Project Statement: Recruitment Tracker

1. Introduction

In the modern workplace, effective recruitment processes are crucial for attracting and retaining top talent. Traditional methods of tracking job applications and candidate progress are often cumbersome and inefficient. This project aims to develop a robust and efficient Recruitment Tracker application using Python and MongoDB. The system will streamline the recruitment process by providing a centralized platform for managing job postings, tracking candidate progress, and storing essential candidate information. The Recruitment Tracker will enhance the efficiency of HR departments, improve candidate experience, and ensure a more organized approach to hiring.

2. Objectives

- **Centralized Management**: Provide a unified platform to manage job postings and track candidate applications.
- **Efficiency and Organization**: Streamline the recruitment process to save time and reduce administrative burdens.
- **Enhanced Candidate Experience**: Ensure a seamless and professional interaction with candidates throughout the hiring process.
- **Data Security**: Securely store and manage sensitive candidate information.

3. Features

1. **User Authentication and Roles**:
    - Secure login and authentication system for HR personnel.
    - Different roles (e.g., HR Manager, Recruiter) with appropriate access levels.
2. **Job Postings Management**:
    - Create, edit, and delete job postings.
    - Categorize job postings by department, location, and job type.
    - Set application deadlines and visibility status.
3. **Candidate Tracking**:
    - Add and update candidate information.

- Track candidate progress through various stages of the recruitment process (e.g., Applied, Screening, Interview, Offer).
- Schedule and manage interview slots.
- Notify candidates of their application status via email.

4. **Resume and Document Management**:
   - Upload and store candidate resumes and related documents.
   - Download and view documents as needed.
5. **Search and Filter**:
   - Search for candidates by name, email, job title, or application status.
   - Filter job postings and candidates based on various criteria.
6. **Notifications and Reminders**:
   - Send automated email notifications to candidates at different stages of the application process.
   - Reminders for HR personnel about upcoming interviews and deadlines.
7. **Reporting and Analytics**:
   - Generate reports on recruitment metrics such as number of applicants, time-to-hire, and candidate conversion rates.
   - Visualize data through charts and graphs.

4. Technical Requirements

Backend:

- **Framework**: Flask/Django
- **Database**: MongoDB
- **Authentication**: JWT (JSON Web Tokens) for secure authentication
- **APIs**: RESTful APIs for interaction between frontend and backend

Frontend:

- **Framework**: React.js
- **UI Components**: Bootstrap/Material-UI for responsive design
- **State Management**: Redux for managing application state

Deployment:

- **Server**: Should be able to run locally on a defined port e.g. http://localhost:3000
- **Docker**: Should be able to run on a Docker container. Should include necessary dockerfile with the code

5. Database Schema

- **Users**:
  - _id (ObjectId)
  - username (String)
  - password_hash (String)
  - role (String)
  - email (String)
- **JobPostings**:
  - _id (ObjectId)
  - title (String)
  - department (String)
  - location (String)
  - job_type (String)
  - description (String)
  - application_deadline (Date)
  - visibility_status (Boolean)
- **Candidates**:
  - _id (ObjectId)
  - name (String)
  - email (String)
  - phone (String)
  - resume (Binary)
  - documents (Array of Binary)
  - job_posting_id (ObjectId, reference to JobPostings)
  - application_status (String)
  - interview_schedule (Array of Date)

6. Implementation Plan

Part 1: Setup and Initial Development

- Setup project repository and environment.
- Implement user authentication and roles.
- Design database schema and configure MongoDB.

Part 2: Core Functionality

- Develop CRUD operations for job postings.
- Implement candidate tracking and application status management.
- Integrate resume and document upload functionality.

Part 3: Search, Filter, and Notifications

- Implement search and filter functionalities for job postings and candidates.
- Setup automated email notifications and reminders.

Part 4: Reporting and Analytics

- Develop reporting features to track recruitment metrics.
- Integrate data visualization for analytics.

Part 5: Testing and Deployment

- Conduct thorough testing (unit, integration, and user acceptance testing).
- Deploy locally or deploy to a docker container.

7. Security Considerations

- **Data Encryption**: Encrypt sensitive data (e.g., passwords, candidate documents) in transit and at rest.
- **Access Control**: Implement role-based access control to restrict data access based on user roles.
- **Input Validation**: Validate all user inputs to prevent SQL injection and other attacks.
- **Regular Audits**: Perform regular security audits and vulnerability assessments.

8. Timeline

The project is expected to be completed within the hackathon timeframe.

9. Conclusion

The Recruitment Tracker aims to revolutionize the recruitment process by providing a comprehensive and efficient solution for managing job applications and candidate progress. By leveraging Python and MongoDB, the application will offer a robust backend and a user-friendly interface, ensuring a seamless experience for HR personnel and candidates alike. The project's success will be measured by its ability to improve recruitment efficiency, enhance candidate experience, and provide valuable insights through reporting and analytics.

This project, though ambitious, is well within the scope of a 6-hour hackathon due to its modular design and focused feature set. By prioritizing core functionalities and ensuring effective collaboration among team members, the Recruitment Tracker can be developed and deployed within the given timeframe, providing a valuable tool for modern HR departments.

Important Instructions

1. You are welcome to use ChatGPT or other LLMs for accelerating development by seeking help for generating code etc.
2. Use of coding assistants is encouraged.
3. The goal is to assess how resourceful and efficient you are at finding a solution using freely available resources on the internet.
4. Seeking someone else's help is not allowed.
5. You will be expected to provide a detailed walkthrough of the entire code repository.
6. If the entire project is not completed in the allotted time, the project will be evaluated based on items that are complete and fully functional.