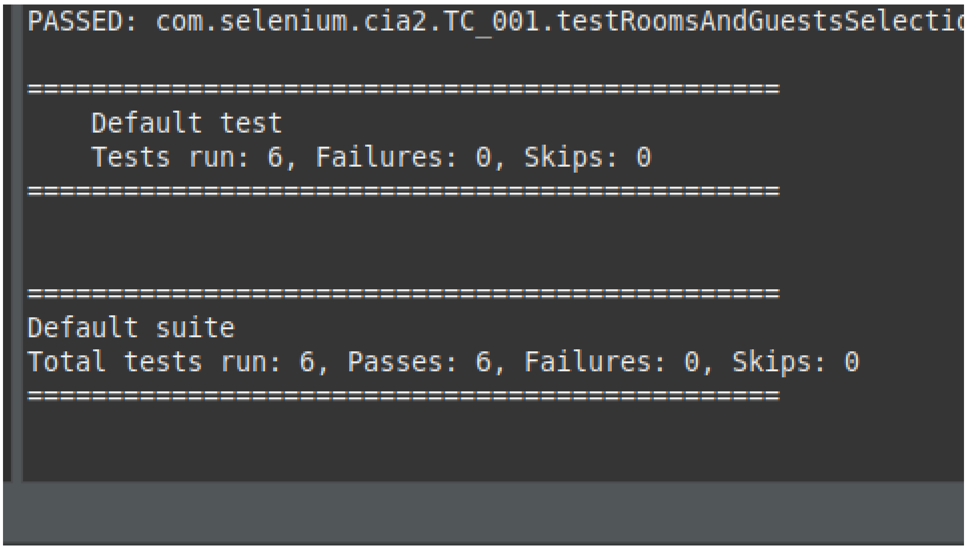
CIA 2

727721EUIT171
THARUN DEEKSITH K

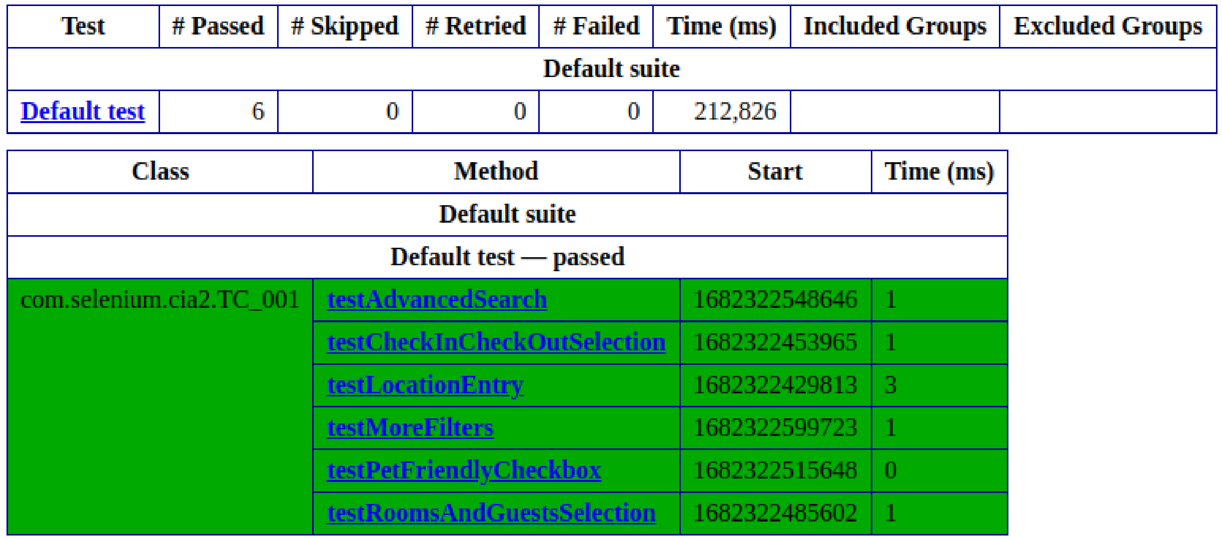
TC_001

```
        package com.selenium.cia2;

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.openqa.selenium.support.ui.Select;
import org.testng.Assert;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import io.github.bonigarcia.wdm.WebDriverManager;

public class TC_001 {
        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();
  }

  @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.xpath("//*[@id=\"__next\"]/div[1]/main/div[2]/div[2]/div/div/div/div[3]/div/butto
n"));
        Select dropdown = new Select(rating);
        dropdown.selectByIndex(1);
    }
    @Test(priority = 6)
    public void testMoreFilters() {
             WebElement wifi =
driver.findElement(By.xpath("//*[@id=\"__next\"]/div[1]/main/div[2]/div[2]/div/div/div/div[5]/div/button"));
             Select dropdown = new Select(wifi);
        dropdown.selectByIndex(1);
        dropdown.selectByIndex(5);
        dropdown.selectByIndex(6);

    }
               @AfterMethod
               public void afterMethod() {
                            driver.close();
               }

}
```



```
PASSED: com.selenium.cia2.TC_001.testRoomsAndGuestsSelecti

===============================================
    Default test
    Tests run: 6, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 6, Passes: 6, Failures: 0, Skips: 0
===============================================
```

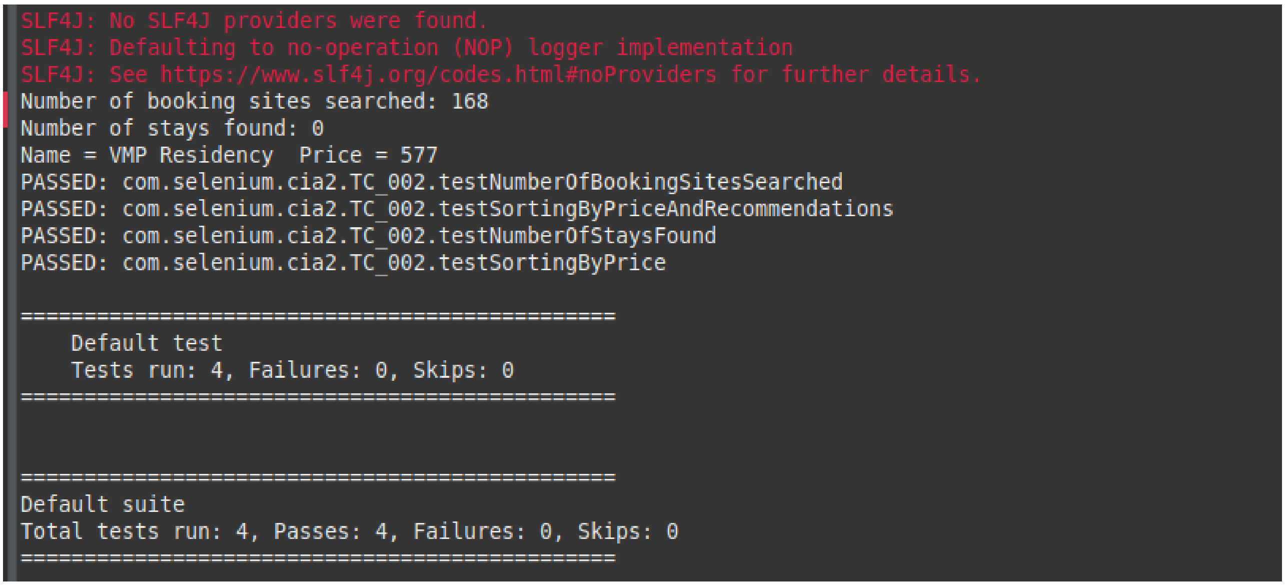


Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	6	0	0	0	212,826		

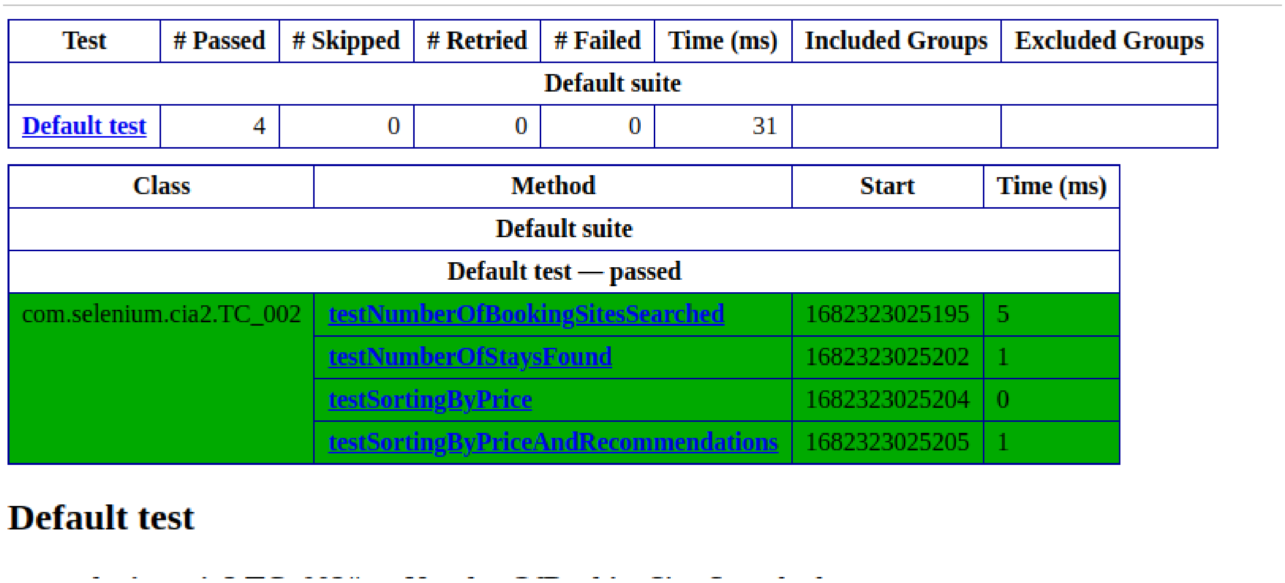
Class	Method	Start	Time (ms)
Default suite			
Default test — passed			
com.selenium.cia2.TC_001	testAdvancedSearch	1682322548646	1
	testCheckInCheckOutSelection	1682322453965	1
	testLocationEntry	1682322429813	3
	testMoreFilters	1682322599723	1
	testPetFriendlyCheckbox	1682322515648	0
	testRoomsAndGuestsSelection	1682322485602	1

TC_002

```
package com.selenium.cia2;

import org.testng.annotations.Test;

import io.github.bonigarcia.wdm.WebDriverManager;


import static org.junit.Assert.assertTrue;

import org.junit.AfterClass;
import org.junit.BeforeClass;
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 TC_002 {
	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-qa='booking-sites-dropdown']"));
        bookingSitesDropdown.click();
        int numberOfSites = driver.findElements(By.cssSelector(".df_overlay_container
[data-qa='booking-site-item']")).size();
        System.out.println("Number of booking sites searched: " + numberOfSites);
        assertTrue(numberOfSites > 0);
    }
}
```



```
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
Number of booking sites searched: 168
Number of stays found: 0
Name = VMP Residency  Price = 577
PASSED: com.selenium.cia2.TC_002.testNumberOfBookingSitesSearched
PASSED: com.selenium.cia2.TC_002.testSortingByPriceAndRecommendations
PASSED: com.selenium.cia2.TC_002.testNumberOfStaysFound
PASSED: com.selenium.cia2.TC_002.testSortingByPrice

===============================================
    Default test
    Tests run: 4, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 4, Passes: 4, Failures: 0, Skips: 0
===============================================
```



Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	4	0	0	0	31		

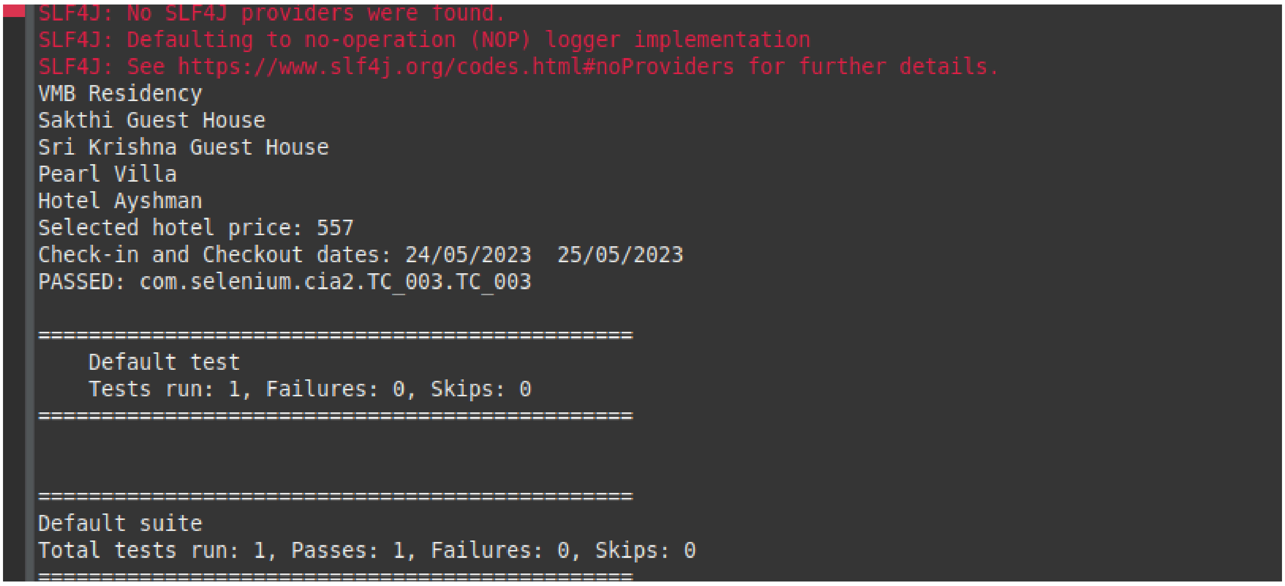
Class	Method	Start	Time (ms)
Default suite			
Default test — passed			
com.selenium.cia2.TC_002	testNumberOfBookingSitesSearched	1682323025195	5
	testNumberOfStaysFound	1682323025202	1
	testSortingByPrice	1682323025204	0
	testSortingByPriceAndRecommendations	1682323025205	1

Default test

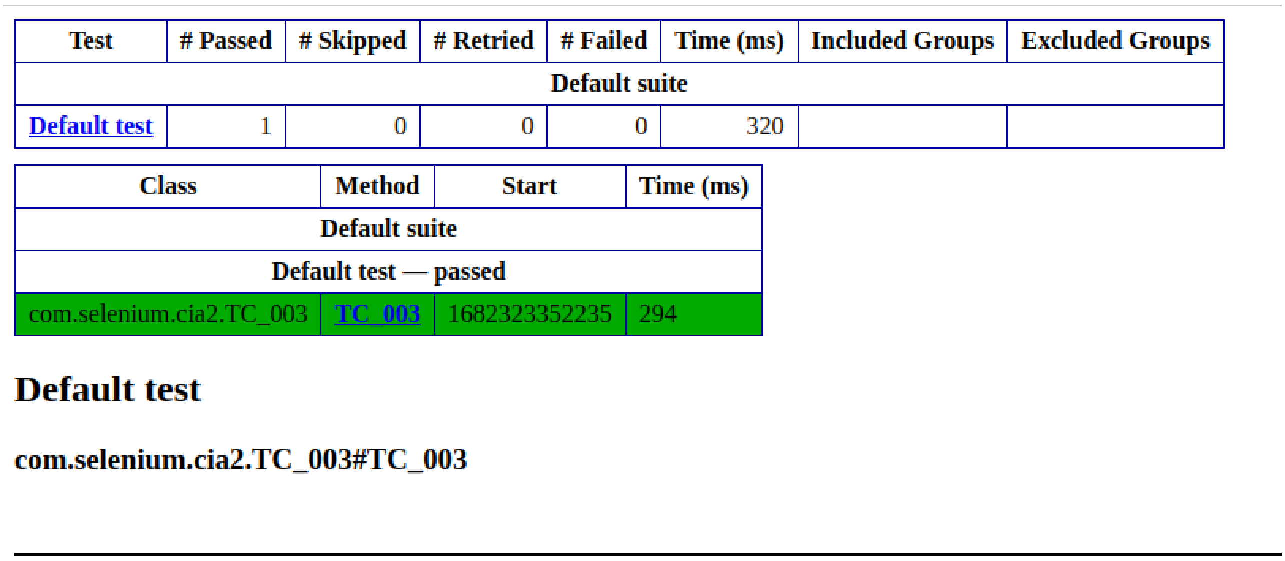
TC_003

```
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() {
             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();
  }
}
```



```
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
VMB Residency
Sakthi Guest House
Sri Krishna Guest House
Pearl Villa
Hotel Ayshman
Selected hotel price: 557
Check-in and Checkout dates: 24/05/2023  25/05/2023
PASSED: com.selenium.cia2.TC_003.TC_003

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================


===============================================
Default suite
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
===============================================
```



Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	1	0	0	0	320		

Class	Method	Start	Time (ms)
Default suite			
Default test — passed			
com.selenium.cia2.TC_003	TC_003	1682323352235	294

Default test

com.selenium.cia2.TC_003#TC_003