Interactive Financial Tools

Stock Market Forecasting Web App

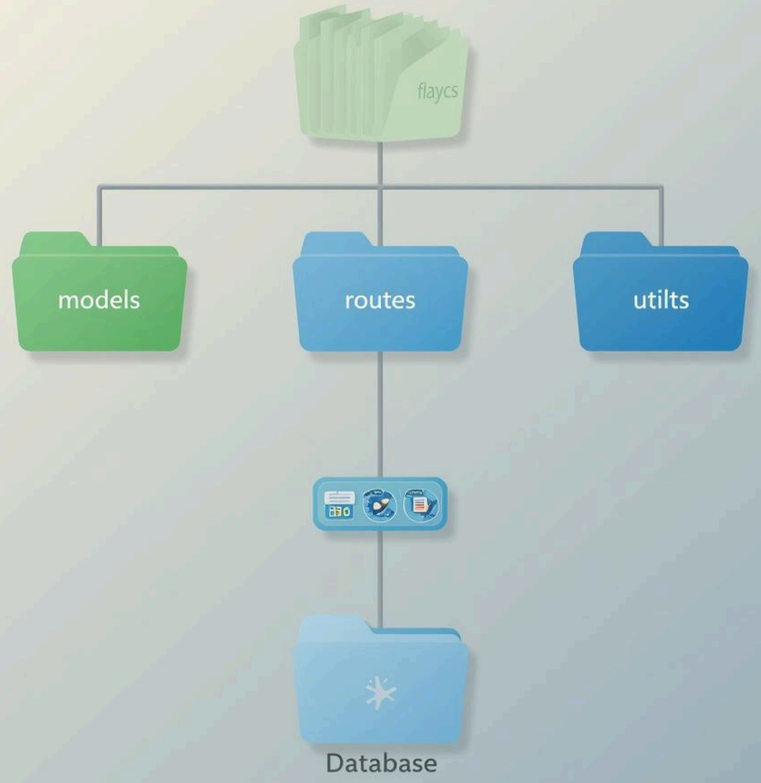
Presented by Janhavi Akarte

Project Overview of the Web App

A Robust Flask-Based Solution

The Stock Forecasting Web App is a full-stack web application designed to predict stock market trends using advanced machine learning models. Built with Flask as the backend framework, it integrates multiple ML algorithms such as ARIMA, Prophet, and Random Forest to analyze historical stock data and provide accurate predictions. The app is structured with a modular folder hierarchy that separates models, routes, templates, and utilities, making it easy to maintain and extend. Users can interact with the application through a responsive and interactive dashboard, which visualizes stock trends and predictions using dynamic charts.

The platform also incorporates additional features to enhance user experience and engagement. A chatbot, powered by AI, allows users to query stock-related information and get real-time insights. The app includes user authentication for secure access and offers features like trending stock tracking, prediction summaries, and data visualization tools. By combining machine learning, data analytics, and a user-friendly interface, this web application provides both investors and developers with a comprehensive solution for monitoring and forecasting stock market behavior.



Organizing the Flask Web App

Key Components of the Project Structure

- models/: Data handling logic
- routes/: API endpoint definitions
- utils/: Helper functions and decorators
- database/: Database connection setup
- templates/: Frontend HTML files

ML Models Overview

Key Approaches to Forecasting

ARIMA Models

ARIMA (AutoRegressive Integrated Moving Average) is a popular statistical method used for time series forecasting, capturing trends and seasonality in stock price movements effectively.

Prophet Models

Developed by Facebook, Prophet is designed for forecasting time series data that exhibits seasonality and trends, making it particularly useful for financial market predictions.

Random Forest

The Random Forest algorithm utilizes multiple decision trees to enhance prediction accuracy by reducing overfitting, providing robust stock forecasts based on historical data patterns.

User-Facing Functionality

Engaging and Interactive Experience

The Stock Forecasting Web App offers a range of interactive features designed for a seamless user experience. Users can log in or register securely through the authentication system, ensuring personal data and preferences are protected. Once logged in, users can select individual stocks from a searchable list and view detailed historical data. The app provides real-time predictions generated by machine learning models like ARIMA, Prophet, and Random Forest, helping users make informed decisions.
In addition to stock predictions, the platform includes a dynamic dashboard where users can visualize trends using interactive charts and graphs. The trending stocks page highlights market movers and patterns, offering quick insights into popular or volatile stocks. Users can also interact with the AI-powered chatbot for personalized queries, such as asking for stock forecasts, risk analysis, or investment suggestions. Overall, the app combines ML-driven insights with an intuitive interface, making it useful for both casual investors and data enthusiasts.

Supporting Modules Overview

Decorators

The **login_required** decorator ensures that users authenticate before accessing certain routes, enhancing security and safeguarding sensitive data within the Flask web application.

Helpers

Helper functions streamline repetitive tasks, such as data processing and validation, making the codebase more efficient and maintainable while improving overall application performance.

Conclusion & Next Steps

Capabilities, Improvements, and Scalability