



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

Σχολή Μηχανικών

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

ΜΠΟΓΚΝΤΑΝ ΒΑΣΚΑΝ

A.M 711161181

Επιβλέπων Καθηγητής

Φοίβος Μυλωνάς, Αναπληρωτής Καθηγητής

Αθήνα-Αιγάλεω, Σεπτέμβριος 2024

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
Σχολή Μηχανικών
Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

Όνοματεπώνυμο
Αριθμός Μητρώου
Επιβλέπων Καθηγητής

Μπογκντάν Βασκάν
711161181
Φοίβος Μυλωνάς, Αναπληρωτής Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή 24.09.2024

--	--	--

Αθήνα-Αιγάλεω, Σεπτέμβριος 2024

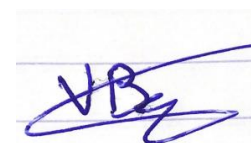
Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος Βασκάν Μπογκντάν του Ιβάν, με αριθμό μητρώου 711161181 φοιτητής του Πανεπιστημίου Δυτικής Αττικής της Σχολής Μηχανικών του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών, δηλώνω υπεύθυνα ότι:

Είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Επίσης, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος. Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου.

Ο Δηλών,
Μπογκντάν Βασκάν



Ευχαριστίες

Με την ολοκλήρωση της παρούσας διπλωματικής εργασίας και επομένως ολοκλήρωση των σπουδών μου θα ήθελα να ευχαριστήσω όλους αυτούς που με υποστήριξαν κατά την διάρκεια των σπουδών μου. Θα ήθελα θερμά να ευχαριστήσω τον επιβλέποντα καθηγητή τον κ. Φοίβο Μυλωνά για την συνεχή στήριξη κατά την διάρκειας της εκπόνησης της διπλωματικής μου εργασίας.

Επιπλέον ευχαριστώ όλους τους καθηγητές του πανεπιστημίου για όλα τα μαθήματα που πήρα από αυτούς και από την πανεπιστημιακή κοινότητα.

Ειδικά, θα ήθελα να ευχαριστήσω την οικογένεια μου η οποία ήταν πάντα δίπλα μου σε κάθε βήμα κατά την διάρκεια των σπουδών μου στο Πανεπιστήμιο Δυτικής Αττικής.

ΠΕΡΙΛΗΨΗ

Αυτή η εργασία επικεντρώνεται στη μελέτη και την υλοποίηση μιας σύγχρονης ψηφιακής βιβλιοθήκης, αξιοποιώντας τις τελευταίες τεχνολογίες και βέλτιστες πρακτικές στον τομέα της ανάπτυξης λογισμικού. Στόχος είναι η δημιουργία ενός εύελικτου, αποδοτικού και φιλικού προς τον χρήστη συστήματος, το οποίο θα επιτρέπει την εύκολη πρόσβαση, διαχείριση και αναζήτηση βιβλιογραφικού υλικού. Η πλατφόρμα ανάπτυξης που χρησιμοποιείται είναι το Laravel για το backend, ένα δημοφιλές PHP framework που προσφέρει δομή και ευελιξία στον κώδικα. Το Laravel παρέχει επίσης προηγμένα χαρακτηριστικά όπως το Eloquent ORM για τη διαχείριση των βάσεων δεδομένων. Για την αποθήκευση και διαχείριση των δεδομένων, χρησιμοποιείται το MySQL, ένα αξιόπιστο σύστημα διαχείρισης βάσεων δεδομένων που εξασφαλίζει την ταχύτητα και την ασφάλεια των δεδομένων. Το frontend της εφαρμογής αναπτύχθηκε με το Vue.js, ένα ισχυρό JavaScript framework που επιτρέπει τη δημιουργία δυναμικών διεπαφών χρήστη. Το Bootstrap χρησιμοποιείται για να διευκολύνει τον σχεδιασμό των σελίδων καθώς και κάποια έτοιμα components από αυτό. Η αρχιτεκτονική MVC που χρησιμοποιεί το Laravel βοηθάει στον διαχωρισμό της λογικής της εφαρμογής από την παρουσίαση, με αποτέλεσμα να είναι εύκολη η συντήρηση και επέκταση της εφαρμογής. Το Filament χρησιμοποιείται για πιο εύκολη και γρηγορότερη ανάπτυξη Admin Panel, που με λίγες μόλις προσαρμογές έχει έτοιμο το CRUD. Επίσης για να μεταφράσεις του frontend χρησιμοποιείται i18n και συγκεκριμένα το πακέτο vue-i18n.

Το τελικό αποτέλεσμα μας προσφέρει την αναζήτηση και φιλτράρισμα των δημοσιεύσεων με διάφορα κριτήρια όπως: Συγγραφέας, τίτλο της δημοσίευσης, περίληψη, εκδότης, Ημερομηνία έκδοσης από και έως, υπάρχει δυνατότητα φιλτραρίσματος επίσης μέσα στο κείμενο της δημοσίευσης καθώς και ανά τύπο της δημοσίευσης (Βιβλίο, περιοδικό, συνέδριο, Διπλωματική εργασία ή Διδακτορική Διατριβή), σχετικές πληροφορίες για το ίδρυμα, Έργα τα οποία φιλτράρονται ανά κατηγορία με χρήση tabs, τα μαθήματα και χρήσιμους σύνδεσμούς. Επιπλέον ο διαχειριστής της εφαρμογής μπορεί να ανεβάζει καινούργιες δημοσιεύσεις, έργα, μαθήματα καθώς και στην σελίδα "Προσωπικό" το περιεχόμενο είναι επίσης δυναμικό.

ABSTRACT

This thesis focuses on the design and implementation of a modern digital library, using the latest technologies and best practices in software development. The goal is to create a flexible, efficient, and user-friendly system that allows easy access, management and search of bibliographic material. The development platform used is Laravel for the backend, a popular PHP framework that offers structure and flexibility in the code. Laravel also provides advanced features such as Eloquent ORM for database management. For data storage and management, MySQL is used, a reliable database management system that ensures speed and security of data. The frontend of the application is developed with Vue.js, a powerful JavaScript framework that allows the creation of dynamic user interfaces. Bootstrap is used to facilitate the design of the pages as well as some ready-made components from it. The MVC architecture used by Laravel helps to separate the application logic from the view, making it easy to maintain and extend the application. Filament is used for easier and faster development of an admin Panel, which with just a few adjustments you have the CRUD ready. In addition, to translate frontend i18n is used and specifically the package vue-i18n.

The final result offers us the search and filtering of publications with various criteria such as: Author, title of publication, abstract, publisher, date of publication from and to, there is also the possibility of filtering within the text of the publication as well as by type of publication (Book, Journal, Conference, Diploma Thesis or PhD Thesis), relevant information about the institution, Projects are filtered by category using tabs, courses and useful links. In addition, the application administrator can upload new publications, projects, courses as well as on the "Staff" page the content is also dynamic.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Ψιφιακή βιβλιοθήκη, fullstack app, frontend, backend, Laravel, Filament, MySQL, Vue.js, bootstrap, i18n, REST API, MVC, Php.

Πίνακας Περιεχομένων

ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ	5
Ευχαριστίες	7
ΠΕΡΙΛΗΨΗ	9
ABSTRACT	9
Πίνακας Περιεχομένων.....	11
Κατάλογος Πινάκων	17
Κατάλογος Εικόνων	18
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ.....	19
ΚΕΦΑΛΑΙΟ 1	20
Εισαγωγή.....	20
1.1 Τι είναι η μια ψηφιακή βιβλιοθήκη;	20
1.2 Ιστορική ανασκόπηση.....	20
1.3 Πέντε νόμοι της βιβλιοθηκονομίας με ψηφιακή βιβλιοθήκη	21
1.4 Αρχές της ψηφιακής βιβλιοθήκης.....	21
1.5 Επιπτώσεις της πανδημίας Covid-19 και αλλαγές στον χώρο των ψηφιακών βιβλιοθηκών .	22
1.6 Η εξέλιξη και το μέλλον των ψηφιακών βιβλιοθηκών	23
1.7 Επίδραση των ψηφιακών πηγών της βιβλιοθήκης στην ανάπτυξη των φοιτητών της τριτοβάθμιας εκπαίδευσης	24
ΚΕΦΑΛΑΙΟ 2	25
Εισαγωγή.....	25
2.1 Vue.js.....	25
2.1.1 Ιστορική αναδρομή	25
2.1.2 Πεδίο εφαρμογής.....	25
2.1.3 Τα Βασικά πλεονεκτήματα του Vue.js.....	25
2.2.4 Τα μειονεκτήματα του Vue.js	25
Bootstrap	26
2.2 Ιστορική Αναδρομή του Bootstrap	26
2.2.1 Εφαρμογή.....	26
2.2.2 Τα πλεονεκτήματα	27
MVC.....	27
2.3 Περιγραφή.....	27
2.3.1 Τα τρία συστατικά του MVC.....	27
2.3.2 Απεικόνιση του πρότυπου	28

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

Laravel.....	28
2.4.2 Κύρια χαρακτηριστικά.....	28
2.4.2.1. Eloquent ORM.....	28
2.4.2.2 Artisan Console.....	29
2.4.2.3 Routing.....	29
2.4.2.4 Middleware.....	29
2.4.2.5 Security.....	29
2.4.2.6 Testing.....	29
2.4.2.7 Package Ecosystem.....	29
2.4.3 Σύνοψη.....	29
2.5 Filament.....	29
2.5.1 Περιγραφή.....	29
2.5.2 Τα πλεονεκτήματα του Filament.....	30
MySQL.....	30
2.6.1 Ιστορική αναδρομή.....	30
2.6.2 Περιγραφή.....	30
2.6.3 Τα βασικά χαρακτηριστικά του MySQL.....	30
2.7 Frontend Περιγραφή.....	30
2.8 Backend Περιγραφή.....	31
2.9. RestAPI.....	31
2.9.1 Ιστορική αναδρομή.....	31
2.9.2 Περιγραφή.....	31
2.9.3 Βασικές αρχές.....	31
2.10 i18n - μεταφράσεις.....	31
2.10.1 Ιστορική αναδρομή.....	31
2.10.2 Περιγραφή.....	32
1.10.3 Οι κύριες λειτουργίες.....	32
1.10.4 Επιπτώσεις και Μέλλον.....	32
1.11 Συμπέρασμα.....	32
ΚΕΦΑΛΑΙΟ 3.....	33
Εισαγωγή.....	33
3.1 Βάση δεδομένων.....	33
3.2 Το βασικό σχήμα της βάσης δεδομένων.....	33
3.2.1 Πίνακας projects.....	34

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

3.2.2 Πίνακας people	36
3.2.3 Πίνακας Course	37
3.2.4 Πίνακας Publication	37
3.2.5 Πίνακας file	38
3.2.6. Πίκας author_publication	38
3.2.7. Πίκας users.....	38
3.3 Default πίνακες του Laravel	39
3.3.1 Το σχήμα	40
3.3.2 Πίνακας failed_jobs	41
3.3.3 Πίνακας personal_access_tokens	41
3.3.4 Πίνακας sessions.....	42
3.4.5 Πίνακας migrations	42
3.4.6 Πίνακας password_reset_tokens.....	42
3.4.7 Πίνακας jobs	42
3.4.8 Πίνακες Cache & cache_locks.....	43
3.4.9 job_batches	43
3.5 Ανάπτυξη backend	44
3.5.1 Docker - περιγραφή	44
3.5.2 Docker Βασικά χαρακτηριστικά	44
3.5.3 Docker Compose - περιγραφή	44
3.5.4 Docker Compose - Κύρια χαρακτηριστικά	44
3.6 Laravel migrations	44
3.6.1 Laravel migrations περιγραφή	45
3.6.2 Laravel migrations πως λειτουργούν	45
3.6.3 Laravel migrations - Η Μέθοδος up	45
3.6.4 Laravel migrations - Η Μέθοδος down.....	45
3.6.5 Laravel migrations - Παραδείγματα	45
3.6.6 Εκτέλεση των Migrations	46
3.6.7 Αναίρεση των Migrations	46
3.6.8 Laravel migrations - Τα πλεονεκτήματα.....	46
3.6.9 Laravel migrations - Συμπέρασμα	46
3.7 Laravel Models	46
ΚΕΦΑΛΑΙΟ 4	48
Frontend	48

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

4.1.1 Σχετικά με μας	48
4.1.2 Έργα - Λίστα.....	49
4.1.3 Έργο.....	50
4.1.4 Δημοσιεύσεις.....	51
4.1.5 Δημοσίευση (Περίληψη).....	52
4.1.6 Μαθήματα.....	53
4.1.7 Προσωπικό.....	54
4.1,8 Σύνδεσμοι.....	55
Backend (Admin Panel)	55
Εισαγωγή	55
4.2.1. Έργα - λίστα	56
4.2.2 Έργα - προσθήκη καινούργιου	57
4.2.3 Έργα - επεξεργασία.....	58
4.2.4 Δημοσιεύσεις.....	59
4.2.5 Δημοσιεύσεις Δημιουργία/Επεξεργασία.....	60
4.2.6 Μαθήματα.....	61
4.2.7 Μαθήματα - δημιουργία / επεξεργασία	62
4.2.8 Προσωπικό.....	63
ΚΕΦΑΛΑΙΟ 5	64
Εισαγωγή.....	64
5.1 Μελλοντικές επεκτάσεις	64
5.2 Προβλήματα κατά την ανάπτυξη της εφαρμογής.....	64
Βιβλιογραφία.....	65
Χρήσιμοι σύνδεσμοι.....	65
Παράρτημα A (frontend).....	66
Main.js	66
Router.js	66
App.vue	67
l18n.js.....	68
pages/AboutPage.vue	69
CoursesPage.vue	70
LinksPage.vue	72
NotFound.vue	74
PeoplePage.vue	74

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

ProjectsPage.vue.....	76
PublicationsPage.vue	80
TheFooter.vue	82
TheHeader.vue.....	82
ProjectsComponent.vue	87
ProjectView.vue	89
PublicationFilters.vue.....	96
ViewAbstract.vue	99
CoursesList.vue	101
PeopleList.vue	103
TabsComponent.vue	104
Src/locales/gr.json	106
Παράρτημα Β (backend)	108
Models	108
PublicationSearch.php.....	108
Course.php.....	109
File.php.....	111
People.php	112
Project.php	113
Publication.php.....	116
Controllers/Api	119
CoursesController.php.....	119
PeopleController.php.....	120
ProjectsController.php.....	122
PublicationsController.php.....	124
Resources	126
CourseResource.php.....	126
PeopleResource.php	127
ProjectResource.php	127
PublicationResource.php.....	129
Filament Resources.....	130
App/Filament/Resources/CourseResource.php	130
App/Filament/Resources/PeopleResource.php	133
App/Filament/Resources/ProjectResource.php.....	135

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

App/Filament/Resources/PublicationResource.php	139
App/Filament/Resources/PublicationResource/Pages/CreatePublication.php	143
App/Filament/Resources/PublicationResource/Pages/EditPublication.php.....	145
Routes/api.php	146
Migrations	147
database/migrations/0001_01_01_000000_create_users_table.php	147
database/migrations/0001_01_01_000001_create_cache_table.php	149
database/migrations/0001_01_01_000002_create_jobs_table.php	150
database/migrations/2024_05_12_100343_init.php	152
database/migrations/2024_05_19_090156_create_author_publication_table.php.....	154
database/migrations/2024_06_15_134435_add_type_column_to_people_table.php.....	156
database/migrations/2024_06_23_100945_add_text_content_to_file_table.php.....	157
database/migrations/2024_07_07_100700_add_description_column_to_course_table.php	157

Κατάλογος Πινάκων

Πίνακας 1 – Projects	34
Πίνακας 2 People	36
Πίνακας 3 Course	37
Πίνακας 4 Publication	37
Πίνακας 5 File.....	38
Πίνακας 6 Author_publication	38
Πίνακας 7 Users	39
Πίνακας 8 Failed_jobs	41
Πίνακας 9 Personal_access_tokens.....	41
Πίνακας 10 Sessions	42
Πίνακας 11 Migrations.....	42
Πίνακας 12 Password_reset_tokens	42
Πίνακας 13 Jobs	43
Πίνακας 14 Cache_locks	43
Πίνακας 15 Cache	43
Πίνακας 16 Job_batches	44

Κατάλογος Εικόνων

Εικόνα 1 MVC	28
Εικόνα 2 Το βασικό σχήμα της βάσης δεδομένων	33
Εικόνα 3 Model: Timestamp.....	34
Εικόνα 4 Models:constants	35
Εικόνα 5 Models: Μέθοδοι.....	35
Εικόνα 6 Models: getTypes μέθοδος.....	36
Εικόνα 7 Default πίνακες του Laravel.....	40
Εικόνα 8 Migrations: μέθοδος up().....	45
Εικόνα 9 Migrations: μέθοδος down()	45
Εικόνα 10 Models: ένα παράδειγμα	47
Εικόνα 11 Frontend - Σχετικά με μας.....	48
Εικόνα 12 Frontend - Λίστα με έργα.....	49
Εικόνα 13 Frontend - Το έργο	50
Εικόνα 14 Frontend - Λίστα δημοσιεύσεων	51
Εικόνα 15 Frontend - Δημοσίευση - Περίληψη	52
Εικόνα 16 Frontend - Λίστα μαθημάτων	53
Εικόνα 17 Frontend - Προσωπικό.....	54
Εικόνα 18 Frontend - Σύνδεσμοι	55
Εικόνα 19 Backend - Λίστα έργων	56
Εικόνα 20 Backend - Προσθήκη έργου	57
Εικόνα 21 Backend - Επεξεργασία έργου	58
Εικόνα 22 Backend - λίστα δημοσιεύσεων	59
Εικόνα 23 Backend - Δημιουργία δημοσίευσης.....	60
Εικόνα 24 Backend - Λίστα μαθημάτων.....	61
Εικόνα 25 Backend - Δημιουργία μαθήματος	62
Εικόνα 26 Backend - Προσωπικό	63

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

MVC: Model-View-Controller

SPA: Single Page Applications

HTTP: Hypertext Transfer Protocol

CORS: Cross-origin resource sharing

CRUD: Create, read, update and delete

PK: Primary key

FK: Foreign key

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Ο σκοπός αυτού του κεφαλαίου είναι να γίνει μια ανασκόπηση θεωρητική και ιστορική σχετικά με τις ψηφιακές βιβλιοθήκες και την εξέλιξη τους.

1.1 Τι είναι η μια ψηφιακή βιβλιοθήκη;

Η ψηφιακή βιβλιοθήκη είναι μια οντότητα που επικεντρώνεται στη δημιουργία και διανομή πληροφοριών σε παγκόσμια δίκτυα, παρέχοντας σχετικό περιεχόμενο στους χρήστες. Μπορεί να περιγράψει ως μια συλλογή καταναμημένων, αυτόνομων δικτυακών τόπων που συνεργάζονται για να παρουσιάσουν μια ενιαία συλλογή στον χρήστη. Αυτοί οι ιστότοποι αποθηκεύουν τεράστιες ποσότητες πληροφοριών σε διάφορες μορφές και μέσα, απευθυνόμενοι σε χρήστες με διαφορετικά επίπεδα γνώσεων πληροφορικής, δεξιοτήτων πλοήγησης και γνώσεων τομέα.

Τα βασικά χαρακτηριστικά των ψηφιακών βιβλιοθηκών περιλαμβάνουν την υποστήριξη της συνεργασίας, τη διατήρηση ψηφιακών εγγράφων, την καταναμημένη διαχείριση βάσεων δεδομένων, το υπερκείμενο, την ανάκτηση πληροφοριών και υπηρεσίες όπως η ανακάλυψη πόρων και η επιλεκτική διάδοση πληροφοριών. Επιτρέπουν την παγκόσμια πρόσβαση στις πληροφορίες, την αντιγραφή χωρίς σφάλματα, τη συμπαγή αποθήκευση και την ταχεία αναζήτηση. Μια πραγματική ψηφιακή βιβλιοθήκη τηρεί επίσης αρχές που διέπουν την ένταξη και την οργάνωση του περιεχομένου. Οι εξελίξεις στην τεχνολογία των πληροφοριών έχουν θολώσει τους παραδοσιακούς ρόλους, επιτρέποντας σε οποιονδήποτε διαθέτει τα κατάλληλα εργαλεία να γίνει τόσο πάροχος όσο και καταναλωτής πληροφοριών.

Η δημιουργία και η συντήρηση μιας ψηφιακής βιβλιοθήκης περιλαμβάνει συνήθως τέσσερις φάσεις: δημιουργία ψηφιακού περιεχομένου, ευρετηρίαση και φιλτράρισμα πληροφοριών, υποστήριξη της καθολικής πρόσβασης και διασφάλιση της διατήρησης.

1.2 Ιστορική ανασκόπηση

Η πρώιμη ιστορία των ψηφιακών βιβλιοθηκών δεν είναι καλά τεκμηριωμένη, αλλά σημαντικές προσωπικότητες συνέβαλαν σημαντικά στην ανάπτυξή της. Στους πρώτους προκατόχους περιλαμβάνονται ο Paul Otlet και ο Henri La Fontaine, οι οποίοι ξεκίνησαν το Mundaneum το 1895 για να καταγράψουν τις γνώσεις του κόσμου με την ελπίδα να προωθήσουν την παγκόσμια ειρήνη. Η έννοια των ψηφιακών βιβλιοθηκών αναπτύχθηκε με την έλευση του Διαδικτύου.

Ο Vannevar Bush και ο J.C.R. Licklider συνέβαλαν καθοριστικά στην προώθηση της ιδέας με τη χρήση της σύγχρονης τεχνολογίας. Ο Bush, εμπνευσμένος από την καταστροφή της Χιροσίμα, οραματίστηκε το "Memex", μια συσκευή με οθόνες, διακόπτες και πληκτρολόγιο που θα επέτρεπε την ταχεία πρόσβαση σε αποθηκευμένα βιβλία και αρχεία. Ο Licklider, με την υποστήριξη του Ιδρύματος Ford, διερεύνησε πώς η τεχνολογία θα μπορούσε να βελτιώσει τις βιβλιοθήκες. Στο βιβλίο του "Libraries of the Future" (Βιβλιοθήκες του μέλλοντος) του 1965 περιέγραψε ένα σύστημα που χρησιμοποιούσε υπολογιστές και δίκτυα για να καταστήσει την ανθρώπινη γνώση προσβάσιμη, διαθέτοντας ένα "προγνωστικό σύστημα" με στοιχεία για τη γνώση, τις ερωτήσεις και τις απαντήσεις.

Στη δεκαετία του 1980, ο ρόλος των βιβλιοθηκών σε μια ηλεκτρονική κοινωνία έγινε κεντρικό θέμα, οδηγώντας στην ανάπτυξη του Ηλεκτρονικού Καταλόγου Δημόσιας Πρόσβασης (OPAC), ο οποίος αντικατέστησε τους

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

παραδοσιακούς καταλόγους καρτών και επέκτεινε την ανταλλαγή πόρων μεταξύ των βιβλιοθηκών. Ένα πρώιμο παράδειγμα ψηφιακής βιβλιοθήκης είναι το Κέντρο Πληροφόρησης Εκπαιδευτικών Πόρων (ERIC), το οποίο ιδρύθηκε το 1964 και ήταν διαθέσιμο στο διαδίκτυο από το 1969.

Οι ψηφιακές βιβλιοθήκες απέκτησαν ευρύτερη αναγνώριση το 1994 μέσω ενός προγράμματος ύψους 24,4 εκατομμυρίων δολαρίων υπό τη διαχείριση του NSF, στο οποίο συμμετείχαν έξι πανεπιστήμια των ΗΠΑ. Αξίζει να σημειωθεί ότι η έρευνα των Sergey Brin και Larry Page στο Πανεπιστήμιο Stanford συνέβαλε στην ίδρυση της Google. Τα πρώτα μοντέλα για τις ψηφιακές βιβλιοθήκες περιλάμβαναν το μοντέλο αναφοράς της ψηφιακής βιβλιοθήκης DELOS και το πλαίσιο 5S.

1.3 Πέντε νόμοι της βιβλιοθηκονομίας με ψηφιακή βιβλιοθήκη

Οι ψηφιακές βιβλιοθήκες, με απλά λόγια, αναφέρονται στην πλήρη υποδομή, τις πολιτικές, τις διαδικασίες και τους οργανωτικούς μηχανισμούς που επιτρέπουν την πρόσβαση σε ψηφιακό περιεχόμενο και τη διατήρησή του. Τις περισσότερες φορές, πρόκειται για επεκτάσεις των παραδοσιακών βιβλιοθηκών στον ψηφιακό χώρο. Οι κλασικοί πέντε νόμοι της βιβλιοθηκονομίας του Ranganathan έχουν επαναδιατυπωθεί για τις ψηφιακές βιβλιοθήκες ως εξής:

1. Οι ψηφιακοί πόροι είναι για χρήση
2. Κάθε χρήστης αναζητά ψηφιακούς πόρους
3. Κάθε ψηφιακός πόρος χρειάζεται τον χρήστη του
4. Εξοικονομήστε τον χρόνο του χρήστη.
5. Οι ψηφιακές βιβλιοθήκες είναι ζωντανοί αναπτυσσόμενοι οργανισμοί σε όλο τον κόσμο.

Οι αρχές αυτές καθοδηγούν την ανάπτυξη και τη διαχείριση των σύγχρονων ψηφιακών συστημάτων πληροφόρησης.

1.4 Αρχές της ψηφιακής βιβλιοθήκης

Μια ψηφιακή βιβλιοθήκη είναι μια πολύπλοκη και απαιτητική σε πόρους δραστηριότητα που απαιτεί την τήρηση ορισμένων βασικών αρχών για την επιτυχία της. Οι αρχές αυτές ισχύουν για βιβλιοθήκες που ασχολούνται με τη μετατροπή αναλογικών αντικειμένων σε ψηφιακή μορφή, τη διαχείριση ψηφιακού περιεχομένου ή τη διαχείριση ενός συνδυασμού και των δύο. Ο κύριος στόχος μιας ψηφιακής βιβλιοθήκης είναι η οργάνωση μεγάλου όγκου ψηφιακών πληροφοριών με τέτοιο τρόπο ώστε να παρέχει εύκολη πρόσβαση. Υπάρχουν δέκα αρχές που διέπουν το σχεδιασμό, την υλοποίηση και τη συντήρηση μιας ψηφιακής βιβλιοθήκης:

1. Αναμονή της αλλαγής - προνοητική στάση απέναντι στις συνεχείς τεχνολογικές και προσανατολισμένες στο χρήστη αλλαγές.
2. Γνωρίστε το περιεχόμενό σας - κατανόηση της φύσης και των απαιτήσεων του περιεχομένου που διαχειρίζεται.
3. Εμπλέξτε τους σωστούς ανθρώπους - εμπλέξτε ειδικούς και άλλους ενδιαφερόμενους στη διαδικασία.
4. Σχεδιάστε για ευχρηστία - ευκολία χρήσης της βιβλιοθήκης από τους χρήστες.
5. Κάντε την ανοικτή - μοιραστείτε την με πολλούς.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

6. Γνωρίστε τα δικαιώματα - Λάβετε υπόψη τα πνευματικά δικαιώματα και τα δικαιώματα κατά τη χρήση των δεδομένων.
7. Αυτοματοποιήστε όπου μπορείτε - Εξοικονόμηση αποδοτικότητας και μείωση της χειρωνακτικής προσπάθειας.
8. Χρησιμοποιήστε πρότυπα και στη συνέχεια συμμορφωθείτε - Υπάρχουν πρότυπα για το ψηφιακό περιεχόμενο και τα συστήματα.
9. Ανησυχήστε για την επιμονή - Εξετάστε τη μακροπρόθεσμη διατήρηση και τη συνεχή προσβασιμότητα του ψηφιακού περιεχομένου.
10. Αυτές είναι δεδομένες, αλλά σημαντικές βασικές αρχές για την καλύτερη δυνατή διασφάλιση της δημιουργίας μιας επιτυχημένης και βιώσιμης ψηφιακής βιβλιοθήκης όταν ο χρόνος και οι πόροι είναι περιορισμένοι.

1.5 Επιπτώσεις της πανδημίας Covid-19 και αλλαγές στον χώρο των ψηφιακών βιβλιοθηκών

Η πανδημία COVID-19 ξεκίνησε από τη Γουχάν της Κίνας τον Δεκέμβριο του 2019 και εξελίχθηκε στην πιο ανεπανάληπτη κρίση παγκοσμίως, επηρεάζοντας σχεδόν όλες τις διαστάσεις της ζωής και τον τρόπο εργασίας των βιβλιοθηκών. Η ασθένεια αυτή προκαλείται από τον ιό SARS-CoV-2 και εξαπλώνεται μέσω πολλαπλών οδών μετάδοσης: επαφή, σταγονίδια, αερομεταφερόμενα σωματίδια κ.λπ. Η πανδημία είχε ήδη εξαπλωθεί στις αρχές του 2020 και επέφερε αλλαγές στην παγκόσμια κοινωνική δομή, αναγκάζοντας πολλούς κλάδους και οργανισμούς, συμπεριλαμβανομένων των βιβλιοθηκών, να προσαρμοστούν με εξαιρετικούς ρυθμούς στη νέα κατάσταση.

Η πανδημία προκάλεσε καταλυτικά την υιοθέτηση και τον μετασχηματισμό της τεχνολογίας για τις βιβλιοθήκες, ιδίως τις ακαδημαϊκές. Λόγω της πανδημίας, οι φυσικές βιβλιοθήκες χρειάστηκε να κλείσουν και έτσι, τα περισσότερα από αυτά τα ιδρύματα έσπευσαν να κάνουν μεταβάσεις σε διαδικτυακά περιβάλλοντα για να συνεχίσουν να εξυπηρετούν τους χρήστες τους. Αυτό παρουσίασε δυνατότητες και προκλήσεις όσον αφορά την τεχνολογία στις υπηρεσίες των βιβλιοθηκών. Στα περισσότερα μέρη του κόσμου, περισσότερο στις αναπτυσσόμενες χώρες, οι προκλήσεις ήταν ιδιαίτερα/επιπρόσθετες λόγω της χαμηλής τεχνολογικής υποδομής, της φτωχής πρόσβασης σε ψηφιακούς πόρους εκτός πανεπιστημιούπολης και της έλλειψης ιδρυματικών αποθετηρίων. Όλα αυτά επικαλύπτονταν περαιτέρω από τις μεταβαλλόμενες συμπεριφορές αναζήτησης πληροφοριών από τους χρήστες και τα διάφορα επίπεδα ψηφιακού αλφαριθμητισμού τους.

Η πανδημία έπληξε επίσης σκληρά την ηγεσία των βιβλιοθηκών, η οποία έπρεπε να ζυγίσει τη διακοπή των παραδοσιακών υπηρεσιών με τις ανησυχίες για την υγεία και την ασφάλεια του προσωπικού και των χρηστών. Αυτό περιελάμβανε την εφαρμογή της κοινωνικής αποστασιοποίησης, την ανάπτυξη νέων πρωτοκόλλων για την απολύμανση του υλικού και την αύξηση της διαδικτυακής πρόσβασης στις πηγές. Η αβεβαιότητα της κατάστασης έφερε προβλήματα στη λήψη αποφάσεων σε αυτούς τους πολύ αβέβαιους καιρούς, ιδίως χωρίς σχέδια ετοιμότητας για πανδημία και με την ξαφνική στροφή σε απομακρυσμένα περιβάλλοντα εργασίας.

Όπως επεσήμανε ο Tammaro (2020), σε μια χώρα όπως η Ιταλία, άλλα ζητήματα που απασχολούσαν τους επαγγελματίες της βιβλιοθήκης ήταν εκείνα που αφορούσαν την ασφάλεια των θέσεων εργασίας, την εξωτερική ανάθεση και τις χρόνιες ελλείψεις χρηματοδότησης. Αυτά επιδεινώνονταν ακόμη από το ψηφιακό χάσμα, τις νομικές προκλήσεις για την ψηφιοποίηση, ιδίως σε θέματα που αφορούσαν τα πνευματικά δικαιώματα, και την έλλειψη εξειδικευμένου προσωπικού. Αν και η πανδημία δοκίμασε πολλές βιβλιοθήκες, επιτάχυνε επίσης την εξέλιξή τους προς μια "έξυπνη βιβλιοθήκη" με την αύξηση των ηλεκτρονικών πηγών και των ψηφιακών υπηρεσιών με καινοτόμες τεχνολογίες για να βοηθήσουν τους σύγχρονους χρήστες των βιβλιοθηκών.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

Εξετάζοντας όλες αυτές τις προκλήσεις, οι επαγγελματίες της βιβλιοθήκης και της επιστήμης της πληροφόρησης εργάστηκαν σκληρά για να προσεγγίσουν τους χρήστες τους μέσω διαδικτυακών και απομακρυσμένων υπηρεσιών. Αυτές οι μεταρρυθμίσεις έκτακτης ανάγκης, που ξεκίνησαν αρχικά από ανάγκη, προσέφεραν πραγματικά μια απροσδόκητη ευκαιρία στις βιβλιοθήκες να επανεξετάσουν και να επαναπροσδιορίσουν τα μοντέλα υπηρεσιών. Αυτός ο επαναπροσδιορισμός κατέχει το κλειδί για τη βελτίωση της εμπειρίας των χρηστών και συνεχίζει να αναπτύσσεται σε έναν κόσμο μετά την πανδημία.

Το σημείο που αναδεικνύεται από τη μελέτη είναι ο κρίσιμος ρόλος που διαδραμάτισαν οι βιβλιοθήκες κατά τη διάρκεια της πανδημίας COVID-19 και οι νέοι ρόλοι που πρέπει να αναλάβουν για να βοηθήσουν τους χρήστες καθώς οι ανάγκες τους συνεχίζουν να αλλάζουν. Συζητά περαιτέρω τις προκλήσεις που αντιμετωπίζουν κατά τη διάρκεια αυτής της περιόδου, όπως η έλλειψη καλύτερων εργαλείων επικοινωνίας, η επιθυμία για καλύτερη ψηφιακή παιδεία των χρηστών και η ανάπτυξη ισχυρών πολιτικών και στρατηγικών. Από αυτή την άποψη, όταν γίνει αυτό, οι βιβλιοθήκες θα μπορέσουν να κλιμακώσουν τις προκλήσεις που έφερε η πανδημία και να βγουν πιο δυνατές και ικανές να εξυπηρετήσουν τις κοινότητές τους στο μέλλον.

1.6 Η εξέλιξη και το μέλλον των ψηφιακών βιβλιοθηκών

Οι ψηφιακές βιβλιοθήκες έχουν θεωρηθεί ως μια χαλαρά καθορισμένη, διεπιστημονική περιοχή- έχει υποστεί σημαντική εξέλιξη τις τελευταίες δεκαετίες. Οι ψηφιακές βιβλιοθήκες προσέλκυσαν αρχικά το ενδιαφέρον ενός μάλλον εκλεκτικού μείγματος οραματιστών, μηχανικών και βιβλιοθηκονόμων. Πράγματι, σε διάφορες περιπτώσεις, έχει προκύψει το ερώτημα σχετικά με το αν οι ψηφιακές βιβλιοθήκες εμπίπτουν στον πραγματικό ορισμό των βιβλιοθηκών και της βιβλιοθηκονομίας. Τα βασικά ζητήματα σε αυτόν τον τομέα περιστρέφονται έτσι γύρω από το ρόλο των βιβλιοθηκών στο μέλλον ως κοινωνικών και πολιτιστικών θεσμών και γύρω από ζητήματα πνευματικής ιδιοκτησίας - και τα δύο αναμφισβήτητα πολύ σημαντικά για να αφηθούν μόνο στους βιβλιοθηκονόμους.

Το όραμα της ψηφιακής βιβλιοθήκης μπορεί να αναχθεί στις αρχές του 20ού αιώνα, όταν οραματιστές όπως ο H.G. Wells και ο Vannevar Bush ήταν από τους πρώτους στοχαστές που πρότειναν νέους τρόπους με τους οποίους η τεχνολογία θα μπορούσε να αξιοποιηθεί στο έργο της οργάνωσης και της παροχής πρόσβασης στην ανθρώπινη γνώση. Στα μέσα της δεκαετίας του 1990, όταν το Διαδίκτυο έφτασε σε πλήρη άνθηση, το ουτοπικό όραμα της παγκόσμιας πρόσβασης στη γνώση φαινόταν σχεδόν εφικτό. Αλλά αυτά τα οράματα μετριάστηκαν γρήγορα από εμπορικά και κυβερνητικά συμφέροντα.

Η τεχνική βάση των σημερινών ψηφιακών βιβλιοθηκών τέθηκε τη δεκαετία του 1960 από τις πρώτες ερευνητικές και εμπορικές υπηρεσίες πληροφόρησης, την αυτοματοποίηση βιβλιοθηκών και τα συστήματα διαχείρισης εγγράφων. Κατά τη διάρκεια πέντε ετών, από το 1994 έως το 2004, αυτή η προγραμματική χρηματοδότηση από οργανισμούς όπως το Εθνικό Ίδρυμα Επιστημών νομιμοποίησε τις ψηφιακές βιβλιοθήκες ως ερευνητικό πεδίο στις Ηνωμένες Πολιτείες. Οι δραστηριότητες αυτές παρείχαν ένα ενεργό περιβάλλον για μια διεπιστημονική κοινότητα: επιστήμονες πληροφορικής, βιβλιοθηκονόμοι και κοινωνικοί επιστήμονες συνεργάστηκαν.

Μέχρι το 2005, η μεγάλη κυβερνητική χρηματοδότηση του πρωτοτύπου της ψηφιακής βιβλιοθήκης είχε μειωθεί και η εστίαση στράφηκε στην παραγωγική ανάπτυξη των τεχνολογιών ψηφιακών βιβλιοθηκών σε τομείς όπως η ηλεκτρονική έρευνα και οι κυβερνητικές υποδομές. Η "επιμέλεια δεδομένων" αναδείχθηκε ως ένας τομέας στον οποίο οι ψηφιακές βιβλιοθήκες θα είχαν σημαντική συμβολή.

Η ψηφιακή διατήρηση των πληροφοριών έχει καταστεί ένα κρίσιμο ζήτημα που αφορά τη μακροπρόθεσμη επιβίωσή τους. Η έρευνα στον τομέα αυτό παρουσιάζει πολλές προκλήσεις, ιδίως όσον αφορά την απόδειξη της αποτελεσματικότητας των μεθόδων διατήρησης για εξαιρετικά μεγάλο χρονικό διάστημα. Τα σχετικά ζητήματα της διαχείρισης, της μνήμης και της ηθικής της ψηφιακής διατήρησης εκτείνονται πέρα από τους παραδοσιακούς επιστημονικούς και μηχανολογικούς τομείς.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

Κοιτάζοντας μπροστά, ορισμένοι ερευνητικοί τομείς σε αυτόν τον τομέα περιλαμβάνουν τη διαχείριση προσωπικών πληροφοριών, τη μακροπρόθεσμη αλληλεπίδραση του ανθρώπου με τα συστήματα πληροφοριών, το ρόλο των ψηφιακών βιβλιοθηκών στην εκπαίδευση και τη δια βίου μάθηση και την ανάπτυξη ενεργών περιβαλλόντων για συνεργατική εργασία. Το ενοποιητικό θέμα για το μέλλον πρέπει να είναι: Πώς οι ψηφιακές βιβλιοθήκες θα ενσωματωθούν σε ευρύτερες κοινωνικές δραστηριότητες. Μέσα στα επόμενα δέκα χρόνια, οι ψηφιακές βιβλιοθήκες θα μπορούσαν να κάνουν τη μετάβαση από μια πειραματική τεχνολογία σε ένα σχεδόν καθολικά διαθέσιμο και ουσιαστικό μέρος του περιβάλλοντος πληροφόρησης.

1.7 Επίδραση των ψηφιακών πηγών της βιβλιοθήκης στην ανάπτυξη των φοιτητών της τριτοβάθμιας εκπαίδευσης

Οι ψηφιακές βιβλιοθήκες έχουν καταστεί αρκετά επιτακτικές στα πανεπιστήμια λόγω των βελτιώσεων που επιτεύχθηκαν στην τεχνολογία της πληροφορίας. Οι βιβλιοθήκες αυτές προσφέρουν κυρίως μεγάλο αριθμό ηλεκτρονικών πηγών, όπως κείμενο, εικόνες, ήχο και κινούμενα σχέδια, οι οποίες οργανώνονται από έναν υπολογιστή και η πρόσβαση σε αυτές γίνεται μέσω υπολογιστή. Ως αποτέλεσμα, πολλά πανεπιστήμια έχουν αρχίσει να διαθέτουν όλο και μεγαλύτερο ποσοστό του προϋπολογισμού τους για την ανάπτυξη και τη βελτίωση των ψηφιακών βιβλιοθηκών, θεωρώντας ότι επιτρέπουν τεράστια πρόσβαση σε εκπαιδευτικό υλικό. Η επένδυση αυτή βασίζεται στη συνειδητοποίηση ότι οι ψηφιακές βιβλιοθήκες μπορούν να βελτιώσουν δραματικά την αποτελεσματικότητα του τρόπου με τον οποίο οι φοιτητές αποκτούν πρόσβαση σε πηγές πληροφοριών και του τρόπου με τον οποίο χρησιμοποιούν τις πληροφορίες αυτές.

Η εξέλιξη της ψηφιακής βιβλιοθήκης έχει επηρεαστεί από τη ραγδαία ανάπτυξη του Διαδικτύου και τους μεταβαλλόμενους τρόπους με τους οποίους οι χρήστες αναζητούν πλέον πληροφορίες. Η ψηφιακή βιβλιοθήκη δεν είναι πλέον στατικά αποθετήρια, αλλά έχει μετατραπεί σε ένα δυναμικό μέσο που προσφέρει στους μαθητές νέες ευκαιρίες μάθησης. Με πολύ λίγα κλικ, οι μαθητές μπορούν να έχουν πρόσβαση σε πολλαπλές σχετικές πηγές πληροφοριών ψηφιακά μέσω των βιβλιοθηκών, με την προϋπόθεση της διαθεσιμότητας βασικών συνδέσεων στο διαδίκτυο και στην ηλεκτρική ενέργεια. Η εύκολη πρόσβαση επιτρέπει στους φοιτητές να έχουν πρόσβαση σε ορισμένες πολύ σημαντικές ακαδημαϊκές πηγές σχεδόν σε οποιοδήποτε σημείο της γης, καθιστώντας τη μάθηση πιο ευέλικτη και ευρείας εμβέλειας.

Μια μελέτη που διεξήχθη στο Μπανγκαλόρ στο JAIN (θεωρούμενο ως Πανεπιστήμιο) επικεντρώθηκε ιδιαίτερα στους πόρους της ψηφιακής βιβλιοθήκης που είναι διαθέσιμοι στο πανεπιστήμιο και στον αντίκτυπό τους στην ανάπτυξη των φοιτητών. Η έρευνα σχεδιάστηκε για να διερευνήσει τον τρόπο με τον οποίο οι εν λόγω πόροι της ψηφιακής βιβλιοθήκης συνέβαλαν στη διαδικασία απόκτησης γνώσεων από τους φοιτητές. Η παρούσα εργασία, ως εκ τούτου, επισφραγίζει ότι η διαθεσιμότητα και η χρήση των πόρων της ψηφιακής βιβλιοθήκης υποστηρίζουν και συμπληρώνουν τις μαθησιακές εμπειρίες των μαθητών προς την κατεύθυνση καλύτερων ακαδημαϊκών αποτελεσμάτων.

Η μελέτη παραπέμπει σε προηγούμενη μελέτη του Ekere, 2016, στην οποία διαπιστώθηκε ότι οι χρήστες είχαν γενικά θετική άποψη για τους πόρους, τις υπηρεσίες και τις διευκολύνσεις της ψηφιακής βιβλιοθήκης της MTN. Η εν λόγω έρευνα είχε διαπιστώσει ότι η βιβλιοθήκη της MTN προσέφερε πιο εξέχουσες υπηρεσίες, όπως ηλεκτρονικό ταχυδρομείο, αναζήτηση στο διαδίκτυο και διαδικτυακά εργαλεία αναφοράς, σε σχέση με τις χαμηλότερες υπηρεσίες που συνηθίζονται σε άλλες βιβλιοθήκες. Έχοντας αυτά υπόψη, οι βιβλιοθήκες συνιστάται να συνεχίσουν να διευρύνουν την πρόσβαση και την παροχή πληροφοριών. Με αυτόν τον τρόπο, οι βιβλιοθήκες μπορούν να ενισχύσουν τη χρησιμότητα και τον αντίκτυπο των ηλεκτρονικών πόρων με τρόπο που να ανταποκρίνεται πολύ καλύτερα στις διάφορες ανάγκες και προσδοκίες των χρηστών και που να εδραιώνει τη βιβλιοθήκη ως αναπόσπαστο στοιχείο του ακαδημαϊκού περιβάλλοντος.

ΚΕΦΑΛΑΙΟ 2

Εισαγωγή

Σε αυτό το κεφάλαιο, θα εξετάσουμε τις βασικές τεχνολογίες που έχω χρησιμοποιήσει στην διπλωματική εργασία μου και τον ρόλο τους στην ανάπτυξη λογισμικού. Οι τεχνολογίες αυτές περιλαμβάνουν το Laravel, το Filament, το MySQL, το Vue.js και το RestAPI, Bootstrap καθώς και άλλες σημαντικές τεχνολογικές έννοιες. Κάθε μία από αυτές τις τεχνολογίες συμβάλλει στη δημιουργία μιας ισχυρής, αποδοτικής και εύχρηστης εφαρμογής.

2.1 Vue.js

2.1.1 Ιστορική αναδρομή

Vue.js Το Vue.js είναι ένα προοδευτικό JavaScript framework που δημιουργήθηκε από τον Evan You το 2014. Ο You είχε εργαστεί προηγουμένως στην Google, και ανέπτυξε το Vue.js ως μια απλούστερη εναλλακτική στις άλλες δημοφιλείς βιβλιοθήκες και frameworks της εποχής, όπως το Angular και το React. Από την κυκλοφορία του, το Vue.js έχει γίνει εξαιρετικά δημοφιλές, χάρη στην ευκολία εκμάθησης και χρήσης του.

2.1.2 Πεδίο εφαρμογής

Vue.js μπορεί να χρησιμοποιηθεί για κατασκευή web-εφαρμογών καθώς και απλές σελίδες, Μπορεί να εφαρμόζεται σταδιακά από μικρά κομμάτια που ενσωματώνονται εύκολα στην εφαρμογή καθώς μπορεί να επεκταθεί σε πιο σύνθετες εφαρμογές.

Η Vue.js χρησιμοποιείται για την ανάπτυξη frontend εφαρμογές, ο πιο δημοφιλής τρόπος είναι να δημιουργείτε SPA που σημαίνει ότι οι περισσότερες αλληλεπιδράσεις του χρήστη με την εφαρμογή γίνονται σε μία μόνο σελίδα, χωρίς την ανάγκη για επαναφόρτωση της ιστοσελίδας για κάθε δράση, πράγμα που βελτιώνει την εμπειρία του χρήστη και μειώνει τους χρόνους φόρτωσης της σελίδας.

2.1.3 Τα Βασικά πλεονεκτήματα του Vue.js

- **Απλότητα και Ευελιξία:** Εύκολο στην εκμάθηση και χρήση, με δυνατότητα προσαρμογής στις ανάγκες εκάστοτε έργου.
- **Επαναχρησιμοποίηση Κώδικα:** Διευκολύνει τη δημιουργία επαναχρησιμοποιήσιμων στοιχείων(components).
- **Δυναμικότητα:** Προσφέρει εξαιρετική απόδοση και δυναμικότητα καθώς χρησιμοποιώντας Vue.js ελάχιστες φορές θα χρειαστεί να επαναφορτώσεις την σελίδα. Χρησιμοποιεί και αλλάζει τα δεδομένα δυναμικά.

2.2.4 Τα μειονεκτήματα του Vue.js

Υπάρχουν μόνο μερικά μειονεκτήματα της Vue.js, όπως η περιορισμένη κοινότητα και το περιορισμένο οικοσύστημα σε σύγκριση με το React και το Angular.

Αν και το Vue.js κερδίζει καθημερινά όλο και περισσότερη δημοτικότητα, η κοινότητά του εξακολουθεί να είναι πολύ μικρότερη από εκείνη άλλων δημοφιλών πλαισίων όπως το React ή το Angular. Εξαιτίας αυτού του γεγονότος, μπορεί να υπάρχουν λιγότερες βιβλιοθήκες, plugins ή εργαλεία τρίτων κατασκευασμένων ακριβώς για το Vue.js, άρα λίγο περισσότερος πόνος για την επίλυση κάποιων συγκεκριμένων προβλημάτων ή ενσωματώσεων.

Γλωσσικό εμπόδιο: Αναπτύχθηκε για πρώτη φορά από έναν προγραμματιστή από την Κίνα, και ένα μεγάλο μέρος της κοινότητας είναι οι Κινέζοι. Εξαιτίας αυτού, ενδέχεται να συναντήσετε κάποια γλωσσικά εμπόδια στις συζητήσεις της κοινότητας, στην τεκμηρίωση και στους πόρους τρίτων. Ενώ η βασική τεκμηρίωση συχνά μεταφράζεται στα αγγλικά, το περιεχόμενο τρίτων δεν μεταφράζεται τόσο πολύ.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

Η ευελιξία οδηγεί σε υπερβάσεις: Αυτή η ευελιξία της Vue.js μπορεί επίσης να είναι ένα δίκοπο μαχαίρι. Είναι πιθανό να φέρει ασυνεπείς πρακτικές κώδικα σε διαφορετικές ομάδες ή έργα, καθιστώντας έτσι δύσκολη τη συντήρηση και την κλιμάκωση μεγάλων εφαρμογών χωρίς αυστηρές κατευθυντήριες γραμμές για την κωδικοποίηση.

Έλλειψη υιοθέτησης σε μεγάλη κλίμακα: Αυτό σημαίνει ότι η Vue.js είναι πιο διαδεδομένη σε νεοσύστατες επιχειρήσεις και μικρομεσαίες επιχειρήσεις. Οι μεγαλύτερες εταιρείες, ωστόσο, προτιμούν frameworks όπως το React ή το Angular επειδή έχουν αντέξει στη μάχη σε πιο σύνθετες και μεγαλύτερες εφαρμογές. Αυτό από μόνο του μπορεί να δώσει λιγότερα παραδείγματα και μελέτες περιπτώσεων για μεγάλες χρήσεις του Vue.js. Ως εκ τούτου, οι νέοι χρήστες μπορεί να δυσκολευτούν ακόμη περισσότερο να δικαιολογήσουν τη χρήση του σε επιχειρηματικά περιβάλλοντα.

Περιορισμένη υποστήριξη για παλαιότερα προγράμματα περιήγησης: Ένα από τα μειονεκτήματα του Vue.js είναι ότι έχει πολύ μικρή υποστήριξη για παλαιότερα προγράμματα περιήγησης σε σύγκριση με άλλα, όπως το Angular, το οποίο πραγματικά λάμπει από αυτή την άποψη. Αυτό θα μπορούσε να αποτελέσει μειονέκτημα για έργα που στοχεύουν σε παλαιότερες εκδόσεις browsers, ιδίως σε επιχειρηματικά πλαίσια όπου μπορεί να χρησιμοποιούνται ακόμη παλαιά συστήματα.

Η αμφίδρομη δέσμευση δεδομένων μπορεί να είναι δίκοπο μαχαίρι: Ενώ ένα από τα πιο ισχυρά χαρακτηριστικά του Vue.js είναι η αμφίδρομη δέσμευση δεδομένων, μπορεί να αντιμετωπίσει προβλήματα απόδοσης σε μεγάλες εφαρμογές λόγω του μεγάλου φόρτου παρακολούθησης εξαρτήσεων και αναβαθμίσεων απόδοσης. Ο προγραμματιστής θα πρέπει, επομένως, να είναι ευαίσθητος και να διαχειρίζεται σωστά αυτό το χαρακτηριστικό, προκειμένου να αποφύγει τυχόν συμφόρηση στην απόδοση.

Περιορισμένο υλικό για εκκμάθηση: παρά τη μεγάλη ανάπτυξη που γνώρισε το πλαίσιο τα τελευταία χρόνια. Σε σύγκριση με άλλα frameworks όσον αφορά τον αριθμό των plugins, το Vue.js εξακολουθεί να υστερεί, σε ορισμένες περιπτώσεις κατά εκατοντάδες.

Ένα άλλο μειονέκτημα των plugins που υποστηρίζονται από τα περισσότερα frameworks, συμπεριλαμβανομένου του Vue.js, είναι ότι πολλά από αυτά δεν παρέχουν καλή υποστήριξη. Το Vue.js, ωστόσο, ωριμάζει και υπάρχει λόγος να ανακαλύψουμε τις νέες εξελίξεις που λαμβάνουν χώρα σε αυτό το πλαίσιο.

Bootstrap

2.2 Ιστορική Αναδρομή του Bootstrap

Το Bootstrap είναι ένα frontend framework για την ανάπτυξη responsive και mobile-first ιστοσελίδων και εφαρμογών. Αναπτύχθηκε αρχικά από τους Mark Otto και Jacob Thornton στην Twitter και κυκλοφόρησε ως έργο ανοιχτού κώδικα το 2011. Το Bootstrap έγινε γρήγορα δημοφιλές λόγω της ευκολίας χρήσης του και της δυνατότητας να δημιουργεί επαγγελματικά σχέδια με ελάχιστη προσπάθεια.

2.2.1 Εφαρμογή

Το Bootstrap βοηθάει στην κατασκευή των ιστοσελίδων καθώς περιέχει πολλά έτοιμα components και κλάσεις που εφαρμόζονται εύκολα στον κώδικα.

Components που περιέχει είναι: Μπάρες πλοήγησης(navigation bars), κουμπιά, φόρμες, πλέγματα(grid) και πολλά άλλα έτοιμα components καθώς και ολόκληρα έτοιμα themes τα οποία μπορεί εύκολα ένας προγραμματιστής να τα εφαρμόσει στην δική του εφαρμογή.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

Τα προσχεδιασμένα πρότυπα του Bootstrap και το ισχυρό σύστημα πλέγματος(grid) επιτρέπουν στους προγραμματιστές να δημιουργούν γρήγορα συνεπή και οπτικά ελκυστικά σχέδια, εξασφαλίζοντας συμβατότητα σε διαφορετικές συσκευές και μεγέθη οθόνης.

2.2.2 Τα πλεονεκτήματα

- **Responsive Design:** Περιέχει έτοιμες classes και elements που διευκολύνουν τη δημιουργία responsive σχεδιασμών.
- **Ευκολία στη Χρήση:** Περιλαμβάνει έτοιμα πρότυπα και στοιχεία που μπορούν να προσαρμοστούν γρήγορα.
- **Συμβατότητα με Όλα τα Προγράμματα Περιήγησης:** Εξασφαλίζει ότι οι εφαρμογές λειτουργούν σωστά σε όλα τα μοντέρνα προγράμματα περιήγησης.

MVC

2.3 Περιγραφή

Το Model-View-Controller (MVC) είναι ένα ευρέως χρησιμοποιούμενο πρότυπο σχεδίασης λογισμικού για την ανάπτυξη διεπαφών χρήστη, το οποίο σπάει τη λογική του προγράμματος σε τρία αλληλοσυνδεόμενα στοιχεία: το **Model**, την **View** και **Controller**.

Αρχικά εφαρμόστηκε σε γραφικές διεπαφές χρήστη (GUI), το MVC έχει αποκτήσει σημαντική δημοτικότητα στην ανάπτυξη εφαρμογών ιστού και χρησιμοποιείται πάρα πολύ μέχρι και σήμερα. Διάφορες γλώσσες προγραμματισμού προσφέρουν MVC frameworks και το Laravel είναι ένα από αυτά.

Αλλά frameworks που χρησιμοποιούν το πρότυπο σχεδίασης λογισμικού MVC:

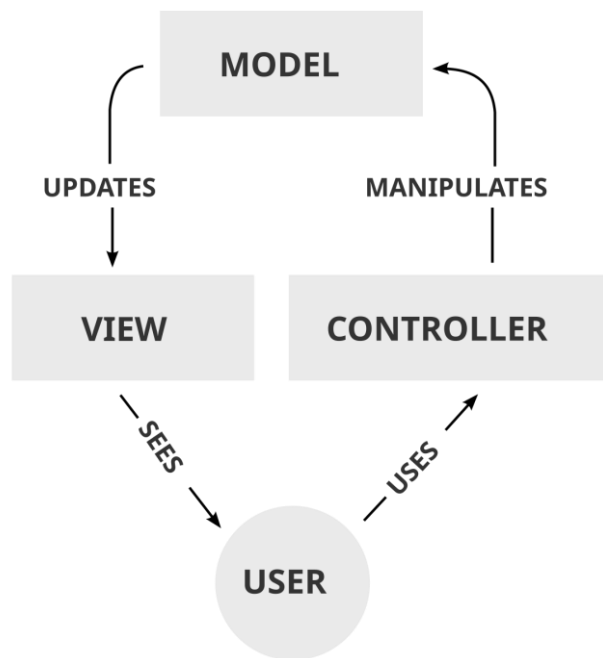
- Ruby on Rails
- Django
- Spring
- Express.js

Σίγουρα υπάρχουν και άλλα frameworks που χρησιμοποιούν το ίδιο πρότυπο.

2.3.1 Τα τρία συστατικά του MVC

- Το **Model** διαχειρίζεται την εσωτερική αναπαράσταση των πληροφοριών, διαχειρίζεται αλληλεπίδραση με την βάση δεδομένων, επίσης χειρίζεται την λογική της επιχειρηματικής διαδικασίας μιας εφαρμογής
- Το **View** χειρίζεται την παρουσίαση και την αλληλεπίδραση με τον χρήστη.
- Ο **Controller** συνδέει το **Model** και το **View**, με άλλα λόγια χειρίζεται την αλληλεπίδραση του χρήστη με την εφαρμογή, ενημερώνει το Model και την View (Προβολή) για τις αλλαγές που έχουν γίνει στην εφαρμογή (μέσω του Controller).

2.3.2 Απεικόνιση του πρότυπου



Εικόνα 1 MVC

Laravel

2.4 Ιστορική αναδρομή

Το Laravel είναι ένα php framework που δημιουργήθηκε από τον Taylor Otwell και παρουσιάστηκε για πρώτη φορά το 2011. Ο Otwell ανέπτυξε το Laravel ως μια λύση στις περιορισμένες δυνατότητες των υπαρχόντων frameworks, προσφέροντας ένα σύστημα που θα συνδυάζε την απλότητα και τη δύναμη. Από την κυκλοφορία του, το Laravel γνώρισε τεράστια ανάπτυξη, με πολλές αναβαθμίσεις και προσθήκες, που το καθιστούν ένα από τα πιο δημοφιλή frameworks στον κόσμο της ανάπτυξης ιστοσελίδων.

2.4.1 Περιγραφή του Laravel

Το Laravel είναι ένα open-source framework, βασίζεται στην γλώσσα προγραμματισμού php.

Το Laravel κάνει την ανάπτυξη λογισμικού πιο εύκολη, γρήγορη και είναι πολύ ευέλικτο.

Ακολουθεί το αρχιτεκτονικό πρότυπο MVC και παρέχει μια πλήρη, υψηλής ποιότητας και χωρίς προβλήματα δομή για την ανάπτυξη εφαρμογών ιστού(web) οποιασδήποτε πολυπλοκότητας, έτσι μπορείς να αναπτύξεις και μεγάλα σύνθετα έργα σε αυτό.

2.4.2 Κύρια χαρακτηριστικά

2.4.2.1. Eloquent ORM

Το Eloquent ORM (Object-Relational Mapping) απλοποιεί τις αλληλεπιδράσεις με τις βάσεις δεδομένων, επιτρέποντας στους προγραμματιστές να αποφύγουν να γράφουν SQL queries(ερωτήματα), αντί για αυτό μπορούμε να γράφουμε εντολές με εκφραστικό συντακτικό που κάνει πιο εύκολη την δουλειά.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

2.4.2.2 Artisan Console

Το Artisan είναι η διεπαφή γραμμής εντολών που παρέχεται από το Laravel. Προσφέρει πολλές χρήσιμες εντολές για εργασίες ρουτίνας, όπως migrate βάσεων δεδομένων, διαχείριση πακέτων, βελτιώνοντας τη ροή εργασιών ανάπτυξης.

2.4.2.3 Routing

Το σύστημα δρομολόγησης(routing) του Laravel είναι απλό και ευέλικτο, πράγμα το οποίο επιτρέπει στους προγραμματιστές να καθορίζουν αποτελεσματικά τις διαδρομές για τις εφαρμογές τους.

Υποστηρίζει route groups - ομαδοποίηση των routing, middleware και resource routing, καθιστώντας εύκολη τη διαχείριση και την οργάνωση των διαδρομών(routes).

2.4.2.4 Middleware

Το Middleware στο Laravel παρέχει έναν μηχανισμό για το φιλτράρισμα των αιτήσεων HTTP που κάνουν requests (μπαίνουν) στην εφαρμογή.

Μπορεί να χρησιμοποιηθεί για εργασίες όπως ο έλεγχος ταυτότητας(authentication) , logging και το CORS.

2.4.2.5 Security

Το Laravel περιλαμβάνει ενσωματωμένα(build-in) χαρακτηριστικά ασφαλείας, όπως προστασία από έγχυση SQL(SQL Injection), πλαστογράφηση αιτήσεων cross-site (CSRF) και cross-site scripting (XSS).

Παρέχει επίσης μηχανισμούς για κατακερματισμό κωδικού πρόσβασης και πιστοποίηση ταυτότητας.

2.4.2.6 Testing

Το Laravel παρέχει εργαλεία και frameworks με τα οποία εύκολα μπορείς να κάνεις testing την εφαρμογή σου, περιλαμβάνει πολλές βοηθητικούς μεθόδους και λειτουργίες προσομοίωσης για να διασφαλιστεί η αξιοπιστία της εφαρμογής.

2.4.2.7 Package Ecosystem

Το Laravel διαθέτει ένα μεγάλο οικοσύστημα πακέτων και εργαλείων που επεκτείνουν τις δυνατότητές του.

Αυτό περιλαμβάνει επίσημα πακέτα όπως το Cashier (για τιμολόγηση), το Passport (για έλεγχο ταυτότητας API), το Scout (για αναζήτηση) και το Horizon (για ουρές), καθώς και ένα ευρύ φάσμα πακέτων που συνεισφέρει η κοινότητα.

2.4.3 Σύνοψη

Το Laravel είναι ένα πολύ ισχυρό εργαλείο με πολλά build-in features το οποίο έχει πολλά άλλα features, συνεχώς αναπτύσσεται, καθώς έχει και πολύ μεγάλη κοινότητα.

2.5 Filament

2.5.1 Περιγραφή

Το Filament είναι μια βιβλιοθήκη διαχειριστικών διεπαφών για το Laravel, σχεδιασμένη για να επιταχύνει την ανάπτυξη και τη διαχείριση backend συστημάτων. Δημιουργήθηκε για να παρέχει στους προγραμματιστές ένα έτοιμο σύνολο εργαλείων και widgets που απλοποιούν τη δημιουργία διοικητικών πάνελ(Admin panel). Παρόλο που είναι σχετικά νέο, έχει ήδη καταφέρει να κερδίσει την εμπιστοσύνη πολλών προγραμματιστών χάρη στην ευκολία και την αποδοτικότητά του.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

Παρέχει ένα φιλικό προς το χρήστη περιβάλλον εργασίας και μια σειρά από εργαλεία που απλοποιούν τη δημιουργία και τη διαχείριση εργασιών, λειτουργίες CRUD, οπτικοποίηση δεδομένων και διαχείριση πόρων.

Το Filament είναι εξαιρετικά προσαρμόσιμο, υποστηρίζοντας τόσο απλές όσο και προηγμένες περιπτώσεις χρήσης, καθιστώντας το ιδανική επιλογή για προγραμματιστές που επιθυμούν να δημιουργήσουν ισχυρά συστήματα backend γρήγορα και αποτελεσματικά.

2.5.2 Τα πλεονεκτήματα του Filament

- **Ευκολία Χρήσης:** Παρέχει προρυθμισμένα στοιχεία(components) που μειώνουν τον χρόνο ανάπτυξης.
- **Ευελιξία:** Επιτρέπει την προσαρμογή των Admin Panels στις ανάγκες του εκάστοτε έργου.
- **Ασφάλεια:** Περιλαμβάνει ενσωματωμένες λειτουργίες ασφαλείας για την προστασία των δεδομένων.

MySQL

2.6.1 Ιστορική αναδρομή

Το MySQL είναι ένα από τα πιο διαδεδομένα συστήματα διαχείρισης βάσεων δεδομένων ανοιχτού κώδικα. Αναπτύχθηκε από τους Σουηδούς προγραμματιστές Michael Widenius και David Axmark το 1995. Από τότε, το MySQL εξελίχθηκε σε ένα πανίσχυρο εργαλείο που χρησιμοποιείται από εκατομμύρια εφαρμογές παγκοσμίως. Το 2008, η Sun Microsystems εξαγόρασε το MySQL, και αργότερα, το 2010, η Oracle αγόρασε τη Sun Microsystems.

2.6.2 Περιγραφή

Το MySQL χρησιμοποιεί τη Δομημένη Γλώσσα Ερωτήσεων (SQL) για τη διαχείριση και τον χειρισμό δεδομένων.

Η MySQL χρησιμοποιείται ευρέως για διαδικτυακές εφαρμογές, συστήματα διαχείρισης περιεχομένου και διάφορα λογισμικά επιχειρηματικού επιπέδου. Είναι γνωστή για την αξιοπιστία της, την ευκολία χρήσης και την ισχυρή υποστήριξη της κοινότητας.

2.6.3 Τα βασικά χαρακτηριστικά του MySQL

- **Υψηλή Απόδοση:** Ικανό να διαχειρίζεται μεγάλες ποσότητες δεδομένων με ταχύτητα και ακρίβεια.
- **Κλιμακωτότητα:** Μπορεί να προσαρμοστεί σε μικρές και μεγάλες εφαρμογές.
- **Ασφάλεια:** Παρέχει ισχυρά εργαλεία για την προστασία των δεδομένων.

2.7 Frontend Περιγραφή

Το frontend αναφέρεται στο κομμάτι της εφαρμογής που ένας χρήστης βλέπει και αλληλεπιδρά άμεσα.

Είναι ο σχεδιασμός της σελίδας, κουμπιά, το περιεχόμενο που βλέπει ο χρήστης όταν μπαίνει σε μια σελίδα.

Οι τεχνολογίες που χρησιμοποιούνται συνήθως για την ανάπτυξη οι βασικές : HTML, CSS & Javascript αλλά στις μέρες μας περισσότεροι προτιμούν να αναπτύξουν εφαρμογές σε δημοφιλείς βιβλιοθήκες ή frameworks όπως React.js, Angular και Vue.js

Το Frontend επικοινωνεί με το Backend με API : Το frontend(πελάτης) στέλνει αίτημα στο backend(server) και ο server απαντάει και γυρνάει τα δεδομένα που αιτήθηκε ο πελάτης.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

2.8 Backend Περιγραφή

Ο όρος backend αναφέρεται στο τμήμα της εφαρμογής που διαχειρίζεται την λογική της επιχειρηματικής διαδικασίας, στο backend γίνεται η αλληλεπίδραση με την βάση δεδομένων και την διαχείριση αιτήσεων με σκοπό να γυρίσει στο frontend τα δεδομένα που χρειάζεται.

Μερικές δημοφιλείς γλώσσες προγραμματισμού καθώς και frameworks που χρησιμοποιούνται για ανάπτυξη backend είναι: Php, Laravel, node.js, Java κ.α.

2.9. RestAPI

2.9.1 Ιστορική αναδρομή

Το RestAPI, ή Representational State Transfer Application Programming Interface, είναι μια αρχιτεκτονική που εισήχθη για πρώτη φορά από τον Roy Fielding το 2000. Η αρχιτεκτονική REST έγινε γρήγορα δημοφιλής λόγω της απλότητάς της και της δυνατότητάς της να υποστηρίζει την επικοινωνία μεταξύ των πελατών και των διακομιστών μέσω του πρωτοκόλλου HTTP.

2.9.2 Περιγραφή

Ένα REST API, Representational State Transfer Application Programming Interface, είναι ουσιαστικά ένα σύνολο γραμμών και συμβάσεων που χρησιμοποιούνται για τη δημιουργία, καθώς και για την αλληλεπίδραση με υπηρεσίες ιστού. Χρησιμοποιεί τις τυποποιημένες μεθόδους HTTP για τη λειτουργία πόρων που προσδιορίζονται από διευθύνσεις URL.

Τα REST APIs είναι stateless. Κάθε αίτημα(request) που στέλνει ένας πελάτης στον διακομιστή περιλαμβάνει όλες τις πληροφορίες που είναι απαραίτητες για την εξυπηρέτηση του εν λόγω αιτήματος και δεν βασίζεται σε αποθηκευμένο πλαίσιο στον διακομιστή.

Αυτό το χαρακτηριστικό τα καθιστά πολύ δημοφιλή λόγω της απλότητας, της επεκτασιμότητας και της συμβατότητας με τα πρότυπα ιστού.

2.9.3 Βασικές αρχές

- **Stateless:** Κάθε αίτηση από τον πελάτη προς τον διακομιστή πρέπει να περιέχει όλες τις πληροφορίες που απαιτούνται για την επεξεργασία της.
- **Cacheable:** Οι απαντήσεις μπορούν να αποθηκευτούν στην cache για να βελτιωθεί η απόδοση.
- **Uniform Interface:** Χρησιμοποιεί ένα σταθερό σύνολο κανόνων για την αλληλεπίδραση με τους πόρους.

2.10 i18n - μεταφράσεις

Με την βοήθεια του πακέτου “vue-i18n” έγιναν οι μεταφράσεις στο frontend.

2.10.1 Ιστορική αναδρομή

Όπως αναφέρθηκε πριν, το Vue.js κυκλοφόρησε για πρώτη φορά το 2014, όμως το πακέτο vue-i18n βγήκε μόλις το 2016 για να καλύψει τις ανάγκες των μεταφράσεων σε πολλές γλώσσες εντός μιας εφαρμογής σε vue.js. Μετά το

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

2020 όταν βγήκε η καινούργια έκδοση Vue.js 3 το πακέτο αναβαθμίστηκε για να υποστηρίζει τις νέες δυνατότητες και τη βελτιωμένη αρχιτεκτονική του Vue 3.

2.10.2 Περιγραφή

Το vue-i18n είναι ένα ολοκληρωμένο και ισχυρό εργαλείο που διευκολύνει την ανάπτυξη πολυγλωσσικών εφαρμογών Vue.js, επιτρέποντας στους προγραμματιστές να προσφέρουν μια βελτιωμένη και προσαρμοσμένη εμπειρία χρήστη σε διεθνές επίπεδο.

Το vue-i18n αποτελεί ένα από τα πιο διαδεδομένα εργαλεία διεθνοποίησης για την πλατφόρμα Vue.js. Η ανάπτυξή του αντικατοπτρίζει την αυξανόμενη ανάγκη για πολυγλωσσικές εφαρμογές στο παγκόσμιο διαδίκτυο.

Η εγκατάσταση και η χρήση του vue-i18n είναι απλή και μπορεί να προσαρμοστεί εύκολα στις ανάγκες κάθε εφαρμογής. Επίσης, υποστηρίζει τη χρήση εξωτερικών αρχείων για τη διαχείριση των μεταφράσεων, επιτρέποντας καλύτερη οργάνωση και συντήρηση. Επιπλέον, είναι αποδοτικό και σχεδιασμένο για να λειτουργεί καλά ακόμη και με μεγάλες και περίπλοκες εφαρμογές.

1.10.3 Οι κύριες λειτουργίες

- Υποστήριξη πολλών γλωσσών
- Μορφοποίηση ημερομηνιών και αριθμών σύμφωνα με τα τοπικά πρότυπα
- Δυνατότητες για πληθυντικό αριθμό και ενσωμάτωση δυναμικών δεδομένων στα μεταφρασμένα κείμενα

1.10.4 Επιπτώσεις και Μέλλον

Το vue-i18n έχει κάνει τη δημιουργία πολυγλωσσικών εφαρμογών πιο εύκολη και αποδοτική, συμβάλλοντας σημαντικά στην παγκόσμια διάδοση του Vue.js. Καθώς η ανάγκη για πολυγλωσσικές εφαρμογές συνεχίζει να αυξάνεται, αναμένεται ότι το vue-i18n θα συνεχίσει να εξελίσσεται και να προσαρμόζεται στις νέες απαιτήσεις και προκλήσεις.

1.11 Συμπέρασμα

Η συνδυαστική χρήση αυτών των τεχνολογιών συμβάλλει στην δημιουργία μιας ολοκληρωμένης (full-stack) και αποδοτικής εφαρμογής. Το Laravel παρέχει το θεμέλιο για την ανάπτυξη, ενώ το Filament διευκολύνει τη διαχείριση των διοικητικών διεπαφών. Η MySQL εξασφαλίζει την αποθήκευση και διαχείριση των δεδομένων, το Vue.js επιτρέπει τη δημιουργία εντυπωσιακών και δυναμικών διεπαφών ενώ με το πακέτο vue-i18n δημιουργείς μια πολυγλωσσική εφαρμογή, και το RestAPI διασφαλίζει την επικοινωνία μεταξύ πελατών και διακομιστών. Αυτές οι τεχνολογίες επιτρέπουν την υλοποίηση εφαρμογών που είναι ευέλικτες, ασφαλείς, αποδοτικές και συντηρήσιμες.

ΚΕΦΑΛΑΙΟ 3

Εισαγωγή

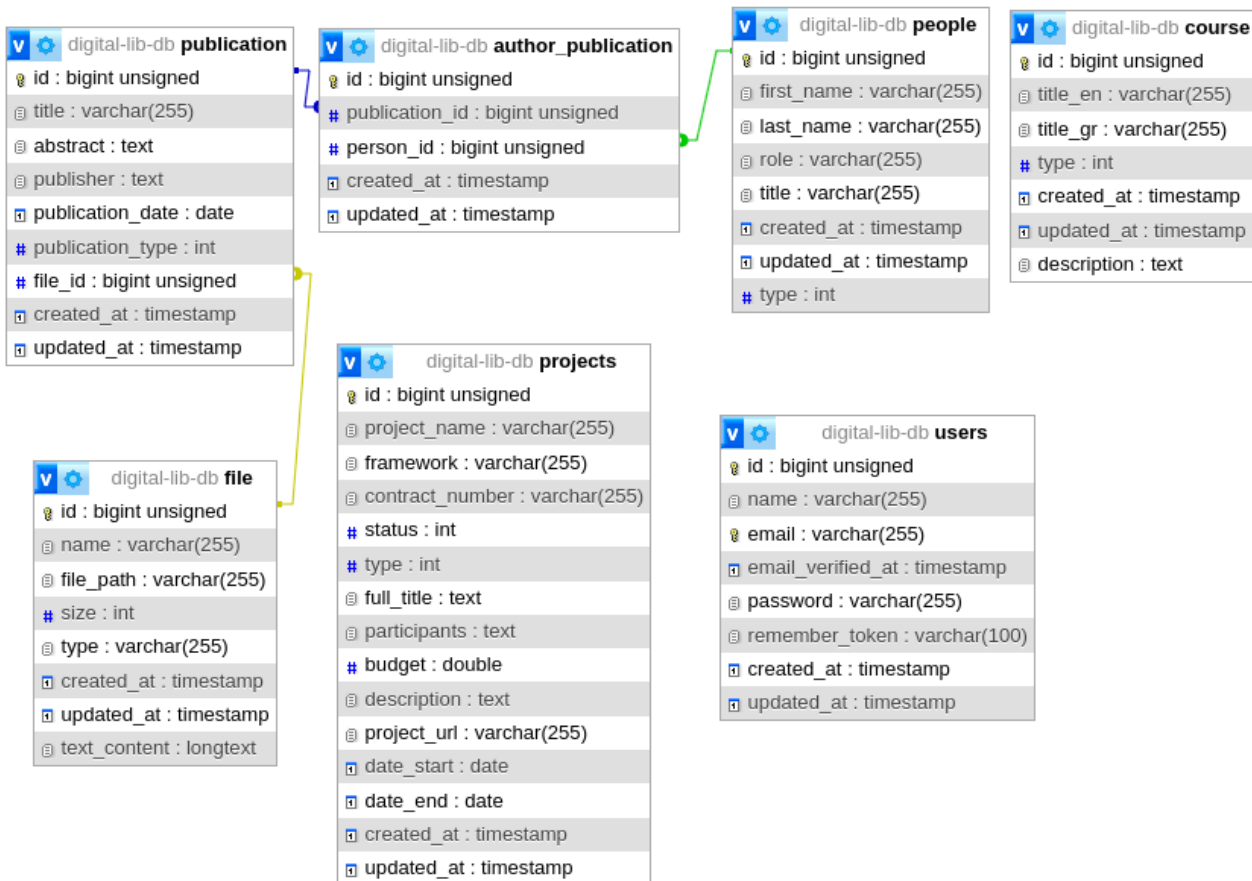
Ο στόχος αυτού του κεφαλαίου είναι να παρουσιαστεί η εφαρμογή των τεχνολογιών που παρουσιάστηκαν στο προηγούμενο κεφάλαιο για ανάπτυξη ολοκληρωμένης web-based εφαρμογής και αναλυτικά τα βήματα που έγιναν για ολοκλήρωση της διπλωματικής εργασίας από την τεχνική άποψη.

3.1 Βάση δεδομένων

Η ανάλυση, μελέτη και η σχεδίαση της βάσης δεδομένων σε ένα τέτοιο έργο όπως αυτό (custom web app) αποτελεί το θεμέλιο της εφαρμογής.

Το πρώτο βήμα που έγινε στην αρχή της διπλωματικής ήταν αυτό. Ο σκοπός αυτού του υποκεφαλαίου είναι να παρουσιάσω την βάση δεδομένων αναλυτικά με τους πίνακες και τα πεδία.

3.2 Το βασικό σχήμα της βάσης δεδομένων



Εικόνα 2 Το βασικό σχήμα της βάσης δεδομένων

Στην παραπάνω εικόνα απεικονίζεται το βασικό σχήμα της βάσης δεδομένων της εφαρμογής καθώς και όλοι οι πίνακες που χρησιμοποιήθηκαν για την ανάπτυξή της εφαρμογής.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

3.2.1 Πίνακας projects

Στον Πίνακα “Projects” αποθηκεύονται όλες οι πληροφορίες ότι αφορά σχετικά με τα έργα και οτι πληροφορία χρειάζεται για την απεικόνιση των έργων.

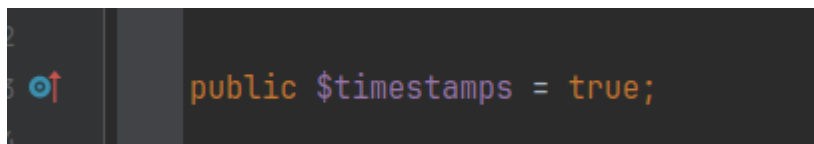
Αναλυτικά ο Πίνακας με τους τύπους δεδομένων:

Column	Type	null	Default	Comments
Id(PK)	bigint	No	None	AUTO_INCREMENT
project_name	varchar(255)	No	None	
framework	varchar(255)	No	None	
contract_number	varchar(255)	No	None	
status	int	No	None	
type	int	No		EU / GR
full_title	text	No	None	
participants	text	No	None	
budget	double	No	None	
description	text	No	None	
project_url	varchar(255)	Yes	NULL	
date_start	date	No	None	
date_end	date	No	None	
created_at	timestamp	No	NULL	
updated_at	timestamp	No	NULL	

Πίνακας 1 – Projects

Όπως φαίνεται από τον πίνακα όλα τα πεδία είναι υποχρεωτικά εκτός από τα πεδία project_url, το οποίο εξ αρχής ήταν υποχρεωτικό αλλά κατά την εισαγωγή δεδομένων στην εφαρμογή παρατήρησα ότι πολλά έργα δεν έχουν δικά τους ιστότοπους, οπότε πάρθηκε απόφαση να γίνει μη υποχρεωτικό για να μπορούν να εισαχθούν περισσότερα έργα.

Τα πεδία created_at & updated_at επίσης δεν είναι υποχρεωτικά, τα πεδία αυτά είναι timestamps και ενημερώνονται αυτόματα από το Laravel. Γράφοντας μόνο μια γραμμή κώδικα στο αντίστοιχο Model:



```
public $timestamps = true;
```

Εικόνα 3 Model: Timestamp

Τα πεδία status & type είναι int για τον λόγο ότι στην πραγματικότητα αποθηκεύονται τα “id” τους και όχι το λεκτικό που απεικονίζεται στην εφαρμογή.

Αυτό εφαρμόζεται ως εξής:

Φτιάχνουμε στο Model, τα constants με τα “id” τους:

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
1 usage
const TYPE_EU = 1;

1 usage
const TYPE_GR = 2;

1 usage
const STATUS_IN_PROGRESS = 1;

1 usage
const STATUS_COMPLETED = 2;
```

Εικόνα 4 Models:constants

Και μετά γυρνάμε ένα associative array από μια μέθοδο μέσα από το Model:

```
57 /**
58  * @return string[]
59  */
60 bogdanGR
61 public static function getTypes(): array
62 {
63     return [
64         self::TYPE_EU => 'EU Funding',
65         self::TYPE_GR => 'Greek Funding',
66     ];
67 }
68 /**
69  * @return string[]
70  */
71 bogdanGR
72 3 usages bogdanGR
73 public static function getStatuses(): array
74 {
75     return [
76         self::STATUS_IN_PROGRESS => 'In Progress',
77         self::STATUS_COMPLETED => 'Completed',
78     ];
79 }
```

Εικόνα 5 Models: Μέθοδοι

Έτσι ο κώδικας και γενικά η εφαρμογή γίνεται πιο επεκτάσιμη, καθώς και πιο αποτελεσματική, αποφεύγουμε έτσι επίσης τα λάθη κατά την διάρκεια των SQL ή ORM ερωτημάτων στην βάση δεδομένων, ειδικά σε περίπτωση που η εφαρμογή είναι πολύγλωσση.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

3.2.2 Πίνακας people

Στον Πίνακα “people” αποθηκεύονται οι πληροφορίες που απεικονίζονται στην σελίδα “Προσωπικό” καθώς και στην σελίδα “ Δημοσιεύσεις”, αφού αυτός ο πίνακας σχετίζεται με τον πίνακα publications ο οποίος θα παρουσιαστεί στα επόμενα υποενότητες.

Αναλυτική παρουσίαση του πίνακα με τους τύπους δεδομένων του εκάστοτε πεδίου, με σχόλια:

Column	Type	null	Default	Comments
id	bigint	No	None	AUTO_INCREMENT
first_name	varchar(255)	No	None	
last_name	varchar(255)	No	None	
role	varchar(255)	Yes	NULL	Faculty, Researchers
title	varchar(255)	Yes	NULL	Assistant, professor
created_at	timestamp	Yes	None	
updated_at_at	timestamp	Yes	NULL	
type	int	No	NULL	

Πίνακας 2 People

Τα βασικά πεδία αυτού του πίνακα είναι first_name, last_name καθώς και type για να προσδιορίσουμε τον τύπου ή αλλιώς την κατηγορία η οποία ανήκει του ανθρώπου.

Ο χειρισμός του πεδίου “type” γίνεται με τον ίδιο τρόπο όπως για παράδειγμα το status στον πίνακα projects.

Η μέθοδος που επιστρέφει όλους τους διαθέσιμους τύπους στην εφαρμογή:

```
public static function getTypes(): array
{
    return [
        self::TYPE_AUTHOR => 'Author',
        self::TYPE_STUDENT => 'Student',
        self::TYPE_TEACHER => 'Teacher',
        self::TYPE_LABORATORY_TEACHING_STAFF => 'Laboratory Teaching Staff'
    ];
}
```

Εικόνα 6 Models: getTypes μέθοδος

Η απεικόνιση του ονοματεπώνυμου γίνεται με διαδικασία “concat” δηλαδή “κολλάμε” το όνομα και το επώνυμο μαζί και επιστρέφουμε σε ένα άλλο “εικονικό” πεδίο “full_name”.

3.2.3 Πίνακας Course

Στον πίνακα course αποθηκεύονται τα μαθήματα που φαίνεται στην αντίστοιχη σελίδα στην εφαρμογή.

Παρακάτω ο αναλυτικός πίνακας με τους τύπους των δεδομένων και σχολιασμούς:

Column	Type	null	Default	Comments
id	bigint	No	None	AUTO_INCREMENT
title_en	varchar(255)	No	None	
title_gr	varchar(255)	No	None	
type	int	No	None	Undergraduate Graduate
description	text	Yes	NULL	
created_at	timestamp	Yes	NULL	
updated_at_at	timestamp	Yes	NULL	

Πίνακας 3 Course

Εδώ έχουμε ξεχωριστά τον τίτλο στα αγγλικά & στα ελληνικά για να τα δείξουμε και τα 2 ταυτόχρονα, Όπως και σε στον πίνακα reople και εδώ τα βασικότερα και τα υποχρεωτικά πεδία είναι οι τίτλοι και ο τύπος, για να προσδιορίσουμε σωστά την κάθε εγγραφή.

Επίσης προαιρετικά υπάρχει δυνατότητα να προσθέσουμε μια περιγραφή που έχει τύπος δεδομένων “text” που σημαίνει ότι μπορεί να πάρει αρκετά μεγάλο αριθμό λέξεως, σε σχέση με τους τίτλους που παίρνουν μέχρι 255 χαρακτήρες.

3.2.4 Πίνακας Publication

Ο πίνακας αυτός είναι υπεύθυνος να κρατάει όλες τις δημοσιεύσεις της εφαρμογής και περιλαμβάνει τα εξής πεδία:

Column	Type	null	Default	Comments
id	bigint	No	None	AUTO_INCREMENT
title	varchar(255)	No	None	
abstract	text	No	None	
publisher	text	No	None	
publication_date	date	No	None	
publication_type	int	No	None	
file_id(FK)	bigint	Yes	NULL	
created_at	timestamp	Yes	NULL	
updated_at	timestamp	Yes	NULL	

Πίνακας 4 Publication

Όλα τα πεδία είναι υποχρεωτικά για την σωστή απεικόνιση της κάθε δημοσίευσης εκτός από το file_id, created_at & updated_at.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

Το πεδίο `field_id` σχετίζεται με τον πίνακα `file`, έτσι ώστε να μπορούμε να πάρουμε όλες τις πληροφορίες από τον πίνακα αυτό.

3.2.5 Πίνακας `file`

Αυτός ο πίνακας κρατάει όλες τις απαραίτητες πληροφορίες ότι αφορά τα αρχεία που έχουν ανέβει στην εφαρμογή από τον διαχειριστή του συστήματος.

Τα πεδία που έχει με τους τύπους δεδομένων:

Column	Type	null	Default	Comments
id	bigint	No	None	AUTO_INCREMENT
name	varchar(255)	No	None	
file_path	text	No	None	
size	text	No	None	
type	date	No	None	
text_content	longtext	Yes	NULL	
created_at	timestamp	Yes	NULL	
updated_at	timestamp	Yes	NULL	

Πίνακας 5 File

Κρατάμε το όνομα του αρχείου που ανέβηκε όπως είναι στο πεδίο “name”, στο πεδίο “file_path” κρατάμε την διαδρομή που αποθηκεύτηκε το αρχείο μέσα στο σύστημα, έτσι ώστε να το χρησιμοποιήσουμε μετά στο frontend για να μπορεί ο χρήστης να δει το αρχείο αλλά και να το κατεβάσει. Κρατάμε επίσης το μέγεθος του αρχείου καθώς και το mime-type του, στην περίπτωση αυτής την εφαρμογής όλα είναι “application/octet-stream” καθώς επιτρέπεται να ανεβάσεις μόνο PDFs αρχεία, και ίσως το σημαντικότερο πεδίο αυτού του πίνακα είναι το “text_content” το οποίο έχει τύπο δεδομένων “longtext” το οποίο χρησιμοποιείται για να αποθηκεύσει το περιεχόμενο του αρχείου σαν κείμενο σε αυτό το πεδίο, έτσι ώστε να γίνει δυνατή η αναζήτηση μέσα στο κείμενο του αρχείου, αυτή η λειτουργία παρουσιάζεται στο επόμενο κεφάλαιο, στην σελίδα “Δημοσιεύσεις”.

3.2.6. Πίνακας `author_publication`

Ο πίνακας “author_publication” είναι ένα pivot table, που δημιουργήθηκε με σκοπό να έχουμε δυνατότητα να έχουμε σχέση N:M δηλαδή πολλά προς πολλά.

Column	Type	null	Default	Comments
id	bigint	No	None	AUTO_INCREMENT
publication_id(FK)	bigint	No	None	
person_id(FK)	bigint	No	None	
created_at	timestamp	Yes	NULL	
updated_at	timestamp	Yes	NULL	

Πίνακας 6 Author_publication

Αυτός ο πίνακας σχετίζει τις δημοσιεύσεις με τους “συγγραφείς” δηλαδή τον πίνακα `publication` (id) & `person_id` από τον πίνακα `people`.

3.2.7. Πίνακας `users`

Και τέλος ο πίνακας “users” είναι ένας default πίνακας, που παρέχει αυτόματα το Laravel.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

Και έχει από default τα εξής πεδία:

Column	Type	null	Default	Comments
id	bigint	No	None	AUTO_INCREMENT
name	bigint	No	None	
email	bigint	No	None	
email_verified_at	timestamp	Yes	NULL	
password	timestamp	Yes	NULL	
remember_token	varchar(100)	Yes	NULL	
created_at	timestamp	Yes	NULL	
updated_at	timestamp	Yes	NULL	

Πίνακας 7 Users

Ο σκοπός του πίνακα είναι να κρατάει όλους τους χρήστες του συστήματος στην βάση δεδομένων,

παρ' όλ' αυτά αυτά δεν παίζει πολύ μεγάλο ρόλο στην παρούσα εφαρμογή, καθώς κρατάμε μόνο έναν χρήστη (τον διαχειριστή) του συστήματος, ο οποίος είναι ο μόνος που μπορεί να ανεβάζει/ενημερώνει τα δεδομένα της παρούσας εφαρμογής.

3.3 Default πίνακες του Laravel

Το Laravel είναι ένα πολύ ισχυρό Framework όπως περιέγραψα σε προηγούμενο κεφάλαιο, και μόλις το εγκαταστήσεις περιλαμβάνει πολλά έτοιμα εργαλεία καθώς και λειτουργίες.

3.3.1 Το σχήμα

Table Name	Fields
digital-lib-db failed_jobs	id : bigint unsigned, uuid : varchar(255), connection : text, queue : text, payload : longtext, exception : longtext, failed_at : timestamp
digital-lib-db personal_access_tokens	id : bigint unsigned, tokenable_type : varchar(255), tokenable_id : bigint unsigned, name : varchar(255), token : varchar(64), abilities : text, last_used_at : timestamp, expires_at : timestamp, created_at : timestamp, updated_at : timestamp
digital-lib-db sessions	id : varchar(255), user_id : bigint unsigned, ip_address : varchar(45), user_agent : text, payload : longtext, last_activity : int
digital-lib-db migrations	id : int unsigned, migration : varchar(255), batch : int
digital-lib-db cache_locks	key : varchar(255), owner : varchar(255), expiration : int
digital-lib-db cache	key : varchar(255), value : mediumtext, expiration : int
digital-lib-db password_reset_tokens	email : varchar(255), token : varchar(255), created_at : timestamp
digital-lib-db jobs	id : bigint unsigned, queue : varchar(255), payload : longtext, attempts : tinyint unsigned, reserved_at : int unsigned, available_at : int unsigned, created_at : int unsigned
digital-lib-db job_batches	id : varchar(255), name : varchar(255), total_jobs : int, pending_jobs : int, failed_jobs : int, failed_job_ids : longtext, options : mediumtext, cancelled_at : int, created_at : int, finished_at : int

Εικόνα 7 Default πίνακες του Laravel

Παραπάνω παρουσιάζονται οι default πίνακες του Laravel, τα οποία υπάρχουν με το που εγκαταστήσεις το Laravel.

Αυτοί οι πίνακες είναι βοηθητικοί και ο κάθε πίνακας έχει τον σκοπό του.

3.3.2 Πίνακας failed_jobs

Αυτός ο πίνακας αποθηκεύει πληροφορίες σχετικά με τις εργασίες που απέτυχαν κατά την εκτέλεση.

Παρακάτω παρουσιάζεται αναλυτικά ο πίνακας με τα πεδία του, τύποι δεδομένων του κάθε πεδίου, αν μπορεί να είναι null ή όχι, οι προκαθορισμένες τιμές και τα σχόλια για τα πεδία.

Column	Type	null	Default	Comments
id	bigint	No	None	AUTO_INCREMENT
uuid	varchar(255)	No	None	Καθολικά μοναδικό αναγνωριστικό για την αποτυχημένη εργασία.
connection	text	No	None	Το όνομα της σύνδεσης στην οποία στάλθηκε η εργασία.
queue	text	No	None	Το όνομα της ουράς αναμονής στην οποία στάλθηκε η εργασία.
payload	longtext	No	None	
exception	longtext	No	None	Το μήνυμα σφάλματος από την αποτυχημένη εργασία.
failed_at	timestamp	No	Current timestamp	Πότε απέτυχε η εργασία.

Πίνακας 8 Failed_jobs

3.3.3 Πίνακας personal_access_tokens

Αυτός ο πίνακας χρησιμοποιείται από το Laravel Sanctum για την αποθήκευση κωδικών API για την πιστοποίηση των χρηστών.

Column	Type	null	Default	Comments
id	bigint	No	None	AUTO_INCREMENT
tokenable_type(FK)	varchar(255)	No	None	Ο τύπος του μοντέλου με το οποίο σχετίζεται το διακριτικό (π.χ. App\Models\User).
tokenable_id(FK)	bigint	No	None	Το αναγνωριστικό του μοντέλου με το οποίο σχετίζεται το token.
name	varchar(255)	No	None	Το όνομα του token.
Token(FK)	varchar(64)	No	None	Το κατακεραματισμένο token.
abilities	text	No	Null	Πεδίο JSON που απαριθμεί τις ικανότητες που έχουν παραχωρηθεί στο διακριτικό.
last_used_at	timestamp	No	Null	Χρονοσφραγίδα για το πότε χρησιμοποιήθηκε για τελευταία φορά το token.
expires_at	timestamp		Null	Χρονοσφραγίδα για το πότε λήγει το token.
created_at	timestamp		Null	Χρονοσφραγίδα για το πότε δημιουργήθηκε το token
updated_at	timestamp		Null	Χρονοσφραγίδα για το πότε ενημερώθηκε τελευταία φορά το token.

Πίνακας 9 Personal_access_tokens

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

3.3.4 Πίνακας sessions

Ο πίνακας αυτός αποθηκεύει δεδομένα των sessions για τους χρήστες.

Ακολουθούν οι στήλες του:

Column	Type	null	Default	Comments
Id(PK)	varchar(255)	No	None	
user_id	bigint	Yes	Null	Το αναγνωριστικό του χρήστη που σχετίζεται με τη session.
ip_address	varchar(45)	Yes	Null	Η διεύθυνση IP από την οποία προήλθε session
user_agent	text	Yes	Null	Η συμβολοσειρά User Agent του προγράμματος περιήγησης που χρησιμοποιήθηκε για τη session.
payload	longtext	No	None	
last_activity(FK)	int	No	None	Χρονοσφραγίδα για την τελευταία δραστηριότητα στη session.

Πίνακας 10 Sessions

3.4.5 Πίνακας migrations

Αυτός ο πίνακας είναι ένας από τους πιο σημαντικούς στο Laravel καθώς παρακολουθεί όλες τις “migrations” που έχουν εκτελεστεί.

Παρακάτω ο πίνακας με τα πεδία του και με τους τύπων δεδομένων, καθώς και τα σχόλια.

Column	Type	null	Default	Comments
Id(PK)	int	No	None	Auto increment
migration	bigint	Yes	Null	Το όνομα του migration
batch	varchar(45)	Yes	Null	Ο αριθμός παρτίδας στην οποία ανήκει η migration.

Πίνακας 11 Migrations

*Η έννοια των migrations θα αναλυθεί στις επόμενες υποενότητες.

3.4.6 Πίνακας password_reset_tokens

Αυτός ο πίνακας αποθηκεύει tokens επαναφοράς κωδικού πρόσβασης των χρηστών.

Column	Type	null	Default	Comments
Email(PK)	varchar(255)	No	None	Η διεύθυνση ηλεκτρονικού ταχυδρομείου (email) του χρήστη που ζητά την επαναφορά του κωδικού πρόσβασης.
token	varchar(255)	No	None	Το διακριτικό επαναφοράς κωδικού πρόσβασης.
created_at	varchar(45)	Yes	Null	Χρονοσφραγίδα για το πότε δημιουργήθηκε η αίτηση επαναφοράς κωδικού πρόσβασης.

Πίνακας 12 Password_reset_tokens

3.4.7 Πίνακας jobs

Αυτός ο πίνακας αποθηκεύει τις εργασίες που βρίσκονται στην ουρά.

Column	Type	null	Default	Comments
--------	------	------	---------	----------

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

id	varchar(255)	No	None	Auto increment
queue	varchar(255)	No	None	Το όνομα της ουράς αναμονής στην οποία στάλθηκε η εργασία.
payload	longtext	No	None	
attempts	tinyint	No	None	Ο αριθμός των προσπαθειών που έγιναν για την επεξεργασία της εργασίας.
reserved_at	int	Yes	Null	Χρονοσφραγίδα για το πότε κρατήθηκε η εργασία.
available_at	int	No	None	Χρονοσφραγίδα για το πότε η εργασία θα είναι διαθέσιμη.
created_at	int	No	None	Χρονοσφραγίδα για το πότε δημιουργήθηκε η εργασία.

Πίνακας 13 Jobs

3.4.8 Πίνακες Cache & cache_locks

Αυτοί οι δυο πίνακες είναι υπεύθυνοι για την cache στην εφαρμογή.

Πίνακας `cache_locks`:

Column	Type	null	Default	Comments
Key(PK)	varchar(255)	No	None	Το κλειδί για την κλειδαριά.
owner	varchar(255)	No	None	Ο ιδιοκτήτης της κλειδαριάς.
expiration	int	No	None	Η χρονοσφραγίδα λήξης του κλειδώματος.

Πίνακας 14 Cache_locks

Ο πίνακας “`cache`”

Column	Type	null	Default	Comments
Key(PK)	varchar(255)	No	None	Το κλειδί για τα αποθηκευμένα δεδομένα.
value	mediumtext	No	None	Serialized αποθηκευμένα δεδομένα.
expiration	int	No	None	Η χρονοσφραγίδα λήξης των αποθηκευμένων δεδομένων.

Πίνακας 15 Cache

3.4.9 job_batches

Αυτός ο πίνακας αποθηκεύει πληροφορίες σχετικά με τις μαζικές εκτελέσεις των εργασιών.

Ακολουθούν οι στήλες του:

Column	Type	null	Default	Comments
Id(PK)	varchar(255)	No	None	
name	varchar(255)	No	None	Το όνομα της μαζικής εργασίας.
total_jobs	int	No	None	Ο συνολικός αριθμός των εργασιών της παρτίδας.
pending_jobs	int	No	None	Ο αριθμός των εκκρεμών εργασιών της παρτίδας.
failed_jobs	int	Yes	Null	Ο αριθμός των αποτυχημένων εργασιών στη παρτίδα.
failed_job_ids	longtext	No	None	Πεδίο JSON που αποθηκεύει τα αναγνωριστικά των αποτυχημένων εργασιών.
options	mediumtext	No	None	Πεδίο JSON που αποθηκεύει επιλογές για τη παρτίδα εργασιών.
cancelled_at	int	Yes	Null	Χρονοσφραγίδα για το πότε δημιουργήθηκε η παρτίδα εργασιών.

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

created_at	int	No	None	Χρονοσφραγίδα για το πότε ακυρώθηκε η παρτίδα εργασιών.
finished_at	int	Yes	Null	Χρονοσφραγίδα για το πότε ολοκληρώθηκε η παρτίδα εργασιών.

Πίνακας 16 Job_batches

3.5 Ανάπτυξη backend

Ανάπτυξη του backend κομμάτι της εφαρμογής έγινε με χρήση docker & docker-compose έτσι ώστε να γίνει δυνατή η εγκατάσταση της τελευταίας έκδοσης Laravel 11, καθώς υποστηρίζει η php εκδόσεις 8.2 & 8.3, ενώ λόγω δουλειάς χρησιμοποιώ πιο παλιά έκδοση της php.

3.5.1 Docker - περιγραφή

Το Docker είναι μια πλατφόρμα λογισμικού που επιτρέπει στους προγραμματιστές να δημιουργούν, να διανέμουν και να εκτελούν εφαρμογές σε απομονωμένα περιβάλλοντα, γνωστά ως containers. Τα containers είναι ελαφριά και απομονωμένα πακέτα λογισμικού που περιλαμβάνουν όλα τα απαραίτητα εξαρτήματα για την εκτέλεση μιας εφαρμογής, όπως κώδικας, βιβλιοθήκες και εξαρτήσεις (dependencies). Αυτό εξασφαλίζει ότι η εφαρμογή θα τρέχει ακριβώς με τον ίδιο τρόπο σε διαφορετικά περιβάλλοντα χωρίς προβλήματα συμβατότητας.

3.5.2 Docker Βασικά χαρακτηριστικά

- **Απομόνωση και Ασφάλεια:** Τα containers παρέχουν απομόνωση εφαρμογών, εξασφαλίζοντας ότι οι εφαρμογές δεν επηρεάζονται μεταξύ τους.
- **Φορητότητα:** Τα Docker images μπορούν να μεταφερθούν και να εκτελεστούν σε οποιοδήποτε σύστημα που υποστηρίζει Docker.
- **Αποδοτικότητα Πόρων:** Τα containers είναι ελαφριά και χρησιμοποιούν λιγότερους πόρους σε σύγκριση με τις παραδοσιακές εικονικές μηχανές (VMs).

3.5.3 Docker Compose - περιγραφή

Το Docker Compose είναι ένα εργαλείο για τον ορισμό και τη διαχείριση Docker εφαρμογών με πολλαπλά containers. Με το Docker Compose, οι προγραμματιστές μπορούν να καθορίσουν τις υπηρεσίες, τα δίκτυα και τους όγκους που χρειάζονται για την εφαρμογή τους σε ένα μόνο αρχείο YAML. Στη συνέχεια, με μια απλή εντολή, μπορούν να ξεκινήσουν και να διαχειριστούν όλη την εφαρμογή.

3.5.4 Docker Compose - Κύρια χαρακτηριστικά

- Η δημιουργία και η διαχείριση πολλών containers γίνεται μέσω ενός απλού αρχείου YAML.
- Ομαδική Εκκίνηση: Επιτρέπει την εκκίνηση και τη διαχείριση όλων των υπηρεσιών της εφαρμογής με μια εντολή (docker-compose up).
- Επαναληψιμότητα: Διασφαλίζει ότι όλες οι υπηρεσίες της εφαρμογής θα ξεκινήσουν με τις ίδιες ρυθμίσεις κάθε φορά.

3.6 Laravel migrations

Μετά από την ανάλυση του έργου, σχεδίαση της βάσης δεδομένων, το επόμενο βήμα είναι η δημιουργία των migrations για να δημιουργηθούν οι πίνακες και τα πεδία.

3.6.1 Laravel migrations περιγραφή

Laravel migrations είναι ένα σύστημα διαχείρισης εκδόσεων για τη βάση δεδομένων στο πλαίσιο του Laravel. Χρησιμοποιούνται για τη δημιουργία, τροποποίηση και διαγραφή πινάκων και στηλών στη βάση δεδομένων, διευκολύνοντας την ανάπτυξη και τη διατήρηση της εφαρμογής. Οι migrations καταγράφονται σε αρχεία PHP και εκτελούνται μέσω του artisan, του ενσωματωμένου command-line interface του Laravel.

3.6.2 Laravel migrations πως λειτουργούν

Κάθε migration δημιουργείται μέσω της εντολής `php artisan make:migration`, η οποία παράγει ένα νέο αρχείο migration στο φάκελο `database/migrations`. Το αρχείο αυτό περιέχει δύο κύριες μεθόδους: `up` και `down`.

3.6.3 Laravel migrations - Η Μέθοδος `up`

Αυτή η μέθοδος περιγράφει τις αλλαγές που θα γίνουν στη βάση δεδομένων, όπως η δημιουργία νέων πινάκων ή η προσθήκη νέων στηλών.

3.6.4 Laravel migrations - Η Μέθοδος `down`

Αυτή η μέθοδος καθορίζει πώς θα αναιρεθούν οι αλλαγές που έγιναν από τη μέθοδο `up`. Συνήθως περιλαμβάνει την αναίρεση των ενεργειών της `up`, όπως η διαγραφή των πινάκων ή των στηλών που δημιουργήθηκαν.

3.6.5 Laravel migrations - Παραδείγματα

Παρακάτω απεικονίζονται τα παραδείγματα για δημιουργίας / διαγραφής του πίνακα `author_publication`.

Που περιλαμβάνει και δημιουργία των Foreign keys στις γραμμές 20 και 21.

Μέθοδος `up()`:

```
14 Schema::create( table: 'author_publication', function (Blueprint $table) {
15     $table->id();
16     $table->unsignedBigInteger( column: 'publication_id');
17     $table->unsignedBigInteger( column: 'person_id');
18     $table->timestamps();
19
20     $table->foreign( columns: 'publication_id')->references( columns: 'id')->on( table: 'publication')->onDelete( action: 'cascade');
21     $table->foreign( columns: 'person_id')->references( columns: 'id')->on( table: 'people')->onDelete( action: 'cascade');
22 });
23 }
```

Εικόνα 8 Migrations: μέθοδος `up()`

Μέθοδος `down()`:

```
28 public function down(): void
29 {
30     Schema::dropIfExists( table: 'author_publication');
31 }
32 };
```

Εικόνα 9 Migrations: μέθοδος `down()`

3.6.6 Εκτέλεση των Migrations

Για να εφαρμοστούν οι migrations στη βάση δεδομένων, χρησιμοποιείται η εντολή `php artisan migrate`. Αυτή η εντολή εκτελεί όλες τις εκκρεμείς migrations και ενημερώνει το αρχείο καταγραφής των migrations στη βάση δεδομένων.

3.6.7 Αναίρεση των Migrations

Εάν χρειάζεται να αναιρεθούν οι migrations, χρησιμοποιείται η εντολή `php artisan migrate:rollback`, η οποία εκτελεί τη μέθοδο `down` των τελευταίων εκτελεσμένων migrations.

3.6.8 Laravel migrations - Τα πλεονεκτήματα

- **Ευκολία στη Συνεργασία:** Οι αλλαγές στη βάση δεδομένων καταγράφονται και μπορούν να διαμοιραστούν εύκολα μεταξύ των μελών της ομάδας.
- **Επαναληψιμότητα:** Οι migrations μπορούν να εκτελεστούν και να αναιρεθούν πολλές φορές, διευκολύνοντας τη διαδικασία ανάπτυξης και δοκιμών.
- **Αυτοματοποίηση:** Η χρήση των migrations επιτρέπει την αυτοματοποίηση των αλλαγών στη βάση δεδομένων κατά την ανάπτυξη της εφαρμογής.

3.6.9 Laravel migrations - Συμπέρασμα

Laravel migrations αποτελούν ένα πολύ σημαντικό εργαλείο για την ανάπτυξη εφαρμογών, επιτρέποντας την ασφαλή και συστηματική διαχείριση των αλλαγών στη βάση δεδομένων.

3.7 Laravel Models

Μετά από τα migrations είναι σημαντικό να φτιαχτούν τα models για κάθε πίνακα που δημιουργήθηκε σε προηγούμενο βήμα, για να υπάρχει η δυνατότητα χειρισμού αυτών των πινάκων και να δηλωθούν βοηθητικοί μέθοδοι.

Ένα model μπορεί να δημιουργηθεί αυτόματα με την χρήση `console artisan`.

Εκτελώντας εντολή `php artisan make:model File` . Δημιουργείται το Model "File".

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

Παράδειγμα ενός Model με κώδικα:

```
3 namespace App\Models;
5 use ...
8
9 class File extends Model
10 {
11     use HasFactory;
12
13     protected $table = 'file';
14
15     no usages
16     protected $fillable = [
17         'name',
18         'file_path',
19         'size',
20         'type',
21         'text_content',
22     ];
23     public $timestamps = true;
24
25     /**
26      * @return HasMany
27      */
28     no usages bogdanGR
29     public function publications(): HasMany
30     {
31         return $this->hasMany(related: Publication::class);
32     }
33 }
```

Εικόνα 10 Models: ένα παράδειγμα

Το Laravel είναι αρκετά έξυπνο για να καταλάβει ακόμα και τις σχέσεις μεταξύ πινάκων, οπότε τρέχοντας μόνο την εντολή, έχουμε και έτοιμους μεθόδους με τις σχέσεις του.

Έτσι όταν τραβάμε από την βάση δεδομένων π.χ τα Files μπορούμε να πάρουμε και τις σχετικές δημοσιεύσεις που σχετίζονται.

Για την δημιουργία των αντικείμενων του κάθε Model πρέπει να δηλώσουμε fillable array που περιλαμβάνει όλα τα πεδία που θέλουμε να ενημερώσουμε κατά την δημιουργία και ενημέρωση εκάστοτε εγγραφής, αλλιώς δεν θα μπορούμε να το πετύχουμε.

ΚΕΦΑΛΑΙΟ 4

Ο σκοπός αυτού του κεφαλαίου είναι να παρουσιαστεί η εφαρμογή η οποία δημιουργήσα στα πλαίσιο της διπλωματικής εργασίας. Αναπτύχθηκε μια **fullstack**(frontend & backend) εφαρμογή - μια σύγχρονη ψηφιακή βιβλιοθήκη.

Αρχικά παρουσιάζεται με screenshots το frontend κομμάτι της εφαρμογής, είναι αυτό με το οποίο μπορεί να αλληλεπιδρά ο κάθε χρήστης.

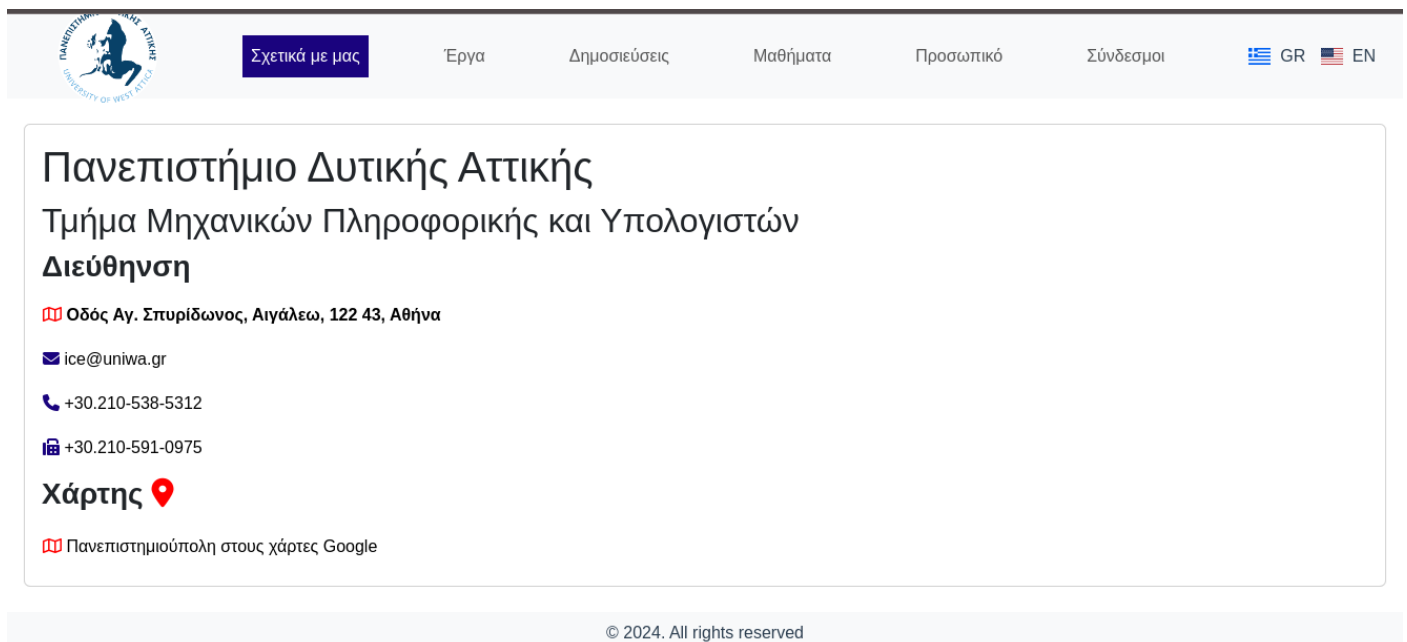
Μετά γίνεται η επισκόπηση στο 2ο κομμάτι της εφαρμογής που είναι το backend.

Με το backend μπορούν να αλληλεπιδρούν μόνο οι διαχειριστές (πιστοποιημένοι χρήστες).

Frontend

4.1.1 Σχετικά με μας

Σε αυτή την σελίδα ένας φοιτητής μπορεί να δει τις βασικές πληροφορίες για το πανεπιστήμιο, όπως την διεύθυνση, email, τηλέφωνο και φαξ.



ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

Σχετικά με μας Έργα Δημοσιεύσεις Μαθήματα Προσωπικό Σύνδεσμοι GR EN

Πανεπιστήμιο Δυτικής Αττικής

Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών

Διεύθυνση

📍 Οδός Αγ. Σπυριδωνος, Αιγάλεω, 122 43, Αθήνα

✉ ice@uniwa.gr

☎ +30.210-538-5312

☎ +30.210-591-0975

📍 Χάρτης

📍 Πανεπιστημιούπολη στους χάρτες Google

© 2024. All rights reserved

Εικόνα 11 Frontend - Σχετικά με μας

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

4.1.2 Έργα - Λίστα

Μια λίστα με τα αναπτυξιακά – εκπαιδευτικά έργα τα οποία χωρίζεται σε 4 κατηγορίες:

Ευρωπαϊκά σε εξέλιξη, Ευρωπαϊκά ολοκληρωμένα, Εθνικά σε εξέλιξη και Εθνικά ολοκληρωμένα.

Στην σελίδα αυτή προστέθηκαν επίσης και “tabs” με τα οποία μπορείς να φιλτράρεις και να δεις τα έργα ανά κατηγορία ή και όλα.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

Σχετικά με μας Έργα Δημοσιεύσεις Μαθήματα Προσωπικό Σύνδεσμοι GR EN

Όλα Ευρωπαϊκά σε εξέλιξη Ευρωπαϊκά ολοκληρωμένα Εθνικά σε εξέλιξη Εθνικά ολοκληρωμένα

Ευρωπαϊκά σε εξέλιξη

Πλαίσιο	H2020
Όνομα Έργου	GR digiGOV-Innohub
Πλήρες Όνομα Έργου	European Digital Innovation Hub for Digital Governance
Συμμετέχοντες	-
Προϋπολογισμός	5.821.600,00 €
Διάρκεια	2023 - 2025 (36 μήνες)

Πλαίσιο	CIP-ICT-PSP.2011.2.3
Όνομα Έργου	EAwareness
Πλήρες Όνομα Έργου	Europeana Awareness
Συμμετέχοντες	http://pro.europeana.eu/web/europeana-awareness/project-partners
Προϋπολογισμός	132.600,00 €
Διάρκεια	2021 - 2026 (72 μήνες)


Ευρωπαϊκά ολοκληρωμένα

Πλαίσιο	DG XVII
Όνομα Έργου	Horizontal Energy Renewable Multimedia Educational Software (HERMES)
Πλήρες Όνομα Έργου	«Horizontal Energy Renewable Multimedia Educational Software (HERMES)», Program ALTENER, XVII Directorate-General for Energy
Συμμετέχοντες	-
Προϋπολογισμός	0,00 €
Διάρκεια	1996 - 1997 (24 μήνες)

Εικόνα 12 Frontend - Λίστα με έργα

4.1.3 Έργο

Πατώντας στο “Όνομα έργου” ή πλήρες όνομα έργου” μπορεί ο χρήστης να δει τις λεπτομέρειες του έργου σε αυτή την σελίδα, όπως: “Contract Number”, περίληψη του έργου καθώς και τον σύνδεσμο του έργου.

Σχετικά με μας Έργα Δημοσιεύσεις Μαθήματα Προσωπικό Σύνδεσμοι 🇬🇷 GR 🇺🇸 EN

Framework	H2020
Project Name	GR digiGOV-Innohub
Contract Number	88888
Full Project Name	European Digital Innovation Hub for Digital Governance
Participants	-
Budget	5.821.600,00 €
Duration	2023 - 2025 (36 months)
Description	The European Digital Innovation Hub for Digital Governance – GR digiGOV-innoHUB supports the development of a new generation of public services based on advanced digital technologies (AI, ML, IoT, Blockchain, HPC), with open source, open data and open standards. GR digiGOV-innoHUB is coordinated by GRNET.
URL	https://digigov.innohub.gr/en/

© 2024. All rights reserved

Εικόνα 13 Frontend - Το έργο

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

4.1.4 Δημοσιεύσεις

Σε αυτή την σελίδα απεικονίζονται όλες οι δημοσιεύσεις που αποτελεί την κύρια ενότητα της εφαρμογής. Επίσης υπάρχουν τα φίλτρα με τα οποία μπορείς να φιλτράρεις την λίστα με βάση τον συγγραφέα, τον τίτλο, την περίληψη, τον εκδότη, με ημερομηνίες έκδοσης από και έως, μπορείς να φιλτράρεις επίσης με βάση το κείμενο(περιεχόμενο) της δημοσίευσης και με βάση τον τύπο της δημοσίευσης: Περιοδικό, βιβλίο, συνέδριο, Διπλωματική εργασία και Διδακτορική διατριβή.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

Σχετικά με μας Έργα **Δημοσιεύσεις** Μαθήματα Προσωπικό Σύνδεσμοι GR EN

Συγγραφέας Τίτλος Περίληψη Εκδότης

Από Έως Κείμενο¹ Τύπος

mm/dd/yyyy mm/dd/yyyy

¹ελάχιστο μήκος 2 χαρακτήρες

Αναζήτηση

Phivos Mylonas, Apostolos Maniatis, Evaggelos Spyrou
A Deep Learning Approach for Human Action Recognition Using Skeletal Information
PubMed • 2020 • Book
Περίληψη PDF




Phivos Mylonas
Semantic Processing
CRC Press • 2019 • Book
Περίληψη PDF

Phivos Mylonas
Detecting and Verifying Dissimilar Patterns in Unlabelled Data
- • 2003 • Conference

Εικόνα 14 Frontend - Λίστα δημοσιεύσεων

4.1.5 Δημοσίευση (Περίληψη)

Πατώντας Περίληψη στην σελίδα “Δημοσιεύσεις”, μπορείς να δεις αυτή την σελίδα όπου μπορείς να διαβάσεις την περίληψη της δημοσίευσης, καθώς και απεικονίζονται οι πληροφορίες όπως: οι συγγραφείς, τίτλος, τύπος δημοσίευσης, το έτος έκδοσης και μπορείς να δεις και να κατεβάσεις το pdf στον υπολογιστή ή το κινητό σου.

[Σχετικά με μας](#) [Έργα](#) [Δημοσιεύσεις](#) [Μαθήματα](#) [Προσωπικό](#) [Σύνδεσμοι](#)  GR  EN

Phivos Mylonas, Manolis Wallace, Stefanos Kollias

Using k-Nearest Neighbor and Feature Selection as an Improvement to Hierarchical Clustering

ABSTRACT

Clustering of data is a difficult problem that is related to various fields and applications. Challenge is greater, as input space dimensions become larger and feature scales are different from each other. Hierarchical clustering methods are more flexible than their partitioning counterparts, as they do not need the number of clusters as input. Still, plain hierarchical clustering does not provide a satisfactory framework for extracting meaningful results in such cases. Major drawbacks have to be tackled, such as curse of dimensionality and initial error propagation, as well as complexity and data set size issues. In this paper we propose an unsupervised extension to hierarchical clustering in the means of feature selection, in order to overcome the first drawback, thus increasing the robustness of the whole algorithm. The results of the application of this clustering to a portion of dataset in question are then refined and extended to the whole dataset through a classification step, using k-nearest neighbor classification technique, in order to tackle the latter two problems. The performance of the proposed methodology is demonstrated through the application to a variety of well known publicly available data sets.

Springer-Verlag Berlin Heidelberg • 2004 • Book • [PDF](#)

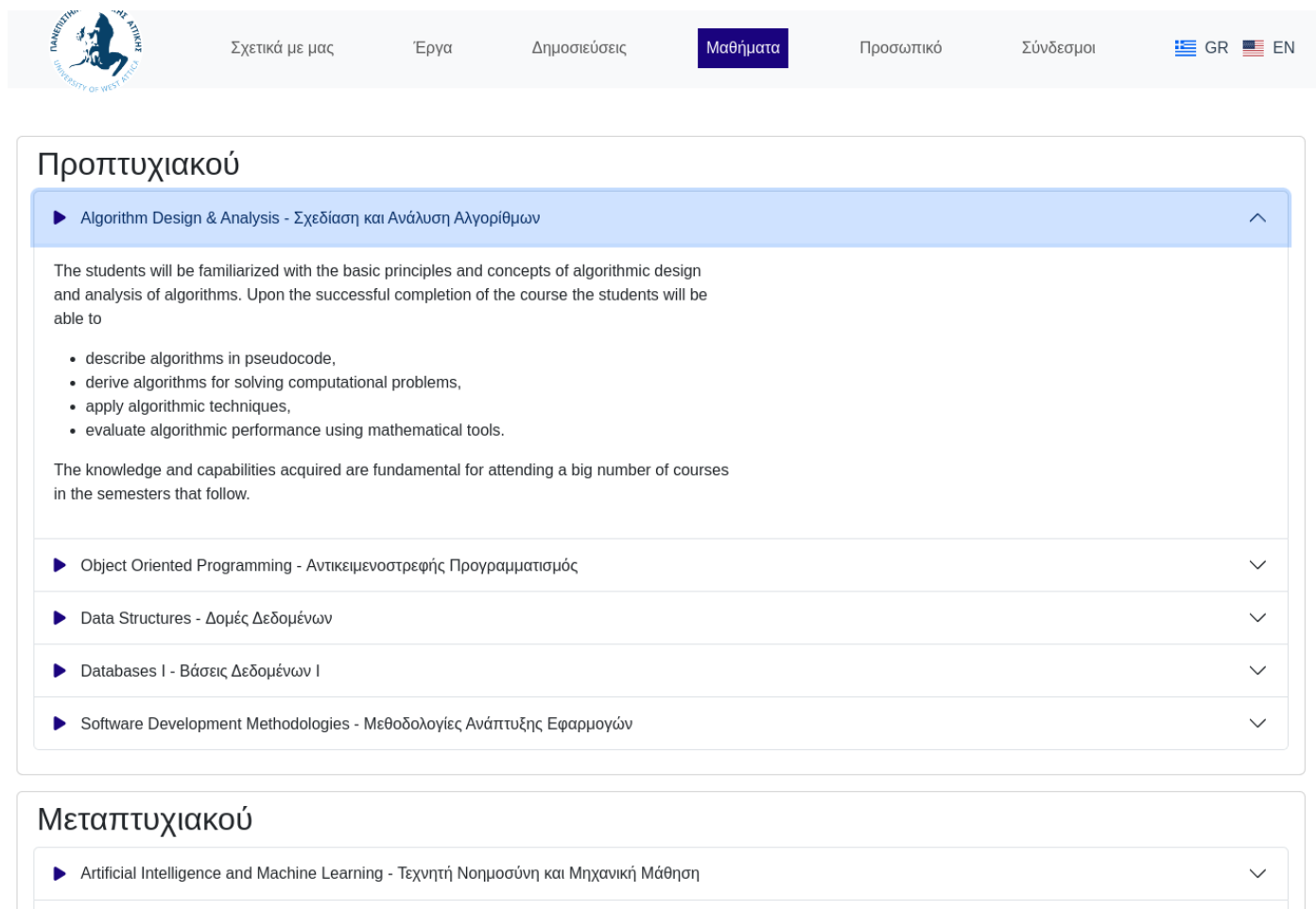
© 2024. All rights reserved

Εικόνα 15 Frontend - Δημοσίευση - Περίληψη

4.1.6 Μαθήματα

Σε αυτή την σελίδα απεικονίζονται τα μαθήματα του τμήματος και χωρίζονται σε Προπτυχιακού/ Μεταπτυχιακού προγράμματος σπουδών.

Έχει υλοποιηθεί και accordion, πατώντας πάνω σε ένα μάθημα μπορείς να δεις και τις λεπτομέρειες του εκάστοτε μαθήματος.



The screenshot shows the website of the University of West Attica. At the top, there is a navigation menu with the following items: "Σχετικά με μας", "Έργα", "Δημοσιεύσεις", "Μαθήματα" (highlighted in a dark blue box), "Προσωπικό", "Σύνδεσμοι", and language flags for "GR" and "EN".

The main content area is titled "Προπτυχιακού" (Undergraduate) and contains a list of courses in an accordion format. The first course, "Algorithm Design & Analysis - Σχεδίαση και Ανάλυση Αλγορίθμων", is expanded, showing its description and a list of learning objectives. The other courses are collapsed.

Προπτυχιακού

▶ Algorithm Design & Analysis - Σχεδίαση και Ανάλυση Αλγορίθμων

The students will be familiarized with the basic principles and concepts of algorithmic design and analysis of algorithms. Upon the successful completion of the course the students will be able to

- describe algorithms in pseudocode,
- derive algorithms for solving computational problems,
- apply algorithmic techniques,
- evaluate algorithmic performance using mathematical tools.

The knowledge and capabilities acquired are fundamental for attending a big number of courses in the semesters that follow.

▶ Object Oriented Programming - Αντικειμενοστρεφής Προγραμματισμός

▶ Data Structures - Δομές Δεδομένων

▶ Databases I - Βάσεις Δεδομένων I

▶ Software Development Methodologies - Μεθοδολογίες Ανάπτυξης Εφαρμογών

Μεταπτυχιακού

▶ Artificial Intelligence and Machine Learning - Τεχνητή Νοημοσύνη και Μηχανική Μάθηση

Εικόνα 16 Frontend - Λίστα μαθημάτων

4.1.7 Προσωπικό

Σε αυτή την σελίδα φαίνεται το προσωπικό του ιδρύματος καθώς και μπορούν να προστεθούν οι φοιτητές, μέλη ΔΕΠ και ΕΔΙΠ.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

Σχετικά με μας Έργα Δημοσιεύσεις Μαθήματα Προσωπικό Σύνδεσμοι

GR EN

Διδακτικό Ερευνητικό Προσωπικό (ΔΕΠ)

- Phivos Mylonas
- George Prezerakos
- Ioannis Voyiatzis

Φοιτητές

- Bogdan Vaskan
- Vaggelis Karamanos

Εργαστηριακό Διδακτικό Προσωπικό (ΕΔΙΠ)

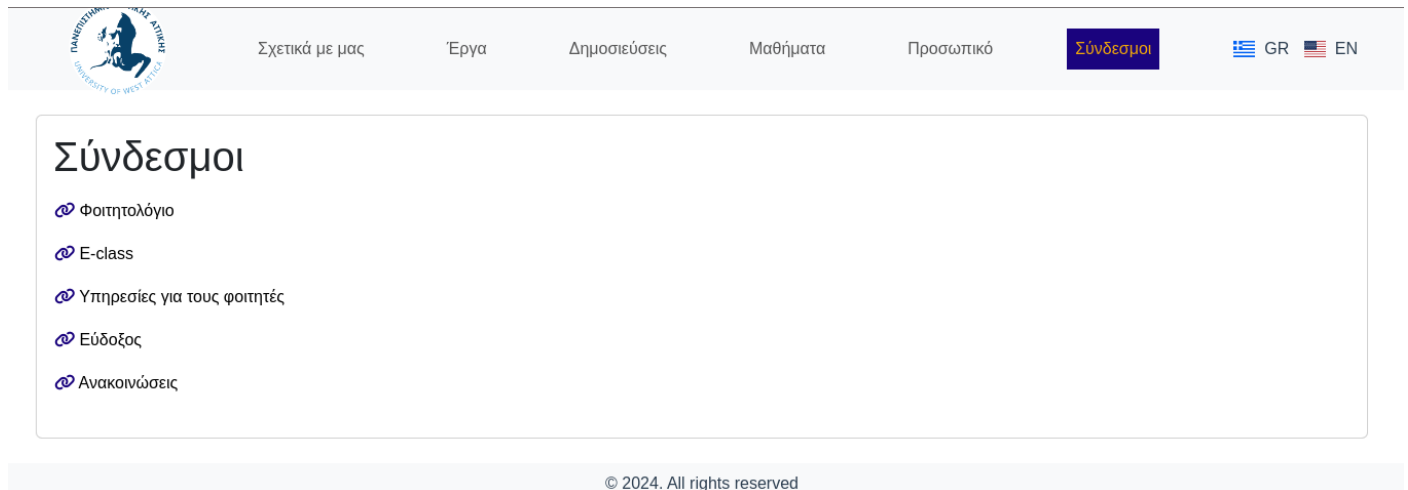
- Georgios Antoniou
- Zacharenia Garofalaki
- Akrivi Krouska
- Georgios Meletiou
- Angelos Mitsaras

© 2024. All rights reserved

Εικόνα 17 Frontend - Προσωπικό

4.1,8 Σύνδεσμοι

Σε αυτή την σελίδα απεικονίζονται χρήσιμοι σύνδεση που αφορούν το ίδρυμα, όπως το φοιτητολόγιο, e-class, ηλεκτρονικές υπηρεσίες για τους φοιτητές, εύδοξος και ανακοινώσεις του τμήματος.



Εικόνα 18 Frontend - Σύνδεσμοι

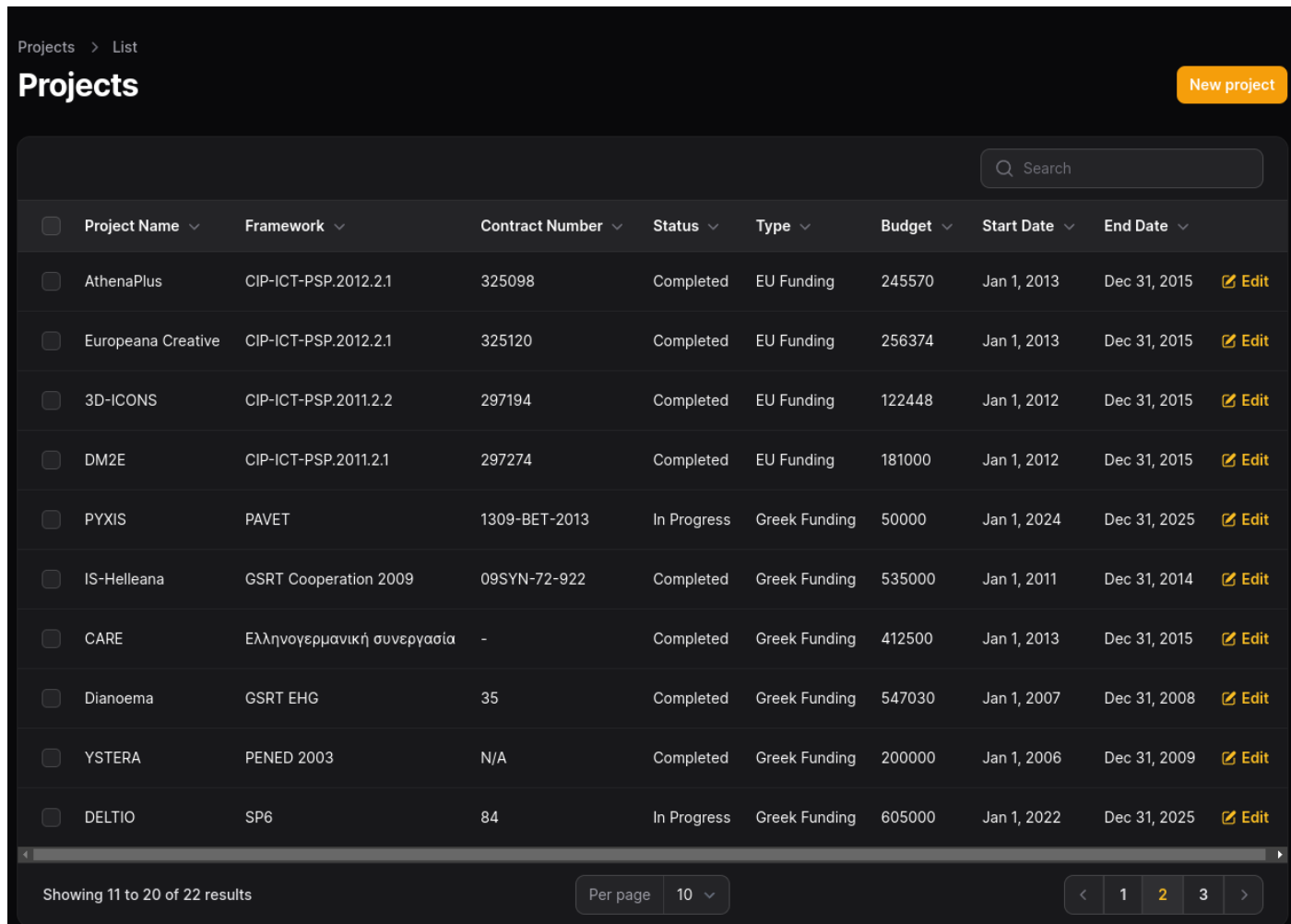
Backend (Admin Panel)

Εισαγωγή

Πρόσβαση στο διαχειριστικό έχουν μόνο οι διαχειριστές της εφαρμογής, μόνο αυτοί μπορούν να ανεβάζουν, επεξεργάζονται και διαγράφουν το υλικό στην εφαρμογή.

4.2.1. Έργα - λίστα

Σε αυτή την σελίδα ο διαχειριστής βλέπει όλη την λίστα με τα δημοσιευμένα έργα.



<input type="checkbox"/>	Project Name	Framework	Contract Number	Status	Type	Budget	Start Date	End Date	
<input type="checkbox"/>	AthenaPlus	CIP-ICT-PSP.2012.2.1	325098	Completed	EU Funding	245570	Jan 1, 2013	Dec 31, 2015	Edit
<input type="checkbox"/>	Europeana Creative	CIP-ICT-PSP.2012.2.1	325120	Completed	EU Funding	256374	Jan 1, 2013	Dec 31, 2015	Edit
<input type="checkbox"/>	3D-ICONS	CIP-ICT-PSP.2011.2.2	297194	Completed	EU Funding	122448	Jan 1, 2012	Dec 31, 2015	Edit
<input type="checkbox"/>	DM2E	CIP-ICT-PSP.2011.2.1	297274	Completed	EU Funding	181000	Jan 1, 2012	Dec 31, 2015	Edit
<input type="checkbox"/>	PYXIS	PAVET	1309-BET-2013	In Progress	Greek Funding	50000	Jan 1, 2024	Dec 31, 2025	Edit
<input type="checkbox"/>	IS-Helleana	GSRT Cooperation 2009	09SYN-72-922	Completed	Greek Funding	535000	Jan 1, 2011	Dec 31, 2014	Edit
<input type="checkbox"/>	CARE	Ελληνογερμανική συνεργασία	-	Completed	Greek Funding	412500	Jan 1, 2013	Dec 31, 2015	Edit
<input type="checkbox"/>	Dianoema	GSRT EHG	35	Completed	Greek Funding	547030	Jan 1, 2007	Dec 31, 2008	Edit
<input type="checkbox"/>	YSTERA	PENED 2003	N/A	Completed	Greek Funding	200000	Jan 1, 2006	Dec 31, 2009	Edit
<input type="checkbox"/>	DELTIO	SP6	84	In Progress	Greek Funding	605000	Jan 1, 2022	Dec 31, 2025	Edit

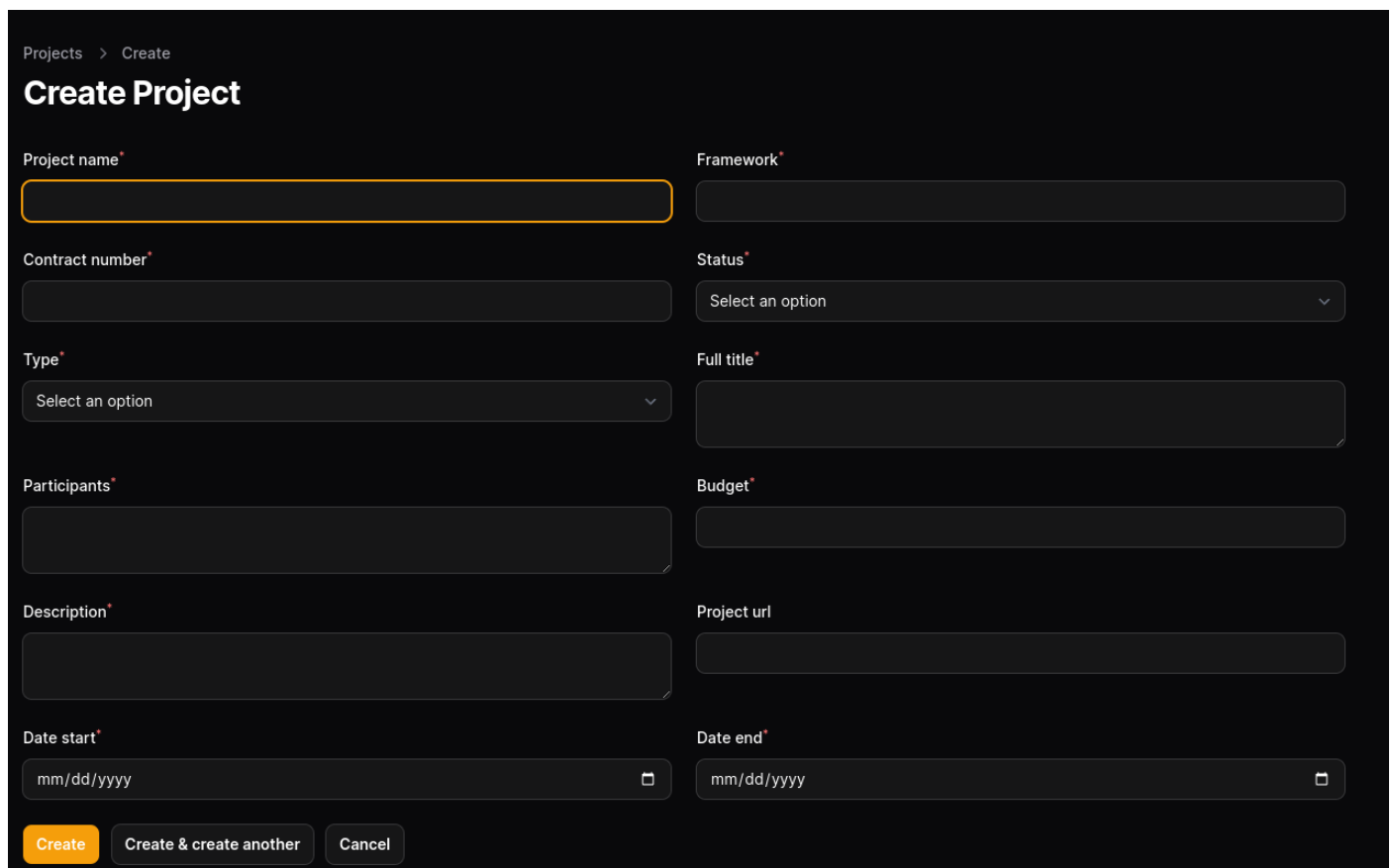
Εικόνα 19 Backend - Λίστα έργων

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

4.2.2 Έργα - προσθήκη καινούργιου

Από εδώ οι διαχειριστές μπορούν να ανεβάζουν καινούργια έργα συμπληρώνοντας όλα τα απαραίτητα πεδία όπως: όνομα του έργου, Framework, αριθμός συμβολαίου, κατάσταση του έργου (Σε εξέλιξη ή Ολοκληρωμένο), Τύπος (Εθνικό ή Ευρωπαϊκό), Πλήρες τίτλος, Συμμετάσχοντες, Προϋπολογισμός, περιγραφή του έργου, ημερομηνία έναρξης και λήξης του έργου.

Το πεδίο του σύνδεμου του έργου είναι μη υποχρεωτικό.



Projects > Create

Create Project

Project name*

Framework*

Contract number*

Status*

Type*

Full title*

Participants*

Budget*

Description*

Project url

Date start*

Date end*

Εικόνα 20 Backend - Προσθήκη έργου

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

4.2.3 Έργα - επεξεργασία

Όταν ο διαχειριστής θέλει να επεξεργαστεί ένα έργο, τα πεδία είναι ήδη προσυμπληρωμένα, και μπορεί απλά να επεξεργαστεί εκάστοτε πεδίο ξεχωριστά.

The screenshot shows a 'Projects > Edit' page with the title 'Edit Project'. A red 'Delete' button is in the top right. The form is organized into two columns:

- Project name***: Input field with 'AMBROSIA'.
- Contract number***: Input field with '601023'.
- Type***: Dropdown menu with 'EU Funding' selected.
- Participants***: Input field with 'N/A'.
- Description***: Text area with 'The objective of AMBROSIA is to promote the wider reuse of the digital cultural resources available through Europeana by the Creative Industries to boost creativity'.
- Date start***: Date picker with '01/01/2014'.
- Framework***: Input field with 'CIP-ICT-PSP-2013-7'.
- Status***: Dropdown menu with 'Completed' selected.
- Full title***: Input field with 'Ambrosia: Europeana Food and Drink'.
- Budget***: Input field with '151000'.
- Project url**: Input field with 'http://foodanddrinkeurope.eu/'.
- Date end***: Date picker with '12/31/2016'.

At the bottom, there are 'Save changes' and 'Cancel' buttons.

Εικόνα 21 Backend - Επεξεργασία έργου

4.2.4 Δημοσιεύσεις

Λίστα με όλες τις δημοσιεύσεις

Publications > List

Publications

New publication

<input type="checkbox"/>	Title	Authors	Publisher	P
<input type="checkbox"/>	A Deep Learning Approach for Human Action Recognition Using Skeletal Information	Phivos, Apostolos, Evaggelos	PubMed	B
<input type="checkbox"/>	Semantic Processing	Phivos	CRC Press	B
<input type="checkbox"/>	Detecting and Verifying Dissimilar Patterns in Unlabelled Data	Phivos	-	C
<input type="checkbox"/>	E-learning and intelligent content adaptation	Phivos, Paraskevi, Stefanos	Inderscience Enterprises Ltd.	Ji
<input type="checkbox"/>	Personalized Content Retrieval in Context Using Ontological Knowledge	Phivos, Miriam, David	IEEE	Ji
<input type="checkbox"/>	Offering Access to Personalized Interactive Video	Phivos, Giorgos, Manolis, Stefanos	-	B
<input type="checkbox"/>	Using k-Nearest Neighbor and Feature Selection as an Improvement to Hierarchical Clustering	Phivos, Manolis, Stefanos	Springer-Verlag Berlin Heidelberg	B
<input type="checkbox"/>	Detecting and Verifying Dissimilar Patterns in Unlabelled Data	Phivos, Manolis, Stefanos	-	B
<input type="checkbox"/>	Self-tuning Personalized Information Retrieval in an Ontology-Based Framework	David, Miriam, Phivos	Springer-Verlag Berlin Heidelberg	B
<input type="checkbox"/>	A Context-based Region Labeling Approach for Semantic Image Segmentation	Phivos, Thanos, Yannis	-	B

Showing 1 to 10 of 130 results

Per page 10

1 2 3 4 ... 12 13 >

Εικόνα 22 Backend - λίστα δημοσιεύσεων

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

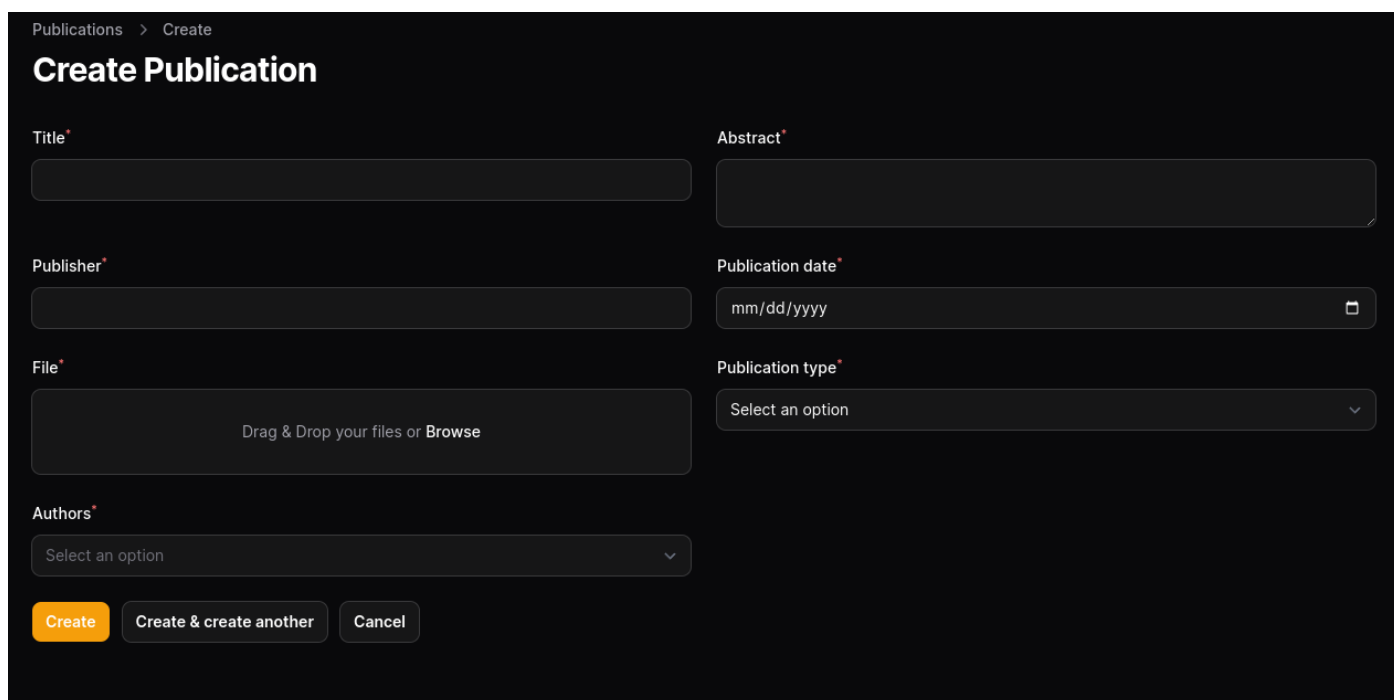
4.2.5 Δημοσιεύσεις Δημιουργία/Επεξεργασία

Σε αυτή την σελίδα οι διαχειριστές μπορούν να ανεβάζουν καινούργιες δημοσιεύσεις που θα φαίνονται μετά στο frontend, συμπληρώνοντας Τον τίτλο, την περίληψη, τον εκδότη, την ημερομηνία έκδοσης,

Το αρχείο (σε μορφή pdf), τον τύπο της δημοσίευσης, και μπορούν να επιλέξουν τους συγγραφείς από ένα multi-box.

Η επεξεργασία της εκάστοτε δημοσίευσης γίνεται με τον ίδιο τρόπο, απλά τα πεδία είναι προσυμπληρωμένα εκτός από το αρχείο.

Για να φτιάξει ο διαχειριστής μια καινούργια δημοσίευση, πρέπει να συμπληρώσει όλα τα πεδία.



The screenshot shows a 'Create Publication' form with the following fields and controls:

- Title***: A text input field.
- Abstract***: A larger text input field.
- Publisher***: A text input field.
- Publication date***: A date input field with a placeholder 'mm/dd/yyyy' and a calendar icon.
- File***: A file upload area with the text 'Drag & Drop your files or Browse'.
- Publication type***: A dropdown menu with the placeholder 'Select an option'.
- Authors***: A dropdown menu with the placeholder 'Select an option'.
- Buttons**: Three buttons at the bottom: 'Create' (highlighted in orange), 'Create & create another', and 'Cancel'.

Εικόνα 23 Backend - Δημιουργία δημοσίευσης

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

4.2.6 Μαθήματα

Σε αυτή την λίστα φαίνεται η λίστα με τα μαθήματα που έχουν ήδη ανέβει στην εφαρμογή.

<input type="checkbox"/>	Title en	Title gr	Type	
<input type="checkbox"/>	Algorithm Design & Analysis	Σχεδίαση και Ανάλυση Αλγορίθμων	Undergraduate	Edit Delete
<input type="checkbox"/>	Object Oriented Programming	Αντικειμενοστρεφής Προγραμματισμός	Undergraduate	Edit Delete
<input type="checkbox"/>	Data Structures	Δομές Δεδομένων	Undergraduate	Edit Delete
<input type="checkbox"/>	Databases I	Βάσεις Δεδομένων I	Undergraduate	Edit Delete
<input type="checkbox"/>	Software Development Methodologies	Μεθοδολογίες Ανάπτυξης Εφαρμογών	Undergraduate	Edit Delete
<input type="checkbox"/>	Artificial Intelligence and Machine Learning	Τεχνητή Νοημοσύνη και Μηχανική Μάθηση	Graduate	Edit Delete
<input type="checkbox"/>	Data Mining	Εξόρυξη Δεδομένων	Graduate	Edit Delete
<input type="checkbox"/>	Image Processing and Analysis	Επεξεργασία και Ανάλυση Εικόνας	Graduate	Edit Delete
<input type="checkbox"/>	Advanced Databases	Προηγμένες Βάσεις Δεδομένων	Graduate	Edit Delete
<input type="checkbox"/>	Deep Learning	Βαθιά Μάθηση	Graduate	Edit Delete

Showing 1 to 10 of 10 results

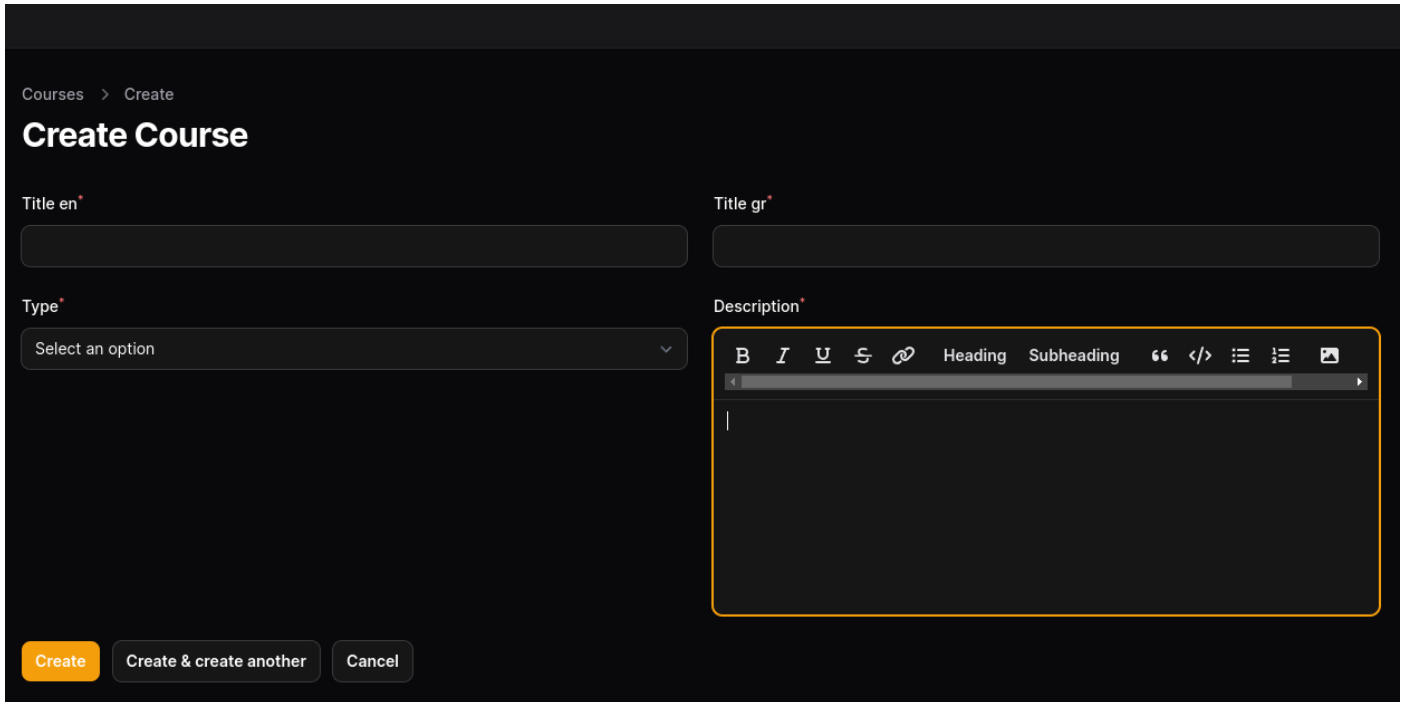
Per page 10

Εικόνα 24 Backend - Λίστα μαθημάτων

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

4.2.7 Μαθήματα - δημιουργία / επεξεργασία

Σε αυτή την σελίδα μπορεί ο διαχειριστής της εφαρμογής να ανεβάσει καινούργιο μάθημα συμπληρώνοντας υποχρεωτικά όλα τα πεδία: Τίτλο στα αγγλικά, τίτλο στα ελληνικά, τον τύπο του μαθήματος (Προπτυχιακό / Μεταπτυχιακό), και την περιγραφή που είναι ένα πεδίο με **Rich text editor** που μπορεί να δώσει επιθυμητή μορφοποίηση του κειμένου.



Courses > Create

Create Course

Title en*

Title gr*

Type*

Select an option

Description*

B I U ~~ABC~~ 🔗 Heading Subheading “ </> ☰ ☷ 🖼️

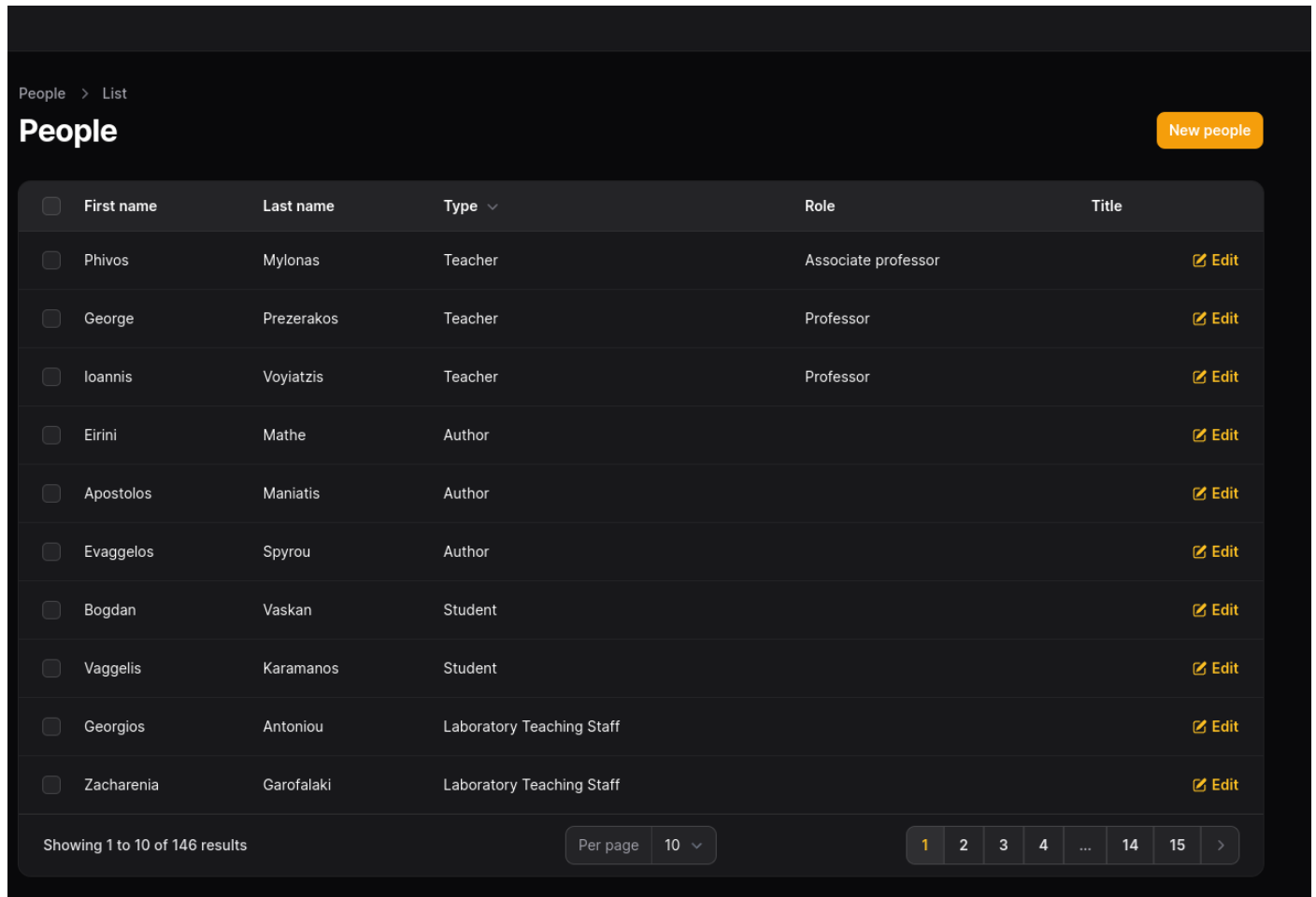
Create Create & create another Cancel

Εικόνα 25 Backend - Δημιουργία μαθήματος

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

4.2.8 Προσωπικό

Σε αυτή την σελίδα απεικονίζονται όλη η λίστα με τους ανθρώπους που απεικονίζεται στην ενότητα “Προσωπικό” στο frontend. Εκτός από τους συγγραφείς οι οποίοι δεν φαίνονται σε αυτή την σελίδα στο frontend καθώς είναι πάρα πολλοί και για λόγους αισθητικής, δεν φαίνονται.



People > List

People New people

<input type="checkbox"/>	First name	Last name	Type	Role	Title	<input type="checkbox"/>
<input type="checkbox"/>	Phivos	Mylonas	Teacher	Associate professor		<input type="checkbox"/>
<input type="checkbox"/>	George	Prezerakos	Teacher	Professor		<input type="checkbox"/>
<input type="checkbox"/>	Ioannis	Voyiatzis	Teacher	Professor		<input type="checkbox"/>
<input type="checkbox"/>	Eirini	Mathe	Author			<input type="checkbox"/>
<input type="checkbox"/>	Apostolos	Maniatis	Author			<input type="checkbox"/>
<input type="checkbox"/>	Evaggelos	Spyrou	Author			<input type="checkbox"/>
<input type="checkbox"/>	Bogdan	Vaskan	Student			<input type="checkbox"/>
<input type="checkbox"/>	Vaggelis	Karamanos	Student			<input type="checkbox"/>
<input type="checkbox"/>	Georgios	Antoniou	Laboratory Teaching Staff			<input type="checkbox"/>
<input type="checkbox"/>	Zacharenia	Garofalaki	Laboratory Teaching Staff			<input type="checkbox"/>

Showing 1 to 10 of 146 results

Per page 10

1 2 3 4 ... 14 15 >

Εικόνα 26 Backend - Προσωπικό

ΚΕΦΑΛΑΙΟ 5

Εισαγωγή

Σε αυτό το κεφάλαιο αναλύονται οι μελλοντικές επεκτάσεις και τα προβλήματα κατά την ανάπτυξη της εφαρμογής.

5.1 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Υλοποίηση Εγγραφής / Εισόδου των χρηστών στο frontend, επέκταση ρόλων στην εφαρμογή (αυτή την στιγμή υπάρχει μόνο ένας - ο διαχειριστής), μελλοντικά να μπορούν οι χρήστες να προτείνουν και να ανεβάζουν τις δημοσιεύσεις στην εφαρμογή και μετά από έγκριση του διαχειριστή να φαίνονται στο σύστημα.

Επιπλέον σε αυτό θα μπορούσε να δημιουργηθεί η σελίδα “Προφίλ του χρήστη”, έτσι ώστε να μπορεί να ενημερώσει τα στοιχεία του ή να διαγράψει τον λογαριασμό του.

Υλοποίηση λειτουργίας “Προσθήκη στην λίστα αγαπημένων” και της ίδιας της σελίδας έτσι ώστε ο χρήστης να μπορεί να βρει εύκολα τις δημοσιεύσεις που του αρέσουν και θέλει να τα διαβάσει ή να κατεβάσει στον υπολογιστή του ή κινητή συσκευή.

Βελτιώσεις στις μεταφράσεις του περιεχομένου τα οποία δεν έχουν μεταφραστεί όλα (κυρίως το περιεχόμενο της εφαρμογής).

Δημιουργία σελίδας “Person” όπου θα εμφανίζονται περισσότερες πληροφορίες για το προσωπικό.

Δημιουργίας ξεχωριστές σελίδας με λίστα όλων των συγγραφείς που έχουν έστω μια δημοσίευση.

Δημιουργία σελιδαρίθμηση(pagination), ειδικά για την σελίδα “Δημοσιεύσεις” για να γίνει πιο εύχρηστο, να γυρνάει δηλαδή το backend δεδομένα ανά σελίδα π.χ 20 δημοσιεύσεις ανά σελίδα και θα πρέπει να υλοποιηθεί αντίστοιχα ο μηχανισμός για την σωστή απεικόνιση και αλλαγή στα αιτήματα στο backend από το frontend.

5.2 Προβλήματα κατά την ανάπτυξη της εφαρμογής

Το μεγαλύτερο πρόβλημα κατά την ανάπτυξη της εφαρμογής για μένα ήταν η έλλειψη χρόνου, καθώς εργάζομαι με πλήρη απασχόληση ως Software Engineer εδώ και αρκετά χρόνια, ταυτόχρονα αυτό με βοήθησε καθώς έχω αποκτήσει πολλές γνώσεις, από όλα αυτά Το Laravel δεν έχω χρησιμοποιήσει στα πλαίσια της εργασίας μου αλλά είχα κάνει ένα σεμινάριο και μερικά εκπαιδευτικά έργα στο παρελθόν. Οπότε χρειαζόμουν κάποιο χρόνο για να θυμηθώ πράγματα ή να τα μάθω εκείνη την στιγμή,

Vue.js από την άλλη χρησιμοποιώ στην δουλειά μου αλλά με άλλη μεθοδολογία, οπότε πάλι ήθελε ψάξιμο.

Βιβλιογραφία

Lynch, Clifford (2005). "Where Do We Go From Here? The Next Decade for Digital Libraries". D-Lib Magazine. 11 (7/8). doi:10.1045/july2005-lynch. ISSN 1082-9873.

Hsinchun Chen, Andrea L. Houston (2008), "Digital Libraries: Social Issues and Technological Advances" ISBN 0-12-01 2 148-4

Jie Sun, Bao-Zhong Yuan (2012), "Development and Characteristic of Digital Library as a Library Branch"

Murtaza Ashiq, Farhat Jabeen, Khalid Mahmood (2022), "Transformation of libraries during Covid-19 pandemic: A systematic review"

Matt Stauffer, (2003). " Laravel: Up & Running: A Framework for Building Modern PHP Apps", 978-1098153267

Rj45 (2008), Simple Example of MVC (Model View Controller) Design Pattern for Abstraction

Hassan Djirdeh, Nate Murray, and Ari Lerner, (2018), " Fullstack Vue: The Complete Guide to Vue.js"

Leonard Richardson, Mike Amundsen, Sam Ruby (2013), "RESTful Web APIs". 978-1449358068

Hugh E. Williams Hugh E. Williams, Seyed M. M. Tahaghoghi, "Learning MySQL: Get a Handle on Your Data". 978-0596008642

Χρήσιμοι σύνδεσμοι

<https://laravel.com/>

<https://filamentphp.com/>

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

<https://vuejs.org/>

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

<https://www.i18next.com/>

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

[Σύνδεσμος για την ΔΕ \(Frontend\)](#)

[Σύνδεσμος για την ΔΕ \(Backend - Admin Panel\)](#)

[Github repository - Frontend](#)

[Github repository - Backend](#)

Παράρτημα Α (frontend)

Σε αυτό το παράρτημα παρουσιάζεται ο κώδικας του **fronted**.

Main.js

```
import i18n from './i18n'  
import 'bootstrap/dist/css/bootstrap.css'  
import "/node_modules/flag-icons/css/flag-icons.min.css";  
import { createApp } from 'vue'  
import App from './App.vue'  
import router from './router.js';
```

```
const app = createApp(App);  
app.use(i18n);  
app.use(router);  
app.mount('#app');
```

```
import "bootstrap/dist/js/bootstrap.js"
```

Router.js

```
import { createRouter, createWebHistory } from "vue-router";  
import AboutPage from './pages/AboutPage.vue';  
import ProjectsPage from "@pages/ProjectsPage.vue";  
import ProjectView from "@components/project/ProjectView.vue";  
import PublicationsPage from "@pages/PublicationsPage.vue";  
import ViewAbstract from "@components/publication/ViewAbstract.vue";  
import CoursesPage from "@pages/CoursesPage.vue";  
import PeoplePage from "@pages/PeoplePage.vue";  
import LinksPage from "@pages/LinksPage.vue";  
import NotFound from "@pages/NotFound.vue";
```

```
const router = createRouter({  
  history: createWebHistory(),  
  routes: [  
    { path: '/', redirect: '/about' },  
    { path: '/about', component: AboutPage },
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
{ path: '/projects', component: ProjectsPage },
{ name: 'project-view', path: '/projects/:id', component: ProjectView, props: true },
{ path: '/publications', component: PublicationsPage },
{ name: 'view-abstract', path: '/publication/:publicationId', component: ViewAbstract, props: true },
{ path: '/courses', component: CoursesPage },
{ path: '/people', component: PeoplePage },
{ path: '/links', component: LinksPage },
{ path: '/:notFound(.*)', component: NotFound } // catch all urls that doesn't exist in app
],
scrollBehavior(to, from, savedPosition) {
  if (savedPosition) {
    return savedPosition;
  }

  return { left: 0, top: 0 };
}
});
```

```
export default router;
```

App.vue

```
<template>
  <the-header></the-header>
  <main>
    <router-view></router-view>
  </main>
  <footer class="footer navbar navbar-expand-lg bg-body-tertiary">
    <the-footer></the-footer>
  </footer>
</template>

<script>
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
import TheHeader from './components/layouts/TheHeader.vue';
import TheFooter from "@components/layouts/TheFooter.vue";

export default {
  name: 'App',
  components: {
    TheHeader,
    TheFooter,
  },
}
</script>
```

```
<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
}
</style>
```

I18n.js

```
import { createI18n } from 'vue-i18n'
import us from './locales/us.json'
import gr from './locales/gr.json'

function loadLocaleMessages() {
  const locales = [ { us: us }, { gr: gr } ]
  const messages = {}
  locales.forEach(lang => {
    const key = Object.keys(lang)
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
    messages[key] = lang[key]
  })
  return messages;
}

export default createI18n({
  locale: 'us',
  fallbackLocale: 'us',
  messages: loadLocaleMessages()
})
```

pages/AboutPage.vue

```
<template>
  <div class="container">
    <div class="card mt-4 mb-4">
      <div class="card-body text-start">
        <h1>{{ $t('University of West Attica') }}</h1>
        <h2>{{ $t('Department of Informatics and Computer Engineering')}}</h2>

        <h3 class="mr-auto"><strong>{{ $t('Address') }}</strong></h3>
        <p class="mb-0">
          <strong>
            <a href="https://maps.app.goo.gl/TwLzsnJdhEmD8x518" target="_blank"><i class="fa-regular fa-map"
            style="color: red"></i> {{ $t('Ag. Spyridonos Str., Egaleo, Postal Code 12243, Athens') }}</a>
          </strong>
        </p>
        <a class="d-block" href="mailto:ice@uniwa.gr"><i class="fa-solid fa-envelope" style="color: #1a037e"></i>
ice@uniwa.gr</a>
        <a class="d-block" href="tel:2105385312"><i class="fa-solid fa-phone" style="color: #1a037e"></i> +30.210-
538-5312</a>
        <a class="d-block" href="tel:2105910975"><i class="fa-solid fa-fax" style="color: #1a037e"></i> +30.210-591-
0975</a>
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
<h3 class="mr-auto"><strong>{{ $t('Map') }}<i class="fa-solid fa-location-dot m-lg-2" style="color: red"></i></strong></h3>
```

```
<a href="https://maps.app.goo.gl/DjqWjd8sXgLmm8sd8" target="_blank"> <i class="fa-regular fa-map" style="color: red"></i> {{ $t('Campus on Google maps') }}</a>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</template>
```

```
<style scoped>
```

```
a {
```

```
  color: black;
```

```
  padding-left: 0;
```

```
}
```

```
a:hover {
```

```
  color: #1a037e;
```

```
}
```

```
</style>
```

```
<script setup lang="ts">
```

```
</script>
```

CoursesPage.vue

```
<template class="text-start">
```

```
<div class="container text-start mt-5 mb-5">
```

```
<div class="card mb-3">
```

```
<h2 class="card-title mt-2" style="margin-left: 20px">{{ $t("Undergraduate") }}</h2>
```

```
<div class="text-start d-flex flex-column mb-2 pt-0 mt-0 card-body">
```

```
<courses-list :courses="undergraduateCourses" identifier="undergraduate"></courses-list>
```

```
</div>
```

```
</div>
```

```
<div class="card">
```

```
<h2 class="card-title mt-2" style="margin-left: 20px">{{ $t("Graduate") }}</h2>
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
<div class="text-start d-flex flex-column mb-2 mt-0 pt-0 card-body">
  <courses-list :courses="graduateCourses" identifier="graduate"></courses-list>
</div>
</div>
</div>

</template>

<script>
import CoursesList from "@components/CoursesList.vue";
import axios from "axios";

export default {
  components: {
    CoursesList
  },
  data() {
    return {
      apiUrl: process.env.VUE_APP_API_BASE_URL,
      undergraduateCourses: [],
      graduateCourses: []
    }
  },
  methods: {
    fetchData() {
      axios.get(`${this.apiUrl}/courses`)
        .then(response => {
          this.undergraduateCourses = response.data['data'].filter(course => course.type === 'Undergraduate');
          this.graduateCourses = response.data['data'].filter(course => course.type === 'Graduate');
        })
        .catch(error => {
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
        console.error(error);
    });
}
},
mounted() {
    this.fetchData();
}
}
</script>
```

```
<style scoped>
```

```
</style>
```

LinksPage.vue

```
<template>
```

```
<div class="container">
  <div class="card mt-4 mb-4">
    <div class="card-body text-start">
      <h1>{{ $('Links') }}</h1>
      <div class="d-flex">
        <ul class="text">
          <li class="list-group-item m-0">
            <a href="https://services.uniwa.gr" target="_blank"><i class="fa-solid fa-link" style="color: #1a037e"></i>
            {{ $('Services') }}</a>
          </li>
          <li class="list-group-item m-0">
            <a href="https://eclass.uniwa.gr/" target="_blank"><i class="fa-solid fa-link" style="color: #1a037e"></i> E-
            class</a>
          </li>
          <li class="list-group-item m-0">
            <a href="https://www.uniwa.gr/foitites/" target="_blank"><i class="fa-solid fa-link" style="color:
            #1a037e"></i> {{ $('Services for students') }}</a>
```


Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
</li>
<li class="list-group-item m-0">
  <a href="https://eudoxus.gr/" target="_blank"><i class="fa-solid fa-link" style="color: #1a037e"></i>
  {{ $t('Eudoxus') }}</a>
</li>
<li class="list-group-item m-0">
  <a href="https://ice.uniwa.gr/announcements-all/" target="_blank"><i class="fa-solid fa-link" style="color:
  #1a037e"></i> {{ $t('Announcements') }}</a>
</li>
</ul>
</div>
</div>
</div>
</div>
</div>
</template>

<style scoped>
a {
  color: black;
  padding-left: 0;
}
a:hover {
  color: #1a037e;
}
ul > li {
  min-width: 250px;
}
ul {
  padding-left: 0;
}
</style>
<script setup lang="ts">
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
</script>
```

NotFound.vue

```
<template>
```

```
<h2>
```

```
Page not found! Maybe view our <router-link to="/about">{{ $t("About") }}</router-link>
```

```
</h2>
```

```
</template>
```

PeoplePage.vue

```
<template>
```

```
<div class="container mt-5">
```

```
<div class="row mb-5">
```

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

```
<h3 class="text-start">{{ $t('Teachers') }}</h3>
```

```
<people-list :people="teachers"></people-list>
```

```
</div>
```

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

```
<h3 class="text-start">{{ $t('Students') }}</h3>
```

```
<people-list :people="students"></people-list>
```

```
</div>
```

```
</div>
```

```
<div class="row mb-5">
```

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

```
<h3 class="text-start">{{ $t('Laboratory Teaching Staff') }}</h3>
```

```
<people-list :people="labTeachingStaff"></people-list>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</template>
```

```
<script>
```

```
import PeopleList from "@components/PeopleList.vue";
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
import axios from "axios";

export default {
  components: {
    PeopleList
  },
  data() {
    return {
      apiUrl: process.env.VUE_APP_API_BASE_URL,
      teachers: [],
      authors: [],
      students: [],
      labTeachingStaff: [],
    }
  },
  methods: {
    fetchData() {
      axios.get(`${this.apiUrl}/people`)
        .then(response => {
          this.teachers = response.data['data'].filter(person => person.type === 'Teacher');
          this.authors = response.data['data'].filter(person => person.type === 'Author');
          this.students = response.data['data'].filter(person => person.type === 'Student');
          this.labTeachingStaff = response.data['data'].filter(person => person.type === 'Laboratory Teaching Staff');
        })
        .catch(error => {
          console.error(error);
        });
    }
  },
  mounted() {
    this.fetchData();
  }
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
}  
}  
</script>  
<style scoped>  
  
</style>
```

ProjectsPage.vue

```
<template>  
  <div class="container">  
    <tabs-component  
      class="mt-5"  
      :tabs="['All', 'EU In Progress', 'EU Completed', 'Greek In Progress', 'Greek Completed']"  
      v-model:activeTab="activeTab"  
    ></tabs-component>  
  
    <div v-if="activeTab === 'All' || activeTab === 'EU In Progress'">  
      <h2 class="text-start">{{ $( 'European In Progress' ) }}</h2>  
      <projects-component  
        v-for="project in projects_eu_active"  
        :key="project.id"  
        :id="project.id"  
        :framework="project.framework"  
        :title="project.project_name"  
        :fullProjectName="project.full_title"  
        :participants="project.participants"  
        :budget="project.budget"  
        :duration="project.duration"  
        :year_start="project.year_start"  
        :year_end="project.year_end"  
        :url="project.project_url"  
        :contract_number="project.contract_number"
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
      :description="project.description"  
    ></projects-component>  
</div>
```

```
<div v-if="activeTab === 'All' || activeTab === 'EU Completed'">  
  <h2 class="text-start">{{ $t('European Completed')}}</h2>  
  <projects-component  
    v-for="project in projects_eu_completed"  
    :key="project.id"  
    :id="project.id"  
    :framework="project.framework"  
    :title="project.project_name"  
    :fullProjectName="project.full_title"  
    :participants="project.participants"  
    :budget="project.budget"  
    :duration="project.duration"  
    :year_start="project.year_start"  
    :year_end="project.year_end"  
    :url="project.project_url"  
    :contract_number="project.contract_number"  
    :description="project.description"  
  ></projects-component>  
</div>
```

```
<div v-if="activeTab === 'All' || activeTab === 'Greek In Progress'">  
  <h2 class="text-start">{{ $t('Greek In Progress') }}</h2>  
  <projects-component  
    v-for="project in projects_gr_active"  
    :key="project.id"  
    :id="project.id"  
    :framework="project.framework"
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
:title="project.project_name"
:fullProjectName="project.full_title"
:participants="project.participants"
:budget="project.budget"
:duration="project.duration"
:year_start="project.year_start"
:year_end="project.year_end"
:url="project.project_url"
:contract_number="project.contract_number"
:description="project.description"
></projects-component>
</div>

<div v-if="activeTab === 'All' || activeTab === 'Greek Completed'">
  <h2 class="text-start">{{ $('Greek Completed')}}</h2>
  <projects-component
    v-for="project in projects_gr_completed"
    :key="project.id"
    :id="project.id"
    :framework="project.framework"
    :title="project.project_name"
    :fullProjectName="project.full_title"
    :participants="project.participants"
    :budget="project.budget"
    :duration="project.duration"
    :year_start="project.year_start"
    :year_end="project.year_end"
    :url="project.project_url"
    :contract_number="project.contract_number"
    :description="project.description"
  ></projects-component>
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
</div>
</div>
</template>

<script>
import ProjectsComponent from "@/components/project/ProjectsComponent.vue";
import TabsComponent from "@/components/TabsComponent.vue";

import axios from "axios";

export default {
  components: {
    TabsComponent,
    ProjectsComponent
  },
  data() {
    return {
      apiUrl: process.env.VUE_APP_API_BASE_URL,
      projects_eu_active: [],
      projects_eu_completed: [],
      projects_gr_active: [],
      projects_gr_completed: [],
      activeTab: 'All'
    }
  },
  methods: {
    fetchData() {
      axios.get(`${this.apiUrl}/projects`)
        .then(response => {
          this.projects_eu_active = response.data['data'].filter(project => project.type === 'EU Funding' &&
project.status === 'In Progress');

          this.projects_eu_completed = response.data['data'].filter(project => project.type === 'EU Funding' &&
project.status === 'Completed');
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
this.projects_gr_active = response.data['data'].filter(project => project.type === 'Greek Funding' && project.status === 'In Progress');
```

```
this.projects_gr_completed = response.data['data'].filter(project => project.type === 'Greek Funding' && project.status === 'Completed');
```

```
  })
```

```
  .catch(error => {
```

```
    console.error(error);
```

```
  });
```

```
}
```

```
},
```

```
mounted() {
```

```
  this.fetchData();
```

```
}
```

```
}
```

```
</script>
```

```
<style scoped>
```

```
</style>
```

PublicationsPage.vue

```
<template>
```

```
<div class="container">
```

```
<publication-filters @search="applyFilters" class="mt-5"></publication-filters>
```

```
<publication-card
```

```
  :key="publication.id"
```

```
  v-for="publication in publications"
```

```
  :authors="publication.authors"
```

```
  :title="publication.title"
```

```
  :file_url="publication.file"
```

```
  :id="publication.id"
```

```
  :publisher="publication.publisher"
```

```
  :publication_date="publication.publication_date"
```

```
  :publication_type="publication.publication_type"
```


Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
></publication-card>
</div>
</template>

<script>
import PublicationFilters from "@components/publication/PublicationFilters.vue";
import PublicationCard from "@components/publication/PublicationCard.vue";
import axios from "axios";
export default {
  components: {
    PublicationFilters,
    PublicationCard
  },
  data() {
    return {
      apiUrl: process.env.VUE_APP_API_BASE_URL,
      publications: [],
      filters: {}
    }
  },
  methods: {
    fetchData() {
      axios.get(`${this.apiUrl}/publications`, { params: this.filters })
        .then(response => {
          this.publications = response.data['data'];
        })
        .catch(error => {
          console.error(error);
        });
    },
    applyFilters(filters) {

```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
    this.filters = filters;
    this.fetchData();
  }
},
mounted() {
  this.fetchData();
}
}
</script>
<style scoped>

</style>
```

TheFooter.vue

```
<template>
  <div class="container">
    <div class="col-md-12">
      <span class="d-flex justify-content-center">© 2024. All rights reserved</span>
    </div>
  </div>
</template>

<style scoped>

</style>
```

TheHeader.vue

```
<template>
  <header class="header-wrapper">
    <nav class="navbar navbar-expand-lg bg-body-tertiary">
      <div class="container">
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
<a
  class="navbar-brand pb-0"
  href="https://ice.uniwa.gr/"
  target="__blank"
>
  
</a>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav me-auto mb-2 mb-lg-0">
    <li class="nav-item">
      <router-link class="nav-link" to="/about">
        {{ $t("About") }}
      </router-link>
    </li>
    <li class="nav-item">
      <router-link class="nav-link" to="/projects">
        {{ $t("Projects") }}
      </router-link>
    </li>
    <li class="nav-item">
      <router-link class="nav-link" to="/publications">
        {{ $t("Publications") }}
      </router-link>
    </li>
    <li class="nav-item">
      <router-link class="nav-link" to="/courses">
        {{ $t("Courses") }}
      </router-link>
    </li>
    <li class="nav-item">
      <router-link class="nav-link" to="/people">
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
    {{ $t("People") }}
  </router-link>
</li>
<li class="nav-item">
  <router-link class="nav-link" to="/links">
    {{ $t("Links") }}
  </router-link>
</li>
</ul>
<div class="language-switcher">
  <div
    v-for="locale in $i18n.availableLocales"
    :key="`locale-${locale}`"
    @click="switchLanguage(locale)"
    class="language-option"
  >
    <span
      :class="['flag-icon', `flag-icon-${locale.toLowerCase()}`]"
    ></span>
    <!-- Apply flag CSS class -->
    {{ locale == "us" ? "EN" : locale.toUpperCase() }}
  </div>
  <select v-model="$i18n.locale" class="hidden">
    <option
      v-for="locale in $i18n.availableLocales"
      :key="`locale-${locale}`"
      :value="locale"
    >
      {{ locale }}
    </option>
  </select>
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
</div>
</div>
</div>
</nav>
</header>
</template>

<script>
export default {
  methods: {
    switchLanguage(locale) {
      // Logic to switch language using the $i18n plugin
      this.$i18n.locale = locale;
    },
  },
};
</script>

<style>
.language-switcher {
  display: flex;
  align-items: center;
}

.language-option {
  margin-right: 10px;
  cursor: pointer;
}

.flag-icon {
  width: 30px; /* Adjust as needed */
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
height: auto;

display: inline-block;

/* Add any other necessary styles */

}

/* Hide the select element */

.hidden {

display: none;

}

header {

width: 100%;

height: 5rem;

background-color: #11005c;

}

nav {

height: 100%;

}

.header-wrapper ul {

list-style: none;

margin: 0;

padding: 0;

height: 100%;

display: flex;

justify-content: center;

align-items: center;

}

.header-wrapper li {

margin: 0 2rem;
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
}  
  
a {  
  font: inherit;  
  background: transparent;  
  border: 1px solid transparent;  
  cursor: pointer;  
  color: white;  
  padding: 0.5rem 1.5rem;  
  display: inline-block;  
  text-decoration: none;  
}  
  
.nav-link:hover,  
.nav-link:active,  
.nav-link.router-link-active {  
  color: white;  
  border-color: #f1a80a;  
  background-color: #1a037e;  
}  
  
.nav-link:hover, .nav-link:focus {  
  color: #f1a80a;  
}  
</style>
```

ProjectsComponent.vue

```
<template>  
  <div class="card text-start mb-3">  
    <div class="card-body">  
      <div class="row">  
        <div class="col-md-2">
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
    {{ $t('Framework')}}
</div>
<div class="col-md-10">
    {{ framework }}
</div>
<div class="col-md-2">
    {{ $t('Project Name')}}
</div>
<div class="col-md-10">
    <router-link class="p-0" :to="viewProjectUrl">{{ title }}</router-link>
</div>
<div class="col-md-2">
    {{ $t('Full Project Name')}}
</div>
<div class="col-md-10">
    <router-link class="p-0" :to="viewProjectUrl">{{ fullProjectName }}</router-link>
</div>
<div class="col-md-2">
    {{ $t('Participants')}}
</div>
<div class="col-md-10">
    {{ participants }}
</div>
<div class="col-md-2">
    {{ $t('Budget')}}
</div>
<div class="col-md-10">
    {{ budget }}
</div>
<div class="col-md-2">
    {{ $t('Duration')}}
</div>
```


Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
</div>
<div class="col-md-10">
  {{ year_start }} - {{ year_end }} ({{ duration }} {{ $t('months') }})
</div>
</div>
</div>
</div>
</template>

<script>
export default {
  props: ['title', 'category_title', 'framework', 'fullProjectName', 'participants', 'budget', 'duration', 'year_start', 'year_end',
'id'],
  computed: {
    viewProjectUrl() {
      return {
        name: 'project-view',
        params: { id: this.id }
      }
    },
  }
}
</script>

<style scoped>
a {
  color: black;
}
</style>
```

ProjectView.vue

```
<template>
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
<div class="container mt-3 mb-3 pb-3 pt-3 text-start card pt-0">  
  <div class="row">  
    <div class="col-md-2">  
      Framework  
    </div>  
    <div class="col-md-10">  
      {{ framework }}  
    </div>  
    <div class="col-md-2">  
      Project Name  
    </div>  
    <div class="col-md-10">  
      {{ title }}  
    </div>  
    <div class="col-md-2">  
      Contract Number  
    </div>  
    <div class="col-md-10">  
      {{ contract_number }}  
    </div>  
    <div class="col-md-2">  
      Full Project Name  
    </div>  
    <div class="col-md-10">  
      {{ fullProjectName }}  
    </div>  
    <div class="col-md-2">  
      Participants  
    </div>  
    <div class="col-md-10">  
      {{ participants }}  
    </div>  
  </div>  
</div>
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
</div>
<div class="col-md-2">
  Budget
</div>
<div class="col-md-10">
  {{ budget }}
</div>
<div class="col-md-2">
  Duration
</div>
<div class="col-md-10">
  {{ year_start }} - {{ year_end }} ({{ duration }} months)
</div>
<div class="col-md-2">
  Description
</div>
<div class="col-md-10">
  {{ description }}
</div>
<div class="col-md-2">
  URL
</div>
<div class="col-md-10">
  <a href="project_url" target="_blank" class="p-0">{{ project_url }}</a>
</div>
</div>
</div>
</div>
</template>

<script>
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
import axios from "axios";

export default {
  props: ['id'],
  data() {
    return {
      apiUrl: process.env.VUE_APP_API_BASE_URL,
      framework: "",
      title: "",
      fullProjectName: "",
      participants: "",
      budget: 0,
      duration: 0,
      description: "",
      year_start: null,
      year_end: null,
      project_url: "",
      contract_number: "",
    }
  },
  methods: {
    fetchData() {
      axios.get(`${this.apiUrl}/projects/${this.id}`)
        .then(response => {
          this.framework = response.data['data']['framework'];
          this.title = response.data['data']['project_name'];
          this.fullProjectName = response.data['data']['full_title'];
          this.participants = response.data['data']['participants'];
          this.budget = response.data['data']['budget'];
          this.duration = response.data['data']['duration'];
          this.description = response.data['data']['description'];
        });
    }
  }
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
    this.year_start = response.data['data']['year_start'];
    this.year_end = response.data['data']['year_end'];
    this.project_url = response.data['data']['project_url'];
    this.contract_number = response.data['data']['contract_number'];

  })
  .catch(error => {
    console.error(error);
  });
}
},
mounted() {
  this.fetchData();
}
}
</script>
```

```
<style scoped>
```

```
a {
  color: black;
}
</style>
```

PublicationCard.vue

```
<template>
  <div class="card text-start mb-3">
    <div class="card-body">
      <p class="mb-0">
        <span>{{ authorNames }}</span>
      </p>
      <h4 class="card-title">{{ title }}</h4>
      <ul class="d-flex justify-content-start publication_list">
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
<li class="ms-0">{{ publisher }}</li>
<li class="ms-0">{{ publication_date }}</li>
<li class="ms-0">{{ publication_type }}</li>
</ul>
</div>
<div class="d-flex mb-3">
  <router-link class="btn btn-primary mx-2" :to="abstractLink">{{ $('Abstract') }}</router-link>
  <a :href="file_url" target="_blank" class="btn btn-secondary">PDF</a>
</div>
</div>
</template>

<script>
export default {
  inject: ['publications'],
  provide() {
    return {
      authors: this.authors,
      description: this.description,
    }
  },
  props: ['authors', 'title', 'id', 'file_url', 'publisher', 'publication_date', 'publication_type'],
  computed: {
    abstractLink() {

      console.log(this.id);

      return {
        name: 'view-abstract',
        params: { publicationId: this.id }
      }
    }
  }
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
    },  
    authorNames() {  
        return this.authors.map(author => author.full_name).join(', ');  
    }  
}  
}
```

```
</script>
```

```
<style scoped>
```

```
.publication_list {  
    list-style: none;  
    margin: 0;  
    padding: 0;  
    height: 100%;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
}
```

```
.publication_list li {  
    position: relative;  
    margin: 0 2rem;  
}
```

```
.publication_list li::after {  
    content: "•";  
    position: absolute;  
    right: -20px;  
    color: black;  
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
.publication_list li:last-child::after {  
  content: "";  
}  
</style>
```

PublicationFilters.vue

```
<template>  
  <form @submit.prevent="handleSearch">  
    <div class="row">  
      <div class="mb-3 text-start d-flex justify-content-start">  
        <div class="col-md-2 mx-2">  
          <label for="author" class="form-label">{{ $t('Author') }}</label>  
          <input type="text" class="form-control" id="author" v-model="filters.author">  
        </div>  
        <div class="col-md-2 mx-2">  
          <label for="title" class="form-label">{{ $t('Title') }}</label>  
          <input type="text" class="form-control" id="title" v-model="filters.title">  
        </div>  
        <div class="col-md-2 mx-2">  
          <label for="abstract" class="form-label">{{ $t('Abstract') }}</label>  
          <input type="text" class="form-control" id="abstract" v-model="filters.abstract">  
        </div>  
        <div class="col-md-2 mx-2">  
          <label for="publisher" class="form-label">{{ $t('Publisher') }}</label>  
          <input type="text" class="form-control" id="publisher" v-model="filters.publisher">  
        </div>  
      </div>  
      <div class="d-flex text-start justify-content-start">  
        <div class="col-md-2 mx-2">  
          <label for="from" class="form-label">{{ $t('From') }}</label>  
          <input type="date" class="form-control" id="from" v-model="filters.from">  
        </div>  
      </div>  
    </form>
```


Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
</div>
<div class="col-md-2 mx-2">
  <label for="to" class="form-label">{{ $t('To') }}</label>
  <input type="date" class="form-control" id="to" v-model="filters.to">
</div>
<div class="col-md-2 mx-2">
  <label for="text" class="form-label d-flex">{{ $t('Text') }} <sub>1</sub></label>
  <input type="text" class="form-control" id="text" v-model="filters.text">
</div>
<div class="col-md-2 mx-2">
  <label for="type" class="form-label">{{ $t('Type') }}</label>
  <select class="form-select" aria-label="Default select example" v-model="filters.type">
    <option selected>{{ $t('All') }}</option>
    <option :value="publicationType" v-for="publicationType in publicationTypes" :key="publicationType">
      {{ $t(publicationType) }}
    </option>
  </select>
</div>
</div>
<div class="d-flex mt-3 mx-2"><sub>1</sub>{{ $t('min length 2 chars') }}</div>
<div class="d-flex mt-2 mx-2">
  <button class="btn btn-primary mt-2 mb-5" type="submit">{{ $t('Search') }}</button>
</div>
</div>
</form>
</template>

<script>
export default {
  data() {
    return {
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
filters: {
  author: "",
  title: "",
  abstract: "",
  publisher: "",
  from: "",
  to: "",
  text: "",
  type: 'All',
},
publicationTypes: [
  'Journal',
  'Book',
  'Conference',
  'Diploma Thesis',
  'Phd Thesis'
]
},
methods: {
  handleSearch() {
    this.$emit('search', this.filters);
  }
}
}
</script>

<style scoped>
</style>
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

ViewAbstract.vue

```
<template>

  <div class="container mt-3 mb-3 pb-3 pt-3 text-start card pt-0">

    <p class="mt-2 mb-2">{{ authorNames }}</p>

    <h4>{{ description }}</h4>

    <h2>{{ title }}</h2>

    <h2 class="mt-2">ABSTRACT</h2>

    <p>{{ abstract }}</p>

    <ul class="d-flex justify-content-start publication_list">

      <li class="ms-0">{{ publisher }}</li>

      <li class="ms-0">{{ publication_date }}</li>

      <li class="ms-0">{{ publication_type }}</li>

      <li class="ms-0"><a :href="file_url" target="_blank" class="btn btn-secondary">PDF</a></li>

    </ul>

  </div>

</template>

<script>

import axios from "axios";

export default {
  props: ['publicationId'],
  data() {
    return {
      apiUrl: process.env.VUE_APP_API_BASE_URL,
      title: "",
      description: "",
      authors: "",
      abstract: [],
      publication_date: "",
    }
  }
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
    publisher: "",
    publication_type: "",
  }
},
computed: {
  authorNames() {
    return this.authors ? this.authors.map(author => author.full_name).join(', ') : null;
  }
},
methods: {
  fetchData() {
    axios.get(`${this.apiUrl}/publications/${this.publicationId}`)
      .then(response => {
        this.title = response.data['data']['title'];
        this.abstract = response.data['data']['abstract'];
        this.authors = [...response.data.data.authors];
        this.publication_date = response.data['data']['publication_date'];
        this.publisher = response.data['data']['publisher'];
        this.publication_type = response.data['data']['publication_type'];
        this.file_url = response.data['data']['file'];

      })
      .catch(error => {
        console.error(error);
      });
  }
},
mounted() {
  this.fetchData();
}
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
</script>
```

```
<style scoped>
```

```
.publication_list {  
  list-style: none;  
  margin: 0;  
  padding: 0;  
  height: 100%;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

```
.publication_list li {  
  position: relative;  
  margin: 0 2rem;  
}
```

```
.publication_list li::after {  
  content: "•";  
  position: absolute;  
  right: -20px;  
  color: black;  
}
```

```
.publication_list li:last-child::after {  
  content: "";  
}
```

```
</style>
```

CoursesList.vue

```
<template>
```

```
<div :id=""`accordion-${identifier}`" class="accordion">
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
<div v-for="(course, index) in courses" :key="course.id" class="accordion-item">
  <h2 class="accordion-header" :id="`heading-${identifier}-${index}`">
    <button
      class="accordion-button collapsed"
      type="button"
      data-bs-toggle="collapse"
      :data-bs-target="`#collapse-${identifier}-${index}`"
      aria-expanded="false"
      :aria-controls="`collapse-${identifier}-${index}`"
    >
      <i class="fa-solid fa-play" style="color: #1a037e; margin-right: 15px"></i>
      {{ course.title_en }} - {{ course.title_gr }}
    </button>
  </h2>
  <div
    :id="`collapse-${identifier}-${index}`"
    class="accordion-collapse collapse"
    :aria-labelledby="`heading-${identifier}-${index}`"
    :data-bs-parent="`#accordion-${identifier}`"
  >
    <div class="accordion-body">
      <div v-html="course.description"></div>
    </div>
  </div>
</div>
</template>

<script>
export default {
  props: ['courses', 'identifier'],
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
}  
</script>  
  
<style scoped>  
.course-link {  
  color: black;  
}  
.course-link:hover,  
.course-link:active {  
  color: #1a037e;  
}  
</style>
```

PeopleList.vue

```
<template>  
  <ul class=" m-0 p-0 d-flex flex-column align-items-start flex-wrap course-list justify-content-start">  
    <li class="m-1 border-1 border-bottom p-0 text-start" aria-current="true" :key="person.id" v-for="person in  
people">  
      <a :href="person.link" class="course-link p-0 pt-2 pb-2" target="_blank">  
        <i class="fa-solid fa-circle" style="color: #1a037e"></i> {{ person.first_name + ' ' + person.last_name }}  
  
        <span v-if="person.title">({{ person.title }})</span>  
      </a>  
    </li>  
  </ul>  
</template>  
  
<script>  
export default {  
  props: ['people'],  
}  
</script>
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
<style scoped>
.course-list .course-link {
  color: black;
}
.course-list .course-link:hover,
.course-list .course-link:active {
  color: #1a037e;
}
.course-list {
  list-style: none;
  margin: 0;
  padding: 0;
  height: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
}
.course-list li {
  margin: 0 2rem;
}
</style>
```

TabsComponent.vue

```
<!-- TabsComponent.vue -->
<template>
  <div>
    <ul class="nav nav-tabs mb-5">
      <li v-for="tab in tabs" :key="tab" class="nav-item">
        <a
          :class="['nav-link', { active: activeTab === tab }]"
```


Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
      @click="selectTab(tab)"
      href="#"
    >
      {{ $t(tab) }}
    </a>
  </li>
</ul>
</div>
</template>

<script>
export default {
  props: {
    tabs: {
      type: Array,
      default: () => ['All', 'EU In Progress', 'EU Completed', 'Greek In Progress', 'Greek Completed']
    },
    activeTab: {
      type: String,
      default: 'All'
    }
  },
  methods: {
    selectTab(tab) {
      this.$emit('update:activeTab', tab);
    }
  }
}
</script>

<style scoped>
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
.nav-tabs .nav-link {  
  cursor: pointer;  
}  
.nav-tabs .nav-link.active {  
  color: white;  
  border-color: #f1a80a;  
  background-color: #1a037e;  
}  
</style>
```

Src/locales/gr.json

```
{  
  "About": "Σχετικά με μας",  
  "Projects": "Έργα",  
  "Publications": "Δημοσιεύσεις",  
  "Courses": "Μαθήματα",  
  "People": "Προσωπικό",  
  "Links": "Σύνδεσμοι",  
  "University of West Attica": "Πανεπιστήμιο Δυτικής Αττικής",  
  "Department of Informatics and Computer Engineering": "Τμήμα Μηχανικών Πληροφορικής και Υπολογιστών",  
  "Address": "Διεύθυνση",  
  "Ag. Spyridonos Str., Egaleo, Postal Code 12243, Athens": " Οδός Αγ. Σπυρίδωνος, Αιγάλεω, 122 43, Αθήνα",  
  "Map": "Χάρτης",  
  "Campus on Google maps": "Πανεπιστημιούπολη στους χάρτες Google",  
  "European In Progress": "Ευρωπαϊκά σε εξέλιξη",  
  "EU In Progress": "Ευρωπαϊκά σε εξέλιξη",  
  "European Completed": "Ευρωπαϊκά ολοκληρωμένα",  
  "EU Completed": "Ευρωπαϊκά ολοκληρωμένα",  
  "Greek In Progress": "Εθνικά σε εξέλιξη",  
  "Greek Completed": "Εθνικά ολοκληρωμένα",  
  "All": "Όλα τα έργα",
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

"Framework": "Πλαίσιο",
"Project Name": "Όνομα Έργου",
"Full Project Name": "Πλήρες Όνομα Έργου",
"Participants": "Συμμετέχοντες",
"Budget": "Προϋπολογισμός",
"Duration": "Διάρκεια",
"months": "μήνες",
"Author": "Συγγραφέας",
"Title": "Τίτλος",
"Abstract": "Περίληψη",
"Publisher": "Εκδότης",
"From": "Από",
"To": "Έως",
"Text": "Κείμενο",
"Type": "Τύπος",
"All": "Όλα",
"min length 2 chars": "ελάχιστο μήκος 2 χαρακτήρες",
"Search": "Αναζήτηση",
"Journal": "Περιοδικό",
"Book": "Βιβλίο",
"Conference": "Συνέδριο",
"Diploma Thesis": "Διπλωματική Εργασία",
"Phd Thesis": "Διδακτορική Διατριβή",
"Undergraduate": "Προπτυχιακού",
"Graduate": "Μεταπτυχιακού",
"Teachers": "Διδακτικό Ερευνητικό Προσωπικό (ΔΕΠ)",
"Authors": "Συγγραφείς",
"Laboratory Teaching Staff": "Εργαστηριακό Διδακτικό Προσωπικό (ΕΔΙΠ)",
"Students": "Φοιτητές",
"Services": "Φοιτητολόγιο",
"Services for students" : "Υπηρεσίες για τους φοιτητές",

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
"Eudoxus" : "Εύδοξος",  
"Announcements": "Ανακοινώσεις"  
}
```

Παράρτημα Β (backend)

Σε αυτό το παράρτημα παρουσιάζεται ο κώδικας του **backend**.

Models

PublicationSearch.php

```
<?php  
  
namespace App\Models\search;  
  
use App\Models\Publication;  
use Illuminate\Http\Request;  
  
class PublicationSearch  
{  
    public static function apply(Request $request)  
    {  
        $query = Publication::with('authors');  
  
        if ($request->filled('author')) {  
            $query->whereHas('authors', function($q) use ($request) {  
                $q->where('first_name', 'like', '%' . $request->input('author') . '%')  
                    ->orWhere('last_name', 'like', '%' . $request->input('author') . '%');  
            });  
        }  
        if ($request->filled('title')) {  
            $query->where('title', 'like', '%' . $request->input('title') . '%');  
        }  
    }  
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
if ($request->filled('abstract')) {
    $query->where('abstract', 'like', '%' . $request->input('abstract') . '%');
}

if ($request->filled('publisher')) {
    $query->where('publisher', 'like', '%' . $request->input('publisher') . '%');
}

if ($request->filled('from')) {
    $query->where('publication_date', '>=', $request->input('from'));
}

if ($request->filled('to')) {
    $query->where('publication_date', '<=', $request->input('to'));
}

if ($request->filled('text')) {
    $query->whereHas('file', function ($q) use ($request) {
        $q->where('text_content', 'like', '%' . $request->input('text') . '%');
    });
}

if ($request->filled('type') && $request->input('type') !== 'All') {
    $query->where('publication_type', Publication::getPublicationTypeIdByType($request->input('type')));
}

return $query->get();
}
}
```

Course.php

```
<?php
```

```
namespace App\Models;
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
```

```
class Course extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table = 'course';
```

```
    protected $fillable = [
```

```
        'title_en',
```

```
        'title_gr',
```

```
        'type',
```

```
        'description'
```

```
    ];
```

```
    const UNDERGRADUATE = 1;
```

```
    const GRADUATE = 2;
```

```
    public $timestamps = true;
```

```
    /**
```

```
     * @return string[]
```

```
     */
```

```
    public static function types(): array
```

```
{
```

```
    return [
```

```
        self::UNDERGRADUATE => 'Undergraduate',
```

```
        self::GRADUATE => 'Graduate',
```

```
    ];
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
}
```

```
}
```

File.php

```
<?php
```

```
namespace App\Models;
```

```
use Illuminate\Database\Eloquent\Factories\HasFactory;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
use Illuminate\Database\Eloquent\Relations\HasMany;
```

```
class File extends Model
```

```
{
```

```
    use HasFactory;
```

```
    protected $table = 'file';
```

```
    protected $fillable = [
```

```
        'name',
```

```
        'file_path',
```

```
        'size',
```

```
        'type',
```

```
        'text_content',
```

```
    ];
```

```
    public $timestamps = true;
```

```
/**
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
* @return HasMany
*/
public function publications(): HasMany
{
    return $this->hasMany(Publication::class);
}
}
```

People.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsToMany;

class People extends Model
{
    use HasFactory;

    protected $table = 'people';

    protected $fillable = [
        'first_name',
        'last_name',
        'role',
        'title',
        'type',
    ];
}
```


Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
public $timestamps = true;

const TYPE_TEACHER = 1;
const TYPE_LABORATORY_TEACHING_STAFF = 2;
const TYPE_AUTHOR = 3;
const TYPE_STUDENT = 4;

/**
 * @return BelongsToMany
 */
public function publications(): \Illuminate\Database\Eloquent\Relations\BelongsToMany
{
    return $this->belongsToMany(Publication::class, 'author_publication', 'person_id', 'publication_id');
}

/**
 * @return array[]
 */
public static function getTypes(): array
{
    return [
        self::TYPE_AUTHOR => 'Author',
        self::TYPE_STUDENT => 'Student',
        self::TYPE_TEACHER => 'Teacher',
        self::TYPE_LABORATORY_TEACHING_STAFF => 'Laboratory Teaching Staff'
    ];
}
}
```

Project.php

```
<?php
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
namespace App\Models;

use Carbon\Carbon;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Project extends Model
{
    use HasFactory;

    public $timestamps = true;

    const TYPE_EU = 1;
    const TYPE_GR = 2;

    const STATUS_IN_PROGRESS = 1;
    const STATUS_COMPLETED = 2;

    protected $fillable = [
        'project_name',
        'framework',
        'contract_number',
        'status',
        'type',
        'full_title',
        'participants',
        'budget',
        'description',
        'project_url',
    ]
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
'date_start',
'date_end',
];

/**
 * @return string[]
 */
public static function getTypes(): array
{
    return [
        self::TYPE_EU => 'EU Funding',
        self::TYPE_GR => 'Greek Funding',
    ];
}

/**
 * @return string[]
 */
public static function getStatuses(): array
{
    return [
        self::STATUS_IN_PROGRESS => 'In Progress',
        self::STATUS_COMPLETED => 'Completed',
    ];
}

/**
 * @param string $dateStart
 * @param string $dateEnd
 * @return int
 */
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
public static function calculateMonthsDifference(string $dateStart, string $dateEnd): int
{
    $start = Carbon::createFromFormat('Y-m-d', $dateStart);
    $end = Carbon::createFromFormat('Y-m-d', $dateEnd);
    $end->addDay();
    return $start->diffInMonths($end);
}

/**
 * @param string $date
 * @return int
 */
public static function getYearFromDateString(string $date): int
{
    $carbonDate = Carbon::createFromFormat('Y-m-d', $date);
    return $carbonDate->year;
}
}
```

Publication.php

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Database\Eloquent\Relations\BelongsTo;
use Illuminate\Database\Eloquent\Relations\BelongsToMany;

class Publication extends Model
{
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
use HasFactory;
```

```
protected $table = 'publication';
```

```
protected $fillable = [
```

```
    'title',
```

```
    'author_id',
```

```
    'abstract',
```

```
    'publisher',
```

```
    'publication_date',
```

```
    'publication_type',
```

```
    'file_id',
```

```
    'created_at',
```

```
    'updated_at',
```

```
];
```

```
const TYPE_JOURNAL = 1;
```

```
const TYPE_BOOK = 2;
```

```
const TYPE_CONFERENCE = 3;
```

```
const TYPE_DIPLOMA = 4;
```

```
const TYPE_PHD = 5;
```

```
public $timestamps = true;
```

```
/**
```

```
 * @return BelongsToMany
```

```
*/
```

```
public function authors(): \Illuminate\Database\Eloquent\Relations\BelongsToMany
```

```
{
```

```
    return $this->belongsToMany(People::class, 'author_publication', 'publication_id', 'person_id');
```

```
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
/**
 * @return BelongsTo
 */
public function file(): \Illuminate\Database\Eloquent\Relations\BelongsTo
{
    return $this->belongsTo(File::class, 'file_id', 'id');
}

/**
 * @return string[]
 */
public static function getPublicationTypes(): array
{
    return [
        self::TYPE_JOURNAL => 'Journal',
        self::TYPE_BOOK => 'Book',
        self::TYPE_CONFERENCE => 'Conference',
        self::TYPE_DIPLOMA => 'Diploma Thesis',
        self::TYPE_PHD => 'Phd Thesis',
    ];
}

public static function getPublicationTypeIdByType($type): ?int
{
    $types = array_flip(self::getPublicationTypes());
    return $types[$type] ?? null;
}
}
```

Controllers/Api

CoursesController.php

```
<?php
```

```
namespace App\Http\Controllers\Api;
```

```
use App\Http\Controllers\Controller;
```

```
use App\Http\Resources\CourseResource;
```

```
use App\Models\Course;
```

```
use Illuminate\Http\Request;
```

```
class CoursesController extends Controller
```

```
{
```

```
/**
```

```
 * Display a listing of the resource.
```

```
*/
```

```
public function index()
```

```
{
```

```
    return CourseResource::collection(Course::all());
```

```
}
```

```
/**
```

```
 * Store a newly created resource in storage.
```

```
*/
```

```
public function store(Request $request)
```

```
{
```

```
    //
```

```
}
```

```
/**
```

```
 * Display the specified resource.
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
*/  
public function show(string $id)  
{  
    //  
}  
  
/**  
 * Update the specified resource in storage.  
 */  
public function update(Request $request, string $id)  
{  
    //  
}  
  
/**  
 * Remove the specified resource from storage.  
 */  
public function destroy(string $id)  
{  
    //  
}  
}
```

PeopleController.php

```
<?php  
  
namespace App\Http\Controllers\Api;  
  
use App\Http\Controllers\Controller;  
use App\Http\Resources\PeopleResource;  
use App\Models\People;
```


Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
use Illuminate\Http\Request;
```

```
class PeopleController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        return PeopleResource::collection(People::all());
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store(Request $request)
    {
        //
    }

    /**
     * Display the specified resource.
     */
    public function show(string $id)
    {
        //
    }

    /**
     * Update the specified resource in storage.
     */
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
public function update(Request $request, string $id)
```

```
{  
    //  
}
```

```
/**
```

```
 * Remove the specified resource from storage.
```

```
*/
```

```
public function destroy(string $id)
```

```
{  
    //  
}
```

```
}
```

ProjectsController.php

```
<?php
```

```
namespace App\Http\Controllers\Api;
```

```
use App\Http\Controllers\Controller;
```

```
use App\Http\Resources\ProjectResource;
```

```
use App\Models\Project;
```

```
use Illuminate\Http\Request;
```

```
class ProjectsController extends Controller
```

```
{  
    /**  
     * Display a listing of the resource.  
     */  
    public function index()  
    {
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
    return ProjectResource::collection(Project::all());  
}
```

```
/**
```

```
* Store a newly created resource in storage.
```

```
*/
```

```
public function store(Request $request)
```

```
{
```

```
    //
```

```
}
```

```
/**
```

```
* Display the specified resource.
```

```
*/
```

```
public function show(Project $project)
```

```
{
```

```
    return new ProjectResource($project);
```

```
}
```

```
/**
```

```
* Update the specified resource in storage.
```

```
*/
```

```
public function update(Request $request, string $id)
```

```
{
```

```
    //
```

```
}
```

```
/**
```

```
* Remove the specified resource from storage.
```

```
*/
```

```
public function destroy(string $id)
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
{  
    //  
}  
}
```

PublicationsController.php

```
<?php
```

```
namespace App\Http\Controllers\Api;
```

```
use App\Http\Controllers\Controller;
```

```
use App\Http\Resources\PublicationResource;
```

```
use App\Models\Publication;
```

```
use App\Models\search\PublicationSearch;
```

```
use Illuminate\Http\Request;
```

```
class PublicationsController extends Controller
```

```
{  
    /**  
     * Display a listing of the resource.  
     */  
    public function index(Request $request)  
    {  
        $publications = PublicationSearch::apply($request);  
        return PublicationResource::collection($publications);  
    }  
  
    /**  
     * Store a newly created resource in storage.  
     */  
    public function store(Request $request)
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
{  
    //  
}  
  
/**  
 * Display the specified resource.  
 */  
public function show(Publication $publication)  
{  
    $publication->load('authors');  
    return new PublicationResource($publication);  
}  
  
/**  
 * Update the specified resource in storage.  
 */  
public function update(Request $request, string $id)  
{  
    //  
}  
  
/**  
 * Remove the specified resource from storage.  
 */  
public function destroy(string $id)  
{  
    //  
}  
}
```

Resources

CourseResource.php

```
<?php

namespace App\Http\Resources;

use App\Models\Course;
use Illuminate\Http\Request;
use Illuminate\Http\Resources\Json\JsonResource;

class CourseResource extends JsonResource
{
    /**
     * Transform the resource into an array.
     *
     * @return array<string, mixed>
     */
    public function toArray(Request $request): array
    {
        return [
            'id' => $this->id,
            'title_en' => $this->title_en,
            'title_gr' => $this->title_gr,
            'type_id' => $this->type,
            'type' => Course::types()[$this->type],
            'description' => $this->description,
        ];
    }
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

PeopleResource.php

```
<?php
```

```
namespace App\Http\Resources;
```

```
use App\Models\People;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Http\Resources\Json\JsonResource;
```

```
class PeopleResource extends JsonResource
```

```
{
```

```
/**
```

```
 * Transform the resource into an array.
```

```
 *
```

```
 * @return array<string, mixed>
```

```
 */
```

```
public function toArray(Request $request): array
```

```
{
```

```
    return [
```

```
        'id' => $this->id,
```

```
        'first_name' => $this->first_name,
```

```
        'last_name' => $this->last_name,
```

```
        'full_name' => $this->first_name . ' ' . $this->last_name,
```

```
        'role' => $this->role,
```

```
        'title' => $this->title,
```

```
        'type_id' => $this->type,
```

```
        'type' => People::getTypes()[$this->type]
```

```
    ];
```

```
}
```

```
}
```

ProjectResource.php

```
<?php
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
namespace App\Http\Resources;

use App\Models\Project;
use Illuminate\Http\Request;
use Illuminate\Http\Resources\Json\JsonResource;
use Carbon\Carbon;
use NumberFormatter;
use OpenSpout\Writer\XLSX\Helper\DateHelper;

class ProjectResource extends JsonResource
{
    /**
     * Transform the resource into an array.
     *
     * @return array<string, mixed>
     */
    public function toArray(Request $request): array
    {
        $formatter = new NumberFormatter('el_GR', NumberFormatter::CURRENCY);

        return [
            'id' => $this->id,
            'project_name' => $this->project_name,
            'framework' => $this->framework,
            'contract_number' => $this->contract_number,
            'status' => !empty($this->status) ? Project::getStatuses()[$this->status] : null,
            'type' => !empty($this->type) ? Project::getTypes()[$this->type] : null,
            'full_title' => $this->full_title,
            'participants' => $this->participants,
            'budget' => $formatter->formatCurrency($this->budget, 'EUR'),
```


Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
'description' => $this->description,  
'project_url' => $this->project_url,  
'duration' => Project::calculateMonthsDifference($this->date_start, $this->date_end),  
'year_start' => Project::getYearFromDateString($this->date_start),  
'year_end' => Project::getYearFromDateString($this->date_end),  
];  
}  
}
```

PublicationResource.php

```
<?php
```

```
namespace App\Http\Resources;  
  
use App\Models\Publication;  
use Illuminate\Http\Request;  
use Illuminate\Http\Resources\Json\JsonResource;  
use Illuminate\Support\Facades\Storage;  
  
class PublicationResource extends JsonResource  
{  
    /**  
     * Transform the resource into an array.  
     *  
     * @return array<string, mixed>  
     */  
    public function toArray(Request $request): array  
    {  
        return [  
            'id' => $this->id,  
            'title' => $this->title,
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
'abstract' => $this->abstract,
'publisher' => $this->publisher,
'publication_date' => date("m.d.Y", strtotime($this->publication_date)),
'publication_type_id' => $this->publication_type,
'publication_type' => !empty($this->publication_type) ? Publication::getPublicationTypes()[$this->publication_type] : null,
'file' => $this->file ? url(Storage::url($this->file->file_path)) : null,
'authors' => PeopleResource::collection($this->whenLoaded('authors')),
];
}
}
```

Filament Resources

App/Filament/Resources/CourseResource.php

```
<?php
```

```
namespace App\Filament\Resources;

use App\Filament\Resources\CourseResource\Pages;
use App\Filament\Resources\CourseResource\RelationManagers;
use App\Models\Course;
use Filament\Forms;
use Filament\Forms\Components\RichEditor;
use Filament\Forms\Components>Select;
use Filament\Forms\Form;
use Filament\Resources\Resource;
use Filament\Tables;
use Filament\Tables\Table;
use Illuminate\Database\Eloquent\Builder;
use Illuminate\Database\Eloquent\SoftDeletingScope;

class CourseResource extends Resource
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
{
    protected static ?string $model = Course::class;

    protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';

    public static function form(Form $form): Form
    {
        return $form
            ->schema([
                Forms\Components\TextInput::make('title_en')->required(),
                Forms\Components\TextInput::make('title_gr')->required(),
                Select::make('type')
                    ->required()
                    ->options(Course::types()),
                RichEditor::make('description')->required(),
            ]);
    }

    public static function table(Table $table): Table
    {
        return $table
            ->columns([
                Tables\Columns\TextColumn::make('title_en'),
                Tables\Columns\TextColumn::make('title_gr'),
                Tables\Columns\TextColumn::make('type')
                    ->label('Type')
                    ->formatStateUsing(function ($state) {
                        $types = Course::types();
                        return $types[$state] ?? 'Unknown';
                    }),
            ])
    }
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
->filters([
    //
])
->actions([
    Tables\Actions\EditAction::make(),
    Tables\Actions\DeleteAction::make(),
])
->bulkActions([
    Tables\Actions\BulkActionGroup::make([
        Tables\Actions\DeleteBulkAction::make(),
    ]),
]);
}
```

```
public static function getRelations(): array
```

```
{
    return [
        //
    ];
}
```

```
public static function getPages(): array
```

```
{
    return [
        'index' => Pages\ListCourses::route('/'),
        'create' => Pages\CreateCourse::route('/create'),
        'edit' => Pages\EditCourse::route('/{record}/edit'),
    ];
}
}
```

App/Filament/Resources/PeopleResource.php

```
<?php
```

```
namespace App\Filament\Resources;
```

```
use App\Filament\Resources\PeopleResource\Pages;
```

```
use App\Filament\Resources\PeopleResource\RelationManagers;
```

```
use App\Models\People;
```

```
use App\Models\Project;
```

```
use Filament\Forms;
```

```
use Filament\Forms\Components\Select;
```

```
use Filament\Forms\Form;
```

```
use Filament\Resources\Resource;
```

```
use Filament\Tables;
```

```
use Filament\Tables\Columns\TextColumn;
```

```
use Filament\Tables\Table;
```

```
use Illuminate\Database\Eloquent\Builder;
```

```
use Illuminate\Database\Eloquent\SoftDeletingScope;
```

```
class PeopleResource extends Resource
```

```
{
```

```
    protected static ?string $model = People::class;
```

```
    protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';
```

```
    public static function form(Form $form): Form
```

```
    {
```

```
        return $form
```

```
            ->schema([
```

```
                Forms\Components\TextInput::make('first_name')->required(),
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
Forms\Components\TextInput::make('last_name')->required(),
Forms\Components\TextInput::make('role'),
Forms\Components\TextInput::make('title'),
Select::make('type')
    ->options(People::getTypes())
    ->required(),
]);
}

public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('first_name'),
            Tables\Columns\TextColumn::make('last_name'),
            TextColumn::make('type')
                ->label('Type')
                ->formatStateUsing(fn($state) => People::getTypes()[$state] ?? '-')
                ->sortable(),
            Tables\Columns\TextColumn::make('role'),
            Tables\Columns\TextColumn::make('title'),
        ])
        ->filters([
            //
        ])
        ->actions([
            Tables\Actions\EditAction::make(),
        ])
        ->bulkActions([
            Tables\Actions\BulkActionGroup::make([
                Tables\Actions\DeleteBulkAction::make(),
            ])
        ])
    ;
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
    ]),  
    ]);  
}
```

```
public static function getRelations(): array  
{  
    return [  
        //  
    ];  
}
```

```
public static function getPages(): array  
{  
    return [  
        'index' => Pages\ListPeople::route('/'),  
        'create' => Pages\CreatePeople::route('/create'),  
        'edit' => Pages\EditPeople::route('/{record}/edit'),  
    ];  
}  
}
```

App/Filament/Resources/ProjectResource.php

```
<?php
```

```
namespace App\Filament\Resources;
```

```
use App\Filament\Resources\ProjectResource\Pages;
```

```
use App\Models\Project;
```

```
use Filament\Forms;
```

```
use Filament\Forms\Form;
```

```
use Filament\Resources\Resource;
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
use Filament\Tables;
use Filament\Tables\Table;
use Filament\Forms\Components\TextInput;
use Filament\Forms\Components>Select;
use Filament\Forms\Components\Textarea;
use Filament\Forms\Components\DatePicker;
use Filament\Tables\Columns\TextColumn;

class ProjectResource extends Resource
{
    protected static ?string $model = Project::class;

    protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';

    public static function form(Form $form): Form
    {
        return $form
            ->schema([
                TextInput::make('project_name')
                    ->required()
                    ->maxLength(255),
                TextInput::make('framework')
                    ->required()
                    ->maxLength(255),
                TextInput::make('contract_number')
                    ->required()
                    ->maxLength(255),
                Select::make('status')
                    ->options(Project::getStatuses())
                    ->required(),
                Select::make('type')
```


Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
->options(Project::getTypes())
->required(),
Textarea::make('full_title')
->required(),
Textarea::make('participants')
->required(),
TextInput::make('budget')
->required()
->numeric(),
Textarea::make('description')
->required(),
TextInput::make('project_url')
->url()
->required(),
DatePicker::make('date_start')
->required(),
DatePicker::make('date_end')
->required(),
]);
}

public static function table(Table $table): Table
{
    return $table
        ->columns([
            TextColumn::make('project_name')->label('Project Name')->searchable()->sortable(),
            TextColumn::make('framework')->label('Framework')->searchable()->sortable(),
            TextColumn::make('contract_number')->label('Contract Number')->searchable()->sortable(),
            TextColumn::make('status')
                ->label('Status')
                ->formatStateUsing(fn($state) => Project::getStatuses()[$state] ?? '-')
        ])
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
->sortable(),
TextColumn::make('type')
->label('Type')
->formatStateUsing(fn($state) => Project::getTypes()[$state] ?? '-')
->sortable(),
TextColumn::make('budget')->label('Budget')->sortable(),
TextColumn::make('date_start')->label('Start Date')->date()->sortable(),
TextColumn::make('date_end')->label('End Date')->date()->sortable(),
)
->filters([
    // Add filters if needed
])
->actions([
    Tables\Actions\EditAction::make(),
])
->bulkActions([
    Tables\Actions\DeleteBulkAction::make(),
]);
}
```

```
public static function getRelations(): array
```

```
{
    return [
        // Define relations if needed
    ];
}
```

```
public static function getPages(): array
```

```
{
    return [
        'index' => Pages\ListProjects::route('/'),
    ];
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
'create' => Pages\CreateProject::route('/create'),  
'edit' => Pages\EditProject::route('/{record}/edit'),  
];  
}  
}
```

App\Filament\Resources\PublicationResource.php

```
<?php
```

```
namespace App\Filament\Resources;  
  
use App\Filament\Resources\PublicationResource\Pages;  
use App\Filament\Resources\PublicationResource\RelationManagers;  
use App\Models\File;  
use App\Models\People;  
use App\Models\Publication;  
use Filament\Forms;  
use Filament\Forms\Components\Select;  
use Filament\Forms\Form;  
use Filament\Resources\Resource;  
use Filament\Tables;  
use Filament\Tables\Table;  
use Illuminate\Database\Eloquent\Builder;  
use Illuminate\Database\Eloquent\SoftDeletingScope;  
use Filament\Forms\Components\FileUpload;  
use Illuminate\Support\Facades\Log;  
use Illuminate\Support\Facades\Storage;  
use Spatie\PdfToText\Pdf;  
  
class PublicationResource extends Resource  
{  
    protected static ?string $model = Publication::class;  
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
protected static ?string $navigationIcon = 'heroicon-o-rectangle-stack';

public static function form(Form $form): Form
{
    return $form
        ->schema([
            Forms\Components\TextInput::make('title')->required(),
            Forms\Components\Textarea::make('abstract')->required(),
            Forms\Components\TextInput::make('publisher')->required(),
            Forms\Components\DatePicker::make('publication_date')->required(),
            FileUpload::make('file')
                ->required(fn (string $context) => $context === 'create')
                ->directory('files')
                ->disk('public')
                ->maxFiles(1)
                ->acceptedFileTypes(['application/pdf'])
                ->getUploadedFileNameForStorageUsing(function (FileUpload $component, $file) {
                    return $file->hashName();
                })
                ->saveUploadedFileUsing(function ($file) {
                    $path = $file->store('files', 'public');

                    // Create a new File record
                    $fileModel = File::create([
                        'name' => $file->getClientOriginalName(),
                        'file_path' => $path,
                        'size' => $file->getSize(),
                        'type' => $file->getClientMimeType(),
                    ]);
                });
        });
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
        return $fileModel->id;
    })
    ->afterStateUpdated(function ($state, $set) {
        $set('file_id', $state);
    }),
Select::make('publication_type')
    ->required()
    ->options(Publication::getPublicationTypes()),
Forms\Components\Select::make('authors')->multiple()
    ->relationship('authors', 'id')
    ->required()
    ->options(function () {
        return People::all()->mapWithKeys(function ($person) {
            return [$person->id => $person->first_name . ' ' . $person->last_name];
        });
    })->preload(),
]);
}
```

```
public static function table(Table $table): Table
{
    return $table
        ->columns([
            Tables\Columns\TextColumn::make('title'),
            Tables\Columns\TextColumn::make('authors.first_name')
                ->label('Authors')
                ->getStateUsing(function ($record) {
                    return $record->authors->pluck('first_name')->join(', ');
                }),
            Tables\Columns\TextColumn::make('publisher'),
            Tables\Columns\TextColumn::make('publication_type')
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
->label('Publication Type')
->formatStateUsing(function ($state) {
    $types = Publication::getPublicationTypes();
    return $types[$state] ?? 'Unknown';
}),
Tables\Columns\TextColumn::make('file.name')
->label('File')
->url(function ($record) {
    return Storage::url(optional($record->file)->file_path);
})
->getStateUsing(function ($record) {
    return optional($record->file)->name;
}),
Tables\Columns\TextColumn::make('publication_date')->dateTime(),
])
->filters([
    //
])
->actions([
    Tables\Actions\EditAction::make(),
])
->bulkActions([
    Tables\Actions\BulkActionGroup::make([
        Tables\Actions\DeleteBulkAction::make(),
    ]),
]);
}

public static function getRelations(): array
{
    return [
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
//  
];  
}  
  
public static function getPages(): array  
{  
    return [  
        'index' => Pages\ListPublications::route('/'),  
        'create' => Pages\CreatePublication::route('/create'),  
        'edit' => Pages\EditPublication::route('/{record}/edit'),  
    ];  
}  
}
```

App/Filament/Resources/PublicationResource/Pages/CreatePublication.php

<?php

```
namespace App\Filament\Resources\PublicationResource\Pages;
```

```
use App\Filament\Resources\PublicationResource;
```

```
use App\Models\File;
```

```
use App\Models\Publication;
```

```
use Filament\Actions;
```

```
use Filament\Resources\Pages\CreateRecord;
```

```
use Illuminate\Support\Facades\Log;
```

```
use Spatie\PdfToText\Pdf;
```

```
class CreatePublication extends CreateRecord
```

```
{
```

```
    protected static string $resource = PublicationResource::class;
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
protected function handleRecordCreation(array $data): \Illuminate\Database\Eloquent\Model
{
    $fileId = $data['file'];
    unset($data['file']);

    $publication = static::getModel()::create($data);

    $publication->file_id = $fileId;
    $publication->save();

    if (!empty($fileId)) {
        $fileModel = File::find($fileId);
        $path = public_path('storage/' . $fileModel->file_path);
        try {
            // Extract text from the PDF
            $pdfText = Pdf::getText($path);
        } catch (\Exception $e) {
            Log::error('PDF text extraction failed: ' . $e->getMessage(), [
                'file' => $path
            ]);
            $pdfText = null;
        }

        $fileModel->update(['text_content' => $pdfText]);
    }

    return $publication;
}
}
```


App/Filament/Resources/PublicationResource/Pages/EditPublication.php

```
<?php
```

```
namespace App\Filament\Resources\PublicationResource\Pages;
```

```
use App\Models\File;
```

```
use App\Models\Publication;
```

```
use App\Filament\Resources\PublicationResource;
```

```
use Filament\Actions;
```

```
use Filament\Resources\Pages\EditRecord;
```

```
use Illuminate\Database\Eloquent\Model;
```

```
use Illuminate\Support\Facades\Log;
```

```
use Spatie\PdfToText\Pdf;
```

```
class EditPublication extends EditRecord
```

```
{
```

```
    protected static string $resource = PublicationResource::class;
```

```
    protected function getHeaderActions(): array
```

```
    {
```

```
        return [
```

```
            Actions\DeleteAction::make(),
```

```
        ];
```

```
    }
```

```
    protected function handleRecordUpdate(Model $record, array $data): Model
```

```
    {
```

```
        $fileId = $data['file'];
```

```
        unset($data['file']);
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
if (!empty($fileId)) {
    $record->file_id = $fileId;
    $record->save();
}

if (!empty($fileId)) {
    $fileModel = File::find($fileId);
    $path = public_path('storage/' . $fileModel->file_path);
    Log::info('File path for text extraction: ' . $path);
    try {
        $pdfText = Pdf::getText($path);
    } catch (\Exception $e) {
        Log::error('PDF text extraction failed: ' . $e->getMessage(), [
            'file' => $path
        ]);
        $pdfText = null;
    }

    $fileModel->update(['text_content' => $pdfText]);
}

$record->update($data);

return $record;
}
}
```

Routes/api.php

```
<?php
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
use App\Http\Controllers\Api\CoursesController;
use App\Http\Controllers\Api\PeopleController;
use App\Http\Controllers\Api\ProjectsController;
use App\Http\Controllers\Api\PublicationsController;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Route;
```

```
Route::get('/user', function (Request $request) {
    return $request->user();
})->middleware('auth:sanctum');
```

```
Route::apiResource('courses', CoursesController::class);
Route::apiResource('people', PeopleController::class);
Route::apiResource('projects', ProjectsController::class);
Route::apiResource('publications', PublicationsController::class);
```

Migrations

database/migrations/0001_01_01_000000_create_users_table.php

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration
```

```
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
Schema::create('users', function (Blueprint $table) {
    $table->id();
    $table->string('name');
    $table->string('email')->unique();
    $table->timestamp('email_verified_at')->nullable();
    $table->string('password');
    $table->rememberToken();
    $table->timestamps();
});

Schema::create('password_reset_tokens', function (Blueprint $table) {
    $table->string('email')->primary();
    $table->string('token');
    $table->timestamp('created_at')->nullable();
});

Schema::create('sessions', function (Blueprint $table) {
    $table->string('id')->primary();
    $table->foreignId('user_id')->nullable()->index();
    $table->string('ip_address', 45)->nullable();
    $table->text('user_agent')->nullable();
    $table->longText('payload');
    $table->integer('last_activity')->index();
});
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
Schema::dropIfExists('users');
Schema::dropIfExists('password_reset_tokens');
Schema::dropIfExists('sessions');
}
};
```

database/migrations/0001_01_01_000001_create_cache_table.php

```
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('cache', function (Blueprint $table) {
            $table->string('key')->primary();
            $table->mediumText('value');
            $table->integer('expiration');
        });

        Schema::create('cache_locks', function (Blueprint $table) {
            $table->string('key')->primary();
            $table->string('owner');
            $table->integer('expiration');
        });
    }
};
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::dropIfExists('cache');
    Schema::dropIfExists('cache_locks');
}
};

database/migrations/0001_01_01_000002_create_jobs_table.php
<?php
```

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('jobs', function (Blueprint $table) {
            $table->id();
            $table->string('queue')->index();
            $table->longText('payload');
            $table->unsignedTinyInteger('attempts');
            $table->unsignedInteger('reserved_at')->nullable();
        });
    }
}
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
    $table->unsignedInteger('available_at');
    $table->unsignedInteger('created_at');
});
```

```
Schema::create('job_batches', function (Blueprint $table) {
    $table->string('id')->primary();
    $table->string('name');
    $table->integer('total_jobs');
    $table->integer('pending_jobs');
    $table->integer('failed_jobs');
    $table->longText('failed_job_ids');
    $table->mediumText('options')->nullable();
    $table->integer('cancelled_at')->nullable();
    $table->integer('created_at');
    $table->integer('finished_at')->nullable();
});
```

```
Schema::create('failed_jobs', function (Blueprint $table) {
    $table->id();
    $table->string('uuid')->unique();
    $table->text('connection');
    $table->text('queue');
    $table->longText('payload');
    $table->longText('exception');
    $table->timestamp('failed_at')->useCurrent();
});
}
```

```
/**
```

```
* Reverse the migrations.
```

```
*/
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
public function down(): void
{
    Schema::dropIfExists('jobs');
    Schema::dropIfExists('job_batches');
    Schema::dropIfExists('failed_jobs');
}
};
```

database/migrations/2024_05_12_100343_init.php

<?php

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration
```

```
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('file', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('file_path');
            $table->integer('size');
            $table->string('type');
            $table->timestamps();
        });

        Schema::create('course', function (Blueprint $table) {
```


Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
$table->id();  
$table->string('title_en');  
$table->string('title_gr');  
$table->integer('type')->comment('Undergraduate / Graduate');  
$table->timestamps();  
});
```

```
Schema::create('people', function (Blueprint $table) {  
    $table->id();  
    $table->string('first_name');  
    $table->string('last_name');  
    $table->string('role')->comment('Faculty, Researchers')->nullable();  
    $table->string('title')->comment('Assistant professor etc..')->nullable();  
    $table->timestamps();  
});
```

```
Schema::create('projects', function (Blueprint $table) {  
    $table->id();  
    $table->string('project_name');  
    $table->string('framework');  
    $table->string('contract_number');  
    $table->integer('status');  
    $table->integer('type')->comment('EU / GR');  
    $table->text('full_title');  
    $table->text('participants');  
    $table->double('budget');  
    $table->text('description');  
    $table->string('project_url');  
    $table->date('date_start');  
    $table->date('date_end');  
    $table->timestamps();  
});
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
});

Schema::create('publication', function (Blueprint $table) {
    $table->id();
    $table->string('title');
    $table->text('abstract');
    $table->text('publisher');
    $table->date('publication_date');
    $table->integer('publication_type');
    $table->bigInteger('file_id')->unsigned()->nullable();
    $table->timestamps();

    $table->foreign('file_id')->references('id')->on('file');
});
}

/**
 * Reverse the migrations.
 */
public function down(): void
{
    Schema::dropIfExists('file');
    Schema::dropIfExists('course');
    Schema::dropIfExists('people');
    Schema::dropIfExists('projects');
    Schema::dropIfExists('publications');
}
};

database/migrations/2024_05_19_090156_create_author_publication_table.php
<?php
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('author_publication', function (Blueprint $table) {
            $table->id();
            $table->unsignedBigInteger('publication_id');
            $table->unsignedBigInteger('person_id');
            $table->timestamps();

            $table->foreign('publication_id')->references('id')->on('publication')->onDelete('cascade');
            $table->foreign('person_id')->references('id')->on('people')->onDelete('cascade');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('author_publication');
    }
};
```

database/migrations/2024_06_15_134435_add_type_column_to_people_table.php

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('people', function (Blueprint $table) {
            $table->integer('type');
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::table('people', function (Blueprint $table) {
            $table->integer('type');
        });
    }
};
```

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

database/migrations/2024_06_23_100945_add_text_content_to_file_table.php

<?php

```
use Illuminate\Database\Migrations\Migration;
```

```
use Illuminate\Database\Schema\Blueprint;
```

```
use Illuminate\Support\Facades\Schema;
```

```
return new class extends Migration
```

```
{
```

```
/**
```

```
 * Run the migrations.
```

```
*/
```

```
public function up(): void
```

```
{
```

```
    Schema::table('file', function (Blueprint $table) {
```

```
        $table->longText('text_content')->nullable();
```

```
    });
```

```
}
```

```
/**
```

```
 * Reverse the migrations.
```

```
*/
```

```
public function down(): void
```

```
{
```

```
    Schema::table('file', function (Blueprint $table) {
```

```
        $table->dropColumn('text_content');
```

```
    });
```

```
}
```

```
};
```

database/migrations/2024_07_07_100700_add_description_column_to_course_table.php

<?php

Μελέτη και υλοποίηση μίας σύγχρονης ψηφιακής βιβλιοθήκης

```
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::table('course', function (Blueprint $table) {
            $table->text('description')->nullable();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::table('course', function (Blueprint $table) {
            $table->dropColumn('description');
        });
    }
};
```