

# Assignment #1: Stock Market Simulation with REST, GraphQL, and WebSocket

---

**Time Allowed: 2 Weeks**

**Total Points: 100 (representing 13% of the course grade)**

## Objective:

Develop a stock market simulation where users can fetch stock prices, observe real-time price changes, and compare the efficacy of REST API, GraphQL, and WebSocket in delivering this data.

## Requirements:

### 1. Stock Data Entry and Retrieval using REST API (20 points)

- Develop a REST API to allow for the addition of new stocks (name, ticker symbol, current price).
- Using the same API, users should be able to fetch a list of all stocks and their current prices.

### 2. Fetching Specific Stock Details using GraphQL (25 points)

- Implement a GraphQL schema and resolvers to fetch detailed data for a particular stock, such as historical price data, highest and lowest prices, and trading volume.
- Using a UI, allow users to send a query to select a specific stock and retrieve its detailed data using GraphQL.

### 3. Real-time Stock Price Updates using WebSocket (35 points)

- Implement WebSocket functionality to send real-time updates of stock price changes to connected clients.
- Display a live updating chart or list that shows stock prices updating in real-time.

4. **Performance and Benefits Analysis** (15 points)

- You should use both the REST and GraphQL approaches to repeatedly fetch data and then compare it with the real-time data received via WebSockets.
- Write a brief report comparing the efficiency, speed, and benefits of using WebSocket versus traditional request-response methods for this application.

5. **Documentation** (5 points)

- Clearly document the setup, execution process, and any assumptions made during the development.

**Bonus (up to 5 extra points):** - Implement user-specific watchlists where users can add specific stocks they want to monitor and receive real-time updates only for those stocks.

### Submission Guidelines:

- Submit all source code (using github), the performance and benefits analysis report, and documentation. (One PDF File)
- Ensure your code is free from significant bugs.

### Evaluation Criteria:

- Code Quality and Structure (20%)
- Functionality (40%)
- Analysis Depth and Understanding (30%)
- Documentation (10%)