**GUJARAT TECHNOLOGICAL UNIVERSITY**



**GUJARAT TECHNOLOGICAL UNIVERSITY**

**Established under Gujarat Act No. 20 of 2007)**

**(**

**ુ**

**જરાતટકનોલોકલ**

**ુ**

**િનવિસટ**

(

ુ

જરાત

અિધિનયમ

મા

ં

કઃ

ર૦

/

ર૦૦૭

ારા

સ્થાિપત

)

**Appendix 2**

Logo

GTU

Logo



# **PYTHON & AUTOMATION**

**A PROJECT REPORT**

## *Submitted by*

**Dhruvi Vimalbhai Virani**

**190320111070**

***In partial fulfillment for the award of the degree of***

**BACHELOR OF ENGINEERING**

## *in*

**ELECTRONICS & COMMUNICATION ENGINEERING**

**L. J.INSTITUTE OF ENGINEERING AND TECHNOLOGY, AHMEDABAD**

**Gujarat Technological University, Ahmedabad**

**[MAY , 2023]**

Institute

Logo



GTU

Logo



**L. J.INSTITUTE OF ENGINEERING AND TECHNOLOGY, AHMEDABAD**

**LJ Campus, LJ College Rd, near Sanand - Sarkhej Road, Makarba, Sarkhej-Okaf, Gujarat 382210**

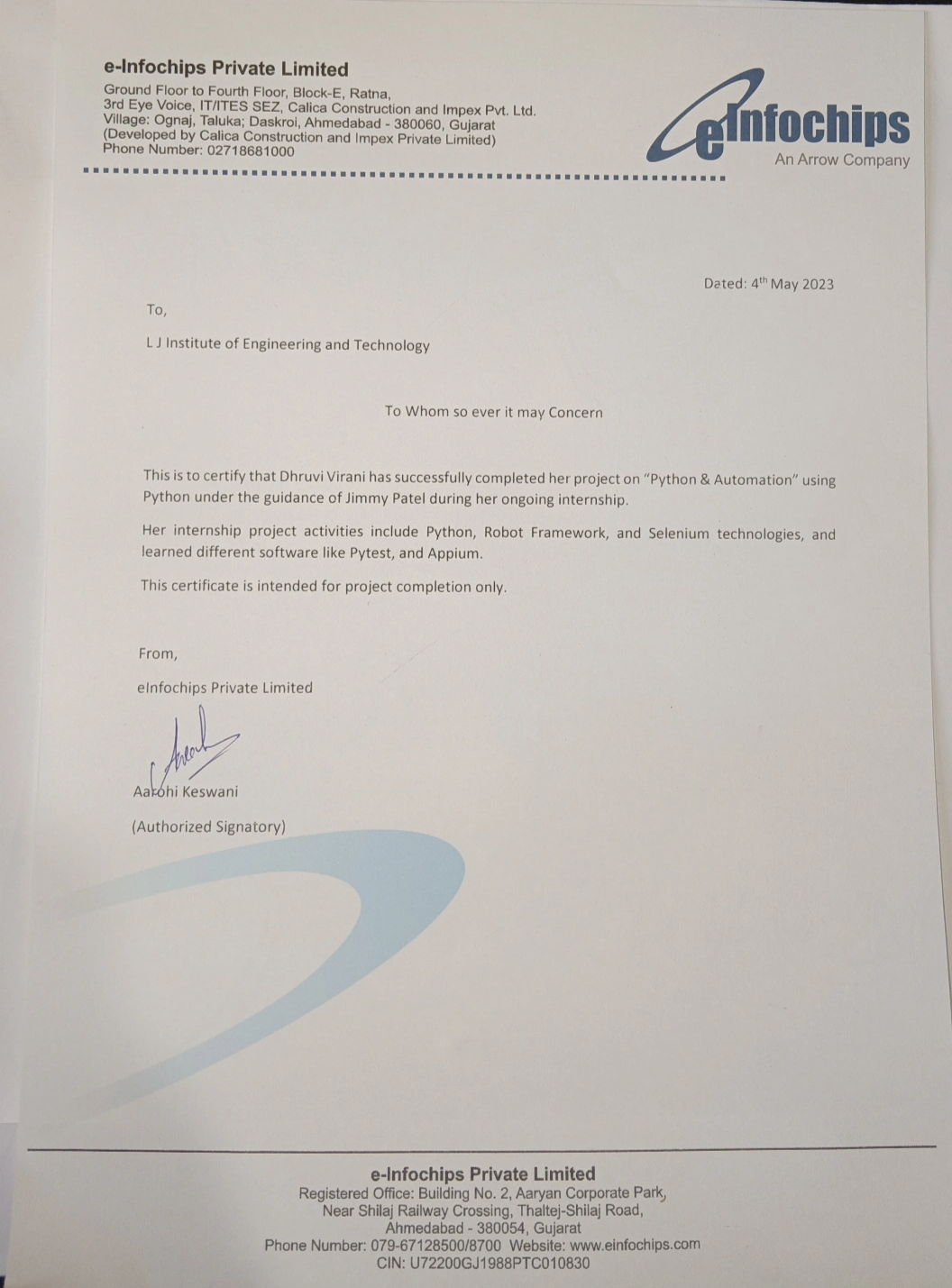
CERTIFICATE 

This is to certify that the project report submitted along with the project entitledPYTHON & AUTOMATIONhas been carried out by **Dhruvi Vimalbhai Virani** under my guidance in partial fulfillment for the degree of Bachelor of Engineering in Electronics and Communication , 8th Semester of Gujarat Technological University, Ahmadabad during the academic year 2021-22.

Mosam Pandya Mosam Pandya

Internal Guide Head of the Department

EINFOCHIPS OFFER LETTER



## ACKNOWLEDGEMENT

The internship opportunity I am having with Einfochips-An Arrow Company was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it.

Bearing in mind previous I am using this opportunity to express my deepest gratitude and special thanks to the JIMMY PATEL, Technical lead at the Company who in spite of being extraordinarily busy with her duties, took time out to hear, guide, and keep me on the correct path and allowing me During My training at Einfochips-An Arrow Company.

I express my deepest thanks to Miss. Shruti Pethani , Head of customer support for taking part in useful decisions& giving necessary advice and guidance and arranging all facilities to make life easier. I choose this moment to acknowledge his/her contribution gratefully.

## ABSTRACT

*Industrial training is an important phase of a student’s life. A well-planned, properly executed, and evaluated industrial training helps a lot in developing a professional attitude. It develops an awareness of the industrial approach to problem-solving, based on a broad understanding of the process and mode of operation of an organization.*

*The aim and motivation of this industrial training are to receive discipline, skills, teamwork, and technical knowledge through a proper training environment, which will help me, as a student in the field of IT, to develop responsiveness to the self-disciplinary nature of problems.*

*During 12 weeks of training at Einfochips-An Arrow Company, I was assigned as a technical supporter to help the company in the field of IT. Throughout this*

*industrial training, I have learned new software development languages used in*

*automation and software such as Python, Python in Selenium, Robot Framework.*

*I am also able to implement what I have learned over the past year as an* ***Electronics and***

***Communication*** *student at* ***L.J. INSTITUTE OF ENGINEERING & TECHNOLOGY*** *, Ahmedabad.*

## Table of Contents

Acknowledgement……………………………………………………………. i

Abstract………………………………………………………………………. ii

List of Figures………………………………………………………………... iii

Chapter 1: Company Overview........................................................

* 1. History
  2. Services Provided by Einfochips
     1. Digital Engineering
     2. Device Engineering
     3. Quality engineering
     4. Silicon Engineering
  3. Capacity of Plant
  4. Chart of Organization
  5. Overview Of Different Department Of The Organization

Chapter 2: My Role in my ongoing Internship at the Company.....................

Chapter 3: Introduction to Python & Automation.............................................

3.1 Introduction

3.2 Advantages of Python & Automation

3.3 Disadvantages of Python & Automation

Chapter 4: Python (Project).........................................................

4.1 Movie Guessing Game

4.2 Rock, Paper, scissor Game

4.3 Flames Game

4.4 Tic Tac Toe Game

4.5 Snake and Ladder game

4.6 Auction

4.7 Game Zone

Chapter 5: Python with Selenium...............................................................

Chapter 6: Python Plagiarism checking from text files..............................

Chapter 7: Robot Framework Automation..................................................

Chapter 8: GitHub.................................................

Chapter 9: Full Automation of Websites Using Robot Framework............

9.1 Automation on PhpTravels Website

9.2 Automation on Make My Trip Website

### Chapter 1: Company Overview

**1.1 History**

--> EInfochips, an Arrow company, is a leading global provider of product engineering and semiconductor design services. With over 500+ products developed and 40M deployments in 140 countries, eInfochips continues to fuel technological innovations in multiple verticals. The company’s service offerings include digital transformation and connected IoT solutions across various cloud platforms, including AWS and Azure.

--> Vision of Einfochips:

“We want to make eInfochips a leading global innovative technology company that will have a transformational impact on the society by creating leaders and generating stakeholder value”

– Sumit Sethi, COO.

Founded in 1994, our work culture is built over years of experience in providing innovative solutions to our clients and our indomitable spirit to excel in all aspects of our engagement. We are geographically spread across India, Japan, and the USA with 10 design centers and offices.

The fuel to our growth is driven by our passion and strong commitment to our core values:

--Customer First

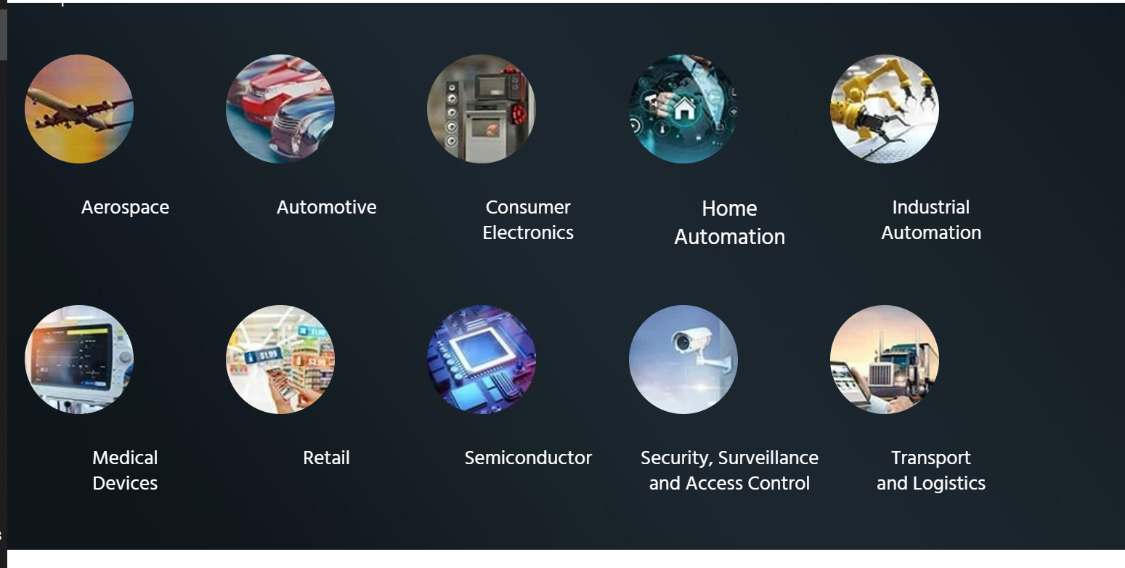
--Disciplined Execution

--Embrace Impossible Challenges

--Continuous Learning

--Serving Society

Diagram, text

Description automatically generated with medium confidence



**1.2** **Services provided by Einfochips:**

#### --Digital Engineering

~Cloud Services

~DevOps Services

~IoT Solutions & Services

~Mobility Solutions & Services

~Machine Learning Services

~Data Analytics Services ~Remote Device Management

#### --Device Engineering

~Hardware Design Services

~Design to Manufacturing

~Embedded Systems & Software Development

~Multimedia & Digital Solutions

~Image Tuning Service

~Product Sustenance Engineering

#### 

#### --Quality Engineering

~Product Testing Services

~IoT Testing Services

~Device-to-Cloud QA Automation Services

~Cognitive QA Services

~Mobile and Web Testing Services

~Quality Process Consulting ~Wireless Testing Services

#### --Silicon Engineering

~ASIC/FPGA Design & Development

~Design Verification & Validation

~Physical Design and DFT

**1.3   Capacity of plant**

Size: 1001 to 5000 Employees

* 1. **Chart of Organization**

**Graphical user interface, application, Teams

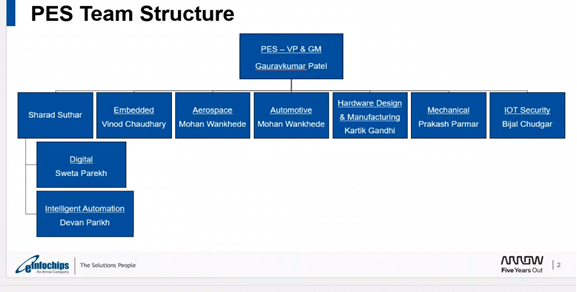
Description automatically generated**

* 1. **OVERVIEW OF DIFFERENT DEPARTMENT OF THE ORGANIZATION**

There are mainly 2 departments at E-Infochips.

* ASIC
* PES
* QEG

Diagram

Description automatically generated

**Chapter 2: My Role in my ongoing Internship at the Company**

🡪 My role in Einfochips- An Arrow Company is an engineer in Quality assurance(QA).

🡪 I have been placed in the department of Intelligence Automation(IA).

🡪 Under the mentorship of Jimmy Patel who is the Technical Guide- Level 1 of the IA department, I have been assigned in automation field in robot framework with python which is one of the most important framework in the field of automation.

🡪 I have been given a project of Python to make certain games using different functions and classes at first.

🡪 Python with selenium was also taught to us to understand the working of how automation can be done.

🡪 Further, using robot framework, I have been given several tasks to automate different websites.

### Chapter 3: Introduction to Python & Automation

**3.1 Introduction:**

**Python:**

Python was designed for readability, and has some similarities to the English language with influence from maPython uses new lines to complete a command, as opposed to other programming languages which often usePython relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and class. Python was designed for readability, and has some similarities to the English language with influence from mathematics. Python was designed for readability, and it uses new lines to complete a command, as opposed toother programming languages which often use semicolons or parentheses.Python relies on indentation, using whitespace, to define scope; such as the scope of loops,functions and classes.Python was designed for readability, and has some similarities to the English language with influencefrom mathematics.

It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python can be used on a server to create web applications.

Python can be used alongside software to create workflows.

Python can connect to database systems. It can also read and modify files.

Python can be used to handle big data and perform complex mathematics.

Python can be used for rapid prototyping, or for production-ready software development

Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).

Python has a simple syntax similar to the English language.

Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.

Python can be treated in a procedural way, an object-oriented way or a functional way.

Python was designed for readability, and has some similarities to the English language with influence from mathematics.

Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.

Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

#### Automation:

Automation Testing is a software testing technique that performs using special automated testing software tools to execute a test case suite. On the contrary, Manual Testing is performed by a human sitting in front of a computer carefully executing the test steps.

The automation testing software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Software Test Automation demands considerable investments of money and resources.

Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool, it’s possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required. This improved ROI of Test Automation. The goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing altogether.

Test Automation is the best way to increase the effectiveness, test coverage, and execution speed in software testing. Automated software testing is important due to the following reasons:

~ Manual Testing of all workflows, all fields, all negative scenarios is time and money consuming

~It is difficult to test for multilingual sites manually

~Test Automation in software testing does not require Human intervention. You can run automated test~Test Automation in software testing does not require Human intervention. You can run automated test unatte unattended.

~Test Automation increases the speed of test execution

~Automation helps increase Test Coverage

~Manual Testing can become boring and hence error-prone.

**3.2 Advantages of Python & Automation**:

The knowledge of a programming language can help a QA specialist in different ways.

Besides providing access to test automation tools, the ability to understand code also advances manual testing. One way or another, it enhances a person’s competencies and makes a software testing company better prepared for the market challenges. The question is what programming language to learn and use for writing tests. Usually, tech specialists face a “Python vs Java” dilemma. Let’s focus on the former in today’s post.

Python test automation is simply using Python for test automation.

Python is one of the testing teams’ favorite programming languages. It has multiple features that make it suitable for automated testing. For example, you can benefit from Python’s dynamic typing but also make use of checkers like Pyre for static typing. Python is easy to learn and can also be used for complex problems. Therefore, users of different skill levels can use Python.

Python is well-known for having a library available for almost everything, so you don’t need to waste time writing code from scratch. And if you find something that’s not exactly what you wanted, you can also make changes to the libraries and use them as per your use case. And in addition to all this, Python has an amazing community to help you when you’re stuck.

PyTest is a native Python test library with a superset of PyUnit’s features. Rather than modeling JUnit’s architecture, it has a distinctly Python flavor. It makes heavy use of Python decorators and assertions.

PyTest also supports parameterized testing (without the aid of plugins like Nose) that improves code reuse and simplifies code coverage.

You can create a program using Python for your web automation tasks like gathering data from websites and putting the data into an Excel spreadsheet. Most repeated workflows involving actions on the web with defined steps can be automated by writing a Python program. Once you have your scripts written, it’s simple to reuse them to make your routine tasks easier.

For an all-around better web automation solution, look to robotic process automation (RPA). With RPA, website automation is made easy for even the most basic business users. Plus, with a great RPA solution, you can easily integrate any existing Python scripts while moving forward with automation for other departments in your enterprise-wide workflows so you don’t have to start completely from scratch. The best RPA software should be easy to use with simple drag-and-drop features. That way, anyone on staff can become an automation expert without having to become a programmer.

**3.2 Disadvantages of Python & Automation**:

The downside of using Python for web automation is the programming. You’re going to need some pretty extensive programming skills, and if you’re not a trained programmer—or don’t have one on staff—setting up automated tasks with Python can be a daunting task. If you don’t have a Python expert, you shouldn’t rely on Python for automating your workflows. And if you happen to have legacy Python programs already implemented that break, you’ll need that expert around to put the pieces back together.

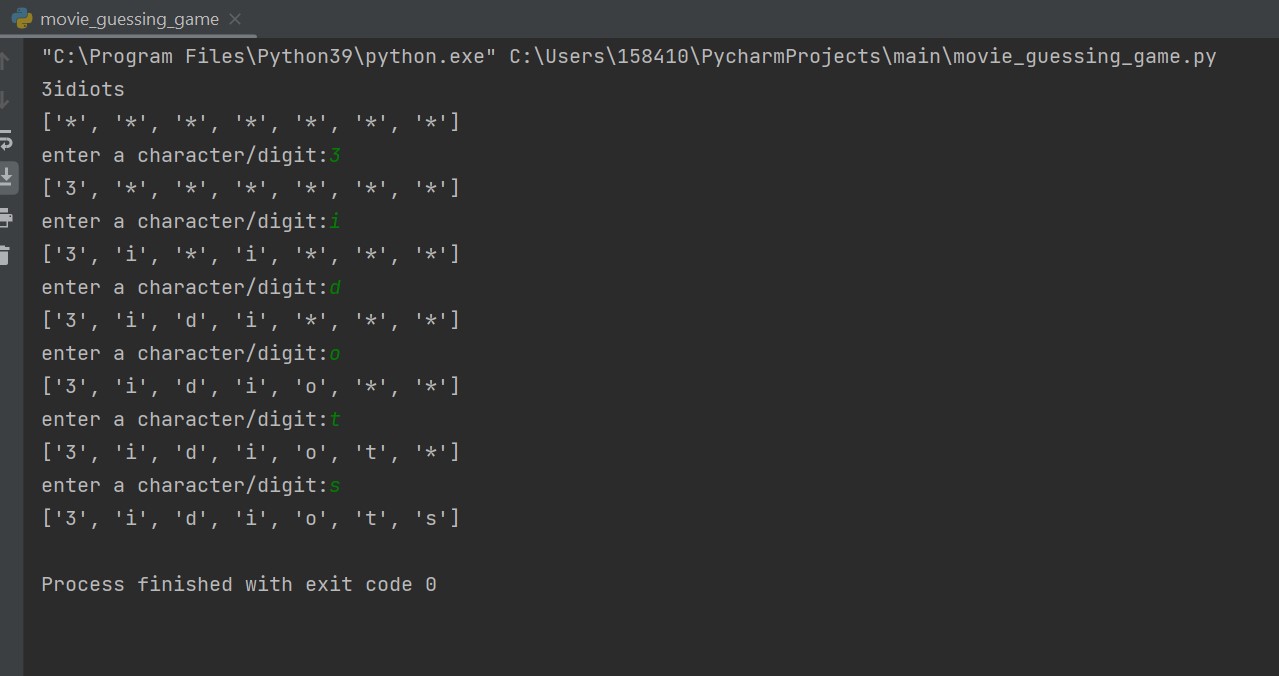
### Chapter 4: Python

**4.1 Movie Guessing Game:**

My first code of the project was to make a movie guessing game which is a one user game in which a random movie will be selected from the list of movie and there will be 5 chances to guess the correct character of the movie name.

import random  
  
def A():  
 #game = 30  
 movie = ["3idiots", "Shershah", "Pathaan", "Yeh jawani Hai deewani"]  
 movie = random.choice(movie)  
 movie = movie.replace(" ", "")  
 movie = movie.lower()  
 print(movie)  
 movie = list(movie)  
 movie1 = ["\*"] \* len(movie)  
 print(movie1)  
 chance = 5  
 while (movie != movie1 and chance > 0):  
 c = input("enter a character/digit:")  
 if c in movie:  
 for i in range(len(movie)):  
 if movie[i] == c:  
 movie1[i] = c  
 else:  
 chance = chance - 1  
 print("wrong answer you have", chance, "chances left")  
 print(movie1)  
A()

Output:



**4.2 Rock ,Paper, scissor game**:

My second game of the project was to make a rock, paper scissor game which is a two player game in which first user will enter r(rock) or p(paper) or s(scissor) and computer will choose random input.

import random  
  
def B():  
 # rock paper scissor game  
 chance = 0  
 while chance < 3:  
 turn = ["r", "p", "s"]  
 turn = random.choice(turn)  
 user = input("enter your r for rock,p for paper and s for scissor: ")  
 print(turn)  
 # turn=list(turn)  
 b = turn  
 if user == "r" and b == "p":  
 print("computer is winner")  
 elif user == "s" and b == "r":  
 print("computer is winner")  
 elif user == "s" and b == "p":  
 print("computer is winner")  
 elif user == b:  
 print("tie")  
 else:  
 print("user is winner")  
 chance += 1  
B()

**Output:**

Text

Description automatically generated

**4.3 Flames Game:** My third game was to make Flames game in which user will enter two inputs of different names and the relation between that two names will be shown in the form of flames ('F' for Friend, 'L' for Love, 'A' for Affection, 'M' for Marriage, 'E' for Enemy, 'S' for Sibling)

def C():  
 # FLAMES GAME  
 string1 = "FLAMES"  
 lst1 = list(string1)  
 name1 = input("Enter the first name:").lower()  
 name2 = input("Enter the second name :").lower()  
 lsname1 = list(name1)  
 lsname2 = list(name2)  
 for x in lsname1[:]:  
 if x in lsname2:  
 lsname1.remove(x)  
 lsname2.remove(x)  
 total = len(lsname1) + len(lsname2)  
 len\_lst1 = len(lst1)  
 count = 0  
 # To define the range how many times the game need to be played  
 while ((len(lst1)) != 1): # While length is =1 then the game is not possible  
 for x in range(len(lst1)):  
 count += 1  
 if (count == total):  
 count = 0  
 lst1.remove(lst1[x])  
 # print("The list size ",len\_lst1)  
 # len\_lst1-=1  
 print(lst1)  
C()

**Output of Flames game:** Text

Description automatically generated

**4.4 Tic Tac Toe Game:** My fourth game of the project was to make a tic tac toe game which is a two player game.

import numpy as np

def G():

mat = np.array([["\_", "\_", "\_"], ["\_", "\_", "\_"], ["\_", "\_", "\_"]])

def print\_board(mat\_copy):

global mat

print(mat\_copy)

def check\_winner():

global cwin

for i in range(0, 3):

if mat[i][0] == mat[i][1] and mat[i][0] == mat[i][2]:

if mat[i][0] == "x":

print("X won the match")

cwin = "win"

break

else:

cwin = "win"

break

else:

cwin = "loss"

for i in range(0, 3):

if mat[0][i] == mat[1][i] and mat[0][i] == mat[2][i]:

if mat[i][0] == "x":

print("X won the match")

cwin = "win"

break

else:

cwin = "win"

break

else:

cwin = "loss"

turn = 1

cwin = "loss"

print\_board(mat)

# 1 for x and 0 for O

while (cwin != "win"):

if (turn == 1):

print("x turn")

row = int(input("please enter a value of row: "))

column = int(input("please enter a value of column: "))

mat[row][column] = "x"

# print(x)

print\_board(mat)

print(cwin)

cwin = check\_winner()

print(cwin)

turn = 0

else:

print("y turn")

row = int(input("please enter a value of row: "))

column = int(input("please enter a value of column: "))

mat[row][column] = "O"

# print(x)

print\_board(mat)

cwin = check\_winner()

turn = 1

if (cwin == "win"):

print("match over")

G()**4.5 Snake and Ladder game:** My Fifth game of my project was to create a snake and ladder game which is a two player game.

import random

ladder={1:38,4:14,8:30,21:42,28:76,50:67,71:92,80:99} snake={97:78,95:56,88:24,62:18,48:26,36:6,32:10} i=0 position=0 b="y" while position<100 and b=="y":

c=0

b=input("Do you want to play: ")

Dice=random.randint(1,6) print(Dice) for i,j in ladder.items(): if i==position and c==0:

position=j

print("position is: ",position) position=position+Dice print("position is: ",position) c=1 if position>100:

position=position-Dice print("position is: ",position) for k,n in snake.items(): if k==position and c==0:

position=n

print("position is: ",position) position=position+Dice print("position is: ",position) c=1 if position>100:

position=position-Dice if c==0:

position=position+Dice if position>100:

position=position-Dice if position==100: print("you won")

**4.6 Auction:** My Sixth task of the project was to create an auction between buyers and sellers.

S1\_vm = {"S1":7,"S2":6,"S3":5,"S4":9}

S1\_bid = {"S1":20,"S2":23,"S3":19,"S4":30}

U1\_vm={"B1":4 ,"B2":3,"B3":2,"B4":5 }

U1\_bid={"B1":25 ,"B2":22,"B3":17,"B4":21 }

sorted\_s1 = dict(sorted(S1\_bid.items(), key=lambda kv: kv[1]))

sorted\_u1 = dict(sorted(U1\_bid.items(), key=lambda kv: kv[1], reverse=True)) print(sorted\_s1) print(sorted\_u1) dict1={} for i in sorted\_u1.keys(): for j in sorted\_s1.keys(): if S1\_vm[j]>=U1\_vm[i]:

dict1[i]=j

S1\_vm[j]=S1\_vm[j]-U1\_vm[i]

#del U1\_vm[i] break print("Allocation: ",dict1) lst\_seller=[] lst\_user=[] for i,j in dict1.items():

lst\_seller.append(S1\_bid[j]) lst\_user.append(U1\_bid[i])

print("Price to be paid by each buyer will be: ",max(lst\_seller)) print("Payment received to all sellers will be: ",min(lst\_user)) a1={} a2={} for i,j in dict1.items():

p=max(lst\_seller)-S1\_bid[j] q=U1\_bid[i]-min(lst\_user) a1[j]=p

# print(a1) a2[i]=q

# print(a2)

print("profit of each seller is ",a1) print("profit of each user is ",a2)

sum1=0 for i, j in a1.items(): for k, m in dict1.items(): if (i==m):

sum1=sum1+(j\*U1\_vm[k]) print(sum1)

print("the average of seller after selling the machines: ",sum1/len(a1))

sum2=0 for i,j in a2.items(): for k,m in U1\_vm.items(): if(i==k):

sum2=sum2 + (j\*m) print(sum2)

print("the average profit of seller after selling the machines: ",sum2/len(a2))

b1=len(a1) b2=len(a2)

print("the average profit of buyer per machine is ",(sum(a1.values())/b1)) print("the average profit of user per machine is ",(sum(a2.values())/b2))

**Explanation:**

STEP1:

S1\_vm = {"S1":7,"S2":6,"S3":5,"S4":9}

S1\_bid = {"S1":20,"S2":23,"S3":19,"S4":30}

U1\_vm={"B1":4 ,"B2":3,"B3":2,"B4":5 }

U1\_bid={"B1":25 ,"B2":22,"B3":17,"B4":21 }

~ S1\_vm is the dictionary for available VMs, S1\_bid is the dictionary for cost of each VM.

~ U1\_vm is the dictionary for required VMs of buyer, U1\_bid is the dictionary for bid of each VM.

~ We have 4 sellers(CSP=CLOUD SERVICE PROVIDER) and 4 buyers/users here.

~ We are taking availability of virtual machines(VMs) with each seller and the cost per VM.

~ Similarly we are taking requirement of each buyer of VM and the bid that they are keeping.

STEP2:

sorted\_s1 = dict(sorted(S1\_bid.items(), key=lambda kv: kv[1]))

sorted\_u1 = dict(sorted(U1\_bid.items(), key=lambda kv: kv[1], reverse=True)) print(sorted\_s1)

print(sorted\_u1)

~ Now we are sorting the number of sellers in ascending order of their bid

~ Also we are sorting the number of buyers in descending order of their bid.

STEP3:

dict1={} for i in sorted\_u1.keys(): for j in sorted\_s1.keys(): if S1\_vm[j]>=U1\_vm[i]:

dict1[i]=j

S1\_vm[j]=S1\_vm[j]-U1\_vm[i]

#del U1\_vm[i] break

print("Allocation: ",dict1)

~ In next step, we are allocating the buyer to the seller if seller has sufficient resources then allocated. ~ We have stored the allocation in an empty dictionary named dict1.

STEP4:

lst\_seller=[] lst\_user=[] for i,j in dict1.items():

lst\_seller.append(S1\_bid[j]) lst\_user.append(U1\_bid[i])

print("Price to be paid by each buyer will be: ",max(lst\_seller)) print("Payment received to all sellers will be: ",min(lst\_user))

~ In this step, we are creating two empty list to store maximum bid of seller and minimum bid of buyer/user respectively.

STEP5: a1={} a2={} for i,j in dict1.items():

p=max(lst\_seller)-S1\_bid[j] q=U1\_bid[i]-min(lst\_user) a1[j]=p # print(a1) a2[i]=q # print(a2)

print("profit of each seller is ",a1)

print("profit of each user is ",a2)

~ In this step, we are creating another two empty lists to store profit of each seller and each buyer in those empty l respectively.

~ For calculating profit of each seller, we are considering maximum value of cost and subtracting remaining values from that maximum

Here in my code, it is max(lst\_seller)-S1\_bid[j]

~ Similarly, for calculating profit of each buyer/user, we are considering minimum value of bid and subtracting remaining values to that minimum value. Here in my code, it is U1\_bid[i]-min(lst\_user)

STEP6: b1=len(a1) b2=len(a2)

print("the average profit of seller is ",(sum(a1)/b1)) print("the average profit of user is ",(sum(a2)/b2))

~ In this step, we are calculating the average of profit of seller and buyer/user.

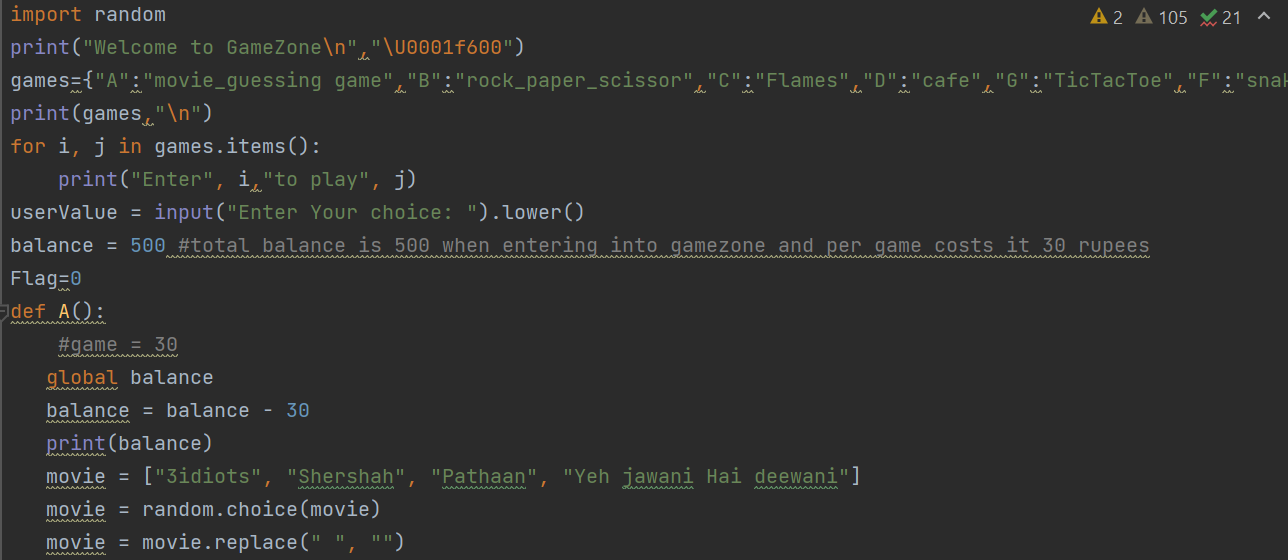
~ So, here sum(a1) is addition of profits stored in a1 and sum(a2) is addition of profits stored in a2.

Text

Description automatically generated

**:**

**4.7 Game Zone:**  My final task of the project was to merge all the games into a gamezone in which user will have a bonus card with a limit of 500 points and each game will cost 30 points. Also there will be cafe from which user can buy food from the available balance.



Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

#### Text Description automatically generated

#### 

#### Text Description automatically generated

#### Text Description automatically generated

#### Text Description automatically generated

#### Text Description automatically generated

#### Text Description automatically generated

#### Text Description automatically generated with medium confidence

#### Text Description automatically generated

#### A picture containing text Description automatically generated

#### Text Description automatically generated

#### A picture containing text Description automatically generated

#### Text Description automatically generated

print("Your current balance is ", balance)

**Output:**

Text

Description automatically generated

#### Chapter 5: Python With Selenium

🡪Selenium Python bindings provide APIs using which you can write functional tests using the Selenium WebDriver. Like other Selenium language bindings, Selenium Python APIs can be leveraged to develop highly efficient tests that let you interact with the WebElements on the AUT ( Application Under Test).

The bindings support local browser drivers (e.g., Chrome, Firefox, Internet Explorer, etc.) and provide a remote WebDriver that lets you connect to a remote cloud Selenium Grid.

🡪Different websites such as kubernetes, webucator, docker, pizza hut has been automated by me.

🡪Using different actions such as .click(), .is\_displayed(), .send\_keys(), .scroll\_location\_into\_view(), .is\_selected(), driver.find\_elements(), action chains, etc, we can automate any type of website.

🡪To automate a website, we need to inspect the particular element by Xpath, css locators, id, class,etc.

🡪Several libraries and keys need to be install and import at the beginning of the python file. such as

* from selenium import webdriver
* from selenium.webdriver.chrome.service import Service as Service
* from webdriver\_manager.chrome import ChromeDriverManager
* from selenium.webdriver.common.by import By
* from Reusable import common
* from selenium.webdriver.support.select import Select
* from selenium.webdriver.common.action\_chains import ActionChains
* from selenium.webdriver.common.alert import Alert your text

**Code of automation to validate sign in of webucator website:**

def Sign\_in(self):

driver.get("https://www.webucator.com/account/login/")

driver.implicitly\_wait(10)

time.sleep(2)

driver.find\_element(By.NAME, "login").click()

driver.find\_element(By.NAME, "login").send\_keys(common.resdXMLAsPerNode("email"))

driver.find\_element(By.NAME, "password").click()

driver.find\_element(By.NAME, "password").send\_keys(common.resdXMLAsPerNode("passc"))

time.sleep(3)

driver.find\_element(By.XPATH, "(//button[text()='Sign In'])").click()

# if driver.find\_element(By.XPATH, "(//div[@class='text-danger']/strong)").get\_attribute('value')!="" :

if driver.find\_element(By.XPATH,

"(//div[@class='text-danger']/strong)").text==common.resdXMLAsPerNode("passc"):

print(common.resdXMLAsPerNode("error"))

time.sleep(3)

**Code of automation to validate the search box and scrolling of flipkart website:**

def practice\_website2(self): global driver

driver = webdriver.Chrome() driver.get("https://www.flipkart.com/") driver.implicitly\_wait(10) driver.maximize\_window()

driver.find\_element(By.XPATH,"//button[@class='\_2KpZ6l \_2doB4z']").click() time.sleep(2)

driver.find\_element(By.XPATH,

"//form[@class='\_2M8cLY header-form-search']").is\_displayed() time.sleep(2)

driver.find\_element(By.XPATH,

"//input[@class='\_3704LK']").send\_keys("i phone 12 128bg") driver.find\_element(By.XPATH,

"//button[@class='L0Z3Pu']").click() time.sleep(2)

ele = driver.find\_element(By.XPATH,

"//div[text()='APPLE iPhone 12 Pro (Gold, 128 GB)']") actionChains = ActionChains(driver) actionChains.move\_to\_element(ele).perform()

#driver.find\_element(By.XPATH,

"//div[text()='APPLE iPhone 12 Pro (Gold, 128 GB)']").location\_once\_scrolled\_into\_view

#driver.find\_element(By.XPATH,

"//div[text()='APPLE iPhone 12 Pro (Gold, 128 GB)']").is\_displayed() time.sleep(2)

driver.find\_element(By.XPATH,

"//div[text()='APPLE iPhone 12 Pro (Gold, 128 GB)']").click() time.sleep(3)

**Code of automation to drag and drop:**

def test\_DemoWebsite(self):

global driver driver = webdriver.Chrome()

driver.get("http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html”) driver.implicitly\_wait(5) driver.maximize\_window()

src=driver.find\_element(By.XPATH,"//div[@id='box5']") dest=driver.find\_element(By.XPATH,"//div[@id='box105']") actionChains=ActionChains(driver) actionChains.drag\_and\_drop(src,dest).perform()

**Code of automation to validate the alert and confirm box:**

class test\_selenium: def handle\_Web\_Alert(self):

driver.get("https://the-internet.herokuapp.com/javascript\_alerts”) driver.implicitly\_wait(10) time.sleep(2)

driver.find\_element(By.XPATH, "//button[contains(.,'Click for JS Alert')]").click() time.sleep(2) alert = Alert(driver) if alert.text == "I am a JS Alert":

alert.accept() time.sleep(1)

driver.find\_element(By.XPATH,

"//p[text()='You successfully clicked an alert']").is\_displayed() print("Click for Js Alert Pass.") else:

print("Click for Js Alert Fail.")

# Click for Js Confirmation.........

driver.find\_element(By.XPATH, "//button[contains(.,'Click for JS Confirm')]").click() time.sleep(2) alert = Alert(driver) if alert.text == "I am a JS Confirm":

alert.dismiss() time.sleep(2)

driver.find\_element(By.XPATH, "//p[text()='You clicked: Cancel']").is\_displayed() print("Click for Js confim pass.") else:

print("Click for Js confim fail.")

# Click for Js Confirmation.........

driver.find\_element(By.XPATH, "//button[contains(.,'Click for JS Prompt')]").click() time.sleep(2) alert = Alert(driver) if alert.text == "I am a JS prompt":

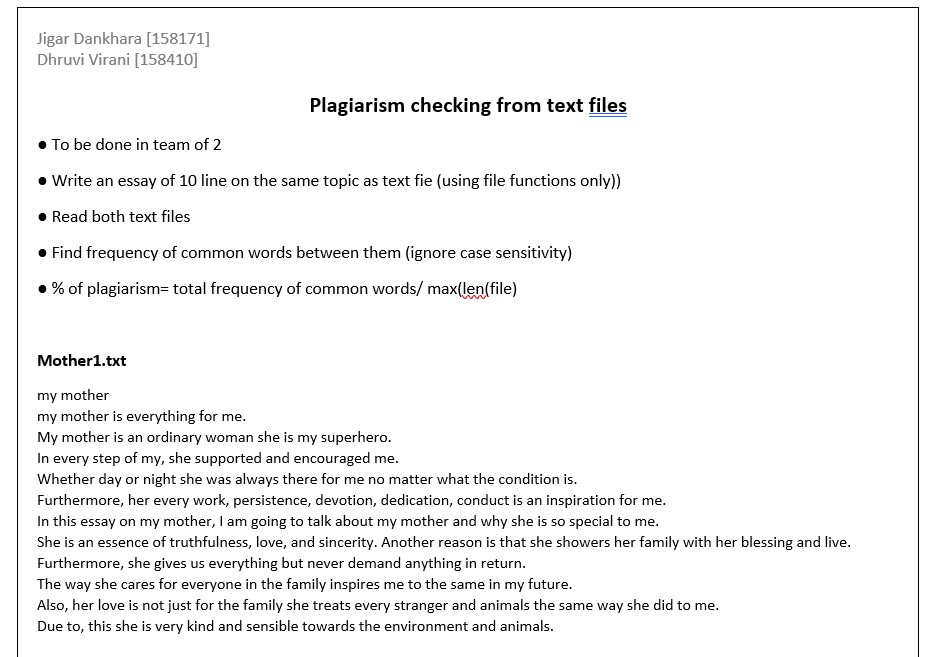
alert.send\_keys("dhruvi") alert.accept() time.sleep(2)

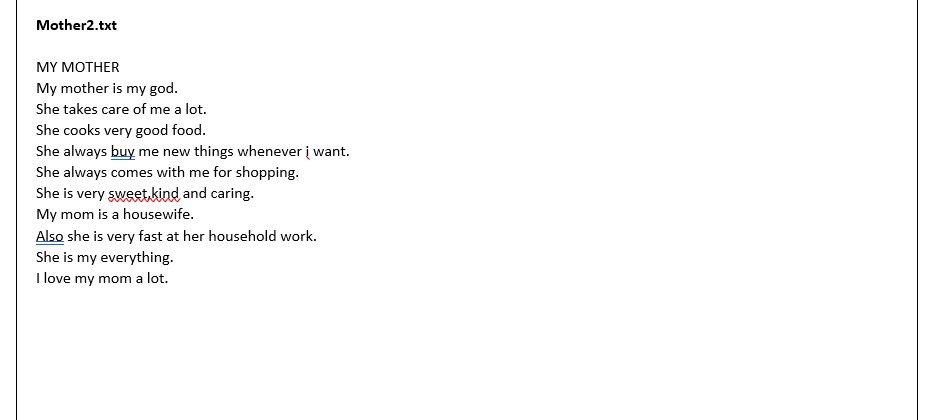
driver.find\_element(By.XPATH, "//p[text()='You entered: dhruvi']").is\_displayed() print("Click for Js Prompt With name pass.") else:

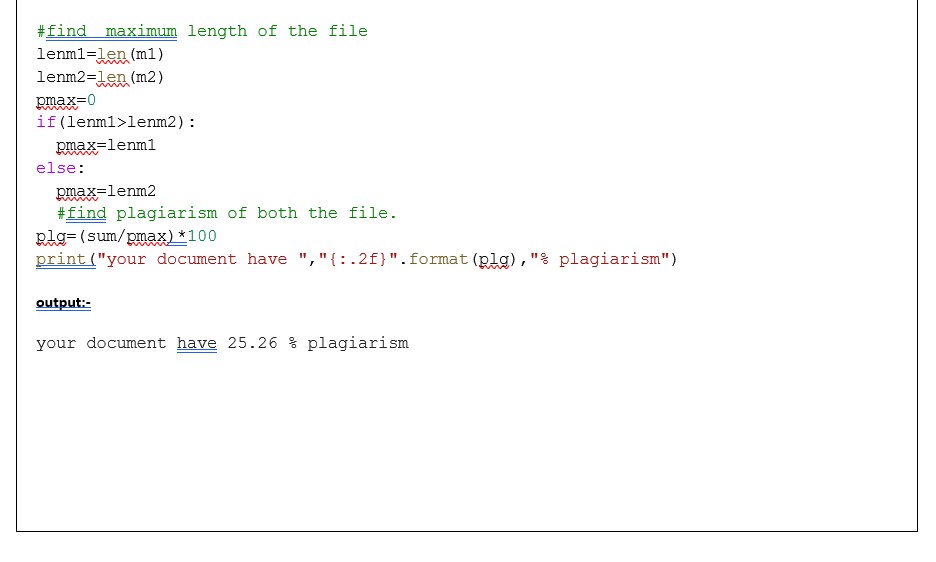
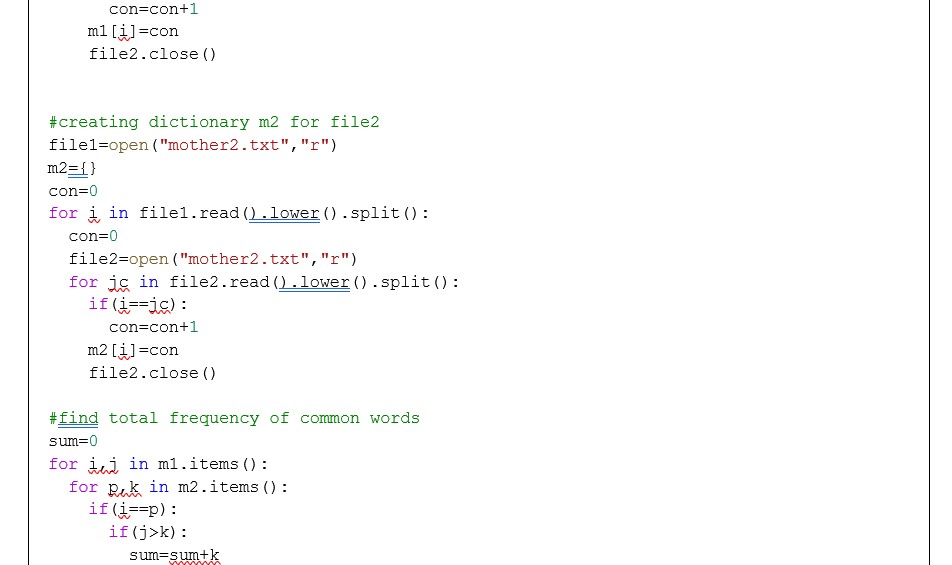
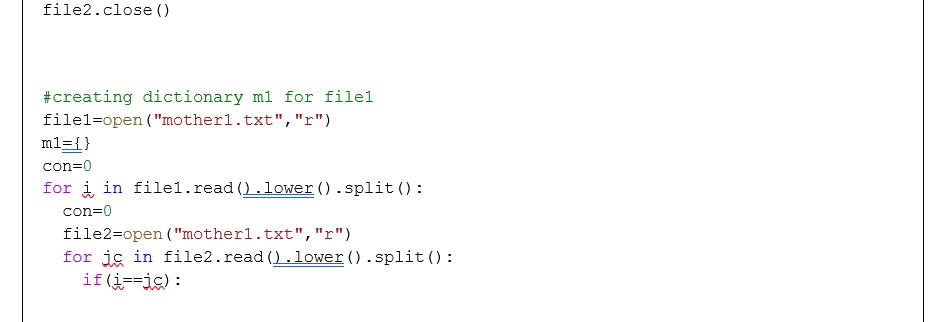
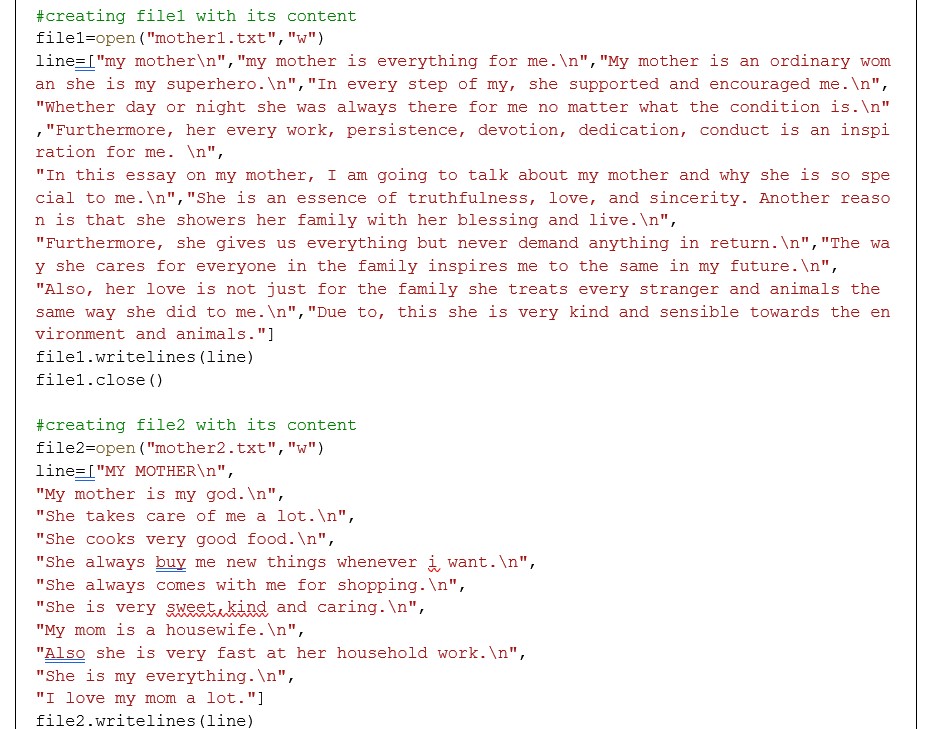
print("Click for Js Prompt With namefail.") obj=test\_selenium obj.handle\_Web\_Alert()

### Chapter 7: Python Plagiarism checking from text files

To check the plagiarism of two different codes using python code







### Chapter 8: Robot Framework Automation

Robot Framework is a generic, Python-based, open-source automation framework. It can be used for test automation and robotic process automation (RPA).

Robot Framework is supported by Robot Framework Foundation. Many industry-leading companies use the tool in their software development.

Robocorp provides tools to write, execute and orchestrate software robots that are powered by Robot Framework to be used in RPA, so understanding the basics is fundamental for any Software Robot Developer.

Robot Framework code aims to be easily readable so that even an untrained eye can understand what the code does.

Keywords are operations that your robot can execute to accomplish various tasks, and are the foundation of any robot script. Keywords can be reused, and they can be composed, which means that you can create keywords by calling other keywords.

Robot Framework itself is not built to interact with a specific system and does not have a strictly defined list of functionalities. Instead, it is built to be easily extended to work with any possible target system. These extensions to the framework are called libraries.

The framework itself ships with standard libraries, used to handle common cases. There are many libraries available to cover a variety of situations and systems.

Robocorp maintains a set of libraries that are built specifically to support RPA cases. These libraries form the RPA Framework.

Keywords are the building blocks of robots, and libraries will get you more keywords, but a robot does nothing if you don't assign it a Task.

**Automation Code to perform the validation of holiday list given in the intranet website:**

**TestCase File:**

\*\*\* Settings \*\*\*

Library SeleniumLibrary

Library Collections

Resource ../../Common/testCase\_resources.robot

Set Selenium Timeout 10 seconds

Suite Setup Open the browser

Suite TearDown Close Browser Session #Author:Dhruvi Virani

\*\*\* Test Cases \*\*\*

Validate the Intranet Holiday List

Go To Home And Then Holiday List

Fetch the Holidays list 2023 Pune

Fetch The Holidays List 2023 Ahmedabad

#Compare And Find The Difference Between Holiday List Of Both The Years

#${diff\_list}= List Difference ${HolidayList} ${HolidayList1}

**Keywords File:**

\*\*\* Settings \*\*\*

Library SeleniumLibrary

\*\*\* Keywords \*\*\*

Open the browser

Open Browser ${url} Chrome

Maximize Browser Window

Set Selenium Timeout 30 seconds

Wait Until Element Is Visible ${ElementVisible}

Go To Home And Then Holiday List

Wait Until Element Is Visible ${MouseHoverToHome}

Mouse Over ${MouseHoverToHome}

Wait Until Element Is Visible ${ClickOnHolidayList}

Click Element ${ClickOnHolidayList}

Fetch The Holidays List

[Arguments] ${year} ${location}

Click Element ${SelectHolidayYearDropdown}

Wait Until Element Is Visible ${SelectHolidayYearDropdown}

Select From List By Label ${SelectHolidayYearDropdown} ${year}

Click Element ${SelectMonths}

Click Button ${SelectSubmitButton}

@{ExpectedHolidayList} = Create List ${DescriptionOfHolidayList}

${Elements} = Get WebElements ${DescriptionOfHolidayList}

Set Global Variable ${Elements}

@{HolidayList} = Create List

Set Global Variable @{HolidayList}

FOR ${Element} IN @{Elements}

Log ${Element.text}

Append To List ${HolidayList} ${Element.text}

END

${Count}= Click Element ${ClickPagination}

${LengthOfPagination}= Get Length ${ClickPagination}

FOR ${Count} IN RANGE ${LengthOfPagination} 1 -1

Click Element [${Count}+1]

#Continue For Loop If '${Count}' == '${LengthOfPagination}'

Log ${Count}

END

FOR ${Element} IN @{Elements}

Log ${Element.text} Append To List ${HolidayList} ${Element.text}

END

Log ${HolidayList}

[return] ${HolidayList}

Compare And Find The Difference Between Holiday List Of Both The Years

${list1}= Get Length ${Elements}

Log ${list1}

${list2}= Get Length ${Elements1}

Log ${list2}

List Difference

[Arguments] ${arg1} ${arg2}

@{New\_list}= Create List @{arg1} @{arg2}

FOR ${item} IN @{arg1}

Remove Values From List ${New\_list} ${item}

END

Log ${New\_list}

Get Length ${New\_list}

Close Browser Session

Close Browser

**Variables File:**

\*\*\* Variables \*\*\*

${url} https://intranet.einfochips.com/

${ElementVisible} //a[text()='Dhruvi Virani']

${MouseHoverToHome} css=div[class='collapse navbar-collapse navbar-ex1-collapse']> ul>li>a[class='dropdown-toggle nav-link'] ${ClickOnHolidayList} //a[text()='Holiday List']

${SelectHolidayYearDropdown} css=select[id='holiday\_year']

${ClickYear} css=option[value='2022']

${SelectMonths} css=select[id='month\_id']>option

${SelectSubmitButton} css=button[class='btn btn-success']

${DescriptionOfHolidayList} css=table[id='mt'] tbody td:nth-child(4)

${ClickPagination} //ul[@class='pagination pagination-lg']

${SelectYear} css=option[value='2023']

**Resources File:**

\*\*\* Settings \*\*\*

Resource testCase\_variable.robot

Resource testCase\_labels.robot

Resource testCase\_common\_keywords.robot

Library ExtendedLibrary.py

**Automation Code to perform the validation of Practice Automation Website:**

**TestCase File:**

\*\*\* Settings \*\*\*

Resource ../../Common/Testcase\_resources.robot

\*\*\* Test Cases \*\*\*

Validate Practice Automation website

Open the browser

Validate Page Title

Validate and click Home

Click on the radio button

Type the suggestion of a country name to get the options

Select dropdown

Click on the checkbox

Validate Switch window example

Validate Switch Tab example

Validate Switch to alert example

Scroll down and validate hide and show button

Scroll down and validate mouse hover

Close Browser Session

**Keywords File:**

\*\*\* Settings \*\*\*

Library SeleniumLibrary

\*\*\* Keywords \*\*\*

Open the browser

Open Browser ${url} Chrome

Maximize Browser Window sleep 2

Validate Page Title

${title}= Get Title

Validate and click Home

Click Element ${validate\_home}

Sleep 5

Go Back

Click on the radio button

Click Element ${click\_radio\_button}

Sleep 2

Type the suggestion of a country name to get the options

Click Element ${suggestion\_textbox}

Input Text ${suggestion\_textbox} ind

Sleep 2

Click Element ${select\_dropdown}

Sleep 1

Select dropdown

Click Element ${dropdown\_example}

Sleep 2

Click Element ${select\_option}

Sleep 1

Click on the checkbox

Click Element ${select\_checkbox}

Sleep 2

Validate Switch window example

Click Element ${switch\_window}

Sleep 7

Switch Window MAIN

Sleep 3

Validate Switch Tab example

Click Element ${switch\_tab}

Sleep 12

Switch Window MAIN

Sleep 5

Validate Switch to alert example

Click Element ${switch\_to\_alert}

Input Text ${switch\_to\_alert} dhruvi

Sleep 2

Click Element ${alert\_button}

Sleep 5

Scroll down and validate hide and show button

Scroll Element Into View ${scroll\_to\_element}

Click Element ${hide\_button}

Sleep 1

Click Element ${show\_button}

Sleep 2

Scroll down and validate mouse hover

Scroll Element Into View ${scroll\_down}

Sleep 2

Mouse Over ${mouse\_hover}

Sleep 2

Click Element ${hover\_to\_top}

Sleep 2

Close Browser Session

Close Browser

**Variables File:**

\*\*\* Variables \*\*\*

${Url} https://rahulshettyacademy.com/AutomationPractice/

${validate\_home} //button[text()='Home']

${click\_radio\_button} name:radioButton

${suggestion\_textbox} id:autocomplete

${select\_dropdown} //li/div[text()='India']

${dropdown\_example} id:dropdown-class-example

${select\_option} css:option[value='option3']

${select\_checkbox} name:checkBoxOption1

${switch\_window} id:openwindow

${switch\_tab} id:opentab

${switch\_to\_alert} css:input[placeholder='Enter Your Name']

${alert\_button} id:alertbtn

${scroll\_to\_element} css:div[class='totalAmount']

${hide\_button} id:hide-textbox

${show\_button} id:show-textbox

${scroll\_down} id:courses-iframe

${mouse\_hover} id:mousehover

${hover\_to\_top} css:a[href='#top']

**Resources File:**

\*\*\* Settings \*\*\*

Library SeleniumLibrary

Resource Testcase\_common\_keywords.robot

Resource Testcase\_labels.robot

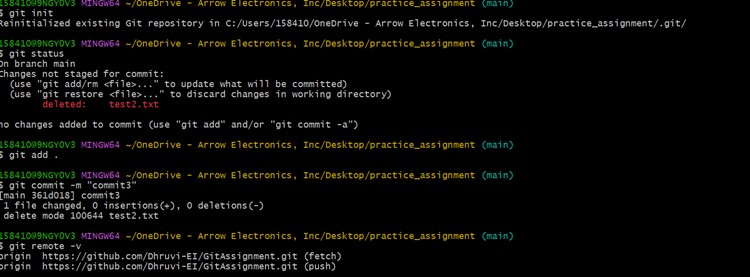
Resource Testcase\_variables.robot

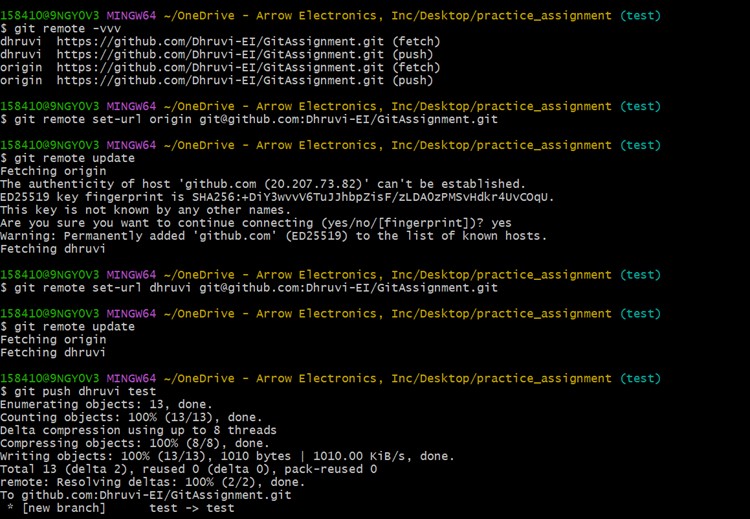
Library ExtendedLibrary.py

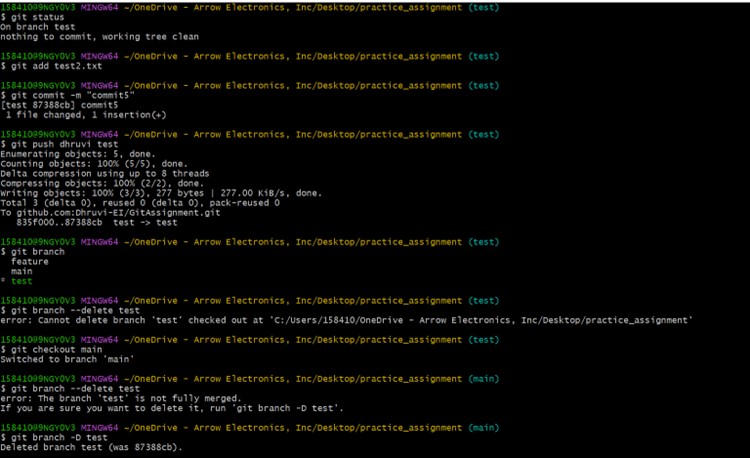
### Chapter 9: GitHub

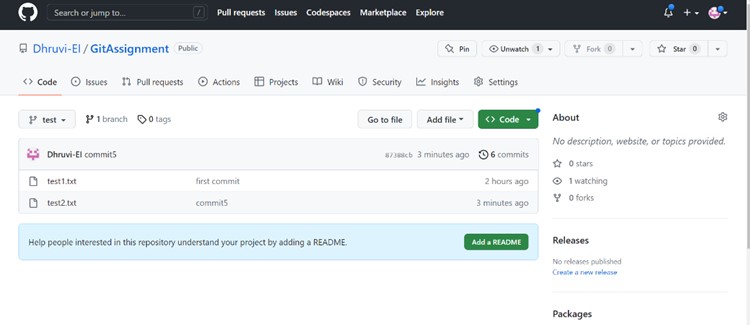
Assignment3: Create or delete a new branch in GIT, add files and commit and push your changes in branch.

1. Clone your GIT repository
2. Create a new branch in your repository
3. Add some files in it and commit and push your changes
4. Edit files and commit in same branch and push your changes
5. Delete your newly created branch









**Assignment3: Pulling out changes that you are not supposed to push in git repository.**

1. Create a new branch in your repository

1. Add some files <File1.txt> & <File2.txt> in it and commit and push your changes

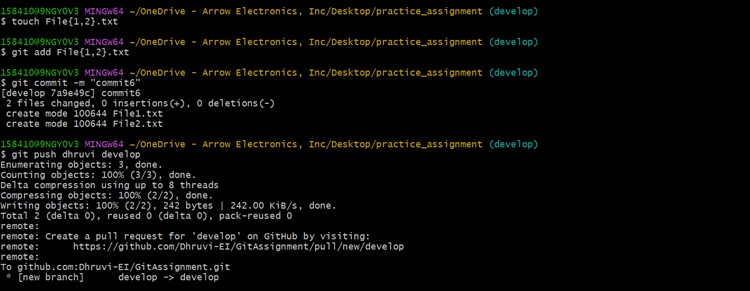
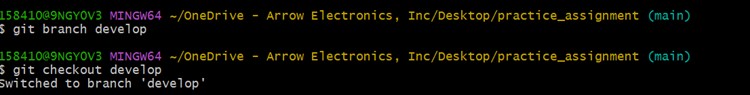
1. Edit file <File1.txt> and add this file again in GIT as you have made some changes in it

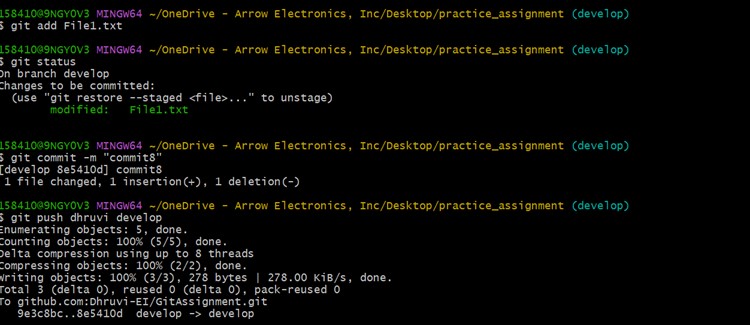
1. Now as you don’t want to make this changes so backing out all the changes in your working directory

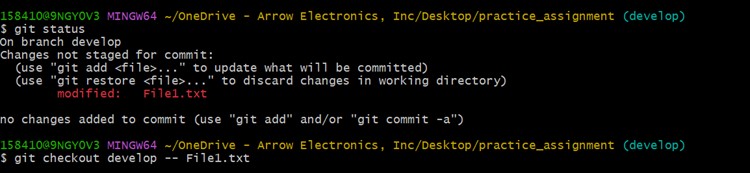
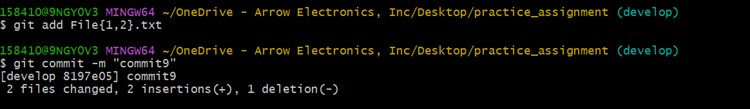
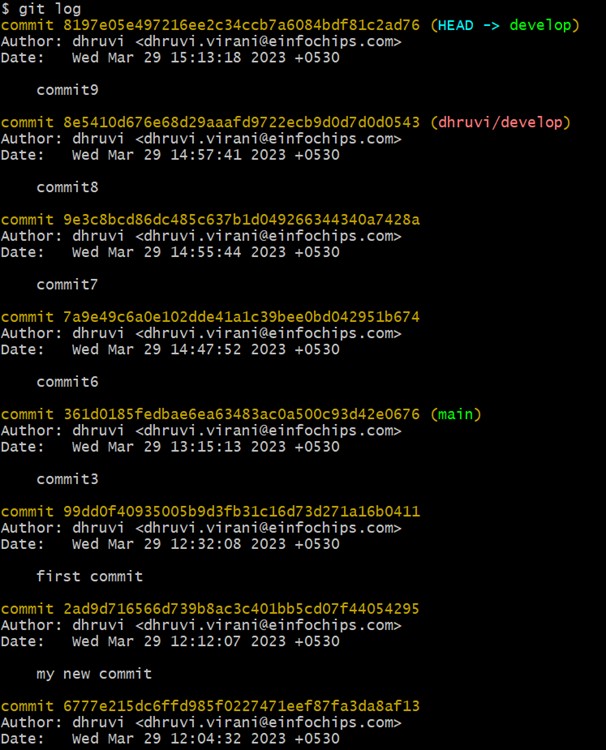
1. Now your changes are unstaged, so checkout the file to discard any changes made in <File1.txt>

1. Edit files <File1.txt> & <File2.txt> again and commit in same branch and don’t push your changes

1. Reset all the changes that you have not pushed yet and reset your branch with latest master branch









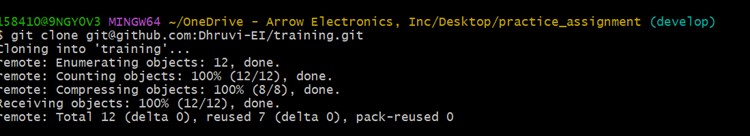
**Assignment4: Pulling out changes that you are not supposed to push in git repository.**

1. Add new remote in GIT. You can use below repository as new remote

a. https://github.com/ishabhatt/training.git

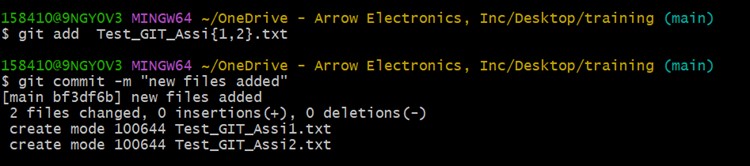
1. Create new branch (Test\_Branch\_1) in your repository from ishabhatt’s master branch

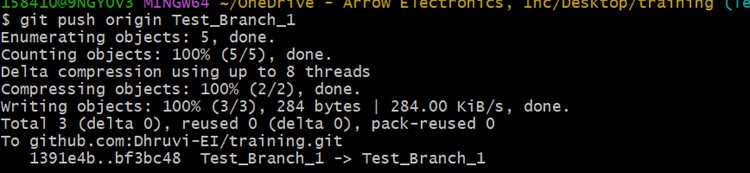
1. Add new files Test\_GIT\_Assi1.txt & Test\_GIT\_Assi2.txt in newly created branch
2. Commit your changes
3. Push your branch in your repository <YourRepo>
4. Create pull request to merge your changes in ishabhatt’s repository







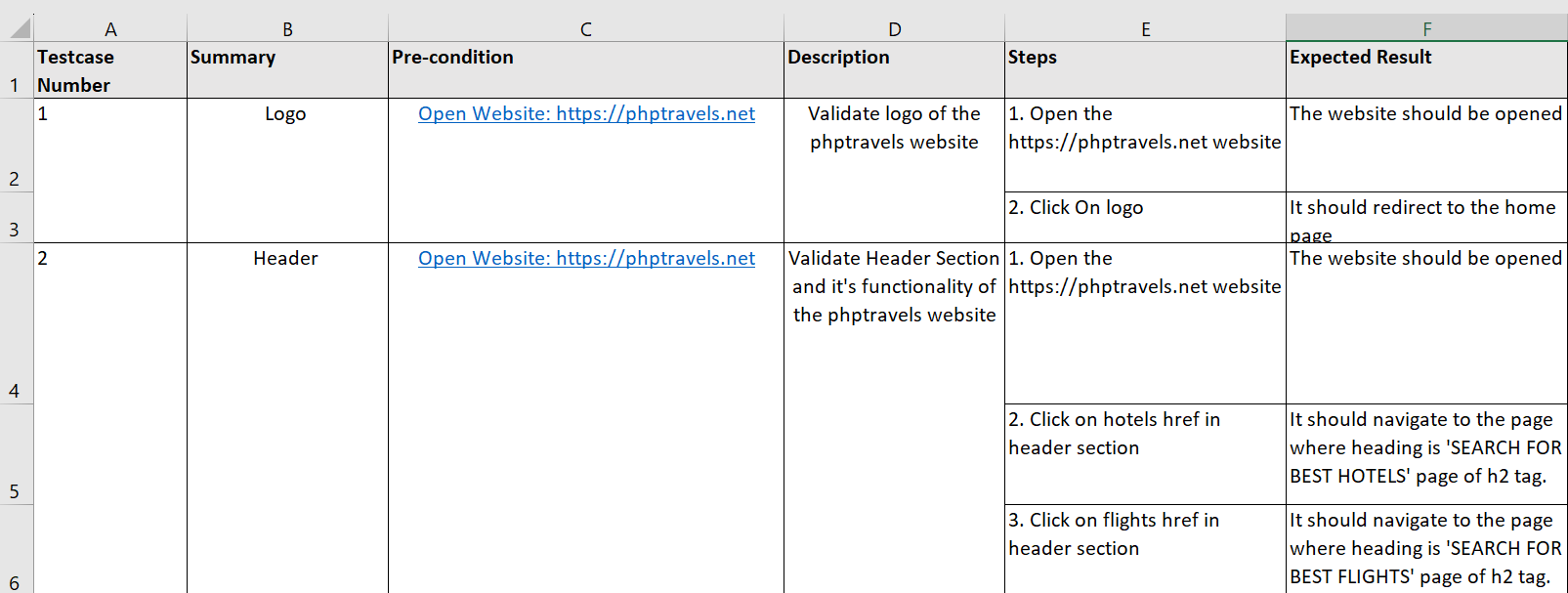




**Chapter 9: Full Automation of Websites using Robot Framework**

**9.1 Automation on PhpTravels Website:**

🡪Manual Test Cases Of Phptravels website



A picture containing graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

A picture containing calendar

Description automatically generated

Calendar

Description automatically generated with low confidence

Graphical user interface

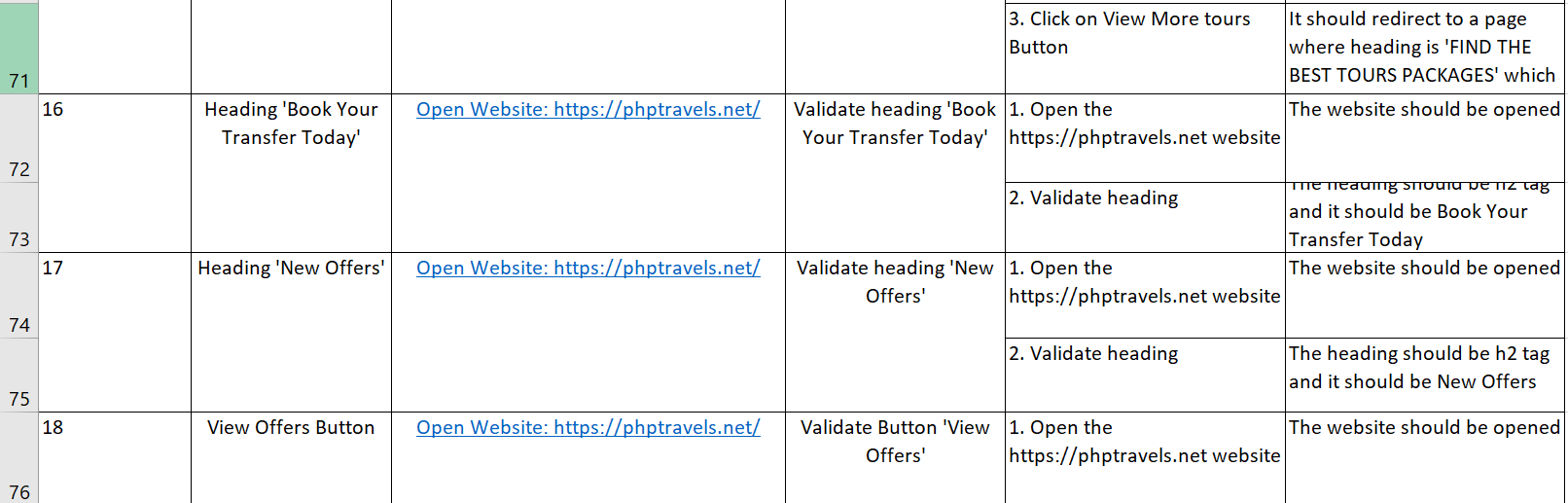
Description automatically generated with medium confidence

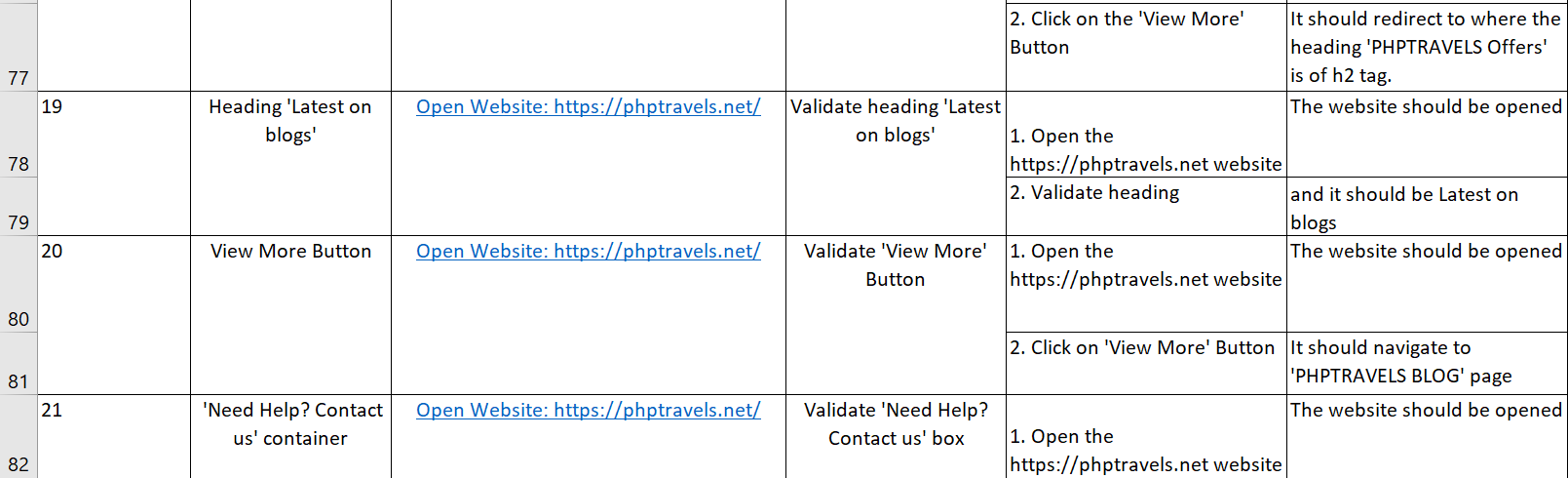
Table

Description automatically generated

Calendar

Description automatically generated with medium confidence





Calendar

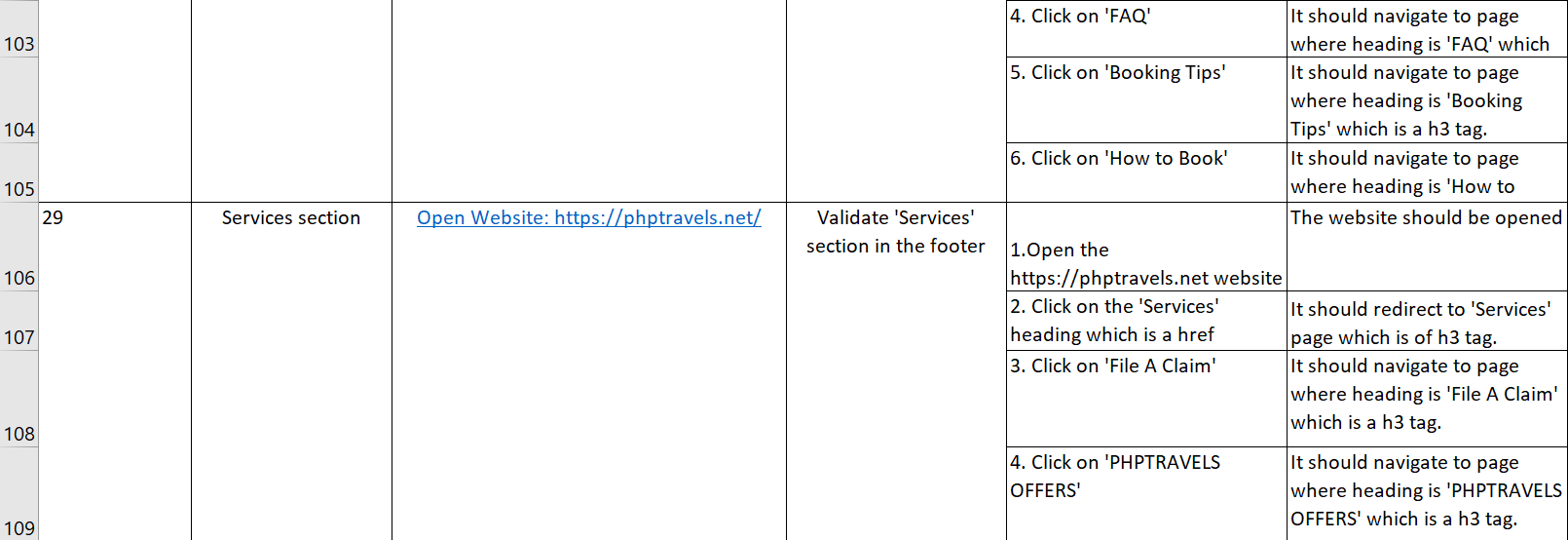
Description automatically generated

Calendar

Description automatically generated

Graphical user interface, text, application

Description automatically generated



Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated with medium confidence

A picture containing calendar

Description automatically generated

**Code :**

**TestCase File:**

\*\*\* Settings \*\*\*

Resource ../Common/testCase\_resources.robot

#Suite Setup Open The browser

#Suite TearDown Close Browser Session

#Author: Dhruvi.Virani

\*\*\* Test Cases \*\*\*

Validate Phptravels Website

Open The Browser

Validate Logo Of The Phptravels Website

Validate Header Section

Validate Ghandi Drop-down Option In The Header section

Validate English dropdown In The Header Section

Validate USD Drop-down In The Header Section

Validate Account Drop-down In The Header section

Validate Heading

Validate Hotels In Tablist

Validate Flights In Tablist

Validate Tours In Tablist

Validate Transfers In Tablist

Validate Visa In Tablist

Validate Heading 'Top Flight Destinations'

Validate Heading 'Featured Hotels'

Validate Heading 'Featured Tours'

Validate Heading 'Book Your Transfer Today'

Validate Heading 'Latest on blogs'

Validate View More Button

Validate Need Help Box

Validate Subscribe To See Secret Deals Heading

Validate Enter Email Address Box

Validate Footer Logo

#Validate Footer Mail

Validate Footer Contact Us

Validate Footer Company Section

Validate Footer Support Section

Validate Footer Services Section

Validate Footer Facebook Icon

Validate Footer Twitter Icon

Validate Footer Youtube Icon

Validate Footer Whatsapp Icon

Validate Footer Instagram Icon

Validate Footer PHPTRAVELS href

Validate Top-Up Button

Close Browser Section

\*\*\* Keywords \*\*\*

Open The Browser

Open Browser ${PhpUrl} Chrome

Maximize Browser Window

Wait Until Element Is Visible ${PhpMainHeading}

Validate Logo Of The Phptravels Website

Wait Until Element Is Visible ${PhpLogo}

Get Title

Validate Header Section

Wait Until Element Is Visible ${Header}

Get Text ${Header}

Click Element ${HeaderHotels}

Wait Until Element Is Visible ${WaitHotelsHeading}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Click Element ${HeaderFlights}

Wait Until Element Is Visible ${WaitFlightsHeading}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Click Element ${HeaderTours}

Wait Until Element Is Visible ${WaitToursHeading}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Click Element ${HeaderTransfer}

Wait Until Element Is Visible ${WaitTransferHeading}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Click Element ${HeaderVisa}

Wait Until Element Is Visible ${WaitVisaHeading}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Click Element ${HeaderBlogs}

Wait Until Element Is Visible ${WaitBlogHeading}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Validate Ghandi Drop-down Option In The Header section

Mouse Over ${HeaderGhandi}

Wait Until Element Is Visible ${GhandiDrop-down}

Click Element ${HeaderGhandi}

Wait Until Element Is Visible ${GhandiHeading}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Mouse Over ${HeaderGhandi}

Wait Until Element Is Visible ${GhandiDrop-down}

Click Element ${GhandiFAQ}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Mouse Over ${HeaderGhandi}

Wait Until Element Is Visible ${GhandiDrop-down}

Click Element ${GhandiTermsOfUse}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Mouse Over ${HeaderGhandi}

Wait Until Element Is Visible ${GhandiDrop-down}

Click Element ${GhandiHowToBook}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Mouse Over ${HeaderGhandi}

Wait Until Element Is Visible ${GhandiDrop-down}

Click Element ${GhandiAboutUs}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Validate English dropdown In The Header Section

Click Element ${HeaderEnglishSection}

Wait Until Element Is Visible ${EnglishLanguages}

Click Element ${EnglishTurkish}

Wait Until Element Is Visible ${PhpMainHeading}

Click Element ${HeaderEnglishSection}

Wait Until Element Is Visible ${EnglishLanguages}

Click Element ${EnglishLanguage}

Wait Until Element Is Visible ${PhpMainHeading}

Validate USD Drop-down In The Header Section

Click Element ${USDButton}

Wait Until Element Is Visible ${EnglishLanguages}

Click Element ${USDEuro}

Wait Until Element Is Visible ${PhpMainHeading}

Click Element ${USDButton}

Wait Until Element Is Visible ${EnglishLanguages}

Click Element ${USD}

Wait Until Element Is Visible ${PhpMainHeading}

Validate Account Drop-down In The Header section

Set Selenium Timeout 10 seconds

Click Element ${AccountButton}

Wait Until Element Is Visible ${EnglishLanguages}

Click Element ${CustomerLogin}

Wait Until Element Is Visible ${CustomerLoginHeading}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Click Element ${AccountButton}

Wait Until Element Is Visible ${EnglishLanguages}

Click Element ${CustomerSignIn}

Wait Until Element Is Visible ${CustomerLoginHeading}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Click Element ${AccountButton}

Wait Until Element Is Visible ${EnglishLanguages}

Click Element ${SupplierLogin}

Switch Window NEW

Wait Until Element Is Visible ${SupplierLoginHeading}

${GetTitle}= Get Text ${SupplierLoginHeading}

Log ${GetTitle}

Switch Window MAIN

Validate Heading

Wait Until Element Is Visible ${PhpMainHeading}

Element Text Should Be ${PhpMainHeading} Let’s book your next trip!

${title}= Get Text ${PhpMainHeading}

Log ${title}

Validate Hotels In Tablist

Sleep 1

Element Text Should Be ${BodyHotelsText} Hotels

Click Element ${ClickHotels}

Input Text ${InputHotels} Ahmedabad

Wait Until Element Is Visible ${HotelsFlagIn}

Click Element ${HotelsFlagIn}

Click Element ${HotelsTravellers}

Click Element ${HotelsSearch}

Wait Until Element Is Visible ${HotelsAfterSearchPage}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Validate Flights In Tablist

Element Text Should Be ${BodyFlightsText} Flights

Click Element ${ClickFlights}

Click Element ${FlightsRoundTrip}

Click Element ${FlightsInputFlyingFrom}

Input Text ${FlightsInputFlyingFrom} Ahmedabad

Wait Until Element Is Visible ${FlightsSourceWaitElementVisible}

Click Element ${FlightsSourceWaitElementVisible}

Click Element ${FlightsInputDestination}

Input Text ${FlightsInputDestination} Dubai

Wait Until Element Is Visible ${FlightsDestinationWaitElementVisible}

Click Element ${FlightsDestinationWaitElementVisible}

Click Element ${FlightsSwapButton}

Click Element ${FlightsTravellers}

Click Element ${FlightsSubmit}

Wait Until Element Is Visible ${FlightsAfterSearch}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Validate Tours In Tablist

Set Selenium Timeout 10 seconds

Element Text Should Be ${BodyToursText} Tours

Click Element ${ClickTours}

Click Element ${ToursSelectCountry}

Input Text ${ToursInput} Ahmedabad

Wait Until Element Is Visible ${HotelsFlagIn}

Click Element ${HotelsFlagIn}

Click Element ${ToursSubmitButton}

Wait Until Element Is Visible ${ToursWaitAfterSubmit}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Validate Transfers In Tablist

Element Text Should Be ${BodyTransferText} Transfers

Click Element ${ClickTransfer}

Click Element ${TransferSelectCountry}

Input Text ${TransferInput} Ahmedabad

Wait Until Element Is Visible ${HotelsFlagIn}

Click Element ${HotelsFlagIn}

Click Element ${TransferClickAfterInput}

Input Text ${TransferInputTextDestination} Dubai

Wait Until Element Is Visible ${TransferWaitUntilDestinationIsVisible}

Click Element ${TransferWaitUntilDestinationIsVisible}

Click Element ${TransferSubmit}

Wait Until Element Is Visible ${TransferWaitAfterSubmit}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Validate Visa In Tablist

Element Text Should Be ${BodyVisaText} Visa

Click Element ${ClickVisa}

execute javascript window.scrollTo(0,300)

Wait Until Element Is Visible ${VisaScroll}

Click Element ${VisaSelectCountrySource}

Input Text ${VisaInputSource} India

Wait Until Element Is Visible //li[text()='India']

Click Element ${VisaInputText}

Click Element ${VisaSelectCountryDestination}

Input Text ${VisaInputDestination} American Samoa

Wait Until Element Is Visible ${VisaInputTextDest}

Click Element ${VisaInputTextDest}

Click Element ${VisaSubmitButton}

Wait Until Element Is Visible ${VisaWaitAfterSubmit}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Validate Heading 'Top Flight Destinations'

Wait Until Element Is Visible ${ValidateHeading1}

Element Text Should Be ${TopFlightDestinations} Top Flight Destinations

${Heading1}= Get Text ${ValidateHeading1}

Log ${Heading1}

Validate Heading 'Featured Hotels'

Wait Until Element Is Visible ${ValidateHeading2}

Element Text Should Be ${FeaturedHotels} Featured Hotels

${Heading2}= Get Text ${ValidateHeading2}

Log ${Heading2}

Validate Heading 'Featured Tours'

Wait Until Element Is Visible //h2[@class='sec\_\_title\_left']

Element Text Should Be css=h2[class='sec\_\_title\_left'] Featured Tours

${Heading3}= Get Text //h2[@class='sec\_\_title\_left']

Log ${Heading3}

Validate Heading 'Book Your Transfer Today'

Wait Until Element Is Visible ${ValidateHeading3}

Element Text Should Be ${BookYourTransferToday} Book Your Transfer Today

${Heading4}= Get Text ${ValidateHeading3}

Log ${Heading4}

Validate Heading 'Latest on blogs'

Wait Until Element Is Visible ${ValidateHeading4}

Element Text Should Be ${LatestOnBlogs} Latest on blogs

${Heading5}= Get Text ${ValidateHeading4}

Log ${Heading5}

Validate View More Button

execute javascript window.scrollTo(0,4000)

Sleep 3

Click Element ${ViewButton}

Wait Until Element Is Visible ${AfterViewButton}

Go Back

Wait Until Element Is Visible ${ViewButton}

Validate Need Help Box

execute javascript window.scrollTo(0,4250)

Sleep 3

Click Element ${NeedHelpBox}

Wait Until Element Is Visible ${AfterViewButton}

Go Back

Wait Until Element Is Visible ${PhpMainHeading}

Validate Subscribe To See Secret Deals Heading

execute javascript window.scrollTo(0,4500)

Sleep 1

Element Text Should Be ${SubscribeHeading} Subscribe to see Secret Deals

Wait Until Element Is Visible ${SubscribeHeading}

${SectionHeading}= Get Text ${SubscribeHeading}

Log ${SectionHeading}

Validate Enter Email Address Box

execute javascript window.scrollTo(0,4500)

Sleep 1

Element Text Should Be ${FooterEmailText} Enter email addresss

${email}= Get Text ${FooterEmail}

Log ${email}

Element Text Should Be   ${SubscribeInput}     Subscribe to see Secret Deals

    Input Text   ${FooterEmail}  abc@gmail.com

    Click Element    ${SubscribeButton}

Validate Footer Logo

execute javascript  window.scrollTo(0,4600)

    Sleep   1

    Click Element   ${FooterLogo}

    Sleep   2

Validate Footer Mail

Switch Window MAIN

execute javascript window.scrollTo(0,4700)

Sleep 1

#Wait Until Element Is Visible css=a[href='mailto:info@travelagency.com']

Click Element //a[@href='mailto:info@travelagency.com']

Sleep 8

Close Window

Validate Footer Contact Us

Switch Window MAIN

execute javascript window.scrollTo(0,4800)

Sleep 1

Click Element ${FooterContactUs}

Wait Until Element Is Visible ${WaitAfterContactUs}

Go Back

Sleep 2

Validate Footer Company Section

execute javascript window.scrollTo(0,4800)

Sleep 1

Click Element ${FooterCompany}

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Sleep 1

Click Element ${FooterAboutUs}

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Sleep 1

Click Element ${FooterTermsOfUse}

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Sleep 1

Click Element ${FooterCookiesPolicy}

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Sleep 1

Click Element ${FooterPrivacyPolicy}

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Sleep 1

Validate Footer Support Section

execute javascript window.scrollTo(0,4800)

Sleep 1

Click Element ${FooterSupport}

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Sleep 1

Click Element ${Footersupplier}

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Sleep 1

Click Element ${FooterFAQ}

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Sleep 1

Click Element ${FooterBookingTips}

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Sleep 1

Click Element ${FooterHowToBook}

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Sleep 1

Validate Footer Services Section

execute javascript window.scrollTo(0,4800)

Sleep 1

Click Element css=a[href='services']

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Switch Window Main

execute javascript window.scrollTo(0,4850)

Sleep 1

Wait Until Element Is Visible ${FooterFileAClaim}

Click Element ${FooterFileAClaim}

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Switch Window MAIN

execute javascript window.scrollTo(0,4850)

Sleep 1

Wait Until Element Is Visible ${FooterOffers}

Click Element ${FooterOffers}

Wait Until Element Is Visible ${OfferHeading}

Go Back

Switch Window MAIN

execute javascript window.scrollTo(0,4850)

Sleep 1

Wait Until Element Is Visible ${SignupSupplier}

Click Element ${SignupSupplier}

Wait Until Element Is Visible ${SupplierSignupHeading}

Go Back

Switch Window MAIN

execute javascript window.scrollTo(0,4850)

Sleep 1

Wait Until Element Is Visible ${FooterCareers}

Click Element ${FooterCareers}

Wait Until Element Is Visible ${CompanyHeadings}

Go Back

Sleep 1

Validate Footer Facebook Icon

execute javascript window.scrollTo(0,5000)

Sleep 1

Click Element ${FooterFacebook}

Switch Window NEW

Wait Until Element Is Visible ${FacebookWait}

Click Element ${FacebookWait}

Wait Until Element Is Visible ${BackFromFacebook}

Close Window

Validate Footer Twitter Icon

Switch Window MAIN

execute javascript window.scrollTo(0,5000)

Sleep 1

Click Element ${FooterTwitter}

Switch Window NEW

Wait Until Element Is Visible ${TwitterWait}

Close Window

Validate Footer Youtube Icon

Switch Window MAIN

execute javascript window.scrollTo(0,5000)

Sleep 1

Click Element ${FooterYoutbe}

Switch Window NEW

Wait Until Element Is Visible ${YoutubeWait}

Close Window

Validate Footer Whatsapp Icon

Switch Window MAIN

execute javascript window.scrollTo(0,5000)

Sleep 1

Click Element ${FooterWhatsapp}

Switch Window NEW

Wait Until Element Is Visible ${WhatsappWait}

Close Window

Validate Footer Instagram Icon

Switch Window MAIN

execute javascript window.scrollTo(0,5000)

Sleep 1

Click Element ${FooterInstagram}

Switch Window NEW

Wait Until Element Is Visible ${InstagramWait}

Close Window

Validate Footer PHPTRAVELS href

Switch Window MAIN

Click Element ${FooterPHPtravels}

execute javascript window.scrollTo(0,5000)

Sleep 1

Click Element ${PhptravelsPage}

Switch Window NEW

Wait Until Element Is Visible ${PhpWait}

Close Window

Validate Top-Up Button

Switch Window MAIN

execute javascript window.scrollTo(0,5000)

Sleep 1

Click Element ${Top-Up}

Wait Until Element Is Visible ${PhpMainHeading}

Close Browser Section

Close Browser

**Variables locators file:**

\*\*\* Variables \*\*\*

${PhpUrl} https://phptravels.net/

${PhpMainHeading} css=h2[class='pb-1']

${PhpLogo} css=div[class='logo']

${Header} css=div[class='w-100']

${HeaderHotels} css=a[class='active\_hotels waves-effect']

${WaitHotelsHeading} //h2[text()='SEARCH FOR BEST HOTELS']

${HeaderFlights} css=a[class='active\_flights waves-effect']

${WaitFlightsHeading} //h2[text()='SEARCH FOR BEST FLIGHTS']

${HeaderTours} css=a[class='active\_tours waves-effect']

${WaitToursHeading} //h2[text()='Find the best tours packages']

${HeaderTransfer} css=a[class='active\_cars waves-effect']

${WaitTransferHeading} (//h2[text()='Book Your Transfer Today'])[1]

${HeaderVisa} css=a[class='active\_visa waves-effect']

${WaitVisaHeading} //h2[text()='Submit Your Visa Today!']

${HeaderBlogs} css=a[class='active\_blog waves-effect']

${WaitBlogHeading} //h2[text()='PHPTRAVELS Blog']

${HeaderGhandi} (//li[@class='footm'])[1]

${GhandiDrop-down} (//ul[@class='dropdown-menu-item'])[1]

${GhandiHeading} css=h3[class='title stroke-shape']

${GhandiFAQ} (//a[text()='FAQ'])[1]

${GhandiTermsOfUse} (//a[text()='Terms of Use'])[1]

${GhandiHowToBook} (//a[text()='How to Book'])[1]

${GhandiAboutUs} (//a[text()='About Us'])[1]

${HeaderEnglishSection} (//button[@class='btn btn-default dropdown-toggle waves-effect'])[1]

${EnglishLanguages} css=ul[class='dropdown-menu show']

${EnglishTurkish} //a[text()=' Turkish']

${EnglishLanguage} //a[text()=' English']

${USDButton} //button[@id='currency']

${USDEuro} //a[text()=' EUR']

${USD} //a[text()=' USD']

${AccountButton} //button[@id='ACCOUNT']

${CustomerLogin} //a[text()='Customer Login']

${CustomerLoginHeading} css=h5[class='modal-title title']

${CustomerSignIn} //a[text()='Customer Signup']

${SupplierLogin} css=a[href='https://phptravels.net/api/supplier']

${SupplierLoginHeading} css=h1[class='display-5 mb-0']

${BodyHotelsText} //span[@class='text']

${ClickHotels} //span[@id='select2-hotels\_city-container']

${InputHotels} css=input[class='select2-search\_\_field']

${HotelsFlagIn} css=i[class='flag in']

${HotelsTravellers} (//a[@class='dropdown-toggle dropdown-btn travellers waves-effect'])[1]

${HotelsSearch} (//div[@class='btn-search text-center'])[1]

${HotelsAfterSearchPage} css=a[class='mt-5 mb-2 btn btn-primary text-center loading\_effect waves-effect']

${BodyFlightsText} (//span[@class='text'])[2]

${ClickFlights} (//span[@class='d-none d-lg-block d-xl-block'])[2]

${FlightsRoundTrip} //input[@id='round-trip']

${FlightsInputFlyingFrom} (//input[@placeholder='Flying From'])[1]

${FlightsSourceWaitElementVisible} //div[text()='Ahmedabad, India']

${FlightsInputDestination} (//input[@placeholder='To Destination'])[1]

${FlightsDestinationWaitElementVisible} //strong[text()=' Dubai Intl']

${FlightsSwapButton} css=div[class='swap-places waves-effect shadow']

${FlightsTravellers} (//p[text()='Travellers '])[2]

${FlightsSubmit} css=button[id='flights-search']

${FlightsAfterSearch} css=div[class='card-header']

${BodyToursText} (//span[@class='text'])[3]

${ClickTours} (//span[@class='d-none d-lg-block d-xl-block'])[3]

${ToursSelectCountry} //span[@id='select2-tours\_city-container']

${ToursInput} css=input[class='select2-search\_\_field']

${ToursSubmitButton} (//button[@id='submit'])[2]

${ToursWaitAfterSubmit} css=a[class='mt-5 mb-2 btn btn-primary text-center loading\_effect waves-effect']

${BodyTransferText} (//span[@class='text'])[4]

${ClickTransfer} (//span[@class='d-none d-lg-block d-xl-block'])[4]

${TransferSelectCountry} css=span[id='select2-carfrom-container']

${TransferInput} css=input[class='select2-search\_\_field']

${TransferClickAfterInput} css=span[id='select2-carto-container']

${TransferInputTextDestination} css=input[aria-controls='select2-carto-results']

${TransferWaitUntilDestinationIsVisible} css=li[class='select2-results\_\_option select2-results\_\_option--highlighted']

${TransferSubmit} (//button[@id='submit'])[3]

${TransferWaitAfterSubmit} //strong[text()='Back To Search']

${BodyVisaText} (//span[@class='text'])[5]

${ClickVisa} //span[text()='visa']

${VisaScroll} css=i[class='la la-bullhorn']

${VisaSelectCountrySource} (//span[text()='Select Country'])[1]

${VisaInputSource} css=input[class='select2-search\_\_field']

${VisaInputText} //li[text()='India']

${VisaSelectCountryDestination} (//span[text()='Select Country'])

${VisaInputDestination} css=input[class='select2-search\_\_field']

${VisaInputTextDest} //li[text()='American Samoa']

${VisaSubmitButton} (//button[@id='submit'])[4]

${VisaWaitAfterSubmit} css=h2[class='sec\_\_title\_list text-center my-5']

${ValidateHeading1} //h2[text()='Top Flight Destinations']

${TopFlightDestinations} (//h2[@class='sec\_\_title line-height-55'])[1]

${ValidateHeading2} //h2[text()='Featured Hotels']

${FeaturedHotels} css=h2[class='sec\_\_title\_left']

${ValidateHeading3} //h2[text()='Book Your Transfer Today']

${BookYourTransferToday} (//h2[@class='sec\_\_title line-height-55'])[3]

${ValidateHeading4} //h2[text()='Latest on blogs']

${LatestOnBlogs} (//h2[@class='sec\_\_title line-height-55'])[4]

${ViewButton} css=a[class='theme-btn waves-effect']

${AfterViewButton} css=h2[class='sec\_\_title text-white']

${NeedHelpBox} css=i[class='la la-phone']

${SubscribeHeading} css=h2[class='sec\_\_title font-size-30 text-white']

${FooterEmailText} css=label[class='label-text text-white']

${FooterEmail} css=input[placeholder='Enter your email']

${SubscribeInput} css=h2[class='sec\_\_title font-size-30 text-white']

${SubscribeButton} //button[text()='Subscribe']

${FooterLogo} (//a[@href='https://phptravels.net/'])[2]

${FooterContactUs} //strong[text()='Contact Us']

${WaitAfterContactUs} css=h2[class='sec\_\_title text-white']

${FooterCompany} css=a[href='company']

${CompanyHeadings} css=h3[class='title stroke-shape']

${FooterAboutUs} (//a[text()='About Us'])[2]

${FooterTermsOfUse} (//a[text()='Terms of Use'])[2]

${FooterCookiesPolicy} //a[text()='Cookies policy']

${FooterPrivacyPolicy} //a[text()='Privacy Policy']

${FooterSupport} css=a[href='supprt']

${Footersupplier} (//a[text()='Become Supplier'])[1]

${FooterFAQ} (//a[text()='FAQ'])[2]

${FooterBookingTips} (//a[text()='Booking Tips'])[1]

${FooterHowToBook} (//a[text()='How to Book'])[2]

${FooterService} css=a[href='services']

${FooterFileAClaim} //a[text()='File a claim']

${FooterOffers} (//a[@href='https://phptravels.net/offers'])[3]

${OfferHeading} css=h2[class='sec\_\_title text-white']

${SignupSupplier} (//a[@href='https://phptravels.net/signup-supplier'])[2]

${SupplierSignupHeading} css=h5[class='modal-title title']

${FooterCareers} (//a[@href='https://phptravels.net/careers'])

${FooterFacebook} css=i[class='lab la-facebook']

${FacebookWait} css=div[class='x92rtbv x10l6tqk x1tk7jg1 x1vjfegm']

${BackFromFacebook} css=h1[class='x1heor9g x1qlqyl8 x1pd3egz x1a2a7pz']

${FooterTwitter} css=i[class='lab la-twitter']

${TwitterWait} (//span[@class='css-901oao css-16my406 r-poiln3 r-bcqeeo r-qvutc0'])[5]

${FooterYoutbe} css=i[class='lab la-youtube']

${YoutubeWait} (//img[@class='style-scope yt-img-shadow'])[2]

${FooterWhatsapp} css=i[class='lab la-whatsapp']

${WhatsappWait} css=a[id='action-button']

${FooterInstagram} css=i[class='lab la-instagram']

${InstagramWait} css=div[class='\_aagx']

${FooterPHPtravels} css=button[id='cookie\_stop']

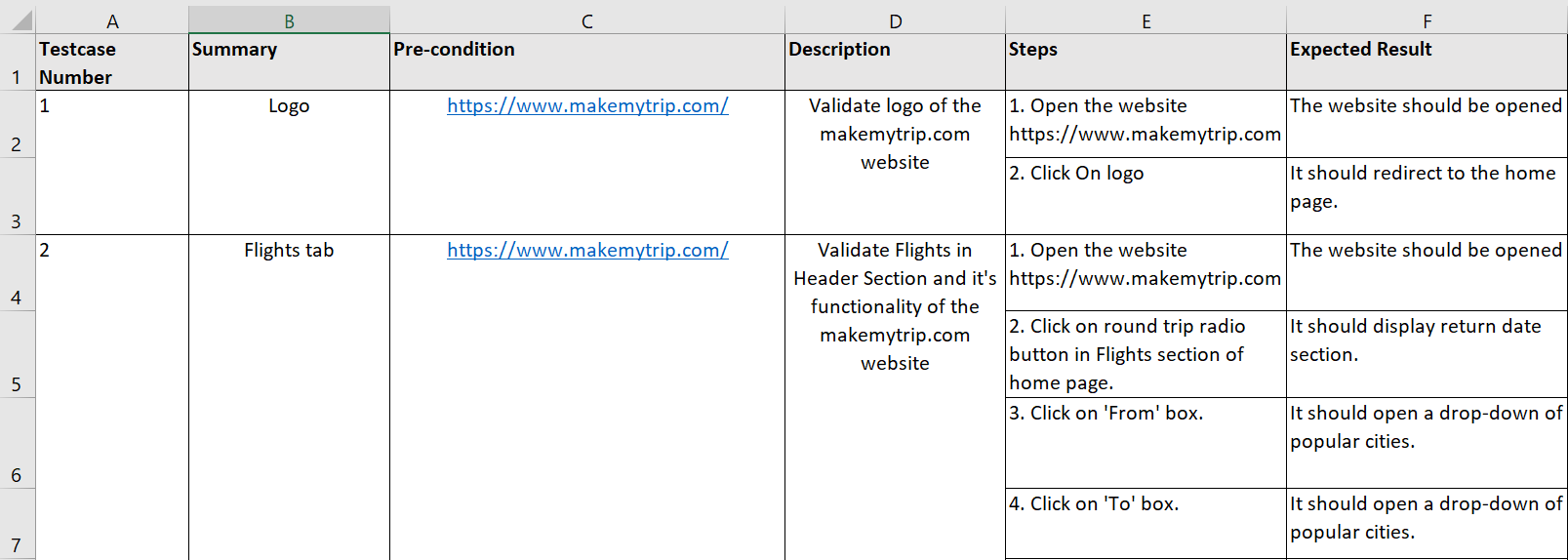
${PhptravelsPage} //a[@href='http://www.phptravels.com']

${PhpWait} css=h1[class='hero-text-1']

${Top-Up} css=i[class='la la-angle-up']

**9.2 Automation on Make My Trip Website**

🡪 Manual Testcases Of Make My Trip Website

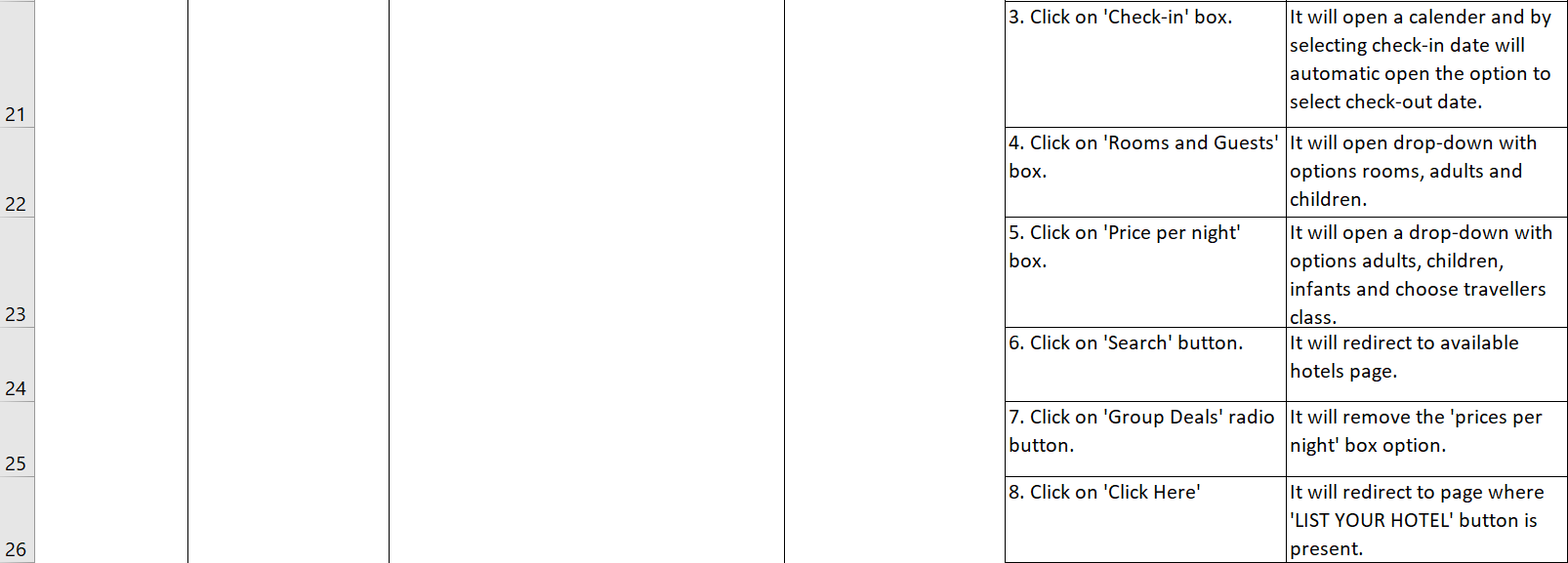


A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence



A screenshot of a computer screen

Description automatically generated with low confidence

A screen shot of a computer

Description automatically generated with low confidence

A picture containing text, screenshot, software, number

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, screenshot, number, font

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

A picture containing text, software, screenshot

Description automatically generated

A picture containing text, screenshot, software, display

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with low confidence

**Code:**

**Testcase file:**

\*\*\* Settings \*\*\*

Resource ../Common/testCase\_resources.robot

#Suite Setup Open The browser

#Suite TearDown Close Browser Session

#Author: Dhruvi.Virani

\*\*\* Test Cases \*\*\*

Validate MakeMyTrip Website

Open The Browser

Validate Logo

Validate Header Bar

Validate Flights Tab

Validate Hotels Tab

Validate Home Stays Tab

Validate Holiday Packages Tab

Validate Trains Tab

Validate Buses Tab

Validate Cabs Tab

Validate Forex Tab

Validate Charter Flights Tab

Validate Activities Tab

Validate Footer Section

Close The Browser

\*\*\* Keywords \*\*\*

Open The Browser

Set Selenium Timeout 10 seconds

Open Browser https://www.makemytrip.com/ } Chrome

Maximize Browser Window

Sleep 3

Wait Until Element Is Visible css=img[src='//imgak.mmtcdn.com/pwa\_v3/pwa\_hotel\_assets/header/mmtLogoWhite.png']

Validate Logo

Click Element css=img[src='//imgak.mmtcdn.com/pwa\_v3/pwa\_hotel\_assets/header/mmtLogoWhite.png']

Wait Until Element Is Visible css=img[src='//imgak.mmtcdn.com/pwa\_v3/pwa\_hotel\_assets/header/mmtLogoWhite.png']

Validate Header Bar

Click Element css=p[class='font10 userPerText appendTop3']

Switch Window New

Wait Until Element is Visible //span[text()='FILTERS']

Switch Window Main

Wait Until Element Is Visible css=img[src='//imgak.mmtcdn.com/pwa\_v3/pwa\_hotel\_assets/header/mmtLogoWhite.png']

Click Element //p[text()='Business Travel Solution']

Sleep 3

Wait Until Element Is Visible css=span[data-cy='myBizLoginClose']

Click Element css=span[data-cy='myBizLoginClose']

Click Element css=span[class='mbizUse\_\_close']

Wait Until Element Is Visible ${logo}

Click Element css=span[class='latoBold capText font11']

Click Element //button[text()='Apply']

Validate Flights Tab

Click Element //li[text()='Round Trip']

Click Element (//p[@class='code makeRelative'])[1]

Wait Until Element Is Visible //p[text()='Mumbai, India']

Click Element //p[text()='Mumbai, India']

Click Element (//p[@class='code makeRelative'])[1]

Wait Until Element Is Visible //p[text()='Bangkok, Thailand']

Click Element //p[text()='Bangkok, Thailand']

Click Element (//p[@class='blackText font20 code lineHeight36'])[2]

Click Element (//div[@class='dateInnerCell']/p[text()='1'])[1]

Click Element (//div[@class='dateInnerCell']/p[text()='15'])[1]

#Click Element //li[text()='First Class']

#Click Element css=button[class='primaryBtn btnApply pushRight']

Click Element css=a[class='primaryBtn font24 latoBold widgetSearchBtn ']

Wait Until Element Is Visible css=p[class='font24 blackFont whiteText appendBottom20 journey-title']

Go Back

Wait Until Element Is Visible ${logo}

Mouse Over //p[text()='Armed Forces ']

Mouse Over //p[text()='Student ']

Mouse Over //p[text()='Senior Citizen ']

Mouse Over //p[text()='Doctors & Nurses ']

execute javascript  window.scrollTo(0,500)

    Wait Until Element Is Visible   //font[text()='Offers']

    execute javascript  window.scrollTo(0,1200)

    Wait Until Element Is Visible   css=h3[class='appDnldCnt\_\_heading']

    Element Text Should Be      css=h3[class='appDnldCnt\_\_heading']     Download App Now !

execute javascript  window.scrollTo(1200,0)

Validate Hotels Tab

Click Element css=span[class='chNavIcon appendBottom2 chSprite chHotels']

Wait Until Element Is Visible //span[text()='City, Property name or Location']

Click Element //span[text()='City, Property name or Location']

Scroll Element Into View //span[text()='City, Property name or Location']

Wait Until Element Is Visible (//p[text()='Mumbai'])[1]

Click Element (//p[text()='Mumbai'])[1]

Click Element (//span[@class='lbl\_input latoBold appendBottom5'])[4]

Wait Until Element Is Visible //li[text()='₹2500-₹5000']

Click Element //li[text()='₹2500-₹5000']

Click Element css=button[id='hsw\_search\_button']

Wait Until Element Is Visible //p[text()='Showing Properties in Mumbai']

Go Back

Wait Until Element Is Visible ${logo}

execute javascript  window.scrollTo(0,500)

    Wait Until Element Is Visible   //font[text()='Offers']

    execute javascript  window.scrollTo(0,1200)

    Wait Until Element Is Visible   css=h3[class='appDnldCnt\_\_heading']

    Element Text Should Be      css=h3[class='appDnldCnt\_\_heading']     Download App Now !

execute javascript  window.scrollTo(1200,0)

Validate Home Stays Tab

Wait Until Element Is Visible ${logo}

Click Element //span[text()='Homestays']

Sleep 1

Click Element //span[text()='City, Property name or Location']

#Wait Until Element Is Visible css=div[class='hsw\_inputBox selectHtlCity ']

Click Element css=input[class='hsw\_inputField font30 lineHeight36 latoBlack']

Click Element css=input[id='checkin']

Click Element (//div[text()='15'])[1]

Click Element (//div[text()='25'])[1]

Click Element css=button[class='primaryBtn btnApply']

Wait Until Element Is Visible //p[text()='Select a Reason (optional)']

Click Element //p[text()='Select a Reason (optional)']

Wait Until Element Is Visible //li[text()='Leisure']

Click Element //li[text()='Leisure']

Click Element css=button[id='hsw\_search\_button']

Wait Until Element Is Visible //p[text()='Showing Properties in Goa']

Go Back

execute javascript  window.scrollTo(0,500)

Sleep 1

Wait Until Element Is Visible //p[text()='Find Your Wonderland']

execute javascript  window.scrollTo(0,2000)

    Wait Until Element Is Visible   //font[text()='Offers']

execute javascript  window.scrollTo(2000,0)

Validate Holiday Packages Tab

Wait Until Element Is Visible ${logo}

Click Element //span[text()='Holiday Packages']

Sleep 1

Click Element css=input[id='fromCity']

Wait Until Element Is Visible //li[text()='Chennai']

Click Element //li[text()='Chennai']

Click Element //span[text()='To City/Country/Category']

Wait Until Element Is Visible //div[text()='Kashmir']

Click Element //div[text()='Kashmir']

#Click Element css=span[class='appendBottom5 field-label down-arrow']

Click Element (//p[text()='30'])[1]

Sleep 1

Click Element //button[text()='APPLY']

execute javascript window.scrollTo(0,100)

Sleep 1

Click Element //div[@class='action-container']/button

Click Element css=button[class='primaryBtn font24 latoBlack widgetSearchBtn lineHeight29']

Wait Until Element is Visible //h1[text()='Kashmir']

Go Back

Wait Until Element Is Visible ${logo}

execute javascript window.scrollTo(100,0)

Sleep 1

Click Element //span[text()='Destinations']

Click Element //li[text()='Emerging']

Click Element //a[text()='Gujarat']

Switch Window New

Wait Until Element Is Visible css=h1[class='topHeading']

Switch window Main

Wait Until Element Is Visible ${logo}

Click Element //span[text()='Honeymoon']

Click Element //a[text()='Maldives']

Switch Window New

Wait Until Element Is Visible //h1[text()='Maldives']

Switch window Main

Wait Until Element Is Visible ${logo}

Click Element //span[text()='Featured']

Wait Until Element Is Visible //a[text()='Adventure']

Click Element //a[text()='Adventure']

Switch Window New

Wait Until Element Is Visible //h1[text()='Adventure']

Switch window Main

Wait Until Element Is Visible ${logo}

Click Element //span[text()='Travel Guidelines']

Click Element //li[text()='International']

Click Element //a[text()='International Travel Guidelines']

Wait Until Element Is Visible ${logo}

execute javascript  window.scrollTo(0,2000)

Wait Until Element Is Visible //h2[text()='Jackpot Deals on Top Selling Packages']

execute javascript  window.scrollTo(0,2500)

Wait Until Element Is Visible //h2[text()='International Destinations!']

execute javascript  window.scrollTo(0,2850)

Wait Until Element Is Visible //h2[text()='Explore the Unexplored']

execute javascript  window.scrollTo(0,3100)

Wait Until Element Is Visible   //font[text()='Offers']

execute javascript  window.scrollTo(3100,0)

Validate Trains Tab

Click Element //span[text()='Trains']

Sleep 1

Click Element //span[text()='From']

Wait Until Element Is Visible //span[text()='Kolkata Howrah Junction']

Click Element //span[text()='Kolkata Howrah Junction']

Wait Until Element Is Visible //span[text()='Delhi']

Click Element //span[text()='Delhi']

Click Element (//div[text()='12'])[1]

Click Element //li[text()='Third AC']

Click Element css=a[class='primaryBtn font24 latoBold widgetSearchBtn']

Wait Until Element Is Visible css=button[class=' primaryBtn font18 latoBlack widgetSearchBtn']

Go To https://www.makemytrip.com/

Wait Until Element Is Visible ${logo}

Click Element //span[text()='Trains']

Sleep 1

Click Element //span[text()='CHECK PNR STATUS']

Element Should Be Visible css=input[placeholder='Enter 10 digit PNR']

Click Element //span[text()='LIVE TRAIN STATUS']

Element Should Be Visible css=label[class='lbl\_input makeFlex column latoBold']

execute javascript  window.scrollTo(0,250)

    Wait Until Element Is Visible   //font[text()='Offers']

Validate Buses Tab

Click Element css=span[class='chNavIcon appendBottom2 chSprite chBuses']

Sleep 1

Click Element //span[text()='From']

Wait Until Element Is Visible //span[text()='Pune, Maharashtra']

Click Element //span[text()='Pune, Maharashtra']

Wait Until Element Is Visible //span[text()='Delhi, Delhi']

Click Element //span[text()='Delhi, Delhi']

Click Element (//div[text()='15'])[1]

Click Element css=button[id='search\_button']

Wait Until Element Is Visible css=button[class='primaryBtn font18 latoBlack widgetSearchBtn']

Go Back

Wait Until Element Is Visible ${logo}

Validate Cabs Tab

Click Element css=span[class='chNavIcon appendBottom2 chSprite chCabs']

Sleep 1

Click Element //li[text()='OUTSTATION ROUND TRIP']

Click Element //span[text()='From']

Wait Until Element Is Visible //span[text()='Bangalore']

Click Element //span[text()='Bangalore']

Click Element //span[text()='To']

Wait Until Element Is Visible //span[text()='Mumbai']

Click Element //span[text()='Mumbai']

Click Element //span[text()='DEPARTURE']

Click Element (//div[text()='11'])[1]

Click Element (//div[text()='26'])[1]

Click Element //a[text()='Search']

Wait Until Element Is Visible css=button[class='bluePrimarybtn widgetSearchBtn']

Go Back

Wait Until Element Is Visible ${logo}

Validate Forex Tab

Click Element //span[text()='Forex']

Wait Until Element Is Visible //p[text()='Global INR Card']

Click Element css=a[class='primaryBtn font24 latoBlack widgetSearchBtn']

Wait Until Element Is Visible css=a[class='navbar-brand navbar-logo']

Go Back

Wait Until Element Is Visible ${logo}

Validate Charter Flights Tab

Click Element css=span[class='chNavIcon appendBottom2 chSprite chCharterFlights']

Sleep 1

Click Element //li[text()='Round Trip']

Click Element css=a[class='primaryBtn font24 latoBold widgetSearchBtn']

Wait Until Element Is Visible //p[text()='Sorry, No Charters found for this search']

Go Back

Wait Until Element Is Visible ${logo}

Validate Activities Tab

Click Element //span[text()='Activities']

Wait Until Element Is Visible (//input[@placeholder='Activities, Tours or Destination'])[1]

Validate Footer Section

execute javascript window.scrollTo(0,3000)

Sleep 1

execute javascript window.scrollTo(0,5000)

Sleep 1

#Wait Until Element Is Visible css=span[class='twiiterIcon landingSprite']

Click Element css=span[class='twiiterIcon landingSprite']

Wait Until Element Is Visible css=svg[class='r-13v1u17 r-4qtqp9 r-yyyyoo r-16y2uox r-8kz0gk r-dnmrzs r-bnwqim r-1plcrui r-lrvibr r-lrsllp']

Sleep 1

Go To https://www.makemytrip.com/

Wait Until Element Is Visible ${logo}

execute javascript window.scrollTo(0,3000)

Sleep 1

execute javascript window.scrollTo(0,5000)

Sleep 1

Click Element css=span[class='facebookIcon landingSprite']

Wait Until Element Is Visible css=div[class='x92rtbv x10l6tqk x1tk7jg1 x1vjfegm']

Click Element css=div[class='x92rtbv x10l6tqk x1tk7jg1 x1vjfegm']

Wait Until Element Is Visible //h1[@class='x1heor9g x1qlqyl8 x1pd3egz x1a2a7pz']

Go Back

Wait Until Element Is Visible ${logo}

Close The Browser

Close Browser