FinTrack: An AI-Powered Personal Finance App for Predictive Budgeting and Financial Guidance

Final Year Project Report: Development of an AI-Driven Personal Finance Application

Mariam Mohammed

Student ID: 2389169

BSc Artificial Intelligence with Computer Science
Final Year Project

Supervisor: Dr. Nabeel Khan
Second Marker: Dr. Qamar Natsheh

School of Computer Science
College of Engineering and Physical Sciences
University of Birmingham

Academic Year 2023–24

Abstract

Personal finance management is still difficult for many people due to the limited predictive and personalized capabilities of current digital budgeting tools. These tools usually only track transactions in the past without offering any useful insights into the future, which hinders the ability to make proactive financial decisions. This dissertation addresses this gap through the development and rigorous evaluation of FinTrack, an innovative AI-driven personal finance platform leveraging sophisticated predictive analytics via the XGBoost algorithm to forecast user spending patterns from historical financial data. FinTrack integrates an AI-powered chatbot (FinBot), utilizing Google's Gemini API to deliver real-time, personalized financial guidance and budgeting recommendations, supported by a robust backend architecture employing Flask for API management and Firebase for secure user authentication. A comprehensive comparative analysis demonstrated that XGBoost outperformed alternative methods such as Long Short-Term Memory (LSTM) in accuracy, computational efficiency, and stability, especially with structured financial datasets. User testing further validated FinTrack's effectiveness, revealing significant improvements in financial decision-making and user engagement compared to traditional apps like Mint and YNAB. These findings demonstrate the considerable potential of integrating AI-driven predictive analytics into fintech solutions, establishing FinTrack as a meaningful advancement that directly addresses critical gaps in existing literature and sets a benchmark for future fintech developments aimed at enhancing financial literacy and empowering users to make smarter, data-informed financial decisions.

Acknowledgements

I would like to express my deepest gratitude to Dr. Nabeel Khan, my supervisor, for his constant guidance, encouragement, and insightful feedback throughout this project. I would also like to thank Dr. Qamar Natsheh for serving as the second marker—her words and feedback from the project demonstration have been especially helpful and encouraging.

I am especially thankful to Dr. Ahmed Ibrahim for his encouragement and support throughout my academic journey, which played a meaningful role in the successful completion of this project.

I am grateful to the University of Birmingham and the School of Computer Science for providing the resources and environment necessary for the successful completion of this work.

Special thanks to Zidan for testing the FinTrack application, and to all survey participants who offered valuable feedback.

Lastly, I am sincerely thankful to my family and friends for their unwavering support, patience, and encouragement throughout this journey.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Introduction to Personal Finance Management

Effective management of personal finances is crucial for achieving financial stability and securing long-term economic goals. However, despite its significance, many individuals, particularly students, young professionals, and budget-conscious individuals, struggle to manage their finances effectively. Traditional methods often rely heavily on manual tracking or basic digital tools that offer limited proactive guidance. Such conventional approaches generally lack predictive capabilities, making it difficult for users to anticipate future financial scenarios or adapt proactively to changing economic circumstances (Bhattacharjee et al. 2024).

1.2 Fintech and Its Growing Importance

Financial technology, or fintech, represents the integration of technology into financial services aimed at improving accessibility, efficiency, and personalization. Fintech has disrupted traditional financial services by introducing innovative solutions leveraging artificial intelligence (AI), big data analytics, and real-time processing capabilities.

These advancements offer significant improvements over traditional financial methods by providing real-time financial insights, personalized recommendations, and enhanced user experiences (Iyelolu & Paul 2024). The rapid growth and global adoption of fintech solutions highlight their transformative potential, as illustrated by the steady increase in fintech firms across global regions from 2018 to 2024 (see Figure 1.1) (DealPotential 2024).

Nevertheless, existing fintech apps often remain limited to basic functionalities, including simple expense tracking and historical transaction reporting.

These limitations point to a clear gap in offering personalized predictive insights—a vital feature for effective personal financial management (Gomber et al. 2017). Furthermore, mastering personal financial well-being requires more than just data visualization; it necessitates intelligent integration of budgeting, savings, debt management, and literacy, as conceptualized in the foundational framework presented in Figure 1.2.

Figure 1.1: Global Growth of FinTech Market by Region (2018–2024), showing the number of fintechs worldwide by region (DealPotential 2024).

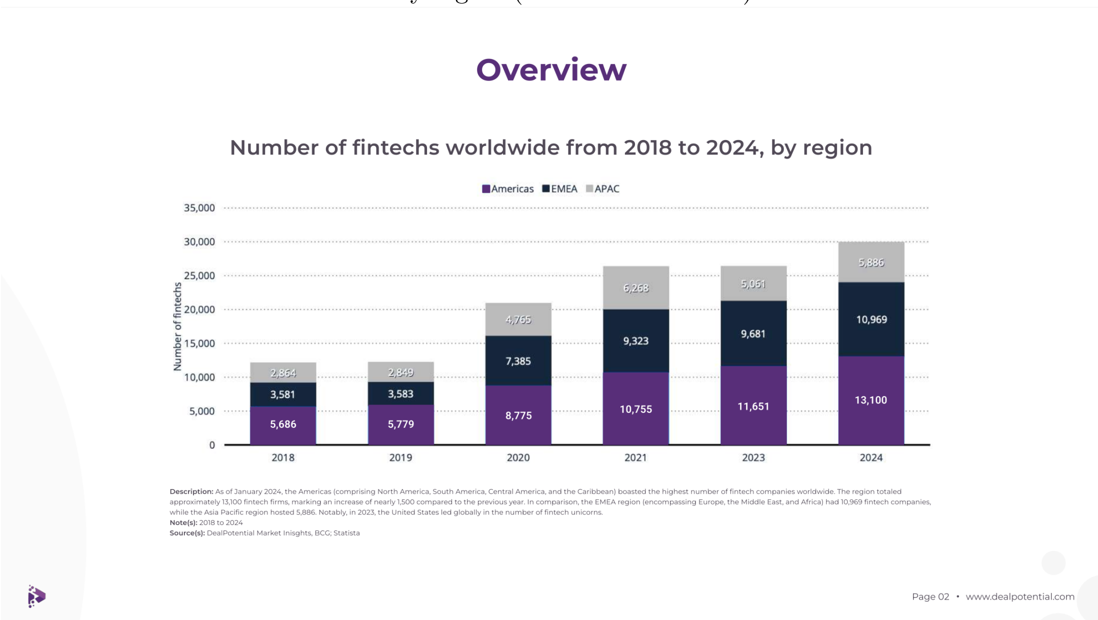
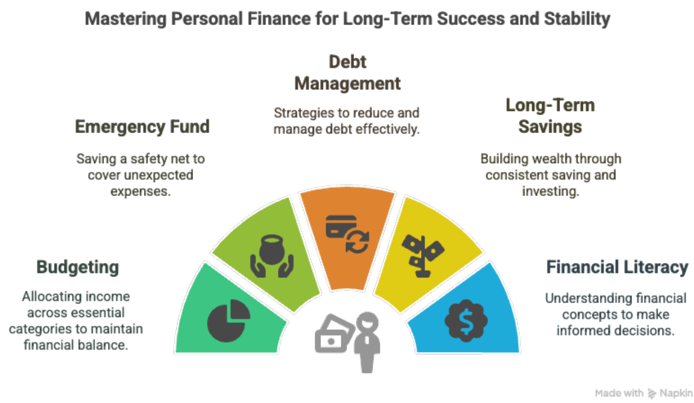


Figure 1.2: Importance of Personal Finance Management in Achieving Long-Term Goals



1.3 Motivation: Challenges of Traditional Finance Apps

While traditional finance apps like Mint and YNAB have improved expense tracking and visualization, they primarily provide reactive insights—displaying historical financial data without robust predictive analytics. Users typically encounter limitations such as inadequate forecasting of future spending, lack of real-time financial advice, and insufficient customization in budget management. These apps predominantly offer basic visualizations and transaction summaries, failing to adequately engage users or support proactive financial decision-making.

Notably, Intuit announced the discontinuation of Mint in early 2024, citing the company's decision to integrate select features into its Credit Karma platform (Intuit 2023). Identified shortcomings include the absence of predictive budgeting capabilities, limited visualization options (e.g., detailed pie charts for spending categories and time-framed analytics), inadequate debt-tracking functionalities, and insufficient real-time AI-driven guidance. Consequently, users often struggle to maintain consistent engagement with these platforms, ultimately undermining their financial goals.

1.4 The Need for AI Integration in Budgeting Platforms

Integrating artificial intelligence into budgeting platforms can address critical gaps identified in traditional finance apps. AI technologies, particularly predictive analytics and intelligent chatbots, can significantly enhance budgeting effectiveness, financial literacy, and user engagement. By proactively forecasting future expenses, providing real-time personalized guidance, and continuously adapting to user behavior, AI-driven platforms offer substantial advantages over conventional budgeting methods.

For instance, (Cai et al. 2021) highlight that AI systems can enhance consumer outcomes by delivering personalized financial advice and facilitating smarter decision-making processes. Additionally, (Bayakhmetova et al. 2025) emphasize that AI-driven tools can reduce psychological biases in financial behavior, offering personalized recommendations that promote responsible financial management.

1.5 Aims and Objectives

The main aim of this dissertation is to develop and critically evaluate *FinTrack*, an innovative AI-powered personal finance and budgeting platform designed specifically to enhance financial literacy, simplify budgeting processes, and empower smarter financial decision-making among students, young professionals, and budget-conscious users globally.

Specific objectives include:

Objective 1: Intuitive Budgeting

- Develop user-friendly, visually appealing analytics (graphs, pie charts) to clearly illustrate income, expenses, and spending categories.
- Implement advanced predictive analytics (using XGBoost) to accurately forecast user spending in various time frames, helping users plan proactively and efficiently.

Objective 2: AI Guidance

- Deploy an AI-powered chatbot (FinBot) leveraging Google's Gemini API to provide real-time personalized financial advice, answering user queries promptly and effectively.
- Ensure FinBot delivers reliable, general financial insights and recommendations based on established best practices in personal finance.

Objective 3: Improved Decision-Making

- Enhance user capability to make informed and proactive financial decisions through comprehensive stock market analytics, debt management features, and predictive spending insights.
- Aim to indirectly measure improvement in decision-making effectiveness through user engagement and qualitative feedback.

Objective 4: Automation and Real-World Integration

- Implement a native Android SMS parsing feature that automatically extracts transaction data from bank messages and integrates it into the user's ledger.

1.6 Research Questions

To comprehensively evaluate the effectiveness and innovation of FinTrack, this dissertation addresses the following research questions:

- How effectively does the FinTrack platform, with its integrated XGBoost predictive analytics and real-time financial tracking, improve users' budgeting outcomes and spending forecast accuracy?
- How does the inclusion of a conversational AI chatbot (FinBot) for real-time financial advice, alongside intuitive debt management and automated transaction input (e.g. SMS parsing), influence user experience and decision-making in personal finance management?
- What are the innovative contributions of the FinTrack platform's integrated features (predictive spending analytics, AI-driven advice, real-time stock tracking, and automated debt management) to personal finance management, and how do these combined functionalities compare to existing personal finance tools in enhancing financial planning?

CHAPTER 2

Literature Review

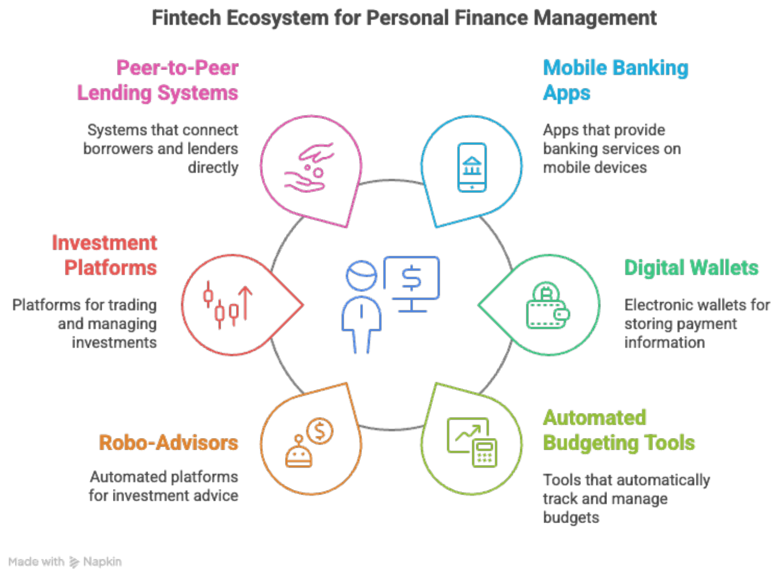
2.1 Overview of Fintech and Personal Finance Management

Financial technology, widely known as fintech, refers to the application of technology to enhance, automate, and improve the efficiency and accessibility of financial services. Over recent years, fintech has witnessed exponential global growth, dramatically reshaping the financial landscape and significantly impacting the delivery and consumption of financial services (Buckley et al. 2016). Technological innovations such as digital payments, peer-to-peer lending, blockchain, and robo-advisors exemplify fintech's transformative potential, fundamentally changing how individuals and businesses manage their finances (Nicoletti 2017).

Traditionally, personal finance management relied heavily on manual processes such as cash budgeting, manual transaction recording, and simple spreadsheet-based tracking. This traditional approach offered limited proactive capabilities and often resulted in delayed or inefficient financial decisions. However, the emergence of fintech solutions has revolutionized personal finance management by introducing automated tools and real-time insights. Mobile banking apps, digital wallets, and automated budgeting applications now provide users with immediate access to their financial information, simplified transaction tracking, and enhanced control over their financial decisions (Lee & Shin 2017).

This transformation is effectively illustrated in Figure 2.1, which shows how fintech technologies, through interconnected digital infrastructure, empower end-users with tools for financial autonomy, efficiency, and decision-making support.

Figure 2.1: Fintech-driven personal finance management ecosystem (adapted from Lee & Shin, 2018)



One of fintech's most profound disruptions in personal finance has been democratizing financial services previously accessible only to high-net-worth individuals or institutions. Tools such as robo-advisors, automated investment platforms, and digital budgeting applications have expanded financial inclusion, enabling a broader demographic to access sophisticated financial management resources. This inclusivity not only benefits individual users by improving financial literacy and decision-making but also contributes positively to broader economic stability and growth (Demirgüç-Kunt et al. 2018). Recent studies also highlight the strategic role of Generative AI in driving innovation and enhancing performance within the FinTech sector (Gowda & Gowda 2024).

Despite the evident benefits and rapid adoption of fintech solutions globally, several persistent challenges remain prevalent in personal finance management. Low financial literacy continues to pose significant barriers, restricting effective engagement with available technological resources. Studies indicate that large segments of populations worldwide lack essential financial knowledge, significantly undermining the effectiveness of fintech applications designed to promote informed financial behaviors (Lusardi & Mitchell 2014). Recent findings from the (for Economic Co-operation & Development 2023) reveal that foundational financial literacy levels are low among adults, including younger, digitally active individuals. This emphasizes the necessity for educational tools like FinTrack that can enhance financial understanding and decision-making.

Maintaining user engagement is a major hurdle for many fintech platforms. Despite advanced features, a lack of personalization and predictive insights often causes users to lose interest, limiting the platform's ability to support lasting financial behavior change.

Most apps offer static summaries rather than forecasting tools, preventing users from managing their finances proactively. Without predictive guidance, these tools fail to deliver strategic value.

Privacy and data security also remain top concerns. Users expect strong protection and transparency when handling sensitive information. Without this trust, even the most feature-rich platforms struggle to retain users long term. Addressing these challenges through innovative fintech solutions—specifically leveraging advanced predictive analytics, personalized interactions, robust security measures, and improved financial literacy resources—remains crucial. These elements form the foundation of effective, proactive, and user-centered personal finance management tools, guiding future fintech developments and improvements.

2.2 Personal Finance Applications

Recent studies reinforce fintech's growing impact on personal finance applications. (Bhattacharjee et al. 2024) highlight its transformative role in enhancing efficiency, accessibility, and cost-effectiveness, calling for platforms that provide personalized, forward-looking financial guidance—an area FinTrack aims to address. Similarly, (Iyelolu & Paul 2024) examine how fintech disrupts traditional banking by introducing agile, customer-focused services that challenge outdated models. (Tanda & Schena 2019) expand on this by exploring the evolving business models of fintech startups and the competitive pressures they face in scaling innovative solutions.

In addition to structural disruption, personalization has emerged as a key differentiator. A report by (Cloudester 2025) emphasizes that AI-driven customization significantly improves user engagement by delivering contextually relevant suggestions. (Omarini 2017) further demonstrates the value of real-time analytics and automation in building stronger digital customer relationships. Collectively, these findings highlight the growing expectation for fintech tools that go beyond transaction tracking—offering intelligent, personalized, and proactive financial support, which FinTrack is designed to fulfill.

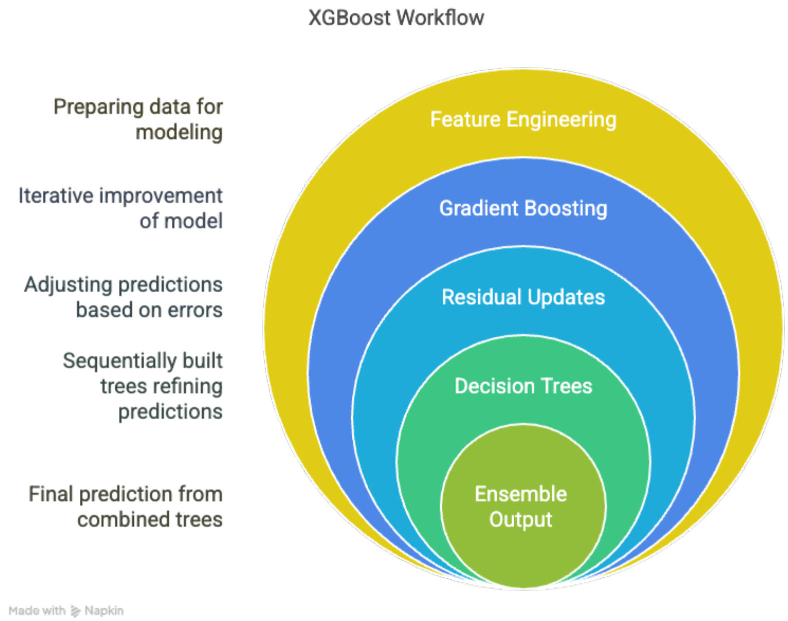
2.3 Time-Series Forecasting Literature

Predictive analytics plays a critical role in modern personal financial management, enabling users to make informed decisions by forecasting future spending patterns, budgeting needs, and savings goals. Time-series forecasting models help extract meaningful patterns from historical financial data to support proactive financial planning. Among the most prominent techniques used in this domain are XGBoost (Extreme Gradient Boosting) and Long Short-Term Memory (LSTM) neural networks. Both models have shown strong results in financial contexts, though they differ significantly in complexity, interpretability, and suitability depending on the structure and size of the dataset.

XGBoost in Personal Finance Forecasting

XGBoost, introduced by (Chen & Guestrin 2016), is a scalable, tree-based ensemble learning algorithm that builds on gradient boosting principles. It has become a leading model in financial forecasting due to its high computational efficiency, predictive accuracy, and ability to handle structured tabular data, including those with missing values and categorical variables. In the context of personal finance, datasets typically consist of small, structured, and transactional records—an environment where XGBoost performs exceptionally well. (Khandani et al. 2010) pioneered the use of machine learning in consumer finance by demonstrating that gradient boosting models significantly outperform traditional credit-scoring techniques. Their findings laid the foundation for the widespread adoption of such models in financial prediction tasks.

Figure 2.2: XGBoost workflow showing input processing, sequential tree boosting, and final ensemble output for financial forecasting.



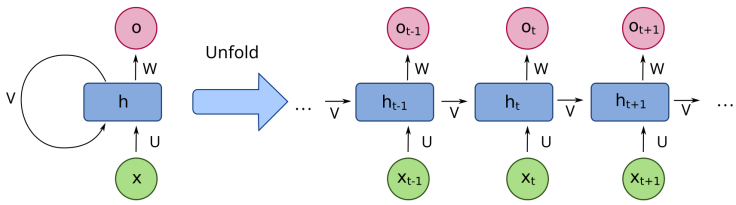
As illustrated in Figure 2.2, XGBoost begins by ingesting structured financial inputs, such as user transaction histories. After feature selection and engineering, the model applies a series of boosted decision trees, each learning from the residuals of the previous one. This iterative refinement enables highly accurate forecasting even with modest data volumes. (Nielsen 2016) highlighted the effectiveness of XGBoost in direct response modeling tasks, noting that with appropriate hyperparameter tuning, the algorithm achieved significantly improved predictive performance over baseline models such as logistic regression. (Zou et al. 2022) demonstrated that XGBoost delivers high accuracy and strong generalization performance on small, structured datasets. In their comparative study, the XGBoost model significantly outperformed deep neural networks (DNN) and support vector regression (SVR), highlighting its effectiveness in scenarios where data is limited—such as in personal finance applications.

LSTM in Sequential Financial Data

LSTM (Long Short-Term Memory) networks, proposed by (Hochreiter & Schmidhuber 1997), are a specialized type of Recurrent Neural Network (RNN) designed to capture long-range dependencies in sequential data. They have proven effective in domains with continuous and high-frequency data, such as stock market prediction, macroeconomic trend analysis, and energy consumption forecasting.

LSTMs are particularly suited for problems where past values influence future outcomes over long time horizons. However, their black-box architecture, high computational overhead, and requirement for large datasets present challenges for small-scale financial applications. In personal finance scenarios, where users typically have only 3–6 months of transaction history, LSTM models are vulnerable to overfitting and struggle with the lack of interpretability needed to build user trust (Siami-Namini et al., 2018). (Siami Namini et al. 2018). Moreover, they often demand deep learning frameworks like TensorFlow or PyTorch and access to GPUs, which may limit their usability in lightweight web apps such as FinTrack.

Figure 2.3: Simplified LSTM architecture unrolled over time, showing input sequences, hidden states, and outputs across multiple time steps.



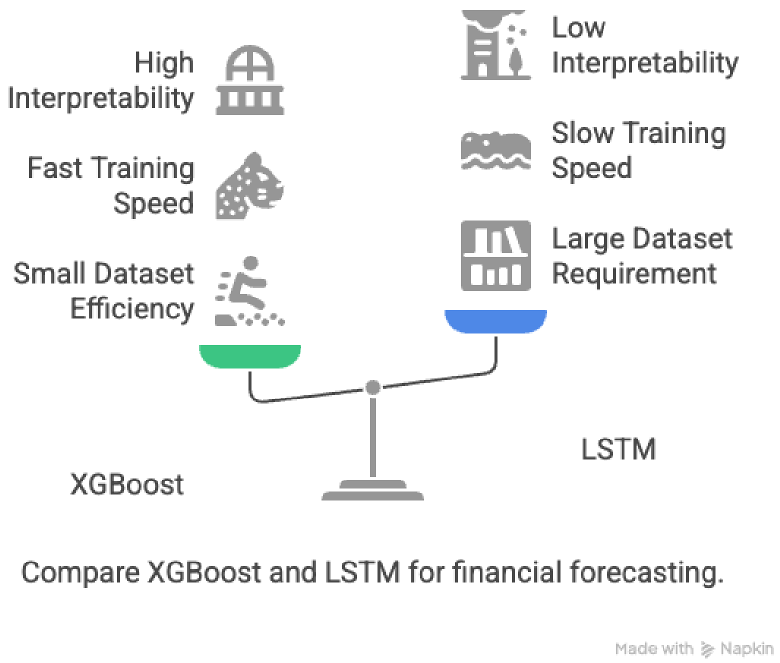
LSTMs extend traditional recurrent neural networks (RNNs) by incorporating gated mechanisms—specifically the input, forget, and output gates—that regulate the flow of information and preserve long-term dependencies through a memory cell. This architecture is illustrated in Figure 2.3, which shows how LSTM models process input sequences over time to maintain memory of prior states.

However, these sophisticated mechanisms also increase model complexity and reduce interpretability. In scenarios with limited user data and a need for transparency—such as FinTrack—simpler, faster models like XGBoost offer a more suitable and effective solution. Figure 2.4 provides a qualitative comparison between LSTM and XGBoost, highlighting XGBoost's advantages in explainability, resource efficiency, and integration simplicity for financial forecasting tasks in FinTrack. This aligns with broader trends in machine learning, where explainability is often prioritized over complexity in high-stakes domains like finance. As Molnar (2025) notes, model interpretability significantly enhances user trust and model adoption in financial decision-making contexts.

Table 2.1: Summary of Time-Series Forecasting Models Referenced in the Literature

Authors	Year	Model(s)	Application Domain	Dataset Type
Chen & Guestrin	2016	XGBoost	General financial forecasting	Structured, tabular
Khandani et al.	2010	Gradient Boosting	Consumer credit risk modeling	Structured credit data
Nielsen	2016	XGBoost	Direct response prediction	Tabular with categorical features
Zou et al.	2022	XGBoost, DNN, SVR	Small-scale finance forecasting	Small, structured datasets
Hochreiter & Schmidhuber	1997	LSTM	Sequential pattern modeling	Long, continuous sequences
Siami-Namini et al.	2018	LSTM	Financial/economic time series	Small and large sequential data

Figure 2.4: Qualitative analysis between XGBoost and LSTM for financial forecasting in FinTrack.



2.4 Existing Personal Finance Apps Analysis

The rise of digital personal finance tools has made it easier for users to track spending, set budgets, and monitor financial goals. Popular apps like Mint, You Need A Budget (YNAB), PocketGuard, and Spendee are widely praised for their usability and automation. However, they share key limitations—most notably the absence of predictive budgeting, real-time AI guidance, and adaptive decision support—gaps that FinTrack aims to fill.

Mint, developed by Intuit, aggregates account data and auto-categorizes transactions, offering users visualizations like charts and summaries (Intuit Inc. 2023). Yet, it remains reactive, with no forecasting or personalized recommendations, making it difficult for users to plan ahead.

YNAB promotes intentional budgeting through its "every dollar has a job" philosophy You Need A Budget (2023). While it encourages discipline and provides goal-tracking features, it lacks automation and intelligent suggestions, requiring manual effort that may not suit all users.

PocketGuard helps prevent overspending by showing how much disposable income remains. However, its guidance relies on fixed rules rather than adaptive models. It does not offer forecasting or conversational support, limiting its interactivity PocketGuard Inc. (2023).

Spendee emphasizes visual appeal and supports shared wallets and manual or synced tracking. Despite these strengths, it offers no predictive analytics or AI-based budgeting, functioning primarily as a descriptive tool (Spendee Ltd. 2023).

Overall, these apps provide valuable tracking features but remain static in functionality. Without integrated forecasting engines or AI-driven assistants, they fall short in enabling proactive, personalized financial planning.

FinTrack addresses these limitations by introducing forecasting capabilities through an XGBoost model, which predicts future spending based on historical data. Although FinTrack does not yet provide fully personalized financial advice, it incorporates an AI-powered chatbot using Google's Gemini API to deliver general financial guidance. This chatbot can assist with common finance-related questions in real-time, adding a layer of interaction not found in the apps above. Additionally, FinTrack estimates time-to-goal progress based on savings inputs, offering users calculated, forward-looking insights.

In summary, current leading apps offer strong basic features but lack intelligence and foresight. FinTrack fills these gaps by introducing predictive modeling and AI-assisted interaction, even though its advice is not yet tailored to individual financial behavior. This foundation sets the stage for future enhancements where true personalization—based on user-specific data—can be achieved.

Table 2.2: Comparison of Key Features in Personal Finance Applications

App	**Predictive Analytics**	**AI Chatbot**	**Real-Time Insights**	**Automation Level**	**Personalized Suggestions**
Mint	No	No	Limited	Moderate	No
YNAB	No	No	Moderate	Low	No
PocketGuard	No	No	Moderate	Moderate	No
Spendee	No	No	Limited	Moderate	No
FinTrack	Yes (XGBoost)	Yes (Gemini API)	Yes	High	Partially (General Advice)

These studies collectively highlight the transformative potential of AI and machine learning in personal finance management. FinTrack builds upon these insights by integrating XGBoost for predictive analytics and incorporating AI-powered chatbots to provide real-time financial guidance, aiming to enhance user engagement and financial decision-making.

2.5 Related Academic Work and Its Influence on FinTrack

Academic research has played a crucial role in shaping modern personal finance management tools, particularly through the integration of artificial intelligence (AI) and predictive analytics. These studies identify both the potential and limitations of current methods—many of which informed FinTrack's design choices.

Agarwal et al. (2024) introduced MyFinanceAI, an advanced AI-powered financial assistant that addresses complex consumer finance challenges through a multi-layered architecture combining real-time analysis, personalized recommendations, and predictive modeling. A six-month pilot involving 1,000 users reported improvements in savings behavior, reduced financial stress, and overall financial well-being. Their work illustrates the value of integrating intelligent systems to improve user engagement and outcomes—an approach central to FinTrack.

In a comparative study, Adebayo et al. (2024) evaluated the forecasting performance of XGBoost, LSTM, and ARIMA in financial contexts. Their findings revealed that while LSTM showed slightly better long-term accuracy, XGBoost outperformed in speed, resource efficiency, and interpretability—supporting its selection for FinTrack, which targets short- to medium-term spending forecasts with limited user data.

Additionally, Hidayat et al. (2024) explored the broader impact of AI on financial decision-making, highlighting its utility in predictive analysis, risk assessment, and automation. Their research emphasizes the growing role of AI in shaping strategic financial tools, aligning with FinTrack's goal of enhancing personal finance through transparent, data-driven support.

2.6 Clearly Defined Literature Gap

The comprehensive review of existing scholarly and practical work reveals a persistent and significant gap in current personal finance applications: the lack of robust predictive analytics and real-time AI-driven user interaction. While mainstream apps like Mint, YNAB, PocketGuard, and Spendee have simplified expense tracking and offered basic financial visualization, they fall short in enabling users to proactively manage their finances through dynamic forecasting or adaptive advice systems. Most tools operate in a retrospective fashion, providing historical insights without leveraging data to predict future trends or deliver intelligent, contextual guidance.

Academic studies (e.g., (Hidayat et al. 2024); (Omarini 2017)) support this observation, highlighting that many existing fintech tools function primarily as passive financial trackers rather than intelligent, interactive assistants. Even research projects—such as the development of MyFinanceAI by (Agarwal et al. 2024)—emphasize the need for real-time analytics, AI-driven personalization, and contextual financial recommendations to improve user engagement and financial outcomes.

Furthermore, recent comparative analyses (e.g., (Li 2023); (Siami Namini et al. 2018)) demonstrate that while traditional forecasting models like LSTM offer strong long-term prediction capabilities, they often fall short in smaller-scale, real-time use cases. This supports FinTrack's approach to integrating lightweight, interpretable models like XGBoost alongside interactive AI guidance to bridge the engagement and functionality gaps in current personal finance applications.

FinTrack addresses these exact gaps by integrating XGBoost, a lightweight and highly accurate algorithm that effectively forecasts user spending patterns, even with small datasets. In addition, FinTrack deploys a Google Gemini-powered chatbot to provide AI-generated financial guidance.

By combining predictive analytics, goal-based projection tools, and interactive AI features, FinTrack emerges as a forward-thinking solution that aligns with the growing demand for intelligent, user-centric fintech applications. In addition to advancing current digital finance tools, it lays the foundation for future personalization and adaptability in personal finance technology.

CHAPTER 3

Methodology

3.1 Overview

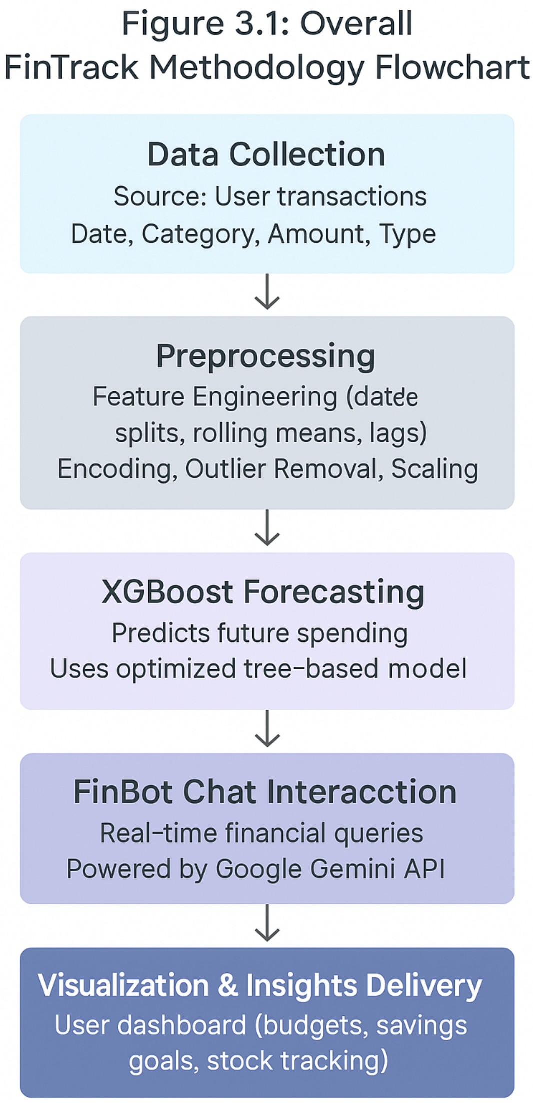
The primary objective of this project is to develop a personal finance management platform—FinTrack—that enhances conventional budgeting tools through the integration of predictive analytics and AI-powered conversational support. Aimed primarily at students and young professionals, the system offers insights into spending trends and provides intelligent assistance for making informed financial decisions.

The solution is centered around two main components. First, a predictive analysis engine based on the XGBoost algorithm forecasts future spending patterns using historical transaction data. This equips users with forward-looking insights to better manage their budgets. Second, a real-time financial chatbot, FinBot, is built using the Google Gemini API, enabling users to receive instant answers to general financial queries through natural language conversation.

The backend stack is built using Python, Flask, and Firebase for authentication, while financial records are stored locally using SQLite. Additionally, the Tingo API is integrated for live stock market analysis. On the frontend, developed in Flutter, financial visualizations such as pie charts, line graphs, and bar charts are rendered using Flutter widgets to ensure a seamless and responsive user interface.

Figure 3.1 provides an overview of the FinTrack system's architecture and methodology, illustrating how data flows between the predictive model, chatbot, database, and frontend components. This high-level flowchart helps contextualize the interaction between modules and supports the explanation of key design decisions in subsequent sections. Together, this combined architecture delivers an engaging and intelligent budgeting experience.

Figure 3.1: Overall FinTrack Methodology Flowchart



3.2 Data Collection and Preprocessing

3.2.1 Dataset Overview

The dataset used for developing FinTrack's predictive analytics component is the Personal Budget Prediction Dataset sourced from Kaggle (available at: Kaggle Link). This dataset is well-suited for financial forecasting as it reflects real-world individual budget behavior across daily transactions. The dataset, while suitable for proof-of-concept development, was sourced from Kaggle and may not fully represent broader or more diverse financial behaviors. This limitation is acknowledged and discussed in Chapter 6.

It contains typical financial features including:

- `amount` – the transaction value
- `category` – the transaction type (e.g., groceries, rent, utilities)
- `date` – the timestamp of the transaction

The structure provides an ideal foundation for both categorical pattern recognition and time-based forecasting, which are core to building a budget prediction engine.

3.2.2 Preprocessing Steps

Preprocessing was conducted using Python libraries such as `pandas`, `numpy`, `scikit-learn`, and `xgboost`, with the pipeline executed in Google Colab for cloud-based reproducibility. Initial steps involved parsing the `date` field into multiple features including:

- `year, month, day, day_of_week`, and a binary `is_weekend` feature

This temporal feature engineering enabled the model to capture periodic spending behaviors.

Next, categorical encoding was performed using `LabelEncoder` to numerically transform the `category` field into a machine-readable format, producing a `category_encoded` column. While more complex encodings like One-Hot Encoding were considered, Label Encoding was sufficient given XGBoost's ability to handle ordinal and categorical splits internally.

Outlier removal was applied using the Interquartile Range (IQR) method to eliminate anomalously high or low transaction values that could skew model training. The cleaned dataset was then augmented with rolling statistical features such as `rolling_mean_7`, `rolling_mean_30`, and lag variables including `lag_1`, `lag_7`, and `lag_30`. These time-window-based features were essential for capturing both recent and seasonal trends.

Further, `total_spending_category`, `month_over_month_change`, and `previous_day_spending` were engineered to give the model additional context on behavioral trends. The final step involved normalizing input features using `RobustScaler`, chosen for its resilience to outliers.

This preprocessing pipeline was encapsulated in a dedicated function and later deployed on the Flask backend as part of the `/predict_spending` endpoint, ensuring dynamic, real-time preprocessing of user transaction data before prediction. This seamless integration of preprocessing and model inference allows FinTrack to support high-frequency prediction requests at scale.

3.3 Predictive Analysis Methodology

3.3.1 Model Selection

To build the predictive budgeting engine within FinTrack, the XGBoost (Extreme Gradient Boosting) algorithm was selected due to its exceptional performance with structured financial data and its scalability in real-world production environments. XGBoost is a tree-based ensemble learning method introduced by (Chen & Guestrin 2016), optimized for speed and accuracy. Its ability to handle missing values, capture non-linear interactions, and deliver interpretable results made it particularly well-suited to this use case, where the data is tabular, relatively small-scale, and time-structured.

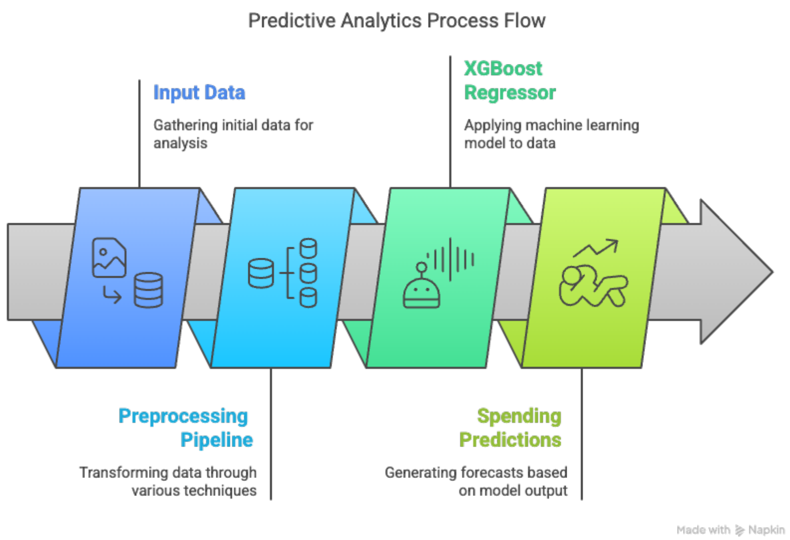
The dataset was preprocessed using feature engineering techniques detailed in Section 3.2. After preprocessing, the data was split into training and testing sets using an 80/20 ratio to ensure generalization. The model was trained on the `amount` variable as the target feature, using engineered features like `rolling_mean`, `day_of_week`, lag variables, and more.

To optimize performance, `Optuna` was used for hyperparameter tuning across 50 trials. The final model was trained with the best-found parameters and evaluated using the following metrics:

- **Mean Absolute Error (MAE)** – measures average prediction error
- **Root Mean Squared Error (RMSE)** – penalizes larger errors more heavily
- **R^2 Score** – evaluates goodness-of-fit
- **Mean Absolute Percentage Error (MAPE)** – measures percentage-based prediction accuracy

Figure 3.2 visually outlines the entire model pipeline, from raw user transaction inputs through preprocessing, feature extraction, model prediction using XGBoost, and output generation. This diagram provides a clear representation of the system's data flow and emphasizes the modular design approach adopted to facilitate real-time financial forecasting within FinTrack.

Figure 3.2: Model pipeline for predictive budgeting in FinTrack, outlining the data flow from raw inputs to final spending predictions using XGBoost.



3.3.2 Alternative Models Considered

An LSTM (Long Short-Term Memory) model was initially tested due to its strength in modeling sequential dependencies and financial time-series. LSTM, proposed by (Hochreiter & Schmidhuber 1997), is a variant of Recurrent Neural Networks (RNNs) specifically designed to handle long-term dependencies.

However, in the context of FinTrack, the LSTM model presented practical limitations:

- **Small dataset size:** LSTMs typically require large volumes of data to avoid overfitting.
- **Computational inefficiency:** Training was slower and required significant GPU resources.
- **Interpretability:** LSTM outputs are often opaque, making it difficult to explain predictions to end users.

Due to these constraints, LSTM was not adopted for deployment.

3.4 Chatbot Development Methodology

3.4.1 System Design and API Selection

To enhance user interaction and support financial literacy, FinTrack includes a conversational assistant—FinBot, implemented using Google's Gemini API. Among competing large language models (LLMs) such as OpenAI's ChatGPT and Anthropic's Claude, Gemini was selected for several strategic reasons. Notably, Gemini offered better contextual handling, cost-efficiency, and seamless integration via Google AI Studio, which simplified deployment for the Flask backend.

The chatbot is designed to handle queries related to:

- Personal budgeting
- Savings tips
- Financial planning
- Stock trends
- Fintech literacy

The integration is secured using environment variables via a `.env` file, ensuring that the API key is never hardcoded. Prompt engineering was applied to define a finance-specific system prompt that ensures consistent, domain-relevant responses. Gemini's endpoint is accessed using standard `POST` requests, and response handling logic gracefully captures and interprets the LLM's output.

3.4.2 Backend Architecture (Flask API)

The backend of FinTrack is built using Flask, following a modular and RESTful design to serve AI-powered predictions and chatbot responses efficiently. The application communicates with the Flutter frontend through well-defined `HTTP POST` endpoints, ensuring seamless interaction between the user interface and backend logic.

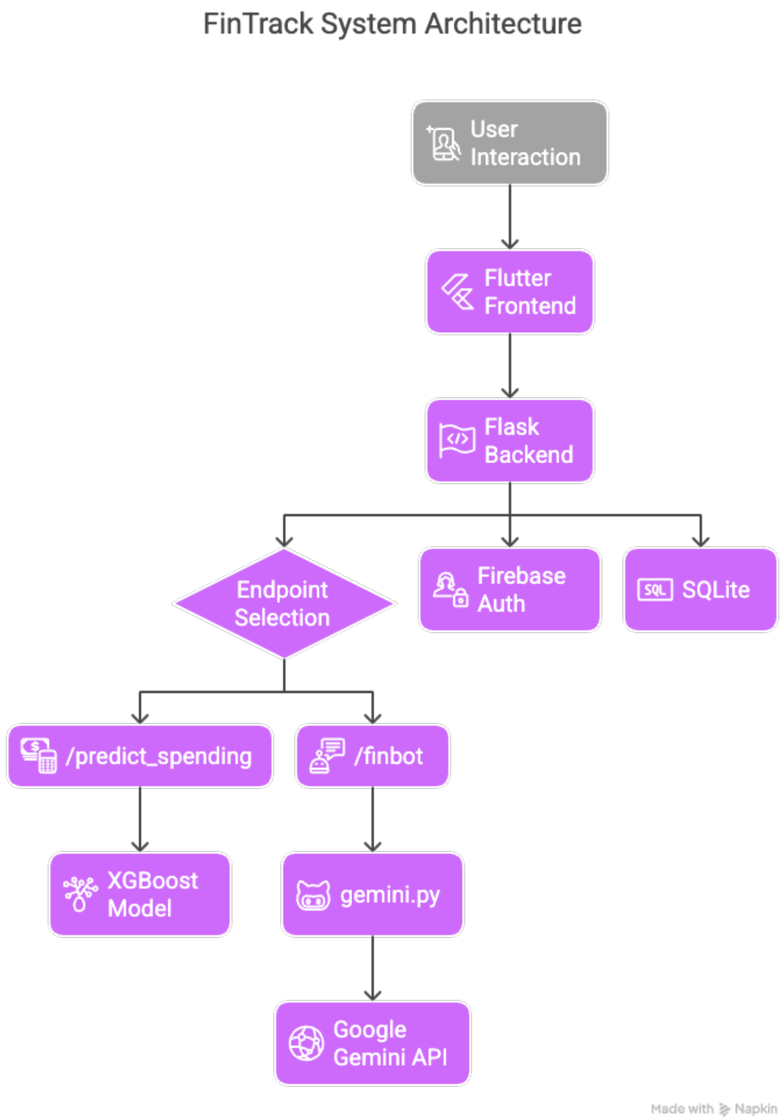
There are many key routes in the backend, but some examples are:

- `/predict_spending`: Accepts user transaction history, preprocesses it, and generates future spending forecasts using a pre-trained XGBoost model. This endpoint also incorporates goal comparison logic and returns warnings when predicted expenses exceed income.
- `/finbot`: Handles user-submitted financial queries. It delegates the query to the `process_finbot_query()` function in `gemini.py`, which communicates with the Google Gemini API and returns a domain-specific, AI-generated response.

Additionally, Firebase Authentication is integrated to support user login and sign-up, but it operates independently from the predictive and chatbot components.

Figure 3.3 presents an architectural diagram of the FinTrack system, visually mapping how the Flutter frontend interacts with the Flask backend via RESTful endpoints. It illustrates the data flow from user input to processing and output generation, highlighting the modular separation between the prediction engine, chatbot module, authentication system, and storage layers. This visual representation reinforces the system's clean architecture and supports its scalability and maintainability.

Figure 3.3: Architecture of FinTrack, illustrating how the Flutter frontend interfaces with Flask endpoints for prediction and chatbot responses.



3.4.3 SMS Parsing via Native Android Integration

To automate transaction entry and reduce manual input, FinTrack integrates a native SMS parsing feature specifically for Android devices. This feature is designed to extract financial transaction details directly from bank SMS alerts using platform channels, enabling communication between Flutter and native Kotlin code.

When a user taps the "Load SMS Transactions" button on the Flutter interface, a `MethodChannel` is triggered (`fintrack/sms`), invoking native Android code to access the SMS content provider. The Kotlin logic scans the inbox for messages containing transaction-related keywords (e.g., “AED”, “debited”, “credited”) and returns the filtered messages to Dart.

In the Flutter layer, regular expressions are used to:

- Extract the amount
- Detect transaction type (credit/income or debit/expense)
- Identify the merchant name

A keyword-category map is then used to auto-categorize merchants, such as mapping “KFC” to Food or “Uber” to Transportation. To maintain data integrity, the system also checks for duplicates by comparing the parsed SMS with existing transactions (by amount, type, and description).

This feature is particularly relevant for regions like UAE, where users commonly receive banking alerts via SMS rather than APIs. By automating the ingestion and categorization of such messages, the system significantly enhances real-world usability and user convenience.

3.5 Tools, Libraries, and Environment

FinTrack was developed using a modern tech stack that supports machine learning integration and mobile interaction. The backend was built in Python, utilizing the Flask web framework for handling API endpoints efficiently. SQLite was used as the primary local database for storing user transactions and budget data due to its simplicity and ease of use.

Firebase Authentication was incorporated to manage user login and registration securely. For the machine learning component, XGBoost was selected for its accuracy and performance with tabular data, while Pandas, NumPy, Scikit-learn, and Matplotlib were used for data preprocessing, transformation, and visualization. The AI chatbot functionality was powered by the Google Gemini API, integrated via a custom `gemini.py` module, and financial market data was retrieved using the Tingo API.

All development and testing were performed locally. The Flutter frontend was executed on a Google Android emulator, which allowed for real-time interaction testing without deployment.

CHAPTER 4

Implementation

4.1 System Architecture

FinTrack is designed as a modular, personal finance application that balances predictive analytics, financial interaction, and lightweight deployment. The system follows a client-server model, with a Flutter frontend acting as the user interface and a Flask backend providing data management, AI forecasting, and chatbot capabilities.

The backend communicates with three main functional pillars:

- a locally stored SQLite database,
- a trained XGBoost model for financial forecasting, and
- the Google Gemini API for real-time chatbot responses.

All modules are tied together using HTTP POST and GET endpoints handled by Flask.

As shown in Figure 4.1, the user communicates with FinTrack through a mobile interface (tested on a Google Android emulator), which sends requests to the Flask API. The API then performs model inference, database transactions, or AI chatbot generation depending on the request type.

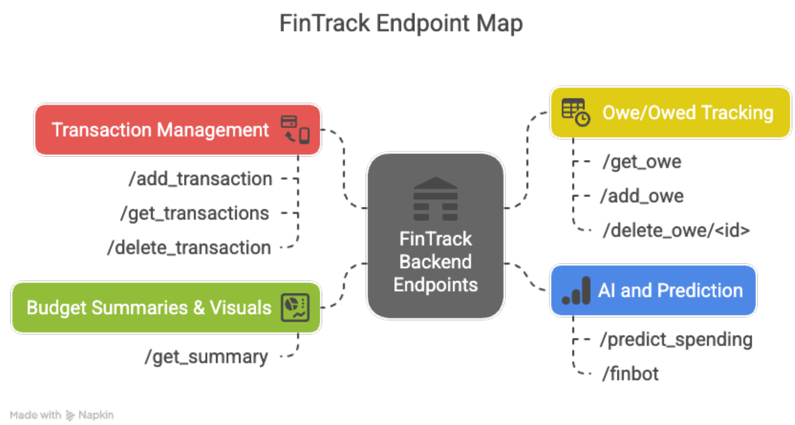
Backend Layer and API Routes

The Flask backend contains multiple endpoints organized by function:

- **Transactions:** `/add_transaction`, `/get_transactions`, `/delete_transaction`
- **AI Forecasting:** `/predict_spending` (uses XGBoost and preprocessing)
- **Chatbot:** `/finbot` (calls Gemini API via `gemini.py`)
- **Owe/Owed Tracking:** `/get_owe`, `/add_owe`, `/delete_owe/<id>`
- **Summary:** `/get_summary` returns categorized budget summaries

Each endpoint routes requests to internal helper modules such as `personal_ledger.py`, `transactions.py`, and `gemini.py`, which carry out the application's core logic. These interactions are mapped in Figure 4.2, which complements the broader architectural overview in Figure 4.1 by detailing the exact entry points and backend control flow.

Figure 4.1: FinTrack Flask Endpoint Map illustrating route-specific interactions and internal service handling logic.



4.2 Feature-Level Implementation

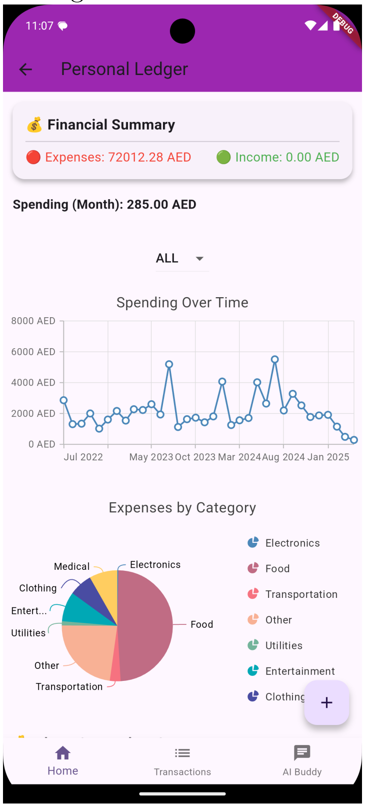
4.2.1 Personal Ledger and Visual Analytics

FinTrack's core functionality revolves around a robust personal ledger system powered by SQLite, a lightweight relational database suited for local data storage. The ledger stores detailed financial records including transaction ID, user ID, amount, category, type (income or expense), timestamp, currency, and description. This schema ensures every transaction is traceable, categorized, and aligned with user-specific financial activity.

Users interact with the ledger via the mobile frontend, where they can submit transactions through the `/add_transaction` endpoint. Each transaction is stored in the `transactions` table and categorized to allow visual summarization and budget tracking. Users can also delete entries using `/delete_transaction` and fetch historical data with `/get_transactions`.

To promote financial awareness and usability, the application integrates Flutter charting widgets for real-time visual analytics. A pie chart is used to represent spending distribution by category, while line and bar graphs illustrate weekly and monthly spending trends. Additionally, a dedicated Owe/Owed page visualizes outstanding personal debts using a separate `owe` table. This includes names, amounts, and dates, helping users manage interpersonal financial responsibilities effectively.

Figure 4.2: Example Flutter UI displaying category-based pie chart and weekly line graph for expense tracking.



4.2.2 AI Forecasting with XGBoost

FinTrack integrates predictive analytics using XGBoost (Extreme Gradient Boosting) to generate future spending forecasts based on a user's historical transaction data. The model was trained offline using engineered features and stored locally as a `.pkl` file via `Joblib`. During runtime, it is loaded into the Flask backend and used for real-time predictions through the `/predict_spending` endpoint.

Data Flow and Preprocessing

When a user selects a forecast timeframe (e.g., 1M, 3M, 6M, 1Y), the backend first calls `fetch_daily_sums()` to compute a user's daily expenditure history from SQLite. The resulting DataFrame is passed into `preprocess_for_xgboost()`, a custom preprocessing function that prepares the input with time-based features like:

- `year, month, day, day_of_week`
- Rolling averages (`rolling_mean_7`, `rolling_mean_30`)
- Lag variables (`lag_1`, `lag_7`, `lag_30`)
- Spending trend indicators (`previous_day_spending`, `month_over_month_change`)

The function ensures that the final input vector matches the order and structure expected by the XGBoost model, which improves performance and prediction accuracy.

Dynamic Forecasting Logic

The backend uses the most recent 30 days of cleaned data to maintain the continuity of rolling and lag-based features. Then, for each future day, a synthetic row is generated based on the last row's values. It simulates new day-level fields (date, weekday, weekend), recalculates features, and feeds the resulting vector into the XGBoost model to get a predicted amount. This loop continues until the requested number of days is forecasted.

A custom warning system compares the total predicted spending with the user's reported income. If the forecast exceeds income, a caution message is returned to inform users of potential overspending.

4.2.3 FinBot Chatbot Functionality

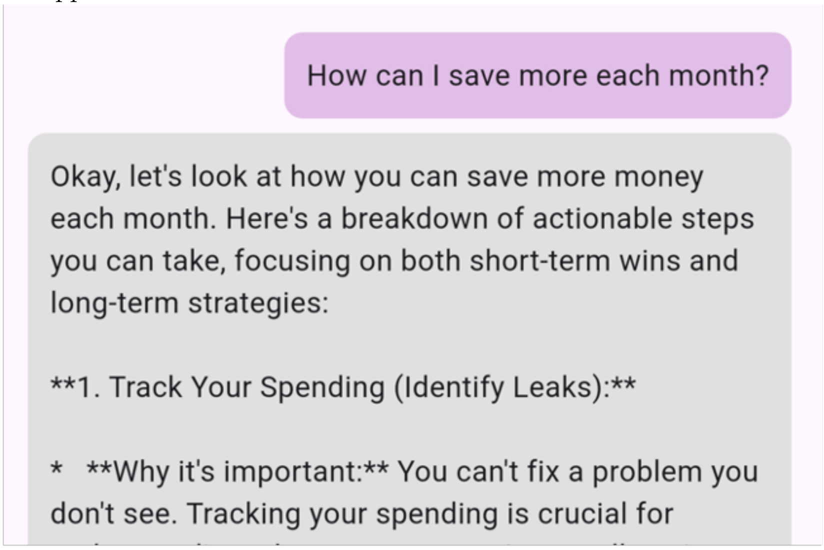
FinTrack integrates a lightweight yet powerful conversational agent, FinBot, designed to enhance user experience by providing finance-related guidance through natural language queries. The chatbot is built using the Google Gemini API, accessed via a dedicated module named `gemini.py`. This file defines a single function, `process_finbot_query()`, which constructs a finance-specific prompt, sends the query to the Gemini API, and parses the structured response.

The function wraps user input in a system prompt that ensures all answers relate to personal finance topics such as budgeting, investing, or saving. This prompt engineering approach helps generate contextually relevant, actionable responses.

The Flask endpoint `/finbot` receives JSON input from the frontend, containing the user's message. It then invokes the `process_finbot_query()` function and returns the result as a JSON response formatted for direct rendering in the Flutter UI. The chatbot interaction is stateless and does not maintain memory between sessions, which simplifies deployment and avoids persistent context handling.

FinBot allows users to ask personalized finance questions and receive immediate, AI-generated suggestions, complementing the app's predictive analytics with a conversational interface. A visual example of this interaction is shown in Figure 4.3, which displays an actual FinBot exchange within the mobile application.

Figure 4.3: Screenshot of FinBot chatbot conversation within the FinTrack mobile application.



4.2.4 Stock Market Tracking (Tingo API)

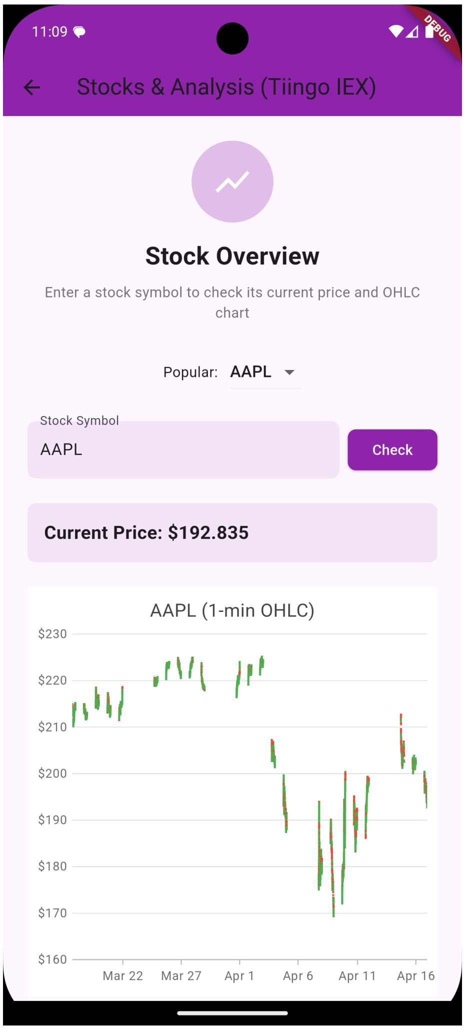
To enhance user decision-making and financial awareness, FinTrack integrates real-time stock tracking functionality using the Tingo API, a publicly available financial market data service. This feature enables users to search and view performance data for a wide range of stock symbols.

The API is queried from the Flutter frontend, which either dynamically constructs a request to the Tingo API or calls a backend proxy route to retrieve current price data and historical candlestick information. Users are provided with a dropdown menu containing top stock tickers (e.g., AAPL, MSFT, TSLA) to streamline the experience, but they can also search for custom tickers manually.

Upon receiving the response, the frontend parses key financial indicators and renders them into an interactive candlestick chart to visually represent stock movement trends. This visualization helps users interpret technical patterns and make informed investment decisions. An example of this output is shown in Figure 4.4, which displays a real-time candlestick chart generated within the FinTrack mobile application.

However, the integration is subject to limitations of the Tingo API's free tier, including request rate caps and restricted historical depth. Despite this, the functionality offers valuable supplementary insight for budget-conscious investors.

Figure 4.4: Candlestick chart generated using the Tingo API within the FinTrack mobile application.



4.2.5 SMS Parsing Feature

To enhance the user experience and minimize manual entry, a native SMS parsing feature was added to FinTrack. When users tap the "Load SMS Transactions" button, the app fetches bank SMS alerts from the device inbox using native Android code (Kotlin) via Flutter's platform channels.

The native code scans messages for patterns like "AED 100.00 debited at Carrefour", and sends relevant ones back to Flutter. These messages are then parsed using regular expressions to extract the amount, merchant, and transaction type (income or expense).

A mapping system was implemented to auto-categorize merchants (e.g., "KFC" → Food, "Uber" → Transportation). To avoid duplicates, each parsed transaction is checked against existing entries before being saved.

4.3 Technical Challenges and Solutions

During the development of FinTrack, several technical challenges emerged across both the backend and frontend, particularly during AI integration and third-party API usage. These obstacles were addressed iteratively using research, documentation, paid tools, and improved engineering practices.

One early challenge involved integrating Firebase Authentication. Initial SDK configuration and token handling created delays; however, this was resolved through official documentation and implementation tutorials.

The AI forecasting system presented more significant difficulties. The initial attempt to use LSTM models in Google Colab ran into GPU limitations due to the dataset size and compute constraints. This led to the purchase of Colab Pro ($10.99) for extended runtime and memory. However, even with improved resources, LSTM models proved unstable and overfit on limited data. As a result, the project transitioned to XGBoost, which handled tabular data more efficiently with significantly better training speed and accuracy.

Additionally, integrating the trained XGBoost model into the app surfaced prediction continuity issues related to time-series features like rolling averages and lag values. To address this, logic was implemented in the backend to recalculate these features daily during prediction loops.

The Gemini chatbot (FinBot) also presented formatting inconsistencies, particularly when queries were ambiguous or not strictly finance-related. A fallback mechanism was added, and stricter prompt templates were used to ensure the chatbot remained aligned with financial topics.

On the frontend, mobile responsiveness and chart visuals were initially subpar. Manually coded charts lacked clarity, especially for financial graphs like spending breakdowns and candlestick stock visuals. This was resolved by adopting Flutter charting widgets, which significantly improved aesthetic quality and data readability. Following principles of human-centered AI design improves trust and usability in financial applications, especially among users unfamiliar with technical tools (Shneiderman 2020).

Finally, while Finnhub API was the preferred choice for stock tracking, its free tier only supported limited fields like current, low, and high prices—insufficient for rendering candle charts. Tingo API was chosen instead, as it offered richer historical data even under its free tier, allowing for proper visualization and better integration into the stock tracking module.

CHAPTER 5

Integrated Results and Discussion

5.1 Predictive Analytics: Model Selection and Performance

5.1.1 Model Selection Rationale

The predictive analysis module within FinTrack was developed to generate forward-looking spending insights by learning from historical transaction behaviors. An XGBoost (Extreme Gradient Boosting) regression model was selected due to its exceptional performance on structured financial data, interpretability, and computational efficiency. XGBoost is particularly well-suited for tabular datasets and has demonstrated superior performance in various financial forecasting tasks (Chen & Guestrin 2016). While Long Short-Term Memory (LSTM) networks are renowned for capturing long-term dependencies in sequential data, they often require large datasets to perform effectively. Given the limited size of the available dataset, LSTM was deemed less suitable for this application.

5.1.2 XGBoost Performance

To build the predictive forecasting engine within FinTrack, the XGBoost algorithm was implemented due to its ability to efficiently model structured tabular data with high predictive accuracy. XGBoost, or eXtreme Gradient Boosting, combines the principles of gradient boosting and decision trees and is known for its performance, speed, and ability to handle missing values—making it particularly suited for real-world personal finance applications.

The model development was conducted in Google Colab using Python. The raw transaction dataset, imported from Google Drive, underwent comprehensive preprocessing. Date fields were parsed into temporal features such as year, month, day, and weekday. Categorical variables like spending category were label-encoded, and outliers were removed using the IQR method. To enable temporal pattern recognition, lag-based features (`lag_1`, `lag_7`, `lag_30`) and rolling means (`rolling_mean_7`, `rolling_mean_30`) were calculated. Additional trend-based features included `month_over_month_change`, `previous_day_spending`, and total category-based spending. These features were then scaled using RobustScaler to minimize the influence of extreme values.

Model tuning was conducted using Optuna, an automated hyperparameter optimization framework. Across 50 trials, Optuna identified the optimal configuration that minimized Mean Absolute Error (MAE). The final hyperparameters included: `n_estimators`=289, `max_depth`=8, `learning_rate`=0.0502, `min_child_weight`=7, `reg_lambda`=1.52, and `reg_alpha`=0.25.

Upon training, the XGBoost model was evaluated on a test set using standard regression metrics:

- MAE: 0.2687
- MSE: 0.3201
- RMSE: 0.5658
- R^2 Score: 0.987
- MAPE: 4.28%

These results indicate that the model successfully captured approximately 99% of the variance in user spending behavior. Figure 5.1 demonstrates this performance, where predicted values closely follow actual transaction patterns, validating the forecasting capability. Figure 5.2 shows the importance of different features, highlighting that temporal and behavior-based variables such as `month_over_month_change`, `lag_1`, and `rolling_mean_7` had the highest predictive influence.

Further diagnostic analysis was conducted to evaluate the model's residuals. As shown in Figure 5.3, the residual plot displays a random distribution around zero, indicating a lack of systematic bias and confirming that the model generalizes well without overfitting. Moreover, Figure 5.4 illustrates the residual distribution's histogram and KDE curve, showing a symmetric, near-normal distribution centered around zero. This pattern reinforces that prediction errors are low and evenly distributed, which is ideal for trust and reliability in financial forecasting applications.

Collectively, these results confirm that the XGBoost pipeline, powered by engineered time-series features and advanced tuning techniques, delivers accurate, interpretable, and production-ready spending predictions suitable for real-world use cases in FinTrack.

Figure 5.1: Actual vs. Predicted Spending Line Plot

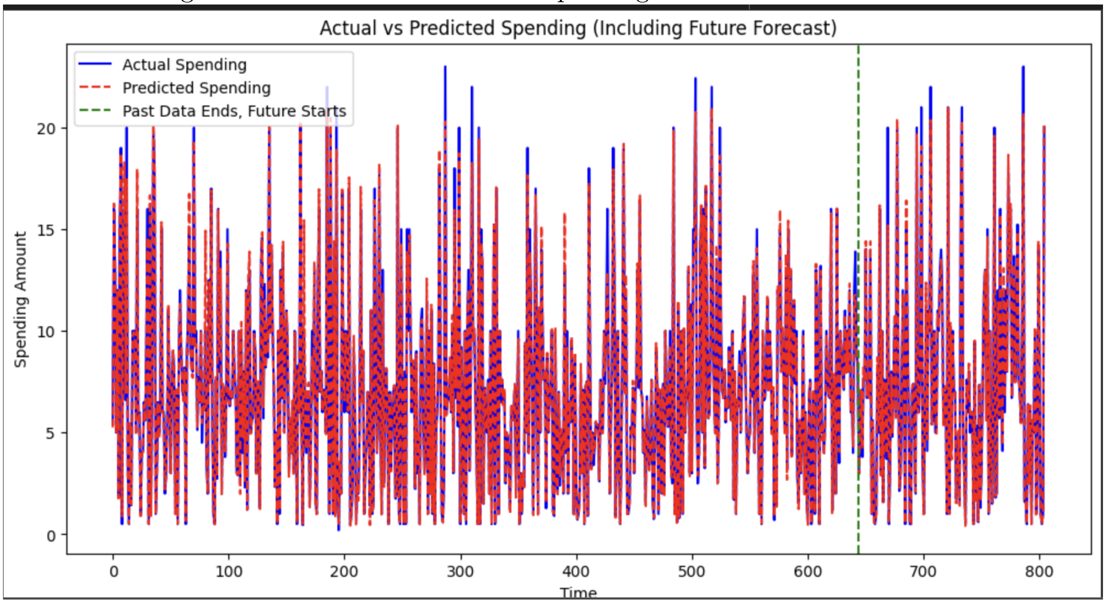


Figure 5.2: XGBoost Feature Importance showing the top predictors of financial behavior

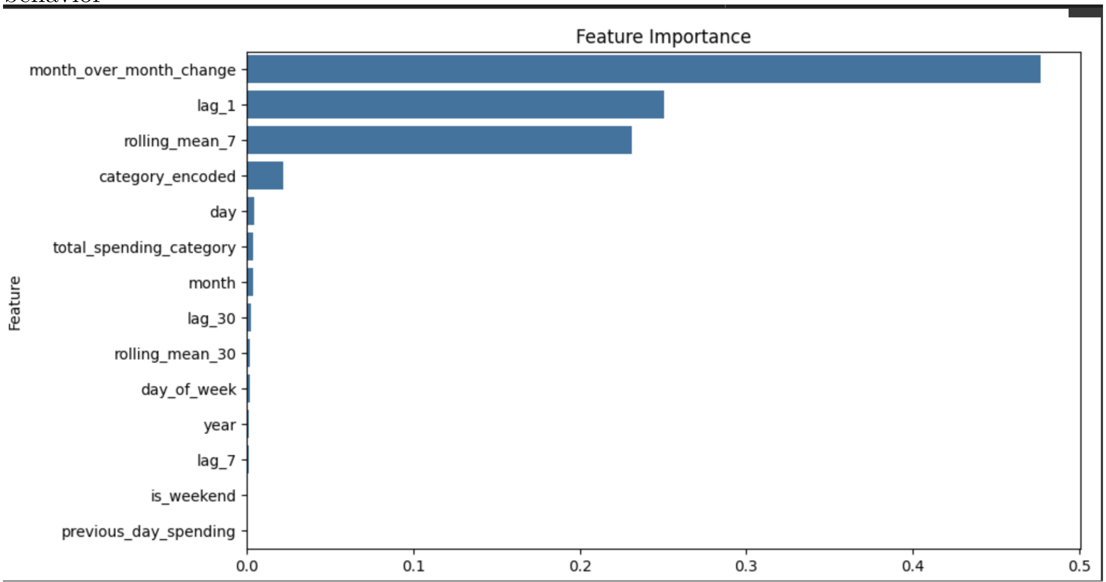


Figure 5.3: Residual Plot (Predicted Spending vs. Error)

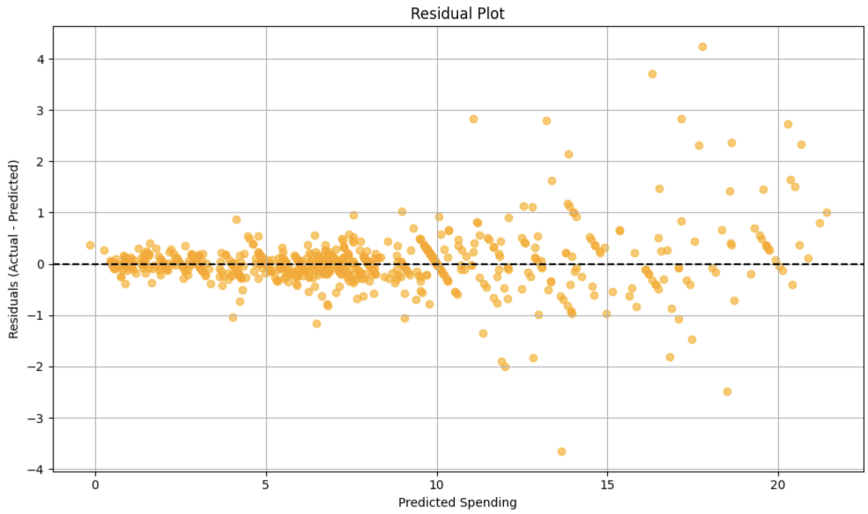
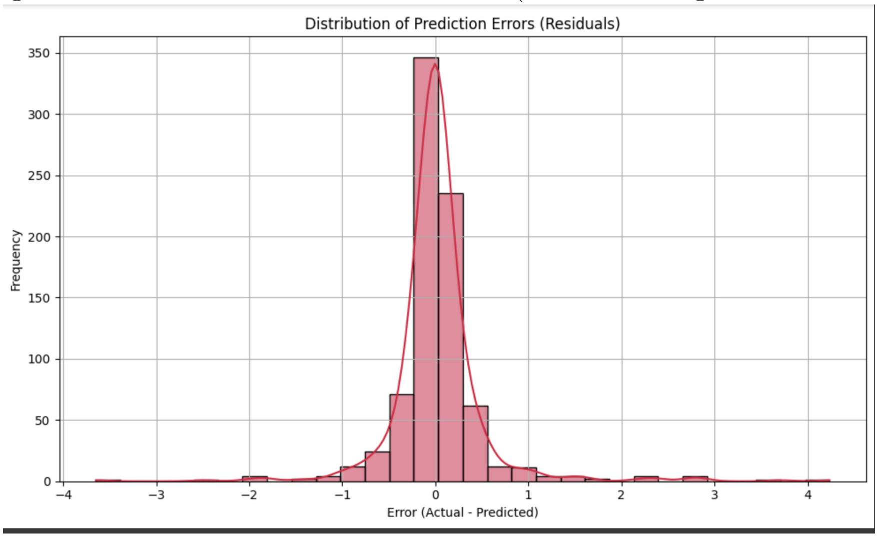


Figure 5.4: Distribution of Prediction Errors (Residual Histogram with KDE)



5.1.3 LSTM Comparison (Rejected Model)

As part of the exploratory phase, a Long Short-Term Memory (LSTM) neural network was implemented to evaluate its suitability for personal finance time-series forecasting. While LSTM is often praised for capturing long-term dependencies in sequential data (Hochreiter & Schmidhuber 1997), the model's application in this project yielded suboptimal outcomes.

After training the LSTM on the preprocessed dataset, the results were as follows:

- MAE: 0.2298

- MSE: 0.00397
- RMSE: 0.0629
- R^2 Score: -6.32×10^{12} (extremely poor)

Although MAE and RMSE appear reasonable, the catastrophically negative R^2 score highlights a severe model failure. Possible reasons include:

- The model memorized training patterns without generalizing to test data
- Data leakage or poor validation strategy
- Divergence in loss curves due to poor convergence
- Absence of normalization layers or gradient explosion

Due to these constraints, LSTM was not adopted for deployment. Its instability, long training time, and lack of interpretability confirmed XGBoost as the superior choice for FinTrack's predictive engine.

5.1.4 FinBot: AI Chatbot Evaluation

Introduction

FinBot is an AI-driven chatbot integrated into the FinTrack application, powered by Google's Gemini API. Designed to deliver real-time responses to user queries, FinBot provides assistance with budgeting tips, savings strategies, and general financial guidance.

Recent research suggests that embedding intelligent virtual assistants within financial platforms can significantly enhance user trust and engagement—especially when these assistants offer personalized recommendations and maintain secure communication protocols (Ng et al. 2020).

This section qualitatively evaluates FinBot's effectiveness by analyzing the clarity, relevance, and actionability of its responses across sample interactions that reflect common financial questions users may encounter.

This section qualitatively evaluates FinBot's effectiveness by analyzing the clarity, relevance, and actionability of its responses through sample interactions that reflect common financial questions users might ask.

Prompt Examples and Responses

Example 1 — User: *"How can I save 1000 AED in 3 months?"*
FinBot (Summary): *Start by breaking your savings goal into monthly targets (e.g., 333 AED/month). Focus on essentials and avoid impulse purchases. Use cash envelopes for categories like food or transport. Take advantage of Ramadan and Eid offers. Track progress weekly.* FinBot's interface response is shown in Figure 5.5.
Summary: FinBot offers practical and localized saving advice. It breaks the goal into manageable parts (333 AED/month), promotes disciplined spending using cash envelopes, and encourages leveraging regional events like Ramadan for discounts. Weekly tracking reinforces accountability.

Example 2 — User: *"Is it a good idea to invest in stocks on a low income?"*
FinBot (Summary): *Yes, but cautiously. Begin with small amounts. Focus on index funds or low-risk options. Investing builds discipline, teaches financial literacy, and can beat inflation. Prioritize emergency savings and avoid debt.*Figure Reference: FinBot's response to this question is presented in Figure 5.6.
Summary: FinBot balances encouragement with caution. It suggests starting small and focusing on safer investments. The response emphasizes financial discipline, long-term benefits, and prioritizing emergency savings before investing.

Example 3 — User: *"What percentage of my salary should go to rent?"*
FinBot (Summary): *Ideally, your rent should not exceed 30–35% of your monthly income. This allows room for other expenses like savings, food, transport, and emergencies. Adjust based on your city's cost of living.* This interaction is visualized in Figure 5.7.
Summary: FinBot provides a widely accepted budgeting guideline by suggesting rent remain within 30–35% of income. It also acknowledges geographic variability and the importance of leaving space for savings and essential spending.
Evaluation: Very clear and educational. The chatbot provides a rule-of-thumb with room for personalization, showing awareness of varying financial contexts. Practical and user-friendly.

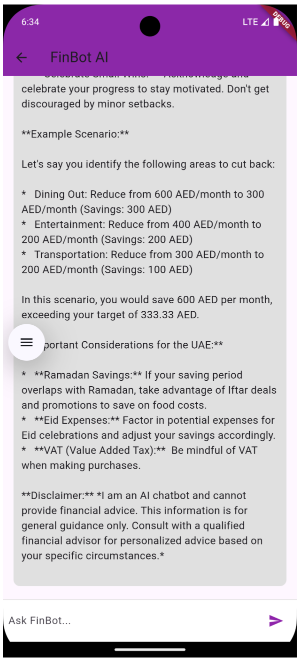


Figure 5.5: FinBot's Answer to Example 1

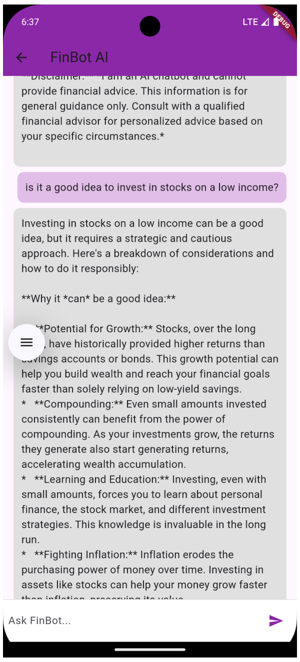


Figure 5.6: FinBot's Answer to Example 2

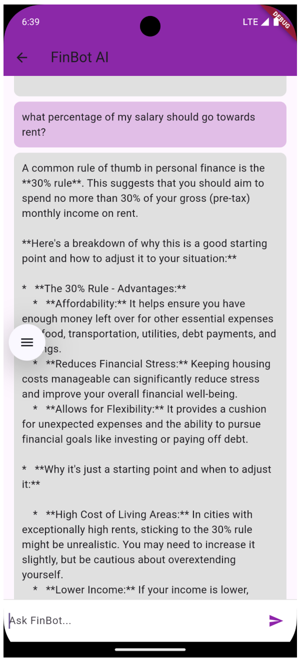


Figure 5.7: FinBot's Answer to Example 3

5.1.5 SMS Parsing Feature Evaluation

Introduction

To streamline financial tracking and minimize manual input, FinTrack integrates a native Android SMS parsing feature that automatically extracts and logs transaction data from SMS banking alerts. This system is built using Flutter's platform channel architecture, allowing communication between Dart code and Android-native methods implemented in Kotlin. The feature uses a custom `SmsService` class that invokes the `getSms()` method via a platform channel to fetch SMS messages from the device inbox.

Once received, the raw SMS data is processed in the Dart frontend within the `TransactionsPage` widget. A combination of regular expressions and keyword-based matching is used to extract the transaction amount, type (credited or debited), merchant name, and date. The extracted data is then auto-categorized using a predefined keyword mapping dictionary for merchants such as KFC, Carrefour, ADNOC, Talabat, and more. Each new transaction is checked against existing records to prevent duplication before being added to the user's ledger.

Emulator Testing and Evaluation

To evaluate the accuracy and reliability of the SMS parser, synthetic banking messages were sent to the Android emulator via the Device Manager in Android Studio. Example messages included realistic patterns such as:

```
"AED 250.00 was debited from your account at Carrefour."
```

Upon receiving such a message, the system correctly performed the following operations:

- Extracted the amount as `250.00` using the pattern `AED (\d+(\.\d{1,2})?)`
- Detected the transaction type as `Expense` by searching for the keyword `"debited"`
- Parsed the merchant name as `Carrefour` using the phrase structure "at <merchant>"
- Matched the merchant to the predefined category `Groceries`
- Validated against the transaction list to ensure no duplicate entries existed before saving

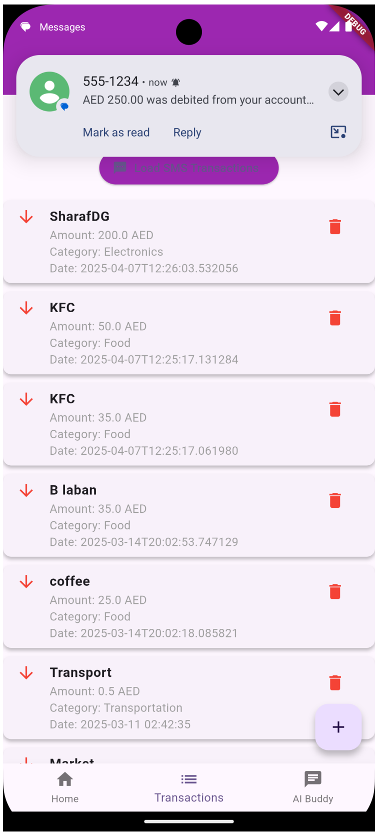


Figure 5.8: Message being sent to the emulator device

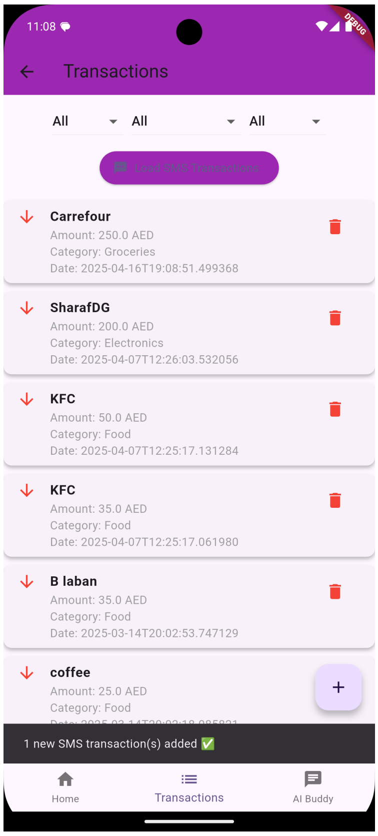


Figure 5.9: SMS successfully parsed and added to records

This process was executed in the `loadSmsFromInbox()` method, which first requests SMS read permission using the `permission_handler` package. A synthetic SMS message is sent to the emulator and parsed using regular expressions to extract the amount, merchant name, and transaction type, followed by automatic categorization and storage via `TransactionsService.addTransaction()`, as illustrated in Figures 5.8 and 5.9.

Robustness and Use Case Coverage

The parser was tested with multiple message templates representing various financial scenarios:

- Salary credit notifications — correctly classified as `Income`
- Food delivery (e.g., Talabat, Zomato) — classified as `Food`
- Fuel station alerts (e.g., ADNOC) — categorized as `Fuel`
- Unknown merchants — defaulted to the `Other` category
- Repeated SMS messages — successfully filtered out using a duplicate-checking mechanism based on amount, type, and description

The system maintained high parsing accuracy across tests and demonstrated reliable handling of edge cases, such as missing merchant names or ambiguous phrasing. In such cases, it gracefully fell back to "Unknown Merchant" and applied a default "Other" category.

Impact and Future Expansion

This feature significantly enhances the FinTrack user experience by enabling localized, semi-automated transaction logging, particularly in regions like the UAE where SMS banking remains a primary notification method. It eliminates the friction of manual entry and encourages regular financial tracking.

Future improvements may include:

- Integration of natural language processing (NLP) to better understand diverse message templates
- Use of persistent memory or caching to retain already parsed messages across sessions
- Multi-language support for Arabic or other regional SMS formats
- Expansion of the merchant-category mapping dictionary via crowdsourced feedback

Overall, the SMS parser represents a cornerstone of FinTrack's value proposition, combining user convenience, automation, and intelligent categorization to simplify personal finance management.

5.2 User Evaluation and Summary

5.2.1 Emulator Testing

To assess usability and core functionality, Zidan (a final-year Business student) tested the app using a Pixel 9 Pro connected to the developer's laptop via an emulator session. He successfully:

- Added a transaction
- Interacted with the FinBot chatbot
- Viewed financial breakdowns via interactive charts
- Checked NVIDIA)

Feedback Summary:
"The app is clean and easy to navigate. The chatbot gave useful and clear answers, and the category-based visuals were really helpful to understand spending patterns quickly. Only issue was the stock market analysis page—it was slow to load."

Based on feedback from the emulator testing, visual clarity was praised, but performance issues in the stock analysis screen led to streamlining data calls. Additionally, the user recommended making FinBot more proactive — a feature now planned for future iterations.

5.2.2 Survey Findings

To understand user preferences and attitudes toward AI in finance, a short Google Form survey was distributed and received 10 anonymous responses. The goal was to evaluate the demand for intelligent personal finance tools and identify user expectations.

Figure 5.10 shows that while a minority of users currently use budgeting apps, interest in such tools is growing. 60% of respondents do not use any personal finance app, highlighting a gap in adoption and an opportunity for simpler, AI-integrated solutions like FinTrack.

Notably, 90% of respondents expressed interest in an app that combines AI-powered budgeting with an interactive chatbot. This confirms strong demand for predictive insights and conversational interfaces in financial planning.

The most desired features were:

- Spending prediction
- Smart budget suggestions
- Category-based analysis
- Chatbot-based financial advice

Respondents also shared insights around trust and transparency. While many were open to AI-based forecasts, some suggested the need for clear logic or disclaimers explaining how predictions are made. This aligns with FinTrack's approach of integrating ethical AI practices and visible prompts in FinBot.

Overall, the survey validates FinTrack's core features while also identifying areas for enhancement—particularly in improving user trust through transparent design and educational explanations.

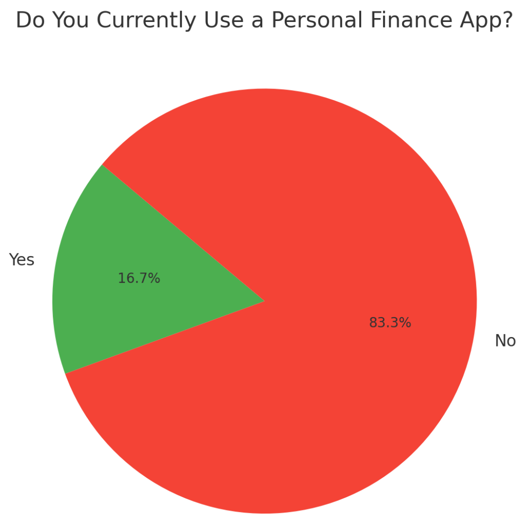


Figure 5.10: Do You Currently Use a Personal Finance App?

5.2.3 Interpretation and Comparison with Literature

Interpretation of Results

The results from the FinTrack system demonstrate the practical viability and effectiveness of integrating AI for both predictive budgeting and real-time financial assistance. The predictive model, powered by XGBoost, yielded highly promising outcomes, achieving a Mean Absolute Percentage Error (MAPE) of approximately 4.28% and an R^2 score of 0.98. These metrics indicate a very strong correlation between actual and predicted values, showcasing the model's precision in forecasting short-term spending patterns. The use of rolling features and lag-based predictors ensured temporal consistency in the time series, making the model more reliable for near-future forecasts. While long-term predictions beyond 90 days may accumulate slight inaccuracies, the 30-day and 3-month forecasts remained highly consistent, supporting their utility for routine budgeting purposes. Integration of the trained XGBoost model into the Flask backend and its subsequent connection to the frontend allowed for real-time, low-latency predictions.

The chatbot component, FinBot, provided relevant and easy-to-understand financial guidance. Most responses, especially those involving savings strategies and budgeting, were both actionable and context-aware. For example, when asked about saving 1000 AED in three months, the chatbot offered step-by-step budgeting advice, categorized expense breakdowns, and even UAE-specific tips like Ramadan deals and VAT awareness. However, limitations emerged in responses that required deeper contextual memory or nuanced investment advice—where the output was sometimes too generic or lacked disclaimers. While FinBot was effective in answering domain-specific queries, its lack of contextual memory limits its ability to carry forward previous conversations. Additionally, it does not yet support real-time data ingestion or live financial updates. These issues reflect the inherent limits of using a single-prompt Gemini API instance without session continuity.

A noteworthy enhancement during the project was the addition of native Android SMS parsing, enabling FinTrack to automatically ingest and categorize financial transactions from user inboxes. This reduces manual entry and improves real-world usability—especially in regions like the UAE where SMS alerts are more prevalent than open banking APIs. The integration of platform channels and native Android content resolvers also exemplifies cross-platform mobile engineering best practices.

User evaluation, while limited to early testing, further validated the system's design choices. Zidan, a final-year business student, tested the app on a Pixel 9 Pro using emulator pairing and successfully navigated all core features—adding transactions, generating predictions, viewing charts, and querying FinBot. He praised the intuitive UI and visual spending breakdowns but noted that the stock analysis page was slightly slow, likely due to limitations in the free tier of the Tingo API. In parallel, a survey conducted with ten users showed that 90% were interested in a finance app combining AI forecasting and chatbot assistance. While the sample size is small, it supports the growing demand for accessible, intelligent personal finance tools, especially among digitally literate users.

Comparison with Literature

The integration of XGBoost into FinTrack's architecture resulted in a high-performing predictive engine, achieving an $\mathbf{R^2}$ score of 0.987 and a Mean Absolute Percentage Error (MAPE) of 4.28%. These metrics indicate strong predictive reliability and low deviation from actual spending patterns, affirming the algorithm's suitability for small-scale, structured financial data. This outcome is consistent with observations by Nielsen (2016), who examined XGBoost's success in direct-response modeling and other structured-data problems, attributing its performance to high predictive accuracy and accessible interpretability through feature importance scores. XGBoost has been recognized for its efficiency in handling structured data, making it suitable for applications requiring real-time decision support, such as personal finance management tools like FinTrack.

In contrast, the Long Short-Term Memory (LSTM) model—though theoretically appropriate for sequential forecasting—exhibited significant instability during implementation. Its inability to generalize, evidenced by a catastrophically negative R^2 value and erratic outputs, aligns with earlier findings by Siami Namini et al. (2018), who emphasized LSTM's tendency to underperform on small-scale datasets. Given that personal finance data typically spans limited durations (e.g., 3–6 months of history), LSTM's need for long sequences and larger training volumes rendered it impractical for deployment within FinTrack's lightweight architecture.

The FinBot component, built using Google's Gemini API, demonstrated practical efficacy in delivering domain-relevant financial advice. Despite its stateless nature and lack of conversational memory, FinBot consistently produced clear, context-aware responses. This performance aligns with findings by Zarifis & Cheng (2024), who examined how trust in generative AI varies depending on the specificity of financial questions, and emphasized the importance of transparency and human oversight in fostering user trust. Although FinBot currently lacks session persistence, its structured prompt engineering and user-friendly tone validate the potential of stateless large language models (LLMs) as accessible entry points into AI-supported financial literacy tools.

Furthermore, the role of FinBot in promoting financial awareness aligns with broader academic insights on financial literacy. (Lusardi & Mitchell 2014) have highlighted the persistent gaps in financial knowledge, particularly among young individuals and underserved communities. In this context, tools like FinBot can act as accessible intermediaries, providing simplified and on-demand financial guidance that bridges knowledge gaps without overwhelming users. By integrating intelligent, conversational responses into routine budgeting tasks, this project advances the vision of making financial education more inclusive and actionable. It demonstrates how conversational AI can function as a supportive layer in personal finance management, especially for users who may lack access to traditional financial advisory services.

Together, the results from the FinTrack application not only confirm the technical feasibility of combining XGBoost with LLM-powered financial assistants but also underscore their combined potential to democratize access to financial planning tools. Future enhancements—such as context-aware FinBot sessions and user-specific financial guidance—could further elevate the system's impact, transforming it into a truly adaptive financial companion.

5.3 Implications of the Findings

5.3.1 For Users

FinTrack showcases how predictive analytics and conversational AI can transform financial self-management. By offering personalized forecasts through XGBoost and real-time guidance via FinBot, users are empowered to make informed decisions with greater confidence. Even minimal early testing helped highlight usability concerns and shape planned improvements like proactive tips and improved chart loading. The ability to visualize category-based spending, simulate future expenses, and ask finance-related questions provides a level of interactivity and education not typically available in free financial apps. These features cater especially well to students, young professionals, and budget-conscious users—groups often underserved by existing platforms. This dual integration of predictive tools and AI-driven conversation bridges the gap between passive tracking and active financial planning, thus promoting better habits and improving awareness of personal financial health.

5.3.2 For Developers and Fintech Startups

From a development perspective, the project highlights key lessons in designing smart, modular financial systems. For example, handling rolling averages and lag features dynamically in the prediction pipeline was crucial for maintaining accuracy—especially when generating forward-looking forecasts in real time. This emphasizes the importance of time-series preprocessing in budget-based machine learning models. Additionally, the seamless integration of lightweight technologies—Flask, SQLite, Gemini API, and XGBoost—demonstrates that building intelligent financial tools does not require enterprise-level infrastructure or massive datasets. With a structured backend and prompt engineering, AI chatbots and ML-driven forecasts can be deployed in scalable, resource-efficient ways. Future versions of FinBot could leverage models like Gemini Pro or Gemini 1.5 with extended context windows to support memory and personalized session continuity.

5.5.3 For the Fintech Industry

The overwhelming interest in AI-powered financial planning, as seen in the 90% positive survey response rate, reflects a growing demand among younger demographics for intuitive, hybrid solutions. These users seek not just budgeting tools, but intelligent systems that offer proactive advice and relatable insights.

FinTrack's architecture suggests a future where AI-driven assistants become an integral part of personal finance, particularly in regions or user groups where traditional financial education is lacking. The industry stands to benefit by adopting more accessible, transparent, and low-barrier tools that combine analytics, personalization, and conversational interfaces—all while maintaining user trust and simplicity.

A 2024 report by (McKinsey & Company 2024) highlights the increasing integration of embedded AI solutions in personal finance platforms, emphasizing the transformative potential of AI-driven personalization and automation in the financial services industry. The report notes that organizations are beginning to create structures and processes that lead to meaningful value from generative AI, with large companies leading the way. However, experts warn that the large-scale deployment of AI in financial systems may introduce systemic risks, including volatility, feedback loops, and algorithmic bias in lending models. (Danielsson & Uthemann 2023) argue that while AI can enhance efficiency in micro-regulations, its application in macro-financial stability is limited due to the infrequent and unique nature of financial crises, which frustrates machine learning models.

CHAPTER 6

Conclusion and Future Work

6.1 Summary of Achievements

This project successfully developed an intelligent personal finance management application—FinTrack—that integrates predictive budgeting through XGBoost and real-time conversational support via a Gemini-powered AI chatbot (FinBot). The core aim was to enhance users' financial literacy and planning capabilities by enabling smarter decision-making based on data-driven insights and AI interaction.

From a technical standpoint, the app features a robust XGBoost model capable of forecasting short- and long-term spending trends with high accuracy. The model uses a rolling feature-enhanced pipeline and was integrated into the backend for real-time responsiveness. FinBot, the chatbot component, is powered via the Google Gemini API and allows users to ask financial questions and receive instant, context-specific advice.

The backend architecture—developed using Flask, SQLite, and Firebase authentication—is modular and lightweight, ensuring adaptability and efficiency. Visual feedback mechanisms were implemented through Flutter charting widgets, including category-wise pie charts, bar graphs for historical trends, and a clear owe/owed financial overview.

The app's interface, tested on a Pixel 9 Pro emulator, proved user-friendly and intuitive. Critically, FinTrack fills a major gap in the current fintech ecosystem by offering a rare blend of AI-driven forecasting and interactive financial guidance, all built without enterprise-scale infrastructure—demonstrating what's possible with focused, resource-conscious development.

6.2 Research Limitations

Despite the successful implementation of FinTrack, several limitations were encountered throughout the development and evaluation phases.

Data constraints posed a key limitation. The predictive model was trained on a single dataset sourced from Kaggle, which may affect its ability to generalize across diverse financial behaviors. Additionally, while an LSTM-based model was briefly explored, it was ultimately abandoned due to computational inefficiencies, reducing the depth of comparative analysis.

User testing was limited by the lack of public deployment. Evaluation occurred through emulator testing with a single user, and while a survey was conducted with 10 participants, the small sample size limits broader generalizability.

FinBot, while useful, lacks contextual memory and live data integration for stock candlestick charts, preventing it from proactively adjusting its responses based on past interactions or real-time financial changes. Additionally, the chatbot operates on static prompt inputs without hallucination prevention or live API augmentation, which can affect depth and accuracy in certain financial topics. Despite their promise, AI-powered financial advisors remain limited by their inability to offer regulated, fiduciary-level guidance, often lacking deep personalization and legal clarity. The "black box" nature of many AI algorithms raises significant transparency and accountability concerns, especially in finance, where explainability is crucial for compliance and trust (Reuters 2024).

Finally, due to infrastructure constraints, cloud hosting, multi-device testing, and comprehensive performance benchmarking were not conducted during the project.

6.3 Future Work

Several promising directions can enhance the capabilities and impact of FinTrack in future iterations. To begin with, FinBot can be improved by integrating contextual memory, enabling the chatbot to learn from prior conversations and provide follow-up insights.

On the forecasting side, expanding the prediction engine to incorporate real-time income and transaction data would make budgeting advice more dynamic. If larger datasets become available, exploring deep learning models like LSTM or Transformer-based architectures could uncover richer patterns in user behavior.

From a usability standpoint, deploying the app publicly and conducting end-to-end user testing will help refine the interface and gather valuable feedback. Additional features such as bank account integration, goal-based financial planning, and enhanced stock market analysis could make FinTrack a comprehensive financial assistant for young professionals and budget-conscious users alike.

Incorporating ethical safeguards in AI-driven financial systems, such as transparency and user consent, is crucial. Floridi et al. (2018) emphasize that responsible AI design in finance should prioritize fairness, explainability, and user autonomy, especially when personal financial data is involved. Future work should also explore fairness-aware learning models to reduce bias in financial predictions, particularly in credit scoring systems, where machine learning can inadvertently encode discriminatory patterns (Giang et al. 2024).

Bibliography

Adebayo, J., Ogunleye, S. & Afolayan, M. (2024), 'Comparative analysis of xgboost, lstm and arima in financial time series forecasting', *Mountain Top University Journal of Applied Science and Technology (MUJAST)* **4**(2), 15–25. Available at: https://mujast.mtu.edu.ng/storage/issues/Year_2024_Vol_4/Number_2/1729800557_MUJAST_240801.pdf (Accessed: 12 April 2025).

Agarwal, V., Ray, R. & Varghese, N. (2024), An ai-powered personal finance assistant: Enhancing financial literacy and management, *in* 'FOSS Approaches towards Computational Intelligence and Language Technology (FOSS-CIL T24)'. Available at: https://www.researchgate.net/publication/381563265_An_AI-Powered_Personal_Finance_Assistant_Enhancing_Financial_Literacy_and_Management.

Bayakhmetova, A., Rudenko, L., Krylova, L., Suleimenova, B., Niyazbekova, S. & Nurpeisova, A. (2025), 'Artificial intelligence in financial behavior: Bibliometric ideas and new opportunities', *Journal of Risk and Financial Management* **18**(3), 159.

Bhattacharjee, D., Srivastava, D., Mishra, P., Adhav, D. & Singh, M. (2024), 'The rise of fintech: Disrupting traditional financial services', *Educational Administration: Theory and Practice* **30**, 89–97.

Buckley, R., Arner, D. & Barberis, J. (2016), 'The evolution of fintech: A new post-crisis paradigm?', *Georgetown Journal of International Law* **47**, 1271–1319.

Cai, C. W., Saha, A. & Zhang, Y. (2021), 'Artificial intelligence and financial decision-making: Enhancing consumer outcomes through smart systems', *Journal of Financial Services Marketing* **26**(2), 87–99.

Chen, T. & Guestrin, C. (2016), Xgboost: A scalable tree boosting system, *in* 'Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, San Francisco, CA, USA, pp. 785–794.

Cloudester (2025), ‘Revolutionizing user engagement: The role of ai in app personalization’. Available at: https://cloudester.com/revolutionizing-user-engagement-ai-app-personalization/ (Accessed: 13 April 2025).

Danielsson, J. & Uthemann, A. (2023), ‘On the use of artificial intelligence in financial regulations and the impact on financial stability’, *SSRN Electronic Journal* . Available at: https://ssrn.com/abstract=4604628.

DealPotential (2024), ‘Global fintech market analysis report 2024’, https://dealpotential.com/wp-content/uploads/2024/03/global-fintech-market-analysis-report-2024.pdf. Accessed: 2025-04-12.

Demirgüç-Kunt, A., Klapper, L., Singer, D., Ansar, S. & Hess, J. (2018), ‘The global findex database 2017: Measuring financial inclusion and the fintech revolution’, *World Bank Group* . https://globalfindex.worldbank.org/.

Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., Luetge, C., Madelin, R., Pagallo, U., Rossi, F. et al. (2018), ‘Ai4people—an ethical framework for a good ai society: Opportunities, risks, principles, and recommendations’, *Minds and Machines* **28**(4), 689–707.

for Economic Co-operation, O. & Development (2023), Oecd/infe 2023 international survey of adult financial literacy, Technical Report 39, OECD Publishing. Available at: https://doi.org/10.1787/56003a32-en. Accessed: 15 April 2025.

Giang, H. T. T., Doan, T. V. & Le, T. Q. (2024), ‘An experimental study on fairness-aware machine learning for credit scoring problem’, *arXiv preprint arXiv:2412.20298* . Available at: https://arxiv.org/abs/2412.20298.

Gomber, P., Koch, J.-A. & Siering, M. (2017), ‘Digital finance and fintech: Current research and future research directions’, *Journal of Business Economics* **87**(5), 537–580.

Gowda, P. & Gowda, A. N. (2024), ‘Benefits and risks of generative ai in fintech’, *Journal of Scientific and Engineering Research* **11**(5), 267–275.

Hidayat, M., Defitri, S. & Hilman, H. (2024), ‘The impact of artificial intelligence (ai) on financial management’, *Management Studies and Business Journal (PRODUCTIVITY)* **1**, 123–129.

Hochreiter, S. & Schmidhuber, J. (1997), ‘Long short-term memory’, *Neural Computation* **9**, 1735–1780.

Intuit (2023), ‘Mint is transitioning to credit karma’, https://mint.intuit.com/mint-transition/. Accessed: 2025-04-12.

Intuit Inc. (2023), ‘Mint personal finance app’. Available at: https://mint.intuit.com.

Iyelolu, T. & Paul, P. (2024), ‘Disruption of traditional banking by fintech: A review and financial analysis’, *Open Access Research Journal of Science and Technology* **11**, 055–063.

Khandani, A. E., Kim, A. J. & Lo, A. W. (2010), 'Consumer credit-risk models via machine-learning algorithms', *Journal of Banking Finance* **34**(11), 2767–2787.

Lee, I. & Shin, Y. J. (2017), 'Fintech: Ecosystem, business models, investment decisions, and challenges', *Business Horizons* **61**.

Li, K. (2023), 'Prediction of stock prices based on lstm and xgboost algorithms', *Asian Economic and Financial Review* **13**(5), 453–462. Available at: `https://www.ewadirect.com/proceedings/aemps/article/view/8624` (Accessed: 12 April 2025).

Lusardi, A. & Mitchell, O. (2014), 'The economic importance of financial literacy: Theory and evidence', *Journal of Economic Literature* **52**(1), 5–44.

McKinsey & Company (2024), 'The state of ai: How organizations are rewiring to capture value'. Available at: `https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai`.

Molnar, C. (2025), *Interpretable Machine Learning: A Guide For Making Black Box Models Explainable*, Christoph Molnar. Available at: `https://christophm.github.io/interpretable-ml-book/` (Accessed: 15 April 2025).

Ng, M. Y., Coopamootoo, K., Toreini, E., Aitken, M., Elliot, K. & van Moorsel, A. (2020), 'Simulating the effects of social presence on trust, privacy concerns & usage intentions in automated bots for finance', *arXiv preprint arXiv:2006.15449* . Available at: `https://arxiv.org/abs/2006.15449`.

Nicoletti, B. (2017), *The Future of FinTech*, Springer, Cham, Switzerland. Published: January 2017.

Nielsen, D. (2016), Tree boosting with xgboost – why does xgboost win "every" machine learning competition?, Master's thesis, Norwegian University of Science and Technology (NTNU). Available at: `https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2433761` (Accessed: 12 April 2025).

Omarini, A. (2017), 'The digital transformation in banking and the role of fintechs in the new financial intermediation scenario', *International Journal of Finance, Economics and Trade (IJFET)* **Int J Finance, Economics and Trade**.

PocketGuard Inc. (2023), 'Pocketguard app overview'. Available at: `https://pocketguard.com`.

Reuters (2024), 'Legal transparency in ai finance: facing the accountability dilemma in digital decision-making', *Reuters* . Available at: `https://www.reuters.com/legal/transactional/legal-transparency-ai-finance-facing\protect\penalty\z@-accountability-dilemma-digital-decision-2024-03-01/`.

Shneiderman, B. (2020), 'Human-centered artificial intelligence: Reliable, safe & trustworthy', *International Journal of Human–Computer Interaction* **36**(6), 495–504.

Siami Namini, S., Tavakoli, N. & Siami Namin, A. (2018), A comparison of arima and lstm in forecasting time series, *in* '2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)', IEEE, Orlando, FL, USA, pp. 1394–1401.

Spendee Ltd. (2023), 'Spendee: Budget and expense tracker'. Available at: `https://www.spendee.com`.

Tanda, A. & Schena, C.-M. (2019), *FinTech, BigTech and Banks: Digitalisation and Its Impact on Banking Business Models*, Palgrave Macmillan Studies in Banking and Financial Institutions, 1 edn, Palgrave Pivot Cham, Cham. eBook ISBN: 978-3-030-22426-4, Published: 30 July 2019. Available at: `https://doi.org/10.1007/978-3-030-22426-4` (Accessed: 12 April 2025).

You Need A Budget (2023), 'Ynab – you need a budget'. Available at: `https://www.youneedabudget.com`.

Zarifis, A. & Cheng, X. (2024), 'How to build trust in answers given by generative ai for specific, and vague, financial questions', *arXiv preprint arXiv:2408.14593* . Available at: `https://arxiv.org/abs/2408.14593`.

Zou, H., Xiao, Y., Wang, L. & Liu, H. (2022), 'A comparative study of machine learning algorithms for credit scoring: A case study using xgboost', *Applied Sciences* **12**(15), 7720.

Appendix A: FinTrack UI/UX Screenshots

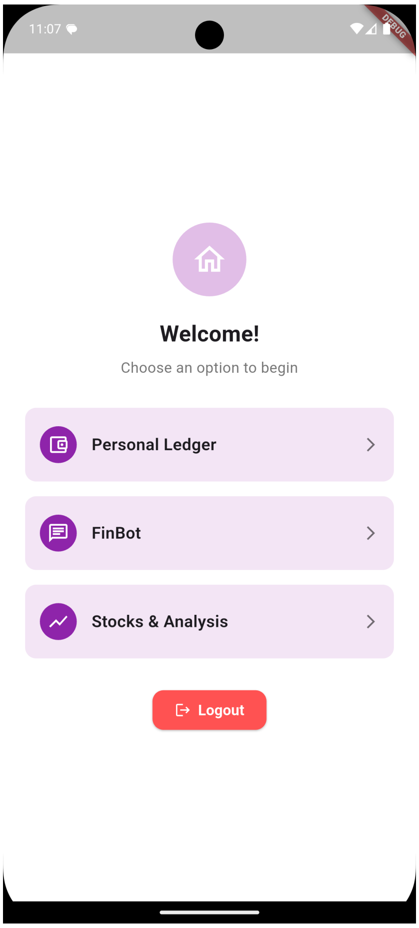


Figure B.1: Home screen of the FinTrack app showing navigation to Personal Ledger, FinBot, and Stock Analysis features.

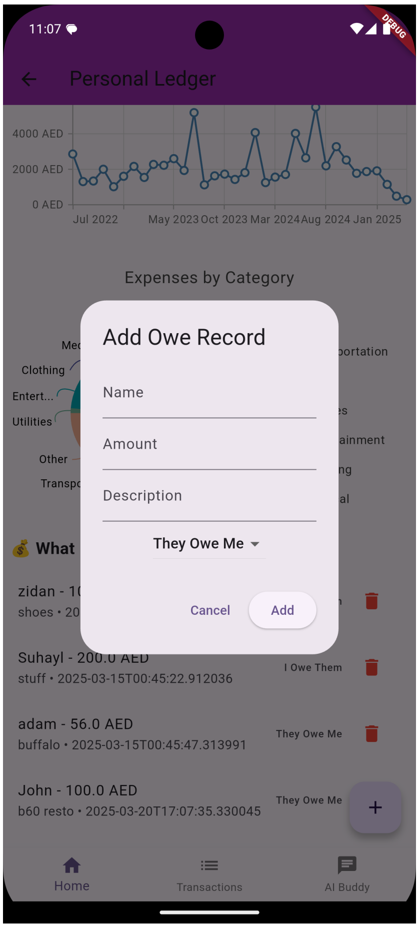


Figure B.2: Add Owe Record screen in the Personal Ledger module, enabling users to log debts with name, amount, and description.

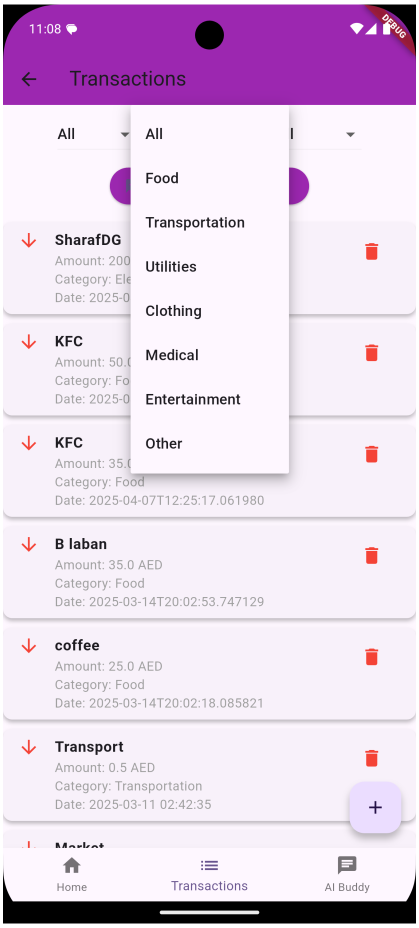


Figure B.3: Transaction history with category filter dropdown to view expenses under specific categories such as Food, Utilities, etc.

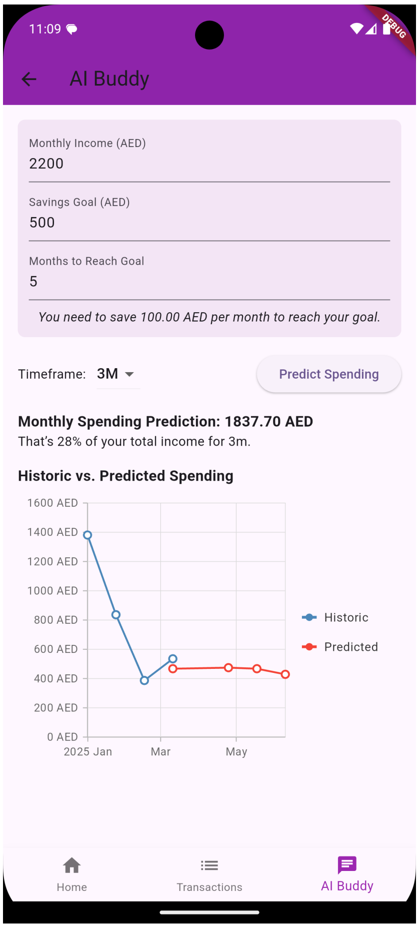
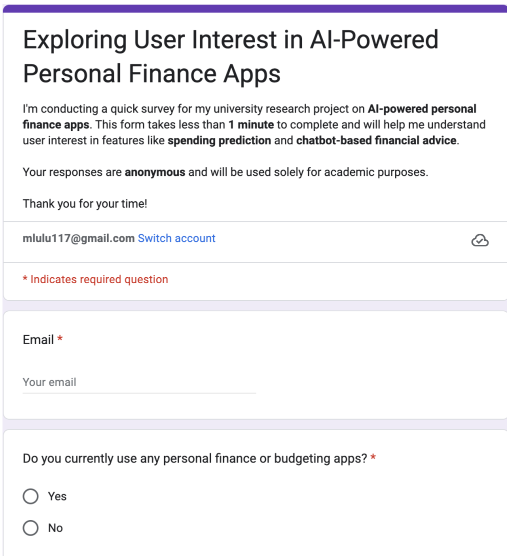


Figure B.4: AI Buddy interface showing monthly savings goal calculator and visual comparison between historic and predicted spending.

Appendix B: User Survey Form



Exploring User Interest in AI-Powered Personal Finance Apps

I'm conducting a quick survey for my university research project on **AI-powered personal finance apps**. This form takes less than **1 minute** to complete and will help me understand user interest in features like **spending prediction** and **chatbot-based financial advice**.

Your responses are **anonymous** and will be used solely for academic purposes.

Thank you for your time!

mlulu117@gmail.com Switch account

* Indicates required question

Email *

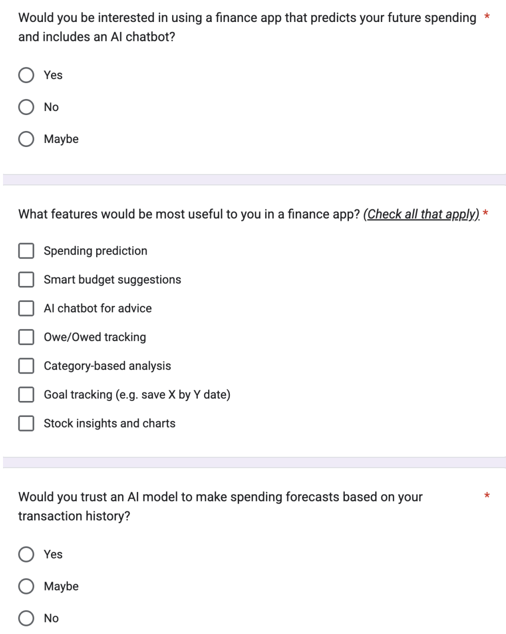
Your email

Do you currently use any personal finance or budgeting apps? *

Yes

No

Figure C.1: Introduction and first question from the Google Form used to collect survey data on user interest in AI-powered finance apps.



Would you be interested in using a finance app that predicts your future spending and includes an AI chatbot? *

- ◯ Yes
- ◯ No
- ◯ Maybe

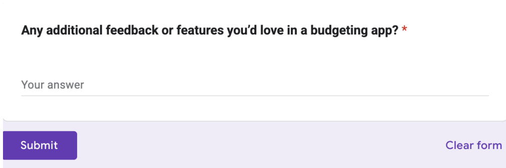
What features would be most useful to you in a finance app? *(Check all that apply)* *

- ☐ Spending prediction
- ☐ Smart budget suggestions
- ☐ AI chatbot for advice
- ☐ Owe/Owed tracking
- ☐ Category-based analysis
- ☐ Goal tracking (e.g. save X by Y date)
- ☐ Stock insights and charts

Would you trust an AI model to make spending forecasts based on your transaction history? *

- ◯ Yes
- ◯ Maybe
- ◯ No

Figure C.2: Key questions assessing user interest in features such as AI chatbots, spending prediction, and budgeting tools.



Any additional feedback or features you'd love in a budgeting app? *

Your answer

Submit Clear form

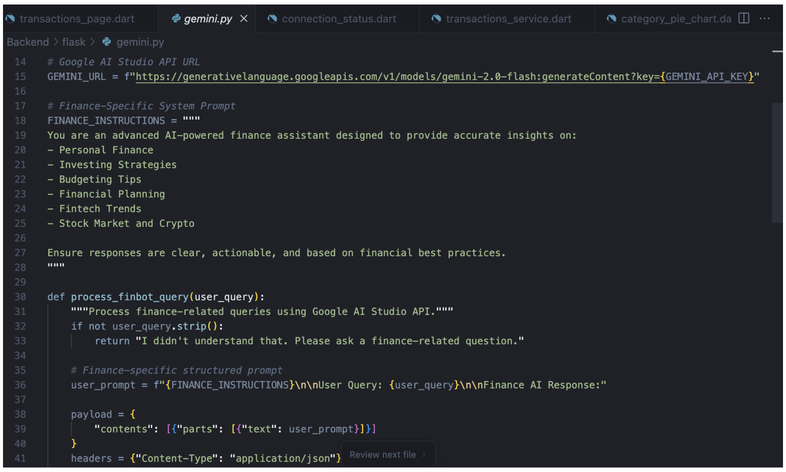
Figure C.3: Final question in the form allowing for open-ended feedback on desired features in budgeting apps.

Appendix C: Relevant Links

The complete source code and project files for the FinTrack application can be accessed through the University of Birmingham's GitLab repository at the following link:

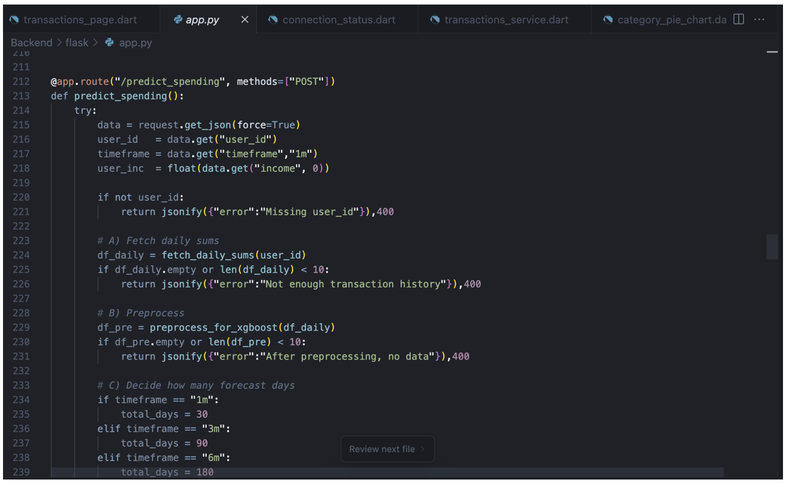
FinTrack Project Repository:
`https://git.cs.bham.ac.uk/projects-2024-25/mxm1399`

Appendix D: Code Snippets and Backend Implementation



```python
# Google AI Studio API URL
GEMINI_URL = f"https://generativelanguage.googleapis.com/v1/models/gemini-2.0-flash:generateContent?key={GEMINI_API_KEY}"

# Finance-Specific System Prompt
FINANCE_INSTRUCTIONS = """
You are an advanced AI-powered finance assistant designed to provide accurate insights on:
- Personal Finance
- Investing Strategies
- Budgeting Tips
- Financial Planning
- Fintech Trends
- Stock Market and Crypto

Ensure responses are clear, actionable, and based on financial best practices.
"""

def process_finbot_query(user_query):
    """Process finance-related queries using Google AI Studio API."""
    if not user_query.strip():
        return "I didn't understand that. Please ask a finance-related question."

    # Finance-specific structured prompt
    user_prompt = f"{FINANCE_INSTRUCTIONS}\n\nUser Query: {user_query}\n\nFinance AI Response:"

    payload = {
        "contents": [{"parts": [{"text": user_prompt}]}]
    }
    headers = {"Content-Type": "application/json"}
```

Figure D.1: ‘gemini.py‘ – Finance-specific prompt and Google Gemini API request handler for FinBot queries.



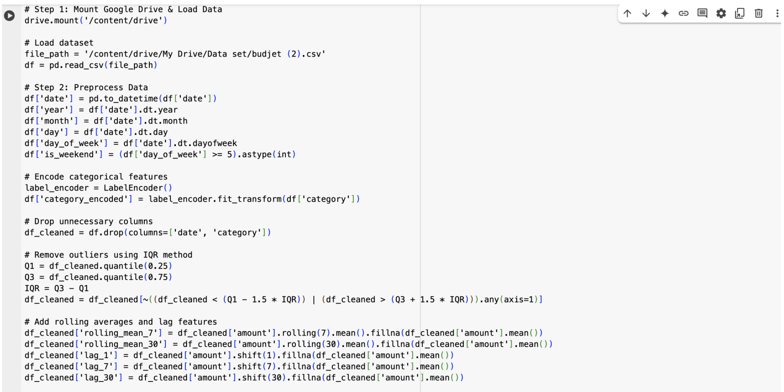
transactions_page.dart app.py connection_status.dart transactions_service.dart category_pie_chart.da

Backend > flask > app.py

```python

@app.route("/predict_spending", methods=["POST"])
def predict_spending():
    try:
        data = request.get_json(force=True)
        user_id   = data.get("user_id")
        timeframe = data.get("timeframe","1m")
        user_inc  = float(data.get("income", 0))

        if not user_id:
            return jsonify({"error":"Missing user_id"}),400

        # A) Fetch daily sums
        df_daily = fetch_daily_sums(user_id)
        if df_daily.empty or len(df_daily) < 10:
            return jsonify({"error":"Not enough transaction history"}),400

        # B) Preprocess
        df_pre = preprocess_for_xgboost(df_daily)
        if df_pre.empty or len(df_pre) < 10:
            return jsonify({"error":"After preprocessing, no data"}),400

        # C) Decide how many forecast days
        if timeframe == "1m":
            total_days = 30
        elif timeframe == "3m":
            total_days = 90
        elif timeframe == "6m":
            total_days = 180
```

Review next file

Figure D.2: ‘app.py‘ – Spending prediction endpoint using Flask, parsing input, validating data, and routing to the XGBoost model.



```python
# Step 1: Mount Google Drive & Load Data
drive.mount('/content/drive')

# Load dataset
file_path = '/content/drive/My Drive/Data set/budjet (2).csv'
df = pd.read_csv(file_path)

# Step 2: Preprocess Data
df['date'] = pd.to_datetime(df['date'])
df['year'] = df['date'].dt.year
df['month'] = df['date'].dt.month
df['day'] = df['date'].dt.day
df['day_of_week'] = df['date'].dt.dayofweek
df['is_weekend'] = (df['day_of_week'] >= 5).astype(int)

# Encode categorical features
label_encoder = LabelEncoder()
df['category_encoded'] = label_encoder.fit_transform(df['category'])

# Drop unnecessary columns
df_cleaned = df.drop(columns=['date', 'category'])

# Remove outliers using IQR method
Q1 = df_cleaned.quantile(0.25)
Q3 = df_cleaned.quantile(0.75)
IQR = Q3 - Q1
df_cleaned = df_cleaned[~((df_cleaned < (Q1 - 1.5 * IQR)) | (df_cleaned > (Q3 + 1.5 * IQR))).any(axis=1)]

# Add rolling averages and lag features
df_cleaned['rolling_mean_7'] = df_cleaned['amount'].rolling(7).mean().fillna(df_cleaned['amount'].mean())
df_cleaned['rolling_mean_30'] = df_cleaned['amount'].rolling(30).mean().fillna(df_cleaned['amount'].mean())
df_cleaned['lag_1'] = df_cleaned['amount'].shift(1).fillna(df_cleaned['amount'].mean())
df_cleaned['lag_7'] = df_cleaned['amount'].shift(7).fillna(df_cleaned['amount'].mean())
df_cleaned['lag_30'] = df_cleaned['amount'].shift(30).fillna(df_cleaned['amount'].mean())
```

Figure D.3: Google Collab – Data preprocessing steps: encoding, cleaning, feature engineering for time series input.

```python
# Step 6: Train Model with Best Hyperparameters
best_model = XGBRegressor(**best_params, random_state=42)
best_model.fit(X_train, y_train)

# Step 7: Model Evaluation
y_pred = best_model.predict(X_test)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
mape = mean_absolute_percentage_error(y_test, y_pred)

print(f"MAE: {mae}")
print(f"MSE: {mse}")
print(f"RMSE: {rmse}")
print(f"R²: {r2}")
print(f"MAPE: {mape * 100:.2f}%")

# ✅ Step 10: **Visualize Results**

## 📊 1. **Actual vs Predicted Spending Over Time**
plt.figure(figsize=(12, 6))
plt.plot(y_test.values, label="Actual Spending", color='blue')
plt.plot(y_pred, label="Predicted Spending", color='red', linestyle='dashed')
plt.axvline(len(y_test) * 0.8, color='green', linestyle='dashed', label='Past Data Ends, Future Starts')
plt.legend()
plt.title("Actual vs Predicted Spending (Including Future Forecast)")
plt.xlabel("Time")
plt.ylabel("Spending Amount")
plt.show()

## 📊 2. **Feature Importance**
importances = best_model.feature_importances_
features = df_cleaned.drop(columns=['amount']).columns
```

Figure D.4: Google Collab – Model evaluation using MAE, RMSE, R^2, and MAPE. Output includes prediction plots and feature importance.