Rajeshwari Reddy Gummadi

U2276440

Reflection of Lab Exercise week 4

**Web and Mobile Application Development**

This week we learnt about REST API. To create an application to display the function of REST API using node js. URLs are used to identify resources, such as users, products, etc.  
These resources are operated on using the HTTP methods GET, POST, PUT, and DELETE.   
Requests are answered by the server with a representation of the resource, usually in XML or JSON forms.

These are some of the HTTP methods are available to the client when submitting a request:

* GET: A read operation that retrieves data from the server.
* POST: Usually used to create new resources (Create operation), this protocol sends data to the server.
* PUT: An update operation that is used to update the server's current data.
* DELETE: This action deletes data from the server.
* It provides two additional functions, use and ALL, in addition to these.

The body-parser is required for processing incoming data in a number of formats, including JSON, URL-encoded form data, and text data. Body-parser is a middleware that is used to parse the incoming request.

Express is a web framework that Node.js provides. It has some features that can be helpful when developing RESTful APIs.

For creating express web application in node js we need to open Visual Studio code.

In the terminal window write npm install express - -save command to install Express. I also created Express directory where package.json file was created. I created a new file called index.js

**Exercise 1: Write the code for the basic express hello application . NodeJS Core module used: “express”**

Index.js file

A screenshot of a computer program

Description automatically generated

When the code is saved in the file, write node index.js in terminal window along side the directory and execute the program. Here you see the console message saying server is running on port 3000. Next go to the browser and type localhost:3000 in the search engine. Localhost is the IP address we have created and port number as 3000.

This is the text message displayed on the web page.

A screenshot of a computer

Description automatically generated

Next step is to go to terminal window and press Ctrl C to terminate the running program.

**Exercise 2**

These are the three URLs used:

1. Localhost:3000

2. Localhost:3000/about

3. Localhost:3000/users/33/books/123

These three URLs show different web pages according to their resources sush as users and products.



A screenshot of a computer

Description automatically generated

This is the first URL I browsed. This was the message showed

A black background with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

This is the second URL I browsed. This was the message showed

A black background with white text

Description automatically generated

A screenshot of a computer

Description automatically generated

This is the third URL I browsed. This was the message showed, user id and book id.

A screenshot of a computer

Description automatically generated

Here I tried using different user id and book id.

**Exercise 3**

Created student.json file and I added the code.

A screen shot of a computer program

Description automatically generated

The code related to json file in index.js. Here dir name is used, that returns the path of the folder where the current js file exists .

Index.js file

A screen shot of a computer code

Description automatically generated

Save all the files and in the terminal window write node index .js and go to the browser to check.



This is what is displayed in the window when I run the URL localhost:3000/GetStudents. It is the information related to the student.

A screen shot of a computer screen

Description automatically generated

Localhost:5000/ GetStudentid/1 if I put this URL I see the information related to the student ID 1

A screenshot of a computer

Description automatically generated

Localhost:5000/ GetStudentid/2 when this URL is put I can see in formation of the student ID 2

A screenshot of a computer

Description automatically generated

Localhost:5000/ GetStudentid/3 here information related to the student ID 3 is seen on the web page.

A screenshot of a computer

Description automatically generated

Localhost:5000/ GetStudentid/4 here is the information that will be related to student ID 4.

A screen shot of a computer code

Description automatically generated

**Exercise 4**

Now the POST method is used.

One more file is created named as Studentinfo.html. Form is created using HTML code. This is enrolment form .

A screen shot of a computer program

Description automatically generatedA computer screen shot of text

Description automatically generated

add Studentinfo.html in index.js file.

A computer screen shot of code

Description automatically generated

This is how the form looks like. Now I fill in the details and click the submit button.

A screenshot of a computer

Description automatically generated

I observed the input that I have given is displayed on the web page.

A screenshot of a computer program

Description automatically generated