**Postman Tool Assignment**

# **Objective**

To equip learners with comprehensive, hands-on experience in API testing and automation using the Postman tool. This assignment will cover key aspects of Postman, from workspace creation to API documentation and CLI execution with Newman.

# **Instructions**

Complete the following tasks sequentially. Each task builds on the previous ones, so make sure to understand and complete each step before moving to the next.

# **Assignment Tasks**

1. **Postman Workspace Creation:**

**Task:** Create a new workspace in Postman.

**Details:** Name your workspace according to the theme or API project you’re working on. A workspace allows you to organize your collections, environments, and other Postman elements in one place.

1. **Collection Creation:**

**Task:** Within your newly created workspace, create a collection.

**Details:** Name your collection descriptively to reflect the API set you will be working with. Collections are used to group related API requests together for easier management.

1. **Add Requests into Collection:**

**Task:** Add at least three API requests (e.g., GET, POST, PUT/DELETE) to your collection.

**Details:** Ensure that each request targets a different endpoint and serves a distinct purpose (e.g., fetching data, creating a new resource, updating existing data).

1. **API Chaining Concept Implementation:**

**Task:** Implement API chaining within your collection.

**Details:** Use the response from one API request as the input for a subsequent request. For example, extract an ID from the first request's response and use it in the URL or body of the next request.

1. **Use of Variables:**

**Global Variable**

**Task:** Define a global variable.

**Details:** Global variables are accessible across all collections and environments within your workspace.

**Collection Variable**

**Task:** Set a collection-level variable.

**Details:** Collection variables are specific to the collection and override global variables if there’s a conflict.

**Environment Variable**

**Task:** Create an environment with relevant variables.

**Details:** Environments are useful for switching between different setups (e.g., Development, Testing, Production) by adjusting variables like URLs, tokens, or credentials.

**Data Variable**

**Task:** Use data variables to run a request with multiple sets of data.

**Details:** Implement data-driven testing by using a CSV or JSON file to provide different data inputs for your requests.

1. **Use of Postman Scripts**

**Pre-Scripts:**

**Task:** Write a pre-request script for one of your API requests.

**Details:** Pre-request scripts run before the request is sent, allowing you to set variables, modify the request, or perform any necessary setup.

**Post Scripts:**

**Task:** Write test scripts for your API requests.

**Details:** Test scripts validate the response, set variables, or perform any additional checks after the request is completed.

1. **Use of Authentication Types:**

**Task:** Implement and test different authentication types for your API requests.

**Details:**

**Basic Authentication:** Username and password are sent in the header.

**Digest Authentication:** More secure than Basic; it hashes credentials.

**API Key Authentication:** Include an API key in the header or URL.

**Bearer Token Authentication:** Use tokens for secure authentication, typically in the header.

1. **Run Collection:**

**Task:** Execute all requests in your collection.

**Details:** Use the “Run” button in Postman to run the collection. Check for any issues and ensure that all requests work as expected.

1. **Export and Import Collection:**

**Task:** Export your collection as a JSON file.

**Details:** This allows you to share or backup your collection. Then, import the JSON file into a new Postman workspace to verify that the collection imports correctly.

1. **Use of Curl Command:**

**Task:** Export one of your API requests as a Curl command.

**Details:** Use the terminal or command line to execute the Curl command, testing the API endpoint outside of Postman.

1. **Use of Newman for CLI Execution:**

**Task:** Use the Newman tool to execute your collection from the command line.

**Details:**

**Installation:** Install Newman via npm (Node Package Manager).

**Execution:** Run your collection using Newman and generate an HTML report to summarize the results.

**Documentation:** Explore the different options available in Newman for running collections, such as specifying environments, data files, or exporting reports.

1. **Postman API Documentation:**

**Task:** Generate API documentation using Postman’s built-in features.

**Details:** Document each request in your collection, including descriptions, sample responses, and usage instructions. Publish this documentation and share the link with your classmates or instructor.

# **Submission Requirements**

* Submit the exported Postman collection JSON file.
* Provide the published API documentation link.
* Submit the HTML report generated by Newman.