

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**«Οπτικοποίηση δεδομένων σχετικά με την Ελληνική
Επανάσταση»**

Καρβούνης Αλέξανδρος ΑΜ: Π2019020

Επιβλέπουσα: Σωτηροπούλου Άννα

Κέρκυρα 2024

|

Πίνακας Περιεχομένων

Πρόλογος	5
ΚΕΦΑΛΑΙΟ 1: Εισαγωγή	6
1.1. Ορισμός Προβλήματος.....	6
1.2. Διάθρωση Εργασίας	7
ΚΕΦΑΛΑΙΟ 2 : Χρησιμοποιούμενα Εργαλεία, Δομές και Τεχνολογίες	8
2.1. CSV αρχεία	8
2.1.1. Ορισμός CSV αρχείου	8
2.1.2. Δομή CSV αρχείου	9
2.1.3. Περιπτώσεις Χρήσης.....	10
2.1.4. Υποστήριξη από γλώσσες προγραμματισμού.....	11
2.1.5. Δημοφιλή λογισμικά για εισαγωγή και διαχείριση CSV αρχείων	12
2.1.6. Οφέλη.....	13
2.1.7. Τρωτά σημεία	15
2.2. Γλώσσα Python.....	18
2.2.1. Σύντομη Ιστορική Αναδρομή	18
2.2.2. Χαρακτηριστικά Γλώσσας Python.....	19
2.2.3. Εφαρμογές γλώσσας Python	23
2.2.4. Περιορισμοί γλώσσας Python.....	28
2.3. Anaconda Spyder	29
ΚΕΦΑΛΑΙΟ 3: ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ	31
3.1. Εισαγωγή - Βασική Φιλοσοφία Υλοποίησης.....	31
3.2. Περιβάλλον υλοποίησης και απαραίτητες βιβλιοθήκες	32
3.3. Αρχιτεκτονική Εφαρμογής	33
3.4. Αρχική παρουσίαση συνόλου δεδομένων	34
3.5. Παρουσίαση λειτουργικότητας της εφαρμογής.....	42
3.6. Σχολιασμός της εφαρμογής.....	48
Συμπεράσματα – Μελλοντική Εργασία.....	49
ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ.....	51
ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ.....	62

Ευρετήριο Εικόνων

Εικόνα 1: Ιστορική εξέλιξη της γλώσσας Python (Πηγή: https://unstop.com/blog/what-is-python)	19
Εικόνα 2: Περιβάλλον Anaconda Spyder.....	32
Εικόνα 3: Κατανομή πλήθους εγγραφών ανά μήνα.....	41
Εικόνα 4: Κατανομή πλήθους εγγραφών ανά έτος	41
Εικόνα 5: Αρχική Οθόνη Εφαρμογής.....	42
Εικόνα 6: Επιλογή CSV αρχείου.....	43
Εικόνα 7: Επιτυχής φόρτωση CSV αρχείου	44
Εικόνα 8: Επιλογή λογαριασμού προς οπτικοποίηση.....	45
Εικόνα 9: Πίνακας δοσοληψιών λογαριασμού	46
Εικόνα 10: Γράφημα με ποσά που δίνει ο λογαριασμός	46
Εικόνα 11: Γράφημα με ποσά που λαμβάνει ο λογαριασμός	47

Ευρετήριο Πινάκων

Πίνακας 3: Στήλες συνόλου δεδομένων.....	35
Πίνακας 4: Διαστάσεις συνόλου δεδομένων	35
Πίνακας 5: Πλήθος ελλειπόν τιμών ανά στήλη.....	35
Πίνακας 6: Πλήθος πρώτων λογαριασμών.....	36
Πίνακας 7: Λίστα πρώτων λογαριασμών.....	40

Πρόλογος

Η συγκεκριμένη εργασία υλοποιήθηκε με σκοπό τη διαπίστωση της ανάγκης που υπάρχει για αποτελεσματική διαχείριση και οπτικοποίηση μεγάλου όγκου δεδομένων. Για τον λόγο αυτό, πέρα από τη βιβλιογραφική ανασκόπηση η οποία θα παρατεθεί, θα παρουσιαστεί και πρακτικό μέρος, το οποίο χρησιμοποιώντας τις προηγμένες δυνατότητες της γλώσσας Python, θα καταβάλει προσπάθεια ώστε να διαχειριστεί και να οπτικοποιήσει το περιεχόμενο ενός συνόλου δεδομένων προερχόμενο από αρχείο CSV, και το οποίο αφορά δεδομένα σχετικά με την Ελληνική Επανάσταση.

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

1.1. Ορισμός Προβλήματος

Η επεξεργασία και η οπτικοποίηση των αρχείων CSV (Comma-Separated Values) είναι απαραίτητες διαδικασίες, καθώς τα ακατέργαστα δεδομένα στην αρχική τους μορφή μπορεί να είναι δυσνόητα και δύσκολα στην ερμηνεία. Τα αρχεία CSV χρησιμοποιούνται συχνά για την αποθήκευση μεγάλων συνόλων δεδομένων και χωρίς επεξεργασία, τα αρχεία αυτά μπορεί να περιέχουν σφάλματα, ελλιπή δεδομένα ή ασυνεπείς μορφές που δυσχεραίνουν την ανάλυση. Η επεξεργασία των δεδομένων επιτρέπει στους χρήστες να καθαρίσουν, να φιλτράρουν και να οργανώσουν τις πληροφορίες, καθιστώντας τις κατάλληλες για περαιτέρω διερεύνηση. Για παράδειγμα, η επεξεργασία μπορεί να περιλαμβάνει τη μετατροπή τύπων δεδομένων, τη διαχείριση ελλιπών τιμών ή την εφαρμογή μετασχηματισμών για την κανονικοποίηση των δεδομένων. Έτσι, το σύνολο δεδομένων γίνεται πιο εύχρηστο, εξασφαλίζοντας την ακρίβεια και την αξιοπιστία των συμπερασμάτων που προκύπτουν από αυτό.

Η οπτικοποίηση είναι εξίσου σημαντική, καθώς βοηθά στη μετάφραση των επεξεργασμένων δεδομένων σε γραφικές αναπαραστάσεις που είναι πιο διαισθητικές και εύκολες στην κατανόηση. Τα διαγράμματα, οι γραφικές παραστάσεις και τα γραφήματα επιτρέπουν στους χρήστες να εντοπίζουν γρήγορα τάσεις, πρότυπα ή ακραίες τιμές που μπορεί να μην είναι εμφανείς από σειρές αριθμών. Η οπτικοποίηση των δεδομένων CSV μπορεί να προσφέρει πληροφορίες που βοηθούν στη λήψη αποφάσεων, είτε σε επιχειρηματικά πλαίσια, είτε στην έρευνα, είτε σε προσωπικά έργα. Για παράδειγμα, η χαρτογράφηση των τάσεων πωλήσεων με την πάροδο του χρόνου ή η ανάλυση δημογραφικών στοιχείων πελατών μέσω οπτικών εργαλείων επιτρέπει στους ενδιαφερόμενους να λαμβάνουν τεκμηριωμένες αποφάσεις βάσει σαφών και ουσιαστικών πληροφοριών. Εν ολίγοις, ο συνδυασμός επεξεργασίας και οπτικοποίησης μετατρέπει τα ακατέργαστα δεδομένα CSV σε ουσιαστική πληροφορία.

Στα πλαίσια της συγκεκριμένης εργασίας, βασικός στόχος ήταν η επεξεργασία και η οπτικοποίηση ενός πρωτότυπου συνόλου δεδομένων αποθηκευμένο σε CSV αρχείο, το οποίο αφορά στοιχεία δοσοληψιών μεταξύ λογαριασμών της Επανάστασης. Στα

κεφάλαια που θα ακολουθήσουν θα καταστεί κατανοητό πως επετεύχθη το συγκεκριμένο χρησιμοποιώντας τις δυνατότητες που παρέχει η γλώσσα Python.

1.2. Διάρθρωση Εργασίας

Η διάρθρωση της εργασίας έχει ως εξής:

Στο **δεύτερο κεφάλαιο** πραγματοποιείται μια παρουσίαση των τεχνολογιών, των εργαλείων και των δομών που χρησιμοποιήθηκαν στα πλαίσια του πρακτικού μέρους της εργασίας. Πιο συγκεκριμένα θα παρουσιαστούν βασικά στοιχεία που αφορούν τη γλώσσα Python, η οποία αποτέλεσε τη γλώσσα υλοποίησης της εφαρμογής. Επιπλέον, θα παρατεθούν τα χαρακτηριστικών των CSV αρχείων, τα οποία αποτελούν το επίκεντρο ενδιαφέροντος όσον αφορά την επεξεργασία και οπτικοποίηση. Τέλος, θα συνοψιστούν τα χαρακτηριστικά του εργαλείου Anaconda Spyder, το οποίο αποτέλεσε το περιβάλλον υλοποίησης και δοκιμών της εφαρμογής.

Στο **τρίτο κεφάλαιο** πραγματοποιείται αναλυτική παρουσίαση της εφαρμογής που υλοποιήθηκε. Πιο συγκεκριμένα, παρατίθεται η βασική της φιλοσοφία και οι στόχοι της. Στην συνέχεια, αναφέρονται το περιβάλλον υλοποίησής της, αλλά και σημαντικές βιβλιοθήκες της γλώσσας Python, που είναι απαραίτητες για τη λειτουργία της. Ακολουθεί η ανάλυση της αρχιτεκτονικής της και μια πρωταρχική παρουσίαση του δοκιμαστικού αρχείου, με βάση το οποίο υλοποιήθηκε. Έπειτα παρουσιάζεται η λειτουργικότητά της μέσω screenshots και τέλος αναφέρονται σχόλια που έχουν εξαχθεί κατά την πραγματοποίηση της όλης υλοποίησης.

ΚΕΦΑΛΑΙΟ 2 : Χρησιμοποιούμενα Εργαλεία, Δομές και Τεχνολογίες

2.1. CSV αρχεία

2.1.1. Ορισμός CSV αρχείου

Τα αρχεία CSV [1] [2] είναι μια μορφή απλής αποθήκευσης δεδομένων. Ένα αρχείο CSV είναι ένα απλό αρχείο κειμένου που αποθηκεύει δεδομένα πίνακα σε δομημένη μορφή. Σε ένα αρχείο CSV, κάθε γραμμή αντιπροσωπεύει μια ενιαία εγγραφή και κάθε εγγραφή αποτελείται από ένα ή περισσότερα πεδία που χωρίζονται από έναν οριοθέτη, συνήθως κόμμα. Η πρώτη γραμμή του αρχείου περιέχει συχνά τις κεφαλίδες που περιγράφουν τα περιεχόμενα κάθε στήλης.

Αποτελούν μια δημοφιλή επιλογή για την αποθήκευση και ανταλλαγή δεδομένων, επειδή είναι απλά και εύχρηστα. Μπορούν να δημιουργηθούν και να επεξεργαστούν με οποιοδήποτε πρόγραμμα επεξεργασίας κειμένου και μπορούν να εισαχθούν και να εξαχθούν από πολλές εφαρμογές λογισμικού. Τα αρχεία CSV μπορούν εύκολα να εισαχθούν σε εφαρμογές υπολογιστικών φύλλων όπως το Microsoft Excel ή τα φύλλα Google, καθιστώντας εύκολο τον χειρισμό και την ανάλυση μεγάλων συνόλων δεδομένων.

2.1.2. Δομή CSV αρχείου

Ένα αρχείο CSV είναι ένα αρχείο απλού κειμένου και αξιοποιεί μια απλή δομή που δεν είναι απλά εύκολη στην κατανόηση από τους υπολογιστές, αλλά είναι επίσης αναγνώσιμη από τον άνθρωπο.

Η δομή ενός τυπικού αρχείου CSV περιλαμβάνει τρία κύρια στοιχεία:

- **Γραμμή κεφαλίδας:** Η πρώτη γραμμή (σειρά), που χρησιμεύει για την παράθεση των τίτλων των στηλών, υποδεικνύοντας τον τύπο δεδομένων που κατέχει κάθε στήλη. Χρησιμοποιεί την ίδια μορφή με τις επόμενες γραμμές δεδομένων, με τιμές που διαχωρίζονται από έναν οριοθέτη.
- **Οριοθέτης:** Ένας χαρακτήρας που διαχωρίζει τις τιμές σε κάθε γραμμή. Ενώ το κόμμα είναι ο τυπικός οριοθέτης σε αρχεία CSV (εξού και το όνομα), μπορούν επίσης να χρησιμοποιηθούν και άλλοι οριοθέτες όπως tabs (σε αρχεία TSV), ανάλογα με τη μορφή αρχείου και τις απαιτήσεις.
- **Γραμμές δεδομένων:** Μετά την κεφαλίδα, κάθε γραμμή αντιπροσωπεύει μια νέα καταχώρηση δεδομένων, με τιμές οργανωμένες σύμφωνα με τους τίτλους κεφαλίδας και διαχωρισμένες από τον επιλεγμένο οριοθέτη. Στο λογισμικό υπολογιστικών φύλλων, αυτές οι γραμμές αντιστοιχούν σε μεμονωμένες γραμμές στο πλέγμα, παρουσιάζοντας τα δεδομένα σε οργανωμένη μορφή πίνακα.

Παράδειγμα ενός CSV αρχείου δίνεται στη συνέχεια:

Username	Identifier	One-time password	Recovery code	First name	Last name	Department	Location
booker12	; 9012	;12se74	; rb9012	Rachel	Booker	Sales	Manchester
grey07	; 2070	;04ap67	; lg2070	Laura	Grey	Depot	London
johnson81	; 4081	;30no86	; cj4081	Craig	Johnson	Depot	London
jenkins46	; 9346	;14ju73	;mj9346	Mary	Jenkins	Engineering	Manchester
smith79	; 5079	;09ja61	; js5079	Jamie	Smith	Engineering	Manchester

2.1.3. Περιπτώσεις Χρήσης

Οι πιο συνηθισμένες περιπτώσεις χρήσης των CSV αρχείων είναι οι παρακάτω:

- **Εισαγωγή και εξαγωγή δεδομένων:** Μία από τις πιο συνηθισμένες περιπτώσεις χρήσης για τα αρχεία CSV είναι η εισαγωγή και εξαγωγή δεδομένων. Δεδομένης της απλότητας και της συμβατότητάς τους, τα αρχεία CSV χρησιμοποιούνται συχνά για τη μεταφορά δεδομένων μεταξύ διαφορετικών τύπων λογισμικού, εφαρμογών και βάσεων δεδομένων. Για παράδειγμα, ένας χρήστης μπορεί να εξάγει δεδομένα από μια βάση δεδομένων MySQL σε ένα αρχείο CSV για χρήση σε ένα υπολογιστικό φύλλο Excel ή σε ένα εργαλείο ανάλυσης δεδομένων Python.
- **Ανάλυση δεδομένων:** Τα αρχεία CSV είναι μια δημοφιλής επιλογή στην ανάλυση δεδομένων λόγω της συμβατότητάς τους με διάφορα εργαλεία και γλώσσες ανάλυσης δεδομένων, όπως R και Python. Οι αναλυτές μπορούν εύκολα να χειριστούν και να αναλύσουν δεδομένα που είναι αποθηκευμένα σε αρχεία CSV, καθιστώντας τα μια κοινή μορφή για την αποθήκευση συνόλων δεδομένων.
- **Μετεγκατάσταση δεδομένων:** Κατά τη μετεγκατάσταση δεδομένων από το ένα σύστημα στο άλλο, τα αρχεία CSV χρησιμοποιούνται συχνά λόγω της απλότητας και της συμβατότητάς τους. Για παράδειγμα, κατά τη μεταφορά δεδομένων πελατών από ένα σύστημα διαχείρισης σχέσεων πελατών (CRM) σε ένα άλλο, τα δεδομένα ενδέχεται να εξαχθούν σε ένα αρχείο CSV και στη συνέχεια να εισαχθούν στο νέο σύστημα.
- **Δημιουργία αντιγράφων ασφαλείας δεδομένων:** Μπορούν να χρησιμοποιηθούν για τη δημιουργία αντιγράφων ασφαλείας δεδομένων. Δεδομένης της απλότητας και της μορφής απλού κειμένου, τα αρχεία CSV μπορούν να είναι μια καλή επιλογή για τη δημιουργία αναγνώσιμων, προσβάσιμων αντιγράφων ασφαλείας σημαντικών δεδομένων.
- **Αναφορά:** Τα αρχεία CSV χρησιμοποιούνται συχνά για τη δημιουργία αναφορών, ειδικά για εφαρμογές με σύνολα δεδομένων μεγάλου όγκου. Αυτές οι αναφορές μπορούν στη συνέχεια να ανοίξουν, να διαχειριστούν και να αναλυθούν εύκολα χρησιμοποιώντας λογισμικό υπολογιστικών φύλλων.
- **Μηχανική μάθηση:** Στη μηχανική μάθηση, τα αρχεία CSV χρησιμοποιούνται συχνά για την αποθήκευση συνόλων δεδομένων εκπαίδευσης. Αυτά τα αρχεία μπορούν εύκολα να φορτωθούν σε βιβλιοθήκες μηχανικής μάθησης, όπου μπορούν να χρησιμοποιηθούν για την εκπαίδευση διαφόρων τύπων μοντέλων.

2.1.4. Υποστήριξη από γλώσσες προγραμματισμού

Πολλές γλώσσες προγραμματισμού παρέχουν ενσωματωμένες βιβλιοθήκες για ανάγνωση και εγγραφή σε αρχεία CSV. Μερικές από τις πιο συχνά χρησιμοποιούμενες γλώσσες προγραμματισμού που υποστηρίζουν αρχεία CSV είναι οι παρακάτω:

- **Python:** Η Python διαθέτει το module `csv`, που παρέχει λειτουργικότητα για ανάγνωση και εγγραφή αρχείων CSV. Πιο συγκεκριμένα, επιτρέπει την αποθήκευση CSV δεδομένων σε δομές δεδομένων και αντίστροφα.
- **Java:** Η Java διαθέτει βιβλιοθήκες όπως η `OpenCSV` και η `Apache Commons CSV`, που παρέχουν API για λειτουργίες πάνω σε αρχεία CSV. Αυτές οι βιβλιοθήκες προσφέρουν δυνατότητες ανάλυσης δεδομένων CSV, χειρισμού διαφορετικών οριοθετών και εγγραφής σε αρχεία CSV.
- **JavaScript:** Ενώ ή ίδια η JavaScript δεν διαθέτει ενσωματωμένη υποστήριξη για αρχεία CSV, υπάρχουν πολλές βιβλιοθήκες τρίτων για την επεξεργασία δεδομένων CSV σε περιβάλλοντα JavaScript, τόσο στην πλευρά του διακομιστή (Node.js) όσο και στην πλευρά του πελάτη.
- **C#:** Το .NET framework περιλαμβάνει βιβλιοθήκες όπως η `CsvHelper` για εργασία με αρχεία CSV στην C#. Αυτές οι βιβλιοθήκες προσφέρουν λειτουργίες για την ανάλυση δεδομένων CSV και την εγγραφή σε αρχείο CSV.
- **Ruby:** Η Ruby παρέχει βιβλιοθήκες όπως η `CSV` και `FasterCSV` (βιβλιοθήκη CSV για παλαιότερες εκδόσεις) για πραγματοποίηση ανάλυσης και δημιουργίας δεδομένων CSV. Επιπλέον, αυτές οι βιβλιοθήκες προσφέρουν χρήσιμες μεθόδους για την ανάγνωση και την εγγραφή αρχείων CSV σε προγράμματα Ruby.
- **PHP:** Η PHP παρέχει συναρτήσεις, όπως η `fgetcsv()` και η `fputcsv()` για ανάγνωση και δημιουργία δεδομένων CSV αντίστοιχα. Αυτές οι συναρτήσεις αποτελούν μέρος της βασικής γλώσσας PHP και παρέχουν βασικές δυνατότητες χειρισμού CSV.
- **R:** Η συγκεκριμένη γλώσσα παρέχει συναρτήσεις όπως η `read.csv()` και η `write.csv()` για ανάγνωση και εγγραφή σε αρχεία CSV. Αυτές οι συναρτήσεις αποτελούν μέρος της γλώσσας R και χρησιμοποιούνται συνήθως για εργασίες εισαγωγής/εξαγωγής δεδομένων στη γλώσσα R.

2.1.5. Δημοφιλή λογισμικά για εισαγωγή και διαχείριση CSV αρχείων

Στα πλαίσια της συγκεκριμένης ενότητας παρουσιάζονται δημοφιλή λογισμικά, τα οποία χρησιμοποιούνται για εισαγωγή και διαχείριση CSV αρχείων:

- **Microsoft Excel:** Ένα από τα πιο δημοφιλή λογισμικά υπολογιστικών φύλλων που επιτρέπει την εισαγωγή αρχείων CSV και την εκτέλεση διαφόρων λειτουργιών στα δεδομένα που αυτό περιέχει.
- **Google Sheets:** Ένα λογισμικό υπολογιστικών φύλλων που βασίζεται σε cloud λογική, παρόμοιο με το Excel, επιτρέποντας την εύκολη εισαγωγή και τον χειρισμό αρχείων CSV στο διαδίκτυο.
- **Συστήματα διαχείρισης βάσεων δεδομένων SQL (MySQL, PostgreSQL, SQLite):** Αυτά τα συστήματα διαχείρισης βάσεων δεδομένων επιτρέπουν την εισαγωγή αρχείων CSV απευθείας σε πίνακες βάσεων δεδομένων, με σκοπό την πραγματοποίηση περαιτέρω αναλύσεων.
- **Tableau:** Ένα δημοφιλές λογισμικό απεικόνισης δεδομένων που επιτρέπει την εισαγωγή αρχείων CSV για τη δημιουργία διαδραστικών απεικονίσεων και πινάκων ελέγχου.
- **Power BI:** Το εργαλείο business analytics της Microsoft που επιτρέπει την εισαγωγή αρχείων CSV για ανάλυση και οπτικοποίηση δεδομένων.
- **Qlik Sense:** Ένα άλλο εργαλείο επιχειρηματικής ευφυΐας και οπτικοποίησης δεδομένων που υποστηρίζει την εισαγωγή αρχείων CSV.

2.1.6. Οφέλη

Η υποκείμενη δομή της μορφής CSV συχνά την καθιστά μια βιώσιμη επιλογή για την αποθήκευση δεδομένων και αρκετά δημοφιλή (όπως και η περίπτωση της XML αναπαράστασης). Αυτά τα διαρθρωτικά οφέλη έχουν ιδιαίτερη επίδραση σε τομείς που περιλαμβάνουν τη μεταφορά δεδομένων, την ανάλυση δεδομένων, τον χειρισμό δεδομένων και τη διαχείριση δεδομένων.

Ευρεία συμβατότητα: Ένα από τα πιο σημαντικά πλεονεκτήματα της μορφής CSV είναι η ευρεία συμβατότητά της. Τα CSV αρχεία έχουν σχεδιαστεί για να είναι καθολικά προσβάσιμα, με τις περισσότερες εφαρμογές λογισμικού να μπορούν εύκολα να τα διαβάζουν και να τα επεξεργάζονται. Αυτή η συμβατότητα καλύπτει ένα ευρύ φάσμα βάσεων δεδομένων, διάφορες γλώσσες προγραμματισμού και εργαλεία ανάλυσης και οπτικοποίησης δεδομένων. Είναι χαρακτηριστικό το γεγονός ότι όλες οι σύγχρονες εφαρμογές υπολογιστικών φύλλων υποστηρίζουν αρχεία CSV. Στον πυρήνα τους, τα αρχεία CSV έχουν σχεδιαστεί για να είναι σχετικά σαφή και εύχρηστα. Έχουν τυποποιημένη και φιλική μορφοποίηση και αποτελούνται πάντα από γραμμές που αντιπροσωπεύουν εγγραφές δεδομένων και στήλες που αντιπροσωπεύουν τα αντίστοιχα χαρακτηριστικά τους. Αυτή η δομή δεν απαιτεί κανένα πολύπλοκο λογισμικό για να ανοιχθεί και να υποστεί επεξεργασία, ενώ επιτρέπει στα δεδομένα CSV να γίνονται εύκολα κατανοητά από χρήστες που δεν είναι εξοικειωμένοι με τεχνικούς όρους. Η δυνατότητα χρήσης τους σχεδόν σε κάθε πλατφόρμα καθιστά τα αρχεία CSV μια απίστευτα δημοφιλή επιλογή και για τον λόγο αυτό εφαρμόζονται ευρέως στο πεδίο της ανταλλαγής δεδομένων μεταξύ διαφορετικών συστημάτων. Είναι επίσης ο πιο χρησιμοποιούμενος τύπος αρχείου, όταν οι χρήστες πρέπει να εισάγουν ή να εξαγάγουν δεδομένα σε διαφορετικές πλατφόρμες ή μέσα σε μια εφαρμογή.

Αποδοτικότητα: Η αποδοτικότητα είναι ένα άλλο βασικό όφελος που προσφέρουν τα αρχεία CSV. Η απλή, ελαφριά δομή τους στερείται μορφοποίησης ή στυλ, που θα αύξανε το μέγεθος των αρχείων ή θα επιβράδυνε την απόδοσή τους. Η συμπαγής φύση τους επιτρέπει επίσης να συμπιέζονται εύκολα και να αποστέλλονται μέσω ηλεκτρονικού ταχυδρομείου. Τα δεδομένα CSV μπορούν να δημιουργηθούν, να αποθηκευτούν και να μεταφερθούν εύκολα, καθιστώντας τα ιδανικά για σενάρια όπου τα δεδομένα πρέπει να ανταλλάσσονται ή να αναλύονται γρήγορα. Αυτό το χαρακτηριστικό είναι επίσης ιδιαίτερα επωφελές όταν πραγματοποιείται διαχείριση μεγάλου όγκου δεδομένων. Επιπλέον, τα αρχεία CSV είναι συμβατά με συστήματα ελέγχου εκδόσεων, υποστηρίζοντας λεπτομερή παρακολούθηση των αλλαγών με την πάροδο του χρόνου.

Ευελιξία: Τα αρχεία CSV προσφέρουν υψηλό βαθμό ευελιξίας λόγω της ικανότητάς τους να αποθηκεύουν ένα ευρύ φάσμα τύπων και δομών δεδομένων. Σε αντίθεση με ορισμένες άλλες μορφές αρχείων που ενδέχεται να έχουν περιορισμούς στους τύπους δεδομένων που μπορούν να φιλοξενήσουν, τα αρχεία CSV μπορούν να χειριστούν κείμενο, αριθμούς, ημερομηνίες και ακόμη πιο σύνθετες δομές όπως δεδομένα JSON ή XML. Κάθε τιμή σε ένα αρχείο CSV διαχωρίζεται συνήθως με κόμμα, αλλά μπορούν να χρησιμοποιηθούν και άλλοι οριοθέτες όπως tabs ή ερωτηματικά, ενισχύοντας την προσαρμοστικότητα της μορφής σε διαφορετικές ανάγκες. Αυτή η ευελιξία επιτρέπει στα αρχεία CSV να χρησιμοποιούνται σε ποικίλες εφαρμογές, από απλή αποθήκευση δεδομένων έως πολύπλοκα σενάρια ανταλλαγής δεδομένων. Επιπλέον, η απλότητα και η διαφάνεια των αρχείων CSV συμβάλλουν στην ευελιξία τους. Επειδή είναι αρχεία απλού κειμένου, τα αρχεία CSV μπορούν εύκολα να ανοίξουν και να προβληθούν σε οποιοδήποτε πρόγραμμα επεξεργασίας κειμένου, καθιστώντας τα αναγνώσιμα από τον άνθρωπο και κατανοητά χωρίς την ανάγκη εξειδικευμένου λογισμικού. Αυτή η διαφάνεια διευκολύνει τη συνεργασία και την αντιμετώπιση προβλημάτων, καθώς οι ενδιαφερόμενοι μπορούν να επιθεωρήσουν απευθείας το αρχείο για να επαληθεύσουν το περιεχόμενό του ή να εντοπίσουν τυχόν προβλήματα.

Υποστήριξη από πολλές γλώσσες προγραμματισμού: Πολλές γλώσσες προγραμματισμού προσφέρουν ενσωματωμένες βιβλιοθήκες για ανάγνωση και εγγραφή αρχείων CSV. Αυτές οι βιβλιοθήκες παρέχουν λειτουργίες και μεθόδους για την αποτελεσματική διαχείριση των δεδομένων CSV, επιτρέποντας στους προγραμματιστές να ενσωματώνουν απρόσκοπτα την επεξεργασία αρχείων CSV στα έργα λογισμικού τους. Είτε πρόκειται για το module CSV της Python, τις κλάσεις CSVReader και CSVWriter της Java ή παρόμοια βοηθητικά προγράμματα σε άλλες γλώσσες, αυτά τα εργαλεία απλοποιούν εργασίες όπως ανάγνωση δεδομένων από αρχεία CSV, εγγραφή δεδομένων σε αρχεία CSV και εκτέλεση λειτουργιών όπως φιλτράρισμα, ταξινόμηση και μετατροπή δεδομένων CSV.

2.1.7. Τρωτά σημεία

Στα πλαίσια της συγκεκριμένης ενότητας, παρατίθενται τα τρωτά σημεία τα οποία σχετίζονται με τη χρήση των CSV αρχείων:

Σφάλμα χρήστη και ακατάλληλα μορφοποιημένα δεδομένα CSV: Ενώ τα αρχεία CSV μπορούν να παρέχουν έναν απλό τρόπο αποθήκευσης και μεταφοράς δεδομένων, είναι μη τυποποιημένα και επιρρεπή σε σφάλματα χρήστη. Παρόλο που η μορφή CSV χρησιμοποιείται εδώ και δεκαετίες, δεν υπάρχει τυποποιημένη μορφή που να επιβάλλεται σε όλες τις εφαρμογές. Το 2005, κυκλοφόρησε το RFC 4180, αλλά οι περισσότερες πλατφόρμες δεν το ακολουθούν (π.χ. το Microsoft Excel δεν το κάνει ρητά). Οι χρήστες μπορούν συχνά να εισάγουν κατά λάθος λανθασμένα ή ελλιπή δεδομένα, όπως διπλές κεφαλίδες, ελλείπουσες τιμές, λανθασμένα δεδομένα σε πεδία ή μη υποστηριζόμενους χαρακτήρες (προκαλώντας προβλήματα κωδικοποίησης).

Όπως υποδηλώνει το όνομα, τα δεδομένα CSV διαχωρίζονται συνήθως με κόμματα. Εξαιτίας αυτού, οι περισσότερες τυπικές βιβλιοθήκες αναμένουν ένα κόμμα ως οριοθέτη και απαιτούν από τους χρήστες να αλλάξουν χειροκίνητα τον οριοθέτη σε περίπτωση που χρησιμοποιείται κάποιος διαφορετικός. Είναι επίσης δυνατό να γραφεί κώδικας που επιχειρεί να ανιχνεύσει τον οριοθέτη που χρησιμοποιείται σε ένα αρχείο, ωστόσο, η αντιμετώπιση ασυνήθιστων οριοθετών μπορεί να προκαλέσει σημαντικά προβλήματα και να απαιτήσει μη αυτόματο έλεγχο πριν από τη σωστή φόρτωση των δεδομένων.

Δεδομένου ότι μπορούν να προκαλέσουν ορισμένα προβλήματα διαχείρισης δεδομένων, είναι σημαντικό να διασφαλιστεί ότι τα δεδομένα που εισάγονται από αρχεία CSV είναι ακριβή, πλήρη και συνεπή με την προβλεπόμενη μορφή. Οι εφαρμογές συνήθως πρέπει να λαμβάνουν αρχεία CSV σε μια συγκεκριμένη δομή (και με σωστά εισαγόμενα δεδομένα) για να αποφύγουν απροσδόκητη συμπεριφορά ή άκυρες αναλύσεις.

Προκλήσεις στη μεταφορά δεδομένων: Δεδομένου ότι η εξαγωγή αρχείων CSV είναι ένα κοινό χαρακτηριστικό στις περισσότερες εφαρμογές ιστού, η μορφή CSV είναι μια κοινή μέθοδος μεταφοράς αρχείων. Για σύνθετα σύνολα δεδομένων ή αυτά που πρέπει να εισαχθούν σε συγκεκριμένη μορφή, η επένδυση χρόνου για τη δημιουργία ροής εισαγωγής CSV που καθοδηγεί τους χρήστες μέσω του καθαρισμού και της μορφοποίησης των δεδομένων CSV πριν από την εισαγωγή του αρχείου, μπορεί επίσης να βοηθήσει, ειδικά όταν απαιτείται συγκεκριμένη, σύνθετη μορφοποίηση του αρχείου. Για εφαρμογές όπου ένας χρήστης εισάγει αρχεία CSV, εάν υπάρχουν αυστηρές απαιτήσεις επικύρωσης, η έλλειψη καθοδήγησης κατά τη

διαδικασία εισαγωγής μπορεί να προκαλέσει πολλά προβλήματα. Λόγω των πολλών δυνατοτήτων σχετικά με τους τύπους σφαλμάτων μορφοποίησης και εισαγωγής δεδομένων, η επίλυση σφαλμάτων εισαγωγής απαιτεί συχνά μη αυτόματο εντοπισμό των σφαλμάτων από τεχνικά και μη τεχνικά μέλη της ομάδας και αναγκάζει τους πελάτες να εγκαταλείψουν κατά τη διάρκεια των διαδικασιών ενσωμάτωσης λόγω πρόσθετων απογοητεύσεων.

Προβλήματα απόδοσης σε μεγάλα σύνολα δεδομένων: Τα προβλήματα απόδοσης με μεγάλα σύνολα δεδομένων των αρχείων CSV προέρχονται κυρίως από τη φύση τους, που βασίζεται στο κείμενο και την έλλειψη αποτελεσματικών μηχανισμών ευρετηρίασης και συμπίεσης. Όσοι διαχειρίζονται μεγάλα αρχεία CSV, μπορεί να αντιμετωπίσουν πολλές προκλήσεις που επηρεάζουν την ταχύτητα επεξεργασίας, τη χρήση μνήμης και τη συνολική απόδοση του συστήματος.

Αρχικά, η ανάγνωση και η εγγραφή μεγάλων αρχείων CSV μπορεί να είναι αργή, ειδικά σε περιβάλλοντα με περιορισμένους πόρους ή όταν χρησιμοποιούνται βιβλιοθήκες λογισμικού που δεν είναι βελτιστοποιημένες για τον αποτελεσματικό χειρισμό μεγάλων αρχείων. Δεδομένου ότι τα αρχεία CSV είναι αρχεία απλού κειμένου, πρέπει να αναλύονται γραμμή προς γραμμή και κάθε γραμμή πρέπει να είναι tokenized. Αυτή η διαδοχική επεξεργασία μπορεί να οδηγήσει σε σημαντική επιβάρυνση, ειδικά για αρχεία με εκατομμύρια ή και δισεκατομμύρια εγγραφές δεδομένων. Επιπλέον, η ανάγνωση ή η εγγραφή μεγάλων αρχείων CSV μπορεί να έχει σημαντικές απαιτήσεις μνήμης για την αποθήκευση των δεδομένων, οδηγώντας σε αυξημένη χρήση μνήμης και πιθανά προβλήματα απόδοσης που σχετίζονται με τη μνήμη.

Δεύτερον, η εκτέλεση λειτουργιών όπως φιλτράρισμα, ταξινόμηση ή συγκέντρωση δεδομένων σε μεγάλα αρχεία CSV μπορεί να είναι υπολογιστικά δαπανηρή και χρονοβόρα. Χωρίς κατάλληλη ευρετηρίαση ή δομές δεδομένων βελτιστοποιημένες για τέτοιες λειτουργίες, οι εφαρμογές λογισμικού μπορεί να χρειαστεί να σαρώσουν ολόκληρο το σύνολο δεδομένων διαδοχικά, γεγονός που μπορεί να οδηγήσει σε μεγάλους χρόνους επεξεργασίας, ειδικά για σύνθετα ερωτήματα ή αναλύσεις. Επιπλέον, τα αρχεία CSV δεν διαθέτουν ενσωματωμένους μηχανισμούς συμπίεσης δεδομένων, επομένως μεγάλα σύνολα δεδομένων που αποθηκεύονται σε μορφή CSV ενδέχεται να καταναλώνουν σημαντικό χώρο στο δίσκο, οδηγώντας σε αυξημένη επιβάρυνση εισόδου/εξόδου και βραδύτερες λειτουργίες ανάγνωσης/εγγραφής, ιδιαίτερα σε συστήματα αποθήκευσης με περιορισμένο εύρος ζώνης ή απόδοση.

Ζητήματα ασφαλείας: Ενώ η μορφή αρχείου CSV προσφέρει πολλά πλεονεκτήματα, έχει επίσης πιθανούς κινδύνους ασφαλείας που πρέπει να ληφθούν υπόψη. Μία από τις πιο αξιοσημείωτες ευπάθειες είναι τα CSV Injections. Το CSV Injection είναι μια ευπάθεια ασφαλείας που εμφανίζεται όταν ένα αρχείο CSV που περιέχει κακόβουλη είσοδο ερμηνεύεται από ένα πρόγραμμα υπολογιστικών φύλλων. Συγκεκριμένα, αυτές οι επιθέσεις μπορούν να εκτελεστούν όταν τα δεδομένα CSV περιλαμβάνουν καταχωρήσεις που ερμηνεύονται ως τύποι από αυτά τα προγράμματα υπολογιστικών φύλλων. Επιπλέον, ακολουθούν ορισμένες άλλες ανησυχίες σχετικά με την ασφάλεια χρήσης των CSV αρχείων:

- **Ακεραιότητα δεδομένων:** Τα αρχεία CSV δεν υποστηρίζουν κρυπτογράφηση ή προστασία με κωδικό πρόσβασης. Αυτό σημαίνει ότι μπορούν να ανοίξουν από οποιονδήποτε έχει πρόσβαση στο εκάστοτε αρχείο, θέτοντας ενδεχομένως σε κίνδυνο την ακεραιότητα και την εμπιστευτικότητα των δεδομένων.
- **Διαρροή δεδομένων:** Δεδομένου ότι τα αρχεία CSV περιέχουν απλό κείμενο, μπορούν εύκολα να διαβαστούν εάν υποκλαπούν κατά τη μετάδοση ή εάν αποθηκευτούν χωρίς επαρκή προστασία. Αυτό θα μπορούσε να οδηγήσει σε διαρροή δεδομένων, ειδικά εάν τα δεδομένα είναι ευαίσθητα ή εμπιστευτικά.
- **Έλλειψη επικύρωσης δεδομένων:** Τα αρχεία CSV δεν διαθέτουν ενσωματωμένους μηχανισμούς επικύρωσης δεδομένων. Αυτό σημαίνει ότι θα μπορούσαν να περιέχουν λανθασμένα ή κακόβουλα δεδομένα, τα οποία θα μπορούσαν να οδηγήσουν σε ανακριβείς αναλύσεις ή ευπάθειες ασφαλείας.

Προκειμένου να μετριάσουν αυτοί οι κίνδυνοι ασφαλείας, είναι σημαντικό να επικυρωθούν και να καθαριστούν τα δεδομένα προτού γραφτούν σε ένα αρχείο CSV, να αντιμετωπίζονται όλα τα αρχεία CSV από μη αξιόπιστες πηγές ως δυνητικά επικίνδυνα και να εξετάζεται η χρήση πιο ασφαλών μορφών αρχείων κατά το χειρισμό ευαίσθητων δεδομένων.

2.2. Γλώσσα Python

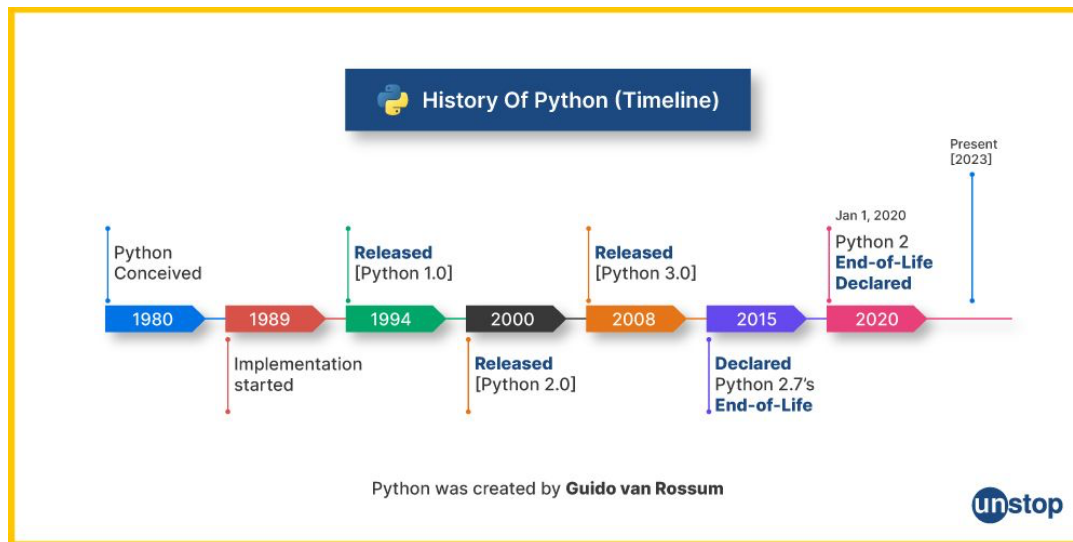
2.2.1. Σύντομη Ιστορική Αναδρομή

Η Python [3] - [10], μια γλώσσα προγραμματισμού υψηλού επιπέδου, σχεδιάστηκε από τον Guido van Rossum στα τέλη της δεκαετίας του 1980. Παίρνοντας το όνομά της από τη βρετανική κωμική ομάδα Monty Python, η Python στόχευε να τονίσει την αναγνωσιμότητα του κώδικα με την καθαρή και απλή σύνταξή της. Κατά τη διάρκεια της δεκαετίας του 1990, η Python κέρδισε το πρώιμο, αλλά περιορισμένο ενδιαφέρον των προγραμματιστών λόγω της ευελιξίας και της ευκολίας χρήσης της, ειδικά στη συγγραφή scripts, την ανάπτυξη ιστοσελίδων και την επιστημονική πληροφορική.

Στις αρχές της δεκαετίας του 2000, η δημοτικότητα της Python αυξήθηκε με την εμφάνιση frameworks όπως το Django και το Flask, διευκολύνοντας την ανάπτυξη κώδικα στο Διαδίκτυο. Η απλότητα και η ευελιξία της προσέλκυαν τόσο αρχάριους όσο και έμπειρους προγραμματιστές. Η κοινότητα της Python αναπτύχθηκε γρήγορα, προωθώντας τη συνεργατική ανάπτυξη και τη δημιουργία ενός τεράστιου οικοσυστήματος βιβλιοθηκών και εργαλείων. Η υιοθέτηση της γλώσσας επεκτάθηκε πέρα από τους παραδοσιακούς τομείς στην τεχνητή νοημοσύνη, την επιστήμη των δεδομένων και τη μηχανική μάθηση.

Η Python 2.X κυριάρχησε στο τοπίο μέχρι το 2010 και το επίσημο τέλος της δόθηκε το 2020. Η μετάβαση στην Python 3.X ήταν αρχικά αργή λόγω προβλημάτων συμβατότητας, αλλά τα οφέλη της βελτιωμένης απόδοσης, της ασφάλειας και της υποστήριξης της Python 3 για σύγχρονα πρότυπα προγραμματισμού σταδιακά υπερέβησαν τις προκλήσεις της μετάβασης. Η Python 3 εμφανίστηκε το 2008 και έχει γίνει το πλέον διαδεδομένο πρότυπο, με τους προγραμματιστές να αποδέχονται τα χαρακτηριστικά της και την κοινότητα να συσπειρώνεται πίσω από τη συνεχιζόμενη ανάπτυξη και υποστήριξή της.

Από τις αρχές της δεκαετίας του 2020, η Python παραμένει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού παγκοσμίως, τροφοδοτώντας ένα ευρύ φάσμα εφαρμογών, από την ανάπτυξη ιστοσελίδων και την επιστημονική πληροφορική έως την τεχνητή νοημοσύνη και τη μηχανική μάθηση. Η απλότητα, η αναγνωσιμότητα και το εκτεταμένο οικοσύστημα υποστήριξης που διαθέτει, συνεχίζουν να προσελκύουν προγραμματιστές σε διάφορους τομείς.



Εικόνα 1: Ιστορική εξέλιξη της γλώσσας Python (Πηγή: <https://unstop.com/blog/what-is-python>)

2.2.2. Χαρακτηριστικά Γλώσσας Python

Εύκολη στην κατανόηση: Η Python θεωρείται μία από τις πιο φιλικές προς τον χρήστη γλώσσες προγραμματισμού. Εντός ολίγων ημερών, μπορεί κανείς να κατανοήσει γρήγορα τις βασικές αρχές της Python, ώστε να αποκτήσει άνεση με το συντακτικό της και να αρχίσει να δημιουργεί απλά προγράμματα. Ωστόσο, μπορεί να απαιτηθεί λίγος χρόνος ώστε να αποκτήσει εξειδικευμένες γνώσεις στην Python και να μπορεί να δημιουργήσει πιο περίπλοκα προγράμματα. Γενικά μιλώντας, σε σύγκριση με άλλες γλώσσες προγραμματισμού είναι ευκολότερη στην κατανόηση.

Απλός κώδικας: Ένα άλλο σημαντικό χαρακτηριστικό της Python είναι ότι έχει μια πολύ απλή σύνταξη. Συνήθως αποτελείται από όρους αγγλικής γλώσσας, εύκολα κατανοητούς. Επιπλέον, σε αντίθεση με άλλες γλώσσες προγραμματισμού (όπως C, C++, Java κλπ.), οι προγραμματιστές δεν χρειάζεται να ανησυχούν για το άνοιγμα ή το κλείσιμο των αγκυλών, καθώς δεν απαιτούνται για τη συγγραφή προγραμμάτων. Ωστόσο θα πρέπει να είναι ιδιαίτερα προσεκτικοί όσον αφορά τη στοίχιση των τμημάτων κώδικα που χρησιμοποιούν.

Ακολουθεί ένα απλό παράδειγμα για την υποστήριξη του παραπάνω ισχυρισμού:

```
if ( x < y ):
    print( " x is greater than y " )
else:
    print( " y is greater than x " )
```

Κώδικας 1: Παράδειγμα χρήσης indentation στην Python

Ερμηνευσιμότητα: Η Python είναι μια ερμηνευόμενη γλώσσα, που σημαίνει ότι, σε αντίθεση με άλλες γλώσσες προγραμματισμού, ο κώδικάς της εκτελείται γραμμή προς γραμμή και όχι όλες οι γραμμές ταυτόχρονα. Αυτή η προσέγγιση γραμμής προς γραμμή καθιστά επίσης τον εντοπισμό σφαλμάτων του κώδικα απλό.

Αντικειμενοστρεφής Προγραμματισμός: Οι ιδέες των κλάσεων και των αντικειμένων χρησιμοποιούνται και στην περίπτωση της συγκεκριμένης γλώσσας. Οι κλάσεις λειτουργούν ως το πρότυπο ενός αντικειμένου, περιέχοντας τα δεδομένα καθώς και τις μεθόδους που τα χειρίζονται. Ο στόχος του αντικειμενοστρεφούς προγραμματισμού είναι να υποστηρίξει κώδικα που είναι τόσο επαναχρησιμοποιήσιμος όσο και καλά αφηρημένος. Η υποστήριξη της αντικειμενοστρέφειας είναι ένα από τα πιο σημαντικά χαρακτηριστικά της γλώσσας Python. Επιπλέον, η Python είναι συμβατή με διάφορες αντικειμενοστρεφείς λειτουργίες προγραμματισμού, συμπεριλαμβανομένης της κληρονομικότητας, του πολυμορφισμού, των κλάσεων και της ενσωμάτωσης δεδομένων.

Ακολουθεί ένα παράδειγμα χρήσης κλάσεων στην γλώσσα Python:

```
class Employee:
    def __init__( self, name, salary ):
        self.name = name
        self.salary = salary

    def print_details( self ):
        print( " My name is " + self.name )

e1 = Employee( " John " , 1200 )
e1.print_details( )
```

Κώδικας 2: Παράδειγμα αντικειμενοστρέφειας στην Python

Ανοιχτού κώδικα: Η Python είναι μια γλώσσα ανοιχτού κώδικα, πράγμα που σημαίνει ότι μπορεί να χρησιμοποιηθεί χωρίς περιορισμούς όσον αφορά τα πνευματικά δικαιώματα και σε οποιοδήποτε είδος λειτουργικού συστήματος. Πιο συγκεκριμένα, από την επίσημη ιστοσελίδα της, η Python μπορεί να μεταφορτωθεί από οποιονδήποτε, μαζί με την τεκμηρίωση και τις βιβλιοθήκες της.

Πολυπλατφορμική Γλώσσα: Ένα από τα άλλα σημαντικά χαρακτηριστικά της γλώσσας Python αποτελεί το γεγονός ότι είναι από την φύση της cross-platform. Για παράδειγμα, η σύνταξη ενός προγράμματος Python σε Mac μπορεί να εκτελεστεί χωρίς τροποποίηση σε Linux, Windows ή οποιοδήποτε άλλο λειτουργικό σύστημα. Αυτό συμβαίνει επειδή ο κώδικας Python εκτελείται αφού μεταφραστεί για πρώτη φορά σε μια ενδιάμεση μορφή που ονομάζεται Bytecode.

Γλώσσα Υψηλού Επιπέδου: Η Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου, επομένως είναι εύκολο για τους χρήστες να γράφουν, να διαβάζουν, καθώς και να ερμηνεύουν κώδικα. Ένας προγραμματιστής που χρησιμοποιεί μια γλώσσα προγραμματισμού υψηλού επιπέδου μπορεί να δημιουργήσει κώδικα λιγότερο εξαρτώμενο από τον τύπο του υλικού του υπολογιστικού συστήματος. Επιπλέον, αξίζει να αναφερθεί ότι η Python είναι μια εξαιρετικά ισχυρή αφηρημένη γλώσσα. Ο προγραμματιστής δεν χρειάζεται να ανησυχεί για τον τύπο επεξεργαστή του υπολογιστικού συστήματος, τη διαχείριση μνήμης ή την αρχιτεκτονική κατά την ανάπτυξη του κώδικα.

Υποστήριξη προγραμματισμού GUI: Αυτό είναι ένα από τα πιο σημαντικά χαρακτηριστικά της γλώσσας προγραμματισμού Python. Η Python χρησιμοποιεί πολλές βιβλιοθήκες GUI, όπως είναι οι PSide2, PyQt5, Tkinter κλπ. Μια γραφική διεπαφή χρήστη (GUI) διευκολύνει την αλληλεπίδραση του χρήστη με ένα πρόγραμμα και το σύστημα. Τα χρησιμοποιούμενα modules προσφέρουν δυνατότητες που καθιστούν την ανάπτυξη λογισμικού με χρήση γραφικών γρήγορη και απλή.

Ακολουθεί ένα πολύ απλό πρόγραμμα σε γλώσσα Python με χρήση της Tkinter, το οποίο δημιουργεί ένα στοιχείο τύπου combo box, προκειμένου να γίνει κατανοητή η απλότητα και η χρησιμότητα της συγκεκριμένης δημοφιλούς βιβλιοθήκης:

```
import tkinter as tk
from tkinter import ttk

def on_select( event ):
    selected= combo_box.get( )
    label.config( text= " Selected Item: " + selected )

root = tk.Tk( )
root.title( " this is a demo " )

label = tk.Label( root, text= " Item that was selected: " )
label.pack( pady=8 )

combo_box = ttk.Combobox( root, values=[ " Option 1 " , " Option 2 " , " Option 3 "
] )
combo_box.pack( pady=4 )

combo_box.set( " Option 2 " )
```

```
combo_box.bind( "<<ComboboxSelected>>" , on_select )  
  
root.mainloop( )
```

Κώδικας 3: Παράδειγμα χρήσης της βιβλιοθήκης tkinter στην Python

Ποικιλία εγκατεστημένων βιβλιοθηκών: Η Python προσφέρει ένα ευρύ φάσμα modules και συναρτήσεων και αποτελείται κυρίως από βιβλιοθήκες πολλαπλών πλατφορμών. Αυτές οι βιβλιοθήκες λειτουργούν με διάφορα λειτουργικά συστήματα, συμπεριλαμβανομένων των Windows, Mac, Linux και άλλων. Επειδή υπάρχουν τόσες πολλές διαθέσιμες βιβλιοθήκες, οι προγραμματιστές μπορούν να εισάγουν και να χρησιμοποιήσουν την απαραίτητη λειτουργικότητα, αντί να γράφουν κώδικα από την αρχή. Βέβαια, σε περίπτωση που επιθυμούν επιπλέον δυνατότητες, μπορούν να επιλέξουν να εγκαταστήσουν άλλα πακέτα που δεν περιλαμβάνονται στην προεπιλεγμένη βιβλιοθήκη.

Ισχυρή υποστήριξη κοινότητας: Η Python έχει γίνει όλο και πιο δημοφιλής με τα χρόνια, με μία από τις μεγαλύτερες κοινότητες στο Meetup και στο Stack Overflow. Υπάρχει μια μεγάλη και ενεργή κοινότητα Python που είναι πάντα πρόθυμη να βοηθήσει με ερωτήσεις. Αυτοί οι ιστότοποι περιέχουν πληθώρα απαντήσεων σε συχνές ερωτήσεις σχετικά με την Python, τις οποίες οι χρήστες της Python μπορούν να χρησιμοποιήσουν ανάλογα με τις εκάστοτε ανάγκες τους.

2.2.3. Εφαρμογές γλώσσας Python

2.2.3.1. Web Development

Προσφέρει πολλές επιλογές για την ανάπτυξη εφαρμογών διαδικτύου [11] – [15], συμπεριλαμβανομένων frameworks όπως τα: Django, Pyramid, Flask και Bottle, καθώς και προηγμένα συστήματα διαχείρισης περιεχομένου όπως το Plone και το Django CMS. Αυτά τα frameworks περιέχουν βιβλιοθήκες και modules που απλοποιούν εργασίες όπως διαχείριση περιεχομένου, αλληλεπίδραση βάσης δεδομένων και διασύνδεση με πρωτόκολλα Διαδικτύου όπως HTTP, SMTP, XML, JSON, FTP, IMAP και POP. Τα web frameworks της Python είναι γνωστά για την ασφάλεια, την επεκτασιμότητα και την ευελιξία τους. Το ευρετήριο πακέτων της γλώσσας περιλαμβάνει χρήσιμες βιβλιοθήκες όπως: BeautifulSoup, Paramiko, Feedparser και Twisted Python.

Η Python είναι μια εξαιρετική επιλογή για ανάπτυξη εφαρμογών ιστού λόγω του ευρύτερου φάσματος frameworks όπως το Flask, το Django και το Bottle, τα οποία εξορθολογίζουν τις εργασίες των προγραμματιστών. Η γλώσσα παρέχει επίσης ενσωματωμένες βιβλιοθήκες και εργαλεία που καθιστούν τη διαδικασία ανάπτυξης εφαρμογών ιστού αβίαστη. Τα χαρακτηριστικά της Python όπως η οπτικοποίηση, η ευκολία στην ανάπτυξη, η βελτιωμένη ασφάλεια και η γρήγορη διαδικασία ανάπτυξης συμβάλλουν στη δημιουργία των καλύτερων δυνατών εφαρμογών.

Η χρήση της για ανάπτυξη εφαρμογών ιστού προσφέρει οφέλη όπως ασφάλεια, επεκτασιμότητα και ευκολία στη διαδικασία ανάπτυξης. Η Python υποστηρίζει διάφορα πρωτόκολλα ιστού, συμπεριλαμβανομένων των: HTTP, HTTPS συχνά χρησιμοποιούμενων πρωτοκόλλων ηλεκτρονικού ταχυδρομείου και FTP. Διαθέτει επίσης μια μεγάλη συλλογή βιβλιοθηκών που ενισχύουν τη λειτουργικότητα των εφαρμογών ιστού και απλοποιούν την ανάπτυξη.

Το ευρύ φάσμα frameworks, CMSs και βιβλιοθηκών απλοποιεί τη διαδικασία ανάπτυξης, εξασφαλίζει ασφάλεια και επιτρέπει τη δημιουργία κλιμακωτών και οπτικά ελκυστικών εφαρμογών ιστού. Η χρήση της Python για την ανάπτυξη ιστοσελίδων δημιουργεί σημαντικές δυνατότητες για τους προγραμματιστές, ανεξάρτητα από το επίπεδο δεξιοτήτων τους.

2.2.3.2. Ανάπτυξη παιχνιδιών

Η Python είναι μια ευέλικτη γλώσσα προγραμματισμού που δεν είναι μόνο δημοφιλής για την ανάπτυξη εφαρμογών ιστού, αλλά βρίσκει επίσης σημαντική χρήση στην ανάπτυξη παιχνιδιών [16] – [19]. Προσφέρει ένα ευρύ φάσμα βιβλιοθηκών και εργαλείων ειδικά σχεδιασμένων για τη δημιουργία διαδραστικών παιχνιδιών. Μερικές αξιοσημείωτες βιβλιοθήκες που χρησιμοποιούνται στην ανάπτυξη παιχνιδιών σε γλώσσα Python περιλαμβάνουν το PySoy, το PyGame, το Pycap, το Construct, το Panda3D και το PyOpenGL.

Τα πλεονεκτήματά της στην ανάπτυξη παιχνιδιών υπερβαίνουν τη δημιουργία των ίδιων των παιχνιδιών. Οι σχεδιαστές μπορούν να χρησιμοποιήσουν την Python για να αναπτύξουν εργαλεία που απλοποιούν εργασίες όπως ο σχεδιασμός επιπέδων και η δημιουργία δέντρου διαλόγου. Αυτά τα εργαλεία μπορούν στη συνέχεια να εξαχθούν σε μορφές συμβατές με τις κύριες μηχανές παιχνιδιών. Η Python χρησιμεύει επίσης ως γλώσσα σεναρίων για πολλές μηχανές παιχνιδιών, ενισχύοντας περαιτέρω τη χρησιμότητά της στη διαδικασία ανάπτυξης παιχνιδιών.

Το πλούσιο οικοσύστημα βιβλιοθηκών της Python και η ευελιξία της την καθιστούν κατάλληλη επιλογή για την ανάπτυξη παιχνιδιών. Η επιτυχής εφαρμογή της σε αξιοσημείωτα παιχνίδια και η ικανότητά της να απλοποιεί συγκεκριμένες ενέργειες και εργασίες στο σχεδιασμό παιχνιδιών την καθιστούν πολύτιμο εργαλείο τόσο για προγραμματιστές όσο και για σχεδιαστές παιχνιδιών.

2.2.3.3. Μηχανική μάθηση και Τεχνητή Νοημοσύνη

Τα έργα τεχνητής νοημοσύνης και μηχανικής μάθησης ([20] – [26]) διαφέρουν από τα παραδοσιακά έργα λογισμικού όσον αφορά τα εργαλεία, τις τεχνολογίες και το σύνολο δεξιοτήτων που απαιτούνται για την ανάπτυξή τους. Η Python έχει αναδειχθεί ως μια αγαπημένη γλώσσα για τους επαγγελματίες της τεχνητής νοημοσύνης και μηχανικής μάθησης λόγω της σταθερότητας, της ασφάλειας, της ευελιξίας και της διαθεσιμότητας εργαλείων που ανταποκρίνονται στις μοναδικές απαιτήσεις τέτοιων έργων. Η απλότητα, η συνέπεια, η ανεξαρτησία της πλατφόρμας, η εκτεταμένη συλλογή βιβλιοθηκών και η ενεργή κοινότητα της Python την καθιστούν κατάλληλη για την ανάπτυξη εφαρμογών τεχνητής νοημοσύνης και μηχανικής μάθησης.

Η Python προσφέρει μια σειρά από ισχυρές βιβλιοθήκες, συμπεριλαμβανομένων των SciPy για προηγμένη υπολογιστική μηχανικής, Pandas για ανάλυση δεδομένων γενικής χρήσης, Seaborn για οπτικοποίηση δεδομένων και δημοφιλή frameworks όπως Keras, TensorFlow και Scikit-learn για εργασίες μηχανικής μάθησης. Επιπλέον, το NumPy παρέχει επιστημονικές δυνατότητες ανάλυσης δεδομένων υψηλής απόδοσης. Άλλες βιβλιοθήκες που βασίζονται σε Python όπως NLTK, Caffe, PyTorch και Accord.NET είναι επίσης πολύτιμοι πόροι για έργα τεχνητής νοημοσύνης και μηχανικής μάθησης.

Επιπλέον, τα ενσωματωμένα εργαλεία και βιβλιοθήκες της Python απλοποιούν την ανάπτυξη αλγορίθμων μηχανικής μάθησης και τεχνητής νοημοσύνης. Ο συνοπτικός και ευανάγνωστος κώδικας της επιτρέπει στους προγραμματιστές να δημιουργούν πιο εύκολα πολύπλοκους αλγόριθμους. Επιπλέον, η Python χρησιμοποιείται ευρέως στον τομέα της τεχνητής νοημοσύνης και μηχανικής μάθησης, όπως αποδεικνύεται από τη θέση της ως η δεύτερη πιο δημοφιλής γλώσσα στο GitHub και η πιο δημοφιλής γλώσσα για μηχανική μάθηση.

Η τεχνητή νοημοσύνη και η μηχανική μάθηση έχουν κερδίσει σημαντική προσοχή και είναι από τα πιο σημαντικά επιστημονικά πεδία αυτής της δεκαετίας. Η Python, μαζί με άλλες γλώσσες προγραμματισμού, έχει βιώσει μια αύξηση στη χρήση της για την ανάπτυξη λύσεων στα συγκεκριμένα πεδία.

2.2.3.4. Επιστήμη δεδομένων και οπτικοποίηση δεδομένων

Τα δεδομένα διαδραματίζουν κρίσιμο ρόλο στον σημερινό σύγχρονο κόσμο λόγω της ικανότητάς τους να παρέχουν πολύτιμες γνώσεις σχετικά με τις προτιμήσεις και τις συμπεριφορές των ανθρώπων. Αυτός είναι ο λόγος για τον οποίο ο τομέας της επιστήμης των δεδομένων περιστρέφεται γύρω από τη συλλογή και την ανάλυση κρίσιμων πληροφοριών σχετικά με τα άτομα και τις αλληλεπιδράσεις τους με τον κόσμο. Η διαδικασία της επιστήμης των δεδομένων περιλαμβάνει διάφορα βήματα, συμπεριλαμβανομένης της αναγνώρισης προβλημάτων, της συλλογής δεδομένων, της επεξεργασίας δεδομένων, της εξερεύνησης δεδομένων, της ανάλυσης δεδομένων και της απεικόνισης δεδομένων.

Μέσα στο οικοσύστημα της γλώσσας Python, υπάρχουν πολλές διαθέσιμες βιβλιοθήκες για την αποτελεσματική αντιμετώπιση των προκλήσεων της επιστήμης των δεδομένων ([27] – [30]). Αυτές οι βιβλιοθήκες ειδικεύονται στη δημιουργία και την τελειοποίηση μοντέλων βαθιάς μάθησης και μηχανικής μάθησης, καθώς και στην εκτέλεση πολύπλοκων χειρισμών και υπολογισμών δεδομένων.

Η οπτικοποίηση των δεδομένων καθίσταται απαραίτητη όταν πρόκειται για τη μεταφορά ευρημάτων σε ενδιαφερόμενους και ομάδες. Η Python προσφέρει μια σειρά από ευρέως χρησιμοποιούμενα εργαλεία απεικόνισης δεδομένων, συμπεριλαμβανομένων των Plotly, Matplotlib, Seaborn, Ggplot και Geoplotlib. Αυτές οι βιβλιοθήκες επιτρέπουν τη δημιουργία οπτικά ελκυστικών αναπαραστάσεων δεδομένων, διευκολύνοντας την αποτελεσματική επικοινωνία πληροφοριών και ευρημάτων.

2.2.3.5. Desktop GUI

Η Python δεν είναι γνωστή μόνο για τη σύνταξη που μοιάζει με τα αγγλικά, αλλά και για τη ρεθωτή αρχιτεκτονική και τη συμβατότητα μεταξύ πλατφορμών, γεγονός που την καθιστά δημοφιλή επιλογή για την ανάπτυξη εφαρμογών GUI ([31] – [34]) που βασίζονται σε επιτραπέζιους υπολογιστές. Τα πλούσια εργαλεία επεξεργασίας κειμένου ενισχύουν περαιτέρω την καταλληλότητά της για ανάπτυξη GUI.

Προσφέρει ένα ευρύ φάσμα εργαλείων GUI και frameworks που απλοποιούν τη διαδικασία δημιουργίας εφαρμογών επιφάνειας εργασίας. Μερικά από τα αξιοσημείωτα GUI frameworks με βάση την Python περιλαμβάνουν τα PyQt, PyGtk, Kivy, Tkinter, wxPython, PyGUI και PySide. Αυτά τα frameworks παρέχουν στους προγραμματιστές τα απαραίτητα εργαλεία για τη δημιουργία εξαιρετικά λειτουργικών και φιλικών προς το χρήστη γραφικών διεπαφών χρήστη (GUI).

Επιπλέον, η αρθρωτή προσέγγιση προγραμματισμού της Python και η εύκολη στην κατανόηση σύνταξη αποτελούν βασικούς παράγοντες για την ανάπτυξη ανταποκρινόμενων και αποτελεσματικών GUI. Αυτά τα χαρακτηριστικά απλοποιούν τη διαδικασία ανάπτυξης και συμβάλλουν στη συνολική απόδοση της εφαρμογής GUI.

2.2.3.6. Επιχειρηματικές εφαρμογές

Η Python είναι κατάλληλη για την ανάπτυξη εξατομικευμένων λύσεων υψηλής απόδοσης, τόσο για καταναλωτές όσο και για επιχειρήσεις. Οι επιχειρηματικές εφαρμογές ([35] , [36]) έχουν ιδιαίτερα χαρακτηριστικά που τις διακρίνουν από τις εφαρμογές που χρησιμοποιούν απλοί χρήστες. Επικεντρώνονται σε συγκεκριμένα χαρακτηριστικά προσαρμοσμένα στις ανάγκες ενός οργανισμού αντί να προσφέρουν ένα ευρύ φάσμα λειτουργιών.

Η ασφάλεια είναι μια κρίσιμη πτυχή κάθε εφαρμογής και έχει ιδιαίτερη σημασία στις επιχειρηματικές εφαρμογές λόγω της εξάρτησης από την ασφάλεια των πληροφοριών. Η Python προσφέρει ισχυρά χαρακτηριστικά ασφαλείας, καθιστώντας την ελκυστική επιλογή για την ανάπτυξη ασφαλών επιχειρηματικών εφαρμογών.

Η επεκτασιμότητα είναι ένα άλλο βασικό στοιχείο για τις επιχειρηματικές εφαρμογές, καθώς πρέπει να προσαρμοστούν στην ανάπτυξη και να παρακολουθούν τις μεταβαλλόμενες απαιτήσεις. Η Python υπερέχει στην οικοδόμηση επεκτάσιμων εφαρμογών, επιτρέποντας στις επιχειρήσεις να κλιμακώσουν αποτελεσματικά τις δραστηριότητές τους.

2.2.4. Περιορισμοί γλώσσας Python

Απόδοση και ταχύτητα: Η Python είναι μια διερμηνευόμενη γλώσσα που μεταγλωττίζεται κατά το χρόνο εκτέλεσης από μια εικονική μηχανή ή έναν διερμηνευτή. Αυτό την κάνει πιο αργή και λιγότερο αποτελεσματική από τις μεταγλωττισμένες γλώσσες όπως ή C ή C++.

Διαχείριση μνήμης: Χρησιμοποιεί έναν «συλλέκτη απορριμμάτων» για να διαχειρίζεται τη μνήμη και να διαγράφει αυτόματα τα μη χρησιμοποιούμενα αντικείμενα. Ενώ αυτό μπορεί να διευκολύνει τη σύνταξη και τη διατήρηση κώδικα, μπορεί επίσης να οδηγήσει σε αναποτελεσματικότητα και επιβράδυνση εάν δεν χρησιμοποιηθεί σωστά. Για το λόγο αυτό, θα πρέπει να λαμβάνονται σοβαρά υπόψη όλα τα παραπάνω για τη δημιουργία εφαρμογών υψηλών απαιτήσεων, όπως είναι οι επιχειρηματικές εφαρμογές.

Στατικοί τύποι: Είναι μια δυναμική γλώσσα, που σημαίνει ότι οι μεταβλητές δεν έχουν σταθερό τύπο και μπορούν να εκχωρηθούν με οποιαδήποτε τιμή ανά πάσα στιγμή. Ενώ αυτό μπορεί να είναι βολικό και ευέλικτο, μπορεί επίσης να κάνει τη σύλληψη σφαλμάτων κατά τη στιγμή της σύνταξης δύσκολη σε περιπτώσεις που χρησιμοποιείται μια μεταβλητή, η οποία μπορεί να λάβει διαφορετικούς τύπους τιμών. Αντίθετα, άλλες γλώσσες όπως η Java ή η C++ απαιτούν οι μεταβλητές να δηλώνονται ρητά με έναν συγκεκριμένο τύπο, ο οποίος μπορεί να βοηθήσει στην πρόληψη σφαλμάτων και στη βελτίωση της ποιότητας του κώδικα.

Υπερφόρτωση τελεστών: Ως γλώσσα δεν υποστηρίζει υπερφόρτωση τελεστών, επομένως οι προγραμματιστές δεν μπορούν να ορίσουν προσαρμοσμένη συμπεριφορά για ενσωματωμένους τελεστές. Αυτό μπορεί να δυσκολέψει τον ορισμό προσαρμοσμένων τύπων ή δομών δεδομένων που χρησιμοποιούν αυτούς τους τελεστές.

Βασική Βιβλιοθήκη: Η βασική βιβλιοθήκη της Python δεν είναι τόσο εκτεταμένη όσο άλλων γλωσσών, όπως η Java ή η C++. Αυτό σημαίνει ότι οι προγραμματιστές μπορεί να χρειαστεί να βασίζονται σε βιβλιοθήκες ή frameworks τρίτων για να έχουν πρόσβαση σε ορισμένες λειτουργίες, οι οποίες μπορούν να προσθέσουν πολυπλοκότητα και εξαρτήσεις στα έργα τους.

Δυναμική φύση: Η δυναμική φύση της Python μπορεί να δυσκολέψει την εκτέλεση στατικής ανάλυσης ή βελτιστοποίησης. Αυτό μπορεί να δυσκολέψει τη βελτιστοποίηση της απόδοσης ή της αποτελεσματικότητας του κώδικα Python, καθιστώντας πιο δύσκολη την ενσωμάτωση με άλλες γλώσσες ή εργαλεία.

2.3. Anaconda Spyder

Το Spyder [37] – [39], συντομογραφία του "Scientific Python Development Environment" (Περιβάλλον Ανάπτυξης Python για Επιστημονική Χρήση), είναι ένα ανοιχτού κώδικα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) που προσαρμόζεται ειδικά για την ανάλυση δεδομένων και την επιστημονική υπολογιστική με τη χρήση της Python. Αποτελεί μέρος της διανομής Anaconda, η οποία περιλαμβάνει ένα ολοκληρωμένο σύνολο εργαλείων και βιβλιοθηκών για ανάλυση δεδομένων, μηχανική μάθηση και επιστημονική έρευνα. Το Spyder προσφέρει έναν μοναδικό συνδυασμό προηγμένων δυνατοτήτων επεξεργασίας, διαδραστικού εντοπισμού σφαλμάτων και οπτικοποίησης δεδομένων, καθιστώντας το μια προτιμώμενη επιλογή για επιστήμονες, μηχανικούς και αναλυτές δεδομένων.

Τα κύρια χαρακτηριστικά του Anaconda Spyder συνοψίζονται στη συνέχεια:

Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE): Το Spyder παρέχει μια ισχυρή εμπειρία IDE με έναν ισχυρό επεξεργαστή κώδικα που υποστηρίζει χαρακτηριστικά όπως αυτόματη συμπλήρωση κώδικα και έλεγχο σφαλμάτων σε πραγματικό χρόνο. Έχει σχεδιαστεί για να διευκολύνει την ταχεία και αποδοτική εκτέλεση, ιδιαίτερα για πολύπλοκες επιστημονικές εργασίες.

Διαδραστική Κονσόλα Python: Διαθέτει μια διαδραστική κονσόλα Python που επιτρέπει την εκτέλεση κώδικα σε πραγματικό χρόνο. Αυτή η κονσόλα επιτρέπει στους χρήστες να εκτελούν αποσπάσματα κώδικα ή ολόκληρα σενάρια. Η κονσόλα υποστηρίζει επιπλέον το IPython, το οποίο προσφέρει προηγμένα χαρακτηριστικά όπως διαδραστικά widgets και πλούσιες αναπαραστάσεις μέσω.

Εξερεύνηση Μεταβλητών: Ένα από τα εξαιρετικά χαρακτηριστικά του Spyder είναι η εξερεύνηση μεταβλητών (variable exploring), η οποία παρέχει ένα γραφικό περιβάλλον για την επιθεώρηση μεταβλητών, πινάκων δεδομένων και άλλων δομών δεδομένων σε πραγματικό χρόνο. Αυτό το χαρακτηριστικό είναι ιδιαίτερα χρήσιμο για την ανάλυση δεδομένων και τον εντοπισμό σφαλμάτων και επιτρέπει στους χρήστες να οπτικοποιούν και να χειρίζονται δεδομένα απευθείας μέσα στο IDE.

Ενσωματωμένα Εργαλεία Εντοπισμού Σφαλμάτων: Περιλαμβάνει έναν ενσωματωμένο εντοπιστή σφαλμάτων που βοηθά τους χρήστες να εντοπίζουν και να διορθώνουν σφάλματα στον κώδικά τους. Ο εντοπιστής σφαλμάτων υποστηρίζει τυπικές λειτουργίες εντοπισμού, όπως ορισμός σημείων διακοπής, βήμα προς βήμα εκτέλεση κώδικα και επιθεώρηση της στοίβας κλήσεων. Αυτή η ενσωμάτωση διευκολύνει τον εντοπισμό σφαλμάτων και τη βελτιστοποίηση πολύπλοκων σεναρίων.

Υποστήριξη Επιστημονικών Βιβλιοθηκών: Ως μέρος της διανομής Anaconda, το Spyder ενσωματώνεται με δημοφιλείς επιστημονικές βιβλιοθήκες όπως οι NumPy, SciPy, Pandas και Matplotlib. Αυτή ή ενσωμάτωση καθιστά ένα ισχυρό εργαλείο για την εκτέλεση ανάλυσης δεδομένων, στατιστικής μοντελοποίησης και οπτικοποίησης χωρίς να απαιτείται επιπλέον ρύθμιση.

Επεκτάσιμη και Πλήρως Παραμετροποιήσιμη Αρχιτεκτονική: Το Spyder είναι εξαιρετικά παραμετροποιήσιμο, επιτρέποντας στους χρήστες να επεκτείνουν τη λειτουργικότητά του μέσω πρόσθετων (plugins). Οι χρήστες μπορούν να προσθέσουν νέα εργαλεία, σενάρια και ροές εργασίας για να προσαρμόσουν το περιβάλλον στις συγκεκριμένες ανάγκες τους. Αυτή η ευελιξία καθιστά το Spyder προσαρμοστικό σε ένα ευρύ φάσμα επιστημονικών και αναλυτικών εργασιών.

Σύστημα Τεκμηρίωσης και Βοήθειας: Το Spyder διαθέτει ένα ενσωματωμένο σύστημα βοήθειας που παρέχει γρήγορη πρόσβαση στην τεκμηρίωση και στις αναφορές λειτουργιών. Αυτό περιλαμβάνει βοήθεια ευαίσθητη στο περιεχόμενο για λειτουργίες και βιβλιοθήκες της Python διευκολύνοντας τους χρήστες να βρίσκουν τις πληροφορίες που χρειάζονται κατά την συγγραφή κώδικα.

Υποστήριξη Πολλαπλών Γλωσσών: Αν και το Spyder είναι κυρίως ένα IDE για Python, υποστηρίζει και άλλες γλώσσες όπως η R μέσω πρόσθετων. Αυτό το καθιστά ένα ευέλικτο εργαλείο για χρήστες που εργάζονται σε πολυγλωσσικά περιβάλλοντα ή που μεταβαίνουν από άλλες γλώσσες προγραμματισμού στην Python.

ΚΕΦΑΛΑΙΟ 3: ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ

3.1. Εισαγωγή - Βασική Φιλοσοφία Υλοποίησης

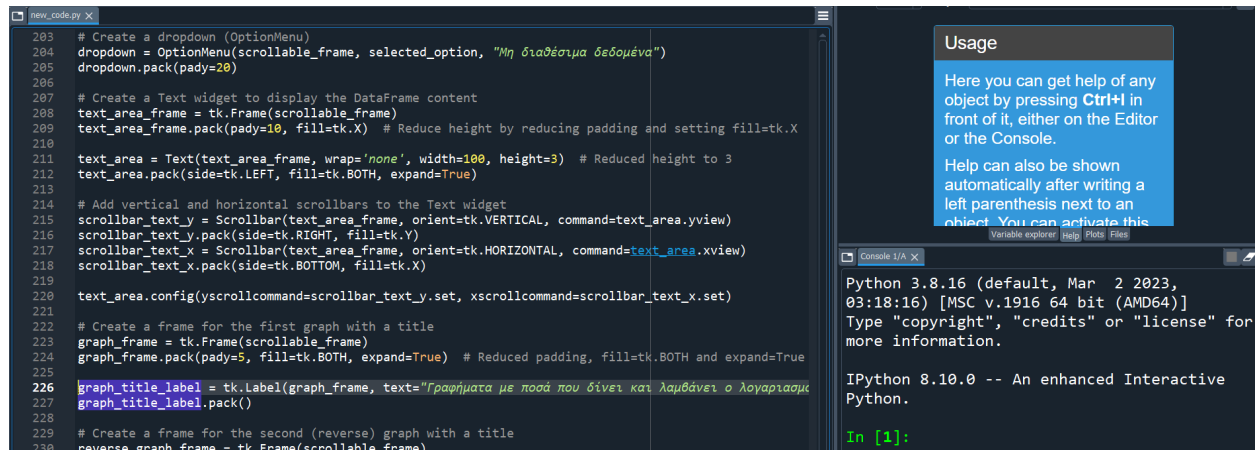
Στο πλαίσιο του πρακτικού μέρους της εργασίας, υλοποιήθηκε εφαρμογή, η οποία ως στόχο είχε τη δημιουργία μιας εφαρμογής σε γλώσσα Python, η οποία διαβάζοντας ένα αρχείο CSV το οποίο περιέχει πληροφορίες σχετικά με ποσά που διακινούνται μεταξύ λογαριασμών της Ελληνικής Επανάστασης (τις πληροφορίες του οποίου θα παρουσιάσουμε στη συνέχεια), επεξεργάζεται το περιεχόμενό του και παρουσιάζει με τρόπο φιλικό προς τον χρήστη τα αποτελέσματα της επεξεργασίας του.

Επομένως, έγινε προσπάθεια να γίνει με τον πιο εμφανή τρόπο, πρακτική εφαρμογή των όσων θεωρητικών αναφέρθηκαν σε προηγούμενα κεφάλαια, δείχνοντας με τον πιο κατηγορηματικό τρόπο την χρησιμότητα της γλώσσας Python όσον αφορά την επεξεργασία και οπτικοποίηση δεδομένων που προέρχονται από αρχεία CSV.

Πιο συγκεκριμένα με βάση το CSV αρχείο, το οποίο θα έχει συγκεκριμένη μορφή (format), δηλαδή συγκεκριμένες στήλες, θα οπτικοποιούνται πληροφορίες με μορφή κατευθυνόμενων γραφημάτων σχετικά με ποσά που δίνει ή λαμβάνει ένας λογαριασμός της Ελληνικής Επανάστασης. Με τον τρόπο αυτό, θα μπορούν εύκολα να κατανοηθούν οι συσχετίσεις μεταξύ των λογαριασμών με τρόπο εύκολο και διαδραστικό.

3.2. Περιβάλλον υλοποίησης και απαραίτητες βιβλιοθήκες

Όσο αφορά το περιβάλλον υλοποίησης της εφαρμογής ήταν το Anaconda Spyder, το οποίο αναλύθηκε σε προηγούμενη ενότητα και διευκόλυνε σε αρκετά μεγάλο βαθμό την πραγματοποίηση της πειραματικής διαδικασίας.



Εικόνα 2: Περιβάλλον Anaconda Spyder

Όσον αφορά τις βιβλιοθήκες, που είναι απαραίτητες για την εξασφάλιση της λειτουργικότητας της εφαρμογής και απαραίτητο να εγκατασταθούν με σκοπό την επιτυχή εκτέλεσή της, είναι οι:

- **Tkinter:** Η Tkinter είναι η προεπιλεγμένη βιβλιοθήκη GUI (Γραφικό Περιβάλλον Χρήστη) για την Python, η οποία παρέχει έναν εύκολο και αποδοτικό τρόπο δημιουργίας εφαρμογών. Στην περίπτωση της εφαρμογής που υλοποιήθηκε, χρησιμοποιήθηκε για τη δημιουργία του γραφικού της περιβάλλοντος.
- **Pandas:** Πρόκειται για βιβλιοθήκη, που περιέχει μια ισχυρή δομή δεδομένων, το DataFrame. Το DataFrame αποτέλεσε τη βασική δομή για τη διαχείριση και επεξεργασία των δεδομένων της εφαρμογής.
- **Networkx:** Είναι η βιβλιοθήκη η επέτρεψε την σχεδίαση των απαιτούμενων δικτύων, τα οποία δημιουργούνται στο πλαίσιο της εφαρμογής.
- **Matplotlib:** Πρόκειται για βιβλιοθήκη απαραίτητη για την δημιουργία των κατάλληλων γραφημάτων στα πλαίσια της εφαρμογής.
- **Numpy:** Βιβλιοθήκη για την διαχείριση των δεδομένων σε μορφή συστοιχιών (numpy arrays).

3.3. Αρχιτεκτονική Εφαρμογής

Στα πλαίσια της σχεδίασης της εφαρμογής, αποφασίστηκε να δημιουργηθεί ένα εύχρηστο περιβάλλον διεπαφής, μέσω του οποίου ο χρήστης θα μπορεί να ανεβάζει ένα αρχείο τύπου CSV με συγκεκριμένο format, όπως θα παρουσιαστεί στη συνέχεια.

Αυτό θα είναι το πρώτο βήμα κατά την λειτουργία της εφαρμογής. Μετά την επιτυχή φόρτωση του αρχείου, μέσω μιας αναδυόμενης λίστας (dropdown list) και χωρίς να απαιτείται χειροκίνητη πληκτρολόγηση, ο χρήστης θα μπορεί να επιλέγει τον λογαριασμό που επιθυμεί. Για τον συγκεκριμένο λογαριασμό θα δημιουργούνται σαν αποτέλεσμα της επεξεργασίας των δεδομένων δύο γραφήματα.

Στα γραφήματα που θα δημιουργούνται, τα βασικά δομικά τους στοιχεία είναι:

- **Κορυφή :** Περιέχει το όνομα του λογαριασμού.
- **Κατευθυνόμενη ακμή:** Συνδέει δύο λογαριασμούς. Η ακμή ξεκινάει από τον λογαριασμό από τον οποίο δίνεται το ποσό και καταλήγει στον λογαριασμό στον ο οποίος λαμβάνει το ποσό. Επιπλέον πάνω στην ακμή, αναγράφεται το ποσό το οποίο διακινείται μεταξύ των λογαριασμών (με διαφορετικό πάχος ακμής, ανάλογα με το ύψος του ποσού).

Αποφασίστηκε η συγκεκριμένη δομή δεδομένων για την αναπαράσταση της πληροφορίας, καθώς με ξεκάθαρο και φιλικό προς τον χρήστη τρόπο θα παρουσιάζονται οι δοσοληψίες μεταξύ των λογαριασμών της Ελληνικής Επανάστασης.

Επιπλέον, με σκοπό την καλύτερη δόμηση της πληροφορίας, αποφασίστηκε να δημιουργηθούν δύο γραφήματα:

- Ένα γράφημα που θα περιέχει πληροφορίες σχετικά με τα ποσά που δίνει ο επιλεγμένος λογαριασμός.
- Ένα γράφημα που θα περιέχει πληροφορίες σχετικά με τα ποσά που λαμβάνει ο λογαριασμός.

Τέλος, αξίζει να αναφερθεί ότι μετά την εμφάνιση των αποτελεσμάτων, ο χρήστης θα έχει την δυνατότητα να επιλέξει νέο λογαριασμό από την λίστα ώστε να προβληθούν τα ανανεωμένα αποτελέσματα.

3.4. Αρχική παρουσίαση συνόλου δεδομένων

Στα πλαίσια της συγκεκριμένης ενότητας, παρουσιάζονται χρήσιμες πληροφορίες που αφορούν ένα δοκιμαστικό αρχείο εισόδου, το οποίο χρησιμοποιήθηκε κατά την υλοποίηση της εφαρμογής. Πρόκειται για πληροφορίες, οι οποίες είχαν εξαχθεί κατά την πρώτη φάση του πειραματικού μέρους της συγκεκριμένης εργασίας, πριν καταλήξουμε στην τελική αρχιτεκτονική και λειτουργικότητα της εφαρμογής. Η πρώτη αυτή προεπεξεργασία, μας οδήγησε στα εξής συμπεράσματα:

- Ήταν επιτακτική η δημιουργία μιας εφαρμογής με γραφικό περιβάλλον.
- Η επεξεργασία των δεδομένων έπρεπε να γίνει χωρίς πολλές καθυστερήσεις.
- Η οπτικοποίηση των δεδομένων έπρεπε να γίνεται με τρόπο κατανοητό ακόμη και στον πιο αρχάριο χρήστη.

Αρχικά, παρουσιάζεται ένα απόσπασμα των δεδομένων, που περιέχει το CSV αρχείο. Στο σημείο αυτό αξίζει να αναφερθεί ότι το CSV αρχείο περιέχει δεδομένα που αφορούν την Ελληνική Επανάσταση.

etos,minas,me,protos,deyteros,poso,d_1
1823,5,1,Επαρχία & Φρούριο Ναυπλίου,Χρεωστοδανειστές, " 12,533.50 " ,να δώσει
1823,6,17,Επαρχία & Φρούριο Ναυπλίου,Χρεωστοδανειστές, " 37,605.75 " ,να δώσει
1823,6,30,Επαρχία & Φρούριο Ναυπλίου,Πλαπούτας Δημήτριος στρατηγός, " 100,000.00 " ,να δώσει
1823,8,7,Επαρχία & Φρούριο Ναυπλίου,Πλαπούτας Δημήτριος στρατηγός, " 13,845.58 " ,να δώσει
1823,9,12,Επαρχία & Φρούριο Ναυπλίου,Χρεωστοδανειστές, " 10,000.00 " ,να δώσει
1823,10,25,Επαρχία & Φρούριο Ναυπλίου,Χρεωστοδανειστές,130.00,να δώσει
1823,11,1,Επαρχία & Φρούριο Ναυπλίου,Χρεωστοδανειστές, " 5,085.00 " ,να δώσει
1823,11,10,Επαρχία & Φρούριο Ναυπλίου,Δικαίος Γρηγόριος, " 23,000.00 " ,να δώσει
1823,11,29,Επαρχία & Φρούριο Ναυπλίου,Χρεωστοδανειστές, " 2,298.00 " ,να δώσει
1823,12,10,Επαρχία & Φρούριο Ναυπλίου,Αγοραστές των εν Ναυπλίω εκποιήσεων,617.50,να δώσει
1824,1,7,Επαρχία & Φρούριο Ναυπλίου,Χρεωστοδανειστές, " 15,805.00 " ,να δώσει
1824,1,15,Επαρχία & Φρούριο Ναυπλίου,Σταματελόπουλος Νικήτας Στρατηγός, " 14,250.00 " ,να δώσει
1824,1,20,Επαρχία & Φρούριο Ναυπλίου,Πάνος Θεοδώρου Κολοκοτρώνης, " 4,700.00 " ,να δώσει
1824,1,20,Επαρχία & Φρούριο Ναυπλίου,Χρεωστοδανειστές, " 2,239.50 " ,να δώσει
1824,1,20,Επαρχία & Φρούριο Ναυπλίου,Εθνικό Ταμείο,692.50,να δώσει
1824,7,31,Επαρχία & Φρούριο Ναυπλίου,Χρεωστοδανειστές, " 5,085.00 " ,να δώσει
1824,7,31,Επαρχία & Φρούριο Ναυπλίου,Πάνος Θεοδώρου Κολοκοτρώνης, " 4,379.95 " ,να δώσει
1824,9,6,Επαρχία & Φρούριο Ναυπλίου,Χρεωστοδανειστές, " 19,500.00 " ,να δώσει
1824,9,20,Επαρχία & Φρούριο Ναυπλίου,Περρούκας Χαράλαμπος, " 12,434.25 " ,να δώσει

1824,10,10,Επαρχία & Φρούριο Ναυπλίου,Εθνικές πρόσοδοι, “ 151,200.00 “ ,να δώσει
1823,5,22,Επαρχία & Φρούριο Ναυπλίου,Χρεωστοδανειστές,130.00,να λάβει
1823,6,2,Επαρχία & Φρούριο Ναυπλίου,Ενοικιαστές Τελωνείου Ναυπλίου, “ 61,200.00 “ ,να λάβει
1823,6,7,Επαρχία & Φρούριο Ναυπλίου,Πλαπούτας Δημήτριος στρατηγός, “ 90,000.00 “ ,να λάβει
1824,1,20,Επαρχία & Φρούριο Ναυπλίου,Χρεωστοδανειστές, “ 2,627.25 “ ,να λάβει
1824,1,20,Επαρχία & Φρούριο Ναυπλίου,Αγοραστές των εν Ναυπλίο εκποιήσεων,617.50,να λάβει
1824,10,10,Επαρχία & Φρούριο Ναυπλίου,Επαρχία & Φρούριο Κορίνθου, “ 1,109.60 “ ,να λάβει

Όσον αφορά τις στήλες του συνόλου δεδομένων, παρατίθεται ο παρακάτω πίνακας:

Όνομα Στήλης
Έτος
Μήνας
ΜΕ
1 ^{ος} Λογαριασμός
2 ^{ος} Λογαριασμός
Ποσό
Είδος (να δώσει ή να λάβει)

Πίνακας 1: Στήλες συνόλου δεδομένων

Όσον αφορά, τις διαστάσεις του συνόλου δεδομένων, παρατίθεται ο παρακάτω πίνακας:

Πλήθος Γραμμών	2878
Πλήθος Στηλών	7

Πίνακας 2: Διαστάσεις συνόλου δεδομένων

Επιπλέον, παρουσιάζεται το πλήθος των ελλιπών τιμών ανά στήλη:

Όνομα Στήλης	Πλήθος Ελλιπών τιμών
Έτος	0
Μήνας	0
ΜΕ	0
1 ^{ος} Λογαριασμός	0
2 ^{ος} Λογαριασμός	0
Ποσό	0
Είδος (να δώσει ή να λάβει)	0

Πίνακας 3: Πλήθος ελλιπών τιμών ανά στήλη

Επιπλέον, εμφανίζεται πληροφορία σχετικά με το πλήθος των πρώτων λογαριασμών:

Πλήθος πρώτων λογαριασμών
181

Πίνακας 4: Πλήθος πρώτων λογαριασμών

Ακολουθεί επίσης η λίστα με τα ονόματα των πρώτων λογαριασμών, όπως πρόκυψαν από την διαδικασία επεξεργασίας των δεδομένων του αρχείου CSV:

'Επαρχία & Φρούριο Ναυπλίου',
'Χρεωστοδανειστές',
'Επαρχία & Φρούριο Μονεμβασίας',
'Ενοικιαστές της Γαστούνης',
'Επαρχία Γαστούνης',
'Μαυρομιχάλης Κωνσταντίνος (ενοικιαστής Καλαμάτας)',
'Φρουραρχία & έξ. Φρουρίου Ναυπλίου',
'Επαρχία Καλαμάτας',
'Δικαίος Γρηγόριος',
'Εξοδα',
'Εθνικές πρόσοδοι',
'Επαρχία Νησιού',
'Παπαγιαννόπουλος Πανάγος (ενοικιαστής τμ. Δερβενοχωριών)',
'Εξοδα εις την επισκευήν του Βουλευτηρίου',
'Επαρχία (τμήμα) Δερβενοχωριών',
'Γεωργακόπουλος Νικόλαος χιλιαρχος (ενοικιαστής)',
'Εξοδα του Δανείου',
'Επαρχία & Φρούριο Νεοκάστρου',
'Κάτοικοι Αγκιστρίου (ενοικιαστές Αγκιστρίου επαρχ. Πόρου Αιγίνης)',
'Φροντιστήρια Πελοποννήσου',
'Επαρχία Πόρου Αιγίνης Αγκιστρίου;',
'Φροντιστήριο Στρατοπέδου Σαλώνων',
'Επαρχία Εμλακίων',
'Πρόβλεψη Ξυλικής & άλλων ειδών',
'Επαρχία Λεονταρίου',
'Αναγνωστόπουλος Παναγιώτης (ενοικιαστής Κουτζούκ μάνης)',
'Συνδρομές προς τους διασωθέντας Ψαριανούς',
'Επαρχία Κουτζούκ Μάνης',
'Παπατσώνης Δημήτρης',
'Συνεισφορές εις Άργος για την υπόθεση Ναυπλίου',
'Επαρχία Ανδρούσης',
'Ενοικιαστές Τελωνείου Ναυπλίου',
'Ψωμιού παρτίδα',
'Πολεμοφοδίων παρτίδα',
'Αγοραστές των εν Τριπολιτσά εκποιήσεων',
'Εθνικό Ταμείο',

'Ναυτικοί μισθοί της παρούσης 'β περιόδου',
'Παραστάτες της 'β περιόδου',
'Προσωρινός Ταμίας Σπύρος Παπαλεξόπουλος',
'Ενοικιαστές Προσόδων Κορίνθου περυσινοί',
'Υπουργείον της Οικονομίας',
'Στρατόπεδα Ύδρας & Σπετσών',
'Υπουργείον του Πολέμου',
'Υπουργείον της Θρησκείας',
'Υπουργείον της Αστυνομίας',
'το Επαρχείον της Τριπολιτσάς',
'Δανειστές της Α περιόδου',
'Στρατόπεδα Στερεάς Ελλάδας',
'Υπουργείον της Θρησκείας & του Δικαίου Α περιόδου',
'Αξιωματικοί & Στρατιώται του Α πεζικού Συντάγματος & του Φιλελληνικού',
'Χατζη-Ανδρέας Αργύρης',
'Φρουράς της Διοίκησης ήτοι εκτελεστική δύναμις',
'Εθνικές δια τύπου ομολογίες',
'Οι Αρεοπαγίται',
'Κωνσταντίνος Μεταξάς γενικός Έπαρχος Αιτωλίας & Ακαρνανίας',
'Έπαρχοι των Νησιών της Α περιόδου',
'Ιωάννης Γ. Πέτας',
'Επαρχία Καρύταινας',
'Σταματελόπουλος Νικήτας Στρατηγός',
'Επαρχία Άργους',
'Στρατόπεδα της Πελοποννήσου',
'Πάνος Θεοδώρου Κολοκοτρώνης',
'Επαρχία Φαναρίου',
'Σύνταγμα Νέον Ελληνικόν υπό την οδηγίαν του Ροδίου',
'Ενοικιαστές Τριπολιτσάς',
'Αγοραστές των εν Άργος εκποιήσεων',
'Επαρχία Τριπολιτσάς',
'Ομολογίαι δια τύπου του υπ. Αρ. α Νόμου του Κώδικος',
'Επαρχία Παλαιών Πατρών',
'Εφόδια Ληφθέντα εις Τριπολιτσά',
'Επαρχία & Φρούριο Κορίνθου',
'Ενοικιαστές Κάτω Ναχαγιέ',
'Ζαφειρόπουλος Ιωάννης',
'Ιατρός Μιχαήλ',
'Επαρχία Κάτω Ναχαγιέ',
'Επαρχία Αγίου Πέτρου',
'Επαρχία Μεθώνης & Κορώνης',
'Βρεσθένης Θεοδώρητος',
'Επαρχία Λακεδαίμονος',
'Πλαπούτας Δημήτριος στρατηγός',
'Ενοικιαστές Μονεμβασίας',
'Επαρχία Μονεμβασίας',
'Καραμάννου Δημήτριος',

'Επαρχία Πραστού',
'Υπουργείον των Ναυτικών της Α περιόδου',
'Ομολογίαι χειρόγραφοι',
'Τερουσία της Πελοποννήσου',
'Παρτίς ομολογιών χειρόγραφοι',
'Μεταξάς Ανδρέας',
'Κωλέττης Ιωάννης',
'Υπουργείον της Αστυνομίας της Α περιόδου',
'Χαραλάμπης Σωτήριος',
'Υπουργείον των Εσωτερικών & του Πολέμου της Α περιόδου',
'Λουκόπουλος Αλέξιος',
'Παπαγιαννόπουλος Αναγνώστης',
'Μαυροκορδάτος Αλέξανδρος',
'Κολοκοτρώνης Θεόδωρος',
'Ούλς Βας έμπορος Άγγλος εις Ζάκυνθον',
'Ζαήμης Ανδρέας',
'Ξένος Εμμανουήλ',
'Βάρβογλης Γεώργιος',
'Ζαφειρόπουλοι Κωνσταντίνος & Παναγιώτης',
'Σπηλιωτάκης Αναγνώστης',
'Υπουργείον Αρχιγραμματέως & των Εξωτερικών της Α Περιόδου',
'Σπηλιωτόπουλοι (Αδελφοί)',
'Εφόδια Πολεμικά',
'Σπυρίδων Πίλικας ιατρός εις Ιθάκην',
'Μπότασης Παναγιώτης (ο Αντιπρόεδρος)',
'Επαρχία Ευβοίας',
'Άμιλτον Ναύαρχος Άγγλος (Χάμιλτον προφανώς;)',
'Άρειος Πάγος',
'Καρακατσάνης Γκίκας',
'Ψωμάδες Αργείοι',
'Υπουργείον των Ναυτικών',
'Υπουργείον των Εσωτερικών',
'Επαρχία Πύργου',
'Πονηρόπουλος Νικόλαος',
'Εράνου της Πελοποννήσου παρτίς',
'ο Γενικός Ταμίας του Εκτελεστικού',
'Ενοικιαστές εκ των Εθνικών Δικαιωμάτων Κορίνθου',
'Ορλάνδος Ιωάννης',
'Νέγρης Θεόδωρος',
'Επιστάτες & Δημογέρονται της επαρχίας Αρκαδίας',
'Επαρχία Αρκαδίας',
'Ενοικιαστές Εθνικών Δικαιωμάτων Καλαβρύτων',
'το Επαρχείον των Καλαβρύτων',
'Τζανέτου Νικόλαος (παραστάτης του Φαναρίου)',
'Βιλάντζον της Α Περιόδου',
'Παραστάτες Διάφοροι',
'Παπαδιαμαντόπουλος Ιωάννης',

'Ανδρούσης Ιωσήφ',
'το Επαρχείον του Αγίου Πέτρου',
'το Επαρχείον της Λακεδαιμόνος',
'το Επαρχείον του Πύργου',
'Περρούκας Χαράλαμπος',
'Αγορά Γεννημάτων',
'Δασμών παρτίς',
'Φροντιστήριον του εν Ισθμού Στρατοπέδου',
'το Επαρχείον της Κορίνθου',
'Αγοραστές των εν Ναυπλίω εκποιήσεων',
'Παλαμήδης Ρήγας (φροντιστής του εν Ισμό στρατοπέδου)',
'Εθνικών κτημάτων εκποιήσεις',
'Πρόκριτοι & κάτοικοι της Λακεδαιμόνος',
'Ενοικιαστές της Βοστίτσας',
'Επαρχία Βοστίτσας',
'Ενοικιαστές Καλαβρύτων',
'Επαρχία Καλαβρύτων',
'Αγοραστές Ενοικίων/Ιλτιζαμίων',
'Ενοίκια/Ιλτιζάμια του παρόντος χρόνου του 1824',
'Μαρίνος Μεταξάς Μαντζαράτος',
'Λόντος Ανδρέας ο Στρατηγός',
'Πρόβλεψη Γεννημάτων εις το Ναύπλιον',
'Φροντιστήριον των εν Ύδρα & Σπέτσες Στρατοπέδων',
'Νοταράς Ιωάννης ο Στρατηγός',
'Δημοπρατική Επιτροπή Κορίνθου, Βοστίτσας & Καλαβρύτων',
'Επαρχίες της Πελοποννήσου',
'Οικονόμου Νικόλαος προσωρινός ταμίας του εκτελεστικού',
'Αγοραστές εφετεινοί του τελωνίου Ναυπλίου',
'Επιτροπές δημοπρατικάς της Πελοποννήσου',
'η Τριμελής Επιτροπή της Διοικήσεως εν Τριπολιτσά',
'Καλλέργης Νικόλαος & Ξένος Εμμανουήλ',
'η του Ελληνικού Στόλου παρτίς',
'Εθνικό Ταμείο νέα παρτίδα',
'Τύρος Συναλλαγματικών',
'Η δια της Φλώριδος ελθούσα Μερὶς του Δανείου',
'Η εν Λονδίνω Επιτροπή του Δανείου συγκείμενη από τους Ιω. Ορλάνδο, Ανδρ. Λουριώτη & Ιω. Ζαήμη',
'Ιονική Διοίκησις',
'Κουντουριώτης Γεώργιος ο πρόεδρος',
'Υπουργείον του Δικαίου',
'Επιτροπή του ενοικιάσματος των οσπιτίων & εργαστηρίων Ναυπλίου',
'Ενοικιαστές των επαρχιών Ναυπλίου & Ἀργους',
'Πρόκριτοι της επαρχίας Τριπολιτσάς',
'Επιτροπή των εκποιήσεων',
'Μοναστήρια Διάφορα για τα Ασημικά',
'Επαρχίες διάφορες για τα Ασημικά',
'Επαρχίες της Στερεάς Ελλάδος',

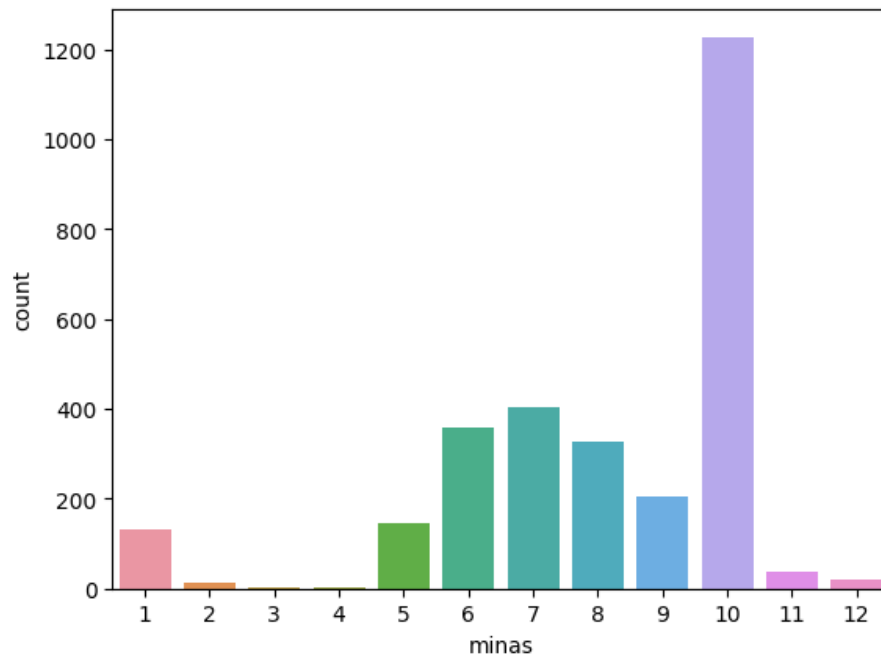
'Χρεώστες της Α & Β περιόδου', 'Εξοδα της Επικρατείας', 'Πρόσοδοι Εθνικάί της Γ περιόδου', 'Δανειστές της Α & Β περιόδου', 'Βιλάντζον της Β Περιόδου'

Πίνακας 5: Λίστα πρώτων λογαριασμών

Από την παραπάνω λίστα έγινε σαφές ότι δεν γινόταν να υλοποιηθεί μια εφαρμογή η οποία να ζητάει χειροκίνητα από τον χρήστη να δίνει το όνομα του λογαριασμού, για δύο κυρίως λόγους:

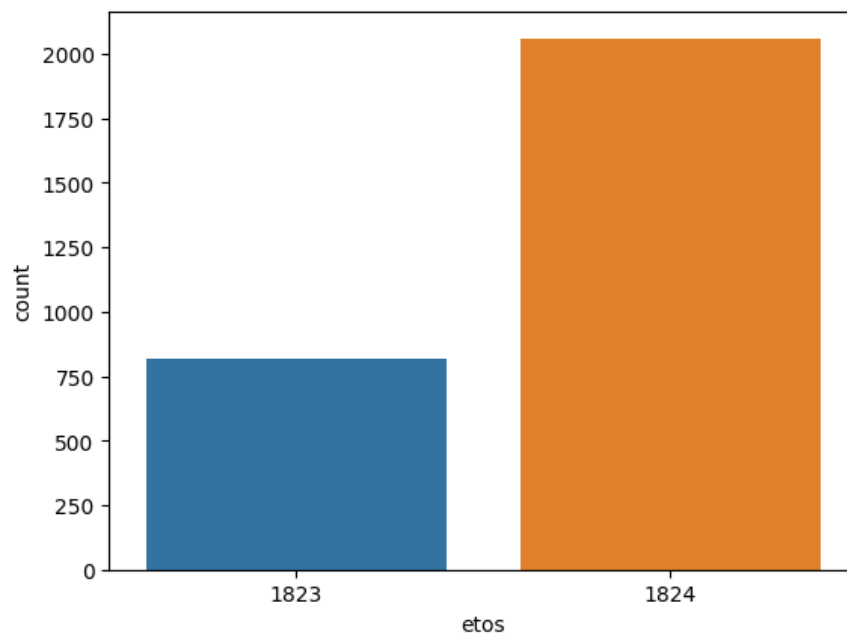
- Μεγάλο πλήθος λογαριασμών.
- Ονόματα λογαριασμών με δυσκολία στη γραφή.

Παρουσιάζοντας επιπλέον στοιχεία, που έχουν εξαχθεί από την επεξεργασία του δοκιμαστικού αρχείου εισόδου, παρατίθεται το παρακάτω γράφημα, το οποίο παρουσιάζει την κατανομή του πλήθους των εγγραφών ανά μήνα:



Εικόνα 3: Κατανομή πλήθους εγγραφών ανά μήνα

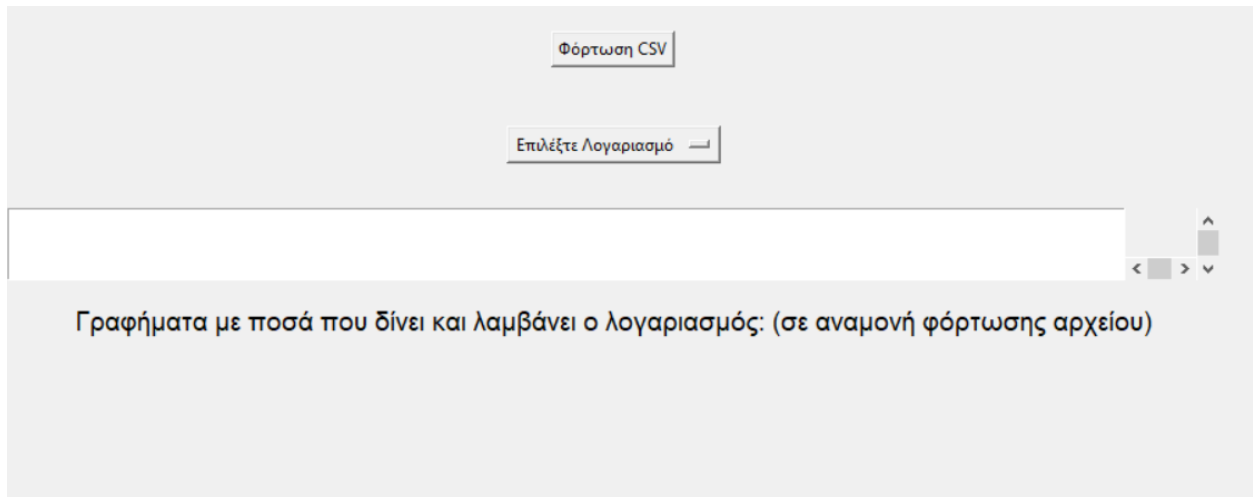
Επιπροσθέτως, παρατίθεται το ιστόγραμμα που παρουσιάζει την κατανομή του πλήθους των εγγραφών ανά έτος:



Εικόνα 4: Κατανομή πλήθους εγγραφών ανά έτος

3.5. Παρουσίαση λειτουργικότητας της εφαρμογής

Αρχικά, παρουσιάζεται η αρχική οθόνη της εφαρμογής:



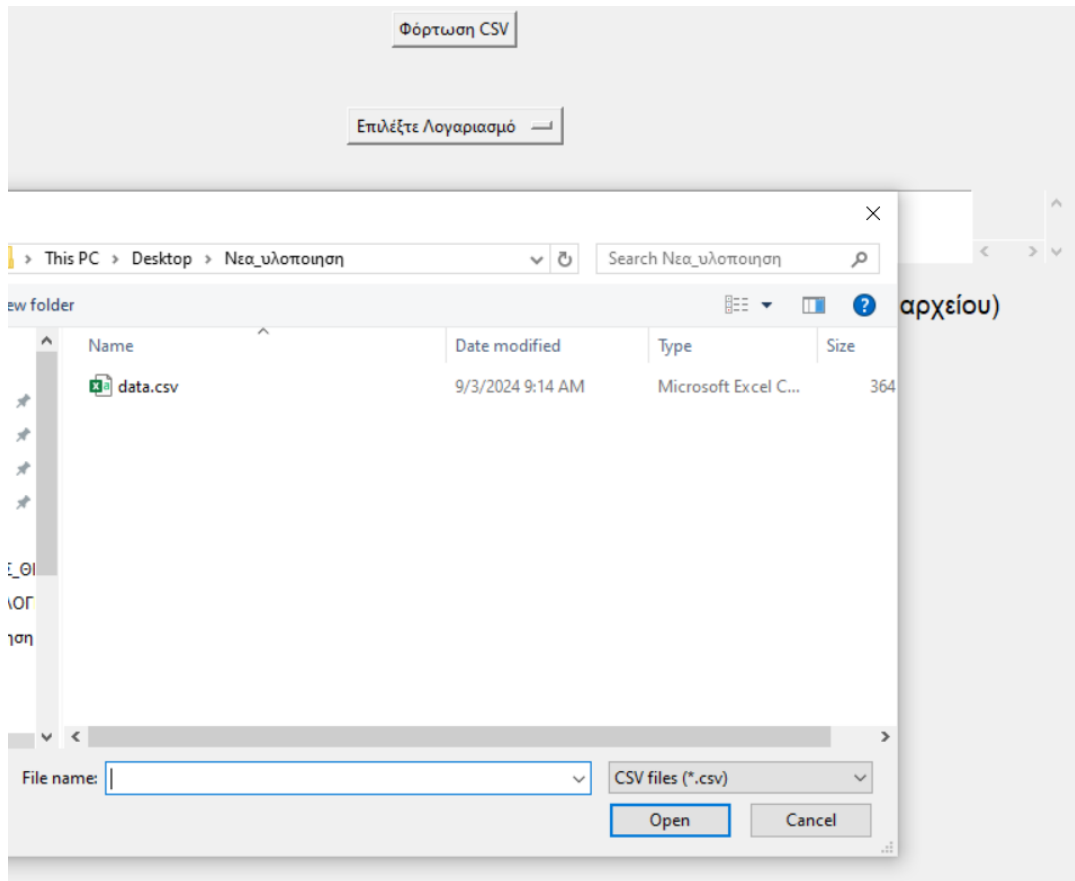
Εικόνα 5: Αρχική Οθόνη Εφαρμογής

Στην αρχική αυτή οθόνη περιέχεται:

- Κουμπί για την φόρτωση του CSV αρχείου..
- Δυνατότητα επιλογής λογαριασμού.

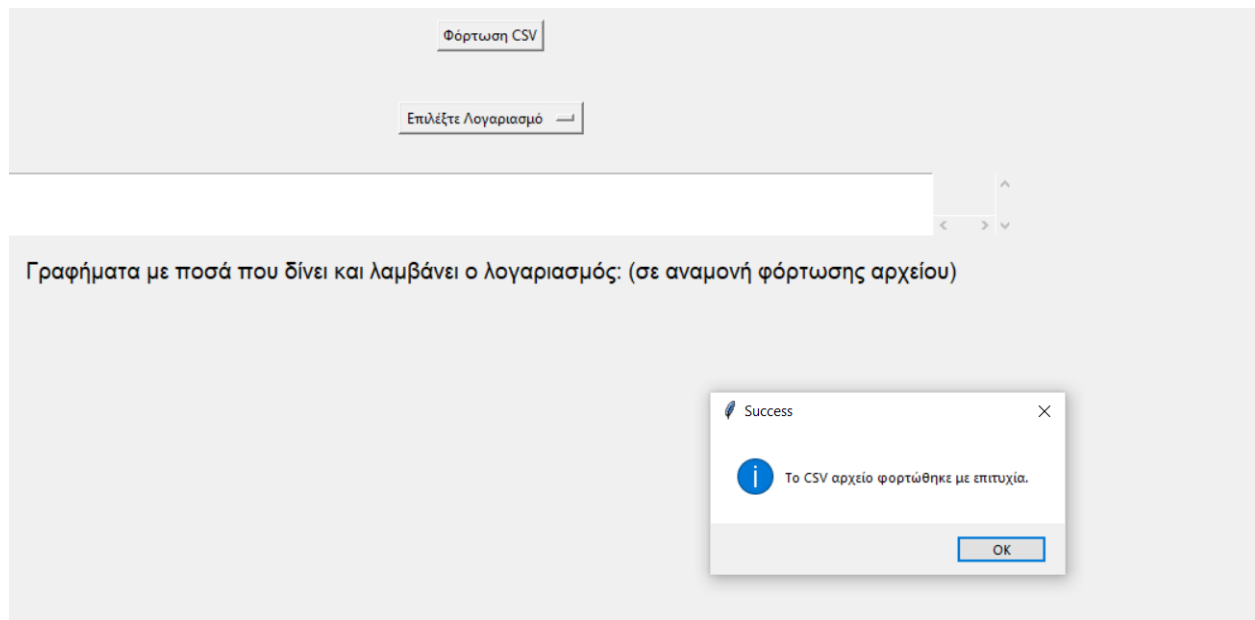
Επιπλέον, όπως είναι προφανές, χωρίς να έχει φορτωθεί κάποιο CSV αρχείο, δεν εμφανίζονται αποτελέσματα.

Στη συνέχεια, παρατίθεται screenshot, όπου φαίνεται η δυνατότητα επιλογής του CSV αρχείου:



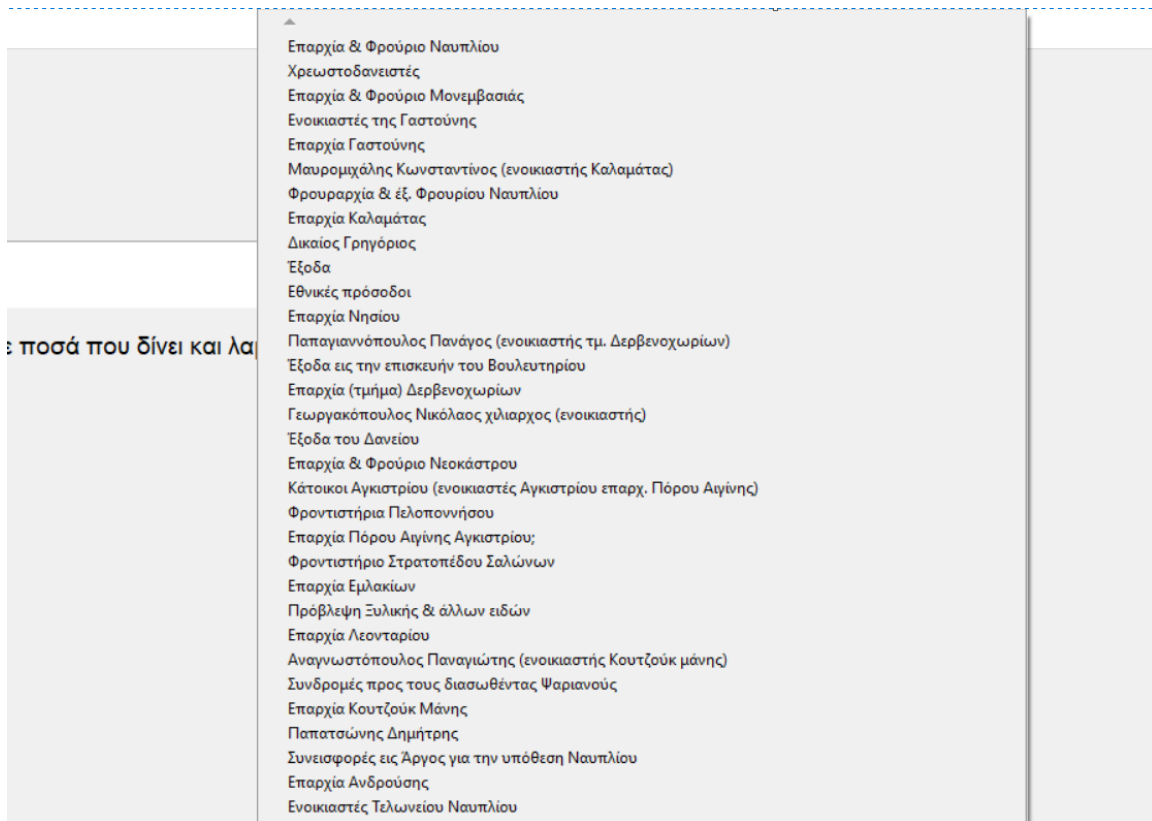
Εικόνα 6: Επιλογή CSV αρχείου

Μετά την επιτυχή φόρτωση του CSV αρχείου, εμφανίζεται μήνυμα όπως αυτό που φαίνεται στη συνέχεια:



Εικόνα 7: Επιτυχής φόρτωση CSV αρχείου

Μετά την επιτυχή φόρτωση του CSV, υπάρχει δυνατότητα επιλογής λογαριασμού όπως φαίνεται και στη συνέχεια:



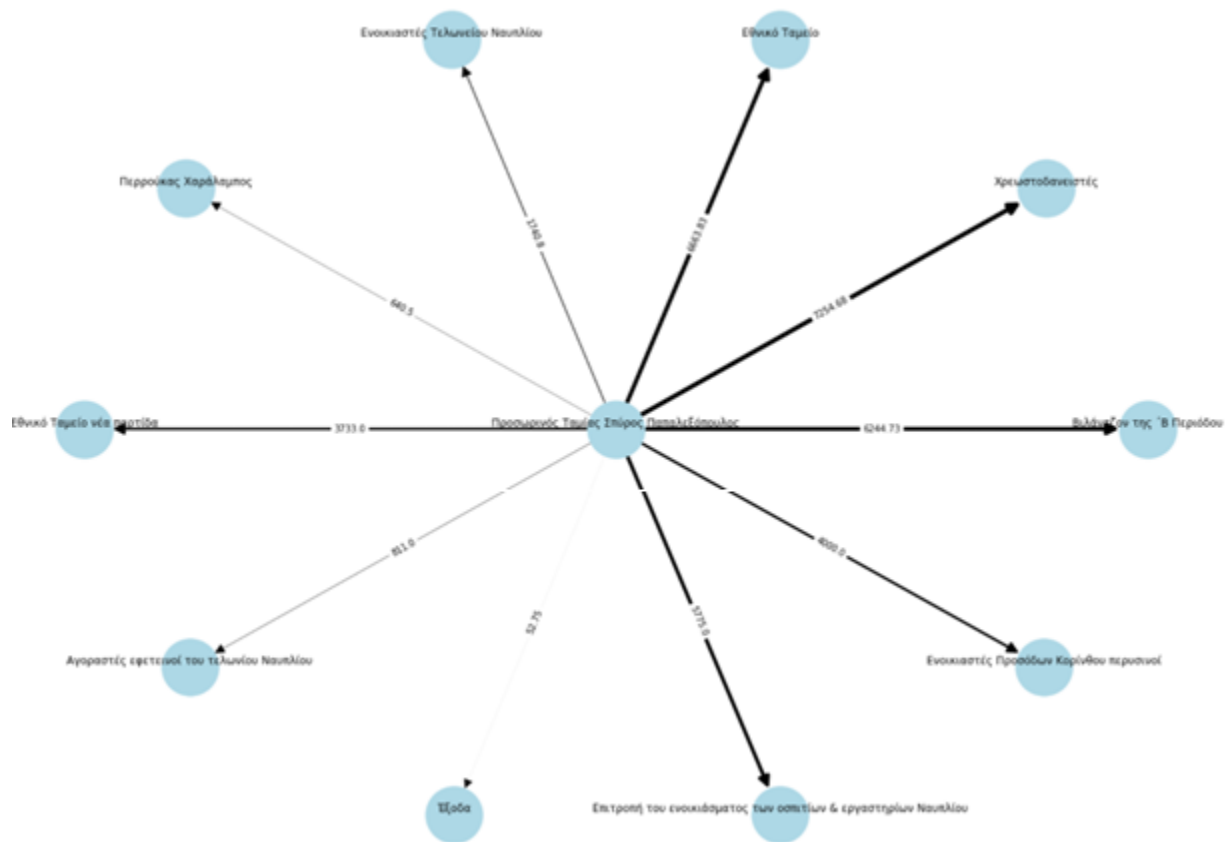
Εικόνα 8: Επιλογή λογαριασμού προς οπτικοποίηση

Μετά την επιλογή του λογαριασμού, αρχικά εμφανίζεται ο παρακάτω πίνακας:

Φόρτωση CSV									
Επιλέξτε Λογαριασμό									
Έτος	Μήνας	ΜΕ	Πρώτος				Δεύτερος	Ποσό	Είδος
1823	7	1	Προσωρινός Ταμίας	Σπύρος	Παπαλεξόπουλος		Χρεωτοδανειστής	31.50	να δώσει
1823	7	31	Προσωρινός Ταμίας	Σπύρος	Παπαλεξόπουλος		Χρεωτοδανειστής	4,521.00	να δώσει

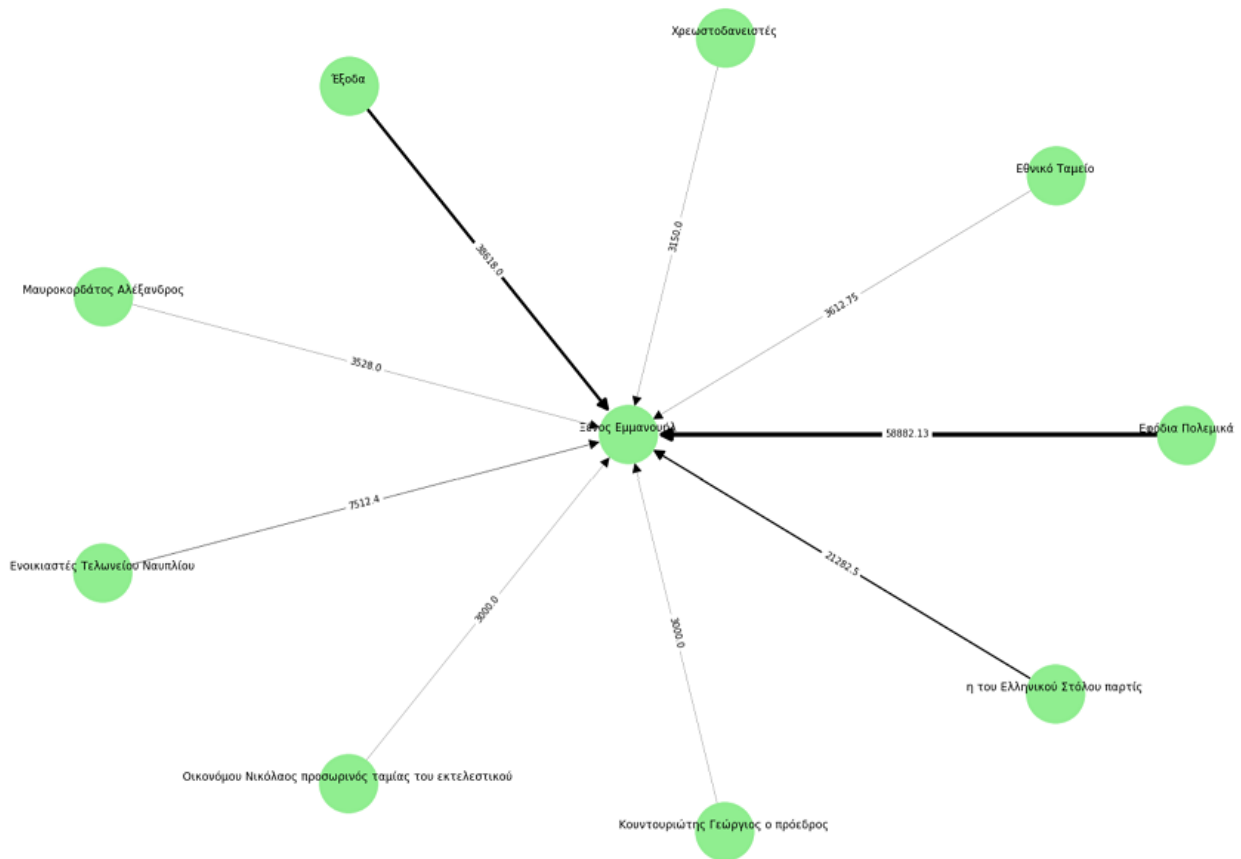
Εικόνα 9: Πίνακας δοσοληψιών λογαριασμού

Ο πίνακας αυτός περιέχει πληροφορίες σχετικά με τις δοσοληψίες του λογαριασμού που επελέγη. Ακολουθεί το γράφημα που προέκυψε σχετικά με τα ποσά που δίνει ένας λογαριασμός:



Εικόνα 10: Γράφημα με ποσά που δίνει ο λογαριασμός

Ακολουθεί επίσης, το γράφημα το οποίο δημιουργείται σχετικά με τα ποσά που λαμβάνει ένας λογαριασμός:



Εικόνα 11: Γράφημα με ποσά που λαμβάνει ο λογαριασμός

3.6. Σχολιασμός της εφαρμογής

Ακολουθούν συνοπτικά κάποιες χρήσιμες παρατηρήσεις σχετικά με την εφαρμογή, η οποία υλοποιήθηκε:

- Η φόρτωση και επεξεργασία των δεδομένων γίνεται αρκετά γρήγορα χωρίς να απαιτείται κάποιο χρονικό διάστημα το οποίο να είναι έντονα αντιληπτό στον χρήστη.
- Τα δύο γραφήματα δημιουργούνται με τρόπο φιλικό (έγινε προσπάθεια για όσο το δυνατόν πληρέστερη και καθαρή αναπαράσταση της πληροφορίας, πλην ελαχίστων περιπτώσεων, όπου υπάρχει μεγάλος όγκος δοσοληψιών).
- Η χρήση διαφορετικών χρωμάτων όσον αφορά τα δύο γραφήματα συμβάλλει στην φιλικότερη χρήση της εφαρμογής.
- Δυνατό σημείο της εφαρμογής αποτελεί η χρήση διαφορετικού πάχους ακμής ανάλογα με το ποσό που αντιπροσωπεύει. Με τον τρόπο αυτό, μπορούν εύκολα να οπτικοποιηθούν ζεύγη λογαριασμών που παρουσιάζουν υψηλό βαθμό συσχέτισης μεταξύ τους.
- Σημαντικό επίσης χαρακτηριστικό της εφαρμογής, αποτελεί η χρήση δύο διαφορετικών τύπων οπτικοποίησης (πίνακες και γραφήματα).

Συμπεράσματα – Μελλοντική Εργασία

Τα συμπεράσματα, τα οποία προκύπτουν από την υλοποίηση του πρακτικού μέρους είναι τα παρακάτω:

- Η πειραματική προσπάθεια πέτυχε τον σκοπό της, καθώς πραγματοποιήθηκε με επιτυχία αποτελεσματική επεξεργασία και οπτικοποίηση του CSV αρχείου.
- Η γλώσσα Python διαδραμάτισε, όπως αναμενόταν, κρίσιμο ρόλο στην επιτυχία του πειραματικού μέρους μέσω των βιβλιοθηκών που διαθέτει.
- Η δομή DataFrame μπορεί να λειτουργήσει αποδοτικά για την αποθήκευση και διαχείριση μεγάλου όγκου δεδομένων.
- Η χρήση της βιβλιοθήκης network συνέβαλλε σημαντικά στον κύριο στόχο της οπτικοποίησης, που ήταν η απεικόνιση της συσχέτισης μεταξύ των λογαριασμών.
- Κρίσιμη ήταν επίσης και η συμβολή της βιβλιοθήκης Tkinter όσον αφορά την δημιουργία του γραφικού περιβάλλοντος διεπαφής της εφαρμογής.
- Η οπτικοποίηση των αποτελεσμάτων επεξεργασίας των δεδομένων όσον αφορά τα ποσά που δίνει και λαμβάνει ένας λογαριασμός, κατέστη εφικτή στις περισσότερες των περιπτώσεων. Σε περιπτώσεις που υπήρχε τεράστιος όγκος συσχετίσεων και επομένως μεγάλα γραφήματα, η οπτικοποίηση παρουσίαζε κάποια προβλήματα (κορυφές που συμπίπτουν, με αποτέλεσμα να μην είναι εμφανής η πληροφορία).

Σαν μελλοντικές επεκτάσεις προτείνονται:

- Δοκιμαστική χρήση της εφαρμογής με διαφορετικά αρχεία εισόδου της ίδιας μορφής.
- Δημιουργία μιας πιο δυναμικής εφαρμογής, που θα διαβάσει CSV αρχεία με διαφορετική δομή και θα μπορεί να οπτικοποιεί τις συσχετίσεις μεταξύ των επιλεγμένων από τον χρήστη στηλών.
- Δημιουργία web-based έκδοσης της εφαρμογής.
- Προσπάθεια για αντιμετώπιση των προβλημάτων και αποδοτικότερη αναπαράσταση μεγάλων σε όγκο γραφημάτων.
- Δυνατότητα διάδρασης του χρήστη με το γράφημα: Για παράδειγμα, δυνατότητα επιλογής λογαριασμού επιλέγοντας την αντίστοιχη κορυφή στο γράφημα και όχι μόνο μέσω αναδυόμενης λίστας.
- Περισσότερες λειτουργίες στα προκύπτοντα γραφήματα: πχ. αποθήκευση, εστίαση σε συγκεκριμένα σημεία του γραφήματος.
- Προσθήκη δυνατότητας ιστορικού αναζητήσεων.
- Πρόταση προβολής στοιχείων λογαριασμών με βάση λογαριασμούς που έχουν ήδη προβληθεί και παρουσιάζουν υψηλό βαθμό συσχέτισης.

ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ

```
import tkinter as tk

from tkinter import filedialog, StringVar, OptionMenu, Text, Scrollbar, messagebox

import pandas as pd

import networkx as nx

import matplotlib.pyplot as plt

from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg

import numpy as np


# Global variable to hold the DataFrame
df = pd.DataFrame()


# Function to open and load the CSV file into a DataFrame
def load_csv():

    global df

    file_path = filedialog.askopenfilename(filetypes=[("CSV files", "*.csv")])

    if not file_path:

        return

    try:

        df = pd.read_csv(file_path)

        if 'Πρώτος' in df.columns:

            distinct_values = df['Πρώτος'].unique().tolist()

            update_dropdown(distinct_values)
```

```
        messagebox.showinfo("Success", "Το CSV αρχείο φορτώθηκε με επιτυχία.")
    else:
        messagebox.showerror("Error", "Column 'Πρώτος' not found in the CSV file.")
        update_dropdown([])
except Exception as e:
    messagebox.showerror("Error", f"Failed to load CSV file: {e}")
    update_dropdown([])

def update_dropdown(values):
    menu = dropdown['menu']
    menu.delete(0, 'end') # Clear existing options
    if values:
        for value in values:
            menu.add_command(label=value, command=lambda v=value:
on_option_selected(v))
    else:
        menu.add_command(label="Μη διαθέσιμα δεδομένα", command=lambda:
None)

def on_option_selected(value):
    global df
    if not df.empty:
        try:
            filtered_df = df[df['Πρώτος'] == value]
            display_dataframe(filtered_df)
```

```
seconds = filtered_df['Δεύτερος'].unique().tolist()

dinei = [0] * len(seconds)
lamvanei = [0] * len(seconds)

for i in range(len(filtered_df)):
    for j in range(len(seconds)):
        if filtered_df.iloc[i, 4] == seconds[j]:

            amount = float(filtered_df.iloc[i, 5].replace(',', ''))
            if filtered_df.iloc[i, 6] == 'να δώσει':
                dinei[j] += amount

            elif filtered_df.iloc[i, 6] == 'να λάβει':
                lamvanei[j] += amount

create_graphs(value, seconds, dinei, lamvanei)

except KeyError as e:
    messagebox.showerror("Error", f"Missing column: {e}")
except ValueError as e:
    messagebox.showerror("Error", f"Value error: {e}")

def display_dataframe(dataframe):
    text_area.delete('1.0', tk.END)

    if not dataframe.empty:
```

```

text_area.insert(tk.END, dataframe.to_string(index=False))

def clear_graph_frames():
    for widget in graph_frame.winfo_children():
        widget.destroy()
    for widget in reverse_graph_frame.winfo_children():
        widget.destroy()

def circular_layout_with_center(graph, center_node):
    pos = nx.circular_layout(graph)
    pos[center_node] = np.array([0, 0]) # Place the central node at the origin
    angle_step = 2 * np.pi / (len(pos) - 1)
    radius = 1

    for i, node in enumerate(graph.nodes()):
        if node != center_node:
            pos[node] = np.array([radius * np.cos(i * angle_step), radius * np.sin(i *
angle_step)])

    return pos

def create_graphs(value, seconds, dinei, lamvanei):
    clear_graph_frames()

    # Create the first graph with larger size
    G = nx.DiGraph()

```

```

G.add_node(value)

for second, weight in zip(seconds, dinei):
    if weight > 0:
        G.add_edge(value, second, weight=weight)

pos = circular_layout_with_center(G, value)

fig1, ax1 = plt.subplots(figsize=(20, 15)) # Increased figure size for better label
visibility

# Scale edge widths based on weights
edges = G.edges(data=True)
edge_weights = [data['weight'] for _, _ data in edges]
edge_widths = [weight / max(edge_weights) * 5 for weight in edge_weights] # Scale
widths

nx.draw(G, pos, with_labels=False, ax=ax1, node_color='lightblue',
font_weight='bold', node_size=4000, arrowsize=25, width=edge_widths)

# Draw labels below nodes with reduced font size
nx.draw_networkx_labels(G, pos, labels={node: node for node in G.nodes()},
ax=ax1, verticalalignment='bottom', font_size=12)

edge_labels = nx.get_edge_attributes(G, 'weight')
nx.draw_networkx_edge_labels(G, pos, edge_labels=edge_labels, ax=ax1,
font_size=10)

```

```

ax1.set_title(f"Ποσά που δίνει ο λογαριασμός: {value}", fontsize=16)

plt.tight_layout() # Adjust the layout to ensure labels fit within the figure

# Embed the first graph in the Tkinter canvas
canvas1 = FigureCanvasTkAgg(fig1, master=graph_frame)
canvas1.draw()
canvas1.get_tk_widget().pack(fill=tk.BOTH, expand=True)

# Create the second graph with larger size
G_reverse = nx.DiGraph()
G_reverse.add_node(value)
for second, weight in zip(seconds, lamvanei):
    if weight > 0:
        G_reverse.add_edge(second, value, weight=weight)

pos_reverse = circular_layout_with_center(G_reverse, value)

fig2, ax2 = plt.subplots(figsize=(20, 15)) # Increased figure size for better label
visibility

# Scale edge widths based on weights
reverse_edges = G_reverse.edges(data=True)
reverse_edge_weights = [data['weight'] for _, _, data in reverse_edges]
reverse_edge_widths = [weight / max(reverse_edge_weights) * 5 for weight in
reverse_edge_weights] # Scale widths

```

```

nx.draw(G_reverse, pos_reverse, with_labels=False, ax=ax2,
node_color='lightgreen', font_weight='bold', node_size=4000, arrowsize=25,
width=reverse_edge_widths)

# Draw labels below nodes with reduced font size
nx.draw_networkx_labels(G_reverse, pos_reverse, labels={node: node for node in
G_reverse.nodes()}, ax=ax2, verticalalignment='bottom', font_size=12)

edge_labels_reverse = nx.get_edge_attributes(G_reverse, 'weight')
nx.draw_networkx_edge_labels(G_reverse, pos_reverse,
edge_labels=edge_labels_reverse, ax=ax2, font_size=10)
ax2.set_title(f"Ποσά που λαμβάνει ο λογαριασμός: {value}", fontsize=16)

plt.tight_layout() # Adjust the layout to ensure labels fit within the figure

# Embed the second graph in the Tkinter canvas
canvas2 = FigureCanvasTkAgg(fig2, master=reverse_graph_frame)
canvas2.draw()
canvas2.get_tk_widget().pack(fill=tk.BOTH, expand=True)

# Ensure that the canvas size is updated to fit the graphs
graph_frame.update_idletasks()
reverse_graph_frame.update_idletasks()

canvas.config(scrollregion=canvas.bbox("all"))

```

```
# Initialize Tkinter window with a larger size
root = tk.Tk()
root.title("CSV Loader")
root.geometry("1600x900") # Increased window size to 1600x900 pixels

# Create a Scrollable Canvas
main_frame = tk.Frame(root)
main_frame.pack(fill=tk.BOTH, expand=True)

canvas = tk.Canvas(main_frame)
canvas.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

scrollbar = Scrollbar(main_frame, orient=tk.VERTICAL, command=canvas.yview)
scrollbar.pack(side=tk.RIGHT, fill=tk.Y)
canvas.config(yscrollcommand=scrollbar.set)

# Create a frame inside the canvas
scrollable_frame = tk.Frame(canvas)
scrollable_frame.bind(
    "<Configure>",
    lambda e: canvas.config(
        scrollregion=canvas.bbox(tk.ALL)
    )
)
```

```
canvas.create_window((0, 0), window=scrollable_frame, anchor="nw")

# Update canvas size when the window is resized
canvas.bind('<Configure>', lambda e: canvas.config(scrollregion=canvas.bbox("all")))

# Create a button to load the CSV
load_button = tk.Button(scrollable_frame, text="Φόρτωση CSV", command=load_csv)
load_button.pack(pady=20)

# Create a StringVar for the selected option in the dropdown
selected_option = StringVar(root)
selected_option.set("Επιλέξτε Λογαριασμό")

# Create a dropdown (OptionMenu)
dropdown = OptionMenu(scrollable_frame, selected_option, "Μη διαθέσιμα  
δεδομένα")
dropdown.pack(pady=20)

# Create a Text widget to display the DataFrame content
text_area_frame = tk.Frame(scrollable_frame)
text_area_frame.pack(pady=10, fill=tk.X) # Reduce height by reducing padding and  
setting fill=tk.X

text_area = Text(text_area_frame, wrap='none', width=100, height=3) # Reduced  
height to 3
```

```
text_area.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

# Add vertical and horizontal scrollbars to the Text widget
scrollbar_text_y = Scrollbar(text_area_frame, orient=tk.VERTICAL,
command=text_area.yview)
scrollbar_text_y.pack(side=tk.RIGHT, fill=tk.Y)
scrollbar_text_x = Scrollbar(text_area_frame, orient=tk.HORIZONTAL,
command=text_area.xview)
scrollbar_text_x.pack(side=tk.BOTTOM, fill=tk.X)

text_area.config(yscrollcommand=scrollbar_text_y.set,
xscrollcommand=scrollbar_text_x.set)

# Create a frame for the first graph with a title
graph_frame = tk.Frame(scrollable_frame)
graph_frame.pack(pady=5, fill=tk.BOTH, expand=True) # Reduced padding,
fill=tk.BOTH and expand=True for larger area

graph_title_label = tk.Label(graph_frame, text="Γραφήματα με ποσά που δίνει και
λαμβάνει ο λογαριασμός: (σε αναμονή φόρτωσης αρχείου)", font=("Arial", 14))
graph_title_label.pack()

# Create a frame for the second (reverse) graph with a title
reverse_graph_frame = tk.Frame(scrollable_frame)
reverse_graph_frame.pack(pady=5, fill=tk.BOTH, expand=True) # Reduced padding,
fill=tk.BOTH and expand=True for larger area
```

```
# Start the Tkinter event loop
```

```
root.mainloop()
```

ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ

- [1].Mitlöhner, J., Neumaier, S., Umbrich, J., & Polleres, A. (2016, August). Characteristics of open data CSV files. In 2016 2nd International Conference on Open and Big Data (OBD) (pp. 72-79). IEEE.
- [2].Shafranovich, Y. (2005). Common format and MIME type for comma-separated values (CSV) files (No. rfc4180).
- [3].Srinath, K. R. (2017). Python–the fastest growing programming language. International Research Journal of Engineering and Technology, 4(12), 354-357.
- [4].Hoving, R., Slot, G., & Jansen, S. (2013, July). Python: Characteristics identification of a free open source software ecosystem. In 2013 7th IEEE international conference on digital ecosystems and technologies (DEST) (pp. 13-18). IEEE.
- [5].Sharma, A., Khan, F., Sharma, D., Gupta, S., & Student, F. Y. (2020). Python: the programming language of future. Int. J. Innovative Res. Technol, 6(2), 115-118.
- [6].van Rossum, G., & Drake, F. L. (Eds.). (2006). An introduction to Python. Network Theory Limited.
- [7].Summerfield, M. (2009). Programming in Python 3: a complete introduction to the Python language. Addison-Wesley Professional.
- [8].Gowrishankar, S., & Veena, A. (2018). Introduction to Python programming. Chapman and Hall/CRC.
- [9].Taori, P., & Dasararaju, H. K. (2019). Introduction to Python. Essentials of Business Analytics: An Introduction to the Methodology and its Applications, 917-944.
- [10]. Saabith, A. S., Fareez, M. M. M., & Vinothraj, T. (2019). Python current trend applications-an overview. International Journal of Advance Engineering and Research Development, 6(10).

- [11]. Forcier, J., Bissex, P., & Chun, W. J. (2008). Python web development with Django. Addison-Wesley Professional.
- [12]. Patkar, U., Singh, P., Panse, H., Bhavsar, S., & Pandey, C. (2022). Python for web development. International Journal of Computer Science and Mobile Computing, 11(4), 36.
- [13]. Taneja, S., & Gupta, P. R. (2014). Python as a tool for web server application development. JIMS8I-International Journal of Information Communication and Computing Technology, 2(1), 77-83.
- [14]. Grinberg, M. (2018). Flask web development. “ O'Reilly Media, Inc. ” .
- [15]. Chan, J., Chung, R., & Huang, J. (2019). Python API Development Fundamentals: Develop a full-stack web application with Python and Flask. Packt Publishing Ltd.
- [16]. McGugan, W. (2007). Beginning game development with Python and Pygame: from novice to professional. Apress.
- [17]. Parker, J. R. (2021). Game Development Using Python. Mercury Learning and Information.
- [18]. Chaudhary, P., & Agrawal, Y. Python For Game Development: A Comprehensive.
- [19]. Simian, D., & Vulpeanu, A. (2022). Using Python in developing video games. Bulletin of the Transilvania University of Brasov. Series III: Mathematics and Computer Science, 203-214.
- [20]. Raschka, S., Patterson, J., & Nolet, C. (2020). Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. Information, 11(4), 193.
- [21]. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.
- [22]. Raschka, S., & Mirjalili, V. (2019). Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2. Packt publishing ltd.

- [23]. Sarkar, D., Bali, R., & Sharma, T. (2018). Practical machine learning with Python. Book “ Practical Machine Learning with Python, 25-30.
- [24]. Müller, A. C., & Guido, S. (2016). Introduction to machine learning with Python: a guide for data scientists. “ O'Reilly Media, Inc. “ .
- [25]. Raschka, S. (2017). Pdf Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow, by.
- [26]. Swamynathan, M. (2017). Mastering machine learning with python in six steps: A practical implementation guide to predictive data analytics using python. Manohar Swamynathan.
- [27]. Cao, S., Zeng, Y., Yang, S., & Cao, S. (2021). Research on Python data visualization technology. In Journal of physics: Conference series (Vol. 1757, No. 1, p. 012122). IOP Publishing.
- [28]. Sial, A. H., Rashdi, S. Y. S., & Khan, A. H. (2021). Comparative analysis of data visualization libraries Matplotlib and Seaborn in Python. International Journal, 10(1), 45.
- [29]. Embarak, D. O., Embarak, & Karkal. (2018). Data analysis and visualization using python. Berkeley, CA, USA: Apress.
- [30]. Yim, A., Chung, C., & Yu, A. (2018). Matplotlib for Python Developers: Effective techniques for data visualization with Python. Packt Publishing Ltd.
- [31]. Podrzaj, P. (2019, August). A brief demonstration of some Python GUI libraries. In Proceedings of the 8th International Conference on Informatics and Applications ICIA2019 (pp. 1-6).
- [32]. Moore, A. D. (2021). Python GUI Programming with Tkinter: Design and build functional and user-friendly GUI applications. Packt Publishing Ltd.
- [33]. Meier, B. (2019). Python GUI Programming Cookbook: Develop functional and responsive user interfaces with tkinter and PyQt5. Packt Publishing Ltd.

- [34]. Moore, A. D., & Harwani, B. M. (2019). Python GUI Programming-A Complete Reference Guide: Develop responsive and powerful GUI applications with PyQt and Tkinter. Packt Publishing Ltd.
- [35]. Shmueli, G., Bruce, P. C., Gedeck, P., & Patel, N. R. (2019). Data mining for business analytics: concepts, techniques and applications in Python. John Wiley & Sons.
- [36]. Kumar, A., & Babcock, J. (2017). Python: Advanced Predictive Analytics Gain practical insights by exploiting data in your business to build advanced predictive modeling applications. Packt Publishing.
- [37]. Kadiyala, A., & Kumar, A. (2017). Applications of Python to evaluate environmental data science problems. Environmental Progress & Sustainable Energy, 36(6), 1580-1586.
- [38]. Rolon-Mérette, D., Ross, M., Rolon-Mérette, T., & Church, K. (2016). Introduction to Anaconda and Python: Installation and setup. Quant. Methods Psychol, 16(5), S3-S11.
- [39]. Vaughan, L. (2023). Python Tools for Scientists: An Introduction to Using Anaconda, JupyterLab, and Python's Scientific Libraries. No Starch Press.