**Title: Various Types of Testing**

**I. Manual Testing:**

**II. Automated Testing:**

**III. Functional Testing:**

* Explanation of functional testing's purpose.
* Different levels of functional testing (unit, integration, system, acceptance).

**IV. Non-Functional Testing:**

* Tools used for non-functional testing (e.g., JMeter, BurpSuite).

**Title: Testing Techniques**

**I. Black Box Testing:**

**II. White Box Testing:**

**III. Grey Box Testing:**

**IV. Equivalence Partitioning:**

* Equivalence Partitioning divides input data into classes to reduce the number of test cases while maintaining test coverage.

**Title: Defect/Bug life Cycle**

Mainly Five Stages get involved identification, reporting, analysis, fixing, and verification.

**Title: Agile Methodology**

**I. Agile Principles:**

**II. Iterative and Incremental Development:**

**III. Scrum Framework:**

**IV. Continuous Delivery:**

**V. Flexibility and Adaptability:**

**VI. Collaboration and Empowerment:**

**Types of Authentications**

1. **No Auth:** This is the default authorization method in Postman. It means no authentication is applied, and the API endpoint is accessible without any credentials.
2. **Basic Auth:** Basic Authentication is a simple authorization method where the user's credentials (username and password) are encoded and sent as part of the request header.
3. **Bearer Token:** represents the user's identity and access rights. Bearer tokens are widely used for stateless authentication in web APIs.
4. **Digest Auth:** Digest Authentication is a more secure alternative to Basic Authentication. Here the client's password is not sent directly but rather hashed with a server-generated nonce. This makes it more resistant to eavesdropping attacks.
5. **OAuth 2.0:** OAuth 2.0 is a widely adopted authorization framework that enables secure authentication and authorization for APIs. It allows users to grant specific permissions to applications and obtain access tokens to access protected resources.
6. **API Key:**.

**Why to validate Json Schema in Postman?**

 **Data Integrity**

 **Early Detection of Issues:**

 **Security and Data Safety**

**Advanced Xpath in Automation**

**1. Selecting Elements Based on Text:** You can use XPath to locate elements based on their text content. For example, if you want to find a button with a specific label, you can use the

**2. Using Contains and Starts-with Functions:**

**3. Combining Multiple Conditions:**

**4. Using Axes:** XPath axes allow you to navigate through the DOM hierarchy relative to your current element. For instance, you can find the parent, child, sibling, or ancestor elements of a given element:

python

# Finding parent element

parent\_xpath = "//div[@class='child']//parent::div"

# Finding child elements

child\_xpath = "//div[@class='parent']//child::span"

# Finding sibling elements

sibling\_xpath = "//div[@class='current']/following-sibling::div"

**Fixture in Pytest Framework**

Fixtures provide a way to manage test data, configure test environments, and perform other setup and cleanup tasks before and after test execution. They promote reusability and help in maintaining a clean and organized

**Hooks in Pytest-bdd**

**1. before\_feature(context, feature):**

**2. after\_feature(context, feature):**.

**3. before\_scenario(context, scenario):**

**4. after\_scenario(context, scenario**

**5. before\_step(context, step):**

**6. after\_step(context, step):**

**Scenario Outlines**

In pytest-bdd, Scenario Outlines are used to run the same scenario with multiple sets of data. They allow you to define a scenario once and then execute it multiple times with different input data, making it easier to test various scenarios and data combinations without duplicating code. Scenario Outlines are a powerful feature in pytest-bdd and are especially useful for data-driven testing.