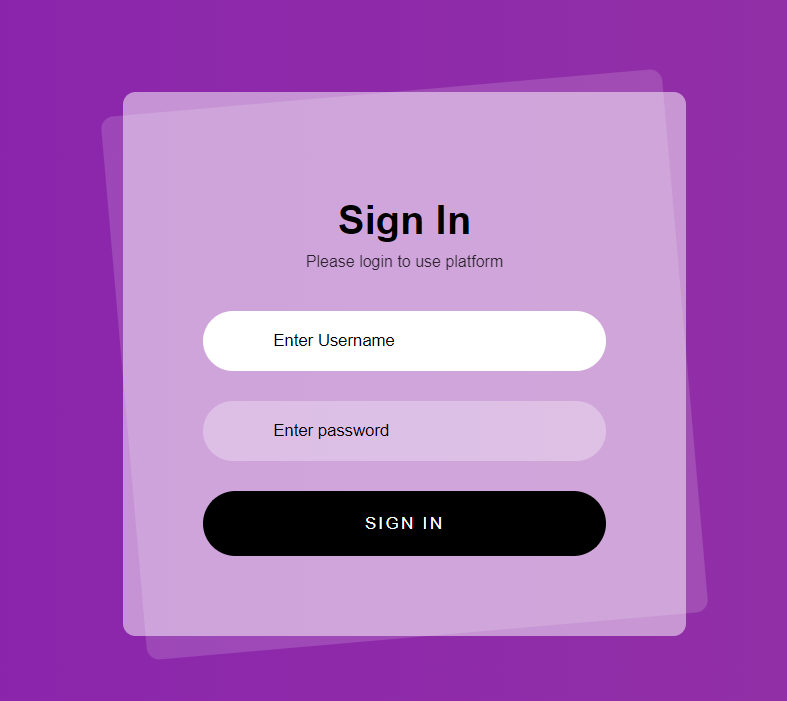
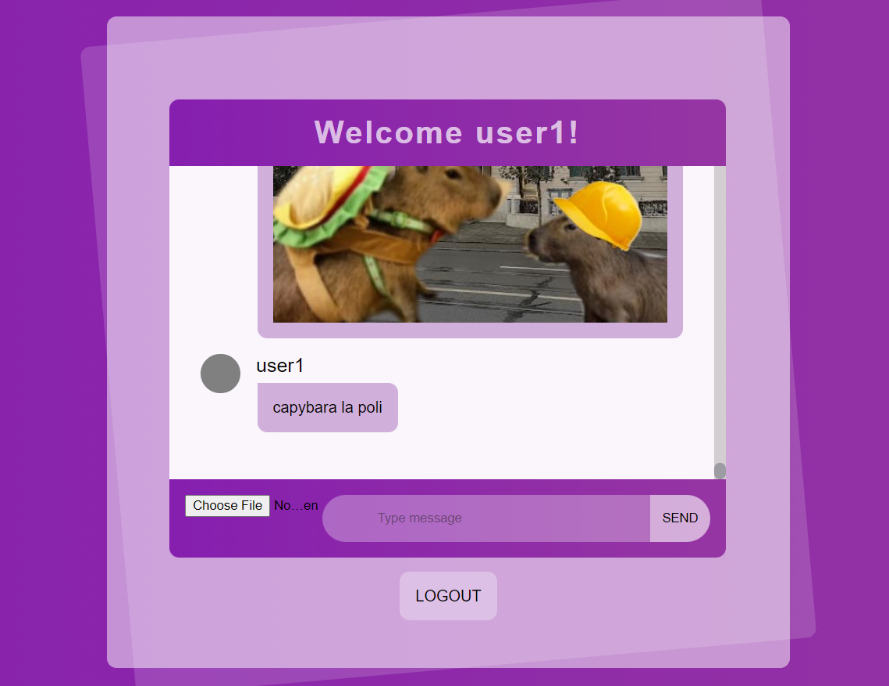
Aplicație de chat în Python-Django

Jurnal

Dunca Denisa Mihaela

Aplicația de chat este o aplicație ce utilizează ca și tehnologii Python și Django precum si HTML și CSS. Aplicația are funcționalitatea de login pentru useri, acești odată logați, sunt redirectați la o pagina de home unde pot să vadă toate mesajele până atunci trimise și pot să trimită la rândul lor mesaje și fotografii.

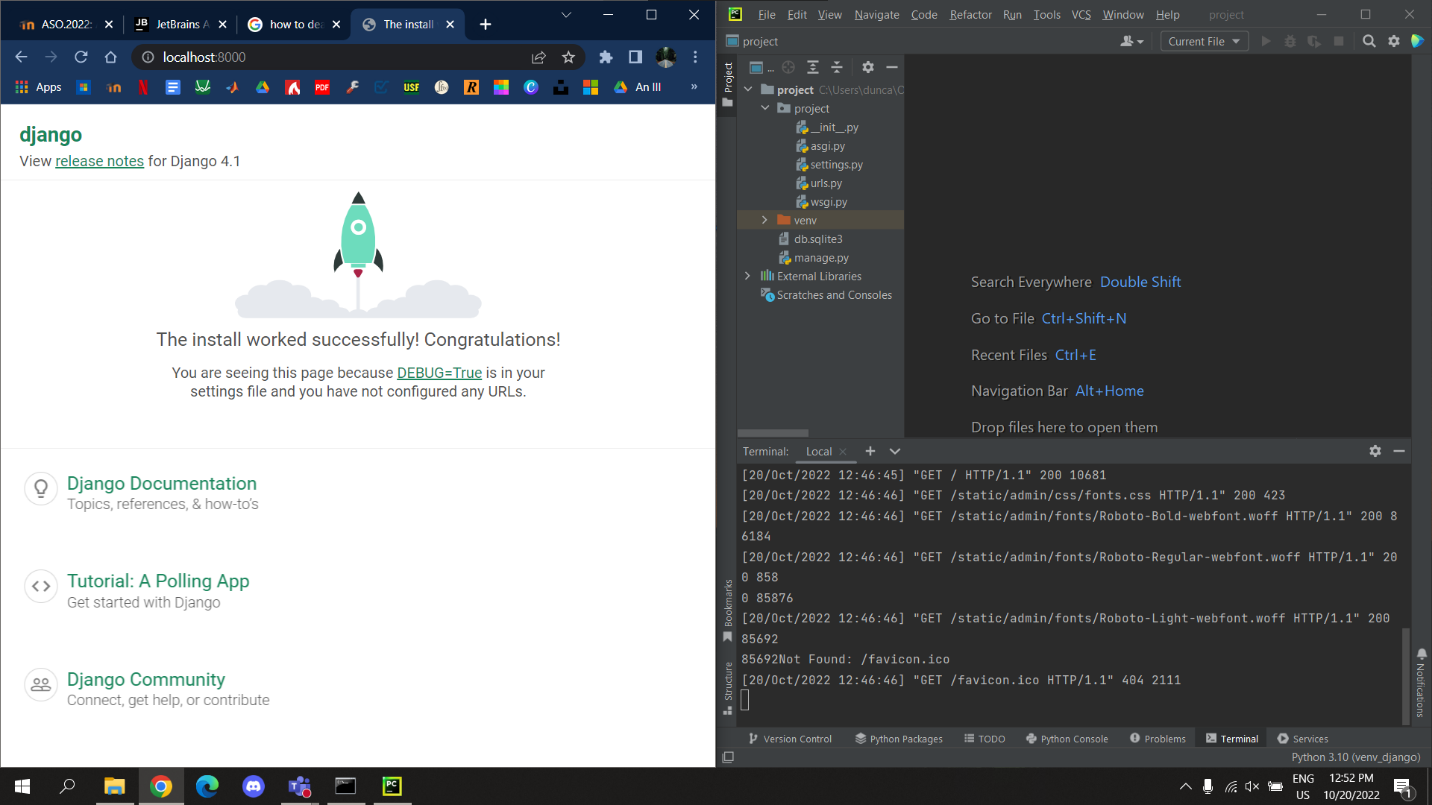




**Modul de rezolvare**

Graphical user interface, website

Description automatically generated**Faza 1**

Primul pas pentru crearea proiectului a fost parcurgerea tutorialului de creare a unui proiect Django. Apoi folosind comanda python manage.py runserver am pornit aplicația pe linkul localhost:8000.

După ce am adăugat partea de aplicație am putut sa accesez baza de date cu userii si mesajele.

**Faza 2**

Apoi am început sa creez adresa de url pentru login si userii din baza de date. Am făcut tamplate ul pentru partea de login si in view.py am adăugat metodele de logare si delogare. Pentru a realiza partea de login si autentificare am folosit din django.contrib.auth metodele authenticate, login, logout.

Dupa am creat o pagina de home, in care userii pot trimite mesaje, pot sa le vada si sa se delogeze. Pentru asta am creat un nou tamplate numit home si o funtie noua in views.py numita sendMessage. Aceasta trimite mesaje de tipul text in baza de date. Am adaugat si ca functionalitate partea de trimitere poze aceste poze sunt trimise spre FileSystemStorage si sunt salvate local in fisierul media.

Pentru a realiza partea cu postatul pozelor in settings.py am adugat urmatoarele:

O imagine care conține text

Descriere generată automat

De asemenea in urls.py am adăugat si:

O imagine care conține text

Descriere generată automat

Aplicația ia numele fișierului ales din interfața si verifica daca conține extensia ”.png”, in caz ca da pe front in loc sa fie trimis doar textul, va fi trimis un url pentru imagine.

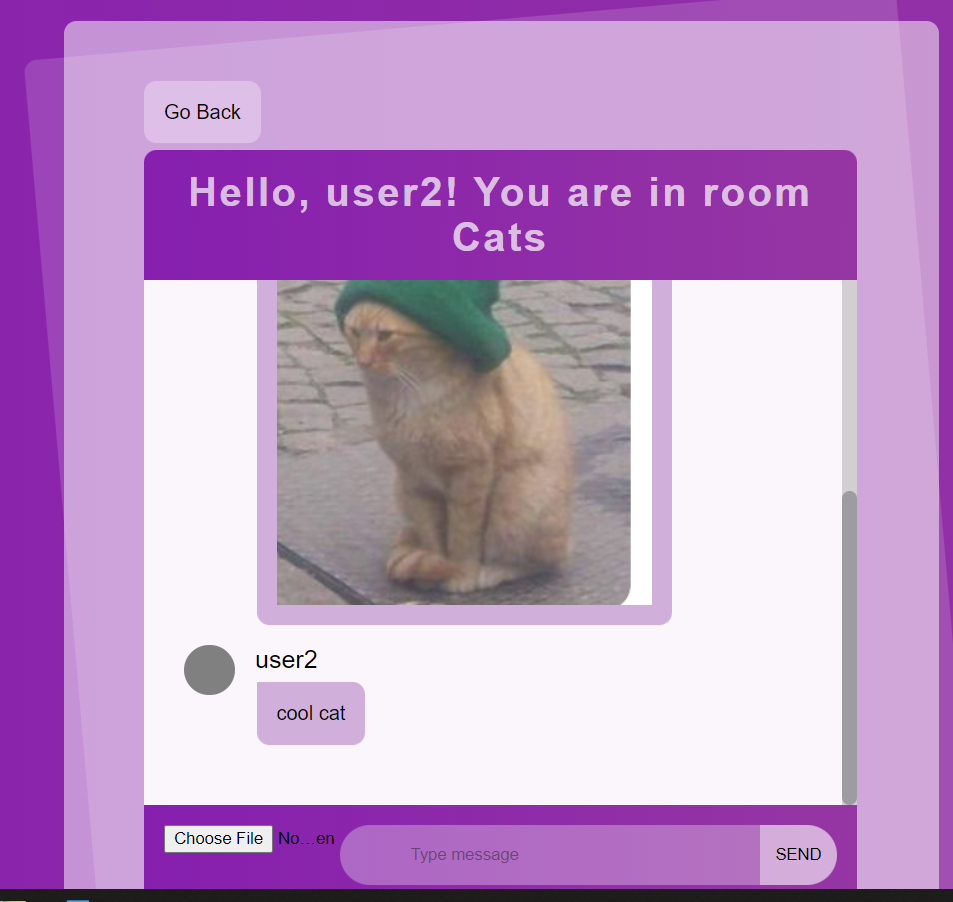


Pentru a realiza acest lucru am importat din django si default\_storage.

O imagine care conține text

Descriere generată automat

A doua funcționalitate adăugată in proiect este posibilitatea utilizatorilor de a crea camere si de a scrie mesaje in fiecare camera creata. Userul poate sa dea un nume camerei si sa creeze o camera sau poate sa selecteze din camerele deja create.

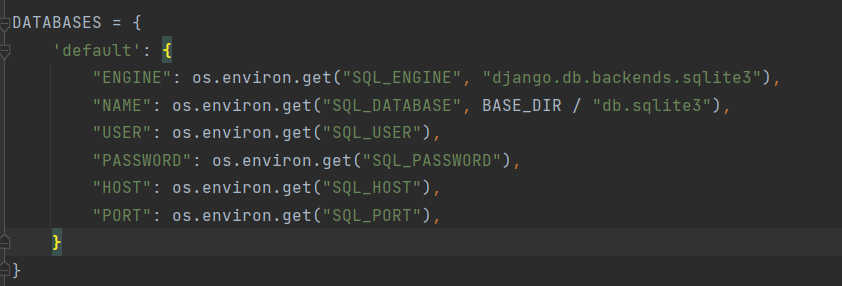


Așa arata interfața pentru trimitere de mesaje in camerele de chat.

**Faza 3**

Am instalat Docker si am creat un fisier Dockerfile si un fisier docker-compose.yml pentru aplicatie. In docker-compose.yml am adaugat si specificatiile pentru crearea containerului de postgresSQL.

Text

Description automatically generated

Dupa am testat daca aplicatia functioneaza in modul Developer pe localhost:8000.

Graphical user interface, application

Description automatically generated

Apoi pentru a configura aplicatia pentru production am creat docker-compose.prod.yml si Dockerfile.prod.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidenceApoi am adaugat si partea de nginx si am configurat setarile pentru fisierele statice si media.

Dupa am testat daca aplicatia functioneaza in Production pe localhost:1337.

Graphical user interface, text, application

Description automatically generated

Pentru a face ca aplicatia sa functioneze si pe Development si pe Production am creat fiisre de env pentru dev si pentru prod.

Graphical user interface, text

Description automatically generated

Asa arata containerul din Docker cu aplicatia care ruleaza si pe Development si pe Production:

Graphical user interface, text, application

Description automatically generated

Comenzile utilizate pentru a construi containerele si pentru a face migrarea datelor pentru postgresSQL:

Text

Description automatically generated

**Probleme întâlnite și modul de rezolvare**

Singurele probleme întâlnite au fost la integrarea funcționalității de trimitere de imagini, deoarece având la dispoziție puține informații si soluții de pe net legate de trimiterea de imagini a durat ceva pana sa îmi dau seama cum funcționează de fapt File System Storage.Încă o problema întâlnită atunci când am început sa adaug css este partea de linkurile intre html si css întrucât aici trebuie făcută prin-un director static.

Pentru deploy-ul pe production am intampinat probleme la adaugarea fisierelor static si media, dar le-am rezolvat.

**Concluzii**

Mi-a plăcut sa realizez această aplicație, am utilizat tehnologii noi și am învățat cum se utilizează o aplicație care are si frontendul si backendul in același proiect (diferit de Java cu Spring si React învățate pana acum).