

26 NOVEMBER 2024

EXPENSE TRACKER

TRACK YOUR WAY TO SMARTER SPENDINGS !!!

INTRODUCTION

In today's fast-paced world, managing personal finances can often feel overwhelming. Unintentional overspending, forgotten bills, and a lack of financial clarity can lead to stress and financial instability. That's where expense-tracking applications become an absolute necessity in overcoming these challenges and gaining control over personal finances. With its ability to categorise expenses, set budgets and generate insightful reports, you can identify areas for savings, make informed financial decisions and ultimately achieve your financial goals.

Expense Tracker is a project aimed at developing an easily accessible application that would collect data from users, including monthly budget, salary, daily expenses and generate personalised financial reports. It will help users to identify their potential areas for savings and make more informed financial decisions.

DESCRIPTION

Expense Tracker is a Python web application that helps to manage finances by gathering inputs from users and using them in creating detailed financial reports and visually appealing bar and line graphs to showcase spending patterns. The app allows users to add new expenses by categories and date, edit the existing ones, and set spending limits to control their monthly expenditure. The app then stores this information in a database, making it easily accessible for future reference. By providing actionable insights, this user-friendly app simplifies personal finance management, saving your time and effort.

WORKFLOW

When user opens the application, they are required to fill their budget and expenses details, then the app formulates accordingly.

1. EXPENSE ENTRY

User Interface: Display a form for entering expense details.

User Input: Amount, Category (e.g., food, transportation, entertainment), Date, Description

Backend Processing: Validate input data and Store expense details in database, associating them with the user's profile.

2. EXPENSE LISTING AND VIEWING

User Interface: Display a list of expenses, categorised by date or category.

Backend Processing: Retrieve expense data from the database based on user's profile and filters.

User Interface: Render the expense list, showing details like amount, category, date and description.

User Interaction: User can Edit or delete an expense, Add another expense.

3. GRAPH REPRESENTATION

User Interface: Display a graph generation screen.

User Input: User selects a date range or category to view more details.

Backend Processing: Retrieve expense data from the database based on the selected criteria.

Graph Generation: Creates graph as bar graph and line graph: Total expenses, Category-wise spending, Spending trends over time

4. FEEDBACK SUBMISSION

User Interface: Display a feedback form with fields for: Rating (e.g: star rating) and Suggestions to improve or to add any features(text box)

User Input: User provides feedback and submits the form.

Backend Processing: Validate the feedback data.

Store the feedback in a database.

Send a mail to the development team or relevant personnel.

5. SETTING BUDGET

User Interface: Display a budget setting screen with fields for: Budget Amount

User Input: User enters the budget details.

Backend Processing: Validate the input data and Store the budget information in the database, associated with the user's profile.

TECHNICAL IMPLEMENTATION

Languages and Frameworks Used

○ Python

Purpose: Python is the backend programming language for this project, powering the server-side logic and interaction with the database.

Usage in this project:

Flask: A micro web framework used to create the web server and handle HTTP requests. It allows for the creation of routes, rendering HTML templates, and managing form submissions.

SQLite: A lightweight database engine used for storing and managing expenses and user profile data. SQLite is used through Python's sqlite3 library to interact with the database.

○ HTML (HyperText Markup Language)

Purpose: HTML is the standard markup language used to structure content on the web.

Usage in this project:

Creating Views: HTML is used in all the pages (index.html, add_expense.html, profile.html, etc.) to structure the content, such as headings, tables, forms, buttons, and lists.

○ CSS (Cascading Style Sheets)

Purpose: CSS is used to style and visually present the HTML content. It controls layout, colours, fonts, spacing, and other visual elements.

Usage in this project:

Bootstrap: The project uses the Bootstrap framework (a popular open-source front-end framework) to simplify the design of responsive and mobile-first websites. It provides pre-built CSS classes for layout, grid systems, buttons, tables, and more.

Custom Styles: Although the main styling is done by Bootstrap, some additional custom styles to adjust or extend the default design.

○ Chart.js

Purpose: Chart.js is a JavaScript library used to render interactive charts in your web pages.

Usage in this project:

Monthly Expenses Chart: A line chart that visualises total expenses per month.

Category Expenses Chart: A bar chart that visualises expenses broken down by category.

○ SQLite Database

Purpose: SQLite is a relational database used to store the data of the expense tracker, including expenses and user profiles.

Usage in this project:

Tables:

expenses: Stores the expense records (date, category, amount, description).

users: Stores the user profile information (first name, last name).

Database Operations: SQL queries are used to interact with the SQLite database to manage the data.

CHALLENGES FACED

- **Missing Budget for the Current Month**

Problem: If a user views the dashboard without setting a budget for the current month, the remaining budget calculation may throw an error or display as negative.

Solution: We check whether a budget exists before performing the subtraction

- **SQL Errors for Duplicate Budget Entries**

Problem: Attempting to set a budget for the same month could result in a conflict in the database if INSERT is used.

Solution: Use INSERT or REPLACE to handle duplicate entries gracefully

CONCLUSION

The Expense Tracker app is an user-friendly and easily accessible tool that help user to manage their finances. With features like expense categorisation, setting spending limits , and personalised budgeting tools, the app transforms what can be a tedious task into an accessible and even empowering experience. The app's focus on user-centric design ensures ease of navigation, even for non-technical users. The Expense Tracker app, developed using Python, demonstrates the effective application. The app's architecture and design prioritise user experience and performance.

This project offers an accessible and time-saving solution for individuals needing a budget manager to help in their savings.