Functional Testing for E-commerce Web Application

Please note: This is only for the 1st scenario of the tests.

Project Agenda: To perform functional testing on an e-commerce web application using the test automation tool Selenium and TestNG

Description:
As part of this project, you must clone the below e-commerce application:
git clone https://github.com/Simplilearn-Edu/ATE_PEP2_Testing_Using_TestNG.git and paste the cloned folder on the given path to deploy on the Apache 2 server: /var/www/html

This application is based on Java, HTML, and CSS.

The application will be available at the URL: localhost:80/ecomm and then use Selenium, TestNG, and Cucumber framework to perform testing.

In the first part of the project, create a Maven project in the Eclipse IDE and install Selenium and TestNG to perform this project.

Write Selenium and Cucumber scripts to:
To validate and test the Home and Contact pages, as well as alerts
Use different locators to test elements like textboxes, buttons, checkboxes, and radio buttons on the application
Use the TestNG framework to execute the Selenium scripts and generate the TestNG report
In the second part of the project, create a new project with the Cucumber framework
Write a feature file to validate the Home and Checkout pages
Write a step definition file for the features
Validate the test cases by executing the feature file using JUnit

Set up a scenario:
• Clone the repository: https://github.com/SimplilearnEdu/ATE_PEP2_Testing_Using_TestNG.git
• Deploy the code on the Apache 2 server
• The application will be available on localhost:80/ecomm
• Create a simple Maven project In Eclipse IDE
• In the POM.xml file, add the dependencies to install Selenium and TestNG
• To execute the second part of the project, create a Maven project, download the Cucumber plugin, and add JUnit and Cucumber dependencies in the POM.xml file
Tools required: Eclipse IDE, Selenium, Cucumber, TestNG, and Apache 2

Detailed scenario 1:
Create a Base test class and write steps to:
• Create a method and name it OpenBrowser()
• Add steps to open the Chrome browser and maximize the window
• Add steps to open the application URL in the browser
• Add steps to code to delete cookies
• Add Selenium commands in the PageLoadTimeout code
• Create a new class to test elements on the e-commerce application's Home page
• The HomePage class should extend the test base class so that we can import method OpenBrowser()
• Create a test method and name it HomePageTest()
• Write steps to perform mouse hover action on menu option Pages
• Write steps to click on Contact in the dropdown list
• Write steps to handle the Alert box that will appear