Python Django Pearson

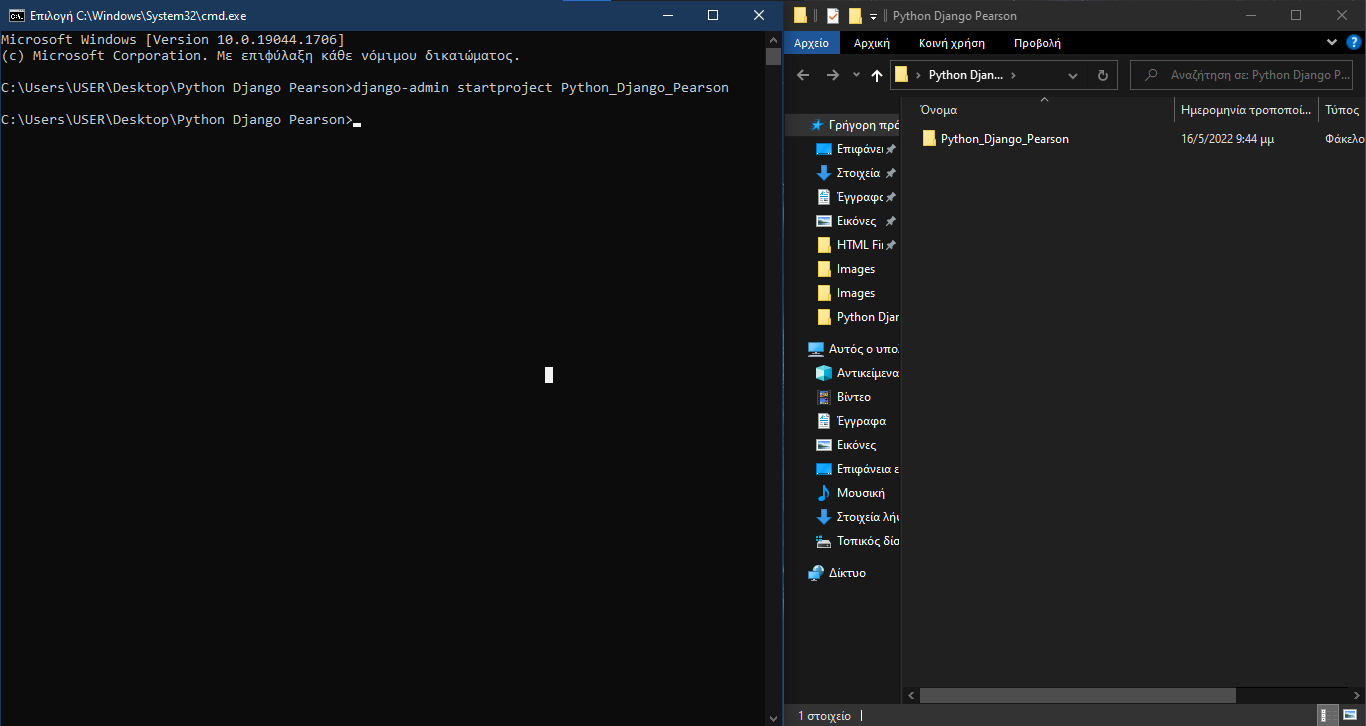
Για να γίνει η εκκίνηση της άσκησης, θα πρέπει να υπάρχουν προ-εγκαταστημένα συγκεκριμένα προγράμματα στον υπολογιστή. Αυτά τα προγράμματα είναι:

* Python version 3 και πάνω
* Εγκατάσταση του shell της Python
* Εγκατάσταση του pip που βρίσκεται στον εγκαταστάτη της Python
* Εγκατάσταση του Django μέσω του shell του συστήματος μας
* Έναν text editor

Η συγκεκριμένη άσκηση έγινε σε περιβάλλον των Windows 10, όποτε θα γίνει η εκτέλεση του μέσω cmd των Windows.

Για την εκκίνηση της άσκησης θα πρέπει να κατευθυνθούμε στον φάκελο που θα δημιουργήσουμε την εργασία μας και γράψουμε την εντολή:

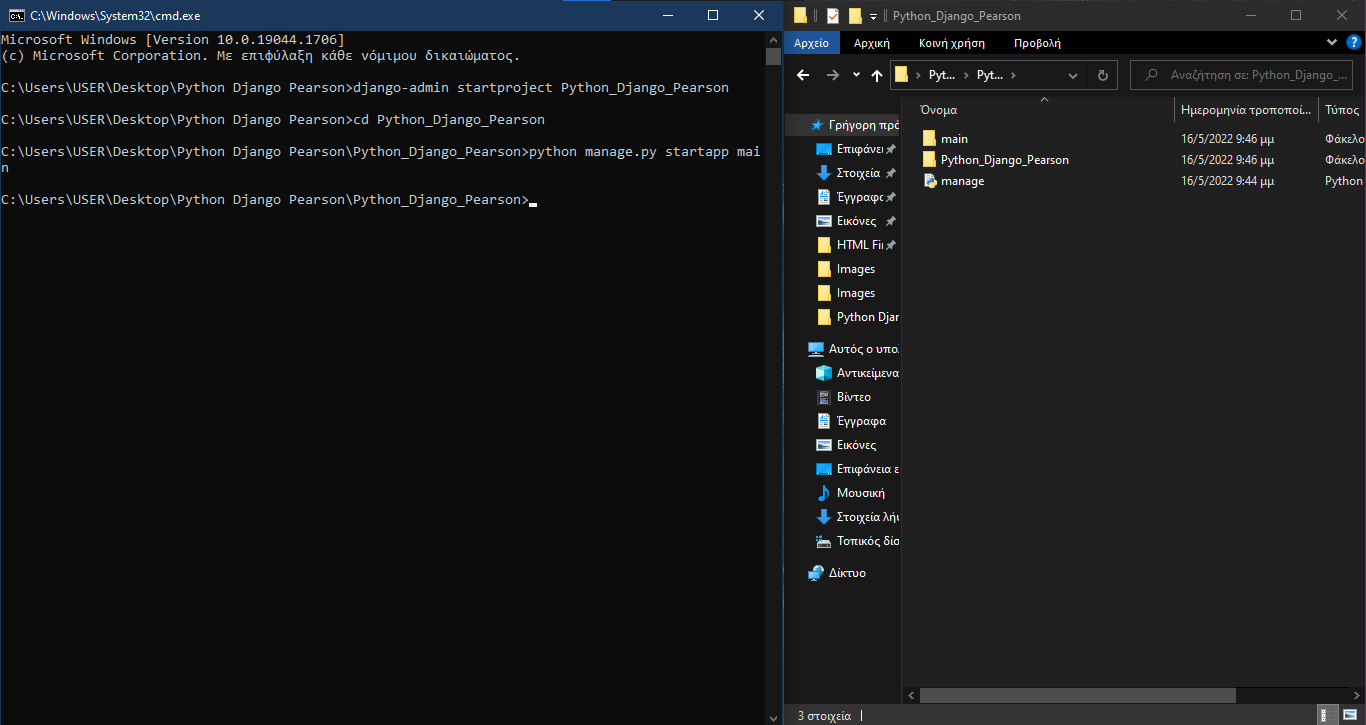
* django-admin startproject Python\_Django\_Pearson

 Έτσι, δημιουργούμε έναν φάκελο που εκεί θα ξεκινήσουμε το project μας.

Έπειτα θα πρέπει να κατευθυνθούμε μέσα στον φάκελο που δημιουργήσαμε μέσω της εντολής cd και μετά το όνομα του project μας.

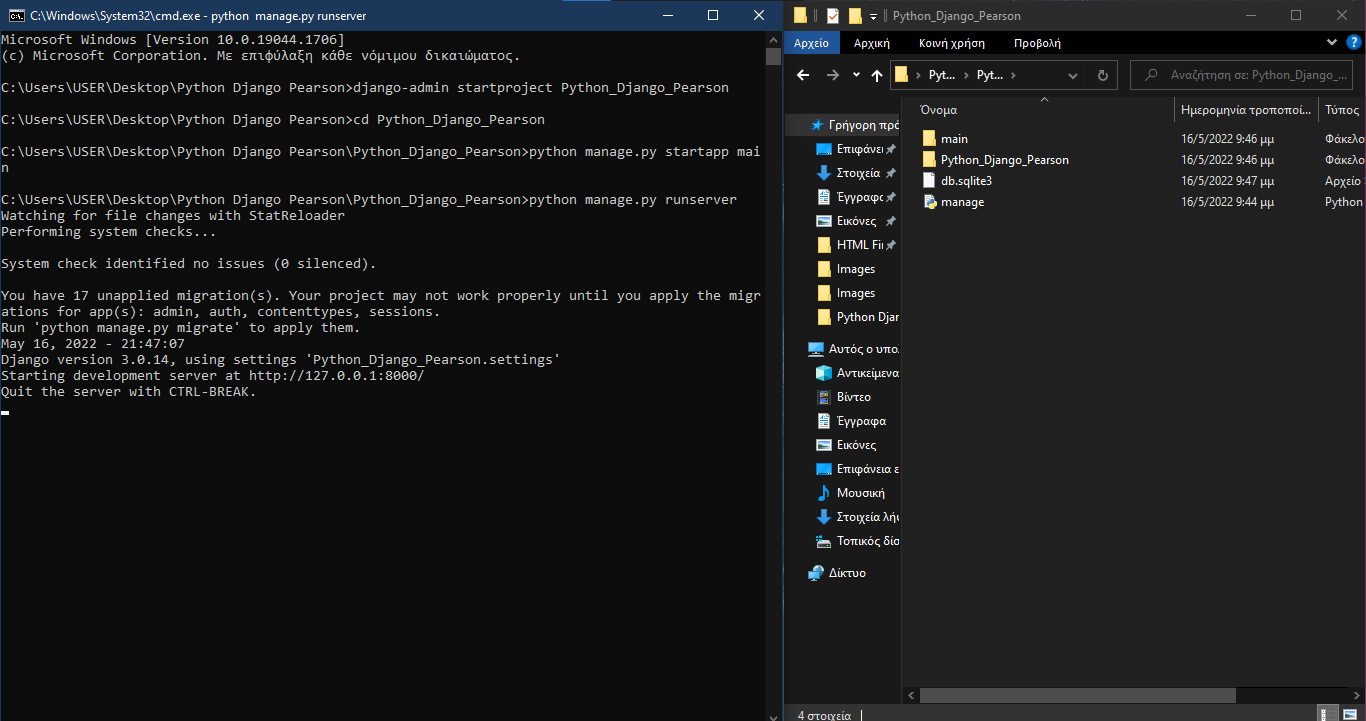
Έπειτα θα πρέπει να γράψουμε αυτήν την εντολή ώστε να ξεκινήσουμε τις τροποποιήσεις μας:

* python manage.py startapp main



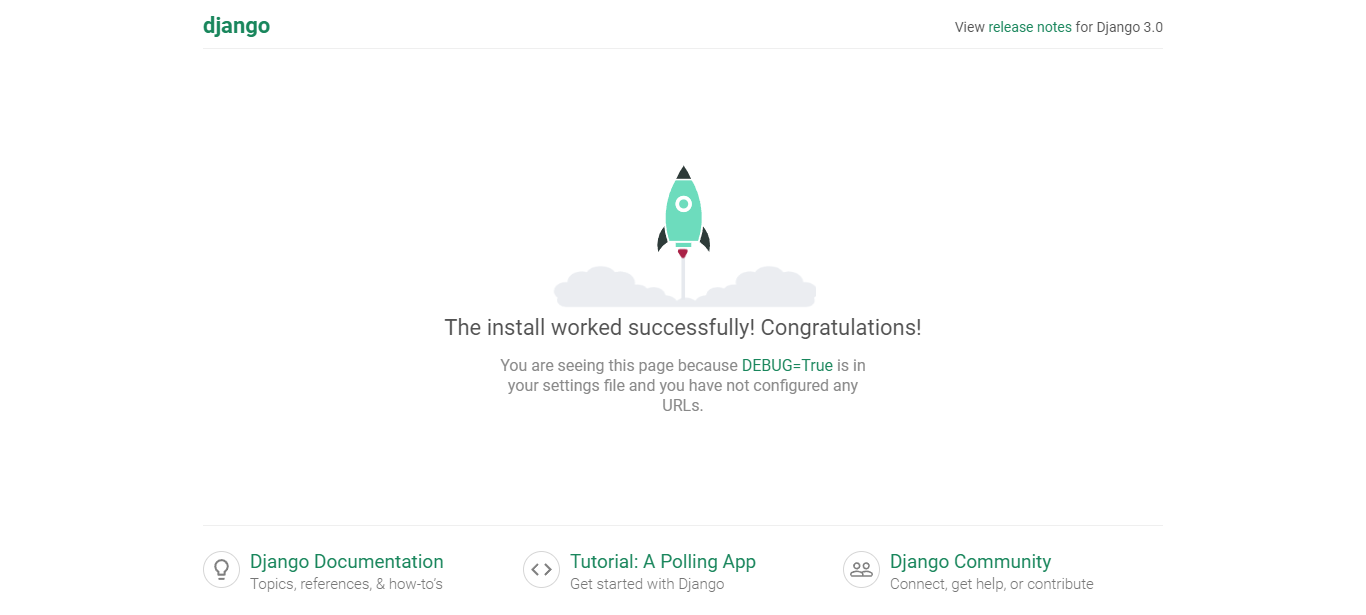
Για να αρχίσει να τρέχει ο server μας, θα πρέπει να γράψουμε την παρακάτω εντολή:

* python manage.py runserver



Με αυτό, επιτυνχάνουνε να ξεκινήσουμε έναν τοπικό server στον υπολογιστή μας και δημηουργεί το αρχείο db.sqlite3 το οποίο χρησιμοποιήτε για την χρήση της SQLlite3 που είναι εσωτερικά φτιαχμένη στο Django.

Άμα γίνει η αναζήτηση της διεύθυνσης IP που μας έδωσε το cmd, εφόσον δεν έχουμε κάνει καμία άλλη αλλαγή, μας εμφανίζει την επόμενη σελίδα που μας ενημερώνει ότι έχουμε κάνει σωστά την διαδικασία μας έως τώρα.



Για την έξοδο από τον server αυτόν, πατάμε CTRL+C στο πληκτρολόγιο στο cmd.

Όλα τα αρχεία που θα τροποποιήσουμε, βρίσκονται μέσα στα αρχεία:

* main
  + \_pycache\_
  + Migrations
  + Templates
  + \_init\_.py
  + admin.py
  + apps.py
  + tests.py
  + urls.py
  + views.py
* Python\_Django\_Pearson
  + \_pycache\_
  + \_init\_.py
  + asgi.py
  + settings.py
  + urls.py
  + wsgi.py
  + db.sqlite3
  + manage.py

**main**

**views.py**

Αυτό είναι το αρχείο που ασχολείτε με τον τρόπο με τον οποίο τα δεδομένα και οι πληροφορίες μας παρουσιάζονται στον χρήστη. Εκεί δημιουργούμε functions or classes τα οποία παίρνουν ένα web request και επιστρέφουν μια web response.

Αυτό το response θα μπορούσε να είναι είτε μια απλή HTTP response, ένα άνοιγμα ενός αρχείου template HTML ή ένα HTTP redirect που κατευθύνει τον χρήστη σε μια άλλη σελίδα.

Επίσης, κάποιες άλλες λειτουργείες του views.py, είναι ότι μπορεί να ασχοληθεί με forms, models και γενικώς της αποθήκευσης μιας πληροφορίας από τον χρήστη.

**urls.py**

Αυτό είναι το αρχείο που ασχολείτε με την διαχείρηση των συνολικών URL που μπορεί να έχουμε στο project μας. Αυτό είναι αρχείο το οποίο δημιουργούμε εμείς.

Αρχικά κάνουμε import όλα τα απαραίτητα αρχεία και πληροφορίες που θα μπορούσαμε να χρειαστούμε. Έπειτα γράφουμε urlpatterns που εκεί αποθηκεύουμε τις πληροφορίες των URL σελίδων μας. Στο πρώτο κενό βάζουμε την κατάληξη στην οποία θα έχει το site μας, στο επόμενο κενό βάζουμε σε ποια function ή class της views.py θα προοριστεί, και στο τρίτο μπορούμε να βάλουμε και ένα όνομα για το URL μας, το οποίο κομμάτι είναι προαιρετικό ανάλογα αυτό που θέλουμε να κάνουμε.

**templates**

**main**

Εκεί δημιουργούμε τις σελίδες HTML που θα χρησιμοποιήσουμε. ΠΡΟΣΟΧΗ! Το όνομα στο οποίο τοποθετούμε τα αρχεία HTML,πρέπει να έχει και το ίδιο όνομα με το startapp μας.

Μέσα στα αρχεία αυτά, έχουμε την ικανότητα να δημιουργήσουμε τα αρχεία HTML μας τοποθετώντας μέσα σ’αυτά κώδικες python βάζοντας τα σύμβολα {% %} και τελειώνοντας την εντολή αυτή με το {% endfor %},{% endif %}, e.t.c. ,καθώς και να κάνουμε extent μια σελίδα μας ώστε ο κώδικας που βρίσκεται στο αρχικό μας αρχείο, να προβάλετε και σε μια άλλη σελίδα που έχουμε βάλει αυτήν την εντολή.

Επίσης, μπορούμε να χρησιμοποιήσουμε και το blocks (εφόσον έχουμε χρησιμοποιήσει την εντολή extend από ένα άλλο αρχείο HTML), ώστε να τοποθετήσουμε συγκεκριμένες πληροφορίες μέσα σε αυτά ώστε να γίνει και η προβολή της σελίδας που έχουμε κάνει extend, και τις έξτρα πληροφορίες που έχουμε τοποθετήσει.

**Python\_Django\_Pearson**

**urls.py**

Αυτό το αρχείο ασχολείτε με τα directories που θα μπορέσει να πάρει τις πληροφορίες για την σωστή λειτουργεία του urls.py μέσα στο αρχείο της main/urls.py.

**Άσκηση**

Στην συγκεκριμένη άσκηση δημιούργησα ένα project Django το οποίο δέχεται από τον χρήστη 2 νούμερα μέσω form και 2 input και τα προσθέτει.

Για να γίνει αυτό, δημιούργησα 4 αρχεία HTML(τα οποία τοποθετούνται σε έναν φάκελο που δημιουργούμε που λέγεται templates και έπειτα έναν υπό-φάκελο main). Μία για την αρχική σελίδα, μία για την εισαγωγή των αριθμών από τον χρήστη (το οποίο έχει και τα input και το form), μία που δείχνει το αποτέλεσμα μέσα της HTML, και ένα αρχείο HTML το οποίο χρησιμοποιείτε στις σελίδες add και Calculator μέσω της εντολής extend.

Έπειτα τοποθέτησα στο αρχείο views.py της main 3 classes, 1 για την αρχική σελίδα, 1 για την εισαγωγή δεδομένων και 1 για την αναπαράσταση του αποτελέσματος. Ταυτόχρονα στοπ αρχείο urls.py μέσα στην main, τοποθέτησα τα ονόματα των URLS μαζί και σε ποιο απ’το καθένα αντιστοιχεί στην κάθε class μέσα στο αρχείο views.py.

Στο αρχείο base.html, βρίσκονται οι πληροφορίες για την σελίδα HTML που εφαρμόζεται και στο αρχείο Calculator.html και στο αρχείο add.html μέσω της εντολής extend στα αρχεία αυτά. Μέσα στο αρχείο αυτό, τοποθετούνται και οι εντολές {% block title %}, {% block style %} και {% block content %} (που τελειώνουν με την εντολή {% endblock %} ) στις οποίες τοποθετούμε ή αρχικά προεπιλεγμένες επιλογές στο αρχικό αρχείο μας(για τον τίτλο της σελίδας μας), ή τοποθετούμε δικές μας πληροφορίες από το αρχείο που βάλαμε το extend.

Μέσα στο αρχείο Calculator.html, υπάρχουν 2 input(το 1ο με όνομα NumberX και το 2ο με όνομα NumberY), 1 button και 1 form (το οποίο αναφέρουμε τις πληροφορίες που τοποθετούμε στην HTML). Τα input δέχονται μόνο νούμερα και μέσω του button, μας κάνει redirect σε μια άλλη σελίδα HTML που έχουμε φτιάξει που προβάλει την απάντηση.

Μέσα στο αρχείο views.py, στην class που θα μάς προβάλλει την σελίδα που δείχνει το αποτέλεσμα της πράξης, το αντιστοιχούμε σε μια μεταβλητή(που δηλώνουμε αρχικά μετά το render, το είδος της πληροφορίας και της σελίδας HTML μας σε μια βιβλιοθήκη καθορίζοντας και ένα όνομα για την μεταβλητή μας). Η πληροφορία που τοποθετεί ο χρήστης στην σελίδα Calculator.html μέσω των input, τις μετέτρεψα σε interger βάζοντας και το είδος της πληροφορίας, την μέθοδο που χρησιμοποιεί η form μας και το όνομα που περιέχει το κάθε input, διότι από προεπιλεγμένη τιμή, η πληροφορία που τοποθετεί ο χρήστης είναι σε μορφή string και αντί για την πρόσθεση των τιμών, μάς δημιουργεί την ένωση 2 κειμένων.

Πριν ξεκινήσει η δημιουργία ενός τοπικού server στον υπολογιστή μας, για να μην μάς προκαλέσει κάποια βλάβη, γράφουμε την εντολή αυτήν στην cmd:

* python manage.py migrate

Μετά από αυτό, τρέχουμε τον τοπικό server μας αντιγράφοντας τον κώδικα URL που μάς προκύπτει. Αρχικά θα μας κάνει direct στο Homepage.html που εκεί πρέπει να πατήσουμε το link για τον Calculator.html. Μόλις προσδιορισθούμε εκεί, τοποθετούμε τα νούμερα που θέλουμε να κάνουμε την πρόσθεση και πατάμε το κουμπί για την πρόσθεση. Αυτό θα μας προσδιορίσει στην σελίδα add.html που εκεί δείχνει το αποτέλεσμα μας. Για την επιστροφή στην σελίδα που γίνεται η πράξη, ο χρήστης χρειάζεται να πατήσει το link για το Calculator που βρίσκεται στην πλαϊνή μπάρα.