Project Document

Workspace Login URL : https://bluestock.in/backoffice-software/

Login Details : You will receive over mail

Work Policy: https://bluestock.in/hr/work-policy.html

Team Structure

ID	Name	Mail	Мо	Role
11	Saivivek Errabelli	errabellisaivivek@gmai l.com	8341319267	Team Lead
12	Aayush Shrikant Athawale	athawaleaayush@gmail. com	9307099927	Team Lead
13	Nithin Jeripothula	nithin05j@gmail.com	7416457672	Co-Team Lead
14	Keerthana M G	keerthanamg19@gmail.	9445087093	Co-Team Lead
15	Aditya Singh Bais	adityasbsinghbais7@g mail.com	8815727271	Dev
16	Adesh Gaikwad	adeshg1508@gmail.co m	8766539268	Dev
17	Vishwas Patil K S	vishwashivu567@gmail .com	8217511855	Dev
18	Annamdasu Geetha rani	geetaraniannamdasu@g mail.com	7995163839	Dev
19	manasa hiremath	m33849595@gmail.co m	7676345510	Dev
20	Shaikh Alisha Siraj	alishashaikh186555@g mail.com	9666186555	Dev

Project: IPO Web Application & REST API Development

Company: Bluestock Fintech Website: www.bluestock.in Mail Id: hello@bluestock.in **WA Contact**:7038202440

Date: 01/04/2025

Department: #BF-SOFT-DEV

Project Manager - Yash Kale **Mail id** - yashkale@bluestock.in

WA No: 9209550273

Important :-

Check UI UX Design-https://www.figma.com/design/ Qz0zqYktZHH1iQLHNFg8rE/UI-UX-Design---IPO-Web-App?node-id=0-1&t=SFaWLUo9w2eNXt1z-1

Check System Design: - https://www.figma.com/board/
https://www.figma.com/board/
g9bjreevYNJkfMuwRacyaP/System-Design?t=rhom703DRI5pdHkG-1

Assets (Logo,SVG, PNG Etc: https://drive.google.com/drive/folders/1yH9Y mlgqEkZXtzhgHSuFtEwFOr8BXH5?usp=drive link

Task Management for (Team lead & Co Team Lead):

- Use Notion for assigning and tracking tasks within the team.
- Developers must report daily work status to the Team Leader.
- Developers also can change their role by taking permission of team leader
- The Team Leader should report work progress every 7 days to Mr. Yash Kale.
- Write high quality industry standard code.
- Take regularly google meeting with team members

GitHub Repo Link: [Clone this repo and start working on it] Git Link - (Check Mail) –

If Production repo access not granted, Team Lead must create private repository for entire team member and give access to all team members.

Introduction

This report outlines the production-level software development tasks assigned to our interns at Bluestock Fintech. The project involves developing an IPO web application and REST API for the Bluestock website/app and our clients' websites/apps. These tasks are designed to enhance the interns' practical knowledge and contribute to our ongoing projects. Each task aligns with our company's goals and the interns' skills and career aspirations.

Objective:

Develop a web application and REST API that provides IPO-related information to the public. The application will display information such as company logo, name, price band, opening and closing dates, issue size, issue type, listing date, status, IPO price, listing price, listing gain, current market price (CMP), and current return. Additionally, it will include downloadable RHP and DRHP PDFs.

Software Requirements

Backend:

- SDK: Python (Version 3.12.3)
- Framework: Django (Version 5.0.6) `pip install Django`
- API: Django REST Framework (Version 3.15.1) `pip install djangorestframework`
- Tools: Postman for API testing, Git & GitHub for version control

Frontend:

- Technologies: HTML, CSS, plain JavaScript (no NodeJS)
- Framework: Bootstrap 5 (via CDN link)
- Database: PostgreSQL
- IDE: Visual Studio Code (VS Code)

1. Prerequisites

Interns are required to familiarize themselves with the following concepts before starting the project:

What is Stock Market? - https://youtu.be/gWrR5qPEmWE?si=QAf19b_i-uSMn3vV

Project Features

The IPO Web Application will provide the following information to the public:

- Company Logo
- Company Name

- Price Band
- Open Date
- Close Date
- Issue Size
- Issue Type
- Listing Date
- Status
- IPO Price
- Listing Price
- Listing Gain
- Current Market Price (CMP)
- Current Return
- RHP PDF
- DRHP PDF

This information will be available on both the Bluestock website/app and clients' websites/apps.

Development Overview

Important Notes:-

- This is a production-level project; therefore, write quality code and test features regularly.
- Apply necessary security rules to ensure data protection and integrity.
- Utilize ChatGPT for testing code, writing features, and designing the database schema.

1. References

https://youtu.be/s7aINQPGNDM?si=ydfab7GMUaMCg91W

https://www.youtube.com/watch?v=C1NgOmoOszc&list=PLjVLYmrlmjGcyt3m6rt21nfjhYSWP_Ue_

https://youtu.be/5EY6JFptZgw?si=nwZ5DE7nVc4TXs8X

9.Deadlines

Date	Work Status
-	Complete All Frontend (Client Side, Admin Dashboard)
-	Complete Entire Project

//Deadline for entire project submission updated in workspace

Note: The timeline is flexible. Discuss with your team leader to determine a suitable schedule for all developers.

Submission Guidelines

- Do not commit buggy or error-prone code.
- Thoroughly test your code on your local machine before committing.
- Commit changes to the main branch and push them.

10. Resources for This Project Assets

(Logo,SVG, PNG Etc):

https://drive.google.com/drive/folders/

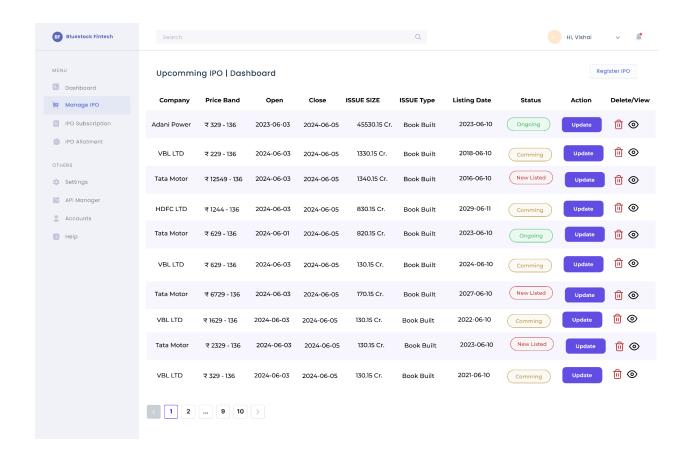
11Q6kapopPp4tVn82QKRCX8cpHYHO5GCK?usp=drive_link

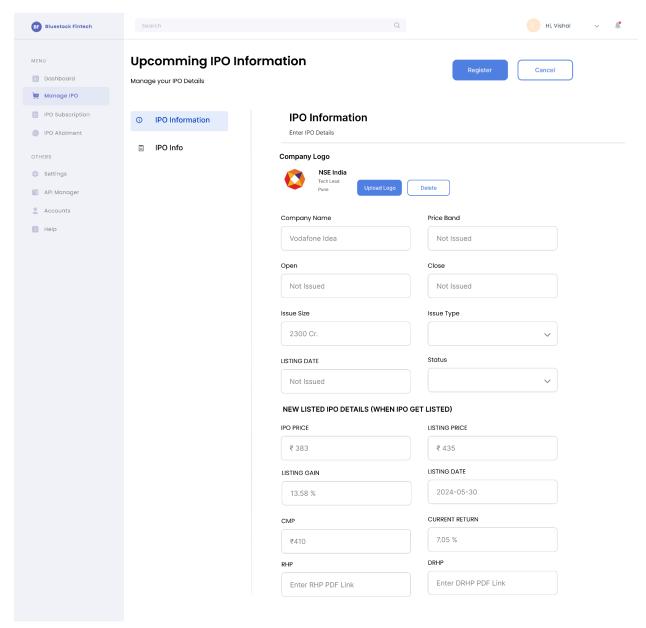
(Request Access)

Figma UI/UX Prototype:

Please visit link for complete details -

https://www.figma.com/design/Qz0zqYktZHH1iQLHNFg8rE/UI-UX-Design---IPO-Web-App?node-id=0-1&t=SFaWLUo9w2eNXt1z-1

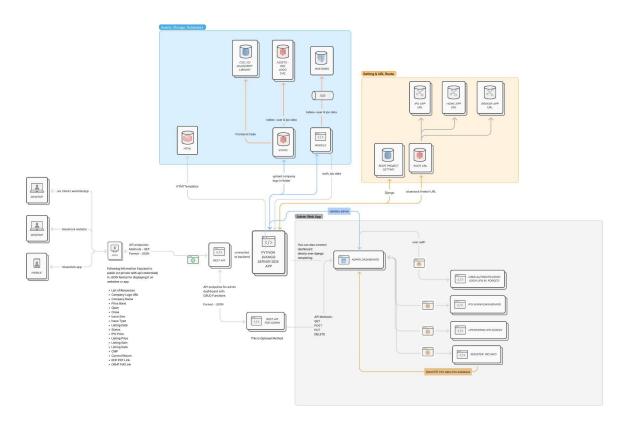




System Design:

View More: -

https://www.figma.com/board/g9bjreevYNJkfMuwRacyaP/System-Design?node-id=0-1&p=f



System Design: IPO Web Application & REST API By Bluestock Fintech

Database Diagram:

User table already created Username- admin Password – admin

ER Diagram Description:

Entities and Attributes:

- 1. Companies
 - company_id: INT (Primary Key, Auto-Increment)
 - company_name: VARCHAR(255) (Not Null)
 - o company_logo: VARCHAR(255) (URL or file path)
- 2. **IPOs**
 - o **ipo_id**: INT (Primary Key, Auto-Increment)
 - company_id: INT (Foreign Key referencing Companies.company_id)

- price_band: VARCHAR(50)
- open_date: DATE
- close_date: DATE
- o issue_size: VARCHAR(100)
- issue_type: VARCHAR(50)
- o listing date: DATE
- status: ENUM('Upcoming', 'Open', 'Closed', 'Listed')
- o ipo price: DECIMAL(10,2)
- o listing_price: DECIMAL(10,2)
- o listing_gain: DECIMAL(5,2)
- current_market_price: DECIMAL(10,2)
- current_return: DECIMAL(5,2)

3. Documents

- document_id: INT (Primary Key, Auto-Increment)
- o **ipo_id**: INT (Foreign Key referencing IPOs.ipo_id)
- rhp_pdf: VARCHAR(255) (URL or file path)
- drhp_pdf: VARCHAR(255) (URL or file path)

Relationships:

1. Companies to IPOs

- Type: One-to-Many
- Description: One company can have multiple IPOs, but each IPO is associated with exactly one company.
- Foreign Key: company_id in IPOs references company_id in Companies.
- On Delete: CASCADE (if a company is deleted, all related IPOs are deleted).

2. IPOs to Documents

- o **Type**: One-to-Many
- Description: One IPO can have multiple documents, but each document is associated with exactly one IPO.
- Foreign Key: ipo_id in Documents references ipo_id in IPOs.
- On Delete: CASCADE (if an IPO is deleted, all related documents are deleted).

```
[Companies] --(1:N)-- [IPOs] --(1:N)-- [Documents]
  | company_id (PK)
                           | ipo_id (PK)
                                              | document_id (PK)
  | company_name
                         | company_id (FK)
                                              | ipo_id (FK)
  | company_logo
                         | price_band
                                              | rhp_pdf
                         | open_date
                                               drhp_pdf
                         | close_date
                         | issue_size
                         | issue_type
                         | listing_date
                         | status
                         | ipo_price
                         | listing_price
                         | listing_gain
                         | current_market_price |
                         | current_return
```

User Flow:

N/A