A

MINI PROJECT

REPORT ON

Movie Ticket Booking System

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE.
FOR
LAB PRACTICE II

Unit Testing with Unittest

BACHELOR OF ENGINEERING (COMPUTER ENGINEERING)
SUBMITTED BY

Name: Prajakta kamble	PRN – 72000288F
Name: Rutuja Bachhav	PRN – 72000282G
Name: Sayama Kazi	PRN – 7200089D

DEPARTMENT OF COMPUTER ENGINEERING
D.Y.PATIL COLLEGE OF ENGINEERING AKURDI, PUNE-44.
SAVITRIBAI PHULE PUNE UNIVERSITY, 2021-22 SEM-I

INDEX

PROJECT DETAILS

Project Title

Unit testing on Movie Ticket Booking System using python selenium.

Problem Statement

To test the different test cases of Movie ticket booking system using python selenium.

ABSTRACT

The main purpose to of online booking system is to provide another way for customer to buy movie tickets. Online movie ticket booking system is to book tickets for movie it will be asked to data from the users and checking the all test cases. We using the python selenium with the unit testing for test all test cases regarding to different modules like username, movie name, time etc. To achieve this goal this project work we will do Unit testing. Unit Testing is a software testing technique by means of which individual units of software i.e. group of computer program modules, usage procedures and operating procedures are tested to determine whether they are suitable for use or not. WE using Selenium with Python is used to carry out automated test cases for browsers or web applications. You can easily use it to simulate tests such as tapping on a button, entering content to the structures, skimming the entire site, etc.

INTRODUCTION

The Project "Online Movie Ticket Booking System" as a wide scope as it is generalized software and can be easily used in any ticket booking process system with little or no change. The Changes in software can be easily accommodated. The addition and deletion of the modules in software can be easily adjusted. In Movie booking system we using python selenium with unit testing to test different test cases of different modules into the system like if username is correct it will show all test cases are passed, We test the all modules test cases using unit testing. When customer enters their data to the system then system will check all entered data will correct or not and check whether all test cases are passed if all test cases are passed it will show all test cases are passed otherwise no test cases passed.

SOFTWARE /HARDWARE REQUIREMENTS

- **Software Requirement Specifications:**

  For development:

  Pycharm IDE

  For Testing:

  UnitTest:

- **Hardware Requirement Specification**

  Intel Pentium Processor

  32 MB RAM or higher

  1.2 GB Hard Disk or greater

SYSTEM ARCHITECTURE

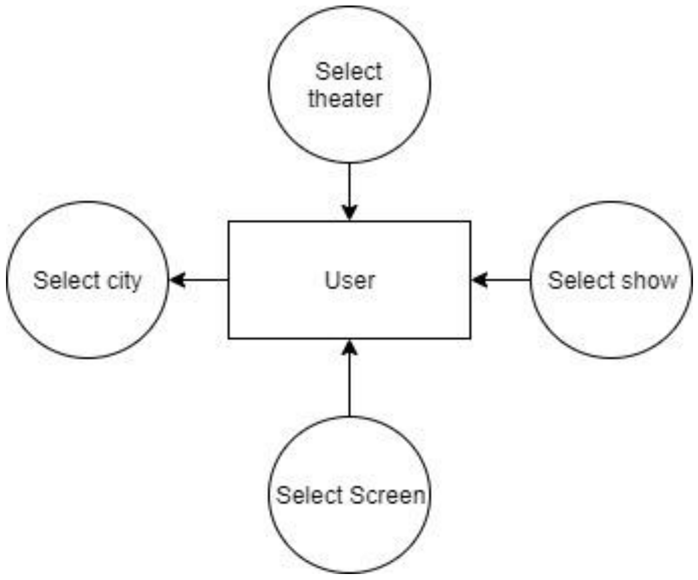


FIG.1 Movies Ticket Booking System

TEST PLAN

Sr. No.	Test class	Description	Expected Result	Actual Result
1	Admin login	Ensures Admin login successfully	Page should open and Admin Login should be done successfully	Page open and Login done successfully

TESTING STRATEGY

Unit Test: - Unit testing is white box testing. Testing is performed by Developer.

Module: - User Login

Test case Id	**Test case objective**	**Steps**	**Input**	**Expected Result**	**Actual Result**	**Status**
TC-1	Check for the Admin Username field	Click on Admin username field and enter valid customer username	Prajakta	Admin username field should accept only valid Admin username which available in database	Admin username field accept only valid Admin username which available in database	Pass
TC - 2	Check for the admin Password field	Click on admin Password field and enter admin valid password	Rutuja	Admin password field should accept only valid admin password which available in database	Admin password field accept only valid admin password which available in database	Pass
TC - 3	Check for city	Enter choice	1	Its submitted successfully	It’s done successfully	Pass
TC - 4	Check Tether	Enter choice	2	It should be submitted successfully	Done successfully	Pass

TC - 5	Check Movies	Enter choice	1	It should be accept choice	Done successfully	Pass
TC - 6	Check Screen	Enter choice	2	It should be accept choice	Done successfully	Pass

TECHNICAL DETAILS

Unit Testing

Unit testing is one of the software testing types which includes the initial testing phase where the smallest components or the modules of a software are tested individually. With this method of testing, both testers and developers can isolate each module, identify and fix the system defects at a very early stage of the software development lifecycle (SDLC).A typical unit test consists of three phases which include the first initialization phase where it initializes a small piece of an application it wants to test. The second phase is the addition phase where it adds a stimulus to the system under test and finally, the third phase is the result phase where it observes the resulting application behavior. Evidently, if the observed behavior is consistent with expectations, then the unit test passes else it fails. This indicates there is a problem somewhere in the system under test.

Integration Testing

Integration testing is a systematic technique or construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. Scope of testing summarizes the specific functional, performance, and internal design characteristics that are to be tested. It employs top-down testing and bottom-up testing methods for this case Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.Unit testing uses modules for testing purpose, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules

The workflow of unit testing is performed in 4 stages:

1. Creating test cases

2. Reviewing test cases

3. Baselining test cases

4. Executing test cases

Unit test framework

The Python unit testing framework, sometimes referred to as "PyUnit," is a Python language version of JUnit developed by Kent Beck and Erich Gamma. PyUnit forms part of the Python Standard Library as of Python version 2.1.

Python unit testing framework supports test automation, sharing of setup and shutdown code for tests, aggregation of tests into collections, and independence of the tests from the reporting framework. The unittest module provides classes that make it easy to support these qualities for a set of tests. As the name indicates, Unit testing is a software testing method meant for testing small units of code. These are typically small, automated pieces of code written by software developers to check and ensure that the particular piece of code only behaves the way it was intended to. In Python, there are several frameworks available for this purpose and here we will discuss the major "python test automation frameworks

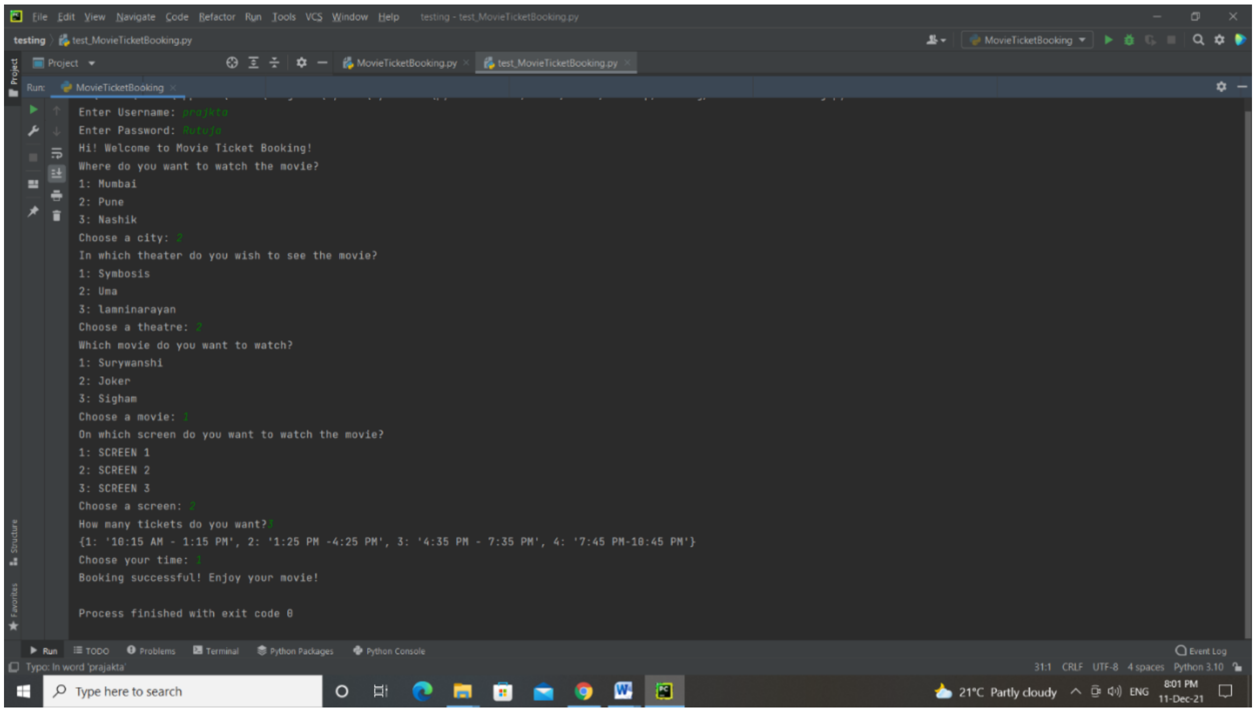
SYSTEM CODE

```
def username():
    a = input("Enter Username: ")
    return a

def password():
    a = input("Enter Password: ")
    return a
# This function is used to select the city
def city():
    print("Hi! Welcome to Movie Ticket Booking!")
    print("Where do you want to watch the movie?")
    print("1: Mumbai")
    print("2: Pune")
    print("3: Nashik")
    a = int(input("Choose a city: "))
    return a

# This function is used to select the theatre
def theatre():
    print("In which theater do you wish to see the movie?")
    print("1: Symbosis")
    print("2: Uma")
    print("3: lamninarayan")
    a = int(input("Choose a theatre: "))
    return a

# This function is used to select the movie
def movie():
    print("Which movie do you want to watch?")
    print("1: Surywanshi")
    print("2: Joker")
    print("3: Sigham")
    a = int(input("Choose a movie: "))
    return a

# This function is used to select the screen
def screen():
    print("On which screen do you want to watch the movie?")
    print("1: SCREEN 1")
    print("2: SCREEN 2")
    print("3: SCREEN 3")
    a = int(input("Choose a screen: "))
    b = int(input("How many tickets do you want?" ))
    return a

# This function is used to select the timing
def timing(a):
    time1 = {
        1: "10:00 AM - 1:00 PM",
        2: "1:10 PM - 4:10 PM",
        3: "4:20 PM - 7:20 PM",
        4: "7:30 PM - 10:30 PM"
    }
    time2 = {
        1: "10:15 AM - 1:15 PM",
        2: "1:25 PM -4:25 PM",
        3: "4:35 PM - 7:35 PM",
        4: "7:45 PM-10:45 PM"
    }
    time3 = {
        1: "10:30 AM - 1:30 PM",
        2: "1:40 PM - 4:40 PM",
        3: "4:50 PM - 7:50 PM",
        4: "8:00 PM - 10:45 PM"
    }
    if a == 1:
        print(time1)
        t = int(input("Choose your time: "))
        x = time1[t]
        return t
    elif a == 2:
        print(time2)
        t = int(input("Choose your time: "))
        x = time2[t]
        return t
    elif a == 3:
        print(time3)
        t = int(input("Choose your time: "))
        x = time3[t]
        return t

if __name__ == '__main__':
    username()
    password()
    city()
    theatre()
    movie()
    s=screen()
    tm=timing(s)
    print("Booking successful! Enjoy your movie!")
```

TESTING CODE

```
import unittest
import MovieTicketBooking

class TestBooking(unittest.TestCase):
    def test_username(self):
        username = "prajakta"
        result = MovieTicketBooking.username()
        self.assertEqual(username, result, msg='Incorrect Password!')

    def test_password(self):
        password = "Rutuja"
        result = MovieTicketBooking.password()
        self.assertEqual(password, result, msg='Incorrect Password!')

    def test_city(self):
        result = MovieTicketBooking.city()
        self.assertTrue(0<result<4, msg='Incorrect City Choice!')

    def test_theatre(self):
        result = MovieTicketBooking.theatre()
        self.assertTrue(0<result<4, msg='Incorrect Theatre Choice!')

    def test_movie(self):
        result = MovieTicketBooking.movie()
        self.assertTrue(0<result<4, msg='Incorrect Movie Choice!')

    def test_screen(self):
        result = MovieTicketBooking.screen()
        self.assertTrue(0<result<4, msg='Incorrect Screen Choice!')



if __name__ == '__main__':
        unittest.main()
```

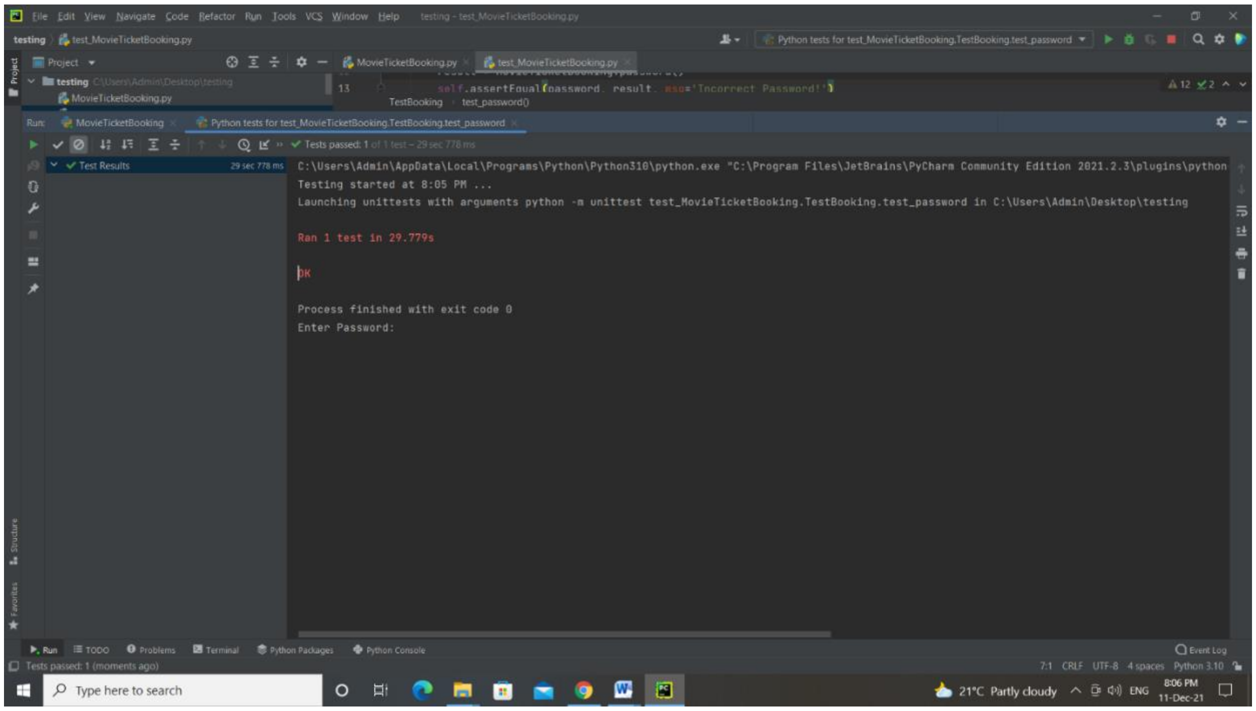
RESULT



```
Enter Username: prajkta
Enter Password: Rutuja
Hi! Welcome to Movie Ticket Booking!
Where do you want to watch the movie?
1: Mumbai
2: Pune
3: Nashik
Choose a city: 2
In which theater do you wish to see the movie?
1: Symbosis
2: Uma
3: lamninarayan
Choose a theatre: 2
Which movie do you want to watch?
1: Surywanshi
2: Joker
3: Sigham
Choose a movie: 1
On which screen do you want to watch the movie?
1: SCREEN 1
2: SCREEN 2
3: SCREEN 3
Choose a screen: 2
How many tickets do you want?3
{1: '10:15 AM - 1:15 PM', 2: '1:25 PM -4:25 PM', 3: '4:35 PM - 7:35 PM', 4: '7:45 PM-10:45 PM'}
Choose your time: 1
Booking successful! Enjoy your movie!

Process finished with exit code 0
```

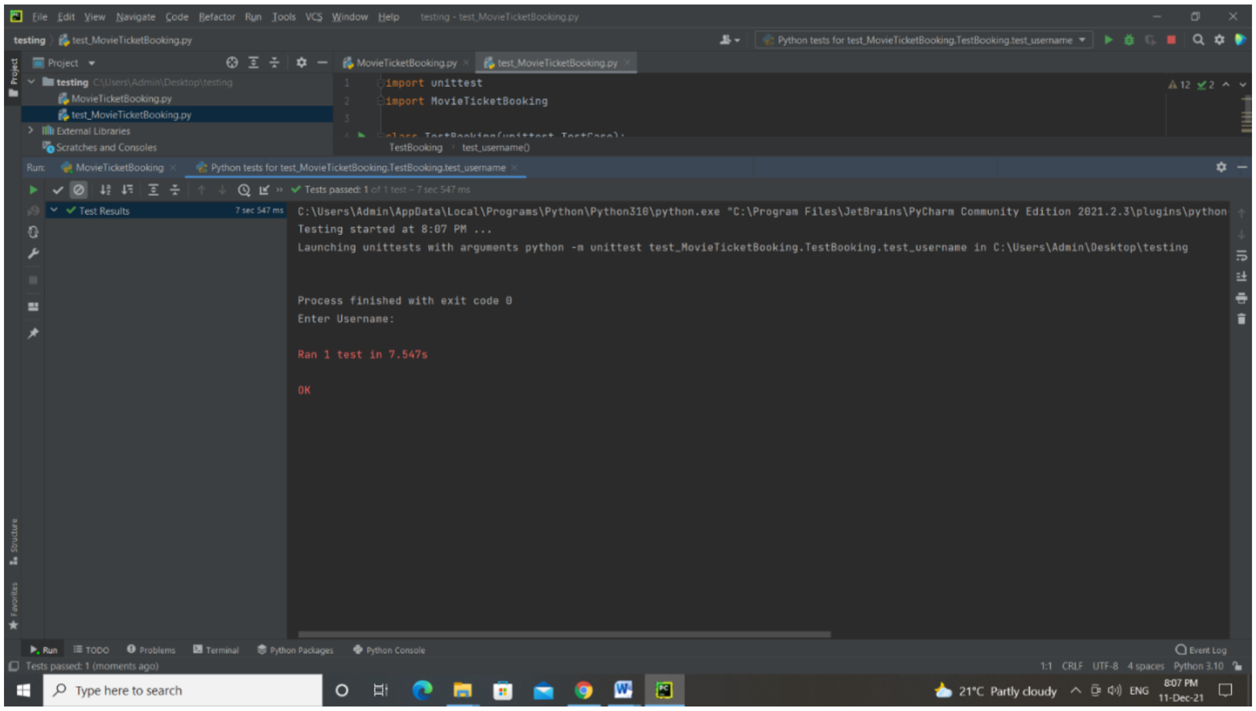
File Edit View Navigate Code Refactor Run Tools VCS Window Help
testing - test_MovieTicketBooking.py
MovieTicketBooking.py
test_MovieTicketBooking.py
Run: MovieTicketBooking
Typo: In word 'prajakta'

TESTING



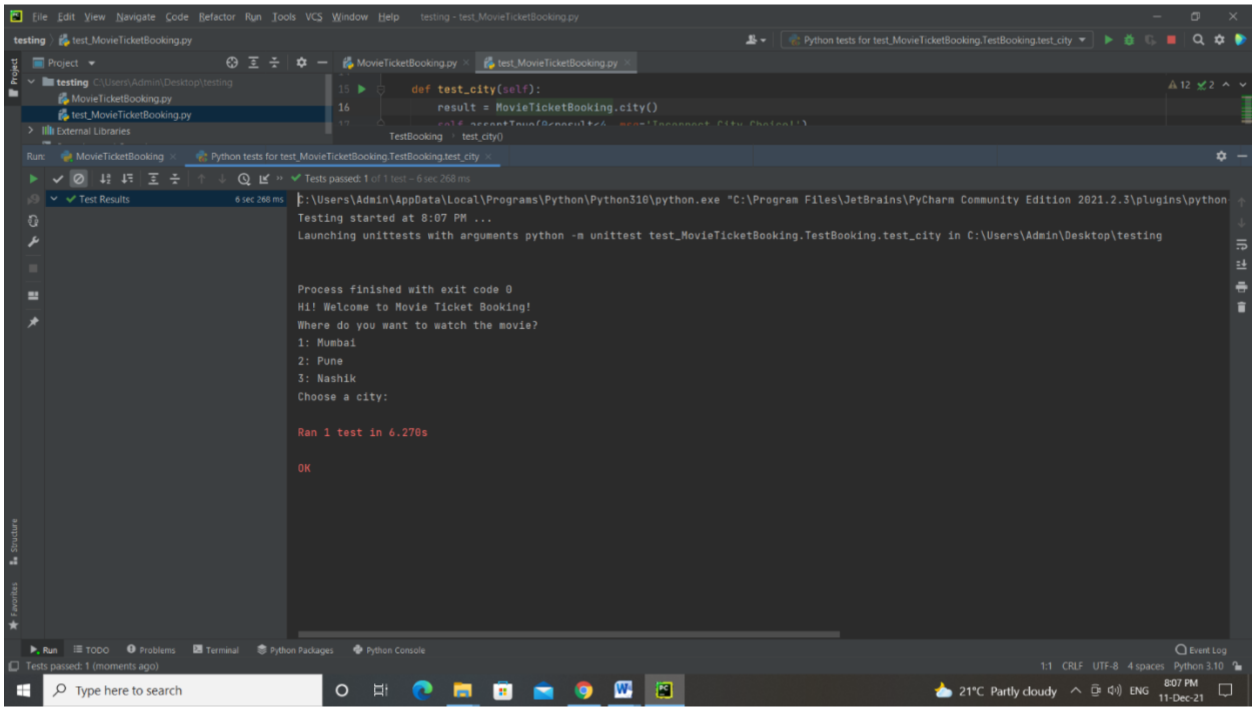
```
self.assertEqual(password, result, msg='Incorrect Password!')
```

```
C:\Users\Admin\AppData\Local\Programs\Python\Python310\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.3\plugins\python
Testing started at 8:05 PM ...
Launching unittests with arguments python -m unittest test_MovieTicketBooking.TestBooking.test_password in C:\Users\Admin\Desktop\testing

Ran 1 test in 29.779s

OK

Process finished with exit code 0
Enter Password:
```



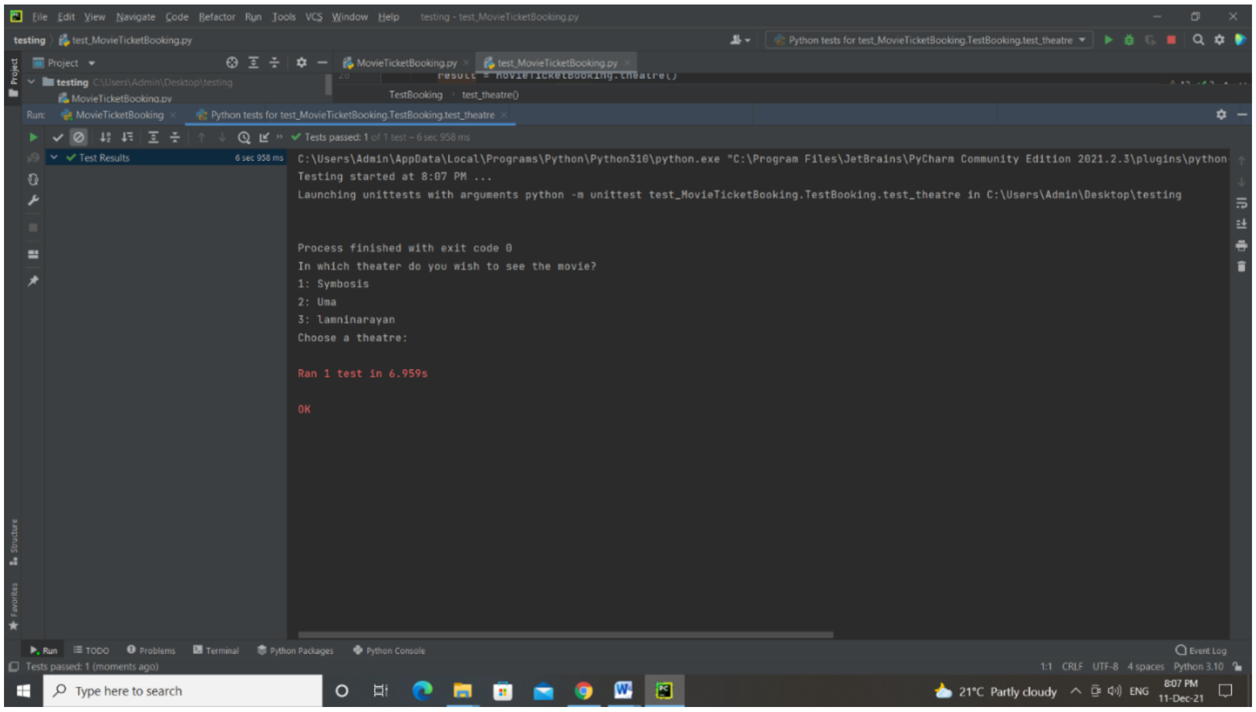
```
import unittest
import MovieTicketBooking

class TestBooking(unittest.TestCase):
```

```
C:\Users\Admin\AppData\Local\Programs\Python\Python310\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.3\plugins\python
Testing started at 8:07 PM ...
Launching unittests with arguments python -m unittest test_MovieTicketBooking.TestBooking.test_username in C:\Users\Admin\Desktop\testing


Process finished with exit code 0
Enter Username:

Ran 1 test in 7.547s

OK
```

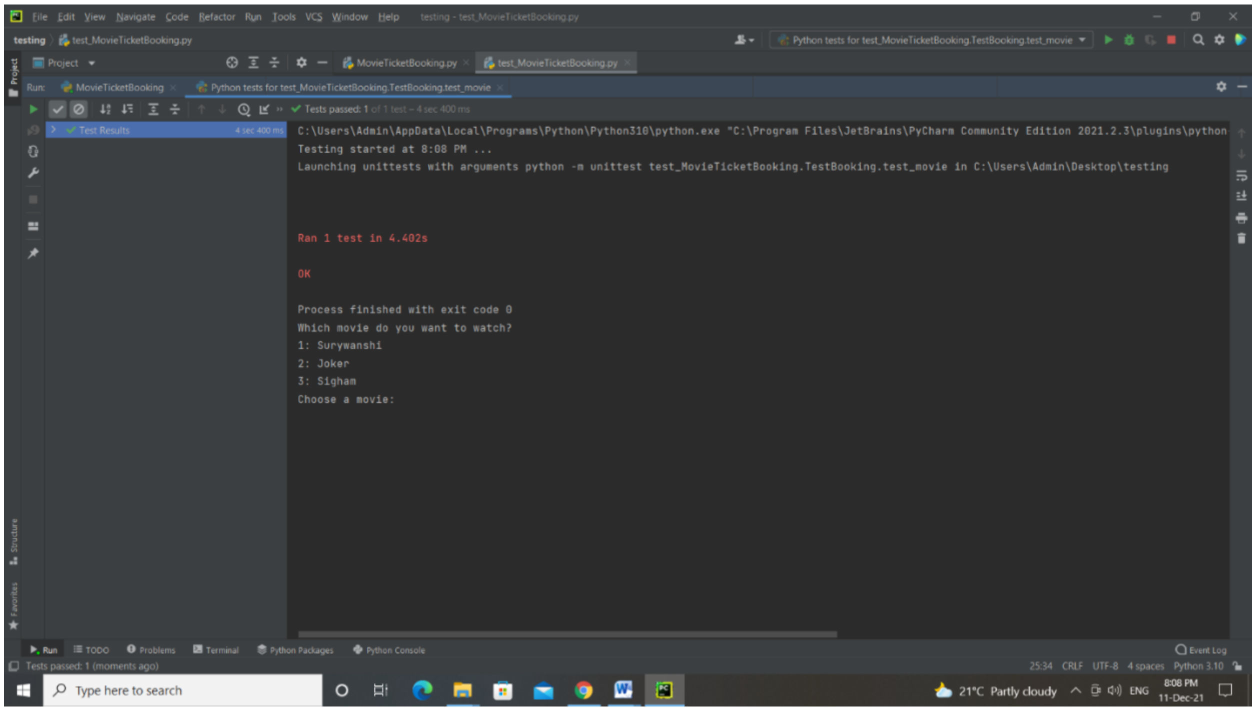


```
def test_city(self):
    result = MovieTicketBooking.city()
```

```
C:\Users\Admin\AppData\Local\Programs\Python\Python310\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.3\plugins\python
Testing started at 8:07 PM ...
Launching unittests with arguments python -m unittest test_MovieTicketBooking.TestBooking.test_city in C:\Users\Admin\Desktop\testing


Process finished with exit code 0
Hi! Welcome to Movie Ticket Booking!
Where do you want to watch the movie?
1: Mumbai
2: Pune
3: Nashik
Choose a city:

Ran 1 test in 6.270s

OK
```

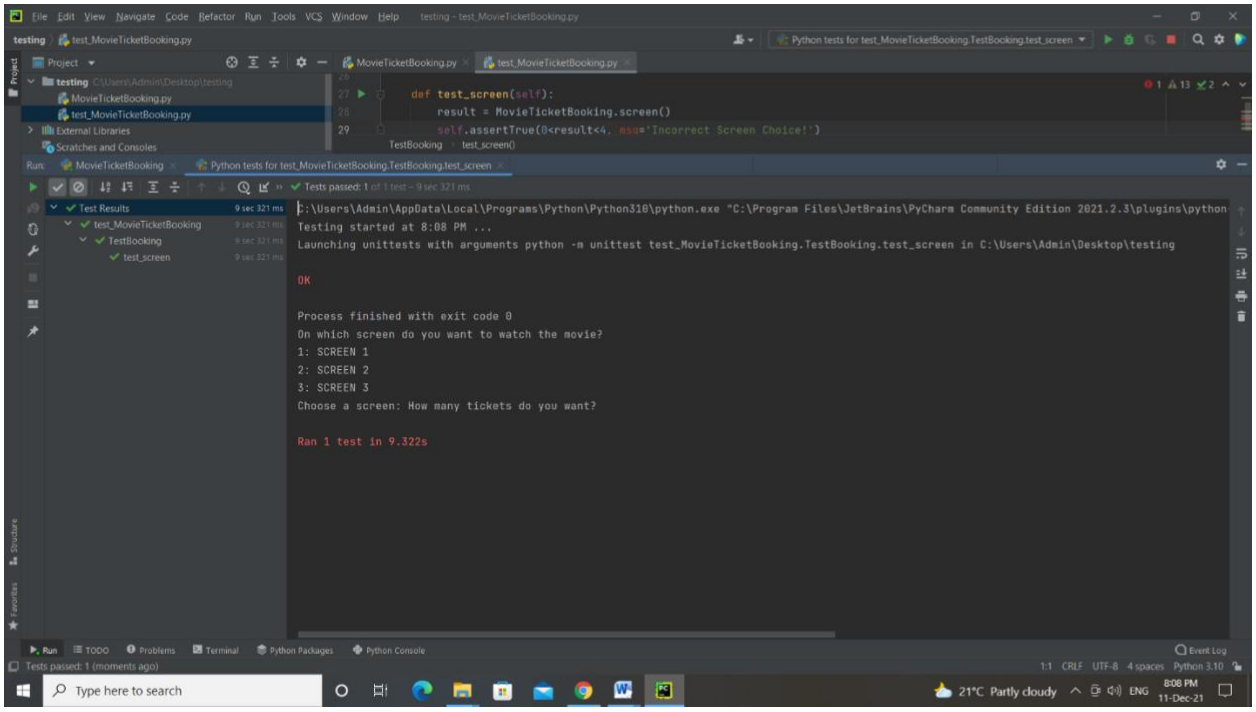


```
C:\Users\Admin\AppData\Local\Programs\Python\Python310\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.3\plugins\python
Testing started at 8:07 PM ...
Launching unittests with arguments python -m unittest test_MovieTicketBooking.TestBooking.test_theatre in C:\Users\Admin\Desktop\testing


Process finished with exit code 0
In which theater do you wish to see the movie?
1: Symbosis
2: Uma
3: lamninarayan
Choose a theatre:

Ran 1 test in 6.959s

OK
```



Tests passed: 1 of 1 test – 4 sec 400 ms

```
C:\Users\Admin\AppData\Local\Programs\Python\Python310\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.3\plugins\python
Testing started at 8:08 PM ...
Launching unittests with arguments python -m unittest test_MovieTicketBooking.TestBooking.test_movie in C:\Users\Admin\Desktop\testing



Ran 1 test in 4.402s

OK

Process finished with exit code 0
Which movie do you want to watch?
1: Suryawanshi
2: Joker
3: Sigham
Choose a movie:
```



```
    def test_screen(self):
        result = MovieTicketBooking.screen()
        self.assertTrue(0<result<4, msg='Incorrect Screen Choice!')
```

Tests passed: 1 of 1 test – 9 sec 321 ms

```
C:\Users\Admin\AppData\Local\Programs\Python\Python310\python.exe "C:\Program Files\JetBrains\PyCharm Community Edition 2021.2.3\plugins\python
Testing started at 8:08 PM ...
Launching unittests with arguments python -m unittest test_MovieTicketBooking.TestBooking.test_screen in C:\Users\Admin\Desktop\testing

OK

Process finished with exit code 0
On which screen do you want to watch the movie?
1: SCREEN 1
2: SCREEN 2
3: SCREEN 3
Choose a screen: How many tickets do you want?

Ran 1 test in 9.322s
```

ADVANTAGE

- Online booking systems save your staff time
- Removing the bottleneck of phone booking systems
- Greater sales and marketing synergy
- A modern approach to booking
- Increased revenue thanks to upselling
- Can come at a cost
- Requires internet access

CONCLUSION

We have studied about unit testing and python selenium for test all test cases of different modules of movie ticket booking system. In movie ticket booking system it detect all test cases are passed or not.

REFERENCES

- Wikipedia
- https://www.guru99.com/python-unit-testing-guide.html
- https://www.javatpoint.com
- https://www.python.org/
- https://www.tutorialspoint/