GlobalLogic®

A Hitachi Group Company

GloEventify Portal

Spring HandsOn Assessment

Assessment Flavour - **Java Developer**

Complexity - Medium

Target Band - Band x / Band 1 / Band 2

Title of the Project -

Time: 3 hours

Topics Covered:

Topics
REST API
Spring Data JPA
Exception Handling

SOFTWARE REQUIREMENTS

- **Java Latest Version**
- **Any IDE that supports Java Application Development**
    - **Intelij** or **STS**

TABLE OF CONTENT

{Intentionally left blank}

PURPOSE

You must develop APIs for the Event Registration Portal website. Users will use the website to register for various available events.

- Users can register for a profile on the website by filling in the information in a web form.
- The users can also see which events they have registered for.
- The user can also cancel their registration for a particular event.

PROBLEM STATEMENT

This project entails the creation of Java-based REST APIs for an Event Registration System. These APIs will enable users to register profiles, sign up for events, view their registered events, and cancel registrations. The objective is to streamline event management processes and facilitate efficient participation

PROJECT IMPLEMENTATION

Database Design**:** Application schema and tables

Note: You can ignore the create table command if your application is configured to auto-create tables using Hibernate/JPA properties.

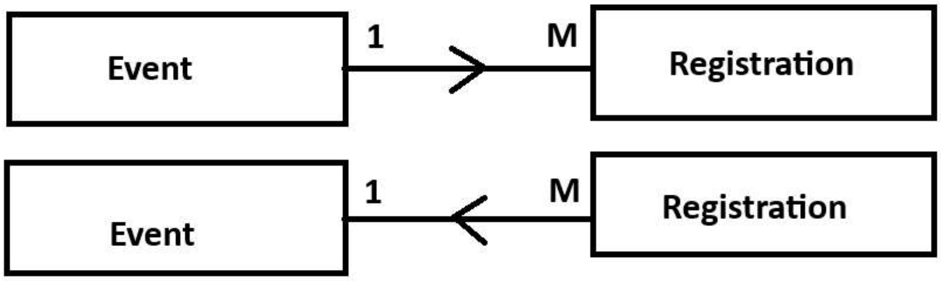
Table Structure with data types

Event		Description
eventId	Long	It must be of Integer data type only. It is the primary ID that is unique for each product. Example: 1
eventName	String	It must be of String data type. It is the name chosen by the event organizer. Example: "Holi Party"

eventDate	LocalDate	It must be of String data type. It is the date on which the event will occur. Example: "yyyy-MM-dd"
location	String	It must be of String data type. It is the area where the event will occur. Example: “Alof Hotel, Whitefield”

Registration		Description
registrationId	Long	It Must be of Integer data type only. It is the primary ID that is unique for each user. Example: 1
name	String	It must be of String data type. It is the name chosen by the user for the profile. Example: "JohnDoe"
email	String	It must be of String data type. It is the user's email address used for communication and log in. Example: "johndoe@example.com"
contactNo	String	It must be of String data type. It is the contact no of the user who is doing the registration Example: "9831231234"
status	RegistrationStatus	It is of RegistrationStatus type which is an Enum and has the status CONFIRMED and CANCELED Example: "CONFIRMED"

Entity Relationship



Note:
There is bidirectional mapping which is one-to-many and many-to-one for the Entities as shown in the diagram.

PROJECT BACKEND IMPLEMENTATION DETAILS

Application flow with functionality:

Note**:** Define appropriate custom exceptions for invalid values and handle the exceptions with appropriate error messages.

API Details:

Event Entity			
API Endpoints	**HttpMethod**	**Description**	**Status Code**
/api/events	POST	Create a new event.	201 Created
		Throws an exception with an informative message if the data is invalid.	400 Bad Request
/api/events/{eventId}	GET	Retrieve details of a specific event.	200 OK
		Throws an exception with an informative message if the event ID is invalid.	404 Not Found
/api/events/{eventId}	DELETE	Deletes the event using a specific event ID	200 OK
		Throws an exception with an informative message if the event ID is invalid.	404 Not Found

Registration Entity			
API Endpoints	**HttpMethod**	**Description**	**Status Code**
/api/events/register	POST	Create a new registration	201 Created
		Throws an exception with an informative message if the data is invalid.	400 Bad Request
		Throws an exception if email id is already used in another registration	

/api/events/{eventId}/registerstatus/{registrationId}	GET	Retrieve the status of the registration for the particular event	200 OK
		Throws an exception with an informative message if the event ID or registration ID is invalid.	404 Not Found
		Throws an exception with an informative message if the event ID does not belong to the specific user or vice versa	404 Not Found

RUNNING THE BACKEND APPLICATION

Test the application by running the main file in the starter code.

Note: Main file code is commented. Uncomment the code, test the application, and comment it back before submitting the assessment.

EVALUATION METRICS

Si. No.	Assessment Parameter	Marks
1.	Event Entity User Stories (3 Rest Endpoints)	24 (8 *3)
	Creating the JPA Entity with appropriate Entity Mapping and Creating the Repository layer (2 * 3)	
	Creating the Business Logic (3 *3)	
	Handling the invalid data in the Dto class using the Hibernate Validator and Creating the well-defined Controller Layer (3 * 3)	
2.	Event Entity User Stories (3 Rest Endpoints)	16 (8 *2)
	Creating the JPA Entity with appropriate Entity Mapping and Creating the Repository layer (2 * 2)	
	Creating the Business Logic (3 * 2)	
	Handling the invalid data in the Dto class using the Hibernate Validator and	

	Creating the well-defined Controller Layer (3 * 2)	
3.	Proper Exception Handling (Using the User-Defined Exceptions and Exception Handler)	6
4.	Best Coding Practices	4
Total: 50 marks		