**HACKATHON PROJECT**

**PROJECT TEAM NAME: The Debuggers**

**PROJECT: CALCULATE TRIP COST**

**PROJECT MEMBERS**

**Mauli Baraskar : 2090965**

**Shivani Ambavane : 2090826**

**Ujjwal Patel : 2091262**

**Mohammad Shakeer : 2090773**

**Shreya LNU : 2088417**

**Shiva Chinta : 2091475**

**PROJECT DESCRIPTION:**

1. Display Hotel name, total amount and charges per night for 3 holiday homes for 4 people in Nairobi for 5 days of stay from tomorrow's date.
2. Sort the list with highest traveller rating & elevator/ List access
3. Pick one cruise line & pick a respective cruise ship under Cruises;   
                   a. Retrieve all the languages offered and store in a List; Display the same  
                   b. Display passengers, crew & launched year

**WEBSITE USED FOR AUTOMATION:**

<https://www.tripadvisor.in/>

**SOFTWARE SPECIFICATIONS:**

* Programming language used : JAVA
* IDE used : Eclipse
* Automation framework : Selenium
* Libraries added : Maven dependency Library, TestNG Library
* Browser used : Chrome, MS Edge

|  |  |
| --- | --- |
| **Browser** | **Version** |
| MS Edge | 97.0.1072.55 (64-bit) |
| Chrome | 95.0.4638.69(64-bit) |

**HARDWARE SPECIFICATIONS:**

* Processor - intel core i3 7th Gen
* RAM - 4GB Minimum / 8GB Recommended
* Operating System - Windows 10

**AUTOMATION FEASIBILITY:**

Feasible, Selenium is able to locate all web elements and perform actions on them

**KEY AUTOMATION SCOPE:**

* Handling different browser windows, search option
* Validation of date controls
* Capture warning message
* Extract list items & store in collections
* Navigating back to home page
* Scrolling down in web page
* To capture the Screenshot

**AUTOMATION FEASIBILITY CHECKLIST MODEL:**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **READINESS CHECK** | **YES/NO** |
| 1 | Product stable | yes |
| 2 | Any planned enhancement in future | yes |
| 3 | Bugs affect the major functionality | yes |
| 4 | Are the test and precondition are detailed | yes |
| 5 | Test case and test detail analysis done | yes |

**BASIC REQUIREMENTS OF PROJECT:**

* Initially we need to create a Maven Project.
* In that project we have to create test scenarios and test classes.
* According to this project we need to create methods in that pages.
* We have to use Base Class for driver setup and other utilities.
* Data driven concept is used for the pages to read data and write data.
* These are the basic requirements of this project

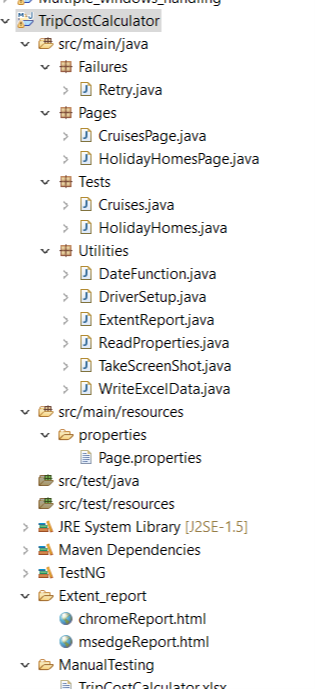
**STEPS TO AUTOMATE:**

* Extract the zip file and save it in folder.
* Import the extracted folder from the eclipse as existing project.
* Right click on the **testng.xml**file and click **run as** then choose **TestNG Suite** it will run on Chrome browser after that on MS Edge browser.
* After the result, check the passes and failed testcase on the testing console.
* And then check the screenshot in folder.
* Finally, the automated concept is completed.

**PROJECT IMPLEMENTATIONS:**

* **Maven :** To simplify Build Process and adding of dependencies.
* **TestNG Library :** For separate execution of each test case in Functional requirement.
* **Page object Model (POM) :** To get the returned objects of each page class during automation.
* **Apache POI :** To read excel file and extract values from it during run time.
* **Config Properties :** To read user properties during test execution.

**PROJECT STRUCTURE:**

****

**src/main/java**

* **Failures**

>Retry - This class retries the method that are failed

* **Pages** - This package contains page wise individual methods stored in **.java** files

> HolidayHomesPage

> CruisesPage

* **Tests**- This package contains two classes with individual methods

>Cruises

> HolidayHomes

* **Utilities**- This package contains all the reusable classes

>DateFunction

> DriverSetup

>ExtentReport

>ReadProperties

>TakeScreenShot

**src/main/resources**

**Properties** -Data’s that are required to automate the tests are stored here

>Page.properties

**src/test/java**

**src/test/resources**

**Extent\_report**

**>**chromeReport.html – The extent report for Chrome browser

>msedgeReport.html – The extent report for MS-Edge browser

**ManualTesting**

>TripCostCalculator.xlsx – Test cases for manual testing

**Readme**

>Readme.docx

**ScreenShots** – The Screenshots of the project are saved here

**pom.xml**

> The dependency needed for the project will be stored here.

**testng.xml -** This file contains ordered way of execution of the full project.

**testng\_regression.xml -** This file contains ordered way of execution of the “regression” group.

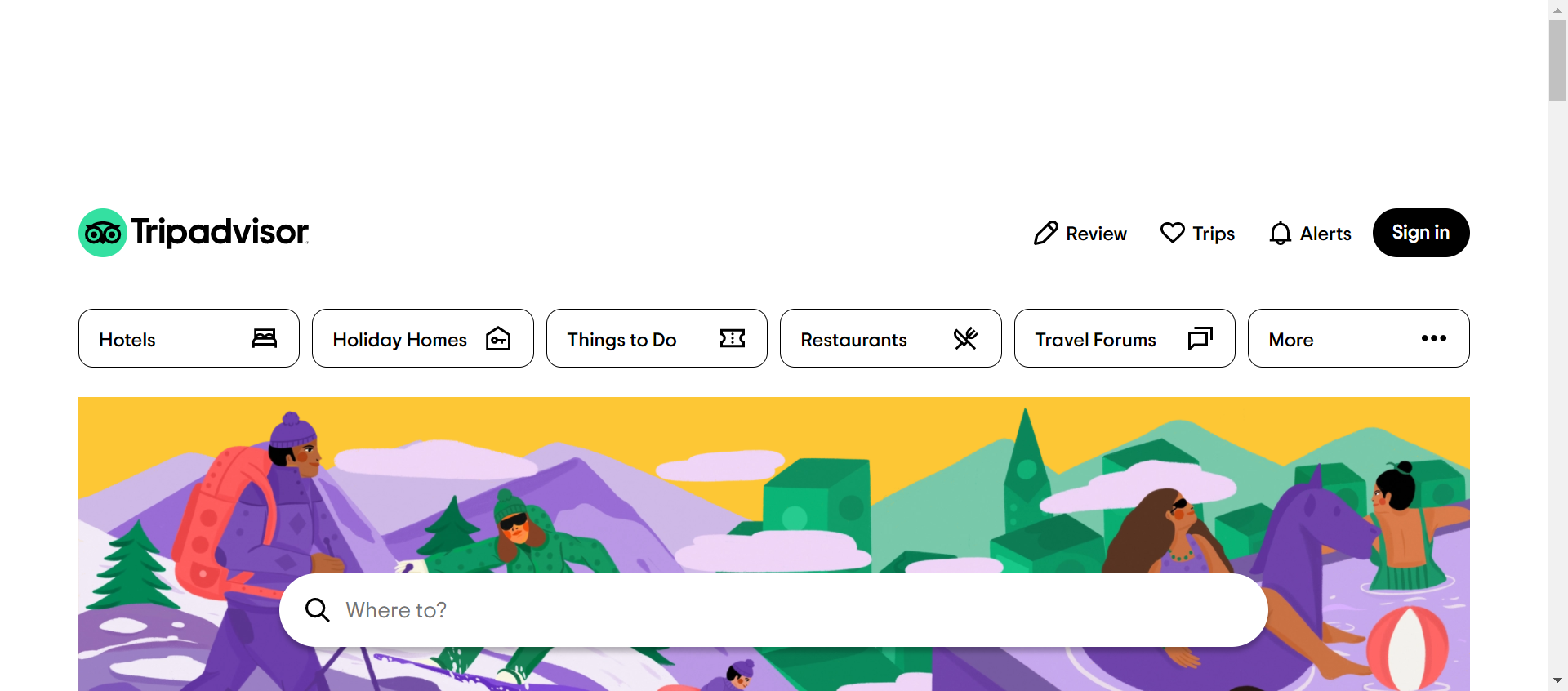
**testng\_smoke.xml -** This file contains ordered way of execution of the “smoke” group.

**EXECUTION:**

Right click on the **testng.xml page** and click **run as** then choose **TestNG Suite**

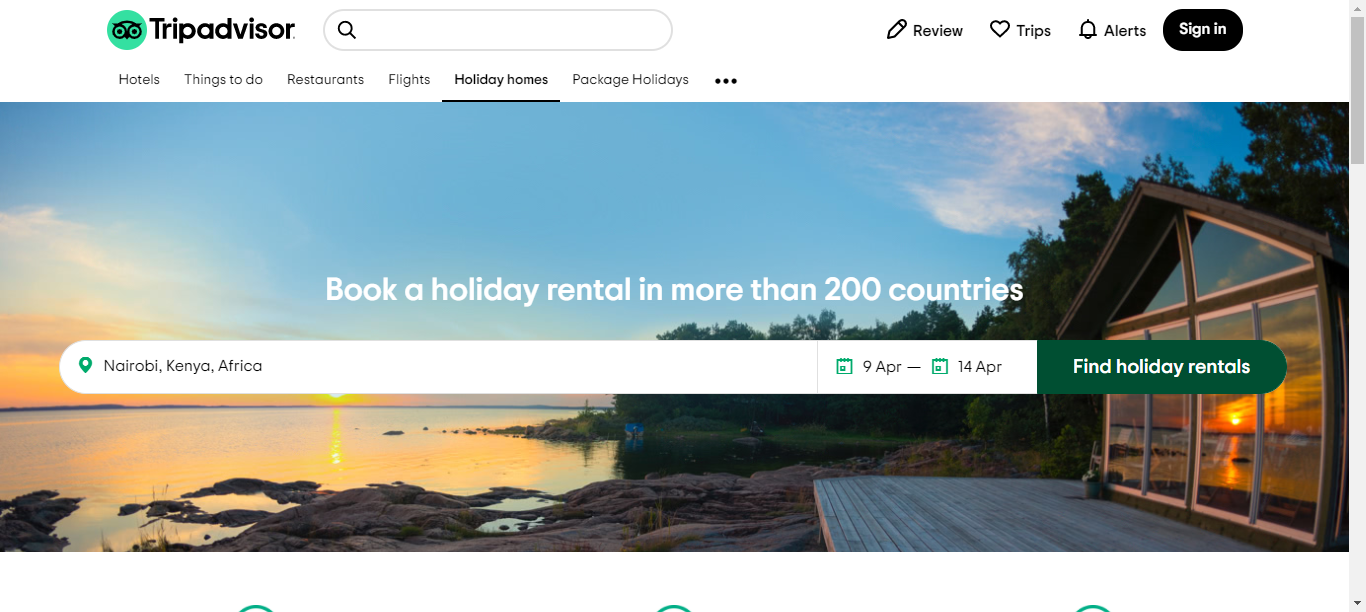
SCREENSHOTS:

**1.TripAdvisor\_HomePage**



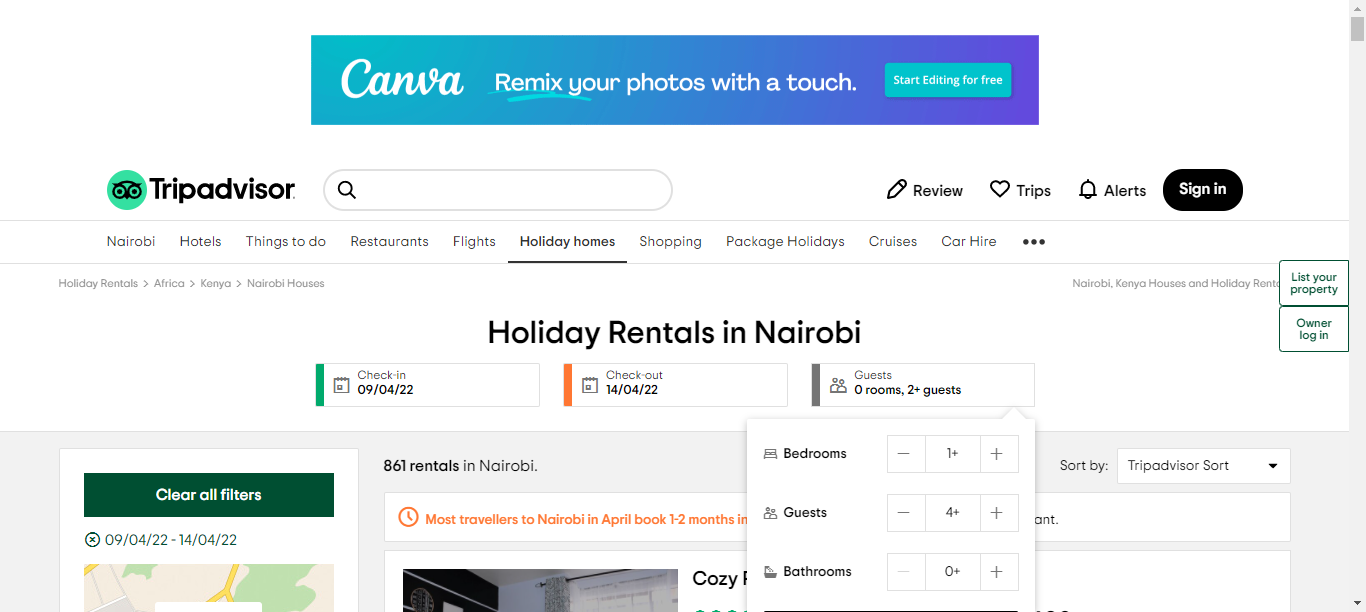
In this Screenshot, we can see the home page of the website **tripadvisor.in,** By clicking Holiday Homes tab, it redirects to Holiday Rentals page.

**2.Rentals\_Page**



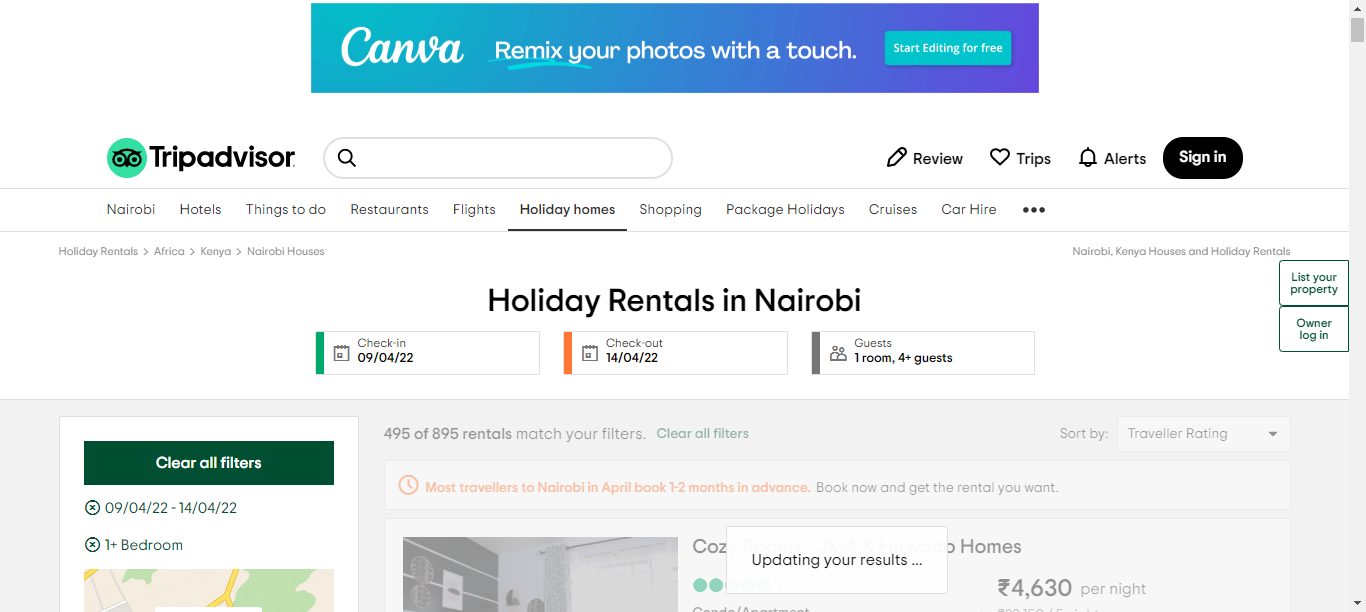
In this screenshot we can see the selected destination and check in & check out date.

**3.Add\_Guests**



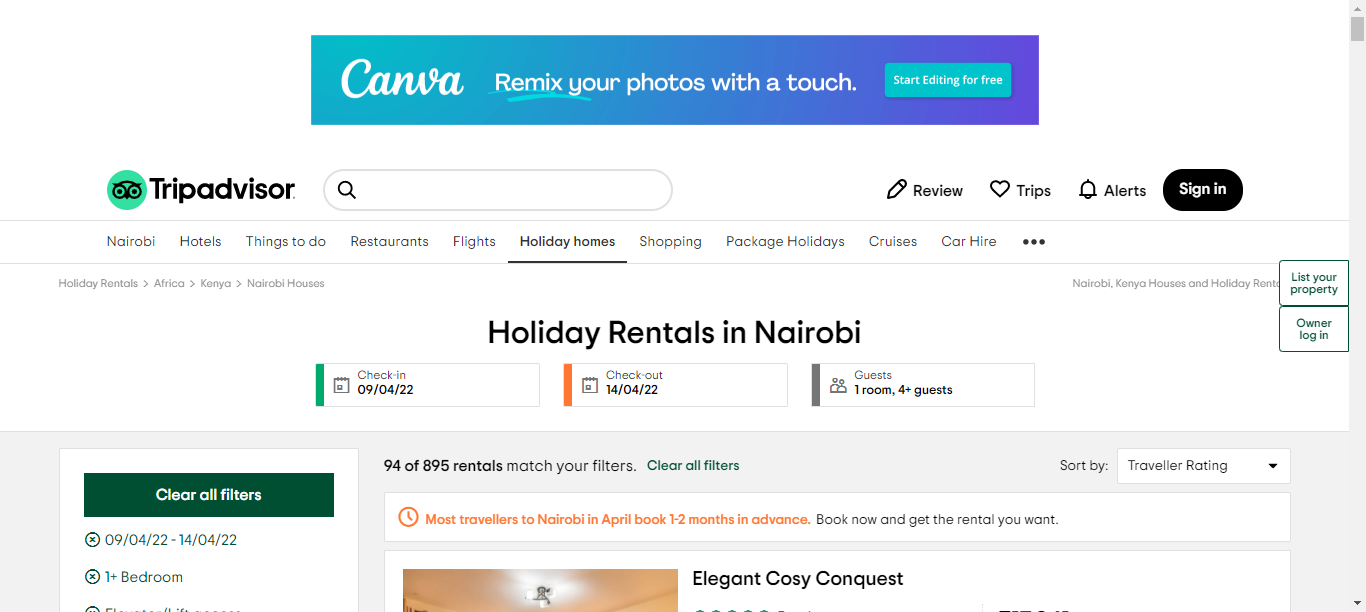
Adding the no of bedrooms and guests in guests tab

**4.Travellers\_Rating**



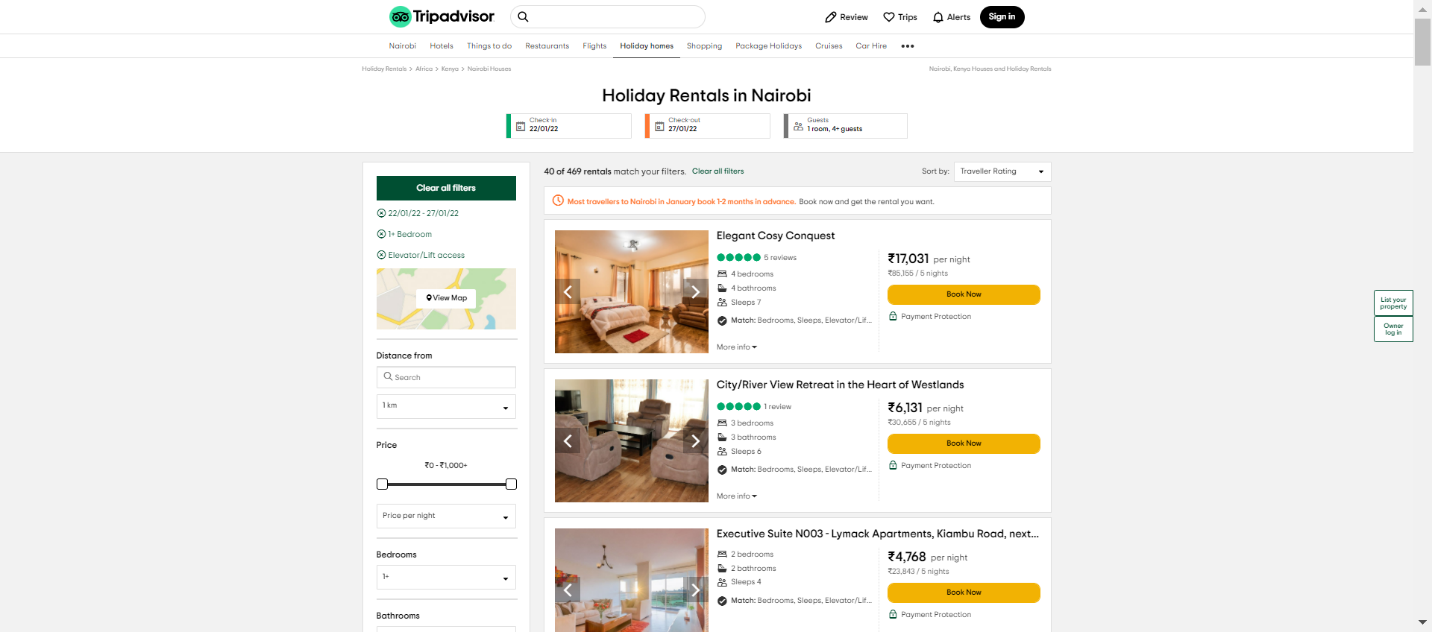
Applying the highest traveller rating filter

**5.Lift\_Access**



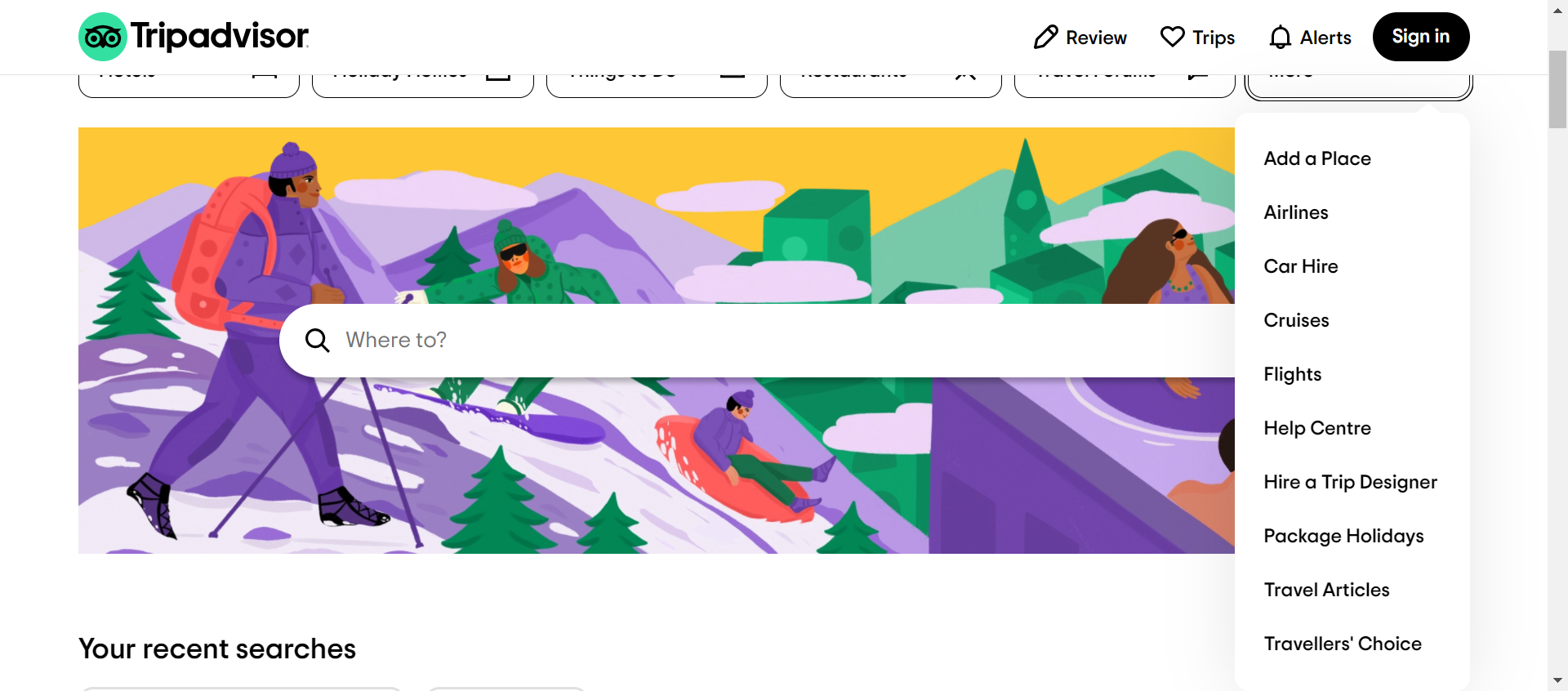
Selecting elevator and lift access in Amenities

**6.Result\_Page**



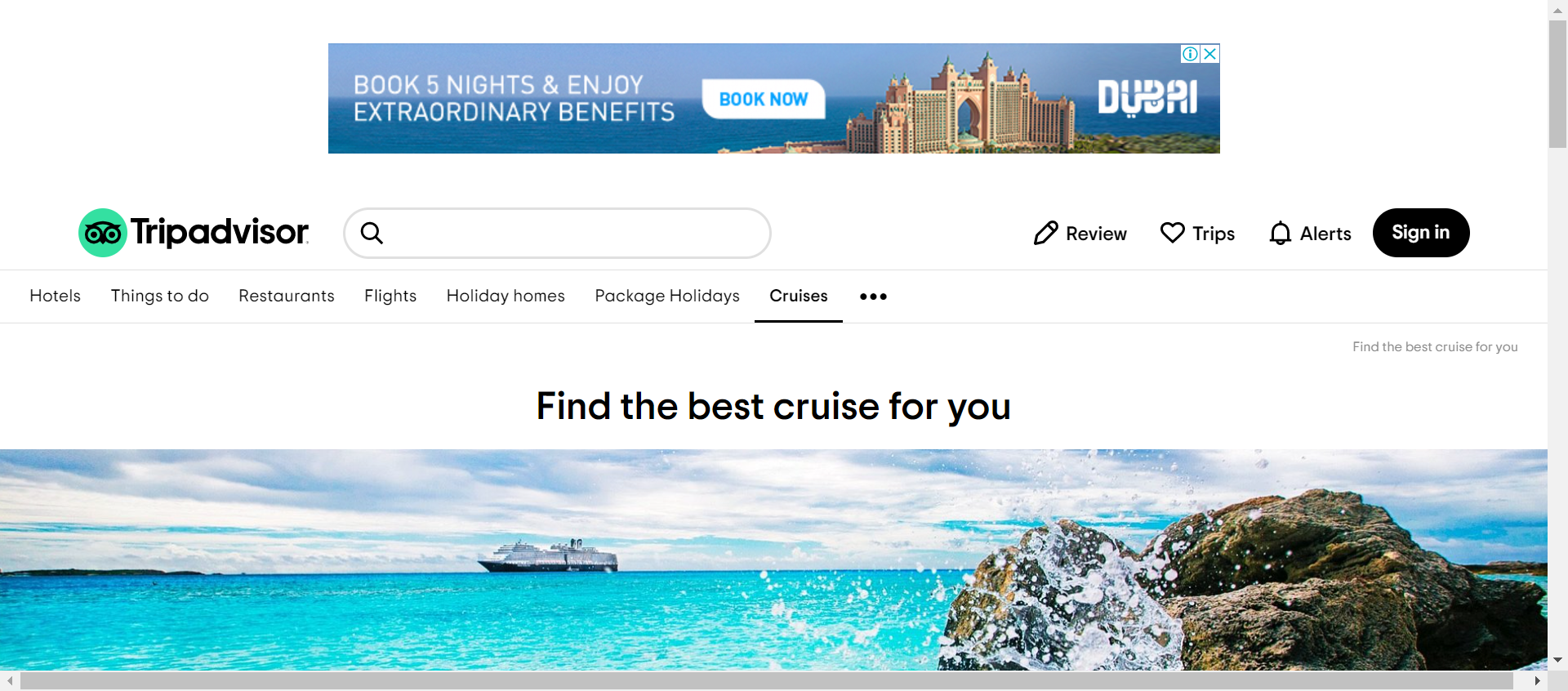
By applying all the filters, we got 40 rentals as a result

**7.Cruises\_Tab**



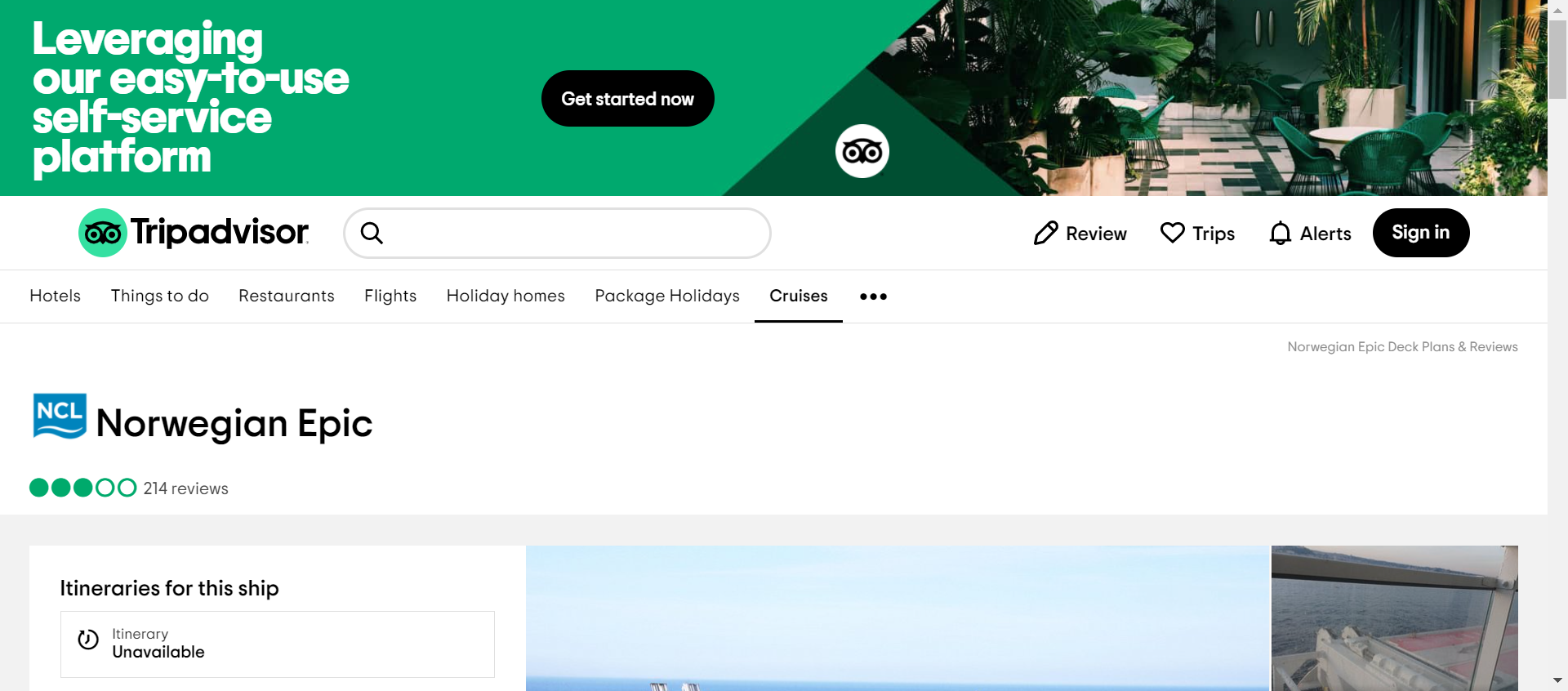
Selecting the “cruises” from more tab after returning to homepage

**8.Select\_Cruise**



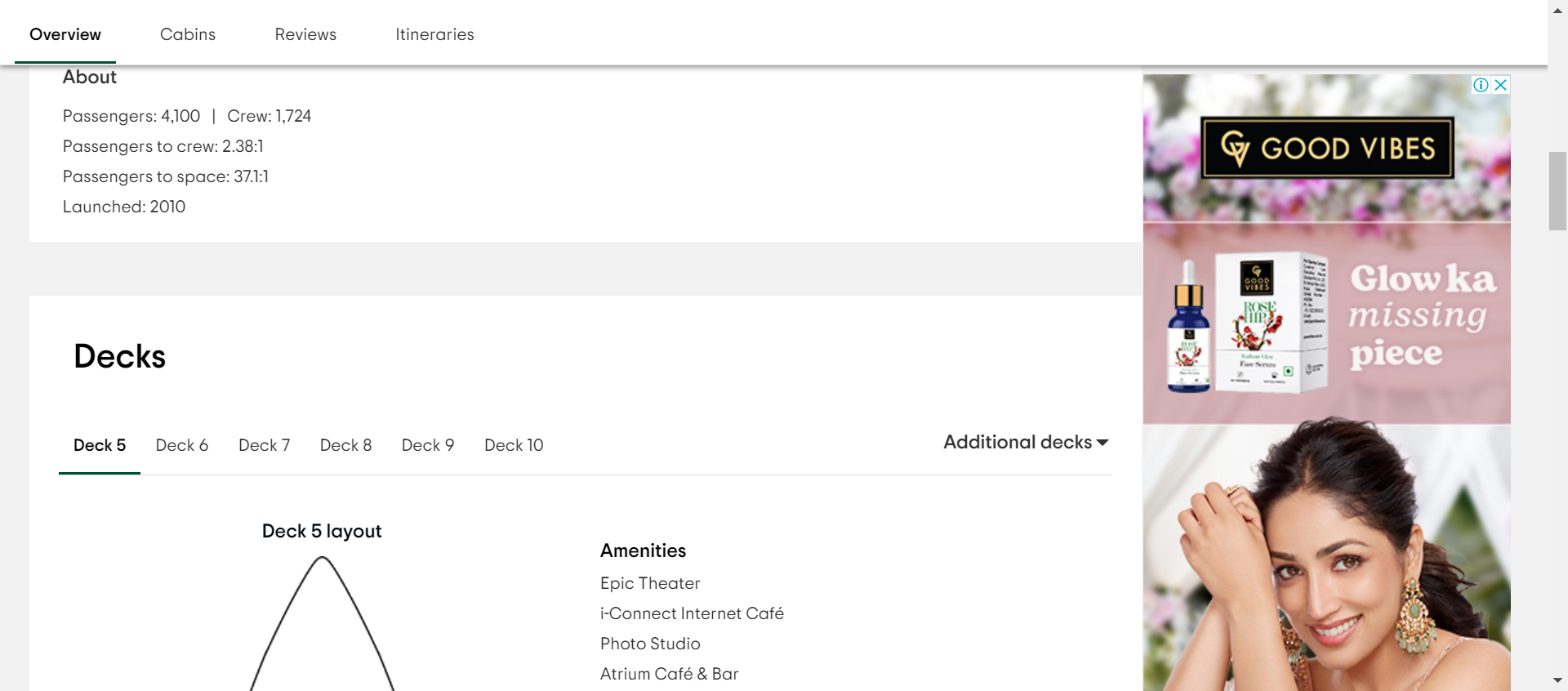
After selecting “Cruises” it redirects to the Cruises page

**9.Cruises\_Page**



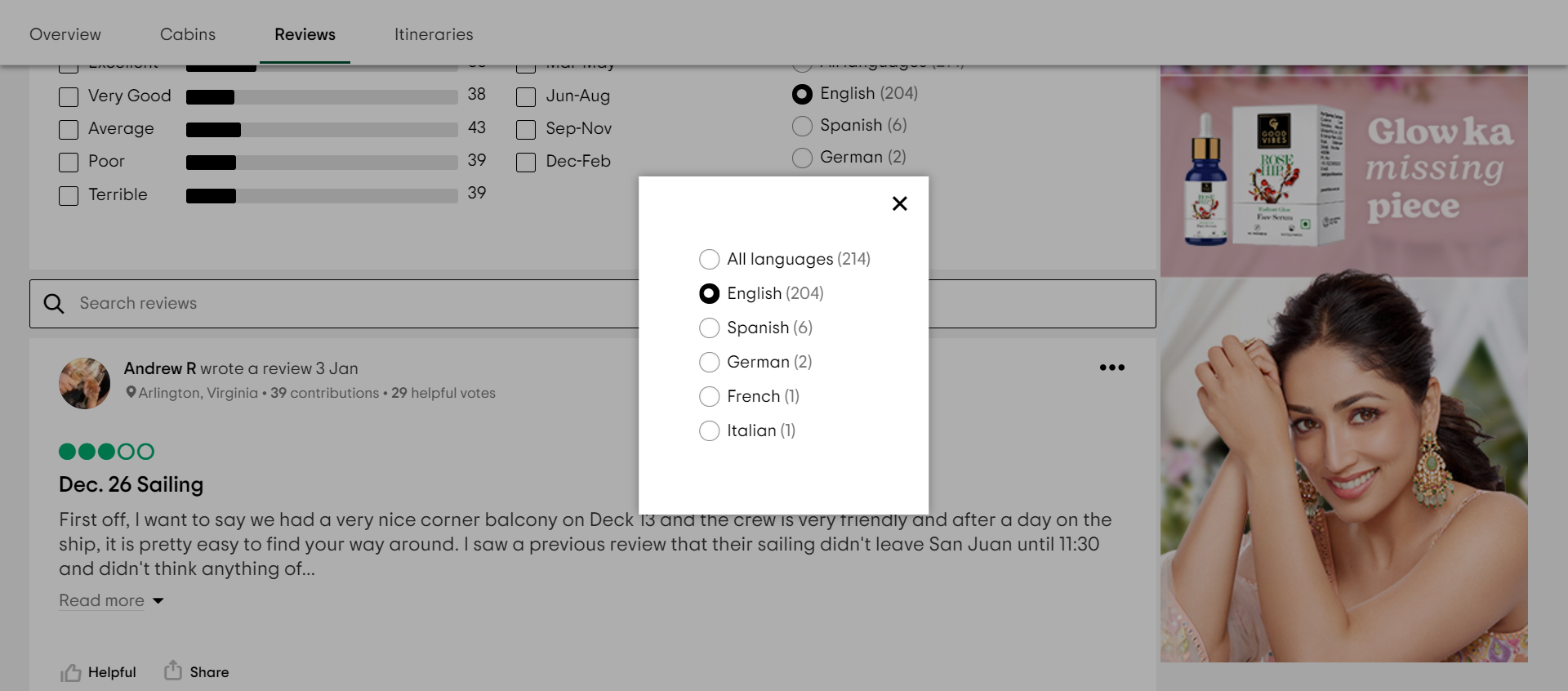
After Selecting cruise line as Norwegian Cruise line and cruise ship as Norwegian Epic

**10.Ship\_Crew**



Above Screenshot displays the passengers , crew members and launched year

**11.All\_Languages**



This Screenshot displays the available languages for the selected cruise line and cruise ship

**After completing the automation in chrome browser, the same process will be repeated in the MS Edge browser**

**AFTER EXECUTION:**

Refresh the project after completing the full execution of code

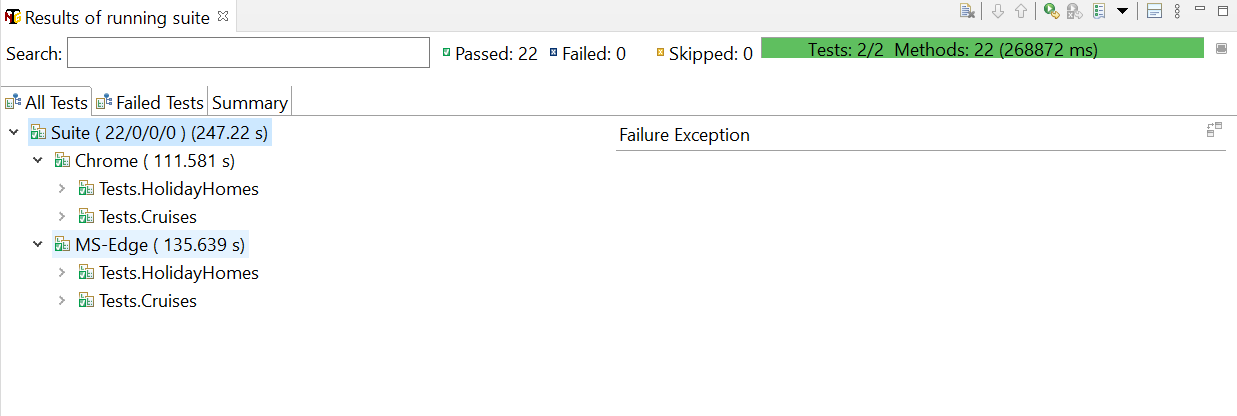
**REPORT:**

**KEY:** We used two browsers to execute the project – One is Chrome and another one is MS Edge.

**CONSOLE REPORT:**

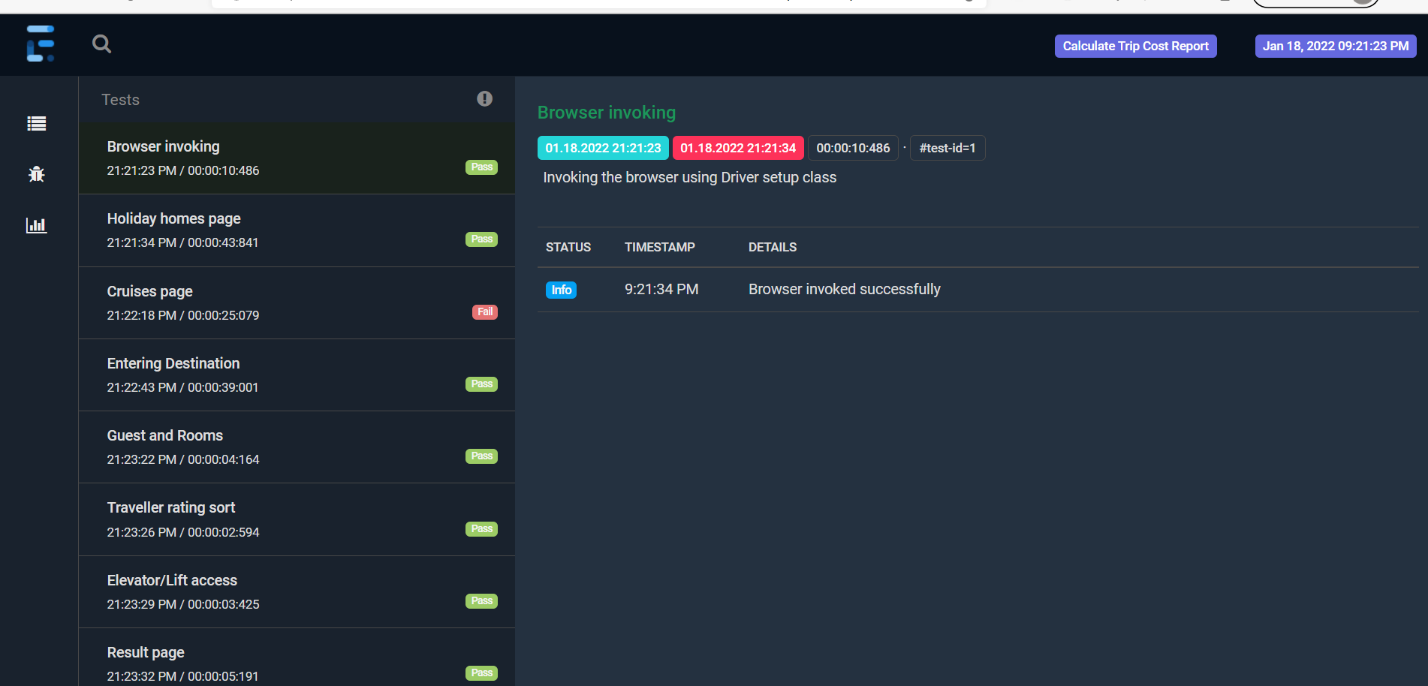
Check the top-rated hotels with the filters applied in the console page.

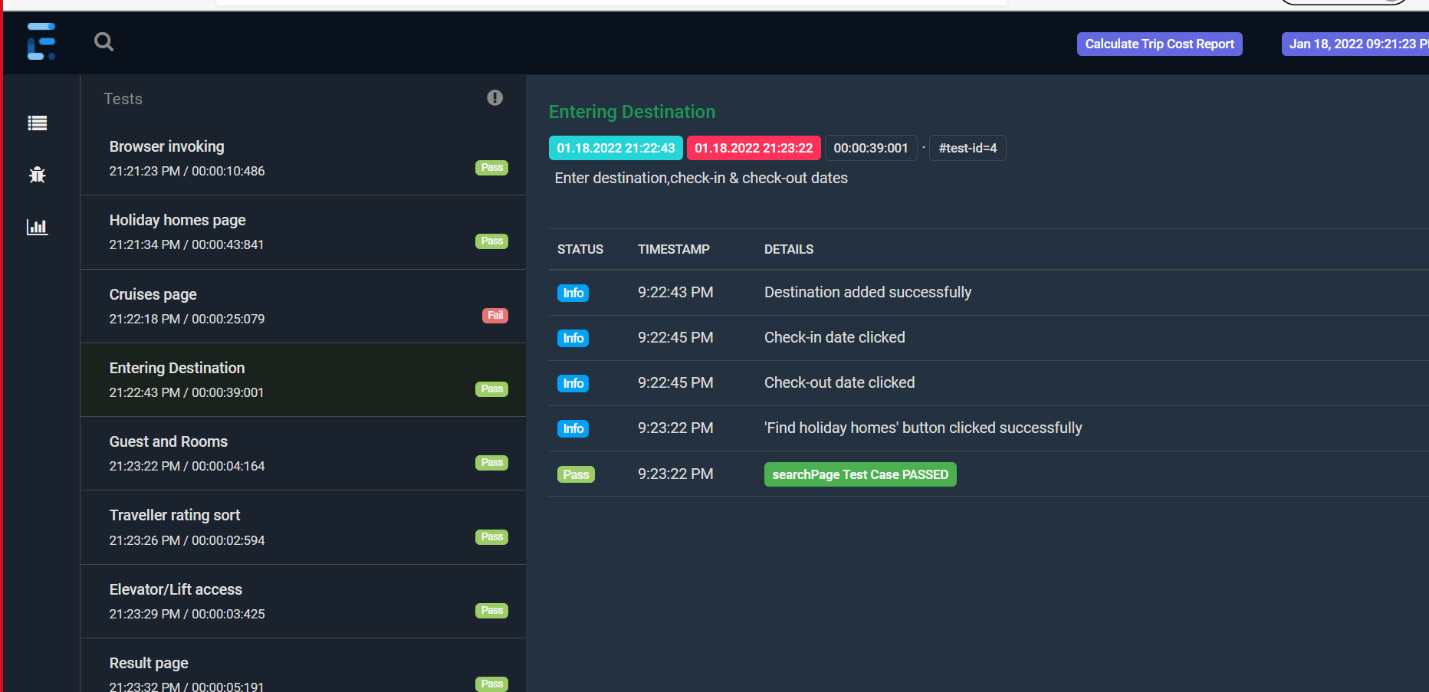
**TESTNG REPORT:**



**EXTENT REPORT:**

No Test cases is failed on both browsers hence no exception is reported on extent report. If Exception is raised the exception message will be captured and stored on the extent report





The results of all the test cases is displayed in the console. The Extent report of the project is generated once the automation gets completed and is seen in the Extent\_report folder . All the screenshots will be available in the “ScreenShots” folder.