Ex: 02 **STUDENT MANAGEMENT SYSTEM** **Date: 18.12.2025**

Name: Ajay Kumar S

Roll no: 2024503701

Aim:

To design student management system.

Description:

The webpage is created using various HTML and CSS tags. Below are the tags used along with a short description:

JavaScript Concepts Used:

- **DOM Manipulation** –
  document.getElementById() is used to access and update HTML elements dynamically.
- **Event Handling** –
  addEventListener("submit", ...) handles form submission.
- **Objects** –
  Student details are stored as objects (name, roll, email, number).
- **Arrays** –
  An array is used to store multiple student objects.
- **Functions** –
  Separate functions are used for add, display, delete, and update operations.
- **Loops (forEach)** –
  Used to iterate through student records and display them.
- **Conditional Statements (if–else)** –
  Used for validation and error handling.
- **Template Literals** –
  Used to display student data dynamically in HTML.
- **Form Handling** –
  preventDefault() prevents page reload and reset() clears form inputs.
- **Type Conversion** –
  parseInt() converts input values into numbers.

Code:

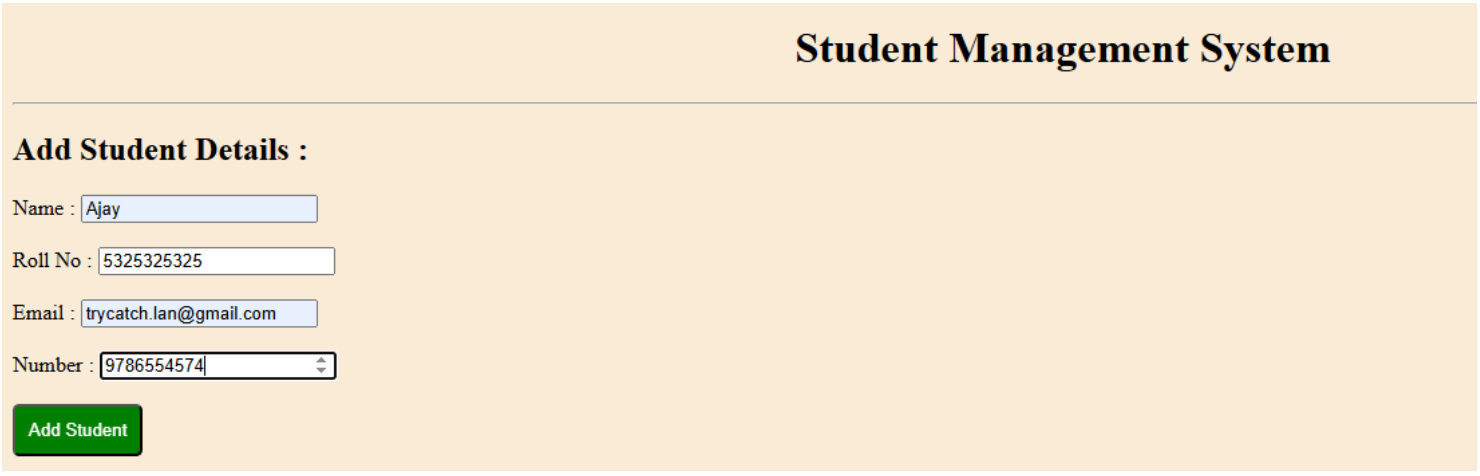
```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Student Management System</title>
   <link rel="stylesheet" href="style.css">
</head>
<body>
<h1 class="heading">Student Management System</h1>
<hr>
<h2>Add Student Details :</h2>
<form id="studentForm">
```

```
    <label>Name :</label>
    <input type="text" id="name"><br><br>
    <label>Roll No :</label>
    <input type="number" id="roll"><br><br>
    <label>Email :</label>
    <input type="text" id="email"><br><br>
    <label>Number :</label>
    <input type="number" id="number"><br><br>
    <button type="submit" class="addbtn">Add Student</button>
</form>
<br>
<button onclick="getStudent()" class="getbtn">Get Student Details</button><br>
<br>
    <label>Enter the id for Deletion (index - 0 )  :</label>
    <input type="number" id="del"><br>
    <br>
<button onclick="deleteStudent()" class="delbtn">Delete Student </button>
<hr>
<h3>Update Student Number</h3>
<label>Enter Student Index :</label>
<input type="number" id="updateIndex"><br><br>
<label>New Phone Number :</label>
<input type="number" id="updateNumber"><br><br>
<button onclick="updateStudent()" class="updbtn">Update Student</button>
<hr>
<div id="studentList"></div>
<script src="script.js"></script>
</body>
</html>

let name1 = document.getElementById("name");
let roll = document.getElementById("roll");
let email = document.getElementById("email");
let number = document.getElementById("number");
let studentList = document.getElementById("studentList");
let del = document.getElementById("del");

const studentArray = [];

document.getElementById("studentForm").addEventListener("submit", function (e) {
  e.preventDefault();

  const student = {
    stuname: name1.value,
    stuRoll: roll.value,
    stuEmail: email.value,
```

```
    stuNumber: number.value
  };

  studentArray.push(student);
  this.reset();
  alert("Student Added Successfully");
});

function getStudent() {
  studentList.innerHTML = "";

  if (studentArray.length === 0) {
    studentList.innerHTML = "<p>No student data available</p>";
    return;
  }

  studentArray.forEach((stu, index) => {
    const div = document.createElement("div");
    div.innerHTML = `
      <h3>Student ${index }</h3>
      <p><strong>Name:</strong> ${stu.stuname}</p>
      <p><strong>Roll:</strong> ${stu.stuRoll}</p>
      <p><strong>Email:</strong> ${stu.stuEmail}</p>
      <p><strong>Number:</strong> ${stu.stuNumber}</p>
      <hr>
    `;
    studentList.appendChild(div);
  });
}
function deleteStudent() {
  const delIndex = parseInt(document.getElementById("del").value);

  if (delIndex >= 0 && delIndex < studentArray.length) {
    const removed = studentArray.splice(delIndex, 1);
    alert(`Deleted student: ${removed[0].stuname}`);
    getStudent();
  } else {
    alert("Error: Student index not found.");
  }
  document.getElementById("del").value = "";
}
  function updateStudent() {
  const index = parseInt(document.getElementById("updateIndex").value);
  const newNumber = document.getElementById("updateNumber").value;

  if ( index >= 0 && index < studentArray.length) {
```

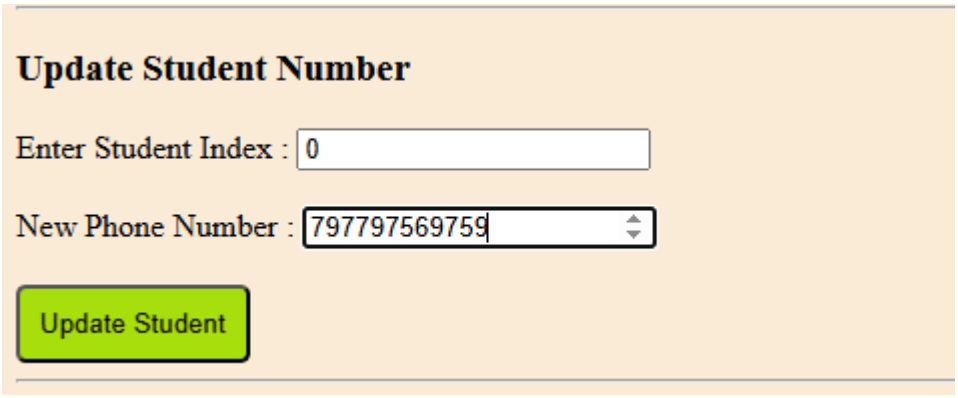
```
    studentArray[index].stuNumber = newNumber;
    alert(`Updated number for ${studentArray[index].stuname}`);
    getStudent();
  } else {
    alert("Error: Student index not found.");
  }
  document.getElementById("updateIndex").value = "";
  document.getElementById("updateNumber").value = "";
}
```

Output:

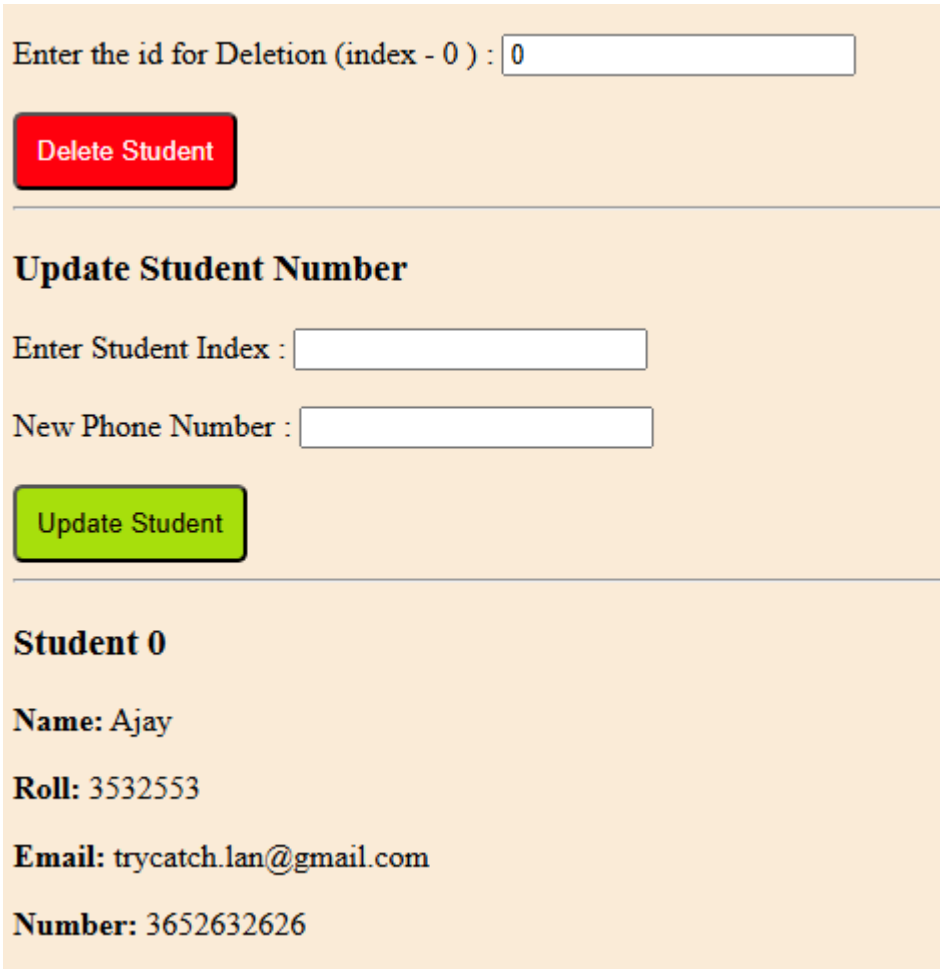


Figure(a)

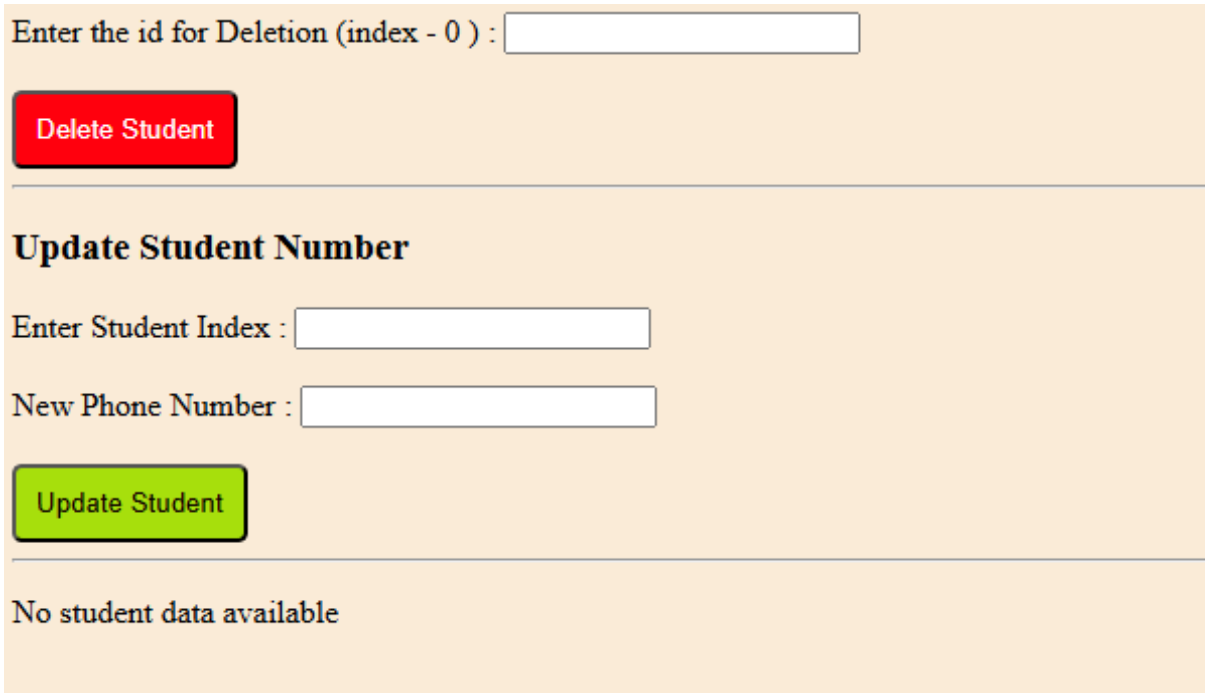
Figure(b)



Figure(c)



Figure(d)



Figure(e)

Result:

Thus, the student management system is done and the output is taken successfully by performing add, delete, update, and display operations.