

# Jurnal Faza 2 / Partea 1

## Cerințele rezolvate

Pentru a doua parte a proiectului am facut dezvoltarea aplicatiei folosind Django pe partea de frontend si backend(django template). S-a dezvoltat o aplicatie in care userul isi poate face cont si se poate loga pentru a comunica cu diferiti useri din baza de dat.

## Modul de rezolvare

Pentru a realiza logarea am folosit baza de date deja implementata in Django(Users) si pe pagina de sign up userul trebuie sa introduca un username si parola pentru a fi introdus in baza de date si sa se poata loga. Pentru aceasta am folosit functiile din libraria django.contrib.auth.

data

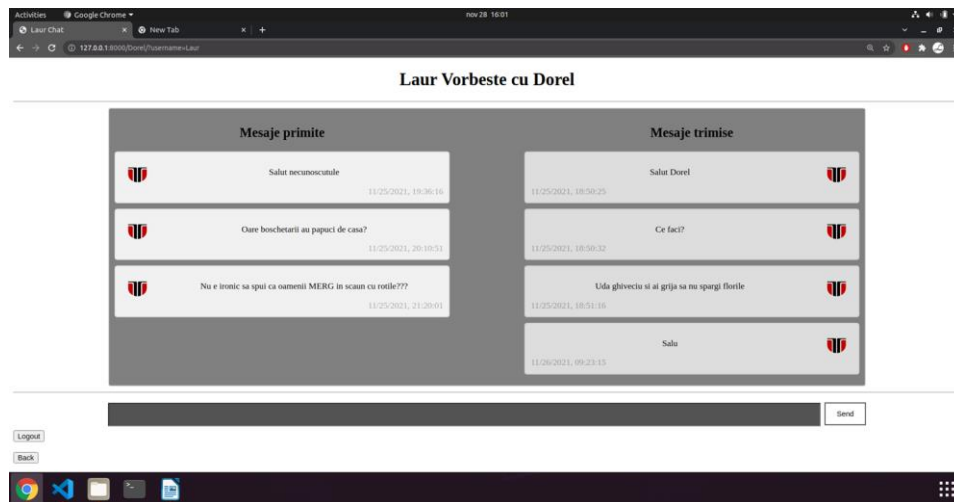
0  
logat



utilizatorul, acesta va fi dus pe pagina lui in care poate vedea si alege un user cu care sa vorbeasca.

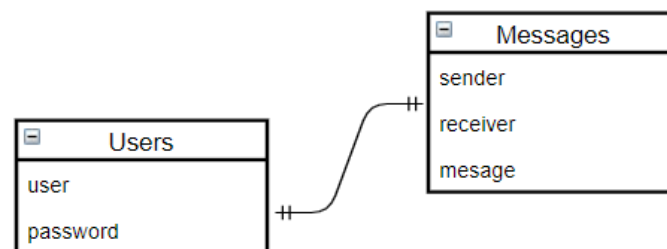


Dupa ce a ales userul cu care sa vorbeasca, acesta este dus pe pagina de chat in care poate vedea mesajele primite, trimise si ora. Tot aici poate trimite mesajele.



Pe partea de implementare am creat un model Messages format din sender, receiver, mesaj si data cand s-a creat mesajul. In fisierul views am implementat functionalitatile clare ale proiectului(logare, auth, logout, etc). Parsarea datelor dintre backend si frontend s-a facut folosind django templates si metode specifice din python care ia datele din front.

Baza de date este reprezentata in figura de mai jos.



## Probleme întâlnite și modul de rezolvare

Problemele intalnite au fost:

- transmiterea datelor intre back si font dar mi-am dat seama ca numele din django template trebuie sa fie acelasi cu cel declarat in python
- partea de creare a patr-urilor deoarece avem nevoie in path de utilizatorul logat si folosit in alte pagini dar am rezolvat scriind in urls.py `path('<str: receiver/ send>', views.send, name = 'send')`
- afisarea mesajelor trimise si primite in pagina. Am rezolvat creand doua functii (`getSentMessages` si `getReceivedMessages`) care ia datele din baza de date si returneaza un dictionar de forma 'mesaj: data' pe baza sender-ului si receiver-ului.
- sa imi dau seama cum functioneaza QuerySet-ul si cum iau datele in python din QuerySet.

## Concluzii

In concluzie, dezvoltarea aplicatiei a dus la aprofundarea termenilor din aplicatiile fullstack si invatarea si obisnuirea cu un nou framework. Mai mult de atat a aprofundat cunostiintele legate de limbajul de programare Python.