CHAPTER 1: INTRODUCTION

The financial market is influenced by a combination of historical price patterns, macro events, and public sentiment. Retail investors and students often lack an integrated platform that combines price forecasting, news-based sentiment context, and a safe sandbox to test portfolio decisions. This project implements a Stock Price Prediction and Management system that integrates forecasting and sentiment analysis within a simulated trading and portfolio management workflow.

1.1 Background

Stock forecasting models attempt to learn relationships from historical price series, while sentiment analysis estimates how recent news or social discussions may influence market expectations. When combined, these signals can provide a more informative decision-support view than either source alone.

1.2 Problem Definition

Most stock-prediction demonstrations provide either:

- Forecasting models without user-facing workflows, or
- Trading simulators without integrated analytics.

This project bridges this gap by offering a single platform that fetches market data, generates short-horizon forecasts, computes model error, extracts news sentiment, and provides an interpretable recommendation in a simulated trading environment.

1.3 Project Overview

The Stock Price Prediction and Management system provides a unified web application that supports:

- short-horizon stock forecasting using Linear Regression,
- recent financial-news sentiment analysis,
- simulated trading and portfolio tracking and an admin monitoring dashboard.

1.4 Scope

1.4.1 In Scope

- Full-stack Flask web application
- Role-based authentication (user / admin)
- Market data retrieval (primary: yfinance; fallback: Alpha Vantage)
- Stock price forecasting using Linear Regression (7-day horizon)
- Multi-source news sentiment engine with fallbacks
- Portfolio tracking: holdings, transactions, dividends, wallet, broker commission
- Admin dashboard: brokers, transactions, users, companies, summary statistics

1.4.2 Out of Scope

- Real brokerage integration / real-money trading
- Advanced MLOps (scheduled retraining, model registry, experiment tracking)
- Production hardening (distributed caches, job queues, multi-tenant scaling)

1.5 Technology Stack

- Backend: Python, Flask
- Database/ORM: SQLite, SQLAlchemy
- ML & Analytics: scikit-learn (Linear Regression), pandas, numpy
- Sentiment / NLP: NLTK VADER, TextBlob, BeautifulSoup, requests, newspaper3k, aiohttp, tenacity
- Frontend: HTML/CSS, Bootstrap, JavaScript, D3.js