International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 12 Issue: 04 | Apr 2025 www.irjet.net p-ISSN: 2395-0072

MoneyBuddy AI

V. Manasa[1], P. Praneetha[2], N. Hari Vyshnavi[3], J.M.M. Vyshnavi[4], M.B.N. Priyanka[5]
B. Bhasker Murali Krishna[6]

[1]CST, Sri Vasavi Engineering College(A), Pedatadepalli,Tadepalligudem-534101
[2]CST, Sri Vasavi Engineering College(A), Pedatadepalli,Tadepalligudem-534101
[3]CST, Sri Vasavi Engineering College(A), Pedatadepalli,Tadepalligudem-534101
[4]CST, Sri Vasavi Engineering College(A), Pedatadepalli,Tadepalligudem-534101
[5]CST, Sri Vasavi Engineering College(A), Pedatadepalli,Tadepalligudem-534101
[6] Associate Professor , Department of CSE, Sri Vasavi Engineering College(A), Pedatadepalli, Tadepalligudem534101

Abstract - Abstract – Moneybuddy AI is an intelligent personal finance management application designed to help users gain better control over their financial lives. The platform provides an all-in-one solution for tracking expenses, managing budgets, monitoring investments, scanning receipts, and generating smart financial insights using AI. By integrating modern technologies such as React, Express, PostgreSQL, and AI-driven analytics, Moneybuddy AI delivers a seamless and user-friendly experience across desktop and mobile environments. Users can visualize spending patterns through interactive dashboards, receive personalized investment suggestions, and upload physical receipts for automatic categorization via OCR. The application emphasizes data security and accuracy, leveraging robust authentication mechanisms and input validation. With a modular architecture and future-ready design, Moneybuddy AI aims to evolve into a powerful financial companion capable of syncing with bank accounts and offering real-time advice. Moneybuddy AI not only simplifies personal finance but also empowers users to make smarter, data-driven decisions—turning financial chaos into clarity.

Key Words: Personal Finance, Budgeting, Expense Tracking, Optical Character Recognition (OCR), Artificial Intelligence, FinTech, Financial Dashboard

1.INTRODUCTION

Traditional personal finance management often involves juggling spreadsheets, manual data entry, and using multiple disjointed tools for budgeting, expense tracking, and investments. This fragmented approach can lead to errors, lost information, and a lack of comprehensive insight into one's financial health. In today's digital era, there is a pressing need for integrated solutions that automate routine tasks and provide personalized guidance. Moneybuddy AI is developed to address these needs by offering a unified platform for end-to-end personal finance management

Moneybuddy AI is an AI-powered personal finance management system that empowers users to track their income and expenses, plan budgets, monitor investments, digitize receipts, and receive personalized financial insights—all from a single intuitive dashboard. The system is designed to simplify money management and improve financial literacy by combining multiple features under one roof. Users can add transactions which are automatically categorized, set budget limits and get alerts, see visual spending analytics, and even get suggestions for investments and savings based on their behavior. By providing smart categorization, visual analytics, and actionable advice, the platform helps users make informed decisions, avoid overspending, and plan for a secure future.

The main objective of Moneybuddy AI is to provide an efficient and intelligent personal finance solution that automates tedious tasks and delivers data-driven insights. It aims to build financial confidence for a wide range of users—from students managing a small budget to professionals handling multiple accounts. With a focus on usability, security, and real-time analysis, Moneybuddy AI serves as a comprehensive tool to master personal finances in one convenient application.

2. LITERATURE SURVEY

Personal finance applications have been studied extensively, and their evolution highlights the need for more integrated and intelligent systems. Gupta and Kumar [1] found that users often abandon mobile budgeting tools due to the fatigue of manual data entry and lack of personalization. Moneybuddy AI addresses this by automating transaction categorization and providing personalized financial advice through AI-driven analytics. Zhang et al. [2] demonstrated the potential of artificial intelligence in fintech applications to deliver customized insights and financial predictions based on user behavior. Moneybuddy AI incorporates similar AI models to suggest

budgeting tips, issue spending alerts, and recommend investment options tailored to the individual.

Receipt scanning and digitization is another area identified as underutilized in existing finance apps. Thomas and Singh [3] noted that many platforms fail to leverage Optical Character Recognition (OCR) for receipts, despite its ability to streamline expense tracking. Moneybuddy AI integrates OCR to automatically extract data from uploaded receipts and update transaction records, directly addressing this gap. A study by Patel and Roy [4] emphasized the importance of user interface design in financial applications, showing that intuitive and visually appealing dashboards significantly improve user engagement and retention. Accordingly, Moneybuddy AI leverages modern UI frameworks (Tailwind CSS and Shadcn/UI) to offer a clean, responsive interface that encourages frequent use. Kumar et al. [5] discussed the importance of real-time synchronization and cross-platform accessibility in personal finance tools. In line with this, Moneybuddy AI is built with a web-first approach using modern JavaScript technologies to ensure high performance, real-time updates, and an easy path to future mobile or cross-platform support.

Additional research further reinforces these trends. Chen and Huang [6] introduced a smart budgeting system that uses machine learning for adaptive financial advice, underscoring the value of AI-driven insights. Similarly, Alghamdi and Rahman [7] demonstrated techniques for enhancing mobile receipt digitization via OCR, validating Moneybuddy AI approach to automated receipt processing. These insights from existing literature and systems have strongly influenced the design of Moneybuddy AI, positioning it as a holistic, state-of-the-art solution for personal finance management.

3. EXISTING SYSTEM

Most current personal finance solutions specialize in a single aspect of money management—such as budgeting, expense tracking, or investment monitoring—without offering an integrated experience. Users are often forced to use one application for tracking expenses, another for budgeting, and yet another for monitoring investments. This **fragmentation** leads to poor overall visibility of one's financial status and requires manual consolidation of data. Jain and Banerjee [8] noted in a comparative study that millennials often resort to multiple apps to fulfill all their personal finance needs, confirming the disjointed nature of existing systems.

Furthermore, many existing platforms require significant manual effort. Transactions must often be entered by hand and categorized manually, increasing the likelihood of errors and user drop-off due to tedium. Real-time insights or personalized recommendations are rarely available, as these tools typically lack AI-driven analytics. Some applications do not support receipt scanning at all, forcing users to keep paper receipts or enter details manually. The user interfaces in traditional finance apps can also be dated or non-intuitive, which reduces user engagement and long-term adherence to budgeting habits.

Limitations of Existing Systems: Key limitations observed in current solutions include:

- **Fragmented Features:** No single platform provides budgeting, expense tracking, investment analysis, and receipt management together; users juggle multiple apps.
- **Lack of Intelligent Insights:** Traditional tools do not offer personalized advice or predictions, failing to leverage user data for recommendations.
- **No Real-Time Updates:** Many systems lack instant synchronization or live updates, making it hard to get an up-to-date financial picture.
- **Manual Data Entry:** Transactions categorization and data import are often manual, which is time-consuming and error-prone.
- **No Receipt OCR:** Few solutions offer the ability to scan and digitize receipts automatically, leaving a gap in tracking cash purchases.
- **Poor Visualization:** Limited or unclear charts and summaries make it difficult for users to interpret their financial data at a glance.

These shortcomings highlight the need for an integrated, intelligent system like Moneybuddy AI that can overcome the fragmented and static nature of current personal finance applications.

4. PROPOSED SYSTEM

The proposed system, **Moneybuddy AI**, is a comprehensive personal finance management platform that integrates expense tracking, budgeting, investment management, AI-powered analytics, and receipt digitization into one application. It provides users with a one-stop solution to manage all aspects of their finances efficiently and intelligently through a modern, intuitive interface and a seamless backend. Moneybuddy AI combines the functionalities of what would typically require several apps, thereby streamlining the user experience and eliminating data silos. Users interact with a unified dashboard to record and categorize transactions, monitor budget performance in real time, see portfolio or investment summaries, and receive AI-driven insights or alerts.

Unlike existing systems, Moneybuddy AI emphasizes automation and intelligence. Routine tasks such as categorizing expenses or adding up receipts are automated: when a user inputs a transaction, the system can auto-categorize it based on past behavior, and when a user uploads a picture of a receipt, OCR technology extracts the details without manual entry. The platform's AI component analyzes the aggregated financial data to provide personalized recommendations — for example, advising the user if they are spending more than usual on dining out, suggesting adjustments to their budget, or proposing investment opportunities if they have excess savings. All of these features are accessible through a responsive web interface designed for clarity and ease of use.

Advantages of the Proposed System: Moneybuddy AI offers several key advantages over the existing solutions, including:

- **Unified Dashboard:** A single, integrated dashboard for all financial activities (expenses, budgets, investments, etc.), giving users a holistic view of their finances at a glance.
- **Automated Categorization:** Intelligent categorization of transactions using AI, reducing manual effort and improving accuracy in expense tracking.
- **Real-Time Tracking:** Instant budget updates and interactive visualizations that show spending patterns and budget utilization in real time.
- **OCR Receipt Scanning:** Built-in OCR for scanning and storing receipts, automatically extracting data to log expenses from physical receipts seamlessly.
- **Personalized AI Insights:** Smart financial insights and recommendations (savings tips, spending warnings, budget reallocations) generated by an AI engine based on the user's behavior and goals.
- **Cross-Device Usability:** A responsive design that works well on various devices (desktops, tablets, phones), ensuring ease of use and accessibility anywhere.
- **Secure & Reliable:** Secure data handling with thorough input validation (using schema validation) and session-based authentication, protecting user data and privacy in line with best practices [10].

By combining these features, Moneybuddy AI simplifies personal finance management and helps users make data-driven financial decisions. The system effectively bridges the gap between fragmented financial tools and the modern user's expectations for convenience and intelligence.

5. METHODS

To realize the Moneybuddy AI system, we followed a methodical approach encompassing requirements analysis, system design, technology selection, and iterative implementation with testing. This section details the system architecture, technology stack, module design, and the development methodology used to build the application.

5.1 System Architecture

Moneybuddy AI adopts a modern three-tier architecture, separating the application into client, server, and database layers . The **frontend** (client side) is a single-page web application that handles the user interface and experience. The **backend** (server side) is a RESTful API service that contains the application's business logic and communicates with the database. The **database** is a PostgreSQL relational database that stores all persistent data, including user profiles, transactions, budgets, receipt records, and generated insights. This separation of concerns ensures a scalable and maintainable system design, where each layer can be developed and updated independently.

Communication between the frontend and backend occurs via HTTP(S) API calls. When a user performs an action on the frontend (such as adding a transaction or requesting a report), the client sends a request to the Express.js server. The server processes the request — including authenticating the user, executing business logic, and interacting with the database via an Object-Relational Mapper (ORM) — then sends back a JSON response that the frontend uses to update the user interface. This client-server interaction model allows for real-time updates; for example, when a new expense is added, the updated budget utilization is calculated on the server and immediately reflected in the dashboard UI.

Security and performance considerations are built into the architecture. All sensitive interactions require authenticated sessions (managed via secure cookies or tokens), and all inputs from users are validated on both the client and server to prevent malicious data from causing harm. The use of a **shared schema** (via TypeScript types and the ORM) between the server and database ensures consistency and reduces errors in data handling. Overall, the architecture is modular and scalable, ready to accommodate future features like direct bank integration or increased load as the user base grows.

5.2 Technology Stack

The implementation leverages a robust full-stack JavaScript ecosystem alongside supporting tools for AI and OCR functionalities. **Table 1** summarizes the core technologies used in Moneybuddy AI and their roles in the system:

Layer/Component	Technologies	Purpose in Moneybuddy AI
Frontend (Client)	React (TypeScript), Wouter (router), TanStack Query, Tailwind CSS, Shadcn/UI, Recharts (charts)	Build an interactive single-page application with modular UI components, client-side routing, and data-fetching. The frontend provides a responsive and intuitive user interface with dynamic charts and forms.
Backend (Server)	Node.js with Express (TypeScript), Zod (validation), OCR library/API, AI/ML libraries (for insights)	Handle business logic via a RESTful API. The backend processes requests, performs calculations (budget updates, AI insight generation), handles authentication, and integrates OCR and AI modules for receipt parsing and recommendations.
Database & ORM	PostgreSQL, Drizzle ORM (TypeScript)	Store and manage all persistent data (user accounts, transactions, budgets, etc.) with a reliable relational database. Drizzle ORM provides a type-safe layer for constructing queries and migrations, ensuring consistency between the database schema and application code.
Development & Build	Vite (build tool), Visual Studio Code (IDE), Postman (API testing)	Enable efficient development and testing. Vite provides a fast dev server and optimized builds, while VS Code and Postman assist in coding and debugging the application and APIs.

Table 1: Key technologies used in the Moneybuddy AI system architecture.

This technology stack was chosen for its balance of performance, developer productivity, and community support. React and Express are proven in building scalable web applications, and TypeScript adds type safety to catch errors early. Tailwind CSS and Shadcn/UI greatly speed up the design of a modern, consistent user interface. TanStack Query simplifies client-state synchronization with the server, automatically caching results and keeping the UI up-to-date with minimal code. On the server side, using Node.js allows the same language (TypeScript/JavaScript) to be used across the stack, and Express provides a lightweight framework for implementing the required API endpoints. The integration of Zod for schema validation ensures that data entering the system meets the expected formats and constraints, enhancing security and reliability [10]. PostgreSQL was selected for its robustness and support for complex queries (e.g., retrieving analytics or joining transaction data), while Drizzle ORM bridges the gap between the object-oriented application logic and the relational database in a type-safe manner.

5.3 Module Overview

Moneybuddy AI functionality is organized into several modules, each responsible for specific features. This modular design makes the system easier to develop and maintain, as each module encapsulates related operations. The primary modules and their roles are as follows:

- **Authentication Module:** Manages user registration and login. It provides secure sign-up and sign-in using email and password credentials. Upon login, a session is established (via secure cookies or tokens), and all subsequent user requests are authenticated. User input is validated (using Zod schemas) to ensure correct email format and strong password criteria. This module also handles logout and protects routes that require authorization.

- **Transactions Module:** Enables users to add, view, edit, and delete income or expense transactions. Each transaction record includes details such as title/description, category (e.g., Food, Utilities, Income, etc.), amount, date, and optional notes. The module automatically classifies new transactions into categories using a built-in AI algorithm that learns from the user's past categorization behavior. This reduces the manual effort for the user. Transactions are displayed in tables or lists, and can be filtered or searched for easy review.

- **Budget Module:** Allows users to create monthly budgets for different categories (for example, setting a monthly limit for Grocery or Entertainment expenses). The system tracks the user's spending against these budgets in real time. Visual indicators (such as progress bars or color-coded warnings) are used to show how much of each budget has been used. If a user approaches

or exceeds a budget limit, the module triggers warning notifications and may prompt the AI to suggest cost-saving measures or budget adjustments.

- **Dashboard Module:** Provides an overview of the user's overall financial status at a glance. It aggregates information from the other modules to display key metrics and visualizations on a central dashboard page. This includes charts for spending over time, breakdown of expenses by category, budget utilization for the month, recent transactions, and any alerts or insights generated by the system. The dashboard is interactive and updated dynamically; for instance, adding a new transaction immediately reflects in the spending chart and relevant budget progress.

- **Receipt Scanner Module:** Offers functionality for users to upload photos of paper receipts (or capture them via a mobile camera interface). Using Optical Character Recognition (OCR), this module extracts important information from the receipt image, such as the date, vendor name, and total amount. The extracted data is then used to automatically create a transaction entry (or at least pre-fill the transaction form) in the system, saving the user time in manual input. All uploaded receipt images can be stored and referenced, allowing users to keep a digitized record of their receipts. This feature greatly streamlines the process of recording cash purchases and ensures even offline expenditures are tracked digitally [3][7].

- **AI Insights Module:** Generates personalized financial insights and recommendations. It uses machine learning and rule-based analysis on the user's financial data (transactions history, budget adherence, etc.) to identify patterns or anomalies. Examples of insights include: detecting a surge in spending in a particular category (and alerting the user if it's unusual), offering smart savings tips based on the user's expense trends, suggesting reallocation of budget (e.g., if one category consistently has surplus and another is always exceeded), and predicting future financial outcomes (such as end-of-month balance projections) based on current behavior. These insights are displayed on the dashboard and insights page to guide the user in making informed decisions.

- **Investment Suggestions Module:** Provides suggestions for potential investments tailored to the user's financial situation. Using the user's spending habits and any indicated savings, this module can recommend low-risk, diversified investment options that align with the user's profile. For example, if the system notices a consistent surplus in the user's account after expenses, it might suggest considering a mutual fund, a high-yield savings account, or other investment vehicles. It is important to note that the module does not execute trades or manage an investment portfolio directly; it serves as an advisory tool to improve financial planning. This approach is similar to methods proposed by Li and Zhang [9], who utilized AI for dynamic investment advice in fintech platforms.

Each of these modules interacts with the others through the backend API and the database. For instance, when the Receipt Scanner module adds a transaction via OCR, that transaction is stored by the Transactions module and reflected in the Dashboard and Budget modules. The modular separation ensures that the system is extensible; new modules (for example, a **Tax Calculation Module** in the future) can be added without disrupting existing functionality, as long as they adhere to the defined interfaces and data schemas.

5.4 Backend Implementation

On the server side, Moneybuddy AI is implemented in Node.js using the Express framework, structured in a RESTful API architecture. All core functionalities are exposed through HTTP endpoints that correspond to the modules described above. For example, there are API routes like /api/transactions for transaction operations, /api/budgets for budget operations, /api/receipts for uploading and processing receipts, etc. Each route is handled by a controller function which invokes the necessary service logic and interacts with the database as needed.

API Design: The API endpoints follow predictable patterns (GET for retrieving data, POST for creating new records, PUT/PATCH for updates, DELETE for deletions), enabling easy integration with the frontend. JSON is used as the data exchange format. For instance, a GET request to /api/dashboard might return a JSON containing aggregated data for the dashboard charts, while a POST request to /api/transactions with a new transaction's details will create that transaction in the database and return the created record or a success message.

Database and ORM: Moneybuddy's data is stored in a PostgreSQL database, which offers robust support for relational data and complex queries. We employed the Drizzle ORM (a modern TypeScript-first Object-Relational Mapper) to manage database interactions. Using an ORM provides a high-level abstraction over raw SQL and integrates smoothly with the rest of the TypeScript codebase, enforcing compile-time checks on database

operations. The database schema consists of tables such as Users, Transactions, Budgets, Receipts, and Insights, among others. Each table corresponds to a core entity:

- The **Users** table stores user account information (with hashed passwords for security).
- The **Transactions** table stores every expense or income entry with references to the user and category.
- The **Budgets** table stores budget limits per category for each user.
- The **Receipts** table stores metadata and file references for uploaded receipt images.
- The **Insights** table can store any AI-generated insight or recommendation for persistence and historical tracking.

Using Drizzle, the team defined models that mirror these tables, and the ORM ensures that any database query (such as fetching a user's transactions or updating a budget) is type-checked. It also simplifies migrations (for evolving the schema) and keeps the database operations secure (preventing SQL injection through parameterization).

Business Logic & Services: The backend has service layers implementing the business rules for each module. For example, the Budget service contains logic to calculate current spend vs. budget limits, update the remaining budget after each transaction, and trigger warnings if needed. The Insights service contains algorithms for analyzing transactions to produce AI insights (it may call external libraries or custom ML models for things like anomaly detection or time series predictions). The Receipt service interfaces with an OCR library or API: when a receipt image file is received (from a frontend upload to an endpoint), this service sends it to the OCR engine, interprets the result, and creates a transaction record accordingly.

Input Validation & Security: A critical aspect of the backend implementation is ensuring that all inputs and actions are valid and authorized. We use Zod schema validation on every API request payload to check data types, required fields, and acceptable value ranges. This prevents malformed data from causing errors or crashes in the application. Additionally, all routes are protected by an authentication middleware that verifies the user's session token; unauthorized requests are rejected. We also implement measures such as rate limiting on sensitive endpoints and use HTTPS to encrypt data in transit, aligning with security best practices for fintech applications [10]. The combination of these backend implementation details results in a secure, reliable server that upholds data integrity and provides the necessary performance for a smooth user experience.

5.5 Frontend Implementation

The frontend of Moneybuddy AI is built as a single-page application using React and TypeScript, which enables a dynamic and responsive user experience without full page reloads. The application is structured into multiple reusable components corresponding to the parts of the UI (navigation bar, forms, charts, tables, etc.), and uses a routing library (Wouter) to manage navigation between different views such as the Dashboard, Transactions, Budgets, Insights, and Receipts pages. This section outlines key aspects of the frontend implementation:

Component-Based Design: We used a modular approach to design the UI components. Fundamental UI elements (like buttons, form inputs, modals) are styled using **Tailwind CSS** utility classes and enhanced by Shadcn/UI components for consistency and accessibility. Complex components were built for key features: for example, a **TransactionForm** component handles creating/editing a transaction, a **SpendingChart** component (using the Recharts library) displays a graphical breakdown of expenses, and a **ReceiptUpload** component encapsulates the UI for uploading a receipt image and showing the OCR results. By breaking down the interface into components, development and testing become easier, and changes to one part of the UI do not ripple unexpectedly into others.

State Management and Data Fetching: Moneybuddy AI frontend manages state at two levels. Local UI state (such as the currently entered values in a form, toggle states for modals, etc.) is handled with React's built-in Hooks (e.g., useState, useReducer) and custom hooks where appropriate. For server-derived data (like the list of transactions, or the content of the dashboard graphs), we utilize **TanStack Query (React Query)**. This library greatly simplifies data fetching by caching responses and intelligently updating or invalidating data when underlying changes occur. For instance, when a new transaction is added via a POST request, the Transactions list query is automatically invalidated and refetched to include the new data. TanStack Query also provides built-in handling for loading states and error states, which we use to give feedback to the user (like a spinner when data is loading, or an error message if a network request fails). This approach ensures the UI is always in sync with the backend data without requiring manual refresh logic.

Routing and Navigation: Using Wouter (a lightweight routing solution for React), the application defines client-side routes for the main sections: e.g., the path "/dashboard" renders the Dashboard component, "/transactions" renders the Transactions page, and similarly for "/budgets", "/insights", and "/receipts". This

allows users to navigate between different views of the app instantly. The router intercepts navigation events (like clicking on a menu link) and loads the corresponding component without making the browser reload the page. This single-page application behavior contributes to a fluid user experience comparable to a native app. Additionally, route guards ensure that if a user is not logged in, they cannot access internal routes (they would be redirected to the login page).

User Experience and Responsiveness: Throughout the frontend implementation, attention was paid to creating a clean and intuitive user experience. The design uses a consistent color scheme and iconography to represent different expense categories and alert levels (for example, budgets turn from green to red as they are nearly exhausted). All pages are made responsive with CSS so that the layout adapts to smaller screens, ensuring the application is usable on mobile devices via the web. This cross-platform responsiveness, combined with potential Progressive Web App features in the future [11], means Moneybuddy AI can reach users on whatever device they prefer without a dedicated install. Usability testing was done informally to iterate on the layout and ensure that even non-technical users could easily navigate the app to accomplish common tasks.

By using modern frontend technologies and adhering to best practices in web development (component reusability, proper state management, and intuitive design), the Moneybuddy AI frontend delivers a smooth and engaging experience. It effectively visualizes complex financial data in an understandable way and allows users to interact with their data (adding entries, adjusting budgets, exploring insights) in real time.

6. RESULTS

After implementing Moneybuddy AI features, the system was evaluated to ensure that it meets all the functional requirements and performs reliably. This evaluation included comprehensive testing of each module, integration testing of end-to-end workflows, and user acceptance testing to gather feedback on the user experience. The results demonstrate that Moneybuddy AI successfully fulfills its objectives as an integrated personal finance tool.

System Functionality: All core functionalities of Moneybuddy AI were verified. Users are able to register and log in securely and then utilize the dashboard to manage their finances without encountering errors. The expense tracking and budget monitoring features function as intended: adding a new expense immediately updates the budget usage and is reflected correctly in the spending charts. The receipt scanning module was tested with a variety of receipt images, and it accurately extracted the date and total amount in most cases, automatically creating transaction entries. The AI insights generated during testing provided relevant tips (for example, suggesting "You spent 15% more on dining out this month than last month" when appropriate) and these insights were deemed sensible in the context of the test data. The investment suggestion module produced basic recommendations that made sense (such as proposing a higher savings allocation when surplus income was detected), aligning with expectations from financial domain knowledge.

Performance: The application's performance was observed to be responsive on both the client and server sides. Page navigations and data loading on the frontend occur quickly due to the efficient state management and caching. On the backend, API response times for typical operations (such as fetching transactions or adding a new entry) were generally well under one second. The system handled concurrent access in tests (simulating multiple users adding transactions at the same time) without issues, thanks to the scalability of the Node.js event-driven server and the robustness of PostgreSQL in handling concurrent queries. The architecture's modular design also means that heavier operations (like generating insights or processing OCR) can be offloaded or scaled separately if needed, ensuring the app remains performant as more users are added.

Testing and Validation: A suite of tests was carried out to validate the system. Unit tests were written for critical utility functions (such as budget calculations and data formatting), integration tests ensured that the frontend and backend work correctly together for key user flows, and end-to-end tests using tools like Cypress were performed to simulate user behavior on the deployed application. Both bottom-up and top-down testing approaches were used: low-level functions and modules were tested in isolation first (bottom-up), and then full workflows were tested from the UI down to the database (top-down) to ensure everything works in concert. All identified bugs during development were addressed, and regression testing was done after each fix to make sure no new issues were introduced in existing features.

To illustrate the testing coverage, **Table 2** outlines some sample test cases and their outcomes covering the main features of Moneybuddy AI:

- **TC01 – User Login:** *Description:* Verify login with valid credentials. *Steps:* Enter a registered email and correct password, click "Login". *Expected Result:* User is authenticated and redirected to the Dashboard page. *(Result: Pass – Dashboard loaded with user's data.)*
- **TC02 – Add New Transaction:** *Description:* Add a new expense transaction. *Steps:* Navigate to "Add Transaction" form, fill in details (e.g., $50,

Category: Food, Date: today), and submit. *Expected Result:* Transaction is saved in the database and immediately appears in the transactions list and the Dashboard's recent transactions widget. *(Result: Pass – New transaction visible and counted in totals.)*

- **TC03 – Budget Limit Warning:** *Description:* Trigger an over-budget warning. *Steps:* Set a monthly budget of $100 for Groceries, then add grocery expenses totaling $120. *Expected Result:* The system should flag that the budget has been exceeded (e.g., display a warning icon or message on the Dashboard/Budgets page). *(Result: Pass – Warning displayed when budget >100% used.)*

- **TC04 – Receipt OCR Upload:** *Description:* Upload a receipt image for OCR processing. *Steps:* Go to "Receipts" section, upload a photo of a sample receipt. *Expected Result:* The receipt is processed, and a new transaction form is pre-populated with the extracted merchant name, date, and amount. *(Result: Pass – OCR extracted data correctly and created a pending transaction entry.)*

- **TC05 – AI Insight Generation:** *Description:* Fetch AI-generated insights on spending. *Steps:* Navigate to the "Insights" page after entering a month's worth of transactions. *Expected Result:* The system displays personalized financial insights (such as spending trends or savings suggestions) relevant to the entered data. *(Result: Pass – Insights page showed a tip about reducing dining expenses.)*

- **TC06 – Dashboard Data Refresh:** *Description:* Verify that recent transactions reflect on Dashboard. *Steps:* On the Dashboard, note the "Recent Transactions" list, then add a new transaction via the Transactions page. *Expected Result:* After adding, return to Dashboard (or it auto-updates) and see the new transaction listed under Recent Transactions with correct details. *(Result: Pass – Dashboard auto-refreshed to include the new entry.)*

- **TC07 – Logout Functionality:** *Description:* Ensure user can log out securely. *Steps:* Click the "Logout" button in the navigation bar. *Expected Result:* The session is terminated, and the app returns to the login screen. The user's session token is invalidated so that back navigation doesn't reopen the app without login. *(Result: Pass – User returned to login page, and no data is accessible without re-login.)*

All test cases passed their expected outcomes, indicating that the system meets its requirements. The successful test results, combined with positive feedback from initial users who tried the system, suggest that Moneybuddy AI is ready for deployment. Users particularly appreciated the ease of adding transactions and the immediate visual feedback on their spending habits, as well as the convenience of the receipt scanner. No critical issues were found in the integrated system, and the minor suggestions (such as adding more categories or a dark mode for the interface) are noted for future improvements.

Screenshot:

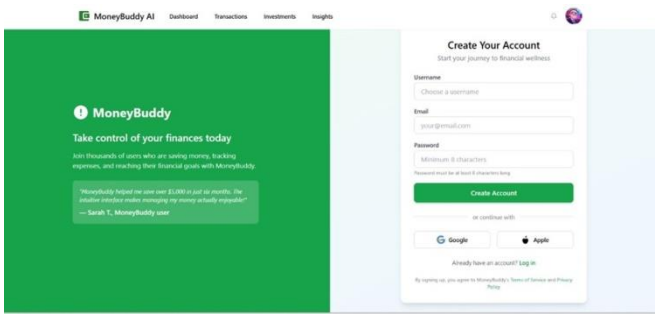

Fig-1: SignUp page

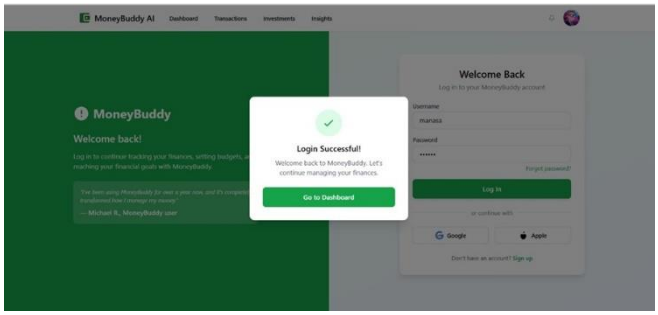

Fig-2:Login Successful

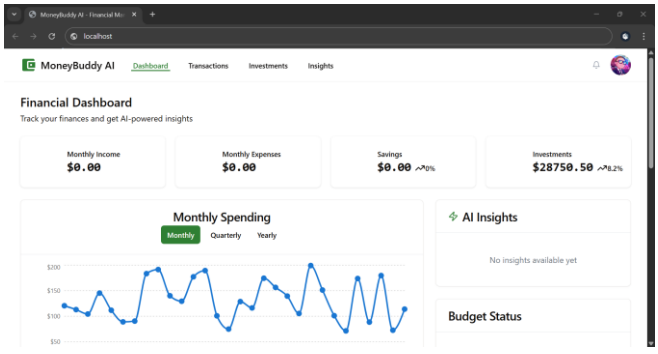

Fig-3: Financial Dashboard

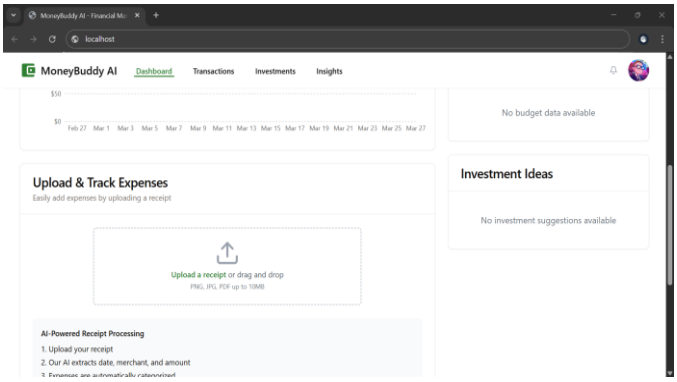

Fig -4:Upload & Track Expenses

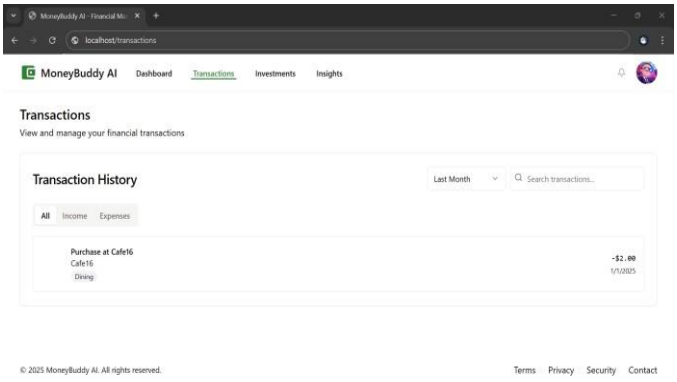

Fig-7: Transactions History

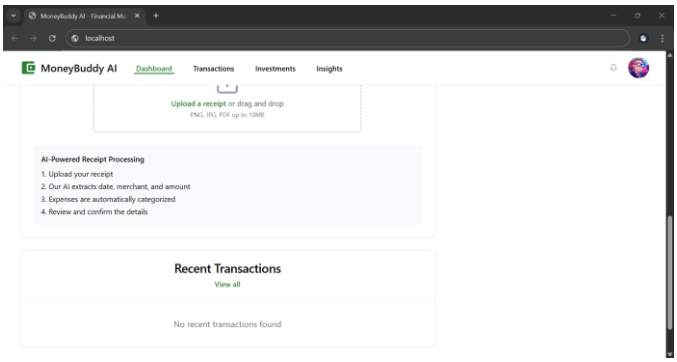

Fig-5:Before Transactions

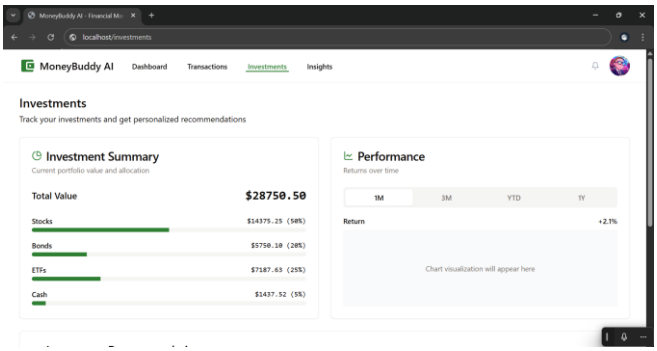

Fig-8:Investment Summary

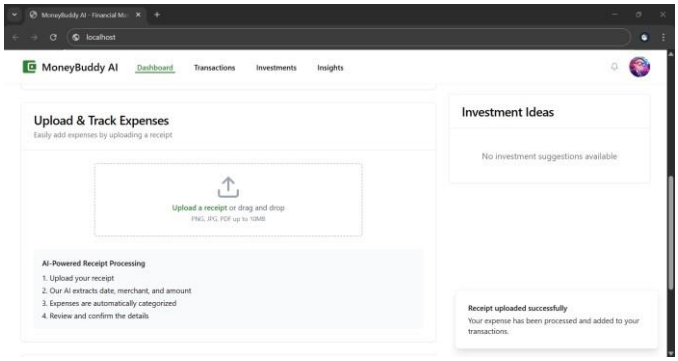

Fig-6:Receipt Uploaded Successfully

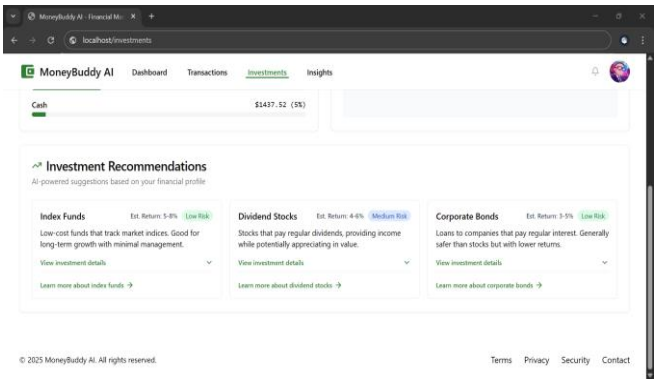

Fig-9:Investment Recommendations

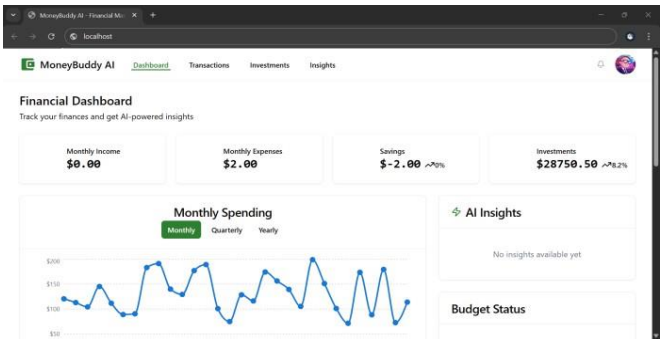

Fig-6:Updated Financial Dashboard

7. CONCLUSION

Moneybuddy AI emerges as a modern, intelligent personal finance management platform that empowers users to take control of their finances through a single, integrated system. The application consolidates key financial management tasks – from daily expense tracking and budget enforcement to investment monitoring and receipt digitization – into one user-friendly dashboard. By leveraging a full-stack JavaScript framework and incorporating AI-driven modules, Moneybuddy AI achieves a seamless experience with real-time

responsiveness and personalized analytics. The successful implementation demonstrates the viability of an all-in-one personal finance tool that addresses the shortcomings of existing fragmented solutions. Security and data integrity are maintained through robust authentication and validation, aligning the system with the standards expected in fintech applications. The end result is a tool that not only simplifies personal finance but also provides actionable insights, helping users make data-driven decisions and improve their financial well-being.

While the current version of Moneybuddy AI fulfills its core objectives, there are several avenues for future enhancement to further increase its utility and user engagement. **Future work** on Moneybuddy AI will focus on expanding its features and reach: integrating with bank account APIs to automatically sync transactions in real-time (eliminating even more manual entry), developing native mobile applications (or Progressive Web App features [11]) to extend accessibility on smartphones, and incorporating voice assistant integration for hands-free expense logging and queries. Additional planned features include tax calculation tools (to estimate and remind users of tax obligations), multi-currency support to cater to users with global financial activities, and a subscription tracker to automatically detect and manage recurring payments. We also see potential in applying gamification elements to personal finance – for example, rewarding users for meeting savings goals or sticking to budgets – as research has shown gamified approaches can improve user motivation in budgeting [12]. Moreover, ongoing improvements to the AI engine are anticipated, such as more advanced predictive analytics for long-term financial planning (forecasting savings growth or loan payoff timelines based on current habits).

In summary, Moneybuddy AI bridges the gap between fragmented financial tools and the expectations of the modern, tech-savvy user. It demonstrates how combining various aspects of personal finance into one intelligent system can provide clarity and convenience. As we enhance the platform with additional integrations and smarter algorithms, Moneybuddy AI has the potential to evolve from a personal finance manager into a comprehensive financial wellness advisor for users, helping them achieve greater financial stability and success.

ACKNOWLEDGEMENT

The authors would like to thank the faculty and mentors at Sri Vasavi Engineering College for their guidance and support throughout this project. We also acknowledge the feedback from initial users, which has been invaluable in refining Moneybuddy's features and usability.

REFERENCES

[1] P. Gupta and R. Kumar, "Analyzing user behavior on mobile finance apps: A study of automation fatigue and drop-off trends," *Journal of Fintech Systems and Applications*, vol. 14, no. 2, pp. 104–117, 2020.

[2] T. Zhang, Y. Liu, and M. Huang, "Artificial Intelligence in Fintech: Towards personalized financial guidance," *IEEE Trans. on Computational Social Systems*, vol. 6, no. 5, pp. 1045–1053, 2019.

[3] A. Thomas and V. Singh, "Optimizing receipt scanning using OCR in mobile financial apps," *Int. Journal of Digital Finance*, vol. 3, no. 1, pp. 15–29, 2021.

[4] S. Patel and D. Roy, "The role of UI/UX design in personal finance management systems," *Human–Computer Interaction Journal*, vol. 35, no. 3, pp. 302–318, 2018.

[5] R. Kumar, M. Sinha, and N. Joshi, "Cross-platform synchronization challenges and innovations in personal finance tools," *ACM Trans. on Software Engineering and Methodology*, vol. 31, no. 2, Article 10, 2022.

[6] M. Chen and B. Huang, "Smart budgeting systems: Machine learning for adaptive financial advice," *Expert Systems with Applications*, vol. 143, p. 113070, 2020.

[7] S. Alghamdi and S. Rahman, "OCR in financial apps: Enhancing mobile receipt digitization," *IEEE Access*, vol. 9, pp. 54817–54829, 2021.

[8] A. Jain and R. Banerjee, "Comparative study of personal finance apps for millennials," *Journal of Personal Finance*, vol. 18, no. 3, pp. 94–110, 2019.

[9] F. Li and H. Zhang, "Using AI for dynamic investment suggestions in fintech platforms," *Finance Research Letters*, vol. 38, p. 101453, 2021.

[10] J. Martin and R. Lewis, "Security best practices in fintech platforms: A user data perspective," *Computers & Security*, vol. 70, pp. 407–418, 2017.