Report

MOOC Report

Course name: Data Structures and Algorithms

CSC10004_23CLC09

Students:
Nguyen Le Ho Anh Khoa -
23127211

Teacher:
Bui Duy Dang
Truong Tan Khoa
Nguyen Thanh Tinh

August 13, 2024

Contents

1 Student Information

Class: 23CLC09
Student ID: 23127211
Full name: Nguyen Le Ho Anh Khoa

2 Summary of the courses

2.1 Algorithm Toolbox — University of California San Diego — Cousera
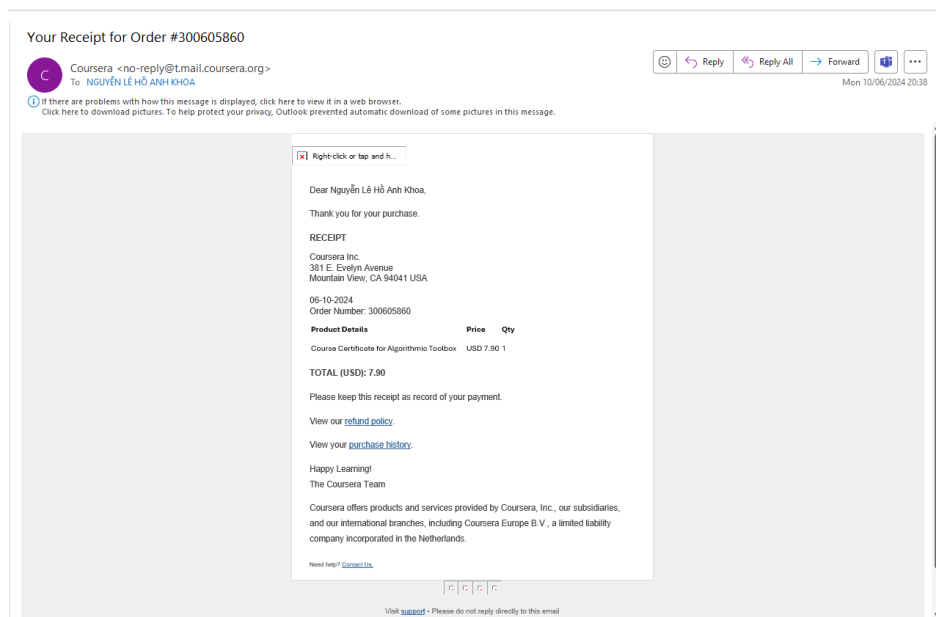

Figure 1: Algorithm Toolbox Certificate on Cousera

Figure 2: Algorithm Toolbox Course start on Jun 10 2024

Figure 3: Algorithm Toolbox Course completed on August 7 2024

Figure 4: Graded of Algorithm Toolbox Course

2.1.1 Course Introduction

- **Course Name:** Algorithm Toolbox

- **Institution:** University of California San Diego

- **Platform:** Cousera

- **Level:** Intermediate

- **Commitment:** 5 weeks, 4-8 hours per week

- **Instructor:** Pavel Pevzner, Michael Levin

- **Course Link:** Algorithm Toolbox

2.1.2 Course Content

This is Course 1 of 6 in the Data Structures and Algorithms Specialization

I am taking this course to supplement my university course on Data Structures and Algorithms. This online course provides additional practice and deeper insights into algorithmic techniques, which will help me better understand and apply the concepts taught in my university course.

The course content is divided into six modules:

- **Module 1: Introduction: Asymptotic Analysis and Getting Started with Programming Challenges.** In this module, students are introduced to algorithmic thinking and asymptotic analysis. Basic programming challenges are provided to help students get started.

- **Module 2: Algorithmic Warm-up: Fibonacci Numbers and the Greatest Common Divisor.** This module covers efficient computation of Fibonacci numbers and algorithms for finding the greatest common divisor (GCD).

- **Module 3: Greedy Algorithms: Minimum Spanning Trees, Clustering, Huffman Codes.** This module I learn about greedy algorithms and their applications in minimum spanning trees, clustering, and Huffman coding for data compression.

- **Module 4: Divide and Conquer: Sorting and Searching, QuickSort, Binary Search Trees.** This module introduces the divide and conquer strategy, sorting algorithms like Quick-Sort, and binary search trees for efficient searching.

- **Module 5: Dynamic Programming: Edit Distance, Knapsack.** I am introduced to dynamic programming, solving the edit distance problem, and the knapsack problem and its variations.

- **Module 6: Dynamic Programming 2: Advanced Topics, Longest Common Subsequence, Overlapping Subproblems.** This module covers advanced topics in dynamic programming, the longest common subsequence problem, and handling overlapping subproblems.

2.2 Data Structures — University of California San Diego — Cousera
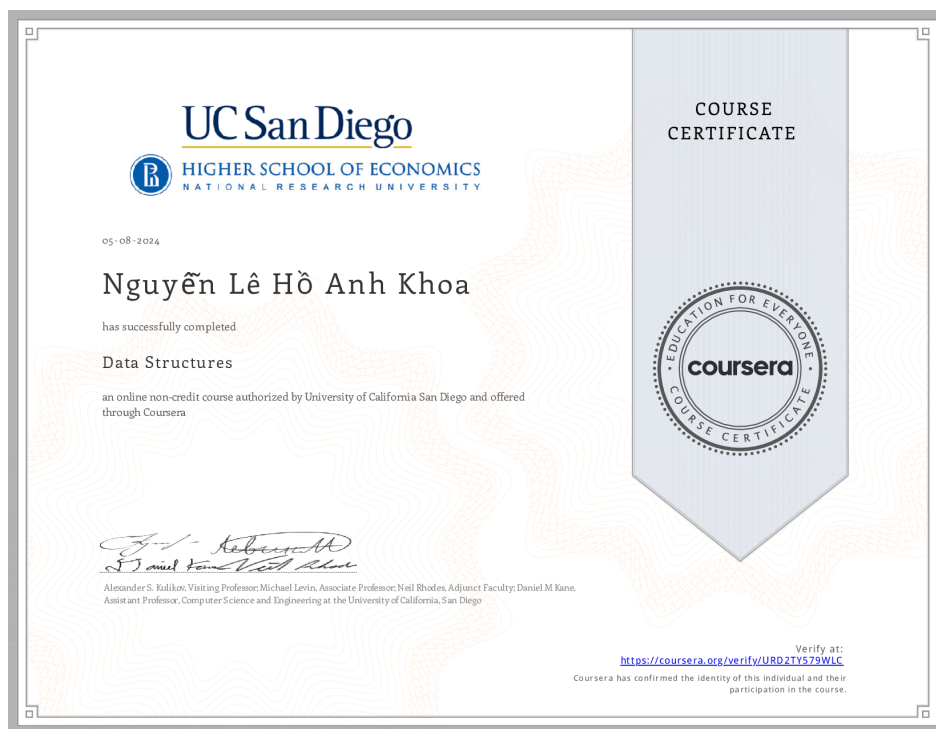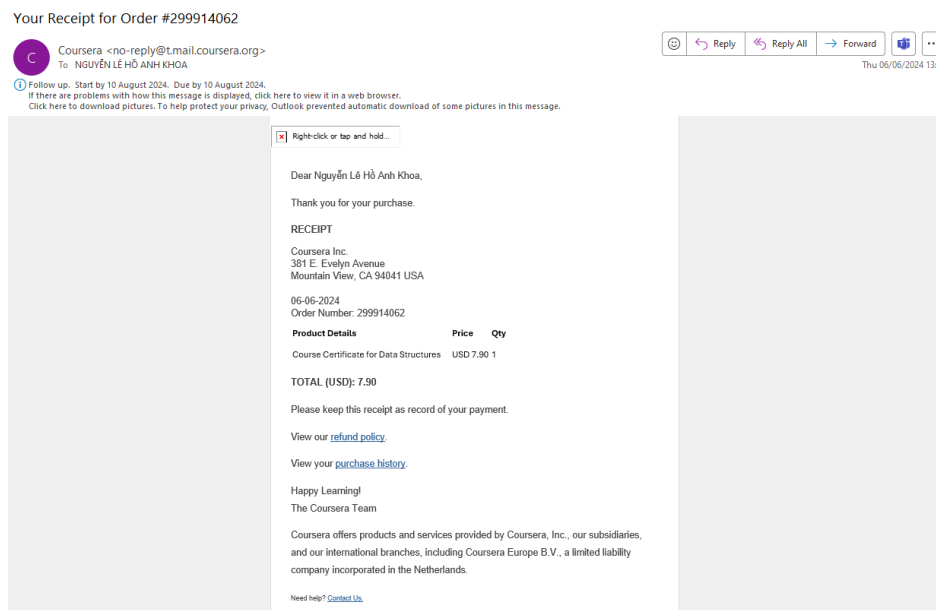

Figure 5: Data Structures Certificate on Cousera

Figure 6: Data Structures Course start on Jun 6 2024

Figure 7: Data Structures Course completed on August 5 2024

Figure 8: Graded of Data Structures Course

2.2.1 Course Introduction

- **Course Name:** Data Structures

- **Institution:** University of California San Diego

- **Platform:** Cousera

- **Level:** Intermediate

- **Commitment:** 5 weeks, 4-8 hours per week

- **Instructor:** Pavel Pevzner, Michael Levin

- **Course Link:** Data Structures

2.2.2 Course Content

This is Course 2 of 6 in the Data Structures and Algorithms Specialization.

This is an opportunity for me to consolidate my knowledge at school. By engaging in this process, I can reinforce the concepts and skills I have learned, ensuring a deeper understanding and retention of the material. Additionally, this consolidation will help me identify any gaps in my knowledge, allowing me to address them proactively. Ultimately, this effort will contribute to my overall academic success and prepare me for future challenges.

The course content is divided into six modules:

- **Module 1: Basic Data Structures: Arrays, Linked Lists, Stacks, Queues.** In this module, I am introduced to basic data structures including arrays, linked lists, stacks, and queues.

- **Module 2: Trees: Binary Search Trees, AVL Trees, Splay Trees.** This module covers binary search trees and their properties, as well as balanced trees like AVL and Splay trees.

- **Module 3: Hash Tables: Hash Functions, Collisions, Rabin-Karp Algorithm.** Students learn about hash tables and hash functions, handling collisions, and the Rabin-Karp algorithm.

- **Module 4: Priority Queues: Heaps, Heap Sort, Balanced Binary Heaps.** This module introduces priority queues and heaps, including heap sort and balanced binary heaps.

- **Module 5: Hash Tables 2: Open Addressing, Cryptographic Hashing.** This module covers open addressing in hash tables and cryptographic hashing techniques.

- **Module 6: Binary Search Trees: Splay Trees, AVL Trees, Deletion, Randomized BSTs.** This module delves into advanced topics in binary search trees, including deletion operations and randomized binary search trees.

2.3 HTML, CSS, JavaScript, React - Online Certification Course — Udemy
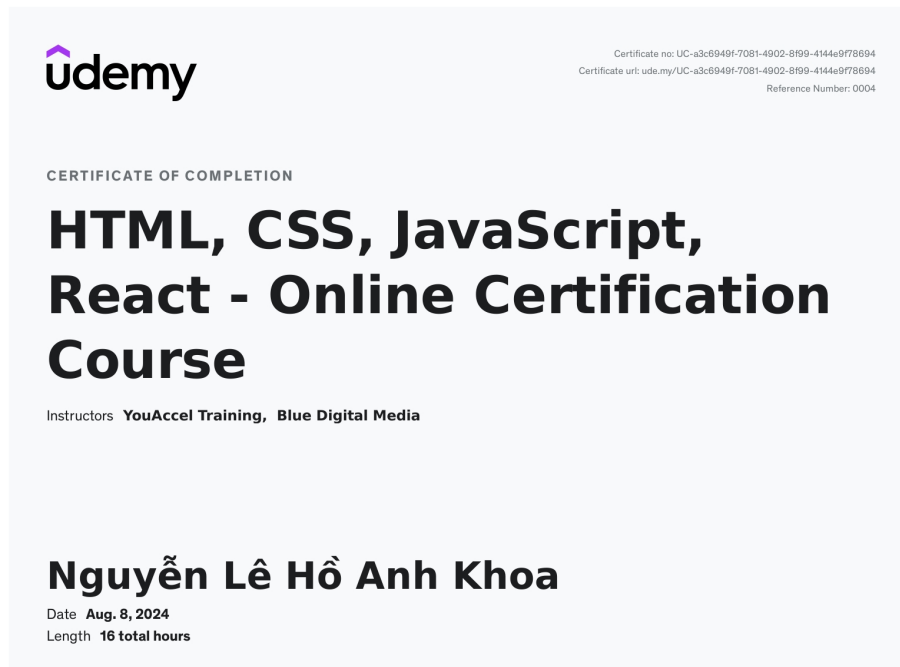
Figure 9: HTML, CSS, JavaScript, React Certificate on Udemy completed on August 8 2024

Figure 10: HTML, CSS, JavaScript, React - Online Certification Course start on July 18 2024

2.3.1 Course Introduction

- **Course Name:** HTML, CSS, JavaScript, React - Online Certification Course

- **Platform:** Udemy

- **Level:** Beginner

- **Commitment:** 4 weeks, 4-6 hours per week

- **Instructor:** YouAccel Training, Blue Digital Media

- **Course Link:** HTML, CSS, JavaScript, React - Online Certification Course

2.3.2 Course Content

This course is designed to provide a comprehensive introduction to web development, covering HTML, CSS, JavaScript, and React. By the end of the course, I will have a solid foundation in front-end web development and be able to create interactive and responsive web applications.

Udemy is a popular online learning platform that offers a wide range of courses on various topics. The HTML, CSS, JavaScript, React course is designed for beginners who want to learn web development from scratch. The course covers the basics of HTML, CSS, and JavaScript, as well as more advanced topics like React.

Udemy don't have system graded like Cousera, so I captured some exercises I did.

The course content is divided into four main contents:

- **Content 1: HTML.** In this content, I learn the basics of HTML, including tags, attributes, and elements. I also learn how to create a simple sales website using HTML.
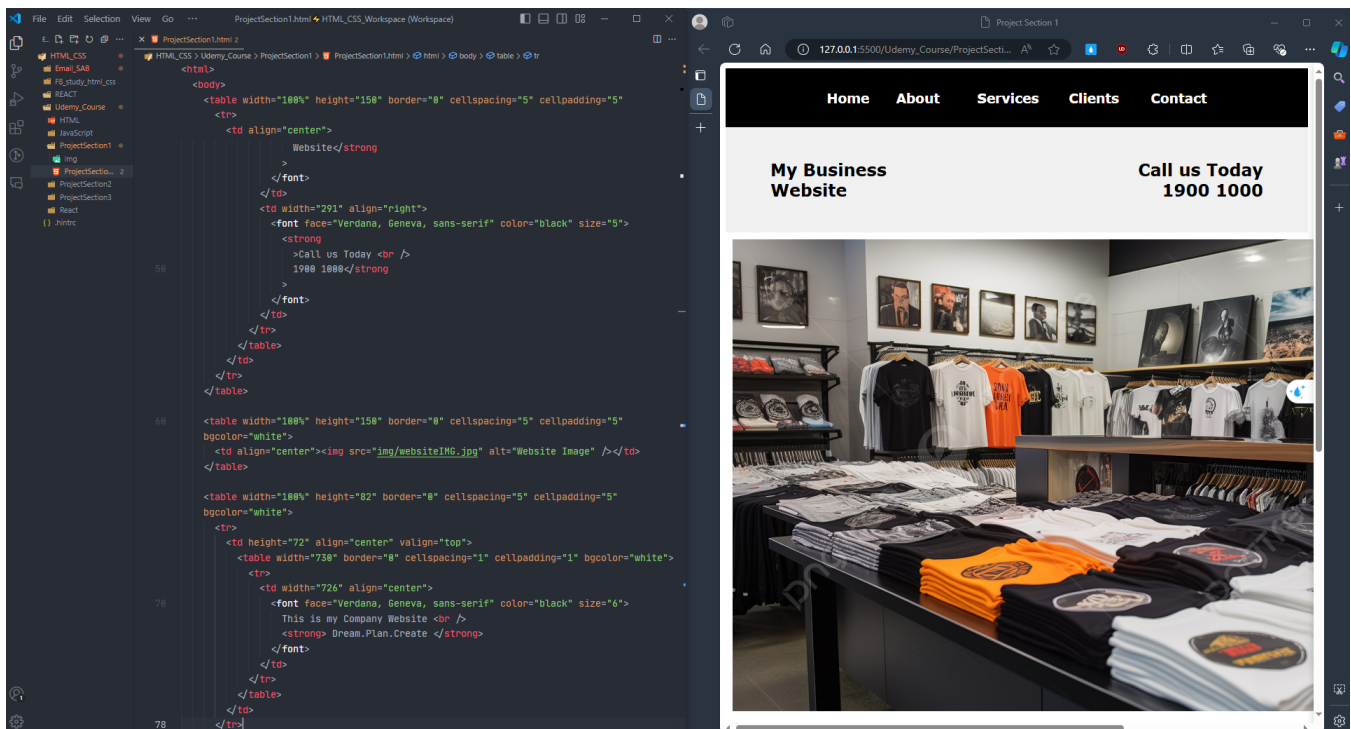
Figure 11: A homepage of sales website by HTML

- **Content 2: CSS.** This content covers CSS fundamentals, including selectors, properties, and values. I learn how to style the sales website created in the previous module using CSS.
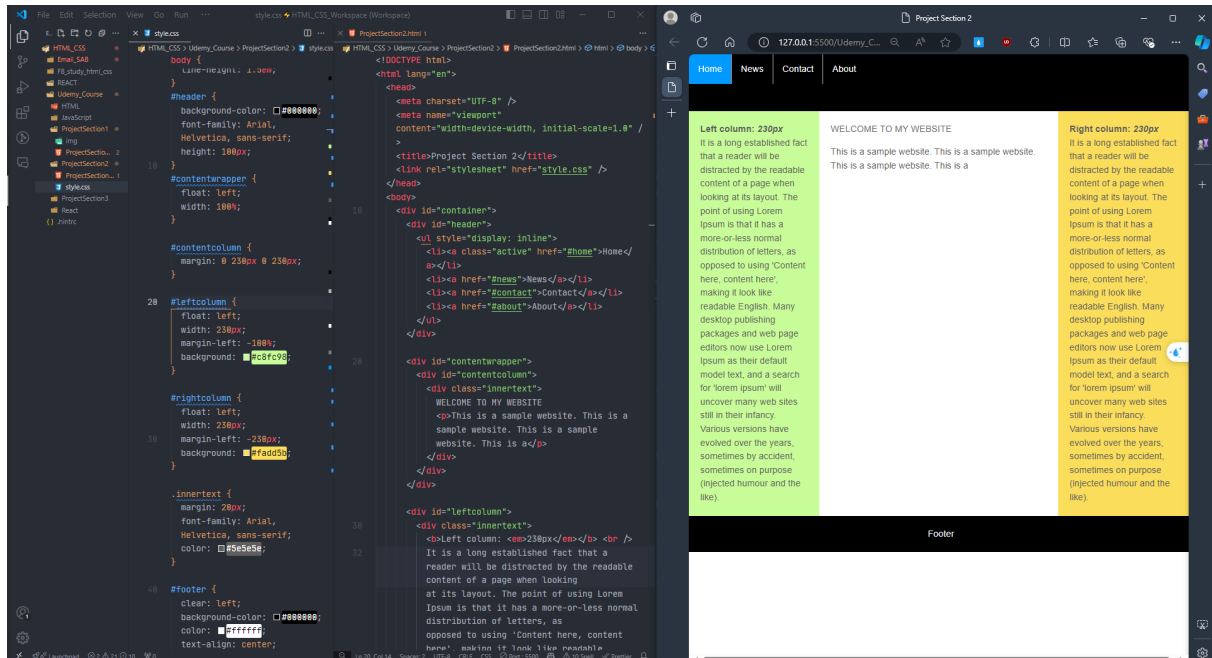

Figure 12: A demo homepage of a blog website by HTML and CSS

- **Content 3: JavaScript.** This content covers JavaScript fundamentals, including variables, data types, functions, and control structures.
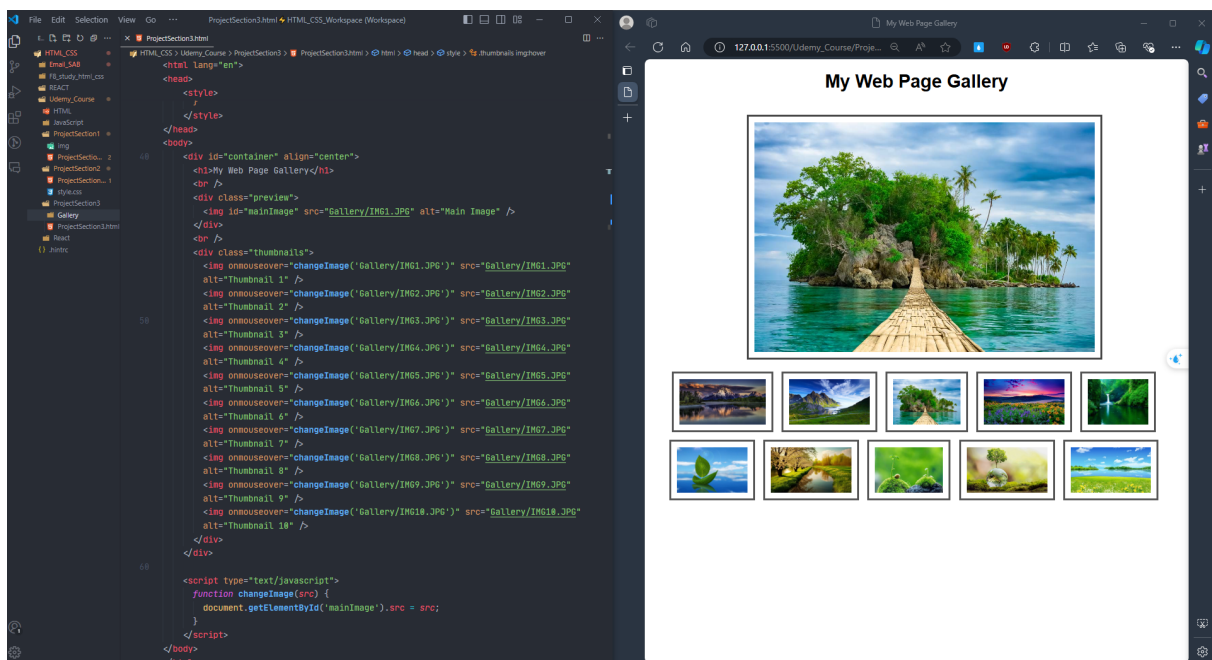
Figure 13: A simple gallery website by HTML, CSS, and JavaScript (The main image will enlarge if you hover over it)

- **Content 4: React.** I am introduced to React, a popular JavaScript library for building user interfaces. I learn how to create components, manage state, and handle events in React applications. This is a simple calculator application built with React. It can perform basic arithmetic operations like addition, subtraction, multiplication, and division. It also has a clear button to clear the input field. Because this is the first project I've done, so inevitably copy existing source code to learn and understand how it works. I will try to understand the code and make some changes to it.

Figure 14: A simple calculator app by React

3   Self - Summary

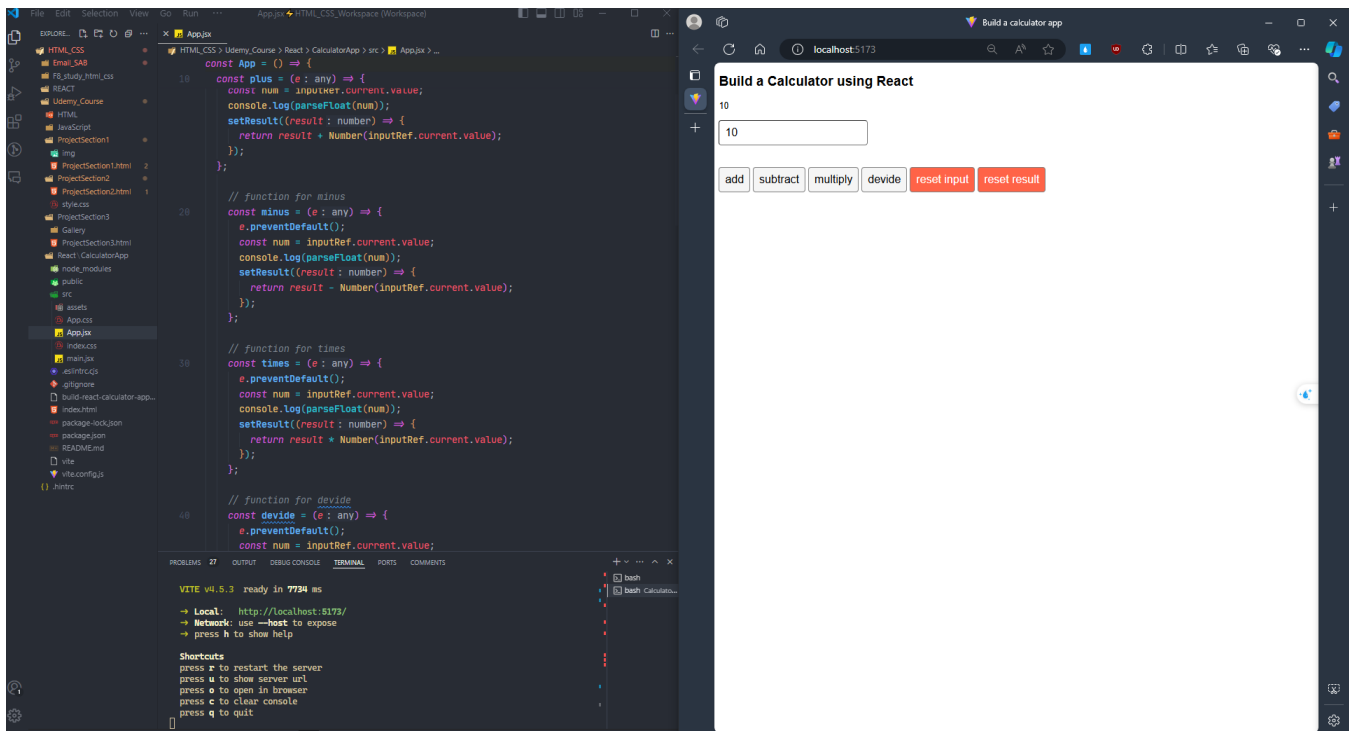## 3.1   What I learned

- **Data Structures and Algorithms:** I have learned about the basic data structures and algorithms, such as arrays, linked lists, stacks, queues, trees, graphs, searching, and sorting algorithms. I have also learned about the time complexity and space complexity of these algorithms.

- **Problem-solving skills:** I have learned how to solve problems using data structures and algorithms. I have also learned how to analyze the time complexity and space complexity of algorithms.

- **Programming skills:** I have improved my programming skills by implementing various data structures and algorithms in C++ and Python.

- **Front-end development:** I have learned how to create a simple web application using HTML, CSS, and JavaScript.

- **Version control:** I have learned how to use Git and GitHub for version control. And I stored my code in my personal repository.

3.2   What was my difficult

- **Time management:** I found it challenging to manage my time effectively to complete the assignments and projects on time.

- **Understanding complex algorithms:** I found it challenging to understand some complex algorithms, such as dynamic programming and graph algorithms.

- **Debugging:** I found it challenging to debug my code when it did not work as expected.

- **Language barrier:** I found it challenging to understand some of the lectures and materials due to the language barrier.

- **Lack of practice:** I found it challenging to practice regularly to improve my problem-solving and programming skills.