**#4. Doctor Appointment Management Application**

Develop a Web based Doctor Appointment Management Application for a clinic online.

1. Doctor Management: Create, read, update, and delete doctor details which includes specialization, availability timings, etc.
2. Patient Information Management: Create, read, update, and delete patient information, including basic details and contact information.
3. Search/Book appointment: Search the Doctors List and pick the respective illness specialist doctor and book an appointment. (Payment feature excluded)

Requirements

* Develop a Web Application for managing doctor appointments using Spring MVC
* Implement CRUD operations for doctors and patients as well as bookings
* Implement a user-friendly web interface using JSP pages for adding doctor details, managing appointment (executive), searching doctor, booking an appointment, registering himself (patient)
* Integrate with a database for persistent data storage
* Implement Spring AOP for cross-cutting concerns like logging
* Implement error handling to catch and handle exceptions gracefully
* Implement bean validations to ensure data integrity
* Utilise Spring Data JPA for data access and persistence management

Technologies

* Spring Web
* Validation
* Spring Data JPA
* Hibernate
* JSP
* AspectJ Weaving
* Apache Maven
* H2/ MySQL

Instructions

1. Create a Spring Boot project using Spring Initializr or your preferred IDE.
2. Define entity classes for doctor, patient and appointment.
3. Create repository interfaces for each entity class using Spring Data JPA.
4. Develop respective controller classes to handle API requests related to required operations.
5. Implement Spring AOP aspects for logging using AspectJ annotations.
6. Implement error handling to catch and handle any exceptions that occur during API requests.
7. Implement bean validations
8. Develop JSP Pages for user interfaces related to adding/updating doctor details, managing appointments (executive), searching for doctor and booking appointment, registering himself, viewing appointment status (patient)
9. Deploy the application on Apache Tomcat server
10. Run the application using a browser and test the functionalities

Grading Rubric

* Functionality (50%):
  + Correct implementation of CRUD operations for all entities.
  + Seamless working of all processes
* Code Structure (20%):
  + Clear and well-organised code
  + Proper use of Spring Boot annotations and patterns
  + Effective use of JSP pages for user interfaces
  + Clear separation of concerns between business logic and cross-cutting concerns using Spring AOP
* Documentation (20%):
  + Clear and comprehensive API documentation
  + Detailed explanation of error handling and bean validation implementation
  + Explanation of Spring AOP aspects for logging
  + Instructions for deploying and testing the application
  + User manual for web interfaces
* Presentation (10%):
  + Professional presentation of the project code and documentation
  + Clear and concise explanation of the application's functionality
  + Effective demonstration of the application's features and integration with Spring technologies