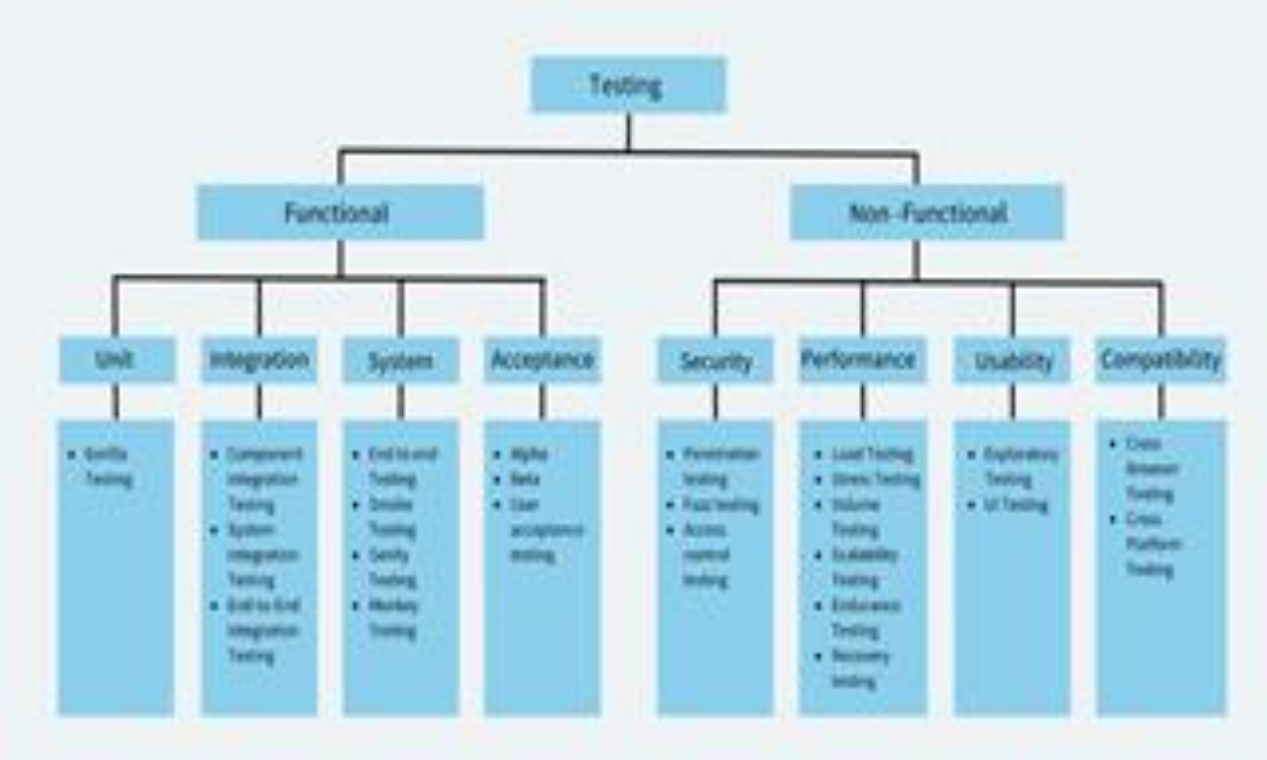
ASSIGNMENT – 6

1) GIVE AN EXAMPLE FOR DIFFERENT TESTING TECHNIQUES :

DEFINITION :

Testing techniques comprise the various ways and angles from which any software can be verified to ensure that it works and appears (UI elements, design) as expected during the planning and requirements gathering stage.

EXAMPLE :



BLACK BOX TESTING

In this case, a tester uses the software to check that it functions accurately in every situation. However, the test remains completely unaware of the software's internal design, back-end architecture, components, and business/technical requirements.
The intent of black-box testing is to find performance errors/deficiencies, entire functions missing (if any), initialization bugs, and glitches that may show up when accessing any external database.

WHITE BOX TESTING

Here, testers verify systems they are deeply acquainted with, sometimes even ones they have created themselves. No wonder white box testing has alternate names like open box testing, clear box testing, and transparent box testing.
White box testing is used to analyze systems, especially when running unit, integration, and system tests.

FUNCTIONAL TESTING

Functional tests are designed and run to verify every function of a website or app. It checks that each function works in line with expectations set out in corresponding requirements documentation.

There are multiple sub-sets of function testing, the most prominent of which are:

- **Unit Testing**: Each individual component is tested before the developer pushes it for merging. Unit tests are created and run by the devs themselves.
- **Integration Testing**: Units are integrated and tested to check that they work together seamlessly.
- **System Testing**: All systems elements, hardware and software, are tested for overall functioning to check that it works according to the system's specific requirements. Regression Testing is a type of System Testing that is performed before every release.
- **Acceptance Testing**: Often called **User Acceptance Testing**, this sub-set of testing puts the software in the hands of a control group of potential users, and note their feedback. It's the app's first test in a truly real-world scenario.

NON FUNCTIONAL TESTING

It's in the name. Non functional tests check the non functional attributes of any software – performance, usability, reliability, security, quality, responsiveness, etc. These tests establish software quality and performance in real user conditions.

The main sub-sets of non-functional testing are:

- **Performance Testing**: Software is tested for how efficiently and resiliently it handles increased loads in traffic or user function. Load testing, stress testing, spike testing, and endurance testing are various ways in which software resilience is verified.
- **Compatibility Testing**: Is your software compatible with different browsers, browser versions, devices (mobile and desktop), operating systems and OS versions? Can a Samsung user play with your app as easily as an iPhone user? Cross Browser Compatibility tests help you check that.
- **Security Testing**: Conducted from the POV of a hacker/attacker, security tests look for gaps in security mechanisms that could be exploited for data theft or for making unauthorized changes.
- **Usability Testing**: Usability tests are run to verify if a software can be used without hassle by actual target users. So, you put it in the hands of a few prospective users, in order to get feedback before its actual deployment.
- **Visual Testing**: Visual tests check if all UI elements are rendering as expected, with the right shape, size, color, font, and placement. The question these tests answer is: Does the software as it was meant to in the requirements?
  Percy and App Percy by BrowserStack allows you to run Automated Visual Regression Tests using Visual Diff.
- **Accessibility Testing**: Accessibility testing evaluates if the software can be used by individuals who are disabled. Its goal is to optimize the apps so that differently-abled users can perform all key actions without external assistance.
- **Responsive Testing**: Responsive tests verify if the app/site renders well on screen sizes and resolutions offered by different devices, mobile, tablets, desktops, etc. A site's responsive design is massively important since most people access the internet from their mobile devices, and expect software to work without hassle on their personal endpoints.