

# **MoneyGuard – Finance Management with Stock Prediction**

*Major Project-II Report*

*Submitted in partial fulfillment of the requirement*

*for the degree of*

**Bachelor of Technology**

**In**

**Artificial Intelligence and Data Science**

January-June 2025

**Guided By:**

Prof. Priyanka Maheshwari

**Submitted By:**

Arpita Vishwakarma  
[0704AD211009]



Department of Artificial Intelligence and Data Science

Mahakal Institute of Technology, Ujjain

Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal



🛡️ **MONEYGUARD** 🛡️

*"Smart Finance, Smarter Investments."*

💰 **Integrated Personal Finance & Stock Prediction System** 💰

🔍 **Track | Analyze | Predict** 🔍

## PROJECT APPROVAL SHEET

The project entitled “MoneyGuard : Finance Management with Stock Prediction” submitted by Arpita vishwakarma as partial fulfillment for the award of **Bachelor of Technology in Artificial Intelligence and Data Science** by Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal.

Internal Examiner

Date :

External Examiner

Date :

## RECOMMENDATION

The project entitled “MoneyGuard : Finance Management with Stock Prediction” submitted by Arpita vishwakarma as partial is a satisfactory account of the bonafide work done under our guidance is recommended towards partial fulfillment for the award of the **Bachelor of Technology in Artificial Intelligence and Data Science** from Mahakal Institute of Technology, Ujjain by Rajiv Gandhi Proudhyogiki Vishwavidyalaya, Bhopal.

### Project Guide

Prof. Priyanka Maheshwari

Date :

Sign :

### Project Coordinator

Prof. Neeraj Sharma

Date :

Sign :

### Endorsed By

Dr. Kamlesh Ahuja

Head

Department of Artificial Intelligence and Data Science

Mahakal Institute of Technology, Ujjain

## ACKNOWLEDGEMENT

It is with great reverence that we express our gratitude to our guide “**Prof. Priyanka Maheshwari**” Department of Artificial Intelligence and Data Science, Mahakal Institute of Technology, Ujjain, (M.P.) for her precious guidance and help in this project work. The credit for the successful completion of this project goes to her keen interest timing guidance and valuable suggestion otherwise our endeavor would have been futile.

We sincerely thank “**Prof. Priyanka Maheshwari**” for his guidance and encouragement in carrying out this project work.

We owe to regard to “**Dr. Kamlesh Ahuja**” Head of Department, Information Technology for his persistent encouragement and blessing which were bestowed upon us.

We owe our sincere thanks to honorable Director “**Dr. Mukesh Shukla**” for his kind support which he rendered us in the envisagement for great success of our project.

**Arpita Vishwakarma**  
[0704AD211009]

## TABLE OF CONTENTS

	Page No.
Project Approval Sheet	I
Recommendation	II
Acknowledgement	III
Table of Contents	IV
Abstract	VI
Figure Index	VII
Table Index	VIII
List of Abbreviations	VIII
<b>Chapter 1 Introduction</b>	<b>(1-3)</b>
1.1 Introduction	1
1.2 Identification of Problem Domain	2
<b>Chapter 2 Literature Review</b>	<b>(4-6)</b>
2.1 Literature Review	4
2.1.1 Study of personal finance management tools	4
2.1.2 Study of stock prediction and investment guidance	4
2.1.3 Study of data privacy and security in financial application	5
2.1.4 Study of financial literacy and decision making	5
2.2 Limitation of existing system	5
<b>Chapter 3 Rationale and Process</b>	<b>(7-9)</b>
3.1 Objective	7
3.2 Software Model Adapted	8
<b>Chapter 4 System Analysis Overview</b>	<b>(10-19)</b>
4.1 Requirement Analysis	10

4.1.1 Hardware Requirement	10
4.1.2 Software Requirement	10
4.1.3 Functional & Non-functional Requirements	11
4.2 Use -Case diagram & Use -Case description	11
4.3 Sequence Diagram	15
4.4 System Flow Diagram	17
<b>Chapter 5 System Design Overview</b>	<b>( 20-30 )</b>
5.1 Class Diagram	20
5.2 Data Flow Diagram	23
5.3 Data Dictionary	27
5.4 Extended ER Diagram	29
<b>Chapter 6 Workplan and System Database Structure</b>	<b>( 31-34 )</b>
6.1 Time Frame Work	31
6.2 Design Display Table	34
<b>Chapter 7 Testing Strategy Adapted</b>	<b>( 35-36 )</b>
7.1 Testing Strategy Adapted	35
7.2 System Testing	36
7.3 Test Cases	36
7.4 Deployment Strategy	37
<b>Chapter 8 Conclusion and Future Scope</b>	<b>( 38-39 )</b>
8.1 Conclusion	38
8.2 Future Scope	38
<b>REFERENCES</b>	<b>40</b>
<b>APPENDIX A ( Screenshots )</b>	<b>( 41-54 )</b>

## ABSTRACT

MoneyGuard is an integrated financial management and stock prediction platform designed to empower users with efficient money management tools and AI-powered stock forecasting. It is built using HTML, CSS, JavaScript, and Bootstrap for the frontend, Flask/Django(python framework) for the backend and provides a seamless user experience with secure financial tracking and real-time investment insights.

The platform consists of two core functionalities: finance management and stock prediction. In the finance management module, users must first register using their email to access MoneyGuard's financial tools. A dashboard presents an overview of total transactions, expenses, and financial trends through visual graphs, allowing users to monitor their financial health easily. The transaction module enables users to log transactions, specifying details such as date, category, amount, and payment method. Additionally, a statistics section categorizes and summarizes expenses, providing valuable insights into spending habits and financial planning.

The stock prediction module enhances investment decision-making by leveraging LSTM-based machine learning models to predict real-time stock prices. The system fetches historical stock data, visualizing past trends to help users make informed investment choices. A next-day price prediction feature assists users in anticipating market movements, giving them an edge in stock trading.

Security is a top priority in MoneyGuard, implementing AES-256 encryption and HTTPS protocols to safeguard user data. The platform ensures financial transparency and helps users make data-driven decisions with confidence.

By integrating finance management with intelligent stock predictions, MoneyGuard serves as a comprehensive tool for individuals seeking to optimize their budgets and enhance their investment strategies.

**Keywords:** Finance management, stock prediction, machine learning, LSTM, transaction tracking, expense analysis, real-time stock forecasting, data security, financial insights.



## FIGURE INDEX

Figure No.	Title of Figure	Page No.
Fig 1.1	Finance Management	1
Fig 1.2	Stock Predictions	1
Fig 1.3	Agile Software Development Model Adapted	8
Fig 1.4	Use Case Diagram	15
Fig 1.5	Sequence Diagram	17
Fig 1.6	System Flow Diagram	19
Fig 1.7	Class Diagram	23
Fig 1.8	Data Flow Diagram [Level 0]	24
Fig 1.9	Data Flow Diagram [Level 1]	25
Fig 2.0	Data Flow Diagram [Level 2]	26
Fig 2.1	Extended ER Diagram	30

## TABLE INDEX

<b>Table No.</b>	<b>Title of Table</b>	<b>Page No.</b>
Table 1.1	Comparison between MoneyGuard and the Existing Systems	6
Table 1.2	Data Dictionary for User Module	27
Table 1.3	Data Dictionary for Expense Management	27
Table 1.4	Data Dictionary for Budget and Financial Goals	28
Table 1.5	Data Dictionary for Stock Prediction Module	28
Table 1.6	Data Dictionary for Security and Session	28
Table 1.7	Data Dictionary for Machine Learning Module	29
Table 1.8	Data Display Table	34
Table 1.9	Types of Testing in MoneyGuard	35
Table 2.0	Test Cases involved in MoneyGuard	36

## LIST OF ABBREVIATIONS

MONEYGUARD	Money Optimization Network and Enhanced Yield Guide for Users' Assistance in Risk Detection
YNAB	You need a Budget

## CHAPTER 1

### INTRODUCTION

#### 1.1 INTRODUCTION

In today's dynamic financial landscape, effective personal finance management and informed investment decisions are crucial for financial stability. With increasing financial responsibilities and volatile stock markets, individuals require intelligent digital solutions that not only track expenses but also provide actionable investment insights. **MoneyGuard** is designed to bridge this gap by integrating finance management with stock prediction, empowering users to optimize their budgets while making data-driven investment choices for better financial growth and security.

This project falls within the **fintech domain**, where technology-driven financial solutions are reshaping personal money management. With advancements in artificial intelligence and machine learning, predictive analytics have become a game-changer for both budgeting and investing. Research suggests that financial planning tools enhance decision-making, improve savings, and contribute to long-term wealth generation, emphasizing the growing need for personalized and data-driven finance platforms that cater to modern financial challenges.



Fig 1.1 Finance Management



Fig 1.2 Stock Predictions

MoneyGuard stands out by offering a seamless blend of expense tracking and AI-powered stock forecasting. Unlike many existing financial apps that focus solely on budgeting or investments, MoneyGuard provides an integrated, secure, and user-friendly solution. Utilizing LSTM-based machine learning models, real-time stock data, and AES-256 encryption, the platform ensures accurate predictions and robust data security. The system allows users to register securely, track transactions, monitor spending habits through interactive graphs, and gain valuable financial insight

Users can also analyze historical stock trends, predict future stock prices, and receive tailored investment recommendations based on market conditions.

Security and privacy are critical aspects of digital finance management, and MoneyGuard prioritizes user data protection by implementing advanced encryption protocols and secure authentication mechanisms. Many existing financial apps fail to offer comprehensive security measures, leaving users vulnerable to cyber threats and unauthorized access. MoneyGuard addresses this issue by integrating HTTPS security, encrypted databases, and secure login authentication, ensuring that user data remains safe from breaches.

Additionally, MoneyGuard's intuitive and user-friendly dashboard presents financial summaries, categorized expenses, and stock market trends in a visually appealing format, making it easier for users to understand their financial health. The platform eliminates the need for multiple financial apps by consolidating expense management, investment insights, and stock market predictions into one streamlined interface.

By equipping users with the right financial insights and tools, MoneyGuard fosters financial literacy, enabling individuals to take control of their financial future with confidence, efficiency, and reliability. Whether for daily expense tracking or long-term investment planning, MoneyGuard serves as a powerful and intelligent financial assistant that enhances money management and investment strategies in a secure, user-friendly, and data-driven manner.

## **1.2 IDENTIFICATION OF PROBLEM DOMAIN**

In the digital era, managing personal finances effectively while making informed investment decisions has become increasingly challenging. Individuals face difficulties in tracking expenses, analyzing financial trends, and making data-driven investment choices due to the absence of integrated tools that cater to both finance management and stock prediction. Although numerous financial applications exist, most focus solely on basic expense tracking and budgeting, lacking real-time investment insights and predictive analytics. As a result, users often rely on multiple fragmented platforms, leading to inefficiencies in managing their finances and investments.

A significant gap in the current financial technology landscape is the lack of AI-driven stock predictions within finance management tools. Many users struggle with investment decisions due to the absence of reliable, machine learning-based stock forecasting, often leading to missed opportunities or poor investment choices. Additionally, data security remains a major concern, as financial platforms store sensitive personal and transactional information. Many existing applications do not provide strong encryption or secure authentication mechanisms, increasing the

risk of data breaches and unauthorized access. With cyber threats and financial fraud on the rise, users need a highly secure platform that ensures data privacy while providing meaningful financial insights.

Furthermore, most existing solutions fail to offer a user-friendly and visually intuitive interface that simplifies financial management. Many platforms overwhelm users with complex dashboards and unstructured data, making it difficult to interpret financial trends effectively. A well-structured, easy-to-use system that provides real-time financial insights through interactive graphs, statistics, and categorized expenses is essential for better financial decision-making.

Another critical issue is the lack of automation and smart analytics in financial management. Most personal finance applications require users to manually input and track their expenses, which can be time-consuming and prone to errors. Additionally, these platforms do not leverage AI-powered insights to provide users with personalized recommendations for saving, budgeting, or investing. Users often struggle to analyze financial patterns and optimize their spending habits due to the absence of advanced analytics tools. A system that automates financial tracking, categorizes expenses intelligently, and suggests investment opportunities based on market trends would significantly enhance financial planning and decision-making.

To address these challenges, MoneyGuard provides a unified, secure, and AI-powered platform that seamlessly integrates expense tracking, financial analytics, and real-time stock price predictions. By leveraging LSTM-based machine learning models, MoneyGuard offers users accurate stock forecasts, helping them make informed investment decisions. The platform also prioritizes data security through AES-256 encryption and HTTPS protocols, ensuring that user information remains protected. Additionally, the platform provides an intuitive and interactive dashboard, displaying financial summaries and market trends in a user-friendly format.

By combining finance management with intelligent stock predictions, MoneyGuard empowers individuals with financial literacy, better control over their budgets, and the confidence to navigate the stock market efficiently. The integration of cutting-edge technologies with practical financial tools makes MoneyGuard a comprehensive, secure, and future-ready financial assistant that bridges the gap between personal finance management and smart investing.

## **CHAPTER 2**

### **LITERATURE REVIEW**

#### **2.1 LITERATURE REVIEW**

The integration of financial management and stock prediction has gained significant attention in the fintech industry. Various applications offer budgeting tools, stock trading, or investment recommendations, but very few provide a unified platform that combines all these features with predictive analytics. Research highlights the importance of real-time financial tracking, AI-powered stock forecasting, data security, and financial literacy in modern financial applications. This section explores existing solutions, identifies their limitations, and discusses how MoneyGuard addresses these gaps by offering an all-in-one financial management and stock prediction system.

##### **2.1.1. Study of Personal Finance Management Tools**

Personal finance management tools help users track expenses, create budgets, and analyze spending patterns. Applications like Mint, YNAB (You Need a Budget), and PocketGuard provide basic features such as categorizing transactions, setting financial goals, and monitoring cash flow. However, these platforms primarily focus on budgeting and lack real-time investment tracking or predictive analytics for stock prices. Users who want a more integrated approach often have to rely on separate applications for financial management and investment tracking, leading to fragmented financial planning. MoneyGuard addresses this gap by combining financial management with stock prediction, enabling users to track expenses and make informed investment decisions from a single platform.

##### **2.1.2. Study of Stock Prediction and Investment Guidance**

Investment platforms like Robinhood, E\*TRADE, and Webull allow users to trade stocks and manage portfolios but do not provide predictive analytics for stock price forecasting. Research suggests that machine learning algorithms, particularly Long Short-Term Memory (LSTM) networks, can significantly improve stock price prediction accuracy by analyzing historical trends and market fluctuations. Several studies have demonstrated that deep learning models outperform traditional statistical methods in forecasting financial markets. MoneyGuard incorporates LSTM-based stock prediction to offer real-time stock analysis, historical data insights, and next-day price forecasts, helping users make data-driven investment decisions.

### **2.1.3. Study of Data Privacy and Security in Financial Applications**

With the increasing use of digital financial platforms, data security and privacy concerns have become critical. Many fintech applications lack strong encryption measures and sufficient authentication mechanisms, making users vulnerable to cyber threats and financial fraud. Studies indicate that AES-256 encryption and HTTPS protocols are essential for ensuring secure transactions and protecting user data from unauthorized access. MoneyGuard addresses these concerns by implementing AES-256 encryption for data storage, HTTPS for secure communication, and email authentication for user access, ensuring a safe and reliable financial management experience.

### **2.1.4. Study of Financial Literacy and Decision-Making**

A significant number of users lack the necessary financial knowledge to make informed budgeting and investment decisions. Platforms like Acorns and Stash offer micro-investing and savings features but do not focus on educating users about investment risks and strategies. Research highlights that providing personalized financial insights, interactive data visualizations, and educational content can greatly improve financial literacy and decision-making. MoneyGuard bridges this gap by offering interactive dashboards, categorized expense analysis, and investment insights, enabling users to understand their financial patterns and optimize their spending and investment choices.

By integrating finance management, stock prediction, data security, and financial education, MoneyGuard stands out as a comprehensive fintech solution that not only tracks user finances but also empowers them with intelligent investment strategies and robust security measures.

## **2.2 LIMITATION OF EXISTING SYSTEM**

Despite the availability of numerous financial management and stock investment platforms, most existing solutions fail to offer a comprehensive, secure, and intelligent approach to managing finances and making investment decisions. The following are key limitations of current financial and stock prediction systems:

### **1. Lack of Unified Financial Management and Stock Prediction**

Most platforms either focus on expense tracking and budgeting (such as Mint and YNAB) or investment and trading (such as Robinhood and Webull). Very few integrate both features, requiring users to manage their personal finances and investments separately, leading to inconvenience and fragmented financial planning.

## 2. **Limited Predictive Analytics in Investment Platforms**

Many investment platforms allow stock trading but do not provide AI-driven stock price predictions. Without predictive analytics, users lack foresight into market trends, making investment decisions more speculative rather than data-driven.

## 3. **Weak Data Security Measures**

Several fintech applications fail to implement robust security measures, exposing users to potential data breaches, identity theft, and financial fraud. Without AES-256 encryption, HTTPS protection, and multi-factor authentication, sensitive financial information remains vulnerable.

## 4. **Inadequate Financial Literacy Support**

While some platforms provide investment tracking, they do not educate users about financial management, market trends, or investment strategies. A lack of personalized financial insights results in poor decision-making, making it difficult for users to improve their financial health.

## 5. **Absence of Personalized Recommendations**

Most existing financial platforms provide generic insights without considering user-specific financial habits, income, expenses, and investment goals. This lack of customized financial advice makes it difficult for users to optimize their finances effectively.

MoneyGuard overcomes these limitations by integrating financial management and stock prediction, leveraging machine learning for precise stock forecasting, ensuring top-tier security measures, and providing personalized financial insights to help users make informed financial decisions.

Feature	Existing Systems	MoneyGuard
Comprehensive Expense Management	Limited, lacks integration	Detailed tracking with categories and visualizations
Stock Prediction	Basic, generic predictions	AI-powered, personalized stock recommendations
Data Privacy	Basic security measures	Enhanced encryption and secure login
Multi-Platform Support	Often limited to mobile	Cross-platform (web and mobile) compatibility
User Guidance	Minimal investment advice	Insights and education for better financial literacy

Table 1.1 Comparison between MoneyGuard and the Existing Systems



## CHAPTER 3

### RATIONALE AND PROCESS

#### 3.1 OBJECTIVE

The primary objective of the "MoneyGuard" project is to provide an integrated, intelligent, and secure financial management system that helps users track their expenses, analyse their financial health, and make data-driven investment decisions using AI-powered stock prediction. The platform bridges the gap between personal finance tracking and predictive stock analytics, enabling users to manage their money efficiently while exploring profitable investment opportunities.

##### **Specific Objectives:**

1. Develop a Secure Finance Management System
  - Implement expense tracking, budgeting, and financial analytics in a single platform.
  - Allow users to log transactions, categorize expenses, and visualize their financial trends through interactive graphs.
2. Integrate AI-Based Stock Prediction
  - Utilize LSTM-based machine learning models to predict stock prices.
  - Enable users to access historical stock data, track real-time stock prices, and forecast next-day stock trends.
3. Enhance Data Privacy and Security
  - Implement AES-256 encryption, HTTPS protocols, and secure authentication mechanisms to protect user data.
  - Ensure that financial transactions and stock predictions remain confidential and tamper-proof.
4. Create an Interactive and User-Friendly Interface
  - Design a responsive web application using HTML, CSS, JavaScript, and Bootstrap for an enhanced user experience.
  - Provide an intuitive dashboard where users can track expenses and investment insights easily.
5. Empower Users with Financial Insights and Investment Awareness
  - Offer graphical representations of expenses and stock performance to help users identify spending patterns.
  - Guide users in making smarter investment decisions through personalized recommendations and stock market trend analysis.

By integrating finance management with AI-driven stock predictions, MoneyGuard offers a comprehensive and innovative solution that enhances financial security, promotes better budgeting, and empowers users with data-driven investment strategies.

### 3.2 SOFTWARE MODEL ADAPTED

For the development of MoneyGuard, an efficient software development model is essential to ensure scalability, maintainability, and security while incorporating finance management and AI-driven stock prediction. Given the dynamic nature of this project, the Agile Software Development Model is the most suitable approach. Agile offers iterative development, adaptability, and continuous improvement, making it ideal for a project that integrates real-time financial tracking, machine learning stock prediction, and secure transaction handling.

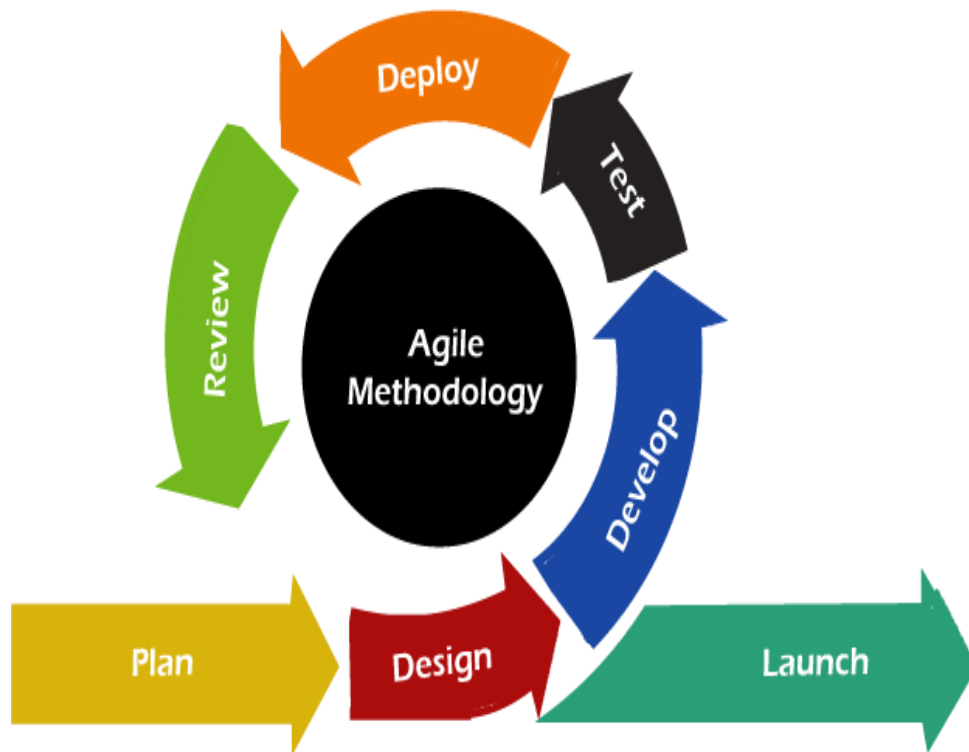


Fig 1.3 Agile Software development model approach

#### Why Agile for MoneyGuard?

##### 1. Incremental & Iterative Development

- Agile allows the development of finance tracking, stock prediction, and security features in stages, ensuring continuous testing and real-time feedback.
- Each sprint will focus on refining dashboard analytics, transaction tracking, stock market forecasting, and user experience to enhance the platform.

##### 2. User-Centric & Stakeholder Collaboration

- Agile prioritizes user feedback, which is crucial for improving data visualization, predictive stock models, and financial security protocols.
- Regular stakeholder involvement ensures that the platform meets evolving user expectations and industry standards.

### 3. Flexibility & Continuous Improvement

- The stock prediction model (LSTM) may need updates based on real-time market trends, requiring an adaptable development approach.
- Security enhancements like encryption, authentication mechanisms, and compliance with financial data regulations can be implemented dynamically.

### 4. Cross-Functional Team Collaboration

- Development involves a combination of backend engineers (Flask, SQLite), frontend developers (HTML, CSS, JS, Bootstrap), data scientists (LSTM model for stock prediction), and security experts.
- Agile fosters seamless collaboration among these teams, ensuring the best integration of all components.

### 5. Frequent Deliverables & Continuous Testing

- Agile follows a sprint-based approach, ensuring that features like transaction tracking, financial insights, stock price prediction, and secure authentication are developed in phases.
- Frequent testing and validation ensure accuracy in finance tracking and reliability in stock predictions.

By adopting the Agile methodology, MoneyGuard ensures a user-driven, secure, and efficient financial management system that adapts to real-world financial needs and stock market dynamics, ultimately empowering users with smart money management and investment tools. The iterative nature of Agile allows for continuous enhancements in stock prediction accuracy, financial insights, and security measures, ensuring the platform remains relevant and reliable.

Additionally, Agile's modular development approach facilitates seamless scalability, making it easier to integrate new financial services and AI-driven analytics in future updates. With a strong focus on performance optimization and user satisfaction, MoneyGuard is set to redefine digital financial management and investment tracking.

## **CHAPTER 4**

### **SYSTEM ANALYSIS OVERVIEW**

#### **4.1 REQUIREMENT ANALYSIS**

The requirement analysis phase is critical in defining the system's capabilities, ensuring that the MoneyGuard platform provides seamless finance management and stock prediction functionalities. This phase focuses on identifying the hardware and software requirements, as well as specifying the functional and non-functional aspects of the system.

##### **4.1.1 Hardware requirements**

To ensure efficient data processing, secure storage, and smooth user experience, the following hardware requirements are essential:

1. **Cloud Hosting:** A cloud-based infrastructure (e.g., AWS, Google Cloud, or Render) to host the Flask-based backend and ensure scalability.
2. **Database Server:** A SQLite database for storing user transactions, financial data, and stock prediction results securely.
3. **User Devices:** Web browsers (Chrome, Edge, Firefox) and mobile-friendly accessibility on smartphones, tablets, and desktops for enhanced usability.

##### **4.1.2 Software requirements**

To build a secure, scalable, and efficient system, the following technologies are required:

1. **Frontend Technologies:**
  - HTML, CSS, JavaScript, Bootstrap for a responsive UI.
  - Chart.js or D3.js for interactive financial graphs and stock trend visualizations.
2. **Backend Technologies:**
  - Flask (Python) to handle API requests and user authentication.
  - RESTful APIs to connect the frontend with backend services.
3. **Database Management System (DBMS):**
  - SQLite as the primary database for storing financial transactions and user data.
  - SQLAlchemy ORM for secure and efficient database management.

4. Machine Learning & Stock Prediction:

- Python (NumPy, Pandas, Scikit-learn, TensorFlow/Keras) for financial data analysis.
- LSTM (Long Short-Term Memory) model to predict next-day stock prices.
- Yahoo Finance API or Alpha Vantage API for fetching historical and real-time stock data.

5. Security Measures:

- HTTPS for secure communication between client and server.
- AES-256 encryption for sensitive financial data storage.
- OAuth-based authentication (Email sign-up/login).

### 4.1.3 Functional and non-functional requirements

Functional and non-functional requirements are essential components of system requirements that define what a system should do (functional) and how it should operate (non-functional).

#### 1. Functional requirements

These define the core features and operations that MoneyGuard must support:

[1] User Registration & Authentication:

- Secure email-based login and authentication system.
- Password encryption and multi-factor authentication (MFA) for added security.

[2] Expense Tracking & Budgeting:

- Users can log income and expenses with categories, payment methods, and notes.
- Budget allocation and spending insights for better financial planning.

[3] Dashboard & Financial Insights:

- Visual overview of total balance, income, and expenses.
- Interactive financial graphs and trend analytics.

[4] Stock Market Predictions & Investment Assistance:

- Users can enter a stock symbol to fetch real-time market data.
- Graphical visualization of past stock trends.
- AI-powered LSTM model predicts future stock prices.

[5] Secure Data Handling & User Privacy:

- End-to-end encryption for storing user financial data.
- Automatic backups for transaction records and stock data.

[6] Cross-Platform Accessibility:

- Compatible with web browsers and mobile devices for ease of access.

## 2. Non-functional requirements

These define performance, security, usability, and reliability factors:

### [1] Performance & Speed:

- The system must process financial transactions within 2 seconds.
- Stock price predictions must be generated within 5-10 seconds.

### [2] Scalability & Future Expansion:

- The backend should support thousands of users simultaneously.
- The system should allow for future enhancements, including cryptocurrency tracking and AI-driven investment advice.

### [3] Reliability & Availability:

- Ensure 99.9% uptime with regular data backups.
- Implement error-handling mechanisms to prevent crashes.

### [4] Security & Data Protection:

- AES-256 encryption for securing user financial records.
- Multi-factor authentication (MFA) for added user protection.
- Compliance with GDPR and financial security regulations.

### [5] Usability & User Experience (UX):

- The interface should be intuitive and user-friendly, even for non-technical users.
- Responsive web and mobile-friendly design for seamless experience.

### [6] Maintainability & Code Structure:

- The codebase should be modular and well-documented for easy updates.
- Future AI enhancements should be easily integrated into the system.

## 4.2 USE CASE DIAGRAM & USE-CASE DESCRIPTION

A Use Case Diagram is a visual representation of the interactions between users and the system. It defines the functionalities provided by the system and how different users (actors) interact with them. For MoneyGuard, the use case diagram plays a crucial role in outlining the core features of the platform, ensuring smooth financial management and stock prediction functionalities.

MoneyGuard is designed to be a user-friendly and secure financial management system that provides a seamless experience for tracking expenses, budgeting, and obtaining AI-driven stock predictions.

The system is primarily accessed by Users, managed by Admins, and integrates a Machine Learning (ML) System to generate predictive financial insights.

## 1) Actors in the System

The **primary actors** interacting with the system are:

### [1] User:

- The end-user who interacts with the platform for financial tracking and stock predictions.
- Can register, authenticate, manage budgets, track expenses, and receive stock predictions.

### [2] Admin:

- Oversees platform security, manages users, and maintains the system.
- Ensures data encryption and overall system functionality.

### [3] ML System:

- The AI-powered engine responsible for processing stock market data and generating predictions.
- Provides insights based on historical trends and financial analysis.

## 2) Description of Use cases

### [1] User Use Cases

The user interacts with the system for various financial management functionalities:

#### 1. Register/Authenticate

- Allows users to sign up, log in securely, and recover passwords.
- Uses encrypted credentials and multi-factor authentication for security.

#### 2. Manage Budget

- Users can set a monthly budget, update it, and track how expenses align with the budget.

#### 3. Track Expenses

- Enables users to log financial transactions, categorize expenses, and view spending patterns.
- Includes a dashboard displaying real-time expense summaries.

#### 4. Receive Stock Predictions

- Provides AI-driven stock market forecasts.
- Allows users to access real-time stock insights based on historical data.

#### 5. View Financial Insights

- Generates financial reports, graphical analysis, and recommendations for users.

#### 6. Update Profile

- Users can modify personal details, update security settings, and manage linked financial accounts.

## [2] Admin Use Cases

The **admin role** is essential for maintaining platform security, managing users, and ensuring smooth operations.

### 1. Data Encryption & Security

- Implements encryption (AES-256) to protect user financial data and transactions.
- Ensures secure communication via HTTPS protocols.

### 2. Manage Users

- Admins oversee user accounts, verify identities, and assist users in case of login or transaction issues.
- Handles account suspensions or restrictions if security threats are detected.

### 3. Update System

- The admin is responsible for maintaining system updates, fixing bugs, and improving security patches.
- Ensures seamless integration of new features and ML model updates.

## [3] Machine Learning (ML) System Use Cases

The **ML system** powers the financial prediction engine in MoneyGuard.

### 1. Stock Prediction Engine

- Uses LSTM-based machine learning models to analyze historical stock data.
- Provides real-time forecasts for selected stocks based on AI algorithms.

### 2. Financial Insight Generation

- Processes user financial data to generate reports and personalized insights.
- Recommends budgeting strategies based on spending habits.



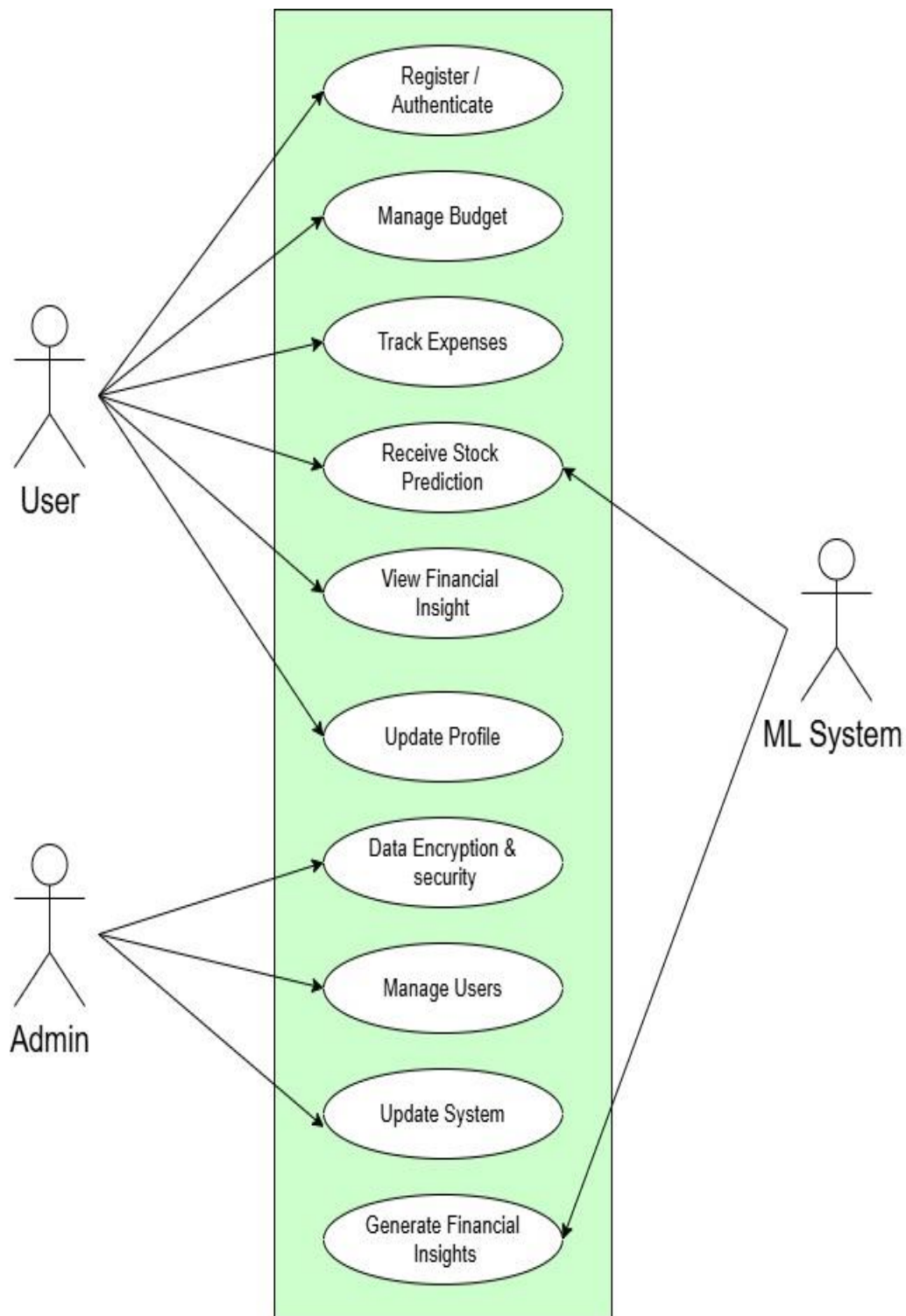


Fig 1.4 Use Case Diagram

### 4.3 SEQUENCE DIAGRAM

A Sequence Diagram illustrates the interaction between system components over time, demonstrating the sequential flow of operations. Based on the MoneyGuard project description and the Use Case Diagram, here's the updated sequence diagram flow:

## 1) Actors & Components

- [1] User – Performs actions like logging in, tracking expenses, and requesting stock predictions.
- [2] System – Handles authentication, processes transactions, and manages interactions between the User, Database, and Machine Learning System.
- [3] Database – Stores user credentials, transactions, and financial records.
- [4] Machine Learning System – Processes stock prediction requests and generates insights.

## 2) Sequence Flow description

### [1] User Login & Authentication

- User enters login credentials.
- System sends credentials to Database for verification.
- Database checks and confirms validity.
- System grants access on success; otherwise, an error message is sent.

### [2] Expense Tracking & Budget Management

- User enters expense details.
- System logs the expense and updates budget records.
- Database stores the expense and updates the user's financial balance.
- System confirms the update to the user.

### [3] Stock Prediction Request & Analysis

- User requests a stock prediction.
- System forwards the request to the Machine Learning System.
- ML System processes historical stock data and user preferences.
- ML System sends back the predicted stock insights.
- System displays predictions to the user.

### [4] Data Security & Encryption

- Every transaction and request is encrypted before sending data to the Database and ML System.
- The System ensures user data is securely processed, stored, and retrieved.

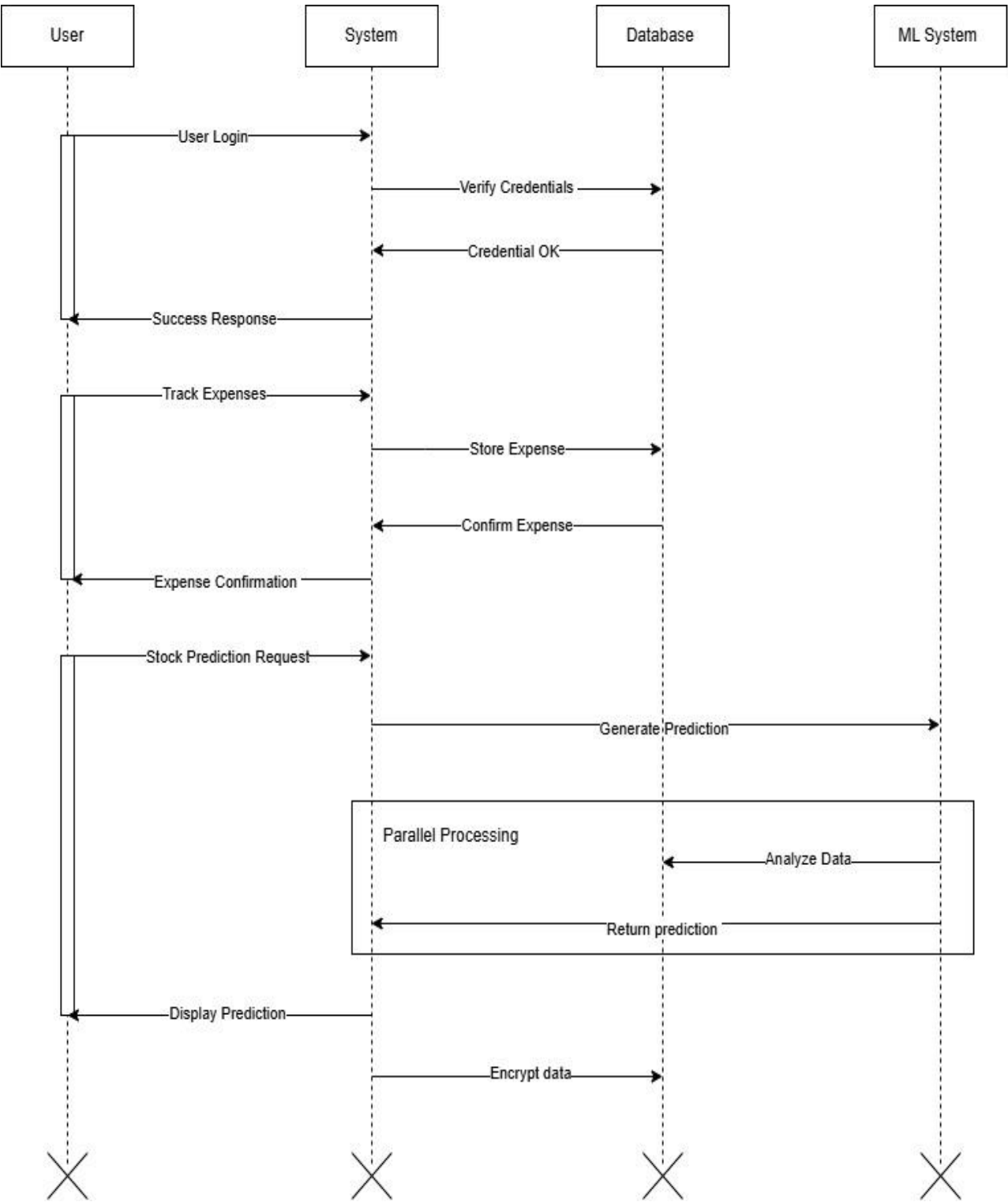


Fig 1.5 Sequence Diagram

4.4 SYSTEM FLOW DIAGRAM

A System Flow Diagram (SFD) visually represents the flow of data and processes within the MoneyGuard system. It helps illustrate how different components interact and how data is processed from input to output.

## 1) Overview of System Flow

The MoneyGuard system integrates personal finance management with stock prediction using machine learning. The flow diagram follows how users interact with the system, how data is processed, and how the system responds.

## 2) System Flow Description

### Step 1: User Authentication

- User registers or logs in.
- Credentials are verified through the database.
- If authentication is successful, access is granted; otherwise, an error is returned.

### Step 2: Expense Management

- User adds expenses with details (amount, category, date).
- Data is stored in the database.
- System updates financial summaries.

### Step 3: Budget Tracking & Insights

- System calculates and displays remaining balance.
- User views transaction history and financial insights.
- Data security ensures encryption of personal records.

### Step 4: Stock Prediction Request

- User requests stock prediction data.
- The system sends a request to the Machine Learning model.
- The ML model fetches historical stock data, processes predictions, and sends results.
- The system displays stock price forecasts.

### Step 5: Data Security & Storage

- All transactions, predictions, and user details are securely stored in the database.
- Encryption ensures data privacy.

### Step 6: Admin & System Maintenance

- Admin manages users, updates system security, and ensures system performance.
- The system undergoes periodic updates for feature improvements.

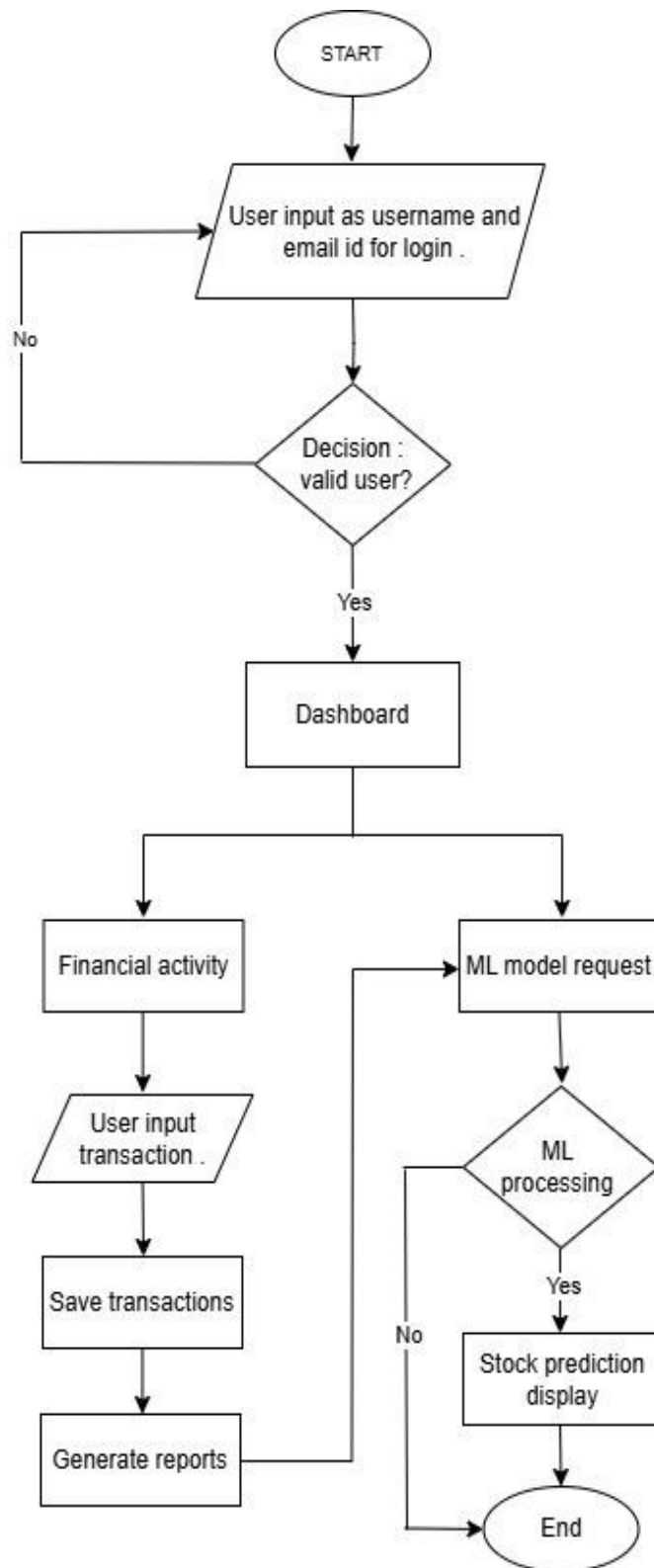


Fig 1.6 System Flow Diagram

## CHAPTER 5

### SYSTEM DESIGN OVERVIEW

#### 5.1 CLASS DIAGRAM

A Class Diagram provides a blueprint for how different system components interact within MoneyGuard: Finance Management with Stock Prediction. It describes the data structure, functionalities, and relationships between key entities in the system.

##### [1] Detailed Class Descriptions

###### a) User Class

The User is the central entity interacting with both financial management and stock prediction features.

Attributes:

- userID: int → Unique identifier for each user.
- username: string → User's registered name.
- password: string → Stores encrypted user password.
- email: string → Contact email address.

Methods:

- + login() → Authenticates user credentials.
- + register() → Registers a new user account.
- + updateProfile() → Updates user details.
- + viewReports() → Generates a financial summary for the user.

Relationships:

- One User can have multiple Expenses (1:M).
- One User owns one or more Accounts (1:M).
- A User Requests Stock Predictions from the StockPrediction System.

###### b) Expense Class

This class tracks user expenses, categorizing transactions to help users manage their spending.

Attributes:

- expenseID: int → Unique identifier for each expense.
- amount: float → Amount spent.
- category: string → Expense category (e.g., food, travel, shopping).
- date: DateTime → Timestamp for the expense.

Methods:

- + addExpense() → Allows users to add an expense.
- + editExpense() → Enables modification of an existing expense.
- + viewExpense() → Retrieves and displays expense records.

Relationships:

- One User can have multiple Expenses (1:M).
- Expenses belong to an Account.

### c) Account Class

The Account class maintains user balances and financial records.

Attributes:

- accountID: int → Unique identifier for an account.
- balance: string → Current balance in the account.
- currency: string → Type of currency used (e.g., USD, INR).

Methods:

- + addTransaction() → Logs a transaction in the account.
- + viewBalance() → Retrieves and displays the account balance.
- + updateBalance() → Updates balance after an expense or deposit.

Relationships:

- A User Has One or More Accounts (1:M).
- Accounts Securely Store Expenses and Transactions.

### d) StockPrediction Class

The StockPrediction class manages real-time and historical stock predictions for users.

Attributes:

- predictionID: int → Unique identifier for a stock prediction.
- stockSymbol: string → Stock ticker symbol (e.g., AAPL, TSLA).
- predictionPrice: float → Forecasted stock value.
- predictionDate: DateTime → Date of prediction.

Methods:

- + generatePrediction() → Calls the ML model to predict stock prices.
- + viewPrediction() → Displays forecasted stock values.

Relationships:

- The User Requests Predictions (1:M).

- The StockPrediction System Generates Predictions using the MachineLearningSystem.

#### e) MachineLearningSystem Class

This class handles machine learning models used to predict stock prices.

Attributes:

- modelType: String → Defines the ML model used (e.g., LSTM).
- trainingData: DataFrame → Stores historical stock data for training.

Methods:

- + trainModel() → Trains the model with historical stock data.
- + predictStockPrices() → Generates stock price forecasts based on current market trends.

Relationships:

- Generates Stock Predictions for the StockPrediction System.

#### f) Security Class

The Security class ensures data encryption, authentication, and user validation.

Attributes:

- encryptionMethod: string → Defines encryption techniques (e.g., AES-256).
- authToken: string → Stores authentication tokens.

Methods:

- + encryptData() → Encrypts financial data for protection.
- + decryptData() → Decrypts stored information for authorized access.
- + generateAuthToken() → Creates secure login tokens.
- + validateUser() → Validates login credentials.

Relationships:

- Secures User Data for the Account and StockPrediction Systems.



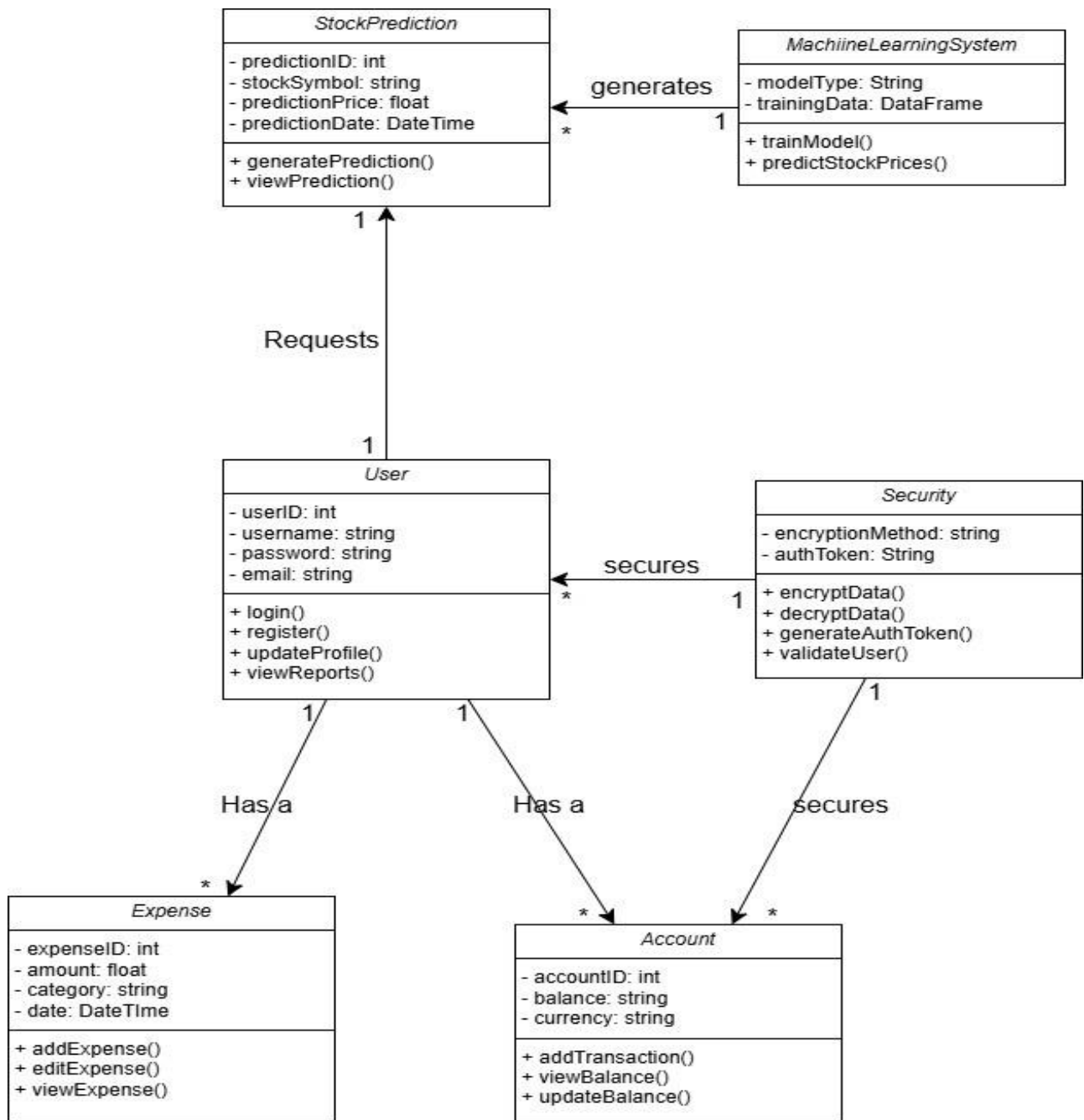


Fig 1.7 Class Diagram

## 5.2 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation that illustrates how data moves within the MoneyGuard system, highlighting data inputs, processes, storage, and outputs. The DFD helps in understanding the functional aspects of the system by breaking down its components and interactions at different levels.

### A. Levels of Data Flow Diagram

The MoneyGuard system's DFD is structured into multiple levels, from a high-level context diagram (Level 0) to a more detailed breakdown (Levels 1 and 2).

[1] Level 0 – Context Diagram

The Context Diagram (Level 0) represents MoneyGuard as a single entity that interacts with external actors (User and Admin). It defines the overall input and output data flows but does not provide details of internal processes.

Actors:

- User: Provides input data (registration, transactions, stock prediction requests) and receives system-generated reports and insights.
- Admin: Manages system security, data integrity, and financial analytics.

Major Data Flows:

- User provides Login Credentials, which are verified.
- User enters Transactions (Expenses, Income, Budgeting) to update financial records.
- User requests Stock Predictions, which are processed by the Machine Learning System.
- Admin manages the platform by Reviewing Data, Managing Users, and Updating System Policies.

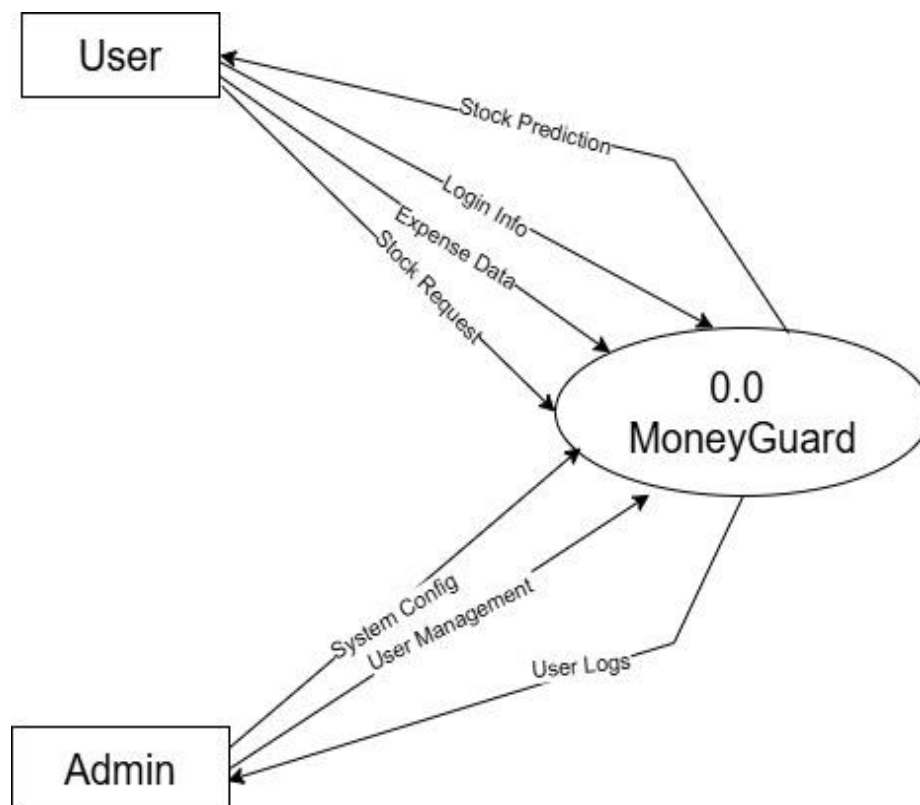


Fig 1.8 Data Flow Diagram [Level 0]

[2] Level 1 – Detailed System Process Breakdown

In Level 1, the major processes within the system are broken down into multiple sub-processes, showing how different functionalities interact with data storage and external actors.

Key Processes:

- User Authentication: Verifies login credentials using the database.
- Expense and Budget Management: Users add/view expenses and budgets, and the system updates financial records.
- Stock Prediction Module: Fetches stock market data and uses the ML model to generate predictions.
- Financial Report Generation: Summarizes expenses, savings, and stock trends for user insights.
- Data Security and Encryption: Ensures secure storage and transmission of financial data.

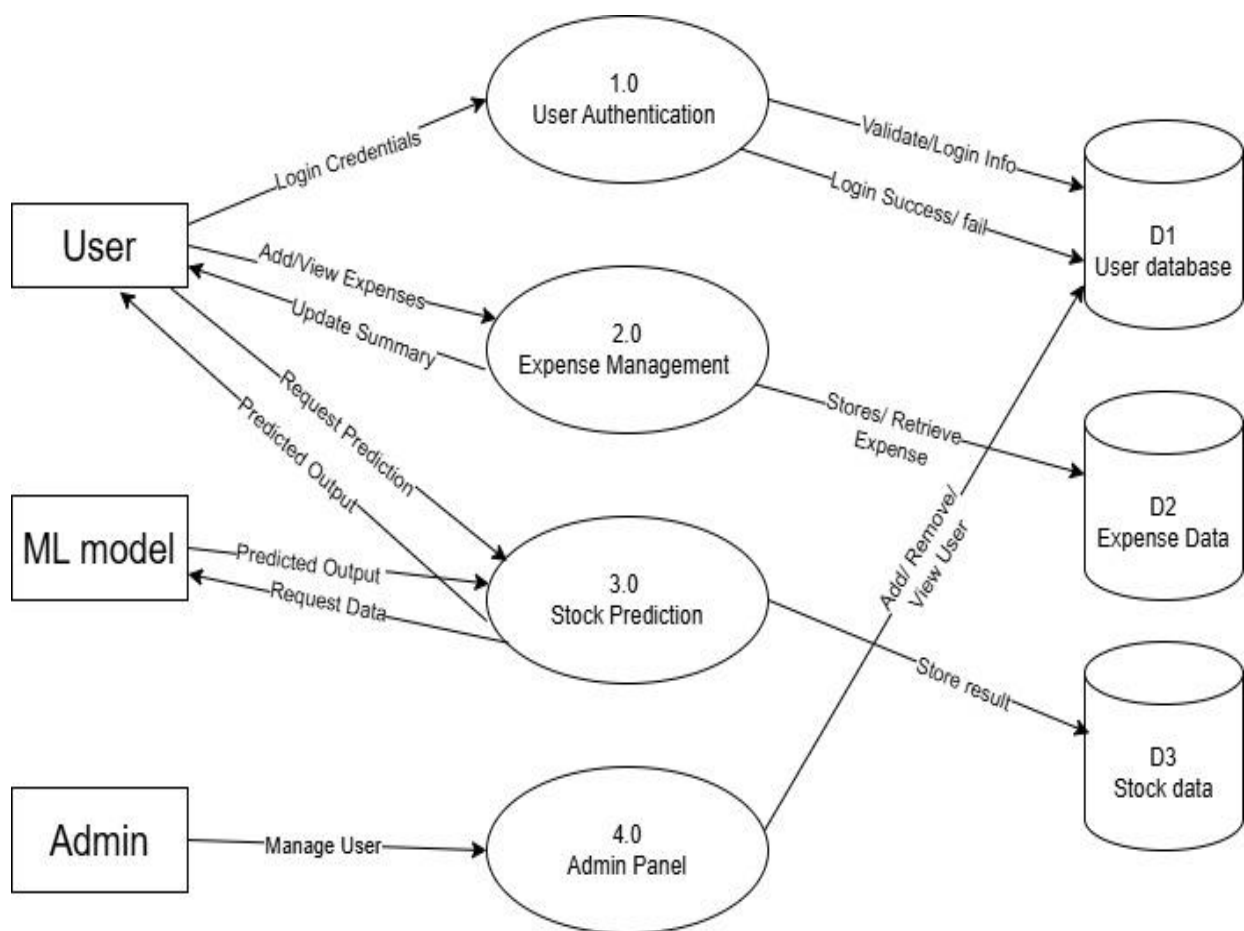


Fig 1.9 Data Flow Diagram [Level 1]

[3] Level 2 – Detailed Data Flow

Level 2 provides deeper insights into critical operations, such as Expense Management and Stock Prediction, by breaking them down into specific steps.

Example: Stock Prediction Process

- User requests a stock prediction based on a selected stock.

- System retrieves stock data from an external API or database.
- ML Model analyzes data and generates predictions.
- System displays forecasted trends to the user.

Example: Expense Tracking Process

- User enters new expense details (category, amount, date).
- System stores the record in the database.
- User can view/edit/delete past transactions.
- System generates reports and insights based on spending trends.

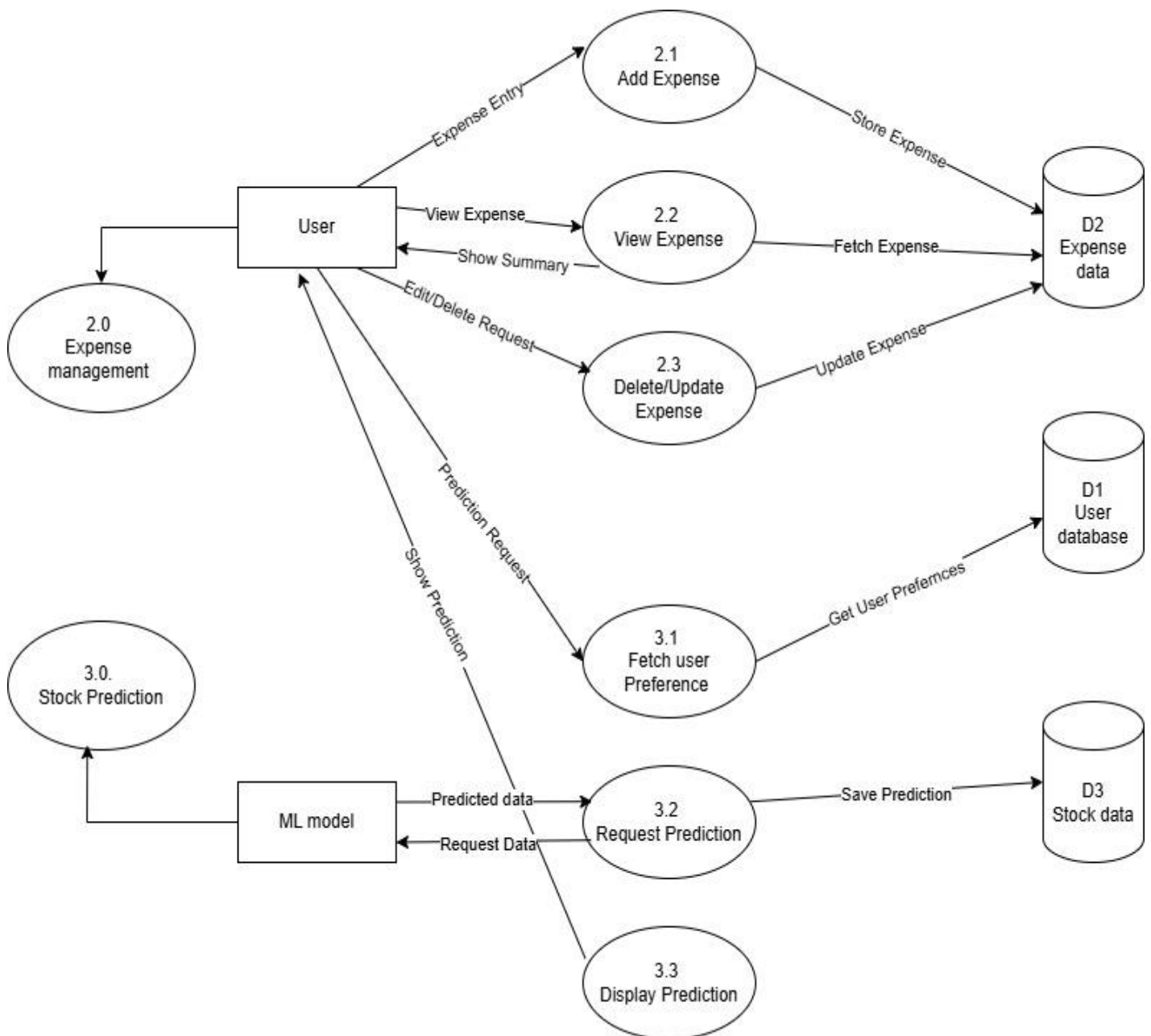


Fig 2.0 Data Flow Diagram [Level 2]

### 5.3 DATA DICTIONARY

A Data Dictionary is a structured collection of information about the data used in a system. It defines each data element, including its type, format, and relationships with other data. In the MoneyGuard project, the data dictionary ensures data consistency, accuracy, and integrity within financial management and stock prediction processes.

#### [1] Purpose of the Data Dictionary

The primary purpose of the Data Dictionary in MoneyGuard is to:

- Define and document the data structures used in finance management and stock prediction.
- Ensure data integrity and accuracy by standardizing data formats.
- Improve database management by providing clear definitions of attributes and relationships.
- Facilitate system development and debugging by providing a reference for database design.

#### [2] Key Tables in MoneyGuard System

The MoneyGuard system consists of multiple tables storing user information, financial transactions, stock predictions, and security-related data. Below is an overview of key tables and their attributes.

##### a) User Module

Field Name	Data Type	Length	Description
user_id	Integer	10	Unique identifier for the user
full_name	Varchar	100	Full name of the user
email	Varchar	100	Email ID used for login
password	Varchar	255	Encrypted password
created_at	Timestamp	-	Account creation timestamp
auth_token	Varchar	255	Encrypted token used for sessions

Table 1.2 Data Dictionary for user module

##### b) Expense Management

Field Name	Data Type	Length	Description
expense_id	Integer	10	Unique identifier for each expense
user_id	Integer	10	Foreign key linking to the user
category	Varchar	50	Category of the expense

amount	Decimal	10,2	Expense amount
description	Text	-	Description of the transaction
date	Date	-	Date of the expense

Table 1.3 Data Dictionary for Expense Management

## c) Budget and financial goals

Field Name	Data Type	Length	Description
goal_id	Integer	10	Unique identifier for the goal
user_id	Integer	10	Foreign key to identify user
goal_type	Varchar	50	Type of goal (e.g., Savings, Investment)
current_amount	Decimal	10,2	Amount saved so far
deadline	Date	-	Deadline to achieve the goal

Table 1.4 Data Dictionary for Budget and Financial goals

## d) Stock prediction module

Field Name	Data Type	Length	Description
prediction_id	Integer	10	Unique ID for stock prediction instance
user_id	Integer	10	Foreign key to identify user
stock_symbol	Varchar	10	Symbol of the stock
prediction_date	Date	-	Date for which prediction is made
predicted_price	Decimal	10,2	Predicted stock price
model_accuracy	Decimal	5,2	Accuracy percentage of the prediction

Table 1.5 Data Dictionary for Stock Prediction module

## e) Security and session

Field Name	Data Type	Length	Description
session_id	Varchar	255	Unique session ID for user login
user_id	Integer	10	Foreign key to identify user
login_time	Timestamp	-	Login timestamp
logout_time	Timestamp	-	Logout timestamp
ip_address	Varchar	45	IP address of the user session

Table 1.6 Data Dictionary for Security and Session

## f) Machine Learning model logs

Field Name	Data Type	Length	Description
log_id	Integer	10	Unique ID for each log
prediction_id	Integer	10	Reference to predicted stock
timestamp	Timestamp	-	Time when prediction generated
model_version	Varchar	20	Version of the ML model used
input_data_snapshot	Text	-	Serialized snapshot of input data

Table 1.7 Data Dictionary for Machine Learning model bugs

## 5.4 EXTENDED ER DIAGRAM

An Extended Entity-Relationship (EER) Diagram expands on the traditional ER model by incorporating advanced concepts such as inheritance, specialization, generalization, aggregation, and complex relationships. The MoneyGuard system's EER diagram captures the interactions between users, financial transactions, stock predictions, and security components in a structured and scalable format.

### A. Entities & Relationships description

#### [1] User Entity & Its Relationships:

- A User can create and manage multiple Accounts.
- A User can add Transactions (Income or Expense), which are linked to their account.
- A User sets Financial Goals, which guide their budgeting and investment strategies.
- A User can request Stock Predictions, which are generated based on their investment preferences.
- A User interacts with the Security System, ensuring data protection through authentication and encryption.

#### [2] Account & Financial Transactions:

- Each Account stores financial data related to income, expenses, and budgets.
- An Account can have multiple Transactions, categorized as Income or Expense.
- Transactions are classified into different categories (Food, Rent, Salary, Investments, etc.).
- Budgets are associated with an Account, helping users track their expenses.

#### [3] Machine Learning Model & Stock Prediction:

- The Machine Learning Model processes historical stock data and user preferences to generate personalized stock predictions.

- The Stock Prediction entity stores forecasted stock prices, confidence scores, and analysis reports.
- Users receive Stock Predictions based on their investment preferences and stock selection.

[4] Security & Data Protection:

- The Security System handles authentication (Login, Registration, Encryption).
- User credentials are securely stored with hashing and encryption techniques.
- Security Logs track user activity, login attempts, and potential security threats.

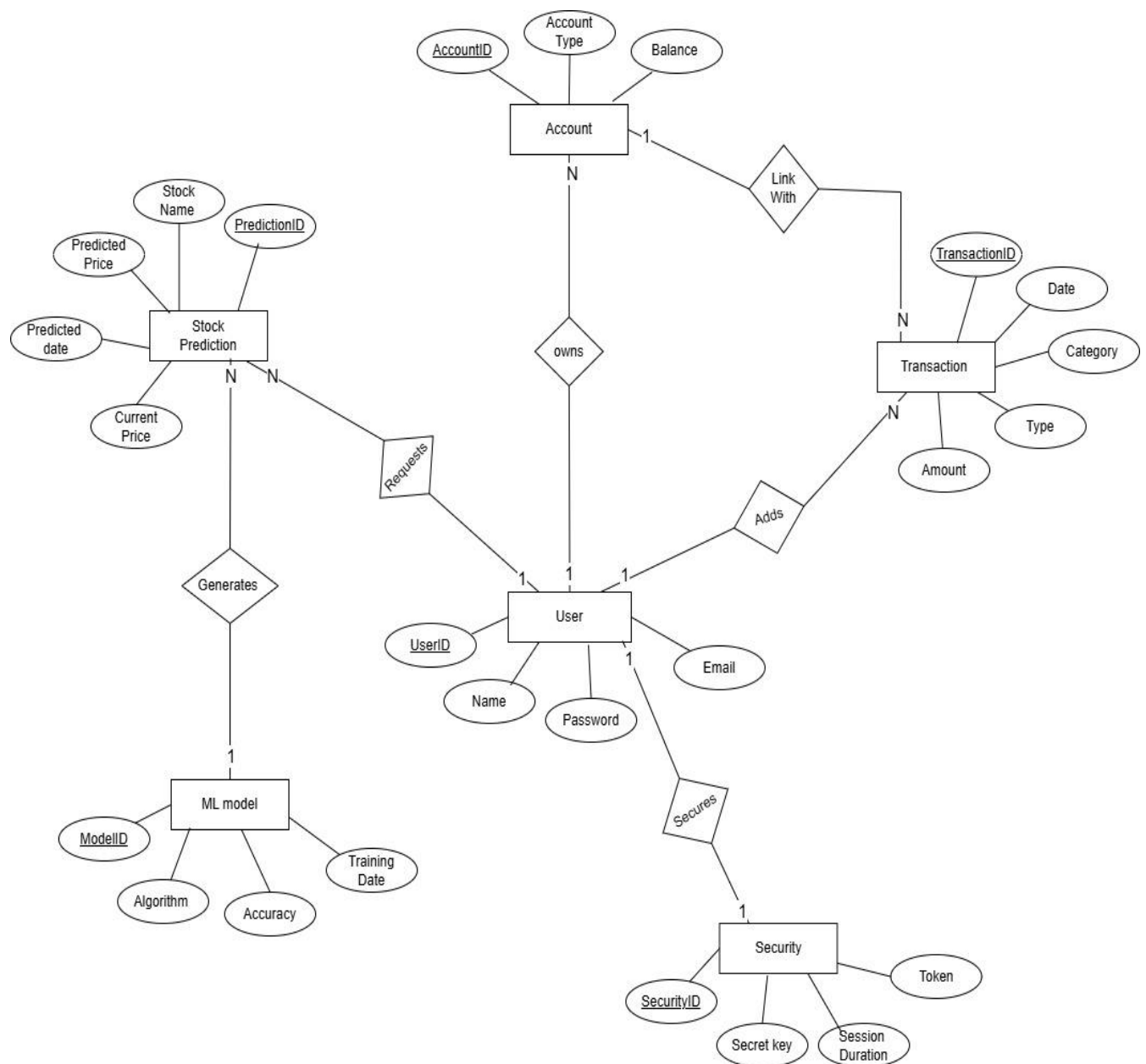


Fig 2.1 Extended ER Diagram



## CHAPTER 6

### WORKPLAN AND SYSTEM DATABASE STRUCTURE

#### 6.1 TIME FRAME WORK

Developing our MoneyGuard involves multiple stages, from requirement gathering to deployment. The estimated timeline is outlined below, ensuring an iterative and agile approach for efficient implementation.

##### **Week 1: Project Planning and Requirement Analysis**

Objective: Establish a comprehensive understanding of the project scope, functionalities, and technical requirements.

Tasks:

##### 1. Project Scope & Definition:

- Clearly define the goals, features, and challenges of MoneyGuard.
- Identify key users (individual investors, finance enthusiasts, general users).

##### 2. Requirement Gathering:

- Research existing finance management and stock prediction platforms.
- Finalize core features, including expense tracking, stock analysis, and security measures.
- Identify platform requirements (web-based, with potential mobile integration).

##### 3. Technology Stack Selection:

- Finalize frontend (HTML, CSS, JavaScript, Bootstrap), backend (Flask), and database (SQLite). And choose LSTM for stock price prediction and API sources (Yahoo Finance).
- Establish security standards (AES-256 encryption, HTTPS, user authentication).

##### **Week 2: System Design & Architecture Planning**

Objective: Design the system's database, frontend layout, backend structure, and API flow.

Tasks:

##### 1. System Architecture Design:

- Develop a high-level system architecture diagram.
- Define data flow (user registration → dashboard → transaction entry → stock prediction → insights).

##### 2. Database Design:

- Design the database schema for storing:
  - ✓ User profiles (ID, name, email, encrypted password)

- ✓ Transaction records (date, amount, category, payment method)
- ✓ Stock prediction logs (ticker, historical data, predicted price)

3. UI/UX Wireframing:

- Design wireframes for key pages:
  - ✓ Login/Signup, Dashboard, Expense Entry, Stock Prediction, Reports & Insights.

### **Week 3: Backend & Security Implementation**

Objective: Develop the backend APIs and implement security protocols.

Tasks:

1. Flask API Development:

- Implement APIs for user authentication, transaction management, and stock data retrieval.
- Set up database interaction with SQLite and SQLAlchemy ORM.

2. Security & Encryption Implementation:

- Encrypt sensitive user data using AES-256.
- Implement JWT-based authentication for secure login sessions.

### **Week 4: Frontend Development & Integration**

Objective: Develop a responsive and interactive frontend interface.

Tasks:

1. Web Development (HTML, CSS, JavaScript, Bootstrap):

- Create Dashboard UI for expense tracking, stock prediction, and analytics visualization.
- Develop forms for adding transactions and budget tracking.

2. Stock Prediction Integration:

- Connect Yahoo Finance API to fetch stock data.
- Implement LSTM machine learning model for price prediction.

3. Visualization & Graphs:

- Integrate Chart.js or D3.js to display expense trends and stock performance.

### **Week 5: Machine Learning Model Training & Optimization**

Objective: Train, test, and fine-tune the LSTM model for stock price prediction.

Tasks:

1. Dataset Collection:

- Gather historical stock data from Yahoo Finance.

2. Model Development:

- Train LSTM model using TensorFlow/Keras for next-day stock prediction.

- Fine-tune hyperparameters (epochs, batch size, learning rate) for accuracy.
3. Model Testing & Evaluation:
- Validate predictions against real stock market trends by optimizing the model to reduce prediction errors.

### **Week 6: Feature Enhancements & Bug Fixing**

Objective: Refine features based on internal testing.

Tasks:

1. User Experience (UX) Improvements:
  - Optimize page load speed and UI responsiveness.
2. Error Fixes & Debugging:
  - Fix any backend API issues, database inconsistencies, or stock prediction inaccuracies.

### **Week 7: Testing & Quality Assurance**

Objective: Conduct rigorous testing before deployment.

Tasks:

1. Unit Testing (Backend & Frontend):
  - Test individual Flask APIs, database interactions, and stock prediction model.
2. Integration Testing:
  - Test frontend-backend communication for smoothness of website.
3. Security Testing:
  - Ensure encryption, authentication, and user data protection meet industry standards.
4. Performance Testing:
  - Verify that the system handles concurrent users efficiently.

### **Week 8: Deployment & Finalization**

Objective: Deploy the MoneyGuard platform for public access.

Tasks:

1. Cloud Deployment:
  - Host the backend on Render.
  - Deploy frontend to GitHub Pages.
2. Database Optimization & Backup Setup:
  - Enable automated backups and database security patches.
3. Final Testing & Feedback Collection:
  - Implement minor UI fixes, bug patches, and documentation updates.

## 6.2 DESIGN DISPLAY TABLE

The Design Display Table provides a structured overview of the key components of the MoneyGuard platform, outlining its modules, functionality, inputs, outputs, and technologies used. This structured representation helps in understanding how each component interacts within the system.

Module Name	Description	Input	Output	Tech Used
User Authentication	Handles user registration, login, and security	Username, Email, Password	Secure user login, JWT token for session management	Flask, SQLite, AES-256 Encryption, HTTPS
Dashboard	Displays overall financial summary, stock updates, and insights	User ID (retrieved from login session)	Personalized financial overview with visual analytics	HTML, CSS, JavaScript, Bootstrap, Chart.js
Expense Tracking & Budgeting	Enables users to log expenses and track spending	Expense details (amount, category, date)	Transaction added, updated financial summary	Flask, SQLite, JavaScript
Stock Prediction System	Uses LSTM model to predict future stock prices	Ticker symbol, historical stock data.	Predicted stock price for the next day	Python, TensorFlow, Flask API, Yahoo Finance API
Financial Insights & Visualization	Generates graphs and reports for financial trends	User transaction data, stock data	Graphs for spending patterns, investment returns	Chart.js, D3.js, JavaScript
Security & Data Encryption	Ensures secure storage and transmission of data	User credentials, financial records	Encrypted data stored securely	AES-256, HTTPS, JWT Authentication

Table 1.8 Data display Table

CHAPTER 7

IMPLEMENTATION AND TESTING

7.1 TESTING STRATEGY ADAPTED

The testing strategy for MoneyGuard ensures that the platform functions correctly, providing secure finance management and accurate stock predictions. Testing is conducted at different levels to verify the system's functionality, security, and performance while ensuring a seamless user experience.

[1] Objectives of Testing

- Ensure that the expense tracking, stock prediction, and transaction management work as expected.
- Verify data security measures, including encryption and authentication.
- Test the machine learning model for stock predictions to ensure accuracy.
- Identify and resolve bugs, performance bottlenecks, and security vulnerabilities.
- Ensure cross-platform compatibility (desktop, mobile).

[2] Testing Types Used

Testing Type	Purpose	Approach	Tools used
Unit Testing	Validate core functions like transactions and predictions	Test Key modules for accuracy	PyTest, Unittest
Integration Testing	Check interaction between backend, database, and ML model	Ensure seamless data flow	Postman, SQLite Testing
Functional Testing	Verify features like tracking, insights, and predictions	Test user flows and feature outputs	Selenium, manual Testing
Performance Testing	Assess response time and load handling	Evaluate speed and speed efficiency	JMeter, Locust
Security Testing	Ensure data encryption and safe access	Test vulnerabilities and authentication	OWASP ZAP, SQLMap
Usability Testing	Improve UI experience and ease of use	Collect insights for better navigation	User Feedback, Manual Testing

Table 1.9 Types of Testing in Moneyguard

## 7.2 SYSTEM TESTING

System testing ensures that MoneyGuard operates smoothly under different conditions. It evaluates the platform as a whole, verifying that all components work together correctly.

### [1] System Testing Approach

- **Requirement Validation:** Ensure that all functional and non-functional requirements are implemented correctly.
- **End-to-End Testing:** Test user workflows (e.g., logging in, adding transactions, checking stock predictions).
- **Error Handling:** Check how the system responds to invalid inputs, failed transactions, and security breaches.
- **Load and Stress Testing:** Simulate **high traffic conditions** to assess system stability and scalability.
- **Regression Testing:** Ensure new updates or bug fixes do not disrupt existing functionalities.

## 7.3 TEST CASES

Test cases define specific scenarios to validate system performance. Below are some key test cases for MoneyGuard:

ID	Test Scenario	Expected Output	Status
TC-01	User registration with valid credentials	Account successfully created	Pass/Fail
TC-02	User login with incorrect password	Error message displayed	Pass/Fail
TC-03	Add a financial transaction	Transaction stored in database	Pass/Fail
TC-04	Generate financial insights and reports	Displays accurate financial charts	Pass/Fail
TC-05	Predict stock price using ML model	Displays next-day stock price prediction	Pass/Fail
TC-06	Attempt SQL injection attack	System prevents unauthorized access	Pass/Fail
TC-07	Load test with 1000+ users	System maintain stable performance	Pass/Fail
TC-08	Check security of encrypted transactions	Data remains protected	Pass/Fail

Table 2.0 Test Cases involved in MoneyGuard

## 7.4 DEPLOYMENT STRATEGY

The deployment phase is a crucial part of any software development lifecycle, ensuring that the project is accessible, scalable, and usable in a real-world environment. For the *MoneyGuard* project, the deployment was carried out using Render, a cloud-based platform that simplifies web service hosting for modern applications.

The aim was to make the platform accessible online with full functionality, including secure user login, financial transaction tracking, and live stock prediction using machine learning models.

### [1] Hosting Platform: Render

Render was chosen due to its ease of deployment for full-stack web applications, support for Python/Flask, and automatic deployment pipelines from GitHub. It also provides SSL (HTTPS) by default, enhancing the security of financial data handled in the MoneyGuard application.

### [2] Steps Followed in Deployment

The deployment process followed these major steps:

1. Codebase Preparation:
  - The complete Flask application was organized into proper folders (static, templates, models, etc.).
  - All required packages were listed in a requirements.txt file.
  - A Procfile was created to instruct Render on how to start the Flask app.
2. Database Handling:
  - SQLite was used for lightweight local storage.
  - On deployment, the database was initialized on Render using a startup shell command.
3. Connecting to GitHub Repository:
  - The project was pushed to GitHub. Render was connected to the GitHub repository for automatic deployments on push.
4. Environment Configuration:
  - Environment variables such as secret keys and API tokens were set securely in Render's dashboard. SSL (HTTPS) was enabled by default.
5. Model Integration:
  - The trained LSTM model for stock prediction was uploaded along with the project.
  - On Render, prediction requests are processed using live data fetched via APIs (e.g., Yahoo Finance), which is parsed and fed into the model.
6. Testing Live Application:
  - After deployment, the application was tested in real-time by multiple users.

## CHAPTER 8

### CONCLUSION AND FUTURE EXTENSION

#### 8.1 CONCLUSION

The MoneyGuard project successfully addresses the increasing need for an integrated finance management and stock prediction platform that combines personalized budgeting, expense tracking, and AI-powered stock forecasts in a secure and user-friendly environment. By integrating these features, MoneyGuard empowers users to make smarter financial decisions while ensuring data security through robust encryption and authentication mechanisms.

The platform not only enhances financial planning but also promotes financial literacy by providing real-time insights, investment guidance, and trend analysis. Unlike traditional finance management tools, MoneyGuard leverages machine learning (LSTM) to predict stock trends, enabling users to make informed investment choices based on data-driven forecasts.

The implementation of Flask-based backend, secure SQLite database storage, and interactive UI with HTML, CSS, and JavaScript ensures that the platform is scalable, accessible, and easy to use. Furthermore, the integration of HTTPS security, AES-256 encryption, and multi-factor authentication makes MoneyGuard a trustworthy solution for managing sensitive financial data.

Through comprehensive testing across functional, security, and performance domains, the system has been optimized to deliver high accuracy, efficiency, and seamless user experience. The successful execution of this project demonstrates the potential of fintech innovations in reshaping personal finance management and investment strategies. MoneyGuard serves as a powerful tool for users seeking financial stability, security, and growth in an increasingly digital world.

#### 8.2 FUTURE SCOPE

The MoneyGuard platform has significant potential for future enhancements, allowing it to evolve into a more advanced and user-centric financial management tool. Several improvements and expansions can be incorporated to increase its usability, accuracy, and global reach:

1. **Advanced AI-Based Stock Prediction** – Future iterations can integrate more sophisticated deep learning models to enhance the accuracy of stock price predictions by analyzing a broader range of financial indicators, news sentiment, and real-time market trends.
2. **Personalized Financial Insights & Recommendations** – Implementing AI-driven personalized budget planning, saving suggestions, and automated alerts based on user spending patterns will provide intelligent financial assistance tailored to individual needs.



3. **Multi-Platform Expansion & Mobile App Development** – Extending MoneyGuard to a dedicated mobile application (Android & iOS) would make it more accessible and convenient, enabling real-time finance tracking and stock predictions on the go.
4. **Integration with Financial APIs & Banking Services** – Incorporating APIs for bank account linking, real-time transaction synchronization, and credit score monitoring would create a seamless financial management ecosystem within the platform.
5. **Enhanced Security & Blockchain Integration** – Strengthening data privacy with blockchain-based transaction verification and decentralized finance (DeFi) integration can ensure a tamper-proof, secure financial management experience.
6. **Multi-Currency & Global Stock Market Support** – Supporting multiple currencies and integrating global stock market analysis will expand the platform's reach to international users and investors.
7. **Voice & Chatbot-Based Financial Assistance** – Implementing an AI-powered chatbot or voice assistant can help users interact with the platform more intuitively, answering financial queries and offering investment suggestions.

By incorporating these advancements, MoneyGuard has the potential to revolutionize personal finance management, making it an intelligent, data-driven financial companion that enhances both budgeting efficiency and investment decision-making.

## REFERENCES

Here are some reference which we have used in our project MoneyGuard : Finance Management with Stock Prediction which are as follows :-

### Books and Tutorials

- [1] Y. Hilpisch, *Python for Finance*.
- [2] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*.

### APIs and Libraries Documentation

- [3] *Yahoo Finance API Documentation*. Available: <https://pypi.org/project/yfinance/>
- [4] *Matplotlib Documentation*. Available: <https://matplotlib.org/stable/users/index.html>
- [5] *SQLite Documentation*. Available: <https://www.sqlite.org/docs.html>

### Research Papers

- [6] "AI in Stock Market Predictions".
- [7] "The Role of Data Visualization in Financial Decision-Making".
- [8] "Python-Based Financial Management Systems: Opportunities and Challenges".

### Websites for Real-Time Information

- [9] *Yahoo Finance*. Available: <https://finance.yahoo.com/>

### Github links for understanding

- [10] <https://github.com/Shridhar-T-P/python-flask--personal-expense-tracker>
- [11] <https://github.com/shreyai347/Stock-Price-Prediction>
- [12] <https://github.com/Joshwen7947/User-Authentication-with-Flask>

### Websites for guidance

- [13] <https://www.thenerdnook.io/p/secure-authentication-for-flask-web>

# APPENDIX A

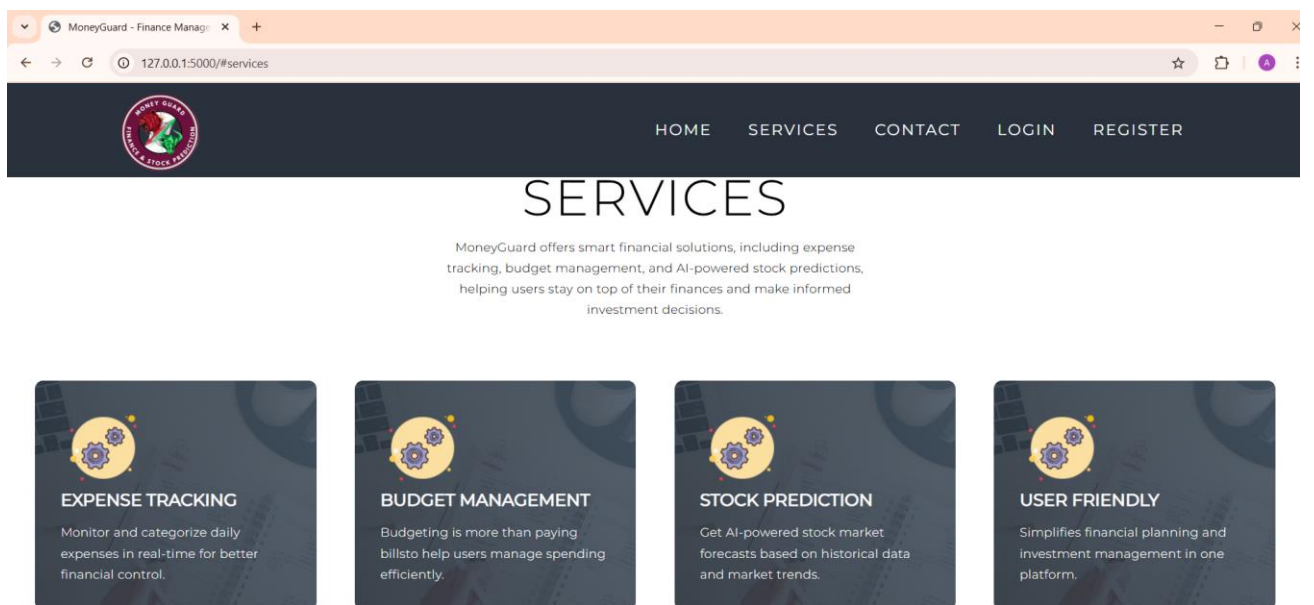
## ( Screenshots )

Here are some UI/UX of website MoneyGuard

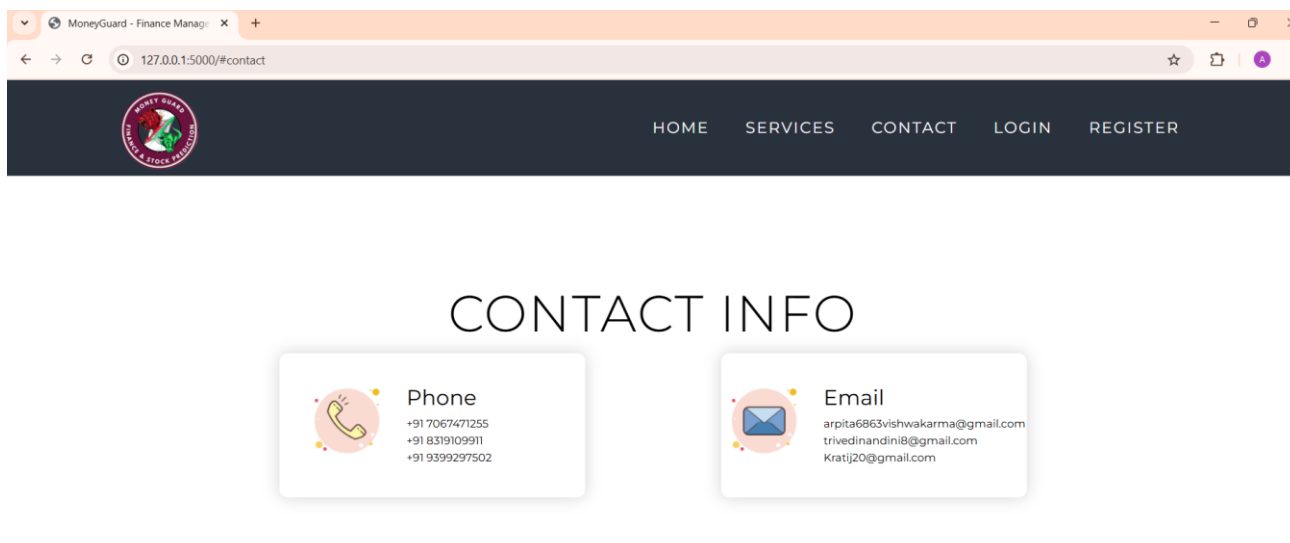
### 1. Landing page of MoneyGuard (Home page)



### 2. Services section of MoneyGuard (on clicking services on home page)



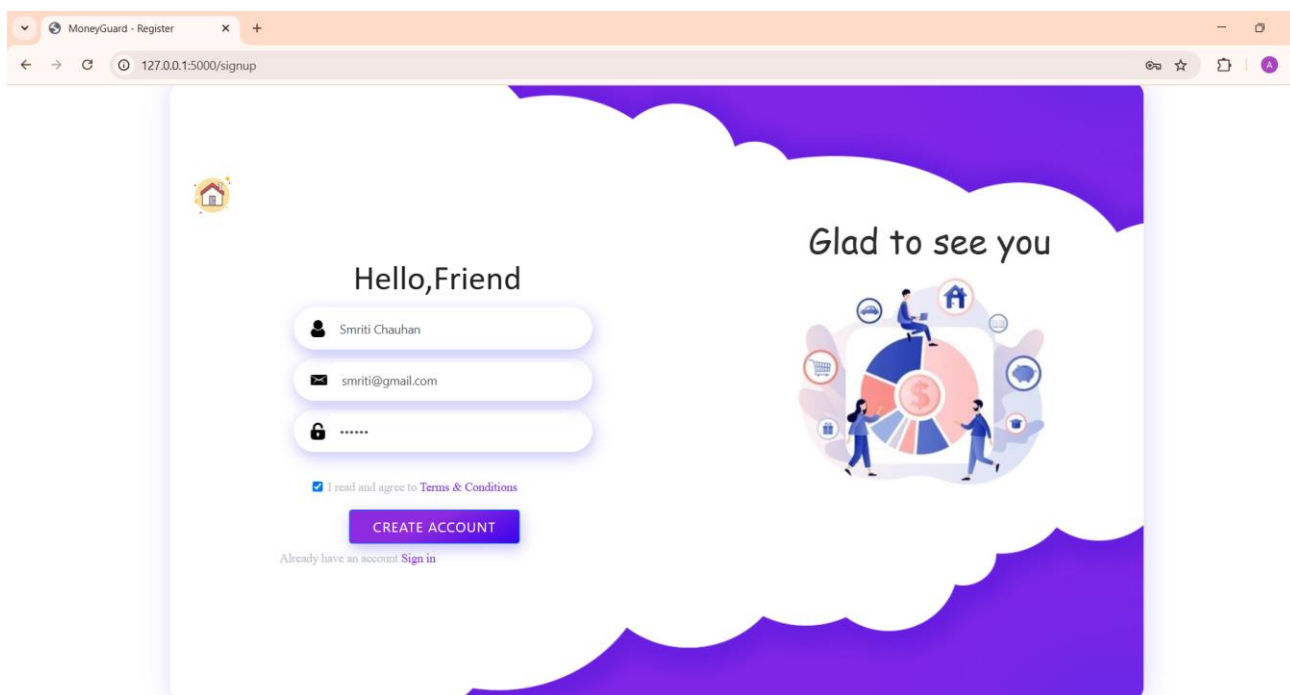
### 3. Contact section of MoneyGuard ( on clicking contact on home page )



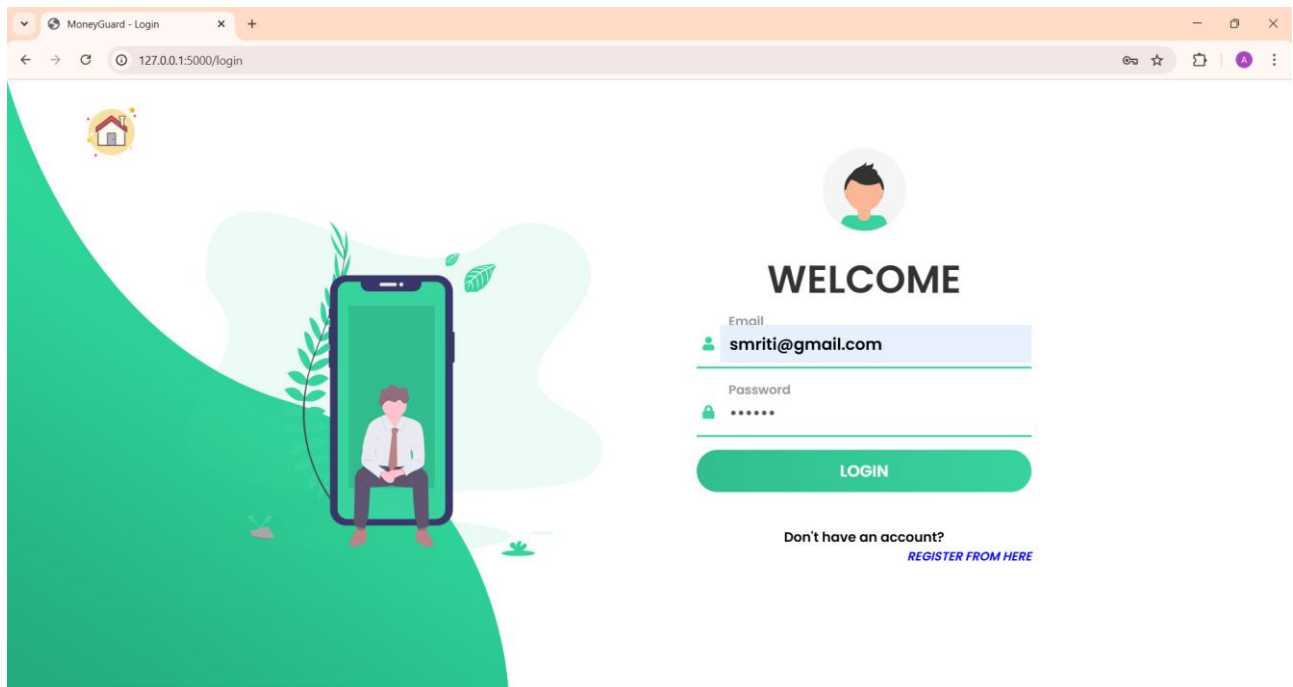
### 4. Footer section of MoneyGuard



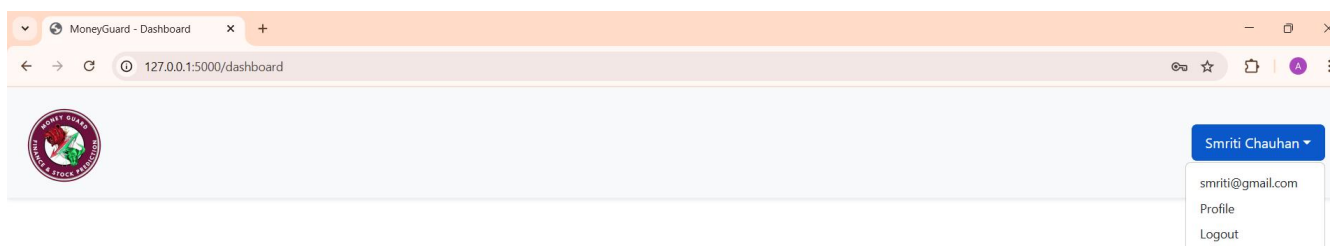
### 5. Register page of MoneyGuard (before begin with the Moneyguard, you need to register first then only you can login with email and password)



### 6. Login Page of MoneyGuard (after registering only one can login)



### 7. The dashboard of the website after login in the website

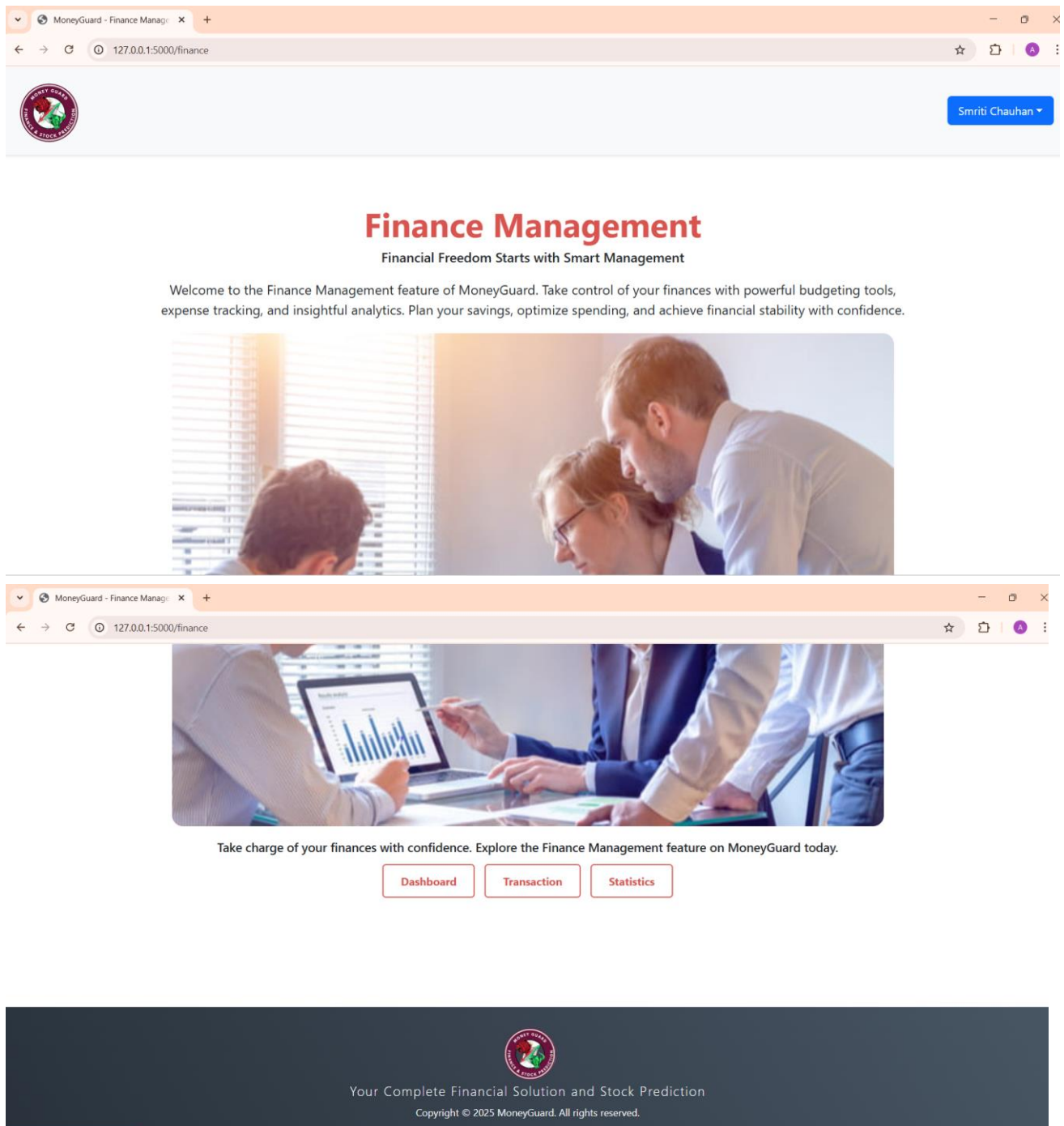


#### LET'S START THE JOURNEY

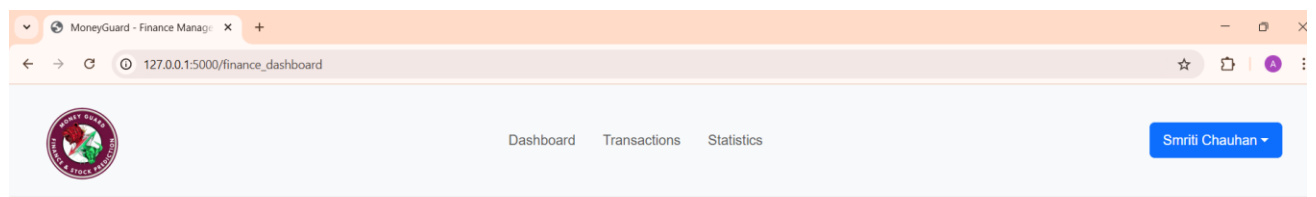
MoneyGuard is your personal finance companion, helping you track expenses, manage budgets, and make smarter investment decisions. With a powerful stock prediction feature, we bring the future of finance to your fingertips. Our platform offers real-time insights, secure data handling, and a user-friendly interface designed to simplify financial management. Whether you're a beginner looking to save or an investor seeking growth, MoneyGuard has something for you. Start your journey toward financial stability today. Explore expense tracking, budgeting, and intelligent stock market predictions—all in one place.

[Finance Management](#)[Stock Prediction](#)

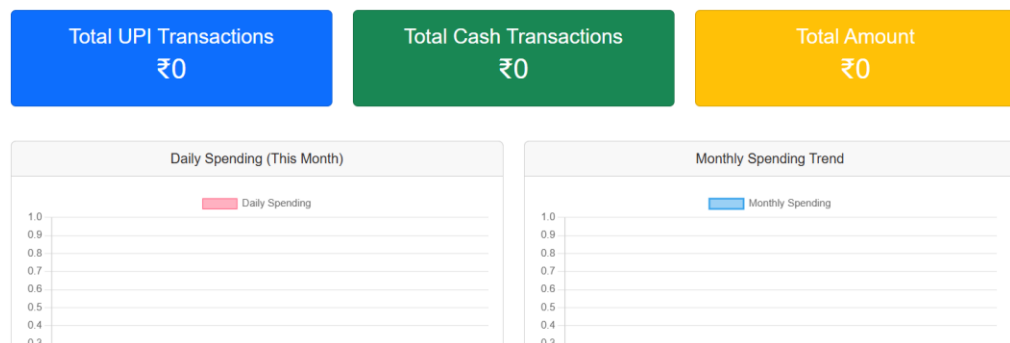
8. Only clicking the “finance management” on the previous page we get this interface.



9. After clicking the Dashboard button , we are entered in the finance management part of MoneyGuard.

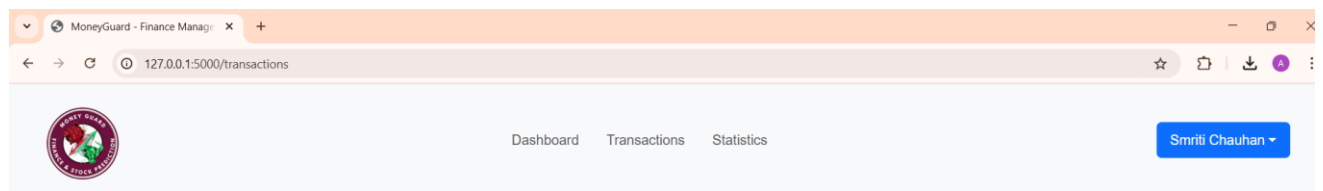


**Welcome, Smriti Chauhan !**



10. On clicking the transaction button at the top center, we are navigated on the transaction page of finance management.

As smriti is the new user so she had not yet added any transaction in her account



11. On clicking Add transaction we are navigated on the popup window to add transaction.

The screenshot shows the 'Add Transaction' modal form in the MoneyGuard application. The form is open over the 'Transactions' page. The modal contains the following fields and buttons:

- Date: dd-mm-yyyy
- Category: Entertainment
- Amount:
- Payment Method: UPI
- Notes:
- Add Transaction button
- CSV button

After filling the pop up form for adding the transaction, the information is added on the table.

The screenshot shows the 'Add Transaction' modal form filled out and the transaction added to the table. The modal shows the following information:

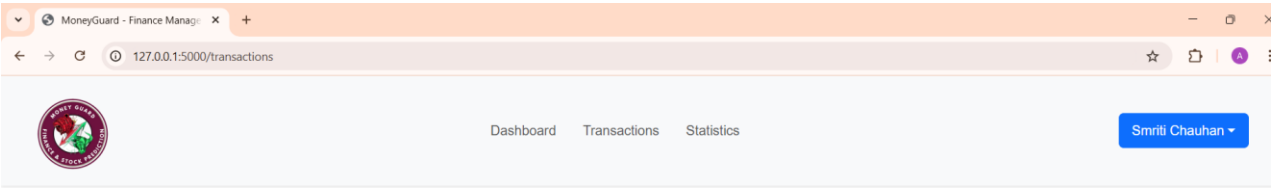
- Date: 12-03-2025
- Category: Education
- Amount: 5000
- Payment Method: UPI
- Notes:
- Add Transaction button
- CSV button

The table below shows the transaction added:

Date	Category	Method	Notes	Action
2024-04-12	Food			
2024-06-12	Travel expenses			
2024-12-16	Subscriptions			
2025-01-12	Utilities			
2025-02-12	Entertainment			

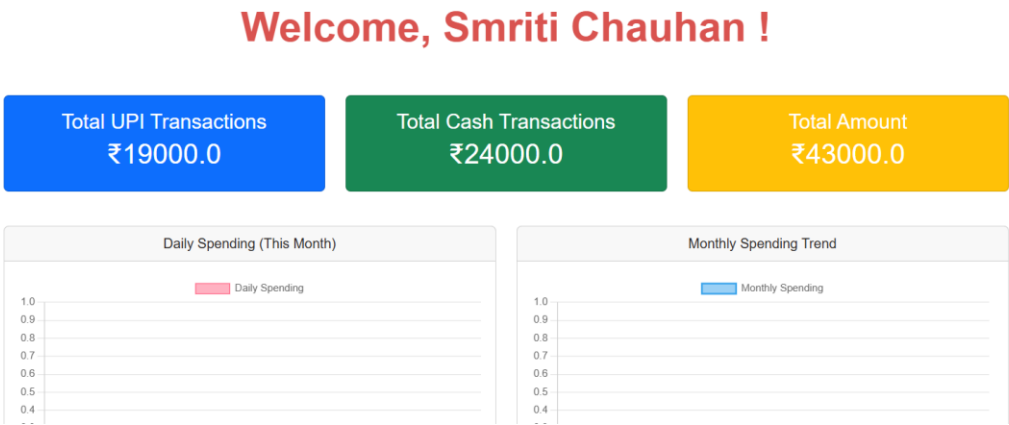
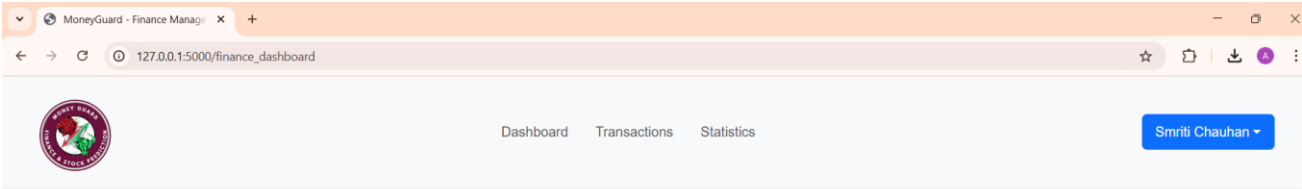
Here all the transaction detail added on the table



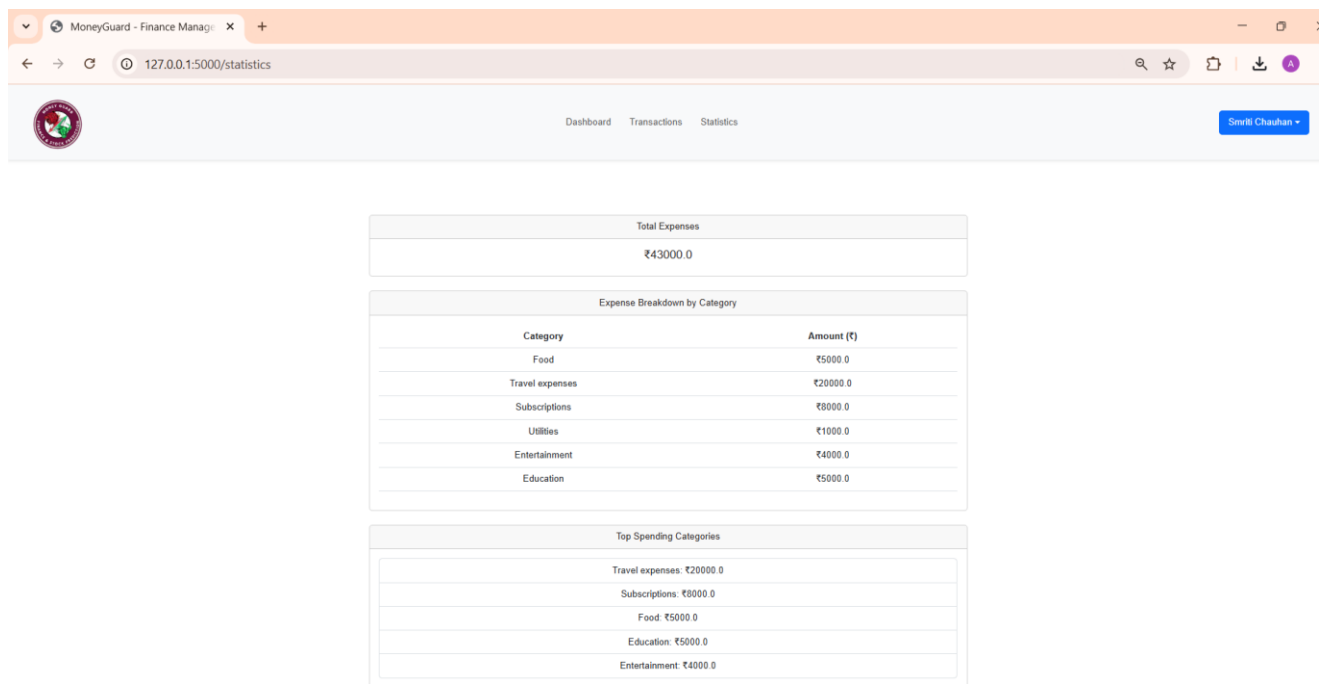


Transactions					
Date	Category	Amount	Payment Method	Notes	Action
2024-04-12	Food	₹5000.0	UPI		
2024-06-12	Travel expenses	₹20000.0	Cash		
2024-12-16	Subscriptions	₹8000.0	UPI		
2025-01-12	Utilities	₹1000.0	UPI		
2025-02-12	Entertainment	₹4000.0	Cash		
2025-03-12	Education	₹5000.0	UPI		

12. All the total transaction amount are calculated and printed on the dashboard and statistics of the data on the statistic page of finance management.



## MoneyGuard : Finance Management with Stock Prediction



Total Expenses	
₹43000.0	

Expense Breakdown by Category	
Category	Amount (₹)
Food	₹5000.0
Travel expenses	₹20000.0
Subscriptions	₹8000.0
Utilities	₹1000.0
Entertainment	₹4000.0
Education	₹5000.0

Top Spending Categories	
Travel expenses: ₹20000.0	
Subscriptions: ₹8000.0	
Food: ₹5000.0	
Education: ₹5000.0	
Entertainment: ₹4000.0	

13. Now navigating to the second part of moneyguard i.e. stock prediction.

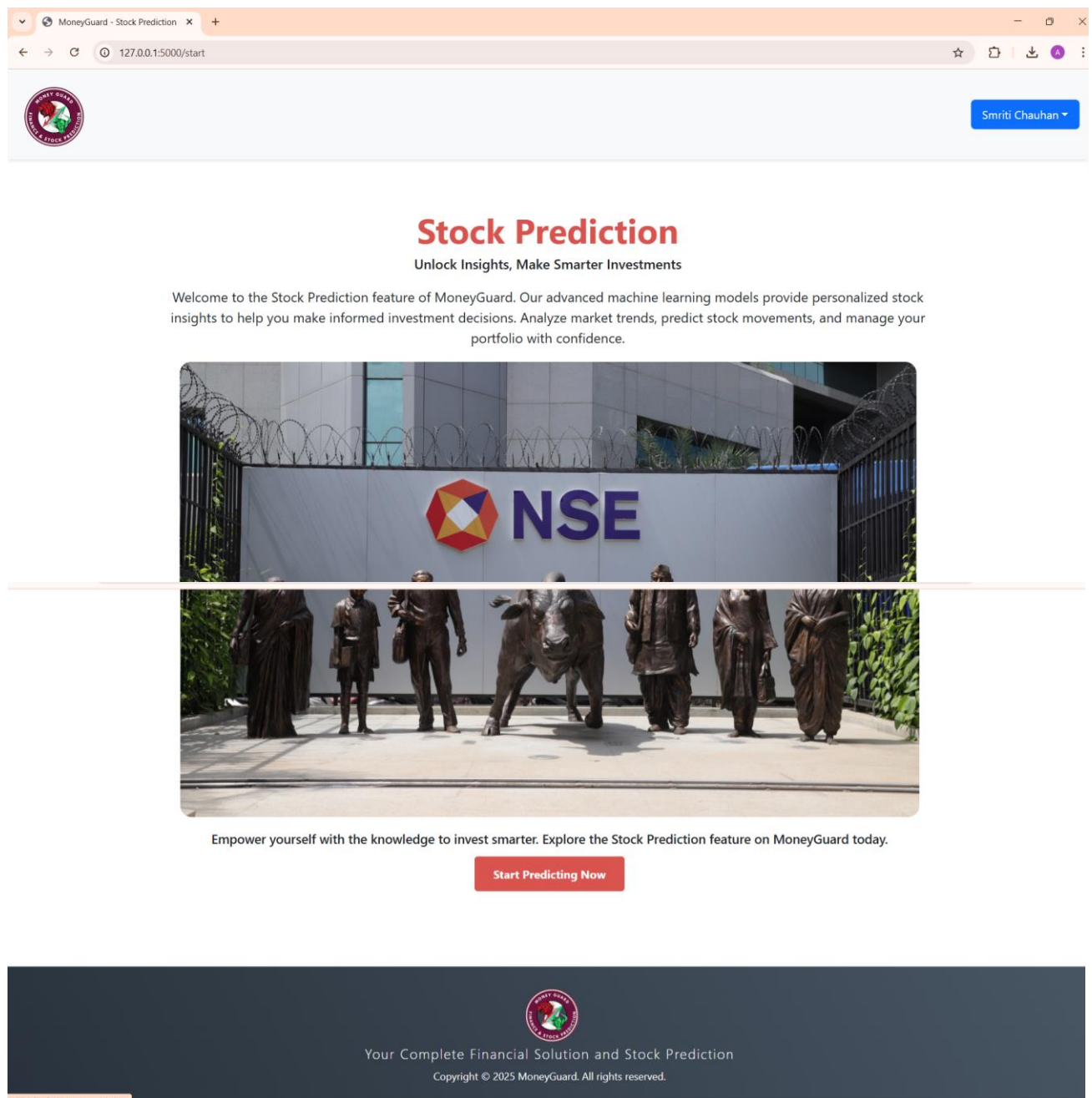


### LET'S START THE JOURNEY

MoneyGuard is your personal finance companion, helping you track expenses, manage budgets, and make smarter investment decisions. With a powerful stock prediction feature, we bring the future of finance to your fingertips. Our platform offers real-time insights, secure data handling, and a user-friendly interface designed to simplify financial management. Whether you're a beginner looking to save or an investor seeking growth, MoneyGuard has something for you. Start your journey toward financial stability today. Explore expense tracking, budgeting, and intelligent stock market predictions—all in one place.

[Finance Management](#)[Stock Prediction](#)

On clicking the stock prediction button on the page we are navigated to the stock prediction part.



On clicking the start predicting now , we can now get the real time share price, next day share price, and also the historical data related to the stock symbol.

The screenshot shows the MoneyGuard - Stock Prediction web application. The browser address bar displays "127.0.0.1:5000/start\_predicting". The page features a header with the title "Stock Price Prediction" and the tagline "Predict, Invest, Prosper, Repeat". Below the header, there are three main sections: "Predict Next Day's Stock Price", "Get Today's Stock Price", and "Get Historical Stock Data". Each section contains a text input field for "Enter Stock Symbol:" and a corresponding button ("Predict", "Get Today's Price", and "Get Historical Stock Data" respectively).

14. Firstly calculating the next day stock price by filling the stock symbol in the box.

This screenshot shows the MoneyGuard - Stock Prediction web application with the stock symbol "AAPL" entered in the "Predict Next Day's Stock Price" section. The browser address bar displays "127.0.0.1:5000/predict". The page layout is consistent with the previous screenshot, but the "Predict" button is now highlighted, indicating it has been clicked.

### Stock Price Prediction

Predict, Invest, Prosper, Repeat

#### Predicted Stock Prices for AAPL

##### Next Day's Predicted Prices

**Predicted Opening Price:** 198.60196

**Predicted Closing Price:** 199.14076

Now getting the real time stock price of the company AAPL i.e. apple , by fillin the stock symbol in the get todays stock price



### Stock Price Prediction

Predict, Invest, Prosper, Repeat

#### Predict Next Day's Stock Price

Enter Stock Symbol:

Predict

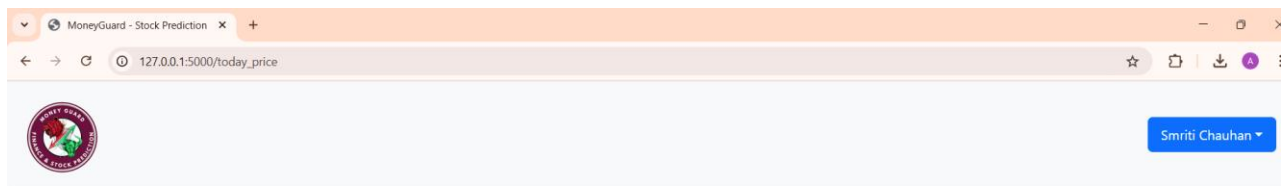
#### Get Today's Stock Price

Enter Stock Symbol:

AAPL

Get Today's Price

Get Historical Stock Data



### Stock Price Prediction

Predict, Invest, Prosper, Repeat

#### Today's Stock Price

AAPL - Today's Prices

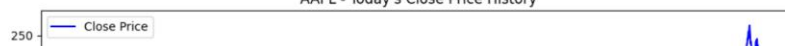
**Open Price:** Ticker AAPL 6.422876 Name: 2010-01-04 00:00:00, dtype: float64

**Close Price:** Ticker AAPL 203.190002 Name: 2025-04-03 00:00:00, dtype: float64

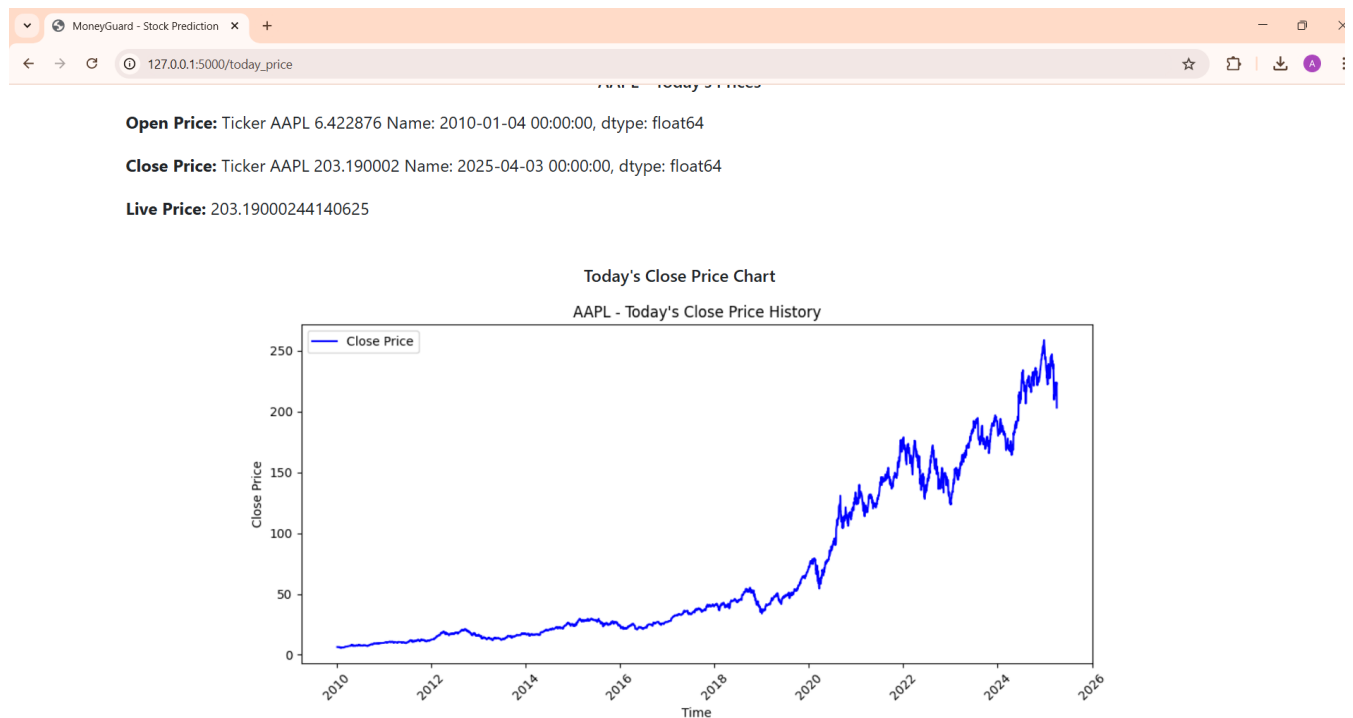
**Live Price:** 203.19000244140625

Today's Close Price Chart

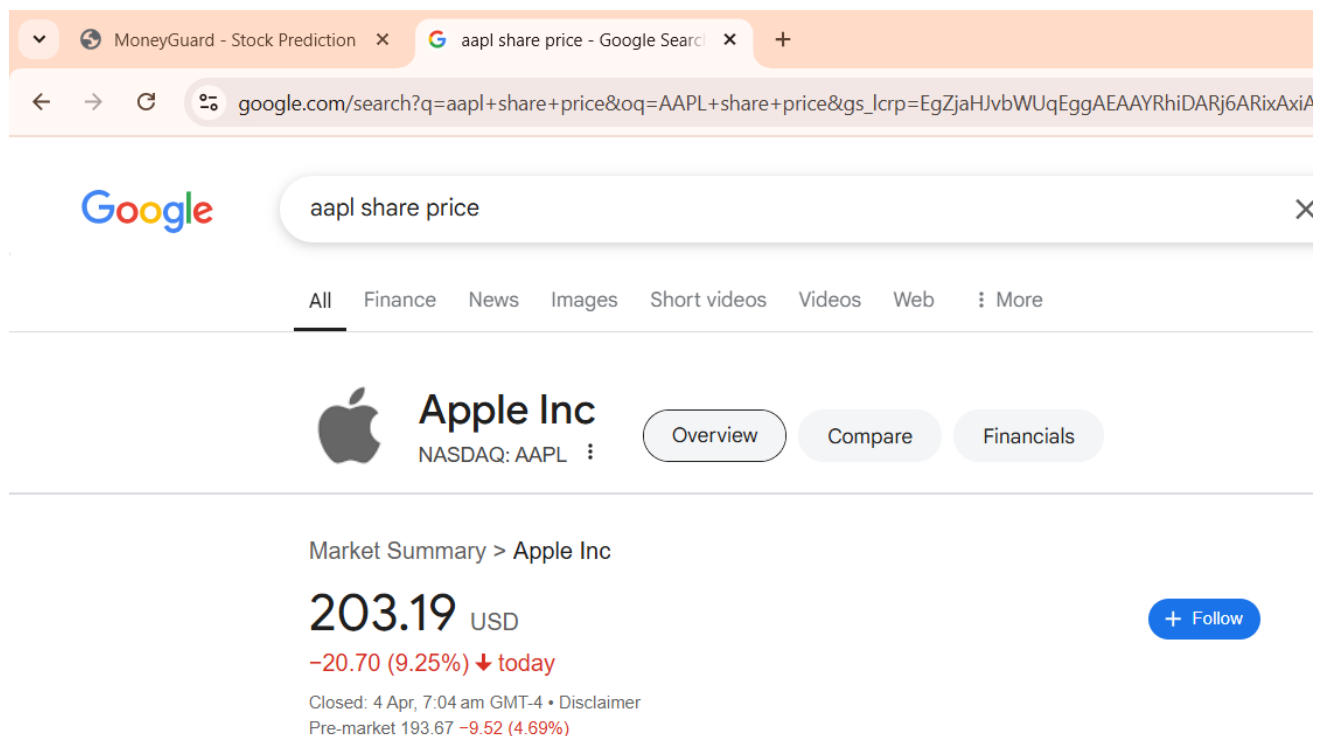
AAPL - Today's Close Price History



With the graph of todays close price history



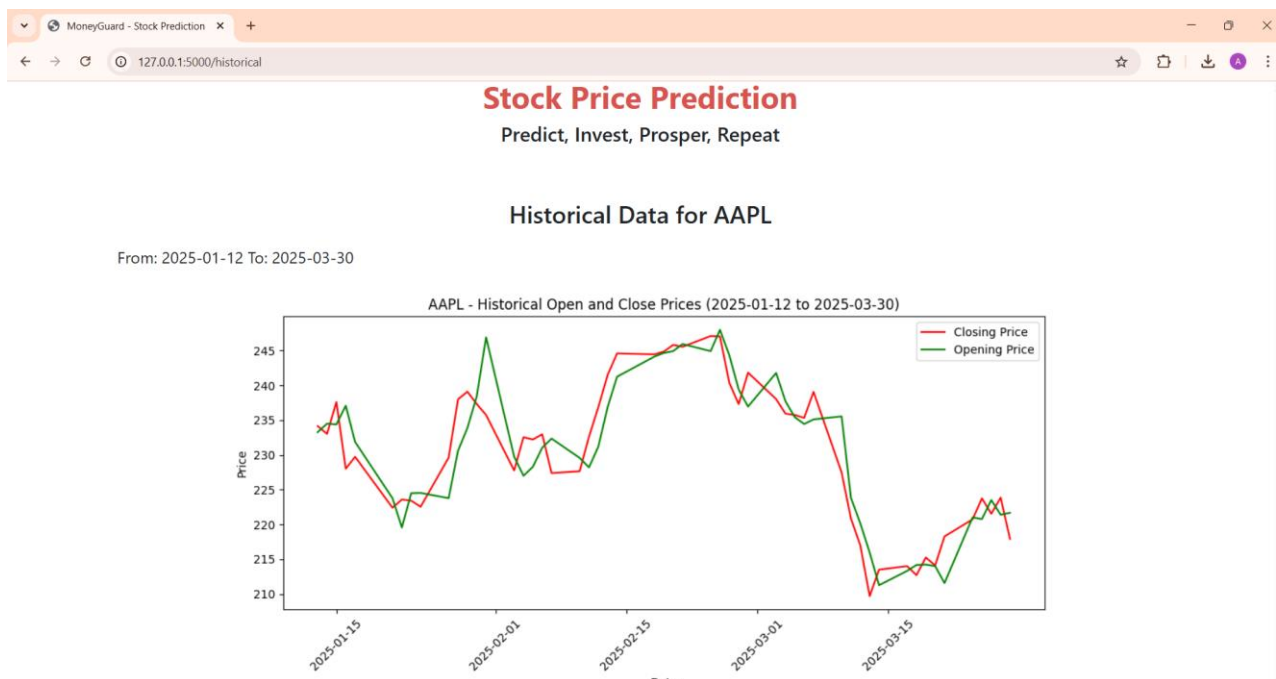
To check that is the real time stock price is accurate as on 4 april the price is same as shown in the live price in the stock prediction of Moneyguard



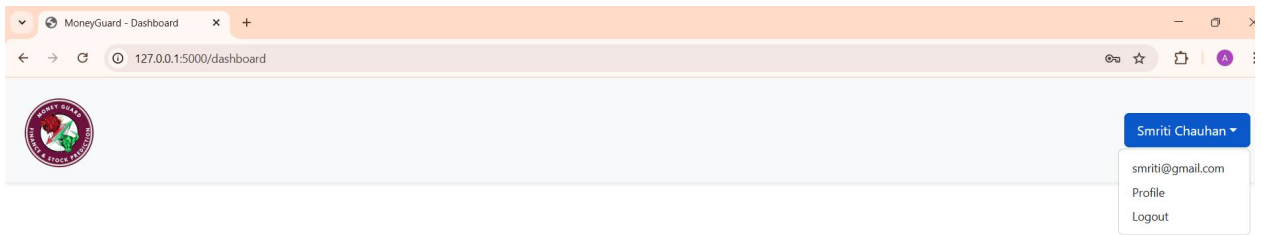
15. Now getting the historical share prices from the special data so in the stock prediction part of moneyguard , on entering the stock symbol with the specific dates and clicking the get historical data

The screenshot shows a web browser window with the URL `127.0.0.1:5000/start_predicting`. The page displays a form titled "Get Historical Stock Data". The form has three input fields: "Enter Stock Symbol:" with the value "AAPL", "Select Start Date:" with the value "12-01-2025", and "Select End Date:" with the value "30-03-2025". Each date field has a calendar icon. At the bottom of the form is a red button labeled "Get Historical Data".

We get this chart



16. On clicking the logout button in the dropdown menu after logging the moneyguard,



### LET'S START THE JOURNEY

MoneyGuard is your personal finance companion, helping you track expenses, manage budgets, and make smarter investment decisions. With a powerful stock prediction feature, we bring the future of finance to your fingertips. Our platform offers real-time insights, secure data handling, and a user-friendly interface designed to simplify financial management. Whether you're a beginner looking to save or an investor seeking growth, MoneyGuard has something for you. Start your journey toward financial stability today. Explore expense tracking, budgeting, and intelligent stock market predictions—all in one place.

[Finance Management](#)[Stock Prediction](#)

You are logout and navigated on the home page of moneyguard.

