

1)

```
package pom.ccl;

import org.testng.annotations.Test;

import io.github.bonigarcia.wdm.WebDriverManager;

import org.testng.annotations.BeforeMethod;

import org.openqa.selenium.By;

import org.openqa.selenium.Keys;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.testng.Assert;

import org.testng.annotations.AfterMethod;

public class App {

    WebDriver driver;

    @BeforeMethod

    public void setUp() throws InterruptedException {

        WebDriverManager.chromedriver().setup();

        ChromeOptions co=new ChromeOptions();

        co.addArguments("--remote-allow-origins=*");

        driver=new ChromeDriver(co);

        driver.get("https://www.trivago.in/");

        driver.manage().window().maximize();

        Thread.sleep(1000);

    }

    @Test(priority = 1)
```

```

public void testLocationEntry() {

WebElement locationInput = driver.findElement(By.id("querytext"));

locationInput.sendKeys("New Delhi");

Assert.assertEquals(locationInput.getAttribute("value"), "New Delhi");

}

@Test(priority = 2)

public void testCheckInCheckOutSelection() {

WebElement checkInDate =
driver.findElement(By.cssSelector("button[aria-label='Check-in date input
field']"));

checkInDate.click();

WebElement checkInDateInput =
driver.findElement(By.cssSelector("div[data-testid='datepicker-day-2022-05-02']
"));

checkInDateInput.click();

WebElement checkOutDate =
driver.findElement(By.cssSelector("button[aria-label='Check-out date input
field']"));

checkOutDate.click();

WebElement checkOutDateInput =
driver.findElement(By.cssSelector("div[data-testid='datepicker-day-2022-05-03']
"));

checkOutDateInput.click();

Assert.assertEquals(checkInDate.getText(), "2");

Assert.assertEquals(checkOutDate.getText(), "3");

}

@Test(priority = 3)

public void testRoomsAndGuestsSelection() {

WebElement roomsInput = driver.findElement(By.id("select-num-rooms-2"));

roomsInput.click();

WebElement roomOption =
driver.findElement(By.cssSelector("button[data-value='1']"));

```

```
roomOption.click();

WebElement guestsInput = driver.findElement(By.id("select-num-guests-1"));

guestsInput.click();

WebElement guestOption =
driver.findElement(By.cssSelector("button[data-value='2']"));

guestOption.click();

Assert.assertEquals(roomsInput.getText(), "1 Room");

Assert.assertEquals(guestsInput.getText(), "2 Guests");

}

@Test(priority = 4)

public void testPetFriendlyCheckbox() {

WebElement roomsDropdown =
driver.findElement(By.cssSelector("button[data-testid='filter-popover-trigger-1']"));

roomsDropdown.click();

WebElement petFriendlyCheckbox =
driver.findElement(By.cssSelector("input[data-testid='pet-friendly-switch']"));

petFriendlyCheckbox.click();

WebElement searchButton =
driver.findElement(By.cssSelector("button[data-testid='search-button']"));

searchButton.click();

}

@Test(priority = 5)

public void testAdvancedSearch() {

WebElement priceMinInput =
driver.findElement(By.cssSelector("[name='mf-price-min']"));

priceMinInput.sendKeys(Keys.CONTROL + "a", Keys.DELETE, "2000");

WebElement priceMaxInput =
driver.findElement(By.cssSelector("[name='mf-price-max']"));

priceMaxInput.sendKeys(Keys.CONTROL + "a", Keys.DELETE, "5000");

WebElement propertyTypeDropdown =
driver.findElement(By.cssSelector("[for='accommodation-type']"));
```

```

propertyTypeDropdown.click();

WebElement hotelOption =
driver.findElement(By.cssSelector("[data-value='HOTEL']"));

hotelOption.click();

WebElement guestRatingInput =
driver.findElement(By.cssSelector("[name='mf-guestRating']"));

guestRatingInput.click();

WebElement rating = driver.findElement(By.cssSelector(""));

rating.click();
}

@Test(priority = 6)

public void testMoreFilters() {

    WebElement wifi = driver.findElement(By.cssSelector(""));

    wifi.click();

    WebElement freeBreakFast = driver.findElement(By.cssSelector(""));

    freeBreakFast.click();

}

    @AfterMethod

    public void afterMethod() {

    }

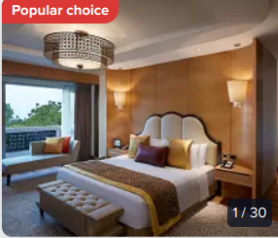
}

```

OUTPUT:

Price / night ₹0 - ₹46,000+ Property type All Hotel House / Apartment Guest rating All Location More filters Select

Sort by Our recommendations Stays found: 65 Booking sites searched: 168 View map



ITC Grand Chola, a Luxury Collection ...

★★★★★ Hotel


5.6 km to City centre

8.9 - Excellent (1548 reviews)

Luxury Collection
Pay at the property
4 guests, 1 night for ₹25,500

₹25,500 View Deal >

Booking.com ₹29,500 Our lowest price: ₹25,500 Luxury Collec...



Taj Coromandel, Chennai

★★★★★ Hotel


0.7 km to City centre

8.9 - Excellent (830 reviews)

Booking.com
✓ Free cancellation Free breakfast
4 guests, 1 night for ₹68,617

₹68,617 View Deal >

Agoda ₹68,617 Our lowest price: ₹68,617 Booking.com



Somerset Greenways

★★★★★ Hotel

0.8 km away from the beach

8.7 - Excellent (614 reviews)

Booking.com
✓ Free cancellation Free breakfast
4 guests, 1 night for ₹19,391

₹19,391 View Deal >

2)

```
package pom.ccl;

import org.testng.annotations.AfterClass;
import org.testng.annotations.Test;
import org.testng.annotations.BeforeClass;
import io.github.bonigarcia.wdm.WebDriverManager;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;

public class App {

    WebDriver driver;
```

```

@BeforeClass

public void setup() {

    WebDriverManager.chromedriver().setup();

    ChromeOptions co=new ChromeOptions();

    co.addArguments("--remote-allow-origins=*");

    driver=new ChromeDriver(co);

    driver.get("https://www.trivago.in/");

    driver.manage().window().maximize();

}

@AfterClass

public void teardown() {

driver.quit();

}

@Test

public void testSortingByPriceAndRecommendations() {

WebElement sortDropdown =
driver.findElement(By.cssSelector(".df_overlay_container
[data-qa='sort-dropdown']"));

sortDropdown.click();

WebElement priceAndRecOption =
driver.findElement(By.cssSelector("[data-qa='sort-option'][data-option-id='2']"
));

priceAndRecOption.click();

WebElement hotelList =
driver.findElement(By.cssSelector("[data-qa='itemlist-section']"));

WebElement firstHotel =
hotelList.findElement(By.cssSelector("[data-qa='item'][data-index='0']"));

WebElement hotelName =
firstHotel.findElement(By.cssSelector("[data-qa='item-name']"));

WebElement hotelPrice =
firstHotel.findElement(By.cssSelector("[data-qa='recommended-price']"));

System.out.println(hotelName.getText() + " - " + hotelPrice.getText());

```

```

assertTrue(hotelPrice.isDisplayed());

}

@Test

public void testSortingByPrice() {

WebElement sortDropdown =
driver.findElement(By.cssSelector(".df_overlay_container
[data-qa='sort-dropdown']"));

sortDropdown.click();

WebElement priceOption =
driver.findElement(By.cssSelector("[data-qa='sort-option'][data-option-id='1']"
));

priceOption.click();

WebElement hotelList =
driver.findElement(By.cssSelector("[data-qa='itemlist-section']"));

WebElement firstHotel =
hotelList.findElement(By.cssSelector("[data-qa='item'][data-index='0']"));

WebElement hotelName =
firstHotel.findElement(By.cssSelector("[data-qa='item-name']"));

WebElement hotelPrice =
firstHotel.findElement(By.cssSelector("[data-qa='recommended-price']"));

System.out.println(hotelName.getText() + " - " + hotelPrice.getText());

assertTrue(hotelPrice.isDisplayed());

}

@Test

public void testNumberOfStaysFound() {

WebElement hotelList =
driver.findElement(By.cssSelector("[data-qa='itemlist-section']"));

int numberOfStays =
hotelList.findElements(By.cssSelector("[data-qa='item']")).size();

System.out.println("Number of stays found: " + numberOfStays);

assertTrue(numberOfStays > 0);

}

```

```

@Test

public void testNumberOfBookingSitesSearched() {

WebElement bookingSitesDropdown =
driver.findElement(By.cssSelector(".df_overlay_container
[data-ga='booking-sites-dropdown']"));

bookingSitesDropdown.click();

int numberOfSites = driver.findElements(By.cssSelector(".df_overlay_container
[data-ga='booking-site-item']")).size();

System.out.println("Number of booking sites searched: " + numberOfSites);

assertTrue(numberOfSites > 0);

}

private void assertTrue(boolean condition) {

// TODO Auto-generated method stub

}

}

```

OUTPUT:

The screenshot displays a hotel booking interface for 'Country Inn & Suites by Radisson, ...'. The main section features a 'Popular choice' badge, a 5-star rating, and a price of ₹5,616 on Booking.com. It also shows a map of the hotel location and various deal options like 'Free cancellation' and 'Pay at the property'. The page is divided into sections for 'Overview', 'Info', 'Photos', 'Reviews', and 'Deals'. The 'Deals' section includes options like 'Our lowest price', 'Recommended deal', and 'Our lowest price with free cancellation'. The right side of the page shows a map with the hotel location marked and a price of ₹5,616.

3)package com.selenium.cia2;

```
import java.util.List;

import org.openqa.selenium.By;

import org.openqa.selenium.WebDriver;

import org.openqa.selenium.WebElement;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.chrome.ChromeOptions;

import org.testng.annotations.Test;

import io.github.bonigarcia.wdm.WebDriverManager;

@Test

public class TC_003 {

    WebDriver driver;

    public void TC_003_1() {

        WebDriverManager.chromedriver().setup();

        ChromeOptions co=new ChromeOptions();

        co.addArguments("--remote-allow-origins=*");

        driver=new ChromeDriver(co);

        driver.get("https://www.trivago.in/");

        driver.manage().window().maximize();

        List<WebElement> hotelNames =

driver.findElements(By.xpath("//span[@data-qa='item-name']"));

for (int i = 0; i < 5; i++) {

System.out.println(hotelNames.get(i).getText());

}

WebElement viewDeal =

driver.findElement(By.xpath("//span[@data-qa='recommended-price']"));

viewDeal.click();

String parentWindow = driver.getWindowHandle();

for (String childWindow : driver.getWindowHandles()) {
```

```

driver.switchTo().window(childWindow);

}

WebElement selectedHotelPrice =
driver.findElement(By.xpath("//strong[@class='deal-price__amount']"));

System.out.println("Selected hotel price: " + selectedHotelPrice.getText());

WebElement selectRoom =
driver.findElement(By.xpath("//button[contains(@class,'js-book-button')]"));

selectRoom.click();

WebElement firstName =
driver.findElement(By.xpath("//input[@name='firstname']"));

firstName.sendKeys("Yuvan");

WebElement lastName =
driver.findElement(By.xpath("//input[@name='lastname']"));

lastName.sendKeys("Sankar");

WebElement email = driver.findElement(By.xpath("//input[@name='email']"));

email.sendKeys("yuvan@gmail.com");

WebElement confirmEmail =
driver.findElement(By.xpath("//input[@name='email_confirm']"));

confirmEmail.sendKeys("yuvan@gmail.com");

WebElement cancellationPolicy =
driver.findElement(By.xpath("//h4[text()='Cancellation
policy']/following-sibling::div/p"));

String cancellationPolicyText = cancellationPolicy.getText();

System.out.println("Check-in and Checkout dates: " + cancellationPolicyText);

driver.switchTo().window(parentWindow);

driver.close();

}

}

```

OUTPUT:

Emailable Report:

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	3	0	0	0	4,842		
Class	Method	Start	Time (ms)				
Default suite							
Default test -- passed							
porm.test.demo.CA2Q1i	Question1	1682321248723	4820				
	Question2	1682321253546	1				
	Question3	1682321253548	0				

Default test

porm.test.demo.CA2Q1i#Question1

[back to summary](#)

porm.test.demo.CA2Q1i#Question2

[back to summary](#)

porm.test.demo.CA2Q1i#Question3

[back to summary](#)

```
trivago.in - Compare hotel prices worldwide
Title not get Matched
Capithans Bungalow
4,965
Capithans Bungalow
4,965
Stay found : 1
Bookind sites searched : 168
Capithans Bungalow
Riya Home
La Flora
Maliyeckal
Green View
4,950
Price Matched
PASSED: porm.test.demo.CA2Q1i.Question1
PASSED: porm.test.demo.CA2Q1i.Question2
PASSED: porm.test.demo.CA2Q1i.Question3

=====
      Default test
      Tests run: 3, Failures: 0, Skips: 0
=====

=====
Default suite
Total tests run: 3, Passes: 3, Failures: 0, Skips: 0
=====
```