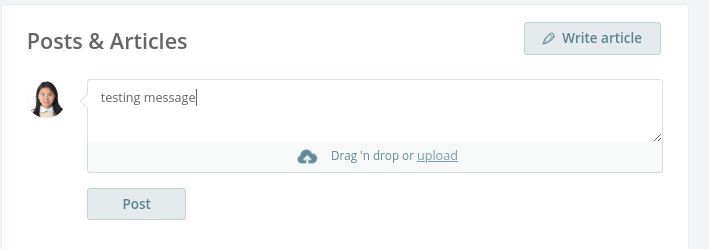
Part 2

Document any possible test case that you would perform in order to test the following form embedded in a website. Include testing about functionality, security, performance, etc.

Part 2 Answers:

1. UI testing: mainly focus on clarity of images, color matching, and style
2. Functionality test cases
3. “write article” button:
   1. check if the “write article” button is displyed as expected.
   2. Click the “write article” button, check if user can input text
4. Text input box:
   1. Text length testing case (There should be a length limitation of the input, for example 600 characters), then we need to design the test cases with
      1. Input text with “text length” < 600 to post and check if it is saved and displayed as expected.
      2. Input text with “text length=600” to post and check if it is saved and displayed as expected.
      3. Input text with “text length” > 600 to check if there is a check warning from the UI side, and if the invalid data will be sent to server, and how the server will handle this invalid input.
      4. Copy the text from existing file and paste the content into the input box, check if it is saved and posted as expected.
      5. When the user input text exceeds the displaying range of the text input box, there should be a scrollbar to help user preview the content
   2. Data type testing case
      1. Input text with normal letters
      2. Input text with numbers
      3. Input text with special characters, like `~!@#$%^&\*()\_+-={}[]|\:;”’ <>,./?, NUL,\n
      4. Input Chinese
   3. Empty input: post with empty input to check that if there will be a warning message or if it will cause an error
   4. Invalid input check:input invalid text, check if there is warning from the UI.
5. Drag a file to upload(There should be a limitation to the dragger file, for example 1M)
   1. Drag a file that if smaller than 1M
      1. The file(<1M) we dragged only includes normal letters
      2. The file we dragged includes images
      3. The file we dragged includes specifical characters
      4. The file we dragged includes some chinese characters
      5. When there are more words than the scope input windows, user should be able to preview the text
   2. Drag file that is 1M, and test with the conditions in section 3.1 to check if the file content can be posted correctly.
      1. The file(<1M) we dragged only includes normal letters
      2. The file we dragged includes images
      3. The file we dragged includes specifical characters
      4. The file we dragged includes some chinese characters
   3. Drag file that is bigger than 1M, check if there is a warning from UI side, or if it will cause an error.
6. Click the upload button to upload(There should be a limitation to the type, size of the dragger file’s, for example it should be txt, doc file 1M)
   1. Test with different file type
      1. Upload a valid txt file, check if the file content will be displayed correctly.
      2. Upload a valid doc file, check if the file content will be displayed correctly.
      3. Upload an invalid exe file, it should be rejected and there should be a warning message.
   2. Test with different file size
      1. Upload a file which is smaller than 1M
      2. Upload a file that is 1M
      3. Upload a file that is bigger than 1M
   3. Click this “upload” button continuously, and check if only one upload window is opened.
7. Post
   1. Check “post” button is displayed as expected.
   2. Click “post” button, there should be a confirm button to ask user double confirm. User can click “Cancel” button to cancel the post action, or click the “OK” button to post the input text. If user choose “OK”, we need to make sure the post or artical is sent and saved as expected, check the posts&articles are displayed correctly.
   3. Click this button continuously, and check if several items are posted to the server.
8. Check the “profile picture” and all the text in the buttons and menus are displayed correctly
9. API testing with postman or jmeteror fiddler
10. Session testing:
    1. Validity time of the session using test case design method of equivalent class and boundary values etc
    2. Check whether session is safe
11. Cookies testing:
    1. Expiration date of cookie
    2. Whether cookies are safe
    3. Rejecting cookies
    4. Testing the first party & thid party cookies
    5. Testing session cookies
    6. Testing persisitent cookies
12. Compatibility
    1. Browser
       1. Access the posts&article website with different browsers , like IE, chrome, firefox etc to check the compatibility.
       2. Try to input text/upload file/drag file and post to server with different browsers.
    2. OS:
       1. Windows
       2. MacOS
    3. Different screens and resolution ratio
13. User experience testing
14. Security
    1. Check if the username/password are displayed in the post message
    2. SQL injection check: for example test if user can access with the normal url combinded with a piece of SQL like “1=1”
    3. Check if user can access the server before login the posts&articles system
    4. Check if user can upload a file that includes “Trojan files”, if the UI system will reject it.
15. Performance
    1. Check how long it takes to post and display the content for users
    2. Implement automation test case to input text and post with a static frenquency，normal peak load and other abnormal traffic to monitor the CPU/Memory/IO usage of the system
16. Stress Testing
    1. It refers to testing the system's bottlenecks or unacceptable performance points to obtain the maximum service level that the system can provide;
    2. Implement automation test case to simulate more and more users to visit the system, and monitor the system CPU/Memory/IO usage of the system at same time. Then obtain the maximun service level that the system can provide, like how many users can visit the system at the same time.
17. Load Testing
    1. Simulate the users action with thread, start the thread one by one, watch the system performance at the same time, try to get the threadhold of the system.
       1. Test with different actions, including inputting text and posting to server, dragging a file and post to server, uploading a file and post to server etc
18. Concurrency Testing
    1. Simulate multi-users to access the post&article system to “write text”/drag file/upload file concurrently, and check all the users can post the content successfully
19. Endurance Testing & Reliability Testing:
    1. Simulate multiple users with the threads, and access the system with a long, monitor the change of CPU usage, memory usage, IO usage to check if there are some issues about the resource release.
20. Spike Testing
    1. Simulate the users with more than the maximum load in a short time, check if the system can handle correctly and will not crash or lost data.
21. Failover Testing
    1. When the user input text/click the post button, simulate the failover conditions:
       1. Close and open the broswer
       2. Disconnect the network, and reconnect