Assignment – Backend Engineer[Python]

You need to develop a scheduler microservice that allows job scheduling while maintaining critical job-related information. The service should have API endpoints for job management, such as listing all jobs, retrieving details of a specific job by ID, and creating new jobs.

You don't need to worry about the job service logic. Just use any dummy job, such as email notifications, number crunching, etc.

Features:

1. Job Scheduling: The microservice will facilitate the scheduling of customized jobs with flexibility in configuration
    - *Note: POC logic for how jobs will be scheduled and executed is required. For example, if I schedule a job to be executed every Monday, there should be code that can do that*
2. API Endpoints:
    - GET /jobs: This endpoint will list all available jobs, providing a comprehensive overview of the scheduled tasks
    - GET /jobs/:id: By accessing this endpoint with a specific job ID, users can view detailed information about the selected job, including its scheduling details
    - POST /jobs: This endpoint allows users to create new jobs. The API validates input data and adds the job to the scheduling table
3. Database Integration: The database will contain fields such as job name, last run timestamp, next run timestamp, and other pertinent job details
4. Customization: Each job will be customizable, allowing users to define specific attributes, scheduling intervals, and other relevant parameters
5. Scalability: The application should be scalable to handle increased applicant complexity, ~10,000 users spread globally, ~1,000 services, and ~6,000 API requests per minute

What we will primarily look at:

- Your ability to meet all the points enumerated in the "features" section
- Best coding principles
- Modularized production-ready code
- Adhering to the SOLID principles

What will get you brownie points:

- Optimized performant code
- Use of library for API documentation

Submission:

Share your solution as a git repository with a readme file for project setup. One pager explaining how scaling can be achieved in terms of multiple services and managing APIs.