11/18/2022

Group Name: Radeon

Manan Amitkumar Patel – S361298

Milan Sukhadiya – S355516

Sahil Hada – S359974

Harashdeep Singh – S360797

Ridham Nadoda – S353334

PRT582 – Software Engineering: process and tools

Software Group Project - Report

**Table of Contents**

[1. Abstract: 2](#_Toc119269139)

[2. Introduction: 3](#_Toc119269140)

[A. Background: 3](#_Toc119269141)

[B. Aim: 4](#_Toc119269142)

[C. Requirements and User Stories: 4](#_Toc119269143)

[D. Team Composition and Task Allocation: 14](#_Toc119269144)

[3. Non-Functional Testing: 15](#_Toc119269145)

[A. Usability testing: 15](#_Toc119269146)

[B. Performance testing: 22](#_Toc119269147)

[C. Security testing: 29](#_Toc119269148)

[4. Functional Testing: 32](#_Toc119269149)

[A. Test plan and Automation Strategy: 32](#_Toc119269150)

[B. Comparison and Justification of Automation Tool: 33](#_Toc119269151)

[C. Test cases and results: 35](#_Toc119269152)

[5. Conclusion: 60](#_Toc119269153)

[A. Ethical Consideration: 60](#_Toc119269154)

[B. Future Works: 60](#_Toc119269155)

[C. Lesson Learnt: 61](#_Toc119269156)

# **Abstract:**

The following report covers the extensive testing of "Movie Ticket Booking System," a web application written in PHP using front end technologies such as HTML, CSS, and others. The program is linked to a SQL database and was designed for quick data store and retrieval. JavaScript is the most common programming language, accounting for around 47% of all code, followed by HTML (40.9%), PHP (1.9%), and CSS (10.6%). It is important to note that the following will be tested using the ROBOT framework, which will test functional testing.

Considering the website's limitation is interface limitation, where the mobile experience is limited from a design standpoint. Furthermore, certain e-commerce functionality is absent when compared to other comparable idea websites.

ROBOT Framework is a community, open source, and free to use project that is constantly updated by community members for bug patches, new advancements, and other similar updates that might aid in improved testing. Automated testing delivers a wide variety of analyses and insights, which may assist bring fresh insights into projects and keep new procedures short. ROBOT Framework was chosen because it is simple to use, has a straightforward, human-readable code structure, and is easily extensible using various Python, Java, and other framework libraries.

# **Introduction:**

## **Background:**

With the advancement of technology and the broad availability of the internet, there has been a revolution in the digitization of numerous entertainment materials. Cinema theaters are no exception in this respect, and sophisticated online booking systems have been built that not only enable clients to pick the seating position of their choice but will also assist them in locating the greatest view. This has given businesses a lot of options and enabled consumers to have a hassle-free booking experience while also checking their internet exposure. Fans may now book their favorite premieres from the convenience of their own homes, eliminating the need to travel to a ticket office, which is yet another outstanding procedure. Similarly, it has aided theatres in expanding their operations and satisfying moviegoers' needs.

The following project is centered on the concept of giving viewers with an online platform for purchasing tickets, picking their own and customizable seating class arrangements, and other options. It is required for the following procedures to supply these demands via the use of multiple technology stacks, which will not only offer a faster process but will also aid in the development of interfaces that are not only attractive to the eye but can also be implemented. The following project makes use of the SQL Database for fast-paced data access and storage, making it incredibly quick to book tickets, log in, log out, amend their changes, and assisted in giving thorough implementation.

## **Aim:**

* Various methods of testing will be used in the following report to completely assess the overall functioning, performance, Security, and usability ratio of the software and web application.
* The report's goal is to provide the necessary conclusion on whether the system is completely equipped to manage with all types of loads and scenarios given by the user, as well as to assist in striking the ideal balance between the performance and usability of a web-based application.
* The comments received while testing the web application and the flaws discovered in the framework may be displayed and implemented to improve the application and make it more relevant.

## **Requirements and User Stories:**

The following web application includes several relevant features that are essential to its operation and are listed below. These functionalities are included into the web application and are a necessary component of the appropriate solution. These relevant software user stories are described below.

**(1): Theater Module:**

As a User Admin

I want able to configure my dashboard and activities

So that I can manage the web application accordingly.

**Acceptance criteria:**

I know I am done when:

1. Able to add movies in the system.
2. Able to edit show timings.
3. Able to edit upcoming movie lists.
4. Able to access movie attendance.
5. Able to edit movie trailer list.
6. Able to add new movies news.

**Size:**

Graphical user interface, application

Description automatically generatedMedium

**(2): User login Module:**

As a User

I want able to login/sign up and logout at any time

So that web application can manage user profile.

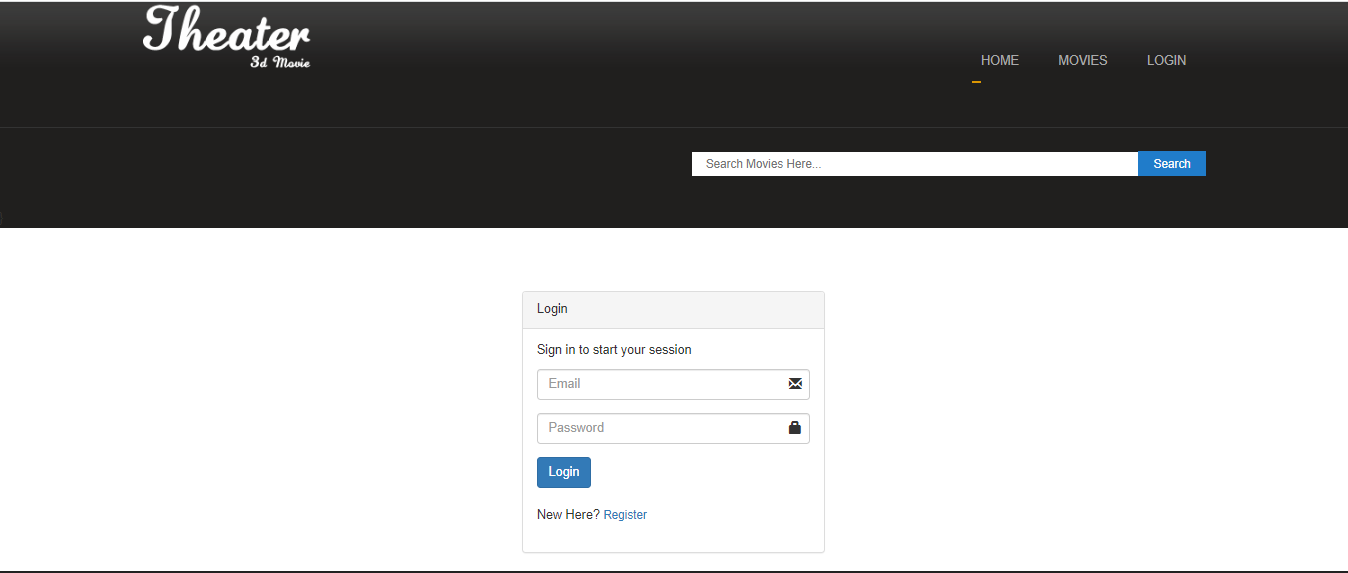
**Acceptance criteria:**

User can do following things:

1. New user able to sign up in the system.
2. Existing users can login into the system.
3. Logout the system when user wants.
4. There should be forget password option where users can change their password if they forget their password.

**Size:**

Small



**(3): User registration:**

As a User

I want able to register in the system

So that I can manage my profile in the web application by login In and out

**Acceptance criteria:**

User can do following things:

1. Allows the user to register in the system.
2. Perform proper validation in data Input.
3. Able to browse the website with full user access.
4. Able to see previous booking of the movies.
5. Able to search the movies.
6. Able to book the movie after registration of the user.

**Size:**

Small

Graphical user interface, application

Description automatically generated

**(4): Movie selection**

As a User

I want able to search movie and select time shows.

So that web application can show appropriate timing according shows and movies according to the search.

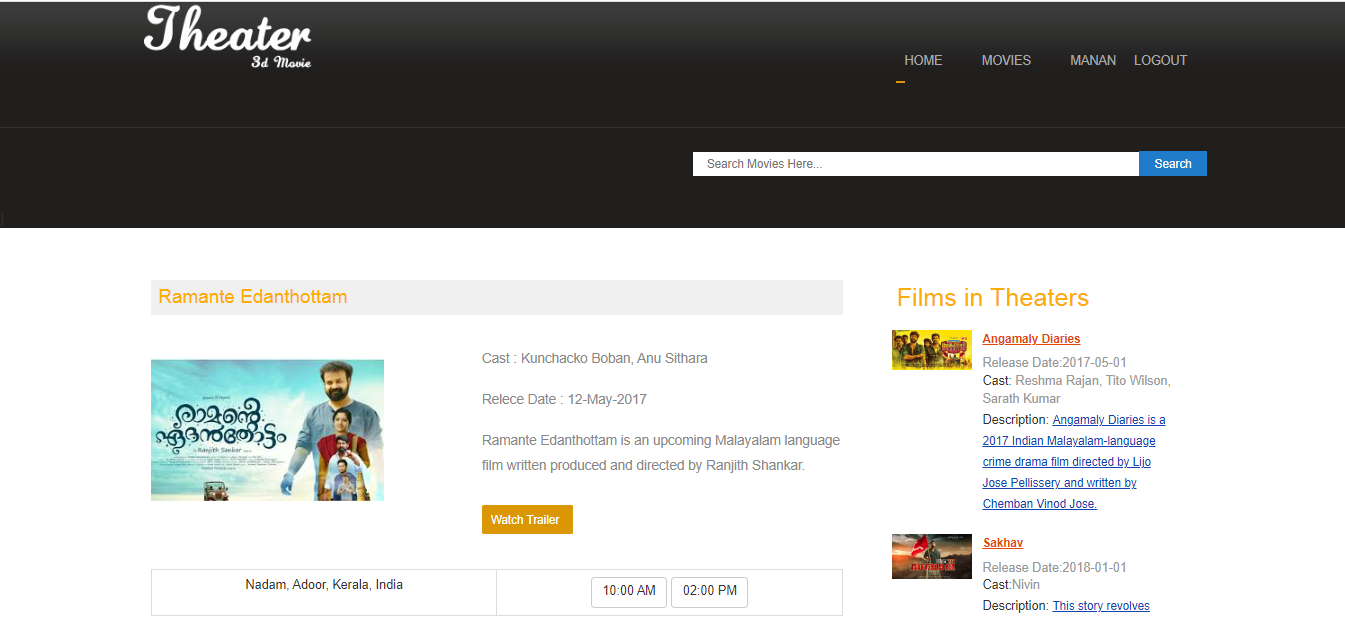
**Acceptance criteria:**

User can do following things:

1. Click on the search button and enter movie name so that find the result.
2. Select that movie and redirect to the trailer and movie show timing page.
3. Select any of the suitable timing from the list
4. At the end do the seat selection on that show and proceed to the payment.

**Size:**

Small



**(5): Watching trailers:**

As a User

I want able to watch the trailer of selected movie.

So that Users can watch the trailer of the movie by simply selecting watch trailer and redirect them to the video application or website.

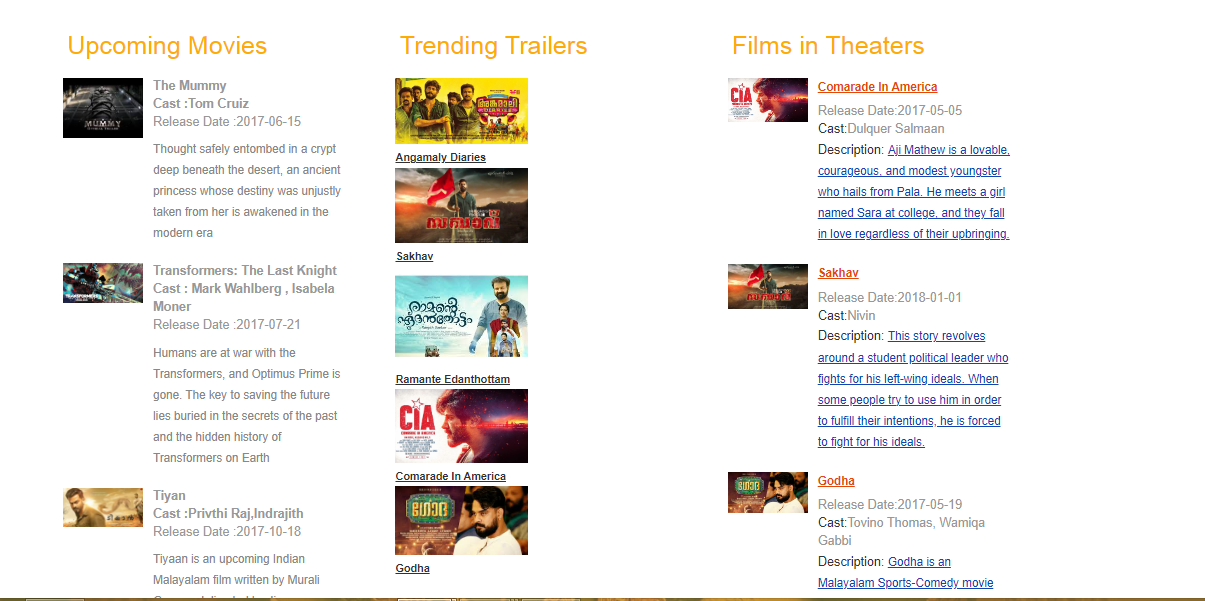
**Acceptance criteria:**

User can do following things:

1. Click on the selected movie option.
2. Play the trailer that is available on movie and shows page.
3. Pause and play the trailer according to the third-party player option.
4. If possible, able to watch the trailer in the web application rather than redirecting to the other web application.

**Size:**

Small



**(6): Making payments:**

As a User

I want able to do payment with credit/debit card.

So that web application can book the tickets when received the successful response from payment gateway.

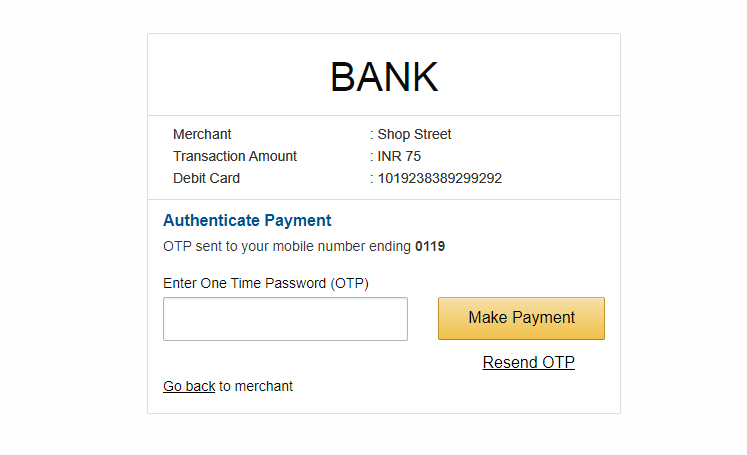
**Acceptance criteria:**

User can do following things:

1. User can proceed to the payment page after selection of seats.
2. Enter the card details (card number, name and CVV)
3. Redirect to the OTP page.
4. User can enter the right OTP that will be receive in mobile number.
5. After successful response redirect back to the site.

**Size:**

Large



Now, with adequate justification, the functional and non-functional criteria are listed below:

**(1): Functional Requirements:**

An unregistered user cannot reserve a ticket; to do so, the consumer must first register. The first step in ordering tickets for anyone is to register.

For purchasing or selling movie tickets, there are several aspects or functional requirements. Some of them are listed below.

* Registration
* Login
* View Upcoming Movies
* View current movies trailers
* Search movie
* Payment
* Validating OTP
* Generate ticket
* Logout
* Add movie
* Add trailer

Now that we have an overview, we can outline all these functional needs.

* Registration:

The consumer must register if he/she wants to book the movie ticket. Customers who are not registered cannot book movie tickets.

* Login:

Customers that have a valid user ID and password can log in to the system.

* View Upcoming Movies:

Customer can view the upcoming movies name and description without login and registration on the home page.

* View movies trailers:

Customer can view the current movies trailers without login and registration on the home page. This will redirect to the video player in their browser.

* Search movies:

For customer reference while reserving tickets, customers may check the title and description of the most recent movies on the homepage without logging in or registering.

* Payment:

Numerous billing options for customers will be paid in advance as debit or credit. Any client who wants to see a movie must pay.

* Validating OTP:

Proper gateway has been used for payment using validating OTP (One time Password).

* Generate ticket:

Once the payment has been validated and successful, customer will get the generated movie ticket to be shown in their portal.

* Technical issues:

System work on client-server Architecture. It will require an internet server.

* Interface requirements:

Various interfaces for the product could be:

* Login page
* Registration page
* Home page, which consists data of Upcoming movies, trending trailers and films in the theater.
* Movie Page, which shows current movies on theater with show timings and trailer.
* Admin page, where admin can manage the movies and view the movie attendance.

**(2): Non - Functional Requirements:**

* **Security:**

After a certain amount of inactivity, the system needs to log off all users automatically.

The system shouldn't save any cookies revealing the user's password on the client's machine.

Admin can only be able to manage to add movies and setup the show timing and all.

Before being delivered across unsecure links like the internet, sensitive data will be encrypted.

* **Availability:**

The user should always be able to access the system via a web browser, with the sole exception being any downtime experienced by the server that hosts the system.

The administrator should download and store database backups from the server in case of hardware failure or database corruption as well.

After that, the service will restart. It denotes availability every single day of the week.

* **Maintainability:**

Proper database should be used for managing the customer confidential data.

* **Portability and backup:**

The application is built using PHP and JavaScript. Therefore, the end-user portion is completely portable, and any device running any web browser.

Moreover, proper backup of the data should be taken using third party backup tools.

* **Accessibility:**

The system will be a web-based application it is going to be accessible on the web browser and device like mobile or computer or laptop or tablets etc.

* **Performance:**

Initial loading of the product will take some time, depending on the speed of the internet connection and the media source.

The client's/hardware customer's components will determine performance.

* **Supportability:**

This system's source code must be kept up to date in a configuration management tool.

## **Team Composition and Task Allocation:**

We formed a team of five members and named it Radeon. We assigned assignments based on the abilities and characteristics of each member. Having varied experiences and abilities that work together for us as a team to complete the project testing.

We have assigned distinct tasks and duties based on the expertise of our team members. The table below depicts the project's task allocation.

|  |  |
| --- | --- |
| Team members | Roles and Responsibilities |
| Harashdeep Singh – S360797 | * Creating User stories * Creating Functional and Non-functional Requirements. |
| Milan Sukhadiya – S355516 | * Performing Usability Testing |
| Manan Amitkumar Patel – S361298 | * Performing Performance Testing * Performing Functional testing using Robot Framework |
| Ridham Nadoda – S353334 | * Performing Security Testing |
| Sahil Hada – S359974 | * Helping with functional and Non-Functional Requirement. * Performing Functional Testing using ROBOT framework. |

# **Non-Functional Testing:**

## **Usability testing:**

Usability testing is a vital aspect that demonstrates the ease of access the user will have when exploring various areas of the website and experimenting with various capabilities. There are several things that can be incorporated with proper usability testing and feedback. It is vital to distribute their needed work, which may be accomplished by focusing on usability via various technologies. Some of these are defined further down with the proper comparison.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Sr No.* | *Tool Name* | *Description* | *Advantages* | *Disadvantages* |
| 1 | Maze | Allows the collection of qualitative and quantitative data all at one place. Can be integrated with sigma, adobe ad and other. | Supports all kinds of testing such as  A/B Testing  Click Testing  Prototype Testing  Provides detailed insights and analytics as needed. | Prices can be quite high when upgraded to premium and free account doesn’t caters for all the required needs. |
| 2 | Lookback | A user experience UX screen recording tool which can record facial features which change of the user when the user works on different perspectives. | The real time remote research feature is the biggest plus since the up designer can get feedback on the system. Similarly, the analytics can further help in finding the issues. | The UX tool is only applicable for real time applications and doesn’t provide with solid and tested features testing mechanisms. The facial expressions might not be the best way to find the issue. |
| 3 | TryMyUI | Like lookback but with more flexible options such as demographics curation, survey featurette and other such processes. | The tool for usability testing provides great groundwork for prototype testing, survey testing and other analytics. | Unlike the Lookback it doesn’t have the facial expression recordings which can be a real plus for testing usability of a website. |
| 4 | Usability Hub | Usability Hub is a remote user research platform that evaluates design decisions with real users, reducing uncertainties from the design process. | Programmers may use Usability Hub to develop customized surveys, add and edit many questions, and apply filters to replies based on a variety of parameters. | Participants reported confusion about how to complete some tasks, especially those involving graphics, because of unclear instructions. In most cases, we'll provide spoken guidance in addition to visual indications. |

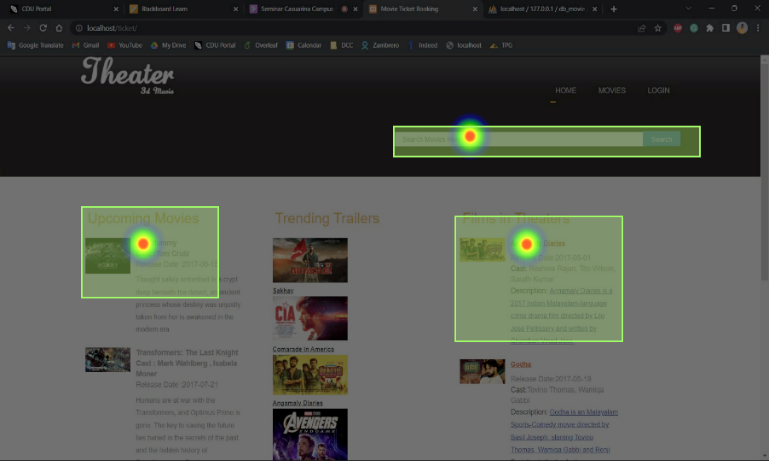
* **Usability Hub:**

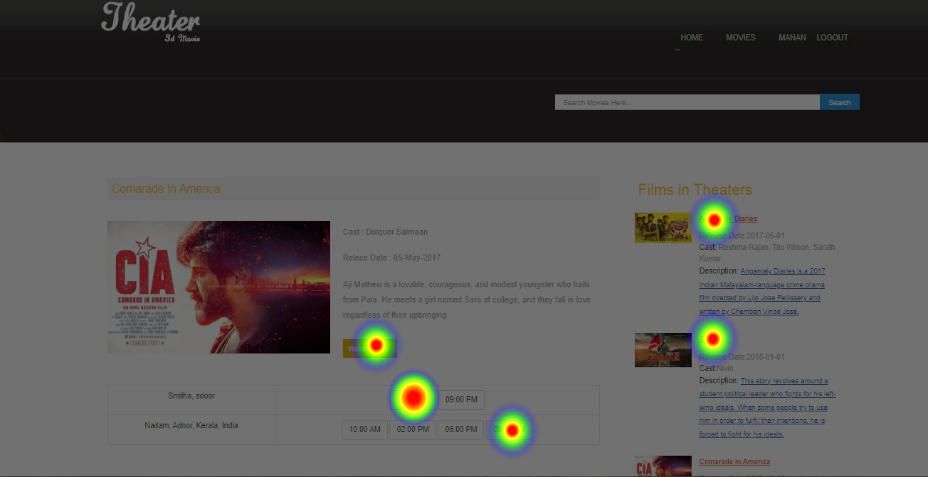
Usability Hub is one of the usability testing tools on this page that aims to be straightforward and easy to use. One of its advantages is its ease of use; participants need just see a single page for a total of five seconds. Then, readers are prompted to share their recollections of the page, and their comments are collected below. Finally, consumers get a word cloud including the most frequently recalled terms. Having this information lets designers make tweaks that will have a long-lasting impact.

Usability Hub’s innovative heat maps, which do more than simply calculate click-density, are one of our favorite features. In addition to the heat map, this system also offers data and statistics, such as the average time it takes for a user to click on a link. In the same vein as previous usability testing tools, it lets users choose test subjects from an existing pool or enlist new ones. If users furnish their own subjects, there is no limit on the number of individuals who may take part in research. If you want to depend on their participant database, you may specify which populations are of particular importance to you.

We do live and remote usability testing without reviewers for this project. The goal of the test is to discover possible defects or weak points in the Movie Ticket Booking System web application. For our testing purposes, four users, two from our group and two from my university colleagues, used the web application. Most respondents had no trouble comprehending the improvement, the product's key selling feature, or how it operated.

Here from the user’s experience, we extracted the heat map which helps us to analyze where the participants click and measure the click density of the web pages of the web application. Participants are allowed to perform 5- seconds test for first click testing and the results screenshots has been posted below:





Graphical user interface, application

Description automatically generated

* **Analysis of Result:**

The heat map pictures are created when a user clicks on a certain sector, and the heat map is used for detail analysis to make the usability interface more suited for the users. Multiple consumers utilized this website, and we developed a heat map of first click testing based on the result images.

* **Suggestions for improving usability testing:**

First, we must competently describe the demand to the consumers. Furthermore, the goals have been clearly outlined for the users.

Assure that users are appropriately choosing their task, which implies that if they want to look for a movie by name, they should properly follow the searching case by picking on that portion of the website.

* **Persona and Scenario Testing:**

A persona is a creative representation of a typical consumer for a particular service or item. A persona is a hypothetical representation of a certain kind of consumer who would purchase your products. This hypothetical image of your ideal consumer that you might use to influence the design of your product.

As per our project we have created two persona and scenario testing based on the typical customers. The results have been posted further in the proper format.

|  |  |
| --- | --- |
| C:\Users\K\Desktop\download.jpg  Malik  Customer | **Profile Background**  Malik is a Customer from Australia  **Goals and Tasks**  He just wants to book a ticket online instead of booking ticket at cinema. |
| **Anxieties and Motivations**  He gets motivation and feel better by watching movie.  **Frustrations and Concerns**  He is frustrated from sold out ticket’s situation.  **Needs and Expectations**  Needs an application for booking tickets online and he want to do payment by online banking. | **Scenario**  Malik wants to save time for buying ticket and select movie by watching trailer. |

|  |  |
| --- | --- |
| C:\Users\K\Desktop\download.jpg  **Robyn**  Businessman | **Profile Background**  Robyn is a businessman from Russia. He is very busy with his schedule.  **Goals and Tasks**  His goal is to save time and secure payment online. |
| **Anxieties and Motivations**  He feels refresh by watching movies and feel calm from daily hectic schedule.  **Frustrations and Concerns**  He is very frustrated by standing in lines for booking ticket.  **Needs and Expectations**  He wants to book movie tickets in few clicks. | **Scenario**  He just wants to book tickets in few clicks and secure payment using OTP. |

|  |  |
| --- | --- |
| profile.png  **Lovin**  Customer | **Profile Background**  Lovin is a customer from China and leaves far from cinema.  **Goals and Tasks**  Her goal is to book ticket online by watching show times and movie types. |
| **Anxieties and Motivations**  She just wants to refresh herself as well her child’s.  **Frustrations and Concerns**  She is very frustrated with show timings because if she comes early than she must wait more for show.  **Needs and Expectations**  She wants relevant show timings according to her schedule. | **Scenario**  She wants to book tickets for upcoming movies as well. |

|  |  |
| --- | --- |
| profile.png  **Riya**  Customer | **Profile Background**  Riya is a customer from India and wants to book seat according to her choice.  **Goals and Tasks**  Her goal is to book back side and center ticket or any seating. |
| **Anxieties and Motivations**  She feels better if she gets seats as per her choice.  **Frustrations and Concerns**  She is very frustrated with seats selection if show is near to full.  **Needs and Expectations**  She wants relevant show timings according to her schedule. | **Scenario**  She wants to book tickets in advance so that she can get seats as per her choice. |

## **Performance testing:**

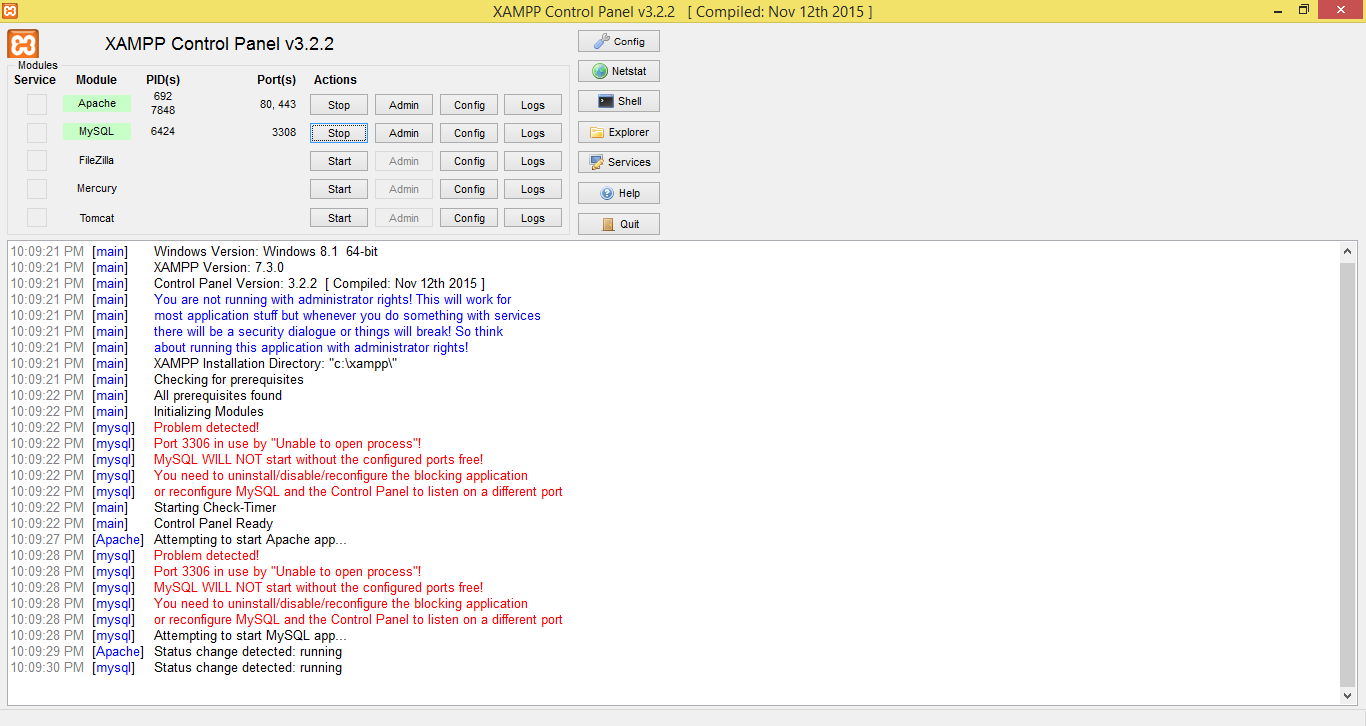
The application's performance testing is determined by how effectively it is written and the different good programming practices that go with it. As a result, they must be incorporated into practice. The testing was done for benchmarking the performance to ascertain the following and to determine if the application was coded in accordance with excellent PHP coding standards.

To educate and assess on the many components of testing, the following table highlights the numerous performance testing tools that may be used to carry out such testing. It is critical to provide effective performance resources for this context. The following are examples of methods for improving performance that have been proved to be effective via research.

|  |  |  |  |
| --- | --- | --- | --- |
| *Sr #* | *Tool Name* | *Description* | *Advantages* |
| 1 | LoadNinja | Provides really quick, script fewer testing services, reduces time for testing by 50 % and provides browser based metrics at exceptional pace. | Real simulations of browser-based loads test.  Debugging option for visual studio in real tests.  Allows managing virtual user activity in real time. |
| 2 | Xhprof | Provides enterprise level, sophisticated test techniques which are relevant to the tests and provides with the reliable, real scenario-based situations and insightful analytics. | Cloud computing powered requiring no installation and minimal memory load. Customer support for resolving queries. Deploys solutions to detect bottlenecks. |
| 3 | Kobiton | Provides flexible computing options ranging from public, private cloud as well as on site testing options. AI powered, script less engine provides options for automated testing. | Kobiton’s Ai power engine helps creating automated tests from manual scripts which are executable on multiple devices. |
| 4 | JMeter | free and publicly available program, written entirely in Java, used for functional load testing and performance evaluation. | There is no installation necessary. In that it enables you to create your own tests, it is very extensible. Additionally, JMeter has visualization plugins that you may use to improve your testing. |

* **JMeter*:***

Here in our project, we have done the performance and load testing using JMeter. JMeter was first created to test the Apache server, but it has subsequently been modified to test a wide range of applications. Here we are running our PHP based on the Apache server.



If we are looking for the performance testing, or we are wondering how much load this site can take and will looking to do some measuring and try to figure out performance of the Movie ticket Booking System, JMeter is the best appropriate option for us. JMeter's extensive reporting options and ability to simulate high demand by making repeated requests to the target server may be used to examine performance.

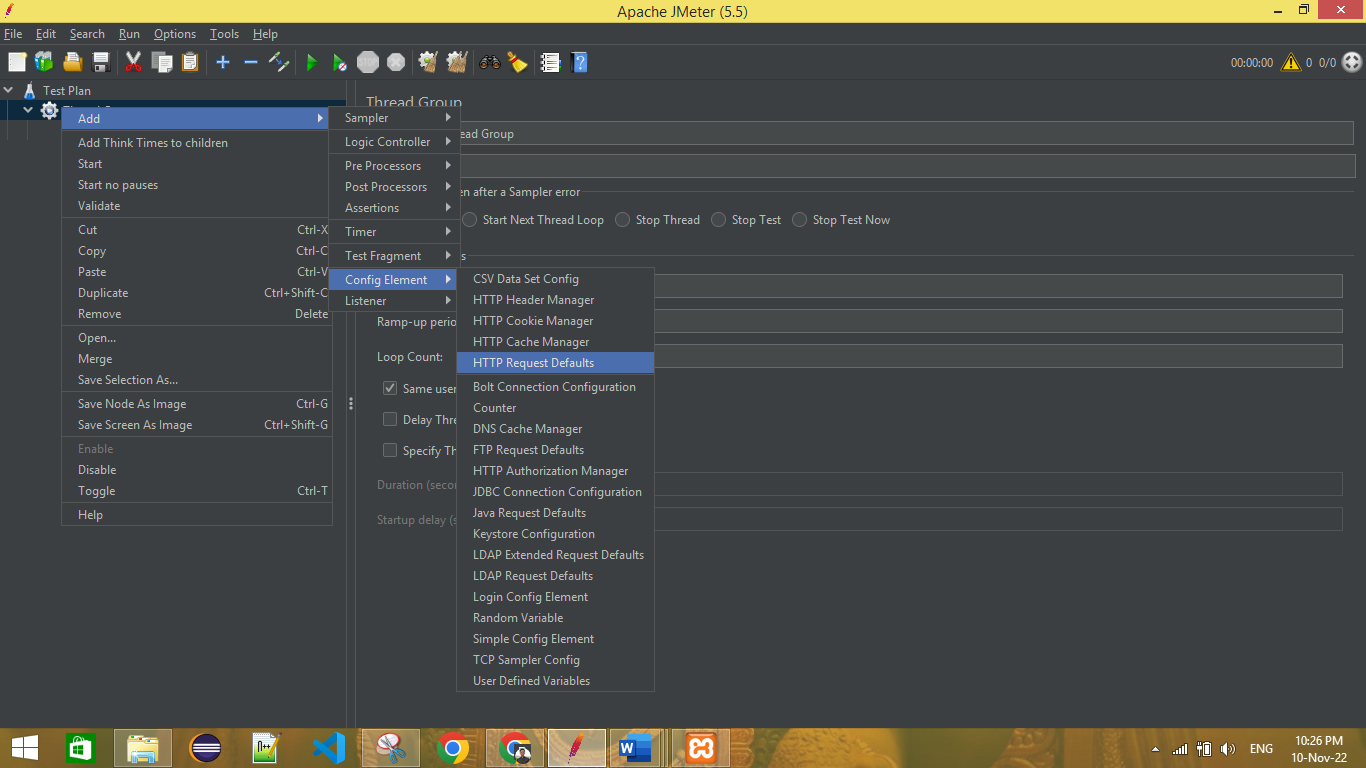
JMeter's graphical user interface (GUI) is used to construct tests, and these tests are structured in a tree-like hierarchy that specifies the hierarchies and sequence of actions. Moreover,

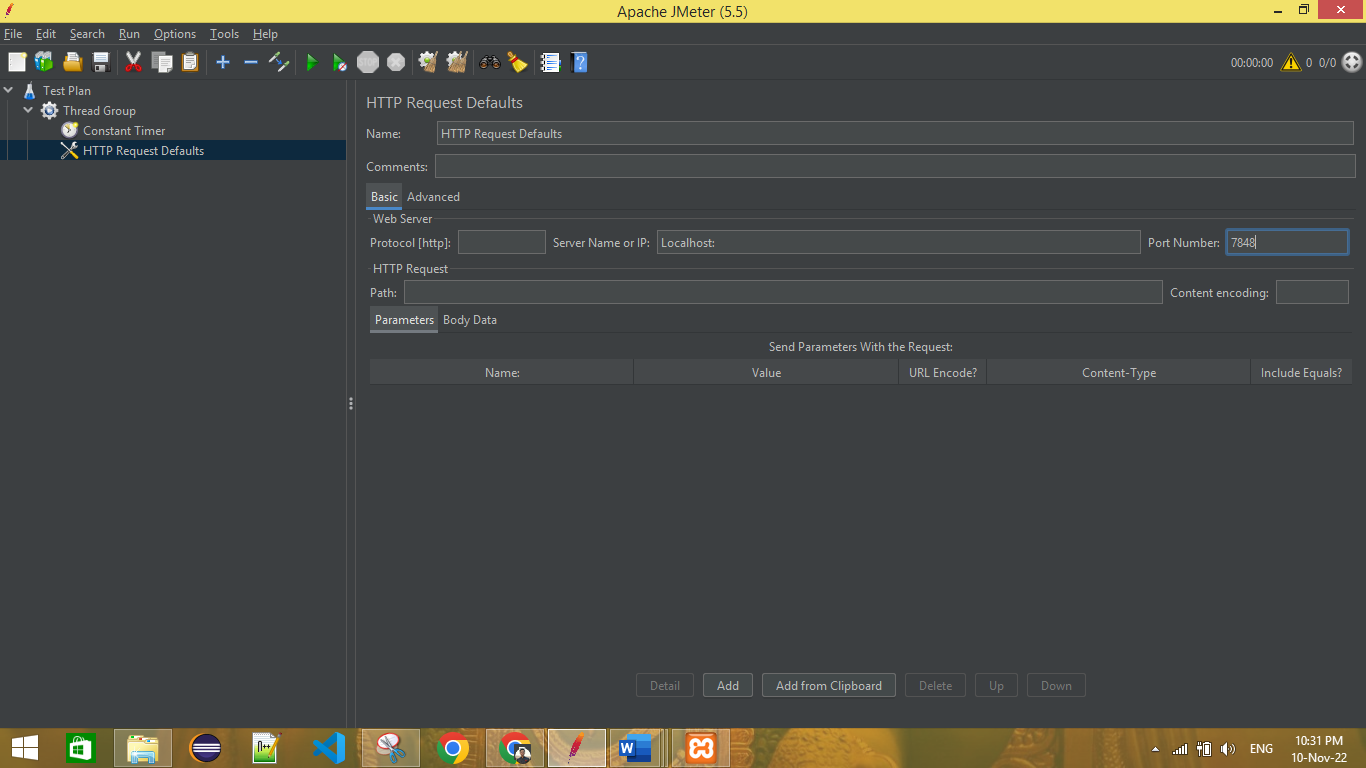
According to our project testing, we developed a test plan by creating a new thread group with different properties. Here, the number of users is stated to be 50, and the ramp-up period per second is declared to be 5. The screenshot below shows JMeter's capabilities for establishing a new test plan.

Graphical user interface, text

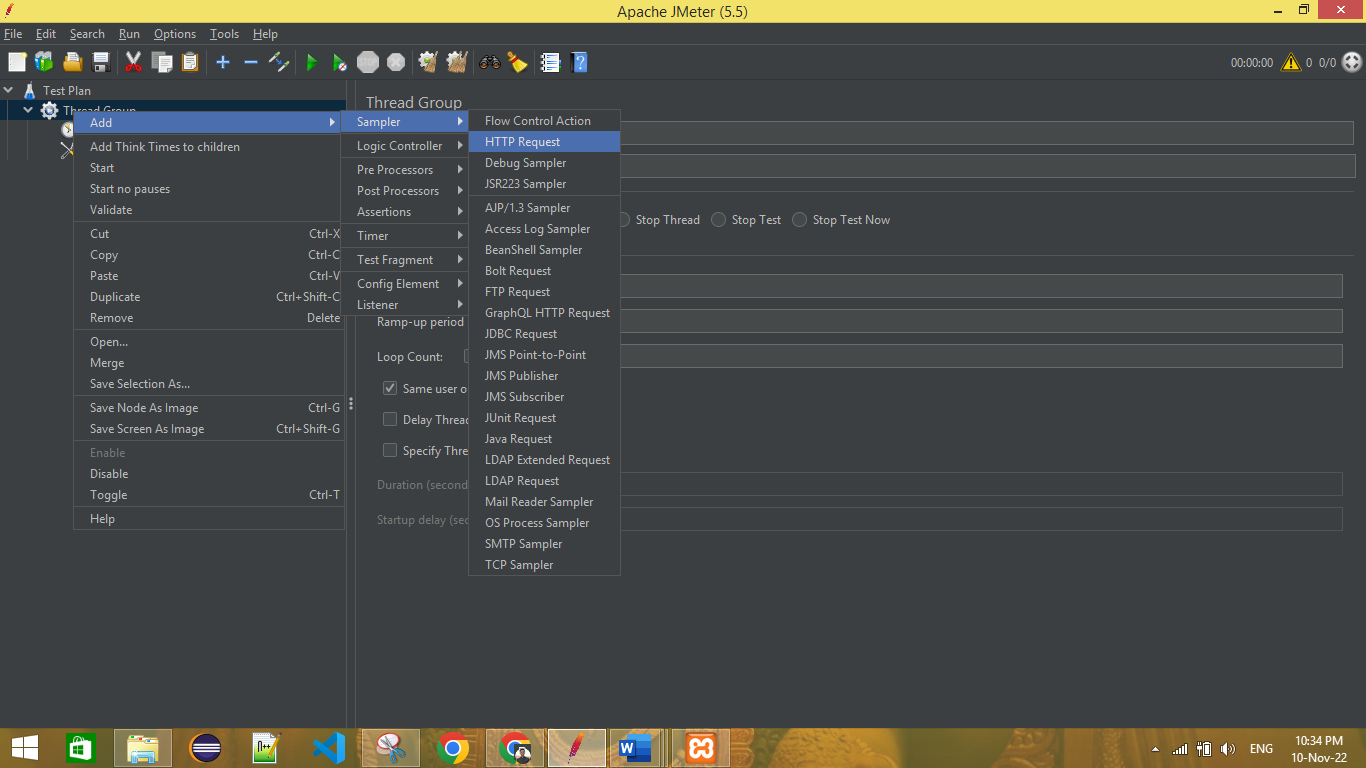
Description automatically generated

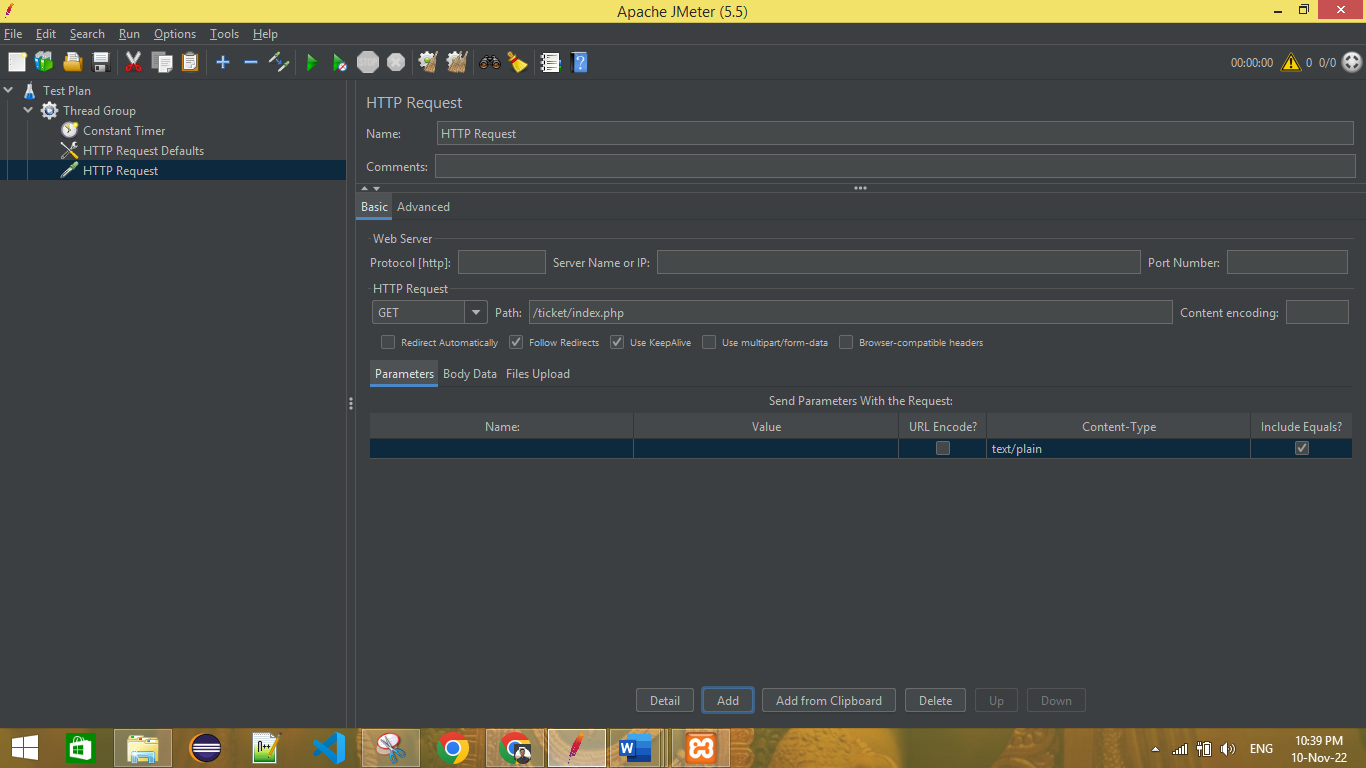
Now, that we have the open-source web application operating on our local host, we need to add the config components of HTTP request defaults, which will make the default requests on the localhost with our PORT number stated. The two screenshots below show how that capability works.



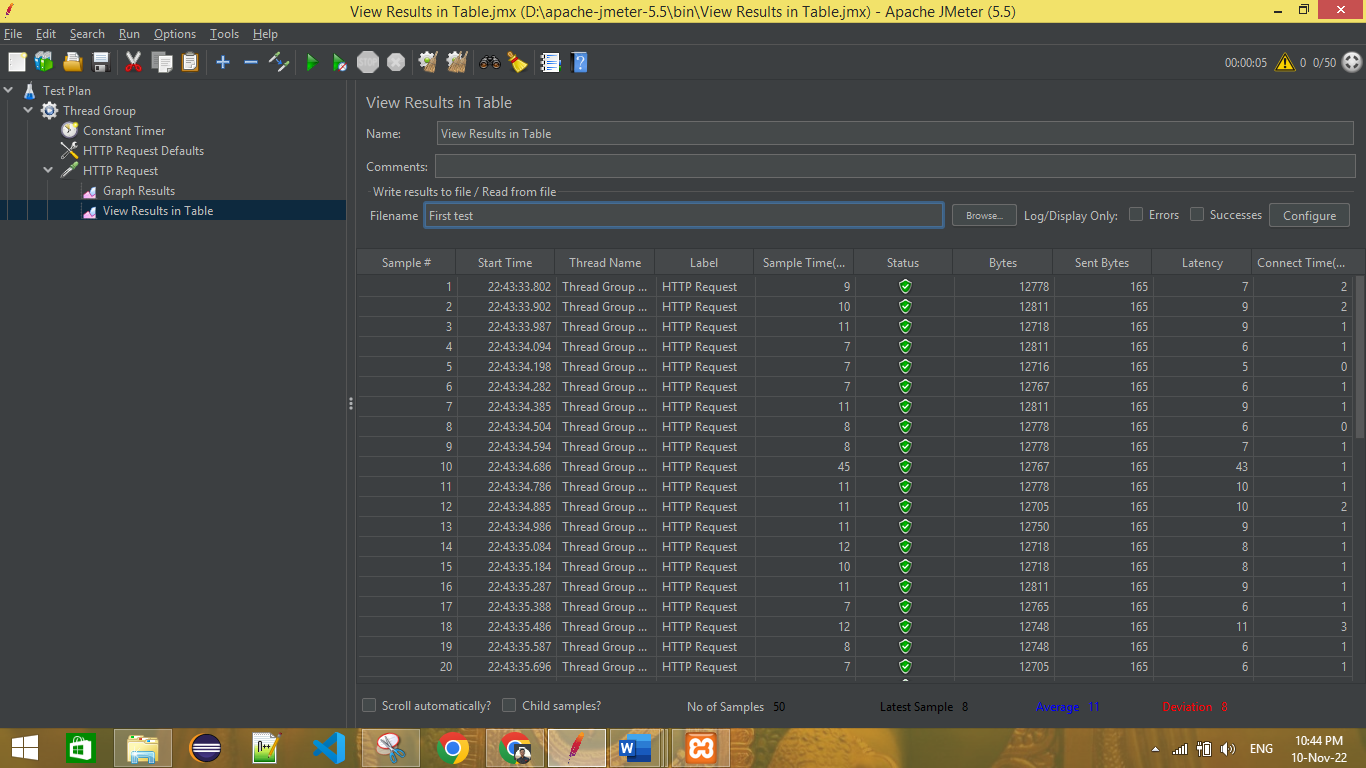


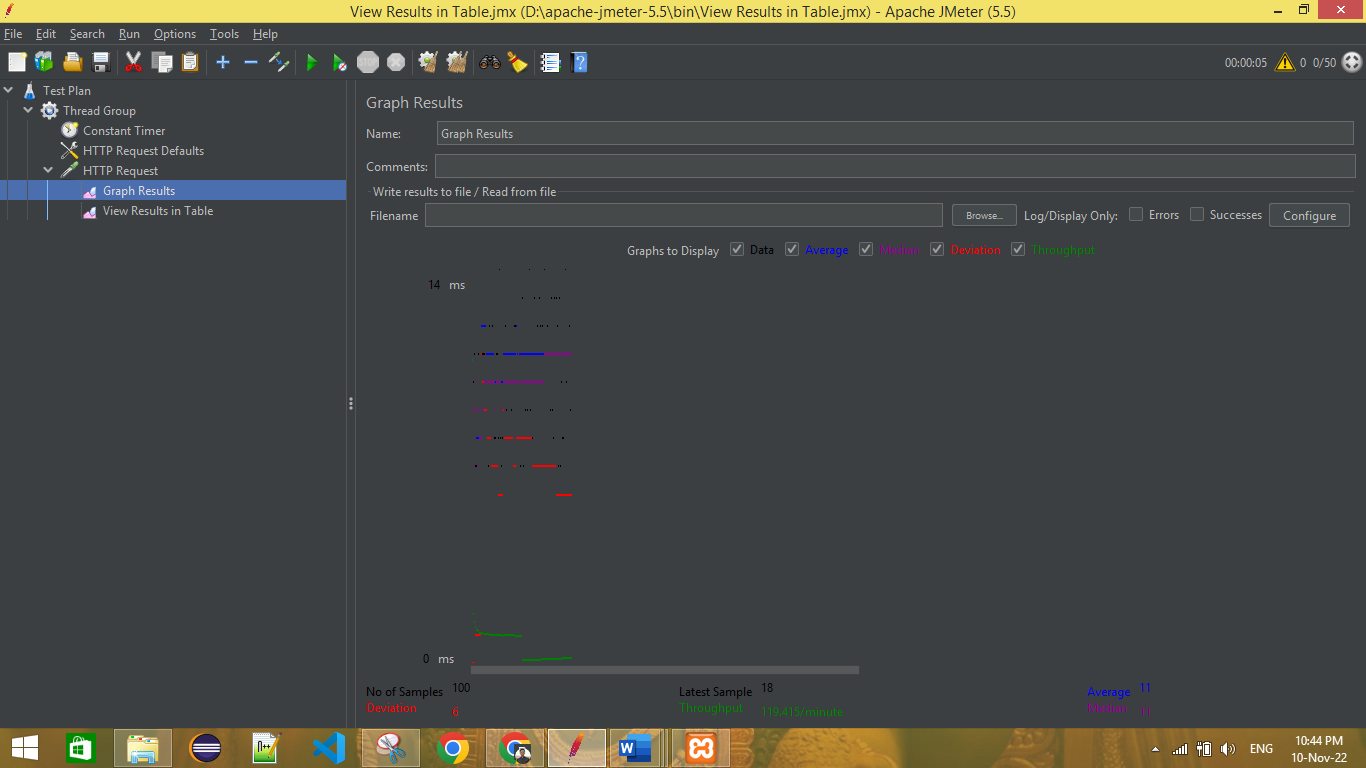
Now, we must include an HTTP sampler, which will enable us to send several requests to the server. This indicates that the sampler is sent to the web server for testing using JMeter Http Requests. We must include the Web application path and run the test. The screenshots below show this feature.





Now, as we need to perform the testing after all setup of threads and HTTP requests server and path, we need to add the listener for analyzing the results. Here we have added the Graph representation results and Table representation results. So, as per the results, we have performed the performance and load testing of the Online Movie Ticket Booking System.





* **Analysis of Result:**

We can see from the picture above that two sorts of test results were generated. One is in graphical format, while the other is in tabular format. Analyzing the table structure of the results reveals that the various thread users completed the test with a success status. The number of threads is 50, with a variance of 6 and an average time sample of 11 minutes.

* **Suggestions for improving Performance Testing:**

Prepare the server route for the HTTP request to submit the test case to the web server as a sampler. Furthermore, we must specify the number of threads that will be used to assess the results and make suitable modifications.

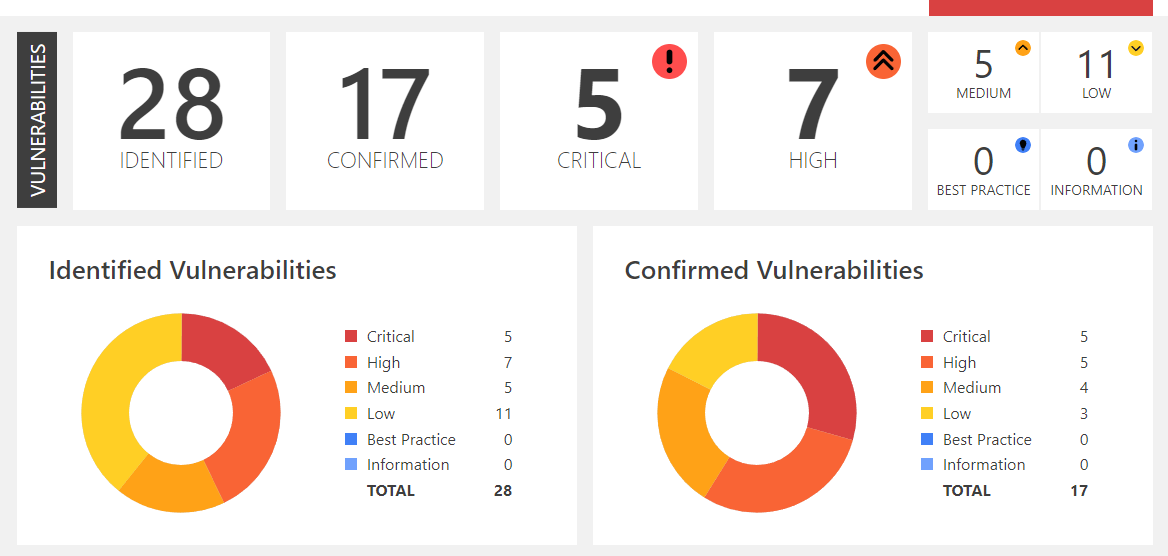
## **Security testing:**

The application's security is a critical general feature that must be applied and implemented to ensure that the problem is handled. Security may be linked, and numerous security tools can be supplied; below is a list of them, along with a comparison.

|  |  |  |  |
| --- | --- | --- | --- |
| Sr # | Name | Description | Features |
| 1 | Google Nogotofail | Google powered security testing tool which can help provide with various processes and can be applied as needed. Works with web, IOS, Android and window applications and can be tested to find various secuirty lapses. | Provides with the deteciton of HTTP related errors such as  SSL Certificate  TLS/SSL bugs  MiTM Attacks  Can be used as proxy, VPN server, router etc. |
| 2 | SonarQube | A code quality monitroing application which can provide optimization insights based on static code analysis. Can be applied for 25 plus programming langauges such as C#, JS and VB. | Provides customizeable security standards which the application must meet.  Built in Sonar way which is automated way of developing the best in perfromance and quality standrads which are compliant with security standards. |
| 3 | Invicti | An enterprise level security setup and testing tool based on blackbox, for identifying flaws in security proportions, websites and web applications. | Integraiton with various source controls such as Github, Azure and other related.  Provides actionable evidence and also actions to fix the issue. |

**Invicti Tool:**

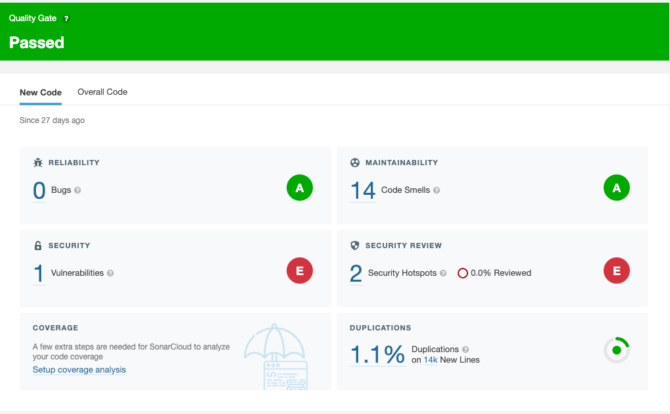
The program was created using an upgraded version of PHP, which addressed all security concerns as vulnerabilities were addressed in the code. However, since there is no genuine SSL certificate as a standard, the application is vulnerable to security vulnerabilities in which the information given may be retrieved and applied. It is vital to use SSL certificates to protect the application from hackers. SQL injections are also a risk and may cause an internet outage. Validations are not implemented in fields, which may result in incorrect data submission where the query to the database may result in errors, preventing the application from operating.



**SonarQube Tool:**

Our major security testing emphasis is code quality and code security, as well as improving process. We also concentrate on application security and technological debt. We also created regexes to prevent stack overflows and other sorts of integration and feedback during the process. SonarQube enables us to do various tests quickly and simply.

Here the screenshots of the security testing have been provided further.



A screenshot of a computer

Description automatically generated

* **Analysis of Result:**

Both Invicti and SonarQube security testing tools provide detailed reports on flaws, code smells, vulnerabilities, and code duplications. According to our findings, we have 1.1% code duplication in the registration and login page code in PHP. Furthermore, code maintainability is 14%, which includes incorrect database and misdirected programming, such as not creating the ID of the input variables for automated testing.

* **Suggestions of Improving the Security testing:**

In terms of strengthening the security of our website, there is a possibility of SQL injection attack where the code of our website database has not been created properly. So, the initial proposal is to improve the database code's proficiency and dependability. Another idea is for code maintainability, where duplications of code should be reduced and measured during all frequent updates.

# **Functional Testing:**

## **Test plan and Automation Strategy:**

Test automation, or the use of an automation tool, is a major step in the software development process since it enables the developer to run automated tests at predetermined intervals, examine the results, and suggest improvements. The application has been well laid out, and there are several frameworks available to ensure top-notch testing and application for optimal results. Since automated testing pushes the limits of end user demands and what is possible for them, it must be included into the final system.

**Automation Strategy:**

The next phases, each of which will be outlined below, will indicate the automated testing approach as well as how the testing will continue with testing requirements being examined and overlayed.

1. **Defining Goals:**

The following items would make up the final objectives of the automated testing, and their completion rates, as well as the applicability of their performances and an analysis of their benchmarks, would provide a clear picture of their accomplishments. Determining precise goals for the way the system need to function will be given priority, and those goals will be taken into consideration.

1. **Comparing Results:**

Benchmarking will be the criterion here, with the user cases evaluated and the subsequent findings supplied so that a comparison can be made on how the system operated, met expectations, and where the results might have been confirmed. All of this would then be constructed based on the logging of the robot framework, and reuses could be identified.

## **Comparison and Justification of Automation Tool:**

Here in the below table, we have discussed the different types of automation tools are available in the market with the brief description and features.

|  |  |  |  |
| --- | --- | --- | --- |
| Sr # | Name | Description | Features |
| 1 | TestProject | TestProject is a framework for completely automating the testing of mobile and web apps. It aims to simplify your test automation experience while also giving endless customizing choices through a gallery of add-ons. | One of the most important features of this automation solution is the in-browser smart recording test recorder. Furthermore, it has an Insite element inspector and explorer for specific testing elements. Further it is available for free. |
| 2 | MicroFocusUFT | MicroFocusUFT is a tool that assists with the functional testing of corporate applications for the web, mobile, and API platforms, and it cooperates with end to end testing. | A network of authorized partners that are spread out over the world. In addition, having a large collection of useful resources readily available on the internet. Working with a diverse array of technological options. |
| 3 | IBM Rational Functional Tester | Automated functional and regression testing may be performed with the help of IBM Rational Functional Tester. This software supports automated UI, functional, regression, and data-driven testing. It's compatible with a broad variety of programs, including those that run in a web browser (.Net, Java, etc.). | The ability to see how a service works while testing makes the process more productive. Compatible with any web browser and designed to work with other programs. |
| 4 | Robot framework | Robot framework is a generic open source automation framework. “ROBOT Framework” which is a python-based framework which provides with detailed considerations on the keyword-based use cases. | The Robot Framework is free and easy to add on to. Strong and adaptable automation systems may be built using Robot Framework in conjunction with almost any other tool. Any usage of Robot Framework does not need the purchase of a license. |

Here from the above demonstration, we have selected ROBOT framework for the automation testing of our web application.

Following are the big advantages Robot Framework has over the others which makes it an ideal choice for automated testing of our website.

**Variables:**

Complex use cases are supported using variables such as dict, scalar and list helping in the applicability such that the user can easily modify them.

**Libraries:**

There are several libraries which are supported by the framework. These include FTP, HTTP, Selenium and Database libraries making it an ideal choice over issues. Also, these libraries are supported through the multiple use cases and various frameworks allowing for better participation as opposed to others. The framework has built in libraries for strings, numbers, and dates.

**Data Driven Test Cases:**

The use of data driven style for testing helps in real time testing of the framework which in return provides detailed implementation on how these processes are related and how they can provide detailed processes in return. Data driven testing has always been preferred since it’s a tweaked version of real time usage which ultimately will help in the processing.

**Reports and Logs:**

Test case suite, execution trace and the results with complete analysis over several parameters is provided in each test. The environment provides execution details which will run, and the execution will help in the rendering of results.

## **Test cases and results:**

Here in this section, we have provided the test cases of each test with the relevant screenshots.

* **Common Reusable functions for Admin Test case:**

Here the below screenshots of AdminResource.Robot file contains all common reusable functions for admin test cases.

* 1. **Test case – 1:**

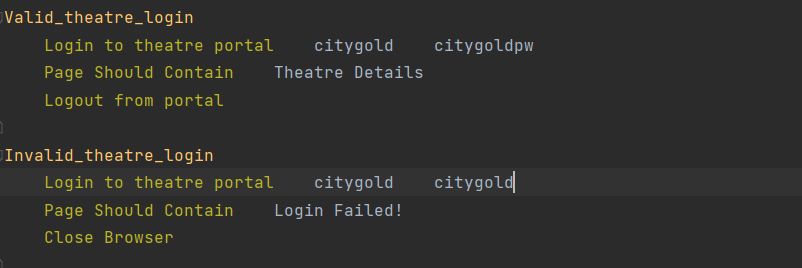
Project Name: Movie Ticket Booking System Module Name: Admin Login Functionality

Created By: Tester Created Date: 13-11-2022

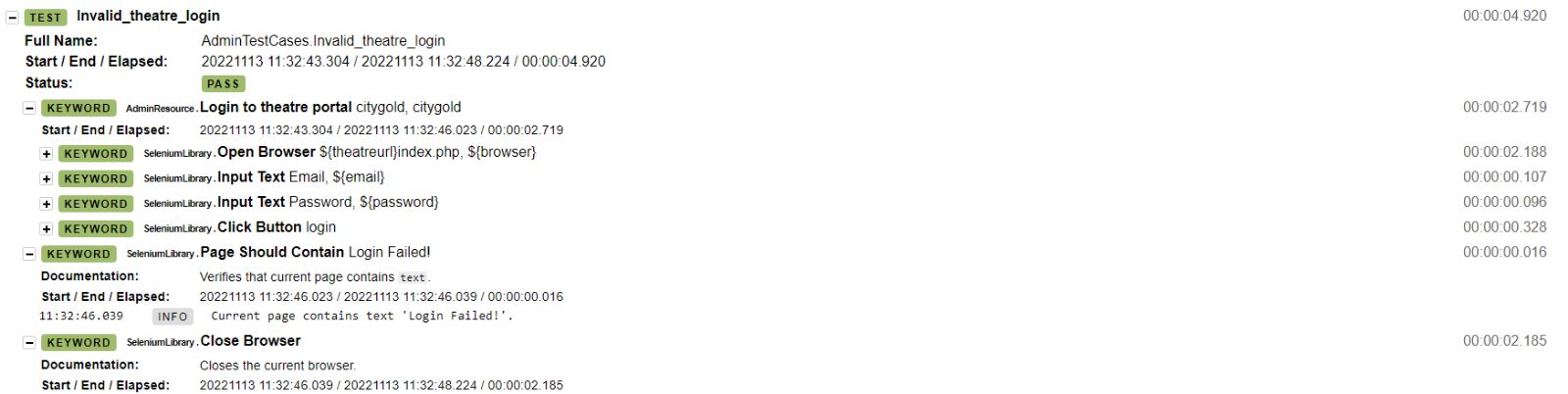
Executed by: Tester Executed Date: 13-11-2022

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Step | Preconditions | Test Data | Expected Result | Actual Result | Status |
| Valid\_theatre\_login | Verify login functionality with valid username and password | Navigate to login page |  |  | Able to see login page | As Expected | Pass |
|  |  | Enter valid username | Valid username | Username: citygold | Credentials can be entered | As Expected | Pass |
|  |  | Enter valid password | Valid password | Password: citygoldpw | Credentials can be entered | As Expected | Pass |
|  |  | Click on login button |  |  | Theatre admin logged | Theatre admin logged successfully | Pass |
| Invalid\_theatre\_login | Verify login functionality with valid username and invalid password | Navigate to login page |  |  | Able to see login page | As Expected | Pass |
|  |  | Enter valid username | Valid username | Username: citygold | Credentials can be entered | As Expected, | Pass |
|  |  | Enter valid password | Invalid password | Password: citygold | Credentials can be entered | As Expected, | Pass |
|  |  | Click on login button |  |  | Theatre admin logged | Theatre admin unsuccessful login | Fail |

**Test Case 1 – Code and result:**







* 1. **Test case – 2:**

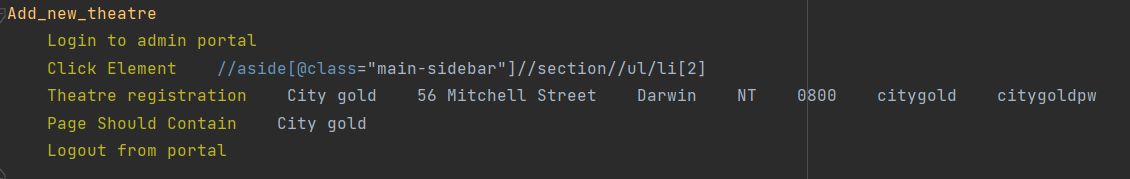
Project Name: Movie Ticket Booking System Module Name: Add new theatre functionality

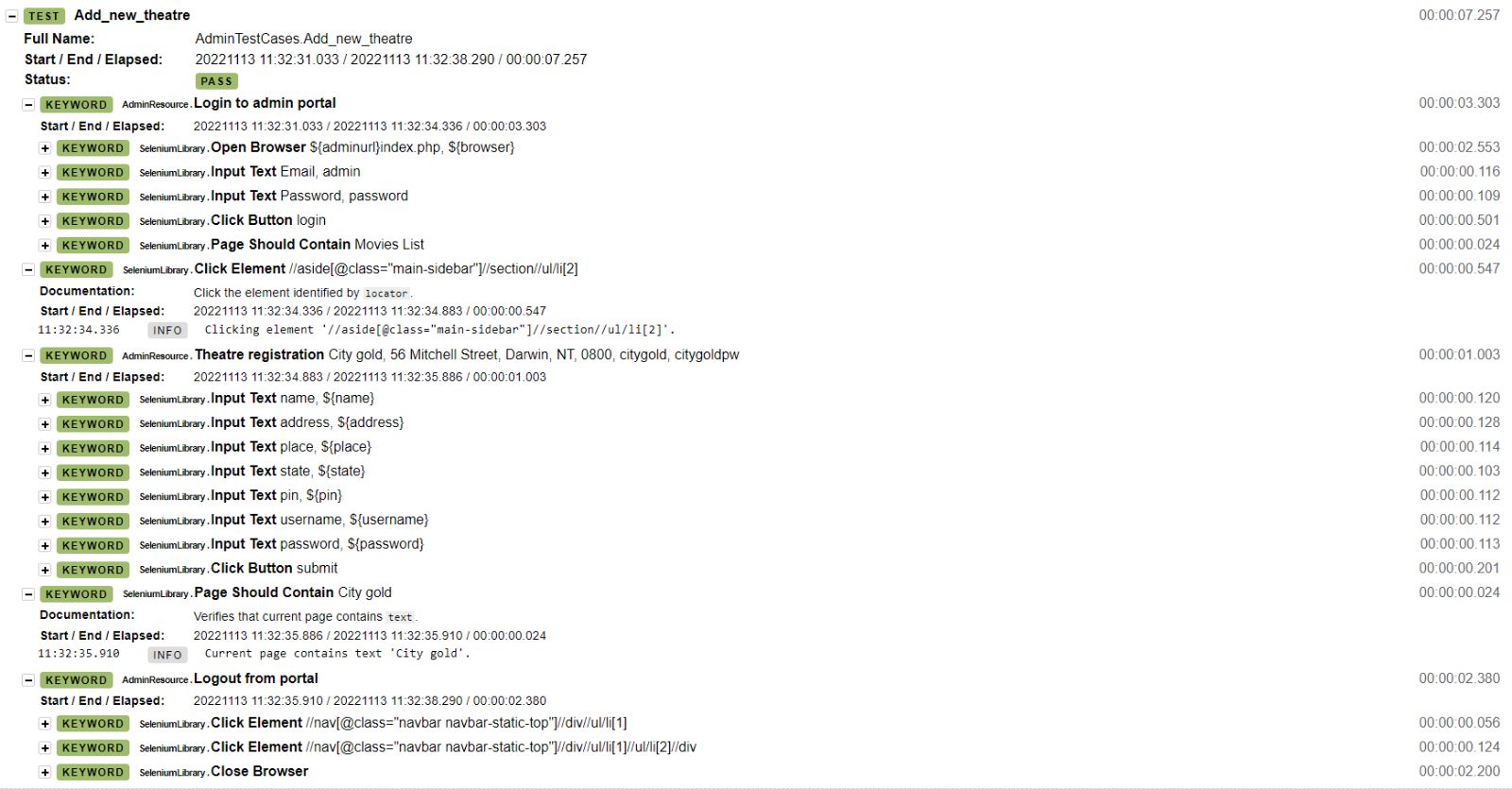
Created By: Tester Created Date: 13-11-2022

Executed by: Tester Executed Date: 13-11-2022

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Step | Preconditions | Test Data | Expected Result | Actual Result | Status |
| Add\_new\_theatre | Add new theatre with valid details | Navigate to add new theatre page |  |  | Able to see new theatre add page | As Expected | Pass |
|  |  | Enter valid details | Enter all the fields | Name=city gold,address=56 mitchell street,place=darwin,state=nt,pin=0800,username=citygold,password=citygoldpw | All the details can be entered | As Expected | Pass |
|  |  | Click on add button |  |  | New theatre added | Theatre admin details entered successfully | Pass |

**Test case – 2 : Code and Result:**





* 1. **Test case – 3:**

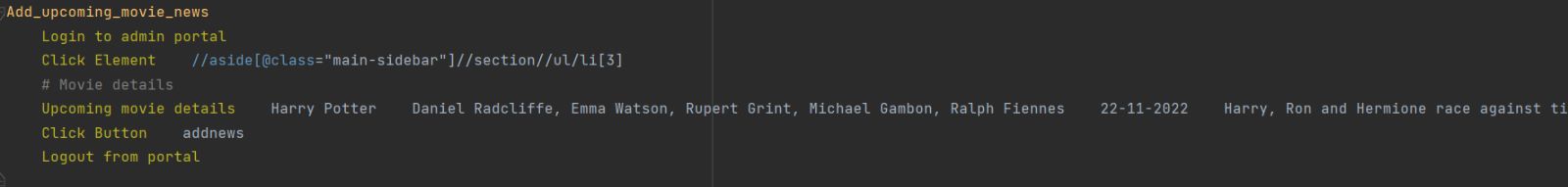
Project Name: Movie Ticket Booking System Module Name: Add upcoming movie news

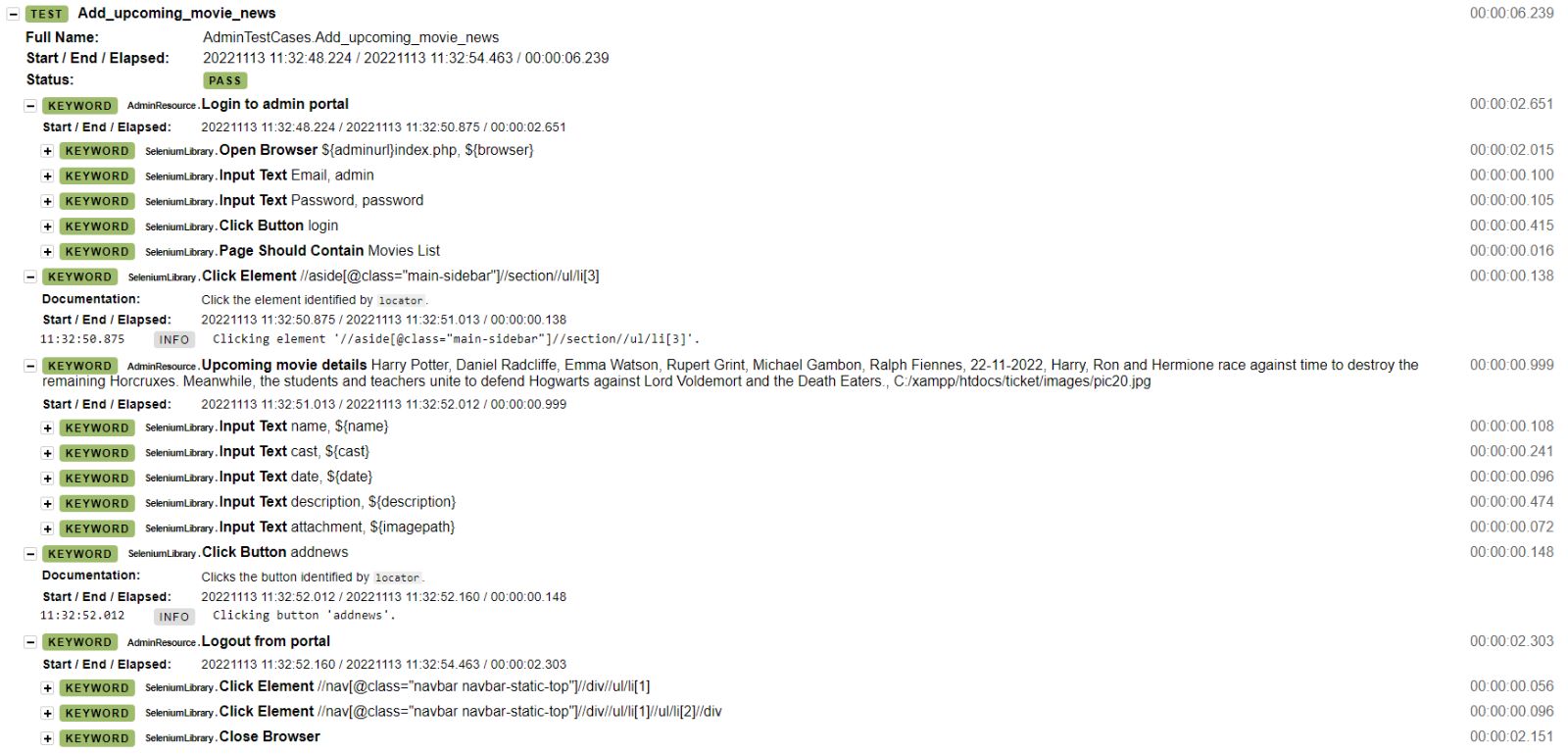
Created By: Tester Created Date: 13-11-2022

Executed by: Tester Executed Date: 13-11-2022

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Step | Preconditions | Test Data | Expected Result | Actual Result | Status |
| Add\_upcoming\_movie\_news | Add upcoming movie news with valid details | Navigate to add upcoming movie news page |  |  | Able to see upcoming movie news add page | As Expected | Pass |
|  |  | Enter valid details | Enter all the fields | Name=Harry Potter ,cast=Daniel Radcliffe, Emma Watson, Rupert Grint, Michael Gambon, Ralph Fiennes ,date=22-11-2022,description=Harry, Ron and Hermione…, Image path = image selection | All the details can be entered | As Expected | Pass |
|  |  | Click on add button |  |  | Upcoming movie news added | Upcoming movie news details entered successfully | Pass |

**Test cases - Code and result:**

****



* 1. **Test case – 4:**

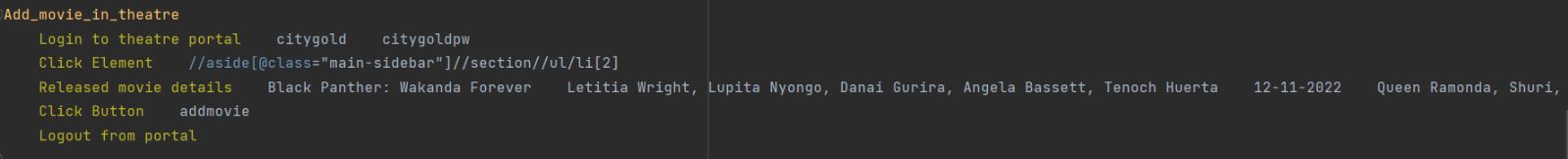
Project Name: Movie Ticket Booking System Module Name: Add movie in theatre

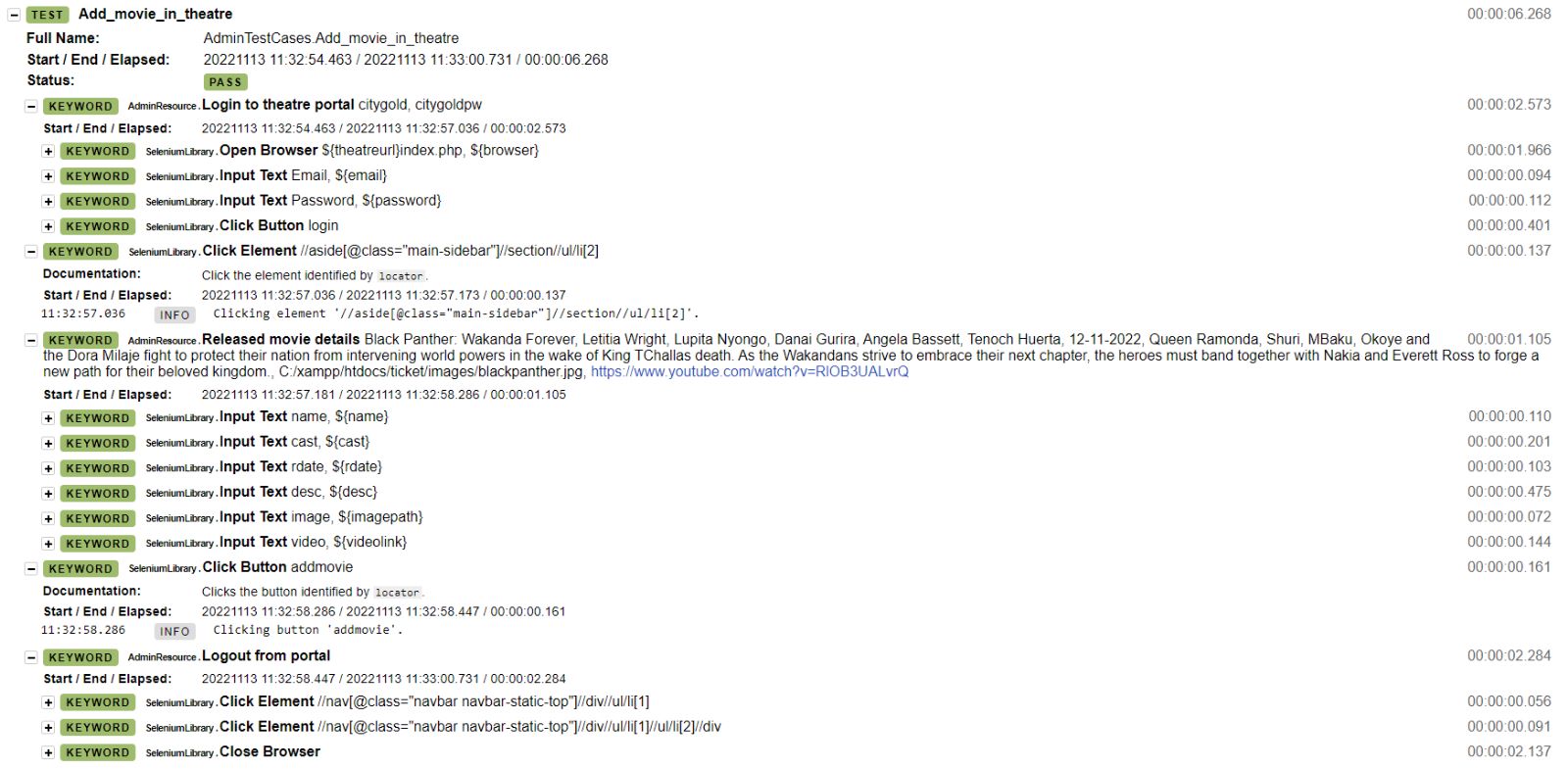
Created By: Tester Created Date: 13-11-2022

Executed by: Tester Executed Date: 13-11-2022

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Step | Preconditions | Test Data | Expected Result | Actual Result | Status |
| Add\_movie\_in\_theatre | Add movie in theatre with valid details | Navigate to add movie page |  |  | Able to see add movie in theatre page | As Expected | Pass |
|  |  | Enter valid details | Enter all the fields | Name=Black Panther : Wakanda Forever ,Cast= Letitia Wright, Lupita Nyongo, Danai Gurira, Angela Bassett..,rdate=12-11-2022,desc=Queen Ramonda…,image=image path…,video=videopath… | All the details can be entered | As Expected | Pass |
|  |  | Click on add button |  |  | Movie is added in theatre | New movie is entered successfully | Pass |

**Test Cases: code and result:**





* 1. **Test case – 5:**

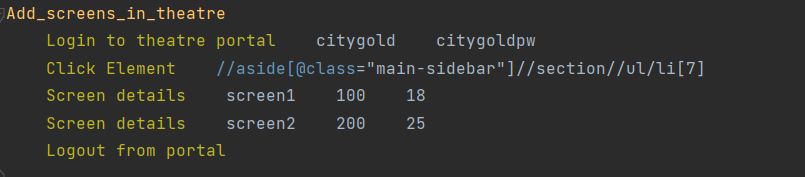
Project Name: Movie Ticket Booking System Module Name: Add screen in theatre

Created By: Tester Created Date: 13-11-2022

Executed by: Tester Executed Date: 13-11-2022

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Step | Preconditions | Test Data | Expected Result | Actual Result | Status |
| Add\_screens\_in\_theatre | Add screen in theatre with valid details | Navigate to add screen in theatre page |  |  | Able to see add screen in theatre page | As Expected | Pass |
|  |  | Enter valid details | Enter all the fields | Name=screen 1, Seats=100, Charge=18 | All the details can be entered | As Expected | Pass |
|  |  | Click on add button |  |  | Screen is added in theatre | New screen is entered successfully | Pass |

**Test cases – code and result:**





* 1. **Test case – 6:**

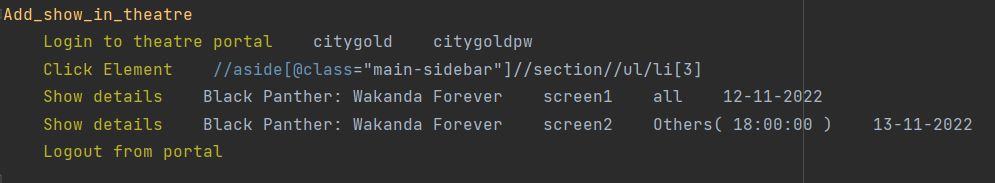
Project Name: Movie Ticket Booking System Module Name: Add show in theatre

Created By: Tester Created Date: 13-11-2022

Executed by: Tester Executed Date: 13-11-2022

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Step | Preconditions | Test Data | Expected Result | Actual Result | Status |
| Add\_show\_in\_theatre | Add show in theatre with valid details | Navigate to add show in theatre page |  |  | Able to see add show in theatre page | As Expected | Pass |
|  |  | Enter valid details | Enter all the fields | Movie= Black Panther, Screen= Screen1, Date= 12-11-2022 | All the details can be entered | As Expected | Pass |
|  |  | Click on add button |  |  | Show is added in theatre | Show is entered successfully | Pass |

**Test case – code and result:**



Graphical user interface, text, application, email

Description automatically generated

* 1. **Test case – 7:**

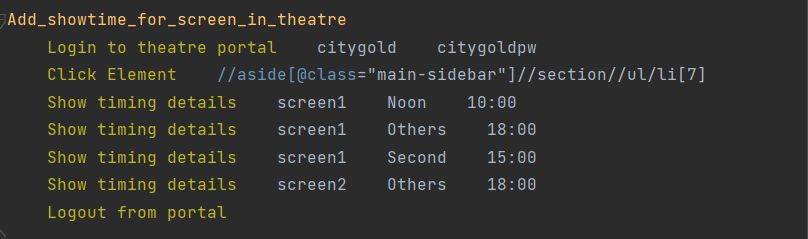
Project Name: Movie Ticket Booking System Module Name: Add showtime for screen

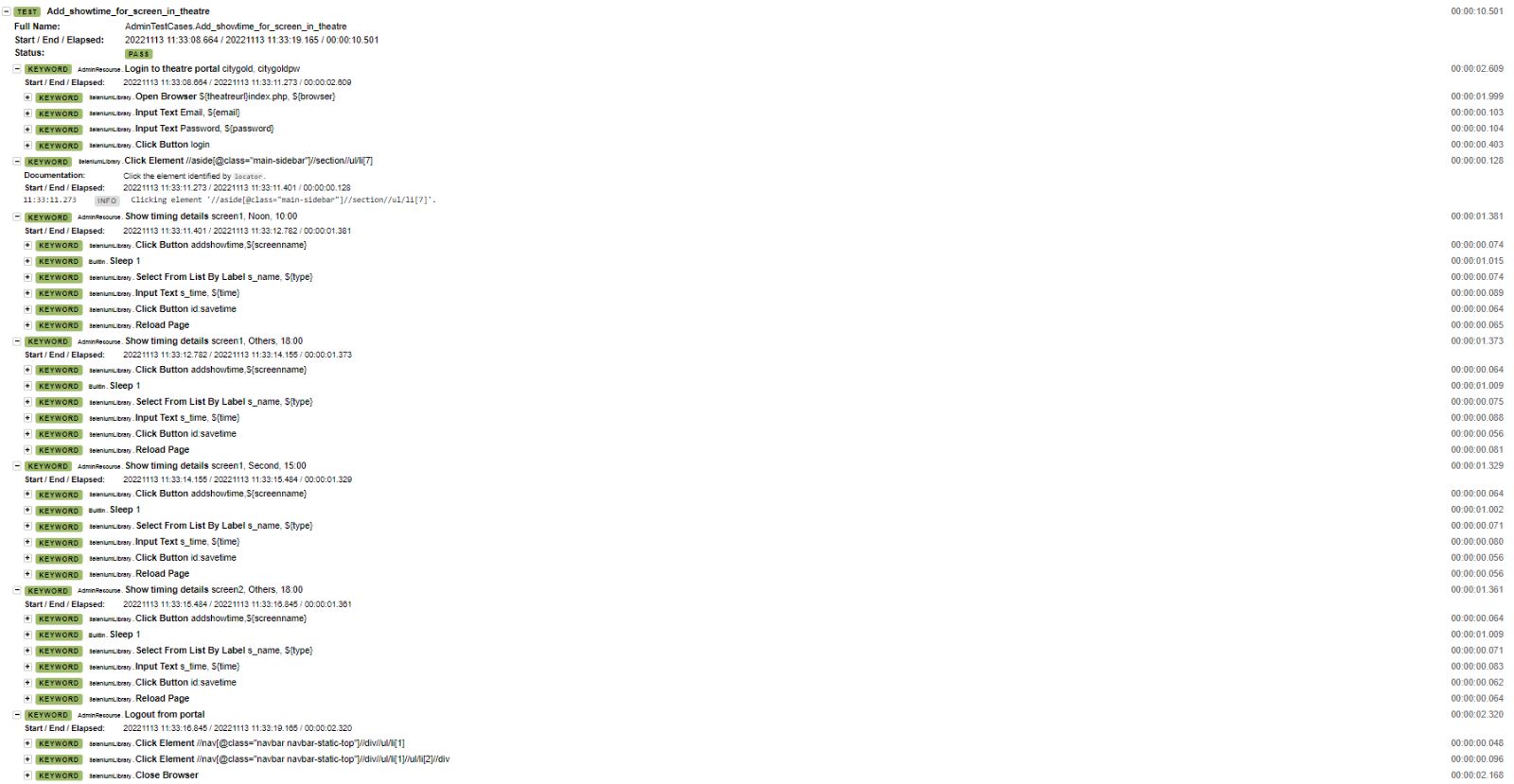
Created By: Tester Created Date: 13-11-2022

Executed by: Tester Executed Date: 13-11-2022

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Step | Preconditions | Test Data | Expected Result | Actual Result | Status |
| Add\_showtime\_for\_screen\_in\_theatre | Add showtime for screen in theatre with valid details | Navigate to add showtime for screen in theatre page |  |  | Able to see add showtime for screen in theatre page | As Expected | Pass |
|  |  | Enter valid details | Enter all the fields | S\_name=screen1,Dropdownmenu=select screen type,S\_time=10:00 | All the details can be entered | As Expected | Pass |
|  |  | Click on add button |  |  | Show time is added in theatre | Show time for screen is entered successfully | Pass |

**Test case – code and result:**





* 1. **Test case – 8:**

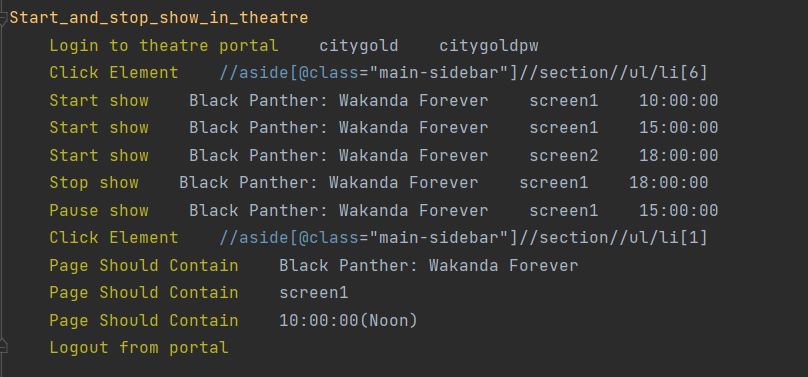
Project Name: Movie Ticket Booking System Module Name: Start and stop show time in theatre

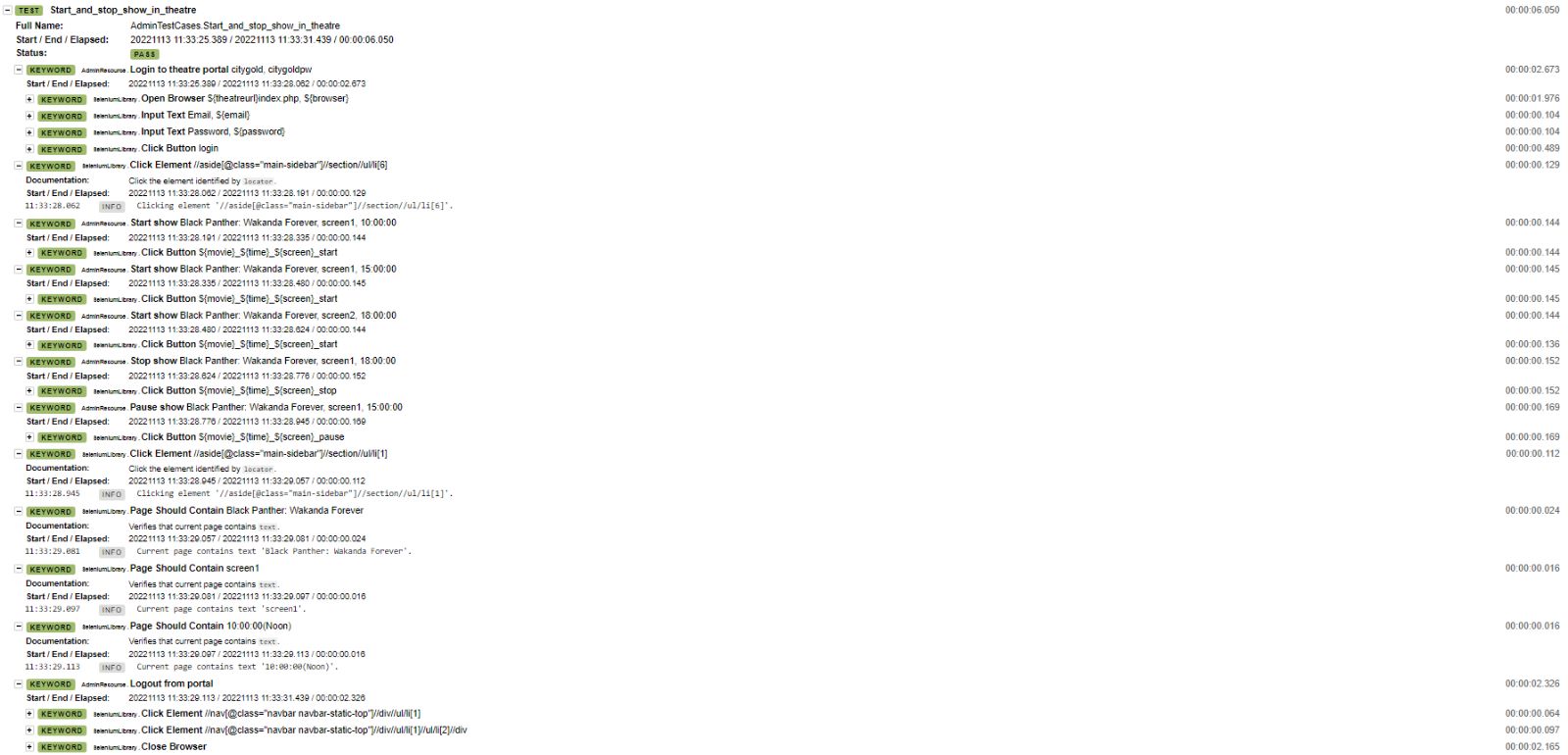
Created By: Tester Created Date: 13-11-2022

Executed by: Tester Executed Date: 13-11-2022

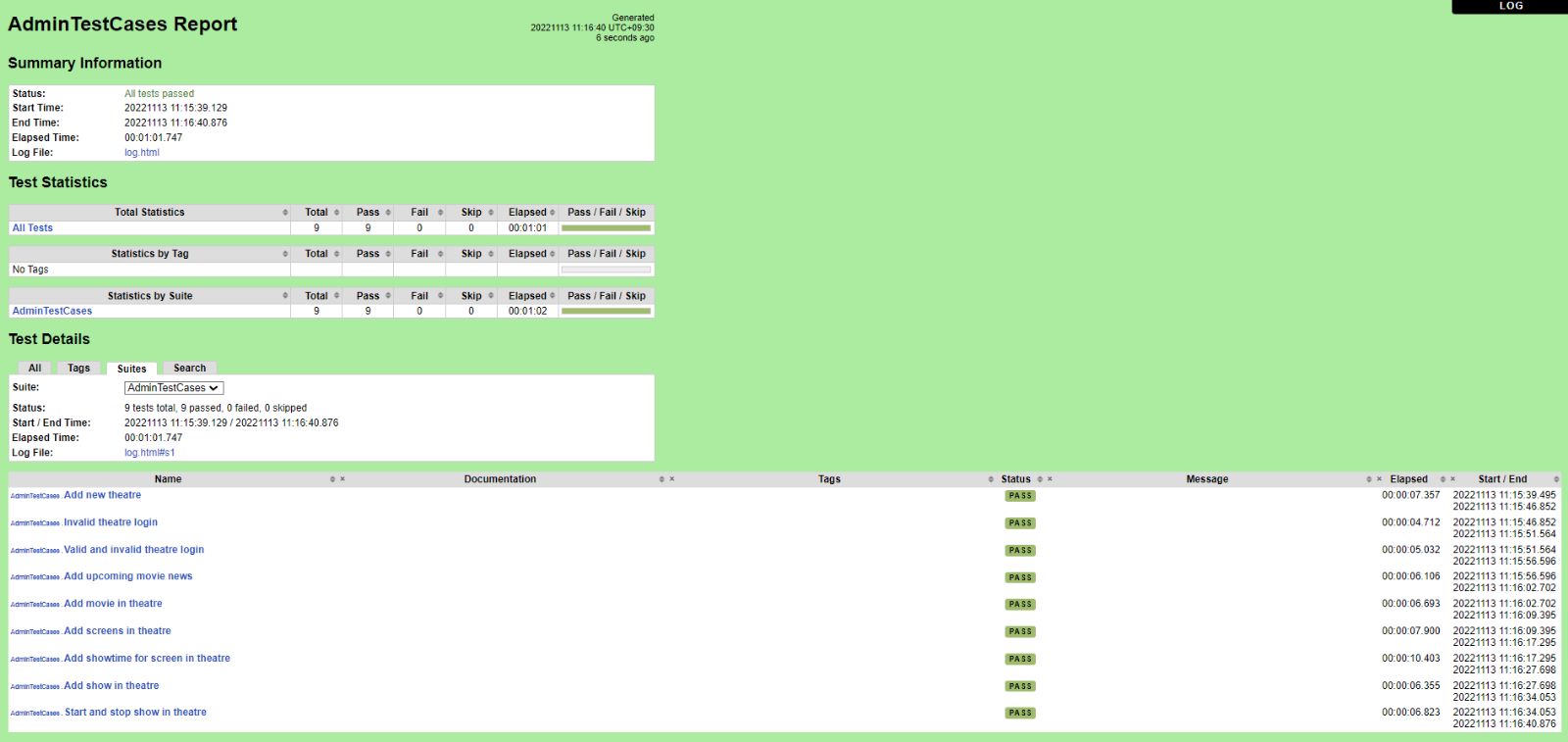
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Step | Preconditions | Test Data | Expected Result | Actual Result | Status |
| Start\_and\_stop\_show\_in\_theatre | Start and show the shows by clicking the button | Navigate to start and stop show page in theatre |  |  | Able to see start and stop show page in theatre | As Expected | Pass |
|  |  | Click on start show button | Click on the button | Click on the button | Valid click | Show is started | Pass |
|  |  | Click on stop show button | Click on the button | Click on the button | Valid click | Show is stopped | Pass |

**Test case – code and result:**



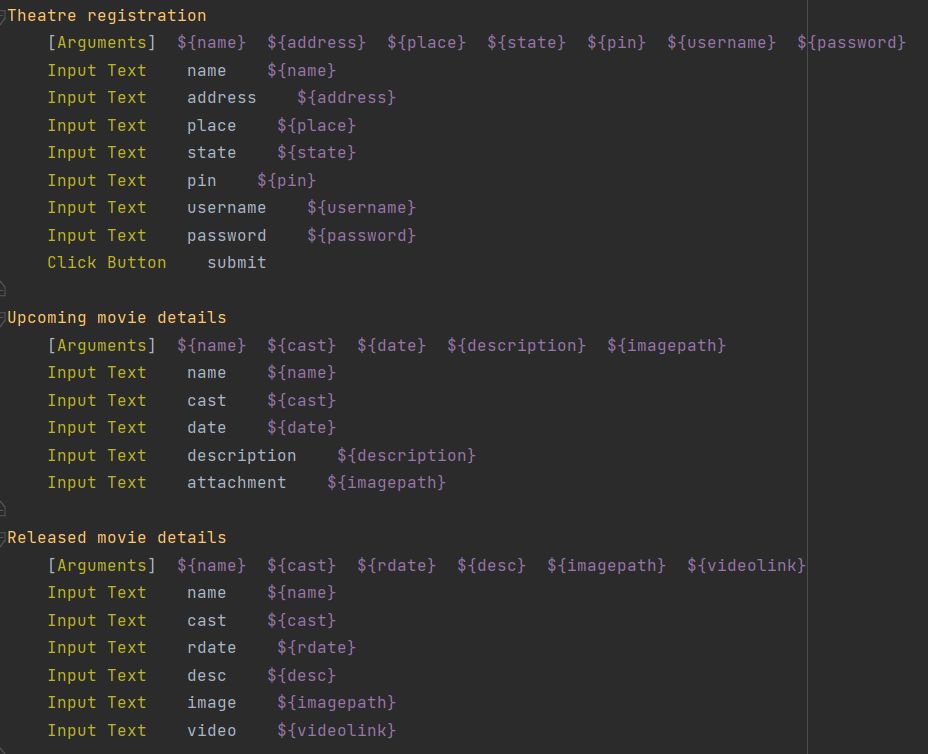


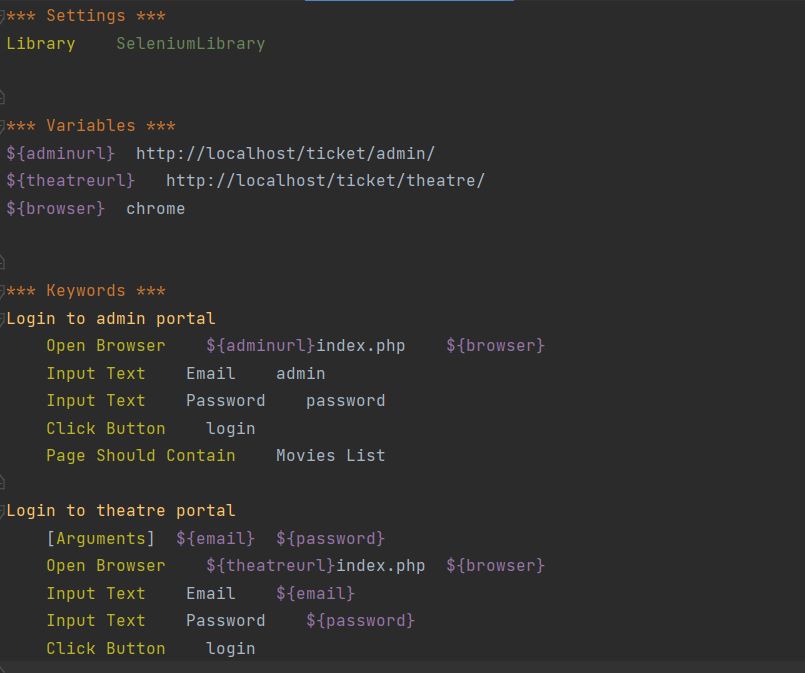
* **All Admin Test Case Report:**



* **Common Reusable Functions for Users**

Here the below screenshots are from UserResource.robot file. It contains all the reusable functions for user test cases.





* 1. **Test case – 9:**

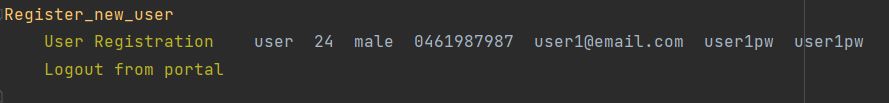
Project Name: Movie Ticket Booking System Module Name: Register new user

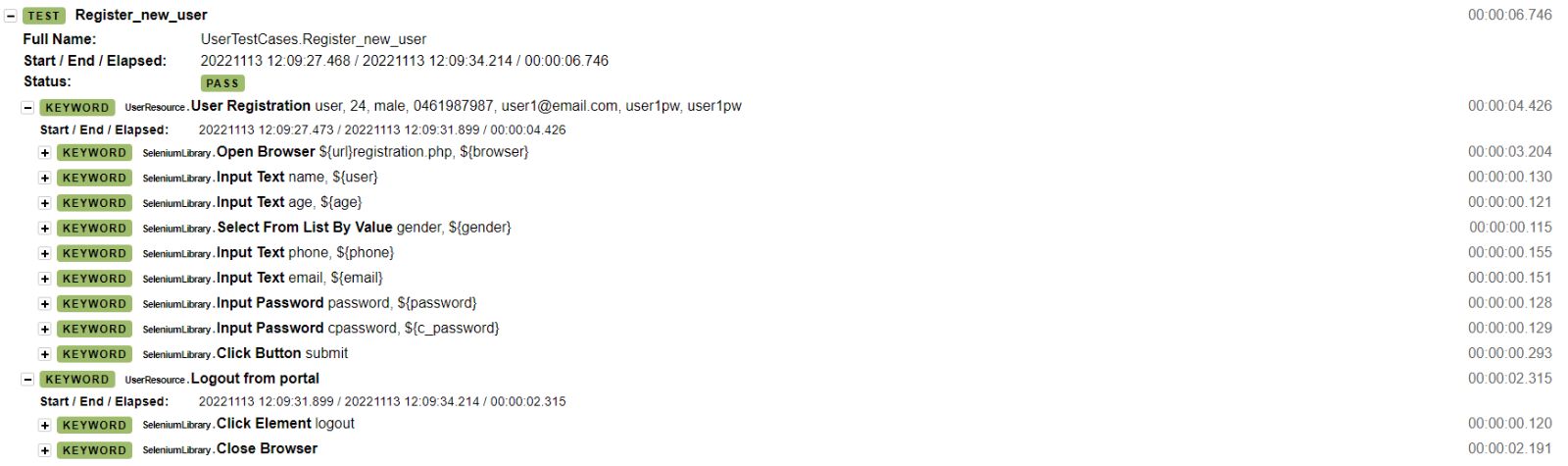
Created By: Tester Created Date: 13-11-2022

Executed by: Tester Executed Date: 13-11-2022

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Step | Preconditions | Test Data | Expected Result | Actual Result | Status |
| Register\_new\_user | Register new user by clicking the button | Navigate to register new user page |  |  | Able to see register new user page | As Expected | Pass |
|  |  | Enter valid details | Enter all the fields | Name=user,age=24,gender=male,phone=0461987987,email=user1@email.com,password=user1pw,cpassword=user1pw | All the details can be entered | As Expected | Pass |
|  |  | Click on register button |  |  | User is registered | New user is registered successfully | Pass |

**Test case – code and Result:**





* 1. **Test case – 10:**

Project Name: Movie Ticket Booking System Module Name: User Login Functionality

Created By: Tester Created Date: 13-11-2022

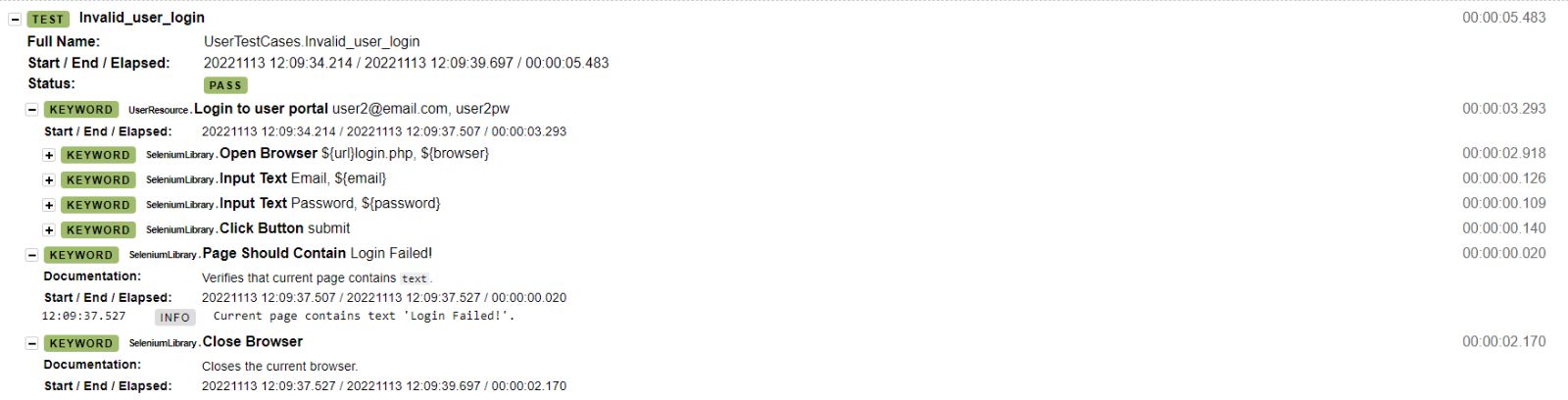
Executed by: Tester Executed Date: 13-11-2022

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Step | Preconditions | Test Data | Expected Result | Actual Result | Status |
| Valid\_user\_login | Verify login functionality with valid username and password | Navigate to login page |  |  | Able to see login page | As Expected | Pass |
|  |  | Enter valid username | Valid username | Username: user1@email.com | Credentials can be entered | As Expected | Pass |
|  |  | Enter valid password | Valid password | Password: user1pw | Credentials can be entered | As Expected | Pass |
|  |  | Click on login button |  |  | User logged | User logged successfully | Pass |
| Valid\_and\_invalid\_user\_login | Verify login functionality with valid username and invalid password | Navigate to login page |  |  | Able to see login page | As Expected | Pass |
|  |  | Enter valid username | Valid username | Username: user2@email.com | Credentials can be entered | As Expected | Pass |
|  |  | Enter valid password | Invalid password | Password: user2pw | Credentials can be entered | As Expected | Pass |
|  |  | Click on login button |  |  | User logged | User unsuccessful login | Fail |

**Test case – code and result:**

****





* 1. **Test case – 11:**

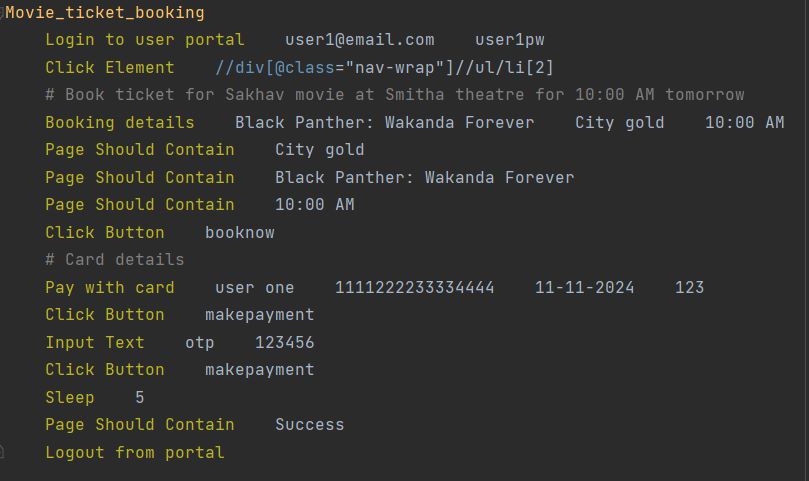
Project Name: Movie Ticket Booking System Module Name: Movie Ticket Booking

Created By: Tester Created Date: 13-11-2022

Executed by: Tester Executed Date: 13-11-2022

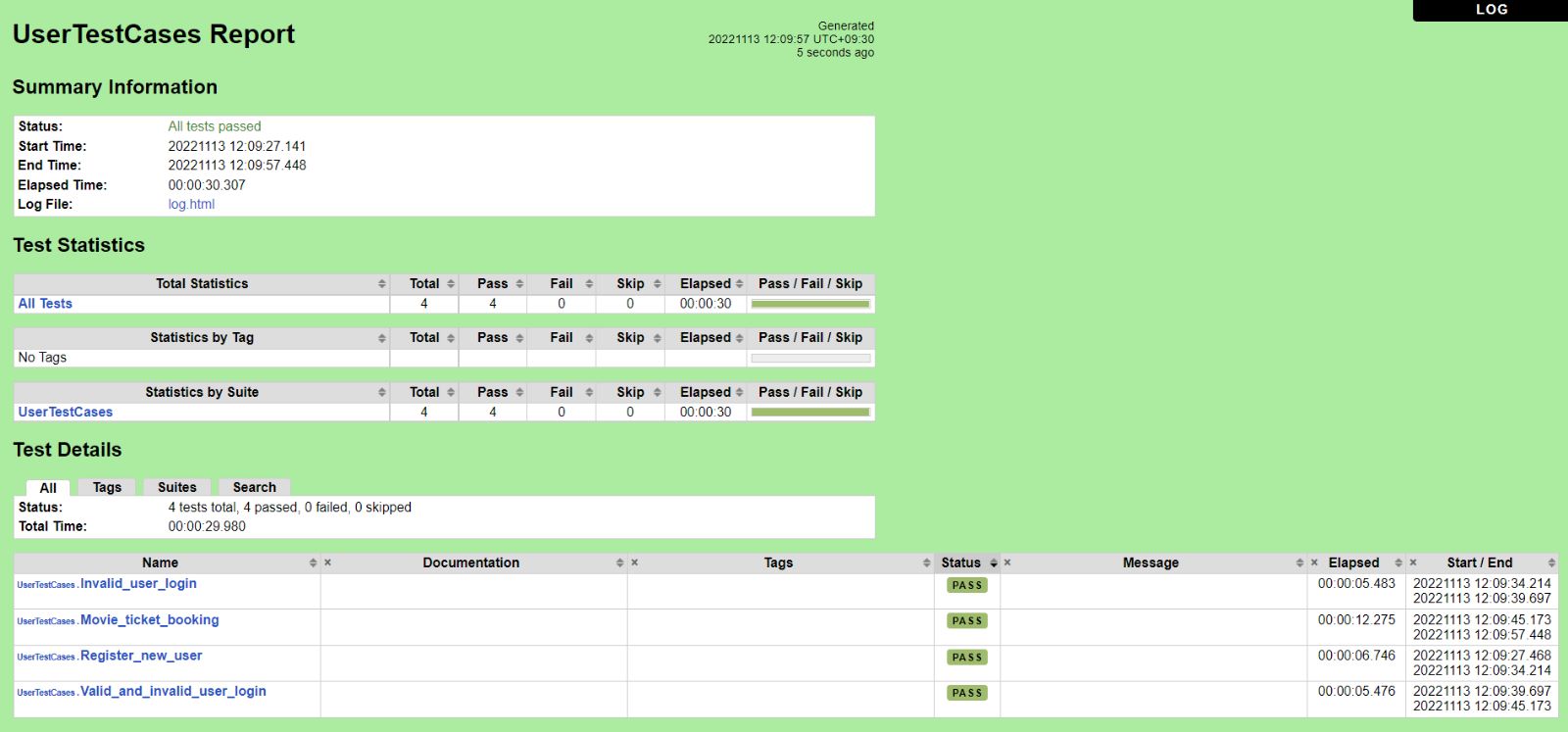
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Test Step | Preconditions | Test Data | Expected Result | Actual Result | Status |
| Movie\_ticket\_booking | Book a ticket by clicking the button | Navigate to ticket booking page |  |  | Able to see ticket booking page | As Expected | Pass |
|  |  | Select and enter valid details | Select and enter all the fields | Select Movie, Select Time, Theater and Movie, Card Name=user one ,Card Number=1111222233334444,Card Date=11-11-2024, Card CVV=123,OTP=123456 | All the details can be entered | As Expected | Pass |
|  |  | Click on ticket book button |  |  | Ticket is booked | Ticket is booked successfully | Pass |

**Test case – code and result:**

****



* **All User Test Cases Report:**

****

# 

# **Conclusion:**

## **Ethical Consideration:**

Following main concepts were taken into regards while the whole testing process i.e., Functional as well as non-functional was formalized. It was prepared with the process to be fair and straight with unbiased results as the core principle.

* **Fairness of Test Scenario:**

The testing procedure was equally applied as per the standard practices and no unfair practices were utilized. The best of the knowledge of tester was applied as per required perspective. Similarly, there were several processes which were represented, and it was indeed shown how the user could use the system, what mistakes can be made however extreme limits were balanced to lower chances of degrading the testing.

* **Utility of Tests for Evaluating Education:**

The tests were made with data and demographics in user registration as fictional. None of the data was real but it was profiled according to the users. So, the data can be utilized, and the results are available for usage without any issue.

## **Future Works:**

Although the current testing, functional and nonfunctional was completed with extensive digging however there were several processes which might have got overlooked. This includes the proper profiling of the users who utilize the system and what exceptions they can make which could lead to the system malfunctioning. In the future work, the system will be compelled wo work such that the user will be able to implement them in the required status.

## **Lesson Learnt:**

Robot framework enlightened the automated testing with much of the issues that could be resolved in the process. it is necessary to include them in the process. it is also applicable to have them applied as needed by the user. It is also relevant to the process and the user could also have the applicability of improving the system and the user experience. The crux of the working is the idea of user centered design and user experience. Both are extremely viable and should be applied at the available moment.

## **GitHub Links:**

Open-source web application for movie ticket booking:

<https://github.com/projectworldsofficial/Online-Movie-Ticket-Booking-System-in-php>

Our project testing:

<https://github.com/sh-13/movie-ticket-booking-testing-robot>