

---

# System Requirements Specification Index

For

## Insurance Policy Application

Version 1.0

IIHT Pvt. Ltd.  
fullstack@iiht.com

## Contents

1	Business-Requirement:	3
1.1	Problem Statement	3
1.1.1	Insurance Policy Application	3
2.	Template Code Structure	
2.1	Insurance Policy Controller	
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 purpose of this application is to provide a platform to find all insurance policies of all types. Where a company can add, edit, delete and search any policy.

The Insurance Policy Application is designed to streamline the management of insurance policies for users.

You are tasked to create a user-friendly platform that allows access to a home page and enables the addition of new insurance policies. Each policy should include fields such as firstName, lastName, amount, interest, tenure, startDate, nominee, and policyName. The application should allow users to view a comprehensive list of all policies. It should also include functionalities to edit and delete any policy. A search feature should be integrated, making it possible for users to locate specific policies by person name or policy name, enhancing the efficiency of managing insurance details.

### 1.1.1 Insurance Policy Application:

**The Insurance Policy Application** allows you to:

1. Access the home page.
2. Should be able to add new insurance.
3. It can have basic fields like firstName, lastName, amount, interest, tenure, startDate, nominee and policyName.
4. Should be able to get the list of policies along with options to sort in ascending and descending order in each field.
5. Should be able to edit and delete any policy.
6. Should be able to search for a policy by person name or policy name.

## 2. TEMPLATE CODE STRUCTURE:

### 2.1 INSURANCEPOLICY CONTROLLER

---

Method Exposed	Purpose
<code>listInsurancePolicies()</code>	Should return page "list-insurancePolicies" with required data.
<code>showFormForAdd()</code>	Should return page "add-insurancePolicy-form" for adding a policy.
<code>saveInsurancePolicy()</code>	Should save a tutor and return "insurancePolicy/list" with required data.
<code>showFormForUpdate()</code>	Should show policy details in page "update-insurancePolicy-form" to edit a tutor.
<code>deleteInsurancePolicy()</code>	Should delete a policy and return "insurancePolicy/list" with required data.
<code>searchInsurancePolicies()</code>	Should search for a policy and return "list-insurancePolicies" with required data.
<code>updateAvailability()</code>	Should show and update the status of insurance policy (Claim Now/Policy Claimed)

### 2.2 RESOURCES AVAILABLE:

---

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

### 3 SUGGESTED WIREFRAMES:

#### 1. Homepage – Visitor Landing Page



A screenshot of a web browser showing a page titled "Insurance Policy List". The browser's address bar shows "localhost:8081". The page has a header with a left arrow, a refresh icon, a search icon, and a status icon. Below the header, there is a button labeled "Add InsurancePolicy". Underneath the button is a search bar with the placeholder text "Search By Name/Policy Name" and a "Search" button. Below the search bar is a table with the following columns: "S No.", "First Name", "Last Name", "Policy Name", "Amount", "Interest", "Tenure", "Start Date", "Nominee", and "Action". Each of the first eight columns has "Desc" and "Asc" links next to it. The table is currently empty, and the text "No Record Found" is displayed below it.

Insurance Policy List

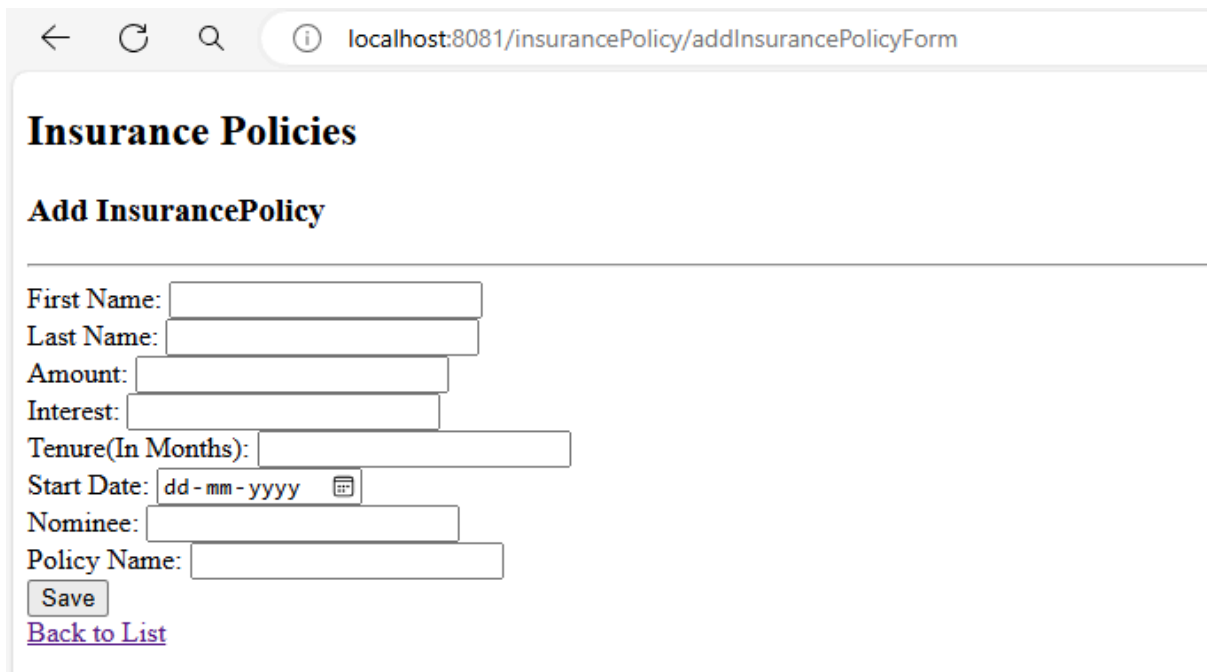
Add InsurancePolicy

Search By Name/Policy Name Search

S No.	First Name	Last Name	Policy Name	Amount	Interest	Tenure	Start Date	Nominee	Action
-------	------------	-----------	-------------	--------	----------	--------	------------	---------	--------

No Record Found

#### 2. Create an Insurance



A screenshot of a web browser showing a page titled "Insurance Policies". The browser's address bar shows "localhost:8081/insurancePolicy/addInsurancePolicyForm". The page has a header with a left arrow, a refresh icon, a search icon, and a status icon. Below the header, there is a button labeled "Add InsurancePolicy". Below the button is a form with the following fields: "First Name:", "Last Name:", "Amount:", "Interest:", "Tenure(In Months):", "Start Date:" (with a date picker icon), "Nominee:", and "Policy Name:". Below the form is a "Save" button and a link labeled "Back to List".

Insurance Policies

Add InsurancePolicy

First Name:

Last Name:

Amount:

Interest:

Tenure(In Months):

Start Date: dd-mm-yyyy

Nominee:


Policy Name:

Save

[Back to List](#)

localhost:8081/insurancePolicy/addInsurancePolicyForm

### Add InsurancePolicy

First Name:	<input type="text" value="John"/>
Last Name:	<input type="text" value="Doe"/>
Amount:	<input type="text" value="500000"/>
Interest:	<input type="text" value="2"/>
Tenure(In Months):	<input type="text" value="60"/>
Start Date:	<input type="text" value="04 - 06 - 2024"/> 
Nominee:	<input type="text" value="Mercy"/>
Policy Name:	<input type="text" value="New-senior citizen policy"/>
<input type="button" value="Save"/>	
<a href="#">Back to List</a>	

#### 4. Update an Insurance

localhost:8081/insurancePolicy/updateInsurancePolicyForm?insurancePolicyId=1

## InsurancePolicies

### Update Insurance Policy

---


First Name:

Last Name:

Amount:

Interest:

Tenure(In Months):

Start Date:  

Nominee:

Policy Name:

[Back to List](#)

S No.	First Name <small>Desc Asc</small>	Last Name <small>Desc Asc</small>	Policy Name <small>Desc Asc</small>	Amount <small>Desc Asc</small>	Interest	Tenure	Start Date <small>Desc Asc</small>	Nominee	Action
1	John	Doe	New-senior citizen policy	500000.0	1	68	2024-06-04	Mercy M	<a href="#">Update</a> <a href="#">Delete</a> <a href="#">Claim Now</a>
2	Mary	lilly	Family 54 policy	100000.0	1	55	2024-06-06	Alex	<a href="#">Update</a> <a href="#">Delete</a>

Policy not applied yet

# 5. Delete an Insurance

←↻🔍localhost:8081/insurancePolicy/list

Insurance Policy List

Add InsurancePolicy

Search By Name/Policy NameSearch

S No.	First NameAscDesc	Last NameAscDesc	Policy NameAscDesc	AmountDescAsc	Interest	Tenure	Start DateAscDesc	Nominee	Action	
1	John	Doe	New-senior citizen policy	500000.0	1	68	2024-06-04	Mercy M	<a href="#">Update</a> <a href="#">Delete</a> <a href="#">Claim Now</a>	
2	Mary	lilly	Family 54 policy	100000.0	1	55	2024-06-06	Alex	<a href="#">Update</a> <a href="#">Delete</a>	Policy not applied yet
3	Liam	Levi	Health insurance policy	60000.0	2	12	2024-06-04	Jose	<a href="#">Update</a> <a href="#">Delete</a> <a href="#">Claim Now</a>	

1

localhost:8081 says

Are you sure you want to delete this insurancePolicy?

OK

Cancel

←↻🔍localhost:8081/insurancePolicy/list

Insurance Policy List

Add InsurancePolicy

Search By Name/Policy NameSearch

S No.	First NameAscDesc	Last NameAscDesc	Policy NameAscDesc	AmountDescAsc	Interest	Tenure	Start DateAscDesc	Nominee	Action	
1	John	Doe	New-senior citizen policy	500000.0	1	68	2024-06-04	Mercy M	<a href="#">Update</a> <a href="#">Delete</a> <a href="#">Claim Now</a>	
2	Liam	Levi	Health insurance policy	60000.0	2	12	2024-06-04	Jose	<a href="#">Update</a> <a href="#">Delete</a> <a href="#">Claim Now</a>	

1



## 6. Search an Insurance

\*\*\*By First/Last Name\*\*\*

←↻🔍📍localhost:8081/insurancePolicy/search

Insurance Policy List

Add InsurancePolicy

john

Search

S No.	First Name	Desc.	Asc.	Last Name	Desc.	Asc.	Policy Name	Desc.	Asc.	Amount	Desc.	Asc.	Interest	Tenure	Start Date	Desc.	Asc.	Nominee	Action
1	John			Doe			New-senior citizen policy			500000.0			1	68	2024-06-04			Mercy M	<a href="#">Update</a> <a href="#">Delete</a> <a href="#">Claim Now</a>

1

\*\*\*By Policy Name\*\*\*

←↻🔍📍localhost:8081/insurancePolicy/search

Insurance Policy List

Add InsurancePolicy

health

Search

S No.	First Name	Desc.	Asc.	Last Name	Desc.	Asc.	Policy Name	Desc.	Asc.	Amount	Desc.	Asc.	Interest	Tenure	Start Date	Desc.	Asc.	Nominee	Action
1	Liam			Levi			Health insurance policy			60000.0			2	12	2024-06-04			Jose	<a href="#">Update</a> <a href="#">Delete</a> <a href="#">Claim Now</a>

1

## 7. Claim an Insurance Policy

←↻🔍📍localhost:8081/insurancePolicy/list

Insurance Policy List

Add InsurancePolicy

Search By Name/Policy Name

Search

S No.	First Name	Desc.	Asc.	Last Name	Desc.	Asc.	Policy Name	Desc.	Asc.	Amount	Desc.	Asc.	Interest	Tenure	Start Date	Desc.	Asc.	Nominee	Action
1	John			Doe			New-senior citizen policy			500000.0			1	68	2024-06-04			Mercy M	<a href="#">Update</a> <a href="#">Delete</a> <a href="#">Policy Claimed</a>
2	Liam			Levi			Health insurance policy			60000.0			2	12	2024-06-04			Jose	<a href="#">Update</a> <a href="#">Delete</a> <a href="#">Claim Now</a>

1

## 4 BUSINESS VALIDATIONS

---

1. Id must be of type id.
  2. First name value should not be blank, min 3 and max 50 characters.
  3. Last name value should not be blank, min 3 and max 50 characters.
  4. Amount value is not null.
  5. Interest value is not null.
  6. Tenure value is not null.
  7. Start date value is not null and in format (yyyy-mm-dd).
  8. Nominee should not be blank.
  9. Policy Name 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 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.