Assignment: Build a Candidate Referral Management System

Objective:
Simulate a part of Worko's functionality by building a **Candidate Referral Management**

System. The goal is to test your ability to build and integrate APIs, manage state in a React frontend, and implement basic backend functionality.

Requirements:
Frontend (React):

1. Build a user interface with the following features:

- **Dashboard:**
- Display a list of referred candidates (fetched from the backend API).
- Each candidate card should display:
- Name
- Job Title
- Status (e.g., Pending, Reviewed, Hired)
- A search bar to filter candidates by job title or status.
- **Referral Form:**
- Allow users to refer a candidate.
- Form fields:
- Candidate Name
- Email
- Phone Number
- Job Title
- Resume (optional, accept only .pdf format).
- On submission, the form should make a POST request to the backend API.
- **Update Candidate Status:**
- Add a dropdown or button to update a candidate’s status from Pending to Reviewed or Hired.

Use **React Hooks** (e.g., useState, useEffect) or a state management library like **Redux** for managing data and API calls.

Backend (Node.js + Express):

1. **API Endpoints:**
    a. POST /candidates: Save a new candidate's data.
    b. GET /candidates: Fetch all referred candidates.
    c. PUT /candidates/:id/status: Update the status of a candidate (e.g., Pending → Reviewed).
    d. Optional: DELETE /candidates/:id: Delete a candidate.
2. **Database:**
3. any DB (**NoSql** is prefered) to store candidate details:
    a. Name, Email, Phone, Job Title, Status, and Resume File URL (if uploaded).
4. **Validation:**
    a. Ensure email and phone number formats are valid.
    b. Restrict resume uploads to .pdf files.
5. **Error Handling:**

    a. Return appropriate error responses for invalid data or server issues.

Bonus Points (Optional):

- **Authentication:** Add basic user authentication using JWT or session-based login.
- **Deployment:** Deploy the application (frontend + backend) and share the live URL.
- **Resume Upload:** Store resumes on cloud storage (e.g., AWS S3, Google Cloud Storage).
- **Metrics Dashboard:** Display basic stats like:
    - Total candidates referred.
    - Candidates by status (Pending/Reviewed/Hired).

Deliverables:

1. Source code hosted on a repository (GitHub/GitLab).
2. Postman Collection or API documentation for backend endpoints.
3. README file detailing:
    a. Features implemented.
    b. Steps to run the project locally.
    c. Any assumptions or limitations.
4. (Optional) Live demo link, if deployed.