KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTION)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH,
Narayanguda, Hyderabad – 500029

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL

WEB TECHNOLOGIES

LAB

B.Tech. III YEAR I SEM (KR20)

INDEX

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Narayanguda, Hyderabad – 500029

Department of Computer Science & Engineering

Vision of the Institution:

To be the fountain head of latest technologies, producing highly skilled, globally competent engineers.

Mission of the Institution:

- To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modal platforms and prepare students to become successful professionals.
- To establish Industry Institute Interaction to make students ready for the industry.
- To provide exposure to students on latest hardware and software tools.
- To promote research based projects/activities in the emerging areas of technology convergence.
- To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises
- To induce a spirit of nationalism which will enable the student to develop, understand India's challenges and to encourage them to develop effective solutions.
- To support the faculty to accelerate their learning curve to deliver excellent service to students

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Narayanguda, Hyderabad – 500029

Department of Computer Science & Engineering

Vision of the Department:

To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

Mission of the Department:

- To provide faculty with state of the art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
- To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
- To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
- To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
- To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefits the society-at-large.
- To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities.

Department of Computer Science & Engineering

PROGRAM OUTCOMES (POs)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6: The Engineer and Society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and Sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Narayanguda, Hyderabad – 500029

Department of Computer Science & Engineering

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: An ability to analyze the common business functions to design and develop appropriate Computer Science solutions for social upliftment.

PSO2: Shall have expertise on the evolving technologies like Python, Machine Learning, Deep Learning, Internet of Things (IOT), Data Science, Full stack development, Social Networks, Cyber Security, Big Data, Mobile Apps, CRM, ERP etc.

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY
(AN AUTONOMOUS INSTITUTE)
Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH,
Narayanguda, Hyderabad – 500029

Department of Computer Science & Engineering

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.

PEO2: Graduates will try to provide solutions to challenging problems in their profession by applying computer engineering principles.

PEO3: Graduates will engage in life-long learning and professional development by rapidly adapting to the changing work environment.

PEO4: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility.

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Narayanguda, Hyderabad – 500029

B. Tech. in COMPUTER SCIENCE AND ENGINEERING
III Year I Semester Course Syllabus (KR20)

L	T	P	C
0	0	4	2

WEB TECHNOLOGIES LAB (CS506PC)

Prerequisites/ Corequisites:

1. CS401PC – Java Programming Course
2. CS402PC- Database Management Systems Course
3. CS503PC- Web Technologies Course

Course Objectives: The course will help to

1. Comprehend and apply the concepts of JavaScript
2. Gain knowledge and to build a web page using HTML and Bootstrap.
3. Implement the server-side functionality using NodeJS and Express JS.
4. Comprehend and demonstrate the application of Express routing mechanism.
5. Practice and demonstrate the working of NoSQL databases and to implement the integration of front end with backend database (MongoDB) using Express JS.

Course Outcomes: After learning the concepts of this course, the student is able to

1. Apply asynchronous programming techniques using java script.
2. Apply HTML and JavaScript effectively to create an interactive and responsive website.
3. Develop a simple client server model using NodeJS and Express JS.
4. Analyze and Implement routing methodologies using Express JS.
5. Design and create a NoSQL (Mongo)database and then integrate it with the front end using Express JS, this, by better understanding client-server communication.

List of Experiments:

1. Write a JavaScript program --
   a) which accepts a string as input and swaps the case of each character. For example if you input 'The Quick Brown Fox' the output should be 'tHEqUICKbROWNfOX'.
   b) to find the most frequent item of an array
   c) to remove duplicate items from an array (ignore case sensitivity)
   d) to perform a binary search
   e) to list the properties of a JavaScript object.
   f) to check whether an object contains given property.
   g) to sort a list of elements using Quick sort.
   h) to implement Bubble Sort
   i) to read from a JSON object and display the data in a table (HTML page)
   j) that takes name, rollno, marks from user from a form and on form submission displays them in a tabular format, with GPA (like a marks sheet)
2. Write JS code in an HTML page such that based on location selected by user an AJAX request is made and weather details for that location are fetched and displayed
3. Write a Node JS program that accepts a port from the user and runs a node server at that port
4. Write a NodeJS program to read from a file and display the content on screen
5. Write a NodeJS program to accept a file name from user, text from user, if file exists append the text to the file . If not, create a new file and add the text to it.
6. Create a student database in MongoDB with all the details of students of a class.
7. Create a form such that , based on student roll number provided by user, the student details should be fetched (using ExpressJS)

Additional Programs-

8. Create a form such that CRUD operations can be performed on the student DB using ExpressJS
9. Create a simple website for the CRUD operations on student DB and apply Express Routing.

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

(AN AUTONOMOUS INSTITUTE)

Accredited by NBA & NAAC, Approved by AICTE, Affiliated to JNTUH, Narayanguda, Hyderabad – 500029

Department of Computer Science & Engineering

Course Outcomes and CO-PO-PSO Mapping

Course Outcomes:

After learning the contents of this course, the student is able to

CO1	Apply asynchronous programming techniques using java script.
CO2	Apply HTML and JavaScript effectively to create an interactive and responsive website.
CO3	Develop a simple client server model using NodeJS and Express JS.
CO4	Analyze and Implement routing methodologies using Express JS.
CO5	Design and create a NoSQL (Mongo)database and then integrate it with the front end using Express JS, this, by better understanding client-server communication

CO-PO-PSO MAPPING:

		PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12
Web Technologies Lab	CO1	2	2	2	3								
	CO2	2	2	3	3								
	CO3	2	3	3	3								
	CO4	2	2	3	2								
	CO5	2	2	3	3								

Software Requirements

Software Required:

1. **visual studio code** (Editor for writing code)

2. **nodejs**(Node.js is an open source server environment, to run JavaScript on the server)

3. **mongodb** (MongoDB is an open source NoSQL database management program.)

Installation process:

1. Visual studio code https://code.visualstudio.com/download
   a. Download software according to your system/laptop specifications.
   b. Follow the instructions accordingly and go on to install software.

2. Node.js https://nodejs.org/en/download/
   a. Download software according to your system/laptop specifications.
   b. Follow the instructions accordingly and go on to install software.

3. MongoDB https://www.mongodb.com/try/download/community
   a. Download community software according to your system/laptop specifications.
   b. Follow the instructions accordingly and go on to install software.

EXPERIMENT 1

1.a) Write a JavaScript program which accepts a string as input and swap the case of each character. For example if you input 'The Quick Brown Fox' the output should be 'tHEqUICKbROWNfOX'. – "one.js"

```
const readline = require('readline');
var RL = readline.createInterface(process.stdin, process.stdout);
RL.question('Please Enter Text:  ', (name)=>{
  let x=name;
  let y="";
for(let i=0;i<x.length;i++)
{
if (x.charAt(i) >='A' && x.charAt(i) <= 'Z')
y=y+x.charAt(i).toLowerCase();
else if(x.charAt(i) >='a' && x.charAt(i) <= 'z')
y=y+x.charAt(i).toUpperCase();
}

  console.log(`Output is is ${y}`);

});
```

EXPECTED OUTPUT:

Sample input: The Quick Brown Fox
Sample Output: tHEqUICKbROWNfOX

b) **. Write a JavaScript program to find the most frequent item of an array. – "two.js"**

```
var arr1=[3, 'a', 'a', 'a', 2, 3, 'a', 3, 'a', 2, 4, 9, 3];
var mf = 1;
var m = 0;
var item;
for (var i=0; i<arr1.length-1; i++)
{
	for (var j=i; j<arr1.length; j++)
	{
		if (arr1[i] == arr1[j])
		 m++;
		if (mf<m)
		{
		  mf=m;
		  item = arr1[i];
		}
	}
	m=0;
}
console.log(item+" ( " +mf +" times ) ") ;
```

EXPECTED OUTPUT
a (5 times)

if the input contains same number of occurrences for two entries then it prints only least significant one

c). Write a JavaScript program to remove duplicate items from an array – 'three.js'

```
function removeDuplicates(num) {
      len=num.length;
      uniqueChars=[];

      num.forEach((c) => {
         if (!uniqueChars.includes(c)) {
            uniqueChars.push(c);
         }
      });
   return uniqueChars;
 }
 let Mynum = [1, 2, 2, 4, 5, 4, 7, 8, 7, 3, 6];
 result = removeDuplicates(Mynum);
 console.log("Original List:   "+Mynum);
 console.log("Unique List:    "+result);
```

EXPECTED OUTPUT:

d) Write a JavaScript program to perform a binary search. – "four.js"

```
let iterativeFunction = function (arr, x) {
    let start=0, end=arr.length-1;
    while (start<=end){
        let mid=Math.floor((start + end)/2);
        if (arr[mid]===x) return true;
        else if (arr[mid] < x)
            start = mid + 1;
        else
            end = mid - 1;
    }

    return false;
}


let arr = [1, 3, 5, 7, 8, 9];
let x = 5;
console.log(iterativeFunction(arr, x) )   ;
```

EXPECTED OUTPUT:

-> if x : 5 output: true

-> if x: 6 output: false

e) **Write a JavaScript program to list the properties of a JavaScript object – “five.js”**

```
let object = {
    name: 'Jack',
    age: 25,
    college: 'KMIT',
    year: 3,
    sem: 1
  };
let properties = Object.keys(object)
console.log(properties);
```

EXPECTED OUTPUT:

f) Write a JavaScript function to check whether an object contains given property. – "six.js"

***1. hasOwnProperty()* method**

```
let object = {
    name: 'Jack',
    age: 25,
    college: 'KMIT',
    year: 3,
    sem: 1
  };
console.log(object.hasOwnProperty('name'));
```

2. in operator Method:

```
let object = {
    name: 'Jack',
    age: 25,
    college: 'KMIT',    year: 3,    sem: 1
  };
console.log('name' in object);
```

3. Comparing with undefined Method:

```
let object = {
    name: 'Jack',
    age: 25,
    college: 'KMIT',    year: 3,    sem: 1
  };
console.log(object.name);
console.log(object.fee);
```

Expected output: here Name property is available so programs gives you output as 'Jack' but fee property is not available so it is giving output as undefined.

g) Write a JavaScript program to sort a list of elements using Quick sort. –"seven.js"

```
function quick_Sort(origArray) {
    if (origArray.length <= 1) {
            return origArray;
    } else {

            var left = [];
            var right = [];
            var newArray = [];
            var pivot = origArray.pop();
            var length = origArray.length;

            for (var i = 0; i < length; i++) {
                    if (origArray[i] <= pivot) {
                            left.push(origArray[i]);
                    } else {
                            right.push(origArray[i]);
                    }
            }

            return newArray.concat(quick_Sort(left), pivot, quick_Sort(right));
    }
}

var myArray = [3, 0, 2, 5, -1, 4, 1 ];

console.log("Original array: " + myArray);
var sortedArray = quick_Sort(myArray);
console.log("Sorted array: " + sortedArray);
```

Expected Output:

h) Write a JavaScript program to implement Bubble Sort. –“eight.js”

```
function swap(arr, first_Index, second_Index){
   var temp = arr[first_Index];
   arr[first_Index] = arr[second_Index];
   arr[second_Index] = temp;
}

function bubble_Sort(arr){

   var len = arr.length,
      i, j, stop;

   for (i=0; i < len; i++){
      for (j=0, stop=len-i; j < stop; j++){
         if (arr[j] > arr[j+1]){
            swap(arr, j, j+1);
         }
      }
   }

   return arr;
}
myArray=[3, 0, 2, 5, -1, 4, 1];
console.log("Original array: " + myArray);
var sortedArray = bubble_Sort(myArray);
console.log("Sorted array: " + sortedArray);
```

Expected Output:

i) Write a JS program to read from a JSON object and display the data in a table (HTML page)

s1.json

```
{ "student":[
  {"name": "Bhavana", "age": 20,  "college": "KMIT", "year": 3, "sem": 1 },
  { "name": "Ram", "age": 21, "college": "JNTU", "year": 4,"sem": 2 },
  { "name": "John", "age": 26, "college": "KMEC", "year": 1,"sem": 1 },
  { "name": "Reena", "age": 19, "college": "NGIT", "year": 3,"sem": 1 }
]
}
```

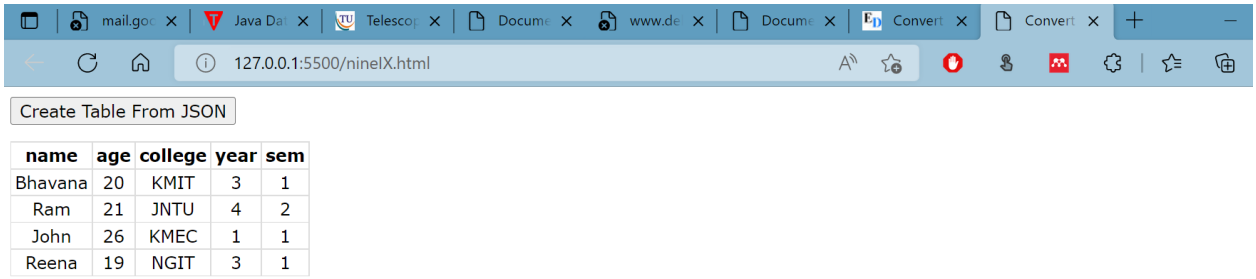
nineIX.html

```
<!DOCTYPE html>
<html>
<head>
   <title>Convert JSON Data to HTML Table</title>
   <style>
      th, td, p, input {
         font:14px Verdana;
      }
      table, th, td
      {
         border: solid 2px #DDD;
         border-collapse: collapse;
         padding: 2px 3px;
         text-align: center;
      }
      th {
```

```
        font-weight:bold;
    }
  </style>
</head>
<body>
  <input type="button" onclick="CreateTableFromJSON()" value="Create Table From JSON"
/>
  <p id="showData"></p>
</body>

<script>
  function CreateTableFromJSON() {
    fetch("s1.json")
    .then (response => response.json())
    .then(data => {
    // EXTRACT VALUE FOR HTML HEADER.
    // ('Name', 'Age', 'College', 'Year',Sem)
    var col = [];
    for (var i = 0; i < data.student.length; i++) {
      for (var key in data.student[i]) {
        if (col.indexOf(key) === -1) {
          col.push(key);
        }
      }
    }
      console.log(col);
    // CREATE DYNAMIC TABLE.
    var table = document.createElement("table");

    // CREATE HTML TABLE HEADER ROW USING THE EXTRACTED HEADERS
ABOVE.

    var tr = table.insertRow(-1);                   // TABLE ROW.
```

```
        for (var i = 0; i < col.length; i++) {
            var th = document.createElement("th");      // TABLE HEADER.
            th.innerHTML = col[i];
            tr.appendChild(th);
        }

        // ADD JSON DATA TO THE TABLE AS ROWS.
        for (var i = 0; i < data.student.length; i++) {

            tr = table.insertRow(-1);

            for (var j = 0; j < col.length; j++) {
                var tabCell = tr.insertCell(-1);
                tabCell.innerHTML = data.student[i][col[j]];
            }
        }

        // FINALLY ADD THE NEWLY CREATED TABLE WITH JSON DATA TO A
CONTAINER.
        var divContainer = document.getElementById("showData");
        divContainer.innerHTML = "";
        divContainer.appendChild(table);
    })
    }
</script>
</html>
```

EXPECTED OUTPUT:



name	age	college	year	sem
Bhavana	20	KMIT	3	1
Ram	21	JNTU	4	2
John	26	KMEC	1	1
Reena	19	NGIT	3	1

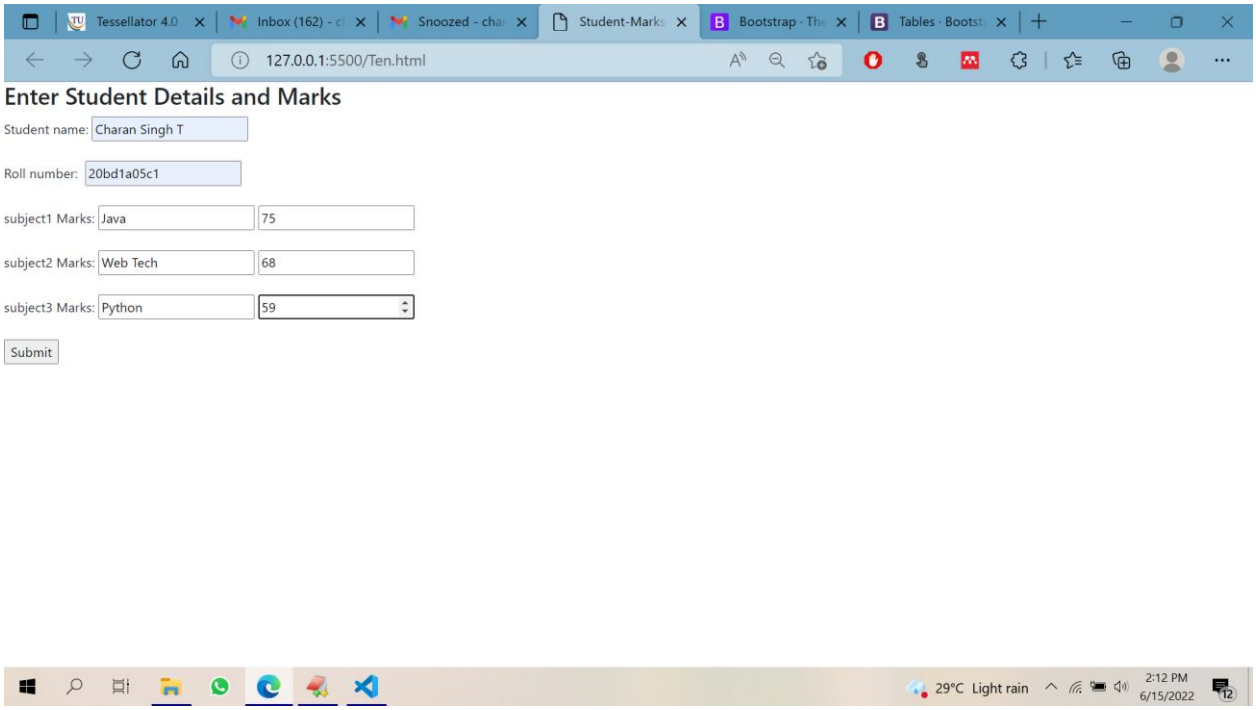
j) **Create a JS application that accepts the student's roll number, name, and marks and, when the form has been submitted, displays the student's name, roll number, and marks in a tabular format along with their GPA (like a marks sheet) – "ten.html"**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Student-Marks-Sheet</title>
    <!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFBL6DOitfPri4tjfHxaWutUpFmBp4vmVor" crossorigin="anonymous">

    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>

    <style>
      .sdetails{

      }
    </style>
</head>
<body>
  <div id="mydata">
```

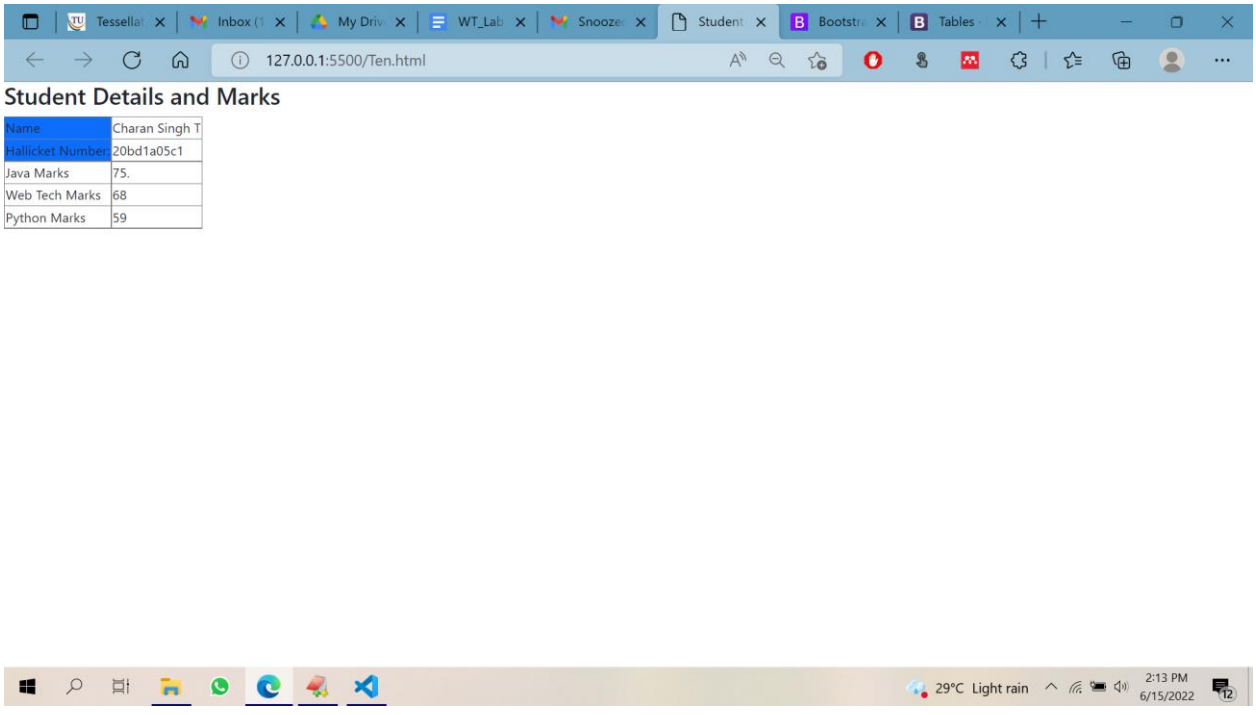
```
</div>

<div id="myformdiv">
 <h3> Enter Student Details and Marks</h3>
 <form id="myForm" method="post" >
    <label>Student name:</label>
     <input type="text" id="sname" ><br> <br>
     <label>Roll number:  </label>
     <input type="text" id="rollno"><br><br>
     <label>subject1 Marks:</label>
     <input type="text" id="sub1name" placeholder="sub1">
     <input type="number" id="marks1"><br><br>
     <label>subject2 Marks:</label>
     <input type="text" id="sub2name" placeholder="sub2" >
     <input type="number" id="marks2"><br><br>
     <label>subject3 Marks:</label>
     <input type="text" id="sub3name" placeholder="sub3">
     <input type="number" id="marks3"><br><br>
    <input type="button" onclick="myFunction()" value="Submit">
  </form>
 </div>
  <script>
    function myFunction() {
      document.getElementById('myformdiv').style.display='none';
      let myInfo = `
      <div class='sdetails'>
      <h3> Student Details and Marks</h3>
       <table class=table-bordered table-primary>
        <tr>
          <td class="bg-primary"> Name</td> <td>${
document.getElementById('sname').value}</td>
          </tr>
          <tr> <td class="bg-primary"> Hallicket Number:  </td> <td>${
```

```
document.getElementById('rollno').value}</td>
            </tr>
            <tr>
               <td > ${ document.getElementById('sub1name').value} Marks </td><td>${
document.getElementById('marks1').value}.</td>
            </tr>
            <tr> <td>  ${ document.getElementById('sub2name').value} Marks </td><td> ${
document.getElementById('marks2').value}</td>
            </tr>
            <tr> <td>  ${ document.getElementById('sub3name').value} Marks </td><td> ${
document.getElementById('marks3').value}</td>
            </tr>
            </tr>
          </table>
          </div>

         `;
         document.getElementById('mydata').innerHTML = myInfo;

       }
       </script>

</body>
</html>
```

EXPECTED OUTPUT


127.0.0.1:5500/Ten.html
Enter Student Details and Marks
Student name: Charan Singh T
Roll number: 20bd1a05c1
subject1 Marks: Java 75
subject2 Marks: Web Tech 68
subject3 Marks: Python 59
Submit
29°C Light rain
2:12 PM
6/15/2022

on submit


127.0.0.1:5500/Ten.html
Student Details and Marks
Name Charan Singh T
Hallicket Number 20bd1a05c1
Java Marks 75.
Web Tech Marks 68
Python Marks 59
29°C Light rain
2:13 PM
6/15/2022

EXPERIMENT 2

2) Write JS code in an HTML page such that based on location selected by the user an AJAX request is made and weather details for that location are fetched and displayed. – "index.html"

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
      <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css">
       <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
       <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
       <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
       <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"></script>
      <link rel="stylesheet" href="mystyle.css">
   <title>Weather App</title>
```

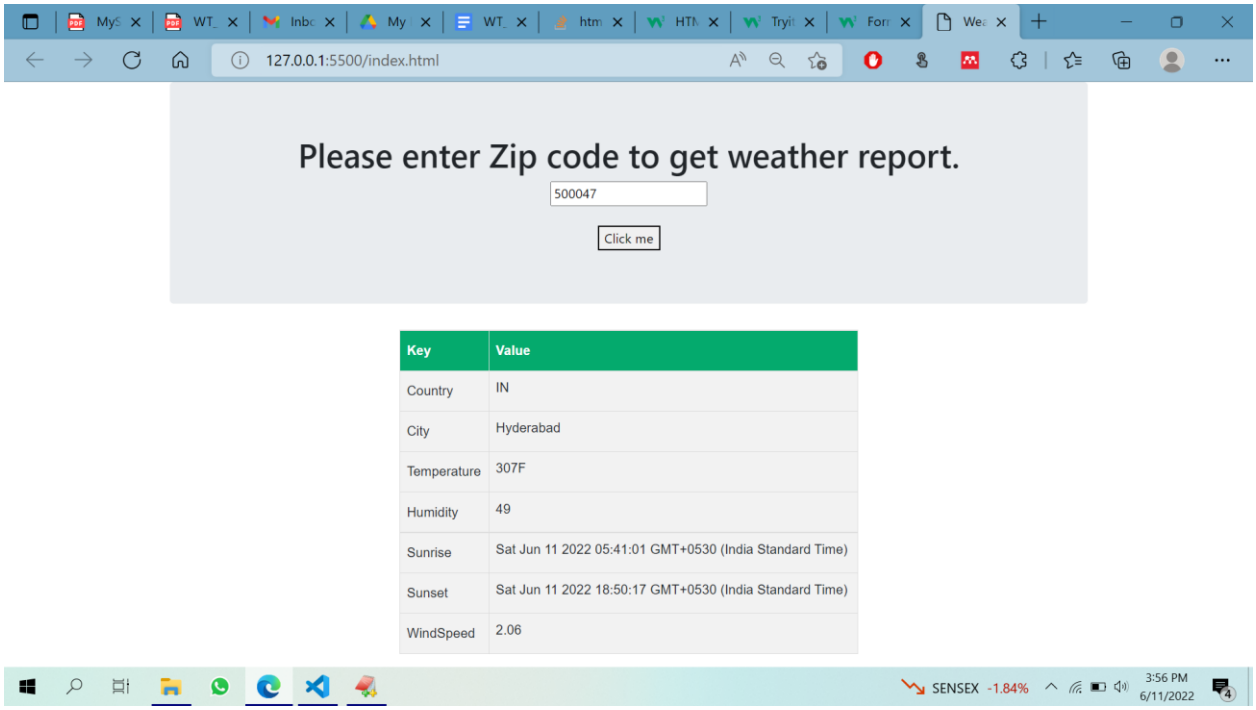
```
  <style>
     #weather {
  font-family: Arial, Helvetica, sans-serif;
  border-collapse: collapse;
  width: 50%;
  margin-left:25%;
  margin-right:15%;
}

#weather td, #weather th {
  border: 1px solid #ddd;
  padding: 8px;
}

#weather tr:nth-child(even){background-color: #f2f2f2;}

#weather tr:hover {background-color: #ddd;}

#weather th {
  padding-top: 12px;
  padding-bottom: 12px;
  text-align: left;
  background-color: #04AA6D;
  color: white;
}
  </style>
  <script>
        function getWeather(){
        //doucment.getElementBy

        let request = new XMLHttpRequest();
        let zip=document.getElementById('tb1').value;
```

```
request.open('GET','https://api.openweathermap.org/data/2.5/weather?q='+zip+'&appid=93f26e
3c57081a6210de53b8dcfdfea4',true);



    request.onload = function() {
  if( request.status >=200 && request.status < 400){
    let data = JSON.parse(request.responseText);
    console.log(data);
    var icon = "https://openweathermap.org/img/w/"+data.weather[0].icon+".png"
        document.getElementById('temp').innerHTML = data.main.temp+ 'F';
        document.getElementById('country').innerHTML=data.sys.country;
        document.getElementById('city').innerHTML=data.name;
        let tim= new Date(data.sys.sunrise * 1000);
        document.getElementById('sunrise').innerHTML=tim;
        let tim1= new Date(data.sys.sunset * 1000);
        document.getElementById('sunset').innerHTML=tim1;
        document.getElementById('windspeed').innerHTML=data.wind.speed;
        document.getElementById('humid').innerHTML=data.main.humidity;



      }
  else{
    console.log('failed connecting')
  }
}

request.onerror = function() {
  console.log(" Error ")
}

request.send();
```

```
        }


    </script>
</head>
<body>
    <div class="container">
    <div class="jumbotron text-center">
        <h1>Please enter Zip code to get  weather report.</h1>
        <input type="text" placeholder="enter zip code"id="tb1"><br><br>
    <button type="submit" vlaue="Click for weather" onclick="getWeather()">Click
me</button>
      </div>



   <table class="table-bordered table-striped" id="weather" >
    <thead>
        <tr>
          <th>Key</th>
          <th>Value</th>
        </tr>
      </thead>

      <tbody>
        <tr>
            <td>Country</td>
            <td><label id="country"></label></td>
        </tr>
        <tr>
            <td>City</td>
            <td ><label id="city"></label></td>
        </tr>
        <tr>
            <td>Temperature</td>
            <td ><label id="temp"></label></td>
```

```
        </tr>
        <tr>
            <td>Humidity</td>
            <td ><label id="humid"></label></td>
        </tr>
        <tr>
            <td>Sunrise</td>
            <td ><label id="sunrise"></label></td>
        </tr>
        <tr>
            <td>Sunset</td>
            <td ><label id="sunset"></label></td>
        </tr>
        <tr>
            <td>WindSpeed</td>
            <td ><label id="windspeed"></label></td>
        </tr>
        </tbody>

    </table>
</div>
</body>
</html>
```

EXPECTED OUTPUT :



EXPERIMENT 3

3) Write a Node JS program that accepts a port from the user and runs a node server at that port --"server.js"

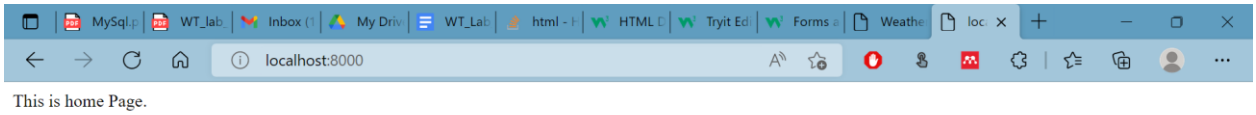
```
var http = require('http');


var server = http.createServer(function (req, res) {
   if (req.url == '/') { //check the URL of the current request


      // set response header
      res.writeHead(200, { 'Content-Type': 'text/html' });
```

```
    // set response content
    res.write('<html><body><p>This is home Page.</p></body></html>');
    res.end();
  }
  else if (req.url == "/student") {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.write('<html><body><p>This is student Page.</p></body></html>');
    res.end();
  }
  else if (req.url == "/admin") {
    res.writeHead(200, { 'Content-Type': 'text/html' });
    res.write('<html><body><p>This is admin Page.</p></body></html>');
    res.end();
  }
  else
    res.end('Invalid Request!');


});

server.listen(8000);
console.log('Node.js web server at port 8000 is running..')
```

EXPECTED OUTPUT

node server.js

Node.js web server at port 8000 is running..



EXPERIMENT 4

4. Write a NodeJS program to read from a file and display the content on screen – "readfile.js"

```
var fs = require('fs');
try {
   var data = fs.readFileSync('my-file.txt', 'utf8');
```

```
    console.log(data);
} catch(e) {
    console.log('Error:', e.stack);
}
```

my-file.txt

Keshav Memorial Institute of Technology (KMIT), established in 2007, is one of the premier engineering colleges in the state of Telangana.

KMIT is sponsored by Keshav Memorial Education Society (KMES), well known in Hyderabad, for the past 75 years, for running various educational institutions of repute.

KMIT is approved by All India Council for Technical Education (AICTE), New Delhi, and affiliated to Jawaharlal Nehru Technological University (JNTU), Hyderabad and recognized by the Govt. of Telangana.

KMIT is co-promoted and powered by Genesis Solutions Pvt. Ltd, a premier institute in Hyderabad imparting industry focused software training and education in emerging technologies and having tie-ups with leading MNCs.

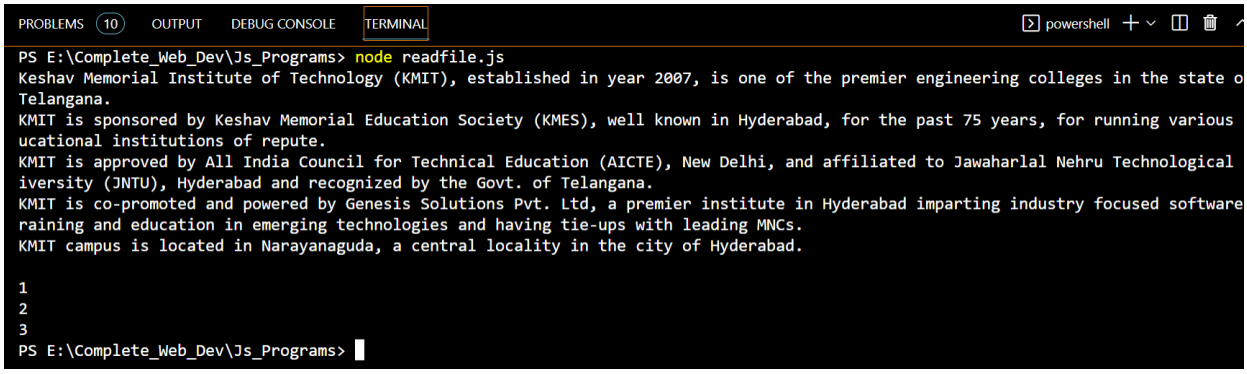
KMIT campus is located in Narayanaguda, a central locality in the city of Hyderabad.

1

2

3

EXPECTED OUTPUT



EXPERIMENT 5

5. **Create a NodeJS programme that allows users to submit text and a file name, and if the file already exists, appends the text to the file. If not, make a fresh file and include the text in it.**

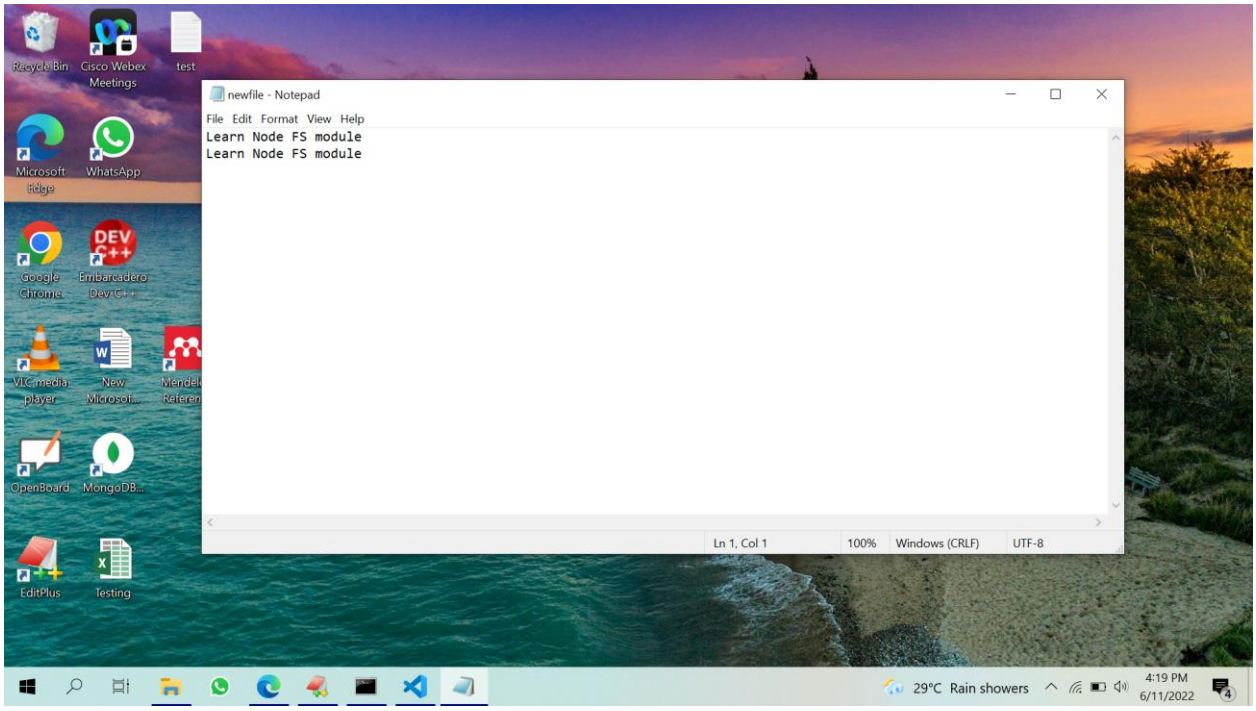
```
var fs = require('fs');
```

```
// appendFile function with filename, content and callback function
fs.appendFile('newfile.txt', `Learn Node FS module \r\n`, function (err) {
  if (err) throw err;
  console.log('File is Appended successfully.');
});
```

EXPECTED OUTPUT

File is appended successfully.

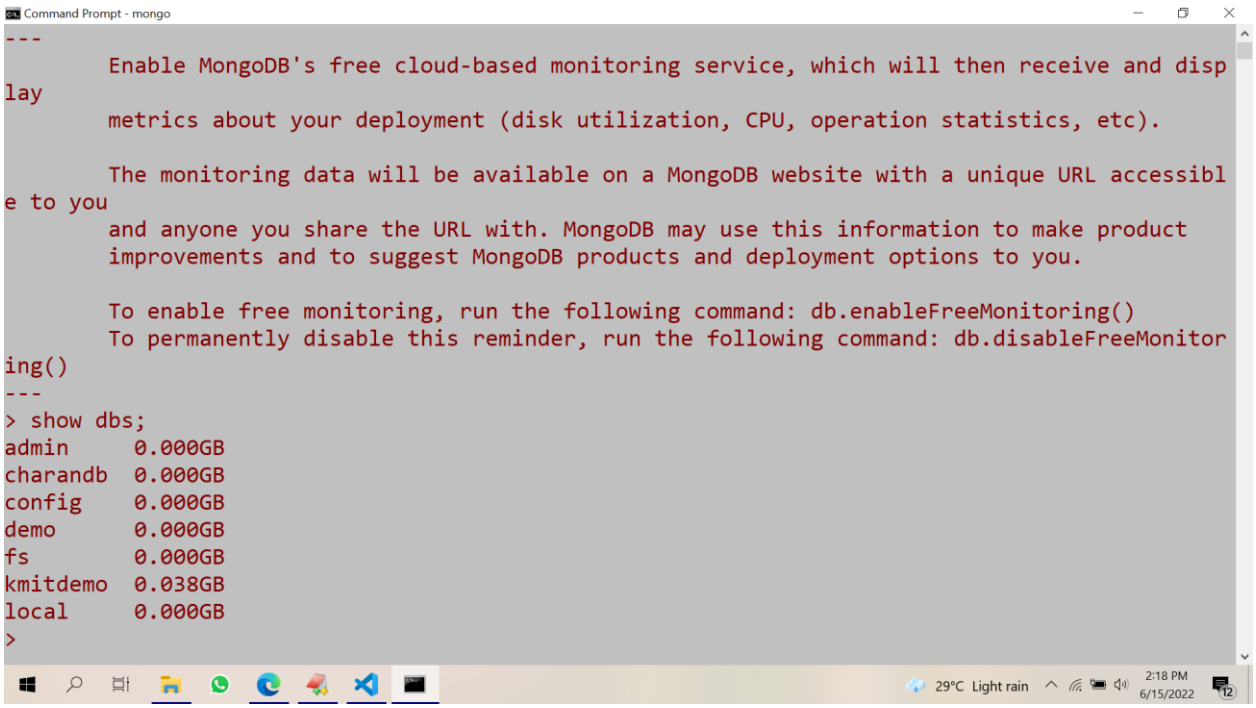
newfile.txt –



EXPERIMENT 6

6. Create a student database in MongoDB with all the details of students of a class

1.show dbs;



2. use student;

switched to db student

insert into studentinfo collection

3. db.studentinfo.insert({name:"john",id:"20bd1a05051",course:"b.tech",branch:"cse"})

WriteResult({ "nInserted" : 1 })

4. db.studentinfo.insert({name:"reena",id:"20bd1a0502",course:"M.tech",branch:"it"})

WriteResult({ "nInserted" : 1 })

5. db.studentinfo.insert({name:"ram",id:"20bd1a0503",course:"b.tech",branch:"cse"})

WriteResult({ "nInserted" : 1 })

EXPECTED OUTPUT

view studentinfo collection:

6. db.studentinfo.find({})

{ "_id" : ObjectId("62a99e693dbaba59a0af05cf"), "name" : "john", "id" : "20bd1a05051", "course" : "b.tech", "branch" : "cse" }

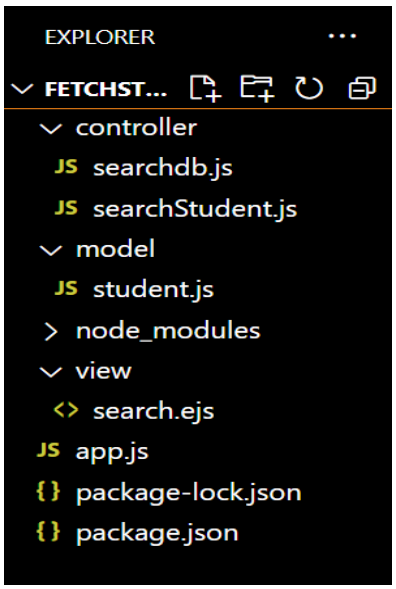
{ "_id" : ObjectId("62a99ebd3dbaba59a0af05d0"), "name" : "reena", "id" : "20bd1a0502", "course" : "M.tech", "branch" : "it" }

{ "_id" : ObjectId("62a99f123dbaba59a0af05d1"), "name" : "ram", "id" : "20bd1a0503", "course" : "b.tech", "branch" : "cse" }

EXPERIMENT 7

7. **Create a form such that, based on student roll number provided by user, the student details should be fetched (using ExpressJS)**

File structure :



app.js

```
var express = require('express');
var app = express();
const html = require('ejs')
const path=require('path')
app.use(express.json());
app.use(express.urlencoded({
  extended: true
}));

app.set('views', path.join(__dirname, 'view'))
app.set('view engine','ejs')
const mongoose=require('mongoose')
mongoose.connect('mongodb://localhost/student', function(error){
    if(error) console.log(error);
        console.log("connection successful");
});

const searchStudentController = require('./controller/searchStudent')
```

```
const fetchController= require('./controller/searchdb');
app.get('/search',searchStudentController)
app.post('/student/fetch',fetchController);

app.listen(3000, () => console.log("App listening on port 3000!"));
```

searchdb.js

```
const Student = require('../model/student.js')
module.exports = (req,res)=>{
      // console.log(req.body.rollno);
      // var test=Student.find({id: req.body.rollno});
      Student.find({id: req.body.rollno}, (error, student) => {
         if(error){
            console.log(student)
             res.render('search',{stuData:student});
         }
         else{

             console.log(student)
             res.render('search',{stuData:student, view:true});
         }

      })
}

searchStudent.js
module.exports = (req, res) =>{
   student=[]
   res.render('search',{stuData:student, view:false});
}
```

model/student.js

```
const mongoose = require('mongoose')
const Schema = mongoose.Schema;

const StudentSchema = new Schema({
    name: {
        type: String,
        required: true,
        unique: true
    },
    id: {
        type: String,
        required: true
    },
    course: {
        type: String,
        required: true
    } ,
    branch: {
        type: String,
        required: true
    }
})


// export model
const Student = mongoose.model('studentinfo',StudentSchema);
module.exports = Student
```

view/search.ejs

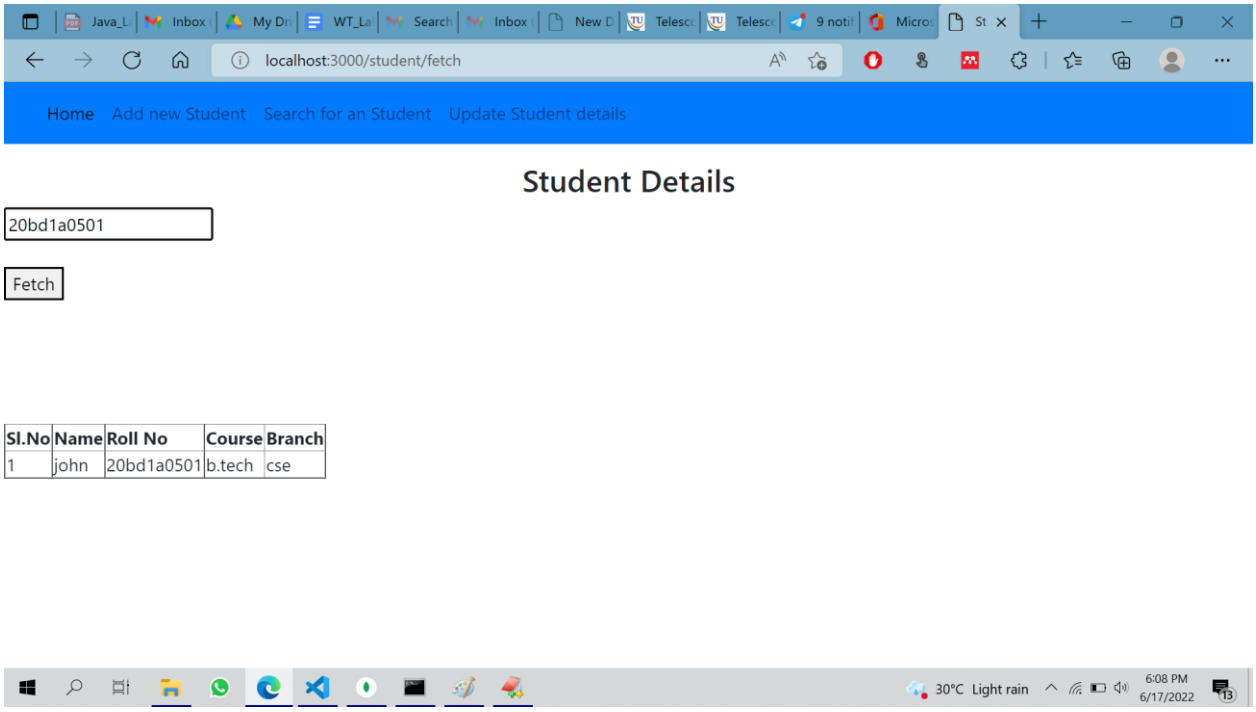
```
<html>
    <head>
        <title>Student Portal</title>
```

```
      <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
      <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
   </head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-primary fixed-top">
   <div class="container-fluid">
    <button
      class="navbar-toggler"
      type="button"
      data-mdb-toggle="collapse"
      data-mdb-target="#navbarExample01"
      aria-controls="navbarExample01"
      aria-expanded="false"
      aria-label="Toggle navigation"
    >

      <i class="fas fa-bars"></i>
    </button>
    <div class="collapse navbar-collapse" id="navbarExample01">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item active">
          <a class="nav-link" aria-current="page" href="/">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Add new Student</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="./search.ejs">Search for an Student </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Update Student details</a>
        </li>
```

```
      </ul>
    </div>
  </div>
</nav>
  <br><br><br>
  <h3 style="text-align: center;">Student Details </h3>

  <form action="/student/fetch" method="POST">
     <input type="text" placeholder="Search by rollno" id="rollno" name="rollno"> <br><br>
     <button type="submit">Fetch</button>
  </form>
  <br><br>
<br><br>
  <%
     if(view){
  %>
        <table border="1" id="table">
     <tr>
       <th>Sl.No</th>
        <th>Name</th>
        <th>Roll No</th>
        <th>Course</th>
        <th>Branch</th>
     </tr>
     <%
     if(stuData.length!=0){
     var i=1;
     stuData.forEach(function(data){
     %>
     <tr>
        <td><%=i; %></td>
        <td><%=data.name %></td>
        <td><%=data.id %></td>
```

```
            <td><%=data.course %></td>
            <td><%=data.branch %></td>
        </tr>
        <%  i++; }) %>
        <% } else{ %>
            <tr>
                <td colspan="4">No Data Found</td>
            </tr>
        <% } %>
    </table>
    <% } %>
</body>
</html>
```

EXPECTED OUTPUT



EXPERIMENT 8

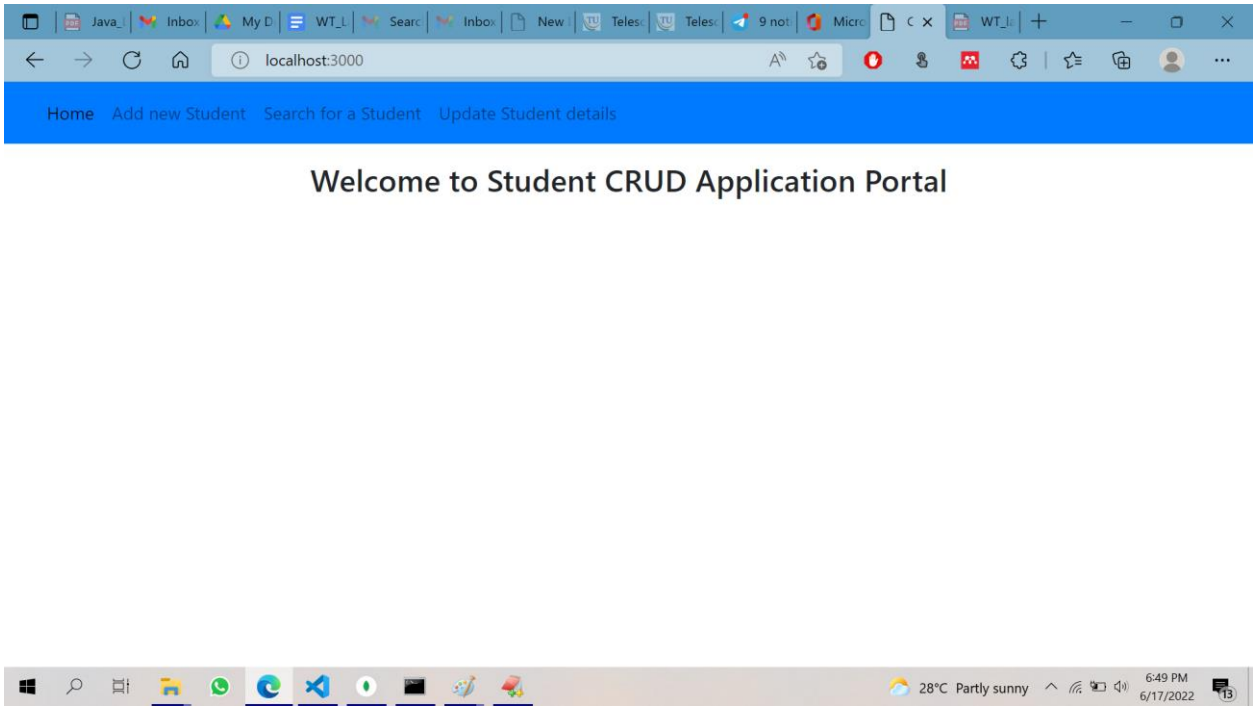
8. **Create a form with ExpressJS that enables CRUD (Create, Read, Update and Delete) operations on the student database.**

home.ejs

```
<html>
   <head>
      <title>Customer Portal</title>
      <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
      <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
   </head>
<body>



<nav class="navbar navbar-expand-lg navbar-light bg-primary fixed-top">
   <div class="container-fluid">
     <button
      class="navbar-toggler"
      type="button"
      data-mdb-toggle="collapse"
      data-mdb-target="#navbarExample01"
      aria-controls="navbarExample01"
      aria-expanded="false"
      aria-label="Toggle navigation"
     >

      <i class="fas fa-bars"></i>
     </button>
     <div class="collapse navbar-collapse" id="navbarExample01">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item active">
          <a class="nav-link" aria-current="page" href="/">Home</a>
        </li>
        <li class="nav-item">
```

```
          <a class="nav-link" href="/page/register">Add new Student</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/search">Search for an Student </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/update">Update Student details</a>
        </li>
      </ul>
    </div>
  </div>
 </nav>

  <br><br><br>
  <h3 style="text-align: center;">Welcome to Student CRUD Application Portal </h3>
</body>
</html>
```

EXPECTED OUTPUT :

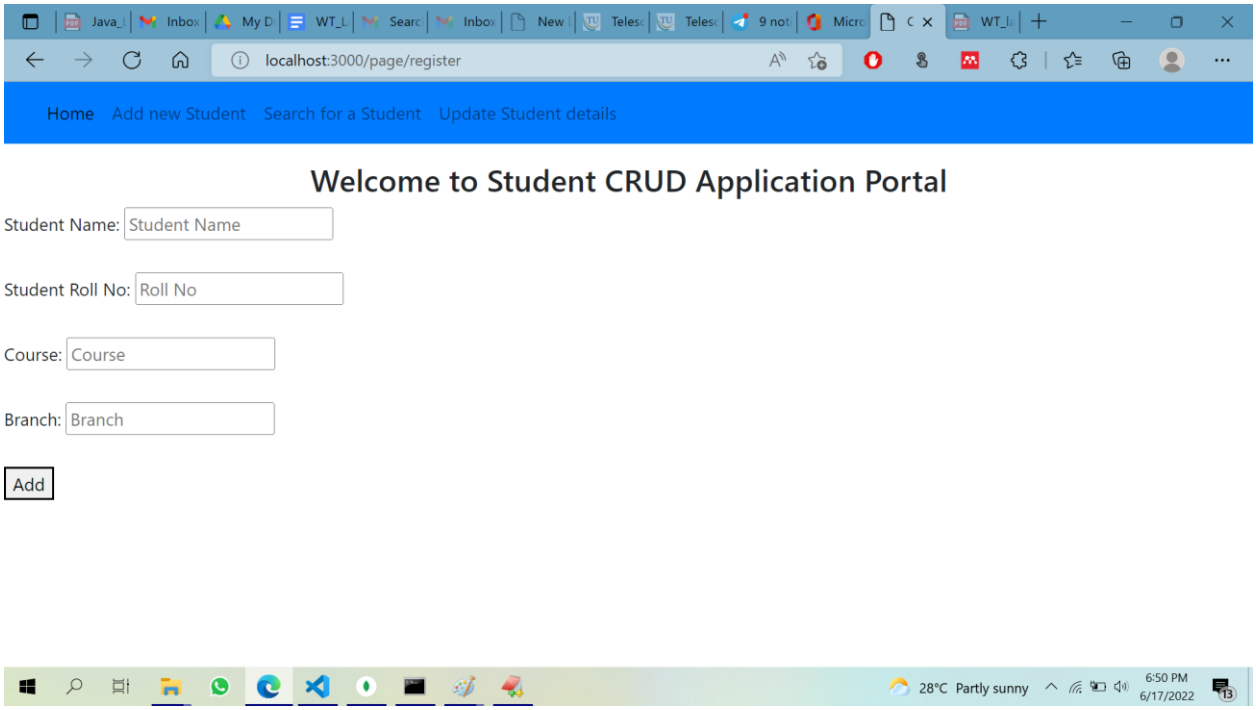


register.ejs

```
<html>
   <head>
      <title>Customer Portal</title>
      <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
      <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
   </head>
<body>

<nav class="navbar navbar-expand-lg navbar-light bg-primary fixed-top">
   <div class="container-fluid">
    <button
     class="navbar-toggler"
     type="button"
     data-mdb-toggle="collapse"
     data-mdb-target="#navbarExample01"
     aria-controls="navbarExample01"
     aria-expanded="false"
     aria-label="Toggle navigation"
    >
     <i class="fas fa-bars"></i>
    </button>
    <div class="collapse navbar-collapse" id="navbarExample01">
     <ul class="navbar-nav me-auto mb-2 mb-lg-0">
      <li class="nav-item active">
       <a class="nav-link" aria-current="page" href="/">Home</a>
      </li>
      <li class="nav-item">
       <a class="nav-link" href="/page/register">Add new Student</a>
      </li>
      <li class="nav-item">
       <a class="nav-link" href="/page/search">Search for a Student </a>
```

```
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/update">Update Student details</a>
        </li>
       </ul>
     </div>
   </div>
 </nav>

   <br><br><br>
   <h3 style="text-align: center;">Welcome to Student CRUD Application Portal </h3>

   <form action="/customer/register" method="POST">
     <label for="sname">Student Name:</label>
       <input type="text" placeholder="Student Name" id="name" name="name"> <br><br>
       <label for="id">Student Roll No:</label>
       <input type="text" placeholder="Roll No" id="id" name="id"> <br><br>
       <label for="course">Course:</label>
       <input type="text" placeholder="Course" id="course" name="course" > <br><br>
       <label for="branch">Branch:</label>
       <input type="text" placeholder="Branch" id="branch" name="branch"> <br><br>
       <button type="submit" >Add</button>
   </form>
   <h1> <%= status %> </h1>
</body>
</html>
```

EXPECTED OUTPUT :



search.ejs

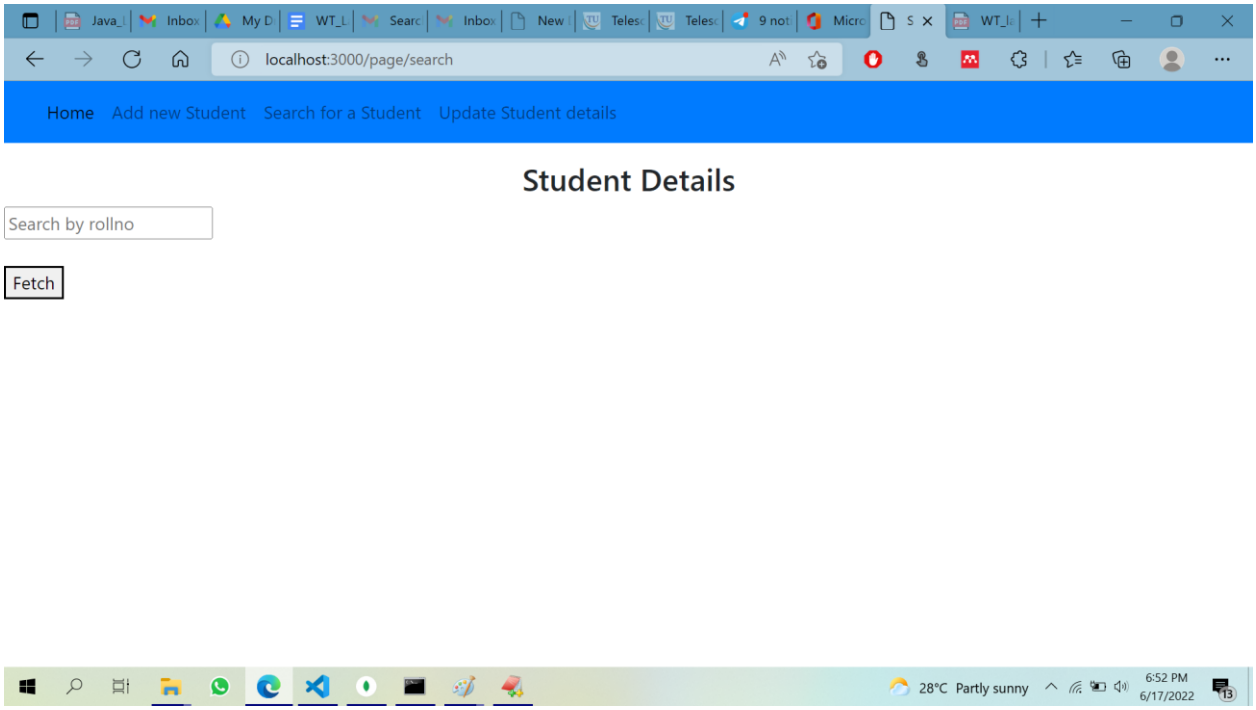
```
<html>
   <head>
      <title>Student Portal</title>
      <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
      <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
   </head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-primary fixed-top">
   <div class="container-fluid">
     <button
      class="navbar-toggler"
      type="button"
      data-mdb-toggle="collapse"
      data-mdb-target="#navbarExample01"
      aria-controls="navbarExample01"
```

```
      aria-expanded="false"
      aria-label="Toggle navigation"
    >

      <i class="fas fa-bars"></i>
    </button>
    <div class="collapse navbar-collapse" id="navbarExample01">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item active">
          <a class="nav-link" aria-current="page" href="/">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/register">Add new Student</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="./search.ejs">Search for a Student </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/update">Update Student details</a>
        </li>
      </ul>
    </div>
  </div>
</nav>
  <br><br><br>
  <h3 style="text-align: center;">Student Details </h3>

  <form action="/student/fetch" method="POST">
      <input type="text" placeholder="Search by rollno" id="rollno" name="rollno"> <br><br>
      <button type="submit">Fetch</button>
  </form>
  <br><br>
<br><br>
```

```
<%
    if(view){
%>
        <table border="1" id="table">
    <tr>
      <th>Sl.No</th>
        <th>Name</th>
        <th>Roll No</th>
        <th>Course</th>
        <th>Branch</th>
    </tr>
    <%
    if(stuData.length!=0){
    var i=1;
    stuData.forEach(function(data){
    %>
    <tr>
        <td><%=i; %></td>
        <td><%=data.name %></td>
        <td><%=data.id %></td>
        <td><%=data.course %></td>
        <td><%=data.branch %></td>
    </tr>
    <%  i++; }) %>
    <% } else{ %>
        <tr>
            <td colspan="4">No Data Found</td>
        </tr>
    <% } %>
</table>
<% } %>
</body>
</html>
```

Ex

EXPECTED OUTPUT :

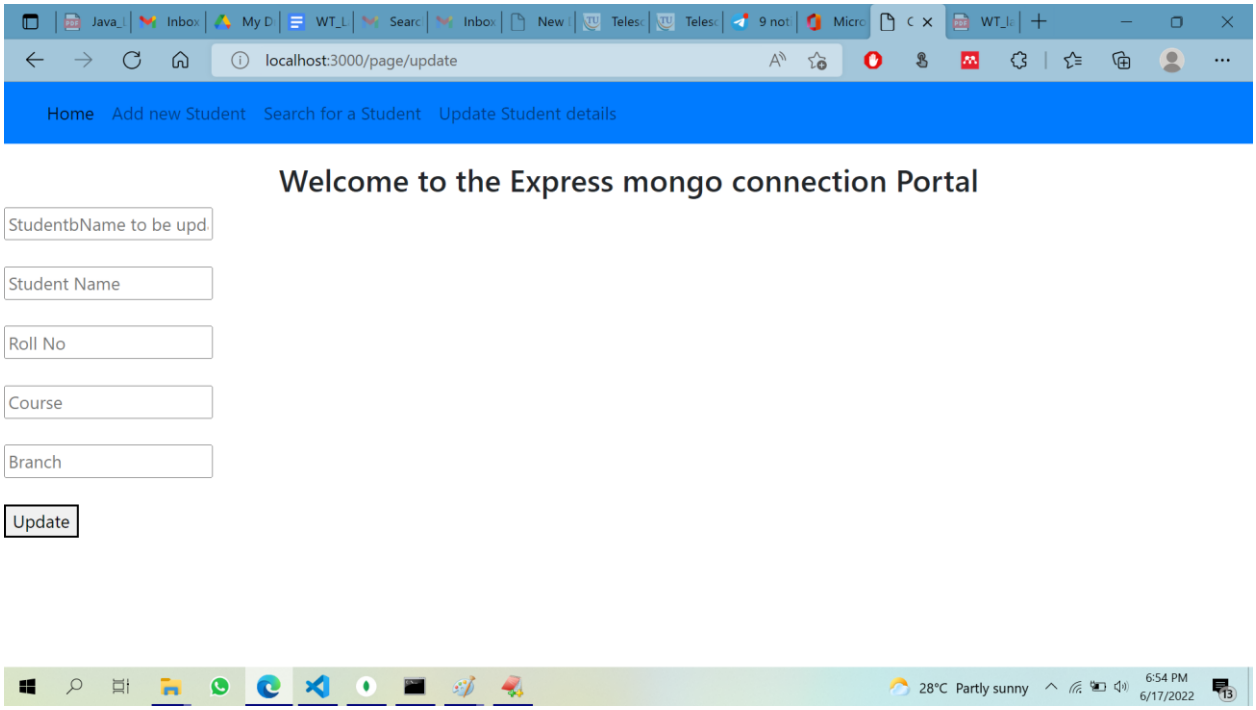


update.ejs

```
<html>
   <head>
      <title>Customer Portal</title>
      <link                                                    rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
      <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
   </head>
<body>

<nav class="navbar navbar-expand-lg navbar-light bg-primary fixed-top">
   <div class="container-fluid">
     <button
      class="navbar-toggler"
      type="button"
```

```
      data-mdb-toggle="collapse"
      data-mdb-target="#navbarExample01"
      aria-controls="navbarExample01"
      aria-expanded="false"
      aria-label="Toggle navigation"
    >
      <i class="fas fa-bars"></i>
    </button>
    <div class="collapse navbar-collapse" id="navbarExample01">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item active">
          <a class="nav-link" aria-current="page" href="/">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/register">Add new Student</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/search">Search for a Student </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/update">Update Student details</a>
        </li>
      </ul>
    </div>
  </div>
 </nav>

  <br><br><br>
  <h3 style="text-align: center;">Welcome to the Express mongo connection Portal </h3>

  <form action="/customer/update" method="POST">
      <input type="text" placeholder="StudentbName to be update" id="cname" name="cname">
<br><br>
      <input type="text" placeholder="Student Name" id="name" name="name"> <br><br>
      <input type="email" placeholder="Roll No" id="id" name="id"> <br><br>
      <input type="text" placeholder="Course" id="course" name="course" > <br><br>
      <input type="text" placeholder="Branch" id="branch" name="branch"> <br><br>
      <button type="submit" >Update</button>
  </form>
  <h1> <%= status %> </h1>
</body>
</html>
```

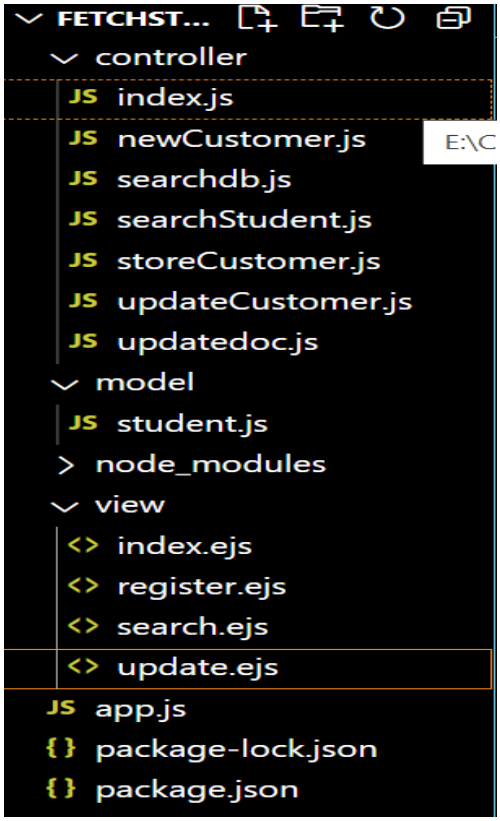
EXPECTED OUTPUT :



EXPERIMENT 9

9. Create a simple website for the CRUD operations on student DB and apply Express Routing.

file Structure:



controller/index.js

```
module.exports = (req, res) =>{
   res.render('index')
}
```

newCustomer.js

```
module.exports = (req, res) =>{
   data = '   '
   res.render('register',{status:data}); // render register.ejs
}
```

searchdb.js

```
const Student = require('../model/student.js')
module.exports = (req,res)=>{
       // console.log(req.body.rollno);
       // var test=Student.find({id: req.body.rollno});
       Student.find({id: req.body.rollno}, (error, student) => {
```

```
            if(error){
                console.log(student)
                res.render('search',{stuData:student});
            }
            else{

                console.log(student)
                res.render('search',{stuData:student, view:true});
            }

        })
}

searchStudent.js

module.exports = (req, res) =>{
    student=[]
    res.render('search',{stuData:student, view:false});
}

storeCustomer.js
const Student = require('../model/student.js')
const path = require('path')

module.exports = (req,res)=>{ console.log(req.body)

    Student.create(req.body, (error, student) => {
        if(error){
            console.log(error)
            data='Can Not Insert Student Details Please Try Agin';
            res.render('register',{status:data});
        }
        else{
            data = 'Student Details Inserted Successfully'
```

```
            res.render('register',{status:data});
        }

    })
}
```

updateCustomer.js

```
module.exports = (req, res) =>{
    data = '   '
    res.render('update',{status:data}); // render update.ejs
}
```

updatedoc.js

```
const Student = require('../model/student.js')

module.exports = (req,res)=>{
    var test={};
    if(req.body.name){
    test.name=req.body.name };
    if(req.body.id){
        test.id=req.body.id };
    if(req.body.course){
            test.course=req.body.course };
    if(req.body.branch){
                test.branch=req.body.branch };
               // console.log(test);

Student.updateOne({name: req.body.cname}, { $set: test  },(error, student) => {
    if(error){
        console.log(student)
        data='Can Not Update Student Details Please Try Agin';
        res.render('update',{status:data});
```

```
    }
    else{
        data = 'Student Details Updated Successfully'
        res.render('update',{status:data});
    }
}
)
}
//{cname: req.body.cnamenew, email: req.body.email, phone: req.body.phone, city:
req.body.city }
```

model/student.js

```
const mongoose = require('mongoose')
const Schema = mongoose.Schema;

const StudentSchema = new Schema({
    name: {
        type: String,
        required: true,
        unique: true
    },
    id: {
        type: String,
        required: true
    },
    course: {
        type: String,
        required: true
    } ,
    branch: {
        type: String,
        required: true
```

```
    }
})


// export model
const Student = mongoose.model('studentinfo',StudentSchema);
module.exports = Student
```

home.ejs

```
<html>
    <head>
        <title>Customer Portal</title>
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
    </head>
<body>



<nav class="navbar navbar-expand-lg navbar-light bg-primary fixed-top">
    <div class="container-fluid">
      <button
        class="navbar-toggler"
        type="button"
        data-mdb-toggle="collapse"
        data-mdb-target="#navbarExample01"
        aria-controls="navbarExample01"
        aria-expanded="false"
        aria-label="Toggle navigation"
      >
```

```
      <i class="fas fa-bars"></i>
    </button>
    <div class="collapse navbar-collapse" id="navbarExample01">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item active">
          <a class="nav-link" aria-current="page" href="/">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/register">Add new Student</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/search">Search for an Student </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/update">Update Student details</a>
        </li>
      </ul>
    </div>
  </div>
 </nav>

  <br><br><br>
  <h3 style="text-align: center;">Welcome to Student CRUD Application Portal </h3>
</body>
</html>
```

register.ejs

```
<html>
  <head>
    <title>Customer Portal</title>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
```

```
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
  </head>
<body>

<nav class="navbar navbar-expand-lg navbar-light bg-primary fixed-top">
  <div class="container-fluid">
   <button
    class="navbar-toggler"
    type="button"
    data-mdb-toggle="collapse"
    data-mdb-target="#navbarExample01"
    aria-controls="navbarExample01"
    aria-expanded="false"
    aria-label="Toggle navigation"
   >
    <i class="fas fa-bars"></i>
   </button>
   <div class="collapse navbar-collapse" id="navbarExample01">
    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
     <li class="nav-item active">
      <a class="nav-link" aria-current="page" href="/">Home</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="/page/register">Add new Student</a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="/page/search">Search for a Student </a>
     </li>
     <li class="nav-item">
      <a class="nav-link" href="/page/update">Update Student details</a>
     </li>
    </ul>
   </div>
```

```
    </div>
  </nav>

    <br><br><br>
    <h3 style="text-align: center;">Welcome to Student CRUD Application Portal </h3>

    <form action="/customer/register" method="POST">
      <label for="sname">Student Name:</label>
        <input type="text" placeholder="Student Name" id="name" name="name"> <br><br>
        <label for="id">Student Roll No:</label>
        <input type="text" placeholder="Roll No" id="id" name="id"> <br><br>
        <label for="course">Course:</label>
        <input type="text" placeholder="Course" id="course" name="course" > <br><br>
        <label for="branch">Branch:</label>
        <input type="text" placeholder="Branch" id="branch" name="branch"> <br><br>
        <button type="submit" >Add</button>
    </form>
    <h1> <%= status %> </h1>
</body>
</html>
```

search.ejs

```
<html>
    <head>
        <title>Student Portal</title>
        <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
        <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
    </head>
<body>
```

```
<nav class="navbar navbar-expand-lg navbar-light bg-primary fixed-top">
   <div class="container-fluid">
    <button
     class="navbar-toggler"
     type="button"
     data-mdb-toggle="collapse"
     data-mdb-target="#navbarExample01"
     aria-controls="navbarExample01"
     aria-expanded="false"
     aria-label="Toggle navigation"
    >

     <i class="fas fa-bars"></i>
    </button>
    <div class="collapse navbar-collapse" id="navbarExample01">
     <ul class="navbar-nav me-auto mb-2 mb-lg-0">
      <li class="nav-item active">
       <a class="nav-link" aria-current="page" href="/">Home</a>
      </li>
      <li class="nav-item">
       <a class="nav-link" href="/page/register">Add new Student</a>
      </li>
      <li class="nav-item">
       <a class="nav-link" href="./search.ejs">Search for a Student </a>
      </li>
      <li class="nav-item">
       <a class="nav-link" href="/page/update">Update Student details</a>
      </li>
     </ul>
    </div>
   </div>
  </nav>
   <br><br><br>
```

```
  <h3 style="text-align: center;">Student Details </h3>

  <form action="/student/fetch" method="POST">
     <input type="text" placeholder="Search by rollno" id="rollno" name="rollno"> <br><br>
     <button type="submit">Fetch</button>
  </form>
  <br><br>
<br><br>
  <%
     if(view){
  %>
        <table border="1" id="table">
     <tr>
       <th>Sl.No</th>
        <th>Name</th>
        <th>Roll No</th>
        <th>Course</th>
        <th>Branch</th>
     </tr>
     <%
     if(stuData.length!=0){
     var i=1;
     stuData.forEach(function(data){
     %>
     <tr>
        <td><%=i; %></td>
        <td><%=data.name %></td>
        <td><%=data.id %></td>
        <td><%=data.course %></td>
        <td><%=data.branch %></td>
     </tr>
     <%  i++; }) %>
     <% } else{ %>
```

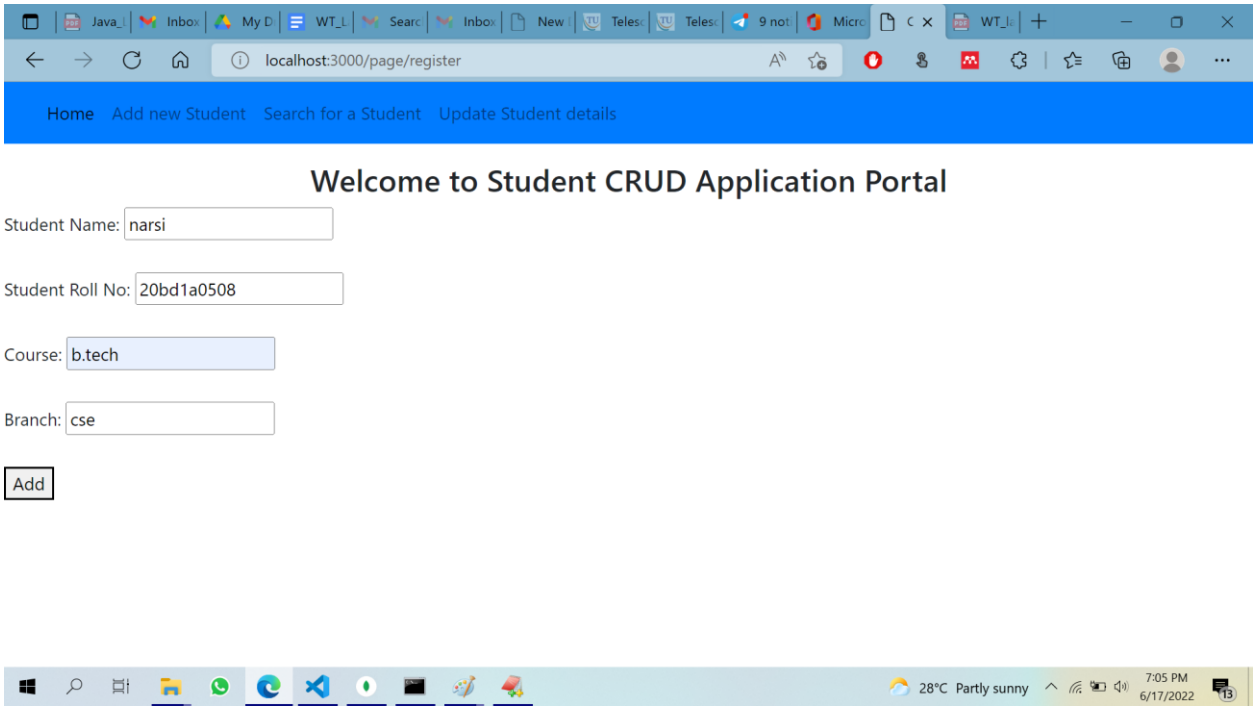
```
        <tr>
            <td colspan="4">No Data Found</td>
        </tr>
      <% } %>
   </table>
   <% } %>
</body>
</html>
```

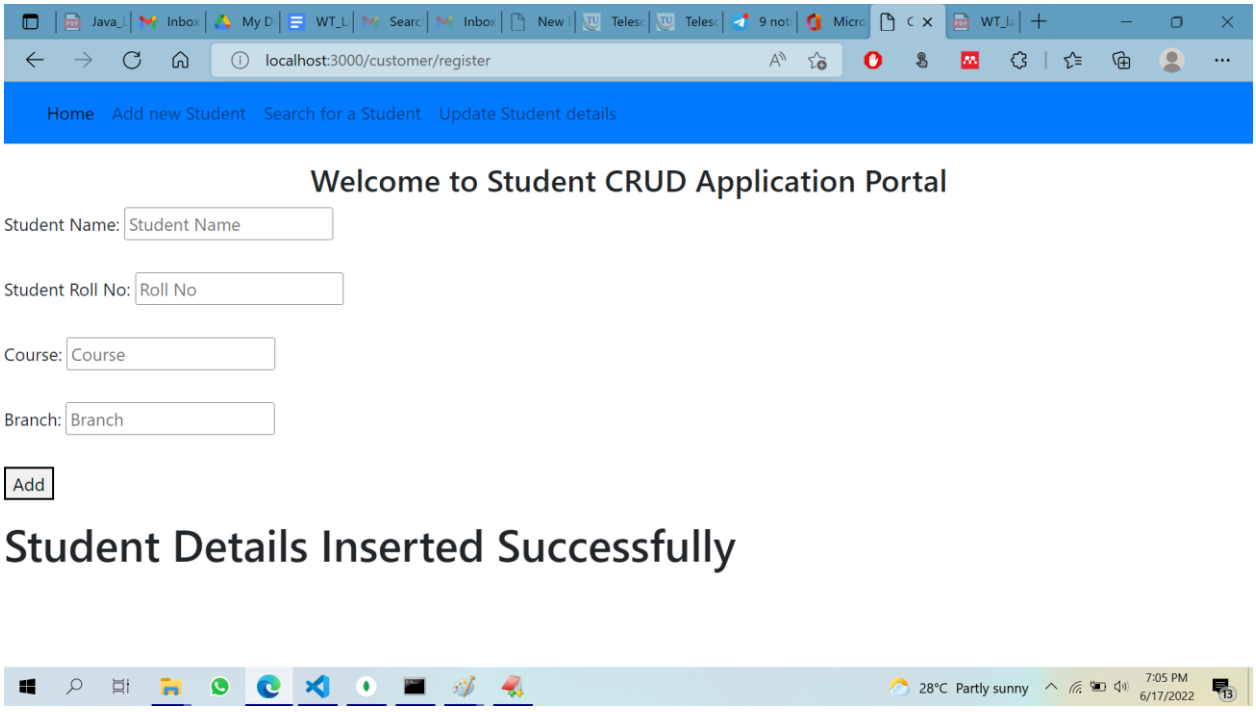
update.ejs

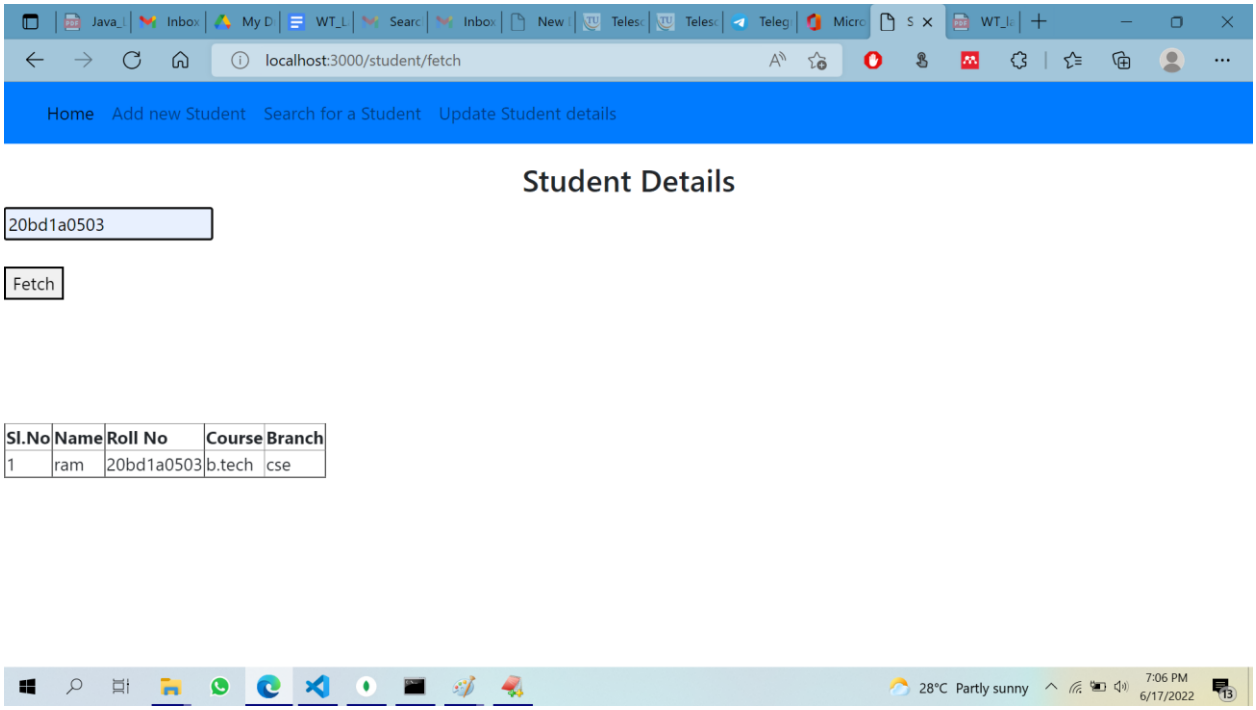
```
<html>
   <head>
      <title>Customer Portal</title>
      <link                                                                    rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
      <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
   </head>
<body>

<nav class="navbar navbar-expand-lg navbar-light bg-primary fixed-top">
   <div class="container-fluid">
    <button
      class="navbar-toggler"
      type="button"
      data-mdb-toggle="collapse"
      data-mdb-target="#navbarExample01"
      aria-controls="navbarExample01"
      aria-expanded="false"
      aria-label="Toggle navigation"
    >
      <i class="fas fa-bars"></i>
    </button>
    <div class="collapse navbar-collapse" id="navbarExample01">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item active">
          <a class="nav-link" aria-current="page" href="/">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/register">Add new Student</a>
        </li>
```

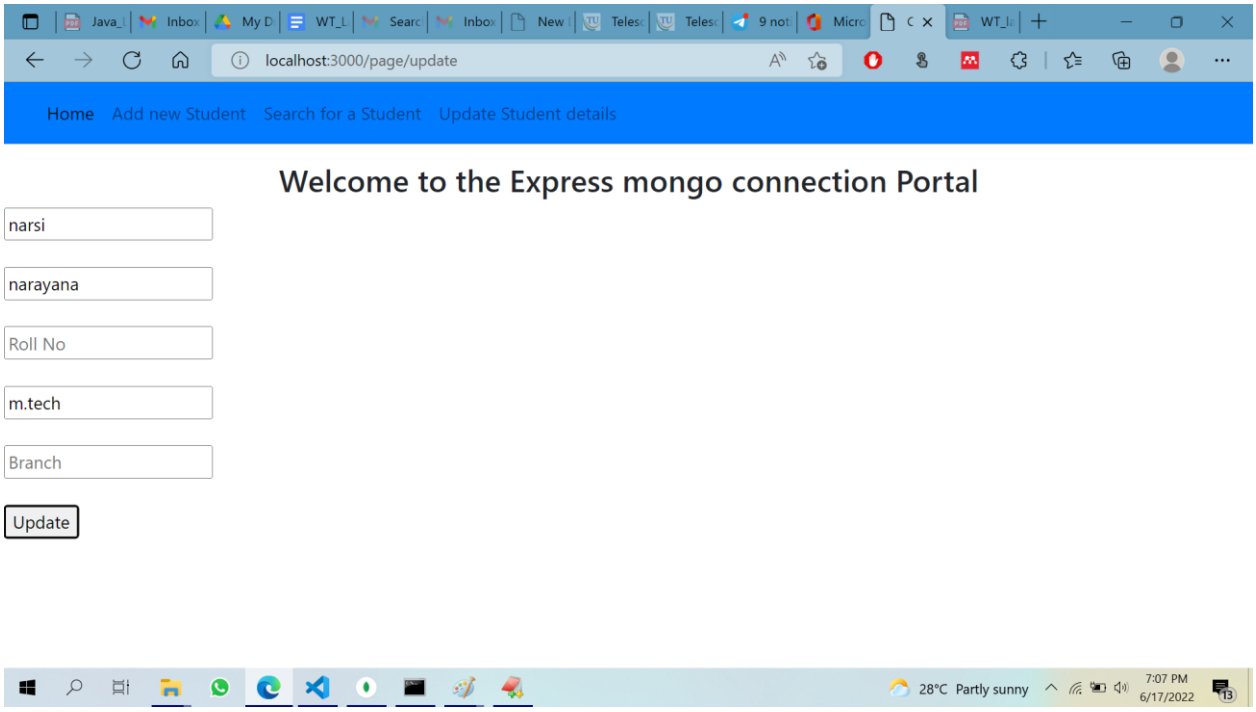
```
        <li class="nav-item">
          <a class="nav-link" href="/page/search">Search for a Student </a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/page/update">Update Student details</a>
        </li>
      </ul>
    </div>
  </div>
 </nav>

  <br><br><br>
  <h3 style="text-align: center;">Welcome to the Express mongo connection Portal </h3>

  <form action="/customer/update" method="POST">
      <input type="text" placeholder="StudentbName to be update" id="cname" name="cname">
<br><br>
      <input type="text" placeholder="Student Name" id="name" name="name"> <br><br>
      <input type="email" placeholder="Roll No" id="id" name="id"> <br><br>
      <input type="text" placeholder="Course" id="course" name="course" > <br><br>
      <input type="text" placeholder="Branch" id="branch" name="branch"> <br><br>
      <button type="submit" >Update</button>
  </form>
  <h1> <%= status %> </h1>
</body>
</html>
```

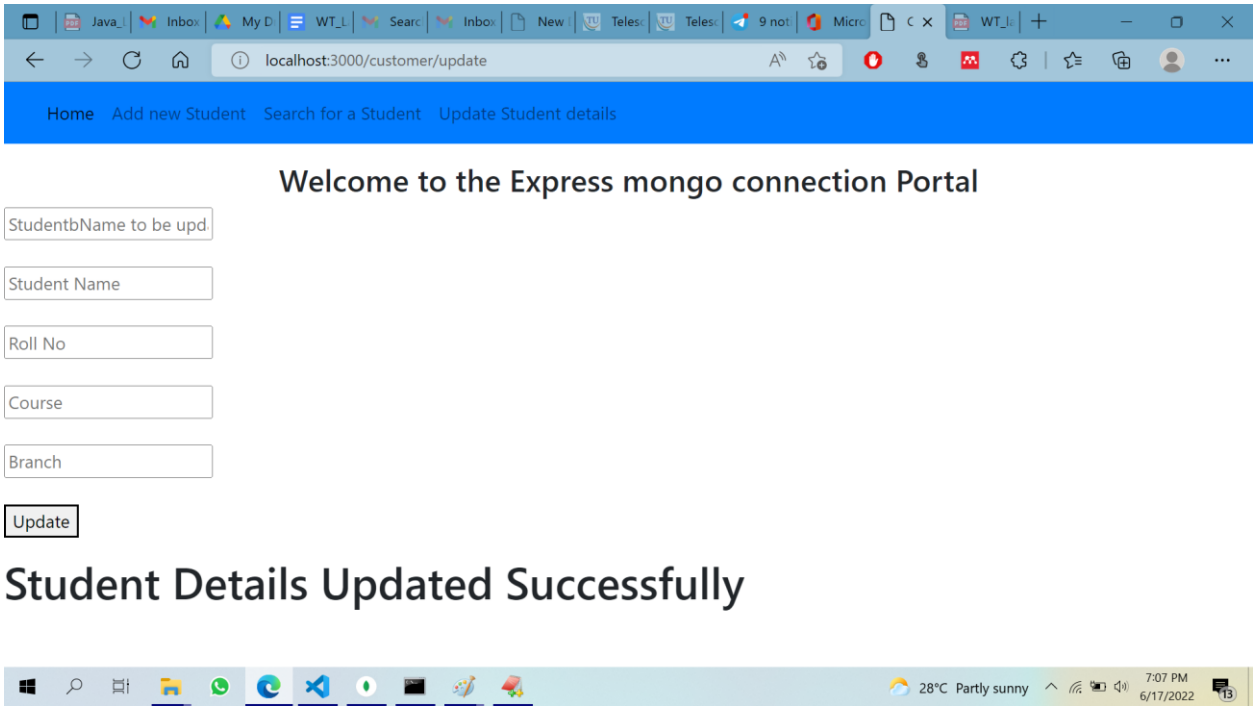
EXPECTED OUTPUT

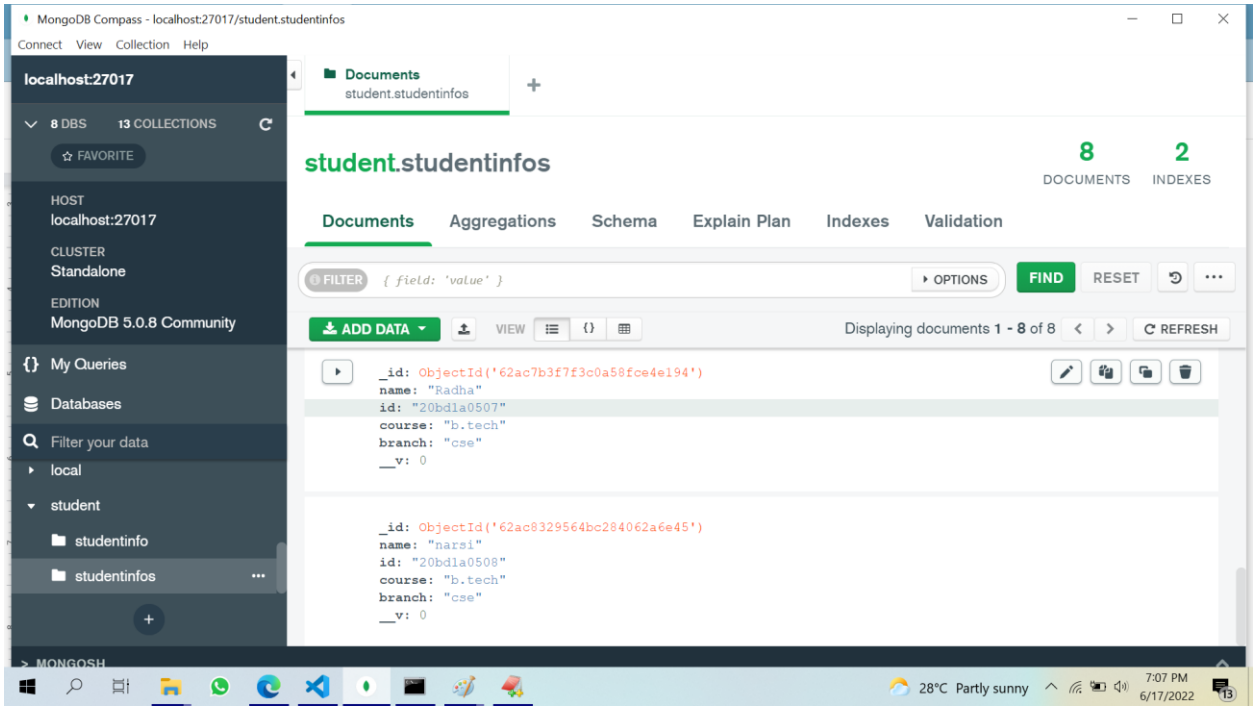

localhost:3000/page/register
Home
Add new Student
Search for a Student
Update Student details
Welcome to Student CRUD Application Portal
Student Name: narsi
Student Roll No: 20bd1a0508
Course: b.tech
Branch: cse
Add
28°C Partly sunny
7:05 PM
6/17/2022

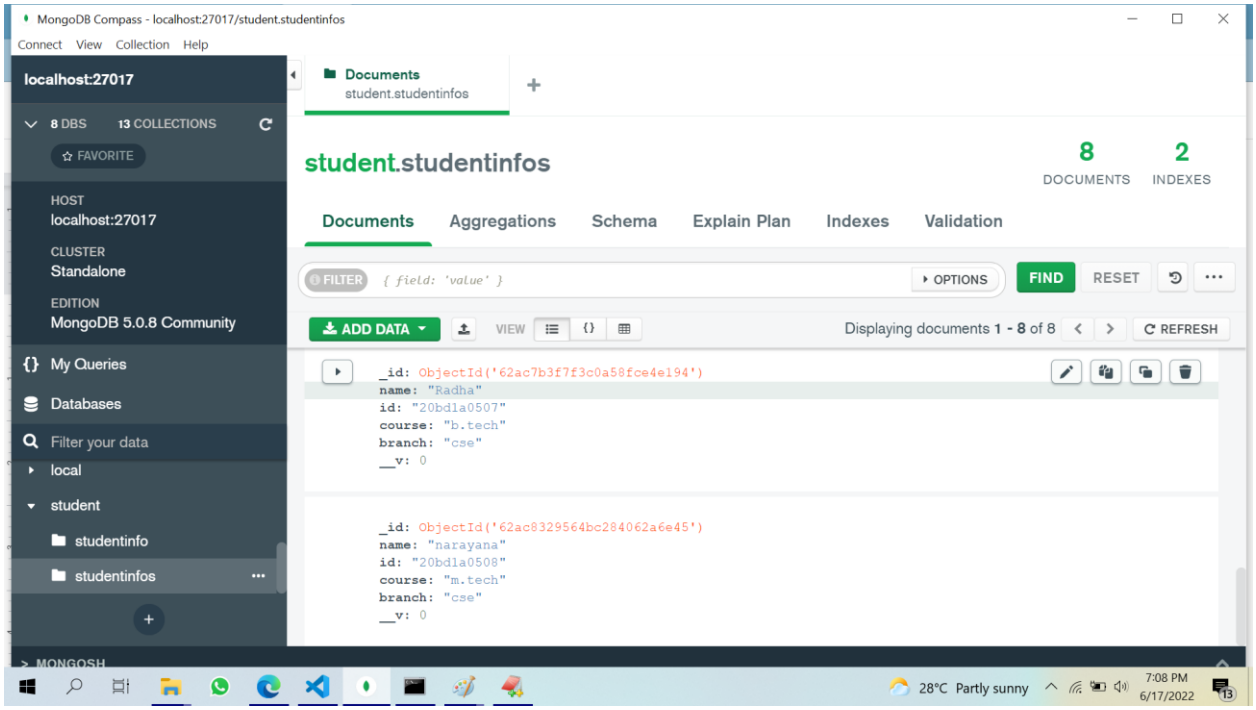

localhost:3000/customer/register
Home
Add new Student
Search for a Student
Update Student details
Welcome to Student CRUD Application Portal
Student Name: Student Name
Student Roll No: Roll No
Course: Course
Branch: Branch
Add
Student Details Inserted Successfully
28°C Partly sunny
7:05 PM
6/17/2022


localhost:3000/student/fetch
Home
Add new Student
Search for a Student
Update Student details
Student Details
20bd1a0503
Fetch
Sl.No
Name
Roll No
Course
Branch
1
ram
20bd1a0503
b.tech
cse
28°C Partly sunny
7:06 PM
6/17/2022


localhost:3000/page/update
Home
Add new Student
Search for a Student
Update Student details
Welcome to the Express mongo connection Portal
narsi
narayana
Roll No
m.tech
Branch
Update
28°C Partly sunny
7:07 PM
6/17/2022


localhost:3000/customer/update
Home
Add new Student
Search for a Student
Update Student details
Welcome to the Express mongo connection Portal
StudentbName to be upd
Student Name
Roll No
Course
Branch
Update
Student Details Updated Successfully


MongoDB Compass - localhost:27017/student.studentinfos
Connect View Collection Help
localhost:27017
8 DBS 13 COLLECTIONS
FAVORITE
HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 5.0.8 Community
My Queries
Databases
Filter your data
local
student
studentinfo
studentinfos
Documents
student.studentinfos
student.studentinfos
8 DOCUMENTS
2 INDEXES
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
FILTER
{ field: 'value' }
OPTIONS
FIND
RESET
ADD DATA
VIEW
Displaying documents 1 - 8 of 8
REFRESH
_id: ObjectId('62ac7b3f7f3c0a58fce4e194')
name: "Radha"
id: "20bd1a0507"
course: "b.tech"
branch: "cse"
__v: 0
_id: ObjectId('62ac8329564bc284062a6e45')
name: "narsi"
id: "20bd1a0508"
course: "b.tech"
branch: "cse"
__v: 0
MONGOSH


MongoDB Compass - localhost:27017/student.studentinfos
Connect View Collection Help
localhost:27017
8 DBS 13 COLLECTIONS
FAVORITE
HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 5.0.8 Community
My Queries
Databases
Filter your data
local
student
studentinfo
studentinfos
Documents
student.studentinfos
student.studentinfos
8
DOCUMENTS
2
INDEXES
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
FILTER
{ field: 'value' }
OPTIONS
FIND
RESET
ADD DATA
VIEW
Displaying documents 1 - 8 of 8
REFRESH
_id: ObjectId('62ac7b3f7f3c0a58fce4e194')
name: "Radha"
id: "20bd1a0507"
course: "b.tech"
branch: "cse"
__v: 0
_id: ObjectId('62ac8329564bc284062a6e45')
name: "narayana"
id: "20bd1a0508"
course: "m.tech"
branch: "cse"
__v: 0
MONGOSH
28°C Partly sunny
7:08 PM
6/17/2022