



University of Sri Jayewardenepura

Faculty of Technology

Department of Information Communication Technology

ITS 4243 - Microservices and Cloud Computing

Assignment 01

ICT/21/929

Part 1

1. What is Spring Boot and why is it used?

Spring Boot is a Java framework. It helps to build Spring applications easily. It gives many built-in tools, so we don't need to do much setup. It is used because it makes development faster and easier.

2. Explain the difference between Spring Framework and Spring Boot.

Spring Framework is big and needs a lot of setup. Spring Boot is a simpler way to use Spring. Spring Boot gives automatic setup and an embedded server. So we can run the project easily.

3. What is Inversion of Control (IoC) and Dependency Injection (DI)?

IoC means the control of objects is given to the Spring container. We don't create objects ourselves.

DI means Spring gives the needed objects to a class automatically. It helps to connect parts of the program easily.

4. What is the purpose of application.properties / application.yml?

These files are used to keep the project's settings. For example, we can add database details, server port, and other options. Spring Boot reads these files when the app starts.

5. Explain what a REST API is and list HTTP methods used.

REST API helps two systems to talk using the internet. It uses HTTP methods like:

GET – to get data

POST – to add data

PUT – to update data

DELETE – to remove data

6. What is Spring Data JPA? What is an Entity and a Repository?

Spring Data JPA helps to work with databases easily.

An Entity is a class that maps to a table in the database.

A Repository is an interface that helps to save, find, update, and delete data.

7. What is the difference between @Component, @Service, @Repository, @Controller, @RestController?

@Component – for general classes

@Service – for business logic

@Repository – for database layer

@Controller – for web pages (HTML)

@RestController – for REST APIs (JSON data)

8. What is @Autowired? When should we avoid it?

@Autowired helps to automatically connect one class to another.

We should avoid it when we use too many of them, or when it makes the code hard to test or understand.

9. Explain how Exception Handling works in Spring Boot (@ControllerAdvice).

@ControllerAdvice helps to handle errors in one place. When an error happens in any controller, it goes to the advice class, and it gives a custom message or response.

10. What is the role of Maven/Gradle in a Spring Boot project?

Maven and Gradle are build tools. They help to add libraries, build the project, and manage versions. They make it easy to run and package the Spring Boot app.

Part 2

Screenshots

The screenshot shows the Postman interface with a dark theme. On the left, there's a sidebar with 'My Workspace' containing a collection named 'Student management system' which includes endpoints for Add Student, Get All Students, Get Student By ID, Update Student, and Delete Student. The main area shows a 'Get Student By ID' request. The URL is set to `http://localhost:8080/api/students/1`. The 'Body' tab is selected, showing a JSON response with one student object:

```
1 [  
2   {  
3     "id": 1,  
4     "name": "John Doe",  
5     "email": "john@example.com",  
6     "course": "Computer Science",  
7     "age": 22  
8   }  
9 ]
```

The status bar at the bottom indicates a 200 OK response with 53 ms latency and 254 B size.

This screenshot is similar to the previous one but shows a 'Get All Students' request instead. The URL is `http://localhost:8080/api/students`. The 'Body' tab shows a JSON response with multiple student objects:

```
1 [  
2   {  
3     "id": 3,  
4     "name": "Jane Yellow",  
5     "email": "yellow@example.com",  
6     "course": "Art",  
7     "age": 25  
8   },  
9   {  
10    "id": 4,  
11    "name": "Jane Brown",  
12    "email": "Brown@example.com",  
13    "course": "Music",  
14    "age": 21  
15  }  
16 ]
```

The status bar at the bottom indicates a 200 OK response with 16 ms latency and 422 B size.

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar is visible with a collection named 'Student management system'. Under this collection, the 'POST Add Student' endpoint is selected. The main workspace displays a POST request to 'http://localhost:8080/api/students'. The 'Body' tab is selected, showing a raw JSON payload:

```
1 {  
2   "name": "Jane Brown",  
3   "email": "Brown@example.com",  
4   "course": "Music",  
5   "age": 21  
6 }
```

The response status is '201 Created' with a response time of 27 ms and a size of 251 B. The response body is also shown in JSON format:

```
1 {  
2   "id": 4,  
3   "name": "Jane Brown",  
4   "email": "Brown@example.com",  
5   "course": "Music",  
6   "age": 21  
7 }
```

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar is visible with a collection named 'Student management system'. Under this collection, the 'PUT Update Student' endpoint is selected. The main workspace displays a PUT request to 'http://localhost:8080/api/students/1'. The 'Body' tab is selected, showing a raw JSON payload:

```
1 {  
2   "id": 1,  
3   "name": "John Doe",  
4   "email": "john@example.com",  
5   "course": "Computer Science",  
6   "age": 30  
7 }
```

The response status is '200 OK' with a response time of 98 ms and a size of 254 B. The response body is also shown in JSON format:

```
1 {  
2   "id": 1,  
3   "name": "John Doe",  
4   "email": "john@example.com",  
5   "course": "Computer Science",  
6   "age": 30  
7 }
```

The screenshot shows the Postman application interface. In the left sidebar, under 'My Workspace', there is a collection named 'Student management system' which contains several API endpoints: POST Add Student, GET Get All Students, GET Get Student By ID, GET Update Student, and GET Delete Student. The 'GET Delete Student' endpoint is currently selected. The main workspace displays a DELETE request to the URL `http://localhost:8080/api/students/2`. The 'Params' tab is active, showing a single parameter 'Key' with the value 'Value'. Below the request, the response section shows a status of '204 No Content' with a response time of '48 ms' and a size of '112 B'. The response body is empty, containing only the number '1'. At the bottom of the screen, there are various navigation and settings icons.