👩‍💻 Fullstack Developer TA Intern – MERN Stack Assignment (Cuvette)

⏱️ Estimated Time: 3–4 Hours

Objective:
We're evaluating your MERN Stack skills and your ability to explain. You'll build a fullstack app, deploy it, solve a DSA problem in JavaScript, and record a video walkthrough.

📌 PART 1: Build a "Student Job Tracker" Web App

🔧 Tech Stack:

- **Frontend**: React (with Hooks)
- **Backend**: Node.js, Express.js
- **Database**: MongoDB (MongoDB Atlas preferred)
- **Deployment**: Vercel (Frontend), Render or Railway (Backend)

🎯 Features to Build:

1. **Add Job Application**
    - Fields: Company, Role, Status (Applied / Interview / Offer / Rejected), Date of Application, Link
2. **List All Applications**
    - Display job applications in a clean, responsive layout
    - Include filtering by status or date
3. **Update Status**
    - Allow user to update the application status

4. **Delete Application**
    - Allow user to delete a job entry

🚀 Deployment (Mandatory)

- Frontend must be deployed on **Vercel**
- Backend must be deployed on **Render** or **Railway**
- Database hosted on **MongoDB Atlas**
- Final app must be **fully functional and hosted online**

🤖 PART 2: AI Tools & LLMs (Bonus)

Use tools like **ChatGPT**, **GitHub Copilot**, or similar to support your development process.

Submit:

- Prompts or ways you used AI
- How it helped (or didn't)
- Manual changes you made after using the tool

🎥 PART 3: Video Walkthrough (Mandatory)

Record a **10–15 minute** screen recording with **camera ON**.

Cover:

- Feature walkthrough
- Folder/code structure explanation
- Approach and thought process

- Any AI tools usage (if applicable)

🧠 PART 4: Data Structures & Algorithms (in JavaScript)

Choose **any ONE** of the following and solve it using JavaScript.

🔢 Problem 1: Job Tracker Sorting (Medium)

Given an array of job applications:

```
{
  company: "Google",
  role: "SDE Intern",
  appliedDate: "2025-04-01"
}
```

Task: Sort jobs by `appliedDate` (latest first).

🔁 Problem 2: Status Frequency Counter (Easy)

Given an array of job applications with a `status` field, write a function that returns the count of each status.

Example Output:

```
{
  Applied: 4,
  Interview: 2,
  Offer: 1,
  Rejected: 3
```

}

🔍 Problem 3: Detect Duplicate Applications (Medium)

Write a function that checks if there are **duplicate applications** based on a combination of `company + role`, ignoring case sensitivity.

📤 What to Submit:

- ✅ GitHub Repo Link (with proper README)
- ✅ Live Deployed App Link (Vercel + Backend)
- ✅ Video Walkthrough Link (Google Drive/YouTube – Unlisted)
- ✅ JS Code File for DSA Problem
- ✅ Optional: Notes on AI Tools Used

💡 Bonus Points For:

- Clean and modular code
- Good UI/UX
- Effective use of AI tools
- Clear explanation with camera ON

Best of LUCK !