

Test Summary

- No. of Sections: 2
- No. of Questions: 10
- Total Duration: 90 min

Section 1 - MCQ

Section Summary

- No. of Questions: 9
- Duration: 15 min

Additional Instructions:

None

Q1. Which of the following is NOT a valid data type in JSON?

Date

String

Number

Boolean

Array

Q2. What is the primary responsibility of the Controller in the MVC architecture?

Managing the application's data and business logic

Rendering the user interface and displaying data to the user

Handling user input and updating the Model

Communicating with the database for CRUD operations

Q3. What is the purpose of the @Value annotation in Spring Framework?

To inject values from application.properties files

To define a bean in the Spring context

To specify a default value for a method parameter

To define a constant value in a class

Q4. Which of the following is the correct configuration for connecting to a MySQL database using JPA in a Spring Boot application?

```
spring.datasource.url=jdbc:mysql://localhost:3306/mydb
spring.datasource.username=root
spring.datasource.password=password
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update
```

```
spring.datasource.url=jdbc:mysql://localhost:3306/mydb
spring.datasource.username=root
spring.datasource.password=password
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=create-drop
```

```
spring.datasource.url=jdbc:mysql://localhost:3306/mydb
spring.datasource.username=root
spring.datasource.password=password
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.jpa.database-platform=org.hibernate.dialect.MySQLDialect
spring.jpa.show-sql=false
spring.jpa.hibernate.ddl-auto=create
```

```
spring.datasource.url=jdbc:mysql://localhost:3306/mydb
spring.datasource.username=root
spring.datasource.password=password
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect
spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=none
```

Q5. Which of the following statements about the @PathVariable annotation in Spring is NOT true?

The @PathVariable annotation is used to bind a part of the URI path to a method parameter.

The @PathVariable annotation is used to extract query parameters from the URL.

The @PathVariable annotation can be used with any HTTP method.

The @PathVariable annotation can be used to handle dynamic parts of the URL.

Q6. How can you specify a default value for a @PathVariable in Spring?

Using the defaultValue attribute of the @PathVariable annotation

By using the @DefaultValue annotation alongside @PathVariable

By using the @Default attribute alongside @PathVariable

It is not possible to specify a default value for a @PathVariable in Spring

Q7. Which of the following JPA query is equivalent to the SQL query "SELECT * FROM Employee WHERE age > 25 AND department = 'HR'?"

SELECT e FROM Employee e WHERE e.age > 25 AND e.department = 'HR'

SELECT e FROM Employee e WHERE e.age > 25 && e.department = 'HR'

SELECT e FROM Employee e WHERE e.age > 25 OR department = 'HR'

SELECT e FROM Employee e WHERE e.age > 25 && department = 'HR'

Q8. What is the HTTP status code commonly returned by a DELETE API to indicate successful deletion of a resource?

200 OK

201 Created

204 No Content

400 Bad Request

Q9. What is the HTTP status code commonly returned by an UPDATE API to indicate successful update of a resource?

200 OK

201 Created

204 No Content

400 Bad Request

Section 2 - Project

Section Summary

- No. of Questions: 1
- Duration: 75 min

Additional Instructions:
None

Q1. Cop Friendly App - ESeva

Overview:
Develop a web-based application called "Cop Friendly App - ESeva" using CRUD operation.

Functional Requirements:

Create 4 folders inside the WORKSPACE/springapp/src/main/java/com/example/springapp

- Controller
- Model
- Repository
- Service

Inside the controller, create a Java file named “ApiController.java”

Inside the model, create a Java file named “Eseva.java”

Create 4 variables

- id – int
- name - string

3. badgeNumber - string
4. department - string

as well as create getters and setters and constructors for the corresponding variables.

- Inside the Repository, create a Java file named "EsevaRepo.java"
- Inside the service, create a Java file named "EsevaService.java"

The project structure looks like this image

Core Platform

OpenJDK 11

API:

Question 1 : (16 Marks)

POST - "/" --> true/false **(8 Marks)**

GET -("/{id}" --> cop object **(4 Marks)**

GET - "/" --> List of cop object**(4 Marks)**

Question 1 : (16 Marks)

PUT -("/{id}" --> cop object **(8 Marks)**

DELETE -("/{id}" --> true/false **(8 Marks)**

Note:

Copy and paste it into the **application.properties** file

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://localhost/cop_friendly_app?createDatabaseIfNotExist=true

spring.datasource.username=root

spring.datasource.password=examly

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.show-sql= true

spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect

API endpoint:

8080

Platform Guidelines:

To run the command use Terminal in the platform.

Spring Boot:

Navigate to the springapp directory => **cd springapp**

To start/run the application '**mvn spring-boot:run**'

Click on the Run Test Case button to pass all the test cases

Answer Key & Solution

Section 1 - MCQ

Q1	Date	
	Solution	
	No Solution	
Q2	Handling user input and updating the Model	
	Solution	
	No Solution	
Q3	To inject values from application.properties files	
	Solution	
	No Solution	
Q4	spring.datasource.url=jdbc:mysql://localhost:3306/mydb spring.datasource.username=root spring.datasource.password=password spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver spring.jpa.database-platform=org.hibernate.dialect.MySQL5Dialect spring.jpa.show-sql=true spring.jpa.hibernate.ddl-auto=update	
	Solution	
	No Solution	
Q5	The @PathVariable annotation is used to extract query parameters from the URL.	
	Solution	
	No Solution	
Q6	Using the defaultValue attribute of the @PathVariable annotation	
	Solution	
	No Solution	
Q7	SELECT e FROM Employee e WHERE e.age > 25 AND e.department = 'HR'	
	Solution	

	No Solution
Q8	204 No Content
	Solution
	No Solution
Q9	200 OK
	Solution
	No Solution
	Section 2 - Project
Q1	Solution cannot be displayed