Stock Analysis Application - Technical Report

# Executive Summary

This report outlines the development of a stock analysis platform that combines real-time financial data with artificial intelligence-driven insights. The solution integrates technical analysis capabilities with a portfolio management system, delivering tailored features for analysts and investors. Powered by Llama 3 through Ollama, the application ensures efficient and insightful stock evaluation.

# 1. System Architecture

## 1.1 Technology Stack

- Frontend: Streamlit (Python-based web framework)

- Backend: Python

- Database: SQLite (via SQLAlchemy ORM)

- AI Model: Llama 3 (via Ollama)

- Stock Data: Yahoo Finance API (yfinance)

- Authentication: Custom JWT-based authentication system

## 1.2 Core Components

1. Authentication System

- JWT-based token authentication

- Role-based access control (Analyst/Investor)

- Secure password hashing

2. Stock Analysis Engine

- Real-time stock data fetching

- Technical analysis capabilities

- AI-powered analysis generation

3. Portfolio Management

- Report assignment system

- Portfolio tracking

- Report history management

# 2. Implementation Details

## 2.1 Data Models

1. User Model

- User authentication details

- Role management

- Portfolio relationships

2. Report Model

- Analysis content

- Timestamp tracking

- Analyst associations

3. Portfolio Model

- Report assignments

- User relationships

- Creation timestamps

## 2.2 Key Features

For Analysts:

- Real-time data fetching

- Technical indicator calculations

- AI-powered report creation

- Report management system

For Investors:

- Portfolio tracking

- Report access

- Historical analysis tracking

# 3. Technical Challenges and Solutions

## 3.1 Data Fetching and Processing

Initial issues involved handling empty DataFrames and duplicate columns. These were addressed through input validation, column renaming, and data consistency checks.

## 3.2 Report Management

Proper tracking and uniqueness of reports was achieved through timestamp tagging and robust ID management.

## 3.3 User Experience

Navigation was simplified using role-specific views and streamlined interfaces for both analysts and investors.

# 4. Security Considerations

- JWT-based secure authentication

- Role-based access control

- Input sanitization and session management

# 5. Future Improvements

1. Additional technical indicators and batch analysis features

2. Improved visual UI and mobile responsiveness

3. Performance enhancements through caching and async handling

# 6. Conclusion

The stock analysis tool integrates AI capabilities with core financial data processing to deliver real-time, intelligent analysis. Serving both analysts and investors, it offers a feature-rich platform with a focus on usability, accuracy, and security.

# 7. Technical Requirements

## 7.1 System Requirements

- Python 3.8 or higher

- SQLite Database

- Ollama with Llama 3 model

- Internet access for real-time data fetching

## 7.2 Dependencies

- streamlit, pandas, yfinance

- sqlalchemy, jwt, passlib, python-dotenv