

Ατομική Διπλωματική Εργασία

**ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΟΥ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ
ΔΙΑΔΙΚΑΣΙΩΝ ΤΟΥ ΠΡΟΣΩΠΙΚΟΥ ΤΟΥ ΤΜΗΜΑΤΟΣ
ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΟΥ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΚΥΠΡΟΥ**

Κωνσταντίνος Κουλουμπής

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2025

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Ανάπτυξη Διαδικτυακού Συστήματος Διαχείρισης Διαδικασιών του Προσωπικού του
Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου**

Κωνσταντίνος Κουλουμπής

Επιβλέπων Καθηγητής
Χρύσης Γεωργίου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων
απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου
Κύπρου

Μάιος 2025

Ευχαριστίες

Θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες στον επιβλέποντα καθηγητή μου Καθ. Χρύση Γεωργίου για την πολύτιμη καθοδήγηση, τη συνεχή υποστήριξη και τις χρήσιμες συμβουλές του καθ' όλη τη διάρκεια της Ατομικής Διπλωματικής μου Εργασίας. Επίσης, ευχαριστώ θερμά την Αναπληρώτρια Καθηγήτρια Γεωργία (Ζέτα) Καπιτσάκη και τη γραμματέα του Τμήματος Πληροφορικής κα. Σαββούλα Ευσταθίου για τη συνεργασία, την άμεση ανταπόκριση και τη βοήθεια που μου παρείχαν σε κάθε στάδιο της διαδικασίας. Θα ήθελα, επιπλέον, να ευχαριστήσω την κα. Άντρη Μιχαηλίδου, Λειτουργό Μηχανογράφησης του Τμήματος, για την τεχνική της υποστήριξη και την προθυμία της να συνδράμει όποτε χρειάστηκε. Δεν θα μπορούσα να παραλείψω την οικογένειά μου και τους φίλους μου, που με στήριξαν έμπρακτα και ηθικά καθ' όλη τη διάρκεια της προσπάθειάς μου. Τέλος, ευχαριστώ τους συμφοιτητές μου για την πολύτιμη συντροφιά και τις εποικοδομητικές συζητήσεις που συνέβαλαν στην εξέλιξη αυτής της εργασίας.

Περίληψη

Η παρούσα διπλωματική εργασία αφορά τον σχεδιασμό και την ανάπτυξη ενός διαδικτυακού πληροφοριακού συστήματος με τίτλο «Σύστημα Καταγραφής Διαδικασιών», το οποίο έχει ως στόχο την ψηφιακή απεικόνιση, καταγραφή και υποστήριξη των βασικών διοικητικών διαδικασιών του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου. Η εργασία υλοποιήθηκε σε συνεργασία με τη γραμματεία του Τμήματος, στο πλαίσιο της γενικότερης προσπάθειας για αναβάθμιση και ψηφιοποίηση των εσωτερικών λειτουργιών του Τμήματος.

Η ανάπτυξη του συστήματος οργανώθηκε σε δύο βασικές φάσεις. Στην πρώτη φάση πραγματοποιήθηκε η συλλογή, αποτύπωση και κωδικοποίηση των διοικητικών διαδικασιών του Τμήματος. Για τον σκοπό αυτό αναλύθηκαν οι επίσημοι τόμοι και κανονισμοί του Τμήματος και του Πανεπιστημίου Κύπρου, ενώ παράλληλα πραγματοποιήθηκαν συνεργασίες με μέλη του προσωπικού (όπως ο κ. Χρύσης Γεωργίου και η κα. Σαββούλλα Ευσταθίου), με στόχο την πλήρη και ορθή καταγραφή όλων των απαραίτητων πληροφοριών. Συνολικά καταγράφηκαν 13 επίσημες διαδικασίες, οι οποίες αναλύθηκαν σε διακριτά βήματα, με παράθεση των εμπλεκομένων μερών, των απαιτούμενων αρχείων, των δειγμάτων προς χρήση, καθώς και των σχετικών νομοθετικών πλαισίων και πηγών.

Στη δεύτερη φάση υλοποιήθηκε η εφαρμογή των διαδικασιών μέσω μιας δυναμικής διαδικτυακής πλατφόρμας, βασισμένης στη βιβλιοθήκη React.js. Το σύστημα επιτρέπει στους χρήστες να περιηγούνται στις διαδικασίες, να ξεκινούν προσωπικά αντίγραφα διαδικασιών προς επεξεργασία, να συμπληρώνουν ή να ανεβάζουν αρχεία, να αποστέλλουν μηνύματα ηλεκτρονικού ταχυδρομείου με ή χωρίς συνημμένα αρχεία, και να διαχειρίζονται την πρόοδό τους σε κάθε βήμα της διαδικασίας. Η πλατφόρμα υποστηρίζει επίσης δυνατότητα επισήμανσης βημάτων ως "Ολοκληρωμένα" ή "Μη Ολοκληρωμένα", ενώ μόνο όταν όλα τα βήματα μιας διαδικασίας έχουν ολοκληρωθεί, ο χρήστης μπορεί να την ορίσει ως Συμπληρωμένη, μεταφέροντάς την σε ξεχωριστή ενότητα προβολής.

Σημαντική λειτουργία αποτελεί και η ενσωματωμένη σελίδα Ημερολογίου, όπου ο χρήστης μπορεί να καταχωρεί προσωπικά γεγονότα ή σημειώσεις. Συνολικά, το σύστημα προσφέρει μια εύελικτη, διαδραστική και παραμετροποιήσιμη εμπειρία, που απλουστεύει σημαντικά την εκτέλεση και την παρακολούθηση εσωτερικών διαδικασιών, προσφέροντας στο Τμήμα Πληροφορικής ένα χρήσιμο εργαλείο για την οργάνωση, αποδοτικότητα και διαφάνεια της λειτουργίας του. Για την υλοποίηση της διαδικτυακής πλατφόρμας χρησιμοποιήθηκαν, μεταξύ άλλων, τεχνολογίες όπως το σύστημα διαχείρισης βάσεων δεδομένων phpMyAdmin, γλώσσες προγραμματισμού HTML, JavaScript, PHP και το Laragon το οποίο βοήθησε στην χρήση του εξυπηρετητή Apache, πάνω στο οποίο έτρεχε η ιστοσελίδα.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή	1
1.1	Κίνητρο και Σκοπός Εργασίας	1
1.2	Μεθοδολογία Ανάπτυξης και Χρονοδιαγράμματα	4
1.3	Οργάνωση Κειμένου	5
1.4	Ορισμοί, Ακρωνύμια και Συντομογραφίες	6
Κεφάλαιο 2	Περιγραφή Συστήματος	8
2.1	Απαιτήσεις και Γενική Περιγραφή Συστήματος	8
2.1.1	Απαιτήσεις Διεπαφών	10
2.1.2	Μη Λειτουργικές Απαιτήσεις	12
2.2	Κατηγορίες Χρηστών	13
2.3	Περιορισμοί Συστήματος	14
2.4	Αρχιτεκτονική Συστήματος	15
2.5	Εργαλεία Ανάπτυξης	16
2.5.1	Εργαλείο Visual Studio Code	16
2.5.2	Γλώσσα Περιγραφής HTML	17
2.5.3	Γλώσσα Προγραμματισμού CSS	17
2.5.4	Γλώσσα Προγραμματισμού PHP	18
2.5.5	Γλώσσα Προγραμματισμού JavaScript	19
2.5.6	Βιβλιοθήκη Fullcalendar JS	20
2.5.7	MySQL	20
2.5.8	Σύστημα Διαχείρισης Βάσεων Δεδομένων phpMyAdmin	21
2.5.9	React Toastify	22
2.5.10	Axios	23
2.5.11	React Router DOM	23
2.6	Ασφάλεια Συστήματος	25
2.7	Βάση Δεδομένων	25
2.7.1	Πίνακες Βάσης Δεδομένων	26
2.7.2	Διάγραμμα Οντοτήτων Σχέσεων	34
Κεφάλαιο 3	Χρήση Συστήματος	38
3.1	Σύνδεση	38
3.2	Αρχική Σελίδα	40

3.3 Όλες οι Διαδικασίες	42
3.4 Προσωπικές Διαδικασίες	45
3.4.1 Έναρξη Νέας Διαδικασίας	46
3.4.2 Τρέχουσες Διαδικασίες	47
3.4.3 Συμπληρωμένες Διαδικασίες	51
3.5 Ημερολόγιο	53
Κεφάλαιο 4 Υλοποίηση Συστήματος	56
4.1 Δομή Αρχείων	56
4.2 Βιβλιοθήκες	58
4.3 Παραδείγματα Υλοποίησης	59
4.3.1 Component Structure & State MANAGEMENT	60
4.3.2 Routing	62
4.3.3 Δυναμική Προβολή Δεδομένων	63
4.3.4 Calendar Integration (FullCalendar JS)	64
4.3.5 Επικοινωνία με Backend (axios)	66
4.3.6 Ειδοποιήσεις και Επιβεβαιώσεις (Toastify Notifications και Προσαρμοσμένα Modals)	68
Κεφάλαιο 5 Συμπεράσματα	73
5.1 Συμπεράσματα	73
5.2 Έλεγχος Συστήματος	73
5.3 Επεκτασιμότητα Συστήματος και Μελλοντική Εργασία	74
5.4 Συνεργασία και Εμπειρίες που Αποκτήθηκαν	76
Βιβλιογραφία	77
Παράρτημα Α	-1-
Παράρτημα Β	-7-

Κεφάλαιο 1

Εισαγωγή

1.1 Κίνητρο και Σκοπός Εργασίας	1
1.2 Μεθοδολογία Ανάπτυξης και Χρονοδιαγράμματα	4
1.3 Οργάνωση Κειμένου	5
1.4 Ορισμοί, Ακρωνύμια και Συντομογραφίες	6

1.1 Κίνητρο και Σκοπός Εργασίας

Η ανάγκη για οργάνωση, καταγραφή και αυτοματοποίηση των εσωτερικών διοικητικών διαδικασιών σε ένα ακαδημαϊκό τμήμα είναι σήμερα πιο επίκαιρη από ποτέ. Η ολοένα αυξανόμενη γραφειοκρατία και οι χρονοβόρες χειροκίνητες διαδικασίες δημιουργούν καθυστερήσεις και αυξάνουν την πιθανότητα σφαλμάτων και παραλείψεων. Η παρούσα διπλωματική εργασία προέκυψε ως απάντηση σε αυτήν την ανάγκη, με στόχο την υποστήριξη του διοικητικού και ακαδημαϊκού προσωπικού του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου μέσω της καταγραφής και της ψηφιακής αναπαράστασης των κυριότερων διαδικασιών που ακολουθούνται στο πλαίσιο της καθημερινής λειτουργίας του.

Ωστόσο, αν και οι διαδικασίες αυτές ακολουθούνταν συστηματικά από το προσωπικό, δεν υπήρχε πουθενά καταγεγραμμένη μία σαφής, βηματική και ευανάγνωστη τεκμηρίωση αυτών των 13 διοικητικών διαδικασιών. Η πληροφορία ήταν διάσπαρτη μέσα σε τόμους, εσωτερικούς κανονισμούς και νομοθεσίες του Πανεπιστημίου, χωρίς να είναι εύκολα προσβάσιμη ή άμεσα κατανοητή από κάποιο νέο ή λιγότερο έμπειρο μέλος του Τμήματος. Έτσι, η πρακτική καθημερινή ανάγκη για την ολοκλήρωση συγκεκριμένων ενεργειών βασιζόταν στη στοχευμένη καθοδήγηση ενός ή δύο πιο έμπειρων υπαλλήλων, οι οποίοι γνώριζαν τις διαδικασίες «απ' έξω» λόγω μακρόχρονης εμπειρίας.

Αυτό είχε ως αποτέλεσμα να δημιουργείται εξάρτηση από συγκεκριμένα πρόσωπα, γεγονός που καθιστούσε την εφαρμογή των διαδικασιών ευάλωτη σε περιπτώσεις απουσίας ή αποχώρησής τους. Η απουσία δομημένης πληροφορίας προκαλούσε καθυστερήσεις, ασάφειες

και αύξανε τον κίνδυνο εσφαλμένων ενεργειών. Η παρούσα διπλωματική εργασία λοιπόν προέκυψε ως πρακτική ανάγκη να καλυφθεί αυτό το θεσμικό κενό, δημιουργώντας ένα σύστημα που όχι μόνο κωδικοποιεί τις διαδικασίες με βηματική δομή, αλλά και προσφέρει ένα διαδραστικό περιβάλλον, προσβάσιμο από όλα τα μέλη του διοικητικού και ακαδημαϊκού προσωπικού, ανεξαρτήτως εμπειρίας ή θέσης.

Ο στόχος της εργασίας είναι διττός: αφενός, η συστηματική καταγραφή και κωδικοποίηση των εσωτερικών διοικητικών διαδικασιών του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου, και αφετέρου, η ανάπτυξη ενός λειτουργικού πληροφοριακού συστήματος που να επιτρέπει την αλληλεπίδραση των χρηστών με αυτές τις διαδικασίες.

Η πρώτη πτυχή της εργασίας αφορούσε την αναλυτική καταγραφή και αποσαφήνιση των 13 σημαντικότερων διαδικασιών που εκτελεί το προσωπικό, όπως για παράδειγμα η διαδικασία ανέλιξης Ακαδημαϊκού Προσωπικού, η διαδικασία αξιολόγησης διατριβών, ή η προκήρυξη θέσεων μεταπτυχιακών προγραμμάτων. Για κάθε διαδικασία έγινε διάσπαση σε διακριτά βήματα, με αναφορά στα εμπλεκόμενα πρόσωπα, τα απαραίτητα έγγραφα, τα δείγματα προς χρήση, καθώς και στις θεσμικές πηγές πληροφόρησης (νόμοι, τόμοι, εσωτερικοί κανονισμοί). Το αποτέλεσμα αυτής της φάσης ήταν η δημιουργία ενός πρότυπου στατικού ιστότοπου (σε HTML, CSS και JavaScript) στον οποίο αποτυπώνονται διαδραστικά και με οργανωμένο τρόπο όλες οι διαδικασίες, ώστε να είναι άμεσα προσβάσιμες από οποιονδήποτε χρήστη.

Οι διαδικασίες που καταγράφηκαν, τεκμηριώθηκαν και ενσωματώθηκαν στο σύστημα είναι οι εξής:

1. Εκλογή Ακαδημαϊκού Προσωπικού (ΑΠ)
2. Ανέλιξη Ακαδημαϊκού Προσωπικού
3. Απονομή θέσης Συνεργάτη Καθηγητή ΠΚ
4. Εκλογή Εκπροσώπου Τμήματος στο Συμβούλιο της Σχολής Θετικών και Εφαρμοσμένων Επιστημών (ΣΘΕΕ)
5. Απασχόληση Άμισθου Επισκέπτη Ερευνητή ή Μεταδιδακτορικού Ερευνητή
6. Επισκέπτες Ακαδημαϊκοί
7. Ειδικοί Επιστήμονες Διδασκαλίας
8. Διαδικασία Αξιολόγησης Διατριβής Master
9. Διαδικασία Περιεκτικής Εξέτασης Διδακτορικού Φοιτητή
10. Διαδικασία Ερευνητικής Πρότασης Διδακτορικού Φοιτητή
11. Διαδικασία Διδακτορικής Διατριβής
12. Προκήρυξη και Προσφορά θέσεων στα Μεταπτυχιακά Προγράμματα
13. Χορήγηση Σαββατικής Άδειας

Η δεύτερη πτυχή της εργασίας επικεντρώθηκε στην ανάπτυξη μιας πλήρως λειτουργικής διαδικτυακής πλατφόρμας βασισμένης στη βιβλιοθήκη React, με χρήση PHP και βάσης δεδομένων μέσω του phpMyAdmin. Το πλατφόρμα αυτή επιτρέπει σε κάθε χρήστη να ξεκινήσει προσωπικά μία ή περισσότερες από τις διαθέσιμες διαδικασίες, να ανεβάζει/κατεβάζει αρχεία, να στέλνει email, να σημειώνει την πρόοδο κάθε βήματος και τελικά να ολοκληρώνει τη διαδικασία. Η διεπαφή του χρήστη είναι πλήρως εξατομικευμένη και φιλική προς τον χρήστη, ενώ η δομή του συστήματος ενθαρρύνει την αυτονομία, την τεκμηρίωση και την ευχρηστία, καταργώντας την ανάγκη για προφορική εξάρτηση από έμπειρους συναδέλφους. Οι συμπληρωμένες διαδικασίες αρχειοθετούνται και παραμένουν διαθέσιμες για αναφορά, χωρίς δυνατότητα τροποποίησης, με τα τροποποιημένα έγγραφα του χρήστη να είναι αποθηκευμένα και προσβάσιμα.

Τα οφέλη που αποκόμισα από την παρούσα εργασία είναι πολλαπλά και ουσιαστικά. Καταρχάς, μου δόθηκε η ευκαιρία να συμμετάσχω ενεργά σε μια πραγματική προσπάθεια εκσυγχρονισμού της λειτουργίας του Τμήματος Πληροφορικής, συμβάλλοντας ουσιαστικά στην ενίσχυση της διοικητικής του αποδοτικότητας. Η συνεργασία με άτομα της γραμματείας, όπως η κα. Σαββούλλα Ευσταθίου, αλλά και με τον επιβλέποντα καθηγητή κ. Χρύση Γεωργίου και την αναπληρώτρια καθηγήτρια κα. Γεωργία (Ζέτα) Καπιτσάκη, μου επέτρεψε να κατανοήσω εις βάθος τις ανάγκες του «πελάτη» και να προσαρμόσω τη λύση μου με γνώμονα την πραγματική καθημερινή χρήση.

Επιπλέον, η εργασία αποτέλεσε μια άριστη ευκαιρία εξοικείωσης με ολόκληρο τον κύκλο ζωής ενός λογισμικού συστήματος, από την ανάλυση αναγκών, τη συλλογή απαιτήσεων και τον σχεδιασμό, μέχρι την υλοποίηση, τη δοκιμή και την τεκμηρίωση. Είχα την ευκαιρία να συνδυάσω θεωρητική γνώση με πρακτικές δεξιότητες, αξιοποιώντας σύγχρονες τεχνολογίες όπως React, PHP, HTML, JavaScript και MySQL μέσω του περιβάλλοντος Laragon. Μέσα από τη διαδικασία αυτή, ενίσχυσα όχι μόνο τις τεχνικές μου ικανότητες, αλλά και τις δεξιότητες επικοινωνίας, συνεργασίας και διαχείρισης χρόνου.

Η εργασία εντάσσεται στα πλαίσια της ανάπτυξης λογισμικού (software engineering) και προσφέρει μια ολοκληρωμένη, πρακτική λύση σε ένα υπαρκτό πρόβλημα του τμήματος, με δυνατότητα επέκτασης και προσαρμογής στο μέλλον, τόσο σε άλλα τμήματα όσο και σε διαφορετικές διοικητικές οντότητες.

1.2 Μεθοδολογία Ανάπτυξης και Χρονοδιαγράμματα

Η ανάπτυξη της παρούσας εργασίας ακολούθησε μια μεθοδική και σαφώς δομημένη προσέγγιση, η οποία διακρίνεται σε δύο βασικά στάδια: (α) την καταγραφή και κωδικοποίηση των διαδικασιών, και (β) την υλοποίηση της διαδικτυακής πλατφόρμας.

Στο πρώτο στάδιο, δόθηκε έμφαση στη συλλογή, κατανόηση και διαμόρφωση των διαδικασιών που ακολουθεί το διοικητικό και ακαδημαϊκό προσωπικό του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου. Η εργασία ξεκίνησε με συναντήσεις και τακτική επικοινωνία με τον επιβλέποντα καθηγητή, κ. Χρύση Γεωργίου, και τη γραμματεία του Τμήματος κ. Σαββούλλα Ευσταθίου, με σκοπό την πλήρη αποτύπωση των αναγκών και των υπάρχουσών πρακτικών.

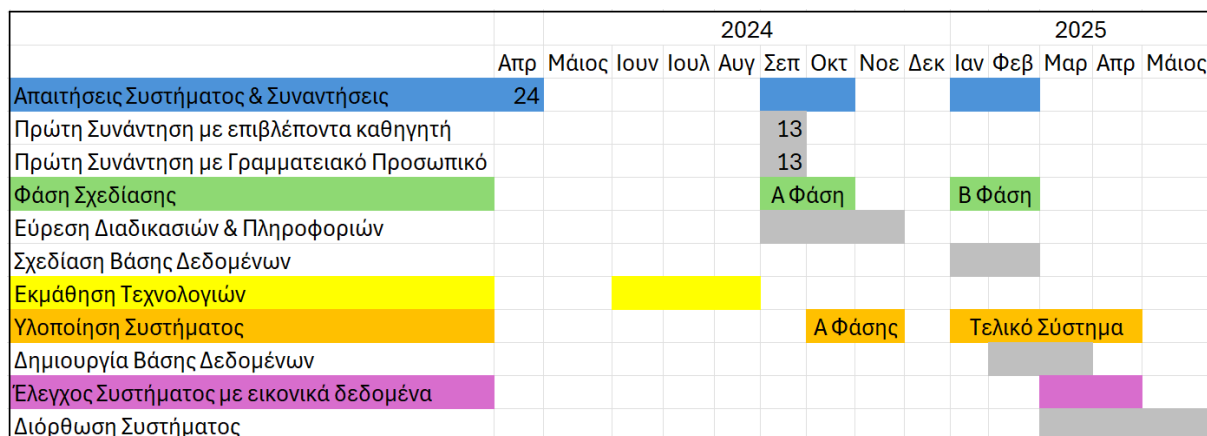
Αξιοποιήθηκαν επίσημοι τόμοι, κανονισμοί και έγγραφα, από τους οποίους αντλήθηκε το απαραίτητο περιεχόμενο για τη δημιουργία 13 διαδικασιών. Οι διαδικασίες αυτές αναλύθηκαν και κωδικοποιήθηκαν σε επιμέρους βήματα, προσδιορίζοντας τα εμπλεκόμενα μέρη, τα απαιτούμενα αρχεία, καθώς και τις αντίστοιχες πηγές τεκμηρίωσης. Ιδιαίτερη συμβολή είχε η συνεργασία με την κ. Σαββούλλα Ευσταθίου, η οποία παρείχε ουσιαστική πληροφόρηση σχετικά με τα πραγματικά δεδομένα και τις απαιτήσεις των διαδικασιών. Επιπλέον, η κ. Σαββούλλα Ευσταθίου παρείχε και το απαραίτητο υλικό τεκμηρίωσης για κάθε διαδικασία, συμπεριλαμβανομένων ενδεικτικών εγγράφων, φορμών και παραδειγμάτων που απαιτούνται κατά την εκτέλεση των επιμέρους βημάτων. Το υλικό αυτό αποτέλεσε αναπόσπαστο μέρος του περιεχομένου που ενσωματώθηκε στο σύστημα, εξασφαλίζοντας την πρακτική του αξία και πληρότητα. Ως αποτέλεσμα της πρώτης φάσης, κατασκευάστηκε μια στατική ιστοσελίδα με HTML, CSS και JavaScript, η οποία λειτούργησε ως προκαταρκτικό μοντέλο παρουσίασης.

Στο δεύτερο στάδιο ακολούθησε η υλοποίηση του διαδραστικού πληροφοριακού συστήματος με τη χρήση React.js για το “frontend” και PHP για το “backend”.

Για την εξακρίβωση των απαιτήσεων του Συστήματος, διεκπεραιώθηκαν συναντήσεις με τον επιβλέποντα καθηγητή κ. Χρύση Γεωργίου και την αναπληρώτρια καθηγήτρια κα. Γεωργία (Ζέτα) Καπιτσάκη. Η βάση δεδομένων διαχειριζόταν μέσω του phpMyAdmin και το Laravel χρησιμοποιήθηκε ως περιβάλλον τοπικής ανάπτυξης. Στην πλατφόρμα ενσωματώθηκαν λειτουργίες όπως έναρξη και διαχείριση διαδικασιών, παρακολούθηση βημάτων, αποστολή email, ανάρτηση αρχείων και προβολή ολοκληρωμένων διαδικασιών. Παράλληλα, υλοποιήθηκε και ένα προσωπικό ημερολόγιο για κάθε χρήστη, μέσω του οποίου είναι δυνατή η διαχείριση μελλοντικών ενεργειών.

Η ανάπτυξη του συστήματος συνοδεύτηκε από συνεχείς δοκιμές και προσαρμογές, βάσει της ανατροφοδότησης που προέκυπτε από τη συνεργασία με τα εμπλεκόμενα μέρη. Παράλληλα, έγινε προσπάθεια βελτιστοποίησης της εμπειρίας χρήσης (UX - User Experience) του συστήματος, μέσα από σχόλια και παρατηρήσεις που προήλθαν από μη-εμπλεκόμενα άτομα, τα οποία αξιολόγησαν τη λειτουργικότητα και την εργονομία της εφαρμογής με “φρέσκια ματιά”. Η συμβολή τους υπήρξε πολύτιμη στην αποσαφήνιση σημείων της διεπαφής που απαιτούσαν επεξηγήσεις ή επανασχεδιασμό.

Το συνολικό χρονοδιάγραμμα της διπλωματικής εργασίας παρατίθεται στο Σχήμα 1.



Σχήμα 1 – Χρονοδιάγραμμα Υλοποίησης Εργασίας

1.3 Οργάνωση Κειμένου

Η παρούσα πτυχιακή εργασία διαρθρώνεται σε πέντε κεφάλαια, με σκοπό την αναλυτική παρουσίαση όλων των σταδίων ανάπτυξης του πληροφοριακού συστήματος και την τεκμηρίωση των αποφάσεων που ελήφθησαν καθ' όλη τη διάρκειά της.

Στο Κεφάλαιο 2, παρουσιάζεται η γενική περιγραφή και οι απαιτήσεις του συστήματος. Περιγράφονται οι λειτουργικές και μη λειτουργικές απαιτήσεις, οι κατηγορίες χρηστών, οι τεχνολογίες που χρησιμοποιήθηκαν, καθώς και η αρχιτεκτονική και η βάση δεδομένων του συστήματος. Το κεφάλαιο αυτό αποσκοπεί στην παροχή ενός πλήρους τεχνικού πλαισίου για την κατανόηση του τρόπου με τον οποίο σχεδιάστηκε και στηρίχθηκε η ανάπτυξη της εφαρμογής.

Στο Κεφάλαιο 3, περιγράφεται αναλυτικά το υλοποιημένο σύστημα από την πλευρά του τελικού χρήστη. Παρουσιάζονται οι επιμέρους λειτουργικές ενότητες της εφαρμογής, όπως η αρχική σελίδα, η σελίδα προσωπικών διαδικασιών, η παρακολούθηση και διαχείριση βημάτων, το ημερολόγιο χρήστη, και η προβολή ολοκληρωμένων διαδικασιών. Γίνεται χρήση

επεξηγηματικών στιγμιότυπων (screenshots) ώστε να γίνει κατανοητός ο τρόπος λειτουργίας του συστήματος στην πράξη.

Το Κεφάλαιο 4 επικεντρώνεται στην τεχνική υλοποίηση του συστήματος. Αρχικά, παρουσιάζεται η ανάπτυξη ενός στατικού πληροφοριακού πρωτοτύπου, το οποίο κατασκευάστηκε με HTML, CSS και JavaScript με σκοπό την οπτική αναπαράσταση των διαδικασιών και την αρχική διερεύνηση της εμπειρίας χρήστη (UX). Στη συνέχεια, αναλύεται η πλήρης υλοποίηση της διαδικτυακής πλατφόρμας, εστιάζοντας στη δομή των αρχείων, στους επιμέρους μηχανισμούς που αναπτύχθηκαν, καθώς και στη συνεργασία των τεχνολογιών React.js, PHP και MySQL για τη δυναμική λειτουργία της εφαρμογής. Περιγράφονται επίσης κρίσιμα τμήματα κώδικα και βασικά σημεία διασύνδεσης μεταξύ “frontend” και “backend”.

Τέλος, το Κεφάλαιο 5 περιλαμβάνει τα συμπεράσματα της εργασίας, την αξιολόγηση του συστήματος ως προς τις αρχικές απαιτήσεις, τις δυνατότητες επεκτασιμότητας και τις προσωπικές εμπειρίες που αποκτήθηκαν κατά τη διάρκεια της ανάπτυξης της εφαρμογής. Αξιολογείται ο βαθμός επίτευξης των στόχων, ενώ καταγράφονται προτάσεις για μελλοντική βελτίωση και ενίσχυση της λειτουργικότητας.

1.4 Ορισμοί, Ακρωνύμια και Συντομογραφίες

Ορισμοί, Ακρωνύμια και Συντομογραφίες	Επεξήγηση
Χρήστης	Οποιοσδήποτε έχει λογαριασμό και χρησιμοποιεί το σύστημα για παρακολούθηση και διαχείριση διαδικασιών.
ΠΚ	Πανεπιστήμιο Κύπρου.
API	Application Programming Interface – Σύνολο συναρτήσεων και διαδικασιών για πρόσβαση σε δεδομένα ή λειτουργίες άλλου λογισμικού.
ΒΔ	Βάση Δεδομένων – Αποθηκευτικός χώρος οργανωμένων δεδομένων, τα οποία χρησιμοποιούνται από το σύστημα.
Server	Ο εξυπηρετητής που φιλοξενεί και εξυπηρετεί το λογισμικό και τη βάση δεδομένων.
React.js	Βιβλιοθήκη JavaScript για την κατασκευή διαδραστικών διεπαφών χρήστη (user interfaces).
PHP	Γλώσσα προγραμματισμού για δυναμική διαχείριση περιεχομένου και επικοινωνία με τον server.
Laragon	Εργαλείο τοπικής ανάπτυξης λογισμικού (development environment) με ενσωματωμένο Apache server, PHP και MySQL.
phpMyAdmin	Διεπαφή διαχείρισης βάσεων δεδομένων MySQL μέσω web περιβάλλοντος.
Directory	Φάκελος στον οποίο αποθηκεύονται αρχεία και υποφάκελοι, οργανώνοντας τη δομή του κώδικα.

Components	Επαναχρησιμοποιήσιμα κομμάτια διεπαφής χρήστη στο React, που υλοποιούν λειτουργικές μονάδες του frontend.
UI	Διεπαφή χρήστη (User Interface)
UX	Εμπειρία χρήστη (User Experience)

Κεφάλαιο 2

Περιγραφή Συστήματος

2.1 Απαιτήσεις και Γενική Περιγραφή Συστήματος	8
2.1.1 Απαιτήσεις Διεπαφών	10
2.1.2 Μη Λειτουργικές Απαιτήσεις	12
2.2 Κατηγορίες Χρηστών	13
2.3 Περιορισμοί Συστήματος	14
2.4 Αρχιτεκτονική Συστήματος	15
2.5 Εργαλεία Ανάπτυξης	16
2.5.1 Εργαλείο Visual Studio Code	16
2.5.2 Γλώσσα Περιγραφής HTML	17
2.5.3 Γλώσσα Προγραμματισμού CSS	17
2.5.4 Γλώσσα Προγραμματισμού PHP	18
2.5.5 Γλώσσα Προγραμματισμού JavaScript	19
2.5.6 Βιβλιοθήκη Fullcalendar JS	20
2.5.7 MySQL	20
2.5.8 Σύστημα Διαχείρισης Βάσεων Δεδομένων phpMyAdmin	21
2.5.9 React Toastify	22
2.5.10 Axios	23
2.5.11 React Router DOM	23
2.6 Ασφάλεια Συστήματος	25
2.7 Βάση Δεδομένων	25
2.7.1 Πίνακες Βάσης Δεδομένων	26
2.7.2 Διάγραμμα Οντοτήτων Σχέσεων	34

2.1 Απαιτήσεις και Γενική Περιγραφή του Συστήματος

Ο καθορισμός των απαιτήσεων αποτελεί θεμέλιο λίθο σε κάθε έργο ανάπτυξης λογισμικού, καθώς αυτές καθοδηγούν ολόκληρο τον κύκλο ζωής του συστήματος. Η λειτουργικότητα και ο σχεδιασμός της εφαρμογής πρέπει να αντανakλούν με ακρίβεια τις ανάγκες του τελικού χρήστη. Στο πλαίσιο της παρούσας εργασίας, στόχος ήταν η ανάπτυξη ενός πληροφοριακού

συστήματος για την καταγραφή, παρουσίαση και παρακολούθηση των γραμματειακών διαδικασιών του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου.

Ο προσδιορισμός των απαιτήσεων έγινε μέσα από τακτικές συναντήσεις με τον επιβλέποντα καθηγητή κ. Χρύση Γεωργίου, την Αναπλ. Καθηγήτρια κα. Γεωργία (Ζέτα) Καπιτσάκη, καθώς και τη στενή συνεργασία με τη γραμματέα του Τμήματος, κα. Σαββούλλα Ευσταθίου. Σκοπός των συζητήσεων ήταν να καταγραφούν οι ανάγκες του γραμματειακού προσωπικού, να αναλυθούν τα επιμέρους βήματα των 13 διαδικασιών που εφαρμόζονται στην πράξη, και να διαμορφωθεί ένα περιβάλλον φιλικό, ευανάγνωστο και πρακτικό.

Το τελικό σύστημα αποτελεί μία διαδικτυακή πλατφόρμα που ενσωματώνεται στην εσωτερική πύλη (Portal) του Τμήματος Πληροφορικής του Πανεπιστημίου. Η πρόσβαση στο σύστημα γίνεται χωρίς ξεχωριστή διαδικασία σύνδεσης (login), καθώς αξιοποιείται το session login του πανεπιστημιακού portal. Ο χρήστης αναγνωρίζεται αυτόματα μέσω του session, από το οποίο λαμβάνονται πληροφορίες όπως το όνομα και η κατάσταση σύνδεσης (logged in).

Όλοι οι χρήστες του συστήματος θεωρούνται ισότιμοι — δεν υπάρχουν διακριτοί ρόλοι ή διαφορετικά επίπεδα πρόσβασης. Οποιοδήποτε μέλος της γραμματείας ή του ακαδημαϊκού προσωπικού έχει τη δυνατότητα να εκκινήσει, παρακολουθήσει και διαχειριστεί διαδικασίες με τον ίδιο τρόπο. Ο σχεδιασμός αυτός στοχεύει στην ευέλικτη χρήση του συστήματος.

Οι βασικές λειτουργίες του Συστήματος περιλαμβάνουν:

- Προβολή Όλων των Διαδικασιών: Εμφανίζονται οι 13 προκαθορισμένες διαδικασίες, με την αναλυτική τους περιγραφή, τα απαιτούμενα βήματα και τα συνοδευτικά αρχεία (π.χ. έντυπα, δείγματα, οδηγίες).
- Έναρξη Νέας Διαδικασίας: Ο χρήστης μπορεί να επιλέξει και να ξεκινήσει μία διαδικασία, η οποία αποθηκεύεται ως προσωπική και εντάσσεται στο περιβάλλον «Τρέχουσες Διαδικασίες».
- Διαχείριση Βημάτων: Για κάθε βήμα μιας διαδικασίας, ο χρήστης έχει δυνατότητα προβολής πρόσθετων πληροφοριών, λήψης ή ανάρτησης αρχείων, αποστολής email προς τον επόμενο εμπλεκόμενο, και σήμανσης του βήματος ως «ολοκληρωμένο» ή «μη ολοκληρωμένο».

- Αρχαιοθέτηση Ολοκληρωμένων Διαδικασιών: Οι πλήρως διεκπεραιωμένες διαδικασίες μεταφέρονται αυτόματα στην ενότητα "Συμπληρωμένες Διαδικασίες" με χρονική σήμανση, συνοδευόμενες με τα τροποποιημένα αρχεία που αναρτήθηκαν από τον χρήστη.
- Προσωπικό Ημερολόγιο: Παρέχεται λειτουργικότητα προσθήκης προσωπικών συμβάντων στο ημερολόγιο, προσβάσιμα αποκλειστικά από τον χρήστη.

Το σύστημα έχει σχεδιαστεί με τρόπο επεκτάσιμο, έτσι ώστε να μπορεί να προσαρμοστεί σε μελλοντικές ανάγκες, όπως προσθήκη νέων διαδικασιών ή ενσωμάτωση με εξωτερικά συστήματα. Ιδιαίτερη έμφαση δόθηκε στη λειτουργικότητα, την εργονομία και την απλότητα, εξασφαλίζοντας έτσι μια θετική εμπειρία χρήσης (UX) για όλα τα μέλη της γραμματείας.

2.1.1 Απαιτήσεις Διεπαφών

Το γραφικό περιβάλλον χρήστη του «Συστήματος Καταγραφής Διαδικασιών» σχεδιάστηκε με γνώμονα την απλότητα, τη διαδραστικότητα και την εύκολη προσβασιμότητα από κάθε μέλος του διοικητικού και ακαδημαϊκού προσωπικού του Τμήματος Πληροφορικής. Πρόκειται για μια διαδικτυακή εφαρμογή που απευθύνεται κυρίως σε σταθερούς και φορητούς υπολογιστές, ενώ έχει διασφαλιστεί και βασική συμβατότητα με φορητές συσκευές μέσω “responsive” σχεδίασης.

Η πρόσβαση στο σύστημα πραγματοποιείται χωρίς ξεχωριστή διαδικασία ταυτοποίησης, καθώς ενσωματώνεται στην πύλη (Portal) του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου. Η αναγνώριση των χρηστών γίνεται αυτόματα μέσω του session του πανεπιστημίου, εξαλείφοντας την ανάγκη για login screen, διαπιστευτήρια ή χειροκίνητη εισαγωγή στοιχείων.

Το περιβάλλον της εφαρμογής ακολουθεί σύγχρονες σχεδιαστικές αρχές και προσεγγίσεις UI/UX, ώστε η πλοήγηση και η χρήση του συστήματος να είναι εύκολη, φιλική και άμεση. Το μενού πλοήγησης στην αριστερή πλευρά της οθόνης παρέχει πρόσβαση σε όλα τα βασικά περιβάλλοντα της εφαρμογής (Αρχική, Όλες οι Διαδικασίες, Προσωπικές Διαδικασίες, Ημερολόγιο), με συνεχή ορατότητα ανεξαρτήτως σελίδας.

Τα βασικά γραφικά στοιχεία διεπαφής (UI elements) που χρησιμοποιήθηκαν είναι:

- **Κουμπιά (Buttons):** Χρησιμοποιούνται σε όλα τα βασικά σημεία της εφαρμογής για την ενεργοποίηση λειτουργιών, όπως έναρξη νέας διαδικασίας, προβολή πληροφοριών, αποστολή email, υποβολή αρχείων και ολοκλήρωση βημάτων.
- **Εισαγωγικά Πεδία (Input Fields):** Παρότι το σύστημα βασίζεται κυρίως στην πλοήγηση και διαχείριση δεδομένων, σε συγκεκριμένα σημεία (όπως το ημερολόγιο ή το email form) επιτρέπεται η εισαγωγή δεδομένων από τον χρήστη (π.χ. τίτλος συμβάντος, παραλήπτης email).
- **Πτυσσόμενες Λίστες (Dropdown Lists):** Παρέχουν εύκολη επιλογή προκαθορισμένων τιμών (όπως τρέχουσες διαδικασίες, επιλογή αρχείου για λήψη ή μεταφόρτωση κ.λπ.).
- **Ημερολόγιο (Calendar Widget):** Κάθε χρήστης διαθέτει προσωπικό ημερολόγιο, στο οποίο μπορεί να προσθέτει, διαγράφει ή τροποποιεί προσωπικά συμβάντα, χωρίς αυτά να είναι ορατά σε άλλους χρήστες.
- **Ετικέτες (Labels):** Χρησιμοποιούνται σε όλο το σύστημα για την οπτική διαφοροποίηση σημαντικών πληροφοριών και ενεργειών (π.χ. Ολοκληρωμένα / Μη Ολοκληρωμένα Βήματα).
- **Μενού Πλοήγησης:** Σταθερά τοποθετημένο στην αριστερή στήλη, επιτρέπει γρήγορη πλοήγηση μεταξύ των βασικών σελίδων. Το μενού είναι πάντα ορατό, προσφέροντας άμεση πρόσβαση σε όλα τα διαθέσιμα περιβάλλοντα του συστήματος.
- **Εικονίδια / Σύμβολα (Icons):** Ενσωματώθηκαν κατάλληλα SVG icons για να ενισχύσουν την κατανόηση λειτουργιών και να κάνουν το περιβάλλον πιο διαδραστικό. Χρησιμοποιούνται σε σημεία όπως τα κουμπιά, οι ειδοποιήσεις και οι ενέργειες των διαδικασιών.
- **Διαδραστικά Cards:** Κάθε διαδικασία ή επιμέρους βήμα εμφανίζεται σε μορφή κάρτας με ξεκάθαρη δομή, εμφανή κουμπιά ενεργειών και δυνατότητα εμφάνισης περισσότερων πληροφοριών με «εναλλαγή κατάστασης» (toggle).

Διεπαφές Υλικού

Για τη χρήση του «Συστήματος Καταγραφής Διαδικασιών» απαιτείται η ύπαρξη μιας βασικής υπολογιστικής συσκευής με δυνατότητα σύνδεσης στο διαδίκτυο. Η εφαρμογή έχει σχεδιαστεί ώστε να είναι πλήρως λειτουργική μέσω προγράμματος περιήγησης (web browser), χωρίς να απαιτείται οποιαδήποτε εγκατάσταση λογισμικού στο τοπικό μηχάνημα του χρήστη.

Οι βασικές διεπαφές υλικού που απαιτούνται είναι:

- Οθόνη προβολής, για την παρουσίαση του γραφικού περιβάλλοντος.
- Πληκτρολόγιο, για την εισαγωγή δεδομένων και πλοήγηση στο ημερολόγιο ή στη συμπλήρωση φορμών.

- Ποντίκι ή Touchpad, για την πλοήγηση, την επιλογή λειτουργιών, την ανάρτηση αρχείων και την εκτέλεση ενεργειών εντός της πλατφόρμας.

Το σύστημα λειτουργεί χωρίς προβλήματα σε σταθερούς υπολογιστές, φορητούς υπολογιστές, αλλά και σε tablets ή κινητές συσκευές, αρκεί να διαθέτουν σύγχρονο φυλλομετρητή (browser) με υποστήριξη JavaScript. Ωστόσο, προτείνεται η χρήση σε “desktop” περιβάλλον για βέλτιστη εμπειρία χρήσης και άνετη διαχείριση των πληροφοριών και των αρχείων.

2.1.2 Μη Λειτουργικές Απαιτήσεις

Η υλοποίηση του Συστήματος Καταγραφής Διαδικασιών έλαβε υπόψη μια σειρά από μη λειτουργικές απαιτήσεις, που κρίθηκαν απαραίτητες για την εύρυθμη, ομαλή και αξιόπιστη λειτουργία της εφαρμογής. Οι απαιτήσεις αυτές αναλύονται ως εξής:

Απαιτήσεις Απόδοσης

Η εφαρμογή σχεδιάστηκε με στόχο την ταχύτερη απόκριση στις ενέργειες του χρήστη, τόσο κατά τη φόρτωση δεδομένων από τη βάση όσο και κατά την πλοήγηση στο περιβάλλον χρήστη. Η αρχιτεκτονική του συστήματος (client-server) και η χρήση τεχνολογιών όπως React.js και PHP, επιτρέπει άμεση ενημέρωση του περιεχομένου μέσω ασύγχρονων αιτήσεων (AJAX), περιορίζοντας τις καθυστερήσεις. Το σύστημα μπορεί να εξυπηρετεί πολλαπλούς χρήστες ταυτόχρονα, με το επίπεδο απόδοσης να εξαρτάται κυρίως από τις δυνατότητες του server φιλοξενίας. Σημαντικό είναι επίσης ότι ο όγκος των δεδομένων που ανταλλάσσονται είναι μικρός, καθιστώντας την εφαρμογή γρήγορη και αποδοτική ακόμη και σε περιβάλλοντα με περιορισμένο εύρος ζώνης.

Απαιτήσεις Ευχρηστίας (Usability)

Το περιβάλλον της εφαρμογής έχει σχεδιαστεί με έμφαση στην απλότητα και την εργονομία. Η πλοήγηση είναι άμεση και ευανάγνωστη, με σαφείς τίτλους, κουμπιά και εικονίδια που καθοδηγούν τον χρήστη στις βασικές λειτουργίες. Οι χρήστες, ακόμα και με ελάχιστη εξοικείωση με τεχνολογικά συστήματα, μπορούν να χρησιμοποιήσουν το σύστημα χωρίς ιδιαίτερη δυσκολία. Για κάθε βασική ενέργεια υπάρχει οπτική ανατροφοδότηση (π.χ. επιβεβαίωση ολοκλήρωσης, μηνύματα σφαλμάτων κλπ), ενώ η ροή χρήσης των λειτουργιών βασίζεται σε λογικά βήματα που ανταποκρίνονται στη φυσική ροή των διαδικασιών. Ο

σχεδιασμός βασίστηκε σε αρχές UX και ενισχύθηκε μέσα από ανατροφοδότηση που ελήφθη από πραγματικούς χρήστες του Τμήματος.

Απαιτήσεις Διαθεσιμότητας

Το σύστημα δεν απαιτεί εγκατάσταση τοπικού λογισμικού, καθώς πρόκειται για διαδικτυακή εφαρμογή. Είναι προσβάσιμο μέσω της Πύλης (Portal) του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου, εξασφαλίζοντας τη διαθεσιμότητά του σε 24ωρη βάση, από οποιαδήποτε συσκευή διαθέτει σύνδεση στο διαδίκτυο και σχετικό session χρήστη. Η σύνδεση πραγματοποιείται μέσω του μηχανισμού ταυτοποίησης του Πανεπιστημίου και το σύστημα αναγνωρίζει τον ενεργό χρήστη από το session. Έτσι, εξασφαλίζεται τόσο η ασφάλεια όσο και η ευκολία πρόσβασης, χωρίς την ανάγκη ξεχωριστής διαδικασίας login ή δημιουργίας λογαριασμού.

Απαιτήσεις Γνώσεων από τον Χρήστη

Η χρήση του συστήματος δεν προϋποθέτει εξειδικευμένες γνώσεις. Ο χρήστης χρειάζεται μόνο βασικές δεξιότητες χειρισμού ηλεκτρονικού υπολογιστή και προγράμματος περιήγησης. Όλες οι λειτουργίες είναι σχεδιασμένες να είναι αυτονόητες, και βασίζονται σε κοινά πρότυπα διεπαφών που συναντώνται σε σύγχρονες διαδικτυακές εφαρμογές.

2.2 Κατηγορίες Χρηστών

Το Σύστημα Καταγραφής Διαδικασιών απευθύνεται αποκλειστικά στα μέλη του διοικητικού και ακαδημαϊκού προσωπικού του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου. Σε αντίθεση με άλλα συστήματα που διαφοροποιούν τους χρήστες ανάλογα με τον ρόλο τους (π.χ. διαχειριστής, υπάλληλος, επόπτης), το συγκεκριμένο σύστημα δεν διαχωρίζει τους χρήστες σε ρόλους ή επίπεδα πρόσβασης. Όλοι οι χρήστες έχουν τις ίδιες δυνατότητες, δικαιώματα και λειτουργίες εντός της εφαρμογής.

Η απόφαση αυτή ελήφθη βάσει της πραγματικής λειτουργίας του Τμήματος, όπου κάθε μέλος του διοικητικού και ακαδημαϊκού προσωπικού ενδέχεται να αναλάβει την εκτέλεση οποιασδήποτε διαδικασίας, ανάλογα με τις εκάστοτε ανάγκες. Συνεπώς, το σύστημα υιοθετεί ένα ενιαίο περιβάλλον εργασίας για όλους τους χρήστες, εξασφαλίζοντας ομοιομορφία στη χρήση και ευκολότερη εκπαίδευση.

Όπως ήδη ειπώθηκε, η πρόσβαση στην εφαρμογή πραγματοποιείται μέσω της πύλης (Portal), όπου το σύστημα αντλεί αυτόματα τα στοιχεία του χρήστη από το session του Πανεπιστημίου, χωρίς να απαιτείται ξεχωριστή διαδικασία σύνδεσης ή δημιουργίας λογαριασμού. Η προσέγγιση αυτή εξασφαλίζει ότι μόνο εξουσιοδοτημένα άτομα του Τμήματος έχουν τη δυνατότητα χρήσης της εφαρμογής, διατηρώντας ταυτόχρονα την ασφάλεια και την απλότητα της εμπειρίας χρήσης.

Επομένως, παρόλο που η εφαρμογή υποστηρίζει πολλαπλούς ταυτόχρονους χρήστες, δεν απαιτείται ιδιαίτερη διαχείριση χρηστών, καθώς κάθε χρήστης λειτουργεί αυτόνομα εντός του προσωπικού του περιβάλλοντος εργασίας, παρακολουθώντας τις δικές του διαδικασίες, χωρίς να έχει πρόσβαση ή δυνατότητα παρέμβασης στις διαδικασίες άλλου χρήστη.

2.3 Περιορισμοί Συστήματος

Το Σύστημα Καταγραφής Διαδικασιών, παρότι σχεδιάστηκε με γνώμονα τη χρηστικότητα, την προσβασιμότητα και την απλότητα, συνοδεύεται από ορισμένους περιορισμούς που αφορούν τόσο τις λειτουργικές του δυνατότητες όσο και τις τεχνολογικές του προδιαγραφές.

Η πρόσβαση στην εφαρμογή προϋποθέτει τη σύνδεση της συσκευής του χρήστη στο διαδίκτυο, ώστε να είναι δυνατή η επικοινωνία με τον εξυπηρετητή και τη βάση δεδομένων. Η εφαρμογή είναι προσβάσιμη αποκλειστικά μέσω του εσωτερικού Portal του Τμήματος Πληροφορικής, και συνεπώς απαιτεί την προηγούμενη επιτυχημένη ταυτοποίηση του χρήστη σε αυτό.

Η εφαρμογή σχεδιάστηκε με γνώμονα την απλότητα και την ευκολία χρήσης, γεγονός που σημαίνει ότι δεν περιλαμβάνει προχωρημένες λειτουργίες ή εξατομικευμένα “dashboards”. Όλες οι σελίδες είναι ενιαίες και κατανοητές, με στόχο ακόμη και χρήστες χωρίς ιδιαίτερη εξοικείωση με ψηφιακά περιβάλλοντα να μπορούν να πλοηγηθούν με ευκολία.

Η διεπαφή της εφαρμογής υποστηρίζει μόνο την Ελληνική γλώσσα. Παρότι καλύπτει πλήρως τις ανάγκες των χρηστών του Τμήματος Πληροφορικής, περιορίζεται στη συγκεκριμένη γλωσσική έκδοση και δεν περιλαμβάνει δυνατότητες μετατροπής σε άλλες γλώσσες.

Αν και πολλοί χρήστες μπορούν να ξεκινήσουν τις δικές τους διαδικασίες, κάθε διαδικασία είναι αποκλειστικά συνδεδεμένη με το χρήστη που την εκκίνησε. Δεν προβλέπεται συνεργατική επεξεργασία της ίδιας διαδικασίας από πολλούς χρήστες ταυτόχρονα.

Οι παραπάνω περιορισμοί δεν μειώνουν την αξία και τη λειτουργικότητα του συστήματος. Αντιθέτως, αποτελούν συνειδητές επιλογές που ενισχύουν την ασφάλεια, την απλότητα και την αποδοτικότητα της εφαρμογής στο συγκεκριμένο περιβάλλον λειτουργίας της.

2.4 Αρχιτεκτονική Συστήματος

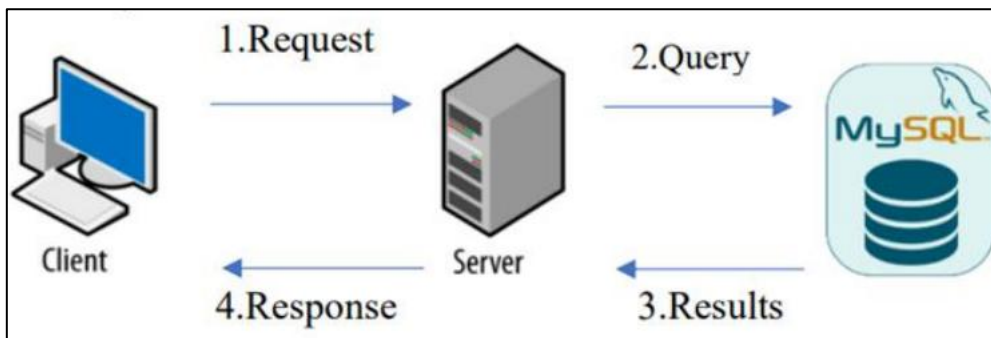
Η αρχιτεκτονική της εφαρμογής βασίζεται στο μοντέλο Πελάτη-Εξυπηρετητή, μια διαδεδομένη και αποδοτική αρχιτεκτονική για διαδικτυακές εφαρμογές. Ο χρήστης (πελάτης) αλληλεπιδρά με το σύστημα μέσω του προγράμματος περιήγησης, αποστέλλοντας αιτήματα (requests) στον εξυπηρετητή (server). Ο εξυπηρετητής, με τη σειρά του, διαχειρίζεται αυτά τα αιτήματα, εκτελεί τις κατάλληλες διεργασίες και επικοινωνεί με τη βάση δεδομένων MySQL [10] για την ανάκτηση ή αποθήκευση πληροφοριών.

Τα δεδομένα που επιστρέφονται από τη βάση δεδομένων στέλνονται στον εξυπηρετητή, ο οποίος τα επεξεργάζεται και αποστέλλει την τελική απάντηση πίσω στον πελάτη (response). Η διεπαφή χρήστη (frontend) χειρίζεται την απόκριση και αποδίδει στην οθόνη την κατάλληλη απεικόνιση.

Η αρχιτεκτονική του συστήματος αξιοποιεί τεχνολογίες React.js [1] για τη δυναμική απεικόνιση του frontend, ενώ το backend υποστηρίζεται από PHP και επικοινωνεί με τη βάση δεδομένων μέσω phpMyAdmin/MySQL [10]. Η δομή αυτή διασφαλίζει σαφή διαχωρισμό των επιπέδων της εφαρμογής, ενισχύοντας τη συντηρησιμότητα και την επεκτασιμότητα του συστήματος.

Η επικοινωνία μεταξύ πελάτη και εξυπηρετητή ακολουθεί τις βασικές αρχές HTTP και RESTful υπηρεσιών [12], χωρίς την ύπαρξη μηχανισμών αυθεντικοποίησης στο ίδιο το σύστημα, καθώς η ταυτοποίηση γίνεται μέσω του τμηματικού portal. Κάθε ενεργός χρήστης αναγνωρίζεται μέσω session, και το σύστημα διαμορφώνει προσωποποιημένο περιβάλλον λειτουργίας με βάση το όνομα χρήστη που επιστρέφεται.

Η παρακάτω εικόνα παρουσιάζει σχηματικά τη βασική ροή της αρχιτεκτονικής:



Εικόνα 1 – Αρχιτεκτονική Συστήματος

2.5 Εργαλεία Ανάπτυξης

Πιο κάτω αναλύονται τα εργαλεία και οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος.

2.5.1 Εργαλείο Visual Studio Code

Το Visual Studio Code (VS Code) [13] είναι ένα ελαφρύ αλλά ισχυρό εργαλείο επεξεργασίας κώδικα, το οποίο αναπτύχθηκε από τη Microsoft και είναι ευρέως διαδεδομένο στην κοινότητα των προγραμματιστών για την ευχρηστία, τις δυνατότητες επέκτασης και την υποστήριξη πολλών γλωσσών προγραμματισμού. Αποτελεί ένα cross-platform περιβάλλον ανάπτυξης που υποστηρίζει λειτουργικά συστήματα Windows, macOS και Linux.

Το VS Code δεν είναι ένα ολοκληρωμένο IDE όπως το Visual Studio, αλλά παρέχει δυνατότητες εντοπισμού σφαλμάτων, IntelliSense (έξυπνη συμπλήρωση κώδικα), terminal, διαχείρισης Git και εγκατάστασης επεκτάσεων (extensions) για υποστήριξη σύγχρονων τεχνολογιών και frameworks όπως React.js, PHP, MySQL, HTML, CSS και JavaScript.

Για την ανάπτυξη της πλατφόρμας του «Συστήματος Καταγραφής Διαδικασιών», το VS Code αποτέλεσε το βασικό περιβάλλον συγγραφής και διαχείρισης του κώδικα για το frontend και το backend μέρος της εφαρμογής. Χάρη στη δυνατότητα εγκατάστασης επεκτάσεων, προστέθηκαν plugins για καλύτερη υποστήριξη της JavaScript και React, βελτιώνοντας την εμπειρία ανάπτυξης και επιταχύνοντας την υλοποίηση της πλατφόρμας.

2.5.2 Γλώσσα Περιγραφής HTML

Η HTML (HyperText Markup Language) [14] αποτελεί τη βασική γλώσσα σήμανσης για την ανάπτυξη ιστοσελίδων και διαδικτυακών εφαρμογών. Παρόλο που δεν είναι γλώσσα προγραμματισμού, αποτελεί δομικό εργαλείο για την οργάνωση και παρουσίαση περιεχομένου στο διαδίκτυο. Μέσω της HTML καθορίζεται η δομή και η ιεραρχία των στοιχείων μιας ιστοσελίδας, όπως τίτλοι, παράγραφοι, πίνακες, σύνδεσμοι, φόρμες και διαδραστικά στοιχεία.

Η σύνταξη της HTML βασίζεται στη χρήση ετικετών (tags), οι οποίες περικλείονται από τα σύμβολα < >. Κάθε ετικέτα δηλώνει τον ρόλο ενός στοιχείου στη σελίδα και συχνά χρησιμοποιείται σε ζεύγη (π.χ. <div></div>, <p></p>), περιλαμβάνοντας περιεχόμενο ή άλλες εσωτερικές δομές.

Στο πλαίσιο της παρούσας εργασίας, η HTML αξιοποιήθηκε για την υλοποίηση όλων των διεπαφών χρήστη (user interfaces) της διαδικτυακής πλατφόρμας. Τόσο η στατική παρουσίαση των διαδικασιών κατά την πρώτη φάση της ανάπτυξης, όσο και η τελική εφαρμογή, βασίστηκαν στη γλώσσα HTML για τη δημιουργία των οθονών, των φορμών και της δομής της πλατφόρμας.

Η χρήση της HTML σε συνδυασμό με τις γλώσσες CSS και JavaScript επέτρεψε την υλοποίηση ενός λειτουργικού και αισθητικά προσεγμένου περιβάλλοντος, προσαρμοσμένου στις ανάγκες των χρηστών του γραμματειακού προσωπικού.

2.5.3 Γλώσσα Προγραμματισμού CSS

Η CSS (Cascading Style Sheets) [15] είναι μια γλώσσα μορφοποίησης η οποία χρησιμοποιείται για να καθορίζει την εμφάνιση και το στυλ των HTML στοιχείων σε μια ιστοσελίδα. Μέσω της CSS, οι προγραμματιστές έχουν τη δυνατότητα να σχεδιάσουν ενιαία και ευέλικτα περιβάλλοντα χρήστη, ρυθμίζοντας ιδιότητες όπως χρώματα, γραμματοσειρές, αποστάσεις, στοίχιση, σκιές και διατάξεις.

Η CSS προτάθηκε από το World Wide Web Consortium (W3C) το 1996 και έκτοτε αποτελεί αναπόσπαστο μέρος της διαδικτυακής ανάπτυξης. Το βασικό της πλεονέκτημα είναι ο διαχωρισμός της λογικής δομής (που παρέχεται από την HTML) από την αισθητική παρουσίαση των δεδομένων, γεγονός που ενισχύει την επαναχρησιμοποίηση, τη συντήρηση και την καθαρότητα του κώδικα.

Στο πλαίσιο αυτής της εργασίας, η CSS εφαρμόστηκε τόσο κατά την ανάπτυξη της στατικής σελίδας κατά την πρώτη φάση, όσο και κατά την τελική υλοποίηση της διαδραστικής πλατφόρμας. Χρησιμοποιήθηκε για να διαμορφώσει ένα καθαρό, λιτό και φιλικό προς τον χρήστη περιβάλλον, στο οποίο κάθε λειτουργία και πληροφορία να είναι ευδιάκριτη και προσβάσιμη.

Η χρήση CSS συνέβαλε καθοριστικά στη διαμόρφωση των σελίδων του συστήματος, προσφέροντας συνεπή αισθητική, σωστή στοίχιση, ελκυστικά κουμπιά και ευανάγνωστα μηνύματα, δημιουργώντας μια εμπειρία πλοήγησης κατάλληλη για χρήστες με διαφορετικά επίπεδα εξοικείωσης με τις τεχνολογίες.

2.5.4 Γλώσσα Προγραμματισμού PHP

Η PHP (Hypertext Preprocessor) [9] είναι μια ευρέως διαδεδομένη γλώσσα σεναρίων ανοιχτού κώδικα (scripting language), σχεδιασμένη ειδικά για την ανάπτυξη δυναμικών ιστοσελίδων και διαδικτυακών εφαρμογών. Πρόκειται για μια server-side γλώσσα, γεγονός που σημαίνει ότι η εκτέλεση του κώδικα πραγματοποιείται στον διακομιστή (server), και το αποτέλεσμα της επεξεργασίας αποστέλλεται στον πελάτη με τη μορφή HTML.

Η PHP υποστηρίζει τόσο διαδικαστικό όσο και αντικειμενοστραφή προγραμματισμό, παρέχοντας ευελιξία και ισχυρές δυνατότητες για την ανάπτυξη πολύπλοκων πληροφοριακών συστημάτων. Αποτελεί σήμερα μια από τις πιο καθιερωμένες λύσεις για το backend development, λόγω της μεγάλης της κοινότητας, της υποστήριξης από δημοφιλή frameworks, και της απλής σύνδεσης με βάσεις δεδομένων όπως η MySQL.

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, η PHP χρησιμοποιήθηκε για τον χειρισμό της επιχειρησιακής λογικής του συστήματος, καθώς και για την επικοινωνία με τη βάση δεδομένων. Πιο συγκεκριμένα, ανέλαβε την υλοποίηση των λειτουργιών του backend, όπως:

- την αποθήκευση και ανάκτηση των δεδομένων των διαδικασιών και των βημάτων τους,
- τη διαχείριση των ανεβασμένων αρχείων των χρηστών,
- την ενημέρωση της κατάστασης των διαδικασιών,
- την επικοινωνία μεταξύ frontend και βάσης δεδομένων μέσω API endpoints.

Η χρήση της PHP ενσωματώθηκε αρμονικά με το περιβάλλον Laragon, προσφέροντας σταθερότητα, ταχύτητα εκτέλεσης και ευκολία διαχείρισης της τοπικής ανάπτυξης. Η επιλογή

της κρίθηκε ιδανική για τις ανάγκες του έργου, λόγω της απλότητας σύνδεσης με MySQL, αλλά και της δυνατότητας δημιουργίας RESTful endpoints για την ανταλλαγή δεδομένων με το React frontend.

2.5.5 Γλώσσα Προγραμματισμού JavaScript

Η JavaScript [8] αποτελεί μια από τις βασικότερες τεχνολογίες του Παγκόσμιου Ιστού, μαζί με την HTML και την CSS. Πρόκειται για μια γλώσσα προγραμματισμού υψηλού επιπέδου, η οποία εκτελείται κυρίως στην πλευρά του πελάτη (client-side), επιτρέποντας τη δημιουργία διαδραστικών και δυναμικών ιστοσελίδων. Η επεξεργασία του κώδικα πραγματοποιείται απευθείας στον φυλλομετρητή (web browser) του χρήστη, με αποτέλεσμα την άμεση ανταπόκριση στις ενέργειές του χωρίς την ανάγκη συνεχούς επικοινωνίας με τον διακομιστή.

Η JavaScript υποστηρίζεται από όλα τα σύγχρονα προγράμματα περιήγησης και είναι συμβατή με το πρότυπο ECMAScript. Υποστηρίζει χαρακτηριστικά όπως αντικειμενοστραφή προγραμματισμό (OOP), προγραμματισμό με βάση τα γεγονότα (event-driven) και λειτουργικό προγραμματισμό (functional programming). Διαθέτει επίσης πρόσβαση στο DOM (Document Object Model), επιτρέποντας την τροποποίηση της δομής και του περιεχομένου μιας ιστοσελίδας σε πραγματικό χρόνο.

Στο πλαίσιο της παρούσας εργασίας, η JavaScript χρησιμοποιήθηκε εκτενώς και σε δύο φάσεις:

- Στη στατική προκαταρκτική σελίδα, συνδυάστηκε με HTML και CSS για την υλοποίηση βασικής διαδραστικότητας και δυναμικής πλοήγησης στις διαδικασίες.
- Στην τελική διαδικτυακή πλατφόρμα, η JavaScript αξιοποιήθηκε κυρίως μέσω της βιβλιοθήκης React.js για την ανάπτυξη του frontend μέρους της εφαρμογής. Παρείχε τη δυνατότητα απόκρισης στις ενέργειες του χρήστη, ενημέρωσης της κατάστασης των διαδικασιών, εναλλαγής μεταξύ σελίδων και υποενότιων χωρίς ανανέωση της σελίδας (SPA – Single Page Application), και ενσωμάτωσης διαδραστικών στοιχείων όπως modals, buttons, ετικέτες και ημερολόγια.

Η JavaScript συνέβαλε καθοριστικά στη δημιουργία μιας ομαλής και αποδοτικής εμπειρίας χρήστη, συνδέοντας το λειτουργικό σχεδιασμό με την πρακτική αλληλεπίδραση.

2.5.6 Βιβλιοθήκη Fullcalendar JS

Η FullCalendar [4] αποτελεί μια ισχυρή και ευρέως διαδεδομένη βιβλιοθήκη JavaScript για την εμφάνιση και διαχείριση συμβάντων μέσω ημερολογίου σε διαδικτυακές εφαρμογές. Είναι ιδιαίτερα κατάλληλη για έργα που απαιτούν τη διαδραστική απεικόνιση γεγονότων, παρέχοντας στον τελικό χρήστη τη δυνατότητα να περιηγηθεί, να δει, να δημιουργήσει ή και να επεξεργαστεί γεγονότα με φιλικό και ευέλικτο τρόπο.

Η βιβλιοθήκη υποστηρίζει διαφορετικές προβολές ημερολογίου (όπως μήνας, εβδομάδα, ημέρα) και επιτρέπει λειτουργίες όπως:

- επιλογή ημερομηνίας,
- εμφάνιση τίτλων και περιγραφών συμβάντων,
- δυνατότητα πλοήγησης και δημιουργίας γεγονότων με modal παραθύρων,
- απεικόνιση δεδομένων με τοπική υποστήριξη γλώσσας (στην περίπτωση της παρούσας εργασίας, χρησιμοποιήθηκε η υποστήριξη ελληνικής γλώσσας μέσω elLocale).

Στην παρούσα εργασία, το FullCalendar ενσωματώθηκε μέσω του React περιβάλλοντος (με χρήση της βιβλιοθήκης @fullcalendar/react και των plugins dayGridPlugin και interactionPlugin). Τα δεδομένα των γεγονότων ανακτώνται και αποθηκεύονται δυναμικά μέσω REST API που αναπτύχθηκε με PHP και επικοινωνεί με τη βάση δεδομένων του συστήματος. Οι χρήστες έχουν τη δυνατότητα να:

- βλέπουν τα συμβάντα τους σε μορφή ημερολογίου,
- δημιουργούν νέα γεγονότα μέσω φόρμας,
- προβάλλουν λεπτομέρειες υπαρχόντων γεγονότων

Η υλοποίηση του ημερολογίου με το FullCalendar βελτιώνει ουσιαστικά την εμπειρία χρήστη, προσφέροντας έναν εύχρηστο και οπτικά κατανοητό τρόπο για την οργάνωση και παρακολούθηση προσωπικών συμβάντων. Η απλότητα ενσωμάτωσης του στο React περιβάλλον και η επεκτασιμότητά του το καθιστούν κατάλληλο εργαλείο για σύγχρονες web εφαρμογές όπως αυτή που αναπτύχθηκε στο πλαίσιο της παρούσας διπλωματικής.

2.5.7 MySQL

Η MySQL [10] είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων ανοικτού κώδικα (Relational Database Management System – RDBMS), το οποίο χρησιμοποιεί την SQL (Structured Query Language) για την αλληλεπίδραση με τις βάσεις δεδομένων. Αποτελεί ένα

από τα πιο διαδεδομένα συστήματα διαχείρισης βάσεων δεδομένων παγκοσμίως και χρησιμοποιείται ευρέως σε διαδικτυακές εφαρμογές.

Η MySQL υποστηρίζει βασικές λειτουργίες όπως:

- δημιουργία, ανάκτηση, τροποποίηση και διαγραφή δεδομένων,
- σχεδιασμό πινάκων και σχέσεων μεταξύ δεδομένων,
- έλεγχο πρόσβασης και εξουσιοδότησης χρηστών,
- διαχείριση ευρετηρίων και βελτιστοποίηση ερωτημάτων.

Το σύστημα παρέχει υψηλή απόδοση, αξιοπιστία και ευελιξία, ενώ είναι συμβατό με πολλαπλές πλατφόρμες λειτουργικών συστημάτων.

Στο πλαίσιο της παρούσας διπλωματικής, υλοποιήθηκε μια βάση δεδομένων MySQL για την αποθήκευση των δεδομένων της πλατφόρμας, όπως:

- οι διαδικασίες και τα επιμέρους βήματά τους,
- τα έγγραφα και τα δείγματα που συνοδεύουν τα βήματα,
- τα προσωπικά γεγονότα του ημερολογίου κάθε χρήστη,
- καθώς και τα τροποποιημένα αρχεία που υποβάλλονται ανά διαδικασία.

Η επικοινωνία με τη βάση δεδομένων πραγματοποιείται μέσω PHP, με χρήση προσαρμοσμένων SQL ερωτημάτων τα οποία αποστέλλονται και επεξεργάζονται στο backend του συστήματος. Τα δεδομένα μεταφέρονται στη διεπαφή χρήστη μέσω REST API endpoints, διασφαλίζοντας την αποδοτική διαχείριση και ανάκτηση των απαιτούμενων πληροφοριών.

Η χρήση της MySQL συνέβαλε καθοριστικά στη διασφάλιση της δομής, συνέπειας και αξιοπιστίας των δεδομένων του συστήματος, ενώ παράλληλα εξασφάλισε επεκτασιμότητα για μελλοντικές λειτουργικές προσθήκες.

2.5.8 Σύστημα Διαχείρισης Βάσεων Δεδομένων phpMyAdmin

Το phpMyAdmin [16] είναι ένα δημοφιλές εργαλείο διαχείρισης βάσεων δεδομένων MySQL μέσω γραφικού περιβάλλοντος στο διαδίκτυο. Πρόκειται για μια διαδικτυακή εφαρμογή που αναπτύχθηκε σε γλώσσα PHP, και επιτρέπει στους διαχειριστές και προγραμματιστές να εκτελούν όλες τις βασικές και προχωρημένες λειτουργίες σε μια βάση δεδομένων, χωρίς την ανάγκη γραφής απευθείας SQL ερωτημάτων.

Μέσω του phpMyAdmin είναι δυνατή η:

- δημιουργία και επεξεργασία πινάκων,
- εισαγωγή και εξαγωγή δεδομένων σε διάφορες μορφές (CSV, SQL, JSON κ.ά.),
- εκτέλεση SQL ερωτημάτων,
- διαχείριση ευρετηρίων και σχέσεων,
- καθώς και ο έλεγχος και η συντήρηση της βάσης δεδομένων.

Καθ' όλη τη διάρκεια της εργασίας, το phpMyAdmin αποτέλεσε το βασικό εργαλείο για τη δημιουργία, διαχείριση και τροποποίηση της βάσης δεδομένων MySQL του συστήματος. Μέσω της φιλικής του διεπαφής, διευκολύνθηκε η πλοήγηση στις εγγραφές, η εισαγωγή αρχικών δεδομένων, καθώς και η αποσφαλμάτωση κατά τη φάση της υλοποίησης και δοκιμών του backend.

Η χρήση του phpMyAdmin συνέβαλε ουσιαστικά στην εύρυθμη ανάπτυξη της βάσης, καθώς και στη γρήγορη οπτική επισκόπηση της δομής και των περιεχομένων της χωρίς την ανάγκη συνεχούς χρήσης της γραμμής εντολών.

2.5.9 React Toastify

Η βιβλιοθήκη React Toastify [17] αποτελεί ένα ιδιαίτερα δημοφιλές εργαλείο για την εμφάνιση προσωρινών ειδοποιήσεων (toast notifications) σε εφαρμογές React. Οι ειδοποιήσεις τύπου “toast” εμφανίζονται προσωρινά στην οθόνη χωρίς να εμποδίζουν τη ροή του συστήματος ή να απαιτούν επιβεβαίωση από τον χρήστη, καθιστώντας τη διάδραση πιο ευέλικτη και αποτελεσματική.

Το React Toastify παρέχει έτοιμα στυλ και μηχανισμούς εμφάνισης μηνυμάτων επιτυχίας, σφάλματος ή προειδοποίησης με ελάχιστο κώδικα. Επιτρέπει επίσης τη ρύθμιση της διάρκειας προβολής, της θέσης εμφάνισης και της δυνατότητας απόκρυψης από τον χρήστη.

Στην παρούσα εφαρμογή, το React Toastify χρησιμοποιήθηκε κυρίως για:

- Εμφάνιση επιβεβαιώσεων κατά την επιτυχή υποβολή ή αποθήκευση δεδομένων (π.χ. γεγονότα στο ημερολόγιο).
- Ενημέρωση του χρήστη για τυχόν σφάλματα κατά την επικοινωνία με τον διακομιστή.
- Αντιστροφή ενεργειών (undo) ή απλές ειδοποιήσεις.

Η χρήση του συνέβαλε στη βελτίωση της εμπειρίας χρήστη, παρέχοντας άμεση ανατροφοδότηση με τρόπο φιλικό και οπτικά διακριτό. Για την ενεργοποίησή του

χρησιμοποιήθηκε το στοιχείο `<ToastContainer />` σε επίπεδο `root`, ενώ τα μηνύματα καλούνταν μέσω `toast.success()` ή `toast.error()` συναρτήσεων.

2.5.10 Axios

Το Axios [3] είναι μια βιβλιοθήκη JavaScript βασισμένη σε promises που χρησιμοποιείται για την αποστολή αιτημάτων HTTP από τον client προς τον server. Υποστηρίζει τόσο τον browser όσο και τον Node.js, και προσφέρει ένα απλό, συνεκτικό API για επικοινωνία με RESTful υπηρεσίες.

Στην παρούσα εφαρμογή, το Axios αποτέλεσε αναπόσπαστο κομμάτι του μηχανισμού επικοινωνίας με τον εξυπηρετητή (server-side PHP API). Χρησιμοποιήθηκε για την αποστολή και λήψη δεδομένων από και προς τη βάση δεδομένων, με τις πιο συχνές περιπτώσεις να περιλαμβάνουν:

- GET requests για ανάκτηση όλων των διαδικασιών, προσωπικών διαδικασιών και γεγονότων ημερολογίου.
- POST requests για δημιουργία νέων γεγονότων ημερολογίου και ανάρτηση αρχείων.
- Χειρισμό σφαλμάτων μέσω των catch blocks για την αποτροπή απρόσμενων συμπεριφορών σε περίπτωση αποτυχίας επικοινωνίας.

Η χρήση του Axios, σε συνδυασμό με τη React, προσφέρει καθαρότητα στον κώδικα, δυνατότητα παραμετροποίησης των headers, καθώς και καλύτερο έλεγχο των απαντήσεων μέσω interceptor μηχανισμών, εφόσον χρειαστεί μελλοντικά.

Παράδειγμα χρήσης:

```
axios
  .get("http://localhost/process-tracking-api/index.php?request=processes")
  .then((response) => setProcesses(response.data.data))
  .catch(() => setProcesses([]));
```

Η επιλογή του Axios αντί του fetch() της JavaScript έγινε για την αυξημένη ευχρηστία και σαφήνεια που προσφέρει στην ανάγνωση και διαχείριση των αιτημάτων.

2.5.11 React Router DOM

Το React Router DOM [2] είναι η βιβλιοθήκη πλοήγησης (routing library) που χρησιμοποιείται στο οικοσύστημα του React για την υλοποίηση πλοήγησης μεταξύ διαφορετικών σελίδων ή views μιας μονοσέλιδης εφαρμογής (Single Page Application – SPA).

Σε αντίθεση με τις παραδοσιακές εφαρμογές όπου κάθε σελίδα απαιτεί πλήρη φόρτωση από τον server, το React Router DOM επιτρέπει την αλλαγή περιεχομένου εντός της ίδιας σελίδας, χωρίς ανανέωση του φυλλομετρητή, προσφέροντας έτσι μία πιο ομαλή, γρήγορη και responsive εμπειρία χρήστη.

Χρήση στο πλαίσιο της εργασίας:

Η τεχνολογία React Router DOM χρησιμοποιήθηκε για τη διαχείριση των διαδρομών της εφαρμογής, επιτρέποντας στους χρήστες να μεταβαίνουν στις ακόλουθες βασικές ενότητες του συστήματος:

- Αρχική σελίδα (/)
- Όλες οι Διαδικασίες (/all-processes)
- Προσωπικές Διαδικασίες (/my-processes)
- Ημερολόγιο (/calendar)

Η πλοήγηση επιτυγχάνεται με τη χρήση των React components όπως <BrowserRouter>, <Routes>, <Route>, και <NavLink>. Κάθε διαδρομή οδηγεί στο αντίστοιχο component, χωρίς επαναφόρτωση της σελίδας.

Παράδειγμα:

```
<Routes>
  { /* Set Dashboard as the default route */ }
  <Route path="/" element={<Dashboard />} />
  <Route path="/all-processes" element={<AllProcesses />} />
  <Route path="/my-processes" element={<MyProcesses />} />
  <Route path="/my-processes/all" element={<AllMyProcesses />} />
  <Route path="/my-processes/running" element={<RunningProcesses />} />
  <Route path="/my-processes/completed" element={<CompletedProcesses />} />
  <Route path="/calendar" element={<Calendar />} />
</Routes>
```

Οφέλη από τη χρήση:

- Δεν απαιτείται ανανέωση σελίδας κατά την αλλαγή περιεχομένου.
- Αυξημένη ταχύτητα και απόκριση του UI.
- Καλύτερη διαχείριση της εμπειρίας του χρήστη.
- Ευελιξία στον ορισμό και στην προστασία διαδρομών.

Το React Router DOM συνέβαλε ουσιαστικά στη δημιουργία μιας συνεκτικής και δυναμικής εμπειρίας πλοήγησης, διατηρώντας την εφαρμογή εύχρηστη και μοντέρνα.

2.6 Ασφάλεια Συστήματος

Η ασφάλεια αποτελεί κεντρικό πυλώνα σχεδιασμού της εφαρμογής και διασφαλίζεται μέσα από μια συνδυαστική προσέγγιση που περιλαμβάνει μηχανισμούς αυθεντικοποίησης, ελέγχου πρόσβασης και διαχείρισης ρόλων.

Η διαδικασία αυθεντικοποίησης των χρηστών δεν υλοποιείται τοπικά εντός της εφαρμογής αλλά βασίζεται στην ήδη υπάρχουσα υποδομή του Πανεπιστημίου Κύπρου. Συγκεκριμένα, οι χρήστες συνδέονται χρησιμοποιώντας τους ιδρυματικούς λογαριασμούς που διαχειρίζεται το Active Directory. Με την επιτυχή σύνδεση, ο web server δημιουργεί session και αποθηκεύει στο global αντικείμενο `$_SESSION` κρίσιμες μεταβλητές.

Πριν επιτραπεί οποιαδήποτε πρόσβαση στην εφαρμογή, διενεργείται έλεγχος αυτών των μεταβλητών. Η εφαρμογή επιβεβαιώνει αρχικά ότι ο χρήστης είναι πράγματι συνδεδεμένος και ακολούθως ότι διαθέτει τον απαραίτητο ρόλο, καθώς το σύστημα προορίζεται αποκλειστικά για χρήση από γραμματειακά μέλη του Τμήματος Πληροφορικής. Πραγματοποιείται επιπλέον επαλήθευση έναντι της τοπικής βάσης δεδομένων της εφαρμογής, κατά την οποία ελέγχεται εάν το username του συνδεδεμένου χρήστη υπάρχει καταχωρημένο στον πίνακα users. Με αυτόν τον τρόπο, επιτυγχάνεται ένα δεύτερο επίπεδο περιορισμού της πρόσβασης, ώστε μόνο εξουσιοδοτημένα μέλη να έχουν δυνατότητα εισόδου, ακόμη και αν διαθέτουν έγκυρο ιδρυματικό λογαριασμό με κατάλληλο ρόλο.

Σε περίπτωση που δεν πληρούνται οι παραπάνω συνθήκες, η εφαρμογή προβαίνει σε αυτόματη ανακατεύθυνση του χρήστη σε κατάλληλη σελίδα απόρριψης πρόσβασης και θα εμφανίζει μήνυμα που τον ενημερώνει για την απουσία των απαραίτητων δικαιωμάτων.

2.7 Βάση Δεδομένων

Σε αυτό το κεφάλαιο παρουσιάζονται το διάγραμμα οντοτήτων συσχετίσεων και οι πίνακες της Βάσης Δεδομένων, που αναπτύχθηκε για τις ανάγκες της πλατφόρμας.

2.7.1 Πίνακες Βάσης Δεδομένων

Πιο κάτω παρουσιάζεται αναλυτική επεξήγηση κάθε πίνακα της βάσης δεδομένων που προέκυψαν από το διάγραμμα οντοτήτων σχέσεων.

Πίνακας Users

Ο πίνακας users αποτελεί τον βασικό πίνακα για την ταυτοποίηση των χρηστών του συστήματος. Περιλαμβάνει σημαντικά γνωρίσματα για τον καθορισμό της ταυτότητας κάθε χρήστη, όπως το όνομα, το email και μοναδικό αναγνωριστικό (id).

#	Name	Type
1	id 	int
2	user_id 	int
3	title	varchar(255)
4	description	text
5	date_time	timestamp
6	created_at	timestamp

Keyname	Type	Unique	Packed	Column
PRIMARY	BTREE	Yes	No	id
user_id	BTREE	No	No	user_id

Πίνακας Processes

Ο πίνακας processes περιέχει τα βασικά μεταδεδομένα κάθε καταγεγραμμένης διαδικασίας στο σύστημα. Αποτελεί ένα από τα κεντρικά σημεία αναφοράς της εφαρμογής, αφού κάθε διαδικασία περιγράφεται και συνδέεται με τα αντίστοιχα βήματα, πληροφορίες, καθώς και με ενέργειες των χρηστών.

Γνωρίσματα:

- name: Το όνομα της διαδικασίας.
- description: Αναλυτική περιγραφή της διαδικασίας.
- reference_data και reference_name: Πεδία τεκμηρίωσης που περιλαμβάνουν συνήθως πληροφορίες ή αναφορές (π.χ. σχετική νομοθεσία, έγγραφα, εσωτερικοί κανονισμοί).
- created_at, updated_at: Χρονικά σημεία δημιουργίας και τελευταίας ενημέρωσης.

#	Name	Type
1	id 	int
2	name	varchar(255)
3	description	text
4	created_at	timestamp
5	updated_at	timestamp
6	reference_data	text
7	reference_name	text

Πίνακας Steps

Ο πίνακας steps ορίζει όλα τα βήματα που συνθέτουν μία διαδικασία του συστήματος. Κάθε διαδικασία (process) αποτελείται από διαδοχικά βήματα που περιγράφουν τη ροή ενεργειών που πρέπει να ακολουθηθούν από τον χρήστη. Ο σχεδιασμός αυτός επιτρέπει τη δομημένη παρακολούθηση και εκτέλεση διαδικασιών με σαφή προσδιορισμό ενδιάμεσων σταδίων.

Γνωρίσματα:

- **id:** Πρωτεύον κλειδί του πίνακα. Μοναδικό αναγνωριστικό κάθε βήματος.
- **process_id:** Ξένο κλειδί που συνδέει το βήμα με μια συγκεκριμένη διαδικασία στον πίνακα processes.
- **step_number:** Καθορίζει τη σειρά εκτέλεσης του βήματος μέσα στη διαδικασία. Μαζί με το process_id, σχηματίζουν ένα μοναδικό συνδυασμό για την ταυτοποίηση του σταδίου.
- **title:** Τίτλος του βήματος, περιγράφει συνοπτικά τη φύση του (π.χ. "Έγκριση από ΕΠΣΤ").
- **description:** Αναλυτική περιγραφή του περιεχομένου ή των απαιτήσεων του βήματος.
- **created_at, updated_at:** Χρονικά σημεία δημιουργίας και τελευταίας τροποποίησης.
- **supplementary_details:** Προαιρετικό πεδίο για επιπρόσθετες πληροφορίες ή οδηγίες.
- **upload_btn:** Boolean (tinyint), υποδεικνύει αν απαιτείται από τον χρήστη να ανεβάσει αρχείο στο συγκεκριμένο βήμα.
- **sendfile_btn:** Boolean (tinyint), υποδεικνύει αν το βήμα ενεργοποιεί την αποστολή του ανεβασμένου αρχείου προς έλεγχο ή επόμενο χειρισμό.

#	Name	Type
1	id 🔑	int
2	process_id 🔑	int
3	step_number 🔑	int
4	title	enum('Επιτροπή Μεταπτυχιακών Σπουδών του Τμήματος')
5	description	text
6	created_at	timestamp
7	updated_at	timestamp
8	supplementary_details	text
9	upload_btn	tinyint(1)
10	sendfile_btn	tinyint(1)

Keyname	Type	Unique	Packed	Column
PRIMARY	BTREE	Yes	No	id
process_id	BTREE	Yes	No	process_id step_number

Πίνακας running_processes

Ο πίνακας running_processes καταγράφει όλες τις διαδικασίες που βρίσκονται σε εξέλιξη από τους χρήστες. Κάθε εγγραφή αντιστοιχεί σε μία διαδικασία που έχει ξεκινήσει από συγκεκριμένο χρήστη, με παρακολούθηση της προόδου της.

Γνωρίσματα:

- user_id: Το αναγνωριστικό του χρήστη που ξεκίνησε τη διαδικασία.
- process_id: Η διαδικασία που έχει ενεργοποιηθεί.
- step_id: Το τρέχον βήμα στο οποίο βρίσκεται η διαδικασία.
- started_at, completed_at: Τα χρονικά σημεία έναρξης και ενδεχόμενης ολοκλήρωσης της διαδικασίας.
- process_name: Το όνομα της διαδικασίας για ευκολότερη αναγνώριση.
- step_status: Η κατάσταση του βήματος, δηλωμένη ως Pending ή Completed.

#	Name	Type
1	id 🔑	int
2	user_id 🔑	int
3	process_id 🔑	int
4	step_id 🔑	int
5	started_at	timestamp
6	completed_at	timestamp
7	created_at	timestamp
8	updated_at	timestamp
9	process_name	varchar(255)
10	step_status	enum('Pending', 'Completed')

Keyname	Type	Unique	Packed	Column
PRIMARY	BTREE	Yes	No	id
user_id	BTREE	No	No	user_id
process_id	BTREE	No	No	process_id
step_id	BTREE	No	No	step_id

Πίνακας running_steps

Ο πίνακας running_steps καταγράφει την πρόοδο εκτέλεσης κάθε βήματος (step) για μια διαδικασία που βρίσκεται σε εξέλιξη. Αποτελεί κρίσιμο κομμάτι του συστήματος καθώς υποστηρίζει την παρακολούθηση και καταγραφή της κατάστασης κάθε βήματος μιας ενεργής διαδικασίας από κάθε χρήστη ξεχωριστά.

Γνωρίσματα:

- **id**: Πρωτεύον κλειδί του πίνακα.
- **running_process_id**: Αναφέρεται στην ενεργή διαδικασία στην οποία ανήκει το βήμα.
- **step_id**: Αναφέρεται στο μοναδικό id του βήματος της αντίστοιχης διαδικασίας.
- **user_id**: Ο χρήστης που εκτελεί ή έχει την ευθύνη του συγκεκριμένου βήματος.
- **status**: Κατάσταση του βήματος (Pending ή Completed), δείχνοντας αν έχει ολοκληρωθεί ή όχι.
- **updated_at**: Χρονική σφραγίδα που δείχνει την τελευταία τροποποίηση (status update) του βήματος.

#	Name	Type
1	id 🔑	int
2	user_id 🔑	int
3	process_id 🔑	int
4	step_id 🔑	int
5	started_at	timestamp
6	completed_at	timestamp
7	created_at	timestamp
8	updated_at	timestamp
9	process_name	varchar(255)
10	step_status	enum('Pending', 'Completed')

Keyname	Type	Unique	Packed	Column
PRIMARY	BTREE	Yes	No	id
user_id	BTREE	No	No	user_id
process_id	BTREE	No	No	process_id
step_id	BTREE	No	No	step_id

Πίνακας completed_processes

Ο πίνακας completed_processes καταγράφει όλες τις διαδικασίες που έχουν ολοκληρωθεί από χρήστες του συστήματος. Κάθε εγγραφή αντιστοιχεί σε μια επιτυχώς διεκπεραιωμένη διαδικασία.

Γνωρίσματα:

- id: Πρωτεύον κλειδί.
- user_id: Αναφορά στον χρήστη που ολοκλήρωσε τη διαδικασία.
- process_id: Αναφορά στη διαδικασία που ολοκληρώθηκε.
- process_name: Το όνομα της διαδικασίας κατά τη στιγμή ολοκλήρωσης.
- completed_at: Χρονική σήμανση της ολοκλήρωσης.

#	Name	Type
1	id 🔑	int
2	user_id 🔑	int
3	process_id 🔑	int
4	process_name	varchar(255)
5	completed_at	timestamp

Πίνακας completed_steps

Ο πίνακας completed_steps αποθηκεύει πληροφορίες για όλα τα βήματα που έχουν ολοκληρωθεί στα πλαίσια μιας διαδικασίας που έχει κλείσει. Πρόκειται για μια ιστορική καταγραφή που επιτρέπει την ανασκόπηση των ενεργειών που πραγματοποιήθηκαν από τον χρήστη.

Γνωρίσματα:

- id: Πρωτεύον κλειδί της εγγραφής.
- completed_process_id: Ξένο κλειδί που παραπέμπει στον πίνακα completed_processes.
- step_number: Ο αύξων αριθμός του βήματος εντός της διαδικασίας.
- title: Ο τίτλος του συγκεκριμένου βήματος.
- description: Περιγραφή του βήματος που ολοκληρώθηκε.
- supplementary_details: Προαιρετικό πεδίο για επιπρόσθετες πληροφορίες ή οδηγίες.

#	Name	Type
1	id 🔑	int
2	completed_process_id 🔗	int
3	step_number	int
4	title	varchar(255)
5	description	text
6	supplementary_details	text

Πίνακας completed_files

Ο πίνακας completed_files αποθηκεύει τα αρχεία που έχουν ανέβει από χρήστες κατά την ολοκλήρωση μιας διαδικασίας. Παρέχει πολύτιμη πληροφορία για το πότε, ποιος και σε ποιο βήμα ανέβασε το κάθε αρχείο.

Γνωρίσματα:

- id: Πρωτεύον κλειδί της εγγραφής.
- completed_process_id: Ξένο κλειδί που παραπέμπει στην ολοκληρωμένη διαδικασία (completed_processes).
- completed_step_id: Ξένο κλειδί προς το σχετικό ολοκληρωμένο βήμα (completed_steps).
- user_id: Ο χρήστης που ανέβασε το αρχείο.
- file_name: Το όνομα του αρχείου όπως το ανέβασε ο χρήστης.
- file_path: Η διαδρομή αποθήκευσης του αρχείου στον server.
- uploaded_at: Χρονική στιγμή κατά την οποία έγινε η μεταφόρτωση.

#	Name
1	id 
2	completed_process_id 
3	completed_step_id 
4	user_id 
5	file_name
6	file_path
7	uploaded_at

Πίνακας samples

Ο πίνακας samples αποθηκεύει ενδεικτικά αρχεία που συνδέονται με συγκεκριμένα βήματα κάθε διαδικασίας, με σκοπό την παροχή παραδειγμάτων προς καθοδήγηση των χρηστών κατά την εκτέλεση των βημάτων.

Αποτελεί σημαντικό στοιχείο για την τεκμηρίωση και επεξήγηση της κάθε διαδικασίας, καθώς κάθε χρήστης μπορεί να δει ή να κατεβάσει τα συγκεκριμένα αρχεία (π.χ. δείγματα αιτήσεων, φόρμες, κείμενα).

Γνωρίσματα:

- id: Πρωτεύον κλειδί, μοναδικό αναγνωριστικό του δείγματος.
- step_id: Ξένο κλειδί που αναφέρεται στο σχετικό βήμα της διαδικασίας (πίνακας steps).
- file_name: Το όνομα του αρχείου.
- displayed_name: Το εμφανιζόμενο όνομα του αρχείου (π.χ. για εμφάνιση στη διεπαφή).
- file_path: Η διαδρομή αποθήκευσης του αρχείου στο σύστημα.
- uploaded_at: Χρονική στιγμή αποθήκευσης του δείγματος.

#	Name	Type
1	id 	int
2	step_id 	int
3	file_name	varchar(255)
4	displayed_name	varchar(255)
5	file_path	varchar(500)
6	uploaded_at	timestamp

Keyname	Type	Unique	Packed	Column
PRIMARY	BTREE	Yes	No	id
samples_ibfk_1	BTREE	No	No	step_id

Πίνακας sample_versions/uploads

Ο πίνακας sample_versions διαχειρίζεται τις διαφορετικές εκδόσεις αρχείων που ανεβαίνουν ως δείγματα σε συγκεκριμένα βήματα (steps) μίας διαδικασίας σε εξέλιξη (running process). Παρέχει τη δυνατότητα αποθήκευσης πολλαπλών εκδόσεων για το ίδιο δείγμα, επιτρέποντας την ανίχνευση αλλαγών ή διορθώσεων.

Γνωρίσματα:

- id: Πρωτεύον κλειδί, μοναδικό αναγνωριστικό κάθε εγγραφής.
- step_id: Το βήμα της διαδικασίας στο οποίο σχετίζεται το δείγμα.
- running_process_id: Η τρέχουσα διαδικασία στην οποία ανήκει το δείγμα.
- user_id: Ο χρήστης που ανέβασε την εκάστοτε έκδοση.
- file_name: Το όνομα του αρχείου στο σύστημα.
- file_path: Η διαδρομή στο σύστημα αρχείων όπου αποθηκεύεται το αρχείο.
- uploaded_at: Η χρονική στιγμή ανάρτησης της έκδοσης.

#	Name	Type
1	id 	int
2	step_id 	int
3	running_process_id 	int
4	user_id 	int
5	file_name	varchar(255)
6	file_path	varchar(500)
7	uploaded_at	timestamp

Keyname	Type	Unique	Packed	Column
PRIMARY	BTREE	Yes	No	id
user_id	BTREE	No	No	user_id
running_process_id	BTREE	No	No	running_process_id
sample_versions_ibfk_1	BTREE	No	No	step_id

Πίνακας process_information

Ο πίνακας process_information σχεδιάστηκε με σκοπό να καλύψει διαδικασίες που δεν ακολουθούν αυστηρή ροή βημάτων (όπως αυτές που υλοποιούνται μέσω του πίνακα steps), αλλά αποτελούνται από πληροφοριακά τμήματα που παρουσιάζουν θεωρητικά ή γενικά στοιχεία στον χρήστη. Χρησιμοποιείται κυρίως για τη διαδικασία «Διορισμός Ειδικής Επιτροπής για τις Βαθμίδες Καθηγητή και Αναπληρωτή Καθηγητή», η οποία δεν απαιτεί διαδραστική εμπλοκή από τους χρήστες.

Γνωρίσματα:

- **id**: Πρωτεύον κλειδί.
- **process_id**: Αναφορά στη διαδικασία που αφορά η πληροφορία.
- **section_title**: Ο τίτλος της ενότητας που παρουσιάζεται.
- **content**: Το περιεχόμενο της ενότητας, το οποίο μπορεί να περιλαμβάνει επεξηγηματικά κείμενα ή οδηγίες.

#	Name	Type
1	id 🔑	int
2	process_id 🔑	int
3	section_title	varchar(255)
4	content	text

Πίνακας **calendar_events**

Ο πίνακας **calendar_events** χρησιμοποιείται για την αποθήκευση των προσωπικών γεγονότων (events) που καταχωρούν οι χρήστες στο ημερολόγιο της πλατφόρμας. Αποτελεί κεντρικό στοιχείο της λειτουργικότητας ημερολογίου, παρέχοντας στον χρήστη τη δυνατότητα να οργανώνει και να παρακολουθεί προγραμματισμένες ενέργειες ή υπενθυμίσεις.

Γνωρίσματα:

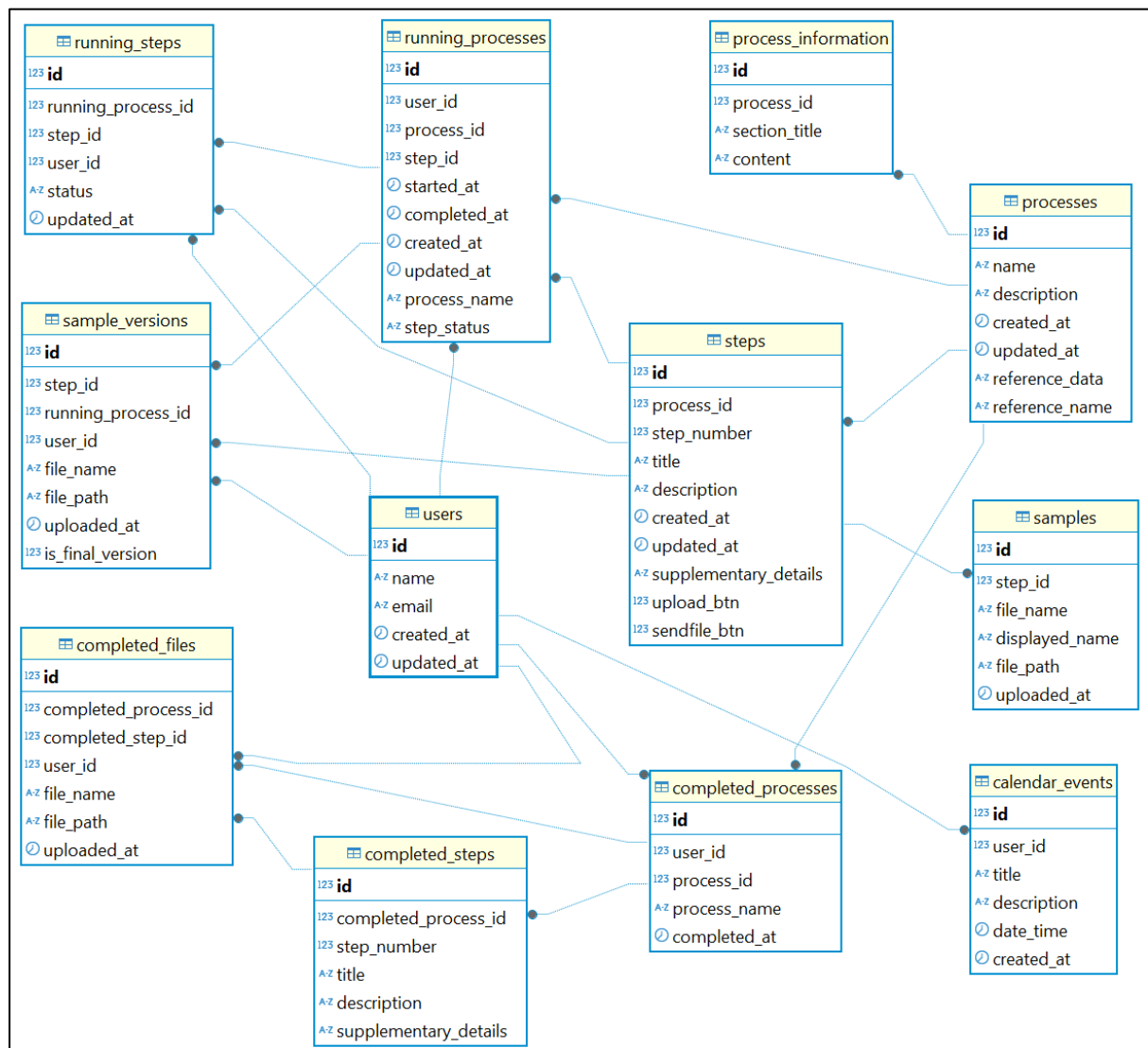
- **id**: Πρωτεύον κλειδί.
- **user_id**: Αναφέρεται στον χρήστη στον οποίο ανήκει το γεγονός.
- **title**: Ο τίτλος του γεγονότος.
- **description**: Προαιρετική περιγραφή του γεγονότος.
- **date_time**: Η χρονική στιγμή κατά την οποία έχει προγραμματιστεί το γεγονός.
- **created_at**: Η χρονική στιγμή κατά την οποία καταχωρήθηκε το γεγονός στο σύστημα.

#	Name	Type
1	id 🔑	int
2	user_id 🔑	int
3	title	varchar(255)
4	description	text
5	date_time	timestamp
6	created_at	timestamp

2.7.2 Διάγραμμα Οντοτήτων Σχέσεων

Στην Εικόνα 2 παρουσιάζεται το επεκταμένο διάγραμμα οντοτήτων σχέσεων της βάσης δεδομένων.

Τα διαγράμματα οντοτήτων σχέσεων (ERD) χρησιμοποιούνται για την εννοιολογική αναπαράσταση δεδομένων.



Εικόνα 2 – Διάγραμμα Οντοτήτων Σχέσεων

Συσχετίσεις Πινάκων:

Για την καλύτερη κατανόηση της βάσης δεδομένων, ακολουθεί μια αναλυτική περιγραφή των βασικών πινάκων και των μεταξύ τους σχέσεων. Η ανάλυση που παρουσιάζεται παρακάτω εξηγεί πώς οι πίνακες αλληλεπιδρούν μεταξύ τους, πώς αποτυπώνονται οι ενέργειες των χρηστών και πώς οργανώνονται οι διαδικασίες, τα βήματα, τα αρχεία και τα γεγονότα μέσα στη βάση. Με τον τρόπο αυτό, επιτυγχάνεται η συνοχή των δεδομένων και η λειτουργικότητα της πλατφόρμας σε κάθε στάδιο χρήσης της.

Ένας χρήστης (user):

- Μπορεί να έχει πολλά calendar events (calendar_events.user_id).
- Μπορεί να συμμετέχει σε πολλές διεργασίες (running_processes.user_id και completed_processes.user_id).
- Μπορεί να ανεβάσει αρχεία ή να καταθέσει δείγματα (completed_files.user_id, sample_versions.user_id).

Μια διαδικασία (process) συνδέεται:

- Με τον πίνακα steps: Μια διαδικασία μπορεί να περιέχει πολλαπλά βήματα.
- Με τον πίνακα running_processes και completed_processes: Υλοποιήσεις/εφαρμογές της ίδιας διαδικασίας από διάφορους χρήστες.
- Με τον πίνακα process_information: Συμπληρωματικές πληροφορίες που συνοδεύουν κάθε διαδικασία.

Κάθε βήμα (step):

- ανήκει σε μία διαδικασία (πολλά προς ένα με τον πίνακα processes).
- μπορεί να σχετίζεται με πολλά running_steps (κατά την ενεργή εκτέλεση μιας διαδικασίας από χρήστη).
- μπορεί να έχει συσχετισμένα δείγματα εγγράφων (samples), εκδόσεις (sample_versions) και καταχωρημένα αρχεία (completed_files) ανά εκτέλεση.

Κάθε τρέχουσα διαδικασία (running_processes):

- Αναφέρεται σε ένα συγκεκριμένο βήμα (το πρώτο) της διαδικασίας μέσω του πεδίου step_id, το οποίο συνδέεται με τον πίνακα steps.
- Μπορεί να σχετίζεται με πολλά επιμέρους βήματα σε εκτέλεση μέσω του πίνακα running_steps (running_steps.running_process_id).
- Μπορεί να έχει πολλαπλές εκδόσεις αρχείων ή υποβληθέντα δείγματα μέσω του πίνακα sample_versions.

Κάθε τρέχον βήμα (running step):

- ανήκει σε μία τρέχουσα διαδικασία μέσω του running_process_id.
- αντιστοιχεί σε ένα βήμα μιας προκαθορισμένης διαδικασίας μέσω του step_id.
- εκτελείται από έναν χρήστη μέσω του user_id.

Κάθε completed process:

- ανήκει σε έναν χρήστη μέσω του user_id.
- σχετίζεται με μία διαδικασία μέσω του process_id.
- μπορεί να συσχετίζεται με ένα ή περισσότερα αρχεία (π.χ. στον πίνακα completed_files).
- Συνδέεται με τα ολοκληρωμένα βήματα της διαδικασίας (πίνακας completed_steps).

Κάθε ολοκληρωμένο βήμα (completed step):

- ανήκει σε μία ολοκληρωμένη διαδικασία (completed_process_id).
- Μέσω της σχέσης με τον πίνακα completed_processes, τα βήματα αυτά σχετίζονται έμμεσα και με τον χρήστη που ολοκλήρωσε τη διαδικασία.

Κάθε completed file:

- Ανήκει σε ένα συγκεκριμένο ολοκληρωμένο βήμα μέσω του completed_step_id.
- Εντάσσεται σε μία συγκεκριμένη ολοκληρωμένη διαδικασία μέσω του completed_process_id.
- Συνδέεται με τον χρήστη που το ανήρτησε (user_id).

Κάθε sample:

- Ανήκει σε ένα μόνο βήμα διαδικασίας (step_id).
- Μπορεί να προβληθεί από οποιονδήποτε χρήστη ακολουθεί το αντίστοιχο βήμα.
- Χρησιμοποιείται ως τεκμηρίωση για την ορθή συμπλήρωση/εκτέλεση του βήματος.

Κάθε sample_version/upload:

- Συνδέεται με ένα συγκεκριμένο βήμα μέσω του step_id.
- Ανήκει σε μία τρέχουσα διαδικασία (running_process_id).
- Έχει αναρτηθεί από έναν χρήστη (user_id).

Κάθε εγγραφή στον πίνακα process_information:

- Συνδέεται με μια συγκεκριμένη διαδικασία μέσω του process_id.
- Αποτελεί ανεξάρτητη ενότητα πληροφορίας (section) εντός της διαδικασίας.

Κάθε calendar event:

- σχετίζεται με έναν και μόνο χρήστη (user_id), ο οποίος έχει τη δυνατότητα να το δει, να το δημιουργήσει ή να το διαγράψει μέσω της διεπαφής ημερολογίου της εφαρμογής.

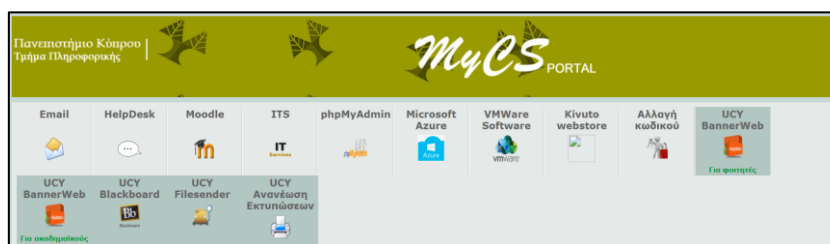
Κεφάλαιο 3

Χρήση Συστήματος

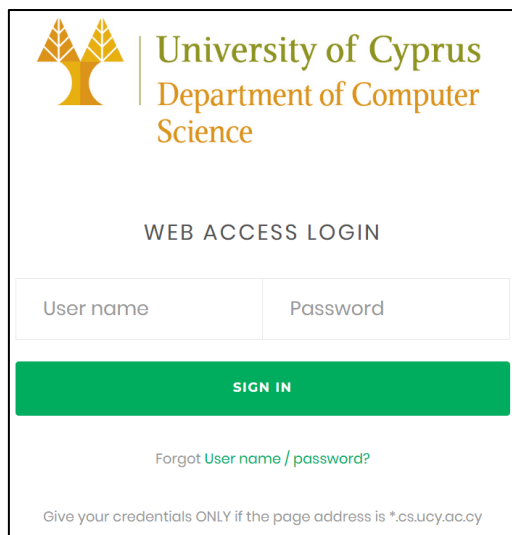
3.1 Σύνδεση	38
3.2 Αρχική Σελίδα	40
3.3 Όλες οι Διαδικασίες	42
3.4 Προσωπικές Διαδικασίες	45
3.4.1 Έναρξη Νέας Διαδικασίας	46
3.4.2 Τρέχουσες Διαδικασίες	47
3.4.3 Συμπληρωμένες Διαδικασίες	51
3.5 Ημερολόγιο	53

3.1 Σύνδεση

Η πρόσβαση στο σύστημα καταγραφής διαδικασιών πραγματοποιείται μέσω ενός συνδέσμου που θα ενσωματωθεί στο Portal του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου (<https://portal.cs.ucy.ac.cy/>) (Εικόνα 3). Ο σύνδεσμος θα οδηγεί τον χρήστη στην κεντρική σελίδα της εφαρμογής, η οποία προστατεύεται από μηχανισμό ελέγχου ταυτότητας βασισμένο στην υποδομή LDAP/Active Directory του Πανεπιστημίου Κύπρου (Εικόνα 3).



Εικόνα 3 – Σελίδα portal.cs.ucy.ac.cy



The image shows a web login form for the University of Cyprus Department of Computer Science. At the top left is the university's logo, a stylized tree. To its right, the text 'University of Cyprus' is in a large, bold, dark blue font, and 'Department of Computer Science' is in a smaller, bold, dark blue font below it. In the center, the text 'WEB ACCESS LOGIN' is in a bold, dark blue font. Below this, there are two input fields: 'User name' and 'Password', both with light blue borders. Underneath these fields is a large, solid blue button with the text 'SIGN IN' in white, bold, uppercase letters. Below the button, there is a link that says 'Forgot User name / password?' in a smaller, dark blue font. At the very bottom, in a small, dark blue font, it says 'Give your credentials ONLY if the page address is *.cs.ucy.ac.cy'.

Εικόνα 4 – Συστήμα Login

Η διαδικασία αυθεντικοποίησης δεν πραγματοποιείται τοπικά εντός της εφαρμογής. Αντιθέτως, οι χρήστες συνδέονται χρησιμοποιώντας τους ιδρυματικούς λογαριασμούς που διαχειρίζεται το Active Directory του Πανεπιστημίου. Μετά από επιτυχή σύνδεση, ο web server του πανεπιστημίου δημιουργεί session για τον χρήστη και ορίζει μεταβλητές στο global αντικείμενο `$_SESSION`, όπως οι παρακάτω:

- `$_SESSION['logged'] = true` — δείχνει ότι ο χρήστης είναι συνδεδεμένος.
- `$_SESSION['role']` — δηλώνει τον ρόλο του χρήστη στο Πανεπιστήμιο.
- `$_SESSION['name']` — περιέχει το όνομα χρήστη ή username (ιδανικά μοναδικό αναγνωριστικό).

Η εφαρμογή πραγματοποιεί τον παρακάτω έλεγχο πριν επιτρέψει την πρόσβαση:

1. Ελέγχει αν ο χρήστης είναι όντως συνδεδεμένος (`$_SESSION['logged'] == true`).
2. Επαληθεύει αν έχει τον απαιτούμενο ρόλο (`$_SESSION['role'] == "Required Role"`), καθώς η εφαρμογή προορίζεται αποκλειστικά για γραμματειακά μέλη του Τμήματος.
3. Διενεργεί και δεύτερο επίπεδο ελέγχου έναντι της τοπικής βάσης δεδομένων, όπου ελέγχει αν το όνομα χρήστη (`$_SESSION['name']`) υπάρχει στον πίνακα users της εφαρμογής. Η προσέγγιση αυτή διασφαλίζει ότι ακόμα και αν κάποιος μη εξουσιοδοτημένος χρήστης διαθέτει ιδρυματικό λογαριασμό με ρόλο "Required Role", δεν θα αποκτήσει πρόσβαση εφόσον δεν είναι καταχωρημένος στην εφαρμογή.

Αν οποιοδήποτε από τα παραπάνω κριτήρια δεν πληρούται, ο χρήστης ανακατευθύνεται αυτόματα σε σελίδα απόρριψης ή ειδοποιείται σχετικά με την έλλειψη δικαιωμάτων.

Η προσέγγιση αυτή εξασφαλίζει αυστηρό και ελεγχόμενο μηχανισμό πρόσβασης, περιορίζοντας τη χρήση της εφαρμογής αποκλειστικά στα εξουσιοδοτημένα άτομα του Τμήματος Πληροφορικής. Παράλληλα, αξιοποιείται η υπάρχουσα υποδομή ασφάλειας και διαχείρισης ταυτοτήτων του Πανεπιστημίου, αποφεύγοντας την ανάγκη διαχείρισης κωδικών εντός της εφαρμογής.

3.2 Αρχική Σελίδα

Η αρχική σελίδα του Συστήματος Καταγραφής Διαδικασιών (Εικόνα 5) αποτελεί το σημείο εκκίνησης για την πλοήγηση του χρήστη στις βασικές ενότητες της εφαρμογής. Ο σχεδιασμός της σελίδας εστιάζει στην απλότητα, την άμεση πληροφόρηση, και τη φιλικότητα προς τον χρήστη, με στόχο να παρέχει συνοπτική εικόνα της τρέχουσας κατάστασης και να προσφέρει άμεση πρόσβαση σε σημαντικές λειτουργίες.

Διάταξη και Περιεχόμενο

Η αρχική σελίδα περιλαμβάνει τρία βασικά παράθυρα/ενότητες:

1. Προεπισκόπηση Ημερολογίου

Στο αριστερό τμήμα εμφανίζεται το προσωπικό ημερολόγιο του χρήστη για τον τρέχοντα μήνα, με επισήμανση της τρέχουσας ημερομηνίας. Πατώντας πάνω στο παράθυρο, ο χρήστης μεταφέρεται στη σελίδα Ημερολόγιο, όπου μπορεί να δημιουργήσει ή να διαχειριστεί προσωπικά συμβάντα.

2. Όλες οι Διαδικασίες

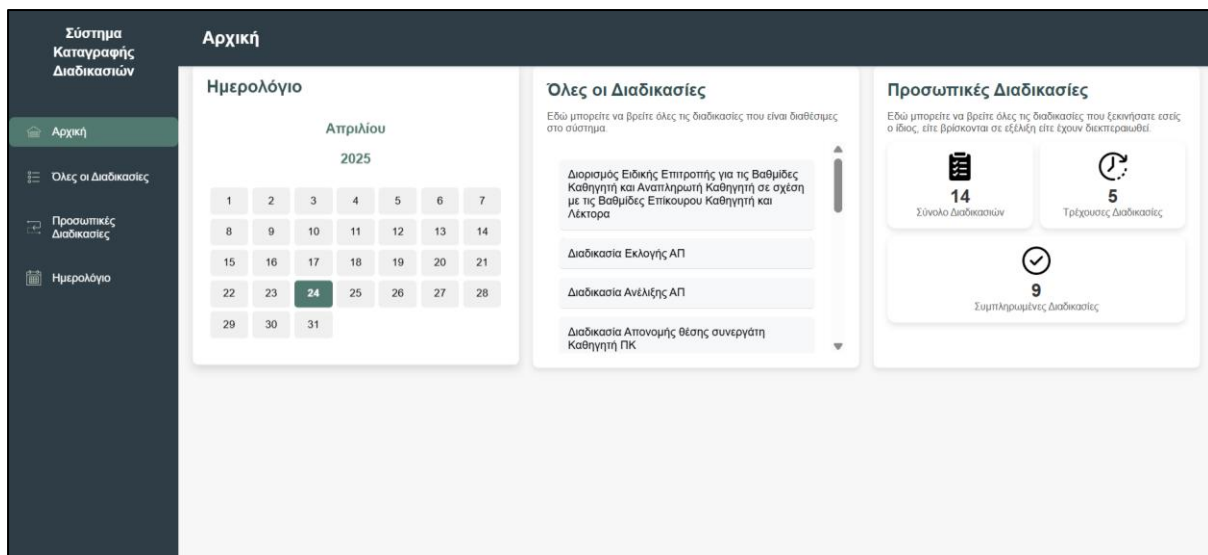
Το μεσαίο τμήμα λειτουργεί ως σύντομος σύνδεσμος προς την ενότητα Όλες οι Διαδικασίες, στην οποία παρουσιάζονται όλες οι διαθέσιμες διαδικασίες του Τμήματος. Ο χρήστης μπορεί να διαβάσει λεπτομέρειες και να ξεκινήσει οποιαδήποτε από αυτές.

3. Προσωπικές Διαδικασίες

Το δεξί τμήμα προβάλλει βασικά στατιστικά σχετικά με τις τρέχουσες και τις συμπληρωμένες διαδικασίες του χρήστη:

- Συνολικές διαδικασίες που ξεκίνησε
- Πόσες βρίσκονται σε εξέλιξη
- Πόσες έχουν ολοκληρωθεί

Λειτουργεί ως σύντομος σύνδεσμος προς την ενότητα Προσωπικές Διαδικασίες, κατά την οποία ο χρήστης μπορεί να μεταβεί για περαιτέρω διαχείριση.



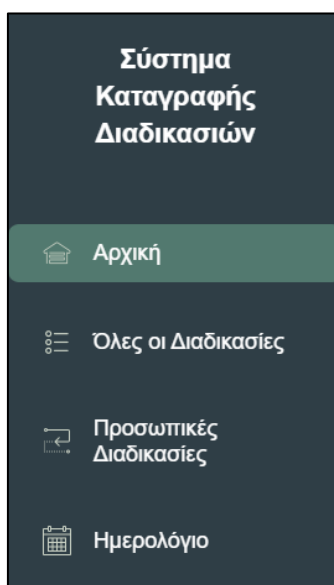
Εικόνα 5 – Αρχική Σελίδα

Μενού Πλοήγησης (Sidebar)

Στην αριστερή πλευρά της σελίδας υπάρχει σταθερό μενού πλοήγησης (Sidebar) (Εικόνα 6), το οποίο εμφανίζεται σε κάθε σελίδα της εφαρμογής. Παρέχει άμεση πρόσβαση στις βασικές ενότητες του συστήματος:

- Αρχική Σελίδα
- Όλες οι Διαδικασίες
- Προσωπικές Διαδικασίες
- Ημερολόγιο

Τα εικονίδια στο sidebar χρησιμοποιούνται για να διευκολύνουν την αναγνώριση κάθε ενότητας και να προσδώσουν φιλικό χαρακτήρα στη διεπαφή.



Εικόνα 6 – Μενού Πλοήγησης (Sidebar)

Κεφαλίδα Εφαρμογής (Header)

Σε κάθε σελίδα του συστήματος εμφανίζεται σταθερά μια κεφαλίδα (header) στο πάνω μέρος της οθόνης (Εικόνα 7), η οποία ενισχύει τη συνοχή της εμπειρίας πλοήγησης. Ο ρόλος της κεφαλίδας είναι:

- Η εμφάνιση του τίτλου της τρέχουσας σελίδας
- Η διατήρηση σταθερής ταυτότητας του συστήματος
- Η ενίσχυση της αισθητικής και λειτουργικής ομοιομορφίας σε όλες τις ενότητες

Η κεφαλίδα συμβάλλει στη γρήγορη αναγνώριση του περιεχομένου κάθε σελίδας και καθιστά την πλοήγηση πιο οργανωμένη και ευχάριστη.



Εικόνα 7 – Κεφαλίδα Εφαρμογής (Header)

3.3 Όλες οι Διαδικασίες

Η ενότητα Όλες οι Διαδικασίες παρέχει μια πλήρη και δομημένη επισκόπηση όλων των διαθέσιμων διαδικασιών που προσφέρει το Τμήμα μέσω του Συστήματος Καταγραφής Διαδικασιών (Εικόνα 8). Ο χρήστης έχει τη δυνατότητα να περιηγηθεί και να ενημερωθεί για το περιεχόμενο κάθε διαδικασίας, χωρίς να μπορεί να την επεξεργαστεί από αυτή την οθόνη.

Διάταξη και Λειτουργικότητα

Κάθε διαδικασία εμφανίζεται ως κάρτα με τον τίτλο της, και με ένα κλικ πάνω της, προβάλλονται αναλυτικά όλες οι σχετικές πληροφορίες, όπως:

- Ο σκοπός της διαδικασίας
- Οι εμπλεκόμενοι ρόλοι και φορείς
- Τα επιμέρους βήματα (εφόσον υπάρχουν)
- Οδηγίες, σχόλια και παρατηρήσεις
- Επισυναπτόμενα έγγραφα ή παραδείγματα προς αναφορά
- Η πηγή από την οποία πηγάζουν οι πληροφορίες της κάθε διαδικασίας (τόμος, νόμος)

Η προβολή αυτή λειτουργεί αποκλειστικά ενημερωτικά, και είναι προσβάσιμη από όλους τους χρήστες του συστήματος.

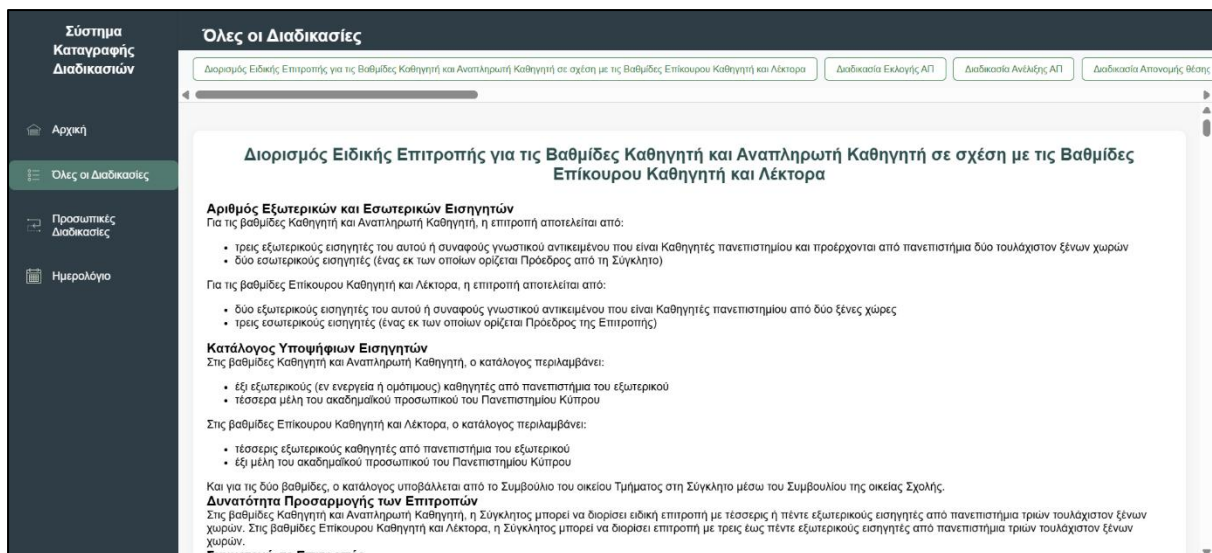
Στόχος της Ενότητας

Ο βασικός στόχος της ενότητας είναι:

- Η παροχή διαφάνειας ως προς τη φύση και το περιεχόμενο των διαδικασιών
- Η υποβοήθηση του χρήστη στην επιλογή της κατάλληλης διαδικασίας που επιθυμεί να ξεκινήσει αργότερα από την ενότητα Προσωπικές Διαδικασίες
- Η κεντρική διαχείριση γνώσης αναφορικά με τις τυπικές και άτυπες διαδικασίες του Τμήματος

Χαρακτηριστικά

- Οι διαδικασίες είναι ταξινομημένες κατά τρόπο απλό και ευδιάκριτο
- Η πλοήγηση είναι γρήγορη, με δυνατότητα προβολής πολλών διαδικασιών σε μία σελίδα
- Η προβολή είναι ενιαία για όλους τους χρήστες, χωρίς διαφοροποίηση ρόλων



Εικόνα 8 – Όλες οι Διαδικασίες

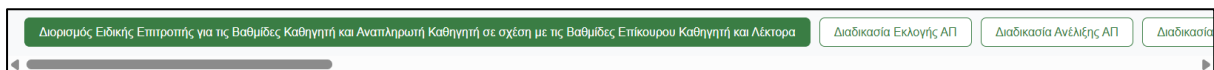
Navigation Bar – Εύκολη Περιήγηση ανάμεσα στις Διαδικασίες

Στο πάνω μέρος της ενότητας «Όλες οι Διαδικασίες» ενσωματώνεται μια οριζόντια navigation bar (Εικόνα 9), η οποία παρέχει στον χρήστη τη δυνατότητα να περιηγηθεί γρήγορα και αποτελεσματικά ανάμεσα στις διαθέσιμες διαδικασίες του συστήματος.

Η μπάρα πλοήγησης περιλαμβάνει κουμπιά με τον τίτλο κάθε διαδικασίας, ενώ:

- Το κουμπί της επιλεγμένης διαδικασίας εμφανίζεται με πράσινο χρώμα ώστε να υποδεικνύει οπτικά στον χρήστη ποια διαδικασία είναι αυτή τη στιγμή ενεργή (active view).
- Τα υπόλοιπα κουμπιά εμφανίζονται με λευκό φόντο και πράσινο περίγραμμα και παρέχουν άμεση πρόσβαση με ένα κλικ στην αντίστοιχη διαδικασία, η οποία αντικαθιστά τη μέχρι τώρα προβολή.
- Η μπάρα είναι οριζοντίως scrollable, υποστηρίζοντας τον χρήστη ακόμα και όταν υπάρχουν πολλές διαδικασίες.

Αυτός ο σχεδιασμός ενισχύει την ευχρηστία και τη γρήγορη εναλλαγή περιεχομένου, χωρίς την ανάγκη μετάβασης σε άλλη σελίδα ή φόρτωσης του συστήματος, καθιστώντας τη συνολική εμπειρία του χρήστη πιο ευέλικτη και αποτελεσματική.



Εικόνα 9 – Στιγμιότυπο της navigation bar διαδικασιών

Μορφή Παρουσίασης Διαδικασιών

Κάθε διαδικασία που παρουσιάζεται στην ενότητα Όλες οι Διαδικασίες ακολουθεί μια ομοιογενή και επεξηγηματική δομή (Εικόνα 10), η οποία επιτρέπει στον χρήστη να αντιληφθεί άμεσα τα στάδια που την απαρτίζουν.

Αριθμημένη Παρουσίαση Βημάτων

Κάθε επιμέρους βήμα αριθμείται αυτόματα (1, 2, 3, ...) και συνοδεύεται από τίτλο, περιγραφή και ενδεχομένως περαιτέρω υποενότητες ή συνδέσμους. Αυτή η αρίθμηση υποστηρίζει τη διαδοχική κατανόηση της ροής της διαδικασίας.

Κουμπί “More Details” / “Less Details”

Σε περιπτώσεις όπου υπάρχουν επιπρόσθετες λεπτομέρειες για ένα βήμα, αυτές αποκρύπτονται αρχικά για λόγους ευχρηστίας. Ο χρήστης μπορεί να τις εμφανίσει επιλέγοντας το κουμπί “More Details”, ενώ αντίστροφα έχει τη δυνατότητα να τις αποκρύψει με το “Less Details”.

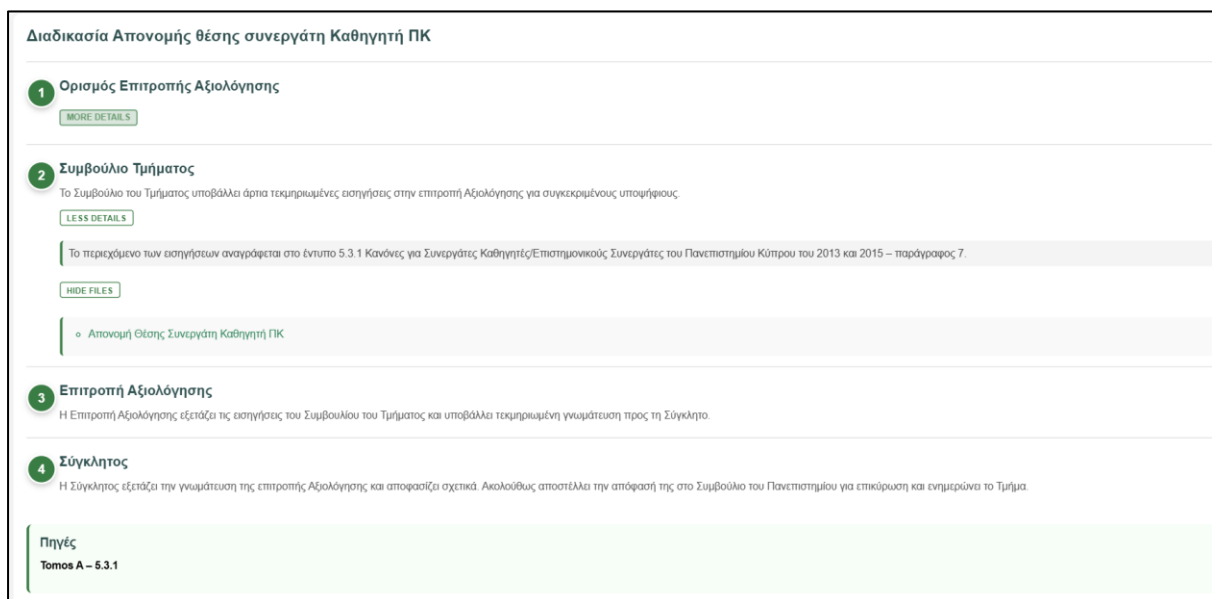
Η δυνατότητα αυτή εξασφαλίζει καθαρότητα στην οθόνη και περιορίζει την πληροφοριακή υπερφόρτωση.

Προβολή Αρχείων – “View Files” / “Hide Files”

Ορισμένα βήματα περιλαμβάνουν συνημμένα αρχεία (π.χ. πρότυπα, οδηγίες, έγγραφα αναφοράς), τα οποία προβάλλονται με το κουμπί “View Files”. Οι χρήστες μπορούν να τα κατεβάσουν τοπικά στον υπολογιστή τους.

Αναφορά σε Πηγές

Σε κάποιες περιπτώσεις όπου ένα βήμα βασίζεται σε θεσμικά ή κανονιστικά πλαίσια, εμφανίζεται στο τέλος της διαδικασίας πεδίο “Πηγές”, όπου καταγράφονται παραπομπές (π.χ. “Tomos A – 5.3.1”). Αυτό ενισχύει τη διαφάνεια και την αξιοπιστία της πληροφορίας, καθώς ο χρήστης έχει σαφή εικόνα για το θεσμικό υπόβαθρο κάθε ενέργειας.

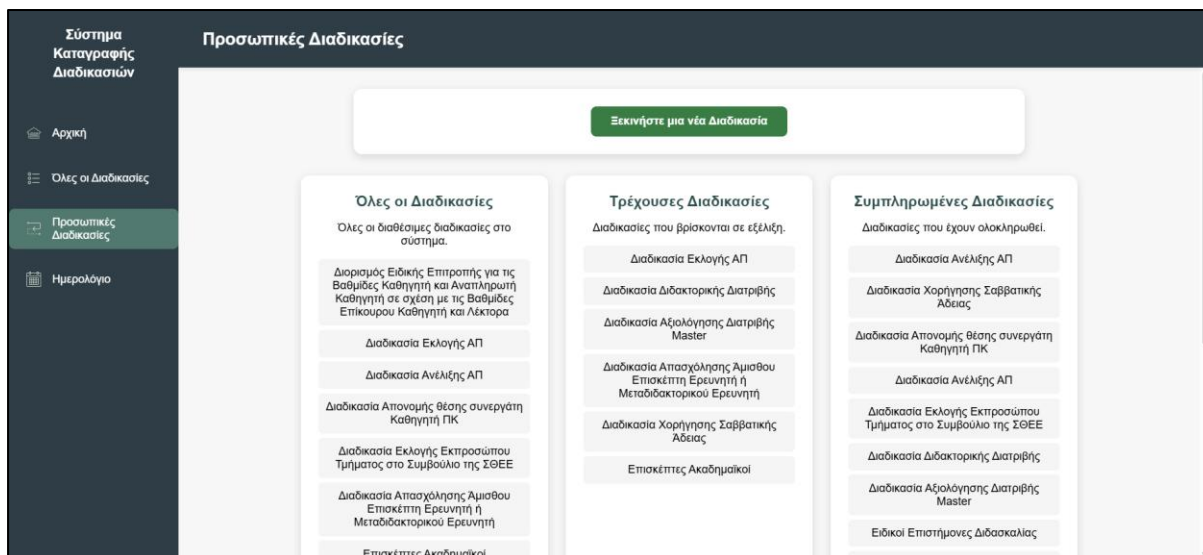


Εικόνα 10 – Στιγμιότυπο Παρουσίασης Διαδικασιών

3.4 Προσωπικές Διαδικασίες

Η ενότητα Προσωπικές Διαδικασίες αποτελεί τον βασικό πυρήνα αλληλεπίδρασης του χρήστη με το σύστημα (Εικόνα 11). Από εδώ, ο χρήστης μπορεί να:

- να μεταβεί στην ενότητα Όλες οι Διαδικασίες,
- ξεκινήσει μια νέα διαδικασία,
- δει όσες έχει ξεκινήσει και βρίσκονται σε εξέλιξη,
- ή να περιηγηθεί σε όσες έχει ολοκληρώσει.



Εικόνα 11 – Στιγμιότυπο Προσωπικές Διαδικασίες

3.4.1 Εκκίνηση Νέας Διαδικασίας

Κατά την επιλογή του κουμπιού "Ξεκινήστε μια νέα Διαδικασία" (Εικόνα 12), εμφανίζεται αναδυόμενο παράθυρο με λίστα όλων των διαθέσιμων διαδικασιών. Ο χρήστης επιλέγει τη διαδικασία που επιθυμεί, εισάγει ένα όνομα (ή αποδέχεται το προκαθορισμένο), και πατώντας "Εκκίνηση" ξεκινάει η διαδικασία.

Η διαδικασία μεταφέρεται αυτόματα στο περιβάλλον Τρέχουσες Διαδικασίες, αποκτώντας προσωπικό χαρακτήρα και γινόμενη επεξεργάσιμη αποκλειστικά από τον χρήστη.

Ακύρωση

Επιλέξτε την Διαδικασία που επιθυμείτε να ξεκινήσετε:

Διαδικασία Εκλογής ΑΠ

Διαδικασία Ανέλιξης ΑΠ

Διαδικασία Απονομής θέσης συνεργάτη Καθηγητή ΠΚ

Διαδικασία Εκλογής Εκπροσώπου Τμήματος στο Συμβούλιο της ΣΘΕΕ

Διαδικασία Απασχόλησης Άμισθου Επισκέπτη Ερευνητή ή Μεταδιδακτορικού Ερευνητή

Επισκέπτες Ακαδημαϊκοί

Ειδικοί Επιστήμονες Διδασκαλίας

Διαδικασία Αξιολόγησης Διατριβής Master

Διαδικασία Περιεκτικής Εξέτασης Διδακτορικού Φοιτητή

Διαδικασία Ερευνητικής Πρότασης Διδακτορικού Φοιτητή

Διαδικασία Διδακτορικής Διατριβής

Διαδικασία Προκήρυξης και Προσφοράς θέσεων στα Μεταπτυχιακά προγράμματα

Διαδικασία Χορήγησης Σαββατικής Άδειας

Όνομα Διαδικασίας:

Διαδικασία Εκλογής ΑΠ

Εκκίνηση

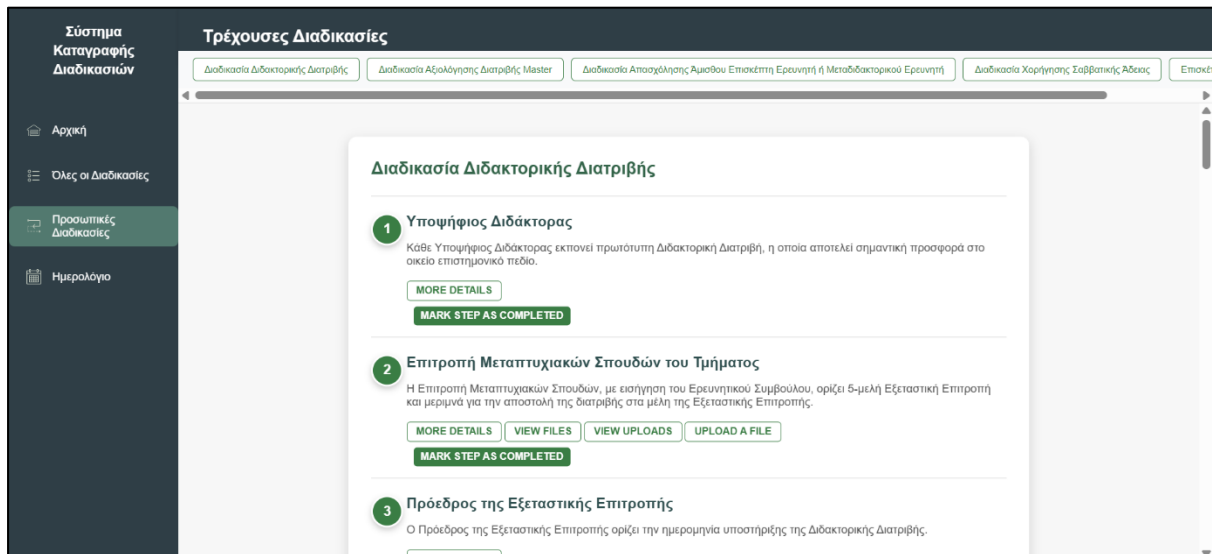
Εικόνα 12 – Στιγμιότυπο Εκκίνηση Νέας Διαδικασίας

3.4.2 Τρέχουσες Διαδικασίες

Το τμήμα Τρέχουσες Διαδικασίες, όπως αυτό διακρίνεται στην Εικόνα 13, προβάλλει όλες τις ενεργές διαδικασίες που έχει ξεκινήσει ο χρήστης. Κάθε διαδικασία περιλαμβάνει όλα της τα βήματα, μαζί με δυνατότητα:

- Προβολής επιπλέον πληροφοριών (κουμπί More Details),
- Εμφάνισης και λήψης αρχείων (κουμπί View Files / View Uploads),
- Ανάρτησης αρχείων (κουμπί Upload a File),

- Αποστολής email στον επόμενο εμπλεκόμενο (κουμπί Στείλε Email),
- Ορισμού του βήματος ως ολοκληρωμένο (Mark Step as Completed),
- Ή αντίστροφα, ως μη-ολοκληρωμένο (Unmark Step from Completed).



Εικόνα 13 – Στιγμιότυπο Τρέχουσες Διαδικασίες

Ανάλυση Οθόνης Τρέχουσες Διαδικασίες

Η εικόνα παρουσιάζει μια χαρακτηριστική περίπτωση ενεργής διαδικασίας, συγκεκριμένα τη Διαδικασία Διδακτορικής Διατριβής, όπως εμφανίζεται στον χρήστη στην ενότητα Τρέχουσες Διαδικασίες.

Η διαδικασία αποτελείται από διαδοχικά βήματα (steps), τα οποία εμφανίζονται με αριθμημένη σειρά, παρέχοντας έτσι μια καθαρή αίσθηση προόδου.

Στοιχεία κάθε Βήματος

Κάθε επιμέρους βήμα περιλαμβάνει:

Τίτλο βήματος – για παράδειγμα: Υποψήφιος Διδάκτορας, Επιτροπή Μεταπτυχιακών Σπουδών του Τμήματος κ.λπ.

Περιγραφή – συνοπτική επεξήγηση του ρόλου του βήματος, ώστε ο χρήστης να γνωρίζει το περιεχόμενό του.

Κουμπίά Δράσεων:

- More Details: Εμφανίζει τυχόν επιπλέον πληροφορίες για το συγκεκριμένο βήμα, επεκτείνοντας τη βασική περιγραφή.
- View Files / View Uploads: Δίνουν πρόσβαση στα υπάρχοντα αρχεία που συνοδεύουν το βήμα (π.χ., δείγματα, έντυπα).
- Upload a File: Επιτρέπει στον χρήστη να ανεβάσει τα τροποποιημένα του έγγραφα ή συμπληρωμένα έντυπα (Εικόνα 14).

5 Πρόεδρος του Τμήματος

Ο Πρόεδρος του Τμήματος διαπιστώνει τη νομιμότητα της όλης διαδικασίας και υποβάλλει την εισήγηση προς τη Σχολή Μεταπτυχιακής Εκπαίδευσης για τα περαιτέρω.

[VIEW FILES](#) [CANCEL UPLOAD](#)

Drag & Drop a file here
or click to upload

[ΣΤΕΪΛΕ EMAIL](#) ☒ Χωρίς συνημμένα αρχεία ☐ Με συνημμένα αρχεία

[MARK STEP AS COMPLETED](#)

Εικόνα 14 – Στιγμιότυπο Ανάρτησης Αρχείου

- **Mark Step as Completed:** Όταν ολοκληρωθεί το βήμα, ο χρήστης μπορεί να το μαρκάρει ως ολοκληρωμένο. Το κουμπί γίνεται πράσινο, επιβεβαιώνοντας την ενέργεια. Αντίστροφα, μπορεί να το μαρκάρει και ως εκκρεμή “Unmark Step From Completed” (Εικόνα 15).

1 Υποψήφιος Διδάκτορας

Κάθε Υποψήφιος Διδάκτορας εκπονεί πρωτότυπη Διδακτορική Διατριβή, η οποία αποτελεί σημαντική προσφορά στο οικείο επιστημονικό πεδίο.

[MORE DETAILS](#)

[UNMARK STEP FROM COMPLETED](#)

Εικόνα 15 – Στιγμιότυπο Ολοκληρωμένου Βήματος

- **Στείλε Email:** Αποστολής email στον επόμενο εμπλεκόμενο, με (Εικόνα 17) ή χωρίς (Εικόνα 18) συνημμένα τα ανεβασμένα αρχεία που τροποποιήθηκαν από τον χρήστη (Εικόνα 16).

5 Πρόεδρος του Τμήματος

Ο Πρόεδρος του Τμήματος διαπιστώνει τη νομιμότητα της όλης διαδικασίας και υποβάλλει την εισήγηση προς τη Σχολή Μεταπτυχιακής Εκπαίδευσης για τα περαιτέρω.

[HIDE FILES](#)

Εντυπο Υποβολής Δικαιολογητικών Απονομής Διδακτορικού Τίτλου - ENGLISH
Εντυπο Υποβολής Δικαιολογητικών Απονομής Διδακτορικού Τίτλου - ΕΛΛΗΝΙΚΑ

[HIDE UPLOADS](#)

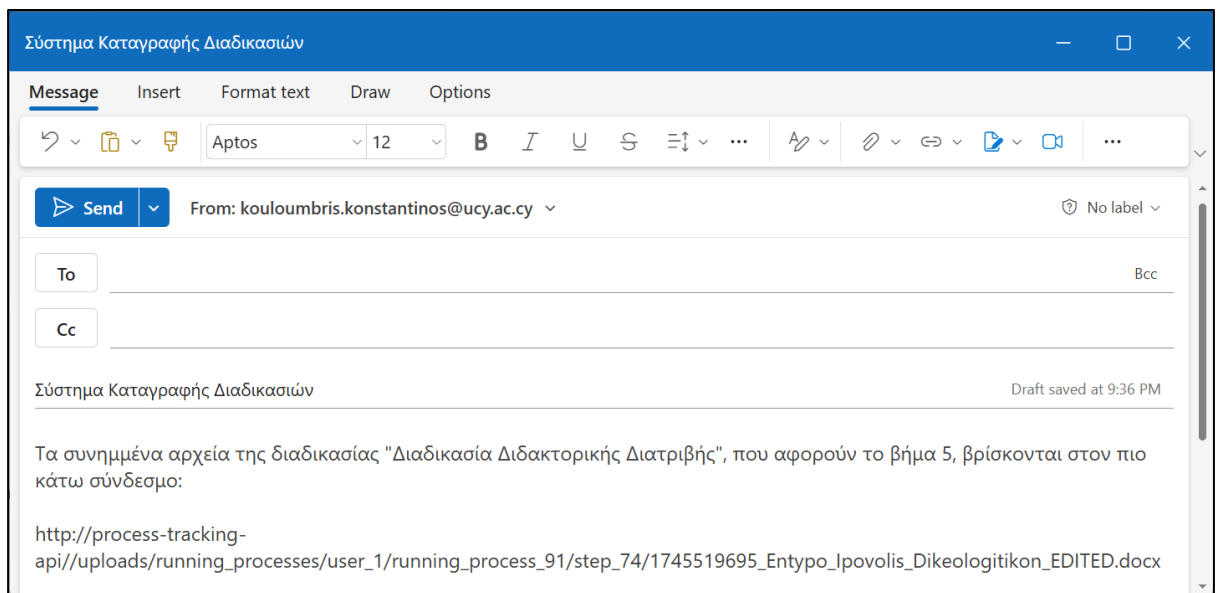
Entypo_Ipovolis_Dikeologitikon_EDITED.docx

[UPLOAD A FILE](#)

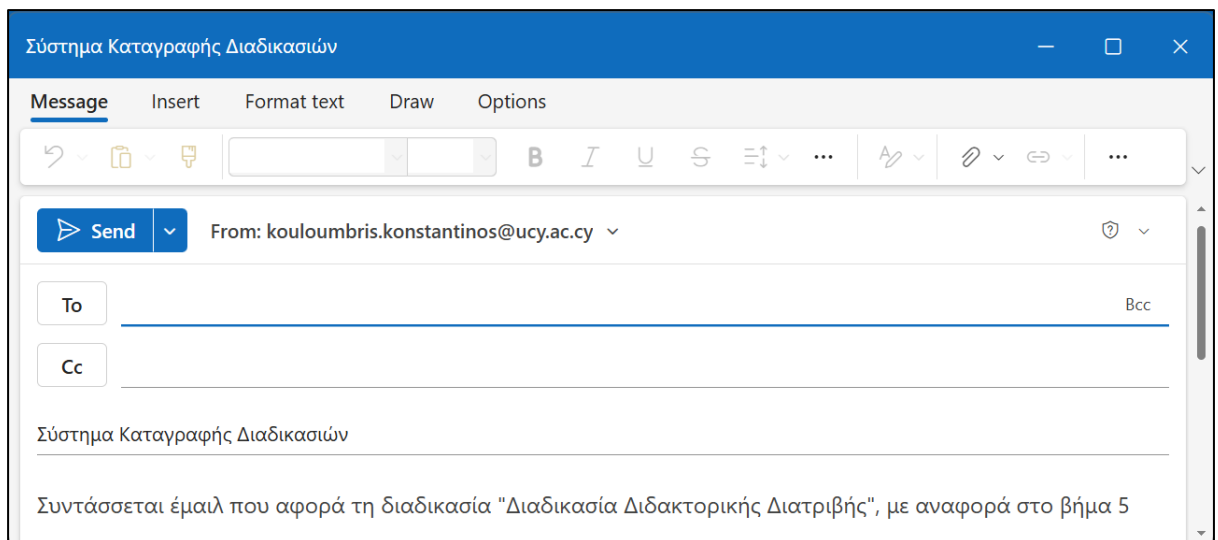
[ΣΤΕΪΛΕ EMAIL](#) ☒ Χωρίς συνημμένα αρχεία ☐ Με συνημμένα αρχεία

[MARK STEP AS COMPLETED](#)

Εικόνα 16 – Στιγμιότυπο Αποστολής Email



Εικόνα 17 – Στιγμιότυπο Αποστολής Εμάιλ με Συνημμένα Αρχεία



Εικόνα 18 – Στιγμιότυπο Αποστολής Εμάιλ χωρίς Συνημμένα Αρχεία

Οπτικά και Λειτουργικά Χαρακτηριστικά

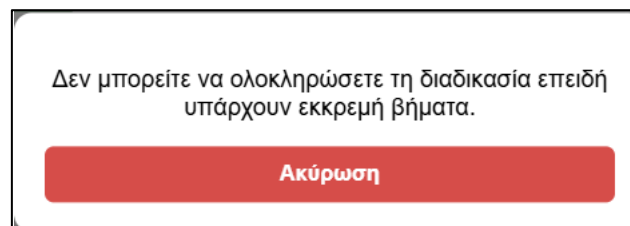
- Η διάταξη είναι κάθετη και διακριτή, με κάθε βήμα να περιβάλλεται από οπτικά όρια για σαφήνεια.
- Τα κουμπιά είναι χρωματικά ουδέτερα (λευκό με πράσινο περίγραμμα) ώστε να εστιάζει ο χρήστης στις λειτουργίες χωρίς να αποσπάται από χρώματα.
- Το κουμπί ολοκλήρωσης (Mark Step as Completed) ξεχωρίζει με έντονο πράσινο, προσελκύοντας την προσοχή και υποδηλώνοντας σημαντική ενέργεια.
- Η αρίθμηση των βημάτων υποστηρίζει την έννοια της προόδου και της σειράς.

Πλοήγηση

Στην κορυφή της σελίδας εμφανίζεται navigation bar με καρτέλες, ώστε ο χρήστης να μπορεί να μετακινηθεί γρήγορα μεταξύ των ενεργών διαδικασιών του. Η ενεργή διαδικασία είναι εμφανώς τονισμένη με πράσινο φόντο.

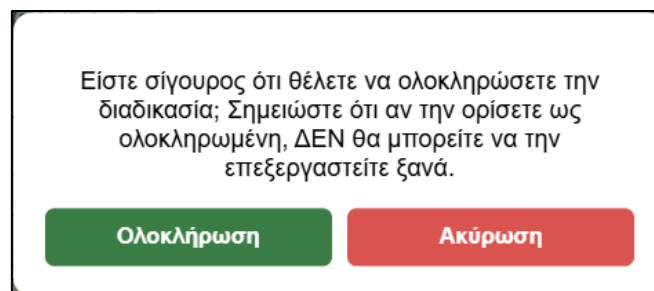
Ολοκλήρωση διαδικασίας

Ο χρήστης έχει τη δυνατότητα να ορίσει μία διαδικασία ως ολοκληρωμένη μόνο αν έχουν ολοκληρωθεί όλα τα βήματά της. Αν υπάρχουν εκκρεμή βήματα, εμφανίζεται σχετικό μήνυμα: "Δεν μπορείτε να ολοκληρώσετε τη διαδικασία επειδή υπάρχουν εκκρεμή βήματα." (Εικόνα 19).



Εικόνα 19 – Στιγμιότυπο Προειδοποίησης Εκκρεμότητας

Σε περίπτωση που όλα τα βήματα είναι συμπληρωμένα, η εφαρμογή εμφανίζει ένα παράθυρο επιβεβαίωσης πριν από την τελική ενέργεια: "Είστε σίγουρος ότι θέλετε να ολοκληρώσετε την διαδικασία; Σημειώστε ότι αν την ορίσετε ως ολοκληρωμένη, ΔΕΝ θα μπορείτε να την επεξεργαστείτε ξανά." (Εικόνα 20).



Εικόνα 20 – Στιγμιότυπο Προειδοποίησης Ολοκλήρωσης Διαδικασίας

Μετά την επιβεβαίωση, η διαδικασία μεταφέρεται αυτόματα στο περιβάλλον Συμπληρωμένες Διαδικασίες, καθίσταται πλέον μη επεξεργάσιμη και παραμένει προσβάσιμη μόνο για προβολή.

3.4.3 Συμπληρωμένες Διαδικασίες

Το υποσύστημα Συμπληρωμένες Διαδικασίες, όπως αυτό διακρίνεται στην Εικόνα 21, αποτελεί τον τελικό σταθμό κάθε διαδικασίας που έχει ολοκληρωθεί επιτυχώς από τον χρήστη.

Σε αυτό το περιβάλλον, ο χρήστης μπορεί να δει αναλυτικά όλες τις διαδικασίες που έχει τερματίσει, με πλήρη καταγραφή των βημάτων και των ενεργειών που έλαβαν χώρα κατά την εκτέλεσή τους.

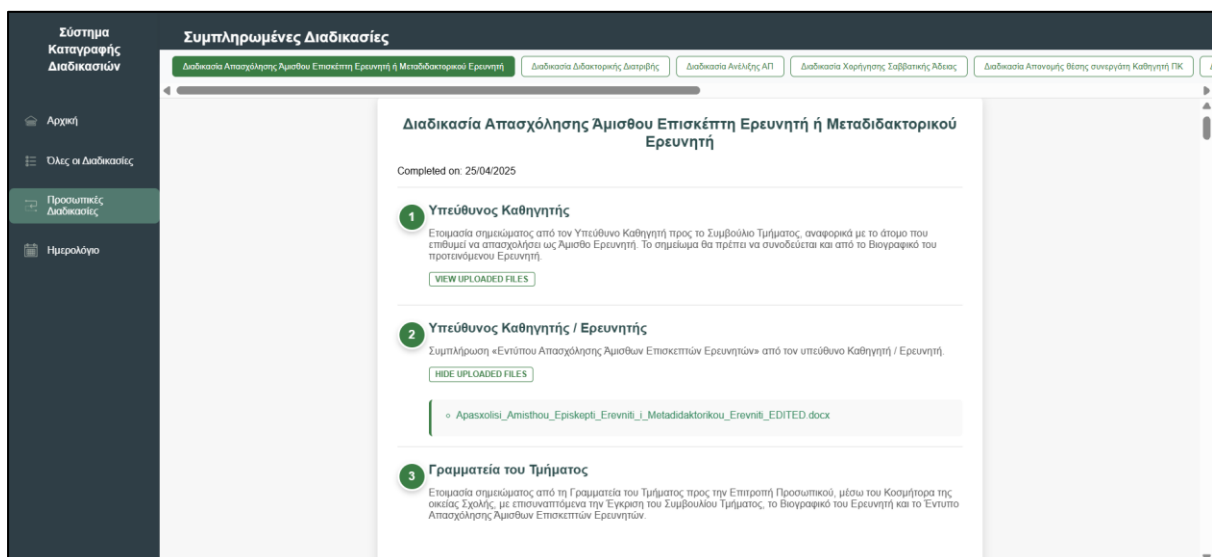
Λειτουργικότητα και Παρουσίαση

- Κάθε διαδικασία συνοδεύεται από ημερομηνία ολοκλήρωσης, που υποδεικνύει πότε μαρκαρίστηκε ως ολοκληρωμένη.
- Παρουσιάζονται όλα τα βήματα της διαδικασίας, μαζί με τις σχετικές περιγραφές, τις πληροφορίες, καθώς και τα αρχεία που ο ίδιος ο χρήστης ανάρτησε στη διάρκεια της διαδικασίας (τα αρχικά αρχεία/δείγματα δεν παρουσιάζονται πλέον).
- Ο χρήστης μπορεί να δει επιπλέον πληροφορίες κάθε βήματος πατώντας στο κουμπί More Details ή να αναπτύξει την ενότητα των αρχείων με View Uploaded Files.

Σημαντικές Παρατηρήσεις

- Δεν παρέχεται δυνατότητα επεξεργασίας ή αποστολής email σε οποιοδήποτε βήμα. Η κατάσταση είναι τελική και μη αναστρέψιμη.
- Οι διαθέσιμες ενέργειες περιορίζονται μόνο στην προβολή των δεδομένων και στην λήψη των τελικών αρχείων.

Η ενότητα αυτή επιτρέπει στους χρήστες να κρατούν αρχείο των ολοκληρωμένων διαδικασιών τους και να ανατρέχουν σε αυτές για αναφορά ή αποθήκευση αρχείων όταν χρειάζεται.



Εικόνα 21 – Στιγμιότυπο Συμπληρωμένες Διαδικασίες

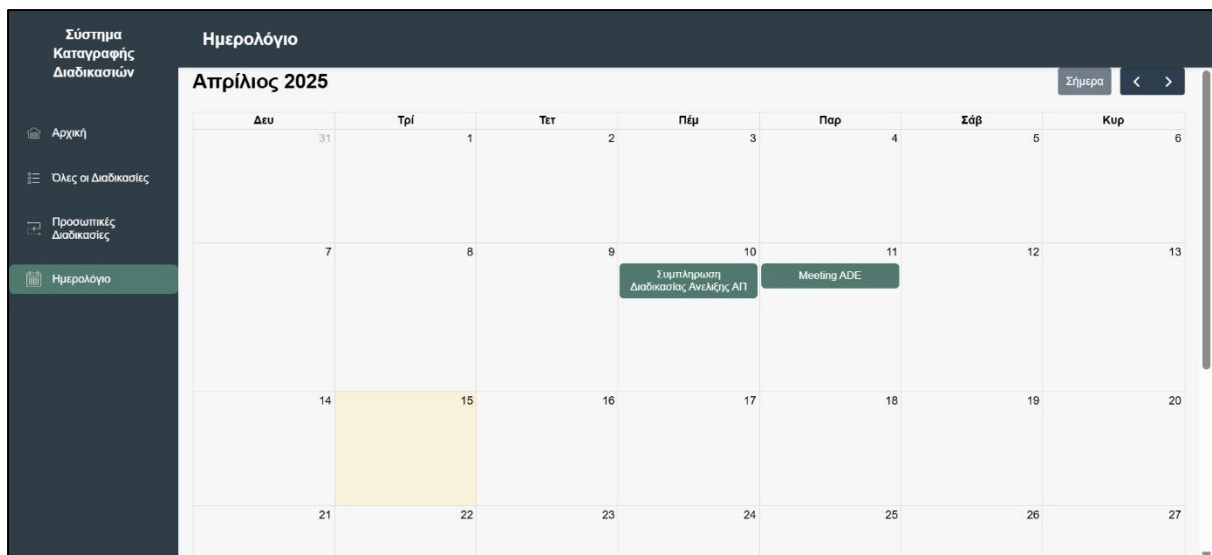
Πλοήγηση

Στην κορυφή της σελίδας εμφανίζεται navigation bar με καρτέλες, ώστε ο χρήστης να μπορεί να μετακινηθεί γρήγορα μεταξύ των συμπληρωμένων διαδικασιών του. Η ενεργή διαδικασία είναι εμφανώς τονισμένη με πράσινο φόντο.

3.5 Ημερολόγιο

Το υποσύστημα Ημερολόγιο (Εικόνα 22) αποτελεί ένα βοηθητικό εργαλείο οργάνωσης και διαχείρισης των προσωπικών υποχρεώσεων του χρήστη. Η λειτουργία του έχει καθαρά προσωπικό χαρακτήρα, και όλα τα γεγονότα που καταχωρούνται είναι προσβάσιμα αποκλειστικά από τον ίδιο τον χρήστη που τα δημιουργήσε.

Η κύρια σελίδα του Ημερολογίου αποτελείται από ένα πλήρως διαδραστικό ημερολόγιο, που απεικονίζει τον τρέχοντα μήνα και τα γεγονότα που έχουν καταχωρηθεί. Τα γεγονότα αυτά εμφανίζονται με σαφή και συνοπτικό τρόπο σε μορφή καρτέλας πάνω στην αντίστοιχη ημέρα.



Εικόνα 22 – Στιγμιότυπο Ημερολογίου

Προσθήκη Νέου Γεγονότος

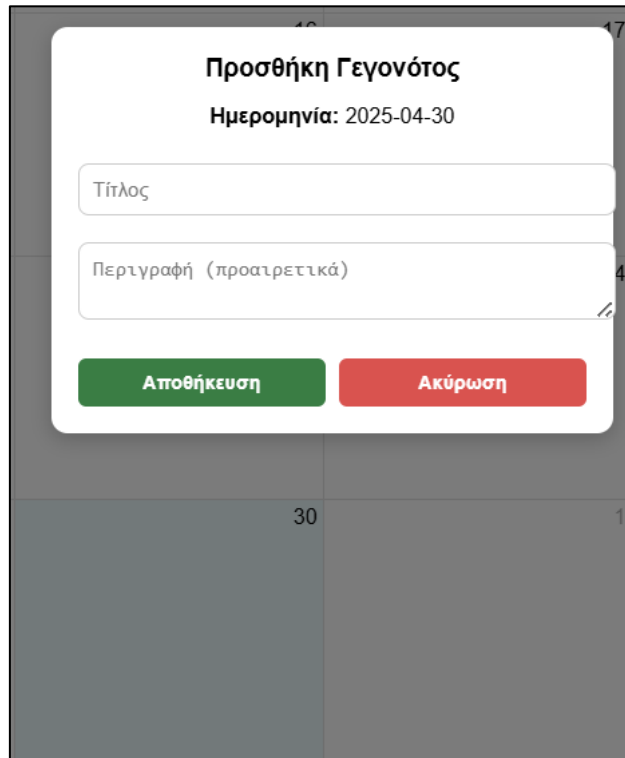
Ο χρήστης έχει τη δυνατότητα να προσθέσει ένα νέο προσωπικό γεγονός, κάνοντας απλώς κλικ επάνω στην επιθυμητή ημερομηνία στο ημερολόγιο. Με αυτή την ενέργεια, εμφανίζεται ένα αναδυόμενο παράθυρο με τίτλο "Προσθήκη Γεγονότος" (Εικόνα 23), το οποίο περιλαμβάνει τα εξής πεδία:

- Ημερομηνία: Εμφανίζεται αυτόματα με βάση την επιλεγμένη ημέρα.
- Τίτλος: Υποχρεωτικό πεδίο στο οποίο καταχωρείται η ονομασία του γεγονότος.
- Περιγραφή (προαιρετικά): Χώρος για επιπλέον επεξηγήσεις ή λεπτομέρειες σχετικά με το γεγονός.

Ο χρήστης έχει δύο επιλογές δράσης:

- Αποθήκευση: Το γεγονός καταγράφεται και εμφανίζεται πλέον ορατό στο ημερολόγιο.
- Ακύρωση: Κλείνει το παράθυρο χωρίς να γίνει καμία ενέργεια.

Η απλότητα και η εργονομία της διεπαφής καθιστούν τη διαδικασία εύκολη και άμεση, ενισχύοντας τη χρηστικότητα του Ημερολογίου για προσωπική διαχείριση χρόνου.

The image shows a modal dialog box titled "Προσθήκη Γεγονότος" (Add Event) with a date of "2025-04-30". It contains two input fields: "Τίτλος" (Title) and "Περιγραφή (προαιρετικά)" (Description (optional)). At the bottom, there are two buttons: a green "Αποθήκευση" (Save) button and a red "Ακύρωση" (Cancel) button. The dialog is overlaid on a dark grey background that appears to be a calendar grid.

Εικόνα 23 – Στιγμιότυπο Προσθήκης Νέου Γεγονότος

Προβολή Γεγονότος

Ο χρήστης έχει τη δυνατότητα να επιλέξει οποιοδήποτε γεγονός στο ημερολόγιο, και θα εμφανιστεί ένα αναδυόμενο παράθυρο με τις λεπτομέρειες του γεγονότος (Εικόνα 24). Οι πληροφορίες που παρουσιάζονται περιλαμβάνουν:

- Ημερομηνία του συμβάντος
- Τίτλος/περιγραφή

Η απεικόνιση είναι απλή και λειτουργική, προσφέροντας δυνατότητα γρήγορης ενημέρωσης χωρίς πλοήγηση εκτός της σελίδας.

Προβολή Γεγονότος

Ημερομηνία: 2025-04-10

Τίτλος: Συμπλήρωση Διαδικασίας Ανελιξης ΑΠ

Κλείσιμο

Εικόνα 24 – Στιγμιότυπο Προβολή Γεγονότος

Χρήσεις και Οφέλη

Το Ημερολόγιο προσφέρει σημαντικά οφέλη στον χρήστη:

- Συγκεντρώνει και παρουσιάζει προσωπικά γεγονότα που σχετίζονται με διαδικασίες ή ανεξάρτητες δραστηριότητες.
- Επιτρέπει τη χρονική οργάνωση των ενεργειών του χρήστη.
- Παρέχει μια οπτική εποπτεία των ημερολογιακών υποχρεώσεων με διακριτό τρόπο.

Η χρήση του ημερολογίου δεν είναι υποχρεωτική, αλλά ενδείκνυται για όσους επιθυμούν να συντονίσουν αποτελεσματικότερα τις υποχρεώσεις τους εντός του Συστήματος Καταγραφής Διαδικασιών.

Κεφάλαιο 4

Υλοποίηση Συστήματος

4.1 Δομή Αρχείων	56
4.2 Βιβλιοθήκες	58
4.3 Παραδείγματα Υλοποίησης	59
4.3.1 Component Structure & State MANAGEMENT	60
4.3.2 Routing	62
4.3.3 Δυναμική Προβολή Δεδομένων	63
4.3.4 Calendar Integration (FullCalendar JS)	64
4.3.5 Επικοινωνία με Backend (axios)	66
4.3.6 Ειδοποιήσεις και Επιβεβαιώσεις (Toastify Notifications και Προσαρμοσμένα Modals)	68

4.1 Δομή Αρχείων

Η εφαρμογή αναπτύχθηκε ως ένα React-based Single Page Application (SPA), με στόχο την παροχή ενός δυναμικού και διαδραστικού περιβάλλοντος για την καταγραφή και παρακολούθηση διαδικασιών. Η δομή των αρχείων του έργου είναι οργανωμένη κατά τρόπο που ενισχύει τη συντηρησιμότητα, την επαναχρησιμοποίηση κώδικα και την καθαρότητα στην ανάπτυξη των επιμέρους λειτουργικών τμημάτων.

Η δομή των κύριων φακέλων του React frontend περιγράφεται αναλυτικά παρακάτω:

- `src/components/`: Περιέχει όλα τα επαναχρησιμοποιήσιμα components της εφαρμογής, τα οποία αντιπροσωπεύουν ανεξάρτητα γραφικά και λειτουργικά τμήματα του UI.

Ενδεικτικά περιλαμβάνει τα:

- `CalendarPreview/`: Προεπισκόπηση ημερολογίου.
- `Header/`, `Sidebar/`, `NavigationBar/`: Δομικά στοιχεία πλοήγησης.
- `ProcessCard/` και `StepCard.js`: Κάρτες παρουσίασης διαδικασιών και βημάτων.
- `StatsCard/`: Παρουσίαση στατιστικών στοιχείων.

- `src/pages/`: Περιέχει τις βασικές σελίδες της εφαρμογής, οι οποίες αντιστοιχούν σε διακριτά `views` του περιβάλλοντος χρήστη. Συγκεκριμένα:
 - `Dashboard/`: Κεντρική σελίδα χρήστη με στατιστικά και προεπισκοπήσεις.
 - `Calendar/`: Περιβάλλον προσθήκης και προβολής ημερολογιακών γεγονότων.
 - `AllProcesses/`: Σελίδα παρουσίασης όλων των διαθέσιμων διαδικασιών.
 - `MyProcesses/`: Διαχείριση προσωπικών διαδικασιών (τρέχουσες/ολοκληρωμένες).
- `src/routes/`: Περιλαμβάνει τον ορισμό των διαδρομών (routing) της εφαρμογής μέσω `React Router`. Το αρχείο `AppRoutes.js` καθορίζει τη ροή πλοήγησης μεταξύ των σελίδων.
- `src/assets/`: Περιλαμβάνει αρχεία πολυμέσων και εικονίδια (π.χ. `calendar-icon.png`, `processes-icon.png`), τα οποία χρησιμοποιούνται στο UI για βελτιωμένη οπτική εμπειρία.
- `src/styles/`: Περιέχει κοινά `CSS styles` για την εμφάνιση της εφαρμογής. Περιλαμβάνονται το `global.css` για καθολικό styling και επιμέρους αρχεία, όπως `ProcessCard.css`, για συγκεκριμένα `components`.
- Ριζικά αρχεία (`App.js`, `index.js`): Το `App.js` αποτελεί το κύριο `component` που περιβάλλει όλη την εφαρμογή και διαχειρίζεται τη δρομολόγηση και το `layout`. Το `index.js` είναι το `entry point` που εκκινεί την εφαρμογή `React` και πραγματοποιεί τη σύνδεση με το `root` στοιχείο του `DOM`.

Από πλευράς backend, αξιοποιήθηκε `PHP` με `custom routing` δομή, τοποθετημένη στον φάκελο `process-tracking-api`. Ο backend ακολουθεί `modular` προσέγγιση, με ξεχωριστούς φακέλους και αρχεία ως εξής:

- `routes/`: Ο πυρήνας του backend. Περιλαμβάνει διακριτά `endpoints` για διάφορες λειτουργίες όπως `start_process.php`, `steps.php`, `calendar_events.php`, `send_email.php` και `update_step_status.php`. Η αρχιτεκτονική του backend είναι `RESTful`, με κάθε `endpoint` να εξυπηρετεί συγκεκριμένο σκοπό.
- `config/`: Περιέχει το αρχείο `database.php` με τις ρυθμίσεις σύνδεσης με τη βάση δεδομένων.
- `uploads/`: Χώρος αποθήκευσης αρχείων που σχετίζονται με τα βήματα των διαδικασιών.
 - `original_samples/`: Φάκελοι κατά βήμα με ενδεικτικά έγγραφα.
 - `running_processes/user_X/`: Δομή ανά χρήστη και ανά διαδικασία με τα αρχεία που έχουν υποβληθεί ή τροποποιηθεί κατά την εξέλιξη των βημάτων.

Η δομή των αρχείων ακολουθεί τις καλές πρακτικές διαχωρισμού ευθυνών (separation of concerns), επιτρέποντας την ομαλή συνεργασία frontend–backend και διατηρώντας τον κώδικα ευανάγνωστο και επεκτάσιμο.

4.2 Βιβλιοθήκες

Κατά την ανάπτυξη της εφαρμογής αξιοποιήθηκε πληθώρα βιβλιοθηκών και πακέτων του οικοσυστήματος του React, προκειμένου να επιτευχθεί ταχύτατη ανάπτυξη, ευχρηστία και διαδραστικότητα στο τελικό αποτέλεσμα. Παρακάτω παρατίθενται οι βασικότερες εξαρτήσεις που ενσωματώθηκαν στην εφαρμογή, μαζί με τη λειτουργία που επιτελούν.

react-router-dom

Η βιβλιοθήκη react-router-dom χρησιμοποιείται για την πλοήγηση (routing) μεταξύ των σελίδων της εφαρμογής. Μέσω του BrowserRouter, Routes και Route, επιτυγχάνεται η διαχείριση διαδρομών χωρίς ανανέωση της σελίδας, εξασφαλίζοντας ένα smooth Single Page Application (SPA) περιβάλλον.

Χρήση:

- Πλοήγηση μεταξύ των σελίδων Dashboard, All Processes, My Processes, Calendar.
- Επαναχρησιμοποιήσιμος routing μηχανισμός μέσω AppRoutes.js.

fullcalendar

Η βιβλιοθήκη @fullcalendar/react [4] χρησιμοποιείται για την ενσωμάτωση διαδραστικού ημερολογίου, μέσω του οποίου ο χρήστης μπορεί να προσθέσει προσωπικά γεγονότα.

Χρήση:

- Απόδοση εβδομαδιαίας/μηνιαίας προβολής ημερολογίου.
- Δυναμική απεικόνιση γεγονότων με ημερομηνία.
- Αλληλεπίδραση μέσω clicks σε ημερομηνίες και γεγονότα.

sweetalert2

Η βιβλιοθήκη sweetalert2 [6] χρησιμοποιείται για την εμφάνιση ενημερωτικών παραθύρων (modals) με ελκυστική και responsive εμφάνιση.

Χρήση:

- Εμφάνιση alerts κατά την ολοκλήρωση ενός βήματος ή διαδικασίας.
- Επιβεβαιώσεις πριν την αποστολή email ή ανάρτηση αρχείου.
- Παρουσίαση επιτυχημένων ενεργειών με οπτική ανατροφοδότηση.

axios

Το axios αποτελεί HTTP client για αποστολή αιτημάτων στον backend μέσω AJAX.

Χρήση:

- Ανάκτηση δεδομένων διαδικασιών, βημάτων, αρχείων (GET).
- Υποβολή αλλαγών κατάστασης, έναρξη ή ολοκλήρωση διαδικασιών (POST).
- Υποστήριξη αποστολής multipart/form-data για αρχεία.

tailwindcss

Το tailwindcss [5] είναι utility-first CSS framework που χρησιμοποιήθηκε για τη σχεδίαση του UI. Προσφέρει έτοιμες κλάσεις για αποδοτική σχεδίαση χωρίς δημιουργία custom CSS αρχείων.

Χρήση:

- Στυλιστική ομοιομορφία στην εφαρμογή.
- Responsive layout.
- Ταχεία ανάπτυξη χωρίς προσθήκη πολύπλοκου CSS.

lucide-react

Η βιβλιοθήκη lucide-react [7] προσφέρει ένα σύνολο από minimal SVG εικονίδια React components.

Χρήση:

- Εικονίδια πλοήγησης (sidebar, header).
- Οπτική διάκριση μεταξύ διαφορετικών τύπων διαδικασιών.
- Βελτίωση προσβασιμότητας και αναγνωρισιμότητας του UI.

Αυτές οι βιβλιοθήκες αποτέλεσαν τον κορμό της λειτουργικότητας και του UI design του συστήματος. Με τη χρήση τους διασφαλίστηκε ότι η εφαρμογή είναι επεκτάσιμη, εύκολα συντηρήσιμη και προσφέρει θετική εμπειρία χρήστη.

4.3 Παραδείγματα Υλοποίησης

Η παρούσα ενότητα παραθέτει ενδεικτικά αποσπάσματα κώδικα και παραδείγματα υλοποίησης που αναδεικνύουν βασικές πτυχές της λειτουργίας του Συστήματος Καταγραφής Διαδικασιών. Μέσω των παραδειγμάτων επιτυγχάνεται καλύτερη κατανόηση του τρόπου με τον οποίο υλοποιήθηκαν σημαντικές λειτουργίες όπως η διαχείριση κατάστασης (state management), η πλοήγηση σε πολλαπλές σελίδες (routing), η δυναμική απόδοση περιεχομένου

(dynamic rendering), η ενσωμάτωση ημερολογίου, η επικοινωνία με το backend και η παρουσίαση διαδραστικών ειδοποιήσεων.

Τα παραδείγματα βασίζονται στην τεχνολογική στοίβα του React, αξιοποιώντας hooks, components και τρίτες βιβλιοθήκες. Η παρουσίασή τους ακολουθεί θεματική διάρθρωση, ώστε να αποσαφηνίζεται το πλαίσιο και η χρησιμότητα κάθε ενότητας.

4.3.1 Component Structure & State Management

Η αρχιτεκτονική των components του συστήματος ακολουθεί την αρχή της επαναχρησιμοποίησης και της διάσπασης της λειτουργικότητας σε επιμέρους δομικά στοιχεία. Κάθε διαδικασία και τα βήματά της απεικονίζονται με τη χρήση διακριτών components, τα οποία επικοινωνούν μεταξύ τους μέσω props και αξιοποιούν hooks όπως useState και useEffect για τη διαχείριση της κατάστασης και την ανάκτηση δεδομένων από το backend.

Παράδειγμα: ProcessCard.js

Το component ProcessCard αποτελεί βασικό δομικό στοιχείο για την εμφάνιση μιας διαδικασίας. Χρησιμοποιεί την useState για την αποθήκευση των βημάτων (steps) και για την ένδειξη φόρτωσης (loading). Η useEffect χρησιμοποιείται για να ανακτηθούν τα δεδομένα των βημάτων μέσω axios, μόνο όταν η διαδικασία δεν περιλαμβάνει sections.

Επιπλέον, γίνεται έλεγχος της κατάστασης loading και γίνεται conditional rendering της σελίδας: αν υπάρχουν sections, εμφανίζονται με dangerouslySetInnerHTML, αλλιώς αποδίδεται δυναμικά η λίστα των βημάτων με χρήση του StepCard component.

Κύρια Σημεία Υλοποίησης:

- useEffect: για fetch δεδομένων από API όταν το component "ανεβαίνει".
- useState: για την παρακολούθηση της λίστας των βημάτων και της φόρτωσης.
- props: για τη μετάδοση του αντικειμένου process από γονικό component.
- Conditional rendering για να αποδοθούν διαφορετικά layouts ανάλογα με τη δομή της διαδικασίας.

```
const [steps, setSteps] = useState([]);
const [loading, setLoading] = useState(true);
```

```
useEffect(() => {
  if (!hasSections) {
    axios
      .get(
        `http://localhost/process-tracking-api/index.php?request=steps&process_id=${process.id}`
      )
      .then((res) => {
        setSteps(res.data.steps || []);
        setLoading(false);
      })
      .catch((err) => {
        console.error("Error fetching steps:", err);
        setLoading(false);
      });
  } else {
    setLoading(false);
  }
}, [process.id, hasSections]);
```

Παράδειγμα: StepCard.js

Το component StepCard διαχειρίζεται ένα μεμονωμένο βήμα μιας διαδικασίας. Παρόμοια με το ProcessCard, κάνει χρήση των useState και useEffect για να ανακτήσει τα συνημμένα αρχεία του βήματος και να ελέγξει την κατάσταση εμφάνισης λεπτομερειών ή αρχείων (showDetails, showFiles).

Ο χρήστης έχει τη δυνατότητα να προβάλει επιπλέον λεπτομέρειες για το βήμα με χρήση toggle state, καθώς και να αναπτύξει λίστα αρχείων μέσω δυναμικού rendering.

Κύρια Σημεία Υλοποίησης:

- useState: για τον έλεγχο εμφάνισης λεπτομερειών και αρχείων.
- useEffect: για ανάκτηση αρχείων από το backend με βάση το step.id.
- Conditional rendering: για εναλλαγή περιεχομένου ανάλογα με τις ενέργειες του χρήστη.
- dangerouslySetInnerHTML: για ασφαλή εμφάνιση HTML περιεχομένου των supplementary details.

```

const StepCard = ({ step }) => {
  const [showDetails, setShowDetails] = useState(false);
  const [showFiles, setShowFiles] = useState(false);
  const [files, setFiles] = useState([]);
  const [loadingFiles, setLoadingFiles] = useState(true);

  useEffect(() => {
    axios
      .get(
        `http://localhost/process-tracking-api/get_original_step_files.php?step_id=${step.id}`
      )
      .then((response) => {
        if (response.data.success) {
          setFiles(response.data.files);
        }
        setLoadingFiles(false);
      })
      .catch((error) => {
        console.error("Error fetching files:", error);
        setLoadingFiles(false);
      });
  }, [step.id]);
}

```

4.3.2 Routing

Η πλοήγηση εντός του συστήματος πραγματοποιείται μέσω της βιβλιοθήκης react-router-dom, η οποία παρέχει ένα ευέλικτο σύνολο εργαλείων για τη διαχείριση διαφορετικών διαδρομών (routes) σε μία εφαρμογή React.

Για την υλοποίηση του routing δημιουργήθηκε το αρχείο AppRoutes.js, το οποίο ορίζει όλα τα βασικά routes του συστήματος. Το component Routes περικλείει όλα τα Route στοιχεία, όπου κάθε Route συσχετίζει μια συγκεκριμένη διαδρομή URL με το αντίστοιχο page component.

Κύρια Σημεία Υλοποίησης:

- Η αρχική σελίδα ("/") οδηγεί στο Dashboard.
- Υπάρχει ξεχωριστή διαδρομή για κάθε κύρια λειτουργία της εφαρμογής:
 - /all-processes → Σελίδα παρουσίασης όλων των διαδικασιών.
 - /my-processes → Είσοδος στις προσωπικές διαδικασίες του χρήστη.
 - /my-processes/running → Προβολή των διαδικασιών που είναι σε εξέλιξη.
 - /my-processes/completed → Προβολή των διαδικασιών που έχουν ολοκληρωθεί.
 - /calendar → Πρόσβαση στο προσωπικό ημερολόγιο του χρήστη.

Το αρχείο AppRoutes.js διατηρεί την πλοήγηση σαφή και απλή, διευκολύνοντας τη συντήρηση και επεκτασιμότητα του project στο μέλλον.

```
const AppRoutes = () => {
  return (
    <Routes>
      { /* Set Dashboard as the default route */ }
      <Route path="/" element={ <Dashboard /> } />
      <Route path="/all-processes" element={ <AllProcesses /> } />
      <Route path="/my-processes" element={ <MyProcesses /> } />
      <Route path="/my-processes/running" element={ <RunningProcesses /> } />
      <Route path="/my-processes/completed" element={ <CompletedProcesses /> } />
      <Route path="/calendar" element={ <Calendar /> } />
    </Routes>
  );
};
```

Οφέλη της Χρήσης του react-router-dom:

- Δημιουργία Single Page Application (SPA) χωρίς επαναφόρτωση σελίδας.
- Οργάνωση και διαχείριση πολλαπλών σελίδων με καθαρό και modular τρόπο.
- Ευκολία στην επέκταση με nested routes ή dynamic routes σε μελλοντικές αναβαθμίσεις.

4.3.3 Δυναμική Προβολή Δεδομένων

Η δυναμική προβολή δεδομένων αποτελεί βασικό στοιχείο της λειτουργικότητας του συστήματος. Μέσω αυτής, το περιεχόμενο που εμφανίζεται στους χρήστες δημιουργείται και προσαρμόζεται σε πραγματικό χρόνο, ανάλογα με τα δεδομένα που αντλούνται από το backend.

Η υλοποίηση επιτυγχάνεται με τη χρήση των React hooks (useState, useEffect) και της μεθόδου .map() της JavaScript, η οποία επιτρέπει την απόδοση λίστας components με βάση τα δεδομένα που επιστρέφονται από το API.

Παράδειγμα Υλοποίησης – Σελίδα Όλων των Διαδικασιών (AllProcesses.js):

Στην εν λόγω σελίδα, γίνεται δυναμική ανάκτηση όλων των διαδικασιών μέσω HTTP αιτήματος προς το backend:

```
.get("http://localhost/process-tracking-api/index.php?request=processes")
.then((response) => {
  setProcesses(response.data.data || []);
  setLoading(false);
})
```

Τα δεδομένα αποθηκεύονται στο state hook processes και, εφόσον ολοκληρωθεί η φόρτωσή τους (loading === false), γίνεται δυναμική απεικόνισή τους με χρήση της συνάρτησης .map():

```
processes.map((process) => (
  <ProcessCard key={process.id} process={process} />
))
```

Το παραπάνω block δημιουργεί και αποδίδει ένα ProcessCard component για κάθε διαθέσιμη διαδικασία, προσδίδοντας στο σύστημα μια responsive και δυναμική εμπειρία χρήσης.

Επιπλέον Δυναμική Συμπεριφορά:

Η δυνατότητα επιλογής συγκεκριμένης διαδικασίας μέσα από τη NavigationBar ενεργοποιεί scroll προς το αντίστοιχο στοιχείο στη σελίδα μέσω της συνάρτησης scrollToView():

```
onSelect={ (id) => {
  const el = document.getElementById(`process-${id}`);
  if (el) {
    el.scrollToView({ behavior: "smooth", block: "start" });
    setSelectedProcessId(id);
  }
}}
```

Οφέλη της Προσέγγισης:

- Καλύτερη εμπειρία χρήστη μέσω προσαρμοσμένου περιεχομένου.
- Βελτιωμένη απόδοση με αποδοτική διαχείριση δεδομένων.
- Κωδικός ευανάγνωστος, επεκτάσιμος και εύκολος στη συντήρηση

4.3.4 Calendar Integration (FullCalendar JS)

Στο πλαίσιο της υλοποίησης του Συστήματος Καταγραφής Διαδικασιών, ενσωματώθηκε ένα δυναμικό ημερολόγιο με χρήση της βιβλιοθήκης FullCalendar JS, προσφέροντας στον χρήστη τη δυνατότητα διαχείρισης προσωπικών συμβάντων.

Η ενσωμάτωση επιτεύχθηκε μέσω του React component FullCalendar και περιλάμβανε τις βασικές λειτουργίες δημιουργίας και προβολής γεγονότων.

Βασικά χαρακτηριστικά ενσωμάτωσης:

- Προβολή Ημερολογίου: Το FullCalendar απεικονίζει τα γεγονότα με την προβολή dayGridMonth.
- Επιλογή Ημερομηνίας (Date Click): Ο χρήστης μπορεί να δημιουργήσει νέο γεγονός κλικάροντας σε ημερομηνία.
- Προβολή Υφιστάμενων Γεγονότων (Event Click): Ο χρήστης μπορεί να δει λεπτομέρειες ενός υπάρχοντος γεγονότος.

- Διαχείριση Γεγονότων μέσω API: Τα γεγονότα αποθηκεύονται και ανακτώνται μέσω backend endpoints (PHP API).

Κώδικας υλοποίησης ημερολογίου (Calendar.js):

Το ημερολόγιο δημιουργείται με τη χρήση των plugins dayGridPlugin και interactionPlugin, υποστηρίζοντας την επιλογή ημερομηνιών και την αλληλεπίδραση με συμβάντα:

```
<FullCalendar
  plugins={[dayGridPlugin, interactionPlugin]}
  initialView="dayGridMonth"
  editable={false}
  selectable
  events={events}
  dateClick={handleDateClick}
  eventClick={handleEventClick}
  locale={elLocale}
/>
```

Όλα τα γεγονότα φορτώνονται δυναμικά από το API μέσω της συνάρτησης fetchEvents, ενώ η αποστολή νέων γεγονότων γίνεται με axios.post:

```
const fetchEvents = useCallback(async () => {
  try {
    const response = await axios.get(
      `${API_BASE}?request=calendar-events&user_id=${userId}`
    );
  }, [userId]);
```

Στιγμιότυπο FetchEvent/axios

```
const handleSubmitEvent = async () => {
  const response = await axios.post(
    `${API_BASE}?request=create-event`,
    newEvent,
    { headers: { "Content-Type": "application/json" } }
```

Στιγμιότυπο FetchEvent/axios

Η χρήση του locale elLocale καθιστά το ημερολόγιο διαθέσιμο στην Ελληνική γλώσσα.

Διαχείριση Backend - calendar_events.php:

Το backend endpoint διαχειρίζεται:

- Ανάκτηση γεγονότων (calendar-events request):
Επιστρέφει όλα τα γεγονότα ενός χρήστη.

- Δημιουργία νέου γεγονότος (create-event request):
Εισάγει νέο γεγονός στη βάση δεδομένων.

Παράδειγμα SQL query για την ανάκτηση γεγονότων:

```
SELECT id, user_id, title, description,  
       DATE_FORMAT(date_time, '%Y-%m-%dT%H:%i:%s') as start  
FROM calendar_events  
WHERE user_id = ?  
ORDER BY id ASC
```

Οφέλη της Προσέγγισης:

- Ευχρηστία: Φιλική και άμεση διαχείριση ημερολογιακών συμβάντων.
- Αποδοτικότητα: Επικοινωνία με backend σε πραγματικό χρόνο.
- Επεκτασιμότητα: Δυνατότητα μελλοντικής ενσωμάτωσης λειτουργιών, όπως διαγραφή ή ενημέρωση γεγονότων.

4.3.5 Επικοινωνία με Backend (axios)

Η επικοινωνία του React frontend με το backend του Συστήματος Καταγραφής Διαδικασιών υλοποιείται μέσω της βιβλιοθήκης [axios](#), διευκολύνοντας τις αιτήσεις και απαντήσεις HTTP (GET, POST) μεταξύ client και server.

Η διασύνδεση αυτή είναι θεμελιώδης για:

- Ανάκτηση διαδικασιών (τρέχουσες και συμπληρωμένες).
- Ανάκτηση και αποστολή αρχείων.
- Ενημέρωση κατάστασης βημάτων και διαδικασιών.
- Προσθήκη γεγονότων στο ημερολόγιο.

Παραδείγματα Επικοινωνίας

1. Ανάκτηση Τρεχουσών Διαδικασιών

Στο RunningProcesses.js, χρησιμοποιείται η μέθοδος axios.get για την ανάκτηση όλων των διαδικασιών που είναι σε εξέλιξη για τον χρήστη:

```

axios
  .get(
    `http://localhost/process-tracking-api/index.php?request=running-processes&user_id=${userId}`
  )
  .then((response) => {
    setRunningProcesses(response.data.data || []);
    setLoading(false);
  })
  .catch((error) => {
    console.error("Error fetching running processes:", error);
    setLoading(false);
  });

```

Η απάντηση από το backend περιλαμβάνει τις διαδικασίες μαζί με τα αντίστοιχα βήματά τους (steps), ώστε να είναι διαθέσιμα για απόδοση στο UI.

2. Ολοκλήρωση Διαδικασίας (Μετάβαση σε Συμπληρωμένες)

Η μεταφορά μιας διαδικασίας από "τρέχουσα" σε "συμπληρωμένη" υλοποιείται με axios.post στο update-process-status endpoint:

```

axios
  .post(
    "http://localhost/process-tracking-api/index.php?request=update-process-status",
    JSON.stringify({
      user_id: userId,
      process_id: process.running_process_id,
    }),
    { headers: { "Content-Type": "application/json" } }
  )

```

Εφόσον η απάντηση είναι επιτυχής (response.data.success), ενημερώνεται η κατάσταση της διαδικασίας στο frontend.

3. Ολοκλήρωση ή Ακύρωση Βημάτων

Σε επίπεδο μεμονωμένων βημάτων (StepDetails component), γίνεται επίσης χρήση axios.post για την αλλαγή κατάστασης ενός βήματος:

```

axios
  .post(
    "http://localhost/process-tracking-api/index.php?request=update-step-status",
    JSON.stringify({ user_id: userId, step_id: step.id, action }),
    { headers: { "Content-Type": "application/json" } }
  )

```

Ανάλογα με την επιλογή του χρήστη (ολοκλήρωση ή μη-ολοκλήρωση), αποστέλλεται αντίστοιχο action ("complete" ή "uncomplete").

4. Ανάκτηση Συμπληρωμένων Διαδικασιών

Στο CompletedProcesses.js, με την ίδια λογική, χρησιμοποιείται axios.get για την ανάκτηση διαδικασιών που έχουν ήδη ολοκληρωθεί:

```

axios
.get(`http://localhost/process-tracking-api/index.php?request=completed-processes&user_id=${userId}`)
.then((response) => {
  setCompletedProcesses(response.data.data || []);
  setLoading(false);
})
.catch((error) => {
  console.error("Error fetching completed processes:", error);
  setLoading(false);
});

```

Η απάντηση περιλαμβάνει τα βήματα κάθε ολοκληρωμένης διαδικασίας καθώς και τυχόν ανεβασμένα αρχεία.

5. Ανάκτηση και Αποστολή Αρχείων

Ανάκτηση αρχείων βημάτων:

```

axios
.get(
  `http://localhost/process-tracking-api/get_original_step_files.php?step_id=${step.id}`
)

```

Αποστολή αρχείων (upload) με multipart/form-data:

```

axios
.post("http://localhost/process-tracking-api/upload_file.php", formData, {
  headers: { "Content-Type": "multipart/form-data" },
})

```

Το FormData αντικείμενο περιλαμβάνει πληροφορίες για το αρχείο, το id του βήματος, το id του χρήστη και το id της διαδικασίας.

Σημαντικές Παρατηρήσεις:

- Για τις POST αιτήσεις αποστέλλονται JSON σώματα (Content-Type: application/json).
- Όλες οι κλήσεις axios περιβάλλονται με error handling (catch) για ασφάλεια και σταθερότητα της εφαρμογής.
- Το localhost URL δείχνει ότι το σύστημα εκτελείται σε τοπικό development περιβάλλον.

4.3.6 Ειδοποιήσεις και Επιβεβαιώσεις (Toastify Notifications και Προσαρμοσμένα Modals)

Για την επικοινωνία σημαντικών γεγονότων και την παροχή άμεσης πληροφόρησης προς τον χρήστη, το σύστημα χρησιμοποιεί πλέον τη βιβλιοθήκη react-toastify [17] για την εμφάνιση προσωρινών ειδοποιήσεων στην οθόνη, καθώς και προσαρμοσμένα modals επιβεβαίωσης υλοποιημένα με React και CSS. Η επιλογή αυτών των εργαλείων αποσκοπεί στη βελτίωση της

εμπειρίας χρήστη, προσφέροντας οπτικά φιλικότερες και λιγότερο ενοχλητικές αλληλεπιδράσεις σε σύγκριση με τα κλασικά alert boxes της JavaScript.

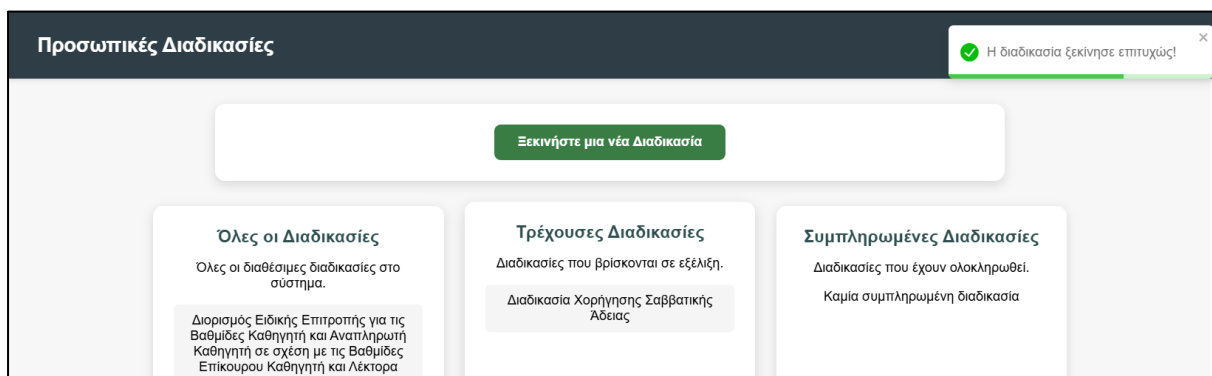
Toastify Notifications:

Τα toastify notifications διαφοροποιούνται ανάλογα με το περιεχόμενο και τη σοβαρότητα του μηνύματος. Συγκεκριμένα:

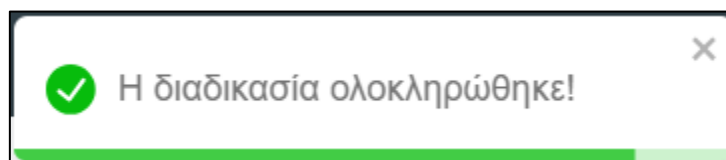
- Το **toast.success()** χρησιμοποιείται για θετικές επιβεβαιώσεις, όπως επιτυχής ολοκλήρωση ενός βήματος ή διαδικασίας.
- Το **toast.error()** εμφανίζεται όταν συμβαίνουν σφάλματα, όπως αποτυχία επικοινωνίας με τον server ή απόρριψη αιτήματος από το backend.
- Το **toast.warning()** αφορά προειδοποιήσεις που υποδεικνύουν ότι ο χρήστης πρέπει να διορθώσει ή να συμπληρώσει κάτι (π.χ. υποχρεωτικό πεδίο δεν έχει συμπληρωθεί).
- Το **toast.info()** χρησιμοποιείται για ουδέτερη ενημέρωση του χρήστη σε περιπτώσεις που δεν απαιτείται άμεση δράση, αλλά παρέχεται χρήσιμη πληροφορία (π.χ. απουσία αρχείων για αποστολή).

Η διάκριση αυτή επιτρέπει πιο φυσική και ξεκάθαρη επικοινωνία με τον χρήστη, ενισχύοντας τη χρηστικότητα και την αλληλεπίδραση με το σύστημα.

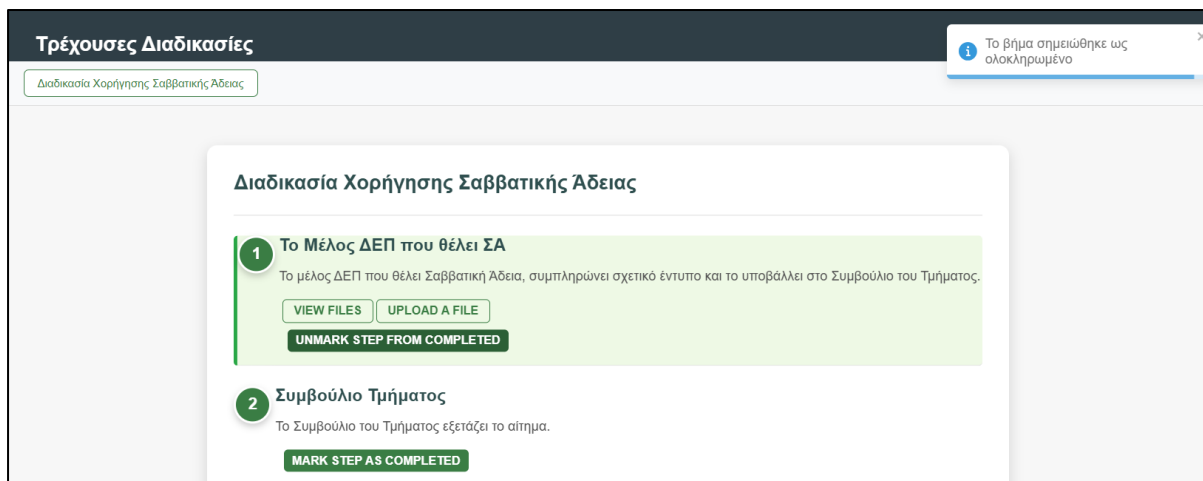
Ενδεικτικά παραδείγματα:



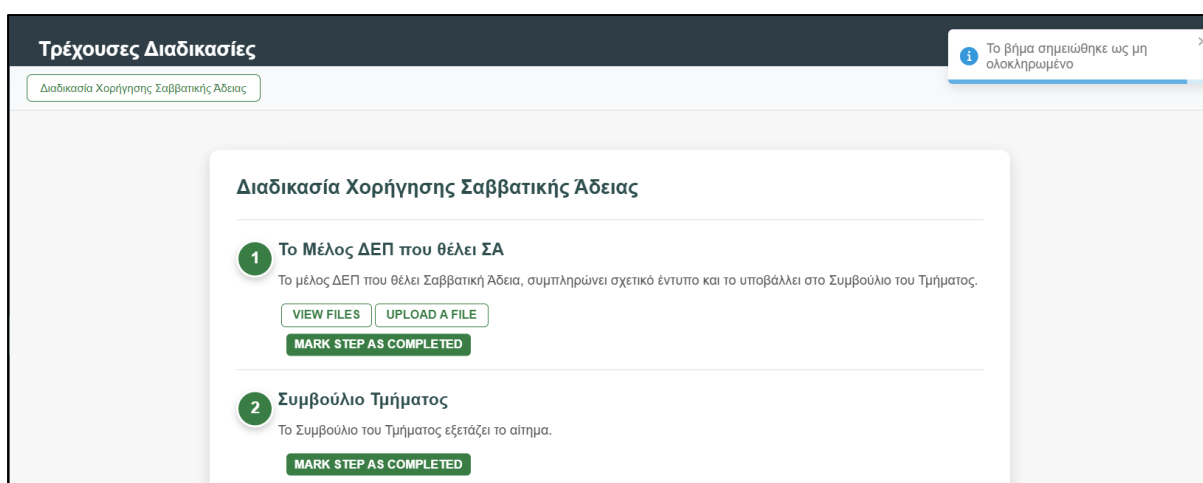
Στιγμιότυπο toast.success(): Επιτυχής Εκκίνηση μιας Διαδικασίας



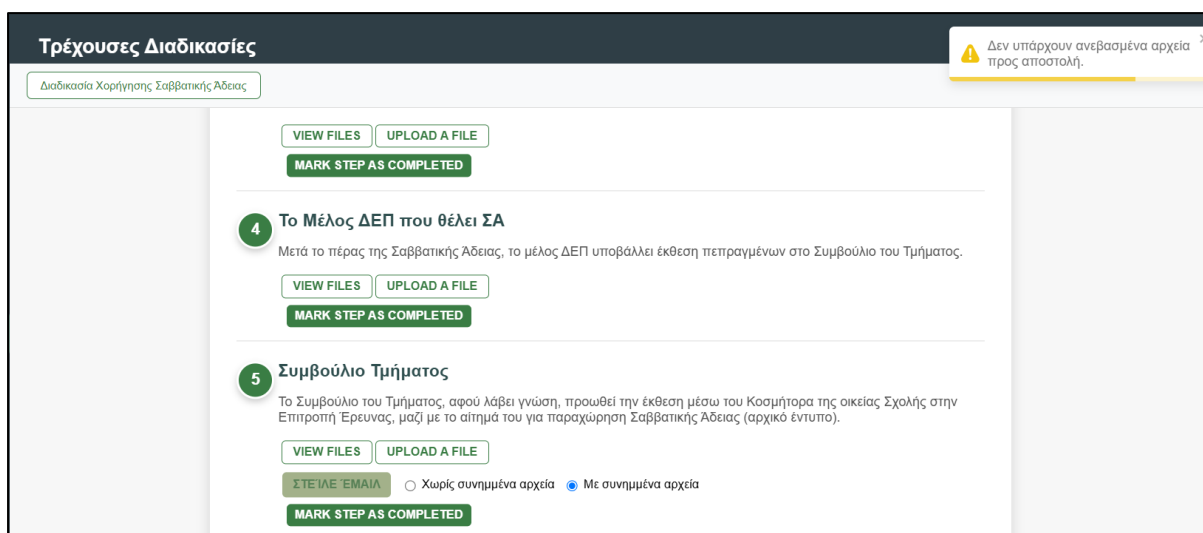
Στιγμιότυπο toast.success(): Επιτυχής Ολοκλήρωση μιας Διαδικασίας



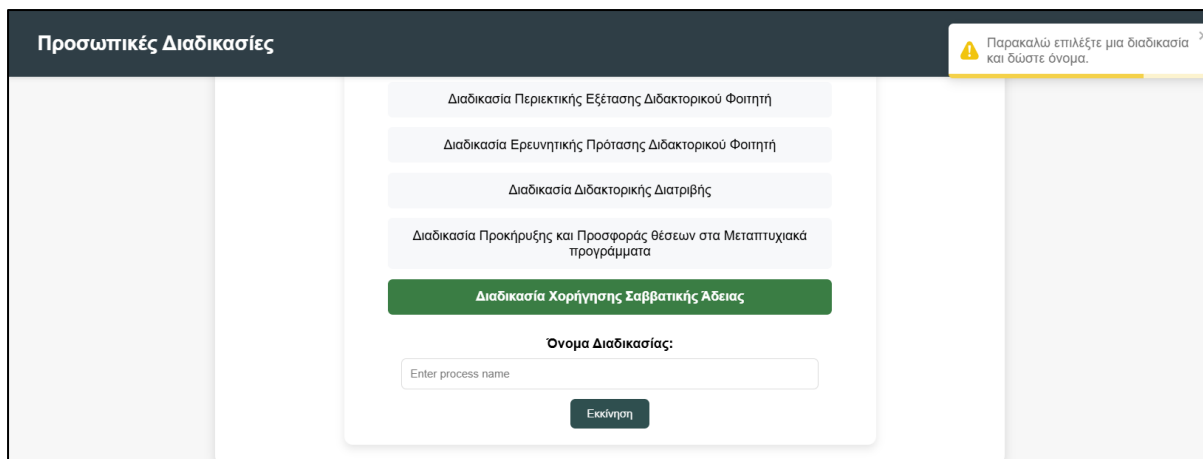
Στιγμιότυπο `toast.info()`: Επιτυχής Ολοκλήρωση Βήματος



Στιγμιότυπο `toast.info()`: Ορισμός Βήματος ως Εκκρεμές



Στιγμιότυπο `toast.warning()`: Απαιτούνται αρχεία προς αποστολή



Στιγμιότυπο `toast.warning()`: Απαιτείται Όνομα Διαδικασίας

Αυτή η μέθοδος ενισχύει την ομαλότητα και την αισθητική συνοχή της διεπαφής χρήστη, χωρίς να διακόπτει τη ροή των ενεργειών του.

Custom Modals (React-based):

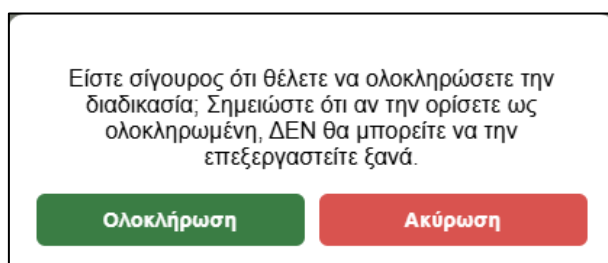
Για πιο κρίσιμες ενέργειες, όπως η ολοκλήρωση μιας διαδικασίας ή η προειδοποίηση για μη ολοκληρωμένα βήματα, χρησιμοποιούνται προσαρμοσμένα modals που δημιουργούνται με React components.

Αυτά τα modals προσφέρουν:

- Βελτιωμένη αισθητική με κουμπιά δράσης διαφορετικών χρωμάτων.
- Πιο φιλική εμπειρία για την επιβεβαίωση ή ακύρωση ενεργειών.
- Διαχωρισμό μεταξύ "Επιβεβαίωσης" και "Προειδοποίησης".

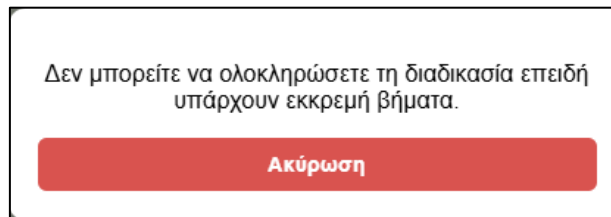
Συγκεκριμένα:

- Όταν ο χρήστης επιχειρεί να ολοκληρώσει μία διαδικασία, εμφανίζεται modal επιβεβαίωσης με επιλογές "Ολοκλήρωση" και "Ακύρωση", όπως φαίνεται στην Εικόνα 25:



Εικόνα 25 – Στιγμιότυπο Μήνυμα Επιβεβαίωσης

- Σε περίπτωση που υπάρχουν μη ολοκληρωμένα βήματα και ο χρήστης προσπαθήσει να ολοκληρώσει τη διαδικασία, εμφανίζεται προειδοποιητικό modal (Εικόνα 26) που ενημερώνει ότι δεν είναι δυνατή η ολοκλήρωση:



Εικόνα 24 – Στιγμιότυπο Προειδοποιητικό Μήνυμα

Η λογική εμφάνισης αυτών των παραθύρων ελέγχεται μέσω state μεταβλητών (showModal, modalMessage, onConfirm) και απλών χειριστών (showInfoModal, showConfirmationModal) στα components.

Ενδεικτικός κώδικας δημιουργίας custom modal:

```
{showModal && (
  <div className="modal-overlay">
    <div className="modal">
      <p>{modalMessage}</p>
      <div className="modal-buttons">
        {onConfirm !== null && (
          <button className="primary-btn" onClick={onConfirm}>
            Ολοκλήρωση
          </button>
        )}
        <button
          className="cancel-btn"
          onClick={() => setShowModal(false)}
        >
          Ακύρωση
        </button>
      </div>
    </div>
  </div>
)}
```

Η διαφοροποίηση της εμπειρίας ειδοποίησης ανάλογα με τη σημασία του συμβάντος αποτελεί σημαντική προσθήκη στη χρηστικότητα και αξιοπιστία του συστήματος.

Κεφάλαιο 5

Συμπεράσματα

5.1 Συμπεράσματα	73
5.2 Έλεγχος Συστήματος	73
5.3 Επεκτασιμότητα Συστήματος και Μελλοντική Εργασία	74
5.4 Συνεργασία και Εμπειρίες που Αποκτήθηκαν	76

5.1 Συμπεράσματα

Με την ολοκλήρωση της ανάπτυξης και της υλοποίησης του «Συστήματος Καταγραφής Διαδικασιών», εκπληρώθηκε με επιτυχία ο κύριος στόχος της εργασίας: η οργάνωση, η διαχείριση και η διευκόλυνση της παρακολούθησης διοικητικών και ακαδημαϊκών διαδικασιών του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου. Το σύστημα προσφέρει ένα εύχρηστο περιβάλλον στους χρήστες για την εκκίνηση, παρακολούθηση και ολοκλήρωση διαδικασιών με διαδραστικό και αποτελεσματικό τρόπο, συμβάλλοντας στη μείωση της γραφειοκρατίας και στην αποδοτικότερη διεκπεραίωση των εργασιών.

Παρέχει επίσης άμεση πρόσβαση σε κρίσιμες πληροφορίες, όπως απαιτούμενα έγγραφα, περιγραφές βημάτων, πηγές και αρχεία, ενώ ενισχύει την ατομική οργάνωση μέσω του προσωπικού ημερολογίου. Συνολικά, το σύστημα συμβάλλει στη βελτίωση της εσωτερικής επικοινωνίας, της ατομικής παραγωγικότητας και της συνολικής διοικητικής και ακαδημαϊκής λειτουργίας του Τμήματος.

5.2 Έλεγχος Συστήματος

Κατά τον μήνα Απρίλιο, το «Σύστημα Καταγραφής Διαδικασιών» παρουσιάστηκε σε μέλη του διοικητικού προσωπικού και του ακαδημαϊκού προσωπικού του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου. Στην παρουσίαση αυτή, πραγματοποιήθηκε αναλυτική επισκόπηση των βασικών λειτουργιών του συστήματος, με στόχο την αξιολόγηση της πληρότητας, της λειτουργικότητας και της ευχρηστίας του.

Η ανατροφοδότηση που λήφθηκε ήταν ιδιαίτερα ενθαρρυντική, αναδεικνύοντας τη σαφήνεια της διεπαφής χρήστη και την πληρότητα των πληροφοριών που παρέχονται για κάθε διαδικασία. Παράλληλα, τέθηκαν κάποιες προτάσεις για περαιτέρω βελτίωση της εμπειρίας χρήσης, όπως η απαγόρευση επεξεργασίας μιας διαδικασίας μετά την οριστική ολοκλήρωσή της, με σκοπό τη διατήρησή της αποκλειστικά για λόγους ενημέρωσης. Επιπλέον, τα τροποποιημένα αρχεία του χρήστη διατηρούνται και παραμένουν διαθέσιμα προς εγκατάσταση. Οι αλλαγές αυτές υλοποιήθηκαν με στόχο τη διασφάλιση της ακεραιότητας και της εγκυρότητας των συμπληρωμένων διαδικασιών.

Ακολούθησε η φάση ελέγχου του συστήματος με πραγματικά δεδομένα, μέσω εισαγωγής διαδικασιών και βημάτων που χρησιμοποιούνται καθημερινά στο πλαίσιο της λειτουργίας του Τμήματος. Ο έλεγχος επικεντρώθηκε στην ορθή λειτουργικότητα των διαδικασιών εκκίνησης, παρακολούθησης και ολοκλήρωσης, καθώς και στη σωστή διαχείριση των συνημμένων αρχείων.

Τέλος, στα μέσα Μαΐου πραγματοποιήθηκε στοχευμένος έλεγχος ευχρηστίας και συνολικής εμπειρίας χρήστη, με τη συμμετοχή εκπροσώπων από διαφορετικές κατηγορίες χρηστών του Τμήματος. Συγκεκριμένα, μεταξύ άλλων, στη διαδικασία αξιολόγησης συμμετείχαν:

Η κα. Σαββούλλα Ευσταθίου, μέλος της Γραμματείας, η οποία εστίασε στην ευκολία χρήσης της πλατφόρμας και στη σαφήνεια των οδηγιών.

Ο κ. Χρύσης Γεωργίου, επιβλέπων καθηγητής, ο οποίος εξέτασε την πληρότητα των λειτουργιών και την τεχνική ορθότητα.

Η κα. Ζέτα Καπιτσάκη, Αναπλ. Καθηγήτρια, που αξιολόγησε την ακαδημαϊκή συμβατότητα του συστήματος με τις διαδικασίες του Τμήματος.

Κατά τον έλεγχο αυτό, δοκιμάστηκαν βασικές ροές εργασίας, όπως η εκκίνηση νέας διαδικασίας, η αλληλεπίδραση με τα βήματα, η χρήση του ημερολογίου και η αρχειοθέτηση εγγράφων. Η εμπειρία χρήσης χαρακτηρίστηκε ιδιαίτερα θετική, ενώ δεν καταγράφηκαν πρόσθετες ανάγκες για αλλαγές ή προσθήκες. Το σύστημα κρίθηκε ώριμο και πλήρως συμβατό με τις ανάγκες που τέθηκαν κατά τον αρχικό σχεδιασμό, επιβεβαιώνοντας την ευχρηστία, αξιοπιστία και λειτουργικότητά του στην πράξη.

5.3 Επεκτασιμότητα Συστήματος και Μελλοντική Εργασία

Το «Σύστημα Καταγραφής Διαδικασιών», αν και ολοκληρωμένο και λειτουργικό στη σημερινή του μορφή, διαθέτει σημαντικά περιθώρια περαιτέρω εξέλιξης και εμπλουτισμού.

Στο πλαίσιο της μελλοντικής εργασίας και επεκτασιμότητας, εντοπίζονται διάφορες δυνατότητες βελτίωσης και διεύρυνσης των παρεχόμενων λειτουργιών.

Αρχικά, θα μπορούσε να προστεθεί η δυνατότητα ζωντανής επικοινωνίας (live chat) εντός της πλατφόρμας, επιτρέποντας στους χρήστες να ανταλλάσσουν άμεσα μηνύματα κατά τη διάρκεια της εκτέλεσης των διαδικασιών. Επιπλέον, η ενσωμάτωση αποστολής ειδοποιήσεων μέσω ηλεκτρονικού ταχυδρομείου (email) ή και μηνυμάτων SMS, σε περιπτώσεις όπως η ολοκλήρωση ενός βήματος ή η ανάγκη ανάληψης δράσης, θα ενίσχυε τη διαδραστικότητα και την αμεσότητα της πλατφόρμας.

Μία ακόμα επέκταση θα μπορούσε να αφορά τη δυνατότητα κεντρικής παρουσίασης σημαντικών εγγράφων, όπως ενημερωτικά σημειώματα, ανακοινώσεις τμήματος ή συλλογικά πρακτικά συναντήσεων, καθιστώντας το σύστημα έναν κεντρικό κόμβο πληροφόρησης για όλα τα μέλη του Τμήματος. Παράλληλα, θα μπορούσε να προστεθεί ένα σύστημα υποβολής παραπόνων ή εισηγήσεων μέσω ειδικής φόρμας («κουτί προτάσεων»), ενισχύοντας τη διαδραστικότητα και δίνοντας στους χρήστες φωνή για συνεχή βελτίωση του συστήματος.

Από άποψης στρατηγικής ανάπτυξης, μία πιθανή μελλοντική εργασία θα μπορούσε να είναι η επέκταση του συστήματος ώστε να καλύπτει επιπρόσθετες ομάδες χρηστών, πέραν του ακαδημαϊκού και διοικητικού προσωπικού, όπως φοιτητές. Για παράδειγμα, οι φοιτητές θα μπορούσαν να χρησιμοποιούν το σύστημα για την παρακολούθηση προσωπικών ακαδημαϊκών διαδικασιών, την υποβολή αιτήσεων, ή ακόμα και τη διαχείριση προσωπικών συμβάντων, δημιουργώντας έτσι ένα ενιαίο, ενοποιημένο περιβάλλον ακαδημαϊκής διαχείρισης.

Τέλος, δεδομένου ότι το σύστημα υποστηρίζει προς το παρόν μόνο την ελληνική γλώσσα, μία σημαντική μελλοντική επέκταση αφορά την προσθήκη υποστήριξης για την αγγλική γλώσσα. Η πολυγλωσσική διεπαφή θα επιτρέψει την ευρύτερη υιοθέτηση του συστήματος, διευκολύνοντας τη χρήση του από μη ελληνόφωνους χρήστες και ενισχύοντας τη διαλειτουργικότητά του σε διεθνές ή διαπανεπιστημιακό πλαίσιο.

Η υλοποίηση των παραπάνω προτάσεων θα αναβαθμίσει τη λειτουργικότητα και την αξία του «Συστήματος Καταγραφής Διαδικασιών», καθιστώντας το ένα ολοκληρωμένο εργαλείο οργάνωσης, συνεργασίας και επικοινωνίας για όλα τα μέλη του Τμήματος Πληροφορικής.

5.4 Συνεργασία και Εμπειρίες που Αποκτήθηκαν

Κατά τη διάρκεια της Ατομικής Διπλωματικής Εργασίας, είχα την ευκαιρία να συνεργαστώ σε ένα πραγματικό έργο που αφορούσε την κάλυψη πραγματικών αναγκών του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου. Η συνεργασία αυτή αποτέλεσε μια πολύτιμη εμπειρία, καθώς με έφερε σε άμεση επαφή με τη διαδικασία λήψης απαιτήσεων, επικοινωνίας με ενδιαφερόμενους φορείς και προσαρμογής σε πραγματικά λειτουργικά περιβάλλοντα.

Η στενή συνεργασία με τον κ. Χρύση Γεωργίου και την κ. Σαββούλλα Ευσταθίου υπήρξε καταλυτική για τη σωστή κατανόηση των διαδικασιών, την αποσαφήνιση των επιμέρους απαιτήσεων και την έγκαιρη επίλυση αποριών που προέκυπταν κατά την υλοποίηση. Μέσα από αυτή την επικοινωνία αναπτύχθηκαν δεξιότητες επαγγελματισμού, υπομονής και συνεργατικότητας, οι οποίες θεωρούνται θεμελιώδεις σε ένα απαιτητικό εργασιακό περιβάλλον.

Η ανάπτυξη του «Συστήματος Καταγραφής Διαδικασιών» απαίτησε την εκμάθηση και εφαρμογή νέων τεχνολογιών και μεθοδολογιών. Εμβάθυνα περαιτέρω στη χρήση της ReactJS για την υλοποίηση της διεπαφής χρήστη, στο Axios για τη διαχείριση αιτημάτων προς τον server, καθώς και στη διασύνδεση με backend λογική βασισμένη σε PHP και MySQL. Επιπλέον, αναπτύχθηκαν δεξιότητες στη διαχείριση καταστάσεων (state management), στην οργανωμένη ανάπτυξη component-based αρχιτεκτονικής και στη βελτιστοποίηση της εμπειρίας χρήστη με τη χρήση βιβλιοθηκών όπως FullCalendar και SweetAlert2.

Μέσα από την ολοκλήρωση του έργου, απέκτησα πολύτιμη εμπειρία στη σχεδίαση, ανάλυση και υλοποίηση πληροφοριακών συστημάτων από το μηδέν, μαθαίνοντας πώς να εντοπίζω προβλήματα, να βρίσκω βέλτιστες λύσεις και να υλοποιώ επεκτάσιμες και φιλικές προς τον χρήστη εφαρμογές.

Η ενασχόληση με την ανάλυση και την υλοποίηση πραγματικών διαδικασιών με δίδαξε επίσης τη σημασία της ακρίβειας, της προσεκτικής τεκμηρίωσης και της προσαρμογής στις ανάγκες του τελικού χρήστη, στοιχεία απαραίτητα για την επιτυχία κάθε έργου πληροφορικής.

Συνολικά, η εμπειρία αυτή συνέβαλε ουσιαστικά στην προσωπική και επαγγελματική μου ανάπτυξη και με εφοδίασε με γνώσεις και δεξιότητες απαραίτητες για την ομαλή μετάβαση μου σε ένα επαγγελματικό περιβάλλον πληροφορικής.

Βιβλιογραφία

- [1] "React – A JavaScript library for building user interfaces", ReactJS Official Documentation, [Ηλεκτρονικό], Διαθέσιμο από: <https://react.dev/>
- [2] "React Router – Declarative routing for React", React Router Documentation, [Ηλεκτρονικό], Διαθέσιμο από: <https://reactrouter.com/en/main>
- [3] "Axios – Promise based HTTP client for the browser and Node.js", Axios Documentation, [Ηλεκτρονικό], Διαθέσιμο από: <https://axios-http.com/docs/intro>
- [4] "FullCalendar – JavaScript Event Calendar", FullCalendar Official Documentation, [Ηλεκτρονικό], Διαθέσιμο από: <https://fullcalendar.io/docs>
- [5] "Tailwind CSS – Rapidly build modern websites without ever leaving your HTML", TailwindCSS Documentation, [Ηλεκτρονικό], Διαθέσιμο από: <https://tailwindcss.com/docs>
- [6] "SweetAlert2 – A beautiful, responsive, customizable replacement for JavaScript's popup boxes", SweetAlert2 Documentation, [Ηλεκτρονικό], Διαθέσιμο από: <https://sweetalert2.github.io/>
- [7] "Lucide – Beautiful & consistent icon toolkit made by the community", Lucide-React Documentation, [Ηλεκτρονικό], Διαθέσιμο από: <https://lucide.dev/>
- [8] "MDN Web Docs – JavaScript Guide", MDN Contributors, 2024, [Ηλεκτρονικό], Διαθέσιμο από: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
- [9] "PHP Manual", The PHP Group, [Ηλεκτρονικό], Διαθέσιμο από: <https://www.php.net/manual/en/index.php>
- [10] "MySQL 8.0 Reference Manual", Oracle Corporation, [Ηλεκτρονικό], Διαθέσιμο από: <https://dev.mysql.com/doc/>

- [11] "10 Usability Heuristics for User Interface Design", Jakob Nielsen, 1994, [Ηλεκτρονικό], Διαθέσιμο από: <https://www.nngroup.com/articles/ten-usability-heuristics/>
- [12] "REST – Representational State Transfer", MDN Web Docs, Mozilla Contributors, [Ηλεκτρονικό], Διαθέσιμο από: <https://developer.mozilla.org/en-US/docs/Glossary/REST>
- [13] "Visual Studio Code – Code Editing. Redefined.", Microsoft Corporation, [Ηλεκτρονικό], Διαθέσιμο από: <https://code.visualstudio.com/>
- [14] "HTML: HyperText Markup Language", MDN Web Docs, Mozilla Contributors, [Ηλεκτρονικό], Διαθέσιμο από: <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [15] "CSS: Cascading Style Sheets", MDN Web Docs, Mozilla Contributors, [Ηλεκτρονικό], Διαθέσιμο από: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [16] "phpMyAdmin – A web interface for MySQL and MariaDB", phpMyAdmin Official Documentation, [Ηλεκτρονικό], Διαθέσιμο από: <https://www.phpmyadmin.net/docs/>
- [17] "React Toastify – React notification made easy", React Toastify Documentation, [Ηλεκτρονικό], Διαθέσιμο από: <https://fkhadra.github.io/react-toastify/introduction/>

Παράρτημα Α

Στο παράρτημα αυτό ακολουθείται το εγχειρίδιο χρήσης του συστήματος.

Σύνδεση

Για να εισέλθετε στο Σύστημα Καταγραφής Διαδικασιών, θα πρέπει να ακολουθήσετε τον σύνδεσμο που βρίσκεται στην πύλη του Τμήματος Πληροφορικής, στη σελίδα <https://portal.cs.ucy.ac.cy/> (Εικόνα 3, Ενότητα 3.1). Ο σύνδεσμος οδηγεί στη σελίδα σύνδεσης του Πανεπιστημίου (Εικόνα 4, Ενότητα 3.1), όπου πρέπει να εισάγετε τα ιδρυματικά σας στοιχεία.

1. Εισάγετε το όνομα χρήστη και τον κωδικό πρόσβασης που σας έχει δοθεί από το Πανεπιστήμιο.
2. Πατήστε το κουμπί «Sign in».
3. Εφόσον η σύνδεση είναι επιτυχής, θα αποκτήσετε πρόσβαση στην εφαρμογή.

Μενού Πλοήγησης

Το μενού πλοήγησης βρίσκεται στο αριστερό μέρος της σελίδας (Sidebar) και περιλαμβάνει συνδέσμους προς:

- Αρχική Σελίδα
- Όλες οι Διαδικασίες
- Προσωπικές Διαδικασίες
- Τρέχουσες Διαδικασίες
- Συμπληρωμένες Διαδικασίες
- Ημερολόγιο

Αρχική Σελίδα

Η Αρχική Σελίδα του Συστήματος Καταγραφής Διαδικασιών αποτελεί το κεντρικό σημείο ενημέρωσης και σύνοψης για τον χρήστη.

Δομή της Αρχικής Σελίδας:

Η σελίδα χωρίζεται σε τρεις βασικές ενότητες:

1. Ημερολόγιο

Το ημερολόγιο βρίσκεται στο αριστερό μέρος της κεντρικής σελίδας και παρουσιάζει τον τρέχοντα μήνα με δυνατότητα προβολής ημερολογιακών ημερομηνιών. Στο παρόν στάδιο, το ημερολόγιο είναι στατικό και χρησιμεύει κυρίως για οπτική αναφορά.

2. Όλες οι Διαδικασίες

Βρίσκεται στο κέντρο της οθόνης και παρουσιάζει όλες τις διαδικασίες που είναι διαθέσιμες προς έναρξη. Κάθε διαδικασία εμφανίζεται με τίτλο σε μορφή λίστας, ενώ παρέχεται δυνατότητα κύλισης (scroll) για την εμφάνιση περισσότερων διαδικασιών.

3. Προσωπικές Διαδικασίες

Βρίσκεται στη δεξιά πλευρά της οθόνης και παρουσιάζει συνοπτικά:

- Σύνολο Διαδικασιών: Ο συνολικός αριθμός διαδικασιών που ο χρήστης έχει εκκινήσει.
- Τρέχουσες Διαδικασίες: Ο αριθμός των διαδικασιών που βρίσκονται σε εξέλιξη.
- Συμπληρωμένες Διαδικασίες: Ο αριθμός των διαδικασιών που έχουν ολοκληρωθεί.

Χρήσεις της Αρχικής Σελίδας:

- Άμεση εποπτεία της προόδου στις προσωπικές διαδικασίες.
- Γρήγορη πρόσβαση στη λίστα όλων των διαθέσιμων διαδικασιών.
- Επισκόπηση ημερολογιακής πληροφόρησης.
- Βασικό σημείο εκκίνησης για πλοήγηση σε άλλες λειτουργικές ενότητες του συστήματος.

Σελίδα Όλες οι Διαδικασίες

Η σελίδα Όλες οι Διαδικασίες προσφέρει στους χρήστες τη δυνατότητα προβολής όλων των διαθέσιμων προς εκκίνηση διαδικασιών που υποστηρίζει το σύστημα.

Δομή της Σελίδας:

Η σελίδα χωρίζεται σε δύο βασικές ενότητες:

1. Λωρίδα Επιλογής Διαδικασιών

Βρίσκεται στο επάνω μέρος της σελίδας και παρουσιάζει με μορφή καρτελών (tabs) όλες τις διαθέσιμες διαδικασίες. Ο χρήστης μπορεί να περιηγηθεί ανάμεσα στις διαδικασίες με οριζόντια κύλιση (scroll), όταν οι διαθέσιμες διαδικασίες υπερβαίνουν το πλάτος της οθόνης.

2. Προβολή Περιεχομένου Διαδικασίας

Μόλις ο χρήστης επιλέξει μια διαδικασία από τη λωρίδα, εμφανίζεται το αναλυτικό περιεχόμενο της διαδικασίας.

Παρουσιάζονται πληροφορίες όπως:

- Αναλυτική περιγραφή της διαδικασίας.
- Σημαντικές οδηγίες.
- Προϋποθέσεις ή κριτήρια.
- Οδηγίες βημάτων ή επιμέρους ενεργειών που πρέπει να ακολουθήσει ο χρήστης.

Λειτουργικότητα:

Ο χρήστης μπορεί εύκολα και άμεσα να ενημερωθεί για το τι απαιτεί η κάθε διαδικασία πριν ξεκινήσει την εκτέλεσή της.

Η μορφοποίηση του κειμένου περιλαμβάνει:

- Επικεφαλίδες για διάκριση τμημάτων της διαδικασίας.
- Λίστες με κουκκίδες (bullet points) για ευκολότερη ανάγνωση.
- Έμφαση σε κρίσιμες πληροφορίες.

Χρήσεις της Σελίδας:

- Ενημέρωση και κατανόηση των απαιτήσεων πριν την έναρξη μιας διαδικασίας.
- Επιλογή της κατάλληλης διαδικασίας προς έναρξη.
- Εκπαίδευση του χρήστη πάνω στις εσωτερικές διαδικασίες του οργανισμού.
- Ανάκτηση απαραίτητων εγγράφων προς συμπλήρωση, με απώτερο σκοπό την ολοκλήρωση διαδικασιών.

Σελίδα Προσωπικές Διαδικασίες

Η σελίδα Προσωπικές Διαδικασίες αποτελεί το κεντρικό σημείο διαχείρισης των διαδικασιών που έχει ξεκινήσει, διαχειρίζεται ή έχει ολοκληρώσει ένας χρήστης. Ο σχεδιασμός της είναι απλός, ευανάγνωστος και προσανατολισμένος στη διευκόλυνση της καθημερινής χρήσης.

Στη σελίδα εμφανίζονται τα εξής κύρια τμήματα:

1. Έναρξη Νέας Διαδικασίας

Στο επάνω μέρος υπάρχει το κουμπί «Ξεκινήστε μια νέα Διαδικασία».

- Πατώντας το κουμπί, εμφανίζεται ένα αναδυόμενο παράθυρο στο οποίο ο χρήστης μπορεί να επιλέξει από λίστα ποια τυποποιημένη διαδικασία επιθυμεί να εκκινήσει.
- Η επιλογή επιβεβαιώνεται με το κουμπί «Εκκίνηση» ή μπορεί να ακυρωθεί με το κουμπί «Ακύρωση».

2. Λίστες Διαδικασιών (Με Τρεις Κατηγορίες)

Κάτω από το κουμπί εκκίνησης, η σελίδα χωρίζεται σε τρεις στήλες:

- Όλες οι Διαδικασίες: Περιλαμβάνει όλες τις διαθέσιμες διαδικασίες που μπορεί να ξεκινήσει ο χρήστης.
- Τρέχουσες Διαδικασίες: Δείχνει όλες τις διαδικασίες που έχουν ξεκινήσει αλλά δεν έχουν ακόμα ολοκληρωθεί.
- Συμπληρωμένες Διαδικασίες: Εμφανίζει όλες τις διαδικασίες που ο χρήστης έχει ολοκληρώσει.

Κάθε διαδικασία εμφανίζεται ως κουμπί/καρτέλα, το οποίο πατώντας το, οδηγεί τον χρήστη στη λεπτομερή προβολή και διαχείριση της αντίστοιχης διαδικασίας.

3. Ειδικές Λειτουργίες ανά Διαδικασία

- Για τρέχουσες διαδικασίες, ο χρήστης μπορεί να συνεχίσει την επεξεργασία ή την ολοκλήρωσή τους, με βάση τα διαθέσιμα βήματα και τις απαιτήσεις της κάθε διαδικασίας.
- Για συμπληρωμένες διαδικασίες, η πρόσβαση είναι μόνο για ανάγνωση. Δεν επιτρέπεται τροποποίηση των βημάτων ή νέες υποβολές, διασφαλίζοντας την ακεραιότητα των δεδομένων.

4. Αισθητική και Χρηστικότητα

- Η παρουσίαση είναι καθαρή και ομαδοποιημένη για εύκολη εύρεση.
- Τα κουμπιά είναι μεγάλα, με ευανάγνωστα κείμενα και χωρίζονται με ευδιάκριτα πλαίσια.
- Υπάρχει ομοιογένεια στα χρώματα και στο στυλ, διατηρώντας την ταυτότητα του συστήματος.

Σελίδα Συμπληρωμένες Διαδικασίες

Στη σελίδα Συμπληρωμένες Διαδικασίες του Συστήματος Καταγραφής Διαδικασιών παρουσιάζονται όλες οι διαδικασίες που έχουν ολοκληρωθεί από τον χρήστη.

Δομή Σελίδας:

- **Πλοήγηση Διαδικασιών:** Στο άνω μέρος της σελίδας εμφανίζονται σε μορφή κουμπιών όλες οι ολοκληρωμένες διαδικασίες. Κάθε κουμπί αντιστοιχεί σε μία διαφορετική διαδικασία και με την επιλογή του φορτώνονται οι λεπτομέρειες της.
- **Προβολή Ολοκληρωμένης Διαδικασίας:** Για κάθε ολοκληρωμένη διαδικασία παρουσιάζονται αναλυτικά:
 - **Τίτλος Διαδικασίας και Ημερομηνία Ολοκλήρωσης.**
 - **Βήματα Διαδικασίας:** Η διαδικασία αναλύεται σε επιμέρους βήματα, παρουσιασμένα με αύξοντα αριθμό και τίτλο. Κάθε βήμα περιλαμβάνει:
 - Περιγραφή της ενέργειας ή της αρμοδιότητας.
 - Προαιρετικά, σύνδεσμο για να εμφανιστούν ή να αποκρυφτούν περισσότερες λεπτομέρειες (κουμπιά "MORE DETAILS" και "LESS DETAILS").
 - Σύνδεσμο ή κουμπί "VIEW UPLOADED FILES" όπου είναι διαθέσιμα αρχεία που είχαν επισυναφθεί από τον χρήστη κατά την πορεία της διαδικασίας. Τα αρχεία ανοίγουν σε νέο παράθυρο.
- **Λειτουργικότητα:**

Οι χρήστες μπορούν να εξετάσουν αναδρομικά όλα τα στοιχεία των ολοκληρωμένων διαδικασιών τους. Δεν παρέχεται δυνατότητα τροποποίησης ή διαγραφής δεδομένων, καθώς οι διαδικασίες έχουν πλέον κλειδώσει σε μη επεξεργάσιμη μορφή για λόγους ασφάλειας και αρχειοθέτησης.

Σημαντικά Σημεία:

- **Διατήρηση Δεδομένων:** Τα αρχεία που είχαν υποβληθεί σε κάθε βήμα διατηρούνται διαθέσιμα για αναφορά ακόμα και μετά την ολοκλήρωση της διαδικασίας.
- **Χρονολόγηση:** Εμφανίζεται με σαφήνεια η ημερομηνία κατά την οποία ολοκληρώθηκε κάθε διαδικασία.
- **Εμπειρία Χρήστη:** Η διάκριση ανάμεσα σε τίτλους βημάτων, περιγραφές και αρχεία δημιουργεί μια οργανωμένη και εύκολη προς κατανόηση διεπαφή.

Σελίδα Ημερολόγιο

Η σελίδα Ημερολόγιο προσφέρει μια καθαρή και δυναμική επισκόπηση όλων των προγραμματισμένων γεγονότων και συμβάντων που αφορούν τον χρήστη. Είναι χτισμένη με

τη βοήθεια της βιβλιοθήκης FullCalendar, προσφέροντας μια σύγχρονη και διαδραστική εμπειρία χρήστη.

1. Κεντρική Εμφάνιση Ημερολογίου

- Το κύριο μέρος της σελίδας είναι αφιερωμένο σε ένα μηνιαίο ημερολόγιο, όπου κάθε μέρα εμφανίζεται ως ξεχωριστό κελί.
- Οι εκδηλώσεις και τα συμβάντα εμφανίζονται μέσα στις αντίστοιχες ημερομηνίες ως μπάρες με τίτλο, σε διακριτό χρώμα.
- Στην κορυφή εμφανίζεται πάντα ο τρέχων μήνας και έτος.

2. Πλοήγηση

- Ο χρήστης μπορεί να πλοηγηθεί μεταξύ μηνών με τα κουμπιά προηγούμενος και επόμενος μήνας.
- Υπάρχει επίσης κουμπί Σήμερα για άμεση επιστροφή στην τρέχουσα ημερομηνία.

3. Διαχείριση Συμβάντων

Το ημερολόγιο δεν είναι μόνο για απλή προβολή, αλλά προσφέρει διαδραστικότητα:

- Επιλογή Ημερομηνίας: Ο χρήστης μπορεί να κλικάρει σε μια κενή ημέρα για να δημιουργήσει νέο γεγονός.
- Επιλογή Συμβάντος: Κλικάροντας σε υπάρχον συμβάν, εμφανίζεται αναδυόμενο παράθυρο με τις λεπτομέρειες του γεγονότος (όπως τίτλος, περιγραφή κ.λπ.).
- Τα συμβάντα αφορούν:
 - Ολοκλήρωση ενεργειών ή διαδικασιών.
 - Προγραμματισμένες συναντήσεις και προσωπικά γεγονότα του χρήστη.

4. Εμφάνιση και Σχεδίαση

- Η αισθητική παραμένει απλή και καθαρή ώστε να διατηρείται η ευχρηστία.
- Κάθε συμβάν εμφανίζεται σε μπάρα πράσινου τόνου, διαφοροποιώντας το από τα απλά κελιά του ημερολογίου.
- Το ημερολόγιο είναι πλήρως ανταποκρίσιμο (responsive) ώστε να προσαρμόζεται σωστά σε όλες τις διαστάσεις οθόνης.

5. Εξυπηρέτηση Χρήστη

- Το ημερολόγιο ενισχύει την προσωπική οργάνωση του χρήστη, εμφανίζοντας σε συγκεντρωμένη μορφή όλες τις ενέργειες που τον αφορούν.
- Διευκολύνει τη διαχείριση χρόνου και την ενημέρωση για σημαντικά γεγονότα.

Παράρτημα Β

Στο παρόν παράρτημα παρουσιάζονται αναλυτικά όλες οι διαδικασίες που είναι καταχωρημένες στο «Σύστημα Καταγραφής Διαδικασιών», περιλαμβάνοντας τα βήματά τους, τους εμπλεκόμενους φορείς και τις σχετικές περιγραφές.

1. Εκλογή Ακαδημαϊκού Προσωπικού

Βήμα 1: Συμβούλιο Τμήματος

Το Συμβούλιο Τμήματος υποβάλλει κατάλογο στη Σύγκλητο για μέλη της ειδικής επιτροπής.

Βήμα 2: Σύγκλητος

Η Σύγκλητος διορίζει Ειδική Επιτροπή.

Βήμα 3: Πρόεδρος της Επιτροπής

Ο Πρόεδρος της Επιτροπής αποστέλλει πρόσκληση στα εξωτερικά μέλη της ειδικής Επιτροπής.

Βήμα 4: Πρόεδρος της Επιτροπής

Ο Πρόεδρος της Επιτροπής κοινοποιεί όλα τα αποδεικτικά στοιχεία των προσόντων των υποψηφίων στα μέλη της Επιτροπής εντός δύο εβδομάδων από την ημερομηνία λήξεως της προθεσμίας υποβολής αιτήσεων.

Βήμα 5: Μέλη της Ειδικής Επιτροπής

Κάθε μέλος της επιτροπής εξετάζει τους υποψηφίους και σημειώνει τους υποψηφίους που θα επιθυμούσαν να προσκληθούν σε συνέντευξη.

Βήμα 6: Πρόεδρος της Επιτροπής

Ο Πρόεδρος της Επιτροπής μέσα σε έξι εβδομάδες από την ημερομηνία λήξεως της προθεσμίας υποβολής των αιτήσεων ετοιμάζει τελικό κατάλογο των υποψηφίων που θα κληθούν σε συνέντευξη.

Βήμα 7: Μέλη της Ειδικής Επιτροπής

Τα μέλη της Ειδικής Επιτροπής αποφασίζουν την/ις ημερομηνία/ες που θα γίνουν οι συνεντεύξεις.

Βήμα 8: Πρόεδρος της Επιτροπής

Εντός δώδεκα εβδομάδων από την ημερομηνία λήξεως της υποβολής αιτήσεων ο Πρόεδρος της Επιτροπής καλεί σε συνέντευξη τους υποψηφίους οι οποίοι περιλαμβάνονται στον τελικό κατάλογο.

Βήμα 9: Μέλη της Ειδικής Επιτροπής

Με βάση τις συνεντεύξεις και μετά από συζήτηση, συντάσσουν αρκούντως αιτιολογημένη και τεκμηριωμένη έκθεση.

Βήμα 10: Πρόεδρος της Επιτροπής

Μέσα σε δεκαπέντε μέρες από τη συμπλήρωση των συνεντεύξεων ο Πρόεδρος της Επιτροπής αποστέλλει στο Συμβούλιο της Σχολής την έκθεση με εμπιστευτικό χαρακτήρα.

Βήμα 11: Κοσμήτορας

Ο Κοσμήτορας της οικείας σχολής καλεί σε συνεδρία το Εκλεκτορικό Σώμα μέσα σε τρεις εβδομάδες από την υποβολή στο Συμβούλιο της οικείας Σχολής της τεκμηριωμένης έκθεσης της Επιτροπής.

Βήμα 12: Εκλεκτορικό Σώμα

Το Εκλεκτορικό Σώμα εξετάζει τα στοιχεία των υποψηφίων και την έκθεση της επιτροπής και υποβάλλει στην Σύγκλητο αιτιολογημένη έκθεση εμπιστευτικής φύσης το αποτέλεσμα για την απόφαση εκλογής.

Βήμα 13: Υπηρεσία Ανθρώπινου Δυναμικού

Η Υπηρεσία Ανθρώπινου Δυναμικού αποστέλλει σε κάθε υποψήφιο αντίγραφο του μέρους της έκθεσης της Επιτροπής το οποίο αφορά τόσο τον ίδιο όσο και τον επιλεγέντα.

2. Ανέλιξη Ακαδημαϊκού Προσωπικού

Βήμα 1: Συμβούλιο Τμήματος

Το Συμβούλιο Τμήματος υποβάλλει κατάλογο στη Σύγκλητο για μέλη της ειδικής επιτροπής.

Βήμα 2: Σύγκλητος

Η Σύγκλητος διορίζει Ειδική Επιτροπή.

Βήμα 3: Πρόεδρος της Επιτροπής

Ο Πρόεδρος της Επιτροπής αποστέλλει πρόσκληση στα εξωτερικά μέλη της ειδικής Επιτροπής.

Βήμα 4: Υποψήφιος

Ο υποψήφιος υποβάλλει στον Πρόεδρο της Επιτροπής φάκελο αξιολόγησης, το αργότερο ένα μήνα μετά τον ορισμό της Ειδικής Επιτροπής.

Βήμα 5: Πρόεδρος της Ειδικής Επιτροπής

Ο Πρόεδρος της Ειδικής Επιτροπής κοινοποιεί στα υπόλοιπα μέλη της Επιτροπής ολόκληρο το φάκελο του υποψηφίου εντός δύο εβδομάδων από την ημέρα υποβολής του.

Βήμα 6: Μέλη της Ειδικής Επιτροπής

Τα μέλη της Ειδικής Επιτροπής επιλέγουν τρεις Ανεξάρτητους Κριτές, Καθηγητές πανεπιστημίων ή ερευνητές εγνωσμένου κύρους σε διεθνούς φήμης Ερευνητικά Ινστιτούτα (σε βαθμίδα ισότιμη του Καθηγητή) από τουλάχιστον δύο διαφορετικές χώρες του εξωτερικού, ειδικούς στο γνωστικό αντικείμενο του υποψηφίου.

Βήμα 7: Πρόεδρος της Ειδικής Επιτροπής

Ο Πρόεδρος της Ειδικής Επιτροπής κοινοποιεί το φάκελο του υποψηφίου σε κάθε Ανεξάρτητο Κριτή ζητώντας του να υποβάλει έκθεση αξιολόγησης που θα απαντά και σε συγκεκριμένα ερωτήματα. Ο φάκελος δεν περιλαμβάνει τις συστατικές επιστολές των τριών καθηγητών από τους οποίους ο υποψήφιος έχει ζητήσει την αποστολή συστατικών επιστολών.

Βήμα 8: Ανεξάρτητοι Κριτές

Οι γραπτές αξιολογήσεις των Ανεξάρτητων Κριτών υποβάλλονται στον Πρόεδρο της Ειδικής Επιτροπής εντός τριών μηνών από την παραλαβή του φακέλου του υποψηφίου. Εάν κάποιος από τους Ανεξάρτητους Κριτές αδυνατεί να ανταποκριθεί σε αυτό το αίτημα, τότε η Ειδική Επιτροπή τον αντικαθιστά.

Βήμα 9: Ειδική Επιτροπή

Η Ειδική Επιτροπή μελετά τον φάκελο του υποψηφίου, τις αξιολογήσεις των Ανεξάρτητων Κριτών και τις συστατικές επιστολές, και αποφασίζει αν ο υποψήφιος θα κληθεί σε ανοικτή διάλεξη και συνέντευξη, προκειμένου να κριθεί για ανέλιξη. Η απόφαση λαμβάνεται κατά πλειοψηφία.

Βήμα 10: Πρόεδρος της Ειδικής Επιτροπής

Ο Πρόεδρος καταθέτει στον Κοσμήτορα και στο Εκλεκτορικό Σώμα την τεκμηριωμένη πρόταση της Επιτροπής, επισυνάπτοντας τα απαιτούμενα δικαιολογητικά.

Βήμα 11: Κοσμήτορας

Ο Κοσμήτορας κοινοποιεί την έκθεση της Ειδικής Επιτροπής στον υποψήφιο.

Βήμα 12: Εκλεκτορικό Σώμα

Το Εκλεκτορικό Σώμα εξετάζει την έκθεση της Ειδικής Επιτροπής και τις παρατηρήσεις του υποψηφίου, λαμβάνει την απόφαση και υποβάλλει τεκμηριωμένη έκθεση στη Σύγκλητο για επικύρωση.

Βήμα 13: Σύγκλητος

Η Σύγκλητος εξετάζει την έκθεση του Εκλεκτορικού Σώματος και διαβιβάζει την απόφασή της στο Συμβούλιο.

Βήμα 14: Πρόεδρος Συμβουλίου

Η τελική απόφαση κοινοποιείται στον υποψήφιο από τον Πρόεδρο του Συμβουλίου, μαζί με την έκθεση του Εκλεκτορικού Σώματος.

3. Απονομή θέσης συνεργάτη Καθηγητή ΠΚ

Βήμα 1: Ορισμός Επιτροπής Αξιολόγησης

Βήμα 2: Το Συμβούλιο του Τμήματος

Το Συμβούλιο του Τμήματος υποβάλλει άρτια τεκμηριωμένες εισηγήσεις στην επιτροπή Αξιολόγησης για συγκεκριμένους υποψήφιους.

Βήμα 3: Επιτροπή Αξιολόγησης

Η Επιτροπή Αξιολόγησης εξετάζει τις εισηγήσεις του Συμβουλίου του Τμήματος και υποβάλλει τεκμηριωμένη γνωμάτευση προς τη Σύγκλητο.

Βήμα 4: Σύγκλητος

Η Σύγκλητος εξετάζει την γνωμάτευση της επιτροπής Αξιολόγησης και αποφασίζει σχετικά.

Ακολουθώς αποστέλλει την απόφασή της στο Συμβούλιο του Πανεπιστημίου για επικύρωση και ενημερώνει το Τμήμα.

4. Εκλογή Εκπροσώπου Τμήματος στο Συμβούλιο της ΣΘΕΕ

Βήμα 1: Πρόεδρος του Τμήματος

Ο Πρόεδρος του Τμήματος συγκαλεί γραπτώς ειδική συνεδρία του Συμβουλίου του Τμήματος κατά την οποία εκλέγονται οι δύο εκπρόσωποι του Τμήματος στο Συμβούλιο της Σχολής.

Βήμα 2: Μέλη Συμβουλίου

Κάθε μέλος του Συμβουλίου μπορεί κατά τη συνεδρία να προτείνει υποψηφίους, τα ονόματα των οποίων αναγράφονται σε πίνακα κατά αλφαβητική σειρά.

Βήμα 3: Μυστική ψηφοφορία

Ακολουθεί μυστική ψηφοφορία μεταξύ των υποψηφίων που πρότειναν τα μέλη του Συμβουλίου.

Βήμα 4: Διεξαγωγή Εκλογών

Η εκλογή διεξάγεται χωριστά για τον πρώτο και τον δεύτερο εκπρόσωπο.

Βήμα 5: Πρόεδρος του Τμήματος

Ο Πρόεδρος του Τμήματος ειδοποιεί αμέσως τον Κοσμήτορα της Σχολής για τα αποτελέσματα της εκλογής.

5. Απασχόληση Άμισθου Επισκέπτη Ερευνητή ή Μεταδιδακτορικού Ερευνητή

Βήμα 1: Υπεύθυνος Καθηγητής

Ετοιμασία σημειώματος από τον Υπεύθυνο Καθηγητή προς το Συμβούλιο Τμήματος, αναφορικά με το άτομο που επιθυμεί να απασχολήσει ως Άμισθο Ερευνητή. Το σημείωμα θα πρέπει να συνοδεύεται και από το Βιογραφικό του προτεινόμενου Ερευνητή.

Βήμα 2: Υπεύθυνος Καθηγητής / Ερευνητής

Συμπλήρωση «Εντύπου Απασχόλησης Άμισθων Επισκεπτών Ερευνητών» από τον υπεύθυνο Καθηγητή / Ερευνητή.

Βήμα 3: Γραμματεία του Τμήματος

Ετοιμασία σημειώματος από τη Γραμματεία του Τμήματος προς την Επιτροπή Προσωπικού, μέσω του Κοσμήτορα της οικείας Σχολής, με επισυναπτόμενα την Έγκριση του Συμβουλίου Τμήματος, το Βιογραφικό του Ερευνητή και το Έντυπο Απασχόλησης Άμισθων Επισκεπτών Ερευνητών.

6. Επισκέπτες Ακαδημαϊκοί

Βήμα 1: Προκήρυξη Θέσης

Το Συμβούλιο του Τμήματος προκηρύσσει θέση σε γνωστικά αντικείμενα.

Βήμα 2: Συμβούλιο του Τμήματος

Το Συμβούλιο του Τμήματος αφού εξετάσει τις αιτήσεις των υποψηφίων, υποβάλλει στην ΥΑΔ μέσω της οικείας Σχολής, εισήγηση Επισκέπτη Ακαδημαϊκού για κάλυψη της θέσης.

Βήμα 3: Σύγκλητος

Ο Επισκέπτης Ακαδημαϊκός εγκρίνεται από τη Σύγκλητο.

Βήμα 4: Συμβούλιο του Πανεπιστημίου

Ο Επισκέπτης Ακαδημαϊκός διορίζεται από το Συμβούλιο του Πανεπιστημίου.

7. Ειδικοί Επιστήμονες Διδασκαλίας

Βήμα 1: Συμβούλιο του Τμήματος

Το Προεδρείο του Τμήματος υποβάλλει αίτημα για θέσεις ειδικών επιστημόνων στην επιτροπή οικονομικών και προσωπικού.

Βήμα 2: Συμβούλιο του Τμήματος

Το συμβούλιο του τμήματος συντάσσει Ανακοίνωση Εκδήλωσης Ενδιαφέροντος για τις θέσεις που της έχουν κατανεμηθεί.

Βήμα 3: Συμβούλιο του Τμήματος

Το συμβούλιο του τμήματος, πριν τη λήξη της προθεσμίας υποβολής αιτήσεων, ορίζει Επιτροπή Αξιολόγησης (ΕΠΣ).

Βήμα 4: Επιτροπή Αξιολόγησης

Καλεί σε συνέντευξη τουλάχιστον τους τρεις καλύτερους ενδιαφερόμενους που πληρούν τα προσόντα για κάθε θέση, ή λιγότερους μόνο στην περίπτωση που δεν υπάρχουν τουλάχιστον τρεις ενδιαφερόμενοι που να πληρούν τα κριτήρια για τη συγκεκριμένη θέση. Τυχόν απόφαση για μη πρόσκληση ενδιαφερομένων σε συνέντευξη πρέπει να αιτιολογείται επαρκώς.

Βήμα 5: Επιτροπή Αξιολόγησης

Η Επιτροπή Αξιολόγησης ετοιμάζει κατάλογο ανάθεσης δεδομένου φόρτου επιλογής υποψηφίων.

Βήμα 6: Συμβούλιο του Τμήματος

Το Συμβούλιο Τμήματος ενημερώνει γραπτώς την Υπηρεσία Ανθρώπινου Δυναμικού του Πανεπιστημίου Κύπρου για την τελική επιλογή επισυνάπτοντας τον Κατάλογο Επιλογής Υποψηφίων.

Βήμα 7: Υπηρεσία Ανθρώπινου Δυναμικού

Η Υπηρεσία Ανθρώπινου Δυναμικού προωθεί τα σχετικά στοιχεία στο Συμβούλιο του Πανεπιστημίου Κύπρου ή στην εξουσιοδοτημένη Επιτροπή του για επικύρωση του διορισμού των επιλεγέντων.

Βήμα 8: Υπηρεσία Ανθρώπινου Δυναμικού

Η Υπηρεσία Ανθρώπινου Δυναμικού είναι αρμόδια για την ενημέρωση των υποψηφίων και την ετοιμασία και υπογραφή των σχετικών συμβολαίων εργοδότησης.

8. Διαδικασία Αξιολόγησης Διατριβής Master

Βήμα 1: Πρόεδρος του Τμήματος

Ο Πρόεδρος του Τμήματος συγκροτεί τριμελή Εξεταστική Επιτροπή με Πρόεδρο αυτής τον Ερευνητικό Σύμβουλο του Μεταπτυχιακού φοιτητή.

Βήμα 2: Πρόεδρος της Επιτροπής

Ο Πρόεδρος της Επιτροπής είναι υπεύθυνος για την προφορική παρουσίαση της Διατριβής Μάστερ ενώπιον της Εξεταστικής Επιτροπής.

Βήμα 3: Εξεταστική Επιτροπή

Η Εξεταστική Επιτροπή υποβάλλει την εισήγηση της για την αποδοχή ή απόρριψη της Διατριβής Μάστερ προς στο Τμήμα, σε ειδικό έντυπο Αξιολόγησης.

Βήμα 4: Συμβούλιο του Τμήματος

Το Συμβούλιο του Τμήματος εξετάζει την εισήγηση της Εξεταστικής Επιτροπής και προωθεί τη θετική απόφασή του προς τη Σχολή Μεταπτυχιακών Σπουδών για την απονομή του σχετικού τίτλου.

9. Διαδικασία Περιεκτικής Εξέτασης Διδακτορικού Φοιτητή

Βήμα 1: Συμβούλιο του Τμήματος

Το Συμβούλιο του Τμήματος ορίζει Ερευνητικό Σύμβουλο.

Βήμα 2: Επιτροπή Μεταπτυχιακών Σπουδών του Τμήματος

Η Επιτροπή Μεταπτυχιακών Σπουδών του Τμήματος, μετά από αίτημα του Ερευνητικού Συμβούλου του φοιτητή, ορίζει τριμελή Επιτροπή Περιεκτικής εξέτασης η οποία και έχει την ευθύνη για την διεξαγωγή της Περιεκτικής Εξέτασης.

Βήμα 3: Συμβούλιο του Τμήματος

Επικυρώνει τη σύνθεση της επιτροπής.

Βήμα 4: Επιτροπή Περιεκτικής Εξέτασης

Η Επιτροπή Περιεκτικής Εξέτασης εξετάζει την υποβολή γραπτού δοκιμίου του φοιτητή την οποία και αξιολογεί με αποτέλεσμα Επιτυχία/Αποτυχία.

Βήμα 5: Επιτροπή Περιεκτικής Εξέτασης

Η Επιτροπή Περιεκτικής Εξέτασης εξετάζει την, ανοικτή στο κοινό, προφορική παρουσίαση του δοκιμίου του φοιτητή.

Βήμα 6: Επιτροπή Περιεκτικής Εξέτασης

Η Επιτροπή Περιεκτικής Εξέτασης εξετάζει την, κλειστή στο κοινό, προφορική παρουσίαση του δοκιμίου του φοιτητή.

Βήμα 7: Επιτροπή Περιεκτικής Εξέτασης

Η Επιτροπή Περιεκτικής Εξέτασης καταθέτει στο Συμβούλιο του Τμήματος έκθεση με το δοκίμιο και το πόρισμα της Περιεκτικής Εξέτασης προς επικύρωση.

Βήμα 8: Συμβούλιο του Τμήματος

Το Συμβούλιο του Τμήματος προωθεί την έκθεση με το δοκίμιο και το πόρισμα της Περιεκτικής Εξέτασης στη Σχολή Μεταπτυχιακών Σπουδών.

10. Διαδικασία Ερευνητικής Πρότασης Διδακτορικού Φοιτητή

Βήμα 1: Επιτροπή Μεταπτυχιακών Σπουδών του Τμήματος

Η Επιτροπή Μεταπτυχιακών Σπουδών του Τμήματος, μετά από αίτημα του Ερευνητικού Συμβούλου του φοιτητή, ορίζει τριμελή Επιτροπή Ερευνητικής Πρότασης (Ερευνητική Επιτροπή).

Βήμα 2: Συμβούλιο του Τμήματος

Επικυρώνει τη σύνθεση της επιτροπής.

Βήμα 3: Υποψήφιος Διδάκτορας

Κάθε Υποψήφιος Διδάκτορας υποβάλλει γραπτώς δομημένη Πρόταση για Διδακτορική Διατριβή στην Ερευνητική του Επιτροπή.

Βήμα 4: Ερευνητική Επιτροπή

Η Ερευνητική Επιτροπή εξετάζει την Πρόταση του υποψηφίου Διδάκτορα και, μετά από προφορική παρουσίαση και εξέταση (παρόμοια με Περιεκτική Εξέταση), συντάσσει αναφορά.

Βήμα 5: Ερευνητική Επιτροπή

Η Ερευνητική Επιτροπή καταθέτει στο Συμβούλιο του Τμήματος την αναφορά μαζί με την Ερευνητική Πρόταση προς επικύρωση.

Βήμα 6: Συμβούλιο του Τμήματος

Το Συμβούλιο του Τμήματος προωθεί την αναφορά μαζί με την Ερευνητική Πρόταση στη Σχολή Μεταπτυχιακών Σπουδών.

11. Διαδικασία Διδακτορικής Διατριβής

Βήμα 1: Υποψήφιοι Διδάκτορες

Κάθε Υποψήφιος Διδάκτορας εκπονεί πρωτότυπη Διδακτορική Διατριβή, η οποία αποτελεί σημαντική προσφορά στο οικείο επιστημονικό πεδίο.

Βήμα 2: Επιτροπή Μεταπτυχιακών Σπουδών

Η Επιτροπή Μεταπτυχιακών Σπουδών, με εισήγηση του Ερευνητικού Συμβούλου, ορίζει 5-μελή Εξεταστική Επιτροπή και μεριμνά για την αποστολή της διατριβής στα μέλη της Εξεταστικής Επιτροπής..

Βήμα 3: Πρόεδρος της Εξεταστικής Επιτροπής

Ο Πρόεδρος της Εξεταστικής Επιτροπής ορίζει την ημερομηνία υποστήριξης της Διδακτορικής Διατριβής.

Βήμα 4: Εξεταστική Επιτροπή

Η Εξεταστική Επιτροπή υποβάλλει προς τον Πρόεδρο του Τμήματος γραπτή Έκθεση Αξιολόγησης της Διδακτορικής Διατριβής, και του Υποψηφίου Διδάκτορα γενικότερα, μαζί με την εισήγησή της.

Βήμα 5: Πρόεδρος του Τμήματος

Ο Πρόεδρος του Τμήματος διαπιστώνει τη νομιμότητα της όλης διαδικασίας και υποβάλλει την εισήγηση προς τη Σχολή Μεταπτυχιακής Εκπαίδευσης για τα περαιτέρω.

12. Προκήρυξη και Προσφορά Θέσεων στα Μεταπτυχιακά Προγράμματα

Βήμα 1: Σχολή Μεταπτυχιακών Σπουδών

Κάνει προκήρυξη θέσεων για τα Μεταπτυχιακά προγράμματα.

Βήμα 2: Επιτροπή Μεταπτυχιακών Προγραμμάτων του Τμήματος

Οι αιτήσεις υποβάλλονται ηλεκτρονικά στο σύστημα Μεταπτυχιακών Αιτήσεων της Σχολής Μεταπτυχιακών Σπουδών και μελετώνται από την Επιτροπή Μεταπτυχιακών Προγραμμάτων του Τμήματος.

Βήμα 3: Επιτροπή Μεταπτυχιακών Προγραμμάτων του Τμήματος

Οι εισηγήσεις της Επιτροπής Μεταπτυχιακών Προγραμμάτων υποβάλλονται στο Συμβούλιο του Τμήματος για έγκριση.

Βήμα 4: Πρόεδρος του Τμήματος

Ο Πρόεδρος του Τμήματος ενημερώνει τη Σχολή Μεταπτυχιακών Σπουδών για τους φοιτητές οι οποίοι έγιναν δεκτοί.

Βήμα 5: Επιτροπή Μεταπτυχιακών Προγραμμάτων του Τμήματος

Οι εγκεκριμένες Εισηγήσεις ΕΜΣ του Τμήματος καταχωρούνται, μέσω του συστήματος ΕΡΜΗΣ, στο Σύστημα Μεταπτυχιακών Αιτήσεων.

13. Χορήγηση Σαββατικής Άδειας

Βήμα 1: Το Μέλος ΔΕΠ που θέλει ΣΑ

Το μέλος ΔΕΠ που επιθυμεί να λάβει Σαββατική Άδεια, συμπληρώνει τη σχετική αίτηση και την υποβάλλει στο Συμβούλιο του Τμήματος.

Βήμα 2: Συμβούλιο Τμήματος

Το Συμβούλιο του Τμήματος εξετάζει το αίτημα για τη χορήγηση της Σαββατικής Άδειας.

Βήμα 3: Συμβούλιο Τμήματος

Αν εγκριθεί το αίτημα, τότε το Συμβούλιο του Τμήματος αποστέλνει σχετικό σημείωμα μέσω του Κοσμήτορα της οικείας Σχολής στον Πρύτανη του ΠΚ.

Βήμα 4: Το Μέλος ΔΕΠ που θέλει ΣΑ

Μετά το πέρας της Σαββατικής Άδειας, το μέλος ΔΕΠ υποβάλλει έκθεση πεπραγμένων στο Συμβούλιο του Τμήματος.

Βήμα 5: Συμβούλιο Τμήματος

Το Συμβούλιο του Τμήματος, αφού λάβει γνώση, προωθεί την έκθεση μέσω του Κοσμήτορα της οικείας Σχολής στην Επιτροπή Έρευνας, μαζί με το αίτημά του για παραχώρηση Σαββατικής Άδειας (αρχικό έντυπο).