Name: Sanskar Kumar Email Id: kumarsanskar01@gmail.com

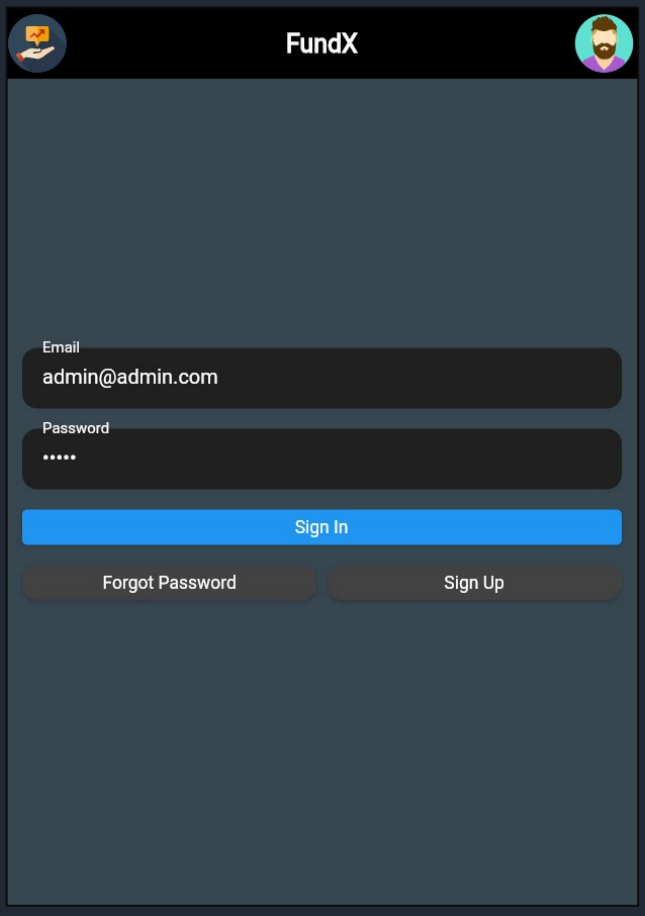
Product Requirements Document (PRD)

1. **Introduction:**
   Product Name: Fundx(**GROWW** like Stock Market App)
   Overview: The Stock Market App allows users to sign up, sign in, and view live stock prices. The app fetches live stock data from the Finnhub API and enables users to save and track their preferred stocks.

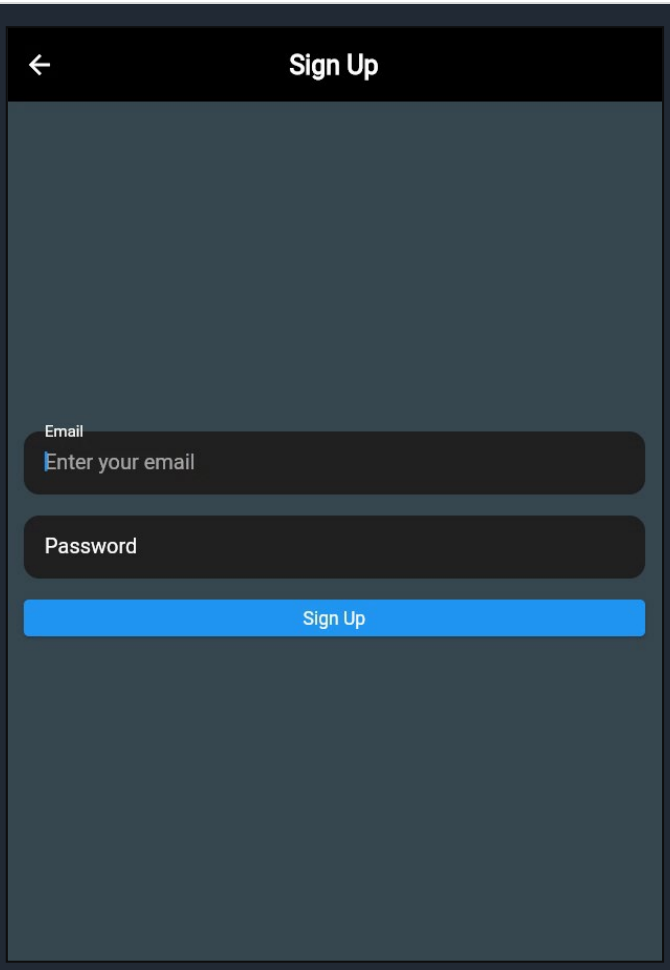
2. **Objectives:**
   - Provide real-time stock price information.
   - Allow users to create accounts and manage their stock preferences.
   - Ensure secure user authentication and data storage.

3. **Key Features:**
   a. **User Authentication:**
      - Sign up with a username and password.
      - Sign in with a username and password.
      - JWT token-based authentication for secure endpoints.

   b. **Stock Data Management:**
      - Fetch and display real-time stock prices.
      - Allow users to save their preferred stocks.
      - Fetch saved stocks for the authenticated user.



c. **Database Management:**
   - Store user information securely.
   - Maintain user-specific stock preferences.



4. User Stories

- **Sign Up:**
  As a user, I want to create an account with a username and password so that I can access personalized features.

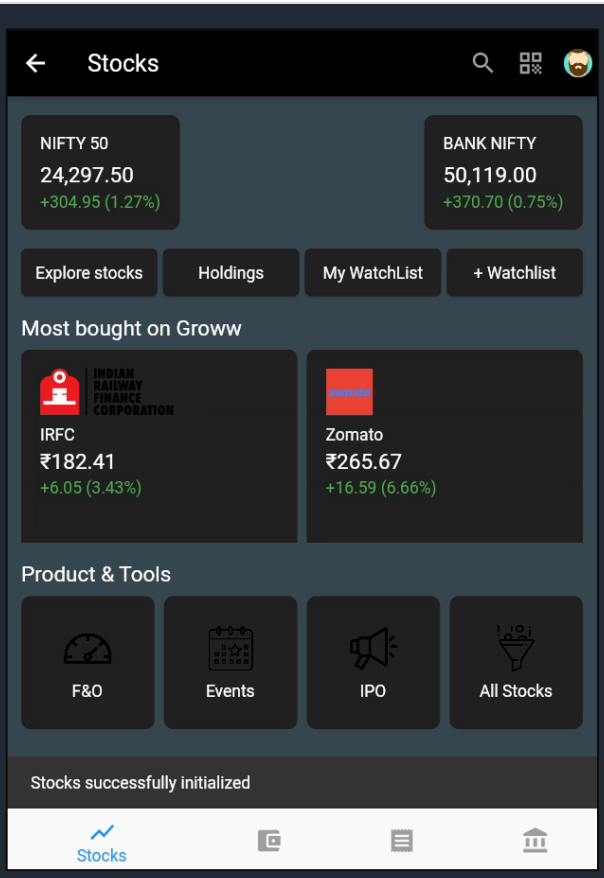
- **Sign In:**
  As a user, I want to sign in with my username and password so that I can access my saved stocks.

- **View Stock Prices:**
  As a user, I want to view real-time stock prices so that I can stay updated with the market.

- **Save Preferred Stocks:**
  As a user, I want to save my preferred stocks so that I can easily track them.

- **Fetch Saved Stocks:**
  As a user, I want to view my saved stocks so that I can quickly access the stocks I'm interested in.

Name: Sanskar Kumar Email Id: kumarsanskar01@gmail.com



5. Technical Requirements

Frontend:

- Framework: Flutter, Dart
- API Integration: Fetch stock data from the backend
- User Authentication: Handle sign-up and sign-in functionality using JWT
- Stock Display: Display real-time stock prices and user-specific stocks

Backend:

- Framework: Express.js with TypeScript
- Database: PostgreSQL
- Authentication: JWT token-based authentication
- API Integration: Finnhub API for stock data
- Endpoints:
  - POST /signup: User registration
  - POST /signin: User login
  - GET /test-db: Test database connection
  - POST /init-stocks: Initialize stock symbols for the user
  - GET /stocks: Fetch stock data for the authenticated user
  - GET /stock/:symbol: Fetch stock data for a specific symbol
  - GET /protected: Protected route for authenticated users

Database Schema:

- **Users Table:**

```
CREATE TABLE users (
  id SERIAL PRIMARY KEY,
  username VARCHAR(255) UNIQUE NOT NULL,
  password VARCHAR(255) NOT NULL
);
```

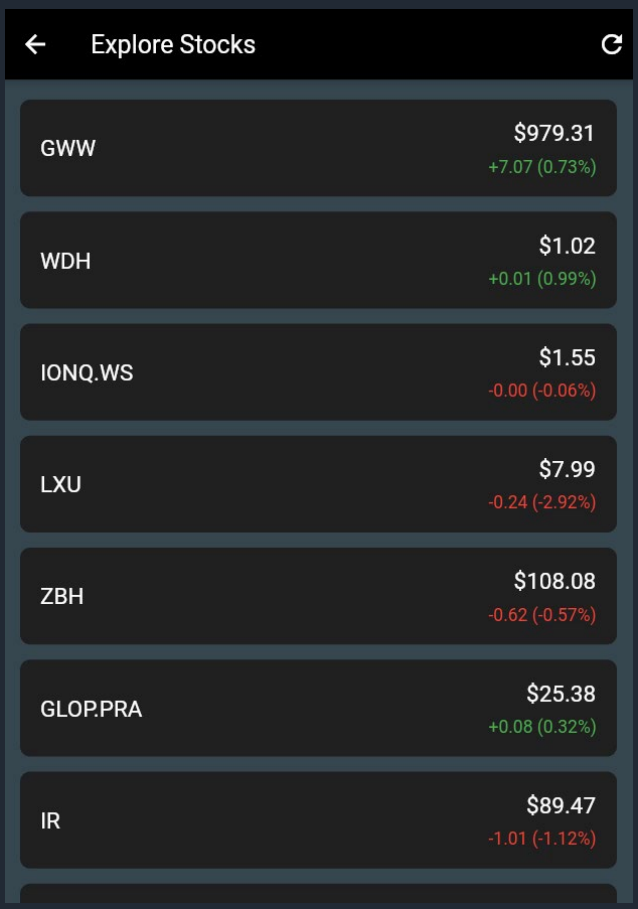
- **User Stocks Table:**

```
CREATE TABLE user_stocks (
  user_id INTEGER NOT NULL REFERENCES users(id),
  symbol VARCHAR(10) NOT NULL,
  name VARCHAR(255),
  PRIMARY KEY (user_id, symbol)
);
```

Name: Sanskar Kumar Email Id: kumarsanskar01@gmail.com

6. Non-functional Requirements

- **Security:** Ensure secure storage of passwords using hashing (bcrypt). Protect API endpoints with JWT tokens.
- **Performance:** Ensure real-time fetching of stock data without significant delays.
- **Scalability:** Design the system to handle a growing number of users and stock data.

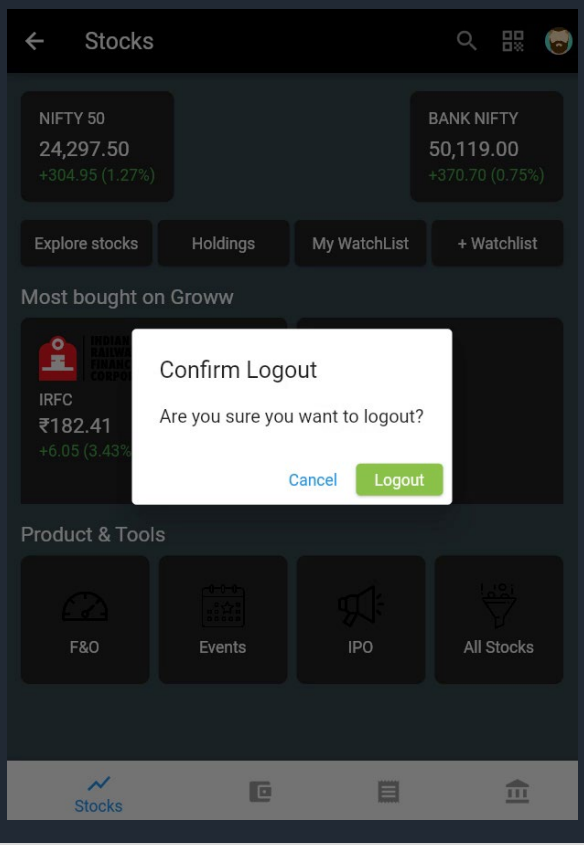


7. Assumptions

- Users have internet access to fetch real-time stock data.
- The Finnhub API provides the necessary stock data as per the documented API responses.
- The database is set up and maintained for secure storage and retrieval of user data.

8. Constraints:

- Rate limits imposed by the Finnhub API.
- Potential latency in fetching real-time stock data from the API.



9. Risks:

- API rate limits might affect the frequency of data fetching.
- Security breaches if JWT tokens are not managed securely.

10. Glossary:

- **JWT (JSON Web Token):** A compact, URL-safe means of representing claims to be transferred between two parties.
- **API (Application Programming Interface):** A set of rules that allows one piece of software to interact with another.
- **Finnhub API:** A stock market API used to fetch real-time stock data.