

Documentation for clone this proj

Step to set up and run the server

1. Clone the Repository

```
git clone https://github.com/rupam241/expense_tracker  
cd backend
```

2. Install Dependencies

Run the following command to install all the required packages:

```
npm install
```

3. Start the Server

To start the server in development mode:

```
nodemon server
```

4. Environment Variables

Create a .env file in the root directory and add the necessary environment variables:

```
PORT=3
```

000

DATABASE_URL=YOUR DATABASE URL

JWT_SECRET=YOUR SECRET KEY

5. Access the Server

Once the server is running, you can access it at:

<http://localhost:3000>

Step to set up and run the client

1. Clone the Repository

Use the upper link to copy the repo

cd frontend

2. Install Dependencies

Install all the required packages:

npm install

3. Start the Frontend Server

To start the frontend development server:

☑ 5. Access the Application

Once the server is running, kindly access the application in your browser:

<http://localhost:5173>

Database Setup

Follow these steps to set up the **PostgreSQL** database using **Prisma**.

1. Install PostgreSQL

Make sure **PostgreSQL** installed on your machine.

Download and install it from: <https://www.postgresql.org/download/>

📁 2. Create a PostgreSQL Database

1. Open your **PostgreSQL** client (e.g., psql, **pgAdmin**, or **DBeaver**).
2. Run the following command to create a new database:

```
CREATE DATABASE your_database_name;
```

3. Configure Prisma

1. Go to the .env file in project's root directory.
2. Update the DATABASE_URL to point to your PostgreSQL database:

```
DATABASE_URL="postgresql://username:password@localhost:5432/your_database_name"
```

Replace:

- username with your PostgreSQL username (default is postgres).
- password with your PostgreSQL password.
- your_database_name with the name of your database.

4. Run Prisma Migrations

To create the database tables using your **Prisma schema**, run the following commands:

```
npx prisma migrate dev
```

This will:

- Create all necessary tables in your PostgreSQL database.
- Apply any migrations defined in the prisma/migrations folder.

Additional Notes

Features and Enhancements

- **User Authentication:** Implemented secure user authentication using **bcrypt** for password hashing and **JWT** for token-based authentication.
- **Role-Based Access Control:** Added role-based permissions to secure different routes and functionalities.
- **Pagination & Filtering:** Implemented efficient pagination, sorting, and filtering for API endpoints to improve performance.
- **Error Handling:** Integrated custom error handling to provide meaningful error messages to users.
- **Responsive UI:** Ensured the frontend is fully responsive across different devices.
- **Environment Configuration:** Used **dotenv** for managing environment variables securely.
- **State Management :** Use redux and its middleware to manage the state globally

