```
// === Job Listings Board (MERN Stack) ===

// === Step 1: Create Project Directories ===
// mkdir job-backend
// cd job-backend

// === Step 2: Initialize Backend Project ===
// npm init -y
// npm install express mongoose mongodb cors

// === Step 3: Create server.js ===
const express = require('express');
const mongoose = require('mongoose');
const cors = require('cors');

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

mongoose.connect('mongodb://localhost:27017/jobboard', {
  useNewUrlParser: true,
  useUnifiedTopology: true
});

const db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error:'));
db.once('open', () => console.log('Connected to MongoDB'));

const jobSchema = new mongoose.Schema({
  position: String,
  company: String,
  location: String,
  type: String,
  description: String
});

const Job = mongoose.model('Job', jobSchema);

app.get('/jobs', async (req, res) => {
  try {
    const jobs = await Job.find();
    res.json(jobs);
  } catch (error) {
    res.status(500).send(error.message);
  }
});

app.post('/jobs', async (req, res) => {
  try {
    const newJob = new Job(req.body);
    await newJob.save();
    res.status(201).json(newJob);
  } catch (error) {
    res.status(400).send(error.message);
  }
});

app.put('/jobs/:id', async (req, res) => {
  try {
    const updatedJob = await Job.findByIdAndUpdate(req.params.id, req.body, { new: true });
    res.json(updatedJob);
  } catch (error) {
    res.status(400).send(error.message);
  }
});

app.delete('/jobs/:id', async (req, res) => {
  try {
    await Job.findByIdAndDelete(req.params.id);
    res.status(204).send();
  } catch (error) {
    res.status(500).send(error.message);
  }
});

const PORT = 4000;
app.listen(PORT, () => console.log(`Server running on port ${PORT}`));

// === Step 4: Set up Frontend ===
// Open new terminal:
// npx create-react-app job-client
// cd job-client

// === Replace App.js with the following ===
import React, { useEffect, useState } from 'react';
import './App.css';

const App = () => {
  const [jobs, setJobs] = useState([]);
  const [form, setForm] = useState({ position: '', company: '', location: '', type: '', description: '' });
  const [editJob, setEditJob] = useState(null);
  const [errors, setErrors] = useState({});

  const fetchJobs = () => {
    fetch('http://localhost:4000/jobs')
      .then(res => res.json())
      .then(setJobs);
  };

  useEffect(() => { fetchJobs(); }, []);

  const handleChange = (e) => {
    const { name, value } = e.target;
    const currentForm = editJob ? editJob : form;
    const updatedForm = { ...currentForm, [name]: value };
    if (editJob) setEditJob(updatedForm);
    else setForm(updatedForm);
    setErrors({ ...errors, [name]: '' });
  };

  const validate = (data) => {
    const errs = {};
    ['position', 'company', 'location', 'type', 'description'].forEach(key => {
      if (!data[key]) errs[key] = `${key} is required.`;
    });
    setErrors(errs);
    return Object.keys(errs).length === 0;
  };

  const addJob = () => {
    if (!validate(form)) return;
    fetch('http://localhost:4000/jobs', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(form)
    })
      .then(res => res.json())
      .then(() => {
        fetchJobs();
        setForm({ position: '', company: '', location: '', type: '', description: '' });
      });
  };

  const updateJob = () => {
    if (!validate(editJob)) return;
    fetch(`http://localhost:4000/jobs/${editJob._id}`, {
      method: 'PUT',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(editJob)
    })
      .then(() => {
        fetchJobs();
        setEditJob(null);
      });
  };

  const deleteJob = (id) => {
    if (window.confirm('Are you sure?')) {
      fetch(`http://localhost:4000/jobs/${id}`, { method: 'DELETE' })
        .then(fetchJobs);
    }
  };

  return (
    <div className="container">
      <h1 className="title">Job Listings Board</h1>
      {/* Job Table and Form Rendering */}
    </div>
  );
};
```

```
export default App;

// === Step 5: Start React ===
// npm start
```