Budget Tracker Web Application

1. Project Overview

The Budget Tracker Web Application was developed to support individual in managing their personal, set per-category budgets, monitor financial goals (such as savings, debt repayment, mortgage targets) and analyze spending habits. The primary goal was to create a user-friendly and dynamic web application that provides real-time financial insights and improves budgetary discipline.

The application includes user authentication such as registration, logic, logout, password reset and a personalized dashboard displaying key financial metrics. Through intuitive interfaces and dynamic charts, users can monitor their financial status, receive early warnings when approaching budget limits and manage their savings objectives.

2. Technical Approach

The backend of the application is implanted using Java 21 with the Spring Boot 3.4 framework. The project architecture follows the Model-View-Controller pattern, ensuring a clean separation of concerns and maintainability. The backend handles business logic, user authentication, budget calculations, goal management and serves REST APIs for dynamic data interactions.

SQLite was chosen as the database for its simplicity and lightweight nature, integrated with Hibernate JPA for object-relational mapping. Authentication and security measures were implemented using Spring Security, ensuring secure password handling with BCrypt encryption.

The Front end is based on Thymeleaf templates, enhanced with HTMX for dynamic partial page updates without full reloads. For data visualization, Chart.js was integrated to display line charts, bar charts, pie charts and spending analyses.

The application's architecture also integrates responsive design principles, ensuring accessibility across different devices and screen sizes.

The complete codebase is hosted on GitHub at: https://github.com/Katsiatyna/BudgetTracker

3. Development Process

The project was developed following an iterative approach. During the conception phase, a high-level system architecture was defined, and the target features were selected. Throughout the development and reflection phases, several adjustments were made:

- Dynamic filtering by date range was added to improve analytics.
- A warning system was integrated to alert users when reaching 90% of a category budget.
- A goal-tracking module was introduced to monitor savings, debt and mortgage targets.
- CSV export functionality was added to allow users to download their financial data.

4. Lessons Learned

Throughout the development of the Budget Tracker Web Application, several technical and organizational challenges emerged, offering important lessons for future projects. One significant challenge was ensuring effective synchronization between the front end and backend. Designing a consistent API structure and integrating HTMX for dynamic page updates required careful planning to ensure smooth and responsive user experiences.

Another important lesson was the critical role of data validation and security. It became clear that security managing user credentials and validating user inputs at both the frontend and backend layers was essential to maintaining application robustness and protecting user information.

The iterative nature of the development process also revealed that adapting to evolving requirements is vital for project success. For example, new features such as category spending warnings and CSV data export were incorporated based on mid-development reflections, highlighting the importance of flexibility and ongoing evaluation.

Finally, the need for responsiveness and accessibility became increasingly apparent during testing across different devices. By adopting responsive design principles from the early stages of development, the application remained accessible and user-friendly on various screen sizes without requiring major redesigns later.

Overall, the project strengthened practical skills in full-stack web development, Spring Boot architecture design, REST API integration and effective use of version control systems such as GitHub.