**Technical Documentation for the Project**

**1. Project Documentation**

**Overview**

This project is designed to provide a platform for real estate crowdfunding by delivering core functionalities such as managing investments, tracking tokens, and exploring promotions. The goal is to provide users with an intuitive and secure interface for exploring and managing their investment portfolio.

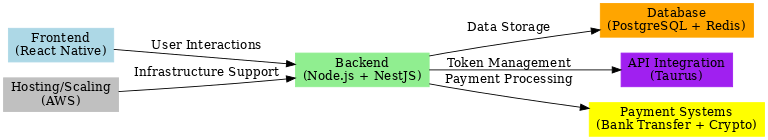
**Codebase Details**

* **Programming Languages**: TypeScript
* **Frontend Framework**: React Native (Expo)
* **Backend Framework**: Node.js with Express
* **Database**: PostgreSQL
* **Additional Libraries**: Axios (API calls), Redux Toolkit (state management), React Navigation (navigation), JWT (authentication), dotenv (environment variables management).

**Architecture**

The project follows a client-server architecture:

1. **Frontend (React Native)**:
   * Handles user interactions and renders UI components.
   * Communicates with the backend via RESTful APIs.
2. **Backend (Node.js/Express)**:
   * Processes requests from the frontend.
   * Interfaces with the PostgreSQL database for data persistence.
   * Implements business logic and authentication.
3. **Database (PostgreSQL)**:
   * Stores all application data including users, tokens, investments, and promotions.



**2. Environment Setup Guide**

**Setup Instructions**

1. Clone the repository:

git clone [repository-url]

cd [project-directory]

1. Install dependencies:

npm install

1. Configure environment variables by creating a .env file in the project root:

DB\_HOST=localhost

DB\_PORT=5432

DB\_USER=[your-database-user]

DB\_PASSWORD=[your-database-password]

JWT\_SECRET=[your-jwt-secret]

API\_URL=[base-url-for-api]

1. Run the development server:

npm start

**Development Environment**

* **IDE**: Visual Studio Code
* **Plugins**:
  + ESLint (linting)
  + Prettier (code formatting)

**Test Environment**

1. Install test dependencies:

npm install --save-dev jest supertest

1. Run tests:

npm test

**3. Configuration Management**

**Configuration Files**

* **.env**: Stores sensitive environment variables.
* **config.json**: Contains configuration for different environments (development, staging, production).

**Environment Variables**

|  |  |
| --- | --- |
| **Variable** | **Description** |
| DB\_HOST | Hostname for the database server |
| DB\_PORT | Port number for the database |
| JWT\_SECRET | Secret key for JWT authentication |
| API\_URL | Base URL for the API |

**Credentials**

* Use a .env file for storing credentials securely.
* Ensure the .env file is added to .gitignore to prevent it from being committed to the repository.

**4. Code Review Readiness**

**Known Issues**

* [Issue 1]: [Description, e.g., "Token prices are not updating in real-time."]
* [Issue 2]: [Description, e.g., "Dashboard transitions are occasionally slow."]

**Future Improvements**

* Implement direct payment gateway integration for enhanced user experience.
* Optimize API response times for the secondary market module.
* Enhance animations to align better with the steampunk aesthetic.

**Guidelines**

* Review the backend/routes/ folder to ensure RESTful conventions are followed.
* Focus on code structure and clarity in frontend/components/.
* Check for proper error handling in API calls (backend/middleware/errorHandler.js).
* Test the security of sensitive operations such as login and token trading.