Question 1

**BaseTest class:**

import io.github.bonigarcia.wdm.WebDriverManager;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.chrome.ChromeDriver;

import org.testng.annotations.BeforeSuite;

import org.testng.annotations.AfterSuite;

public class BaseTest {

protected WebDriver driver;

@BeforeSuite

public void initializeDriver() {

// Setup the WebDriver using WebDriverManager

WebDriverManager.chromedriver().setup();

driver = new ChromeDriver();

}

@AfterSuite

public void closeDriver() {

if (driver != null) {

driver.quit();

}

}

}

**HomePage class:**

public class HomePage {

WebDriver driver;

public HomePage(WebDriver driver) {

this.driver = driver;

}

By flipkartLogo = By.xpath("//a[@title='Flipkart']");

public boolean isFlipkartLogoDisplayed() {

return driver.findElement(flipkartLogo).isDisplayed();

}

}

**ProductPage class:**

public class ProductPage {

WebDriver driver;

public ProductPage(WebDriver driver) {

this.driver = driver;

}

By searchBox = By.xpath("//input[@title='Search for products, brands and more']");

By firstProduct = By.xpath("//div[@data-id='your\_product\_id']");

By addToCartButton = By.xpath("//button[text()='Add to Cart']");

public void searchProduct(String productName) {

driver.findElement(searchBox).sendKeys(productName);

driver.findElement(searchBox).sendKeys(Keys.ENTER);

}

public void clickFirstProduct() {

driver.findElement(firstProduct).click();

}

public void clickAddToCart() {

driver.findElement(addToCartButton).click();

}

}

**Test class:**

public class ECommerceTest extends BaseTest {

HomePage homePage;

ProductPage productPage;

@BeforeClass

public void setup() {

homePage = new HomePage(driver);

productPage = new ProductPage(driver);

}

@Test

public void testECommerceScenario() {

driver.get("https://www.flipkart.com/");

Assert.assertTrue(homePage.isFlipkartLogoDisplayed());

productPage.searchProduct("Macbook air m2");

productPage.clickFirstProduct();

productPage.clickAddToCart();

}

}

//To verify the successful addition of an item to the cart after clicking the "Add To Cart" button,

**CartPage class**

public class CartPage {

WebDriver driver;

public CartPage(WebDriver driver) {

this.driver = driver;

}

By successMessage = By.xpath("//div[contains(text(), 'Added to cart')]");

public boolean isItemAddedToCart() {

WebDriverWait wait = new WebDriverWait(driver, 10);

return wait.until(ExpectedConditions.visibilityOfElementLocated(successMessage)).isDisplayed();

}

}

**Integration into Test Class:**

public class ECommerceTest extends BaseTest {

HomePage homePage;

ProductPage productPage;

CartPage cartPage;

@BeforeClass

public void setup() {

homePage = new HomePage(driver);

productPage = new ProductPage(driver);

cartPage = new CartPage(driver);

}

@Test

public void testECommerceScenario() {

driver.get("https://www.flipkart.com/");

Assert.assertTrue(homePage.isFlipkartLogoDisplayed());

productPage.searchProduct("Macbook air m2");

productPage.clickFirstProduct();

productPage.clickAddToCart();

// Verifying item is added to the cart

Assert.assertTrue(cartPage.isItemAddedToCart(), "Item is added to the cart.");

}

}

**TestNG XML file:**

<!DOCTYPE suite SYSTEM "<http://testng.org/testng-1.0.dtd>">

<suite name="ECommerceTestSuite">

<test name="ECommerceTests">

<classes>

<class name="com.capstone.ECommerceTest"/>

</classes>

</test>

</suite>

**Pom.xml**

**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0** [**https://maven.apache.org/xsd/maven-4.0.0.xsd**](https://maven.apache.org/xsd/maven-4.0.0.xsd)**"**>  
 <**modelVersion**>4.0.0</**modelVersion**>  
 <**groupId**>ECo < mmerce</**groupId**>  
 <**artifactId**>ECommerce</**artifactId**>  
 <**version**>0.0.1-SNAPSHOT</**version**>  
 <**build**>  
 <**plugins**>  
 <**plugin**>  
 <**groupId**>org.apache.maven.plugins</**groupId**>  
 <**artifactId**>maven-compiler-plugin</**artifactId**>  
 <**configuration**>  
 <**source**>8</**source**>  
 <**target**>8</**target**>  
 </**configuration**>  
 </**plugin**>  
 </**plugins**>  
 </**build**>  
   
 <**properties**>  
 <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  
 </**properties**>  
   
 <**dependencies**>  
 <**dependency**>  
 <**groupId**>org.apache.maven.plugins</**groupId**>  
 <**artifactId**>maven-compiler-plugin</**artifactId**>  
 <**version**>3.3</**version**>  
 <**type**>maven-plugin</**type**>  
 </**dependency**>  
 <**dependency**>  
 <**groupId**>org.apache.maven.plugins</**groupId**>  
 <**artifactId**>maven-surefire-plugin</**artifactId**>  
 <**version**>2.12.4</**version**>  
 <**type**>maven-plugin</**type**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>org.seleniumhq.selenium</**groupId**>  
 <**artifactId**>selenium-java</**artifactId**>  
 <**scope**>system</**scope**>  
 <**version**>3.4.0</**version**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\selenium-java-3.4.0.jar</**systemPath**>  
 </**dependency**>  
   
   
 <**dependency**>  
 <**groupId**>junit</**groupId**>  
 <**artifactId**>junit</**artifactId**>  
 <**scope**>system</**scope**>  
 <**version**>4.8.2</**version**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\junit-4.8.2.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>org.hamcrest</**groupId**>  
 <**artifactId**>hamcrest-all</**artifactId**>  
 <**scope**>system</**scope**>  
 <**version**>1.3</**version**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\hamcrest-all-1.3.jar</**systemPath**>  
 </**dependency**>  
   
   
  
   
  
 <**dependency**>  
 <**groupId**>org.apache.commons</**groupId**>  
 <**artifactId**>commons-configuration2</**artifactId**>  
 <**scope**>system</**scope**>  
 <**version**>2.0</**version**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\commons-codec-1.9.jar</**systemPath**>  
 </**dependency**>  
   
   
 <**dependency**>  
 <**groupId**>org.testng</**groupId**>  
 <**artifactId**>testng</**artifactId**>  
 <**scope**>system</**scope**>  
 <**version**>6.9.4</**version**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\testng-6.9.4.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>org.apache.httpcomponents</**groupId**>  
 <**artifactId**>httpclient</**artifactId**>  
 <**scope**>system</**scope**>  
 <**version**>4.5.13</**version**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\httpclient-4.5.13.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>org.apache.httpcomponents</**groupId**>  
 <**artifactId**>httpcore</**artifactId**>  
 <**scope**>system</**scope**>  
 <**version**>4.4.13</**version**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\httpcore-4.4.13.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>org.projectlombok</**groupId**>  
 <**artifactId**>lombok</**artifactId**>  
 <**scope**>system</**scope**>  
 <**version**>1.18.8</**version**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\lombok-1.18.8.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>com.microsoft.sqlserver</**groupId**>  
 <**artifactId**>mssql-jdbc</**artifactId**>  
 <**scope**>system</**scope**>  
 <**version**>6.2.2.jre8</**version**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\mssql-jdbc-6.2.2.jre8.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>com.oracle</**groupId**>  
 <**artifactId**>ojdbc</**artifactId**>  
 <**version**>6</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\ojdbc6.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>javax.xml.soap</**groupId**>  
 <**artifactId**>javax.xml.soap-api</**artifactId**>  
 <**version**>1.4.0</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\javax.xml.soap-api-1.4.0.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>com.aventstack</**groupId**>  
 <**artifactId**>extentreports</**artifactId**>  
 <**version**>4.0.9</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\extentreports-4.0.9.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>com.relevantcodes</**groupId**>  
 <**artifactId**>extentreports</**artifactId**>  
 <**version**>2.41.2</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\extentreports-2.41.2.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>org.freemarker</**groupId**>  
 <**artifactId**>freemarker</**artifactId**>  
 <**version**>2.3.23</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\freemarker-2.3.23.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>org.apache.poi</**groupId**>  
 <**artifactId**>poi</**artifactId**>  
 <**version**>4.1.0</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\poi-4.1.0.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>net.sf.ezmorph</**groupId**>  
 <**artifactId**>ezmorph</**artifactId**>  
 <**version**>1.0.6</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\ezmorph-1.0.6.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>net.sf.json-lib</**groupId**>  
 <**artifactId**>json-lib</**artifactId**>  
 <**version**>2.4</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\json-lib-2.4-jdk15.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>org.json</**groupId**>  
 <**artifactId**>json</**artifactId**>  
 <**version**>20171018</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\json-20171018.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>org.apache.commons</**groupId**>  
 <**artifactId**>commons-lang3</**artifactId**>  
 <**version**>2.4</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\commons-lang-2.4.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>commons-collections</**groupId**>  
 <**artifactId**>commons-collections</**artifactId**>  
 <**version**>3.2.1</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\commons-collections-3.2.1.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>commons-beanutils</**groupId**>  
 <**artifactId**>commons-beanutils</**artifactId**>  
 <**version**>1.9.4</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Automation Suite\repos\additionalAutomationJars\commons-beanutils-1.9.4.jar</**systemPath**>  
 </**dependency**>  
   
 <**dependency**>  
 <**groupId**>xom</**groupId**>  
 <**artifactId**>xom</**artifactId**>  
 <**version**>1.0b3</**version**>  
 <**scope**>system</**scope**>  
 <**systemPath**>C:\Auto   
  
 </**dependencies**>  
   
</**project**>

Question 2

Create a folder named **ecommerce\_test** and within that folder, create a file named **test\_ecommerce.py**. This file will contain the test scenarios.

**test\_ecommerce.py**.

import pytest

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.support import expected\_conditions as EC

from selenium.webdriver.support.ui import WebDriverWait

@pytest.fixture

def browser():

driver = webdriver.Chrome() # You can use any other browser webdriver here

yield driver

driver.quit()

def test\_add\_macbook\_to\_cart(browser):

# 1. Launch the URL and verify 'Flipkart' is present on the left side of the top.

browser.get("https://www.flipkart.com/")

flipkart\_logo = WebDriverWait(browser, 10).until(

EC.visibility\_of\_element\_located((By.CSS\_SELECTOR, "a.\_2xm1JU"))

)

assert flipkart\_logo.text == "Flipkart"

# 2. Enter "Macbook air m2" in the search box and search

search\_box = browser.find\_element(By.CSS\_SELECTOR, "input.\_3704LK")

search\_box.send\_keys("Macbook air m2")

browser.find\_element(By.CSS\_SELECTOR, "buttonL0Z3Pu").click()

# 3. Click on the first displayed item

first\_item = WebDriverWait(browser, 10).until(

EC.visibility\_of\_element\_located((By.CSS\_SELECTOR, "a[data-testid='product-card']"))

)

first\_item.click()

# 4. Click on Add To Cart button

add\_to\_cart\_button = WebDriverWait(browser, 10).until(

EC.visibility\_of\_element\_located((By.CSS\_SELECTOR, "button.\_2KpZ6l.\_2U9uOA.\_3v1-ww"))

)

add\_to\_cart\_button.click()

# 5. Verify the item has been added to the cart successfully

cart\_element = WebDriverWait(browser, 10).until(

EC.visibility\_of\_element\_located((By.CSS\_SELECTOR, "span[title='My Cart']"))

)

assert cart\_element.text == "My Cart"

….........................

**base\_page.py**

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.common.by import By

class BasePage:

def \_\_init\_\_(self, driver):

self.driver = driver

def wait\_for\_element(self, locator):

return WebDriverWait(self.driver, 10).until(

lambda driver: self.driver.find\_element(\*locator)

)

**home\_page.py**

from pages.base\_page import BasePage

from selenium.webdriver.common.keys import Keys

from selenium.webdriver.common.by import By

class HomePage(BasePage):

SEARCH\_BOX = (By.NAME, 'q')

FIRST\_DISPLAYED\_ITEM = (By. XPATH, '//div[@data-id='your\_product\_id'] ')

ADD\_TO\_CART\_BUTTON = (By.XPATH, '//button[text()='Add to Cart'] ')

def search\_for\_product(self, product\_name):

search\_box = self.wait\_for\_element(self.SEARCH\_BOX)

search\_box.send\_keys(product\_name)

search\_box.send\_keys(Keys.ENTER)

def click\_first\_displayed\_item(self):

first\_item = self.wait\_for\_element(self.FIRST\_DISPLAYED\_ITEM)

first\_item.click()

def click\_add\_to\_cart(self):

add\_to\_cart = self.wait\_for\_element(self.ADD\_TO\_CART\_BUTTON)

add\_to\_cart.click()

**test\_ecommerce.py**

import pytest

from selenium import webdriver

from pages.home\_page import HomePage

@pytest.fixture

def browser():

driver = webdriver.Chrome()

yield driver

driver.quit()

def test\_ecommerce\_flow(browser):

home\_page = HomePage(browser)

browser.get('https://www.flipkart.com/')

home\_page.search\_for\_product("Macbook air m2")

home\_page.click\_first\_displayed\_item()

home\_page.click\_add\_to\_cart()

Question 3



