

Testing Document

Project Title: Rettiwt

Document Version Number: 1.0

Printing Date: 22/4/2023

Group ID: E3

Group Members:

TSE Hui Tung, 1155158864

CHEUNG Kwong Tai, 1155142517

NG Man Tik, 1155158302

LAI Chuen Fung, 1155144433

Department: Department of Computer Science and Engineering

University: The Chinese University of Hong Kong

Table of Content

1. TEST PLAN	
1.1. Objective and Test Approach	p.3
1.2. Testing Environment	p.3
1.3. Testing Tools	p.3
2. TEST CASES	
2.1. Frontend	
2.1.1. Like and Dislike	
2.1.1.1. Test set 1-5	p.
2.1.2. Login and Sign-up System	
2.1.2.1. Test set 1-9	p.
2.1.3. Comment, edit bio, post and retweet's description	
2.1.3.1. Test set 1-5	p.
2.1.4. Post image and user avatar	
2.1.4.1. Test set 1-4	p.
2.1.5. Profile editing	
2.1.5.1. Test set 1-4	p.
2.1.6. Deleting post and retweeted post	
2.1.6.1. Test set 1-3	p.
2.1.7. Admin	
2.1.7.1. Test set 1-3	p.
2.2. Backend API	
2.2.1. Account Route	
2.2.1.1. Test set 1-6	p.
2.2.2. Follower Route	
2.2.2.1. Test set 1-8	p.
2.2.3. Tweet Route	
2.2.3.1. Test set 1-3	p.
2.2.4. Comment Route	
2.2.4.1. Test set 1-2	p.
2.2.5. Like Route	
2.2.5.1. Test set 1-4	p.
2.2.6. Dislike Route	
2.2.6.1. Test set 1-4	p.
2.3. Security	
2.3.1. Test set 1-4	p.

1. TEST PLAN

1.1. Objective and Test Approach

The objective of this test plan is to perform black box testing on the front-end application and backend APIs to ensure their correct and reliable behavior. The scope covers five main categories: Login and Sign-up system, User Profile, Posting and Interaction, Navigation and Display, and Admin Functions. Our test approach involves preparing test cases with a combination of valid and invalid inputs, defining expected outputs, executing test cases, and comparing actual outputs with expected outputs.

1.2. Testing Environment

- Hardware: Multiple devices with various screen resolutions
- Software: Multiple web browsers (e.g., Chrome, Firefox, Safari) and operating systems (e.g.: MacOS, Window)

1.3. Testing Tools

- Frontend Application: Web browsers for UI testing
- Backend APIs: Postman for API testing

2. TEST CASES

2.1 Frontend

2.1.1 Like/Dislike

Test Case 1

Purpose	To test whether the like and dislike module works with the increased like/dislike counter after doing so.
Inputs	Like/Dislike the post which hasn't been liked and disliked before.
Expected Outputs	The post's like/dislike button will be turned into the clicked one, and the corresponding like/dislike count will increase by 1.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 2

Purpose	To test if the user can change their selection from like/dislike.
Inputs	Like/Dislike the post where the user has previously disliked/liked it.
Expected Outputs	If the user is trying to like the post, the dislike button will be unclicked, and the dislike counter will be decreased by 1. Meanwhile, the like button will be clicked with the counter increased by 1, and vice versa.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 3

Purpose	To test if the website can block users from spam click the like/dislike button.
---------	---

Inputs	Click the like button on the new post and do so on the same post within 2 seconds. Similarly, to the dislike post.
Expected Outputs	A warning about spam clicking will be shown, while the button and the counter will remain unchanged.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 4

Purpose	To see if the like/dislike status can be fetched correctly.
Inputs	Like the post posted by the user, then refresh the page and check that post in 1. Home page 2. Profile page. Similarly, to the dislike post.
Expected Outputs	The clicked like/dislike button with the increased counter will be fetched on the home page and profile page.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 5

Purpose	To test if local storage is cleared when different users are used on the same PC.
Inputs	Click the like/dislike button. Then log out and log in with a different account who doesn't like and disliked that post.
Expected Outputs	The like and dislike buttons are both unclicked, but the like/dislike counter is increased by 1.
Pass/Fail Criteria	Pass this test case if and only if the second user is following the first user and the outcome is exactly the same as the expected output.

2.1.2 Login and Sign-up System

Test Case 1

Purpose	To check if a logged in user can log out successfully.
Inputs	Click the logout button in the Home, Profile or Admin pages.
Expected Outputs	It will redirect to the masked home page, with only the home button, and it will show the log in/ sign up modal.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 2

Purpose	To verify that the user is unable to sign up using an email that has been previously logged in using Google login
Inputs	An email that has been previously used to log in by Google login.
Expected Outputs	The sign-up box will display "Email registered using Google sign in. Please continue with Google." instead of creating a new account.
Pass/Fail Criteria	Pass this test case if and only if no new account is created for the Google-logged email.

Test Case 3

Purpose	To check if the "forgot password" function works.
Inputs	1. In login page, click the “forgot password” button and input the email which is previously registered. 2. Open the verification email (if any) and click the embedded link to redirect back to Rettiwt 3. Input the new password which is valid and not duplicated with the old one
Expected Outputs	The sign-up box will display “A password reset email has been sent to your email address.” A verification email is sent to the input email address and the link can redirect back to the forgot password page. Able to login with new password.
Pass/Fail Criteria	Pass this test case if and only if all outputs are as expected.

Test Case 4

Purpose	To test the restriction in sign up’s username field, not longer than 8characters.
Inputs	Input the following to corresponding field: Email: tsehuitung@gmail.com (valid email) Username: tesing12345 Password: 123 Confirm Password: 123 ReCAPTCHA: true
Expected Outputs	Output error message: “username should not longer than 8 characters.”
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 5

Purpose	To test the restriction in sign up’s email field, with not valid email.
Inputs	Input the following to corresponding field in the sign-up page: Email: tsehuitung.gmail.com (invalid email) Username: tesing12345 Password: 123 Confirm Password: 123 ReCAPTCHA: true
Expected Outputs	Output error message: “Invalid email format.”
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 6

Purpose	To check if the confirm password does not match with previous input in sign-up component.
Inputs	A password input in the confirm password which does not match with the previous password input.
Expected Outputs	An alert message “the password does not match the previous password” and registration request is not accepted
Pass/Fail Criteria	Pass this test case if and only if the no account will be register with a mismatched confirm password.

Test Case 7

Purpose	Check if the system can reject the signup without ReCAPTCHA done
Inputs	In the sign-up page, input: Email: 123@link.cuhk.edu.hk Username: user123 Password: aA112233 Confirmed password: aA112233 ReCAPTCHA: false
Expected Outputs	Output error message: "ReCAPTCHA hasn't been done yet."
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 8

Purpose	Check if the system can reject the login if the password for that account is wrong
Inputs	Assume that user is registered, in the login page, input: Email: 123@link.cuhk.edu.hk Password: a123 (where the correct password is aA112233)
Expected Outputs	Output error message: "Invalid password"
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 9

Purpose	Check if a previously signed up user can successfully login
Inputs	Login with signed up user: Email: tsehuitung@gmail.com Password: 12345678
Expected Outputs	Able to login and show all the user information after logging in, including avatar, username and userId.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output. Fail if the user is unable to login.

2.1.3 Comment, edit bio, post and retweet's description

Test Case 1

Purpose	To test if it can trim the very long comment.
Inputs	In text area of comment section, input: Hello ... (50 new line) World
Expected Outputs	It will show "Hello World" in the same line.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 2

Purpose	Check the div's maximum word count in 200 words using typing method
Inputs	In the text area of the comment section, input more than 200 words.

Expected Outputs	The input section should be able to stop at 200 words and no longer be able to keep typing but deleting words is allowed
Pass/Fail Criteria	Fail if the word's count is not counting properly or "" counting one word

Test Case 3

Purpose	Check the div's maximum word count in 200 words using pasting method
Inputs	In text area of comment section, repeatedly pasting 300 words
Expected Outputs	The input section should be able to stop at 200 words and only the first 200 words be retrieved into the input section
Pass/Fail Criteria	Fail if the words pasting can exceed 200 words or the words stopped at 200 words but the div still extended base on exceeded lines.

Test Case 4

Purpose	Check the edge cases of input section, such as inputting emoji.
Inputs	In text area of comment section, input '@', '#', '\$' and emoji
Expected Outputs	The div should be able to show the tag and emoji properly with any error
Pass/Fail Criteria	Fail if the div can't fetch the content properly or showing the symbol and emoji in encoded text.

Test Case 5

Purpose	Test if the comment module is functioning properly.
Inputs	In one of the posts, click the comment button and input the following: "Hello World!"
Expected Outputs	The "Comment Submitted" notification will be shown, the new comment will be shown after the page is refreshed.
Pass/Fail Criteria	Fail if the div can't fetch the content properly or showing the symbol and emoji in encoded text.

2.1.4 Post image & avatar

Test Case 1

Purpose	To test if user can upload a wrong file type as the image
Inputs	In the tweet page, upload a .exe file. With the non-null content
Expected Outputs	Warning message of invalid file type will be shown. And the posting will be rejected.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 2

Purpose	To test if user upload an image with very large size
Inputs	In the tweet page, upload a .png file with 4k resolution. With the non-null content.
Expected Outputs	"Post successfully" is shown and the post is with the resized image.

Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output. And fails if error message is shown OR the image with original size is uploaded.
--------------------	---

Test Case 3

Purpose	To test if the damaged image file can be uploaded.
Inputs	In the tweet page, upload a .png file which is converted by the .zip file. With the non-null content
Expected Outputs	A warning message will be shown. And the posting will be rejected.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 4

Purpose	To test if the basic functionality of post system
Inputs	In the tweet page, upload a .png file with 8MB file size and "" in description part
Expected Outputs	Return "Posted Successfully" response and successfully upload the post in home and profile pages. The image size reduced to 1280 * 1280 pixels with 1:1 aspect ratio.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output. Fail if the image size didn't resize.

2.1.5 Profile editing

Test Case 1

Purpose	To test if user can set null username
Inputs	In the profile page, click the edit profile button and set the following: Username: ""
Expected Outputs	Warning messages will be shown, and the editing will be rejected.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 2

Purpose	Check if the username is set to 8 characters at maximum
Inputs	In the profile page, click the edit profile button and input the following: Username: "\$#dasDS234@#\$"
Expected Outputs	The input section will stop at the 8 characters.
Pass/Fail Criteria	Fail if the username can exceed 8 characters.

Test Case 3

Purpose	Check if the privacy setting is functioning properly
Inputs	In the profile page, click the edit profile button and toggle the privacy button to “Private”
Expected Outputs	The pending request list show be showing in below the follower list and for those who are not following this account, they can’t view the posts.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 4

Purpose	Check if the background opacity setting is functioning properly
Inputs	In the profile page, click the edit profile button and change the background opacity to 1.0
Expected Outputs	The background turns to dark theme
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

2.1.6 Deleting post and retweeted post

Test Case 1

Purpose	Check if a post can be successfully deleted in profile page.
Inputs	In the profile page, click the delete button on the top right corner of the post.
Expected Outputs	Showing the delete post successfully message, then refresh the page after 3 seconds and the post will remove from the profile page, also with all the retweeted posts and comments deleted too.
Pass/Fail Criteria	Fail if the retweeted posts didn’t remove or the post is not removing properly in the profile page, or the success/error message didn’t pop up.

Test Case 2

Purpose	Check if a retweeted post can be successfully deleted in profile page.
Inputs	In the profile page, click the delete button on the top right corner of the retweeted post.
Expected Outputs	Showing the deleted retweeted post successfully message, then refresh the page after 3 seconds and the retweeted post will remove from the profile page. The post being retweeted still exists and is visible.
Pass/Fail Criteria	Fail if the retweeted posts didn’t remove properly in the profile page or the success/error message didn’t pop up.

Test Case 3

Purpose	To test if the user can keep deleting post before the server done the previous deletion
Inputs	In the profile page, click the delete button on the top right corner of the post and before the server response “Deleted Post Successfully”, click the delete button again in another post/retweeted post.
Expected Outputs	Restricted the user to delete another post/retweeted post and show an error message: “Deleting post is processing...”
Pass/Fail Criteria	Fail if another post/retweeted post can be deleted too during deleting previous post.

2.1.7 Admin

Test Case 1

Purpose	To test if admin can deactivate the user who is currently log in
Inputs	In admin side, click “deactivate” button next to the user record to deactivate the user who is currently logging in.
Expected Outputs	In that user side, log in/ sign up page will be redirected whenever the page is refreshed, and he cannot log in again using valid username and password.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output.

Test Case 2

Purpose	To test if a user account that has been removed by an admin is no longer accessible.
Inputs	In admin page, click “remove” button next to a user record to remove that user account.
Expected Outputs	The removed user cannot login and other users cannot search for the removed user.
Pass/Fail Criteria	Pass this test case if and only if the removed account is no longer accessible.

Test Case 3

Purpose	To test if a post that has been removed by an admin is no longer visible.
Inputs	In admin page, click “remove” button next to a post record to remove that post
Expected Outputs	The removed post is invisible in the home page and the poster’s profile page
Pass/Fail Criteria	Pass this test case if and only if the removed post is no longer visible.

2.2 Backend API

2.2.1 Account Route

Test Case 1

Purpose	Check if admin can get all the user information using the API
Inputs (Request)	GET http://localhost:5500/account/
Expected Outputs	Status 200 OK, a list of accounts is returned. If there is no user information in the database, return an empty list.
Pass/Fail Criteria	This case fails if any account information in response body is missing, or the returned status is incorrect.

Test Case 2

Purpose	Check if user can get search result using the API
Inputs (Request)	GET http://localhost:5500/account/search/fafdb1/Kw
Expected Outputs	Status 200 OK, a list of matched accounts is returned. If there is no matched user information in database, return an empty list.
Pass/Fail Criteria	This case fails if any account information in response body is missing, or the returned status is incorrect.

Test Case 3

Purpose	Check if user can get his/her or another's avatar using the API
Inputs (Request)	GET http://localhost:5500/account/profile/464b67/
Expected Outputs	Status 200 OK, the avatar link is returned: "http://localhost:5500/account/profile/avatar/meme5.jpg" If the user has no avatar, return "".
Pass/Fail Criteria	This case fails if the expected avatar link is missing, or the returned status is incorrect.

Test Case 4

Purpose	Check if user can upload his/her avatar using the API
Inputs (Request)	PATCH http://localhost:5500/account/profile/avatar/464b67/
Expected Outputs	Status 200 OK, the successful message is returned
Pass/Fail Criteria	This case fails if the expected message is not returned the returned status is incorrect.

Test Case 5

Purpose	Check if a guest can login to his/her existing account using the API
Inputs (Request)	PATCH http://localhost:5500/account/login/ Request Body: { "email": "nktesting@gmail.com" , "password": "TestingOnly123" }
Expected Outputs	Status 200 OK, the logged in account record is returned
Pass/Fail Criteria	This case fails if any account information in response body is missing, or the returned status is incorrect.

Test Case 6

Purpose	Check if a guest can sign up for a new account using the API
Inputs (Request)	POST http://localhost:5500/account/ Request Body: { "username": "testing" , "email": "nktesting@gmail.com" , "password": "TestingOnly123" , "IsGoogleSign": false }
Expected Outputs	Status 200 OK, the inserted account record is returned
Pass/Fail Criteria	This case fails if any account information in response body is missing, or the returned status is incorrect.

2.2.2 Follower Route

Test Case 1

Purpose	Check if user can get all his/her followed user list using the API
Inputs (Request)	GET http://localhost:5500/follow/followed/638f63
Expected Outputs	Status 200 OK, a list of followed user accounts is returned. If the user did not follow anyone, return an empty list.
Pass/Fail Criteria	This case fails if any information in response body is missing, or the returned status is incorrect.

Test Case 2

Purpose	Check if user can get all his/her follower list using the API
Inputs (Request)	GET http://localhost:5500/follow/follower/638f63

Expected Outputs	Status 200 OK, a list of follower accounts is returned. If the user has no follower, return an empty list.
Pass/Fail Criteria	This case fails if any information in response body is missing, or the returned status is incorrect.

Test Case 3

Purpose	Check if user can get all his/her pending follower list using the API
Inputs (Request)	GET http://localhost:5500/follow/pending/638f63
Expected Outputs	Status 200 OK, a list of pending follower accounts is returned. If the user has no pending follower, return an empty list.
Pass/Fail Criteria	This case fails if any information in response body is missing, or the returned status is incorrect.

Test Case 4

Purpose	Check if user can get a list of follow suggestion using the API
Inputs (Request)	GET http://localhost:5500/follow/suggestion/638f63
Expected Outputs	Status 200 OK, a list of pending follower accounts is returned. If the user followed all top 10 most followed users, return an empty list.
Pass/Fail Criteria	This case fails if any information in response body is missing, or the returned status is incorrect.

Test Case 5

Purpose	Check if user can get number of followers, followed user and pending followers using the API
Inputs (Request)	GET http://localhost:5500/follow/stat/638f63
Expected Outputs	Status 200 OK, the follow statistics are returned.
Pass/Fail Criteria	This case fails if any information in response body is missing, or the returned status is incorrect.

Test Case 6

Purpose	Check if user can follow another user using the API
Inputs (Request)	POST http://localhost:5500/follow/ Request body: { "followedUserId": "638f63", "followerUserId": "fafdb1" }
Expected Outputs	Status 200 OK, the inserted follow record is returned.
Pass/Fail Criteria	This case fails if any information in response body is missing, or the returned status is incorrect.

Test Case 7

Purpose	Check if user can accept a pending follower request using the API
Inputs (Request)	PATCH http://localhost:5500/follow/ Request body: { "followedUserId": "638f63", "followerUserId": "fafdb1" }
Expected Outputs	Status 200 OK, the mongodb default update record is returned.
Pass/Fail Criteria	This case fails if modifiedCount is 0, matchedCount is 0 or the returned status is incorrect.

Test Case 8

Purpose	Check if user can reject a pending follower request and unfollow a followed user using the API
Inputs (Request)	DELETE http://localhost:5500/follow/ Request query: { "followedUserId": "638f63", "followerUserId": "fafdb1" }
Expected Outputs	Status 200 OK, the successful message is returned: "Deleted successfully"

Pass/Fail Criteria	This case fails if the "Document not found or not deleted" message is returned or the returned status is incorrect.
--------------------	---

2.2.3 Tweet Route

Test Case 1

Purpose	Check if user can post a tweet using the API
Inputs (Request)	POST http://localhost:5500/tweet/ Request body: { "image": { "filename": "Screenshot 2023-04-13 at 5.33.17 PM.png", "contentType": "image/png"}, "userId": "fafdb1", "desc": "Testing"}
Expected Output	Status 200 OK, the successful message is returned: "File uploaded and stored in MongoDB"
Pass/Fail Criteria	This case fails if the expected message is not returned or the returned status is incorrect.

Test Case 2

Purpose	Check if user can delete his/her tweet using the API
Inputs (Request)	DELETE http://localhost:5500/tweet/643e578f6c36c54117d83900
Expected Outputs	Status 200 OK, the successful message is returned: "Post and image deleted successfully"
Pass/Fail Criteria	This case fails if the expected message is not returned, or the returned status is incorrect.

Test Case 3

Purpose	Check if user can get all the tweets posted by his/her following users using the API
Inputs (Request)	GET http://localhost:5500/tweet?limit=10&page=0&userId= 638f63
Expected Outputs	Status 200 OK, a list of tweets posted by following users is returned in response body.
Pass/Fail Criteria	This case fails if any tweet information is missing in response body or the returned status is incorrect.

2.2.4 Comment Route

Test Case 1

Purpose	Check if user can get all the comments by postId using the API
Inputs (Request)	GET http://localhost:5500/comment/:postId
Expected Outputs	Status 200 OK, a list of comments is returned. If there is no comment for the given postId, return an empty list.
Pass/Fail Criteria	This case fails if any comment information in response body is missing, or the returned status is incorrect.

Test Case 2

Purpose	Check if user can delete an existing comment using the API
Inputs (Request)	POST http://localhost:5500/comment/ Request body: { "postId": "6ab2bbe1-d", "userId": "638f63",

	"comment": "testing"}
Expected Outputs	Status 200 OK, and returns the saved comment in response body.
Pass/Fail Criteria	This case fails if any comment information in response body is missing or the returned status is incorrect.

2.2.5 Like Route

Test Case 1

Purpose	Check if user can like a post using the API given that the user didn't dislike that post before
Inputs (Request)	POST http://localhost:5500/like/ Request body: { "postId": "6ab2bbe1-d", "userId": "638f63" }
Expected Outputs	Status 200 OK and returns the saved like record in response body.
Pass/Fail Criteria	This case fails if any information in response body is missing, or the returned status is incorrect.

Test Case 2

Purpose	Check if user can like a post using the API given that the user disliked that post before
Inputs (Request)	POST http://localhost:5500/like/ Request body: { "postId": "6ab2bbe1-d", "userId": "638f63" }
Expected Outputs	Status 200 OK and returns the message: "You have already disliked this post. Please un-dislike it before liking it."
Pass/Fail Criteria	This case fails if the expected message is not returned or the returned status is incorrect.

Test Case 3

Purpose	Check if user can unlike a post using the API given that the user liked that post before
Inputs (Request)	Delete http://localhost:5500/like/ Request query: { "postId": "6ab2bbe1-d", "userId": "638f63" }
Expected Outputs	Status 200 OK and returns the message: "Deleted successfully"
Pass/Fail Criteria	This case fails if the expected message is not returned, or the returned status is incorrect.

Test Case 4

Purpose	Check if user has already liked the post using the API
Inputs (Request)	GET http://localhost:5500/like/ Request query: { "postId": "6ab2bbe1-d", "userId": "638f63" }
Expected Outputs	Status 200 OK and returns the like data as response body.
Pass/Fail Criteria	This case fails if any information in response body is missing, or the returned status is incorrect.

2.2.6 Dislike Route

Test Case 1

Purpose	Check if user can dislike a post using the API given that the user didn't like that post before
Inputs (Request)	POST http://localhost:5500/dislike/ Request body: { "postId": "6ab2bbe1-d", "userId": "638f63" }
Expected Outputs	Status 200 OK and returns the saved dislike record.
Pass/Fail Criteria	This case fails if any information in response body is missing, or the returned status is incorrect.

Test Case 2

Purpose	Check if user can dislike a post using the API given that the user liked that post before
Inputs (Request)	POST http://localhost:5500/dislike/ Request body: {"postId": "6ab2bbe1-d", "userId": "638f63"}
Expected Outputs	Status 200 OK and returns the message: "You have already liked this post. Please unlike it before disliking."
Pass/Fail Criteria	This case fails if the expected message is not returned, or the returned status is incorrect.

Test Case 3

Purpose	Check if user can un-dislike a post using the API given that the user disliked that post before
Inputs (Request)	Delete http://localhost:5500/dislike/ Request query: {"postId": "6ab2bbe1-d", "userId": "638f63"}
Expected Outputs	Status 200 OK and returns the message: "Deleted successfully"
Pass/Fail Criteria	This case fails if the expected message is not returned, or the returned status is incorrect.

Test Case 4

Purpose	Check if user has already disliked the post using the API
Inputs (Request)	Get http://localhost:5500/dislike/ Request query: {"postId": "6ab2bbe1-d", "userId": "638f63"}
Expected Outputs	Status 200 OK and returns the like data as response body.
Pass/Fail Criteria	This case fails if any information in response body is missing, or the returned status is incorrect.

3. Security

Test Case 1

Purpose	To test if the attacker can bypass the password checking in the sign-up page using no SQL injection.
Inputs	In the sign up page: Email: 123@link.cuhk.edu.hk Name: name123 Password: "\$gt": "" Confirmed Password: "\$gt": "" Captcha: True
Expected Outputs	A warning message regarding the invalid password will be shown.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output. And fails if the verification email is sent to that email.

Test Case 2

Purpose	To test if the attacker can bypass the password checking in the log in page using no SQL injection.
Inputs	Email: 123@link.cuhk.edu.hk (Must be an existing account) Password: "\$ne": ""

Expected Outputs	A warning message regarding the invalid password will be shown and he cannot log into that account.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output. And fails if login is successful and home page is redirected.

Test Case 3

Purpose	To test if the attacker can attack the website using HTML injection and XSS attack in comment section.
Inputs	In the text area of the comment section, input: <h1>HTML Injection testing</h1> <script>alert ('XSS attack') ;</script> Click the comment button
Expected Outputs	The plain text version will be shown.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output. And fails if “HTML Injection” is shown in h1 heading OR “XSS attack” alert script is run.

Test Case 4

Purpose	Check if the attacker can bypass the deactivation using the forgotten password.
Inputs	For the deactivated account, click the forgot password and input that email for the account. Then click the reset password link in the email box and type in the new password.
Expected Outputs	Warning message regarding the deactivation will be shown and the log in is rejected but the new password will be set.
Pass/Fail Criteria	Pass this test case if and only if the outcome is exactly the same as the expected output. And fails if login successfully and redirected to the home page OR the new password is not set.