

## **Task no.1:**

### **Write a manual test case:**

**Manual testing:** Manual testing is one of the fundamental software testing processes that involves executing test cases manually without using any automated tools. All test cases are executed by the tester manually according to the end user's perspective. It ensures whether the application is working, as mentioned in the requirement document or not.

### **Example of a manual test case:**

Test Case ID: TC\_Login\_001

Test Case Name/Title: Login Functionality Test

Test Objective/Purpose: To verify that the user can successfully log in to the application using valid credentials.

Prerequisites: The user must have a valid account.

Test Steps:

Step 1: Open the application login page.

Step 2: Enter a valid username and password.

Step 3: Click on the login button.

Step 4: Verify that the user is redirected to the home page.

Test Data: Valid username and password.

Expected Result: The user is successfully logged in and redirected to the home page.

Test Environment/Setup: Windows 10, Chrome browser.

Test Execution Instructions: Perform the steps as described above.

Actual Result: The user is successfully logged in.

Pass/Fail: Pass

Comments/Notes: N/A

## Task no. 2:

**Definition with an example:** Unit testing, Smoke testing, User Acceptance, Integration Testing, Regression testing, Performance Testing, Load Testing, Security Testing, Portability Testing, Accountability Testing, Reliability Testing, Efficiency Testing, Volume Testing, Recovery Testing, Responsive Testing, Visual Testing.

**Unit Testing:** Unit Testing is a software testing technique using which individual units of software i.e. group of computer program modules, usage procedures, and operating procedures are tested to determine whether they are suitable for use or not.

**Example:** Adding two numbers in Python

```
def add(a, b):
```

```
    return a + b
```

Unit test for the function:

```
import unittest
```

```
class TestAddFunction(unittest.TestCase):
```

```
    def test_add(self):
```

```
        self.assertEqual(add(1, 2), 3)
```

```
        self.assertEqual(add(-1, 1), 0)
```

```
        self.assertEqual(add(0, 0), 0)
```

```
if __name__ == '__main__':
```

```
    unittest.main()
```

**Smoke Testing:** Smoke testing, also called build verification testing or confidence testing, is a software testing method that is used to determine if a new software build is ready for the next testing phase.

**Example:** Suppose we have a web application with the following critical functions:

1. Login functionality.
2. Home page loading.
3. User registration.

```
import unittest
```

```
from selenium import webdriver
```

```
class SmokeTest(unittest.TestCase):
```

```
    @classmethod
```

```
    def setUpClass(cls):
```

```
        cls.driver = webdriver.Chrome()
```

```

        cls.driver.implicitly_wait(10) # Implicit wait to handle dynamic content loading

def test_login(self):

    self.driver.get('https://example.com/login')

    # Add your login test logic here

    pass

def test_home_page_loading(self):

    self.driver.get('https://example.com/')

    # Add your home page loading test logic here

    pass

def test_user_registration(self):

    self.driver.get('https://example.com/register')

    # Add your user registration test logic here

    pass

@classmethod

def tearDownClass(cls):

    cls.driver.quit()

if __name__ == '__main__':

    unittest.main()

```

**User Acceptance:** User Acceptance Testing (UAT) is a type of testing performed by the end-user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration, and system testing is done.

**Example:** Before releasing a new version of a mobile app, a group of actual users is invited to test the app and provide feedback on its usability, functionality, and overall user experience.

**Integration Testing:** Integration testing is a type of software testing where components of the software are gradually integrated and then tested as a unified group.

**Example:** In a web application, integration testing might involve testing the interaction between the front-end user interface and the back-end database to ensure data is correctly displayed and stored.

**Regression testing:** Regression Testing is the process of testing the modified parts of the code and the parts that might get affected due to the modifications to ensure that no new errors have been introduced in the software after the modifications have been made.

**Example:** After fixing a bug in a software application, regression testing is performed to ensure that the bug fix did not introduce any new issues.

**Performance Testing:** Performance testing is the practice of evaluating how a system performs in terms of responsiveness and stability under a particular workload.

**Example:** A performance test might involve simulating a large number of users accessing a website simultaneously to see how the website performs under high-traffic conditions.

**Load Testing:** Load Testing is a non-functional software testing process in which the performance of a software application is tested under a specific expected load.

**Example:** Load-testing a web server to see how it performs when 1000 users access the website simultaneously.

**Security Testing:** Security Testing is a type of Software Testing that uncovers vulnerabilities in the system and determines that the data and resources of the system are protected from possible intruders.

**Example:** Testing a web application for vulnerabilities such as SQL injection or cross-site scripting (XSS).

**Portability Testing:** Portability testing is the process of testing how easily software or a product can be transferred from one environment to another. It measures the level of effort needed to move it from one system to another.

**Example:** Testing a mobile app to ensure it works correctly on different devices and operating systems.

**Accountability Testing:** Accountability testing is a type of testing that ensures that the software accurately tracks and reports user actions and data.

**Example:** Testing an e-commerce website to ensure that all transactions are recorded accurately and that user accounts are updated correctly.

**Reliability Testing:** Reliability Testing is a software testing process that checks whether the software can perform a failure-free operation in a particular environment for a specified period.

**Example:** Testing a mobile app to ensure that it functions correctly in different network conditions and on different devices.

**Efficiency Testing:** Efficiency testing is the process of testing a software application to ensure that it uses system resources efficiently and performs tasks quickly.

**Example:** Testing a video editing software to ensure that it renders videos quickly and without using excessive system resources.

**Volume Testing:** Volume testing is a type of testing that involves testing a software application with a large amount of data to ensure it can handle the data volume without performance degradation.

**Example:** Volume testing a database to ensure it can handle a large number of records without slowing down.

**Recovery Testing:** Recovery testing is a type of system testing that aims at testing whether a system can recover from failures or not.

**Example:** Recovery testing might involve intentionally crashing a software application and then testing its recovery mechanisms to ensure they work correctly.

**Responsive Testing:** Responsive testing is the process of testing a software application to ensure that it responds quickly to user input.

**Example:** Testing a mobile app to ensure that it responds quickly to user interactions such as button clicks and swipes.

**Visual Testing:** Visual testing is a type of testing that involves testing a software application's user interface to ensure it looks correct and functions as expected.

**Example:** Visual testing might involve comparing screenshots of a web page before and after a change to ensure that the layout and design have not been affected.