COL216 Assignment-1 Subtask-2: Report

Aneeket Yadav 2022CS11116 Kartik Gulia 2022CS11076

February 2024

1 Introduction

The goal of this document is to outline the design choices and challenges encountered during the subtask. We have attempted to keep the code as clean and modular as possible.

2 Design Choices

The number of clock cycles taken for evaluating the sum of first n natural numbers were calculated for the C-program and equivalent RISC-V program for the following two schemes:

2.1 React.js for Frontend:

- Modular Components: Promotes reusable and organized components.
- Declarative Syntax: Simplifies state and UI management.
- Virtual DOM: Enhances rendering performance.
- Efficient Updates: Selective re-rendering for improved efficiency.

2.2 Chart.js for Plotting:

- Easy Integration: Simple library for quick chart creation.
- Responsive Design: Automatic adaptation to various screens.
- Versatility: Supports a range of chart types for diverse data.

2.3 yfinance for Financial Data:

- Historic Tick Data: Includes valuable historical tick data which is essential for plotting a smooth curve but is not provided by jugaad-data.
- Rich Information: Provides comprehensive financial metrics.

2.4 Flask Framework for Backend:

- Simplicity: Minimalist design for straightforward development.
- Beginner-Friendly: Gentle learning curve for new developers.
- Flexibility: Unopinionated structure for adaptability.
- Built-in Server: Convenient testing with the included development server.

2.5 Ant Design (antd) for Styling:

- Ready-Made Components: Extensive pre-designed components for rapid development.
- Customization: Easily adaptable to match specific visual requirements.
- Responsiveness: Components designed to be responsive by default.
- Accessibility: Inclusive design with features for users with disabilities.
- Active Community: Large and supportive community for ongoing assistance.

3 Major Challenges

3.1 CORS

- When a web page makes an XMLHttpRequest or fetch API request to a different domain (cross-origin request), the browser enforces the Same-Origin Policy by blocking the request. CORS(Cross-Origin Resource Sharing.) provides a way for servers to declare which origins are permitted to access their resources and which HTTP methods (GET, POST, PUT, DELETE, etc.) are allowed.
- If you don't include CORS headers on the server and you attempt to make a cross-origin request (e.g., using JavaScript in a web browser) to an API that is hosted on a different domain, the browser will enforce the Same-Origin Policy, and the request will likely be blocked.

3.2 Simultaneous Fetch Request

- Unlike jugaad-data, yfinance does not support accessing financial data of multiple stocks simultaneously. This lead to significant when plotting comparison charts.
- Finally a series model(i.e delayed request model) was implemented to access stock data for comparison.

3.3 Learning Curve

By far,the greatest challenge in the assignment was the learning curve. We realised early that developing a clean interface had to involve expanding beyond HTML to CSS and Javascript. In particular, having to learn `react.js` in such a short time period was the toughest part.

4 Acknowledgements

We are thankful to freeCodeCamp for their excellent tutorials on HTML, CSS, Flask and React.js