



## **Software Engineering**

### **IT314**

## **Fake News Faux Real**

### **GUI TESTING:**

- All the buttons are working correctly for every page.
- All pages are compatible with different browsers.
- For any page, the font size and indentation is consistent.
- The text displayed on the page is readable.

#### **1. Home Page**

- Layout and design of the webpage is good and the content is readable.
- Navigation through the page is easy.

- Navigation menu is working correctly, by clicking on each link the user is directed to the intended page.
- "Get Started" button, ensuring that it redirects the user to the appropriate page.
- Text under the features is aligned properly and symbols used are in accordance.
- Privacy policy statement in the "Privacy is our top priority" feature is correct and up-to-date.
- Web page is responsive, and displays correct data on various screen sizes and resolutions.
- Test the compatibility of the web page across different browsers and platforms, ensuring that it works correctly on each one.

## 2. Signin/Signup

- The page title is "Sign Up".
- The "Sign Up" tab is active with the inactive "Sign In" tab.
- There is an icon displayed on the page.
- On clicking sign up the signup page is displayed.
- With appropriate placeholders, there are input fields for "Name", "Email", "Password", and "Confirm Password".
- There is a "Sign Up" and a "Back" button.
- By clicking on the back button we are redirected to the previous page.
- An error message is displayed on entering an invalid email address in the "Email" field.
- No error message is displayed on entering a valid email address in the "Email" field.
- For too short or too long passwords in the "Password" field, an error message is displayed.
- For a valid password in the "Password" field, no error message is displayed.
- If the password doesn't match the "Confirm Password" field, an error message is displayed.
- For a password that matches the "Confirm Password" field, no error message is displayed.

## 3. Login

- Login form is displayed on the page.
- It includes two input fields for email and password, respectively.
- Email input field accepts valid email addresses only and if the email address is not valid displays an error message for invalid email addresses.
- Password input field accepts passwords that are between 6 to 25 characters long.

- "Log In" button is present and is clickable as well.
- On clicking the "Log In" button without entering any details or with invalid login details displays appropriate error messages.
- By clicking the "Log In" button with valid login details logs the user in and redirects to the home page.
- The image and icon on the page are displayed correctly.
- The "Back" button is present and by clicking it redirects the user to the previous page.
- The page is responsive to different screen sizes and on different browsers.

#### **4. Result**

- The page loads without any errors and all the elements are visible and functional, including the navigation bar, the header, the footer, and the main section containing the result.
- By clicking on the links in the navigation bar redirects to the correct page.
- The elements adjust accordingly by resizing the browser window ensures the responsiveness of the page.
- The functionality of all interactive elements on the page, such as the collapse button in the navigation bar or buttons that present it is working properly.
- The loading speed of the page is good enough.

#### **5. News\_detection**

- Navbar is displayed correctly and the links are working as expected.
- "Detect News" section is displaying correctly with the correct form fields and buttons.
- Form gives correct responses for different input methods (text and URL) and the newsletter options are displayed or hidden based on the selected input method.
- Form is submitted correctly for both the input methods.
- Cross-browser testing: Check the webpage on different web browsers to ensure that it is displaying correctly and functioning properly on all browsers.

## 6. About us

- The web page loads without any errors.
- The title of the web page is "Fake News Detection".
- The text in the "About Us" section is displayed correctly.
- The font Family of the text on the page is "Roboto".
- The "Work with heart" feature box, "Reliable services" feature box and "Great support" feature box are correctly displayed, including the text and icon.
- At the bottom of the page, the footer is displayed correctly.
- The copyright notice in the footer is correctly written and the current year is displayed.
- The links to the CSS files and JavaScript files are functioning properly as required.

## 7. FAQ

- Links on the navigation bar are working properly.
- Accordion is used to display the frequently asked questions (FAQs). It works correctly, and the questions and answers are displayed and hidden as expected.
- Logout button works correctly.
- Title of the page is displayed correctly, and it is easy to read.
- The content of the page, such as the FAQ questions and answers, is displayed correctly and easy to read.

## 8. Contact us

- Navbar is displayed properly and all the navigation links are working correctly.
- "Contact Us" page is displayed properly and the form is displayed correctly.
- Form has four input fields: name, email, subject, and message and all the input fields have proper labels and placeholders.
- Email field only accepts valid email addresses.
- Clicking the "Submit" button sends the form data to the email specified in the "action" attribute of the form.

## 9. Forgot Password

- "Password Recovery" header is displayed on the page.
- Email input field and "Send" button are present on the page.
- "Send" button triggers a server request to the correct URL.
- Entering a valid email address in the input field and clicking the "Send" button triggers a password recovery email to the correct email address.
- Entering an invalid email address in the input field and clicking the "Send" button displays an error message that email is not registered.
- "Back" button redirects the user to the login page.

## GUI Testing using Selenium:

### Test1\_basic:

### Code:

```
1 # Generated by Selenium IDE
2 import pytest
3 import time
4 import json
5 from selenium import webdriver
6 from selenium.webdriver.common.by import By
7 from selenium.webdriver.common.action_chains import ActionChains
8 from selenium.webdriver.support import expected_conditions
9 from selenium.webdriver.support.wait import WebDriverWait
10 from selenium.webdriver.common.keys import Keys
11 from selenium.webdriver.common.desired_capabilities import DesiredCapabilities
12
13 class TestTest1basic():
14     def setup_method(self, method):
15         self.driver = webdriver.Firefox()
16         self.vars = {}
17
18     def teardown_method(self, method):
19         self.driver.quit()
20
21     def test_test1basic(self):
22         # Test name: Test1_basic
23         # Step # | name | target | value
24         # 1 | open | /home |
25         self.driver.get("http://127.0.0.1:2000/home")
26         # 2 | setWindowSize | 1550x830 |
27         self.driver.set_window_size(1550, 830)
28         # 3 | click | linkText=Get Started |
29         self.driver.find_element(By.LINK_TEXT, "Get Started").click()
30         # 4 | click | css=a |
31         self.driver.find_element(By.CSS_SELECTOR, "a").click()
32         # 5 | mouseDown | css=body |
33         element = self.driver.find_element(By.CSS_SELECTOR, "body")
34         actions = ActionChains(self.driver)
35         actions.move_to_element(element).click_and_hold().perform()
36         # 6 | mouseUp | css=wrapper |
```

```

37 element = self.driver.find_element(By.CSS_SELECTOR, ".wrapper")
38 actions = ActionChains(self.driver)
39 actions.move_to_element(element).release().perform()
40 # 7 | click | css=body |
41 self.driver.find_element(By.CSS_SELECTOR, "body").click()
42 # 8 | click | id=login |
43 self.driver.find_element(By.ID, "login").click()
44 # 9 | click | css=first |
45 self.driver.find_element(By.CSS_SELECTOR, ".first").click()
46 # 10 | click | id=login |
47 self.driver.find_element(By.ID, "login").click()
48 # 11 | type | id=login | aayush@123.com
49 self.driver.find_element(By.ID, "login").send_keys("aayush@123.com")
50 # 12 | click | id=password |
51 self.driver.find_element(By.ID, "password").click()
52 # 13 | type | id=password | 123456
53 self.driver.find_element(By.ID, "password").send_keys("123456")
54 # 14 | click | css=fourth |
55 self.driver.find_element(By.CSS_SELECTOR, ".fourth").click()
56 # 15 | click | id=input-data |
57 self.driver.find_element(By.ID, "input-data").click()
58 # 16 | type | id=input-data | (CNN) On Veterans Day we recognize and honor the sacrifices our service members and their families make for our country. We owe our
59 self.driver.find_element(By.ID, "input-data").send_keys("(CNN) On Veterans Day we recognize and honor the sacrifices our service members and their families make
60 # 17 | click | css=btn |
61 self.driver.find_element(By.CSS_SELECTOR, ".btn").click()
62 # 18 | click | linkText=Logout |
63 self.driver.find_element(By.LINK_TEXT, "Logout").click()
64 # 19 | click | linkText=Get Started |
65 self.driver.find_element(By.LINK_TEXT, "Get Started").click()
66 # 20 | click | css=inactive |
67 self.driver.find_element(By.CSS_SELECTOR, ".inactive").click()
68 # 21 | click | id=login |
69 self.driver.find_element(By.ID, "login").click()
70 # 22 | type | id=login | aayush@123.com
71 self.driver.find_element(By.ID, "login").send_keys("aayush@123.com")
72 # 23 | click | id=password |

```

```

72 # 23 | click | id=password |
73 self.driver.find_element(By.ID, "password").click()
74 # 24 | type | id=password | 123456
75 self.driver.find_element(By.ID, "password").send_keys("123456")
76 # 25 | click | css=fourth |
77 self.driver.find_element(By.CSS_SELECTOR, ".fourth").click()
78 # 26 | mouseDownAt | id=input-method | -120,-156.39999389648438
79 element = self.driver.find_element(By.ID, "input-method")
80 actions = ActionChains(self.driver)
81 actions.move_to_element(element).click_and_hold().perform()
82 # 27 | mouseMoveAt | id=input-method | -120,-156.39999389648438
83 element = self.driver.find_element(By.ID, "input-method")
84 actions = ActionChains(self.driver)
85 actions.move_to_element(element).perform()
86 # 28 | mouseUpAt | id=input-method | -120,-156.39999389648438
87 element = self.driver.find_element(By.ID, "input-method")
88 actions = ActionChains(self.driver)
89 actions.move_to_element(element).release().perform()
90 # 29 | click | id=input-method |
91 self.driver.find_element(By.ID, "input-method").click()
92 # 30 | select | id=input-method | label=URL
93 dropdown = self.driver.find_element(By.ID, "input-method")
94 dropdown.find_element(By.XPATH, "//option[. = 'URL']").click()
95 # 31 | mouseDownAt | id=select-newsletter | -120,-399.6000061035156
96 element = self.driver.find_element(By.ID, "select-newsletter")
97 actions = ActionChains(self.driver)
98 actions.move_to_element(element).click_and_hold().perform()
99 # 32 | mouseMoveAt | id=select-newsletter | -120,-399.6000061035156
100 element = self.driver.find_element(By.ID, "select-newsletter")
101 actions = ActionChains(self.driver)
102 actions.move_to_element(element).perform()
103 # 33 | mouseUpAt | id=select-newsletter | -120,-399.6000061035156
104 element = self.driver.find_element(By.ID, "select-newsletter")
105 actions = ActionChains(self.driver)
106 actions.move_to_element(element).release().perform()
107 # 34 | click | id=select-newsletter |

```

```

108 self.driver.find_element(By.ID, "select-newsletter").click()
109 # 35 | select | id=select-newsletter | label=Times Of India
110 dropdown = self.driver.find_element(By.ID, "select-newsletter")
111 dropdown.find_element(By.XPATH, "//option[. = 'Times Of India']").click()
112 # 36 | click | id=input-data |
113 self.driver.find_element(By.ID, "input-data").click()
114 # 37 | type | id=input-data | https://timesofindia.indiatimes.com/us-elections-2016/Donald-Trump-elected-President-of-the-United-Sta
115 self.driver.find_element(By.ID, "input-data").send_keys("https://timesofindia.indiatimes.com/us-elections-2016/Donald-Trump-elected-
116 # 38 | click | css=btn |
117 self.driver.find_element(By.CSS_SELECTOR, ".btn").click()
118
119

```

## Result:

The screenshot shows the Selenium IDE interface with a project named 'Demo\_Test\*'. The test 'Test1\_basic\*' is selected and its steps are listed in a table:

Command	Target	Value
open	/home	
set window size		1560x830
click	linkText=Get Started	
click	css=a	
mouse down	css=body	

Below the table, there are input fields for Command, Target, Value, and Description. The Log section at the bottom shows the execution steps and their results:

- 33. mouseUpAt on id=select-newsletter with value -120, 399 6000061035156 OK 20:54:38
- 34. click on id=select-newsletter OK 20:54:40
- 35. select on id=select-newsletter with value label=Times Of India OK 20:54:40
- 36. click on id=input-data OK 20:54:40
- 37. type on id=input-data with value https://timesofindia.indiatimes.com/us-elections-2016/Donald-Trump-elected-President-of-the-United-States-in-a-massive-upset/articleshow/55326174.cms OK 20:54:40
- 38. click on css=a OK 20:54:40
- 39. click on css=body OK 20:54:40

The test 'Test1\_basic' completed successfully at 20:54:40.

## Test2\_basic:

## Code:

```
1 # Generated by Selenium IDE
2 import pytest
3 import time
4 import json
5 from selenium import webdriver
6 from selenium.webdriver.common.by import By
7 from selenium.webdriver.common.action_chains import ActionChains
8 from selenium.webdriver.support import expected_conditions
9 from selenium.webdriver.support.wait import WebDriverWait
10 from selenium.webdriver.common.keys import Keys
11 from selenium.webdriver.common.desired_capabilities import DesiredCapabilities
12
13 class TestTest2basic():
14     def setup_method(self, method):
15         self.driver = webdriver.Firefox()
16         self.vars = {}
17
18     def teardown_method(self, method):
19         self.driver.quit()
20
21     def test_test2basic(self):
22         # Test name: Test2_basic
23         # Step # | name | target | value
24         # 1 | open | / |
25         self.driver.get("http://127.0.0.1:2000/")
26         # 2 | setWindowSize | 1536x692 |
27         self.driver.set_window_size(1536, 692)
28         # 3 | click | linkText=Get Started |
29         self.driver.find_element(By.LINK_TEXT, "Get Started").click()
30         # 4 | click | css=.inactive |
31         self.driver.find_element(By.CSS_SELECTOR, ".inactive").click()
32         # 5 | click | id=login |
33         self.driver.find_element(By.ID, "login").click()
34         # 6 | type | id=login | aayushbrahmhatt03@gmail.com
35         self.driver.find_element(By.ID, "login").send_keys("aayushbrahmhatt03@gmail.com")
36         # 7 | click | id=password |
```

```

36 | # 7 | click | id=password |
37 | self.driver.find_element(By.ID, "password").click()
38 | # 8 | type | id=password | 12345678
39 | self.driver.find_element(By.ID, "password").send_keys("12345678")
40 | # 9 | click | css=.fourth |
41 | self.driver.find_element(By.CSS_SELECTOR, ".fourth").click()
42 | # 10 | click | id=input-data |
43 | self.driver.find_element(By.ID, "input-data").click()
44 | # 11 | click | id=input-method |
45 | self.driver.find_element(By.ID, "input-method").click()
46 | # 12 | select | id=input-method | label=URL
47 | dropdown = self.driver.find_element(By.ID, "input-method")
48 | dropdown.find_element(By.XPATH, "//option[. = 'URL']").click()
49 | # 13 | click | css=#input-method > option:nth-child(2) |
50 | self.driver.find_element(By.CSS_SELECTOR, "#input-method > option:nth-child(2)").click()
51 | # 14 | click | id=input-data |
52 | self.driver.find_element(By.ID, "input-data").click()
53 | # 15 | type | id=input-data | Support Us Should I Get Botox? \nsource Add To The Conversation Using Facebook Comments November 1, 2016 at 6:25 am \nI think that
54 | self.driver.find_element(By.ID, "input-data").send_keys("Support Us Should I Get Botox? \\nsource Add To The Conversation Using Facebook Comments November 1, 2016
55 | # 16 | click | css=.btn |
56 | self.driver.find_element(By.CSS_SELECTOR, ".btn").click()
57 | # 17 | assertAlert | Please enter a valid URL |
58 | assert self.driver.switch_to.alert.text == "Please enter a valid URL"
59 | # 18 | click | id=select-newsletter |
60 | self.driver.find_element(By.ID, "select-newsletter").click()
61 | # 19 | select | id=select-newsletter | label=Times Of India
62 | dropdown = self.driver.find_element(By.ID, "select-newsletter")
63 | dropdown.find_element(By.XPATH, "//option[. = 'Times Of India']").click()
64 | # 20 | click | css=#select-newsletter > option:nth-child(2) |
65 | self.driver.find_element(By.CSS_SELECTOR, "#select-newsletter > option:nth-child(2)").click()
66 | # 21 | click | id=input-data |
67 | self.driver.find_element(By.ID, "input-data").click()
68 | # 22 | type | id=input-data | https://www.thehindu.com/opinion/editorial/batting-for-borrowers-on-the-reserve-bank-of-indias-recent-draft-circular/article66774120.e
69 | self.driver.find_element(By.ID, "input-data").send_keys("https://www.thehindu.com/opinion/editorial/batting-for-borrowers-on-the-reserve-bank-of-indias-recent-dr
70 | # 23 | click | css=.btn |
71 | self.driver.find_element(By.CSS_SELECTOR, ".btn").click()

```

```

67 | self.driver.find_element(By.ID, "input-data").click()
68 | # 22 | type | id=input-data | https://www.thehindu.com/opinion/editorial/batting-for-borrowers-on-the-reserve-bank-of-indias-recent-draft-circular/article66774120.e
69 | self.driver.find_element(By.ID, "input-data").send_keys("https://www.thehindu.com/opinion/editorial/batting-for-borrowers-on-the-reserve-bank-of-indias-recent-dr
70 | # 23 | click | css=.btn |
71 | self.driver.find_element(By.CSS_SELECTOR, ".btn").click()
72 | # 24 | assertAlert | Please select the correct newsletter |
73 | assert self.driver.switch_to.alert.text == "Please select the correct newsletter"
74 | # 25 | click | id=select-newsletter |
75 | self.driver.find_element(By.ID, "select-newsletter").click()
76 | # 26 | select | id=select-newsletter | label=The Guardian
77 | dropdown = self.driver.find_element(By.ID, "select-newsletter")
78 | dropdown.find_element(By.XPATH, "//option[. = 'The Guardian']").click()
79 | # 27 | click | css=option:nth-child(4) |
80 | self.driver.find_element(By.CSS_SELECTOR, "option:nth-child(4)").click()
81 | # 28 | click | css=.btn |
82 | self.driver.find_element(By.CSS_SELECTOR, ".btn").click()
83 | # 29 | assertAlert | Please select the correct newsletter |
84 | assert self.driver.switch_to.alert.text == "Please select the correct newsletter"
85 | # 30 | click | id=input-data |
86 | self.driver.find_element(By.ID, "input-data").click()
87 | # 31 | click | id=select-newsletter |
88 | self.driver.find_element(By.ID, "select-newsletter").click()
89 | # 32 | select | id=select-newsletter | label=The Hindu
90 | dropdown = self.driver.find_element(By.ID, "select-newsletter")
91 | dropdown.find_element(By.XPATH, "//option[. = 'The Hindu']").click()
92 | # 33 | click | css=option:nth-child(3) |
93 | self.driver.find_element(By.CSS_SELECTOR, "option:nth-child(3)").click()
94 | # 34 | click | css=.btn |
95 | self.driver.find_element(By.CSS_SELECTOR, ".btn").click()
96 |
97 |

```



Result:

Project: Demo\_Test\*

Tests +

Search tests...

Test1\_basic

Test2\_basic\*

http://127.0.0.1:2000

	Command	Target	Value
30	click	id=input-data	
31	click	id=select-newsletter	
32	select	id=select-newsletter	label=The Hindu
33	click	css=option:nth-child(3)	
34	click	css= btn	

Command

Target

Value

Description

LogReference

29. assertAlert on Please select the correct newsletter OK

30. click on id=input-data OK

31. click on id=select-newsletter OK

32. select on id=select-newsletter with value label=The Hindu OK

33. click on css=option:nth-child(3) OK

34. click on css= btn OK

'Test2\_basic' completed successfully

21:12:25

21:12:26

21:12:26

21:12:26

21:12:26

21:12:26

21:12:26