Spring boot Training Assignment (Session1-2)

Topics Covered: -

Session 1: Introduction

· Introduction to web services.

· Introduction to Spring boot

· Dependency Injection

· Setting up a Spring Boot project using spring initializer and traditional method.

Session 2: RESTful Web Services

· Building RESTful APIs with Spring Boot

· Handling HTTP Methods (GET, POST, PUT, PATCH, DELETE)

· Status code overview.

· Working with Spring Boot Starter projects

Assignment: -

Questoin.1-> Define what web services are and their significance in modern software development.

Question.2->Explain the types of web services (e.g., SOAP, REST) and their differences.

Question.3->Discuss the features and benefits of using Spring Boot for building web applications.

Question.4->Explain how to set up a Spring Boot project using Spring Initializer.

Question.5->Define dependency injection and its importance in Spring framework.

Question.6->Discuss the use of annotations such as

@RestController,

@RequestMapping,

@GetMapping,

@PostMapping,

@PutMapping,

@PathVariable,

@RequestParam,

@Bean,

@Service,

@Component,

@Autowired,

@ResponseBody,

@ResponseStatus,

@RequestMethod, for defining API endpoints.

Question.7->Briefly explain the significance of @Bean annotation in spring boot.

Question. 8-> Create a new Spring Boot project using Spring Initializer.
Include necessary dependencies such as Spring Web for building RESTful APIs.

A-> Define a Department class with attributes such as id, name
Implement getters and setters for the Department class.
-->Implement an endpoint to add a new Department to the system.
-->Implement an endpoint to retrieve the details of a specific Department by their ID.
-->Implement an endpoint to update the details of an existing Department.
-->Implement an endpoint to delete an Department from the system based on their ID.
-->Return appropriate response codes and messages for success and failure cases.

B-> Define an Employee class with attributes such as id, name, designation, departmentId and salary.
Implement getters and setters for the Employee class.
-->Implement an endpoint to add a new employee to the system.
-->Implement an endpoint to retrieve the details of a specific employee by their ID.
-->Implement an endpoint to update the details of an existing employee.

-->Implement an endpoint to delete an employee from the system based on their ID.
-->Return appropriate response codes and messages for success and failure cases.

C-> Implement an endpoint to add or remove employee in the department with corner cases like same employee can not be added again to same department.
-->Implement an endpoint to get particular number of employees in single department with their names.
-->Implement an endpoint to get particular number of department related to single employee with their names.