# System Requirements Specification Index

For

# To-Do Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

# Contents


1 Business-Requirement: 3

1.1 Problem Statement: 3

1.1.1 To-Do Application 3

2. Template Code Structure 4

2.1 ToDo Controller 4

2.2 Resources Available: 4

3 Suggested Wireframe: 5

4 Business Validations 10

5 Considerations 10

6 Execution Steps to Follow 10

# 1 BUSINESS-REQUIREMENT:

## 1.1 PROBLEM STATEMENT:

The Todo Application is designed to help users manage their daily tasks effectively.

You are tasked to develop a platform that provides access to a home page and enables users to add new todos with fields such as task, task description, and taskDate. It should offer features to edit and delete any todo entry. Additionally, a search function should also be included, allowing users to locate specific todos by name, description, or date, thereby enhancing the user's ability to organize and prioritize their tasks efficiently.

### 1.1.1 ToDo App:

**The Todo Application** allows you to:

1. Access the home page.
2. Should be able to add a new todo.
3. It can have basic fields like task, task description and taskDate.
4. Should be able to get the list of todos along with options to sort in ascending and descending order in each field.
5. Should be able to edit and delete any todo.
6. Should be able to search for a todo by name or description or date.

# 2. Template Code Structure:

## 2.1 Todo Controller

| Method Exposed | Purpose |
|---|---|
| **listTodos()** | Should return page "list-todos" with required data. |
| **showFormForAdd()** | Should return page "add-todo-form" for adding an event. |
| **saveTodo()** | Should save a todo and return "todo/list" with required data. |
| **showFormForUpdate()** | Should show todo details in page "update-todo-form" to edit a todo. |
| **deleteTodo()** | Should delete a todo and return "todo/list" with required data. |
| **searchTodos()** | Should search a todo and return "list-todos" with required data. |
| **updateStatus()** | Should show and update the current status of todo task (Pending/Completed/Decline) |

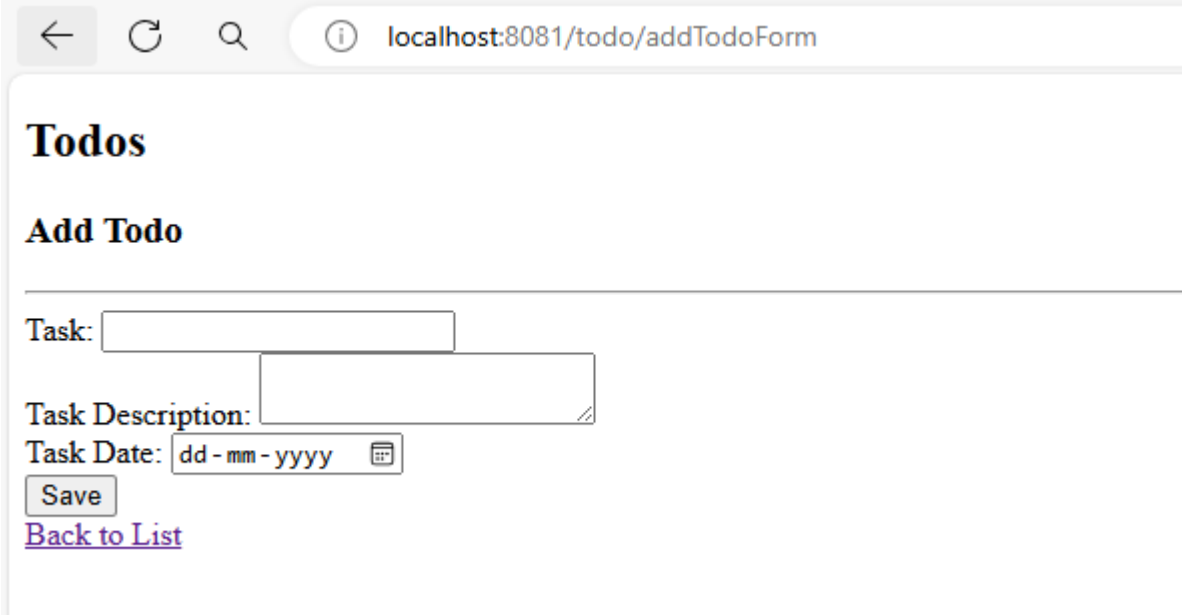
## 2.2 Resources Available:

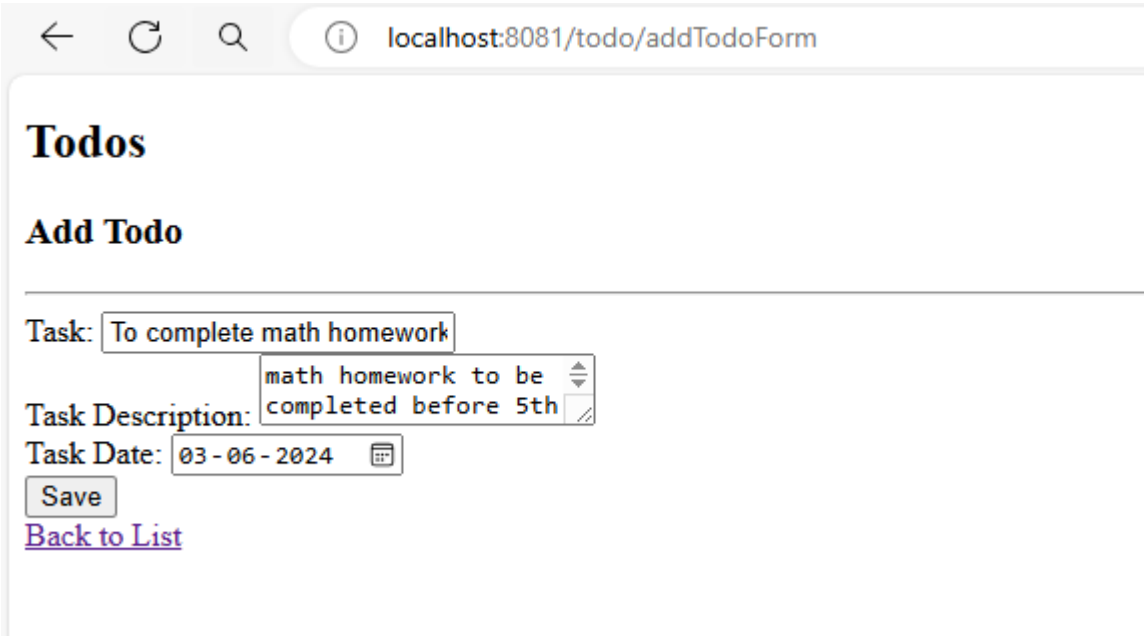
| Description | View Pages Name | Remarks |
|---|---|---|
| Common UI | | |
| Home Page | list-todos | Contains a homepage which shows a list of all todos along with options to add, edit , delete, search and update status of a todo. |
| All todos | list-todos | |
| Add a todo | add-todo-form | |
| Update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

# 3 SUGGESTED WIREFRAMES:

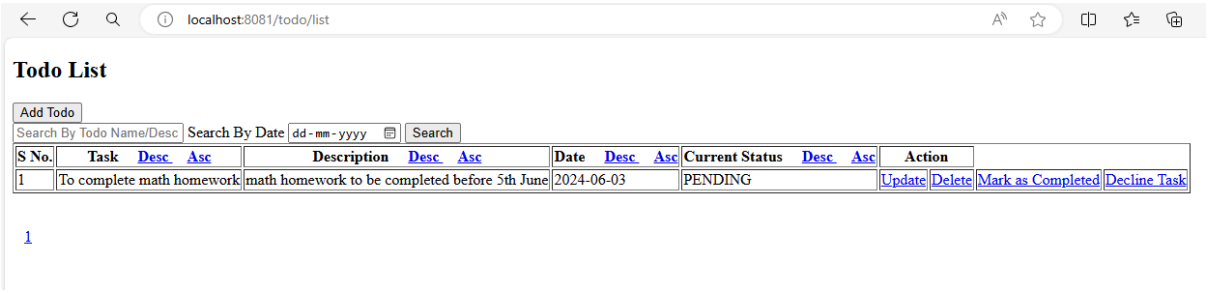
## 1. Homepage – Visitor Landing Page


## 2. Create a Todo

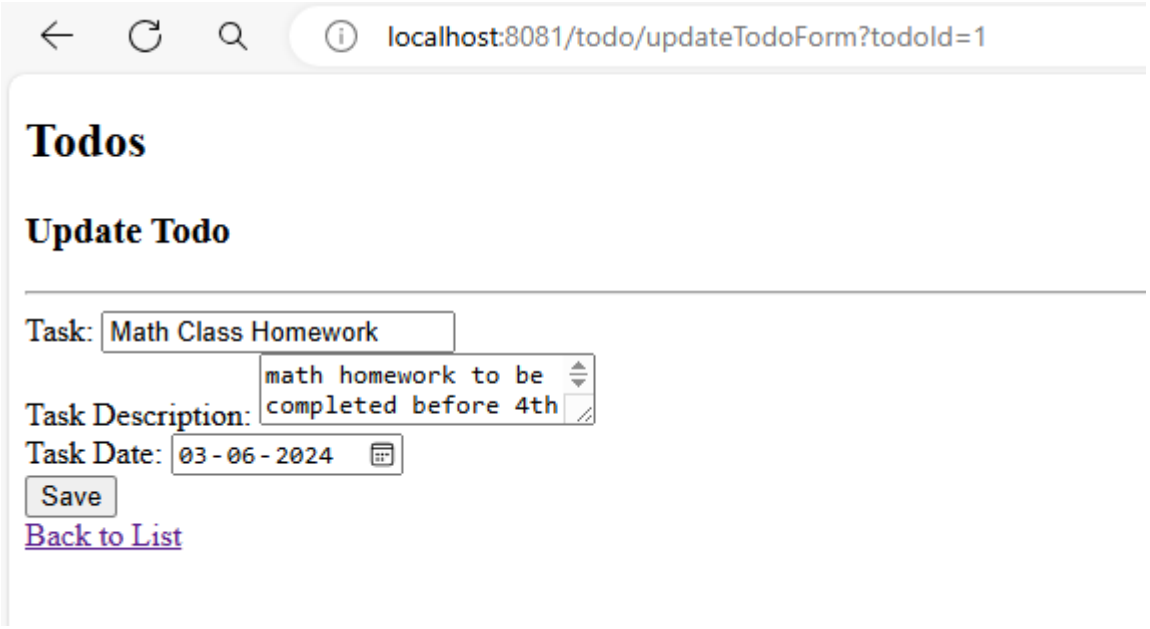


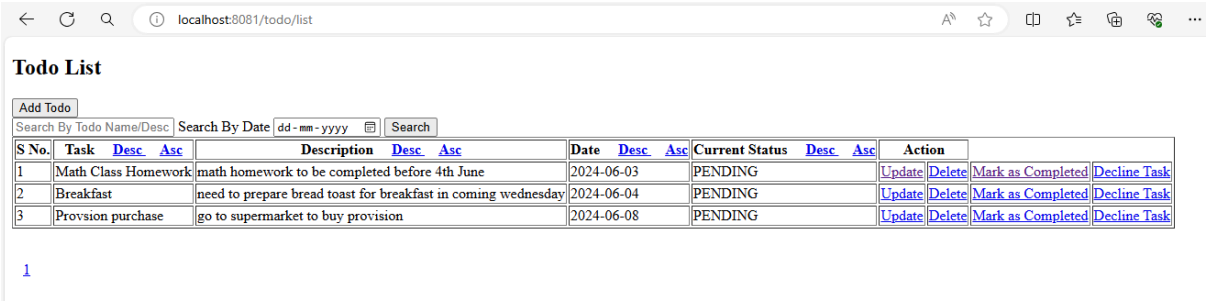

localhost:8081/todo/addTodoForm
Todos
Add Todo
Task: To complete math homework
Task Description: math homework to be completed before 5th
Task Date: 03-06-2024
Save
Back to List

## 3. Todo List

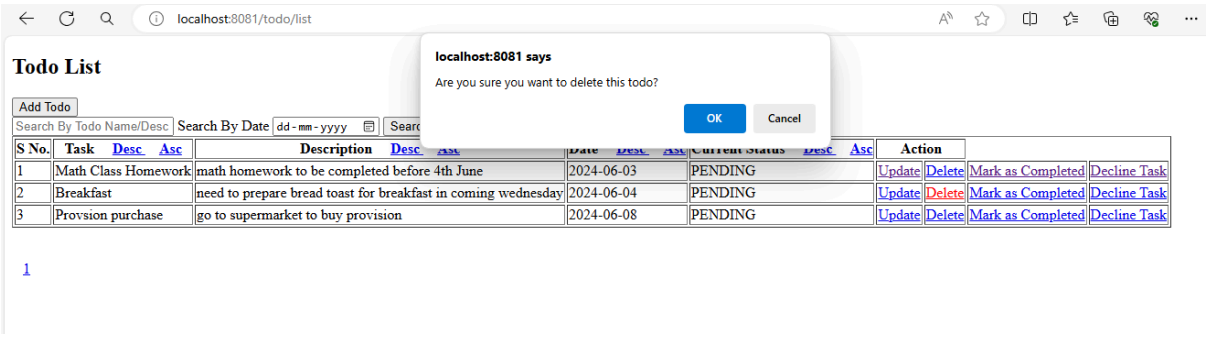

localhost:8081/todo/list
Todo List
Add Todo
Search By Todo Name/Desc
Search By Date dd-mm-yyyy
Search
S No. Task Desc Asc Description Desc Asc Date Desc Asc Current Status Desc Asc Action
1 To complete math homework math homework to be completed before 5th June 2024-06-03 PENDING Update Delete Mark as Completed Decline Task
1

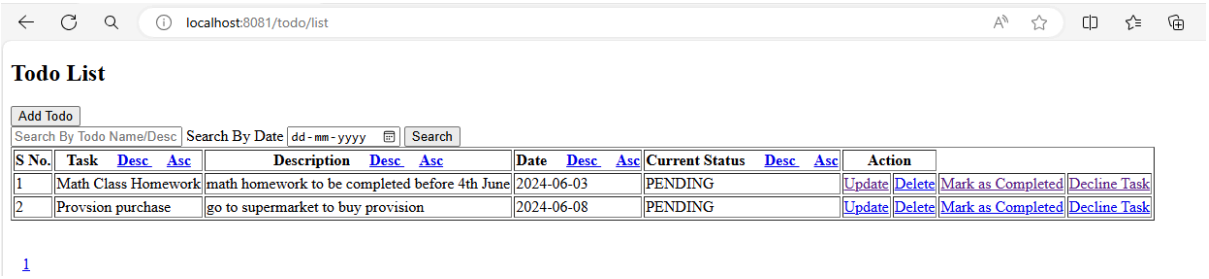
# 4. Update a Todo



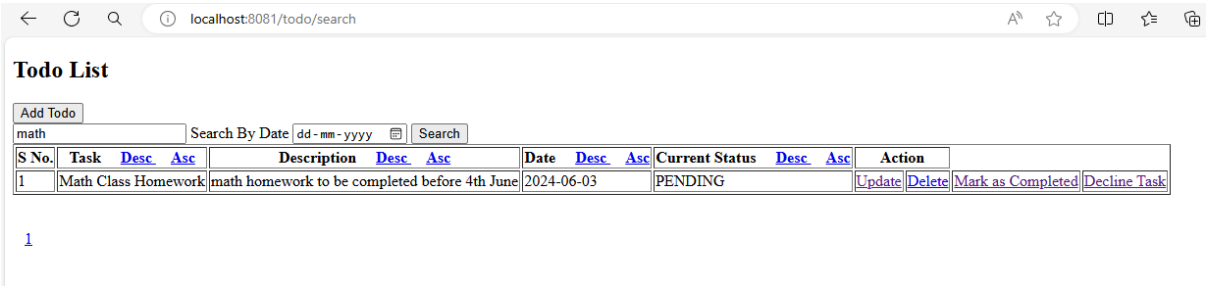
# 5. Delete a Todo



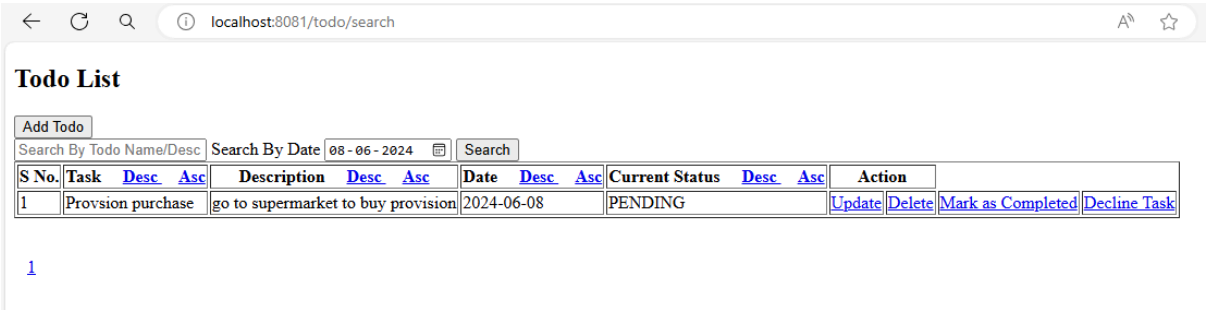




# 6. Search a Todo

*****By Name*****

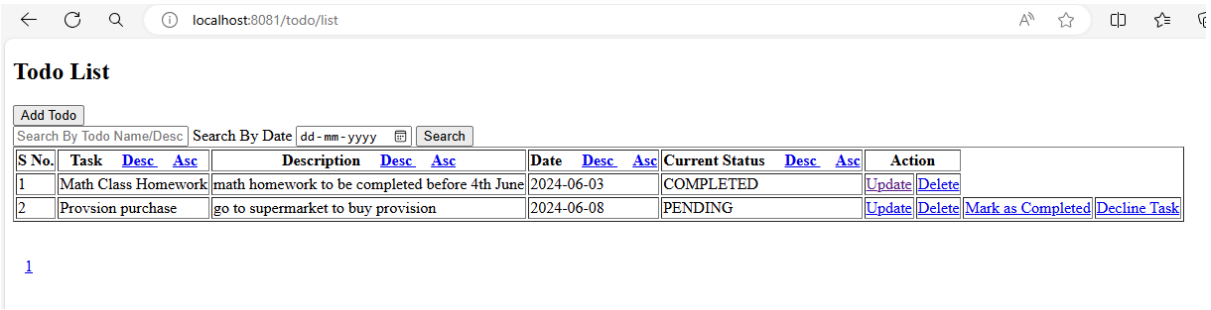



*****By Date*****

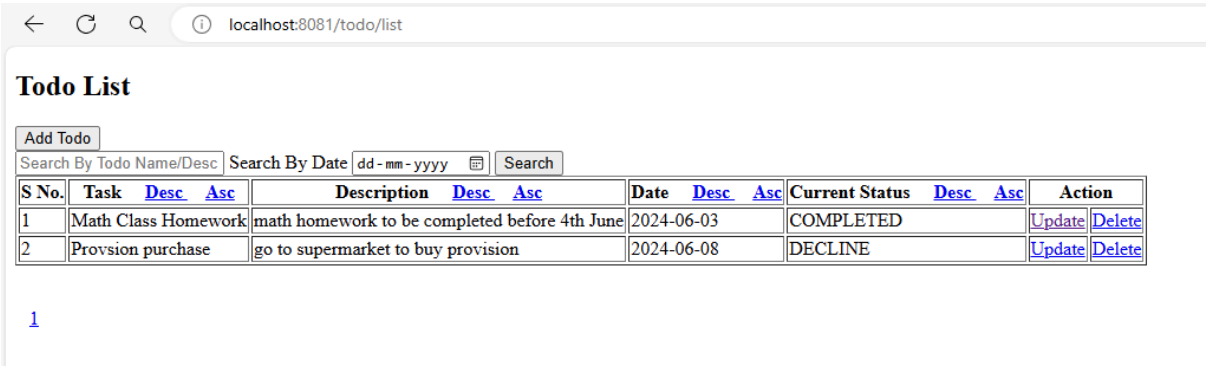


# 7. Todo Status

***Completed***




***Decline***



# 4 Business Validations

1. Id must be of type id.
2. Task value should not be blank, min 2 and max 200 characters.
3. Description value should not be blank, min 2 and max 200 characters.
4. Task date value is not null and in 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 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 the second sql command (sudo systemctl start mysql), you may encounter a warning message like :**

    **System has not been booted with systemd as init system (PID 1). Can't operate. Failed to connect to bus: Host is down**

    **>> Please note that this warning is 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.**