AI-Powered Personal Finance Management Web Application

Project Summary

The problem we aim to solve is the inefficiency and limitations present in current financial management tools. Many existing applications rely on static budgeting and basic trend analysis, offering minimal predictive capabilities or actionable insights. Users often struggle with manually categorizing expenses, which is time-consuming and prone to errors. Our solution is an AI-powered Personal Finance Management Web Application that will leverage machine learning to provide real-time financial analysis, predictive analytics, and personalized recommendations. The application will automate expense categorization, forecast future spending patterns, and help users optimize their financial habits.

The client for this project includes individual users, small business owners, financial advisors, and personal budget managers. Each of these user segments requires an efficient tool to automate financial tracking, predict spending trends, and provide data-driven recommendations to improve financial decision-making.

The deliverables for the project will include a fully functional web application offering real-time financial insights and automated expense tracking. The frontend will be developed using React to provide an intuitive, responsive user interface, while the backend API will be developed using Node.js/Express to handle user requests and data processing. The machine learning-powered predictive analytics will be developed using Python to deliver personalized financial recommendations. MongoDB will securely store user data, and detailed API documentation and testing reports will ensure the application is both robust and user-friendly. Additionally, a user guide will be provided for both end-users and administrators.

This application will benefit the client by reducing manual data entry and categorization, increasing financial awareness, and helping users make smarter financial decisions. By offering predictive analytics and personalized recommendations, the tool will enhance decision-making for small business owners and financial advisors, leading to improved overall financial health and user satisfaction.

Data Summary

The raw data for the application will come from transactional data inputs (Jagtap, 2024), which can be uploaded manually or retrieved through integrations with financial institutions, such as banks and payment platforms. The data will be processed and stored securely in MongoDB. Machine learning algorithms will process the transactional data to categorize expenses, predict future spending patterns, and provide personalized financial recommendations based on user behavior.

The data management lifecycle will include the design phase, where the data model will be structured to capture financial transactions and user behavior. During development, the data will be ingested, processed, and categorized in real-time by the application. The maintenance phase will involve continuous monitoring to ensure data accuracy and to make adjustments to the machine learning models as new data is collected.

Any data anomalies, such as incomplete or inconsistent data, will be handled by implementing data validation rules and outlier detection techniques within the machine learning models. Ethical concerns will be addressed by ensuring the data complies with privacy regulations like GDPR, with steps taken to anonymize user data whenever possible. All data will be encrypted in transit and at rest to ensure that sensitive financial information is protected.

Implementation

The application will be developed using an Agile methodology, divided into key phases. The first phase, requirements gathering, will involve collecting input from stakeholders such as individual users, financial advisors, and small business owners to define functional requirements. In the design and prototyping phase, wireframes and prototypes will be created to define the user interface and experience. The development phase will involve the frontend being built using React, the backend developed with Node.js/Express, and machine learning integration using Python to provide predictive analytics. Testing and validation will ensure that the app meets user needs, including performance optimization and security audits. The final phase will focus on deployment and post-launch monitoring to ensure smooth operation and the incorporation of user feedback.

Timeline

The project will span seven months, starting on January 1, 2025, and concluding on July 31, 2025. The first two weeks will be dedicated to project setup, ensuring all stakeholders are aligned and the development environment is prepared. From January 16, 2025, through March 31, 2025, the core development phase will focus on building the main functionalities, including the dashboard, backend API, and basic predictive features. From April 1 to May 31, 2025, machine learning capabilities will be integrated to enable predictive financial insights and personalized recommendations. The application will undergo comprehensive testing, performance optimization, and security audits during June 2025, followed by deployment in July 2025. User training will also be provided during this phase.

Evaluation Plan

The verification of the project will occur at each stage of development through unit tests, integration tests, and user acceptance testing (UAT). After the application is deployed, it will undergo continuous monitoring to ensure its stability, and any identified issues will be addressed

promptly. Validation will occur upon project completion by ensuring that all functional requirements have been met, including the accuracy of predictive analytics and the overall user experience.

Resources and Costs

The hardware and software costs will include licenses for development tools such as VS Code or WebStorm, costing $200 per developer annually. GitHub will be used for version control at $100 per month, while MongoDB Atlas will provide cloud database services for $200 per month, and AWS Cloud Hosting will be used for infrastructure at a cost of $500 per month.

Labor costs for the project will include:

- Frontend developers, with two developers at $120k per year each.
- Backend developers, with two developers at $130k per year each.
- A machine learning engineer at $140k per year.
- A DevOps engineer at $125k per year.
- A QA engineer at $90k per year. Additionally, support roles such as a project manager, business analyst, and UX designer will also be required.

Environmental costs will include deployment and hosting, estimated at $500 per month on AWS Cloud, with ongoing maintenance and software updates factored into the project's budget.

Resources

Jagtap, S. (2024). *Indian Personal Finance and Spending Habits* [Data set]. Kaggle.
https://www.kaggle.com/datasets/shriyashjagtap/indian-personal-finance-and-spending-habits