Documentatie

Proiect ASO

Faza 2

Gorgan Raul - Alexandru

Grupa 30641

Cerinte rezolvate

In aceasta etapa a proiectului se cere implementarea unui chat minimalist intre mai multi utilizatori. Utilizatorii se pot loga, trimite mesaje si vizualiza mesajele postate. In plus, cerintele realizate sunt: posibilitatea de a crea camere de chat (de catre utilizatori) si trimiterea de mesaje in acele camere; pe langa mesaj text, utilizatorii pot posta si imagini, care vor fi salvate ca fisiere media.

Modul de rezolvare

Pentru comunicare am folosit Channels. Acest pachet extinde capabilitatile Django pentru a putea fi folosit in proiecte de care e nevoie de conexiuni de lunga durata, nu doar HTTP.

A fost creat un proiect iar apoi o aplicatie pentru acest proiect. Aceasta prima aplicatie, numita ‘core’, contine partea de baza a proiectului (login, register, template-ul de baza pentru paginile din cadrul aplicatiei). Utilizatorul se poate loga sau isi poate crea un cont, cu care sa intre in aplicatie. Dupa login, acesta va fi directionat la pagina care contine camerele de chat si form-ul de creare a unei camere de chat noi. Pentru partea de camera de chat am creat o aplicatie noua, numita ‘room’. Aceasta contine template-urile pentru pagina principala si paginile fiecarei camere de chat. Legaturile cu backend-ul sunt realizate in fisierele urls.py. Fiecare aplicatie contine un astfel de fisier unde au fost mapate toate url-urile necesare. Actiunile de pe fiecare pagina au fost specificare in fisierele views.py din fiecare aplicatie. Pentru form-urile necesare, s-a creat un fisier forms.py, in care au fost declarate. Pentru login si logout au fost folosite template-urile standard LoginView si LogoutView. Modelele aplicatiei sunt: Room, care contine un nume si un id; Message cu campurile: room, user, content (mesaj) si data adaugarii. A fost folosita baza de date SQLite folosind ORM-ul din Django.

Pentru configurare, aplicatiile create (core si room) au fost trecute in fisierul settings.py din proiect, la fel si ‘channels’, pentru ca Django sa stie ca le folosim in proiect. In acest fisier au fost declare si url-urile pentru login, logout, pentru fisiere media si locul de salvare a acestora, si a fost declarat ASGI\_APPLICATION si layer-ele pentru a putea folosi Channels.

Managementul de mesaje si conexiune la camera de chat este realizat in fisierul consumers.py, care contine metodele connect (pentru conectare), disconnect (pentru deconectare), receive, chat\_message (prin care se trimit mesajele) si save\_message (prin care se salveaza mesajele in DB).

Design-ul pentru paginile HTML a fost realizat cu CSS.

Probleme intalnite si modul de rezolvare

O prima problema a fost utilizarea limbajului de programare Python, pentru care a fost nesesara consultarea de documentatii si utilizarea Django pentru care au fost necesare tutoriale si documentatii care sa descrie modul de functionare a componentelor.

Probleme am intalnit la realizarea cerintei de trimitere de imagini in loc de text. Solutia a fost cautarea si urmarirea de tutoriale care sa ajute la o buna intelegere a modului de functionare a mecanismului de comunicare, a modului de salvare a fisierelor media si diferenta dintre secventele de procesare a unui mesaj text si a unui mesaj multi-media.

Concluzii

Realizarea acestei faze de proiect a adus multe beneficii, precum: familiarizarea cu limbajul de programare Python si ulitizarea unor functionalitati oferite Django; familiarizarea cu WebSockets; intelegerea multor concepte de baza in comunicarea pe web si modul de functionare al acesteia.

Link catre repository-ul de Github cu codul sursa: https://github.com/gorganraul23/ASO-Chat-App/tree/master