College of Computer Science and Engineering Department of Software Engineering

CCSW-323 Software Testing and Validation

Course Project

Student Name	ID
Reem Essam Hussain	2115694
Laila Najem Alzahrani	2110808
Maram Mohamed Almagadi	2110267

Section- EB8 Tool-

Postman API- OMDb

API

Provided for- Dr. Muna Altherwi

The Tool: Postman.

What is postman?

Postman platform includes a comprehensive set of tools that help accelerate the API lifecycle, from design, testing, documentation, and mocking to the sharing and discoverability of the APIs. We can build and run tests directly in Postman or as part of the CI/CD (Continuous integration/continuous deployment) pipeline through Newman (a Collection Runner that enables you to run and test a Postman Collection directly from the command line). Postman can be used to write functional tests, integration tests, regression tests, and more.

Postman's Nodes.js-based runtime contains support for common patterns and libraries that we can use to build tests quickly.

Some of the most common approaches to API testing are contract testing, unit testing, end-to-end testing, and load testing.

Tests confirm that your API is working as expected, that integrations between services are functioning reliably, and that any changes have not broken existing functionality. We can write test scripts for Postman API requests in JavaScript. We can also use test code to aid the debugging process when something goes wrong with our API project. For example, we might write a test to validate our API's error handling by sending a request with incomplete data or wrong parameters.

API.

Application Programming Interface.

Set of rules and protocols that allows one software application to interact with another. APIs define the methods and data formats that applications can use to communicate with each other. They provide a way for

different software systems to exchange information and functionality, allowing them to work together seamlessly. They are used in a variety of contexts, including web development, mobile app development, and integration between different software systems.

We decided to test the **OMDb API** using Postman testing tool

We used letters to represent each characteristic to simplify writing the coverage criteria.

Characteristics	Block #1	Block #2	Block #3
A: Plot	requesting a full or short plot	requesting a keyword or a description as the plot	no plot requested
B: media IMDb ID	an ID with the proper format of having 2 Ts in the beginning and 8 numbers following it. the ID should match an existing one in the IMDb database. ttxxxxxxxx	an ID with an improper format, having multiple different letters in the beginning and more or less than 8 numbers following it. the ID doesn't match an existing one in the IMDb database. xxxxxxxxxx	no ID entered
C:Page	requesting a page number from 1-100 that the media searched for is found on. and a search parameter is needed for it. page=x00	requesting a page number out of bound or/and the media isn't found on. page=xxxx	no page number requested
D:title	a title that exists, either generally or on the database itself, with no spelling errors t=string	a title that doesn't exist, isn't on the database, or has spelling errors t=inappropriate string	no ID entered
E:Released year	a possible current year when media started being released, past year, or future year that the media is expected to be released on.	a year when media wasn't invented or released yet or a year too far in the future y=<1888 or >2024	no year entered

	y=1888-2024		

The criteria coverage we chose is base choice criteria.

$1 + \Sigma i=1 (Bi -1).$

1 + (3 - 1) + (3 - 1) + (3 - 1) + (3 - 1) + (3 -1) = 11

base:A1,B1,C1,D1,E1

1+2+2+2+2+2=11 tests.

Base	**A1,B1,C1,D1,E1**
A	**A2**,B1,C1,D1,E1
A	**A3**,B1,C1,D1,E1
B	A1,**B2**,C1,D1,E1
B	A1,**B3**,C1,D1,E1
C	A1,B1,**C2**,D1,E1
C	A1,B1,**C3**,D1,E1
D	A1,B1,C1,**D2**,E1
D	A1,B1,C1,**D3**,E1
E	A1,B1,C1,D1,**E2**

E	A1,B1,C1,D1,**E3**

ACTUAL VALUES:

note that a space means an empty value, in reality we don't add the parameter in the API url for an empty parameter.

Base:plot=full, i=tt0371746,page=1,t=iron man,y=2008
plot=fighting ninjas, i=tt0371746,page=1,t=iron man,y=2008
plot= ,i=tt0371746,page=1,t=iron man,y=2008
plot=full, i=bbwe54,page=1,t=iron man,y=2008
plot=full, i= ,page=1 ,t=iron man,y=2008
plot=full, i=tt0371746,page=378,t=iron man,y=2008
plot=full, i=tt0371746,page= ,t=iron man,y=2008
plot=full, i=tt0371746,page=1,t=key of nature,y=2008
plot=full, i=tt0371746,page=1,t= ,y=2008
plot=full, i=tt0371746,page=1,t=iron man,y=1678
plot=full, i=tt0371746,page=1,t=iron man,y=

Test cases:

Test case ID	Test case description	Test steps	Test data	Expected results	Actual results	Pass/Fail
TC01	Verify that the response body contains the media full plot.	1. Open Postman. 2. Create a new request. 3. Set the request type to GET. 4. Set the URL to: http://www.omdbapi.com and string the test data to it 5. Send the request. 6. Check the response body for the expected result.	/?apikey=53da334d&t=iron_man&plot=full&r=json	Tony Stark. Genius, billionaire, playboy, philanthropist. Son of legendary inventor and weapons contractor Howard Stark. When Tony Stark is assigned to give a weapons presentation to an Iraqi unit led by Lt. Col. James Rhodes, he's given a ride on enemy lines. That ride ends badly when Stark's Humvee that he's riding in is attacked by enemy combatants. He survives - barely - with a chest full of shrapnel and a car battery attached to his heart. In order to survive he comes up	Tony Stark. Genius, billionaire, playboy, philanthropist. Son of legendary inventor and weapons contractor Howard Stark. When Tony Stark is assigned to give a weapons presentation to an Iraqi unit led by Lt. Col. James Rhodes, he's given a ride on enemy lines. That ride ends badly when Stark's Humvee that he's riding in is attacked by enemy combatants. He survives - barely - with a chest full of	Pass

				with a way to miniaturize the battery and figures out that the battery can power something else. Thus Iron Man is born. He uses the primitive device to escape from the cave in Iraq. Once back home, he then begins work on perfecting the Iron Man suit. But the man who was put in charge of Stark Industries has plans of his own to take over Tony's technology for other matters.	shrapnel and a car battery attached to his heart. In order to survive he comes up with a way to miniaturize the battery and figures out that the battery can power something else. Thus Iron Man is born. He uses the primitive device to escape from the cave in Iraq. Once back home, he then begins work on perfecting the Iron Man suit. But the man who was put in charge of Stark Industries has plans of his own to take over Tony's technology .for other matters	

Test case ID	Test case description	Test steps	Test data	Expected results	Actual results	Pass/Fail
TC02	Verify that the response body contains the media full plot.	1. Open Postman. 2. Create a new request. 3. Set the request type to GET. 4. Set the URL to: http://www.omdbapi.com and string the test data to it 5. Send the request. 6. Check the response body for the expected result.	/?apikey=53da334d&t=iron_man&plot=tony stark. Genius, billonaire	Tony Stark. Genius, billionaire, playboy, philanthropist. Son of legendary inventor and weapons contractor Howard Stark. When Tony Stark is assigned to give a weapons presentation to an Iraqi unit led by Lt. Col. James Rhodes, he's given a ride on enemy lines. That ride ends badly when Stark's Humvee that he's riding in is attacked by enemy combatants. He survives - barely - with a chest full of shrapnel and a car battery attached to his heart. In order to survive he comes up with a way to miniaturize the battery and figures out that	Default short plot due to not specifying if the plot is full or short. "After being held captive in an Afghan cave, billionaire engineer Tony Stark creates a unique weaponized suit of armor to fight evil."	Fail

				the battery can power something else. Thus Iron Man is born. He uses the primitive device to escape from the cave in Iraq. Once back home, he then begins work on perfecting the Iron Man suit. But the man who was put in charge of Stark Industries has plans of his own to take over Tony's technology for other matters.		

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC03	Verify that the response body contains the correct information for a specific episode when fetching details with the correct IMDb ID.	1. Open Postman. Open Postman. 2. Create a new request. 3. Set the request type to GET. 4. Set the URL to: http://www.omdbapi.com and string the test data to it 5. Send the request. Check the response body for the .expected result	/?apikey=53da334d&i=tt16992176&r=jason	Title": "Episode " ,"#4.1 ,"Year": "2024" ,"Rated": "N/A" Released": " ,""N/A ,"Season": "4" ,"Episode": "1" Runtime": " ,""N/A Genre": " "Comedy, Drama, ,"Romance	Title": "Episode " ,"#4.1 ,"Year": "2024" ,"Rated": "N/A" Released": " ,""N/A ,"Season": "4" ,"Episode": "1" Runtime": " ,""N/A Genre": " "Comedy, Drama, ,"Romance	Pass

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC04	Verify that the response body doesn't contain information for a specific episode when fetching details with an incorrect IMDb ID.	1. Open Postman. Open Postman. 2. Create a new request. 3. Set the request type to GET. 4. Set the URL to: http://www.omdbapi.com and string the test data to it 5. Send the request. Check the response body for the .expected result	/?apikey=53da334d&i=bb3652&r=json	No data returned	No data returned error message shows	Pass

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail

TC05	Verify that the response body doesn't contain information for a specific episode when fetching details with an empty IMDb ID.	1. Open Postman. Open Postman. 2. Create a new request. 3. Set the request type to GET. 4. Set the URL to: http://www.omdbapi.com and string the test data to it 5. Send the request. Check the response body for the .expected result	/?apikey=53da334d&i= &r=jason	No data returned	No data returned error message shows	Pass

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC06	Verify that the response contains the correct page and correct keyword search of search results when	1. Open Postman. Open Postman. 2. Create a new request. 3. Set the request type to GET. 4. Set the URL to: http://www.omdbapi.com and string the test data to it 5. Send the request. Check the response body for the expected result	apikey=53da334d&s=action&page=2?/	Movies with search keyword Action, and page number 2	Movies with search keyword Action, and page number 2	Pass

	searching for movies with pagination					

Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC07	Verify that the response contains the correct page and incorrect keyword search of search results when searching for movies with pagination	Open Postman. Open .Postman .Create a new request Set the request type to .GET Set the URL to: http://www.omdbapi.com and string the test data to it .Send the request Check the response body for the expected result	apikey=53da334d&s=acyon&page=2?/	media not found and an error message	media not found and an error message	Pass

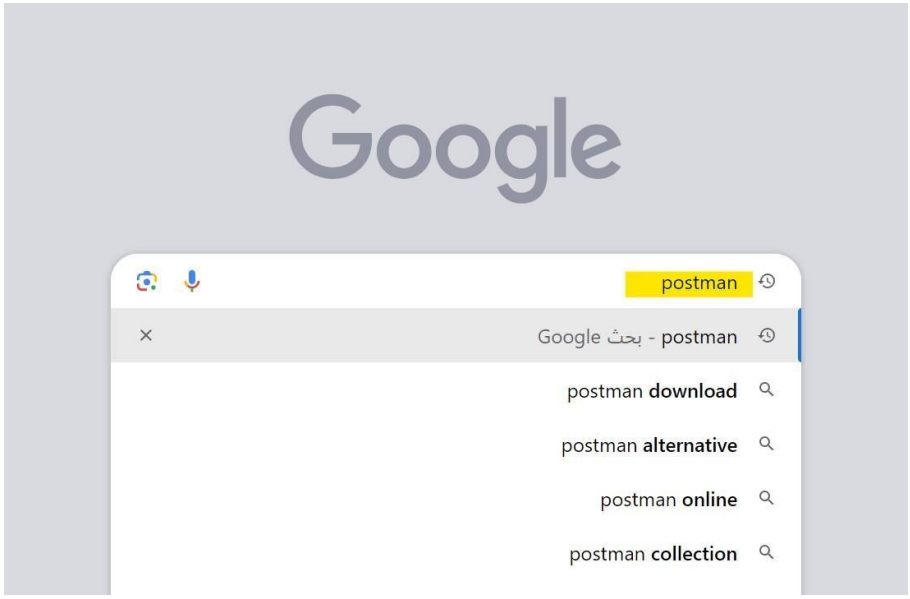
Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC08	Verify that the response contains an incorrect page and correct keyword search of search results when searching for movies with pagination	Open Postman. Open .Postman .Create a new request .Set the request type to GET Set the URL to: http://www.omdbapi.com and string the test data to it .Send the request Check the response body for the expected result	apikey=53da334d&s=action&page=247?/	error	error	Pass

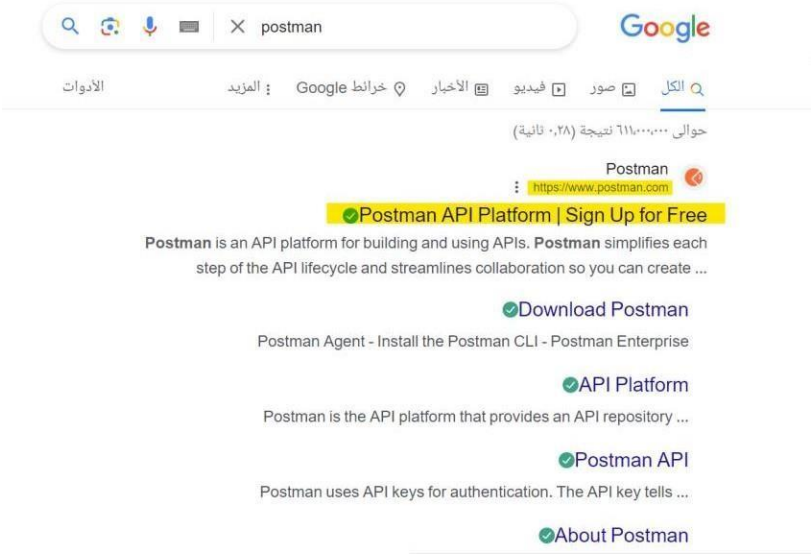
Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC09	Verify that the response contains an error when attempting to get media details with an invalid release year.	1- Open Postman. 2- Create a new request. 3- Set the request type to GET. 4- Set the URL to:http://www.omdbapi.com/?apikey=53da334d&t=jurassic_park&y=abc&plot=full&r=json 5-send the request 6-check the response body for an error message	Movie title: Jurassic Park. Invalid release year: abc.	Error message indicating that the release year is invalid	The whole movie details	Fail

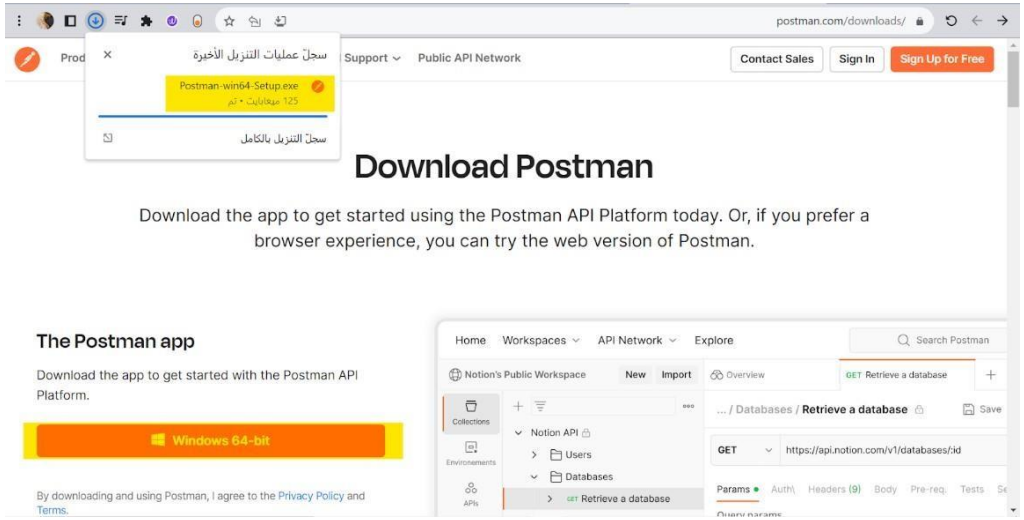
Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC10	Verify that the response contains an error when attempting to get movie details with an valid release year and invalid movie title	1- Open Postman. 2- Create a new request. 3- Set the request type to GET. 4- Set the URL to:http://www.omdbapi.com/?apikey=53da334d&t=jic_park&y=1993&plot=full&r=json 5-send the request 6-check the response body for an error message	Movie title: Jic Park. Invalid release year: 1993.	movie not found	movie not found	Pass

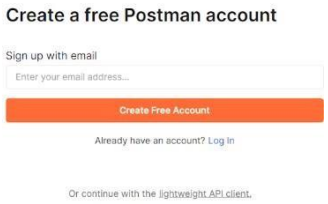
Test Case ID	Test Case Description	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
TC11	Verify that the response contains movie details with valid release year and valid movie title	1- Open Postman. 2- Create a new request. 3- Set the request type to GET. 4- Set the URL to:http://www.omdbapi.com/?apikey=53da334d&t=jurassic_park&y=1993&plot=full&r=json 5-send the request 6-check the response body for movie details	Movie title: Jurassic Park. Invalid release year: 1993.	movie details	movie details	Pass

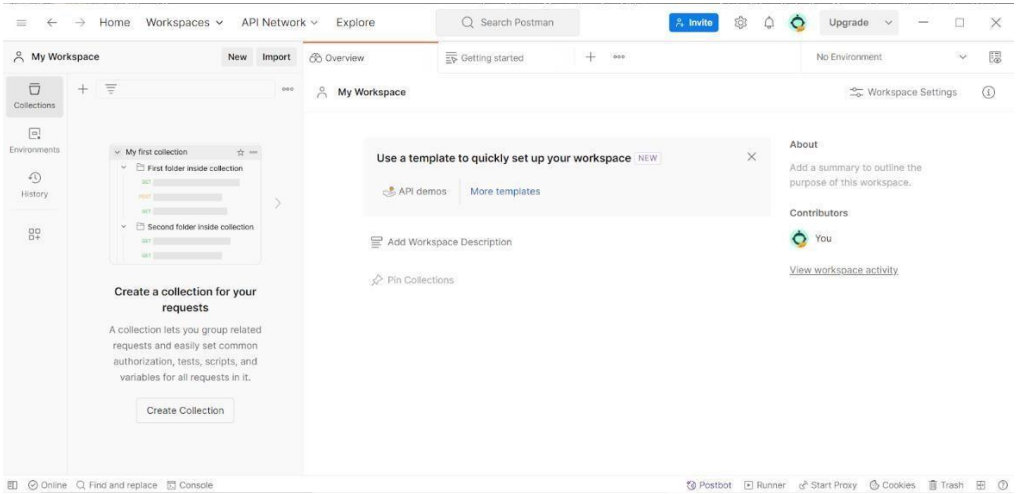
Postman installation:





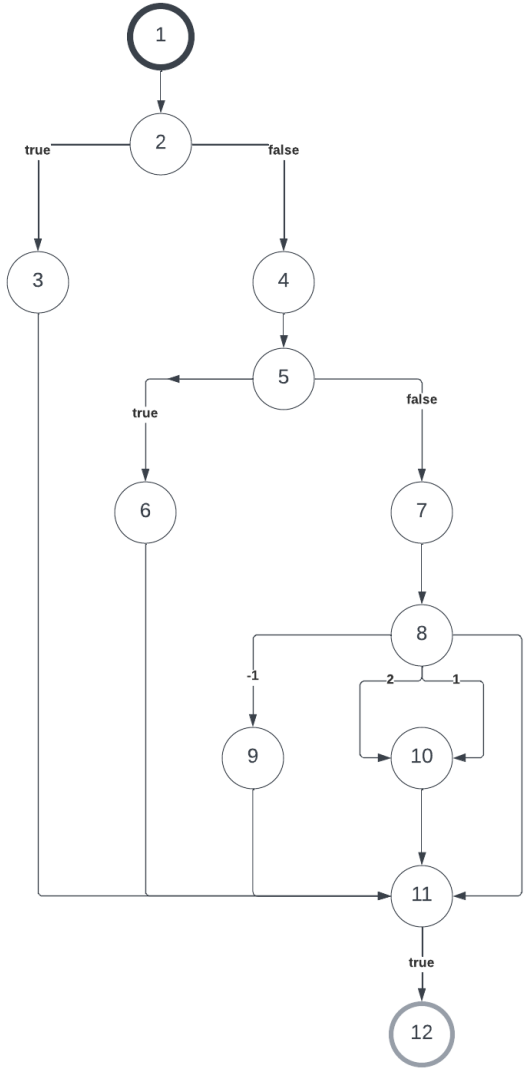





Home
Workspaces
API Network
Explore
Search Postman
Invite
Upgrade
My Workspace
New
Import
Overview
Getting started
No Environment
Collections
Environments
History
My first collection
First folder inside collection
Second folder inside collection
Create a collection for your requests
A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.
Create Collection
My Workspace
Workspace Settings
Use a template to quickly set up your workspace
NEW
API demos
More templates
Add Workspace Description
Pin Collections
About
Add a summary to outline the purpose of this workspace.
Contributors
You
View workspace activity
Online
Find and replace
Console
Postbot
Runner
Start Proxy
Cookies
Trash

Additional requirement, open source code graph coverage:

Graph:



Source code:

```
private void validatePassword(){
    if (RegistrationHelper.isNullOrEmptyString(user.getPassword())) {
        messages.add("Password is required");
    } else {
        final int passwordLen = user.getPassword().length();
        if (passwordLen > Constants.MAX_PASSWORD_LENGTH || passwordLen < Constants.MIN_PASSWORD_LENGTH) {
            messages.add("Length of password should be between " + Constants.MIN_PASSWORD_LENGTH + " and "
                + Constants.MAX_PASSWORD_LENGTH);
        } else {
            // length OK, check password strength.
            int strength = RegistrationHelper.calculatePasswordStrength(user.getPassword());
            switch (strength) {
            case -1:
                messages.add("Password cannot end with a number");
                break;
            case 1:
            case 2:
                messages.add("Password is too weak");
                break;
            default:
                break;
            }
        }
    }
```

```
    if (null != user.getPassword() && !user.getPassword().equals(user.getConfirmPassword())) {
        messages.add("Password and confirm password should be exactly same");
    }
}
```

Github link:

https://github.com/topcoder-platform/tc-website/blob/15ab92adf0e60afb1777b3d548b5ba3c3f6c12f7/src/main/com/topcoder/reg/actions/RegisterAction.java#L524

Node coverage:

Initial node: 1

Final node: 12

Node Coverage

(TR) Test Requirement:	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}
Test Path:	[1,2,3,11,12] [1,2,4,5,6,11,12] [1,2,4,5,7,8,9,11,12] [1,2,4,5,7,8,10,11,12]

Edge Coverage

(TR) Test Requirement:	{(1, 2), (2, 3), (2, 4), (4, 5), (5, 6), (5, 7), (7, 8), (8, 9), (8, 10), (8, 11), (9, 11), (10, 11), (11, 12), (3, 11), (6, 11)}
Test Path:	[1,2,3,11,12] [1,2,4,5,6,11,12] [1,2,4,5,7,8,11,12] [1,2,4,5,7,8,9,11,12] [1,2,4,5,7,8,10,11,12]

Edge-Pair Coverage

(TR) Test Requirement:	{[1, 2, 3], [1, 2, 4], [2, 4, 5], [4, 5, 6], [4, 5, 7], [5, 7, 8], [7, 8, 9], [7, 8, 10], [7, 8, 11], [8, 9, 11], [8, 11, 12], [8, 10, 11], [9, 11, 12], [10, 11, 12], [2, 3, 11], [3, 11, 12], [6, 11, 12]}
Test Path:	[1, 2, 3, 11, 12] [1, 2, 4, 5, 6, 11, 12] [1, 2, 4, 5, 7, 8, 9, 11, 12] [1, 2, 4, 5, 7, 8, 10, 11, 12] [1, 2, 4, 5, 7, 8, 11, 12]

Test Scenarios: [1,2,3,11,12]. [1,2,4,5,6,11,12]

2 test paths are needed for **Node coverage**

1. Test Path: [1,2,3,11,12]

   [start

   enter an empty or null password.

   true, message appears “password is required”.

   check two conditions “Matching passwords”, confirm password.

   unsuccessful, passwords must be the same].

   Test Case:

   Test Path: [1, 2, 3, 11, 12]

   ID: TC01

   Description: Validate empty or null password entry and confirm password matching.

   Input Example:

   username: reemhussain

   password: [Empty]

   confirm password: [Empty]

   Steps:

   1. Start. Navigate to the registration or password update page.
   2. Enter username, and leave the “Password” field empty.

3. Message appears “Password is required”.
4. check the passwords matching.
5. Message appears “Passwords must be the same”.

Expected Result: The system should display an error message: “Password is required!”.

2 test paths are needed for **Node coverage**

2. Test Path: [1,2,4,5,6,11,12]

[start

enter an empty or null password.

false, message appears “password is correct”.

check password length.

true, error length message.

check two conditions “Matching passwords”, confirm password.

unsuccessful, passwords must be the same].

Test Case:

Test Path: [1,2,4,5,6,11,12]

ID: TC02

Description: Validate **non** empty or null password entry and confirm password length and matching.

Input Example:

username: reemhussain

password: reem2023

confirm password: reem

Steps:

1. Start.
2. Enter username and password.
3. Message appears “Password is correct”.
4. Check password length.
5. Message appears “Error length message”.
6. Check the passwords matching.
7. Message appears “Passwords must be the same”.

Expected Result: The system should display an error message: “Password must be at least 8 characters”.

Test Scenarios: [1,2,3,11,12], [1,2,4,5,7,8,11,12]

2 test paths are needed for **Edge coverage**

1. **Test Path:** [1,2,3,11,12]

[start

enter an empty or null password.

true, message appears “password is required”.

check two conditions “Matching passwords”, confirm password.

unsuccessful, passwords must be the same].

Test Case:

Test Path: [1, 2, 3, 11, 12]

ID: TC01

Description: Validate empty or null password entry and confirm password matching.

Input Example:

username: reemhussain

password: [Empty]

confirm password: [Empty]

Steps:

1. Start. Navigate to the registration or password update page.
2. Enter username, and leave the “Password” field empty.
3. Message appears “Password is required”.
4. check the passwords matching.
5. Message appears “Passwords must be the same”.

Expected Result: The system should display an error message: “Password is required!”.

2 test paths are needed for **Edge coverage**

2. Test Path: [1,2,4,5,7,8,11,12]

[start.

enter an empty or null password.

false, message appears “password is correct”.

check password length.

false, length ok.

check password strength.

check two conditions “Matching passwords”, confirm password.

Message appears “Passwords must be the same”.]

Test Case:

Test Path: [1,2,4,5,7,8,11,12].

ID: TC02.

Description: Validate **non** empty or null password entry, password length, strength, and matching.

Input Example:

username: reemhussain

password: reem2023

confirm password: reem

Steps:

1. Start.
2. Enter username and password.
3. Message appears “Password is correct”.
4. Check password length.
5. Length Ok.
6. Check strength.
7. Check the passwords matching.
8. Message appears “Passwords must be the same”.

Expected Result: The system should display an error message: “Passwords must be the same”.

Test Scenarios: [1, 2, 3, 11, 12], [1, 2, 4, 5, 7, 8, 10, 11, 12].

2 test paths are needed for **Edge Pair Coverage.**

1. **Test Path:** [1, 2, 3, 11, 12]

   [start.

   enter an empty or null password.

   true, message appears “password is required”.

   check two conditions “Matching passwords”, confirm password.

   unsuccessful, passwords must be the same].

   Test Case:

   Test Path: [1, 2, 3, 11, 12].

   ID: TC01

Description: Validate empty or null password entry and confirm password matching.

Input Example:

username: reemhussain

password:[Empty]

confirm password: [Empty]

Steps:

1. Start. Navigate to the registration or password update page.
2. Enter username, and leave the “Password” field empty.
3. Message appears “Password is required”.
4. check the passwords matching.
5. Message appears “Passwords must be the same”.

Expected Result: The system should display an error message: “Passwords must be the same”.

2 test paths are needed for **Edge Pair Coverage.**

1. **Test Path: [1, 2, 4, 5, 7, 8, 10, 11, 12].**

   [start.

   enter an empty or null password.

   false, message appears “password is correct”.

   check password length.

false, length ok.

check password strength.

the password is too weak.

check two conditions “Matching passwords”, confirm password.

Message appears “Passwords must be the same”.]

Test Cases:

Test Path: [1, 2, 4, 5, 7, 8, 10, 11, 12].

ID: TC02.

Description: Validate non empty or null password entry, password length, strength, strong, and matching.

Input Example:

username: reemhussain

password: 123

password: 123

Steps:

1. Start.
2. Enter username and password.
3. Message appears “Password is correct”.
4. Check password length. 5
5. Length Ok.
6. Check strength.
7. Check the strongest
8. Check the passwords matching.

9. Message appears “Passwords must be the same”.

Expected Result: The system should display an error message: “Passwords must be the same and password is weak”.