**Week 1 portfolio**

**Reflections.**

During week 1 I was able to learn the basics of Web App Design and development. I was introduced to the basics of computer networks and key points. Some of the things I learned are

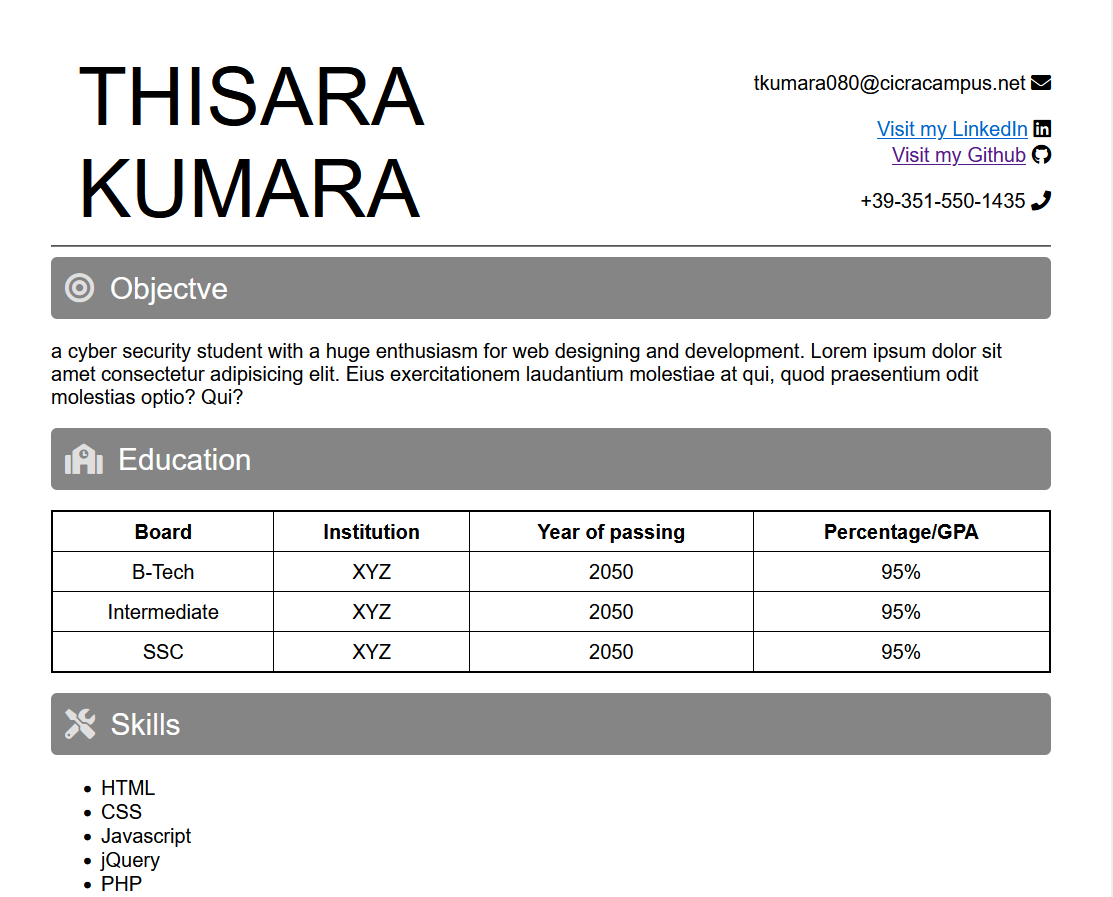
* Client - Server networks : Describes the computers that receive services and computers that provide services respectively.
* internet and Internet : The internet with simple “i” refers to a group of computers that are connected together forming a computer network. Where as the Internet with capital “I” is the entire collection of networks on the planet, which is the biggest network of computing systems and computer networks.
* WWW : WWW or the World Wide Web is a collection of electronic documents that are linked together.
* Hypertext Document : A document which contains the links to other documents,and are highlighted words in text or graphics.
* URL (Uniform Resource Locator) : The unique address of each page. It acts as the identifier of the pages location on the server.
* HTTP (Hyper Text Transfer Protocol) : Describes the way hypertext Documents are fetched over the internet.

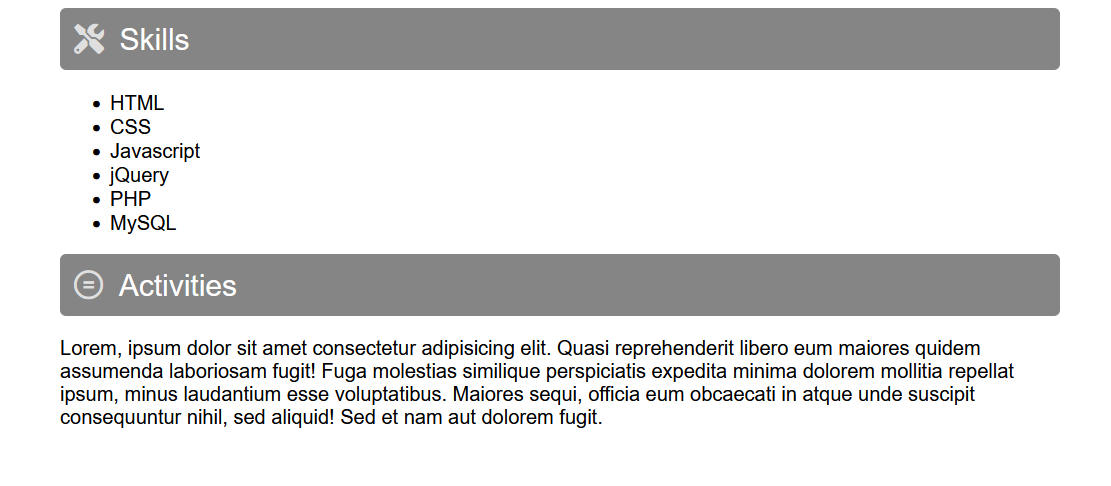
I also learned about web servers, web applications and HTML including HTML versions. I also learned basic HTML tags on the first week. Some are

* Header tags.
* Text adjusting tags.
* Nested tags etc. And we were asked to do a research on tags.

**Tasks.**

**Resume.**





Git hub link : https://github.com/river080/SIT120-RWA-assignments/tree/main/week1

**Week 2 Portfolio**

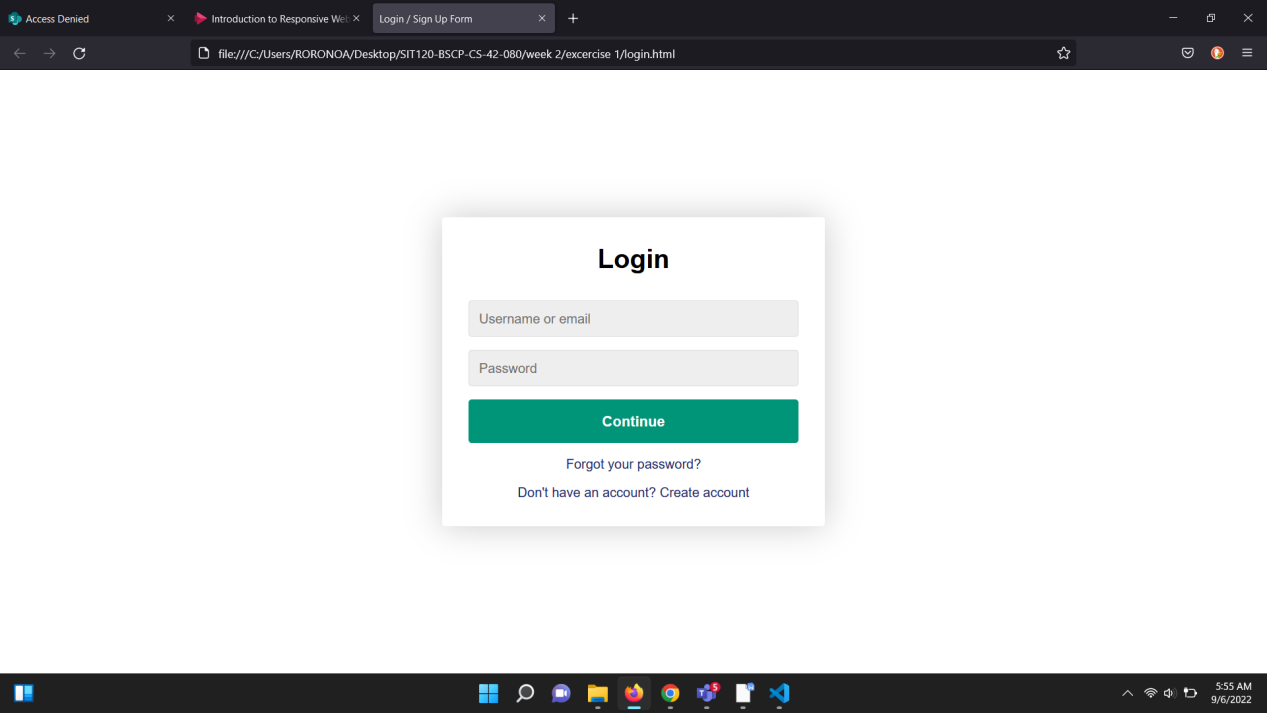
**Reflections**

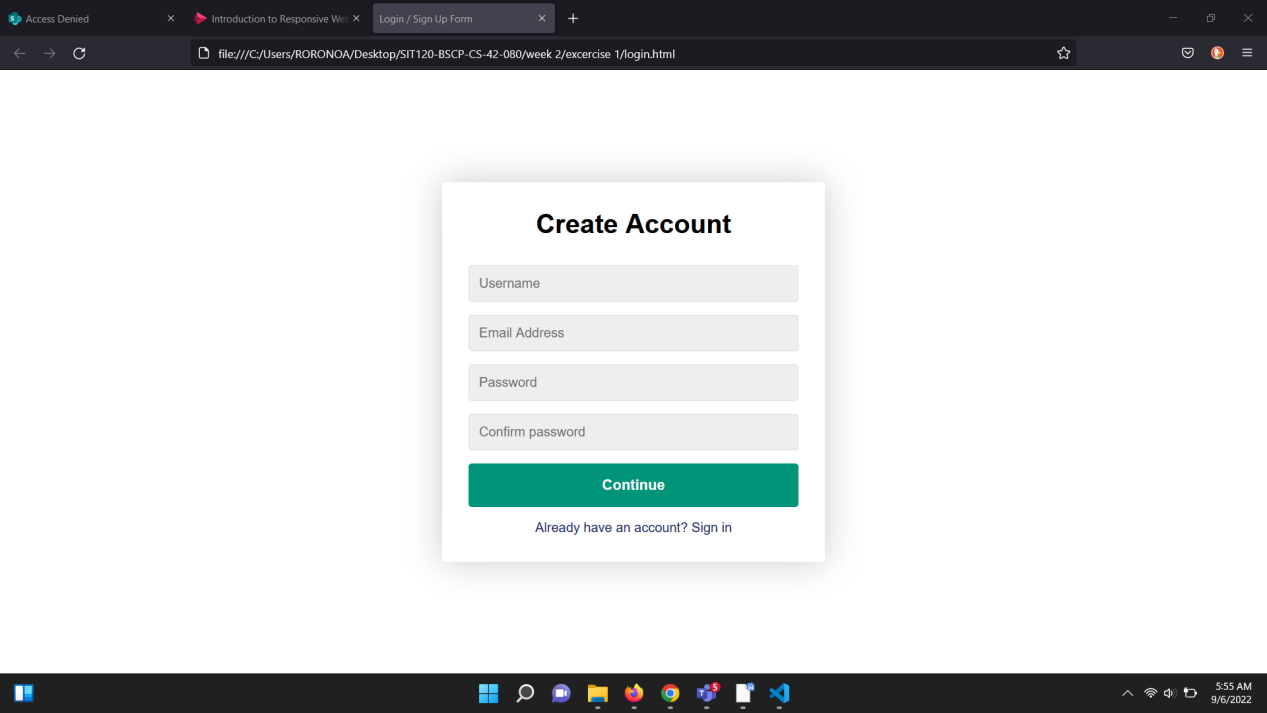
During week 2 we were taught about UI and UX process. GUI or UI designing means building interfaces in software or other computerized devices and UX designing works towards creating a better experience in using the products and services. Some of the things I learned are

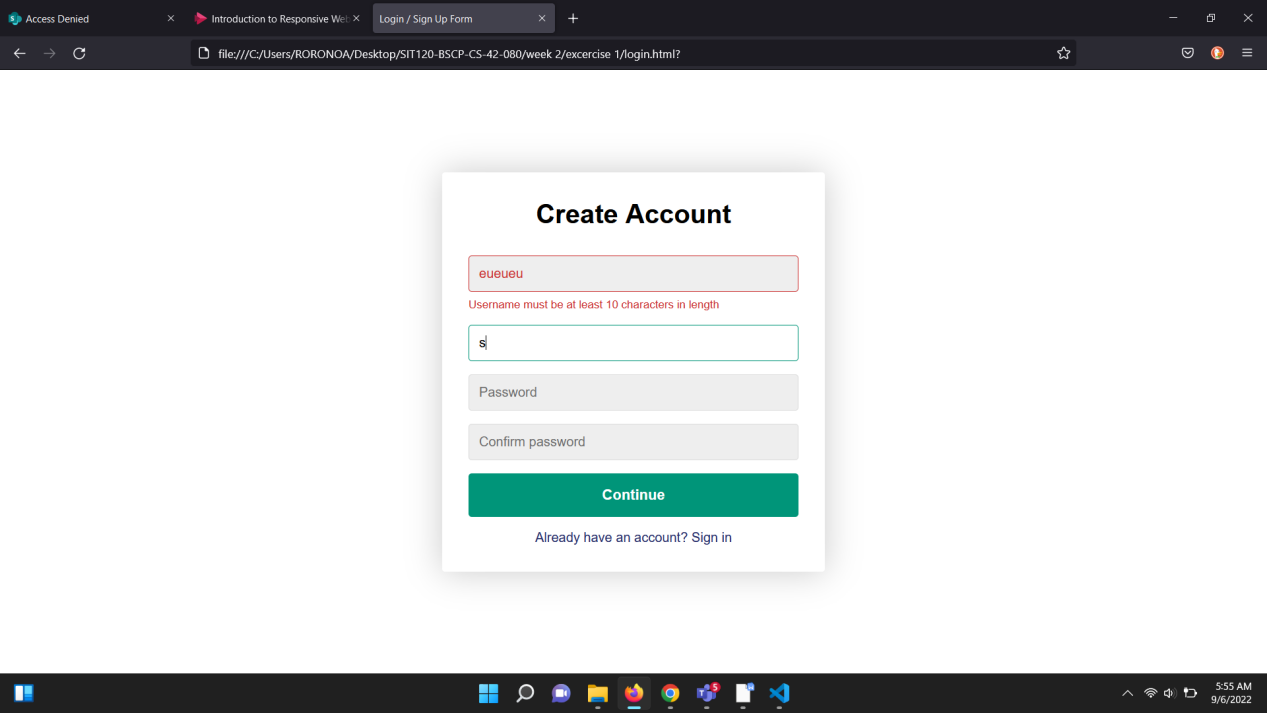
* GUI - an interface that allows the user to access and use a computer system without any complicated commands or text based system by the use of graphics. Our everyday using mobile phones , laptops use these graphical icons and other visual indicators to make a better and easier experience.
* UX - User Experience stands for how someones experience is when using a device or a software. It could be any website or an application or an OS. UX designing is the process of making a better environment and feels for anyone who would be using the software/device etc.
* Main factors that include when it comes to influencing a design are Aesthetics, Cost, Time, Safety. And there are 4 ways that we’ve been taught to improve a design for the user. They are : Mapping, Affording , Feedback and Constraints.

**Tasks**

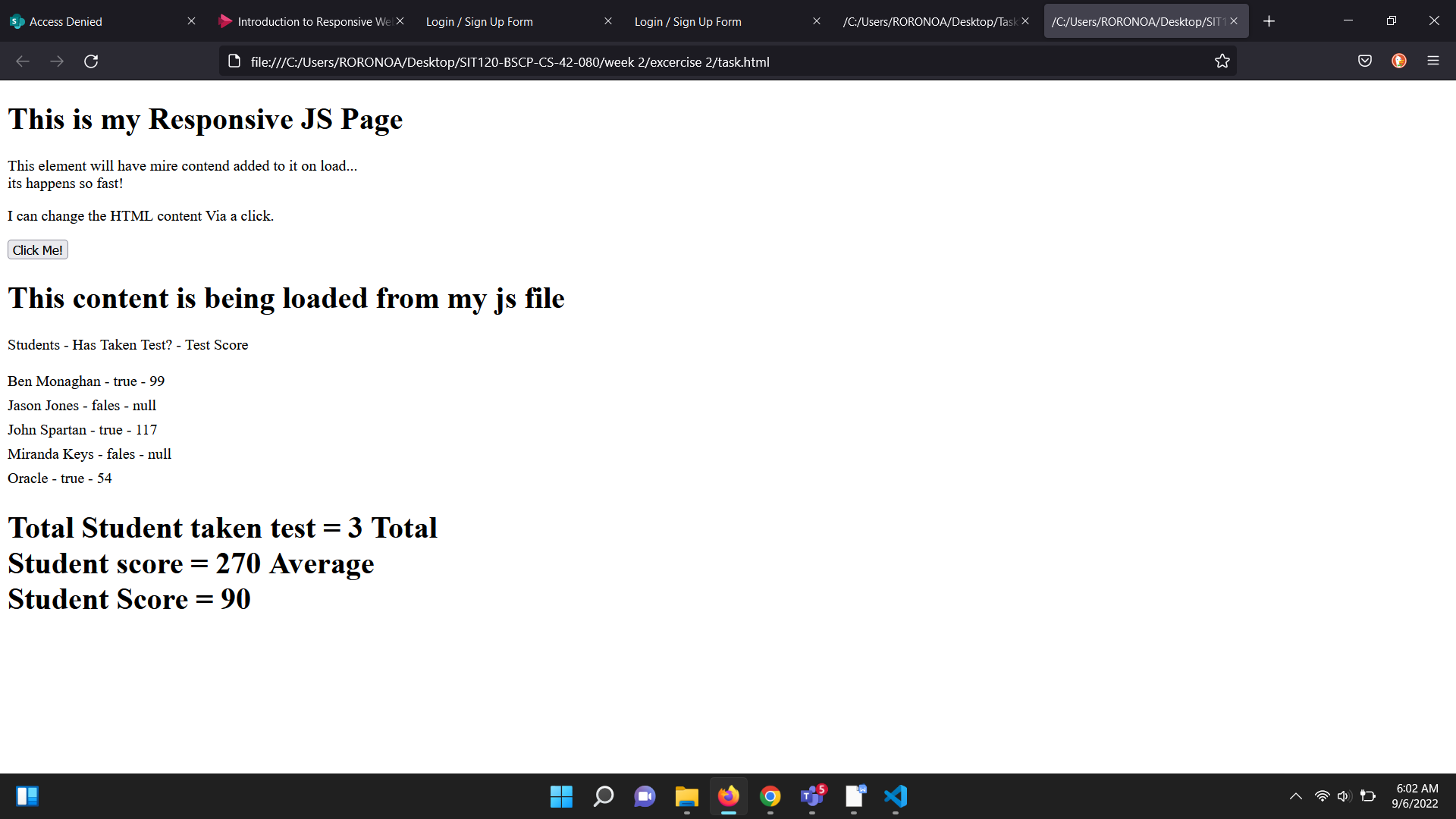
**Exercise 1. Login form.**







**Exercise 2.**



Github link : <https://github.com/river080/SIT120-RWA-assignments/tree/main/week2>

**Week 3 Portfolio**

**Reflections.**

During week 3 we were taught starting with JavaScript and basic functions. We were taught what JS is and what it does.

* It’s a scripting or programming language used today on the Web.
* It is used to enhance the functionality and appearance of web pages
* JavaScript is a client-side scripting language - Because JavaScript is interpreted by a browser,
* It is considered to be a client-side scripting language.
* We were also taught about JS being an object oriented language and is case sensitive as well.
* There are 3 kinds of declarations in JavaScript.

1. Var - declares a variable.
2. Let - declares a local variable initializing it to a value.
3. Const - declares a read-only names constant

* We also learned about JavaScript functions and advantages of functions. Namely 2.

1.Code re usability: We can call a function several times so it

save coding.

2.Less coding: It makes our program compact. We don’t need to

write many lines of code each time to perform a common

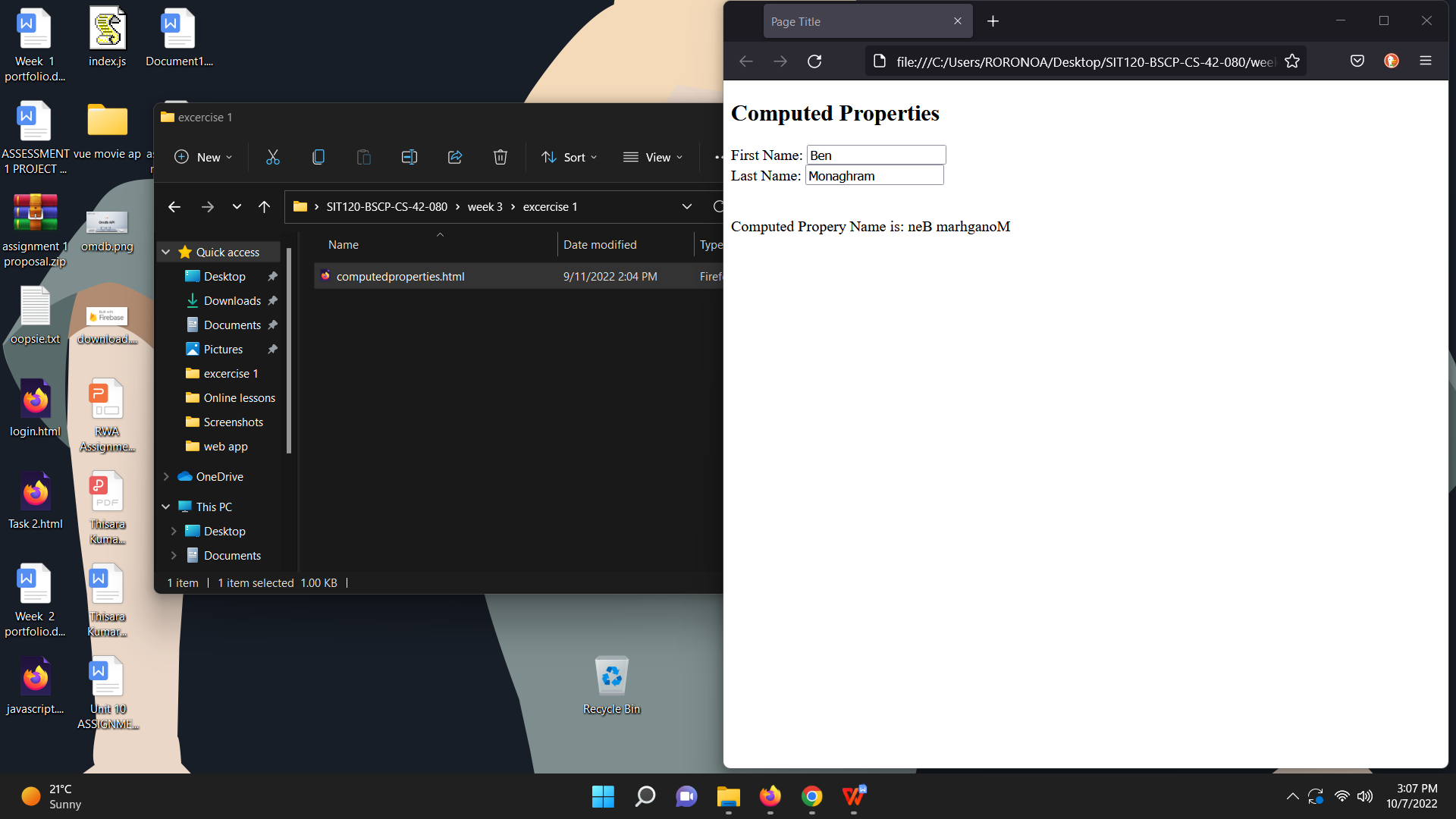
task.

* DOM - Document Object Model
* DOM is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content. The DOM represents the document as nodes and objects; that way, programming languages can interact with the page.
* JSON - JavaScript Object Notation stands for a text written with JS object notation and is a syntax for storing and exchanging data.
* We also learned about the JQuery library. Which is a widely used framework that simplifies JS programming. It contains the following major features.

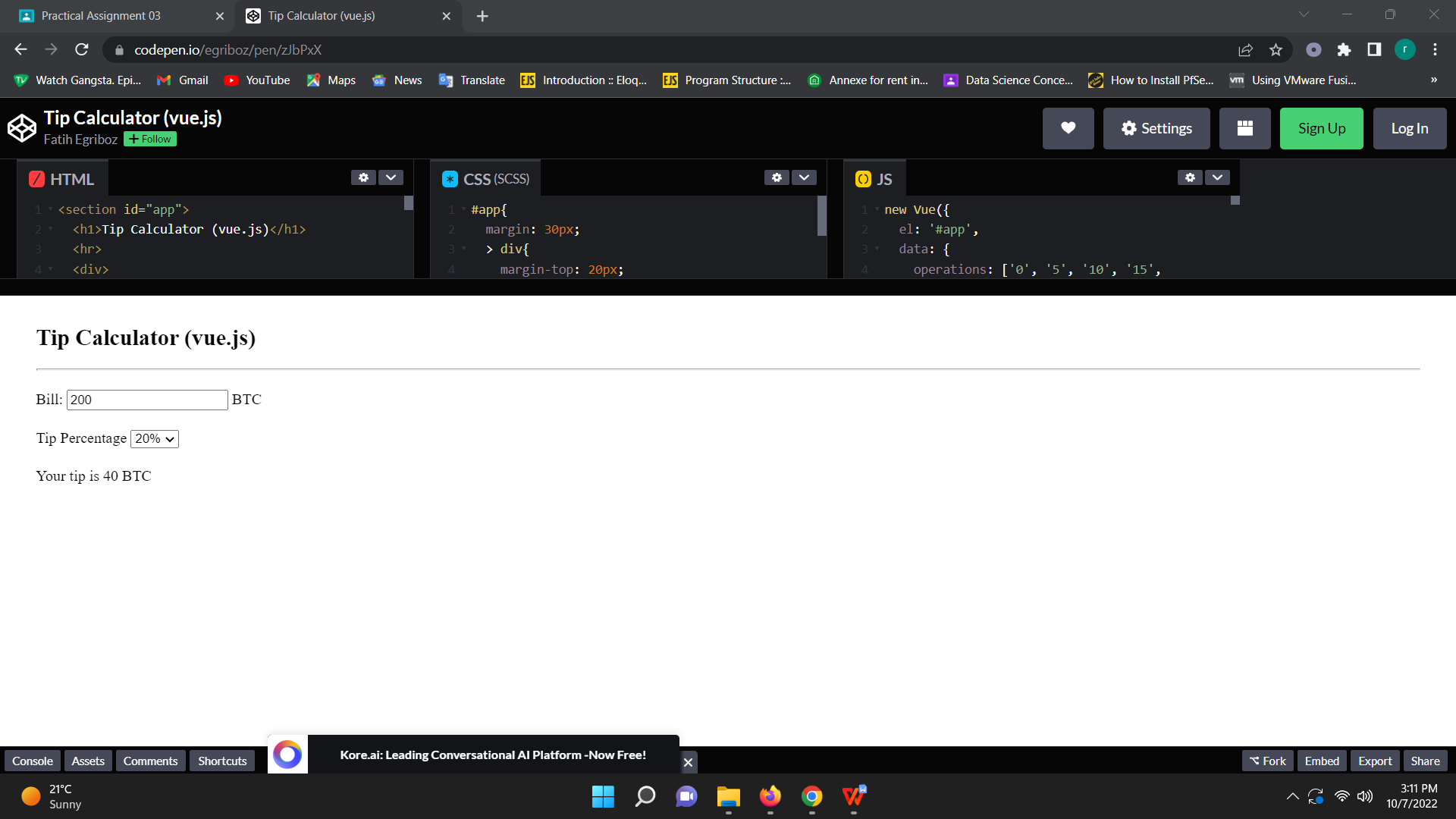
1. HTML/DOM manipulation
2. CSS manipulation
3. HTML event methods
4. Effects and animations
5. AJAX
6. Utilities.

**Tasks.**

**Exercise 1. Computed Properties.**



**Exercise 2. Tip Calculator.**



**Week 4 portfolio**

**Reflections**

We were introduced to VueJS frame work.

Principles of web design.

* Availability
* Performance
* Reliability
* Scalability
* Manageability
* Cost

Core Components of Web Applications

* UI (Front End (DOM, Framework))
* Request Layer (Web API)
* Back End (Database, Logic)

We also learned what a framework is.

* Software Framework designed to reduce overhead in web development

Types of Framework Architectures

* Model-View-Controller (MVC)
* Push vs Pull Based

Most MVC Frameworks user push-based architecture “action based” (Django, Ruby on Rails, Symfony, Stripes)

Pull-based or “component based” (Lift, Angular2, React)

Three Tier Organization

1. Client (Usually the browser running HTML/Javascipt/CSS)
2. Application (Running the Business Logic)
3. Database (Data Storage)

Types of Frameworks

* Server Side: Django, Ruby on Rails
* Client Side: Angular, React, Vue

MVC (Model View Controller)

A Web Application Development Framework

* Model (M):

Where the data for the DOM is stored and handled)

This is where the back-end connects

* View (V):

Think of this like a Page which is a single DOM

Where changes to the page are rendered and displayed

* Control (C):

This handles user input and interactions.

Buttons

Forms

General Interface

We were introduced to the vue framework and taught the basic coding concepts during practical sessions.

**Week 5 Portfolio**

**Reflections**

What is VueJS used for?

Vue. js (or simply Vue) is a lightweight, JavaScript framework for building reactive web user interfaces. Vue extends standard HTML and CSS to create a suite of powerful tools for building the front end of interactive web applications

User Input Handling with Vue.

• V- model directive is used to create two-way data bindings on form

input, textarea, and select elements

• It automatically picks the correct way to update the element based

on the input type.

• V-model will ignore the initial value, checked, or

selected attributes found on any form elements. It will always treat

the Vue instance data as the source of truth. You should declare the

initial value on the JavaScript side, inside the data option of your

Component

V-model

* V-model internally uses different properties and emits different

events for different input elements:

Text and textarea use ‘value’ property and ‘input’ event;

Checkboxes and radiobuttons use ‘checked’ property and ‘change’ event;

Select fields use ‘value’ as a prop and ‘change’ as an event.

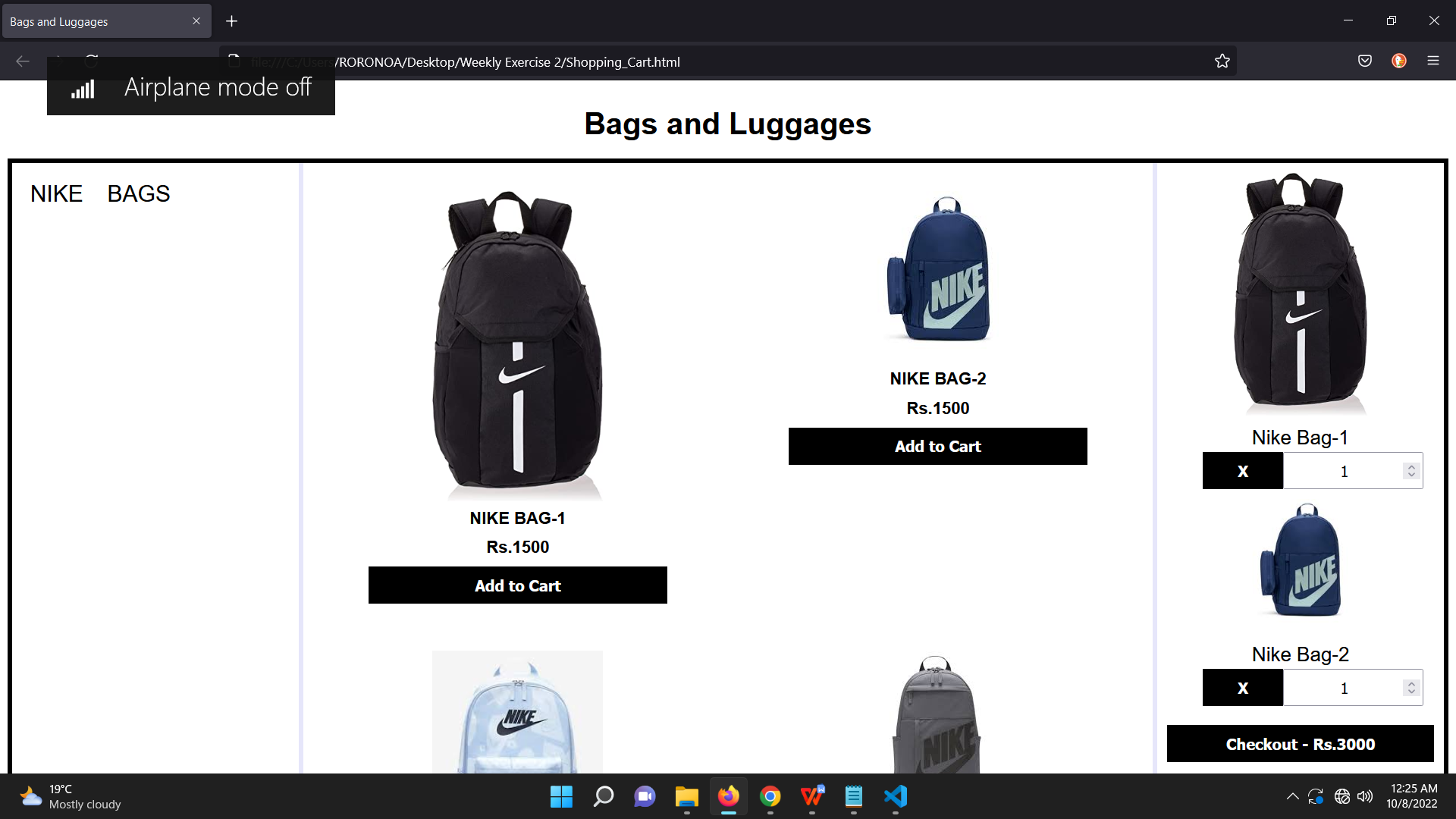
**Week 6 portfolio**

**Reflections**

During this week we learned about computed properties. You can use computed properties to calculate and display values based on a value or set of values in the data model. We were taught the method usage and examples on computed properties.

**Tasks.**

**Exercise. Shopping cart.**



Github Link : https://github.com/river080/SIT120-RWA-assignments/tree/main/week5

**Week 7 portfolio**

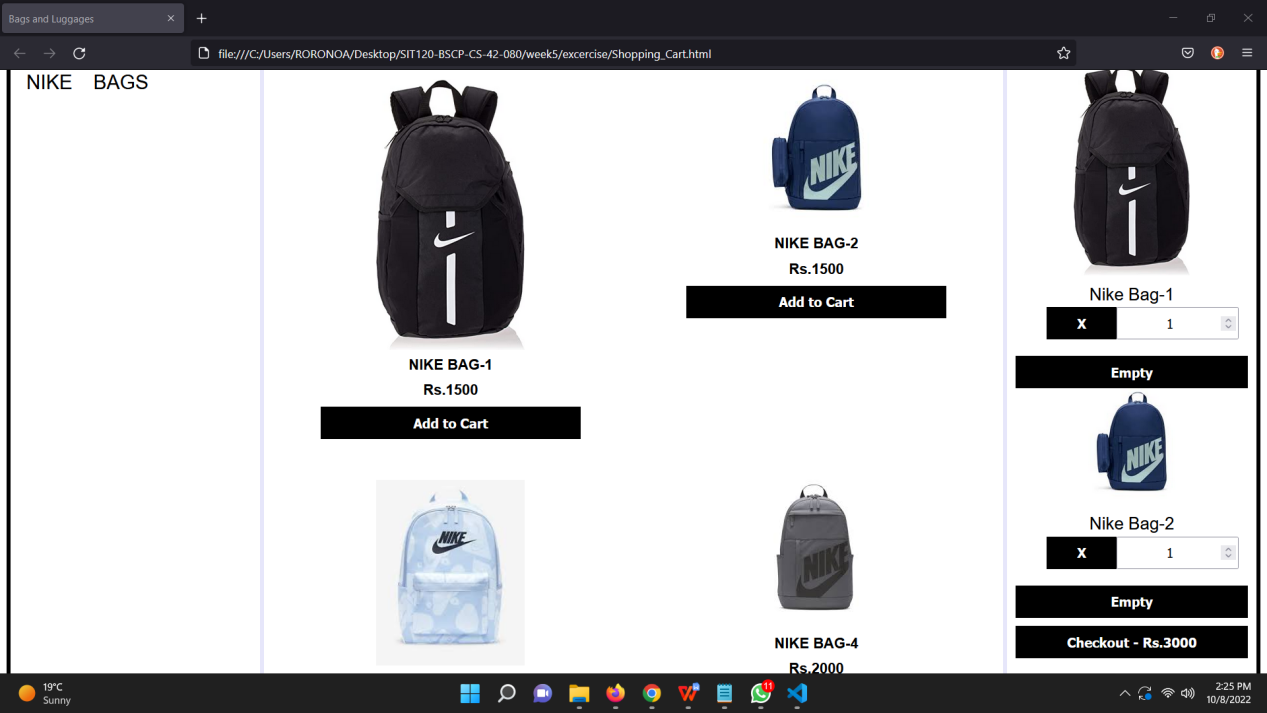
**Reflections**

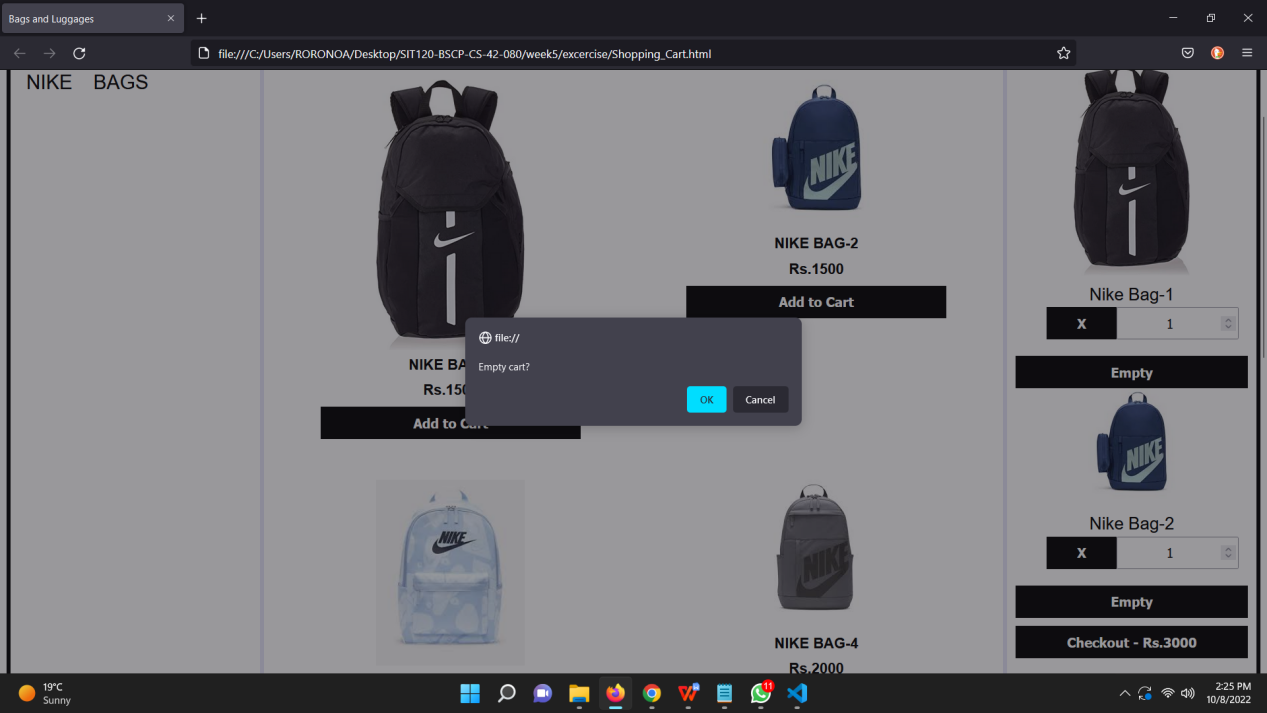
During this week, we learned about Transition Effects. A transition is a change in the style properties on an element to be transitioned in a single step. They are frequently handled only by CSS. Some of the things I learned are

* When elements are added to or removed from the DOM, Vue Transition applies automatic transition effects.
* The three options to implement transitions with Vue.js are: CSS transitions, CSS animations, JavaScript functions
* The benefits of transitions and animations like ; How it helps the user navigate the app by understanding connections between elements, It can draw attention to new or essential features and elements, It creates a unique and memorable user experience, etc.
* An animation is more complex than a transition. Animations are usually multi-step and sometimes run continuously. They will call on JavaScript to pick up where CSS’ lack of logic drops off.
* The 2 components provided by Vue to help with our animations and transitions are ; Transition -for applying animations when an element or component is entering and leaving the DOM, and Transition group - for applying animations when an element or component is inserted into, removed from, or moved within a v-for list.
* A transition element is a wrapper that enables you to give your elements transition functionality.
* <TransitionGroup> is a built-in component designed for animating the insertion, removal, and order change of elements or components that are rendered in a list.

**Tasks**

**Exercise. shopping cart.**





Github link : <https://github.com/river080/SIT120-RWA-assignments/tree/main/week6>

**Week 8 Portfolio**

**Reflections**

During week 8, we learned about Components. Components make it possible to separate the UI into separate, reusable parts and consider each one separately. I learned how to create a component. Some more of the things i learned are

* Dynamic components are created using the keyword<component></component>and it is bound using a property
* You can reuse components as many times as you'd like.
* Each time you use a component, a new instance of it is created.
* A  component’s data option must be a function, so that each instance can maintain an independent copy of the returned data object.
* Props are custom attributes you can register on a component

**Week 9 portfolio**

**Reflections**

During week 9 we learned about Props. State is transmitted to child components using props. Components can accept data from components that include them using props.

Some of the things I learned are

* Props are read-only and cannot be modified by the child component because the parent component "owns" that value
* You can have multiple props by appending them to the array
* You can specify the type of a prop by using an object instead of an array by using the name of the property as the key of each property, and the type as the value
* You can even build a custom validator, which is ideal for complex data

**Week 10 portfolio**

**Reflections**

During week 10, we were taught about slots. A slot is a tag that we may provide in the child component and then replace with information from the parent.The parent component can embed content, including HTML elements, in the child component via slots. Some of the things I learned are

* By using props, the parent can transfer values to the child.
* Using emit, we can trigger events and pass data up the component hierarchy.
* Vue $emit is a function that lets us emit, or send custom events from a child component to its parent.
* How to create and use slots - Where we want the overriding content from the parent to show, we add an open-and-close slot tag to the template of the child component.
* The overriding content is added between the component's open and close tags when it is included in the parent template.
* If the parent doesn't offer any overriding content, Vue allows us to specify default content in the slot.
* Vue allows us to have multiple slots in a single component. However, Vue wants us to label each slot element in order to make sure that the content we provide in the parent does not override the incorrect slot in the child.
* To name a slot, we attach the name attribute to the slot element and specify a unique name as its value.