# System Requirements Specification Index

For

# Shipping Charges-Junit

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

# Table of Contents

# Shipping Charges
## System Requirements Specification

## 1 Project Abstract

The **Java-Shipping Charges** project presents developers with a vital task: to design and implement a comprehensive set of test cases using JUnit to validate the functionality of the shipping charge calculation.

Your task is to develop a robust suite of test cases that thoroughly evaluate the shipping charge calculation system under various scenarios, ensuring accurate results and error-free performance.

The **Java-Shipping Charges** test suite aims to ensure the accuracy and reliability of the shipping charge calculation system, providing confidence in its performance and enhancing customer satisfaction.

## 2 Code Structure

### 2.1 Package: Com.Shipping.Service

**Resources**

| Class/Interface | Description | Status |
|---|---|---|
| **ShippingService(class)** | • This class represents a service for calculating shipping costs based on the weight of the package and the distance it needs to be shipped.<br>• It takes the weight and distance as input parameters and calculates the shipping cost according to predefined rates.<br>• The billing logic is structured such that different rates are applied based on different weight ranges of the package. | Already implemented. |

| | | |
|---|---|---|
| | • Don't modify any in this class as this is already implemented. | |

### 2.2 Package: Com.Shipping.Test

**Resources**

| Class/Interface | Description | Status |
|---|---|---|
| **ShippingTest(class)** | • This class contains JUnit test cases to verify the correctness of the calculateShippingCost() method in the ShippingService class.<br>• Each test case should instantiate the ShippingService class with specific input values representing the weight of the package and the distance it needs to be shipped. It then asserts that the calculated shipping cost matches the expected value.<br>• These test cases ensure that the shipping cost calculation implemented in the ShippingService class produces accurate results for different scenarios of package weight and shipping distance.<br>• Make sure the test cases you write achieves 100% code coverage. | To be implemented. |

# 3 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. **This editor Auto Saves the code.**
4. **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.**
5. **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.**
6. **To execute and run test cases:**
   **mvn clean install exec:java -Dexec.mainClass="mainapp.MyApp" -DskipTests=true**
7. **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.**