UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA
FACULTATEA DE AUTOMATICA ȘI CALCULATOARE

ADMINISTAREA SISTEMELOR DE OPERARE

Deploying Django and Docker

Student: Ioana Vîrnă

Profesor îndrumător: Andrei Bogdan Leucuta

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Introducere

In acest proiect, va fi realizat un web-site minimal si se va simula lansarea acestuia in productie.

Site-ul va fi realizat folosind framework-ul Django, bazat pe Python.

Realizarea sa va cuprinde 3 etape si anume:

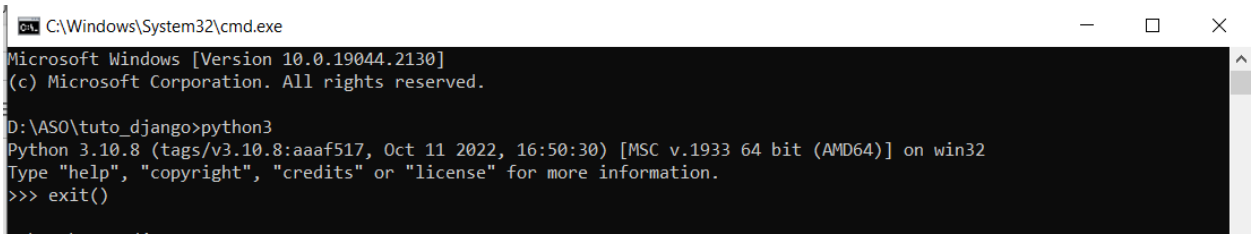
- Etapa 1: Instalare Python, Django si crearea unui site minimalist
- Etapa 2: Crearea unui chat minimalist
- Etapa 3: Presupunand ca site-ul este functional, il pregatim de lansarea in piata

Etapa I

In aceasta etapa se vor instala resursele necesare in realizarea temei si se vor implementa cerintele minimale, avand astfel o baza a proiectului.

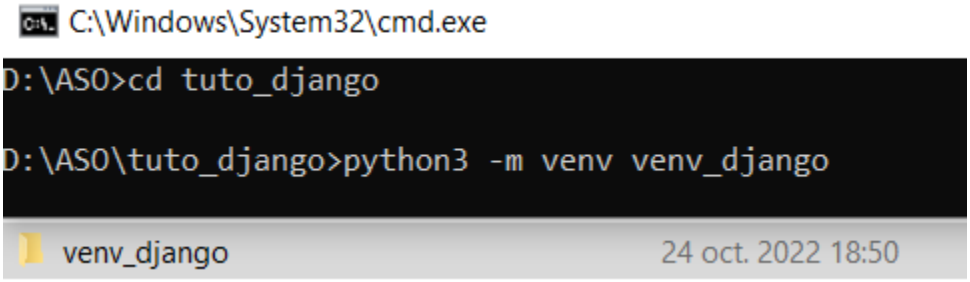
Pasi pentru crearea proiectuli de baza:

1. Verificarea existentei **python3**. Daca nu exista, vom fi ghidati s ail instalam.



2. Deschidem linia de comanda din locatia unde vom dori sa cream proiectul

3. Cream un director numit venv_django, utilizand comanda **python3 -m venv venv_django**



4. Activam virtual environment-ul creat. Se poate observa ca este activat atunci cand in fata prompt-ului obisnuit de CMD, avem intre paranteze numele al virtual environmentului.

```
D:\ASO\tuto_django>venv_django\Scripts\activate

(venv_django) D:\ASO\tuto_django>pip list
Package    Version
---------- -------
pip        22.2.2
setuptools 63.2.0

[notice] A new release of pip available: 22.2.2 -> 22.3
[notice] To update, run: python.exe -m pip install --upgrade pip
```

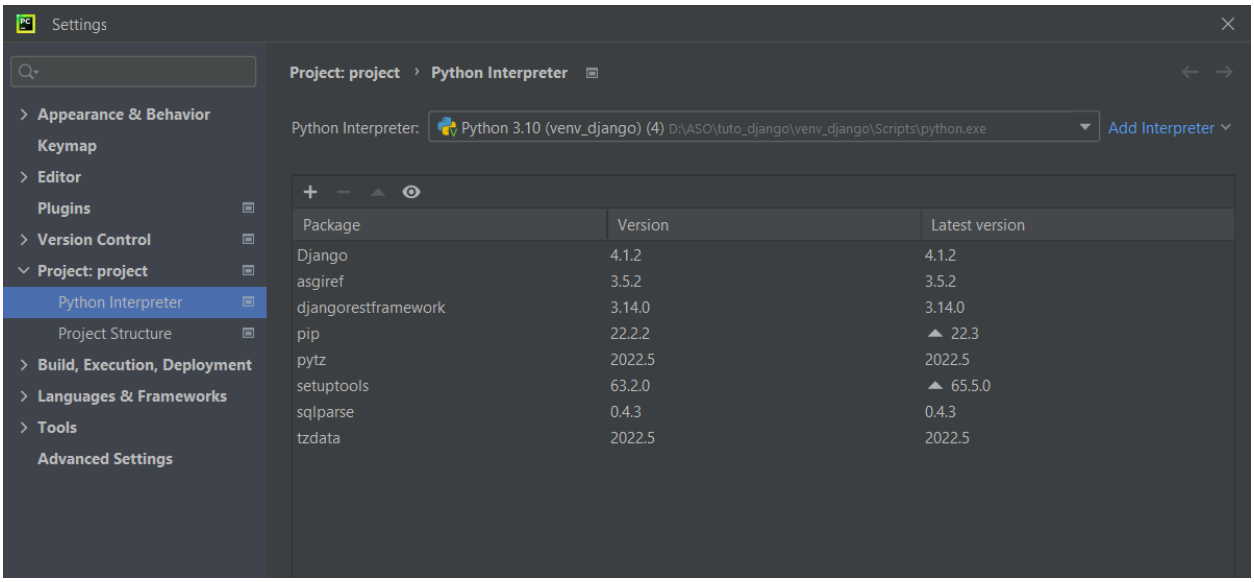
5. Instalam modulele Django si djangorestframework utilizand comanda **pip**

```
(venv_django) D:\ASO\tuto_django>cd venv_django

(venv_django) D:\ASO\tuto_django\venv_django>pip install django djangorestframework
Collecting django
  Using cached Django-4.1.2-py3-none-any.whl (8.1 MB)
Collecting djangorestframework
  Using cached djangorestframework-3.14.0-py3-none-any.whl (1.1 MB)
Collecting tzdata
  Using cached tzdata-2022.5-py2.py3-none-any.whl (336 kB)
Collecting asgiref<4,>=3.5.2
  Using cached asgiref-3.5.2-py3-none-any.whl (22 kB)
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.3-py3-none-any.whl (42 kB)
Collecting pytz
  Using cached pytz-2022.5-py2.py3-none-any.whl (500 kB)
Installing collected packages: pytz, tzdata, sqlparse, asgiref, django, djangorestframework
Successfully installed asgiref-3.5.2 django-4.1.2 djangorestframework-3.14.0 pytz-2022.5 sqlparse-0.4.3 tzdata-2022.5

[notice] A new release of pip available: 22.2.2 -> 22.3
[notice] To update, run: python.exe -m pip install --upgrade pip
```

6. Din directorul care contine directorul cu virtual environmental creat, initiem un proiect Django cu ajutorul comenzii **Django-admin startproject project**

```
(venv_django) D:\ASO\tuto_django\venv_django>cd ..

(venv_django) D:\ASO\tuto_django>django-admin startproject project
```

7. Folosind mediul de dezvoltare **PyCharm**, deschidem directorul tuto_django/project. Configuram din setari interpretorul de python (mediul virtual creat mai devreme)



8. Invocam pornirea serverului utilizand comanda **python manage.py runserver** din terminalul PyCharm. Se va crea o aplicatie web Django, insa fara o baza de date.

```
PS D:\ASO\tuto_django\project> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply t
he migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
October 24, 2022 - 18:57:03
Django version 3.2, using settings 'project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

9. Cream aplicatia scumboard prin comanda **python manage.py startapp scrumboard**. Dupa ce s-a creat, in sectiunea INSTALLED_APPS din setting.py vom adauga si ‘scumboard’.

```
PS D:\ASO\tuto_django\project> python.exe .\manage.py startapp scrumboard
PS D:\ASO\tuto_django\project> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...
```


```python
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'scrumboard',
]
```

10. Definim modelele aplicatiei, **List** si **Card** in models.py din directorul scrumboard.

settings.py | models.py

```python
from django.db import models

# Create your models here.
class List(models.Model):
    name=models.CharField(max_length=50)

class Card(models.Model):
    title = models.CharField(max_length=100)
    descriptiom=models.TextField(blank=True)
    # relation with foreign key below
    # Each Card must belong to a list
    list= models.ForeignKey(List, related_name="cards",on_delete=models.CASCADE)
    story_points = models.IntegerField(null=True, blank=True)
    business_value=models.IntegerField(null=True, blank=True)

```

11. Apelam comanda **python manage.py makemigrations**, urmata de **python manage.py migrate**. Makemigrations doar pregateste baza de date pentru manipulare, migrate o modifica efectiv.

```
You have 18 unapplied migration(s). Your project may not work properly until you apply t
he migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying scrumboard.0001_initial... OK
  Applying sessions.0001_initial... OK
PS D:\ASO\tuto_django\project>
```

12. Daca ne uitam in SQLite, vom putea vedea tabelele scrumboard_card si scrumboard_list unde vom putea vedea campurile definite in clasele List si Card din models.py