



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
Προσαρμογή και Επέκταση του Wordpress CMS



Του φοιτητή
Παπαδόπουλου Αλέξανδρου
Αρ. Μητρώου: 164828

Επιβλέπων
Μιχαήλ Σαλαμπάση

Ημερομηνία 2024

Τίτλος Π.Ε.: Προσαρμογή και Επέκταση του Wordpress CMS

Κωδικός Π.Ε.: 22341

Όνοματεπώνυμο φοιτητή: Αλέξανδρος Παπαδόπουλος

Όνοματεπώνυμο εισηγητή: Μιχαήλ Σαλαμπάσης

Ημερομηνία ανάληψης Π.Ε. 14 – 11 - 2022

Ημερομηνία περάτωσης Π.Ε. 5-2024

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

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

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

Πρόλογος

Η επιλογή του θέματος αυτής της πτυχιακής εργασίας έγινε με γνώμονα την αυξανόμενη σημασία των διαδικτυακών εκπαιδευτικών πλατφορμών στην σύγχρονη εκπαίδευση. Ως φοιτητής Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων, θεώρησα εξαιρετικά ενδιαφέρον και επίκαιρο να διερευνήσω την ανάπτυξη μιας τέτοιας πλατφόρμας με τη χρήση προηγμένων τεχνολογιών, όπως το WordPress και τα Advanced Custom Fields (ACF). Η συγκεκριμένη εργασία μου προσέφερε πολύτιμες γνώσεις και εμπειρίες στον τομέα του σχεδιασμού και της ανάπτυξης διαδικτυακών εφαρμογών, ενώ ταυτόχρονα με βοήθησε να κατανοήσω καλύτερα τις ανάγκες και τις προκλήσεις που αντιμετωπίζουν οι εκπαιδευτικοί οργανισμοί στην ψηφιακή εποχή. Ελπίζω η παρούσα εργασία να συμβάλει στην περαιτέρω ανάπτυξη και βελτίωση των εκπαιδευτικών τεχνολογιών και να αποτελέσει χρήσιμο εργαλείο για άλλους ερευνητές και επαγγελματίες του χώρου.

Περίληψη

Στο πλαίσιο αυτής της πτυχιακής εργασίας θα ερευνηθεί και έπειτα θα αναπτυχθεί μια διαδικτυακή εφαρμογή για εκπαιδευτικούς οργανισμούς. Ο στόχος της είναι να κάνει εύκολη την διεπαφή κάποιου καθηγητή με τον μαθητή. Πιο συγκεκριμένα, θα υπάρχει η δυνατότητα στον κάθε καθηγητή να προσθέτει το μάθημά του στο σύστημα με τις μέρες και ώρες που αυτό διδάσκεται. Αντίστοιχα, θα υπάρχει και κάποιο ημερολόγιο όπου θα μπορούν να βλέπουν οι μαθητές πότε και αν θα γίνονται τα μαθήματα, τα οποία θα γίνονται διαδικτυακά. Επίσης, θα μπορεί ο κάθε καθηγητής να έχει και μια σειρά από βιντεοσκοπημένες διαλέξεις των μαθημάτων του όπου θα μπορεί ο κάθε μαθητής να βλέπει όποτε μπορέσει.

Ένας βασικός στόχος της παρούσας πτυχιακής είναι η μελέτη για το πως μπορεί να δημιουργηθούν ευέλικτες, επεκτάσιμες, ημι-δομημένες Web εφαρμογές/ιστοσελίδες με την προσαρμογή και επέκταση του WP CMS και την ευρεία χρήση και δυνατοτήτων Themes.

Η διαδικτυακή αυτή εφαρμογή, υλοποιήθηκε με τη χρήση του εργαλείου WordPress, το οποίο κάνει χρήση της προγραμματιστικής γλώσσας PHP, της γλώσσας φύλου ύφους CSS και της γλώσσας σήμανσης HTML.

Customizing the WordPress CMS: a case study

Alexandros Papadopoulos

Abstract

This thesis explores the implementation of a comprehensive educational platform utilizing Advanced Custom Fields (ACF) in WordPress. The primary focus is on how ACF can be leveraged to create and manage custom fields for various content types, such as instructors and courses, thereby enhancing the organization and presentation of educational material. The platform features dynamic connections between different types of content, allowing for efficient linking of instructors to their respective courses. Additionally, the integration of WordPress shortcodes facilitates the dynamic display of content within pages and posts, further streamlining the management and user experience of the educational platform. Key benefits include the ability to customize field groups, organize course information effectively, and dynamically link related content to enhance navigability and relevance. One of the main objectives of this thesis is to study how flexible, extensible, semi-structured web applications/websites can be created by adapting and extending the WP CMS and leveraging the extensive capabilities of themes. The thesis details the technical implementation steps, including the use of specific ACF functionalities and the creation of shortcodes, and discusses the resultant improvements in educational content delivery.

Περιεχόμενα

Πρόλογος.....	3
Περίληψη.....	4
Abstract	5
Περιεχόμενα	6
Κεφάλαιο 1ο: Εισαγωγή.....	8
1.1 Βασικός σκοπός της πτυχιακής.....	8
1.2 Τεχνική Περιγραφή του WordPress	8
1.2.1 Βασικά Χαρακτηριστικά του WordPress	8
1.2.2 Plugins.....	9
1.2.3 Themes	9
1.2.4 Shortcodes	9
1.3 Περιορισμοί των Απλών Σελίδων και Δημοσιεύσεων	10
1.4 Λύσεις με Themify και Plugins.....	10
1.5 Επίλογος.....	10
Κεφάλαιο 2ο: Έρευνα συστημάτων εκπαιδευτικού σκοπού	10
2.1 Εισαγωγή.....	10
2.2 Σύστημα εκπαιδευτικού σκοπού Udemy.....	11
2.3 Σύστημα εκπαιδευτικού σκοπού Coursera	12
2.4 Σύστημα εκπαιδευτικού σκοπού hbstudio.....	12
2.5 Επίλογος.....	13
Κεφάλαιο 3ο: Τεχνολογίες που χρησιμοποιήθηκαν	14
3.1 Εισαγωγή.....	14
3.2 Τεχνολογίες και εργαλία	14
3.2.1 Wordpress.....	14
3.2.2 Θέμα – Themify	15
3.2.3 Advanced Custom Fields (ACF)	16
3.2.4 Wordpress Shortcodes.....	16
3.3 Επίλογος.....	17
Κεφάλαιο 4ο: Υλοποίηση.....	17
4.1 Εισαγωγή.....	17

4.2	Δημιουργία Layout Parts (Themify).....	18
4.2.1	Teacher Layout Part	18
4.2.2	Lesson Layout Part.....	18
4.3	Δημιουργία Κατηγορίας.....	19
4.4	Χρήση Advanced Custom Fields	20
4.4.1	Teacher Group.....	20
4.4.2	Lesson Group	21
4.5	Χρήση του ACF group	22
4.6	Κώδικας.....	23
4.7	Teacher Shortcodes	24
4.8	Lessons Shortcodes	29
4.9	Επίλογος.....	45
Κεφάλαιο 5ο: Συμπεράσματα.....		46
5.1	Περίληψη.....	46
5.2	Κύρια Επιτεύγματα	46
5.3	Προκλήσεις και Λύσεις.....	46
5.4	Μελλοντικές Κατευθύνσεις.....	46

Κεφάλαιο 1ο: Εισαγωγή

1.1 Βασικός σκοπός της πτυχιακής

Ο κύριος στόχος αυτής της πτυχιακής εργασίας είναι η μελέτη των τρόπων με τους οποίους μπορεί να προσαρμοστεί, να επεκταθεί και να δομηθεί το WordPress για την ανάπτυξη ευέλικτων και επεκτάσιμων web εφαρμογών. Η εργασία αυτή επικεντρώνεται στην αξιοποίηση των δυνατοτήτων που προσφέρει το WordPress CMS (Content Management System) μέσω της χρήσης θεμάτων (themes), πρόσθετων (plugins) και συντομογραφιών κώδικα (shortcodes).

1.2 Τεχνική Περιγραφή του WordPress

Το WordPress είναι ένα σύστημα διαχείρισης περιεχομένου (CMS) ανοιχτού κώδικα, το οποίο χρησιμοποιείται ευρέως για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Αποτελεί μια από τις πιο δημοφιλείς πλατφόρμες λόγω της ευελιξίας, της επεκτασιμότητας και της ευκολίας χρήσης του.

1.2.1 Βασικά Χαρακτηριστικά του WordPress

- Πυρήνας του WordPress (WordPress Core):
 - Πρόσβαση και Διαχείριση: Ο πυρήνας του WordPress παρέχει τη βασική λειτουργικότητα που επιτρέπει στους χρήστες να διαχειρίζονται το περιεχόμενο της ιστοσελίδας τους. Περιλαμβάνει το σύστημα χρήστη, το σύστημα ρόλων και δικαιωμάτων, καθώς και το περιβάλλον διαχείρισης (Dashboard).
 - Διαχείριση Περιεχομένου: Επιτρέπει τη δημιουργία, επεξεργασία και δημοσίευση περιεχομένου, όπως άρθρα (posts) και σελίδες (pages), με μια φιλική προς το χρήστη διεπαφή.
- Θέματα (Themes):
 - Εμφάνιση και Σχεδιασμός: Τα θέματα καθορίζουν την εμφάνιση και τη διάταξη της ιστοσελίδας. Μπορούν να προσαρμοστούν μέσω του Customizer του WordPress ή με τη χρήση παιδικών θεμάτων (child themes) για να διατηρηθούν οι αλλαγές κατά τις ενημερώσεις.
 - Δημιουργία Προσαρμοσμένων Θεμάτων: Οι προγραμματιστές μπορούν να δημιουργήσουν από την αρχή προσαρμοσμένα θέματα που καλύπτουν συγκεκριμένες ανάγκες.
- Πρόσθετα (Plugins):
 - Επέκταση Λειτουργικότητας: Τα plugins προσθέτουν νέες λειτουργίες και δυνατότητες στο WordPress. Μπορούν να χρησιμοποιηθούν για την προσθήκη SEO εργαλείων, φόρμες επικοινωνίας, ηλεκτρονικών καταστημάτων και πολλών άλλων.
 - Δημιουργία Προσαρμοσμένων Plugins: Οι προγραμματιστές μπορούν να δημιουργήσουν προσαρμοσμένα plugins για να καλύψουν συγκεκριμένες ανάγκες που δεν καλύπτονται από τα διαθέσιμα plugins.
- Συντομογραφίες Κώδικα (Shortcodes):
 - Δυναμικό Περιεχόμενο: Τα shortcodes επιτρέπουν την εισαγωγή δυναμικού περιεχομένου στις σελίδες και τις δημοσιεύσεις. Μπορούν να χρησιμοποιηθούν για την προσθήκη σύνθετων στοιχείων, όπως φόρμες, γκαλερί και διαδραστικό περιεχόμενο.
 - Εύκολη Ενσωμάτωση: Οι συντομογραφίες είναι εύκολο να ενσωματωθούν και να διαχειριστούν, κάνοντας την προσθήκη νέων λειτουργιών απλή και γρήγορη.
- Προσαρμοσμένοι Τύποι Αναρτήσεων (Custom Post Types):

- Οργάνωση Περιεχομένου: Επιτρέπουν τη δημιουργία διαφορετικών τύπων περιεχομένου πέρα από τις τυπικές αναρτήσεις και σελίδες, όπως προϊόντα, γεγονότα, μαθήματα κ.ά.
- Προσαρμοσμένες Ταξινομίες (Custom Taxonomies): Χρησιμοποιούνται για την κατηγοριοποίηση και την οργάνωση περιεχομένου με πιο εξειδικευμένους τρόπους.
- API (Application Programming Interface):
 - REST API: Το WordPress REST API επιτρέπει την αλληλεπίδραση με την πλατφόρμα μέσω HTTP αιτημάτων, διευκολύνοντας την ενσωμάτωση και την επικοινωνία με άλλες εφαρμογές και υπηρεσίες.
 - Hooks (Actions & Filters): Επιτρέπουν στους προγραμματιστές να συνδέονται σε συγκεκριμένα σημεία της ροής εκτέλεσης του WordPress και να τροποποιούν τη συμπεριφορά του χωρίς να επεμβαίνουν απευθείας στον κώδικα του πυρήνα.
- Διαχείριση Χρηστών και Δικαιωμάτων:
 - Ρόλοι και Δικαιώματα: Το σύστημα ρόλων και δικαιωμάτων του WordPress επιτρέπει τον καθορισμό διαφορετικών επιπέδων πρόσβασης και δικαιωμάτων για τους χρήστες της ιστοσελίδας, από διαχειριστές έως συντάκτες και συνδρομητές.

1.2.2 Plugins

Τα plugins είναι επεκτάσεις που προσθέτουν νέες δυνατότητες και λειτουργίες στο WordPress. Μερικοί τρόποι με τους οποίους τα plugins μπορούν να προσαρμόσουν και να επεκτείνουν τις δυνατότητες μιας WordPress εφαρμογής περιλαμβάνουν:

- Προσθήκη νέων λειτουργιών: Τα plugins μπορούν να προσθέσουν νέα χαρακτηριστικά, όπως φόρμες επικοινωνίας, διαχείριση χρηστών, SEO εργαλεία και πολλά άλλα.
- Ενσωμάτωση με τρίτες υπηρεσίες: Μέσω plugins, το WordPress μπορεί να συνδεθεί με εξωτερικές υπηρεσίες όπως κοινωνικά δίκτυα, εργαλεία ανάλυσης και ηλεκτρονικά καταστήματα.
- Βελτίωση της ασφάλειας: Plugins όπως τα Wordfence και iThemes Security προσθέτουν επιπλέον επίπεδα ασφαλείας στις WordPress εφαρμογές.

1.2.3 Themes

Τα θέματα στο WordPress καθορίζουν την εμφάνιση και τη δομή της ιστοσελίδας. Υπάρχουν διάφοροι τρόποι προσαρμογής και επέκτασης μέσω των θεμάτων:

- Προσαρμογή εμφάνισης: Τα θέματα επιτρέπουν την αλλαγή της αισθητικής της ιστοσελίδας μέσω προσαρμοσμένων σχεδίων, χρωμάτων, γραμματοσειρών και layout.
- Χρήση Child Themes: Τα child themes επιτρέπουν την προσαρμογή ενός υπάρχοντος θέματος χωρίς να χάνονται οι αλλαγές κατά τις ενημερώσεις του αρχικού θέματος.
- Δημιουργία Custom Themes: Οι προγραμματιστές μπορούν να δημιουργήσουν από την αρχή θέματα που καλύπτουν συγκεκριμένες ανάγκες και απαιτήσεις ενός έργου.

1.2.4 Shortcodes

Τα shortcodes είναι σύντομες γραμμές κώδικα που ενσωματώνονται σε σελίδες και δημοσιεύσεις για την προσθήκη δυναμικού περιεχομένου:

- Εύκολη εισαγωγή περιεχομένου: Τα shortcodes επιτρέπουν στους χρήστες να προσθέτουν σύνθετα στοιχεία, όπως φόρμες, χάρτες και γκαλερί, χωρίς την ανάγκη για κωδικοποίηση.
- Επαναχρησιμοποιήσιμα στοιχεία: Οι σύντομοι κώδικες μπορούν να δημιουργηθούν μια φορά και να χρησιμοποιηθούν επανειλημμένα σε διαφορετικές σελίδες και δημοσιεύσεις, βελτιώνοντας την αποδοτικότητα.
- Δυναμικό περιεχόμενο: Χρησιμοποιώντας shortcodes, το περιεχόμενο μπορεί να εμφανίζεται δυναμικά, προσαρμόζοντας την εμφάνισή του ανάλογα με τις ανάγκες των χρηστών.

1.3 Περιορισμοί των Απλών Σελίδων και Δημοσιεύσεων

Οι απλές σελίδες και δημοσιεύσεις του WordPress συχνά δεν επαρκούν για τη διαχείριση σύνθετων εκπαιδευτικών πλατφορμών για τους εξής λόγους:

- Έλλειψη Ευελιξίας: Οι απλές σελίδες και δημοσιεύσεις παρέχουν περιορισμένες δυνατότητες προσαρμογής των πεδίων και των στοιχείων που εμφανίζονται.
- Δυναμική Σύνδεση Περιεχομένου: Δεν υποστηρίζουν εύκολα τη δυναμική σύνδεση μεταξύ διαφορετικών τύπων περιεχομένου, όπως μαθήματα και καθηγητές, που είναι απαραίτητη για μια εκπαιδευτική πλατφόρμα.

Αυτοί οι περιορισμοί καθιστούν αναγκαία τη χρήση πρόσθετων εργαλείων και τεχνολογιών για την κάλυψη των απαιτήσεων μιας σύγχρονης εκπαιδευτικής πλατφόρμας.

1.4 Λύσεις με Themify και Plugins

Η χρήση του θέματος Themify και διάφορων plugins επιτρέπει την αντιμετώπιση των περιορισμών των απλών σελίδων και δημοσιεύσεων μέσω των εξής δυνατοτήτων:

- Προσαρμοσμένα Layouts: Το Themify επιτρέπει τη δημιουργία προσαρμοσμένων layout parts, που βελτιώνουν την οπτική ελκυστικότητα και τη χρηστικότητα της πλατφόρμας. Η δυνατότητα αυτή επιτρέπει στους χρήστες να δημιουργούν και να διαχειρίζονται διαφορετικές διατάξεις με ευκολία και χωρίς την ανάγκη προχωρημένης κωδικοποίησης.
- Κατηγοριοποίηση Περιεχομένου: Η χρήση των Advanced Custom Fields (ACF) επιτρέπει τη δημιουργία προσαρμοσμένων πεδίων και τη δυναμική κατηγοριοποίηση του περιεχομένου. Αυτό διευκολύνει την οργάνωση και την παρουσίαση του εκπαιδευτικού υλικού, καθιστώντας το πιο προσβάσιμο και κατανοητό για τους μαθητές.
- Δημιουργία Shortcodes: Η δημιουργία shortcodes διευκολύνει την εισαγωγή και διαχείριση δυναμικού περιεχομένου σε σελίδες και δημοσιεύσεις, βελτιώνοντας έτσι την εμπειρία του χρήστη και την πλοήγηση. Τα shortcodes επιτρέπουν την προσθήκη λειτουργιών και στοιχείων που διαφορετικά θα απαιτούσαν εκτενή κωδικοποίηση.

1.5 Επίλογος

Η ευελιξία του WordPress επιτρέπει την προσαρμογή, επέκταση και δόμηση web εφαρμογών με διάφορους τρόπους, καθιστώντας το ένα ισχυρό εργαλείο για την ανάπτυξη σύγχρονων και δυναμικών ιστοσελίδων. Μέσω της χρήσης plugins, themes, shortcodes και άλλων τεχνικών, οι προγραμματιστές μπορούν να δημιουργήσουν μοναδικές και λειτουργικές λύσεις που ικανοποιούν τις συγκεκριμένες ανάγκες των χρηστών.

Κεφάλαιο 2ο: Έρευνα συστημάτων εκπαιδευτικού σκοπού

2.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα διερευνήσουμε διάφορα συστήματα εκπαιδευτικού σκοπού, προκειμένου να παράσχουμε ένα υπόβαθρο για την ανάπτυξη της προτεινόμενης εκπαιδευτικής πλατφόρμας. Η ανάλυση περιλαμβάνει δημοφιλείς πλατφόρμες όπως το Udemy και το Coursera, αναδεικνύοντας τα χαρακτηριστικά, τα πλεονεκτήματα και τα μειονεκτήματά τους.

2.2 Σύστημα εκπαιδευτικού σκοπού UdeMy

The screenshot shows the UdeMy website interface. At the top, there is a navigation bar with the UdeMy logo, a search bar, and links for 'UdeMy Business', 'Teach on UdeMy', 'Log in', and 'Sign up'. Below the navigation bar is a promotional banner for new students, stating 'New to UdeMy? Lucky you. Courses start at €14.99. Get your new-student offer before it expires.' and featuring an illustration of a hand holding a scroll over an alarm clock. Underneath the banner is a section titled 'A broad selection of courses' with a sub-header 'Choose from 213,000 online video courses with new additions published every month'. A list of course categories is provided: Python, Excel, Web Development, JavaScript, Data Science, AWS Certification, and Drawing. A featured course for Python is highlighted with the text 'Expand your career opportunities with Python' and a 'Explore Python' button.

Το UdeMy είναι μια διαδικτυακή εφαρμογή που παρέχει διάφορες σειρές μαθημάτων με βιντεοσκοπημένες διαλέξεις.

Θετικά

- Στο UdeMy μπορεί κάποιος μαθητής να δει το περιεχόμενο του μαθήματος και τον αριθμό των βίντεο που θα πρέπει να παρακολουθήσει.
- Μπορεί κάποιος να δει πόσα και ποια μαθήματα έχει ο κάθε Διδάσκοντας.
- Δυνατότητα λήψης πιστοποίησης.
- Εύκολη επικοινωνία μεταξύ καθηγητή και μαθητή

Αρνητικά

- Μπορεί οποιοσδήποτε να αυτοαποκαλείται καθηγητής και να προσφέρει σειρές από δικά του μαθήματα.
- Δεν υπάρχει η έννοια του ζωντανού μαθήματος και η χρήση του ημερολογίου

2.3 Σύστημα εκπαιδευτικού σκοπού Coursera

Learn without limits

Start, switch, or advance your career with more than 5,400 courses, Professional Certificates, and degrees from world-class universities and companies.

Join for Free Try Coursera for Business

We collaborate with 275+ leading universities and companies

ILLINOIS Duke Google M IBM Imperial College London Stanford Penn

Θετικά

- Στο Coursera υπάρχει η έννοια του καθηγητή σαν πιο σοβαρό πρόσωπο και δεν μπορεί να γίνει οποιοσδήποτε.
- Δυνατότητα λήψης πιστοποίησης.
- Μπορεί κάποιος να δει πόσα και ποια μαθήματα έχει ο κάθε Διδάσκοντας.
- Επικοινωνία μεταξύ καθηγητή και μαθητή.

Αρνητικά

- Δεν υπάρχει η έννοια του ζωντανού μαθήματος και η χρήση του ημερολογίου.

HB STUDIO'S 25TH ANNIVERSARY

HB ABOUT CLASSES PROGRAMS CALENDAR RENTALS NEWS CONTACT GIVING LOGIN DONATE

PROFESSIONAL CLASSES IN THE DRAMATIC ARTS

HB Studio offers both [online classes](#) and [in-person classes](#) at the Studio. Proof of Covid-19 vaccination with Booster is required for in-person classes. Before your first class, please review the information outlined on the relevant page(s): [Online Classes](#) | [In-Person Classes](#)

Classes are in NYC's Eastern Time Zone.

Weekly Classes

HB Studio offers professional classes in the dramatic arts for committed actors, directors, and playwrights—beginning and experienced—to hone their skills in a community of peers and mentors, in the spirit of our founders, [Uta Hagen](#) and [Herbert Berghof](#). We seek to foster an environment where creative freedom prevails and artists are challenged to take their work to the next level.

Browse classes in the navigation menu to the left (desktop) or above (mobile).

HB offers a progressive sequence of ongoing weekly classes, which meet for 10-week terms four terms a year. All online classes (except writing classes) are offered in two 5-week parts (Part A & Part B) which can be taken independently or together.

Offerings include:

QUICK LINKS

[Faculty](#)

[Full-Time Hagen Core Training](#)

[Hagen Summer Intensive](#)

[International Students](#)

[Financial Aid](#)

[REGISTER HERE](#)

2.4 Σύστημα εκπαιδευτικού σκοπού hbstudio

Το hbstudio είναι πιο κοντά σε αυτό που θα υλοποιήσαμε στην πτυχιακή αυτή.

Θετικά

- Έχει την έννοια του ημερολογίου για τα μαθήματα.
- Υπάρχουν οι διαδικτυακές σειρές μαθημάτων.
- Έχει την έννοια του καθηγητή σε λίγο πιο σοβαρά πλαίσια καθώς δεν μπορεί να γίνει όποιος θέλει και να ανεβάζει υλικό

Αρνητικά

- Η πληροφορία παρουσιάζεται ασύνδετα και δυσανάγνωστα, με αποτέλεσμα ο τελικός χρήστης να μπερδεύεται και να δυσκολεύεται στην πλοήγηση

2.5 Επίλογος

Στο παρόν κεφάλαιο έγινε μια σύντομη αναφορά σε web applications που προσφέρουν παρόμοιες λειτουργίες με την πτυχιακή εργασία. Παρατηρήθηκε, όμως, ότι δεν υφίσταται έως τώρα λύση που να ενσωματώνει το σύνολο των επιθυμητών χαρακτηριστικών.

Στο επόμενο κεφάλαιο θα εξεταστεί η υλοποίηση της πτυχιακής εργασίας με χρήση WordPress.

Κεφάλαιο 3ο: Τεχνολογίες που χρησιμοποιήθηκαν

3.1 Εισαγωγή

Η υλοποίηση της πτυχιακής εργασίας βασίστηκε σε ένα σύνολο σύγχρονων τεχνολογιών, οι οποίες συνεργάστηκαν αρμονικά για την δημιουργία ενός ολοκληρωμένου και λειτουργικού συστήματος.

Στο παρόν κεφάλαιο θα γίνει λεπτομερής ανάλυση των τεχνολογιών που χρησιμοποιήθηκαν και του κώδικα, εστιάζοντας στα χαρακτηριστικά, τα πλεονεκτήματα και τον τρόπο με τον οποίο συνέβαλαν στην υλοποίηση της πτυχιακής εργασίας.

Συγκεκριμένα, θα εξεταστούν:

- WordPress: Το δημοφιλές CMS που αποτέλεσε την βάση για την ανάπτυξη του συστήματος.
- Plugin ACF: Ένα ισχυρό plugin που επέτρεψε την εύκολη δημιουργία custom fields και δυναμικών περιεχομένων.
- Google Calendar API: Η ενσωμάτωση με το Google Calendar
- Themify: Το theme που χρησιμοποιήθηκε για την δημιουργία μίας εύχρηστης και κομψής εμφάνισης.

Στόχος του κεφαλαίου:

- Να εξηγήσει λεπτομερώς τις τεχνολογίες.
- Να αναδείξει τα οφέλη και τις δυνατότητες που προσφέρουν.

3.2 Τεχνολογίες και εργαλία

3.2.1 Wordpress

Το WordPress είναι ένα κορυφαίο σύστημα διαχείρισης περιεχομένου (CMS) ανοιχτού κώδικα, γνωστό για την ισχύ και την ευελιξία του. Ένα CMS παρέχει μια φιλική προς το χρήστη διαχειριστική διεπαφή για τη δημιουργία, επεξεργασία και διαχείριση περιεχομένου ιστοτόπου χωρίς την ανάγκη για βαθιές γνώσεις HTML, CSS ή προγραμματισμού. Το WordPress υπερέχει σε αυτόν τον τομέα, προσφέροντας στους ιδιοκτήτες ιστοτόπων έναν εύχρηστο τρόπο για να δημιουργούν και να διατηρούν δυναμικούς ιστοτόπους.

Γιατί WordPress;

- Ευελιξία και επεκτασιμότητα: Το WordPress προσφέρει απaráμιλλη δυνατότητα προσαρμογής μέσω της τεράστιας βιβλιοθήκης θεμάτων και plugins του. Αυτό επιτρέπει την προσθήκη συγκεκριμένων χαρακτηριστικών και λειτουργιών χωρίς να χρειαστεί να δημιουργήσετε κάθε στοιχείο από το μηδέν.
- Ευκολία χρήσης: Ο διαισθητικός πίνακας διαχείρισης του WordPress το καθιστά προσβάσιμο σε χρήστες με διαφορετικό επίπεδο τεχνικής εμπειρίας. Οι δημιουργοί περιεχομένου και οι διαχειριστές μπορούν εύκολα να προσθέσουν νέες σελίδες, να ανεβάσουν μέσα ενημέρωσης και να διαχειριστούν τις ρυθμίσεις του ιστοτόπου με ελάχιστη εκπαίδευση.
- Ισχυρή υποστήριξη κοινότητας: Η τεράστια κοινότητα του WordPress παρέχει πληθώρα πόρων, όπως τεκμηρίωση, οδηγίες, φόρουμ και υποστήριξη προγραμματιστών. Αυτό μεταφράζεται σε άμεσα διαθέσιμη βοήθεια και λύσεις σε κοινές προκλήσεις.

Το WordPress στην εκπαιδευτική πλατφόρμα:

- **Θεμέλιο περιεχομένου:** Το WordPress λειτουργεί ως η ραχοκοκαλιά της δομής του ιστοτόπου. Αναλαμβάνει την οργάνωση σελίδων, δημοσιεύσεων ιστολογίου, περιγραφών μαθημάτων, προφίλ εκπαιδευτικών και άλλων βασικών τύπων περιεχομένου.
- **Διαχείριση χρηστών:** Το WordPress παρέχει τους μηχανισμούς για εγγραφή χρήστη, σύνδεση και βασική διαχείριση προφίλ, απλοποιώντας τη διαδικασία εγγραφής των μαθητών στην πλατφόρμα.
- **Βάση σχεδιασμού:** Ενώ εφαρμόζεται προσαρμοσμένο στυλ, το σύστημα θεμάτων του WordPress παρέχει ένα σημείο εκκίνησης για τη συνολική εμφάνιση και διάταξη του ιστοτόπου.
- **Κέντρο λειτουργικότητας:** Το WordPress λειτουργεί ως το κεντρικό σημείο όπου ο προσαρμοσμένος κώδικας, τα plugins όπως τα Advanced Custom Fields και οι εξωτερικές ενσωματώσεις όπως το Google Calendar API ενώνονται για να προσφέρουν την πλήρη λειτουργικότητα της εκπαιδευτικής πλατφόρμας.

3.2.2 Θέμα – Themify

Για να παρέχουμε μια οπτικά ελκυστική και φιλική προς το χρήστη δομή για αυτήν την εκπαιδευτική πλατφόρμα, επιλέχθηκε το θέμα Themify. Το Themify προσφέρει ευελιξία στον εξατομικευμένο σχεδιασμό, επιτρέποντας τη δημιουργία μοναδικών διατάξεων για την καλύτερη αναπαράσταση του περιεχομένου και των λειτουργιών της πλατφόρμας.

Αναλυτική Περιγραφή του Θέματος Themify

1. Ευελιξία Σχεδιασμού:

Το Themify επιτρέπει τη δημιουργία προσαρμοσμένων διατάξεων μέσω ενός φιλικού προς το χρήστη drag-and-drop builder. Οι χρήστες μπορούν να διαμορφώσουν τις σελίδες τους χωρίς την ανάγκη για προχωρημένες γνώσεις προγραμματισμού, χρησιμοποιώντας προετοιμασμένα layout parts και modules.

2. Προσαρμοσμένα Layout Parts:

Ένα από τα βασικά χαρακτηριστικά του Themify είναι η χρήση των "Layout Parts". Αυτά είναι επαναχρησιμοποιήσιμα μπλοκ περιεχομένου που μπορούν να σχεδιαστούν και να εισαχθούν σε διαφορετικούς τύπους σελίδων. Στην περίπτωση αυτής της πλατφόρμας, δημιουργήθηκαν προσαρμοσμένα layout parts για καθηγητές και μαθήματα, τα οποία εξασφαλίζουν μια συνεπή και οργανωμένη παρουσίαση των πληροφοριών.

3. Ενσωμάτωση με ACF (Advanced Custom Fields):

Για να συμπληρωθούν οι διατάξεις καθηγητών και μαθημάτων με δυναμικές πληροφορίες, αναπτύχθηκαν προσαρμοσμένοι κώδικες Shortcode σε συνδυασμό με τα Advanced Custom Fields (ACF). Οι κώδικες Shortcode λειτουργούν ως αποθέσεις μέσα στο Layout Part και, όταν αποδίδεται η σελίδα, τα ACF εμφανίζουν τις κατάλληλες πληροφορίες από τη βάση δεδομένων.

4. Βελτιωμένη Χρήση και Διαχείριση Περιεχομένου:

Το Themify επιτρέπει στους διαχειριστές του ιστότοπου να προσθέτουν και να διαχειρίζονται περιεχόμενο εύκολα μέσω του WordPress backend. Οι αλλαγές μπορούν να γίνουν άμεσα και χωρίς προβλήματα, ενώ οι προσαρμοσμένες διατάξεις εξασφαλίζουν ότι το περιεχόμενο παρουσιάζεται με έναν οπτικά ελκυστικό και οργανωμένο τρόπο.

5. Συμβατότητα και Απόδοση:

Το Themify είναι συμβατό με τα περισσότερα plugins και widgets του WordPress, γεγονός που το καθιστά ιδανικό για τη δημιουργία μιας ολοκληρωμένης εκπαιδευτικής πλατφόρμας. Επίσης, είναι βελτιστοποιημένο για ταχύτητα και απόδοση, διασφαλίζοντας γρήγορη φόρτωση των σελίδων και ομαλή εμπειρία χρήστη.

Συμπέρασμα

Η χρήση του θέματος Themify παρείχε την απαραίτητη ευελιξία και τα εργαλεία για τη δημιουργία μιας οπτικά ελκυστικής, λειτουργικής και εύχρηστης εκπαιδευτικής πλατφόρμας. Οι δυνατότητες προσαρμογής και η ενσωμάτωση με το ACF εξασφάλισαν ότι οι πληροφορίες μπορούν να διαχειρίζονται και να εμφανίζονται δυναμικά, καλύπτοντας πλήρως τις ανάγκες της πλατφόρμας.

3.2.3 Advanced Custom Fields (ACF)

Το Advanced Custom Fields (ACF) είναι ένα ισχυρό plugin για WordPress που επεκτείνει σημαντικά τις δυνατότητες διαχείρισης περιεχομένου. Ξεπερνά τα βασικά πεδία που προσφέρει το WordPress (όπως τίτλος, περιγραφή και κύρια εικόνα) επιτρέποντας την προσθήκη προσαρμοσμένων πεδίων (custom fields) σε σελίδες, δημοσιεύσεις, χρήστες, ταξινομήσεις, και άλλους τύπους περιεχομένου. Η διαισθητική διεπαφή του ACF εξασφαλίζει εύκολη δημιουργία και διαχείριση αυτών των προσαρμοσμένων πεδίων, χωρίς να απαιτεί κατανόηση εκτενούς προγραμματισμού.

Πώς το ACF χρησιμοποιήθηκε στην εκπαιδευτική πλατφόρμα

Στην υλοποίηση της εκπαιδευτικής πλατφόρμας, το ACF χρησιμοποιήθηκε με τρόπο που αποτέλεσε θεμέλιο στοιχείο:

- **Δημιουργία πεδίων για καθηγητές:** Το ACF μας επέτρεψε να εισάγουμε συγκεκριμένα πεδία πληροφοριών σχετικά με κάθε καθηγητή, όπως όνομα, φωτογραφία, σύντομο βιογραφικό, καθώς και μαθήματα που διδάσκει. Αυτά τα προσαρμοσμένα πεδία ομαδοποιήθηκαν σε "Ομάδες Πεδίων" (Field Groups), δίνοντας δομή και οργάνωση στη διαχείριση πληροφοριών κάθε καθηγητή.
- **Δημιουργία πεδίων για μαθήματα:** Παράλληλα, αξιοποιήσαμε το ACF για να προσθέσουμε οργανωμένα πεδία σχετικά με τα μαθήματα της πλατφόρμας. Αυτά μπορεί να περιλαμβάνουν τον τίτλο, μια περιγραφή, μια κύρια εικόνα, τους βασικούς στόχους του μαθήματος ή ακόμα και επιπλέον σχετικό υλικό.
- **Δυναμικές Συνδέσεις:** Ένα ζωτικής σημασίας πλεονέκτημα του ACF είναι η δυνατότητα δημιουργίας σχέσεων μεταξύ διαφορετικών τύπων περιεχομένου. Στην περίπτωση μας, χρησιμοποιήσαμε πεδία σχέσεων για να συνδέσουμε καθηγητές με τα συγκεκριμένα μαθήματα που διδάσκουν, διευκολύνοντας την οργάνωση και την παρουσίαση των πληροφοριών.
- **Ευέλικτη εμφάνιση περιεχομένου:** Σε συνδυασμό με τους shortcodes, το ACF αποτέλεσε τη βάση για να εμφανίζονται δυναμικά οι πληροφορίες καθηγητών και μαθημάτων στις αντίστοιχες σελίδες. Τα Shortcodes λειτουργούν ως "placeholders" μέσα στο περιεχόμενο, αντλώντας τα δεδομένα απευθείας από τα πεδία ACF και επιτρέποντας ευελιξία και ευκολία στις ενημερώσεις.

3.2.4 Wordpress Shortcodes.

Τα Shortcodes στο WordPress είναι απλές, κωδικοποιημένες οδηγίες που τοποθετούνται μέσα στο περιεχόμενό σας και δίνουν τη δυνατότητα εύκολης εμφάνισης δυναμικών στοιχείων σε σελίδες,

δημοσιεύσεις ή widgets. Στην ουσία, λειτουργούν ως "συντομογραφίες" που κατά την επεξεργασία της σελίδας, αντικαθίστανται από πιο σύνθετο περιεχόμενο ή εκτελούν συγκεκριμένες λειτουργίες.

3.2.4.1 Πώς λειτουργούν τα Shortcodes

- Δημιουργία Shortcode: Ένα shortcode ορίζεται μέσα στον PHP κώδικα, συνήθως στο functions.php αρχείο του theme σας ή με τη χρήση ενός plugin. Σε αυτό το στάδιο, ορίζετε την εμφάνιση του shortcode (πχ. [gallery]) και τον κώδικα που θα εκτελεστεί όταν βρεθεί μέσα στο περιεχόμενο.
- Χρήση Shortcode: Όταν δημιουργείτε μία σελίδα ή δημοσίευση, μπορείτε να εισάγετε shortcodes μέσα στο πρόγραμμα επεξεργασίας κειμένου του WordPress. Αυτά οπτικά μοιάζουν κλεισμένα σε αγκύλες, όπως [my_shortcode].
- Επεξεργασία Σελίδας: Όταν κάποιος επισκέπτεται τη σελίδα, ο κώδικας του WordPress εντοπίζει τα shortcodes και εκτελεί τις αντίστοιχες οδηγίες που έχετε ορίσει. Το αποτέλεσμα "υποκαθιστά" το shortcode στην τελική εμφάνιση της σελίδας.

3.2.4.2 Πλεονεκτήματα χρήσης Shortcodes

- Δυναμικό περιεχόμενο: Εύκολη προσθήκη σύνθετης λειτουργικότητας όπως γκαλερί, φόρμες, χάρτες, ή ακόμα και προσαρμοσμένα layout χωρίς εκτενή κώδικα.
- Ευκολία χρήσης: Δεν απαιτούνται γνώσεις προγραμματισμού για την χρήση τους.
- Οργάνωση: Κρατούν τον κώδικά σας καθαρό και τακτοποιημένο.

Τύποι Shortcodes

- Built-in: Το WordPress έρχεται προεγκατεστημένο με μερικά βασικά shortcodes (πχ. [gallery], [caption]).
- Plugins: Πολλά plugins προσθέτουν τα δικά τους εξειδικευμένα shortcodes.
- Custom: Μπορείτε να δημιουργήσετε τα δικά σας shortcodes για πλήρη προσαρμογή.

Εμείς θα ασχοληθούμε με τα Custom Short Codes

3.3 Επίλογος

Συνοψίζοντας, οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εκπαιδευτικής πλατφόρμας προσέφεραν σημαντικά πλεονεκτήματα όσον αφορά την ευελιξία, την οργάνωση και τη δυναμική παρουσίαση των πληροφοριών. Το WordPress, σε συνδυασμό με το Advanced Custom Fields (ACF) και τα shortcodes, επέτρεψε την αποτελεσματική δημιουργία και διαχείριση προσαρμοσμένων πεδίων για καθηγητές και μαθήματα, διασφαλίζοντας την εύκολη ενημέρωση και παρουσίαση των δεδομένων. Η υλοποίηση των δυναμικών συνδέσεων μεταξύ των διαφόρων τύπων περιεχομένου βελτίωσε σημαντικά την εμπειρία χρήστη, διευκολύνοντας την πλοήγηση και την πρόσβαση στις σχετικές πληροφορίες. Τα εργαλεία που χρησιμοποιήθηκαν απέδειξαν ότι μπορούν να προσφέρουν ολοκληρωμένες λύσεις για την ανάπτυξη σύγχρονων και λειτουργικών εκπαιδευτικών πλατφορμών.

Κεφάλαιο 4ο: Υλοποίηση

4.1 Εισαγωγή

Στην ενότητα αυτή, θα επικεντρωθούμε στην πρακτική υλοποίηση της εκπαιδευτικής πλατφόρμας, αναλύοντας τα βασικά συστατικά και τη δομή της. Η υλοποίηση περιλαμβάνει τη δημιουργία και τη ρύθμιση των "Layout Parts" με το Themify, την κατηγοριοποίηση περιεχομένου, τη χρήση των Advanced Custom Fields (ACF) για τη διαχείριση δεδομένων, και τη δημιουργία shortcodes για την

εμφάνιση των πληροφοριών. Κάθε στοιχείο θα εξεταστεί λεπτομερώς, προκειμένου να κατανοηθεί ο τρόπος με τον οποίο συμβάλλει στη συνολική λειτουργικότητα και αποτελεσματικότητα της πλατφόρμας. Η δομή αυτή επιτρέπει την ευέλικτη διαχείριση των εκπαιδευτικών πόρων και βελτιώνει την εμπειρία των χρηστών, διευκολύνοντας την πλοήγηση και την πρόσβαση στο εκπαιδευτικό περιεχόμενο.

4.2 Δημιουργία Layout Parts (Themify)

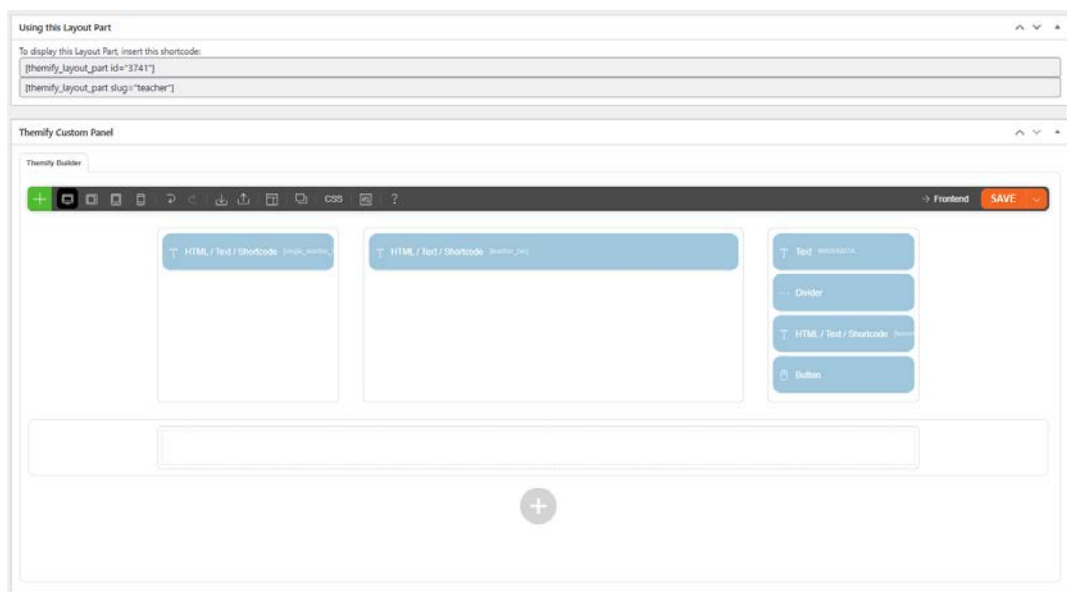
Τα Layout Parts αποτελούν ένα ισχυρό χαρακτηριστικό του Themify theme που επιτρέπει στους χρήστες να δημιουργούν και να αποθηκεύουν επαναχρησιμοποιήσιμες μονάδες περιεχομένου, διευκολύνοντας τη δημιουργία ομοιόμορφων και ευέλικτων layouts. Σκεφτείτε τα σαν "δομικά στοιχεία" που μπορείτε να συνδυάσετε για να χτίσετε σελίδες και να διατηρήσετε μια συνεπή εμφάνιση σε όλο το website σας.

4.2.1 Teacher Layout Part

Εδώ βλέπουμε ένα από τα δύο Layout Parts που έφτιαξα που αντιστοιχεί στην σελίδα των καθηγητών.

Στο αριστερό μέρος έχουμε ένα Custom Shortcode με το όνομα [single_teacher_header] το οποίο με την βοήθεια των ACF δείχνει ένα Header με το όνομα του καθηγητή και μία μικρή περιγραφή του καθηγητή.

Με παρόμοιο τρόπο έχουμε στο κέντρο [teacher_bio] όπου δείχνει μία μικρή παράγραφο για τον καθηγητή και στα δεξιά πάνω έχουμε ένα κείμενο που λέει ΜΑΘΗΜΑΤΑ έπειτα για λόγους μορφοποίησης ένα divider που μας δίνει το θέμα Themify όπου ακολουθεί άλλο ένα Custom Shortcode [lessons_teaching] που δείχνει τα μαθήματα που διδάσκει ο καθηγητής.



Εικόνα 1 Teacher Layout Part

4.2.2 Lesson Layout Part

Εδώ βλέπουμε το δεύτερο Layout Part που είναι για τις σελίδες των μαθημάτων.

Στο αριστερό μέρος έχουμε ένα custom Shortcode [teachers_display] για την εμφάνιση των καθηγητών που διδάσκουν το επιλεγμένο μάθημα.

Στο κέντρο έχουμε άλλα 4 Shortcodes όπου 1 από αυτά είναι από Plugin ACF ενώ τα άλλα 3 είναι custom shortcodes.

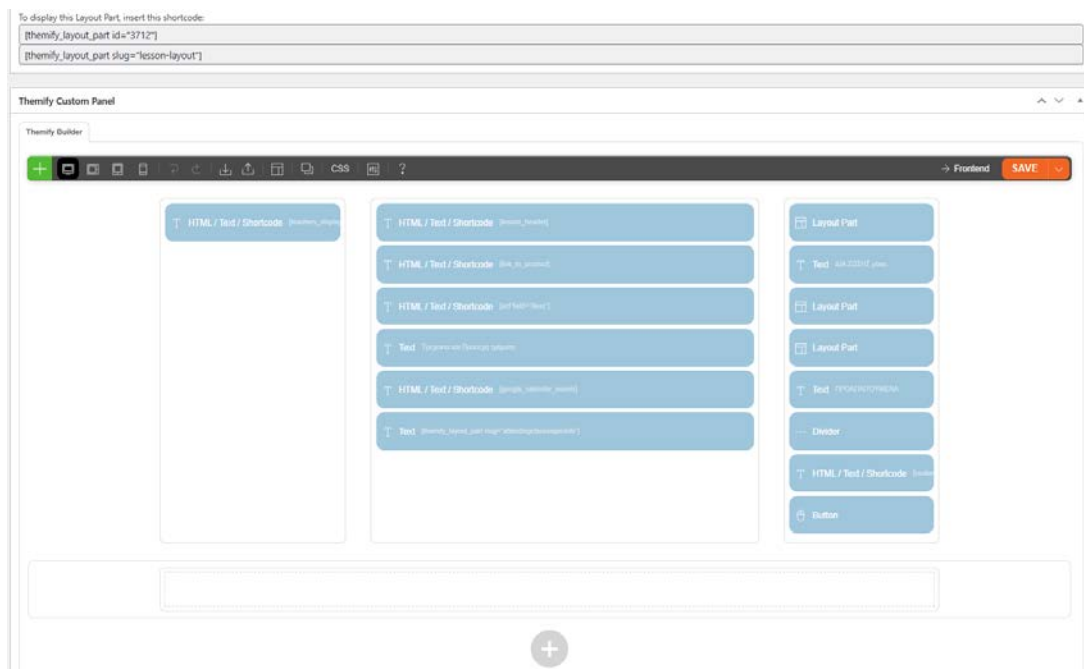
Το πρώτο είναι το [lesson_header] όπου δείχνει τον τίτλο του μαθήματος ποιος το διδάσκει και ένα κουμπί για την εμφάνιση του Google Calendar εφόσον αυτό έχει οριστεί

Το δεύτερο είναι custom shortcode για την εμφάνιση ενός κουμπιού που θα μας μεταφέρει στο αντικείμενο προς αγορά για το μάθημα εφόσον αυτό υπάρχει και έχει οριστεί από τον διαχειριστή

Το τρίτο είναι ένα shortcode που είναι ενσωματωμένο στο Plugin ACF όπου θα το αναλύσουμε αργότερα.

Και το Τέταρτο custom shortcode είναι [google_calendar_events] το οποίο κάνει χρήση του google calendar API ώστε να εμφανίσει με προσαρμοσμένο τρόπο τα event για το μάθημα

Στο δεξιά μέρος έχουμε το [custom_prerequisites] όπου δείχνει τα προαπαιτούμενα μαθήματα εφόσον αυτά υπάρχουν.

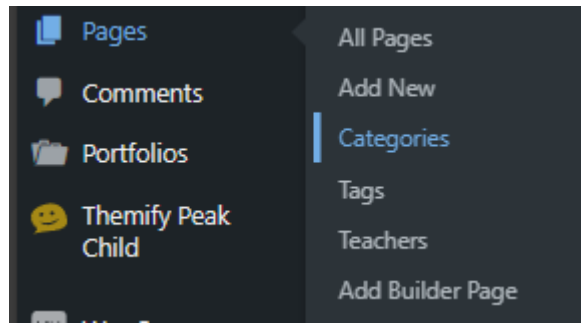


Εικόνα 2 Lesson Layout Part

4.3 Δημιουργία Κατηγορίας

Αρχικά θα εγκαταστήσουμε το Pages with category and tag plugin ώστε να μας δώσει την επιλογή να διαλέξουμε σε ποια κατηγορία θα ανήκουν οι σελίδες καθώς αυτό το επιτρέπει μόνο στα post το wordpress.

Έπειτα θα πάμε στην κατηγορία pages στο backend του wordpress και θα επιλέξουμε 'Categories'



Και θα δημιουργήσουμε 2 κατηγορίες Lessons και Teachers

<p>Add New Category</p> <p>Name <input type="text" value="Lessons"/></p> <p><small>The name is how it appears on your site.</small></p> <p>Slug <input type="text" value="lessons"/></p> <p><small>The "slug" is the URL-friendly version of the name. It is usually all lowercase and contains only letters, numbers, and hyphens.</small></p> <p>Parent Category <input type="text" value="None"/></p> <p><small>Categories, unlike tags, can have a hierarchy. You might have a Jazz category, and under that have children categories for Bebop and Big Band. Totally optional.</small></p> <p>Description <input type="text" value="This is the Category for lessons"/></p> <p><small>The description is not prominent by default; however, some themes may show it.</small></p> <p><input type="button" value="Add New Category"/></p>	<p>Add New Category</p> <p>Name <input type="text" value="Teacher"/></p> <p><small>The name is how it appears on your site.</small></p> <p>Slug <input type="text" value="teacher"/></p> <p><small>The "slug" is the URL-friendly version of the name. It is usually all lowercase and contains only letters, numbers, and hyphens.</small></p> <p>Parent Category <input type="text" value="None"/></p> <p><small>Categories, unlike tags, can have a hierarchy. You might have a Jazz category, and under that have children categories for Bebop and Big Band. Totally optional.</small></p> <p>Description <input type="text" value="This is the Category for the teacher"/></p> <p><small>The description is not prominent by default; however, some themes may show it.</small></p> <p><input type="button" value="Add New Category"/></p>
---	---

4.4 Χρήση Advanced Custom Fields

Δημιούργησα 2 Ομάδες από ACF μία για τις σελίδες των καθηγητών και μία για τις σελίδες μαθημάτων
Η κάθε ομάδα εμφανίζεται μόνο στις σελίδες που έχουν την κατηγορία Lesson και Teacher αντίστοιχα

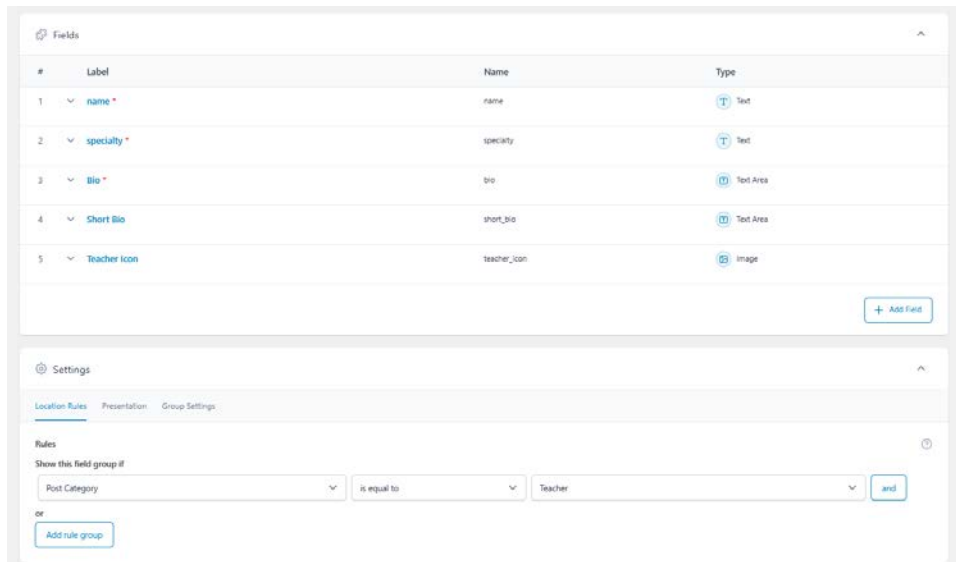
4.4.1 Teacher Group

Για την ομάδα των καθηγητών

Έχουμε 5 πεδία

- Name – το όνομα του καθηγητή που είναι τύπου ‘κείμενο’
- Specialty – ειδικότητα του καθηγητή που είναι τύπου ‘κείμενο’
- Bio – Μία παράγραφος/βιογραφικό του καθηγητή που είναι τύπου ‘Περιοχή κειμένου’
- Short Bio – Μια περίληψη του βιογραφικού που θα φαίνεται στα μαθήματα που θα διδάσκει που είναι τύπου ‘Περιοχή κειμένου’
- Teacher Icon – Εικόνα του καθηγητή που είναι τύπου ‘εικόνα’

Και από κάτω έχουμε επιλέξει να εμφανίζετε μόνο στις σελίδες που έχουν την κατηγορία Teacher που δημιουργήσαμε στο προηγούμενο βήμα (3.2)



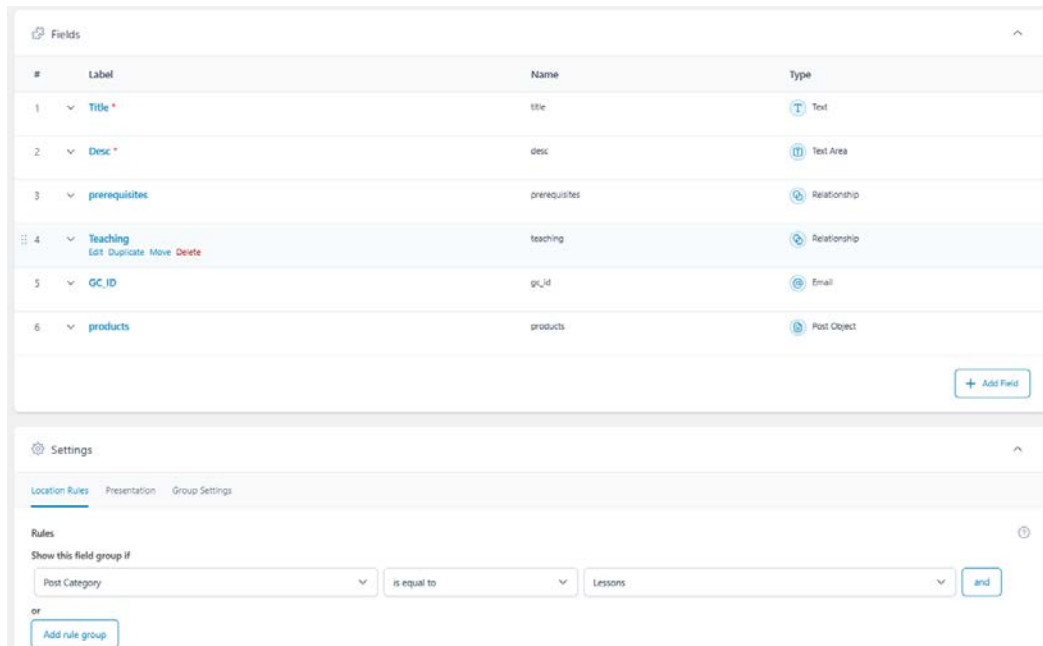
Εικόνα 3Teacher ACF Group

4.4.2 Lesson Group

Για την ομάδα μαθημάτων έχουμε 6 πεδία

- Title – Ο τίτλος του μαθήματος με τύπο ‘κείμενο’
- Desc – Η περιγραφή του μαθήματος με τύπο ‘Περιοχή κειμένου’
- Prerequisites – Τα προαπαιτούμενα μαθήματα που ίσως έχει το μάθημα με τύπο ‘relationship’ όπου μπορείς να επιλέξεις σελίδες που ανήκουν στην κατηγορία ‘Μαθήματα’
- Teaching – Οι καθηγητές που διδάσκουν το μάθημα με τύπο ‘relationship’ όπου μπορείς να επιλέξεις σελίδες που ανήκουν στην κατηγορία ‘Καθηγητής’
- GC_ID – Το google calendar ID για την εμφάνιση των event
- Products – Το/Τα αντικείμενο/α που θα αντιστοιχούν στο μάθημα τύπου ‘relationship’ όπου μπορείς να επιλέξεις τα αντικείμενα από το Plugin Woo-commerce (Το οποίο δεν θα το αναλύσουμε σε αυτήν την πτυχιακή)

Και από κάτω έχουμε επιλέξει να εμφανίζετε μόνο στις σελίδες που έχουν την κατηγορία Teacher που δημιουργήσαμε στο προηγούμενο βήμα (3.2)

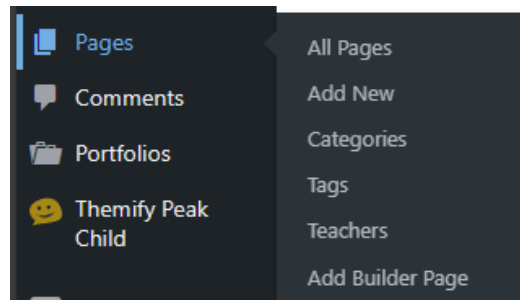


Εικόνα 1 Lesson ACF Group

4.5 Χρήση του ACF group

Αρχικά θα πρέπει να φτιάξουμε μία νέα Σελίδα στο wordpress όπου θα κάνουμε χρήση τα Custom Fields

Αφού συνδεθούμε στο backend και περιηγηθούμε στην καρτέλα σελίδες και πατήσουμε 'Add New'.



Θα μας μεταφέρει στην σελίδα δημιουργίας σελίδας του wordpress.

Στο δεξιό panel μπορούμε να βρούμε την καρτέλα Categories όπου και θα επιλέξουμε την κατηγορία Lessons που φτιάξαμε πριν



Με το που γίνει η επιλογή αυτόματα θα εμφανιστεί το custom Group που φτιάξαμε πριν για τα μαθήματα.

Title, Desc, prerequisites, Teaching, GC_ID, products

Συμπληρώνουμε όλα τα πεδία εκτός του gc_id, το οποίο θα αναλύσουμε αργότερα.

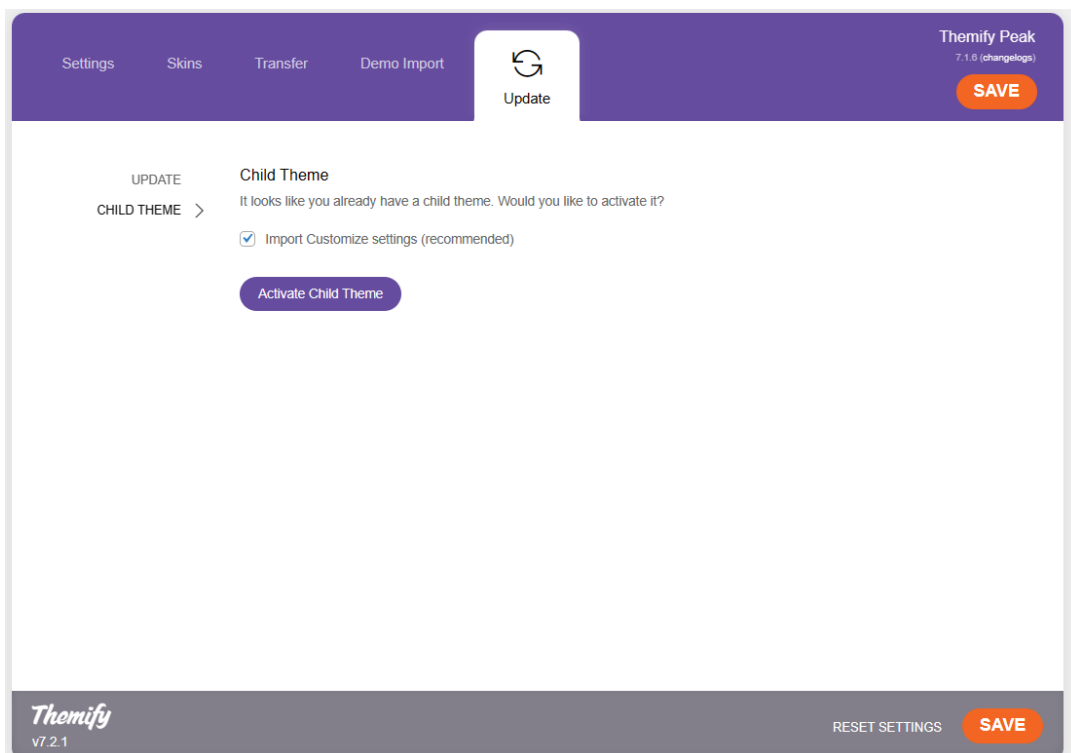
Κάνουμε το ίδιο και με τις σελίδες των μαθημάτων.

4.6 Κώδικας

Για να εμφανίσουμε τα ACF με δικό μας τρόπο και μορφοποίηση θα φτιάξουμε κάποια shortcodes για την προσαρμοσμένη εμφάνιση των πεδίων μας.

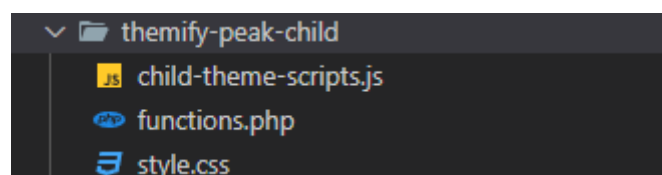
Το πρώτο και σημαντικότερο πράγμα που πρέπει να κάνουμε πριν γράψουμε κώδικα είναι να δημιουργήσουμε ένα Child Theme στο wordpress.

Μέσα από τις ρυθμίσεις του Themify theme μας δίνετε η επιλογή να ενεργοποιήσουμε το Child theme όπου θα δημιουργήσει ένα νέο θέμα και εκεί θα προσθέσουμε τον κώδικά μας που θα προσθέτει ή θα αντικαταστήσει υπάρχουσες λειτουργίες



Εικόνα 1 Themify Child Theme Settings

Το κουμπί Activate Child Theme θα δημιουργήσει ένα νέο Θέμα / υποφάκελο κάτω από τον φάκελο themes του wordpress και θα τον ενεργοποιήσει



Εικόνα 2 Themify Child theme folder

Εδώ βλέπουμε 3 αρχεία

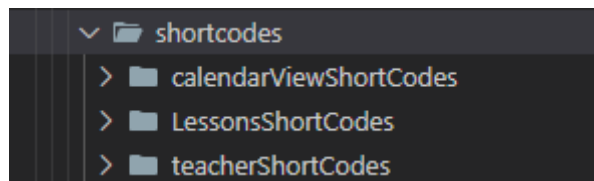
- child-theme-scripts.js – Είναι το αρχείο που θα γράφουμε τον JavaScript κώδικα που θα τρέχει στις σελίδες μας όπου χρειάζεται
- Functions.php – Είναι το αρχείο που θα γράφουμε τον php κώδικα μας για τα Shortcodes μας, είναι και το αρχείο που θα ασχοληθούμε περισσότερο
- style.css – Είναι το αρχείο μορφοποίησης που θα γράφουμε για να διαμορφώσουμε την εμφάνιση του site στα μέτρα μας

Εμείς θα δημιουργήσουμε έναν ακόμα φάκελο shortcodes για να μοιράσουμε τον κώδικα σε υποφακέλους και αρχεία για να βελτιώσουμε την δομή και την αναγνωσιμότητα του κώδικα



Έπειτα δημιουργούμε άλλους 3 φακέλους μέσα στο shortcode για να χωρίσουμε τα shortcodes για τους καθηγητές τα μαθήματα και τα Calendar event

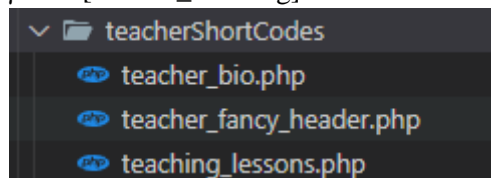
- calendarViewShortCodes
- LessonsShortCodes
- teacherShortCodes



4.7 Teacher Shortcodes

Θα ξεκινήσουμε με τα Shortcodes για τους καθηγητές. Θα δημιουργήσουμε 3 Αρχεία ένα για το κάθε Shortcode

- teacher_bio.php για το shortcode [teacher_bio]
- teacher_fancy_header.php για το [single_teacher_header]
- teaching_lessons.php για το [lessons_teaching]



Προτού προχωρήσουμε όμως πρέπει να εισάγουμε τα αρχεία στο functions.php του θέματος μας

```
//custom short codes for Teacher pages
include('shortcodes/teacherShortCodes/teacher_fancy_header.php');
include('shortcodes/teacherShortCodes/teacher_bio.php');
include('shortcodes/teacherShortCodes/teaching_lessons.php');
```

Όταν φορτώνει μία σελίδα εκτελείτε και το αρχείο function.php οπότε πρέπει να κάνουμε εισαγωγή τα εξωτερικά αρχεία μας ώστε να εκτελούνται και αυτά

Με την συνάρτηση include θα προσθέσουμε τα αρχεία μας από του άλλους φακέλους.

Στο πρώτο αρχείο teacher_bio.php θα δημιουργήσουμε την συνάρτηση που θα εμφανίζει την περιγραφή του καθηγητή

```
function bio()
{
    You, 1 second ago • Uncommitt
}

add_shortcode('teacher_bio', 'bio');
```

Με την συνάρτηση `add_shortcode()` του wordpress εγγράφουμε ένα shortcode που θα εμφανίζεται καλώντας `[lessons_bio]` και θα τρέχει την συνάρτηση `bio()`

Μέσα στην συνάρτηση `bio` θα πάρουμε με τις συναρτήσεις του ACF το πεδίο `bio` από το ACF group `Teacher` που είδαμε πριν.

```
function bio()
{
    $bio = get_field('bio');
}
```

Έπειτα θα γράψουμε τον κώδικα για την εμφάνιση του στην σελίδα

```
<p class="black_p">
    <?php echo $bio ?>
</p>
```

Το τελικό αρχείο θα πρέπει να μοιάζει κάπως έτσι

```
<?php
0 references
function bio()
{
    $bio = get_field('bio');

    ?>
    <p class="black_p">
        <?php echo $bio ?>
    </p>
    <?php
}

add_shortcode('teacher_bio', 'bio');
```

Και όπως βλέπουμε είναι ένα απλό html tag `p` και το αποτέλεσμα θα είναι κάπως έτσι

Teacher Full BIO

Με παρόμοιο τρόπο θα κάνουμε και τα υπόλοιπα shortcodes.

Στο δεύτερο αρχείο `teacher_fancy_header.php` θα προσθέσουμε το shortcode με όνομα `single_teacher_header` που θα τρέχει την συνάρτηση `fancy_header_shortcode`.

```
function fancy_header_shortcode()
{

}

add_shortcode('single_teacher_header', 'fancy_header_shortcode');
```

Σε αυτή την συνάρτηση θα χρειαστούμε τα πεδία name, specialty, icon, short_bio

```
$name = get_field('name');
$specialty = get_field('specialty');
$icon = get_field('teacher_icon');
$short_bio = get_field('short_bio');
```

Έπειτα θα βάλουμε το View το οποίο θα μοιάζει κάπως έτσι

```
<div class="custom-fancy-heading">
  <?php if ($icon != ""): ?>
    <div>
      <img src=<?php echo $icon ?> alt="Girl in a jacket" width="200" height="150">
    </div>
  <?php endif; ?>
  <h3 class="custom-fancy-h3 ">
    <span class="custom-fancy-main-head tf_block">
      <?php echo $name ?>
    </span>
    <span class="custom-fancy-sub-head tf_block">
      <?php echo $specialty ?>
    </span>
  </h3>
  <?php
  if ($short_bio != ""): ?>
    <div>
      <p class="teacher_desc">
        <?php echo $short_bio ?>
      </p>
    </div>
  <?php endif; ?>
</div>
```

Όπου πρώτα εμφανίζουμε την εικόνα του καθηγητή μετά το όνομα, ειδικότητα και έπειτα την σύντομη περιγραφή του με την βοήθεια της css θα έχουμε ως αποτέλεσμα αυτό:



Teacher Name

Teacher Specialty

Short Bio for the teacher

Και τέλος στο αρχείο `teaching_lessons.php` θα εμφανίζουμε τα μαθήματα που διδάσκει ο καθηγητής.

Με τον ίδιο τρόπο κάνουμε χρήση του `add_shortcode()`

```
function teachingLessons()  
{  
}  
  
add_shortcode('lessons_teaching', 'teachingLessons');
```

Εδώ θέλουμε να πάρουμε τα μαθήματα που διδάσκει ο καθηγητής, στο group το μαθημάτων είχαμε ένα πεδίο 'Teaching' όπου εκεί, όπως θα δούμε και παρακάτω, θα μας εμφανίζετε μία λίστα με όλους τους καθηγητές, δηλαδή οι σελίδες που ανήκουν στην κατηγορία 'Teacher'.

Για να πάρουμε τα μαθήματα που διδάσκει ο παρών καθηγητής θα πρέπει να κάνουμε ένα ερώτημα στην βάση του wordpress.

Πρώτα θα πάρουμε το id της σελίδας, το οποίο αντιστοιχεί και στον καθηγητή,

```
$current_page_id = get_the_ID();
```

Στην συνέχεια θα φτιάξουμε μία μεταβλητή για το ερώτημα

```

$args = array(
    'post_type' => 'page',
    'posts_per_page' => -1,
    'category_name' => 'lessons',
    'meta_query' => array(
        array(
            'key' => 'teaching',
            'value' => $current_page_id,
            'compare' => 'LIKE',
        ),
    ),
);

```

- 'post_type' => 'page': Καθορίζει τον τύπο του περιεχομένου που θέλουμε να ανακτήσουμε. Σε αυτήν την περίπτωση, στοχεύουμε σε "pages" (σελίδες).
- 'posts_per_page' => -1: Αναφέρει στο WordPress να επιστρέψει όλες τις σελίδες που ταιριάζουν με τα κριτήρια. Ο αριθμός -1 είναι ένας ειδικός τρόπος να πούμε στο WordPress να μη βάλει όριο.
- 'category_name' => 'lessons': Φιλτράρει τα αποτελέσματα ώστε να περιλαμβάνουν μόνο σελίδες που ανήκουν στην κατηγορία "lessons".
- 'meta_query' => array(...): Εδώ είναι που τα ερώτημα γίνεται πιο σύνθετο. Χρησιμοποιούμε ένα 'meta_query' για να φιλτράρουμε τις σελίδες βάσει προσαρμοσμένων πεδίων (custom fields). Συγκεκριμένα:
- 'key' => 'teaching': Αυτό δείχνει ότι ψάχνουμε για ένα custom field με όνομα κλειδιού "teaching".
- 'value' => \$current_page_id: Η τιμή που ψάχνουμε στο custom field είναι το \$current_page_id (πιθανώς να είναι το ID της τρέχουσας σελίδας).
- 'compare' => 'LIKE': Τέλος, το 'LIKE' σημαίνει ότι θέλουμε το περιεχόμενο του πεδίου "teaching" να περιλαμβάνει την τιμή που δόθηκε στο current_page_id

Με την βοήθεια της συνάρτησης WP_Query θα κάνουμε την κλήση στην βάση

```
$lesson_query = new WP_Query($args);
```

Θα βάλουμε έναν έλεγχο σε περίπτωση που κάτι πάει λάθος και έπειτα θα κάνουμε μία επανάληψη για να εμφανίσουμε τα μαθήματα.

```

if ($lesson_query) {
    echo '<ul>';
    while ($lesson_query->have_posts()) {
        $lesson_query->the_post();
        ?>
        <li class="teacher_desc">
            <?php echo '<a href="' . esc_url(get_permalink()) . '" target="_blank">' . esc_html(get_field('title')) . '</a>'; ?>
        </li>
        You, 4 months ago • teacher component
    <?php
    }
    wp_reset_postdata();
    echo '</ul>';
} else {
    echo 'lesson_query no';
}

```

- while (\$lesson_query->have_posts()): Αυτό ξεκινάει έναν loop "while" που θα εκτελεστεί όσο υπάρχουν ακόμη δημοσιεύσεις (posts) από τα αποτελέσματα του query.

- `$lesson_query->the_post()`: Μέσα στο loop, αυτή η γραμμή ρυθμίζει το τρέχον post στο επόμενο διαθέσιμο αποτέλεσμα του query. Είναι σημαντική για να έχουμε πρόσβαση στις σωστές πληροφορίες (πεδία ACF, περιεχόμενα κλπ.) κάθε μαθήματος στον loop.
- HTML με PHP: Αυτός ο κώδικας δημιουργεί ένα στοιχείο λίστας (``) για κάθε μάθημα. Μέσα, υπάρχει ένας σύνδεσμος που περιλαμβάνει:
- `esc_url(get_permalink())`: Κατεβάζει την καθαρή διεύθυνση URL (περνώντας την μέσα από μια συνάρτηση για ασφάλεια) του μαθήματος.
- `esc_html(get_field('title'))`: Κατεβάζει τον τίτλο του μαθήματος από ένα custom field ACF με το όνομα 'title'.

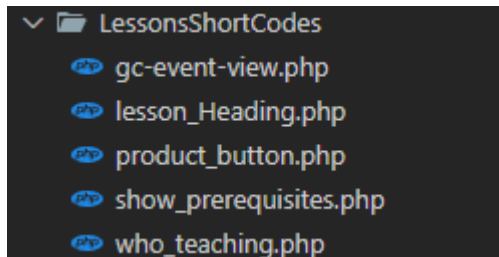
Το τελικό αποτέλεσμα εφόσον έχουμε επιλέξει κάποιο μάθημα θα είναι κάπως έτσι

ΜΑΘΗΜΑΤΑ

- [test lesson](#)

4.8 Lessons Shortcodes

Για τα μαθήματα θα δημιουργήσουμε 5 αρχεία μέσα στον φάκελο LessonsShortCodes



- `gc-event-view.php` – Ο κώδικας που θα παίρνει τα event από το Google Calendar API και θα τα εμφανίζει κάτω από την περιγραφή του μαθημάτος
- `lesson_Heading.php` – Ο τίτλος του μαθήματος ποιος το διδάσκει και link για το ημερολόγιο
- `product_button.php` – Ένα κουμπί που θα μας μεταφέρει στην σελίδα του επιλεγμένου προωίων
- `show_prerequisites.php` – Τα προ-απαιτούμενα μαθήματα αν υπάρχουν
- `who_teaching.php` – Οι καθηγητές που διδάσκουν το μάθημα

Αρχικά θα εισάγουμε τα παραπάνω αρχεία στο `function.php`

```
//custom short codes for Lessons pages
include('shortcodes/LessonsShortCodes/gc-event-view.php');
include('shortcodes/LessonsShortCodes/who_teaching.php');
include('shortcodes/LessonsShortCodes/show_prerequisites.php');
include('shortcodes/LessonsShortCodes/lesson_Heading.php');
include('shortcodes/LessonsShortCodes/product_button.php');
```

Στην συνέχεια θα ξεκινήσουμε με τον τίτλο του μαθήματος στο αρχείο `lesson_Heading.php` όπου για αρχή θα προσθέσουμε πάλι ένα shortcode με όνομα 'lesson_header' και θα εκτελεί την συνάρτηση 'lessonHeader' στην οποία θα πάρουμε τις τιμές από το custom πεδίο `teaching`, `title` και `gc_id`

```
$teachers = get_field('teaching');
$name = get_field('title');
$gc_id = get_field('gc_id');
```

Η μεταβλητή teachers είναι πίνακας οπότε θα κάνουμε μία επανάληψη για να φτιάξουμε το view με τους καθηγητές που διδάσκουν το μάθημα όπου και θα προσθέσουμε ένα link που θα μας κάνει ανακατευθύνση στην σελίδα του καθηγητή

```
$teacher_names = array();
foreach ($teachers as $teacher_id) {
    $teacher_names[] = '<a class="red_p" href="' . get_permalink($teacher_id) . '" target="_blank">' . get_field('name', $teacher_id) . '</a>';
}
```

Με την βοήθεια της συνάρτησης του wordpress get_permalink() και περνώντας το ID της σελίδας του καθηγητή μπορούμε να πάρουμε το link που θα μας κατευθύνει στην σελίδα του και με την βοήθεια της συνάρτησης get_field και με το όνομα του πεδίου και το ID της σελίδας μπορούμε να πάρουμε το όνομα του καθηγητή

Έπειτα θα εμφανίσουμε όλα τις τιμές

```
<div class="custom-fancy-heading">
  <h3 class="custom-fancy-h3 ">
    <span class="custom-fancy-main-head tf_block">
      <?php echo $name ?>
    </span>
    <span class="custom-fancy-sub-head tf_block">
      <?php if (is_array($teachers) && count($teachers) > 0) {
        ?>
        Διδάσκει:&nbsp;
        <?php
        echo implode('&nbsp;', $teacher_names);
      } ?>
    </span>
    <?php
    if ($gic_id !== '') {
      ?>
      <span class="custom-fancy-sub-head tf_block">
        <a href="http://s2c/calendar/?calendar_ID=<?php echo $gic_id; ?>" target="_blank">
          Ημερολόγιο
        </a>
      </span>
      <?php
    }
    ?>
  </h3>
</div>
```

Κάνοντας χρήση της συνάρτησης implode θα δημιουργήσουμε μία σειρά από a tags που θα ενώνονται με ','. Εφόσον υπάρχει και το calendarID θα προσθέσουμε και το κουμπί που θα μας ανακατευθύνει στην σελίδα του ημερολογίου και ως calendar_ID παράμετρο το αναγνωριστικό του ημερολογίου που θα δείξουμε στην συνέχεια πως θα το πάρουμε από την Google

Το τελικό αποτέλεσμα θα πρέπει να μοιάζει κάπως έτσι

test lesson

Διδάσκει: **Teacher Name**

Ημερολόγιο

Για το κουμπί του ημερολογίου θα δημιουργήσουμε ένα ακόμα page στο wordpress το με το όνομα calendar το οποίο θα αναλύσουμε αργότερα.

Στην αριστερή πλευρά της σελίδας θα πρέπει να φαίνονται και τα στοιχεία των καθηγητών οπότε το επόμενο αρχείο που θα δούμε είναι το who_teaching.php στο οποίο θα προσθέσουμε 'teachers_display' shortcode και θα εκτελεί την συνάρτηση show_Teachers όπου θα πάρουμε τους καθηγητές πάλι με την συνάρτηση get_field() θα πάρουμε το πεδίο teaching και έπειτα με μια επανάληψη για τον κάθε καθηγητή θα εμφανίσουμε τα πεδία των καθηγητών

```
foreach ($teachers as $teacher_id) {
    $name = get_field('name', $teacher_id);
    $specialty = get_field('specialty', $teacher_id);
    $url = get_permalink($teacher_id);
    $short_bio = get_field('short_bio', $teacher_id);
    $icon = get_field('icon', $teacher_id)

    ?>
    <div class="custom-fancy-heading">
        <a href=<?php echo $url ?>>
            <h3 class="custom-fancy-h3 ">
                <span class="custom-fancy-main-head tf_block">
                    <?php echo $name ?>
                </span>
                <?php if ($specialty !== ""): ?>
                    <span class="custom-fancy-sub-head tf_block">
                        <?php echo $specialty ?>
                    </span>
                <?php endif; ?>
            </h3>
        </a>
    </div>

    <?php if ($icon !== ""): ?>
        <div>
            <p class="teacher_desc">
                <?php echo $icon ?>
            </p>
        </div>
    <?php endif;
    if ($short_bio !== ""): ?>
        <div>
            <p class="teacher_desc">
                <?php echo $short_bio ?>
            </p>
        </div>
    <?php endif;
}
```

Με την βοήθεια της μορφοποίησης θα έχουμε αυτό το αποτέλεσμα

Teacher Name

Teacher Specialty

Short Bio for the teacher

test lesson

Διδάσκει: **Teacher Name**

Ημερολόγιο

Στην δεξιά πλευρά θα βάλουμε τα προ-απαιτούμενα που είναι τα μαθήματα που ίσως χρειάζεται να έχει παρακολουθήσει για να πάρει και αυτό το μάθημα.

Στο αρχείο show_prerequisites.php θα βάλουμε ακόμα ένα shortcode custom_prerequisites που θα τρέχει την συνάρτηση show_prerequisites() στην οποία παίρνουμε το πεδίο 'prerequisites', στην περίπτωση που δεν υπάρχει είναι κενή η λίστα θα εμφανίσουμε ένα κείμενο ότι το μάθημα δεν έχει προαπαιτούμενα, επειδή είναι σε 2 σημεία αυτό το κείμενο κάναμε μία συνάρτηση 'no_prerequisites' για να μην γράφουμε τον ίδιο κώδικα δύο φορές.

```
function no_prerequisites()
{
    ?>
    <div class="no_prerequisites">
        <span class="no_prerequisites_p">Απευθύνεται σε όλους </span>
    </div>
    <?php
}
```

Στην show_prerequisites συνάρτηση αφού έχουμε πάρει το πεδίο με τα προαπαιτούμενα με την βοήθεια της foreach θα φτιάξουμε έναν πίνακα με a html tags και με πάλι με την χρήση της implode θα ενώσουμε τον πίνακα ο κώδικας μας θα μοιάζει κάπως έτσι:

```
function show_prerequisites($atts)
{
    $lessons = get_field('prerequisites');
    if ($lessons == "") {
        no_prerequisites();
    } else {
        $filtered_lessons = array_diff($lessons, array(get_the_ID()));

        if (empty($filtered_lessons)) {
            no_prerequisites();
        } else {
            $lessons_names = array();
            if (is_array($filtered_lessons)) {
                foreach ($filtered_lessons as $lesson) {
                    if ($lesson != '') {
                        $lessons_names[] = '<a class="red_p" href="' . get_permalink($lesson) . '" target="_blank"> . ' . get_field('title', $lesson) . '</a>';
                    }
                }
            }

            ?>
            <div class="custom-fancy-heading">
                <?php
                echo implode('&nbsp;', $lessons_names);
                ?>
            </div>
            <?php
        }
    }
}
```

Στο αρχείο product_button.php θα φτιάξουμε ένα shortcode link_to_product που θα εκτελεί την συνάρτηση product_button η οποία αν υπάρχει επιλεγμένο κάποιο προϊόν θα εμφανίζει ένα κουμπί για να μας ανακατευθύνει στην σελίδα του προϊόντος. Η συνάρτηση θα μοιάζει κάπως έτσι

```

function product_button()
{
    if (get_field('products')) {
        $selected_products = get_field('products');
        $product_permalink = get_the_permalink($selected_products);
        $link_text = get_the_title($selected_products);
        ?>

        <a href=<?php echo $product_permalink ?> target="_blank">
            <div class="buy-redirect">
                <span class="buy-redirect-head tf_block">
                    Αγοράστε συνδρομή online ▷
                </span>
                <span class="buy-redirect-sub-head tf_block">
                    <?php echo $link_text ?>
                </span>
            </div>
        </a>
        <?php
    }
}

```

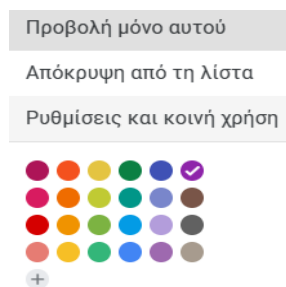
Τέλος ένα από τα μεγαλύτερα κομμάτια των custom shortcodes είναι η εμφάνιση του calendar

Για αρχή θα πρέπει μέσα από το google console να δημιουργήσουμε ένα api key για το google calendar το οποίο θα το κάνουμε χρήση για να παίρνουμε τα events από την Google.

Το api key θα το βάλουμε στο wp-config.php αρχείο του Wordpress ,

```
define('GOOGLE_API_KEY', 'google api key string');
```

Έπειτα θα δημιουργήσουμε ένα Calander και στις 'ρυθμίσεις και κοινή χρήση'



Θα επιλέξουμε το κουμπί 'Διαθέσιμο δημόσια'

Δικαιώματα πρόσβασης για συμβάντα

Διαθέσιμο δημόσια

Προβολή όλων των λεπτομερειών συμβάντων ▾

Έπειτα μπορούμε να πάρουμε το 'Αναγνωριστικό ημερολογίου' το οποίο μοιάζει κάπως έτσι longrandomstring@group.calendar.google.com και μπορούμε να το βάλουμε στο πεδίο gc_id στην σελίδα του μαθήματος.

Αρχικά η google κάνει χρήση του Recurrence Rule ή αλλιώς RRULE για τα event το οποίο είναι μία μορφή κειμένου που μας επιτρέπει να καθορίσουμε διάφορες πτυχές ενός επαναλαμβανόμενου συμβάντος, όπως:

- Frequency: Πόσο συχνά επαναλαμβάνεται το συμβάν (π.χ., DAILY, WEEKLY, MONTHLY, YEARLY)
- Interval: Ο αριθμός των φορών που η συχνότητα εμφανίζεται μεταξύ επαναλήψεων (π.χ., every 2 weeks)
- Weekday: Η/Οι συγκεκριμένες ημέρες της εβδομάδας κατά τις οποίες λαμβάνει χώρα το συμβάν (π.χ., MO, TU, WE, TH, FR)
- Monthday: Η/Οι συγκεκριμένες ημέρες του μήνα κατά τις οποίες λαμβάνει χώρα το συμβάν (π.χ., BYDAY=MO,2ND)
- Count: Ο συνολικός αριθμός εμφανίσεων για το συμβάν
- Until: Η ημερομηνία κατά την οποία η επανάληψη θα πρέπει να τερματιστεί

Οπότε πρέπει να κάνουμε εγκατάσταση μία βιβλιοθήκη για να μετατρέπουμε το κείμενο RRULE σε κάποιο αντικείμενο στην php.

Πρώτα θα πρέπει να εκτελέσουμε μία εντολή στον root φάκελο του wordpress

```
composer require rlanvin/php-rrule
```

Αυτό θα μας εγκαταστήσει την βιβλιοθήκη rlanvin/php-rrule που θα μας βοηθήσει με τα events.

Στο αρχείο index.php στον root φάκελο του wordpress θα βάλουμε αυτήν την γραμμή

```
require_once 'vendor/autoload.php';
```

Για να φορτώσουμε το 'vendor/autoload.php' το οποίο είναι απαραίτητο για να λειτουργήσουν οι βιβλιοθήκες που κάνουμε εγκατάσταση όπως και η βιβλιοθήκη 'rlanvin/php-rrule'

Έπειτα πάνω πάνω στο αρχείο θα βάλουμε πάνω πάνω στο αρχείο gc-event-view.php αυτήν την γραμμή

```
use RRule\RRule;
```

για να μπορέσουμε να κάνουμε χρήση την συνάρτηση 'RRULE()' από την βιβλιοθήκη κατευθείαν και να μην γράφουμε 'RRULE/RRULE()'

Έπειτα θα δημιουργήσουμε στο αρχείο gc-event-view.php το shortcode google_calendar_events που εκτελεί την συνάρτηση 'display_google_calendar_events'.

Μέσα στην συνάρτηση θα πάρουμε το api_key πεδίο 'gc_id' με την βοήθεια της συνάρτησης get_field()

```
//api key from wp-config
$api_key = GOOGLE_API_KEY;
//gets the calendar id to retrieve the events
$calendar_id = get_field('gc_id');
```

Θα δούμε αν η μεταβλητή calendar_id είναι κενή ώστε να μην συνεχίσει ο κώδικας. Στην συνέχεια θα πάρουμε την τωρινή ημερομηνία με την χρήση της συνάρτησης date('c') όπου το c είναι για να παράξουμε μία ημερομηνία σε μορφή ISO 8601 η αλλιώς RFC3339 το οποίο είναι το προτιμώμενο πρότυπο του Google Calendar API.

Μετά θε περάσουμε την ημερομηνία από την συνάρτηση urlencode() το οποίο θα κωδικοποιήσει το κείμενο / ημερομηνία για να μπορεί να χρησιμοποιηθεί με ασφάλεια ως μέρος ενός URL.

Θα φτιάξουμε το URL με βάση το calendar ID την ημερομηνία και το API key το οποίο θα μοιάζει κάπως έτσι:

```
$api_url = "https://www.googleapis.com/calendar/v3/calendars/{$calendar_id}/events?timeMin={$min_time}&key={$api_key}";
```

Και θα το περάσουμε στην συνάρτηση wp_remote_get() για να εκτελέσουμε ένα API ερώτημα στην google και θα ελέγξουμε αν υπάρχει κάποιο σφάλμα κατά την ανάκτηση των events.

```
$response = wp_remote_get($api_url);  
  
if (is_wp_error($response)) {  
    return "Error fetching events."  
}
```

Στην συνέχεια θα πάρουμε το σώμα (body) της απάντησης με την χρήση της wordpress συνάρτησης wp_remote_retrieve_body() και θα κάνουμε μετατρέψουμε το σώμα (body) σε php associative array με την χρήση της json_decode() και την δεύτερη παράμετρο true

```
$body = wp_remote_retrieve_body($response);  
$events_data = json_decode($body, true);
```

Έπειτα θα ελέγξουμε αν υπάρχουν τα items και αν δεν είναι κενός πίνακας θα τον περάσουμε σε μία άλλη συνάρτηση η οποία θα εμφανίσει και τα event μας.

```
if (isset($events_data['items'])) {  
    $eventList = $events_data['items'];  
    //function to display teh events!  
    if (count($eventList) == 0) {  
        echo 'Δεν υπάρχουν ακόμα events';  
        return;  
    }  
    return displayEvents($eventList);  
} else {  
    return 'No events found.';  
}
```

Θα δημιουργήσουμε την συνάρτηση displayEvents για να μας βοηθήσει να κάνουμε τον κώδικα συντηρήσιμο και ευανάγνωστο.

Πριν δούμε την συνάρτηση displayEvents ας δούμε πρώτα από τι μπορεί να αποτελείτε το κάθε στοιχείο του πίνακα eventList που είναι τα event που μας έδωσε η Google.

Κάποιες από ιδιότητες που μπορεί να έχει ένα event της google και θα κάνουμε χρήση εμείς είναι οι εξής:

- id (string): Ένα μοναδικό αναγνωριστικό για το συμβάν. Αυτό είναι ζωτικής σημασίας για την αναφορά, την ενημέρωση ή τη διαγραφή του συμβάντος συγκεκριμένα.

- **summary (string):** Ο τίτλος ή το κύριο θέμα του συμβάντος.
- **description (string):** Μια πιο λεπτομερής περιγραφή του συμβάντος. Αυτό μπορεί να περιέχει πρόσθετες σημειώσεις, ημερήσιες διατάξεις ή άλλες σχετικές πληροφορίες.
- **start (object):** Δείχνει πότε ξεκινά η εκδήλωση. Έχει δύο υπο-ιδιότητες:
- **dateTime (string):** Η ημερομηνία και η ώρα έναρξης της εκδήλωσης σε μορφή ISO 8601 (π.χ., "2023-12-25T09:30:00-05:00"). Αυτό ενσωματώνει τη ζώνη ώρας.
- **date (string):** Η ημερομηνία έναρξης της εκδήλωσης σε μορφή YYYY-MM-DD (π.χ., "2023-12-25"). Χρήσιμο ειδικά για εκδηλώσεις όλης της ημέρας.
- **end (object):** Παρόμοιο με το 'start', αυτό δείχνει πότε ολοκληρώνεται η εκδήλωση. Έχει επίσης δύο υπο-ιδιότητες:
- **dateTime (string):** Η ημερομηνία και η ώρα λήξης της εκδήλωσης σε μορφή ISO 8601.
- **date (string):** Η ημερομηνία λήξης της εκδήλωσης σε μορφή YYYY-MM-DD.
- **htmlLink (string):** Ένας άμεσος σύνδεσμος προς το συμβάν μέσα στη διαδικτυακή διεπαφή του Ημερολογίου Google. Αυτό είναι βολικό για τους χρήστες, ώστε να έχουν γρήγορη πρόσβαση στις λεπτομέρειες του συμβάντος.

Εφόσον πάρουμε τον τίτλο και την περιγραφή του event θα δημιουργήσουμε μία μεταβλητή για την ημερομηνία έναρξης του event καθώς είτε θα είναι όλη μέρα άρα θα υπάρχει η μεταβλητή date ή θα είναι συγκεκριμένη ώρα άρα θα υπάρχει η μεταβλητή dateTime. Θα φτιάξουμε και άλλες 3 μεταβλητές που θα μας βοηθήσουν για την μορφοποίηση του event καθώς η εμφάνιση θα είναι όλη προσαρμοσμένη.

- **startDate:** Θέλουμε να ξέρουμε πότε ξεκινάει το event σε περίπτωση επανάληψης ώστε να ξέρουμε υπολογίσουμε τον αριθμό των εβδομάδων.
- **every:** εκεί θα μπει η λέξη every/κάθε εφόσον το event επαναλαμβάνετε.
- **numberOfWeeks:** εκεί θα μπει ο αριθμός των εβδομάδων που θα γίνετε η επανάληψη.
- **time:** προσαρμοσμένη ώρα που θα φαίνεται στο event.

```
// get the title of the event if exists else put No title
$summary = isset($event['summary']) ? $event['summary'] : 'No title?';
//declare start Date for the weeks counter
$startDate = null;
$endDate = null;
// variable to add Every or not if its repeated event or not (Firday/Every Friday)
$every = '';
// what time the event takes place
$time = '';
```

Αφού έχουμε αρχικοποιήσει κάποιες από τις μεταβλητές

Έπειτα θα κάνουμε έναν έλεγχο για το αν το event λαμβάνει χώρα όλη την ημέρα ή συγκεκριμένες ώρες.

Στην περίπτωση που είναι για συγκεκριμένη ώρα θα βάλουμε startingDate πάρουμε τις τιμές dateTime από το start και end πεδίο του event και θα τις βάλουμε σε μία συνάρτηση την κάθε μία ώστε να μας φέρει πίσω την ώρα στα ελληνικά με την κατάλληλη μορφοποίηση.

Στην άλλη περίπτωση το event θα είναι όλη την ημέρα οπότε ως startingDate θα μπει το πεδίο date από το αντικείμενο start του event και στην μεταβλητή time θα βάλουμε το κείμενο '(Όλη μέρα)'.

```

if (isset($event['start']['dateTime'])) {
    //calculates the time based on event start/end time
    $startDate = $event['start']['dateTime'];
    $startDateTime = new DateTime($event['start']['dateTime']);
    $endDateTime = new DateTime($event['end']['dateTime']);
    //formatTimeInGreek function to return greek πμ/μμ
    $time = formatTimeInGreek($startDateTime) . ' - ' . formatTimeInGreek($endDateTime) . ' ';
    //ex time = 09:00PM - 11:00PM
} else {
    //else all day
    $startDate = $event['start']['date'];
    $time = '(Όλη μέρα) ';
}

```

Προτού πάμε παρακάτω πάμε να δούμε λίγο την συνάρτηση formatTimeInGreek.

Σε αυτή την συνάρτηση δεχόμαστε ως παράμετρο μία μεταβλητή της μορφής DateTime, όπου θα πάρουμε την ώρα τα λεπτά και την περίοδο (π.μ., μ.μ.) και θα επιστρέψουμε ένα κείμενο με την ώρα.

Πιο συγκριμένα θα κάνουμε χρήση της συνάρτησης format που έχει το DateTime και με τα γράμματα g, i, a, θα πάρουμε την ώρα τα λεπτά και την περίοδο αντίστοιχα.

- a - Πεζά για Ante meridiem και Post meridiem: Αυτό εμφανίζει το "π.μ." για ώρες πριν το μεσημέρι και το "μ.μ." για ώρες μετά το μεσημέρι.
- i - Λεπτά με μηδενικά μπροστά: Αυτό εμφανίζει τα λεπτά με δύο ψηφία, προσθέτοντας ένα μηδενικό μπροστά αν χρειάζεται.
- g - Μορφή 12 ωρών χωρίς μηδενικά μπροστά: Αυτό εμφανίζει την ώρα σε 12ωρη μορφή, που κυμαίνεται από 1 έως 12.

Αφού ελέγξουμε την περίοδο για να την μετατρέψουμε στα ελληνικά.

Έπειτα για λόγους μορφοποίησης θα ελέγξουμε αν τα λεπτά είναι 0 και θα εμφανίσουμε είτε την ώρα με την περίοδο είτε την ώρα με τα λεπτά και την περίοδο.

```

function formatTimeInGreek($time)
{
    $hour = $time->format('g');
    $minutes = $time->format('i');
    $period = $time->format('a');

    // Translate "am" and "pm" to Greek
    $period = ($period === 'am') ? 'π.μ.' : 'μ.μ.';

    if ($minutes === '00') {
        return "$hour$period";
    } else {
        return "$hour:$minutes$period";
    }
}

```

Έπειτα θα πάρουμε τις ημέρες που θα εκτελεστή το event

```
$daysOfWeek = getDaysOfWeek($event, $startDate);
```

Θα περάσουμε παραμετρικά το event και την ημέρα έναρξης.

Στην συνάρτηση `getDaysOfWeek` θα φτιάξουμε έναν πίνακα για να μετατρέψουμε τις μέρες από το google event σε ελληνικά.

```
$dayTranslations = array(
    'mo' => 'Δευτέρα',
    'tu' => 'Τρίτη',
    'we' => 'Τετάρτη',
    'th' => 'Πέμπτη',
    'fr' => 'Παρασκευή',
    'sa' => 'Σάββατο',
    'su' => 'Κυριακή'
);
```

Θα αgioποιήσουμε 2 πίνακες ένας για τις μέρες και ένα για την μεταγλωττισμένες.

```
$days = array();
$translatedDaysOfWeek = array();
```

Αν το event είναι επαναλαμβανόμενο θα πάρουμε το κείμενο `rrule` και με την βοήθεια του `regex` θα πάρουμε το πεδίο `BYDAY` που περιέχει τα πρώτα 2 γράμματα της κάθε ημέρες του event, αλλιώς θα πάρουμε από το `startingDate` την ημέρα με την βοήθεια του `format('l')` και του `substr` για να πάρουμε τα πρώτα 2 γράμματα ώστε να μπορούμε να κάνουμε χρήση τον ίδιο πίνακα από πριν.

```
if (!empty($event['recurrence'])) {
    preg_match('/BYDAY=([A-Za-z,]+)/', $event['recurrence'][0], $matches);
    $days = array_merge($days, explode(',', $matches[1]));
} else {
    $startDate = new DateTime($startDate);
    $days[] = substr($startDate->format('l'), 0, 2);
}
```

Θα εκτελέσουμε μια επανάληψη σε όσες μέρες βρούμε και αφού μετατρέψουμε τις μέρες σε πεζά θα ελέγξουμε αν υπάρχουν στον πίνακα και θα επιστρέψουμε τα ελληνικά.

```
foreach ($days as $day) {
    $day = strtolower($day);
    $translatedDay = isset($dayTranslations[$day]) ? $dayTranslations[$day] : $day;
    $translatedDaysOfWeek[] = $translatedDay;
}
```

Και τέλος θα ενώσουμε τον πίνακα με κόμμα πριν τον επιστρέψουμε

```
return implode(', ', $translatedDaysOfWeek);
```

Στην συνέχεια της `displayEvent` συνάρτησης θα ελέγξουμε αν το event είναι επαναλαμβανόμενο και θα πάρουμε την πρώτη τιμή του πίνακα καθώς θα έχουμε ένα μόνο επαναληπτικό κανόνα για λόγους διευκόλυνσης και επειδή τα events αντιστοιχούν σε τμήματα από το κάθε μάθημα θα έχει έναν επαναληπτικό κανόνα.

Σε περίπτωση που υπάρχει το πεδίο `recurrence` σημαίνει ότι έχουμε επαναλαμβανόμενο event άρα θα εκτελέσουμε μία συνάρτηση `formatEventDateRepeat()` αλλιώς θα εκτελέσουμε την συνάρτηση `formatEventDateSingle()` και οι δύο θα μας επιστρέψουν ένα πίνακα με τις μεταβλητές:

- numOfWeeks: πόσες εβδομάδες θα εκτελείτε το event.
- seasonWithYear: την εποχή έναρξης με τον χρόνο (π.χ. Άνοιξη 2024).
- formattedDate: αριθμός ημέρας και μήνας.

```

if (isset($event['recurrence'])) {
    $rruleString = $event['recurrence'][0];
    $every = "Κάθε ";
    $fullEventDate = formatEventDateRepeat($rruleString, $startDate);
} else {
    $fullEventDate = formatEventDateSingle($startDate);
}

```

Στην συνάρτηση formatEventDateRepeat() θα δώσουμε ως παραμέτρους το rrule κείμενο και την αρχική μέρα των events. Θα φτιάξουμε ένα αντικείμενο τύπου date και θα το κάνουμε format με την παράμετρο 'F' η οποία θα μας φέρει τον μήνα έπειτα θα εκτελέσουμε μία συνάρτηση που θα πάρει ως παράμετρο τον μήνα και θα επιστρέψει την εποχή και θα το βάλουμε σε μια μεταβλητή μαζί με τον χρόνο τον οποίο θα τον πάρουμε με το format('Y').

Στην συνέχεια αν το event επαναλαμβάνετε για πάντα θα πάρουμε την αρχική ημέρα και από το αντικείμενο date και θα την κάνουμε format με την παράμετρο 'j F' το οποίο θα μας μετατρέψει το αντικείμενο σε μορφή κειμένου με τον αριθμό της ημέρας και τον μήνα (π.χ. 24 May) και θα επιστρέψουμε ένα πίνακα με το formattedDate όπου είναι η ημερομηνία με μέρα και μήνα το seasonWithYear που είναι η εποχή και ο χρόνος (π.χ. Άνοιξη 2025) και τέλος την μεταβλητή numOfWeeks η οποία θα είναι κενή.

```

$dateObject = new DateTime($startDate);
$seasonWithYear = getSeason($dateObject->format('F')) . ' ' . $dateObject->format('Y');

$rrule = new RRule($rruleString, $startDate);
if ($rrule->isInfinite()) {
    $formattedDate = $dateObject->format('j F');
    return (object) array(
        'formattedDate' => $formattedDate,
        'seasonWithYear' => $seasonWithYear,
        'numOfWeeks' => ''
    );
}

```

Σε περίπτωση που το event θα εκτελείτε για συγκεκριμένες μέρες θα πάρουμε τις επαναλήψεις με την συνάρτηση getOccurrences και θα πάρουμε το πρώτο και τελευταίο event και θα υπολογίσουμε την διαφορά ημερών για να δούμε πόσες εβδομάδες θα γίνεται η επανάληψη έπειτα θα διαιρέσουμε με το επτά καθώς η εβδομάδα έχει επτά μέρες και επειδή θα κάνουμε χρήση του floor θα προσθέσουμε μία εβδομάδα. Και μετά αν το συμβάν επαναλαμβάνετε για πάνω από μία εβδομάδα θα βάλουμε την λέξη 'Εβδομάδες' αλλιώς την λέξη 'Εβδομάδα' και θα τα βάλουμε όλα μαζί στην μεταβλητή numOfWeeks. Έπειτα από το πρώτο και το τελευταίο event θα πάρουμε την μέρα και τον μήνα με το format('j F') και θα το βάλουμε στην μεταβλητή formattedDate για να το επιστρέψουμε μαζί με το numOfWeeks και seasonWithYear

```

$occurrences = $rrule->getOccurrences();
$startWeek = $occurrences[0];
$endWeek = end($occurrences);
$diffInDays = $endWeek->diff($startWeek)->days;
$weeks = $diffInWeeks = floor($diffInDays / 7) + 1;
$suffixes = "Εβδομάδες";
if ($weeks == 1)
    $suffixes = "Εβδομάδα";
$numOfWeeks = "( $weeks $suffixes )";

$formattedStartDate = $startWeek->format('j F');
$formattedEndDate = $endWeek->format('j F');
$formattedDate = "$formattedStartDate - $formattedEndDate";

return (object) array(
    'formattedDate' => $formattedDate,
    'seasonWithYear' => $seasonWithYear,
    'numOfweeks' => $numOfWeeks
);

```

Στην περίπτωση που το event εκτελείτε μία φορά στην συνάρτηση formatEventDateSingle θα περάσουμε παραμετρικά την μέρα έναρξης και θα πάρουμε την εποχή με τον ίδιο τρόπο όπως πριν απλά στην ημερομηνία με την μέρα και τον μήνα θα πάρουμε μόνο από την μέρα έναρξης και στο numOfWeeks θα βάλουμε το κείμενο 'Μια φορά'

```

function formatEventDateSingle($startDate)
{
    $dateObject = new DateTime($startDate);
    $seasonWithYear = getSeason($dateObject->format('F')) . ' ' . $dateObject->format('Y');
    $formattedDate = $dateObject->format('j F');
    return (object) array(
        'formattedDate' => $formattedDate,
        'seasonWithYear' => $seasonWithYear,
        'numOfweeks' => 'Μία φορά'
    );
}

```

Η συνάρτηση getSeason όπως αναφέραμε και παραπάνω δέχεται τον μήνα και επιστρέφει την εποχή, αυτό επιτυγχάνετε με την βοήθεια της switch case και του strtolower ώστε να έχουμε πάντα το κείμενο σε πεζά γράμματα.

```

function getSeason($startMonth)
{
    $startMonth = strtolower($startMonth);

    switch ($startMonth) {
        case 'december':
        case 'january':
        case 'february':
            $season = 'Χειμώνας';
            break;
        case 'march':
        case 'april':
        case 'may':
            $season = 'Άνοιξη';
            break;
        case 'june':
        case 'july':
        case 'august':
            $season = 'Καλοκαίρι';
            break;
        case 'september':
        case 'october':
        case 'november':
            $season = 'Φθινόπωρο';
            break;
        default:
            $season = '';
    }

    return "$season";
}

```

Τέλος εφόσον έχουμε όλα τα δεδομένα μας μπορούμε να εμφανίσουμε το event με τον δικό μας προσαρμοσμένο τρόπο όπου θα κάνουμε χρήση το πεδίο htmlLink ώστε όταν κάνουμε κλικ το event θα μας μεταφέρει στην σελίδα του google calendar όπου θα μπορούμε να δούμε παραπάνω πληροφορίες για το event και θα βάλουμε τον τίτλο (summary) numOfWeeks , seasonWithYear , την μεταβλητή Every η οποία είναι η κενή η θα έχει την λέξη Κάθε , την μεταβλητή daysOfWeek , την time και τέλος formattedDate και όλα αυτά με σωστή χρήση των html tag θα μοιάζει κάπως έτσι

```

echo '
<a class="custom-event" href=' . $event['htmlLink'] . '>
  <div class="tb_text_wrap">
    <p style="text-align: left">
      <span style="color: #ff0000">
        ' . $summary . ' ' . $fullEventDate->numOfWeeks . ' - ' . $fullEventDate->seasonWithYear . ' <br />
      </span>
      <span style="color: #000000">
        ' . $every . ' ' . $daysOfWeek . ' ' . $time . $fullEventDate->formattedDate . ' <br />
      </span>
    </p>
  </div>
</a>';

```

Και με την σωστή μορφοποίηση θα έχουμε αυτό το αποτέλεσμα θα είναι κάπως έτσι

Κάθε Δευτέρα event (10 Εβδομάδες) - Άνοιξη 2024

Κάθε Δευτέρα 11π.μ. - 12μ.μ. 29 April - 1 July

Κάθε Δευτέρα και Τρίτη (5 Εβδομάδες) - Άνοιξη 2024

Κάθε Δευτέρα 10μ.μ. - 11μ.μ. 20 May - 18 June

Μια μέρα όλη μέρα Μία φορά - Άνοιξη 2024

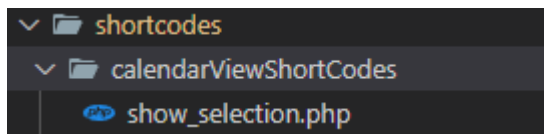
Τετάρτη (Όλη μέρα) 22 May

Μία μέρα Μία φορά - Άνοιξη 2024

Τετάρτη 7:30μ.μ. - 8:30μ.μ. 22 May

Για την σελίδα του ημερολογίου (Calendar) θα βάλουμε ένα custom shortcode για την εμφάνιση τοθ μενού με τα μαθήματα για επιλογή και ένα custom html κώδικα για την εμφάνιση του ενσωματωμένου ημερολογίου. Αυτό επιτρέπει στους επισκέπτες να βλέπουν εύκολα και απευθείας στον ιστότοπο το ημερολόγιο όλων των μαθημάτων.

Μέσα στον φάκελο shortcodes θα προσθέσουμε έναν ακόμα υπο-φάκελο με όνομα calendarViewShortcodes και εκεί θα δημιουργήσουμε ένα αρχείο με το όνομα show_selection.php.



Στο αρχείο show_selection.php θα προσθέσουμε ένα shortcode με το όνομα category_list και θα εκτελεί την συνάρτηση listView

```
function listView()  
{  
  
}  
  
add_shortcode('category_list', 'listView');
```

στην οποία θα πρέπει πρώτα να πάρουμε όλα τα μαθήματα που δεν έχουν κενό το gc_id από το ACF με ένα ερώτημα στην βάση όπου θα δώσουμε ως παραμέτρους

- post_type => pages – για να πάρουμε τις σελίδες.
- category_name => lessons – για να φιλτράρουμε και να κρατήσουμε μόνο τις σελίδες που έχουν κατηγορία lessons
- posts_per_pages => -1 – για να πάρουμε όλες τις σελίδες χωρίς περιορισμούς
- meta_query => array() – ένας πίνακας με την έξτρα μεταβλητή που θέλουμε να φιλτράρουμε
 - key => gc_id – το custom field που θέλουμε και να μην είναι κενό
 - compare => != – το σύμβολο του διάφορου
 - value => ‘’ – και η τιμή που δεν θέλουμε να έχει το gc_id

```

$args = array(
    'post_type' => 'page',
    'category_name' => 'lessons',
    'posts_per_page' => -1,
    'meta_query' => array(
        array(
            'key' => 'gc_id',
            'compare' => '!=',
            'value' => '',
        )
    )
);

```

Έπειτα θα εκτελέσουμε την συνάρτηση 'new WP_Query(\$args);' για να πάρουμε τις σελίδες και θα πάρουμε και από τις παραμέτρους του url το calendarID εφόσον υπάρχει.

```

$pages = new WP_Query($args);
$calendarIDParams = isset($_GET['calendar_ID']) ? sanitize_text_field($_GET['calendar_ID']) : '';

```

Στην συνέχεια θα βάλουμε σε μία επανάληψη της σελίδες που βρήκαμε πρώτα θα δούμε αν υπάρχουν σελίδες στην μεταβλητή pages για να εμφανίσουμε ένα παράθυρο στο οποίο θα υπάρχει ένα πεδίο για την αναζήτηση και για όσο έχουν post δηλαδή είναι τύπου post/σελίδα για το wordpress θα εμφανίσουμε ένα κουμπί για κάθε μάθημα που θα έχει το όνομα του το οποίο θα πάρουμε με την συνάρτηση get_field('title'); που θα μας δώσει το πεδίο title από το ACF και get_field('gc_id'); για να πάρουμε το αναγνωριστικό του ημερολογίου.

```

if ($pages->have_posts()) {
    $output = '<div class="page-checkbox-container">';
    $output .= '<input type="text" id="page-filter" placeholder="Filter pages">';
    $output .= '<form id="page-checkbox-form">';
    while ($pages->have_posts()) {
        $pages->the_post();
        $calendarID = get_field('gc_id');
        $name = get_field('title');
        $output .= '<div class="page-checkbox-label-container">';
        $output .= '<input id=".' . $calendarID . '" class="page-checkbox" ' . ($calendarID === $calendarIDParams ? 'checked="checked"' : '') . ' name=".' . $name . '" type="checkbox" value=".' . $calendarID . '">';
        $output .= '<label for=".' . $calendarID . '" class="page-checkbox-label"> . $name . '</label>';
        $output .= '</div>';
    }
    $output .= '</form>';
    $output .= '</div>';

    return $output;
} else {
    return 'No pages found in the specified category.';
}

```

Στην σελίδα Calendar θα βάλουμε στο αριστερό της μέρος αυτό το shortcode ενώ στο δεξιά μέρος θα βάλουμε κάποιο javascript κώδικα για την εμφάνιση του ενσωματωμένου ημερολογίου.

Θα βάλουμε ένα div με το id calendar-container και θα προσθέσουμε ένα script που θα παίρνει αν υπάρχει από τα params το calendarID και θα βάζει το embedded κώδικα που δίνει η google για το ημερολόγιο με παράμετρο src το calendarID.

```

function getUrlParameter(name) {
    const urlParams = new URLSearchParams(window.location.search);
    return urlParams.get(name);
}
const id = getUrlParameter('calendar_ID');
let container = document.getElementById('calendar-container');
let child;
if (id !== null){
    child = document.createElement('iframe');
    child.id = 'calendar-id';
    child.width = '800';
    child.height = '600';
    child.frameBorder = '0';
    child.scrolling = 'no';
    child.src = 'https://calendar.google.com/calendar/embed?src=' + id;
}else{
    child = document.createElement('span');
    child.textContent = 'No param?';
}
container.appendChild(child);

```

Στο αρχείο javascript του child θέματος που φτιάξαμε πριν θα βάλουμε κάποια event listeners, ένα για όταν αλλάζει το search input και ένα για όταν γίνεται αλλαγή στα checkboxes.

Στο πρώτο θα πάρουμε το στοιχείο με αναγνωριστικό 'page-filter' και τα checkboxes με το classname 'page-checkbox' και όταν αλλάζει το input θα φιλτράρουμε τα checkboxes απλά αλλάζοντας το style τους από flex σε none ή το ανάποδο ανάλογα με το κείμενο που εισάγει ο χρήστης.

```

document.addEventListener('DOMContentLoaded', function () {
    var pageFilterInput = document.getElementById('page-filter');
    var pageCheckboxes = document.querySelectorAll('.page-checkbox');

    pageFilterInput.addEventListener('input', function () {
        var filterText = pageFilterInput.value.toLowerCase();
        pageCheckboxes.forEach(function (checkbox) {
            console.log(checkbox);
            var labelText = checkbox.name.toLowerCase();
            if (labelText.includes(filterText)) {
                checkbox.parentNode.style.display = 'flex';
            } else {
                checkbox.parentNode.style.display = 'none';
            }
        });
    });
});

```

Στο δεύτερο listener θα βάλουμε στην περίπτωση της επιλογής κάποιου html tag από το form που δημιουργήσαμε πριν να αλλάζει το embedded κώδικα της google με τα ανανεωμένα αναγνωριστικά ημερολογίου.

```

document.addEventListener('DOMContentLoaded', function () {
  var form = document.getElementById('page-checkbox-form');
  var src = 'https://calendar.google.com/calendar/embed?src='
  let child;

  form.addEventListener('change', function (event) {
    var checkedCheckboxes = document.querySelectorAll('.page-checkbox:checked');
    var calendarFrame = document.getElementById("calendar-container");
    var selectedValues = [];

    checkedCheckboxes.forEach(function (checkbox) {
      selectedValues.push(checkbox.value);
    });

    if (selectedValues.length > 0) {
      console.log(selectedValues);
      var newURL = src + selectedValues.join("&src=");
      child = document.createElement('iframe');
      child.id = 'calendar-id';
      child.width = '800';
      child.height = '600';
      child.src = newURL;
    } else {
      child = document.createElement('span');
      child.className = "no-gc-id";
      child.textContent = 'Please Select a lesson!!';
    }
    calendarFrame.innerHTML = '';
    calendarFrame.appendChild(child);
  });
});

```

4.9 Επίλογος

Στο κεφάλαιο αυτό, παρουσιάστηκε αναλυτικά η υλοποίηση της εκπαιδευτικής πλατφόρμας, επικεντρώνοντας στις βασικές συνιστώσες και τη δομή της. Η διαδικασία περιελάμβανε τη δημιουργία και ρύθμιση των "Layout Parts" μέσω του Themify, την κατηγοριοποίηση του περιεχομένου, τη χρήση των Advanced Custom Fields (ACF) για τη διαχείριση δεδομένων, και την ανάπτυξη των shortcodes για την προβολή των πληροφοριών. Κάθε ένα από αυτά τα στοιχεία εξετάστηκε λεπτομερώς, προσφέροντας μια ολοκληρωμένη εικόνα του τρόπου με τον οποίο συμβάλλουν στη συνολική λειτουργικότητα και αποτελεσματικότητα της πλατφόρμας.

Η προτεινόμενη δομή της πλατφόρμας επιτρέπει ευέλικτη διαχείριση των εκπαιδευτικών πόρων, βελτιώνοντας την εμπειρία των χρηστών και διευκολύνοντας την πλοήγηση και την πρόσβαση στο εκπαιδευτικό περιεχόμενο. Τα εργαλεία και οι τεχνικές που χρησιμοποιήθηκαν προσφέρουν τη δυνατότητα εύκολης προσαρμογής και επέκτασης του συστήματος, καθιστώντας το ένα χρήσιμο εργαλείο για την εκπαιδευτική διαδικασία.

Με την ολοκλήρωση της υλοποίησης, καθίσταται εμφανές ότι η χρήση σύγχρονων τεχνολογιών και πρακτικών, όπως το WordPress, το ACF, και τα shortcodes, μπορεί να οδηγήσει σε σημαντικές βελτιώσεις στη διαχείριση και παρουσίαση του εκπαιδευτικού υλικού, συμβάλλοντας στην επίτευξη των εκπαιδευτικών στόχων της πλατφόρμας.

Κεφάλαιο 5ο: Συμπεράσματα

5.1 Περίληψη

Κατά τη διάρκεια της πτυχιακής εργασίας, επικεντρωθήκαμε στην αξιοποίηση της ευελιξίας και της προσαρμοστικότητας που προσφέρουν το WordPress και το ACF για την κατασκευή μιας ολοκληρωμένης εκπαιδευτικής πλατφόρμας. Η πλατφόρμα σχεδιάστηκε για να διευκολύνει την οργάνωση και την παρουσίαση του εκπαιδευτικού περιεχομένου, δίνοντας ιδιαίτερη προσοχή στα εξής στοιχεία:

1. Η χρήση του Themify επέτρεψε τη δημιουργία προσαρμοσμένων μερών διάταξης, βελτιώνοντας την οπτική ελκυστικότητα και τη χρηστικότητα της πλατφόρμας.
2. Κατηγοριοποίηση Περιεχομένου: Τα Advanced Custom Fields χρησιμοποιήθηκαν για τη δομή και τη διαχείριση διαφόρων τύπων περιεχομένου, όπως μαθήματα και πληροφορίες διδασκόντων. Αυτή η κατηγοριοποίηση επέτρεψε την αποδοτική διαχείριση του περιεχομένου και τη βελτίωση της πλοήγησης των χρηστών.
3. Δυναμική Εμφάνιση Περιεχομένου: Τα shortcodes έπαιξαν σημαντικό ρόλο στην δυναμική εμφάνιση περιεχομένου σε διαφορετικές σελίδες. Αυτή η προσέγγιση διευκόλυνε τη διαδικασία ενημέρωσης και εξασφάλισε τη συνέπεια σε όλη την πλατφόρμα.

5.2 Κόρια Επιτεύγματα

Η φάση υλοποίησης αυτού του έργου ανέδειξε αρκετά σημαντικά επιτεύγματα:

1. Βελτιωμένη Χρηστικότητα: Ο σχεδιασμός της πλατφόρμας επικεντρώθηκε στην εμπειρία του χρήστη, εξασφαλίζοντας ότι τόσο οι μαθητές όσο και οι εκπαιδευτικοί μπορούσαν εύκολα να πλοηγηθούν και να έχουν πρόσβαση στις απαραίτητες πληροφορίες.
2. Ευελιξία και Επεκτασιμότητα: Ο συνδυασμός του WordPress, του ACF και των shortcodes παρείχε μια ευέλικτη και επεκτάσιμη λύση, επιτρέποντας μελλοντικές επεκτάσεις και τροποποιήσεις χωρίς μεγάλες ανατροπές.
3. Αποδοτική Διαχείριση Περιεχομένου: Η δομημένη προσέγγιση στην κατηγοριοποίηση περιεχομένου και η δυναμική σύνδεση σχετικού περιεχομένου βελτίωσαν σημαντικά την αποδοτικότητα της διαχείρισης και της παράδοσης του περιεχομένου.

5.3 Προκλήσεις και Λύσεις

Κατά τη διαδικασία υλοποίησης, αντιμετωπίστηκαν διάφορες προκλήσεις, οι οποίες επιλύθηκαν μέσω διαφόρων λύσεων:

1. Οργάνωση Περιεχομένου: Η διασφάλιση ότι το περιεχόμενο ήταν καλά οργανωμένο και εύκολα προσβάσιμο απαιτούσε προσεκτικό σχεδιασμό και στρατηγική χρήση του ACF για τη δημιουργία σχετικών πεδίων και σχέσεων.
2. Προσαρμογή και Συμβατότητα: Η προσαρμογή της πλατφόρμας διατηρώντας παράλληλα τη συμβατότητα με διάφορες συσκευές και προγράμματα περιήγησης ήταν ένα περίπλοκο έργο. Η χρήση του Themify και οι δοκιμές σε πολλαπλά περιβάλλοντα βοήθησαν στην επίλυση αυτών των ζητημάτων.
3. Δυναμικές Ενημερώσεις: Η διατήρηση του περιεχομένου ενημερωμένου δυναμικά απαιτούσε την αποτελεσματική χρήση των shortcodes και των πεδίων ACF, που διευκόλυναν τις ενημερώσεις σε πραγματικό χρόνο χωρίς χειροκίνητες παρεμβάσεις.

5.4 Μελλοντικές Κατευθύνσεις

Βάσει των γνώσεων που αποκομίστηκαν από αυτή τη διπλωματική εργασία, προτείνονται διάφορες μελλοντικές κατευθύνσεις:

1. Περαιτέρω Προσαρμογή: Η επέκταση των επιλογών προσαρμογής για να συμπεριλάβουν περισσότερες διαδραστικές δυνατότητες και εξατομικευμένες εμπειρίες χρηστών μπορεί να ενισχύσει την αποτελεσματικότητα της πλατφόρμας.
2. Ενσωμάτωση με Άλλα Εργαλεία: Η ενσωμάτωση της πλατφόρμας με επιπλέον εκπαιδευτικά εργαλεία και πόρους, όπως συστήματα διαχείρισης μάθησης (LMS) ή εργαλεία τηλεδιάσκεψης, μπορεί να προσφέρει μια πιο ολοκληρωμένη εκπαιδευτική εμπειρία.
3. Ανατροφοδότηση Χρηστών και Επανάληψη: Η συνεχής συλλογή ανατροφοδότησης από τους χρήστες και η επανάληψη στον σχεδιασμό και τη λειτουργικότητα της πλατφόρμας θα διασφαλίσει ότι καλύπτει τις εξελισσόμενες ανάγκες των χρηστών της.

Αυτή η διπλωματική εργασία απέδειξε επιτυχώς τις δυνατότητες χρήσης του WordPress, ενισχυμένου με Advanced Custom Fields και shortcodes, για τη δημιουργία μιας ισχυρής και αποδοτικής εκπαιδευτικής πλατφόρμας. Η υλοποίηση ανέδειξε τη σημασία της ευελιξίας, της επεκτασιμότητας και του σχεδιασμού με επίκεντρο τον χρήστη στην ανάπτυξη εκπαιδευτικών τεχνολογιών. Αντιμετωπίζοντας τις προκλήσεις και αξιοποιώντας τα πλεονεκτήματα αυτών των εργαλείων, η πλατφόρμα πέτυχε τους στόχους της για τη βελτίωση της διαχείρισης και της παράδοσης του εκπαιδευτικού περιεχομένου. Τα ευρήματα αυτής της μελέτης παρέχουν πολύτιμες γνώσεις για τις μελλοντικές εξελίξεις στην εκπαιδευτική τεχνολογία, συμβάλλοντας στις συνεχιζόμενες προσπάθειες για τη βελτίωση των ψηφιακών περιβαλλόντων μάθησης.