PROJECT NAME : PORTFOLIO MANAGEMENT SYSTEM

GROUP:
GROUP 10

TEAM MEMBERS:

PALANISELVAM PRIYANKA (Team Leader) - A0307250R

DHARSHINI CHELLAPPA CHETTY RAJAN – A0307202X

ARYAN CHAKRABORTY - A0307245J

CHETANA BUDDHIRAJU - A0297857B

SWETA PATTNAIK - A0296537M

BUSINESS CASE / MARKET RESEARCH:

OVERVIEW:

In today's complex financial markets, robust portfolio management is essential for optimizing returns and minimizing risks. The system democratizes portfolio management by providing **automated, intelligent financial advice** to everyday investors, not just professionals.

The project focuses on building a portfolio management system using machine learning techniques.

RELEVANCE:

In today's financial landscape, effective stock market prediction plays a pivotal role in helping investors navigate the complexities of modern markets. Accurate stock prediction offers an opportunity to stabilize returns despite market volatility, which is especially relevant for financial hubs like Singapore. With a 2023 GDP per capita of over US$84,000, Singapore exemplifies how advanced predictive models can enhance financial resilience by enabling investors to respond proactively to fluctuations in both local and global markets.

For investors, reliable stock prediction models facilitate more confident decision-making by offering insights into future market movements. Studies show that models leveraging machine learning (ML), such as neural networks (NN), support vector regression (SVR), and hybrid methods combining genetic algorithms (GA) with long short-term memory (LSTM), can achieve higher precision in forecasting, enabling informed strategies and reduced risk exposure.

Automated portfolio management systems (PMS) democratize financial management by integrating real-time predictions and customized advice into user-friendly platforms, lowering costs and expanding access to robust financial planning tools. Furthermore, robust stock prediction tools are critical for optimizing profit potential, helping investors capitalize on market opportunities and anticipate downturns. As markets grow more complex, effective prediction models can support diversified portfolios by managing assets across classes like stocks, bonds, and real estate, ensuring consistent returns amid market fluctuations. With these tools, investors can build portfolios that reflect both personal financial goals and risk tolerance, providing a pathway to sustained growth in today's dynamic financial environment.

MILESTONES :

Literature Review Completion: Conduct a thorough review of existing literature on machine learning techniques used for stock market prediction, identifying best practices and gaps in current research.

Data Collection and Preprocessing: Gather historical stock data and technical indicators, followed by preprocessing to ensure data quality and readiness for model training.

Portfolio construction, selection and optimization using clustering algorithms like k-means, apriori and optimization techniques like PSO, GA, SA.

Model Development and Testing: Implement the LSTM model on other machine learning algorithms, to compare performance against baseline models.

User Interface Design: Develop a prototype of the user interface for the stock prediction platform, incorporating user feedback to enhance usability and functionality.

SYSTEM DESIGN:

This section outlines the overall architecture for our portfolio optimization system. The system integrates clustering, optimization, and predictive modeling in order to construct diversified as well as risk-adjusted portfolios for investors. The system comprises the following components:

Data Pipeline:

- This module imports and preprocesses historical financial data, such as stock prices, trading volume, and market capitalization from online repositories. In this module the data is cleaned, normalized, dimension reduction algorithms such as PCA used; this sets a strong foundation for the data for downstream processes.

Clustering Module:

- To facilitate the creation of targeted portfolios, the clustering module groups stocks based on key characteristics, such as historical returns, volatility, and other relevant metrics. Two clustering methods are utilized:
    - K-Means Clustering: Grouping of stocks with similar properties.
    - Apriori Algorithm: Understanding associations among stocks to generate clusters based on frequently co-occurring attributes.

Portfolio Construction:

- Different portfolio types are created within each cluster, such as Equally Weighted, Market-Cap Weighted, Global Minimum Variance, and Maximum Sharpe Ratio portfolios. The main function allows the user to select the type they are comfortable with and also giving them options to choose from diversified strategies whichever is in line with their risk-return preferences. To diversify the investment options, the project supports four types of portfolios:

- Equally Weighted Portfolio (EW):
  - All stocks within a cluster are assigned equal weights. This simple approach reduces exposure to individual stock risk, promoting diversification.
- Market-Cap Weighted Portfolio (MCW):
  - Stocks are weighted by their market capitalization, aligning the portfolio with broader market trends. This type suits investors seeking exposure to larger, more stable companies.
- Global Minimum Variance Portfolio (GMV):
  - This portfolio minimizes risk by assigning weights that produce the lowest possible portfolio volatility. GMV is ideal for risk-averse investors.
- Maximum Sharpe Ratio Portfolio (MSR):
  - MSR maximizes the risk-adjusted return by balancing weights for the highest return per unit of risk, a favored metric among investors looking for optimized risk-return profiles.

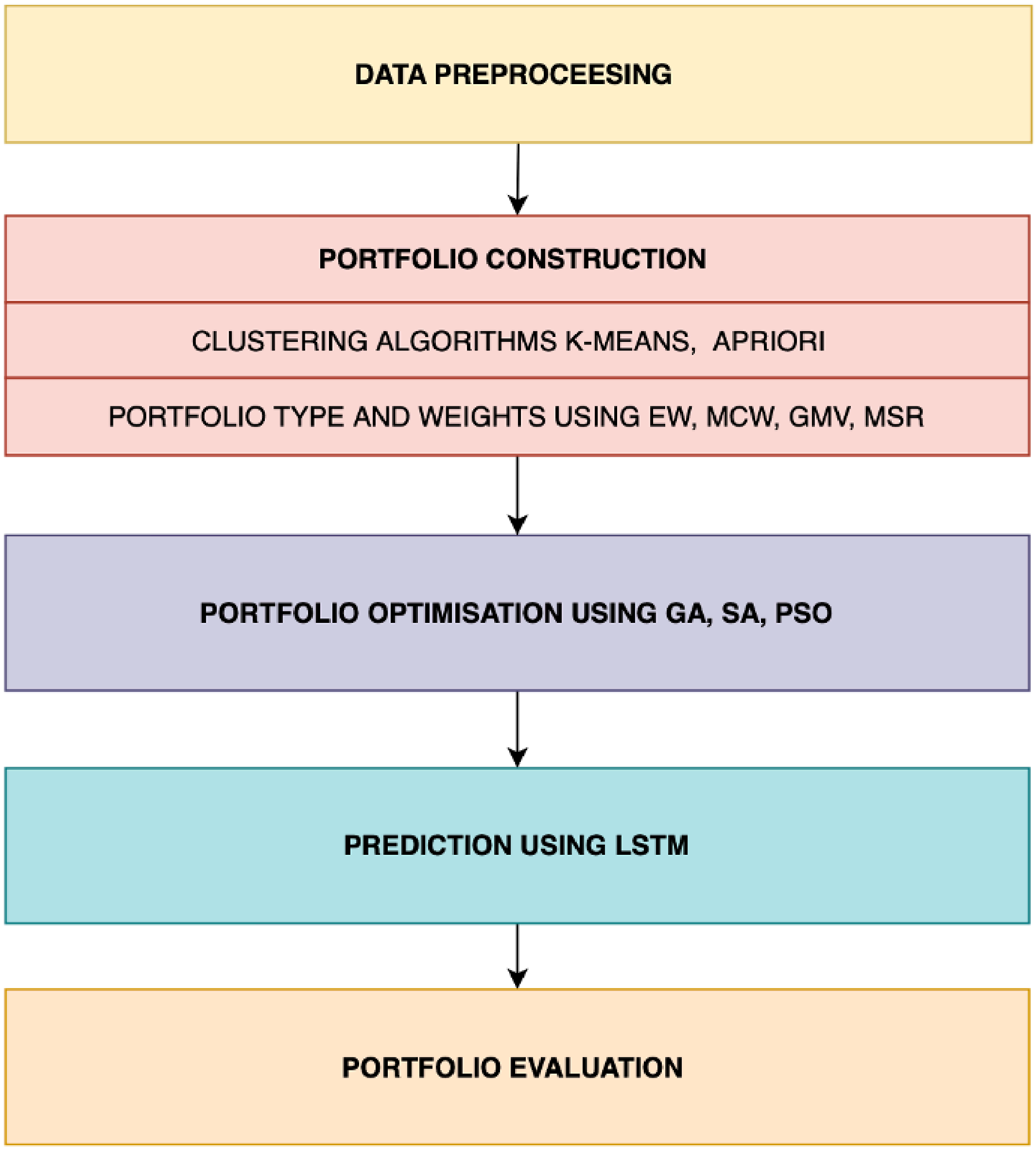
Portfolio Optimization Module:

- The system applies GA, SA, or PSO (we analyse the results from each of these) to optimize portfolio weights for each portfolio type. Each optimization technique aims to balance risk and return as the objective.

Forecast Model:

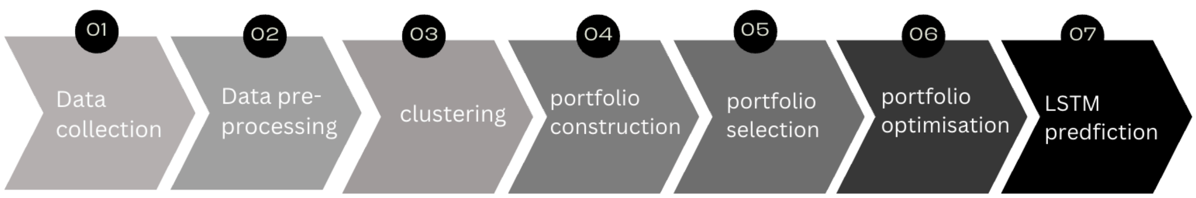
- The Long Short-Term Memory (LSTM) model predicts future stock prices, which provides valuable insights into expected returns and portfolio performance. By training on historical price data, LSTM can account for temporal dependencies and generate accurate forecasts.

HIGH LEVEL DIAGRAM:



IMPLEMENTATION AND FLOW OF THE PROJECT:

The project experimentation is divided into five main pathways, each involving portfolio construction, optimization, and prediction. The key components of each pathway are clustering, portfolio optimization, weight optimization, and LSTM prediction models.



Data collection

This project gathered data from multiple sources to analyze financial and economic factors affecting stock performance. This involved combining stock data for companies and ETFs with macroeconomic indicators, divided into two main parts: financial market data collection and macroeconomic indicator collection.

1. **Financial market data collection**
   Using yfinance, daily stock data was collected for SGX listings, major international stocks, and popular ETFs. Data was gathered from 1 January 2020 to 1 September 2024, including historical price, volume, and metrics such as Beta, P/E ratio, Market Cap, 52-Week High/Low, moving averages, dividend yield, revenue growth, and profit margin. Missing data was handled by omitting unavailable tickers. All datasets were combined and saved into a csv file.

2. **Macroeconomic indicator collection**
   Macroeconomic indicators for Singapore were retrieved from the World Bank using wbdata. Key indicators include GDP, Inflation, Interest Rates, Corporate Tax, Household Consumption, Government Spending, and FDI, covering January 2020 to September 2024. The data was standardized, and missing values were replaced with a placeholder (0.00001). The resulting dataset was saved as a csv file.

3. **PCA for macroeconomic indicators data**
   Principal Component Analysis (PCA) was applied to simplify the macroeconomic dataset. Data was standardized using StandardScaler, and three principal components were extracted to explain the majority of the variance. The PCA result was saved as a csv file.

4. **Data integration**

The stock data and processed macroeconomic indicators were merged by year to create a comprehensive dataset for modeling and analysis. This combined data allows for insights into the impact of economic conditions on stock performance, incorporating both company-specific and macroeconomic factors.

Data preprocessing

The dataset is ontains various financial indicators over time, such as stock prices, fundamental financial ratios, and other key metrics. The preprocessing starts with converting the date column to a datetime object to extract time-based features. Then all the features are standardised using the standard scaler. After that, the potential outliers are handled using outlier capping. Then feature engineering is performed to get daily_return that gives insights into daily volatility and market sentiment. After this, log transformation is carried out to reduce skewness in the distribution of certain features. Then we perform a selection to retain the most relevant columns for further analysis. Then, any missing values are handled by assigning a very small value such as 0.00001. And finally, all the irrelevant columns are dropped to get a final preprocessed dataset.

Experimentation overview

1. **Portfolio Construction**: For each approach, four portfolios were constructed:
    - **Equally Weighted Portfolio**: All stocks have equal weights.
    - **Market-Cap Weighted Portfolio**: Weights are assigned based on the market capitalization of each stock.
    - **Global Minimum Variance (GMV) Portfolio**: Weights are optimized to minimize the overall portfolio variance.
    - **Maximum Sharpe Ratio (MSR) Portfolio**: Weights are optimized to achieve the highest Sharpe ratio, balancing return and risk.
2. **Selection of Portfolios**: After constructing the four portfolios, one portfolio was chosen for the risk minimization approach and one for the return maximization approach.
3. **Stock Selection**: A function was implemented after clustering to obtain the 10 best stocks for risk minimization and return maximization.
4. **Optimization**: The chosen portfolio weights were further optimized using Genetic Algorithm (GA), Particle Swarm Optimization (PSO), or Simulated Annealing (SA) to enhance risk minimization or return maximization.
5. **Prediction**: LSTMs were trained to predict the future values of stocks, helping assess portfolio performance over time.

Pathways overview

1. **K-Means and Genetic Algorithm (GA)**
    a. Clustering: K-Means was used to group stocks based on selected financial features.

b. For optimisation, market cap portfolio was selected for risk minimisation portfolio and Global Minimum Variance portfolio was chosen for return maximisation portfolio.
c. Optimization: GA was employed to find the optimal weights for risk minimization and return maximization.

2. **K-Means and Particle Swarm Optimization (PSO)**
    a. Clustering: K-Means was used to group stocks based on selected financial features.
    b. For optimisation, market cap portfolio was selected for risk minimisation portfolio and Global Minimum Variance portfolio was chosen for return maximisation portfolio.
    c. Optimization: Instead of GA, PSO was used to optimize portfolio weights.

3. **Rule-Based Algorithm (Apriori) and Particle Swarm Optimization (PSO)**
    a. Association Rule Mining: Apriori was used to create clusters of assets based on frequent associations.
    b. For optimisation, market cap portfolio was selected for risk minimisation portfolio and Global Minimum Variance portfolio was chosen for return maximisation portfolio.
    c. Optimization: PSO was used to optimize portfolio weights.

4. **K-Means and Simulated Annealing (SA)**
    a. Clustering: K-Means was used to group stocks based on selected financial features.
    b. For optimisation, market cap portfolio was selected for risk minimisation portfolio and Global Minimum Variance portfolio was chosen for return maximisation portfolio.
    c. Optimization: SA was used for portfolio weight optimization.

5. **Rule-Based Algorithm (Apriori) and Simulated Annealing (SA) + LSTM**
    a. Association Rule Mining: Apriori was used to create clusters of assets based on frequent associations.
    b. For optimisation, market cap portfolio was selected for risk minimisation portfolio and Global Minimum Variance portfolio was chosen for return maximisation portfolio.
    c. Optimization: SA was employed for weight optimization.

PSUEDO CODE FOR THE ENTIRE FLOW:

1. Load Data:
- Import historical stock data (prices, volume, adjusted close, etc.).
- Get market capitalization for each stock.

2. Define Methods (Pathways):

- Combine clustering and optimization techniques:
  - K-Means + Genetic Algorithm (GA)
  - K-Means + Particle Swarm Optimization (PSO)
  - Apriori Algorithm + PSO
  - K-Means + Simulated Annealing (SA)
  - Apriori Algorithm + SA + LSTM

3. For Each Method:

a. Clustering Stocks:

- If using K-Means:
  - Cluster stocks based on selected financial features.
- If using Apriori Algorithm:
  - Identify frequent patterns to cluster stocks.

b. Construct Portfolios:

- Create these portfolios:
  - Equally Weighted
  - Market-Cap Weighted
  - Global Minimum Variance (GMV)
  - Maximum Sharpe Ratio (MSR)
- Choose portfolios for optimization:
  - Risk Minimization: Market-Cap Weighted Portfolio
  - Return Maximization: GMV Portfolio

c. Select Top Stocks:

- From each chosen portfolio, pick the top 10 stocks based on specific criteria.

d. Optimize Portfolio Weights:

- For the selected stocks, optimize weights using:
  - GA, PSO, or SA (as per the method).
- Objective:
  - Minimize risk or maximize return.
- Constraints:
  - Total weights sum to 1.
  - No negative weights (no short selling).

e. Predict Future Performance (if using LSTM):

- Train LSTM models on historical stock data.
- Predict future stock prices.
- Estimate future portfolio performance.

f. Assess Performance:

- Calculate metrics like expected return, variance, and Sharpe ratio.

- Compare the performance of optimized portfolios.
- Visualize results.

WHY EACH MODEL IS USED:

1. **K-Means Clustering**: This unsupervised learning method clusters stocks based on their features, such as Volume, Adj Cloase, P/E ratio, Beta, Daily Return, PC1,PC2 and PC3 to identify risk profiles and return potential. This helps in diversifying the portfolio by assigning assets into different clusters, optimizing risk exposure. This allows us to understand the different types of stocks present. This clustering also enables visual analysis through scatter plots, helping to distinguish characteristics across different stock types.
2. **Apriori Algorithm**: Apriori is used to identify frequent associations among financial assets.this model leverages associations that might not be apparent through traditional clustering, enabling more strategic diversification.
3. **Optimization Algorithms**:
   - **Genetic Algorithm (GA)**: GA mimics evolutionary processes to find the optimal portfolio weights that minimize risk or maximize returns. It is highly effective in exploring a large search space and escaping local optima.
   - **Particle Swarm Optimization (PSO)**: PSO models the behavior of a flock of birds, using candidate solutions that move toward better solutions over time. PSO is computationally efficient and often converges faster than GA.
   - **Simulated Annealing (SA)**: Simulated annealing (SA) is a probabilistic technique for approximating the global optimum of a given function aiming to escape local optima by using a temperature-dependent probability to accept worse solutions early in the process.
4. **LSTM**: The LSTM model was used to predict stock prices for each cluster and optimized portfolio. LSTM is a type of recurrent neural network (RNN) effective for time-series forecasting, providing insights into expected future performance and enhancing decision-making for portfolio management

WEB BASED IMPLEMENTATION USING FLASK

Flask was used to implement the final web interface for the financial portfolio prediction. Flask was used because it is a lightweight application in Python. The application allows users to interact with the results of the LSTM model, enabling them to explore and assess potential returns on investment by selecting different portfolio approaches, either focused on risk minimization or return maximization.

The web interface was implemented using Flask to serve the application pages, manage user inputs, and deliver predictions. The pre-trained LATM model was integrated into the application.

Core functionalities implemented

1. User Input and Portfolio selection

The flask application allows users to specify an input investment amount, as well as select one of the two strategies, low risk or high return.

2. Pretrained LSTM model integration

The LSTM model used is pre-trained with trained weights stored for later use. This ensures faster response time, as the model does not require to be trained from scratch each time a prediction is required. Upon receiving user input, the Flask application loads the pre-saved LSTM weights, along with the optimized portfolio weights for the selected approach (low risk or high return), to make predictions on the expected return over one month and three months.

3. Prediction

The output consists of the predicted Return, which is the expected financial returns over the next one-month and three-month periods, and investment Allocation, the calculated amount to be allocated to each stock within the selected portfolio, based on the optimized weight strategy.

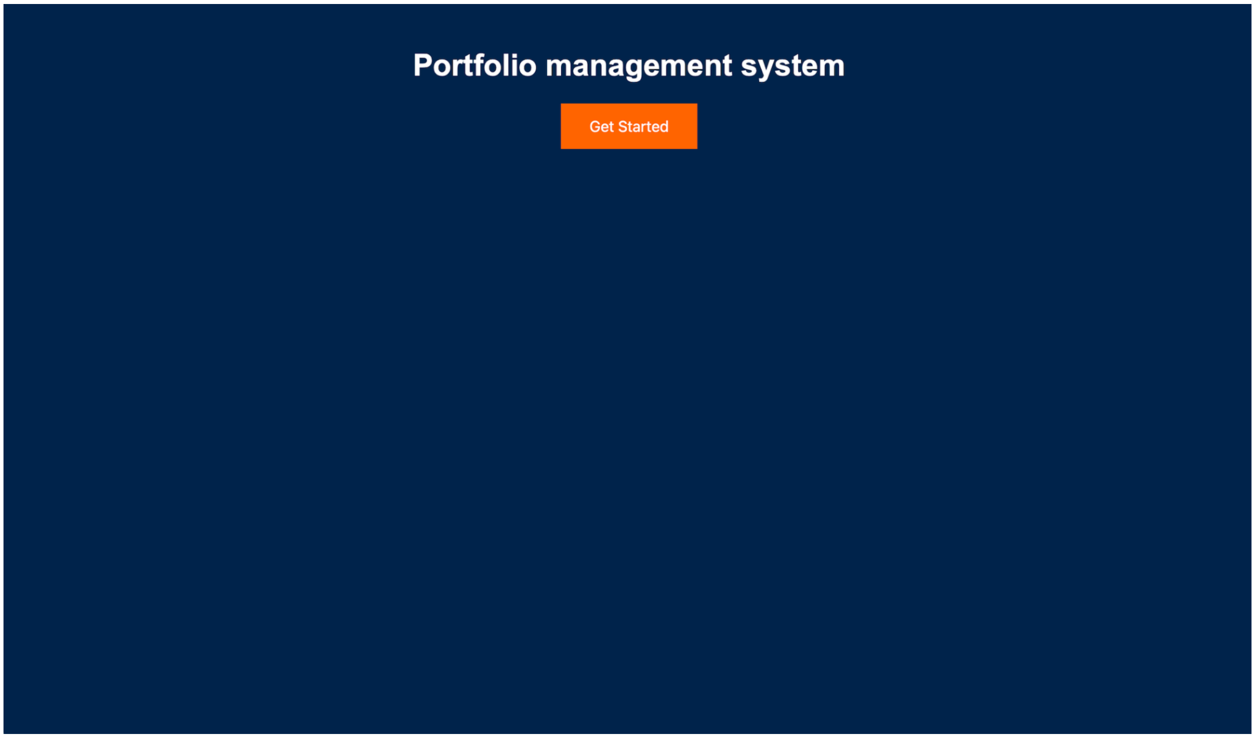
Technical aspects:

1. Routing: Flask's routing functionality was used to manage different URLs
2. Templates: HTML templates were employed.
3. Error Handling: Basic error handling was implemented to ensure that users received meaningful feedback for potential issues, such as missing inputs or invalid values.

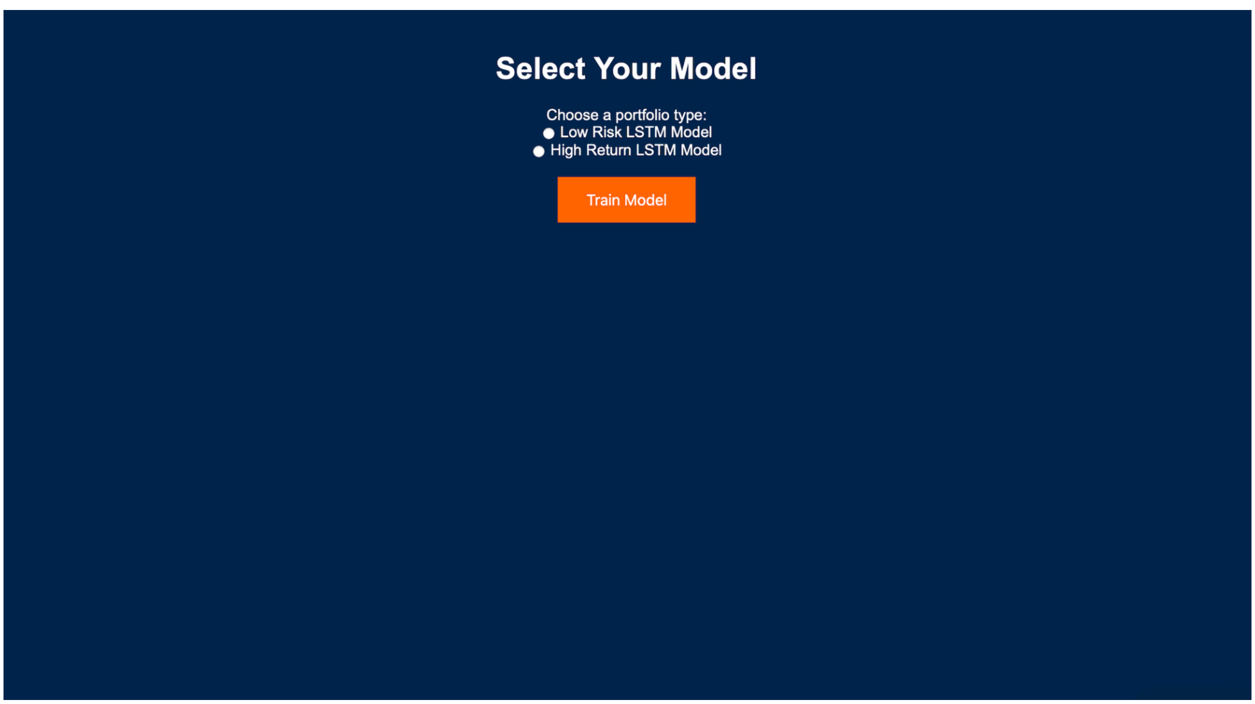
The integration of HTML, CSS, and Flask created a seamless experience where end users can directly explore the project's outcomes without requiring detailed technical knowledge, thereby enhancing the project's real-world applicability and impact.

WEB BASED APPLICATION DEMO SNIPPETS:

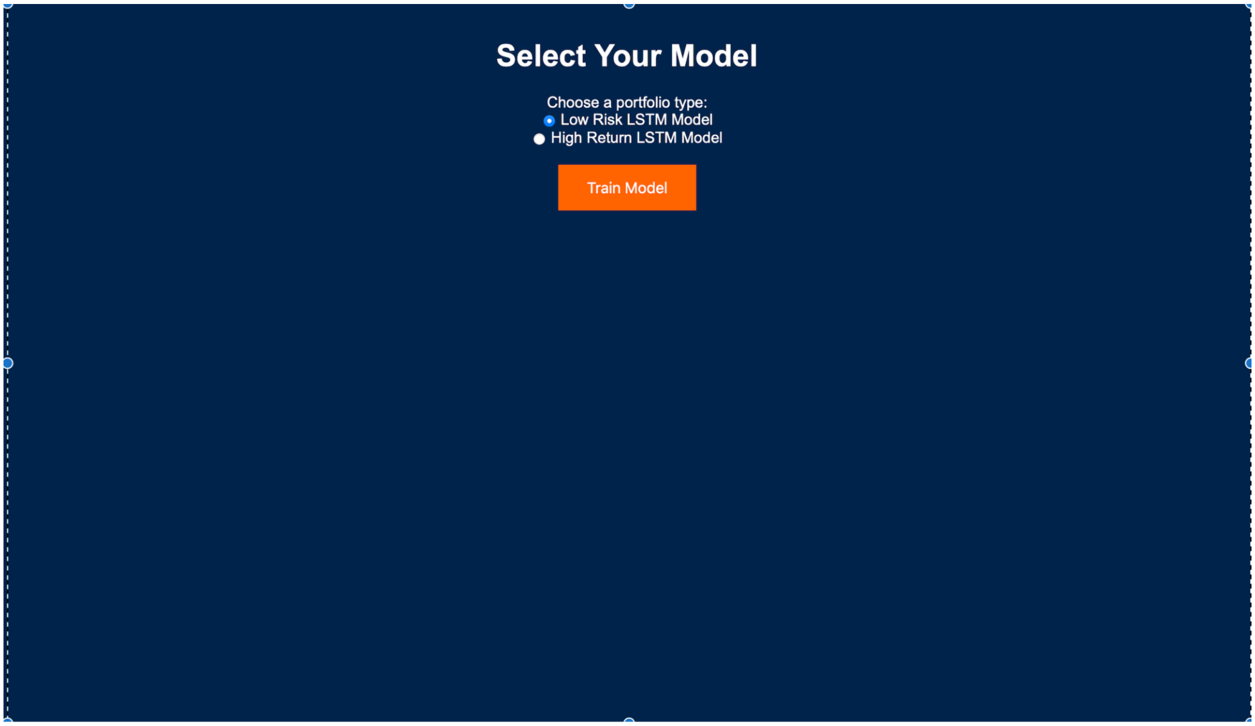
HOME PAGE OF THE WEB BASED APPLICATION:



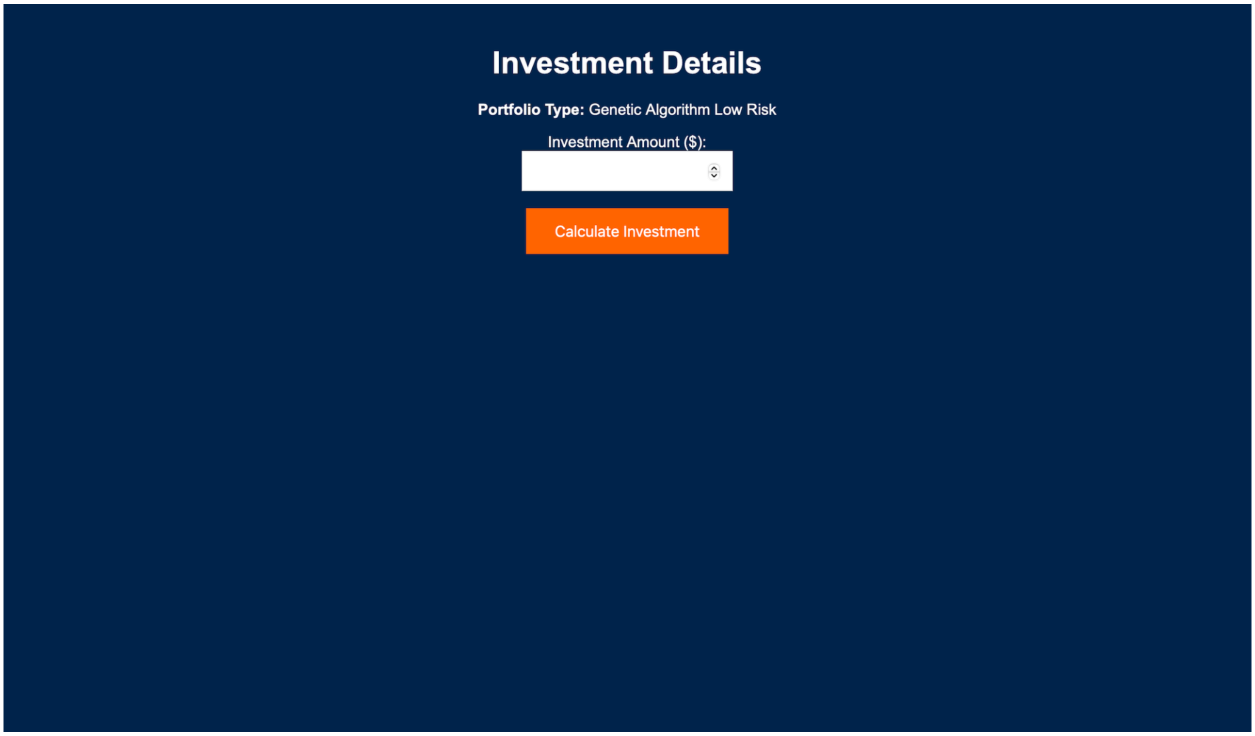
MODEL SELECTION PAGE:



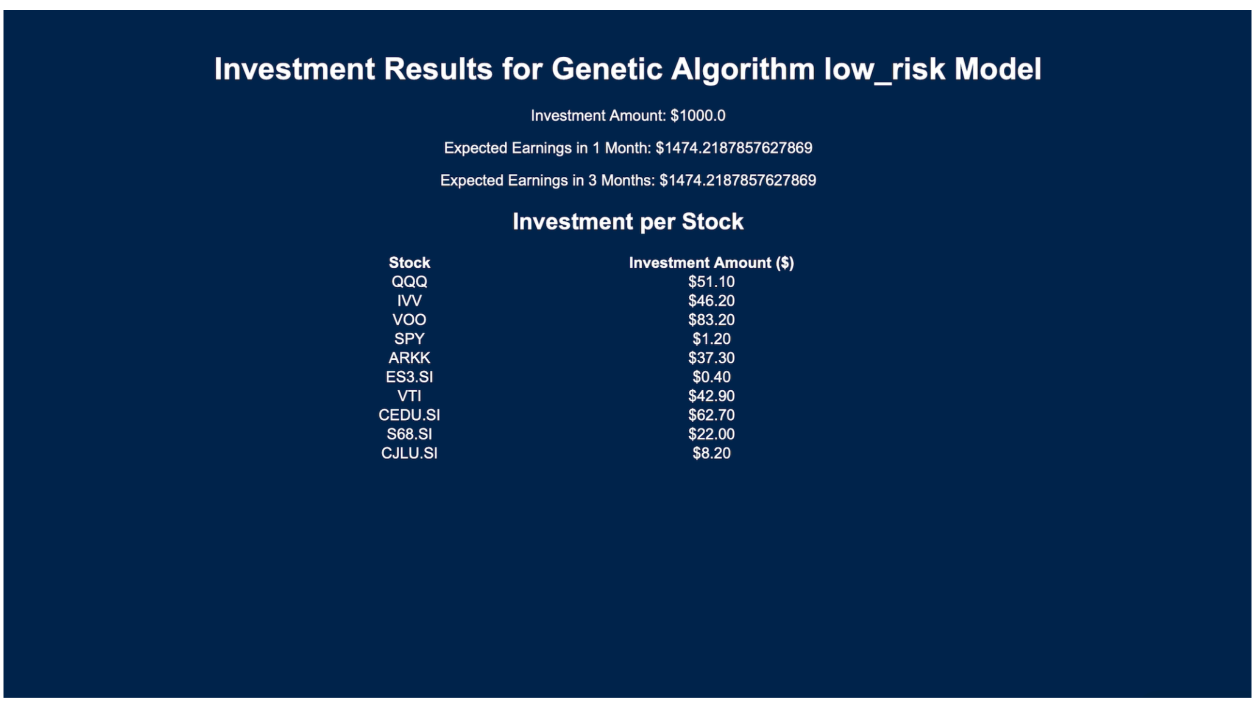
SELECTING LOW RISK APPROACH:



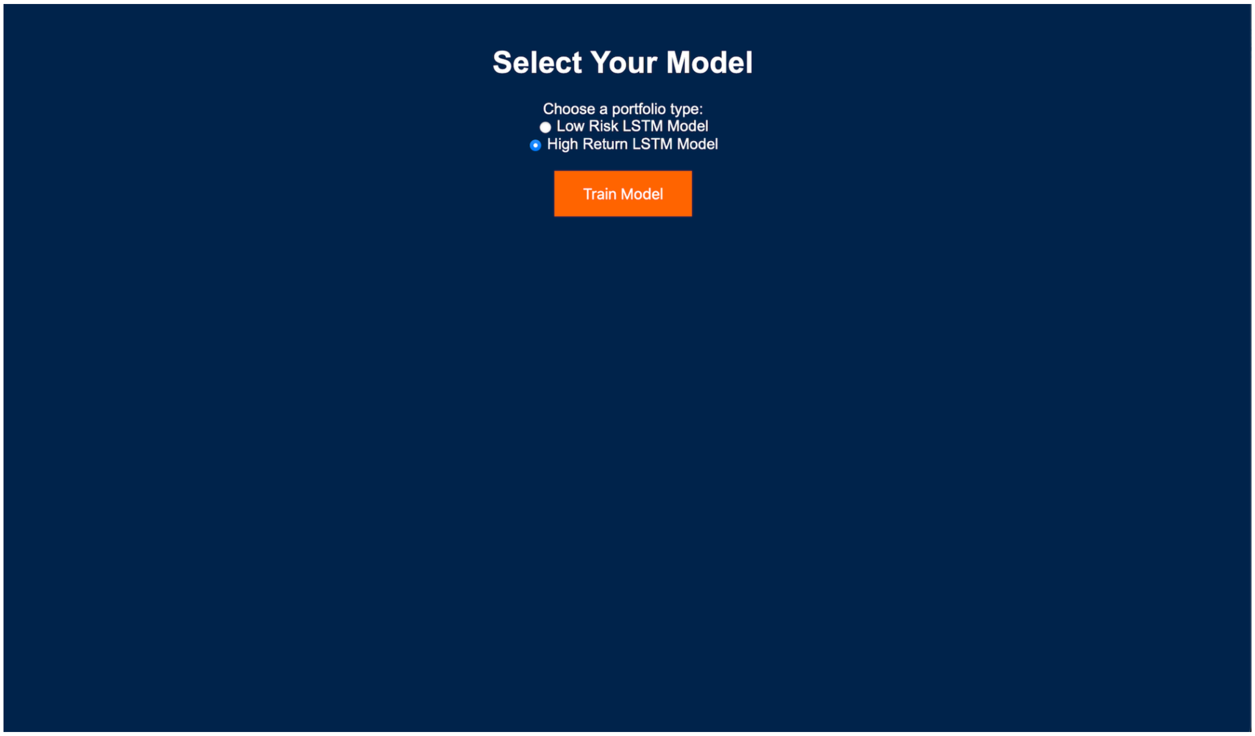
INVESTMENT AMOUNT SELECTION PAGE:



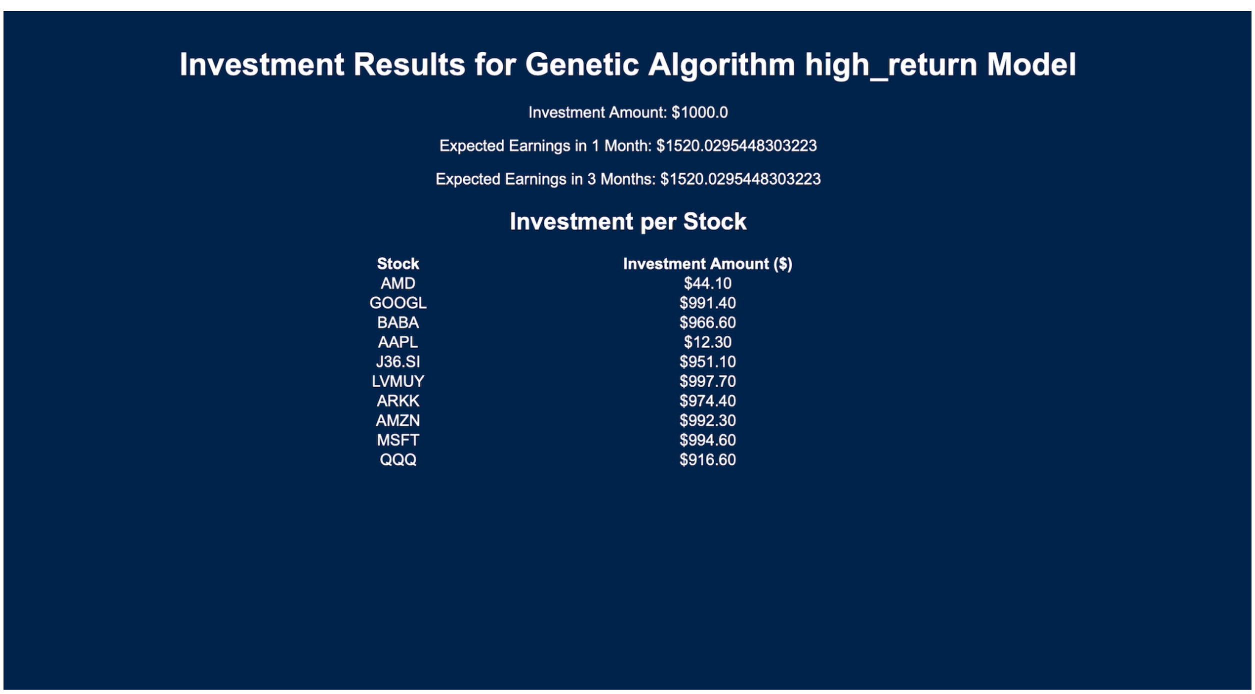
INVESTMENT RESULTS FOR LOW RISK APPROACH:



SELECTING HIGH RETURN APPROACH:



INVESTMENT RESULTS FOR HIGH RETURN APPROACH:



Findings and discussions

Return and volatility are calculated after building portfolio based on 4 portfolio construction methods, Equal-Weight (EW) Portfolio, Market-Cap Weighted (MCW) Portfolio, Global Minimum Variance (GMV) Portfolio, Maximum Sharpe Ratio (MSR) Portfolio.

Low Risk Portfolio								
	ew		mc		gmv		msr	
	return	volatility	return	volatility	return	volatility	return	volatility
kmeans - GA	0.0008	0.0094	0.0002	0.0083	0.0016	0.0154	0.0071	0.0616
kmeans - PSO	0.0004	0.0763	0.0000	0.0767	3.39E-05	0.0001	0.0031	0.7639
apriori - PSO	0.0080	3.03E-06	0.0010	5.18E-05	0.0009	3.67E-06	0.0008	4.17E-06
kmeans - SA	NA	NA	NA	NA	0.0001	0.0044	NA	NA
apriori - SA	0.0300	0.0141	0.0300	0.014142	0.0300	0.0140	0.0300	0.0141

High Return Portfolio								
	ew		mc		gmv		msr	
	return	volatility	return	volatility	return	volatility	return	volatility
kmeans - GA	0.0005	0.0127	0.0007	0.0136	0.0009	0.0103	0.0003	0.0278
kmeans - PSO	0.0055	0.1175	0.0000	0.2199	0.0005	0.0157	0.0127	0.2448
apriori - PSO	0.0045	1.84E-06	0.0078	1.29E-05	0.0067	2.11E-05	0.0029	7.88E-06
kmeans - SA	NA	NA	NA	NA	NA	NA	0.0101	0.2649
apriori - SA	0.0200	3.00E-05	0.0200	3.53E-05	4.5E-06	2.28E-06	0.0200	1.60E-05

Based on the returns and volatility, one portfolio is selected based on low risk and one based on high return.
For all the approaches, MSR portfolio has high returns and GVM has low risk.
These selected portfolios are further optimized to find the Root Mean Squared Error (Loss), Mean Squared Error (MSE) and Mean Absolute Error (MAE).

Low-Risk Analysis

- K-Means with PSO: Achieves a loss of 0.0679, with MSE also at 0.0679, and a relatively low MAE of 0.2165.
- Apriori with PSO: Shows the lowest loss of 0.0084 but has a high MSE of 3.4520, while MAE is at 0.0048.
- K-Means with Genetic Algorithm (GA): Results in a loss of 0.0480 and MSE of 0.0480, with a low MAE of 0.1427.
- Apriori with GA: No results available (NA) for loss, MSE, and MAE.

- K-Means with Simulated Annealing: Shows a very low loss of 0.0013 and a high MSE of 2.3188, with an MAE of 1.1421.
- Apriori with Simulated Annealing: Loss is 0.0891, with MSE of 0.0891, and MAE of 0.2568.

High-Return Analysis

- K-Means with PSO: Results in a loss of 0.0748, with MSE also at 0.0748 and an MAE of 0.2331.
- Apriori with PSO: Has the lowest loss of 0.0070, with MSE of 3.3500 and an MAE of 0.0047.
- K-Means with GA: Shows a loss of 0.0746, MSE of 0.0746, and MAE of 0.1847.
- Apriori with GA: No results available (NA) for loss, MSE, and MAE.
- K-Means with Simulated Annealing: Shows a high loss of 0.1441, MSE of 0.4173, and MAE of 0.4936.
- Apriori with Simulated Annealing: Loss is 0.0856, with MSE of 0.0856, and MAE of 0.2505.

LOW RISK				
Clustering	Optimization	Loss	MSE	MAE
K means	PSO	0.0679	0.0679	0.2165
Apriori	PSO	0.0084	3.4520	0.0048
K means	GA	0.0480	0.0480	0.1427
Apriori	GA	NA	NA	NA
K means	Simulated Annealing	0.0013	2.3188	1.1421
Apriori	Simulated Annealing	0.0891	0.0891	0.2568

HIGH RETURN				
Clustering	Optimization	Loss	MSE	MAE
K means	PSO	0.0748	0.0748	0.2331
Apriori	PSO	0.0070	3.3500	0.0047
K means	GA	0.0746	0.0746	0.1847
Apriori	GA	NA	NA	NA
K means	Simulated Annealing	0.1441	0.4173	0.4936
Apriori	Simulated Annealing	0.0856	0.0856	0.2505

Appendix

A. Project Proposal

Title: Portfolio Management Systems

Objective: To develop a web application that utilizes machine learning to aid investors in selecting optimized portfolio strategies. This application provides an interactive interface where users can input investment amounts and choose between low-risk and high-return portfolio approaches. By leveraging pre-trained Long Short-Term Memory (LSTM) models, the application calculates expected returns for one-month and three-month horizons, assisting users in making informed investment decisions.

Project Scope:
1. Data Processing and Model Training: Utilizing historical financial data to train an LSTM model for predicting portfolio returns.
2. Web Application Development: Building an interactive Flask-based web application to display prediction results to users.
3. User-Friendly Interface: Ensuring the application is accessible and easily understandable for both technical and non-technical users.

Expected Outcome: A user-friendly financial tool that leverages predictive modeling for portfolio selection, empowering users to make data-driven investment decisions.

B. Mapping of Functionalities Against Modular Courses

1. Machine Reasoning:
Application: Implementation of LSTM-based prediction models for decision-making processes aligns with the core principles of machine reasoning.
Contribution: The ability to process investment inputs, select optimized portfolios, and predict returns demonstrates reasoning capabilities rooted in quantitative analysis.

2. Reasoning Systems:
Application: The system leverages a structured process for risk and return prediction by organizing inputs, model weights, and strategies.
Contribution: It embodies a reasoning system by mapping user inputs to logical, predictive outputs based on model-driven insights, aligning with concepts in probabilistic and logical reasoning.

3. Cognitive Generative Systems:
Application: The application's generative aspect is demonstrated by dynamically generating predictions and suggesting investment allocations based on portfolio type.
Contribution: Through the interactive and iterative approach, where users can continuously select different strategies and obtain feedback, the system exhibits cognitive generative properties.

C. Installation and User Guide

1. Installation

Prerequisites:
- Python 3.8 or higher
- Flask
- Necessary libraries (e.g., TensorFlow, NumPy, Pandas)

Steps:
1. Clone the repository
2. Navigate to the project folder
3. Install dependencies:
   pip install -r requirements.txt
4. Start the Flask server:
   python app.py
5. Open the application in a browser at `http://127.0.0.1:5000`.

2. User Guide

Navigating the Application:
Home Page: Enter investment amount and choose between "Low-Risk" or "High-Return" portfolio strategies.
Result Page: Review predicted returns and suggested stock allocations.

Features:
Portfolio Strategy Selection: Choose between risk-focused or return-focused approaches.
Investment Allocation: Suggested allocation based on the selected strategy.

Error Handling:
If inputs are invalid or missing, clear messages will guide users to enter valid values.