Personal Budget Tracker - Full Stack Application

This is a full-stack Personal Budget Tracker application built using:

Backend: Django + Django REST Framework

Frontend: React.js + D3.js

Features

Backend (Django)

* JWT Authentication

* Transaction CRUD operations

* Filter & paginate transactions by category, date range, and amount

* API endpoint for financial summary (`/api/summary/`)

* Monthly budget comparison API (`/api/monthly-report/`)

Frontend (React)

* Login page with JWT login

* Dashboard:

  * Total income, expenses, and balance

  * Pie chart for income vs. expenses

  * Category-wise expenses chart

 * Transaction Management:

* Add/Edit/Delete transactions

* Filters for category, date, amount

* Paginated view

* Budget page:

* Monthly budget vs. actual expense bar chart

How to Clone and Run this Project

Backend Setup (Django)

1. Navigate to your backend directory

```
cd budget_tracker
```

2. Create and activate a virtual environment

```
python -m venv venv

venv\Scripts\activate  # or source venv/bin/activate (Linux/macOS)
```

3. Install dependencies

```
pip install -r requirements.txt
```

4. Run migrations and create superuser

```
python manage.py makemigrations

python manage.py migrate

python manage.py createsuperuser
```

5. Start the server

```
python manage.py runserver
```

Frontend Setup (React)

1. Navigate to the frontend folder

```
cd budget-frontend
```

2. Install dependencies

```
npm install
```

3. Start the React app

```
npm start
```

API Endpoints (Django)

- /api/token/
- /api/transactions/
- /api/summary/
- /api/monthly-report/
- /api/budgets/
- /api/admin/
- /api/token/refresh/
- /api/categories/

Data Filtering (Supported Query Params)

* category: Filter by category name

* date_from / date_to: Filter by date range (YYYY-MM-DD)

* min_amount / max_amount: Filter by amount range

Example:

```
/api/transactions/?category=groceries&date_from=2025-01-01&min_amount=100&page=2
```

Visualizations (Frontend)

Dashboard

  * D3 pie chart: Income vs Expenses

  * D3 pie chart: Category-wise Expenses

Budget Page

  * D3 bar chart: Monthly Budget vs Actual Expenses

Security & Auth

* All APIs (except `/token/`) require JWT access token

* Token is stored in localStorage

* Token auto-refresh logic handled in `axios.js`

Deployment Tips

Django(python.anywhere)

* Set `ALLOWED_HOSTS`, `DEBUG=False`

* Use SQLite in production

* Add CORS settings for frontend domain

React(Netlify)

* Update `axios` baseURL to point to deployed backend

Referenced links

- **React + Django Integration Tutorial | Hello World App by Dennis Ivy**
- **Django and React: Full CRUD part1 backend API by Simple UE is better then complex**
- **How to Connect Django with Reactjs ? | GeeksforGeeks**