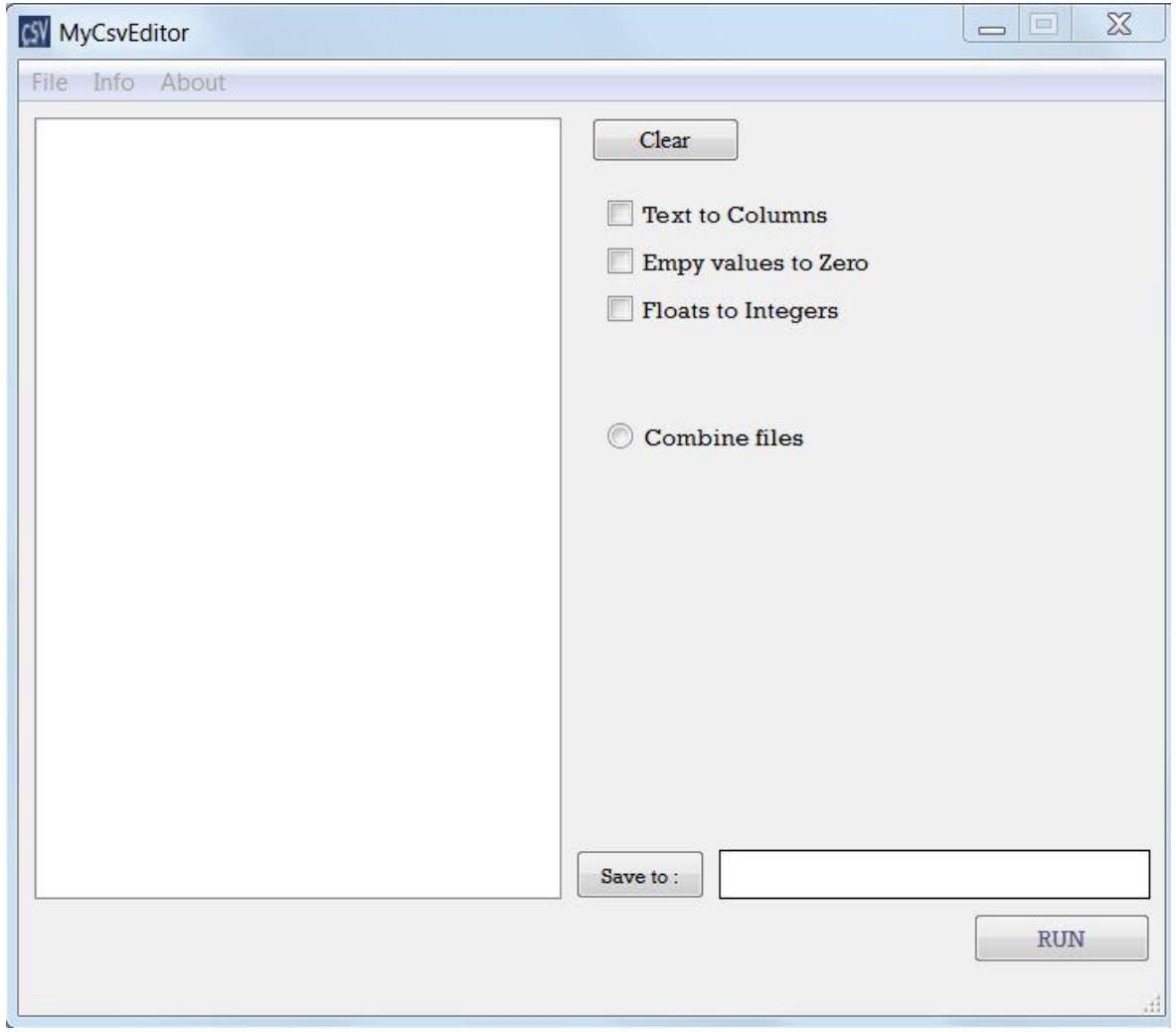
Ανάπτυξη εφαρμογής με σκοπό την διαχείριση και επεξεργασία πολλαπλών αρχείων csv, με την χρήση της γλώσσας προγραμματισμού Python

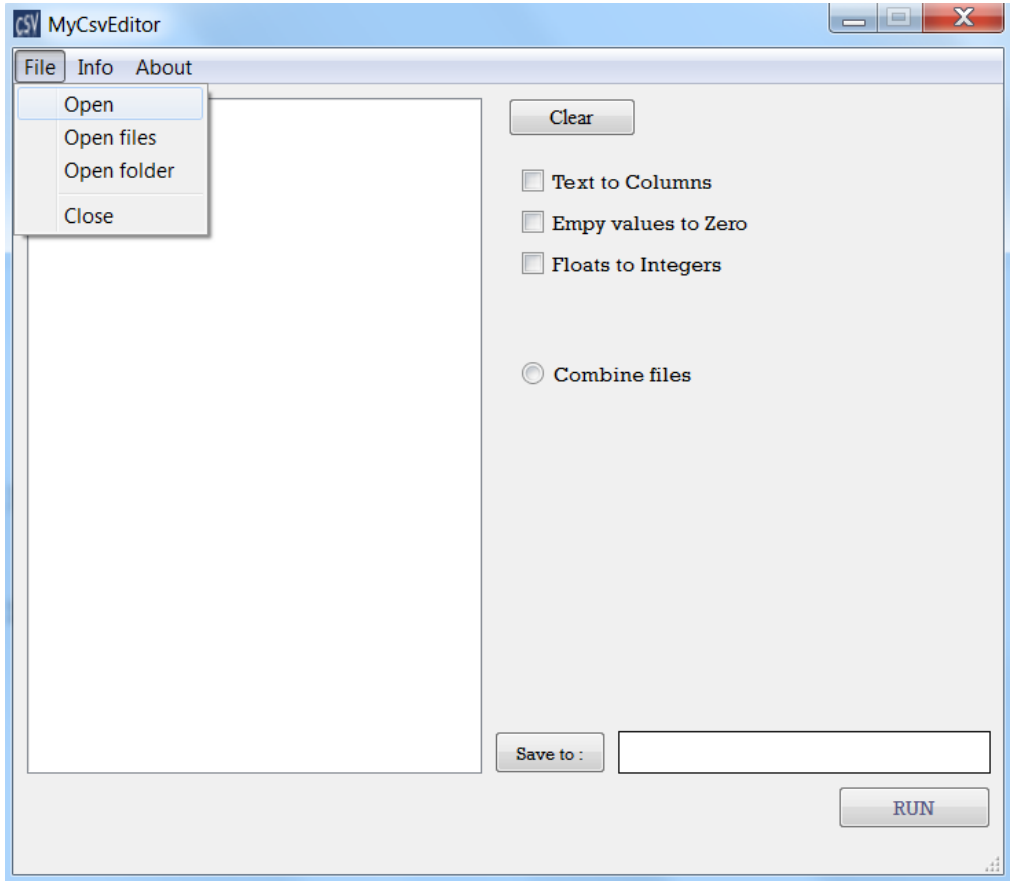
MyCsvEditor



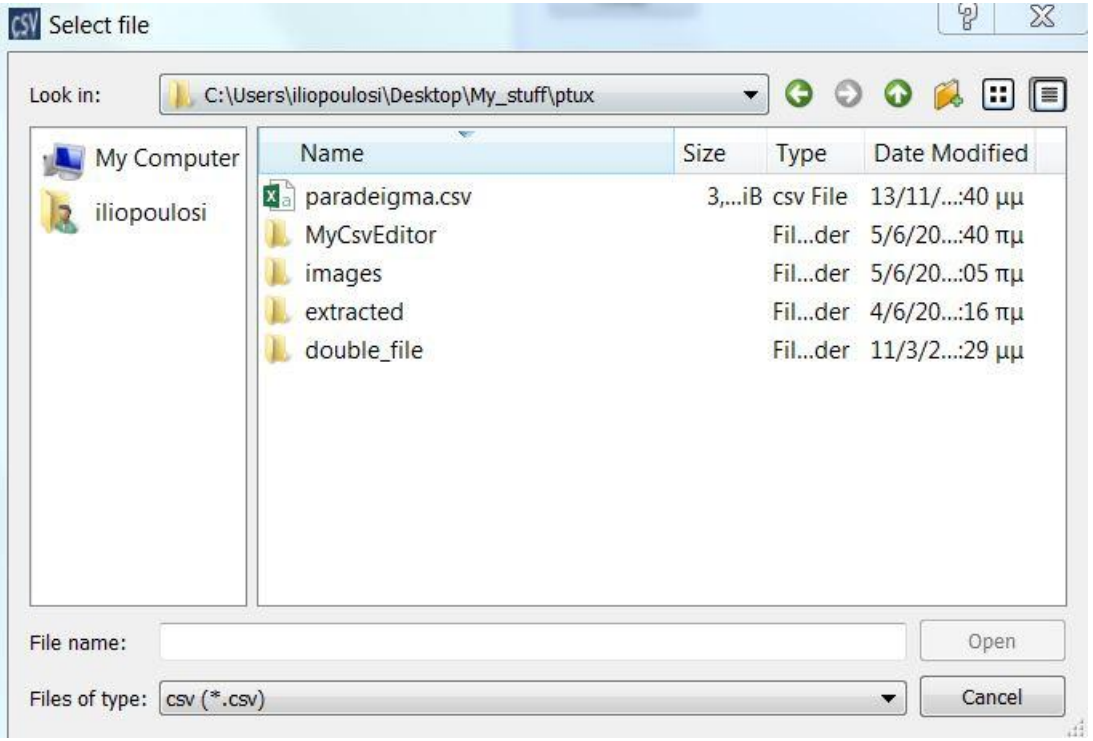
ΤΟ ΚΕΝΤΡΙΚΟ ΜΕΝΟΥ

Στο κεντρικό μενού της εφαρμογής υπάρχουν τέσσερις επιλογές:

- Open
- Open files
- Open folder
- Close



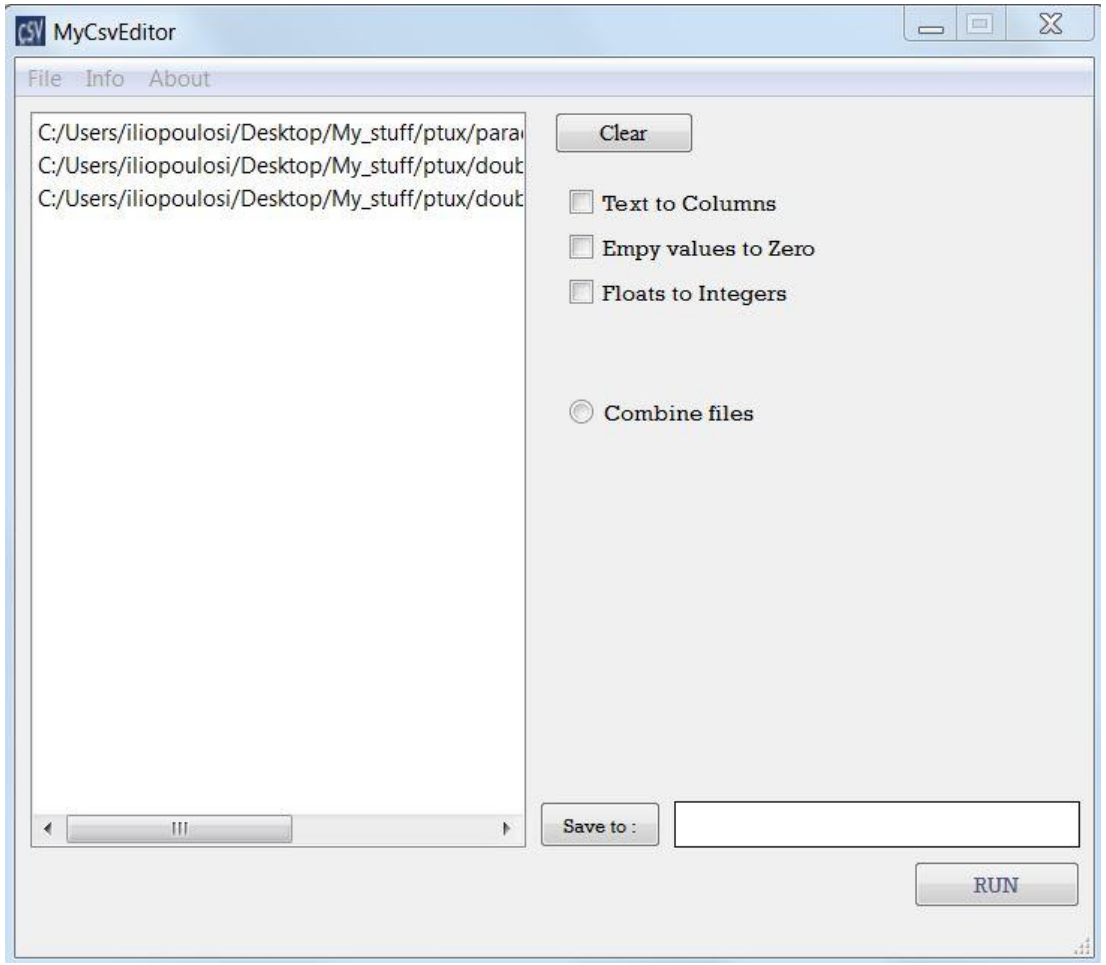
Επιλέγοντας **Open** θα αναδυθεί ένα παράθυρο στο οποίο ο χρήστης καλείται να επιλέξει ένα αρχείο στον υπολογιστή του ώστε να προστεθεί στη λίστα του προγράμματος προς επεξεργασία. Το αρχείο θα πρέπει οπωσδήποτε να είναι csv αρχείο. Αν ο χρήστης επιλέξει μία άλλη μορφή αρχείου, τότε το αρχείο δεν θα προστεθεί στη λίστα του προγράμματος.



Οι επιλογές **Open files** και **Open folder** λειτουργούν με παρόμοιο τρόπο. Στην πρώτη περίπτωση καλείται ο χρήστης να επιλέξει πολλαπλά αρχεία csv, και στη δεύτερη έναν φάκελο που περιλαμβάνει csv αρχεία.
Με την επιλογή **Close** η εφαρμογή τερματίζεται.

Η ΛΙΣΤΑ ΜΕ ΤΑ ΑΡΧΕΙΑ csv ΠΡΟΣ ΕΠΕΞΕΡΓΑΣΙΑ

Εφόσον ο χρήστης έχει επιλέξει ποια αρχεία θέλει να υποστούν επεξεργασία μέσα από τις δυνατότητες του κεντρικού μενού, τα αρχεία που επέλεξε προστίθενται σε μια λίστα, η οποία απεικονίζεται στο κεντρικό παράθυρο της εφαρμογής και προβάλλει ολοκληρωμένα τις διευθύνσεις των αρχείων αυτών στον υπολογιστή στον οποίο τρέχει η εφαρμογή.



ΤΑ ΑΠΕΙΚΟΝΙΖΟΜΕΝΑ ΠΛΗΚΤΡΑ

Στο κεντρικό παράθυρο της εφαρμογής φαίνονται τρία πλήκτρα:

- Clear
- Save to:
- RUN

Το πλήκτρο **Clear** αφαιρεί τα αρχεία που πρόσθεσε ο χρήστης στη λίστα και την αφήνει κενή.

Το πλήκτρο **Save to:** εμφανίζει ένα νέο παράθυρο όπου ο χρήστης καλείται να διαλέξει σε ποιο φάκελο θα αποθηκευτούν τα νέα παραγόμενα αρχεία από τα παλιά που υπέστησαν επεξεργασία

Το πλήκτρο **RUN** ξεκινά τη διαδικασία επεξεργασίας των αρχείων csv που βρίσκονται στη λίστα, με βάση τις επιλογές τροποποίησης που έκανε ο χρήστης.

ΕΠΙΛΟΓΕΣ ΤΡΟΠΟΠΟΙΗΣΗΣ ΕΙΣΑΓΩΜΕΝΩΝ ΑΡΧΕΙΩΝ

Δεξιά από τη λίστα με τα αρχεία που προσθέτει ο χρήστης υπάρχουν κάποια σημεία ελέγχου, checkbox, τα οποία μπορεί ο τελευταίος να επιλέξει ανάλογα με το αποτέλεσμα στο οποίο ο ίδιος αποσκοπεί:

- Text to Columns
- Empty values to Zero
- Floats to Integers
- Combine files

Επιλέγοντας **Text to Columns** στο νέο παραγόμενο αρχείο, όταν αυτό ανοιχτεί με το πρόγραμμα Excel, όλες οι τιμές διαχωρίζονται με στήλες και δεν εμφανίζονται χωρισμένες με κόμμα στην ίδια στήλη.

Επιλέγοντας **Empty values to Zero** το πρόγραμμα εντοπίζει στα αρχεία της λίστας κενές τιμές (NaN, n.a.) και τις αντικαθιστά με την τιμή μηδέν 0.

Επιλέγοντας **Floats to Integers** όλες οι δεκαδικές τιμές στα αρχεία θα μετατραπούν σε ακέραιες.

Τέλος, αν ο χρήστης θελήσει να ενώσει αρχεία csv μεταξύ τους μπορεί να το πετύχει πολύ εύκολα με την επιλογή **Combine files.**

ΚΩΔΙΚΑΣ ΓΙΑ ΤΟ ΚΕΝΤΡΙΚΟ ΜΕΝΟΥ

Στην python έχει αναπτυχθεί μια βιβλιοθήκη, η PyQt5, η οποία συμπεριλαμβάνει όλα τα αντικείμενα (objects) που χρησιμοποιούνται στην εφαρμογή. Τα objects αυτά χαρακτηρίζονται από το πρώτο γράμμα στο όνομα τους, το «Q». Τα object που περιλαμβάνονται στην εφαρμογή είναι τα εξής:

- Το QFileDialog είναι ένα module που χρησιμοποιείται σε περιπτώσεις κατά τις οποίες ο χρήστης καλείται να ανοίξει ένα αρχείο ή να αποθηκεύσει ένα αρχείο όπου εμφανίζει ένα παράθυρο με τις παρούσες διευθύνσεις του υπολογιστή
- Το QListWidget αντιπροσωπεύει ένα αντικείμενο λίστας
- Το QDialog χρησιμοποιείται για αναδυόμενα παράθυρα
- Το QLabel αντιπροσωπεύει μία ετικέτα στο πρόγραμμα
- Το QPushButton περιλαμβάνει τα πλήκτρα (buttons) της εφαρμογής

Το QCheckBox και το QRadioButton αντιστοιχεί σε κουτάκια τσεκαρίσματος (σημεία ελέγχου)

Τα πιο πάνω objects αντιστοιχούν σε όλα τα εικονιζόμενα σχέδια του κεντρικού παραθύρου της εφαρμογής.

•Το πλήκτρο **Open** από το πτυσσόμενο μενού καταχωρείται στον κώδικα με τις εντολές:

```
open = self.findChild(QtWidgets.QAction, 'actionOpen')
open.triggered.connect(self.fileopener)
```

Το όνομα του πιο πάνω αντικειμένου είναι ‘actionOpen’. Στην πρώτη εντολή αποθηκεύεται στη μεταβλητή open και στη δεύτερη εντολή καλείται η συνάρτηση fileopener η οποία ενεργοποιείται εάν πατηθεί η επιλογή **Open**. Στη συνέχεια η συνάρτηση fileopener

```
#Call open dialog for a single file
 def fileopener(self):
     Dialog.openFileNameDialog(self)
```

συνδέεται στην συνάρτηση openFileNameDialog του αρχείου dialogs.py.

```
#Open single file dialog function
    def openFileNameDialog(self):
        options = QFileDialog.Options()
        options |= QFileDialog.DontUseNativeDialog
        fileName, _ = QFileDialog.getOpenFileName(self, "Select file", "","csv (*.csv)", options=options)
        list = self.findChild(QListWidget, 'listWidget')
        QListWidget.addItem(list, fileName)
```

Στην μεταβλητή fileName εισάγεται το αρχείο που επιλέγει ο χρήστης το οποίο επιτρέπει μόνο την εισαγωγή αρχείων csv.
Το όνομα της λίστας σαν αντικείμενο του προγράμματος είναι το 'listWidget'. Αυτή αποθηκεύεται στην μεταβλητή list και τέλος, με την τελευταία γραμμή κώδικα της εικόνας 16, προστίθεται το αρχείο που επέλεξε ο χρήστης στη λίστα.

•Το πλήκτρο **Open files** από το πτυσσόμενο μενού καταχωρείται στον κώδικα με τις εντολές:

```
openfiles = self.findChild(QtWidgets.QAction, 'actionOpen_files')
openfiles.triggered.connect(self.filesopener)
```

Το όνομα του αντικειμένου είναι ‘actionOpen_files’. Στην πρώτη εντολή αποθηκεύεται στη μεταβλητή openfiles και στη δεύτερη εντολή καλείται η συνάρτηση filesopener η οποία ενεργοποιείται εάν πατηθεί η επιλογή **Open files**. Στη συνέχεια η συνάρτηση filesopener

```
#Call open dialog for multiple files
 def filesopener(self):
     Dialog.openFileNamesDialog(self)
```

συνδέεται στην συνάρτηση openFileNamesDialog του αρχείου dialogs.py.

```
#Open multiple files dialog function
    def openFileNamesDialog(self):
        options = QFileDialog.Options()
        options |= QFileDialog.DontUseNativeDialog
        files, _ = QFileDialog.getOpenFileNames(self,"Select files", "","csv (*.csv)", options=options)
        for filename in files:
            if filename.endswith(".csv"):
              list = self.findChild(QListWidget, 'listWidget')
              QListWidget.addItem(list, filename)
```

Στην μεταβλητή files εισάγονται τα αρχεία που επιλέγει ο χρήστης.
Με την επανάληψη for γίνεται έλεγχος για κάθε αρχείο και εάν είναι csv αρχείο προστίθεται στη συνέχεια στη λίστα.

•Το πλήκτρο **Open folder** από το πτυσσόμενο μενού καταχωρείται στον κώδικα με τις εντολές:

```
openfolder = self.findChild(QtWidgets.QAction, 'actionOpen_folder')
openfolder.triggered.connect(self.folderopener)
```

Το όνομα του αντικειμένου εδώ είναι ‘actionOpen_folder’. Στην πρώτη εντολή αποθηκεύεται στη μεταβλητή openfolder και στη δεύτερη εντολή καλείται η συνάρτηση folderopener η οποία ενεργοποιείται εάν πατηθεί η επιλογή **Open folder**. Στη συνέχεια η συνάρτηση folderopener

```
#Call open dialog for folder
 def folderopener(self):
     Dialog.openFolderNameDialog(self)
```

συνδέεται στην συνάρτηση openFolderNameDialog του αρχείου dialogs.py.

ΚΩΔΙΚΑΣ ΤΩΝ ΕΠΙΛΟΓΩΝ ΔΙΑΜΟΡΦΩΣΗΣ

Σε αυτό το κεφάλαιο βρίσκεται το κύριο τμήμα του κώδικα της εφαρμογής και για αυτό το λόγο δίνεται πιο λεπτομερής επεξήγηση στις εντολές που χρησιμοποιήθηκαν.
Ο χρήστης μπορεί να επιλέξει ανάμεσα σε τέσσερις επιλογές τροποποίησης των αρχείων προς επεξεργασία. Επιπλέον, έχει τη δυνατότητα να επιλέξει παραπάνω από μία επιλογή ταυτόχρονα, προκειμένου να επεξεργασθούν τα αρχεία του.
Αυτό γίνεται με τη χρήση πολλαπλής συνθήκης if (εάν... τότε...)

```
else:
    dtm = self.findChild(QtWidgets.QRadioButton, 'radioButton')
    ttc = self.findChild(QtWidgets.QCheckBox, 'checkBox')
    etz = self.findChild(QtWidgets.QCheckBox, 'checkBox_2')
    fti = self.findChild(QtWidgets.QCheckBox, 'checkBox_3')
    if ttc.isChecked() and not dtm.isChecked():
        Preferences.makecolumns(self)
    elif dtm.isChecked() and ttc.isChecked():
        Preferences.concatenate(self)
    elif dtm.isChecked() and not ttc.isChecked():
        Preferences.onlyconc(self)
```

Τα αντικείμενα των σημείων ελέγχου καταχωρούνται σε μεταβλητές και στη συνέχεια χρησιμοποιούνται στον κώδικα

Η ανάπτυξη της συγκεκριμένης εφαρμογής έχει ως κύριο σκοπό να διευκολύνει την λειτουργία της επιχείρησης και να αποτελέσει ένα χρήσιμο εργαλείο για τον αναλυτή δεδομένων, καθώς και οποιονδήποτε θα θελήσει να διαχειρισθεί, μέσω κάποιων λειτουργιών, πολλά αρχεία csv ταυτόχρονα.

Στην προσπάθεια δημιουργίας της εφαρμογής, χρησιμοποιήθηκαν ως γλώσσα προγραμματισμού, η γλώσσα python καθώς και ένα γραφικό περιβάλλον σχεδιασμού και ανάπτυξης εφαρμογών, το QtCreator.

Μέσω της εφαρμογής επιτεύχθηκε η δυνατότητα να μπορεί ένας επιστήμονας (data scientist) ή αναλυτής δεδομένων (data analyst) να κάνει κάποιες βασικές τροποποιήσεις πλήθους αρχείων csv σύντομα και με μεγάλη ευκολία.

Οι τροποποιήσεις αυτές περιλαμβάνουν ένωση αρχείων, χωρισμό τιμών από διαχωριστή κόμμα σε στήλες σε αρχεία excel, αντικατάσταση κενών τιμών με το 0 (μηδέν), στρογγυλοποίηση τιμών κ.α.

Η εφαρμογή θα μπορούσε να αναβαθμιστεί και να επεκταθεί ποικιλοτρόπως. Θα αποτελούσε έτσι ένα πολύ-λειτουργικό πρόγραμμα στο οποίο θα υπάρχουν επιπλέον δυνατότητες διαμόρφωσης όπως είναι η επεξεργασία ημερομηνίας και ώρας στα αρχεία, η αλλαγή του διαχωριστή, ανίχνευση συγκεκριμένης τιμής, αναζήτηση εύρους τιμών βάσει σύγκρισης

ΤΕΛΟΣ ΠΑΡΟΥΣΙΑΣΗΣ