Proiect ASO

Jurnal Faza 3

Student: Suciu Ioana Magdalena

Grupa: 30644

Cuprins

[Cerințe rezolvate 3](#_Toc121384330)

[Modul de rezolvare 3](#_Toc121384331)

[**Descriere structurală** 3](#_Toc121384332)

[**Aplicația Account** 3](#_Toc121384333)

[**Aplicația Chat** 3](#_Toc121384334)

[**Pachete utilizate** 4](#_Toc121384335)

[Probleme întâlnite și modul de utilizare 4](#_Toc121384336)

[Bibliografie 4](#_Toc121384337)

[Git: 4](#_Toc121384338)

# Cerințe rezolvate

* Instalarea Docker si crearea unui Dockerfile pentru aplicația realizata in faza 2 a proiectului
* In aplicația realizată în cadrul fazei 2 au fost modificate setarile astfel incat baza de date sa nu mai fie un fisier SQLite, ci o baza de date reala, in cazul acesta s-a utilizat Postgres. Serverul bazei de date rulează de asemenea intr-un continer Docker separat. Cele doua servicii sunt definite intr-un fisier docker-compose.yml

# Modul de rezolvare

Pentru a realiza o rezolvare cât mai completă a cerințelor a fost necesară împărțirea proiectului în două sub aplicații specifice Framework-ului Django. O primă aplicație a fost realizată pentru a asigura conectivitatea utilizatorului în aplicație, iar o a doua aplicație a fost realizată pentru a gestiona comunicarea prin mesaje ale utilizatorilor. În cazul fișierelor .html a fost utilizată biblioteca Bootstrap pentru realizarea stylingului aplicației.

## **Descriere structurală**

În continuare vor fi descrise punctual cele două părți ale aplicației

### **Aplicația Account**

În cadrul acestei părți ale aplicației am realizat procesul de autentificare a unui utilizator în aplicație, în cazul în care un nou utilizator dorește să își creeze un cont această opțiune este oferită de aplicație.

Pentru a sigura procesul de autentificare a fost utilizată procedura de autentificare oferită de Django în pachetul ***django.contrib.auth.*** Datorită utilizării acestui pachet nu a mai fost necesară crearea de modele specifice utilizatorilor. Pentru a asigura siguranța în aplicație toate funcționalitățile în care este necesară autentificarea utilizatorului au fost securizate.

Pentru accesul în aplicație au fost definite trei url-uri: pentru crearea unui cont, pentru autentificare și pentru ieșirea unui utilizator din cont.

În fișierul View.py au fost definite 3 funcții, fiecare proces având o funcție specifică, iar pentru procedurile de autentificare și înregistrare au fost construite fișiere tamplate care au ca scop interfața utilizator.

### **Aplicația Chat**

În cadrul aplicației Chat a fost implementată logica de bază a aplicației. Pentru asigurarea mai multor camere de chat a fost necesară crearea unui model numit Chat în baza de date care are ca atribute un nume și o listă a tuturor utilizatorilor care sunt înregistrați în cameră. Este de menționat că un utilizator are posibilitatea de a intra într-un astfel de chat doar dacă este invitat sau își creează propria cameră de chat. Pentru reținerea mesajelor a fost construit un model numit Message care are ca principal scop stocarea de records de tip mesaj, acestea având ca atribute numele camerei de chat unde au fost trimise, numele utilizatorului care a trimis mesajul, textul mesajului și momentul în care a fost trimis mesajul. De fiecare dată când aceste modele au suferit transformări o nouă migrațiune a bazei de date a trebuit rulată.

Plecând de la premisa că un grup de utilizatori vor comunica în timp real a fost necesară realizarea mai multor path-uri url prin care se pot trimite mesaje între interfața utilizator și backend. Rutările au fost realizate preponderent statică, excepție făcând URL-ul de transmitere a mesajelor de către utilizator.

Pentru fiecare astfel de path a fost creată o metodă în fișierul View al aplicației. Pentru comunicarea intre frontend și backend au fost utilizate doar request-uri API de tipul GET și POST.

Comunicarea între utilizatori se realizează în timp real. Acest lucru a fost posibil prin utilizarea unor JQuery care vor face apeluri de API din partea de interfață și a de impresia unei încărcări în tip real a mesajelor.

## **Pachete utilizate**

**Widget Tweaks 🡪** utilizat pentru realizarea și customizarea formularelor de înregistrarea și autentificare

# Probleme întâlnite

Instalarea pachetelor si a versiunilor cerute impșreuna cu realizarea containerului pentru simularea modului de productie al aplicatiei.

# Bibliografie

[How To Build A Realtime Chat App With Django (Part 2/2) - YouTube](https://www.youtube.com/watch?v=ynyk1z8NLr8&t=922s)

[Cards · Bootstrap (getbootstrap.com)](https://getbootstrap.com/docs/4.3/components/card/)

[Dockerizing Django with Postgres, Gunicorn, and Nginx | TestDriven.io](https://testdriven.io/blog/dockerizing-django-with-postgres-gunicorn-and-nginx/)

# Git:

[IoanaMagdalenaSuciu/ASO\_Faza3 (github.com)](https://github.com/IoanaMagdalenaSuciu/ASO_Faza3)