# System Requirements Specification Index

For

## **Laptop Store Application**

Version 1.0



## Contents

1.	Business-Requirement	3
1	1 Problem Statement	3
	1.1.1 Laptop Store Application	3
2.	Template Code Structure	4
2	.1 Laptop 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 Laptop Store App is designed to provide a user-friendly interface for managing a diverse collection of laptops.

You are tasked to create a user-friendly platform that enables users to access the homepage to add new laptops, entering essential details such as laptop name, price, brand, storage capacity, RAM, and processor specifications. The system allows users to view a comprehensive list of all laptops. Functionality extends to editing and deleting laptop entries, as well as conducting targeted searches by laptop name. Additionally, users can update the status of each laptop, marking it as a best seller or not, to reflect its popularity and sales performance.

#### 1.1.1 Laptop Store Application:

#### The Laptop Store Application allows you to:

- 1. Access the home page.
- 2. Should be able to add a new laptop.
- 3. It should have basic fields like laptop name, price, brand, storage, ram and processor.
- 4. Should be able to get the list of laptops along with options to sort in ascending and descending order in each field.
- 5. Should be able to edit and delete any laptop.
- 6. Should be able to search for a laptop by laptop name.
- 7. Should be able to update the status of a laptop whether a best seller or not.

## 2. TEMPLATE CODE STRUCTURE:

## 2.1 LAPTOPCONTROLLER:

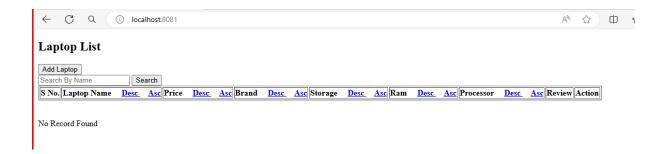
Method Exposed	Purpose
listLaptops()	Should return page "list-laptops" with required
	data.
showFormForAdd()	Should return page "add-laptop-form" for adding a
	policy.
saveLaptop()	Should save a laptop and return "laptop/list" with
	required data.
showFormForUpdate()	Should show laptop details in page
	"update-laptop-form" to edit a laptop.
deleteLaptop()	Should delete a laptop and return "laptop/list" with
	required data.
searchLaptops()	Should search for a laptop and return "list-laptops"
	with required data.
updateBestSeller()	Should show the status of a laptop whether a best
	seller or not.

## 2.2 Resources Available:

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

#### 3 SUGGESTED WIREFRAMES:

## 1. Homepage – Visitor Landing Page



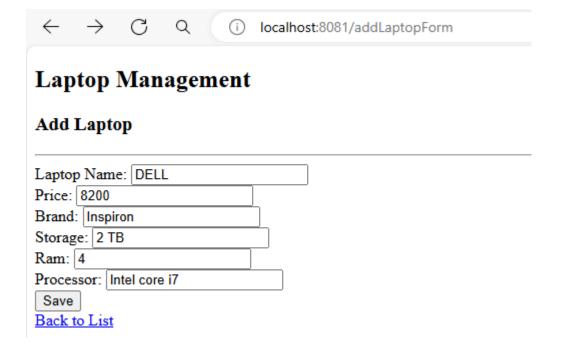
## 2. Create a Laptop



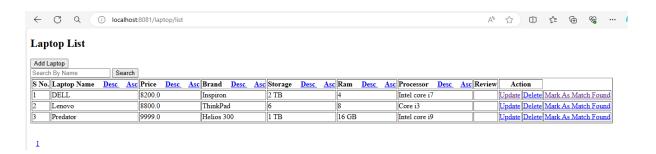
## Laptop Management

#### Add Laptop

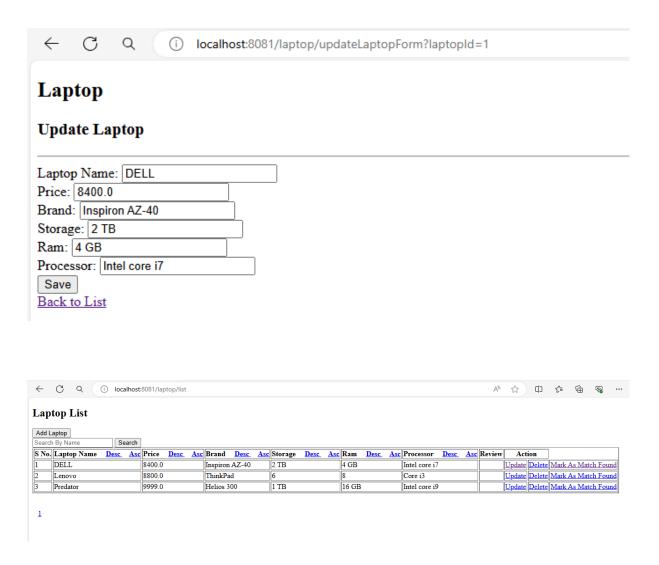
Laptop Nan	ne:	
Price:		
Brand:		
Storage:		
Ram:		
Processor:		
Save		
Back to Lis	<u>t</u>	



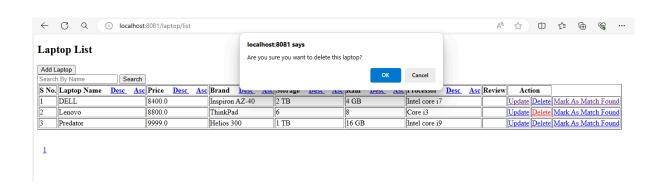
## 3. Laptop List

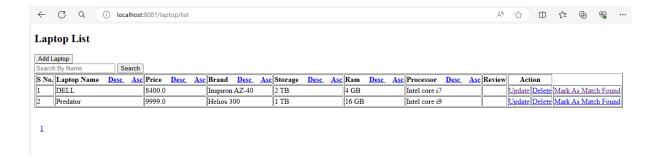


## 4. Update a Laptop

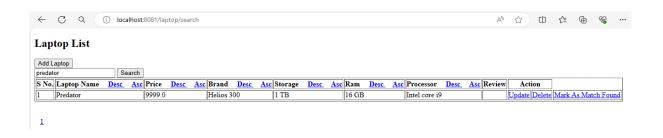


## 5. Delete a Laptop

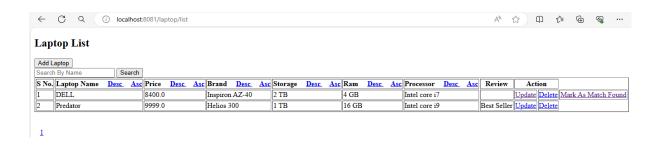




## 6. Search a Laptop



## 7. Laptop Status



#### 4 Business Validations

- 1. Id must be of type id.
- 2. Name value should not be blank, min 3 and max 20 characters.
- 3. Price value should not be null, min 0 and max 9999.
- 4. Brand value should not be blank.
- 5. Storage value should not be blank.
- 6. Ram value should not be blank.
- 7. Processor value 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.