

Project Title: IPO Management System- Bluestocks Fintech

Intern Name: Panchadip Bhattacharjee

Internship Organization: Bluestock Fintech

Duration: Nov 1, 2024- Dec 30, 2024

Date of Submission: Jan 5, 2025

Acknowledgments

I would like to express my gratitude to Bluestock Fintech for providing me the opportunity to work on this exciting project. My heartfelt thanks to the entire technical and managerial team at Bluestocks for their guidance and support throughout my internship. I also appreciate the constructive feedback and mentorship received, which helped me grow as a developer. Lastly, I would like to thank my family and friends for their constant encouragement during this journey.

Table of Contents

1. Introduction
 2. Project Objective
 3. Tools and Technologies Used
 4. System Design and Architecture
 5. Features of the System
 6. Implementation
 7. Testing and Debugging
 8. Results
 9. Conclusion
 10. References
-

Introduction

Bluestock Fintech is an innovative company dedicated to providing financial solutions with a focus on stock market applications. During this internship, I was assigned the task of developing an IPO Management System. The purpose of this project was to create a comprehensive web application and REST API that provides users with information about Initial Public Offerings (IPOs). The application facilitates subscription to IPOs, viewing IPO details, and managing user transactions related to IPO investments.

The main objective of this internship project was to gain hands-on experience in full-stack development, working with Django for the backend, PostgreSQL for the database, and using HTML, CSS, and JavaScript for the frontend.

Project Objective

The IPO Management System is designed to provide users with relevant IPO-related data, including but not limited to company names, logos, price bands, issue dates, issue sizes, and IPO-related documents like RHP and DRHP PDFs. Additionally, users can subscribe to IPOs and track their investment transactions. The key features of this application are:

- Viewing detailed IPO information
 - Managing user subscriptions to IPOs
 - Displaying transaction history and returns
 - Accessing downloadable IPO-related PDFs
 - A REST API for clients' websites and apps to fetch IPO data
-

Tools and Technologies Used

- **Programming Languages:** Python (Backend), HTML, CSS, JavaScript (Frontend)
 - **Frameworks:** Django (Backend), Bootstrap (Frontend)
 - **Database:** PostgreSQL, SQLite3
 - **API:** Django REST Framework for the API
 - **Version Control:** Git & GitHub
 - **Testing Tools:** Postman (for API testing)
 - **IDE:** Visual Studio Code
-

System Design and Architecture

The IPO Management System follows a **Model-View-Template (MVT)** architecture, where:

- **Model:** Represents the database schema for IPOs, transactions, user subscriptions, etc.
- **View:** Defines the frontend templates that render the data for the user interface.
- **Template:** Manages the communication between the model and view. In this case, templates such as HTML pages (Subscribe.html, ipos_list.html, etc.) will display the information fetched by the backend.

The system architecture also includes the use of REST APIs to fetch and manage IPO data, allowing external clients to integrate with our platform.

Features of the System

1. **IPO Listings:** Users can view a list of upcoming and ongoing IPOs, including detailed information such as company name, price band, issue date, status, and more.

2. **User Subscription:** Users can subscribe to an IPO, view their subscription details, and track the subscription status.
 3. **Transaction History:** Users can view a history of their IPO transactions, including IPO price, listing price, and listing gain.
 4. **Downloadable PDFs:** Users can download the Red Herring Prospectus (RHP) and Draft Red Herring Prospectus (DRHP) documents for each IPO.
 5. **API Integration:** A REST API to integrate IPO data into third-party platforms, such as clients' websites or applications.
-

Implementation

Backend Development:

1. **Setup the Django Project:** I started by setting up the Django project and configured the PostgreSQL database for managing IPO-related data.
2. **Models Implementation:**
 - Created models for IPO, Transactions, and User Subscriptions. Each model had fields corresponding to the IPO details (name, price band, dates, etc.) and user transaction details (subscription status, IPO price, listing price).
3. **API Development:** I developed a REST API using Django REST Framework to allow users to access IPO data programmatically. The API endpoints supported operations such as retrieving IPO listings, subscribing to an IPO, and getting transaction history.

Frontend Development:

1. **HTML, CSS, JavaScript:**
 - Designed clean and responsive web pages using HTML and CSS.
 - Implemented Bootstrap to ensure a mobile-friendly layout.
 - Developed functionality to dynamically display IPO listings and transaction data using JavaScript.
2. **Subscribe Page:** Created a user-friendly subscription page where users can select and subscribe to IPOs.

User Authentication:

Implemented basic user authentication using Django's built-in authentication system, allowing users to register, log in, and manage their profiles.

Testing and Debugging

I used several methods to test the system:

1. **Unit Testing:** Wrote test cases for backend logic, especially the models and API views, to ensure correctness and integrity of the data.

2. **Integration Testing:** Tested the integration of frontend with backend APIs to ensure seamless communication.
3. **Manual Testing:** Tested the web application in various browsers and devices to ensure compatibility and responsiveness.

Some common bugs faced included issues with rendering the dynamic data on the frontend, which was resolved by improving the backend API and ensuring correct data format handling.

Results

The final product was a fully functional IPO Management System with the following features:

- Display of IPO details, including downloadable PDFs
- User authentication for subscription management
- An API that integrates IPO data into external systems
- A clean, responsive UI for ease of use on all devices

The project was successfully deployed and tested, and the application works as expected.

Conclusion

This internship project allowed me to gain valuable practical experience in full-stack development. I learned how to integrate various technologies like Django, PostgreSQL, Bootstrap, and JavaScript to build a robust web application. The project also improved my problem-solving skills, particularly in API development and testing.

In the future, I would recommend adding more advanced features such as a payment gateway for subscriptions and more detailed reporting for transaction history.