AI Based Financial Advisor

Mandar Mali[1], Sagar Gundwade[2], Arif Shaikh[3] Prof. N. S. Hunnargi.

Department of Electronics & Telecommunication Engineering, ATS's Sanjay Bhokare Group of Institutes, Miraj

---***---

Abstract - This paper presents the development of an AI-powered financial advisory system designed to provide personalized investment recommendations. The platform collects user-specific financial data such as salary, expenses, savings, profession, city, risk appetite, and investment horizon. It employs machine learning models, including Random Forest Regressor and Multi Output Regressor, along with a rule-based logic engine to generate asset allocation suggestions across equity, debt, gold, and fixed deposits. The web application is built using FastAPI and features secure authentication, an admin dashboard, a chatbot interface, and real-time stock data integration. The system was tested using both functional and model validation methods to ensure accuracy and user-friendliness. Results indicate that the platform offers reliable and personalized financial recommendations, making it a cost-effective and scalable alternative to traditional advisory services.

1. INTRODUCTION

In the current digital era, managing personal finances has become increasingly important yet challenging for individuals lacking financial literacy or access to professional advisors. The growing complexity of investment options—such as equities, fixed deposits, debt instruments, and gold—requires intelligent decision-making tools that are both accessible and personalized. Traditional financial advisory services often come with high costs and are not easily scalable to serve a broader population.

Artificial Intelligence (AI) and Machine Learning (ML) offer significant potential in democratizing financial planning through intelligent automation and data-driven recommendations. This paper presents an AI-powered financial investment advisor designed to collect user-specific data and provide personalized portfolio suggestions using ML algorithms and rule-based logic. The system also integrates a chatbot for user interaction, secure login, and real-time stock market data.

Developed using Python's FastAPI framework and Scikit-learn for model implementation, the application aims to simplify the investment process for individuals from various economic backgrounds. By offering an intuitive web-based platform with modular design, the system delivers scalable and efficient financial guidance.

This paper outlines the technical framework, data collection methods, model selection strategies, and system architecture of the proposed financial advisor. It also discusses the testing methodologies and results, demonstrating how AI can enhance financial decision-making and accessibility.

2. LITERATURE SURVEY

Patil and Kulkarni et.al in [1] investigated the adoption of AI-driven chatbot advisors in the financial services sector, particularly focusing on Indian consumers. Their research, based on the Technology Acceptance Model (TAM), extended traditional constructs by integrating perceived risk, privacy, enjoyment, social influence, and perceived strength of control. Using a survey-based approach with 310 respondents and analyzing data via PLS-SEM, the study concluded that perceived usefulness, ease of use, enjoyment, and social influence positively influence chatbot adoption, while perceived risk negatively affects it. The study emphasized that while chatbots enhance customer experience through instant, 24/7 interaction, security and trust remain crucial concerns for users, especially in handling sensitive financial data. Furthermore, their findings highlighted that AI chatbot advisors can significantly enhance customer engagement, reduce operational costs, and provide scalable solutions for financial institutions, although human advisors still have an edge in complex decision-making contexts. Their work provides a foundational empirical base for understanding consumer behavior towards AI-based financial advisory systems in emerging economies like India.

Olivia Brown et.al in [2] presents a comprehensive exploration of the role of AI-powered chatbots and virtual assistants in the transformation of the financial services industry. Her study synthesizes findings from global consulting firms and institutional case studies, demonstrating how AI has become pivotal in automating customer service, providing financial advice, and improving fraud detection. According to the research, financial institutions like Bank of America, JPMorgan Chase, and HSBC have leveraged virtual assistants to enhance operational efficiency, reduce costs, and deliver real-time, personalized customer support. Brown emphasizes that natural language processing (NLP) and predictive analytics are central technologies enabling chatbots to analyze customer behavior and market data for tailored financial solutions. However, the study also raises concerns about the limitations of chatbots in handling complex queries, and underlines the ethical and regulatory challenges associated with data privacy and algorithmic bias. Notably, the paper points toward the growing integration of generative AI models like GPT for more sophisticated and human-like interactions in finance. The findings support the conclusion that AI-powered chatbots are set to play an increasingly strategic role in financial decision-making and customer engagement, while also calling for robust governance and ethical frameworks to manage their growing influence.

Vitaliy M. Kobets and Kyrylo H. Kozlovskyi et.al in [3] explored the development and application of personalized financial advisory chatbots, emphasizing the significance of anthropomorphic design in enhancing user trust and engagement. Their study investigated how design elements such as human-like names, avatars, and interaction patterns can increase the sense of social presence and thereby influence users to trust and follow financial recommendations made by chatbots. They developed a Telegram-based financial advisory bot that leverages Python, Aiogram, Docker, and AWS to offer real-time investment advice, considering user income, inflation, and consumption goals. Notably, the bot applies a mathematical model to calculate optimal saving plans to achieve stable long-term consumption. Their work not only addresses the technological framework but also evaluates user psychology through controlled experiments, showing that trust significantly impacts the likelihood of following bot-generated advice. This research highlights the growing role of intelligent financial bots in democratizing financial planning, particularly for users with limited access to traditional financial advisors, and sets a benchmark for integrating human-computer interaction principles into robo-advisory systems.

3. METHODOLOGY

Requirement Analysis

The AI-Based Financial Advisor system was designed to provide secure, personalized, and intelligent investment recommendations through a modular and scalable web application. The first step in its development involved analyzing user roles, expected functionalities, and selecting appropriate technologies.

The system targets two main user groups: regular users and administrators. Regular users interact with the system via a chatbot interface to input financial details such as salary, expenses, savings, profession, and investment goals. Administrators monitor system activity and manage user data via a dedicated dashboard.

Key requirements identified included:

Secure user authentication and role management A chatbot-based interface for guided financial input Integration of real-time stock market data Machine learning-driven investment prediction Administrative control panel for data oversight

To meet these requirements, the following technologies and tools were utilized:

Python: The core development language, selected for it s simplicity, support for asynchronous operations, and robust ecosystem for machine learning and API integration.

FastAPI: A high-performance web framework that supports rapid API development, automatic documentation (Swagger UI), and strong type validation through Pydantic.
Scikit-learn: Used for implementing and training machine learning models such as Random Forest Regressor and MultiOutput Regressor. StandardScaler and LabelEncoder were used for data preprocessing.
HTML, CSS, JavaScript: Standard web technologies used for building a responsive and interactive frontend interface.
SQLAlchemy with SQLite/PostgreSQL: An Object-Relational Mapping (ORM) library used to manage the database, ensuring portability and ease of development.
OAuth2 and JWT (JSON Web Tokens): Implemented for secure authentication and session management.
Passlib was used to hash and store user credentials securely.
External APIs: Real-time stock data was integrated using services such as Yahoo Finance or Alpha Vantage to align financial recommendations with current market conditions. Pickle
Model Storage: Trained models were stored as .pkl files and managed by service layers to enable dynamic model selection and inference.

The chosen architecture and technology stack ensure the application remains extensible, secure, and adaptable to future enhancements such as mobile integration and expanded financial product coverage.

International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 04 Issue: 06 | June – 2025 DOI: 10.55041/ISJEM04684
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

SYSTEM DESIGN

The architecture of the AI-powered financial advisor is designed in a modular fashion for scalability and maintainability. As shown in Fig. 3.1, the application comprises multiple independent modules including authentication, user management, investment prediction engine, and admin analytics.

The backend system is implemented using FastAPI, a high-performance web framework based on Python. The modules are as follows:
main.py acts as the primary entry point for routing requests. auth.py
manages authentication and JWT (JSON Web Token) issuance.
user.py and admin.py define endpoints and logic for user and admin roles respectively.crud.py handles Create, Read, Update, and Delete (CRUD) operations on the database.

ml_models directory contains serialized machine learning models used for portfolio recommendations.

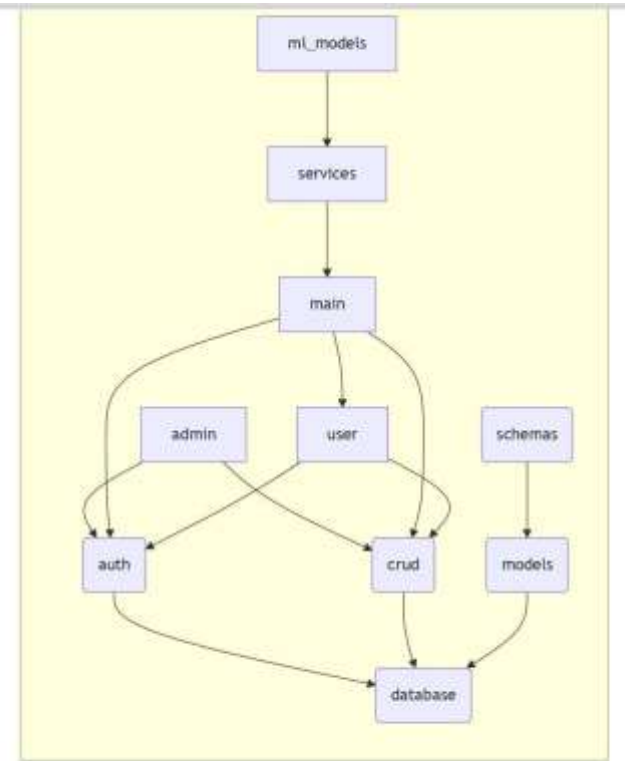

Fig.3.1. System Architecture

Admin Module Design

The administrative module plays a critical role in managing, monitoring, and maintaining the integrity of the AI-Based Financial Advisor system. It is responsible for overseeing user activity, accessing prediction histories, managing database records, and ensuring that the system performs reliably and securely.

The admin interface is implemented using FastAPI, with dedicated routing defined in the admin.py module. All administrative actions such as viewing registered users, accessing financial input logs, and validating prediction outputs are routed through this module. The admin panel is secured via role-based authentication using JWT (JSON Web Tokens) and OAuth2 protocols, ensuring that only authorized personnel can access sensitive functionalities.

Database operations performed by the admin, such as viewing and filtering user records or system logs, are abstracted through the crud.py module. This ensures a clean separation between interface logic and backend data operations. The admin credentials and roles are securely managed and validated using the auth.py module, which verifies each token before granting access to protected routes.

The architecture of the admin module is illustrated in Fig. 3.2, which shows the interaction between components including the API endpoints, authentication logic, and database management layers.

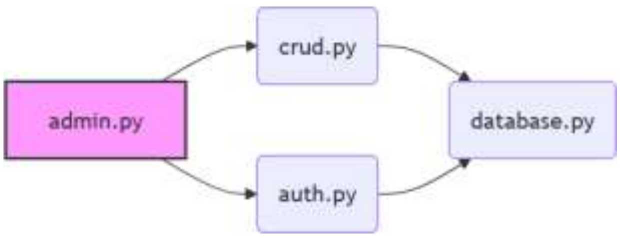

Fig.3.2. Admin side Architecture

This module also offers system-level visibility, allowing administrators to:
Track the number of registered users and their last login status.
View past prediction logs for validation or system audit.
Export user data for analysis or improvement of the ML models.
Monitor chatbot conversation logs to evaluate user experience. By
centralizing control, the admin module enhances data governance, supports analytics, and ensures that the financial advisor system remains compliant, efficient, and secure.

User Module Design

The user module is the core component of the AI-Based Financial Advisor platform, designed to provide an interactive and personalized experience for regular users. It is responsible for collecting user data, enabling secure access to services, guiding users via chatbot, and generating investment recommendations through machine learning (ML) models or rule-based logic.

Each user interaction begins with a secure login, authenticated via JWT tokens and handled through the auth.py module. Upon successful authentication, the user gains access to the dashboard where they can input financial details through a guided chatbot interface. These details typically include income, expenses, savings, profession, city, investment goals, risk appetite, and investment duration.

The chatbot interface, implemented using HTML, JavaScript, and FastAPI backend services, ensures a conversational and user-friendly flow. Inputs are collected in real-time and passed to the appropriate prediction engine via the user.py route handler. The selected model (base, enhanced, or rule-based) then processes the input and returns the asset allocation output.As illustrated in Fig. 3.3, the user module interacts with multiple components of the system:

main.py routes HTTP requests to the appropriate module (user or admin).
user.py handles form submissions, chatbot queries, and dashboard updates.
crud.py performs backend data operations such as saving user entries or fetching past records.
database.py establishes and maintains a connection with the SQL database using SQLAlchemy.
auth.py ensures session validation for all user-level protected actions.

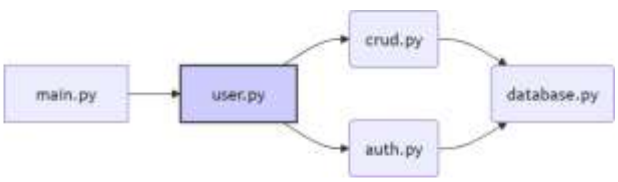

Fig.3.3. User Side Architecture

The user module also includes:

Real-time access to stock prices through integrated APIs
Dynamic recommendation rendering with visual trend indicators
Storage of user history for future reference and administrative review
This modular structure ensures that each user receives accurate, consistent, and personalized financial advice in a secure and intuitive environment.

Model Development and Implementation

This section clearly explains how your models work, how they're integrated, and what roles they serve.

Model Development and Implementation

The investment recommendation system is powered by three independent approaches: a base machine learning model, an enhanced model with demographic features**,** and a rule-based expert system**.** Each model is designed to predict an optimal allocation of a user's funds across four asset classes—equity, debt, gold, and fixed deposits (FD)—based on financial inputs and risk appetite.

Base Model: Random Forest Regressor

The base model uses a Random Forest Regressor**,** an ensemble machine learning algorithm known for its robustness and interpretability. It is trained on structured financial data including income, expenses, savings, investment horizon, and risk profile. This model was selected due to its ability to handle non-linear relationships and provide stable outputs across varied data inputs.

The base model predicts each asset class independently and is saved as a .pkl file using Python's Pickle module. At runtime, the model is loaded into memory and used to compute user-specific asset allocation.

Enhanced Model: Multi Output Regressor with Demographics

To improve prediction accuracy, an enhanced model was developed using a Multi Output Regressor architecture. It wraps the Random Forest Regressor to predict multiple targets (i.e., equity, debt, gold, FD) simultaneously.

Unlike the base model, this version includes additional demographic inputs such as profession and city, allowing the model to account for lifestyle differences and regional economic factors. These features are encoded using Label Encoder and normalized via Standard Scaler before training.

Rule-Based Recommendation Model

The rule-based model serves users who prefer deterministic or risk-specific outputs. It uses pre-defin ed static rules based on the user's risk profile (low, medium, or high). For example:

Low risk**:** Higher allocation to FDs and debt instruments

Medium risk**:** Balanced mix across equity, debt, and gold

High risk: Aggressive allocation toward equity and gold

This model is especially useful when limited user data is available, or if the ML model fails to provide a confident prediction

Model Storage and Runtime Inference

All models are saved as .pkl files and loaded dynamically using the chatbot_service.py logic layer.

This service determines which model to use based on available user inputs and interaction context. The system also supports offline analytics by exporting processed input-output data to CSV format via a data service module.

The modular structure ensures flexibility in updating models, switching logic engines, or integrating additional algorithms in the future. Model inference latency was optimized for sub-second response times, ensuring a smooth user experience.

Results and Discussion The AI-Based Financial Advisor system was tested to validate both functional accuracy and model reliability. Each module—authentication, chatbot interface, data collection, model prediction, and stock data integration—was evaluated individually and as part of the integrated platform.

5.1 Functional Testing

Key features such as user registration, login, data input via chatbot, and secure access control using JWT were tested across multiple user accounts. The admin module successfully retrieved system logs, user profiles, and prediction records. All functionalities performed as expected under standard load conditions.

5.2 Model Output Validation

The machine learning models (base and enhanced) were tested using a variety of input scenarios. The enhanced model consistently provided more accurate and context - aware asset allocation due to the inclusion of profession and city. The rule-based system also performed as expected, delivering consistent outputs aligned with predefined risk tiers.

Real-Time Data Integration

The integration of external APIs for stock data allowed users to view live prices and align their investment decisions with current market trends. API endpoints responded within acceptable latency limits, ensuring a responsive user experience.

Overall, the system met its intended goals of delivering secure, intelligent, and personalized financial advice through a web-based interface.

ISSN: 2583-6129
DOI: 10.55041/ISJEM04684

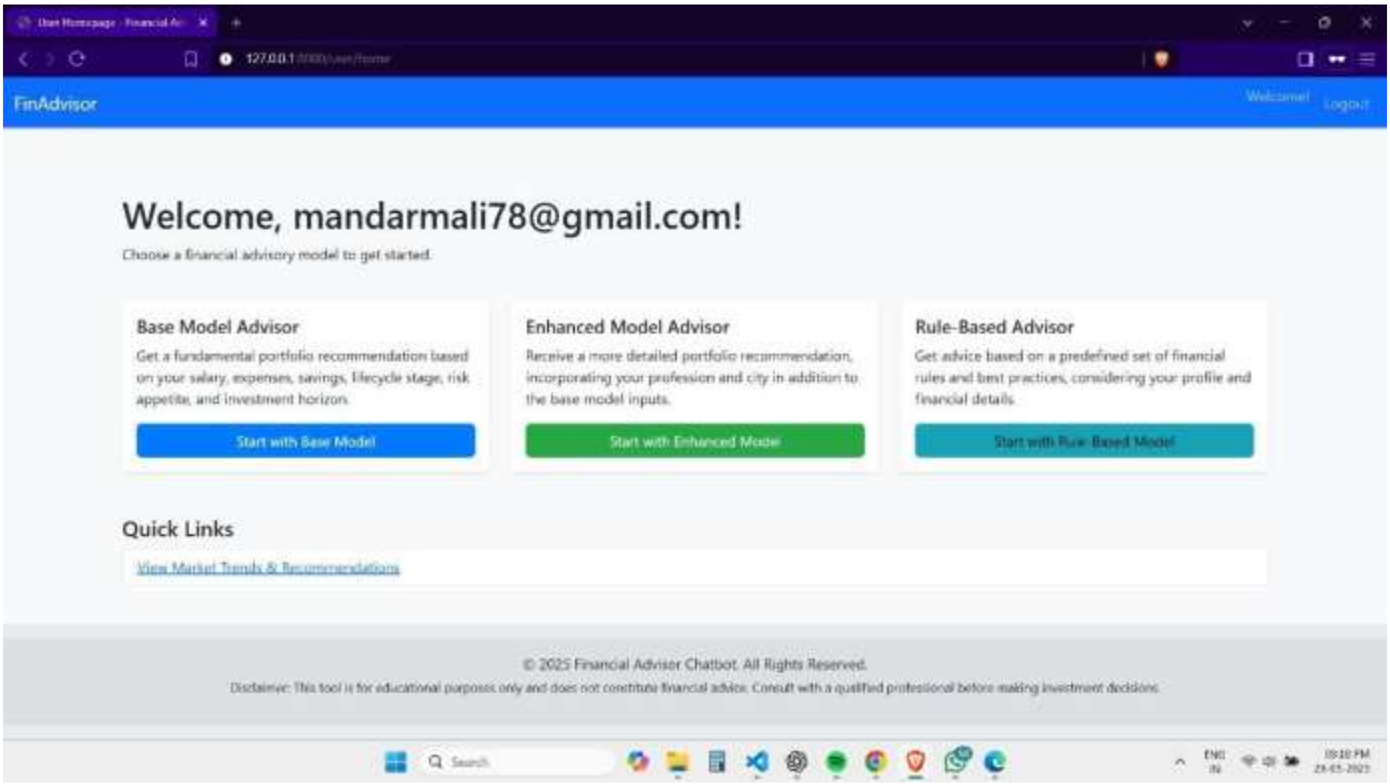

Fig.3.4. Homepage

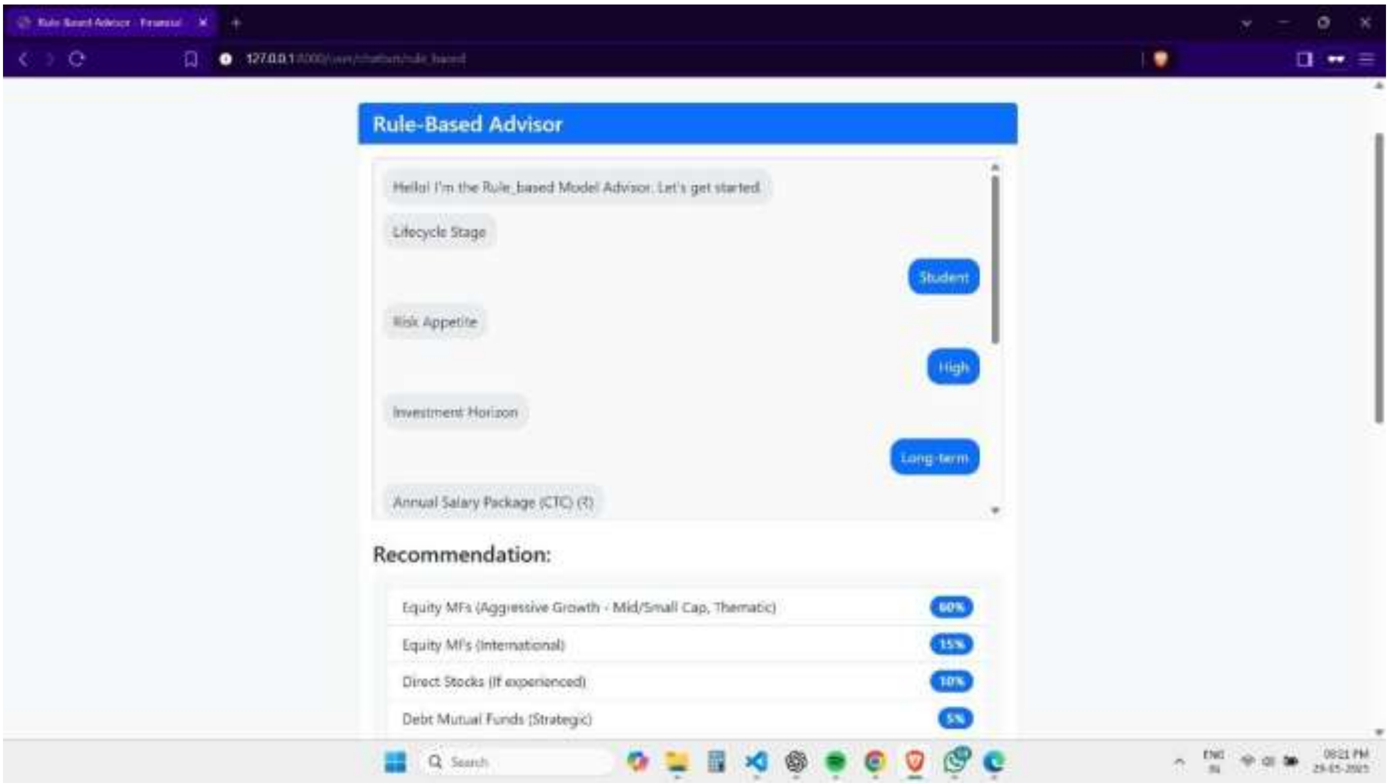

Fig.3.4. Rule based model interface

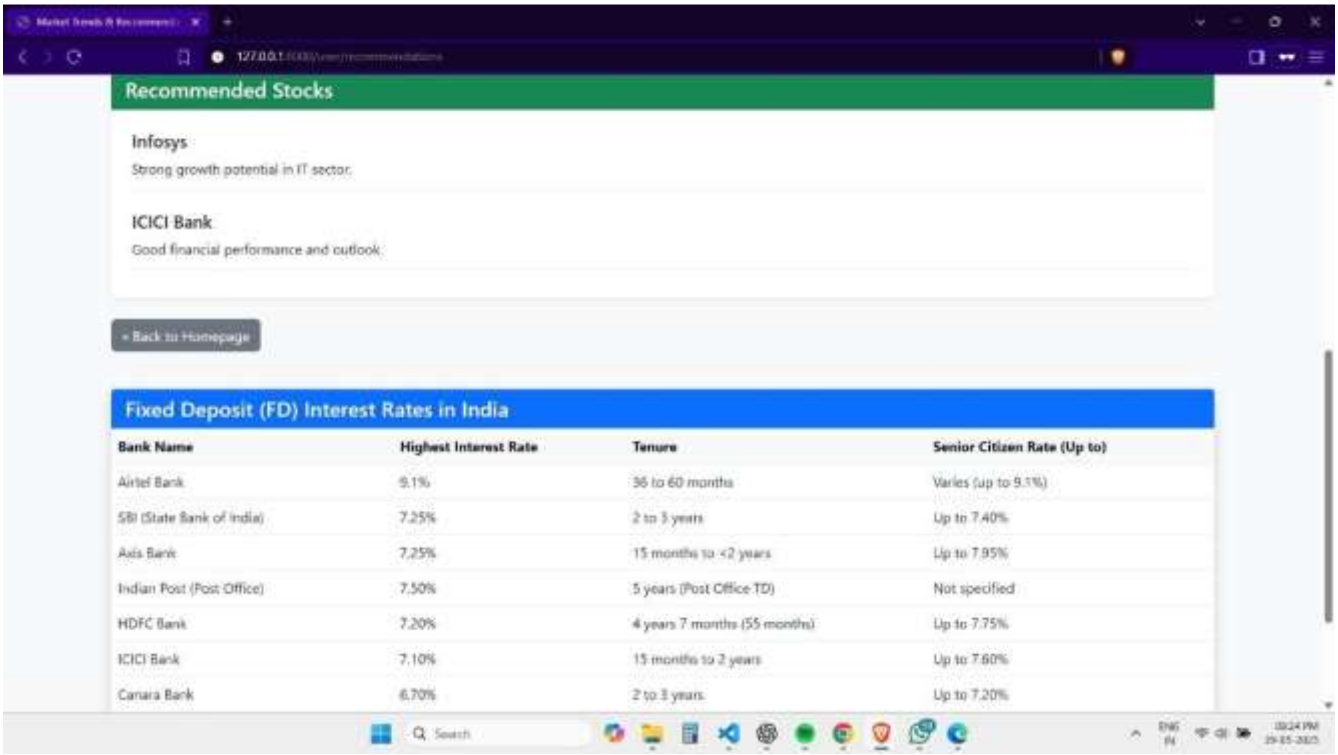

Fig.3.6. Recommendations

4. CONCLUSION

This paper presented the design and development of an AI-Based Financial Advisor aimed at democratizing access to intelligent investment recommendations. By combining machine learning models with a rule-based logic engine, the system effectively analyzes user-specific financial and demographic data to provide personalized asset allocation advice across equity, debt, gold, and fixed deposits.

The platform was implemented using Python, FastAPI, and scikit-learn, with secure user authentication via OAuth2 and JWT. The system supports interactive data input through a chatbot interface, a real-time stock information dashboard, and a role-specific admin panel. Model predictions were validated across various test scenarios, confirming the system's ability to deliver reliable and context-sensitive recommendations.

The modular architecture ensures scalability and flexibility for future enhancements, while the integration of real-time APIs improves the decision-making process for end users. With its accessible design and data-driven approach, the proposed system offers a cost-effective alternative to traditional financial advisory services.

This work demonstrates how artificial intelligence (AI) and modern web technologies can be combined to address practical problems in the financial domain, particularly for individuals with limited access to professional financial guidance.

ACKNOWLEDGMENT

The authors express their sincere gratitude to Prof. Mrs. N. S. Hunnargi, Project Guide, Department of Electronics and Telecommunication Engineering, Sanjay Bhokare Group of Institutes, Miraj, for his invaluable guidance, continuous support, and encouragement throughout the course of this project.

The authors also extend their thanks to Prof. Mr. M. N. Potdar, Head of Department, and Dr. A. C. Bhagali, Director of the institute, for providing the necessary infrastructure and a conducive research environment.

Special appreciation is due to Mr. Sachin Rokade and Mr. Vinayak Kurane for their technical assistance and insights during the implementation phase. Finally, the authors are thankful to their peers and all those who directly or indirectly contributed to the successful completion of this research work.

REFERENCES

[1] Kanchan Patil, Mugdha S Kulkarni Artificial Intelligence in Financial Services: Customer Chatbot Advisor Adoption" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-1, November 2019.

[2]] Olivia Brown The Future of Financial Services: AI-Powered Chatbots and Virtual Assistants Date:October 2024.

[3] Vitaliy M. Kobets, Kyrylo H. Kozlovskyi Application of chat bots for personalized financial advice DOI: https://doi.org/10.15276/hait.05.2022.18 UDC004.67:004.891.2:330.322.