Here is a detailed step-by-step plan for completing our project:

1. **Defining the Project Scope:**
   * Objective: Using Time Series Forecasting to forecast stock prices, using reinforcement learning systems to predict risk with stocks and recommend stocks for different portfolio/build portfolios.
   * Data Source: We will be using Yahoo Finance to gather historical data on stocks, indices, and possibly commodities or currencies.
2. **Data Collection:**
   * We did Web Scraping.
   * Used the yfinance Python library to fetch historical stock data (Date, Open, High, Low, Close, Volume, Dividends, Stock Splits).
   * Specifying the stocks and time range (from 1 January 1970 – 6 September 2024 EST).
   * Market indices is NYSE.
3. **Data Preprocessing:**
   * Handling Missing Data
   * Filling missing values with interpolation or remove rows if necessary.
   * Feature Engineering
   * Calculating technical indicators like moving averages, volatility, RSI, and momentum.
   * Create features for returns, percentage change, and volatility.
   * Outlier Detection
   * Identify outliers in stock prices and volumes.
   * Data Normalization
   * Normalize or standardize the data for better model performance.
   * Data Splitting
   * Split data into training, validation, and test sets.
4. **Exploratory Data Analysis (EDA):**
   * Statistical Summary
   * Generate summary statistics (mean, median, standard deviation, etc.).
   * Visualizations
   * Plot time series charts for stock prices.
   * Use heatmaps to visualize correlations between stocks.
   * Plot stock returns distribution and volatility trends.
   * Trend and Seasonality Detection
   * Use decomposition to identify trends and seasonality in stock prices.
5. **Time Series Forecasting (Price Prediction):**
   * Model Selection
   * Start with basic models like \*ARIMA\* or \*Exponential Smoothing\* to predict stock prices.
   * Move to advanced models like \*LSTM\* or \*Prophet\* for more complex patterns.
   * Model Training
   * Train models using historical stock data.
   * Use rolling windows to account for dynamic changes in stock behavior.
   * Model Evaluation
   * Use RMSE or MAE to evaluate the model’s prediction accuracy.
   * Cross-validate using train-test splits or time series cross-validation.
6. **Risk Identification (Reinforcement Learning & Risk Metrics):**
   * Reinforcement Learning Setup
   * Use \*Q-Learning\* or \*Deep Q-Networks (DQN)\* to model the decision-making process for buying/selling stocks based on stock price movements and risk levels.
   * Risk Metrics Calculation
   * Compute risk measures like:
   * Value at Risk (VaR)
   * Conditional Value at Risk (CVaR)
   * Sharpe Ratio
   * Maximum Drawdown
   * Simulating Risk
   * Train the RL model to optimize returns while minimizing risk by interacting with a simulated stock environment.
   * Evaluation
   * Backtest the RL system using historical stock data to check if it predicts risk accurately.
7. **Portfolio Optimization and Recommendations:**
   * Optimization Techniques
   * Use \*Markowitz Modern Portfolio Theory\* or other optimization techniques to recommend a balanced portfolio.
   * Set risk tolerance levels and optimize for maximum returns with minimum risk.
   * Reinforcement Learning
   * Implement an RL system that learns how to allocate stocks dynamically based on market conditions.
   * Test different portfolio strategies (e.g., aggressive, balanced, conservative).
   * Rebalancing
   * Introduce periodic rebalancing (monthly, quarterly) to adjust portfolio allocations.
8. **Backtesting:**
   * Backtest the Models
   * Test the portfolio strategies and risk predictions on past stock market data to evaluate performance.
   * Include transaction costs and slippage in the backtesting environment.
   * Key Metrics
   * Evaluate portfolio performance using Sharpe Ratio, cumulative returns, and maximum drawdown.
9. **Implementation & Deployment:**
   * User Interface
   * Build a dashboard using \*Streamlit\* or \*R Shiny\* to display stock recommendations, forecasted prices, and risk levels.
   * Real-time Updates
   * Implement live data fetching from Yahoo Finance to keep the model updated with the latest stock prices.
   * Notifications & Alerts
   * Add a notification system to alert users about significant price changes, portfolio rebalancing, or risk events.

**10. Final Report/Presentation:**

* + Documentation
  + Document each step of the process including data sources, model choices, evaluation metrics, and final recommendations.
  + Visualizations
  + Include visualizations of forecasted prices, risk levels, and portfolio allocations in your report.
  + Conclusion
  + Summarize findings and highlight the performance of your recommendation system and risk identification.

This is the plan we will be using to complete the project in a structured manner, focusing on both predictive accuracy and practical portfolio recommendations.