
System Requirements Specification Index

For

Event Management

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Contents

1	Business-Requirement:	3
1.1	Problem Statement:	3
1.1.1	Event Management	3
2.	Template Code Structure	
1.1	Event Controller	
3.	Resources AVAILABLE:	4
4	Suggested WIREFRAMES:	5
5	Business Validations	9
6	Considerations	9
7	Execution Steps to Follow	9

1 BUSINESS-REQUIREMENT:

1.1 PROBLEM STATEMENT:

The purpose of this application is to provide a platform to manage events. Where users can add, edit and delete their to-do events.

1.1.1 Event Management:

The Event Management allows you to:

1. Access home page
2. Should be able to add a new event.
3. It can have basic fields like name, description, place and eventDate.
4. Should be able to get the list of events.
5. Should be able to edit and delete any event.
6. Should be able to search for an event.

2. TEMPLATE CODE STRUCTURE:

2.1 EVENT CONTROLLER

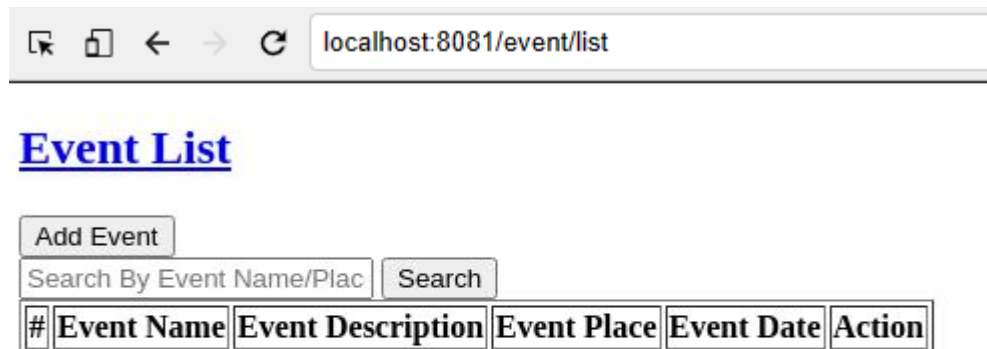
Method Exposed	Purpose
listEvents()	Should return page "list-events" with required data.
showFormForAdd()	Should return page "add-event-form" for adding an event.
saveEvent()	Should save an event and return "event/list" with required data.
showFormForUpdate()	Should show event details in page "update-event-form" to edit an event.
deleteEvent()	Should delete an event and return "event/list" with required data.
searchEvents()	Should search an event and return "list-events" with required data.

3. RESOURCES AVAILABLE:

Description	View Pages Name	Remarks
Common UI		
Home Page	list-events	Contains a homepage which shows a list of all events along with options to add, edit , delete and search a event.
All events	list-events	
Add an event	add-event-form	
Update an event	update-event-form	
Search an event	list-events	

3 SUGGESTED WIREFRAMES:

1. Homepage – Visitor Landing Page



A browser window showing the URL `localhost:8081/event/list`. The page has a title **Event List**. Below the title is an **Add Event** button. Underneath is a search bar with the placeholder text "Search By Event Name/Plac" and a **Search** button. At the bottom is a table with the following headers: #, Event Name, Event Description, Event Place, Event Date, and Action.

#	Event Name	Event Description	Event Place	Event Date	Action
---	------------	-------------------	-------------	------------	--------

2. Create an Event



A browser window showing the URL `localhost:8081/event/addEventForm`. The page has a title **Events**. Below the title is a section header **Add Event**. The form contains the following fields: **Event Name** (text input), **Event Description** (text area), **Event Place** (text input), and **Event Date** (text input with a placeholder `mm/dd/yyyy` and a calendar icon). Below the form is a **Save** button and a [Back to List](#) link.

localhost:8081/event/addEventForm

Events

Add Event

Event Name

Event Description

Event Place

Event Date

[Back to List](#)

BUSINESS VALIDATIONS

1. Id must be of type id.
2. Name value not blank, min 2 and max 40 characters.
3. Description value not blank, min 2 and max 200 characters.
4. Place value not blank.
5. Event date value not blank.

4 CONSIDERATIONS

The Code template already contains skeleton methods for service and controller layer. Please write your logic in it.

5 EXECUTION STEPS TO FOLLOW

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu
(Three horizontal lines at left top) → Terminal → New Terminal
3. To build your project use command:
mvn clean package -Dmaven.test.skip
4. To launch your application:

java -jar <your application war file name>

5. This editor Auto Saves the code
6. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use **CTRL+Shift+B**-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
7. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
8. To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.
9. This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.

Note: The application will not run in the local browser

10. Default credentials for MySQL:
 - a. Username: **root**
 - b. Password: **pass@word1**
11. To login to mysql instance: Open new terminal and use following command:
 - a. **sudo systemctl enable mysql**
 - b. **sudo systemctl start mysql**
 - c. **mysql -u root -p**

The last command will ask for password which is '**pass@word1**'

12. Mandatory: Before final submission run the following command:
mvn test
13. You need to use **CTRL+Shift+B** - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.