**Abstract**

This project focuses on developing an AI-based stock market prediction system with a user-friendly interface. Utilizing historical stock data and deep learning models, particularly LSTM (Long Short-Term Memory) networks, the system forecasts future stock closing prices. The project incorporates technical indicators such as SMA, EMA, RSI, and MACD to enhance prediction accuracy. A Streamlit-powered interface enables user interaction, real-time predictions, and graphical visualizations. Additionally, automation through Windows Task Scheduler allows for daily predictions and performance tracking against actual market data.

**Chapter 1: Introduction**

**1.1 Overview** Stock market prediction has long intrigued investors and researchers. With the rise of machine learning and deep learning, it's now possible to forecast stock prices more reliably by analyzing large datasets and discovering patterns.

**1.2 Problem Statement** Predicting stock market prices is challenging due to market volatility, noise, and countless influencing factors. Traditional models often fail to capture non-linear dependencies.

**1.3 Objectives**

* Build an LSTM-based predictive model for stock prices
* Integrate technical indicators for better accuracy
* Create a user-friendly UI for real-time predictions
* Automate daily predictions and evaluate performance

**1.4 Organization of the Report** This report includes literature review, system analysis, design, implementation, testing, results, and conclusion with suggestions for future enhancements.

**Chapter 2: Literature Survey** A variety of machine learning techniques have been used in the past for stock prediction including ARIMA, SVM, and Random Forest. Deep learning methods, especially LSTM networks, have shown significant improvement due to their ability to learn time dependencies.

**Chapter 3: Analysis and Requirements**

**3.1 Existing System** Conventional prediction methods rely heavily on statistical models and lack real-time interaction or automation.

**3.2 Drawbacks of Existing System**

* Low accuracy for non-linear data
* Limited to historical analysis
* No automation or real-time prediction

**3.3 Proposed System** A deep learning-based stock price predictor using LSTM, enriched with technical indicators, offering predictions through a user interface and automating daily updates.

**3.4 Advantages**

* Improved prediction accuracy
* Real-time predictions and visualization
* Automated daily predictions with evaluation

**3.5 Requirement Specification**

* Python, TensorFlow, Keras, Scikit-learn
* Streamlit (for UI)
* yFinance API
* Windows OS with Task Scheduler

**Chapter 4: Design**

**4.1 System Architecture**

* Data collection (CSV + yFinance)
* Preprocessing and scaling
* LSTM model training
* Prediction and evaluation
* UI display and automation

**4.2 Dataflow Diagram**

1. Input: Historical stock data
2. Preprocess & Feature Engineering
3. LSTM Model
4. Predict future price
5. Display results

**4.3 Use Case Diagram** Actors: User, System Use Cases: Input data, Generate prediction, View history, Schedule task

**4.4 Class Diagram** Classes: DataHandler, ModelTrainer, Predictor, UIHandler

**4.5 Sequence Diagram** User -> UIHandler -> Predictor -> Model -> Result -> UI

**Chapter 5: Implementation**

**5.1 Module Description**

* Data Preprocessing
* Model Training
* UI Display
* Daily Automation

**5.2 Algorithms Used**

* LSTM (Long Short-Term Memory)
* SMA, EMA, RSI, MACD for feature enhancement

**5.3 Materials Integrated**

* TensorFlow/Keras
* Streamlit
* yFinance
* Pandas, Numpy

**5.4 Main User Defined Functions**

* predict\_next()
* save\_prediction()
* compare\_with\_actual()

**Chapter 6: Testing and Result**

**6.1 Unit Testing** Each module like prediction, saving log, and UI tested independently.

**6.2 Integration Testing** Tested interaction between model and UI.

**6.3 System Testing** Verified the end-to-end workflow from data input to prediction and visualization.

**6.4 Results** The system shows strong predictive capabilities. Logged predictions are close to actual closing prices with measurable error.

**Chapter 7: Conclusion and Future Enhancement**

**7.1 Conclusion** An AI-based predictive system was successfully implemented using LSTM, integrated with a UI and automated daily forecasting. It improves reliability and usability for retail investors.

**7.2 Future Enhancement**

* Add support for multiple stocks
* Integrate sentiment analysis from news/Twitter
* Enable mobile UI or notifications

**Bibliography**

* TensorFlow Documentation
* Keras LSTM Tutorials
* Streamlit Docs
* Financial Market Research Papers

**Glossary**

* LSTM: Long Short-Term Memory
* UI: User Interface
* RSI: Relative Strength Index
* MACD: Moving Average Convergence Divergence

**Acronyms**

* AI: Artificial Intelligence
* CSV: Comma Separated Values
* API: Application Programming Interface

**Language Description** Python 3.12 used for implementation with standard ML/DL libraries.