

**A PROJECT REPORT**  
**ON**  
**AUTOMATION TESTING**  
*Project report submitted in partial fulfillment of the requirement for the degree of*  
**BACHELOR OF TECHNOLOGY**  
**IN**  
**ELECTRONICS AND COMMUNICATION ENGINEERING**

Submitted By

**Mehul Bandlish (171047)**

**UNDER THE SUPERVISION OF**  
**COGNIZANT TEAM**



**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,**  
**WAKNAGHAT**

## DECLARATION

I hereby declare that the work reported in the B.Tech Project Report entitled "**Automation Testing**" submitted at **Jaypee University of Information Technology, Waknaghat , India** is a true full proved record of my work done during the internship at **Cognizant**. I don't have submitted my work anywhere else for some other degree.



Mehul Bandlish (171047)

This is to bring in notice that the above mentioned statements made by the candidate are correctly verified by the concerned authorities and there are no copyright issues.

Cognizant Team

Date:

Head of the Department/Project Coordinator

## **ACKNOWLEDGEMENT**

I firstly thank God for giving me the strength and courage which were key features for the completion of the project. It acknowledges all the people who gave me their extreme support in making the project successful. I would like to thank our company POCs, trainers and coaches for their constant encouragement throughout the making of the project. I am also thankful to all my co-members in the company for their valuable support that helped me to complete the project. In the end, I would like to extend my thanks to my beloved parents for their support at each and every step of progress I made in the project work and continuous cooperation in any way possible for the betterment of the project.

# **AUTOMATION TESTING**

## **ABSTRACT**

Automation Testing , also known as Test Automation is a software testing technique which performs advanced testing using special automated testing tools to run a test case suite. On the other hand, Manual Testing is done by a person who executes each and every test step , which is time consuming ,in comparison to Automation Testing.

The goal of Automation is to enhance the performance of the test, by reducing the number of test cases to be run manually and not eliminating the Manual Testing completely. Test Automation is best method for increasing effectiveness , test coverage and enhance speed in software testing.

For data testing and scripting knowledge, I learnt SQL and vbScript. SQL is a database computer language designed for storing, manipulating, retrieving and managing the data , both in actual and relational database. SQL stands for Structured Query Language. Microsoft VBScript (Visual Basic Script) is a general purpose scripting language, which is developed by Microsoft and modeled on Visual Basic. VBScript is a primary scripting language for Quick Test Professional (QTP), which is an advanced automation tool, that helps testers to perform automated regression tests, for identifying the defects in test report. The actual results should match the expected results.

# INDEX

## CONTENTS

ABSTRACT	Page No.
<b>1.INRODUCTION.....</b>	<b>1</b>
1.1. What is Automation Testing?.....	1
1.2. Why Automation Testing?.....	1
1.3. Process Involved in Automation Testing.....	1
1.4. Types of Automation Testing.....	2
<b>2. LITERATURE REVIEW.....</b>	<b>3</b>
2.1.Functional Testing .....	3
2.1.1. Functional Testing vs Non-Functional Testing.....	3
2.1.2. Steps Involved .....	3
2.2. SQL.....	4
2.2.1. Syntax.....	4
2.2.2. Data Types in SQL.....	4
2.2.3. Keys in SQL.....	4
2.2.3.1. Primary.....	5
2.2.3.2. Foreign.....	5
2.2.3.3. Unique.....	5
2.3. VBScript.....	6
2.3.1. Formatting.....	6
2.3.2. Case Sensitivity.....	7
2.3.3. Comments in VBScript.....	7
2.3.4. Variables in VBScript.....	7

2.3.5. Operators in VBScript.....	8
2.3.6. Loops.....	9
2.3.6.1. For Loop.....	9
2.3.6.2. For..Each Loop.....	9
2.3.6.3. While..Wend Loop.....	10
2.3.6.4. Do..While Loop.....	10
2.3.6.5. Do..Until Loop.....	11
2.3.7. Strings.....	11
2.3.8. Date and Time Functions.....	12
2.3.9. Regular Expressions in VBScript.....	13
2.3.10. Error Handling in VBScript.....	14
<b>3. MICROFOCUS UNIFIED FUNCTIONAL TESTING.....</b>	<b>15</b>
3.1. Introduction to UFT.....	15
3.2. Create and Run Automated GUI tests.....	15
3.2.1. GUI tests and actions.....	16
3.2.2. Object Repositories.....	16
3.2.3. Adding steps to test.....	16
3.2.4. Run and analyze GUI tests.....	17
3.2.5. Parameterization.....	17
3.2.6. Checkpoints and Output Values.....	17
3.2.7. Functions and Function Library.....	18
<b>4. HANDS ON WITH UFT.....</b>	<b>19</b>
<b>5. RESULTS AND DISCUSSIONS.....</b>	<b>25</b>
<b>6. CONCLUSION.....</b>	<b>26</b>

## LIST OF FIGURES

1. Fig. 2.2.1. SQL SELECT, JOINS, WHERE CLAUSE.....	5
2. Fig. 2.2.2. CREATING DATA TABLE .....	6
3. Fig. 2.3.1. Logical Operators1.....	8
4. Fig. 2.3.2. Logical Operators2.....	8
5. Fig. 2.3.3. For Loop.....	9
6. Fig. 2.3.4. For..Each Loop.....	9
7. Fig. 2.3.5. While..Wend Loop.....	10
8. Fig. 2.3.6. Do..While Loop.....	10
9. Fig. 2.3.7. Do..Until Loop.....	11
10. Fig. 2.3.8. Replacing a string.....	11
11. Fig. 2.3.9. Ucase string.....	12
12. Fig. 2.3.10. Date and Time Functions 1.....	12
13. Fig. 2.3.11. Date and Time Functions 2.....	13
14. Fig. 2.3.12. Regular Expressions in VBScript.....	13
15. Fig. 2.3.13. Error Handling in VBScript.....	14
16. Fig. 3.1. Script.....	20
17. Fig. 3.2. Output.....	20
18. Fig. 3.3. Step1.....	21
19. Fig. 3.4. Step2.....	21
20. Fig. 3.5. Step3.....	22
21. Fig. 3.6. Step4.....	22
22. Fig. 3.7. Step5.....	23
23. Fig. 3.8. Step6.....	23
24. Fig. 3.9. Object Repository.....	24

# CHAPTER – 1

## INTRODUCTION

### 1.1. WHAT IS AUTOMATION TESTING?

Automation Testing , also known as Test Automation is a software testing technique which performs advanced testing using special automated testing tools to run a test case suite. On the other hand, Manual Testing is done by a person who executes each and every test step , which is time consuming ,in comparison to Automation Testing.

### 1.2. WHY AUTOMATION TESTING?

Test Automation is one of the best method to boost up the efficiency, covers wide range of test cases in short duration of time, also increases the execution speed while testing the softwares. It is time consuming and economical technique. Its very easy to use and give all the results in handy, while on other hand it is a tough job to test for multilingual sites manually. Automation Testing doesn't require a person to write test cases, that is you can run automated tests unattended. Manual Testing can sometimes become boring and thus prone to error.

### 1.3. PROCESS INVOLVED IN AUTOMATION TESTING:

There are some steps that are followed in Automation Process

1. Selection of test tool: depending on the technology and testing application.
2. Defining an Automation scope: Technical feasibility, how complex the test cases are.
3. Plan, Design and Develop: selection of appropriate Automation tools, framework design and deliverables
4. Execute test cases: writing scripts, input data is entered, reports are generated after the complete execution of tests.
5. Maintain tests: checking the proper working of new functionalities added to software, added to scripts and maintained for enhancing the effectiveness of automation scripts.



#### **1.4. TYPES OF AUTOMATION TESTING:**

- Smoke Testing: It determines that the software deployed is ready to use or not. It is like a confirmation sent to QA department to continue with the upcoming stages in software testing.
- Unit Testing: Here, all the individual components involved in making of software are tested. It is a validation stage that validates the each line as expected, it is done by developers, during the coding phase.
- Integration Testing: Combining the individual modules and arrange all units in a group.
- Functional Testing: Tests functionality of each part, by providing a valid input and verifying the output obtained against their functionality.
- Regression Testing: It is re assuring that the software deployed and tested previously, still performs all the tasks after some changes are made.
- Black and White box testing: The Black box testing tests the overall behavior of the software while White box testing tests the insights of operations built in system.

## **CHAPTER - 2**

### **LITERATURE REVIEW**

#### **2.1. FUNCTIONAL TESTING**

Its main goal is validating the functionalities involved in software system, against their specific requirements. We provide the valid input and verify the output, the actual result should match the expected result.

Functional testing mostly involves the Black box testing and its features are not concerned with the actual source code involved in building of an application. It keeps a check on few things like User Interface, Database involved, APIs to be tested, MFA security and some other functionalities are tested. This test can be perform either way, manually or Automation testing.

##### **2.1.1. FUNCTIONAL TESTING VS NON FUNCTIONAL TESTING**

Functional Testing tests the system using functional specification provided by the user and verify them against the specific functionalities. On the other hand, Non-Functional Testing keeps a check on features like system performance, system reliability and other non-functionalities of the system.

The Functional testing is to be executed first, followed by Non-Functional Testing. Functional testing gives the information about the product and what actually a product does, and Non-Functional testing tells how well the product works.

Some examples of Functional Testing are: Unit testing, Black and White Box Testing, Integration Testing, Smoke Testing, Regression Testing,etc.

Some examples of Non-Functional Testing are: Performance Testing, Compatibility Testing, Security Testing, etc.

##### **2.1.2. STEPS INVOLVED**

1. Identification of test input (input data)
2. Computation of expected results with the valid test input data

3. Executing the test cases
4. Comparing the actual results with the expected results

## **2.2. SQL**

SQL stands for Structured Query Language, which is used for storing, manipulating, retrieving the data from actual or relational database, in case the data table entities are in relation to each other.

### **2.2.1. SYNTAX**

There are several important terms used in SQL syntax like:

1. Clauses: these are used while specifying particular data from database. Eg: WHERE Clause, SET Clause, UPDATE Clause
2. Expressions: to produce scalar values
3. Queries: for retrieving the data on the basis of some specific criteria. It is a key element in SQL.
4. Statements: used for controlling program flow, acts like connectors.

### **2.2.2. DATA TYPES IN SQL**

The data types in SQL is broadly classified under 3 categories:

1. Predefined data types: Character (CHAR), varying character (VARCHAR), Binary (BINARY), Numeric Types (FLOAT, INTEGER), Date and time types (DATE, TIME), Boolean, XML and JSON.
2. Constructed data types: ARRAY, REF (reference), ROW
3. User defined data types

### **2.2.3. KEYS IN SQL**

There is a huge amount of data stored in multiple tables, together forming a database. Also, each table might contain number of entries, rows. Among those number of rows, most of the

data is redundant and duplicate, thus we need to manage those records and find out the valuable data present inside a database.

An SQL key is the solution to this particular problem, which is either a single column or attribute or a group of columns that identifies the unique entries in a data table and filter out the valuable data we need for processing. Thus, it ensures that there are no repetitive rows in a data table and it helps to establish the relation between those multiple tables.

### 2.2.3.1. PRIMARY

It is a key in Relational database that is unique (like an id, telephone number) for each record.

There is only one primary key associated with a relational database. It cannot accept any NULL value.

### 2.2.3.2. FOREIGN

It is a column or a group of columns in Relational database that links data between the tables.

### 2.2.3.3. UNIQUE

It is similar to primary key, but it can accept only one NULL value, but no duplicate values.

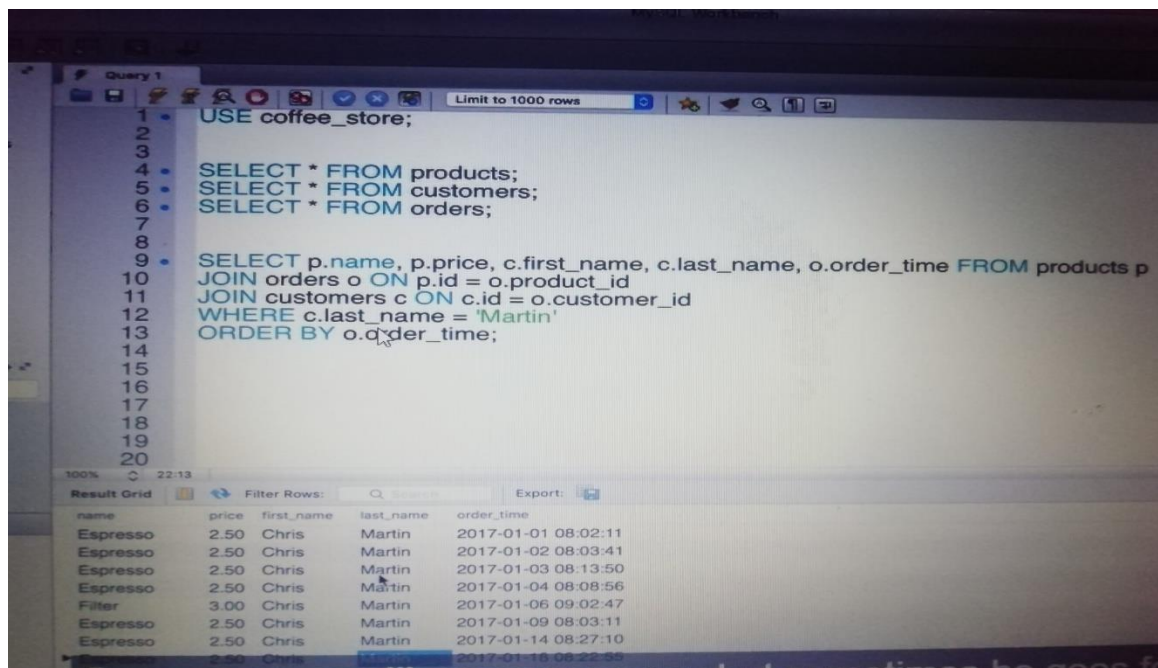


Fig. 2.2.1. SQL SELECT, JOINS, WHERE CLAUSE

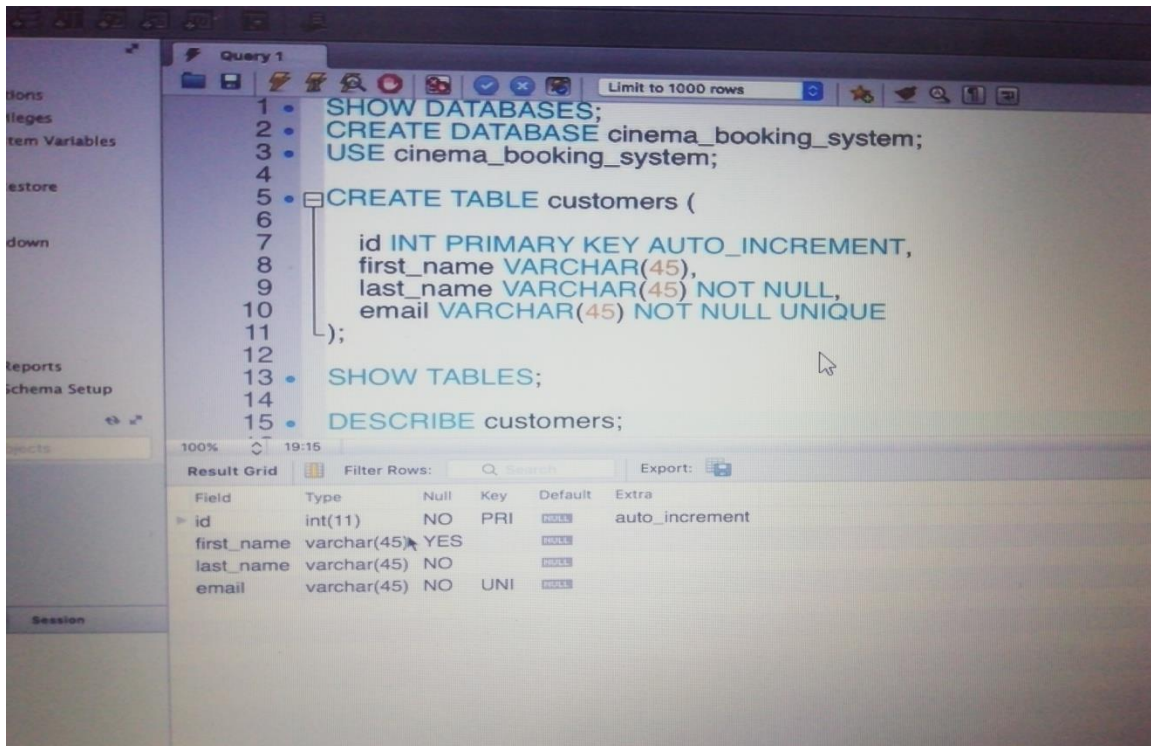


Fig. 2.2.2. CREATING DATA TABLE

## 2.3. VBSCRIPT

Microsoft VBScript (Visual Basic Script) is a general purpose scripting language, which is developed by Microsoft and modeled on Visual Basic. VBScript is a primary scripting language for Quick Test Professional (QTP), which is an advanced automation tool, that helps testers to perform automated regression tests, for identifying the defects in test report.

### 2.3.1. FORMATTING

In VBScript, no semicolon is used for terminating a statement, unlike other programming scripts. We use colons when more than one line of script is to be written in a line, thus colons are used as line separators.

```

<scripting language = "vbscript" type = "texttype ">
  variableX = 5 : variableY = 10
</script>

```

### 2.3.2. CASE SENSITIVITY

VBScript is case – insensitive scripting language.

Eg: option explicit, Option Explicit, OPTION EXPLICIT have the same meaning in this scripting language.

### 2.3.3. COMMENTS IN VBSCRIPT

We use comments as a key points, which are ignored by the interpreter during the execution of the program. There are mainly 2 ways to denote comments in VBScript.

1. Using single quote (') at the beginning of any statement:

```
<scripting language = "vbscript" type = "texttype">  
  <!--  
    ' Happy Learning  
    ' I welcome you all  
  //- >  
</script>
```

2. Using the keyword “REM” at the beginning of the statement

```
<scripting language = "vbscript" type = "texttype">  
  <!--  
    REM Happy Learning  
    REM I welcome you all  
  //- >  
</script>
```

### 2.3.4. VARIABLES IN VBSCRIPT

A variable is an entity that holds value in it, which can be changed while executing a script.

VBScript has only one data type called Variant.

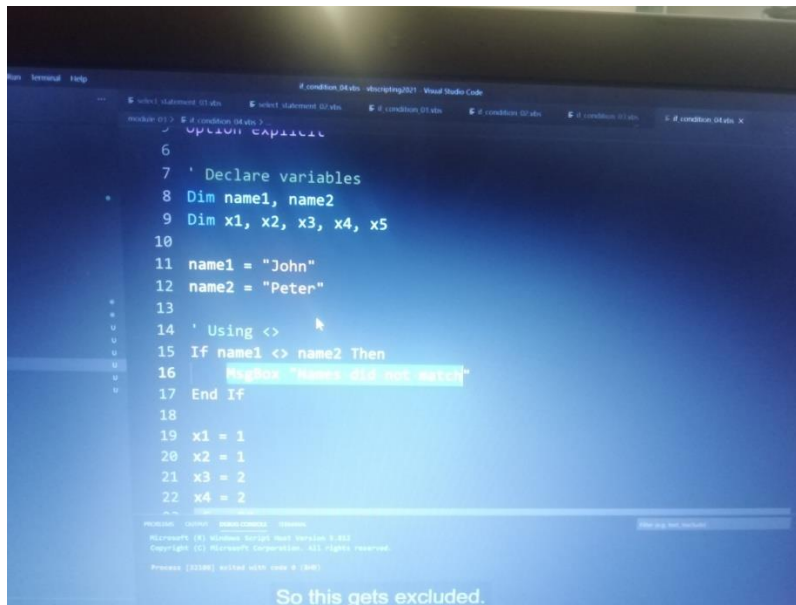
Variables are declared using a keyword “Dim”, which means dimension. Variables that are declared using this keyword are available to all the functions and procedures inside a single script.

### 2.3.5. OPERATORS IN VBSCRIPT

Operators are used to perform any mathematical or logical functions.

There are mainly 4 types of operators in VBScript:

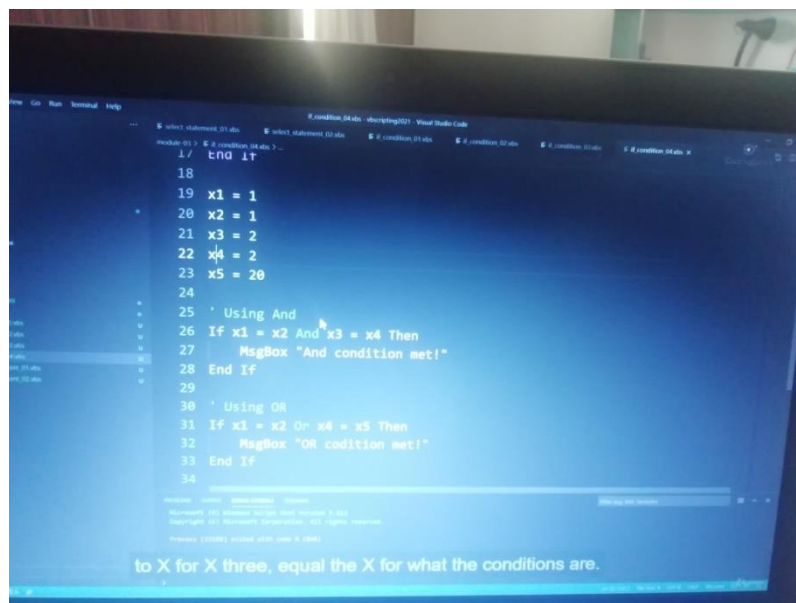
- Arithmetic : +,-,\*,/,%,^
- Comparison: ==,<>,>,<,>=,<=
- Logical or Relational: AND, OR, NOT, XOR
- Concatenation: +,&



```
6  
7 ' Declare variables  
8 Dim name1, name2  
9 Dim x1, x2, x3, x4, x5  
10  
11 name1 = "John"  
12 name2 = "Peter"  
13  
14 ' Using <>  
15 If name1 <> name2 Then  
16     MsgBox "names did not match!"  
17 End If  
18  
19 x1 = 1  
20 x2 = 1  
21 x3 = 2  
22 x4 = 2
```

So this gets excluded.

Fig. 2.3.1. Logical operators1



```
18  
19 x1 = 1  
20 x2 = 1  
21 x3 = 2  
22 x4 = 2  
23 x5 = 20  
24  
25 ' Using And  
26 If x1 = x2 And x3 = x4 Then  
27     MsgBox "And condition met!"  
28 End If  
29  
30 ' Using OR  
31 If x1 = x2 Or x4 = x5 Then  
32     MsgBox "OR condition met!"  
33 End If  
34
```

to X for X three, equal the X for what the conditions are.

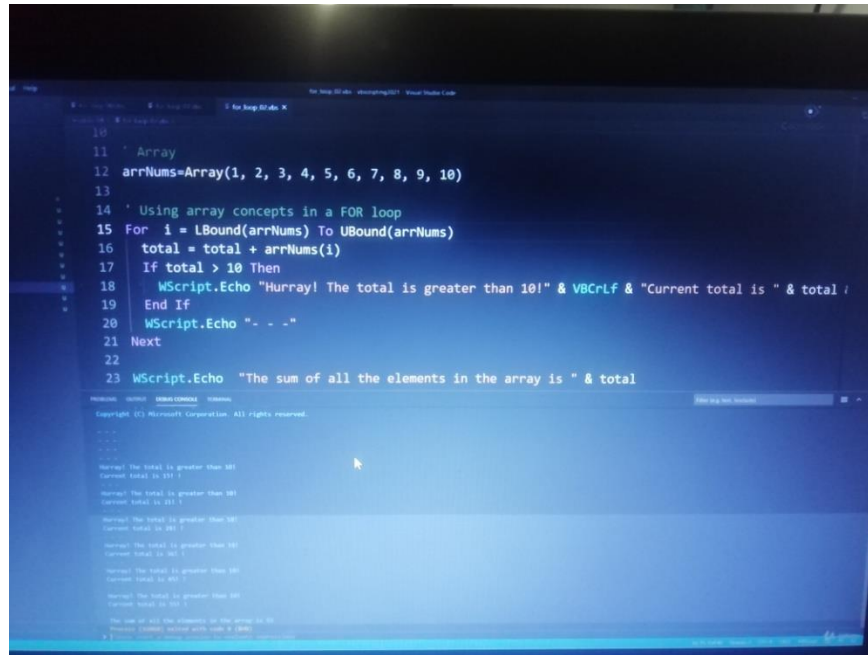
Fig. 2.3.2. Logical Operators2

### 2.3.6. LOOPS

Loops are used to repeat a number of steps in a program until a desired condition is fulfilled.

There are main 5 types of loops used in VBScript:

#### 2.3.6.1. FOR LOOP

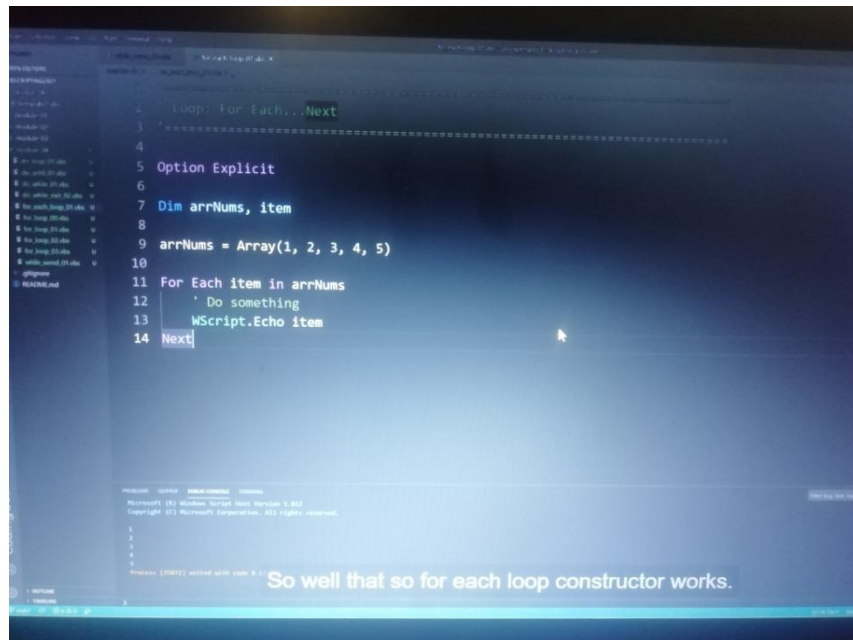


```
10  
11 ' Array  
12 arrNums=Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)  
13  
14 ' Using array concepts in a FOR loop  
15 For i = LBound(arrNums) To UBound(arrNums)  
16     total = total + arrNums(i)  
17     If total > 10 Then  
18         WScript.Echo "Hurray! The total is greater than 10!" & vbCrLf & "Current total is " & total /  
19     End If  
20     WScript.Echo "- -"  
21 Next  
22  
23 WScript.Echo "The sum of all the elements in the array is " & total
```

The screenshot shows a VBScript code editor with a dark blue background. The code defines an array of numbers from 1 to 10 and uses a FOR loop to iterate through each element. Inside the loop, the current element is added to a running total. If the total exceeds 10, a message is printed. After the loop, the final sum is printed. The output window at the bottom shows the execution results, including the 'Hurray!' messages and the final sum of 55.

Fig. 2.3.3. For Loop

#### 2.3.6.2. FOR..EACH LOOP



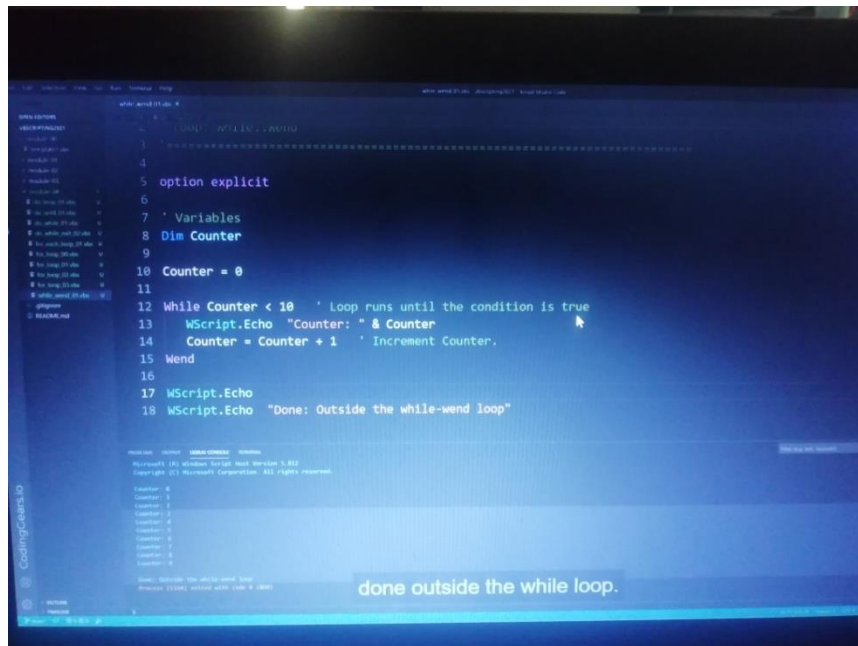
```
4 ' Loop: For Each...Next  
5  
6 Option Explicit  
7 Dim arrNums, item  
8  
9 arrNums = Array(1, 2, 3, 4, 5)  
10  
11 For Each item in arrNums  
12     ' Do something  
13     WScript.Echo item  
14 Next
```

The screenshot shows a VBScript code editor with a dark blue background. The code uses a FOR..EACH loop to iterate through an array of numbers from 1 to 5. Inside the loop, each item is printed. The output window at the bottom shows the execution results, including the 'So well that so for each loop constructor works.' message.

Fig 2.3.4. For..Each Loop



### 2.3.6.3 WHILE..WEND LOOP

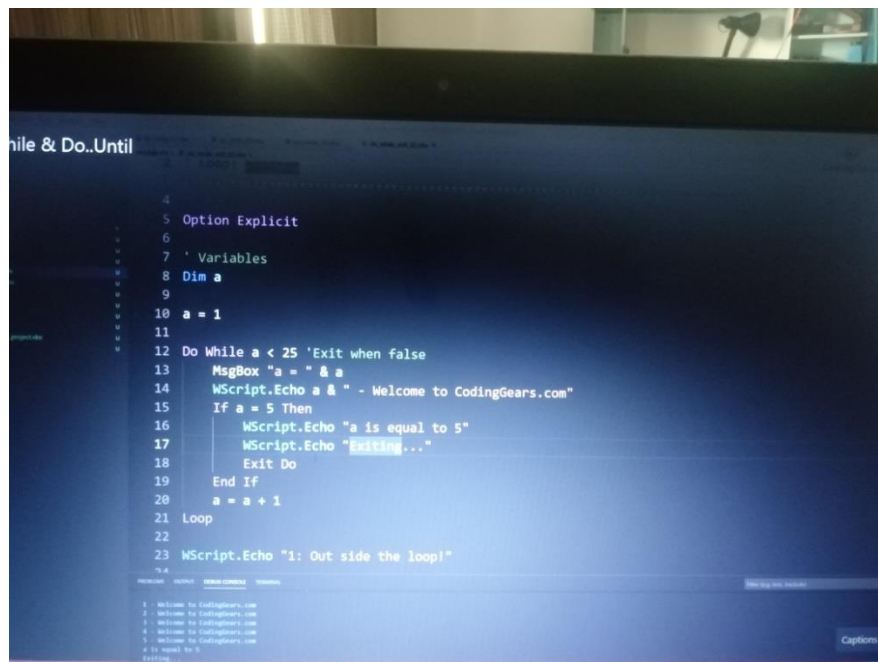


```
1 ' Loop "While..Wend"  
2 .....  
3 .....  
4 .....  
5 option explicit  
6 .....  
7 ' Variables  
8 Dim Counter  
9 .....  
10 Counter = 0  
11 .....  
12 While Counter < 10 ' Loop runs until the condition is true  
13 WScript.Echo "Counter: " & Counter  
14 Counter = Counter + 1 ' Increment Counter.  
15 Wend  
16 .....  
17 WScript.Echo  
18 WScript.Echo "Done: Outside the while-wend loop"
```

done outside the while loop.

Fig. 2.3.5. While..Wend Loop

### 2.3.6.4. DO WHILE LOOP

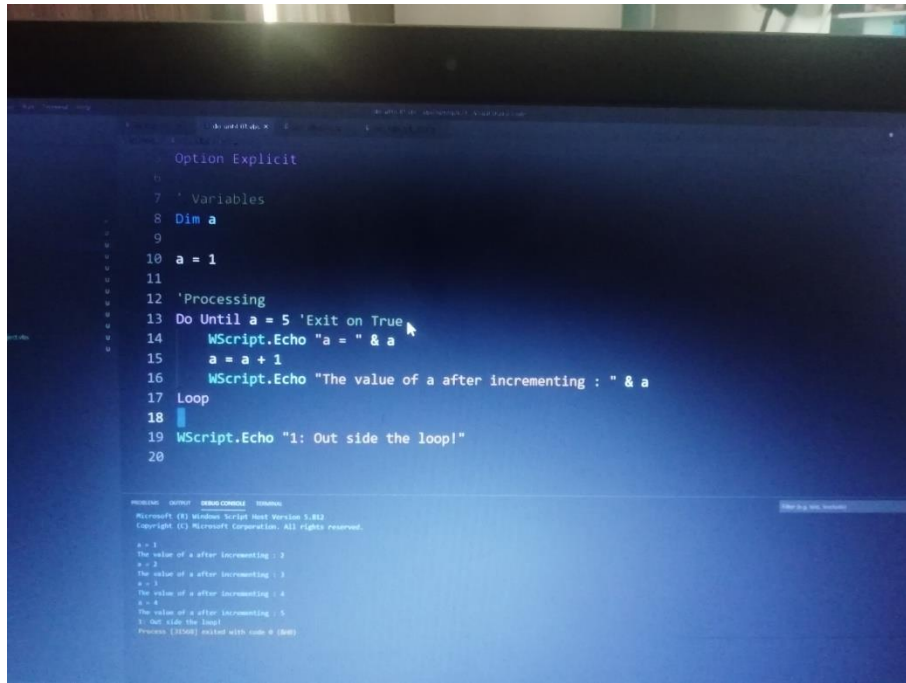


```
4 .....  
5 Option Explicit  
6 .....  
7 ' Variables  
8 Dim a  
9 .....  
10 a = 1  
11 .....  
12 Do While a < 25 'Exit when false  
13 MsgBox "a = " & a  
14 WScript.Echo a & " - Welcome to CodingGears.com"  
15 If a = 5 Then  
16 WScript.Echo "a is equal to 5"  
17 WScript.Echo "Exiting..."  
18 Exit Do  
19 End If  
20 a = a + 1  
21 Loop  
22 .....  
23 WScript.Echo "1: Out side the loop!"
```

1 - Welcome to CodingGears.com  
2 - Welcome to CodingGears.com  
3 - Welcome to CodingGears.com  
4 - Welcome to CodingGears.com  
5 - Welcome to CodingGears.com  
6 - a is equal to 5  
7 - Exiting  
8 - 1: Out side the loop!

Fig. 2.3.6. Do..While Loop

### 2.3.6.5. DO UNTIL LOOP



```
Option Explicit
1
7 ' Variables
8 Dim a
9
10 a = 1
11
12 'Processing
13 Do Until a = 5 'Exit on True
14     WScript.Echo "a = " & a
15     a = a + 1
16     WScript.Echo "The value of a after incrementing : " & a
17 Loop
18
19 WScript.Echo "1: Out side the loop!"
20
```

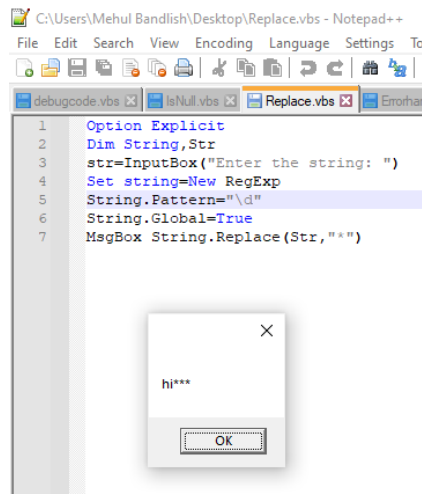
Microsoft (R) Windows Script Host Version 5.812  
Copyright (C) Microsoft Corporation. All rights reserved.

```
a = 1
The value of a after incrementing : 2
a = 2
The value of a after incrementing : 3
a = 3
The value of a after incrementing : 4
a = 4
The value of a after incrementing : 5
1: Out side the loop!
Process [10588] exited with code 0 (0x0)
```

Fig. 2.3.7. Do..Until Loop

### 2.3.7. STRINGS

Strings can be defined as the sequence of characters like alphabets, numbers, special characters and combination of all these. A variable enclosed within double quotes is considered as a string. Some of the predefined string methods used in VBScript are InStr, Lcase, Ucase, Mid, LTrim, Rtrim, Trim, Len, Replace, StrComp, StrReverse, etc.



```
C:\Users\Mehul Bandish\Desktop\Replace.vbs - Notepad++
File Edit Search View Encoding Language Settings To
debugcode.vbs x leNull.vbs x Replace.vbs x Errorhan
1 Option Explicit
2 Dim String,Str
3 str=InputBox("Enter the string: ")
4 Set string=New RegExp
5 String.Pattern="\d"
6 String.Global=True
7 MsgBox String.Replace(Str,"*")
```

hj\*\*\*

Fig. 2.3.8. Replacing a string

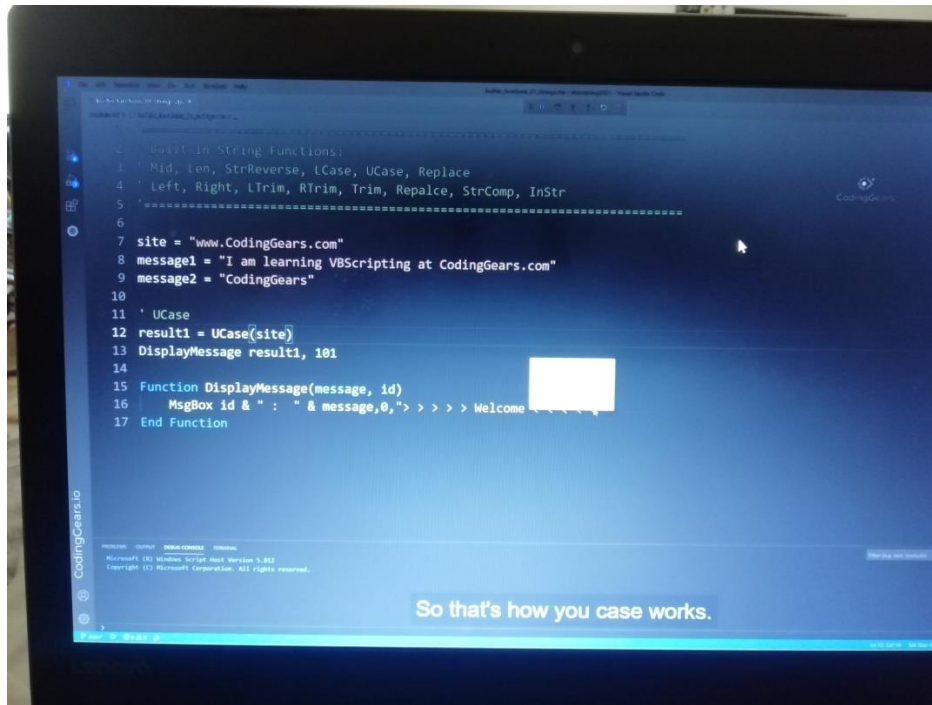


Fig. 2.3.9. Ucase string

### 2.3.8. DATE AND TIME FUNCTIONS

We use VBScript Date and Time functions for converting date and time from one format to another format, or to express the date and time in a particular format. There are 5 standard date and time functions. Although we can customize them to form many others, i.e, user defined D&T functions.

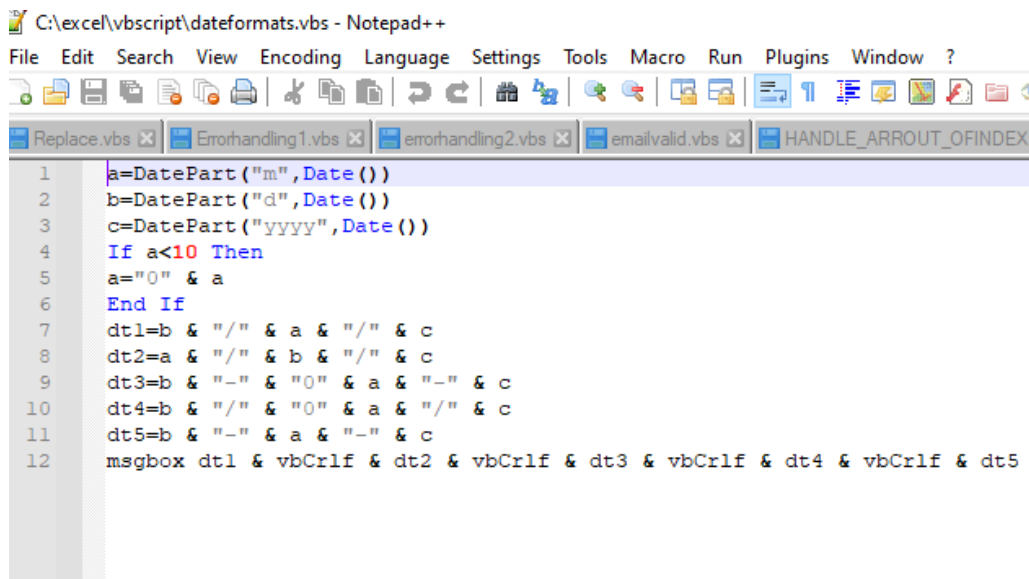


Fig. 2.3.10. Date and Time Functions 1

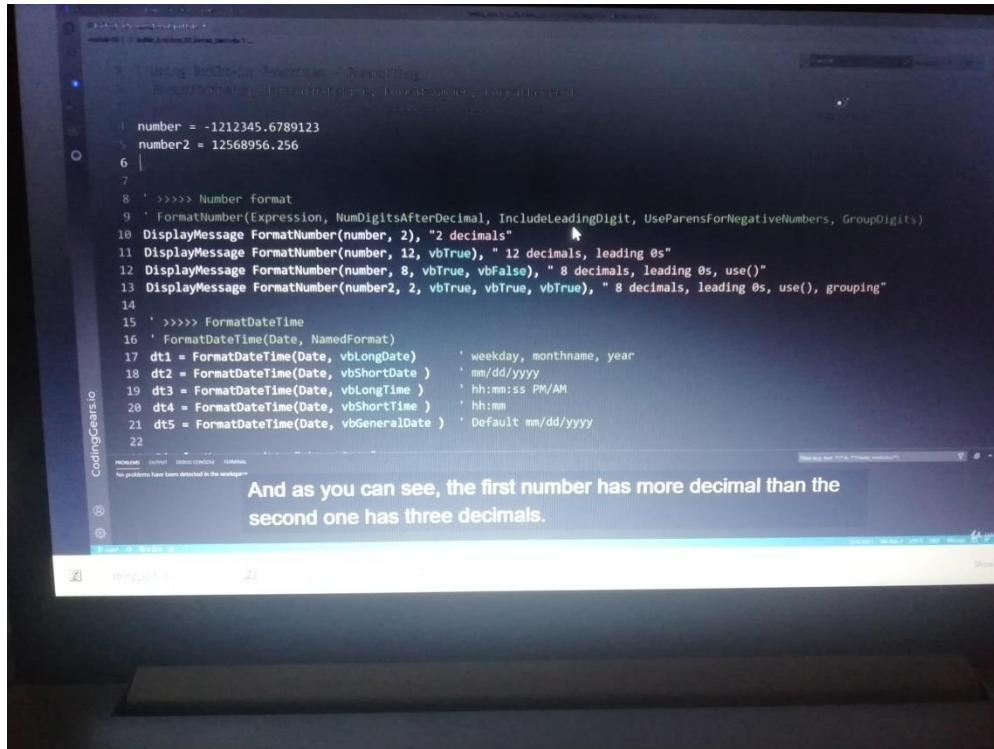


Fig. 2.3.11. Date and time Functions 2

### 2.3.9. REGULAR EXPRESSIONS IN VBSCRIPT

It refers to the pattern formed by characters, mainly used for replacing or searching, in order to match a desired strings. It helps a developer to extract the useful characters on the basis of some condition and perform the necessary operation on them. RegExp object is used to match the string pattern.

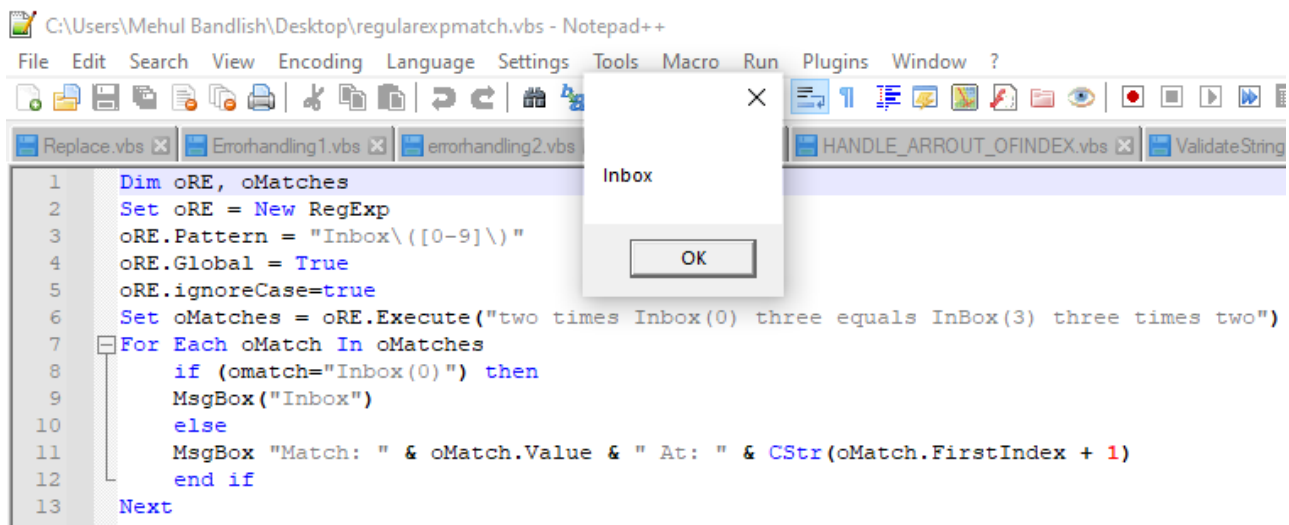
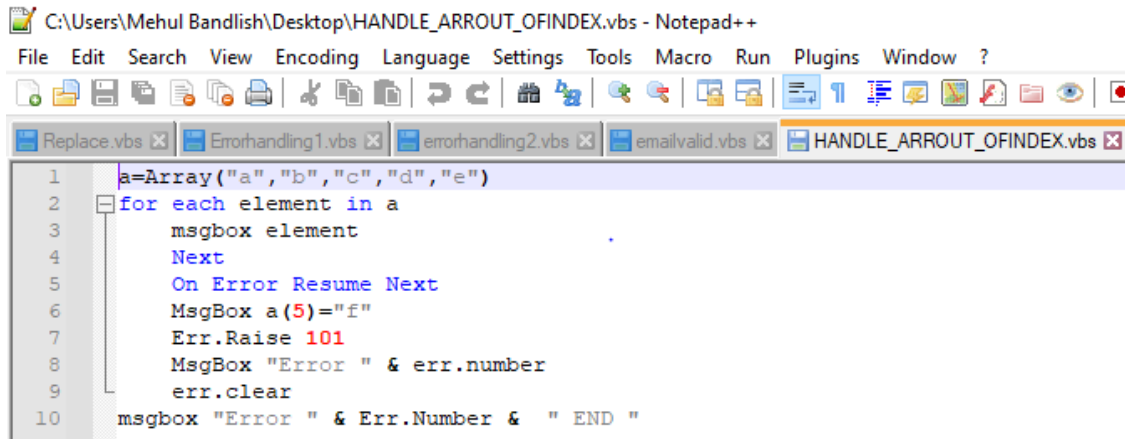


Fig. 2.3.12. Regular Expression in VBScript

### 2.3.10. ERROR HANDLING IN VBSCRIPT

Error Handling is defined as how a program behaves, responses to any error that arise in it. It is basically the way to detect, tackle the errors and anticipate accordingly in any software application. The syntax used in Error handling is “On Error”..”Perform the particular step”.



```
C:\Users\Mehul Bandlish\Desktop\HANDLE_ARROUT_OFINDEX.vbs - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
Replace.vbs x Errorhandling1.vbs x errorhandling2.vbs x emailvalid.vbs x HANDLE_ARROUT_OFINDEX.vbs x
1 a=Array("a","b","c","d","e")
2 for each element in a
3     msgbox element
4     Next
5     On Error Resume Next
6     MsgBox a(5)="f"
7     Err.Raise 101
8     MsgBox "Error " & err.number
9     err.clear
10 msgbox "Error " & Err.Number & " END "
```

Fig. 2.3.13. Error Handling in VBScript

## **CHAPTER – 3**

### **MICROFOCUS UNIFIED FUNCTIONAL TESTING**

#### **3.1. INTRODUCTION TO UFT**

Microfocus Unified Functional testing, now called Quick Test Professional (QTP), is a software which does functional and regression automation testing for variety of software applications environment. UFT uses VBScript scripting language for specifying test cases, object manipulation and end to end automation for testing. UFT was first found by Mercury Interactive. The Mercury used Win Runner, Test Director and Load Runner. After that when HP acquired Mercury, it named the following terms as Quick Test Professional, Quality center and Load Runner. Finally when this division was sold to Micro Focus, it is named as Unified Functional testing, ALM (Application Lifecycle Management) and Load Runner. This is how UFT was evolved.

#### **3.2. CREATE AND RUN AUTOMATED GUI TESTS**

To create a GUI test, it involves number of steps:

1. To create object repository which contains the test objects for all the objects used in your application.
2. Creating functions to be used in your test inside function library.
3. Recording and adding steps in your action script .
4. Adding checkpoints for validating any particular text or image in your application using various checkpoints in UFT like bitmap, text, standard, database, xml checkpoints.
5. Parameterizing the test object values to check the response, how application respond to different input values entered inside application.
6. Running and analyzing the results once all the test cases pass.

### **3.2.1. GUI TESTS AND ACTIONS**

There are 2 actions named Internal and External Actions. Internal action is stored in the local test, while external action is used as a reference to an action , where we call the particular action stored in different test.

When you make a call to a new action, its default settings is Reusable, which enables the user to call the specific action from any test. In case, you want to prevent some action from being used in other tests, you can go to settings and make the action non reusable.

There are 3 ways to invoke an action to a test :

1. Call to new action: It creates a new, blank page in your test where you can write an action.
2. Call to copy of an action: It inserts a copy of existing action in your test, where you can modify that test according to your need .
3. Call to an existing action: It inserts a call to an existing action, but here you cannot do any modifications in the already existing action.

### **3.2.2. OBJECT REPOSITORIES**

The test objects used to test your application, once collected is then stored in object repository with the properties and values of that object. It chooses a unique name for your test object.

There are 2 types of object repositories: Shared Object Repository and Local Object Repository

The test objects in shared object repository can be used in any other test, by associating the repositories. Local object repository contains tests objects that are localized to specific action. It cannot be shared with other actions. In UFT, the default object repository is local object repository.

### **3.2.3. ADDING STEPS TO TESTS**

You have to tell the UFT that what actions to be performed on the tests objects. The UFT then loops those actions on run time objects in your application.

This can be achieved through various methods like Keyword View, Data Driver,Editor, Recording, Step Generator.

### **3.2.4. RUN AND ANALYZE GUI TESTS**

For UFT to open your application through a URL, you can configure the record and run settings in the record dropdown and paste the URL there to invoke your application. The Other way to run an application is a command , `systemutil.Run "Browser name", "URL"`.

Once the application is open and you recorded the steps, or wrote a script in a editor window, the results are displayed in a separate window telling the status like Pass, Fail and also it gives the warnings for each step in detail (in case warnings arise).

### **3.2.5. PARAMETERIZATION**

You can parameterize any data present in a data table with the help of keyword view, or data driver. Parameterization allows you to pick different values for runtime objects. It reduces time and effort. Thus it is an important feature in UFT. After parameterizing, you can add subsequent data inside your data table like excel sheet and perform the tasks as required for each and every iteration in data sheet.

### **3.2.6. CHECKPOINTS AND OUTPUT VALUES**

We add or insert the checkpoints to validate, to see that your application is running absolutely fine. It is a separate step in the overall test flow. We can use the output values to obtain a value generated by a particular step or object and use this value in further stepp of your application.

There are various checkpoints in UFT : Standard Checkpoint which checks the values of the object's properties, image or bitmap checkpoint which checks the image is present or not at that particular time instant, text checkpoint which checks the text string is properly displayed at the appropriate place inside an application, text area checks that the text string is displayed within a desired location, and various others checkpoints are present in UFT . We use those checkpoints whenever and wherever necessary.

Some of the output values in UFT: Standard output value takes the value of object present in your application and stores it , text and text area output value takes the text or text area of an object in the application, XML takes the output of all the elements that are present in the XML document.



### **3.2.7. FUNCTIONS AND FUNCTIONS LIBRARY**

There are many built in functions present in UFT which helps us in testing any application. While if in case, you have to test some other functionalities that are not present in built in functions, you can create user defined functions for that task. You can then save the defined function in function library file, and then you can make a function call whenever you need to perform that task.

A function is set of code that performs a certain task for which there is no direct method available. Once the functions are created and stored in function library, you can call them by invoking it in editor window. These function library is similar to object repository for user defined functions.

# CHAPTER – 4

## HANDS ON WITH UFT

Here, we will see the test application and all the things we discussed in Microfocus UFT.

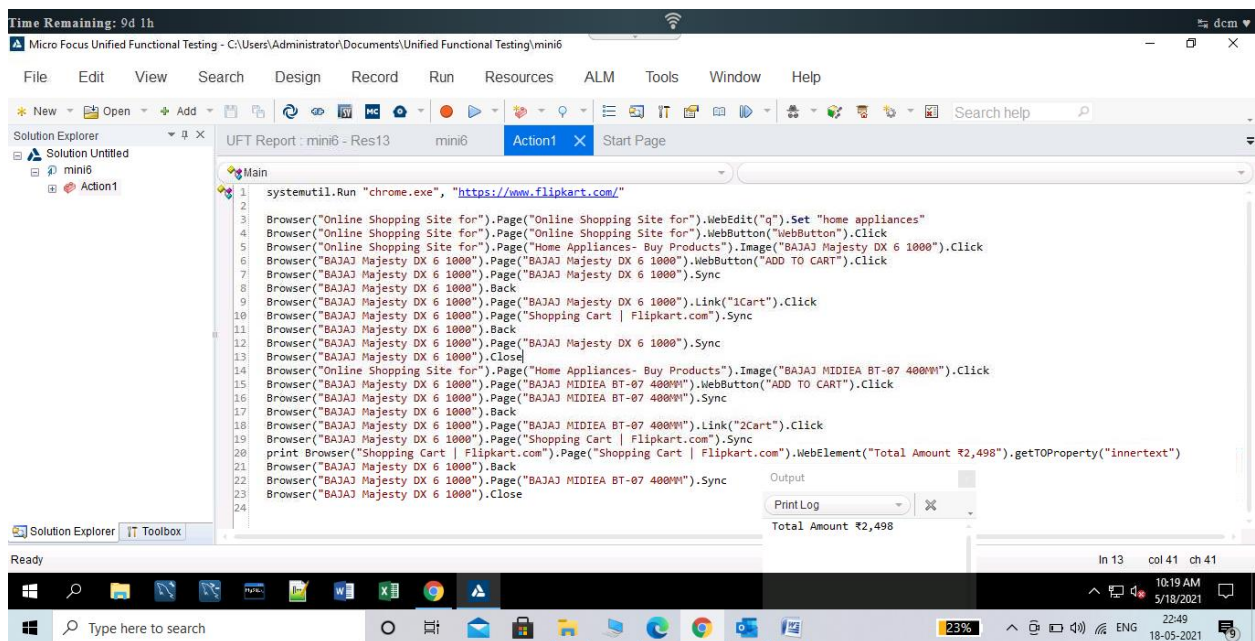
Case Study: ONLINE SHOPPING AUTOMATION

### Problem Statement:

Automate Add to Cart functionality and Order amount validation of Online shopping Website

Use the website which has below components

1. Online shopping website dealing with Home appliances
2. Online shopping website having Add to Cart facility



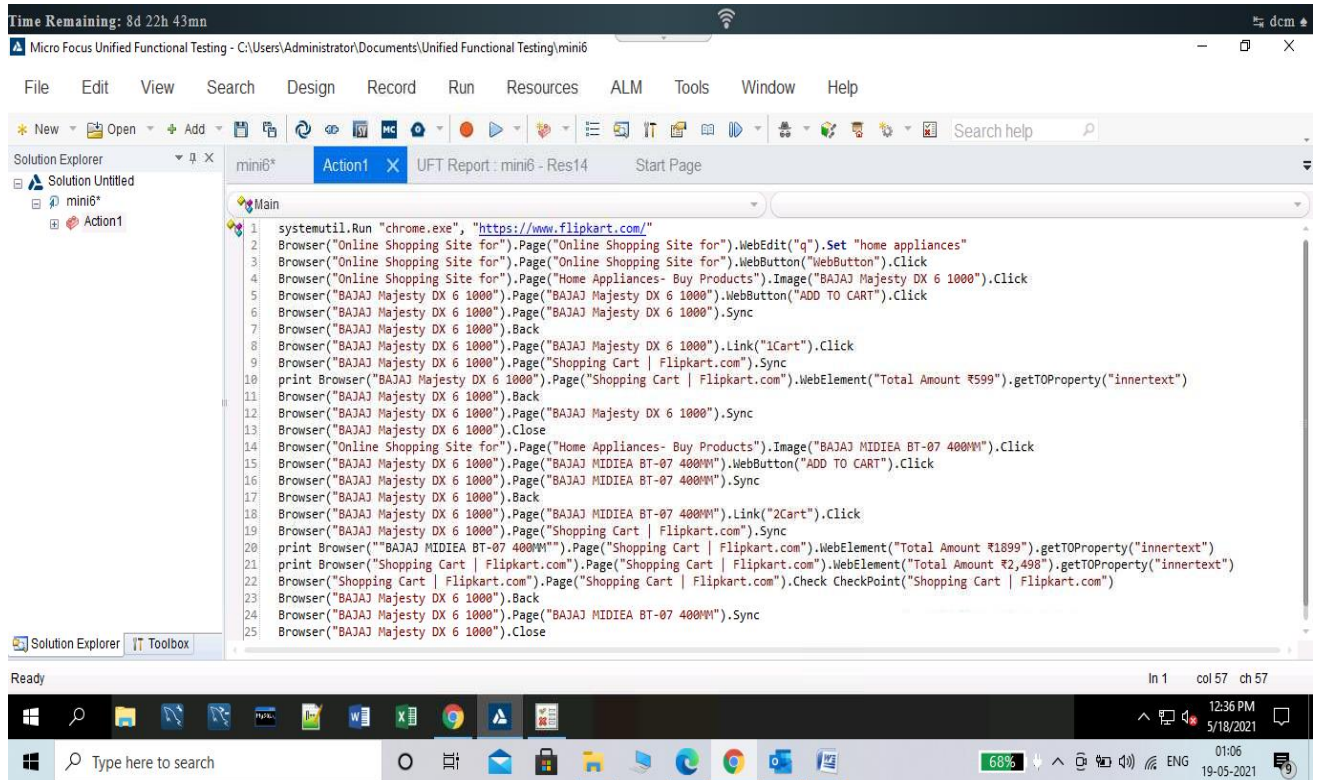


Fig. 3.1. Script

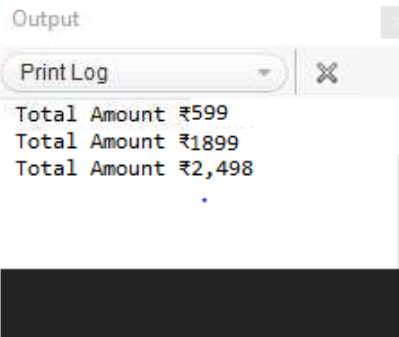


Fig. 3.2. Output

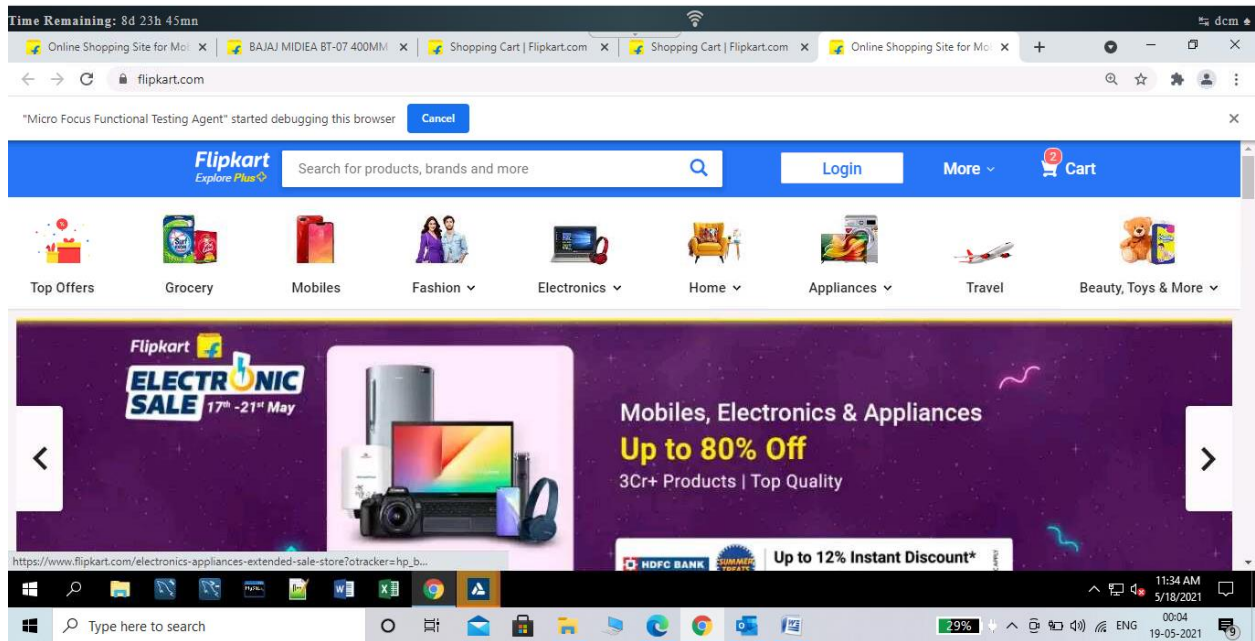


Fig. 3.3. Step1

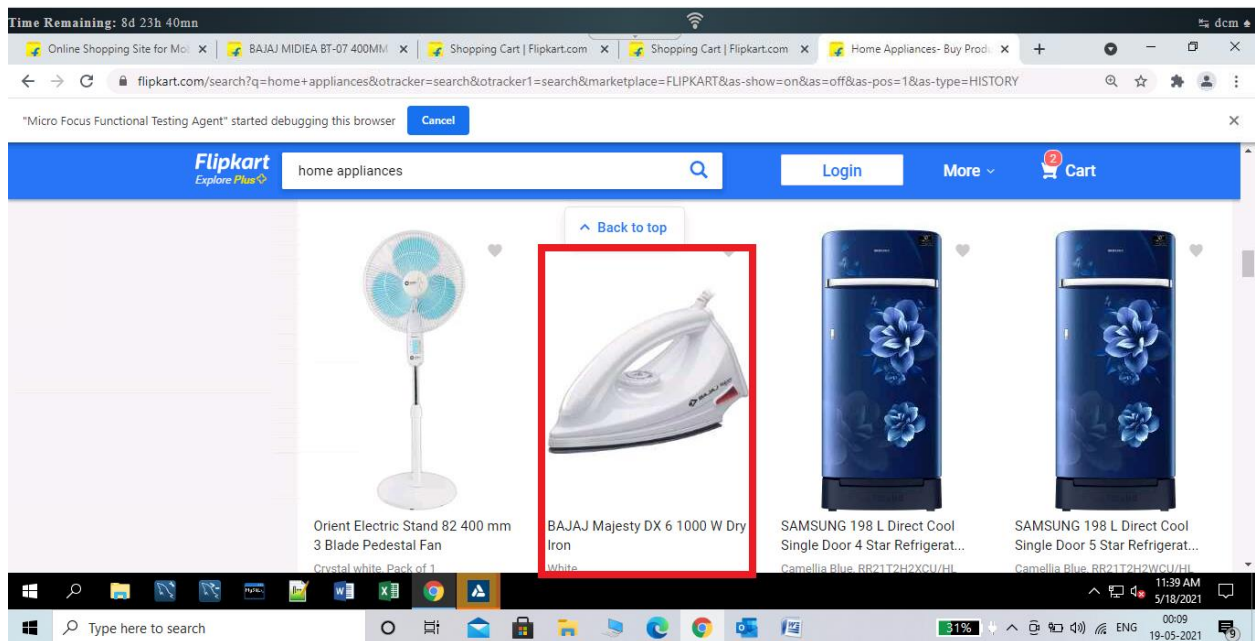


Fig. 3.4. Step2

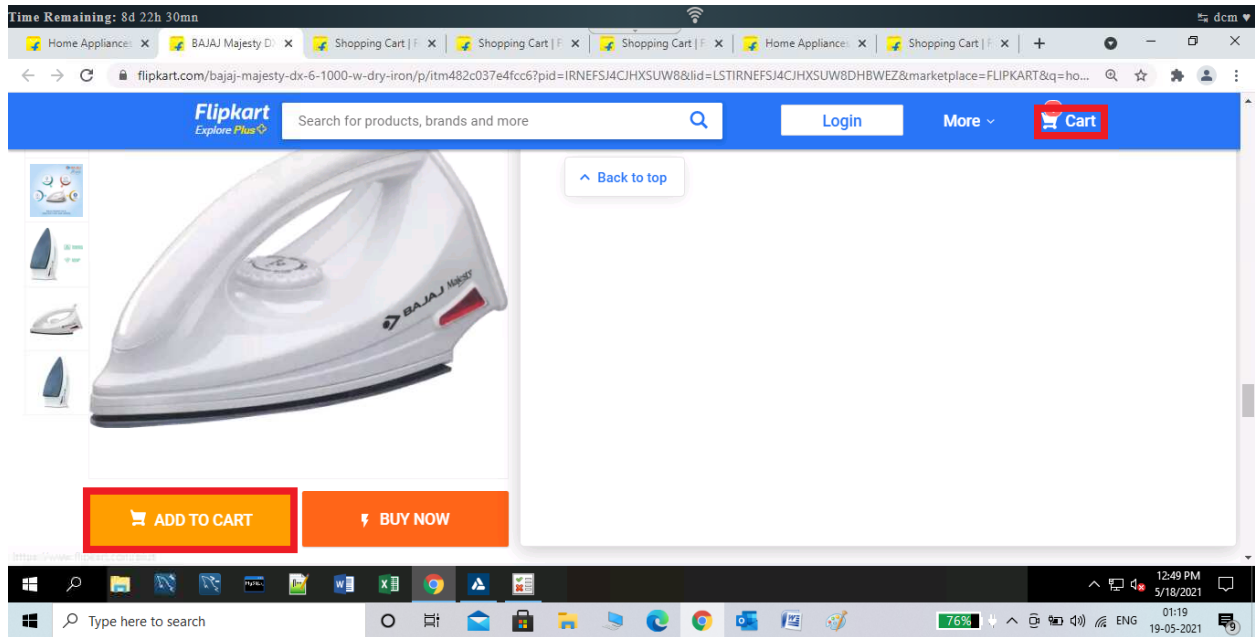


Fig. 3.4. Step3

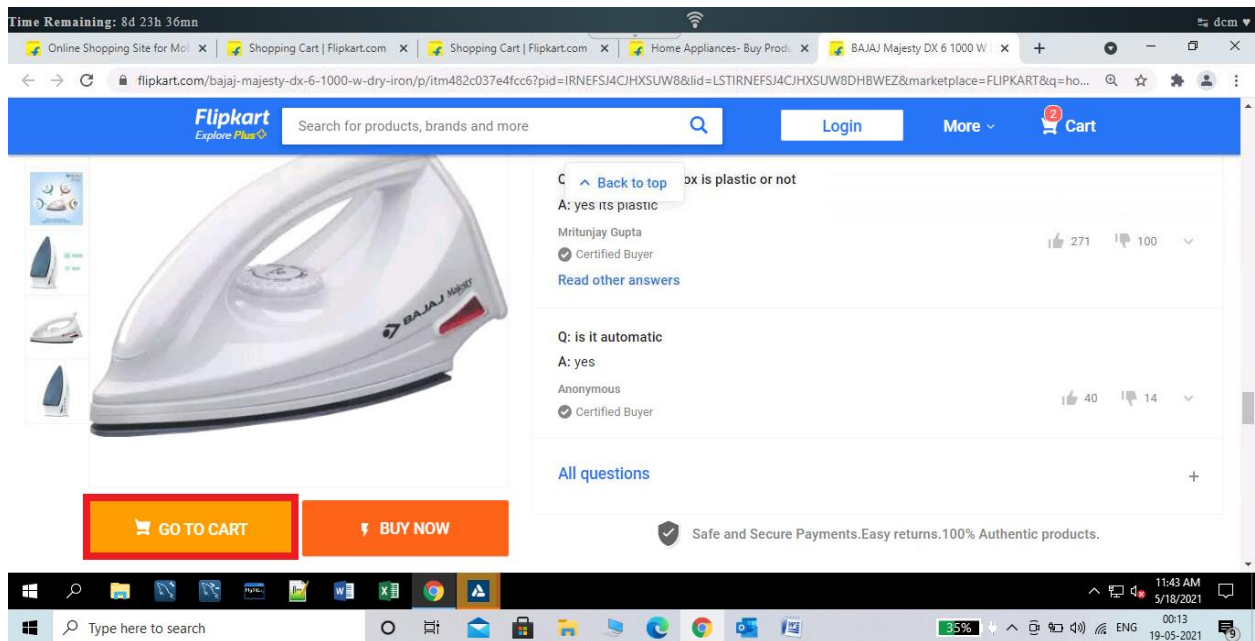


Fig. 3.5. Step4

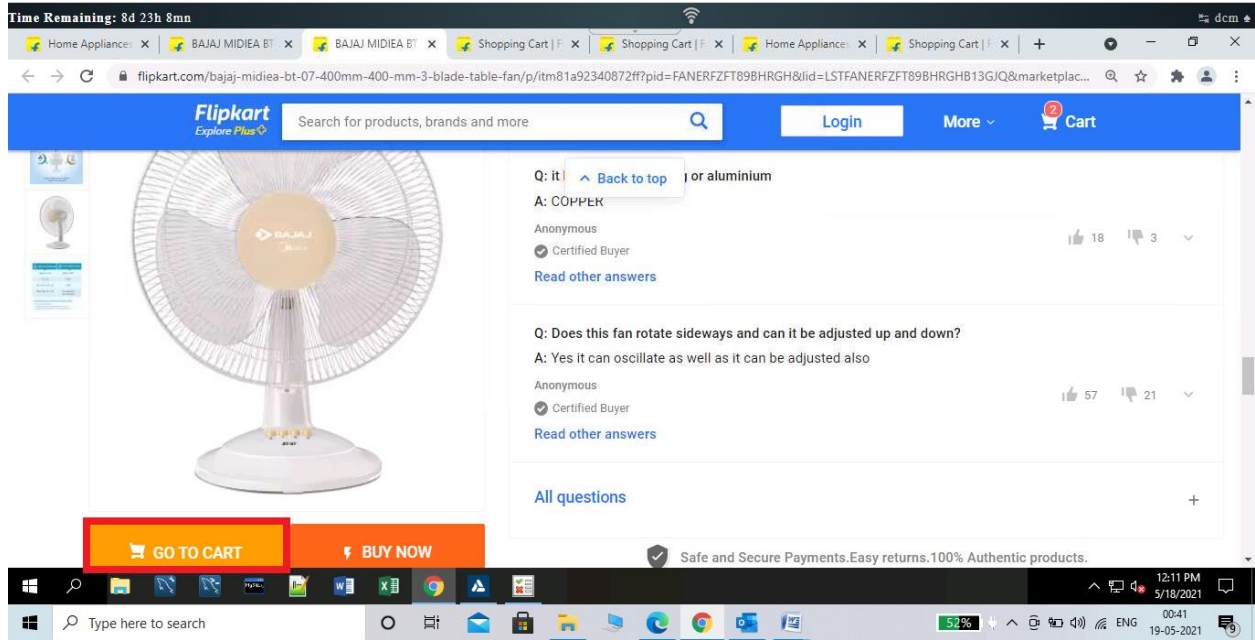


Fig. 3.6. Step5

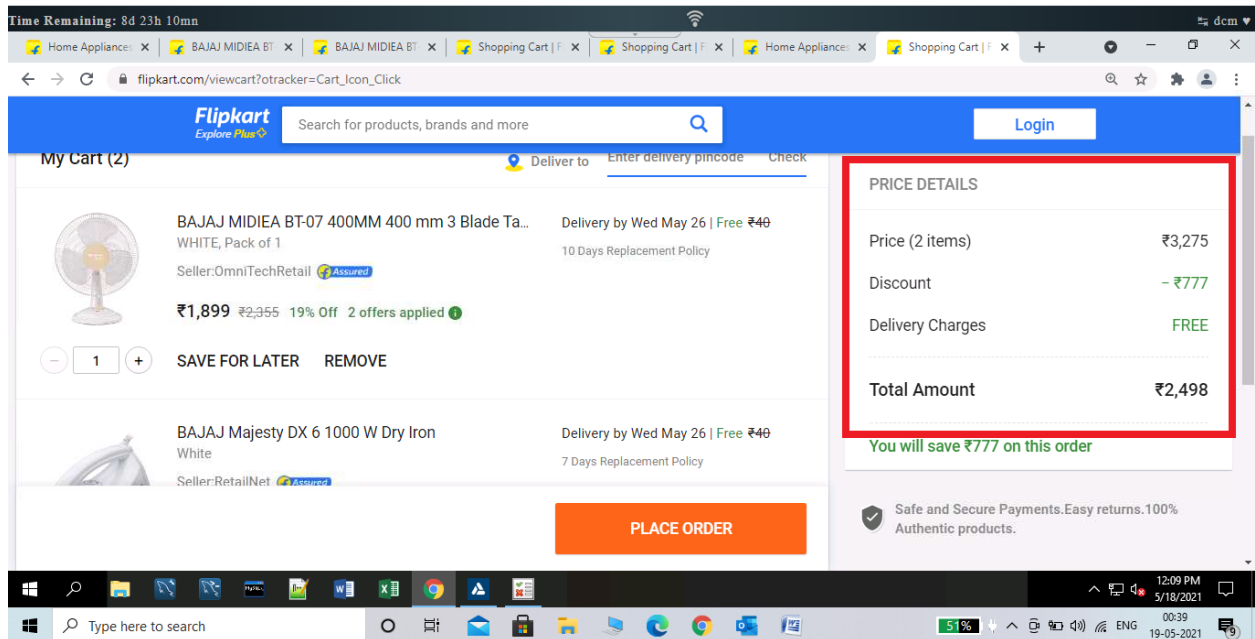


Fig. 3.7. Step6

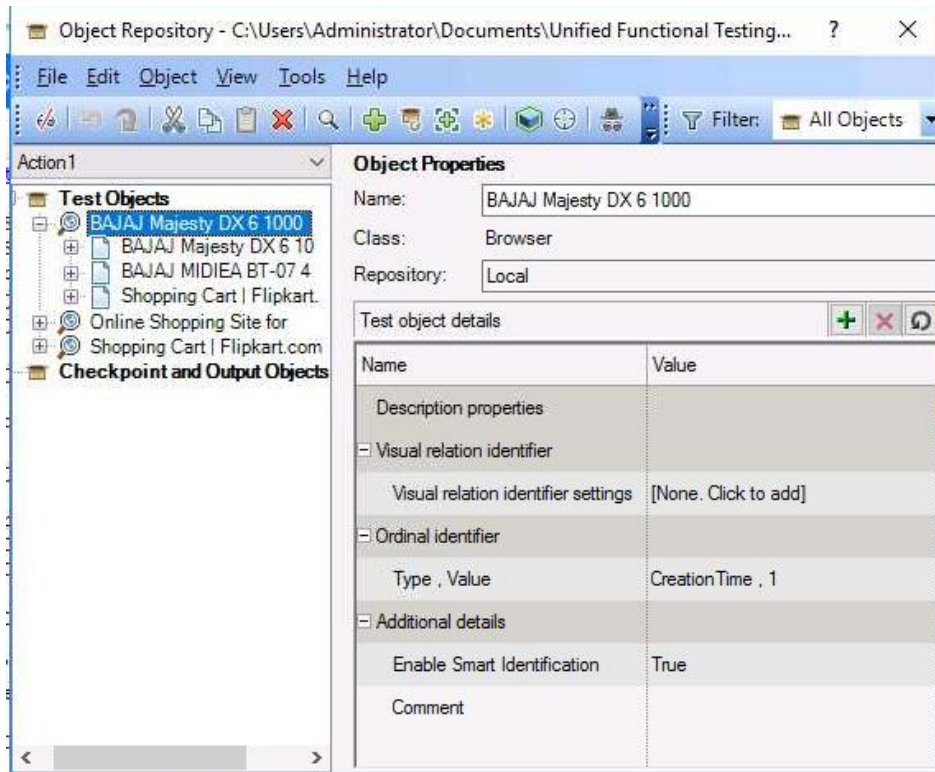


Fig. 3.8. Object Repository

## CHAPTER – 5

### RESULTS AND DISCUSSIONS

- Scheduling the test execution is easy 24/7: With the help of Automation testing, the flexibility to execute testing is increased. We can perform tests at any time, from everywhere. On the other hand, in case of manual testing, you need your personal computer for testing any application. It is not practically feasible to carry your own personal devices like laptops, computer, everywhere. In Automation testing, you can schedule your test case execution to run at anytime in a day, remote access is available and once the tests are done, report is automatically generated.
- Performing the regression testing is time consuming, and its completely anonymous about the performance of test cases the same way everytime. Thus, automation testing helps us out to overcome such problems, here we do not have to run and perform the entire regression cycle again and again, its just a one time effort.
- Reusability of the test scripts is another major benefit of Automation Testing.
- The proper and clear test reports are generated which helps in better understanding of test cases, thus bug fixing is quite more easier.
- With effective resource utilization, there is reduction in business costs and no human intervention is required.
- It improves the quality of Manual tests, better efficiency and hence maximizes the test coverage for complex web based applications.



## **CHAPTER – 6**

### **CONCLUSIONS**

Although there are several benefits of Automation testing and it is a growing field in IT sector, but still Manual testing has its own importance. We always need Manual Testing for better automated test cases. Moreover, one should choose the Automation Testing tool wisely, as automation testing is highly tool dependend. A wrong Automation tool selected might fail in submission of project deliverables on time.

Well, the learning doesn't end here, I'll discover more advanced techniques in this field and come up with innovative ideas soon.