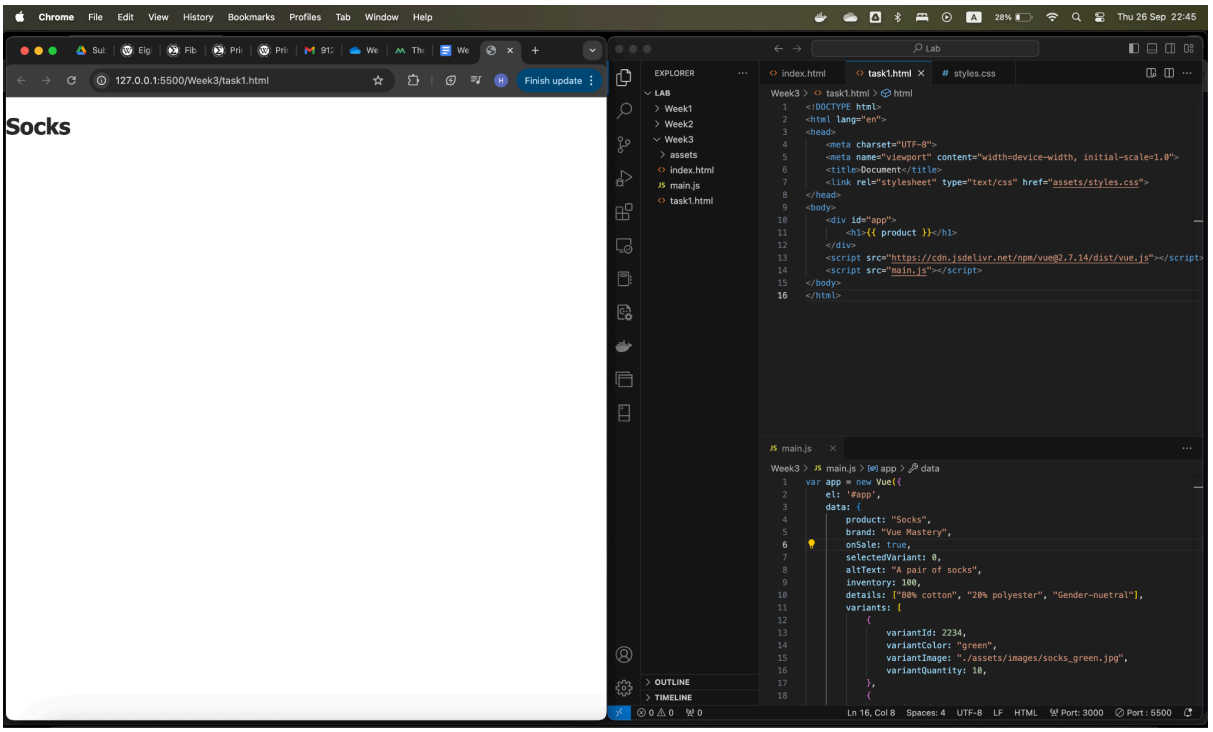
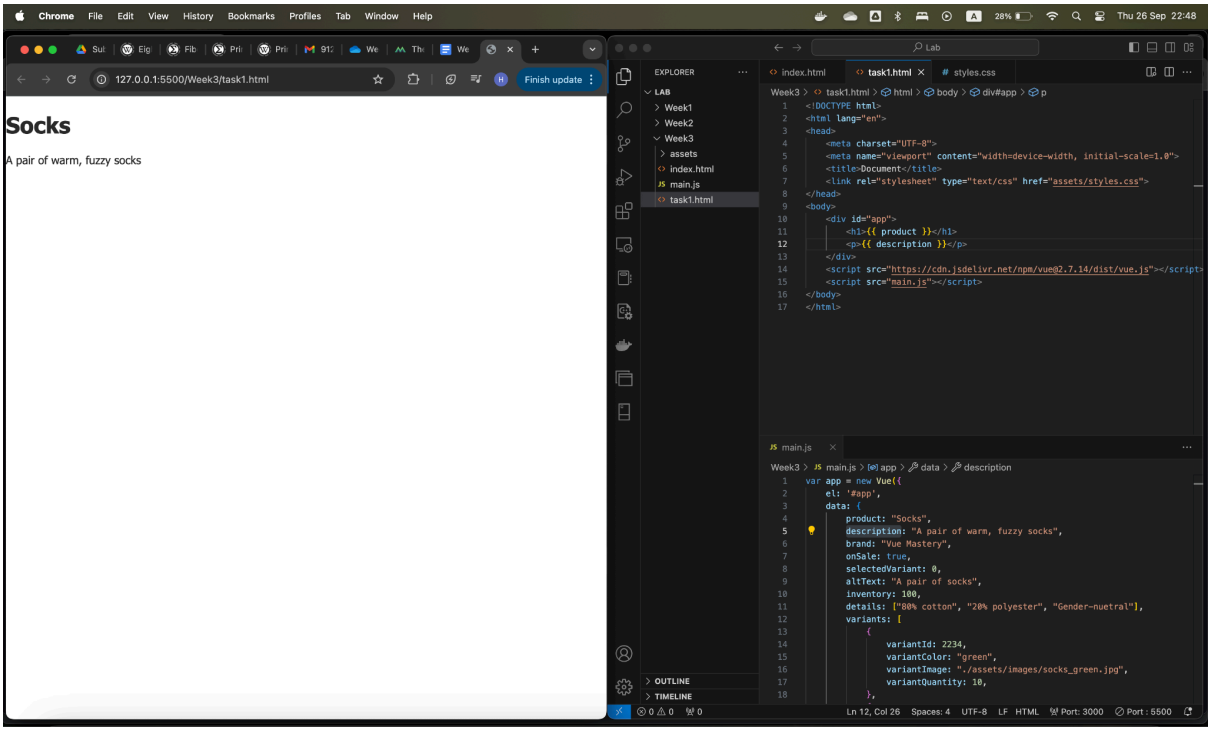
Full Name / Student ID	Huynh Thanh Tuyen / 001323067
Submisstion Title	COS1103 Task 1 & 2
Course Name	Web Programming 2
Course Code	COS1103

Task 1 (Lesson 1-4)

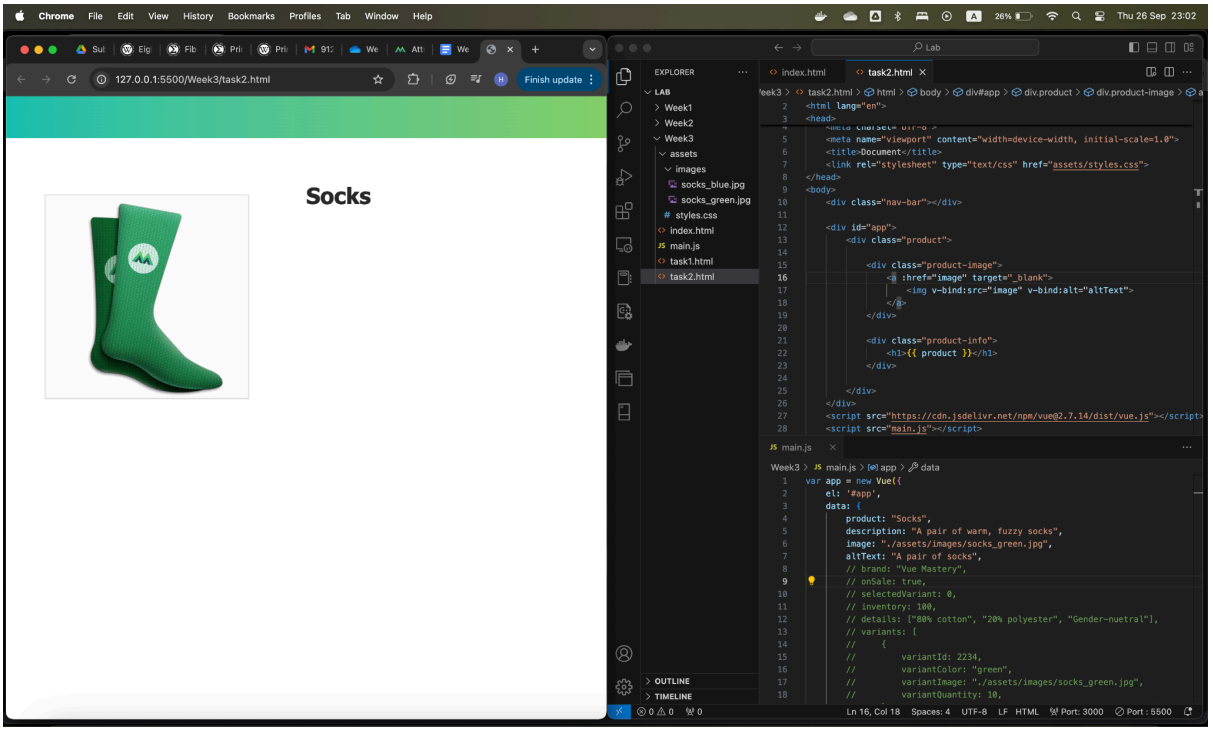
Lesson 1: The Vue Instance



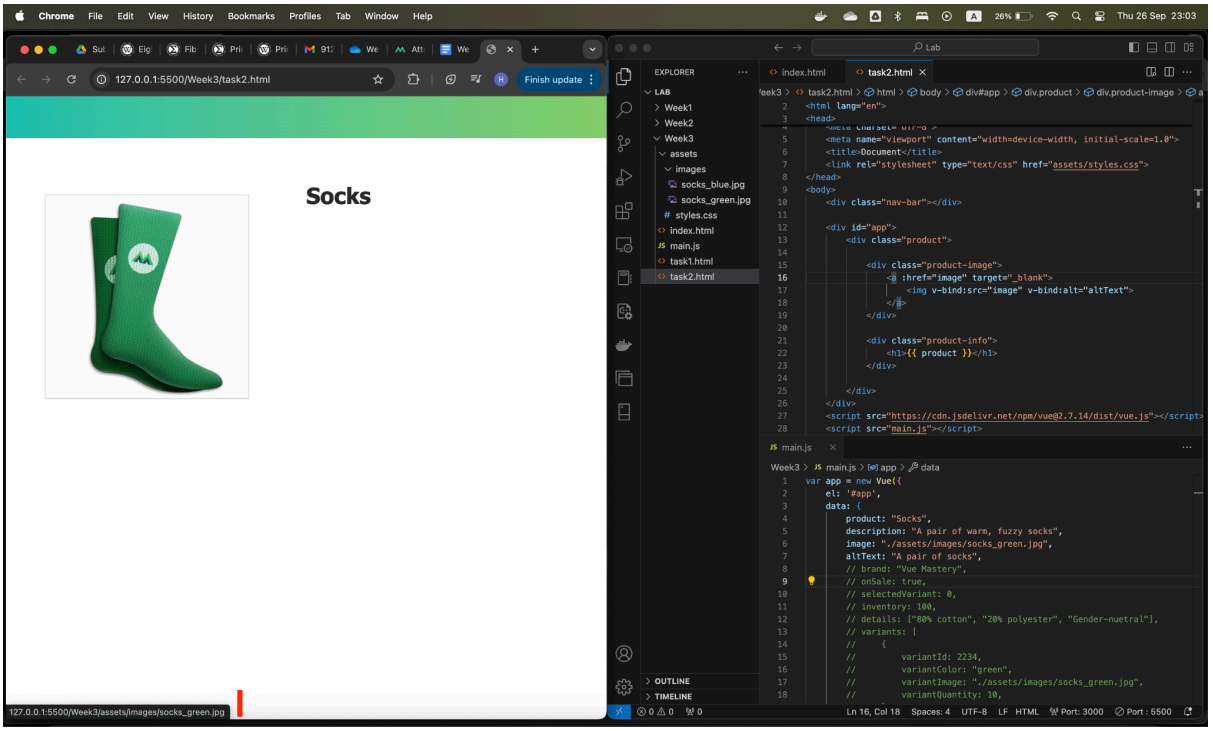
Challenge of lesson 1:



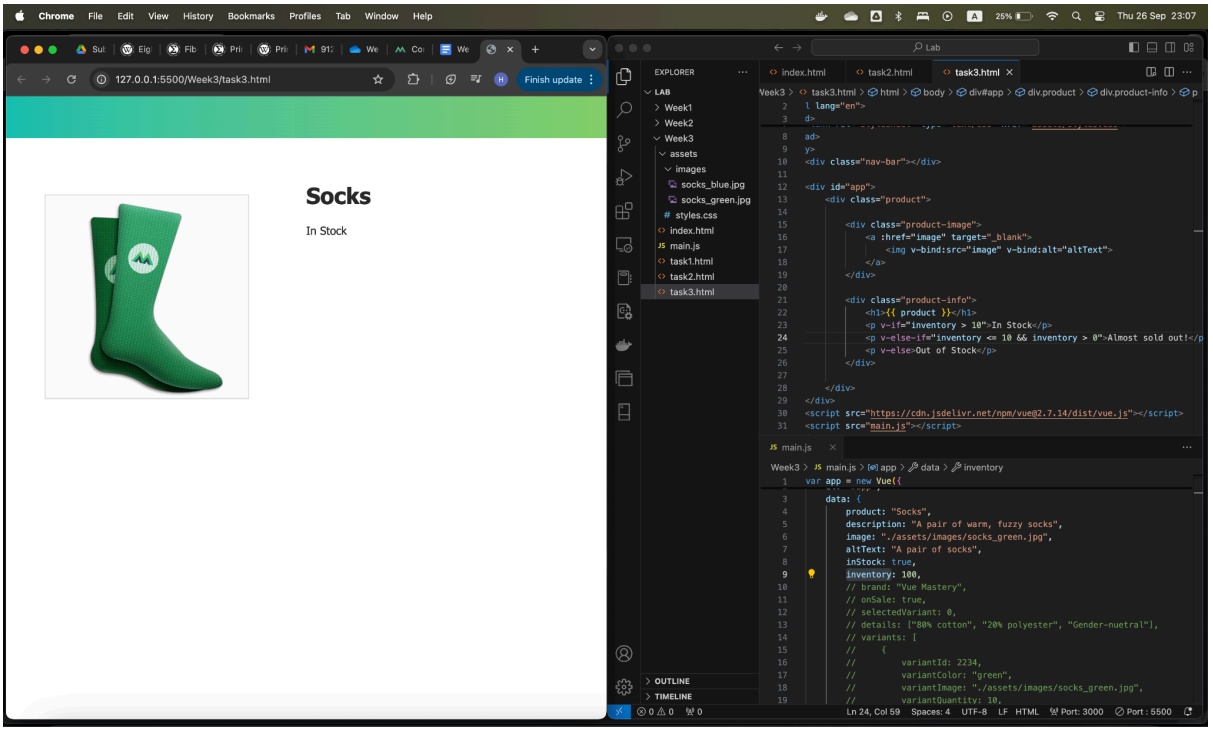
Lesson 2: Attribute Binding

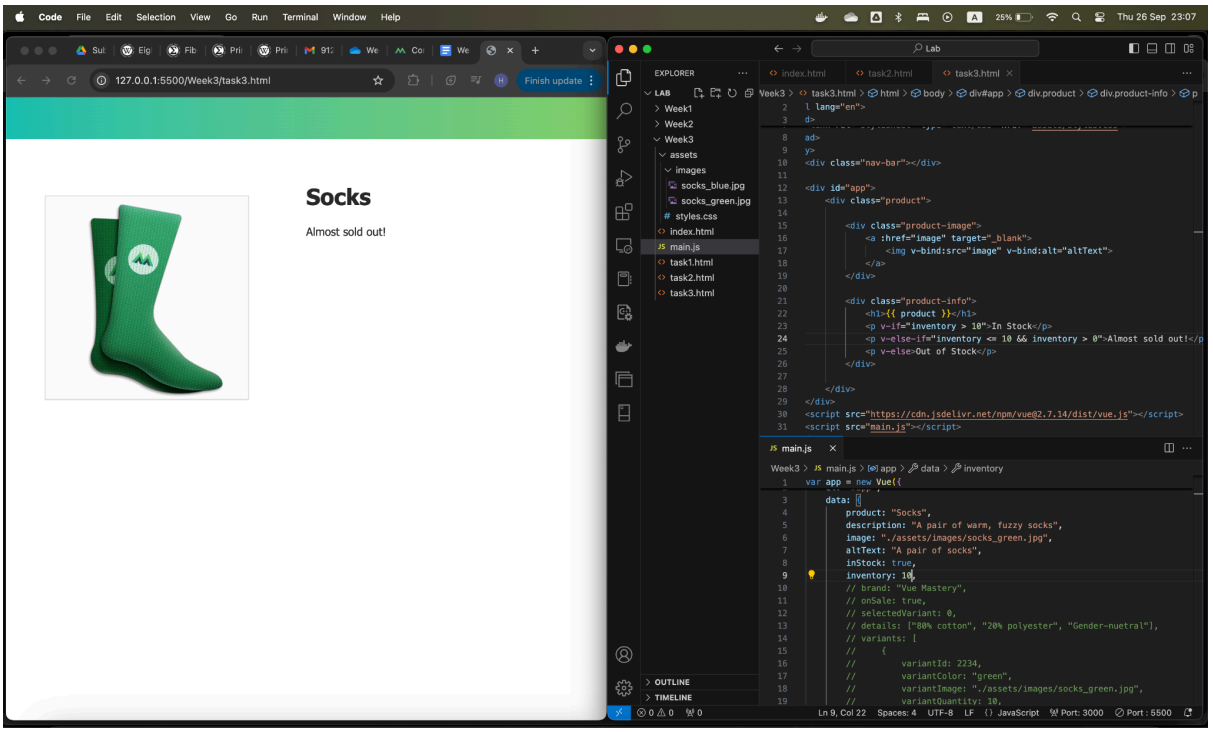


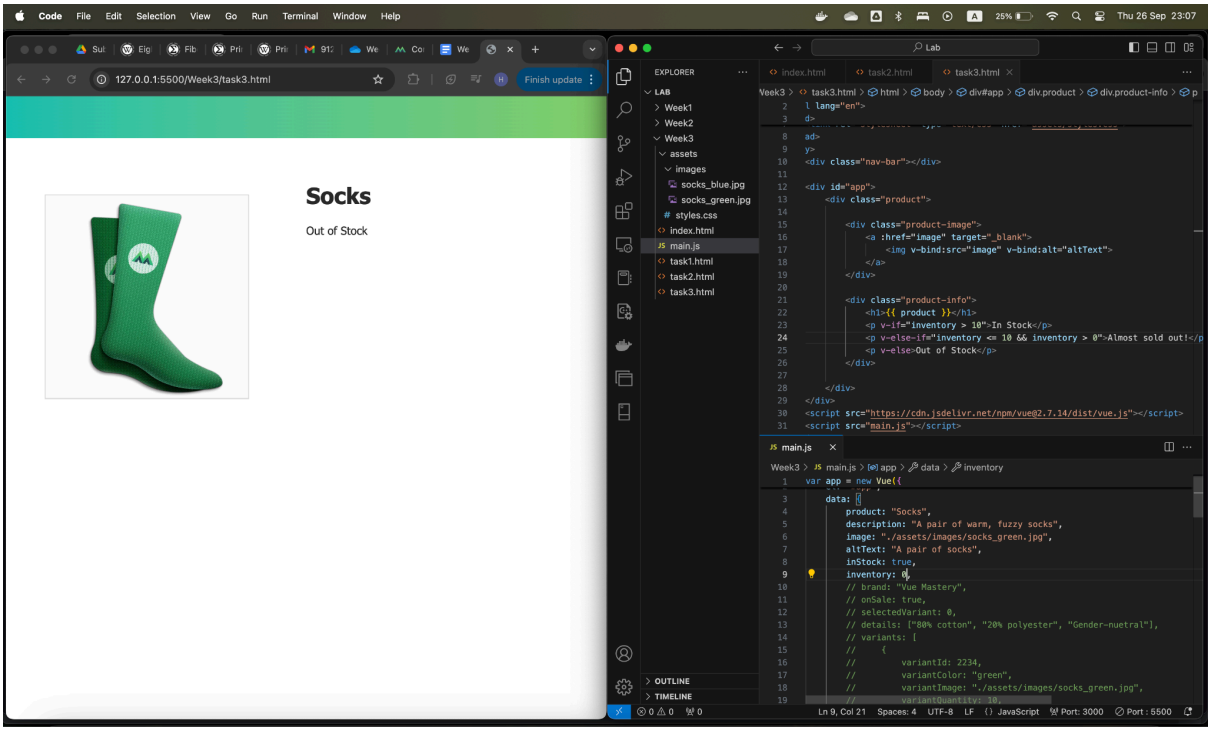
Challenge of lesson 2



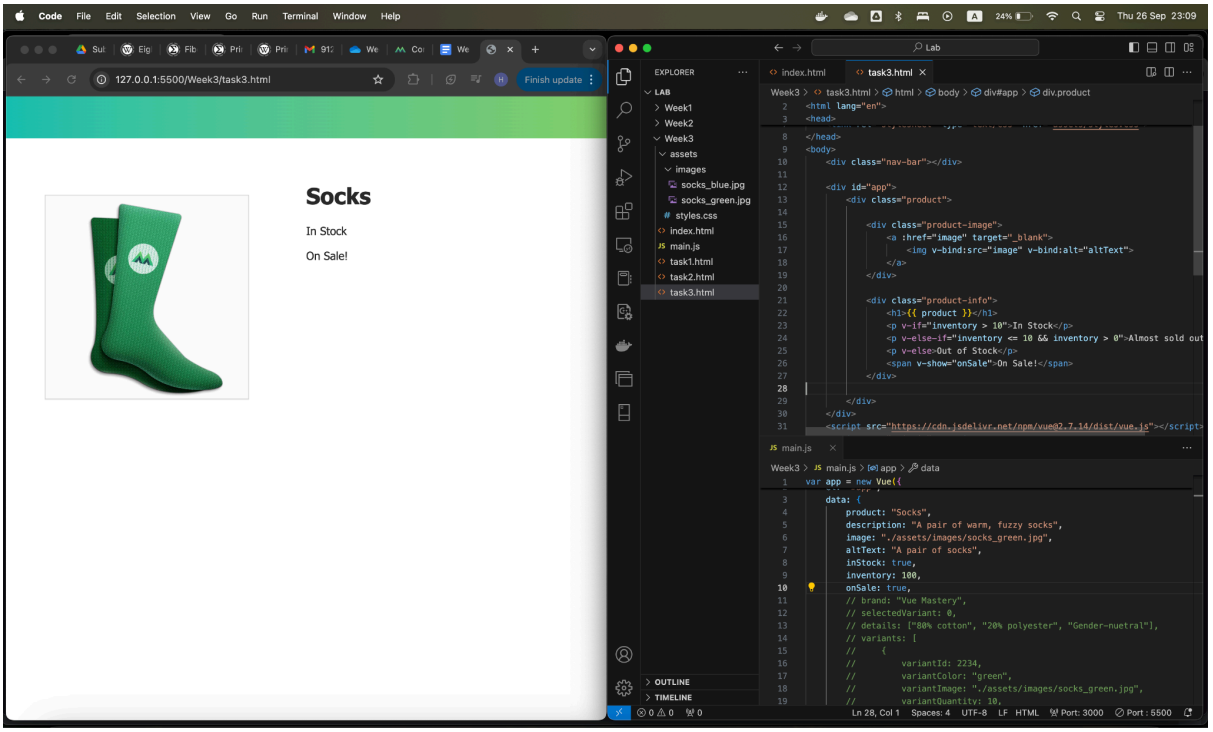
Lesson 3: Conditional Rendering

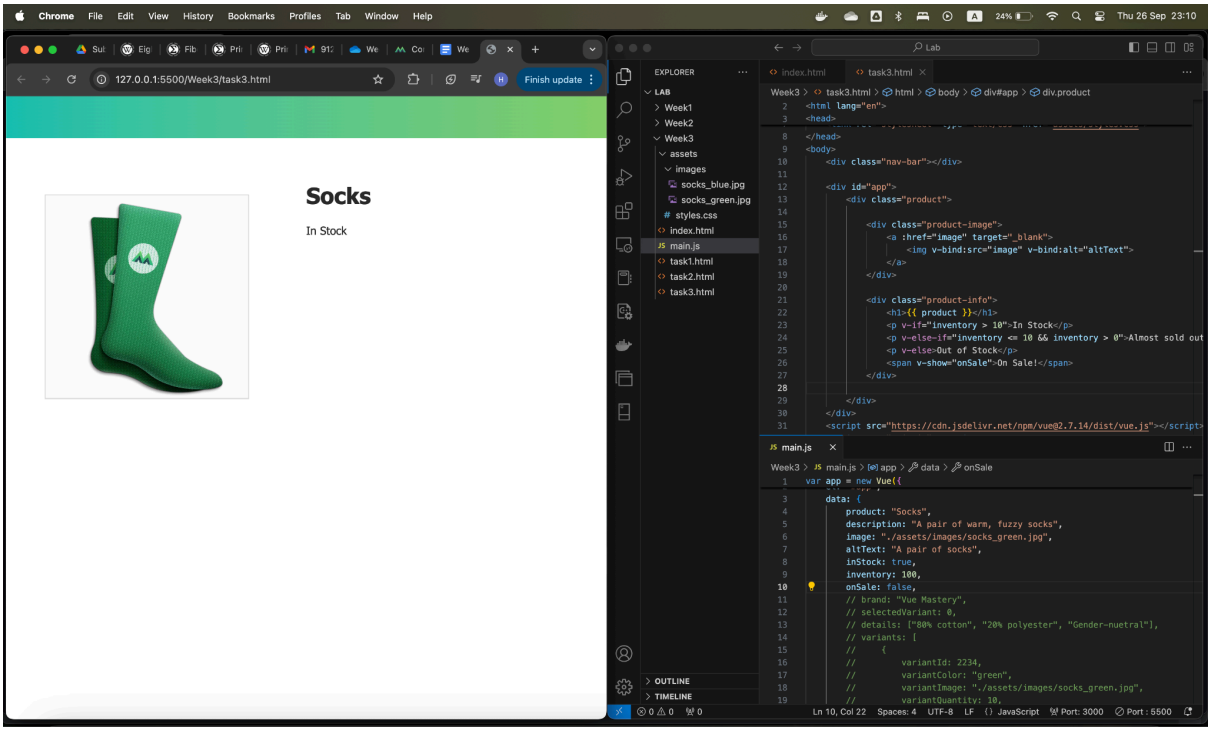




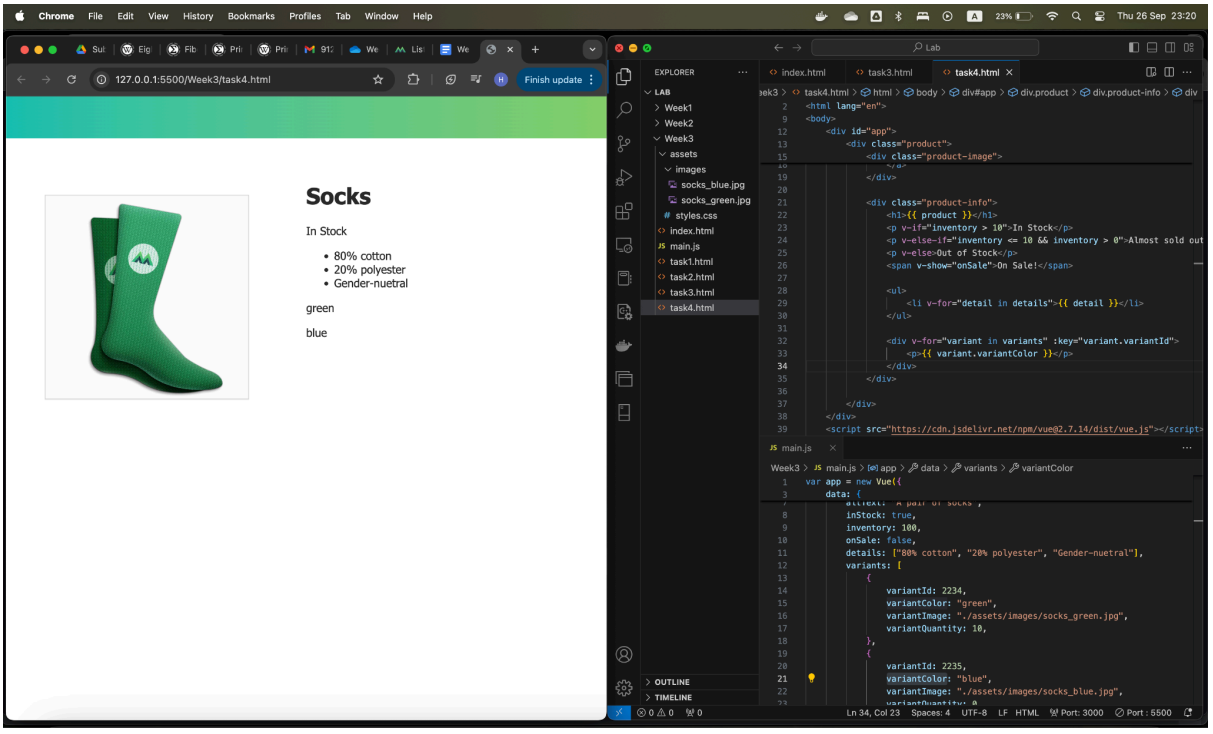


Challenge of lesson 3

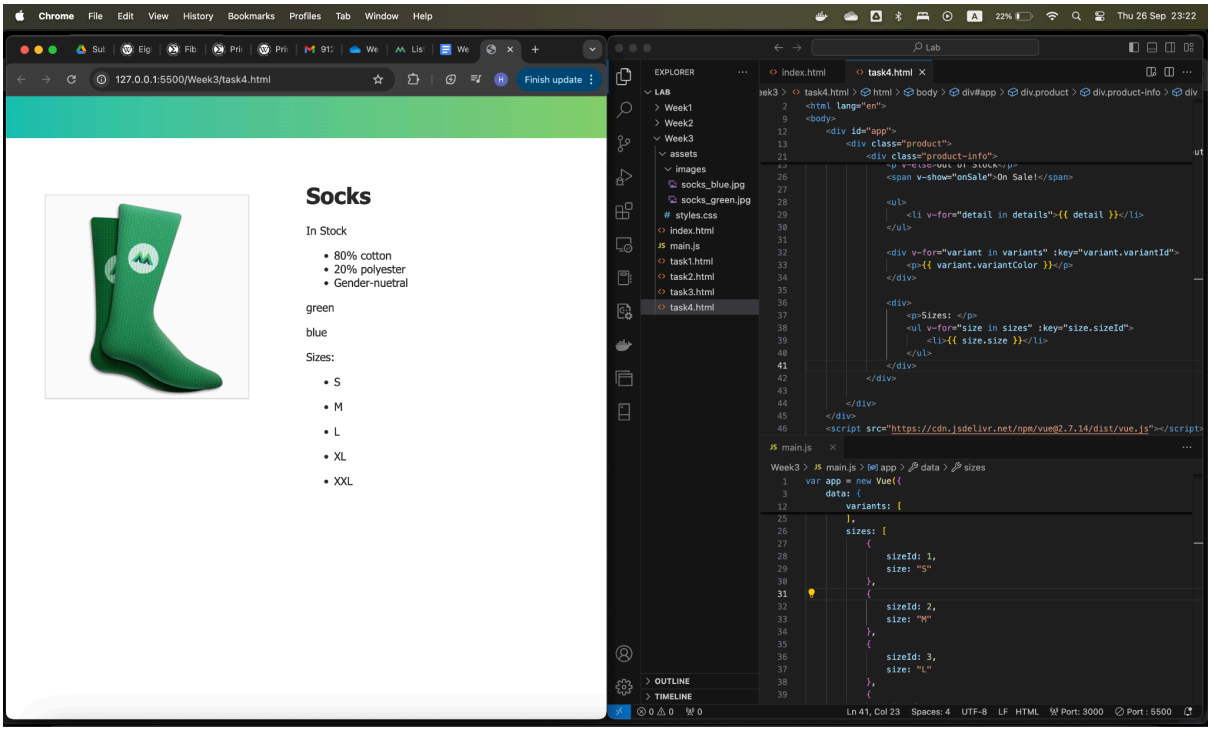




Lesson 4: List Rendering



Challenge of lesson 4

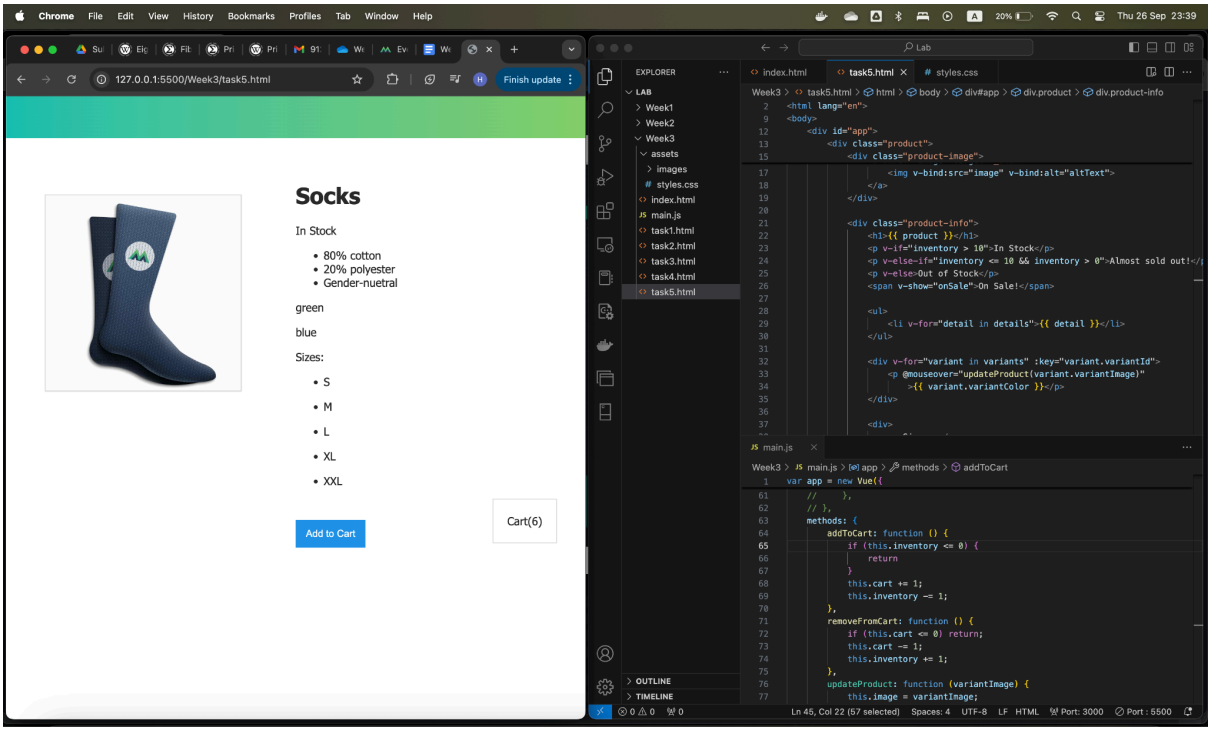


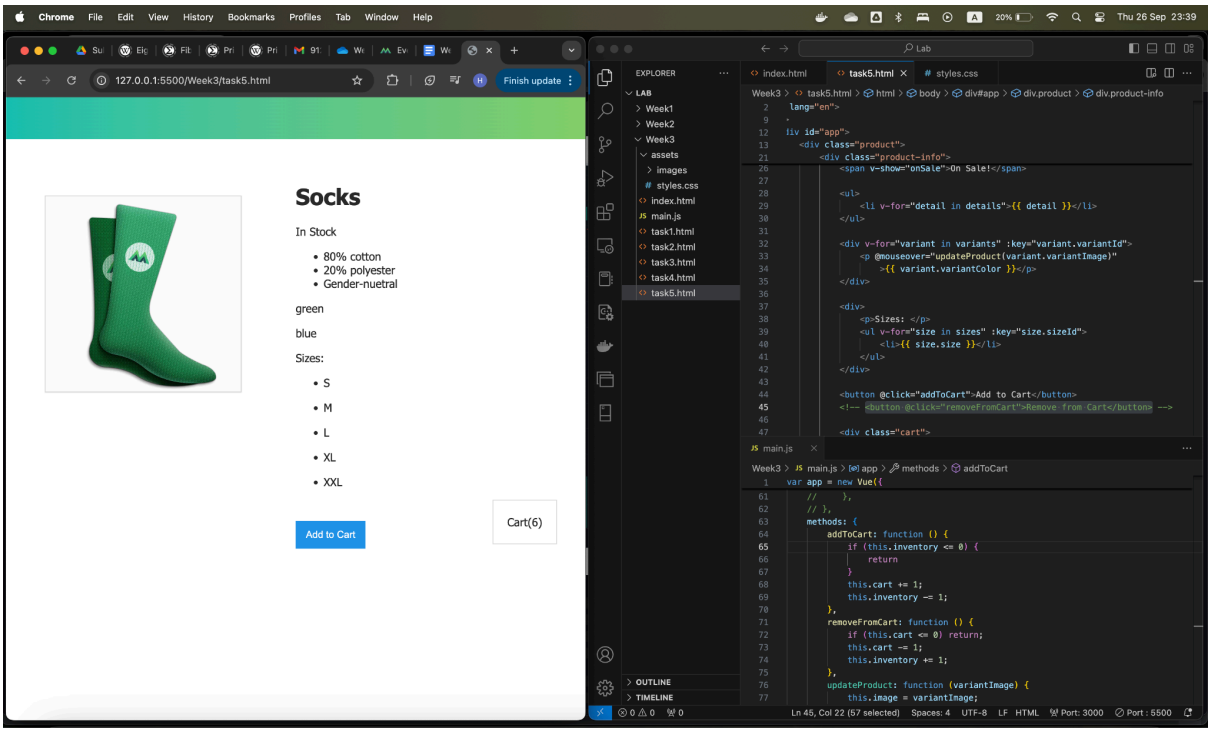
Evaluation

From lesson 1, I understood how to create a Vue instance for managing a Vue app and HTML elements rendered by the VueJS framework. In lesson 2, I have just learned how to bind data to html tags to create dynamic data to display on the html; in addition, Vue instance has a place to define data properties, which helps developers to manage data easily. For lesson 3, I understood how to use conditional statements to render HTML tags according to conditional logic. This statement can assist in rendering only necessary HTML tags for business logic instead of many HTML tags. Finally, list rendering is a helpful feature that helps developers create list data with a for loop instead of creating many duplicate HTML tags manually.

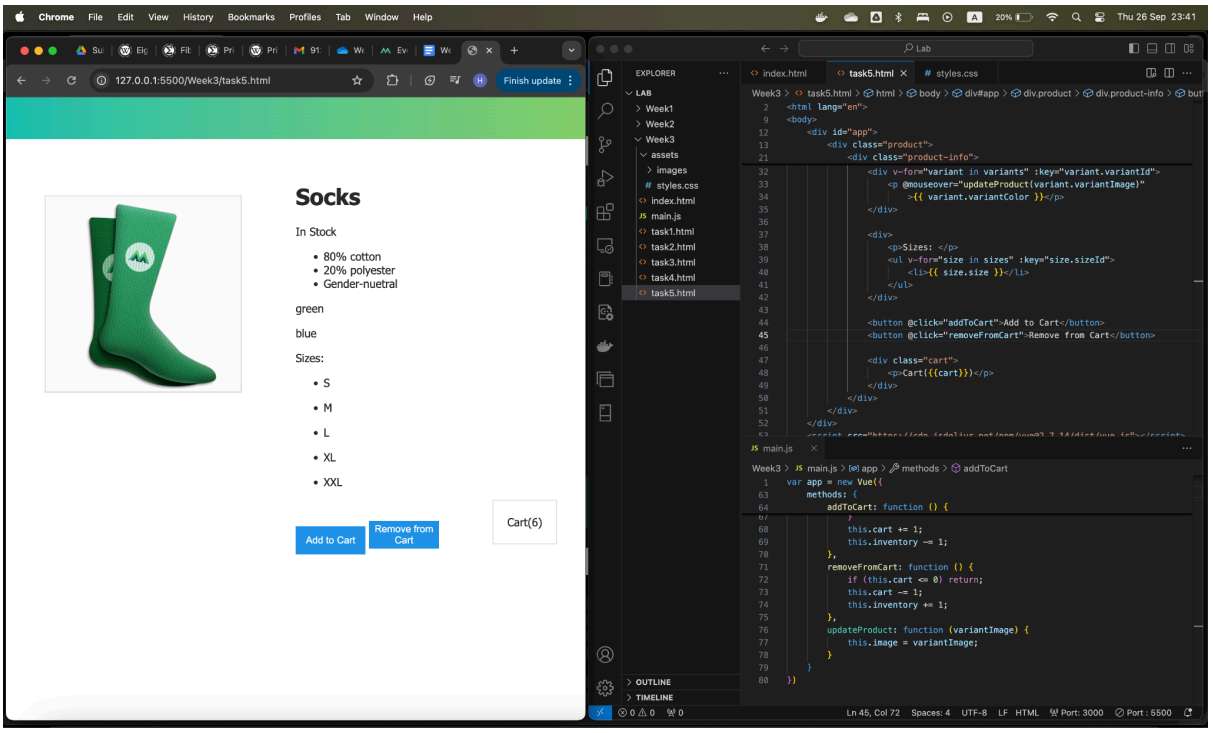
Task 2 (Lesson 5-7)

Lesson 5: Event Handling

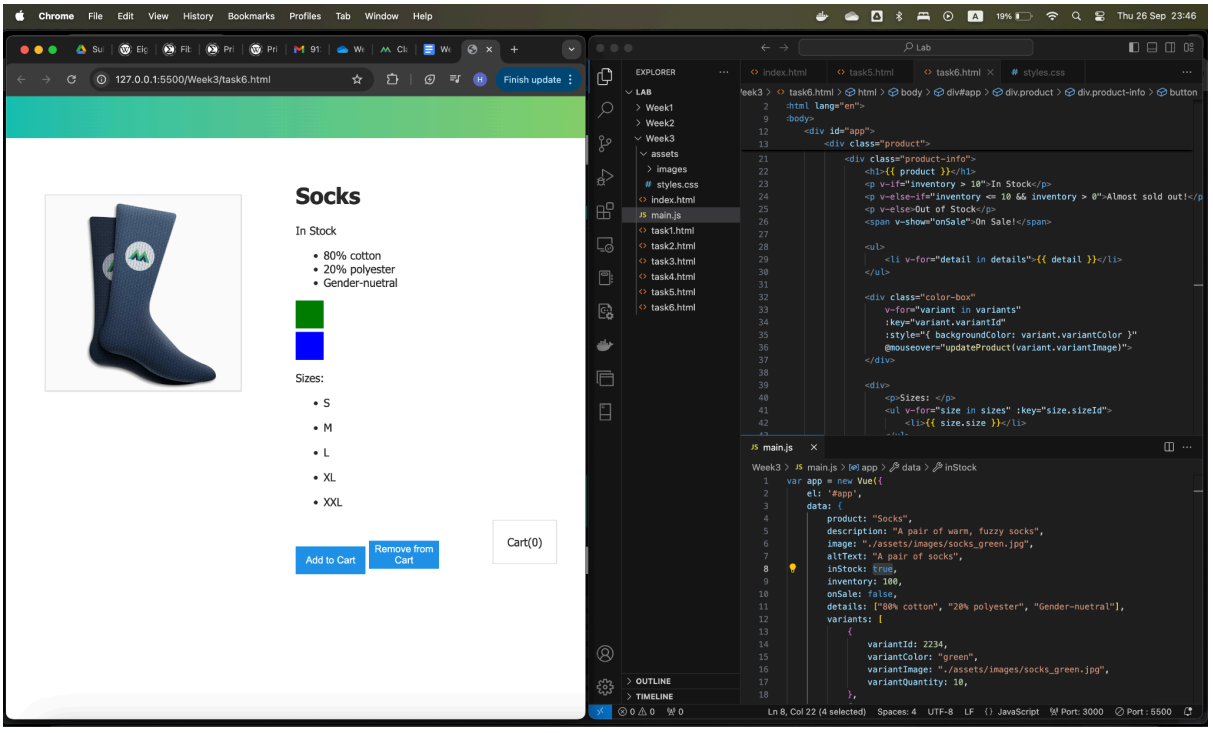




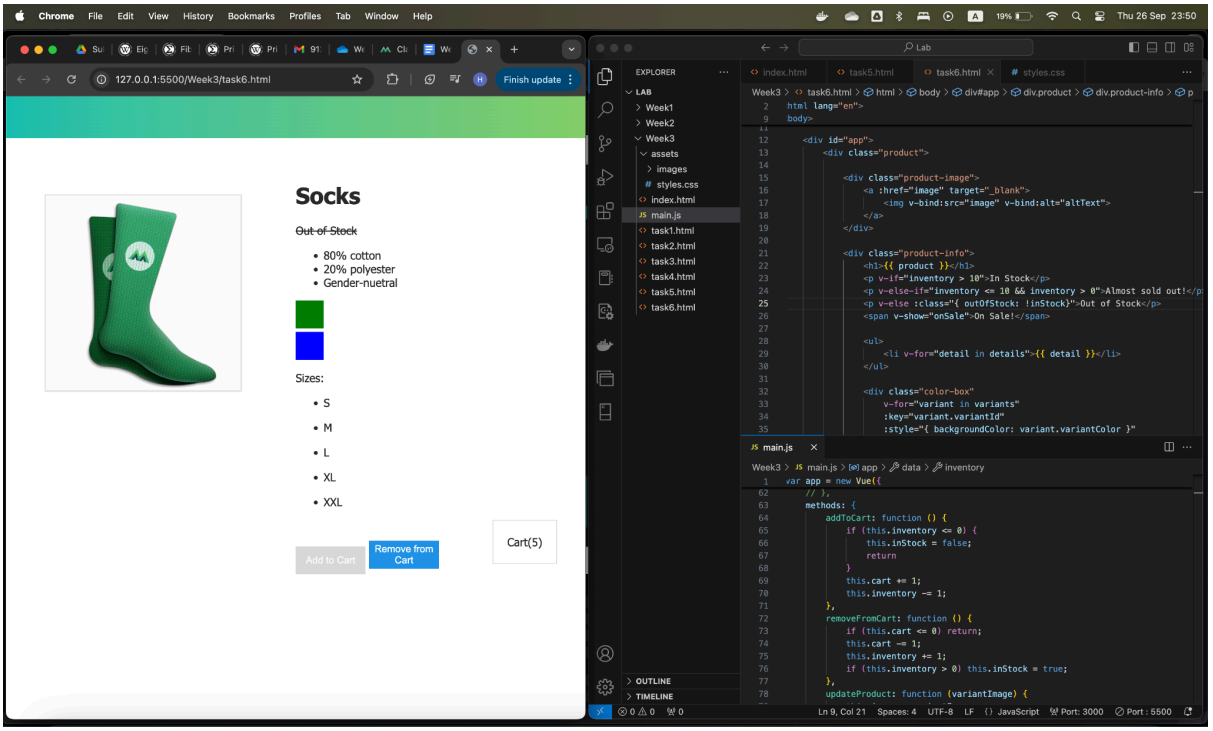
Challenge of lesson 5



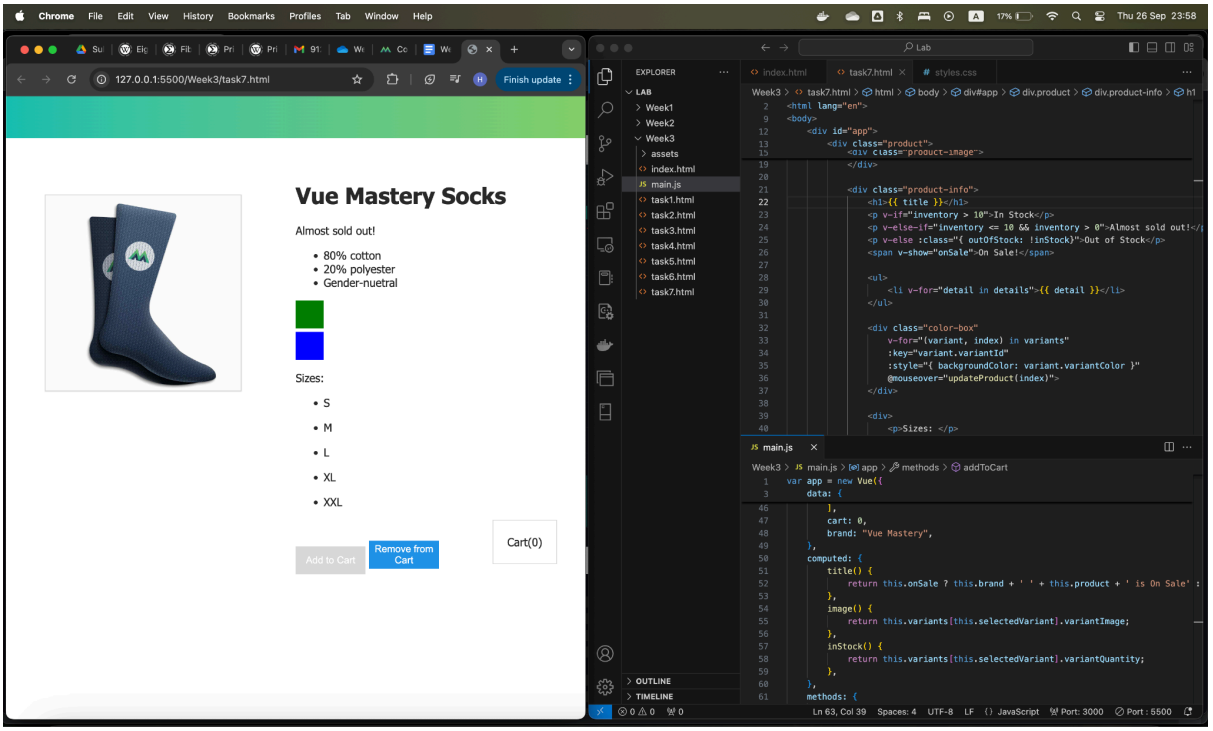
Lesson 6: Class & Style Binding



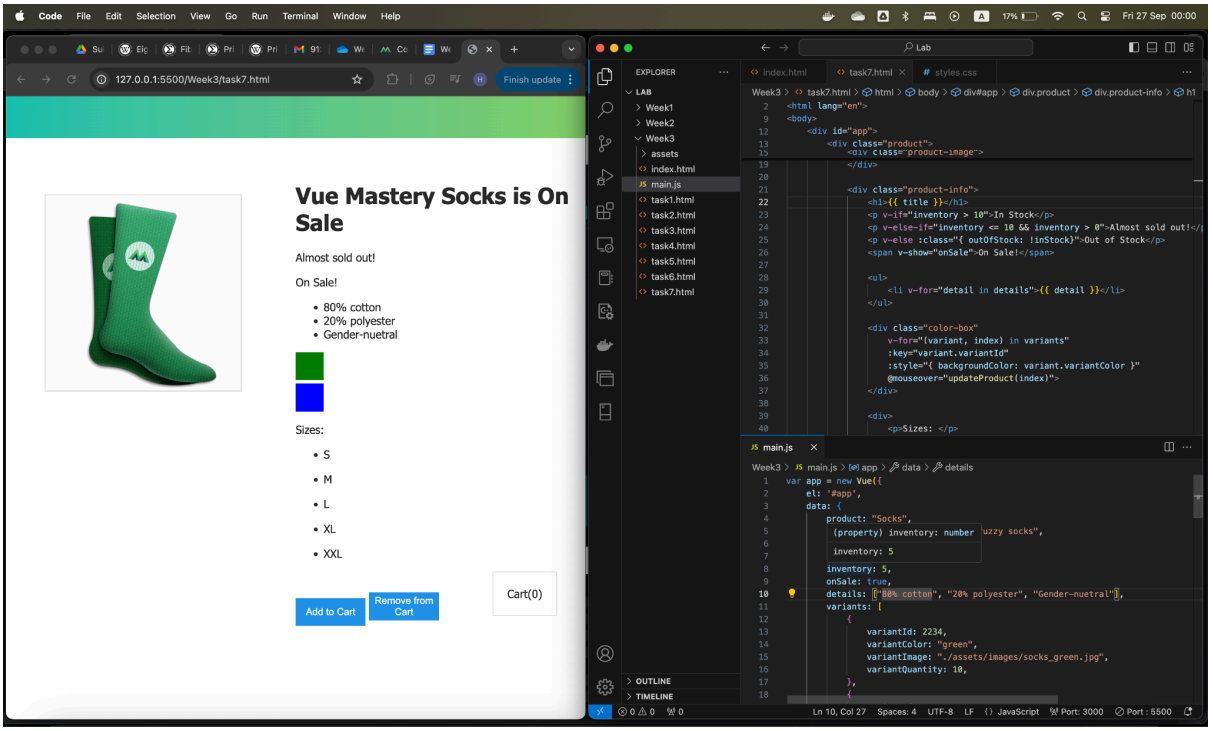
Challenge of lesson 6



Lesson 7: Computed Properties



Challenge of lesson 7

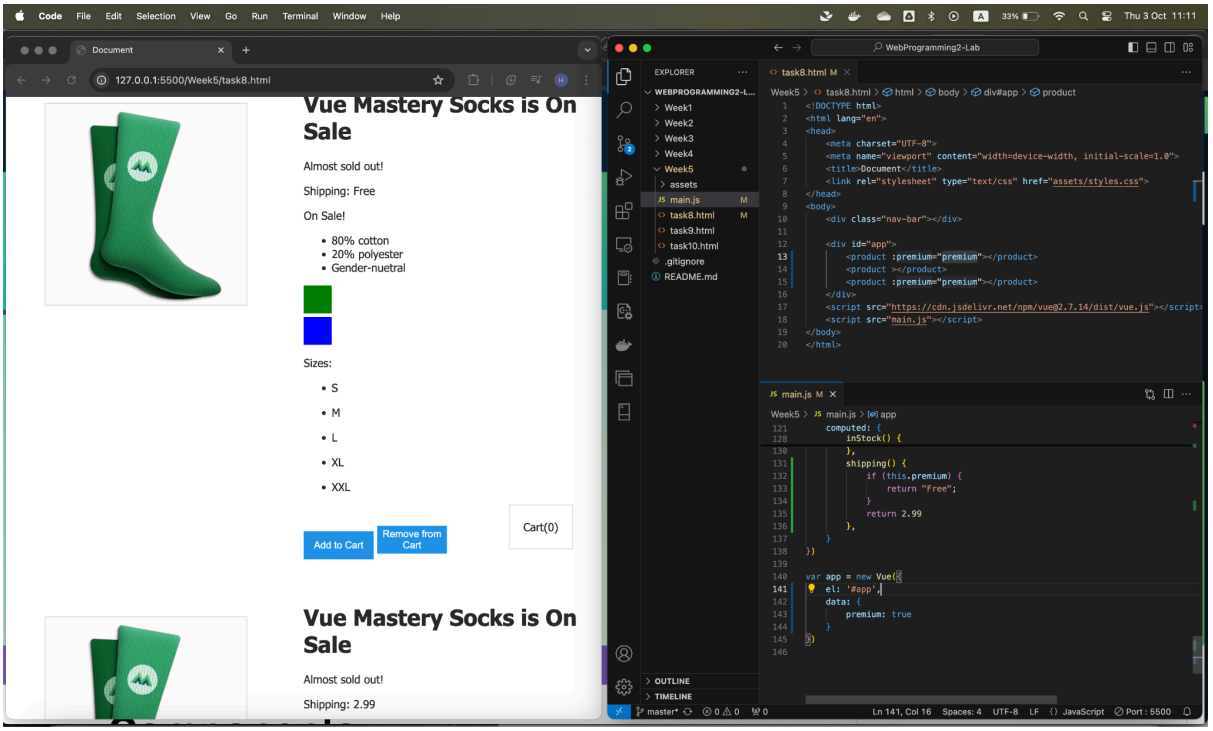


Evaluation

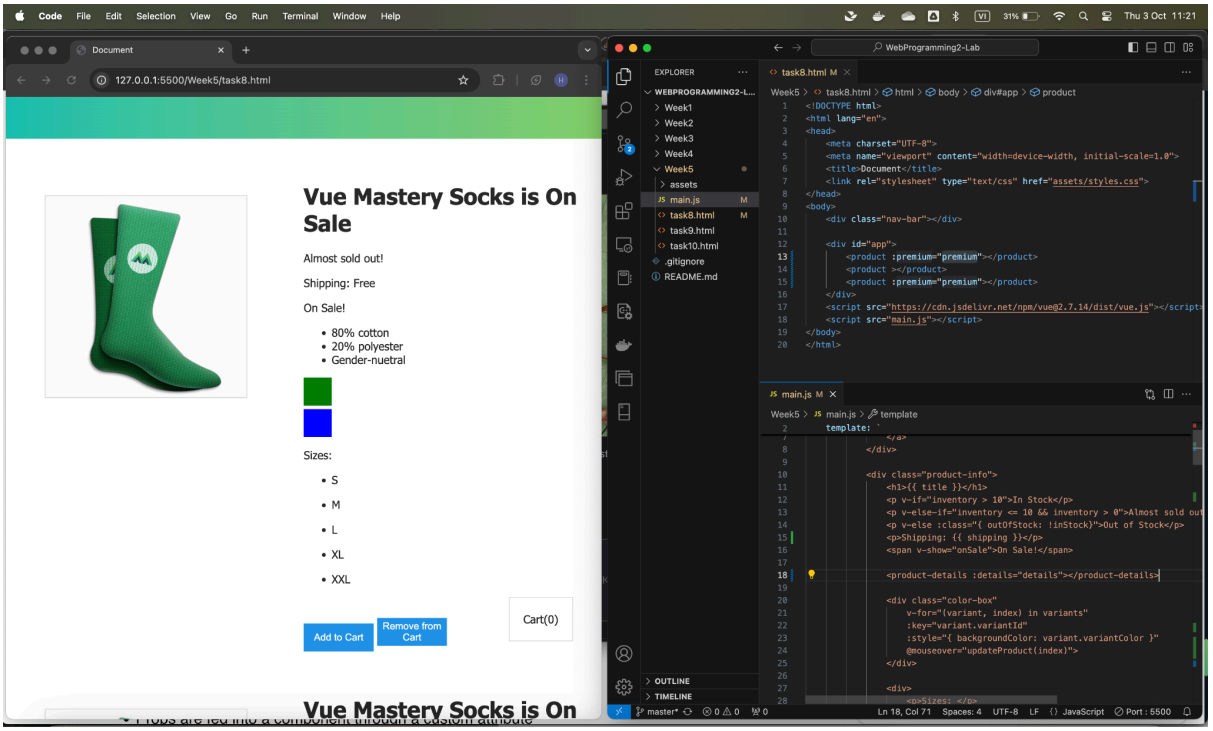
In lesson 5, I learned how to use event handling effectively with VueJS. With this mechanism, it helps me to manage events better. For example, I define an event listener on the HTML tags, then I create an event handler in the methods property of the Vue instance to handle incoming events from the HTML tags. It helps manage the event signal mechanism efficiently. Moreover, class and style binding help to add CSS dynamically following data. For example, I can change the background colour of the colour box following the data of socks. With the computed properties, it helps to create computed data based on other data dynamically; in addition, it can help to save memory in creating many variables to store data. Finally, lessons 5–7 help understand more about event handling in processing user interactions.

Task 3 (Lesson 8 - 10)

Lesson 8: Components

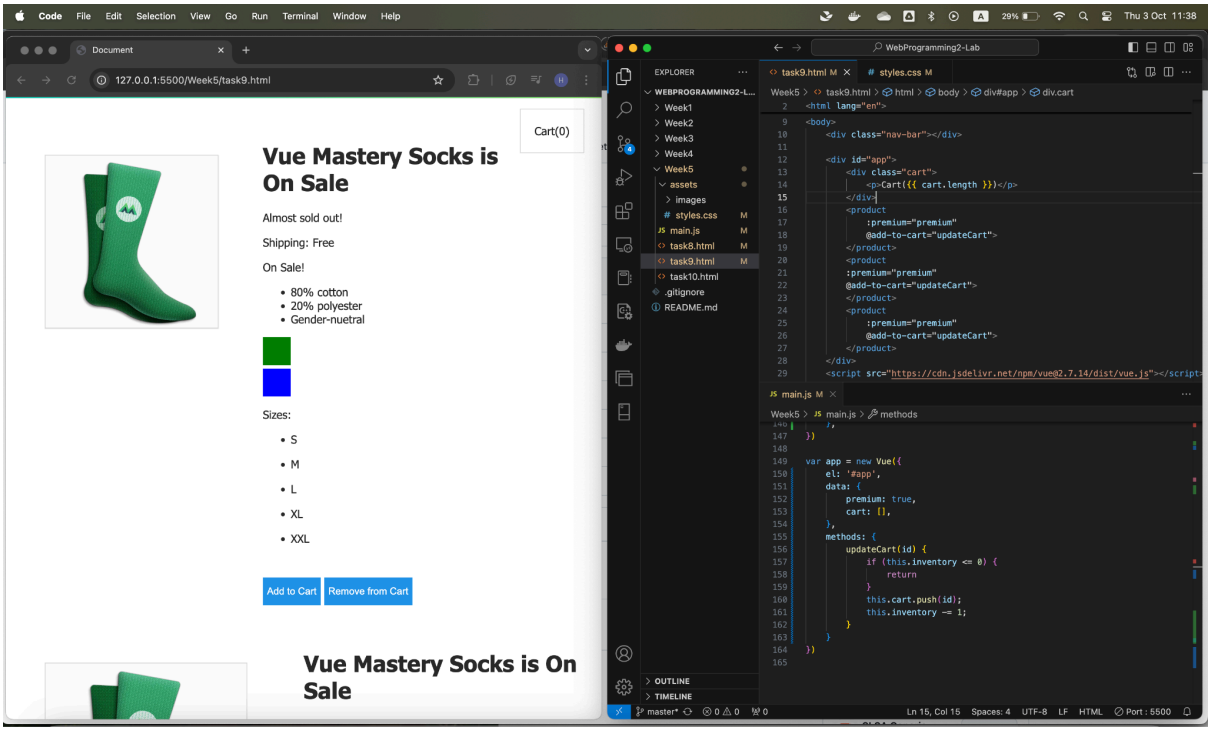


Challenge of lesson 8

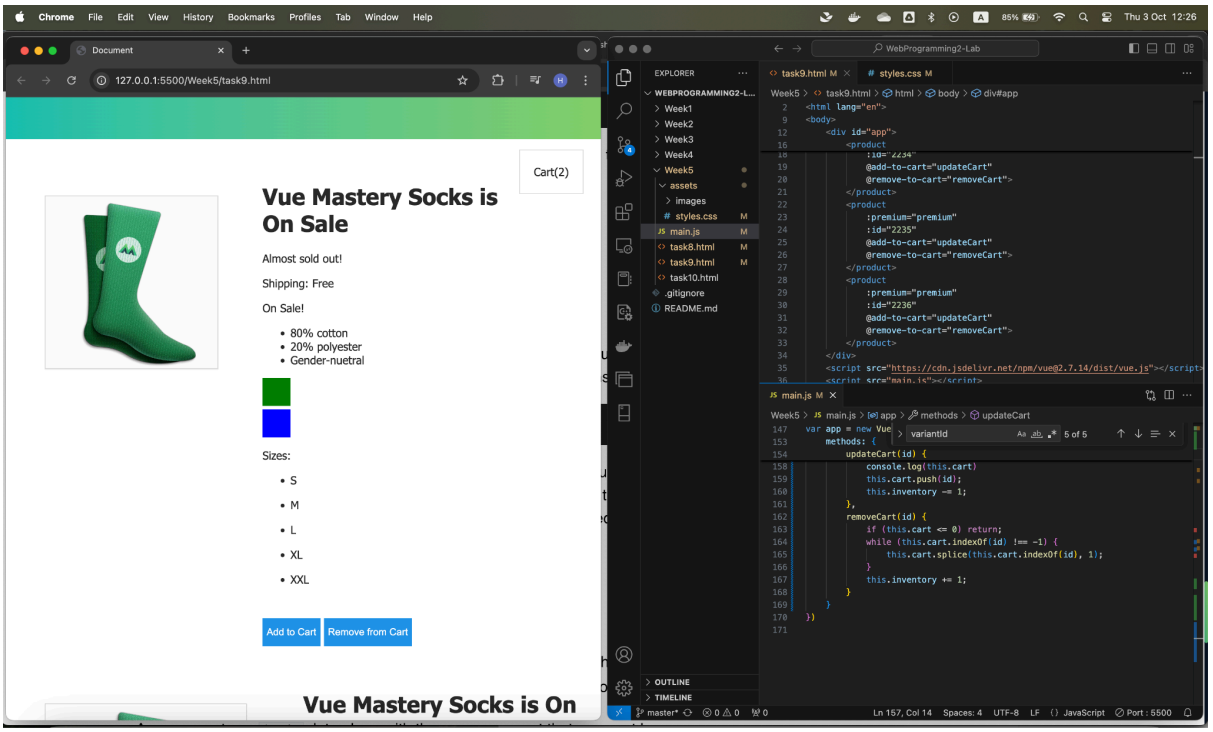


Evaluation

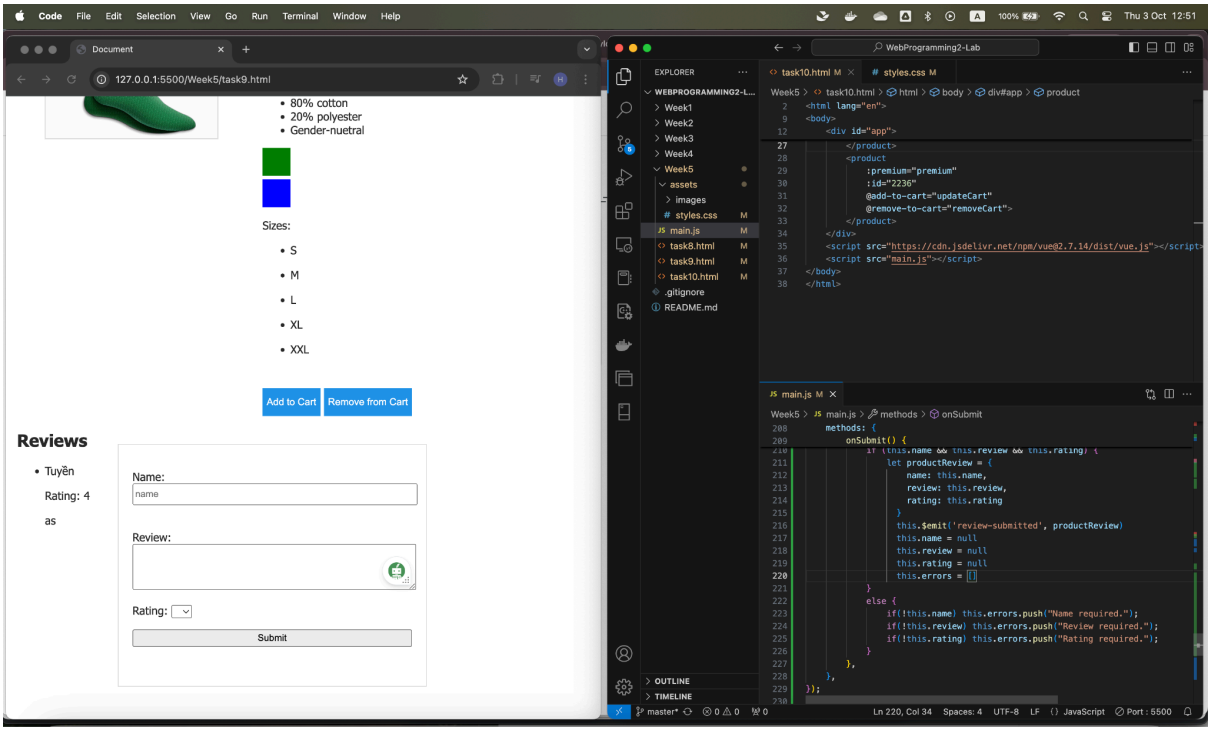
Lesson 9: Communicating Event



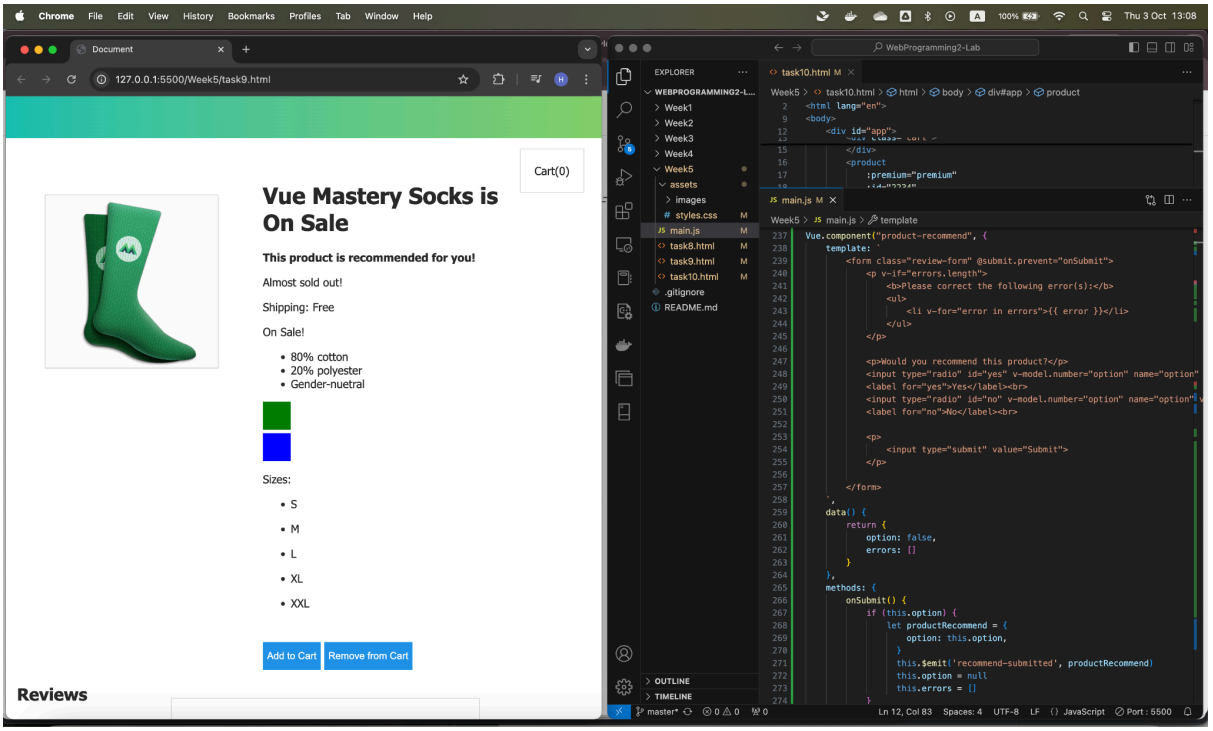
Challenge of lesson 9:

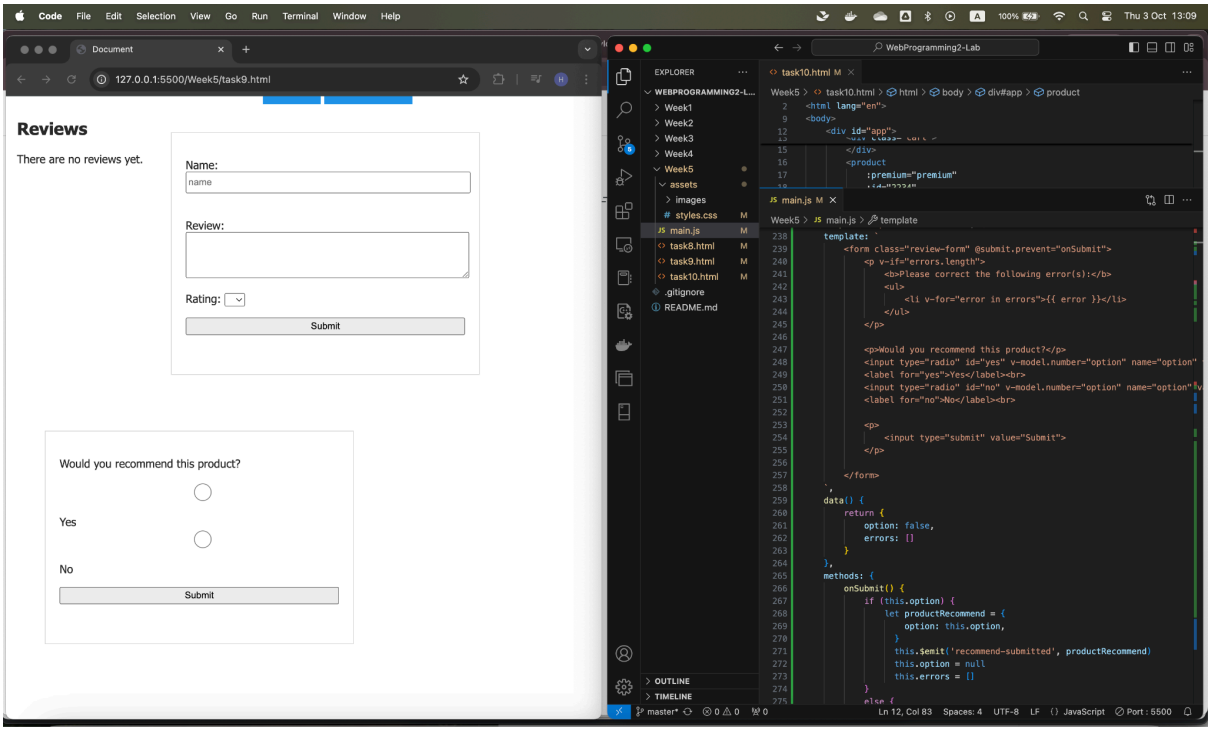


Lesson 10: Forms & v-model



Challenge of lesson 10





Evaluation

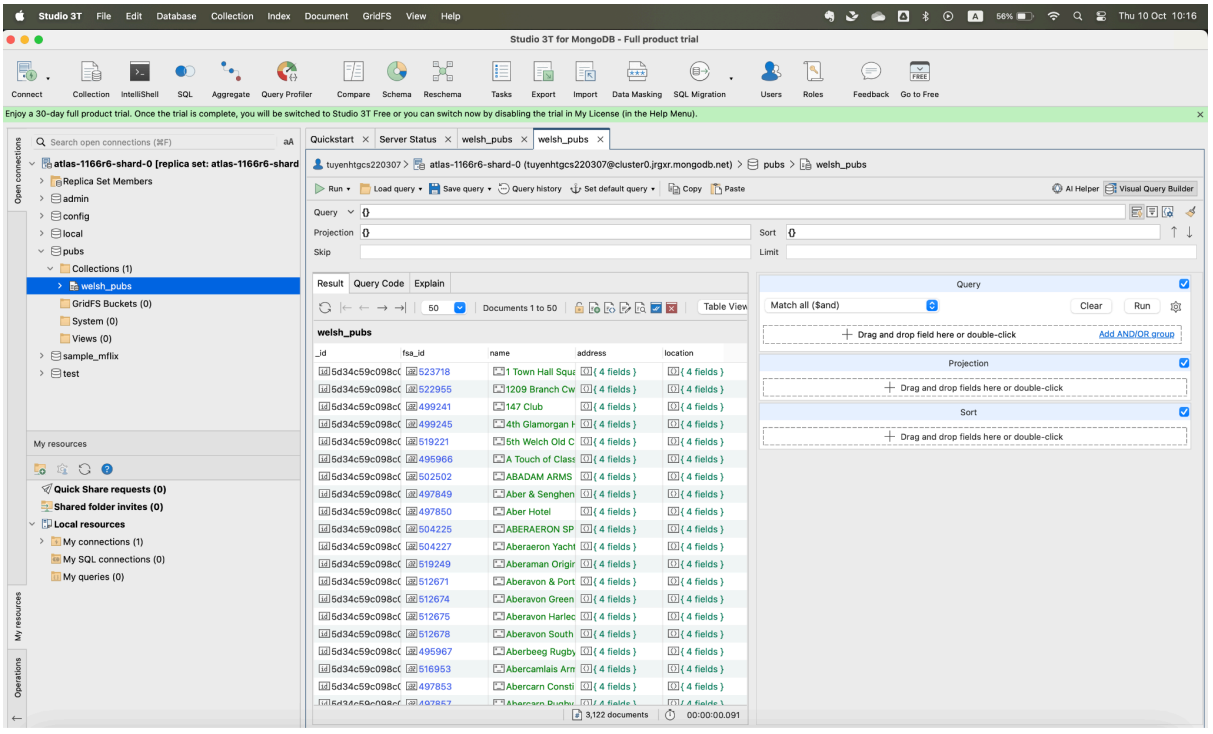
In lesson 8, I learned how to create and use Vue components. Components help break down the user interface into smaller pieces, making the app more manageable. For example, I can pass data to these

components using props, allowing different parts of the app to communicate with each other. This helps manage the UIs by dividing them into sections with their own functionality. In lesson 9, I learned about communicating events between components. I used $emit to send custom events from child to parent components, allowing the parent to listen for these events and respond. In lesson 10, I learned how to work with forms in Vue using the v-model directive for two-way data binding. For example, I could bind input fields directly to data in the Vue instance. Additionally, I learned how to handle form submissions and perform validation making data management more efficient when processing user inputs.These lessons deepen my understanding of building interactive Vue.js applications with components, communication between components, and handling user input.

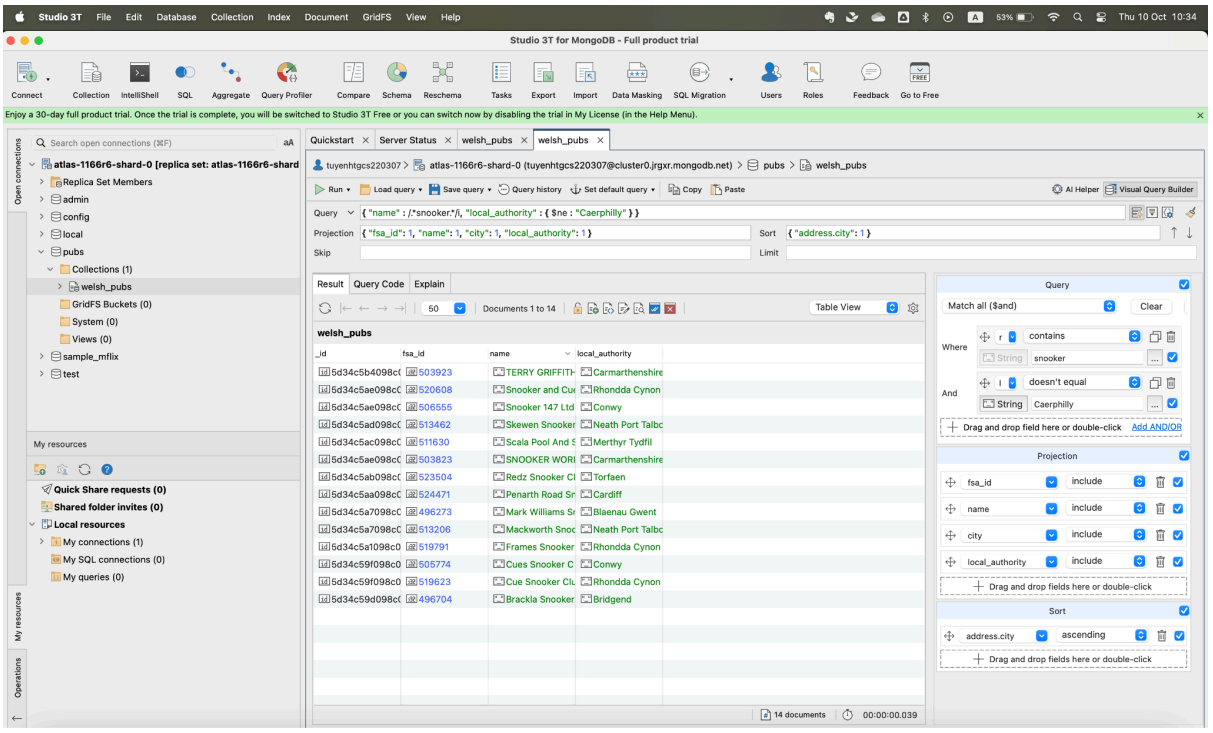
Tutorial 8 - Week 6

Task 2

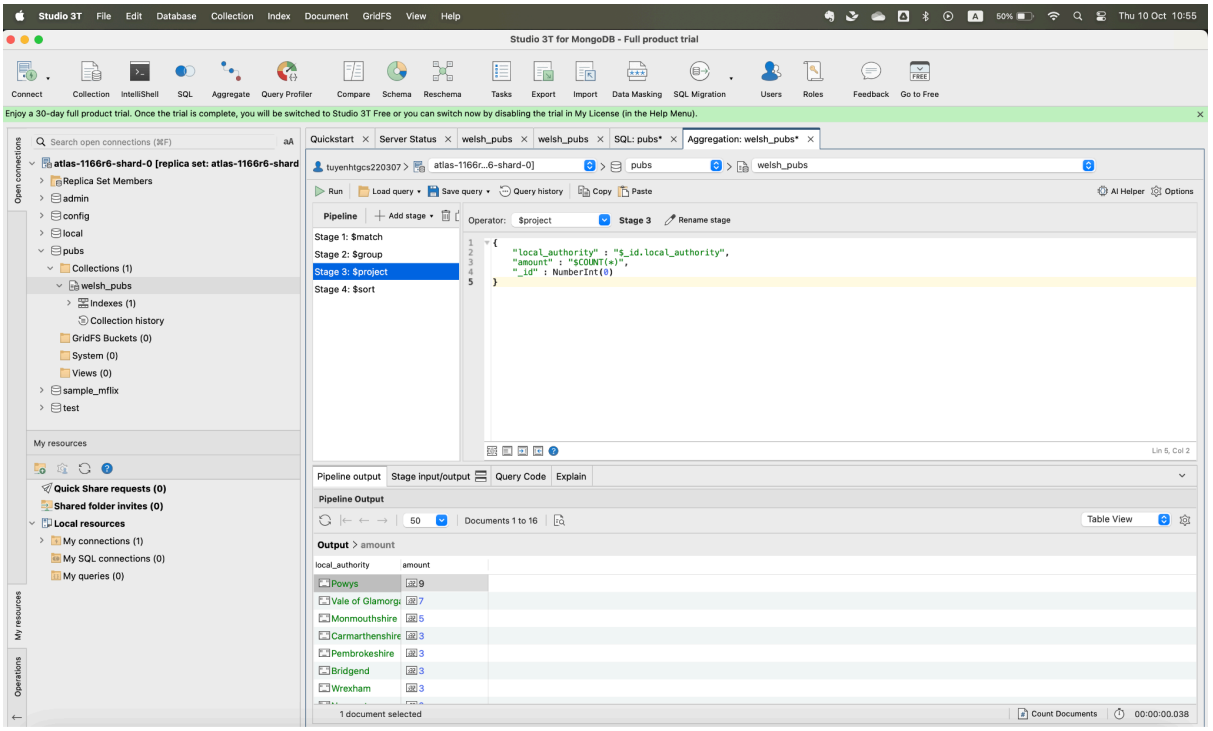
End of lesson 4 exercise 1



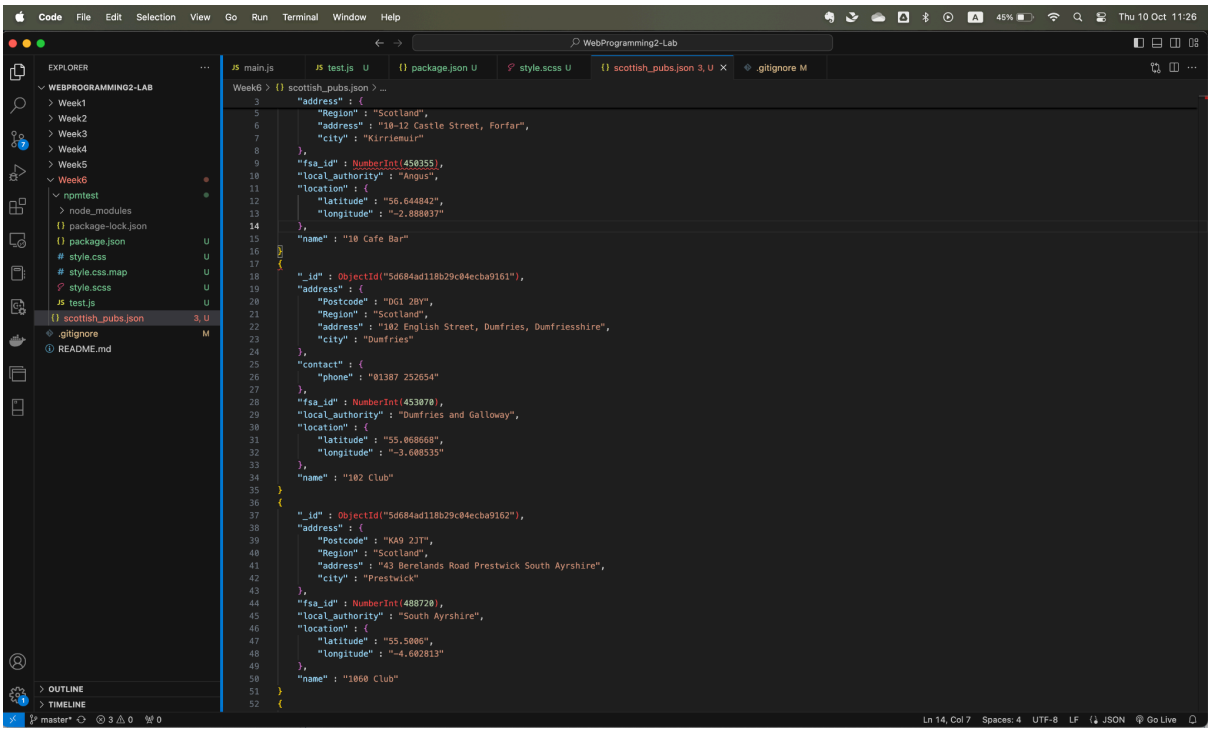
End of lesson 4 exercise 3



End of lesson 5 exercise 3



End of lesson 6 exercise 2



Conclusion

In the fourth session, MongoDB and Mongo Atlas offered a cloud database that was straightforward to use and study for novice users. Additionally, it offers security configurations like network and database access. In addition, the tutorial offers fundamental understanding of database, collection, and document management. In addition, MongoDB offers a solid query build that includes software development kits for Python, Node.js, Ruby, and several additional languages, as well as SQL statements like SQL databases and mongoshells. Lastly, it shares a significant benefit with NoSQL in that collections may be created dynamically without a database schema being predefined.

MongoDB offers a wide range of capabilities for aggregating data across multiple programming languages, as demonstrated in lesson 5. Furthermore, Studio3T offers a visual aggregate editor for data analysis. Because Mongo offers a pipeline for aggregating data, developers are able to set up a comprehensive pipeline that includes query, group, project, and sorting capabilities.

MongoDB offers strong features in lesson 6 that make it simple to import and export data from data sources. Developers may effortlessly import data into collections using sizable CSV and JSON files. Furthermore, Mongo makes data exporting effortless. Instead of exporting all of the data from the database, developers can export data from queries using studio3T.

Appendix

Git Link

https://github.com/huynhthanhtuyen456/WebProgramming2-Lab