COMP-1842 WEB PROGRAMMING 2

ZOBAIYA HASAN

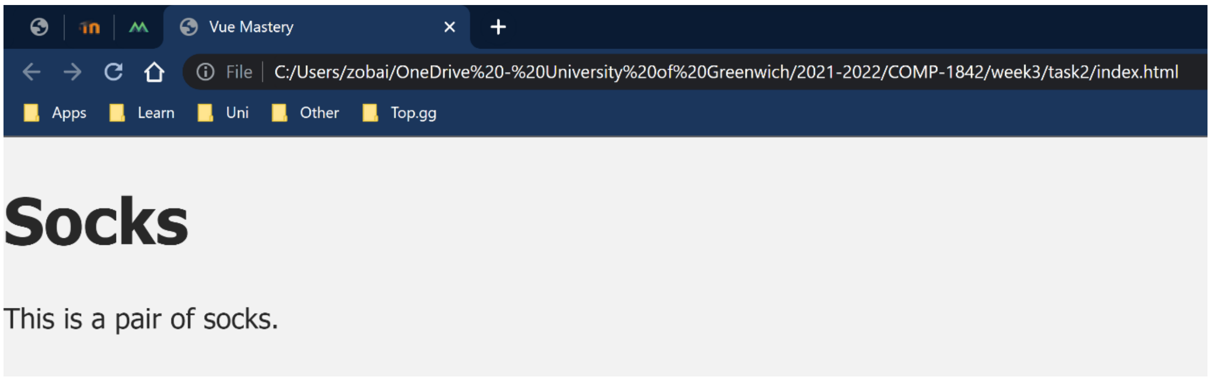
ID: 001113069

Contents

Weekly Tasks

Week 3

Lesson 2: Creating the Vue App

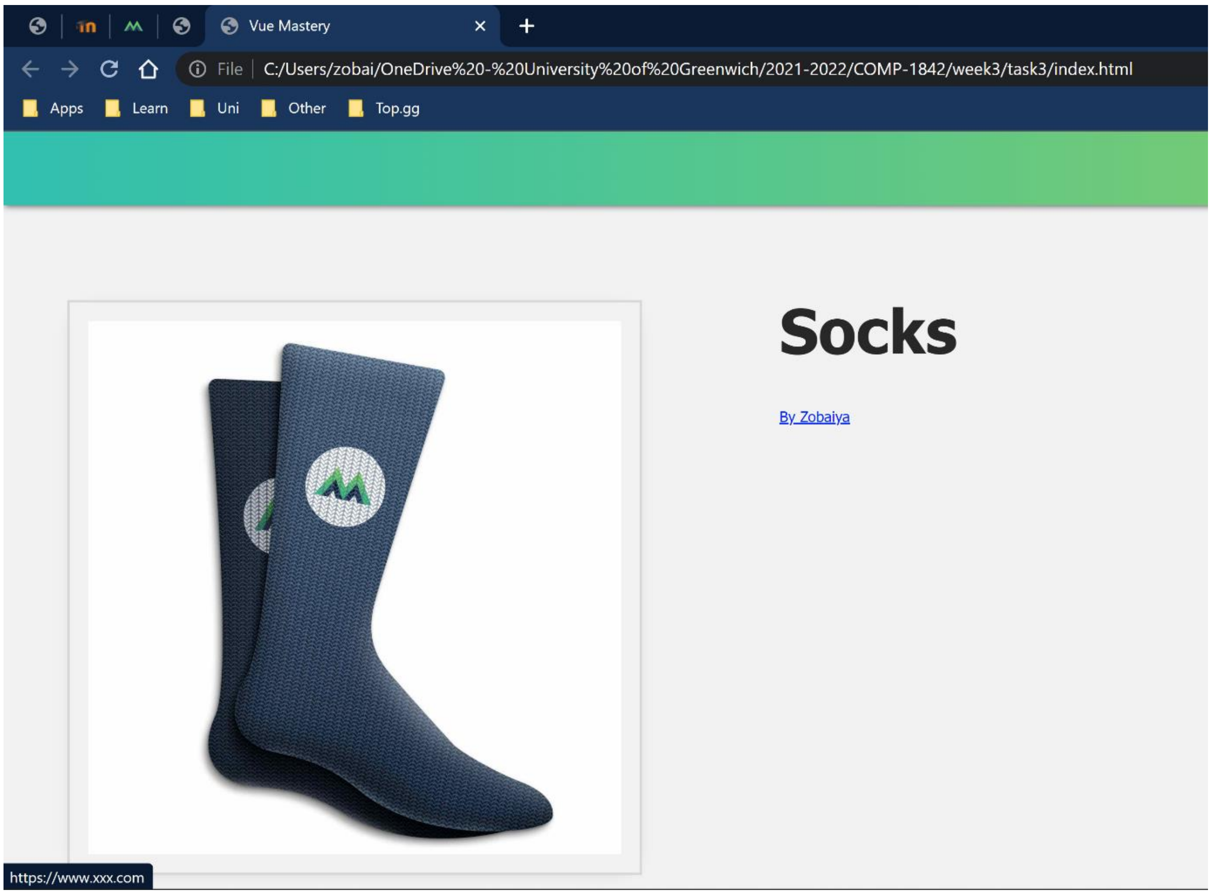

Vue Mastery
File
C:/Users/zobai/OneDrive%20-%20University%20of%20Greenwich/2021-2022/COMP-1842/week3/task2/index.html
Apps
Learn
Uni
Other
Top.gg

Socks

This is a pair of socks.

Figure 1: week 3, lesson 2

Lesson 3: Attribute Binding

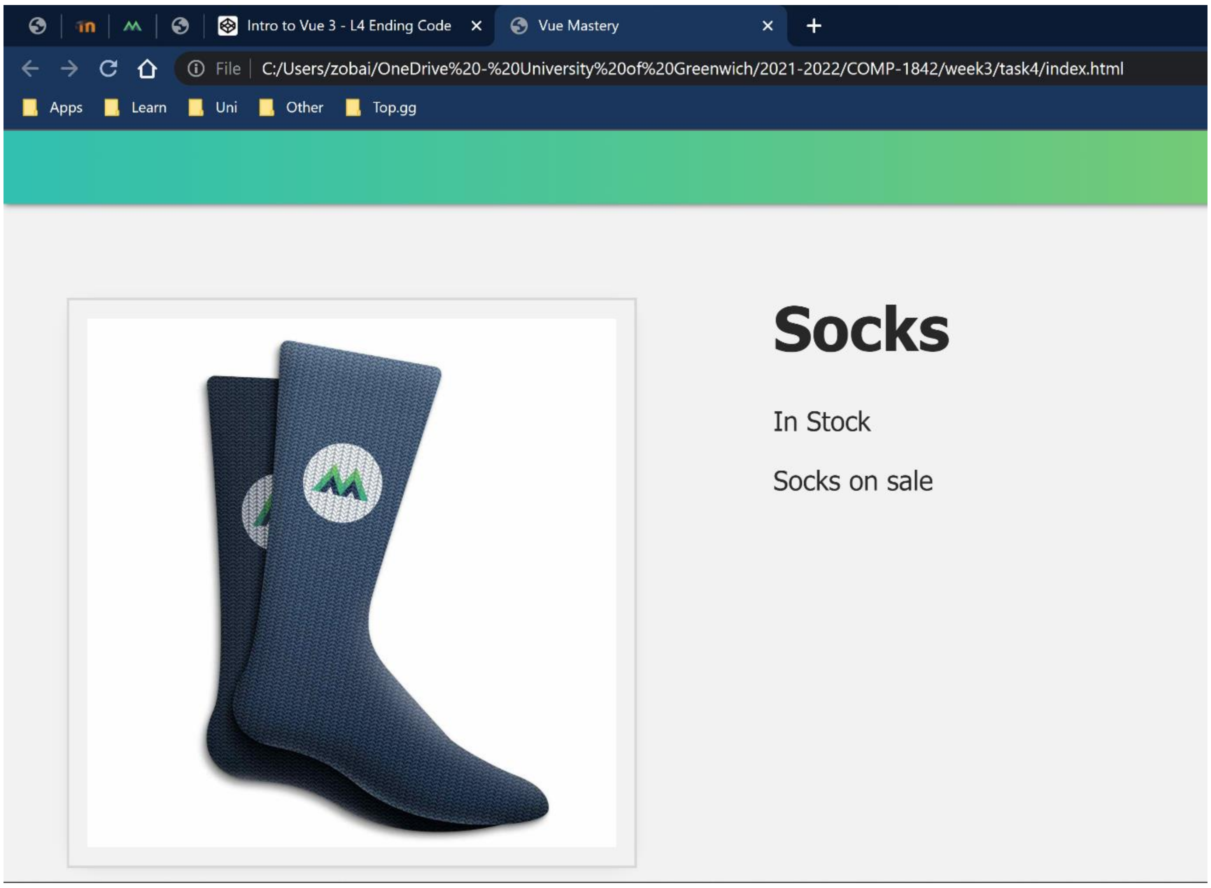

Vue Mastery
File
C:/Users/zobai/OneDrive%20-%20University%20of%20Greenwich/2021-2022/COMP-1842/week3/task3/index.html
Apps
Learn
Uni
Other
Top.gg
https://www.xxx.com

Socks

By Zobaiya

Figure 2: week 3, lesson 3

Lesson 4: Conditional Rendering

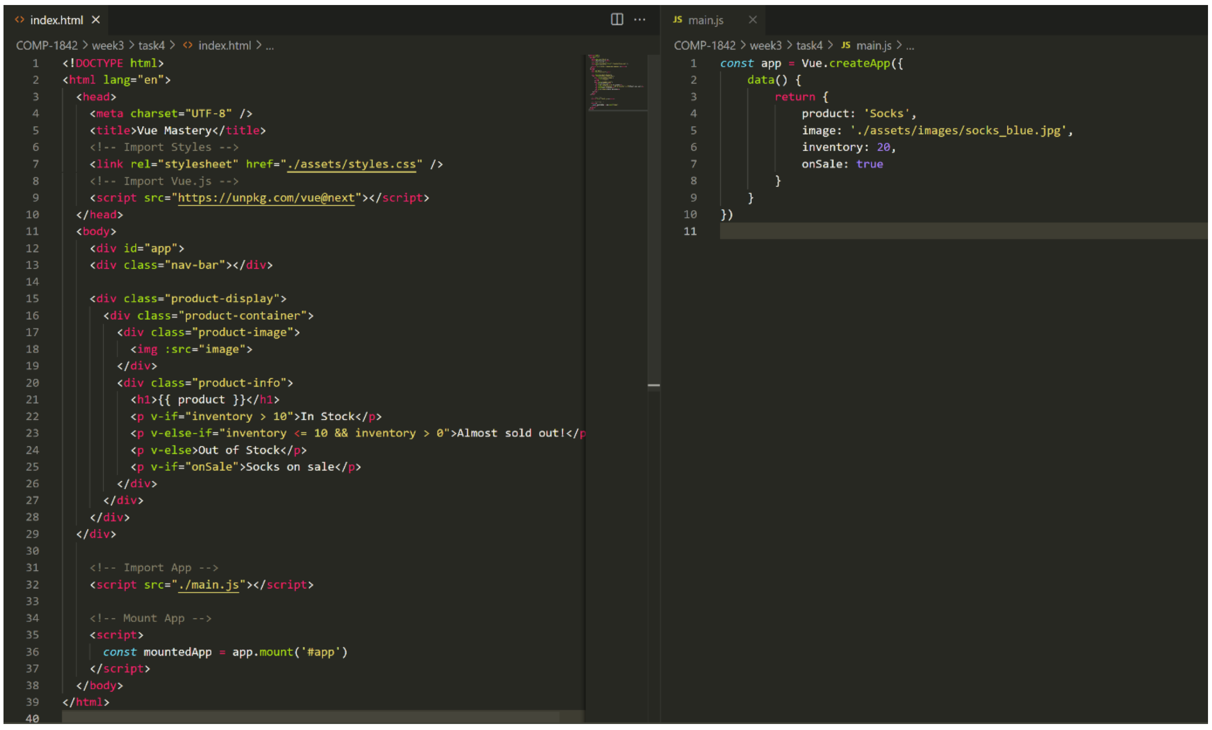

Intro to Vue 3 - L4 Ending Code
Vue Mastery
File | C:/Users/zobai/OneDrive%20-%20University%20of%20Greenwich/2021-2022/COMP-1842/week3/task4/index.html
Apps Learn Uni Other Top.gg

Socks

In Stock

Socks on sale

Figure 3: week 3, lesson 4


index.html

COMP-1842 > week3 > task4 > index.html > ...

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Vue Mastery</title>
    <!-- Import Styles -->
    <link rel="stylesheet" href="./assets/styles.css" />
    <!-- Import Vue.js -->
    <script src="https://unpkg.com/vue@next"></script>
  </head>
  <body>
    <div id="app">
    <div class="nav-bar"></div>

    <div class="product-display">
      <div class="product-container">
        <div class="product-image">
          <img :src="image">
        </div>
        <div class="product-info">
          <h1>{{ product }}</h1>
          <p v-if="inventory > 10">In Stock</p>
          <p v-else-if="inventory <= 10 && inventory > 0">Almost sold out!</p
          <p v-else>Out of Stock</p>
          <p v-if="onSale">Socks on sale</p>
        </div>
      </div>
    </div>
  </div>

    <!-- Import App -->
    <script src="./main.js"></script>

    <!-- Mount App -->
    <script>
      const mountedApp = app.mount('#app')
    </script>
  </body>
</html>
```

main.js

COMP-1842 > week3 > task4 > main.js > ...

```javascript
const app = Vue.createApp({
    data() {
        return {
            product: 'Socks',
            image: './assets/images/socks_blue.jpg',
            inventory: 20,
            onSale: true
        }
    }
})
```

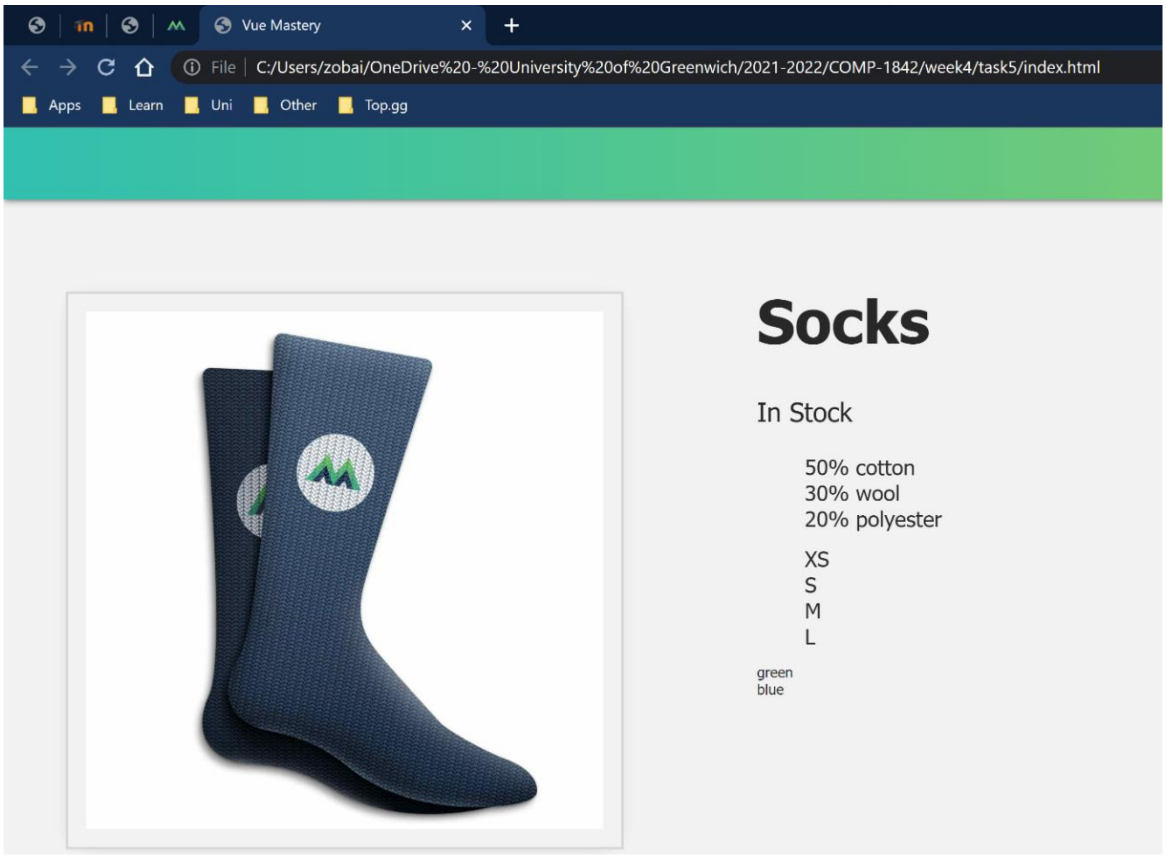
Figure 4: week 3, lesson 4 code

Week 3 Explanation

In lesson 2 I learnt how to create the main page of the app itself - how to write the basics such as the name and description of the app. They also showed that you have to import and mount the app into the DOM (Document Object Model) which is the interface for web documents. In lesson 3, I learnt to add an image to the data. I did this by adding a path to the image in the main.js file and then adding the img template to the index.html file. Here I have used the v-bind directive which enabled me to bind an attribute to an expression, in this case it was the src to the image. The coding challenge for lesson 3 was to add a url to the data object which I have done using the same steps as when adding the image. In lesson 4, I have learnt to change the HTML elements based on a condition (e.g. if there are less than 10 socks then it should display out of stock). Here I have used the v-if directive and v-else directive to show if the item is in stock or out of stock. I have also used v-show directive to toggle the element's visibility.

Week 4

Lesson 5: List Rendering


Vue Mastery
C:/Users/zobai/OneDrive%20-%20University%20of%20Greenwich/2021-2022/COMP-1842/week4/task5/index.html
Apps
Learn
Uni
Other
Top.gg

Socks

In Stock

50% cotton
30% wool
20% polyester

XS
S
M
L

green
blue

Figure 5: week 4, lesson 5

Lesson 6: Event Handling

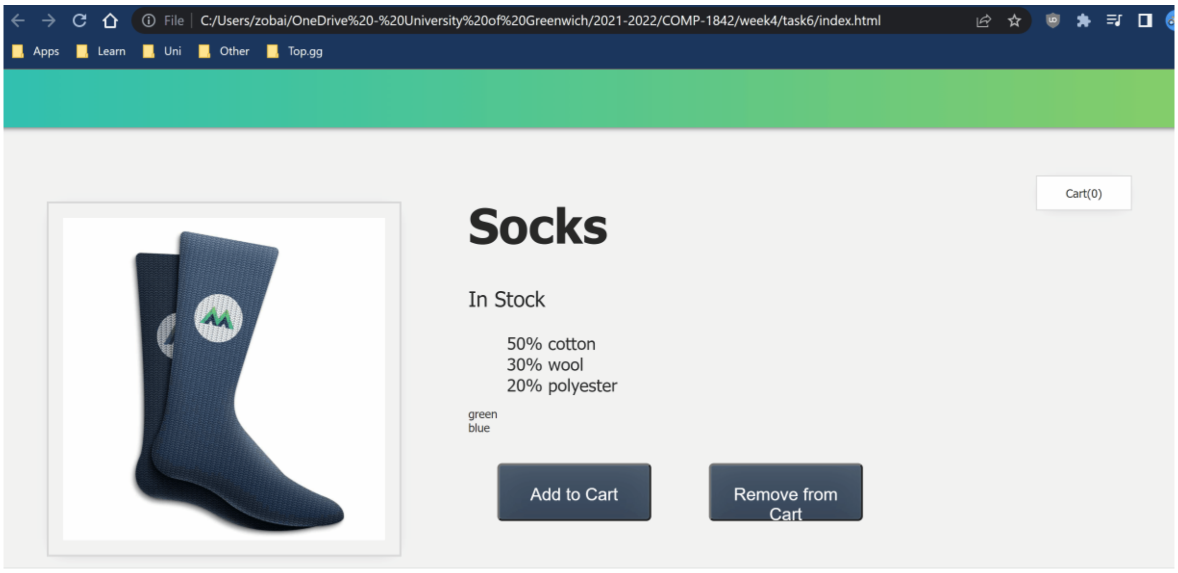


Figure 6: week 4, lesson 6

Lesson 7: Class & Style Binding

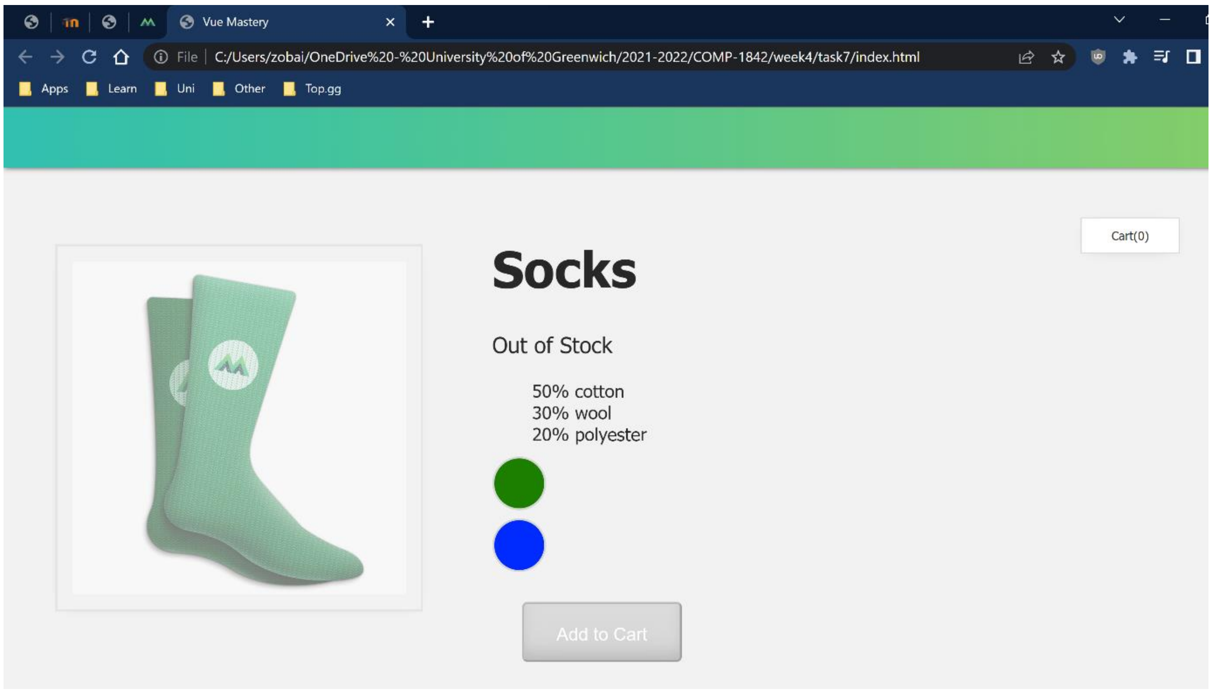
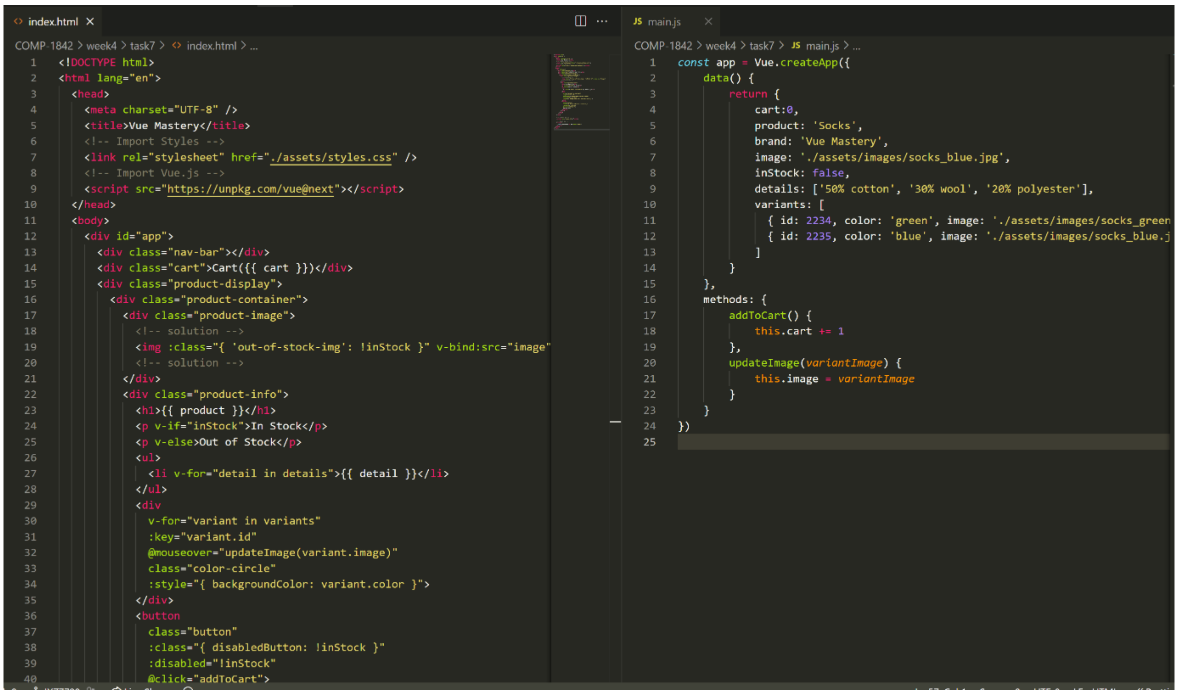

Vue Mastery
C:/Users/zobai/OneDrive%20-%20University%20of%20Greenwich/2021-2022/COMP-1842/week4/task7/index.html
Apps
Learn
Uni
Other
Top.gg
Cart(0)
Socks
Out of Stock
50% cotton
30% wool
20% polyester
Add to Cart

Figure 7: week 4, lesson 7


index.html
COMP-1842 > week4 > task7 > index.html > ...

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Vue Mastery</title>
    <!-- Import Styles -->
    <link rel="stylesheet" href="./assets/styles.css" />
    <!-- Import Vue.js -->
    <script src="https://unpkg.com/vue@next"></script>
  </head>
  <body>
    <div id="app">
      <div class="nav-bar"></div>
      <div class="cart">Cart({{ cart }})</div>
      <div class="product-display">
        <div class="product-container">
          <div class="product-image">
            <!-- solution -->
            <img :class="{ 'out-of-stock-img': !inStock }" v-bind:src="image"
            <!-- solution -->
          </div>
          <div class="product-info">
            <h1>{{ product }}</h1>
            <p v-if="inStock">In Stock</p>
            <p v-else>Out of Stock</p>
            <ul>
              <li v-for="detail in details">{{ detail }}</li>
            </ul>
            <div
              v-for="variant in variants"
              :key="variant.id"
              @mouseover="updateImage(variant.image)"
              class="color-circle"
              :style="{ backgroundColor: variant.color }">
            </div>
            <button
              class="button"
              :class="{ disabledButton: !inStock }"
              :disabled="!inStock"
              @click="addToCart">
```

main.js
COMP-1842 > week4 > task7 > main.js > ...

```javascript
const app = Vue.createApp({
    data() {
        return {
            cart:0,
            product: 'Socks',
            brand: 'Vue Mastery',
            image: './assets/images/socks_blue.jpg',
            inStock: false,
            details: ['50% cotton', '30% wool', '20% polyester'],
            variants: [
              { id: 2234, color: 'green', image: './assets/images/socks_green
              { id: 2235, color: 'blue', image: './assets/images/socks_blue.j
            ]
        }
    },
    methods: {
        addToCart() {
            this.cart += 1
        },
        updateImage(variantImage) {
            this.image = variantImage
        }
    }
})
```

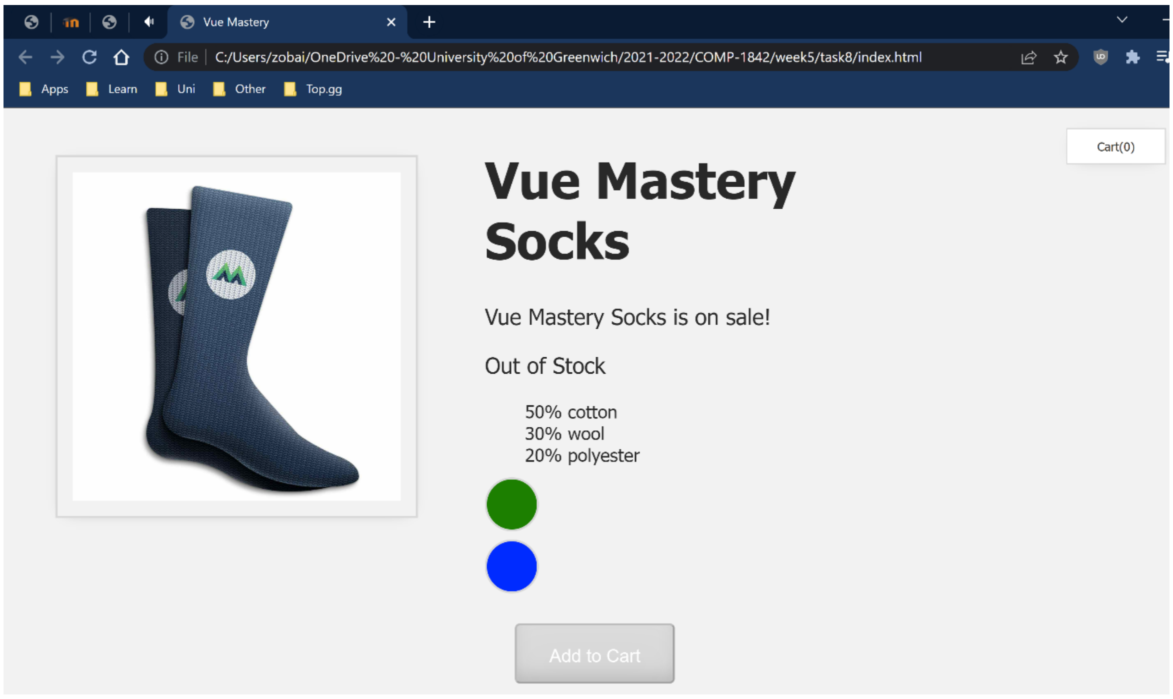
Figure 8: week 4, lesson 7 code

Week 4 Explanation

In lesson 5 I have learnt to add a list/essentially a description of the product to the data. I did this by adding an array of details in the main.js file and then using the v-for directive to display the details in a list. The key attribute helps to manage the data as they update. In lesson 6, I have learnt to add a button which increases the amount of items in the cart. I did this by adding the v-on directive to listen for an event "click", which increases the value of the cart by 1 once clicked. After that was added to the index.html file, I have also added a new addToCart method which will run after you click on the button. I then have also learnt to change the colour of the item once you hover over the colour you want by adding a mouseover event (what Vue calls hover). The coding challenge was to decrease the value of the cart once you click on a button so I followed the same steps to recreate the incrementing button. In lesson 7, I have added styles to the elements based on the app's data. I have added 2 divs and styled them like circles by adding a class to the variant div. I have learnt about class binding to ensure that the user cannot add the product to the cart if it is out of stock.

Week 5

Lesson 8: Computed Properties


Vue Mastery

C:/Users/zobai/OneDrive%20-%20University%20of%20Greenwich/2021-2022/COMP-1842/week5/task8/index.html

Apps Learn Uni Other Top.gg

Cart(0)

Vue Mastery Socks

Vue Mastery Socks is on sale!

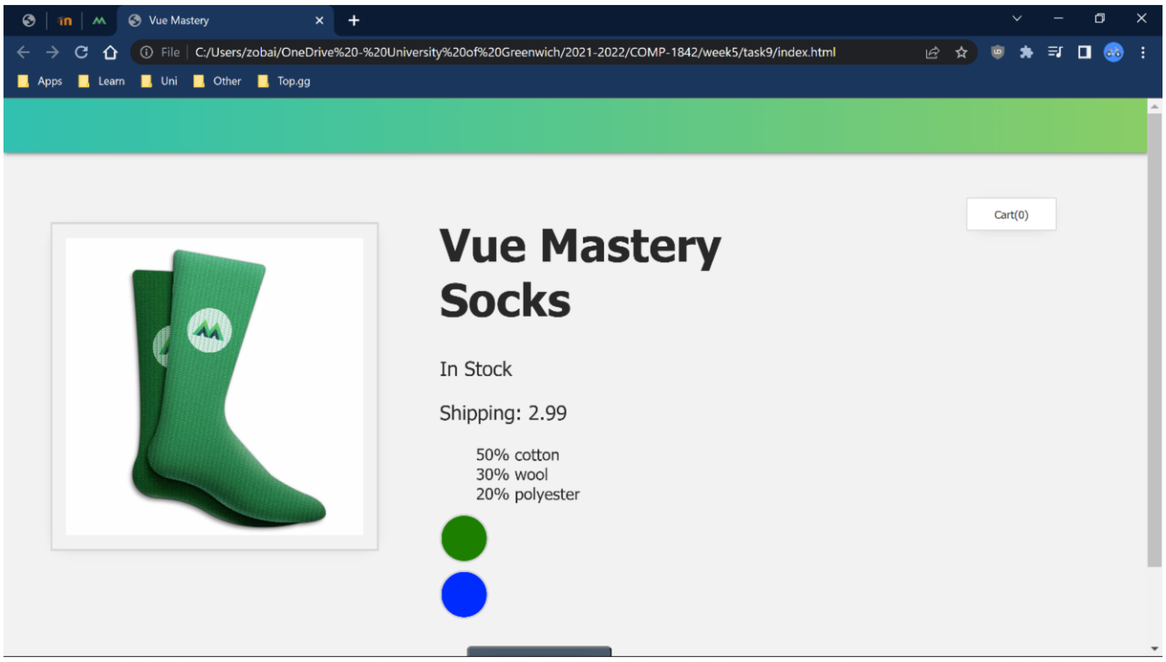
Out of Stock

50% cotton
30% wool
20% polyester

Add to Cart

Figure 9: week 5, lesson 8

Lesson 9: Components & Props


Vue Mastery

C:/Users/zobai/OneDrive%20-%20University%20of%20Greenwich/2021-2022/COMP-1842/week5/task9/index.html

Apps Learn Uni Other Top.gg

Cart(0)

Vue Mastery Socks

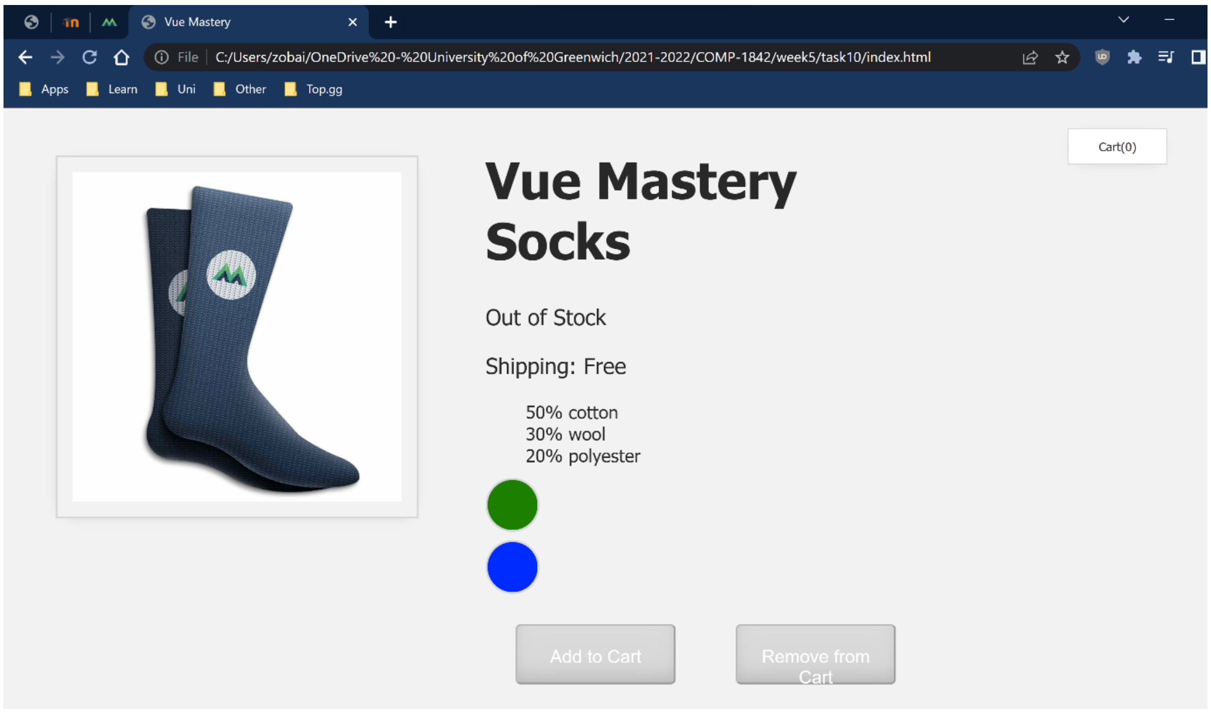
In Stock

Shipping: 2.99

50% cotton
30% wool
20% polyester

Figure 10: week 5, lesson 9

Lesson 10: Communicating Events


Vue Mastery
C:/Users/zobai/OneDrive%20-%20University%20of%20Greenwich/2021-2022/COMP-1842/week5/task10/index.html
Apps
Learn
Uni
Other
Top.gg
Cart(0)
Add to Cart
Remove from Cart

Vue Mastery Socks

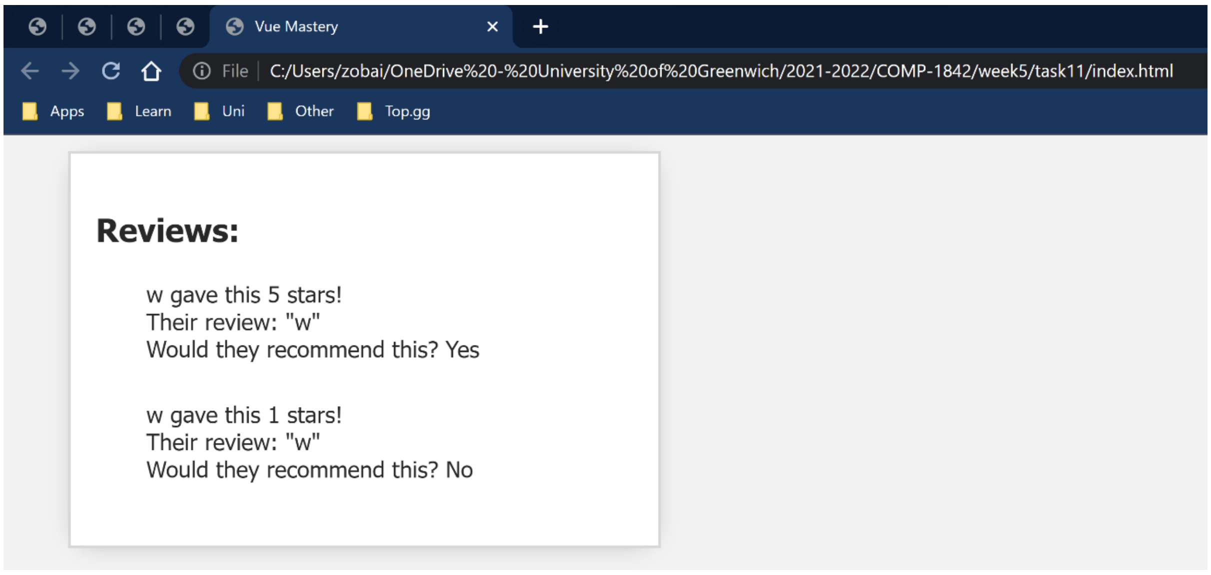
Out of Stock

Shipping: Free

50% cotton
30% wool
20% polyester

Figure 11: week 5, lesson 10

Lesson 11: Forms & v-model

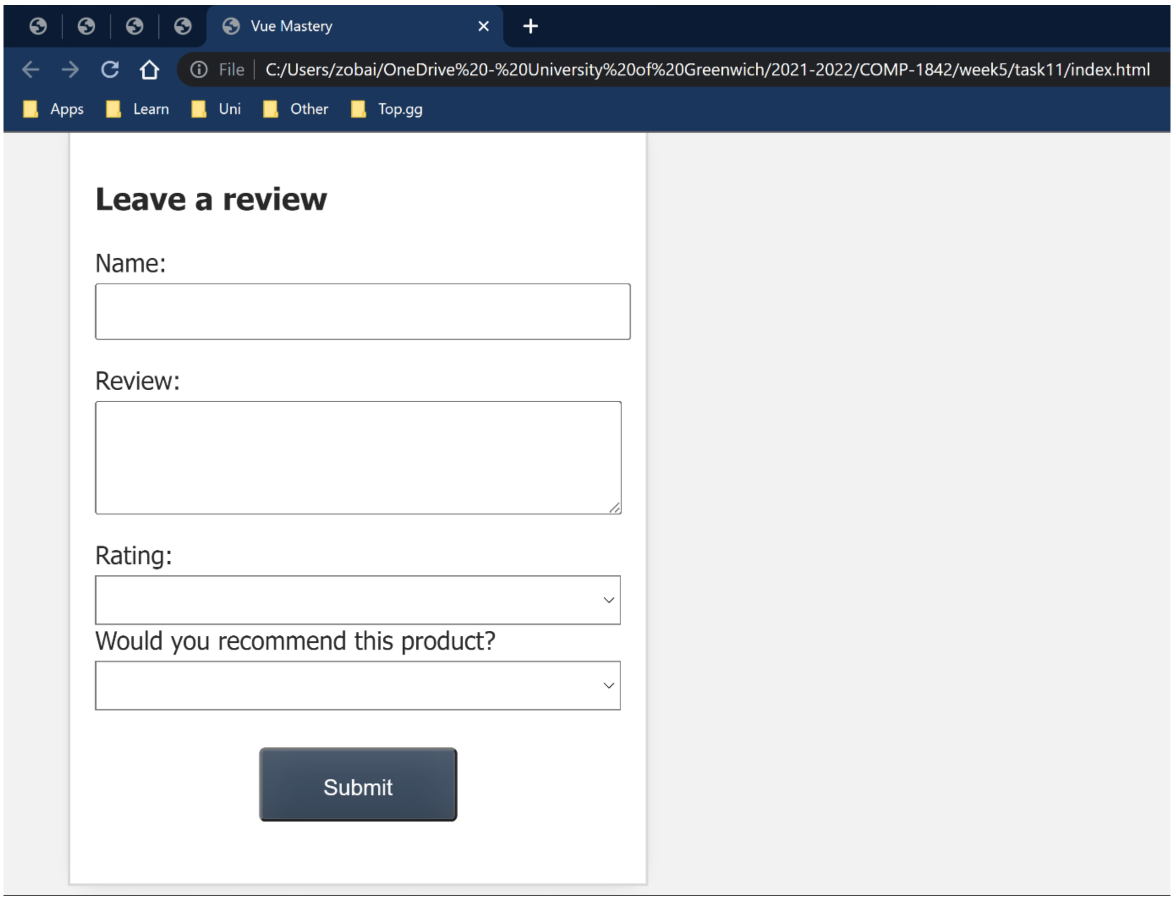

Vue Mastery
C:/Users/zobai/OneDrive%20-%20University%20of%20Greenwich/2021-2022/COMP-1842/week5/task11/index.html
Apps
Learn
Uni
Other
Top.gg

Reviews:

w gave this 5 stars!
Their review: "w"
Would they recommend this? Yes

w gave this 1 stars!
Their review: "w"
Would they recommend this? No

Figure 12: week 5, lesson 11


Vue Mastery

File | C:/Users/zobai/OneDrive%20-%20University%20of%20Greenwich/2021-2022/COMP-1842/week5/task11/index.html

Apps | Learn | Uni | Other | Top.gg

Leave a review

Name:

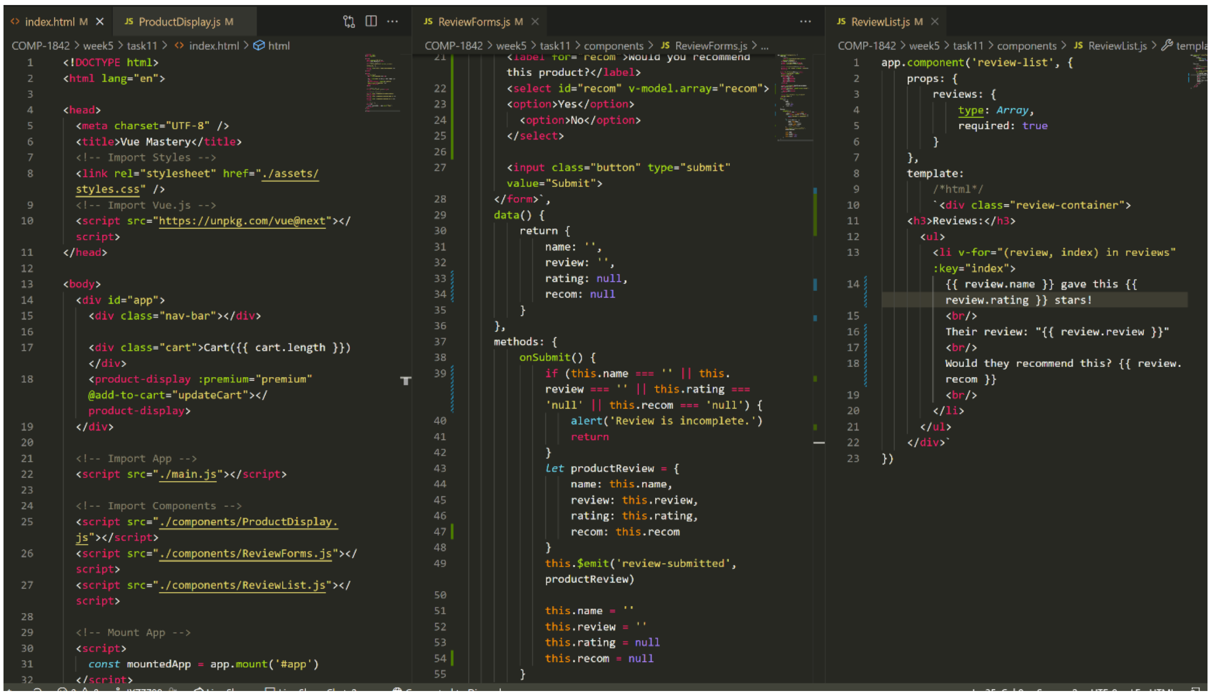
Review:

Rating:

Would you recommend this product?

Submit

Figure 13: week 5, lesson 11


index.html M | ProductDisplay.js M

COMP-1842 > week5 > task11 > index.html > html

```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <title>Vue Mastery</title>
  <!-- Import Styles -->
  <link rel="stylesheet" href="./assets/styles.css" />
  <!-- Import Vue.js -->
  <script src="https://unpkg.com/vue@next"></script>
</head>

<body>
  <div id="app">
    <div class="nav-bar"></div>

    <div class="cart">Cart({{ cart.length }})</div>
    <product-display :premium="premium" @add-to-cart="updateCart"></product-display>
  </div>

  <!-- Import App -->
  <script src="./main.js"></script>

  <!-- Import Components -->
  <script src="./components/ProductDisplay.js"></script>
  <script src="./components/ReviewForms.js"></script>
  <script src="./components/ReviewList.js"></script>

  <!-- Mount App -->
  <script>
    const mountedApp = app.mount('#app')
  </script>
```

ReviewForms.js M

COMP-1842 > week5 > task11 > components > ReviewForms.js > ...

```javascript
      <label for="recom">Would you recommend this product?</label>
      <select id="recom" v-model.array="recom">
      <option>Yes</option>
        <option>No</option>
      </select>

      <input class="button" type="submit" value="Submit">
    </form>`,
    data() {
      return {
        name: '',
        review: '',
        rating: null,
        recom: null
      }
    },
    methods: {
      onSubmit() {
        if (this.name === '' || this.review === '' || this.rating === 'null' || this.recom === 'null') {
          alert('Review is incomplete.')
          return
        }
        let productReview = {
          name: this.name,
          review: this.review,
          rating: this.rating,
          recom: this.recom
        }
        this.$emit('review-submitted', productReview)

        this.name = ''
        this.review = ''
        this.rating = null
        this.recom = null
      }
```

ReviewList.js M

COMP-1842 > week5 > task11 > components > ReviewList.js > templa

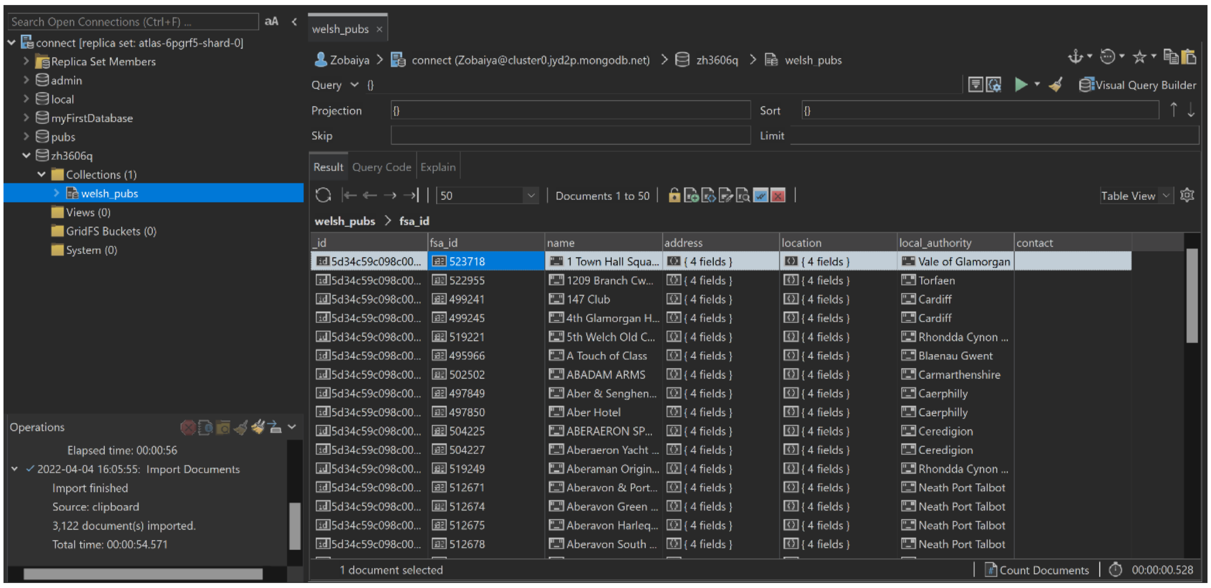
```javascript
app.component('review-list', {
    props: {
        reviews: {
            type: Array,
            required: true
        }
    },
    template:
        /*html*/
        `<div class="review-container">
    <h3>Reviews:</h3>
      <ul>
        <li v-for="(review, index) in reviews" :key="index">
          {{ review.name }} gave this {{ review.rating }} stars!
          <br/>
          Their review: "{{ review.review }}"
          <br/>
          Would they recommend this? {{ review.recom }}
          <br/>
        </li>
      </ul>
    </div>`
})
```

Figure 14: week 5, lesson 11 code

Week 5 Explanation

In lesson 8, I have learnt to update the variant image and whether it's in stock or not by using computed properties. For example, I have combined the brand and the product in the data property by using an expression with both of them in it at first. Afterwards I have added a computed option to the app and created the title property in the main.js file. This allowed the app to compute the value for me instead. Then I have added a quantity property which would determine whether the product was in stock or not by adding a new method called updateVariant(). In lesson 9, I have learnt about components and props so following the steps I have added a components folder with the separate component files. I then cleaned up each file and added their respective codes and fixed main.js which at the end only had cart and methods options for later. Props are custom attributes which lets us pass data in a component. Using props, I have added a premium feature for the component which means users who are premium users do not have to pay for shipping. If they are not premium users then they would have to pay for shipping. In lesson 10, I have learnt about emitting an event and adding a product ID to the cart so that the cart contains the IDs of the products that are added to it. For the coding challenge, I have added a button that removes a product from the cart. In lesson 11, I have learnt about forms - I added a form for users to add reviews about the product at the bottom of the page. I have used the v-model directive to create two-way data binding.

Lesson 4: MongoDB Basics



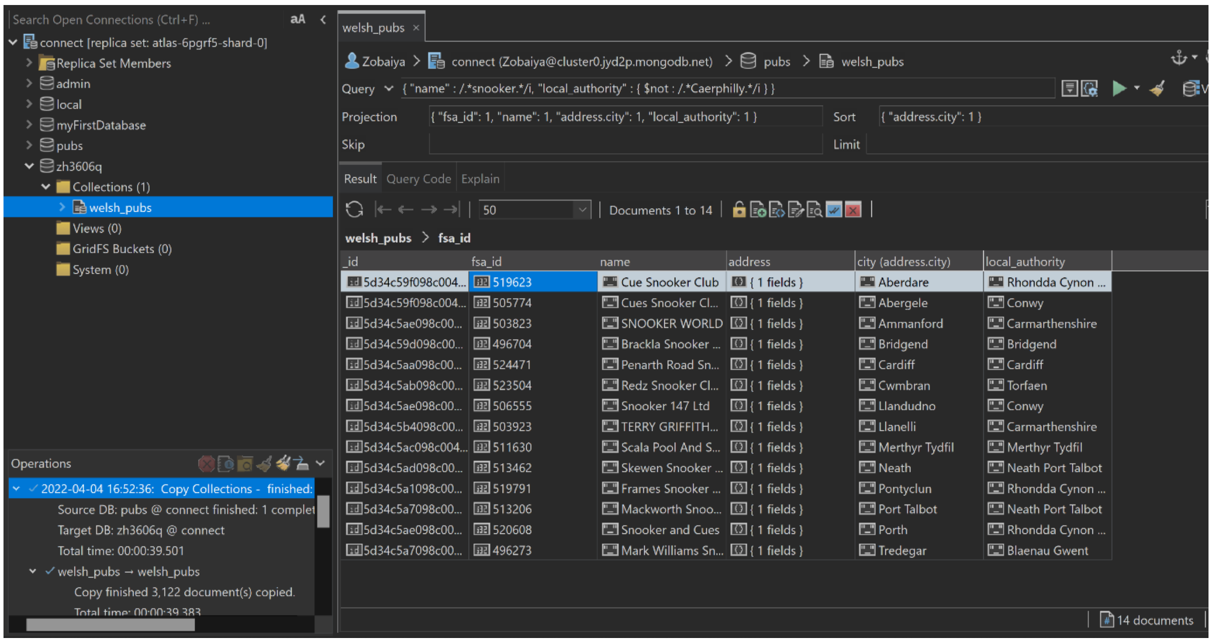
```
Search Open Connections (Ctrl+F) ...
connect [replica set: atlas-6pgrf5-shard-0]
  Replica Set Members
  admin
  local
  myFirstDatabase
  pubs
  zh3606q
    Collections (1)
      welsh_pubs
    Views (0)
    GridFS Buckets (0)
    System (0)

Operations
  Elapsed time: 00:00:56
  2022-04-04 16:05:55: Import Documents
    Import finished
    Source: clipboard
    3,122 document(s) imported.
    Total time: 00:00:54.571
```

```
welsh_pubs
Zobaiya > connect (Zobaiya@cluster0.jyd2p.mongodb.net) > zh3606q > welsh_pubs
Query       {}
Projection  {}                Sort   {}
Skip                          Limit
Result | Query Code | Explain
50   Documents 1 to 50                                   Table View
welsh_pubs > fsa_id
```

_id	fsa_id	name	address	location	local_authority	contact
5d34c59c098c00...	523718	1 Town Hall Squa...	{ 4 fields }	{ 4 fields }	Vale of Glamorgan	
5d34c59c098c00...	522955	1209 Branch Cw...	{ 4 fields }	{ 4 fields }	Torfaen	
5d34c59c098c00...	499241	147 Club	{ 4 fields }	{ 4 fields }	Cardiff	
5d34c59c098c00...	499245	4th Glamorgan H...	{ 4 fields }	{ 4 fields }	Cardiff	
5d34c59c098c00...	519221	5th Welch Old C...	{ 4 fields }	{ 4 fields }	Rhondda Cynon ...	
5d34c59c098c00...	495966	A Touch of Class	{ 4 fields }	{ 4 fields }	Blaenau Gwent	
5d34c59c098c00...	502502	ABADAM ARMS	{ 4 fields }	{ 4 fields }	Carmarthenshire	
5d34c59c098c00...	497849	Aber & Senghen...	{ 4 fields }	{ 4 fields }	Caerphilly	
5d34c59c098c00...	497850	Aber Hotel	{ 4 fields }	{ 4 fields }	Caerphilly	
5d34c59c098c00...	504225	ABERAERON SP...	{ 4 fields }	{ 4 fields }	Ceredigion	
5d34c59c098c00...	504227	Aberaeron Yacht ...	{ 4 fields }	{ 4 fields }	Ceredigion	
5d34c59c098c00...	519249	Aberaman Origin...	{ 4 fields }	{ 4 fields }	Rhondda Cynon ...	
5d34c59c098c00...	512671	Aberavon & Port...	{ 4 fields }	{ 4 fields }	Neath Port Talbot	
5d34c59c098c00...	512674	Aberavon Green ...	{ 4 fields }	{ 4 fields }	Neath Port Talbot	
5d34c59c098c00...	512675	Aberavon Harleq...	{ 4 fields }	{ 4 fields }	Neath Port Talbot	
5d34c59c098c00...	512678	Aberavon South ...	{ 4 fields }	{ 4 fields }	Neath Port Talbot	

```
1 document selected                     Count Documents   00:00:00.528
```

Figure 15: lesson 4, exercise 1



```
Search Open Connections (Ctrl+F) ...
connect [replica set: atlas-6pgrf5-shard-0]
  Replica Set Members
  admin
  local
  myFirstDatabase
  pubs
  zh3606q
    Collections (1)
      welsh_pubs
    Views (0)
    GridFS Buckets (0)
    System (0)

Operations
  2022-04-04 16:52:36: Copy Collections - finished:
    Source DB: pubs @ connect finished: 1 complet
    Target DB: zh3606q @ connect
    Total time: 00:00:39.501
    welsh_pubs → welsh_pubs
      Copy finished 3,122 document(s) copied.
      Total time: 00:00:39.383
```

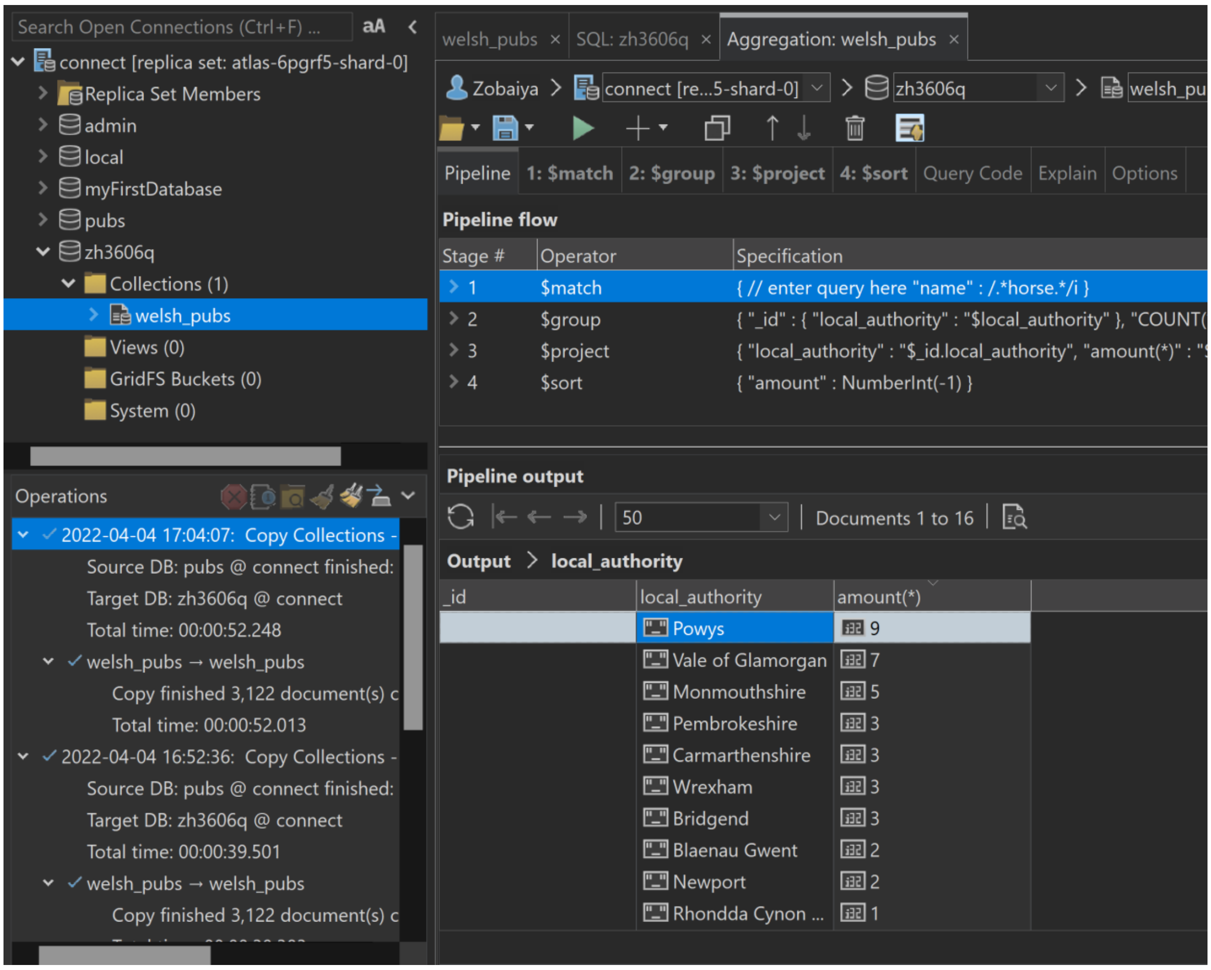
```
welsh_pubs
Zobaiya > connect (Zobaiya@cluster0.jyd2p.mongodb.net) > pubs > welsh_pubs
Query       { "name" : /.*snooker.*/i, "local_authority" : { $not : /.*Caerphilly.*/i } }
Projection  { "fsa_id": 1, "name": 1, "address.city": 1, "local_authority": 1 }   Sort  { "address.city": 1 }
Skip                                                                               Limit
Result | Query Code | Explain
50   Documents 1 to 14
welsh_pubs > fsa_id
```

_id	fsa_id	name	address	city (address.city)	local_authority
5d34c59f098c004...	519623	Cue Snooker Club	{ 1 fields }	Aberdare	Rhondda Cynon ...
5d34c59f098c004...	505774	Cues Snooker Cl...	{ 1 fields }	Abergele	Conwy
5d34c5ae098c00...	503823	SNOOKER WORLD	{ 1 fields }	Ammanford	Carmarthenshire
5d34c59d098c00...	496704	Brackla Snooker ...	{ 1 fields }	Bridgend	Bridgend
5d34c5aa098c00...	524471	Penarth Road Sn...	{ 1 fields }	Cardiff	Cardiff
5d34c5ab098c00...	523504	Redz Snooker Cl...	{ 1 fields }	Cwmbran	Torfaen
5d34c5ae098c00...	506555	Snooker 147 Ltd	{ 1 fields }	Llandudno	Conwy
5d34c5b4098c00...	503923	TERRY GRIFFITH...	{ 1 fields }	Llanelli	Carmarthenshire
5d34c5ac098c004...	511630	Scala Pool And S...	{ 1 fields }	Merthyr Tydfil	Merthyr Tydfil
5d34c5ad098c00...	513462	Skewen Snooker ...	{ 1 fields }	Neath	Neath Port Talbot
5d34c5a1098c00...	519791	Frames Snooker ...	{ 1 fields }	Pontyclun	Rhondda Cynon ...
5d34c5a7098c00...	513206	Mackworth Snoo...	{ 1 fields }	Port Talbot	Neath Port Talbot
5d34c5ae098c00...	520608	Snooker and Cues	{ 1 fields }	Porth	Rhondda Cynon ...
5d34c5a7098c00...	496273	Mark Williams Sn...	{ 1 fields }	Tredegar	Blaenau Gwent

```
14 documents
```

Figure 16: lesson 4, exercise 3

Lesson 5: SQL in MongoDB Aggregation



```
Search Open Connections (Ctrl+F) ...
connect [replica set: atlas-6pgrf5-shard-0]
  Replica Set Members
  admin
  local
  myFirstDatabase
  pubs
  zh3606q
    Collections (1)
      welsh_pubs
    Views (0)
    GridFS Buckets (0)
    System (0)

Operations
  ✓ 2022-04-04 17:04:07:  Copy Collections -
      Source DB: pubs @ connect finished:
      Target DB: zh3606q @ connect
      Total time: 00:00:52.248
    ✓ welsh_pubs → welsh_pubs
        Copy finished 3,122 document(s) c
        Total time: 00:00:52.013
  ✓ 2022-04-04 16:52:36:  Copy Collections -
      Source DB: pubs @ connect finished:
      Target DB: zh3606q @ connect
      Total time: 00:00:39.501
    ✓ welsh_pubs → welsh_pubs
        Copy finished 3,122 document(s) c
```

welsh_pubs × | SQL: zh3606q × | Aggregation: welsh_pubs ×

Zobaiya > connect [re...5-shard-0] > zh3606q > welsh_pu

Pipeline | 1: $match | 2: $group | 3: $project | 4: $sort | Query Code | Explain | Options

Pipeline flow

Stage #	Operator	Specification
1	$match	{ // enter query here "name" : /.*horse.*/i }
2	$group	{ "_id" : { "local_authority" : "$local_authority" }, "COUNT(
3	$project	{ "local_authority" : "$_id.local_authority", "amount(*)" : "$
4	$sort	{ "amount" : NumberInt(-1) }

Pipeline output

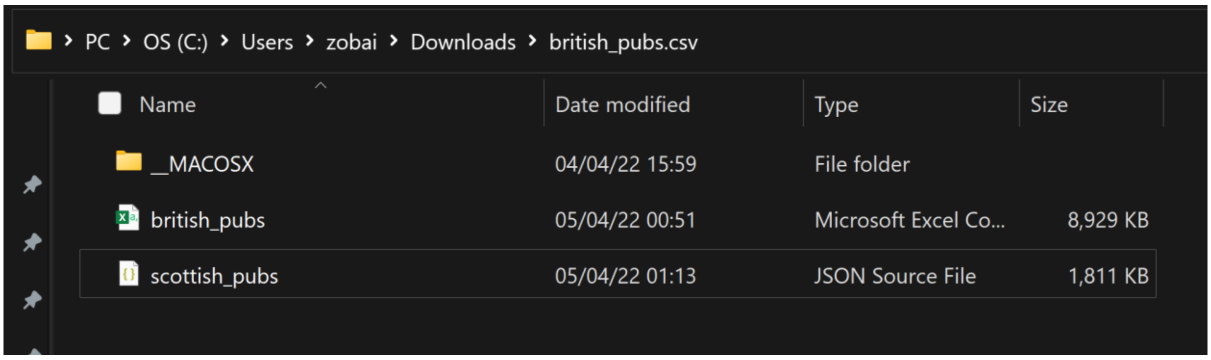
50 | Documents 1 to 16

Output > local_authority

_id	local_authority	amount(*)
	Powys	9
	Vale of Glamorgan	7
	Monmouthshire	5
	Pembrokeshire	3
	Carmarthenshire	3
	Wrexham	3
	Bridgend	3
	Blaenau Gwent	2
	Newport	2
	Rhondda Cynon ...	1

Figure 17: lesson 5, exercise 3

Lesson 6: Importing/Exporting MongoDB Data



PC > OS (C:) > Users > zobai > Downloads > british_pubs.csv

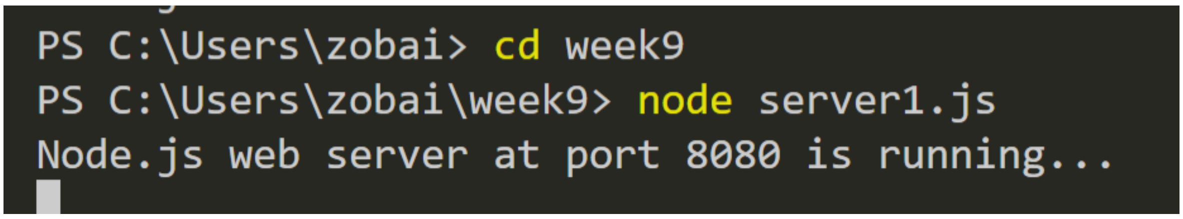
Name	Date modified	Type	Size
__MACOSX	04/04/22 15:59	File folder	
british_pubs	05/04/22 00:51	Microsoft Excel Co...	8,929 KB
scottish_pubs	05/04/22 01:13	JSON Source File	1,811 KB

Figure 18: lesson 6, exercise 2

Week 8 Explanation

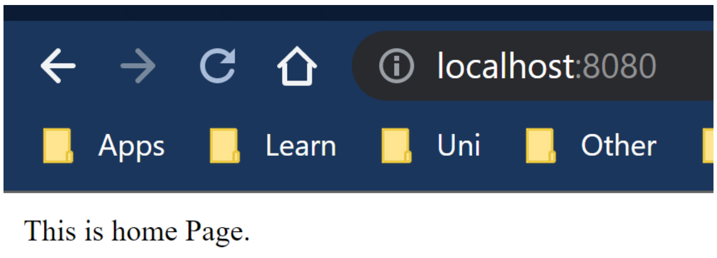
In lesson 4, I have learnt about MongoDB which is a NoSQL database program and how to use Studio 3T to connect my database. I have created a collection and learnt about the different MongoDB data types. I then have used Studio 3T to only view the data/edit the data that is needed from the hundreds of data submitted in the file provided with the course. In lesson 5, I have learnt to use the SQL Query command prompt which was easy as I had the knowledge from using it previously in a different project. I have also edited queries in the aggregation editor which has allowed me to view lesser amount of data. Lesson 6 was about exporting data to a .json file which was pretty simple as well, I could have figured it out myself. NoSQL has a lot of advantages such as horizontal scaling which is adding more instances of a machine instead of increasing the capacity of one instead. The queries in NoSQL databases can also be faster than those in SQL databases because the data stored in NoSQL databases is stored together hence the queries are fast. Many components in MongoDB are similar to SQL, however instead of tables MongoDB stores data in BSON documents. The documents are fairly easy to create and write when storing information since you can add groups of information in one place, whereas in SQL you have to create separate tables to join them together.

Week 9



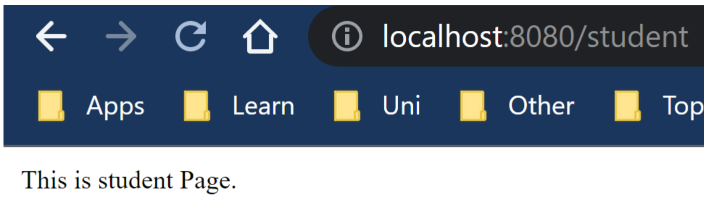
```
PS C:\Users\zobai> cd week9
PS C:\Users\zobai\week9> node server1.js
Node.js web server at port 8080 is running...
```

Figure 19: screenshot 1



This is home Page.

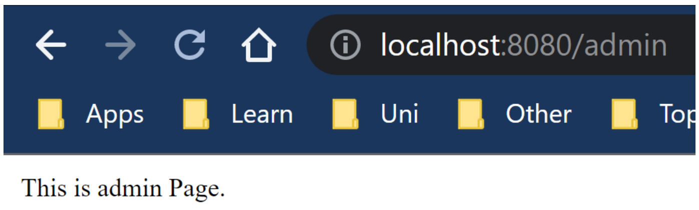
Figure 20: screenshot 2


localhost:8080/student

Apps Learn Uni Other Top

This is student Page.

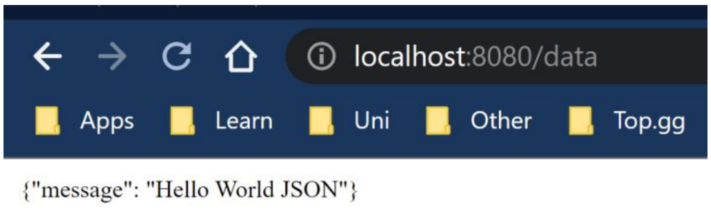
Figure 21: screenshot 3


localhost:8080/admin

Apps Learn Uni Other Top

This is admin Page.

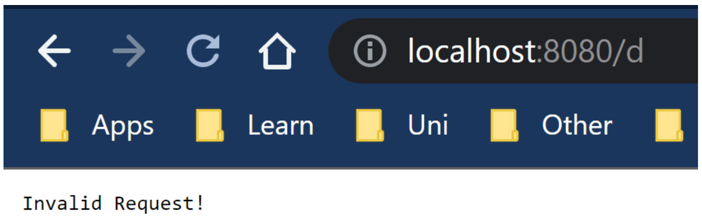
Figure 22: screenshot 4


localhost:8080/data

Apps Learn Uni Other Top.gg

```
{"message": "Hello World JSON"}
```

Figure 23: screenshot 5


localhost:8080/d

Apps Learn Uni Other

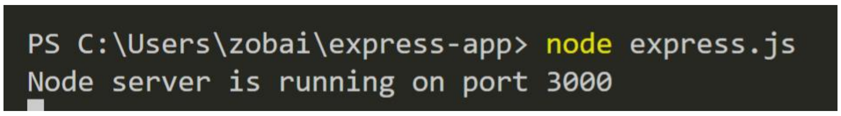
```
Invalid Request!
```

Figure 24: screenshot 6

Week 9 Explanation

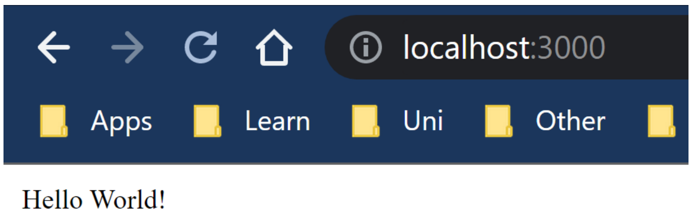
In week 9 I have learnt about Node.js which can be used to build different types of applications (in this case, web applications). Node.js handles file requests differently than PHP/ASP, as it executes it faster. I have used Node.js to create a backend server and got it running so I could create a simple web server. Node.js can collect form data, add/delete/edit data, can generate page content and the files' extensions are .js. Node.js also allows me to create my own web server to handle the HTTP requests asynchronously. It has a built-in module called HTTP which can create a server that listens to ports.

Week 10



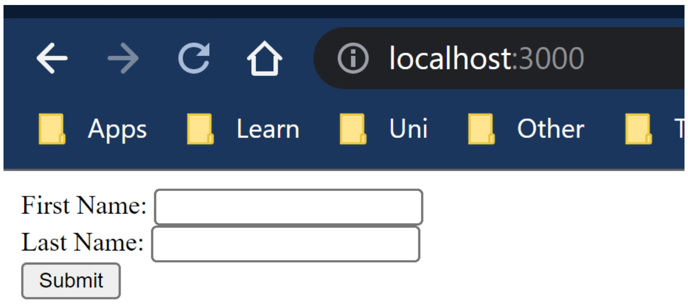
```
PS C:\Users\zobai\express-app> node express.js
Node server is running on port 3000
```

Figure 25: screenshot 1, slide 11



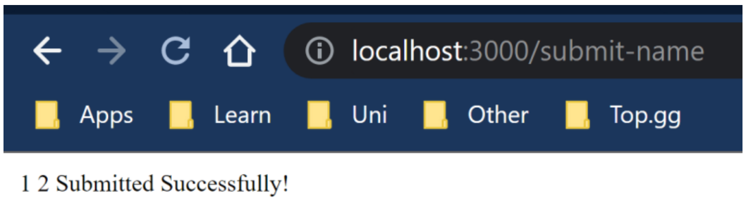
Hello World!

Figure 26: screenshot 2, slide 11



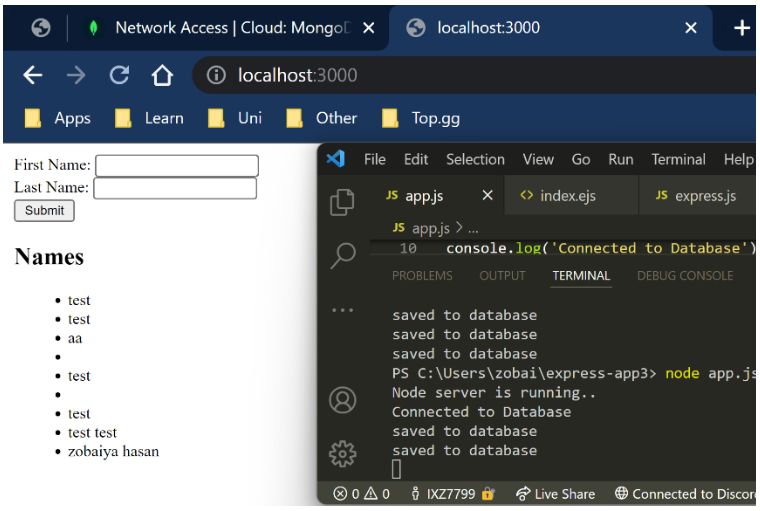
First Name:
Last Name:
Submit

Figure 27: screenshot 3, slide 19



1 2 Submitted Successfully!

Figure 28: screenshot 4, slide 19



First Name:
Last Name:
Submit

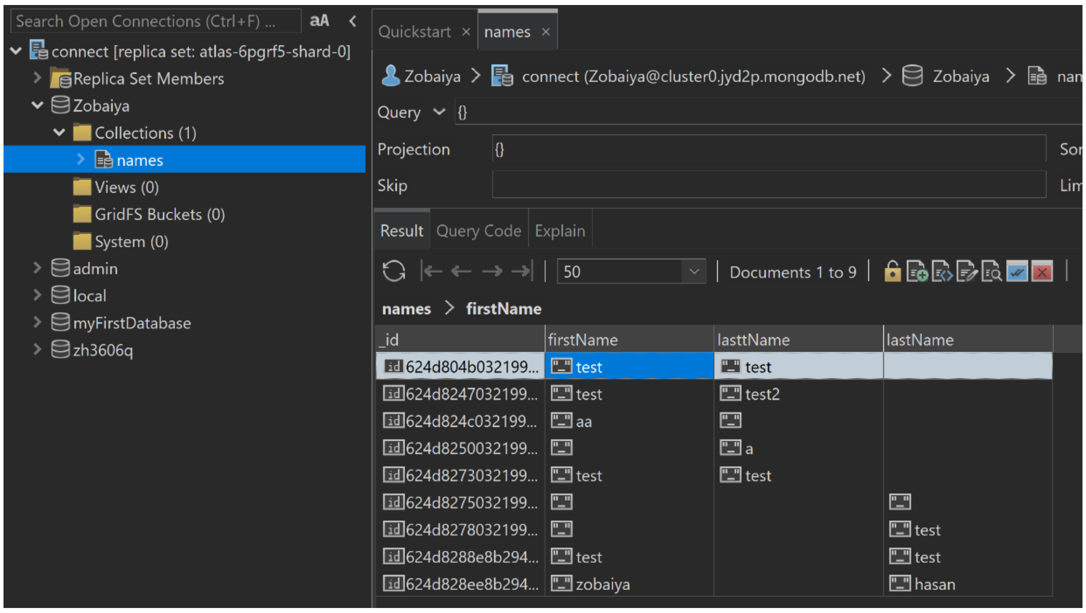
Names

- test
- test
- aa
-
- test
-
- test
- test test
- zobaiya hasan

```
app.js > ...
    console.log('Connected to Database')
```

```
saved to database
saved to database
saved to database
PS C:\Users\zobai\express-app3> node app.js
Node server is running..
Connected to Database
saved to database
saved to database
```

Figure 29: screenshot 5, slide 36



Search Open Connections (Ctrl+F) ...

- connect [replica set: atlas-6pgrf5-shard-0]
  - Replica Set Members
  - Zobaiya
    - Collections (1)
      - names
    - Views (0)
    - GridFS Buckets (0)
    - System (0)
  - admin
  - local
  - myFirstDatabase
  - zh3606q

Zobaiya > connect (Zobaiya@cluster0.jyd2p.mongodb.net) > Zobaiya > nan

Query {}

Projection {}

Skip

Result | Query Code | Explain

50 | Documents 1 to 9

names > firstName

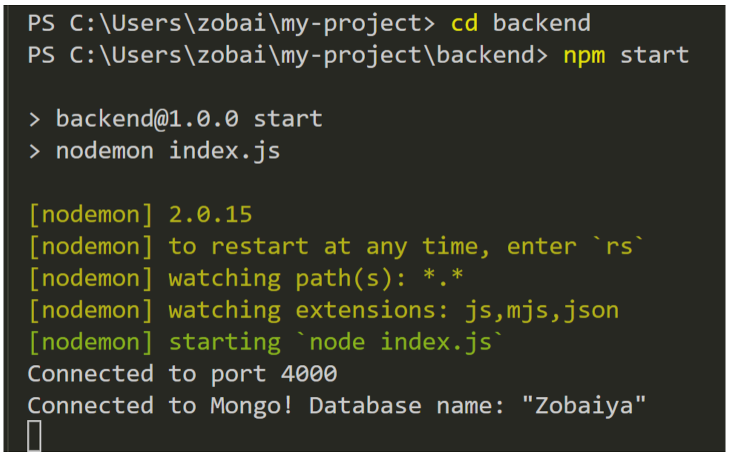
_id	firstName	lasttName	lastName
624d804b032199...	test	test	
624d8247032199...	test	test2	
624d824c032199...	aa		
624d8250032199...		a	
624d8273032199...	test	test	
624d8275032199...			
624d8278032199...			test
624d8288e8b294...	test		test
624d828ee8b294...	zobaiya		hasan

Note: I wrote "lasttName" hence it wasn't working before

Figure 30: screenshot 6, Studio 3T

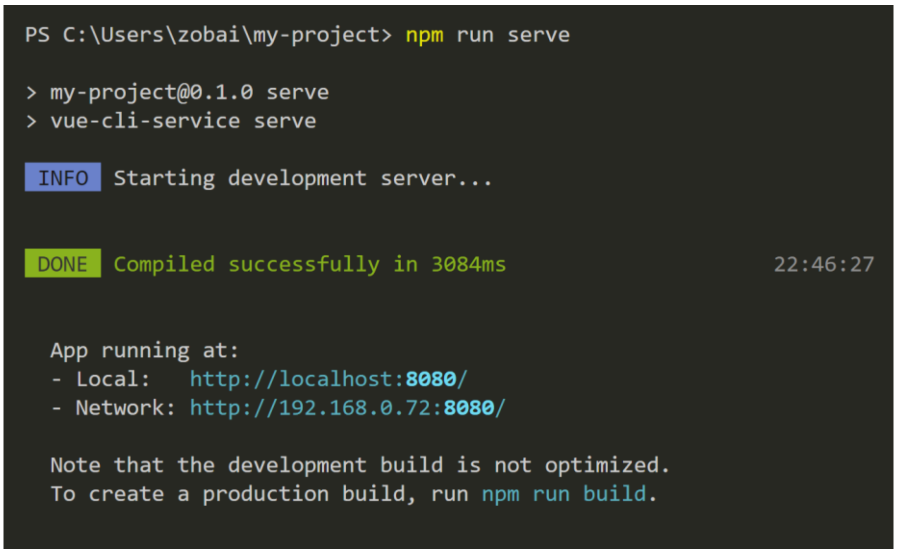
Week 10 Explanation

In week 10, I have learnt to create a basic add/read app using Express which is a backend web framework for Node.js. Express is also an unopinionated software which means it leaves developers with a lot of flexibility when using tools. I have used Express to run a localhost path and a CRUD system (create, read, update and delete) to create a basic app. In the app, you can submit your name and surname in the webpage which will save in the database (showed in Studio 3T) and it shows up in the webpage itself. I ran this webpage in my own laptop so the process to connect to MongoDB was a bit tedious - it took a few tries to get the connection working correctly.



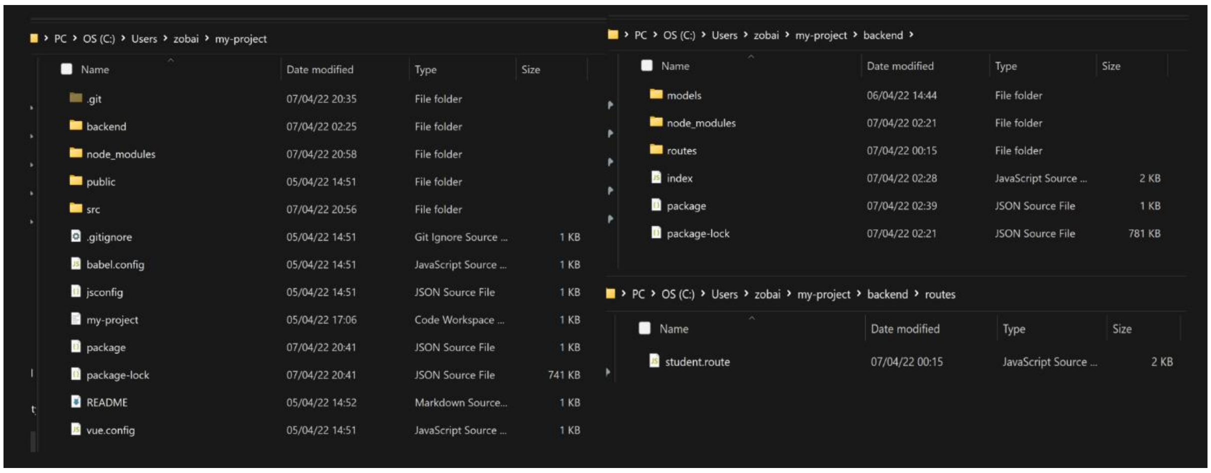
```
PS C:\Users\zobai\my-project> cd backend
PS C:\Users\zobai\my-project\backend> npm start

> backend@1.0.0 start
> nodemon index.js

[nodemon] 2.0.15
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Connected to port 4000
Connected to Mongo! Database name: "Zobaiya"
```

Figure 31: backend start-up



```
PS C:\Users\zobai\my-project> npm run serve

> my-project@0.1.0 serve
> vue-cli-service serve

 INFO  Starting development server...


 DONE  Compiled successfully in 3084ms                       22:46:27


  App running at:
  - Local:   http://localhost:8080/
  - Network: http://192.168.0.72:8080/

  Note that the development build is not optimized.
  To create a production build, run npm run build.
```

Figure 32: frontend start-up



PC › OS (C:) › Users › zobai › my-project

Name	Date modified	Type	Size
.git	07/04/22 20:35	File folder	
backend	07/04/22 02:25	File folder	
node_modules	07/04/22 20:58	File folder	
public	05/04/22 14:51	File folder	
src	07/04/22 20:56	File folder	
.gitignore	05/04/22 14:51	Git Ignore Source ...	1 KB
babel.config	05/04/22 14:51	JavaScript Source ...	1 KB
jsconfig	05/04/22 14:51	JSON Source File	1 KB
my-project	05/04/22 17:06	Code Workspace ...	1 KB
package	07/04/22 20:41	JSON Source File	1 KB
package-lock	07/04/22 20:41	JSON Source File	741 KB
README	05/04/22 14:52	Markdown Source...	1 KB
vue.config	05/04/22 14:51	JavaScript Source ...	1 KB

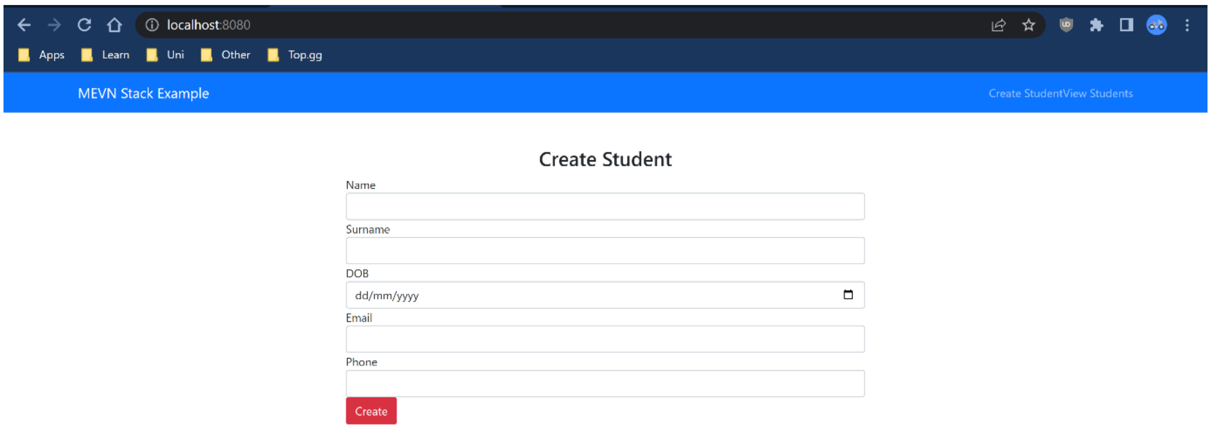
PC › OS (C:) › Users › zobai › my-project › backend ›

Name	Date modified	Type	Size
models	06/04/22 14:44	File folder	
node_modules	07/04/22 02:21	File folder	
routes	07/04/22 00:15	File folder	
index	07/04/22 02:28	JavaScript Source ...	2 KB
package	07/04/22 02:39	JSON Source File	1 KB
package-lock	07/04/22 02:21	JSON Source File	781 KB

PC › OS (C:) › Users › zobai › my-project › backend › routes

Name	Date modified	Type	Size
student.route	07/04/22 00:15	JavaScript Source ...	2 KB

Figure 33: completed file structure (backend)



Create Student

Name

Surname

DOB

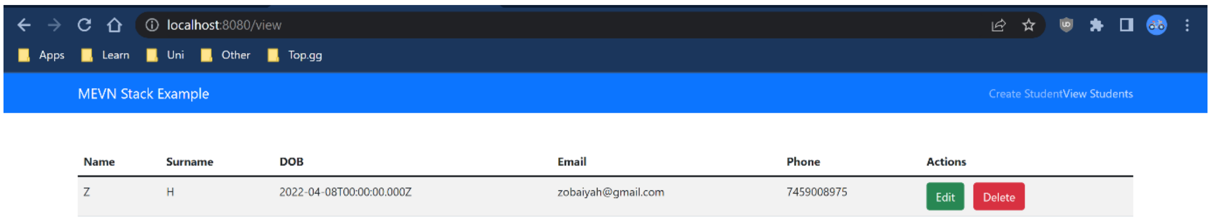
dd/mm/yyyy

Email

Phone

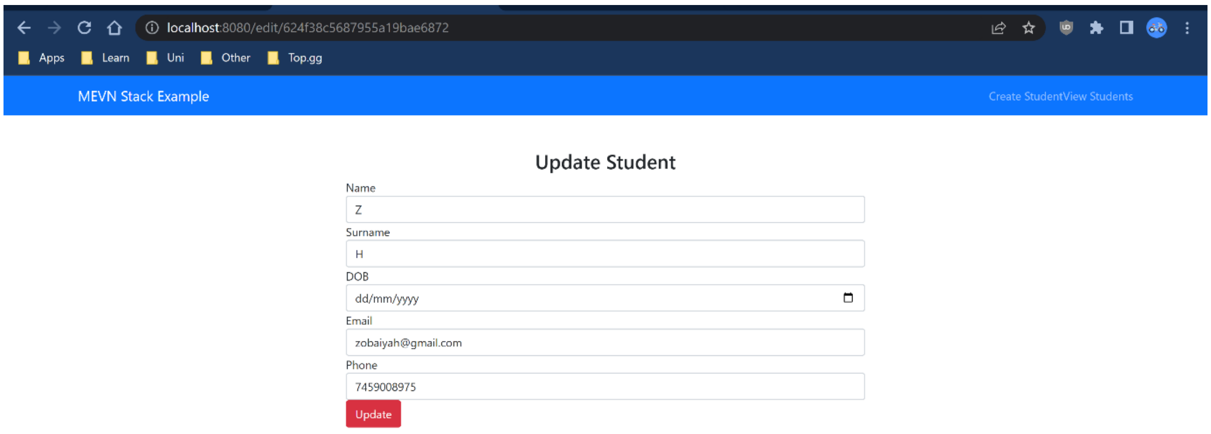
Create

Figure 34: home page running (extra fields added)



Name	Surname	DOB	Email	Phone	Actions
Z	H	2022-04-08T00:00:00.000Z	zobaiyah@gmail.com	7459008975	Edit Delete

Figure 35: view route and data displayed



Update Student

Name

Z

Surname

H

DOB

dd/mm/yyyy

Email

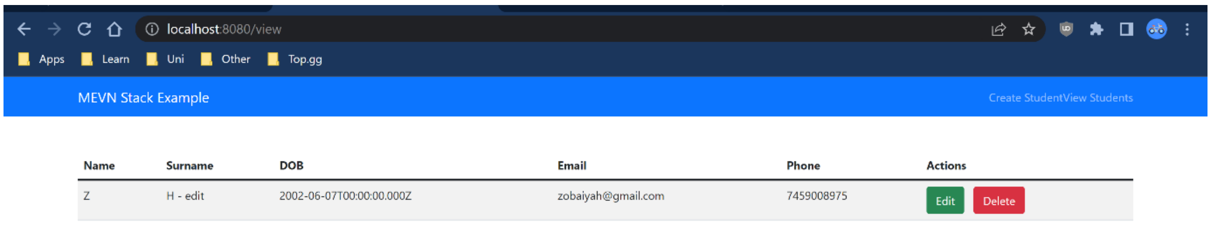
zobaiyah@gmail.com

Phone

7459008975

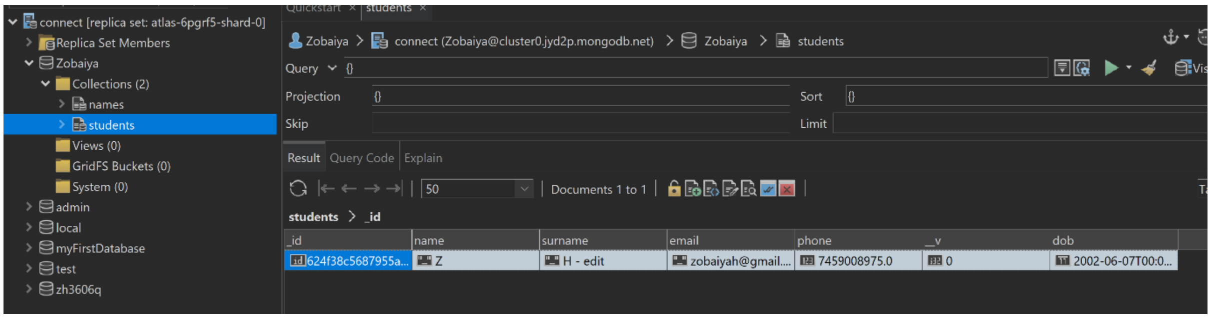
Update

Figure 36: updating details



Name	Surname	DOB	Email	Phone	Actions
Z	H - edit	2002-06-07T00:00:00.000Z	zobaiyah@gmail.com	7459008975	Edit Delete

Figure 37: details updated



_id	name	surname	email	phone	__v	dob
624f38c5687955a...	Z	H - edit	zobaiyah@gmail....	7459008975.0	0	2002-06-07T00:0...

Figure 38: data displayed in database

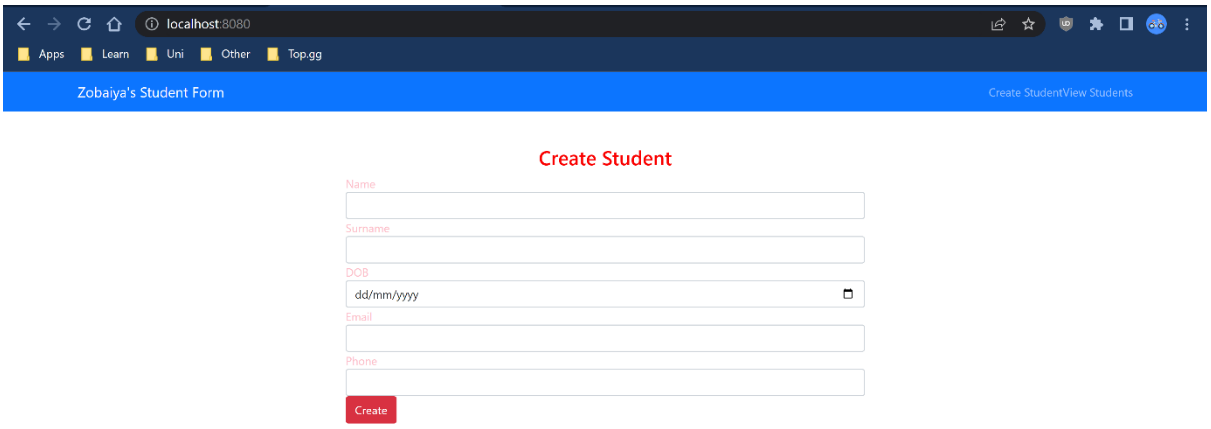


Figure 39: overridden bootstrap css file, edited page header

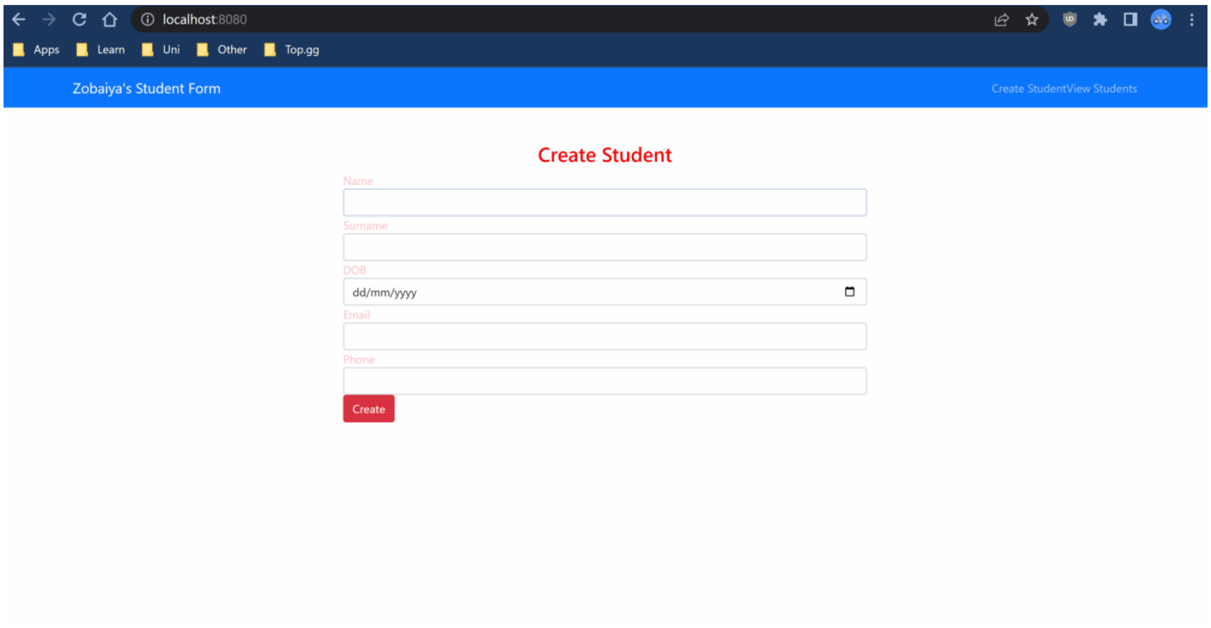


Figure 40: gif of working web system

Report

RDBMS is a Relational Database Management System which is "a program that allows you to create, update, and administer a relational database" (Codecademy, n.d.). NoSQL database systems are non-relational database systems. MEVN stack stands for MongoDB, Express.js, Vue.js and Node.js which is a software that helps developers to build modern web applications. XAMPP stack stands for cross platform, Apache, MySQL, PHP & Perl which allows you to build websites on a local web server. In this report, I will be discussing the key differences between RDBMS and NoSQL database systems, as well as discussing the differences between MEVN stack and XAMPP stack development.

A relational database is a type of database that has pre-determined relationships between a collection of data items such as SQL. Relational databases link information in different tables by using keys such as primary and foreign keys. Primary keys are relational database columns that are used to identify table records. When the primary key is added to a different table, it becomes a foreign key which connects both tables hence creating a relationship between the two. On the other hand, non-relational databases (NoSQL) don't use tables or keys – they use documents which are stored in the form of JSON or other ways like BSON. There are four types of NoSQL databases: document databases, key-value databases, wide-column stores, and graph databases. These types of non-relational databases have specific requirements unlike the relational databases which are based on relationships between the data.

Scaling is a process to "expand the existing configuration (servers/computers) to handle a large number of user requests or to manage the amount of load on the server" (Rout, 2020). There are two types of scaling, horizontal and vertical. Vertical scaling is the process of "increasing the capacity of a single machine by adding more resources such as memory" (Rout, 2020). An example of vertical scaling is MySQL. Horizontal scaling is the process of "adding more instances of the same type to the existing pool of resources" (Rout, 2020). An example of horizontal scaling is NoSQL. There are advantages and disadvantages to both methods such as horizontal scaling does not have a single point of failure since there are multiple servers; whereas vertical scaling is at a disadvantage since if the system fails, the waiting time is longer because there's only one server. Another advantage of using vertical scaling is that there is only one server so it is easier to maintain whereas using horizontal scaling can get complicated due to the multiple servers.

Relational databases and non-relational databases also have different schemas – in NoSQL the schemas are flexible which means the documents do not need to have the same schema whereas in SQL databases you must declare the tables' schemas before inserting data. The queries in non-relational databases are also faster than queries in relational databases. This is because in SQL, as the tables grow, the relationships and joins between each become more complex. On the other hand, in NoSQL, databases are stored in a way that is optimised (for example when using key-value databases). If a user wanted to store information in a relational database, they would have to create separate tables and join them together. If they used a non-relational database, they could potentially group all the information in a single document – no joins are required hence there will be faster queries than relational databases.

When deciding which of the two databases to use, I feel NoSQL is much easier to use due to the amount of data you can store in it, the simple document or key-value databases which are easy to implement or even because of the advantages of using horizontal scaling. I am personally more comfortable with relational databases as I have worked on many projects using SQL, so the usage of tables and keys are simple. Both relational and non-relational databases have their advantages and disadvantages, and both are as good as each other. Due to this fact, I believe when you must choose which to use, you should look at the requirements of the project and which of the two would best suit the project.

In my previous project, I have used the XAMPP stack to create a simple website for students and teachers/administrators. Students can save questions on the website for all users to see. Administrators can manage the authors, modules, and the questions from a hidden page. They would need to sign up/log in to access the administration section. This was all created using PHP and phpMyAdmin to create the tables and relationships between them. In this project, I have developed a CRUD web application using MEVN stack. In phpMyAdmin, I had to create the tables and the relationships between each using primary and foreign keys whereas in MongoDB (a document-oriented database), the collections and documents are created automatically when data is entered. Documents within the same collection can have different structures whereas in an SQL database you must stick to specific tables. This allows MongoDB to give more flexibility to the user when creating a database.

A similarity between phpMyAdmin and MongoDB is that you can use SQL queries in both to retrieve specific sets of data. You can group the data, counting the total amount of data, etc. in both. However, aggregated queries in MongoDB and SQL are different in several ways. SQL has much simpler commands such as "SELECT", "GROUP BY" or "COUNT()" whereas in MongoDB, the aggregation operators are much different such as "$group" and "$project". In figure 17 I have shown how I used SQL aggregation to show the amount of each pub that exists in different authorities. In MongoDB I used four stages to get the amount, whereas if I used SQL, I would have to run fewer queries to get the amount of pubs.

MongoDB was a big part when creating this project. I have used Studio 3T which "provides a graphical user interface (GUI) and integrated development environment (IDE) for accessing and modifying MongoDB databases and their documents" (Studio 3T, n.d.). The cloud service, MongoDB Atlas handles managing users' deployments on a cloud provider. You first have to create a cluster and choose a cloud provider and region. After you have deployed a database you can secure it by adding IP access points and manage who can access to the database as well. I connected to the database by using Node.js – I included the connection string into the backend code. Studio 3T was fairly easy to use – their documents and tutorials explained in depth the basics of MongoDB and they also had a more advanced course. It has an aggregation editor which helps with writing complex queries and it has a feature that allows you to separate the different stages as shown in figure 17. When comparing MongoDB against phpMyAdmin, I found that MongoDB was harder to set up than phpMyAdmin but it was much easier to use.

When using MEVN stack, I have used Node.js to run the backend of the project as shown in figure 31. In the project where I used XAMPP, I only had to run the Apache HTTP server through the click of a button and PHP to run the backend. PHP is much older than Node.js so it is more popular and known however due to its vertical scaling issues it is quickly becoming less and less favourable than Node.js. Node.js provides better packages and is increasing in popularity due to its ability to keep up with the modern projects. Node.js also uses JavaScript which makes it easier for developers "to maintain their backend and frontend logic of developing web applications" (Kaneriya, 2020). Both Node.js and PHP are both useful, however Node.js is preferred more by developers as it has a higher speed and connectivity to the servers, and its performance with the MEVN stack in general is more optimal. Node.js is also asynchronous and functions on a main thread for execution whereas PHP runs on a multi thread blocking execution. They are both well suited to each developers' needs.

Vue.js is a frontend JavaScript framework for building web applications. It was released in 2014 by Evan You who used to work with Angular.js. I have used this to build my webpage about creating a form for students to sign up in where you can also view the created profiles. In figure 34 you can see my homepage running using the MEVN stack. Vue.js is used to build single page applications. SPAs are more user friendly in comparison to MPAs (multi-page applications) and they load faster as MPAs need to be reloaded each time there is an update to the data. I have used Vue.js for the first time in my project and since I have previous knowledge from C#, PHP and JavaScript, it was easy to pick up on. Vue.js comes with a lot of directives that developers use such as v-if, v-show, and v-model. There are also shorter forms for each directive such as ":href" for "v-bind".

It is slowly becoming the preferred JavaScript framework by developers as it has many useful libraries, it is user and developer friendly and it performs very quickly. In an article by Pšenák and Tibenský, it is shown how Vue.js is becoming popular in comparison to React.js or Angular.js. They have said that you can use Vue to slowly integrate it into existing web pages". This also means that companies can switch over to using Vue.js without having to remove the old web pages. Statistics from the first year that Vue.js was released show that Vue was getting popular and easy to learn. Since Vue is not backed by a large company, "decisions to where the framework is heading in the future" are made by the community and development team itself (Wohlgethan, 2018). However downsides to using Vue is that since it is still a new framework, there is not much support for it in comparison to React.js or Angular.js.

In conclusion, relational and non-relational databases are both useful and have advantages/disadvantages; but I prefer using relational databases since I am more familiar with it and have been using it for much longer. MEVN stack and XAMPP stack are also both compatible, however developers are leaning towards MEVN more due to it being more modern than XAMPP. When building web applications, I am more likely to start using the modern stacks and Vue due to its rising popularity in the developer world.

References:

Codecademy. (n.d.). *What is a Relational Database Management System?* [online] Available at: https://www.codecademy.com/article/what-is-rdbms-sql.

Kaneriya, T. (2020). *Node.js vs PHP: A Honest Comparative Study With All The Answers*. [online] Insights on Latest Technologies - Simform Blog. Available at: https://www.simform.com/blog/nodejs-vs-php/.

Pšenák, P. and Tibenský, M. (2020). The usage of Vue JS framework for web application creation. *Mesterséges intelligencia*, 2(2), pp.61–72. [online]

Rout, S. (2020). *Overview of Scaling: Vertical And Horizontal Scaling*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/overview-of-scaling-vertical-and-horizontal-scaling/#:~:text=It%20can%20be%20defined%20as.

Studio 3T. (n.d.). *Introduction to MongoDB and Studio 3T*. [online] Available at:

https://studio3t.com/academy/lessons/introduction-to-mongodb/.

Wohlgethan, E. (2018). *Supporting Web Development Decisions by Comparing Three Major JavaScript Frameworks: Angular, React and Vue.js*. [online] Entscheidungshilfe für die Webentwicklung anhand des Vergleichs von drei führenden JavaScript Frameworks: Angular, React and Vue.js. Available at: https://reposit.haw-hamburg.de/bitstream/20.500.12738/8417/1/BA_Wohlgethan_2176410.pdf.