Frontend Program (HTML + JavaScript)

index.html

```
<!DOCTYPE html>

<html>

<head>

  <title>Naan Muthalvan - Job Application Tracker</title>

  <style>

    body { font-family: Arial, sans-serif; margin: 20px; }

    table { width: 100%; border-collapse: collapse; margin-top: 20px; }

    th, td { border: 1px solid #ccc; padding: 8px; text-align: center; }

    input, select, button { margin: 5px; }

    h1 { color: #333; }

  </style>

</head>

<body>

  <h1>Naan Muthalvan - Job Application Tracker</h1>
```

```
<h3>Add Job Application</h3>

<form id="addForm">

    Company: <input type="text" id="companyName" required>

    Job Role: <input type="text" id="jobRole" required>

    Date: <input type="date" id="applicationDate" required>

    Status:

    <select id="status">

        <option>Applied</option>

        <option>Interview Scheduled</option>

        <option>Offered</option>

        <option>Rejected</option>

    </select>

    Notes: <input type="text" id="notes">

    <button type="submit">Add Application</button>

</form>

<h3>All Applications</h3>

<table id="applicationsTable">

    <thead>

        <tr>

            <th>ID</th>

            <th>Company</th>

            <th>Job Role</th>

            <th>Date</th>

            <th>Status</th>
```

```
                <th>Notes</th>
                <th>Actions</th>
            </tr>
        </thead>
        <tbody></tbody>
    </table>
<script>
let applications = [];
let idCounter = 1;
function renderApplications() {
    const tbody = document.querySelector("#applicationsTable tbody");
    tbody.innerHTML = "";
    applications.forEach(app => {
        const row = document.createElement("tr");
        row.innerHTML = `
            <td>${app.id}</td>
            <td>${app.company}</td>
            <td>${app.role}</td>
            <td>${app.date}</td>
            <td>${app.status}</td>
            <td>${app.notes}</td>
            <td>
                <button onclick="updateStatus(${app.id})">Update Status</button>
                <button onclick="deleteApplication(${app.id})">Delete</button>
            </td>
        `;
        tbody.appendChild(row);
    });
}
document.getElementById("addForm").addEventListener("submit", function(e){
    e.preventDefault();
    const company = document.getElementById("companyName").value;
    const role = document.getElementById("jobRole").value;
    const date = document.getElementById("applicationDate").value;
    const status = document.getElementById("status").value;
    const notes = document.getElementById("notes").value;
    applications.push({ id: idCounter++, company, role, date, status, notes });
    renderApplications();
    // Reset form document.getElementById("addForm").reset();
});
function updateStatus(id) {
    const newStatus = prompt("Enter new status (Applied, Interview Scheduled, Offered, Rejected):");
    if(newStatus) {
        const app = applications.find(a => a.id === id);
        if(app) {
            app.status = newStatus;
            renderApplications();
        }
    }
}
function deleteApplication(id) {
    if(confirm("Are you sure you want to delete this application?")) {
        applications = applications.filter(a => a.id !== id);
        renderApplications();
    }
}
// Initial render
renderApplications();
</script>
</body>
</html>
```

💻 Backend program

```
// server.js
// Single-file fullstack app: backend + file DB + frontend
// Usage:
//   npm init -y
//   npm install express
//   node server.js
//
// Then open http://localhost:5000
const express = require('express');
const fs = require('fs');

const path = require('path');

const { randomUUID } = require('crypto');

const app = express();

const PORT = process.env.PORT || 5000;

const DB_FILE = path.join(__dirname, 'jobs.json');

app.use(express.json());

// --- Simple file DB helpers ---

function readDB() {

  try {

    if (!fs.existsSync(DB_FILE)) {

      fs.writeFileSync(DB_FILE, JSON.stringify({ jobs: [] }, null, 2));

    }

    const raw = fs.readFileSync(DB_FILE, 'utf8');

    return JSON.parse(raw);

  } catch (err) {

    console.error('Read DB error:', err);

    return { jobs: [] };

  }

}

function writeDB(db) {

  fs.writeFileSync(DB_FILE, JSON.stringify(db, null, 2));

}

// Ensure DB file exists
```

```
readDB();

// --- API endpoints ---

// GET /api/jobs  -> list jobs

app.get('/api/jobs', (req, res) => {

  const db = readDB();

  const jobs = db.jobs.slice().sort((a,b) => new Date(b.createdAt) - new Date(a.createdAt));

  res.json(jobs);

});

// POST /api/jobs -> create job

app.post('/api/jobs', (req, res) => {

  const { title, company, status = 'Applied', notes = '' } = req.body || {};

  if (!title || !company) return res.status(400).json({ error: 'title and company are required' });

  const db = readDB();

  const job = {

    _id: randomUUID(),

    title: String(title),

    company: String(company),

    status: String(status),

    notes: String(notes),

    dateApplied: new Date().toISOString(),

    createdAt: new Date().toISOString(),

    updatedAt: new Date().toISOString()

  };

  db.jobs.push(job);
  writeDB(db);

  console.log('Added job:', job.title, 'at', job.company);

  res.status(201).json({ message: 'Job added', job });

});

// GET /api/jobs/:id -> get one job

app.get('/api/jobs/:id', (req, res) => {

  const db = readDB();

  const job = db.jobs.find(j => j._id === req.params.id);

  if (!job) return res.status(404).json({ error: 'Job not found' });

  res.json(job);

});

// PUT /api/jobs/:id -> update job

app.put('/api/jobs/:id', (req, res) => {

  const db = readDB();

  const idx = db.jobs.findIndex(j => j._id === req.params.id);

  if (idx === -1) return res.status(404).json({ error: 'Job not found' });

  const job = db.jobs[idx];

  const { title, company, status, notes } = req.body || {};

  if (title !== undefined) job.title = String(title);

  if (company !== undefined) job.company = String(company);

  if (status !== undefined) job.status = String(status);

  if (notes !== undefined) job.notes = String(notes);

  job.updatedAt = new Date().toISOString();

  db.jobs[idx] = job;
  writeDB(db);

  console.log('Updated job:', job._id);

  res.json({ message: 'Job updated', job });

});

// DELETE /api/jobs/:id -> delete job

app.delete('/api/jobs/:id', (req, res) => {

  const db = readDB();

  const idx = db.jobs.findIndex(j => j._id === req.params.id);

  if (idx === -1) return res.status(404).json({ error: 'Job not found' });

  const removed = db.jobs.splice(idx, 1)[0];

  writeDB(db);

  console.log('Deleted job:', removed._id, removed.title);

  res.json({ message: 'Job deleted' });

});

// Health check

app.get('/health', (req, res) => res.json({ status: 'ok', time: new Date().toISOString() }));

// --- Frontend (HTML/CSS/JS embedded) ---

app.get('/', (req, res) => res.type('html').send(htmlContent()));

app.get('/app.js', (req, res) => res.type('application/javascript').send(jsContent()));

app.get('/styles.css', (req, res) => res.type('text/css').send(cssContent()));

// Start server

app.listen(PORT, () => {

  console.log(` Server running at http://localhost:${PORT}`);

  console.log('Open in browser to use Job Tracker.');
```

```
});

// HTML + CSS + JS

function htmlContent() {

  return `<!doctype html>

<html>

<head>

  <meta charset="utf-8" />

  <meta name="viewport" content="width=device-width,initial-scale=1" />

  <title>Naan Muthalvan — Job Tracker</title>

  <link rel="stylesheet" href="/styles.css" />

</head>

<body>

  <div class="container">

    <h1>🧾 Naan Muthalvan — Job Application Tracker</h1>

    <div id="app">

      <form id="jobForm">

        <input id="jobId" type="hidden" />

        <div class="row">

          <input id="title" placeholder="Job Title" required />

          <input id="company" placeholder="Company Name" required />

        </div>

        <div class="row">

          <select id="status">

            <option>Applied</option>
          <option>Interview Scheduled</option>

          <option>Offered</option>

          <option>Rejected</option>

         </select>

         <button type="submit" id="addBtn">Add</button>

         <button type="button" id="cancelEdit" style="display:none">Cancel</button>

        </div>

        <textarea id="notes" placeholder="Notes (optional)"></textarea>

       </form>

       <h3>All Job Applications</h3>

       <div id="list"></div>

     </div>

   </div>

   <script src="/app.js"></script>

</body>

</html>`;

}

function cssContent() {

  return `

body { font-family: Arial; background:#f8fafc; padding:20px; }

.container { max-width:800px; margin:auto; background:#fff; padding:20px; border-radius:10px; box-shadow:0 2px 6px rgba(0,0,0,0.1); }

.row { display:flex; gap:8px; margin-bottom:8px; }

input, select, textarea { padding:8px; flex:1; border:1px solid #ccc; border-radius:6px; }
```

```
button { padding:8px 12px; border:none; background:#2563eb; color:white; border-radius:6px;
cursor:pointer; }

.job-item { display:flex; justify-content:space-between; border-bottom:1px solid #eee; padding:8px 0; }

.job-meta { color:#666; font-size:13px; }

.btn-danger { background:#dc2626; }

`;

}

function jsContent() {

  return `

(async function(){

  const listEl = document.getElementById('list');

  const form = document.getElementById('jobForm');

  const titleEl = document.getElementById('title');

  const companyEl = document.getElementById('company');

  const statusEl = document.getElementById('status');

  const notesEl = document.getElementById('notes');

  const jobIdEl = document.getElementById('jobId');

  const addBtn = document.getElementById('addBtn');

  const cancelEditBtn = document.getElementById('cancelEdit');

  async function fetchJobs() {

    const res = await fetch('/api/jobs');

    const jobs = await res.json();

    renderJobs(jobs);

  }
  function renderJobs(jobs) {

    listEl.innerHTML = jobs.map(job => \`

      <div class="job-item" data-id="\${job._id}">

        <div>

          <b>\${job.title}</b> — \${job.company}<br/>

          <span class="job-meta">\${job.status} • \${new
Date(job.dateApplied).toLocaleString()}</span><br/>

          <small>\${job.notes || ''}</small>

        </div>

        <div>

          <button onclick="editJob('\${job._id}')">Edit</button>

          <button class="btn-danger" onclick="deleteJob('\${job._id}')">Delete</button>

        </div>

      </div>\`).join('');

  } form.addEventListener('submit', async e => {

    e.preventDefault();

    const id = jobIdEl.value;

    const payload = { title:titleEl.value, company:companyEl.value, status:statusEl.value,
notes:notesEl.value };

    const method = id ? 'PUT' : 'POST';

    const url = id ? '/api/jobs/' + id : '/api/jobs';

    const res = await fetch(url, { method, headers:{'Content-Type':'application/json'},
body:JSON.stringify(payload) });

    const json = await res.json();

    alert(json.message);
    resetForm();

    fetchJobs();

  });

  cancelEditBtn.onclick = resetForm;


  window.editJob = async id => {

    const res = await fetch('/api/jobs/' + id);

    const job = await res.json();

    jobIdEl.value = job._id;

    titleEl.value = job.title;

    companyEl.value = job.company;

    statusEl.value = job.status;

    notesEl.value = job.notes;

    addBtn.textContent = 'Update';

    cancelEditBtn.style.display = 'inline';

  };

  window.deleteJob = async id => {

    if (!confirm('Delete this job?')) return;

    await fetch('/api/jobs/' + id, { method:'DELETE' });

    alert('Job deleted');

    fetchJobs();

  };

  function resetForm(){

    jobIdEl.value=''; titleEl.value=''; companyEl.value=''; notesEl.value='';
```

```
    addBtn.textContent='Add'; cancelEditBtn.style.display='none';
  }
  fetchJobs();
})();
`;
}
```

API Documentation (Demo):

Login endpoint POST/ login

```
ⁱⁱ{
  "email": "user@example.com",
  "password": "securepassword123"
}
```

Response(Demo

```
{
   "success ":true,
   "message ":signed in
    successfully "
}
```

4 .Challenges and Solutions :