DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

IT314 - SOFTWARE ENGINEERING

GUI TESTING

CRIME AND HAZARD MANAGEMENT SYSTEM

GROUP NO: 30

GROUP MEMBERS

202001410	PATEL AYUSH SANJAYBHAI
202001447	VAKANI HETAV ABHAYBHAI
202001467	JAY GROVER
202001421	PATEL KUNJ RAKESH
202001446	GONDALIYA VENIL CHANDUBHAI
202001275	AYUSH JAIN
202001458	KRIS PATEL
202001264	HARSH SANJAY MAKWANA
202001466	KALP KINJALBHAI PANDYA
202001438	NARODIA JEET NILESHKUMAR
202001457	HITARTH VYAS

GUI Testing

Selenium is a popular open-source tool for automated testing of web applications. It supports various programming languages, including Python, Java, Ruby, and C#. In Python, Selenium can be used to perform GUI testing of web applications by automating browser actions, such as clicking on buttons, filling out forms, and navigating between web pages.

In every class implemented for testing we have a constructor and destructor. In the constructor, we connected to mongoDB and inserted test data which we required at the beginning. In destructor, we deleted the test data from the database.

Login GUI tests

Test Case 1: Login success

```
    # test login success
    def test_login(self):
        selenium = webdriver.Edge()
        selenium.get('http://127.0.0.1:8000/myApp/login/')

        username = selenium.find_element(By.NAME, 'UserName')
        password = selenium.find_element(By.NAME, 'Password')
        submit = selenium.find_element(By.NAME, 'Submit')

        username.send_keys('testuser')
        password.send_keys('testpass')
        submit.send_keys(Keys.RETURN)

        assert 'myApp/' in selenium.current_url
        selenium.get('http://127.0.0.1:8000/myApp/logout/')
        selenium.quit()
```

First we go to the login page, then we find the username, password, and submit elements.After that, we then send the username and password to the input fields and submit the form. Then we check that the url redirected is the home page.

Test Case 2: Login failure

```
    # test login failure
    def test_login_failure(self):
        selenium = webdriver.Edge()
        selenium.get('http://127.0.0.1:8000/myApp/login/')

        username = selenium.find_element(By.NAME, 'UserName')
        password = selenium.find_element(By.NAME, 'Password')
        submit = selenium.find_element(By.NAME, 'Submit')

        username.send_keys('testuser')
        password.send_keys('wrongpass')
        submit.send_keys(Keys.RETURN)

        assert 'Invalid username or password' in selenium.page_source
        selenium.quit()
```

we go to the login page, then we find the username, password, and submit elements.After that, we then send the username and password to the input fields and submit the form. Then we check that the page source contains the error message or not.

Change Password GUI tests

Test Case 1: Change password success without login

```
    # test change password success without login
    def test_change_password_without_login(self):
        selenium = webdriver.Edge()
        selenium.get('http://127.0.0.1:8000/myApp/changePassword/')

        username = selenium.find_element(By.NAME, 'UserName')
        dob = selenium.find_element(By.NAME, 'DOB')
        password = selenium.find_element(By.NAME, 'newPassword')
```

```
        confirm_password = selenium.find_element(By.NAME, 'confirmNewPassword')
        submit = selenium.find_element(By.NAME, 'Submit')

        username.send_keys('testuser')
        dob.send_keys('01-01-2000')
        password.send_keys('newpass')
        confirm_password.send_keys('newpass')
        submit.send_keys(Keys.RETURN)

        assert 'myApp/login/' in selenium.current_url
        selenium.quit()
        self.collection.update_many({'UserName': 'testuser'}, {'$set':
{'Password': 'testpass'}})
```

We go to the change password page, then we find the username, dob, password, confirm password, and submit elements.After that, we then send the username, dob, password, and confirm password to the input fields and submit the form. Then we check that the url redirected is the login page or not as expected.

Test Case 2: Change password failure without login

```
    # test change password success with login
    def test_change_password_with_login(self):
        selenium = webdriver.Edge()
        selenium.get('http://127.0.0.1:8000/myApp/login/')

        username = selenium.find_element(By.NAME, 'UserName')
        password = selenium.find_element(By.NAME, 'Password')
        submit = selenium.find_element(By.NAME, 'Submit')

        username.send_keys('testuser')
        password.send_keys('testpass')
        submit.send_keys(Keys.RETURN)

        selenium.get('http://127.0.0.1:8000/myApp/changePassword/')

        username = selenium.find_element(By.NAME, 'UserName')
```

```
        dob = selenium.find_element(By.NAME, 'DOB')
        password = selenium.find_element(By.NAME, 'newPassword')
        confirm_password = selenium.find_element(By.NAME, 'confirmNewPassword')
        submit = selenium.find_element(By.NAME, 'Submit')

        username.send_keys('testuser')
        dob.send_keys('01-01-2000')
        password.send_keys('newpass')
        confirm_password.send_keys('newpass')
        submit.send_keys(Keys.RETURN)

        assert 'myApp/profile/' in selenium.current_url
        selenium.get('http://127.0.0.1:8000/myApp/logout/')
        selenium.quit()
        self.collection.update_many({'UserName': 'testuser'}, {'$set':
{'Password': 'testpass'}})
```

First we login with the testuser account, then we go to the change password page, then we find the username, dob, password, confirm password, and submit elements.After that, we then send the username, dob, password, and confirm password to the input fields and submit the form. Then we check that the url redirected is the profile page or not as expected.

Test Case 3: Change password failure without login (wrong username)

```
    # test change password failure without login (wrong username)
    def test_change_password_without_login_wrong_username(self):
        selenium = webdriver.Edge()
        selenium.get('http://127.0.0.1:8000/myApp/changePassword/')

        username = selenium.find_element(By.NAME, 'UserName')
        dob = selenium.find_element(By.NAME, 'DOB')
        password = selenium.find_element(By.NAME, 'newPassword')
        confirm_password = selenium.find_element(By.NAME, 'confirmNewPassword')
        submit = selenium.find_element(By.NAME, 'Submit')
```

```
        username.send_keys('wronguser')
        dob.send_keys('01-01-2000')
        password.send_keys('newpass')
        confirm_password.send_keys('newpass')
        submit.send_keys(Keys.RETURN)

        assert 'User does not exist' in selenium.page_source
        assert 'myApp/changePassword/' in selenium.current_url
        selenium.quit()
```

Test Case 4: Change password failure without login (wrong dob)

```
    # test change password failure without login (wrong dob)
    def test_change_password_without_login_wrong_dob(self):
        selenium = webdriver.Edge()
        selenium.get('http://127.0.0.1:8000/myApp/changePassword/')

        username = selenium.find_element(By.NAME, 'UserName')
        dob = selenium.find_element(By.NAME, 'DOB')
        password = selenium.find_element(By.NAME, 'newPassword')
        confirm_password = selenium.find_element(By.NAME, 'confirmNewPassword')
        submit = selenium.find_element(By.NAME, 'Submit')

        username.send_keys('testuser')
        dob.send_keys('01-01-2001')
        password.send_keys('newpass')
        confirm_password.send_keys('newpass')
        submit.send_keys(Keys.RETURN)

        assert 'Incorrect Date of Birth' in selenium.page_source
        assert 'myApp/changePassword/' in selenium.current_url
        selenium.quit()
```

For Test Cases 3,4, we go to the change password page, then we find the username, dob, password, confirm password, and submit elements.After that, we then send the username, dob, password, and confirm password to the input fields and submit the form. Then we check that the url redirected is the change password page or not as expected.

Edit Details GUI Tests

Test Case 1: Edit details success

```
    # test edit details success
    def test_edit_details_success(self):
        selenium = webdriver.Edge()
        selenium.get('http://127.0.0.1:8000/myApp/login/')

        username = selenium.find_element(By.NAME, 'UserName')
        password = selenium.find_element(By.NAME, 'Password')
        submit = selenium.find_element(By.NAME, 'Submit')

        username.send_keys('testuser')
        password.send_keys('testpass')
        submit.send_keys(Keys.RETURN)

        selenium.get('http://127.0.0.1:8000/myApp/editprofile/')

        username = selenium.find_element(By.NAME, 'UserName')
        email = selenium.find_element(By.NAME, 'Email')
        first_name = selenium.find_element(By.NAME, 'FirstName')
        last_name = selenium.find_element(By.NAME, 'LastName')
        dob = selenium.find_element(By.NAME, 'DOB')
        addressline1 = selenium.find_element(By.NAME, 'AddressLine1')
        addressline2 = selenium.find_element(By.NAME, 'AddressLine2')
        locality = selenium.find_element(By.NAME, 'Locality')
        pincode = selenium.find_element(By.NAME, 'Pincode')
        city = selenium.find_element(By.NAME, 'City')
        state = selenium.find_element(By.NAME, 'State')
        country = selenium.find_element(By.NAME, 'Country')
```

```
        latitude = selenium.find_element(By.NAME, 'Latitude')
        longitude = selenium.find_element(By.NAME, 'Longitude')
        mobile = selenium.find_element(By.NAME, 'Mobile')
        instagram = selenium.find_element(By.NAME, 'Instagram')
        twitter = selenium.find_element(By.NAME, 'Twitter')
        submit = selenium.find_element(By.NAME, 'submit')

        email.send_keys('newtest@gmail.com')
        first_name.send_keys('newtest')
        last_name.send_keys('newuser')
        dob.send_keys('01-01-2000')
        addressline1.send_keys('newaddress1')
        addressline2.send_keys('newaddress2')
        locality.send_keys('newlocality')
        pincode.send_keys('123456')
        city.send_keys('newcity')
        state.send_keys('newstate')
        country.send_keys('newcountry')
        latitude.send_keys('123')
        longitude.send_keys('456')
        mobile.send_keys('1234567890')
        instagram.send_keys('newinstagram')
        twitter.send_keys('newtwitter')
        submit.send_keys(Keys.RETURN)

        assert 'myApp/profile/' in selenium.current_url
        selenium.get('http://127.0.0.1:8000/myApp/logout/')
        selenium.quit()
```

We go to the login page, then we find the username, password, and submit elements. After that, we then send the username and password to the input fields and submit the form. Then we go to the edit details page, then we find all required elements. After that, we then send all test data to the input fields and submit the form. Then we expect the url redirected is the profile page.

Post Property GUI Tests

Test Case 1: Post property success

```
    # test post property success with login
    def test_post_property_success(self):
        selenium = webdriver.Edge()
        selenium.get('http://127.0.0.1:8000/myApp/login/')

        username = selenium.find_element(By.NAME, 'UserName')
        password = selenium.find_element(By.NAME, 'Password')
        submit = selenium.find_element(By.NAME, 'Submit')

        username.send_keys('testuser')
        password.send_keys('testpass')
        submit.send_keys(Keys.RETURN)

        selenium.get('http://127.0.0.1:8000/myApp/postProperty/')

        title = selenium.find_element(By.NAME, 'Title')
        description = selenium.find_element(By.NAME, 'Description')
        search_text = selenium.find_element(By.ID, 'search-input')
        search_button = selenium.find_element(By.ID, 'search-button')
        latitude = selenium.find_element(By.NAME, 'Latitude')
        longitude = selenium.find_element(By.NAME, 'Longitude')
        price = selenium.find_element(By.NAME, 'price')
        addressline1 = selenium.find_element(By.NAME, 'AddressLine1')
        addressline2 = selenium.find_element(By.NAME, 'AddressLine2')
        city = selenium.find_element(By.NAME, 'City')
        state = selenium.find_element(By.NAME, 'State')
        pincode = selenium.find_element(By.NAME, 'Pincode')
        submit_property = selenium.find_element(By.NAME, 'submit-property')

        title.send_keys('testproperty')
        description.send_keys('testdescription')
        search_text.send_keys('ahmedabad')
        search_button.send_keys(Keys.RETURN)
```

```
        # latitude.send_keys('123')
        # longitude.send_keys('456')
        price.send_keys('123456')
        addressline1.send_keys('testaddress1')
        addressline2.send_keys('testaddress2')
        city.send_keys('testcity')
        state.send_keys('teststate')
        pincode.send_keys('123456')
        submit_property.send_keys(Keys.RETURN)

        assert 'myApp/' in selenium.current_url
        selenium.get('http://127.0.0.1:8000/myApp/logout/')
        selenium.quit()
```

We go to the login page, then posted required data to the input fields and submit the form. Then we go to the post property page, then we find all required elements. After that, we then send all test data to the input fields and submit the form. Then we expect the url redirected is the home page.

Test Case 2: Post property failure

```
    # test post property failure without login
    def test_post_property_failure(self):
        selenium = webdriver.Edge()
        selenium.get('http://127.0.0.1:8000/myApp/postProperty/')

        assert 'myApp/login/' in selenium.current_url
        selenium.quit()
```

We try to go to the post property page without login. As we are not logged in, we expect the url redirected is the login page.

Output after running all test cases simultaneously

```
PS C:\Users\ayush\Desktop\Coding\new\IT314_project_30\Code\project> python manage.py test
Found 9 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
[392:14304:0428/215816.763:ERROR:chrome_browser_cloud_management_controller.cc(162)] Cloud management controller initialization aborted as CBCM is
 not enabled.
[392:14304:0428/215816.780:ERROR:api_wrapper.cc(102)] Calling IsEnclaveTypeSupported, error code 0

DevTools listening on ws://127.0.0.1:64746/devtools/browser/a989f161-854b-4a27-8fed-60d48b18e161
[392:14304:0428/215820.616:ERROR:fallback_task_provider.cc(124)] Every renderer should have at least one task provided by a primary task provider.
 If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a dependency of crbug.com/73978
2.
[392:14304:0428/215821.692:ERROR:fallback_task_provider.cc(124)] Every renderer should have at least one task provided by a primary task provider.
 If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a dependency of crbug.com/73978
2.
.[9784:8384:0428/215830.279:ERROR:chrome_browser_cloud_management_controller.cc(162)] Cloud management controller initialization aborted as CBCM i
s not enabled.
[9784:8384:0428/215830.300:ERROR:api_wrapper.cc(102)] Calling IsEnclaveTypeSupported, error code 0

DevTools listening on ws://127.0.0.1:59260/devtools/browser/640a6af8-4276-4a47-ba25-fa84fd0e38c5
.[19552:2348:0428/215841.036:ERROR:chrome_browser_cloud_management_controller.cc(162)] Cloud management controller initialization aborted as CBCM
is not enabled.
[19552:2348:0428/215841.053:ERROR:api_wrapper.cc(102)] Calling IsEnclaveTypeSupported, error code 0

DevTools listening on ws://127.0.0.1:59303/devtools/browser/c80ceac1-b6c2-47b8-a6e0-6f64c099ea10
.[17748:19572:0428/215850.572:ERROR:chrome_browser_cloud_management_controller.cc(162)] Cloud management controller initialization aborted as CBCM
 is not enabled.
[17748:19572:0428/215850.590:ERROR:api_wrapper.cc(102)] Calling IsEnclaveTypeSupported, error code 0

DevTools listening on ws://127.0.0.1:59344/devtools/browser/7bb2e0d1-8b85-4737-91ec-254481fef04a
.[1092:932:0428/215902.145:ERROR:chrome_browser_cloud_management_controller.cc(162)] Cloud management controller initialization aborted as CBCM is
 not enabled.
[1092:932:0428/215902.164:ERROR:api_wrapper.cc(102)] Calling IsEnclaveTypeSupported, error code 0
```

```
DevTools listening on ws://127.0.0.1:59389/devtools/browser/5c43353c-10df-4f9f-bdf2-0b15f0ce5005
[1092:932:0428/215905.437:ERROR:fallback_task_provider.cc(124)] Every renderer should have at least one task provided by a primary task provider.
If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a dependency of crbug.com/739782
.
[1092:932:0428/215906.638:ERROR:fallback_task_provider.cc(124)] Every renderer should have at least one task provided by a primary task provider.
If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a dependency of crbug.com/739782
.
.[16084:10180:0428/215916.547:ERROR:chrome_browser_cloud_management_controller.cc(162)] Cloud management controller initialization aborted as CBCM
 is not enabled.
[16084:10180:0428/215916.580:ERROR:api_wrapper.cc(102)] Calling IsEnclaveTypeSupported, error code 0

DevTools listening on ws://127.0.0.1:59454/devtools/browser/880145a7-91cc-46a6-ad09-53509eb228e0
[16084:10180:0428/215919.627:ERROR:fallback_task_provider.cc(124)] Every renderer should have at least one task provided by a primary task provide
r. If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a dependency of crbug.com/739
782.
.[20556:21512:0428/215928.468:ERROR:chrome_browser_cloud_management_controller.cc(162)] Cloud management controller initialization aborted as CBCM
 is not enabled.
[20556:21512:0428/215928.480:ERROR:api_wrapper.cc(102)] Calling IsEnclaveTypeSupported, error code 0

DevTools listening on ws://127.0.0.1:59502/devtools/browser/8b1f7564-9dfa-4726-a601-97021799dbb1
.[16152:22552:0428/215939.269:ERROR:chrome_browser_cloud_management_controller.cc(162)] Cloud management controller initialization aborted as CBCM
 is not enabled.
[16152:22552:0428/215939.291:ERROR:api_wrapper.cc(102)] Calling IsEnclaveTypeSupported, error code 0

DevTools listening on ws://127.0.0.1:59545/devtools/browser/39628e47-db43-4973-adc7-98bafda1a960
.[8120:16240:0428/215948.465:ERROR:chrome_browser_cloud_management_controller.cc(162)] Cloud management controller initialization aborted as CBCM
is not enabled.
[8120:16240:0428/215948.478:ERROR:api_wrapper.cc(102)] Calling IsEnclaveTypeSupported, error code 0
```

```
DevTools listening on ws://127.0.0.1:59582/devtools/browser/955d2dc7-4e3b-4765-874d-aef43938e209
[8120:16240:0428/215951.562:ERROR:fallback_task_provider.cc(124)] Every renderer should have at least one task provided by a primary task provider
. If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a dependency of crbug.com/7397
82.
[8120:16240:0428/215952.797:ERROR:fallback_task_provider.cc(124)] Every renderer should have at least one task provided by a primary task provider
. If a "Renderer" fallback task is shown, it is a bug. If you have repro steps, please file a new bug and tag it as a dependency of crbug.com/7397
82.
.
----------------------------------------------------------------------
Ran 9 tests in 107.750s

OK
Destroying test database for alias 'default'...
PS C:\Users\ayush\Desktop\Coding\new\IT314_project_30\Code\project>
```