Design Documentation

Overview

The Stock Price Prediction App, designed using Streamlit and written in Python, forecasts future stock prices using various data analysis techniques and visualization tools. The application leverages historical data and predictive algorithms to generate forecasts for specified periods.

Features

1.Ticker Selection:

Dropdown menu for selecting the stock ticker.
Fetches and displays relevant data based on the selected ticker.
SRS Reference: FR3.1.1.1, FR3.1.1.2.

2. Data Display:

Table displaying historical data for the selected stock, including date, close price, volume, and technical indicators.
SRS Reference: FR3.2.1.1, FR3.2.1.2.

3. Price Forecast:

Predict stock prices for one year using the forecasting algorithms.
Interactive graph showing historical data and forecasted values with confidence intervals.
SRS Reference: FR3.4.1.1, FR3.4.1.2.

4.Graph:

Graph depicting the stock values over time, useful for identifying overbought or oversold conditions.
SRS Reference: FR3.3.1.1, FR3.5.1.1.

5. Comparison of Actual and Predicted Prices:

Line chart comparing actual stock prices with predicted prices.
Visual representation of the model's performance.
SRS Reference: PR3.4.2.1, PR3.4.2.2

6. Close Price and Indicators:
Graph showing the historical close prices.
Additional indicators are plotted for further analysis.
SRS Reference: FR3.3.1.2.

Technical Specifications

1.Frontend:
Developed using Streamlit for creating interactive web applications with Python.
Interactive charts and graphs implemented using libraries like Matplotlib and Plotly.
SRS Reference: UI4.1.1, UI4.1.2.

2.Backend:
Python for handling data processing and API requests.
Data fetching from API. (e.g. yfinance)
SRS Reference: SI4.2.1, SI4.2.2

3.Data Processing:
Pandas for data manipulation and cleaning.
Prophet library for time series forecasting.
Technical analysis indicators calculated using TA-Lib or custom Python functions.
SRS Reference: FR3.2.1.1, FR3.4.1.1.

4. Visualization:
Matplotlib and Plotly for generating charts and graphs.
Interactive components provided by Streamlit for better user engagement and analysis.
SRS Reference: FR3.5.1.1, FR3.5.1.2.

User Interaction Flow

1. Stock Selection:

User selects a stock ticker from the dropdown menu.
SRS Reference: FR3.1.1.1, FR3.1.1.2.

2. Data Analysis:

Displays historical data table and various technical indicators.
Generates and displays the Prophet forecast graph for one year.
SRS Reference: FR3.2.1.1, FR3.4.1.1.

3. Technical Indicators:

Provides additional charts for indicators : RSI, EMA, and MACD.
User can switch between different time frames (e.g., 1 week, 1 month, 6 months, 1 year).
SRS Reference: FR3.3.1.1, FR3.3.1.2.

4. Performance Comparison:

Compares actual and predicted prices to evaluate the model's accuracy.
SRS Reference: PR3.4.2.1, PR3.4.2.2.

SRS Requirement Code	SRS Requirement Description	Design Document Section
FR3.1.1.1	Allow users to select a stock ticker from a dropdown list.	Ticker Selection
FR3.1.1.2	Update dynamically to reflect the data related to the selected stock ticker.	Ticker Selection
PR3.1.2.1	Load dropdown list for stock tickers within two seconds of initiating the application.	Ticker Selection
FR3.2.1.1	Fetch historical stock data from external sources using the yfinance API.	Data Display
FR3.2.1.2	Provide an error message if the selected stock ticker data is unavailable.	Data Display
FR3.3.1.1	Compute technical indicators such as EMA, MACD, and RSI.	Technical Indicators, Graph
FR3.3.1.2	Display these indicators in a comprehensible format on the user interface.	Data Display, Graph, Close Price and Indicators
FR3.4.1.1	Implement the Prophet model to forecast stock prices for up to one year.	Price Forecast
FR3.4.1.2	Implement an LSTM neural network for short-term stock price predictions.	Price Forecast
PR3.4.2.1	Generate and display forecasts within 30 seconds after user requests.	Price Forecast

PR3.4.2.2	Maintain accuracy levels as per predefined benchmarks during testing phases.	Price Forecast, Comparison of Actual and Predicted Prices
FR3.5.1.1	Display historical and predicted stock price data using interactive charts.	Graph, Comparison of Actual and Predicted Prices
FR3.5.1.2	Include options for plotting EMA fast, EMA slow, MACD, MACD signal, and RSI.	Graph, Close Price and Indicators
UI4.1.1	Present a graphical user interface (GUI) accessible via web browsers.	Frontend
UI4.1.2	Display a dropdown menu for stock ticker selection, interactive charts for data visualization, and controls for initiating data fetching and predictions.	Frontend, User Interaction Flow
UI4.1.3	Ensure the GUI is responsive and adapts to both desktop and mobile screen sizes.	Frontend
SI4.2.1	Interface with the yfinance API to fetch historical stock data.	Backend
SI4.2.2	Use the Prophet library for generating future stock price forecasts.	Backend
SI4.2.3	Use TensorFlow and Keras for constructing and training the LSTM model.	Backend