

## Project **rockstore.com**

The main idea of this project I described in readme file, so, if you didn't see it, you should check it.

Here I would like to write and describe project structure and technologies which I used.

Let's get started from technologies.

### **Technologies which are used:**

- Python (3.10)
- Django (4.2.3)
- Postgresql (12)
- Django-channels(4)
- Django all-auth (using Google API)
- Docker

## **Project structure**

The project consists of 4 Django-apps:

- **Registration app** – responses for registration and login
- **User profile app** – This app is about user profile
- **Post app** – This app describes user's posts
- **Chat app** – responses for chat

Every app provides (Except registration\_app) provides CRUD operations  
=> Create, Read, Update, Delete.

### **Registration app**

It's a simple situation here.

The only thing it is, that at this project I use Google authentication  
So, the most of business logic is serviced by Google API.

There are only two views at this app:

- **HomePageView** – returns home page
- **LogoutView**

And also, I changed base templates which are provided by Google API. You can watch it in **registration\_app** folder at **templates** folder.

### **User Profile App**

After registration, the new user has an empty profile.

User's profile consists of the next data:

- Profile photo
- Country
- About

(It's **UserProfile model**, models.py)

There are next views:

- MyProfileView
- UserProfileView
- CreateProfileView
- UpdateProfileView
- DeleteProfileView

### **Post App**

The first – **models**.

- **Category**
- **Post** – describes **PostObject**:
  - User (in other words – creator or post owner)
  - title
  - item\_photo
  - brand
  - item\_model
  - date\_of\_manufacture
  - category
  - price

- possibility\_of\_exchange – means if this item could be bartered on something else.
- description
- liked – how many likes it has
- reposts – the same thing as likes but only reposts
- active – if this posts is available for buying or barterers, or it has been already changed.
- date\_created

## **Comment**

- User (who left the comment)
- Post (under what post comment was left)
- Description

## **Like**

- User
- Post
- Value

## **Repost**

- User
- Post

## The second – **Views:**

There are base views for **creating** posts, comments, likes and reposts, **deleting** them and **updating**, so I am not going to list them.

There are different views for showing posts:

- MyPostsListView – request.user posts
- UserPostsView

- PostView – show post details
- CategoryPostListView – filter posts by category

For comments:

- PostCommentsView
- MyCommentsView
- UserCommentsView

And for reposts:

- MyRepostsView
- UserRepostsView

There is only one view for likes – LikePostView.

## **Chat app**

Models

### **ChatRoom:**

- creator
- member

### **Message:**

- chatroom
- author
- content
- date\_created

And There are 4 basic views:

- CreateChatRoomView
- ChatRoomView
- MyChatroomListView
- DeleteChatRoomView