**AN INDUSTRY ORIENTED MINI PROJECT REPORT ON**

**FINWISE**-**A Financial Advisor Application**

*in the partial fulfillment of the requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

in

**CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

**Submitted by**

**CH.SRI VARSHA 22B81A6674**

**R.ROSHINI 22B81A66A1**

**B.SANJAY 21B81A6645**

Under the guidance of

**Dr. H. N. Lakshmi**

**Assoc. Dean & Professor CSE(AI&ML)**



**CVR COLLEGE OF ENGINEERING**

**(*An Autonomous institution, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)**

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),  
Rangareddy (D), Telangana- 501 510

**APRIL 2025**

**CVR COLLEGE OF ENGINEERING**

**(*An Autonomous institution, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)**

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),  
Rangareddy (D), Telangana- 501 510

**DEPARTMENT OF**

**CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**



**CERTIFICATE**

This is to certify that the Industry Oriented Mini Project report entitled

**“FINWISE – A Financial Advisor Application”** is a bonafide record of work carried out by **CH. SRI VARSHA (22B81A6674), R. ROSHINI (22B81A66A1)** and **B. SANJAY (21B81A6645)** under the guidance of **Dr. H.N. Lakshmi**, **Assoc. Dean & Professor of CSE(AI&ML)** for the requirement of the award of **Bachelor of Technology** in **CSE(AI&ML)** submittedto the CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University, Hyderabad during the year 2024-2025.

**Project Guide Project Coordinator**

**Dr. H. N. Lakshmi Dr. A. Jagadeeswara Rao**

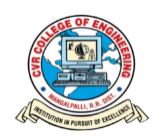
Assoc. Dean & Professor Professor

CSE(AI&ML)

**Head of the Department External Examiner**

**Dr. R. Usha Rani**

Professor**,** CSE(AI&ML)

**CVR COLLEGE OF ENGINEERING**

(UGC Autonomous Institution)

Affiliated to JNTU Hyderabad

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),

Ranga Reddy (Dist.), Hyderabad – 501510, Telangana State

**DECLARATION**

We hereby declare that the Industry Oriented Mini Project report entitled **“FINWISE- A Financial Advisor Application”** is an original work done and submitted to CSE (AI&ML) Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad in partial fulfilment for the requirement of the award of Bachelor of Technology in CSE (AI&ML) and it is a record of bonafide project work carried out by us under the guidance of **Dr. H. N. Lakshmi,** Assoc.Dean & Professor, Department of CSE (AI&ML).

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree in this Institute or any other Institute or University.

Signature of the Student

**CH. SRI VARSHA**

Signature of the Student

**R. ROSHINI**

Signature of the Student

**B. SANJAY**

**Date:**

**Place:**

**ACKNOWLEDGEMENT**

We are thankful for and fortunate enough to get constant encouragement, support, and guidance from all **Teaching** **staff of CSE (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING) Department** which helped us in successfully completing this Industry Oriented Mini Project.

We thank **Dr. A. Jagadeeswara Rao**, Project Coordinator and **Dr. Afreen Fatima Mohammed,** Project Review Committee members for their valuable guidance and support which helped us to complete the Industry Oriented Mini Project Work successfully.

We thank Professor and Head of the Department **Dr**. **R. Usha Rani,** for giving us all the support and guidance, which made us to complete the Industry Oriented Mini Project duly.

We would like to express heartful thanks to our internal guide andAssoc.Dean & Professor, Department of CSE (AI&ML), **Dr. H. N. Lakshmi** for providing us an opportunity and extending support and guidance.

We thank ourVice-Principal **Prof. L. C. Siva Reddy** for providing excellent computing facilities and a disciplined atmosphere for doing our work.

We wish a deep sense of gratitude and heartfelt thanks to our Principal **Dr. Rama Mohan Reddy** and the **Management** for providing excellent lab facilities and tools.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter No.** | | **Contents** | **Page No.** |
|  |  | List of Figures | i |
|  |  | Abbreviations | ii |
|  |  | Abstract |  |
| 1 |  | **Introduction** |  |
|  | 1.1 | Motivation | 1 |
|  | 1.2 | Problem Statement | 3 |
|  | 1.3 | Project Objectives | 5 |
|  | 1.4 | Project Report Organization | 8 |
| 2 |  | **Literature Survey** |  |
|  | 2.1 | Existing work | 9 |
|  | 2.2 | Limitations of Existing work | 12 |
| 3 |  | **Software & Hardware specifications** |  |
|  | 3.1 | Functional Requirements | 14 |
|  | 3.2 | Non-Functional Requirements | 17 |
|  | 3.3 | Software requirements | 19 |
|  | 3.4 | Hardware requirements | 20 |
| 4 |  | **Proposed System Design** |  |
|  | 4.1 | Proposed methods | 21 |
|  | 4.2 | Class Diagram | 24 |
|  | 4.3 | Use case Diagram | 25 |
|  | 4.4 | Activity Diagram | 26 |
|  | 4.5 | Sequence Diagram | 27 |
|  | 4.6 | System Architecture | 28 |
|  | 4.7 | Technology Description | 30 |
| 5 |  | **Implementation & Testing** |  |
|  | 5.1 | **Implementation** | 33 |
|  | 5.2 | **Testing** | 38 |
| 6 |  | **Conclusion & Future Scope** | 41 |
|  |  | **References:** | 43 |
|  |  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 4.2 | Class Digram | 24 |
| 4.3 | Use Case Diagram | 25 |
| 4.4 | Activity Diagram | 26 |
| 4.5 | Sequential Diagram | 27 |
| 4.6 | System Architecture Diagram | 28 |
| 5.1.1 | Log in page Screenshot | 34 |
| 5.1.2 | Index page Screenshot | 35 |
| 5.1.3 | Chatbot page Screenshot | 35 |
| 5.1.4 | Bill Reminders Screenshot | 36 |
| 5.1.5 | Expense Tracker Screenshot | 36 |
| 5.1.6 | Stock Price Prediction Page Screenshot | 37 |
| 5.2.4 | Chatbot query results | 39 |
| 5.2.5 | Stock Price Prediction with News Results | 40 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **Acronym** | **Full Form** |
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| ARIMA | Auto Regressive Integrated Moving Average |
| CCPA | California Consumer Privacy Act |
| CSS | Cascading Style Sheets |
| EMI | Equated Monthly Installment |
| GDPR | General Data Protection Regulation |
| HTML | HyperText Markup Language |
| JS | JavaScript |
| LSTM | Long Short-Term Memory |
| ML | Machine Learning |
| SHA-256 | Secure Hash Algorithm 256-bit |
| SIP | Systematic Investment Plan Amount |
| UI/UX | User Interface / User Experience |

**ABSTRACT**

Managing personal finances efficiently is essential in today’s digital age. This project focuses on developing a comprehensive AI-powered financial advisor designed to assist users in achieving their financial goals with ease and security. A key feature of the system is its **secure login functionality**, ensuring user privacy and the protection of sensitive financial data.

The platform facilitates **budget management and expense tracking**, helping users monitor spending habits, categorize expenses, and set personalized budgets. To enhance financial literacy, it provides **FAQs and educational content**, addressing common questions such as “Why should I invest?” and “What is an investment?” through concise and accurate explanations.

To prevent missed payments, the system includes **bill reminders** that notify users of upcoming due dates and helping them stay on top of their financial obligations. For investment enthusiasts, it offers **real-time stock analysis and portfolio tracking**, delivering actionable insights for smarter financial decisions. Additionally, the system incorporates a **hybrid LSTM + ARIMA model for stock price prediction**, helping users anticipate market trends with improved accuracy.

Users can also access **financial calculators** for fixed deposits, loans, EMIs, SIPs, mutual funds, and compound interest, assisting in long-term financial planning. By integrating advanced financial analytics and automation, this project empowers users to take control of their finances, reduce financial stress, and make informed decisions confidently. The system ensures a **secure, intuitive, and data-driven** approach to financial management, offering a **comprehensive solution for financial well-being**.

**1. INTRODUCTION**

The need for simplified and intelligent financial management tools has never been greater. With multiple financial responsibilities and limited awareness, users often struggle to make sound financial decisions. This project aims to bridge that gap by offering an AI-integrated platform that helps manage bills, track expenses, and predict market trends.

### ****1.1 Motivation****

In today’s increasingly complex financial environment, individuals are expected to make informed decisions regarding savings, investments, spending, and debt management. Yet, most users lack the tools or knowledge to do this effectively. This project is driven by multiple motivating factors that emphasize the urgent need for a smart, accessible, and integrated financial advisor system.

.

**1.1.2 Lack of Financial Literacy and Awareness**

Financial literacy remains alarmingly low across a large segment of the population, especially among youth and first-time earners. Many people are unaware of how to create a budget, the importance of compound interest, or how to make sound investment choices. This gap in financial education can lead to poor financial decisions, overspending, and lack of savings. Existing tools often overlook the educational aspect and focus solely on functionality. Our project is motivated by the need to embed financial learning into day-to-day management tools—turning every interaction into an opportunity to educate the user and empower them with practical financial knowledge.

**1.1.3 Missed Payments and Poor Financial Planning**

Late payment of utility bills, loan EMIs, and credit card dues is a widespread issue caused by forgetfulness, lack of reminders, or disorganized tracking systems. These oversights result in unnecessary late fees, financial stress, and negative credit implications. Simultaneously, many users fail to plan or stick to a monthly budget, leading to impulsive spending and insufficient savings. This scenario underlines the importance of automated tools that not only alert users of upcoming dues but also help them stay informed about their financial habits. Ensuring financial discipline through automation is a key motivator for building such a platform.

**1.1.4 Accessibility of Smart Financial Tools**

While financial tools and apps are abundant, many are either too advanced for casual users or too simplistic for serious financial planning. Some require subscriptions, while others offer only limited functionality. Moreover, tools that are technically powerful often lack user-friendly design, making them intimidating or inconvenient for everyday use. There’s a noticeable gap in the market for a platform that brings the power of intelligent analytics and financial tools to users of all backgrounds—whether tech-savvy or not. This project is driven by the vision of creating a solution that is powerful yet approachable, offering advanced features in an intuitive and accessible manner.

**1.1.5 Leveraging AI for Personalized Financial Guidance**

With advancements in artificial intelligence, it’s now possible to offer financial guidance that is not just automated but also personalized and adaptive. AI can interpret user behavior, track financial patterns, and even forecast future trends to provide highly relevant advice. Chatbots can offer instant support, and predictive models can help users anticipate stock performance or spending outcomes. Our motivation lies in utilizing these AI capabilities to bridge the gap between human financial advisors and digital tools—creating an assistant that’s available 24/7, continuously learning from the user, and delivering real-time, intelligent financial assistance tailored to individual needs.

**1.1.6 Demand for All-in-One Financial Ecosystems**

Users today expect convenience and integration from digital solutions. Whether it's banking, budgeting, investing, or learning, people prefer a single platform where they can access everything they need without juggling multiple apps. This expectation has yet to be fully met in the personal finance domain. Most existing tools solve isolated problems but fail to provide a connected, end-to-end financial experience. The motivation to build a unified financial advisor stems from this unmet demand. By consolidating diverse financial tools into one ecosystem—backed by AI and a user-friendly interface—we aim to create a seamless and consistent financial journey for users.

**1.2 PROBLEM STATEMENT**

Despite the growing number of financial apps and tools, most users still struggle to manage their money effectively. This is largely due to the lack of unified platforms and limited financial awareness, especially among younger audiences. The following issues highlight the need for a smarter, more inclusive solution.

### ****1.2.1 Lack of Comprehensive Financial Management Tools****

Most existing apps focus on one function—budgeting, expense tracking, or investments forcing users to juggle multiple platforms. This fragmented experience causes confusion and inefficiency. What’s missing is a centralized system that offers an integrated approach to financial well-being through a single interface. Users should not be expected to manage budgeting, expense tracking, stock monitoring, and calculators across disconnected systems. A truly efficient financial advisor system must centralize these components to offer clarity, consistency, and time-saving convenience.

**1.2.2 Low Financial Literacy Among Users**

A significant percentage of people, particularly students and young professionals, lack the foundational knowledge needed to make sound financial decisions. Concepts like compound interest, SIPs, budgeting strategies, and stock market basics often feel intimidating or inaccessible to them. As a result, users are prone to mismanaging funds, making poor investment choices, or avoiding financial planning altogether. Most digital tools assume users already possess a baseline understanding of these topics, which is rarely the case. There's a growing need for systems that not only assist with management tasks but also **educate users** through interactive, simplified explanations that improve financial literacy. Without education, even the best tools risk being underutilized or misunderstood.

**1.2.3 Missed Payments and Poor Tracking Habits**

Bill payment deadlines, loan EMIs, and recurring subscriptions are frequently missed by users due to forgetfulness or a lack of structured reminders. These delays often incur penalties and negatively impact credit scores. Alongside this, many users have inconsistent habits when it comes to tracking day-to-day expenses or reviewing their monthly financial activities. Manual tracking methods are tedious and error-prone, and existing applications often fail to provide a proactive solution. This lack of discipline and automated support results in users being unaware of their spending patterns and financial leaks. There is a clear need for automated reminders, intuitive dashboards, and smart budgeting suggestions that can keep users on track effortlessly.

**1.2.4 Absence of Intelligent, Personalized Financial Assistance**

The financial needs of users vary drastically—some aim to reduce debt, others focus on investing, and many want to build better saving habits. However, most platforms provide one-size-fits-all solutions with static tips or generic content. They lack the intelligence to personalize advice based on individual user data such as income, spending trends, or investment preferences. Moreover, very few tools make use of AI-powered technologies like chatbots or real-time behavioral analysis to adapt and evolve with the user. Without intelligent assistance, users miss out on key opportunities to optimize their financial decisions or receive timely, contextual support when they need it most.

**1.2.5 Lack of Financial Behavior Insights and Predictive Planning**

Most existing platforms focus on tracking past transactions or offering static suggestions, but they lack the ability to analyze user behavior over time and provide forward-looking insights. Users often struggle to understand how their current financial habits might affect their future goals. Without predictive planning tools and trend analysis, it becomes difficult for individuals to proactively adjust their strategies. This project addresses the problem by emphasizing intelligent analytics that go beyond surface-level data—offering users insights into spending patterns, forecasting potential shortfalls, and simulating outcomes based on their actions. The absence of such behavioral and predictive tools in most systems reinforces the need for a smarter, more anticipatory financial assistant.

**1.2.6 No Unified Application Offering All Core Functionalities**

While standalone apps exist for tasks like expense tracking, bill reminders, and financial calculators, users are forced to install and manage multiple tools to cover their full financial ecosystem. This leads to fragmented data, inconsistencies in planning, and a disjointed user experience. As financial planning becomes increasingly essential for long-term stability, there is a strong demand for an all-in-one system that consolidates these features into a single, intuitive platform. The absence of such a unified application that merges AI-driven predictions, smart advice, educational content, secure login, and real-time tracking remains a major gap in today’s digital finance market. Addressing this gap would revolutionize the way users approach personal finance, making it more accessible, efficient, and impactful.

**1.3 PROJECT OBJECTIVES**

The objective of this project is to design and develop a comprehensive AI-powered financial assistant that addresses the diverse financial needs of users in a single, unified platform. By integrating advanced analytics, predictive models, educational content, and intelligent automation, the project intends to enhance financial literacy, streamline financial tasks, and empower users to make informed decisions. The following objectives outline the core goals of the system:

**1.3.1 Develop an AI-Powered Financial Assistant**

The core objective of this project is to build a smart AI-driven chatbot that acts as a 24/7 financial advisor. It should be capable of interpreting user queries related to finance and providing helpful responses in real time. The assistant should mimic human-like conversation while maintaining accuracy and professionalism. It will guide users through financial concepts, budgeting decisions, and provide recommendations. and preferences. The chatbot offers consistent and accurate responses to user queries, enabling users to access essential financial information instantly and does not retain conversation history. Its integration will reduce the need for manual financial research and bring expert-level support to every user’s fingertips.

**1.3.2 Enable Accurate Stock Price Prediction**

The system aims to provide reliable short-term and long-term stock market predictions using a hybrid model that combines Long Short-Term Memory (LSTM) neural networks and AutoRegressive Integrated Moving Average (ARIMA) time series analysis. This approach ensures that both seasonal trends and nonlinear dependencies in stock price data are captured. Users will benefit from confidence bounds (upper and lower limits) that indicate the expected range of stock price movement. By offering data-backed insights, the platform supports smarter investment strategies, helping users identify trends and risks. This feature bridges the gap between everyday investors and complex market analytics.

Predictions are available only for publicly listed companies supported by the **Yahoo Finance (yfinance)** API, and require at least **three years of historical data** for accurate modeling. Additionally, these predictions are generated under the assumption of **normal market conditions**—they may not hold in highly volatile or extreme scenarios such as financial crashes or unexpected global events.

**1.3.3 Provide Comprehensive Expense Tracking and Budgeting**

A major objective is to help users monitor their income and expenditures in a structured and insightful manner. The system will allow users to log expenses under different categories (e.g., food, rent, entertainment) and analyze monthly trends. Users can set customizable budgets and receive alerts when they are approaching or exceeding limits. Visual tools such as charts and dashboards will enhance understanding and awareness. The goal is to build a long-term habit of financial discipline by making tracking intuitive and rewarding. This feature not only promotes saving but also gives users a clearer picture of their financial behavior over time.

**1.3.4 Offer Automated Bill Reminders and In-app alerts**

To address the common issue of missed payments, the project includes a reliable notification system that alerts users ahead of due dates. This feature supports various types of bills, including utilities, subscriptions, and EMIs. Reminders will be timely and non-intrusive, allowing users to act without stress or delay. Additionally, a bill reminder feature helps users stay informed about upcoming financial obligations by sending timely alerts, ensuring they don’t miss important due dates. The system aims to improve overall financial discipline and reduce avoidable expenses like late fees. Ensuring timely payments also contributes to a better credit score and peace of mind for users.

**1.3.5 Incorporate Smart Financial Calculators**

The platform will include user-friendly financial calculators for a range of use cases such as EMI computation, fixed deposit interest, SIP growth, loan amortization, and compound interest accumulation. These tools allow users to simulate various financial scenarios by inputting custom values. The calculators will provide detailed breakdowns and visual outputs to help users understand the implications of their financial plans. Whether someone is planning a large purchase or mapping out a long-term savings goal, these calculators serve as essential decision-making aids. Their inclusion adds practical utility and encourages proactive planning.

**1.3.6 Enhance Financial Literacy Through Education and FAQs**

Many users struggle with financial jargon and lack the confidence to manage their finances independently. To address this, the chatbot will offer clear, accessible educational content in the form of FAQs and short explanations on topics such as “What is an investment?” or “How does compound interest work?” This feature transforms the chatbot into not just a tool, but also a learning companion. It will promote financial independence by equipping users with knowledge tailored to their queries. By building financial awareness gradually, the system contributes to long-term behavioral change and better financial decision-making.

**1.3.7 To Integrate Real-Time Financial Data for Enhanced Decision-Making**  
The system will incorporate real-time financial data, including stock market trends and economic news updates, to help users stay informed and make timely investment decisions. By leveraging APIs for live data retrieval, the platform enhances the relevance and accuracy of its financial predictions and recommendations. This integration ensures that users are not solely reliant on static insights but have access to dynamic, up-to-date information tailored to their financial goals.

**1.3.8 Deliver a Unified and Seamless User Experience**

An essential objective of this project is to ensure that all individual features from stock prediction and expense tracking to bill reminders and financial calculators, are integrated within a single cohesive platform. Users should be able to access every functionality without switching between different apps or interfaces. This holistic design reduces friction, increases usability, and encourages long-term engagement. The user interface will be intuitive, responsive, and optimized for both desktop and mobile platforms to cater to a diverse user base. By focusing on user-centric design and experience, the system aims to make financial management feel accessible, efficient, and even enjoyable for users of all financial literacy levels.

**1.4 PROJECT REPORT ORGANIZATION**

This project report is structured into several key chapters that provide a comprehensive overview of **FINWISE – A Financial Advisor Application**, detailing its conceptual foundation, technical design, and real-world applicability. Each chapter is organized to ensure a smooth flow of information from motivation to future enhancements, enabling the reader to understand both the vision and execution behind the project.

**1.4.1 Introduction** This chapter introduces FINWISE by highlighting the motivation for building a smart financial advisor application. It outlines the challenges faced by individuals in managing finances and defines the project’s core objectives, problem statement, and scope.

**1.4.2 Literature Survey** This section explores existing financial applications and systems, analyzing their limitations and gaps. It also reviews relevant technologies and research in the field of financial forecasting, budgeting, and advisory tools, which laid the groundwork for FINWISE.

**1.4.3 Software & Hardware Specifications** This chapter specifies the functional and non-functional requirements, along with detailed software and hardware specifications. It outlines the development environment, tools used, and system prerequisites needed to run the application efficiently.

**1.4.4 Proposed System Design** The design chapter provides the technical architecture and modeling of the system. It includes diagrams such as class, use case, activity, and sequence diagrams, as well as an explanation of the technologies used and the logical flow between modules.

**1.4.5 Implementation & Testing** This section discusses the practical implementation of FINWISE, detailing the development process module by module. It also describes the testing strategies adopted to ensure the accuracy, security, and usability of the application.

**1.4.6 Conclusion & Future Scope** The final chapter presents the overall outcomes of the project, reflecting on its success and user impact. It also identifies potential improvements, additional features, and how FINWISE can adapt to evolving trends in personal finance and artificial intelligence.

**2. LITERATURE SURVEY**

As the demand for smarter financial tools continues to rise, several applications and systems have emerged that cater to specific aspects of personal finance. However, these tools are often fragmented and lack the cohesion required for comprehensive financial guidance. This chapter explores existing solutions and technologies that have influenced the design of FINWISE.

### ****2.1 EXISTING WORK****

This section discusses the currently available technologies, applications, and systems in the financial domain that address individual components of personal finance management. While several specialized tools exist, they typically offer isolated functionalities such as budgeting, investment tracking, or chatbot-based query handling. FINWISE, by contrast, aims to bridge the gap between these fragmented services by providing an all-in-one intelligent financial advisor. The following subsections review prominent systems relevant to FINWISE’s core features:

**2.1.1 Expense Tracking & Budgeting Tools**

Apps like Mint, YNAB (You Need A Budget), Spendee, and PocketGuard are popular tools for tracking income, expenses, and setting budgeting goals. Mint, for instance, automatically categorizes transactions, while YNAB focuses on proactive budget planning. Despite their widespread use, these tools suffer from limitations such as limited AI capabilities, a lack of personalized insights, and a one-size-fits-all approach to budgeting (Brownlee, 2022). **FINWISE** overcomes these limitations by adapting to a user's evolving financial behavior, offering more dynamic and personalized insights through AI-driven analytics.

### ****2.1.2 AI-Based Chatbots in Finance****

### AI-driven chatbots, such as Cleo, Plum, Erica (Bank of America), and KAI, provide basic financial assistance through natural language interfaces. These bots help users check balances, track spending, and set savings goals. While useful, these chatbots often operate on rule-based or script-driven logic, limiting their ability to provide in-depth financial analysis or personalized insights (Jain & Agarwal, 2023). ****FINWISE**** enhances this functionality by integrating deep learning models and combining educational content with personalized financial planning tools.

### ****2.1.3 Stock Prediction Applications****

### Platforms such as TradingView, Zerodha Varsity, Robinhood, and Yahoo Finance offer valuable stock data visualization, market trends, and community-driven insights. However, these tools often require users to interpret charts and perform manual technical analysis. Some tools rely on historical performance and indicators like RSI, MACD, or Bollinger Bands, but rarely employ deep learning techniques for predictive modeling. Moreover, real-time prediction with quantified confidence levels is usually absent. ****FINWISE**** enhances this landscape by implementing a hybrid LSTM + ARIMA model that captures both long-term patterns and short-term fluctuations, while also displaying upper and lower confidence bounds, helping users assess risk tolerance and forecast precision — features not commonly found in public-facing financial tools (Zhang et al., 2021).

### ****2.1.4 Bill Reminder and Payment Apps****

### Applications like Prism, Mobills, and Bills Monitor are specifically designed to send reminders for utility bills, credit card payments, and subscription renewals. These apps are useful for maintaining payment discipline and avoiding late fees. However, most are standalone utilities that don’t integrate with broader financial planning tools such as budget planners, investment portfolios, or savings trackers. Some require manual input of bill details, which can be tedious and prone to user error. Additionally, users must often rely on multiple apps to manage different aspects of their financial lives. ****FINWISE**** overcomes these issues by integrating with a broader view bill reminders of the user's financial activity through features like expense tracking, ensuring users stay informed and in control of their spending (Gensler, 2021).

### ****2.1.5 Financial Calculators & Educational Platforms****

Online calculators provided by websites like BankBazaar, HDFC, ICICI, and Investopedia are widely used to compute EMIs, SIPs, mutual fund returns, loan eligibility, and interest accruals. However, these calculators are usually generic, isolated tools that don’t store data or connect to any larger financial strategy. They lack user personalization, contextual guidance, and AI-driven suggestions. Similarly, educational resources like Investopedia, MyMoneyCoach, or Khan Academy’s finance modules are static in nature, offering valuable theory but no practical interactivity or integration with real-time data. **FINWISE** fills this void by embedding intelligent calculators within its system, linked to a user’s financial profile, and by providing on-demand financial education that responds directly to user queries through the chatbot interface (Krug, 2020).

**2.1.6 Integrated Financial Planning Tools**

Existing financial tools typically focus on isolated aspects of personal finance, such as budgeting, saving, or investing. While apps like Mint or YNAB excel in expense tracking, they often lack integration with broader financial planning needs. Additionally, tools for investment tracking, like Robinhood, provide real-time stock data but do not offer a comprehensive strategy for long-term wealth management. **FINWISE** bridges these gaps by combining budgeting, expense tracking, investment monitoring, and goal setting within a single platform. The system's AI-driven financial planning tools analyze user behavior and offer personalized recommendations, helping users achieve long-term financial goals while managing their day-to-day financial activities effectively (Mathew, 2022).

**2.2 LIMITAIONS OF EXISTING WORK**

While existing financial applications and tools have individually addressed various aspects of personal finance management, they fall short in delivering a **holistic, intelligent, and adaptive** solution. Most of these platforms operate in silos, offering limited interoperability and minimal personalization. This section outlines the key limitations that highlight the need for a unified solution like **FINWISE**.

### ****2.2.1 Fragmented User Experience Across Financial Tools****

Many financial applications focus on a single domain — such as budgeting, investment, or bill tracking — forcing users to switch between multiple apps to manage their finances effectively. This fragmentation not only leads to inefficiency but also reduces the user's ability to gain a consolidated view of their financial status. For instance, while Mint tracks expenses, it does not support investment management; similarly, Robinhood offers trading features but lacks budgeting tools. This scattered ecosystem creates cognitive overload, increased effort, and difficulty in forming long-term financial strategies. **FINWISE** addresses this challenge by providing a centralized platform where users can manage all aspects of their financial life in one place (Zacks Investment Research, 2023).

### ****2.2.2 Lack of Intelligent Predictive Analysis****

Most personal finance applications rely on historical data visualization, static charts, and trend summaries. Very few employ machine learning or deep learning models to perform predictive analytics. Users are left to interpret data manually and make investment or budgeting decisions without algorithmic support. Even platforms that claim to be "smart" typically use basic statistical models or rule-based engines. **FINWISE**, in contrast, introduces a **hybrid LSTM + ARIMA** model, enabling users to receive **data-driven stock price predictions** with confidence intervals — something rarely seen in consumer-facing financial tools. (Zhang et al., 2021).

### ****2.2.3 Limited Personalization and Adaptive Learning****

Traditional finance management systems do not adapt to the evolving behavior of individual users. They offer generic recommendations or fixed templates, which may not suit varying financial goals, risk tolerance, or life stages. For example, budgeting suggestions remain static even when a user’s income or expenditure patterns change. FINWISE leverages AI through a static chatbot and rule-based reminders, offering helpful financial assistance. It ensures relevant insights without storing user data, preserving privacy. (Google Cloud, 2024).

**2.2.4 Inadequate Financial Literacy Support**

Most apps assume a baseline level of financial literacy, offering limited help for users who are new to managing money or investments. Educational content, if present, is usually provided as static FAQs or external blog links, with no interactivity. This approach can be overwhelming or unengaging for users unfamiliar with financial jargon or investment strategies. **FINWISE** integrates an **AI-powered educational chatbot** capable of answering user queries in natural language, providing on-demand, contextual learning tailored to user questions like “What is SIP?” or “How does compound interest work?”

### ****2.2.5 Security and User Privacy Concerns****

Many existing platforms collect sensitive financial information but fail to implement strong authentication mechanisms or privacy protections. Users are often required to input personal data without understanding how it is stored, processed, or protected. Additionally, the lack of secure login features in some apps can make users vulnerable to data breaches. **FINWISE** ensures secure access through encrypted login using **SHA-256**, maintaining user privacy without storing any personal or financial data. (StackPath, 2022).

### ****2.2.6 Absence of an All-in-One Financial Advisor Application****

Despite the presence of multiple tools in the market, there is **no single application** that combines **expense tracking, smart budgeting, stock price prediction, bill reminders, financial education, and personalized advisory** — all in one ecosystem. Users are forced to stitch together their financial routines using multiple, often incompatible platforms. **FINWISE** fills this gap by acting as a **complete AI-powered financial advisor**, offering a seamless, interactive, and highly functional experience that evolves with user needs.

### ****2.2.7 Minimal Integration of Automation and Proactive Alerts****

Many existing financial applications are reactive rather than proactive — they display information only when users seek it, with minimal automation or timely In-app alerts. Critical features such as **automated bill reminders**, **low balance alerts**, or **overspending warnings** are either missing or poorly implemented. This passive approach results in missed payments, overdraft fees, and delayed decision-making. Additionally, few platforms offer calendar integration or automated prompts to help users stay on track with their financial plans. **FINWISE** overcomes this by incorporating **intelligent automation**, including proactive **bill reminders**, **budget alerts**, and In-app alerts, ensuring users are consistently informed and in control of their finances without needing to manually check the app. (Gensler, 2021).

**3. SOFTWARE AND HARDWARE SPECIFICATIONS**

This chapter outlines the essential software and hardware components required for the development and operation of the FINWISE platform. It covers both the system's functional requirements and the technical infrastructure needed to ensure smooth, secure, and efficient operation of the platform.

**3.1 FUNCTIONAL REQUIREMENTS**

The functional requirements define the key capabilities and features of the FINWISE platform that support its core functionalities. These requirements ensure the system meets user expectations and delivers a seamless financial management experience, from secure login to personalized financial insights.

### ****3.1.1 User Authentication****

FINWISE implements a secure login system to ensure that only authorized users can access their personalized financial data. Upon registration, users are required to provide credentials which are validated during login. This ensures the protection of sensitive financial information and personalized settings. The authentication system enables each user to have a separate dashboard that stores their reminders, budget preferences, and financial insights securely. Although password reset functionality is not currently implemented, the system maintains session-based access control for privacy. This functional module forms the base of the platform’s data protection and personalization mechanisms.

### ****3.1.2 Expense Tracking & Budgeting****

The application provides users with the ability to input and categorize their expenses, allowing for meaningful analysis of their financial behavior. Users can view summaries of spending patterns across various categories like food, travel, and shopping, helping them make informed budgeting decisions. The budgeting module visualizes monthly spend trends and compares them against user-defined limits, encouraging financial discipline. The simple yet effective UI supports adding, editing, and deleting expense entries for flexibility. This functionality empowers users to set financial goals and monitor progress consistently through intuitive dashboards.

### ****3.1.3 Financial Calculators****

FINWISE includes a suite of interactive calculators tailored for different financial needs. These include EMI calculators, SIP planners, compound interest estimators, and tools for fixed deposits and mutual fund forecasting. Users can input parameters such as principal amount, interest rate, time duration, and get instant computed results. These calculators help users evaluate their investment options, loan commitments, and future savings potential without needing third-party tools. The calculators are embedded within a clean and accessible interface that simplifies complex financial computations. This module enhances users’ decision-making by making financial planning more approachable.

### ****3.1.4 AI Chatbot for Financial Advice****

An intelligent AI-powered chatbot is integrated to assist users with real-time financial advice. It can answer a variety of queries such as "Why should I invest?" or "How to save more effectively?" The chatbot is designed to enhance financial literacy through natural conversation and provide users with simple explanations to complex financial concepts. It adds a personalized touch by understanding user context and gives financial suggestions. This functionality bridges the gap between static information and active engagement, making FINWISE a more interactive and helpful platform.

### ****3.1.5 Bill Reminders****

FINWISE allows users to schedule and manage bill payment reminders to avoid late payments and penalties. Through the reminders interface, users can input due dates, payment categories, and optional notes. The system displays upcoming due bills prominently, helping users prioritize financial responsibilities. Although real-time In-app alerts or integrations with payment systems are not implemented, the feature still helps users stay organized and timely with their expenses. It serves as a digital planner focused specifically on recurring financial obligations.

### ****3.1.6 Stock Price Prediction****

The application incorporates a smart forecasting module that utilizes a hybrid machine learning model to predict stock prices. By combining Long Short-Term Memory (LSTM) networks and ARIMA, the system captures both long-term patterns and short-term fluctuations in stock trends. Users can input a stock symbol and view predicted price ranges along with upper and lower confidence bounds. This empowers users to make informed investment decisions based on AI-driven insights. While this feature currently serves as a research-based tool, it demonstrates the potential of AI in financial prediction and strategy planning.

### ****3.1.7 FAQ & Financial Education****

To improve financial awareness, FINWISE includes a structured FAQ section that answers common queries regarding investments, savings, and personal finance. This educational content is presented in simple language for users of all backgrounds and age groups. Topics covered include reasons to invest, budgeting tips, and the significance of saving early. The inclusion of this module addresses the knowledge gap many users face in managing their money wisely. It supports the platform’s broader mission to not just manage money, but to help users understand it.

**3.1.8 Contact Us**

The "Contact Us" feature allows users to send messages or inquiries directly through the platform. This functionality enables users to easily reach out for support or provide feedback. Upon submission, the message is sent to the system’s designated support email, ensuring that the team receives and can respond to user concerns promptly. The feature ensures seamless communication between users and the support team, enhancing user satisfaction and addressing any issues or queries efficiently.

**3.2 NON FUNCITONAL REQUIREMTNS**

The non-functional requirements of the **FINWISE** application focus on delivering a secure, high-performing, scalable, and user-friendly financial advisory platform. These attributes are critical for ensuring user trust, long-term reliability, and optimal usability, especially when dealing with sensitive financial data and complex user interactions.

#### 3.2.1 Performance

FINWISE is developed to deliver a responsive and efficient user experience, even in its current local development environment. For most functionalities such as financial calculators, expense tracking, chatbot interactions, and news search, the response time is almost instantaneous, typically well under a few seconds. These modules are lightweight and optimized for quick user interaction.

However, the stock price prediction module, which uses a deep learning model (LSTM with 30 epochs), requires more computational time, resulting in a response time of up to 150 seconds for that specific feature. This is currently acceptable during the development phase and is due to the real-time training and prediction being performed locally.

In future production deployments, this delay will be significantly reduced through the use of pre-trained models, background processing, and efficient deployment strategies such as server-side prediction and GPU acceleration. Overall, the system will be optimized to ensure a seamless and timely experience across all modules.

### 3.2.2 Security

Security is a fundamental requirement for **FINWISE**, particularly in safeguarding sensitive financial data. User credentials will be securely stored in **MongoDB**, utilizing **SHA (Secure Hash Algorithm)** for password hashing to ensure that passwords are stored in a non-reversible format. In addition, **FINWISE** will implement secure authentication mechanisms to protect against unauthorized access. Although the application is running on localhost during development, best practices for security will be followed, including encryption of sensitive data in transit. As the application transitions to a production environment, further security measures will be implemented, such as multi-factor authentication (MFA), and the app will comply with data protection regulations like GDPR and CCPA to ensure user data privacy.

#### 3.2.3 Usability

The usability of **FINWISE** is crucial to providing an intuitive and effective user experience. The user interface should be designed to be intuitive and easy to navigate, ensuring that users can access key features like financial calculators and expense tracking without difficulty. During the development phase, usability will be tested locally to ensure that the interface is user-friendly. As the application moves toward production, further refinements will be made to improve user interaction and provide a more polished experience suitable for a broader audience.

#### 3.2.4 Scalability

Although **FINWISE** is currently running on localhost, scalability is an important consideration for future growth. The application should be designed with scalability in mind, ensuring that as the user base increases, the backend infrastructure can be scaled to handle larger volumes of data and user interactions. Once deployed to a cloud or distributed environment, the system will be optimized to support load balancing, horizontal scaling, and database optimization to meet increasing demands.

#### 3.2.5 Reliability

Reliability is essential for maintaining the consistent operation of **FINWISE**. During the local development phase, the application should be stable and able to recover from basic failures, such as server restarts. Once the application is deployed to a production environment, reliability requirements will expand to include high availability, with an uptime target of 99.9%. The system will implement failover mechanisms, data backups, and continuous monitoring to detect and resolve issues quickly, ensuring the application remains operational at all times.

**3.3 SOFTWARE REQUIREMENTS**

The following software components are required to run the **FINWISE** application in its current development environment. While the application may operate on older systems, the listed specifications are **recommended for smooth and reliable performance**. Compatibility with outdated software is not guaranteed due to dependency on newer Python versions and library support. These dependencies enable all core features of FINWISE, including the chatbot, calculators, and stock prediction functionalities.

#### 3.3.1 Operating System

Recommended operating systems for hosting and running the application:

* **Windows 10/11**
* **Ubuntu 20.04 or later**
* **macOS Monterey or later**

#### 3.3.2 Required Software

The application is developed using Python and runs locally through a Flask server. The following software must be installed:

* **Python 3.8 or higher**  
  Essential for running backend logic, machine learning modules, and web server operations.
* **Python Libraries:**  
  The following packages must be installed using pip:
  + Flask – Lightweight web framework to serve the application locally
  + TensorFlow, Keras – Required for stock price prediction using LSTM
  + pandas, numpy, matplotlib, – For financial data processing and visualization
  + sklearn – For ARIMA and additional model utilities
  + PyMongo – Enables backend connection to the remote MongoDB database
  + googlegenerativeai and requests – for chatbot and news on stock

All dependencies are managed via a requirements.txt file for easy setup.

#### 3.3.3 Web Browser

The user interface is accessible via any modern web browser through localhost.

* **Supported Browsers:**
  + Google Chrome (recommended)
  + Mozilla Firefox, Microsoft Edge, Safari etc

### ****3.4 HARDWARE REQUIREMNTS****

This section outlines the **minimum and recommended hardware specifications** for running the **FINWISE** application in its current development setup (hosted locally). These requirements are intended for the system on which the backend and machine learning modules are executed.

**Note:** These specifications are for running the application locally. End-users accessing the application via browser do not need any specialized hardware.

#### **3.4.1 Minimum Hardware Requirements**

* **Processor:** Intel Core i3 (or equivalent AMD processor)
* **RAM:** 4 GB
* **Storage:** At least 2 GB of free disk space
* **Display:** 720p resolution or higher
* **Network:** Basic internet connectivity (to access hosted MongoDB and external resources)

These specifications are sufficient for running the application with basic features enabled, excluding real-time stock prediction under heavy load.

#### **3.4.2 Recommended Hardware Requirements**

* **Processor:** Intel Core i5 (8th Gen or newer) or AMD Ryzen 5
* **RAM:** 8 GB or higher (for efficient execution of LSTM model and multitasking)
* **Storage:** SSD with at least 5 GB of free space
* **Display:** Full HD (1080p) resolution
* **Network:** Stable broadband internet connection

These specifications are recommended for users intending to run FINWISE locally with all features enabled, particularly stock prediction using deep learning

**4. PROPOSED SYSTEM DESIGN**

This chapter presents the proposed system design for FINWISE, which aims to provide a comprehensive, user-friendly financial management tool. The system is designed to be lightweight, modular, and highly interactive, ensuring an efficient experience for users. Key components, such as the AI-powered chatbot, expense tracking, financial calculators, stock price prediction, and real-time stock-related news, are seamlessly integrated into the platform. The design ensures that each module works cohesively within a local development environment, enabling users to access all features without requiring constant internet connectivity. The architecture prioritizes scalability, security, and ease of use, forming the foundation for a robust financial advisory tool.

**4.1 PROPOSED METHODS**

The proposed system design for **FINWISE** presents a lightweight, modular, and user-friendly financial advisory tool that runs locally. The system includes features such as a chatbot, expense manager, financial calculators, stock price prediction, and stock-related news headlines. The architecture ensures smooth functionality within a local development environment.

#### ****4.1.1 System Architecture****

FINWISE is built using a **client-server architecture**, where the frontend interacts with backend Python scripts through a Flask server running locally.

* **Frontend**:  
  The interface is developed using **HTML, CSS, and JavaScript**, providing easy navigation across modules such as calculators, chatbot, and stock prediction. The pages are rendered and served via Flask routes.
* **Backend**:  
  The backend uses **Flask (Python)** to handle:
  + Form submissions (e.g., expense input)
  + Chatbot query handling
  + Stock price prediction using **LSTM** (with optional ARIMA support)
  + News retrieval based on stock tickers The backend logic also interfaces with the database through **PyMongo**.
* **Database (MongoDB)**:  
  User credentials and expense data are stored in **MongoDB**. This structure allows for efficient retrieval and storage of personal finance data.

#### ****4.1.2 User Interface Design****

The user interface of FINWISE is simple and structured, aimed at providing clarity for all users. Key components include:

* **Home Page** – Centralized entry point with navigation to all features
* **Expense Tracker** - Allows users to monitor expenses
* Bill Reminders – Allows users to monitor bills
* **Financial Calculators** – Tools like SIP and EMI calculators for personalized planning
* **Chatbot** – A conversational assistant for basic financial guidance
* **Stock Prediction** – A form to enter ticker symbols and receive predictions
* **News Headlines** – Displays relevant stock news items for a given ticker

Each module is designed to be intuitive with minimal user effort.

#### ****4.1.3 Database Design****

The system uses **MongoDB** to manage persistent user data. It includes:

* **User Collection**:  
  Stores user login information with passwords hashed using **SHA**.

No other collections or complex relationships exist in the current version.

#### ****4.1.4 Security Measures****

* **Password Security**:  
  FINWISE uses the **SHA hashing algorithm** to securely store user passwords in MongoDB.
* **Data Access**:  
  User-specific data such as expenses is protected so that only authenticated users can access their data.
* **Data in Transit**:  
  Though the app runs on localhost, form data is submitted through secure Flask endpoints following basic input validation.

**4.1.5 Performance and Scalability**

• **Performance:**

The proposed system leverages a client-server architecture with Flask and Python, ensuring quick response times for most modules such as calculators, chatbot interactions, and expense tracking. However, the stock price prediction module, powered by the LSTM model, may take up to 150 seconds to complete due to the complexity of model execution over 30 epochs. This performance trade-off is necessary for generating accurate predictions, but future optimizations could reduce processing time.

• **Scalability:**

The current design of FINWISE is optimized for local use, operating in a single-user environment via localhost. As the system relies on local data storage with MongoDB, it is not yet scalable to support multi-user or distributed access. Future enhancements will include cloud integration and multi-user capabilities to enable synchronization across devices, improve accessibility, and support larger datasets, enhancing the system’s overall scalability.

**4.2 CLASS DIAGRAM**

The class diagram illustrates the major components of the FINWISE application, including classes such as User, Chatbot, StockPredictor, ExpenseTracker, and their relationships. It highlights the key attributes and methods that define the functionality and structure of each module.

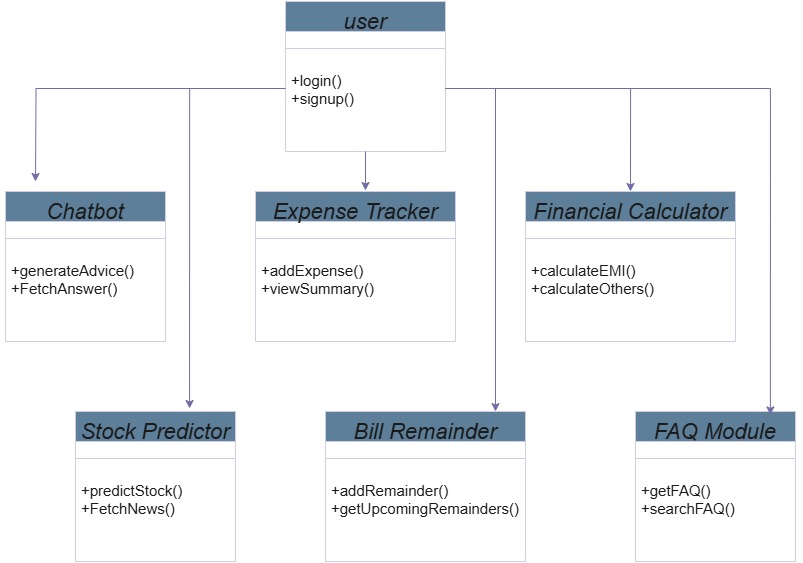
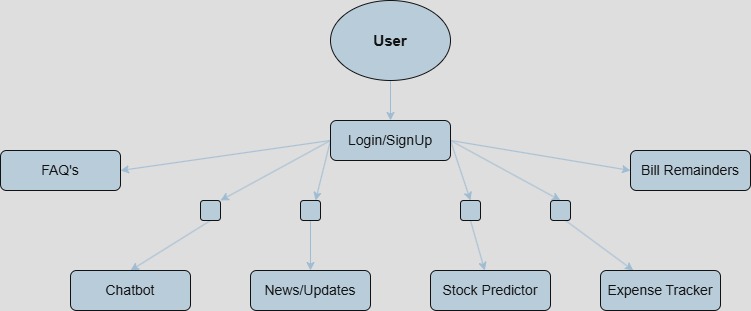
****

Figure 4.2

**4.3 USE CASE DIAGRAM**

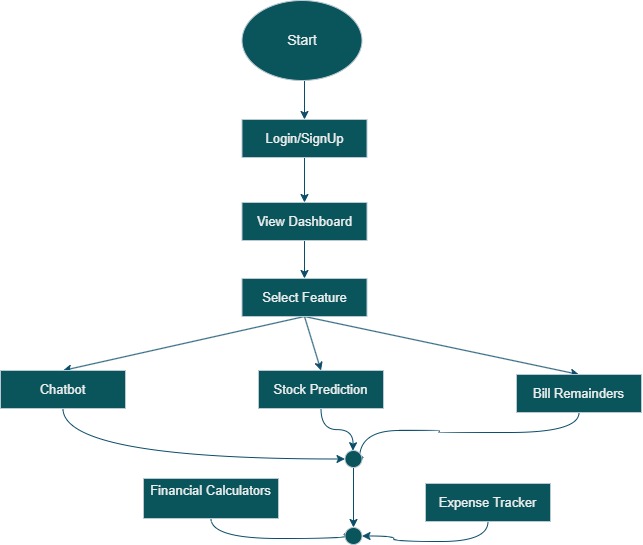
The use case diagram represents the various interactions a user can perform within the FINWISE application. It highlights core functionalities such as logging in, accessing the chatbot, performing stock price predictions, managing expenses, setting bill reminders, using financial calculators, and viewing FAQs. This diagram emphasizes user-centric activities without diving into internal system operations.

Figure 4.3

**4.4 ACTIVITY DIAGRAM**

The activity diagram shows the basic flow of user interaction in the FINWISE application. After login, the user can navigate to features like the chatbot, stock prediction, expense tracker, bill reminders, financial calculators, and FAQs. It gives a clear overview of user actions without going into technical details of each module.

Figure 4.4



**4.5 SEQUENTIAL DIAGRAM**

The sequence diagram illustrates the interaction between the user, frontend, backend, database, and machine learning model (LSTM) during typical operations. It demonstrates how the user initiates a request (such as logging in or using the chatbot), how the frontend forwards this to the backend, which in turn communicates with the database or the LSTM model for data processing. Finally, the results are returned back to the user through the frontend interface, ensuring smooth and responsive user experience.

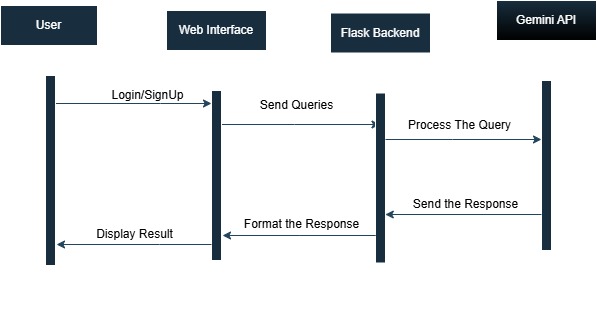
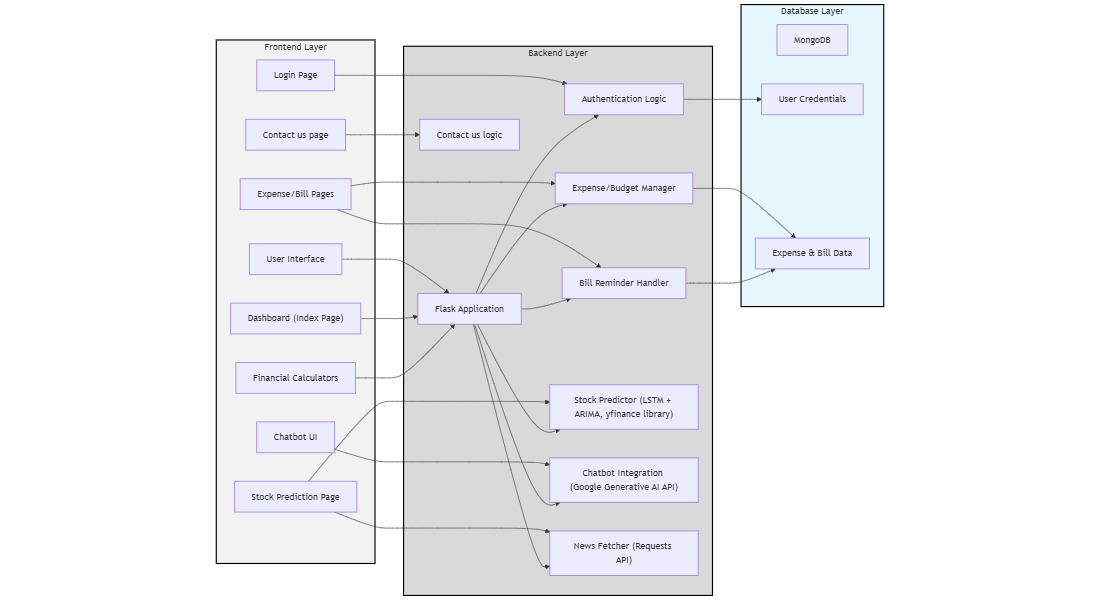
****

Figure 4.5

**4.6 SYSTEM ARCHITECTURE**

The system architecture of FINWISE follows a modular and layered approach, ensuring separation of concerns and ease of maintenance. The application is based on a client-server model, where the frontend interacts with the backend server via HTTP requests, and the backend communicates with the MongoDB database for data persistence. Each core feature — such as stock prediction, chatbot, expense tracking, and financial tools — is designed as a separate module that integrates with the central Flask-based backend.

4**.6.1 System Architecture Diagram**

### **Figure 4.6**

### ****4.6.2 Frontend Layer****

The frontend of FINWISE is developed using standard web technologies and is served locally via a web browser. It provides a simple and intuitive user interface that allows users to interact with features like stock prediction, expense tracking, chatbot, and financial calculators. The frontend communicates with the backend through HTTP requests.

### ****4.6.3 Backend Layer****

The backend is built using Flask (Python) and handles all the core functionalities, including request processing, user authentication, and integration of machine learning models (like LSTM for stock prediction). It acts as a bridge between the frontend and the database, ensuring secure and efficient data flow.

### ****4.6.4 Database Layer****

The application uses MongoDB to store user data, such as login credentials, expenses, and reminders. It provides a flexible and scalable NoSQL solution suited for storing structured and unstructured financial data efficiently.

### ****4.6.5 Security Layer****

FINWISE implements security at both backend and database levels. User passwords are hashed using the SHA-256 algorithm before being stored in MongoDB. This ensures that sensitive information remains secure, even in case of a data breach. Additional security considerations include secure API handling and planned enhancements like multi-factor authentication.

### ****4.6.6 Communication Flow****

The communication flow in FINWISE follows a standard client-server architecture. When a user initiates an action from the frontend (such as logging in, requesting stock predictions, or submitting expenses), a request is sent to the Flask backend via HTTP. The backend processes the request, interacts with the database if needed (e.g., to retrieve or store data), and returns the appropriate response.

This communication ensures smooth coordination between modules:

* **Frontend ↔ Backend:** All requests (login, chatbot queries, financial tools, etc.) are sent via API calls.
* **Backend ↔ Database:** Data such as user credentials, expense records, and reminders are fetched/stored securely in MongoDB.
* **Backend ↔ ML Models:** For features like stock prediction, the backend loads trained LSTM models, performs inference, and returns results to the frontend.

This modular communication flow ensures efficient and scalable interactions among all components.

**4.7 TECHNOLOGY DESCRIPTION**

The FINWISE application leverages a combination of modern web development technologies, machine learning frameworks, database systems, and security mechanism to deliver an intelligent, responsive, and user-centric platform for personal financial management and forecasting. This section elaborates on the primary technologies used in the development and deployment of the application.

### ****4.7.1 Frontend Technologies****

The frontend of FINWISE was developed with a focus on responsiveness, interactivity, and user-friendly design. It serves as the primary interface between users and the backend services.

* **HTML5 & CSS3**: These foundational web technologies were used for building the structure and style of the application. HTML5 supports semantic elements and media integration, while CSS3 ensures a visually pleasing and responsive layout across screen sizes.
* **JavaScript (ES6+)**: JavaScript was used to handle interactivity within the application, enabling dynamic content loading, client-side form validation, and asynchronous API communication.
* **Bootstrap**: A CSS framework that allowed rapid development of responsive UI components with built-in mobile-first support. It also provided consistent styling across all browsers.
* **Chart.js & Plotly.js**: These libraries were used to create interactive and visually appealing data visualizations for financial trends, stock predictions, and user expense summaries.
* **Font Awesome**: For modern and consistent iconography throughout the application, making the interface intuitive and visually clean.
* **Responsive Design**: Media queries and grid systems were used to ensure seamless experiences across desktops, tablets, and smartphones.

### ****4.7.2 Backend Technologies****

The backend was developed with scalability and modularity in mind, capable of handling data processing, API routing, and ML model interaction.

* **Python (3.8+)**: The core backend logic and machine learning components were written in Python. Its extensive library ecosystem supported rapid development and robust performance.
* **Flask**: A lightweight WSGI web application framework used to build the RESTful APIs that serve the frontend. It enabled fast development and easy integration with Python-based ML components.
* **TensorFlow & Keras**: Used for training and running the LSTM deep learning model responsible for stock market predictions. These frameworks offer tools for building, training, and evaluating time-series forecasting models.
* **Pandas & NumPy**: These libraries were employed for data manipulation and numerical analysis. Pandas made it easy to handle structured financial data, while NumPy enabled efficient matrix operations for ML inputs.
* **Matplotlib & Seaborn**: Used to visualize trends during model development and to generate optional static reports.
* **scikit-learn**: Integrated for ARIMA modeling, feature scaling, and various preprocessing steps for prediction tasks.
* **PyMongo**: This Python driver enabled seamless interaction with the MongoDB database, ensuring robust data CRUD operations.
* **Google Generative AI API:** Used to build the chatbot feature, that answers user’s queries on financial insights.
* **Requests:** To retrieve news on entered stock name for prediction.

### ****4.7.3 Database Technologies****

FINWISE uses a flexible, scalable NoSQL database to handle diverse types of structured and semi-structured data.

* **MongoDB**: A document-oriented NoSQL database used to store user credentials, transaction logs, expenses, reminders, and chatbot history.
* **JSON-based Storage**: MongoDB's JSON-like format made it easy to work with financial data and user preferences, allowing flexible schema evolution as the project grew.

### ****4.7.4 Security Technologies****

Security is integral to any financial application. In FINWISE, essential mechanisms were implemented to protect user data and application integrity.

* **SHA-256 Hashing**: User passwords are hashed using SHA-256 before being stored in the database. This one-way hashing algorithm ensures that plaintext passwords are never exposed.
* **Token Validation**: Session tokens were used for validating logged-in users and preventing unauthorized access to restricted features.
* **HTTPS (for Deployment)**: Ensured encrypted communication between client and server, especially during login and financial transactions.
* **Input Validation**: All user inputs were validated at both frontend and backend levels to prevent XSS, SQL/NoSQL injection, and malformed data entry.
* **Session Timeouts**: Users are automatically logged out after a period of inactivity to reduce the risk of unauthorized access.
* **No Sensitive Data in Logs**: Application logs excluded personal or financial identifiers to protect user privacy.

**5. IMPLEMENTATION AND TESTING**

This chapter outlines the development and testing phases of the FINWISE platform. The implementation focuses on utilizing a combination of web technologies and advanced programming techniques to build a seamless and efficient user experience. The testing phase ensures that each component of the platform functions correctly and performs optimally, delivering a high-quality and reliable financial management tool to users.

**5.1 IMPLEMENTATION**

The implementation of **FINWISE** was carried out using a combination of **HTML, CSS, JavaScript, and Python** to ensure a smooth and functional user experience. The application consists of multiple interconnected modules, each built with an optimal technology stack.

#### **Frontend Development (HTML, CSS, JavaScript)**

The frontend was designed using **HTML and CSS**, ensuring a **clean, responsive, and user-friendly interface**. JavaScript was used to enhance interactivity and manage essential client-side functionalities.

* **Bill Reminders**: Implemented using JavaScript to allow users to set and manage due dates directly from the web interface.
* **Expense Tracker**: A JavaScript-powered table that enables users to add, delete, and categorize expenses dynamically.
* **Financial Calculators**: JavaScript functions were used to handle **SIP, EMI, and FD calculations**, processing user inputs instantly without requiring backend processing.
* FAQS : Implemented using Java Script to clarify users doubts on financial insights

#### **Backend Development (Python)**

Python was used primarily for **data-driven functionalities** such as stock price predictions and financial news retrieval.

* Stock Price Prediction and Financial News Retrieval :
  + Implemented using LSTM and ARIMA models to analyze historical stock data and predict future trends. These models analyze **historical stock data** and provide users with **predicted price trends** based on market patterns.
  + Live financial news is fetched using the News API and presented in a categorized, readable format.
  + Both are integrated into the frontend with graphical and textual UI elements elements enhancing for user-friendly engagement
* Chatbot:
  + Integrated using GoogleGenerativeAI API to assist users with financial queries.
  + Provides informative, context-aware responses based on user input and accessible directly from homepage enhancing user support

#### **Data Management & Security**

* **MongoDB** was used to store **user credentials, stock data, and prediction results** for better data persistence.
* **SHA-256 hashing** was applied to securely store user login credentials, ensuring data privacy and security.

Each module was thoroughly tested individually before integration, ensuring smooth interaction between **frontend and backend components**. The following section will present implementation screenshots for each functionality of FINWISE.

**5.1.1 LOGIN PAGE**

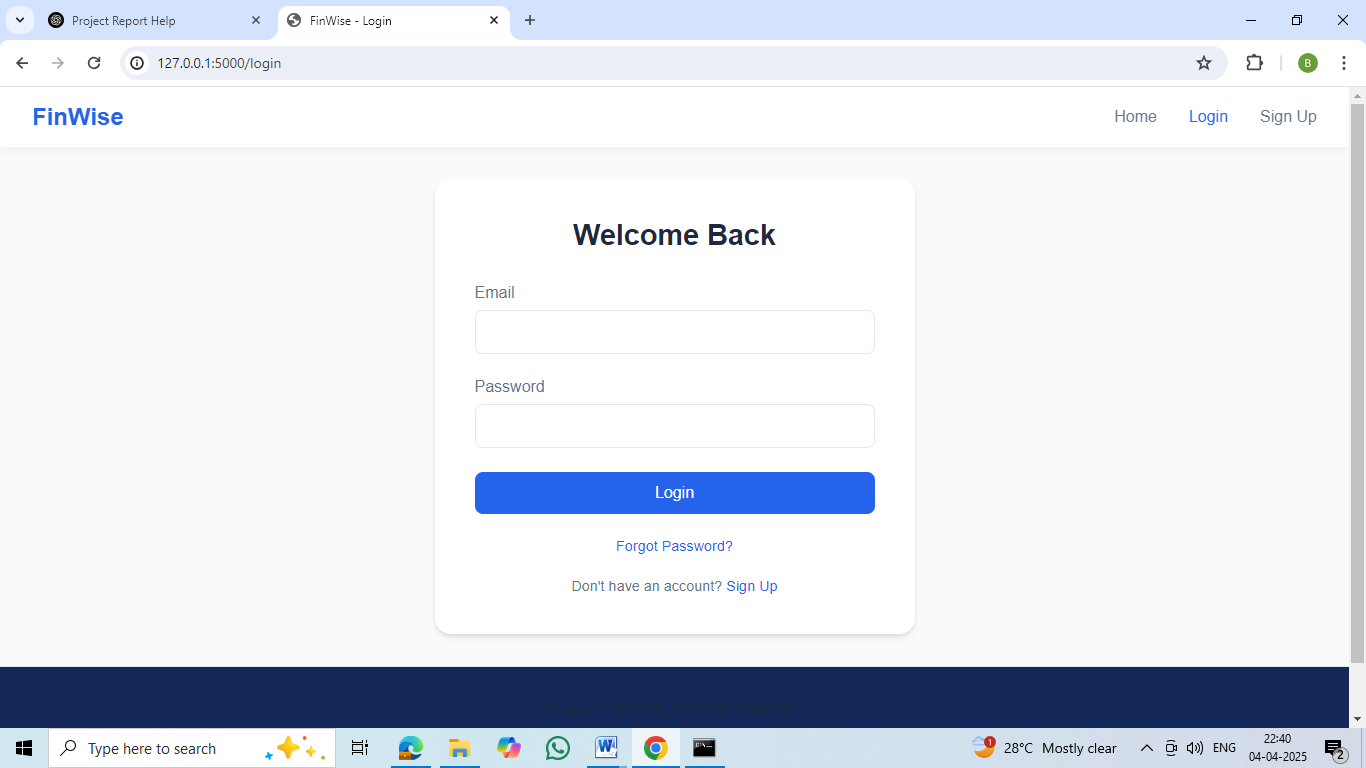
****

Figure 5.1.1

The login page allows users to securely access their FINWISE account by entering their email and password. It ensures that only authorized users can view their personalized financial data.

**5.1.2 INDEX PAGE**

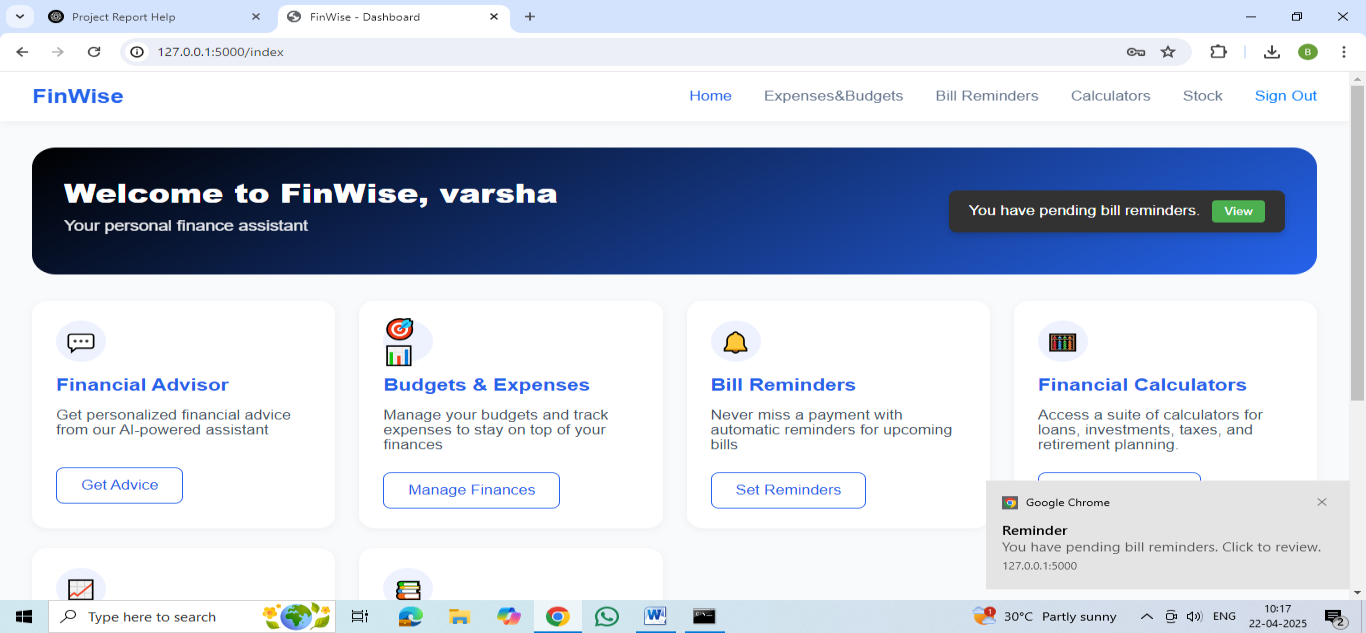


Figure 5.1.2

The index page serves as the main dashboard, providing users with easy access to all key features of FINWISE. From here, users can navigate to modules like bill reminders, chatbot,

Stock prediction, calculators and other

**5.1.3 CHATBOT PAGE**

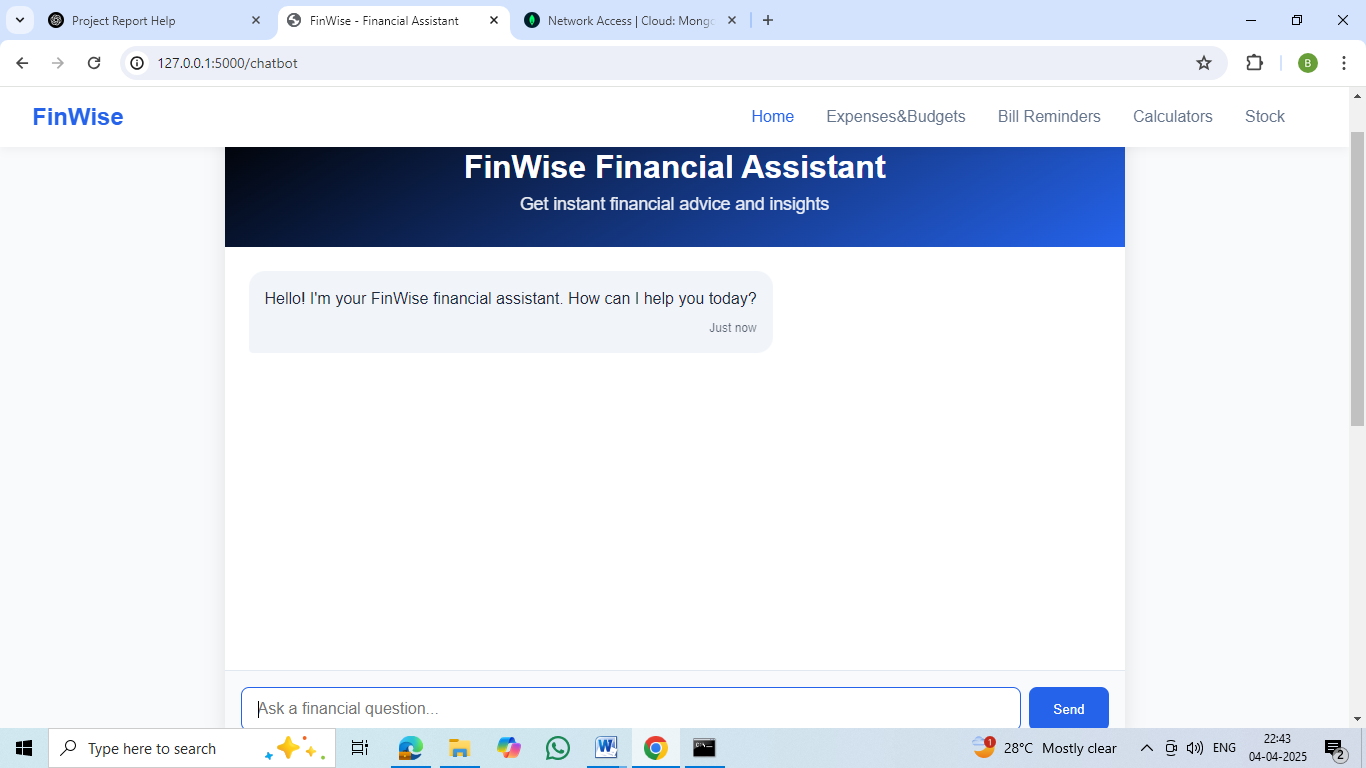
****

Figure 5.1.3

The chatbot provides instant, conversational financial guidance through advices

**5.1.4 BILL REMINDERS PAGE**

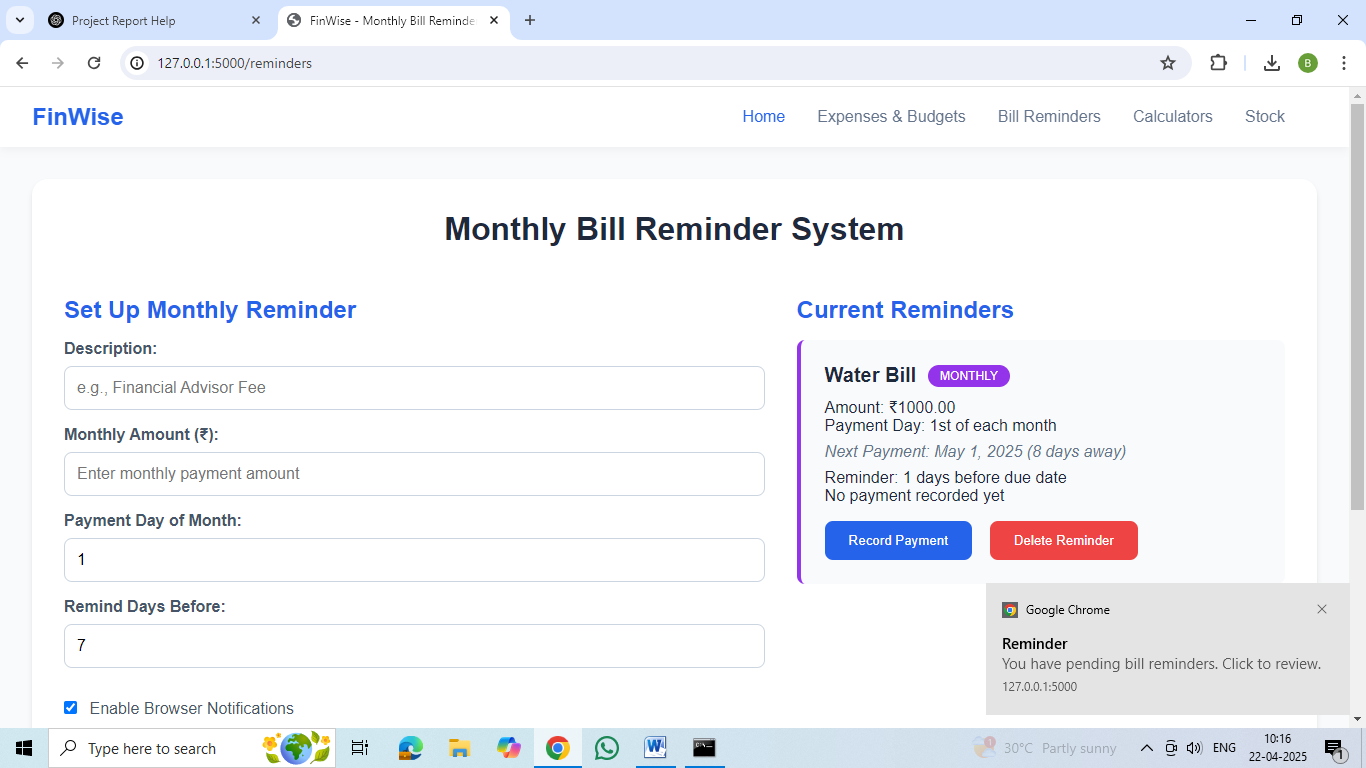


Figure 5.1.4

**Bill Reminders** in FINWISE notify users of upcoming payments through **in-app alerts and browser notifications** whenever they open the app.

**5.1.5 EXPENSE TRACKER PAGE**

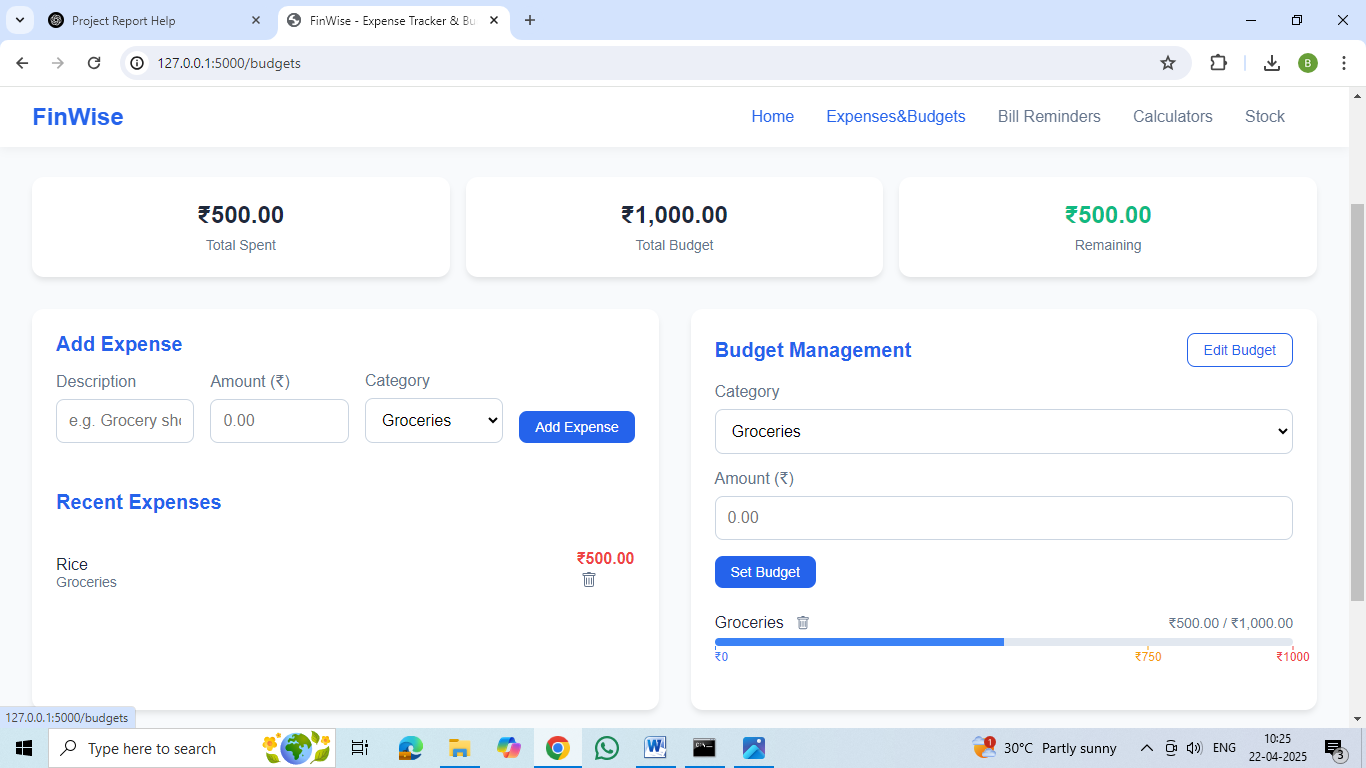


Figure 5.1.5

The **Expense Tracker** in FINWISE helps users monitor, categorize, and manage their spending for better financial control and budgeting.

**5.1.6 STOCK PRICE PREDICTION PAGE**

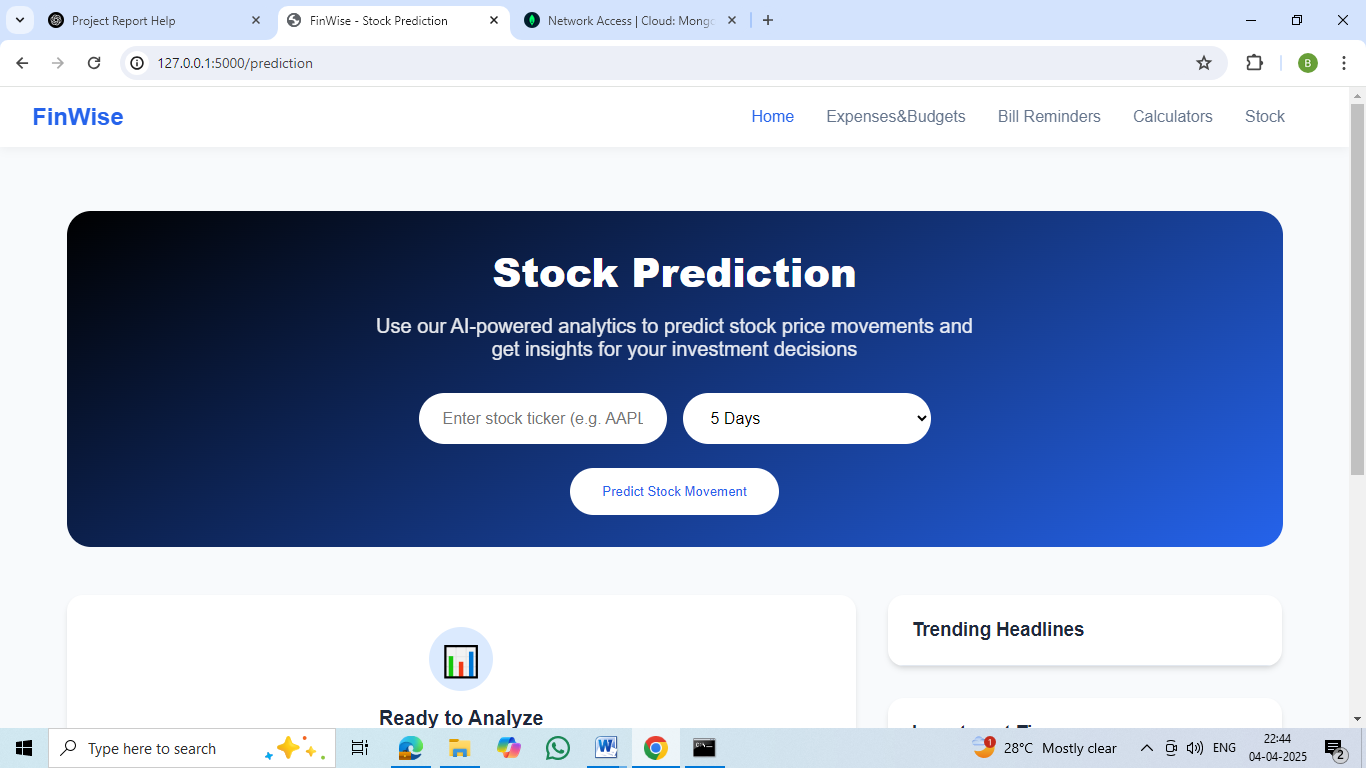
****

Figure 5.1.6

The stock prediction module enables users to input a company’s ticker symbol and receive price forecasts generated through a hybrid LSTM and ARIMA model, offering both accuracy and trend insights. It presents predicted values with upper and lower confidence limits to help users assess risk. Additionally, related news articles are displayed in the trending headlines section to provide market context.

Similarly, the remaining features displayed on the index page, such as the financial calculators, and FAQ section, are also implemented and integrated successfully as part of the application. The expense tracker allows users to log and categorize their spending, while the bill reminders help them stay on top of upcoming payments. Financial calculators assist in planning EMIs, SIPs, and other investments, and the FAQ section enhances financial literacy by providing clear answers to common personal finance questions—all contributing to a seamless and comprehensive user experience.

**5.2 TESTING**

The testing phase of the FINWISE application focused on ensuring functionality, usability, and performance across all modules. Various levels of testing were conducted to identify and eliminate bugs, improve user experience, and ensure smooth functioning of integrated services like prediction models, financial calculators, and API-based features.

#### ****5.2.1 Types of Testing Performed****

1. **Unit Testing**  
   Each module (e.g., chatbot, expense tracker, calculators) was tested independently using console logs and mock inputs to verify their expected output. JavaScript functions were checked for edge cases and boundary values.
2. **Integration Testing**  
   The integration between the Python backend (LSTM, ARIMA, Gemini API) and the frontend was tested to ensure data flow, prediction rendering, and dynamic content loading worked as intended.
3. **UI/UX Testing**  
   The application was tested on standard laptop screens to ensure that the user interface is clean, functional, and all buttons, forms, and outputs behave as expected. Emphasis was placed on intuitive navigation and consistent design elements across pages.
4. **Functional Testing**  
   Each feature was tested end-to-end to validate:
   * User login/signup
   * Chatbot responses
   * Stock prediction display
   * Financial calculator accuracy
   * Expense tracking data updates
   * Reminder alerts
   * News Requests
5. **Manual Testing**  
   Conducted by the team to simulate real-world usage by multiple users. Any bugs found were logged and resolved iteratively.

#### ****5.2.2 Tools and Techniques****

* **Browser Developer Tools** were used to monitor API responses, console logs, and layout behaviors.
* **Postman** was used to manually test Gemini API and validate JSON data.
* **JavaScript Console Logging** helped identify frontend logic issues during development.

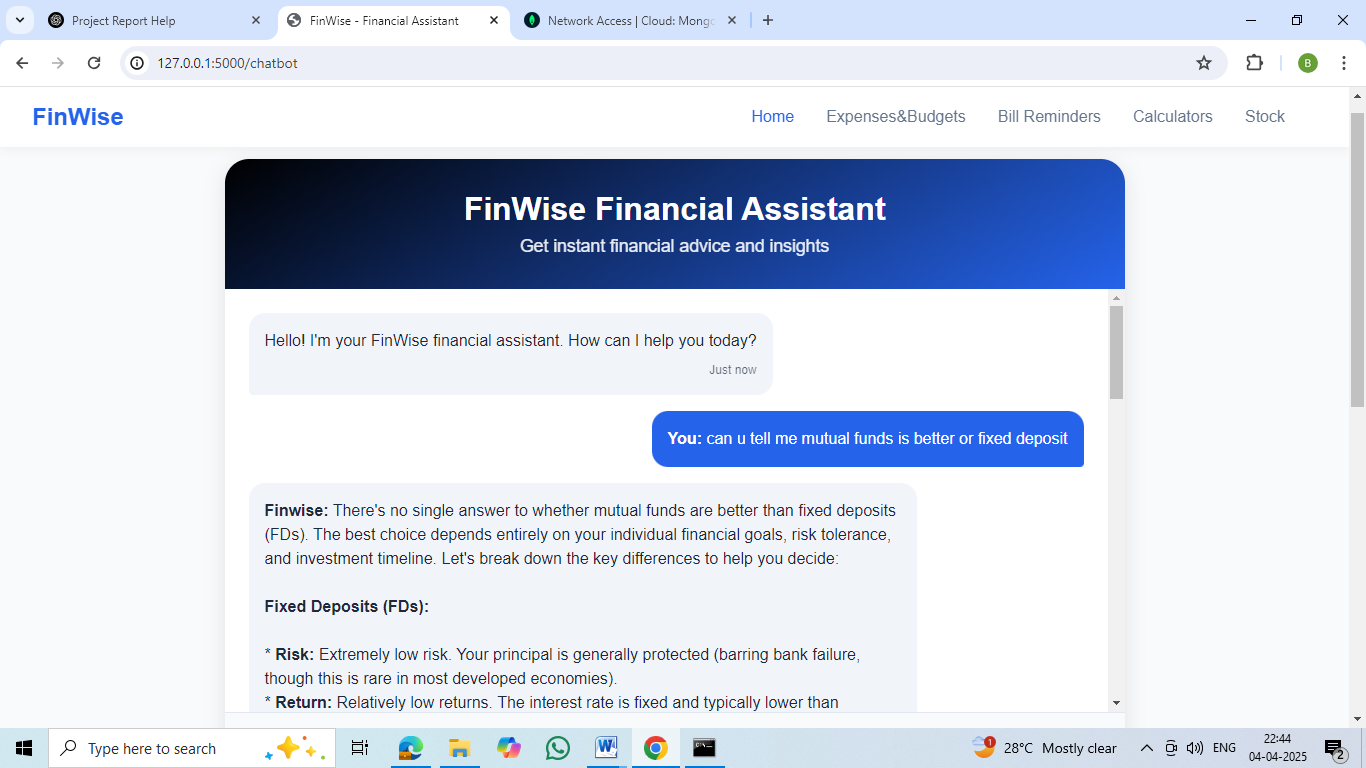
#### ****5.2.3 Results and Observations****

* All core features passed unit and integration tests successfully.
* Few minor layout issues on mobile were identified and resolved.
* Prediction results using LSTM and ARIMA showed consistency with mock data.

**5.2.4 Chatbot query results**

Figure 5.2.4

When asked whether a mutual fund is better than a fixed deposit, the chatbot provided a balanced comparison, outlining the risks and benefits of both options.



**5.2.5 Stock Price Prediction Results with News**

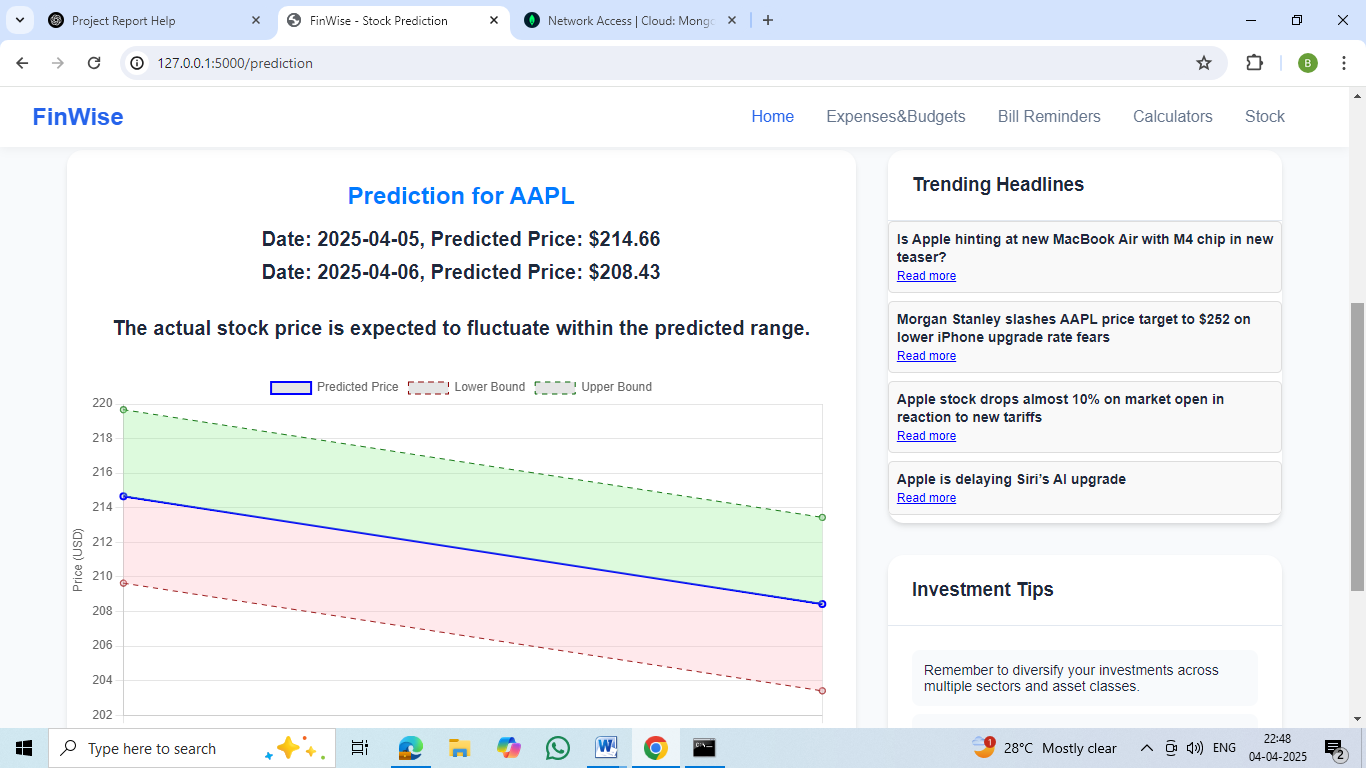


Figure 5.2.4

Upon entering the ticker symbol **AAPL**, the stock prediction module processes historical data using a hybrid **LSTM + ARIMA model** to forecast future price trends. The result displays the **predicted stock prices** along with **upper and lower confidence bounds**, allowing users to understand the possible price range and assess associated risk levels. The model captures both short-term fluctuations and long-term trends, providing a comprehensive view of expected market movement. Additionally, the **Trending Headlines** section is updated in real-time to show the latest news articles related to Apple Inc., offering users valuable market context. This combination of AI-powered prediction and live news helps users make more informed and data-driven investment decisions.

Similarly, other features such as the **expense tracker**, **bill reminders**, **financial calculators** are thoroughly tested to ensure they function smoothly. The expense tracker accurately categorizes and tracks spending, while the bill reminder system sends timely notifications for upcoming payments. The financial calculators provide precise calculations for EMIs, SIPs, and other financial planning tools. All these features are rigorously tested to maintain a seamless and reliable user experience.

**6. CONCLUSION AND FUTURE SCOPE**

**6.1 CONCLUSION**

The **FINWISE** project emerges as a unified digital solution tailored to simplify and optimize personal financial management. In a time where individuals are increasingly seeking control, visibility, and planning in their financial lives, FINWISE provides a one-stop platform that combines critical tools such as **expense tracking, bill reminders, stock market prediction, financial calculators, a smart chatbot, and curated financial news**—all integrated into an accessible web interface.

The core strength of the system lies in its **modular structure** and **intelligent backend processing**. With the help of machine learning algorithms like **LSTM and ARIMA**, the platform delivers near-accurate stock predictions, helping users make timely investment decisions. The **NEWS API integration** further enriches the user experience by delivering live, relevant financial news, enhancing awareness and informed action.

Unlike many single-feature finance tools, **FINWISE** is deliberately designed as a **multi-functional ecosystem**, allowing users to handle diverse financial tasks without navigating across multiple platforms. The **expense tracker** assists in daily budgeting and categorization of spends; the **bill reminder** system prevents missed deadlines by providing timely alerts, and **financial calculators** support instant estimations for loans, savings, and EMIs. The **AI-powered chatbot**, though rule-based, brings conversational convenience and serves as a guide through the application.

From a technological perspective, the project uses a **clean frontend** built with HTML, CSS, and JavaScript, while the backend is powered exclusively by **Python**, utilizing its robust libraries for data processing and model implementation. The data is handled securely using **MongoDB**, and sensitive user information is hashed using **SHA-256**, ensuring a basic layer of data protection.

Overall, FINWISE has been developed with the goal of **bridging the gap between users and financial literacy**, promoting awareness, responsibility, and ease in handling money-related decisions. Its design is not only focused on functionality but also on **user-friendliness, responsiveness, and personalization**, making it suitable for a wide range of audiences, especially students and young professionals who are beginning to manage their finances independently.

### ****6.2 FUTURE SCOPE****

While **FINWISE** currently integrates several essential financial tools in a unified platform, there remains vast potential for future growth and feature enhancement. With evolving financial trends, user expectations, and advancements in technology, the following enhancements can significantly elevate the platform's capabilities:

#### ****6.2.1 Cloud Sync and Cross-Platform Access****

* Implementing cloud synchronization would allow users to access their financial data seamlessly across multiple devices.
* A cross-platform experience, including a mobile application, would enhance accessibility, making FINWISE more scalable and widely adoptable.

#### ****6.2.2 Multi-Currency and Global Support****

* Expanding support for multiple currencies will enable users to track international transactions and conversions.
* Integrating global financial market data can help users make informed investment decisions regardless of their location.

#### ****6.2.3 Real-Time**** In-app alerts ****and Smart Reminders****

* Instant alerts for important financial activities, such as bill due dates, budget thresholds, or suspicious transactions, can improve financial awareness.
* AI-driven smart reminders can analyze spending habits and suggest optimized payment schedules to enhance financial planning.

#### ****6.2.4 Data Security and Privacy Enhancements****

* Strengthening encryption methods and integrating additional authentication layers, such as biometric authentication, can further safeguard user data.
* Compliance with evolving data protection regulations will ensure user privacy and build trust in the application’s security framework.

#### ****6.2.5 API Integration for Real-Time Banking****

* Connecting with banking APIs will enable users to view real-time balances, perform transactions, and receive live updates on financial activities within FINWISE.
* Secure API integrations with third-party financial services can offer enhanced capabilities, such as automated savings plans and investment tracking.

**7. REFERENCES**

1. Brownlee, J. (2022). Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python. Machine Learning Mastery.
2. Zhang, C., & Li, Y. (2021). "Hybrid LSTM-ARIMA Model for Stock Prediction: A Comparative Study." Journal of Computational Finance and Data Science, 9(4), 112–120.
3. National Institute of Standards and Technology. (2023). Secure Hash Standard (SHS) – FIPS PUB 180-4. <https://nvlpubs.nist.gov>
4. Jain, R., & Agarwal, M. (2023). "AI-Powered Financial Assistants: The New Era of Personal Finance Management." IEEE Access, 11, 32045–32059.
5. Gensler, G. (2021). Speech on the Future of Fintech and Digital Finance. U.S. Securities and Exchange Commission (SEC). [www.sec.gov](https://www.sec.gov)
6. Mathew, J. (2022). Modern Web Security Essentials: Protecting Web Applications with Flask and FastAPI. O’Reilly Media.
7. Google Cloud. (2024). Generative AI on Vertex AI: Developer Guide. https://cloud.google.com/generative-ai
8. Zacks Investment Research. (2023). Introduction to Stock Data APIs using YFinance. <https://www.zacks.com>
9. StackPath. (2022). NewsAPI Developer Docs: Real-time News Headlines for Developers. https://newsapi.org/docs
10. Krug, S. (2020). Don’t Make Me Think, Revisited: A Common Sense Approach to Web Usability (3rd ed.). New Riders.