ASSIGNMENT

**PART A – FRONTEND**

**Q1.**

****

**register.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>STUDENT REGISTRATION PAGE</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="form-box">

<h1>STUDENT REGISTRATION</h1>

<form>

<div class="form-group">

<label>Full Name</label>

<input type="text" name="fullname" required>

</div>

<div class="form-group">

<label>Email</label>

<input type="email" name="email" required>

</div>

<div class="form-group">

<label>Password</label>

<input type="password" name="password" required>

</div>

<div class="form-group">

<label>Confirm</label>

<input type="password" name="confirm" required>

</div>

<div class="form-group">

<label>Gender</label>

<div class="options">

<input type="radio" name="gender" value="Male"> Male

<input type="radio" name="gender" value="Female"> Female

<input type="radio" name="gender" value="Other"> Other

</div>

</div>

<div class="form-group">

<label>Course</label>

<select name="course">

<option>--Select--</option>

<option>MCA</option>

<option>MSc</option>

<option>BCA</option>

<option>Other</option>

</select>

</div>

<div class="form-group">

<label>Year</label>

<select name="year">

<option>--Select--</option>

<option>First</option>

<option>Second</option>

<option>Third</option>

</select>

</div>

<div class="form-group">

<label>Skills</label><br>

<div class="options">

<input type="checkbox" name="skills" value="Java"> Java

<input type="checkbox" name="skills" value="Python"> Python

<input type="checkbox" name="skills" value="HTML/CSS"> HTML/CSS

<input type="checkbox" name="skills" value="JavaScript"> JavaScript

<input type="checkbox" name="skills" value="Node.js"> Node.js

<input type="checkbox" name="skills" value="C"> C

</div>

</div>

<div class="form-group">

<button type="submit">Register</button>

</div>

</form>

</div>

</body>

</html>

**style.css**

body {

background:url("https://img.freepik.com/premium-vector/mountain-range-rises-background-with-solitary-tree-prominently-positioned-foreground-flat-vector-illustration\_674398-10552.jpg") no-repeat center center fixed;

background-color: black;

background-size: cover;

}

.form-box {

width: 400px;

margin: 60px auto;

padding: 20px;

background: rgba(255, 255, 255, 0.592);

border-radius: 8px;

}

h1 {

text-align: center;

font-size: 22px;

margin-bottom: 20px;

}

.form-group {

display: flex;

align-items: center;

margin-bottom: 12px;

}

.form-group label {

width: 120px;

font-weight: bold;

}

.form-group input,

.form-group select {

flex: 1;

padding: 5px;

background: rgba(255, 255, 255, 0.592);

border-radius: 5px;

}

button {

width: 100%;

padding: 8px;

background: #256794;

color: white;

border: none;

cursor: pointer;

border-radius: 5px;

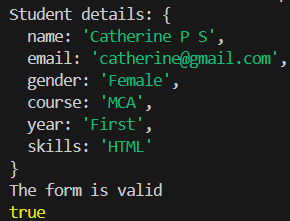
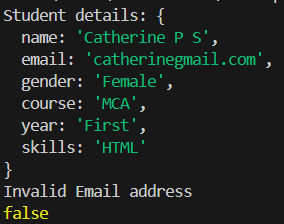
}

button:hover {

background: #457fa0;

}

**Q2.**

** **

**script.js**  
  
var msge = "The form is valid";

var studname = "Catherine P S";

var studemail = "catherine@gmail.com";

var studgender = "Female";

var studcourse = "MCA";

var studyear = "First";

var studskills = "HTML";

var student = {

name: studname,

email: studemail,

gender: studgender,

course: studcourse,

year: studyear,

skills: studskills

};

function validateForm(student) {

let c = true;

if (!student.email || !student.email.includes("@gmail.com")) {

console.log("Invalid Email address");

c = false;

}

if (!student.gender) {

console.log("Gender is required");

c = false;

}

if (c) {

console.log(msge);

}

return c;

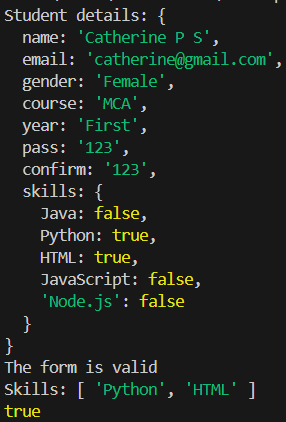
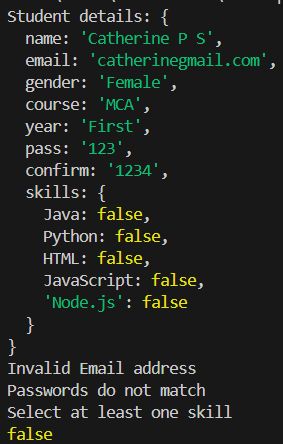
}

console.log("Student details:", student);

var a = validateForm(student);

console.log(a);

**Q3.**

**script.js**

var msge = "The form is valid";

var studname = "Catherine P S";

var studemail = "catherinegmail.com";

var studgender = "Female";

var studcourse = "MCA";

var studyear = "First";

var studpass = "123";

var studconfirm = "1234";

var studskills = {

Java: false,

Python: false,

HTML: false,

JavaScript: false,

"Node.js": false

};

var student = {

name: studname,

email: studemail,

gender: studgender,

course: studcourse,

year: studyear,

pass: studpass,

confirm: studconfirm,

skills: studskills

};

function validateForm(student) {

let c = true;

if (!student.email) {

console.log("The Email is empty");

c = false;

} else if (!student.email.includes("@gmail.com")) {

console.log("Invalid Email address");

c = false;

}

if (student.pass !== student.confirm) {

console.log("Passwords do not match");

c = false;

}

let selectedSkills = [];

for (let i in student.skills) {

if (student.skills[i]) {

selectedSkills.push(i);

}

}

if (selectedSkills.length === 0) {

console.log("Select at least one skill");

c = false;

}

if (c) {

console.log(msge);

console.log("Skills:", selectedSkills);

}

return c;

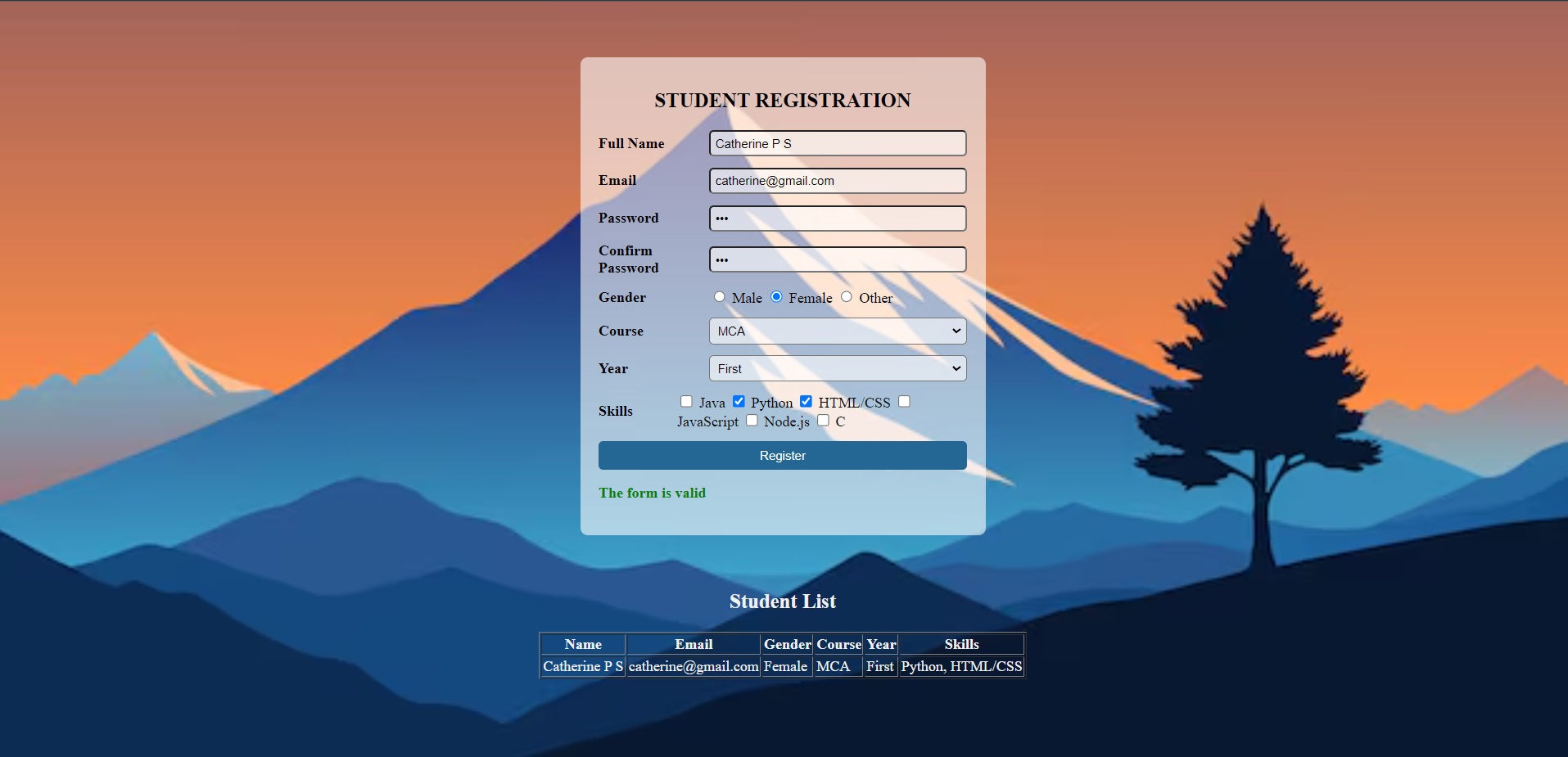
}

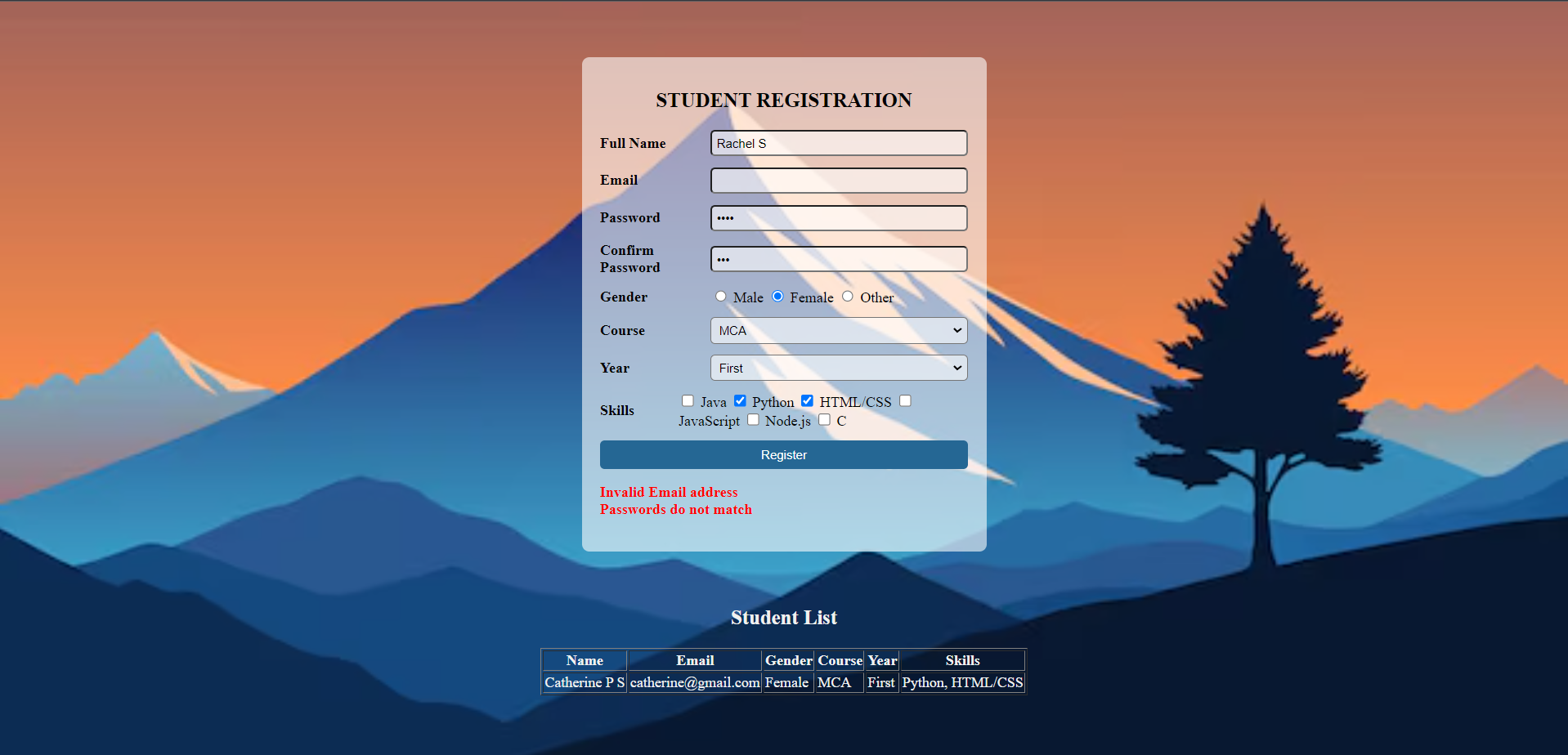
console.log("Student details:", student);

var a = validateForm(student);

console.log(a);

**Q4.**





**student.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>STUDENT REGISTRATION PAGE</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="form-box">

<h1>STUDENT REGISTRATION</h1>

<form id="regForm">

<div class="form-group">

<label>Full Name</label>

<input type="text" name="fullname" id="studname" required>

</div>

<div class="form-group">

<label>Email</label>

<input type="email" name="email" id="studemail" required>

</div>

<div class="form-group">

<label>Password</label>

<input type="password" name="password" id="studpassword" required>

</div>

<div class="form-group">

<label>Confirm Password</label>

<input type="password" name="confirm" id="studconfirm" required>

</div>

<div class="form-group">

<label>Gender</label>

<div class="options">

<input type="radio" name="gender" value="Male"> Male

<input type="radio" name="gender" value="Female"> Female

<input type="radio" name="gender" value="Other"> Other

</div>

</div>

<div class="form-group">

<label>Course</label>

<select name="course" id="studcourse">

<option>--Select--</option>

<option>MCA</option>

<option>MSc</option>

<option>BCA</option>

<option>Other</option>

</select>

</div>

<div class="form-group">

<label>Year</label>

<select name="year" id="studyear">

<option>--Select--</option>

<option>First</option>

<option>Second</option>

<option>Third</option>

</select>

</div>

<div class="form-group">

<label>Skills</label><br>

<div class="options">

<input type="checkbox" class="skill" value="Java"> Java

<input type="checkbox" class="skill" value="Python"> Python

<input type="checkbox" class="skill" value="HTML/CSS"> HTML/CSS

<input type="checkbox" class="skill" value="JavaScript"> JavaScript

<input type="checkbox" class="skill" value="Node.js"> Node.js

<input type="checkbox" class="skill" value="C"> C

</div>

</div>

<div class="form-group">

<button type="button" id="registerBtn">Register</button>

</div>

<p id="msg" style="color:green;font-weight:bold;"></p>

</form>

</div>

<center>

<h1 style="color:rgb(240, 241, 240);">Student List</h1>

<table border="1" id="studentTable" style="color:rgb(240, 241, 240);">

<thead>

<tr>

<th>Name</th>

<th>Email</th>

<th>Gender</th>

<th>Course</th>

<th>Year</th>

<th>Skills</th>

</tr>

</thead>

<tbody>

</tbody>

</table>

<br><br>

</center>

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

</body>

</html>

**style.css**

body {

background:url("https://img.freepik.com/premium-vector/mountain-range-rises-background-with-solitary-tree-prominently-positioned-foreground-flat-vector-illustration\_674398-10552.jpg") no-repeat center center fixed;

background-color: black;

background-size: cover;

}

.form-box {

width: 400px;

margin: 60px auto;

padding: 20px;

background: rgba(255, 255, 255, 0.592);

border-radius: 8px;

}

h1 {

text-align: center;

font-size: 22px;

margin-bottom: 20px;

}

.form-group {

display: flex;

align-items: center;

margin-bottom: 12px;

}

.form-group label {

width: 120px;

font-weight: bold;

}

.form-group input,

.form-group select {

flex: 1;

padding: 5px;

background: rgba(255, 255, 255, 0.592);

border-radius: 5px;

}

button {

width: 100%;

padding: 8px;

background: #256794;

color: white;

border: none;

cursor: pointer;

border-radius: 5px;

}

button:hover {

background: #457fa0;

}

**script.js**

var msge = "The form is valid";

function validateForm(student) {

let c = true;

let error = "";

if (!student.email || !student.email.includes("@gmail")) {

error += "Invalid Email address<br>";

c = false;

}

if (student.password !== student.confirm) {

error += "Passwords do not match<br>";

c = false;

}

if (student.skills.length === 0) {

error += "Select at least one skill<br>";

c = false;

}

if (c) {

document.getElementById("msg").style.color = "green";

document.getElementById("msg").innerHTML = msge;

} else {

document.getElementById("msg").style.color = "red";

document.getElementById("msg").innerHTML = error;

}

return c;

}

document.getElementById("registerBtn").addEventListener("click", function() {

var student = {

name: document.getElementById("studname").value,

email: document.getElementById("studemail").value,

gender: document.querySelector('input[name="gender"]:checked') ?

document.querySelector('input[name="gender"]:checked').value : "",

course: document.getElementById("studcourse").value,

year: document.getElementById("studyear").value,

password: document.getElementById("studpassword").value,

confirm: document.getElementById("studconfirm").value,

skills: []

};

var skillBoxes = document.querySelectorAll(".skill");

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

if (skillBoxes[i].checked) {

student.skills.push(skillBoxes[i].value);

}

}

if (validateForm(student)) {

var table = document.getElementById("studentTable").getElementsByTagName("tbody")[0];

var row = document.createElement("tr");

let values = [student.name, student.email, student.gender, student.course, student.year, student.skills.join(", ")];

for (let val of values) {

let cell = document.createElement("td");

cell.textContent = val;

row.appendChild(cell);

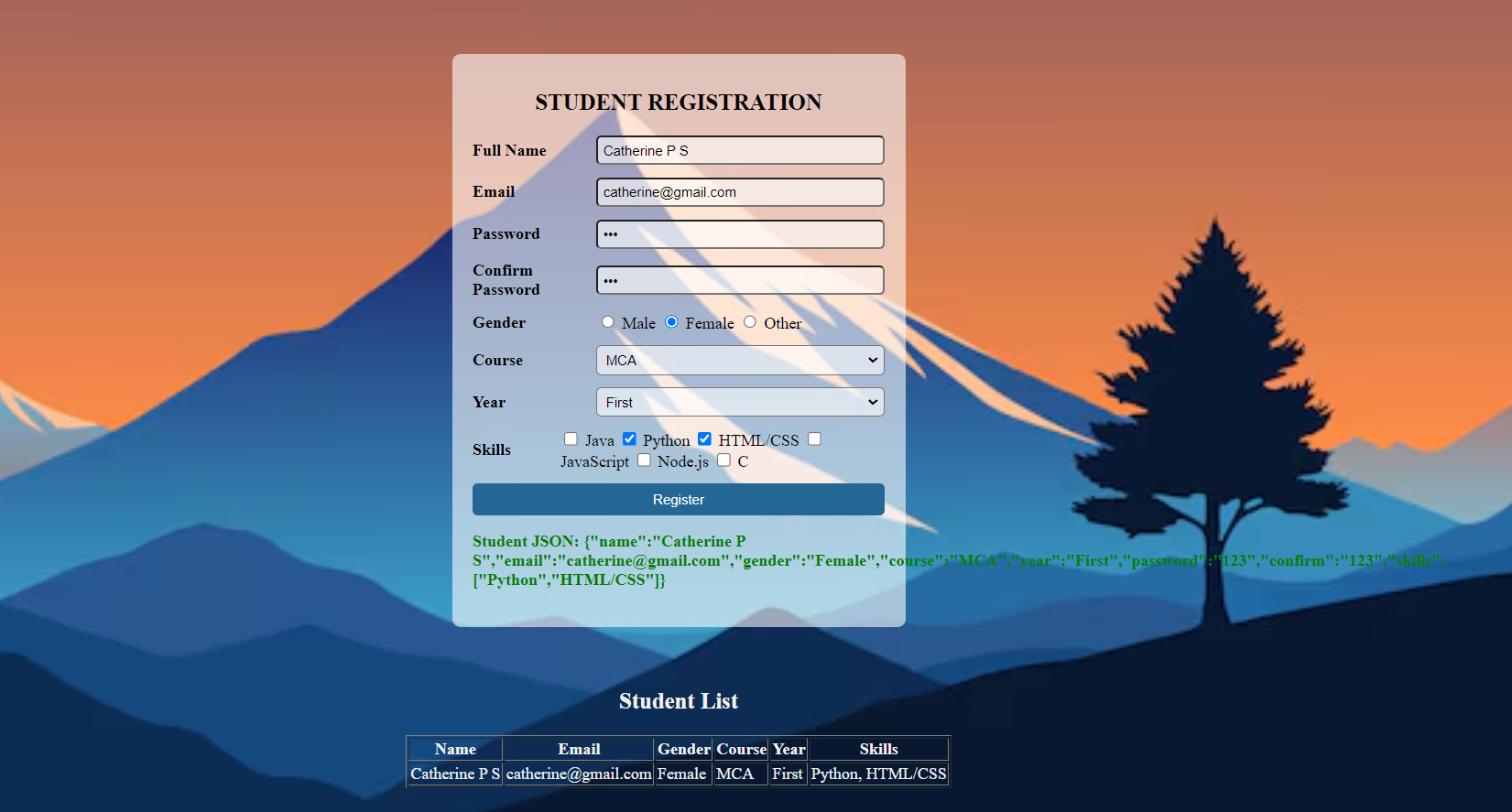
}

table.appendChild(row);

}

});

**Q5.**



**student.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>STUDENT REGISTRATION PAGE</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="form-box">

<h1>STUDENT REGISTRATION</h1>

<form id="regForm">

<div class="form-group">

<label>Full Name</label>

<input type="text" name="fullname" id="studname" required>

</div>

<div class="form-group">

<label>Email</label>

<input type="email" name="email" id="studemail" required>

</div>

<div class="form-group">

<label>Password</label>

<input type="password" name="password" id="studpassword" required>

</div>

<div class="form-group">

<label>Confirm Password</label>

<input type="password" name="confirm" id="studconfirm" required>

</div>

<div class="form-group">

<label>Gender</label>

<div class="options">

<input type="radio" name="gender" value="Male"> Male

<input type="radio" name="gender" value="Female"> Female

<input type="radio" name="gender" value="Other"> Other

</div>

</div>

<div class="form-group">

<label>Course</label>

<select name="course" id="studcourse">

<option>--Select--</option>

<option>MCA</option>

<option>MSc</option>

<option>BCA</option>

<option>Other</option>

</select>

</div>

<div class="form-group">

<label>Year</label>

<select name="year" id="studyear">

<option>--Select--</option>

<option>First</option>

<option>Second</option>

<option>Third</option>

</select>

</div>

<div class="form-group">

<label>Skills</label><br>

<div class="options">

<input type="checkbox" class="skill" value="Java"> Java

<input type="checkbox" class="skill" value="Python"> Python

<input type="checkbox" class="skill" value="HTML/CSS"> HTML/CSS

<input type="checkbox" class="skill" value="JavaScript"> JavaScript

<input type="checkbox" class="skill" value="Node.js"> Node.js

<input type="checkbox" class="skill" value="C"> C

</div>

</div>

<div class="form-group">

<button type="button" id="registerBtn">Register</button>

</div>

<p id="msg" style="color:green;font-weight:bold;"></p>

</form>

</div>

<center>

<h1 style="color:rgb(240, 241, 240);">Student List</h1>

<table border="1" id="studentTable" style="color:rgb(240, 241, 240);">

<thead>

<tr>

<th>Name</th>

<th>Email</th>

<th>Gender</th>

<th>Course</th>

<th>Year</th>

<th>Skills</th>

</tr>

</thead>

<tbody>

</tbody>

</table>

<br><br>

</center>

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

</body>

</html>

**style.css**

body {

background:url("https://img.freepik.com/premium-vector/mountain-range-rises-background-with-solitary-tree-prominently-positioned-foreground-flat-vector-illustration\_674398-10552.jpg") no-repeat center center fixed;

background-color: black;

background-size: cover;

}

.form-box {

width: 400px;

margin: 60px auto;

padding: 20px;

background: rgba(255, 255, 255, 0.592);

border-radius: 8px;

}

h1 {

text-align: center;

font-size: 22px;

margin-bottom: 20px;

}

.form-group {

display: flex;

align-items: center;

margin-bottom: 12px;

}

.form-group label {

width: 120px;

font-weight: bold;

}

.form-group input,

.form-group select {

flex: 1;

padding: 5px;

background: rgba(255, 255, 255, 0.592);

border-radius: 5px;

}

button {

width: 100%;

padding: 8px;

background: #256794;

color: white;

border: none;

cursor: pointer;

border-radius: 5px;

}

button:hover {

background: #457fa0;

}

**script.js**

let students = [];

function validateForm(student) {

if (

student.name === "" ||

student.email === "" ||

student.password === "" ||

student.confirm === "" ||

student.course === "--Select--" ||

student.year === "--Select--"

) {

return false;

}

if (student.password !== student.confirm) {

alert("Passwords do not match!");

return false;

}

return true;

}

function registerStudent(callback) {

var student = {

name: document.getElementById("studname").value,

email: document.getElementById("studemail").value,

gender: document.querySelector('input[name="gender"]:checked')

? document.querySelector('input[name="gender"]:checked').value

: "",

course: document.getElementById("studcourse").value,

year: document.getElementById("studyear").value,

password: document.getElementById("studpassword").value,

confirm: document.getElementById("studconfirm").value,

skills: []

};

document.querySelectorAll(".skill").forEach(skillBox => {

if (skillBox.checked) student.skills.push(skillBox.value);

});

if (validateForm(student)) {

students.push(student);

let jsonData = JSON.stringify(student);

callback(student, jsonData);

} else {

document.getElementById("msg").style.color = "red";

document.getElementById("msg").innerText = "Form validation failed!";

}

}

function displayStudent(student, jsonData) {

var table = document

.getElementById("studentTable")

.getElementsByTagName("tbody")[0];

var row = document.createElement("tr");

let values = [

student.name,

student.email,

student.gender,

student.course,

student.year,

student.skills.join(", ")

];

values.forEach(val => {

let cell = document.createElement("td");

cell.textContent = val;

row.appendChild(cell);

});

table.appendChild(row);

document.getElementById("msg").style.color = "green";

document.getElementById("msg").innerText = "Student JSON: " + jsonData;

}

document

.getElementById("registerBtn")

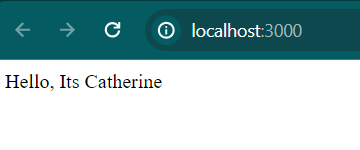
.addEventListener("click", function () {

registerStudent(displayStudent);

});

**PART B – BACKEND**

**Q6.**



**Terminal**

npm install express body-parser

npm install -g nodemon

**index.js**

const e = require("express");

const b= require("body-parser");

const a = e();

a.use(b.json());

a.get("/", (req, res) => {

res.send("Hello, Its Catherine");

});

a.listen(3000, () => {

console.log(`Server running at http://localhost:3000`);

});

**Q7.**

****

**index.js**

const a = require("fs");

a.readFile("sample.txt", "utf8", (err, data) => {

if (err) {

console.error("Error reading file:", err);

return;

}

console.log("Content:", data);

});

**sample.txt**

Hello,Its Catherine

**Q8.**



****

**server.js**

const e = require("express");

const p = require("path");

const a = e();

a.use(e.static(p.join(\_\_dirname)));

a.get("/", (req, res) => {

res.sendFile(p.join(\_\_dirname, "register.html"));

});

a.listen(3000, () => {

console.log(`Server running at http://localhost:3000`);

});

**Q9.**



**studentModule.js**

let s = [];

function addStudent(student) {

s.push(student);

}

function getAllStudents() {

return s;

}

module.exports = {

addStudent,

getAllStudents

};

**server.js**

const s = require('./studentModule');

s.addStudent({

id : 101,name: "Catherine P S",

email: "catherine@gmail.com",

gender: "Female",

course: "MCA",

year: "First",

skills: "HTML" });

s.addStudent({

id : 102,

name: "Treesa",

email: "treesa@gmail.com",

gender: "Female",

course: "MCA",

year: "First",

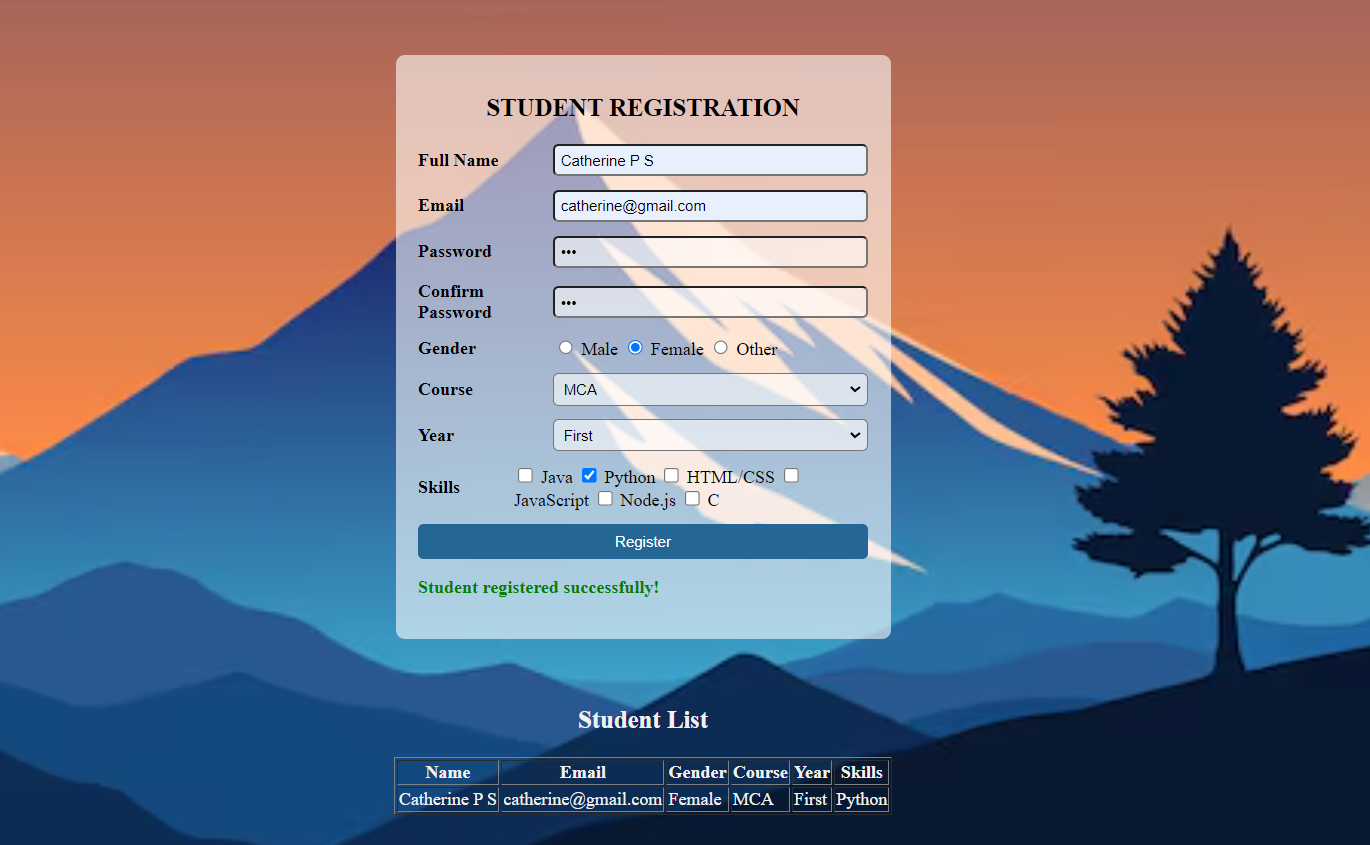
skills: "Python"});

const a = s.getAllStudents();

console.log("List of Students:");

console.log(a);

**Q10.**

****

**server.js**

const http = require('http');

const fs = require('fs');

const path = require('path');

const studentModule = require('./studentModule');

const PORT = 5000;

const server = http.createServer((req, res) => {

if (req.method === 'GET' && req.url === '/') {

fs.readFile(path.join(\_\_dirname, 'student.html'), (err, data) => {

if (err) {

res.writeHead(500);

res.end('Error loading page');

} else {

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

res.end(data);

}

});

} else if (req.method === 'GET' && req.url === '/style.css') {

fs.readFile(path.join(\_\_dirname, 'style.css'), (err, data) => {

if (err) {

res.writeHead(500);

res.end('Error loading CSS');

} else {

res.writeHead(200, { 'Content-Type': 'text/css' });

res.end(data);

}

});

} else if (req.method === 'GET' && req.url === '/script.js') {

fs.readFile(path.join(\_\_dirname, 'script.js'), (err, data) => {

if (err) {

res.writeHead(500);

res.end('Error loading JavaScript');

} else {

res.writeHead(200, { 'Content-Type': 'application/javascript' });

res.end(data);

}

});

} else if (req.method === 'GET' && req.url === '/students') {

const students = studentModule.getAllStudents();

res.writeHead(200, { 'Content-Type': 'application/json' });

res.end(JSON.stringify(students));

} else if (req.method === 'POST' && req.url === '/register') {

let body = '';

req.on('data', chunk => { body += chunk.toString(); });

req.on('end', () => {

try {

const student = JSON.parse(body);

studentModule.addStudent(student);

res.writeHead(200, { 'Content-Type': 'application/json' });

res.end(JSON.stringify({ message: 'Student registered successfully!' }));

} catch {

res.writeHead(400, { 'Content-Type': 'application/json' });

res.end(JSON.stringify({ error: 'Invalid JSON' }));

}

});

} else {

res.writeHead(404, { 'Content-Type': 'text/plain' });

res.end('Route not found');

}

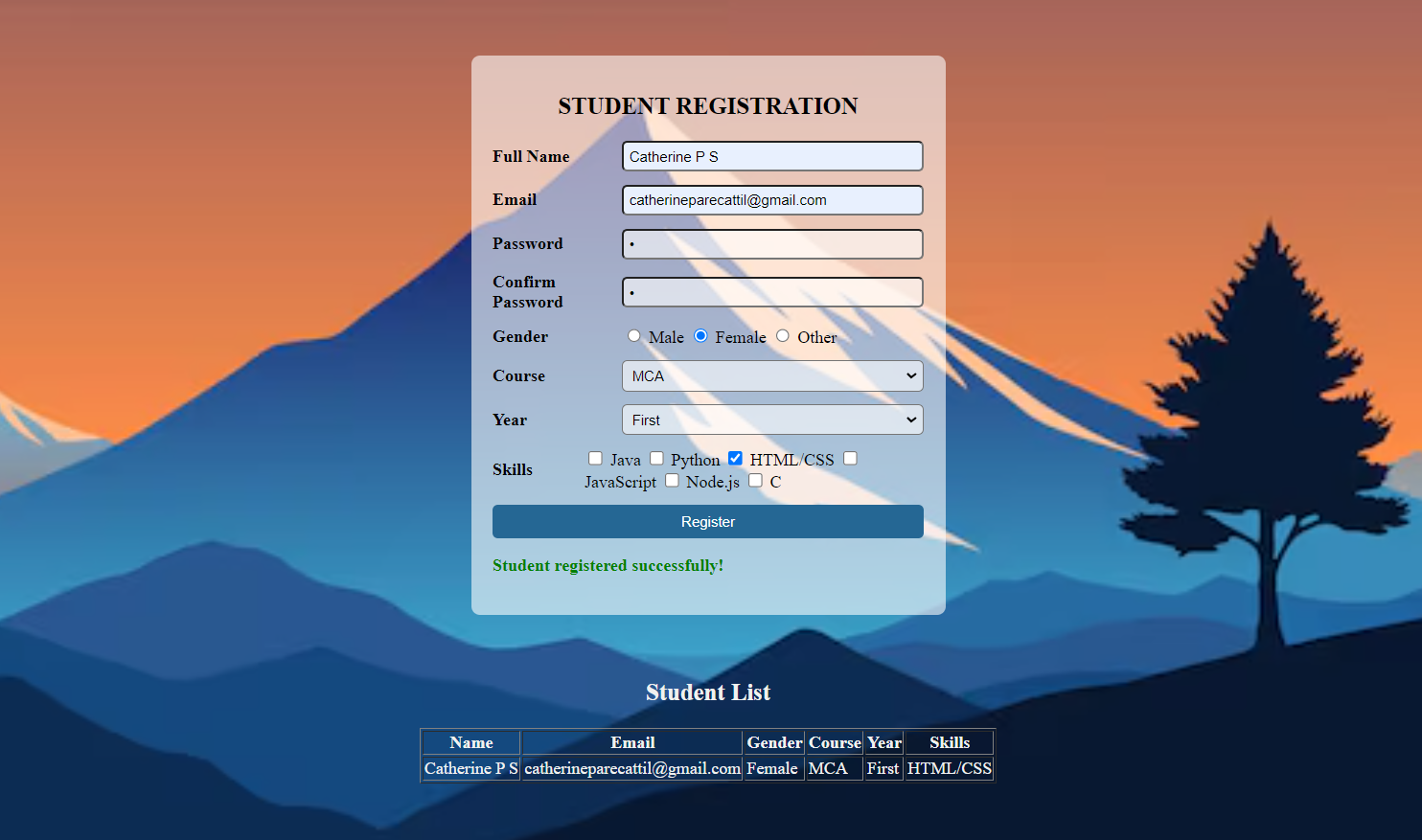
});

server.listen(PORT, () => {

console.log(`Server running at http://localhost:${PORT}`);

});

**Q11.**

****

**controllers\studentController.js**

const studentModel = require('../models/studentModel');

exports.registerStudent = (req, res) => {

try {

const student = req.body;

studentModel.addStudent(student);

res.status(200).json({ message: 'Student registered successfully!' });

} catch (error) {

res.status(400).json({ error: 'Invalid student data.' });

}

};

exports.getStudents = (req, res) => {

const students = studentModel.getAllStudents();

res.status(200).json(students);

};

**models\studentModel.js**

let students = [];

exports.addStudent = (student) => {

students.push(student);

};

exports.getAllStudents = () => {

return students;

};

**routes\studentRoutes.js**

const express = require('express');

const router = express.Router();

const studentController = require('../controllers/studentController');

router.post('/register', studentController.registerStudent);

router.get('/students', studentController.getStudents);

module.exports = router;

**app.js**

const express = require('express');

const bodyParser = require('body-parser');

const path = require('path');

const studentRoutes = require('./routes/studentRoutes');

const app = express();

const PORT = 8000;

app.use(bodyParser.json());

app.get('/', (req, res) => {

res.sendFile(path.join(\_\_dirname, 'public', 'student.html'));

});

app.use(express.static(path.join(\_\_dirname, 'public')));

app.use('/', studentRoutes);

app.listen(PORT, () => {

console.log(`Server is running on http://localhost:${PORT}`);

});