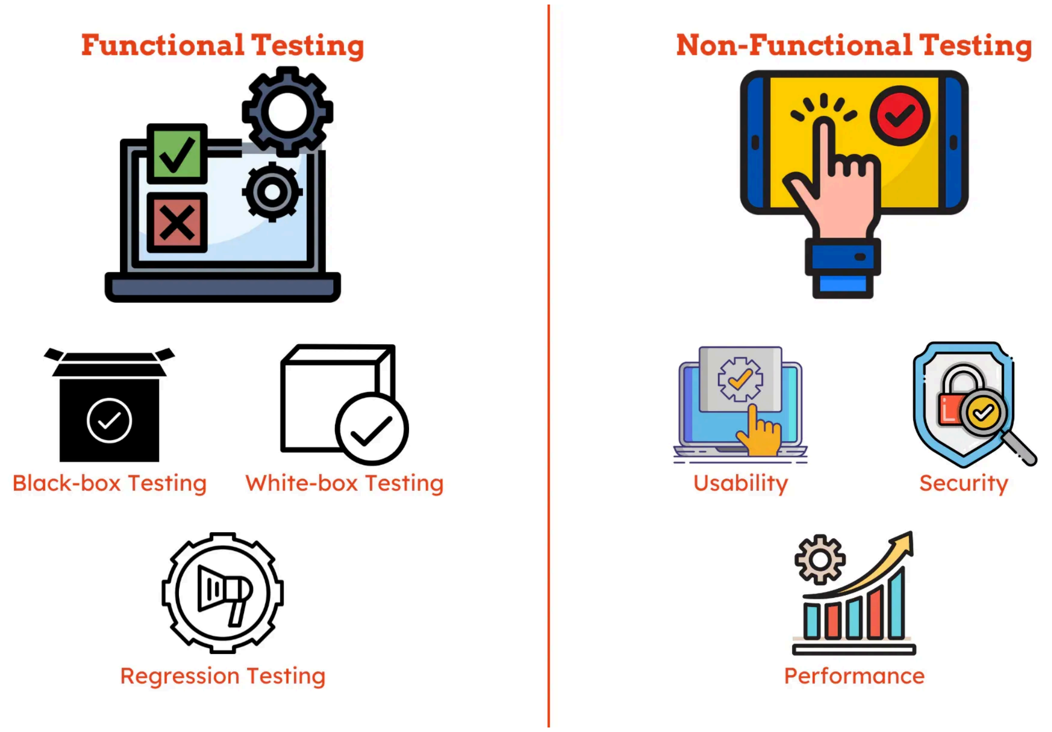
Q:1:- Software Testing Techniques:
- learning about functional and non-functional testing approaches?

Ans: **Software Testing Techniques:**

Software testing is a critical phase of the software development lifecycle, ensuring the quality and reliability of the final product. There are various techniques employed to identify defects and validate functionality.

**Functional Testing:** Functional testing would be like checking if the engine starts, the steering wheel turns the wheels, and the brakes bring the car to a stop. It's all about ensuring the car performs its basic functions as intended. Focuses on what the software does. It verifies if features work according to their specifications and requirements. Think of it as checking off a list of functionalities, ensuring each one performs as designed.

**Non-Functional Testing:** Non-functional testing, on the other hand, would be like taking the car for a spin and evaluating how well it performs those functions. Does the engine roar to life smoothly, or does it sputter and cough? Does the steering respond quickly and precisely, or is there a lag between your turn and the car's movement? These aspects delve into the car's overall experience, not just its basic functionalities. Evaluates how the software performs. It goes beyond the basic "does it work?" question and delves into characteristics like speed, stability, security, usability, and scalability. Here, the tester is like a seasoned driver evaluating the car's handling, comfort, and overall performance.



## Techniques:

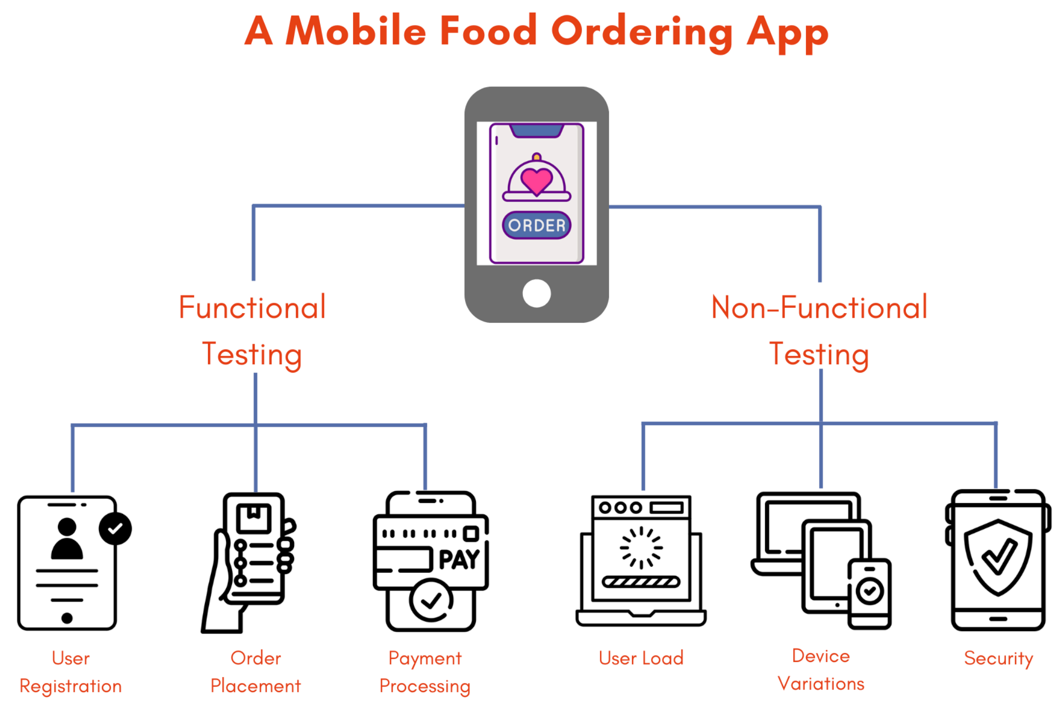
Functional and non-functional testing also use different approaches:

**Functional Testing:** Often employs testing techniques like black-box testing, where the tester interacts with the software without knowing its internal workings, or white-box testing, where the tester has access to the software's code. Regression testing ensures that new features don't break existing functionalities.

**Non-Functional Testing:** This might involve performance testing to measure website loading times under various loads, usability testing to observe how real users interact with the interface, and security testing to identify and plug vulnerabilities.

## Comparison Analysis:

On the other hand, non-functional testing digs deeper. It evaluates the app's performance under real-world conditions, considering factors like user load and device variations. It assesses the user interface's usability, ensuring a smooth and intuitive ordering experience for everyone. Finally, security testing safeguards sensitive user data and ensures the overall safety of the app.



**Functional Testing First:** Functional testing is usually done early in development to make sure the core features work as planned.  Think of it as catching any big mistakes before you invest more time and effort.

**Non-Functional Testing Throughout:** Non-functional testing can happen at different stages. You can check the website's loading speed as features are added, or test the app's usability with real people to see if it's easy to use.

## Conclusion:

In the world of software development, both functional and non-functional testing play vital roles. Functional testing ensures the software fulfils its intended purpose by verifying core features work correctly. Non-functional testing delves deeper, evaluating how well the software performs in terms of speed, usability, security, and other user-centric aspects.

By working together, functional and non-functional testing create robust, high-quality software that not only functions as intended but also provides a positive user experience. This strong alliance ensures the software is built to last and delivers a truly satisfying experience for its users.