\*\*\*\*\*\*\*\*\*\*\*\*\*\*API TESTING:-

an acronym for **Application Programming Interface.**

In software application (app) development, API is the middle layer between the presentation (UI) and the database layer. APIs enable communication and data exchange from one software system to another.

API testing is a software testing practice that tests the APIs directly from their functionality, reliability, performance, to security. Part of integration testing, API testing effectively validates the logic of the build architecture within a short amount of time.

**Where is API testing performed?**

There are three separate layers in a typical app: the **presentation** (or user interface) layer, the **business** layer, and the **database** layer for modeling and manipulating data.

API testing is performed at the most critical layer: business, in which business logic processing is carried out and all transactions between the user interface and database layers happen.

\*\*\*\*\*\*\*Benefits of API Testing:-

### 1.Language-independent:-

Data is exchanged via XML and JSON formats, so any language can be used for test automation. XML and JSON are typically structured data, making the verification fast and stable. There are also built-in libraries to support comparing data using these data formats.

### 2.GUI-independent:-

API testing can be performed in the app prior to GUI testing. Early testing means early feedback and better team productivity. The app's core functionalities can be tested to expose small errors and to evaluate the build's strengths.

### 3.Improved test coverage

Most API/web services have specifications, allowing you to create automated tests with high coverage — including functional testing and non-functional testing.

### 4.Faster releases

It is common that executing API testing saves up to eight hours compared to UI testing, allowing software development teams to release products faster.

**\*\*\*\*\*\*\*\*\*\*\*\***API Testing Types:-

### 1.Validation Testing:-

Validation testing occurs among the final steps and plays an essential role in the development process. It verifies the aspects of product, behavior, and efficiency. In other words, validation testing can be seen as an assurance of the correct development.

### 2. Functional testing:-

Includes testing particular functions in the codebase. These features are the representation of specific scenarios to make sure the API functions are handled well within the planned parameters.

### 3. UI testing:-

UI testing is defined as a test of the user interface for the API and other integral parts. UI testing focuses more on the interface which ties into the API rather than the API testing itself. Although UI testing is not a specific test of API in terms of codebase, this technique still provides an overview of the health, usability, and efficiency of the app’s front and back ends.

### 4. Security testing:-

This practice ensures the API implementation is secure from external threats. Security testing also includes additional steps such as validation of encryption methodologies, and of the design of the API access control. It also includes user rights management and authorization validation.

### 5. Load testing:-

Load testing generally occurs after a specific unit or the whole codebase has been completed. This technique checks if the theoretical solutions work as planned. Load testing monitors the app's performance at both normal and peak conditions.

### 6. Runtime and error detection:-

This testing type is related to the actual running of the API — particularly with the universal results of utilizing the API codebase. This technique focuses on one of the below aspects: monitoring, execution errors, resource leaks, or error detection.

### 7.Penetration testing:-

Penetration testing is considered the second test in the auditing process. In this type, users with limited API knowledge will try to assess the threat vector from an outside perspective, which is about functions, resources, processes, or aim to the entire API and its components.

### 8.Fuzz testing:-

Fuzz testing is another step in the security audit process. In fuzz testing, a vast amount of random data (referred to as "noise" or "fuzz") will be input into the system to detect any forced crashes or negative behaviors. This technique tests the API’s limits to prepare for the "worst-case scenarios."