HRM Project with

Recruiting Microservice

- **Jobs**
  - Admin/HR should be able to create jobs that include all the Job details information.
  - Home page should display list of jobs sorted by recent with pagination enabled.
  - Home page should have a big search bar to search the Jobs by title or description or keyword search
  - Job Title should be linked to Job Details page showing all the list of submissions under the job details for the admin.
  - Job Details should have Apply button so that candidates should be able to apply to the Job.
  - Admin should be able to close the Job manually or after all the positions have been filled.
    - **async communication** ➜ Should publish the job closed message to the Message Queue (RabbitMQ or Azure ServiceBus), Interviewer microservice should subscribe to the event and add delete any future interviews.
      - Event Name ➜ JobClosed
- **Candidates**
  - Candidates should be able to create account, with profile informations and have their resume upload.
    - **async communication** ➜ when a candidate creates account Users table will have info then, that user info will be published to message queue, where recruiting microservice will subscribe to that event and add entry in candidate table.
      - Event Name ➜ CandidateAccountCreated
  - Candidates once log in, should be able to apply job with existing resume and also add additional comments. Goes into Submissions table.
    - **Azure Functions & Blob Storage** ➜ When candidate uploads resume, Azure function should trigger (http) and send the file to Azure Blob storage.
  - Candidates should view their account details, update resume (re-upload), Re-uploading should delete old resume from blob storage and add new one.
  - Candidates should be able to login/logout from website.

Interviews Microservice

- **Interviews**
  - Admin/HR should be able to create interview for submissions, should select interviewer from dropdownlist of interviewers ➜ should go to Interviews table.

        - **sync communication** ➜ When creating interviews, need to make **http** call to Recruiting microservice to get candidate information and submission information, and save into interviews table.
    - Admin/HR should be able to see all the interviews with pagination sorted by interview date. Should have link to interview details.
    - Should be able to re-schedule the interviews, delete/cancel interviews.
        - **Azure Functions**: When interview has been re-scheduled, cancelled or deleted, a message has to be sent to message queue and Azure Function should trigger and send emails using SendGrid.
    - Manager should give feedback, rating and if selected for the position.
    - When Manager login to system , they should have a dashboard of all the interviews sorted by date with pagination.

OnBoarding Microservice

- **Employees**
    - Admin/HR should be able to add, update and view list of employees (with pagination)
    - Admin/HR should be able to terminate employees
        - **async communication** ➜ Should publish the employee termination information to the Message Queue (RabbitMQ or Azure ServiceBus), Interviewer microservice should subscribe and soft delete employee info in Interviewer table if exists.
            - Event Name ➜ EmployeeTerminated
    - Admin should be able to assign an Employee as Interviewer.
        - **async communication** ➜ Should publish the employee details to the Message Queue (RabbitMQ or Azure ServiceBus), Interviewer microservice should subscribe and add entry into Interviewer table.
            - Event Name ➜ EmployeeCanInterview

Authentication Microservice

- **Authentication**
    - Should be able to create account for candidates and Employees with Roles being candidate and employee.
    - When a candidate is converted to employee, should have new email address and some personal info with new password etc in Users table.
    - Employee can have multiple roles such as Employee, Admin, HR, Manager etc.
    - Every employee can have multiple claims to have fine control over functionality.

#hrmproject#