

---

# System Requirements Specification Index

For

## Recipe Book

Version 1.0

IIHT Pvt. Ltd.  
fullstack@iiht.com

## Contents

1	Business-Requirement:	3
1.1	Problem Statement:	3
1.1.1	Recipe Book	3
2.	Template Code Structure	
1.1	Recipe 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 all recipes of a user. Where users can add, edit and delete their recipes.

### 1.1.1 Recipe Book:

The Recipe Book allows you to:

1. Access the home page.
2. Should be able to add a new recipe.
3. It can have basic fields like title, description, ingredients and cookTime.
4. Should be able to get the list of recipes.
5. Should be able to edit and delete any recipe.
6. Should be able to search for an recipe.

## 2. TEMPLATE CODE STRUCTURE:

### 2.1 RECIPE CONTROLLER

Method Exposed	Purpose
listRecipes()	Should return page "list-recipes" with required data.
showFormForAdd()	Should return page "add-recipe-form" for adding an event.
saveRecipe()	Should save an event and return "recipe/list" with required data.
showFormForUpdate()	Should show event details in page "update-recipe-form" to edit an event.
deleteRecipe()	Should delete a recipe and return "recipe/list" with required data.
searchRecipes()	Should search a recipe and return "list-recipes" with required data.

## 3. RESOURCES AVAILABLE:

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

### 3 SUGGESTED WIREFRAMES:

---

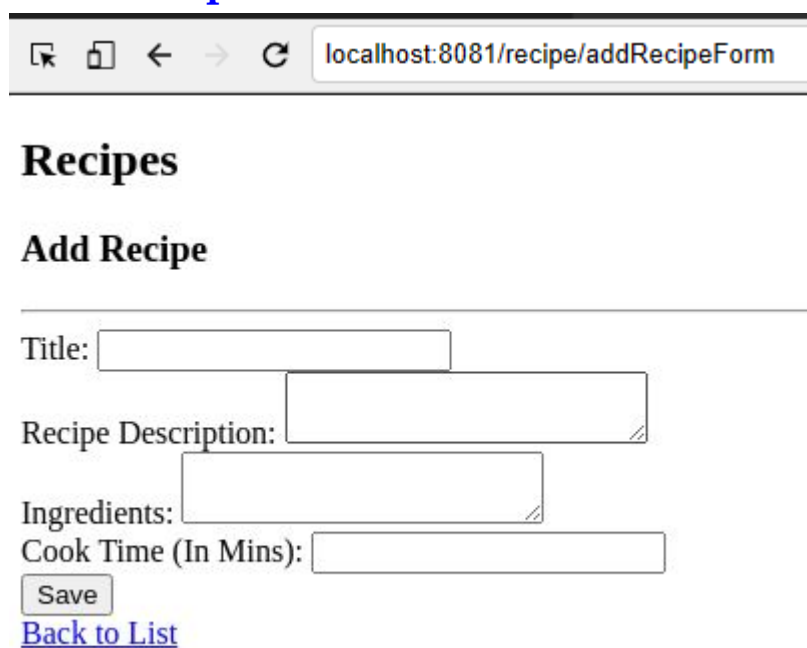
#### 1. Homepage – Visitor Landing Page








A browser window showing the URL `localhost:8081/recipe/list`. Below the browser, the page title is **Recipe List**. There is an **Add Recipe** button, a search bar with the placeholder text "Search By Recipe Name", and a **Search** button. Below these is a table with the following headers: #, Title, Recipe Description, Ingredients, Cook Time (In Mins), and Action.

#	Title	Recipe Description	Ingredients	Cook Time (In Mins)	Action
---	-------	--------------------	-------------	---------------------	--------

#### 2. Create a Recipe



A browser window showing the URL `localhost:8081/recipe/addRecipeForm`. The page title is **Recipes**. Below the title is the section **Add Recipe**. There are four input fields: **Title:**, **Recipe Description:**, **Ingredients:**, and **Cook Time (In Mins):**. Below the input fields is a **Save** button and a [Back to List](#) link.



localhost:8081/recipe/addRecipeForm

## Recipes

### Add Recipe




Title:

Recipe Description:

Ingredients:

Cook Time (In Mins):

[Back to List](#)



localhost:8081/recipe/list

## Recipe List

#	Title	Recipe Description	Ingredients	Cook Time (In Mins):	Action
1	abc	xyz qwe nbv	sdf ghj bnm	120 Mins	<a href="#">Update</a> <a href="#">Delete</a>

## BUSINESS VALIDATIONS

1. Id must be of type id.
2. Title value not blank, min 2 and max 40 characters.
3. Description value not blank, min 2 and max 200 characters.
4. Ingredients value not blank.
5. Cooktime value not null.

## 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.