CSIS 4495 - 002 Applied Research Project

Project Proposal

Finance Tracker Web Application for Budget Management

Nidhi Nidhi, 300378175

CSIS 4495-002

Instructor: Padmapriya Arasanipalai Kandhadai

Date: January 23, 2025

Douglas College

Introduction

Domain and Background

Personal finance management is critical for individuals to achieve financial stability. This project develops a web application that helps expense tracking, budget customization, AI assistance, and pdf report generation.

Problem Statement

- How can users efficiently track income/expenses while gaining actionable insights to save?
- Can AI integration improve personalized budget recommendations?
- Recommendations based on expense and savings data.

Literature Review

Existing tools focus on basic tracking but lack recommendation features. Current gaps include limited AI integration and static visualization tools.

Hypotheses and Benefits

Hypothesis: AI-driven recommendations will improve user engagement and financial decision-making.

Benefits: Enhanced user experience, reduced manual input, and actionable insights for long-term financial planning.

Proposed Research Project

Research Design and Methodology

Objective: Develop a full-stack web app using Node.js (backend), Bootstrap/EJS (frontend), and MongoDB (database).

Methodology: Agile development with iterative testing.

Justification: Agile allows flexibility for AI integration and user feedback.

Technologies

OS: Windows/Linux (cross-platform compatibility).

Frontend: Bootstrap 5, EJS templating.

Backend: Node.js, Express.js.

Database: MongoDB (NoSQL for flexible expense categorization).

AI Tools: Using Google's Gemini API for expense recommendations.

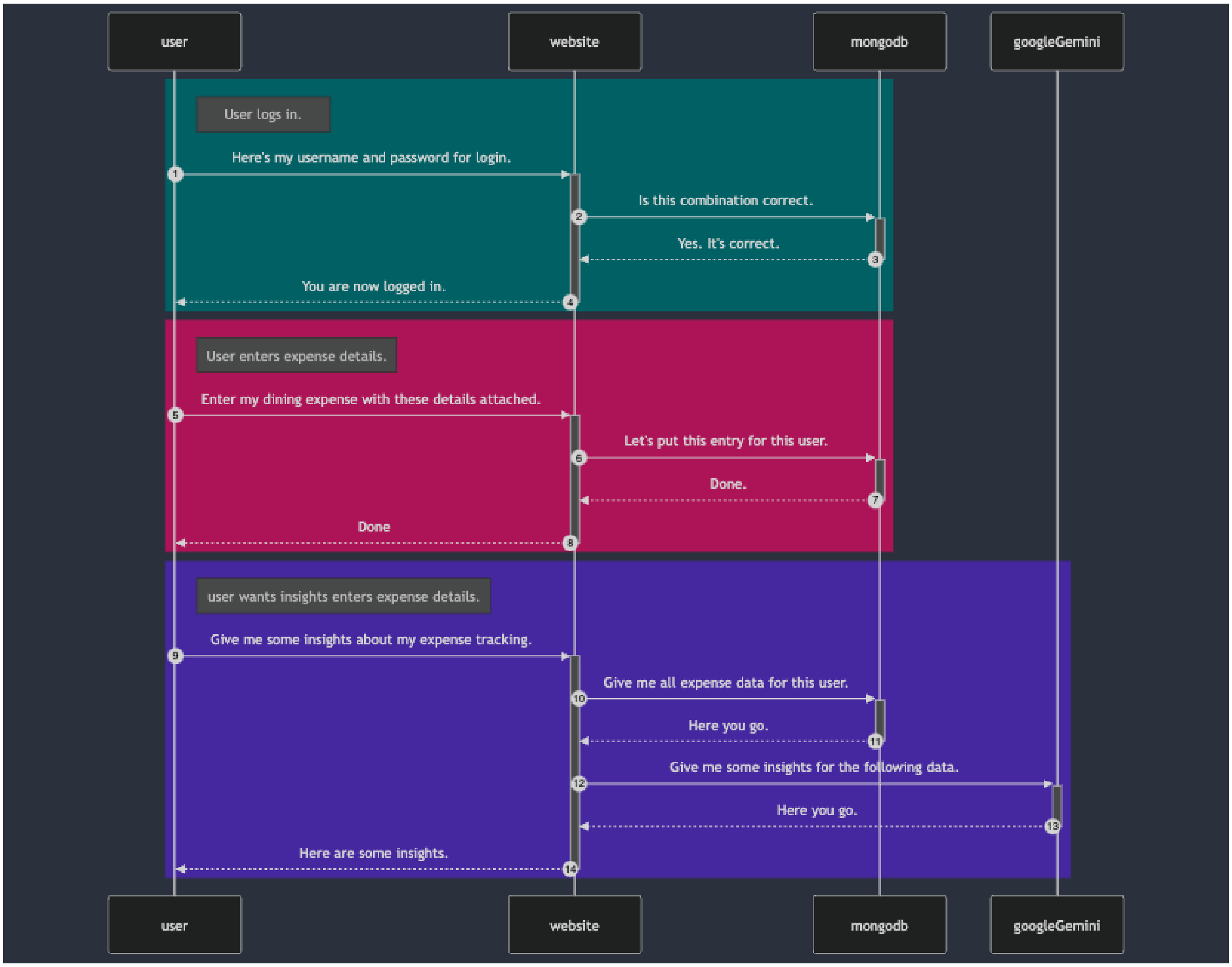
Expected Results

A functional app enabling users to track expenses, set budgets, and generate PDF reports.

AI-driven insights (e.g., "Fuel expenses increased by 20% this month").

Contribution: Empowering users to make data-driven financial decisions.

Sequence Diagram


user
website
mongodb
googleGemini
User logs in.
Here's my username and password for login.
1
Is this combination correct.
2
Yes. It's correct.
3
You are now logged in.
4
User enters expense details.
Enter my dining expense with these details attached.
5
Let's put this entry for this user.
6
Done.
7
Done
8
user wants insights enters expense details.
Give me some insights about my expense tracking.
9
Give me all expense data for this user.
10
Here you go.
11
Give me some insights for the following data.
12
Here you go.
13
Here are some insights.
14
user
website
mongodb
googleGemini

Explanation:

1. User Login

Process Flow:

1. User Action: Navigates to the login page and enters their username and password.

2. System Interaction:
   - The frontend (Bootstrap/EJS) sends credentials to the Node.js backend.
   - The backend queries MongoDB to validate the user's credentials.
   - If valid, MongoDB confirms the match, and the backend grants access.
3. User Outcome: Redirected to the dashboard with personalized financial data.

Key Features:

- Secure authentication using MongoDB's encrypted storage.
- Session management to maintain user login state.

2. Expense Entry

Process Flow:

1. User Action: On the dashboard, clicks "Add Expense" and fills in details (e.g. Dining, amount: $50, date: Jan 25, 2025).
2. System Interaction:
   - The frontend sends the expense data to the Node.js backend.
   - The backend stores the entry in MongoDB under the user's unique ID.
3. User Outcome: A confirmation message ("Done") appears, and the expense is instantly reflected in charts/tables.

Key Features:

- Real-time updates using EJS templating.
- Categorization (e.g., dining, rent) for organized tracking.

3. AI-Powered Insights Generation

Process Flow:

1. User Action: Clicks "Generate Insights" on the dashboard.
2. System Interaction:
   - The backend fetches all historical expense data from MongoDB.
   - Data is sent to Google Gemini for analysis.
   - Gemini identifies patterns (e.g., "Fuel expenses increased by 20% this month") or suggests budget adjustments.
3. User Outcome: Insights displayed as interactive charts or text recommendations (e.g., "Reduce dining expenses by 15% to meet your savings goal").

Key Features:

- Gemini Integration: Predictive analytics for personalized financial advice.
- Dynamic Visualization: Charts update automatically to reflect trends.

End-to-End User Journey Example

1. Login: User authenticates securely.
2. Expense Tracking: Adds daily expenses, categorized for clarity.
3. Budget Management: Sets monthly budgets for categories (e.g., $300 for groceries).
4. Insights & Reports:
   - Uses AI insights to adjust spending habits.
   - Generates PDF reports (extra feature) for offline review.
5. Logout: Session ends, protecting sensitive data.

Integration with Tech Stack

- Frontend (Bootstrap/EJS): Responsive UI for seamless interaction.
- Backend (Node.js/Express.js): Handles logic, authentication, and API calls.

- Database (MongoDB): Stores user data with NoSQL flexibility.
- AI (Google Gemini): Enhances decision-making through data analysis.

Value to Users

- Efficiency: Automates expense tracking and reporting.
- Personalization: Tailored insights improve financial literacy.
- Accessibility: Web-based design ensures cross-device compatibility.

This use case demonstrates how the app transforms raw financial data into actionable insights, empowering users to achieve their financial goals.

Project Planning and Timeline

Timeline (Overview)

Phase	**Dates**	**Deliverables**

Research & Design	Jan 17–31, 2025	Finalize tech stack
Development	Feb 1–28, 2025	Core features (user auth, expense tracking)
Midterm Progress	Feb 28, 2025	Working prototype + Midterm Report
AI Integration	Mar 1–21, 2025	AI recommendation engine, PDF export
Testing & Polish	Mar 22–Apr 4	Beta testing, bug fixes
Final Submission	Apr 11, 2025	Full implementation + Final Report

Responsibilities

Sole developer (Nidhi) responsible for all tasks.

Work Log

Date	**Hours**	**Description**
Jan 23, 2025	3	Researched Node.js, express, EJS & bootstrap frameworks
Jan 26, 2025	2.5	Created Project Proposal

Closing and References

Acknowledgments

Thanks to CSIS 4495 instructor (Priya) for guidance.

References

1. MongoDB Documentation. (2023). https://www.mongodb.com/docs/

2. Gemini API Documentation. https://ai.google.dev/gemini-api/docs/quickstart/

3. Node.js Documentation. https://nodejs.org/

4. Express.js Documentation. https://expressjs.com/

5. Bootstrap Documentation. https://getbootstrap.com/

6. EJS Documentation. https://ejs.co/