Personal Budget Tracker - Documentation

1. Project Overview

The Personal Budget Tracker application helps users track income, expenses, and budgets.
It allows adding categories, recording transactions, managing monthly budgets, and generating summary reports.
This application uses Django REST Framework for the backend and React for the frontend.

2. Tech Stack

Frontend:
- React 18.2.0
- react-donut-chart
- react-hook-form
- react-router-dom
- recharts
- axios

Backend:
- Django
- Django REST Framework

Authentication:
- JWT-based login using DRF Authentication.

3. Features

- User Authentication (Login)
- Add/View/Edit/Delete Transactions (Income & Expense)
- Categorize Transactions
- Set and View Monthly Budget
- View Monthly Financial Summary (using charts)
- Dashboard: Total Income, Expense, and Balance
- Pagination and filtering by date/category/amount

4. API Endpoints

- POST	/auth/signin	: Login user
- GET	/report/getTotalIncomeOrExpense	: Get total income or expense
- GET	/report/getTotalNoOfTransactions	: Get total transactions

- GET	/report/getTotalByCategory	: Get totals grouped by category
- GET	/budget/get	: Fetch user's monthly budget
- POST	/report/createBudget	: Create or update budget
- POST	/transaction/new	: Create new transaction
- GET	/category/getAll	: Get all categories
- GET	/transaction/getByUser	: Get user transactions
- GET	/transaction/getById	: Get transaction by ID
- PUT	/transaction/update	: Update a transaction
- DELETE	/transaction/delete	: Delete a transaction
- POST	/budget/create	: Create monthly budget
- PUT	/category/update	: Update category
- POST	/category/new	: Create new category
- GET	/report/getMonthlySummaryByUser	: Get monthly summary by user

5. Setup & Deployment

- GitHub Repo Frontend : https://github.com/fasi-44/Budget-Tracker-react

- GitHub Repo Backend: https://github.com/fasi-44/Budget-Tracker-django

- Hosted Frontend : https://budget-tracker-react-hu18.onrender.com

- Hosted Backend : https://budget-tracker-django.onrender.com

- **Test Login Credentials**:

  Username: admin@example.com

  Password: admin@123

6. Assumptions

- User registration and password management are out of scope.
- Categories are user-defined and must be created before transactions.
- Budget is set per month per user.

7. License & Acknowledgements

- All third-party libraries used are open-source.
- D3.js-inspired components used via 'recharts' and 'react-donut-chart'.
- React UI libraries used are mentioned in package.json dependencies.
- This document and project are created solely for the technical assessment.

8. LLM / AI Contributions

- Code scaffolding and bug fixes were assisted by OpenAI ChatGPT, especially for:
    - DRF CORS and deployment configuration.
    - Environment-based API URLs
- While I was familiar with Flask, I didn't have much experience with Django. However, I explored its setup, installation, and deployment process. I learned how to use Django and Django REST Framework (DRF) through various resources, enabling me to efficiently build and deploy APIs in Django.