Department of Computing

Django ChatApp

Assignment 2

Student:

Clotan Andrei

Supervisor:

Prof. Andrei Bogdan Lecuta

December 7, 2022

Contents

Chapter 1

Introduction

1.1 Objectives

We will implement a Real-Time Chat application which will permit the users to log-in into their account and interact with each other by chatting.

1.2 Challenges

Developing this type of application will present a list of challenges, like:

- How to use python with Django framework
- How to integrate JavaScript.
- How to allow users to chat with each-other
- How to make the app display the data in real time.

Chapter 2

Realtime Chat

2.1 Technologies

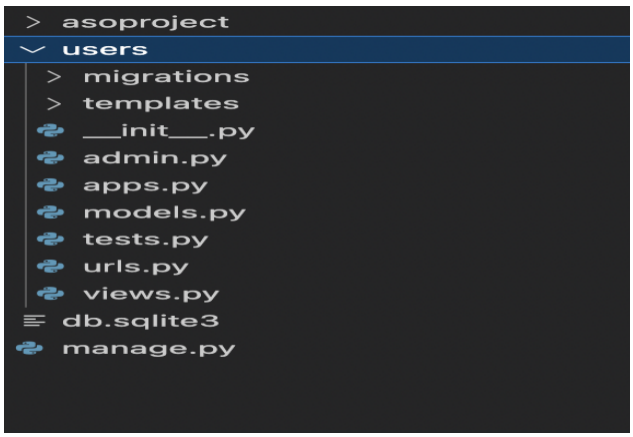
In this project I used:

- Python (Django framework) for the application core.
- Html, CSS, Bootstrap and JavaScript for View files.
- Sqlite3 for DataBase.

2.2 Project Structure

The project is structured in three applications:

- First, we have the Asoproject app (the python generated project), where we have the configuration for the entire project and where we connect external apps.
- We also have an users app, where we make our routes, and handle everything in our application.



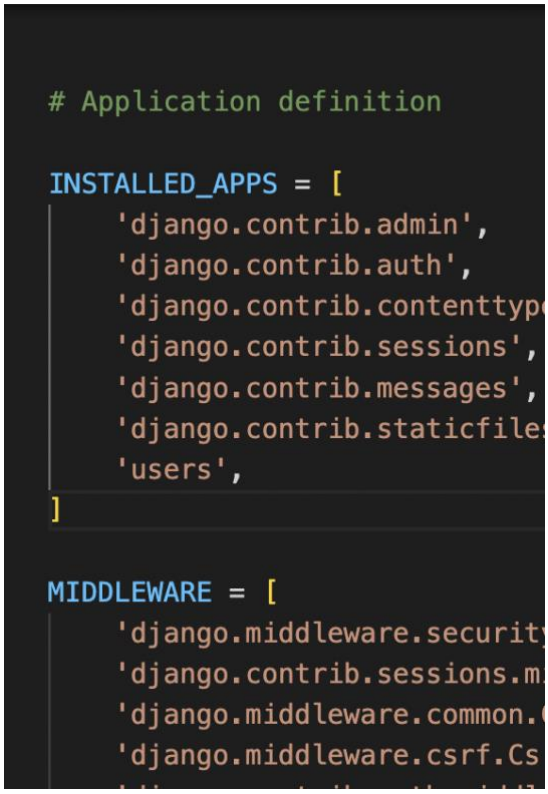
2.3 Database

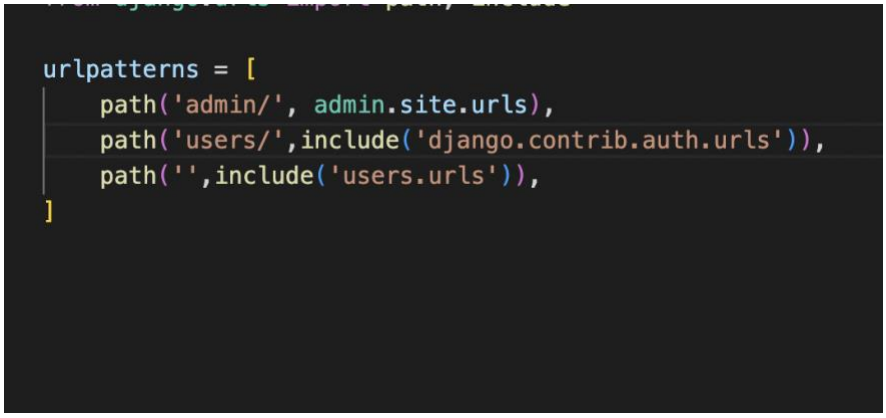
For this project, I chose to use three simple models:

- Users – the model autogenerated from Django admin. It has all the information about the users of the application
- Room – a simple model which has a unique room name. I used this model to create the "rooms" where 2 users talk to each other.
- Message – the most complex model. It stores data about the user, the room, the time it was created, and the room. The message is dependent of a room, and of a user.

2.4 Asoproject App

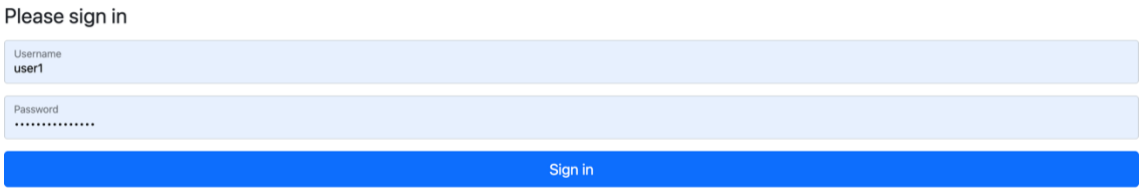
In the Asoproject App, I added configuration for the users app, which is the only external app I am using. Also, I had to add the paths of the project, inside the urls.py file.



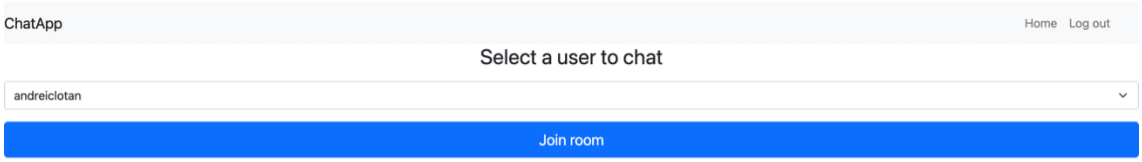


2.5 Users App

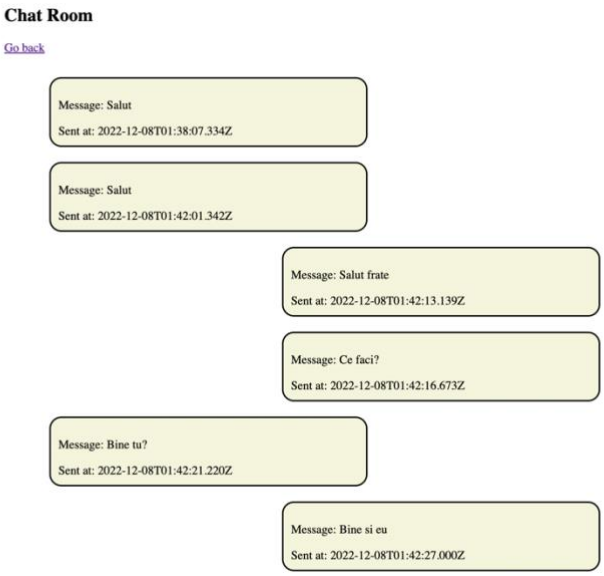
This is where everything happens.
Firstly, I added the models we need for this app: Room and Messages (users is already implemented).
Then, inside admin.py, I had to register the new models to the admin model.
Lastly, I had to implement the routes and the views, which represent the functionality of our app.



This is the homepage, where the user should sign in to use the rest of the application. After signing-in the homepage will change in the home format. For the login mechanism, I used the login method from Django.admin.



On the home page, the user can logout, go home, or select a user to chat with. The selection of user is based on a select picker, which has dynamic options, to show only the registered users. This was implemented by sending a parameter to the template and rendering options using javascript.



After the selection of the “chat-buddy”, the user is redirected to the Chat Interface, where he can see his older messages, and send new ones. He also has the option to go back to the home page.
In the right part is the current user, and in the left part is the user he is chatting to.
The room is created using the unique name of each user, but with the lower one, so if user1 wants to chat with user2, they will both get redirected to the user1 chat room.

Chapter 3

Conclusion

Concluding, our app is more than enough for a simple real time chat app, between users. It can be further improved, by adding images, letting people know when others are typing/other have seen their message.

Also, I understood how Django works, and now I feel more capable of developing a project using python.