IP Practical

Q1. Design a web page having the concept of Typography which displays your name and boxes having different Color Modes.

```
<!DOCTYPE html>
<html>
<head>
<title>Typography & Color Modes</title>
<style>
body{
  font-family: Arial;
  text-align: center;
}
.box{
  width: 120px;
  height: 50px;
  margin: 10px auto;
  line-height: 50px;
  cursor: pointer;
  border-radius: 8px;
  background: lightgray;
}
</style>
</head>
<body>

<h1 style="font-size:45px; letter-spacing:3px;">John Doe</h1>
<p>Click the boxes to change page color</p>

<div class="box" onclick="document.body.style.background='white'">Light Mode</div>
<div class="box" onclick="document.body.style.background='#222';
document.body.style.color='white';">Dark Mode</div>
<div class="box" onclick="document.body.style.background='rgb(255,200,0)'">RGB
Mode</div>

</body>
</html>
```

Q2. Design a web page showing the concept Transitions and Transformations .

```
<!DOCTYPE html>
<html>
<head>
<title>Transitions & Transformations</title>
```

```
<style>
  body{
    font-family: Arial;
    text-align: center;
    padding-top: 40px;
  }

  .box{
    width: 120px;
    height: 120px;
    margin: 15px auto;
    background: blue;
    color: white;
    line-height: 120px;
    font-weight: bold;
    transition: 0.5s; /* Transition added for smooth effect */
  }

  /* Transformation styles */
  #scale:hover { transform: scale(1.3); background: red; }
  #rotate:hover { transform: rotate(45deg); background: green; }
  #skew:hover { transform: skew(20deg); background: purple; }

  /* Transition box */
  #transition:hover{
    background: orange;
    border-radius: 25px;
  }
</style>
</head>

<body>

<h2>Transition Example</h2>
<div id="transition" class="box">Transition</div>

<h2>Transformations</h2>
<div id="scale" class="box">Scale</div>
<div id="rotate" class="box">Rotate</div>
<div id="skew" class="box">Skew</div>

</body>
</html>
```

Q3. Design a web page showing the concept Transformations and Animations

```
<!DOCTYPE html>
<html>
<head>
<title>Transformations & Animations</title>
<style>
body {
   font-family: Arial, sans-serif;
   text-align: center;
   padding: 20px;
}

/* TRANSFORMATION BOXES */
.box {
   width: 100px;
   height: 100px;
   background: tomato;
   color: white;
   display: inline-flex;
   justify-content: center;
   align-items: center;
   margin: 15px;
   border-radius: 10px;
   transition: 0.5s;
   cursor: pointer;
}

/* Hover Transform Effects */
#scale:hover {
   transform: scale(1.3);
}
#rotate:hover {
   transform: rotate(150deg);
}
#skew:hover {
   transform: skewX(20deg);
}

/* ANIMATION BOXES */
.animBox {
   width: 120px;
   height: 50px;
   background: #007bff;
```

```
    color: white;
    display: inline-flex;
    justify-content: center;
    align-items: center;
    margin: 10px;
    border-radius: 8px;
    cursor: pointer;
}

/* Keyframes */
@keyframes move {
    from { transform: translateX(0); }
    to { transform: translateX(150px); }
}

@keyframes spin {
    from { transform: rotate(0deg); }
    to { transform: rotate(360deg); }
}

@keyframes bounce {
    0%,100% { transform: translateY(0); }
    50% { transform: translateY(-40px); }
}
</style>
</head>

<body>

<h2>TRANSFORMATIONS</h2>
<div id="scale" class="box">Scale Me</div>
<div id="rotate" class="box">Rotate Me</div>
<div id="skew" class="box">Skew Me</div>

<h2>ANIMATIONS</h2>
<div id="move" class="animBox">Move Me</div>
<div id="spin" class="animBox">Spin Me</div>
<div id="bounce" class="animBox">Bounce Me</div>

<script>
// Click animations
document.getElementById("move").onclick = function() {
    this.style.animation = "move 1s alternate 3";
};
```

```
document.getElementById("spin").onclick = function() {
   this.style.animation = "spin 1s linear 3";
};

document.getElementById("bounce").onclick = function() {
   this.style.animation = "bounce 1s ease 3";
};
</script>

</body>
</html>
```

Q4. Design a web page showing the concept Transitions, and Animations

```
<!DOCTYPE html>
<html>
<head>
<title>Transitions & Animations</title>
<style>
body{
  font-family: Arial, sans-serif;
  text-align: center;
  padding: 30px;
}

/* TRANSITION BOX */
.transitionBox{
  width: 120px;
  height: 120px;
  background: blue;
  color: white;
  display: inline-flex;
  justify-content: center;
  align-items: center;
  margin: 20px;
  border-radius: 10px;
  transition: 0.5s; /* Smooth effect */
}

/* Hover effect for Transition */
.transitionBox:hover{
  background: orange;
  border-radius: 30px;
```

```
  transform: scale(1.2);
}

/* ANIMATION BOXES */
.animBox{
  width: 120px;
  height: 50px;
  background: #007bff;
  color: white;
  display: inline-flex;
  justify-content: center;
  align-items: center;
  margin: 15px;
  border-radius: 8px;
  cursor: pointer;
}

/* Keyframes Animation */
@keyframes move{
  from { transform: translateX(0); }
  to { transform: translateX(150px); }
}

@keyframes spin{
  from { transform: rotate(0deg); }
  to { transform: rotate(360deg); }
}

@keyframes bounce{
  0%,100%{ transform: translateY(0); }
  50%{ transform: translateY(-40px); }
}
</style>
</head>

<body>

<h2>Transition Example</h2>
<div class="transitionBox">Hover Me</div>

<h2>Animations (Click Me)</h2>
<div id="move" class="animBox">Move</div>
<div id="spin" class="animBox">Spin</div>
<div id="bounce" class="animBox">Bounce</div>
```

```
<script>
document.getElementById("move").onclick = function(){
  this.style.animation = "move 1s alternate 3";
};

document.getElementById("spin").onclick = function(){
  this.style.animation = "spin 1s linear 3";
};

document.getElementById("bounce").onclick = function(){
  this.style.animation = "bounce 1s ease 3";
};
</script>

</body>
</html>
```

Q5. Develop a web page using the Bootstrap Grid system, Navbar, Forms & Button

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Bootstrap Practical</title>

<!-- Bootstrap CDN -->
<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">
</head>

<body>

<!-- NAVBAR -->
<nav class="navbar navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">My Website</a>
  </div>
</nav>

<!-- GRID LAYOUT -->
<div class="container mt-4">
  <div class="row text-center">
```

```
    <div class="col bg-primary text-white p-3 m-1">Box 1</div>
    <div class="col bg-success text-white p-3 m-1">Box 2</div>
    <div class="col bg-danger text-white p-3 m-1">Box 3</div>
  </div>
</div>

<!-- FORM -->
<div class="container mt-4">
  <h3>Contact Form</h3>
  <form>
    <div class="mb-3">
      <label class="form-label">Name</label>
      <input type="text" class="form-control" placeholder="Enter your name">
    </div>

    <div class="mb-3">
      <label class="form-label">Email</label>
      <input type="email" class="form-control" placeholder="Enter email">
    </div>

    <!-- BUTTONS -->
    <button type="submit" class="btn btn-primary">Submit</button>
    <button type="reset" class="btn btn-secondary">Clear</button>
  </form>
</div>

</body>
</html>
```

Q6. Write a program in JavaScript arrow function to find Factorial of a number on button click

```
<!DOCTYPE html>
<html>
<head>
<title>Factorial Using Arrow Function</title>
<script>
// Arrow function to calculate factorial
const findFactorial = () => {
    let num = parseInt(document.getElementById("num").value);
    let fact = 1;

    for(let i = 1; i <= num; i++){
```

```
        fact *= i;
    }

    document.getElementById("result").innerHTML = "Factorial: " + fact;
}
</script>
</head>

<body style="text-align:center; font-family:Arial;">

<h2>Factorial Finder</h2>

<input type="number" id="num" placeholder="Enter number">
<br><br>

<button onclick="findFactorial()">Find Factorial</button>
<br><br>

<h3 id="result"></h3>

</body>
</html>
```

Q7. Write a program in JavaScript arrow function to generate Fibonacci series on button click

```
<!DOCTYPE html>
<html>
<head>
<title>Fibonacci using Arrow Function</title>

<script>
// Arrow Function to generate Fibonacci Series
const fibonacci = () => {
    let n = parseInt(document.getElementById("num").value);
    let a = 0, b = 1, result = "0 ";

    for(let i = 1; i < n; i++){
        result += b + " ";
        let temp = a + b;
        a = b;
        b = temp;
    }
```

```
   document.getElementById("output").innerHTML = result;
}
</script>

</head>

<body style="text-align:center; font-family:Arial;">

<h2>Fibonacci Series</h2>

<input type="number" id="num" placeholder="Enter limit">
<br><br>

<button onclick="fibonacci()">Generate</button>
<br><br>

<h3 id="output"></h3>

</body>
</html>
```

Q8. Write a program in JavaScript arrow function to find cube of a number on button click

```
<!DOCTYPE html>
<html>
<head>
<title>Cube Using Arrow Function</title>

<script>
// Arrow Function to find cube
const findCube = () => {
   let num = document.getElementById("num").value;
   let cube = num * num * num;
   document.getElementById("result").innerHTML = "Cube: " + cube;
}
</script>

</head>

<body style="text-align:center; font-family:Arial;">

<h2>Find Cube</h2>
```

```
<input type="number" id="num" placeholder="Enter number">
<br><br>

<button onclick="findCube()">Find Cube</button>
<br><br>

<h3 id="result"></h3>

</body>
</html>
```

Q9. Write a program in JavaScript (JS) to implement Iterators and Generators

```
<!DOCTYPE html>
<html>
<head>
<title>Iterators & Generators</title>

<script>
// ☑ Simple Iterator
function myIterator(arr) {
   let i = 0;
   return {
      next: () => ({
         value: arr[i++],
         done: i > arr.length
      })
   };
}

// ☑ Simple Generator
function* numberGenerator(){
   yield 1;
   yield 2;
   yield 3;
}

function showOutput(){
   // Iterator
   let nums = [10, 20, 30];
   let it = myIterator(nums); // ☑ corrected
```

```
    let text = "Iterator Output: ";
    text += it.next().value + " ";
    text += it.next().value + " ";
    text += it.next().value + "<br>";

    // Generator
    let gen = numberGenerator();
    text += "Generator Output: ";
    text += gen.next().value + " ";
    text += gen.next().value + " ";
    text += gen.next().value;

    document.getElementById("result").innerHTML = text;
}

</script>
</head>

<body style="text-align:center; font-family:Arial;">
<h2>Iterators and Generators</h2>
<button onclick="showOutput()">Run Program</button>
<p id="result"></p>
</body>
</html>
```

Q10. Write a Javascript Program to show DOM Manipulation and CSS Manipulations

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM & CSS Manipulation</title>
    <style>
        #text {
            font-size: 20px;
            color: blue;
            margin-bottom: 10px;
        }
        #box {
            width: 150px;
            height: 150px;
            background-color: lightgreen;
            margin-top: 10px;
```

```
        }
    </style>
</head>
<body>

<h2>DOM & CSS Manipulation Example</h2>

<p id="text">Original Text</p>

<button onclick="changeText()">Change Text</button>
<button onclick="changeColor()">Change Color</button>
<button onclick="addElement()">Add New Element</button>

<div id="box"></div>

<script>
    // DOM Manipulation: Changing the text of an element
    function changeText() {
        document.getElementById("text").innerHTML = "Text Changed Using DOM!";
    }

    // CSS Manipulation: Changing style properties using JavaScript
    function changeColor() {
        document.getElementById("box").style.backgroundColor = "orange";
    }

    // DOM Manipulation: Adding a new element dynamically
    function addElement() {
        let newPara = document.createElement("p");
        newPara.innerHTML = "New Element Added!";
        document.body.appendChild(newPara);
    }
</script>

</body>
</html>
```

Q11. Design a student information form and validate using JavaScript. Form should contain following information Roll no., name, email id, mobile number, age, hobbies

```
<!DOCTYPE html>
<html>
<head>
<title>Student Form Validation</title>

<style>
body{
   font-family: Arial;
   text-align: center;
}

form{
   border: 1px solid black;
   width: 250px;
   padding: 10px;
   margin: auto;
}

input, select{
   width: 90%;
   margin: 5px;
}
button{
   margin-top: 10px;
}
</style>

<script>
function validateForm(){
   let roll = document.getElementById("roll").value;
   let name = document.getElementById("name").value;
   let email = document.getElementById("email").value;
   let mobile = document.getElementById("mobile").value;
   let age = document.getElementById("age").value;
   let hobby = document.getElementById("hobby").value;

   if(roll == "" || name == "" || email == "" || mobile == "" || age == "" || hobby == ""){
      alert("All fields are required!");
      return false;
```

```
    }
    alert("Form Submitted Successfully!");
    return true;
}
</script>

</head>

<body>

<h3>Student Information Form</h3>

<form onsubmit="return validateForm()">
    <input type="text" id="roll" placeholder="Roll No"><br>
    <input type="text" id="name" placeholder="Name"><br>
    <input type="text" id="email" placeholder="Email"><br>
    <input type="text" id="mobile" placeholder="Mobile"><br>
    <input type="text" id="age" placeholder="Age"><br>

    <select id="hobby">
        <option value="">Select Hobby</option>
        <option>Reading</option>
        <option>Sports</option>
        <option>Music</option>
    </select><br>

    <button type="submit">Submit</button>
</form>

</body>
</html>
```

Q12. Design an Employee information form and validate using JavaScript. Form should contain following information Enroll no., name, email id, mobile number, department, salary

```
<!DOCTYPE html>
<html>
<head>
<title>Employee Form Validation</title>

<style>
```

```
body{
   font-family: Arial;
   text-align: center;
}

form{
   border: 1px solid black;
   width: 260px;
   padding: 10px;
   margin: auto;
}

input, select{
   width: 90%;
   margin: 6px;
}
button{
   margin-top: 8px;
}
</style>

<script>
function validateForm(){
   let enroll = document.getElementById("enroll").value;
   let name = document.getElementById("name").value;
   let email = document.getElementById("email").value;
   let mobile = document.getElementById("mobile").value;
   let dept = document.getElementById("dept").value;
   let salary = document.getElementById("salary").value;

   if(enroll == "" || name == "" || email == "" || mobile == "" || dept == "" || salary == ""){
      alert("All fields are required!");
      return false;
   }

   if(!email.includes("@")){
      alert("Enter valid Email!");
      return false;
   }

   if(mobile.length != 10){
      alert("Mobile must be 10 digits!");
      return false;
   }
```

```
    if(isNaN(salary) || salary <= 0){
        alert("Enter valid Salary!");
        return false;
    }

    alert("Form Submitted Successfully!");
    return true;
}
</script>

</head>

<body>

<h3>Employee Information Form</h3>

<form onsubmit="return validateForm()">
    <input type="text" id="enroll" placeholder="Enroll No"><br>
    <input type="text" id="name" placeholder="Name"><br>
    <input type="text" id="email" placeholder="Email"><br>
    <input type="text" id="mobile" placeholder="Mobile"><br>

    <select id="dept">
        <option value="">Select Department</option>
        <option>HR</option>
        <option>IT</option>
        <option>Sales</option>
        <option>Finance</option>
    </select><br>

    <input type="text" id="salary" placeholder="Salary"><br>

    <button type="submit">Submit</button>
</form>

</body>
</html>
```

Q13. Write a program to demonstrate Breadcrumbs and Jumbotron in the Bootstrap framework.

Page 1 — home.html

```
<!DOCTYPE html>
<html>
<head>
<title>Home - Bootstrap</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>

<body class="text-center mt-5">
<h2>Home Page</h2>
<p><a href="employees.html" class="btn btn-primary">Go to Employees</a></p>
</body>
</html>
```

☑ **Page 2 — employees.html**

```
<!DOCTYPE html>
<html>
<head>
<title>Employees - Bootstrap</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>

<body class="text-center mt-5">
<h2>Employees Page</h2>
<p><a href="details.html" class="btn btn-success">View Employee Details</a></p>
<p><a href="home.html" class="btn btn-secondary">Back to Home</a></p>
</body>
</html>
```

☑ **Page 3 — details.html**

✨ This page contains your **Breadcrumb + Jumbotron** (Main Practical)

```
<!DOCTYPE html>
<html>
<head>
<title>Details - Bootstrap</title>
```

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

<body>

<div class="container mt-4">

  <!-- Breadcrumb -->
  <nav aria-label="breadcrumb">
    <ol class="breadcrumb">
      <li class="breadcrumb-item"><a href="home.html">Home</a></li>
      <li class="breadcrumb-item"><a href="employees.html">Employees</a></li>
      <li class="breadcrumb-item active" aria-current="page">Details</li>
    </ol>
  </nav>

  <!-- Jumbotron -->
  <div class="jumbotron">
    <h2>Employee Details</h2>
    <p>Welcome! This is a Jumbotron Section showing employee information.</p>
    <a href="employees.html" class="btn btn-primary">Back</a>
  </div>

</div>

</body>
</html>
```

Q14. Write a program to demonstrate simple Bootstrap styling of an HTML table

```
<!DOCTYPE html>
<html>
<head>
<title>Bootstrap Table</title>

<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

</head>
<body class="p-4">

<h3 class="text-center mb-4">Student Information Table</h3>
```

```
<table class="table table-bordered table-striped table-hover">
  <thead class="thead-dark">
    <tr>
      <th>Roll No.</th>
      <th>Name</th>
      <th>Course</th>
    </tr>
  </thead>

  <tbody>
    <tr>
      <td>01</td>
      <td>John</td>
      <td>IT</td>
    </tr>

    <tr>
      <td>02</td>
      <td>Neha</td>
      <td>Computer</td>
    </tr>

    <tr>
      <td>03</td>
      <td>Raj</td>
      <td>AI & DS</td>
    </tr>
  </tbody>

</table>

</body>
</html>
```

Q15. Write a program to demonstrate Hierarchical inheritance

```
// Parent class
class Animal {
    void eat() {
        System.out.println("Animals eat food.");
    }
}
```

```
// Child class 1
class Dog extends Animal {
   void bark() {
      System.out.println("Dog barks.");
   }
}

// Child class 2
class Cat extends Animal {
   void meow() {
      System.out.println("Cat meows.");
   }
}

public class HierarchicalInheritance {
   public static void main(String[] args) {
      Dog d = new Dog();
      d.eat();   // from Animal class
      d.bark();  // from Dog class

      System.out.println();

      Cat c = new Cat();
      c.eat();   // from Animal class
      c.meow();  // from Cat class
   }
}
```

Q16. Write a program to demonstrate Multilevel inheritance

```
// Parent class
class Animal {
   void eat() {
      System.out.println("Animals eat food.");
   }
}

// Child class extending Animal
class Dog extends Animal {
   void bark() {
      System.out.println("Dog barks.");
   }
}
```

```
// Grandchild class extending Dog
class Puppy extends Dog {
   void weep() {
      System.out.println("Puppy is weeping.");
   }
}

public class MultilevelInheritance {
   public static void main(String[] args) {
      Puppy p = new Puppy();
      p.eat();   // from Animal class
      p.bark();  // from Dog class
      p.weep();  // from Puppy class
   }
}
```

Q17. Write a program to implement the concept of React Hooks

```
import React, { useState, useEffect } from "react";

function App() {
  const [count, setCount] = useState(0);

  useEffect(() => {
    console.log("Count changed:", count);
  }, [count]);

  return (
    <div style={{ textAlign: "center", marginTop: "50px" }}>
      <h2>React Hooks Example</h2>
      <p>Count: {count}</p>
      <button onClick={() => setCount(count + 1)}>
        Increase Count
      </button>
    </div>
  );
}

export default App;
```

Steps to Run

1. Install Node.js (if not installed) → download from nodejs.org
2. Open Command Prompt
3. Run:

```
npx create-react-app my-hooks-app
```

4. Go inside folder:

```
cd my-hooks-app
```

5. Start project:

```
npm start
```

6. Open file:
➡️ **my-hooks-app/src/App.js**
Delete all and paste React Hooks code.

Q18. Write a react program to implement the concept of Asynchronous programming using promises

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

function App() {
  const [message, setMessage] = useState("Click the button");

  const showMessage = () => {
    setMessage("Loading...");

    // Promise Example
    new Promise((resolve) => {
      setTimeout(() => {
        resolve("Data Loaded Successfully ☑️");
      }, 2000);
    }).then((result) => {
      setMessage(result);
    });
  };

  return (
    <div style={{ textAlign: "center", marginTop: "40px" }}>
      <h2>{message}</h2>
```

```
    <button onClick={showMessage}>Fetch Data</button>
   </div>
  );
}

export default App;
```

Q19. Write a program in Node JS to Create a file, Read the data from file

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

// Create & Write Data to File
fs.writeFile('example.txt', 'Hello! This is node file 😊', (err) => {
  if (err) {
    console.log("Error writing file!");
  } else {
    console.log("File created and data written!");

    // Read File Data
    fs.readFile('example.txt', 'utf8', (err, data) => {
      if (err) {
        console.log("Error reading file!");
      } else {
        console.log("File Data:");
        console.log(data);
      }
    });
  }
});
```

How to Run a Node.js Program (in vs code)

1. **Check if Node.js is installed**
Open Command Prompt and type:

```
node -v
```

If version appears → Node.js is installed ☑

---

2. **Open VS Code**

3. **Create a new folder**
Example: NodePractice

4. **Open the folder in VS Code**
File → Open Folder → Select NodePractice

5. **Create a new file**
Name it: index.js

6. **Write your Node.js code**
Example:

```
console.log("Hello Node!");
```

7. **Open Terminal in VS Code**
Press:

Ctrl + `

8. **Run the program**
In terminal type:

```
node index.js
```

☑ Output will show:

```
Hello Node!
```

Q20. Write a program in Node JS to Read the data from file, Write the data to a file

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

// ☑ Write data to file
fs.writeFileSync("example.txt", "Hello Student! Welcome to Node.js");

// ☑ Read data from file
const data = fs.readFileSync("example.txt", "utf8");

// ☑ Show output in console
console.log("File Content: " + data);
```

Q21. Write a program in Node JS to Create a file, Rename a file.

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

// ☑ Create a file
fs.writeFileSync("oldname.txt", "This is a test file.");

// ☑ Rename the file
fs.renameSync("oldname.txt", "newname.txt");

console.log("File created and renamed successfully!");
```

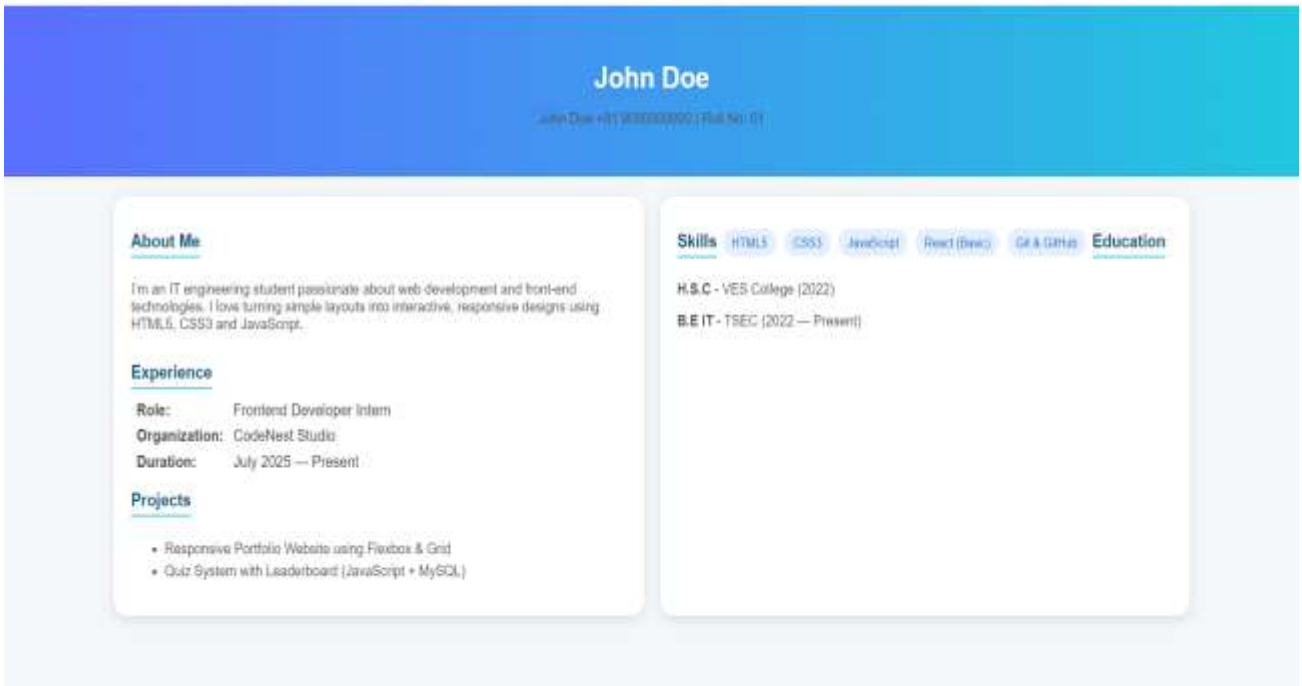
Q22. Design your resume page using HTML elements and apply CSS styling.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>John Doe Resume</title>

<style>
  body {
    font-family: Arial, sans-serif;
    margin: 0;
    background:#f4f7fa;
  }

  /* Header */
  .header {
    background: linear-gradient(90deg, #5b6efe, #21c7de);
    color: #fff;
    text-align: center;
    padding: 35px 10px;
  }
  .header h1 {
    font-size: 30px;
    margin-bottom: 5px;
  }

  /* Content Boxes */
  .container {
    width:85%;
```

```
        margin: 20px auto;
        display:flex;
        gap:20px;
        justify-content:center;
        flex-wrap:wrap;
    }
    .card {
        background: #fff;
        width: 45%;
        min-width: 300px;
        padding: 20px;
        border-radius:15px;
        box-shadow:0px 4px 12px rgba(0,0,0,0.1);
    }

    .card h2 {
        font-size: 18px;
        color:#0a567a;
        margin-bottom:8px;
        border-bottom:2px solid #21c7de;
        display: inline-block;
        padding-bottom:4px;
    }

    ul li { margin-bottom:5px; }
    table td { padding: 3px 5px; color:#444; }
    p, li { color:#555; font-size:15px; }

    /* Skill Tags */
    .skill-tag{
        display:inline-block;
        background:#e4efff;
        color:#0066cc;
        padding:6px 10px;
        margin: 3px;
        border-radius:12px;
        font-size:13px;
    }
</style>
</head>

<body>

    <!-- Resume Header -->
```

```html
<div class="header">
  <h1>John Doe</h1>
  <p>John Doe +91 9000000000 | Roll No: 01</p>
</div>

<!-- Resume Content -->
<div class="container">

  <!-- Left Box -->
  <div class="card">
    <h2>About Me</h2>
    <p>I’m an IT engineering student passionate about web development and front-end technologies. I love turning simple layouts into interactive, responsive designs using HTML5, CSS3 and JavaScript.</p>

    <h2>Experience</h2>
    <table>
      <tr><td><b>Role:</b></td><td>Frontend Developer Intern</td></tr>
      <tr><td><b>Organization:</b></td><td>CodeNest Studio</td></tr>
      <tr><td><b>Duration:</b></td><td>July 2025 — Present</td></tr>
    </table>

    <h2>Projects</h2>
    <ul>
      <li>Responsive Portfolio Website using Flexbox & Grid</li>
      <li>Quiz System with Leaderboard (JavaScript + MySQL)</li>
    </ul>
  </div>

  <!-- Right Box -->
  <div class="card">
    <h2>Skills</h2>
    <span class="skill-tag">HTML5</span>
    <span class="skill-tag">CSS3</span>
    <span class="skill-tag">JavaScript</span>
    <span class="skill-tag">React (Basic)</span>
    <span class="skill-tag">Git & GitHub</span>

    <h2>Education</h2>
    <p><b>H.S.C</b> - VES College (2022)</p>
    <p><b>B.E IT</b> - TSEC (2022 — Present)</p>
  </div>

</div>
```

```
</body>
</html>
```



Q23. Design your marksheet page using HTML elements and apply CSS styling.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>10th Marksheet - John Doe</title>
<style>
  body {
    font-family: Arial, sans-serif;
    background: #eee;
    padding: 30px;
  }
  .marksheet {
    max-width: 700px;
    margin: auto;
    background: #fff;
    padding: 25px;
    border: 2px solid #333;
  }
  h1, h2 {
    text-align: center;
```

```
    margin: 5px 0;
  }
  table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 20px;
  }
  th, td {
    border: 1px solid #555;
    padding: 10px;
    text-align: center;
  }
  th {
    background: #ddd;
  }
  .footer {
    margin-top: 40px;
    display: flex;
    justify-content: space-between;
    font-weight: bold;
  }
</style>
</head>

<body>

<div class="marksheet">
  <h1>NY ENGLISH SCHOOL</h1>
  <h2>10th Grade Marksheet</h2>

  <p><strong>Student Name:</strong> John Doe</p>
  <p><strong>Roll Number:</strong> 123456</p>
  <p><strong>Year:</strong> 2024 - 2025</p>

  <table>
    <tr>
      <th>Subject</th>
      <th>Max Marks</th>
      <th>Marks Obtained</th>
    </tr>
    <tr>
      <td>Mathematics</td>
      <td>100</td>
      <td>92</td>
```

```
    </tr>
    <tr>
      <td>Science</td>
      <td>100</td>
      <td>88</td>
    </tr>
    <tr>
      <td>English</td>
      <td>100</td>
      <td>85</td>
    </tr>
    <tr>
      <td>Social Studies</td>
      <td>100</td>
      <td>90</td>
    </tr>
    <tr>
      <td>Hindi</td>
      <td>100</td>
      <td>80</td>
    </tr>

    <tr style="font-weight:bold; background:#f3f3f3;">
      <td>Total</td>
      <td>500</td>
      <td>435</td>
    </tr>
  </table>

  <div class="footer">
    <p>Principal</p>
    <p>Class Teacher</p>
  </div>

</div>

</body>
</html>
```

NY ENGLISH SCHOOL

10th Grade Marksheet

Student Name: John Doe

Roll Number: 123456

Year: 2024 - 2025

Subject	Max Marks	Marks Obtained
Mathematics	100	92
Science	100	88
English	100	85
Social Studies	100	90
Hindi	100	80
Total	**500**	**435**

Principal **Class Teacher**