



ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ScholarPath: Διαδικτυακή εφαρμογή
διαχείρισης αιτήσεων εισαγωγής σε
Προγράμματα Μεταπτυχιακών Σπουδών**

Της φοιτήτριας

Αθανασία-Ελένη Σαμσούρη

Αρ. Μητρώου: 154533

Επιβλέπων

Στέφανος Ουγιάρογλου

Επικούρος Καθηγητής

10 Σεπτεμβρίου 2023

Title of Dissertation ScholarPath: Διαδικτυακή εφαρμογή διαχείρισης αιτήσεων εισαγωγής σε
Προγράμματα Μεταπτυχιακών Σπουδών
Code of Dissertation 21170
Student's full name Athanasia Eleni Samsouri
Supervisor's full name Stefanos Ougiaroglou
Date of undertaking 15-10-2021
Date of completion 10-09-2023

We hereby affirm the authorship of this paper as well as the acknowledgement and credit of whichever assistance We received in its composition. We have, furthermore, noted the various sources from which We extracted data, ideas, visual or written material, in paraphrase or exact quotation. Moreover, we affirm the exclusive composition of this paper by myself only, for the purpose of it being a dissertation, in the Department of Information and Electronic Engineering of the I.H.U.

This paper constitutes the intellectual property of Athanasia Eleni Samsouri, the student that composed it. According to the open-access policy, the author/composer offers the International Hellenic University authorisation to use the right to reproduce, borrow, publicly present and digitally distribute the paper globally, in electronic form and media of all kinds, for teaching or research purposes, voluntarily. Open access to the full text, by no means grants the right to trespass the intellectual property of the author/composer, nor does it authorise the reproduction, republication, duplication, selling, commercial use, distribution, publication, downloading, uploading, translation, modification of any kind, in part or summary of the paper, without the explicit written consent of the authors.

The approval of this dissertation by the Department of Information and Electronic Engineering of the International Hellenic University, does not necessarily entail the adoption of the author's views, on behalf of the Department.

Αφιέρωση

Αυτή η εργασία αφιερώνεται στους καθηγητές μου, Στέφανο Ουγιάρογλου και Κωνσταντίνο Διαμαντάρα, για την ανεκτίμητη καθοδήγησή τους και την υποστήριξή τους σε αυτήν την μελέτη.

Επίσης, την αφιερώνω στην αγαπημένη μου οικογένεια, που με υποστήριξε σε όλα τα φοιτητικά μου χρόνια και με κίνησε προς την επίτευξη αυτού του στόχου.

Περίληψη

Η παρούσα μελέτη επικεντρώνεται στο ScholarPath, μια εξαιρετικά χρήσιμη πλατφόρμα ανοιχτού λογισμικού που στοχεύει στον τομέα της διαχείρισης των αιτήσεων για μεταπτυχιακά προγράμματα σπουδών στην Ελλάδα. Αντιμετωπίζοντας το πρόβλημα της έλλειψης μιας κοινής πλατφόρμας και της αυξημένης γραφειοκρατίας που επηρεάζει τόσο τους υποψήφιους φοιτητές όσο και τους καθηγητές, το ScholarPath αναδεικνύεται ως ένα ενδιαφέρον εργαλείο με διάφορες πτυχές που αξίζει να εξεταστούν περαιτέρω.

Σε βάθος ανάλυσης, παρουσιάζονται λεπτομερώς οι λειτουργίες του ScholarPath, με έμφαση στον τρόπο με τον οποίο επιτρέπει στους πανεπιστημιακούς φορείς να διαχειρίζονται αποτελεσματικά τα μεταπτυχιακά προγράμματα σπουδών. Επιπλέον, αναδεικνύεται η αξία της διαφάνειας που προσφέρει στη διαδικασία αξιολόγησης των αιτήσεων από τους καθηγητές, καθώς και την αποτελεσματική διαχείριση των προγραμμάτων σπουδών από τα πανεπιστήμια. Επίσης, παρουσιάζονται οι τεχνολογίες που χρησιμοποιούνται στην πλατφόρμα, καθώς και οι λειτουργικές απαιτήσεις της εφαρμογής.

Με το ScholarPath, πανεπιστημιακές αρχές μπορούν να απλοποιήσουν τις διοικητικές διαδικασίες και να εξαλείψουν τη γραφειοκρατία που επηρεάζει τους υποψήφιους φοιτητές και τους καθηγητές. Αυτό έχει ως αποτέλεσμα τη μείωση του χρόνου που απαιτείται για τη διαχείριση των μεταπτυχιακών προγραμμάτων και των αιτήσεων. Με την αυτοματοποίηση των λειτουργιών του και τη διασφάλιση της διαφάνειας, προσφέρει μια ολοκληρωμένη λύση στο πρόβλημα της μεταπτυχιακής εισαγωγής. Βοηθάει σημαντικά στην αποτελεσματική διαχείριση των μεταπτυχιακών προγραμμάτων σπουδών και στην απλοποίηση της διαδικασίας για τους ενδιαφερόμενους. Εν κατακλείδι, το ScholarPath αποτελεί ένα ανεκτίμητο εργαλείο για την ακαδημαϊκή κοινότητα στην Ελλάδα, αποδεικνύοντας πως η τεχνολογία μπορεί να ενισχύσει την αποτελεσματικότητα και τη διαφάνεια στον τομέα της μεταπτυχιακής εκπαίδευσης, επιφέροντας απτά οφέλη τόσο στους υποψήφιους φοιτητές όσο και στους πανεπιστημιακούς φορείς.

Abstract

The present study focuses on Scholar-Path, an extremely useful open-source platform aimed at managing applications for postgraduate programs in Greece. Addressing the issue of the lack of a common platform and the increased bureaucracy that affects both prospective students and professors, ScholarPath emerges as an intriguing tool with various aspects worth further examination.

In-depth analysis presents the functions of ScholarPath in detail, emphasizing how it enables academic institutions to efficiently manage postgraduate study programs. Furthermore, it highlights the value of transparency it offers in the evaluation process of applications by professors, as well as the effective management of study programs by universities. The technologies used in the platform and the functional requirements of the application are also presented.

With ScholarPath, academic authorities can streamline administrative procedures and eliminate bureaucracy that affects prospective students and professors. This results in a reduction in the time required for managing postgraduate programs and applications. Through the automation of its functions and the assurance of transparency, ScholarPath provides a comprehensive solution to the issue of postgraduate admissions. It significantly contributes to the effective management of postgraduate study programs and simplifies the process for those interested. In conclusion, ScholarPath is an invaluable tool for the academic community in Greece, demonstrating how technology can enhance efficiency and transparency in the field of postgraduate education, bringing tangible benefits to both prospective students and academic institutions.

Περιεχόμενα

1	Εισαγωγή	2
1.1	Προγράμματα Μεταπτυχιακών Σπουδών στην Ελλάδα	2
1.2	Διαδικασία αιτήσεων εισαγωγής σε ΠΜΣ	3
1.3	Κίνητρο	4
1.4	Συνεισφορά	5
1.5	Οργάνωση της εργασίας	7
2	Τεχνολογίες	9
2.1	JavaScript	9
2.2	TypeScript	11
2.3	Node.js	12
2.3.1	Node Modules και Node Package Manager	12
2.3.2	Event Loop	13
2.3.3	Koa.js	14
2.4	GraphQL	15
2.4.1	TypeGraphQL	16
2.4.2	Resolvers	16
2.4.3	Queries	17
2.4.4	Mutations	17
2.4.5	GraphQL Playground	18
2.5	Vue.js	19
2.5.1	Οργάνωση Αρχείων	19
2.5.2	Composition API	20
2.6	Quasar Framework	21
2.7	MySQL	22
2.7.1	MikroORM	24
3	Σχεδιασμός και Υλοποίηση του ScholarPath	26
3.1	Αρχιτεκτονική	26

3.2	Σχεδίαση Βάσης Δεδομένων	27
3.2.1	Πρόγραμμα Σπουδών (Course Program)	27
3.2.2	Κάλεσμα για Αιτήσεις (Call For Submissions)	27
3.2.3	Υποψήφιος (Candidate)	29
3.2.4	Συστατική Επιστολή (Reference)	29
3.2.5	Χρήστης (User)	30
3.2.6	Ρόλοι (Roles)	30
3.2.7	Πρόσκληση (Invitation)	31
3.3	Λειτουργικές Απαιτήσεις	31
3.3.1	Διαχειριστές (Admins)	32
3.3.2	Συντονιστές (Moderators)	34
3.3.3	Υποψήφιοι (Candidates)	35
3.3.4	Κλήση με Email (Invited)	36
3.4	Υλοποίηση Backend	37
3.5	Υλοποίηση Frontend	44
3.6	Ο κώδικας του ScholarPath στο Github	50
4	Παρουσίαση του ScholarPath	52
4.1	Σενάριο χρήσης υπόβολής αίτησης	52
4.2	Σενάριο χρήσης δημιουργίας Προγράμματος Σπουδών	55
4.3	Σενάριο χρήσης για δημιουργία CFS	58
4.4	Σενάριο χρήσης για αξιολόγηση	62
4.5	Σενάριο αποδοχής Υποψηφίων	64
4.6	Σενάριο χρήσης σύνταξης και αποστολής Συστατικής Επιστολής	68
4.7	Σενάριο χρήσης διαχείρισης χρηστών	70
5	Συμπεράσματα και Μελλοντικές επεκτάσεις	73
5.1	Συμπεράσματα	73
5.2	Μελλοντικές Επεκτάσεις	74

Κεφάλαιο 1

Εισαγωγή

1.1 Προγράμματα Μεταπτυχιακών Σπουδών στην Ελλάδα

Η Ελλάδα προσφέρει μια πλούσια ποικιλία από Προγράμματα Μεταπτυχιακών Σπουδών (ΠΜΣ) και εξειδικεύσεων που καλύπτουν τις σύγχρονες ανάγκες της εκπαίδευσης και της αγοράς εργασίας. Η ποιότητα της εκπαίδευσης που παρέχεται από τα ελληνικά πανεπιστημιακά ιδρύματα αποτελεί στρατηγικό πλεονέκτημα για την επιλογή της Ελλάδας ως προορισμού μεταπτυχιακών σπουδών.

Οι φοιτητές που επιλέγουν ένα ΠΜΣ στην Ελλάδα έχουν την ευκαιρία να βαθύνουν σε συγκεκριμένους τομείς και να αποκτήσουν εξειδίκευση που θα τους διαφοροποιήσει στην αγορά εργασίας. Η συνεργασία με καθηγητές και ερευνητές που διακρίνονται για την εμπειρία και τις δημοσιεύσεις τους δίνει τη δυνατότητα στους φοιτητές να αντλήσουν γνώσεις από το πραγματικό πεδίο της έρευνας και να αναπτύξουν κριτική σκέψη και αναλυτικές δεξιότητες.

Οι προοπτικές για διεξαγωγή ερευνητικών έργων εντός των ΠΜΣ αποτελούν σημαντικό πλεονέκτημα. Μέσα από τις διπλωματικές εργασίες και τις συμμετοχές σε ερευνητικά προγράμματα, οι φοιτητές έχουν την ευκαιρία να εφαρμόσουν τις γνώσεις τους σε πραγματικά προβλήματα και να συνεισφέρουν στην επιστημονική κοινότητα.

Επίσης, η διεθνής προσέγγιση των ΠΜΣ στην Ελλάδα διευρύνει το ορίζοντα των φοιτητών. Η συνεργασία με διεθνείς φορείς, οι διεθνείς διαλέξεις και σεμινάρια καθώς και η παροχή ευκαιριών για διεθνείς ερευνητικές συνεργασίες εμπλουτίζουν το ακαδημαϊκό και προσωπικό τους προφίλ.

Όλα αυτά συνδυάζονται με την πλούσια πολιτιστική κληρονομιά της Ελλάδας, δημιουργώντας ένα περιβάλλον που προάγει τη διαπολιτισμική κατανόηση και ανταλλαγή απόψεων.

Συνοψίζοντας, η Ελλάδα αναδεικνύεται ως μια πλούσια επιλογή για τη συνέχιση μεταπτυχιακών σπουδών. Οι ΠΜΣ που προσφέρονται στη χώρα συνδυάζουν την υψηλή ποιότητα της εκπαίδευσης, την ευρεία γνώση, την έμφαση στην έρευνα, τη διεθνή διάσταση και την πολιτιστική εμπειρία, προσφέροντας μια ολοκληρωμένη εκπαιδευτική εμπειρία για τους μελλοντι-

κούς επαγγελματίες και ερευνητές.

1.2 Διαδικασία αιτήσεων εισαγωγής σε ΠΜΣ

Η διαδικασία αιτήσεων εισαγωγής σε Προγράμματα Μεταπτυχιακών Σπουδών στην Ελλάδα ακολουθεί ένα σταθερό μονοπάτι, διακριτό από συντονισμένες φάσεις. Κάθε υποψήφιος πρέπει να πληροφορηθεί την οριζόντια αυτή πορεία, που ανοίγει τον δρόμο προς την ακαδημαϊκή του εξέλιξη.

Καταρχάς, οι ενδιαφερόμενοι υποβάλλουν αιτήσεις για εισαγωγή σε ΠΜΣ της επιλογής τους. Κατά τη διάρκεια αυτής της φάσης, συμπληρώνουν τα αναγκαία στοιχεία και προσκομίζουν τα απαραίτητα δικαιολογητικά.

Στη συνέχεια, οι εκπρόσωποι του ΠΜΣ αξιολογούν τις υποβληθείσες αιτήσεις. Κριτήρια όπως το ακαδημαϊκό υπόβαθρο, η επαγγελματική εμπειρία, αλλά και άλλες σχετικές δεξιότητες, λαμβάνονται υπόψη κατά την αξιολόγηση.

Εάν απαιτείται, ορισμένοι οι υποψήφιοι ενδέχεται να κληθούν σε προσωπική συνέντευξη ως μέρος της αξιολόγησής τους. Αυτή η διαδικασία επιτρέπει στους αξιολογητές να αξιολογήσουν προσωπικές δεξιότητες, προσαρμογή σε εξειδικευμένα προγράμματα, εμπειρία και στόχους των υποψηφίων. Επιπλέον, βοηθά στον προσδιορισμό της αρμονίας με το πρόγραμμα και την αντίληψη για το πώς μπορούν να συμβάλλουν σε αυτό.

Ολοκληρωθείσας της αξιολόγησης, ανακοινώνονται τα αποτελέσματα, αναδεικνύοντας τους υποψηφίους που θα εισαχθούν στο πρόγραμμα. Είναι σημαντικό να σημειωθεί ότι κάθε ΠΜΣ διαθέτει προκαθορισμένο αριθμό διαθέσιμων θέσεων. Οι υποψήφιοι που δεν επιλέγονται για την αρχική εισαγωγή εντάσσονται σε μια λίστα αναμονής. Σε περίπτωση που κάποιος από τους θετικά ανακοινωθέντες υποψηφίους δεν αποδεχτεί την προσφερθείσα θέση, οι υποψήφιοι από τη λίστα αναμονής μπορεί να κληθούν να συμπληρώσουν το κενό, ανοίγοντας έτσι νέες προοπτικές για αυτούς.

Τέλος, οι επιλεγμένοι υποψήφιοι διεκπεραιώνουν τη διαδικασία εγγραφής και λαμβάνουν αναλυτικές πληροφορίες για το ΠΜΣ, το πρόγραμμα σπουδών και άλλες απαραίτητες λεπτομέρειες.

Σημαντικό να σημειωθεί ότι κάθε ΠΜΣ μπορεί να έχει διαφορετικά κριτήρια και διαδικασίες, ανάλογα με τον τομέα σπουδών και το περιεχόμενο του προγράμματος. Επίσης, το χρονοδιάγραμμα για την υποβολή αιτήσεων, την αξιολόγηση και την ανακοίνωση αποτελεσμάτων επηρεάζεται από τις πολιτικές και τις προθεσμίες του κάθε ΠΜΣ και πανεπιστημίου.

1.3 Κίνητρο

Όπως αναφέρθηκε νωρίτερα, η Ελλάδα αποτελεί μια εξαιρετική επιλογή για όσους επιθυμούν να συνεχίσουν τις σπουδές τους σε επίπεδο μεταπτυχιακών σπουδών. Η ποικιλία των πανεπιστημίων και των προγραμμάτων που προσφέρουν αποτελεί ένα μεγάλο πλεονέκτημα, αλλά ταυτόχρονα μπορεί να δημιουργήσει μια πρόκληση για όσους αναζητούν την κατάλληλη επιλογή που θα ταιριάζει απόλυτα στις ανάγκες και τις φιλοδοξίες τους.

Οι ενδιαφερόμενοι προτού προχωρήσουν στην αίτηση για μεταπτυχιακά προγράμματα, θα πρέπει να διερευνήσουν προσεκτικά τις διάφορες επιλογές που προσφέρονται από κάθε πανεπιστήμιο. Αυτό σημαίνει ότι θα πρέπει να επισκεφτούν τις ιστοσελίδες των πανεπιστημίων και να διαβάσουν προσεκτικά τις περιγραφές των προγραμμάτων. Κάθε πρόγραμμα συνοδεύεται από λεπτομερείς πληροφορίες για το περιεχόμενο των μαθημάτων, τις απαιτήσεις εισαγωγής, τις διαδικασίες επιλογής, και τους διδάσκοντες.

Επιπλέον, είναι σημαντικό να εξετάσουν το φάσμα των ειδίκευσεων που προσφέρονται σε κάθε πρόγραμμα. Αναλύοντας τις επιλογές ειδίκευσης, οι υποψήφιοι θα μπορούν να προσδιορίσουν αν οι κατευθύνσεις που τους ενδιαφέρουν επικαλύπτονται από τα πεδία που εξειδικεύονται τα προγράμματα.

Η αναζήτηση για το κατάλληλο πρόγραμμα πρέπει να περιλαμβάνει και την εξέταση της φήμης του πανεπιστημίου στον τομέα της ειδίκευσης που ενδιαφέρεται ο υποψήφιος. Οι καθηγητές και οι συνεργάτες ενός προγράμματος μπορεί να έχουν σημαντική επίδραση στην ποιότητα της εκπαίδευσης και της έρευνας που παρέχεται.

Τέλος, η διαδικασία αποφυγής της βιαστικής επιλογής είναι ουσιώδης. Οι υποψήφιοι θα πρέπει να επιδιώκουν να αξιολογήσουν προσεκτικά όλες τις πληροφορίες που έχουν συγκεντρώσει, να συγκρίνουν τα προγράμματα μεταξύ τους και να επιλέξουν αυτό που ταιριάζει καλύτερα στους στόχους και τις φιλοδοξίες τους.

Αυτό το πρόβλημα μπορεί να διαπιστωθεί και μέσα στο Διεθνές Πανεπιστήμιο Ελλάδος (Δι.Πα.Ε.). Οι πολλές επιλογές σε διάφορα τμήματα, όπως Μηχανικών, Διοίκησης, Γεωπονίας, Υγείας και πολλά άλλα, προσφέρουν μια ευρεία γκάμα επιλογών για τους φοιτητές που επιθυμούν να συνεχίσουν τις σπουδές τους.

Ωστόσο, με αυτήν την ποικιλία έρχεται και η πρόκληση της διαφορετικής διαδικασίας αίτησης για κάθε Πρόγραμμα Μεταπτυχιακών Σπουδών. Η διαφορετική πλατφόρμα αιτήσεων για κάθε ΠΜΣ, ακόμη και εντός των διαφορετικών τμημάτων, μπορεί να προκαλέσει σύγχυση στους ενδιαφερόμενους. Κάθε πλατφόρμα μπορεί να έχει διαφορετική δομή, απαιτήσεις και προθεσμίες αίτησης, δημιουργώντας έτσι έναν περιορισμό στην ενοποίηση της πληροφορίας για την εύρεση του κατάλληλου προγράμματος.

Επιπλέον, η έλλειψη μιας κοινής γραμμής και των συγκεντρωμένων πληροφοριών μπορεί να καθιστά τη διαδικασία αναζήτησης και αίτησης πιο χρονοβόρα και περίπλοκη για τους υποψηφίους. Είναι σημαντικό για τους ενδιαφερόμενους να διαθέτουν αρκετό χρόνο για να ερευ-

νήσουν και να κατανοήσουν τις απαιτήσεις κάθε προγράμματος που τους ενδιαφέρει. Προκειμένου να αντιμετωπιστούν αυτές οι προκλήσεις, ήταν αναγκαίο να εξεταστούν λύσεις που θα διευκολύνουν τη διαδικασία αναζήτησης και πρόσβασης στις πληροφορίες για τα Προγράμματα Μεταπτυχιακών Σπουδών του ΔΙ.ΠΑ.Ε. Μια ολοκληρωμένη και κεντρική πλατφόρμα, που θα συγκεντρώνει όλες τις απαραίτητες πληροφορίες για τα προγράμματα, θα διευκολύνει τους υποψηφίους να ενημερωθούν αποτελεσματικά και θα εξοικονομήσει χρόνο και προσπάθεια. Επιπλέον, μια τέτοια πλατφόρμα θα προσφέρει συνέπεια και σαφήνεια στη διαδικασία ενημέρωσης, διευκολύνοντας τη λήψη της απαραίτητης απόφασης για το επόμενο εκπαιδευτικό βήμα των υποψηφίων.

1.4 Συνεισφορά

Στο πλαίσιο της συγκεκριμένης πτυχιακής, έχει αναπτυχθεί μια εφαρμογή ονομασμένη ScholarPath που αντιμετωπίζει αποτελεσματικά το πρόβλημα της διάχυτης πληροφορίας που σχετίζεται με τα διάφορα προγράμματα μεταπτυχιακών σπουδών. Η εν λόγω εφαρμογή αντικατοπτρίζει την ανάγκη για αποτελεσματικότερη διαχείριση των πολλαπλών διαθέσιμων προγραμμάτων, παρέχοντας μια ολοκληρωμένη λύση σε έναν σύνθετο τομέα.

Ένα αξιοσημείωτο στοιχείο αυτής του ScholarPath είναι η πολυμορφία της χρήσης του. Διάφορα τμήματα και πανεπιστημιακά ιδρύματα μπορούν να ωφεληθούν από τις λειτουργίες της εφαρμογής, προσφέροντας μια ενιαία και ολοκληρωμένη πηγή πληροφοριών για τα διάφορα ΠΜΣ. Η open source φύση της εφαρμογής επιτρέπει στα πανεπιστήμια να εγκαταστήσουν το ScholarPath στους δικούς τους διακομιστές και να διαχειρίζονται εύκολα τα μεταπτυχιακά προγράμματα. Με αυτόν τον τρόπο, δημιουργείται ένα ενιαίο σημείο αναφοράς που συμβάλλει στην ευκολότερη πρόσβαση στις πληροφορίες και στην αποφυγή της σύγχυσης που μπορεί να προκύψει από τη διάσπαση των πληροφοριών.

Συγκεκριμένα, η εν λόγω εφαρμογή παρέχει μια εξειδικευμένη και πλήρη πλατφόρμα για τη διαχείριση μεταπτυχιακών προγραμμάτων σπουδών. Μέσω αυτής, οι υπεύθυνοι των ΠΜΣ έχουν την ευκαιρία να δημιουργούν και να διαχειρίζονται εξ ολοκλήρου τα προγράμματα σπουδών τους. Αυτό περιλαμβάνει την δυνατότητα δημιουργίας, επεξεργασίας και ενημέρωσης των αναλυτικών περιεχομένων των προγραμμάτων, καθώς και την καθοριστική διαδικασία της επιλογής των κριτηρίων αξιολόγησης για κάθε πρόγραμμα.

Ενδεικτικά, οι διευθυντές ΠΜΣ έχουν την δυνατότητα να επιλέγουν τα κριτήρια με βάση τα οποία θα αξιολογηθούν οι αιτήσεις υποψηφίων σπουδαστών. Αυτό συνιστά ένα σημαντικό μέρος της διαδικασίας, καθώς επιτρέπει την προσαρμογή της αξιολόγησης στις συγκεκριμένες απαιτήσεις και προδιαγραφές του κάθε προγράμματος. Επιπλέον, η εφαρμογή παρέχει τη δυνατότητα πρόσκλησης νέων χρηστών, που σε αυτήν την περίπτωση είναι καθηγητές που θα συμμετάσχουν στη διαδικασία αξιολόγησης των αιτήσεων.

Η εφαρμογή, επίσης, επιτρέπει την καθορισμένη περίοδο κατά την οποία το κάθε ΠΜΣ θα δέχεται αιτήσεις από υποψηφίους σπουδαστές. Κατά τη διάρκεια αυτής της χρονικής περιόδου, οι υποψήφιοι έχουν τη δυνατότητα να συμπληρώνουν τα απαραίτητα στοιχεία τους και να υποβάλλουν τις αιτήσεις τους για συμμετοχή στο εκάστοτε πρόγραμμα σπουδών.

Από την πλευρά των υποψηφίων, παρέχεται η δυνατότητα σε καθέναν από αυτούς να αναζητήσει τα διαθέσιμα Προγράμματα Μεταπτυχιακών Σπουδών μέσω της εφαρμογής και να περιηγηθεί σε αυτά. Με την ανακάλυψη ενός ΠΜΣ που τραβά το ενδιαφέρον τους, εφόσον το πρόγραμμα είναι ανοιχτό για υποβολή αιτήσεων, έχουν τη δυνατότητα να συμπληρώσουν την προκαθορισμένη φόρμα αίτησης με τα προσωπικά τους στοιχεία και στοιχεία σχετικά με την εκπαίδευση και τα ενδιαφέροντά τους. Αφού ολοκληρώσουν την φόρμα, έχουν τη δυνατότητα να υποβάλλουν την αίτησή τους.

Με την επιτυχή υποβολή της αίτησης, οι εκπρόσωποι του ΠΜΣ, δηλαδή οι καθηγητές που είναι υπεύθυνοι για το συγκεκριμένο πρόγραμμα, έχουν την δυνατότητα να προχωρήσουν στη φάση της αξιολόγησης. Αυτό πραγματοποιείται μετά τη λήξη της περιόδου υποβολής αιτήσεων, και έχουν την ευκαιρία να εξετάσουν και να αξιολογήσουν όλες τις υποβληθείσες αιτήσεις.

Μέσω αυτής της διαδικασίας, οι υποψήφιοι έχουν την ευκαιρία να αναζητήσουν και να αιτηθούν για τα προγράμματα που τους ενδιαφέρουν, ενώ οι καθηγητές έχουν το χρόνο να αξιολογήσουν τις αιτήσεις με προσοχή μετά την παρέλευση της περιόδου υποβολής. Αυτή η συνεκτική διαδικασία συμβάλλει στη διασφάλιση ότι οι υποψήφιοι έχουν ένα σαφές και δίκαιο πλαίσιο για την υποβολή των αιτήσεών τους και ότι οι καθηγητές έχουν τον κατάλληλο χρόνο και πόρους για την αξιολόγηση των αιτήσεων με την απαιτούμενη προσοχή.

Ένα ακόμη χαρακτηριστικό που παρέχει η εφαρμογή είναι η δυνατότητα των Συστατικών Επιστολών για τους υποψήφιους σπουδαστές. Σε περιπτώσεις που το συγκεκριμένο πρόγραμμα απαιτεί Συστατικές Επιστολές (η ποσότητα των οποίων καθορίζεται από τον διευθυντή του προγράμματος για κάθε περίοδο υποβολής αιτήσεων), οι επιλεγμένοι καθηγητές ενημερώνονται μέσω ηλεκτρονικού μηνύματος και λαμβάνουν έναν μοναδικό σύνδεσμο. Μέσω αυτού του συνδέσμου, καλούνται να συμπληρώσουν τα δικά τους προσωπικά στοιχεία και να προσθέσουν τη σύστασή τους για τον συγκεκριμένο υποψήφιο.

Μετά την υποβολή της σύστασης, οι καθηγητές δεν έχουν περαιτέρω δυνατότητα επεξεργασίας ή τροποποίησής της. Αυτή η διαδικασία ενισχύει τη διαφάνεια και την ακεραιότητα του συστήματος, καθώς ο υποψήφιος φοιτητής δεν έχει πρόσβαση στο περιεχόμενο της συστατικής επιστολής που έχει συνταχθεί για αυτόν. Με αυτόν τον τρόπο, διασφαλίζεται η ακεραιότητα της αξιολόγησης και η αποτροπή πιθανών παρεμβάσεων ή επιρροών που μπορεί να επηρεάσουν την αντικειμενικότητα της διαδικασίας επιλογής.

Η διαδικασία ολοκληρώνεται με την αξιολόγηση των υποβληθεισών αιτήσεων. Όπως αναφέρθηκε προηγουμένως, οι διευθυντές των Προγραμμάτων Μεταπτυχιακών Σπουδών έχουν την δυνατότητα να προσκαλούν νέους χρήστες στην πλατφόρμα και να τους αναθέτουν ρόλους σε

εκείνα τα ΠΜΣ που έχουν επιλέξει.

Αφού ολοκληρωθεί η περίοδος υποβολής αιτήσεων, οι καθηγητές που έχουν αναλάβει ρόλους στο συγκεκριμένο πρόγραμμα έχουν την δυνατότητα να αξιολογήσουν τους υποψηφίους σύμφωνα με τα κριτήρια που έχουν καθοριστεί κατά τη δημιουργία του προγράμματος. Κατά τη διάρκεια της αξιολόγησης, οι καθηγητές έχουν πρόσβαση στα προσωπικά στοιχεία των υποψηφίων, καθώς και στα αρχεία που ενδέχεται να έχουν ανεβάσει (όπως βιογραφικό, αποδείξεις πτυχίου) και στις συστατικές επιστολές που έχουν υποβληθεί για τους υποψηφίους.

Όταν ολοκληρωθεί η αξιολόγηση όλων των υποψηφίων, ο διευθυντής του προγράμματος έχει την δυνατότητα να επιλέξει τους υποψηφίους που θα γίνουν δεκτοί στο πρόγραμμα. Η ποσότητα των διαθέσιμων θέσεων καθορίζεται από τον διευθυντή, και η εφαρμογή αυτοματοποιεί τη διαδικασία επιλογής με βάση τις υψηλότερες βαθμολογίες. Στη συνέχεια, αποστέλλονται αυτόματα ηλεκτρονικά μηνύματα σε όλους τους υποψηφίους που επιλέχθηκαν, ενημερώνοντας τους για την αποδοχή τους ή την εισαγωγή τους σε λίστα αναμονής, ανάλογα με τη διαθεσιμότητα θέσεων.

Με αυτόν τον τρόπο, ολοκληρώνεται η διαδικασία επιλογής και αξιολόγησης υποψηφίων, εξασφαλίζοντας τη διαφάνεια και την ακεραιότητα της διαδικασίας και εξασφαλίζοντας ότι οι υποψήφιοι που προσλαμβάνονται ανταποκρίνονται στα υψηλά πρότυπα του προγράμματος.

Αξίζει να επισημανθεί πως μια τέτοια εφαρμογή προάγει τη διαφάνεια, την αποδοτικότητα και τη συνοχή στη διαχείριση των μεταπτυχιακών προγραμμάτων. Από τη στιγμή που η εφαρμογή είναι προσαρμόσιμη στις ανάγκες διαφορετικών τμημάτων και πανεπιστημιακών ιδρυμάτων, συμβάλλει στη δημιουργία ενός συνεκτικού και οργανωμένου περιβάλλοντος που ωφελεί όλους τους εμπλεκόμενους. Με αυτόν τον τρόπο, η εφαρμογή παρέχει μια πιο ολοκληρωμένη και ενημερωμένη προοπτική για τα μεταπτυχιακά προγράμματα σε ολόκληρη την ακαδημαϊκή κοινότητα.

1.5 Οργάνωση της εργασίας

Στο πλαίσιο αυτής της πτυχιακής εργασίας, θα πραγματοποιηθεί μια λεπτομερής ανάλυση των βημάτων που ακολουθήθηκαν για την ανάπτυξη της εφαρμογής ScholarPath. Αρχικά, παρουσιάζονται τεχνολογίες που επιλέχθηκαν, καθώς και τους λόγους πίσω από αυτές τις επιλογές. Αυτό περιλαμβάνει τις γλώσσες προγραμματισμού, τα frameworks, τη βάση δεδομένων, και τον λόγο για τον οποίο επιλέχθηκαν.

Στο τρίτο κεφάλαιο, θα εξετάσουμε τις αρχιτεκτονικές αποφάσεις που λήφθηκαν για την εφαρμογή, το σχήμα της βάσης δεδομένων, και τις λειτουργικές απαιτήσεις που καλύπτονται για τους διάφορους χρήστες που θα αξιοποιήσουν την εφαρμογή. Επίσης, θα παρουσιάσουμε παραδείγματα κώδικα για την ανάπτυξη τόσο του frontend όσο και του backend.

Στο τέταρτο κεφάλαιο, θα εξεταστούν τα διάφορα σενάρια χρήσης για τους διαφορετικούς

χρήστες που θα αλληλεπιδράσουν με την εφαρμογή.

Τέλος, στο πέμπτο κεφάλαιο, θα παρουσιάσουμε κάποια συμπεράσματα και προτάσεις για τον τρόπο με τον οποίο μπορεί να εξελιχθεί η εφαρμογή ScholarPath στο μέλλον.

Αυτή η δομή θα μας βοηθήσει να αναλύσουμε εκτενώς την ανάπτυξη και τη λειτουργία της εφαρμογής, καθιστώντας την Πτυχιακή εργασία πλήρη και κατανοητή για τους αναγνώστες.

Κεφάλαιο 2

Τεχνολογίες

2.1 JavaScript

Η γλώσσα προγραμματισμού JavaScript εμφανίστηκε στα τέλη της δεκαετίας του '90 ως απάντηση στην ανάγκη για μια δυναμική γλώσσα που θα μπορούσε να προσθέσει διαδραστικότητα και δυναμισμό στις ιστοσελίδες. Οι βασικοί παράγοντες που οδήγησαν στην εμφάνιση της JavaScript περιλαμβάνουν την ανάγκη για δυναμική ανταπόκριση των ιστοσελίδων στις ενέργειες των χρηστών και τη δυνατότητα δημιουργίας απλών επιθυμητών εφαρμογών μέσω της περιήγησης.

Πριν από την εμφάνιση της JavaScript, οι ιστοσελίδες ήταν κυρίως στατικές και περιελάμβαναν κυρίως κείμενο και εικόνες. Οι δυνατότητες αλληλεπίδρασης με τον χρήστη ήταν περιορισμένες, καθιστώντας την εμπειρία της περιήγησης περιορισμένη και στατική. Η ανάγκη για δυναμικότητα και διαδραστικότητα οδήγησε στην αναζήτηση μιας λύσης που θα μπορούσε να εκτελείται απευθείας στο περιηγητή του χρήστη.

Η εμφάνιση της Java τη δεκαετία του '90 ενέπνευσε την Netscape Communications Corporation να αναπτύξει τη γλώσσα LiveScript, η οποία ενσωματώθηκε στον περιηγητή Netscape Navigator 2.0. Παρόλο που η LiveScript είχε την ευελιξία που αναζητούσαν οι προγραμματιστές για δυναμική αλληλεπίδραση, το όνομα της έπρεπε να αντικατοπτρίζει περισσότερο τη συνεργασία της με την Java, η οποία ήταν δημοφιλής την εποχή. Αυτή η ανάγκη για μια πιο εμφανή σχέση με την Java οδήγησε στη μετονομασία της γλώσσας σε "JavaScript."

Η πρώτη δημόσια εμφάνιση της JavaScript έγινε μαζί με την έκδοση 2.0 του Netscape Navigator το 1995. Αυτός ήταν ο πρώτος περιηγητής που ενσωμάτωσε τη γλώσσα και προσέφερε στους προγραμματιστές τη δυνατότητα να προσθέσουν δυναμική λειτουργικότητα στις ιστοσελίδες τους. Η JavaScript αρχικά χρησιμοποιήθηκε κυρίως για απλές λειτουργίες όπως επαλήθευση των δεδομένων των φορμών και διαφορετικές μορφές διαδραστικότητας[1].

Από την εκείνη στιγμή, η JavaScript έχει υποστεί διαρκή εξέλιξη, ενσωματώνοντας νέα χαρακτηριστικά και δυνατότητες που την έχουν καταστήσει αναπόσπαστο μέρος του κόσμου του

web προγραμματισμού.

Μια από τις πλέον ουσιώδεις πτυχές της JavaScript είναι η μηχανή εκτέλεσης που ενσωματώνεται στους περιηγητές. Αυτή η μηχανή αναλαμβάνει τον ρόλο της ερμηνείας και εκτέλεσης του κώδικα JavaScript στον περιηγητή του χρήστη, επιτρέποντας τη δημιουργία δυναμικών και αλληλεπιδραστικών ιστοσελίδων.

Εντυπωσιακή είναι επίσης η ασύγχρονη φύση της JavaScript. Μέσω αυτού του μοντέλου προγραμματισμού, η γλώσσα υποστηρίζει τη διαλειτουργικότητα με τον διακομιστή χωρίς να αναστέλλει τη λειτουργία της σελίδας. Αυτό επιτρέπει τη δυναμική ανταλλαγή δεδομένων, εξασφαλίζοντας παράλληλα μια ομαλή και ανεμπόδιση εμπειρία του χρήστη.

Στην καρδιά της JavaScript βρίσκονται οι μεταβλητές και οι τύποι δεδομένων. Οι μεταβλητές λειτουργούν ως αποθηκευτικοί χώροι για δεδομένα, ενώ οι τύποι δεδομένων περιλαμβάνουν αριθμούς, συμβολοσειρές, λογικές τιμές και άλλα. Η παρουσία των συναρτήσεων προσθέτει ακόμα περισσότερη οργάνωση και αφαιρεί την πολυπλοκότητα από τον κώδικα, επιτρέποντας την εκτέλεση συγκεκριμένων εργασιών με ευκολία.

Οι συμβολοσειρές και οι πίνακες συμβάλλουν στην αναπαράσταση δεδομένων. Οι πρώτες αναπαριστούν κείμενο, ενώ οι δεύτεροι αποθηκεύουν πολλές τιμές σε μια δομή. Όλα αυτά τα χαρακτηριστικά παρέχουν τα εργαλεία που χρειάζονται οι προγραμματιστές για τη δημιουργία πλούσιων και λειτουργικών ιστοσελίδων.

Το DOM (Document Object Model) αναδεικνύει την αξία της JavaScript στον κόσμο της ανάπτυξης ιστοσελίδων. Αποτελώντας τη διασύνδεση μεταξύ του κώδικα και των στοιχείων της ιστοσελίδας, το DOM επιτρέπει τη δυναμική αλλαγή του περιεχομένου και την αλληλεπίδραση με τον χρήστη. Με αυτόν τον τρόπο, η JavaScript διαμορφώνει τον τρόπο που αλληλεπιδρούμε με τον κόσμο του διαδικτύου, καθιστώντας την αναπόσπαστο κομμάτι της σύγχρονης ιστορίας του προγραμματισμού.

Παρ' όλα αυτά, ο κόσμος της JavaScript παρουσιάζει μια σειρά από προκλήσεις και τάσεις που επηρεάζουν τον τρόπο με τον οποίο αναπτύσσονται και λειτουργούν οι ιστοσελίδες και οι εφαρμογές. Αυτές οι πτυχές αναδεικνύουν τις προκλήσεις που αντιμετωπίζουν οι προγραμματιστές και τις ευκαιρίες που προσφέρονται για την περαιτέρω εξέλιξη της JavaScript.

Η ασφάλεια αποτελεί έναν από τους βασικούς παράγοντες που καθορίζουν το μέλλον της JavaScript. Εξαιτίας της εκτέλεσης του κώδικα στο περιβάλλον του περιηγητή, η διατήρηση της ασφάλειας αποτελεί προτεραιότητα. Οι προγραμματιστές πρέπει να αντιμετωπίσουν τις απειλές των κακόβουλων επιθέσεων, όπως η παρακολούθηση του χρήστη ή η εκμετάλλευση ευπάθειών στον κώδικα. Είναι αναγκαίο να εφαρμόζονται αυστηρά μέτρα ασφαλείας και να ενημερώνονται διαρκώς για να αντιμετωπίζουν τις νέες απειλές.

Η πολυπλοκότητα των σύγχρονων web εφαρμογών αποτελεί μια πρόκληση που αυξάνεται διαρκώς. Οι προγραμματιστές αντιμετωπίζουν την ανάγκη να διαχειρίζονται μεγάλα και πολύπλοκα σύνολα κώδικα, να διασφαλίζουν τη συντήρηση και την επεκτασιμότητα των εφαρ-

μογών τους. Η χρήση καλών πρακτικών, όπως η οργάνωση του κώδικα σε μικρότερα και πιο αναγνώσιμα κομμάτια, αποτελεί βασικό μέσο για την αντιμετώπιση αυτής της πρόκλησης. Σε κάθε περίπτωση, η JavaScript συνεχίζει να εξελίσσεται και να προσαρμόζεται στις αλλαγές της τεχνολογικής τοπίου, δημιουργώντας νέες ευκαιρίες και αντιμετωπίζοντας νέες προκλήσεις.

2.2 TypeScript

Η TypeScript είναι μια προγραμματιστική γλώσσα που αναπτύχθηκε από τη Microsoft και αποτελεί ένα υπερσύνολο της γλώσσας JavaScript. Η δημιουργία της προήλθε από την ανάγκη για ένα εργαλείο που θα προσέφερε ασφάλεια, ευελιξία και αποδοτικότητα κατά την ανάπτυξη μεγάλων εφαρμογών.

Ένα από τα βασικά χαρακτηριστικά της είναι η υποστήριξη των στατικών τύπων δεδομένων [2]. Αυτό σημαίνει ότι μπορεί να οριστεί ο τύπος των μεταβλητών, των παραμέτρων συναρτήσεων και των αντικειμένων, προσφέροντας μεγαλύτερη ασφάλεια κατά τον χειρισμό σφαλμάτων και την ανάπτυξη. Με τη χρήση στατικών τύπων, μπορούν να εντοπιστούν πιθανά προβλήματα στον κώδικα κατά τη διάρκεια της μεταγλώττισης, προτού αυτός εκτελεστεί, βελτιώνοντας την ποιότητα του κώδικα και μειώνοντας τον κίνδυνο σφαλμάτων.

Η TypeScript παρέχει επίσης ευελιξία, καθώς επιτρέπει τη χρήση των τελευταίων χαρακτηριστικών της ECMAScript. Αυτό σημαίνει ότι μπορούν να αξιοποιηθούν τα νέα εργαλεία και πρακτικές ανάπτυξης, διατηρώντας την συμβατότητα με παλαιότερες εκδόσεις της γλώσσας. Αυτό επιτρέπει την ομαλή μετάβαση από υπάρχοντα έργα και εξασφαλίζει την αποδοτική ανάπτυξη και συντήρηση των εφαρμογών.

Η TypeScript και η JavaScript είναι δύο στενά συνδεδεμένες γλώσσες προγραμματισμού, αλλά υπάρχουν κάποιες κύριες διαφορές μεταξύ τους. Η JavaScript είναι αδύναμα τυποποιημένη γλώσσα, που σημαίνει ότι δεν υπάρχει υποχρεωτική δήλωση τύπων μεταβλητών. Αντίθετα, μέσω της τύπωσης μεταβλητών, η TypeScript εξασφαλίζει ότι κάθε μεταβλητή θα έχει ορισμένο τύπο δεδομένων, μειώνοντας τον κίνδυνο μη αναμενομένων τιμών και συμπεριφορών. Επιπλέον, η αυστηρότητα στον ορισμό των τύπων βοηθά στην ευκολότερη εντοπισμό και επίλυση σφαλμάτων κατά τη διάρκεια της ανάπτυξης.

Ένα άλλο πλεονέκτημα είναι η δυνατότητα πρόληψης σφαλμάτων και ασάφειας κατά τη μεταγλώττιση. Ενώ η JavaScript μπορεί να παρουσιάσει ανεπιθύμητες συμπεριφορές κατά την εκτέλεση, η TypeScript μπορεί να εντοπίσει και να εμποδίσει αυτούς τους κινδύνους κατά την φάση της μεταγλώττισης. Αυτό σημαίνει ότι προλαμβάνονται πιθανά προβλήματα πριν καν αναλάβει την εκτέλεση το πρόγραμμα.

Επιπλέον, η TypeScript κάνει τον κώδικα πιο ευανάγνωστο και οργανωμένο, προσφέροντας ένα πιο δομημένο περιβάλλον ανάπτυξης. Οι τύποι δεδομένων και οι διασφαλίσεις που παρέ-

χονται βοηθούν τους προγραμματιστές να κατανοήσουν τον κώδικα πιο εύκολα και να εντοπίσουν προβλήματα πιο αποτελεσματικά. Αυτό έχει ως αποτέλεσμα μια πιο γρήγορη διαδικασία ανάπτυξης και συντήρησης.

Τέλος, η TypeScript διατηρεί στενή συμβατότητα με τις πιο πρόσφατες εκδόσεις της JavaScript, επιτρέποντας τη χρήση των νέων και σύγχρονων χαρακτηριστικών της γλώσσας. Αυτό συμβάλλει στην αποδοτική αξιοποίηση των εξελίξεων της JavaScript και στην εξέλιξη των εφαρμογών με βάση τα πιο πρόσφατα πρότυπα και πρακτικές.

Συνολικά, η TypeScript προσφέρει ένα ολοκληρωμένο πακέτο πλεονεκτημάτων σε σύγκριση με την καθαρή JavaScript, βελτιώνοντας την ασφάλεια, τη συντηρησιμότητα και την αποτελεσματικότητα της ανάπτυξης εφαρμογών.

2.3 Node.js

Το Node.js αποτελεί μια ανοιχτού κώδικα πλατφόρμα που επιτρέπει την εκτέλεση κώδικα JavaScript στον εξυπηρετητή (server-side). Η εμφάνισή του σηματοδοτεί μια σημαντική εξέλιξη στον κόσμο της ανάπτυξης λογισμικού, καθώς διευρύνει το πεδίο εφαρμογής της γλώσσας JavaScript πέρα από τον περιβάλλον του φυλλομετρητή (browser) και την μετατρέπει σε μια ευέλικτη επιλογή για το backend των εφαρμογών.

Η πρώτη δημοσίευση του Node.js πραγματοποιήθηκε το 2009 από τον Ryan Dahl. Το κίνητρο για τη δημιουργία του Node.js προήλθε από το γεγονός ότι οι περισσότερες γλώσσες που χρησιμοποιούνταν για την ανάπτυξη στο backend δεν είχαν την ίδια ευκολία και αποδοτικότητα με τη JavaScript στον φυλλομετρητή. Έτσι, το Node.js δημιούργησε ένα περιβάλλον που επιτρέπει την εκτέλεση JavaScript στον εξυπηρετητή, επιτρέποντας την ανάπτυξη ολοκληρωμένων εφαρμογών, ανεξάρτητα από το περιβάλλον που τρέχουν.

Η σημαντικότητα του Node.js στο backend είναι πολύπλευρη. Καταρχάς, δημιουργεί μια ενοποιημένη γλώσσα προγραμματισμού, καθιστώντας δυνατή τη χρήση της ίδιας γλώσσας και στον frontend και στον backend. Αυτό μειώνει το κόστος εκμάθησης για τους προγραμματιστές και επιτρέπει την κοινή χρήση κώδικα και λειτουργικότητων ανάμεσα στα δύο μέρη της εφαρμογής.

Το Node.js ανοίγει τον δρόμο για τη χρήση της JavaScript σε ολόκληρο το φάσμα της ανάπτυξης λογισμικού, και μαζί με την TypeScript, δίνει τη δυνατότητα για συστηματική, αποτελεσματική και ασφαλή ανάπτυξη εφαρμογών.

2.3.1 Node Modules και Node Package Manager

Το Node.js έχει μια μεγάλη κοινότητα η οποία μπορεί να διακριθεί από τα ποικίλλα node modules. Η χρήση των Node.js modules [3] αποτελεί ένα βασικό και ουσιαστικό μέρος της ανά-

πτυξης εφαρμογών σε αυτό το περιβάλλον. Τα Node modules αποτελούν την απάντηση στην ανάγκη για επαναχρησιμοποίηση και οργάνωση του κώδικα. Χρησιμοποιώντας την δυνατότητα του npm για τη διαχείριση των εξαρτήσεων, μπορεί να αξιοποιηθεί η ποικιλία πακέτων που προσφέρονται από την Node.js κοινότητα. Αυτά τα πακέτα σχεδιάστηκαν για να καλύψουν διάφορες ανάγκες και προβλήματα που μπορεί να αντιμετωπίσει κανείς κατά την ανάπτυξη. Από πακέτα που παρέχουν βασική λειτουργικότητα μέχρι ειδικές λύσεις για συγκεκριμένα πεδία, η ποικιλία είναι εντυπωσιακή.

Η κοινότητα συνεισφέρει και συνεργάζεται στην ανάπτυξη των Node modules. Πολλά από αυτά δημιουργούνται από εθελοντές προγραμματιστές από όλο τον κόσμο, οι οποίοι μοιράζονται τις λύσεις τους με την κοινότητα. Αυτό δημιουργεί ένα οικοσύστημα αμοιβαίας ωφέλειας, όπου ο κώδικας βελτιώνεται συνεχώς μέσω της ανταλλαγής γνώσης και εμπειρίας.

Επιπλέον, η χρήση των Node modules προσφέρει ταχύτητα και αποδοτικότητα κατά την ανάπτυξη. Αντί να κατασκευάζονται τα πάντα από την αρχή, μπορούν να επαναχρησιμοποιηθούν ήδη υπάρχουσες λύσεις που έχουν δοκιμαστεί και βελτιωθεί από την κοινότητα. Αυτό εξοικονομεί χρόνο και πόρους και επιτρέπει στους προγραμματιστές να επικεντρωθούν στην πραγματική λειτουργικότητα της εφαρμογής.

Το `package.json` είναι ένα κρίσιμο αρχείο που παρέχει πληροφορίες σχετικά με την εφαρμογή και τις εξαρτήσεις της. Αυτό το αρχείο κρατά τις πληροφορίες σχετικά με το όνομα της εφαρμογής, την έκδοση, τον συγγραφέα, τις εξαρτήσεις και πολλά άλλα. Αυτό δεν απλώς παρέχει κατευθυντήριες γραμμές για την εφαρμογή, αλλά επίσης διευκολύνει τη διαχείριση των εξαρτήσεων.

Το Node Package Manager (npm) [4] αναλαμβάνει την διαχείριση των εξαρτήσεων της εφαρμογής. Με την εντολή `npm install`, εγκαταστούνται όλες οι απαιτούμενες εξαρτήσεις που καθορίζονται στο αρχείο `package.json`. Αυτό εξασφαλίζει ότι η εφαρμογή έχει όλα τα απαραίτητα πακέτα για να λειτουργήσει.

Το npm δεν απλώς εγκαθιστά εξαρτήσεις, αλλά προσφέρει επίσης δυνατότητες για τη δημιουργία, την ενημέρωση και τη διανομή των πακέτων. Αυτό επιτρέπει σε προγραμματιστές να συνεισφέρουν στην κοινότητα ανοιχτού κώδικα και να μοιραστούν τις λύσεις τους με άλλους.

Συνολικά, η χρήση των Node modules, του `package.json` και του npm επιτρέπει τη δομή, τη διαχείριση και την ανάπτυξη εφαρμογών στο περιβάλλον του Node.js με αποτελεσματικό τρόπο.

2.3.2 Event Loop

Το Event Loop και η ασυγχρονότητα είναι κρίσιμα χαρακτηριστικά του Node.js που διαμορφώνουν την τρόπο λειτουργίας του και το καθιστούν ένα ισχυρό περιβάλλον για την ανάπτυξη υψηλής απόδοσης εφαρμογών.

Το Event Loop αναφέρεται στο μηχανισμό με τον οποίο το Node.js διαχειρίζεται τις εργασίες και τα γεγονότα στην εφαρμογή του [5]. Αντί να περιμένει μπλοκαρισμένες εργασίες να ολο-

κληρωθούν προτού προχωρήσει σε άλλες, το Node.js χρησιμοποιεί μια ασύγχρονη προσέγγιση. Κατά την εκτέλεση, το Node.js αντιμετωπίζει αιτήματα, εργασίες και γεγονότα ως "γεγονότα" που πρέπει να επιλυθούν, αλλά δεν τα επεξεργάζεται αμέσως. Αντ' αυτού, τα προγράμματα του Node.js συνεχίζουν την εκτέλεσή τους, επιτρέποντας σε άλλα αιτήματα να εισέλθουν στην ουρά του Event Loop.

Η ασυγχρονότητα είναι το κλειδί για την αποτελεσματικότητα του Node.js. Αντί να περιμένει ένα αίτημα ή μια εργασία να ολοκληρωθεί προτού προχωρήσει σε άλλες εργασίες, το Node.js εκμεταλλεύεται το γεγονός ότι πολλές από τις εργασίες στις εφαρμογές δεν απαιτούν πλήρη εξυπηρέτηση αμέσως. Χρησιμοποιεί την ιδέα του να "εμφωλεύει" αιτήματα και εργασίες μέσα σε callbacks, promises και async/await, επιτρέποντας στο Event Loop να παρακολουθεί τα γεγονότα και να αναλάβει δυναμικά τον χειρισμό τους καθώς ολοκληρώνονται.

Αυτή η ασύγχρονη προσέγγιση έχει πολλά θετικά [6] για το Node.js:

- **Υψηλή Απόδοση:** Επιτρέπει στο Node.js να εξυπηρετεί πολλά αιτήματα ταυτόχρονα, χωρίς να μπλοκάρεται αναμένοντας την ολοκλήρωση μίας εργασίας πριν προχωρήσει σε άλλες.
- **Κλιμακωσιμότητα:** Η ασυγχρονότητα είναι καθοριστική για την κατασκευή κλιμακώσιμων εφαρμογών. Το Node.js μπορεί να χειριστεί αιτήματα από πολλούς χρήστες ταυτόχρονα χωρίς να επηρεάζεται από την αυξημένη φόρτο.
- **Αποτελεσματικότητα:** Εφόσον το Node.js μπορεί να συνεχίσει την εκτέλεσή του χωρίς να περιμένει την ολοκλήρωση μιας εργασίας, μειώνονται οι καθυστερήσεις και βελτιώνεται η απόκριση της εφαρμογής.
- **Αποτροπή Μπλοκαρίσματος:** Η ασυγχρονότητα αποτρέπει τον κίνδυνο μπλοκαρίσματος της εφαρμογής λόγω μακροχρόνιων εργασιών. Αυτό εξασφαλίζει ότι οι υπόλοιπες λειτουργίες της εφαρμογής παραμένουν διαθέσιμες και αποτρέπει την ανεπιθύμητη καθυστέρηση.

Ο συνδυασμός του Event Loop και της ασυγχρονότητας αποτελεί θεμέλιο του Node.js και επιτρέπει τη δημιουργία αποδοτικών, ανταποκρινόμενων και κλιμακώσιμων εφαρμογών που μπορούν να αντεπεξέλθουν σε υψηλό φόρτο και πολλαπλές αιτήσεις.

2.3.3 Koa.js

Το Koa.js αντιπροσωπεύει ένα πλαίσιο ανάπτυξης [7] που έχει κατακτήσει μεγάλη αναγνώριση στον κόσμο του Node.js, χάρη στην ευελιξία του και την εξειδίκευσή του στην διαχείριση ασύγχρονων εργασιών. Η βασική του ιδέα, που βασίζεται στην χρήση middlewares, ανοίγει τον δρόμο για τη δημιουργία προσαρμοσμένων και αποδοτικών εφαρμογών με λιγότερη πολυπλοκότητα.

Το Koa.js είναι το αποτέλεσμα της συνεχούς εξέλιξης και βελτίωσης του ευρέως γνωστού πλαισίου Express.js. Αν και το Express προσφέρει πλούσια δυνατότητα και έχει προσδοθεί πολλή

αξία, το Koa προσεγγίζει την ανάπτυξη με έναν πιο λιτό και ελαφρύ τρόπο. Αντί να παρέχει πολλές ενσωματωμένες λειτουργίες, επιτρέπει στους προγραμματιστές να επιλέξουν ποιες λειτουργίες θα χρησιμοποιήσουν μέσω της χρήσης middlewares.

Η καίρια έννοια των middlewares είναι ότι αυξάνουν τη δυνατότητα ανακατανομής της λειτουργικότητας και της επεξεργασίας ενός αιτήματος και μιας απόκρισης. Κάθε middleware μπορεί να προσθέσει, τροποποιήσει ή αφαιρέσει δεδομένα, αλλάζοντας έτσι τη ροή του κώδικα προτού φτάσει στον επόμενο middleware. Αυτό επιτρέπει τη δημιουργία ευέλικτων αλυσίδων επεξεργασίας, με τη δυνατότητα προσθήκης και αφαίρεσης λειτουργικότητας όσο απαιτείται.

Σε ασύγχρονο περιβάλλον όπως το Node.js, η ανάγκη για αποδοτική διαχείριση της ασυγχρονίας είναι ιδιαίτερα σημαντική. Εδώ το Koa.js ξεχωρίζει, καθώς υιοθετεί την χρήση των Promises και των async/await για τη διαχείριση ασύγχρονων εργασιών. Αυτό επιτρέπει την καλύτερη διαχείριση των εργασιών που απαιτούν χρόνο, ενισχύοντας την αποδοτικότητα της εφαρμογής. Συνολικά, το Koa.js προσφέρει μια ανανεωμένη προσέγγιση στον κόσμο της ανάπτυξης εφαρμογών στο περιβάλλον του Node.js. Με την αξιοποίηση της ασύγχρονης φύσης του Node.js και της δυνατότητας προσθήκης λειτουργικότητας μέσω των middlewares, το Koa.js επιτρέπει την δημιουργία αποδοτικών, εξελισσόμενων και ευέλικτων εφαρμογών που μπορούν να αντιμετωπίσουν αποτελεσματικά τις απαιτήσεις του σύγχρονου διαδικτυακού τοπίου.

2.4 GraphQL

Η GraphQL είναι μια γλώσσα ερωτήσεων για το API και ένα σύστημα εκτέλεσης που αναπτύχθηκε από το Facebook το 2012 και ανοίχτηκε για το κοινό το 2015. Η GraphQL επιτρέπει να καθορίζονται ποιες δεδομένες χρειάζονται και σε ποια μορφή, καθορίζοντας έτσι έναν ευέλικτο τρόπο επικοινωνίας με τα APIs.

Σε αντίθεση με τα παραδοσιακά REST APIs, όπου κάθε διαδρομή έχει συγκεκριμένα σταθερά δεδομένα που επιστρέφονται, στο GraphQL [8], η πλευρά του πελάτη καθορίζει ακριβώς ποιες πληροφορίες απαιτούνται. Αυτό μειώνει την υπερβολική ή μη απαραίτητη φόρτωση δεδομένων, που αποτελεί συχνή πρόκληση στα REST APIs.

Η βασική ιδέα της GraphQL είναι ότι μπορεί να καλείται ένα μόνο "end point" για να ανακτηθούν όλα τα απαιτούμενα δεδομένα, αντί να καλούνται πολλά διαφορετικά "end points" για διάφορες περιπτώσεις χρήσης. Αυτό δίνει στους πελάτες την ευθύνη για τον καθορισμό των ακριβών δεδομένων που χρειάζονται, μειώνοντας τον πλεονασμό και την υπερβολική φόρτωση του δικτύου.

Ένα ακόμη σημαντικό χαρακτηριστικό της GraphQL είναι η τυποποίηση των δεδομένων. Ορίζονται οι τύποι δεδομένων και οι πιθανές σχέσεις μεταξύ τους. Αυτό παρέχει στους πελάτες ένα καθαρό μοντέλο δεδομένων, το οποίο μπορούν να χρησιμοποιήσουν για τη δημιουργία των

ερωτημάτων τους.

Τέλος, η GraphQL επιτρέπει επίσης τις μεταλλάξεις, που αφορούν τροποποιήσεις στα δεδομένα. Με αυτόν τον τρόπο, οι πελάτες μπορούν να πραγματοποιούν ενέργειες προς τον διακομιστή, όπως δημιουργία, ενημέρωση ή διαγραφή δεδομένων.

2.4.1 TypeGraphQL

Η TypeGraphQL είναι μια βιβλιοθήκη για την ανάπτυξη σχημάτων GraphQL με τη χρήση της γλώσσας προγραμματισμού TypeScript. Με την χρήση αυτής της βιβλιοθήκης, τα σχήματα GraphQL μπορούν να δημιουργηθούν αυτόματα μέσω των τύπων δεδομένων που ορίζονται.

Η TypeGraphQL προσφέρει μια προσέγγιση που εστιάζει στη δημιουργία σχημάτων GraphQL μέσω διακομιστή, εξομοιώνοντας τα δεδομένα και τις λειτουργίες τους μέσω των τύπων της TypeScript. Αυτό καταστέλλει την ανάγκη χειροκίνητης δημιουργίας σχημάτων, καθιστώντας την διαδικασία πιο αποτελεσματική και εύκολη στη συντήρηση.

Η βιβλιοθήκη παρέχει δυνατότητες για την δημιουργία των τύπων GraphQL μέσω διάφορων δεκομιστών, όπως το `@ObjectType()` για τη δήλωση των αντικειμένων, το `@Field()` για τα πεδία των αντικειμένων, και το `@Query()` για την δήλωση ερωτημάτων. Επιπλέον, προσφέρει τη δυνατότητα δημιουργίας μεταλλάξεων μέσω του `@Mutation()` διακομιστή, επιτρέποντας την τροποποίηση των δεδομένων.

Μια από τις κύριες προσφορές της TypeGraphQL είναι η διασύνδεση της εισαγωγής και εξαγωγής δεδομένων με το σχήμα GraphQL, αφαιρώντας την ανάγκη για χειροκίνητη επαναλαμβανόμενη διαδικασία. Αυτό βοηθά στην εξοικονόμηση χρόνου και προσπάθειας κατά την ανάπτυξη των εφαρμογών.

Συνοπτικά, η TypeGraphQL αποτελεί ένα πολύτιμο εργαλείο για την ανάπτυξη GraphQL σχημάτων με τη χρήση TypeScript, διευκολύνοντας τη δημιουργία και τη διαχείριση των τύπων δεδομένων.

2.4.2 Resolvers

Το λειτουργικό μέρος που χειρίζεται τις ερωτήσεις και τις μεταλλάξεις στο GraphQL ονομάζεται "resolver". Οι resolvers είναι συναρτήσεις που ορίζονται για κάθε πεδίο στο σχήμα GraphQL και υπεύθυνες για την ανάκτηση ή την τροποποίηση των δεδομένων.

Όταν ένα αίτημα γίνεται στο GraphQL server για ένα συγκεκριμένο πεδίο, ο αντίστοιχος resolver εκτελείται για να παρέχει τα δεδομένα. Οι resolvers έχουν στοχαστικά το ίδιο όνομα με τα πεδία του σχήματος και μπορούν να επιστρέψουν την αντίστοιχη τιμή για το πεδίο.

Επίσης, η TypeGraphQL προσφέρει μια εύκολη προσέγγιση για τη δημιουργία των resolvers. Αυτό γίνεται χρησιμοποιώντας τις πληροφορίες από τους τύπους δεδομένων TypeScript για τους διακομιστές και τα πεδία, κατά τρόπο που οι resolvers δημιουργούνται αυτόματα.

Αυτό διευκολύνει την ανάπτυξη, καθώς δεν χρειάζεται να γράφονται χειροκίνητα οι resolvers [9] για κάθε πεδίο. Αντ' αυτού, οι λειτουργίες των resolvers δημιουργούνται αυτόματα με βάση τη δομή του σχήματος.

Οι resolvers αποτελούν το "κινητήριο" μέρος του GraphQL, υπεύθυνοι για την ανάκτηση και την επεξεργασία των δεδομένων που απαιτούνται από τα αιτήματα. Η TypeGraphQL παρέχει μια ευκολότερη προσέγγιση για την δημιουργία των resolvers μέσω της χρήσης των δομών του σχήματος και της TypeScript.

2.4.3 Queries

Τα queries στο περιβάλλον των resolvers χρησιμοποιούνται για την ανάκτηση συγκεκριμένων δεδομένων από το GraphQL API. Αυτά τα queries υλοποιούνται με τη μορφή συναρτήσεων [9] που εκτελούνται κατά τη διεκπεραίωση των αιτημάτων από τους πελάτες.

Κατά την διαδικασία της εκτέλεσης ενός query, οι συγκεκριμένες λειτουργίες που θα πρέπει να εκτελεστούν για να ανακτηθούν τα δεδομένα ορίζονται μέσα στους resolvers. Οι resolvers είναι υπεύθυνοι για τη διαχείριση των αιτημάτων και την επιστροφή των αντίστοιχων αποτελεσμάτων.

Κάθε query έχει μια αντίστοιχη συνάρτηση resolver που εκτελείται όταν γίνεται μια αίτηση για το συγκεκριμένο query. Αυτή η συνάρτηση αναζητεί και ανακτά τα απαιτούμενα δεδομένα, επεξεργάζεται τα δεδομένα αν απαιτείται, και τα επιστρέφει ως αποτέλεσμα.

Τα queries μπορούν να δέχονται παραμέτρους, οι οποίες παρέχονται από τους πελάτες κατά την υποβολή των αιτημάτων. Οι παράμετροι αυτές χρησιμοποιούνται από τους resolvers για να προσαρμόσουν τα δεδομένα που επιστρέφονται.

Στο γενικότερο πλαίσιο, τα queries των resolvers αποτελούν τον μηχανισμό μέσω του οποίου οι πελάτες αιτούνται να ανακτήσουν δεδομένα από το σύστημα. Οι resolvers αναλαμβάνουν την εκτέλεση των απαιτούμενων λειτουργιών για να παράσχουν τα ζητούμενα αποτελέσματα, χρησιμοποιώντας τα δεδομένα από διάφορες πηγές, όπως βάσεις δεδομένων ή εξωτερικούς πόρους.

2.4.4 Mutations

Τα mutations από την άλλη, μέσα στο περιβάλλον των resolvers, χρησιμοποιούνται για την εκτέλεση τροποποιήσεων σε δεδομένα στο GraphQL API. Ο όρος "mutations" αναφέρεται στις λειτουργίες που επιτρέπουν την διαγραφή, την ενημέρωση ή τη δημιουργία νέων δεδομένων.

Κάθε mutation αντιστοιχεί σε μια συνάρτηση resolver, η οποία εκτελείται κατά τη διεκπεραίωση των αιτημάτων τροποποίησης από τους πελάτες. Η λειτουργία του resolver σε ένα mutation είναι να πραγματοποιεί τις απαιτούμενες αλλαγές στα δεδομένα και να επιστρέφει το αποτέλεσμα της τροποποίησης.

Σε περίπτωση που απαιτείται δημιουργία νέων δεδομένων, ο resolver μπορεί να εισάγει τα νέα δεδομένα στη βάση δεδομένων ή σε άλλες πηγές, ανάλογα με την υλοποίηση. Αν απαιτείται ενημέρωση υπαρχόντων δεδομένων, ο resolver μπορεί να επεξεργαστεί τα δεδομένα και να ανανεώσει τις αλλαγές. Σε περίπτωση διαγραφής, το mutation ενεργοποιεί τη διαγραφή των δεδομένων [8].

Τα mutations μπορούν επίσης να δέχονται παραμέτρους από τους πελάτες, αντίστοιχα με τις queries. Αυτό επιτρέπει την εκτέλεση διαφορετικών τροποποιήσεων βάσει των παρεχόμενων παραμέτρων.

Συνολικά, τα mutations στο περιβάλλον των resolvers αποτελούν τον μηχανισμό για την εκτέλεση τροποποιήσεων σε δεδομένα. Οι resolvers για τα mutations εκτελούν τις αναγκαίες λειτουργίες για τη διαχείριση των τροποποιήσεων και την ανανέωση των δεδομένων ανάλογα με την αίτηση των πελατών.

2.4.5 GraphQL Playground

Το GraphQL Playground είναι ένα διαδραστικό περιβάλλον που χρησιμοποιείται για τον ανασχεδιασμό, τη δοκιμή και την επικοινωνία με GraphQL APIs. Αποτελεί ένα εξαιρετικά χρήσιμο εργαλείο για προγραμματιστές και ομάδες ανάπτυξης που χρησιμοποιούν GraphQL για τη δημιουργία των API τους.

Με το GraphQL Playground, οι προγραμματιστές μπορούν να κατανοήσουν και να δοκιμάσουν τα διάφορα ερωτήματα που μπορούν να υποβάλλουν στο GraphQL API. Το περιβάλλον παρέχει ένα εύχρηστο πεδίο κειμένου όπου οι προγραμματιστές μπορούν να γράψουν τα ερωτήματα τους, να προσθέσουν μεταβλητές και να παρακολουθούν τη μορφή των δεδομένων που λαμβάνουν.

Ένα από τα ισχυρά σημεία του GraphQL Playground είναι η δυνατότητα αυτόματης συμπλήρωσης. Καθώς ο προγραμματιστής αρχίζει να πληκτρολογεί το ερώτημά του, το περιβάλλον προτείνει αυτόματα τα διαθέσιμα πεδία, τις μεταβλητές και τις υποδοχές. Αυτό βοηθά στη μείωση των σφαλμάτων και επιταχύνει τη διαδικασία της δημιουργίας των ερωτημάτων.

Επιπλέον, το GraphQL Playground παρέχει δυνατότητες παρουσίασης της δομής του GraphQL σχήματος. Αυτό σημαίνει ότι οι προγραμματιστές μπορούν να εντοπίσουν εύκολα τα διαθέσιμα πεδία, τις σχέσεις και τις υποδοχές που υποστηρίζονται από το API τους, κάτι που βοηθά στην κατανόηση της δομής των δεδομένων.

Καταλήγοντας, το GraphQL Playground είναι ένα εξαιρετικό εργαλείο που επιτρέπει στους προγραμματιστές να εξερευνήσουν, να δοκιμάσουν και να αλληλεπιδράσουν με τα GraphQL APIs τους με άνεση και αποτελεσματικότητα. Αυτό ενισχύει την ανάπτυξη και την αποσφαλμάτωση των εφαρμογών, δίνοντας τη δυνατότητα γρήγορης ανταπόκρισης και βελτιστοποίησης των ερωτημάτων.

2.5 Vue.js

Η Vue.js είναι ένα προοδευτικό πλαίσιο JavaScript που χρησιμοποιείται για την ανάπτυξη διεπαφών χρήστη (UI) σε ιστοσελίδες και εφαρμογές. Το πλαίσιο αυτό δημιουργήθηκε από τον Evan You και πρωτοκυκλοφόρησε τον Φεβρουάριο του 2014. Από τότε, έχει αποκτήσει τεράστια δημοτικότητα στον κόσμο της web ανάπτυξης.

Ένα από τα βασικά χαρακτηριστικά που έκαναν την Vue.js να γίνει γρήγορα δημοφιλής είναι η απλότητα της. Η Vue.js παρέχει μια εύκολη και φιλική στον προγραμματιστή σύνταξη (syntax) που είναι ευανάγνωστη και κατανοητή. Αυτό επιτρέπει στους προγραμματιστές να δημιουργούν γρήγορα επαναχρησιμοποιήσιμα και συντηρήσιμα UI στοιχεία.

Επιπλέον, το λεγόμενο "reactivity system" της Vue.js [10] είναι ένα από τα ισχυρότερα σημεία της. Αυτό το σύστημα επιτρέπει την αυτόματη ενημέρωση των δεδομένων σε πραγματικό χρόνο, καθιστώντας την ανάπτυξη δυναμικών και αναγνωρίσιμων UI στοιχείων εύκολη.

Ένα ακόμη χαρακτηριστικό που συνέβαλε στη δημοτικότητα της Vue.js είναι η δυνατότητα ενσωμάτωσής της σε άλλα έργα και ιστοσελίδες. Οι προγραμματιστές μπορούν να χρησιμοποιήσουν τη Vue.js με μεθόδους που τους είναι ήδη γνωστές, καθιστώντας τη μετάβαση πιο ομαλή.

Το εκτεταμένο οικοσύστημα εργαλείων και πρόσθετων (addons) είναι επίσης σημαντικό πλεονέκτημα της Vue.js. Υπάρχουν πολλά addons και εργαλεία που μπορούν να βοηθήσουν τους προγραμματιστές να επιτύχουν τα επιθυμητά αποτελέσματα με αποτελεσματικότητα.

Αναφορικά με την απόδοση, παρατηρείται ότι η Vue.js διαθέτει ένα εξαιρετικά αποδοτικό σύστημα αντιδραστικότητας. Αυτό το σύστημα είναι ιδιαίτερος κατάλληλο για εφαρμογές που απαιτούν τη συνεχή ανανέωση δεδομένων [11]. Η ικανότητα αυτή επιτρέπει στην εφαρμογή να ανταποκρίνεται με ακρίβεια σε αλλαγές στα δεδομένα, εξασφαλίζοντας την ομαλή και δυναμική εμπειρία χρήστη.

Εξίσου σημαντικό είναι το γεγονός ότι η Vue.js παρουσιάζει πλεονέκτημα σε ό,τι αφορά το μέγεθος των αρχείων της. Σε σύγκριση με το React και το Angular, η Vue.js εκδηλώνει μειωμένες απαιτήσεις σε ό,τι αφορά την αποθήκευση. Αυτό το γεγονός έχει ως αποτέλεσμα την ταχύτερη φόρτωση της εφαρμογής και την αποδοτικότερη ανταπόκριση της στις διάφορες αιτήσεις των χρηστών. Η αύξηση της ταχύτητας απόκρισης αποτελεί σημαντικό πλεονέκτημα κατά την ανάπτυξη ευέλικτων και ευαίσθητων εφαρμογών.

2.5.1 Οργάνωση Αρχείων

Σχετικά με την οργάνωση των αρχείων σε Vue.js εφαρμογές με τη χρήση του χαρακτηριστικού `template-script-style`, ας εξετάσουμε τα εξής:

Το χαρακτηριστικό `template-script-style` αποτελεί μια κατευθυντήρια αρχή για την οργάνωση των αρχείων των συνιστωσών Vue.js. Αναπαριστά τη συνήθη πρακτική του διαχωρισμού της

λογικής, της διαμόρφωσης και του στυλ των συνιστώσών με σκοπό τη βελτιστοποίηση της ανάγνωσης, της συντήρησης και της συνεργασίας μεταξύ των μελών της ομάδας.

Το τμήμα `template` περιέχει το HTML περιεχόμενο της συνιστώσας. Εδώ ορίζεται το γραφικό UI της εφαρμογής, όπως τα στοιχεία DOM και οι διαδραστικές συμπεριφορές.

Το τμήμα `script` περιέχει το JavaScript κώδικα που καθορίζει τη λογική της συνιστώσας. Εδώ μπορούν να οριστούν δεδομένα, μέθοδοι, υπολογισμοί και συμβάντα που αφορούν τη συγκεκριμένη συνιστώσα.

Το τμήμα `style` περιέχει τα στυλ CSS που εφαρμόζονται στη συνιστώσα. Εδώ μπορούν να οριστούν κανόνες σχεδιασμού, εφέ και εμφάνισης των στοιχείων.

Η αναγνωσιμότητα και η συντήρηση ενισχύονται μέσω της φυσικής διαχωρισμένης δομής. Επιπλέον, το χαρακτηριστικό `scoped` μπορεί να προστεθεί στο τμήμα `style` για την επίτευξη του `scoped CSS`, καθιστώντας τα στυλ εφαρμογής μόνο για τη συγκεκριμένη συνιστώσα.

Ο συνδυασμός αυτών των τριών τμημάτων επιτρέπει στους προγραμματιστές να αναπτύσσουν και να συντηρούν Vue.js εφαρμογές με ποιοτικό τρόπο. Ο διαχωρισμός των διαφορετικών πτυχών μιας συνιστώσας βοηθά στο να διατηρηθεί η σαφήνεια, η οργάνωση και η αποδοτικότητα κατά την ανάπτυξη και τη συντήρηση του κώδικα.

2.5.2 Composition API

Το Composition API της Vue.js αποτελεί μια προηγμένη επέκταση του πλαισίου Vue.js που δίνει τη δυνατότητα στον κώδικα να γραφεί με μεγαλύτερη οργάνωση, αυξημένη δυνατότητα επαναχρησιμοποίησης και αυξημένη δυνατότητα αναγνωσιμότητας. Αυτό επιτυγχάνεται μέσω της χρήσης των λεγόμενων "Composables" - μικρών συναρτήσεων που περιλαμβάνουν λογική και μπορούν να συνδυαστούν για τη δημιουργία περισσότερο πολύπλοκης λειτουργικότητας. Η κεντρική αρχή που διέπει το Composition API είναι η δυνατότητα ταξινόμησης του κώδικα βάσει της λειτουργικότητας, αντί των συστατικών του γραφικού περιβάλλοντος [11]. Αυτή η προσέγγιση διευκολύνει την ανάπτυξη, συντήρηση και δοκιμή της εφαρμογής. Μέσω του Composition API, είναι εφικτό να δημιουργηθούν Composables που περιλαμβάνουν την λογική που σχετίζεται με τα δεδομένα, τις δικτυακές αιτήσεις, την κατάσταση και πολλές άλλες λειτουργίες.

Ένα βασικό πλεονέκτημα του Composition API είναι η δυνατότητα επαναχρησιμοποίησης των Composables. Είναι δυνατόν να δημιουργηθούν ομοιόμορφα Composables που μπορούν να εφαρμοστούν σε διάφορες συνιστώσες (components) και τμήματα μιας εφαρμογής.

Επιπλέον, το Composition API είναι ευκολότερο στη διαχείριση από το Options API, καθώς η λογική μπορεί να οργανωθεί σε πιο μικρά και αυτόνομα Composables. Αυτό διευκολύνει την παρακολούθηση, συντήρηση και εντοπισμό σφαλμάτων στον κώδικα.

Συνολικά, το Composition API της Vue.js αποτελεί μια προηγμένη προσθήκη που δίνει στους προγραμματιστές τη δυνατότητα να γράψουν κώδικα με περισσότερη οργάνωση, αυξημένη

αναγνωσιμότητα και επαναχρησιμοποίηση, βελτιώνοντας την αποτελεσματικότητα και την δυνατότητα συντήρησης των Vue.js εφαρμογών.

2.6 Quasar Framework

Το Quasar Framework, ένα πλαίσιο ανάπτυξης UI για εφαρμογές ιστού και κινητές συσκευές, πρωτοεμφανίστηκε το 2016. Αυτή η πρωτοβουλία επιδίωκε να συνδυάσει τις δυνατότητες του Vue.js με ειδικευμένες λειτουργίες για την κατασκευή εφαρμογών UI, προσφέροντας μια ενιαία εμπειρία ανάπτυξης. Από τότε, η ανάπτυξη του Quasar Framework συνεχίζεται, επεκτείνοντας τη λειτουργικότητά του και προσαρμόζοντάς την στις ανάγκες της ανάπτυξης εφαρμογών.

Ο ρόλος του Quasar Framework [12] είναι χρήσιμος για τους προγραμματιστές που επιθυμούν να αναπτύξουν γρήγορα εφαρμογές UI. Με τον παρέχοντας ένα ολοκληρωμένο σετ στοιχείων UI, δυνατοτήτων διαχείρισης κατάστασης και υποστήριξης PWA (Progressive Web Apps) και εφαρμογών κινητών συσκευών, το πλαίσιο διευκολύνει τη διαδικασία ανάπτυξης.

Υπάρχουν αρκετοί λόγοι για τους οποίους το Quasar Framework μπορεί να προτιμηθεί από άλλα Vue.js πλαίσια όπως το Next.js. Αυτοί οι λόγοι προκύπτουν από τις ιδιαιτερότητες του Quasar και την ευελιξία του σε σχέση με τις ανάγκες των αναπτυσσόμενων εφαρμογών.

Το Quasar Framework προσφέρει ένα πλήρες οικοσύστημα UI, παρέχοντας μια ολοκληρωμένη συλλογή στοιχείων UI που είναι έτοιμα για χρήση. Αυτό το χαρακτηριστικό μπορεί να επιταχύνει την ανάπτυξη εφαρμογών, καθώς δεν απαιτείται η δημιουργία από το μηδέν των στοιχείων. Ένα ακόμη χαρακτηριστικό του Quasar είναι η ενσωματωμένη υποστήριξη για Progressive Web Apps (PWA) και εφαρμογές Cordova. Αυτό δίνει τη δυνατότητα στους προγραμματιστές να δημιουργήσουν εφαρμογές που μπορούν να λειτουργούν είτε ως ιστοσελίδες είτε ως εφαρμογές για κινητές συσκευές.

Επιπλέον, το Quasar προσφέρει ευέλικτη διαχείριση κατάστασης, παρέχοντας προηγμένες δυνατότητες που επιτρέπουν στους προγραμματιστές να δημιουργούν πολύπλοκες εφαρμογές με ευκολία.

Τέλος, η κοινότητα του Quasar είναι ενεργή και πρόθυμη να παρέχει υποστήριξη. Αυτό εξασφαλίζει πληροφορίες, συμβουλές και λύσεις για τυχόν προβλήματα που μπορεί να αντιμετωπίζουν οι προγραμματιστές κατά τη διάρκεια της ανάπτυξης.

Ένα ακόμη σπουδαίο στοιχείο του Quasar είναι η ευκολία που προσφέρει πολυ-πλατφορμική υποστήριξη. Η αξιοποίηση του Quasar Framework για την ανάπτυξη εφαρμογών τόσο για desktop όσο και για κινητές συσκευές επιτυγχάνεται με μια ενοποιημένη προσέγγιση. Οι δυνατότητες του πλαισίου αξιοποιούνται για τη δημιουργία εφαρμογών που προορίζονται να λειτουργούν απρόσκοπτα και επίσης να παρέχουν αποτελεσματική λειτουργικότητα σε δύο διαφορετικές περιβάλλοντες πλατφόρμες.

Στο πλαίσιο αυτό, παρέχονται τα παρακάτω:

- **Ποικιλία Στοιχείων UI:** Το Quasar προσφέρει ένα ευρύ φάσμα στοιχείων που μπορούν να προσαρμοστούν ανάλογα με την πλατφόρμα. Αυτή η πολυμορφία δίνει τη δυνατότητα να δημιουργούνται συνεκτικές εμπειρίες χρήσης σε διαφορετικές συσκευές.
- **Αναγνώριση Συσκευής:** Το πλαίσιο επιτρέπει τον προσδιορισμό του είδους της συσκευής και του περιβάλλοντος λειτουργίας, διευκολύνοντας την προσαρμογή της εμφάνισης και της λειτουργίας της εφαρμογής.
- **Προσαρμοστική Σχεδίαση:** Χρησιμοποιώντας το Quasar, δημιουργείται μια σχεδίαση που προσαρμόζεται αυτόματα σε διάφορα μεγέθη και αναλογίες οθονών.
- **Στοιχεία Διασυννοριακής Χρήσης:** Τα στοιχεία του Quasar είναι σχεδιασμένα να λειτουργούν ομαλά σε διάφορες πλατφόρμες χωρίς την ανάγκη για προσαρμογές. Αυτό εξοικονομεί χρόνο και δύναμη στην ανάπτυξη.
- **Κοινός Κώδικας:** Με την υποστήριξη της Vue.js, το Quasar επιτρέπει τη δημιουργία κοινού κώδικα που μπορεί να χρησιμοποιηθεί από τις δύο πλατφόρμες, εξοικονομώντας χρόνο και πόρους.

Τελικώς, το Quasar Framework αξιοποιεί πλήρως τα πλεονεκτήματα του Vue.js. Εκμεταλλευόμενο την αντιδραστικότητα της Vue.js, δημιουργεί σύνθετες διεπαφές χρήστη και επωφελείται από το φιλικό προς τον προγραμματιστή χαρακτήρα της για την ευκολία συντήρησης και ανάπτυξης. Με αυτόν τον τρόπο, συνδυάζει την ευελιξία της Vue.js με εργαλεία που απλοποιούν την κατασκευή εξαιρετικών εφαρμογών UI. Επίσης, προσφέρει ένα ισχυρό εργαλείο για την ανάπτυξη πολυ-πλατφορμικών εφαρμογών, επιτρέποντας τη δημιουργία εντυπωσιακών εφαρμογών για desktop και κινητές συσκευές, χωρίς την ανάγκη για επαναγραφή του κώδικα για κάθε πλατφόρμα ξεχωριστά.

2.7 MySQL

Η MySQL είναι μια ανοικτού κώδικα σχεσιακή βάση δεδομένων που έχει κερδίσει τη φήμη της ως μία από τις πιο δημοφιλείς και αξιόπιστες λύσεις για τη διαχείριση δεδομένων. Η ιστορία της MySQL [13] ξεκινάει τον Δεκέμβριο του 1994, όταν δύο Σουηδοί προγραμματιστές, ο Michael Widenius και ο David Axmark, δημιούργησαν την αρχική έκδοση του συστήματος. Αρχικά, η MySQL αποτέλεσε τμήμα της επιχείρησης TcX, ωστόσο, το 1995, αποφάσισαν να αναπτύξουν ξεχωριστά το προϊόν και δημιούργησαν την MySQL AB.

Η ανοικτή φύση της MySQL σε συνδυασμό με την απλότητα και την αξιοπιστία της συνέβαλαν στην ταχεία διάδοσή της. Μια από τις βασικές αρχές που καθιέρωσε η MySQL AB ήταν η προσέγγιση της "δωρεάν για όλους, πληρωμή για τους πελάτες που το χρειάζονται". Αυτό επέτρεψε σε πολλές μικρές και μεσαίες επιχειρήσεις, καθώς και σε ατομικούς προγραμματιστές, να αξιοποιήσουν τη βάση δεδομένων χωρίς σημαντικό κόστος.

Ένα ακόμα σημαντικό στοιχείο που συνέβαλε στη δημοφιλία της MySQL ήταν η απόδοση και η ευελιξία της. Η MySQL ήταν σε θέση να ανταποκριθεί σε μεγάλα φορτία εργασίας και απαιτητικές εφαρμογές, καθιστώντας την κατάλληλη για διάφορους τομείς όπως ιστοσελίδες, εφαρμογές λογισμικού και ακόμα και ενσωματωμένα συστήματα.

Το 2008, η Sun Microsystems αγόρασε την MySQL AB, και αργότερα αυτή τέθηκε υπό την ιδιοκτησία της Oracle Corporation με την εξαγορά της Sun το 2010 [14]. Αυτή η κίνηση προκάλεσε ανησυχίες για τη μελλοντική κατεύθυνση και την ανοικτή φύση της MySQL, καθώς η Oracle ήταν εταιρεία που είχε ενδιαφέρον για την εμπορική εκμετάλλευση του προϊόντος.

Παρ' όλα αυτά, η MySQL συνεχίζει να είναι δημοφιλής και ευρέως χρησιμοποιούμενη σε όλον τον κόσμο. Η κοινότητα των προγραμματιστών συνεχίζει να συνεισφέρει στην ανάπτυξη της και να διασφαλίζει την διατήρηση της ανοικτής φύσης της. Με τη διάθεση διάφορων εκδόσεων και εκδόσεων με επιπλέον λειτουργίες για επιχειρήσεις, η MySQL παραμένει μια ισχυρή επιλογή για έργα διάφορων μεγεθών και εφαρμογών. Η συνεχής εξέλιξη και βελτίωση της MySQL από την κοινότητα των προγραμματιστών την κρατά στην προκοπή των τεχνολογικών εξελίξεων.

Με τη διάθεση διάφορων εκδόσεων, συμπεριλαμβανομένων εκδόσεων με επιπλέον λειτουργίες που είναι ειδικά σχεδιασμένες για επιχειρήσεις, η MySQL παρέχει επιλογές που προσαρμόζονται στις ανάγκες κάθε έργου. Αυτό διευκολύνει την επιλογή της κατάλληλης εκδόσεως ανάλογα με την κλίμακα του έργου και τις λειτουργικές απαιτήσεις.

Το γεγονός ότι η MySQL παραμένει μια ανοικτή πλατφόρμα σημαίνει ότι η κοινότητα συνεχίζει να συνεισφέρει στην ανάπτυξή της. Αυτή η συλλογική προσπάθεια διασφαλίζει όχι μόνο τη συνεχή βελτίωση και αναβάθμισή της, αλλά και τη διατήρηση της ασφάλειας και της σταθερότητάς της.

Ο κύριος λόγος επιλογής της MySQL ήταν το γεγονός ότι είναι σχεσιακή βάση δεδομένων. Μια σχεσιακή βάση δεδομένων είναι ένα οργανωμένο σύνολο δεδομένων που αποθηκεύονται σε πίνακες με στήλες και γραμμές. Αυτά τα δεδομένα διαχειρίζονται με τη χρήση της SQL (Structured Query Language), που είναι μια γλώσσα προγραμματισμού για τη διαχείριση, την εξαγωγή και τη μετατροπή δεδομένων σε σχεσιακές βάσεις [15].

Η χρήση των σχεσιακών βάσεων δεδομένων σε μεγάλα έργα επιτρέπει την αποτελεσματική διαχείριση και ανάκτηση των δεδομένων [13], αναδεικνύοντας τη σημασία τους. Οι δομημένοι πίνακες προσφέρουν μια σαφή προσέγγιση στην οργάνωση των πληροφοριών, καθιστώντας τη διαχείριση πιο ευαίσθητη και διαφανή.

Επιπλέον, η ευκολία αναζήτησης και επεξεργασίας μέσω της SQL είναι ένα σημαντικό πλεονέκτημα σε μεγάλα έργα, καθώς οι χρήστες μπορούν να αντλήσουν γρήγορα και αποτελεσματικά πληροφορίες από τη βάση δεδομένων. Η δυνατότητα διατήρησης της συνοχής και ακεραιότητας των δεδομένων προσφέρει επίσης ασφάλεια και αξιοπιστία στα μεγάλα έργα.

Όσον αφορά την κλιμάκωση, η δυνατότητα ανάπτυξης της βάσης δεδομένων με την αύξηση

των δεδομένων είναι απαραίτητη για μεγάλα έργα. Οι σχεσιακές βάσεις δεδομένων παρέχουν τεχνικές όπως ο διαμοιρασμός και η αναπαράληψη που επιτρέπουν τη διαχείριση της αύξησης των δεδομένων με αποτελεσματικό τρόπο.

Μια άλλη σημαντική πτυχή των σχεσιακών βάσεων δεδομένων είναι η ενσωμάτωση δεδομένων μέσω συσχετίσεων. Αυτή η δυνατότητα επιτρέπει τη σύνδεση πληροφοριών από διάφορες πηγές, προσφέροντας μια ολοκληρωμένη εικόνα των δεδομένων για πιο αναλυτικές αποφάσεις.

Κατά συνέπεια, οι σχεσιακές βάσεις δεδομένων προσφέρουν μια ολοκληρωμένη λύση για τη διαχείριση μεγάλων έργων, παρέχοντας εξαιρετική ευελιξία, αξιοπιστία και δομημένη ανάλυση δεδομένων.

2.7.1 MikroORM

Τα Object-Relational Mapping (ORMs) [16] είναι εργαλεία προγραμματισμού λογισμικού που διευκολύνουν την αλληλεπίδραση μεταξύ σχεσιακών βάσεων δεδομένων και αντικειμενοστραφών προγραμμάτων. Αυτά τα εργαλεία μετατρέπουν δεδομένα από τη βάση δεδομένων σε αντικείμενα, που χρησιμοποιούνται πιο εύκολα στον κώδικα της εφαρμογής.

Τα ORMs λειτουργούν ως μεσάζοντες μεταξύ της αντικειμενοστραφούς γλώσσας προγραμματισμού και της σχεσιακής βάσης δεδομένων. Αυτό δίνει τη δυνατότητα στον προγραμματιστή να αλληλεπιδρά με τη βάση δεδομένων χρησιμοποιώντας αντικείμενα, αντί να ασχολείται με πίνακες και γραμμές.

Χρησιμοποιώντας τα ORMs, τα δεδομένα μπορούν να φορτώνονται από τη βάση δεδομένων κατά την ανάγκη (απενεργοποίηση) ή εκ των προτέρων (προετοιμασία). Με αυτόν τον τρόπο, αποφεύγονται περιττές φορτώσεις δεδομένων και βελτιστοποιείται η απόδοση της εφαρμογής. Ένα σημαντικό πλεονέκτημα των ORMs είναι η δυνατότητα διαχείρισης συναλλαγών. Μπορούν να δημιουργηθούν, να επιβεβαιωθούν και να αναιρεθούν συναλλαγές με τρόπο προγραμματιστικά ελεγχόμενο.

Πολλά ORMs υποστηρίζουν πολλαπλές σχεσιακές βάσεις δεδομένων, επιτρέποντάς να αντικαθιστάτε τη βάση δεδομένων χωρίς μεγάλες αλλαγές στον κώδικα. Αυτή η ευελιξία βοηθά στη διατήρηση της εφαρμογής και της βάσης δεδομένων.

Το MikroORM [17] συγκεκριμένα είναι ένα ανοιχτού κώδικα ORM εργαλείο που απλοποιεί τη διαδικασία της αλληλεπίδρασης μεταξύ σχεσιακών βάσεων δεδομένων και αντικειμενοστραφών προγραμμάτων. Το MikroORM σχεδιάστηκε για να παρέχει μια ελαφριά, αλλά ισχυρή προσέγγιση στη διαχείριση των δεδομένων, επιτρέποντας στους προγραμματιστές να εστιάζουν στη λογική της εφαρμογής.

Ένα από τα βασικά χαρακτηριστικά του MikroORM είναι η δυνατότητα του να χρησιμοποιείται με διάφορες σχεσιακές βάσεις δεδομένων, όπως MySQL, PostgreSQL, SQLite και Microsoft SQL Server. Αυτό παρέχει μια μεγάλη ευελιξία για την ανάπτυξη εφαρμογών που απαιτούν

διαφορετικές βάσεις δεδομένων.

Μία από τις ιδιαιτερότητες του MikroORM είναι η υποστήριξή του για TypeScript, που επιτρέπει τη χρήση τύπων και παρέχει σταθερότητα κατά την ανάπτυξη. Το MikroORM χρησιμοποιεί τον αυτόματο εντοπισμό τύπων (type inference) του TypeScript για να εξάγει πληροφορίες για τα μοντέλα και τα δεδομένα από τη βάση δεδομένων, εξαλείφοντας την ανάγκη για μηχανισμούς χειροκίνητης ρύθμισης.

Το MikroORM παρέχει δυνατότητες όπως τη δημιουργία και την αναβάθμιση της βάσης δεδομένων με βάση τα μοντέλα της εφαρμογής, καθώς και τη δυνατότητα διαχείρισης συναλλαγών μέσω του αντίστοιχου μηχανισμού. Επίσης, υποστηρίζει διάφορες σχέσεις μεταξύ των μοντέλων, όπως τις συσχετίσεις "One-to-One," "One-to-Many" και "Many-to-Many," διευκολύνοντας τη διαχείριση των δεδομένων και τις αναζητήσεις μεταξύ τους.

Ο λόγος που οι προγραμματιστές θα μπορούσαν να προτιμήσουν το MikroORM αντί για άλλα ORM είναι λόγω των πλεονεκτημάτων που προσφέρει στον τρόπο που αλληλεπιδρά με τις βάσεις δεδομένων και διευκολύνει την ανάπτυξη λογισμικού.

Ένα από τα βασικά πλεονεκτήματα του MikroORM είναι η ευκολία της ενσωμάτωσης. Η δυνατότητα να ορίσετε τα μοντέλα σας με συνήθεις JavaScript κλάσεις καθιστά την αρχιτεκτονική του κώδικα πιο φυσική και κατανοητή. Αυτό επιτρέπει στους προγραμματιστές να αναπτύξουν με ευκολία και να ενσωματώνουν το MikroORM στα έργα τους.

Επίσης, το MikroORM προσφέρει αυτόματη δημιουργία πινάκων, εξοικονομώντας χρόνο και κόπο κατά την ανάπτυξη και τη συντήρηση της βάσης δεδομένων.

Τέλος, η φιλοσοφία του MikroORM να προσφέρει απλότητα και ευκολία χωρίς πολυπλοκές ρυθμίσεις μπορεί να είναι ελκυστική για προγραμματιστές που αναζητούν ένα εργαλείο που να παρέχει τις βασικές λειτουργίες ORM χωρίς να προσθέτει περιττή πολυπλοκότητα.

Συνολικά, το MikroORM προσφέρει ένα συνδυασμό απλότητας, ευελιξίας και αποτελεσματικότητας που μπορεί να είναι ελκυστικός για προγραμματιστές που αναζητούν ένα εύκολο στη χρήση και αποδοτικό ORM για τις ανάγκες τους.

Κεφάλαιο 3

Σχεδιασμός και Υλοποίηση του ScholarPath

3.1 Αρχιτεκτονική

Η ενδεδειγμένη και αποτελεσματική ανάπτυξη του ScholarPath επιδεικνύει μια σοφή και προσεκτική επιλογή αρχιτεκτονικής. Σε αυτό το πλαίσιο, εντοπίζεται ένα ξεχωριστό τμήμα, το client, το οποίο λειτουργεί ως διεπαφή χρήστη και διασφαλίζει την απρόσκοπτη αλληλεπίδραση με τον χρήστη. Από αυτό το σημείο, η επικοινωνία προς και από τη βάση δεδομένων γίνεται μέσω του Node.js API, το οποίο είναι υπεύθυνο για την επεξεργασία όλων των δεδομένων και την αλληλεπίδραση με τη βάση.

Η επιλογή αυτής της αρχιτεκτονικής αντικατοπτρίζει τη βαθιά συνειδητοποιημένη ανάγκη για μια δομή που είναι ταυτόχρονα ευέλικτη, ασφαλής και περιεκτική. Αντιλαμβάνεται την αναγκαιότητα για αποτελεσματική διαχείριση των διαδικασιών που σχετίζονται με τα μεταπτυχιακά προγράμματα σπουδών, παρέχοντας τη βάση για μια ολοκληρωμένη προσέγγιση.

Η κεντρική σπονδυλική στήλη αυτής της αρχιτεκτονικής έχει τις ρίζες της στη χρήση του GraphQL και του Koa για τη δημιουργία ενός αποδοτικού backend. Η προσέγγιση του GraphQL προσφέρει ευελιξία στους καθηγητές για τη δημιουργία και διαχείριση προγραμμάτων σπουδών και αιτήσεων, ενώ εξασφαλίζει σαφήνεια και ακρίβεια στις αιτήσεις των υποψήφιων διδασκτόρων. Παράλληλα, η επιλογή του Koa ως web framework προσφέρει μια ευέλικτη και αποδοτική λύση για τη διαχείριση των HTTP αιτημάτων.

Το MikroORM εμπεριέχεται σε αυτήν την αρχιτεκτονική ως το εργαλείο επικοινωνίας με τη βάση δεδομένων MySQL. Η σαφής οργάνωση των οντοτήτων και των πεδίων τους μέσω του MikroORM αναδεικνύει την ακρίβεια και τη δομημένη προσέγγιση που απαιτούνται για την αποτελεσματική διαχείριση των προγραμμάτων σπουδών και των αιτήσεων.

Ο προσεκτικά εξειδικευμένος συντονισμός ανάμεσα στο frontend και το backend επιτυγχάνεται με αρμονία μέσω του Axios. Η μεταφορά και επεξεργασία των HTTP αιτημάτων ανατίθεται

στους resolvers του server, οι οποίοι συσχετίζονται με τους κατάλληλους τύπους αιτημάτων και προσφέρουν τις απαραίτητες αποκρίσεις με μορφή HTTP.

Ωστόσο, η διασφάλιση της ασφάλειας και των δεδομένων απαιτεί συνεχή προσοχή. Η απόφαση για την υιοθέτηση του HTTPS ακόμη κατά την ανάπτυξη καταδεικνύει μια σφαιρική προσέγγιση προς την προστασία των δεδομένων.

Σε ολόκληρο το φάσμα της αρχιτεκτονικής και του σχεδιασμού, η εφαρμογή ScholarPath αντλεί την δύναμη της από την δυναμική επεκτατικότητα, παρέχοντας μια βάση για την περαιτέρω ανάπτυξη των λειτουργιών της και την αντιμετώπιση νέων προκλήσεων.

3.2 Σχεδίαση Βάσης Δεδομένων

Ο σχεδιασμός της Βάσης Δεδομένων έχει υλοποιηθεί με προσεκτικό τρόπο, δημιουργώντας συνολικά επτά διακριτούς πίνακες που αναλύουν και περιγράφουν την λειτουργία της εφαρμογής. Η επιλογή της MySQL, μιας σχεσιακής βάσης δεδομένων, επιτρέπει την αξιοποίηση των θετικών χαρακτηριστικών των σχεσιακών βάσεων, εξασφαλίζοντας έτσι όχι μόνο την δομημένη οργάνωση των οντοτήτων, αλλά και τις ενδεδειγμένες σχέσεις μεταξύ τους.

Στη συνέχεια παρουσιάζονται οι πίνακες της βάσης δεδομένων, συνοδευόμενοι από τα βασικά τους στοιχεία.

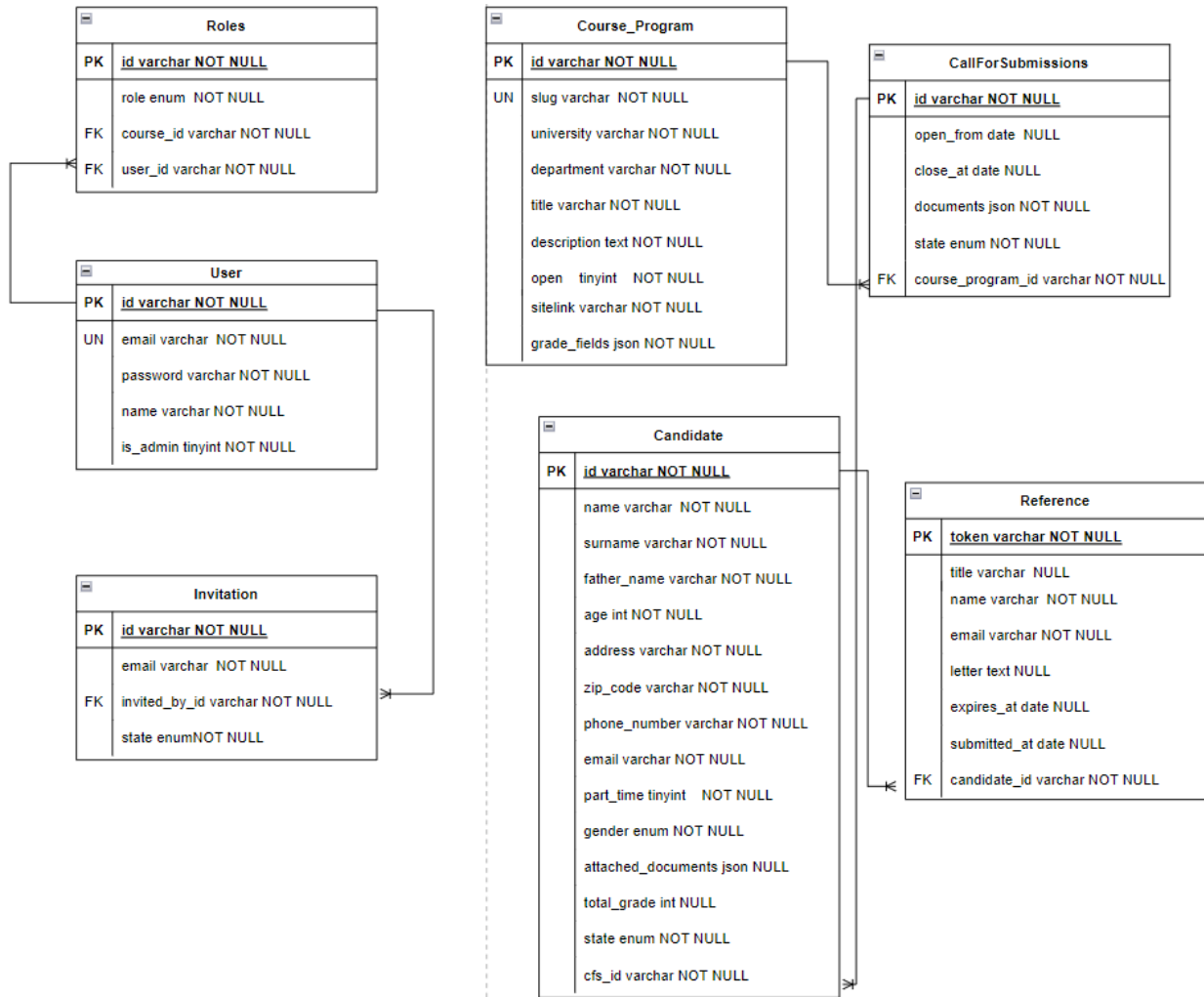
3.2.1 Πρόγραμμα Σπουδών (Course Program)

Ένας από τους πίνακες είναι το "Course Program", ο οποίος αντιπροσωπεύει τα προγράμματα σπουδών. Ο πίνακας αυτός περιλαμβάνει διάφορα πεδία, όπως το πρωτεύον κλειδί (id), και το πεδίο "slug". Το "slug" είναι μια μοναδική τιμή σε κάθε εγγραφή του πίνακα και χρησιμοποιείται ως URL στη σελίδα του client. Επιπλέον, υπάρχουν και άλλα πεδία όπως το "title" και το "description", τα οποία χρησιμοποιούνται για την παροχή πληροφοριών στους υποψήφιους σχετικά με το κάθε Πρόγραμμα Μεταπτυχιακών Σπουδών (ΠΜΣ).

Σημαντικό είναι επίσης το πεδίο "gradeFields", το οποίο έχει τύπο JSON. Αυτό το πεδίο προσδίδει ευελιξία στους καθηγητές που διαχειρίζονται το κάθε ΠΜΣ, καθώς τους επιτρέπει να επιλέγουν τα κριτήρια αξιολόγησης και να καθορίζουν την σχετική σημασία κάθε κριτηρίου. Με αυτόν τον τρόπο, επιτυγχάνεται η ακρίβεια και η εξατομίκευση της διαδικασίας αξιολόγησης για κάθε πρόγραμμα σπουδών.

3.2.2 Κάλεσμα για Αιτήσεις (Call For Submissions)

Ο πίνακας "CallForSubmissions" (CFS) αποτελεί ένα κρίσιμο στοιχείο του συστήματος, καθώς καθορίζει το χρονικό πλαίσιο κατά το οποίο κάθε ΠΜΣ δέχεται αιτήσεις από υποψηφίους. Στη σχέση του με κάθε ΠΜΣ, παρουσιάζει ένα πολύ-προς-ένα (Many-to-One) μοντέλο. Ο χρονικός



Σχήμα 3.1: Μοντέλο Οντοτήτων-Συσχετίσεων

περίορισμός κάθε CFS καθορίζεται από τα πεδία "openFrom" και "closeAt", τα οποία αποτελούν χρονοσφραγίδες (datetime) στη βάση δεδομένων.

Επιπλέον, το πεδίο "state" είναι ένα enum, το οποίο καθορίζει την κατάσταση ενός CFS. Η αρχική τιμή είναι το "published", υποδεικνύοντας ότι το CFS έχει δημιουργηθεί αλλά δεν είναι ακόμη διαθέσιμο για αιτήσεις. Οι καταστάσεις "open" και "closed" υποδεικνύουν αντίστοιχα ότι το CFS βρίσκεται σε περίοδο υποβολής αιτήσεων και ότι η περίοδος αιτήσεων έχει λήξει. Τέλος, η κατάσταση "done" υποδεικνύει ότι οι αιτήσεις έχουν αξιολογηθεί και οι υποψήφιοι έχουν ενημερωθεί για την αποδοχή τους.

Ακόμη, το πεδίο "documents" αντιπροσωπεύει ένα πεδίο τύπου JSON, το οποίο περιγράφει τα αρχεία που απαιτούνται για την υποβολή μιας αίτησης (π.χ. βιογραφικό, αποδείξεις πτυχίου) καθώς και τον αριθμό των απαιτούμενων συστατικών επιστολών. Ο ρόλος αυτού του πεδίου είναι να διευκολύνει τους υποψηφίους στην υποβολή πλήρων αιτήσεων με όλα τα απαιτούμενα

στοιχεία.

Είναι προφανές ότι για να είναι δυνατή η υποβολή μιας αίτησης για ένα συγκεκριμένο ΠΜΣ, πρέπει να υπάρχει ένα ενεργό CFS για αυτό. Κατά συνέπεια, εδραιώνεται και μία ακόμη σχέση μεταξύ των πινάκων, μία ένα-προς-πολλά (One-to-Many) μεταξύ του πίνακα "Candidates" και του πίνακα "CallForSubmissions".

3.2.3 Υποψήφιος (Candidate)

Ο πίνακας "Candidate", όπως αναφέρθηκε, διαδραματίζει τον ρόλο της αντιπροσώπευσης των υποψηφίων που υποβάλλουν αίτηση για εισαγωγή σε ένα Πρόγραμμα Μεταπτυχιακών Σπουδών (ΠΜΣ). Δεδομένου ότι δεν απαιτείται η δημιουργία λογαριασμού χρήστη πριν από την υποβολή της αίτησης, ο πίνακας αυτός αποθηκεύει τις βασικές πληροφορίες του κάθε υποψηφίου στη βάση δεδομένων. Πεδία όπως το όνομα, η ηλικία, η διεύθυνση κ.ά. καταγράφονται εδώ.

Επιπλέον, ο πίνακας "Candidate" περιέχει το πεδίο "attachedDocuments", το οποίο είναι ένα JSON πεδίο και λειτουργεί ως αποθηκευτικός χώρος για τα PDF αρχεία που ζητούνται από το σχετικό "CallForSubmissions" (CFS). Αυτό το πεδίο αντιπροσωπεύει τη συνολική συλλογή των απαιτούμενων εγγράφων που πρέπει να προσκομίσει ο κάθε υποψήφιος μαζί με την αίτησή του, όπως το βιογραφικό και αποδεικτικά διπλωμάτων.

Εντός της βάσης δεδομένων, ο πίνακας "Candidate" διατηρεί μία σχέση ένα-προς-πολλά (One-to-Many) με τον πίνακα "Reference". Αυτό σημαίνει ότι για κάθε υποψήφιο, μπορεί να υπάρχουν πολλαπλές συστατικές επιστολές που έχουν υποβληθεί. Ο πίνακας "Reference" αντιπροσωπεύει τις συστατικές επιστολές που συνοδεύουν τις αιτήσεις των υποψηφίων, και αυτές οι επιστολές είναι συνδεδεμένες με τη σχετική αίτηση του κάθε υποψηφίου.

3.2.4 Συστατική Επιστολή (Reference)

Στις περιπτώσεις όπου ένα Πρόγραμμα Μεταπτυχιακών Σπουδών απαιτεί την υποβολή συστατικής επιστολής, αναπτύσσεται μια διαδικασία όπου για κάθε συστατική επιστολή που υποβάλλεται από τον υποψήφιο, δημιουργείται αντίστοιχη εγγραφή στη βάση δεδομένων. Η αυτόματη δημιουργία αυτών των εγγραφών διευκολύνει την πλήρη διαχείριση των συστατικών επιστολών μέσω της πλατφόρμας ScholarPath, εξασφαλίζοντας παράλληλα την ευκολία πρόσβασης για τους υποψηφίους.

Για κάθε συστατική επιστολή, δημιουργείται μία εγγραφή στη βάση δεδομένων. Καθοριστικό ρόλο στην διαχείριση των εγγραφών αυτών αναλαμβάνει το "token", το οποίο αποτελεί το κύριο κλειδί της εγγραφής. Το "token" αποτελεί μια μοναδική αναγνωριστική τιμή για κάθε συστατική επιστολή, δίνοντας τη δυνατότητα να ανατρέχουμε με ακρίβεια σε κάθε σχετική εγγραφή στη βάση.

Οι πληροφορίες που αποθηκεύονται για κάθε συστατική επιστολή περιλαμβάνουν προσωπικά στοιχεία του ατόμου που συντάσσει την επιστολή. Επιπλέον, το πεδίο "letter" αποτελεί ένα πεδίο τύπου "text" στη βάση δεδομένων. Αυτή η επιλογή επιτρέπει την αποθήκευση του περιεχομένου της συστατικής επιστολής χωρίς περιορισμούς ως προς τον αριθμό των χαρακτήρων. Καθώς οι συστατικές επιστολές μπορεί να περιλαμβάνουν λεπτομερείς πληροφορίες και αξιολογήσεις, το πεδίο αυτό διασφαλίζει την ακριβή αποθήκευση του περιεχομένου και τη διατήρηση της πληρότητας της πληροφορίας.

3.2.5 Χρήστης (User)

Ο πίνακας "user" αναπαριστά τις διάφορες οντότητες των χρηστών που απαιτούν επιτυχημένη σύνδεση στην εφαρμογή. Για να πραγματοποιήσει μία επιτυχημένη σύνδεση, ένας χρήστης αναμένεται να καταχωρίσει το δικό του ηλεκτρονικό ταχυδρομείο (email), το οποίο πρέπει να είναι μοναδικό στη βάση δεδομένων, καθώς και τον αντίστοιχο κωδικό πρόσβασής του (password). Για λόγους ενίσχυσης της ασφάλειας, ο κωδικός αποθηκεύεται στη βάση δεδομένων με τη μορφή κρυπτογραφημένου κειμένου.

Οι κατηγορίες των χρηστών μπορεί να περιλαμβάνουν τους καθηγητές, οι οποίοι αναλαμβάνουν τη διαχείριση των ΠΜΣ, καθώς και αυτούς που έχουν το δικαίωμα να αξιολογούν αιτήσεις. Η διάκριση αυτών των δύο ρόλων πραγματοποιείται μέσω του πεδίου "isAdmin". Το πεδίο αυτό λειτουργεί ως δείκτης του εάν ένας χρήστης διαθέτει διαχειριστικά δικαιώματα (isAdmin = true) ή όχι (isAdmin = false), επιτρέποντας την κατάλληλη αντιστοίχιση καθήκοντος με την αντίστοιχη εξουσία.

3.2.6 Ρόλοι (Roles)

Ο πίνακας "Roles" λειτουργεί ως ενδιάμεσος σχεσιακός πίνακας που ενώνει τις σχέσεις μεταξύ των πινάκων "user" και "role". Με μια σχέση πολλά-προς-ένα, ο πίνακας αυτός αποτελεί το γέφυρα που διασυνδέει κάθε χρήστη με τον αντίστοιχο ρόλο του.

Κάθε εγγραφή στον πίνακα "Roles" αποτελεί μια σύνδεση μεταξύ ενός χρήστη και ενός ρόλου. Αυτό επιτρέπει τη διαχείριση πολλαπλών ρόλων για κάθε χρήστη, επιτρέποντας την εκχώρηση ποικίλων εξουσιών και αρμοδιοτήτων.

Επιπλέον, ο πίνακας "Roles" περιλαμβάνει το πεδίο "role", το οποίο καθορίζει τον ρόλο του χρήστη. Μέσω της χρήσης enum τιμών, αυτό το πεδίο αναγνωρίζει εάν ένας χρήστης έχει την εξουσία του διαχειριστή (admin) ή του Moderator σε ένα συγκεκριμένο ΠΜΣ. Με αυτόν τον τρόπο, καθορίζεται και επικυρώνεται η δυναμική ανάθεση ρόλων σε κάθε χρήστη, εξασφαλίζοντας την ορθή εκτέλεση των αρμοδιοτήτων στο πλαίσιο της εφαρμογής.

3.2.7 Πρόσκληση (Invitation)

Ο πίνακας "invitations" αποτελεί έναν αποθηκευτικό χώρο για τους χρήστες που λαμβάνουν προσκλήσεις μέσω ηλεκτρονικού ταχυδρομείου. Με μια σχέση πολλά-προς-ένα με τον πίνακα "users", ο πίνακας αυτός επιτρέπει τον εύκολο εντοπισμό του χρήστη που έχει αποστείλει την πρόσκληση σε κάθε παραλήπτη.

Μέσω της σχέσης πολλά-προς-ένα με τον πίνακα "users", είναι δυνατό να εντοπιστεί ποιος χρήστης έχει πραγματοποιήσει την πρόσκληση για κάθε παραλήπτη. Αυτό εξασφαλίζει τη διατήρηση του ιστορικού των προσκλήσεων και τη δυνατότητα παρακολούθησης των αλληλεπιδράσεων μεταξύ των χρηστών.

Επιπλέον, σε περίπτωση που απαιτείται η σύνδεση του πίνακα "invitations" με τον χρήστη που δημιουργήθηκε μέσω της πρόσκλησης, αυτό μπορεί να επιτευχθεί μέσω του ηλεκτρονικού ταχυδρομείου (email). Αυτό το μοναδικό χαρακτηριστικό επιτρέπει την αναγνώριση του χρήστη που προέρχεται από μια συγκεκριμένη πρόσκληση, δημιουργώντας μια συνδεσιμότητα μεταξύ του χρήστη και του δημιουργού της πρόσκλησης.

3.3 Λειτουργικές Απαιτήσεις

Η εφαρμογή ScholarPath σχεδιάστηκε με τον στόχο να αναλάβει την ολοκληρωμένη διαχείριση των Προγραμμάτων Μεταπτυχιακών Σπουδών. Σε αυτό το πλαίσιο, καθορίζονται διάφοροι ρόλοι για τους χρήστες του συστήματος, προσδίδοντας συγκεκριμένες λειτουργικότητες σε κάθε κατηγορία. Αναλυτικότερα, οι ακόλουθοι ρόλοι έχουν καθοριστεί:

1. **Admins (Διαχειριστές):** Οι διαχειριστές αντιπροσωπεύουν τους καθηγητές και επιστημονικούς υπεύθυνους που διαχειρίζονται τα ΠΜΣ. Έχουν την εξουσία να διαχειρίζονται τις λειτουργίες των προγραμμάτων, να προσθέτουν νέα ΠΜΣ, να επεξεργάζονται τις λεπτομέρειες των προγραμμάτων και να διαχειρίζονται τις αιτήσεις από υποψήφιους.
2. **Moderators (Συντονιστές):** Οι συντονιστές είναι χρήστες που έχουν προσκληθεί από τους διαχειριστές. Ο ρόλος τους είναι να αξιολογούν τις αιτήσεις που υποβάλλονται από τους υποψήφιους για τα ΠΜΣ που έχουν το κατάλληλο ρόλο. Αυτό επιτρέπει την αποτελεσματική διαχείριση του αξιολογητικού διαδικαστικού.
3. **Candidates (Υποψήφιοι):** Οι υποψήφιοι αναφέρονται σε άτομα που εκδηλώνουν ενδιαφέρον για συμμετοχή σε ΠΜΣ. Ο ρόλος αυτός τους επιτρέπει να υποβάλλουν αιτήσεις για συμμετοχή σε συγκεκριμένα προγράμματα και να παρακολουθούν τη διαδικασία αξιολόγησης.
4. **Invited (Κλήση με Email):** Οι χρήστες που λαμβάνουν κλήση μέσω email ανήκουν σε αυτήν την κατηγορία. Σε αυτούς περιλαμβάνονται άτομα που έχουν κληθεί να υποβάλλουν συστατικές επιστολές για υποψήφιους.

Αυτοί οι ρόλοι επιτρέπουν την κατάλληλη οργάνωση των λειτουργιών της εφαρμογής, εξασφα-

λίζοντας ότι κάθε κατηγορία χρηστών έχει πρόσβαση μόνο στις απαραίτητες λειτουργίες. Με αυτόν τον τρόπο, το ScholarPath διευκολύνει τη διαχείριση και αξιοποίηση των ΠΜΣ, εξασφαλίζοντας την ομαλή λειτουργία της διαδικασίας επιλογής και διαχείρισης.

3.3.1 Διαχειριστές (Admins)

Ο ρόλος του διαχειριστή (Admin) στο ScholarPath αναδεικνύεται ως ο κυρίαρχος και πιο σημαντικός ρόλος της εφαρμογής. Οι διαχειριστές αντιπροσωπεύουν τους καθηγητές και τους επιστημονικούς υπεύθυνους των Προγραμμάτων Μεταπτυχιακών Σπουδών (ΠΜΣ) και επιτρέπεται να διαχειρίζονται τα προγράμματα Σπουδών που έχουν δημιουργήσει, καθώς και τους χρήστες που έχουν προσκαλέσει στην πλατφόρμα. Η ιδιαιτερότητα αυτού του ρόλου είναι ότι οι διαχειριστές έχουν πρόσβαση και επιρροή μόνο στα ΠΜΣ που έχουν δημιουργήσει ή στα οποία έχουν προσκαλέσει να συμμετάσχουν. Αυτό διασφαλίζει την αποκέντρωση της διαχείρισης των προγραμμάτων και τη διατήρηση της ανεξαρτησίας των διαφόρων ΠΜΣ.

Σε ό,τι αφορά τις λειτουργικές απαιτήσεις των διαχειριστών στο πλαίσιο των ΠΜΣ, κάθε χρήστης που έχει τον ρόλο του διαχειριστή έχει τη δυνατότητα να δημιουργεί νέα Προγράμματα Σπουδών μέσω της σελίδας /courses. Σε αυτήν τη σελίδα, συμπληρώνονται βασικές πληροφορίες σχετικά με το κάθε ΠΜΣ, όπως ο τίτλος του προγράμματος, το πανεπιστήμιο που το φιλοξενεί, το τμήμα που ανήκει, καθώς και μια περιγραφή του προγράμματος. Επίσης, οι διαχειριστές μπορούν να προσθέσουν κριτήρια αξιολόγησης τα οποία χρησιμοποιούνται στην αξιολόγηση των αιτήσεων που υποβάλλονται για το πρόγραμμα αυτό. Για κάθε πεδίο αξιολόγησης που προστίθεται, πρέπει να έχει μοναδικό τίτλο, καθώς αυτός χρησιμοποιείται ως αναγνωριστικό πεδίο σε άλλους πίνακες, όπως ο πίνακας Candidate.

Επιπλέον, οι διαχειριστές έχουν τη δυνατότητα να ορίσουν τις τιμές που μπορούν να πάρουν αυτά τα πεδία αξιολόγησης κατά την αξιολόγηση των αιτήσεων. Οι προεπιλεγμένες τιμές καθορίζονται συνήθως στο εύρος 0-5. Είναι σημαντικό να σημειωθεί ότι, κατά τη διάρκεια της ύπαρξης ενός ανοιχτού Call For Submission (CFS) και όσο υπάρχουν αιτήσεις που δεν έχουν αξιολογηθεί ακόμη, δεν επιτρέπεται η αλλαγή των πεδίων αξιολόγησης. Αυτό καθιστά τη διαδικασία αξιολόγησης διαφανή και δίκαιη, αποτρέποντας τυχόν ανισότητες στα κριτήρια αξιολόγησης.

Η διαχείριση των Προγραμμάτων Μεταπτυχιακών Σπουδών (ΠΜΣ) στο ScholarPath και η διαγραφή ενός ΠΜΣ είναι μια λειτουργία που πρέπει να διέπεται από συγκεκριμένες προϋποθέσεις. Συγκεκριμένα, μόνο ο διαχειριστής που δημιούργησε ένα συγκεκριμένο ΠΜΣ έχει τη δυνατότητα να το επεξεργαστεί και να το διαγράψει. Ωστόσο, για να εξασφαλιστεί ότι η διαγραφή ενός ΠΜΣ δεν θα έχει αρνητικές επιπτώσεις, υπάρχει η προϋπόθεση ότι το ΠΜΣ αυτό δεν πρέπει να είναι ανοιχτό, δηλαδή να μην δέχεται αιτήσεις από υποψηφίους.

Αυτή η προϋπόθεση διασφαλίζει ότι η διαγραφή ενός ΠΜΣ δεν θα επηρεάσει υποψηφίους που έχουν υποβάλει αιτήσεις για αυτό το ΠΜΣ. Κατόπιν της διαγραφής, το συγκεκριμένο ΠΜΣ δεν

θα υπάρχει στην εφαρμογή. Ωστόσο, αν ένα ΠΜΣ εξακολουθεί να δέχεται αιτήσεις, ο διαχειριστής δεν έχει τη δυνατότητα να το διαγράψει προκειμένου να διασφαλιστεί η συνέχιση της διαδικασίας αξιολόγησης των υποβληθεισών αιτήσεων χωρίς παρεμβολές

Η δημιουργία και λειτουργία των Call For Submissions (CFS) αποτελεί βασικό στοιχείο για τη διαχείριση των αιτήσεων στα Προγράμματα Μεταπτυχιακών Σπουδών (ΠΜΣ) στο ScholarPath. Ο διαχειριστής, που στην περίπτωση αυτή είναι ο καθηγητής που διαχειρίζεται το ΠΜΣ, μπορεί εύκολα να δημιουργήσει ένα CFS μέσω της σελίδας του αντίστοιχου ΠΜΣ.

Όταν δημιουργεί ένα CFS, ο διαχειριστής καθορίζει το χρονικό διάστημα κατά το οποίο το ΠΜΣ θα δέχεται αιτήσεις. Αυτό γίνεται μέσω των πεδίων "openFrom" και "closeAt," τα οποία καθορίζουν την ημερομηνία και την ώρα έναρξης και λήξης της περιόδου υποβολής αιτήσεων αντίστοιχα.

Επιπλέον, ο διαχειριστής μπορεί να επιλέξει ποια έγγραφα πρέπει να υποβάλλουν οι υποψήφιοι μαζί με την αίτησή τους. Αυτές οι επιλογές περιλαμβάνουν:

- Απόδειξη Πτυχίου: Υποχρεωτική απόδειξη του προηγούμενου πτυχίου του υποψηφίου.
- Απόδειξη άλλων Masters: Εάν απαιτείται, ο υποψήφιος πρέπει να υποβάλει απόδειξη από προηγούμενα μεταπτυχιακά προγράμματα.
- Συστατικές Επιστολές: Εάν απαιτείται, ο διαχειριστής καθορίζει πόσες συστατικές επιστολές χρειάζονται από τους υποψηφίους. Αυτός ο αριθμός πρέπει να είναι θετικός αριθμός μεγαλύτερος του μηδενός.

Η λειτουργία των Call For Submissions (CFS) είναι κρίσιμη για τη διαχείριση των αιτήσεων στα Προγράμματα Μεταπτυχιακών Σπουδών (ΠΜΣ) στο ScholarPath και προσφέρει ευελιξία στους διαχειριστές των ΠΜΣ για τη διαχείριση των περιόδων υποβολής αιτήσεων.

Πριν την έναρξη της περιόδου υποβολής, τα CFS είναι ορατά για όλους τους επισκέπτες της σελίδας, και μπορούν να δουν πότε θα ανοίξει η περίοδος υποβολής αιτήσεων για το συγκεκριμένο ΠΜΣ, ενώ στην περίπτωση που είναι ήδη ανοιχτό μπορούν να δουν μέχρι πότε είναι η διορία. Αυτή η διαφάνεια είναι σημαντική για τους υποψηφίους και επιτρέπει σε όλους να προγραμματίσουν την υποβολή των αιτήσεών τους.

Όσον αφορά τους διαχειριστές των ΠΜΣ, έχουν τη δυνατότητα να επεξεργαστούν τα CFS πριν την έναρξη της περιόδου υποβολής. Αυτό σημαίνει ότι μπορούν να αλλάξουν τις ημερομηνίες έναρξης και λήξης κατά τη διάρκεια αυτής της περιόδου καθώς και τις απαιτήσεις των αιτήσεων. Σε περίπτωση που είναι ήδη ανοιχτό και δέχεται αιτήσεις, μπορούν να επεκτείνουν το διάστημα υποβολής αιτήσεων, καθορίζοντας μια νέα ημερομηνία λήξης που πρέπει να είναι αργότερα από την αρχική.

Είναι σημαντικό να σημειώσουμε ότι μετά την επιτυχή επέκταση της περιόδου υποβολής, η λήξη δεν μπορεί να αλλάξει σε μια νεότερη ημερομηνία, αλλά μπορεί να επεκταθεί ξανά για μια ημερομηνία λήξης που είναι αργότερα από την τρέχουσα. Αυτό εξασφαλίζει ότι οι υποψήφιοι θα έχουν επαρκές χρόνο για να υποβάλουν τις αιτήσεις τους και δεν θα υπάρχουν αλλαγές

στις απαιτήσεις κατά τη διάρκεια της περιόδου υποβολής.

Οι διαχειριστές (admins) φέρουν μια σημαντική ευθύνη στη διαχείριση των χρηστών της εφαρμογής. Καταρχάς, έχουν τη δυνατότητα να προσκαλέσουν νέους χρήστες συμπληρώνοντας το email τους. Αυτό είναι χρήσιμο για να επεκτείνουν την κοινότητα των χρηστών της εφαρμογής. Επιπλέον, οι διαχειριστές έχουν τη δυνατότητα να δουν ποιες προσκλήσεις εκκρεμούν και ποιες έχουν δεχτεί από τους χρήστες του συστήματος. Αυτό τους παρέχει μια σαφή εικόνα της δραστηριότητας των χρηστών στην πλατφόρμα.

Σημαντικότερο ακόμη, οι διαχειριστές μπορούν να διαχειριστούν τις αντιστοιχίσεις των χρηστών σε συγκεκριμένα Προγράμματα Μεταπτυχιακών Σπουδών. Αυτό σημαίνει ότι μπορούν να αναθέσουν σε ορισμένους χρήστες το ρόλο του "moderator" για συγκεκριμένα ΠΜΣ ή ακόμη και να τους παραχωρήσουν δικαιώματα διαχείρισης (admin) για την εφαρμογή. Σε περίπτωση που επιλεγθεί ο δεύτερος ρόλος, οι χρήστες αποκτούν όλα τα δικαιώματα που διαθέτουν και οι υπόλοιποι διαχειριστές (admins), και αυτά τα δικαιώματα δεν μπορούν να αφαιρεθούν. Αυτό εξασφαλίζει ότι οι χρήστες με αυτόν τον ρόλο μπορούν να συμμετέχουν πλήρως στη διαχείριση των συγκεκριμένων ΠΜΣ και να συνεισφέρουν στην ομαλή λειτουργία της εφαρμογής.

3.3.2 Συντονιστές (Moderators)

Ο ρόλος των moderators αποτελεί ένα σημαντικό κομμάτι της διαχείρισης των Προγραμμάτων Μεταπτυχιακών Σπουδών (ΠΜΣ) και παρέχει αναγκαία λειτουργικά χαρακτηριστικά για τη σωστή εκτέλεση της εφαρμογής. Παρόλο που οι moderators διαδραματίζουν ρόλο παρόμοιο με αυτόν των admins, οι ευθύνες τους είναι περιορισμένες και επικεντρώνονται σε συγκεκριμένες λειτουργίες και περιοχές ευθύνης.

Στο πλαίσιο της ανάλυσης των λειτουργικών απαιτήσεων της εφαρμογής, πρέπει να δούμε τον ρόλο των moderators ως έναν απαραίτητο συνεργάτη των admins. Τα κύρια καθήκοντά τους εστιάζονται στην αξιολόγηση των υποψηφίων για τα συγκεκριμένα ΠΜΣ που διαχειρίζονται. Πιο συγκεκριμένα, οι moderators έχουν την ευθύνη για την αξιολόγηση των αιτήσεων που υποβάλλονται στα εν λόγω ΠΜΣ. Μέσω του scholarPath, αξιολογούν τα διάφορα στοιχεία των αιτήσεων βάσει των προκαθορισμένων κριτηρίων, καταχωρώντας τα αποτελέσματα της αξιολόγησής τους. Αυτή η ενέργεια παρέχει στους admins και τους υπόλοιπους συνεργάτες την αναγκαία ενημέρωση για την πρόοδο των αιτήσεων και την κατάσταση του κάθε υποψηφίου. Παρόλο που οι moderators δραστηριοποιούνται κυρίως στον τομέα της αξιολόγησης, είναι σημαντικό να τονίσουμε ότι αποτελούν αναπόσπαστο κομμάτι της ομάδας που διασφαλίζει την ομαλή διεξαγωγή των ΠΜΣ. Μέσω της συνεργασίας τους με τους υπόλοιπους moderators και τους admins, ενισχύουν την διαδικασία αξιολόγησης, συμβάλλουν στην επίλυση προβλημάτων και επιτρέπουν την αποτελεσματική διαχείριση των ΠΜΣ.

Τονίζεται ότι, παρά τον περιορισμένο ρόλο τους, οι moderators εξακολουθούν να αποτελούν κρίσιμο μέλος της ομάδας διαχείρισης των ΠΜΣ, προσφέροντας την αναγκαία υποστήριξη στη

διαδικασία των αιτήσεων και τη διασφάλιση της διαφάνειας και ακρίβειας της αξιολόγησης.

3.3.3 Υποψήφιοι (Candidates)

Ο ρόλος των υποψηφίων (candidates) στην εφαρμογή ScholarPath αποτελεί έναν σημαντικό πυλώνα της διαδικασίας διαχείρισης των Μεταπτυχιακών Προγραμμάτων Σπουδών (ΠΜΣ).. Στο πλαίσιο αυτό, ορίζεται ως υποψήφιος οποιοσδήποτε χρήστης επισκέπτεται το ScholarPath με σκοπό να εξερευνήσει τα διάφορα Μεταπτυχιακά Προγράμματα που προσφέρονται.

Ένα από τα κύρια χαρακτηριστικά που προσφέρει η εφαρμογή είναι η συγκεντρωμένη πληροφορία σχετικά με τα ΠΜΣ. Οι υποψήφιοι μπορούν να αποκτήσουν εκτενή εικόνα για τα διάφορα ΠΜΣ που προσφέρονται, όπως τα πανεπιστήμια στα οποία φιλοξενούνται και τα αντίστοιχα τμήματα στα οποία ανήκουν.

Το ScholarPath παρέχει στους υποψηφίους τη δυνατότητα να πλοηγηθούν με αυτονομία στην εφαρμογή χωρίς την ανάγκη να δημιουργήσουν λογαριασμό. Αυτό διευκολύνει την πρόσβαση τους σε πληροφορίες σχετικά με τα διάφορα ΠΜΣ και συμβάλλει στη δημιουργία μιας ολοκληρωμένης εικόνας για τις διαθέσιμες σπουδές.

Αυτή η δυνατότητα επιτρέπει στους υποψηφίους να αναζητήσουν πληροφορίες σχετικά με τα ΠΜΣ που τους ενδιαφέρουν, να ενημερωθούν για τις πανεπιστημιακές επιλογές τους και να προετοιμαστούν για την διαδικασία εγγραφής. Με αυτόν τον τρόπο, οι υποψήφιοι αποκτούν ολοκληρωμένη εικόνα και πληροφορίες που τους βοηθούν να πάρουν ενημερωμένες αποφάσεις σχετικά με τις μεταπτυχιακές σπουδές τους.

Η διαδικασία για τους υποψηφίους που ενδιαφέρονται για ένα συγκεκριμένο ΠΜΣ είναι σχεδιασμένη για να παρέχει εξαιρετική εμπειρία χρήστη, επιτρέποντάς τους να βρουν όλες τις απαραίτητες πληροφορίες και να υποβάλουν αποτελεσματικά την αίτησή τους.

Μετά την αναζήτηση ενός συγκεκριμένου ΠΜΣ, οι υποψήφιοι μπορούν να εισέλθουν στη σελίδα του ΠΜΣ όπου διατίθενται περαιτέρω λεπτομέρειες που έχουν δώσει οι διαχειριστές (admins) για το συγκεκριμένο Πρόγραμμα. Εκεί, παρέχονται σημαντικές πληροφορίες σχετικά με το τρέχον Call For Submission (CFS), εάν υπάρχει. Εδώ, οι υποψήφιοι μπορούν να βρουν τις ημερομηνίες που ανακοινώνονται για το άνοιγμα και το κλείσιμο των αιτήσεων, παρέχοντας έτσι σαφείς κατευθύνσεις για την υποβολή της αίτησης.

Η φόρμα υποβολής αίτησης σχεδιάστηκε για να είναι απλή και κατανοητή, δίνοντας τη δυνατότητα στους υποψηφίους να συμπληρώσουν εύκολα τα απαραίτητα πεδία. Αυτά περιλαμβάνουν προσωπικές πληροφορίες όπως όνομα, φύλο, ηλικία, προπτυχιακός τίτλος, και άλλα σχετικά. Εάν απαιτείται, μπορεί να ζητηθεί από τους υποψηφίους να ανεβάσουν αρχεία όπως αποδείξεις πτυχίου, βιογραφικό, ή άλλα μεταπτυχιακά προγράμματα που ενδέχεται να έχουν ολοκληρώσει. Η διαδικασία ανεβάσματος αρχείων είναι απλή και ευέλικτη μέσω της εφαρμογής.

Επιπλέον, σε περίπτωση που απαιτείται η υποβολή συστατικών επιστολών, οι υποψήφιοι κα-

λούνται να συμπληρώσουν το όνομα και το email των καθηγητών που επιθυμούν να συμπεριληφθούν.

Η διαδικασία υποβολής αίτησης στο scholarPath είναι σχεδιασμένη για να εξασφαλίσει την ομαλή και σωστή υποβολή των αιτήσεων από τους υποψηφίους. Εάν κάποιο από τα πεδία της φόρμας δεν συμπληρώνεται σωστά, οι υποψήφιοι λαμβάνουν άμεσα ένα μήνυμα λάθους που τους ενημερώνει για το πρόβλημα. Αυτό εξασφαλίζει ότι οι αιτήσεις υποβάλλονται με τις απαραίτητες και σωστές πληροφορίες.

Σε περίπτωση που όλα τα πεδία συμπληρώνονται σωστά και η φόρμα υποβάλλεται επιτυχώς, οι υποψήφιοι ανακατευθύνονται αυτόματα σε μια διαφορετική σελίδα που τους ενημερώνει ότι η υποβολή της αίτησης ήταν επιτυχής. Ταυτόχρονα, ενημερώνονται ότι θα λάβουν ένα email επιβεβαίωσης για τα αποτελέσματα της αίτησης.

Το σύστημα επίσης είναι σχεδιασμένο για να στέλνει αυτόματα email ενημέρωσης στους υποψηφίους μόλις ο admin ολοκληρώσει το CFS. Αυτό το email περιέχει σημαντικές πληροφορίες σχετικά με την αποδοχή της αίτησης, δηλαδή εάν ο υποψήφιος έχει γίνει δεκτός στο ΠΜΣ ή εάν βρίσκεται σε λίστα αναμονής. Αυτή η διαδικασία ενημέρωσης μέσω email εξασφαλίζει ότι οι υποψήφιοι έχουν σαφή πληροφόρηση για την κατάσταση της αίτησής τους και μπορούν να λάβουν τα απαραίτητα μέτρα ανάλογα με το αποτέλεσμα.

Το ScholarPath απλοποιεί την αναζήτηση και την αίτηση για μεταπτυχιακά προγράμματα, καθιστώντας τη διαδικασία πραγματικά εύκολη και αποτελεσματική. Με λίγα κλικ, οι υποψήφιοι έχουν πρόσβαση σε ολοκληρωμένες πληροφορίες για τα ΠΜΣ που τους ενδιαφέρουν και τη δυνατότητα να υποβάλουν αιτήσεις με απλότητα.

3.3.4 Κλήση με Email (Invited)

Η τελευταία κατηγορία χρηστών που πρέπει να εξετάσουμε είναι αυτή που κλήθηκαν να επισκεφθούν το ScholarPath μέσω του ηλεκτρονικού ταχυδρομείου τους. Αυτοί οι χρήστες μπορεί να ανήκουν είτε στην κατηγορία των καθηγητών, οι οποίοι έχουν προσκληθεί από τους διαχειριστές (admins) για να αναλάβουν το ρόλο του moderator σε συγκεκριμένα Προγράμματα Μεταπτυχιακών Σπουδών (ΠΜΣ), είτε στην κατηγορία των καθηγητών που έχουν επιλεγεί από υποψηφίους για τη σύνταξη συστατικών επιστολών.

Στην πρώτη περίπτωση, οι καθηγητές λαμβάνουν ένα ηλεκτρονικό μήνυμα που τους ενημερώνει για την πρόσκληση που τους απευθύνθηκε, καθώς και τα στοιχεία του διαχειριστή που τους προσκάλεσε. Στο μήνυμα περιλαμβάνεται επίσης ένας μοναδικός σύνδεσμος εγγραφής, προκειμένου να ολοκληρώσουν τη διαδικασία εγγραφής τους στο ScholarPath. Αφού συμπληρώσουν τα απαραίτητα δεδομένα και ολοκληρώσουν την εγγραφή τους, εμφανίζεται ένα μήνυμα επιτυχίας, και από αυτήν τη στιγμή μπορούν να συνδεθούν χρησιμοποιώντας το email και τον κωδικό που μόλις δημιούργησαν.

Στη δεύτερη περίπτωση, καθηγητές λαμβάνουν πάλι ένα αυτοματοποιημένο email με ένα μο-

ναδικό σύνδεσμο, προκειμένου να συμπληρώσουν τη συστατική επιστολή. Αυτό το email περιλαμβάνει επίσης πληροφορίες σχετικά με τον υποψήφιο που ζήτησε τη συστατική επιστολή και το συγκεκριμένο Πρόγραμμα Μεταπτυχιακών Σπουδών (ΠΜΣ) για το οποίο απαιτείται η συστατική.

Όταν οι καθηγητές ακολουθούν τον σύνδεσμο, μεταφέρονται σε μια σελίδα της εφαρμογής όπου συμπληρώνουν τα απαραίτητα προσωπικά στοιχεία και τη συστατική επιστολή που θέλουν να υποβάλλουν. Με την επιτυχή υποβολή, ανακατευθύνονται σε μια νέα σελίδα όπου ενημερώνονται για την επιτυχή υποβολή, και τότε ολοκληρώνεται η διαδικασία.

Σε περίπτωση που η πρόσκληση για τη συστατική έχει λήξει ή έχει ήδη συμπληρωθεί, οι επισκέπτες αντί για τη σελίδα συμπλήρωσης βλέπουν αμέσως μια σελίδα που τους ενημερώνει για την αντίστοιχη κατάσταση, και δεν έχουν τη δυνατότητα να συμπληρώσουν τη συστατική επιστολή.

3.4 Υλοποίηση Backend

Το API που υλοποιήθηκε αποτελεί ένα Node.js εφαρμογή, το οποίο δομήθηκε χρησιμοποιώντας TypeScript, το πλαίσιο εργασίας Koa και την τεχνολογία GraphQL. Η οργάνωση των φακέλων της εφαρμογής βασίστηκε στη λειτουργία τους, ένα σχήμα που επιλέχθηκε για να διασφαλίσει την ευανάγνωστη και επεκτάσιμη δομή του κώδικα.

Στο βασικό επίπεδο του ριζικού καταλόγου της εφαρμογής βρίσκονται κυρίως τα αρχεία διαμόρφωσης που είναι απαραίτητα για την ανάπτυξη και την αρμονική λειτουργία της εφαρμογής. Ένα σημαντικό αρχείο σε αυτό το επίπεδο είναι το "package.json," το οποίο περιλαμβάνει πληροφορίες σχετικά με την εφαρμογή, όπως το όνομα, την έκδοση και τις βιβλιοθήκες του Node.js που απαιτούνται για την απρόσκοπτη λειτουργία της εφαρμογής.

Αυτή η δομή επιτρέπει την ευέλικτη προσθήκη νέων χαρακτηριστικών στο μέλλον, καθώς κάθε λειτουργική ενότητα της εφαρμογής βρίσκεται οργανωμένη σε έναν ξεχωριστό φάκελο, ενώ τα αρχεία που την αφορούν βρίσκονται εκεί.

Ο φάκελος "src" αποτελεί τον πυρήνα της εφαρμογής, καθώς περιέχει τον πηγαίο κώδικα που αναπτύχθηκε για την εφαρμογή. Σημαντικό ρόλο παίζει το αρχείο "index.ts," το οποίο λειτουργεί ως αρχικό σημείο εκκίνησης (entry point) της εφαρμογής. Σε αυτό το αρχείο περιέχονται τις βασικές πληροφορίες σχετικά με την εφαρμογή και αρμοδιότητες όπως η ρύθμιση του HTTP server και του Apollo server.

Ουσιαστικά, το "index.ts" εκκινεί την εφαρμογή και είναι υπεύθυνο για τη σύνδεση των διαφόρων συστατικών της. Είναι το σημείο εκκίνησης της διαδικασίας εξυπηρέτησης αιτημάτων HTTP και της χρήσης του GraphQL server για τη διαχείριση των δεδομένων. Τα middlewares που έχουν συγκαταλεγεί σε αυτό το αρχείο επιτρέπουν τη διεκπεραίωση των αιτημάτων HTTP και την αποδοχή των GraphQL ερωτημάτων.

Με τον τρόπο αυτό, το "index.ts" διαδραματίζει κρίσιμο ρόλο στην εκκίνηση και την ομαλή λειτουργία της εφαρμογής σας, καθιστώντας τον κεντρικό σημείο εναρμόνισης όλων των λειτουργιών.

```
const apolloServer = new ApolloServer({
  schema,
  context ({ ctx }: { ctx: Context }): CustomContext {
    return {
      ctx,
      state: ctx.state,
      em: connection.em.fork()
    }
  },
  uploads: false
})

const app = new Koa()
if (ENVIRONMENT === 'production') {
  app.proxy = true
}

app.use(uploadsRouter.routes())

app.use(cors())
app.use(graphqlUploadKoa())
app.use(bodyParser({ jsonLimit: '150mb' }))
app.use(setAuthUser)

app.use(apolloServer.getMiddleware({ cors: false }))
const httpServer = createServer(app.callback())

httpServer.listen({ port: PORT }, () => {
  console.log(`http://${HOST}:${PORT}/graphql`)
})
```

Σχήμα 3.2: Middleware και αρχικοποίηση server στο backend/src/index.ts

Στον φάκελο "types," βρίσκονται όλες οι κλάσεις και διεπαφές (interfaces) που χρησιμοποιεί η εφαρμογή σας. Αυτές οι δομές καθορίζουν τους τύπους δεδομένων που χρησιμοποιούνται

κατά την εκτέλεση του κώδικα.

Πιο συγκεκριμένα, ο φάκελος "entities" αντιπροσωπεύει τις κλάσεις που απεικονίζουν τα δεδομένα που αποθηκεύονται στη βάση δεδομένων της εφαρμογής σας. Εδώ καθορίζονται οι διάφοροι τύποι πεδίων, οι σχέσεις μεταξύ των οντοτήτων, και άλλες σχετικές πληροφορίες σχετικά με τη δομή της βάσης δεδομένων όπως απεικονίζονται στην εικόνα 3.3.

Στον φάκελο "classes," βρίσκουμε ακόμη και τα input fields που αντιπροσωπεύουν το σχήμα των δεδομένων που αναμένουν οι resolvers. Αυτές οι κλάσεις προσδιορίζουν τη δομή των εισερχομένων δεδομένων και τα κριτήρια επικύρωσης για αυτά. Η χρήση του "class-validator" είναι κρίσιμη για την εξασφάλιση ότι τα δεδομένα που παρέχονται είναι έγκυρα.

Αυτό το μέτρο είναι σημαντικό για την ασφάλεια και τη σωστή λειτουργία της εφαρμογής σας. Με το "class-validator," ελέγχεται ότι τα δεδομένα που παρέχονται από τους πελάτες πληρούν τις απαιτούμενες προδιαγραφές και δεν παρουσιάζουν προβλήματα επικύρωσης. Σε περίπτωση που δεν τηρούνται οι κανόνες επικύρωσης, επιστρέφεται αντίστοιχο μήνυμα σφάλματος στον πελάτη, προσφέροντας έτσι μια βελτιωμένη εμπειρία χρήστη και εξασφαλίζοντας τη σταθερότητα της εφαρμογής σας.

Αυτή η προσέγγιση επιτρέπει την αποτροπή δυσλειτουργιών και προβλημάτων ασφαλείας που μπορεί να προκύψουν από ανεπίκυρα δεδομένα και είναι συνήθης πρακτική σε ποιοτικές εφαρμογές.

Ο φάκελος "lib" περιλαμβάνει τα αρχεία που σχετίζονται με τη λειτουργία της εφαρμογής. Συγκεκριμένα, ο φάκελος "resolvers" περιέχει όλους τους resolvers της εφαρμογής. Σε αυτόν τον φάκελο, ο resolver περιλαμβάνει όλα τα queries και τα mutations. Ωστόσο, για την εκτέλεση του επιχειρησιακού λογισμικού και των αντίστοιχων ενεργειών, καλούνται αντίστοιχες μέθοδοι που βρίσκονται στον φάκελο "lib/actions".

Στον φάκελο "tasks" βρίσκουμε επιπλέον λειτουργίες που καλούνται από τις ενέργειες, αλλά δεν έχουν άμεση σχέση με αυτές. Αυτές οι λειτουργίες μπορεί να περιλαμβάνουν τη διαχείριση του ανεβάσματος αρχείων, την εφαρμογή αυθεντικοποίησης (authentication guards) και άλλες παρόμοιες λειτουργίες.

Αυτή η δομή σχεδιάζεται έτσι ώστε να επιτρέπει την οργάνωση και τη συντήρηση του κώδικα με τρόπο που διευκολύνει την κατανόηση και την αποσφαλμάτωση της εφαρμογής. Η διαχωρισμένη δομή μεταξύ resolvers, actions και tasks επιτρέπει στους προγραμματιστές να εργαστούν ανεξάρτητα σε διάφορα μέρη της εφαρμογής χωρίς να επηρεάζουν αμέσως τον κώδικα σε άλλα τμήματα. Αυτό συμβάλλει στην ευκολία συντήρησης και στην αποφυγή σφαλμάτων κατά την ανάπτυξη της εφαρμογής.

Τα παραδείγματα κώδικα 3.4, 3.5 αντιπροσωπεύουν τον τρόπο με τον οποίο αναπτύχθηκε το backend της εφαρμογής σας. Στον φάκελο "resolvers", ο κώδικας υπεύθυνος για την επίλυση των GraphQL queries και mutations λαμβάνει τα απαιτούμενα δεδομένα από τα εισερχόμενα αιτήματα. Επιπλέον, εδώ εφαρμόζονται κατάλληλοι "guards" που ελέγχουν προϋποθέσεις όπως η

```
@Entity()
@ObjectType()
export class CallForSubmissions {
    @PrimaryKey()
    @Field(() => ID)
    id: string = v4()

    @Property({ nullable: true })
    @Field(() => Date, { nullable: true })
    openFrom?: Date

    @Property({ nullable: true })
    @Field(() => Date, { nullable: true })
    closeAt?: Date

    @Embedded(() => AdditionalFiles, { object: true })
    @Field(() => GraphQLJSONObject, { nullable: true })
    documents?: AdditionalFiles

    @Enum(() => CFS_State)
    @Field(() => CFS_State)
    state: CFS_State

    @ManyToOne(() => CourseProgram)
    @Field(() => CourseProgram)
    courseProgram: CourseProgram

    @Field(() => [Candidate])
    @OneToMany(() => Candidate, candidates => candidates.cfs)
    candidates = new Collection<Candidate>(this)
}
```

Σχήμα 3.3: Απόσπασμα κώδικα από το Call For Submissions Entity

ανάγνωση του συνδεδεμένου χρήστη για διακριτικές δικαιώματα, όπως η διαχειριστική πρόσβαση (admin).

```

@Query(() => [CourseProgram])
async getCourseByAdmin (
  @Ctx('em') em: EntityManager,
  @Ctx('ctx') ctx: AuthCustomContext
): Promise<CourseProgram[]> {
  isAdmin(ctx.user)
  return await getCourseByAdminAction(ctx.user.id, em)
}

@Mutation(() => CourseProgram)
async createCourseProgram (
  @Ctx('em') em: EntityManager,
  @Ctx('ctx') ctx: AuthCustomContext,
  @Arg('gradeFields', () => [GradeFieldsInput]) gradeFields: GradeFieldsInput[],
  @Arg('data') data: CourseProgramInput
): Promise<CourseProgram> {
  isAdmin(ctx.user)
  return await createCourseProgramAction(data, gradeFields, ctx.user, em)
}

```

Σχήμα 3.4: Απόσπασμα κώδικα από τον CourseProgramResolver

Στον φάκελο "actions," βρίσκουμε την πραγματική επιχειρησιακή λογική της εφαρμογής. Εδώ γίνεται η εξέταση των δεδομένων, ο έλεγχος σωστότητας, και η αντιμετώπιση σφαλμάτων κατάλληλα. Τα actions αναλαμβάνουν την επικοινωνία με τη βάση δεδομένων, όπου είναι απαραίτητο, και εκτελούν τις κατάλληλες ενέργειες για να εκπληρωθούν τα αιτήματα των resolvers. Από άποψη κώδικα, η υλοποίηση της αποστολής μαζικών email με τη χρήση του event loop του Node.js είναι ένα εξαιρετικό παράδειγμα ασύγχρονης λειτουργίας που επιτρέπει στην εφαρμογή να λειτουργεί αποτελεσματικά χωρίς καθυστερήσεις.

Μέσω της χρήσης των Promises, όπως φαίνεται στην εικόνα 3.6, η εφαρμογή μπορεί να ξεκινήσει τη διαδικασία αποστολής email για πολλούς φοιτητές χωρίς να περιμένει το αποτέλεσμα της κάθε αποστολής. Αυτό εξασφαλίζει ότι η εφαρμογή δεν θα παραμείνει αναμένοντας μέχρι να ολοκληρωθεί κάθε αποστολή, αλλά μπορεί να συνεχίσει να εκτελεί άλλες διαδικασίες, όπως την επεξεργασία άλλων αιτημάτων.

Αυτή η ασύγχρονη προσέγγιση είναι κρίσιμη για εφαρμογές που απαιτούν αποτελεσματική χρήση του χρόνου εκτέλεσης και αποφυγή καθυστερήσεων. Επιτρέπει στην εφαρμογή να αποκρίνεται γρήγορα στις αιτήσεις των χρηστών, ενώ ταυτόχρονα εκτελεί παράλληλα και αποδοτικά διάφορες διαδικασίες, όπως η αποστολή emails. Αυτό συμβάλλει στη βελτίωση της εμπειρίας του χρήστη και την αποφυγή τυχόν αρνητικών επιπτώσεων στην απόδοση της εφαρμογής σας.

```
export async function createCourseProgramAction (
  data: CourseProgramInput,
  gradeFields: GradeFieldsInput[],
  user: User,
  em: EntityManager
): Promise<CourseProgram> {
  const validSlugFlag = await validateSlugs(data.slug, em)
  if (!validSlugFlag) {
    throw new UserInputError('INVALID_SLUG')
  }

  const course = em.create(CourseProgram, {
    slug: data.slug,
    university: data.university,
    department: data.department,
    title: data.title,
    description: data.description,
    sitelink: data.sitelink,
    open: false,
    gradeFields
  })
  em.persist(course)

  const role = em.create(Roles, {
    role: UserRole.admin,
    user: user.id,
    course
  })
  em.persist(role)

  await em.flush()
  return course
}
```

Σχήμα 3.5: Απόσπασμα κώδικα από την υλοποίηση του Create Course Program Mutation

```
const cfs = await em.findOneOrFail(CallForSubmissions, { id: data.cfsId }, { populate: ['courseProgram'] })

const promises = []
for (let i = 0; i < accepted.length; i++) {
  accepted[i].state = Candidate_State.approved
  promises.push(acceptedEmailContent(data, accepted[i], cfs, totalCount))
}
for (let i = 0; i < declined.length; i++) {
  declined[i].state = Candidate_State.waitList
  promises.push(declinedEmailContent(totalCount, data, declined[i], data.capacity + i + 1, cfs.courseProgram.title))
}

Promise.all(promises)
  .catch(err => { console.log(err) })

cfs.state = CFS_State.done
```

Σχήμα 3.6: Απόσπασμα κώδικα για την αποστολή emails στους υποψηφίους.

3.5 Υλοποίηση Frontend

Η δομή του frontend διατηρεί ορισμένες ομοιότητες με τον backend, παρέχοντας έναν οργανωμένο τρόπο για την υλοποίηση της εφαρμογής. Στο root επίπεδο, βρίσκονται κυρίως τα αρχεία διαμόρφωσης (config files) που ρυθμίζουν την συμπεριφορά του frontend. Στον φάκελο "src," αναπτύσσεται ο κώδικας της εφαρμογής.

Ο φάκελος "pages" περιέχει τις σελίδες της εφαρμογής που είναι προσβάσιμες μέσω των αντίστοιχων διευθύνσεων URL. Εδώ, εφαρμόζεται η αρχιτεκτονική του Quasar framework σε συνδυασμό με τη Vue 3 και το Composition API. Ο κώδικας του frontend διαχωρίζεται συνήθως σε δύο τμήματα: το "template," που παράγει το HTML που βλέπουμε στην ιστοσελίδα, και το "script," που περιέχει τη JavaScript λογική της σελίδας.

```

<q-table
  grid
  :rows="result"
  :columns="columns"
  :loading="loading"
  row-key="name"
>
  <template v-slot:item="props">
    <div class="q-pa-xs col-12 cursor-pointer" @click="redirectToPage(props.row.slug)">
      <q-card>
        <q-card-section class="text-center">
          <div class="row items-center no-wrap">
            <div class="col text-h6">{{ props.row.title }}</div>
            <div class="col-auto v-if="createdByLogged(props.row.admin) ">
              <q-btn color="grey-7" round size="sm" flat icon="edit" @click.stop="CreateProgramCourseDialogRef.open(props.row)" />
            </div>
          </div>
          <div class="row items-center no-wrap">
            <div class="col text-subtitle2 text-grey-9">{{ props.row.university }}</div>
            <div class="col-auto v-if="createdByLogged(props.row.admin) && !props.row.open">
              <q-btn color="grey-7" round size="sm" flat icon="delete" @click.stop="confirmDelete(props.row.id)" />
            </div>
          </div>
        </q-card-section>
      </q-card>
    </div>
  </template>
</q-table>

```

Σχήμα 3.7: Απόσπασμα κώδικα την παρουσίαση του πίνακα με τα Προγράμματα Σπουδών.

Αυτή η διάρθρωση επιτρέπει την ξεκάθαρη αποσαφήνιση της λογικής της σελίδας και της δομής της. Το "template" καθορίζει την εμφάνιση της σελίδας, ενώ το "script" περιλαμβάνει την λειτουργικότητα της. Αυτό διευκολύνει την συντήρηση του κώδικα, την ανάπτυξη νέων σελίδων και την αντιμετώπιση σφαλμάτων. Επίσης, η χρήση του Quasar framework παρέχει ένα στέρεο οικοσύστημα εργαλείων για τη δημιουργία γρήγορων και αποδοτικών εφαρμογών. Τα components αποτελούν αυτόνομα, επαναχρησιμοποιούμενα τμήματα κώδικα που ενσωματώνονται σε μια σελίδα της εφαρμογής. Αυτή η προσέγγιση διασφαλίζει τη διατήρηση της καθαρότητας του κώδικα και την καλή δομή του, επιτρέποντας ταυτόχρονα την αποφυγή επα-

```

const routes = [
  {
    path: '/',
    component: () => import('layouts/MainLayout.vue'),
    props: true,
    children: [
      {
        path: '',
        component: () => import('pages/Index.vue'),
        redirect: to => { return { path: '/courses' } }
      },
      { path: '/courses', component: () => import('pages/ProgramCourses.vue') },
      { path: '/courses/:slug', component: () => import('pages/ShowCourseProgram.vue') },
      {
        path: '/courses/:slug/candidates',
        component: () => import('pages/Candidates.vue'),
        beforeEnter: [isLoggedIn]
      },
      { path: '/reference/:token', component: () => import('pages/Reference.vue') },
      {
        path: '/reference/info',
        name: 'referenceInfo',
        component: () => import('pages/static-content/ReferenceInfo.vue')
      },
      { path: '/register/:token', component: () => import('pages/Registration.vue') },
      { path: '/submission/success', component: () => import('pages/static-content/SuccessfulSubmission.vue') },
      {
        path: '/users',
        component: () => import('pages/InvitedUsers.vue'),
        beforeEnter: [isAdmin]
      }
    ]
  }
],

```

Σχήμα 3.8: τα διαθέσιμα routes της εφαρμογής στον browser.

ναλήψεων.

Η χρήση των components επιτρέπει την οργανωμένη ανάπτυξη της εφαρμογής, καθώς κάθε component αναλαμβάνει συγκεκριμένες λειτουργίες ή εμφανίσεις. Αυτό διευκολύνει τον προγραμματιστή να διαχειριστεί τον κώδικα και να αποφύγει τον περιττό συσσωρευμένο κώδικα σε μια σελίδα. Επιπλέον, τα components είναι επαναχρησιμοποιήσιμα, επιτρέποντας την ανακύκλωση τους σε διάφορα μέρη της εφαρμογής χωρίς την ανάγκη για επαναγραφή κώδικα. Συνολικά, αυτή η προσέγγιση συμβάλλει στη βελτιστοποίηση της ανάπτυξης, τη μείωση του κινδύνου σφαλμάτων και τη βελτίωση της συντήρησης του κώδικα, προσφέροντας έναν αποτελεσματικό τρόπο για τη δημιουργία ποιοτικών εφαρμογών.

Ένα ακόμη κρίσιμο στοιχείο για την υλοποίηση του Frontend αποτελούν τα GraphQL ερωτήματα, τα οποία αποστέλλονται στον server μέσω του Axios. Μέσω αυτών των ερωτημάτων, το Frontend αιτείται συγκεκριμένα δεδομένα από τον server προκειμένου να τα αποθηκεύσει και να τα χρησιμοποιήσει στη συνέχεια.

Τα δεδομένα που επιστρέφονται από τον server αποθηκεύονται σε μεταβλητές στο Frontend, καθιστώντας τα προσβάσιμα για χρήση στην πλευρά του client. Αυτό επιτρέπει την δυναμική εμφάνιση και αλληλεπίδραση του χρήστη με την εφαρμογή, καθώς μπορεί να εμφανίζονται και να ανανεώνονται δεδομένα από τον server χωρίς την ανάγκη για ανανέωση της σελίδας.

Η χρήση του GraphQL σε συνδυασμό με το Axios παρέχει μια ευέλικτη και αποτελεσματική λύση για την ανταλλαγή δεδομένων μεταξύ Frontend και Backend, βοηθώντας έτσι στην δημιουργία δυναμικών και αποτελεσματικών εφαρμογών που προσφέρουν μια βελτιωμένη εμπειρία χρήστη.

Για ορισμένες κρίσιμες λειτουργίες, όπως η διαχείριση του συνδεδεμένου χρήστη και οι ενέργειες που σχετίζονται με τους υποψηφίους (candidates), χρησιμοποιήθηκε το Pinia store. Το Pinia store αποτελεί ένα σημαντικό μέρος της αρχιτεκτονικής της εφαρμογής, και χρησιμοποιείται για τη διαχείριση των δεδομένων. Είναι ένα εργαλείο που παρέχει τη δυνατότητα αποθήκευσης και ανανέωσης δεδομένων στην πλευρά του client χωρίς την ανάγκη για συνεχείς αιτήσεις προς τον server.

Ένα από τα βασικά πλεονεκτήματα του Pinia store είναι η δυνατότητα να διαχειρίζεται τα δεδομένα με ομαλό και αποτελεσματικό τρόπο. Επιτρέπει στην εφαρμογή να αντιδρά ευέλικτα σε αλλαγές και ενημερώσεις χωρίς να προκαλεί διακοπές στην εμπειρία του χρήστη. Συγκεκριμένα, στην περίπτωση της αξιολόγησης των υποψηφίων, η χρήση του Pinia store επιτρέπει στους καθηγητές να παρακολουθούν και να ενημερώνουν τα δεδομένα των υποψηφίων με άμεσο και αποτελεσματικό τρόπο. Αυτό βελτιώνει την αποτελεσματικότητα και την ταχύτητα της αξιολόγησης, καθιστώντας τη διαδικασία πιο ευέλικτη και προσαρμόσιμη.

Η χρήση του Pinia store συμβάλλει στη βελτίωση της εμπειρίας των χρηστών και την ομαλή λειτουργία της εφαρμογής, καθιστώντας την αξιόπιστη και αποτελεσματική.

```
export const getCourseProgramBySlug = gql`
  query getCourcebySlug ($slug: String!) {
    course: getCourcebySlug(slug: $slug) {
      id
      slug
      university
      department
      title
      description
      open
      sitelink
      title
      gradeFields
      currentCFS {
        id
        openFrom
        closeAt
        state
        documents
      }
      roles {
        id
        user {
          name
        }
        role
      }
      cfs {
        id
      }
    }
  }
`
```

Σχήμα 3.9: GraphQL για το getCourcebySlug query

```
const fetchBySlug = async (slug) => {
  try {
    loading.value = true
    const response = await api({
      url: '',
      method: 'POST',
      data: {
        query: print(getCourseProgramBySlug),
        variables: {
          slug: slug
        }
      }
    })

    if (response.data.data) {
      return response.data.data.course
    }
  } catch (err) {
    console.log('Error while fetching course program.')
  } finally {
    loading.value = false
  }
}
```

Σχήμα 3.10: HTTP request με axios για το getCoursebySlug query

```
const { fetchBySlug } = fetchAllProgramCourses()

onMounted(() => {
  slug.value = route.params.slug
  fetchBySlug(slug.value).then(res => { course.value = res })
})
```

Σχήμα 3.11: Απόσπασμα κώδικα με την κλήση του fetchBySlug hook βάση του url μέσα στο script της σελίδας

```
export const candidatesStore = defineStore('candidates', {
  state: () => ({
    candidates: []
  }),
  getters: {
    getCandidates: (state) => state.candidates
  },
  actions: {
    async fetchCandidates (cfsId) {
      this.candidates = await fetchAllCandidates().fetch(cfsId)
    },
    async gradeCandidate (id, data) {
      const res = await useCandidateMutations().gradeCandidateMutation(id, data)

      if (!res) return

      const index = this.candidates.findIndex(candidate => candidate.id === id)

      this.candidates[index] = { ...this.candidates[index], totalGrade: res.totalGrade }
    },
    clearState () {
      this.candidates = []
    }
  }
})

export default candidatesStore
```

Σχήμα 3.12: Κώδικας του candidate store

3.6 Ο κώδικας του ScholarPath στο Github

Ο πηγαίος κώδικας του ScholarPath φιλοξενείται στο GitHub και είναι ανοικτός για το κοινό. Τα αντίστοιχα αποθετήρια για το backend και το frontend βρίσκονται στο ίδιο αποθετήριο (repository), αλλά για τη λειτουργία τους απαιτείται να γίνει ξεχωριστή εγκατάσταση για το καθένα.

Μπορείτε να βρείτε τον πηγαίο κώδικα της εφαρμογής στη σελίδα: <https://github.com/nasia-sam/SholarPath>

Για να μπορέσετε να τρέξετε το ScholarPath τοπικά, θα πρέπει να έχετε εγκατεστημένα τα εξής:

- MySQL Server (μπορεί να εγκατασταθεί εύκολα μέσω του Docker)
- Node.js
- npm (Node Package Manager)
- TypeScript
- Quasar Framework
- git

Για να κατεβάσετε το repository:

```
git clone https://github.com/nasia-sam/pms-applications.git
```

Για να ξεκινήσετε τοπικά τον server σε λειτουργία ανάπτυξης (dev mode):

```
cd backend
cp .env.example .env
npm i
npm dev
```

Για να ξεκινήσετε τοπικά τον client σε λειτουργία ανάπτυξης (dev mode):

```
cd ui
cp .env.example .env
npm i
quasar dev
```

Όπως συμβαίνει και με πολλά άλλα έργα ανοιχτού λογισμικού, το ScholarPath μπορεί να δημιουργήσει μια νέα και δυναμική κοινότητα που θα αποτελείται από συνεισφέροντες (contributors) που μπορούν να ενισχύσουν και να βελτιώσουν τις λειτουργίες της εφαρμογής. Αυτή η κοινότητα μπορεί να συνεργαστεί με σκοπό την εξέλιξη και τη βελτίωση του ScholarPath, καθιστώντας το ένα πιο ισχυρό εργαλείο για τους χρήστες του.

Οι contributors μπορούν να συνεισφέρουν με την προσθήκη νέων λειτουργιών, την επίλυση προβλημάτων (bug fixes), την ανάπτυξη νέων υποσυστημάτων, και πολλά άλλα. Με αυτόν τον τρόπο, το ScholarPath μπορεί να επωφεληθεί από τη συλλογική σοφία και τον ενθουσιασμό πολλών προγραμματιστών και προγραμματίστριων που θέλουν να συμβάλουν στην ανάπτυξη της εφαρμογής.

Η δημιουργία μιας ανοικτής κοινότητας μπορεί να επιταχύνει την εξέλιξη του ScholarPath και να το καταστήσει ακόμη πιο χρήσιμο για τους χρήστες του.

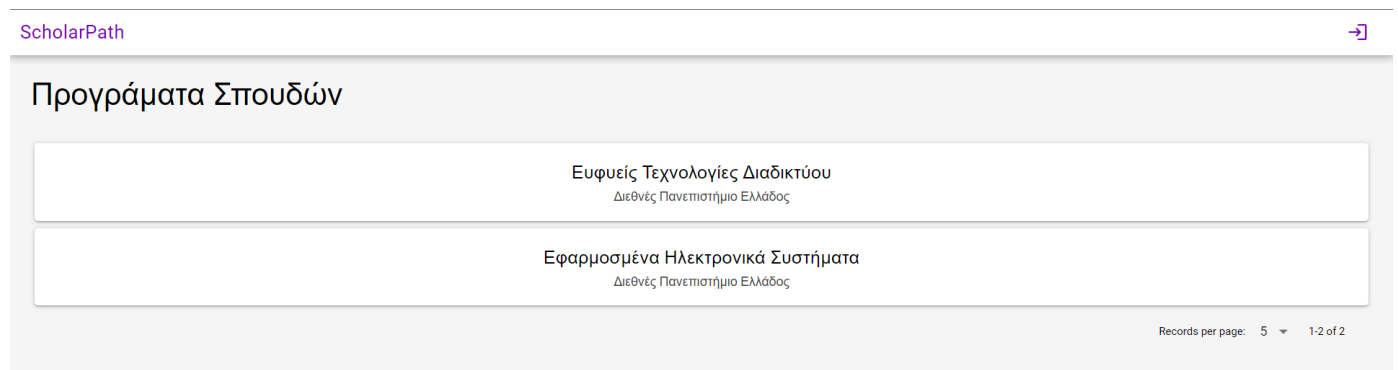
Κεφάλαιο 4

Παρουσίαση του ScholarPath

4.1 Σενάριο χρήσης υπόβολής αίτησης

Για την αίτηση εισαγωγής σε ένα ΠΜΣ, οι υποψήφιοι πρέπει να περιηγηθούν στα διάφορα ΠΜΣ προγράμματα, εντοπίζοντας αυτό που προσφέρει τα κατάλληλα επιστημονικά πεδία για τα ενδιαφέροντά τους. Εάν το επιθυμητό ΠΜΣ είναι ανοιχτό για αιτήσεις, οι υποψήφιοι πρέπει να συμπληρώσουν την κατάλληλη φόρμα αίτησης.

Προκειμένου να αποτυπώσουμε αυτήν τη διαδικασία, παρακάτω παρουσιάζονται εικόνες από την εφαρμογή που αναπτύχθηκε, επιτρέποντας έτσι στους υποψηφίους να κατανοήσουν καλύτερα τη διαδικασία με τη βοήθεια των εικόνων.



Σχήμα 4.1: Σελίδα με όλα τα Προγράμματα Σπουδών

ScholarPath →

Ευφυείς Τεχνολογίες Διαδικτύου

Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, Διεθνές Πανεπιστήμιο Ελλάδος

Οι αιτήσεις είναι ανοιχτές έως 28/09/2023 ¹ 2 [ΑΙΤΗΣΗ](#)

Το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος διοργανώνει από το ακαδημαϊκό έτος 2012-2013 αυτόνομο Πρόγραμμα Μεταπτυχιακών Σπουδών στις Ευφυείς Τεχνολογίες Διαδικτύου (MSc in Web Intelligence). Το πρόγραμμα στοχεύει στην παροχή εκπαίδευσης μεταπτυχιακού επιπέδου και την παραγωγή έρευνας στη διαδικτυακή νοημοσύνη συνδυάζοντας τις προηγμένες τεχνολογίες πληροφοριών στον ιστό και το διαδίκτυο με αυτές των ευφυών συστημάτων. Η διάρκεια σπουδών είναι τρία ακαδημαϊκά εξάμηνα και υπάρχει η δυνατότητα μερικής παρακολούθησης με ελάχιστη διάρκεια ολοκλήρωσης τα 5 εξάμηνα. Τα τέλη φοίτησης για όλη τη διάρκεια των σπουδών στο πρόγραμμα είναι 400 ευρώ. Τα μαθήματα παραδίδονται Τετάρτη, Παρασκευή και Σάββατο. Την Τετάρτη και την Παρασκευή τα μαθήματα ξεκινούν στις 4:00μμ, ενώ το Σάββατο ξεκινούν στις 10:00πμ. Από το Ακαδημαϊκό έτος 2023-24, το 67% των διαλέξεων θα πραγματοποιούνται με μέσα εξ αποστάσεως εκπαίδευσης.

[Visit Website](#)

Σχήμα 4.2: ΠΜΣ με ανοιχτό CFS. Στην εικόνα: 1) Πεδίο στην σελίδα που ενημερώνει μέχρι πότε είναι ανοιχτές οι αιτήσεις 2) κουμπί για να ανοίξει η φόρμα της αίτησης

Αίτηση Υποψηφίου

Επίθετο

Όνομα

Email

Όνομα Πατρός

Ηλικία

Οδός Κατοικίας Ταχυδρομικός Κώδικας

Φύλο

Σχήμα 4.3: Ανοιχτή φόρμα αίτησης

Άνδρας Γυναίκα Άλλο

Νούμερο Τηλεφώνου
6912345678

Προπτυχιακός Τίτλος
Μηχανικός Πληροφορικής

Απαραίτητα Έγγραφα

CV
CV.pdf

Απόδειξη Πτυχίου
Degree.pdf

Ενδιαφέρομαι για Part time

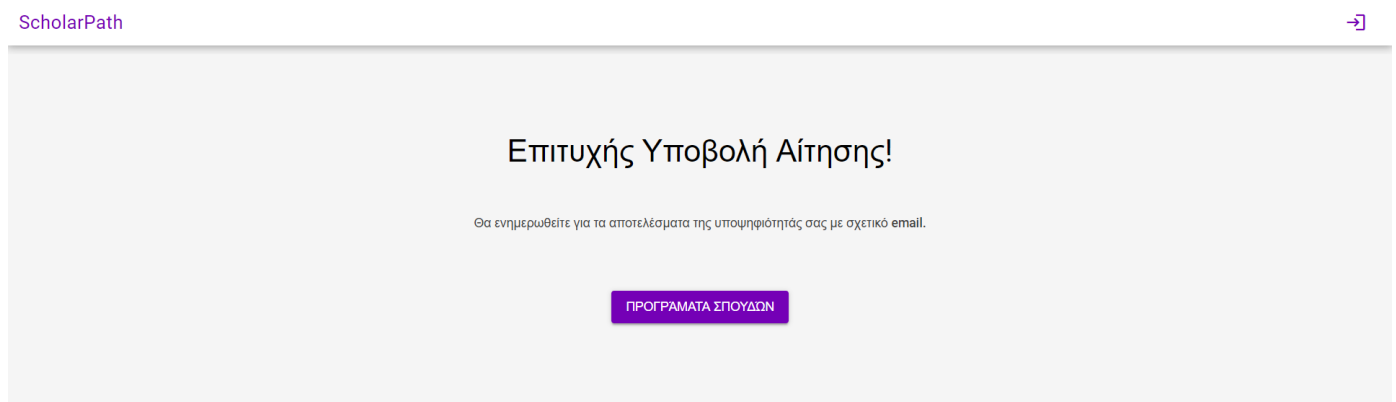
Χρειάζονται 1 Συστατικές
Συμπληρώστε παρακάτω τα στοιχεία όσων θέλετε να σας κάνουν συστατική.

Συστατική #1 Όνομα
Ουγιάρογλου Στέφανος

Συστατική #1 Email
stoug@ihu.gr

SUBMIT ¹ CANCEL

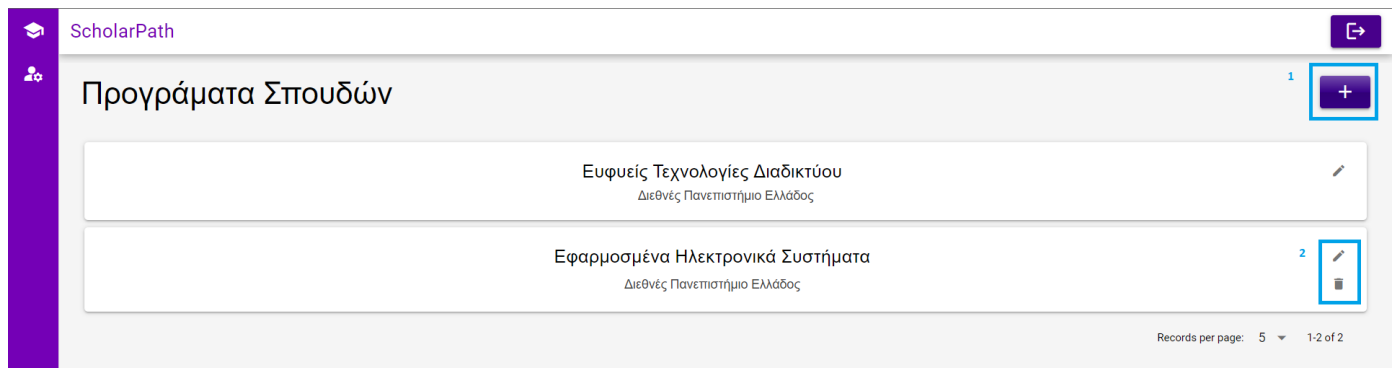
Σχήμα 4.4: Συμπλήρωση και Υποβολή αίτησης. Στην εικόνα: 1) Κουμπί τελικής υποβολής της φόρμας.



Σχήμα 4.5: Σελίδα επιτυχής αίτησης

4.2 Σενάριο χρήσης δημιουργίας Προγράμματος Σπουδών

Για τους διαχειριστές (Admins), η διαδικασία δημιουργίας ενός νέου ΠΜΣ πραγματοποιείται μέσω της σελίδας που περιλαμβάνει τα προγράμματα σπουδών. Κατά τη διάρκεια αυτής της διαδικασίας, οι διαχειριστές συμπληρώνουν τα απαραίτητα πεδία πληροφοριών για το νέο ΠΜΣ, συμπεριλαμβανομένων των πεδίων αξιολόγησης. Αφότου ολοκληρωθεί η διαδικασία αυτή, το νέο ΠΜΣ προστίθεται ως μια νέα εγγραφή στη σελίδα που περιλαμβάνει τα υπόλοιπα ΠΜΣ. Με αυτόν τον τρόπο, οι διαχειριστές διαχειρίζονται τα διάφορα προγράμματα σπουδών και ενημερώνουν την πλατφόρμα με τις απαραίτητες πληροφορίες.



Σχήμα 4.6: Σελίδα ΠΜΣ σε συνδεδεμένο Admin. Στην εικόνα 1) Κουμπί για δημιουργία νέου ΠΜΣ 2) Κουμπιά για την Επεξεργασία και την Διαγραφή υπάρχον ΠΜΣ

Πρόγραμμα Σπουδών

Τίτλος

Slug
Πρέπει να είναι μοναδικό για κάθε Πρόγραμμα Σπουδών

Πανεπιστήμιο

Τμήμα

Περιγραφή
Σύντομη περιγραφή του Προγράμματος Σπουδών

Link του site

Πεδία Αξιολόγησης

Σχήμα 4.7: Ανοιχτή φόρμα δημιουργίας ΠΜΣ.

Πρόγραμμα Μεταπτυχιακών Σπουδών (Π.Μ.Σ.) στην Εφαρμοσμένη Πληροφορική.

Σύντομη περιγραφή του Προγράμματος Σπουδών

Link του site
http://informatics.teicm.gr/msc_informatics/

Πεδία Αξιολόγησης 1

Τίτλος

Πρέπει να είναι ξεχωριστός σε κάθε Πρόγραμμα Σπουδών (case insensitive).

Min Τιμή: 0 Max Τιμή: 5 +

Προπτυχιακό Συνάφεια	Values: 0 - 5	3 🗑️
Βαθμός Προπτυχιακού	Values: 0 - 10	🗑️
Εργασιακή Εμπειρία	Values: 0 - 5	🗑️
Συνέντευξη	Values: 0 - 20	🗑️

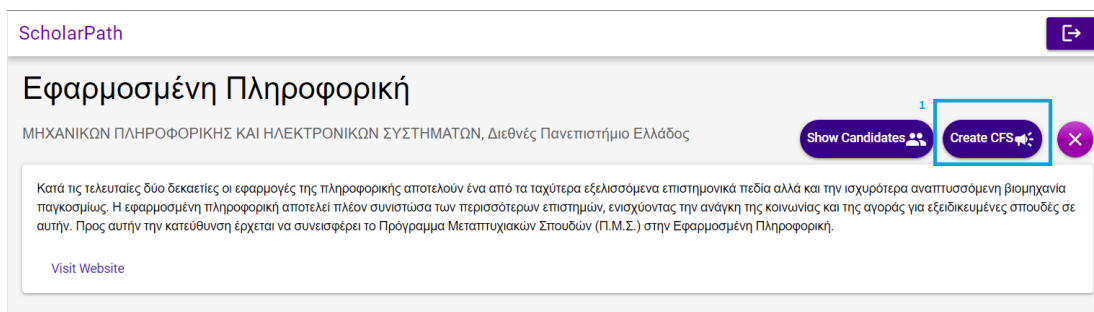
CANCEL ⁴
SAVE

Σχήμα 4.8: Συμπλήρωση και Υποβολή Φόρμας νέου ΠΜΣ. Στην εικόνα: 1) Πεδία για νέα κριτήρια αξιολόγησης 2) Συμπληρωμένα κριτήρια αξιολόγησης 3) Αφαίρεση συγκεκριμένου κριτηρίου αξιολόγησης 4) Κουμπί υποβολής αίτησης

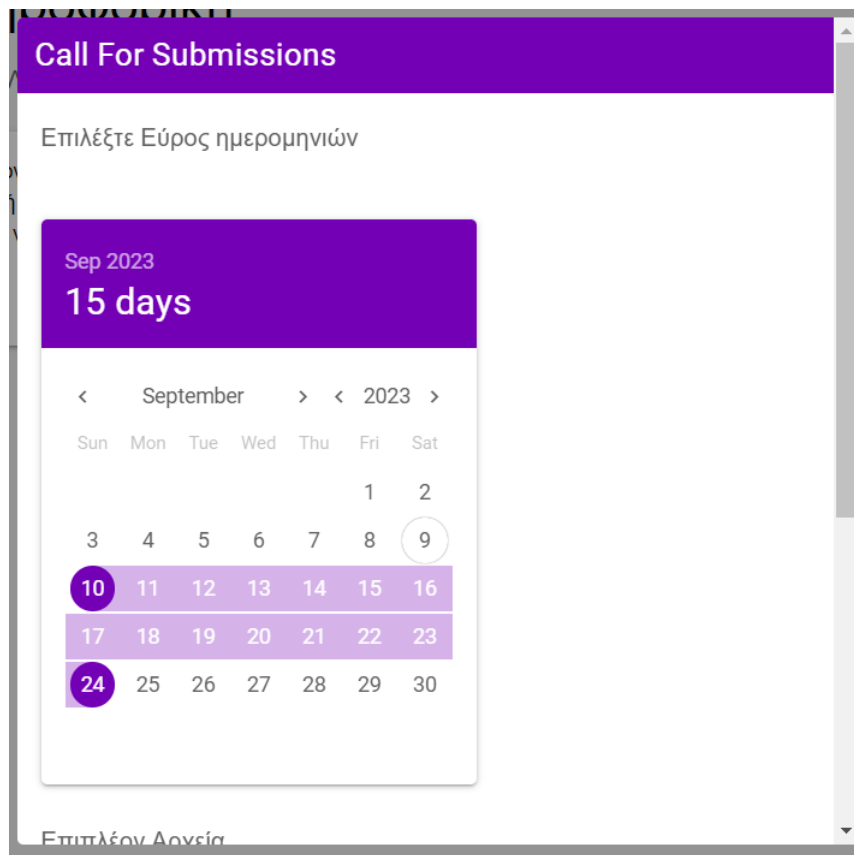
4.3 Σενάριο χρήσης για δημιουργία CFS

Οι διαχειριστές (Admins) έχουν τη δυνατότητα να ανοίξουν μια περίοδο υποβολής αιτήσεων για ένα συγκεκριμένο ΠΜΣ, εάν δεν υπάρχει ήδη κάποια άλλη ενεργή ή προγραμματισμένη περίοδο υποβολής αιτήσεων για το ίδιο ΠΜΣ. Αυτό μπορεί να πραγματοποιηθεί μέσω της σελίδας που αντιστοιχεί στο εν λόγω ΠΜΣ.

Όταν η περίοδος υποβολής αιτήσεων βρίσκεται σε κατάσταση "published" (έχει δημοσιευτεί αλλά δεν έχει ανοίξει ακόμη), οι διαχειριστές μπορούν να την επεξεργαστούν, προσθέτοντας ή τροποποιώντας τις λεπτομέρειες της περιόδου υποβολής αιτήσεων. Ωστόσο, αφού η περίοδος υποβολής ανοίξει (έχει καθοριστεί η ημερομηνία έναρξης), οι διαχειριστές έχουν τη δυνατότητα μόνο να επεκτείνουν τη διάρκειά της, αλλά δεν μπορούν πλέον να την τροποποιήσουν.

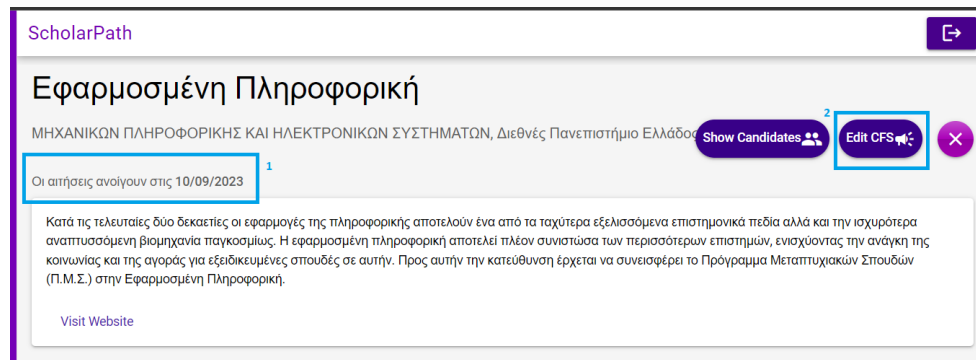


Σχήμα 4.9: Κουμπί για να ανοίξει η φόρμα δημιουργίας CFS

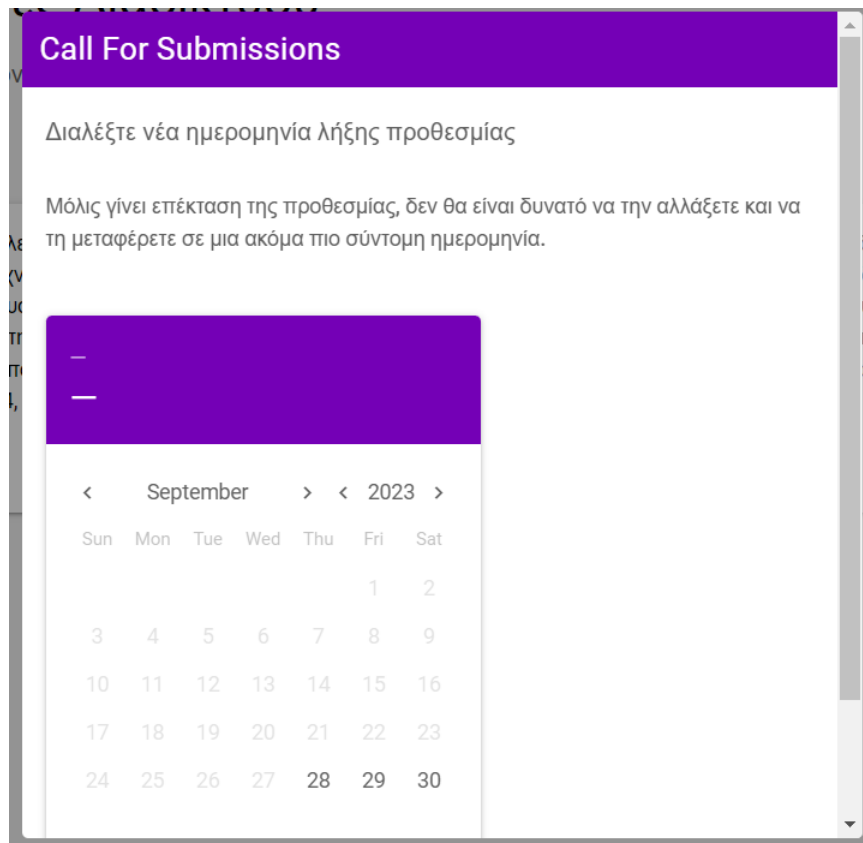


Σχήμα 4.10: Ανοιχτή φόρμα CFS

Σχήμα 4.11: Συμπλήρωση και Υποβολή φόρμας.



Σχήμα 4.12: Η σελίδα του ΠΜΣ μετά την επιτυχή δημοσίευση CFS. Στην εικόνα: 1) Πεδίο στην σελίδα που ενημερώνει για την κατάσταση του CFS 2) Κουμπί για Επεξεργασία του published CFS

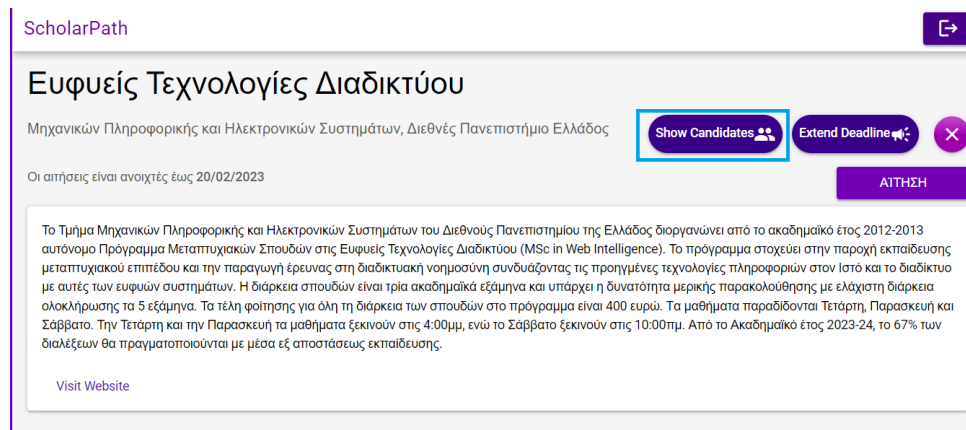


Σχήμα 4.13: Φόρμα επέκτασης διορίας σε ανοιχτό CFS

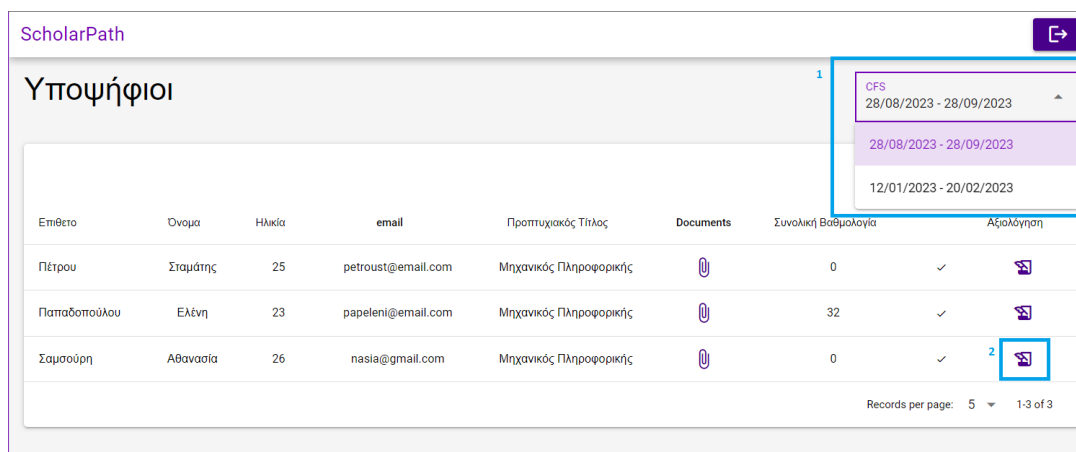
4.4 Σενάριο χρήσης για αξιολόγηση

Μέσω της σελίδας του συγκεκριμένου ΠΜΣ, οι συνδεδεμένοι χρήστες με κατάλληλους ρόλους έχουν τη δυνατότητα να μεταφερθούν στη σελίδα που περιλαμβάνει τους υποψηφίους (candidates) που έχουν υποβάλει αιτήσεις για το εν λόγω ΠΜΣ. Εκεί, με βάση το CFS, μπορούν να προβάλουν λεπτομέρειες σχετικά με τις αιτήσεις που έχουν υποβληθεί και να προχωρήσουν στην αξιολόγησή τους.

Οι συνδεδεμένοι χρήστες, ανάλογα με τους δικούς τους ρόλους και δικαιώματα, μπορούν να προβούν στην αξιολόγηση των υποβληθέντων αιτήσεων, βασιζόμενοι στα δεδομένα που παρέχονται μέσω του CFS. Αυτή η διαδικασία επιτρέπει στους υπεύθυνους να εξετάσουν προσεκτικά τις λεπτομέρειες των αιτήσεων και να καταχωρήσουν την αξιολόγησή τους, βοηθώντας έτσι στην διαδικασία επιλογής των υποψηφίων που θα εγγραφούν στο ΠΜΣ.



Σχήμα 4.14: Κουμπί για μεταφορά στην σελίδα με τους Candidates ενός ΠΜΣ



Σχήμα 4.15: Σελίδα με τους Υποψήφιους ενός ΠΜΣ. Στην εικόνα: 1) Επιλογή CFS για την εμφάνιση των αντίστοιχων αιτήσεων 2) Κουμπί αξιολόγησης συγκεκριμένης αίτησης

Αξιολόγηση Υποψηφίου

Προπτυχικό
5

Συνέντευξη
18

Έρευνα
1

Σχετική Εμπειρία
6

Το πεδίο πρέπει να είναι μικρότερο ή ίσο από 5

CANCEL SAVE

Σχήμα 4.16: Φόρμα αξιολόγησης. Στην εικόνα: 1) Επιβεβαίωση σωστών τιμών βάσει των κριτηρίων που έχουν δοθεί στο ΠΜΣ 2) Κουμπί για υποβολή της αξιολόγησης

ScholarPath

Candidate Review successfully Created

Υποψήφιοι

CFS
28/08/2023 - 28/09/2023

Search

Επώνυμο	Όνομα	Ηλικία	email	Προπτυχιακός Τίτλος	Documents	Συνολική Βαθμολογία	Αξιολόγηση
Πέτρου	Σταμάτης	25	petroust@email.com	Μηχανικός Πληροφορικής	📎	0	✓
Παπαδοπούλου	Ελένη	23	papeleni@email.com	Μηχανικός Πληροφορικής	📎	32	✓
Σαμασούρη	Αθανασία	26	nasia@gmail.com	Μηχανικός Πληροφορικής	📎	29	✓

Records per page: 5 1-3 of 3

Σχήμα 4.17: Επιτυχής Αξιολόγηση. Στην εικόνα: 1) Ειδοποίηση για την επιτυχή αξιολόγηση 2) Αυτόματη ανανέωση της τελικής βαθμολογίας

4.5 Σενάριο αποδοχής Υποψηφίων

Αφού περάσει η διορία του CFS και ολοκληρωθούν οι αξιολογήσεις, οι διαχειριστές (admins) μπορούν να προβούν στην αποδοχή των υποψηφίων. Αυτό γίνεται μέσω μιας αυτοματοποιημένης διαδικασίας, κατά την οποία οι διαχειριστές καθορίζουν τον αριθμό των θέσεων που επιθυμούν να προσλάβουν. Στη συνέχεια, παρέχουν επιπλέον πληροφορίες για τη διαδικασία, όπως την ημερομηνία καταληκτικής προθεσμίας για την υποβολή απαιτούμενων εγγράφων από τους επιλεγμένους υποψηφίους, ή πληροφορίες για τα άτομα που βρίσκονται σε λίστα αναμονής και τότε αναμένεται να λάβουν απόφαση σχετικά με την επιλογή τους.

Η εφαρμογή αξιοποιεί αυτόματα τις πιο υψηλά βαθμολογημένες αιτήσεις, λαμβάνοντας υπόψη τον αριθμό των θέσεων που έχουν ορίσει οι διαχειριστές, και αποστέλλει τα αντίστοιχα emails προς τους επιλεγμένους υποψηφίους.

The screenshot shows the ScholarPath Admin interface. The main heading is 'Υποψήφιοι'. There is a search bar and a 'CFS' status indicator showing '28/08/2023 - 08/09/2023'. A button labeled 'ΑΠΟΔΟΧΗ ΑΙΤΗΣΕΩΝ' is visible. The table below lists three candidates:

Επίθετο	Όνομα	Ηλικία	email	Προπτυχιακός Τίτλος	Documents	Συνολική Βαθμολογία	Αξιολόγηση
Πέτρου	Σταμάτης	25	petroust@email.com	Μηχανικός Πληροφορικής		0	✓
Παπαδοπούλου	Ελένη	23	papeleni@email.com	Μηχανικός Πληροφορικής		32	✓
Σαμσούρη	Αθανασία	26	nasia@gmail.com	Μηχανικός Πληροφορικής		29	✓

Records per page: 5 1-3 of 3

Σχήμα 4.18: Σελίδα Υποψηφίων με κλειστό CFS για τον Admin

Αποδοχή Αιτήσεων

Με αυτή την ενέργεια, όλοι οι υποψήφιοι θα παραλάβουν email είτε αποδοχής στο ΠΜΣ είτε ότι μπήκαν στην λίστα αναμονής. Η ενέργεια δεν είναι αναστρέψιμη και μπορεί να διαρκέσει λίγα λεπτά.

Θετικές Αιτήσεις

Το νούμερο των αιτήσεων που θα πάρουν θετική απάντηση (κατανεμημένα βάσει τελικής βαθμολογίας)

Διορία Αποδοχής Εισαγωγής

Η ημερομηνία έως την οποία μπορούν οι υποψήφιοι να στείλουν τα χαρτιά τους

< September > < 2023 >

Sun Mon Tue Wed Thu Fri Sat

Σχήμα 4.19: Ανοιχτή φόρμα Αποδοχής Υποψηφίων (1)

Διορία Αποδοχής Εισαγωγής
Η ημερομηνία έως την οποία μπορούν οι υποψήφιοι να στείλουν τα χαρτιά τους

—
—

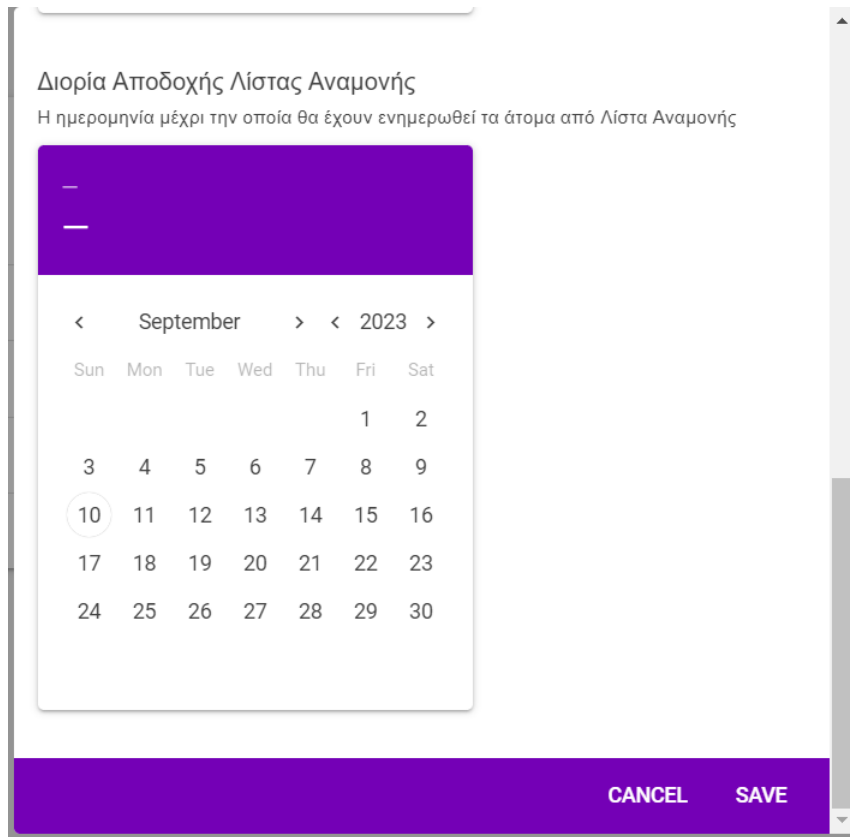
< September > < 2023 >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

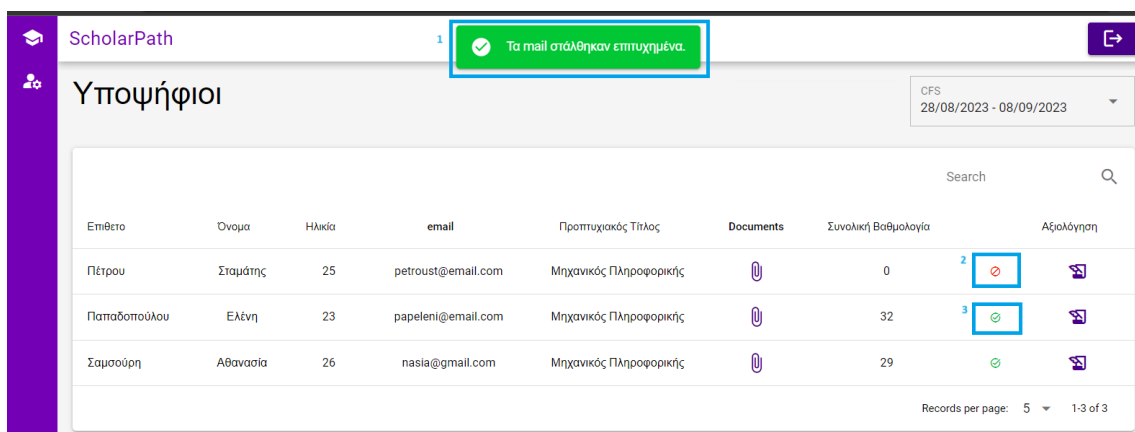
Διορία Αποδοχής Λίστας Αναμονής
Η ημερομηνία μέχρι την οποία θα έχουν ενημερωθεί τα άτομα από Λίστα Αναμονής

—

Σχήμα 4.20: Ανοιχτή φόρμα Αποδοχής Υποψηφίων (2)



Σχήμα 4.21: Ανοιχτή φόρμα Αποδοχής Υποψηφίων (3)



Σχήμα 4.22: Σελίδα Υποψηφίων με επιτυχή αποδοχή υποψηφίων. Στην εικόνα: 1) Ειδοποίηση για ενημέρωση επιτυχούς αποστολής emails 2) Αίτηση σε κατάσταση Λίστας Αναμονής 3) Αίτηση σε κατάσταση Αποδοχής

4.6 Σενάριο χρήσης σύνταξης και αποστολής Συστατικής Επιστολής

Οι καθηγητές που έχουν κληθεί από υποψηφίους για τη συμπλήρωση μιας συστατικής επιστολής λαμβάνουν ένα ηλεκτρονικό μήνυμα (email) που περιλαμβάνει όλες τις απαραίτητες πληροφορίες. Συγκεκριμένα, το email περιλαμβάνει ένα μοναδικό URL που οδηγεί τους καθηγητές στη διαδικτυακή φόρμα όπου μπορούν να συμπληρώσουν την αντίστοιχη συστατική επιστολή.

Αυτή η διαδικασία επιτρέπει στους καθηγητές να αξιολογήσουν τη συγκεκριμένη αίτηση και να παράσχουν τη συστατική επιστολή μέσω της διαδικτυακής φόρμας, καθιστώντας τη διαδικασία αποδοτική και ευκολότερη για όλα τα εμπλεκόμενα μέρη.

ScholarPath →

Συστατική Επιστολή

Όνομα Υποψηφίου: Πέτρου Σταμάτης

Όνοματεπώνυμο

Παπαδοπούλου Μαρία

Email
pmaria@ihu.gr

Παρακαλούμε συμπληρώστε την συστατική επιστολή.

Σε σχέση με το πρόγραμμα μεταπτυχιακού σπουδών με τίτλο "Ευφυείς Τεχνολογίες Διαδικτύου," προτείνω τον φοιτητή Πέτρου Σταμάτης ως εξαιρετική επιλογή για αυτό το πρόγραμμα. Ο Πέτρος διακρίνεται σε πολλά επίπεδα και έχει επιδείξει εξαιρετική δέσμευση στον τομέα των ευφυών τεχνολογιών διαδικτύου.

Έχοντας συνεργαστεί με τον Πέτρο κατά τη διάρκεια των προπτυχιακών του σπουδών, μπορώ να επιβεβαιώσω ότι είναι ένα ιδιαίτερα αφοσιωμένο και εξυμνο άτομο. Οι ικανότητές του στον τομέα της τεχνολογίας είναι εντυπωσιακές, και έχει δείξει συνεχή ενδιαφέρον και προοδευτική σκέψη σχετικά με τις εξελίξεις στον τομέα των ευφυών τεχνολογιών.

Επιπλέον, ο Πέτρος είναι εξαιρετικός στην ανάλυση προβλημάτων και την επίλυσή τους. Έχει επίσης επιδείξει δεξιότητες στην ομαδική εργασία και την επικοινωνία, καθώς και την ικανότητα να εκφράζει τις ιδέες του σε γραπτή και προφορική μορφή με σαφήνεια και ακρίβεια.

Πιστεύω ακράδαντα ότι ο Πέτρος Σταμάτης είναι ο κατάλληλος φοιτητής για αυτό το πρόγραμμα μεταπτυχιακών σπουδών και ότι θα συνεισφέρει σημαντικά στον τομέα των ευφυών τεχνολογιών διαδικτύου. Συνιστώ τον Πέτρο χωρίς επιφυλάξεις για τη συμμετοχή του σε αυτό το πρόγραμμα.

Σχήμα 4.23: Σελίδα για Σύνταξη Συστατικής Επιστολής.

Επιτυχής Υποβολή Συστατικής Επιστολής!

Ευχαριστούμε για τον χρόνο που αφιερώσατε.

ΠΡΟΓΡΑΜΑΤΑ ΣΠΟΥΔΩΝ

Σχήμα 4.24: Σελίδα επιτυχημένης υποβολής Συστατικής Επιστολής.

Η συστατική έχει συμπληρωθεί ήδη.

Ευχαριστούμε για τον χρόνο που αφιερώσατε.

ΠΡΟΓΡΑΜΑΤΑ ΣΠΟΥΔΩΝ

Σχήμα 4.25: Σελίδα υποβλημένης συστατικής επιστολής.

Δεν υπάρχει ενεργή Συστατική Επιστολή.

Ο σύνδεσμος που χρησιμοποιήσατε είναι λάθος ή έχει λήξει.

ΠΡΟΓΡΑΜΑΤΑ ΣΠΟΥΔΩΝ

Σχήμα 4.26: Σελίδα για ληγμένης ή άκυρης Συστατικής Επιστολής.

4.7 Σενάριο χρήσης διαχείρισης χρηστών

Οι διαχειριστές (Admins) έχουν τη δυνατότητα να διαχειρίζονται τις προσκλήσεις προς νέους χρήστες, καθώς και τις προσκλήσεις που έχουν ήδη αποσταλεί. Αφού οι χρήστες ολοκληρώσουν την εγγραφή τους στο ScholarPath, οι διαχειριστές μπορούν να τους αναθέσουν ή να αφαιρέσουν ρόλους σε διάφορα ΠΜΣ που διαχειρίζονται. Αυτό σημαίνει ότι μπορούν να ορίσουν τα δικαιώματα και τις ευθύνες των χρηστών σε συγκεκριμένα ΠΜΣ.

Επιπλέον, οι διαχειριστές έχουν τη δυνατότητα να αναθέσουν δικαιώματα διαχειριστή (administrator) σε συγκεκριμένους χρήστες. Αυτό σημαίνει ότι αυτοί οι χρήστες θα έχουν πρόσβαση και δικαιώματα διαχείρισης του συστήματος ScholarPath, και μπορούν να διαχειρίζονται τα ΠΜΣ και τις προσκλήσεις σε άλλους χρήστες.

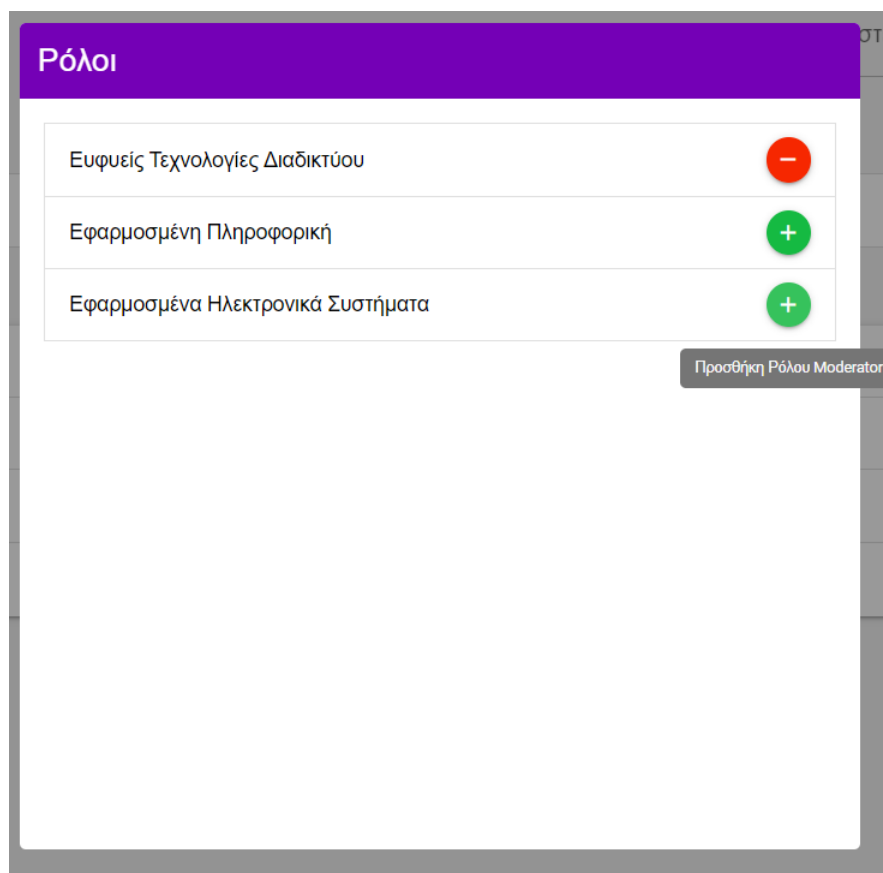
Αυτές οι δυνατότητες επιτρέπουν στους διαχειριστές να διαχειρίζονται αποτελεσματικά την χρήση του ScholarPath και να αναθέτουν δικαιώματα και ρόλους στους χρήστες, ανάλογα με τις ανάγκες του συστήματος.

The screenshot displays the ScholarPath user management interface. At the top, there is a search bar labeled "Προσκαλέστε Χρήστη (email)" with a "ΠΡΟΣΚΛΗΣΗ" button (1). Below this is a section for "Προσκλήσεις" showing a list of invited users, with "test@test.com" visible (2). The main part of the interface is a table titled "Προσκεκλημένοι Χρήστες" (3) with the following columns: Name, email, Ρόλος, and roles. The table contains two rows:

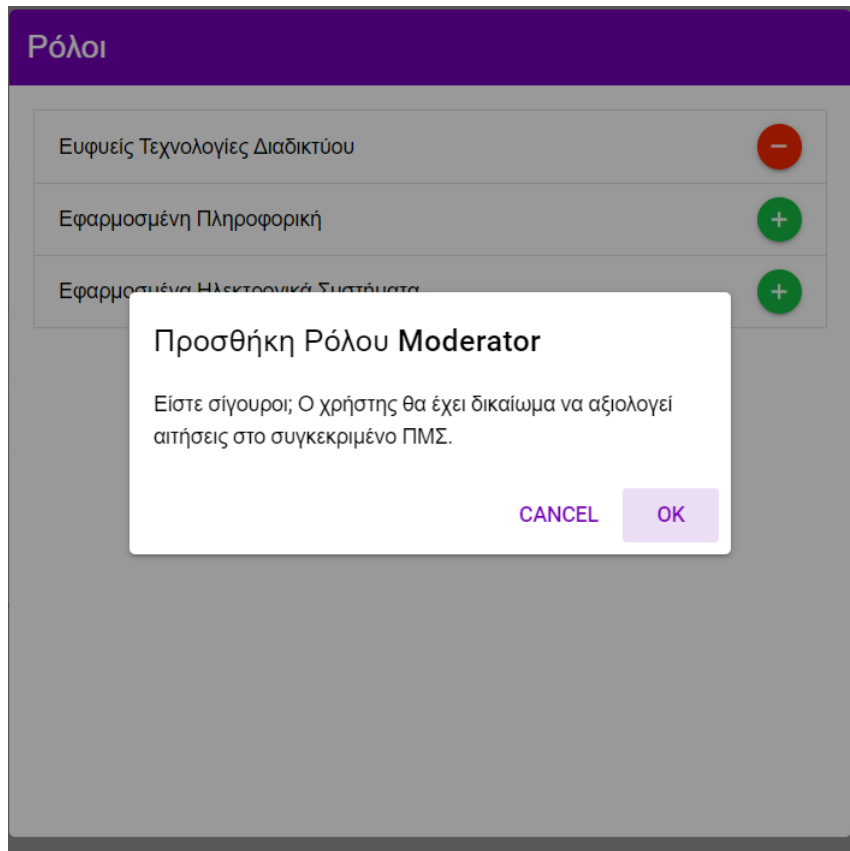
Name	email	Ρόλος	roles
Ζωή Αλεκίδου	nasia@test.com		0
Athanasia Samsouri	nasia@ihu.gr		1

The role icon for Athanasia Samsouri is highlighted with a blue box (4), and the number of roles assigned to her is shown as 1 (5). At the bottom right, there is a "Records per page" dropdown set to 5 and a "1-2 of 2" indicator.

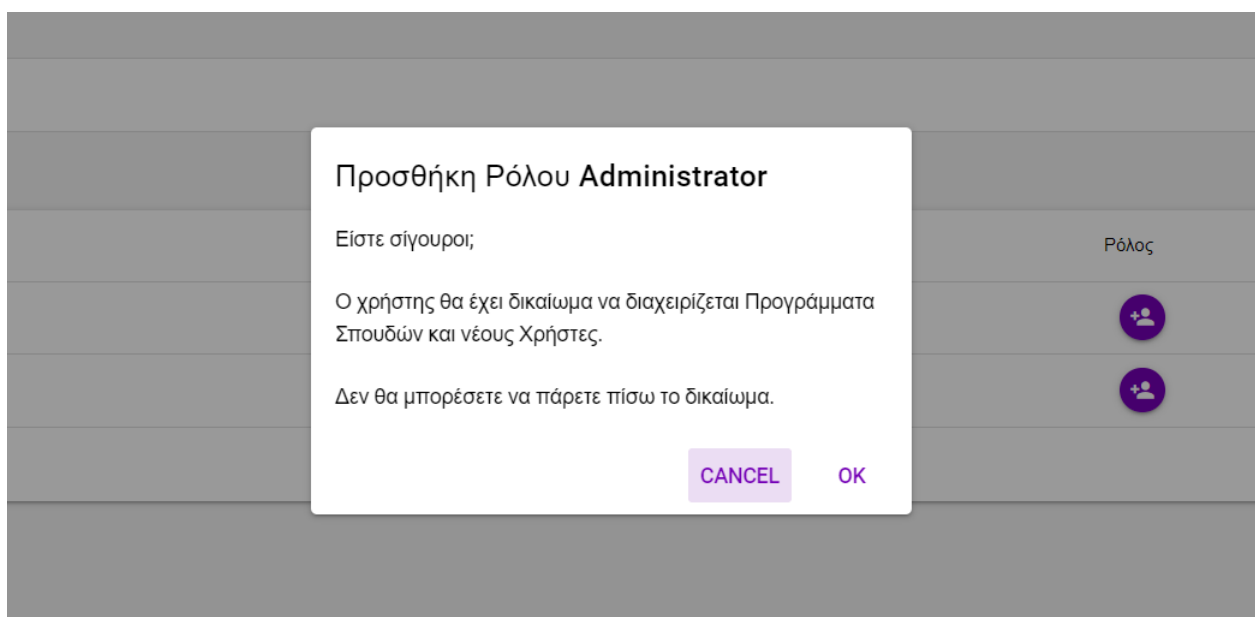
Σχήμα 4.27: Σελίδα με τους προσκεκλημένους users. Στην εικόνα: 1) Εισαγωγή email για αποστολή πρόσκλησης στο ScholarPath 2) Emails που έχει σταλεί πρόσκληση αλλά δεν έχουν προχωρήσει σε εγγραφή 3) Χρήστες που έκαναν εγγραφή με πρόσκληση 4) Ανάθεση Admin ρόλου 5) Πόσους ρόλους έχει ο χρήστης / ανάθεση ρόλου moderator



Σχήμα 4.28: Ανάθεση Ρόλου Moderator σε ΠΜΣ



Σχήμα 4.29: Επιβεβαίωση ανάθεσης ρόλου Moderator.



Σχήμα 4.30: Επιβεβαίωση ανάθεσης ρόλου Διαχειριστή.

Κεφάλαιο 5

Συμπεράσματα και Μελλοντικές επεκτάσεις

5.1 Συμπεράσματα

Οι διαδικασίες αιτήσεων για μεταπτυχιακά προγράμματα σπουδών στην Ελλάδα, τόσο για τους υποψήφιους σπουδαστές όσο και για τους καθηγητές, μπορεί να αποτελούν πηγή άγχους και ανησυχίας. Η απουσία μιας κοινής πλατφόρμας και το χάος στη διαχείριση των προγραμμάτων σπουδών μπορεί να δημιουργήσει προβλήματα και να καθυστερήσει τη διαδικασία.

Η ανάπτυξη του ScholarPath προσφέρει μια καινοτόμα λύση για το πρόβλημα αυτό. Πρόκειται για μια πλατφόρμα που επιτρέπει στα πανεπιστήμια να διαχειρίζονται αποτελεσματικά τα μεταπτυχιακά προγράμματα σπουδών τους. Μέσω του ScholarPath, οι καθηγητές μπορούν να αξιολογούν αιτήσεις και να επιλέγουν τους καλύτερους υποψήφιους με μεγαλύτερη ευκολία και ακρίβεια. Ταυτόχρονα, οι υποψήφιοι φοιτητές έχουν μια διαφανή και αποδοτική πλατφόρμα για την υποβολή των αιτήσεών τους.

Το ScholarPath αυτοματοποιεί τις λειτουργίες και εξαλείφει τον πολυάριθμο χειροκίνητο έλεγχο, προσφέροντας ασφάλεια και αξιοπιστία σε όλη τη διαδικασία. Με αυτόν τον τρόπο, συμβάλλει στην αποτελεσματικότερη και διαφανή διαχείριση των μεταπτυχιακών προγραμμάτων σπουδών.

Για τους υποψήφιους φοιτητές, το ScholarPath προσφέρει μια ενοποιημένη εμπειρία κατά την αναζήτηση και υποβολή αιτήσεων για μεταπτυχιακά προγράμματα σπουδών. Μπορούν να εξερευνήσουν διάφορα προγράμματα σε μία μόνο πλατφόρμα, εξοικονομώντας χρόνο και κόπο. Το ScholarPath διευκολύνει την υποβολή αιτήσεων, καθώς διαχειρίζεται όλα τα απαραίτητα έγγραφα και την αλληλοκοινοποίηση με το πανεπιστήμιο.

Οι υποψήφιοι δεν χρειάζεται να ανησυχούν για την αποστολή email ή την υποβολή επιστολών συστάσεων ξεχωριστά, καθώς όλα αυτά τα βήματα διεκπεραιώνονται εύκολα και αυτόματα

μέσω της πλατφόρμας. Αυτό διευκολύνει τους υποψηφίους και τους επιτρέπει να επικεντρωθούν στην επιλογή των προγραμμάτων που τους ενδιαφέρουν, χωρίς να αντιμετωπίζουν τη γραφειοκρατία και τις περίπλοκες διαδικασίες υποβολής αιτήσεων.

Η χρήση του ScholarPath διασφαλίζει τη διαφάνεια σε όλη τη διαδικασία αναφορικά με τις συστατικές επιστολές και την αξιολόγηση των αιτήσεων. Οι συστατικές επιστολές διαχειρίζονται πλήρως μέσα από την εφαρμογή, εξαλείφοντας την ανάγκη για χειρωνακτική αποστολή ή παρακολούθηση μέσω email.

Επιπλέον, η αξιολόγηση των αιτήσεων γίνεται με ακρίβεια και συμμορφώνεται με τους κανόνες που έχουν οριστεί. Αυτό εξασφαλίζει ότι οι αιτήσεις επεξεργάζονται χωρίς λάθη και χωρίς παρεμβάσεις ανθρώπινου παράγοντα που μπορεί να οδηγήσουν σε ανικανοποίητες αποφάσεις. Η αυτοματοποίηση αυτής της διαδικασίας εξαλείφει πιθανά λάθη και επιτρέπει στους καθηγητές να αξιολογούν τις αιτήσεις αποτελεσματικά και αξιόπιστα.

5.2 Μελλοντικές Επεκτάσεις

Η χρήση του ScholarPath ως εφαρμογή ανοιχτού λογισμικού είναι μια εξαιρετική προσέγγιση που επιτρέπει την επεκτασιμότητα και τη συμμετοχή της κοινότητας. Οι πιθανές επεκτάσεις, όπως η ενσωμάτωση των εγγραφών των φοιτητών μετά την αποδοχή τους στο ΠΜΣ, αποτελούν σημαντικές βελτιώσεις.

Η προσθήκη ενός νέου ρόλου, όπως η γραμματεία, μπορεί να αποδειχθεί εξαιρετικά χρήσιμη. Αυτό θα επιτρέψει την πλήρη διαχείριση των ΠΜΣ μέσω του ScholarPath, από την αρχή μέχρι το τέλος. Οι φοιτητές δεν θα χρειάζεται να έρχονται σε άμεση επαφή με τη γραμματεία του τμήματος, καθώς όλες οι απαραίτητες ενέργειες και εγγραφές θα διαχειρίζονται μέσω του ScholarPath. Αυτό θα βελτιώσει την αποτελεσματικότητα της διαδικασίας και θα διασφαλίσει τη συνεχή επικοινωνία και διαχείριση των δεδομένων μεταξύ των φοιτητών και της γραμματείας, με όλες τις πληροφορίες που είναι διαθέσιμες μέσω της πλατφόρμας.

Μια επιπλέον προσθήκη που μπορεί να αποδειχθεί επωφελής για το ScholarPath είναι η ενσωμάτωση της πολυγλωσσίας στον client. Μέσω της χρήσης του i18n module για τις μεταφράσεις, το ScholarPath μπορεί να καταστεί ακόμη πιο προσβάσιμο σε φοιτητές εκτός Ελλάδας.

Αυτό επιτρέπει στους χρήστες που ομιλούν διάφορες γλώσσες να αλληλεπιδρούν με την εφαρμογή χρησιμοποιώντας τη γλώσσα της προτίμησής τους. Με τη δυνατότητα μετάφρασης του περιεχομένου, οι φοιτητές εκτός Ελλάδας μπορούν να απολαμβάνουν την ίδια επικοινωνία και ευκολία πρόσβασης στις υπηρεσίες που προσφέρει το ScholarPath, ενισχύοντας τη διεθνή διασύνδεση και συνεργασία στον τομέα των μεταπτυχιακών σπουδών.

Συνοψίζοντας, το ScholarPath αποτελεί ένα σημαντικό εργαλείο για τη διαχείριση και την επίλυση προβλημάτων που αντιμετωπίζουν τα πανεπιστήμια και οι υποψήφιοι φοιτητές στη διαδικασία των μεταπτυχιακών προγραμμάτων σπουδών στην Ελλάδα. Με την αυτοματοποίηση

των λειτουργιών και τη διασφάλιση της διαφάνειας, προσφέρει ευκολία και αξιοπιστία σε μια διαδικασία που παρουσιάζει πολλές προκλήσεις.

Από τη διαχείριση των αιτήσεων μέχρι την αξιολόγηση των υποψηφίων, το ScholarPath βοηθά στη βελτίωση της αποτελεσματικότητας και της ποιότητας της διαδικασίας επιλογής και εισαγωγής σε μεταπτυχιακά προγράμματα σπουδών.

Οι προοπτικές επέκτασης του ScholarPath είναι πολλές και ανοιχτές για συνεισφορές από την κοινότητα. Με αυτόν τον τρόπο, μπορεί να συνεχίσει να εξελίσσεται και να προσφέρει ακόμη περισσότερα οφέλη στους χρήστες του, συμβάλλοντας στην αναβάθμιση του εκπαιδευτικού τοπίου της Ελλάδας και πέρα από αυτά των συνόρων της.

Βιβλιογραφία

- [1] D. Flanagan, *JavaScript: The Definitive Guide*. O'Reilly Media, 2011.
- [2] TypeScript Team, "Typescript handbook." Online, Ημερομηνία πρόσβασης: 20 Αυγούστου 2023.
- [3] M. Casciaro, *Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques*. Packt Publishing, 3rd ed., 2021.
- [4] npm, Inc., "npm documentation." <https://docs.npmjs.com/about-npm>, Ημερομηνία πρόσβασης: 20 Αυγούστου 2023.
- [5] M. Cantelon, M. Harter, T. Holowaychuk, and N. Rajlich, *Node.js in Action*. Manning Publications, 2nd ed., 2017.
- [6] "Node.js documentation." Online, Ημερομηνία πρόσβασης: 20 Αυγούστου 2023.
- [7] "Koa.js official documentation." Online, Ημερομηνία πρόσβασης: 20 Αυγούστου 2023.
- [8] E. Porcello, *Learning GraphQL*. O'Reilly Media, 2018.
- [9] PayPal Tech, "GraphQL resolvers best practices." Online, Ημερομηνία πρόσβασης: 20 Αυγούστου 2023.
- [10] E. Hanchett and B. Listwon, *Vue.js in Action First Edition*. Manning Publications, first ed., 2018.
- [11] Vue.js, "Vue.js documentation." Online, Ημερομηνία πρόσβασης: 20 Αυγούστου 2023.
- [12] Quasar Framework, "Quasar framework documentation." Online, Ημερομηνία πρόσβασης: 20 Αυγούστου 2023.
- [13] R. J. Dyer, *Learning MySQL and MariaDB*. O'Reilly Media, Inc., March 2015.
- [14] TechCrunch, "Sun picks up mysql for \$1 billion: Open source is a legitimate business model?," January 2008.
- [15] V. M. Grippa and S. Kuzmichev, *Learning MySQL, 2nd Edition*. O'Reilly Media, Inc., September 2021.
- [16] D. Feury, "What are orms and when to use them?," Ημερομηνία πρόσβασης: 20 Αυγούστου 2023.
- [17] MikroORM, "Mikroorm documentation." Online, Ημερομηνία πρόσβασης: 20 Αυγούστου 2023.