**EXPERIMENT NO: -9**

# Aim: Write test cases for black box testing. Explanation:

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.



The above Black-Box can be any software system you want to test. For Example, an operating system like Windows, a website like Google, a database like Oracle or even your own custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

# Types of Black Box Testing

There are many types of Black Box Testing but the following are the prominent ones –

* Functional testing – This black box testing type is related to the functional requirements of a system; it is done by software testers.
* Non-functional testing – This type of black box testing is not related to testing of specific functionality, but non-functional requirements such as performance, scalability, usability.
* Regression testing – Regression Testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code

# Comparison of Black Box and White Box Testing:

Black Box Testing White Box Testing

the main focus of black box testing is on the validation of your functional requirements.

White Box Testing (Unit Testing) validates internal structure and working of your software code

Black box testing gives abstraction from code and focuses on testing effort on the software system behavior.

To conduct White Box Testing, knowledge of underlying programming language is essential. Current day software systems use a variety of programming languages and technologies and its not possible to know all of them.

Black box testing facilitates testing communication amongst modules

White box testing does not facilitate testing communication amongst modules

# Test Cases for Weather Appliction:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **c** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected**  **Results** | **Actual**  **Results** | **Pass/Fail** |
| 1 | Create Webpage to store Data | 1.Write the code into the editor 2.Save the file with .html extension. | Test the above code for creating front page | Display the Webpage content. | As expected | Pass |
| 2 | Apply styles to the webpage | 1.Write the code into the editor 2.Save the file with .css extension. | Test the above code for styels | Show the styles to the webpage | As expected | Pass |
| 3 | Add react icons | 1. Write the code into the editor 2. Save the file with .html extension. | Test the search bar icon | Show the content which user write. | As expected | Pass |
| 4 | Add behaviour to webpage | 1. Write the code into the editor 2. Save the file with .js extension. | Test the above code. | Show the correct information related weather | As expected | Pass |