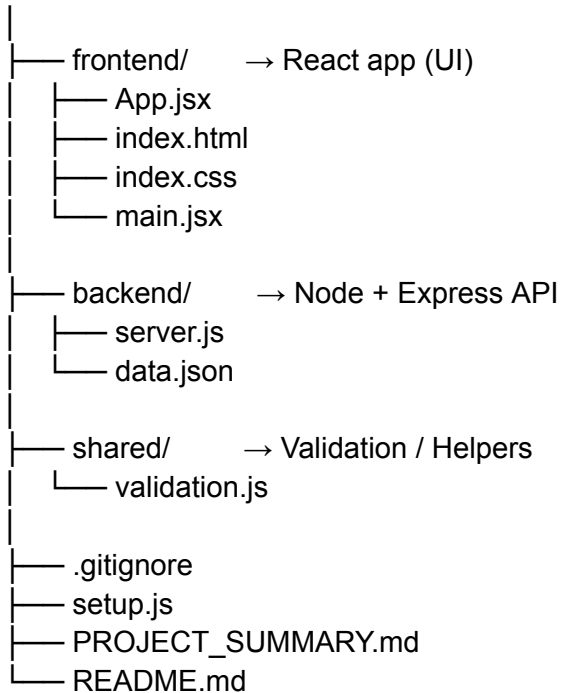




PROJECT STRUCTURE

job-application-tracker/



BACKEND CODE — backend/server.js

```
// backend/server.js
const express = require("express");
const fs = require("fs");
const cors = require("cors");
const app = express();
const PORT = 4000;
```

```
// Middleware
app.use(cors());
app.use(express.json());
```

```
// Read data from file
function readData() {
```

```
    return JSON.parse(fs.readFileSync("backend/data.json", "utf8"));
  }
}
```

```
// Write data to file
function writeData(data) {
  fs.writeFileSync("backend/data.json", JSON.stringify(data, null, 2));
}
```

```
// Get all job applications
app.get("/api/jobs", (req, res) => {
  const data = readData();
  res.json(data.jobs);
});
```

```
// Add new job
app.post("/api/jobs", (req, res) => {
  const data = readData();
  const newJob = {
    id: data.jobs.length + 1,
    title: req.body.title,
    company: req.body.company,
    status: req.body.status || "Applied",
  };
  data.jobs.push(newJob);
  writeData(data);
  res.status(201).json(newJob);
});
```

```
// Update job
app.put("/api/jobs/:id", (req, res) => {
  const data = readData();
  const job = data.jobs.find((j) => j.id === req.params.id);
  if (!job) return res.status(404).json({ error: "Job not found" });

  job.status = req.body.status || job.status;
  writeData(data);
  res.json(job);
});
```

```
// Delete job
app.delete("/api/jobs/:id", (req, res) => {
  const data = readData();
  data.jobs = data.jobs.filter((j) => j.id !== req.params.id);
  writeData(data);
});
```

```
res.json({ message: "Deleted successfully" });
});
```

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



SAMPLE DATA — backend/data.json

```
{
  "jobs": []
}
```



FRONTEND CODE — frontend/App.jsx

```
import React, { useState, useEffect } from "react";
```

```
const API = "http://localhost:4000/api/jobs";
```

```
export default function App() {
  const [jobs, setJobs] = useState([]);
  const [title, setTitle] = useState("");
  const [company, setCompany] = useState("");
```

```
  useEffect(() => {
    fetch(API)
      .then((res) => res.json())
      .then(setJobs);
  }, []);
```

```
  const addJob = async () => {
    const res = await fetch(API, {
      method: "POST",
      headers: { "Content-Type": "application/json" },
      body: JSON.stringify({ title, company }),
    });
    const newJob = await res.json();
    setJobs([...jobs, newJob]);
    setTitle("");
    setCompany("");
  };
}
```

```

const updateStatus = async (id, status) => {
  const res = await fetch(`${API}/${id}`, {
    method: "PUT",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify({ status }),
  });
  const updated = await res.json();
  setJobs(jobs.map((j) => (j.id === id ? updated : j)));
};

const deleteJob = async (id) => {
  await fetch(`${API}/${id}`, { method: "DELETE" });
  setJobs(jobs.filter((j) => j.id !== id));
};

return (
  <div style={{ padding: "20px" }}>
    <h1>Job Application Tracker</h1>

    <input
      placeholder="Job Title"
      value={title}
      onChange={(e) => setTitle(e.target.value)}
    />
    <input
      placeholder="Company"
      value={company}
      onChange={(e) => setCompany(e.target.value)}
    />
    <button onClick={addJob}>Add</button>

    <ul>
      {jobs.map((job) => (
        <li key={job.id}>
          <b>{job.title}</b> @ {job.company} - <i>{job.status}</i>
          <button onClick={() => updateStatus(job.id, "Interviewing")}>
            Update
          </button>
          <button onClick={() => deleteJob(job.id)}>Delete</button>
        </li>
      ))}
    </ul>
  </div>
);

```

```
}
```



SHARED CODE — `shared/validation.js`

```
function validateJob(job) {  
  if (!job.title || !job.company) return false;  
  return true;  
}  
module.exports = { validateJob };
```



SETUP FILE — `setup.js`

```
import { execSync } from "child_process";  
  
console.log("Setting up Job Tracker...");  
execSync("cd backend && npm install", { stdio: "inherit" });  
execSync("cd frontend && npm install", { stdio: "inherit" });  
console.log("✅ Setup complete!");
```



SAMPLE INPUT & OUTPUT

● INPUT 1 – Add a New Job

Frontend Action: User enters

Title: "React Developer"

Company: "TechCorp"

API Call (POST /api/jobs):

```
{  
  "title": "React Developer",  
  "company": "TechCorp"  
}
```

OUTPUT (Response):

```
{
  "id": 1,
  "title": "React Developer",
  "company": "TechCorp",
  "status": "Applied"
}
```

● INPUT 2 – Update Job Status

API Call (PUT /api/jobs/1):

```
{
  "status": "Interviewing"
}
```

OUTPUT:

```
{
  "id": 1,
  "title": "React Developer",
  "company": "TechCorp",
  "status": "Interviewing"
}
```

● INPUT 3 – Get All Jobs

API Call (GET /api/jobs): *Output:*

```
[
  {
    "id": 1,
    "title": "React Developer",
    "company": "TechCorp",
    "status": "Interviewing"
  }
]
```

● INPUT 4 – Delete Job

API Call (DELETE /api/jobs/1)

OUTPUT:

```
{ "message": "Deleted successfully" }
```



OUTPUT ON SCREEN (Frontend)

Job Application Tracker

Job Title: [React Developer]

Company: [TechCorp]

[Add]

List:

1. React Developer @ TechCorp - Interviewing

[Update] [Delete]
