Web Developer INTERN - TASK 5
COMPLETE STUDENT REGISTRATION FORM

🎯 Objective:

Build a complete **Student Registration System** with a frontend form that submits data to the backend and saves it to **MongoDB**, including file upload functionality.

✅ Project Structure

```
bash
CopyEdit
student-registration/
├── client/              # React frontend
└── server/              # Node.js backend with MongoDB
```

🧩 1. BACKEND (Node.js + Express + MongoDB)

➤ 1.1 Initialize Backend

```
mkdir server
cd server
npm init -y
npm install express mongoose cors multer body-parser
```

➤ 1.2 Create `server.js`

```
// server/server.js
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");
const multer = require("multer");
const bodyParser = require("body-parser");
const path = require("path");

const app = express();
app.use(cors());
app.use(bodyParser.json());
app.use('/uploads', express.static('uploads'));

mongoose.connect("mongodb://127.0.0.1:27017/studentDB", {
    useNewUrlParser: true,
    useUnifiedTopology: true,
}).then(() => console.log("MongoDB Connected"))
  .catch((err) => console.log(err));

// Multer config
const storage = multer.diskStorage({
  destination: (req, file, cb) => cb(null, "uploads/"),
  filename: (req, file, cb) => cb(null, Date.now() + "-" +
file.originalname),
});
const upload = multer({ storage });
```

```
// Mongoose Schema
const studentSchema = new mongoose.Schema({
  name: String,
  email: String,
  phone: String,
  ageGroup: String,
  gender: String,
  courses: [String],
  qualification: String,
  learningMode: String,
  comments: String,
  document: String
});

const Student = mongoose.model("Student", studentSchema);

// POST route to handle form
app.post("/register", upload.single("document"), async (req, res) => {
  try {
    const {
      name, email, phone, ageGroup, gender,
      qualification, learningMode, comments
    } = req.body;

    const courses = req.body["courses[]"] || req.body.courses;
    const documentPath = req.file ? req.file.path : "";

    const student = new Student({
      name, email, phone, ageGroup, gender,
      courses: Array.isArray(courses) ? courses : [courses],
      qualification, learningMode, comments,
      document: documentPath
    });

    await student.save();
    res.status(201).json({ message: "Student Registered Successfully" });
  } catch (err) {
    console.error(err);
    res.status(500).json({ message: "Error registering student" });
  }
});

app.listen(5000, () => {
  console.log("Server running on port 5000");
});
```

🧩 2. FRONTEND (React)

➤ 2.1 Setup

```
npx create-react-app client
cd client
npm install axios
```

➤ 2.2 Create `StudentRegistrationForm.js`

```
// client/src/StudentRegistrationForm.js
import React, { useState } from "react";
import axios from "axios";

function StudentRegistrationForm() {
  const [formData, setFormData] = useState({
    name: "",
    email: "",
    phone: "",
    ageGroup: "",
    gender: "",
    courses: [],
    qualification: "",
    learningMode: "",
    comments: "",
  });
  const [file, setFile] = useState(null);

  const handleChange = (e) => {
    const { name, value, type } = e.target;
    if (type === "checkbox") {
      const updatedCourses = [...formData.courses];
      if (e.target.checked) {
        updatedCourses.push(value);
      } else {
        const index = updatedCourses.indexOf(value);
        if (index !== -1) updatedCourses.splice(index, 1);
      }
      setFormData({ ...formData, courses: updatedCourses });
    } else {
      setFormData({ ...formData, [name]: value });
    }
  };

  const handleFileChange = (e) => {
    setFile(e.target.files[0]);
  };

  const handleSubmit = async (e) => {
    e.preventDefault();
    const submitData = new FormData();

    for (const key in formData) {
      if (Array.isArray(formData[key])) {
        formData[key].forEach((item) => submitData.append(`${key}[]`,
item));
      } else {
        submitData.append(key, formData[key]);
      }
    }

    if (file) {
      submitData.append("document", file);
    }

    try {
      await axios.post("http://localhost:5000/register", submitData, {
        headers: { "Content-Type": "multipart/form-data" },
      });
      alert("Student Registered Successfully!");
    } catch (error) {
      console.error(error);
      alert("Failed to register student.");
    }
  };

  return (
    <form onSubmit={handleSubmit} className="form-container">
      <h2>Student Registration</h2>
      <input type="text" name="name" placeholder="Full Name"
onChange={handleChange} required />
```

```
      <input type="email" name="email" placeholder="Email Address"
onChange={handleChange} required />
      <input type="text" name="phone" placeholder="Phone Number"
onChange={handleChange} required />

      <select name="ageGroup" onChange={handleChange} required>
        <option value="">Select Age Group</option>
        <option value="Under 18">Under 18</option>
        <option value="18 to 25">18 to 25</option>
        <option value="26 to 35">26 to 35</option>
        <option value="36+">36+</option>
      </select>

      <label>Gender:</label>
      <label><input type="radio" name="gender" value="Male"
onChange={handleChange} /> Male</label>
      <label><input type="radio" name="gender" value="Female"
onChange={handleChange} /> Female</label>
      <label><input type="radio" name="gender" value="Other"
onChange={handleChange} /> Other</label>

      <label>Courses Interested:</label>
      <label><input type="checkbox" name="courses" value="Web Development"
onChange={handleChange} /> Web Development</label>
      <label><input type="checkbox" name="courses" value="AI & ML"
onChange={handleChange} /> AI & ML</label>
      <label><input type="checkbox" name="courses" value="DevOps"
onChange={handleChange} /> DevOps</label>

      <select name="qualification" onChange={handleChange} required>
        <option value="">Select Qualification</option>
        <option value="A/L">A/L</option>
        <option value="Diploma">Diploma</option>
        <option value="Undergraduate">Undergraduate</option>
        <option value="Graduate">Graduate</option>
      </select>

      <select name="learningMode" onChange={handleChange} required>
        <option value="">Learning Mode</option>
        <option value="Online">Online</option>
        <option value="Physical">Physical</option>
        <option value="Hybrid">Hybrid</option>
      </select>

      <textarea name="comments" placeholder="Additional Comments"
onChange={handleChange}></textarea>

      <label>Upload CV/Certificate:</label>
      <input type="file" name="document" onChange={handleFileChange}
accept=".pdf,.doc,.jpg,.png" />

      <button type="submit">Submit</button>
    </form>
  );
}

export default StudentRegistrationForm;
```

➤ 2.3 Display in `App.js`

```
// client/src/App.js
import React from "react";
import StudentRegistrationForm from "./StudentRegistrationForm";
import './App.css';

function App() {
  return (
    <div className="App">
      <StudentRegistrationForm />
    </div>
  );
}

export default App;
```

➤ 2.4 Style (Optional)

```
/* client/src/App.css */
.form-container {
  max-width: 600px;
  margin: 30px auto;
  padding: 30px;
  background: #f7f7f7;
  border-radius: 10px;
}

input, select, textarea {
  display: block;
  margin-bottom: 15px;
  padding: 10px;
  width: 100%;
}

button {
  padding: 10px 20px;
  background: #28a745;
  color: white;
  border: none;
  border-radius: 5px;
}
```

🚀 FINAL STEPS TO RUN

✅ Start MongoDB

```
mongod
```

✅ Run Backend

```
cd server
node server.js
```

✅ Run Frontend

```
cd client
npm start
```

✅ DONE!

Once running:

- Access: `http://localhost:3000`
- Submit the form
- Check saved data in `studentDB` in MongoDB
- Uploaded files will be in `server/uploads/`