

TECHNICAL DOCUMENTATION- IPO Web APP and REST API for mobile responsiveness

INTRODUCTION : The project is made for both web and app based for the users to manage their IPOs in the Bluestock IPO Dashboard and IPO Dashboard and can observe the number of entries made in the Home page in the analytics overview portion to get a clear understanding of the IPOs by the graphical representation.

INSTALLATION :

- 1) Install all the dependencies that have been provided in the other files of the same zip file.
- 2) Install the Docker Desktop for running it in background which acts as a light weight container for web and database. It packages all application and its dependencies to ensure consistent performance in any environment.
- 3) Ensure IDE-VS Code.

STARTING THE WEB APPLICATION :

- 1) Run in the terminal of the project file- “docker compose build” (to explicitly build or rebuild the Docker images defined within the docker-compose.yml file)
- 2.) Run again in the same terminal- “docker compose up” or “docker compose up-d”(check the docker for web and database to start running as web-1 and db-1)

2) For any technical problem run in the terminal- “docker compose down”(to stop the container and the system) then run again “docker compose up” or “docker compose up-d” or run “docker compose restart”

3.) If any changes made in the models then run first in a new terminal “docker-compose exec web python manage.py makemigrations” (it will show the files that will migrate).

Then run “docker-compose exec web python manage.py migrate” (it will migrate all the files)

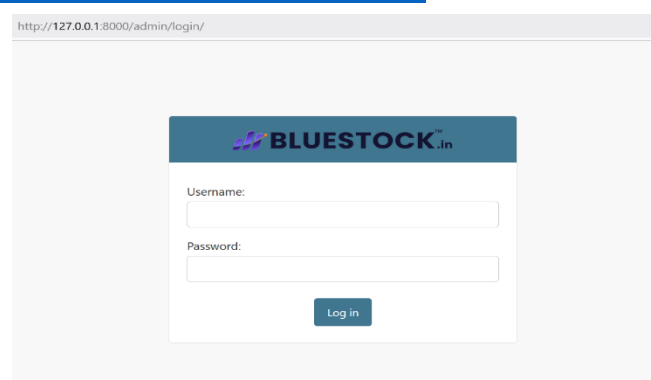
4.) If any changes in the staticfiles then run in the same terminal “docker-compose exec web python manage.py collectstatic –noinput”

5.) After all the changes are made then repeat the steps 1,2 to run the system and the containers again.

6.) For creating a new superuser run in a new terminal or in the same terminal ensuring the docker is running :

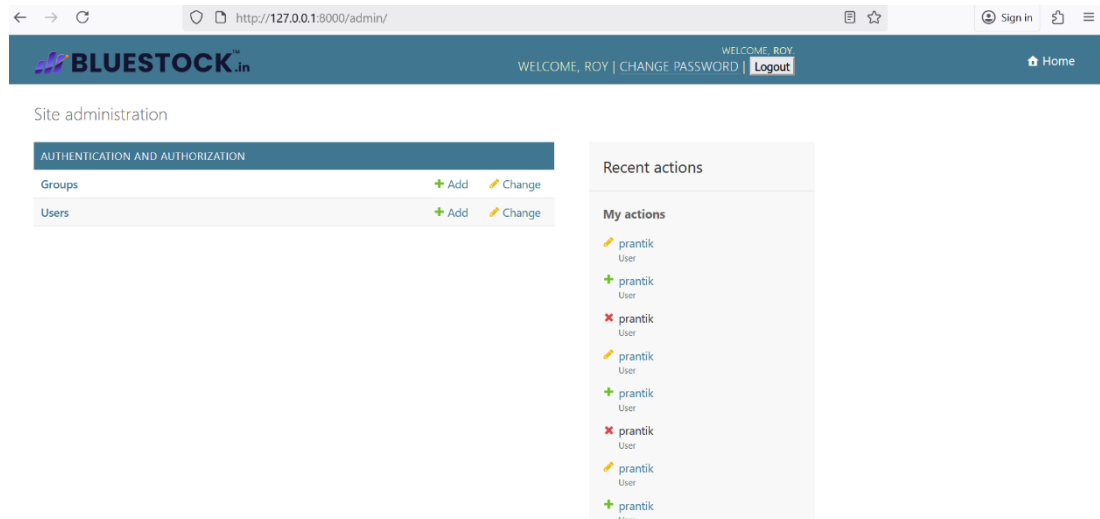
“docker-compose exec web python manage.py createsuperuser” [it will prompt for : email, username, password and confirm password]

RUNNING THE WEB APPLICATION : 1) Open in any browser : <http://127.0.0.1:8000/admin/login/>



A page like this will open (enter all the credentials created by the superuser).

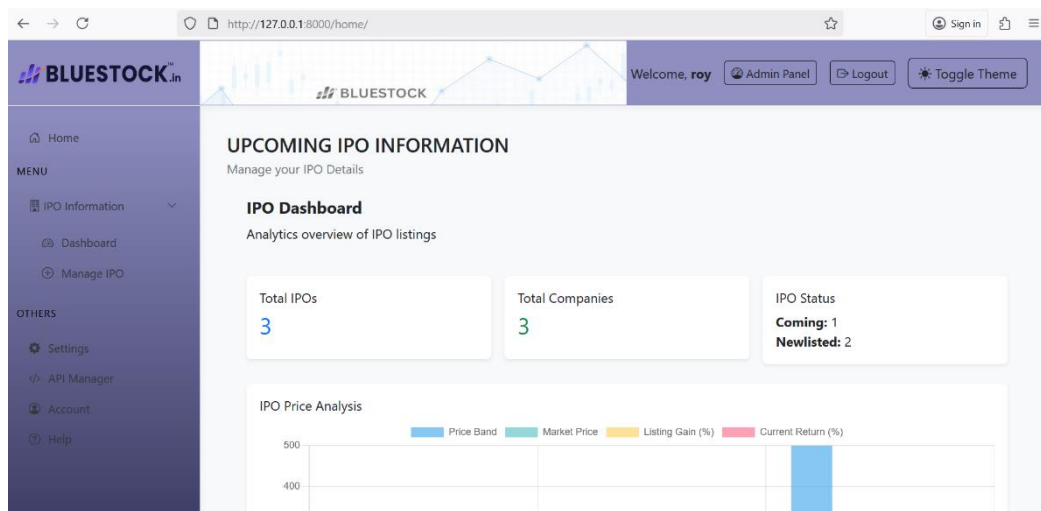
2) Login to the page a page will appear like this :



If we want to logout we can simply click logout from this page we will be logged out.

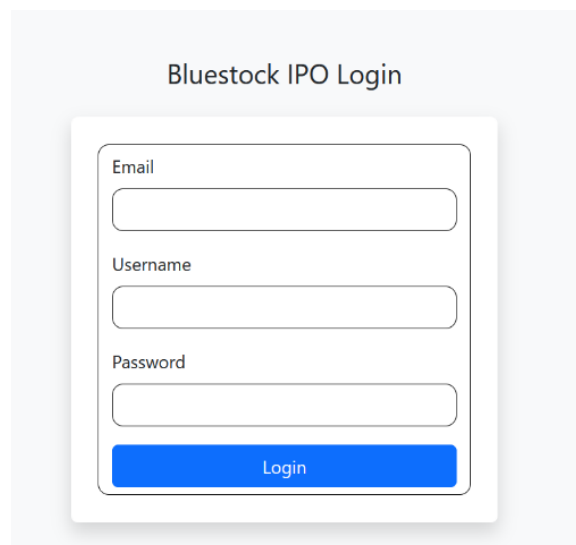
3) Now we can create any staff users and superuser from the Users section of the page.

4) Go to the Home where all the accesses for managing the IPOs and Dashboard were provided. It will appear like this:



5) Now we can select all the options what we want to observe like going back to Admin panel, toggling the theme to dark/white or can logout from just the IPO dashboard.

6) Logout from the IPO dashboard (not from the whole system):

A login form titled "Bluestock IPO Login" is centered on a light gray background. The form is a white rectangle with rounded corners and a subtle drop shadow. It contains three input fields: "Email", "Username", and "Password", each with a label above it. Below the input fields is a blue button with the text "Login" in white.

API TESTING IN POSTMAN : 1) Open POSTMAN and select new collection for http.

2) Paste the URL for IPO to perform GET and POST :

<http://localhost:8000/ipo/api/receive/>

3) Now select the files :

i) Headers- Key – Content-Type, Value- application/json

ii) Body – Select 'raw' and paste a raw data for example:

{

```
"company": 1,  
"status": "newlisted",  
}
```

4) We will get a message in the body as:

```
{  
  "message": "IPO saved successfully",  
  "ipo_id": 60  
}
```

5) Check the main folder we will receive a json file in the project directory.

CONCLUSION : This project ensures proper collection of IPO data in web application and stores all the data so that no data is lost even after edit or adding IPOs. This project allows a well defined UI for the users to run it wisely and easily.