

AI Based Personal Finance Management System

Manasa S, S.K. Shivashankar, P. Prasanna, H.P. Mohan Kumar

Department of Computer Science and Engineering, PES College of Engineering, Mandya, Karnataka, India

Abstract: Managing personal finances has become increasingly challenging in today's dynamic digital economy due to diverse income streams and complex financial obligations. Many existing finance management tools fail to offer adequate personalization, automation, or intelligent recommendations, making them less effective for a broad user base. This paper introduces an AI-driven Personal Finance Management System—a web-based platform built using machine learning, Flask, and financial analytics—to provide users with smart tracking of expenses, budget predictions, and timely alerts. The system supports integrated income and expense tracking, savings goal monitoring, and bill reminders, along with future expense forecasting via Linear Regression. It features a user-friendly dashboard, automated email notifications with Flask-Mail, and background task management through APScheduler. With a modular architecture and responsive design, the system simplifies financial management and serves as an educational model for developers, particularly in the Indian context. Evaluation results highlight its functional effectiveness and potential for further enhancement with advanced forecasting and mobile app integration.

Keywords: Personal Finance Automation, Expense Forecasting, Flask Web Framework, AI in Financial Planning, Linear Regression, Smart Alerts, Email Automation, Task Scheduling, Flowgorithm, Data Visualization, MySQL Integration, Budgeting Assistant.

1. INTRODUCTION

In the current financial landscape, individuals often face the challenge of managing multiple income sources, expenses, loans, and savings simultaneously. While commercial finance applications are widely available, many fall short in offering intelligent insights, contextual customization, or cost-effective access—especially for users in India. Manual financial tracking increases the risk of missed payments, budget mismanagement, and poor financial decisions. To address these issues, this paper presents a customizable, AI-powered personal finance platform. Developed as an open-source web application, the system enables users to monitor their financial activities securely and intelligently. Leveraging technologies such as Python (Flask), MySQL, and modern frontend frameworks, this tool bridges the gap between smart financial recommendations and user-friendly design, with a focus on affordability and personalization.

2. LITERATURE SURVEY

Existing Tools and Systems

Several personal finance applications are available in the market, each offering varying degrees of functionality. For instance:

- Walnut uses SMS-based tracking and reminders but lacks machine learning and forecasting features.
- You Need A Budget (YNAB) supports goal-oriented budgeting and cloud synchronization, but it is subscription-based and does not include automated alerts.
- Money View provides credit score monitoring and expense categorization, yet it offers limited customization and lacks predictive intelligence.
- Goodbudget utilizes envelope-style budgeting but relies heavily on manual input and does not support forecasting.
- Mint offers budgeting tools and credit tracking, but is primarily US-centric and not optimized for Indian users or predictive analytics.

Related Academic Contributions

- A study titled “Personal Financial Management Using AI Techniques” (IEEE, 2020) proposed AI-based models for finance management but lacked implementation details.
- “Predicting Personal Expenses Using Machine Learning” (Springer, 2021) focused on regression models for forecasting expenses but did not provide a user interface or real-time alert mechanisms.
- The “Design and Implementation of a Personal Finance Tracker” (IJERT, 2019) explored basic CRUD operations within a web application, omitting AI capabilities and notification features.

Identified Research Gap

While numerous tools exist, there is a noticeable lack of freely accessible, AI-integrated, and user-friendly financial management systems specifically designed for Indian users. Most current applications do not incorporate forecasting capabilities, proactive alerting, or interactive dashboards driven by machine learning.

3. SYSTEM DESIGN AND METHODOLOGY

Architecture Overview

- Frontend (UI Layer): Built using HTML5, CSS3, and Bootstrap to deliver a responsive and intuitive interface.
- Backend (Application Logic): Implemented with Python's Flask framework, managing user sessions, routing, and machine learning operations.
- Database (Storage Layer): MySQL database used to store user credentials, financial records, and alert data.

Core Functional Modules

1. Authentication Module: Facilitates secure user registration and login.
2. Financial Entry Module: Users can input income, expenses, investment plans, and savings targets.
3. Alert Mechanism: Users can configure custom bill reminders and spending thresholds.
4. Analytics Dashboard: Presents a visual summary of financial status and triggers.
5. Email Notification System: Sends automated alerts using Flask-Mail and background job scheduling with APScheduler.
6. ML-Based Forecasting: Employs Linear Regression to estimate next month's expenses based on historical data.

Development Workflow

Before implementing each module in Python, flowcharts were designed using Flowgorithm to ensure logical clarity and efficient translation into code.

4. IMPLEMENTATION

Technology Stack

The development of the system is structured across three main layers:

- Frontend: Implemented using HTML5, CSS3, Bootstrap, and Jinja2 templates to ensure a responsive, user-friendly interface with clear navigation and visual alerts.
- Backend: Built with Python 3.9 and the Flask 2.2.5 framework. Each functional route (e.g., registration, login, profile management, alert settings, dashboard) is handled through modular Python functions for better maintainability.
- Database: MySQL 8.x is used as the backend database to store user details, financial inputs, alerts, and prediction history.
- Machine Learning: The system uses the Scikit-learn library to implement a Linear Regression model, which learns from past expense data to predict future spending trends.
- Scheduling & Alerts: The APScheduler library schedules background jobs every five minutes to check for alert conditions. When triggered, Flask-Mail is used to send email notifications through an SMTP (Gmail) server.

Implementation Highlights

- A responsive and clean frontend interface was designed with visual indicators, form validation, and flash messaging.
- Flask routing handles backend logic and user interactions with clearly defined modules.
- The ML module trains on user financial history to forecast upcoming expenses.
- Scheduled background tasks ensure continuous monitoring and real-time notifications.

5. RESULTS & DISCUSSION

5.1 Functional Testing

Module	Test Scenario	Outcome
Registration	Valid email and password inputs	Passed
Login	Both correct and incorrect credentials	Passed
Financial Entry	Accepts structured income and expense data	Passed
Alerts	Triggered on bill due or spending limit	Passed
Prediction Module	Generate Forecast from user history	Passed
Email Notification	Sends alerts via Flask-Mail	Passed

Technology Summary

Component	Stack Used
Frontend	HTML5, CSS3, Bootstrap, Jinja2
Backend	Python 3.9, Flask 2.2.5
Database	MySQL 8.x
Email Alerts	Flask-Mail with Gmail SMTP
Scheduler	APScheduler
ML Module	Scikit-learn (Linear Regression)

UI & Usability Observations

- The application is fully responsive across multiple devices and screen sizes.
- Client-side validation ensures accurate data entry and prevents blank submissions.
- Feedback from peer users praised the clean interface, intuitive design, and simplicity.

DISCUSSION

The system demonstrates effective integration of machine learning for personalized expense forecasting. Real-time alerts and a structured dashboard significantly improve the user's financial oversight. The modularity of the backend and clarity of design aid in future enhancements. While the prototype achieves core functionality, upcoming improvements like password encryption, data visualization with charts, and security layers can enhance its robustness.

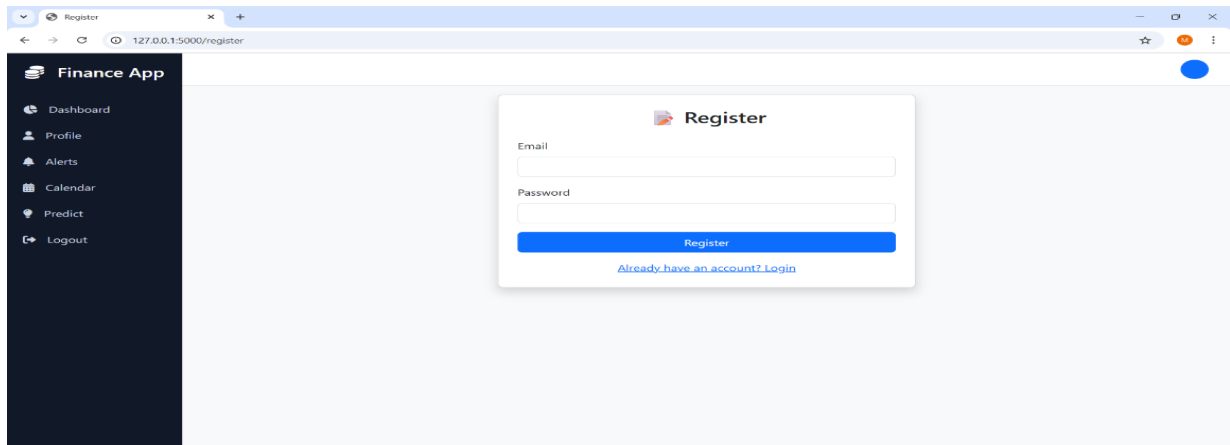
6. SNAPSHOTS

This section presents visual evidence of the system's functionality and user interface. Each screenshot demonstrates the successful implementation of different modules within the AI-Based Personal Finance Management System.

User Registration Page

The registration interface allows new users to sign up by entering their email and password. The interface is responsive and built using Bootstrap 5. Upon successful registration, a confirmation message is shown.

Figure 1: User Registration Interface

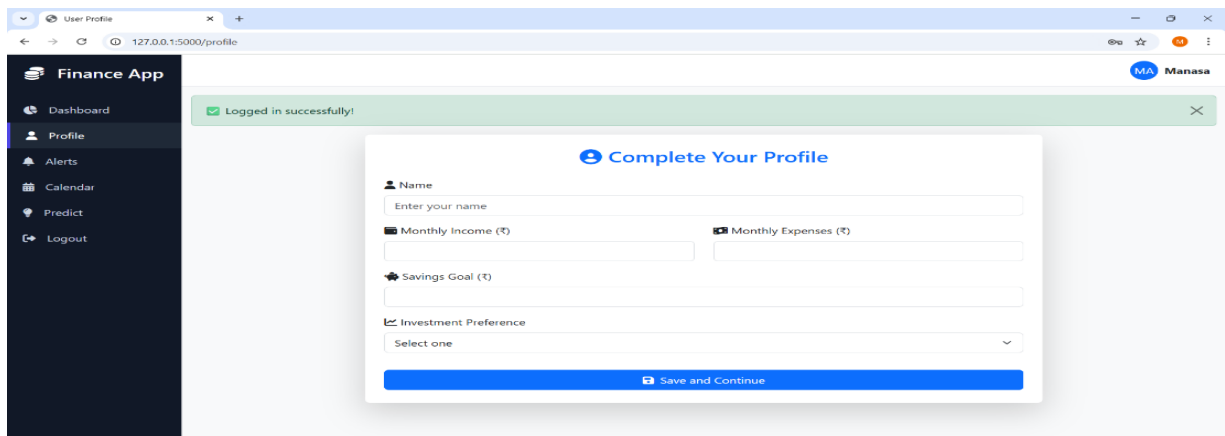


The screenshot shows a web browser window with the title 'Register' and the URL '127.0.0.1:5000/register'. The page features a dark blue sidebar on the left with the 'Finance App' logo and navigation links: Dashboard, Profile, Alerts, Calendar, Predict, and Logout. The main content area is light gray and contains a white registration form titled 'Register'. The form has two input fields: 'Email' and 'Password'. Below these fields is a blue 'Register' button. At the bottom of the form, there is a link that says 'Already have an account? Login'.

User Login and Dashboard

Once logged in, the user is directed to the dashboard, which displays the summary of financial inputs like income, expenses, savings, and alerts for upcoming bills or overspending.

Figure 2: Dashboard Showing Financial Summary and Alerts

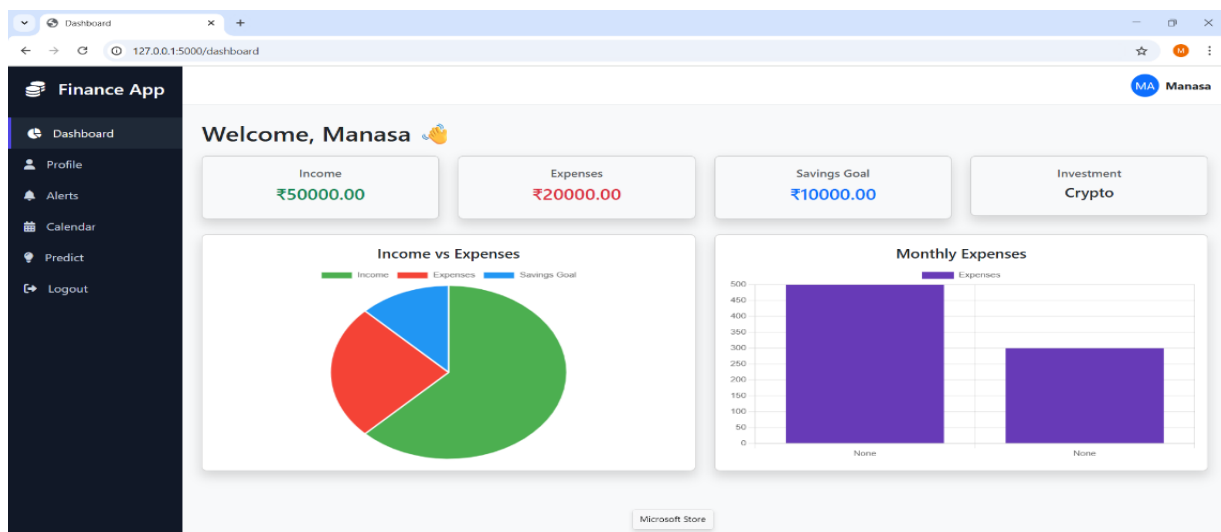


The screenshot shows a web browser window with the title 'User Profile' and the URL '127.0.0.1:5000/profile'. The page features a dark blue sidebar on the left with the 'Finance App' logo and navigation links: Dashboard, Profile, Alerts, Calendar, Predict, and Logout. The main content area is light gray and contains a white profile completion form titled 'Complete Your Profile'. The form has several input fields: 'Name' (with a placeholder 'Enter your name'), 'Monthly Income (₹)', 'Monthly Expenses (₹)', and 'Savings Goal (₹)'. There is also a dropdown menu for 'Investment Preference' with the text 'Select one'. At the bottom of the form is a blue 'Save and Continue' button. A green notification banner at the top of the main content area says 'Logged in successfully!'.

Financial Profile Submission

This form enables users to submit essential financial information including income, expenses, savings goals, and investment preferences. The data feeds the ML module and alert system.

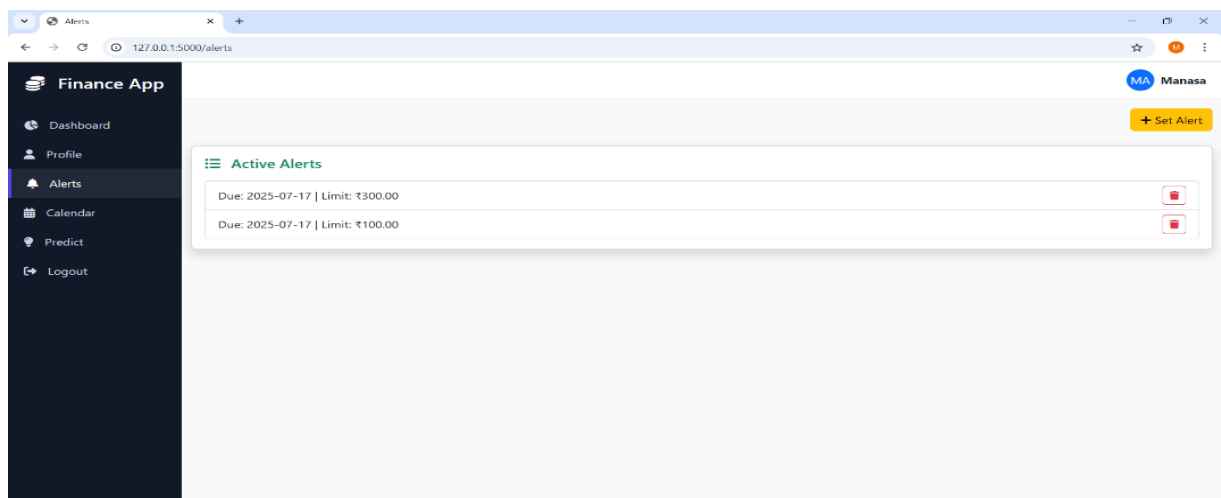
Figure 3: Financial Profile Form



Alert Configuration Interface

Users can define bill due dates and monthly spending limits. These thresholds trigger email alerts when violated.

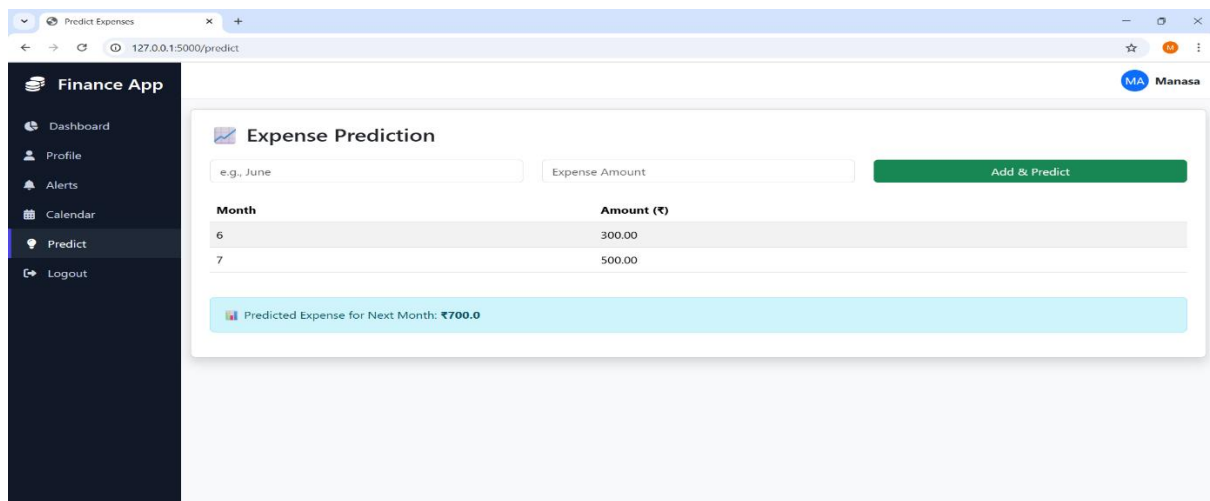
Figure 4: Alert Settings Page



Expense Prediction Using Machine Learning

A separate Predict page allows users to input monthly expenses. The system uses Linear Regression to forecast the next month's expenses and display the predicted value.

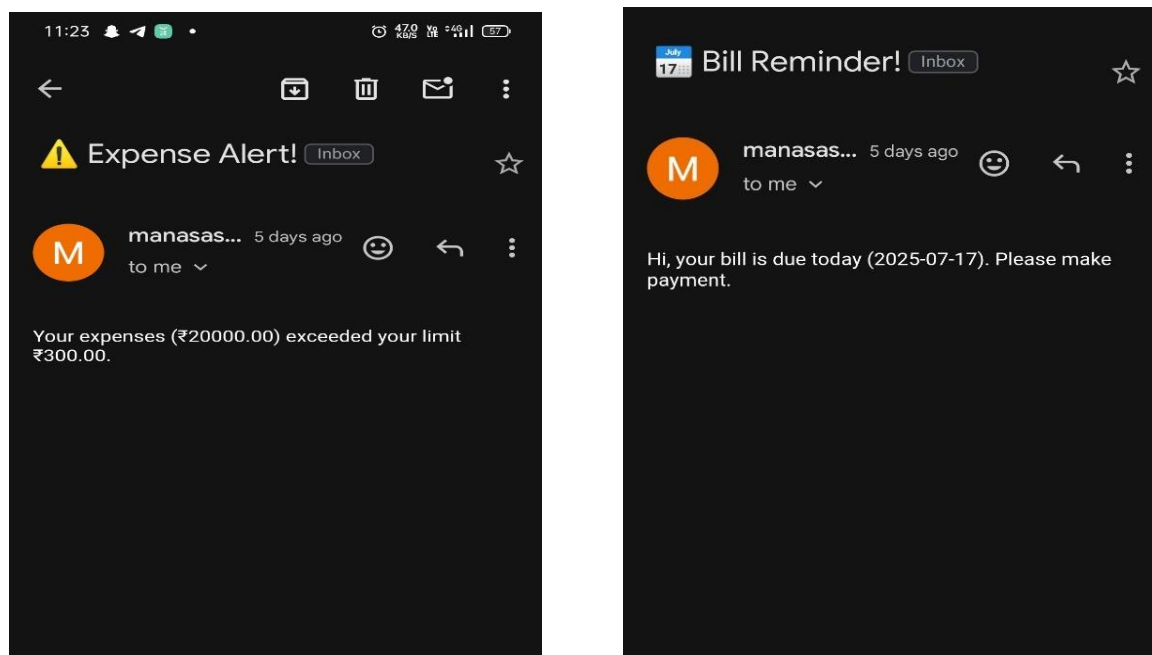
Figure 5: ML-Based Expense Prediction Page



Email Notification Example

If a bill is due or expense exceeds the limit, the user receives an email alert via Flask-Mail and SMTP configuration. This promotes proactive financial management.

Figure 6: Example of Email Notification Sent to User



7. CONCLUSION

This AI-based personal finance platform successfully addresses common challenges faced by individuals in managing their money. By integrating web technologies and machine learning, the system offers a practical solution for expense tracking, forecasting, and timely alerts. The user-centric dashboard and automation features simplify financial decision-making. Moreover, the project demonstrates how academic concepts—such as AI modeling, system design, and backend integration—can be applied in building a real-world application. The system is both functional and scalable, with ample scope for future innovation.

8. FUTURE SCOPE

- **Enhanced Prediction Models:** Implementation of more sophisticated ML algorithms such as LSTM, ARIMA, or Prophet for better accuracy.
- **Security Improvements:** Add support for hashed password storage, OAuth, and multi-factor authentication.
- **Mobile Platform:** Extend system accessibility by developing native or cross-platform mobile apps using Flutter or React Native.
- **Recurring Transactions:** Automate monthly bill reminders and recurring payments.
- **Calendar Integration:** Sync with services like Google Calendar to manage financial deadlines.
- **Data Import Options:** Allow users to upload and analyze bank statements.
- **Cloud Deployment:** Host the application on cloud platforms like AWS or Heroku for real-time access.

9. REFERENCES

- Flask Web Development by Miguel Grinberg – O'Reilly Media
- Learning MySQL by Seyed M.M. & Russell J.T. – O'Reilly
- Python Programming by John Zelle – Franklin, Beedle & Associates
- Flask Documentation – <https://flask.palletsprojects.com>
- MySQL Official Docs – <https://dev.mysql.com/doc/>
- Flask-Mail – <https://pythonhosted.org/Flask-Mail/>
- Bootstrap – <https://getbootstrap.com>
- Jinja2 – <https://jinja.palletsprojects.com>
- Chart.js – <https://www.chartjs.org/docs/>
- FullCalendar.js – <https://fullcalendar.io/docs>
- Stack Overflow – <https://stackoverflow.com>
- Flowgorithm – <https://flowgorithm.org>
- Towards Data Science – <https://towardsdatascience.com>
- Analytics Vidhya – <https://www.analyticsvidhya.com>