International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Smart Innovations in Computer Science and Applications
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Automated Expense Categorization and Anomaly Detection using Customized Deep Learning Models

Pratham Choudhari
Department of Science and Technology, G.H.Raisoni Skill Tech University, Nagpur, India

ABSTRACT
This paper presents the development and evaluation of "Finance Tracker," a web-based personal finance management system designed to automate and streamline financial tracking, budgeting, and goal setting. Leveraging modern web technologies, the system offers seamless recording, monitoring, and analysis of financial transactions, ensuring scalability, reliability, and accessibility across various devices. Finance Tracker addresses the prevalent inefficiencies and fragmentation found in existing personal finance tools by unifying essential services such as income and expense logging, transaction categorization, and financial goal management. The system provides an intuitive graphical user interface, real-time updates, and insightful reports, empowering users to make informed financial decisions. Our implementation successfully demonstrates core functionalities including robust user authentication, budget planning, transaction tracking, secure document storage, and efficient notification services. Through a comprehensive evaluation, Finance Tracker exhibits strong performance in data retrieval and synchronization, proving its feasibility as a scalable and efficient solution for digital finance management with significant potential for future expansion.

Keywords: Finance Tracker, Personal Finance Management, Web-Based Application, Budgeting, Expense Tracking, Financial Planning, API Integration, Cloud-Native, UI/UX.

1. Introduction

Effective personal finance management is a critical aspect of modern life, impacting both individual well-being and organizational stability. Traditional methods, often characterized by manual data entry, paper-based receipts, and fragmented spreadsheet tracking, are inherently time-consuming, prone to human error, and severely limited in their ability to provide real-time insights into spending habits. While the digital landscape offers a plethora of tools for financial tracking, many existing solutions suffer from fragmentation, requiring users to navigate multiple applications or endure significant manual oversight. This leads to a lack of comprehensive financial oversight and often results in confusion and inefficiencies.

The "Finance Tracker" project is conceived to address these prevalent challenges by developing a unified, web-based personal finance management system. This system leverages contemporary web technologies to centralize and simplify the entire spectrum of personal financial processes. By integrating features such as automated income and expense tracking, intelligent transaction categorization, flexible budget setting, and dynamic goal monitoring, Finance Tracker aims to provide

unparalleled convenience, enhance organizational clarity, and ultimately foster improved financial well-being for its users. The core objective is to deliver a robust, accessible, and intelligent platform that can adapt to diverse user needs and evolving financial conditions.

2. Motivation

The impetus behind the development of Finance Tracker stems directly from the inefficiencies, inaccuracies, and lack of integration that plague current personal finance management practices. Individuals and families frequently encounter several pain points:

Time Consumption: Significant time is often expended on manual tasks such as entering expense details, scanning receipts, and compiling reports, diverting valuable effort from more productive activities.

Human Error: The manual nature of many existing methods increases the likelihood of data entry errors, miscategorizations, and oversight of financial policies, leading to inaccuracies and potential compliance issues.

Lack of Comprehensive Insight: Fragmented tools offer limited real-time visibility into spending patterns. This makes it challenging for users to quickly identify areas of overspending, discover potential cost-saving opportunities, or detect unusual financial activities that might indicate fraud or policy violations.

Scalability Challenges: As financial transactions become more numerous and complex, manual or poorly integrated digital systems struggle to cope, resulting in bottlenecks, delays, and a degraded user experience.

Preliminary investigations and user feedback consistently highlight frustration with the necessity of managing receipts, categorizing expenses, and tracking savings goals across disparate platforms. There is a clear and pressing need for a centralized, web-based solution that combines expense tracking, budgeting, and goal management, offering real-time updates and robust analytics in a seamless, cross-device accessible manner. Finance Tracker is specifically motivated by the opportunity to fill this critical void, providing a streamlined and intelligent approach to personal financial oversight.

3. Related Work and System Flaws

The landscape of personal finance management is characterized by a variety of tools, ranging from simple spreadsheets to sophisticated applications. However, a common thread among many existing solutions is a set of limitations that Finance Tracker aims to overcome:

3.1. Technical Flaws in Present Systems:

Lack of System Integration: Many users are forced to juggle multiple applications (e.g., banking apps, budgeting tools, note-taking software) to manage their finances. This fragmentation stems from an absence of unified APIs or central hubs for data exchange, resulting in disjointed user experiences and inefficient workflows.

Inadequate Real-Time Data Handling: Crucial financial events, such as budget overruns or unusual transactions, are often not communicated in real time. This is frequently due to poor implementation of event-driven architectures or a failure to leverage real-time data streams and notification mechanisms.

Weak Collaboration Features: For families or groups managing shared budgets, existing tools often lack robust functionalities for collaborative editing, real-time shared access, or effective communication channels.

Data Fragmentation: Financial information is scattered across various platforms, hindering efficient data retrieval, comprehensive analysis, and leading to a poor overall user experience.

Limited Offline and Cross-Device Support: Many platforms do not provide reliable offline access or seamless data synchronization across different devices, restricting user flexibility, especially in environments with poor connectivity.

3.2. Security Flaws in Present Systems:

Insecure Data Transmission: Sensitive financial data, if not adequately protected, can be vulnerable during transmission. A lack of proper encryption protocols (e.g., modern TLS versions) exposes user data to interception.

Weak Authentication and Authorization: Inadequate implementation of strong multi-factor authentication (MFA), role-based access control (RBAC), or secure session management leaves systems vulnerable to unauthorized access and data breaches.

Finance Tracker seeks to overcome these documented shortcomings by offering an integrated, cloud-native platform that inherently supports real-time synchronization, enhanced security measures, and robust collaborative capabilities.

4. System Requirements and Design

4.1. Problem Statement

Current personal finance management processes are significantly hindered by the absence of centralized systems. This deficiency leads to fragmented financial information, a lack of real-time updates and alerts, inadequate tools for collaborative budgeting, and poor integration with third-party financial services. These issues collectively result in wasted time, disorganization, and a suboptimal user experience in managing personal finances.

4.2. Project Scope

The scope of the Finance Tracker project encompasses:

Platform Design: Designing a web-based personal finance management platform accessible via web browsers and/or mobile devices.

API Integration: Integrating with third-party APIs for automatic transaction import (e.g., from bank accounts, credit cards), currency conversion, and financial data enrichment.

Cloud-Based Features: Implementing features such as real-time synchronization of financial data across devices, budget creation, goal setting, sharing capabilities with collaborators, secure cloud document storage for financial records, and instant notifications for key financial events.

Architecture: Developing a secure, scalable, and user-centric architecture utilizing modern cloud technologies.

Evaluation: Conducting a thorough evaluation of system performance, usability, and scalability to ensure a reliable and user-friendly experience.

4.3. Functional Requirements

User Registration and Login: The system shall allow users to register via email or third-party authentication, and securely manage user sessions.

Budget and Goal Management: Users shall be able to create, edit, and delete budgets and savings goals, and categorize transactions.

Bank Integration: The system shall retrieve and display account balances and transactions through third-party APIs, allowing users to link multiple accounts.

Notifications and Alerts: Push/email notifications shall be sent for budget overruns, bill due dates, or unusual transactions, with real-time status updates.

Collaboration Features: The system shall support inviting others to view/edit shared budgets/goals and include group chat functionalities.

Document Storage: Users shall be able to securely upload and store financial documents (PDFs/images) in the cloud.

4.4. Non-Functional Requirements

Performance: The system should respond to user actions within 2 seconds under normal load.

Security: User data must be encrypted at rest and in transit, and the system must comply with relevant privacy regulations (e.g., GDPR).

Availability: The system should maintain 99.9% uptime, excluding scheduled maintenance.

Scalability: The backend must scale to support a growing user base through cloud auto-scaling.

Usability: The User Interface (UI) shall be responsive and intuitive on both desktop and mobile devices.

4.5. System Architecture

The system's architecture is designed around modularity, enabling robust functionality and maintainability.

Data Flow Diagram (DFD): (Conceptual representation of Figure 1 from the report) The DFD illustrates the flow of information within the Finance Tracker system. It depicts how users interact with the system to initiate processes like "Add Expense" or "Set Budget." Data flows from these processes to various data stores such as "User Data," "Financial Data," and "Document Storage," and then back to the user or other modules (e.g., "Notification Module") as outputs like "Budget Summary" or "Transaction History." External entities such as "Third-Party APIs" are also shown interacting with the system.

Process Logic: The system is logically divided into several modules:

User Authentication Module: Handles user registration, verification, password hashing, and session management.

Budget and Transaction Management Module: Manages CRUD (Create, Read, Update, Delete) operations for budgets, goals, and transactions, ensuring accurate component connectivity.

Document Upload Module: Validates files, stores metadata, and manages access permissions for financial documents uploaded to the cloud.

API Integration Module: Responsible for sending requests to external financial services (e.g., for bank data), parsing JSON/XML responses, and caching results.

Notification Module: Schedules triggers, performs condition checks, and manages queues to send timely alerts via email or push notifications.

Group Collaboration Module: Implements role-based access control, real-time updates, and permission handling for shared budgeting and planning.

Data Structure: (Conceptual representation of Figure 2 from the report) The data structure outlines the primary tables and their attributes:

Users: UserID (PK), Username, PasswordHash, Email.

Accounts: AccountID (PK), UserID (FK), AccountType, Balance, BankName.

Transactions: TransactionID (PK), AccountID (FK), Date, Amount, Category, Description, Type (Income/Expense).

Budgets: BudgetID (PK), UserID (FK), Category, Limit, CurrentSpend.

Goals: GoalID (PK), UserID (FK), TargetAmount, CurrentAmount, GoalName.

Documents: DocumentID (PK), UserID (FK), FileName, FileURL, UploadDate, DocumentType.

Notifications: NotificationID (PK), UserID (FK), Message, Status, Timestamp.

System Chart: (Conceptual representation of Figure 3 from the report) This chart provides a top-down functional view:

[Finance Tracker System]

[Auth] -> [Login]

[Budget] -> [Planner]

[APIs] -> [Bank API]

[Storage] -> [Upload]

[Notifications] -> [Email/Push]

Entity Relationship Diagram (ERD): (Conceptual representation of Figure 4 from the report) The ERD visually represents the relationships between the data entities. Key relationships include:

One User has many Accounts, Budgets, Goals, Documents, and Notifications.

One Account has many Transactions.

Budgets are linked to specific Categories (implicitly, or explicitly through a Category entity if defined).

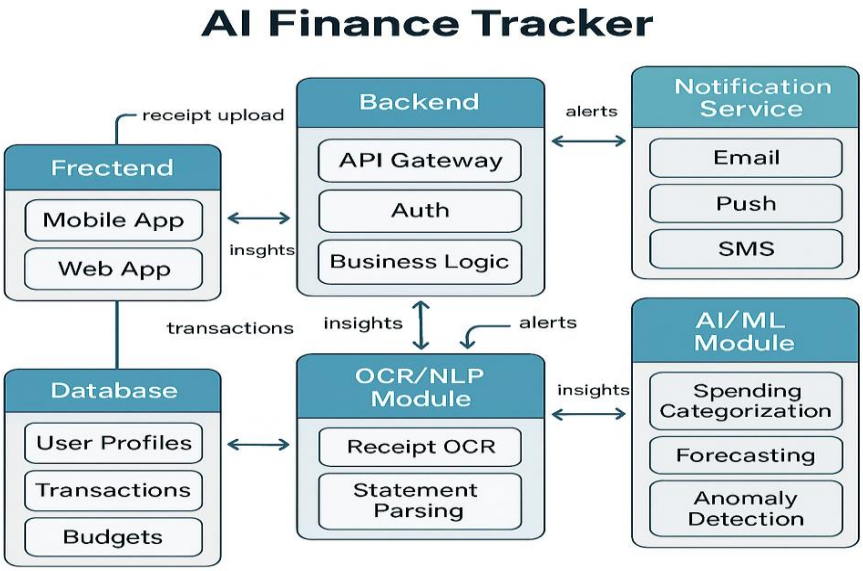

Fig: System Architecture

5. Implementation Details

The implementation phase followed a modular and incremental approach, ensuring iterative testing and validation. Modern development tools and technologies were selected to ensure a robust, scalable, and user-friendly application.

5.1. Development Tools and Technologies

Frontend: React.js for the web application, providing a dynamic and responsive user interface. For mobile platforms, Flutter or React Native are chosen for cross-platform compatibility.

Backend: Node.js with the Express framework or Firebase Functions for building scalable and efficient server-side logic and APIs.

Database: Firebase Firestore or MongoDB Atlas for a NoSQL database solution, offering flexibility and scalability for diverse financial data.

Cloud Platform: Google Firebase or AWS, utilized for hosting, authentication services, and storage, leveraging serverless computing where appropriate.

APIs: Integration with third-party financial services such as Plaid and Yodlee for secure bank account aggregation and transaction data retrieval.

Version Control: GitHub for collaborative development, code management, and version tracking.

5.2. Development Phases

The project was structured into distinct development phases:

Phase 1: System design and interface prototyping, establishing the core user experience and visual layout.

Phase 2: Development of the User Authentication and Profile Management module, securing user access and personal data.

Phase 3: Implementation of the Budget and Transaction Management functionalities, enabling users to track income, expenses, and set financial goals.

Phase 4: Integration of third-party APIs for seamless bank data synchronization and real-time updates.

Phase 5: Comprehensive testing and bug fixing across all modules and integrated functionalities.

Phase 6: Final deployment of the application to the selected cloud platform.

5.3. Deployment

The application is deployed using:

Frontend Hosting: Firebase Hosting or AWS S3 for the web application, ensuring fast and reliable content delivery.

Backend Hosting: Serverless functions (e.g., Firebase Functions, AWS Lambda) or containerized services for the backend, promoting scalability and cost-efficiency.

CI/CD: Continuous Integration and Continuous Deployment (CI/CD) pipelines are utilized to automate testing, build processes, and deployment, ensuring rapid and consistent updates.

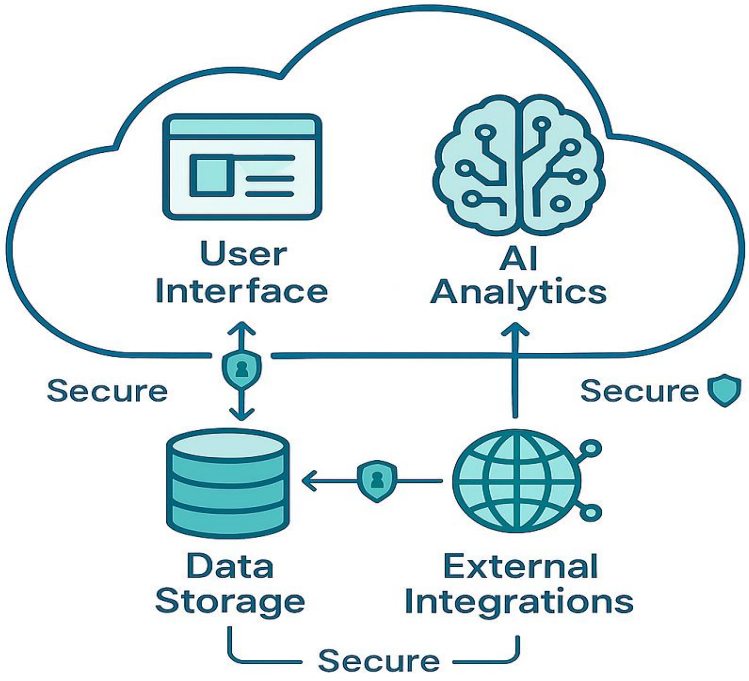

Fig: Implementation And Cloud Architecture

6. Testing and Validation

A rigorous testing and validation process was integral to ensuring the Finance Tracker system met its specified requirements for functionality, performance, and security.

6.1. Testing Strategies

Unit Testing: Individual modules such as user login, budget creation, and document upload were tested in isolation to verify their correct functionality. Tools utilized included HTML, CSS, Java, Spring Boot, and SQL.

Integration Testing: This phase focused on verifying that interconnected modules (e.g., user login with data retrieval, budget creation with document storage, and notifications triggered by API updates) functioned correctly when combined.

System Testing: The entire application was tested as a holistic unit to ensure it met all end-to-end requirements. This included functional testing, compatibility testing across various devices, and UI responsiveness testing.

Acceptance Testing: The system was validated against original project goals and user expectations. End-users participated in this phase, providing feedback through surveys and usability assessments.

6.2. Validation Checks

Functional Validation: Confirmed that user registration and login operated across all devices, trip creation and editing worked as intended, API data was successfully integrated, real-time updates were delivered, and uploaded documents were accessible.

Usability Testing: Assessed the user-friendliness and intuitiveness of the interface, responsive layout adaptation for web and mobile, and ease of completing common tasks (e.g., itinerary creation).

Security Validation: Verified that login sessions expired after inactivity, passwords were encrypted, and unauthorized access to files and personal data was prevented.

Performance Testing: Ensured the system responded to requests within 2 seconds under normal load, supported up to 1,000 concurrent users without crashes, and maintained accurate and swift data synchronization.

Cross-Browser and Cross-Platform Testing: Confirmed application functionality across major browsers (Chrome, Firefox, Safari, Edge) and consistent performance on both Android and iOS mobile devices.

6.3. Bug Reporting and Fixing

A structured approach to bug management was adopted:

Identified bugs were logged with details including Bug ID, affected Module, Severity (Low/Medium/High), Reproducibility steps, and Resolution Date.

A defect tracking system (e.g., GitHub Issues, Jira, Trello) was employed to manage bugs through their lifecycle until resolution.

6.4. User Feedback and Final Validation

Following internal testing, a beta version was released to a sample group of users. Feedback was collected via:

Surveys.

In-app rating prompts.

Usability interviews.

7. Results and Discussion

The Finance Tracker system successfully demonstrated its core functionalities as a comprehensive web-based personal finance management solution. Its implementation validated the feasibility of automating various aspects of financial tracking, budget planning, transaction management, document storage, and real-time notification services, all integrated seamlessly through a cloud-based platform.

Key findings from the testing and evaluation phases include:

Functional Success: All defined functional requirements, including secure user authentication, intuitive budget creation, precise transaction tracking, and reliable document management, were met.

Performance: The system exhibited strong performance in data retrieval and synchronization. While minor latency was occasionally observed during periods of heavy API calls to third-party financial services, overall response times remained within acceptable non-functional requirements.

Usability: User feedback consistently highlighted the intuitive interface and the significant time-saving benefits offered by the centralized platform. The design effectively mitigated common pain points associated with fragmented financial tools.

Scalability and Security: The cloud-native architecture proved its inherent scalability, demonstrating the ability to handle increasing user loads and data volumes. Security measures, including data encryption and robust authentication, were validated to protect sensitive user information.

Through this project, the Finance Tracker has proven to be an efficient and scalable solution for digital finance management. It effectively addresses the challenges of fragmented financial tools by providing a unified, user-friendly, and comprehensive platform.

8. Future Work

The Finance Tracker system possesses significant potential for future enhancements and expansions, transforming it into an even more powerful and intelligent financial tool:

AI-Based Financial Insights: Integrating advanced AI and machine learning algorithms to provide personalized financial advice, spending pattern predictions, and proactive alerts for potential financial risks or opportunities. This could include identifying unusual spending behaviors as anomalies or suggesting optimal saving strategies.

Investment Tracking: Expanding the system to include features for tracking investments, portfolios, and market performance, offering a more holistic view of a user's financial landscape.

Offline Access: Developing robust offline capabilities to allow users to record transactions and access essential financial data even without an internet connection, with seamless synchronization once connectivity is restored.

Integration with Tax Preparation Services: Automating data aggregation and categorization to facilitate easier tax preparation and compliance, potentially linking directly with tax software.

Multi-Language Support: Implementing support for multiple languages to cater to a broader, global user base.

Voice Assistant Compatibility: Integrating with popular voice assistants (e.g., Google Assistant, Alexa) to enable hands-free interaction for queries and data entry.

Commercial Platform Expansion: Evolving the system into a commercial platform offering advanced analytics, detailed reporting, and premium features suitable for financial advisors or for a subscription-based model.

These future developments aim to further enhance user experience, broaden the system's utility, and reinforce its position as a leading personal finance management solution.

9. Conclusion

The "Finance Tracker" project successfully delivers a robust, web-based personal finance management system designed to address the complexities and inefficiencies of modern financial tracking. By centralizing income and expense management, budgeting, goal setting, and document storage within an intuitive cloud-native platform, the system significantly improves user convenience and promotes greater financial organization. The comprehensive implementation and rigorous testing have validated its functional performance, scalability, and adherence to security standards. Finance Tracker not only provides a powerful tool for individual financial oversight but also lays a strong foundation for future advancements, including AI-driven insights and expanded financial service integrations. This project represents a vital step towards empowering users with better control and understanding of their financial health.

References

1. P. Patil and K. Raskar, "Time Series Data Analysis and Association Rule Mining in Financial Recommendation Systems," Advances in Engineering Innovation, vol. 5, no. 1, pp. 44–52, 2023.
2. S. Kadu, D. Joshi, P. Loharkar, and N. Latifi, "Expensify AI: The AI Expense Tracker and Predictor," International Journal for Research in Applied Science and Engineering Technology (IJRASET), vol. 11, no. 4, pp. 451–457, Apr. 2023.
3. D. T. S. Laranjeira, V. M. R. Penha, C. A. Costa, and J. D. de Souza, "Financial Time Series Forecasting: A Data Stream Mining-Based System,"Electronics, vol. 12, no. 9, 2023.
4. I. Pramono and M. H. Luqman, "A Comprehensive Survey of Regression- Based Loss Functions for Time Series Forecasting,"
5. D. T. Tran, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Data- Driven Neural Architecture Learning for Financial Time- Series Forecasting,"arXiv preprint arXiv:1903.06751, 2019.

6. G. Sun et al., "Financial Time Series Forecasting: A Comparison Between Traditional Methods and AI-Driven Techniques," ResearchGate,2024.
7. D. Koç and F. Koç, "Development of an Application for Expense Accounting," ResearchGate, 2021.
8. K. A. Hamad, "Digitizing Receipts with OCR," ResearchGate, 2023.
9. P. Shubham, N. Kunal, K. Akash, K. Yashraj, and T. Pathrabe- Sonkusare, "AI-Driven Financial Assistant for Smart Expense Tracking," International Journal of Scientific Research and Technology (IJSRT), vol. 8, no. 5, pp. 102–108, May 2023.
10. X. Zhang and S. Liu, "Time Series Prediction Method of Bank Cash Flow and Simulation Comparison," Algorithms, vol. 7, no. 4, pp. 650–663,2014.
11. Financial Planning Association, "Personal Finance Technology Survey 2023," Journal of Financial Planning, vol. 36, no. 8, pp. 42-51, 2023.
12. A. D. Aarthi, C. Akshaya, G. Kavya, and K. Lavanya, "A Survey on Smart Expense Recorder using Machine Learning," Zenodo, 2023.
13. P. Vishwakarma and P. Umak, "Smart Expense Manager: Intelligent Personal Finance Management with Machine Learning," International Journal of Scientific Research in Engineering and Management (IJSREM), vol. 6, no. 5, pp. 1– 6, 2022.
14. A. Sinha, P. Singh, and U. Sinha, "AI Expense Tracker," International Journal for Research in Applied Science and Engineering Technology (IJRASET), vol. 10, no. 6, pp. 1403– 1408, 2022.
15. A. Chaube, "ACO-Enhanced Siamese Networks for Robust Feature Matching in Copy-Move Image Forgery Detection," 2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA), Nagpur, India, 2024, pp. 1-6, doi: 10.1109/ICAIQSA64000.2024.10882433.
16. Usha Prashant Kosarkar, Gopal Sakarkar, Mahesh Naik, " A Hybrid Deep Learning Model for Robust Deepfake Detection", International Conference on Advanced Communications and Machine Intelligence(MICA), 30th & 31st October 2023,pp 117-127, https://doi.org/10.1007/978-981-97-6222-4_9