Folder Structure

```
express-mongo-crud/
│
├── server.js
├── models/
│   └── Student.js
├── public/
│   └── index.html
└── package.json
```

Install Dependencies

Open VS Code terminal:

```
mkdir express-mongo-crud
cd express-mongo-crud
npm init -y
npm install express mongoose body-parser cors nodemon
```

Edit `package.json` and add:

```
"scripts": {
  "start": "nodemon server.js"
}
```

Create Mongoose Schema — `models/Student.js`

```
// models/Student.js
```

```
const mongoose = require('mongoose');


const studentSchema = new
mongoose.Schema({
  name: { type: String, required:
true },
  age: { type: Number, required: true
},
  course: { type: String, required:
true }
});


module.exports =
mongoose.model('Student',
studentSchema);
```

Server Code — `server.js`

```
// server.js
const express = require('express');
const mongoose = require('mongoose');
```

```
const bodyParser =
require('body-parser');
const cors = require('cors');
const path = require('path');
const Student =
require('./models/Student');

const app = express();
const PORT = 5000;

// Middleware
app.use(bodyParser.json());
app.use(cors());
app.use(express.static(path.join(__dirnam
e, 'public')));

// MongoDB Connection
mongoose.connect('mongodb://127.0.0.1:270
17/studentDB', {
  useNewUrlParser: true,
```

```
  useUnifiedTopology: true,
});
mongoose.connection.on('connected', () =>
console.log('MongoDB Connected'));
mongoose.connection.on('error', (err) =>
console.log('MongoDB Error:', err));

// ROUTES

// Create
app.post('/students', async (req, res) =>
{
  try {
    const student = new
Student(req.body);
    const saved = await student.save();
    res.status(201).json(saved);
  } catch (err) {
    res.status(400).json({ error:
err.message });
  }
```

```
});

// Read all
app.get('/students', async (req, res) =>
{
  try {
    const students = await
Student.find();
    res.json(students);
  } catch (err) {
    res.status(500).json({ error:
err.message });
  }
});

// Update by ID
app.put('/students/:id', async (req, res)
=> {
  try {
```

```
    const updated = await
Student.findByIdAndUpdate(req.params.id,
req.body, { new: true });
    res.json(updated);
  } catch (err) {
    res.status(400).json({ error:
err.message });
  }
});

// Delete by ID
app.delete('/students/:id', async (req,
res) => {
  try {
    await
Student.findByIdAndDelete(req.params.id);
    res.json({ message: 'Deleted
successfully' });
  } catch (err) {
    res.status(500).json({ error:
err.message });
```

```
  }
});

// Server start
app.listen(PORT, () => console.log(`
Server running on
http://localhost:${PORT}`));
```

Frontend — `public/index.html`

This version displays **all records in a table**, allows **add**, **update**, and **delete**.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Student CRUD App</title>
  <style>
    body { font-family: Arial; background: #f4f4f4; margin:
30px; }
    input, button { padding: 8px; margin: 4px; }
    button { cursor: pointer; }
    table { border-collapse: collapse; width: 100%; margin-top:
20px; background: white; }
    th, td { border: 1px solid #ddd; padding: 8px; text-align:
left; }
    th { background-color: #009879; color: white; }
```

```
  </style>
</head>
<body>
  <h1>🎓 Student Management (MongoDB + Express)</h1>

  <div>
    <input id="name" placeholder="Name">
    <input id="age" type="number" placeholder="Age">
    <input id="course" placeholder="Course">
    <button onclick="addStudent()">Add Student</button>
  </div>

  <table>
    <thead>
      <tr>
        <th>Name</th><th>Age</th><th>Course</th><th>Actions</th>
      </tr>
    </thead>
    <tbody id="studentList"></tbody>
  </table>

  <script>
    const apiUrl = '/students';

    async function loadStudents() {
      const res = await fetch(apiUrl);
      const students = await res.json();
```

```
      const tbody = document.getElementById('studentList');
      tbody.innerHTML = students.map(s => `
        <tr>
          <td>${s.name}</td>
          <td>${s.age}</td>
          <td>${s.course}</td>
          <td>
            <button onclick="editStudent('${s._id}',
'${s.name}', ${s.age}, '${s.course}')">Edit</button>
            <button
onclick="deleteStudent('${s._id}')">Delete</button>
          </td>
        </tr>
      `).join('');
    }

    async function addStudent() {
      const name = document.getElementById('name').value;
      const age = document.getElementById('age').value;
      const course = document.getElementById('course').value;

      await fetch(apiUrl, {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ name, age, course })
      });
      loadStudents();
    }
```

```
    async function deleteStudent(id) {
      await fetch(`${apiUrl}/${id}`, { method: 'DELETE' });
      loadStudents();
    }

    async function editStudent(id, name, age, course) {
      const newName = prompt("Edit name:", name);
      const newAge = prompt("Edit age:", age);
      const newCourse = prompt("Edit course:", course);

      await fetch(`${apiUrl}/${id}`, {
        method: 'PUT',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ name: newName, age: newAge,
course: newCourse })
      });
      loadStudents();
    }

    loadStudents();
  </script>
</body>
</html>
```