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: -**

## Test Case 1: Authentication Test

Test Steps:

1. Obtain a valid authentication token.
2. Call the Report Generation API endpoint with the valid token.
3. Verify that the API responds successfully (HTTP status code 200).
4. Confirm that the API request is authenticated.

## Test Case 2: Invalid Token Handling

Test Steps:

1. Obtain an invalid authentication token.
2. Call the Report Generation API endpoint with the invalid token.
3. Verify that the API responds with an appropriate error status code (e.g., 401 or 403).
4. Confirm that the API request is rejected due to the invalid token.

## Test Case 3: Report Format Selection

Test Steps:

1. Prepare a request for report generation.
2. Send a request to the API specifying a report format (e.g., PDF, CSV, or Excel).
3. Verify that the API responds with the chosen report format.
4. Confirm that the user receives the report in the selected format.

## Test Case 4: Data Range Selection

Test Steps:

1. Prepare a request for report generation.
2. Specify a date range in the API request, starting from 2023-01-01 to 2023-12-31.
3. Call the API with the request.
4. Verify that the API returns data within the specified date range.
5. Confirm that the user receives the report data for the defined period.

## Test Case 5: Data Filters

Test Steps:

1. Prepare a request for report generation.
2. Include specific parameters in the API request, such as filtering by category (e.g., Sales) and region (e.g., North).
3. Send the request to the API.
4. Verify that the API returns data filtered based on the provided parameters.
5. Confirm that the user receives the filtered data in the report.

## Test Case 6: Error Handling

Test Steps:

1. Initiate a report request without sufficient data.
2. Call the API for the report generation.
3. Verify that the API responds with an appropriate error code and message indicating insufficient data.
4. Confirm that the API informs the user about the insufficient data.

## Test Case 7: Multiple Reports Handling

Test Steps:

1. Initiate the generation of multiple reports simultaneously.
2. Send requests to the API for multiple reports.
3. Verify that the API processes each request independently without interference.
4. Confirm that the user receives all requested reports without issues.

## Test Case 8: Report Customization

Test Steps:

1. Prepare a request for report generation.
2. Include customization options in the API request.
3. Call the API with the customized request.
4. Verify that the API generates the report with the customized parameters.
5. Confirm that the user receives the report with the desired customization.

## Test Case 9: Report Sharing

Test Steps:

1. Implement the 'Share' functionality in the API.
2. Use the 'Share' functionality in the API to send a report to a specified email address (e.g., [example@example.com](mailto:example@example.com)).
3. Verify that the API sends the report to the provided email address.
4. Confirm that the user receives the report at the specified email address.

## Test Case 10: Report Archival

Test Steps:

1. Utilize the archival feature in the API to store a generated report for future reference.
2. Verify that the API successfully stores the report for future retrieval.
3. Confirm that the user can access archived reports when needed.

## Test Case 11: API Response Format

Test Steps:

1. Check the format of the API response data.
2. Verify that the API returns data in the expected format (e.g., JSON or XML).
3. Confirm that the user can process the API response data easily.

## Test Case 12: Rate Limiting Test

Test Steps:

1. Generate multiple requests and send them to the API within a short time frame.
2. Verify that the API enforces rate limits and prevents excessive requests.
3. Confirm that the system enforces rate limits on the API to maintain system stability.

These test cases cover a range of scenarios and functionality for the Report Generation API.

# **Gherkin Language for API: -**

Feature: Report Generation API Functionality

Scenario: Authentication Test

Given a valid authentication token

When the Report Generation API endpoint is called with the valid token

Then the API should respond successfully

And the API request should be authenticated

Scenario: Invalid Token Handling

Given an invalid authentication token

When the Report Generation API endpoint is called with an invalid token

Then the API should respond with an appropriate error status code (e.g., 401 or 403)

And the API request should be rejected due to the invalid token

Scenario: Report Format Selection

Given a request for report generation

When a request is sent to the API specifying a report format (e.g., PDF, CSV, or Excel)

Then the API should respond with the chosen report format

And the user should receive the report in the selected format

Scenario: Data Range Selection

Given the need to define the data range for the report

When the date range is specified in the API request, starting from 2023-01-01 to 2023-12-31

Then the API should return data within the specified date range

And the user should receive the report data for the defined period

Scenario: Data Filters

Given the requirement to apply filters to the report

When specific parameters are included in the API request, filtering by category (e.g., Sales) and region (e.g., North)

Then the API should return data filtered based on the provided parameters

And the user should receive the filtered data in the report

Scenario: Error Handling

Given the possibility of insufficient data for the report

When a report is requested without sufficient data

Then the API should respond with an appropriate error code and message indicating insufficient data

And the API should inform the user about the insufficient data

Scenario: Multiple Reports Handling

Given the need for concurrent report generation

When the generation of multiple reports is initiated simultaneously

Then the API should process each request independently without interference

And the user should receive all requested reports without issues

Scenario: Report Customization

Given the requirement for customizing report parameters

When customization options are included in the API request

Then the API should generate the report with the customized parameters

And the user should receive the report with the desired customization

Scenario: Report Sharing

Given the implementation of the 'Share' functionality in the API

When the 'Share' functionality is implemented in the API to send a report to a specified email address (e.g., example@example.com)

Then the API should send the report to the provided email address

And the user should receive the report at the specified email address

Scenario: Report Archival

Given the need to store generated reports for future reference

When the archival feature is utilized in the API

Then the API should store the generated report for future retrieval

And the user should be able to access archived reports when needed

Scenario: API Response Format

Given the requirement for response format validation

When the format of the response data is checked

Then the API should return data in the expected format (e.g., JSON or XML)

And the user should be able to process the API response data easily

Scenario: Rate Limiting Test

Given the necessity for API rate limiting

When multiple requests are sent to the API within a short time frame

Then the API should enforce rate limits and prevent excessive requests

And the system should enforce rate limits on the API