

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ



ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ

---

Ανάπτυξη διαδικτυακού συστήματος διαχείρισης  
δεδομένων του ΠΜΣ Εφαρμοσμένη Πληροφορική

---

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΣΤΑΥΡΟΥ Κ. ΚΑΡΑΚΕΠΕΛΗ

Επιβλέπων Καθηγητής: ΔΗΜΗΤΡΙΟΣ ΒΑΡΣΑΜΗΣ

ΚΑΘΗΓΗΤΗΣ

ΣΕΡΡΕΣ - ΔΕΚΕΜΒΡΙΟΣ 2023

©Σταύρος Κ. Καρακέπελης

©ΔΙ.ΠΑ.Ε.

Ανάπτυξη διαδικτυακού συστήματος διαχείρισης δεδομένων του ΠΜΣ  
Εφαρμοσμένη Πληροφορική

**Υπεύθυνη Δήλωση:** Βεβαιώνω ότι είμαι συγγραφέας αυτής της μεταπτυχιακής διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην διπλωματική εργασία. Επίσης, έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η μεταπτυχιακή διπλωματική εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του Προγράμματος Μεταπτυχιακών Σπουδών στην Εφαρμοσμένη Πληροφορική του Τμήματος Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών του ΔΙ.ΠΑ.Ε.



Υπογραφή

Αφιερώνεται στην οικογένειά μου

# Ευχαριστίες

Πρώτα απ' όλα, θα ήθελα να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας μου, Καθηγητή κ. Βαρσάμπ Δημήτριο, για την πολύτιμη βοήθεια και καθοδήγησή του κατά τη διάρκεια της δουλειάς μου. Επίσης, είμαι ευγνώμων στην οικογένειά μου για την ολόψυχη πίστη που είχαν σε εμένα και την ένθερμη υποστήριξή τους όλα αυτά τα χρόνια.

Τέλος, θα ήθελα να ευχαριστήσω το Τμήμα Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών του Διεθνούς Πανεπιστημίου της Ελλάδος για τη φιλοξενία και τις σπουδές που μου παρείχε καθώς και όλους τους καθηγητές του προγράμματος μεταπτυχιακών σπουδών "Εφαρμοσμένη Πληροφορική".

# Περίληψη

Στόχος της διπλωματικής εργασίας είναι η ανάπτυξη ενός πληροφοριακού συστήματος όπου θα συλλέγονται δεδομένα από τα πληροφοριακά συστήματα του ΠΜΣ στην Εφαρμοσμένη Πληροφορική, θα μοντελοποιούνται με σκοπό τη στατιστική ανάλυσή τους και τη χρήση τους για ερευνητικούς σκοπούς. Το πληροφοριακό σύστημα θα υποστηρίζει την αυτοματοποιημένη διαδικασία καταχώρησης των δεδομένων.

Για την ανάπτυξη του πληροφοριακού συστήματος χρησιμοποιήθηκε η γλώσσα προγραμματισμού υψηλού επιπέδου Python στο προγραμματιστικό περιβάλλον PyCharm με τη βοήθεια του framework Django, καθώς και το σύστημα διαχείρισης βάσης δεδομένων SQLite3.

Ο χρήστης επιλέγει τα επιθυμητά στοιχεία από τη βάση δεδομένων του πληροφοριακού συστήματος και στη συνέχεια τα δεδομένα αυτά εξάγονται σε ένα δομημένο αρχείο της μορφής .xlsx και είναι έτοιμα προς μοντελοποίηση και εξαγωγή αποτελεσμάτων μέσω στατιστικής ανάλυσης.

**Λέξεις-κλειδιά:** Διαδικτυακή πλατφόρμα, Σύστημα διαχείρισης φοιτητών, Φοιτητές, Ανώτατα εκπαιδευτικά ιδρύματα, Προσαρμοστικότητα συστήματος, Τεχνολογίες πληροφορικής και επικοινωνιών, Επεξεργασία δεδομένων, Διαδίκτυο, Python, Django, SQLite, HTML, CSS, JavaScript

# Abstract

The aim of this diploma thesis is to develop an information system where data will be collected from the information systems of the MSc. in Applied Informatics, will be modeled for statistical analysis and used for research purposes. The information system will support the automated data entry process.

The high-level programming language Python, in the PyCharm programming environment, was used to develop the information system with the help of the Django framework and the SQLite3 database management system.

The user selects the desired data from the database of the information system and then it exports them to a structured file (e.g. .csv, .xlsx, .txt) so it is ready for modelling and extraction of results through statistical analysis.

**Keywords:** Web application, student management system, Students, Higher education institutes, Responsive design, Information and communication technologies, ICT, Data Manipulation, Internet, Python, Django, SQLite, HTML, CSS, JavaScript

# Περιεχόμενα

<b>Περίληψη</b>	<b>IV</b>
<b>Abstract</b>	<b>V</b>
<b>1 Εισαγωγή</b>	<b>1</b>
1.1 Εισαγωγικά στοιχεία . . . . .	1
1.2 Αντικείμενο της διπλωματικής . . . . .	2
1.3 Οργάνωση του τόμου . . . . .	3
<b>2 Σχεδιασμός πλατφόρμας</b>	<b>4</b>
2.1 Υλοποίηση . . . . .	4
2.2 Η γλώσσα Python . . . . .	4
2.3 Το πλαίσιο Django . . . . .	7
2.4 Η βάση δεδομένων SQLite3 . . . . .	9
2.5 HTML, CSS και JavaScript . . . . .	11
2.5.1 HTML . . . . .	11
2.5.2 CSS . . . . .	12
2.5.3 JavaScript . . . . .	14
2.6 Επιπρόσθετα εργαλεία και βιβλιοθήκες . . . . .	15
2.6.1 Bootstrap . . . . .	16
2.6.2 jQuery . . . . .	18
2.6.3 Chart.js . . . . .	18
2.6.4 Υπόλοιπες βιβλιοθήκες εφαρμογής . . . . .	19
<b>3 Οδηγός χρήσης</b>	<b>20</b>
3.1 Εγγραφή χρήστη . . . . .	20
3.2 Είσοδος χρήστη . . . . .	22
3.3 Πλατφόρμα διαχείρισης φοιτητών . . . . .	22

3.3.1	Αρχική σελίδα . . . . .	22
3.3.2	Αρχείο φοιτητών . . . . .	23
3.3.3	Αρχείο βαθμολογιών . . . . .	25
3.3.4	Αποθήκευση τελικού αρχείου . . . . .	27
3.4	Γενικές παρατηρήσεις . . . . .	29
<b>4</b>	<b>Συμπεράσματα</b>	<b>31</b>
4.1	Σύνοψη . . . . .	31
4.2	Μελλοντική διερεύνηση . . . . .	32
<b>A'</b>	<b>Python scripts</b>	<b>34</b>
A'.1	models.py . . . . .	34
A'.1.1	Students . . . . .	34
A'.1.2	StudentGrades . . . . .	36
A'.2	views.py . . . . .	37
A'.2.1	Login/Logout . . . . .	38
A'.2.2	Register . . . . .	39
A'.2.3	Update/Create Students . . . . .	40
A'.2.4	Update/Create Grades . . . . .	43
A'.2.5	Preview Students . . . . .	45
A'.2.6	Preview Grades . . . . .	46
A'.2.7	Preview/Download . . . . .	47
A'.3	urls.py . . . . .	50
<b>B'</b>	<b>Database scripts</b>	<b>51</b>
B'.1	DeleteStudents.py . . . . .	51
B'.2	DeleteStudentGrades.py. . . . .	52
B'.3	DeleteDB.py . . . . .	53



# Κατάλογος σχημάτων

2.1	Σελίδα εγκατάστασης της Python . . . . .	5
2.2	Αρχική σελίδα του Django . . . . .	7
2.3	Εγκατεστημένα πακέτα και βιβλιοθήκες της πλατφόρμας . . . . .	19
3.1	Είσοδος ή δημιουργία νέου χρήστη . . . . .	21
3.2	Δημιουργία νέου χρήστη . . . . .	21
3.3	Έξοδος χρήστη από το σύστημα . . . . .	22
3.4	Αρχική σελίδα . . . . .	23
3.5	Σελίδα upload students dataset . . . . .	24
3.6	Σελίδα preview students dataset . . . . .	25
3.7	Σελίδα upload grades dataset . . . . .	26
3.8	Σελίδα preview grades dataset . . . . .	27
3.9	Σελίδα download . . . . .	27
3.10	Σελίδα preview and download . . . . .	28

# Κεφάλαιο 1

## Εισαγωγή

### 1.1 Εισαγωγικά στοιχεία

Στο δυναμικό τοπίο της σύγχρονης εκπαίδευσης, ο ρόλος της τεχνολογίας γίνεται όλο και πιο καθοριστικός. Μία από τις πρωταρχικές ανάγκες των εκπαιδευτικών ιδρυμάτων, από σχολεία έως πανεπιστήμια, είναι η αποτελεσματική διαχείριση των πληροφοριών των μαθητών. Η έλευση των συστημάτων διαχείρισης φοιτητών έφερε επανάσταση στον τρόπο με τον οποίο τα εκπαιδευτικά ιδρύματα χειρίζονται τα διοικητικά καθήκοντα, τα ακαδημαϊκά αρχεία και την επικοινωνία. Γίνεται διερεύνηση της επιτακτικής ανάγκης για τα συστήματα διαχείρισης φοιτητών, εμβαθύνοντας στα πολύπλευρα οφέλη τους και στον μετασχηματιστικό αντίκτυπο που επιφέρουν στο εκπαιδευτικό τοπίο.

Τα εκπαιδευτικά ιδρύματα είναι περίπλοκα οικοσυστήματα με πολυάριθμες διοικητικές εργασίες, όπως η εγγραφή, η παρακολούθηση της παρουσίας, η βαθμολόγηση και ο προγραμματισμός. Τα συστήματα διαχείρισης φοιτητών λειτουργούν ως κεντρικές πλατφόρμες, εξορθολογίζοντας αυτές τις διαδικασίες και μειώνοντας το διοικητικό βάρος για τους εκπαιδευτικούς και το προσωπικό. Η αυτοματοποίηση των καθηκόντων ρουτίνας όχι μόνο βελτιώνει την αποδοτικότητα, αλλά και ελαχιστοποιεί την πιθανότητα λαθών, εξασφαλίζοντας ακριβή και συνεπή τήρηση αρχείων.

Τα συστήματα διαχείρισης των σπουδαστών παρέχουν στα εκπαιδευτικά ιδρύματα πολύτιμη ανάλυση δεδομένων. Συλλέγοντας και αναλύοντας δεδομένα που σχετίζονται με την απόδοση και τη φοίτηση των μαθητών, τα ιδρύματα μπορούν να λαμβάνουν τεκμηριωμένες αποφάσεις για τον εντοπισμό τομέων προς βελτίωση και την εφαρμογή στοχευμένων παρεμβάσεων. Οι πληροφορίες που βασίζονται στα δεδομένα συμβάλλουν σε ένα πιο ευέλικτο και προσαρμοστικό εκπαιδευτικό περιβάλλον.

Τα εκπαιδευτικά ιδρύματα διαχειρίζονται ευαίσθητες πληροφορίες για τους φοιτητές που απαιτούν ισχυρά μέτρα ασφαλείας. Τα συστήματα διαχείρισης φοιτητών εφαρμόζουν αυστηρά πρωτόκολλα ασφάλειας δεδομένων για τη διασφάλιση των αρχείων των φοιτητών και τη διατήρηση της συμμόρφωσης με τους κανονισμούς προστασίας δεδομένων. Αυτό προστατεύει το απόρρητο των μαθητών και ενισχύει την εμπιστοσύνη των ενδιαφερομένων.

Κάθε εκπαιδευτικό ίδρυμα έχει μοναδικές απαιτήσεις και ροές εργασίας. Τα συστήματα διαχείρισης φοιτητών προσφέρουν υψηλό βαθμό προσαρμογής, επιτρέποντας στα ιδρύματα να προσαρμόσουν το σύστημα στις ιδιαίτερες ανάγκες τους. Το σύστημα μπορεί να ευθυγραμμιστεί άφογα με τη δομή και τις διαδικασίες του ιδρύματος.

Τα συστήματα διαχείρισης φοιτητών συμβάλλουν σε μια κουλτούρα λογοδοσίας και διαφάνειας. Οι εκπαιδευτικοί μπορούν να αξιολογούν και να καταγράφουν με ακρίβεια τις επιδόσεις των μαθητών, παρέχοντας σαφή εικόνα των ακαδημαϊκών επιτευγμάτων και των τομέων που χρήζουν βελτίωσης. Αυτή η διαφάνεια επεκτείνεται και στους μαθητές, οι οποίοι αποκτούν εικόνα της προόδου τους, καλλιεργώντας την αίσθηση ευθύνης για τη μαθησιακή τους πορεία.

Συμπερασματικά, η υιοθέτηση των συστημάτων διαχείρισης φοιτητών δεν αποτελεί απλώς μια τεχνολογική αναβάθμιση αλλά αντιπροσωπεύει μια θεμελιώδη αλλαγή στον τρόπο λειτουργίας των εκπαιδευτικών ιδρυμάτων και στην αλληλεπίδραση με τους ενδιαφερόμενους φορείς τους. Η ανάγκη για αυτά τα συστήματα οφείλεται στην επιδίωξη της αποτελεσματικότητας, της διαφάνειας και της βελτίωσης των εκπαιδευτικών αποτελεσμάτων. Τα συστήματα διαχείρισης φοιτητών δεν είναι απλώς εργαλεία διοικητικής ευκολίας, είναι καταλύτες θετικών αλλαγών στο εκπαιδευτικό τοπίο. Καθώς η τεχνολογία συνεχίζει να εξελίσσεται, η ενσωμάτωση των συστημάτων διαχείρισης φοιτητών θα διαδραματίσει καθοριστικό ρόλο στη διαμόρφωση μιας πιο συνδεδεμένης, ενημερωμένης και μαθητοκεντρικής προσέγγισης στην εκπαίδευση.

## 1.2 Αντικείμενο της διπλωματικής

Η παρούσα διπλωματική εργασία πραγματεύεται τη σχολαστική ανάλυση των ποικίλων πτυχών ενός συστήματος διαχείρισης φοιτητών που υλοποιήθηκε με τη χρήση της υψηλού επιπέδου γλώσσας προγραμματισμού Python, σε συνδυασμό με εξειδικευμένες βιβλιοθήκες, το δυναμικό διαδικτυακό πλαίσιο Django, και την ευέλικτη βάση δεδομένων SQLite3. Στόχος αυτής της εργασίας είναι να προσφέρει μια πλήρη ανάλυση

όχι μόνο της βασικής δομής του συστήματος, αλλά και των δυνατοτήτων του, καθώς και των επιδράσεών του στην αποτελεσματική διαχείριση των δεδομένων των φοιτητών του εκπαιδευτικού ιδρύματος του Διεθνούς Πανεπιστημίου της Ελλάδος, του μεταπτυχιακού προγράμματος σπουδών "Εφαρμοσμένη Πληροφορική" του τμήματος Μηχανικών Πληροφορικής, Υπολογιστών και Τηλεπικοινωνιών.

### 1.3 Οργάνωση του τόμου

Η διπλωματική εργασία είναι δομημένη στα εξής κεφάλαια:

Στο πρώτο κεφάλαιο γίνονται εισαγωγικές αναφορές και εξηγούνται οι λόγοι και οι ανάγκες που οδήγησαν στην εκπόνηση της συγκεκριμένης εργασίας και η σημασία των συστημάτων διαχείρισης φοιτητών στον ακαδημαϊκό κύκλο.

Στο δεύτερο κεφάλαιο περιγράφονται οι απαραίτητες τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της πλατφόρμας, οι βιβλιοθήκες και τα απαραίτητα διαδικτυακά πακέτα που πρέπει να εγκατασταθούν για την ορθή λειτουργία της πλατφόρμας.

Το τρίτο κεφάλαιο αποτελεί αναλυτικό οδηγό χρήσης της εφαρμογής του συστήματος διαχείρισης των φοιτητών και καθοδηγεί το χρήστη βήμα-βήμα με στόχο την κατανόηση του τρόπου λειτουργίας και την αντίληψη των χαρακτηριστικών της πλατφόρμας.

Στο τέταρτο κεφάλαιο σκιαγράφονται μελλοντικές βελτιώσεις και επιπρόσθετες λειτουργικότητες που μπορούν να προστεθούν στην πλατφόρμα, ώστε αυτή να γίνει πιο διαδραστική και φιλική στον χρήστη.

Τέλος, στο παράρτημα παρουσιάζονται τα πιο σημαντικά αρχεία του Django καθώς και οι συναρτήσεις που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Πιο συγκεκριμένα, το αρχείο `models.py` το οποίο πραγματεύεται τους πίνακες (μοντέλα) της βάσης δεδομένων SQLite3, το αρχείο `views.py` που περιέχει όλες τις συναρτήσεις που αφορούν τη λειτουργικότητα της εφαρμογής και το αρχείο `urls.py` που περιέχει τους συνδέσμους της εφαρμογής.

# Κεφάλαιο 2

## Σχεδιασμός πλατφόρμας

### 2.1 Υλοποίηση

Για το σχεδιασμό και την υλοποίηση της πλατφόρμας του συστήματος διαχείρισης φοιτητών, χρησιμοποιήθηκε το πλαίσιο ανάπτυξης ιστοσελίδων Django σε συνδυασμό με τη γλώσσα προγραμματισμού Python, τις γλώσσες HTML, CSS, JavaScript καθώς και επιπρόσθετα πακέτα και βιβλιοθήκες.

Σε αυτό το κεφάλαιο γίνεται ανάλυση των παραπάνω, ενώ στο τρίτο κεφάλαιο, δίνεται ένας αναλυτικός οδηγός της πλατφόρμας για το χρήστη.

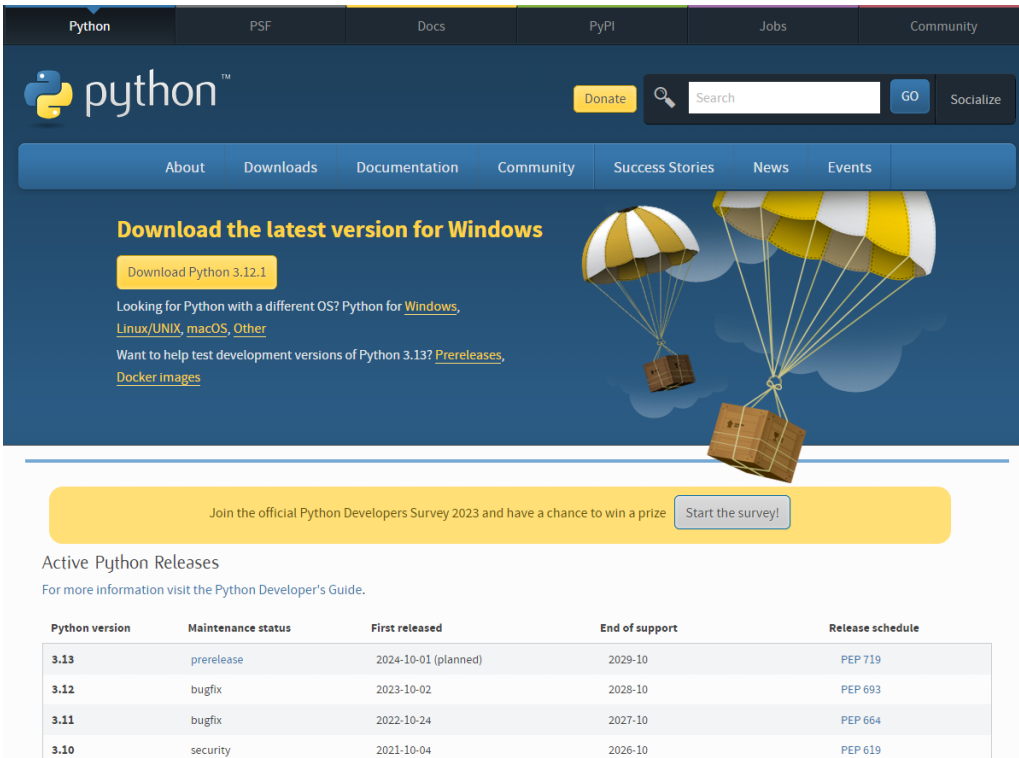
### 2.2 Η γλώσσα Python

Στη σφαίρα των γλωσσών προγραμματισμού, η Python αποτελεί απόδειξη της απλότητας, της αναγνωσιμότητας και της ευελιξίας. Δημιουργήθηκε από τον Guido van Rossum στα τέλη της δεκαετίας του 1980. Η Python ανέβηκε γρήγορα στις τάξεις των πιο ευρέως χρησιμοποιούμενων γλωσσών προγραμματισμού, δίνοντας τη δυνατότητα στους προγραμματιστές σε ένα ευρύ φάσμα τομέων. Το φιλικό προς το χρήστη συντακτικό της και η εκτεταμένη πρότυπη βιβλιοθήκη της, την έχουν καταστήσει δημοφιλής επιλογή τόσο για αρχάριους όσο και για έμπειρους προγραμματιστές [1].

Στην καρδιά της ελκυστικότητας της Python βρίσκεται η έμφαση που δίνει στην αναγνωσιμότητα. Ο κώδικάς της είναι γραμμένος σε ένα στυλ που μοιάζει με φυσική γλώσσα, καθιστώντας τον ευκολότερο στην κατανόηση και στη συντήρησή. Αυτό το χαρακτηριστικό είναι ιδιαίτερα ευεργετικό για τους νεοεισερχόμενους στον προγραμματισμό, καθώς εξαλείφει την απότομη καμπύλη εκμάθησης απαιτείται [2].

Η απλότητα της Python επεκτείνεται πέρα από το συντακτικό της, διαπερνώντας τη συνολική δομή και το σχεδιασμό της. Το σύστημα δυναμικής τυποποίησης, όπου οι τύποι των μεταβλητών καθορίζονται κατά την εκτέλεση, μειώνει τον κώδικα που συχνά απαιτείται σε άλλες γλώσσες. Αυτή η εξορθολογισμένη προσέγγιση συμβάλλει σε ένα πιο συνοπτικό και εκφραστικό στυλ κωδικοποίησης. Για την εγκατάσταση της python στο σύστημά μας, επισκεπτόμαστε την τοποθεσία <https://www.python.org/downloads> [3].

Ένας άλλος βασικός παράγοντας για τη δημοτικότητα της Python είναι η εκτεταμένη τυποποιημένη βιβλιοθήκη της. Αυτή η συλλογή προ-γραμμένων ενοτήτων και συναρτήσεων παρέχει μια πληθώρα λειτουργιών, εξαλείφοντας την ανάγκη να ξανακαλύψουμε τον τροχό για κοινές εργασίες. Από την επεξεργασία δεδομένων και την ανάπτυξη ιστοσελίδων έως τον επιστημονικό υπολογισμό και τη μηχανική μάθηση, η πρότυπη βιβλιοθήκη προσφέρει μια τεράστια εργαλειοθήκη για την αντιμετώπιση ποικίλων προγραμματιστικών προκλήσεων.



Python version	Maintenance status	First released	End of support	Release schedule
3.13	prerelease	2024-10-01 (planned)	2029-10	PEP 719
3.12	bugfix	2023-10-02	2028-10	PEP 693
3.11	bugfix	2022-10-24	2027-10	PEP 664
3.10	security	2021-10-04	2026-10	PEP 619

Σχήμα 2.1: Σελίδα εγκατάστασης της Python

Η ευελιξία της Python την έχει προωθήσει σε ένα ευρύ φάσμα εφαρμογών, καθιστώντας την πολύτιμο εργαλείο για προγραμματιστές σε διάφορους κλάδους. Η δημοτικότητά της πηγάζει από την προσαρμοστικότητά της σε ένα ευρύ φάσμα εργασιών, από την ανάπτυξη ιστοσελίδων και την επιστήμη των δεδομένων μέχρι τη δημιουργία σεναρίων και την αυτοματοποίηση.

**Ανάπτυξη ιστοσελίδων με Python:** Τα πλαίσια ανάπτυξης ιστοσελίδων, όπως το Django που θα χρησιμοποιήσουμε στην δική μας πλατφόρμα, έχουν γίνει βασικοί πυλώνες για την κατασκευή δυναμικών και κλιμακούμενων διαδικτυακών εφαρμογών. Αυτά τα πλαίσια παρέχουν μια ισχυρή βάση για τη δημιουργία διαδραστικών ιστοσελίδων, τη διαχείριση βάσεων δεδομένων και το χειρισμό αλληλεπιδράσεων με τους χρήστες.

**Επιστήμη δεδομένων:** Οι εκτεταμένες βιβλιοθήκες της Python για χειρισμό, ανάλυση και οπτικοποίηση δεδομένων την καθιστούν απαραίτητο εργαλείο για τους επιστήμονες δεδομένων. Βιβλιοθήκες όπως οι Pandas και NumPy προσφέρουν ισχυρά εργαλεία για την προεπεξεργασία, τον καθαρισμό και την ανάλυση μεγάλων συνόλων δεδομένων, ενώ εργαλεία όπως τα Matplotlib και Seaborn διευκολύνουν την οπτικοποίηση δεδομένων.

**Scripting και αυτοματοποίηση:** Η απλότητα και η επεκτασιμότητα της Python την καθιστούν ιδανική επιλογή για την αυτοματοποίηση εργασιών και την ενσωμάτωση συστημάτων. Η ικανότητά της να αλληλεπιδρά με διάφορα στοιχεία λογισμικού και υλικού ανοίγει ένα ευρύ φάσμα δυνατοτήτων για την αυτοματοποίηση επαναλαμβανόμενων διαδικασιών και τον εξορθολογισμό των ροών εργασίας [4].

Η διάχυτη παρουσία της Python στον κλάδο της τεχνολογίας είχε βαθύτατο αντίκτυπο στην ανάπτυξη λογισμικού και την καινοτομία. Η ευελιξία και η ευκολία χρήσης της έχουν εκδημοκρατίσει τον προγραμματισμό, καθιστώντας τον προσιτό σε μια ευρύτερη ομάδα προγραμματιστών. Αυτό έχει οδηγήσει στη δημιουργία αμέτρητων έργων ανοικτού κώδικα, συμβάλλοντας σε μια ακμάζουσα κοινότητα συνεργασίας και ανταλλαγής γνώσεων.

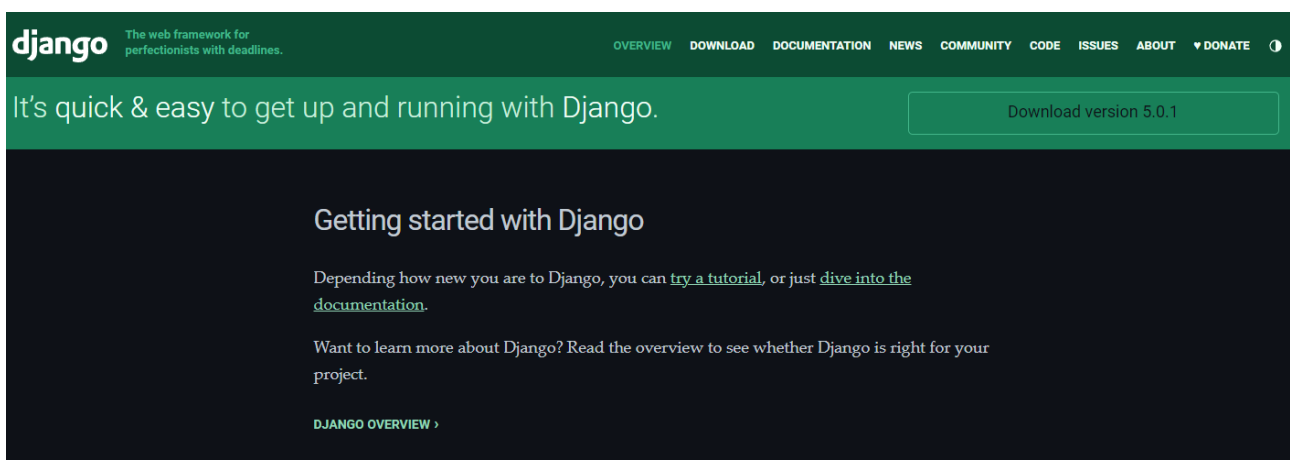
Η Python έχει επίσης διαδραματίσει καθοριστικό ρόλο στην πρόοδο της τεχνητής νοημοσύνης και της μηχανικής μάθησης. Οι βιβλιοθήκες της, όπως η TensorFlow και η scikit-learn, έχουν δώσει τη δυνατότητα στους ερευνητές και τους προγραμματιστές να δημιουργήσουν ισχυρά μοντέλα μηχανικής μάθησης για μια ποικιλία εφαρμογών, από την αναγνώριση εικόνων και την επεξεργασία φυσικής γλώσσας έως τα συστήματα συστάσεων και την ανίχνευση απάτης [5].

Καθώς ο κόσμος της τεχνολογίας συνεχίζει να εξελίσσεται, η σημασία της Python αναμένεται να αυξηθεί. Η προσαρμοστικότητα και η ευελιξία της την καθιστούν κατάλληλη για την αντιμετώπιση των αναδυόμενων προκλήσεων και ευκαιριών σε διάφορους τομείς. Είτε πρόκειται για την ανάπτυξη πρωτοποριακών αλγορίθμων τεχνητής νοημοσύνης, είτε για την ανάλυση σύνθετων συνόλων δεδομένων, είτε για την αυτοματοποίηση καθημερινών εργασιών, η Python προσφέρει ένα ισχυρό και προσιτό σύνολο εργαλείων για την αντιμετώπιση ενός ευρέος φάσματος προβλημάτων. Η Python δεν είναι απλώς

μια γλώσσα προγραμματισμού, αλλά ένα οικοσύστημα εργαλείων, βιβλιοθηκών και μιας ζωντανής κοινότητας που δίνει τη δυνατότητα στους προγραμματιστές να επιλύουν πραγματικά προβλήματα σε διάφορους τομείς. Η απλότητα, η ευελιξία και η εκτεταμένη τυποποιημένη βιβλιοθήκη της την έχουν καταστήσει γλώσσα επιλογής τόσο για αρχάριους όσο και για ειδικούς, και το μέλλον της στο τεχνολογικό τοπίο είναι λαμπρό. Καθώς ο κόσμος γίνεται όλο και πιο διασυνδεδεμένος και καθοδηγούμενος από δεδομένα, η ικανότητα της Python να χειρίζεται πολύπλοκες εργασίες και να ενσωματώνεται απρόσκοπτα με άλλα συστήματα θα συνεχίσει να την καθιστά απαραίτητο εργαλείο για το μέλλον της τεχνολογίας.

## 2.3 Το πλαίσιο Django

Στο διαρκώς εξελισσόμενο τοπίο της ανάπτυξης ιστοσελίδων, τα πλαίσια ανάπτυξης παίζουν καθοριστικό ρόλο στην απλοποίηση της διαδικασίας δημιουργίας ισχυρών και επεκτάσιμων εφαρμογών. Ένα τέτοιο πλαίσιο που έχει αποκτήσει τεράστια δημοτικότητα και αναγνώριση είναι το Django. Αναπτυγμένο σε Python και σχεδιασμένο για να προωθεί την ταχεία ανάπτυξη, το Django ενσωματώνει τη φιλοσοφία του «μην επαναλαμβάνεσαι» (Don't Repeat Yourself - DRY) και ενθαρρύνει τη δημιουργία καθαρού, συντηρήσιμου κώδικα. Η ενότητα αυτή εμβαθύνει στα βασικά χαρακτηριστικά και τις αρχές που καθιστούν το Django μια εξαιρετική επιλογή για τους προγραμματιστές ιστού.



Σχήμα 2.2: Αρχική σελίδα του Django



Το Django, που πήρε το όνομά του από τον θρυλικό καθαρίστα της τζαζ Django Reinhardt, δημιουργήθηκε από τους Adrian Holovaty και Simon Willison το 2003. Αρχικά αναπτύχθηκε για την αίθουσα σύνταξης της εφημερίδας Lawrence Journal-World και κυκλοφόρησε ως έργο ανοικτού κώδικα το 2005. Δίνει τη δυνατότητα στους προγραμματιστές να δημιουργούν εφαρμογές γρήγορα χωρίς να θυσιάζουν τις βέλτιστες πρακτικές, προωθώντας τη χρήση επαναχρησιμοποιήσιμων στοιχείων και σαφούς, περιεκτικού κώδικα [6].

Το Django ακολουθεί την αρχιτεκτονική Model-View-Controller (MVC), η οποία ενισχύει την οργάνωση του κώδικα και τον διαχωρισμό των προβλημάτων. Ωστόσο, η άποψη του Django για το MVC αναφέρεται συχνά ως Model-View-Template (MVT), όπου το πρότυπο λειτουργεί ως γέφυρα μεταξύ της προβολής και του ελεγκτή. Αυτή η δομή εξασφαλίζει έναν καθαρό διαχωρισμό μεταξύ των μοντέλων δεδομένων, της λογικής παρουσίασης και της διεπαφής χρήστη, διευκολύνοντας την ευκολότερη συντήρηση και επεκτασιμότητα.

Ένα από τα δυνατά σημεία του Django είναι τα αποσυνδεδεμένα συστατικά του. Το πλαίσιο είναι αρθρωτό, επιτρέποντας στους προγραμματιστές να χρησιμοποιούν συγκεκριμένα μέρη του ανεξάρτητα. Το Django ORM (Object-Relational Mapping) είναι ένα χαρακτηριστικό παράδειγμα, παρέχοντας ένα υψηλού επιπέδου API για την αλληλεπίδραση με βάσεις δεδομένων χωρίς την ανάγκη για ακατέργαστα ερωτήματα SQL. Αυτή η αποσύνδεση επιτρέπει στους προγραμματιστές να ανταλλάσσουν τα στοιχεία ανάλογα με τις ανάγκες, προωθώντας την ευελιξία και την προσαρμοστικότητα [7]. Για να κατεβάσει κανείς το πλαίσιο Django, αρκεί να μεταβεί στη σελίδα <https://www.djangoproject.com/>

Η ενσωματωμένη διεπαφή διαχείρισης του Django είναι ένα χαρακτηριστικό που ξεχωρίζει, παρέχοντας έναν ισχυρό και προσαρμόσιμο πίνακα διαχείρισης του backend από το κουτί. Με λίγες μόνο γραμμές κώδικα, οι προγραμματιστές μπορούν να δημιουργήσουν μια πλήρως λειτουργική διεπαφή διαχειριστή για τη διαχείριση των εγγραφών της βάσης δεδομένων. Αυτό όχι μόνο απλοποιεί τη διαδικασία ανάπτυξης, αλλά εξασφαλίζει επίσης μια συνεπή και φιλική προς το χρήστη εμπειρία για τους διαχειριστές περιεχομένου [8].

Επιπλέον, η ικανότητα του Django να δημιουργεί αυτόματα μια διεπαφή διαχειριστή με βάση τα μοντέλα δεδομένων είναι ένα χαρακτηριστικό που εξοικονομεί χρόνο. Αυτή η αυτόματη δημιουργία διαχειριστή μειώνει τον κώδικα που απαιτείται για τη δημιουργία λειτουργιών CRUD (Create, Read, Update, Delete), επιτρέποντας στους προγραμματι-

στές να επικεντρωθούν στη δημιουργία χαρακτηριστικών που αφορούν συγκεκριμένες εφαρμογές και όχι σε επαναλαμβανόμενες διαχειριστικές εργασίες.

Το Django δίνει μεγάλη έμφαση στην ασφάλεια, παρέχοντας εργαλεία και χαρακτηριστικά για τον μετριασμό των κοινών ευπαθειών. Περιλαμβάνει ενσωματωμένη προστασία από επιθέσεις SQL injection, cross-site scripting (XSS) και cross-site request forgery (CSRF). Το πλαίσιο ενθαρρύνει τους προγραμματιστές να ακολουθούν βέλτιστες πρακτικές, όπως παραμετροποιημένα ερωτήματα και ασφαλή αποθήκευση κωδικών πρόσβασης [9].

Η επεκτασιμότητα είναι ένας άλλος τομέας στον οποίο το Django υπερέχει. Οι κοινές εργασίες είναι άμεσα διαθέσιμες και οι βελτιστοποιήσεις μπορούν να υλοποιηθούν ανάλογα με τις ανάγκες. Η χρήση ενδιάμεσου λογισμικού, μηχανισμών προσωρινής αποθήκευσης και η υποστήριξη διαφόρων στρατηγικών ανάπτυξης καθιστά το Django κατάλληλο για το χειρισμό εφαρμογών διαφορετικών μεγεθών και πολυπλοκότητας.

Το Django αποτελεί απόδειξη της κομψότητας και της αποτελεσματικότητας που μπορεί να προσφέρει ένα καλά σχεδιασμένο πλαίσιο ιστού στη διαδικασία ανάπτυξης. Η δέσμευσή του στην απλότητα, την ευελιξία και την ασφάλεια το καθιστά αγαπημένο μεταξύ των προγραμματιστών που επιδιώκουν να δημιουργήσουν εύρωστες, συντηρήσιμες εφαρμογές. Καθώς η ανάπτυξη ιστού συνεχίζει να εξελίσσεται, οι διαχρονικές αρχές και τα ισχυρά χαρακτηριστικά του Django εξασφαλίζουν τη σημασία και την ελκυστικότητά του στο δυναμικό τοπίο της τεχνολογίας.

## 2.4 Η βάση δεδομένων SQLite3

Στα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), η βάση δεδομένων SQLite3 ξεχωρίζει ως μια ελαφριά αλλά ισχυρή λύση. Το SQLite3, που κυκλοφόρησε το 2000, έχει αποκτήσει ευρεία αποδοχή λόγω της απλότητας, της φορητότητας και της αποτελεσματικότητάς του. Σε αυτήν την ενότητα θα εμβαθύνουμε στα βασικά χαρακτηριστικά, την αρχιτεκτονική, τις περιπτώσεις χρήσης και τα πλεονεκτήματα του SQLite3, καταδεικνύοντας γιατί έχει γίνει δημοφιλής επιλογή για διάφορες εφαρμογές.

Η SQLite3 είναι μια μη εξυπηρετούμενη, αυτοδύναμη και μηδενικών ρυθμίσεων μηχανή βάσης δεδομένων που λειτουργεί ως βιβλιοθήκη συνδεδεμένη απευθείας με την εφαρμογή. Υλοποιείται στη γλώσσα προγραμματισμού C, γεγονός που την καθιστά ιδιαίτερα αποδοτική και εύκολα ενσωματώσιμη σε έργα λογισμικού. Ένα από τα ιδιαίτερα χαρακτηριστικά της είναι η απουσία ξεχωριστής διεργασίας διακομιστή, επιτρέποντάς

της να λειτουργεί ως ενσωματωμένη βάση δεδομένων μέσα σε εφαρμογές [10].

Η αρχιτεκτονική της SQLite3 περιστρέφεται γύρω από ένα μόνο ελαφρύ αρχείο που περιέχει ολόκληρη τη βάση δεδομένων. Αυτή η προσέγγιση απλοποιεί την ανάπτυξη και τη διαχείριση, καθιστώντας την κατάλληλη για σενάρια όπου ένας κεντρικός διακομιστής είναι περιττός ή ανέφικτος. Η αυτοτελής φύση της SQLite3 εξαλείφει την ανάγκη για πολύπλοκες διαδικασίες εγκατάστασης, επιτρέποντας την απρόσκοπτη ενσωμάτωση σε εφαρμογές με ελάχιστη προσπάθεια [11].

Παρά τον ελαφρύ σχεδιασμό του, η SQLite3 διαθέτει ένα ολοκληρωμένο σύνολο χαρακτηριστικών συγκρίσιμων με μεγαλύτερα RDBMS. Υποστηρίζει την τυπική σύνταξη SQL, εξασφαλίζοντας συμβατότητα με ένα ευρύ φάσμα εφαρμογών και περιβαλλόντων ανάπτυξης. Οι προγραμματιστές που είναι εξοικειωμένοι με την SQL μπορούν να αξιοποιήσουν τις δεξιότητές τους απρόσκοπτα όταν εργάζονται με το SQLite3.

Ένα αξιοσημείωτο χαρακτηριστικό είναι η υποστήριξη των ιδιοτήτων ACID (Atomicity, Consistency, Isolation, Durability), που διασφαλίζουν την αξιοπιστία και τη συνέπεια των συναλλαγών της βάσης δεδομένων. Επιπλέον, η SQLite3 υποστηρίζει σύνθετα ερωτήματα, ενώσεις και περιορισμούς ξένων κλειδιών, καθιστώντας την κατάλληλη για το χειρισμό περίπλοκων σχέσεων δεδομένων.

Ένα από τα σημαντικότερα πλεονεκτήματα του SQLite3 είναι η φορητότητα και η συμβατότητα μεταξύ διαφορετικών πλατφορμών. Η μηχανή βάσεων δεδομένων λειτουργεί απρόσκοπτα σε διάφορα λειτουργικά συστήματα (Windows, macOS, Linux) και σε πλατφόρμες κινητών συσκευών. Αυτή η ευελιξία καθιστά την SQLite3 ιδανική επιλογή για προγραμματιστές που αναζητούν μια συνεπή και αξιόπιστη λύση βάσης δεδομένων σε διαφορετικά περιβάλλοντα [12].

Η δυνατότητα ενσωμάτωσης της SQLite3 είναι ένας βασικός παράγοντας που συμβάλλει στη δημοτικότητά της. Δεδομένου ότι λειτουργεί ως βιβλιοθήκη συνδεδεμένη απευθείας με την εφαρμογή, δεν υπάρχει ανάγκη για ξεχωριστή διεργασία διακομιστή. Αυτό απλοποιεί τη διαδικασία ανάπτυξης και μειώνει το συνολικό αποτύπωμα πόρων, καθιστώντας την SQLite3 ιδιαίτερα κατάλληλη για περιβάλλοντα με περιορισμένους πόρους ή εφαρμογές με περιορισμένους πόρους συστήματος. Ο ελαφρύς σχεδιασμός της και η ευκολία ενσωμάτωσής της την καθιστούν εξαιρετική επιλογή για κινητές εφαρμογές, όπου η αποδοτικότητα και η απλότητα είναι ζωτικής σημασίας. Πολλά μεγάλα προγράμματα περιήγησης στο διαδίκτυο, συμπεριλαμβανομένων των Google Chrome και Firefox, χρησιμοποιούν την SQLite3 για την αποθήκευση σελιδοδεικτών, ιστορικού και άλλων τοπικών δεδομένων.

Επιπλέον, η SQLite3 χρησιμοποιείται συνήθως σε ενσωματωμένα συστήματα, συσκευές IoT και εφαρμογές γραφείου. Η ευελιξία της και η μηδενική ρύθμιση παραμέτρων την καθιστούν ελκυστική επιλογή για προγραμματιστές που εργάζονται σε έργα με ποικίλες απαιτήσεις και περιορισμούς.

Εν κατακλείδι, η άνοδος της SQLite3 στην κορυφή μπορεί να αποδοθεί στην απλότητα, τη φορητότητα και την αποτελεσματικότητά της. Ως αυτοτελής μηχανή βάσεων δεδομένων χωρίς διακομιστή, η SQLite3 έχει αποδείξει την ευελιξία της, προσαρμοζόμενη απρόσκοπτα σε διάφορους τομείς εφαρμογών. Είτε τροφοδοτεί εφαρμογές κινητών τηλεφώνων, ενσωματωμένα συστήματα ή λογισμικό γραφείου, η ελαφριά αλλά ισχυρή φύση της SQLite3 την τοποθετεί ως μια συναρπαστική επιλογή για τους προγραμματιστές που αναζητούν μια αξιόπιστη και εύκολα ενσωματώσιμη λύση βάσης δεδομένων. Περισσότερες πληροφορίες για την SQLite3 μπορεί να βρει κανείς στην επίσημα σελίδα <https://www.sqlite.org/index.html>

## 2.5 HTML, CSS και JavaScript

### 2.5.1 HTML

Η HTML, ή HyperText Markup Language, αποτελεί τον θεμέλιο λίθο του World Wide Web. Από την ίδρυσή της το 1991 από τον Sir Tim Berners-Lee, η HTML έχει διαδραματίσει καθοριστικό ρόλο στη δόμηση και παρουσίαση του περιεχομένου στο διαδίκτυο. Σε αυτήν την ενότητα γίνεται ανάλυση των θεμελιωδών χαρακτηριστικών, της εξέλιξης και της διαρκής σημασίας της HTML στη διαμόρφωση του ψηφιακού τοπίου [13].

Στον πυρήνα της, η HTML είναι μια γλώσσα σήμανσης που δομεί το περιεχόμενο των ιστοσελίδων. Αποτελούμενη από ένα σύνολο ετικετών καθεμία από τις οποίες εξυπηρετεί έναν συγκεκριμένο σκοπό, η HTML επιτρέπει στους προγραμματιστές να καθορίζουν την ιεραρχία και τις σχέσεις μεταξύ των διαφόρων στοιχείων μιας ιστοσελίδας. Τα στοιχεία αυτά μπορεί να κυμαίνονται από επικεφαλίδες, παραγράφους και εικόνες μέχρι πιο σύνθετες δομές όπως φόρμες και πίνακες.

Η HTML έχει υποστεί αρκετές αναθεωρήσεις, με την HTML5 να είναι η τελευταία και πιο ευρέως υιοθετημένη έκδοση. Η HTML5, που παρουσιάστηκε το 2014, έφερε μια σειρά από νέα χαρακτηριστικά και βελτιώσεις, που ανταποκρίνονται στις εξελισσόμενες ανάγκες της ανάπτυξης ιστοσελίδων. Η υποστήριξη πολυμέσων, η βελτιωμένη

σημασιολογία της και η συμβατότητα με τα σύγχρονα προγράμματα περιήγησης την έχουν καταστήσει πρότυπο του κλάδου, οδηγώντας στην καινοτομία και τη δημιουργικότητα στον σχεδιασμό ιστού.

Η δύναμη της HTML έγκειται στην ικανότητά της να παρέχει μια σαφή και ουσιαστική δομή στο περιεχόμενο του ιστού. Μέσω της σημασιολογικής σήμανσης, οι προγραμματιστές μπορούν να μεταφέρουν όχι μόνο την εμφάνιση αλλά και το νόημα των στοιχείων μιας σελίδας. Αυτό όχι μόνο ενισχύει την προσβασιμότητα για τους χρήστες με αναπηρίες, αλλά συμβάλλει και στην καλύτερη βελτιστοποίηση των μηχανών αναζήτησης, διασφαλίζοντας ότι το περιεχόμενο είναι τόσο αναγνώσιμο από τον άνθρωπο όσο και κατανοητό από τις μηχανές.

Ενώ η HTML εστιάζει κυρίως στη δομή του περιεχομένου, η ενσωμάτωσή της με άλλες τεχνολογίες, ιδίως με τη JavaScript και τη CSS (Cascading Style Sheets), δίνει τη δυνατότητα στους προγραμματιστές να δημιουργούν δυναμικές και οπτικά ελκυστικές εμπειρίες στον ιστό. Η HTML αλληλεπιδρά απρόσκοπτα με τη JavaScript για να προσθέσει διαδραστικότητα, ενώ η CSS διασφαλίζει ότι το επίπεδο παρουσίασης διαμορφώνεται λεπτομερώς σε διάφορες συσκευές και μεγέθη οθόνης [14].

Συνοψίζοντας, η HTML αποτελεί τον ακρογωνιαίο λίθο του διαδικτύου, παρέχοντας τη βασική δομή που στηρίζει κάθε ιστοσελίδα. Η εξέλιξή της, από τις πρώτες μέρες του διαδικτύου μέχρι τις εξελιγμένες εμπειρίες του διαδικτύου σήμερα, αντανακλά την προσαρμοστικότητα και την ανθεκτικότητα αυτής της θεμελιώδους γλώσσας. Καθώς οι τεχνολογίες συνεχίζουν να εξελίσσονται, η HTML παραμένει ένα διαχρονικό εργαλείο, που διαμορφώνει τα ψηφιακά σύνορα και επιτρέπει τη δημιουργία ποικίλων και ελκυστικών διαδικτυακών εμπειριών για τους χρήστες παγκοσμίως.

## 2.5.2 CSS

Το CSS, ή Cascading Style Sheets, είναι ένα βασικό συστατικό στον κόσμο της ανάπτυξης ιστοσελίδων, που συνεργάζεται με την HTML για να φέρει οπτική ελκυστικότητα και στυλ στο ψηφιακό πεδίο. Από την ίδρυσή της στα τέλη της δεκαετίας του 1990, η CSS έχει εξελιχθεί σε μια ισχυρή γλώσσα μορφοποίησης, επιτρέποντας στους προγραμματιστές να δημιουργούν οπτικά εντυπωσιακές και ευέλικτες διεπαφές ιστού. Σε αυτήν την ενότητα γίνεται ανάλυση των θεμελιωδών χαρακτηριστικών, της εξέλιξης και της διαρκής σημασίας της CSS στην τέχνη του σχεδιασμού ιστού.

Η CSS είναι μια γλώσσα που έχει σχεδιαστεί για να περιγράφει την παρουσίαση ενός εγγράφου γραμμένου σε HTML ή XML. Ο πρωταρχικός της ρόλος είναι να καθορίζει τις οπτικές πτυχές του περιεχομένου του ιστού, συμπεριλαμβανομένης της διάταξης, των χρωμάτων, των γραμματοσειρών και των διαστημάτων. Επιτρέποντας στους προγραμματιστές να διαχωρίσουν το περιεχόμενο και την παρουσίαση, η CSS παρέχει μια αρθρωτή και συντηρήσιμη προσέγγιση για τη διαμόρφωση ιστοσελίδων [15].

Η CSS έχει υποστεί διάφορες αναθεωρήσεις, με την CSS3 να είναι η τελευταία μεγάλη έκδοση. Η CSS3 παρουσιάστηκε το 1999 και εισήγαγε πολλές βελτιώσεις. Αυτή η έκδοση σηματοδότησε ένα σημαντικό άλμα προς τα εμπρός, επιτρέποντας στους προγραμματιστές να δημιουργούν περίπλοκους και δυναμικούς σχεδιασμούς, τηρώντας παράλληλα τα πρότυπα ιστού.

Η CSS λειτουργεί με ένα σύστημα επιλογών και κανόνων διαμόρφωσης. Οι επιλογείς στοχεύουν σε συγκεκριμένα στοιχεία HTML, ενώ οι κανόνες διαμόρφωσης καθορίζουν τον τρόπο με τον οποίο θα πρέπει να εμφανίζονται αυτά τα στοιχεία. Αυτός ο διαχωρισμός των προβλημάτων όχι μόνο διευκολύνει τον καθαρό και οργανωμένο κώδικα αλλά επιτρέπει επίσης γρήγορες και ολοκληρωμένες ενημερώσεις των οπτικών πτυχών ενός ιστότοπου.

Ένα από τα ιδιαίτερα χαρακτηριστικά της CSS είναι η ικανότητά της να δημιουργεί responsive σχέδια που προσαρμόζονται σε διαφορετικά μεγέθη οθόνης και συσκευές. Τα ερωτήματα πολυμέσων, που εισήχθησαν στην CSS3, επιτρέπουν στους προγραμματιστές να εφαρμόζουν στυλ επιλεκτικά με βάση παράγοντες όπως το πλάτος, το ύψος και την ανάλυση της οθόνης. Αυτή η ευελιξία εξασφαλίζει μια απρόσκοπτη εμπειρία του χρήστη σε διάφορες πλατφόρμες, από οθόνες γραφείου μέχρι smartphones και tablets.

Η CSS δίνει τη δυνατότητα στους προγραμματιστές να βελτιώνουν τις αλληλεπιδράσεις των χρηστών μέσω μεταβάσεων και κινούμενων σχεδίων. Οι μεταβάσεις επιτρέπουν ομαλές αλλαγές στις τιμές των ιδιοτήτων, όπως το χρώμα ή την αδιαφάνεια, κατά τη διάρκεια μιας καθορισμένης διάρκειας. Τα κινούμενα σχέδια, από την άλλη πλευρά, παρέχουν πιο λεπτομερή έλεγχο, επιτρέποντας τη δημιουργία δυναμικών και ελκυστικών οπτικών εφέ. Αυτά τα χαρακτηριστικά συμβάλλουν σε μια πλουσιότερη εμπειρία χρήστη και προσθέτουν ένα επίπεδο πολυπλοκότητας στο σχεδιασμό ιστού.

Συμπερασματικά, η CSS αποτελεί μια δημιουργική δύναμη στο χώρο της ανάπτυξης ιστοσελίδων, προσδίδοντας ζωή και στυλ στον ψηφιακό καμβά. Η εξέλιξή της, από τη βασική μορφοποίηση μέχρι τις εξελιγμένες δυνατότητες της CSS3, αναδεικνύει την προσαρμοστικότητα και την ευελιξία αυτής της γλώσσας μορφοποίησης. Καθώς οι τε-

χνολογίες ιστού συνεχίζουν να εξελίσσονται, η CSS παραμένει ένα απαραίτητο εργαλείο, δίνοντας τη δυνατότητα στους προγραμματιστές να διαμορφώνουν την οπτική ταυτότητα των ιστότοπων και να δημιουργούν συναρπαστικές διαδικτυακές εμπειρίες για τους χρήστες σε όλο τον κόσμο.

### 2.5.3 JavaScript

Η JavaScript, που γεννήθηκε στα μέσα της δεκαετίας του 1990, έχει αναδειχθεί σε μια δυναμική και ευέλικτη γλώσσα προγραμματισμού που δίνει ζωή στις ιστοσελίδες. Η JavaScript, η οποία αρχικά σχεδιάστηκε ως μια απλή γλώσσα σεναρίων για την προσθήκη διαδραστικότητας, έχει εξελιχθεί σε ένα ισχυρό και αναπόσπαστο μέρος της σύγχρονης ανάπτυξης ιστού. Σε αυτήν την ενότητα θα δούμε τα θεμελιώδη χαρακτηριστικά της, την εξέλιξη και τη διαρκή σημασία της JavaScript στη διαμόρφωση του διαδραστικού τοπίου του Παγκόσμιου Ιστού.

Η JavaScript είναι μια υψηλού επιπέδου, διερμηνευμένη γλώσσα προγραμματισμού που έχει σχεδιαστεί για να είναι ελαφριά και εύκολη στην ενσωμάτωση σε ιστοσελίδες. Σε αντίθεση με την HTML και την CSS, οι οποίες επικεντρώνονται στη δομή και την παρουσίαση, η JavaScript είναι μια ολοκληρωμένη γλώσσα προγραμματισμού που επιτρέπει τη δημιουργία δυναμικής, διαδραστικότητας από την πλευρά του πελάτη μέσα στους φυλλομετρητές. Με την ικανότητά της να χειρίζεται το Μοντέλο Αντικειμένου Εγγράφου (Document Object Model - DOM) και να ανταποκρίνεται στις ενέργειες του χρήστη, η JavaScript παίζει καθοριστικό ρόλο στη βελτίωση της εμπειρίας του χρήστη στον ιστό [16].

Το ταξίδι της JavaScript σημαδεύτηκε από μια σειρά εξελίξεων και τυποποιήσεων. Η εισαγωγή της ECMAScript, μιας προδιαγραφής που ορίζει τα βασικά χαρακτηριστικά της γλώσσας σεναρίων, εξασφάλισε τη συμβατότητα μεταξύ των φυλλομετρητών και άνοιξε το δρόμο για συνεκτικές πρακτικές ανάπτυξης. Οι επόμενες εκδόσεις του ECMAScript, όπως το ECMAScript 6 (ES6), έφεραν νέα χαρακτηριστικά και βελτιώσεις στο συντακτικό, επιτρέποντας στους προγραμματιστές να γράφουν πιο συνοπτικό και συντηρήσιμο κώδικα.

Ένα από τα βασικά πλεονεκτήματα της JavaScript έγκειται στην ικανότητά της να αλληλεπιδρά με το DOM, μια ιεραρχική αναπαράσταση της δομής και του περιεχομένου μιας ιστοσελίδας. Η JavaScript μπορεί να τροποποιεί δυναμικά το DOM, επιτρέποντας στους προγραμματιστές να δημιουργούν ευέλικτες και διαδραστικές διεπαφές χρήστη.

Ο χειρισμός συμβάντων, μια κρίσιμη πτυχή της διαδραστικότητας του ιστού, επιτρέπει στους προγραμματιστές να ανταποκρίνονται σε ενέργειες του χρήστη, όπως κλικ, πατήματα πλήκτρων και κινήσεις του ποντικιού, διευκολύνοντας τη δημιουργία ελκυστικών και φιλικών προς το χρήστη εφαρμογών.

Η ασύγχρονη φύση της JavaScript παίζει ζωτικό ρόλο στη διαχείριση εργασιών που μπορεί να χρειάζονται χρόνο για να ολοκληρωθούν, όπως η άντληση δεδομένων από έναν διακομιστή. Η ασύγχρονη JavaScript και XML (AJAX) επιτρέπει στις ιστοσελίδες να ανακτούν και να στέλνουν δεδομένα στο παρασκήνιο χωρίς να απαιτείται επαναφόρτωση της σελίδας. Η δυνατότητα αυτή συμβάλλει στην ανάπτυξη απρόσκοπτων και ευέλικτων εφαρμογών ιστού, βελτιώνοντας τη συνολική εμπειρία του χρήστη.

Το οικοσύστημα της JavaScript έχει οδηγήσει στην εμφάνιση πολυάριθμων πλαισίων και βιβλιοθηκών που απλοποιούν και επιταχύνουν την ανάπτυξη. Βιβλιοθήκες όπως η jQuery και πλαίσια όπως το React, το Angular και το Vue.js παρέχουν στους προγραμματιστές προκατασκευασμένα στοιχεία και αφαιρέσεις, απλοποιώντας τη δημιουργία σύνθετων, διαδραστικών εφαρμογών ιστού. Αυτά τα εργαλεία συμβάλλουν στην αποτελεσματικότητα και την επεκτασιμότητα των έργων που βασίζονται στη JavaScript.

Εν κατακλείδι, η JavaScript είναι ο δυναμικός παλμός της διαδραστικότητας του διαδικτύου, μετατρέποντας στατικές ιστοσελίδες σε δυναμικές και ελκυστικές εμπειρίες για τους χρήστες. Η εξέλιξή της, από μια ταπεινή γλώσσα σεναρίων σε μια στιβαρή και ευέλικτη γλώσσα προγραμματισμού, αντανακλά τη διαρκή σημασία της στο διαρκώς μεταβαλλόμενο τοπίο της ανάπτυξης ιστοσελίδων. Η ικανότητα της JavaScript να χειρίζεται το DOM, να χειρίζεται συμβάντα και να διευκολύνει τον ασύγχρονο προγραμματισμό έχει εδραιώσει το ρόλο της ως απαραίτητη γλώσσα για τη δημιουργία ευέλικτων και διαδραστικών εφαρμογών ιστού. Καθώς η τεχνολογία συνεχίζει να εξελίσσεται, η JavaScript παραμένει στην πρώτη γραμμή της καινοτομίας, οδηγώντας την εξέλιξη του σύγχρονου ιστού.

## 2.6 Επιπρόσθετα εργαλεία και βιβλιοθήκες

Πέρα από τη βασική δομή που χρειάζεται η πλατφόρμα για να λειτουργήσει, χρησιμοποιήθηκαν επιπρόσθετα εργαλεία και βιβλιοθήκες για να ενισχύσουν την εμπειρία του χρήστη. Σε αυτήν την ενότητα γίνεται ανάλυση αυτών των εργαλείων.



## 2.6.1 Bootstrap

Το Bootstrap, ένα front-end πλαίσιο που αναπτύχθηκε από το Twitter (τόρα X), έχει γίνει απαραίτητο εργαλείο στη σύγχρονη ανάπτυξη ιστοσελίδων. Το Bootstrap, που παρουσιάστηκε το 2011, απλοποιεί και επιταχύνει τη διαδικασία κατασκευής ανταποκρινόμενων, οπτικά ελκυστικών ιστότοπων. Στον πυρήνα του, το Bootstrap προσφέρει ένα ισχυρό σύστημα πλέγματος που διευκολύνει τη δημιουργία ευέλικτων και φιλικών προς τα κινητά διατάξεων. Αυτό το σύστημα πλέγματος συμπληρώνεται από μια ολοκληρωμένη συλλογή προ-σχεδιασμένων στοιχείων, όπως μπάρες πλοήγησης, κουμπιά, φόρμες και τυπογραφικά στοιχεία, καθιστώντας το ιδιαίτερα εύχρηστο [17].

Ένα από τα καθοριστικά χαρακτηριστικά του Bootstrap είναι η έμφαση που δίνει στην ανταπόκριση. Χρησιμοποιώντας ένα ευέλικτο σύστημα πλέγματος, οι προγραμματιστές μπορούν να διασφαλίσουν ότι οι ιστότοποί τους προσαρμόζονται απρόσκοπτα σε διάφορα μεγέθη οθόνης και συσκευές, παρέχοντας βέλτιστη εμπειρία χρήσης. Επιπλέον, η εκτεταμένη τεκμηρίωση και η ενεργή υποστήριξη της κοινότητας του Bootstrap το καθιστούν προσιτό σε προγραμματιστές όλων των επιπέδων δεξιοτήτων, προωθώντας ένα συνεργατικό περιβάλλον μάθησης και επίλυσης προβλημάτων.

Πέρα από τα βασικά χαρακτηριστικά του, το Bootstrap προσφέρει μια παραμετροποιήσιμη και θεματοποιήσιμη δομή, επιτρέποντας στους προγραμματιστές να προσαρμόσουν το πλαίσιο ώστε να ανταποκρίνεται στις συγκεκριμένες σχεδιαστικές απαιτήσεις των έργων τους. Με την έλευση του Bootstrap 5, το πλαίσιο συνεχίζει να εξελίσσεται, ενσωματώνοντας νέα χαρακτηριστικά και βελτιώσεις, διατηρώντας παράλληλα τη δέσμευσή του στην απλότητα και την ευελιξία. Στην ουσία, το Bootstrap αποτελεί ένα θεμελιώδες εργαλείο που δίνει τη δυνατότητα στους προγραμματιστές να δημιουργούν αποτελεσματικά ανταποκρινόμενους, πλούσιους σε χαρακτηριστικά ιστότοπους, καθιστώντας το ακρογωνιαίο λίθο στην εργαλειοθήκη των επαγγελματιών ανάπτυξης ιστοσελίδων παγκοσμίως.

### 2.6.1.1 SB Admin 2

Το SB Admin 2 είναι ένα δωρεάν και ανοιχτού κώδικα template οργάνων διαχείρισης, παρέχει ένα ισχυρό θεμέλιο για τη δημιουργία κομφών και πλούσιων σε χαρακτηριστικά διεπαφών διαχείρισης πάνω στο οποίο είναι βασισμένο η πλατφόρμα μας. Αναπτυγμένο από την Start Bootstrap, το SB Admin 2 βασίζεται στο πλαίσιο Bootstrap, κληρονομώντας το ευέλικτο σύστημα πλέγματος και τα βασικά χαρακτηριστικά του. Το SB Admin

2, που κυκλοφόρησε ως πρότυπο του Bootstrap 4, αξιοποιεί την ευελιξία του Bootstrap για να βελτιώσει την ανάπτυξη των πινάκων διαχείρισης.

Αυτό το πρότυπο διακρίνεται για τον ξεχωριστό σχεδιασμό του, προσφέροντας μια σύγχρονη και διαισθητική διεπαφή χρήστη. Το SB Admin 2 ενσωματώνει διάφορα στοιχεία UI, όπως διαγράμματα, πίνακες, φόρμες και στοιχεία πλοήγησης, διευκολύνοντας τη δημιουργία δυναμικών και διαδραστικών πινάκων οργάνων. Το πρότυπο περιλαμβάνει επίσης μια σειρά από προσαρμοσμένα πρόσθετα της JavaScript, ενισχύοντας την εμπειρία του χρήστη με προηγμένες λειτουργίες, όπως οπτικοποίηση δεδομένων και διαδραστικά στοιχεία.

Ένα από τα δυνατά σημεία του SB Admin 2 έγκειται στην ευκολία ενσωμάτωσής του. Οι προγραμματιστές μπορούν να ενσωματώσουν γρήγορα το SB Admin 2 στα έργα τους, εξοικονομώντας πολύτιμο χρόνο και προσπάθεια για το σχεδιασμό και την υλοποίηση διοικητικών διεπαφών. Επιπλέον, το πρότυπο είναι ιδιαίτερα παραμετροποιίσιμο, επιτρέποντας στους προγραμματιστές να προσαρμόζουν την εμφάνιση και τα χαρακτηριστικά του ώστε να ανταποκρίνονται στις συγκεκριμένες απαιτήσεις των εφαρμογών τους.

Η υποστήριξη της κοινότητας γύρω από το SB Admin 2 είναι αξιοσημείωτη, με μια ενεργή βάση χρηστών που συμβάλλει στην ανάπτυξή του και παρέχει βοήθεια μέσω φόρουμ και τεκμηρίωσης. Ως αποτέλεσμα, το SB Admin 2 έχει κερδίσει δημοτικότητα μεταξύ των προγραμματιστών που αναζητούν μια ισχυρή και οπτικά ελκυστική λύση για την κατασκευή πινάκων διαχείρισης, καθιστώντας το πολύτιμο περιουσιακό στοιχείο στην εργαλειοθήκη των επαγγελματιών ανάπτυξης ιστοσελίδων.

### **2.6.1.2 Font Awesome**

Το FontAwesome, που αλλάζει τα δεδομένα στο σχεδιασμό ιστοσελίδων, προσφέρει μια τεράστια συλλογή από κλιμακούμενα διανυσματικά εικονίδια που αναβαθμίζουν την οπτική ελκυστικότητα και τη λειτουργικότητα των ιστοσελίδων. Το FontAwesome παρουσιάστηκε το 2012. Απλοποιεί την ενσωμάτωση εικονιδίων υψηλής ποιότητας σε έργα ιστού, εξαλείφοντας την ανάγκη για παραδοσιακά εικονίδια που βασίζονται σε εικόνες. Αυτή η εργαλειοθήκη ανοιχτού κώδικα παρέχει μια ευρεία σειρά εικονιδίων για διάφορους σκοπούς, από κοινά στοιχεία UI έως σύμβολα κοινωνικών μέσων. Με τα προσαρμόσιμα στυλ, τη στοίβαξη εικονιδίων και τα χαρακτηριστικά προσβασιμότητας, το FontAwesome δίνει τη δυνατότητα στους σχεδιαστές και τους προγραμματιστές να

βελτιώσουν την αισθητική και την εμπειρία χρήσης των ιστότοπων σε διαφορετικές συσκευές, εξασφαλίζοντας μια απρόσκοπτη και οπτικά συναρπαστική παρουσίαση του περιεχομένου [18].

## 2.6.2 jQuery

Η jQuery αποτελεί μια πρωτοποριακή βιβλιοθήκη JavaScript που παρουσιάστηκε το 2006, επηρεάζοντας βαθιά τις πρακτικές ανάπτυξης ιστοσελίδων. Η κύρια δύναμή της έγκειται στην ικανότητά της να απλοποιεί πολύπλοκες εργασίες μέσω μιας διαισθητικής και συνοπτικής σύνταξης, διευκολύνοντας τη συμβατότητα μεταξύ των προγραμμάτων περιήγησης και διευκολύνοντας τις προκλήσεις της ανάπτυξης ιστού. Η ικανότητα της jQuery επεκτείνεται στο χειρισμό του DOM, στο χειρισμό συμβάντων και στα αιτήματα AJAX, επιτρέποντας στους προγραμματιστές να δημιουργούν δυναμικές και ευέλικτες διαδικτυακές εφαρμογές χωρίς κόπο. Επιπλέον, η jQuery διαθέτει ένα εκτεταμένο οικοσύστημα πρόσθετων εφαρμογών που παρέχει στους προγραμματιστές πληθώρα πρόσθετων λειτουργιών. Οι δυνατότητες κινούμενης εικόνας που διαθέτει βελτιώνουν την εμπειρία του χρήστη, προσφέροντας μια απρόσκοπτη και οπτικά ελκυστική πινελιά στις ιστοσελίδες. Με την πάροδο των ετών, η jQuery έχει γίνει ένα απαραίτητο εργαλείο, απλοποιώντας τις ροές εργασίας ανάπτυξης και συμβάλλοντας στη δημιουργία διαδραστικών, πλούσιων σε χαρακτηριστικά ιστότοπων [19].

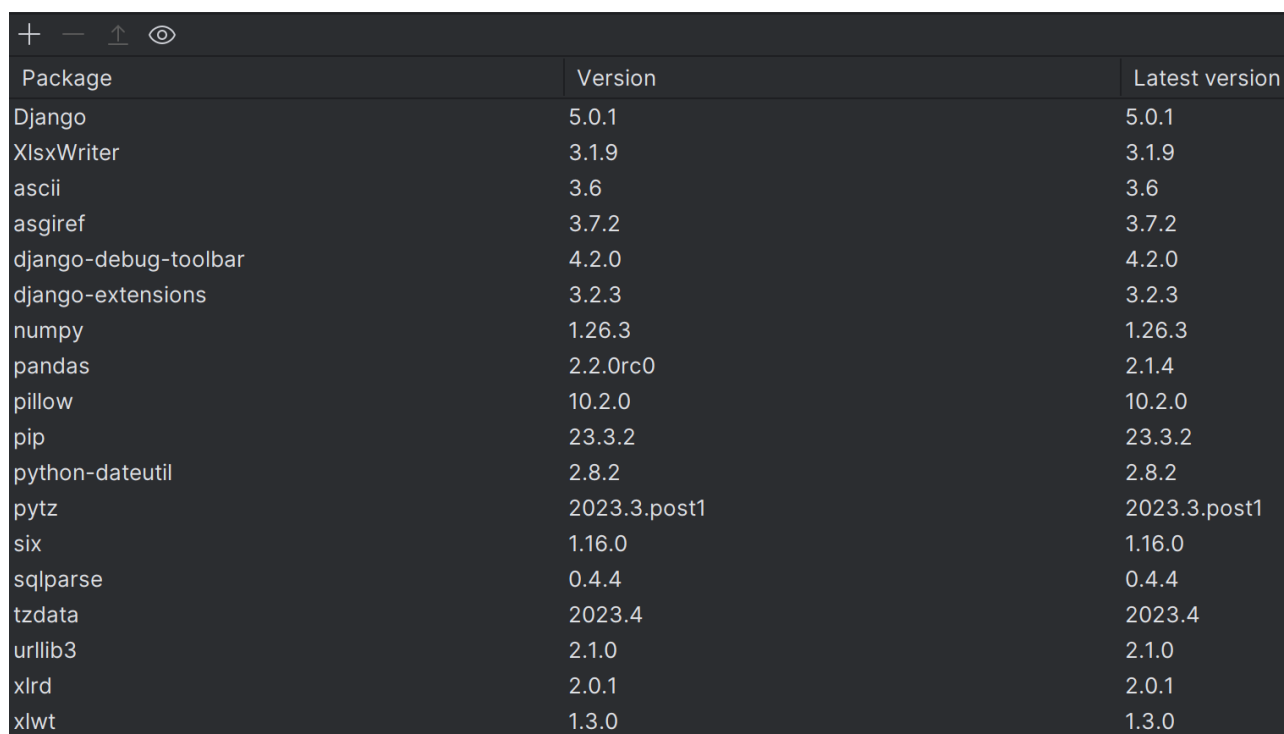
## 2.6.3 Chart.js

Το Chart.js είναι μια βιβλιοθήκη της JavaScript που ξεχωρίζει ως ένα ευέλικτο και φιλικό προς το χρήστη εργαλείο για τη δημιουργία διαδραστικών και οπτικά ελκυστικών διαγραμμάτων στο διαδίκτυο. Δημιουργημένο το 2013, το Chart.js απλοποιεί τη διαδικασία οπτικοποίησης δεδομένων, επιτρέποντας στους προγραμματιστές να αναπαριστούν σύνθετα σύνολα δεδομένων μέσω διαφόρων τύπων διαγραμμάτων απρόσκοπτα. Αυτή η βιβλιοθήκη έχει αποκτήσει ευρεία αποδοχή λόγω της ευκολίας χρήσης, των επιλογών προσαρμογής και της απόκρισης.

Στον πυρήνα της, το Chart.js υποστηρίζει γραμμικά διαγράμματα, ραβδογράμματα, κυκλικά διαγράμματα και διαγράμματα ραντάρ. Η απλότητά του έγκειται στη δηλωτική σύνταξή του, επιτρέποντας στους προγραμματιστές να ορίζουν τις διαμορφώσεις των διαγραμμάτων χωρίς κόπο. Η βιβλιοθήκη είναι συμβατή με τα σύγχρονα προγράμματα περιήγησης και μπορεί να ενσωματωθεί σε εφαρμογές ιστού.

Ένα από τα δυνατά σημεία του Chart.js είναι η απόκριση, εξασφαλίζοντας ότι τα διαγράμματα προσαρμόζονται με χάρη σε διαφορετικά μεγέθη οθόνης και συσκευές. Αυτό το χαρακτηριστικό του, σε συνδυασμό με την ελαφριά φύση της βιβλιοθήκης, το καθιστά πολύτιμο για τη δημιουργία φιλικών προς τα κινητά εφαρμογών και ελκυστικών οπτικοποιήσεων των δεδομένων.

## 2.6.4 Υπόλοιπες βιβλιοθήκες εφαρμογής



Package	Version	Latest version
Django	5.0.1	5.0.1
XlsxWriter	3.1.9	3.1.9
ascii	3.6	3.6
asgiref	3.7.2	3.7.2
django-debug-toolbar	4.2.0	4.2.0
django-extensions	3.2.3	3.2.3
numpy	1.26.3	1.26.3
pandas	2.2.0rc0	2.1.4
pillow	10.2.0	10.2.0
pip	23.3.2	23.3.2
python-dateutil	2.8.2	2.8.2
pytz	2023.3.post1	2023.3.post1
six	1.16.0	1.16.0
sqlparse	0.4.4	0.4.4
tzdata	2023.4	2023.4
urllib3	2.1.0	2.1.0
xlrd	2.0.1	2.0.1
xlwt	1.3.0	1.3.0

Σχήμα 2.3: Εγκατεστημένα πακέτα και βιβλιοθήκες της πλατφόρμας

Το σύνολο των διαθέσιμων πακέτων και βιβλιοθηκών που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος διαχείρισης των φοιτητών, φαίνονται στο σχήμα 2.3. Δεν χρησιμοποιήθηκαν απαραίτητα όλα τα εγκατεστημένα πακέτα, ωστόσο κάποια στιγμή κατά το σχεδιασμό και την ανάπτυξη χρησιμοποιήθηκαν όσα φαίνοντα στην εικόνα. Για την ορθή λειτουργία της πλατφόρμας, ο χρήστης θα πρέπει να εγκαταστήσει όλα αυτά τα πακέτα πρώτα στο περιβάλλον στο οποίο θα «τρέχει» την εφαρμογή. Δεν θα γίνει αναλυτική περιγραφή των συγκεκριμένων, καθώς αυτό ξεφεύγει από το σκοπό της παρούσας διπλωματικής εργασίας.

# Κεφάλαιο 3

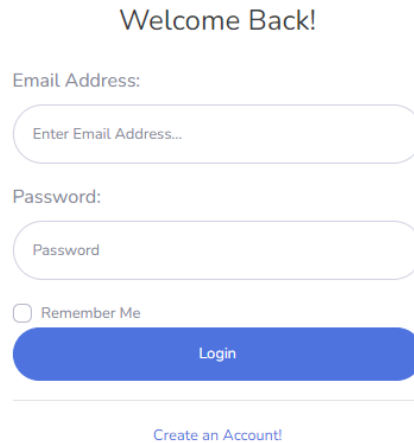
## Οδηγός χρήσης

### 3.1 Εγγραφή χρήστη

Για να τρέξουμε την εφαρμογή, θα πρέπει να έχουν εγκατασταθεί όλα τα προαπαιτούμενα πακέτα και βιβλιοθήκες που έχουν αναφερθεί στο δεύτερο κεφάλαιο. Σε ένα καινούριο τερματικό παράθυρο PowerShell (κατά προτίμηση σε αυτό του IDE), πηγαίνουμε αρχικά στο φάκελο της πλατφόρμας με την εντολή "cd" (change directory) και στη συνέχεια συμπληρώνουμε την ορθή διαδρομή (path) του φακέλου στον οποίον βρίσκεται το αρχείο `manage.py` της εφαρμογής. Εφόσον βρισκόμαστε στη σωστή θέση, για να "τρέξουμε" τον σέρβερ γράφουμε στο τερματικό παράθυρο την εντολή "python manage.py runserver" και πατάμε enter. Το σύστημα (αν δεν έχει παραμετροποιηθεί), από προεπιλογή του θα τρέξει την εφαρμογή στη θέση localhost (127.0.0.1) στη θύρα 8000. Με το που πάμε στην τοποθεσία localhost (<http://127.0.0.1:8000/>), θα μας ζητηθεί να κάνουμε καινούριο λογαριασμό (account). Η οθόνη πρέπει να έχει το σχήμα του σχήματος 3.1. Πατάμε στο κουμπί "Create an Account!" για να συνεχίσουμε. Στο πλαίσιο που μας εμφανίζεται συμπληρώνουμε τα εξής πεδία.

- **First Name** (Όνομα)
- **Last Name** (Επίθετο)
- **Email Address** (Διεύθυνση email). Σε αυτό το πεδίο θα γίνει έλεγχος για το αν η διεύθυνση email έχει τη σωστή μορφή, ωστόσο δεν θα σταλεί σύνδεσμος για επιβεβαίωση στο χρήστη κατά την ολοκλήρωση της εγγραφής
- **Username** (Όνομα χρήστη)
- **Password** (Κωδικός)
- **Repeat Password** (Επανάληψη Κωδικού)

Αφού συμπληρώσουμε τα πεδία, πατάμε το μπλε κουμπί "Register Account" και ο χρήστης έχει δημιουργηθεί επιτυχώς. Η σελίδα μας θα πρέπει να μοιάζει με τη σελίδα του σχήματος 3.2.



Welcome Back!

Email Address:

Password:

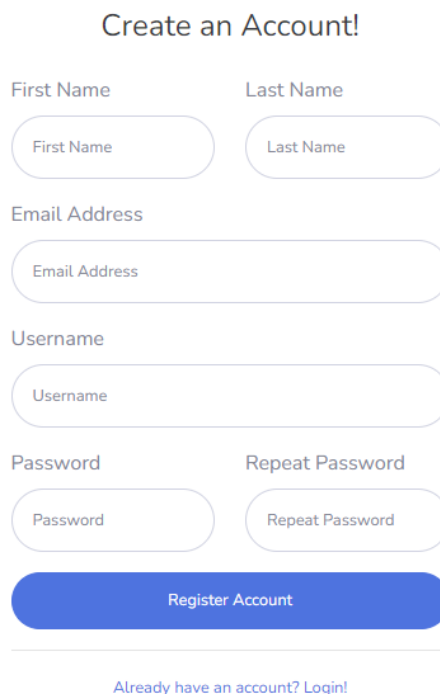
Remember Me

Login

---

[Create an Account!](#)

Σχήμα 3.1: Είσοδος ή δημιουργία νέου χρήστη



Create an Account!

First Name      Last Name

Email Address

Username

Password      Repeat Password

Register Account

---

[Already have an account? Login!](#)

Σχήμα 3.2: Δημιουργία νέου χρήστη

## 3.2 Είσοδος χρήστη

Με τη δημιουργία νέου χρήστη, επιστρέφουμε αυτόματα στη σελίδα του σχήματος 3.1 και πλέον μπορούμε να κάνουμε είσοδο στο σύστημά μας. Συμπληρώνουμε τα στοιχεία μας, επιλέγουμε την επιλογή "Remember Me" αν θέλουμε να παραμένουμε συνδεδεμένοι με τα στοιχεία του λογαριασμού σε μελλοντικές συνεδρίες και πατάμε στο μπλε κουμπί "Login" (Είσοδο).

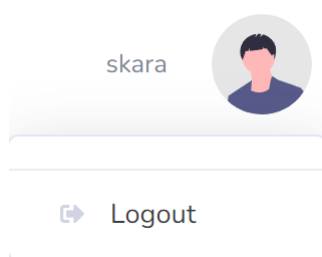
## 3.3 Πλατφόρμα διαχείρισης φοιτητών

### 3.3.1 Αρχική σελίδα

Με το που πραγματοποιήσουμε είσοδο στην διαδικτυακή εφαρμογή, ανακατευθυνόμαστε στην αρχική μας σελίδα, όπως αυτή φαίνεται στο σχήμα 3.4. Η σελίδα αποτελείται από τρία βασικά μέρη.

Στο αριστερό μέρος φαίνεται το "Dashboard" (Πίνακας εφαρμογών / Ταμπλό), μια μπλέ στήλη που εκτείνεται μέχρι το τέλος της σελίδας και περιλαμβάνει τα τμήματα "Upload" και "Download" για τα οποία θα γίνει αναφορά αργότερα.

Στο πάνω μέρος της σελίδας υπάρχει η άσπρη μπάρα που εκτείνεται οριζόντια σε όλη την σελίδα, όπου πάνω δεξιά υπάρχουν πληροφορίες για το χρήστη καθώς και η επιλογή να κάνει "Logout" (Εξοδο) από την εφαρμογή.



Σχήμα 3.3: Έξοδος χρήστη από το σύστημα

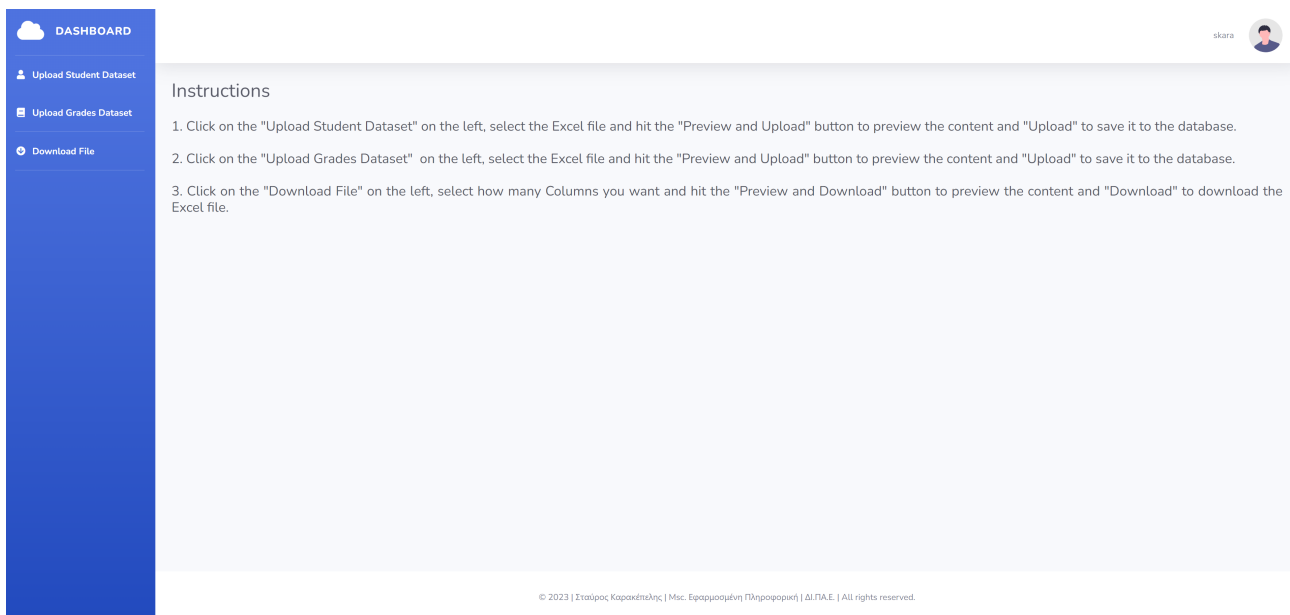
Στο κεντρικό μέρος της σελίδας, κάτω από την άσπρη μπάρα, υπάρχει το κυρίως περιεχόμενο της σελίδας. Συγκεκριμένα, στην αρχική σελίδα ακολουθούν αναλυτικές οδηγίες για τη χρήση της εφαρμογής, ένας απλός οδηγός τριών βημάτων που μετράφεται ως εξής:

1. Επιλέξτε το "Upload Student Dataset" από τα αριστερά, επιλέξτε το αντίστοιχο αρχείο Excel και πατήστε το κουμπί "Preview and Upload" για να κάνετε προεπι-

σκόπηση των περιεχομένων και στη συνέχεια πατήστε "Upload" για να περάσετε τα δεδομένα στη βάση δεδομένων.

2. Επιλέξτε το "Upload Grades Dataset" από τα αριστερά, επιλέξτε το αντίστοιχο αρχείο Excel και πατήστε το κουμπί "Preview and Upload" για να κάνετε προεπισκόπηση των περιεχομένων και στη συνέχεια πατήστε "Upload" για να περάσετε τα δεδομένα στη βάση δεδομένων.
3. Επιλέξτε το "Download File" από τα αριστερά, επιλέξτε τις στήλες που θέλετε να κρατήσετε και πατήστε το κουμπί "Preview and Download" για να κάνετε προεπισκόπηση των περιεχομένων και στη συνέχεια πατήστε "Download" για να κατεβάσετε το τελικό αρχείο Excel.

Τέλος, στο κάτω μέρος της σελίδας υπάρχει το "footer" της σελίδας με πληροφορίες σχετικά με τα πνευματικά δικαιώματα της εφαρμογής.



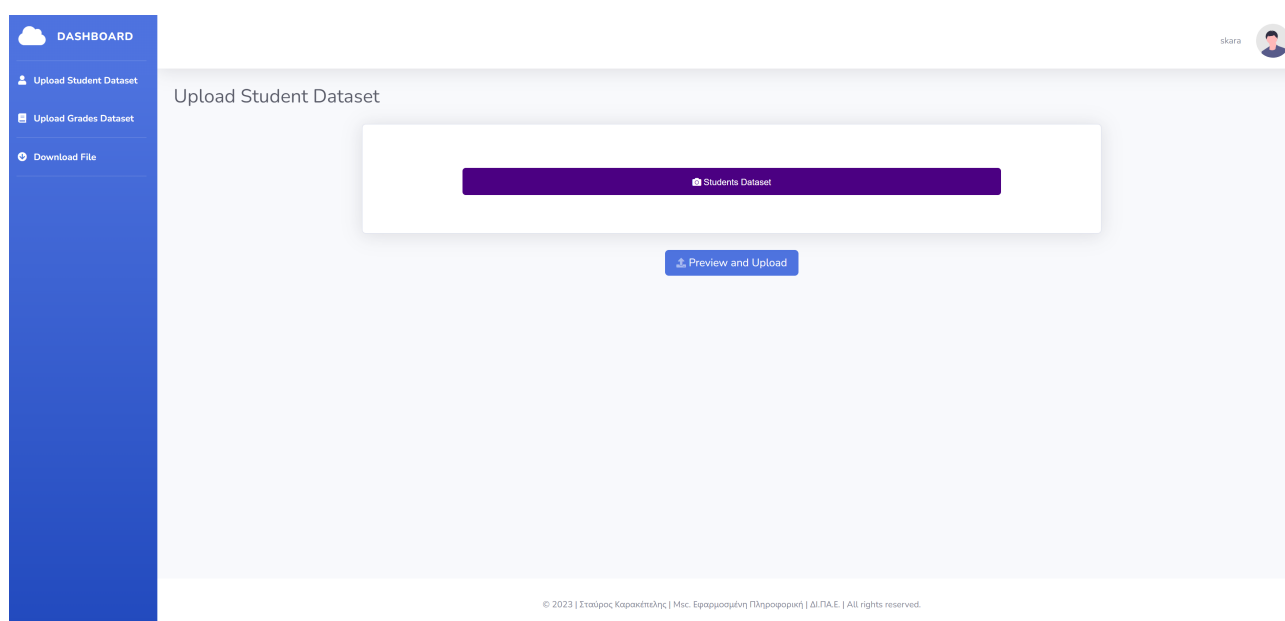
Σχήμα 3.4: Αρχική σελίδα

### 3.3.2 Αρχείο φοιτητών

Σε αυτό το σημείο, θα πρέπει να αναφέρουμε ότι η βάση δεδομένων είναι σε αυτό το σημείο κενή και περιέχει πληροφορίες μόνο για το χρήστη που δημιουργήσαμε, όχι για τους μαθητές και για τις βαθμολογίες που αποτελούν τα κύρια δεδομένα της εφαρμογής. Για αυτό το λόγο, θα πρέπει να περάσουμε τις απαραίτητες πληροφορίες στη βάση ώστε να πάρουμε το τελικό αρχείο με τα ζητούμενα δεδομένα.



Πηγαίνουμε στο αριστερό μέρος του "Dashboard" και κάνουμε κλικ στην επιλογή "Upload Student Dataset". Αμέσως, θα ανακατευθυνθούμε στη διεύθυνση "http://127.0.0.1:8000/dashboard/upload/students" και η σελίδα μας θα έχει την μορφή του σχήματος 3.5. Πατάμε στο κουμπί "Students Dataset" στο μωβ πλαίσιο και διαλέγουμε το αρχείο με τις πληροφορίες των φοιτητών, όπως αυτό έχει εξαχθεί από τη βάση δεδομένων του ΠΜΣ "Εφαρμοσμένη Πληροφορική" σε μορφή .xls, χωρίς να έχει τροποποιηθεί ο αριθμός των στηλών. Το κουμπί μπορεί να δέχεται σαν είσοδο ένα αρχείο Excel τη φορά (single file input) και δεν υποστηρίζει τη μαζική επεξεργασία αρχείων. Με το που επιλέξουμε το κατάλληλο αρχείο, το κουμπί θα γίνει πράσινο και το κείμενο "File Selected" (αρχείο επιλέχθη) θα εμφανιστεί.



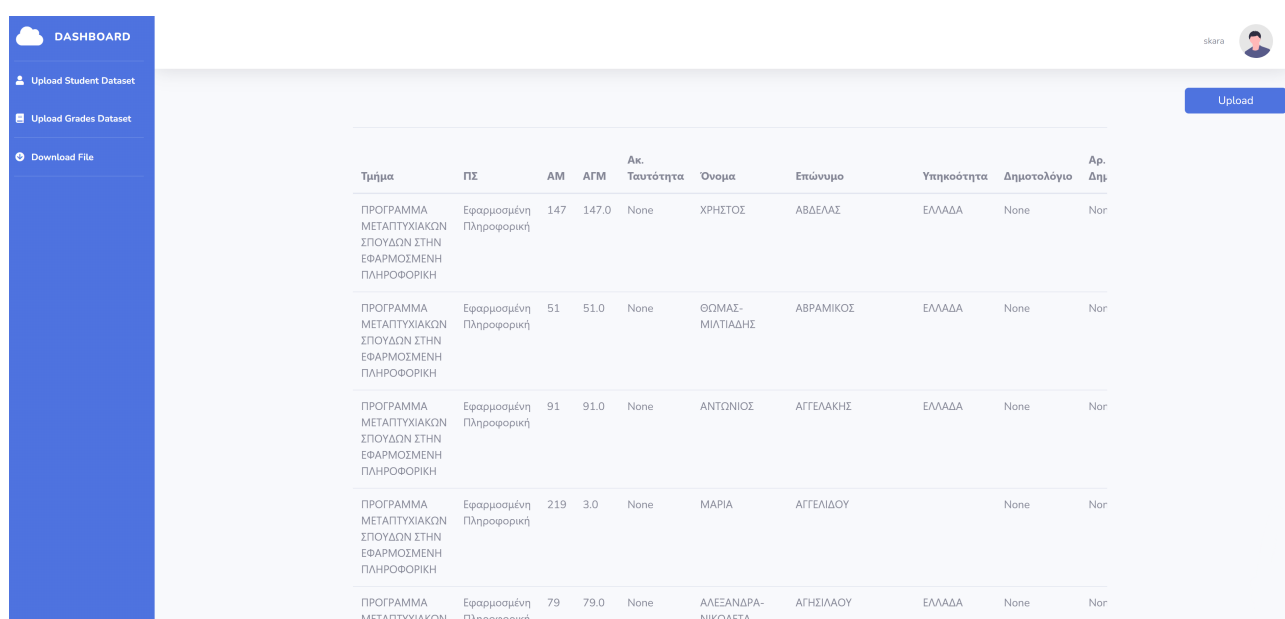
Σχήμα 3.5: Σελίδα upload students dataset

Στη συνέχεια, πατάμε το κουμπί "Preview and Upload" οπού μπορούμε να δούμε μια προεπισκόπηση των δεδομένων του αρχείου μας και πώς αυτά θα περαστούν στη βάση δεδομένων και στη συνέχεια πατάμε το μπλε κουμπί "Upload" που βρίσκεται πάνω δεξιά, ώστε αυτά να περαστούν στη βάση SQLite3. Μετά το πέρας της διαδικασίας, θα μας εμφανιστεί το μήνυμα "Click here. File uploaded successfully!". Σε αυτό το σημείο η διαδικασία με τα στοιχεία των φοιτητών τελειώσε, μπορούμε να πατήσουμε στο "Click here" και ο υπερσύνδεσμος θα μας επιστρέψει στην αρχική σελίδα ή να πατήσουμε αριστερά στο πάνω μέρος του "Dashboard" (στο συννεφάκι).

Ο τρόπος με τον οποίο λειτουργεί ο αλγόριθμος της εγγραφής είναι ότι αρχικά ελέγχει αν υπάρχει εγγραφή του συγκεκριμένου φοιτητή στη βάση από το αρχείο που του ανεβάσαμε

1. Αν δεν υπάρχει εγγραφή, τότε δημιουργεί τον φοιτητή στη βάση με τις πληροφορίες από το αρχείο που του δώσαμε.
2. Αν υπάρχει εγγραφή, τότε θα πάει να βρει την εγγραφή του φοιτητή στη βάση και θα πάει να ενημερώσει τα πεδία του φοιτητή με τις πληροφορίες από το αρχείο που του δώσαμε.

Περισσότερες πληροφορίες σχετικά με τη συνάρτηση του αλγορίθμου μπορεί να βρει κανείς στο παράρτημα, στη συνάρτηση A'.2.3 Update/Create Students.



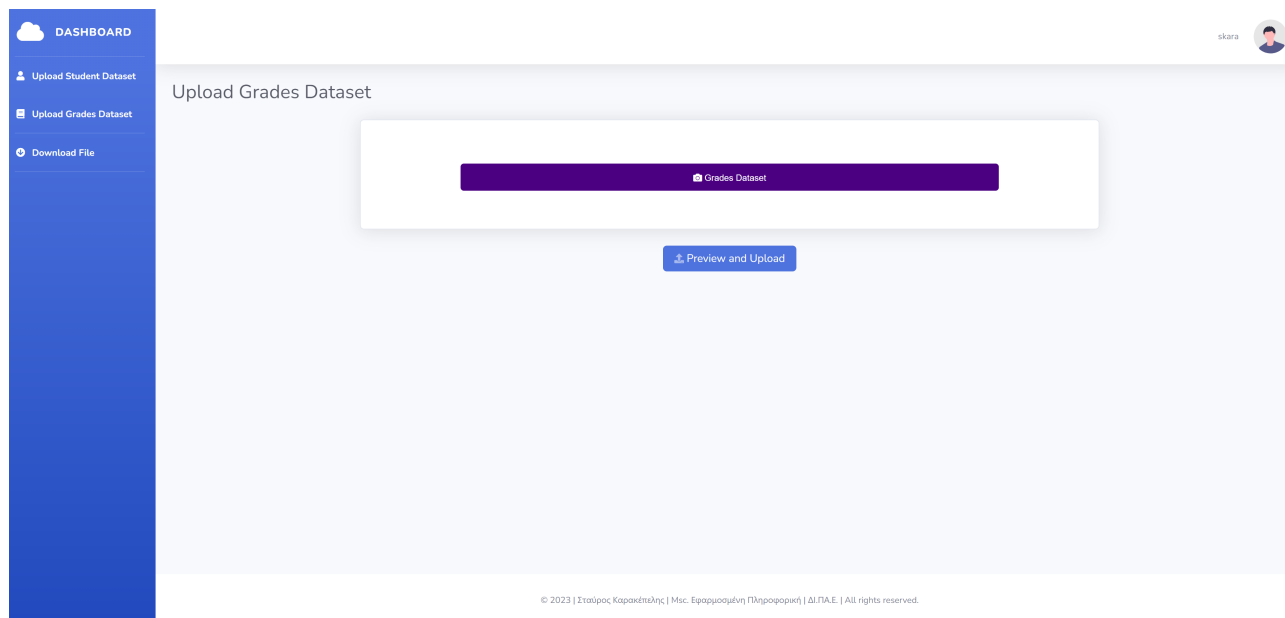
Τμήμα	ΠΣ	AM	AGM	Ακ. Ταυτότητα	Όνομα	Επώνυμο	Υπηκοότητα	Δημοτολόγιο	Αρ. Δημ.
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ	Εφαρμοσμένη Πληροφορική	147	147.0	None	ΧΡΗΣΤΟΣ	ΑΒΔΕΛΑΣ	ΕΛΛΑΔΑ	None	Nor
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ	Εφαρμοσμένη Πληροφορική	51	51.0	None	ΘΩΜΑΣ-ΜΙΛΤΙΑΔΗΣ	ΑΒΡΑΜΙΚΟΣ	ΕΛΛΑΔΑ	None	Nor
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ	Εφαρμοσμένη Πληροφορική	91	91.0	None	ΑΝΤΩΝΙΟΣ	ΑΓΓΕΛΑΚΗΣ	ΕΛΛΑΔΑ	None	Nor
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ	Εφαρμοσμένη Πληροφορική	219	3.0	None	ΜΑΡΙΑ	ΑΓΓΕΛΙΔΟΥ		None	Nor
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ	Εφαρμοσμένη Πληροφορική	79	79.0	None	ΑΛΕΞΑΝΔΡΑ-ΝΙΚΟΛΕΤΑ	ΑΓΗΣΙΛΑΟΥ	ΕΛΛΑΔΑ	None	Nor

Σχήμα 3.6: Σελίδα preview students dataset

### 3.3.3 Αρχείο βαθμολογιών

Πηγαίνουμε στο αριστερό μέρος του "Dashboard" και κάνουμε κλικ στην επιλογή "Upload Grades Dataset". Αμέσως, θα ανακατευθυνθούμε στη διεύθυνση "http://127.0.0.1:8000/dashboard/upload/grades" και η σελίδα μας θα έχει τη μορφή του σχήματος 3.7. Πατάμε στο κουμπί "Grades Dataset" στο μωβ πλαίσιο και διαλέγουμε τα αρχεία με τις πληροφορίες των βαθμών των μαθημάτων των φοιτητών, όπως αυτό έχει εξαχθεί από τη βάση δεδομένων του ΠΜΣ "Εφαρμοσμένη Πληροφορική" σε μορφή .xls, χωρίς να έχει τροποποιηθεί ο αριθμός των στηλών. Το κουμπί μπορεί να δέχεται σαν είσοδο πολλα-

πλά αρχεία Excel τη φορά (multiple file input) και υποστηρίζει τη μαζική επεξεργασία αρχείων. Κάθε αρχείο περιέχει πληροφορία για ένα μάθημα. Με το που επιλέξουμε τα κατάλληλα αρχεία, το κουμπί θα γίνει πράσινο και το κείμενο "File(s) Selected" (αρχεία επιλέχθη) θα εμφανιστεί.



Σχήμα 3.7: Σελίδα upload grades dataset

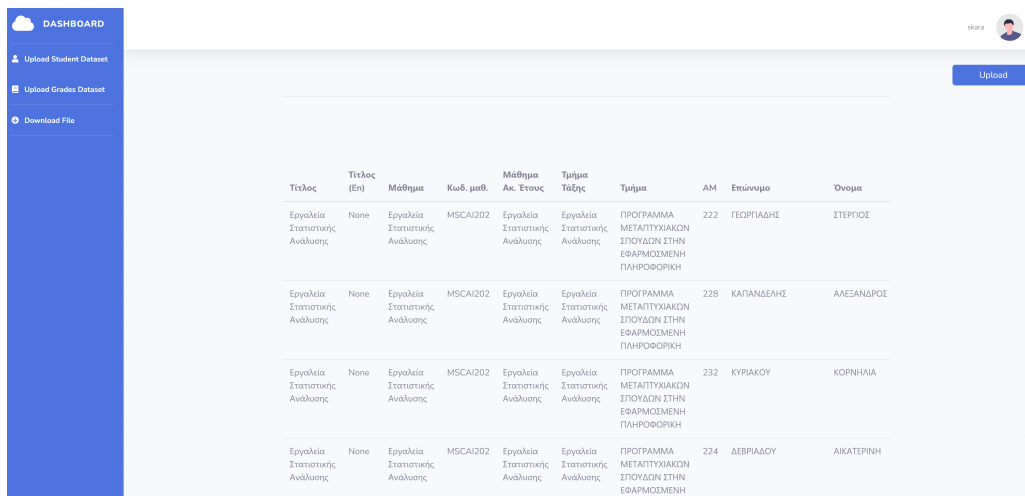
Στη συνέχεια, πατάμε το κουμπί "Preview and Upload" οπού μπορούμε να δούμε μια προεπισκόπηση των δεδομένων του αρχείου μας και πώς αυτά θα περαστούν στη βάση δεδομένων και στη συνέχεια πατάμε το μπλε κουμπί "Upload" που βρίσκεται πάνω δεξιά, ώστε αυτά να περαστούν στη βάση SQLite3. Μετά το πέρας της διαδικασίας, θα μας εμφανιστεί το μήνυμα "Click here. File(s) uploaded successfully!". Σε αυτό το σημείο η διαδικασία με τα στοιχεία με τις βαθμολογίες των φοιτητών τελειώσει, μπορούμε να πατήσουμε στο "Click here" και ο υπερσύνδεσμος θα μας επιστρέψει στην αρχική σελίδα ή να πατήσουμε αριστερά στο πάνω μέρος του "Dashboard" (στο συννεφάκι).

Ο τρόπος με τον οποίο λειτουργεί ο αλγόριθμος της εγγραφής είναι ότι αρχικά ελέγχει αν υπάρχει εγγραφή του συγκεκριμένου φοιτητή και του συγκεκριμένου μαθήματος στη βάση από τα αρχεία που του ανεβάσαμε:

1. Αν δεν υπάρχει εγγραφή, τότε δημιουργεί τον φοιτητή στη βάση με τις πληροφορίες από το αρχείο που του δώσαμε για κάθε μάθημα.
2. Αν υπάρχει εγγραφή, τότε θα πάει να βρει την εγγραφή του φοιτητή στη βάση για κάθε μάθημα και θα πάει να ενημερώσει τα πεδία του φοιτητή με τις πληροφορίες

από το αρχείο που του δώσαμε.

Περισσότερες πληροφορίες σχετικά με τη συνάρτηση του αλγορίθμου μπορεί να βρει κανείς στο παράρτημα, στη συνάρτηση Α'.2.4 Update/Create Grades.

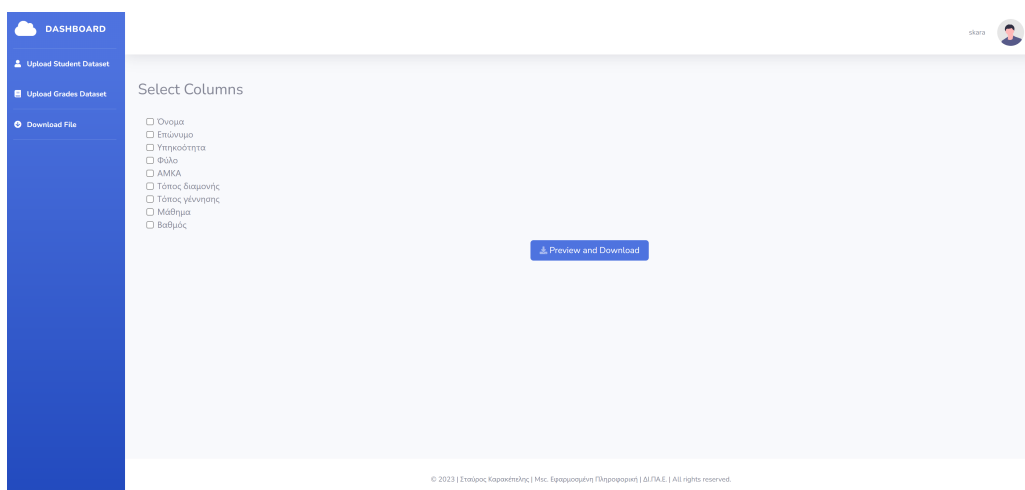


Τίτλος	Τίτλος (Επ)	Μάθημα	Κωδ. μαθ.	Μήθημα Ακ. Έτους	Τμήμα Τάξης	Τμήμα	AM	Επίσημο	Όνομα
Εργασία Στατιστικής Ανάλυσης	None	Εργασία Στατιστικής Ανάλυσης	MSCAI202	Εργασία Στατιστικής Ανάλυσης	Εργασία Στατιστικής Ανάλυσης	ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ	222	ΓΕΩΡΓΙΑΔΗΣ	ΣΤΕΡΓΙΟΣ
Εργασία Στατιστικής Ανάλυσης	None	Εργασία Στατιστικής Ανάλυσης	MSCAI202	Εργασία Στατιστικής Ανάλυσης	Εργασία Στατιστικής Ανάλυσης	ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ	228	ΚΑΠΑΝΔΕΛΗΣ	ΑΛΕΞΑΝΔΡΟΣ
Εργασία Στατιστικής Ανάλυσης	None	Εργασία Στατιστικής Ανάλυσης	MSCAI202	Εργασία Στατιστικής Ανάλυσης	Εργασία Στατιστικής Ανάλυσης	ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ	232	ΚΥΡΙΑΚΟΥ	ΚΟΡΗΝΙΑ
Εργασία Στατιστικής Ανάλυσης	None	Εργασία Στατιστικής Ανάλυσης	MSCAI202	Εργασία Στατιστικής Ανάλυσης	Εργασία Στατιστικής Ανάλυσης	ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ	224	ΔΕΒΡΙΑΔΟΥ	ΑΙΚΑΤΕΡΙΝΗ

Σχήμα 3.8: Σελίδα preview grades dataset

### 3.3.4 Αποθήκευση τελικού αρχείου

Εφόσον έχουμε ανεβάσει πληροφορίες τόσο για τους φοιτητές, όσο και για τα μαθήματα, πηγαίνουμε στο αριστερό μέρος του "Dashboard" και κάνουμε κλικ στην επιλογή "Download file" (Κατέβασμα αρχείου). Αμέσως, θα ανακατευθυνθούμε στη διεύθυνση "http://127.0.0.1:8000/dashboard/download" και η σελίδα μας θα έχει την μορφή του σχήματος 3.9. Στο αριστερό μέρος της σελίδας βλέπουμε τις στήλες που μπορούν να υπάρχουν στο τελικό μας αρχείο. Υπάρχουν οι ακόλουθες επιλογές:



Select Columns

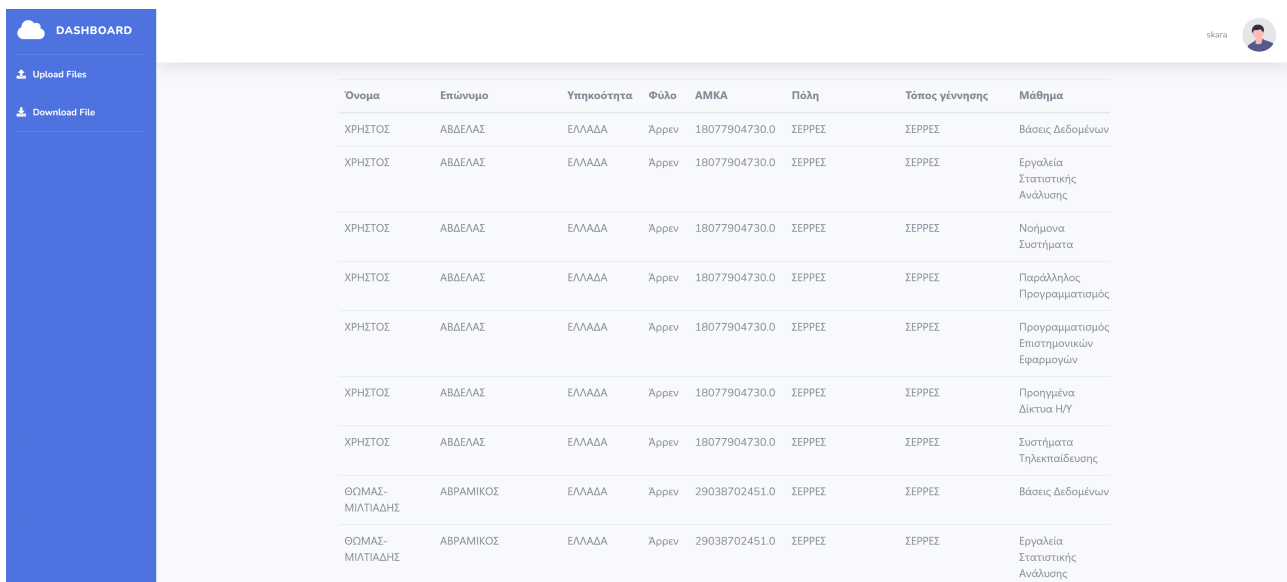
- Όνομα
- Επίσημο
- Υπερκοίτητα
- Φύλο
- Διεύθ
- Τόπος διαμονής
- Τόπος γέννησης
- Μάθημα
- Βαθμός

[Preview and Download](#)

Σχήμα 3.9: Σελίδα download

- Όνομα
- Επώνυμο
- Υπηκοότητα
- Φύλο
- ΑΜΚΑ
- Τόπος διαμονής
- Τόπος καταγωγής
- Μάθημα
- Βαθμός

Οι επιλογές είναι της μορφής checkbox, που σημαίνει με ένα απλό κλικ μπορούμε να επιλέξουμε ή να αποεπιλέξουμε μια στήλη. Μόλις αποφασίσουμε ποιές στήλες θέλουμε να κρατήσουμε, πατάμε στο μπλε κουμπί "Preview and Download" (Προεπισκόπηση και Κατέβασμα) για να δούμε μια προεπισκόπηση του τελικού μας αρχείου και να επιβεβαιώσουμε την ορθότητα των δεδομένων που θέλουμε να αποθηκεύσουμε πριν κατεβάσουμε το αρχείο. Η σελίδα μας θα πρέπει να έχει τη μορφή του σχήματος 3.10. Εφόσον βεβαιωθούμε για τα δεδομένα του αρχείου, πηγαίνουμε στο τέλος της σελίδας και πατάμε το μπλε κουμπί "Download" και το αρχείο θα αποθηκευτεί τοπικά στον υπολογιστή μας.



Όνομα	Επώνυμο	Υπηκοότητα	Φύλο	ΑΜΚΑ	Πόλη	Τόπος γέννησης	Μάθημα
ΧΡΗΣΤΟΣ	ΑΒΔΕΛΑΣ	ΕΛΛΑΔΑ	Άρρεν	18077904730.0	ΣΕΡΡΕΣ	ΣΕΡΡΕΣ	Βάσεις Δεδομένων
ΧΡΗΣΤΟΣ	ΑΒΔΕΛΑΣ	ΕΛΛΑΔΑ	Άρρεν	18077904730.0	ΣΕΡΡΕΣ	ΣΕΡΡΕΣ	Εργαλεία Στατιστικής Ανάλυσης
ΧΡΗΣΤΟΣ	ΑΒΔΕΛΑΣ	ΕΛΛΑΔΑ	Άρρεν	18077904730.0	ΣΕΡΡΕΣ	ΣΕΡΡΕΣ	Νοήμονα Συστήματα
ΧΡΗΣΤΟΣ	ΑΒΔΕΛΑΣ	ΕΛΛΑΔΑ	Άρρεν	18077904730.0	ΣΕΡΡΕΣ	ΣΕΡΡΕΣ	Παράλληλος Προγραμματισμός
ΧΡΗΣΤΟΣ	ΑΒΔΕΛΑΣ	ΕΛΛΑΔΑ	Άρρεν	18077904730.0	ΣΕΡΡΕΣ	ΣΕΡΡΕΣ	Προγραμματισμός Επιστημονικών Εφαρμογών
ΧΡΗΣΤΟΣ	ΑΒΔΕΛΑΣ	ΕΛΛΑΔΑ	Άρρεν	18077904730.0	ΣΕΡΡΕΣ	ΣΕΡΡΕΣ	Προηγμένα Δίκτυα ΗΥ
ΧΡΗΣΤΟΣ	ΑΒΔΕΛΑΣ	ΕΛΛΑΔΑ	Άρρεν	18077904730.0	ΣΕΡΡΕΣ	ΣΕΡΡΕΣ	Συστήματα Τηλεκπαίδευσης
ΘΩΜΑΣ-ΜΙΑΤΙΑΔΗΣ	ΑΒΡΑΜΙΚΟΣ	ΕΛΛΑΔΑ	Άρρεν	29038702451.0	ΣΕΡΡΕΣ	ΣΕΡΡΕΣ	Βάσεις Δεδομένων
ΘΩΜΑΣ-ΜΙΑΤΙΑΔΗΣ	ΑΒΡΑΜΙΚΟΣ	ΕΛΛΑΔΑ	Άρρεν	29038702451.0	ΣΕΡΡΕΣ	ΣΕΡΡΕΣ	Εργαλεία Στατιστικής Ανάλυσης

Σχήμα 3.10: Σελίδα preview and download

## 3.4 Γενικές παρατηρήσεις

Πριν ολοκληρώσουμε τον οδηγό, θα πρέπει να αναφέρουμε μερικά πράγματα που είναι χρήσιμο να γνωρίζει ο χρήστης για μελλοντική χρήση, συντήρηση και τρόπο λειτουργίας της πλατφόρμας.

Αρχικά, ο τρόπος με τον οποίο οι εγγραφές γράφονται στη βάση είναι με μία μέθοδο `for`, η οποία διαβάζει όλα τα στοιχεία ενός πίνακα και είτε αποθηκεύει την έγγραφη είτε την ενημερώνει. Ο αλγόριθμος είναι γραμμικός και τρέχει σε χρόνο  $O(n)$ . Σε αυτό το σημείο, να τονίσουμε ξανά ότι η σχέση φοιτητών/μαθημάτων είναι 1:N (ένας φοιτητής μπορεί να έχει πολλά μαθήματα ταυτόχρονα). Όπως εύκολα μπορεί να συμπεράνει κανείς, όταν μεταφορτώσουμε πολλαπλά μαθήματα βαθμολογιών για πολλούς φοιτητές και για πολλές χρονολογίες, είναι πολύ εύκολο το μέγεθος εγγραφών να ξεφύγει πολύ εύκολα. Αυτό θα οδηγήσει σε μεγάλο χρόνο εκτέλεσης του προγράμματος και σε πιθανό σφάλμα (`runtime error`). Επίσης, ο χρόνος εκτέλεσης του προγράμματος εξαρτάται σε μεγάλο βαθμό και από τα χαρακτηριστικά του συστήματος (`hardware`). Για αυτόν το λόγο, συνίσταται οι συνολικές εγγραφές στη βάση να μην ξεπερνούν τις 1000-1500 τη φορά.

Θα παρατηρήσετε ότι στην προεπισκόπηση πριν την αποθήκευση του τελικού αρχείου, το κουτάκι με την επιλογή "Μάθημα" αντιστοιχεί στη στήλη του μαθήματος και όχι στην επιλογή του. Δηλαδή, ο τρόπος που επιλέγετε ποιά μαθήματα θα περιέχονται στο τελικό αρχείο γίνεται μέσα από το "Upload Grades Dataset". Επειδή η βάση κρατάει τις πληροφορίες και μετά το πέρας της εφαρμογής, αν χρειαστεί ο χρήστης να τρέξει μελλοντικά το πρόγραμμα, θα πρέπει να θυμάται ότι ήδη υπάρχουν αποθηκευμένα κάποια μαθήματα στη βάση. Αν αυτά δεν τα θέλει, τότε θα πρέπει να "καθαρίσει" τα περιεχόμενα της βάσης και πιο συγκεκριμένα, τον πίνακα "StudentGrades". Αυτό μπορεί να επιτευχθεί με δύο τρόπους.

1. Με κάποιο λογισμικό διαχείρισης βάσης δεδομένων, όπως είναι το SQLite-Studio.
2. Με `scripts` της `python`, όπως το B'.1 `DeleteStudents.py`, B'.2 `DeleteStudentGrades.py`, και B'.3 `DeleteDB.py`, στα οποία γίνεται αναφορά στο παράρτημα B'.

Τέλος, η εφαρμογή βρίσκεται σε πειραματικό στάδιο και δεν είναι έτοιμη για εμπορική χρήση (`production-ready`). Πέρα από τις πολλαπλές δοκιμές που θα πρέπει να υποστεί και να δοκιμαστεί σε διαφορετικές καταστάσεις, θα πρέπει να γίνει σωστή ρύθμιση και παραμετροποίηση του `σέρβερ` που θα "τρέχει" την πλατφόρμα, για να μπορεί να διαχειρίζεται κατάλληλα τα αρχεία (στη δική μας περίπτωση `Excel` αρχεία) και

να τα αποθηκεύει στην καθορισμένη διαδρομή. Θα πρέπει, δηλαδή, να φύγουμε από την κατάσταση debug που έχουμε ορίσει στα αρχεία settings.py και urls.py αντίστοιχα, όπως αυτή φαίνεται παρακάτω:

```
#settings.py
# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

#urls.py
if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Αυτή τη στιγμή, ο σέρβερ αποθηκεύει τα αρχεία μας στη διαδρομή της εφαρμογής `\files\merged\student_and_grades.xlsx`, ωστόσο είναι πολύ πιθανόν τα αρχεία μας να αποθηκεύονται και στις Λήψεις του υπολογιστή (Download), πράγμα μη θεμιτό. Για αυτό, πέρα από την παραμετροποίηση του σέρβερ, πολύ πιθανόν να χρειαστεί να ρυθμίσουμε και το πρόγραμμα περιήγησης (browser), ώστε να μην αλλάζει την προεπιλεγμένη τοποθεσία αποθήκευσης του αρχείου της εφαρμογής με αυτήν που έχει ορίσει ο ίδιος ο περιηγητής.

# Κεφάλαιο 4

## Συμπεράσματα

### 4.1 Σύνοψη

Ανακεφαλαιώνοντας, η εφαρμογή ενός συστήματος διαχείρισης φοιτητών σηματοδοτεί ένα ταξίδι στο χώρο της εκπαίδευσης, το οποίο αποφέρει πληθώρα θετικών αποτελεσμάτων. Καθώς τα εκπαιδευτικά ιδρύματα αγκαλιάζουν ολοένα και περισσότερο αυτά τα συστήματα, τα συμπεράσματα που προκύπτουν από την αξιοποίησή τους υπογραμμίζουν τον ζωτικό τους ρόλο στη διαμόρφωση αποτελεσματικών, διαφανών και μαθητοκεντρικών περιβαλλόντων μάθησης.

Η ενσωμάτωση ενός τέτοιου συστήματος εξορθολογίζει τις διοικητικές διαδικασίες, μειώνοντας το χειροκίνητο φόρτο εργασίας και ελαχιστοποιώντας τα λάθη. Η αυτοματοποίηση εργασιών ενισχύει την αποδοτικότητα και διασφαλίζει την ακρίβεια και τη συνέπεια των αρχείων. Αυτό σημαίνει ότι οι εκπαιδευτικοί μπορούν να αφιερώσουν περισσότερο χρόνο στη διδασκαλία και την υποστήριξη των μαθητών.

Οι δυνατότητες λήψης αποφάσεων βάσει δεδομένων του συστήματος διαχείρισης φοιτητών συμβάλλουν σε ενημερωμένες εκπαιδευτικές πρακτικές. Το συμπέρασμα που εξάγεται είναι ότι με την ανάλυση των δεδομένων που σχετίζονται με τις επιδόσεις, τη φοίτηση και τη συμπεριφορά των μαθητών, τα ιδρύματα μπορούν να προβούν σε στοχευμένες παρεμβάσεις για την ενίσχυση των διδακτικών μεθοδολογιών και τη βελτίωση των συνολικών μαθησιακών αποτελεσμάτων. Οι γνώσεις που αποκτώνται από την ανάλυση δεδομένων δίνουν τη δυνατότητα στα ιδρύματα να προσαρμόζονται και να ανταποκρίνονται στις εξελισσόμενες ανάγκες των φοιτητών τους.

Συμπερασματικά, η υιοθέτηση των συστημάτων διαχείρισης μαθητών επιφέρει πληθώρα θετικών αποτελεσμάτων για τα εκπαιδευτικά ιδρύματα. Από τον εξορθολογισμό



των διοικητικών καθηκόντων έως την υποστήριξη της λήψης αποφάσεων βάσει δεδομένων, τα συμπεράσματα που προκύπτουν από την εφαρμογή τέτοιων συστημάτων, διαμορφώνουν την εικόνα ενός εκπαιδευτικού τοπίου με μεγαλύτερη ανταπόκριση, διαφάνεια και εστίαση στους μαθητές. Καθώς η εκπαιδευτική τεχνολογία συνεχίζει να εξελίσσεται, ο διαρκής αντίκτυπος των συστημάτων διαχείρισης φοιτητών επιβεβαιώνει την ιδιότητά τους ως βασικών εργαλείων για τη διαμόρφωση του μέλλοντος της εκπαίδευσης.

## 4.2 Μελλοντική διερεύνηση

Το μέλλον των συστημάτων διαχείρισης φοιτητών παρουσιάζει πολλά υποσχόμενες προοπτικές, καθώς η τεχνολογία συνεχίζει να εξελίσσεται και το εκπαιδευτικό τοπίο υφίσταται δυναμικές μεταμορφώσεις. Διάφοροι δρόμοι για μελλοντικές εργασίες στο πεδίο τέτοιων συστημάτων δείχνουν προς την κατεύθυνση της βελτίωσης της εμπειρίας των χρηστών, της υιοθέτησης των αναδυόμενων τεχνολογιών και της αντιμετώπισης των εξελισσόμενων αναγκών των εκπαιδευτικών ιδρυμάτων.

Μια μελλοντική κατεύθυνση έγκειται στην τελειοποίηση και επέκταση των χαρακτηριστικών που αποσκοπούν στη βελτίωση της συνολικής εμπειρίας του χρήστη. Οι διεπαφές χρήστη μπορούν να βελτιστοποιηθούν περαιτέρω για την προσβασιμότητα, εξασφαλίζοντας ότι το σύστημα θα ανταποκρίνεται στις διαφορετικές ανάγκες των χρηστών. Επιπλέον, η ενσωμάτωση διαισθητικών ταμπλό, εξατομικευμένων ειδοποιήσεων και φιλικών προς το χρήστη εφαρμογών για κινητά τηλέφωνα μπορεί να συμβάλει σε μια πιο απρόσκοπτη και ελκυστική εμπειρία για όλους τους ενδιαφερόμενους.

Η ενσωμάτωση της τεχνητής νοημοσύνης (AI) και της μηχανικής μάθησης (ML) είναι ένας άλλος πολλά υποσχόμενος τομέας για την εξέλιξη των συστημάτων διαχείρισης φοιτητών. Οι μελλοντικές εργασίες θα μπορούσαν να περιλαμβάνουν την αξιοποίηση της τεχνητής νοημοσύνης για την ανάλυση των δεδομένων των μαθητών. Οι αλγόριθμοι ML μπορούν να βοηθήσουν στην πρόβλεψη των ακαδημαϊκών τάσεων, παρέχοντας στους εκπαιδευτικούς πολύτιμες πληροφορίες για να προσαρμόσουν τις μεθοδολογίες διδασκαλίας τους.

Η τεχνολογία blockchain παρουσιάζει μια ενδιαφέρουσα οδό για την ενίσχυση της ασφάλειας και της ακεραιότητας των δεδομένων των μαθητών στο πλαίσιο του SMS. Με την εφαρμογή αποκεντρωμένης και απαραβίαστης αποθήκευσης δεδομένων, τα ιδρύματα μπορούν να διασφαλίσουν την εμπιστευτικότητα και το απόρρητο των αρχείων

των φοιτητών. Η αλυσίδα μπλοκ μπορεί επίσης να διευκολύνει την ασφαλή επαλήθευση των ακαδημαϊκών πιστοποιητικών, συμβάλλοντας σε μια πιο αξιόπιστη και διαφανή διαδικασία πιστοποίησης [20].

Επιπλέον, η εξέλιξη των συστημάτων διαχείρισης φοιτητών μπορεί να περιλαμβάνει βαθύτερη συνεργασία με συστήματα διαχείρισης μάθησης και άλλες εκπαιδευτικές τεχνολογίες. Η ενσωμάτωση τέτοιων πλατφόρμων μπορεί να δημιουργήσει ένα συνεκτικό οικοσύστημα που διαχειρίζεται απρόσκοπτα τόσο τις διοικητικές όσο και τις διδακτικές πτυχές της εκπαίδευσης. Αυτή η συνέργεια μπορεί να προωθήσει μια πιο ολιστική και διασυνδεδεμένη προσέγγιση της εκπαιδευτικής διαχείρισης.

Συμπερασματικά, τα μελλοντικά έργα των συστημάτων διαχείρισης φοιτητών είναι έτοιμα να αγκαλιάσουν τις εξελίξεις της τεχνολογίας και να ανταποκριθούν στις εξελισσόμενες ανάγκες των εκπαιδευτικών ιδρυμάτων. Είτε μέσω της τελειοποίησης των διεπαφών χρήστη, είτε μέσω της ενσωμάτωσης της τεχνητής νοημοσύνης και της ML για εξατομικευμένη μάθηση, είτε μέσω της υιοθέτησης της αλυσίδας μπλοκ (blockchain) για ασφαλή διαχείριση δεδομένων, είτε μέσω της ενσωμάτωσης καθηλωτικών τεχνολογιών, η πορεία των συστημάτων αυτών αντανακλά τη δέσμευση για τη δημιουργία πιο αποτελεσματικών, καινοτόμων και μαθητοκεντρικών εκπαιδευτικών περιβαλλόντων. Καθώς ξεδιπλώνονται αυτά τα μελλοντικά έργα, τα συστήματα διαχείρισης φοιτητών πιθανότατα θα συνεχίσουν να διαδραματίζουν κεντρικό ρόλο στη διαμόρφωση του μέλλοντος της εκπαίδευσης.

# Παράρτημα Α΄

## Python scripts

### Α΄.1 models.py

```
from django.db import models
```

#### Α΄.1.1 Students

```
class Students(models.Model):  
    study_program = models.CharField(max_length=40, blank=True,  
    ↪ null=True)  
    registration_number = models.IntegerField(blank=True, null=True)  
    id_number_1 = models.FloatField(blank=True, null=True)  
    name = models.CharField(max_length=100, null=True)  
    surname = models.CharField(max_length=100)  
    nationality = models.CharField(max_length=10)  
    student_semester = models.IntegerField(blank=True, null=True)  
    latest_year = models.IntegerField(blank=True, null=True)  
    id_gunet = models.CharField(max_length=30)  
    latest_period = models.CharField(max_length=15)  
    year_of_enrolment = models.IntegerField(blank=True, null=True)  
    registration_period = models.CharField(max_length=20)  
    way_of_enrolment = models.CharField(max_length=100)  
    student_category = models.CharField(max_length=50)  
    free_of_tuition = models.CharField(max_length=10)  
    active = models.CharField(max_length=10)  
    web_active = models.CharField(max_length=10)
```

```
exemption_from_prerequisites = models.CharField(max_length=10)
part_time = models.CharField(max_length=10)
current_status = models.CharField(max_length=25)
gender = models.CharField(max_length=20)
amka = models.CharField(max_length=60)
tin = models.CharField(max_length=60)
username = models.CharField(max_length=100)
email = models.EmailField(max_length=100)
mobile_number = models.CharField(max_length=20)
phone_number = models.CharField(max_length=20)
address = models.CharField(max_length=255)
city = models.CharField(max_length=105)
name_in_english = models.CharField(max_length=100)
last_name_in_english = models.CharField(max_length=100)
fathers_name = models.CharField(max_length=100)
mothers_Name = models.CharField(max_length=100)
date_of_birth = models.CharField(max_length=20)
place_of_birth = models.CharField(max_length=100)
id_number_2 = models.CharField(max_length=255)
type_of_id = models.CharField(max_length=50)
rollback_year = models.CharField(max_length=10)
date_of_registration = models.CharField(max_length=20)
department_registration = models.CharField(max_length=200)
code_of_department = models.IntegerField(blank=True, null=True)

def __str__(self):
    return self.name + ' ' + self.surname
```

## A.1.2 StudentGrades

```
class StudentGrades (models.Model) :
    student = models.ForeignKey(Students, default=None,
        ↪ on_delete=models.CASCADE)
    subject = models.CharField(max_length=40)
    subject_code = models.CharField(max_length=20)
    subject_of_academic_year = models.CharField(max_length=40)
    student_class = models.CharField(max_length=40)
    department = models.CharField(max_length=100)
    registration_number = models.IntegerField(null=True,
        ↪ blank=True)
    grade = models.CharField(max_length=40)
    exam_period = models.CharField(max_length=100)
    semester = models.CharField(max_length=40)
    thesis = models.CharField(max_length=10)
    internship = models.CharField(max_length=10)
    ects = models.FloatField(null=True, default=True)
    ps = models.CharField(max_length=40)
    date1 = models.CharField(max_length=40)
    modified_by = models.CharField(max_length=40)
    date2 = models.CharField(max_length=40)

    def __str__(self):
        return self.subject + ' ' + self.student.name + ' ' +
            ↪ self.student.surname
```

## A.2 views.py

```
from django.shortcuts import render, redirect
import pandas as pd
from .models import Students, StudentGrades
import json

from django.contrib.auth import logout as django_logout
from django.contrib.auth.models import User
from django.contrib.auth import authenticate
from django.contrib.auth.models import auth
from django.contrib.auth.decorators import login_required
from django.http import HttpResponse

@login_required(redirect_field_name='next', login_url='/login')
def dashboard(request):
    return render(request, 'home/dashboard.html')

@login_required(redirect_field_name='next', login_url='/login')
def dashboard_redirect(request):
    return redirect('dashboard')

@login_required(redirect_field_name='next', login_url='/login')
def download(request):
    if request.method == 'POST':
        column_name_list = request.POST.getlist("column_name_list")

    return render(request, 'home/download.html')

@login_required(redirect_field_name='next', login_url='/login')
def upload_student_excel(request):
    return render(request, 'home/upload_student.html')

@login_required(redirect_field_name='next', login_url='/login')
def upload_grade_excel(request):
    return render(request, 'home/upload_grades.html')
```

## A.2.1 Login/Logout

```
def login(request):
    page_data = {}
    next = request.GET.get('next')
    if request.POST:
        loginusername = request.POST['username']
        loginpassword = request.POST['password']
        if User.objects.filter(username=loginusername).exists():
            user = authenticate(username=loginusername,
                ↪ password=loginpassword)
            if user is not None:
                auth.login(request, user)
                if next is not None:
                    return redirect(next)
                else:
                    return redirect('/')
            else:
                page_data['error'] = 'Invalid credentials! Please try
                ↪ again'
        else:
            page_data['error'] = "Account Not Found.."
    return render(request, 'auth/login.html', {'page_data': page_data})

def logout(request):
    django_logout(request)
    return redirect('/login')
```

## A.2.2 Register

```
def register(request):
    page_data = {}
    if request.method == "POST":
        first_name = request.POST['first_name']
        last_name = request.POST['last_name']
        username = request.POST['username']
        password1 = request.POST['password1']
        password2 = request.POST['password2']
        email = request.POST['email']
        # phone = request.POST['phone']
        if password1 == password2:
            if User.objects.filter(username=username).exists():
                page_data['error'] = 'Username Taken'
                # return redirect('register')

            elif User.objects.filter(email=email).exists():
                page_data['error'] = 'Email Taken'
                # return redirect('register')
            else:
                user = User.objects.create_user(
                    first_name=first_name,
                    last_name=last_name,
                    username=username,
                    email=email,
                    password=password1)
                user.save()
                context = {'callback_url': "/login", "message": 'Successfully
                → Registered!', 'col': 'success'}
                return render(request, 'auth/register.html', context)
        else:
            page_data['error'] = 'Password not matching'
            # return redirect('register')
    return render(request, 'auth/register.html', {'page_data': page_data})
```



## A.2.3 Update/Create Students

```
def update_or_create_student(df):  
    try:  
        df = df[0]  
    except:  
        return redirect('/dashboard/upload/students')  
  
    for i in range(len(df)):  
        AM = df.iloc[i]['AM']  
        if Students.objects.filter(registration_number=AM).exists():  
            print('updated..')  
            Students.objects.filter(registration_number=AM).update(  
                study_program=df.iloc[i]['ΠΣ'],  
                registration_number=df.iloc[i]['AM'],  
                id_number_1=df.iloc[i]['Ακ.τεστtestΤΕΣΤΤΕΣΤ Ταυτότητα'],  
                name=df.iloc[i]['Όνομα'],  
                surname=df.iloc[i]['Επώνυμο'],  
                nationality=df.iloc[i]['Υπηκοότητα'],  
                student_semester=df.iloc[i]['Εξάμηνο Φοιτητή'],  
                latest_year=df.iloc[i]['Τελευταίο Ακ. Έτος Φοίτησης'],  
                id_gunet=df.iloc[i]['ID GUNET'],  
                latest_period=df.iloc[i]['Τελευταία Περίοδος Φοίτησης'],  
                year_of_enrolment=df.iloc[i]['Ακ. Έτος Εισαγ.'],  
                registration_period=df.iloc[i]['Περίοδος Εισαγ.'],  
                way_of_enrolment=df.iloc[i]['Τρόπος Εισαγ.'],  
                student_category=df.iloc[i]['Κατηγορία Φοιτητή'],  
                free_of_tuition=df.iloc[i]['Απαλλαγή από δίδακτρα'],  
                active=df.iloc[i]['Ενεργός'],  
                web_active=df.iloc[i]['Ενεργός σε web'],  
                exemption_from_prerequisites=df.iloc[i]['Εξάιρεση από  
                → Προαπαιτούμενα'],  
                part_time=df.iloc[i]['Μερικής Φοίτησης'],  
                current_status=df.iloc[i]['Κατάσταση'],  
                gender=df.iloc[i]['Φύλο'],  
                amka=df.iloc[i]['ΑΜΚΑ'],  
                tin=df.iloc[i]['ΑΦΜ'],  
                username=df.iloc[i]['Username'],  
                email=df.iloc[i]['Email'],  
                mobile_number=df.iloc[i]['Κινητό'],  
                phone_number=df.iloc[i]['Τηλέφωνο'],
```

```

address=df.iloc[i]['Διεύθυνση'],
city=df.iloc[i]['Πόλη'],
name_in_english=df.iloc[i]['Όνομα'],
last_name_in_english=df.iloc[i]['Επώνυμο'],
fathers_name=df.iloc[i]["Πατρώνυμο"],
mothers_Name=df.iloc[i]["Μητρώνυμο"],
date_of_birth=df.iloc[i]['Ημ/νία Γέννησης'],
place_of_birth=df.iloc[i]["Τόπος γέννησης"],
id_number_2=df.iloc[i]['Αρ. ταυτότητας'],
type_of_id=df.iloc[i]['Τύπος ταυτότητας'],
rollback_year=df.iloc[i]['Αναδρ. Έτος Εισαγωγής'],
date_of_registration=df.iloc[i]['Ημ/νία Πρώτης Εγγραφής'],
department_registration=df.iloc[i]['Τμήμα Εισαγωγής'],
code_of_department=df.iloc[i]['Κωδικός τμήματος εισαγωγής'],
)
else:
print('created..')
Students(
study_program=df.iloc[i]['ΠΣ'],
registration_number=df.iloc[i]['ΑΜ'],
id_number_1=df.iloc[i]['Ακ. Ταυτότητα'],
name=df.iloc[i]['Όνομα'],
surname=df.iloc[i]['Επώνυμο'],
nationality=df.iloc[i]['Υπηκοότητα'],
student_semester=df.iloc[i]['Εξάμηνο Φοιτητή'],
latest_year=df.iloc[i]['Τελευταίο Ακ. Έτος Φοίτησης'],
id_gunet=df.iloc[i]['ID GUNET'],
latest_period=df.iloc[i]['Τελευταία Περίοδος Φοίτησης'],
year_of_enrolment=df.iloc[i]['Ακ. Έτος Εισαγ.'],
registration_period=df.iloc[i]['Περίοδος Εισαγ.'],
way_of_enrolment=df.iloc[i]['Τρόπος Εισαγ.'],
student_category=df.iloc[i]['Κατηγορία Φοιτητή'],
free_of_tuition=df.iloc[i]['Απαλλαγή από δίδακτρα'],
active=df.iloc[i]['Ενεργός'],
web_active=df.iloc[i]['Ενεργός σε web'],
exemption_from_prerequisites=df.iloc[i]['Εξαίρεση από
→ Προαπαιτούμενα'],
part_time=df.iloc[i]['Μερικής Φοίτησης'],
current_status=df.iloc[i]['Κατάσταση'],
gender=df.iloc[i]['Φύλο'],

```

```

amka=df.iloc[i]['ΑΜΚΑ'],
tin=df.iloc[i]['ΑΦΜ'],
username=df.iloc[i]['Username'],
email=df.iloc[i]['Email'],
mobile_number=df.iloc[i]['Κινητό'],
phone_number=df.iloc[i]['Τηλέφωνο'],
address=df.iloc[i]['Διεύθυνση'],
city=df.iloc[i]['Πόλη'],
name_in_english=df.iloc[i]['Όνομα'],
last_name_in_english=df.iloc[i]['Επώνυμο'],
fathers_name=df.iloc[i]["Πατρώνυμο"],
mothers_Name=df.iloc[i]["Μητρώνυμο"],
date_of_birth=df.iloc[i]['Ημ/νία Γέννησης'],
place_of_birth=df.iloc[i]["Τόπος γέννησης"],
id_number_2=df.iloc[i]['Αρ. ταυτότητας'],
type_of_id=df.iloc[i]['Τύπος ταυτότητας'],
rollback_year=df.iloc[i]['Αναδρ. Έτος Εισαγωγής'],
date_of_registration=df.iloc[i]['Ημ/νία Πρώτης Εγγραφής'],
department_registration=df.iloc[i]['Τμήμα Εισαγωγής'],
code_of_department=df.iloc[i]['Κωδικός τμήματος εισαγωγής'],
).save()

```

## A.2.4 Update/Create Grades

```
def updated_or_create_grades(df):
    try:
        df = df[0]
    except:
        return redirect('/dashboard/upload/grades')

    student_id = None
    for i in range(len(df)):
        reg_no = df.iloc[i]["AM"]
        if Students.objects.filter(registration_number=reg_no).exists():
            student_id = Students.objects.get(registration_number=reg_no)

        if StudentGrades.objects.filter(student=student_id, subject_code=df.iloc[i]["Κωδ.
        ↪ μαθ."]).exists():
            print('Student updated at position i = ', i, ', AM = ', df.iloc[i]["AM"], ', Subject
            ↪ Code = ', df.iloc[i]["Κωδ. μαθ."], ', Student ID = ', student_id)
            StudentGrades.objects.filter(student=student_id, subject_code=df.iloc[i]["Κωδ.
            ↪ μαθ."]).update(
                student=student_id,
                subject_code=df.iloc[i]['Κωδ. μαθ.'],
                subject_of_academic_year=df.iloc[i]['Μάθημα Ακ. Έτους'],
                student_class=df.iloc[i]['Τμήμα Τάξης'],
                department=df.iloc[i]['Τμήμα'],
                registration_number=df.iloc[i]['AM'],
                grade=df.iloc[i]['Βαθμός'],
                exam_period=df.iloc[i]['Εξ. Περ.'],
                semester=df.iloc[i]['Εξάμηνο βαθμολογίας'],
                thesis=df.iloc[i]['Διπλωματική'],
                internship=df.iloc[i]['Πρακτική'],
                ects=df.iloc[i]['ECTS'],
                ps=df.iloc[i]['ΠΣ'],
                subject=df.iloc[i]['Μάθ. ΠΣ'],
                datel=df.iloc[i]['Ημ. Πρ.'],
                modified_by=df.iloc[i]['Τροποποίηση από'],
                date2=df.iloc[i]['Ημ. Επ.'])
        else:
            print('Student created at position i = ', i, ', AM = ', df.iloc[i]["AM"], ', Subject
            ↪ Code = ', df.iloc[i]["Κωδ. μαθ."], ', Student ID = ', student_id)
            StudentGrades(
                student=student_id,
                subject_code=df.iloc[i]['Κωδ. μαθ.'],
                subject_of_academic_year=df.iloc[i]['Μάθημα Ακ. Έτους'],
                student_class=df.iloc[i]['Τμήμα Τάξης'],
                department=df.iloc[i]['Τμήμα'],
                registration_number=df.iloc[i]['AM'],
                grade=df.iloc[i]['Βαθμός'],
                exam_period=df.iloc[i]['Εξ. Περ.'],
                semester=df.iloc[i]['Εξάμηνο βαθμολογίας'],
```

```
thesis=df.iloc[i]['Διπλωματική'],
internship=df.iloc[i]['Πρακτική'],
ects=df.iloc[i]['ECTS'],
ps=df.iloc[i]['ΠΣ'],
subject=df.iloc[i]['Μόθ. ΠΣ'],
date1=df.iloc[i]['Ημ. Πρ.'],
modified_by=df.iloc[i]['Τροποποίηση από'],
date2=df.iloc[i]['Ημ. Επ.'],
).save()
```

## A.2.5 Preview Students

```
@login_required(redirect_field_name='next', login_url='/login')
def student_preview_dataset(request):
    global data
    context = {}
    if request.FILES.get('student_dataset', False):
        try:
            data = request.FILES['student_dataset'].read()
            df = pd.read_excel(data)
            print('The length is : ', len(df))
            columns = list(df.columns)
            json_records = df.to_json(orient='records')
            data = json.loads(json_records)
            temp_data.append(df)
            html_table_body = ""
            for col in data:
                html_table_body = html_table_body + "<tr>"
                for key in col.keys():
                    html_table_body = html_table_body + f"<td>{col[key]}</td>"
                html_table_body = html_table_body + "</tr>"
            context = {'d': data, "columns": columns, "html_table_body": html_table_body}
        except Exception as e:
            print(f"An exception occurred: {str(e)}")
            context = {'callback_url': "/dashboard/upload/students", "message": 'Something went
            ↪ wrong!', 'col': 'danger'}
        return render(request, 'home/preview_dataset.html', context)
    if request.method == "POST":
        # Saving Student datasets in database
        try:
            res = update_or_create_student(temp_data)
            context = {"message": 'File uploaded Successfully!', 'col': 'success',
                       "callback_url": "/dashboard/download"}
            temp_data.clear()
        except Exception as e:
            print(f"An exception occurred: {str(e)}")
            context = {'callback_url': "/dashboard/upload/students", "message": 'Please check your
            ↪ dataset columns!', 'col': 'danger'}
    return render(request, 'home/preview_dataset.html', context)
```

## A.2.6 Preview Grades

```
@login_required(redirect_field_name='next', login_url='/login')
def grade_preview_dataset(request):
    global data
    context = {}
    if request.FILES.get('student_dataset', False):
        try:
            files = request.FILES.getlist('student_dataset')
            for index, data in enumerate(files):
                if not index:
                    df = pd.read_excel(data)
                    continue
                df = pd.concat([df, pd.read_excel(data)])
            columns = list(df.columns)
            json_records = df.to_json(orient='records')
            data = json.loads(json_records)
            temp_data.append(df)
            html_table_body = ""
            for col in data:
                html_table_body = html_table_body + "<tr>"
                for key in col.keys():
                    html_table_body = html_table_body + f"<td>{col[key]}</td>"
                html_table_body = html_table_body + "</tr>"
            context = {'d': data, "columns": columns, "html_table_body": html_table_body}

        except Exception as e:

            print(f"An exception occurred: {str(e)}")

            context = {'callback_url': "/dashboard/upload/grades", "message": 'Something went
            ↪ wrong!', 'col': 'danger'}
        return render(request, 'home/preview_dataset.html', context)

    if request.method == "POST":
        # Saving Student datasets in database
        try:
            res = updated_or_create_grades(temp_data)
            temp_data.clear()
            context = {"message": 'File uploaded Successfully!', 'col': 'success',
                      "callback_url": "/dashboard/download"}

        except Exception as e:

            print(f"An exception occurred: {str(e)}")
            context = {'callback_url': "/dashboard/upload/grades", "message": 'Please check your
            ↪ dataset columns!',
                      'col': 'danger'}

    return render(request, 'home/preview_dataset.html', context)
```

## A.2.7 Preview/Download

```
@login_required(redirect_field_name='next', login_url='/login')
def preview(request):
    global data
    columns = None
    html_table_body = None
    if request.method == 'POST':
        column_name_list = request.POST.getlist("column_name_list")
        print(f"Selected columns: {column_name_list}")
        column_name_list = [x for x in column_name_list[0].split(',')]
        # translated_cols = translate_columns(column_name_list)
        student_cols = []
        grades_cols = []
        rows = request.POST.get('rows')
        if rows == 'all':
            students = Students.objects.all()
        else:
            students = Students.objects.all()[:int(rows)]

        for field in Students._meta.fields:
            student_cols.append(field.name)

        for field in StudentGrades._meta.fields:
            grades_cols.append(field.name)

        selected_student_cols = [x for x in student_cols if x in column_name_list]
        selected_grades_cols = [x for x in grades_cols if x in column_name_list]
        selected_student_cols.append('registration_number')
        datas = list(students.values(*selected_student_cols))
        has_subject = False
        has_grade = False
        print(f>Data retrieved: {datas}")
        for data in datas:
            student_id = Students.objects.get(registration_number=data['registration_number'])
            stud = StudentGrades.objects.filter(student=student_id)
            grade = list(stud.values(*selected_grades_cols))
            grades_list = []
            data.update({'grades':grades_list})
            if len(grade) >= 1:
                for gr in grade:
                    if 'subject' in gr.keys() and 'grade' in gr.keys():
                        has_subject = True
                        has_grade = True
                        grades_list.append({'subject':gr["subject"], 'grade':gr["grade"]})
                    elif 'subject' in gr.keys():
                        has_subject = True
                        grades_list.append({'subject':gr["subject"]})
                    elif 'grade' in gr.keys():
                        has_grade = True
                        grades_list.append({'grade':gr["grade"]})
            data.update({'grades':grades_list})
```



```

print("Rendering HTML table...")
html_table_body = ""
for col in datas:
    if has_subject or has_grade:
        for sub in col["grades"]:
            html_table_body = html_table_body + "<tr>"
            for key in col.keys():
                if key not in ['grades']:
                    html_table_body = html_table_body + f"<td>{col[key]}</td>"
            if has_subject:
                html_table_body = html_table_body + f"<td>{sub['subject']}</td>"
            if has_grade:
                html_table_body = html_table_body + f"<td>{sub['grade']}</td>"
            html_table_body = html_table_body + "</tr>"
        else:
            html_table_body = html_table_body + "<tr>"
            for key in col.keys():
                if key not in ['grades']:
                    html_table_body = html_table_body + f"<td>{col[key]}</td>"
            html_table_body = html_table_body + "</tr>"

print("HTML table rendered.")

if len(datas) < 1:
    return HttpResponse('No Data Found')

columns = translate_columns(list(data.keys()))
if has_subject:
    columns.append("Μάθημα")
if has_grade:
    columns.append("Βαθμός")

data_values = []
for i in datas:
    print("\nI is:", i)
    grades_temp = i['grades'].copy()
    if has_subject or has_grade:
        for grades_obj in grades_temp:
            values=[]
            for value in list(i.values()):
                if not isinstance(value, list):
                    values.append(value)
            if has_subject:
                values.append(grades_obj['subject'])
            if has_grade:
                values.append(grades_obj['grade'])
            data_values.append(values)
        else:
            del i['grades']
            data_values.append(list(i.values()))

```

```
# Save the Excel file
df = pd.DataFrame(data_values, columns=columns)
if 'AM' in df.columns:
    df = df.sort_values(by=['AM'])
print(df)
writer = pd.ExcelWriter('files/merged/student_and_grades.xlsx', engine='xlsxwriter')
df.to_excel(writer, sheet_name='Sheet1', index=False)
writer.close()

return render(request, 'home/preview.html', {"data": columns, "html_table_body":
↪ html_table_body})
```

## A.3 urls.py

```
from django.urls import include, path
from . import views
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('', views.dashboard_redirect, name='dashboard_redirect'),
    path("__debug__/", include("debug_toolbar.urls")),
    path('dashboard', views.dashboard, name='dashboard'),
    path('login', views.login, name='login'),
    path('logout', views.logout, name='logout'),
    path('register', views.register, name='register'),
    path('dashboard/download', views.download, name='download'),
    path('dashboard/preview', views.preview, name='preview'),
    path('dashboard/upload/students', views.upload_student_excel,
        ↪ name='upload_student_excel'),
    path('dashboard/upload/grades', views.upload_grade_excel,
        ↪ name='upload_grade_excel'),
    path('dashboard/preview-student-dataset',
        ↪ views.student_preview_dataset, name='preview-dataset'),
    path('dashboard/preview-grades-dataset', views.grade_preview_dataset,
        ↪ name='preview-dataset'),
]

if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL,
        ↪ document_root=settings.MEDIA_ROOT)
```

# Παράρτημα Β΄

## Database scripts

### Β΄.1 DeleteStudents.py

```
import sqlite3

# Connecting to sqlite
conn = sqlite3.connect('db.sqlite3')

# Creating a cursor object using the cursor() method
cursor = conn.cursor()

# Deleting records
cursor.execute(''DELETE FROM home_students'')

# Retrieving data after delete
print("All STUDENTS deleted successfully!")
cursor.execute("SELECT * from home_students")
print(cursor.fetchall())

# Commit your changes in the database
conn.commit()

# Closing the connection
conn.close()
```

## B.2 DeleteStudentGrades.py.

```
import sqlite3

# Connecting to sqlite
conn = sqlite3.connect('db.sqlite3')

# Creating a cursor object using the cursor() method
cursor = conn.cursor()

# Deleting records
cursor.execute(''DELETE FROM home_studentgrades'')

# Retrieving data after delete
print("All GRADES deleted successfully!")
cursor.execute("SELECT * from home_studentgrades")
print(cursor.fetchall())

# Commit your changes in the database
conn.commit()

# Closing the connection
conn.close()
```

## B.3 DeleteDB.py

```
import sqlite3

# Connecting to sqlite
conn = sqlite3.connect('db.sqlite3')

# Creating a cursor object using the cursor() method
cursor = conn.cursor()

# Deleting records
cursor.execute(''DELETE FROM home_studentgrades'')
cursor.execute(''DELETE FROM home_students'')

# Retrieving data after delete
print("All STUDENTS and GRADES deleted successfully!")
cursor.execute("SELECT * from home_studentgrades")
print(cursor.fetchall())
cursor.execute("SELECT * from home_students")
print(cursor.fetchall())

# Commit your changes in the database
conn.commit()

# Closing the connection
conn.close()
```

# Βιβλιογραφία

- [1] Paul & Harvey Deitel. *Intro to Python for Computer Science and Data Science*. Pearson Education Inc., Φεβρουάριος 2019.
- [2] Δρ. Δημήτριος Βαρσάμης. *Ανάπτυξη Λογισμικού, Μεταπτυχιακές Σημειώσεις*. Διεθνές Πανεπιστήμιο της Ελλάδος, Οκτώβριος 2019.
- [3] Python Development Team. *Python Documentation v3.12.1*. Python Software Foundation, Δεκέμβριος 2023.
- [4] John V. Guttag. *Introduction to Computation and Programming using Python*. The MIT Press, Ιανουάριος 2021.
- [5] Vahid Mirjalili Sebastian Raschka. *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn and TensorFlow*. Packt Publishing Ltd, Σεπτέμβριος 2017.
- [6] William S. Vincent. *Django for beginners 3.1: Build websites with Python & Django*. Leanpub, Οκτώβριος 2020.
- [7] Django Development Team. *Django Documentation v5.0.2*. Django Software Foundation, Δεκέμβριος 2023.
- [8] Daniel & Audrey Feldroy. *Two scoops of Django 3.x: Best practices for the Django web framework*. Two scoops press, Φεβρουάριος 2021.
- [9] Yann Malet Peter Baumgartner. *High Performance Django*. Lincoln Loop, Μάρτιος 2015.
- [10] SQLite 3 Development Team. *SQLite 3 Documentation v3.44.2*. SQLite Software Foundation, Νοέμβριος 2023.
- [11] Jay A. Kreibich. *Using SQLite: Small. Fast. Reliable*. O'Reilly Media, Σεπτέμβριος 2010.
- [12] Mike Owens & Grant Allen. *The Definitive Guide to SQLite*. Apress, Νοέμβριος 2010.
- [13] Bharatsinh Parmar. *Html, CSS and JavaScript: Learn Web Development*. Self-published, Οκτώβριος 2023.

- [14] Jennifer Robbins. *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*. O'Reilly Media, Μάιος 2018.
- [15] Eric Meyer. *CSS: The Definitive Guide*. O'Reilly Media, Μάιος 2023.
- [16] Theophilus Edet. *JavaScript Programming: Building Modern and Interactive Websites*. Self-published, Δεκέμβριος 2023.
- [17] Dave Winer Jake Spurlock. *Bootstrap: Responsive Web Development*. O'Reilly Media, Ιούνιος 2013.
- [18] Font Awesome Development Team. *Font Awesome Documentation v4.7.0*. Fonticons, Inc., Οκτώβριος 2016.
- [19] Jonathan Chaffer Karl Swedberg. *Learning jQuery: Better Interaction Design and Web Development with Simple JavaScript Techniques*. Packt Publishing Ltd, Ιούνιος 2013.
- [20] Arjuna Sky Kok. *Hands-On Blockchain for Python Developers: Gain blockchain programming skills to build decentralized applications using Python*. Packt Publishing, Φεβρουάριος 2019.