<!DOCTYPE html>

<html>

<head>

<meta charset=”utf-8”>

<title>Employee Detail Management System</title>

<style>

Body {

Font-family: Arial;

Margin: 20px;

Padding: 0;

}

.container {

Max-width: 800px;

Margin: auto;

Padding: 20px;

Border: 1px solid #ddd;

Background-color: powderblue;

}

Input, button {

Margin: 10px;

Padding: 10px;

Width: calc(100%);

Border-radius: 15px;

Box-sizing: border-box;

}

Table {

Width: 100%;

Margin-top: 20px;

}

Table, th, td {

Border: 1px solid #ddd;

Padding: 8px;

Text-align: left;

}

Th {

Background-color: white;

}

</style>

</head>

<body>

<div class=”container”>

<h2>Employee Detail Management System</h2>

<input type=”text” id=”name” placeholder=”Employee Name” required>

<input type=”number” id=”emp\_id” placeholder=”Employee ID” required>

<input type=”text” id=”department” placeholder=”Department” required>

<input type=”number” id=”salary” placeholder=”Salary” step=”0.01” required>

<button onclick=”addEmployee()”>Add Employee</button>

<input type=”number” id=”search\_emp\_id” placeholder=”Enter Employee ID to Search”>

<button onclick=”searchEmployee()”>Search Employee</button>

<input type=”number” id=”delete\_emp\_id” placeholder=”Enter Employee ID to Delete”>

<button onclick=”deleteEmployee()”>Delete Employee</button>

<button onclick=”displayEmployees()”>Display Employees</button>

<table id=”employeeTable”>

<thead>

<tr>

<th>Employee ID</th>

<th>Name</th>

<th>Department</th>

<th>Salary</th>

</tr>

</thead>

<tbody id=”employeeData”></tbody>

</table>

</div>

<script>

Class EmployeeNode {

Constructor(emp\_id, name, department, salary) {

This.emp\_id = emp\_id;

This.name = name;

This.department = department;

This.salary = salary;

This.next = null;

}

}

Class EmployeeList {

Constructor() {

This.head = null;

}

addEmployee(emp\_id, name, department, salary) {

const newEmployee = new EmployeeNode(emp\_id, name, department, salary);

if (this.findEmployee(emp\_id)) {

alert(“Employee record already exists.”);

return;

}

If (!this.head || this.head.emp\_id > emp\_id) {

newEmployee.next = this.head;

this.head = newEmployee;

} else {

Let current = this.head;

While (current.next && current.next.emp\_id < emp\_id) {

Current = current.next;

}

newEmployee.next = current.next;

current.next = newEmployee;

}

}

deleteEmployee(emp\_id) {

if (!this.head) return false;

if (this.head.emp\_id === emp\_id) {

this.head = this.head.next;

return true;

}

Let current = this.head;

While (current.next && current.next.emp\_id !== emp\_id) {

Current = current.next;

}

If (current.next) {

Current.next = current.next.next;

Return true;

} else {

Return false;

}

}

findEmployee(emp\_id) {

let current = this.head;

while (current) {

if (current.emp\_id === emp\_id) return current;

current = current.next;

}

Return null;

}

displayEmployees() {

let current = this.head;

const employees = [];

while (current) {

employees.push(current);

current = current.next;

}

Return employees;

}

}

Const employeeList = new EmployeeList();

Function addEmployee() {

Const name = document.getElementById(“name”).value;

Const emp\_id = parseInt(document.getElementById(“emp\_id”).value);

Const department = document.getElementById(“department”).value;

Const salary = parseFloat(document.getElementById(“salary”).value);

If (!name || !emp\_id || !department || !salary) {

Alert(“Please fill all the fields.”);

Return;

}

employeeList.addEmployee(emp\_id, name, department, salary);

clearInputs();

displayEmployees();

}

Function deleteEmployee() {

Const emp\_id = parseInt(document.getElementById(“delete\_emp\_id”).value);

Const result = employeeList.deleteEmployee(emp\_id);

If (result) {

Alert(“Employee record deleted.”);

} else {

Alert(“Employee record not found.”);

}

Document.getElementById(“delete\_emp\_id”).value = “”;

displayEmployees();

}

Function searchEmployee() {

Const emp\_id = parseInt(document.getElementById(“search\_emp\_id”).value);

Const employee = employeeList.findEmployee(emp\_id);

If (employee) {

Alert(“Employee found.”);

} else {

Alert(“Employee not found.”);

}

Document.getElementById(“search\_emp\_id”).value = “”;

}

Function displayEmployees() {

Const employeeData = document.getElementById(“employeeData”);

employeeData.innerHTML = “”;

const employees = employeeList.displayEmployees();

employees.forEach(emp => {

const row = `<tr>

<td>${emp.emp\_id}</td>

<td>${emp.name}</td>

<td>${emp.department}</td>

<td>${emp.salary.toFixed(2)}</td>

</tr>`;

employeeData.innerHTML += row;

});

}

Function clearInputs() {

Document.getElementById(“name”).value = “”;

Document.getElementById(“emp\_id”).value = “”;

Document.getElementById(“department”).value = “”;

Document.getElementById(“salary”).value = “”;

}

</script>

</body>

</html>