
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	4
2.1	Event Controller	4
2.2	Resources Available:	4
3	Suggested Wireframes:	5
4	Business Validations	9
5	Considerations	9
6	Execution Steps to Follow	9

1 BUSINESS-REQUIREMENT:

1.1 PROBLEM STATEMENT:

The Event Management Application serves as a comprehensive platform for organizing and tracking the upcoming events.

Your task is to develop a system that enables users to access a home page, add new events with key details such as name, description, place, and date, and manage these entries effectively. Additionally, users should be able to edit, delete, and search for events by name or location. The system must also include a feature to view and sort past events, enhancing the user's ability to manage event schedules and historical data efficiently.

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 along with options to sort in ascending and descending order in each field.
5. Should be able to edit and delete any event.
6. Should be able to search for any event by event name or place.
7. Should be able to view past events along with options to sort in ascending and descending order in each field.

2. TEMPLATE CODE STRUCTURE:

2.1 EVENT CONTROLLER

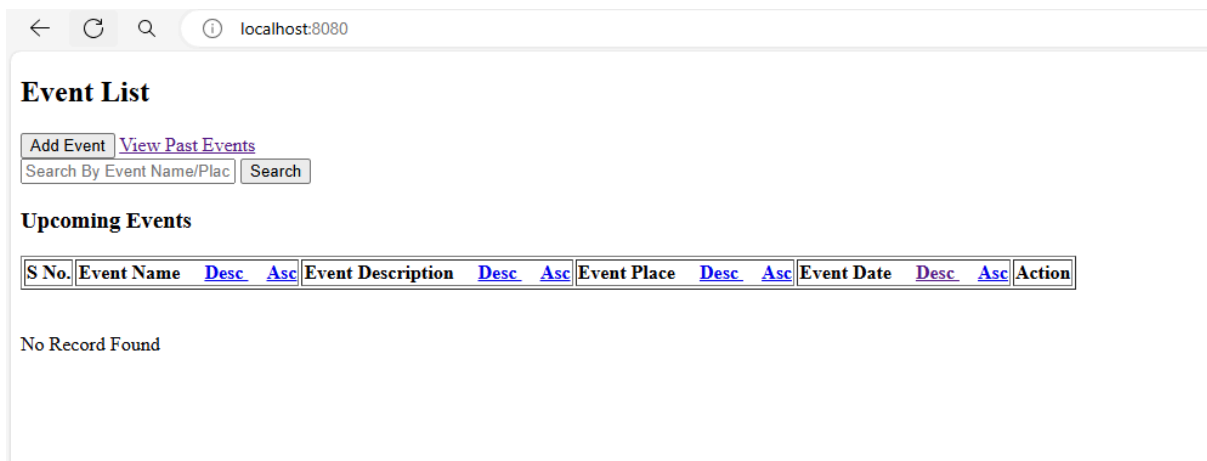
Method Exposed	Purpose
listEvents()	Should return page "list-events" with required data.
showFormForAdd()	Should return page "add-event-form" for adding an event.
saveEvent()	Should save an event and return "event/list" with required data.
showFormForUpdate()	Should show event details in page "update-event-form" 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.
pastEvents()	Should show past event details in the "past-events" page.

2.2 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 an event.
All events	list-events	
Add an event	add-event-form	
Update an event	update-event-form	
Search an event	list-events	
View Past event	past-events	

3 SUGGESTED WIREFRAMES:

1. Homepage – Visitor Landing Page



A screenshot of a web browser showing the 'Event List' page. The browser's address bar displays 'localhost:8080'. The page has a title 'Event List'. Below the title, there are two buttons: 'Add Event' and 'View Past Events'. A search bar is present with the placeholder text 'Search By Event Name/Place' and a 'Search' button. Underneath, the section 'Upcoming Events' is followed by a table header. The table has columns: 'S No.', 'Event Name', 'Event Description', 'Event Place', 'Event Date', and 'Action'. Each of these columns has 'Desc' and 'Asc' links. The table body is empty, and a message 'No Record Found' is displayed below it.

Event List

Add Event View Past Events

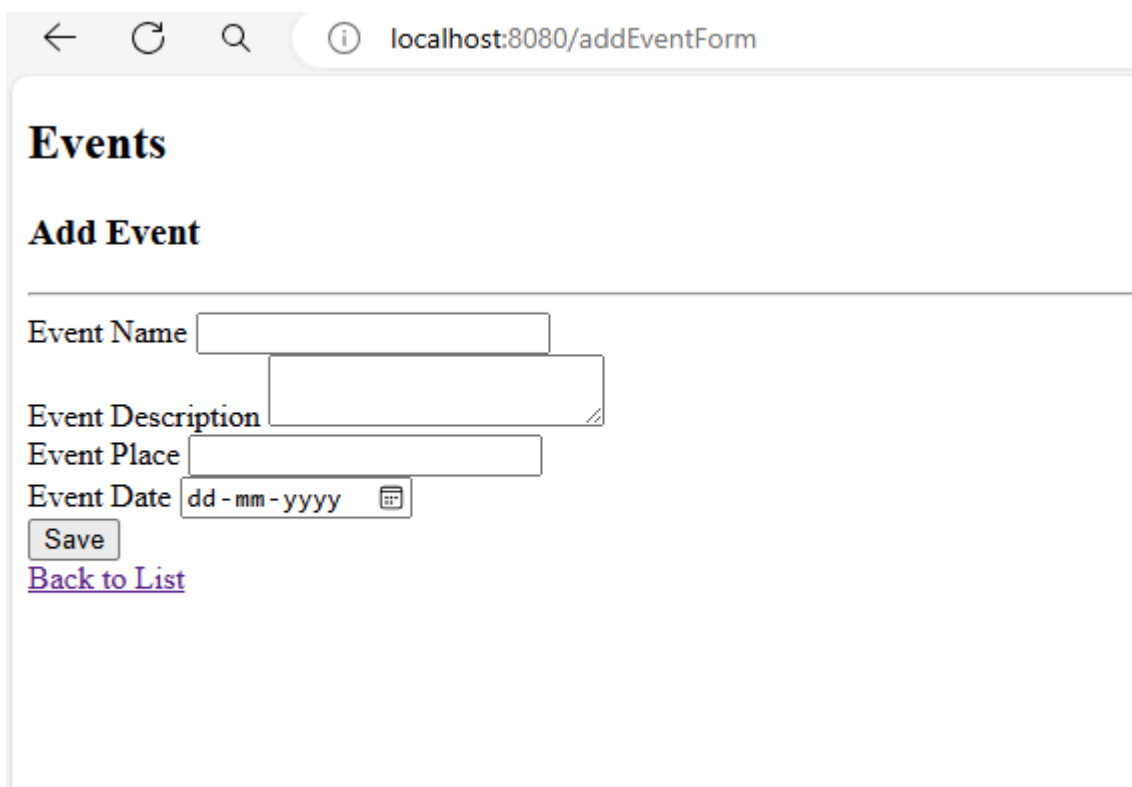
Search By Event Name/Place Search

Upcoming Events

S No.	Event Name	Event Description	Event Place	Event Date	Action

No Record Found

2. Create an Event



A screenshot of a web browser showing the 'Add Event' form. The browser's address bar displays 'localhost:8080/addEventForm'. The page has a title 'Events'. Below the title, the section 'Add Event' is followed by a horizontal line. 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 date picker icon). Below the date field is a 'Save' button and a 'Back to List' link.

Events

Add Event

Event Name

Event Description

Event Place

Event Date dd-mm-yyyy

Save

[Back to List](#)

localhost:8080/addEventForm

Events

Add Event

Event Name

Birthday function

Event Description

1st birthday function,
so to be very grand

Event Place

Noida

Event Date

16-10-2024

Save

[Back to List](#)

3. Event List

localhost:8080/event/list

Event List

Add Event

[View Past Events](#)

Search By Event Name/Plac

Search

Upcoming Events

S No.	Event Name	Desc.	Asc	Event Description	Desc.	Asc	Event Place	Desc.	Asc	Event Date	Desc.	Asc	Action
1	Birthday function			1st birthday function, so to be very grand			Noida			2024-10-16			Update Delete

1

4. Update an Event

localhost:8080/event/updateEventForm?eventId=1

Events

Update Event

Event Name

Birthday function

Event Description

1st birthday function, so to b

Event Place

Noida

Event Date

12-10-2024

Save

[Back to List](#)

localhost:8080/event/list

Event List

Add Event

[View Past Events](#)

Search By Event Name/Plac

Search

Upcoming Events

S No.	Event Name	Desc.	Asc.	Event Description	Desc.	Asc.	Event Place	Desc.	Asc.	Event Date	Desc.	Asc.	Action
1	Birthday function			1st birthday function, so to be very grand			Noida			2024-10-12			Update Delete
2	Marriage			medium budget marriage			Bangalore			2024-06-08			Update Delete

1

5. Delete an Event

← ↻ 🔍

localhost:8080/event/list

Event List

Add Event

View Past Events

Search By Event Name/Plac

Search

Upcoming Events

S No.	Event Name	Desc	Asc	Event Description	Desc	Asc	Event Place	Desc	Asc	Event Date	Desc	Asc	Action
1	Birthday function			1st birthday function, so to be very grand			Noida			2024-10-12			Update Delete
2	Marriage			medium budget marriage			Bangalore			2024-06-08			Update Delete

1

localhost:8080 says

Are you sure you want to delete this event?

OK

Cancel

← ↻ 🔍

localhost:8080/event/list

Event List

Add Event

View Past Events

Search By Event Name/Plac

Search

Upcoming Events

S No.	Event Name	Desc	Asc	Event Description	Desc	Asc	Event Place	Desc	Asc	Event Date	Desc	Asc	Action
1	Birthday function			1st birthday function, so to be very grand			Noida			2024-10-12			Update Delete

1

6. Search an Event

← ↻ 🔍

localhost:8080/event/search

Event List

Add Event

View Past Events

birthday

Search

Upcoming Events

S No.	Event Name	Desc	Asc	Event Description	Desc	Asc	Event Place	Desc	Asc	Event Date	Desc	Asc	Action
1	Birthday function			1st birthday function, so to be very grand			Noida			2024-10-12			Update Delete

1

7. View Past Events

localhost:8080/pastEvents

Past Events

[Back to List](#)

Search By Event Name/Plac

Search

Upcoming Events

S No.	Event Name	Desc	Asc	Event Description	Desc	Asc	Event Place	Desc	Asc	Event Date	Desc	Asc
1	Marriage			medium budget function			Bangalore			2024-01-08		

1

4 BUSINESS VALIDATIONS

1. Id must be of type id.
2. Name value should not be blank, min 2 and max 40 characters.
3. Description value should not be blank, min 2 and max 200 characters.
4. Place value should not be blank.
5. Event date value is not null and should be of format "yyyy/mm/dd"

5 CONSIDERATIONS

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

6 METHOD DESCRIPTIONS

1. Service Class - Method Descriptions

a. EventService – Method Descriptions

- Constructor-based injection is used to initialize **EventRepository**.

Method	Task	Implementation Details
public EventService(Eve	Constructor injection	- Injects EventRepository via constructor

ntRepository eventRepository)		- Initializes eventRepository as private final field
--	--	---

Method	Task	Implementation Details
getEvents(Pageable pageable)	To fetch all upcoming events	<ul style="list-style-type: none"> - Calls eventRepository.findAllUpcomingEvents(LocalDate.now(), pageable) - Returns paginated list of upcoming events
saveEvent(Event event)	To save or update an event	<ul style="list-style-type: none"> - Calls eventRepository.save(event) - Persists new or updated event and returns it
getEvent(Long id)	To fetch event by ID	<ul style="list-style-type: none"> - Calls eventRepository.findById(id).get() - Returns the event associated with given ID
deleteEvent(Long id)	To delete an event by ID	<ul style="list-style-type: none"> - Calls eventRepository.deleteById(id) - Returns true after deletion
searchEvents(String theSearchName, Pageable pageable)	To search upcoming events by name or place	<ul style="list-style-type: none"> - If theSearchName is not empty: <ul style="list-style-type: none"> • Calls eventRepository.findByEventNameAndPlace(...) - Else: <ul style="list-style-type: none"> • Calls findAllUpcomingEvents(...)
pastEvents(String theSearchName, Pageable pageable)	To search past events by name or place	<ul style="list-style-type: none"> - If theSearchName is not empty: <ul style="list-style-type: none"> • Calls eventRepository.findByPastEventNameAndPlace(...) - Else: <ul style="list-style-type: none"> • Calls findAllPastEvents(...)

2. Controller Class - Method Descriptions

a. EventController – Method Descriptions

- Declare a private variable named `eventService` of type `EventService` and inject it using `@Autowired`.

Method	Task	Implementation Details
@Autowired private EventService eventService	Field-based dependency injection	- Annotated with <code>@Autowired</code> - Injects the <code>EventService</code> instance for controller operations
@InitBinder public void InitBinder(WebDa taBinder dataBinder)	Trim whitespace from String inputs	- Registers <code>StringTrimmerEditor</code> to remove leading/trailing whitespaces from all incoming <code>String</code> fields before binding

Method	Task	Implementation Details
listEvents(...))	Display upcoming events	- Calls <code>eventService.getEvents(...)</code> - Adds events and pagination details to model - Returns <code>list-events</code> view
showFormForAdd (...)	Show add-event form	- Adds new <code>Event</code> object to model - Returns <code>add-event-form</code> view
saveEvent(...)	Save or update event with validation	- Uses <code>@Valid</code> with <code>BindingResult</code> for validation - If errors: – Return <code>update-event-form</code> if <code>event.getId() != null</code> – Else return <code>add-event-form</code> - Else saves and redirects to <code>/event/list</code>

showFormForUpdate(...)	Show form for updating event	<ul style="list-style-type: none"> - Fetches event by ID - Adds it to model - Returns update-event-form view
deleteEvent(...)	Delete an event by ID	<ul style="list-style-type: none"> - Calls eventService.deleteEvent(id) - Redirects to /event/list
searchEvents(...)	Search upcoming events by name/place	<ul style="list-style-type: none"> - Calls eventService.searchEvents(...) - Adds search and pagination data to model - Returns list-events view
pastEventsEvents(...)	View/search past events by name/place	<ul style="list-style-type: none"> - Calls eventService.pastEvents(...) - Adds search and pagination data to model - Returns past-events view

7

8 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. 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.
7. 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.

8. 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

9. 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**

NOTE: After typing any of the above commands you might encounter any warnings.

>> Please note that these warnings are expected and can be disregarded. Proceed to the next step.

- 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