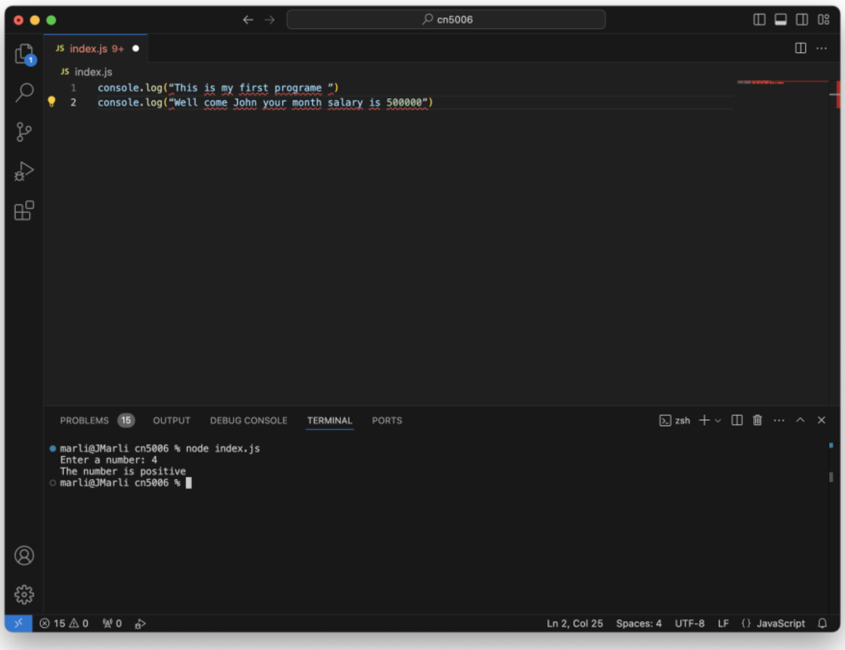
U2283556

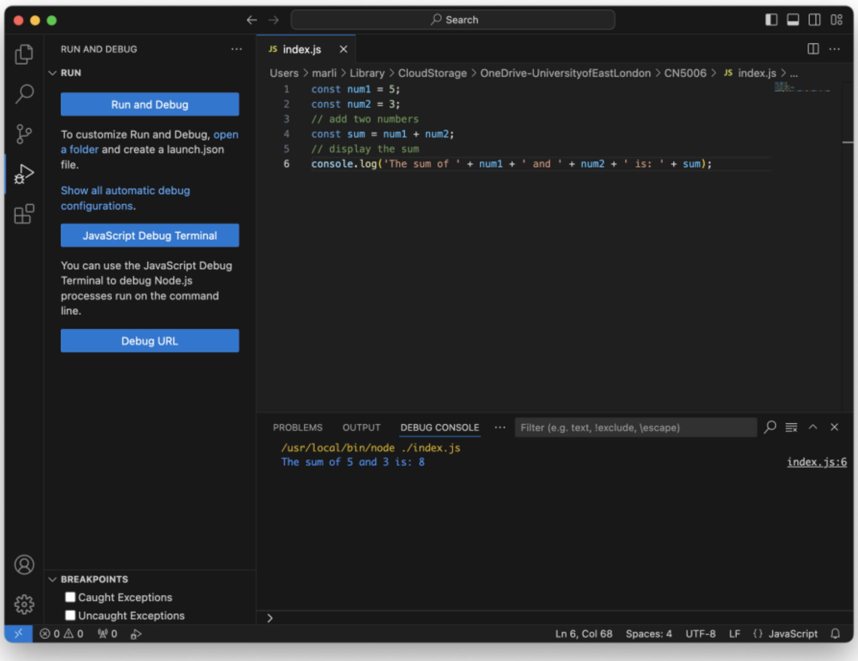
CN5006 PORTFOLIO:

Week 1 code:



```
console.log("This is my first programe ")
console.log("Well come John your month salary is 500000")
```

```
marli@JMarli cn5006 % node index.js
Enter a number: 4
The number is positive
marli@JMarli cn5006 %
```



```
const num1 = 5;
const num2 = 3;
// add two numbers
const sum = num1 + num2;
// display the sum
console.log('The sum of ' + num1 + ' and ' + num2 + ' is: ' + sum);
```

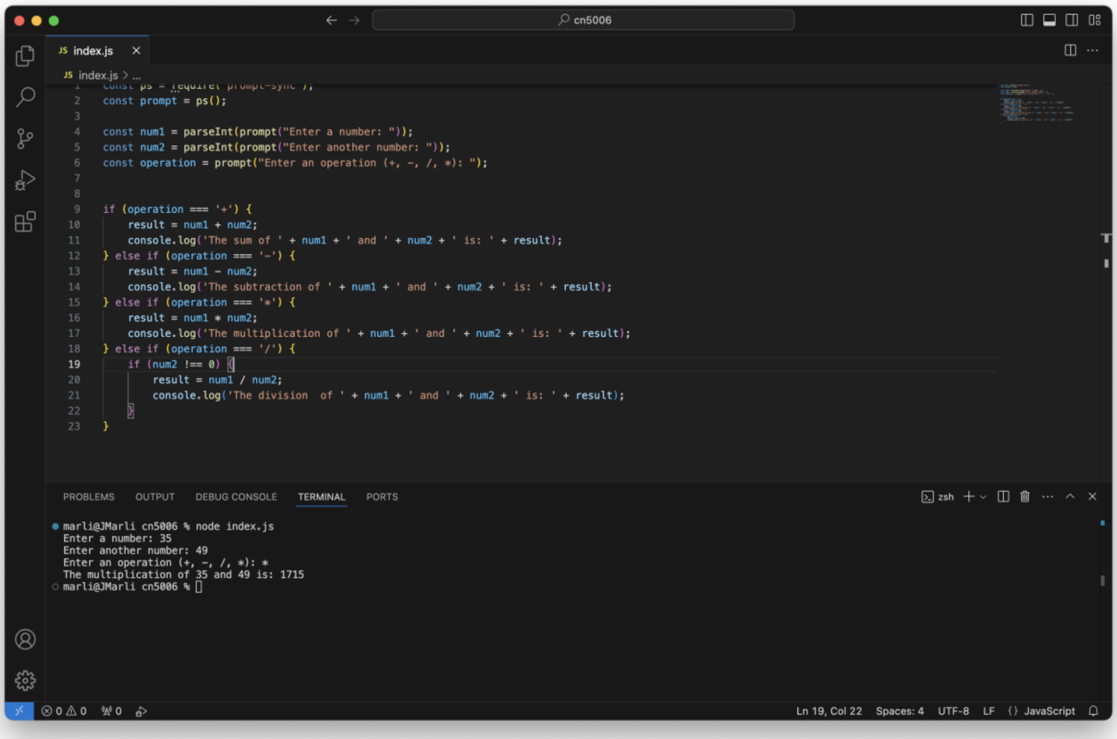
```
/usr/local/bin/node ./index.js
The sum of 5 and 3 is: 8                                index.js:6
```

Self-evaluation:

I have improved my java script language skills and will work more to improve them. This was a great refresher after the summer holiday.

U2283556

<u>Exercise:</u>



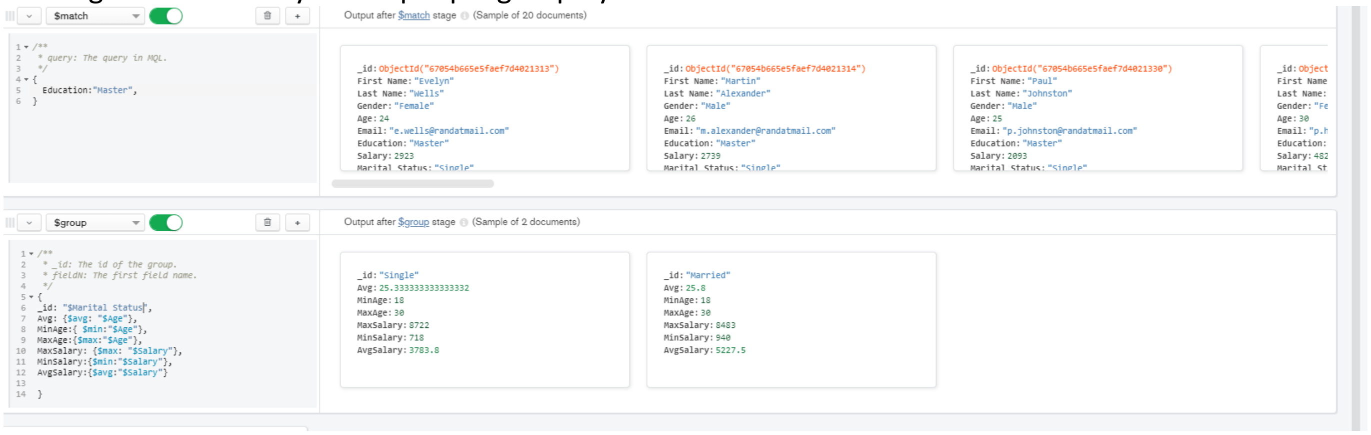
```
const ps = require('prompt-sync');
const prompt = ps();

const num1 = parseInt(prompt("Enter a number: "));
const num2 = parseInt(prompt("Enter another number: "));
const operation = prompt("Enter an operation (+, -, /, *): ");


if (operation === '+') {
    result = num1 + num2;
    console.log('The sum of ' + num1 + ' and ' + num2 + ' is: ' + result);
} else if (operation === '-') {
    result = num1 - num2;
    console.log('The subtraction of ' + num1 + ' and ' + num2 + ' is: ' + result);
} else if (operation === '*') {
    result = num1 * num2;
    console.log('The multiplication of ' + num1 + ' and ' + num2 + ' is: ' + result);
} else if (operation === '/') {
    if (num2 !== 0) {
        result = num1 / num2;
        console.log('The division  of ' + num1 + ' and ' + num2 + ' is: ' + result);
    }
}
```

```
marli@JMarli cn5006 % node index.js
Enter a number: 35
Enter another number: 49
Enter an operation (+, -, /, *): *
The multiplication of 35 and 49 is: 1715
marli@JMarli cn5006 %
```

This program was created to function as a calculator that is capable of four kinds of operations.

<u>Week 2 lab tasks:</u>

1) Repeat the same process to search Education for Master and .Find the avg,min,max age and avg min max Salary of the people group by Marital status.



$match

```
/**
 * query: The query in MQL.
 */
{
  Education:"Master",
}
```

Output after $match stage (Sample of 20 documents)

```
_id: ObjectId("67054b665e5faef7d4021313")
First Name: "Evelyn"
Last Name: "Wells"
Gender: "Female"
Age: 24
Email: "e.wells@randatmail.com"
Education: "Master"
Salary: 2923
Marital Status: "Single"
```

```
_id: ObjectId("67054b665e5faef7d4021314")
First Name: "Martin"
Last Name: "Alexander"
Gender: "Male"
Age: 26
Email: "m.alexander@randatmail.com"
Education: "Master"
Salary: 2739
Marital Status: "Single"
```

```
_id: ObjectId("67054b665e5faef7d4021330")
First Name: "Paul"
Last Name: "Johnston"
Gender: "Male"
Age: 25
Email: "p.johnston@randatmail.com"
Education: "Master"
Salary: 2093
Marital Status: "Single"
```

```
_id: Object
First Name
Last Name:
Gender: "Fe
Age: 30
Email: "p.h
Education:
Salary: 482
Marital St
```

$group

```
/**
 * _id: The id of the group.
 * fieldN: The first field name.
 */
{
 _id: "$Marital Status",
 Avg: {$avg: "$Age"},
 MinAge:{ $min:"$Age"},
 MaxAge:{$max:"$Age"},
 MaxSalary: {$max: "$Salary"},
 MinSalary:{$min:"$Salary"},
 AvgSalary:{$avg:"$Salary"}

 }
```

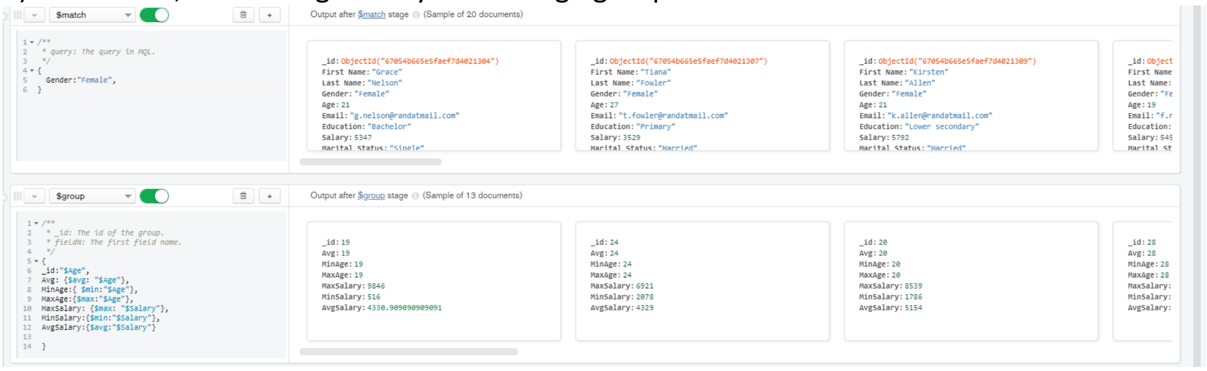
Output after $group stage (Sample of 2 documents)

```
_id: "Single"
Avg: 25.333333333333332
MinAge: 18
MaxAge: 30
MaxSalary: 8722
MinSalary: 718
AvgSalary: 3783.8
```

```
_id: "Married"
Avg: 25.8
MinAge: 18
MaxAge: 30
MaxSalary: 8483
MinSalary: 940
AvgSalary: 5227.5
```

U2283556

2) 2. find min,max average salary of each age group of female



Output after $match stage (Sample of 20 documents)

```
/**
 * query: The query in MQL.
 */
{
  Gender:"Female",
}
```

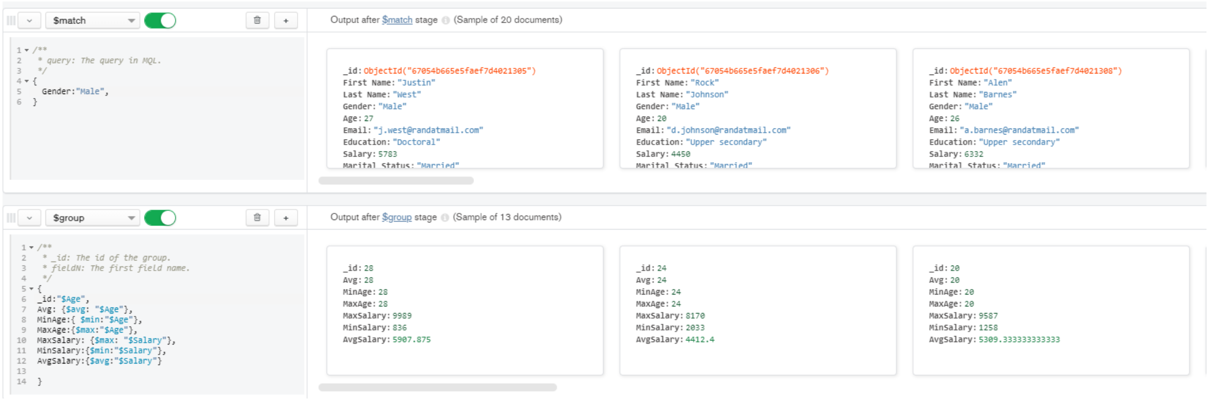
```
_id: ObjectId("67054b665e5faef7d4021304")
First Name: "Grace"
Last Name: "Nelson"
Gender: "Female"
Age: 21
Email: "g.nelson@randatmail.com"
Education: "Bachelor"
Salary: 5347
Marital Status: "Single"

_id: ObjectId("67054b665e5faef7d4021307")
First Name: "Tiana"
Last Name: "Fowler"
Gender: "Female"
Age: 27
Email: "t.fowler@randatmail.com"
Education: "Primary"
Salary: 3529
Marital Status: "Married"

_id: ObjectId("67054b665e5faef7d4021309")
First Name: "Kirsten"
Last Name: "Allen"
Gender: "Female"
Age: 21
Email: "k.allen@randatmail.com"
Education: "Lower secondary"
Salary: 5792
Marital Status: "Married"

_id: Object
First Name
Last Name:
Gender: "Fe
Age: 19
Email: "f.r
Education:
Salary: 549
Marital St
```

Output after $group stage (Sample of 13 documents)

```
/**
 * _id: The id of the group.
 * fieldN: The first field name.
 */
{
 _id:"$Age",
 Avg: {$avg: "$Age"},
 MinAge:{ $min:"$Age"},
 MaxAge:{$max:"$Age"},
 MaxSalary: {$max: "$Salary"},
 MinSalary:{$min:"$Salary"},
 AvgSalary:{$avg:"$Salary"}

 }
```

```
_id: 19
Avg: 19
MinAge: 19
MaxAge: 19
MaxSalary: 9846
MinSalary: 516
AvgSalary: 4330.909090909091

_id: 24
Avg: 24
MinAge: 24
MaxAge: 24
MaxSalary: 6921
MinSalary: 2078
AvgSalary: 4329

_id: 20
Avg: 20
MinAge: 20
MaxAge: 20
MaxSalary: 8539
MinSalary: 1786
AvgSalary: 5154

_id: 28
Avg: 28
MinAge: 28
MaxAge: 28
MaxSalary:
MinSalary:
AvgSalary:
```

3) find min,max average salary of each age group of male



Output after $match stage (Sample of 20 documents)

```
/**
 * query: The query in MQL.
 */
{
  Gender:"Male",
}
```

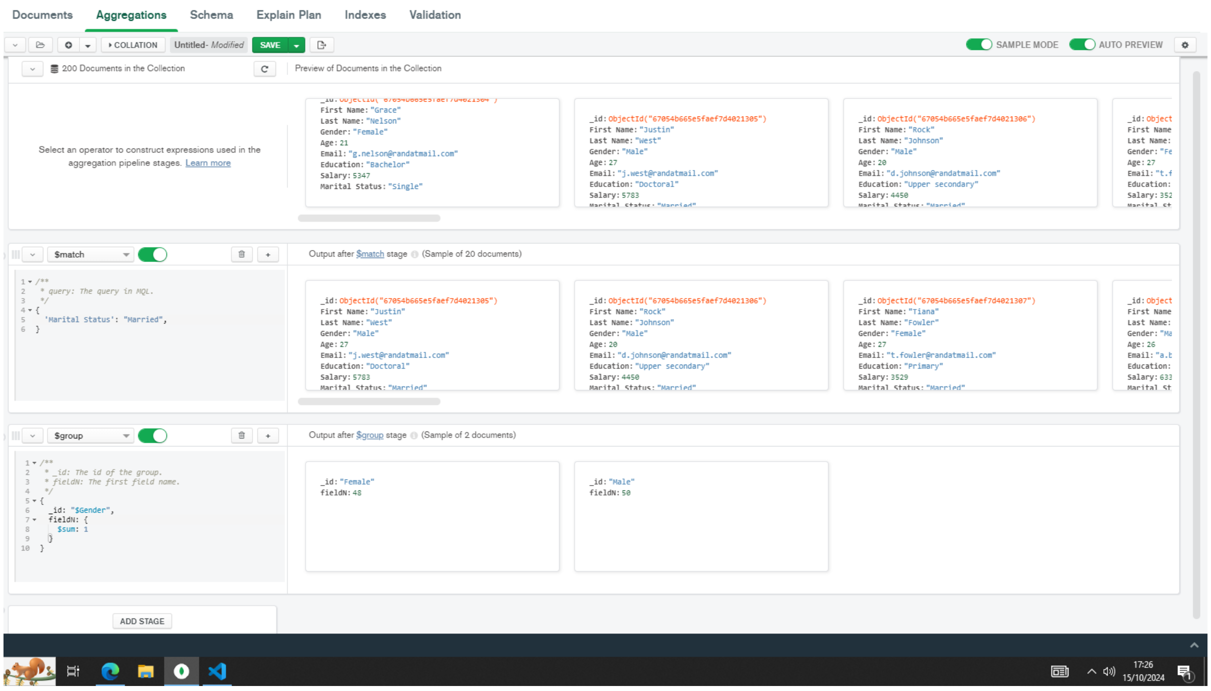
```
_id: ObjectId("67054b665e5faef7d4021305")
First Name: "Justin"
Last Name: "West"
Gender: "Male"
Age: 27
Email: "j.west@randatmail.com"
Education: "Doctoral"
Salary: 5783
Marital Status: "Married"

_id: ObjectId("67054b665e5faef7d4021306")
First Name: "Rock"
Last Name: "Johnson"
Gender: "Male"
Age: 20
Email: "d.johnson@randatmail.com"
Education: "Upper secondary"
Salary: 4450
Marital Status: "Married"

_id: ObjectId("67054b665e5faef7d4021308")
First Name: "Alen"
Last Name: "Barnes"
Gender: "Male"
Age: 26
Email: "a.barnes@randatmail.com"
Education: "Upper secondary"
Salary: 6332
Marital Status: "Married"
```

Output after $group stage (Sample of 13 documents)

```
/**
 * _id: The id of the group.
 * fieldN: The first field name.
 */
{
 _id:"$Age",
 Avg: {$avg: "$Age"},
 MinAge:{ $min:"$Age"},
 MaxAge:{$max:"$Age"},
 MaxSalary: {$max: "$Salary"},
 MinSalary:{$min:"$Salary"},
 AvgSalary:{$avg:"$Salary"}

 }
```

```
_id: 28
Avg: 28
MinAge: 28
MaxAge: 28
MaxSalary: 9989
MinSalary: 836
AvgSalary: 5907.875

_id: 24
Avg: 24
MinAge: 24
MaxAge: 24
MaxSalary: 8170
MinSalary: 2033
AvgSalary: 4412.4

_id: 20
Avg: 20
MinAge: 20
MaxAge: 20
MaxSalary: 9587
MinSalary: 1258
AvgSalary: 5309.333333333333
```

4) Count married and unmarried females and males.



200 Documents in the Collection

Preview of Documents in the Collection

Select an operator to construct expressions used in the aggregation pipeline stages. Learn more

```
_id: ObjectId("67054b665e5faef7d4021304")
First Name: "Grace"
Last Name: "Nelson"
Gender: "Female"
Age: 21
Email: "g.nelson@randatmail.com"
Education: "Bachelor"
Salary: 5347
Marital Status: "Single"

_id: ObjectId("67054b665e5faef7d4021305")
First Name: "Justin"
Last Name: "West"
Gender: "Male"
Age: 27
Email: "j.west@randatmail.com"
Education: "Doctoral"
Salary: 5783
Marital Status: "Married"

_id: ObjectId("67054b665e5faef7d4021306")
First Name: "Rock"
Last Name: "Johnson"
Gender: "Male"
Age: 20
Email: "d.johnson@randatmail.com"
Education: "Upper secondary"
Salary: 4450
Marital Status: "Married"

_id: Object
First Name
Last Name:
Gender: "Fe
Age: 27
Email: "t.f
Education:
Salary: 352
Marital St
```

Output after $match stage (Sample of 20 documents)

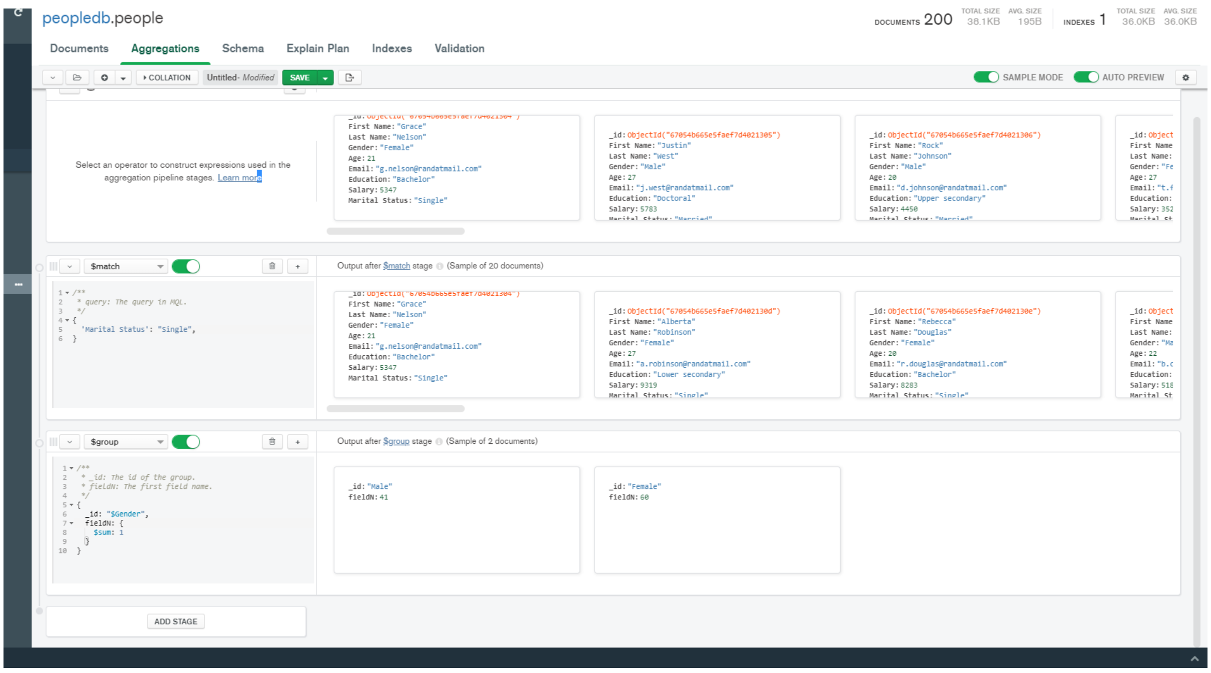
```
/**
 * query: The query in MQL.
 */
{
  'Marital Status': "Married",
}
```

```
_id: ObjectId("67054b665e5faef7d4021305")
First Name: "Justin"
Last Name: "West"
Gender: "Male"
Age: 27
Email: "j.west@randatmail.com"
Education: "Doctoral"
Salary: 5783
Marital Status: "Married"

_id: ObjectId("67054b665e5faef7d4021306")
First Name: "Rock"
Last Name: "Johnson"
Gender: "Male"
Age: 20
Email: "d.johnson@randatmail.com"
Education: "Upper secondary"
Salary: 4450
Marital Status: "Married"

_id: ObjectId("67054b665e5faef7d4021307")
First Name: "Tiana"
Last Name: "Fowler"
Gender: "Female"
Age: 27
Email: "t.fowler@randatmail.com"
Education: "Primary"
Salary: 3529
Marital Status: "Married"

_id: Object
First Name
Last Name:
Gender: "Ma
Age: 26
Email: "a.b
Education:
Salary: 633
Marital St
```

Output after $group stage (Sample of 2 documents)

```
/**
 * _id: The id of the group.
 * fieldN: The first field name.
 */
{
  _id: "$Gender",
  fieldN: {
    $sum: 1
  }
}
```

```
_id: "Female"
fieldN: 48

_id: "Male"
fieldN: 50
```



peopledb.people

DOCUMENTS 200 TOTAL SIZE 38.1KB AVG. SIZE 195B INDEXES 1 TOTAL SIZE 36.0KB AVG. SIZE 36.0KB

Select an operator to construct expressions used in the aggregation pipeline stages. Learn more

```
First Name: "Grace"
Last Name: "Nelson"
Gender: "Female"
Age: 21
Email: "g.nelson@randatmail.com"
Education: "Bachelor"
Salary: 5347
Marital Status: "Single"
```

```
_id: ObjectId("67054b665e5faef7d4021305")
First Name: "Justin"
Last Name: "West"
Gender: "Male"
Age: 27
Email: "j.west@randatmail.com"
Education: "Doctoral"
Salary: 5783
Marital Status: "Married"
```

```
_id: ObjectId("67054b665e5faef7d4021306")
First Name: "Rock"
Last Name: "Johnson"
Gender: "Male"
Age: 20
Email: "d.johnson@randatmail.com"
Education: "Upper secondary"
Salary: 4450
Marital Status: "Married"
```

$match

```
/**
 * query: The query in MQL.
 */
{
  'Marital Status': "Single",
}
```

Output after $match stage (Sample of 20 documents)

```
First Name: "Grace"
Last Name: "Nelson"
Gender: "Female"
Age: 21
Email: "g.nelson@randatmail.com"
Education: "Bachelor"
Salary: 5347
Marital Status: "Single"
```

```
_id: ObjectId("67054b665e5faef7d402130d")
First Name: "Alberta"
Last Name: "Robinson"
Gender: "Female"
Age: 27
Email: "a.robinson@randatmail.com"
Education: "Lower secondary"
Salary: 9319
Marital Status: "Single"
```

```
_id: ObjectId("67054b665e5faef7d402130e")
First Name: "Rebecca"
Last Name: "Douglas"
Gender: "Female"
Age: 20
Email: "r.douglas@randatmail.com"
Education: "Bachelor"
Salary: 8283
Marital Status: "Single"
```

$group

```
/**
 * _id: The id of the group.
 * fieldN: The first field name.
 */
{
  _id: "$Gender",
  fieldN: {
    $sum: 1
  }
}
```

Output after $group stage (Sample of 2 documents)

```
_id: "Male"
fieldN: 41
```

```
_id: "Female"
fieldN: 60
```

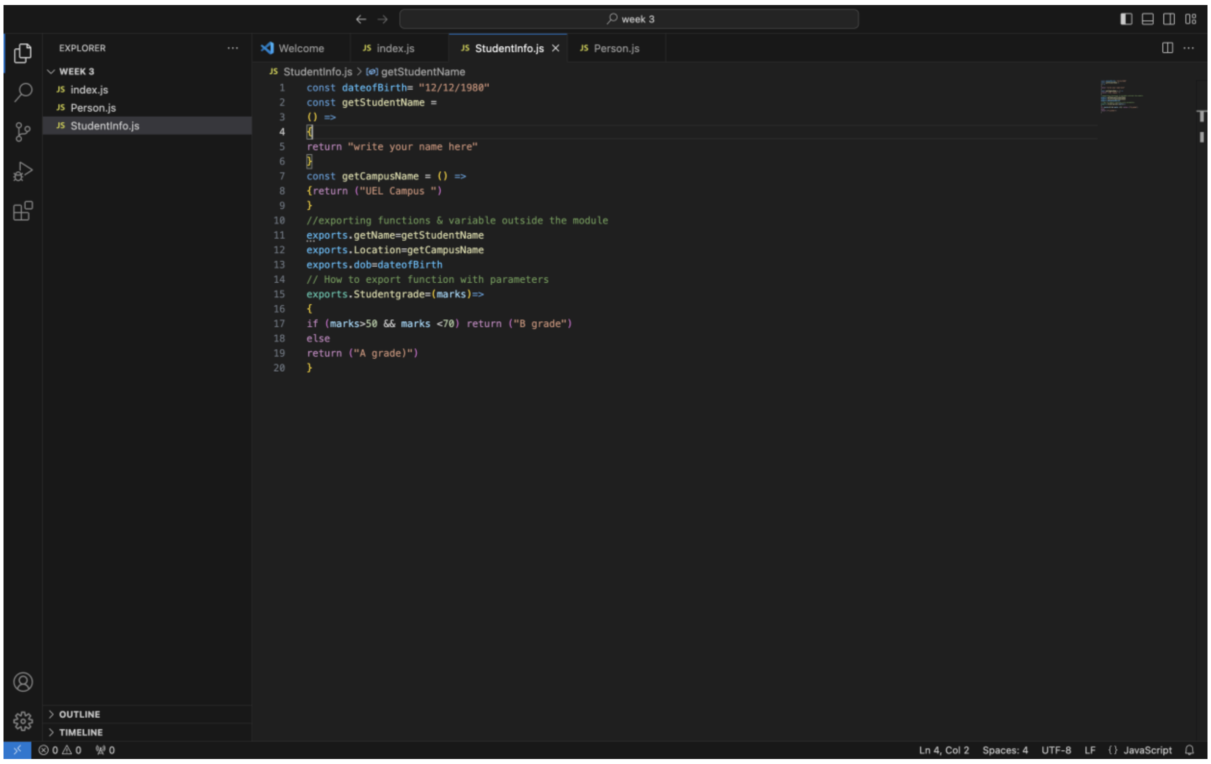
<u>**Report:**</u>

The work that been done with this code has been used to separate the data that has been given from a spread sheet then sorted into different categories depending on the need.

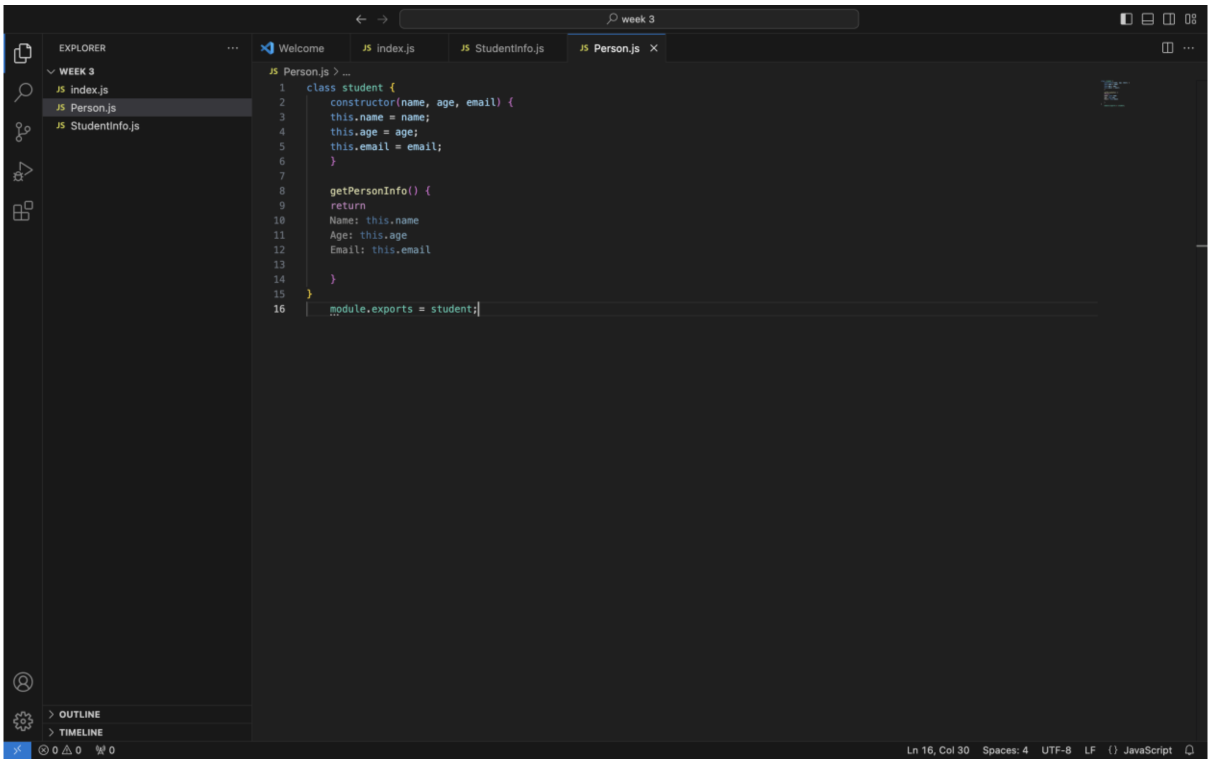
<u>**WEEK 3:**</u>

Submit the code for the completed Exercises i.e.: i) Index.js ii) Person.js iii) Employeeinfo.js iv) Exercise 4.js

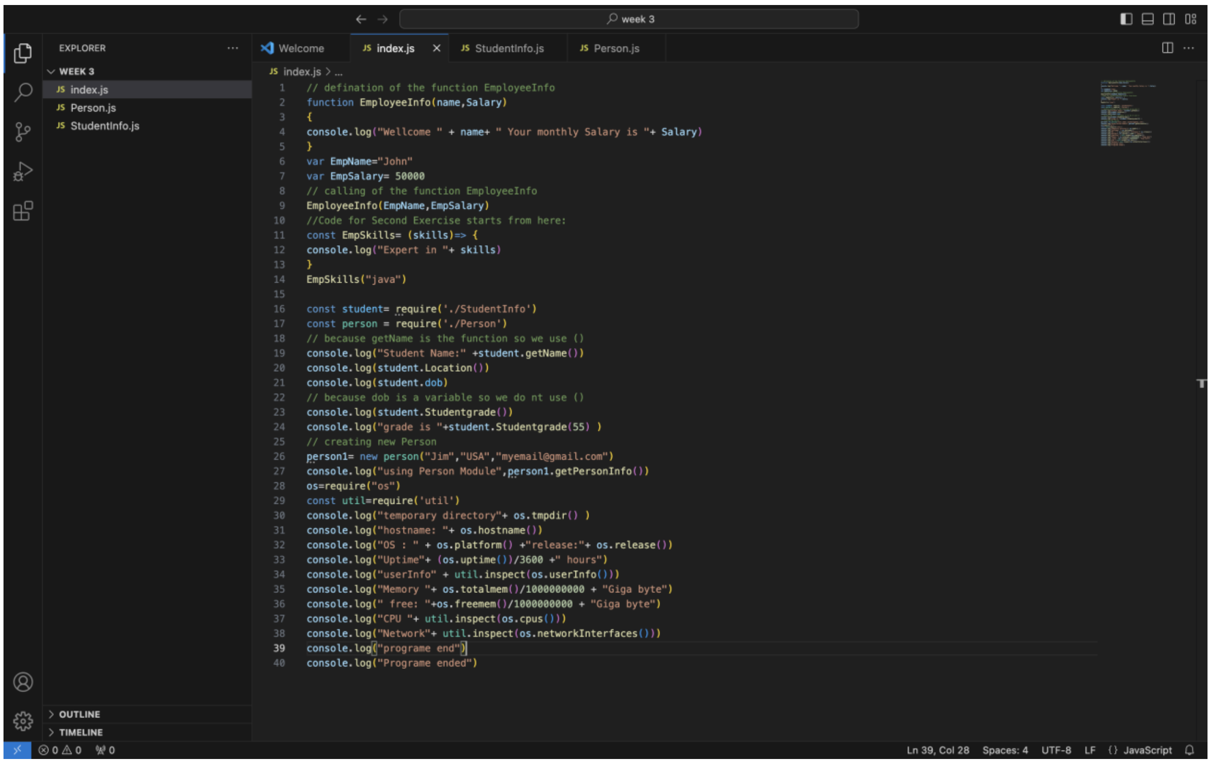
U2283556



```javascript
const dateofBirth= "12/12/1980"
const getStudentName =
() =>
{
return "write your name here"
}
const getCampusName = () =>
{return ("UEL Campus ")
}
//exporting functions & variable outside the module
exports.getName=getStudentName
exports.Location=getCampusName
exports.dob=dateofBirth
// How to export function with parameters
exports.Studentgrade=(marks)=>
{
if (marks>50 && marks <70) return ("B grade")
else
return ("A grade)")
}
```



```javascript
class student {
    constructor(name, age, email) {
    this.name = name;
    this.age = age;
    this.email = email;
    }

    getPersonInfo() {
    return
    Name: this.name
    Age: this.age
    Email: this.email

    }
}
    module.exports = student;
```

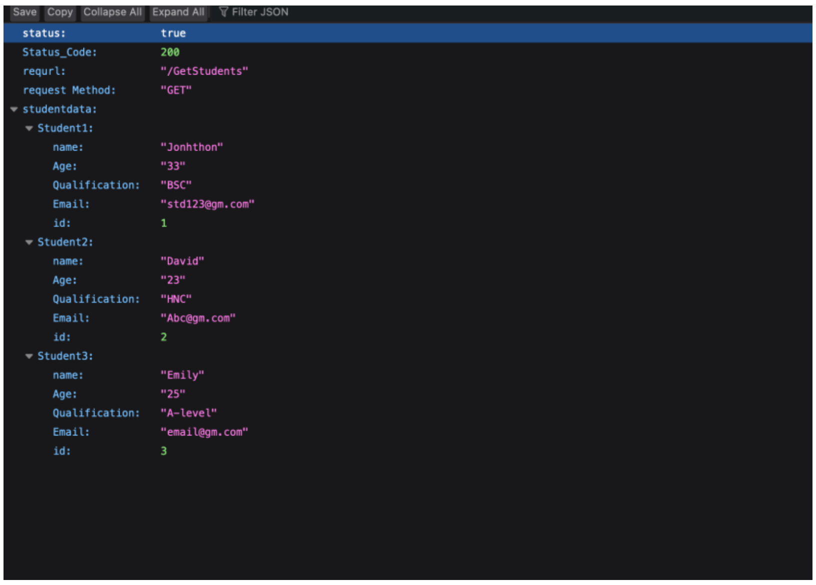


```javascript
// defination of the function EmployeeInfo
function EmployeeInfo(name,Salary)
{
console.log("Wellcome " + name+ " Your monthly Salary is "+ Salary)
}
var EmpName="John"
var EmpSalary= 50000
// calling of the function EmployeeInfo
EmployeeInfo(EmpName,EmpSalary)
//Code for Second Exercise starts from here:
const EmpSkills= (skills)=> {
console.log("Expert in "+ skills)
}
EmpSkills("java")

const student= require('./StudentInfo')
const person = require('./Person')
// because getName is the function so we use ()
console.log("Student Name:" +student.getName())
console.log(student.Location())
console.log(student.dob)
// because dob is a variable so we do nt use ()
console.log(student.Studentgrade())
console.log("grade is "+student.Studentgrade(55) )
// creating new Person
person1= new person("Jim","USA","myemail@gmail.com")
console.log("using Person Module",person1.getPersonInfo())
os=require("os")
const util=require('util')
console.log("temporary directory"+ os.tmpdir() )
console.log("hostname: "+ os.hostname())
console.log("OS : " + os.platform() +"release:"+ os.release())
console.log("Uptime"+ (os.uptime())/3600 +" hours")
console.log("userInfo" + util.inspect(os.userInfo()))
console.log("Memory "+ os.totalmem()/1000000000 + "Giga byte")
console.log(" free: "+os.freemem()/1000000000 + "Giga byte")
console.log("CPU "+ util.inspect(os.cpus()))
console.log("Network"+ util.inspect(os.networkInterfaces()))
console.log("programe end")
console.log("Programe ended")
```

<u>Report:</u>

The coding that I've done during this lesson displays the information of the device that is being used. It also displays the information of the student information that has been given. I learned how to link different methods together.

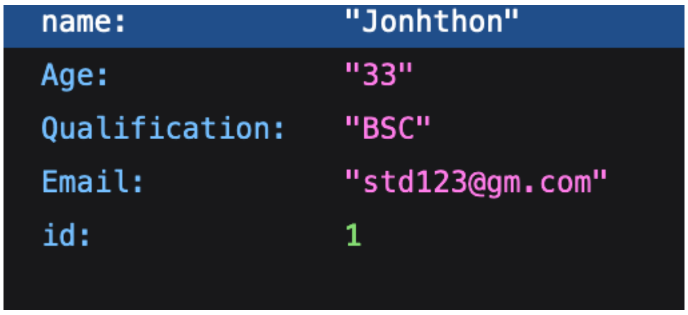
<u>WEEK 4:</u>



```
status:              true
Status_Code:         200
requrl:              "/GetStudents"
request Method:      "GET"
▼ studentdata:
  ▼ Student1:
      name:            "Jonhthon"
      Age:             "33"
      Qualification:   "BSC"
      Email:           "std123@gm.com"
      id:              1
  ▼ Student2:
      name:            "David"
      Age:             "23"
      Qualification:   "HNC"
      Email:           "Abc@gm.com"
      id:              2
  ▼ Student3:
      name:            "Emily"
      Age:             "25"
      Qualification:   "A-level"
      Email:           "email@gm.com"
      id:              3
```

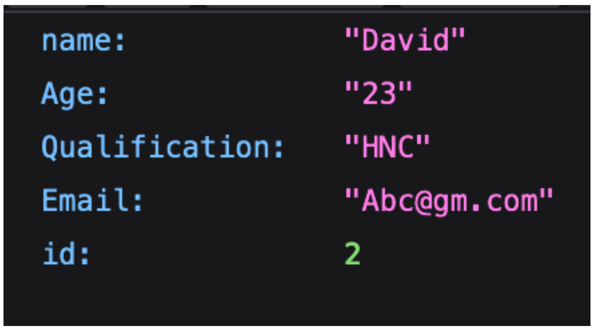
{"status":true,"Status_Code":200,"requrl":"/GetStudents","request Method":"GET","studentdata":{"Student1":{"name":"Jonhthon","Age":"33","Qualification":"BSC","Email":"std123@gm.com","id":1},"Student2":
{"name":"David","Age":"23","Qualification":"HNC","Email":"Abc@gm.com","id":2},"Student3":{"name":"Emily","Age":"25","Qualification":"A-level","Email":"email@gm.com","id":3}}}

Localhost:5000/GetStudents



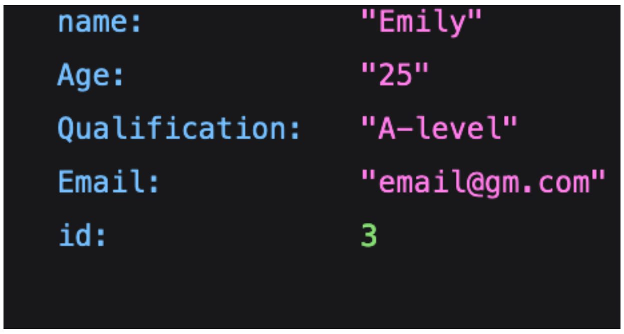
```
name:             "Jonhthon"
Age:              "33"
Qualification:    "BSC"
Email:            "std123@gm.com"
id:               1
```

Localhost:5000/ GetStudentid/1



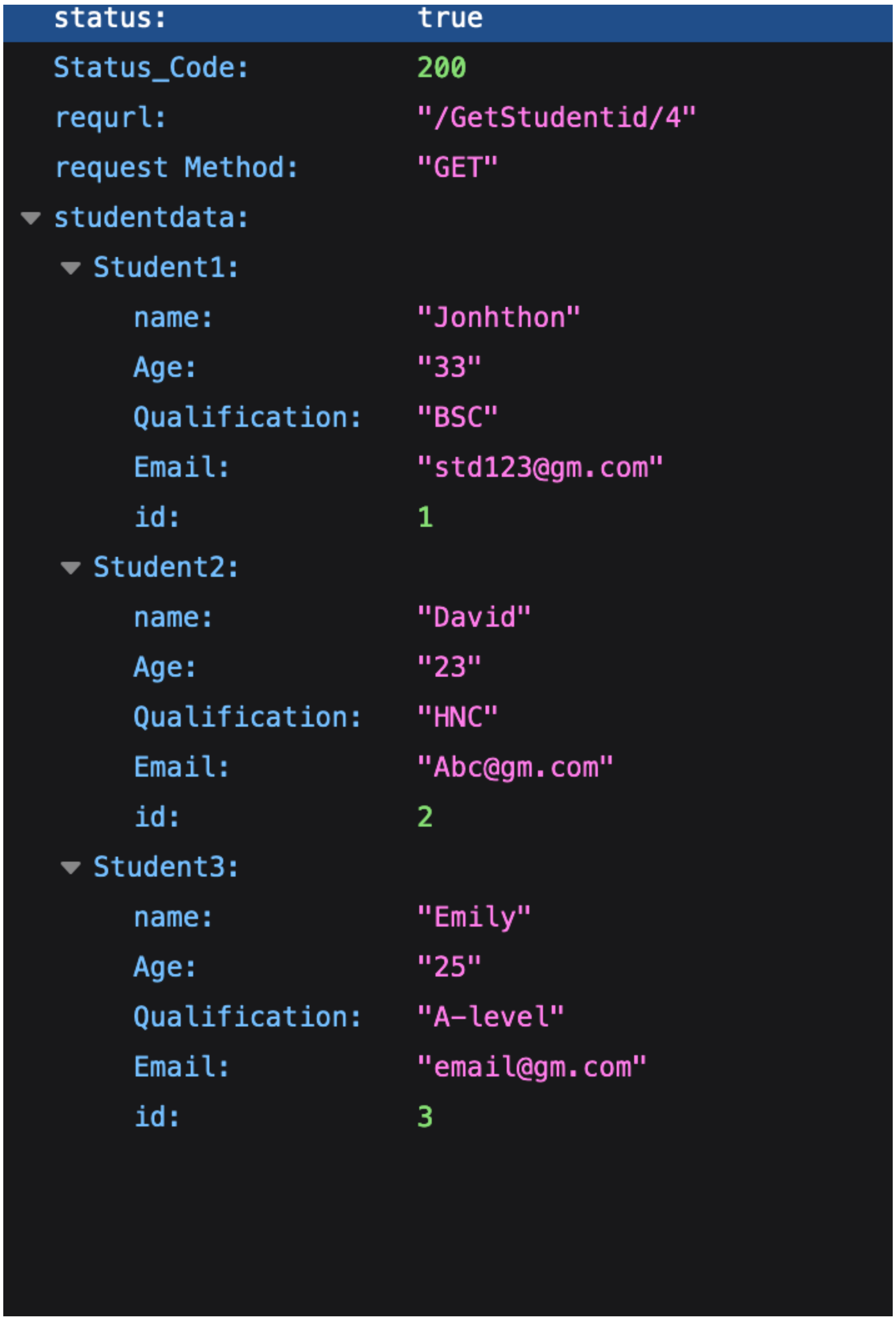
```
name:             "David"
Age:              "23"
Qualification:    "HNC"
Email:            "Abc@gm.com"
id:               2
```

Localhost:5000/ GetStudentid/2



```
name:             "Emily"
Age:              "25"
Qualification:    "A-level"
Email:            "email@gm.com"
id:               3
```

Localhost:5000/ GetStudentid/3

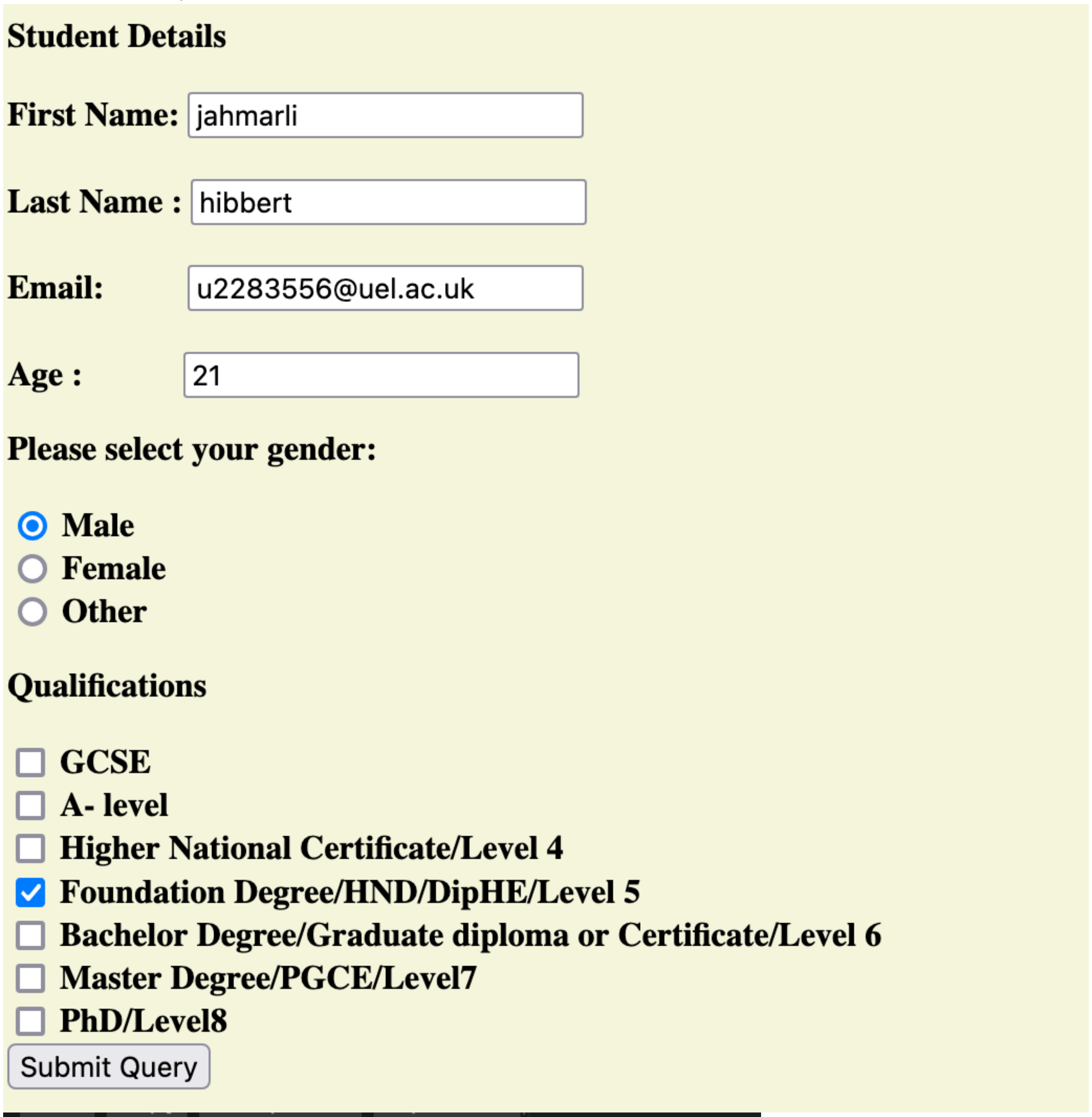


```
status:                 true
Status_Code:            200
requrl:                 "/GetStudentid/4"
request Method:         "GET"
▼ studentdata:
  ▼ Student1:
      name:             "Jonhthon"
      Age:              "33"
      Qualification:    "BSC"
      Email:            "std123@gm.com"
      id:               1
  ▼ Student2:
      name:             "David"
      Age:              "23"
      Qualification:    "HNC"
      Email:            "Abc@gm.com"
      id:               2
  ▼ Student3:
      name:             "Emily"
      Age:              "25"
      Qualification:    "A-level"
      Email:            "email@gm.com"
      id:               3
```

Localhost:5000/GetStudentid/4

U2283556

Localhost:5000/ studentinfo



Student Details

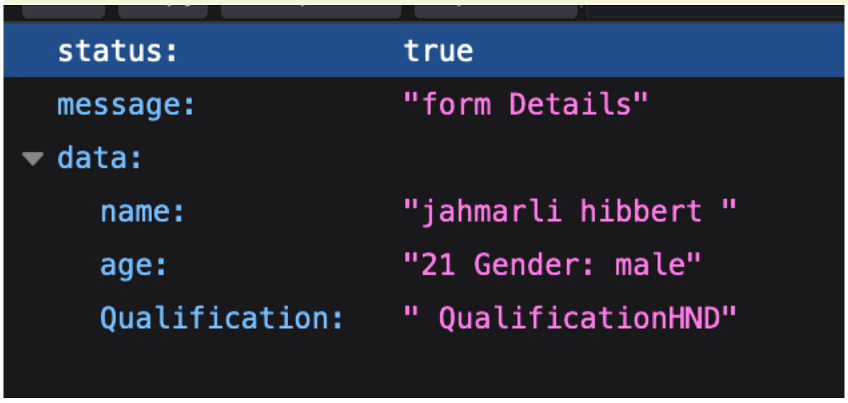
First Name:	jahmarli
Last Name :	hibbert
Email:	u2283556@uel.ac.uk
Age :	21

Please select your gender:

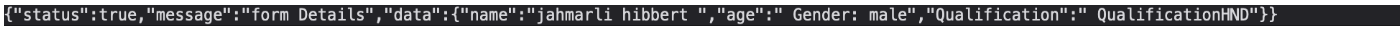
[✓ Male]
Female
Other

Qualifications

GCSE
A- level
Higher National Certificate/Level 4
[✓ Foundation Degree/HND/DipHE/Level 5]
Bachelor Degree/Graduate diploma or Certificate/Level 6
Master Degree/PGCE/Level7
PhD/Level8



```
status:          true
message:         "form Details"
▾ data:
    name:            "jahmarli hibbert "
    age:             "21 Gender: male"
    Qualification:   " QualificationHND"
```



{"status":true,"message":"form Details","data":{"name":"jahmarli hibbert ","age":" Gender: male","Qualification":" QualificationHND"}}

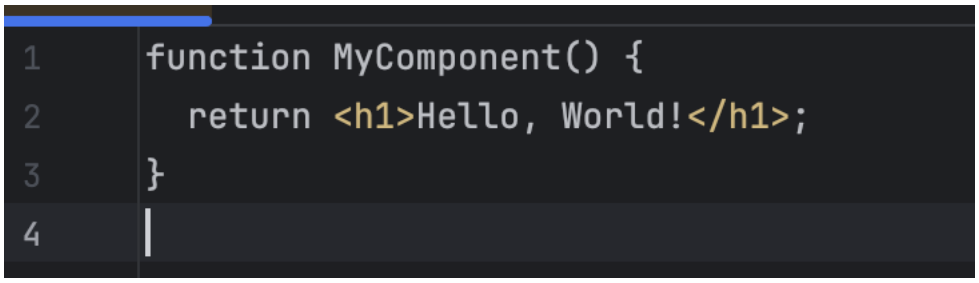
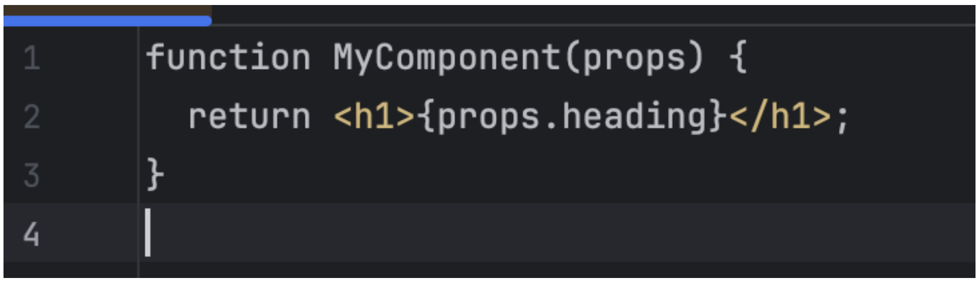
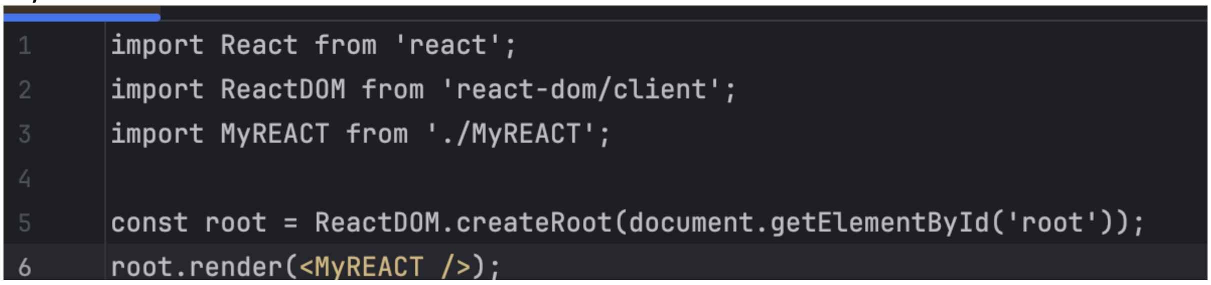
Report:

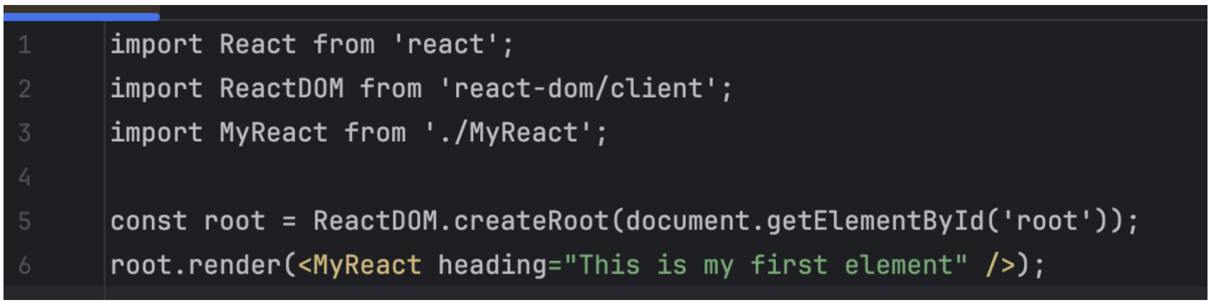
U2283556

This week I learnt how to implement data that has been provided and display it in a format that separates the data into different categories. The data has also been displayed in different formats.

WEEK 5:

For todays Lab submission, After you complete the lab write down a word document answering following questions for your portfolio:

1. What is React
   React is a JavaScript library for building a user interface.
2. What do you understand by React component and what command do you use to create a React component with or without property
   A React Component is a piece of UI logic that uses HTML, CSS, and JavaScript code to represent a part of the user interface.
   without
   
   ```
   function MyComponent() {
     return <h1>Hello, World!</h1>;
   }
   ```
   with
   
   ```
   function MyComponent(props) {
     return <h1>{props.heading}</h1>;
   }
   ```
3. What command will you use to render the the newly created component named as myREACT
   
   ```
   import React from 'react';
   import ReactDOM from 'react-dom/client';
   import MyREACT from './MyREACT';

   const root = ReactDOM.createRoot(document.getElementById('root'));
   root.render(<MyREACT />);
   ```
4. Suppose the MyReact Component has a property heading, write down the code that could be used to render the MYReact Component, and pass the message to the property heading as “this is my first element”



```
import React from 'react';
import ReactDOM from 'react-dom/client';
import MyReact from './MyReact';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<MyReact heading="This is my first element" />);
```

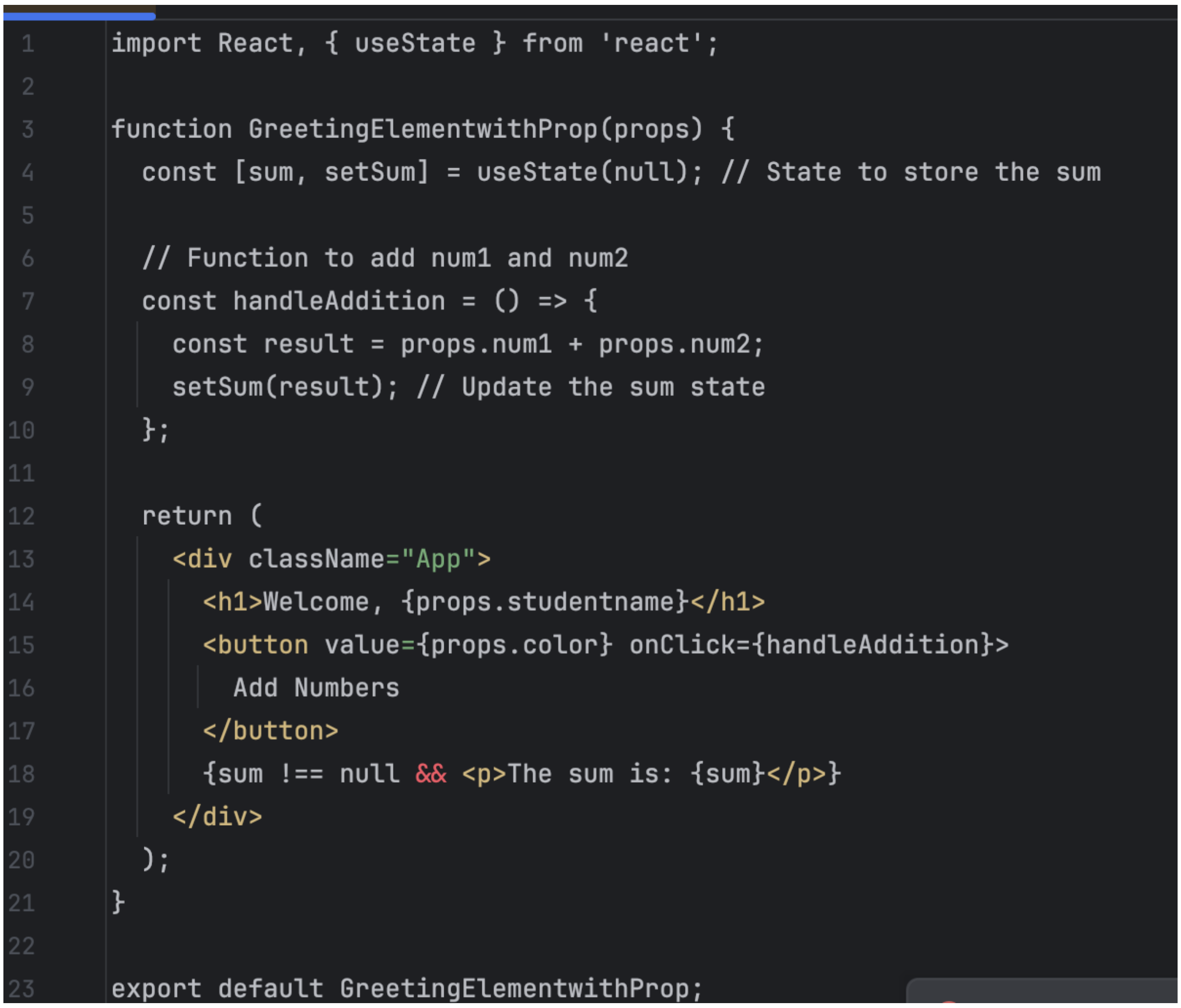
5. Observe this code and answer the questions below
   <AppColor heading="This is first element" lbl
   ="Name :" color="green"/>
   What is the name of the React Component
   AppColor
   How many properties this component uses
   heading, lbl, and color.
6. Look at the following Code:
   function GreetingElementwithProp(props) {

```
return (
<div className="App">
<h1>Wellcome , {props.studentname}</h1>;
</div>
);
}
export default ??????
```

what will you write to make this export this function correctly?
Hint you need to replace ?????? with the correct word.

Add a function that takes two properties as numbers ,add these numbers on the click event of the button and display the sum.

Hint you will be using in jsx
<button value={props.color} onClick={Namdofyourfunc
tion}



```
import React, { useState } from 'react';

function GreetingElementwithProp(props) {
  const [sum, setSum] = useState(null); // State to store the sum

  // Function to add num1 and num2
  const handleAddition = () => {
    const result = props.num1 + props.num2;
    setSum(result); // Update the sum state
  };

  return (
    <div className="App">
      <h1>Welcome, {props.studentname}</h1>
      <button value={props.color} onClick={handleAddition}>
        Add Numbers
      </button>
      {sum !== null && <p>The sum is: {sum}</p>}
    </div>
  );
}

export default GreetingElementwithProp;
```