Healthcare Interview Challenge

Submission Deadline: 11/04/2025

Overview Design and implement a healthcare appointment scheduling system. The challenge focuses on developing a robust, secure backend service that efficiently manages patient data and enables seamless appointment scheduling with healthcare providers.

Core Requirements:

1. **Patient Management** Create backend functionality to register and manage patient profiles, store basic patient information and contact details, and track patient identification and insurance information.
2. **Doctor Management** Implement features to maintain doctor profiles with specializations and manage doctor availability schedules.
3. **Appointment Scheduling** Build functionality to create appointments between patients and doctors, check doctor availability when scheduling, prevent scheduling conflicts and double-bookings, and manage appointment status changes.
4. **Bonus (Pick one):**
    a. **Simple User Interface** Develop a minimal frontend with a framework of your choice to demonstrate and test the core backend functionalities.
    b. **Medical Records** Implement basic functionality to store medical records for patients, link records to specific appointments, and implement appropriate access controls for sensitive information.

Technical Evaluation:

1. **Backend Architecture (25%)** - Design an efficient and maintainable backend structure with clear component organization. Include sequence diagrams for key processes and a comprehensive database schema diagram.
2. **API Design & Documentation (20%)** - Develop RESTful API endpoints following best practices with proper documentation using Swagger/OpenAPI. Implement meaningful error responses and status codes.

3. **Data Modeling (15%)** - Design a database schema with appropriate relationships between patients, doctors, and appointments. Implement data validation and ensure referential integrity between entities.
4. **Security Implementation (10%)** - Implement authentication and authorization with role-based access controls (OAuth 2.0). Secure sensitive healthcare information and document your security approach.
5. **Error Handling, Performance & Testing (10%)** - Build comprehensive error handling with proper validation.
6. **Performance optimizations(20%)** including asynchronous processing and message queuing (Kafka/RabbitMQ/Redis). Include thorough test coverage of core functionality.

Deliverables

1. **Source Code** - Complete backend implementation with database setup scripts or migrations and a README with clear setup instructions.
2. **Documentation** - API documentation using Swagger/OpenAPI, database schema diagram, sequence diagram for appointment booking, and a brief document explaining your design decisions.
3. **Testing** - Unit tests for core business logic and API integration tests that verify endpoint functionality.

Evaluation Criteria Your submission will be evaluated based on API design and documentation quality, database schema design, code quality and organization, error handling and edge cases, security implementation, testing approach, and overall completeness of a working solution.

Technology Stack You may use any of the following:

- **Backend**: Python (Django/Flask/FastAPI)
- **Database**: PostgreSQL, MySQL
- **Documentation**: Swagger/OpenAPI

Submission Instructions: Fill out the submission form here with links to your GitHub repository and other required documentation.