

# Test Report

---

---

# CONTENTS

|                                |          |
|--------------------------------|----------|
| <b>Tests and defects .....</b> | <b>2</b> |
| Testcase Summary.....          | 2.1      |
| Testcase coverage.....         | 2.2      |
| Defect Management .....        | 2.3      |
| <b>Test Execution .....</b>    | <b>3</b> |

## Testcase Summary:

| Functionality        | Total test cases | No. of untested cases | Passed cases | Failed cases | Execution percentage | Completion percentage | state |
|----------------------|------------------|-----------------------|--------------|--------------|----------------------|-----------------------|-------|
| user registration    | 5                | 0                     | 5            | 0            | %100                 | %100                  | Pass  |
| User login           | 6                | 1                     | 5            | 0            | %83.3                | %100                  | Pass  |
| User profile         | 7                | 0                     | 7            | 0            | %100                 | %100                  | Pass  |
| Skills               | 3                | 0                     | 3            | 0            | %100                 | %100                  | Pass  |
| Quizzes              | 6                | 0                     | 6            | 0            | %100                 | %100                  | Pass  |
| Vocabulary page/game | 2                | 0                     | 2            | 0            | %100                 | %100                  | Pass  |
| Read more button     | 1                | 0                     | 1            | 0            | %100                 | %100                  | Pass  |
| Contact/email        | 1                | 0                     | 1            | 0            | %100                 | %100                  | Pass  |
| Website logo         | 1                | 0                     | 1            | 0            | %100                 | %100                  | Pass  |

## Testcase coverage:

| Functionality        | Prioritization | Total possible scenarios | Number of actual cases | Test coverage percentage |
|----------------------|----------------|--------------------------|------------------------|--------------------------|
| user registration    | High           | 5                        | 5                      | %100                     |
| User login           | High           | 6                        | 5                      | %83.3                    |
| User profile         | High           | 7                        | 7                      | %100                     |
| Skills               | medium         | 3                        | 3                      | %100                     |
| Quizzes              | medium         | 6                        | 6                      | %100                     |
| Vocabulary page/game | Low            | 2                        | 2                      | %100                     |
| Read more button     | Low            | 1                        | 1                      | %100                     |
| Contact/email        | Low            | 1                        | 1                      | %100                     |
| Website logo         | Low            | 1                        | 1                      | %100                     |

## Defect Management:

| Defect Level | Defect Count | Resolved Defects | Outstanding Defects | Defect description  |
|--------------|--------------|------------------|---------------------|---|
| High         |              |                  |                     |   |
| Average      |              |                  |                     |   |
| Low          | 1            | 1                | 0                   | Website display unmeaningful message when user finish exam without been logged in |

### Resolved Defects:

Reports have been sent to the development team and defects have been fixed (message has changed).

## Test Execution

### TC\_01- User Registration\_1

Attempt to register new user with valid information:

```
Test1.py x
1 #User Registration_1
2 import time
3 from websiteFinder import Testedwebsite
4
5 Test1 = Testedwebsite()
6 driver2 = Test1.driver
7
8 #Test-1
9
10 #cclick the Login button in top right
11 log_in_header_button = Test1.element_finder_by_path("//button[@id='login-btn-popup']")
12 driver2.execute_script("arguments[0].click();", log_in_header_button)# click it by with javascript
13 time.sleep(4)
14
15 #click Register
16 reg_link = Test1.element_finder_by_path("//a[@class='register-link']")
17 driver2.execute_script("arguments[0].click();", reg_link)
18 time.sleep(4)
19
20 #put username
21 UserName_reg = Test1.element_finder_by_path("//input[@id='register_user']")
22 UserName_reg.send_keys("mohanad")
23
24 #put email
25 email_reg = Test1.element_finder_by_path("//input[@id='register_email']")
26 email_reg.send_keys("moh@gmail.com")
27 time.sleep(4)
28
29 #put password
30 pass_reg = Test1.element_finder_by_path("//input[@id='register_pass']")
31 pass_reg.send_keys("123456")
32 time.sleep(1)
33
34 #Click Register button
35 check_box_cond = Test1.element_finder_by_path("//div[@class='form-box_register']//div[@class='remember-forgot']//input[1]")
36 check_box_cond.click()
37 time.sleep(3)
38
39 #Register success
40 reg_button = Test1.element_finder_by_path("//button[@id='register_btn']")
41 driver2.execute_script("arguments[0].click();", reg_button)
42
43 time.sleep(10)
```

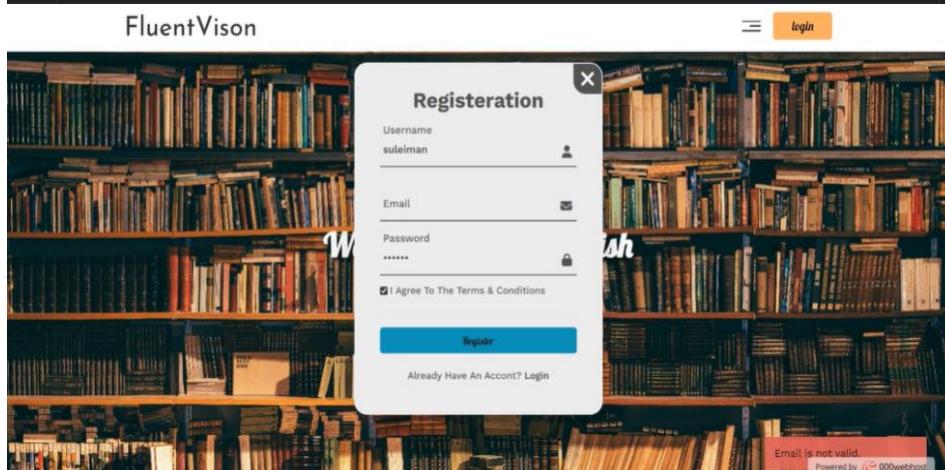


The screenshot shows a library website with a background image of bookshelves. The main content area displays the text "Welcome To Your English Library". In the bottom right corner, there is a green success message box that says "register success". The FluentVision logo is at the top left, and a login button is at the top right.

## TC\_02- User Registration\_2

Attempt to register new user with invalid email or leave the email section empty:

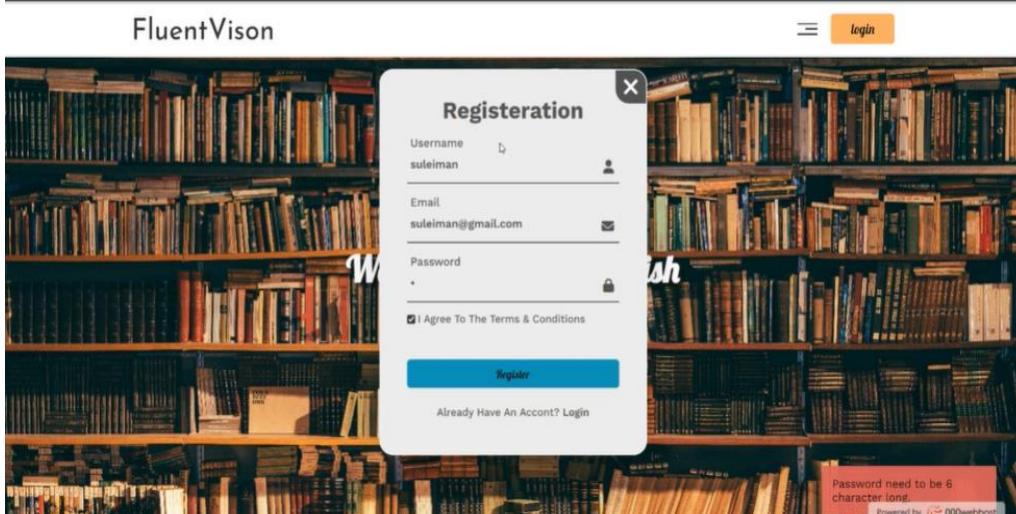
```
 1 #User Registration_2
 2
 3 > import ...
 4
 5 Test2 = Testedwebsite()
 6 driver2 = Test2.driver
 7
 8 #Test_2
 9
10 #click the Login button in top right
11 log_in_header_button = Test2.element_finder_by_path("//button[@id='login-btn-popup']")
12 driver2.execute_script("arguments[0].click();", log_in_header_button)
13
14 #click Register
15 reg_link = Test2.element_finder_by_path("//a[@class='register-link']")
16 driver2.execute_script("arguments[0].click();", reg_link)
17
18 #put username
19 UserName_reg = Test2.element_finder_by_path("//input[@id='register_user']")
20 UserName_reg.send_keys("suleiman")
21
22 #email is empty
23 email_reg = Test2.element_finder_by_path("//input[@id='register_email']")
24 email_reg.send_keys("")
25
26
27 #put password
28 pass_reg = Test2.element_finder_by_path("//input[@id='register_pass']")
29 pass_reg.send_keys("123456")
30 time.sleep(2)
31
32 #Click Register button
33 check_box_cond = Test2.element_finder_by_path("//div[@class='form-box register']//div[@class='remember-forgot']//input[1]")
34 check_box_cond.click()
35 time.sleep(4)
36
37 #Email is not valid
38 reg_button = Test2.element_finder_by_path("//button[@id='register_btn']")
39 driver2.execute_script("arguments[0].click();", reg_button)
40 time.sleep(4)
41
42 time.sleep(10)
```



## TC\_03- User Registration\_3

Attempt to register new user with invalid password or empty password:

```
Test3.py x Test4.py Test2.py Test1.py
1 #User_Registraion_3
2
3 > import ...
4
5
6
7 Test3 = Testedwebsite()
8 driver2 = Test3.driver
9
10 #Test-3
11
12 #click the Login button in top right
13 log_in_header_button = Test3.element_finder_by_path("//button[@id='login-btn-popup']")
14 driver2.execute_script("arguments[0].click();", log_in_header_button)
15
16 #click Register
17 reg_link = Test3.element_finder_by_path("//a[@class='register-link']")
18 driver2.execute_script("arguments[0].click();", reg_link)
19
20 #put username
21 UserName_reg = Test3.element_finder_by_path("//input[@id='register_user']")
22 UserName_reg.send_keys("suleiman")
23
24 #put email
25 email_reg = Test3.element_finder_by_path("//input[@id='register_email']")
26 email_reg.send_keys("suleiman@gmail.com")
27
28 #Not completed password
29 pass_reg = Test3.element_finder_by_path("//input[@id='register_pass']")
30 pass_reg.send_keys("1")
31
32 #Click Register button
33 check_box_cond = test3.element_finder_by_path("//div[@class='form-box register']//div[@class='remember-forgot']//input[1]")
34 check_box_cond.click()
35 time.sleep(4)
36
37 #Password need to be 6 character long
38 reg_button = Test3.element_finder_by_path("//button[@id='register_btn']")
39 driver2.execute_script("arguments[0].click();", reg_button)
40
41 time.sleep(10)
```



## TC\_04- User Registration\_4

Attempt to register new user with existing user information:

```
 1 #User Registration_4
 2
 3 > import ...
 4
 5 Test4 = Testedwebsite()
 6 driver2 = Test4.driver
 7
 8 #Test-4
 9
10 #click the Login button in top right
11 log_in_header_button = Test4.element_finder_by_path("//button[@id='login-btn-popup']")
12 driver2.execute_script("arguments[0].click()", log_in_header_button)
13
14 #click Register
15 reg_link = Test4.element_finder_by_path("//a[@class='register-link']")
16 driver2.execute_script("arguments[0].click()", reg_link)
17
18 #put username
19 UserName_reg = Test4.element_finder_by_path("//input[@id='register_user']")
20 UserName_reg.send_keys("mohanad")
21
22 #put email
23 email_reg = Test4.element_finder_by_path("//input[@id='register_email']")
24 email_reg.send_keys("moh@gmail.com")
25
26
27 #put password
28 pass_reg = Test4.element_finder_by_path("//input[@id='register_pass']")
29 pass_reg.send_keys("123456")
30
31 #Click Register button
32 check_box_cond = test4.element_finder_by_path("//div[@class='form-box register']//div[@class='remember-forgot']//input[1]")
33 check_box_cond.click()
34 time.sleep(4)
35
36 #User already exist
37 reg_button = Test4.element_finder_by_path("//button[@id='register_btn']")
38 driver2.execute_script("arguments[0].click()", reg_button)
39
40
41 time.sleep(10)
```

FluentVision

Registration

Username  
mohanad

Email  
moh@gmail.com

Password  
\*\*\*\*\*

I Agree To The Terms & Conditions

Register

Already Have An Account? Login

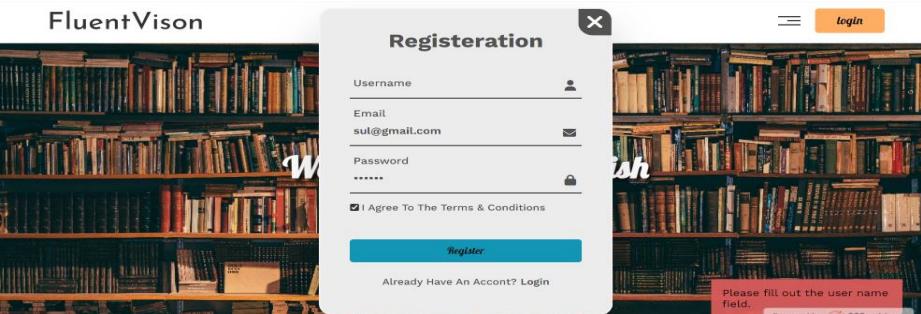
User already exist.

## TC\_05- User Registration\_5

Check if the website will display meaningful error message when the user attempts to press Register when username sections is empty:

```
Test3.py Test4.py Test5.py < Test2.py Test1.py
1 #User_Registraion_5
2
3 > import ...
4
5 Test5 = Testedwebsite()
6 driver2 = Test5.driver
7
8 #Test_5
9
10 #click the Login button in top right
11 log_in_header_button = Test5.element_finder_by_path("//button[@id='login-btn-popup']")
12 driver2.execute_script("arguments[0].click();", log_in_header_button)
13
14 #click Register
15 reg_link = Test5.element_finder_by_path("//a[@class='register-link']")
16 driver2.execute_script("arguments[0].click();", reg_link)
17
18 #no_username
19
20 #put_email
21 email_reg = Test5.element_finder_by_path("//input[@id='register_email']")
22 email_reg.send_keys("sul@gmail.com")
23
24
25
26 #put_password
27 pass_reg = Test5.element_finder_by_path("//input[@id='register_pass']")
28 pass_reg.send_keys("123456")
29
30 #Click Register button
31 check_box_cond = Test5.element_finder_by_path("//div[@class='form-box register']//div[@class='remember-forgot']//input[1]")
32 check_box_cond.click()
33 time.sleep(4)
34
35 #Please fill out the username field
36 reg_button = Test5.element_finder_by_path("//button[@id='register_btn']")
37 driver2.execute_script("arguments[0].click();", reg_button)
38
39 time.sleep(10)
```

**FluentVison**



## TC\_006 User login\_1

Attempt to login with valid information:

```
 1 #User_login_1
 2 > import ...
 3
 4 #Test-6
 5
 6 Test = Testedwebsite()
 7 driver = Test.driver
 8
 9 #click the Login button in top right
10 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
11 driver.execute_script("arguments[0].click();", log_in_header_button)
12
13 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")
14 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")
15 log_in_button = Test.element_finder_by_path("//button[@id='loginBtn']")
16
17 #put_email_and_password
18 log_in_email.send_keys("moh@gmail.com")
19 log_in_pass.send_keys("123456")
20
21 time.sleep(4)
22
23 #click_Login_button
24 driver.execute_script("arguments[0].click();", log_in_button)
```

FluentVision



## TC\_07- User login\_2

Attempt to login with invalid email:

The screenshot shows a browser window with several tabs open at the top: Test5.py, Test6.py, Test7.py (which is active), Test8.py, and Test9.py. Below the tabs is a code editor window displaying Python test script code. The script is titled '#User\_login\_2' and includes imports, a Test class definition, and logic for logging in with an invalid email ('mm@gmail.com') and password ('123456'). A screenshot of a login page is shown in the foreground. The page has a bookshelf background and a 'Login' button in the top right. The login form fields show 'Email: mm@gmail.com' and 'Password: .....'. Below the form are 'Remember Me' and 'Forgot Password?' checkboxes. A blue 'Login' button is centered below the form. At the bottom of the page, there are links for 'Don't Have An Accout? Register' and a red error message box containing the text 'User does not exist. Please check your email and password.' The FluentVison logo is visible at the bottom left of the page.

```
1 #User_login_2
2 > import ...
3 #Test_7
4
5 Test = Testedwebsite()
6 driver = Test.driver
7
8 #click the Login button in top right
9 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
10 driver.execute_script("arguments[0].click();", log_in_header_button)
11
12 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")
13 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")
14 log_in_button = Test.element_finder_by_path("//button[@id='loginBtn']")
15
16 #invalid_email
17 log_in_email.send_keys("mm@gmail.com")
18
19 #put_password
20 log_in_pass.send_keys("123456")
21 time.sleep(4)
22
23 #click Login button
24
25 log_in_button.click()
```

## TC\_08- User login\_3

Attempt to login with invalid password:

The screenshot shows a browser window with several tabs open at the top: Test5.py, Test6.py, Test9.py, Test10.py, Test8.py (which is active), and Test7.py. Below the tabs is a code editor window displaying Python test script code. The script is titled '#User\_login\_3' and includes imports, a Test class definition, and logic for logging in with a valid email ('moh@gmail.com') and an invalid password ('123555'). A screenshot of a login page is shown in the foreground. The page has a bookshelf background and a 'Login' button in the top right. The login form fields show 'Email: moh@gmail.com' and 'Password: .....'. Below the form are 'Remember Me' and 'Forgot Password?' checkboxes. A blue 'Login' button is centered below the form. At the bottom of the page, there are links for 'Don't Have An Accout? Register' and a red error message box containing the text 'User does not exist. Please check your email and password.' The FluentVison logo is visible at the bottom left of the page.

```
1 #User_login_3
2 > import ...
3 #Test_8
4 Test = Testedwebsite()
5 driver = Test.driver
6
7 #click the Login button in top right
8 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
9 driver.execute_script("arguments[0].click();", log_in_header_button)
10
11 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")
12 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")
13 log_in_button = Test.element_finder_by_path("//button[@id='loginBtn']")
14
15 #put_email
16 log_in_email.send_keys("moh@gmail.com")
17
18 #invalid_password
19 log_in_pass.send_keys("123555")
20 time.sleep(4)
21
22 #click Login button
23
24 log_in_button.click()
```

## TC\_09- User login\_4

Attempt to login with non-existing user:

```
Test5.py Test6.py Test9.py X Test10.py Test8.py Test7.py
1 #User_login_4
2
3 > import ...
4
5 #Test_9
6
7 Test = Testedwebsite()
8 driver = Test.driver
9
10
11 #click the Login button in top right
12 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
13 driver.execute_script("arguments[0].click();", log_in_header_button)
14
15 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")
16 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")
17 log_in_button = Test.element_finder_by_path("//button[@id='loginBtn']")
18
19
20 #put non-existing username and password
21 log_in_email.send_keys("salimali@gmail.com")
22 log_in_pass.send_keys("123123")
23 time.sleep(4)
24
25 #click Login button
26 driver.execute_script("arguments[0].click();", log_in_button)
```

FluentVision

## TC\_010- login\_5

Check if the website will display meaningful error message when the user attempts to press login when one of the sections is empty:

```
Test5.py Test6.py Test9.py Test10.py X Test8.py Test7.py
1 #User_login_5
2
3 > import ...
4
5 #Test_10
6
7 Test = Testedwebsite()
8 driver = Test.driver
9
10
11 #click the Login button in top right
12 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
13 driver.execute_script("arguments[0].click();", log_in_header_button)
14 time.sleep(3)
15
16 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")
17 log_in_button = Test.element_finder_by_path("//button[@id='loginBtn']")
18
19
20 #empty username
21
22 log_in_pass.send_keys("123456")
23 time.sleep(4)
24
25 #click Login button
26 driver.execute_script("arguments[0].click();", log_in_button)
```

FluentVision

## TC\_011- User profile\_1

Updating the username:

```
Test10.py Test11.py
26 #click_login_button
27 driver.execute_script("arguments[0].click();", log_in_button)
28 time.sleep(4)
29
30 #click_menu
31 user_menu = Test.element_finder_by_path("//img[@src='../../../../icon/user-man.png']")
32 Test.actions.move_to_element(user_menu).perform()
33 time.sleep(4)
34
35 #click_user_profile
36 user_profile_in_user_menu = Test.element_finder_by_path("//a[normalize-space()='My Profile']")
37 user_profile_in_user_menu.click()
38 time.sleep(3)
39
40
41 user_name_in_profile = Test.element_finder_by_path("//input[@id='getusername']")
42 time.sleep(3)
43
44 user_name_in_profile.clear()
45
46 #update your username
47 user_name_in_profile.send_keys("mohanad1")
48 user_name_in_profile.click()
49 time.sleep(2)

Test10.py Test11.py
1 #User_profile_1
2 > import ...
3
4
5 #Test-11
6
7 Test = Testedwebsite()
8 driver = Test.driver
9
10 #click the Login button in top right
11 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
12 driver.execute_script("arguments[0].click();", log_in_header_button)
13
14 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")
15 log_in_pass = Test.element_finder_by_path("//input[@id='LoginPass']")
16 log_in_button = Test.element_finder_by_path("//button[@id='loginBtn']")
17
18 #put_email
19 log_in_email.send_keys("moh@gmail.com")
20
21 #put_password
22 log_in_pass.send_keys("123456")
23 time.sleep(3)
24
25
```

FluentVison



My Profile Settings

General

Change password

Upload new photo    Reset  
Allowed JPG, GIF or PNG. Max size of 800K

My Grades

Username

E-mail

**Save changes**

Changes have been saved successfully.  
Powered by 000webhost

## TC\_012- User profile\_2

Updating the Email:

```
Test12.py x Test11.py Test13.py
1 #User profile_2
2 > import ...
3 #Test-12
4
5 Test = Testedwebsite()
6 driver = Test.driver
7
8 #click the Login button in top right
9 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
10 driver.execute_script("arguments[0].click()", log_in_header_button)
11
12 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")
13 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")
14 log_in_button = Test.element_finder_by_path("//button[@id='loginBtn']")
15
16 #put email
17 log_in_email.send_keys("moh@gmail.com")
18
19 #put password
20 log_in_pass.send_keys("123456")
21 time.sleep(4)
22
23 #click login button
24 driver.execute_script("arguments[0].click()", log_in_button)

Test11.py x Test12.py
27
28 #click menu
29 user_menu = Test.element_finder_by_path("//img[@src='../../../../icon/user-man.png']")
30 Test.actions.move_to_element(user_menu).perform()
31 time.sleep(5)
32
33 #click user profile
34 user_profile_in_user_menu = Test.element_finder_by_path("//a[normalize-space()='My Profile']")
35 user_profile_in_user_menu.click()
36 time.sleep(4)
37
38 user_email_in_profile = Test.element_finder_by_path("//input[@id='getuseremail']")
39 time.sleep(3)
40
41 #update your email
42 user_email_in_profile.clear()
43 user_email_in_profile.send_keys("moha@gmail.com")
44 time.sleep(3)
45
46 #click on save changes button
47 save_changes_button = Test.element_finder_by_path("//button[@id='updateUser']")
48 save_changes_button.click()
49 time.sleep(10)
```

### My Profile Settings

The screenshot shows a web-based profile settings form. At the top, there's a 'General' section with fields for 'Change password' and 'My Grades'. Below this is a file upload area with 'Upload new photo' and 'Reset' buttons, noting a max size of 800K. The main form contains fields for 'Username' (mohanad1) and 'E-mail' (moha@gmail.com). A 'Save changes' button is at the bottom left, and a green success message 'Changes have been saved successfully.' is displayed on the right. The page is powered by 000webhost.

## TC\_013- User profile\_3

Updating the password:

```
 1 #User_profile_3
 2 > import ...
 3
 4 #Test-13
 5
 6 Test = Testedwebsite()
 7 driver = Test.driver
 8
 9 #click the Login button in top right
10 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
11 driver.execute_script("arguments[0].click()", log_in_header_button)
12 time.sleep(4)
13
14 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")
15 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")
16 log_in_button = Test.element_finder_by_path("//button[@id='LoginBtn']")
17
18
19 #put_email
20 log_in_email.send_keys("moha@gmail.com")
21
22 #put_password
23 log_in_pass.send_keys("123456")
24 time.sleep(4)
25
26 #click login button
27 driver.execute_script("arguments[0].click()", log_in_button)
28
29 #click menu
30 user_menu = Test.element_finder_by_path("//img[@src='../../../../icon/user-man.png']")
31 Test.actions.move_to_element(user_menu).perform()
32 time.sleep(4)
33
34 #click user profile
35 user_profile_in_user_menu = Test.element_finder_by_path("//a[normalize-space()='My Profile']")
36 user_profile_in_user_menu.click()
37 time.sleep(4)
38
39 #click change password
40 change_pass = Test.element_finder_by_path("//a[normalize-space()='Change password']")
41 change_pass.click()
42 time.sleep(2)
43
44 #update your password
45 curr_pass = Test.element_finder_by_path("//input[@id='currentpassword']")
46 curr_pass.send_keys("123456")
47 time.sleep(1)
48 new_pass = Test.element_finder_by_path("//input[@id='newpassword']")
49 new_pass.send_keys("123123")
50 time.sleep(1)
51 rep_pass = Test.element_finder_by_path("//input[@id='newpassword2']")
52 rep_pass.send_keys("123123")
53 time.sleep(5)
54
55 #click on save changes button
56 save_changes = Test.element_finder_by_path("//button[@id='updateUserPassword']")
57 save_changes.click()
58 time.sleep(10)
```

FluentVison



My Profile Settings

|                 |   |
|-----------------|---|
| General         | Current password  |
| Change password | *****   |
| My Grades       | New password  |
|                 | *****   |
|                 | Repeat new password   |
|                 | *****   |
|                 | <input type="button" value="Save changes"/>   |
|                 | <div style="background-color: green; color: white; padding: 5px;">Changes have been saved successfully.</div> |

## TC\_014- User profile\_4

updating the username or email section with the same current information:

```
Test13.py Test14.py x
1 #User_profile_4
2 > import ...
3
4 #Test-14
5
6 test = Testedwebsite()
7 driver = Test.driver
8
9 #click the Login button in top right
10 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
11 driver.execute_script("arguments[0].click();", log_in_header_button)
12
13 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")
14 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")
15 log_in_button = Test.element_finder_by_path("//button[@id='LoginBtn']")
16
17 #put_email
18 log_in_email.send_keys("moha@gmail.com")
19
20 #put_password
21 log_in_pass.send_keys("123123")
22 time.sleep(4)
23
24 #click login button
25 log_in_button.click()
26
27 #click login button
28 driver.execute_script("arguments[0].click();", log_in_button)
29 time.sleep(3)
30
31 #click menu
32 user_menu = Test.element_finder_by_path("//img[@src='../../../../icon/user-man.png']")
33 Test.actions.move_to_element(user_menu).perform()
34 time.sleep(4)
35
36 #click user profile
37 user_profile_in_user_menu = Test.element_finder_by_path("//a[normalize-space()='My Profile']")
38 user_profile_in_user_menu.click()
39
40 user_email_in_profile = Test.element_finder_by_path("//input[@id='getuseremail']")
41 time.sleep(3)
42
43 #same_email
44 user_email_in_profile.clear()
45 user_email_in_profile.send_keys("moha@gmail.com")
46 time.sleep(4)
47
48 #click on save changes button
49 save_changes_button = Test.element_finder_by_path("//button[@id='updateUser']")
50 save_changes_button.click()
```

FluentVision



### My Profile Settings

The screenshot shows a web-based profile settings interface. At the top left, there's a 'General' tab. On the left side, there are sections for 'Change password' and 'My Grades'. The main area contains fields for 'Username' (with value 'mohanad1') and 'E-mail' (with value 'moha@gmail.com'). To the right of these fields is a 'Save changes' button. Below the form, a red banner displays the message 'No changes were made.' At the bottom right of the banner, it says 'Powered by 000webhost'.

## TC\_015- User profile\_5

Updating the password with the same current password:

```
Test13.py Test14.py Test15.py ×
1 #User_profile_5
2 > import ...
3
4 #Test-15
5
6 Test = Testedwebsite()
7 driver = Test.driver
8
9 #click the Login button in top right
10 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
11 driver.execute_script("arguments[0].click()", log_in_header_button)
12
13 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")
14 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")
15 log_in_button = Test.element_finder_by_path("//button[@id='loginBtn']")
16 #put_email
17 log_in_email.send_keys("moha@gmail.com")
18 #put_password
19 log_in_pass.send_keys("123123")
20 time.sleep(4)
21 #click login button
22 driver.execute_script("arguments[0].click()", log_in_button)
23 time.sleep(2)
24
25
26
27 #click menu
28 user_menu = Test.element_finder_by_path("//img[@src='../../../../icon/user-man.png']")
29 Test.actions.move_to_element(user_menu).perform()
30 time.sleep(4)
31 #click user profile
32 user_profile_in_user_menu = Test.element_finder_by_path("//a[normalize-space()='My Profile']")
33 user_profile_in_user_menu.click()
34 time.sleep(3)
35 #to_change_password
36 change_pass = Test.element_finder_by_path("//a[normalize-space()='Change password']")
37 change_pass.click()
38 time.sleep(2)
39 curr_pass = Test.element_finder_by_path("//input[@id='currentpassword']")
40 curr_pass.send_keys("123123")
41 time.sleep(1)
42 new_pass = Test.element_finder_by_path("//input[@id='newpassword']")
43 new_pass.send_keys("123123")
44 time.sleep(1)
45 rep_pass = Test.element_finder_by_path("//input[@id='newpassword2']")
46 rep_pass.send_keys("123123")
47 time.sleep(4)
48 #click_on_save_changes_button
49 save_changes = Test.element_finder_by_path("//button[@id='updateUserPassword']")
50 save_changes.click()
```

FluentVision

My Profile Settings

|                 |  |
|-----------------|--|
| General         | Current password   |
| Change password | *****  |
| My Grades       | New password   |
|                 | *****  |
|                 | Repeat new password  |
|                 | *****  |
|                 | <input type="button" value="Save changes"/>                |
|                 | New password cannot be the<br>same as the current password |

## TC\_016- User profile\_6

Updating the password but the current password and repeat new password section are different:

```
Test13.py Test14.py Test15.py Test16.py ×
1 #User_profile_6
2 > import ...
3
4 #Test-16
5
6 Test = Testedwebsite()
7 driver = Test.driver
8
9 #click the Login button in top right
10 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
11 driver.execute_script("arguments[0].click();", log_in_header_button)
12
13 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")
14 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")
15 log_in_button = Test.element_finder_by_path("//button[@id='loginBtn']")
16
17 #put_email
18 log_in_email.send_keys("moha@gmail.com")
19 #put_password
20 log_in_pass.send_keys("123123")
21
22 #click login button
23 driver.execute_script("arguments[0].click();", log_in_button)
24
25
26
27 #click menu
28 user_menu = Test.element_finder_by_path("//img[@src='../../../../icon/user-man.png']")
29 Test.actions.move_to_element(user_menu).perform()
30 #click user profile
31 user_profile_in_user_menu = Test.element_finder_by_path("//a[normalize-space()='My Profile']")
32 user_profile_in_user_menu.click()
33 time.sleep(4)
34 #Update password with diffreant values for new password and repeat new password
35 change_pass = Test.element_finder_by_path("//a[normalize-space()='Change password']")
36 change_pass.click()
37 time.sleep(2)
38 curr_pass = Test.element_finder_by_path("//input[@id='currentpassword']")
39 curr_pass.send_keys("123123")
40 time.sleep(1)
41 new_pass = Test.element_finder_by_path("//input[@id='newpassword']")
42 new_pass.send_keys("1333333")
43 time.sleep(1)
44 rep_pass = Test.element_finder_by_path("//input[@id='newpassword2']")
45 rep_pass.send_keys("122222")
46 time.sleep(2)
47 #click on save changes button
48 save_changes = Test.element_finder_by_path("//button[@id='updateUserPassword']")
49 save_changes.click()
50 time.sleep(10)
```

FluentVison



My Profile Settings

|                 |                     |
|-----------------|---------------------|
| General         | Current password    |
| Change password | *****               |
| My Grades       | New password        |
|                 | *****               |
|                 | Repeat new password |
|                 | *****               |

**Save changes**

Repeat password does not match the new password.

## TC\_017- User profile\_7

Modify the password when one of the fields is blank:

```
Test16.py          Test17.py ×  
1  #User_profile_7  
2 > import ...  
5  
6 #Test-17  
7  
8 Test = Testedwebsite()  
9 driver = Test.driver  
10  
11 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")  
12 driver.execute_script("arguments[0].click()", log_in_header_button)  
13  
14 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")  
15 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")  
16 log_in_button = Test.element_finder_by_path("//button[@id='loginBtn']")  
17  
18 #put_email  
19 log_in_email.send_keys("moha@gmail.com")  
20  
21 #put_password  
22 log_in_pass.send_keys("123123")  
23  
24 #click login button  
25 driver.execute_script("arguments[0].click()", log_in_button)  
26  
  
Test16.py          Test17.py ×  
27 #click menu  
28 user_menu = Test.element_finder_by_path("//img[@src='../../../../icon/user-man.png']")  
29 Test.actions.move_to_element(user_menu).perform()  
30  
31 #click user profile  
32 user_profile_in_user_menu = Test.element_finder_by_path("//a[normalize-space()='My Profile']")  
33 user_profile_in_user_menu.click()  
34  
35 #Modify the password when one of the fields is blank  
36 change_pass = Test.element_finder_by_path("//a[normalize-space()='Change password']")  
37 change_pass.click()  
38 time.sleep(2)  
39 curr_pass = Test.element_finder_by_path("//input[@id='currentpassword']")  
40 curr_pass.send_keys("123123")  
41 time.sleep(1)  
42 new_pass = Test.element_finder_by_path("//input[@id='newpassword']")  
43 new_pass.send_keys("123456")  
44 time.sleep(1)  
45  
46 #click on save changes button  
47 save_changes = Test.element_finder_by_path("//button[@id='updateUserPassword']")  
48 save_changes.click()  
49 time.sleep(10)
```

FluentVison



### My Profile Settings

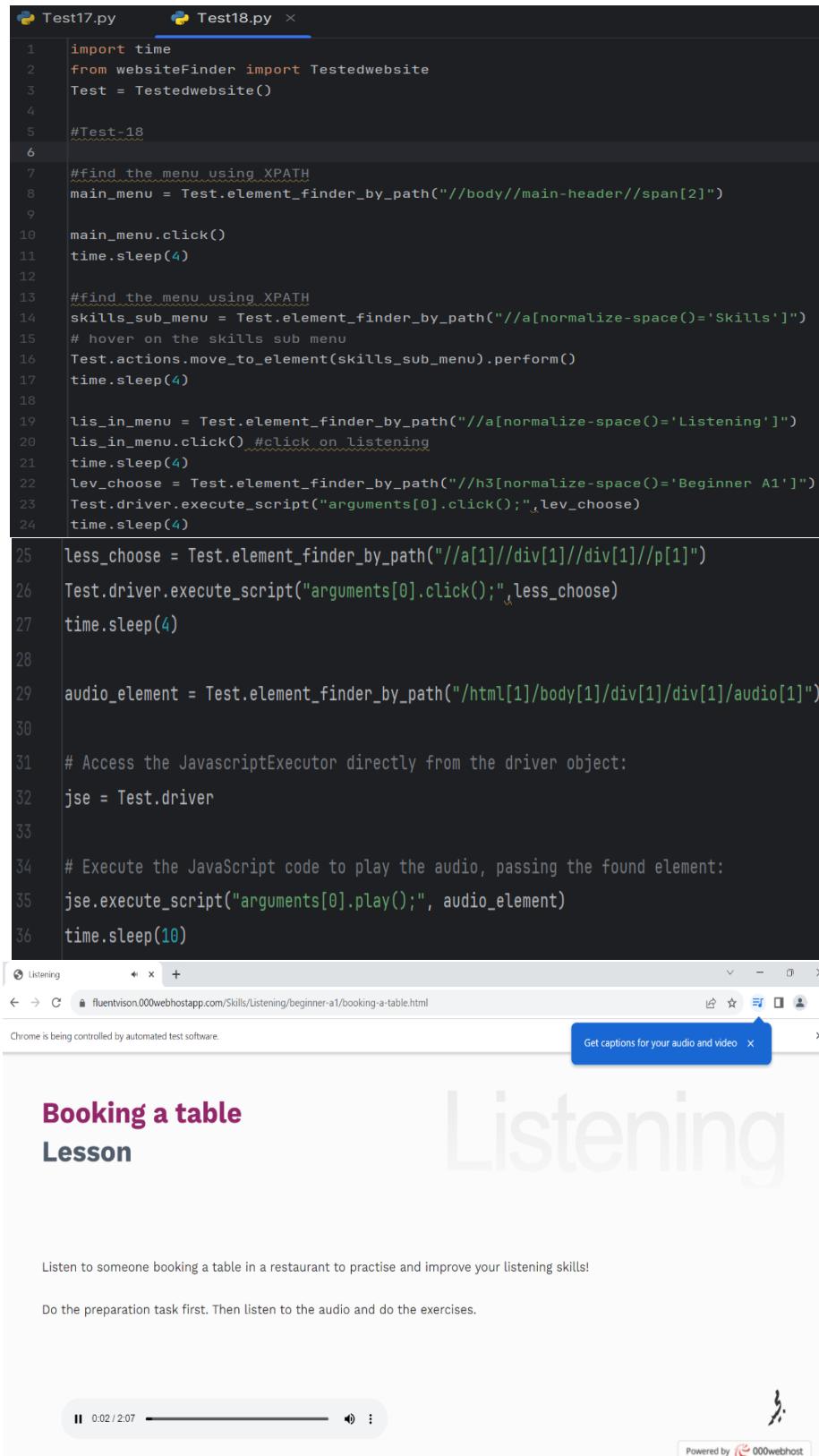
|                 |   |
|-----------------|---|
| General         | Current password                            |
| Change password | <input type="text" value="*****"/>          |
| My Grades       | New password                                |
|                 | <input type="text" value="*****"/>          |
|                 | Repeat new password                         |
|                 | <input type="text"/>                        |
|                 | <input type="button" value="Save changes"/> |

Please fill in all fields.

Powered by 000webhost

## TC\_018- Listening\_Audio

Starting the audio:



The screenshot shows a browser window with the title 'Listening'. The URL in the address bar is 'fluentvision.000webhostapp.com/Skills/Listening/beginner-a1/booking-a-table.html'. A status bar at the bottom of the browser says 'Chrome is being controlled by automated test software.' There is a blue button labeled 'Get captions for your audio and video'.

The main content of the page is titled 'Booking a table Lesson'. Below the title, it says 'Listen to someone booking a table in a restaurant to practise and improve your listening skills!'. It also says 'Do the preparation task first. Then listen to the audio and do the exercises.' At the bottom, there is a media player control bar with a play/pause button, a progress bar showing '002 / 207', and other standard media controls.

```
Test17.py      Test18.py ×
1 import time
2 from websiteFinder import Testedwebsite
3 Test = Testedwebsite()
4
5 #Test-18
6
7 #find the menu using XPATH
8 main_menu = Test.element_finder_by_path("//body//main-header//span[2]")
9
10 main_menu.click()
11 time.sleep(4)
12
13 #find the menu using XPATH
14 skills_sub_menu = Test.element_finder_by_path("//a[normalize-space()='Skills']")
15 # hover on the skills sub menu
16 Test.actions.move_to_element(skills_sub_menu).perform()
17 time.sleep(4)
18
19 lis_in_menu = Test.element_finder_by_path("//a[normalize-space()='Listening']")
20 lis_in_menu.click() #click on listening
21 time.sleep(4)
22 lev_choose = Test.element_finder_by_path("//h3[normalize-space()='Beginner A1']")
23 Test.driver.execute_script("arguments[0].click();",lev_choose)
24 time.sleep(4)
25 less_choose = Test.element_finder_by_path("//a[1]//div[1]//div[1]//p[1]")
26 Test.driver.execute_script("arguments[0].click();",less_choose)
27 time.sleep(4)
28
29 audio_element = Test.element_finder_by_path("//html[1]/body[1]/div[1]/div[1]/audio[1]")
30
31 # Access the JavascriptExecutor directly from the driver object:
32 jse = Test.driver
33
34 # Execute the JavaScript code to play the audio, passing the found element:
35 jse.execute_script("arguments[0].play()", audio_element)
36 time.sleep(10)
```

## TC\_019- Speaking\_Video

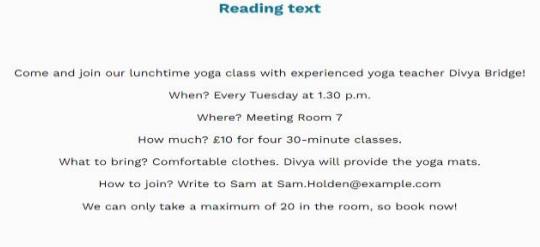
Starting the Video:



```
Test17.py Test18.py Test19.py
1 #Speaking_Video
2 > import ...
3
4 Test = Testedwebsite()
5
6 #Test-19
7
8 #find the menu using XPATH
9 main_menu = Test.element_finder_by_path("//body//main-header//span[2]")
10
11 # hover on the menu
12 Test.actions.move_to_element(main_menu).perform()
13
14 #find the menu using XPATH
15 skills_sub_menu = Test.element_finder_by_path("//a[normalize-space()='Skills']")
16
17 # hover on the skills sub menu
18 Test.actions.move_to_element(skills_sub_menu).perform()
19
20 time.sleep(4)
21 spk_in_menu = Test.element_finder_by_path("//a[normalize-space()='Speaking']")
22 spk_in_menu.click()
23 time.sleep(4)
24
25 lev_choose = Test.element_finder_by_path("//h3[normalize-space()='A1 speaking']")
26 Test.driver.execute_script("arguments[0].click();",lev_choose)
27 time.sleep(4)
28 less_choose = Test.element_finder_by_path("//h3[normalize-space()='Checking understanding']")
29 Test.driver.execute_script("arguments[0].click();",less_choose)
30 time.sleep(4)
31 video_start = Test.element_finder_by_path("//div[@class='video']//video")
32 video_start.click()
```

## TC\_020- Reading\_Text

Checking if the text size, color, format is in correct form:



```
Test19.py Test20.py
1 #Reading_Text
2 > import ...
3 Test = Testedwebsite()
4 #Test-20
5
6 #find the menu using XPATH
7 main_menu = Test.element_finder_by_path("//body//main-header//span[2]")
8
9 main_menu.click()
10
11 #find the menu using XPATH
12 skills_sub_menu = Test.element_finder_by_path("//a[normalize-space()='Skills']")
13
14 # hover on the skills sub menu
15 Test.actions.move_to_element(skills_sub_menu).perform()
16
17 read_in_menu = Test.element_finder_by_path("//a[normalize-space()='Reading']")
18 read_in_menu.click()
19 time.sleep(4)
20 lev_choose = Test.element_finder_by_path("//h3[normalize-space()='A1 reading']")
21 Test.driver.execute_script("arguments[0].click();",lev_choose)
22 time.sleep(4)
23 less_choose = Test.element_finder_by_path("//h3[normalize-space()='A poster at work']")
24 Test.driver.execute_script("arguments[0].click();",less_choose)
```

## TC\_021- Quiz\_1

check if the website will evaluate the user answers and determine of it's true, false / also displaying the number, percentage of the user correct answers:

The screenshot shows a browser window with several tabs open at the top, including 'Test20.py' (active), 'Test21.py', 'Test22.py', 'Test23.py', and 'Test24.py'. Below the tabs, there is a block of Python code for 'Test21.py' which interacts with a website using Selenium. The code includes imports for time, websiteFinder, and Testedwebsite, and performs various actions like hovering over menu items and clicking on specific links.

The main content area displays a 'Multiple Choice Quiz' with six questions:

- When is the table booked for?
  - Tonight
  - Tomorrow morning (selected)
  - Tomorrow night
- When the woman says 'about eight, eight thirty', what does she mean?
  - At eight o'clock
  - At half past eight (selected)
  - Between eight o'clock and half past eight
- What time is the first booking?
  - 7.30 (selected)
  - 8.30
  - 9.00
- Where will the table be now?
  - By the door
  - Close to the kitchen (selected)
  - In the corner
- What time is the new booking?
  - 6.00
  - 7.30 (selected)
  - 8.00

At the bottom of the quiz interface, the results are displayed:  
Correct Answers: 2 / 6  
Percentage: 33.33%  
[Finish](#)   [Restart](#)

## TC\_022- Quiz\_2

check if the website will evaluate the user answers and determine if it's true, false / also displaying the number, percentage of the user correct answers:

```
 1 #Quiz_2
 2 > import ...
 3 #Test_22
 4 Test = Testedwebsite()
 5 #find the menu using XPATH
 6 main_menu = Test.element_finder_by_path("//body//main-header//span[2]")
 7 # hover on the menu
 8 Test.actions.move_to_element(main_menu).perform()
 9 #find the menu using XPATH
10 skills_sub_menu = Test.element_finder_by_path("//a[normalize-space()='Skills']")
11 # hover on the skills sub menu
12 Test.actions.move_to_element(skills_sub_menu).perform()
13
14 spk_in_menu = Test.element_finder_by_path("//a[normalize-space()='Reading']")
15 spk_in_menu.click()
16
17 lev_choose = Test.element_finder_by_path("//h3[normalize-space()='A1 reading']")
18 Test.driver.execute_script("arguments[0].click();",lev_choose)
19 time.sleep(5)
20 less_choose = Test.element_finder_by_path("//h3[normalize-space()='A poster at work']")
21 Test.driver.execute_script("arguments[0].click();",less_choose)
22 time.sleep(5)
23 finish_btn = Test.element_finder_by_path("//button[@id='finishBtn']")
24 Test.driver.execute_script("arguments[0].click();",finish_btn)
```

### True/False Quiz

- |   |             |              |
|---|-------------|--------------|
| 1. Sam Holden is the yoga teacher.                            | <b>True</b> | <b>False</b> |
| 2. The yoga class is once a week.                             | <b>True</b> | <b>False</b> |
| 3. The class is at lunchtime in room 7.                       | <b>True</b> | <b>False</b> |
| 4. You need to bring yoga mats to the class.                  | <b>True</b> | <b>False</b> |
| 5. The class can't take 22 people.                            | <b>True</b> | <b>False</b> |
| 6. You need to call Sam Holden if you want to join the class. | <b>True</b> | <b>False</b> |

**Finish**

### True/False Quiz

- |   |             |              |
|---|-------------|--------------|
| 1. Sam Holden is the yoga teacher.                            | <b>True</b> | <b>False</b> |
| 2. The yoga class is once a week.                             | <b>True</b> | <b>False</b> |
| 3. The class is at lunchtime in room 7.                       | <b>True</b> | <b>False</b> |
| 4. You need to bring yoga mats to the class.                  | <b>True</b> | <b>False</b> |
| 5. The class can't take 22 people.                            | <b>True</b> | <b>False</b> |
| 6. You need to call Sam Holden if you want to join the class. | <b>True</b> | <b>False</b> |

**Correct Answers: 4 / 6**  
**Percentage: 66.67%**

## TC\_023- Quiz\_3

check if the website will evaluate the user answers and determine if it's true, false / also displaying the number, percentage of the user correct answers:

The screenshot shows a Python code editor with several files open (Test21.py, Test22.py, Test23.py, Test24.py, Test25.py, Test26.py) and a browser-based quiz interface.

**Python Script Content:**

```
1 #Quiz_3
2
3 > import ...
4
5
6
7 #Test-23
8
9 Test = Testedwebsite()
10
11 #find the menu using XPATH
12 main_menu = Test.element_finder_by_path("//body//main-header//span[2]")
13
14 # hover on the menu
15 Test.actions.move_to_element(main_menu).perform()
16
17 grammer_in_menu = Test.element_finder_by_path("//a[normalize-space()='Grammar']")
18 grammer_in_menu.click()
19
20 less_choose = Test.element_finder_by_path("//div[@class='grammar-list-b2-content grid-2']//a[1]//p[1]")
21 Test.driver.execute_script("arguments[0].click();", less_choose)
22
23 finish_btn = Test.element_finder_by_path("//button[@id='finishBtn']")
24 Test.driver.execute_script("arguments[0].click();", finish_btn)
25
26 time.sleep(10)
```

**Quiz Interface:**

8. I can't believe you \_\_\_\_\_ your phone at home again! { left }

Enter your answer

9. She suggested \_\_\_\_\_ a break after working for several hours. { take }

Enter your answer

10. If it \_\_\_\_\_ tomorrow, we'll have the picnic indoors. { rain }

Enter your answer

**Statistics:**

Correct Answers: 0 / 10  
Percentage: 0.00%

**Buttons:**

Finish    Restart

**Answered Questions:**

8. I can't believe you \_\_\_\_\_ your phone at home again! { left }  
left

9. She suggested \_\_\_\_\_ a break after working for several hours. { take }  
take

10. If it \_\_\_\_\_ tomorrow, we'll have the picnic indoors. { rain }  
rained

**Final Statistics:**

Correct Answers: 1 / 10  
Percentage: 10.00%

**Buttons:**

Finish    Restart

## TC\_024- Quiz\_4

check if the user can click or choose the answer after he press the finish button:

The screenshot shows a code editor with a Python script titled 'Test24.py' and a web-based quiz interface.

**Code Editor (Test24.py):**

```
1 #Quiz_4
2 import time
3 from websiteFinder import Testedsite
4 from selenium.webdriver.support.select import Select
5 Test = Testedsite()
6 #Test-24
7
8 #Find the menu using XPATH
9 main_menu = Test.element_finder_by_path("//body//main-header//span[2]")
10
11 # hover on the menu
12 Test.actions.move_to_element(main_menu).perform()
13
14 #find the menu using XPATH
15 skills_sub_menu = Test.element_finder_by_path("//a[normalize-space()='Skills']")
16
17 # hover on the skills sub menu
18 Test.actions.move_to_element(skills_sub_menu).perform()
19
20 lis_in_menu = Test.element_finder_by_path("//a[normalize-space()='Listening']")
21 lis_in_menu.click() #click on listening
22
23 lev_choose = Test.element_finder_by_path("//h3[normalize-space()='Beginner A1']")
24 Test.driver.execute_script("arguments[0].click()",lev_choose)
25
26 less_choose = Test.element_finder_by_path("//a[1]//div[1]//div[1]//p[1]")
27 Test.driver.execute_script("arguments[0].click()",less_choose)
28
29 q1 = Test.element_finder_by_path("//div[@id='question0']//label[1]")
30 Test.driver.execute_script("arguments[0].click()",q1)
31
32 q2 = Test.element_finder_by_path("//div[@id='question1']//label[1]")
33 Test.driver.execute_script("arguments[0].click()",q2)
34
35 q3 = Test.element_finder_by_path("//div[@id='question2']//label[1]")
36 Test.driver.execute_script("arguments[0].click()",q3)
37
38 q4 = Test.element_finder_by_path("//div[@id='question3']//label[1]")
39 Test.driver.execute_script("arguments[0].click()",q4)
40
41 q5 = Test.element_finder_by_path("//div[@id='question4']//label[1]")
42 Test.driver.execute_script("arguments[0].click()",q5)
43
44 q6 = Test.element_finder_by_path("//div[@id='questions']//label[1]")
45 Test.driver.execute_script("arguments[0].click()",q6)
46 time.sleep(30)
47 finish_btn = Test.element_finder_by_path("//button[@id='finishBtn']")
48 Test.driver.execute_script("arguments[0].click()",finish_btn)
```

**Quiz Interface:**

Question 5: Where will the table be now?

- By the door
- Close to the kitchen
- In the corner

Question 6: What time is the new booking?

- 6.00
- 7.30
- 8.00

Correct Answers: 1 / 6  
Percentage: 16.67%

Buttons: Finish (red), Restart (green)

## TC\_025- Quiz\_5

check if the website allows the user to retake the test after he press restart button:

```
Test24.py Test25.py x Test26.py
1 #Quiz_5
2 > import ...
3 #Test-25
4 Test = Testedwebsite()
5
6 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
7 Test.driver.execute_script("arguments[0].click();", log_in_header_button)
8
9 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")
10 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")
11 log_in_button = Test.element_finder_by_path("//button[@id='loginBtn']")
12 log_in_email.send_keys("majeed@gmail.com")
13 log_in_pass.send_keys("123123")
14 Test.driver.execute_script("arguments[0].click();", log_in_button)
15
16 #find the menu using XPATH
17 main_menu = Test.element_finder_by_path("//body//main-header//span[2]")
18 main_menu.click()
19 # hover on the menu
20 #Test.actions.move_to_element(main_menu).perform()
21 #find the menu using XPATH
22 skills_sub_menu = Test.element_finder_by_path("//a[normalize-space()='Skills']")
23 # hover on the skills sub menu
24 Test.actions.move_to_element(skills_sub_menu).perform()

Test24.py Test25.py x Test26.py
25
26 lis_in_menu = Test.element_finder_by_path("//a[normalize-space()='Listening']")
27 lis_in_menu.click() #click on listening
28
29 lev_choose = Test.element_finder_by_path("//h3[normalize-space()='Beginner A1']")
30 Test.driver.execute_script("arguments[0].click();", lev_choose)
31
32 less_choose = Test.element_finder_by_path("//a[1]//div[1]//div[1]//p[1]")
33 Test.driver.execute_script("arguments[0].click();", less_choose)
34
35 q1 = Test.element_finder_by_path("//div[@id='question0']//label[1]")
36 Test.driver.execute_script("arguments[0].click();", q1)
37
38 q2 = Test.element_finder_by_path("//div[@id='question1']//label[1]")
39 Test.driver.execute_script("arguments[0].click();", q2)
40
41 q3 = Test.element_finder_by_path("//div[@id='question2']//label[1]")
42 Test.driver.execute_script("arguments[0].click();", q3)
43
44 q4 = Test.element_finder_by_path("//div[@id='question3']//label[1]")
45 Test.driver.execute_script("arguments[0].click();", q4)
46 q5 = Test.element_finder_by_path("//div[@id='question4']//label[1]")
47 Test.driver.execute_script("arguments[0].click();", q5)

48
49 q6 = Test.element_finder_by_path("//div[@id='question5']//label[1]")
50 Test.driver.execute_script("arguments[0].click();", q6)
51 time.sleep(5)
52
53 finish_btn = Test.element_finder_by_path("//button[@id='finishBtn']")
54 Test.driver.execute_script("arguments[0].click();", finish_btn)
55 time.sleep(8)
56
57 rest_btn = Test.element_finder_by_path("//button[@id='restartBtn']")
58 Test.driver.execute_script("arguments[0].click();", rest_btn)
59 time.sleep(10)
```

Jamiei  
 Jamie  
 Janie

5. Where will the table be now?  
 By the door  
 Close to the kitchen  
 In the corner

6. What time is the new booking?  
 6.00  
 7.30  
 8.00

Finish

5. Where will the table be now?

- By the door
- Close to the kitchen
- In the corner

6. What time is the new booking?

- 6.00
- 7.30
- 8.00

Correct Answers: 1 / 6

Percentage: 16.67%

Finish

Restart

- Jamiei
- Jamie
- Janie

5. Where will the table be now?

- By the door
- Close to the kitchen
- In the corner

6. What time is the new booking?

- 6.00
- 7.30
- 8.00

Finish

## TC\_026- User Profile\_4, Quiz\_6c

Check if the website adds the user test grade on the Grades section in profile:

```
Test24.py Test25.py Test26.py ×
1 #User_Profile_4_Quiz_6
2 > import ...
3 #Test-26
4
5
6 Test = Testedwebsite()
7
8 log_in_header_button = Test.element_finder_by_path("//button[@id='login-btn-popup']")
9 Test.driver.execute_script("arguments[0].click();", log_in_header_button)
10
11 log_in_email = Test.element_finder_by_path("//input[@id='loginEmail']")
12 log_in_pass = Test.element_finder_by_path("//input[@id='loginPass']")
13 log_in_button = Test.element_finder_by_path("//button[@id='loginBtn']")
14
15 log_in_email.send_keys("majeed@gmail.com")
16 log_in_pass.send_keys("123123")
17 Test.driver.execute_script("arguments[0].click();", log_in_button)
18 #find the menu using XPATH
19 time.sleep(1)
20 #find the menu using XPATH
21 main_menu = Test.element_finder_by_path("//body//main-header//span[2]")
22 main_menu.click()
23
24
25 grammar_in_menu = Test.element_finder_by_path("//a[normalize-space()='Grammar']")
26 grammar_in_menu.click()
27
28 less_choose = Test.element_finder_by_path("//div[@class='grammar-list-b2-content grid-2']//a[1]//p[1]")
29 Test.driver.execute_script("arguments[0].click();", less_choose)
30
31 finish_btn = Test.element_finder_by_path("//button[@id='finishBtn']")
32 Test.driver.execute_script("arguments[0].click();", finish_btn)
33
34 time.sleep(2)
35 user_menu = Test.element_finder_by_path("//img[@src='.../icon/user-man.png']")
36 Test.actions.move_to_element(user_menu).perform()
37
38 user_profile_in_user_menu = Test.element_finder_by_path("//a[normalize-space()='My Profile']")
39 user_profile_in_user_menu.click()
40
41 time.sleep(1)
42 grades = Test.element_finder_by_path("//a[normalize-space()='My Grades']")
43 Test.driver.execute_script("arguments[0].click();", grades)
44 time.sleep(10)
```

FluentVison



### My Profile Settings

| General          | Level | Skills    | Grades |
|------------------|-------|-----------|--------|
| Change password  | A1    | Listening | 50     |
| <b>My Grades</b> | A1    | Reading   | 50     |
|                  | B2    | Grammer   | 0      |
|                  | A1    | Writing   | 17     |
|                  | A1    | Speaking  | 33     |

Powered by 000webhost

## TC\_027- Vocabulary\_Game

Check if the Play now button in home page or Vocabulary page is working:

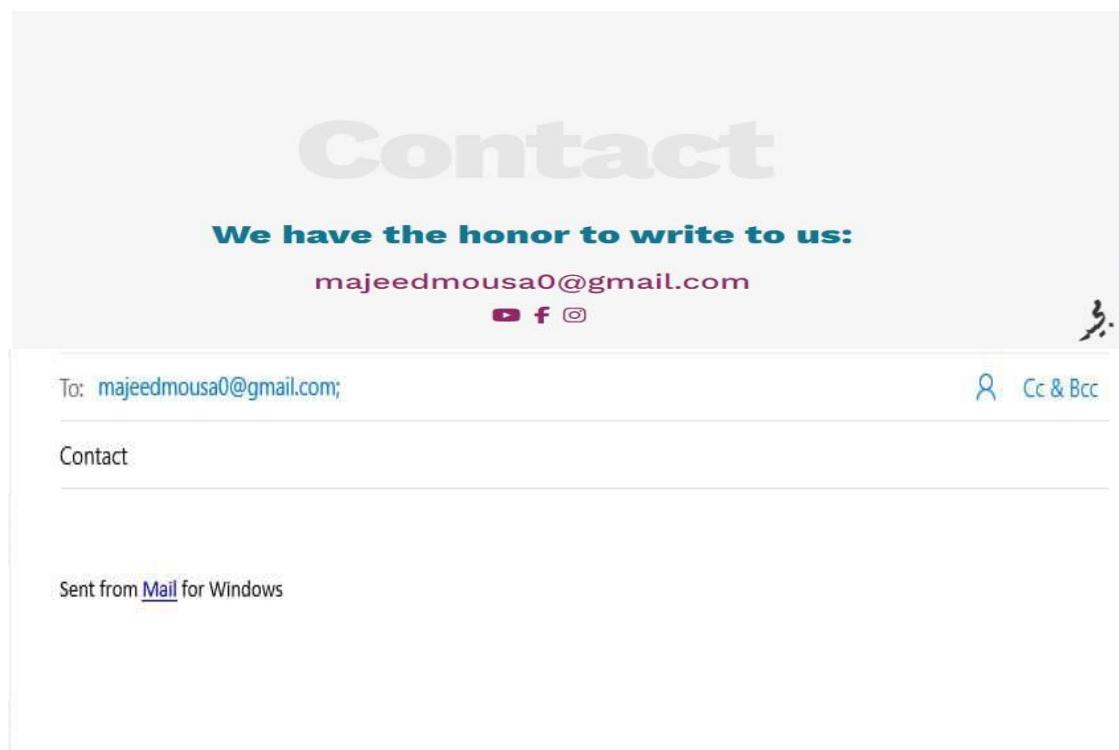
```
1 import time
2 from MainWeb import Testedwebsite
3
4 Test = Testedwebsite()
5
6
7 #put the cursor on the menu
8 menu = Test.element_finder_by_path("//div[@class='links']/div[@class='icon']")
9 Test.actions.move_to_element(menu).perform()
10
11
12 #click on the vocabulary button
13 vocab = Test.element_finder_by_path("//a[normalize-space()='Vocabulary']")
14 vocab.click()
15
16 # click on the game button
17 button = Test.element_finder_by_path("//a[@class='game-link main-button']")
18 Test.driver.execute_script("arguments[0].click()",button)
19
20 time.sleep(5)
21
```

(8% Loaded)

## TC\_028- Contact

Check if the user can contact us through the email provided:

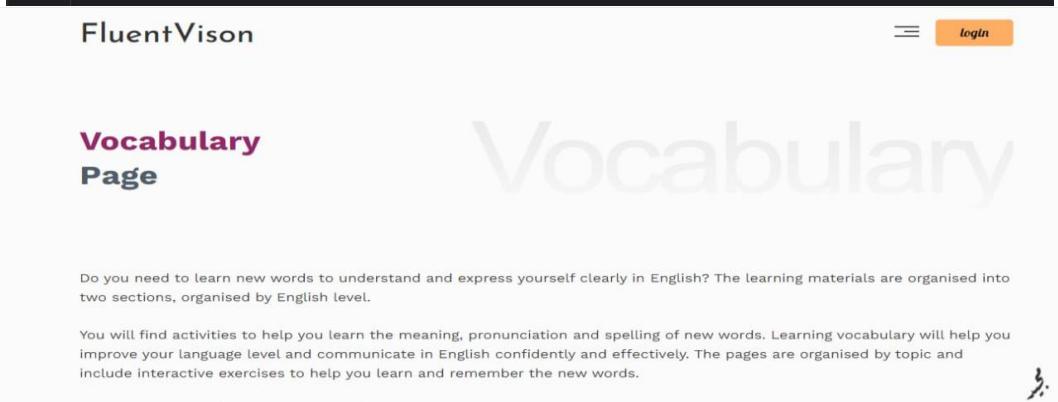
```
1 from MainWeb import Testedwebsite
2 import time
3
4 #Test-28
5
6 Test = Testedwebsite()
7
8 #put the cursor on the menu
9 menu = Test.element_finder_by_path("//div[@class='links']//div[@class='icon']")
10 Test.actions.move_to_element(menu).perform()
11
12 #click on the contact button
13 contact = Test.element_finder_by_path("//a[normalize-space()='Contact']")
14 contact.click()
15
16 #click on email : majeedmousa0@gmail.com
17 button = Test.element_finder_by_path("//a[@class='email']")
18 Test.driver.execute_script("arguments[0].click();",button)
19
20 time.sleep(5)
21
22
```



## TC\_029- Vocabulary button

Check if the website will direct the user to the Vocabulary page:

```
1 from MainWeb import Testedwebsite
2 import time
3
4 #Test-29
5
6 Test = Testedwebsite()
7
8 #Click on the button "Start with important words"
9 Sbutton = Test.element_finder_by_path("//a[normalize-space()='Start With important words']")
10 Test.driver.execute_script("arguments[0].click()", Sbutton)
11
12
13 time.sleep(5)
14 |
```



FluentVison

Vocabulary  
Page

Do you need to learn new words to understand and express yourself clearly in English? The learning materials are organised into two sections, organised by English level.

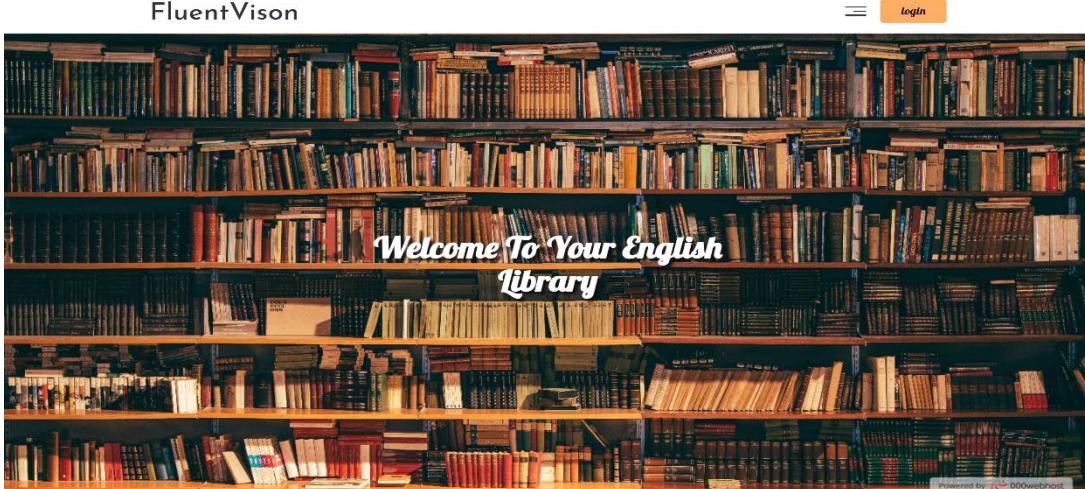
You will find activities to help you learn the meaning, pronunciation and spelling of new words. Learning vocabulary will help you improve your language level and communicate in English confidently and effectively. The pages are organised by topic and include interactive exercises to help you learn and remember the new words.

login

## TC\_030- logo

Check if the user pressed the logo button will be returned to the home page:

```
1 from MainWeb import Testedwebsite
2 import time
3
4 #Test-31
5
6 Test = Testedwebsite()
7 #go To Any page
8 Button = Test.element_finder_by_path("//div[@class='grammer-content grid-2']/div[1]/a[1]")
9 Test.driver.execute_script("arguments[0].click()", Button)
10 time.sleep(2)
11 # Click on the Logo to test if will return home page
12 Logo = Test.element_finder_by_path("//a[normalize-space()='FluentVison']")
13 Logo.click()
14 time.sleep(3)
```



FluentVison

Welcome To Your English  
Library

login

## TC\_031- Read more

Check if the website will direct the user to the Correct page when he press read more (this test case is applied for all read more button in home page):

```
1 from MainWeb import Testedwebsite
2 import time
3
4 #Test-30
5
6 Test = Testedwebsite()
7
8 #Click on the read more buttons in the main page
9 #
10 1 usage
11 def FirstB():
12     # Find and Click on the first read more button
13     FB = Test.element_finder_by_path("//div[@class='grammer-content grid-2']//div[1]/a[1]")
14     Test.driver.execute_script("arguments[0].click()",FB)
15     time.sleep(4)
16     #find the menu and return to the home
17     menu = Test.element_finder_by_path("//body//main-header//span[2]")
18     Test.actions.move_to_element(menu).perform()
19     home = Test.element_finder_by_path("//a[normalize-space()='home']")
20     Test.driver.execute_script("arguments[0].click()", home)
21
22     time.sleep(4)
23
24 1 usage
25 def SecondB():
26     #find and click the second read more button
27     SB = Test.element_finder_by_path("//div[@id='grammer']//div[2]/a[1]")
28     Test.driver.execute_script("arguments[0].click()", SB)
29     time.sleep(4)
30     #find the menu and return to the home
31     menu2 = Test.element_finder_by_path("//div[@class='icon']")
32     Test.actions.move_to_element(menu2).perform()
33     home2= Test.element_finder_by_path("//a[normalize-space()='home']")
34     Test.driver.execute_script("arguments[0].click()", home2)
35     time.sleep(4)
36
37 def ThirdB():
38     #find and click the third read more button
39     TB = Test.element_finder_by_path("//div[@id='grammer']//div[3]/a[1]")
40     Test.driver.execute_script("arguments[0].click()", TB)
41     time.sleep(4)
42
43 FirstB()
44 SecondB()
45 ThirdB()
```

البِرَاءَةُ



## Could have, should have, would have

These past modal verbs are all used hypothetically, to talk about things that didn't really happen in the past.

### Could have + past participle

1: Could have + past participle means that something was possible in the past, or you had the ability to do something in the past, but that you didn't do it. (See also **modals of ability**.)

FluentVison

≡ login

### All Conditional



البِرَاءَةُ



## Articles

Do you know how to use a, an and the?

Look at these examples to see how articles are used.

- She's a doctor.
- I need an umbrella.
- Have you heard the news?
- I don't like spiders.



