System Requirements Specification Index

For

Polling Management System

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

Contents

1 Business-Requirement 3

1.1 Problem Statement 3

1.1.1 Polling Management 3

2 Template Code Structure 4

2.1 Poll Controller 4

2.2 Option Controller 4

2.3 Resources Available 4

3 Suggested Wireframes 6

4 Business Validations 10

5 Considerations 10

6 Execution Steps to Follow 10

1 Business-Requirement:

1.1 Problem Statement:

The Poll Management System provides a streamlined interface for creating and managing polls effectively.

You are tasked with creating a robust platform where users can effortlessly navigate to the homepage, add new polls, each identified by unique poll names. The system organizes all polls in a list that can be sorted by poll names in both ascending and descending order, ensuring easy navigation through entries. Users have the flexibility to modify or remove any poll as needed, enhancing the platform's adaptability to various user requirements. The search functionality allows users to locate specific polls quickly by entering the poll name. Moreover, the system offers advanced features such as adding multiple options to existing polls and the capability to delete these options, providing comprehensive control over poll customization and management.

1.1.1 Polling Management:

The Polling Management allows you to:

1. Access home page
2. Should be able to add more polls.
3. It should have a basic field of poll names to be saved.
4. Should be able to get the poll list along with options to sort in ascending and descending order by poll name.
5. Should be able to edit and delete any poll.
6. Should be able to search for any poll by their poll name.
7. Should be able to add more options for a particular poll and also should be able to delete the added options.

2. Template Code Structure:

2.1 Poll Controller:

Method Exposed	Purpose
addPollsPage()	Should return page “addPolls” for adding a new poll.
home()	Should return "list-polls" with a paginated list of all polls, along with sorting and search features.
showFormForUpdate()	Should return page “updatePolls” for updating an existing poll.
deletePoll()	Should delete a poll and return “poll/list” with required data.
SearchPolls()	Should search a poll and return “list-polls” with required data.
addPollHandleForm()	Should save a poll and return "poll/list" with required data. This method handles both creation and updates based on the poll details.

2.2 Option Controller:

Method Exposed	Purpose
addOptionPage()	Should return page "addOption" for adding options to a specific poll.
addOptionHandleForm()	Should save an option to a specific poll and redirect to: "addOption" with the option added, allowing continuous addition of options.
deletePoll()	Should delete a specific option from a poll and redirect to: "addOption" to continue managing options for the poll.

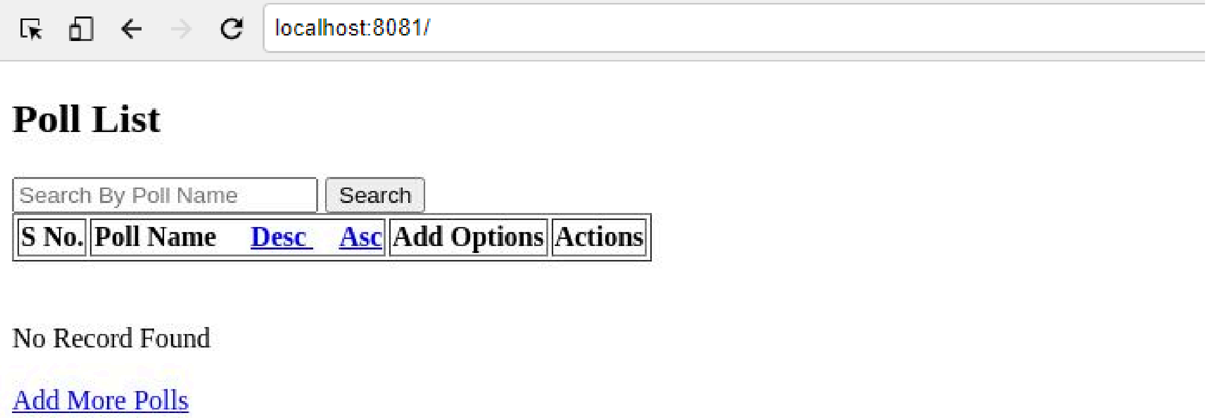
2.3 Resources Available:

Description	View Pages Name	Remarks
Common UI		
Home Page	list-polls	Contains a homepage which shows a list of all polls with options
All Polls	list-polls	
Add a poll	addPolls	
Update a poll	updatePolls	

Search polls	list-polls	to add, edit, delete, and search polls, ensuring easy navigation and interaction.
Add poll option	addOption	Contains a detailed management page for poll options that allows users to add new options to a poll, view all existing options, and delete them as necessary. This integrated approach facilitates seamless interaction and ensures that users can efficiently manage all aspects of poll options from a single, user-friendly interface.
Handle poll option form	addOption (redirects to the same page for continuous addition)	
Delete poll option	addOption (redirects back to addOption after deletion)	

3 Suggested Wireframes:

1. Homepage – Visitor Landing Page



localhost:8081/

Poll List

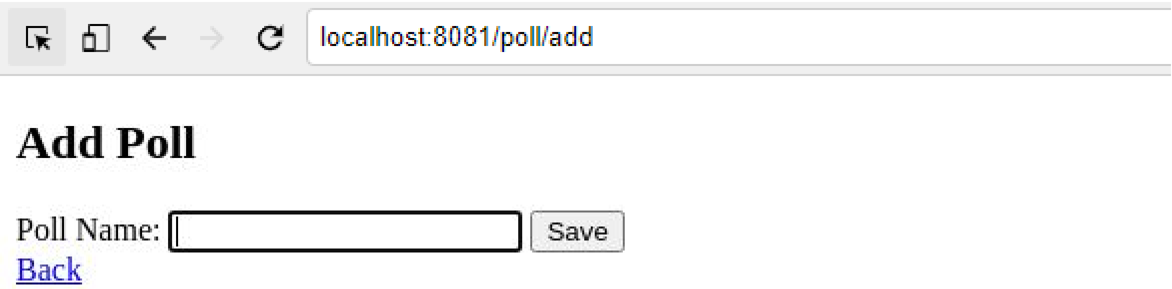
Search By Poll Name | Search

S No.	Poll Name Desc Asc	Add Options	Actions

No Record Found

Add More Polls

2. Add a Poll

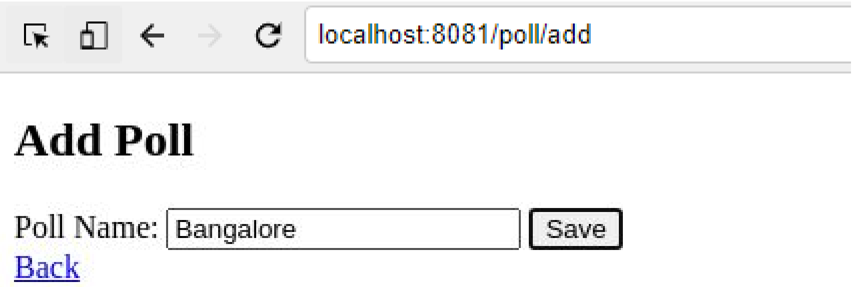


localhost:8081/poll/add

Add Poll

Poll Name: | Save

Back



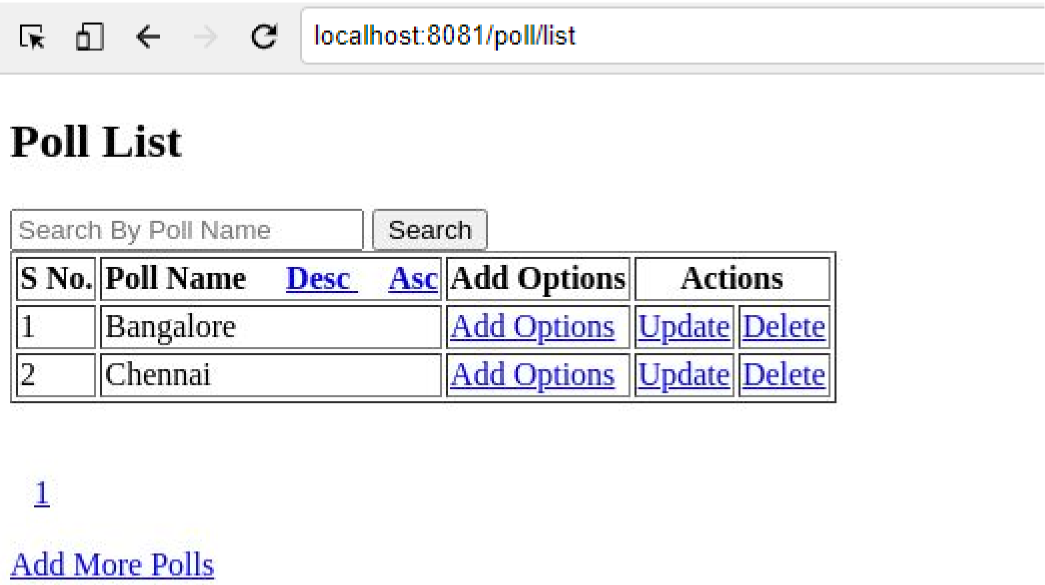
localhost:8081/poll/add

Add Poll

Poll Name: Bangalore | Save

Back

3. Poll List



localhost:8081/poll/list

Poll List

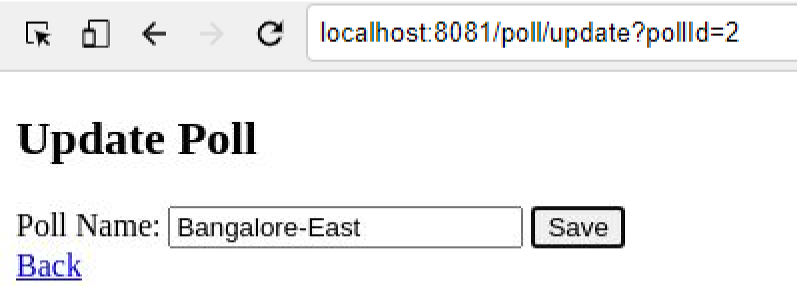
Search By Poll Name

S No.	Poll Name Desc Asc	Add Options	Actions	
1	Bangalore	Add Options	Update	Delete
2	Chennai	Add Options	Update	Delete

1

Add More Polls

4. Update Poll

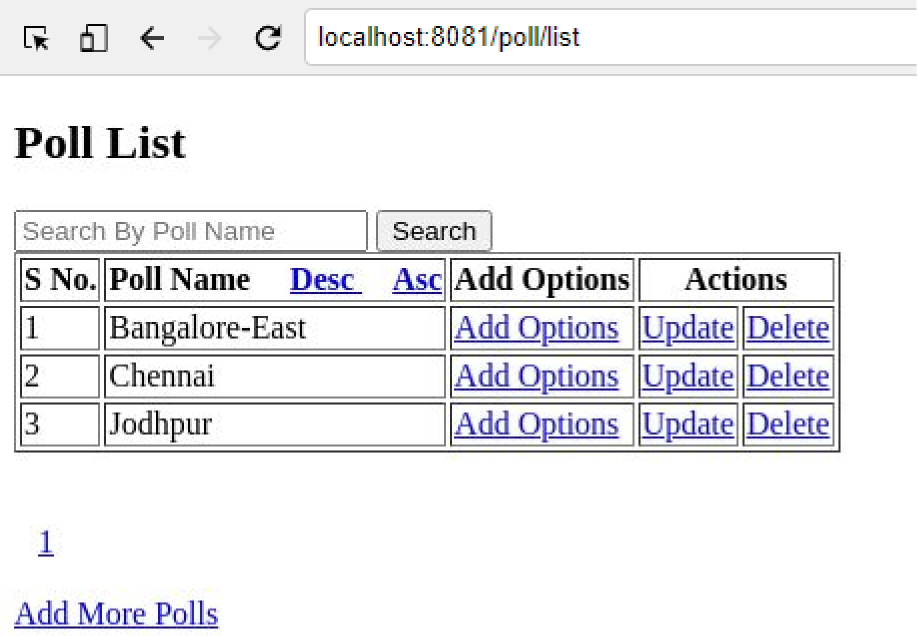


localhost:8081/poll/update?pollId=2

Update Poll

Poll Name: Bangalore-East

Back



localhost:8081/poll/list

Poll List

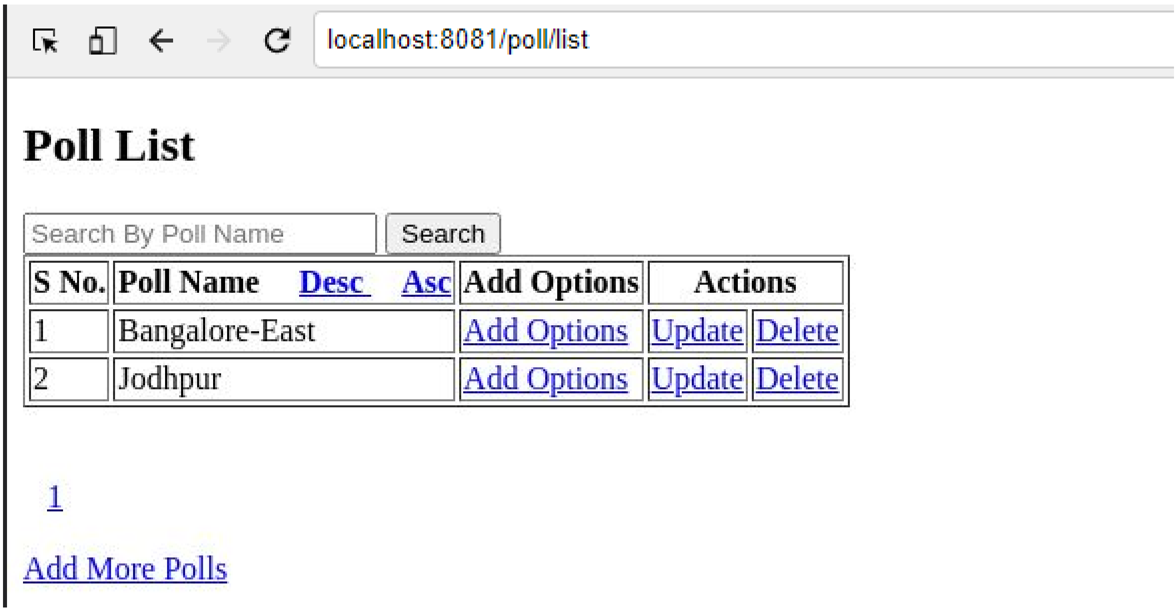
Search By Poll Name | Search

S No.	Poll Name Desc Asc	Add Options	Actions	
1	Bangalore-East	Add Options	Update	Delete
2	Chennai	Add Options	Update	Delete
3	Jodhpur	Add Options	Update	Delete

1

Add More Polls

5. Delete a Poll



localhost:8081/poll/list

Poll List

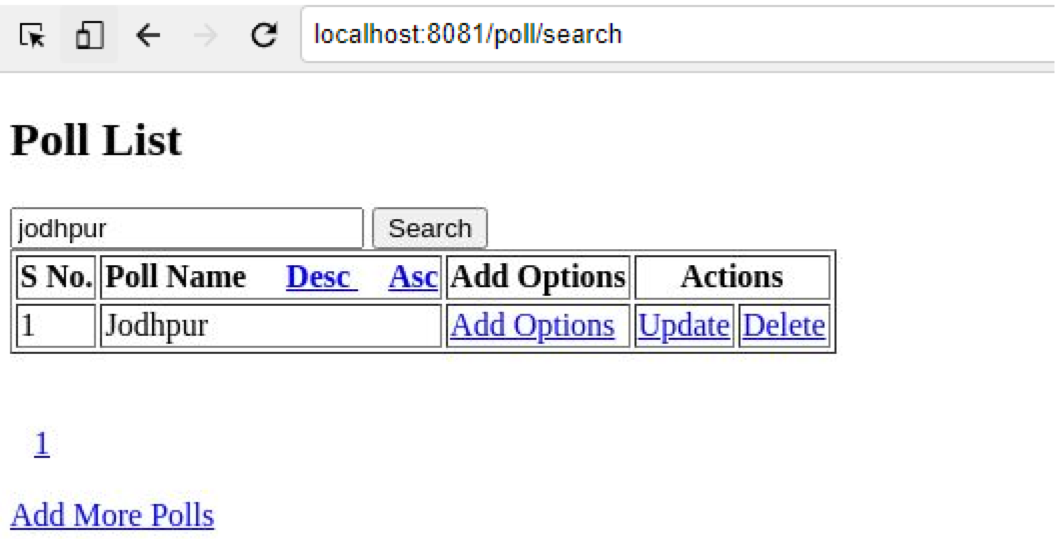
Search By Poll Name | Search

S No.	Poll Name Desc Asc	Add Options	Actions	
1	Bangalore-East	Add Options	Update	Delete
2	Jodhpur	Add Options	Update	Delete

1

Add More Polls

6. Search a Poll



localhost:8081/poll/search

Poll List

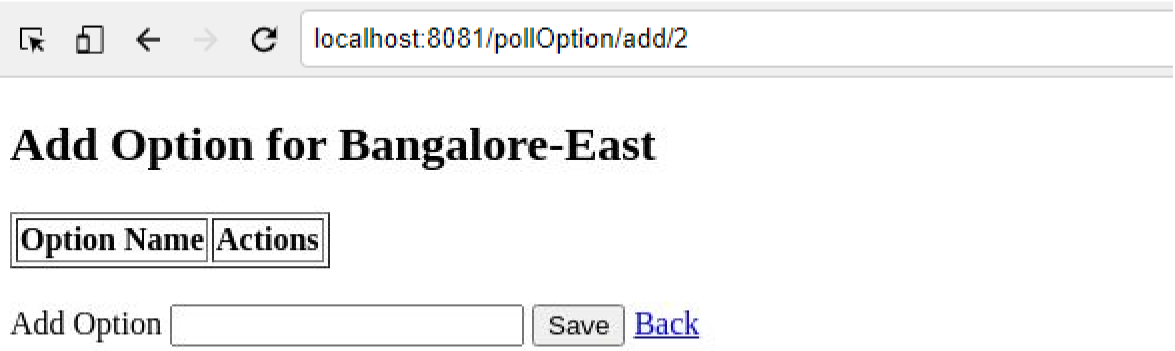
jodhpur

S No.	Poll Name Desc Asc	Add Options	Actions	
1	Jodhpur	Add Options	Update	Delete

1

Add More Polls

7. Add Option

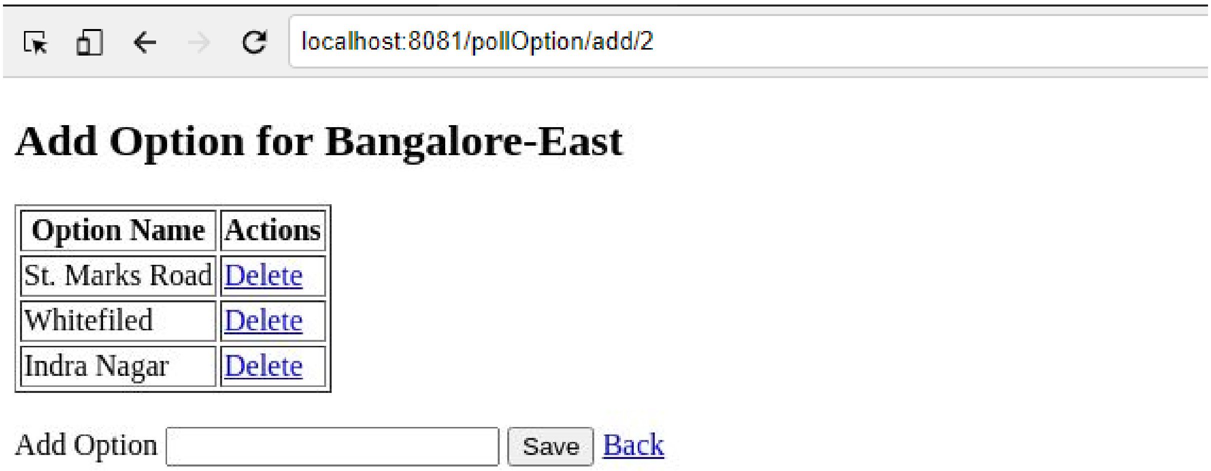


localhost:8081/pollOption/add/2

Add Option for Bangalore-East

Option Name	Actions

Add Option



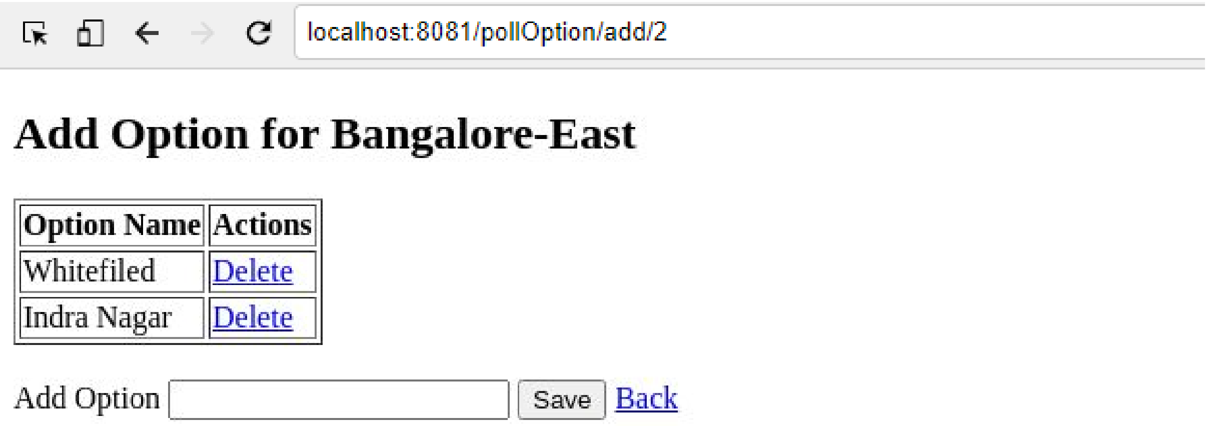
localhost:8081/pollOption/add/2

Add Option for Bangalore-East

Option Name	Actions
St. Marks Road	Delete
Whitefiled	Delete
Indra Nagar	Delete

Add Option

8. Delete Option


localhost:8081/pollOption/add/2

Add Option for Bangalore-East

Option Name	Actions
Whitefiled	Delete
Indra Nagar	Delete

Add Option [ ] Save Back

4 Business Validations

1. Id must be of type id.
2. Full name value should not be blank, min 2 and max 40 characters.
3. Gender is not null.
4. Address should not be blank, min 2 and max 100 characters.
5. Religion should not be blank.
6. Occupation should not be blank.

5 Considerations

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

6 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

       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
    ```
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.