server.js:

```
const express = require('express');
const bodyParser = require('body-parser');
const fs = require('fs');
const cors = require('cors')
const path = require('path')
const app = express();
const DATA_FILE = path.join(__dirname, 'students.json');
app.use(cors());
app.use(bodyParser.json());
app.use(express.static('public'));
function readStudents() {
  if (!fs.existsSync(DATA_FILE)) return [];
  const data = fs.readFileSync(DATA_FILE, 'utf8');
  return JSON.parse(data || '[]');
}
function writeStudents(data) {
  fs.writeFileSync(DATA_FILE, JSON.stringify(data, null, 2));
}
app.get('/students', (req, res) => {
  const students = readStudents();
  res.json(students);
});
app.post('/students', (req, res) => {
  const { name, marks } = req.body;
  const students = readStudents();
  const newStudent = {
    id: Date.now(),
    name,
    marks: Number(marks)
  };
  students.push(newStudent);
  writeStudents(students);
  res.json(newStudent);
});
app.delete('/students/:id', (req, res) => {
  const id = parseInt(req.params.id);
  let students = readStudents();
  students = students.filter(s => s.id !== id);
  writeStudents(students);
  res.json({ message: 'Deleted successfully' });
});
app.listen(3000, () => console.log('🚀 Server running at http://localhost:3000'));
```

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Student Management System</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>🎓 Student Management System</h1>
<div>
  <h3>🔍 Search Student Marks</h3>
  <input type="text" id="searchName" placeholder="Enter student name">
  <button id="searchBtn">Search</button>
```

```html
  <p id="searchResult"></p>
</div>
<hr>
<form id="studentForm">
  <h3>➕ Add New Student</h3>
  <input type="text" id="name" placeholder="Enter student name" required>
  <input type="number" id="marks" placeholder="Enter marks" required>
  <button type="submit">Add Student</button>
</form>
  <table>
    <thead>
      <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Marks</th>
        <th>Action</th>
      </tr>
    </thead>
    <tbody id="studentTable"></tbody>
  </table>
  <script src="script.js"></script>
</body>
</html>
```

Style.css:

```css
body {
  font-family: Arial, sans-serif;
  background: #f7f9fc;
  text-align: center;
  padding: 20px;
```

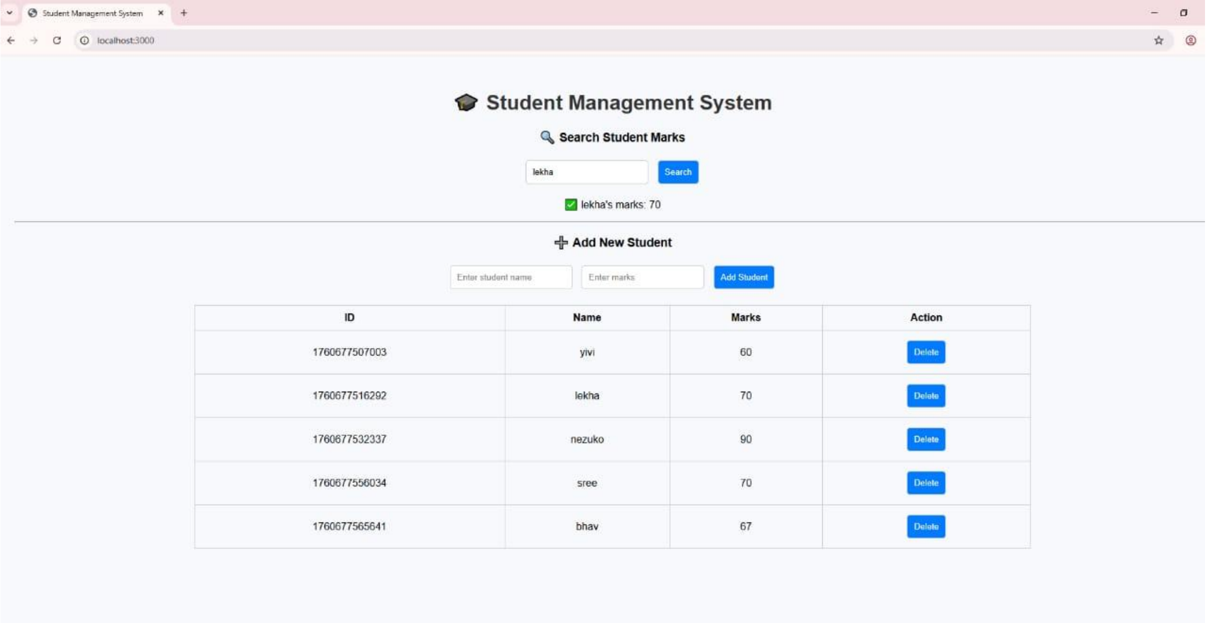
```
}
h1 {
  color: #333;
}
form {
  margin-bottom: 20px;
}
input, button {
  padding: 10px;
  margin: 5px;
  border-radius: 5px;
  border: 1px solid #ccc;
}
button {
  background-color: #007BFF;
  color: white;
  cursor: pointer;
}
button:hover {
  background-color: #0056b3;
}
table {
  margin: 0 auto;
  border-collapse: collapse;
  width: 70%;
}
th, td {
  border: 1px solid #ccc;
  padding: 10px;
```

```
}
```

Script.js:

```
const form = document.getElementById('studentForm');
const tableBody = document.getElementById('studentTable');
const searchInput = document.getElementById('searchName');
const searchBtn = document.getElementById('searchBtn');
const searchResult = document.getElementById('searchResult');
async function loadStudents() {
  const res = await fetch('/students');
  const students = await res.json();
  tableBody.innerHTML = '';
  students.forEach(s => {
    const row = `<tr>
      <td>${s.id}</td>
      <td>${s.name}</td>
      <td>${s.marks}</td>
      <td><button onclick="deleteStudent(${s.id})">Delete</button></td>
    </tr>`;
    tableBody.innerHTML += row;
  });
}
form.addEventListener('submit', async (e) => {
  e.preventDefault();
  const name = document.getElementById('name').value;
  const marks = document.getElementById('marks').value;
  await fetch('/students', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify({ name, marks })
  });
  form.reset();
  loadStudents();
});
async function deleteStudent(id) {
  await fetch(`/students/${id}`, { method: 'DELETE' });
  loadStudents();
}
searchBtn.addEventListener('click', async () => {
  const name = searchInput.value.trim().toLowerCase();
  if (!name) {
    searchResult.textContent = 'Please enter a student name.';
    return;
  }
  const res = await fetch('/students');
  const students = await res.json();
  const found = students.find(s => s.name.toLowerCase() === name);
  if (found) {
    searchResult.textContent = `✅ ${found.name}'s marks: ${found.marks}`;
  } else {
    searchResult.textContent = '❌ Student not found.';
  }
});
loadStudents();
```

OUTPUT:



🎓 Student Management System

🔍 Search Student Marks

lekha

✅ lekha's marks: 70

➕ Add New Student

ID	Name	Marks	Action
1760677507003	yivi	60	Delete
1760677516292	lekha	70	Delete
1760677532337	nezuko	90	Delete
1760677556034	sree	70	Delete
1760677565641	bhav	67	Delete

RESULT: