων εισαγωγής, ενημέρωσης, διαγραφής και επιλογής. Αυτές οι μέθοδοι επιστρέφουν αντικείμενα που μπορούν να χρησιμοποιηθούν για να πάρουμε τα αποτελέσματα των ερωτημάτων, να δημιουργήσουμε loops στα αποτελέσματα, να μετρήσουμε τα αποτελέσματα και να εκτελέσουμε άλλες λειτουργίες.

Η κλάση QueryBuilder του Codeigniter 4 διευκολύνει τη διαχείριση των βάσεων δεδομένων και μας επιτρέπει να εργαστούμε με τη βάση δεδομένων με έναν τρόπο που είναι πιο ασφαλής, πιο αποτελεσματικός και πιο αποδοτικός.



```

1  <?php
2
3  $db      = \Config\Database::connect();
4  $builder->select('title, content, date');
5  $builder->from('mytable');
6  $query = $builder->get();
7  // Executes: SELECT title, content, date FROM mytable

```

Εικόνα 3.7 Παράδειγμα χρήσης QueryBuilder

3.3.12 Διαχείριση βάσης δεδομένων

Το Codeigniter παρέχει χρήσιμα εργαλεία για τη διαχείριση της βάσης δεδομένων, στο επίπεδο σχήματος και της μαζικής εισαγωγής δεδομένων. Τα Migrations και τα Seeders είναι δύο από αυτά τα εργαλεία.

Τα Migrations είναι ένας τρόπος δημιουργίας και τροποποίησης του σχήματος της βάσης δεδομένων με τη χρήση μεθόδων γραμμένων σε PHP, αντί να γράφουμε απευθείας κώδικα SQL. Με τη βοήθεια των Migrations, μπορούμε να δημιουργήσουμε μια νέα βάση δεδομένων, να προσθέσουμε ή να διαγράψουμε πίνακες, να αλλάξουμε τη δομή των πινάκων και να επιτελέσουμε άλλες τέτοιες λειτουργίες.

Για να δημιουργήσετε ένα νέο Migration, μπορείτε να χρησιμοποιήσετε την εντολή ‘make:migration’ στο spark CLI του Codeigniter [21]. Αυτή η εντολή θα δημιουργήσει ένα νέο αρχείο Migration στον φάκελο ‘app/Database/Migrations’. Το όνομα του αρχείου αυτού θα ξεκινάει με την αποτύπωση της χρονικής στιγμής της δημιουργίας (timestamp). Έτσι κάθε αρχείο είναι μοναδικό για το ιστορικό εκτέλεσης αλλαγών στη βάση δεδομένων. Κάθε Migration αρχείο που έχει εκτελεστεί αποθηκεύεται στον πίνακα ‘migrations’, ώστε να μην εκτελεστεί στον επόμενο κύκλο αλλαγών.

Το κάθε αρχείο Migration περιέχει δύο μεθόδους: up() και down(). Η μέθοδος up() χρησιμοποιείται για την εφαρμογή των αλλαγών στη βάση δεδομένων, ενώ η μέθοδος down() χρησιμοποιείται για την αναίρεση αυτών των αλλαγών. Η εντολή ‘migration:rollback’ εκτελεί τις μεθόδους down() από όλα τα Migrations που υπάρχουν στον πίνακα ‘migration’, αναιρώντας έτσι τις αλλαγές που είχαν γίνει.

Τα Seeders είναι ένας τρόπος για την προσθήκη δοκιμαστικών δεδομένων στη βάση δεδομένων. Αυτό μπορεί να είναι ιδιαίτερα χρήσιμο κατά τη διάρκεια της ανάπτυξης, όταν χρειάζεστε μια γρήγορη και ευέλικτη μέθοδο για την πληρωμή της βάσης δεδομένων με δεδομένα για δοκιμές.

Για να δημιουργήσετε ένα νέο Seeder, μπορείτε να χρησιμοποιήσετε την εντολή ‘make:seeder’ του CLI του Codeigniter. Αυτή η εντολή θα δημιουργήσει ένα νέο αρχείο Seeder στον φάκελο ‘app/Database/Seeds’.

Κάθε αρχείο Seeder περιέχει μια μέθοδο run(). Αυτή η μέθοδος χρησιμοποιείται για την προσθήκη δεδομένων στη βάση δεδομένων. Μπορείτε να χρησιμοποιήσετε την εντολή ‘db:seed’ για την εκτέλεση αρχείου Seeder.

3.3.13 MVC Framework

Το CodeIgniter είναι ένα PHP framework που στον πυρήνα του ακολουθεί το μοντέλο Model-View-Controller (MVC). Για τη διευκόλυνση του προγραμματιστή υπάρχουν προκαθορισμένοι φάκελοι μέσα στον φάκελο app, που αναφέρθηκε σε προηγούμενη ενότητα. Τα Models στο CodeIgniter αντιπροσωπεύουν τα δεδομένα και τη λογική της εφαρμογής, διαχειρίζονται τις ερωτήσεις της βάσης δεδομένων και επεξεργάζονται τα δεδομένα. Τα Views είναι υπεύθυνα για την παρουσίαση των δεδομένων στον χρήστη, εμφανίζοντας την έξοδο της εφαρμογής. Τα Controllers συνδέουν τα Models και τα Views, καθοδηγώντας τη ροή της εφαρμογής και διαχειρίζονται τις αιτήσεις των χρηστών. Αυτό το μοντέλο διαχωρισμού ευθυνών προωθεί την οργάνωση και την επαναχρησιμοποίηση του κώδικα, ενώ παράλληλα διευκολύνει τη συντήρηση και την επέκταση της εφαρμογής.

3.3.14 Controllers και δρομολόγηση

Οι Controllers είναι βασικά στοιχεία της αρχιτεκτονικής MVC και αναλαμβάνουν τον ρόλο του διαχειριστή της ροής της εφαρμογής. Στο Codeigniter 4, ένας Controller είναι μια κλάση που περιέχει δημόσιες μεθόδους που αντιπροσωπεύουν διάφορες διευθύνσεις URL της εφαρμογής. Οι μέθοδοι αυτές είναι οι λειτουργίες που θα εκτελεστούν όταν επισκεφτείτε την αντίστοιχη διεύθυνση URL. Αυτές οι λειτουργίες μπορεί να περιλαμβάνουν την επεξεργασία δεδομένων, την επικοινωνία με τη βάση δεδομένων, την παράδοση δεδομένων σε ένα View για παρουσίαση και άλλες τέτοιες εργασίες.

Η δρομολόγηση είναι η διαδικασία που καθορίζει πώς θα αντιστοιχιστεί μια διεύθυνση URL σε έναν συγκεκριμένο Controller. Αυτό επιτυγχάνεται μέσω του αρχείου ρυθμίσεων Routes, που βρίσκεται στον φάκελο 'app/Config'. Το Codeigniter 4 παρέχει διάφορους τρόπους δρομολόγησης, συμπεριλαμβανομένων της δυνατότητας να ορίσετε πολλαπλές διαδρομές για τον ίδιο Controller.

Στο αρχείο Routes, μπορείτε να ορίσετε τη δρομολόγηση με τη βοήθεια της μεθόδου 'get()', η οποία απαιτεί δύο παραμέτρους: τη διεύθυνση URL και τον Controller που θα χειριστεί αυτή τη διεύθυνση. Για παράδειγμα, '\$routes->get('products', 'ProductsController::index');' θα δρομολογήσει τη διεύθυνση '/products' στη μέθοδο 'index' του 'ProductsController'. Αυτό σημαίνει ότι όταν ένας χρήστης επισκέπτεται τη διεύθυνση URL '/products', θα κληθεί η μέθοδος 'index' του 'ProductsController'.

Το Codeigniter 4 παρέχει επίσης ένα ευέλικτο σύστημα φίλτρων που επιτρέπει την εφαρμογή επιπρόσθετων λειτουργιών πριν ή/και μετά την εκτέλεση ενός Controller. Αυτά τα φίλτρα μπορούν να χρησιμοποιηθούν για μια πληθώρα λειτουργιών, όπως η πιστοποίηση χρηστών, η αποσφαλμάτωση, η παρακολούθηση της εφαρμογής ή η εφαρμογή ειδικών λειτουργιών όπως η θέσπιση ειδικών κανόνων για την πρόσβαση σε συγκεκριμένες διευθύνσεις URL. Τα φίλτρα αυτά μπορούν να οριστούν για να εκτελούνται για όλες τις διευθύνσεις URL ή μόνο για επιλεγμένες διευθύνσεις URL σύμφωνα με τις ανάγκες της εφαρμογής.

3.3.15 Δημιουργία αποκρίσεων

Τα Views στο Codeigniter 4 είναι αρχεία PHP που περιέχουν κυρίως HTML και χρησιμοποιούνται για την παρουσίαση των δεδομένων στον χρήστη. Τα Views μπορούν να περιέχουν οποιονδήποτε τύπο κώδικα PHP, αλλά είναι συνήθως υπεύθυνα για την παρουσίαση των δεδομένων και όχι για την επεξεργασία τους. Στο Codeigniter 4, η μέθοδος 'view()' της κλάσης 'View' χρησιμοποιείται για τη φόρτωση ενός View. Αυτή η μέθοδος απαιτεί δύο παραμέτρους: το όνομα του View που θα φορτωθεί

και έναν πίνακα με δεδομένα που θα περαστούν στο View. Καθώς το Codeigniter είναι ένα MVC framework, την εργασία συλλογής δεδομένων και έπειτα κλήσης της μεθόδου, αναλαμβάνει ο Controller.

Η κλάση 'View' παρέχει τη μέθοδο 'render()' για τη φόρτωση ενός View. Αυτή η μέθοδος απαιτεί το όνομα του View που θα φορτωθεί ως παράμετρος και έναν προαιρετικό πίνακα με δεδομένα που θα περαστούν στο View. Τα δεδομένα περνούνται ως ένας συσχετισμένος πίνακας, όπου τα κλειδιά του πίνακα γίνονται μεταβλητές που είναι διαθέσιμες στο View.

Η κλάση 'ViewCell' είναι μια ισχυρή λειτουργία που επιτρέπει τη δυναμική φόρτωση και την ενσωμάτωση περιεχομένου σε ένα View. Αυτό είναι ιδιαίτερα χρήσιμο για την εμφάνιση δυναμικού περιεχομένου, όπως widgets ή εξαρτήματα, που πρέπει να φορτώνονται δυναμικά. Οι μέθοδοι 'ViewCell' μπορούν να κληθούν απευθείας από ένα View χρησιμοποιώντας τη σύνταξη '<?= cell('ClassName::method', ['param1', 'param2']) ?>'

Όταν χρησιμοποιούμε το Codeigniter 4 για τη δημιουργία ενός API, η διαχείριση των αποκρίσεων είναι ένα πολύ σημαντικό ζήτημα. Το Codeigniter 4 παρέχει την κλάση 'ResponseTrait' που περιέχει μια σειρά από χρήσιμες μεθόδους για τη δημιουργία και την επιστροφή αποκρίσεων API. Αυτές οι μέθοδοι μας επιτρέπουν να ορίσουμε τον κωδικό κατάστασης HTTP της απόκρισης, τον τύπο περιεχομένου και τα δεδομένα που θα επιστραφούν.

3.3.16 Models

Τα μοντέλα στο Codeigniter αποτελούν έναν από τους πυλώνες της αρχιτεκτονικής MVC (Model-View-Controller) που χρησιμοποιεί το framework. Τα μοντέλα είναι κλάσεις PHP που βοηθούν στη διαχείριση των δεδομένων της βάσης δεδομένων και συχνά αντιστοιχούν σε έναν πίνακα στη βάση δεδομένων. Κάθε μοντέλο περιέχει ιδιότητες και μεθόδους που αντιστοιχούν στις στήλες και τις λειτουργίες του πίνακα.

Για να δημιουργήσετε ένα μοντέλο, πρέπει πρώτα να φτιάξετε ένα νέο αρχείο PHP με το όνομα του μοντέλου στον φάκελο 'app/Models'. Το όνομα του μοντέλου πρέπει να ξεκινάει με κεφαλαίο γράμμα και να ακολουθεί τους κανόνες της ονομασίας των κλάσεων στην PHP. Στο αρχείο, θα ορίσετε μια κλάση με το ίδιο όνομα όπως το αρχείο.

Για να προσπελάσετε ένα μοντέλο από έναν controller, μπορείτε να το φορτώσετε χρησιμοποιώντας τη μέθοδο model() του controller και στη συνέχεια να καλέσετε τις μεθόδους του μοντέλου όπως είναι απαραίτητο.

Τα μοντέλα Codeigniter παρέχουν μια πληθώρα μεθόδων για την εργασία με δεδομένα, όπως find(), findAll(), insert(), update(), delete() κλπ. Αυτές οι μέθοδοι κάνουν ευκολότερη την εκτέλεση βασικών λειτουργιών βάσης δεδομένων όπως η εισαγωγή, ενημέρωση, ανάκτηση και διαγραφή δεδομένων. Επίσης είναι εφικτό να συνδυαστούν μέθοδοι της κλάσης Model με τις μεθόδους της κλάσης QueryBuilder, που αναφέρθηκαν νωρίτερα στην ενότητα.

Για παράδειγμα, για να ανακτήσετε όλες τις εγγραφές από έναν πίνακα, μπορείτε να χρησιμοποιήσετε τη μέθοδο 'findAll()' του μοντέλου. Αυτή η μέθοδος επιστρέφει έναν πίνακα με όλες τις εγγραφές του πίνακα. Για να εισάγετε νέα δεδομένα στον πίνακα, μπορείτε να χρησιμοποιήσετε τη μέθοδο 'insert()' του μοντέλου. Αυτή η μέθοδος δέχεται έναν πίνακα με τα νέα δεδομένα και επιστρέφει το ID της νέας εγγραφής.

Επίσης υπάρχει η δυνατότητα για την επικύρωση των δεδομένων με τη βοήθεια του Model, πριν αυτά εισέλθουν στη βάση δεδομένων με κάποια μέθοδο όπως `insert()`, `update()` ή `save()`.

Τα μοντέλα Codeigniter επιτρέπουν επίσης την προσαρμογή των λειτουργιών τους χρησιμοποιώντας τις συναρτήσεις επανάκλησης (callbacks). Αυτό σημαίνει ότι μπορείτε να ορίσετε προσαρμοσμένο κώδικα που θα εκτελεστεί πριν ή μετά από συγκεκριμένες λειτουργίες, όπως η εισαγωγή, ενημέρωση ή διαγραφή εγγραφών.

Συνολικά, τα μοντέλα παίζουν κεντρικό ρόλο στην ανάπτυξη εφαρμογών με το Codeigniter, καθώς διευκολύνουν τη διαχείριση των δεδομένων και την επικοινωνία με τη βάση δεδομένων.

3.3.17 Entities

Τα Entities στο Codeigniter 4 είναι κλάσεις που αναπαριστούν οντότητες στην εφαρμογή σας, οι οποίες συνήθως αντιστοιχούν σε μια γραμμή στη βάση δεδομένων. Μπορεί να είναι οτιδήποτε από μια εγγραφή χρήστη μέχρι ένα προϊόν σε ένα ηλεκτρονικό κατάστημα, ή ακόμη και μια διαδικασία ή ένα γεγονός. Τα Entities περιλαμβάνουν μεθόδους για τη διαχείριση των δεδομένων, το χειρισμό των εισόδων και τη χαρτογράφηση των δεδομένων.

Ο κύριος σκοπός των Entities είναι να διευκολύνουν τη διαχείριση των δεδομένων στην εφαρμογή σας. Παρέχουν μια αποτελεσματική, αρχιτεκτονικά καθαρή μέθοδο για τη διαχείριση των δεδομένων εισόδου και αποτρέπουν την εισαγωγή δεδομένων που δεν είναι αποδεκτά ή που δεν ανταποκρίνονται στις απαιτήσεις της εφαρμογής.

Η χρήση των Entities επιτρέπει επίσης την ευκολότερη χαρτογράφηση των δεδομένων στην εφαρμογή σε σχέση με τη βάση δεδομένων. Τα Entities μπορούν να χρησιμοποιηθούν για να ορίσουν πώς τα δεδομένα πρέπει να μετατραπούν ή να αντιστοιχιστούν από την εφαρμογή στο σχήμα της βάσης, επιτρέποντας έτσι την ευέλικτη διαχείριση και μεταφορά των δεδομένων σε διάφορες παρουσιάσεις πληροφορίας στον τελικό χρήστη.

Τέλος, τα Entities επιτρέπουν την ελεγχόμενη μετατροπή των τιμών (type casting). Μπορείτε να ορίσετε πώς μια τιμή πρέπει να μετατραπεί όταν αποθηκεύεται ή ανακτάται από τη βάση δεδομένων, επιτρέποντας έτσι την ομοιόμορφη και ελεγχόμενη διαχείριση των δεδομένων της εφαρμογής. Αυτό μπορεί να είναι ιδιαίτερα χρήσιμο για τη διαχείριση των τύπων δεδομένων, την εξασφάλιση της συνέπειας των δεδομένων και την προσαρμογή των δεδομένων σε συγκεκριμένες απαιτήσεις.

Συνεπώς, τα Entities στο Codeigniter 4 είναι ένα ισχυρό εργαλείο που διευκολύνει την ανάπτυξη εφαρμογών, βοηθώντας τους προγραμματιστές να διαχειριστούν τα δεδομένα με έναν πιο αποτελεσματικό και δομημένο τρόπο.

3.4 Επίλογος

Στο παρόν κεφάλαιο είδαμε την έννοια του framework και πως ένα τέτοιο εργαλείο συμβάλει ριζικά στην διαδικασία ανάπτυξης και συντήρησης μίας PHP εφαρμογής. Παρουσιάστηκαν κάποια από τα πιο δημοφιλή και συχνά χρησιμοποιούμενα frameworks. Έπειτα έγινε μία εκτενέστερη αναφορά στο Codeigniter 4 framework, το οποίο και χρησιμοποιήθηκε στην ανάπτυξη εφαρμογής για την παρούσα εργασία. Κλείνοντας έγινε μία αναφορά στην ενσωμάτωση του προτύπου MVC στο Codeigniter, με τη χρήση συγκεκριμένων κλάσεων που παρέχει το framework.

Κεφάλαιο 4ο: Ανάλυση της εφαρμογής

4.1 Εισαγωγή

Η εφαρμογή που έχει αναπτυχθεί για την παρούσα πτυχιακή εργασία βασίζεται στο δυναμικό framework Codeigniter 4 (Κεφάλαιο 3), συνδυάζοντας το με δημοφιλής τεχνολογίες και εργαλεία που παρουσιάστηκαν κατά τη διάρκεια του Κεφαλαίου 2 για την δημιουργία μιας εύχρηστης και λειτουργικής εφαρμογής. Ο κύριος σκοπός αυτής της εφαρμογής είναι να δημιουργήσει μία εύχρηστη και αποτελεσματική γέφυρα μεταξύ αγγελιών εργασίας και των βιογραφικών, επιτρέποντας στους χρήστες να διευκολύνουν την προσπάθειά τους για απασχόληση. Εξαιτίας αυτού, προκύπτει η ανάγκη για δύο ξεχωριστές κατηγορίες χρηστών. Η πρώτη κατηγορία αφορά τις επιχειρήσεις, οι οποίες μετά την εγγραφή τους στην εφαρμογή, μπορούν να αναρτήσουν τις αγγελίες τους, αυξάνοντας έτσι την προβολή τους στον κόσμο της εργασίας. Η δεύτερη κατηγορία χρηστών αφορά τους υποψήφιους, οι οποίοι, μετά την εγγραφή και τη συμπλήρωση μίας φόρμας βιογραφικού με τα στοιχεία τους, αποκτούν πρόσβαση σε όλες τις δημοσιευμένες αγγελίες, αυξάνοντας έτσι τις ευκαιρίες τους για εύρεση εργασίας.

4.2 Ιστορίες χρήσης

Παρακάτω θα γίνει μία ανάλυση των περιπτώσεων χρήσης για τις δύο κατηγορίες χρηστών, τις επιχειρήσεις και τους υποψήφιους.

Επιχείρηση:

Ως επιχείρηση θέλω να κάνω εγγραφή στην εφαρμογή, ώστε να χειρίζομαι το εξατομικευμένο προφίλ της επιχείρησης.

Ως επιχείρηση θέλω να συμπληρώνω τα στοιχεία της επιχείρησης, ώστε να είναι αυτομάτως διαθέσιμα σε κάθε αγγελία, που θα προστεθεί μετέπειτα.

Ως επιχείρηση θέλω να βλέπω μία λίστα με τις αγγελίες της επιχείρησης, ώστε να έχω μία συγκεντρωτική εικόνα για αυτές.

Ως επιχείρηση θέλω να προσθέτω μία νέα αγγελία, ώστε να γίνει αντιστοίχιση με τους εγγεγραμμένους υποψήφιους και ταυτόχρονα να υπάρχει διαθέσιμη για προβολή στους ενδιαφερόμενους υποψήφιους.

Ως επιχείρηση θέλω να επεξεργάζομαι μία υπάρχουσα αγγελία, ώστε να μπορώ να διορθώσω λάθη στα στοιχεία που καταχωρήθηκαν ή να προσθέσω επιπλέον στοιχεία.

Ως επιχείρηση θέλω να διαγράψω μία αγγελία, ώστε να μην είναι διαθέσιμη για προβολή ή για αντιστοίχιση με υποψήφιους.

Ως επιχείρηση θέλω να βλέπω μία λίστα με τις αντιστοιχίσεις που έχουν γίνει από το σύστημα, ώστε να έχω μία συγκεντρωτική εικόνα για το ποιες αγγελίες ταίριαζαν με κάποιους υποψήφιους.

Ως επιχείρηση θέλω να βλέπω τα στοιχεία ενός υποψήφιου που αντιστοιχήθηκε με κάποια από τις αγγελίες μου, ώστε να μάθω περισσότερες πληροφορίες σχετικά, όπως για παράδειγμα τα στοιχεία επικοινωνίας του.

Υποψήφιος:

Ως υποψήφιος θέλω να κάνω εγγραφή στην εφαρμογή, ώστε να χειρίζομαι το εξατομικευμένο προφίλ μου.

Ως υποψήφιος θέλω να συμπληρώνω τα στοιχεία του βιογραφικού μου, ώστε να είναι αυτομάτως διαθέσιμα σε κάθε νέα αντιστοίχιση με αγγελίες.

Ως υποψήφιος θέλω να βλέπω μία λίστα με τις αγγελίες που έχουν καταχωρηθεί στο σύστημα, ώστε να έχω μία συγκεντρωτική εικόνα για αυτές.

Ως υποψήφιος θέλω να κάνω μία αναζήτηση ή να εφαρμόσω κάποια φίλτρα στις αγγελίες που έχουν καταχωρηθεί στο σύστημα, ώστε να μπορώ να εντοπίσω αν υπάρχουν αγγελίες με τα κριτήρια που με ενδιαφέρουν.

Ως υποψήφιος θέλω να βλέπω μία λίστα με τις αντιστοιχίσεις που έχουν γίνει από το σύστημα, ώστε να έχω μία συγκεντρωτική εικόνα για το ποιες αγγελίες ταίριαζαν με το βιογραφικό μου.

4.3 Κριτήρια αντιστοίχισης

Για την αντιστοίχιση μίας αγγελίας με ένα ή περισσότερα βιογραφικά, και την εγγραφή αυτής στην βάση δεδομένων, υπάρχουν κάποια προκαθορισμένα κριτήρια από την ίδια την εφαρμογή. Τα κριτήρια αυτά βρίσκονται στη βάση και έχουν προκαθορισμένες τιμές, οι οποίες προστέθηκαν με seeders (κεφάλαιο 3), και παρακάτω υπάρχει μία περιγραφή αυτών και της συνθήκης αντιστοίχισης. Να σημειωθεί ότι για την αντιστοίχιση πρέπει να πληρούνται οι συνθήκες για όλα τα κριτήρια και όχι για μερικά από αυτά.

1. Τύπος απασχόλησης

- Τιμές (σταθερές): `Εποχική`, `Πλήρης`
- Ορισμός μορφής στην αγγελία της επιχείρησης.
- Ορισμός τουλάχιστον μίας μορφής από τον υποψήφιο.
- Η αντιστοίχιση θα γίνεται όταν η τιμή αγγελίας είναι ίδια με την τιμή του βιογραφικού.

2. Τομέας απασχόλησης

- Ορισμός τομέα απασχόλησης στην αγγελία της επιχείρησης.
- Ορισμός τουλάχιστον ενός τομέα απασχόλησης από τον υποψήφιο.
- Αντιστοίχιση όταν ο τομέας απασχόλησης της αγγελίας υπάρχει στις επιλογές του βιογραφικού.

3. Περιοχή απασχόλησης

- Ορισμός περιοχής στην αγγελία της επιχείρησης.
- Ορισμός τουλάχιστον μίας περιοχής από τον υποψήφιο.
- Αντιστοίχιση όταν η περιοχή της αγγελίας υπάρχει στις επιλογές του βιογραφικού.

4. Επίπεδο εκπαίδευσης

- Οι τιμές για τα επίπεδα εκπαίδευσης θα βρίσκονται προκαθορισμένες στη βάση, μαζί με αξία/ιεραρχία.
- Ορισμός ελάχιστου επιπέδου εκπαίδευσης στην αγγελία της επιχείρησης.
- Ορισμός μέγιστου επιπέδου εκπαίδευσης από τον υποψήφιο.
- Αντιστοίχιση όταν το επίπεδο εκπαίδευσης της αγγελίας είναι μικρότερο ή ίσο του επιπέδου εκπαίδευσης του υποψήφιου.

5. Γλώσσες επικοινωνίας

- Ορισμός γλώσσας στην αγγελία της επιχείρησης.
- Ορισμός των γλωσσών που ξέρει ο υποψήφιος.
- Αντιστοίχιση όταν η γλώσσα που υπάρχει στην αγγελία υπάρχει στις επιλογές του βιογραφικού.

Πίνακας 1 Κριτήρια αντιστοίχισης αγγελίας - βιογραφικού

Κριτήριο	Αγγελία	Βιογραφικό	Αντιστοίχιση
Τύπος απασχόλησης	1	1 έως πολλές	Μία από τις επιλογές του υποψήφιου είναι ίδια με της αγγελίας
Τομέας απασχόλησης	1	1 έως πολλές	Μία από τις επιλογές του υποψήφιου είναι ίδια με της αγγελίας
Περιοχή απασχόλησης	1	1 έως πολλές	Μία από τις επιλογές του υποψήφιου είναι ίδια με της αγγελίας
Επίπεδο Εκπαίδευσης	1	1	Μία από τις επιλογές του υποψήφιου είναι ίδια με της αγγελίας
Γλώσσες επικοινωνίας	1 έως πολλές	1 έως πολλές	Μία από τις επιλογές του υποψήφιου είναι ίδια με της αγγελίας

4.4 Διαχείριση χρηστών

4.4.1 Εισαγωγή

Η εφαρμογή μας έχει ως πρωταρχικό στόχο τη διαχείριση αγγελιών εργασίας που προσφέρουν διάφορες επιχειρήσεις. Αυτό περιλαμβάνει τη δημιουργία, την τροποποίηση και την διαγραφή αγγελιών. Παράλληλα, παρέχει τη δυνατότητα στους υποψήφιους εργαζόμενους να ανεβάζουν τα βιογραφικά τους, προκειμένου να βρουν την κατάλληλη εργασία που ταιριάζει στις δεξιότητες και τις προτιμήσεις τους. Ένα από τα κύρια χαρακτηριστικά της εφαρμογής είναι η αντιστοίχιση μεταξύ των αγγελιών εργασίας και των βιογραφικών των υποψηφίων, δίνοντας την ευκαιρία στις επιχειρήσεις να βρουν τον κατάλληλο υποψήφιο για κάθε θέση εργασίας που προσφέρουν.

Συνεπώς, ένα από τα βασικότερα ζητήματα που αφορούν την ασφαλή λειτουργία της εφαρμογής είναι η αυθεντικοποίηση και η εξουσιοδότηση των χρηστών. Σε μια εφαρμογή που προσφέρει διαφορετικές λειτουργίες ανάλογα με την κατηγορία του χρήστη, όπως η παρούσα, είναι απολύτως απαραίτητο να υπάρχει ένας αποτελεσματικός μηχανισμός διαχείρισης αυτών των διαδικασιών.

Σύμφωνα με την παραπάνω ανάλυση, είναι εμφανές ότι οι χρήστες της εφαρμογής μπορούν να καταταχθούν σε δύο διακριτές κατηγορίες: τις επιχειρήσεις και τους υποψήφιους εργαζόμενους. Κάθε κατηγορία έχει τις δικές της μοναδικές ανάγκες και δυνατότητες, και η εφαρμογή είναι σχεδιασμένη να ανταποκριθεί αποτελεσματικά σε αυτές. Συνεπώς, η εφαρμογή δέχεται την εγγραφή νέων χρηστών είτε ως 'επιχείρηση', προσφέροντας τους τα απαραίτητα εργαλεία για να βρουν τους κατάλληλους υποψηφίους, είτε ως 'υποψήφιος', προσφέροντας τους την ευκαιρία να βρουν την εργασία που ταιριάζει στις δεξιότητες, τις γνώσεις και τις προτιμήσεις τους.

Για τη διαχείριση των χρηστών της εφαρμογής, επιλέχθηκε να χρησιμοποιηθεί το πλαίσιο Shield [22]. Ένα τέτοιο εργαλείο είναι απαραίτητο αλλά και εξαιρετικά χρήσιμο στην παροχή ενός ασφαλούς και αποτελεσματικού μηχανισμού διαχείρισης χρηστών. Χάρη σε αυτό το εργαλείο, η εφαρμογή αποκτά τη δυνατότητα να ελέγχει την πρόσβαση των χρηστών και να διαχειρίζεται τα δικαιώματά τους με μεγάλη ευελιξία και ακρίβεια. Επομένως, χάρη στο Shield, η εφαρμογή μας είναι πιο σίγουρη και αξιόπιστη, με μεγαλύτερο έλεγχο επί των λειτουργιών που επιτρέπονται σε κάθε χρήστη.

Το πλαίσιο ήρθε σαν πακέτο επέκτασης στην έκδοση 4 του Codeigniter. Αποτελεί το επίσημο προτεινόμενο εργαλείο από την ομάδα του Codeigniter για λειτουργίες όπως η αυθεντικοποίηση χρηστών και η διαχείριση των λογαριασμών τους. Αυτή η πρόταση είναι ξεκάθαρα διατυπωμένη στη σελίδα τεκμηρίωσης του Codeigniter, όπου οι χρήστες μπορούν να βρουν πληροφορίες και οδηγίες για την εγκατάσταση και τη χρήση του.

Η λύση που παρέχεται είναι ουσιαστικά μία ολοκληρωμένη και πολύ λειτουργική εφαρμογή για το πλαίσιο Codeigniter 4, προσφέροντας μια σειρά από προηγμένες λειτουργικότητες. Για την εφαρμογή της παρούσας πτυχιακής εργασίας, θα υιοθετηθούν και θα εφαρμοστούν οι εξής λειτουργίες:

1. Η λειτουργικότητα της αυθεντικοποίησης συνεδρίας (session), που επιτρέπει τη διαχείριση και τον έλεγχο των συνεδριών χρηστών.
2. Είσοδος του χρήστη μέσω του ζευγαριού διαπιστευτηρίων email - κωδικός χρήστη, αυξάνοντας έτσι την ασφάλεια και την ευκολία της πρόσβασης.
3. Λειτουργία 'Να με θυμάσαι'.

4. Διαχείριση της πρόσβασης στις σελίδες βάσει της κατηγορίας του χρήστη ('Επιχείρηση', 'Υποψήφιος'), επιτρέποντας έτσι την προσαρμογή της πρόσβασης ανάλογα με την κατηγορία που ανήκει ο κάθε χρήστης.

4.4.2 Χρήση του Shield στην εφαρμογή

Για την ενσωμάτωση του Shield στην εφαρμογή έγιναν οι παρακάτω ενέργειες, σύμφωνα με την online τεκμηρίωση του:

- **Εγκατάσταση.** Για την εγκατάσταση, θα χρησιμοποιήσουμε τον Composer. Η εντολή στην γραμμή εντολών για την προσθήκη του Shield είναι: `composer require codeigniter4/shield`
- **Παραμετροποίηση.** Το επόμενο βήμα είναι η παραμετροποίηση. Στην κλάση `app/Config/AuthGroups.php`, ορίζουμε τις δύο ομάδες χρηστών της εφαρμογής μας: η `'company'`, που αντιπροσωπεύει την επιχείρηση που θα αναρτήσει τις αγγελίες εργασίας, και η `'candidate'`, που αντιπροσωπεύει τον υποψήφιο που θα συμπληρώσει το βιογραφικό του. Έπειτα, στην κλάση `app/Config/Auth.php`, ορίζουμε τις ανακατευθύνσεις που θα γίνονται όταν ολοκληρώνονται οι ενέργειες εγγραφής και εισόδου από τους χρήστες των παραπάνω ομάδων.
- **Επέκταση κλάσεων.** Το τελευταίο βήμα είναι η επέκταση των κλάσεων. Δημιουργήθηκαν οι κατάλληλες επεκτάσεις (κλάσεις, μέθοδοι, views) του πλαισίου Shield, ώστε να προσαρμοστούν οι λειτουργίες για την εγγραφή και την είσοδο των χρηστών.
- **Ρύθμιση πρόσβασης.** Η πρόσβαση ανά κατηγορία χρήστη προσδιορίζεται με την εφαρμογή των Filters(middleware) επάνω στα Routes(δρομολόγηση).

4.5 Σχήμα βάσης δεδομένων

Η βάση δεδομένων που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής είναι η MariaDB(MySQL), μια δημοφιλής επιλογή που έρχεται ενσωματωμένη στο πακέτο XAMPP. Καταρχάς, έγινε μια λεπτομερής ανάλυση των οντοτήτων που εμφανίζονται στις διαδικασίες της εφαρμογής, ένα βήμα που είναι κρίσιμο για την κατανόηση της λειτουργίας της. Μετά από αυτή την ανάλυση, δημιουργήθηκαν οι κατάλληλοι πίνακες και οι σχέσεις μεταξύ τους, όπου κρίθηκε απαραίτητο. Αυτό επέτρεψε την ορθή οργάνωση και αποτελεσματική διαχείριση των δεδομένων. Η αποθήκευση των δεδομένων στους πίνακες αυτούς δομείται με τα απαραίτητα πεδία, τα οποία έχουν προσεκτικά επιλεγμένες ονομασίες και τύπους δεδομένων για να εξασφαλίσουν την συνοχή και την ακρίβεια.

Κάθε πίνακας διαθέτει ένα αριθμητικό πεδίο 'id', το οποίο προσαυξάνεται αυτόματα από τη βάση σε κάθε νέα εγγραφή, καθιστώντας την κάθε εγγραφή μοναδική. Το πεδίο αυτό αποτελεί το 'κύριο κλειδί' του εκάστοτε πίνακα, είναι δηλαδή το στοιχείο αναφοράς στην μοναδική εγγραφή. Εκτός από το 'κύριο κλειδί', σε αρκετούς πίνακες υπάρχει και ένα 'ξένο κλειδί', στο οποίο αποθηκεύεται η τιμή για το 'κύριο κλειδί' μίας εγγραφής από άλλο πίνακα. Έτσι επιτυγχάνεται η συσχέτιση ένα προς πολλά.

Στους περισσότερους πίνακες, έχουν προστεθεί πεδία που αναπαριστούν χρονικές σημάνσεις δημιουργίας και επεξεργασίας της εγγραφής, τα `'created_at'` και `'updated_at'`. Με αυτόν τον τρόπο, εξασφαλίζεται η παρακολούθηση της δημιουργίας και της τροποποίησης, διασφαλίζοντας ότι το σύστημα καταγράφει ένα ιστορικό των πιο πρόσφατων αλληλεπιδράσεων και των αλλαγών.

Η δημιουργία πινάκων στη βάση δεδομένων γίνεται μέσω κλάσεων που έχουν δημιουργηθεί στον υποφάκελο `app/Database/Migrations`. Αυτές οι κλάσεις επεκτείνουν την κλάση `Migration` του `framework` και υλοποιούν το σώμα των μεθόδων `up()` και `down()` (βλ. 3.3.12). Χρησιμοποιώντας την εντολή `'php spark migrate'` στη γραμμή εντολών, εκτελούνται οι μέθοδοι `up()` από κάθε κλάση, δημιουργώντας τους πίνακες στη βάση δεδομένων. Κάθε φορά που η προηγούμενη εντολή εκτελείται επιτυχώς, τα ονόματα των αρχείων των κλάσεων που εκτελέστηκαν αποθηκεύονται στον πίνακα `migrations`, ο οποίος καταγράφει το ιστορικό των αλλαγών που έχουν γίνει. Έτσι διασφαλίζεται ότι το ίδιο `migration` δεν θα ξαναεκτελεστεί, αλλά και ότι μπορεί να αναιρεθούν οι αλλαγές, εκτελώντας την εντολή `'php spark migrate:rollback'`, η οποία τρέχει τις αντίστοιχες `down()` μεθόδους.

```

1  <?php
2
3  namespace App\Database\Migrations;
4
5  use CodeIgniter\Database\Migration;
6
7  class Candidate extends Migration
8  {
9      public function up()
10     {
11         $this->forge->addField([
12             'id' => [
13                 'type' => 'int',
14                 'constraint' => 11,
15                 'unsigned' => true,
16                 'auto_increment' => true
17             ],
18             'user_id' => ['type' => 'int', 'constraint' => 11, 'unsigned' => true],
19             'education_level_id' => ['type' => 'int', 'constraint' => 11, 'unsigned' => true],
20             'firstname' => ['type' => 'varchar', 'constraint' => 255],
21             'lastname' => ['type' => 'varchar', 'constraint' => 255],
22             'photo' => ['type' => 'varchar', 'constraint' => 255, 'null' => true],
23             'phone' => ['type' => 'varchar', 'constraint' => 20],
24             'mobile' => ['type' => 'varchar', 'constraint' => 20, 'null' => true],
25             'address' => ['type' => 'varchar', 'constraint' => 255],
26             'postal' => ['type' => 'varchar', 'constraint' => 10],
27             'city' => ['type' => 'varchar', 'constraint' => 255],
28             'country' => ['type' => 'varchar', 'constraint' => 255],
29             'notes' => ['type' => 'text', 'null' => true],
30             'visible' => ['type' => 'tinyint', 'constraint' => 1, 'default' => 0],
31             'created_at' => ['type' => 'datetime', 'null' => true],
32             'updated_at' => ['type' => 'datetime', 'null' => true],
33             'deleted_at' => ['type' => 'datetime', 'null' => true],
34         ]);
35
36         $this->forge->addPrimaryKey('id');
37         $this->forge->addForeignKey('user_id', 'users', 'id', 'CASCADE', 'CASCADE');
38         $this->forge->addForeignKey(
39             'education_level_id',
40             'education_level',
41             'id',
42             'CASCADE',
43             'CASCADE'
44         );
45
46         $this->forge->createTable('candidate', true);
47     }
48
49     public function down()
50     {
51         $this->forge->dropTable('candidate', true);
52     }
53 }
54

```

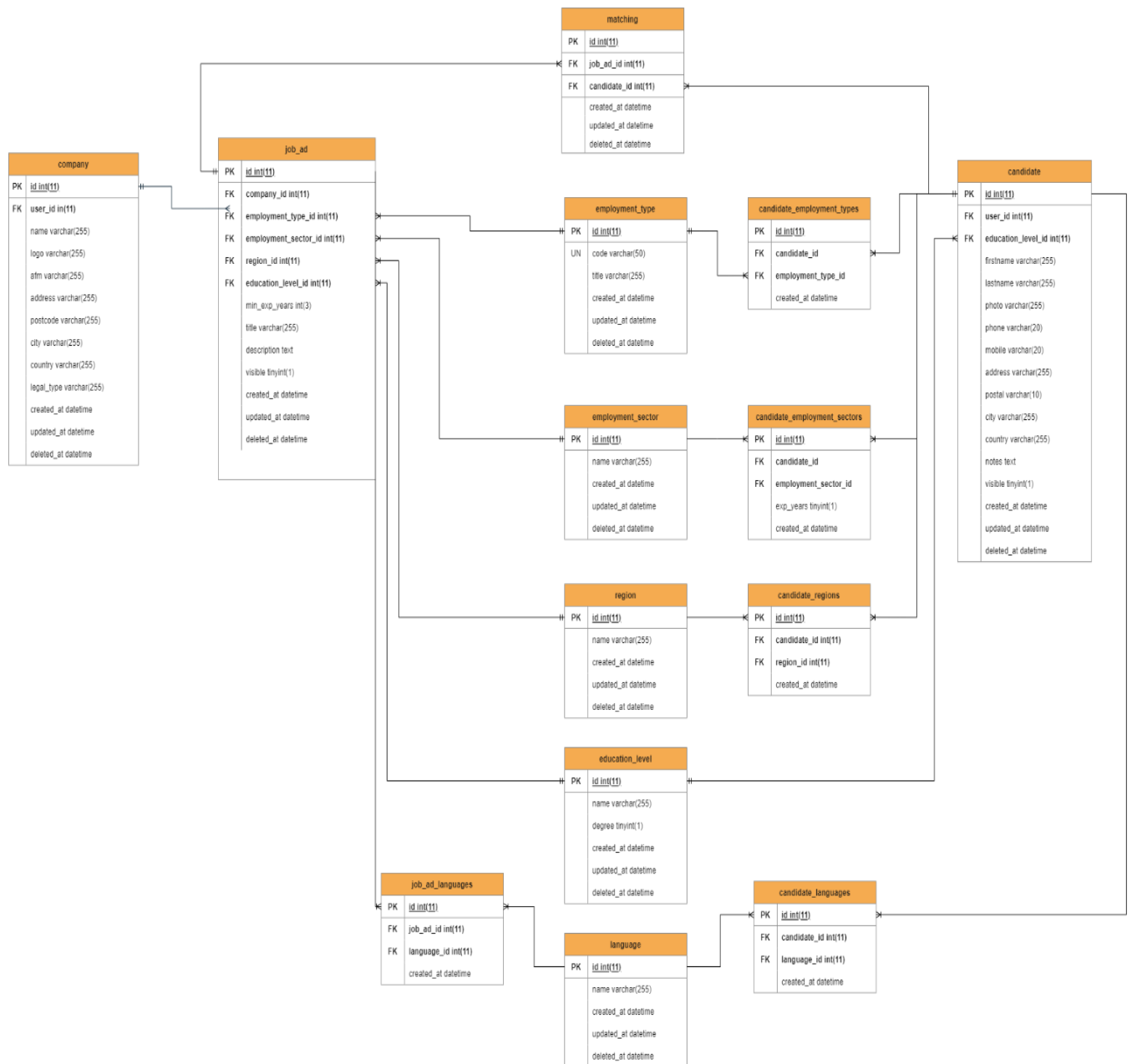
Εικόνα 4.1 Παράδειγμα Migration κλάσης

Για να αρχικοποιηθεί η βάση με κάποια δεδομένα, χρησιμοποιήθηκαν κλάσεις που επεκτείνουν την κλάση Seeder του framework και υλοποιούν το σώμα της μεθόδου run(). Στην συγκεκριμένη εφαρμογή, δημιουργήθηκε επίσης μια κλάση Seeder, η Deploy, η οποία καλεί όλους τους seeders για να μπορούν να εκτελεστούν μαζί. Επιπλέον, δημιουργήθηκε ένας αντίστοιχος πίνακας 'seeders' ώστε να καταγράφονται εκεί οι seeders που έχουν ήδη εκτελεστεί (δεν έχει προβλεφθεί από το framework).

```
1  <?php
2
3  namespace App\Database\Seeds;
4
5  use CodeIgniter\Database\Seeder;
6  use App\Models\Seeders as TrackSeeder;
7  use ReflectionException;
8
9  class Deploy extends Seeder
10 {
11     private array $allParams = [
12         'App\Database\Seeds\EmploymentType',
13         'App\Database\Seeds\EmploymentSector',
14         'App\Database\Seeds\Region',
15         'App\Database\Seeds\EducationLevel',
16         'App\Database\Seeds\Language',
17     ];
18
19     /**
20      * @throws ReflectionException
21      */
22     public function run()
23     {
24         $trackSeeder = new TrackSeeder();
25
26         foreach ($this->allParams as $seederClass) {
27
28             if ($trackSeeder->where('class', $seederClass)->find()) {
29                 continue;
30             }
31
32             $this->call($seederClass);
33
34             $trackSeeder->insert(['class' => $seederClass]);
35         }
36     }
37 }
38
```

Εικόνα 4.2 Η κλάση Deploy seeder

Στις παρακάτω εικόνες (Εικόνα 4.3, Εικόνα 4.4, Εικόνα 4.5, Εικόνα 4.6, Εικόνα 4.7, Εικόνα 4.8,) φαίνεται το διάγραμμα οντοτήτων συσχετίσεων, καθώς και η δομή κάποιων βασικών πινάκων της βάσης, που δημιουργούνται από τα migrations που περιγράφονται παραπάνω.



Εικόνα 4.3 Διάγραμμα βασικών οντοτήτων συσχετίσεων

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα
<input type="checkbox"/>	1 id	int(11)		UNSIGNED	Όχι	Καμία		AUTO_INCREMENT
<input type="checkbox"/>	2 user_id	int(11)		UNSIGNED	Όχι	Καμία		
<input type="checkbox"/>	3 name	varchar(255)	utf8_general_ci		Όχι	Καμία		
<input type="checkbox"/>	4 logo	varchar(255)	utf8_general_ci		Όχι	Καμία		
<input type="checkbox"/>	5 afm	varchar(255)	utf8_general_ci		Όχι	Καμία		
<input type="checkbox"/>	6 address	varchar(255)	utf8_general_ci		Όχι			
<input type="checkbox"/>	7 postcode	varchar(255)	utf8_general_ci		Όχι			
<input type="checkbox"/>	8 city	varchar(255)	utf8_general_ci		Όχι			
<input type="checkbox"/>	9 country	varchar(255)	utf8_general_ci		Όχι	Ελλάδα		
<input type="checkbox"/>	10 legal_type	varchar(255)	utf8_general_ci		Όχι			
<input type="checkbox"/>	11 created_at	datetime			Ναι	NULL		
<input type="checkbox"/>	12 updated_at	datetime			Ναι	NULL		
<input type="checkbox"/>	13 deleted_at	datetime			Ναι	NULL		

Εικόνα 4.4 Δομή πίνακα company

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα
<input type="checkbox"/>	1 id	int(11)		UNSIGNED	Όχι	Καμία		AUTO_INCREMENT
<input type="checkbox"/>	2 company_id	int(11)		UNSIGNED	Όχι	Καμία		
<input type="checkbox"/>	3 employment_type_id	int(11)		UNSIGNED	Όχι	Καμία		
<input type="checkbox"/>	4 employment_sector_id	int(11)		UNSIGNED	Όχι	Καμία		
<input type="checkbox"/>	5 region_id	int(11)		UNSIGNED	Όχι	Καμία		
<input type="checkbox"/>	6 education_level_id	int(11)		UNSIGNED	Όχι	Καμία		
<input type="checkbox"/>	7 min_exp_years	int(3)			Ναι	NULL		
<input type="checkbox"/>	8 title	varchar(255)	utf8_general_ci		Όχι	Καμία		
<input type="checkbox"/>	9 description	text	utf8_general_ci		Ναι	NULL		
<input type="checkbox"/>	10 visible	tinyint(1)			Όχι	1		
<input type="checkbox"/>	11 created_at	datetime			Ναι	NULL		
<input type="checkbox"/>	12 updated_at	datetime			Ναι	NULL		
<input type="checkbox"/>	13 deleted_at	datetime			Ναι	NULL		

Εικόνα 4.5 Δομή πίνακα job_ad

Κεφάλαιο 4

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα
<input type="checkbox"/>	1 id	int(11)		UNSIGNED	Όχι	Καμία		AUTO_INCREMENT
<input type="checkbox"/>	2 user_id	int(11)		UNSIGNED	Όχι	Καμία		
<input type="checkbox"/>	3 education_level_id	int(11)		UNSIGNED	Όχι	Καμία		
<input type="checkbox"/>	4 firstname	varchar(255)	utf8_general_ci		Όχι	Καμία		
<input type="checkbox"/>	5 lastname	varchar(255)	utf8_general_ci		Όχι	Καμία		
<input type="checkbox"/>	6 photo	varchar(255)	utf8_general_ci		Ναι	NULL		
<input type="checkbox"/>	7 phone	varchar(20)	utf8_general_ci		Όχι	Καμία		
<input type="checkbox"/>	8 mobile	varchar(20)	utf8_general_ci		Ναι	NULL		
<input type="checkbox"/>	9 address	varchar(255)	utf8_general_ci		Όχι	Καμία		
<input type="checkbox"/>	10 postal	varchar(10)	utf8_general_ci		Όχι	Καμία		
<input type="checkbox"/>	11 city	varchar(255)	utf8_general_ci		Όχι	Καμία		
<input type="checkbox"/>	12 country	varchar(255)	utf8_general_ci		Όχι	Καμία		
<input type="checkbox"/>	13 notes	text	utf8_general_ci		Ναι	NULL		
<input type="checkbox"/>	14 visible	tinyint(1)			Όχι	0		
<input type="checkbox"/>	15 created_at	datetime			Ναι	NULL		
<input type="checkbox"/>	16 updated_at	datetime			Ναι	NULL		
<input type="checkbox"/>	17 deleted_at	datetime			Ναι	NULL		

Εικόνα 4.6 Δομή πίνακα candidate

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα
<input type="checkbox"/>	1 id	int(11)		UNSIGNED	Όχι	Καμία		AUTO_INCREMENT
<input type="checkbox"/>	2 job_ad_id	int(11)		UNSIGNED	Όχι	Καμία		
<input type="checkbox"/>	3 candidate_id	int(11)		UNSIGNED	Όχι	Καμία		
<input type="checkbox"/>	4 created_at	datetime			Ναι	NULL		
<input type="checkbox"/>	5 updated_at	datetime			Ναι	NULL		
<input type="checkbox"/>	6 deleted_at	datetime			Ναι	NULL		

Εικόνα 4.7 Δομή πίνακα matching

#	Όνομα	Τύπος	Σύνθεση	Χαρακτηριστικά	Κενό	Προεπιλογή	Σχόλια	Πρόσθετα
<input type="checkbox"/>	1 id	int(11)		UNSIGNED	Όχι	Καμία		AUTO_INCREMENT
<input type="checkbox"/>	2 name	varchar(255)	utf8_general_ci		Όχι	Καμία		
<input type="checkbox"/>	3 degree	tinyint(1)		UNSIGNED	Όχι	Καμία		
<input type="checkbox"/>	4 created_at	datetime			Ναι	NULL		
<input type="checkbox"/>	5 updated_at	datetime			Ναι	NULL		
<input type="checkbox"/>	6 deleted_at	datetime			Ναι	NULL		

Εικόνα 4.8 Δομή πίνακα education_level (ένα από τα κριτήρια αντιστοίχισης)

4.6 Κλάσεις Model

Σε κάθε πίνακα που προστέθηκε στη βάση δεδομένων, δημιουργήθηκε μια αντίστοιχη κλάση στον υποφάκελο 'app/Models'. Αυτές οι κλάσεις είναι σχεδιασμένες ώστε να επεκτείνουν την ευελιξία και την λειτουργικότητα της βασικής κλάσης Model του framework. Είναι εξοπλισμένες με προκαθορισμένα πεδία που αναπαριστούν τα χαρακτηριστικά του πίνακα στη βάση δεδομένων, δίνοντας τους τη δυνατότητα να χειριστούν τα δεδομένα με ακρίβεια. Για παράδειγμα, τα πεδία 'protected \$table' και 'protected \$primaryKey' είναι οι σημαντικές μεταβλητές που καθορίζουν το όνομα του πίνακα και το κύριο κλειδί αντίστοιχα. Αυτά τα πεδία είναι κρίσιμα για την παραπομπή στον σωστό πίνακα και για τη διαχείριση των κύριων κλειδιών του. Με την πρόσθεση του πεδίου 'protected \$validationRules', προσφέρεται η δυνατότητα ελέγχου της εγκυρότητας των δεδομένων που θα εισαχθούν στα πεδία του πίνακα.

Η κλάση Model διαθέτει επίσης μια σειρά από μεθόδους που επιτρέπουν την πρόσβαση, την ενημέρωση, την εισαγωγή και τη διαγραφή των δεδομένων του πίνακα. Αυτές οι μέθοδοι βοηθούν στην αποτελεσματική και αξιόπιστη διαχείριση των δεδομένων στη βάση.

```

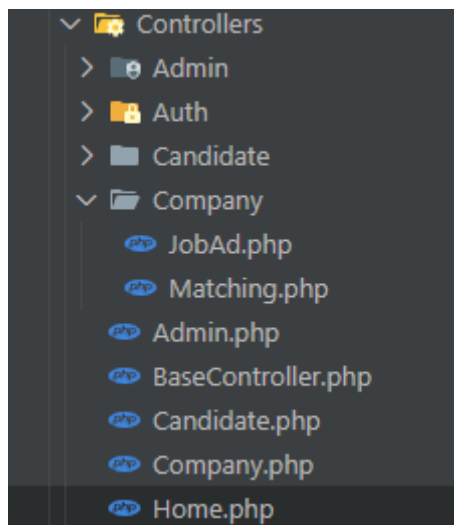
1  class Company extends Model
2  {
3      protected $table           = 'company';
4      protected $primaryKey      = 'id';
5      protected $useAutoIncrement = true;
6      protected $returnType      = \App\Entities\Company::class;
7      protected $useSoftDeletes  = false;
8      protected $protectFields   = true;
9      protected $allowedFields   = [
10         'user_id',
11         'name',
12         'logo',
13         'afm',
14         'address',
15         'postcode',
16         'city',
17         'country',
18         'legal_type',
19     ];
20
21     // Dates
22     protected $useTimestamps = true;
23     protected $dateFormat    = 'datetime';
24     protected $createdField   = 'created_at';
25     protected $updatedField   = 'updated_at';
26     protected $deletedField   = 'deleted_at';
27
28     // Validation
29     protected $validationRules = [
30         'afm'      => 'exact_length[9]',
31         'name'     => 'required',
32         'address'  => 'required',
33         'postcode' => 'required|exact_length[5]',
34         'city'     => 'required',
35         'country'  => 'required',
36     ];

```

Εικόνα 4.9 Κλάση Company Model

4.7 Controllers

Για την αποτελεσματική διαχείριση των αιτημάτων των τελικών χρηστών, έχουν δημιουργηθεί συγκεκριμένοι controllers. Αυτοί οι controllers έχουν οργανωθεί είτε βάσει των ομάδων χρηστών είτε σύμφωνα με τις κοινές λειτουργίες της εφαρμογής. Επιπλέον, η οργάνωση των controllers αυτή επιτρέπει την αποδοτική δρομολόγηση των αιτημάτων, εξασφαλίζοντας ότι κάθε αίτημα επεξεργάζεται από τον κατάλληλο controller. Σε συνδυασμό με την αποτελεσματική διαχείριση των δεδομένων μέσω των κλάσεων Model, οι controllers επεξεργάζονται τα αιτήματα και παράγουν τις κατάλληλες αποκρίσεις. Σε αυτή την εφαρμογή οι αποκρίσεις εμφανίζονται σε html μορφή, και αφορούν την εμφάνιση δεδομένων από τη βάση στην οθόνη του τελικού χρήστη. Παρακάτω φαίνεται η δομή των αρχείων που δημιουργήθηκαν.



Εικόνα 4.10 Αρχεία Controllers

4.8 Views

Έχει δημιουργηθεί μια κεντρική προβολή, η οποία βρίσκεται στη διαδρομή 'app/Views/layout.php', με στόχο τη δημιουργία της τελικής απόκρισης που θα εμφανιστεί στον χρήστη. Αυτή η προβολή ορίζει τη γενική διάταξη που θα ακολουθούν όλες οι HTML σελίδες που αποστέλλονται στον χρήστη. Αυτή η διάταξη καλύπτει τα στοιχεία head και footer της HTML, που είναι κοινά για όλες τις σελίδες, καθώς και άλλα στοιχεία που εμφανίζονται σε ολόκληρη την εφαρμογή, όπως το κεντρικό μενού. Στη συνέχεια, τα εξατομικευμένα views που έχουν δημιουργηθεί για κάθε σελίδα, επεκτείνουν αυτή τη βασική διάταξη, εμφανίζοντας τα δικά τους μοναδικά δεδομένα και πληροφορίες, καθιστώντας έτσι κάθε σελίδα μοναδική.

Στις παρακάτω εικόνες φαίνεται η δομή που ακολουθούν όλες οι HTML σελίδες, και ο τρόπος που ένα εξατομικευμένο view επεκτείνει τη δομή αυτή (Εικόνα 4.11, Εικόνα 4.12).

```
1 <!DOCTYPE html>
2 <html lang="el">
3 <?= view('App\Views\partials\head.php'); ?>
4
5 <body class="body-wrapper">
6   <div class="wrapper">
7
8     <?= view('App\Views\partials\header.php'); ?>
9     <?= view('App\Views\partials\messages.php'); ?>
10    <?= $this->renderSection('main') ?>
11    <?= view('App\Views\partials\footer.php'); ?>
12  </div>
13  <?= view('App\Views\partials\footerscripts.php'); ?>
14  <?= $this->renderSection('modals') ?>
15  <?= $this->renderSection('pageScripts') ?>
16 </body>
17
18 </html>
```

Εικόνα 4.11 Κεντρικό layout view

```
1 <?= $this->extend('App\Views\layout.php') ?>
2
3 <?= $this->section('main') ?>
4 <!-- Page content-->
5 <div class="container">
6   <div class="row justify-content-center text-center mt-5">
7     <h1>Δημιουργία επιχείρησης</h1>
```

Εικόνα 4.12 Επέκταση του layout view

4.9 Μηχανισμός αντιστοίχισης

Για το μηχανισμό αντιστοίχισης έγινε υλοποίηση σε τρία διαφορετικά σημεία του κώδικα.

Πρώτον, δημιουργήθηκαν μέθοδοι εύρεσης αντιστοίχισης ανά κριτήριο σε κάθε οντότητα. Αυτές οι μέθοδοι αναζητούν στη βάση δεδομένων τις εγγραφές της άλλης οντότητας που έχουν την ίδια τιμή για το συγκεκριμένο κριτήριο. Για παράδειγμα, στην αγγελία υπάρχει η μέθοδος `employmentTypeMatch`, η οποία θα αναζητήσει όλους τους υποψήφιους που έχουν την ίδια τιμή για το `employment_type`, το είδος απασχόλησης δηλαδή.

Στη συνέχεια, δημιουργήθηκε η κλάση `'app/Libraries/Matching'`, η οποία παρέχει τις δημόσιες μεθόδους `matchJobAdToCandidates(JobAd $jobAd)` και `matchCandidateToJobAds(Candidate $candidate)`. Αυτές οι μέθοδοι ξεκινούν τη διαδικασία κλήσης όλων των μεθόδων αντιστοίχισης της οντότητας, επιστρέφοντας τα κύρια κλειδιά της άλλης οντότητας, όπου βρέθηκε ότι τα κριτήρια ταιριάζουν. Τέλος, στα αποτελέσματα ανά κριτήριο θα γίνει διατομή για να εντοπιστούν τα κοινά κύρια κλειδιά και να καταγραφούν στη βάση δεδομένων.

Όλη η παραπάνω διαδικασία καλείται από την Model κλάση της εκάστοτε οντότητας, μέσω της εξατομικευμένης μεθόδου `save()`. Αυτό σημαίνει ότι κάθε φορά που προστίθεται μια νέα αγγελία ή ένα νέο βιογραφικό στην βάση δεδομένων, η αντίστοιχη Model κλάση αναλαμβάνει την ευθύνη να ψάξει για αντιστοίχιση μεταξύ αυτών.

```
1 public function saveCandidate(CandidateEntity $candidate): void
2     {
3         if($candidate->hasChanged()){
4             $this->save($candidate);
5         }
6     }
7     $candidate->id = $candidate->id ?? $this->getInsertID();
8
9     $this->saveAllRelated($candidate);
10
11     Matching::matchCandidateToJobAds($candidate);
12 }
```

Εικόνα 4.13 Η μέθοδος του Candidate Model που εκκινεί τη διαδικασία αντιστοίχισης

```

1  /**
2   * Match a candidate entity with jobAds
3   * Exact matches that meet the criteria
4   *
5   * @param Candidate $candidate
6   * @return void
7   */
8  public static function matchCandidateToJobAds(Candidate $candidate): void
9  {
10     $jobAdIds = self::getMatchingIds($candidate);
11
12     try {
13         (new MatchingModel()->where('candidate_id', $candidate->id)->delete());
14     } catch (DatabaseException $ex) {}
15     foreach ($jobAdIds as $jobAdId) {
16         self::saveMatching($jobAdId, $candidate->id);
17     }
18 }
19
20 /**
21  * Save a matching pair of job_ad_id and candidate_id
22  *
23  * @param integer $jobAdId
24  * @param integer $candidateId
25  * @return void
26  */
27 protected static function saveMatching(int $jobAdId, int $candidateId): void
28 {
29     $matching = new MatchingEntity();
30     $matching->job_ad_id = $jobAdId;
31     $matching->candidate_id = $candidateId;
32     try {
33         (new MatchingModel()->save($matching));
34     } catch (DatabaseException $ex) {}
35 }
36
37
38 /**
39  * Get the matching ids for specific entity (jobAd|candidate)
40  *
41  * @param mixed $matchingEntity
42  * @param array $criteria
43  * @return array
44  */
45 protected static function getMatchingIds(
46     mixed $matchingEntity,
47     array $criteria = [
48         'employmentTypeMatch',
49         'employmentSectorMatch',
50         'regionMatch',
51         'educationLevelMatch',
52         'languageMatch',
53     ]
46 ): array
54 {
55     $matchPerCriteria = [];
56     foreach($criteria as $criterion)
57         $matchPerCriteria[] = $matchingEntity->$criterion();
58
59     // Return only the common ids
60     return array_intersect(...$matchPerCriteria);
61 }
62

```

Εικόνα 4.14 Οι μέθοδοι της Matching κλάσης για την αντιστοίχιση υποψήφιου με βιογραφικά

4.10 Δρομολόγηση

Η δρομολόγηση σε μία web εφαρμογή με CodeIgniter 4, αφορά την συσχέτιση ενός URI(Uniform Resource Identifier) με μία μέθοδο ενός controller.

Το CodeIgniter χρησιμοποιεί δύο κύριες μεθόδους δρομολόγησης. Η πρώτη από αυτές είναι η δρομολόγηση με καθορισμένες διαδρομές, που παρέχει τη δυνατότητα στους προγραμματιστές να ορίζουν χειροκίνητα τις διαδρομές των URL. Αυτό επιτρέπει μια αυξημένη ευελιξία σε ό,τι αφορά τη διαμόρφωση του URL, καθώς μπορείτε να το προσαρμόσετε ακριβώς στις ανάγκες σας. Η δεύτερη μέθοδος είναι η αυτόματη δρομολόγηση, η οποία αναλαμβάνει αυτόματα τη δρομολόγηση των αιτήσεων HTTP με βάση ορισμένες προεπιλεγμένες συμβάσεις. Αυτό σημαίνει ότι εκτελεί αυτόματα τις αντίστοιχες μεθόδους ελεγκτή χωρίς την ανάγκη για χειροκίνητο ορισμό διαδρομών. Κατ' αυτόν τον τρόπο, οι προγραμματιστές μπορούν να εξοικονομήσουν χρόνο και να αυξήσουν την αποτελεσματικότητα της διαδικασίας ανάπτυξης.

Στην παρούσα υλοποίηση επιλέχθηκε η πρώτη μέθοδος, με σκοπό την καλύτερη διαχείριση των διαδρομών από τη μεριά του προγραμματιστή, με την προσαρμογή των διαδρομών σε συγκεκριμένες μεθόδους. Στην παρακάτω εικόνα φαίνονται αναλυτικά οι βασικές διαδρομές που είναι διαθέσιμες στην εφαρμογή. Επίσης φαίνεται η ομαδοποίησή τους σε αντιστοιχία με τις κατηγορίες χρηστών, καθώς και η δήλωση εφαρμογής filters, για την πρόσβαση χρηστών στη συγκεκριμένη ομάδα διαδρομών.

```

1 $routes->get('/', 'Home::index');
2 $routes->get('info/candidate', 'Home::forCandidate');
3 $routes->get('info/company', 'Home::forCompany');
4
5 $routes->group('admin', function ($routes) {
6     $routes->get('login', 'Auth\AdminLogin::loginView');
7     $routes->post('login', 'Auth\AdminLogin::loginAction');
8
9     $routes->group('', ['filter' => 'group:admin'], function ($routes) {
10        $routes->get('/', 'Admin::index');
11
12        $routes->group('users', function ($routes) {
13            $routes->get('list', 'Admin\Users::index');
14            $routes->get('edit/(:num)', 'Admin\Users::edit/$1', ['as' => 'admin/users/edit']);
15            $routes->post('save', 'Admin\Users::save');
16            $routes->get('delete/(:num)', 'Admin\Users::delete/$1', ['as' => 'admin/users/delete']);
17        });
18    });
19 });
20
21 $routes->group('candidate', function ($routes) {
22     $routes->get('login', 'Auth\CandidateLogin::loginView');
23     $routes->post('login', 'Auth\CandidateLogin::loginAction');
24     $routes->get('register', 'Auth\CandidateRegister::registerView');
25     $routes->post('register', 'Auth\CandidateRegister::registerAction');
26
27     $routes->group('', ['filter' => 'group:candidate'], function ($routes) {
28        $routes->get('/', 'Candidate::index');
29        $routes->post('save', 'Candidate::save');
30
31        $routes->group('jobs', function ($routes) {
32            $routes->get('list', 'Candidate\JobAd::index');
33            $routes->get(
34                'details/(:num)', 'Candidate\JobAd::details/$1',
35                ['as' => 'candidate/jobs/details']
36            );
37            $routes->get('match', 'Candidate\Matching::index');
38        });
39    });
40 });
41
42 $routes->group('company', function ($routes) {
43     $routes->get('login', 'Auth\CompanyLogin::loginView');
44     $routes->post('login', 'Auth\CompanyLogin::loginAction');
45     $routes->get('register', 'Auth\CompanyRegister::registerView');
46     $routes->post('register', 'Auth\CompanyRegister::registerAction');
47
48     $routes->group('', ['filter' => 'group:company'], function ($routes) {
49        $routes->get('/', 'Company::index');
50        $routes->post('save', 'Company::save');
51
52        $routes->group('jobs', function ($routes) {
53            $routes->get('list', 'Company\JobAd::index');
54            $routes->get('create', 'Company\JobAd::create');
55            $routes->post('add', 'Company\JobAd::add');
56            $routes->get('edit/(:num)', 'Company\JobAd::edit/$1', ['as' => 'company/jobs/edit']);
57            $routes->post('save', 'Company\JobAd::save');
58            $routes->get('delete/(:num)', 'Company\JobAd::delete/$1', ['as' => 'company/jobs/delete']);
59            $routes->get('match', 'Company\Matching::index');
60            $routes->get(
61                'match/details(:num)', 'Company\Matching::details/$1',
62                ['as' => 'company/jobs/match/details']
63            );
64        });
65    });
66 });

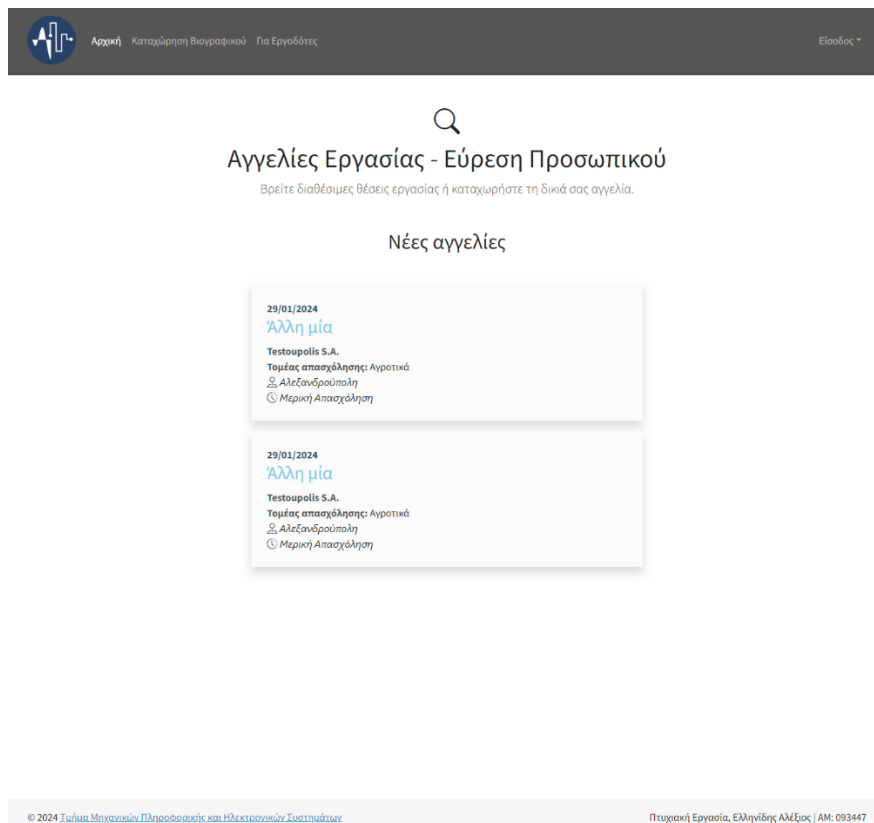
```

Εικόνα 4.15 Αρχείο Routes.php

Κεφάλαιο 5ο: Παρουσίαση της εφαρμογής

5.1 Εισαγωγή

Γενικά για την πλήρη εκμετάλλευση των λειτουργιών της εφαρμογής τόσο για την επιχείρηση, όσο και για τον υποψήφιο, απαιτείται η εγγραφή. Χωρίς εγγραφή κάποιος μπορεί απλώς να δει στην αρχική σελίδα μία λίστα με τις πρόσφατα αναρτημένες αγγελίες, όπως φαίνεται στην *Εικόνα 5.1*



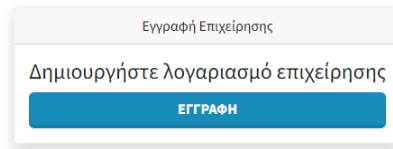
Εικόνα 5.1 Αρχική σελίδα εφαρμογής

Ενώ εάν επιλέξει ένα από τα μενού 'Καταχώρηση Βιογραφικού' ή 'Για εργοδότες', θα εμφανιστεί το αντίστοιχο μήνυμα προτροπής για την εγγραφή στην εφαρμογή. Όπως φαίνεται στην *Εικόνα 5.2* και στην *Εικόνα 5.3*



Βρείτε το κατάλληλο προσωπικό

Αναρτήστε τις δικές σας αγγελίες εργασίας.



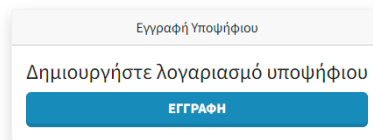
Εικόνα 5.2 Προτροπή για εγγραφή επιχείρησης



Καταχώρηση Βιογραφικού - Θέσεις Εργασίας

Εγγραφείτε και προσθέστε τα στοιχεία του βιογραφικού σας.

Βρείτε τη θέση εργασίας που σας ταιριάζει.



Εικόνα 5.3 Προτροπή για εγγραφή υποψήφιου

5.2 Λειτουργικότητες χρηστών

5.2.1 Εισαγωγή

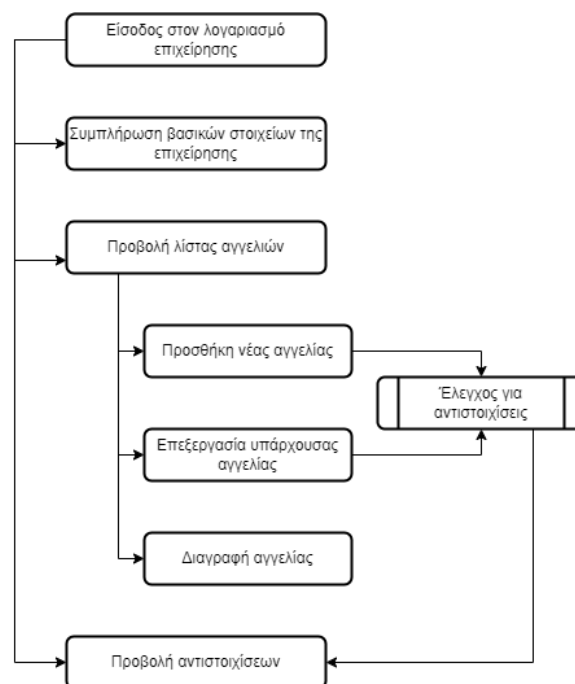
Γενικά, έχει αναπτυχθεί μια δομή που ακολουθεί αυτή των routes για τους controllers της εφαρμογής. Συγκεκριμένα, στη ρίζα του φακέλου 'Controllers', βρίσκονται οι γενικοί controllers που σχετίζονται με έναν δημόσιο χρήστη της εφαρμογής, διασφαλίζοντας την κανονική λειτουργία της εφαρμογής.

Επιπλέον, προστέθηκε ένας ειδικός φάκελος με την ονομασία 'Auth', ο οποίος περιλαμβάνει τους controllers που έχουν δημιουργηθεί ως επέκταση των controllers της βιβλιοθήκης Shield. Αυτό το στοιχείο βοήθησε σημαντικά στην προσαρμογή ορισμένων διαδικασιών και λειτουργιών, όπως για παράδειγμα την ενεργοποίηση του χρήστη κατά την εγγραφή. Στην κανονική διαδικασία, αυτό θα έπρεπε να γίνει μέσω της αποστολής ενός email και της επιβεβαίωσης από την πλευρά του χρήστη, ωστόσο, στην παρούσα εργασία, παρακάμφθηκε οποιαδήποτε ρύθμιση για αποστολή email.

Επίσης, δημιουργήθηκε ένα προσαρμοσμένο view για τις φόρμες εγγραφής και εισόδου, προκειμένου να προσφέρει μια πιο ευχάριστη εμπειρία στον χρήστη. Τέλος, για κάθε κατηγορία χρήστη, όπως οι 'Company' και οι 'Candidate', δημιουργήθηκαν οι αντίστοιχοι φάκελοι και μέσα σε αυτούς προστέθηκαν οι controllers που σχετίζονται με τις δυνατότητες της κάθε κατηγορίας.

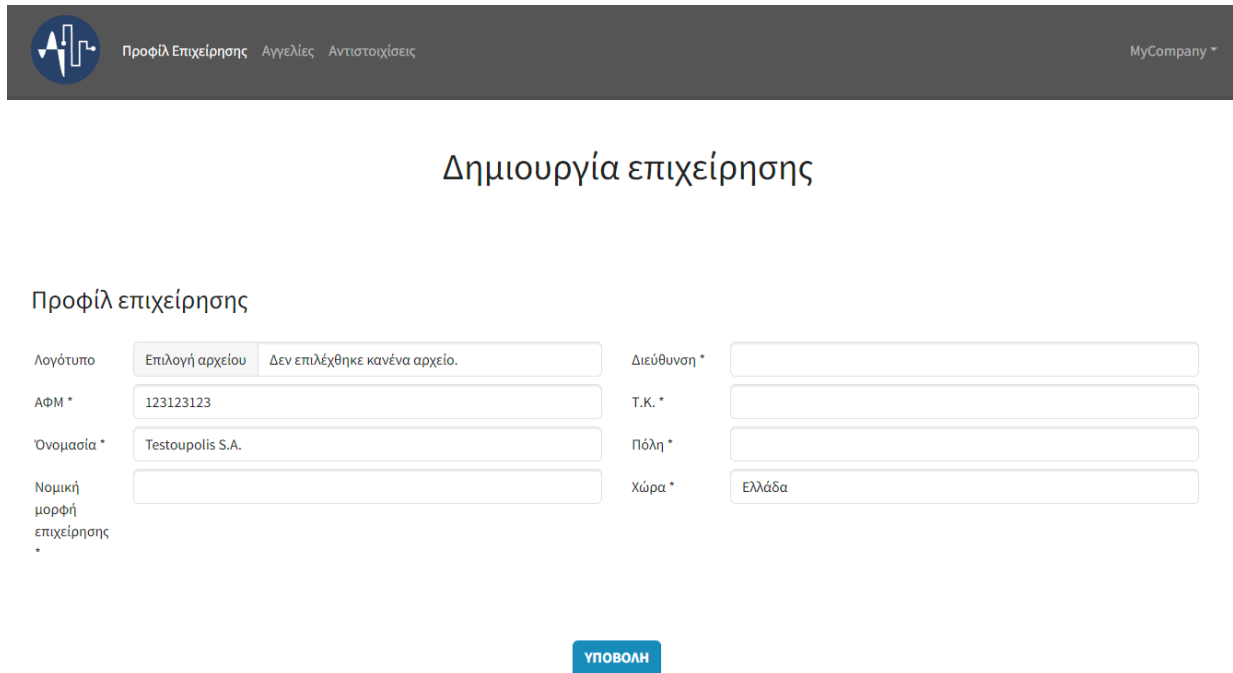
Οι controllers αυτοί διαχειρίζονται τις ενέργειες του εκάστοτε χρήστη, οι οποίες στην γενικότερη μορφή τους είναι είτε ένα αίτημα για να λάβουν μία σελίδα, είτε υποβολή δεδομένων μέσω μίας φόρμας. Αυτό είναι εμφανές και από την προηγούμενη αναφορά στα routes της εφαρμογής (Εικόνα 5.10), όπου όλα τα routes χαρακτηρίζονται ως 'get' ή 'post', καθώς είναι μια εφαρμογή που παράγει μόνο αποκρίσεις HTML από τη μεριά του εξυπηρετή.

5.2.2 Χρήστης 'Επιχείρηση'



Εικόνα 5.4 Λειτουργικότητες χρήστη 'Επιχείρηση'

Ένας χρήστης που ανήκει στην κατηγορία 'Επιχείρηση' και συνδέεται στο λογαριασμό του μετά από μια επιτυχή εγγραφή στην εφαρμογή, αποκτά τη δυνατότητα να εκτελέσει μια σειρά από προσωποποιημένες ενέργειες. Πρώτον, μπορεί να συμπληρώσει μια φόρμα που περιλαμβάνει τα βασικά στοιχεία της επιχείρησής του, όπως το όνομα, τα στοιχεία επικοινωνίας, τη διεύθυνση, κλπ. Αυτές οι πληροφορίες καταχωρούνται στο σύστημα και γίνονται αυτόματα διαθέσιμες κάθε φορά που ο χρήστης προσθέτει μια νέα αγγελία στην πλατφόρμα, εκ μέρους της επιχείρησης. Αυτό επιτρέπει στον χρήστη να εξοικονομεί χρόνο και να καθιστά την διαδικασία προσθήκης αγγελιών πιο αποτελεσματική.



Προφίλ επιχείρησης

Λογότυπο	<input type="text" value="Επιλογή αρχείου"/> Δεν επιλέχθηκε κανένα αρχείο.	Διεύθυνση *	<input type="text"/>
ΑΦΜ *	<input type="text" value="123123123"/>	Τ.Κ. *	<input type="text"/>
Όνομασία *	<input type="text" value="Testoupolis S.A."/>	Πόλη *	<input type="text"/>
Νομική μορφή επιχείρησης *	<input type="text"/>	Χώρα *	<input type="text" value="Ελλάδα"/>

Εικόνα 5.5 Φόρμα στοιχείων επιχείρησης

Η κύρια λειτουργικότητα που προσφέρεται στον χρήστη της κατηγορίας αυτής, η οποία αποτελεί τον πυρήνα της χρησιμότητας της συγκεκριμένης εφαρμογής, είναι η ολοκληρωμένη διαχείριση αγγελιών εργασίας που σχετίζονται άμεσα με την επιχείρηση. Μέσα από αυτήν τη λειτουργικότητα, ο χρήστης έχει τη δυνατότητα να διαχειριστεί εύκολα και γρήγορα τις αγγελίες εργασίας της επιχείρησης. Μπορεί να βλέπει μία πλήρη λίστα με όλες τις αγγελίες που έχει δημιουργήσει, να τροποποιεί τις υπάρχουσες αγγελίες ή να προσθέτει νέες καταχωρήσεις, βελτιώνοντας έτσι την αποτελεσματικότητα της διαχείρισης των αγγελιών και της εύρεσης προσωπικού.



Οι αγγελίες μου

ΔΗΜΙΟΥΡΓΙΑ ΑΓΓΕΛΙΑΣ

#	Αγγελία	Ορατή	Ενέργειες
1	Θέση εργασίας τέστ Τύπος απασχόλησης: Μερική Απασχόληση Τομέας απασχόλησης: Αγροτικά Επίπεδο εκπαίδευσης: Λύκειο	✓	
2	Θέση εργασίας τέστ 2 Τύπος απασχόλησης: Πλήρης Απασχόληση Τομέας απασχόλησης: Logistics - Μεταφορές Επίπεδο εκπαίδευσης: Λύκειο	✓	
4	Τέστ Τύπος απασχόλησης: Πλήρης Απασχόληση Τομέας απασχόλησης: Τουρισμός Επίπεδο εκπαίδευσης: Υποχρεωτική	✓	
8	Άλλη μία Τύπος απασχόλησης: Μερική Απασχόληση Τομέας απασχόλησης: Αγροτικά Επίπεδο εκπαίδευσης: Υποχρεωτική	✓	
9	Άλλη μία Τύπος απασχόλησης: Μερική Απασχόληση Τομέας απασχόλησης: Αγροτικά Επίπεδο εκπαίδευσης: Υποχρεωτική	✓	

Εικόνα 5.6 Αγγελίες της επιχείρησης



Καταχώρηση Αγγελίας

Βασικά στοιχεία αγγελίας

Τίτλος *

Περιγραφή *

Προτιμήσεις

Τύπος απασχόλησης *

Τομέας απασχόλησης *

Ελάχιστη προϋπηρεσία

Να είναι ορατή;

ΑΚΥΡΩΣΗ ΥΠΟΒΟΛΗ

Εικόνα 5.7 Φόρμα προσθήκης αγγελίας

Τέλος, κάθε φορά που αποθηκεύεται μία αγγελία στο σύστημα, ενεργοποιείται μια αυτοματοποιημένη διαδικασία. Αυτή η διαδικασία ψάχνει και βρίσκει τα καταχωρημένα βιογραφικά στο σύστημα που ταιριάζουν με την αγγελία. Η αντιστοίχιση γίνεται βάση των κριτηρίων που περιγράφηκαν νωρίτερα, εξασφαλίζοντας έτσι ότι οι πιθανοί υποψήφιοι είναι οι καταλληλότεροι για τη θέση που ανακοινώνεται.

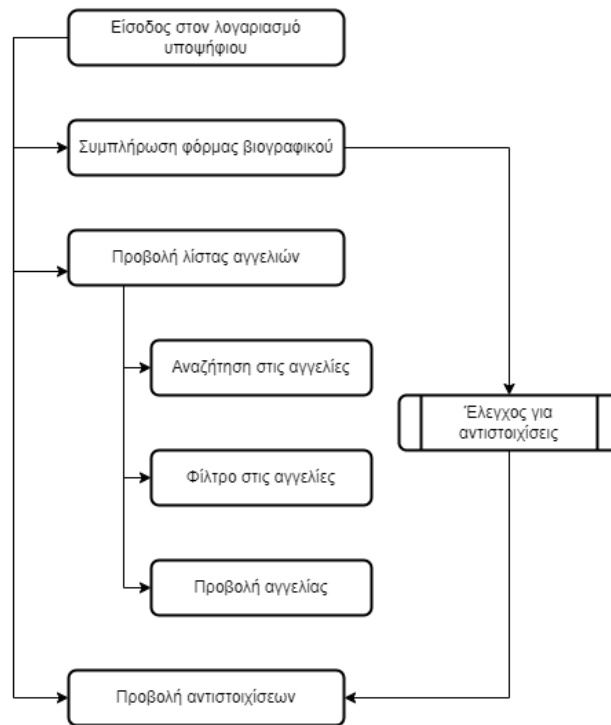
Ημερομηνία	Θέση εργασίας	Υποψήφιος
29/01/2024	Θέση εργασίας τέστ Λύκειο	Αλέξιος Ελληνίδης Πρώτο πτυχίο
29/01/2024	Άλλη μία Υποχρεωτική	Αλέξιος Ελληνίδης Πρώτο πτυχίο
29/01/2024	Άλλη μία Υποχρεωτική	Αλέξιος Ελληνίδης Πρώτο πτυχίο

Εικόνα 5.8 Αντιστοιχίσεις αγγελιών με βιογραφικά

Επιπλέον ο χρήστης, εάν βρεθεί βιογραφικό που να αντιστοιχεί με κάποια αγγελία, μπορεί να προβάλλει λεπτομέρειες για αυτό, ώστε να μπορέσει να έρθει σε επικοινωνία με τον υποψήφιο.

5.2.3 Χρήστης 'Υποψήφιος'

Όπως και ο χρήστης της κατηγορίας 'Επιχείρηση', έτσι και ο χρήστης της κατηγορίας 'Υποψήφιος' θα πρέπει να πραγματοποιήσει εγγραφή και στη συνέχεια να κάνει είσοδο στον προσωπικό του λογαριασμό στην εφαρμογή, ώστε να αποκτήσει πρόσβαση σε προσωποποιημένες λειτουργικότητες που προσφέρει η εφαρμογή. Συγκεκριμένα, μετά την εγγραφή και είσοδο του χρήστη, υπάρχει η δυνατότητα συμπλήρωσης μιας φόρμας με τα διάφορα στοιχεία του βιογραφικού του, όπως οι σπουδές που έχει παρακολουθήσει, τα στοιχεία επικοινωνίας του, κ.λπ. Αυτή η φόρμα περιλαμβάνει επίσης και επιλογές για την καθορισμό των κριτηρίων αντιστοίχισης με τις διάφορες αγγελίες που δημοσιεύονται στην πλατφόρμα. Όλες αυτές οι πληροφορίες υποβάλλονται στο σύστημα με σκοπό την αυτόματη αντιστοίχιση του χρήστη με τις αγγελίες που ταιριάζουν στα κριτήρια που έχει θέσει.



Εικόνα 5.9 Λειτουργικότητες χρήστη 'Υποψήφιος'



Καταχώρηση Βιογραφικού

Προσωπικά στοιχεία

Φωτογραφία Δεν επιλέχθηκε κανένα αρχείο.

Όνομα * Διεύθυνση *

Επίθετο * Τ.Κ. *

Τηλέφωνο Πόλη *

Κινητό Τηλέφωνο * Χώρα *

Επίπεδο εκπαίδευσης

Σημειώσεις

Προτιμήσεις

Τύπος απασχόλησης Περιοχή απασχόλησης


Τομέας απασχόλησης Γλώσσες επικοινωνίας

Να είναι ορατό;

ΥΠΟΒΟΛΗ

Εικόνα 5.10 Φόρμα βιογραφικού

Στην καρτέλα αγγελίες, ο χρήστης έχει τη δυνατότητα να προβάλει τη λίστα με όλες τις δημοσιευμένες αγγελίες. Πρόκειται για μια πλήρη επισκόπηση που προσφέρει μια γενική εικόνα των διαθέσιμων αγγελιών. Στην ίδια καρτέλα, αριστερά, ο χρήστης θα βρει μία φόρμα αναζήτησης - φιλτραρίσματος. Αυτή η φόρμα αναζήτησης είναι ένα εργαλείο που του επιτρέπει να προσαρμόσει την προβολή των αγγελιών με βάση συγκεκριμένα κριτήρια που επιλέγει ο ίδιος. Με αυτόν τον τρόπο, μπορεί να δει μόνο τις αγγελίες που τον ενδιαφέρουν πραγματικά.


Βιογραφικό Αγγελίες Αντιστοιχίσεις
MyCandidate ▾

Αγγελίες Εργασίας

Φίλτρα Αναζήτησης

Όρος αναζήτησης

Τύπος απασχόλησης
(Καμία επιλογή) ▾

Τομέας απασχόλησης
(Καμία επιλογή) ▾

Ελάχιστη προϋπηρεσία

Περιοχή απασχόλησης
(Καμία επιλογή) ▾

Επίπεδο εκπαίδευσης
(Καμία επιλογή) ▾

Γλώσσα επικοινωνίας
(Καμία επιλογή) ▾

ΕΥΡΕΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

29/01/2024
Θέση εργασίας τέστ

Testoupolis S.A.
Τομέας απασχόλησης: Αγροτικά
[Αλεξανδρούπολη](#)
🕒 Μερική Απασχόληση

29/01/2024
Θέση εργασίας τέστ 2

Testoupolis S.A.
Τομέας απασχόλησης: Logistics - Μεταφορές
[Αλεξανδρούπολη](#)
🕒 Πλήρης Απασχόληση

22/01/2024
Τέστ

Testoupolis S.A.
Τομέας απασχόλησης: Τουρισμός
[Αθήνα](#)
🕒 Πλήρης Απασχόληση

29/01/2024
Άλλη μία

Testoupolis S.A.
Τομέας απασχόλησης: Αγροτικά
[Αλεξανδρούπολη](#)
🕒 Μερική Απασχόληση

29/01/2024
Άλλη μία

Testoupolis S.A.
Τομέας απασχόλησης: Αγροτικά
[Αλεξανδρούπολη](#)
🕒 Μερική Απασχόληση

© 2024 Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων
Πτυχιακή Εργασία, Ελληνίδης Αλέξης | AM: 093447

Εικόνα 5.11 Λίστα αγγελιών - φόρμα φίλτρων



Λεπτομέρειες αγγελίας

Αγγελία από: Testoupolis S.A.
Θέση εργασίας τέστ
 29/01/2024

Τομέας απασχόλησης: Αγροτικά
 👤 Αλεξανδρούπολη
 🕒 Μερική Απασχόληση

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Προτιμήσεις
 Ελάχιστη προυπηρεσία: 2 έτη
 Επίπεδο εκπαίδευσης: Λύκειο
 Γλώσσες: Ελληνικά

Εικόνα 5.12 Προβολή λεπτομερειών αγγελίας

Τέλος, το σύστημα μας ανανεώνεται συνεχώς με νέες αγγελίες. Κατά συνέπεια, μπορεί να δημιουργηθούν νέες αντιστοιχίσεις με βάση τα κριτήρια του χρήστη. Αυτές οι νέες αντιστοιχίσεις θα εμφανιστούν αυτόματα στην αντίστοιχη καρτέλα, επιτρέποντας στον χρήστη να ενημερώνεται άμεσα για οποιαδήποτε νέα αγγελία που μπορεί να τον ενδιαφέρει.



Αντιστοιχίσεις με Αγγελίες Εργασίας

29/01/2024	<p>Θέση εργασίας τέστ Testoupolis S.A.</p>
29/01/2024	<p>Άλλη μία Testoupolis S.A.</p>
29/01/2024	<p>Άλλη μία Testoupolis S.A.</p>

Εικόνα 5.13 Λίστα αντιστοιχίσεων με αγγελίες

5.2.4 Χρήστης ‘Διαχειριστής’

Τέλος, προστέθηκε μια νέα ομάδα χρηστών στην εφαρμογή, οι διαχειριστές. Η πρόσβαση στη φόρμα σύνδεσης για το διαχειριστικό δεν είναι διαθέσιμη μέσω συνδέσμου από την αρχική σελίδα της εφαρμογής. Αντιθέτως, ο χρήστης που θα είναι διαχειριστής πρέπει να γνωρίζει το URI της φόρμας σύνδεσης, το οποίο είναι '/admin/login'. Ο ρόλος του διαχειριστή προστέθηκε στην εφαρμογή για να διαχειρίζεται τους εγγεγραμμένους χρήστες, αλλά στο μέλλον θα μπορούσαν να προστεθούν και άλλες λειτουργίες.

#	Χρήστης	Ομάδα	Ενεργός	Ενέργειες
1	MyCandidate Email: alexell2222@gmail.com	Υποψήφιος	✗	
2	MyCompany Email: alexios.ellinidis@gmail.com	Επιχείρηση	✓	
4	IHUAdmin Email: it093447@ihu.gr	Διαχειριστής	✓	

Εικόνα 5.14 Δυνατότητα διαχείρισης εγγεγραμμένων χρηστών

Κεφάλαιο 6ο: Συμπεράσματα – Προτάσεις βελτίωσης

6.1 Συμπεράσματα

Για την υλοποίηση της παρούσας πτυχιακής εργασίας, χρησιμοποιήθηκαν κύριες τεχνολογίες web, όπως HTML, CSS και JavaScript. Αυτές οι τεχνολογίες χρησιμοποιήθηκαν για τη δημιουργία του επιπέδου παρουσίασης. Πιο συγκεκριμένα, το HTML χρησιμοποιήθηκε για τη δομή του ιστότοπου, το CSS για τον οπτικό σχεδιασμό και το JavaScript για την προσθήκη διαδραστικότητας.

Στο επίπεδο της επιχειρησιακής λογικής, επιλέχθηκε ο συνδυασμός PHP-MySQL. Η PHP χρησιμοποιήθηκε για την επεξεργασία των δεδομένων και την επικοινωνία με τη βάση δεδομένων, ενώ η MySQL χρησιμοποιήθηκε για την αποθήκευση και ανάκτηση των δεδομένων.

Επιπλέον, η μελέτη των διάφορων χρήσιμων εργαλείων ανάπτυξης, όπως το Git για τη διαχείριση των εκδόσεων του κώδικα και το VSCode για την επεξεργασία του κώδικα, σε συνδυασμό με την εφαρμογή του προτύπου MVC (Model-View-Controller), συνέβαλαν σημαντικά στην ανάπτυξη μίας διαδραστικής web εφαρμογής.

6.2 Προτάσεις βελτίωσης

Από την σκοπιά του προγραμματιστή θα μπορούσε να εισαχθεί η χρήση Docker [23]. Το Docker είναι ένα ανοιχτού κώδικα εργαλείο, το οποίο λειτουργεί ως ένα εργαλείο που δημιουργεί containers. Αυτοί οι containers επιτρέπουν στους προγραμματιστές να πακετάρουν μια εφαρμογή με όλα τα μέρη που χρειάζεται, όπως βιβλιοθήκες και άλλες εξαρτήσεις, και να την μεταφέρουν ως ένα ολοκληρωμένο πακέτο. Αυτό σημαίνει ότι οι εφαρμογές θα λειτουργούν ομαλά ανεξάρτητα από το περιβάλλον όπου εκτελούνται. Το Docker μειώνει επίσης τον χρόνο ανάπτυξης καθώς επιτρέπει στους προγραμματιστές να επαναχρησιμοποιούν containers για νέες εφαρμογές και να αυτοματοποιούν τις διαδικασίες, όπως η εγκατάσταση και η διαμόρφωση. Το Docker διασφαλίζει ότι οι εφαρμογές θα λειτουργήσουν σωστά σε οποιοδήποτε περιβάλλον, εξαλείφοντας τα προβλήματα συμβατότητας.

Τέλος, το Docker μπορεί να βελτιώσει την παραγωγικότητα της ομάδας ανάπτυξης, καθώς επιτρέπει την ευκολότερη και γρηγορότερη επαναχρησιμοποίηση του κώδικα, τη μείωση των πιθανών σφαλμάτων και τη βελτίωση της συνεργασίας μεταξύ των μελών της ομάδας.

Από τη σκοπιά του τελικού χρήστη και της γενικότερης λειτουργικότητας της εφαρμογής, θα μπορούσε να μελετηθεί εκτενέστερα ο κύκλος ζωής των κυρίως αλληλοεπιδρώντων οντοτήτων, της αγγελίας και του βιογραφικού. Αυτό θα περιλάμβανε μια πιο αναλυτική εξέταση των διαφόρων σεναρίων που μπορεί να προκύψουν, όπως για παράδειγμα τι γίνεται στην περίπτωση που μία θέση εργασίας έχει καλυφθεί, ή στην περίπτωση που ένας υποψήφιος ενημερώσει με νέα στοιχεία το βιογραφικό του.

Επίσης, θα ήταν εξαιρετικά χρήσιμο να προστεθεί ένας μηχανισμός ειδοποίησης που θα ενημερώνει τους ενδιαφερόμενους με κάθε νέα αντιστοίχιση βιογραφικού με αγγελία. Επιπλέον, θα ήταν επωφελές να προστεθούν νέα κριτήρια στην εφαρμογή, όπως για παράδειγμα τα έτη προϋπηρεσίας του υποψηφίου, που θα μπορούσαν να επηρεάσουν την τελική αντιστοίχιση.

Κεφάλαιο 6

Τέλος, είναι σημαντικό θα ήταν να προστεθούν και άλλες δυνατότητες στον διαχειριστή του συστήματος. Αυτός, εκτός από τη διαχείριση χρηστών, θα μπορούσε να ρυθμίζει τις τιμές που έχουν τα διάφορα κριτήρια, όπως η προσθήκη μίας νέας περιοχής ή μίας γλώσσας, καθώς και να εξάγει χρήσιμα στατιστικά δεδομένα για την αγορά εργασίας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] «php.net,» [Ηλεκτρονικό]. Available: <https://www.php.net/manual/en/history.php.php>. [Πρόσβαση 2024].
- [2] M. Kherlakian, «zend.com,» [Ηλεκτρονικό]. Available: <https://www.zend.com/blog/whats-new-php-74>. [Πρόσβαση 2024].
- [3] «MDN Docs - HTML Basics,» [Ηλεκτρονικό]. Available: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/HTML_basics. [Πρόσβαση 2024].
- [4] «MDN Docs - CSS Basics,» [Ηλεκτρονικό]. Available: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/CSS_basics. [Πρόσβαση 2024].
- [5] «W3C,» [Ηλεκτρονικό]. Available: <https://www.w3.org/TR/CSS-access/>. [Πρόσβαση 2024].
- [6] «Web Technology Surveys,» [Ηλεκτρονικό]. Available: https://w3techs.com/technologies/overview/css_framework. [Πρόσβαση 2024].
- [7] «VSCode,» [Ηλεκτρονικό]. Available: <https://code.visualstudio.com/>. [Πρόσβαση 2024].
- [8] «Git,» [Ηλεκτρονικό]. Available: <https://git-scm.com/about>. [Πρόσβαση 2024].
- [9] «Composer Documentation,» [Ηλεκτρονικό]. Available: <https://getcomposer.org/doc/>. [Πρόσβαση 2024].
- [10] «Wikipedia/ModelViewController,» [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>. [Πρόσβαση 2024].
- [11] «Wikipedia/CodeIgniter,» [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/CodeIgniter>. [Πρόσβαση 2024].
- [12] «Codeigniter Change Logs,» [Ηλεκτρονικό]. Available: https://codeigniter.com/user_guide/changelogs/index.html. [Πρόσβαση 2024].
- [13] «Wikipedia/Open closed Principle,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Open%E2%80%93closed_principle. [Πρόσβαση 2024].
- [14] Martin Fowler, «Martin Fowler - Front Controller,» [Ηλεκτρονικό]. Available: <https://martinfowler.com/eaCatalog/frontController.html>. [Πρόσβαση 2004].
- [15] «MDN Docs - HTTP Overview,» [Ηλεκτρονικό]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>. [Πρόσβαση 2024].
- [16] «Codeigniter HTTP Requests,» [Ηλεκτρονικό]. Available: https://codeigniter.com/user_guide/concepts/http.html. [Πρόσβαση 2024].

- [17] «Codeigniter Userguide,» [Ηλεκτρονικό]. Available: <https://codeigniter.com/userguide3/overview/appflow.html>. [Πρόσβαση 2024].
- [18] «Codeigniter Installation,» [Ηλεκτρονικό]. Available: https://www.codeigniter.com/user_guide/installation/running.html. [Πρόσβαση 2024].
- [19] «OWASP Top Ten,» [Ηλεκτρονικό]. Available: <https://owasp.org/www-project-top-ten/>. [Πρόσβαση 2024].
- [20] «Codeigniter Security,» [Ηλεκτρονικό]. Available: https://codeigniter.com/user_guide/concepts/security.html. [Πρόσβαση 2024].
- [21] «Codeigniter Logging,» [Ηλεκτρονικό]. Available: https://codeigniter.com/user_guide/general/logging.html. [Πρόσβαση 2024].
- [22] «Codeigniter Shield Auth,» [Ηλεκτρονικό]. Available: <https://shield.codeigniter.com/>. [Πρόσβαση 2024].
- [23] «Docker Docs,» [Ηλεκτρονικό]. Available: <https://docs.docker.com/>. [Πρόσβαση 2024].
- [24] «mysql.com,» [Ηλεκτρονικό]. Available: <https://www.mysql.com/why-mysql/benchmarks/>. [Πρόσβαση 2024].
- [25] «LinkedIn,» [Ηλεκτρονικό]. Available: <https://www.linkedin.com/pulse/what-php-framework-reza-safari/>. [Πρόσβαση 2024].
- [26] «Laminas Project,» [Ηλεκτρονικό]. Available: <https://getlaminas.org/>. [Πρόσβαση 2024].
- [27] «Symfony,» [Ηλεκτρονικό]. Available: <https://symfony.com/what-is-symfony>. [Πρόσβαση 2024].
- [28] «Yii Framework,» [Ηλεκτρονικό]. Available: <https://www.yiiframework.com/doc/guide/2.0/en/intro-yii>. [Πρόσβαση 2024].
- [29] «Wikipedia/HTML,» [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/HTML>. [Πρόσβαση 2024].
- [30] «Wikipedia/Bootstrap_(front-end-framework),» [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)). [Πρόσβαση 2024].
- [31] «Wikipedia/MySQL,» [Ηλεκτρονικό]. Available: <https://en.wikipedia.org/wiki/MySQL>. [Πρόσβαση 2024].
- [32] «Wikipedia/Visual_Studio_Code,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Visual_Studio_Code. [Πρόσβαση 2024].
- [33] «MDN Docs - JavaScript,» [Ηλεκτρονικό]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [Πρόσβαση 2024].
- [34] «Codeigniter Controllers,» [Ηλεκτρονικό]. Available: https://codeigniter.com/user_guide/incoming/controllers.html. [Πρόσβαση 2024].

- [35] «Codeigniter Building Responses,» [Ηλεκτρονικό]. Available:
https://codeigniter.com/user_guide/outgoing/index.html. [Πρόσβαση 2024].
- [36] «Codeigniter Query Builder,» [Ηλεκτρονικό]. Available:
https://codeigniter.com/user_guide/database/query_builder.html. [Πρόσβαση 2024].
- [37] «Codeigniter Spark CLI,» [Ηλεκτρονικό]. Available:
https://codeigniter.com/user_guide/cli/index.html. [Πρόσβαση 2024].
- [38] «Codeigniter Managing Databases,» [Ηλεκτρονικό]. Available:
https://codeigniter.com/user_guide/dbmgmt/index.html. [Πρόσβαση 2024].
- [39] Sadika, «Medium.com,» [Ηλεκτρονικό]. Available:
<https://medium.com/@sadikarahmantanisha/the-mvc-architecture-97d47e071eb2>. [Πρόσβαση 2024].
- [40] «Wikibooks/ComputerScienceDesignPatterns,» [Ηλεκτρονικό]. Available:
https://en.wikibooks.org/wiki/Computer_Science_Design_Patterns/Model%E2%80%93View_Controller. [Πρόσβαση 2024].
- [41] Martin Fowler, «Martin Fowler - ModelViewController,» [Ηλεκτρονικό]. Available:
<https://martinfowler.com/eaDev/uiArchs.html#ModelViewController>. [Πρόσβαση 2024].

