AI-powered Forex Trading Analysis Bot

CSC-225: Software Engineering

Team Members:
Member 1: Saad Nawaz, NUM-BSCS-2024-67
Member 2: Shahbaz Ali, NUM-BSCS-2024-73
Member 3: Hafsa Gul, NUM-BSCS-2024-24

Department of Computer Science
Namal University, Mianwali

Submission Date: Sunday 2nd November, 2025

Requirement Provider Agreement

This document confirms that the student team of Course CSC-225 Software Engineering has officially selected Mr. Mudassir Ahmad Khan as the Requirement Provider (RP) for the project titled *"Prototype Design for an AI-powered Forex Trading Analysis Bot."* The team agrees that the RP will guide them by providing real-world requirements, reviewing major deliverables, and validating the scope of the project when needed.

The RP has mutually agreed to support the team with requirement clarification and domain-related input throughout the project duration.

Requirement Provider

Name: Mudassir Ahmad Khan

Student Team

Saad Nawaz

Shahbaz Ali

Hafsa Gul

Date: Nov 07, 2025

Contents

1 Introduction

Foreign exchange markets (Forex) are highly liquid and operate 24/5. Traders use a mix of fundamental and technical analysis to determine the likely direction of the price. However, manual analysis is time-consuming and challenging, especially for novice traders who may struggle to combine multiple signals effectively.

This project proposes the development of an AI-powered Forex Trading Analysis Bot, which aims to simplify and automate the analysis process. The system will integrate technical indicators, summarize relevant fundamental data, and provide multi-timeframe directional suggestions (e.g., "major timeframe likely to move up; minor timeframe likely to move down"). Users will also have the flexibility to apply their own analysis tools and customize indicators according to their strategies.

The background for this project comes from the observation of the challenges faced by retail traders in effectively analyzing market data and making timely decisions. The proposed system will help bridge the gap between complex financial analysis and practical trading decisions, making sophisticated analysis accessible to a wider range of traders while providing explanations for its recommendations.

2 Problem Statement

Current traders, especially beginners and intermediate users, face several challenges when navigating the Forex market:

- **Fragmented analysis:** Technical indicators, chart patterns, and fundamental events are often scattered across multiple tools and platforms. Traders must manually collect, compare and interpret data from different sources, which is inefficient and prone to errors.

- **Time-consuming manual work:** Combining multiple timeframes and indicators manually is slow and error prone.

- **Lack of Explainability:** Many existing automated trading systems operate as "black boxes," providing signals without clear explanations. Traders are unable to understand the reasoning behind the signals.

- **Customization Need:** Experienced traders want to test and apply their own strategies and indicator combinations. Most current tools offer limited flexibility, which prevents personalized analysis.

- **Information Overload:** With constant updates in market data, news and economic events, traders can feel overwhelmed and confused. This can lead to inconsistent trading decisions, impacting profitability and confidence.

Goal: Build a prototype that (1) automates multi-timeframe technical and fundamental analysis, (2) offers an AI-assisted analysis mode, (3) allows users to apply their own tools/techniques, and (4) provides actionable, explainable buy/sell directional suggestions.

3 Project Objectives

The main objective of this project is to design a prototype for an AI-powered Forex Trading Analysis Bot that assists traders in making informed decisions. The project will achieve the following measurable objectives:

1. **Data Integeration:** Collect historical and live Forex price data and relevant economic calendar events via at least one public API.

2. **Technical Analysis Engine:** Implement a modular technical indicator engine capable of computing commonly used indicators such as moving averages (SMA, EMA), Relative Strength Index (RSI), MACD, pivot points, and support/resistance levels. The engine should be able to operate across multiple timeframes.

3. **AI Analysis Mode:** Implement an explainable AI model (based on classification or regression) that suggests directional market bias for selected instruments and timeframes. The model should also provide information about the importance of features to explain why a particular signal or prediction was generated.

4. **User Customization:** Provide a user interface to let traders apply their own indicators/thresholds, and test strategies. Users should be able to compare their personalized signals with AI-generated suggestions.

5. **Prototype UI and Reporting:** Deliver a simple web-based interface displaying selected charts, analysis results, multi-timeframe directional suggestions, and exportable reports.

4 Stakeholder Identification

- **End Users (Traders):** Retail traders using the tool for decision support.

- **Requirement Provider:** Domain expert providing requirements, guidance and feedback.

- **Development Team:** Project team responsible for design, implementation, and documentation.

- **Instructors/Evaluators:** Faculty members who assess milestone submissions, evaluate the prototype, and provide academic feedback.

- **Third-party API Providers:** External services that supply live and historical Forex price data, as well as economic calendar information (e.g., OANDA, Alpha Vantage). These APIs are essential for data integration and analysis functionality.

5 Software Development Methodology

5.1 Chosen Methodology: Agile with Scrum Framework

The Agile Scrum-Style methodology has been selected for this project due to its suitability for projects with evolving requirements and the need for stakeholder feedback from time

to time.Scrum-style two-week sprints will enable continuous integration of features and regular demos.

This approach is appropriate because:

- Requirements may change as the team gain deeper understanding of trading domain complexities.

- Frequent feedback from the Requirement Provider is essential for ensuring the system meets real-world needs.

- Regular sprint reviews allow the team to adapt to changing priorities and improve the system.

5.2 High-level 1-Year Schedule

Assuming a 12-month timeline using two-week sprints (approx. 26 sprints per year). Key milestones for each quarter are as follows:

- Project setup, RP onboarding, requirement refinement, initial data collection, basic technical indicators, simple UI mockups.

- Expand indicator library, multi-timeframe analysis, initial AI experiments, basic data storage setup.

- Develop AI model(s), add basic explanations for AI outputs, enable simple user customization features.

- Combine all components, do testing with RP, complete documentation, prepare demo version.

5.3 Sprint-level Plan (tentative)

Sprint	Focus	Deliverable
1	Project initiation	Sign RP agreement, create project repo, schedule first meetings
2	Data collection	Connect to one Forex API, collect sample historical data
3	Basic indicators	Implement SMA, EMA, RSI, MACD for one timeframe
4	Multi-timeframe engine	Run indicators on 4H, 1H, 15m; prepare simple UI mockup
5	Backtesting setup	Simple backtester for indicator signals on historical data
6	Feature engineering	Create features for AI: indicators, price deltas, volatility)
7	Baseline AI model	Train simple classifier/regressor (logistic regression or decision tree)
8	Explainability	Add simple explanations for AI predictions (e.g., feature importance)
9	User customization	Allow users to input own indicators and thresholds in UI
10	Integration	Combine AI outputs and user-defined signals in single view
11	Evaluation	Test on sample data, get feedback from RP, note improvements
12	Demo Prototype	Prepare working prototype for instructor demo, update documentation

6 Tools and Technologies

Languages: Python (for backend and AI model) and JavaScript/TypeScript (for frontend).

Backend frameworks/libraries:

- FastAPI or Flask for creating backend APIs.

- Pandas and NumPy for data processing and calculations.

- TA-Lib or simple Python code for technical indicators (SMA, EMA, RSI, MACD, etc.).

- scikit-learn, XGBoost, or PyTorch/TensorFlow for AI models.

- SHAP or basic explainability methods to show reasons behind AI predictions.

Frontend:

- React (or simple server-rendered pages) for the user interface.

- Plotly.js or TradingView lightweight charts to display Forex data and analysis.

Data / APIs:

- Forex price APIs such as OANDA, Alpha Vantage, or other free exchange rate APIs.

- Economic calendar and news data via events APIs or trusted websites.

Storage:

- PostgreSQL or SQLite for storing data, plus CSV files.

DevOps / Repository:

- GitHub repository with structured folders: `Documentaion/`, `Meeting_Minutes/`, `Readme.md/`, `Meeting_Videos/`.

7 AI Analysis and Evaluation

The system will use AI to analyze Forex market data. It will take inputs like indicator values from different timeframes, price history, volatility, and important economic events. The AI will give a simple directional signal for each timeframe (UP, NEUTRAL, or DOWN) along with a confidence score. If possible, more advanced models like LSTM or CNN may be used on time-series data.

The AI will also provide explanations in simple terms, so users can understand why a certain signal is suggested.

To evaluate the system:

- We will check how accurate the AI predictions are using past market data.

- Feedback from the Requirement Provider will be used to assess usefulness, clarity, and trust in the signals.

- We will also consider how easy it is for users to apply their own tools and interpret AI explanations.

To reduce risks:

- Backup data sources will be used if any API fails, and caching will help avoid limits.

- AI models will be carefully trained to avoid overfitting and lookahead bias.

8 References

1. Requirement Provider (RP), personal communication, Nov. 2025.

2. Team discussions, personal communication, Nov. 2025.

3. Prompts used with AI (ChatGPT) for project planning, objectives review, AI model design, tool and technology guidance, and sprint scheduling, Nov. 2025.