INSTRUCTIONS

To set up and run the application, follow these instructions:

Environment Setup:

Node.js Installation: Ensure Node.js is installed on your system. You can download and install it from the official Node.js website.

Database Setup: Set up a PostgreSQL database. You'll need to have PostgreSQL installed on your system or use a cloud-based PostgreSQL service like Heroku Postgres. Create a new database and note down the connection details (host, port, database name, username, password).

Clone Repository: Clone the project repository from https://github.com/varunyadav1175/FinanceManagementSystem

Install Dependencies: Navigate to the project directory in your terminal and run the following command to install the required dependencies:

npm install

Environment Variables: edit the .env file in the project root directory and configure the following environment variables:

PORT=3000
DB_HOST=<your_database_host>
DB_PORT=<your_database_port>
DB_NAME=<your_database_name>
DB_USER=<your_database_username>
DB_PASSWORD=<your_database_password>
JWT_SECRET=<your_jwt_secret_key>

Running the Application:

Database Migration:

Firstly head over to config/config.js and edit the following to your personal credentials:

```
"development": {
    "username": "postgres",
    "password": "root",
    "database": "postgres",
```

```
    "host": "127.0.0.1",
    "dialect": "postgres"
  }
```

Run the database migration to create the necessary tables. This will create the necessary tables and columns in your postgres Execute the following command in your terminal:

npm run migrate

Start the Server: Start the Node.js server by running the following command:

npm start

Accessing the API: Once the server is running, you can access the API endpoints using tools like Postman or curl. The base URL for the API endpoints will be http://localhost:3000/.

API Testing: You can test the API endpoints manually using tools like Postman.

Application Usage:

1. User Authentication: Use the /auth/signup endpoint to create a new user account. Then, use the /auth/login endpoint to obtain a JWT token for authentication.

2. Transaction Management: Use the /transactions endpoints to manage financial transactions. You can add, retrieve, update, and delete transactions using these endpoints.

3. Secure Access: Ensure that you include the JWT token obtained after authentication in the Authorization header for protected routes. This token will grant access to authorized users only.

Follow these instructions to set up, run, and test the application successfully. If you encounter any issues, contact me for assistance

Varun Yadav (+91-9460114093)
varunyadav1175@gmail.com