**ASSIGNMENT-4**

1. Create a function checkAge that takes a person's age as an argument and checks if they are eligible to vote. If the age is 18 or above, resolve the promise after 2 seconds; if below 18, reject the promise. Use async/await and try/catch to call the function and display a message indicating whether the person is eligible to vote

Ans: <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

</body>

<script>

    function checkAge(age){

        try{

            return new Promise((resolve,reject)=>{

                setTimeout(()=>{

                    if(age>=18){

                        resolve("Eligible to vote");

                    }

                    else{

                        reject("Not eligible to vote")

                    }

                },2000)

            })

        }

        catch(err){

            console.log(err);

        }

    }

    async function check(age) {

        try{

            let res=await checkAge(age);

            console.log(res);

        }

        catch(err){

            console.log(err);

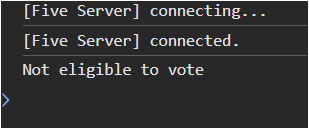
        }

    }

    check(8)

</script>

</html>



2. Write an asynchronous function that will generate a random integer number between 1 and 100 after 3 seconds. Write another function that will check whether the number is prime or not by using a promise. If the number is prime the promise will resolve and reject otherwise. Use async/await and try/catch to call both the functions and display the generated number and also display if it is even or odd.

Ans: <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

</body>

<script>

    function checkPrime(num){

        try{

            return new Promise((resolve,reject)=>{

                if(num==1){

                    resolve(`Neither prime nor composite`)

                }

                for(let i=2;i<Math.ceil(num/2);i++){

                    if(num%i==0){

                        reject(`${num} is not a prime number`)

                    }

                }

                resolve(`${num} is a prime number`)

            })

        }

        catch(err){

            console.log(err);

        }

    }

    async function generateRandom() {

        try{

            let num=parseInt(Math.random()\*100);

            console.log(`The number is ${num}`);

            let res=await checkPrime(num);

            console.log(res);

        }

        catch(err){

            console.log(err);

        }

    }

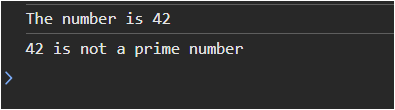
setTimeout(()=>{

    generateRandom()

},3000)

</script>

</html>



3. Fetch products from https://fakestoreapi.com/products using axios. In the HTML page display the product image, name and price in product card format. Use async/await for asynchronous data fetching and try/catch to handle possible errors.

Ans: <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<style>

#products{

display: grid;

grid-template-columns: repeat(4,1fr);

gap:1rem;

}

img{

width:16rem;

height: 20rem;

padding:1rem ;

}

.card{

border: 1px solid black;

justify-items: center;

}

p,h4{

text-align: center;

}

</style>

</head>

<body>

<div id="products">

</div>

</body>

<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>

<script>

let url=`https://fakestoreapi.com/products`;

async function getProducts() {

try{

let res=await axios.get(url);

let products=res.data

displayProducts(products)

}

catch(err){

console.log(err);

}

}

function displayProducts(products){

let card=document.getElementById("products")

products.forEach(product => {

let prod=document.createElement("div");

prod.className="card"

let image=document.createElement("img");

image.src=product.image;

let name=document.createElement("h4");

name.innerText=product.title;

let price=document.createElement("p");

price.innerHTML=`<p>Price: Rs${product.price}</p>`;

prod.appendChild(image)

prod.appendChild(name)

prod.appendChild(price)

card.appendChild(prod)

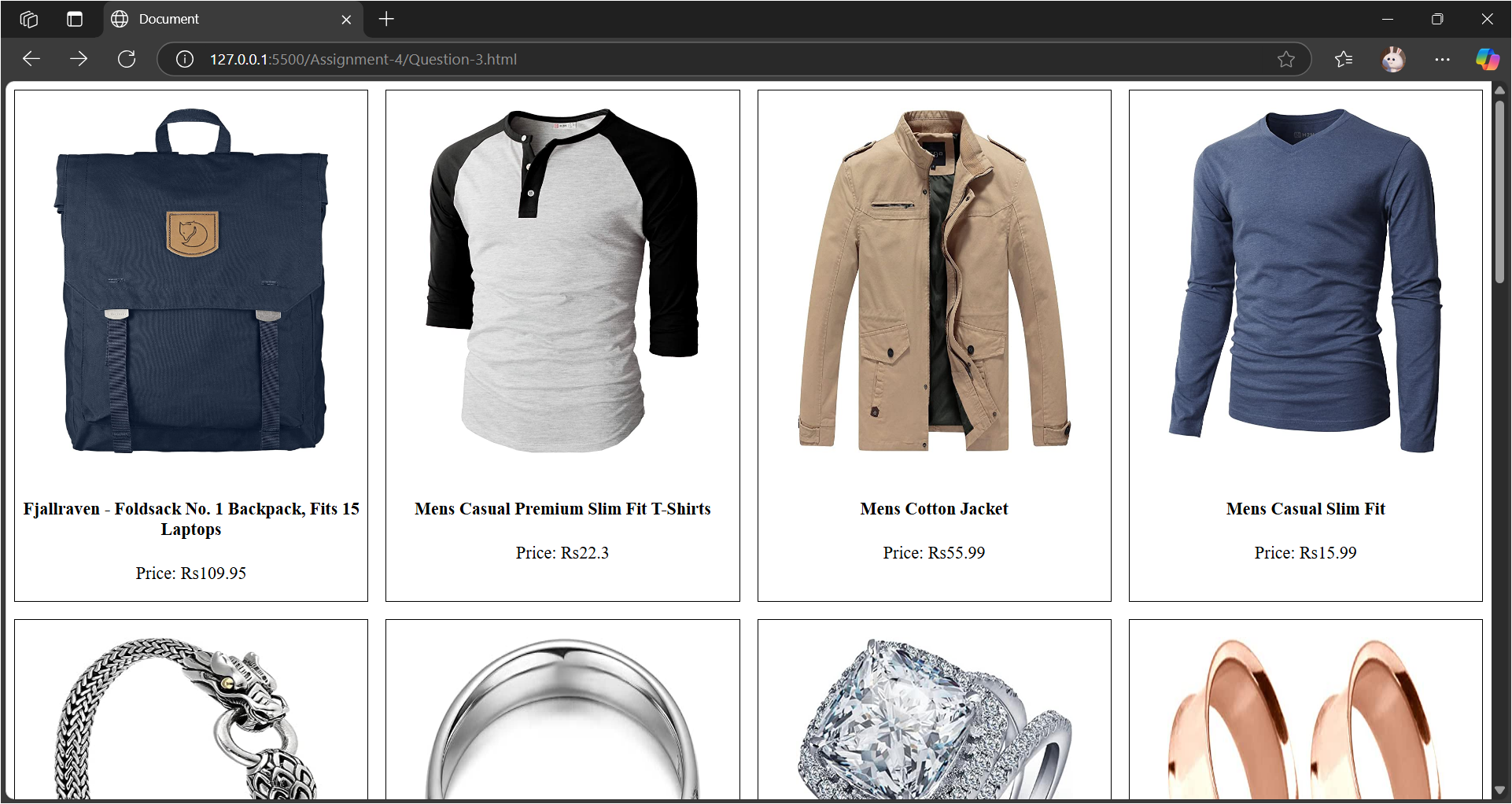
});

}

getProducts()

</script>

</html>



4. Create a JSON server with 3 student records. Each student has an id, name, branch, and CGPA. Run the server and perform the following operations using vanilla JS, and AXIOS. Use async/await and try/catch whenever required.

a. Create an HTML form to add a new student to the JSON server file.

b. Display all the available student records in an HTML file in a tabular format

c. Create two buttons Update and Delete with each row to perform an update and delete operation.

d. On the update page populate the form with the existing data (name, branch, and CGPA) and update the student with newly given details.

e. Create a search field to search students based on their names.

f. Create easy navigation between these pages and style the pages with CSS.

Ans:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        input{

            margin: 10px;

            width: 90%;

            height:30px;

        }

        form{

            border: 2px solid black;

            margin-left: 35%;

            margin-right: 35%;

            margin-top: 12%;

            display: flex;

            flex-direction: column;

        }

    </style>

</head>

<body>

    <form action="" method="post" onsubmit="addStudent(event)">

        <input type="name" id="name" placeholder="Enter your name">

        <input type="branch" id="branch" placeholder="Enter your branch">

        <input type="cgpa" id="cgpa" placeholder="Enter your CGPA">

        <input type="submit" value="Add Student">

    </form>

    <p id="msg"></p>

<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>

<script>

    let url = "http://localhost:3000/students";

    let msg=document.getElementById("msg");

    async function addStudent(e){

        e.preventDefault()

        try{

            let name=document.getElementById("name").value;

            let branch=document.getElementById("branch").value;

            let cgpa=document.getElementById("cgpa").value;

            let response = await axios.post(url, {"name": name, "branch": branch, "cgpa":cgpa})

            if(response.status==201){

                msg.innerHTML="Student data added"

                document.getElementById("name").value = ""

                document.getElementById("branch").value = ""

                document.getElementById("cgpa").value = ""

            }

        }catch(err){

            console.log(err);

            msg.innerHTML="Student data not added"

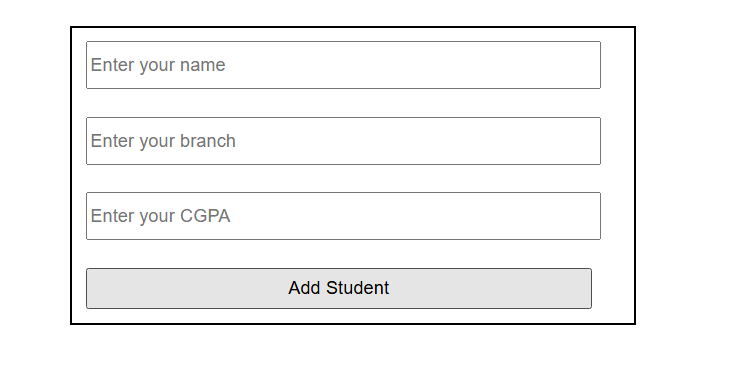
        }

    }

</script>

</body>

</html>



<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        .student{

            padding: 10px;

        }

        #search{

            width: 30%;

            margin-left: 30%;

            height: 30px;

        }

        #students{

            margin-top: 5px;

            display: grid;

            grid-template-columns: repeat(4,1fr);

            gap:10px;

        }

        .student{

            border: 1px solid black;

        }

        a{

            padding-left: 10px;

            padding-right: 10px;

        }

    </style>

</head>

<body>

    <input type="text" id="search" placeholder="Search by name" oninput="filterStudents()">

    <div id="students"></div>

<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>

<script>

    let url = "http://localhost:3000/students";

    let students=document.getElementById("students");

    let studentData=[];

    async function displayStudents(){

        try{

            let response = await axios.get(url)

            studentData=response.data;

            let students=document.getElementById("students")

            if(response.status===200){

                for(let i=0;i<studentData.length;i++){

                    let student=document.createElement("div")

                    let id=document.createElement("p")

                    id.innerHTML=`Student id: ${studentData[i].id}`

                    let name=document.createElement("p")

                    name.innerHTML=`Student name: ${studentData[i].name}`

                    let branch=document.createElement("p")

                    branch.innerHTML=`Student branch: ${studentData[i].branch}`

                    let cgpa=document.createElement("p")

                    cgpa.innerHTML=`Student cgpa: ${studentData[i].cgpa}`

                    let updateStu=document.createElement("a");

                    updateStu.innerHTML="Update";

                    updateStu.href=`Question-4cUpdate.html?id=${studentData[i].id}`;

                    let deleteStu=document.createElement("a");

                    deleteStu.innerHTML="Delete";

                    deleteStu.href=`Question-4cDelete.html?id=${studentData[i].id}`;

                    student.appendChild(id)

                    student.appendChild(name)

                    student.appendChild(branch)

                    student.appendChild(cgpa)

                    student.appendChild(updateStu)

                    student.appendChild(deleteStu)

                    student.className="student"

                    students.appendChild(student)

                }

            }

        }catch(err){

            console.log(err);

        }

    }

    function renderStudents(data) {

            students.innerHTML = "";

            data.forEach(student => {

                let studentDiv = document.createElement("div");

                studentDiv.innerHTML = `

                    <p>Student id: ${student.id}</p>

                    <p>Student name: ${student.name}</p>

                    <p>Student branch: ${student.branch}</p>

                    <p>Student cgpa: ${student.cgpa}</p>

                    <a href="Question-4cUpdate.html?id=${student.id}">Update</a>

                    <a href="Question-4cDelete.html?id=${student.id}">Delete</a>

                `;

                students.appendChild(studentDiv);

            });

        }

    function filterStudents() {

        let query = document.getElementById("search").value.toLowerCase();

        let filteredData = studentData.filter(student =>

            student.name.toLowerCase().includes(query)

        );

        renderStudents(filteredData);

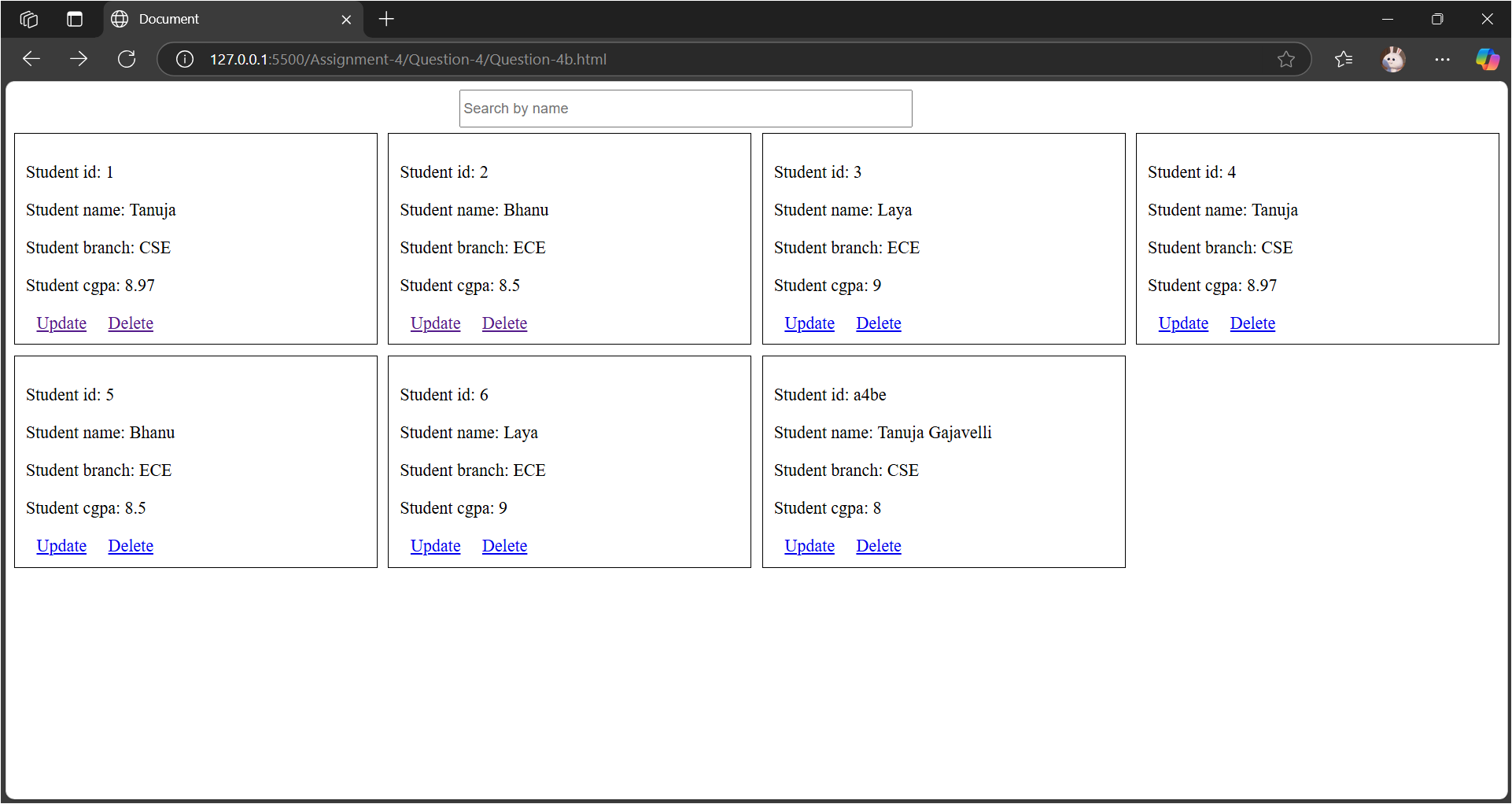
    }

    displayStudents()

</script>

</body>

</html>



<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        input{

            margin: 10px;

            width: 90%;

            height:30px;

        }

        form{

            border: 2px solid black;

            margin-left: 35%;

            margin-right: 35%;

            margin-top: 12%;

            display: flex;

            flex-direction: column;

        }

    </style>

</head>

<body>

    <form method="post" onsubmit="updateStu(event)">

        <input type="text" id="sid" placeholder="Enter your id">

        <input type="text" id="name" placeholder="Enter your name">

        <input type="text" id="branch" placeholder="Enter your branch">

        <input type="text" id="cgpa" placeholder="Enter your cgpa">

        <input type="submit" value="Update">

        <p id="msg"></p>

    </form>

    <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>

    <script>

        let urlParams=new URLSearchParams(window.location.search)

        let id=urlParams.get("id");

        let msg=document.getElementById("msg");

        let url=`http://localhost:3000/students/${id}`

        async function updateStu(e){

            e.preventDefault()

            try {

                let sid = document.getElementById("sid").value

                let name = document.getElementById("name").value

                let branch = document.getElementById("branch").value

                let cgpa = document.getElementById("cgpa").value

                let response = await axios.put(url, {"id": sid, "name": name, "branch": branch, "cgpa": cgpa})

                console.log(response);

                if(response.status === 201){

                    document.getElementById("sid").value = ""

                    document.getElementById("name").value = ""

                    document.getElementById("branch").value = ""

                    document.getElementById("cgpa").value = ""

                    msg.innerHTML = "Student Updated"

                }

            } catch (error) {

                console.log(error);

                msg.innerHTML = "Student Not Updated"

            }

        }

        async function getDetails(){

            try {

                let response = await axios.get(url)

                let student = response.data

                document.getElementById("sid").value = student.id

                document.getElementById("name").value = student.name

                document.getElementById("branch").value = student.branch

                document.getElementById("cgpa").value = student.cgpa

            } catch (error) {

                alert("Invalid Id")

                window.location="Question-4b.html"

            }

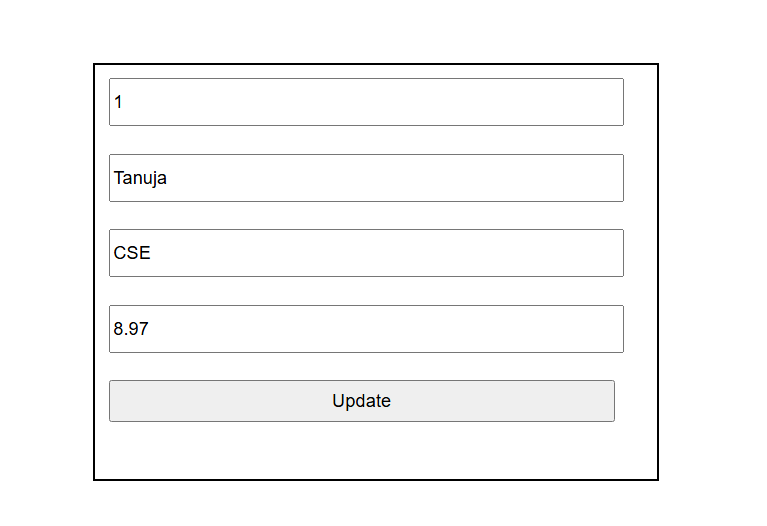
        }

        getDetails();

    </script>

</body>

</html>



<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>

    <script>

        let urlParams = new URLSearchParams(window.location.search)

        let id = urlParams.get("id");

        async function deleteStu(id){

           try {

                let url = "http://localhost:3000/students/"+id

                let response = await axios.delete(url)

                console.log(response);

                if(response.status === 200){

                    alert("Student Deleted")

                    window.location = "Question-4b.html"

                }

           } catch (error) {

            console.log(error);

            alert("Student data not Deleted. Check Id.")

            window.location = "Question-4b.html"

           }

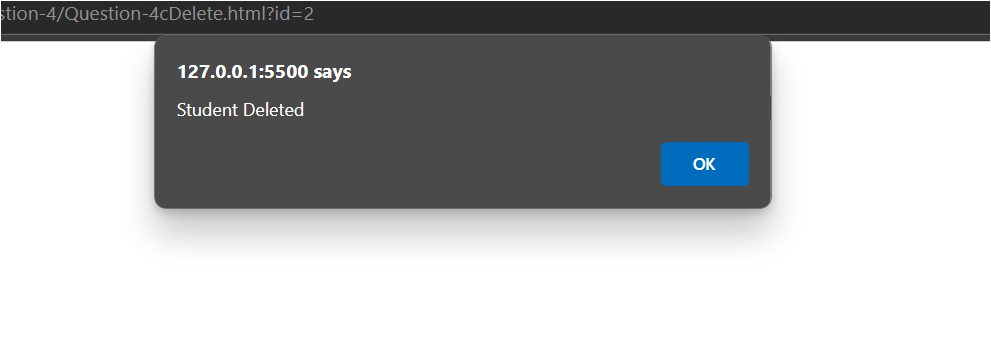
        }

        deleteStu(id)

    </script>

</body>

</html>



After Search Option:

