

Assignment 8

Server.js –

```
import express from "express";
import cors from "cors";
import { MongoClient, ObjectId } from "mongodb";

const app = express();
app.use(cors());
app.use(express.json());

const PORT = 4000;
const DBCONNECT = "mongodb://localhost:27017";
let db;

const connectDB = async () => {
  try {
    const client = new MongoClient(DBCONNECT);
    await client.connect();
    db = client.db("colleges");
    console.log(" Database Connected Successfully");
  } catch (error) {
    console.error(" Database Connection Error:", error);
    process.exit(1);
  }
};

app.get("/getstudent", async (req, res) => {
  const students = await db.collection("student_table").find({}).toArray();
  res.json(students);
});
```

```
});
```

```
app.post("/addstudent", async (req, res) => {  
  try {  
    const { name, email, rollNo, mobile, degree, year } = req.body;  
    if (!name || !email || !rollNo || !mobile || !degree || !year) {  
      return res.status(400).json({ message: "All fields are required" });  
    }  
    const result = await db.collection("student_table").insertOne(req.body);  
    res.status(201).json({ _id: result.insertedId, ...req.body });  
  } catch (error) {  
    res.status(500).json({ message: "Internal Server Error" });  
  }  
});
```

```
app.put("/updatestudent/:id", async (req, res) => {  
  try {  
    const { id } = req.params;  
    const updatedStudent = { ...req.body };  
    delete updatedStudent._id;  
  
    const result = await db.collection("student_table").updateOne(  
      { _id: new ObjectId(id) },  
      { $set: updatedStudent }  
    );  
  
    res.json({ message: "Student updated successfully", modifiedCount: result.modifiedCount });  
  } catch (error) {  
    res.status(500).json({ message: "Internal Server Error" });  
  }  
});
```

```
app.delete("/deletestudent/:id", async (req, res) => {  
  await db.collection("student_table").deleteOne({ _id: new ObjectId(req.params.id) });  
  res.json({ message: "Student deleted successfully" });  
});  
  
connectDB().then(() => app.listen(PORT, () => console.log(`    Server running on port ${PORT}`)));
```

Header.css-

```
.header {  
  color: red;  
  font-size: larger;  
  width: 100%;  
  text-align: center;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  height: 60px;  
  background-color: #f8f9fa;  
  font-weight: bold;  
}
```

```
.heading {  
  color: red;  
  font-size: 24px;  
  font-weight: bold;  
  text-align: center;  
  margin-top: 20px;  
}
```

Header.js-

```
import React from "react";

import { Navbar, Nav, Container } from "react-bootstrap";
import { Link } from "react-router-dom";

function Header() {
  return (
    <Navbar bg="dark" variant="dark" expand="lg">
      <Container>
        <Navbar.Brand as={Link} to="/">Student App</Navbar.Brand>
        <Nav className="ms-auto">
          <Nav.Link as={Link} to="/">Register</Nav.Link>
          <Nav.Link as={Link} to="/records">Registered Records</Nav.Link>
        </Nav>
      </Container>
    </Navbar>
  );
}

export default Header;
```

RegisteredRecord.js –

```
import React, { useState, useEffect } from "react";
import { useNavigate } from "react-router-dom";
import { Button, Table, Form, Container } from "react-bootstrap";
import { getStudentList, deleteStudent } from "../services/service";
import ViewDetails from "../ViewDetails";

function RegisteredRecords() {
```

```
const [students, setStudents] = useState([]);
const [filteredStudents, setFilteredStudents] = useState([]);
const [searchQuery, setSearchQuery] = useState("");
const [modalShow, setModalShow] = useState(false);
const [selectedStudent, setSelectedStudent] = useState(null);
const navigate = useNavigate();
```

```
useEffect(() => {
  fetchStudentData();
}, []);
```

```
const fetchStudentData = async () => {
  const data = await getStudentList();
  setStudents(data);
  setFilteredStudents(data);
};
```

```
const handleDelete = async (id) => {
  if (window.confirm("Are you sure?")) {
    await deleteStudent(id);
    fetchStudentData();
  }
};
```

```
const handleEdit = (student) => {
  navigate("/", { state: { student } });
};
```

```
const handleView = (student) => {
  setSelectedStudent(student);
  setModalShow(true);
};
```

```
};
```

```
const handleSearch = (e) => {  
  const query = e.target.value.toLowerCase();  
  setSearchQuery(query);  
  setFilteredStudents(  
    students.filter((student) =>  
      Object.values(student).some((val) => val.toString().toLowerCase().includes(query))  
    )  
  );  
};  
};
```

```
return (  
  <Container>  
    <h3>Registered Students</h3>  
  
    <Form.Group className="mb-3">  
      <Form.Control type="text" placeholder="Search..." value={searchQuery}  
onChange={handleSearch} />  
    </Form.Group>
```

```
<Table bordered>  
  <thead>  
    <tr>  
      <th>Name</th>  
      <th>Email</th>  
      <th>Roll No</th>  
      <th>Mobile</th>  
      <th>Degree</th>  
      <th>Year</th>  
      <th>Actions</th>
```

```

    </tr>
  </thead>
  <tbody>
    {filteredStudents.map((student) => (
      <tr key={student._id}>
        <td>{student.name}</td>
        <td>{student.email}</td>
        <td>{student.rollNo}</td>
        <td>{student.mobile}</td>
        <td>{student.degree}</td>
        <td>{student.year}</td>
        <td>
          <Button variant="info" onClick={() => handleView(student)}>View</Button>
          <Button variant="warning" className="ms-2" onClick={() =>
handleEdit(student)}>Edit</Button>
          <Button variant="danger" className="ms-2" onClick={() =>
handleDelete(student._id)}>Delete</Button>
        </td>
      </tr>
    )
  )}
  </tbody>
</Table>

<ViewDetails show={modalShow} handleClose={() => setModalShow(false)}
student={selectedStudent} />

</Container>

);
}

export default RegisteredRecords;

```

RegistrationForm.js –

```
import React, { useState, useEffect } from "react";
import { useNavigate, useLocation } from "react-router-dom";
import { Col, Form, Row, Button, Container } from "react-bootstrap";
import { addStudent, updateStudent } from "../services/service";
```

```
function RegistrationForm() {
  const navigate = useNavigate();
  const location = useLocation();
  const studentToEdit = location.state?.student || null;
```

```
  const [formData, setFormData] = useState({
    name: "",
    email: "",
    rollNo: "",
    mobile: "",
    degree: "",
    year: "",
  });
```

```
  const [editId, setEditId] = useState(null);
```

```
  useEffect(() => {
    if (studentToEdit) {
      setFormData(studentToEdit);
      setEditId(studentToEdit._id);
    }
  }, [studentToEdit]);
```

```
  const handleChange = (e) => {
```



```
setFormData({ ...formData, [e.target.name]: e.target.value });  
};
```

```
const handleSubmit = async (e) => {  
  e.preventDefault();  
  if (editId) {  
    await updateStudent(editId, formData);  
  } else {  
    await addStudent(formData);  
  }  
  handleReset();  
  navigate("/records");  
};
```

```
const handleReset = () => {  
  setFormData({ name: "", email: "", rollNo: "", mobile: "", degree: "", year: "" });  
  setEditId(null);  
};
```

```
return (  
  <Container>  
    <h3>{editId ? "Edit Student" : "Student Registration"}</h3>  
    <Form onSubmit={handleSubmit}>  
      {Object.keys(formData).map((key) => (  
        <Form.Group as={Row} className="mb-3" key={key}>  
          <Form.Label column sm="2">{key.charAt(0).toUpperCase() + key.slice(1)}</Form.Label>  
          <Col sm="10">  
            <Form.Control type="text" name={key} value={formData[key]} onChange={handleChange} />  
          </Col>  
        </Form.Group>  
      ))}
```

```

        <Button type="submit">{editId ? "Update" : "Submit"}</Button>

        <Button type="button" variant="secondary" onClick={handleReset} className="ms
2">Reset</Button>

    </Form>

</Container>

);
}

export default RegistrationForm;

```

ViewDetails.js-

```

// import React from "react";

import { Modal, Button } from "react-bootstrap";

function ViewDetails({ show, handleClose, student }) {
  if (!student) return null;

  return (
    <Modal show={show} onHide={handleClose} centered>
      <Modal.Header closeButton>
        <Modal.Title>Student Details</Modal.Title>
      </Modal.Header>
      <Modal.Body>
        {Object.entries(student).map(([key, value]) => (
          <p key={key}><strong>{key}</strong> {value}</p>
        ))}
      </Modal.Body>
      <Modal.Footer>
        <Button variant="secondary" onClick={handleClose}>Close</Button>
      </Modal.Footer>
    </Modal>
  );
}

```

```
    </Modal>
  );
}
```

```
export default ViewDetails;
```

App.js –

```
import React, { useState } from "react";
import { BrowserRouter as Router, Route, Routes } from "react-router-dom";
import RegistrationForm from "../component/RegistrationForm";
import RegisteredRecords from "../component/RegisteredRecord";
import Header from "../component/Header";
import "bootstrap/dist/css/bootstrap.min.css";
import "../App.css";

function App() {
  const [editStudent, setEditStudent] = useState(null);

  return (
    <Router>
      <Header />
      <div className="container mt-4">
        <Routes>
          <Route path="/" element={
            <RegistrationForm editStudent={editStudent}
              setEditStudent={setEditStudent} />
          } />
          <Route path="/records" element={
            <RegisteredRecords onEdit={setEditStudent} />
          } />
        </Routes>
      </div>
    </Router>
  );
}
```

```
}
```

```
export default App;
```

service.js-

```
import axios from "axios";
```

```
const BASE_URL = "http://localhost:4000";
```

```
export const getStudentList = async () => {
```

```
  try {
```

```
    const response = await axios.get(`${BASE_URL}/getstudent`);
```

```
    return response.data;
```

```
  } catch (error) {
```

```
    console.error("Error fetching student list:", error);
```

```
    return [];
```

```
  }
```

```
};
```

```
export const addStudent = async (payload) => {
```

```
  try {
```

```
    const response = await axios.post(`${BASE_URL}/addstudent`, payload);
```

```
    return response.data;
```

```
  } catch (error) {
```

```
    console.error("Error adding student:", error);
```

```
    return null;
```

```
  }
```

```
};
```

```
export const updateStudent = async (id, payload) => {
```

```

    try {
      const response = await axios.put(`${BASE_URL}/updatestudent/${id}`, payload);
      return response.data;
    } catch (error) {
      console.error("Error updating student:", error);
      return null;
    }
  };

export const deleteStudent = async (id) => {
  try {
    const response = await axios.delete(`${BASE_URL}/deletestudent/${id}`);
    return response.data;
  } catch (error) {
    console.error("Error deleting student:", error);
    return null;
  }
};

```

Frontend terminal –

PS C:\Users\Madhura\Desktop\UI_training> npm start

npm error Missing script: "start"

npm error

npm error Did you mean one of these?

npm error npm star # Mark your favorite packages

npm error npm stars # View packages marked as favorites

npm error

npm error To see a list of scripts, run:

npm error npm run

npm error A complete log of this run can be found in: C:\Users\Madhura\AppData\Local\npm

cache_logs\2025-03-27T17_53_02_412Z-debug-0.log

Compiled successfully!

You can now view demo in the browser.

Local: <http://localhost:3000>

On Your Network: <http://192.168.56.1:3000>

Note that the development build is not optimized.

To create a production build, use `npm run build`.

webpack compiled successfully

Server Terminal -

```
PS C:\Users\Madhura\Desktop\UI_training\demo> cd server
```

```
PS C:\Users\Madhura\Desktop\UI_training\demo\server> node server.js
```

```
(node:2884) [MODULE_TYPELESS_PACKAGE_JSON] Warning: Module type of
```

```
f
```

```
ile:///C:/Users/Madhura/Desktop/UI_training/demo/server/server.js is not specified and it doesn't  
parse as CommonJS.
```

```
Reparsing as ES module because module syntax was detected. This incurs a performance overhead.
```

```
To eliminate this warning, add "type": "module" to
```

```
C:\Users\Madhura\Desktop\UI_training\demo\package.json.
```

```
(Use `node --trace-warnings ...` to show where the warning was created)
```

```
Database Connected Successfully
```

```
Server running on port 4000
```