Week 4 Portfolio

Name: Deepak Kumar
Student ID: 2284279

1. Screenshots

a. index.js

JS index.js > app.get('/users/:userId') callback

```
1  //import the express module
2  var express = require('express');
3  //import the fs module for file system I/O
4  var fs = require('fs');
5  //create the express app
6  var app = express();
7
8  var bodyParser = require('body-parser');
9  app.use(bodyParser.urlencoded({ extended: true }));
10
   Tabnine | Edit | Test | Explain | Document | Ask
11 app.get('/', function (req, res) {
12     res.send("hello it is my first express app");
13 });
14
   Tabnine | Edit | Test | Explain | Document | Ask
15 app.get('/about', function (req, res) {
16     res.send("this is the basic application");
17 })
18
   Tabnine | Edit | Test | Explain | Document | Ask
19 app.get('/users/:userId', function (req, res) { //app.get is
   a method in express
20     // /users/:userId is a route
21     // req is the request object
22     // res is the response object
23     res.send(req.params);
24 }); // to retrieve the data from the url
25
   Tabnine | Edit | Test | Explain | Document | Ask
26 app.get('/book/:bookId', function (req, res) {
27     res.send(req.params);
28 });
29
30
```

```
Tabnine | Edit | Test | Explain | Document | Ask
app.get('/GetStudents', function (req, res) {
    fs.readFile(__dirname + "/" + "student.json", 'utf8',
        function (err, data) {

            var students = JSON.parse(data);
            // Retrieve the specific student based on the
            studentId parameter
            var student = students["Student" + req.params.
            studentId];

            console.log("student: ", student);

            // Check if the student exists
            if (student) {
                res.json(student);
            } else {
                res.json({
                    'status': true, 'Status_Code': 200,
                    'requested at': req.localtime, 'requrl':
                    req.url,
                    'request Method': req.method,
                    'studentdata': JSON.parse(data)
                });

            }
        });
});

```

```
Tabnine | Edit | Test | Explain | Document | Ask
app.get('/studentinfo',function(req,res)
{
    res.sendFile('StudentInfo.html', { root: __dirname });
})

Tabnine | Edit | Test | Explain | Document | Ask
app.post('/submit-data', function (req, res) {
    var name = req.body.firstName + ' ' + req.body.lastName;
    var Age = req.body.myAge + ' Gender: ' + req.body.gender;
    var Qual = ' Qualification: ' + req.body.Qual;
    res.send({
        status: true,
        message: 'form Details',
        data: {
            name: name,
            age: Age,
            Qualification: Qual
        }
    });
});

Tabnine | Edit | Test | Explain | Document | Ask
app.listen(3000, function () {
    console.log("listening on port 3000");
});
```

b. Student.json

{} student.json > ...

```json
{
  "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": "Alevel",
    "Email": "email@gm.com",
    "id": 3
  }
}
```

c. StudentInfo.html

StudentInfo.html > html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <meta charset="utf-8" />
    <title>Student Enrollment Form</title>
</head>

<body bgcolor=beige>
    <form action="/submit-data" method="post">
        <p><Strong> Student Details </p>
        First Name: <input name="firstName" type="text" /> <br /><br />
        Last Name : <input name="lastName" type="text" /> <br /><br />
        Email:    <input type="email" id="emailid" name="email" /
        ><br /> <br />
        Age :     <input name="myAge" type="text" /> <br />
        <p>Please select your gender:</p>
        <input type="radio" id="male" name="gender" value="male">
        <label for="male">Male</label><br>
        <input type="radio" id="female" name="gender" value="female">
        <label for="female">Female</label><br>
        <input type="radio" id="other" name="gender" value="other">
        <label for="other">Other</label>
        <p> Qualifications </p>
        <input type="checkbox" name="Qual" value="GCSE"> GCSE <br />
        <input type="checkbox" name="Qual" value="A-level"> A- level <br>
        <input type="checkbox" name="Qual" value="Higher National
        Certificate">
        Higher National Certificate/Level 4 <br />
        <input type="checkbox" name="Qual" value="HND"> Foundation
        Degree/HND/DipHE/Level 5 <br />
        <input type="checkbox" name="Qual" value="B.Sc"> Bachelor Degree/
        Graduate
        diploma or
        Certificate/Level 6 <br />
        <input type="checkbox" name="Qual" value="Master Degree"> Master
        Degree/PGCE/Level7 <br />
        <input type="checkbox" name="Qual" value="PhD"> PhD/Level8 <br />
        <input type="submit" />
    </form>
</body>

</html>
```

d. Form

2. Reflection

In this lab session, I learned how to use Node.js and Express to create REST APIs. Firstly, I created a folder, initialized a `package.json` file, and installed Express using terminal commands. After that, I created a basic Express application with a simple endpoint (`/`) and ran it successfully in the browser. From there, I added new routes, like `/about` and `/users/:userId/books/:bookId`, and learned how to handle dynamic URL parameters using `req.params`.

Next, I learned and worked with JSON data by creating a `Student.json` file. I wrote code to read this file and display its data on a browser. Finally, I created a form using HTML, processed its input with a POST request, and sent the data back as JSON. This showed me how APIs handle user input effectively.

Overall, this lab gave me practical experience in setting up and using Express for REST APIs in backend development. I'm looking forward to front end topics such as React in the upcoming week.