

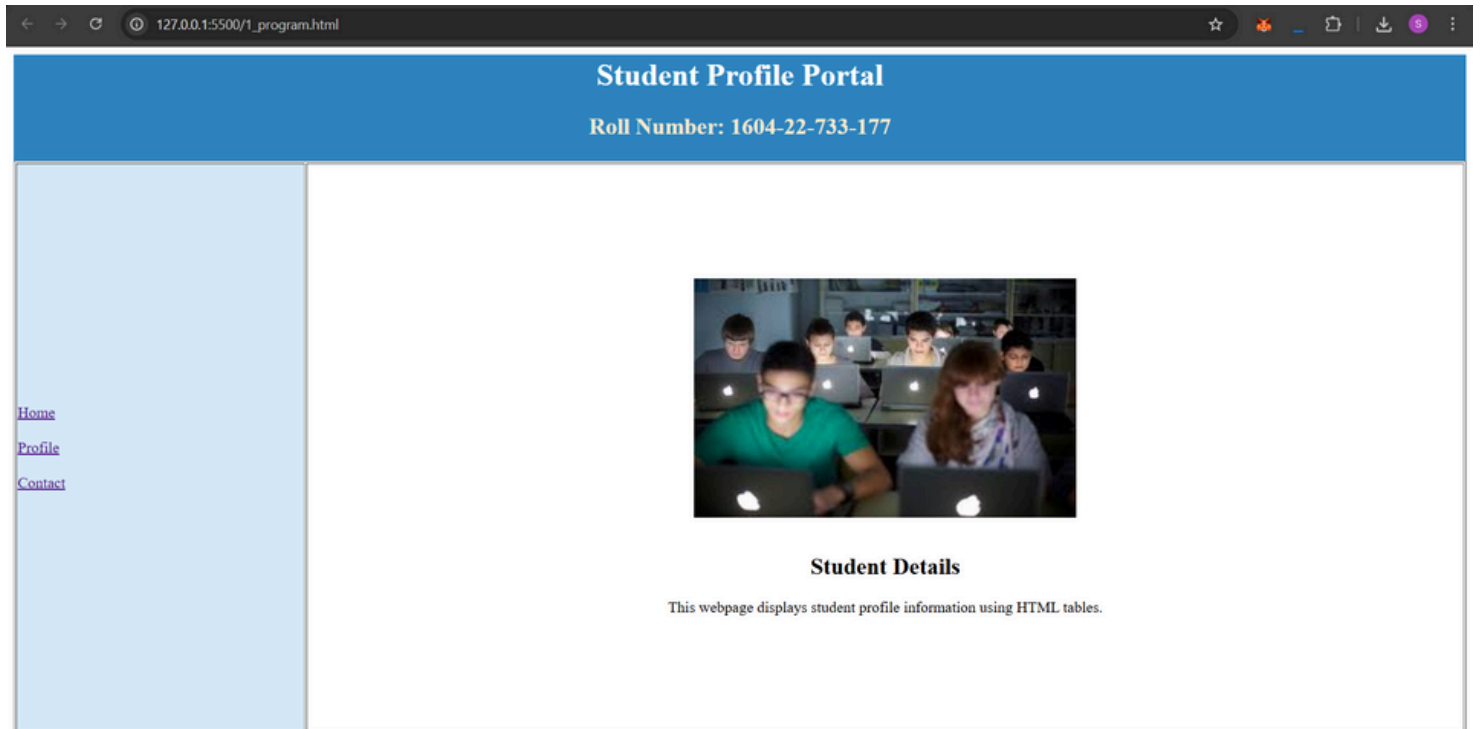
## Program-1: Develop a static website using HTML Tables

```
<html>
<head>
  <title>Student Profile</title>
</head>
<body>
  <tablewidth="100%" bgcolor="#2E86C1">
    <tr>
      <td>
        <center>
          <h1style="color: white">Student Profile Portal</h1>
          <h2style="color: #fdebd0">Roll Number: 1604-22-733-177</h2>
        </center>
      </td>
    </tr>
  </table>

  <tablewidth="100%" height="600" border="1">
    <tr>
      <tdwidth="20%" bgcolor="#D6EAF8">
        <a href="#">Home</a><br /><br />
        <a href="#">Profile</a><br /><br />
        <a href="#">Contact</a>
      </td>

      <tdwidth="80%">
        <center>
          <br/><br />
          <h2>Student Details</h2>
          <p>
            This webpage displays student profile information using HTML
            tables.
          </p>
        </center>
      </td>
    </tr>
  </table>
</body>
</html>
```

**Output:**



## Program-2: Develop a static website using DIV and CSS

```
<html>
<head>
<title>Department Overview</title>

<style>
.header {
background-color:#D4EFDF;
text-align:center;
padding:20px;
}
.left {
width:20%;
float:left;
background-color:#D6EAF8;
height:400px;
}
.right {
width:80%;
float:left;
padding:20px;
}
</style>

<link rel="stylesheet" href="styles.css">
</head>

<body>

<div class="header">
<h1>MJCET</h1>
<h2>Department of CSE</h2>
<p>Roll Number: 1604-22-733-005</p>
</div>

<div class="left">
<a href="#">Vision</a><br><br>
<a href="#">Mission</a><br><br>
<a href="#">Faculty</a>
</div>

<div class="right">
<br><br>
<p>
The CSE department provides quality education and practical exposure.
```

**Output:**

## **MJCET**

### **Department of CSE**

Roll Number: 1604-22-733-177

Vision

Mission

Faculty



The CSE department provides quality education and practical exposure for students to excel in the field of Computer Science.

### Program-3: Develop a registration page using HTML forms

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

<body bgcolor="#F2F3F4">

<h1 align="center">Student Registration Form</h1>

<form>
<table align="center" cellpadding="8" bgcolor="#D6EAF8">

<tr>
<td>Roll Number:</td>
<td><input type="text"></td>
</tr>

<tr>
<td>Name:</td>
<td><input type="text"></td>
</tr>

<tr>
<td>Password:</td>
<td><input type="password"></td>
</tr>

<tr>
<td>Semester:</td>
<td>
<select>
<option>Semester
1</option>
<option>Semester
2</option>
<option>Semester
3</option>
</select>
</td>
</tr>

<tr>
<td>Section:</td>
<td>
<input type="radio" name="sec">A
```

```

<input type="radio" name="sec">B
</td>
</tr>

<tr>
<td>Subjects:</td>
<td>
<input type="checkbox">DBMS
<input type="checkbox">DSA
</td>
</tr>

<tr>
<td>Address:</td>
<td><textarea rows="3" cols="20"></textarea></td>
</tr>

<tr>
<td><input type="submit"></td>
<td><input type="reset"></td>
</tr>

</table>
</form>

</body>
</html>

```

**Output:**

## Student Registration Form

Roll Number:	<input type="text" value="1604-22-733-177"/>
Name:	<input type="text" value="Mohd Safwan Uddin"/>
Password:	<input type="password" value="....."/>
Semester:	<input type="text" value="Semester 3"/> ▼
Section:	<input checked="" type="radio"/> A <input type="radio"/> B
Subjects:	<input checked="" type="checkbox"/> DBMS <input type="checkbox"/> DSA
Address:	<input type="text" value="!!!!!!!!!!!!"/>
<input type="button" value="Submit"/> <input type="button" value="Reset"/>	

## Program-4: Develop a dynamic web page using JavaScript

```
<html>
<head>
  <title>Dynamic Page</title>
</head>

<body>
  <h2id="date"></h2>
  <h2id="greet"></h2>
  <h2id="lucky"></h2>
  <h2id="last"></h2>

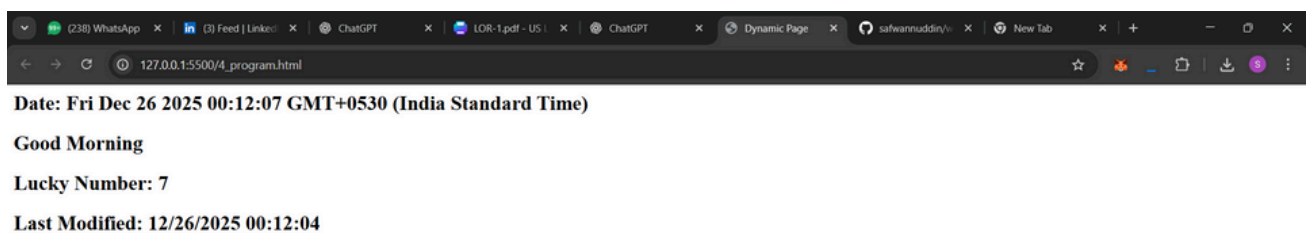
  <script>
    var d=new Date();
    document.getElementById("date").innerHTML = "Date: " + d;

    if(d.getHours() < 12)
      document.getElementById("greet").innerHTML = "Good Morning";
    elsedocument.getElementById("greet").innerHTML = "Good Afternoon";

    var n=Math.floor(Math.random() * 10);
    document.getElementById("lucky").innerHTML = "Lucky Number: " + n;

    document.getElementById("last").innerHTML =
      "LastModified: " + document.lastModified;
  </script>
</body>
</html>
```

## Output:



## Program-5 JavaScript Program to Validate Registration Form

```
<html>
<head>
  <title>Form Validation</title>

  <script>
    function validate() {
      var roll = document.getElementById("roll").value;
      if (roll == "") {
        alert("Roll Number cannot be empty");
        return false;
      }
      if (roll.search(/[0-9]{4}-[0-9]{2}-[0-9]{3}-[0-9]{3}$/) != 0) {
        alert("Invalid Roll Number");
        return false;
      }

      var name = document.getElementById("name").value;
      if (name == "" || name.search(/[A-Z a-z]*/) != 0) {
        alert("Invalid Name");
        return false;
      }

      var ph = document.getElementById("phone").value;
      if (ph.search(/[0-9]{10}$/) != 0) {
        alert("Invalid Phone Number");
        return false;
      }

      var email = document.getElementById("email").value;
      if (email.search(/[A-Z a-z 0-9._]*@[A-Z a-z]*\.[A-Z a-z]*/) != 0) {
        alert("Invalid Email");
        return false;
      }

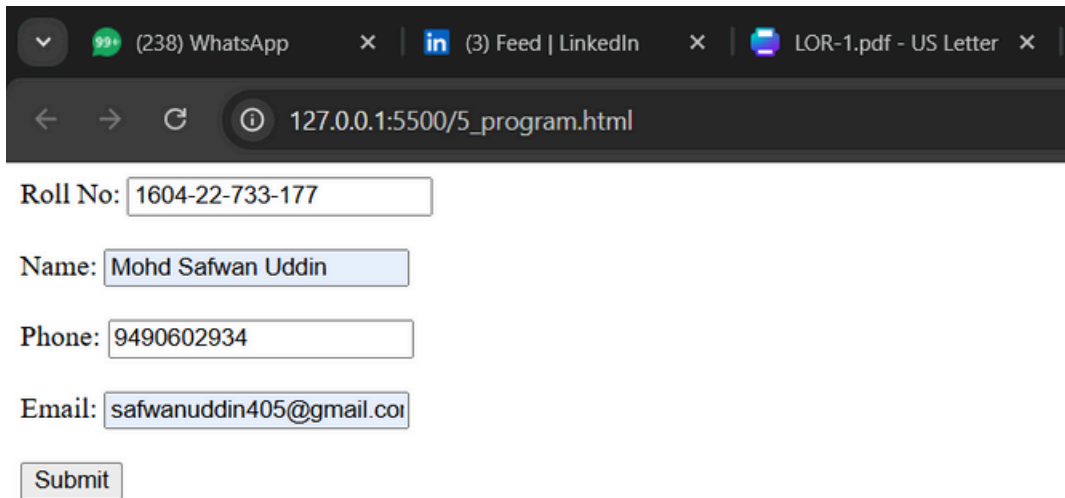
      return true;
    }
  </script>
</head>

<body>
  <form onsubmit="return validate();">
    Roll No: <input type="text" id="roll" /><br /><br />
    Name: <input type="text" id="name" /><br /><br />
```



```
Phone: <input type="text" id="phone" /><br /><br />
Email: <input type="text" id="email" /><br /><br />
<input type="submit" />
</form>
</body>
</html>
```

## Output:



The screenshot shows a web browser window with three tabs: "(238) WhatsApp", "(3) Feed | LinkedIn", and "LOR-1.pdf - US Letter". The address bar displays "127.0.0.1:5500/5\_program.html". The form contains the following fields and a button:

Roll No:	1604-22-733-177
Name:	Mohd Safwan Uddin
Phone:	9490602934
Email:	safwanuddin405@gmail.coi
<input type="submit" value="Submit"/>	

**Program-6:Create a dynamic web page using JavaScript event handling and DOM manipulation.**

```
<html>
<head>
  <title>EventHandling</title>

  <script>
    functionshowMessage() {
      alert("PageLoaded Successfully");
    }

    functionvalidate() {
      varrn=document.getElementById("roll");
      if(rn.value== "") {
        document.getElementById("lbl").style.color = "red";
      } else {
        document.getElementById("lbl").style.color = "black";
      }
    }

    functionmouseover() {
      document.getElementById("msg").innerHTML =
        "RollNumber: 1604-22-733-177";
    }

    functionmouseout() {
      document.getElementById("msg").innerHTML = "";
    }
  </script>
</head>

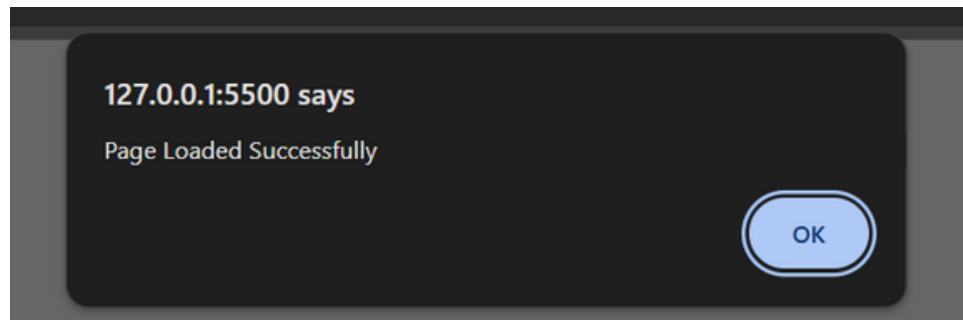
<bodyonload="showMessage()">
  <center>
    <h2>StudentEnrollment Form</h2>

    <formonsubmit="validate()">
      <labelid="lbl">Roll Number</label>
      <input
        ty pe="tex t"
        id="roll"
        onmouseover="mouseover()"
        onmouseout="mouseout()"
      />
      <divid="msg"></div>
```

```
<br /><br />

<input type="submit" />
</form>
</center>
</body>
</html>
```

### Output:



### Student Enrollment Form

Roll Number   
Roll Number: 1604-22-733-177

## Program-7: Demonstrate working of built-in node.js modules and file system

```
import http from "http";
import os from "os";
import fs from "fs";
import {promisify} from "util";

const open = promisify(fs.open);
const writeFile = promisify(fs.writeFile);
const appendFile = promisify(fs.appendFile);
const rename = promisify(fs.rename);
const readFile = promisify(fs.readFile);

const server = http.createServer(async (req, res) => {
  res.writeHead(200, {"Content-Type": "text/html"});

  // Display system information
  res.write("Welcome to Web Technologies Lab<br>");
  res.write("Current Date: " + Date() + "<br>");
  res.write("Operating Platform: " + os.platform() + "<br>");
  res.write("System Architecture: " + os.arch() + "<br>");
  res.write("Host Name: " + os.hostname() + "<br>");
  res.write("Operating System Type: " + os.type() + "<br>");

  try {
    const text = await readFile("content.txt", "utf8");
    res.write("<br><b>File Content:</b><br>" + text);
  } catch (err) {
    res.write("<br><b>File Content:</b><br>Could not read content.txt");
  }

  res.end();
});

server.listen(8080, () => {
  console.log("Server running at http://localhost:8080");
});

(async () => {
  try {
    await open("myfile1.txt", "w");
    console.log("myfile1.txt created");
  } catch (err) {
    console.error("Error creating myfile1.txt:", err);
  }
})
```

```

try {
  await writeFile("myfile3.txt", "Hello Lab");
  console.log("Content written to myfile3.txt");
} catch (err) {
  console.error("Error writing to myfile3.txt:", err);
}

try {
  await appendFile("myfile1.txt", " Hello Web");
  console.log("Content appended to myfile1.txt");
} catch (err) {
  console.error("Error appending to myfile1.txt:", err);
}

try {
  await rename("myfile3.txt", "myfile4.txt");
  console.log("File renamed to myfile4.txt");
} catch (err) {
  console.error("Error renaming myfile3.txt:", err);
}
})();

console.log("Roll Number: 1604-22-733-177");

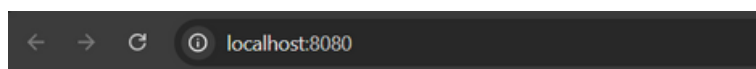
```

## Output:

```

safwa@LAPTOP-0I56PUPU MINGW64 ~/Downloads/7 sem/web-tech-lab (main)
$ node 7_program.js
Roll Number: 1604-22-733-177
Server running at http://localhost:8080
myfile1.txt created
Content written to myfile3.txt
Content appended to myfile1.txt
File renamed to myfile4.txt

```



## Welcome to Web Technologies Lab

Current Date: Fri Dec 26 2025 00:25:19 GMT+0530 (India Standard Time)  
 Operating Platform: win32  
 System Architecture: x64  
 Host Name: LAPTOP-0I56PUPU  
 Operating System Type: Windows\_NT

### File Content:

Could not read content.txt

## Program 8: Demonstrate routes and parameter handling in Express.js index.js

```
var express = require("express");
var app = express();

app.get("/", function(req, res) {
  res.send("Welcome to Home Page created using Express.js by Safwan");
});

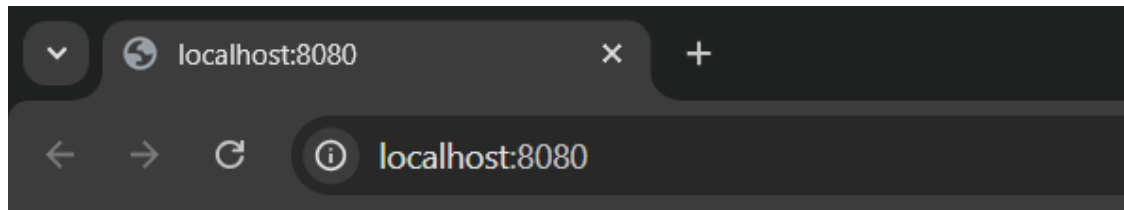
app.get("/signup", function(req, res) {
  res.send("Signup Page Welcome to Home Page created using Express.js by Safwan");
});

app.get("/login", function(req, res) {
  res.send("Login Page");
});

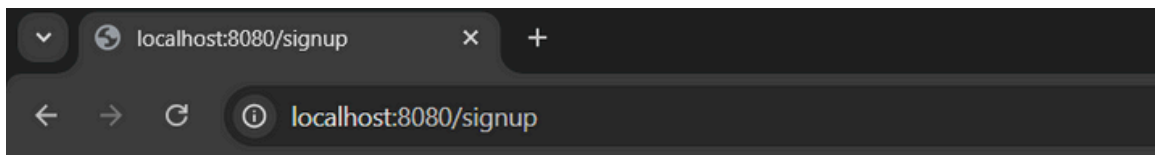
app.get("/display/:name/:pwd", function(req, res) {
  res.send("Username: " + req.params.name +
    "<br>Password: " + req.params.pwd);
});

app.listen(8080, function() {
  console.log("Server Running");
});
```

Output:



Welcome to Home Page created using Express.js by Safwan



Signup Page created using Express.js by Safwan

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

safwa@LAPTOP-0I56PUPU MINGW64 ~/Downloads/7 sem/web-tech-lab (main)
$ npm install express

up to date, audited 66 packages in 1s

22 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

safwa@LAPTOP-0I56PUPU MINGW64 ~/Downloads/7 sem/web-tech-lab (main)
$ node 8_program.js
Server Running at http://localhost:8080
```

## Program-9: Write a program to implement MVC architecture

### Index.js

```
var express=require("express");
var app = express();

app.use(function(req, res, next) {
  res.header("Access-Control-Allow-Origin", "*");
  res.header("Access-Control-Allow-Methods",
"GET,PUT,POST,DELETE,OPTIONS");
  res.header("Access-Control-Allow-Headers", "Origin, X-
Requested-With, Content-Type, Accept");
  next();
});

app.get("/", function(req, res) {
res.send("Welcome to Home Page");
});
app.get("/marks", function(req, res) {
  const marksData = [
    { id: 1, subject: "Math", imarks: 18, emarks: 67 },
    { id: 2, subject: "Physics", imarks: 17, emarks: 70 },
    { id: 3, subject: "Chemistry", imarks: 19, emarks: 65 },
    { id: 4, subject: "English", imarks: 20, emarks: 80 }
  ];
  res.json(marksData);
});

app.listen(8080, function() {
console.log("Server Running");
});
```

### index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Marks Memo</title>
  <script>
    document.addEventListener("DOMContentLoaded", function() {
      fetch('http://localhost:8080/marks')
        .then(response => response.json())
        .then(data => {
          console.log(data )
          const tableBody = document.getElementById('marks-body');
          data.forEach(d => {
```



```

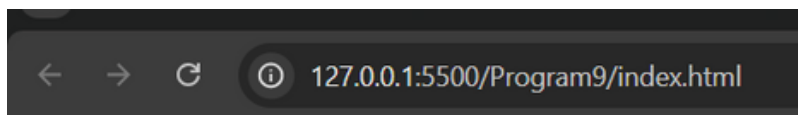
        const row = document.createElement('tr');
        row.innerHTML =
`<td>${d.id}</td><td>${d.subject}</td><td>${d.imarks}</td><td>${d.emarks}</td>`;
        tableBody.appendChild(row);
    });
})
.catch(err => {
    document.getElementById('error').textContent = 'Failed to load marks data.';
});
});
</script>
</head>
<body>

<h1>Student Marks Memo</h1>
<h4>Roll Number: 1604-22-733-177</h4>
<div id="error" style="color:red;"></div>
<table border="2">
    <tr>
        <td>ID</td>
        <td>Subject</td>
        <td>Internal Marks</td>
        <td>External Marks</td>
    </tr>
    <tbodyid="marks-body">
        <tbody>

</body>
</html>

```

**Output:**



# Student Marks Memo

**Roll Number: 1604-22-733-177**

ID	Subject	Internal Marks	External Marks
1	Math	18	67
2	Physics	17	70
3	Chemistry	19	65
4	English	20	80

### Program-10: Demonstrate rendering HTML and JSX using [React.js](#)

```
function WebTech() {  
  var title = "MJCET";  
  var d = new Date();  
  
  return (  
    <div>  
      <h1>{title}</h1>  
      <h2>Web Technologies Laboratory</h2>  
      <h4>Roll Number: 1604-22-733-177 </h4>  
      <p>{d.toDateString()}</p>  
      <GetVision />  
    </div>  
  );  
}  
  
function GetVision() {  
  return (  
    <p>  
    MJCET is a premier institution offering quality education.  
    </p>  
  );  
}  
  
export default WebTech;
```

**Output:**

← → ↻ ⓘ localhost:3000

**MJCET**

**Web Technologies Lab**

Roll Number: 1604-22-733-177

Fri Dec 26 2025

MJCET is a premier institution offering quality education.

### Program-11: Demonstrate use of props, events, lists, forms using

```
function WebTech() {
  return (
    <div>
      <h2>Web Technologies Laboratory</h2>
      <h4>Roll Number: 1604-22-733-177</h4>

      <User name="Mohd Safwan" uid={101} />
      <User name="Uddin" uid={102} />

      <MyForm />
      <StudentList />
      <EventDemo />
    </div>
  );
}

function User(props) {
  return (
    <div>
      <p>Name: {props.name}</p>
      <p>ID: {props.uid}</p>
    </div>
  );
}

function MyForm() {
  return (
    <form>
      Name: <input type="text" /><br /><br />
      Email: <input type="text" /><br /><br />
      Password: <input type="password" /><br /><br />
      <input type="submit" />
      <input type="reset" />
    </form>
  );
}

function StudentList() {
  var students = ["AAA", "BBB", "CCC"];
  return (
    <ul>
      {students.map((s, i) => <li key={i}>{s}</li>)}
    </ul>
  );
}
```

```
function EventDemo() {  
  function show() {  
    alert("Button Clicked");  
  }  
  return <button onClick={show}>Click Me</button>;  
}
```

export default WebTech;

## Output:

---

### Web Technologies Laboratory

Roll Number: 1604-22-733-177

Name: Mohd Safwan

ID: 101

Name: Uddin

ID: 102

---

Name:

Email:

Password:

- 
- AAA
  - BBB
  - CCC

---

## Program-12: Create a Single Page Application (SPA) using REST Service. WebTech.js

```
import { Component } from "react";

class WebTech extends Component {
  state = { students: [], imarks: [], emarks: [] };

  componentDidMount() {
    fetch("http://localhost:8080/students")
      .then(res => res.json())
      .then(data => this.setState({ students: data }));
  }

  loadIMarks = () => {
    fetch("http://localhost:8080/imarks")
      .then(res => res.json())
      .then(data => this.setState({ imarks: data }));
  };

  loadEMarks = () => {
    fetch("http://localhost:8080/emarks")
      .then(res => res.json())
      .then(data => this.setState({ emarks: data }));
  };

  render() {
    return (
      <div>
        <h2>Students Data</h2>
        <button onClick={this.loadIMarks}>Internal Marks</button>
        <button onClick={this.loadEMarks}>External Marks</button>

        <table border="1">
          <tr>
            <th>
              Roll No.
            </th>
            <td>
              1604-22-733-177
            </td>
          </tr>
          <tr>
            <th>Name</th>
            <td>
              WebTech
            </td>
          </tr>
        </table>
      </div>
    );
  }
}
```

```

        <th>IMarks</th>
        <th>EMarks</th>
    </tr>
    <tr>
        <td>{this.state.students.map(s => <div>{s.name}</div>)}</td>
        <td>{this.state.imarks.map(i => <div>{i.marks}</div>)}</td>
        <td>{this.state.emarks.map(e => <div>{e.marks}</div>)}</td>
    </tr>
</table>
</ div>
    );
}
}

```

```
export default WebTech;
```

### Index.js

```

var express = require("express");
var cors = require("cors");

```

```

var app = express();
app.use(cors());

```

```

var students = [
    {id:101,name:"Safwan" },
    {id:102,name:"Ahad" },
    {id:103,name:"haseeb" },
    {id:104,name:"Ansar" },
    {id:105,name:"omer" }
];

```

```

var imarks = [
    {id:101,marks:28 },
    {id:102,marks:30 },
    {id:103,marks:27 },
    {id:104,marks:29 },
    {id:105,marks:26 }
];

```

```

var emarks = [
    {id:101,marks:70 },
    {id:102,marks:68 },
    {id:103,marks:66 },
    {id:104,marks:72 },
    {id:105,marks:65 }
];

```

```
app.get("/students", function (req, res) {  
  res.json(students);  
});
```

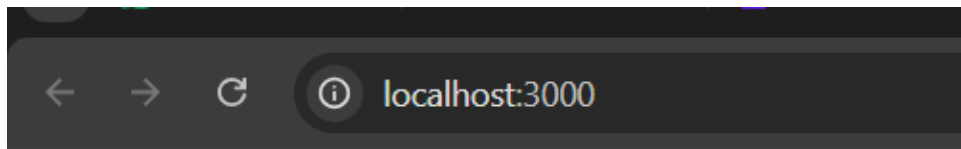
```
app.get("/imarks", function (req, res) {  
  res.json(imarks);  
});
```

```
app.get("/emarks", function (req, res) {  
  res.json(emarks);  
});
```

```
app.listen(8080, function () {  
  console.log("REST Service running on port 8080");  
});
```

```
<td>  
{students.map(st=><div>{st.name}</div>}}  
</td>  
<td>  
{imarks.map(im=><div>{im.marks}</div>}}  
</td>  
<td>  
{emarks.map(em=><div>{em.marks}</div>}}  
</td>  
</tr>  
</table>  
</div>  
}  
}  
export default App;
```

Output:



## Students Data

Internal Marks		External Marks	
Roll No.			
1604-22-733-177			
Name	IMarks	EMarks	
Safwan	28	70	
Ahad	30	68	
Haseeb	27	66	
Ansar	29	72	
Omer	26	65	



### Program-13: Write a Node.js program to create DB and Collections in MongoDB

```
const { MongoClient } = require("mongodb");

const url = "mongodb://localhost:27017";
const dbName = "safwan";

MongoClient.connect(url)

.then(client => {

  console.log("1604-22-733-177 Database connected.");

  const db = client.db(dbName);

  return db.createCollection("students")

  .then(() => {

    console.log("Collection 'students' created.");

    client.close(); // close only after success

  })

  catch(err => {

    console.log("Collection already exists or error:", err.message);

    client.close();

  });

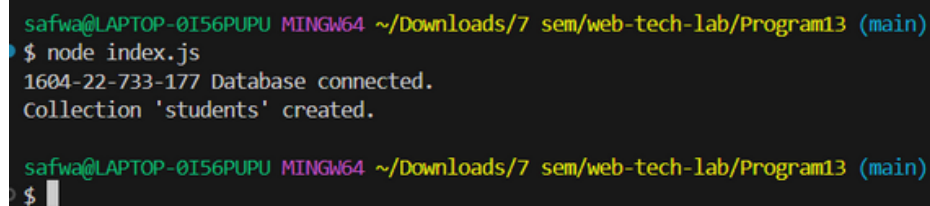
})

.catch(err => {

  console.error("Failed to connect to MongoDB:", err);

});
```

### Output:



```
safwa@LAPTOP-0I56PUPU MINGW64 ~/Downloads/7 sem/web-tech-lab/Program13 (main)
$ node index.js
1604-22-733-177 Database connected.
Collection 'students' created.

safwa@LAPTOP-0I56PUPU MINGW64 ~/Downloads/7 sem/web-tech-lab/Program13 (main)
$
```

## Program-14: Write a react.js program to retrieve data from MongoDB

### Index.js

```
import express from "express";
import cors from "cors";
import { MongoClient } from "mongodb";

const app = express();
app.use(cors());
app.use(express.json());

const url = "mongodb://localhost:27017/";
const dbName = "mymongodb";
let db;

async function connectDB() {
  try {
    const client = await MongoClient.connect(url);
    db = client.db(dbName);
    console.log("Connected to MongoDB");

    // Ensure collection exists and insert dummy data only if empty
    const collection = db.collection("student");
    const count = await collection.countDocuments();
    if (count === 0) {
      const dummyData = [
        { id: 101, name: "Safwan", marks: 92 },
        { id: 102, name: "Ahad", marks: 78 },
        { id: 103, name: "Nabeel", marks: 82 },
        { id: 104, name: "Kazim", marks: 90 },
      ];
      await collection.insertMany(dummyData);
      console.log("Dummy data inserted successfully");
    }
  } catch (err) {
    console.error("Failed to connect to MongoDB:", err);
  }
}

connectDB();

app.get("/retrieve", async (req, res) => {
  try {
    const result = await db.collection("student").find({}).toArray();
    res.json(result);
  } catch (err) {
```

```

    res.status(500).json({ error: "Failed to retrieve data" });
  }
});

app.listen(8080, () => {

  console.log("Server running on port 8080 from 1604-22-733-153");
});

```

### WebTech.js

```

import { Component } from "react";
class WebTech extends Component {
  state = { students: [] };

  componentDidMount() {
    fetch("http://localhost:8080/retrieve")
      .then(res => res.json())
      .then(data => this.setState({ students: data }));
  }

  render() {
    return (
      <div>
        <h2>Student Data from MongoDB</h2>

        <table border="1">
          <tr>
            <th>ID</th>
            <th>Name</th>
            <th>Marks</th>
          </tr>

          {this.state.students.map((s, i) => (
            <tr key={i}>
              <td>{s.id}</td>
              <td>{s.name}</td>
              <td>{s.marks}</td>
            </tr>
          ))}
        </table>
      </div>
    );
  }
}
export default WebTech;

```

## Output:

```
Node.js v24.8.0

safwa@LAPTOP-0I56PUPU MINGW64 ~/Downloads/7 sem/web-tech-lab/program14/restservice (main)
$ node index.js
REST Server running on port 8080 for 1604-22-733-177
Connected to MongoDB - Roll No: 1604-22-733-177, Name: Mohd Safwan Uddin
```

```
localhost:8080/retrieve
Pretty-print
[{"_id":"694e80f474fe1cd64f4f66f","id":101,"name":"Mohd Safwan Uddin","marks":92},{"_id":"694e80f474fe1cd64f4f670","id":102,"name":"Ahad","marks":78}, {"_id":"694e80f474fe1cd64f4f671","id":103,"name":"Nabeel","marks":82}, {"_id":"694e80f474fe1cd64f4f672","id":104,"name":"kazim","marks":90}]
```

```
localhost:3000
```

## Student Data from MongoDB - 1604-22-733-177

ID	Name	Marks
101	Mohd Safwan Uddin	92
102	Ahad	78
103	Nabeel	82
104	kazim	90