Expense Tracker

Abstract

This project focuses on building a secure and user-friendly Expense Tracker web application. The system enables users to register, log in, manage expenses, set budgets, and analyze financial patterns. It emphasizes information security through encrypted passwords and role-based access. The project's main aim is to help users track their expenses effectively while maintaining confidentiality and integrity of data.

Introduction

In today's digital world, personal finance management is essential. People often struggle to control their monthly expenses, which leads to overspending. The proposed Expense Tracker provides a solution by recording, categorizing, and analyzing expenses. It also offers a budget management feature, ensuring that users stay within their financial limits.

This project integrates Flask as the backend framework, Bootstrap for responsive UI, and SQLite/MySQL for secure data storage. The focus remains on simplicity, usability, and security best practices like hashed passwords, session management, and input validation.

Tools Used

- Python (Flask Framework): For backend development.
- SQLite / MySQL: For secure data storage.
- Bootstrap 5: For frontend design and responsiveness.
- Matplotlib / Pandas: For expense visualization and CSV/Excel handling.
- Werkzeug Security: For password hashing.
- Git & GitHub: For version control and deployment.
- Render/Heroku: For cloud deployment.

Steps Involved in Building the Project

1. Requirement Analysis: Identified features such as user registration, login, expense tracking, budget setting, and data visualization.
2. Environment Setup: Installed Python, Flask, Bootstrap, SQLite, and required dependencies.
3. Database Design: Created models for Users, Expenses, and Reports. Implemented relationships and ensured proper constraints for security.

4. User Authentication: Implemented registration and login with password hashing (using Werkzeug). Added Flask-Login for session management.
5. Expense Management: Developed routes for adding, editing, viewing, and deleting expenses. Ensured user-specific access control.
6. Budget Feature: Enabled users to set and update their monthly budget. Triggered alerts if expenses exceed budget.
7. Data Visualization: Integrated Matplotlib and Pandas for graphical expense insights (category-wise charts).
8. Import/Export: Added CSV/Excel upload and export functionality for financial data portability.
9. UI & Dark Mode: Designed responsive Bootstrap UI with optional dark-mode support.
10. Testing & Deployment: Verified all modules, checked for SQL injection prevention, session timeout, and deployed the project on GitHub and Render.

Conclusion

The Expense Tracker Project successfully demonstrates how secure web applications can help individuals manage their finances efficiently. With features like budget alerts, visual analytics, and secure authentication, the project provides both usability and information security.

This project enhanced our understanding of Flask, databases, frontend-backend integration, and secure coding practices. It can be extended further by integrating advanced features such as AI-based expense prediction, mobile app support, and cloud-based storage for scalability.