# AUTOMATION TESTING

*Project report submitted in partial fulfillment of therequirement for the degreeof*

## BACHELOR OF TECHNOLOGY

## IN

## ELECTRONICS AND COMMUNICATION ENGINEERING

By

**Anshul Deorane (171069)**

**UNDER THE GUIDANCE OF**

**Dr. Rajiv Kumar**

**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,WAKNAGHAT**

**May 2021**

# TABLE OF CONTENTS

**CAPTION** **PAGE NO.**

# PROJECT REPORT UNDERTAKING

I Mr. /Ms. Anshul Deorane -Roll No.171069 Branch ECE is doing my internship with Cognizant from 27 Feb 2021 to 11 June 2021

As per procedure I have to submit my project report to the university related to my work that I have done during this internship.

I have compiled my project report. But due to COVID-19 situation my project mentor in the company is not able to sign my project report.

So I hereby declare that the project report is fully designed/developed by me and no part of the work is borrowed or purchased from any agency. And I'll produce a certificate/document of my internship completion with the company to TnP Cell whenever COVID-19 situation gets normal.

Signature

Name Anshul Deorane

Roll No. 171069

Date 20/05/2021

Dr. Rajiv Kumar

Supervisor

# ACKNOWLEDGEMENT

I would like to express my deep and sincere gratitude to my research supervisor, Dr. Rajiv Kumar Bachelor of Technology, MTech and Ph.D. for giving us the opportunity to do this project and providing invaluable guidance throughout this project. His dynamism, vision, sincerity and motivation have deeply inspired us. He has taught us the methodology to carry out the research and to present the works as clearly as possible. It was a great privilege and honor to work and study under his guidance.

He helped us in conceptualizing the project and building procedures used to complete the project. We would also like to thank our Project Head, Dr. Naveen Jaglan for providing us this golden opportunity to work on a project like this, which helped us in doing a lot of research and we came to know about so many things. Finally, we would like to thank our family and friends who guided us throughout the project so as to complete the project on time

# LIST OF FIGURES

# LIST OF TABLES

# ABSTRACT

This Project report is to define the technologies and software's that I have learned while going through the training program of my company. I was given hands on the tech that is used for AUTOMATION TESTING, Testing that is done with the help of scripts and framework. Course was divided into weeks where different relevant software's were taught to help us reach to the Automation part of the testing. We were started with Functional testing where the basic of testing and test report writings were taught. Moving on to MYSQL and learning to raise queries on Databases, followed by learning the scripting through VB Script and finally using UFT to develop the framework for all of them to work together.

# INTRODUCTION

Testing is required in all the fields of IT companies and to make the process smoother for the software companies, automation was introduced in it that not only improved the overall efficiency but made mundane tasks easier, so the tester can focus on the critical part of it. Automation plays a big role when a big part of the software is covered for testing or for simple tasks such as testing the older components after integration of the new ones. There are number of testing tools available in the market but in this report, we will be working with the UFT as its far more superior than the Selenium when it comes to the framework but quite as much expensive as well. Software testing is done to find the faults in the software. It is done to test not only the quality of the product but for the bugs as well, so the product doesn't fall off in the middle of deployment. While functional testing can be done with the help of manual as well as automated testing. Manual Testing is done manually by the testing team whereas automation testing is done with the help of software and scripts. Which enables to execute the test cases with much more pace, saving time. The same time can be used to enhance the product quality and meet the project deadline

# CHAPTER 1
# SOFTWARE TESTING

## 1.1 SDLC

SDLC – Software Development Lifecycle

SDLC is the process followed in the development of a software. It is a detailed process of how to create, develop, test and maintain the software. SDLC helps in creating quality products while maintaining a dedicated process to work through the development of it.

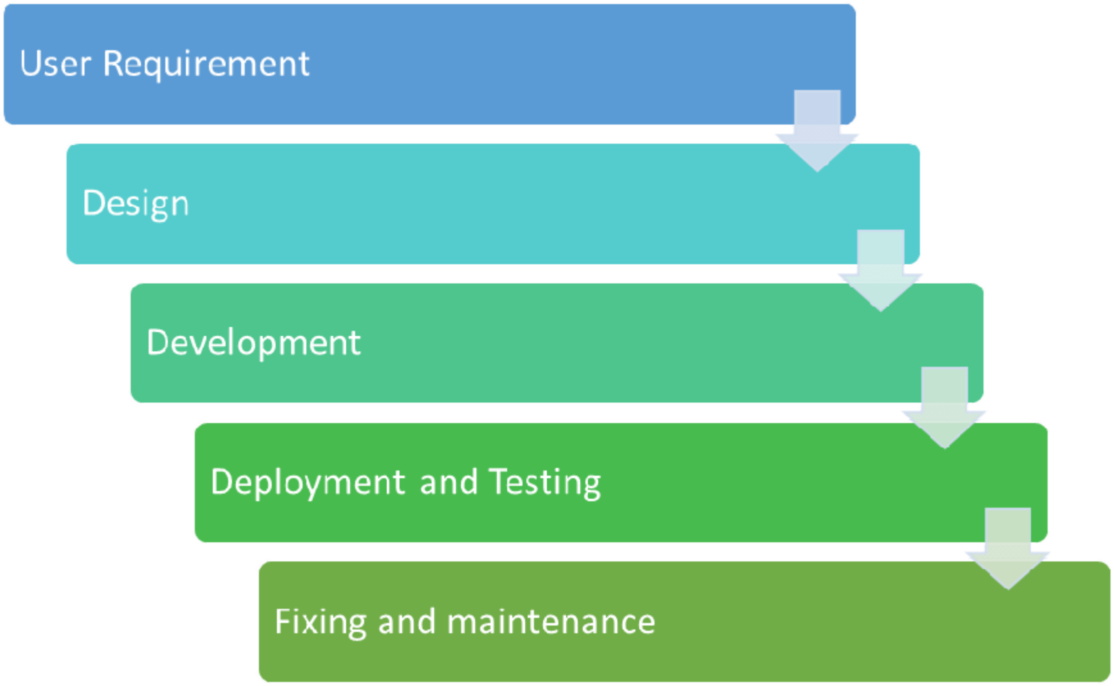


**Figure 1.1**Software Development Lifecycle

**User Requirement**– All the user requirements that are needed for the development of the system are documented in this step. A detailed user specification document is created.

**Design** – In this step the requirements from the previous phase are analyzed and then the architecture of the system design is created. This step is useful in identifying the hardware requirements and system requirements.

**Implementation** – After the architecture of the software have been designed, next step is to implement it. It is segregated into smaller units and are developed in this phase. Each unit has its own functionality, which is defined in the user requirement document.

**Integration and Testing** – In this step all the units are tested individually for any faults or defects, or any functionality not working properly. All the units are then integrated into a system.

**Deployment** – After the system passes the testing and is integrated successfully it is then deployed in the customer system so that its functionality there can be checked before the final release.

**Maintenance** – Maintenance is the step after deployment of our software all the software tweaks that might come up later deployment can be added along with the bugs that might raise using the system on different environments.

## 1.2TYPES OF LIFECYCLES

Development lifecycles can be divided into two types:

- Sequential
- Iterative and Incremental

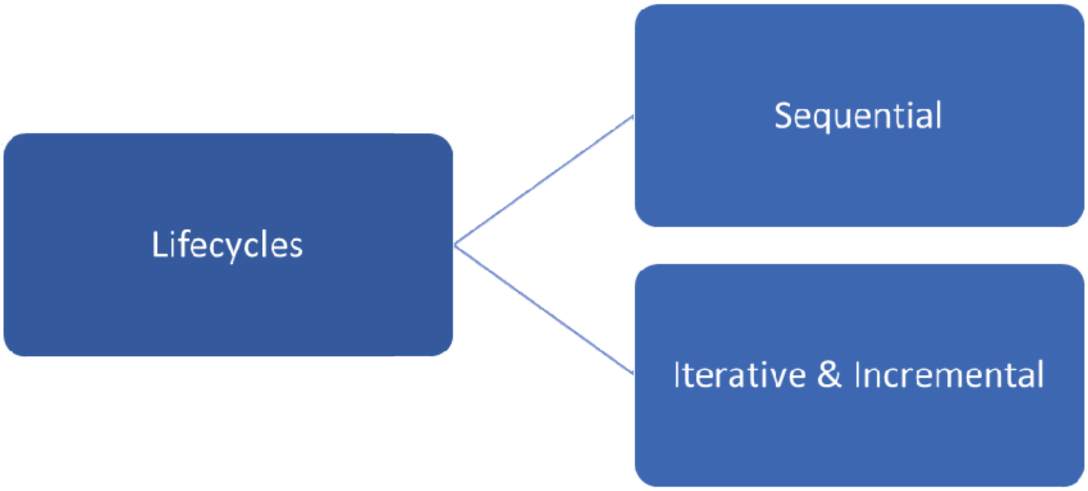


**Figure 1.2**Types of Lifecycles

### 1.3.3 SEQUENTIAL

It is the older ways of developing a software. It is a simpler process in which stages don't overlap and are followed in a sequential manner. It has an origin i.e. when the customer has an idea to develop, it is then followed by the requirements needed to create that idea. Then the design of the software is laid out, followed by the build. The build is then tested and fixed in the next stage and after successful completion of this stage the software is released to the customers.

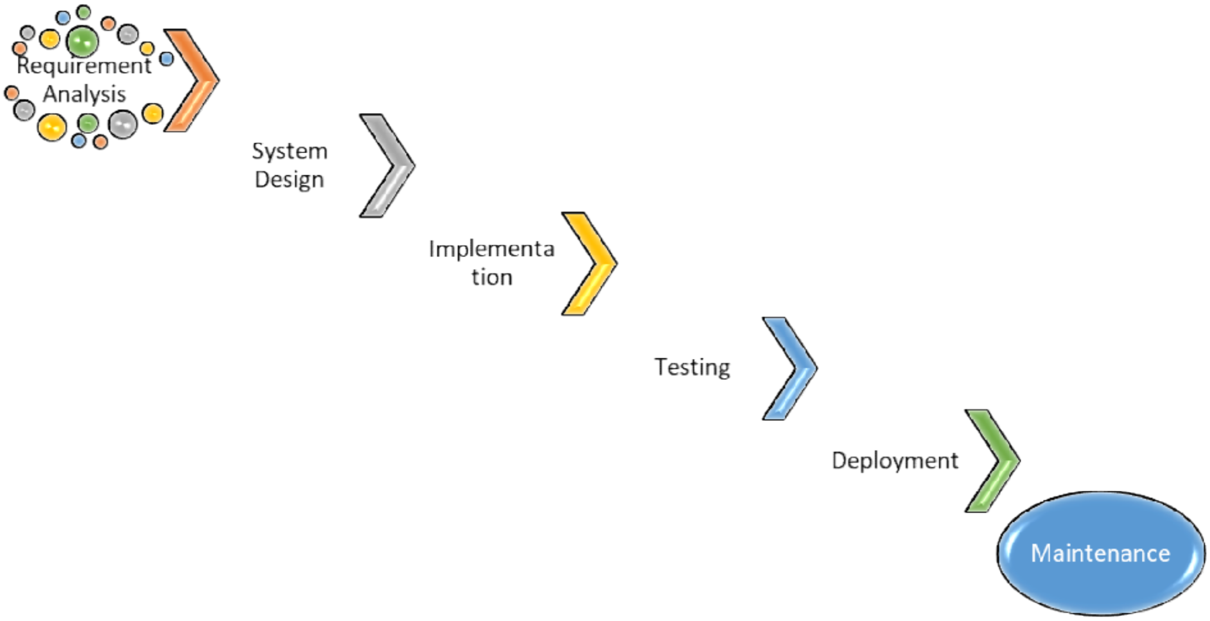
Two types of Sequential models that are used are:

- Waterfall Model
- V-Model

#### 1.2.2.1 WATERFALL MODEL

Waterfall model follows a linear structure and the activities are completed sequentially one after the other. In waterfall model we go from high level to low level. We don't come back to the previous step amid of the current one. It is best applicable for smaller projects where the requirements are well defined.

**Figure 1.3**Waterfall Model



### 1.2.2.2 V-MODEL

V-Model is also a sequential model but it's different from waterfall model in a way that the software is divided into two pars User requirements and Software requirements. This model has more emphasis on testing by parallel testing phases along with development. Due to simultaneous testing it is also known as Validation and Verification model. This model overcomes the faults of the waterfall model that testing starts after the development cycle of the software.

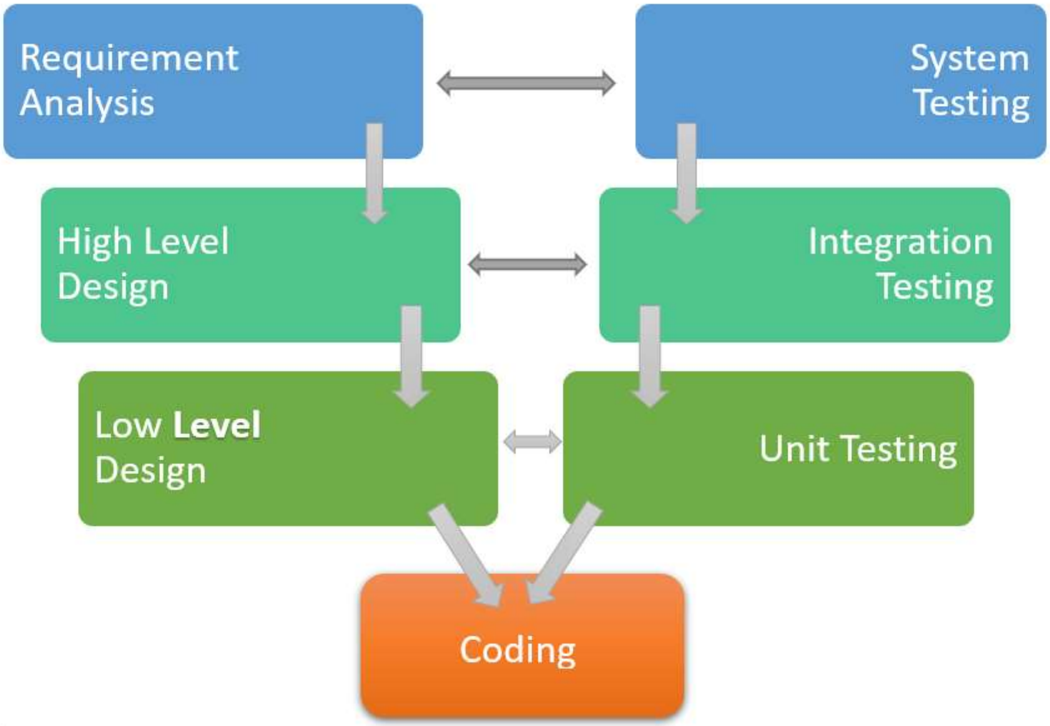


**Figure 1.4**V-Model

Left Side - Software Development Life Cycle - SDLC

Right Side - Software Test Life Cycle – STLC

### 1.3.3 ITERATIVE AND INCREMENTAL

In Sequential SDLC there was one major problem that it didn't enabled the team to make changes mid process, and that was one of the biggest drawbacks. It led to more interactive and active approach of using Iterations and Increments.

#### 1.2.2.1 ITERATIVE MODEL

In Iterative model we divide the software into iterations of layers. It starts off with a simpler version with each iteration increasing its functionality and evolving its implementations. The best feature of this model is that there will be a working system prototype at the early part of the development and can be built upon iteratively while keeping the integration of components and the functionality in the check.

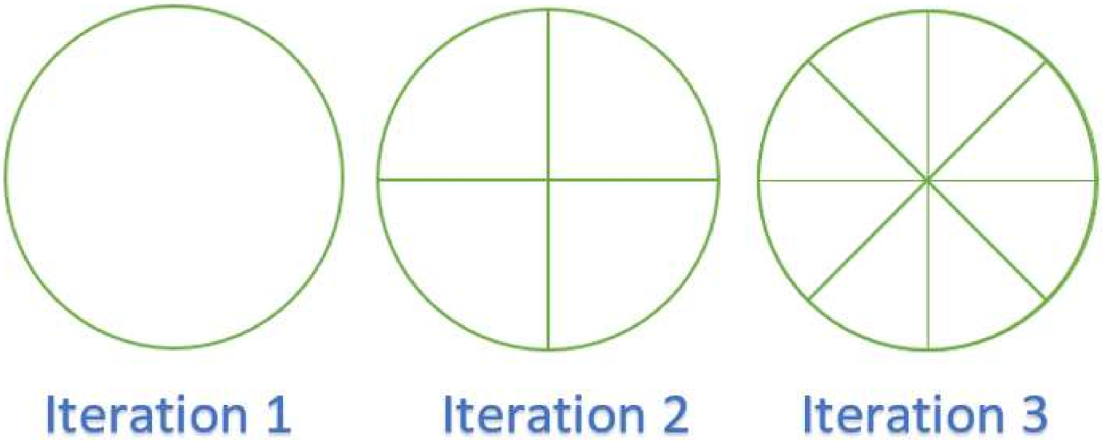
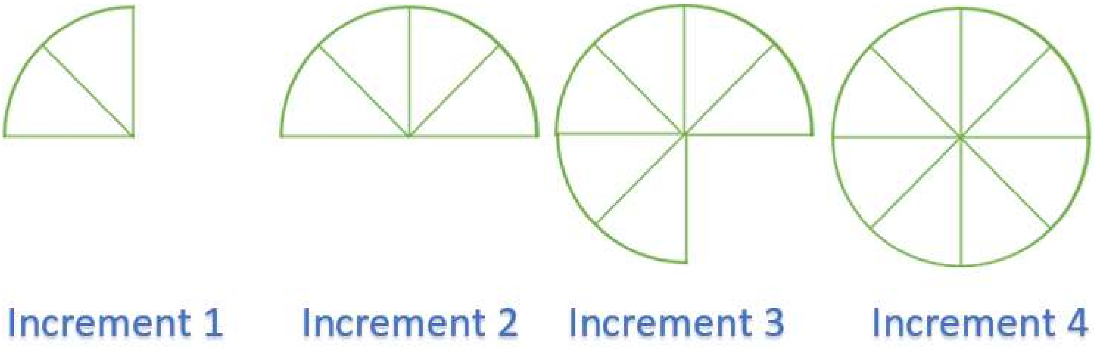


**Figure 1.5**Iteration Model

### 1.2.2.2 INCREMENTAL MODEL

In incremental model the user requirements are broken down into smaller units and then that part is of a section is finished before moving on to the next one. The smaller units are standalone modules that can function on their own. In incremental model fundamental requirements are addressed first and then the secondary functions are added in forthcoming increments.

**Figure 1.6**Incremental Model



## 1.3 SOFTWARE TESTING

Software Testing is a process to assess the quality of the developed software whereas reducing the risk of its failure as well. Since Software testing includes many activities that's why it is known as a process. It can be of two types Dynamic Software Testing and Static Software Testing. Dynamic Software testing includes the execution of the software whereas in the Static Software Testing we do not execute the code, we just review the code and its design etc.

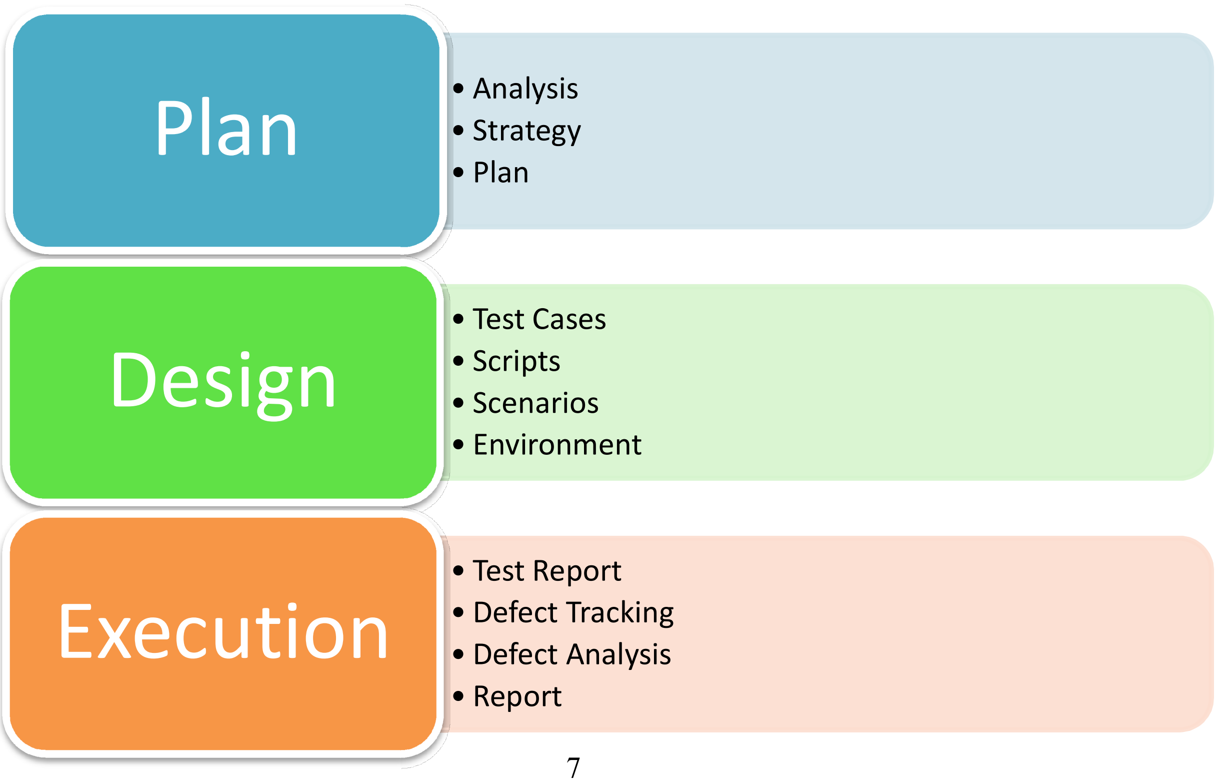
Validation and Verification are two major activities. Validation means building the right product according the user requirement. Verification means we build the product right that is its design, architecture and speed.

Work Products are anything that is produced while we are working inside our software. Example: Requirement, Codes and Design.

### 1.3.3 TEST PROCESS

It is the steps that we go through in-order to test our software.

**Figure 1.7**Test Process



### 1.3.3 TEST PROCESS ACTIVITIES

**Test Planning** – It is the documentation of the Test Plan. In this we document all our test related data, including reviewing the basis of the test.

**Test Monitoring and Control** – In this we monitor our progress regarding the test of the software whereas also keeping in control so that the path doesn't differ from the goal.

**Test Analysis** – In this we search the defects and analyze the test conditions.

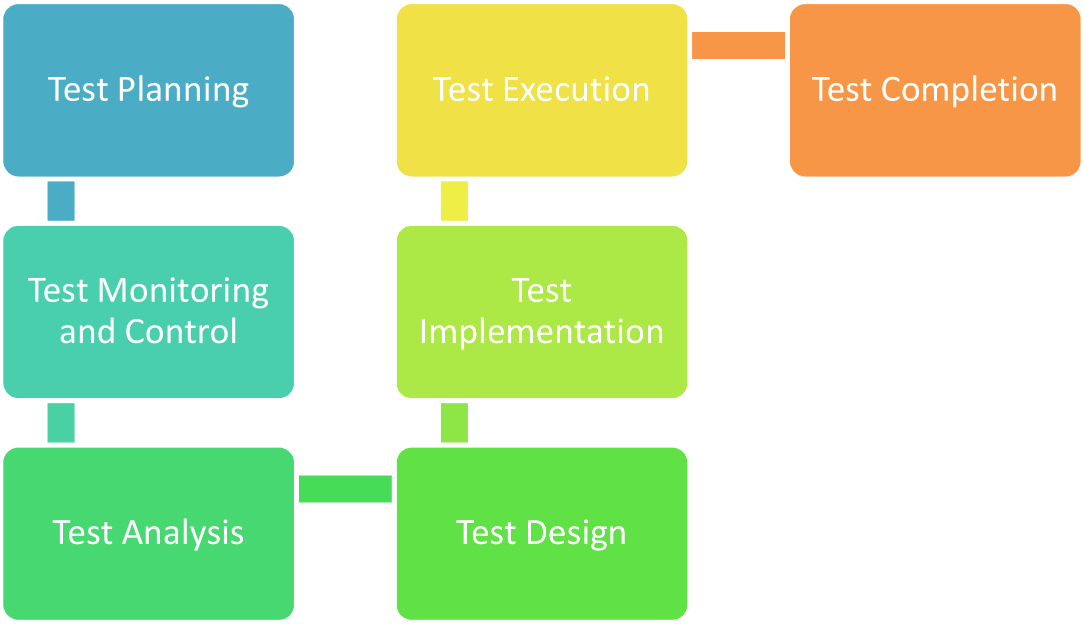
**Test Design** – This is where the different test cases are written.

**Test Implementation** – Here we organize the test cases in the environment (browser, operating system. Etc.)

**Test Execution** – It is the step involving reporting the whole process, retesting the reworked functionalities and regression.

**Test Completion** – After all the previous test are completed, it leads to the completion of the process.

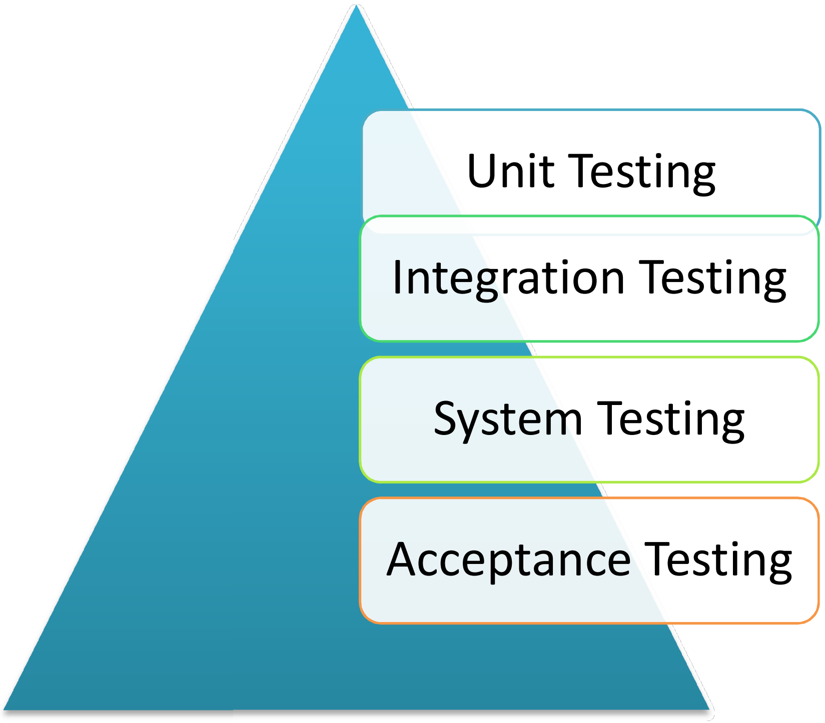
**Figure 1.8**Test Case Writing Process



### 1.3.3 TEST LEVELS

It is the grouping of the test activities, organized together in the Development cycle. Test levels are used to make software testing easier and organized, so it is easier to consider the possible test cases and scenarios. It is used to check from functionalities, performance to the integration of its components.

**Figure 1.9**Test Levels



**Unit Testing** – It is the most basic unit of the system that can be tested separably, it could be modules or classes of the code. It is basically done by the developer during the development stage.

**Integration Testing** – It can be subdivided into two parts Component Integration and System Integration. Component Integration is the integration between the components and is usually done by the developer. On the other hand, System Integration is done in bigger projects which may include many systems like frontend, backend, etc. It is usually done by the tester.

**System Testing** – It is the most important testing in this we test our system in the real environment. It checks everything including integration among components, its stability and performance as well as the security.

**Acceptance Testing** – It is mostly done by the user / stake holders. It is done to check that all the user requirements have been met as per the user requirement document.

### 1.3.7 TESTING TYPES

**Functional Testing** – It tests what the system does. It is to confirm that the system can carry out the functionalities as mentioned in the User Requirement.

**Non – Functional Testing -** In this testing all the non – functional tests are performed. Performance Testing is done to test how the system is working. Stress Testing is done to check how much load can the system take. It is hard to answer in Yes/No and is usually answered in a range.

**Black box Testing** – In black box testing the tester is not known to the internal structure of the system. The tester can only give an Input and check if he had received the appropriate output.

**White box Testing** – In this we do testing while monitoring the internal structure of the system.

**Dynamic Testing** – Testing involving the execution of the system is known as Dynamic Testing.

**Static Testing** – In Static testing there is no execution of the system, it is basically used for reviewing the design and code of the system.

**Retesting** – It is also known as Conformational Testing. It is done when changes are made by the developer after a bug was reported. It uses the same steps to test that caused the defect.

**Regression Testing -** It is done when the developer changes something i.e. changing the requirement, adding new functionalities, or removing a feature.

**Smoke Testing -** Smoke Testing is Testing the main functionalities to ensure that the build is stable enough for testing.

Difference between Regression testing and Retesting is that Regression testing is done to retest the parts that were not changed while making changes to the system. Whereas retesting is only done on the defected component.

### 1.3.7 TEST CASE WRITING

Test Case is a set of preconditions, inputs, actions (where applicable), expected results and post conditions developed based on test conditions.

**Test Case Title** – Title should be appropriate and should be able to give enough information.
**Precondition** – Condition that should be performed before testing.
**Test Steps** – It contains the step that should be carried out in order to perform a test. They should be very clear and discuss important things.
**High Level Test Case** – When we don't provide exact credentials for checking because it is written before/while software is under development.
**Low Level Test Case** – When the developer provides data to be used for testing.
**Expected Result** – It contains what the expected result is from the system after the test steps are followed.
**Test Scenario** – Test Scenario or Test Suite is grouping of similar test cases into one Suite.
**Test Environment** – In this we mention the specifications of the hardware, software and the network used.
**Actual Result** - Result obtained after executing the test case.
**Status** – Indicates the current status of the test case.

- New – Needed to be tested
- Pass – On execution of the test case expected result is same as the actual result
- Fail – On execution of the test case expected result is not same as the actual result
- Blocked – the test case cannot be executed due to a blocked functionality

### 1.3.7 BUG REPORT

Documentation of the occurrence, nature and testing of a defect. A bug occurs when expected result don't match the actual result.

**Bug Report Title** – It includes the Description; enough information is provided in the title.
**Steps to reproduce** – It’s the steps that leads to the bug, must be very specific.
**Expected Result -** What should have happened after the execution.
**Actual Result** – What really happened when the steps were executed.
**Test Environment** – Device/Operating System/Network used for the execution of the test case.
**Screenshot/Video** – Further on media can be attached to the report to make a clear point. It should show full screen, a red rectangle around the defect area and the video should show mouse clicks as well as keyboard key presses.
**Bug Priority** – Indication level of the priority of the bug.
**Critical** – A functionality of the system does not work.

- High – Non-Functional defects or low priority functional defects.
- Medium – Minor performance defects or UI display defects
- Low – Defects like spelling mistakes or Image misalignment

### 1.3.7 APPLICATION

Writing the test cases for an online website. A sheet can be divided into four parts.

1) The first sheet will hold the test scenarios.

**Table 1.1** Test Scenario

| Module | Scenario ID | Scenario Name | Scenario Description | Requirement id |
|---|---|---|---|---|
| Raj Travels Website | RT_01 | Hotel Booking - Valid Details | Attempt to book hotel by providing valid details | REQ1 |
| | RT_02 | Hotel Booking - Invalid Details | Attempt to book hotel by providing invalid details | REQ2 |
| | RT_03 | Hotel Booking - Partial Details | Attempt to book hotel by providing partial details | REQ3 |
| | RT_04 | Validating Fields | Validating fields of the user interface | REQ4 |

2) Followed by the Test Cases, an example of one test case is shown.

**Table 1.2** Test Case

| Test Scenario ID | Test case id | Test case description | Prerequisites | Steps to execute | Expected results | Actual results | Pass/Fail | Defect id | Remarks |
|---|---|---|---|---|---|---|---|---|---|
| RT_01 | RT_011 | Searching for a hotel in Delhi by providing valid details. | 1) Website should be up and running.<br>2) Mandatory fields should be filled.<br>3) Check out date should be greater than or equal to the check in date. | 1) Select the Location to be as India by selecting its radio button.<br>2) Select city to be as Delhi in the drop down box.<br>3)Select the valid "Check In" date in the calendar.<br>4)Select the valid "Check out " date in the calendar.<br>5)Choose the appropriate "Nationality " in the list box<br>6) Select the valid "Number of rooms" in the list box<br>7) Select the number of adults and children's from the drop down box.<br>8)Click on Search button | Data is validated, and the user is redirected to the next screen showing the hotels available in the selected city. | Search results for the city of "Calcutta" are displayed instead of "Delhi. | Fail | ER_11 | Search results for the wrong city is displayed. |

3) Test Report is followed by the defect/bug report. Bug report of the above test case is mentioned.

**Table 1.3** Bug/Defect Report

| Defect id | Description | Reproducible (yes/no) | Steps to reproduce | Severity | Priority | Reported by | Reported date | Status |
|---|---|---|---|---|---|---|---|---|
| ER_11 | On selecting the city to be "Delhi" in the "City" List Box, the application is showing results for the city of Calcutta. | Yes | 1) Select the Location to be as India by selecting its radio button.<br>2) Select city to be as Delhi in the drop down box.<br>3) Select the valid "Check In" date in the calendar.<br>4) Select the valid "Check out " date in the calendar.<br>5) Choose the appropriate "Nationality " in the list box<br>6) Select the valid "Number of rooms" in the list box<br>7) Select the number of adults and children's from the drop down box.<br>8) Click on the Search button | Critical | High | Anshul Deorane | 05-Apr-21 | New |

4) Requirement Traceability Matrix (RTM) is mentioned at the last of the document

**Table 1.4**Requirement Traceability Matrix

| Requirement id | Requirement description | Test scenario id | Test case id | Defect id |
|---|---|---|---|---|
| REQ1 | Hotel Booking - Valid Details | RT_01 | RT_011 | ER_11 |
| | | | RT_012 | |
| | | | RT_013 | |
| | | | RT_014 | |
| | | | RT_015 | |
| | | | RT_016 | |
| | | | RT_017 | |
| | | | RT_018 | |
| | | | RT_019 | |
| | | | RT_0110 | |
| | | | RT_0111 | |
| | | | RT_0112 | |
| REQ2 | Hotel Booking - Invalid Details | RT_02 | RT_021 | |
| | | | RT_022 | ER_22 |
| | | | RT_023 | |
| REQ3 | Hotel Booking - Partial Details | RT_03 | RT_031 | |
| | | | RT_032 | |
| | | | RT_033 | |
| | | | RT_034 | |
| | | | RT_035 | |
| | | | RT_036 | |
| | | | RT_037 | |
| REQ4 | Validating Fields | RT_04 | RT_041 | |
| | | | RT_042 | ER_42 |
| | | | RT_043 | |
| | | | RT_044 | |
| | | | RT_045 | |
| | | | RT_046 | |
| | | | RT_047 | |

## 1.4 BLACKBOX TESTING

Blackbox testing is done without knowing the internal structure of the system. There are four major ways in which black box testing can be done.

1) **Equivalence Partitioning (EP)** – In equivalence partitioning the system is divided into partitions (may not be of the same size). One test case is performed for each partition. In EP it is good to draw the number line. Before and After cases are also provided for the buffer, these are known as invalid Partitions.
2) **Boundary Value Analysis (BVA)** – It is an extension over EP technique, in BVA we will check for the boundary values as well. We study first value in current partition and the last value in the previous partition. Each value will belong to only one partition.
3) **Decision Table Testing** – Different combinations of conditions result in different outcomes. The number of tables for n number of questions with m possibility is $m^n$.
4) **State Transition Testing -** System consists of states and transitions. Behavior bases testing is done.

## 1.5 WHITEBOX TESTING

In white box testing we look at the internal structure, also known as architectural based testing. It and be subdivided into given techniques.

1) **Statement Coverage** – It covers all the statement inside the code and have hundred percent statement coverage.
2) **Decision Coverage** – Tests all the decision outcomes in the code. It is also known as the branch coverage.
3) **Condition Coverage** – Test each condition in the case in true and false cases.
   Path Coverage – Tests all the paths present in the code.
4) **Modified Decision/Condition coverage** – It tests only important conditions.

# CHAPTER 2
# AGILE

## 2.1 INTRODUCTION

Agile Software Development is a method in which a software is developed through continuous iterations rather than following a strict sequential. Agile was developed to provide a better solution for business over sequential. Agile like the word suggests is used to better optimizations of the resources of a Team and making the process more efficient. It includes elements like continuous development, sharing knowledge, team integration, cooperation and enablement of an individual to grow. It has a working prototype from early stages and gets continuously developed with time of the project, it facilitates flexibility to software changes, helps with the timeline and avoids delays as well.

## 2.2 AGILE FOUR VALUES

- **Individual and Interaction over Process and Tools** – That is interaction with the individuals and Relations between is more important than a process or a tool.
- **Working Software over Comprehensive Documentation** - That is it’s important to have a piece of working software over written requirements and documentation.
- **Customer collaboration over Contract Negotiation** - That it is important to get a satisfied customer rather than just following the contract rigidly.
- **Responding to change over Following the plan** – Have the flexibility to be able to make changes to the plan and not follow it too strictly.

## 2.3 WHOLE TEAM APPROACH

In this the complete team is responsible to every outcome and result. It includes including all the individuals that might have different perspective and knowledge to help with the project. Has a small team up to five members. Interaction between the team is relatively high. The testers role is to support and collaborate with the product owner to help them to create proper acceptance test along with the role of a tester for system development. Working with the developers to agree on the testing strategies as well as transfer and extend knowledge to get the best outcome.

**Power of three** – That is the concept of involving tester, developer and business representative.
Early and frequent feedback can help us focus on the features with highest business value. We can manage ourselves better. Requirement misunderstanding is avoided. Project is constantly monitored and there is a consistent project momentum.

**User Story** is used in Agile for the requirements and it is the same as the user requirement document.

## 2.4 AGILE MODEL VS WATERFALL MODEL

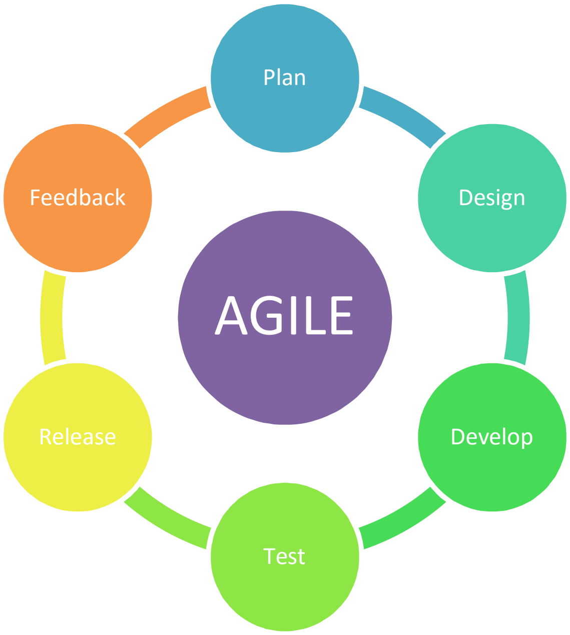
The two models are two different approaches that can be used to develop a software.

**Table 2.1**Difference between Agile and Waterfall model

| AGILE MODEL | WATERFALL MODEL |
|---|---|
| This method approaches software development on building the software through continuous iteration | This method approached software development<br>To have a beginning and end point and the work flows from one level to another. |
| The process is segregated into small parts where individuals are given separate work at a time | In this process is not segregated |
| One of the biggest advantages is that the user has a rather early and continuous opportunity to interact with the software and can decide whether any deviation from the original plan is required | In this approach the user is able to see the final product only and hence not allowing for any changes |

| | |
|---|---|
| Interaction and development are ranked above documentation | Documentation is the highest priority |
| In agile model developers, users and testers work together | In Waterfall model every team works differently |
| In the end of a sprint a product is represented to the user to check their expectations | Only the product is presented to the customer for acceptance testing. |
| A lot of planning and testing is involved where testers and developers work as a single unit. | It is not well planned as agile, things are usually not planned, and it causes delays between development and testing. |

**Figure 2.1**Agile Model



# CHAPTER 3
# DATABASES AND QUERY LANGUAGE

## 3.1 INTRODUCTION

Database is an organized collection of data(information). Although it should be able to allow interaction of user with the data as well. It is done with the help of a Database Management System. DMS takes query from the user and interacts them with the Database.

## 3.2 SQL

SQL stands for Standard query language and is used to communicate with the databases. Used for performing task on the Database. It is used for raising queries to the database.

## 3.3 RDBMS

Relational Database Management System is a collection of data into organized tables. Tables in RDBMS are linked together (or can be linked together). They are also capable of storing common tables.

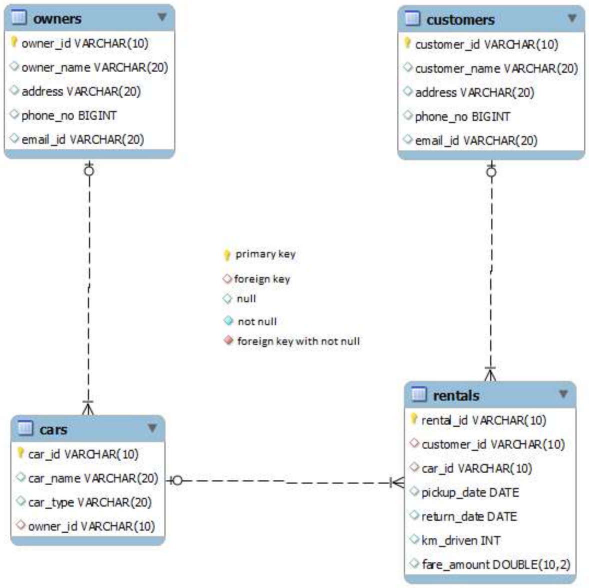
## 3.4 MYSQL

It is a Relational Database Management System (RDBMS). It is based on SQL and provides a UI to access and interact with the database. MYSQL workbench is an integrated development environment for the MYSQL database. It is a visual tool for DB design, DB development and SQL development.

### 3.4.1 APPLICATION

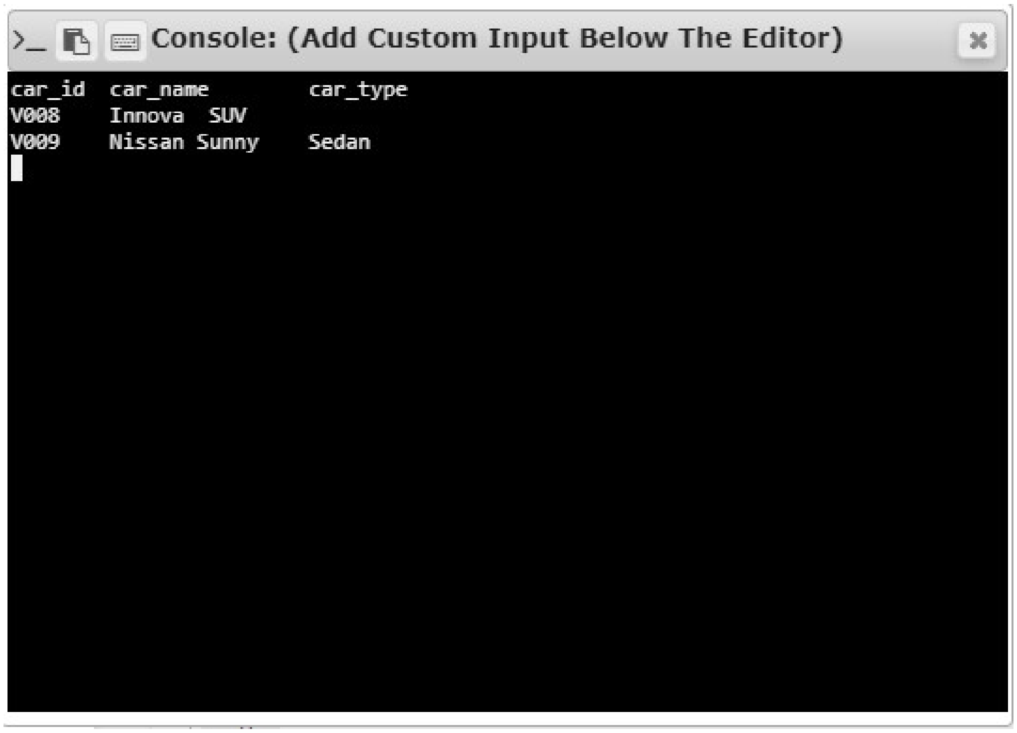
Database Schema for the current code.

**Figure 3.1**Database Schema



#### 3.4.1.1 Write a query to display car id, car name, car type of cars which was not taken for rent. Sort the result based on car id.

```
SELECT c.car_id, c.car_name, c.car_type FROM cars c
LEFT JOIN rentals r ON c.car_id=r.car_id
WHERE r.pickup_date IS NULL
;
```



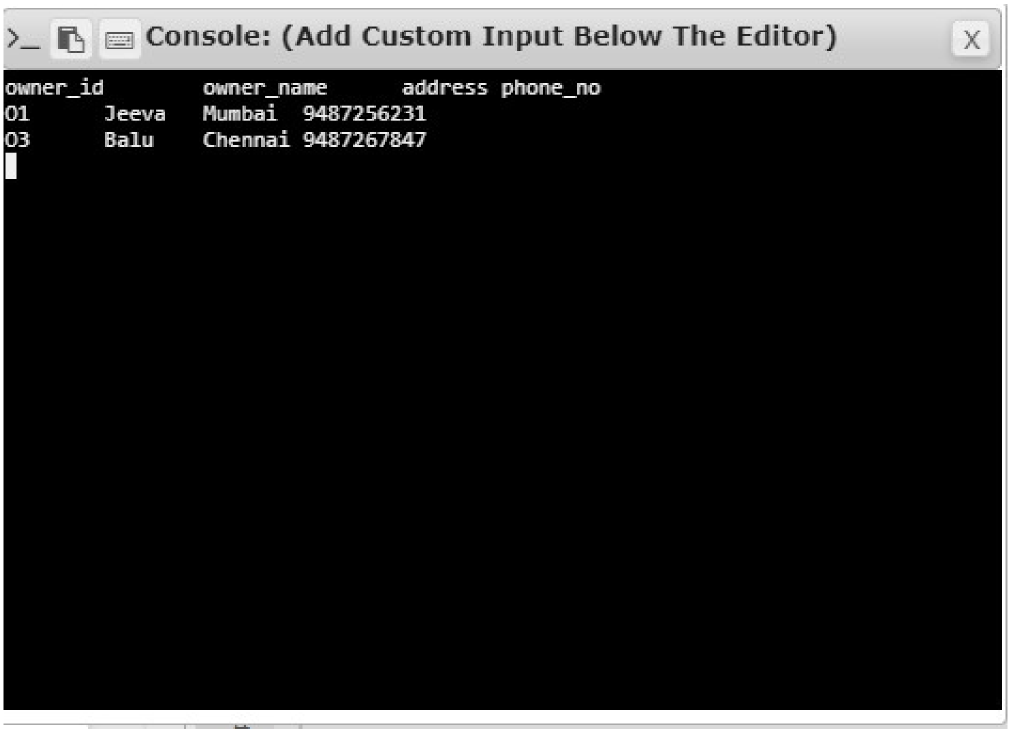
```
car_id  car_name        car_type
V008    Innova  SUV
V009    Nissan Sunny    Sedan
```

**Figure 3.2**Output 1

**3.4.1.2 Write a query to display distinct owner id, owner name, address, and phone no of owners who owns 'Maruthi' company car. Sort the result based on owner id.a query to display car id, car name, car type of cars which was not taken for rent. Sort the result based on car id.**

```
SELECT o.owner_id, o.owner_name, o.address, o.phone_no FROM
JOIN cars c ON o.owner_id = c.owner_id
WHERE c.car_name LIKE ‘%Maruthi%’
;
```

**Figure 3.3**Output 2



```
owner_id        owner_name      address phone_no
01      Jeeva   Mumbai  9487256231
03      Balu    Chennai 9487267847
```

## 3.5JSON

JSON stands for JavaScript object notation. It is a programming language. It was originally developed to hold structured data to be used in JavaScript. In simple words we can define structured data as the data that cannot be stored using just tables.

JSON uses two data type to store the data. First would be array, it is defined by square brackets. An array can hold multiple objects. Objects are JSON dictionaries that are defined with curly braces and have a key & value pair. Each pair is separated by a semicolon and can be of any data type.

JSON also have the concept of nesting. It involves putting arrays and object inside of each other, although white space and indentation doesn't mean anything in JSON.

## 3.6XML

On the other hand, XML as well can be used to hold the structured data. It stands for Extensible markable language. It is also used for holding structured data. It is written using tags, where <abc> denote the start tag and </abc> denotes the ending tag. It is mandatory that the starting and ending tags should match with each other. Tags can contain numbers letters

and even under scores. The content is stored in between the tags, if a content doesn’t have a tag then it is treated as a tag, even without quotes.
Data can be Nested as well with tags inside of other tags. Tags can have attributes as well, they hold simple data(string) and have a key/value pair. They appear in the start tag.

### 3.6.1 WRITING AN XML FILE

First line is XML declaration, it indicates that it is an XML file. Other information is also included in it like the version, encoding etc. It is a good habit too write one, but it still isn’t mandatory.

Developers might use the tag for something in one XML file and for another thing on the other one, when trying to merge the files it might cause a lot of trouble for the developers, to resolve we use the concept of Namespace, it means that the tag can mean one thing in one content and something else in a different content. It is achieved with the help of attributes, by using different attributes for the same tag dissolves the conflict caused by the same tags.
White space and Indent don’t matter like JSON.

Schemas define the stricture of the XML files, they are called an XSD file. Full form of XSD is XML Schema Definition.

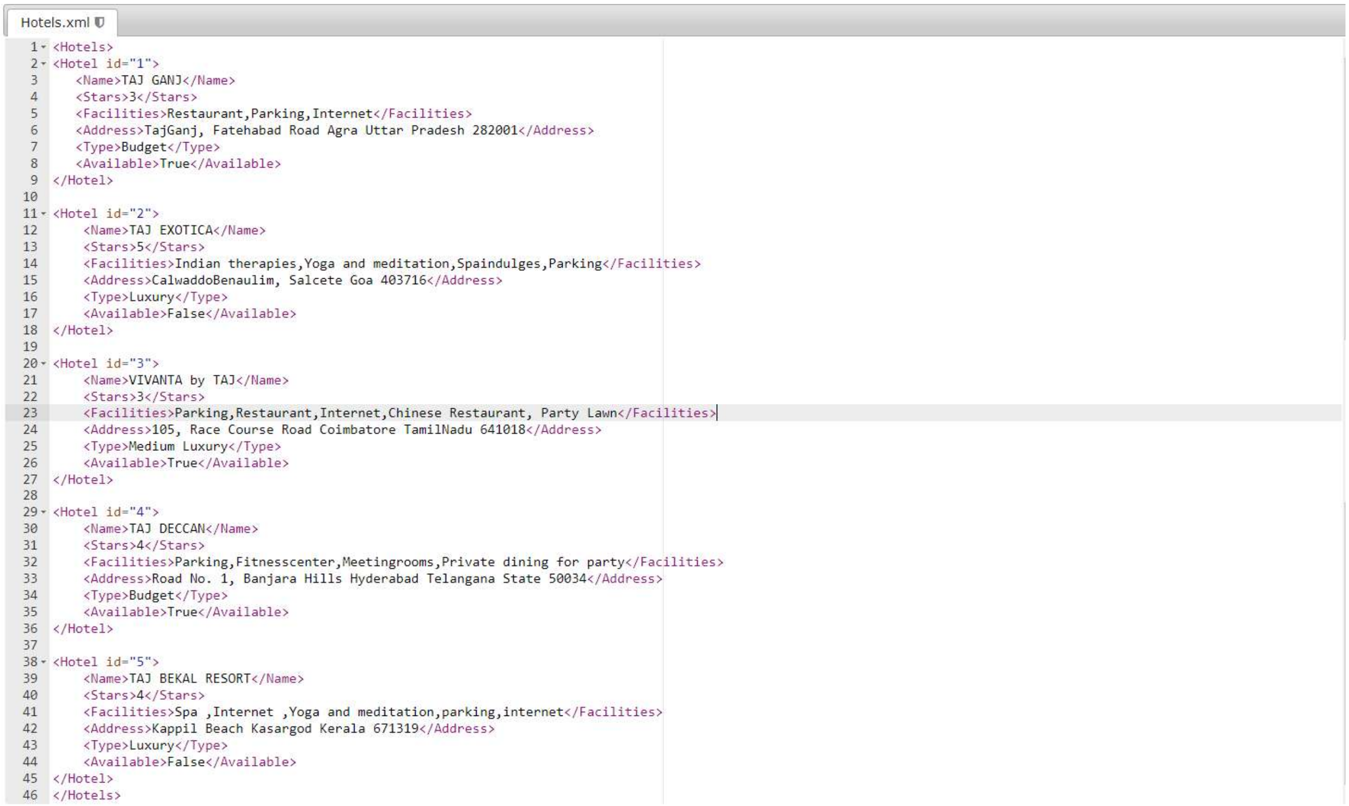
### 3.6.2 APPLICATION

Generating an XML document for the given scenario

**Table 3.1**Scenario

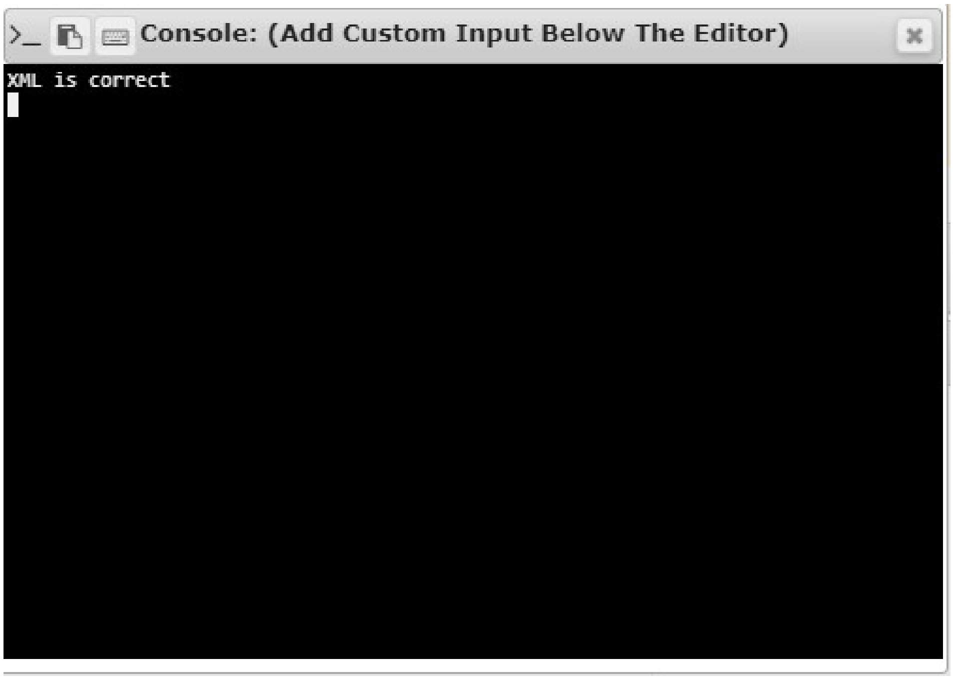
| id | Name | Stars | Facilities | Address | Type | Available |
|---|---|---|---|---|---|---|
| 1 | TAJ GANJ | 3 | Restaurant,Parking,Internet | TajGanj, Fatehabad Road Agra Uttar Pradesh 282001 | Budget | True |
| 2 | TAJ EXOTICA | 5 | Indian therapies,Yoga and meditation,Spaindulges,Parking | CalwaddoBenaulim, Salcete Goa 403716 | Luxury | False |
| 3 | VIVANTA by TAJ | 3 | Parking,Restaurant,Internet, Chinese Restaurant, Party Lawn | 105, Race Course Road Coimbatore TamilNadu 641018 | Medium luxury | True |
| 4 | TAJ DECCAN | 4 | Parking,Fitnesscenter,Meetingrooms,Private dining for party | Road No. 1, Banjara Hills Hyderabad Telangana State 500034 | Budget | True |
| 5 | TAJ BEKAL RESORT | 4 | Spa ,Internet ,Yoga and meditation,parking,internet | Kappil Beach Kasargod Kerala 671319 | Luxury | False |

**Figure 3.4**Code



```xml
<Hotels>
<Hotel id="1">
   <Name>TAJ GANJ</Name>
   <Stars>3</Stars>
   <Facilities>Restaurant,Parking,Internet</Facilities>
   <Address>TajGanj, Fatehabad Road Agra Uttar Pradesh 282001</Address>
   <Type>Budget</Type>
   <Available>True</Available>
</Hotel>

<Hotel id="2">
    <Name>TAJ EXOTICA</Name>
    <Stars>5</Stars>
    <Facilities>Indian therapies,Yoga and meditation,Spaindulges,Parking</Facilities>
    <Address>CalwaddoBenaulim, Salcete Goa 403716</Address>
    <Type>Luxury</Type>
    <Available>False</Available>
</Hotel>

<Hotel id="3">
    <Name>VIVANTA by TAJ</Name>
    <Stars>3</Stars>
    <Facilities>Parking,Restaurant,Internet,Chinese Restaurant, Party Lawn</Facilities>
    <Address>105, Race Course Road Coimbatore TamilNadu 641018</Address>
    <Type>Medium Luxury</Type>
    <Available>True</Available>
</Hotel>

<Hotel id="4">
    <Name>TAJ DECCAN</Name>
    <Stars>4</Stars>
    <Facilities>Parking,Fitnesscenter,Meetingrooms,Private dining for party</Facilities>
    <Address>Road No. 1, Banjara Hills Hyderabad Telangana State 50034</Address>
    <Type>Budget</Type>
    <Available>True</Available>
</Hotel>

<Hotel id="5">
    <Name>TAJ BEKAL RESORT</Name>
    <Stars>4</Stars>
    <Facilities>Spa ,Internet ,Yoga and meditation,parking,internet</Facilities>
    <Address>Kappil Beach Kasargod Kerala 671319</Address>
    <Type>Luxury</Type>
    <Available>False</Available>
</Hotel>
</Hotels>
```

**Figure 3.5**Output 3



```
XML is correct
```

# CHAPTER 5

# VB SCRIPT AND UFT

## 5.3 VBSCRIPT

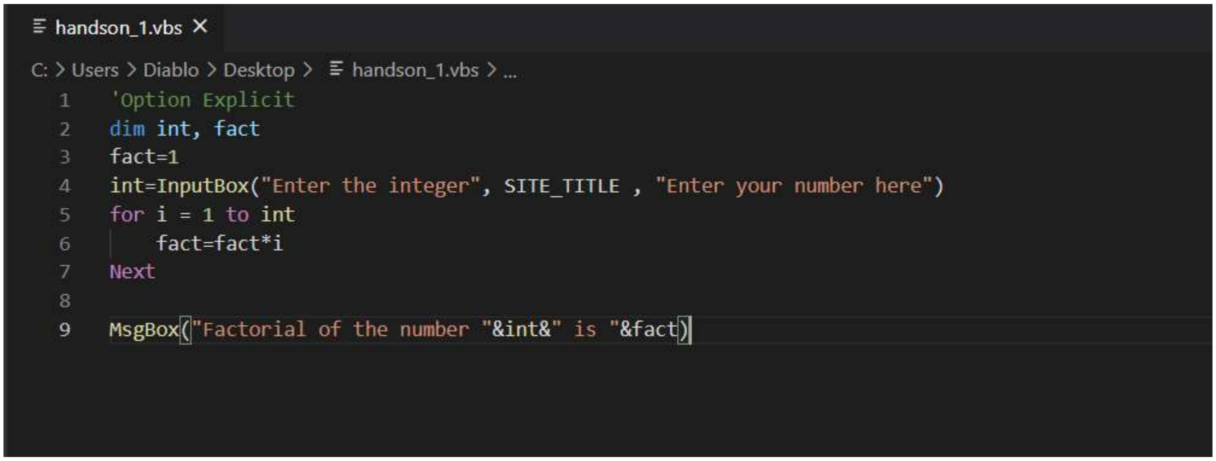
A VBS file is a Virtual Basic script written in the VBScript scripting language. It contains code that can be executed within Windows or Internet Explorer, via the Windows-based script host (Wscript.exe), to perform certain admin and processing functions. VBScript is only a scripting language. So, it cannot run its code on its own. It desires a larger programming language to host it.

## 5.3.2 APPLICATION

### 5.2.1 Elements, Arrays, Procedures, and Program Flow

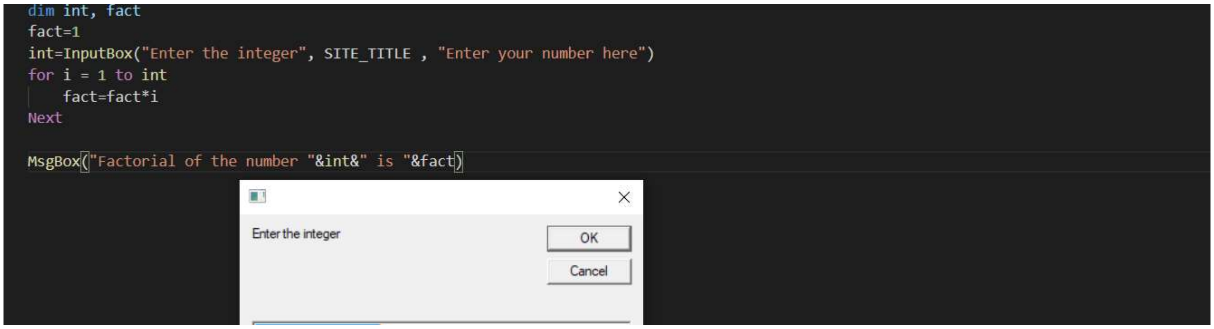
Write a procedure, which will accept an integer value as input and displays the factorial for the given number.

**Figure 4.1**Code Factorial



```
'Option Explicit
dim int, fact
fact=1
int=InputBox("Enter the integer", SITE_TITLE , "Enter your number here")
for i = 1 to int
    fact=fact*i
Next

MsgBox("Factorial of the number "&int&" is "&fact)
```

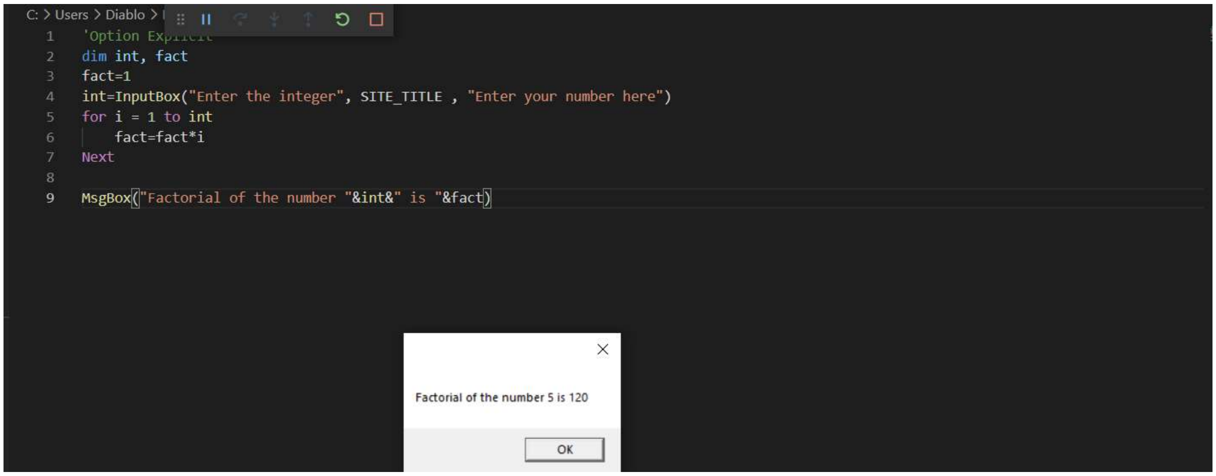
**Figure 4.2**Input Factorial



```
dim int, fact
fact=1
int=InputBox("Enter the integer", SITE_TITLE , "Enter your number here")
for i = 1 to int
    fact=fact*i
Next

MsgBox("Factorial of the number "&int&" is "&fact)
```

Enter the integer

**Figure 4.3**Output Factorial



```
'Option Explicit
dim int, fact
fact=1
int=InputBox("Enter the integer", SITE_TITLE , "Enter your number here")
for i = 1 to int
    fact=fact*i
Next

MsgBox("Factorial of the number "&int&" is "&fact)
```

```
Factorial of the number 5 is 120
```

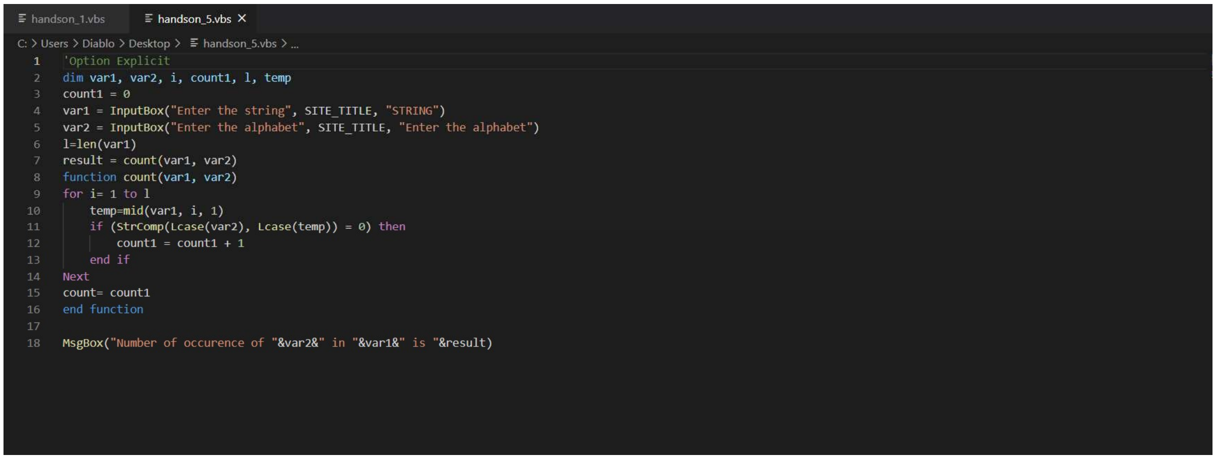
### 5.2.2 Functions, Core Objects, and Classes

Write a function which will accept two inputs

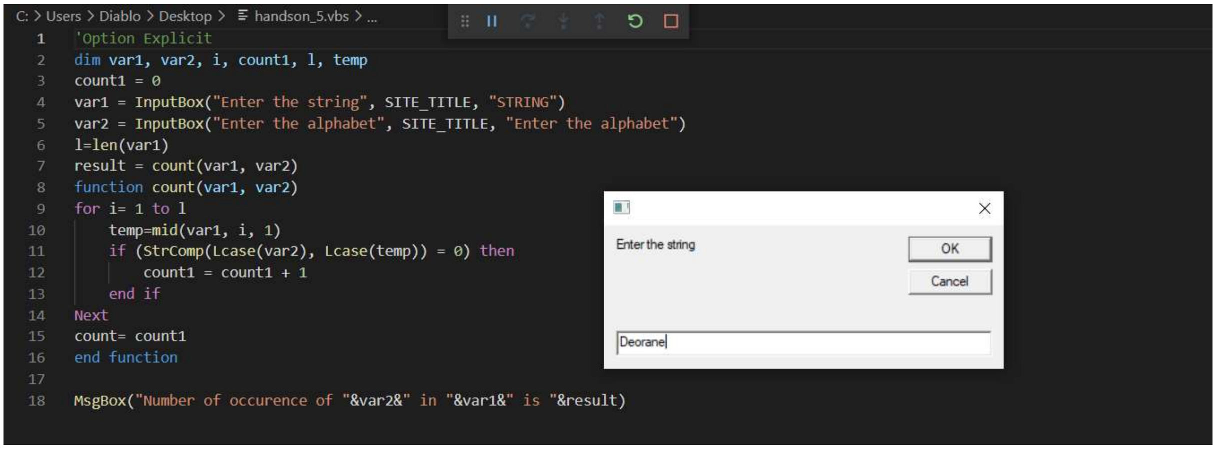
1. Word
2. Alphabet

And the output should be the number of times the alphabet is available in the word.

**Figure 4.4**Code String



```
'Option Explicit
dim var1, var2, i, count1, l, temp
count1 = 0
var1 = InputBox("Enter the string", SITE_TITLE, "STRING")
var2 = InputBox("Enter the alphabet", SITE_TITLE, "Enter the alphabet")
l=len(var1)
result = count(var1, var2)
function count(var1, var2)
for i= 1 to l
    temp=mid(var1, i, 1)
    if (StrComp(Lcase(var2), Lcase(temp)) = 0) then
        count1 = count1 + 1
    end if
Next
count= count1
end function

MsgBox("Number of occurence of "&var2&" in "&var1&" is "&result)
```

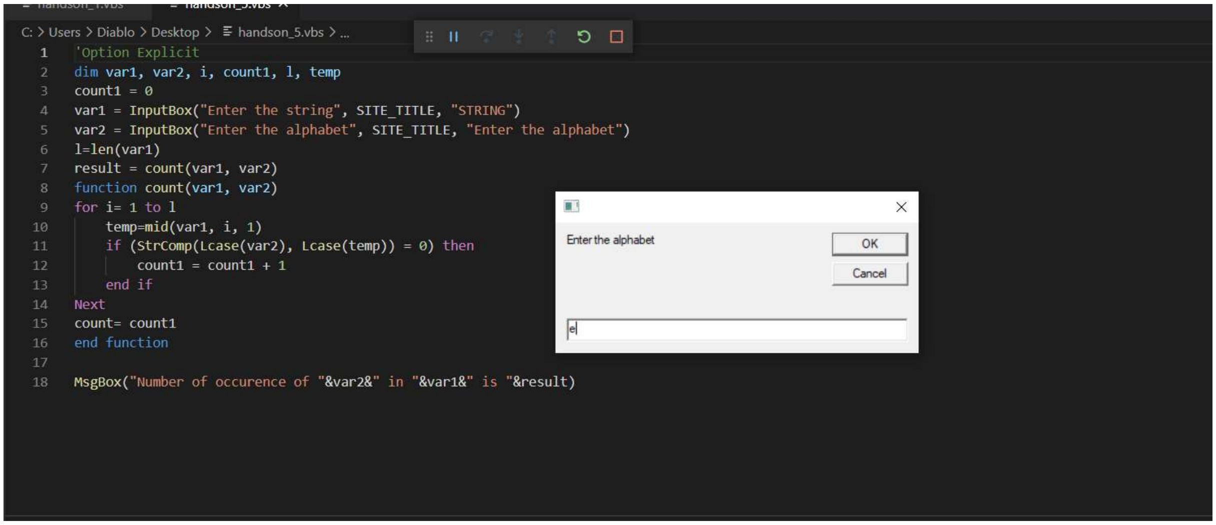


```
'Option Explicit
dim var1, var2, i, count1, l, temp
count1 = 0
var1 = InputBox("Enter the string", SITE_TITLE, "STRING")
var2 = InputBox("Enter the alphabet", SITE_TITLE, "Enter the alphabet")
l=len(var1)
result = count(var1, var2)
function count(var1, var2)
for i= 1 to l
    temp=mid(var1, i, 1)
    if (StrComp(Lcase(var2), Lcase(temp)) = 0) then
        count1 = count1 + 1
    end if
Next
count= count1
end function

MsgBox("Number of occurence of "&var2&" in "&var1&" is "&result)
```

Enter the string

Deorane

**Figure 4.5**Input String

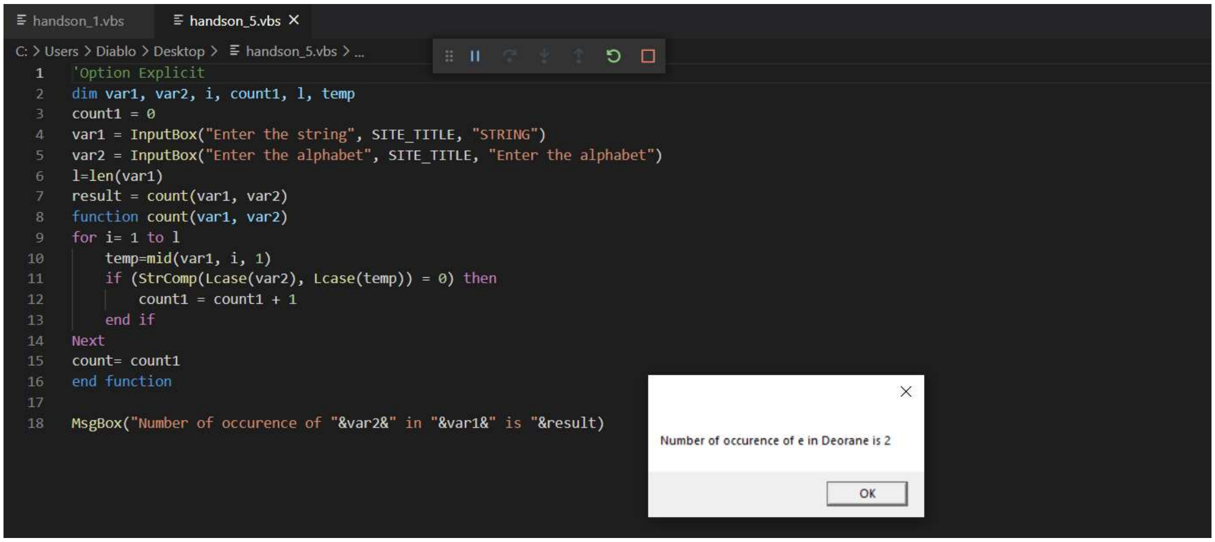


```
'Option Explicit
dim var1, var2, i, count1, l, temp
count1 = 0
var1 = InputBox("Enter the string", SITE_TITLE, "STRING")
var2 = InputBox("Enter the alphabet", SITE_TITLE, "Enter the alphabet")
l=len(var1)
result = count(var1, var2)
function count(var1, var2)
for i= 1 to l
    temp=mid(var1, i, 1)
    if (StrComp(Lcase(var2), Lcase(temp)) = 0) then
        count1 = count1 + 1
    end if
Next
count= count1
end function

MsgBox("Number of occurence of "&var2&" in "&var1&" is "&result)
```

Enter the alphabet

e

**Figure 4.6**Input String 2



```
'Option Explicit
dim var1, var2, i, count1, l, temp
count1 = 0
var1 = InputBox("Enter the string", SITE_TITLE, "STRING")
var2 = InputBox("Enter the alphabet", SITE_TITLE, "Enter the alphabet")
l=len(var1)
result = count(var1, var2)
function count(var1, var2)
for i= 1 to l
    temp=mid(var1, i, 1)
    if (StrComp(Lcase(var2), Lcase(temp)) = 0) then
        count1 = count1 + 1
    end if
Next
count= count1
end function

MsgBox("Number of occurence of "&var2&" in "&var1&" is "&result)
```

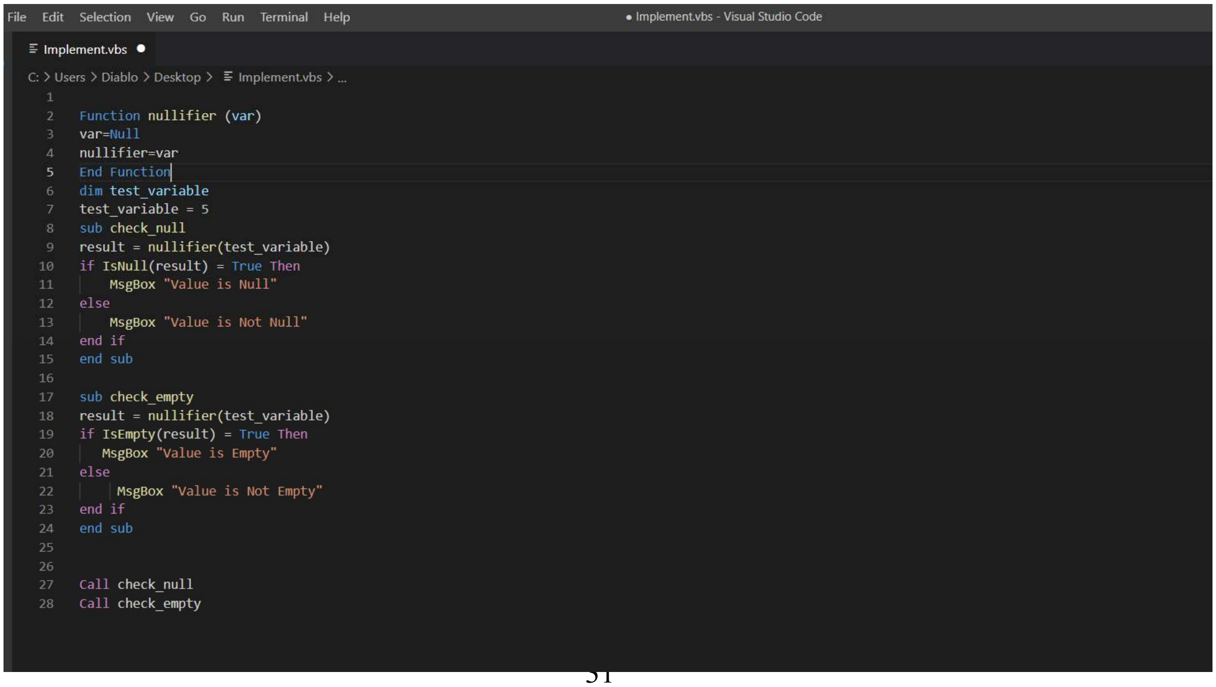
```
Number of occurence of e in Deorane is 2
```

**Figure 4.7**Output String

### 5.2.3 Inspecting Variables Received in Procedures

Return a Null value from a function and evaluate the return value using IsEmpty/IsNull in a sub routine.

**Figure 4.8**Error Handling ISEmpty

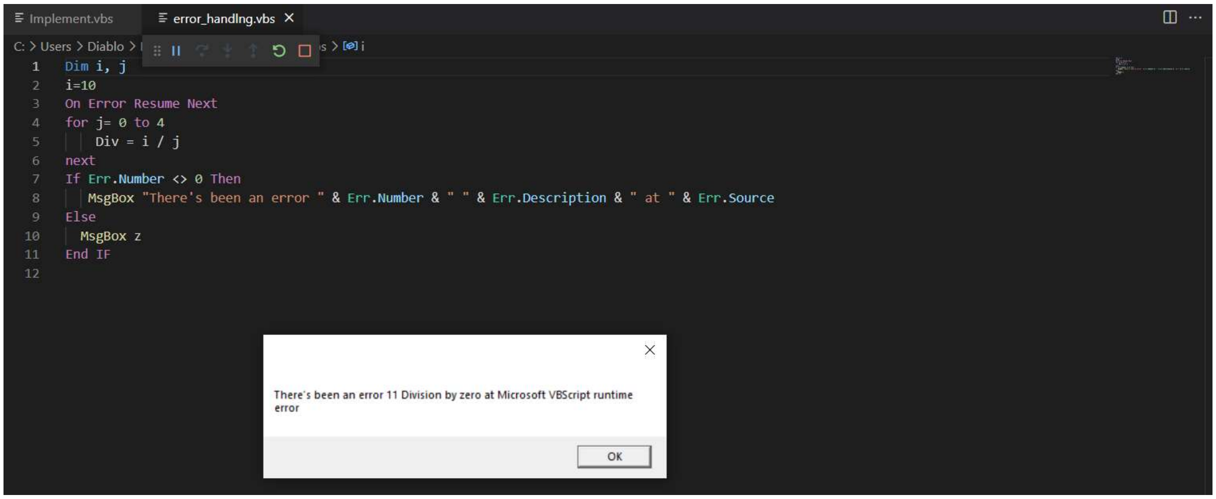

File
Edit
Selection
View
Go
Run
Terminal
Help
● Implement.vbs - Visual Studio Code
Implement.vbs ●
C: > Users > Diablo > Desktop > Implement.vbs > ...

```

Function nullifier (var)
var=Null
nullifier=var
End Function
dim test_variable
test_variable = 5
sub check_null
result = nullifier(test_variable)
if IsNull(result) = True Then
    MsgBox "Value is Null"
else
    MsgBox "Value is Not Null"
end if
end sub

sub check_empty
result = nullifier(test_variable)
if IsEmpty(result) = True Then
   MsgBox "Value is Empty"
else
     MsgBox "Value is Not Empty"
end if
end sub


Call check_null
Call check_empty
```

### 5.2.4 Error Handling Flag and the "Err" Object

Write a program to display the result when an integer is divided by the first five whole numbers. Display the error code and description when an error occurred.

**Figure 4.9** Code Error Handling Division by Zero



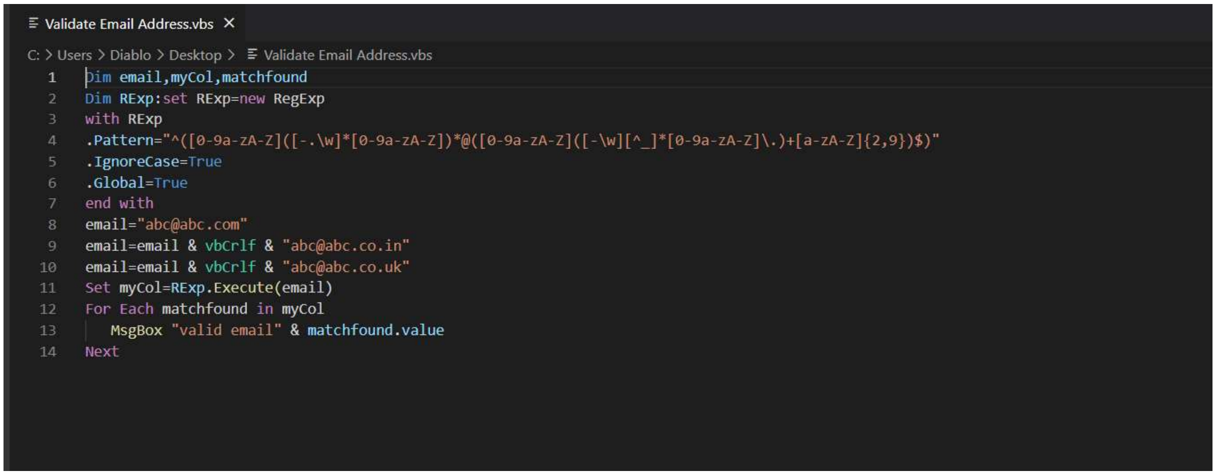
```
Dim i, j
i=10
On Error Resume Next
for j= 0 to 4
    Div = i / j
next
If Err.Number <> 0 Then
  MsgBox "There's been an error " & Err.Number & " " & Err.Description & " at " & Err.Source
Else
  MsgBox z
End IF
```

There's been an error 11 Division by zero at Microsoft VBScript runtime error

### 5.2.5 Regular Expression Pattern Match and Replacement

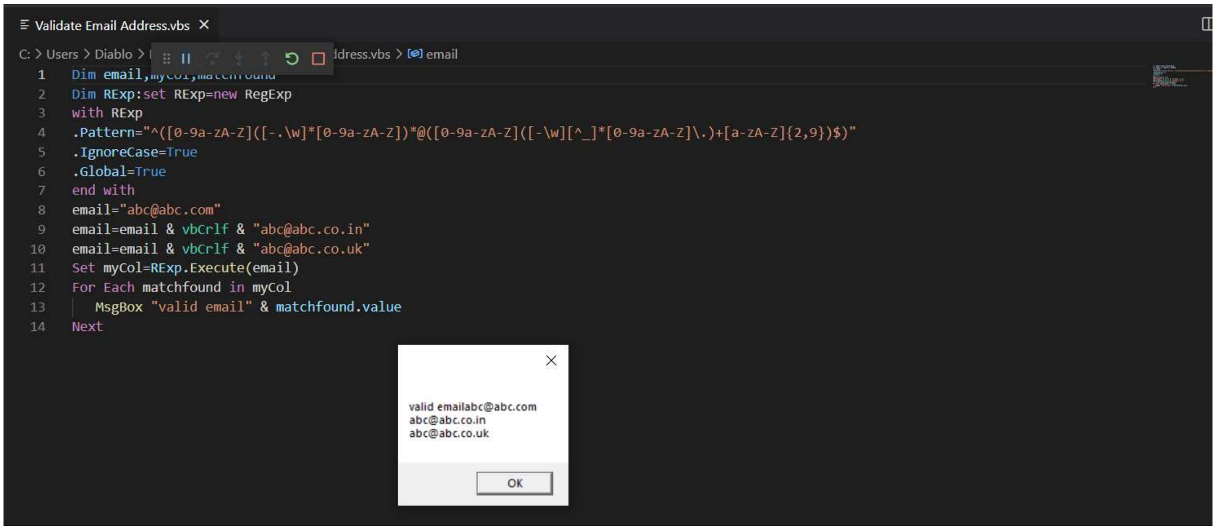
Write a regular expression match for email address. Email format can be of types abc@abc.com, abc@abc.co.in, abc@abc.co.uk

**Figure 4.10** Code Regular Expression


Validate Email Address.vbs
C: > Users > Diablo > Desktop > Validate Email Address.vbs

```
Dim email,myCol,matchfound
Dim RExp:set RExp=new RegExp
with RExp
.Pattern="^([0-9a-zA-Z]([-.\w]*[0-9a-zA-Z])*@([0-9a-zA-Z]([-\w][^_]*[0-9a-zA-Z]\.)+[a-zA-Z]{2,9})$)"
.IgnoreCase=True
.Global=True
end with
email="abc@abc.com"
email=email & vbCrlf & "abc@abc.co.in"
email=email & vbCrlf & "abc@abc.co.uk"
Set myCol=RExp.Execute(email)
For Each matchfound in myCol
  MsgBox "valid email" & matchfound.value
Next
```

**Figure 4.11** Output Regular Expression



Validate Email Address.vbs
C: > Users > Diablo > ddress.vbs > email

```
Dim email,myCol,matchfound
Dim RExp:set RExp=new RegExp
with RExp
.Pattern="^([0-9a-zA-Z]([-.\w]*[0-9a-zA-Z])*@([0-9a-zA-Z]([-\w][^_]*[0-9a-zA-Z]\.)+[a-zA-Z]{2,9})$)"
.IgnoreCase=True
.Global=True
end with
email="abc@abc.com"
email=email & vbCrlf & "abc@abc.co.in"
email=email & vbCrlf & "abc@abc.co.uk"
Set myCol=RExp.Execute(email)
For Each matchfound in myCol
   MsgBox "valid email" & matchfound.value
Next
```

```
valid emailabc@abc.com
abc@abc.co.in
abc@abc.co.uk
```

## 5.3 UFT

UFT is an automatic useful Testing device that facilitates testers to execute computerized checks that allows you to discover any errors, defects or gaps in opposite to the anticipated effects of the utility below test. It turned into designed through Mercury Interactive and afterward obtained through HP and now Microfocus. Full shape of QTP is Quickest Professional at the same time as UFT manner Unified Functional Testing.

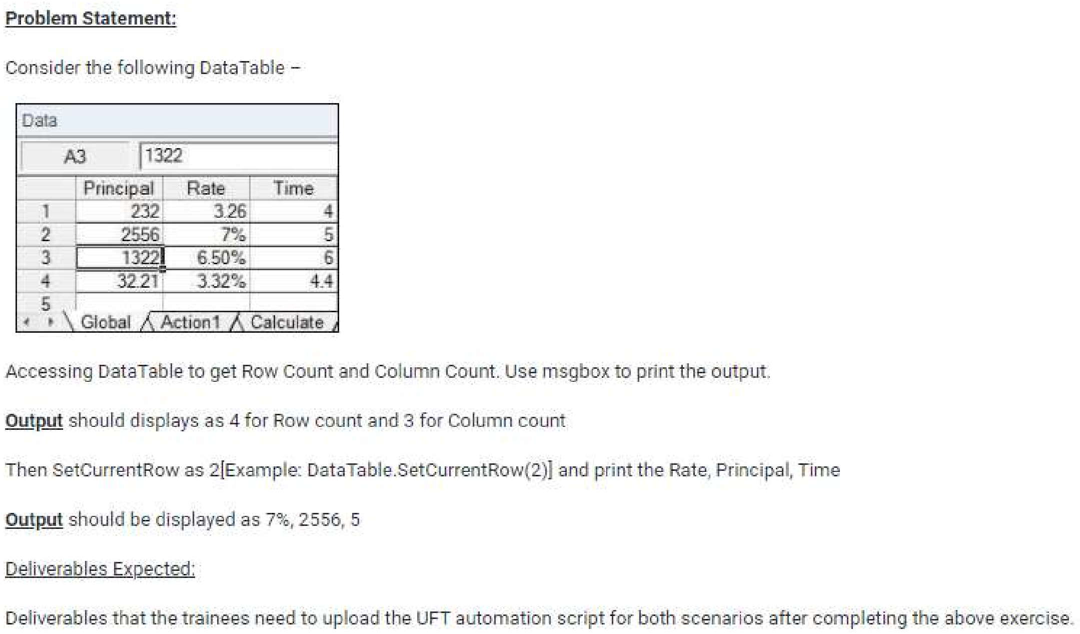
### 5.3.2 ADVANTAGES OF UFT AUTOMATION

- It helps report and playback
- It makes use of a lively display to report scripts and enables tester in referring the display item properties
- It has first rate item identity system or mechanism
- It helps specific add-ins like Oracle, Java, SAP, NET, Web Forms, People soft, etc.
- It permits you to decorate the present checks even without the AUT thru a lively display
- It helps famous automation frameworks- keyword pushed trying out approach, modular trying out approach, data-pushed trying out approach, etc.

- It comes with a built in IDE
- It may be incorporated with Test control equipment like Quality Centre, Test Director, and WinRunner
- Different kinds of suites like Smoke, Regression, Sanity may be without problems maintained
- It helps XML
- Test reporting is feasible thru QTP for evaluation purpose
- Easy to maintain

### 5.3.2 APPLICATION

**Figure 4.11** Problem Statement



**Problem Statement:**

Consider the following DataTable –

| | Principal | Rate | Time |
|---|---|---|---|
| 1 | 232 | 3.26 | 4 |
| 2 | 2556 | 7% | 5 |
| 3 | 1322 | 6.50% | 6 |
| 4 | 32.21 | 3.32% | 4.4 |
| 5 | | | |

Accessing DataTable to get Row Count and Column Count. Use msgbox to print the output.

**Output** should displays as 4 for Row count and 3 for Column count

Then SetCurrentRow as 2[Example: DataTable.SetCurrentRow(2)] and print the Rate, Principal, Time

**Output** should be displayed as 7%, 2556, 5

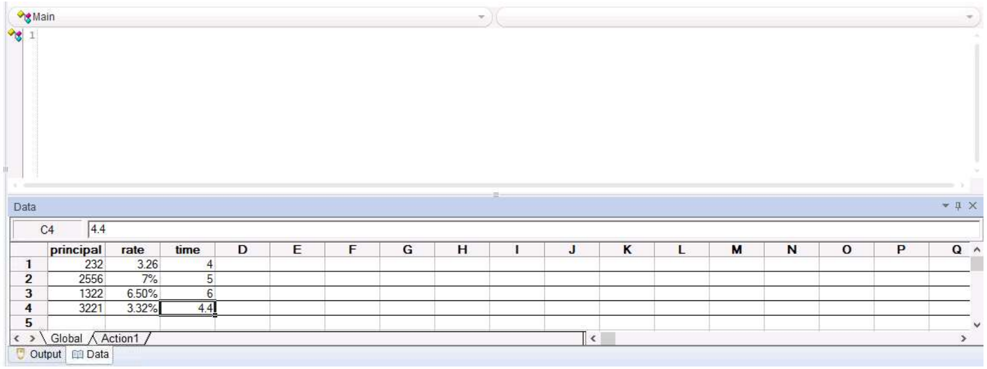
Deliverables Expected:

Deliverables that the trainees need to upload the UFT automation script for both scenarios after completing the above exercise.

**SOLUTION:**

Step 1: Add data to data table

**Figure 4.12** Solution 1

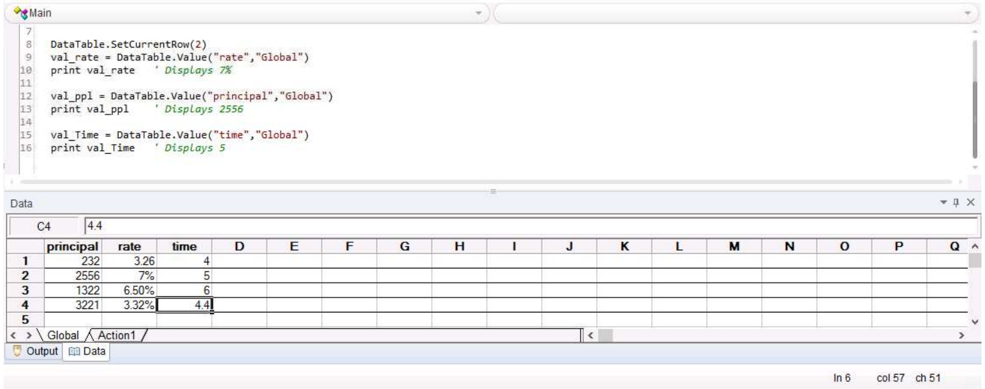


C4 4.4

| | principal | rate | time | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 232 | 3.26 | 4 | | | | | | | | | | | | | | |
| 2 | 2556 | 7% | 5 | | | | | | | | | | | | | | |
| 3 | 1322 | 6.50% | 6 | | | | | | | | | | | | | | |
| 4 | 3221 | 3.32% | 4.4 | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | | |

Step 2: Writing code to in the UFT

**Figure 4.13** Solution 2



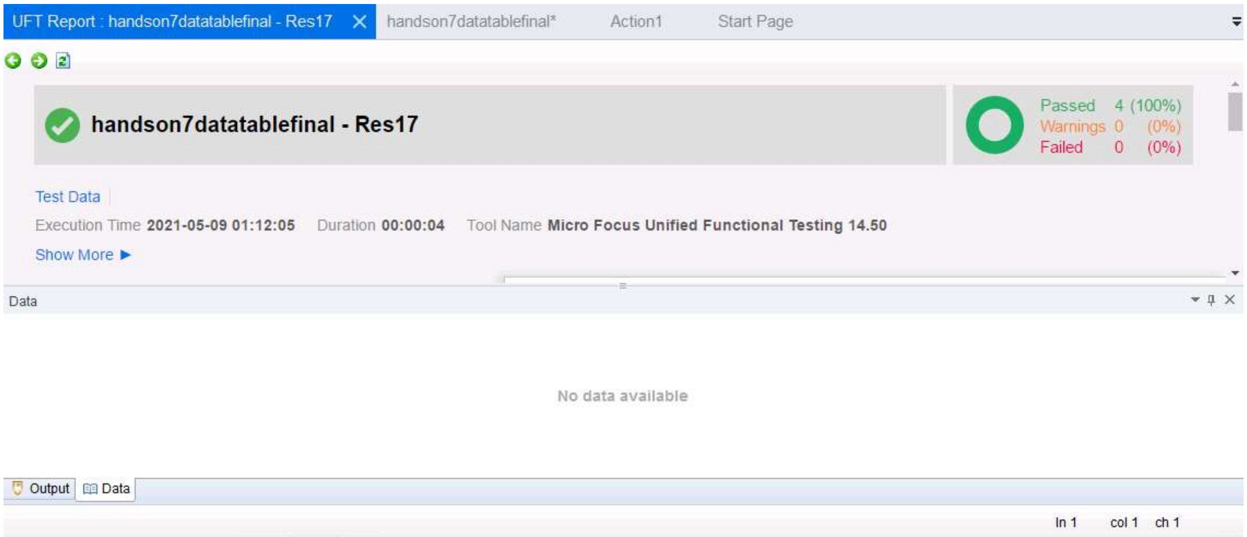
```

DataTable.SetCurrentRow(2)
val_rate = DataTable.Value("rate","Global")
print val_rate   ' Displays 7%

val_ppl = DataTable.Value("principal","Global")
print val_ppl    ' Displays 2556

val_Time = DataTable.Value("time","Global")
print val_Time   ' Displays 5
```

C4 4.4

| | principal | rate | time | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 232 | 3.26 | 4 | | | | | | | | | | | | | | |
| 2 | 2556 | 7% | 5 | | | | | | | | | | | | | | |
| 3 | 1322 | 6.50% | 6 | | | | | | | | | | | | | | |
| 4 | 3221 | 3.32% | 4.4 | | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | | | | | |

Step 3: Report

**Figure 4.14** Solution 3



handson7datatablefinal - Res17

Passed 4 (100%)
Warnings 0 (0%)
Failed 0 (0%)

Test Data

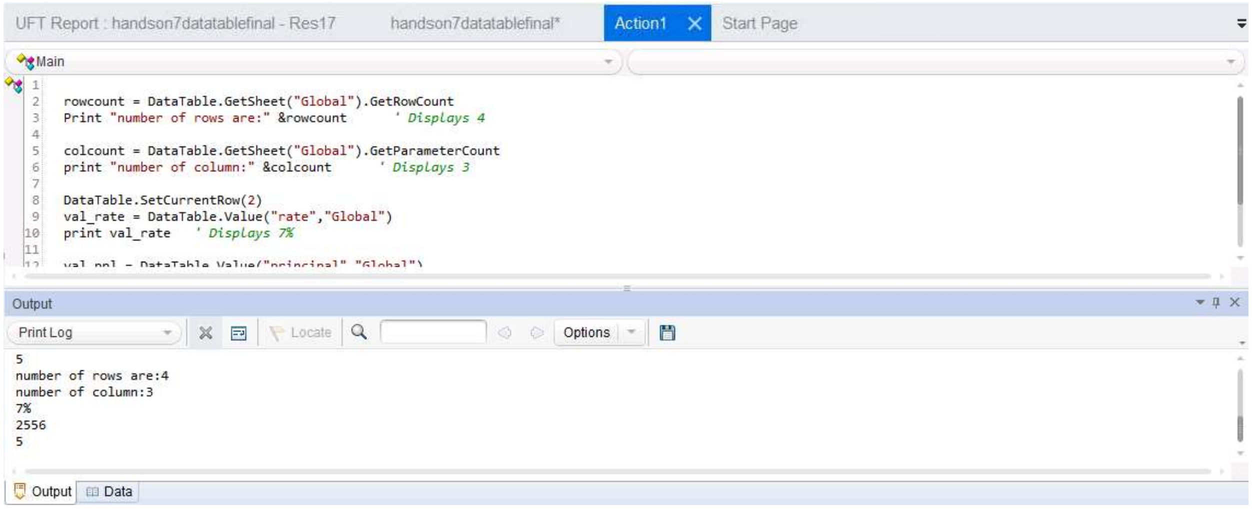
Execution Time **2021-05-09 01:12:05** Duration **00:00:04** Tool Name **Micro Focus Unified Functional Testing 14.50**

Show More ▶

No data available

Step 4: Output

**Figure 4.15** Solution 4



```

rowcount = DataTable.GetSheet("Global").GetRowCount
Print "number of rows are:" &rowcount      ' Displays 4

colcount = DataTable.GetSheet("Global").GetParameterCount
print "number of column:" &colcount      ' Displays 3

DataTable.SetCurrentRow(2)
val_rate = DataTable.Value("rate","Global")
print val_rate   ' Displays 7%

val_ppl = DataTable.Value("principal","Global")
```

```
5
number of rows are:4
number of column:3
7%
2556
5
```

# REFERENCES

1. Everything you need to learn about software testing(Udemy) by **Tarek Roshdy**
2. Understand SQL using the MySQL database. Learn Database Design and Data Analysis with Normalization and Relationships(Udemy) by **Jon Avis**
3. API Documentation1: Teaches technical writers how to document structured data. No programming experience required(Udemy) by **Peter Gruenbaum**
4. Simplest Form of Automation (Udemy) by **Coding Gears**
5. Learn the fundamentals of HP-UFT Functional Automation (Udemy) by **Coding Gears**