

CPSC 304 Project Cover Page

Milestone #: __3__

Date: __Oct 29, 2025__

Group Number: _55__

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Windy Huang	68306661	c1y3i	windyuniapp@gmail.com
Kylie Wang	69075752	o8s0u	kyliewang025@gmail.com
Zhengyang Xu	65484644	v0u7d	xuzhengyang2023@gmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Project summary:

Our project provides recommendations to people on which stocks to sell or buy. People using this application would be able to make more informed decisions when manipulating their stock market portfolio.

Timeline:

Important dates:

- Nov 16 (18 days, likely starting on Nov 6) - complete backend development
- Nov 21 (5 days) - backend frontend integration and HTML peer coding
- Nov 28 (7 days) - complete frontend development

Backend development:

1. Start-Up Script (Assigned: Windy)

Purpose: Initialize and populate the database with S&P 500 company and price data.

Tasks:

- Create a start-up script to populate the Company table with S&P 500 companies
- Connect to the Finnhub API
- For each company:
 - o Extract price history (from 2020 to current date)
 - o Populate the Exchange table as needed
- Challenges: API connection issues (e.g., authentication errors)

2. Update Script (Assigned: Kylie)

Purpose: Keep stock price data up to date after start up.

Tasks:

- Connect to Finnhub API
- Schedule daily execution after 4:30 PM (market close)

- For every stock:
 - o Insert the day's new price info into Price History table
 - o Detect and fill any missing entries since the start up script's execution
 - o Insert new dividend or stock split records
- Challenges: API connection and authentication issues

3. Report Script (Assigned: Windy)

Purpose: Import and process company reports from EDGAR

Tasks:

- Run script when a new report is added
- Scrape HTML reports from EDGAR and insert data into the Report table
- After completed, insert with 10 real reports, use dummy values for the rest
- Challenges: Report formats vary, requires trial and error parsing

4. Analyst Script (Assigned: Zhengyang)

Purpose: Analyze financial data and produce screening metrics.

Tasks:

- Assume a report exists for each company
- Use 10 actual reports, dummy values for the rest
- Implement screening formulas:
 - o Value Screener: $P/E < 20$, $EPS\ growth > 10\%$, $ROE > 15\%$
 - o Momentum Screener: $SMA20 > SMA50$ and 12-month momentum > 0.2
 - o Quality & Dividend Screener: $D/E < 1.0$ and Dividend Yield $> 2\%$
- Challenges: Understanding and correctly implementing formulas

Frontend development

List of stock (scroll)

Search bar (As you type the drop down changes, only click on dropdown, once clicked, the target stock on the left will scroll to the top)

User profile

Filter by industry

Graph of the stock price

If User haven't logged in yet - empty

If User has logged in - display there hold stocks (overlay)

If we click on a stock from the left, it replace the current with that stock

If you unclick the stock from the left, it goes back original

Analyst ratings

Hold

First 2 letter of email

Enter email

Set preferred industry (drop down list)

Structure & Layout

- HTML draft + peer coding (All members)
- Ensure clean structure and semantic layout

CSS implementation (Zhengyang)

- Responsive layout for browser resizing

Stock graph display (Kylie)

- Overlay multiple stock data
- Challenge: Handling overlay/removal logic efficiently

Interactive Components (Windy)

- Filtering and Search Bar
- Display stock list on left panel
- User popup
- Display analyst ratings
- Challenge: Avoiding merge conflicts and ensuring smooth integration of all parts