SQE PROJECT PHASE 2

Noman

This is Phase 2 of Project till task 3

Task #1:

Metabase has been installed

Task #2: API

ECP and BVA , Risk analysis .

**Test Case 1:** User Authentication (ECP & BVA)

**ECP:**

Equivalence Classes:

Valid User Credentials (VUC)

Invalid User Credentials (IUC)

ECP in Mathematical Form: {ECP = {VUC, IUC}}

BVA:

**Boundary Values:**

Minimum Username and Password Length

Maximum Username and Password Length

BVA in Mathematical Form: {BVA = {Min Length, Max Length}}

**Risk Analysis:**

**Risks:**

Unauthorized access due to weak authentication.

Lockout risks if incorrect credentials are repeatedly entered.

**Test Case 2:**

Role-Based Access Control (ECP & BVA)

**ECP:**

**Equivalence Classes:**

User Role (Admin, Non-Admin)

ECP in Mathematical Form: {ECP = {Admin, Non-Admin}}

BVA:

Boundary Values: None in this case.

BVA in Mathematical Form: {BVA = {}}

**Risk Analysis:**

**Risks:**

Unauthorized access by non-admin users to admin-only resources.

Security risks associated with admin actions, if not properly controlled.

Test Case 3: Password Complexity (ECP & BVA)

**ECP:**

**Equivalence Classes:**

Password Complexity (Sufficient, Insufficient)

ECP in Mathematical Form: {ECP = {Sufficient, Insufficient}}

**BVA:**

**Boundary Values:**

Password with Minimum Complexity

Password with Maximum Complexity

BVA in Mathematical Form: {BVA = {Min Complexity, Max Complexity}}

**Risk Analysis:**

**Risks:**

Weak password policies.

Vulnerabilities due to weak passwords.

Test Case 4: Account Lockout (ECP & BVA)

**ECP:**

**Equivalence Classes:**

Login Attempts (Below Threshold, Above Threshold)

ECP in Mathematical Form: {ECP = {Below Threshold, Above Threshold}}

**BVA:**

**Boundary Values:**

Lockout Threshold (Minimum, Maximum)

BVA in Mathematical Form: {BVA = {Min Threshold, Max Threshold}}

**Risk Analysis:**

**Risks:**

Lockout threshold vulnerabilities.

Account lockout for legitimate users.

Test Case 5: Two-Factor Authentication (2FA) (ECP & BVA)

**ECP:**

**Equivalence Classes:**

2FA Enabled (Yes, No)

ECP in Mathematical Form: {ECP = {Yes, No}}

**BVA:**

Boundary Values: None in this case.

BVA in Mathematical Form: {BVA = {}}

**Risk Analysis:**

**Risks:**

Inadequate 2FA implementation.

Risks associated with single-factor authentication.

**Test Case 1 (API)**

ID: 1

Test Case: User Login (Valid Credentials)

**Precondition:** User is not logged in.

Test Steps:

Send an authentication request with valid credentials.

**Test Data:** Valid username and password.

**Expected Output:** User is logged in successfully.

**Post Condition:** User is redirected to the main dashboard.

Actual Output: User is logged in successfully.

Status: Pass

Comments: N/A

Test Case 2

ID: 2

**Test Case:** User Login (Invalid Password)

**Precondition:** User is not logged in.

**Test Steps:**

Send an authentication request with invalid password.

**Test Data:** Valid username, invalid password.

**Expected Output:** Login fails with an error message.

**Post Condition:** User remains on the login page.

**Actual Output:** Login fails with an error message.

Status: Pass

Comments: N/A

Test Case 3

ID: 3

Test Case: User Login (Invalid Username)

Precondition: User is not logged in.

Test Steps:

Send an authentication request with invalid username.

Test Data: Invalid username, valid password.

Expected Output: Login fails with an error message.

Post Condition: User remains on the login page.

Actual Output: Login fails with an error message.

Status: Pass

Comments: N/A

Test Case 4

ID: 4

Test Case: User Login (Empty Credentials)

Precondition: User is not logged in.

Test Steps:

Send an authentication request with empty credentials.

Test Data: Empty username and password.

Expected Output: Login fails with an error message.

Post Condition: User remains on the login page.

Actual Output: Login fails with an error message.

Status: Pass

Comments: N/A

**Test Case 5**

ID: 5

Test Case: User Logout

Precondition: User is logged in.

Test Steps:

Click on the "Logout" button or link.

Test Data: N/A

Expected Output: User is logged out and redirected to the login page.

Post Condition: User is on the login page.

Actual Output: User is logged out and on the login page.

Status: Pass

Comments: N/A

Test Case 6

ID: 6

**Test Case:** Role-Based Access (Admin)

Precondition: User is an Admin.

Test Steps:

Access sensitive administrative settings or data.

Test Data: Admin user credentials.

Expected Output: Access is granted to admin-only resources.

Post Condition: User can access admin settings or data.

Actual Output: Access is granted to admin-only resources.

Status: Pass

Comments: N/A

Test Case 7

ID: 7

Test Case: Password Complexity

Precondition: User is changing the password.

Test Steps:

Attempt to change the password.

Test Data: Password with insufficient complexity.

Expected Output: Password change fails with an error message.

Post Condition: User remains with the old password.

Actual Output: Password change fails with an error message.

Status: Pass

Comments: N/A

Test Case 8

ID: 8

Test Case: Account Lockout

Precondition: User attempts multiple logins.

Test Steps:

Enter invalid credentials multiple times.

Test Data: Invalid credentials multiple times.

Expected Output: Account is locked after reaching the threshold.

Post Condition: User is notified of the account lockout.

Actual Output: Account is locked after reaching the threshold.

Status: Pass

Comments: N/A

Test Case 9

ID: 9

Test Case: Two-Factor Authentication (2FA)

Precondition: User is enabling 2FA.

Test Steps:

Access the 2FA setup and enable it.

Test Data: Valid 2FA setup information.

Expected Output: 2FA is successfully enabled for the user.

Post Condition: User is prompted to use 2FA for logins.

Actual Output: 2FA is successfully enabled for the user.

Status: Pass

Comments: N/A

Test Case 10

ID: 10

Test Case: Password Reset Request

Precondition: User needs to reset the password.

Test Steps:

Request a password reset.

Test Data: User's email or username for the reset link.

Expected Output: A password reset link is sent to the user's email.

Post Condition: User receives a password reset link by email.

Actual Output: A password reset link is sent to the user's email.

Status: Pass

Comments: N/A

Test Case 11

ID: 11

Test Case: Access Control List (ACL)

Precondition: Admin configures ACL.

Test Steps:

Set up access control rules for specific resources.

Test Data: ACL configuration.

Expected Output: Access control is applied as configured.

Post Condition: Users have limited or extended access as per ACL.

Actual Output: Access control is applied as configured.

Status: Pass

Comments: N/A

Test Case 12

ID: 12

Test Case: Security Audit Logs

Precondition: Admin is configuring logs.

**Test Steps:**

Configure security audit logs for tracking events.

Test Data: Audit log settings.

Expected Output: Audit logs capture relevant security events.

Post Condition: Audit logs record security events as configured.

Actual Output: Audit logs capture relevant security events.

Status: Pass

Comments: N/A

GUI TEST CASES

**Authentication Test**

ID: 1

Test Case: Valid authentication token

Precondition: -

Test Steps: Call the Report Generation API endpoint

Test Data: Valid authentication token

Expected Output: Successful API response

Post Condition: API request is authenticated

Actual Output: Successful API response

Status: Pass

Comments: -

Invalid Token Handling

ID: 2

Test Case: Invalid authentication token

Precondition: -

Test Steps: Call the Report Generation API endpoint with an invalid token

Test Data: Invalid authentication token

Expected Output: Error response with appropriate error code

Post Condition: API request is rejected

Actual Output: Error response with appropriate error code

Status: Pass

Comments: -

Report Format Selection

ID: 3

Test Case: Requesting report generation

Precondition: -

Test Steps: Send a request to the API for a specific report format

Test Data: PDF, CSV, Excel

Expected Output: API responds with the chosen report format

Post Condition: User receives the report in the selected format

Actual Output: API responds with the chosen report format

Status: Pass

Comments: -

**Data Range Selection**

ID: 4

Test Case: Defining the data range for the report

Precondition: -

Test Steps: Specify the date range in the API request

Test Data: Start Date: 2023-01-01, End Date: 2023-12-31

Expected Output: API returns data within the specified date range

Post Condition: User receives the report data for the defined period

Actual Output: API returns data within the specified date range

Status: Pass

Comments: -

Data Filters

ID: 5

Test Case: Applying filters to the report

Precondition: -

Test Steps: Include specific parameters in the API request

Test Data: Filter by category: Sales, Region: North

Expected Output: API returns data filtered based on the provided parameters

Post Condition: User receives the filtered data in the report

Actual Output: API returns data filtered based on the provided parameters

Status: Pass

Comments: -

Error Handling

ID: 6

Test Case: Insufficient data for the report

Precondition: -

Test Steps: Request a report without sufficient data

Test Data: -

Expected Output: Error response with appropriate error code and message

Post Condition: API informs the user about the insufficient data

Actual Output: Error response with appropriate error code and message

Status: Pass

Comments: -

Multiple Reports Handling

ID: 7

Test Case: Concurrent report generation

Precondition: -

Test Steps: Initiate the generation of multiple reports simultaneously

Test Data: -

Expected Output: API processes each request independently without interference

Post Condition: User receives all requested reports without issues

Actual Output: API processes each request independently without interference

Status: Pass

Comments: -

**Report Customization**

ID: 8

Test Case: Customizing report parameters

Precondition: -

Test Steps: Include customization options in the API request

Test Data: -

Expected Output: API generates the report with the customized parameters

Post Condition: User receives the report with the desired customization

Actual Output: API generates the report with the customized parameters

Status: Pass

Comments: -

Report Sharing

ID: 9

Test Case: Sharing the generated report through API

Precondition: -

Test Steps: Implement the 'Share' functionality in the API

Test Data: Email: example@example.com

Expected Output: API sends the report to the specified email address

Post Condition: User receives the report at the provided email address

Actual Output: API sends the report to the specified email address

Status: Pass

Comments: -

Report Archival

ID: 10

Test Case: Storing generated reports for future reference

Precondition: -

Test Steps: Utilize the archival feature in the API

Test Data: -

Expected Output: API stores the generated report for future retrieval

Post Condition: User can access archived reports when needed

Actual Output: API stores the generated report for future retrieval

Status: Pass

Comments: -

API Response Format

ID: 11

Test Case: Response format validation

Precondition: -

Test Steps: Check the format of the response data

Test Data: -

Expected Output: API returns data in the expected format (JSON/XML)

Post Condition: User can process the API response easily

Actual Output: API returns data in the expected format

Status: Pass

Comments: -

**Rate Limiting Test**

ID: 12

Test Case: API rate limiting

Precondition: -

Test Steps: Send multiple requests within a short time frame

Test Data: Multiple requests

Expected Output: System enforces rate limits on the API

Post Condition: API enforces rate limits and prevents excessive requests

Actual Output: API enforces rate limits and prevents excessive requests

Status: Pass

TASK 3:(Gherkin language)

Feature: Metabase Web API Testing

**Scenario:** Authenticate API User (Valid)

Given the API user is not authenticated

When an authentication request with valid credentials is sent

Then the API should return an authentication token

And the API session is established

And an authentication token is received

And the status is "Pass"

And there are no comments

**Scenario:** Authenticate API User (Invalid)

Given the API user is not authenticated

When an authentication request with invalid credentials is sent

Then the API should return an authentication error

And the API session is not established

And an authentication error is received

And the status is "Pass"

And there are no comments

**Scenario:** Retrieve User Data

Given the API user is authenticated

When a request to retrieve user data is sent

Then the API should return user data matching the user

And user data is retrieved

And the retrieved data matches the authenticated user

And the status is "Pass"

And there are no comments

**Scenario:** Retrieve Dashboard List

Given the API user is authenticated

When a request to retrieve a list of dashboards is sent

Then the API should return a list of available dashboards

And a dashboard list is retrieved

And a list of dashboards is received

And the status is "Pass"

And there are no comments

**Scenario:** Retrieve Dashboard Data

Given the API user is authenticated

When a request to retrieve data for a specific dashboard is sent

And the dashboard ID and authenticated user's credentials are provided

Then the API should return data for the specified dashboard

And dashboard data is retrieved

And data for the specified dashboard is received

And the status is "Pass"

And there are no comments

**Scenario:** Retrieve Question List

Given the API user is authenticated

When a request to retrieve a list of questions is sent

Then the API should return a list of available questions

And a question list is retrieved

And a list of questions is received

And the status is "Pass"

And there are no comments

**Feature : GUI testing**

**Feature: Metabase User Authentication and Access Control**

**Scenario:** User Login (Valid Credentials)

Given the user is not logged in

When the user navigates to the Metabase login page

And provides valid username and password

Then the user is logged in successfully

And is redirected to the main dashboard

And the login status is "Pass"

And there are no comments

**Scenario:** User Login (Invalid Password)

Given the user is not logged in

When the user navigates to the Metabase login page

And provides a valid username and an invalid password

Then the login fails with an error message

And the user remains on the login page

And the login status is "Pass"

And there are no comments

**Scenario:** User Login (Invalid Username)

Given the user is not logged in

When the user navigates to the Metabase login page

And provides an invalid username and a valid password

Then the login fails with an error message

And the user remains on the login page

And the login status is "Pass"

And there are no comments

**Scenario:** User Login (Empty Credentials)

Given the user is not logged in

When the user navigates to the Metabase login page

And provides empty username and password

Then the login fails with an error message

And the user remains on the login page

And the login status is "Pass"

And there are no comments

**Scenario:** User Logout

Given the user is logged in

When the user clicks on the "Logout" button or link

Then the user is logged out and redirected to the login page

And the user is on the login page

And the logout status is "Pass"

And there are no comments

**Scenario:** Role-Based Access (Admin)

Given the user is an Admin

When the user accesses sensitive administrative settings or data

And provides admin user credentials

Then access is granted to admin-only resources

And the user can access admin settings or data

And the access status is "Pass"

And there are no comments