

Stock Trading System – Implementation

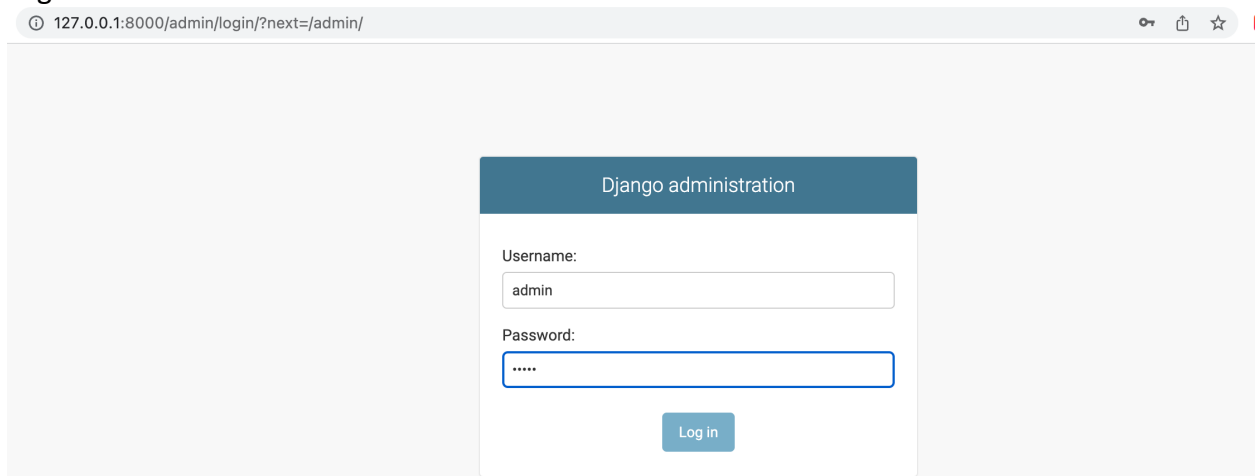
Prerequisites:

1. Python, Django installation.
<https://docs.djangoproject.com/en/4.0/topics/install/>
→ You can either install it in a virtual environment or in your local computer.
2. Creating a superuser.
→ <https://www.geeksforgeeks.org/how-to-create-superuser-in-django/>
→ A superuser is required to login into the admin console to create a group for admin and a user as admin as well.

```
PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL

(stockmanagement) jeffenzio@Jeffreys-MacBook-Air stockmanagement % python manage.py createsuperuser
Username (leave blank to use 'jeffenzio'): admin
Email address: admin@gmail.com
Password:
Password (again):
The password is too similar to the username.
This password is too short. It must contain at least 8 characters.
This password is too common.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
(stockmanagement) jeffenzio@Jeffreys-MacBook-Air stockmanagement %
```

3. Start the server → `python manage.py runserver`
 - 3.1. Create an user -> `stock_admin` using the registration form:
http://127.0.0.1:8000/register_user
 - 3.2. He will act as the admin.
4. Login to the admin console.



127.0.0.1:8000/admin/login/?next=/admin/

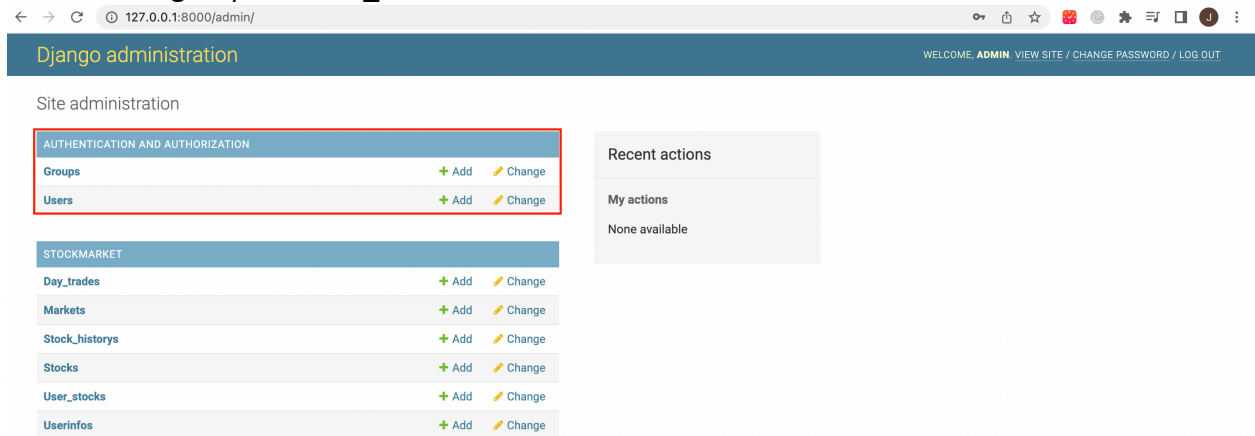
Django administration

Username:
admin

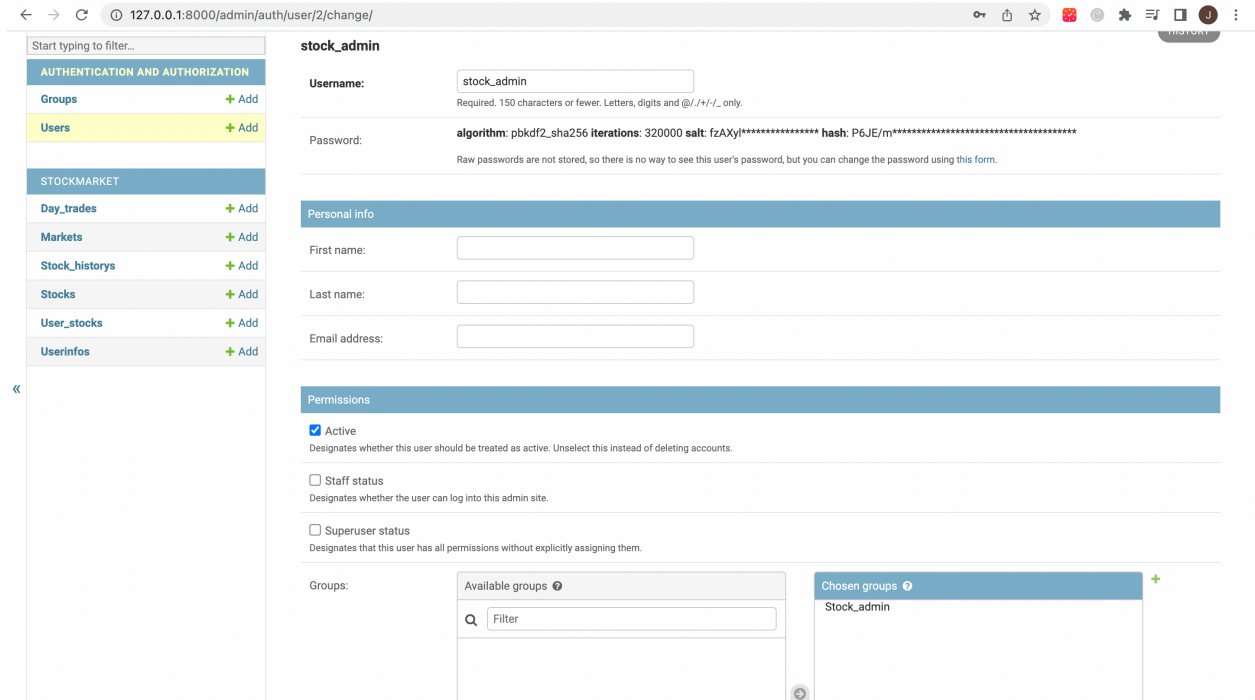
Password:
.....

Log in

4.1. Create a group – “Stock_admin” .



4.2. Adding the admin user (named as stock_admin) into the group as well.



5. Provide the initial Market Schedule by clicking on Markets and adding a record:

The screenshot shows the Django administration interface for the 'stockmarket' app. The left sidebar contains a navigation menu with the following items: 'Groups' (with a '+ Add' link), 'Users' (with a '+ Add' link), 'STOCKMARKET' (highlighted), 'Day_trades' (with a '+ Add' link), 'Markets' (with a '+ Add' link, highlighted in yellow), 'Stock_histories' (with a '+ Add' link), 'Stocks' (with a '+ Add' link), 'User_stocks' (with a '+ Add' link), and 'Userinfos' (with a '+ Add' link). The main content area is titled 'Add market' and contains three form fields: 'Market start time' (set to '06:00:00' with a 'Now' button and a note 'Note: You are 4 hours behind server time.'), 'Market end time' (set to '18:00:00' with a 'Now' button and a note 'Note: You are 4 hours behind server time.'), and 'Market schedule' (a dropdown menu set to 'ALL_DAYS'). At the bottom right of the form are three buttons: 'Save and add another', 'Save and continue editing', and 'SAVE'.

Implementation:

1. Admin Roles:
 - Modifying the market schedule.
 - Creating stocks
2. User Roles:
 - Buy Stock using Cash.
 - Buy Stock using Limit Order.
 - Sell Stock using Cash.
 - Sell Stock using Limit Order.
 - List all transactions.
 - See Limit Order pending history and delete them if required.
 - Look at his stocks and his bank limit.
3. Random Price Generator:
 - Modify the prices using random function.
 - If a stock follows the Limit Order condition set by any user, it will automatically be bought or sold.
 - It needs to be added to crontab (to run every 5 minutes):

```
5 * * * *
```

```
/opt/homebrew/Caskroom/miniforge/base/envs/stockmanagement/bin/python3  
/Users/jeffenzio/LocalDocuments/Documents/Education/Grad/Projects/StockManagem  
ent/stockmanagement/manage.py randomPriceGen
```

Deployment Guide:

1. If the pre-requisites are completed, the application can be started and used.