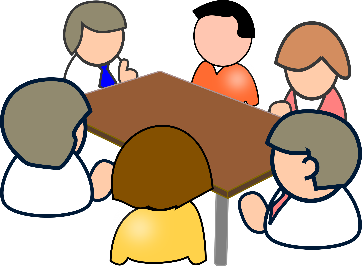
National University of Computer and Emerging Sciences Chiniot-Faisalabad Campus



**META-BASE**

Software Quality Engineering

### Semester Project

### Phase 1ST

### BS (SE) 5A

**API Test Cases:**

**1. Retrieve Dashboard Data:**

Objective: Verify that the API endpoint successfully retrieves dashboard data.

Test Case ID: API\_TC\_01

Test Case Description: Ensure the API endpoint retrieves dashboard data correctly.

Steps:

1. Call the API endpoint to retrieve dashboard data.

2. Validate that the response status code is 200 (OK).

3. Check that the response contains the expected dashboard information.

**2. Execute Query via API:**

Objective: Verify that the API endpoint executes database queries.

Test Case ID: API\_TC\_02

Test Case Description: Confirm that the API endpoint executes a database query.

Steps:

1. Call the API endpoint to execute a database query.

2. Validate that the response status code is 200 (OK).

3. Verify that the response contains the expected query results.

**3. Filter Data via API:**

Objective: Ensure that the API endpoint can filter data.

Test Case ID: API\_TC\_03

Test Case Description: Test the ability of the API to filter data.

Steps:

1. Call the API endpoint to filter data with specific parameters.

2. Validate that the response status code is 200 (OK).

3. Check that the response contains the filtered data.

**4. Export Query Results via API:**

Objective: Verify that the API endpoint exports query results.

Test Case ID: API\_TC\_04

Test Case Description: Test the API endpoint for exporting query results.

Steps:

1. Call the API endpoint to export query results.

2. Validate that the response status code is 200 (OK).

3. Verify that the response provides a downloadable link.

**5. User Permissions via API:**

Objective: Confirm that the API enforces user permissions.

Test Case ID: API\_TC\_05

Test Case Description: Ensure the API enforces user permissions.

Steps:

1. Call the API endpoint with a user having limited access.

2. Validate that the response status code is 403 (Forbidden).

**6. Data Integrity via API:**

Objective: Validate that API updates maintain data integrity.

Test Case ID: API\_TC\_06

Test Case Description: Verify that API updates maintain data integrity.

Steps:

1. Call the API endpoint to update data.

2. Validate that the data integrity is maintained.

**7. Password Change via API:**

Objective: Confirm that the API allows users to change passwords.

Test Case ID: API\_TC\_07

Test Case Description: Test the API endpoint for changing passwords.

Steps:

1. Call the API endpoint to change the user's password.

2. Validate that the response status code is 200 (OK).

3. Verify that the password is updated successfully.

**8. Session Timeout via API:**

Objective: Verify that the API handles session timeouts.

Test Case ID: API\_TC\_08

Test Case Description: Test the API for handling session timeouts.

Steps:

1. Call the API with a valid authentication token.

2. Wait for a session timeout period.

3. Validate that the response status code is 401 (Unauthorized).

**9. Dashboard Customization via API:**

Objective: Verify that the API allows customization of the dashboard.

Test Case ID: API\_TC\_09

Test Case Description: Test the API for adding a widget to the dashboard.

Steps:

1. Call the API endpoint to add a widget to the dashboard.

2. Validate that the response status code is 200 (OK).

3. Verify that the widget is displayed on the dashboard.

**10. Error Handling via API:**

Objective: Confirm that the API returns meaningful error messages.

Test Case ID: API\_TC\_10

Test Case Description: Test the API for clear error messages.

Steps:

1. Call the API endpoint with an invalid query.

2. Validate that the response status code is 400 (Bad Request).

3. Verify that the response contains a clear error message.

**11. Data Deletion via API:**

Objective: Verify that the API allows users to delete records.

Test Case ID: API\_TC\_11

Test Case Description: Test the API endpoint for deleting a record.

Steps:

1. Call the API endpoint to delete a record.

2. Validate that the response status code is 200 (OK).

3. Verify that the record is deleted successfully.

**12. Access Control via API:**

Objective: Confirm that the API enforces access control.

Test Case ID: API\_TC\_12

Test Case Description: Ensure the API enforces access control.

Steps:

1. Call a restricted API endpoint without authentication.

2. Validate that the response status code is 401 (Unauthorized).

**13. Query History via API:**

Objective: Verify that the API endpoint retrieves query history.

Test Case ID: API\_TC\_13

Test Case Description: Test the API for retrieving query history.

Steps:

1. Call the API endpoint to retrieve query history.

2. Validate that the response status code is 200 (OK).

3. Verify that the response contains the user's query history.

**14. Data Validation via API:**

Objective: Confirm that API responses contain valid data.

Test Case ID: API\_TC\_14

Test Case Description: Test the API responses for valid data.

Steps:

1. Call the API endpoint.

2. Validate that the response adheres to the expected data structure.

**15. Pagination via API:**

Objective: Verify that the API supports paginated results.

Test Case ID: API\_TC\_15

Test Case Description: Test the API for paginated results.

Steps:

1. Call the API endpoint with pagination parameters.

2. Validate that the response status code is 200 (OK).

3. Verify that the response contains the specified page of results.

**GHERKIN:-**

**1. Retrieve Dashboard Data:**

Objective: Verify that API endpoint retrieves dashboard data.

Feature: Retrieve Dashboard Data

Scenario: Get Dashboard Data

Given a valid authentication token

When the API endpoint to retrieve dashboard data is called

Then the response should contain the expected dashboard information

**2. Execute Query:**

Objective: Verify that API endpoint executes database queries.

Feature: Execute Database Query via API

Scenario: Execute Query

Given a valid authentication token

When the API endpoint to execute a database query is called

Then the response should contain the expected query results

**3. Filter Data via API:**

Objective: Verify that API endpoint can filter data.

Feature: Filter Data via API

Scenario: Apply Filter

Given a valid authentication token

When the API endpoint to filter data is called with specific parameters

Then the response should contain the filtered data

**4. Export Query Results via API:**

Objective: Verify that API endpoint exports query results.

Feature: Export Query Results via API

Scenario: Export Data

Given a valid authentication token

When the API endpoint to export query results is called

Then the response should provide a downloadable link

**5. User Permissions via API:**

Objective: Verify that API enforces user permissions.

Feature: User Permissions via API

Scenario: Restricted Access

Given a user with limited API access

When they attempt to access restricted data via the API

Then the API should return an access denied response

**6. Data Integrity via API:**

Objective: Verify that API updates maintain data integrity.

Feature: Data Integrity via API

Scenario: Update Data

Given a valid authentication token

When the API endpoint to update data is called

Then the data integrity should be maintained

**7. Password Change via API:**

Objective: Verify that API allows users to change passwords.

Feature: Change Password via API

Scenario: Password Update

Given a valid authentication token

When the API endpoint to change the password is called

Then the password should be updated successfully

**8. Session Timeout via API:**

Objective: Verify that API handles session timeouts.

Feature: Session Timeout via API

Scenario: Inactivity Timeout

Given a valid authentication token

When there is no activity for a specified time

Then the API should return a session timeout response

**9. Dashboard Customization via API:**

Objective: Verify that API allows customization of the dashboard.

Feature: Dashboard Customization via API

Scenario: Add Widget

Given a valid authentication token

When the API endpoint to add a widget to the dashboard is called

Then the response should indicate successful customization

**10. Error Handling via API:**

Objective: Verify that API returns meaningful error messages.

Feature: Error Handling via API

Scenario: Invalid Query

Given a valid authentication token

When the API endpoint to execute an invalid query is called

Then the API should return a clear error message