School of Arts, Humanities and Social Science

Module title and code: Software Development 2 - CMP020L004S Module

Tutor: Dr Lisa Haskel & Dr Mamoona Humayun

Sprint 3 Submission for SmartBudget Application

NAME	STUDENT ID
Md Tanzeem	A00033581
Olarewaju Saheed Musari	A00016697
Mallikharjuna Thallapalli	A00034084
Kamwinula Shilogile	SHI22575536

Project Overview

SmartBudget is a personal finance management web application designed to help users track their expenses, set budgets, and achieve financial goals. The application provides a user-friendly interface for monitoring spending habits, creating budgets for various categories, and visualizing financial progress over time.

User Stories and Implementation Progress

User Authentication

- As a user, I want to securely log in to access my financial data
- As a user, I want to register for a new account
- As a user, I want to log out of the application

Dashboard

- As a user, I want to see an overview of my financial status on a dashboard
- As a user, I want to view my current month's income and expenses at a glance
- As a user, I want to see my remaining balance for the month

Transactions

- As a user, I want to record my income and expenses
- As a user, I want to categorize my transactions
- As a user, I want to view my recent transactions

Budgets

- As a user, I want to set budgets for different spending categories
- As a user, I want to view my budget progress visually
- As a user, I want to be alerted when I'm close to exceeding a budget

Goals

- As a user, I want to set financial goals
- As a user, I want to track my progress toward my goals

Technical Implementation Details:

Database Design and Implementation:

Our application uses a MySQL database with the following key tables:

1. **Users**
    - Stores user authentication details and profile information
    - Primary fields: id, username, email, password (hashed), created_at
2. **Transactions**
    - Records all financial transactions (income and expenses)
    - Primary fields: id, user_id, amount, type (income/expense), category_id, description, date
3. **Categories**
    - Defines transaction categories for better organization
    - Primary fields: id, user_id, name, type (income/expense), color
4. **Budgets**
    - Stores budget limits for each category
    - Primary fields: id, user_id, category_id, amount, period (monthly, weekly, etc.)
5. **Goals**
    - Manages user's financial goals
    - Primary fields: id, user_id, name, target_amount, current_amount, deadline, created_at

Application Architecture

The application follows the MVC (Model-View-Controller) pattern:

- **Models**: Interact with the database and define the data structure
- **Views**: Implemented using Pug templates for dynamic HTML generation
- **Controllers**: Handle user requests and business logic

Key Technologies and Libraries

- **Backend**: Node.js with Express.js
- **Database**: MySQL

- **Frontend**: HTML, CSS, Bootstrap
- **Template Engine**: Pug (formerly Jade)
- **Authentication**: Express-session, bcryptjs
- **Docker**: For containerization of the application and database

Implementation Progress

1. **Authentication System**
    - Implemented user registration, login, and logout functionality
    - Session management using express-session
    - Password hashing with crypts
2. **Database Integration**
    - Successfully set up MySQL database
    - Created database schema with relationships
    - Pre-filled database with sample data for testing
3. **Frontend Development**
    - Created responsive dashboard layout
    - Implemented budget progress bars with color-coding
    - Designed transaction history display
4. **Dynamic Content Rendering**
    - Implemented Pug templates for generating HTML
    - Created static content pages
    - Added dynamic data pulling from the database

Repository and Deployment Information

GitHub Repository

https://github.com/Mdtanzeem702/Smart-Budget-App

Database Setup

- The application uses MySQL 8.0 for data persistence
- Docker configuration handles database initialization
- Schema migrations are managed through the application

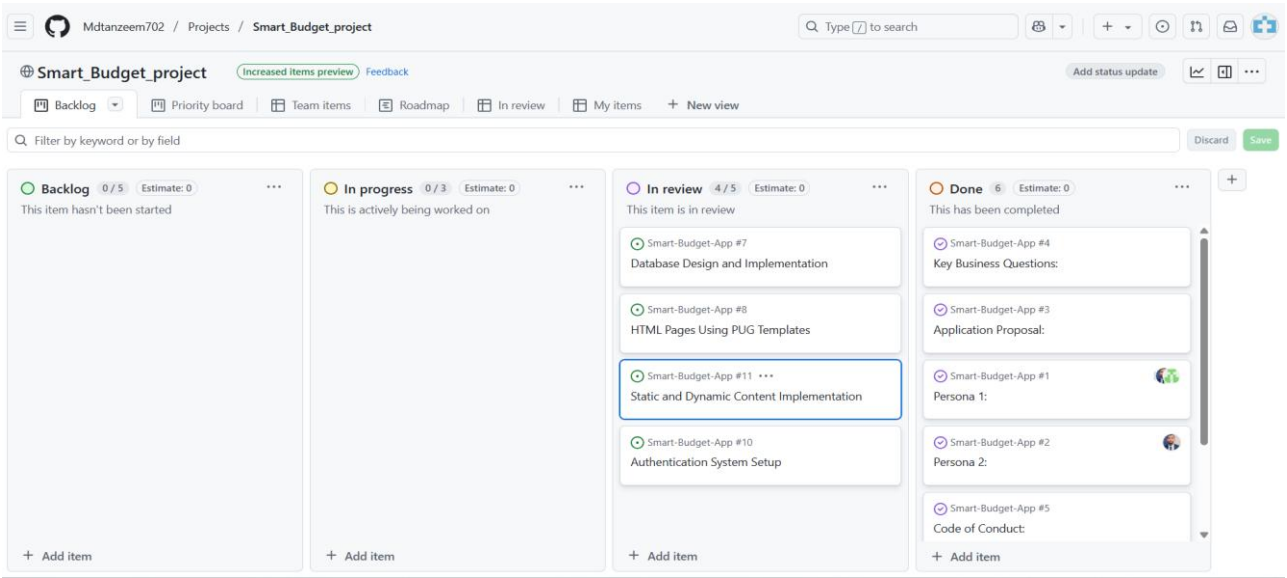
Deployment Instructions

1. Clone the repository
2. Make sure Docker and Docker Compose are installed
3. Run docker-compose up to start the application
4. Access the application at http://localhost:3000

Updated Task Board

Our task board clearly shows our progress toward completing the technical deliverables for Sprint 3:

- Completed database design and implementation
- Created HTML pages using PUG templates
- Implemented static and dynamic content
- Set up authentication system
- Developed core application features



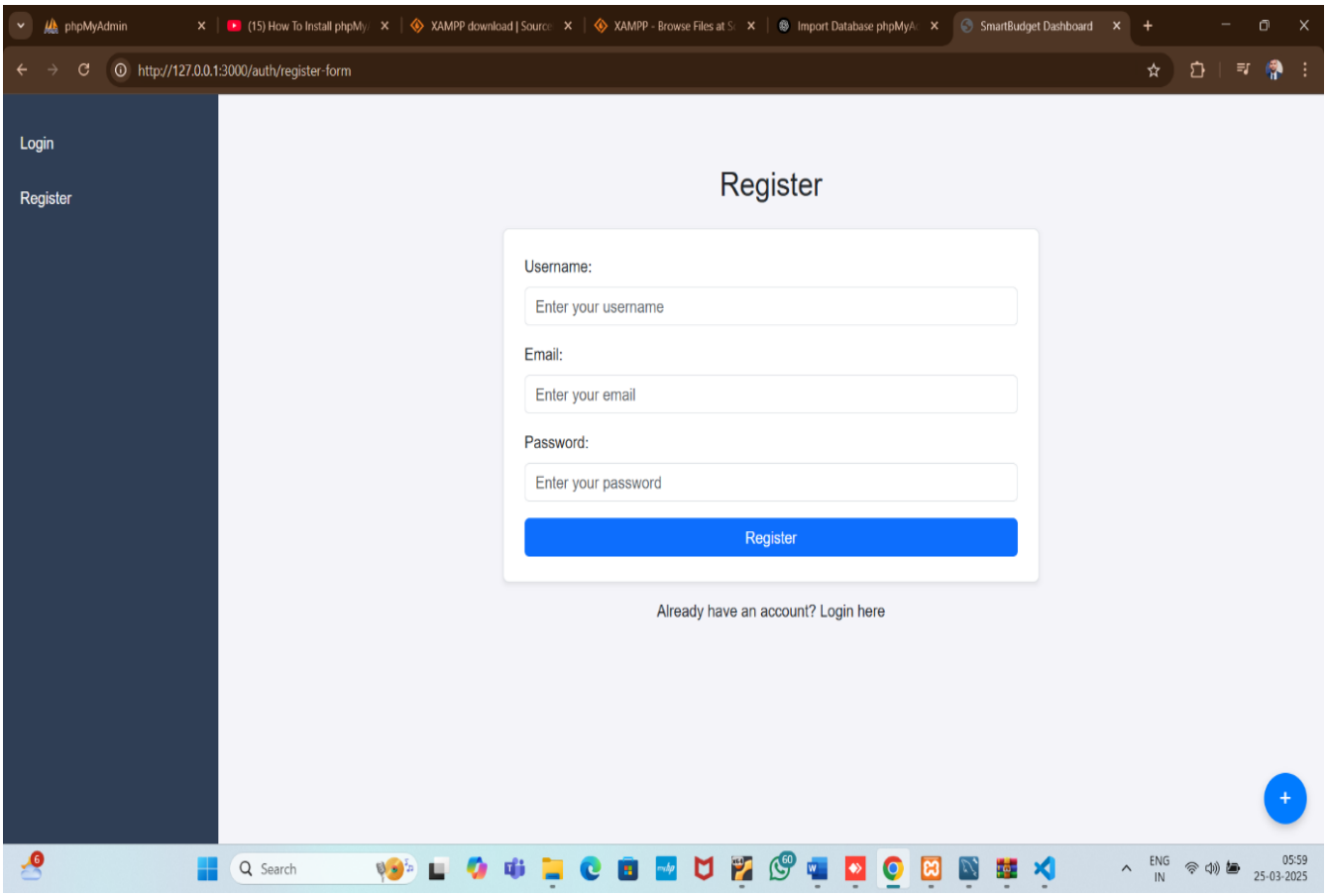
GitHub Repository Activity

Our GitHub repository shows regular commits from all team members throughout the sprint, demonstrating active collaboration and consistent progress. Key commits include:

- Implementation of database models
- Creation of route controllers
- Development of PUG templates
- Setup of authentication middleware
- Docker configuration for deployment

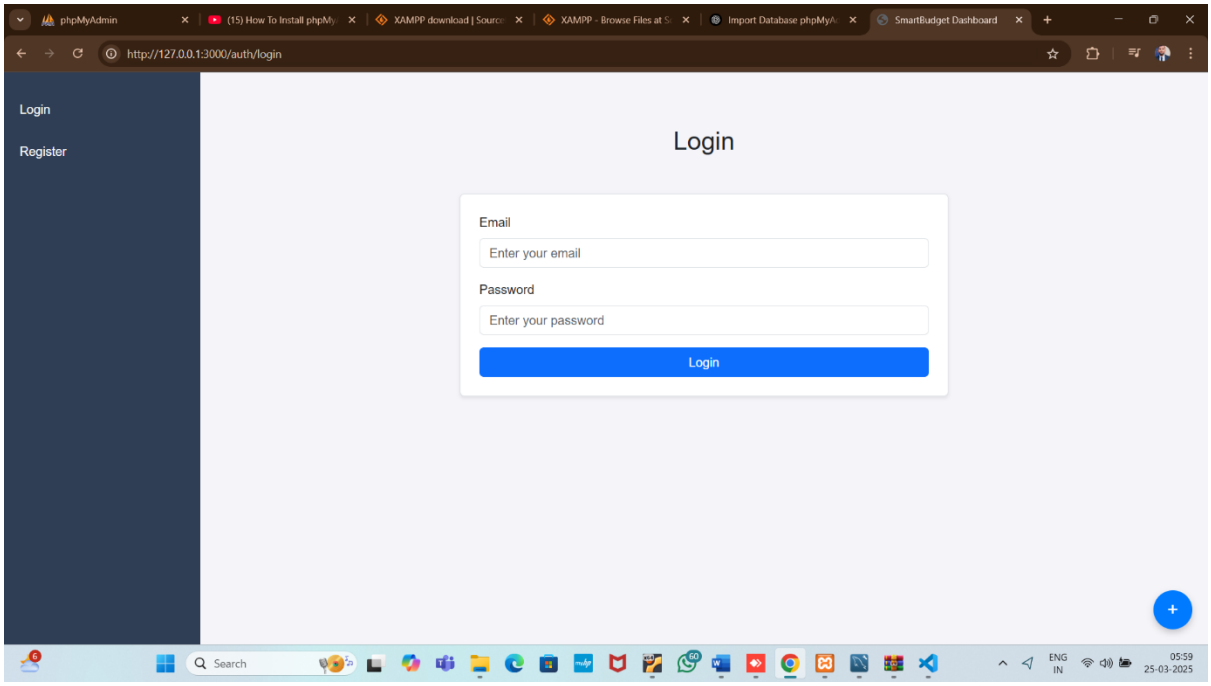
Registration Page:

Clean user registration form with username, email, and password fields.



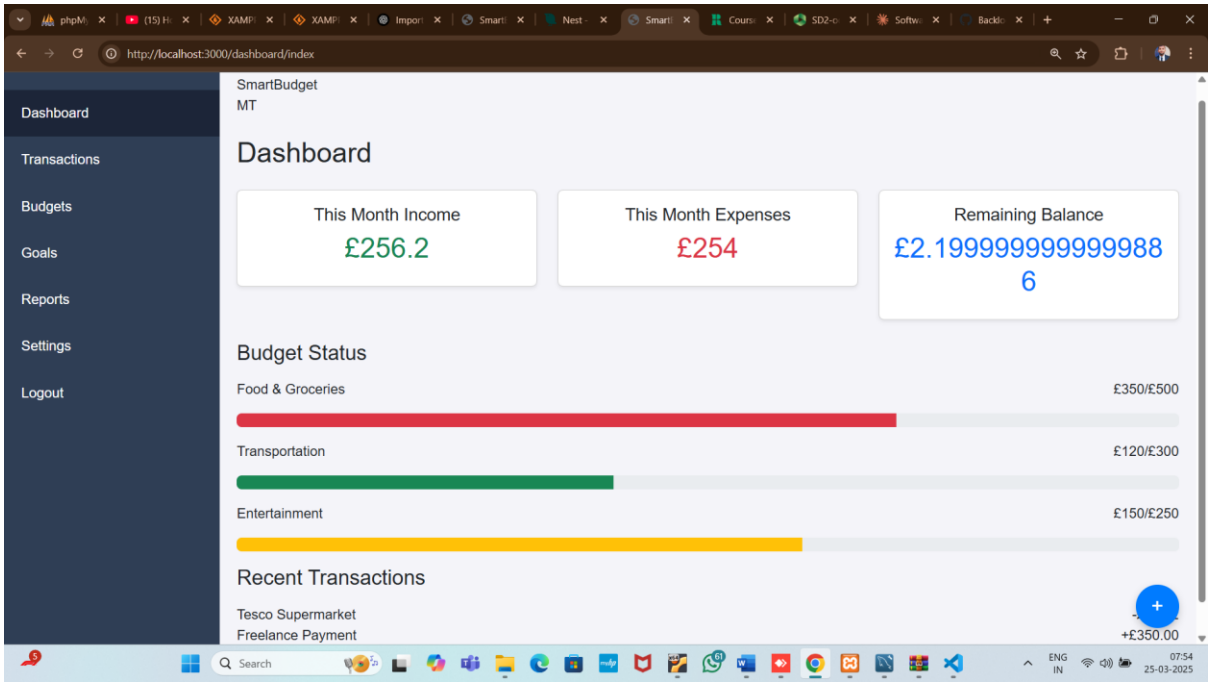
Login Page:

Secure authentication interface matching the design style of the registration page.



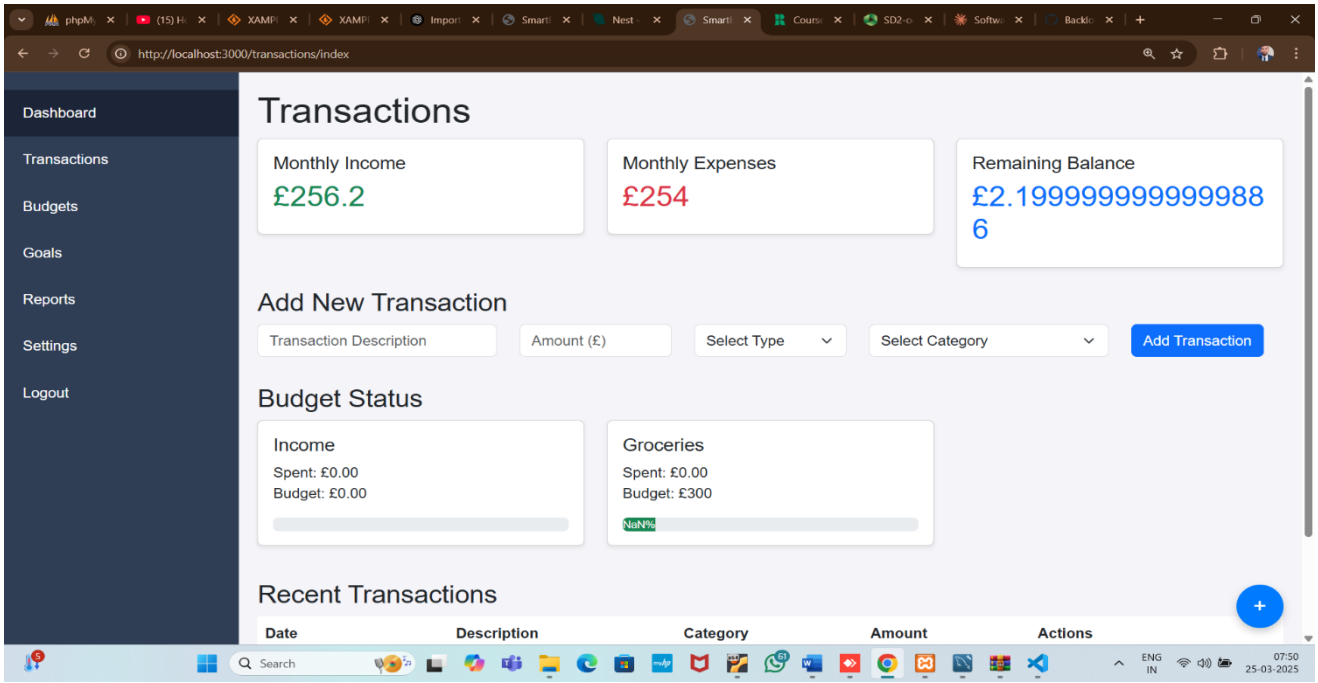
Dashboard:

Main overview with financial summary cards, color-coded budget progress bars, and recent activity display**.**



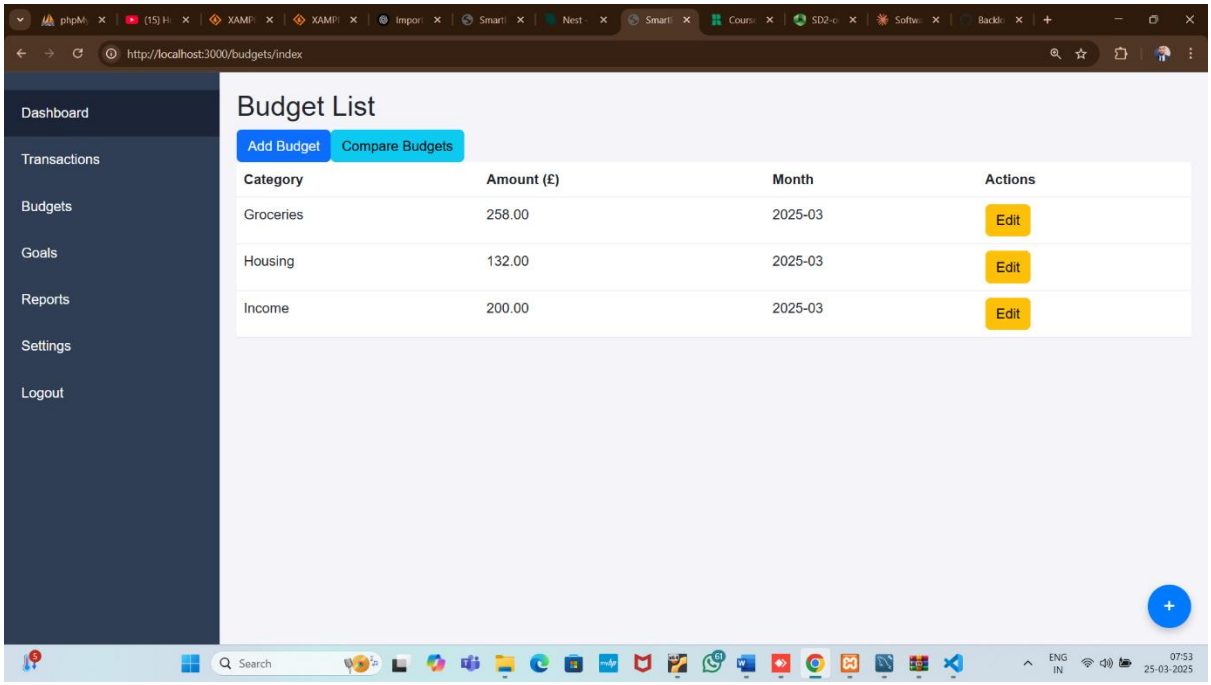
Transactions Page:

Complete transaction management with income/expense tracking, form for adding new entries, and recent transaction history.



Budget List:

Functional budget management interface showing categories, amounts, and edit options.



Conclusion

Sprint 3 has seen significant progress in the development of the SmartBudget application. We have successfully implemented core features including user authentication, transaction management, budget tracking, and a user-friendly dashboard. The application now has a working prototype that demonstrates the concept and provides value to users. Our team has collaborated effectively, with regular commits from all members as shown in our GitHub repository history.