**1. Create a function *division* to take two numbers as arguments and perform the division operation. The function should return the response after 3 seconds. If the second number is zero, create a promise that will reject the operation; resolve otherwise. Call the function with possible parameters. Use async/await and try/catch to display possible outcomes.**

Ans-// 21BCSE30 | ANIKET PATRA

async function division(num1, num2) {

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

setTimeout(() => {

if (num2 === 0) {

reject("Division by zero is not allowed");

} else {

resolve(num1 / num2);

}

}, 3000);

});

}

(async function performDivision() {

try {

const result = await division(10, 2);

console.log("Result of division:", result);

} catch (error) {

console.error("Error during division:", error);

}

})();

(async function performDivisionWithZero() {

try {

const result = await division(10, 0);

console.log("Result of division:", result);

} catch (error) {

console.error("Error during division:", error);

}})();

**Output:-**

PS C:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-4\ass-4> node "c:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-4\ass-4\q1.js"

Result of division: 5

Error during division: Division by zero is not allowed

**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 even or odd by using a promise. If the number is even 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-

// 21BCSE30 | ANIKET PATRA

function generateRandomNumber() {

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

setTimeout(() => {

const randomNumber = Math.floor(Math.random() \* 100) + 1;

resolve(randomNumber);

}, 3000);

});

}

async function checkEvenOrOdd(number) {

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

setTimeout(() => {

if (number % 2 === 0) {

resolve("even");

} else {

reject("odd");

}

}, 0);

});

}

(async function generateAndCheck() {

try {

const randomNumber = await generateRandomNumber();

console.log("Generated number:", randomNumber);

try {

let res = await checkEvenOrOdd(randomNumber);

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

} catch (error) {

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

}

} catch (error) {

console.error("Error generating random number:", error);

}

})();

**Output:-**

PS C:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-4\ass-4> node "c:\Users\DCA2023\Desktop\6th sem\Emerging Tech Lab\Assignment-4\ass-4\q2.js"

Generated number: 88

The number is even.

**3. Fetch products from** https://fakestoreapi.com/products **using js and 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-

Html page

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>21BCSE30 | ANIKET PATRA</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

box-sizing: border-box;

}

#container {

display: flex;

justify-content: center;

align-items: center;

min-height: 100vh;

background-color: #f5f5f5;

}

.products {

display: grid;

grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));

gap: 20px;

max-width: 1200px;

width: 100%;

padding: 20px;

}

.product {

border: 1px solid #ddd;

border-radius: 5px;

padding: 15px;

text-align: center;

background-color: #fff;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

img {

max-width: 100%;

height: 30vh;

margin-bottom: 10px;

}

h3 {

font-size: 1.2em;

margin-bottom: 10px;

}

p {

font-weight: bold;

color: #007bff;

}

</style>

</head>

<body>

<div id="container">

</div>

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

<script src="q3.js"></script>

</body>

</html>

JS PAGE

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

(async function getProducts() {

try {

let res = await axios(baseurl);

let div = `<div class="products">`;

res.data.forEach((product) => {

div += `<div class="product">

<img src="${product.image}" alt="image">

<h4>${product.title}</h3>

<p>$${product.price}</p>

</div>`;

});

div += `</div>`;

document.getElementById("container").innerHTML = div;

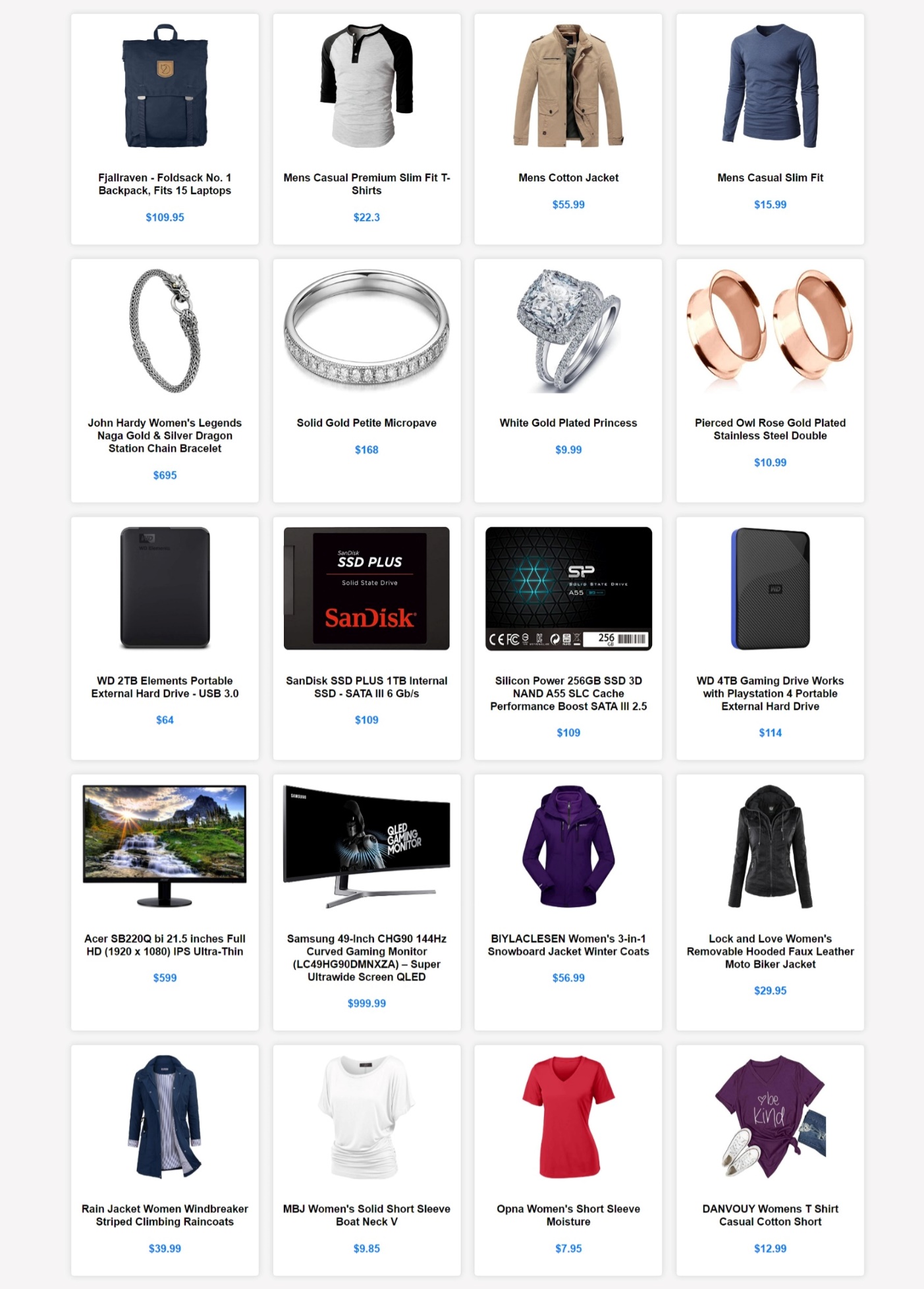
} catch (error) {

console.log(error);

}

})();

**Output:-**



**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-

db.json

{

"students": [

{

"id": "1",

"name": "John Doe",

"branch": "Computer Science",

"CGPA": 3.5

},

{

"id": "2",

"name": "Jane Smith",

"branch": "Electrical Engineering",

"CGPA": 5.6

},

{

"id": "3",

"name": "Alice Johnson",

"branch": "Mechanical Engineering",

"CGPA": 3.2

},

{

"id": "86",

"name": "Aniket Patra",

"branch": "CSE",

"CGPA": 9.63

},

{

"id": "11",

"name": "Anmol Partha Behera",

"branch": "CSE",

"CGPA": 9.77

},

{

"id": "61",

"name": "Ashutosh Panda",

"branch": "CSE",

"CGPA": 8.89

},

{

"id": "66",

"name": "Abhijit Mahakul",

"branch": "CSE",

"CGPA": 9.69

},

{

"id": "80",

"name": "Chandan Nayak",

"branch": "CSE",

"CGPA": 9.98

}

]

}

addStudent.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>21BCSE30 | ANIKET PATRA</title>

<link rel="stylesheet" href="./BS5/bootstrap.min.css" />

</head>

<body>

<nav class="navbar navbar-expand-lg bg-body-tertiary">

<div class="container-fluid">

<a class="navbar-brand" href="#">Student Records</a>

<button

class="navbar-toggler"

type="button"

data-bs-toggle="collapse"

data-bs-target="#navbarNav"

aria-controls="navbarNav"

aria-expanded="false"

aria-label="Toggle navigation"

>

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="./addStudent.html">Add Student</a>

</li>

<li class="nav-item">

<a class="nav-link" href="./displayStudent.html"

>Display Student Records</a

>

</li>

</ul>

</div>

</div>

</nav>

<section class="vh-100" style="background-color: #eee">

<div class="container h-100">

<div class="row d-flex justify-content-center align-items-center h-100">

<div class="col-lg-12 col-xl-11">

<div class="card text-black" style="border-radius: 25px">

<div class="card-body p-md-5">

<div class="row justify-content-center">

<div class="col-md-10 col-lg-6 col-xl-5 order-2 order-lg-1">

<p class="text-center h1 fw-bold mb-5 mx-1 mx-md-4 mt-4">

Add Student

</p>

<form class="mx-1 mx-md-4" onsubmit="return validate()">

<div class="d-flex flex-row align-items-center mb-4">

<i class="fas fa-user fa-lg me-3 fa-fw"></i>

<div class="form-outline flex-fill mb-0">

<label class="form-label" for="form3Example1c"

>Your Name</label

>

<input type="text" id="name" class="form-control" />

</div>

</div>

<div class="d-flex flex-row align-items-center mb-4">

<i class="fas fa-envelope fa-lg me-3 fa-fw"></i>

<div class="form-outline flex-fill mb-0">

<label class="form-label" for="form3Example3c"

>Your Branch</label

>

<input type="text" id="branch" class="form-control" />

</div>

</div>

<div class="d-flex flex-row align-items-center mb-4">

<i class="fas fa-lock fa-lg me-3 fa-fw"></i>

<div class="form-outline flex-fill mb-0">

<label class="form-label" for="form3Example4c"

>Your CGPA</label

>

<input type="text" id="cgpa" class="form-control" />

</div>

</div>

<div

class="d-flex justify-content-center mx-4 mb-3 mb-lg-4"

>

<button

type="button"

class="btn btn-primary btn-lg"

onclick="register()"

>

Register Student

</button>

</div>

</form>

</div>

<div

class="col-md-10 col-lg-6 col-xl-7 d-flex align-items-center order-1 order-lg-2"

>

<img

src="https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-registration/draw1.webp"

class="img-fluid"

alt="Sample image"

/>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</section>

<script src="./BS5/bootstrap.bundle.min.js"></script>

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

<script src="q4.js"></script>

<script>

async function register() {

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

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

const cgpa = parseFloat(document.getElementById("cgpa").value);

if (name === "" || branch === "" || isNaN(cgpa)) {

alert("Please fill all the fields!");

return;

}

if (cgpa < 1 || cgpa > 10) {

alert("Invalid CGPA!!");

return;

}

const student = {

id: `${Math.floor(Math.random() \* 100)}`,

name: name,

branch: branch,

CGPA: cgpa,

};

await addStudent(student);

(async function () {

alert("Redirecting to All students Page .....");

window.location.href = "./displayStudent.html";

})();

}

</script>

</body>

</html>

displayStudent.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<link rel="stylesheet" href="./BS5/bootstrap.min.css" />

<title>21BCSE30 | ANIKET PATRA</title>

<script src="./searchStudent.js"></script>

</head>

<body>

<nav class="navbar navbar-expand-lg bg-body-tertiary">

<div class="container-fluid">

<a class="navbar-brand" href="#">Student Records</a>

<button

class="navbar-toggler"

type="button"

data-bs-toggle="collapse"

data-bs-target="#navbarNav"

aria-controls="navbarNav"

aria-expanded="false"

aria-label="Toggle navigation"

>

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="./addStudent.html">Add Student</a>

</li>

<li class="nav-item">

<a class="nav-link" href="./displayStudent.html"

>Display Student Records</a

>

</li>

</ul>

<form class="d-flex" role="search" onsubmit="prevent(event)">

<input

class="form-control me-2"

type="search"

placeholder="Student name ..."

aria-label="Search"

id="searchInput"

/>

<button class="btn btn-outline-success" onclick="searchStudent()">

Search

</button>

</form>

</div>

</div>

</nav>

<h1 style="text-align: center" class="mt-2 mb-2">Student Table</h1>

<table class="table table-hover" id="studentTable"></table>

<center>

<button

class="btn btn-primary"

onclick="window.location.href='addStudent.html'"

>

Add Student

</button>

</center>

<script src="./BS5/bootstrap.bundle.min.js"></script>

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

<script src="q4.js"></script>

<script>

const data = displayStudents();

data

.then((students) => {

const table = document.querySelector("table");

table.innerHTML =

"<tr><th>S.No</th><th>ID</th><th>Name</th><th>Branch</th><th>CGPA</th><th>Update</th><th>Delete</th></tr>";

students.forEach((student,index) => {

let row = `<tr><td>${index+1}</td><td>${student.id}</td><td>${student.name}</td><td>${student.branch}</td><td>${student.CGPA}</td><td><button class="btn btn-success" onclick="update(${student.id})">Update</button></td><td><button class="btn btn-danger" onclick="deleted(${student.id})">Delete</button></td></tr>`;

table.innerHTML += row;

});

})

.catch((err) => console.log(err));

async function update(id) {

const student = await getStudent(id);

console.log(student);

localStorage.setItem("student", JSON.stringify(student));

window.location.href = "updateStudent.html";

}

async function deleted(id) {

await deleteStudent(id);

window.location.href = "displayStudent.html";

}

</script>

</body>

</html>

updateStudent.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>21BCSE30 | ANIKET PATRA</title>

<link rel="stylesheet" href="./BS5/bootstrap.min.css" />

</head>

<body>

<nav class="navbar navbar-expand-lg bg-body-tertiary">

<div class="container-fluid">

<a class="navbar-brand" href="#">Student Records</a>

<button

class="navbar-toggler"

type="button"

data-bs-toggle="collapse"

data-bs-target="#navbarNav"

aria-controls="navbarNav"

aria-expanded="false"

aria-label="Toggle navigation"

>

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNav">

<ul class="navbar-nav">

<li class="nav-item">

<a class="nav-link" href="./addStudent.html">Add Student</a>

</li>

<li class="nav-item">

<a class="nav-link" href="./displayStudent.html"

>Display Student Records</a

>

</li>

</ul>

</div>

</div>

</nav>

<section class="vh-100" style="background-color: #eee">

<div class="container h-100">

<div class="row d-flex justify-content-center align-items-center h-100">

<div class="col-lg-12 col-xl-11">

<div class="card text-black" style="border-radius: 25px">

<div class="card-body p-md-5">

<div class="row justify-content-center">

<div class="col-md-10 col-lg-6 col-xl-5 order-2 order-lg-1">

<p class="text-center h1 fw-bold mb-5 mx-1 mx-md-4 mt-4">

Update Student

</p>

<form class="mx-1 mx-md-4" onsubmit="return validate()">

<div class="d-flex flex-row align-items-center mb-4">

<i class="fas fa-user fa-lg me-3 fa-fw"></i>

<div class="form-outline flex-fill mb-0">

<label class="form-label" for="form3Example1c"

>Your Name</label

>

<input type="text" id="name" class="form-control" />

</div>

</div>

<div class="d-flex flex-row align-items-center mb-4">

<i class="fas fa-envelope fa-lg me-3 fa-fw"></i>

<div class="form-outline flex-fill mb-0">

<label class="form-label" for="form3Example3c"

>Your Branch</label

>

<input type="text" id="branch" class="form-control" />

</div>

</div>

<div class="d-flex flex-row align-items-center mb-4">

<i class="fas fa-lock fa-lg me-3 fa-fw"></i>

<div class="form-outline flex-fill mb-0">

<label class="form-label" for="form3Example4c"

>Your CGPA</label

>

<input type="text" id="cgpa" class="form-control" />

</div>

</div>

<div

class="d-flex justify-content-center mx-4 mb-3 mb-lg-4"

>

<button

type="button"

class="btn btn-primary btn-lg"

onclick="Update()"

>

Update Student

</button>

</div>

</form>

</div>

<div

class="col-md-10 col-lg-6 col-xl-7 d-flex align-items-center order-1 order-lg-2"

>

<img

src="https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-registration/draw1.webp"

class="img-fluid"

alt="Sample image"

/>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</section>

<script src="./BS5/bootstrap.bundle.min.js"></script>

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

<script src="q4.js"></script>

<script>

let id;

(async function updateStud() {

const student = await JSON.parse(localStorage.getItem("student"));

id = student.id;

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

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

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

localStorage.removeItem("student");

localStorage.clear();

})();

function Update() {

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

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

const cgpa = parseFloat(document.getElementById("cgpa").value);

if (name === "" || branch === "" || isNaN(cgpa)) {

alert("Please fill all the fields!");

return false;

}

if (cgpa < 1 && cgpa > 10) {

alert("Invalid CGPA!!");

return false;

}

const updatedData = {

id: id,

name: name,

branch: branch,

CGPA: cgpa,

};

updateStudent(id, updatedData);

window.location.href = "displayStudent.html";

}

</script>

</body>

</html>

JS FILES

// 21BCSE30 | ANIKET PATRA

async function addStudent(studentData) {

try {

const response = await axios.post(

"http://localhost:3000/students",

studentData

);

if(response.status === 201){

alert("Student added successfully!");

}

} catch (error) {

console.error("Error adding student:", error);

}

}

async function displayStudents() {

try {

const response = await axios.get("http://localhost:3000/students");

return response.data;

} catch (error) {

console.error("Error fetching students:", error);

}

}

async function getStudent(id) {

try {

const response = await axios.get(`http://localhost:3000/students/${id}`);

return response.data;

} catch (error) {

console.error("Error fetching student", error);

}

}

async function updateStudent(id, updatedData) {

try {

const response = await axios.put(

`http://localhost:3000/students/${id}`,

updatedData

);

if (response.status === 200) {

alert("Student updated successfully!");

} else {

throw new Error("Error in updating the Student...");

}

} catch (error) {

console.error("Error updating student:", error);

}

}

async function deleteStudent(id) {

try {

const response = await axios.delete(`http://localhost:3000/students/${id}`);

alert("Student deleted successfully!");

} catch (error) {

console.error("Error deleting student:", error);

}

}

//21BCSE30 | ANIKET PATRA

function prevent(e){

e.preventDefault();

}

async function searchStudent() {

var input, filter, table, tr, td, txtValue;

input = document.getElementById("searchInput");

filter = input.value.toUpperCase();

table = document.getElementById("studentTable");

tr = table.getElementsByTagName("tr");

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

td = tr[i].getElementsByTagName("td")[1]; // Assuming the student name is in the first column

if (td) {

txtValue = td.textContent || td.innerText;

if (txtValue.toUpperCase().indexOf(filter) > -1) {

tr[i].style.display = "";

} else {

tr[i].style.display = "none";

}

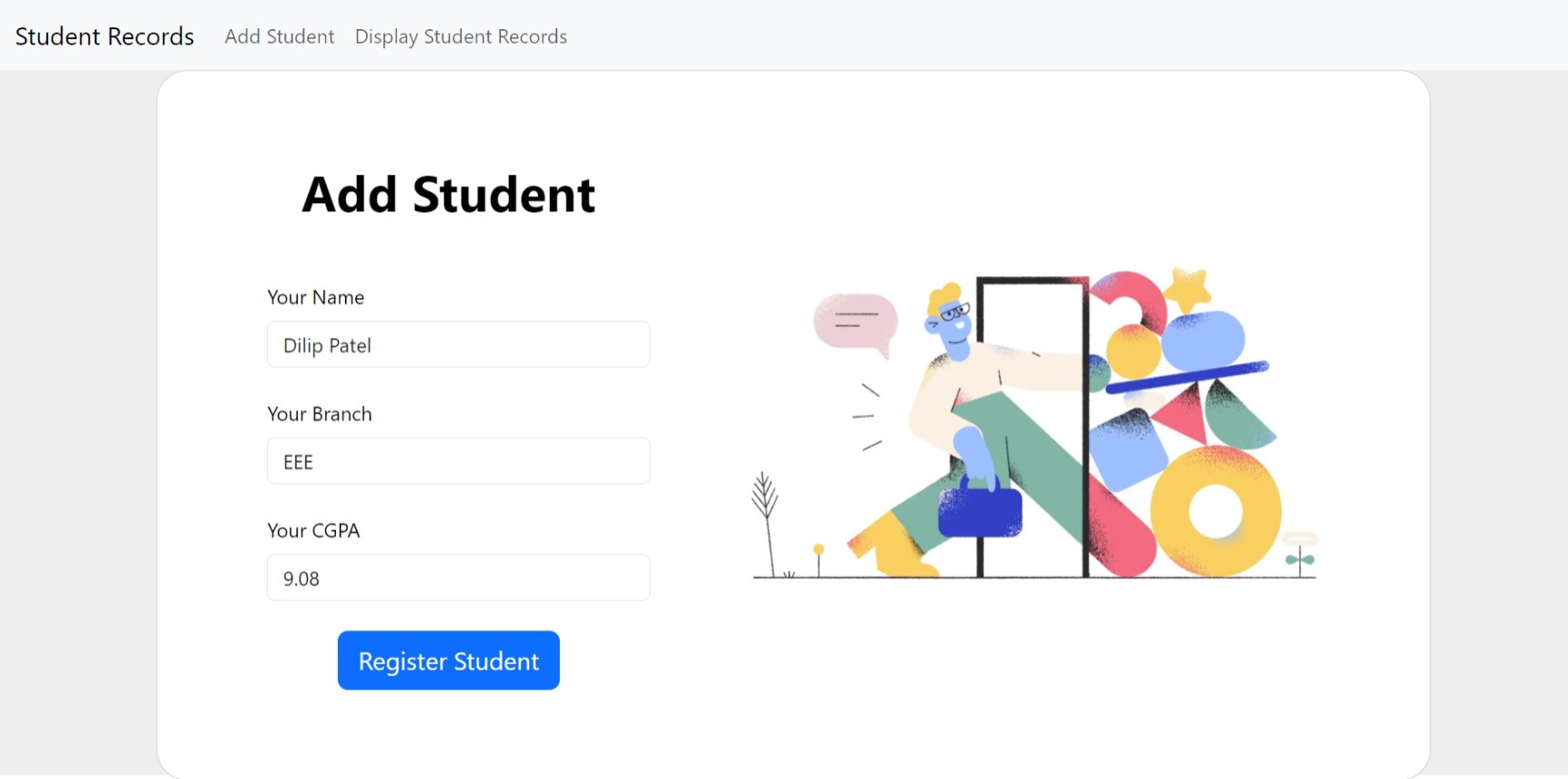
}

}

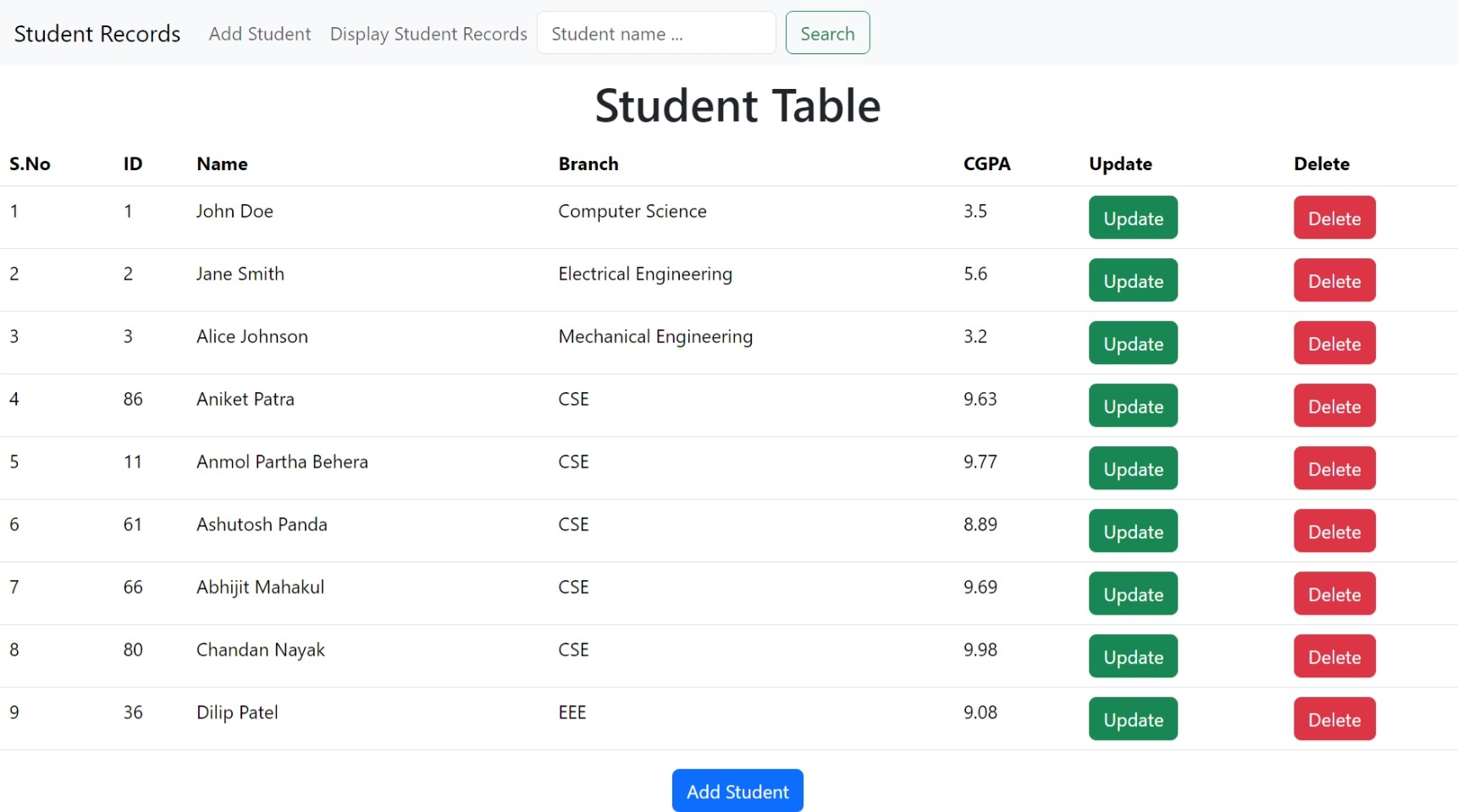
}

**Output:-**

**Add a Student**



**Display All Students**



**Search a student**

