Project Requirements Document (PRD)

Title:

Personal Finance Tracker with Data Visualization

Overview:

This web application helps users track their personal finances by allowing them to log incomes and expenses, view their balance, set saving goals, and visualize data in an intuitive dashboard. The application supports animated charts for better insights and includes reset functionality to start over when needed.

Goals & Objectives:

- Enable users to log incomes and expenses easily
- Automatically calculate total income, expenses, and remaining balance
- Provide interactive charts for financial trends
- Offer a reset feature for data refresh
- Deliver a clean, modern, animated UI with intuitive user experience

Target User:

- Age Group: 18-40
- Individuals seeking basic personal finance management
- Budget-conscious users or students
- People who prefer visual dashboards for tracking

Features:

- Add Income: Form to input income entries (High)
- Add Expense: Form to input expense entries (High)
- View Dashboard: See total income, expenses, balance, and charts (High)
- Reset Data: Button to clear all stored data (High)

- Chart Visualization: Chart.js-based dynamic chart with animation (High)

- Responsive UI: Tailwind-powered responsive layout (Medium)

- Saving Goals (optional): Track goal targets for future savings (Low)

Functional Requirements:

Income & Expense Tracking

- Fields: Title, Amount, Date

- Stored in SQLite via Django models

- Accessible via HTML forms (with CSRF protection)

Dashboard with Chart.js

- Chart displays monthly income vs expense trend

- Data dynamically passed from Django views to Chart.js

- Responsive animations using JavaScript

Reset Functionality

- A POST button that clears all income/expense entries

- Handled in Django views

UI/UX Requirements:

- Modern, card-style layout using Tailwind CSS

- Three primary sections:

  1. Summary (Total Income, Expense, Balance)

  2. Input Forms (Add Income, Add Expense)

  3. Chart Visualization (Canvas element)

Tech Stack:

- Frontend: HTML, Tailwind CSS, Chart.js

- Backend: Django (Python)

- Database: SQLite

- Charting: Chart.js

- Hosting: Localhost / PythonAnywhere

Timeline:

- Requirement Finalization: Day 1

- UI Design: Day 2

- Model & Form Setup: Day 3

- Chart & Reset Integration: Day 4

- Testing: Day 5

- Deployment & Report: Day 6

Success Criteria:

- Dashboard loads and displays income/expense data

- Users can add at least one income and expense entry

- Chart updates in real-time or after page reload

- Reset clears the chart and summary

- No console errors or major bugs during usage

- Mobile-first design verified through testing

Deliverables:

- Working Django app with a modern UI

- Hosted or demoed version

- Code pushed to GitHub

- 2-3 minute screen recording

- Final PDF report with screenshots and features