National University of Computer and Emerging Sciences Chiniot-Faisalabad Campus



**metabase**

Software Quality Engineering

### Semester Project

### Phase 2

### BS (SE) 5A

## Team Name: zzwave-testing

## Team Member’s:

## 21F-9516 -> Suresh Kumar

## 21F-9519 -> Rai umer farooq

## 21f-9510 -> mian fahad akhtar

# **Test Cases of API: -**

**Authentication Success:**

* Request: Send valid credentials to the authentication endpoint.
* Expected Response: Receive a valid authentication token.

**Authentication Failure:**

* Request: Send invalid or expired credentials to the authentication endpoint.
* Expected Response: Receive an error response indicating authentication failure.

**Get Action List:**

* Request: Retrieve the list of available actions with a valid authentication token.
* Expected Response: Receive a list of actions with their details.

**Execute Action:**

* Request: Execute a specific action with a valid authentication token.
* Expected Response: Receive a success response, and the system performs the action.

**Execute Critical Action with Confirmation:**

* Request: Execute a critical action with confirmation.
* Expected Response: Receive a confirmation prompt, and upon confirmation, the action is performed.

**Invalid Action Execution:**

* Request: Execute an action with invalid ID or missing parameters.
* Expected Response: Receive an error response indicating the issue.

**Get Action History:**

* Request: Retrieve the action history with a valid authentication token.
* Expected Response: Receive a list of executed actions with timestamps.

**Filter Actions:**

* Request: Retrieve actions based on specific filters.
* Expected Response: Receive a list of actions matching the filters.

**Invalid Filter Parameters:**

* Request: Retrieve actions with invalid filter parameters.
* Expected Response: Receive an error response for invalid filters.

**Update Action:**

* Request: Update a specific action with valid authentication token.
* Expected Response: Receive a success response, and the action details are updated.

**Invalid Action Update:**

* Request: Update an action with an invalid ID or missing parameters.
* Expected Response: Receive an error response indicating the issue.

**Delete Action:**

* Request: Delete a specific action with a valid authentication token.
* Expected Response: Receive a success response, and the action is deleted.

**Invalid Action Deletion:**

* Request: Delete an action with an invalid ID.
* Expected Response: Receive an error response indicating the issue.

**Rate Limiting:**

* Request: Exceed the allowed request limit.
* Expected Response: Receive a rate-limiting error after exceeding the limit.

**Error Handling:**

* Request: Send malformed or invalid requests to various endpoints.
* Expected Response: Receive appropriate error responses for different error scenarios.

# **Gherkin Language for API: -**

* Feature: MetaBase Action Module API
* Scenario: Authentication Success
* Given valid credentials
* When a request is made to the authentication endpoint
* Then a valid authentication token should be received
* Scenario: Authentication Failure
* Given invalid or expired credentials
* When a request is made to the authentication endpoint
* Then an error response indicating authentication failure should be received
* Scenario: Get Action List
* Given a valid authentication token
* When a request is made to retrieve the list of available actions
* Then a list of actions with their details should be received
* Scenario: Execute Action
* Given a valid authentication token
* When a request is made to execute a specific action
* Then a success response should be received, and the action should be performed
* Scenario: Execute Critical Action with Confirmation
* Given a request to execute a critical action with confirmation
* When the confirmation is received
* Then the action should be performed
* Scenario: Invalid Action Execution
* Given an action with an invalid ID or missing parameters
* When a request is made to execute the action
* Then an error response indicating the issue should be received
* Scenario: Get Action History
* Given a valid authentication token
* When a request is made to retrieve the action history
* Then a list of executed actions with timestamps should be received
* Scenario: Filter Actions
* Given a valid authentication token
* When a request is made to retrieve actions based on specific filters
* Then a list of actions matching the filters should be received
* Scenario: Invalid Filter Parameters
* Given invalid filter parameters
* When a request is made to retrieve actions
* Then an error response for invalid filters should be received
* Scenario: Update Action
* Given a valid authentication token
* When a request is made to update a specific action
* Then a success response should be received, and the action details should be updated
* Scenario: Invalid Action Update
* Given an action with an invalid ID or missing parameters
* When a request is made to update the action
* Then an error response indicating the issue should be received
* Scenario: Delete Action
* Given a valid authentication token
* When a request is made to delete a specific action
* Then a success response should be received, and the action should be deleted
* Scenario: Invalid Action Deletion
* Given an action with an invalid ID
* When a request is made to delete the action
* Then an error response indicating the issue should be received
* Scenario: Rate Limiting
* Given a request exceeding the allowed limit
* When requests are made beyond the limit
* Then a rate-limiting error should be received
* Scenario: Error Handling
* Given malformed or invalid requests
* When requests are made to various endpoints
* Then appropriate error responses for different error scenarios should be received