Test plan Document

By: Mlungisi Gumede
Dated: 08/19/2024

Test Automation Link : https://youtu.be/nvIUhtgdyrs

Introduction:

The Voss Solutions technical assessment is designed to comprehensively evaluate my competencies in Software Quality Assurance (SQA). This assessment will gauge my approach to tackling the tasks at hand, adherence to industry best practices, and overall effectiveness in QA activities. Through this evaluation, I aim to demonstrate a thorough understanding of quality assurance methodologies, my ability to implement effective testing strategies, and my commitment to delivering high-quality software products.

Test Items:

- **Fetch SSL certificate:** For the TC to pass, the SSL certificate needs to be checked for expiration on the 'https://www.ultimateqa.com' site. action using the following link 'https://www.ultimateqa.com/automation/' **(out of scope)**

- **Fake Pricing:** For the TC to pass, the user needs to access the 'Fake Pricing Page' and purchase the Purchase the Basic package successfully for it to pass action using the following link 'https://www.ultimateqa.com/automation/'**. (PASS)**

- **Fill out Forms:** In order for the TC to pass, the user needs to access the '"Fill out forms" page and complete all forms, followed by submit action using the following link 'https://www.ultimateqa.com/automation/'**. (PASS)**
-

- **Automation Login:** In order for the TC to pass, the user needs to access the 'Login automation' link and then sign out again once logged In in order for the TC to pass action using the following link 'https://www.ultimateqa.com/automation/'**. (PASS)**
-

Testing Approach

The testing approach for this assessment will initially involve **manual testing**. All actions will be performed manually to verify that the functionality is correct. Utilizing developer inspection tools, we will thoroughly review the performance and logic of the page to identify any issues or areas for improvement.
Upon successful completion of manual testing, **unit automation tests** will be executed for each test case in our scope using **ROBOTFRAMEWORK**. This step ensures that the tests, once validated manually, can be executed automatically in future cycles, thereby reducing execution time and improving efficiency.

Entry & Exit Criteria

- **Entry**: All manual unit tests passed.
- **Exit:** All test cases passed.

Test Deliverables

- **Test cases**
- **test script**
- **summary report**

Test Schedule

- Test Case Development: 19 August 2024
- Test Execution: 19 August 2024

Responsibilities

Automation scripting: Mlungisi Gumede
Document Validation and compilation: Mlungisi Gumede
Manual execution, Requirements Analysis: Mlungisi Gumede

Risks and Contingencies

- Risk: Delay in testing delivery.
- Mitigation: Schedule buffer of 2 days.

***Approval and Sign-Off - Approved by*:** Grant Wilson **(Project Manager, QA Lead)**, Mlungisi Gumede (**SQA Engineer)**

contact:
Mlungisi Gumede : u19074362@tuks.co.za
+27 65 304 6956
Grant Wilson : Grant.wilson@Voss-solutions.com

Follow up questions

1. **How could you reduce the time to execute some or all these test cases or if you had several sites to test?**
   - Reduce wait times on functions on execution times.
   - Test Automation.

- Environment set up and correct infrastructure.

2. **Briefly (a few sentences are fine) describe how you would set up a pipeline on your preferred source control management platform to perform continuous integration testing of a simple front-end web-based application. It could be anything you like, eg a calculator that outputs the correct results from numbers you input.**

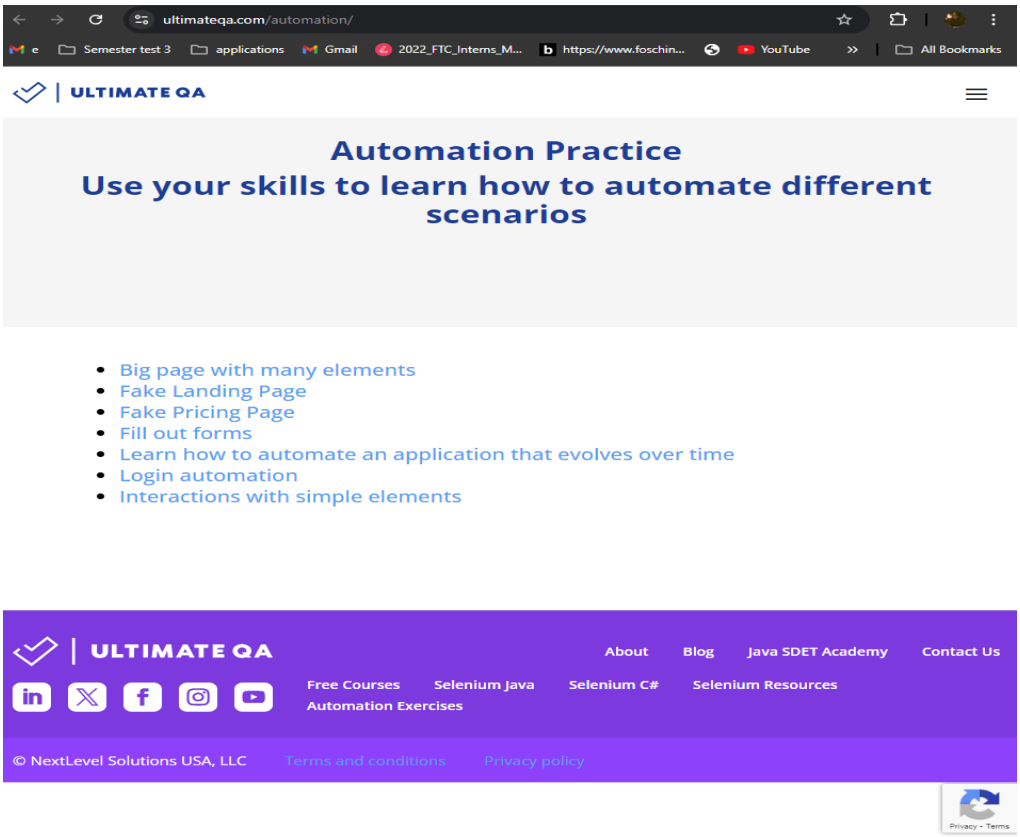
- Repository Setup: Store your project code in a GitHub repository.
- CI Workflow: Create a .GitHub/workflows/ci.yml file in your repository to define the CI pipeline.
- Job Configuration: In the ci.yml file, configure jobs to run on every push or pull request.
- **Build Verification**: Optionally, include steps to build the application and ensure the build succeeds.
- **Automated Deployment**: If needed, add steps to deploy the app to a staging environment after successful testing.

3. **Describe briefly how you would run performance testing against a web-based application.**
   - Tool Selection: Choose a performance testing tool like JMeter
   - Test Plan Creation: Define the performance test scenarios, including user behaviors, target endpoints, and the number of virtual users.
   - Test Execution: Run the performance tests by simulating the defined user load against your application.

4. **Describe briefly what kind of security testing you might perform against a web-based application.**
   - **Penetration Testing:** Conduct manual or automated penetration tests to simulate real-world attacks. This includes attempting to exploit vulnerabilities, gain unauthorized access, and assess the application's resilience against various attack vectors.
   - **Authentication and Authorization Testing**: Evaluate the effectiveness of authentication mechanisms (e.g., login processes) and access control measures to ensure that users can only access resources they are authorized for.
5. **Describe how you might build in exception and error handling to your application.**
   - **Error Detection:** Validation Checks: Implement input validation and checks to catch potential errors early, such as invalid user inputs or malformed data.
   - **User Notifications**: **Friendly Error Messages**: Provide user-friendly error messages to inform users of issues without exposing technical details or system internals.
   - **Testing and Monitoring**: **Error Testing**: Write tests to ensure that exceptions are handled as expected and that error conditions are properly managed.

1. TC Automation Login

Step 1: Open browser link 'https://ultimateqa.com/automation/'

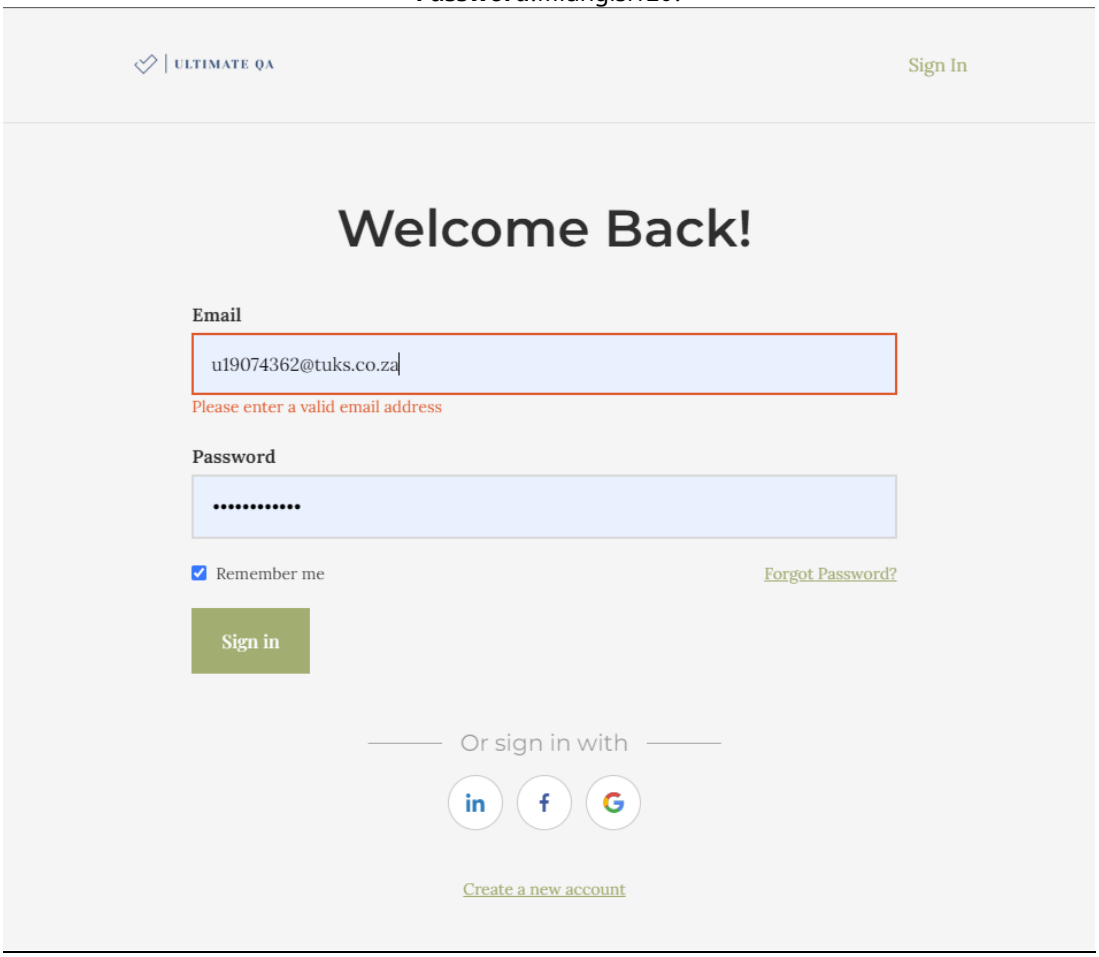


- **Step 2:** Navigate to and select Login automation Link

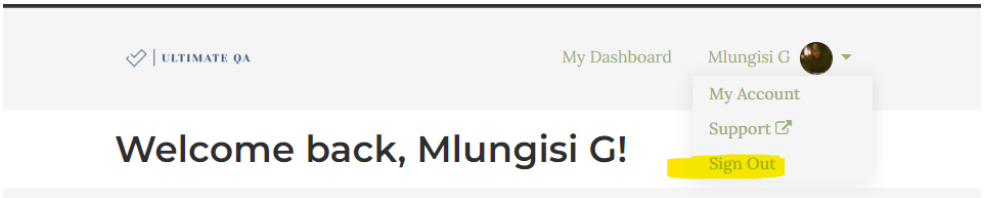
- Big page with many elements
- Fake Landing Page
- Fake Pricing Page
- Fill out forms
- Learn how to automate an application that evolves over time
- Login automation
- Interactions with simple elements

Step 3: Sign in: The user should be logged in

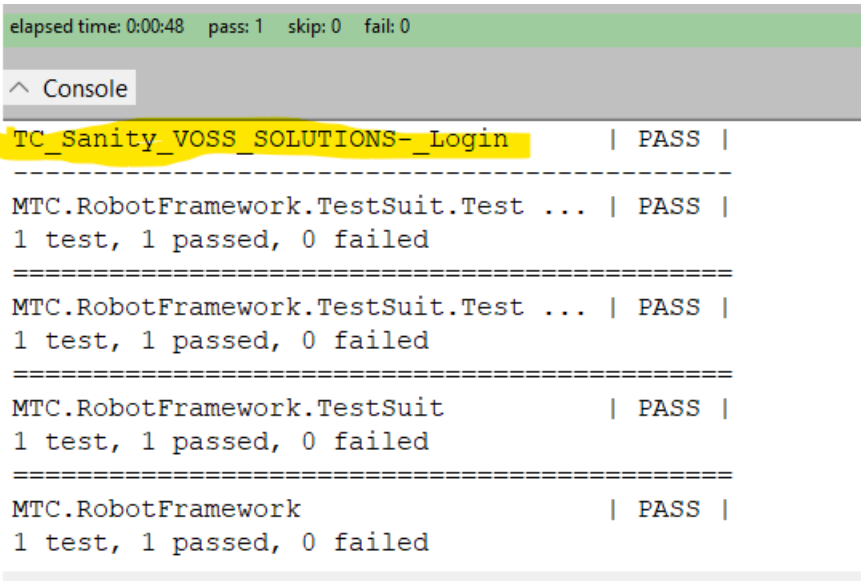
<u>Step 3 :</u> Insert login credentials **email**:u19074362@tuks.co.za
Password:Mlungisi120!



Step 4: Navigate to and select Navigation menu to select 'Sign out' the user should be signed out successfully.

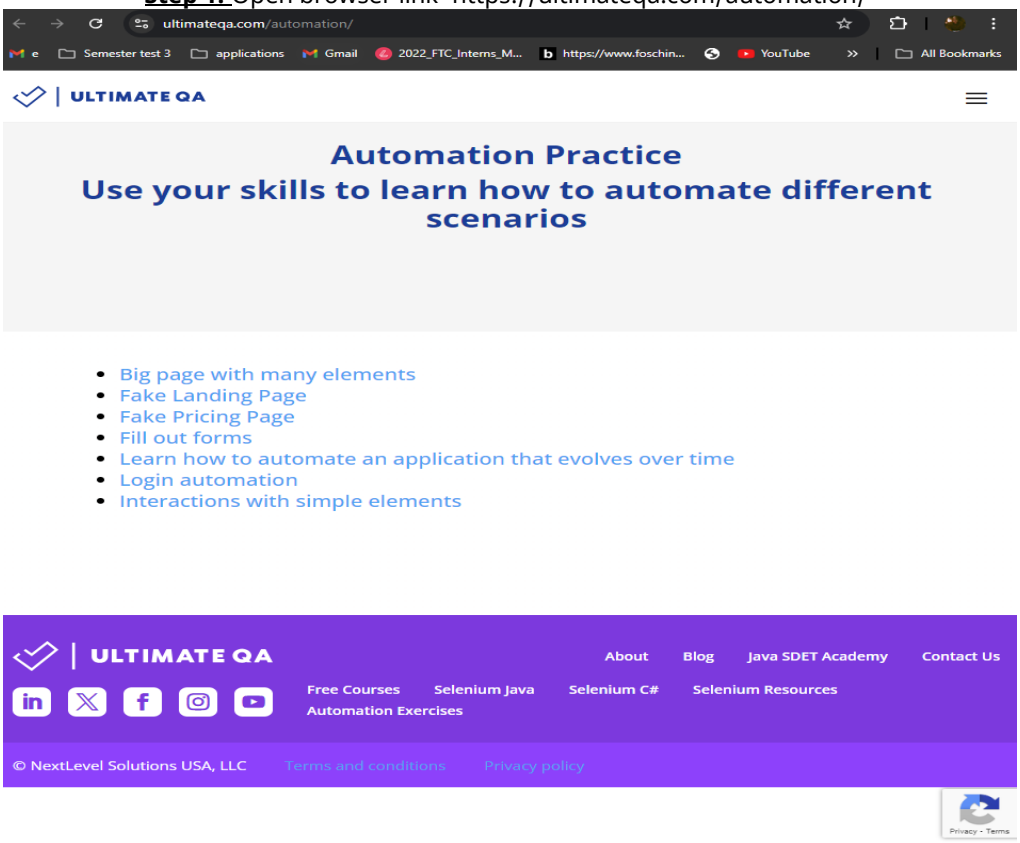


ROBOTFRAMEWORK RESULTS : TC_Sanity_VOSS_SOLUTIONS_Login **(PASS)**



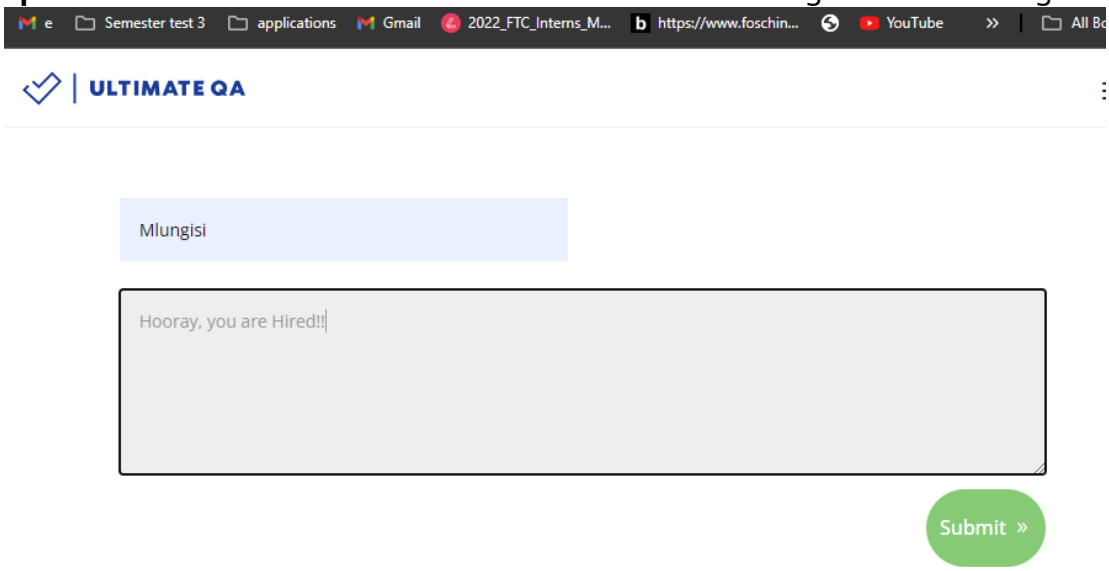
2. **<u>TC Fill out Forms</u>**

Step 1: Open browser link 'https://ultimateqa.com/automation/'

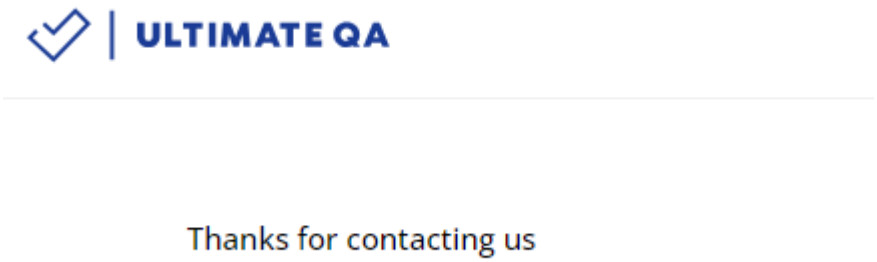


- **Step 2:** Navigate to and select https://ultimateqa.com/filling-out-forms/ Link

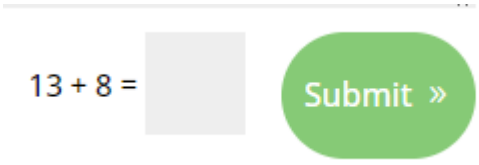
Step 3: Insert Name under ‘Name textbox’ and insert message under ‘Message box’



Step 4: Select ‘Submit button’ and a message ‘Thanks for contacting us’ should appear



Step 5: Repeat steps from step 3 and additionally enter the value ‘21’ in the equation text box

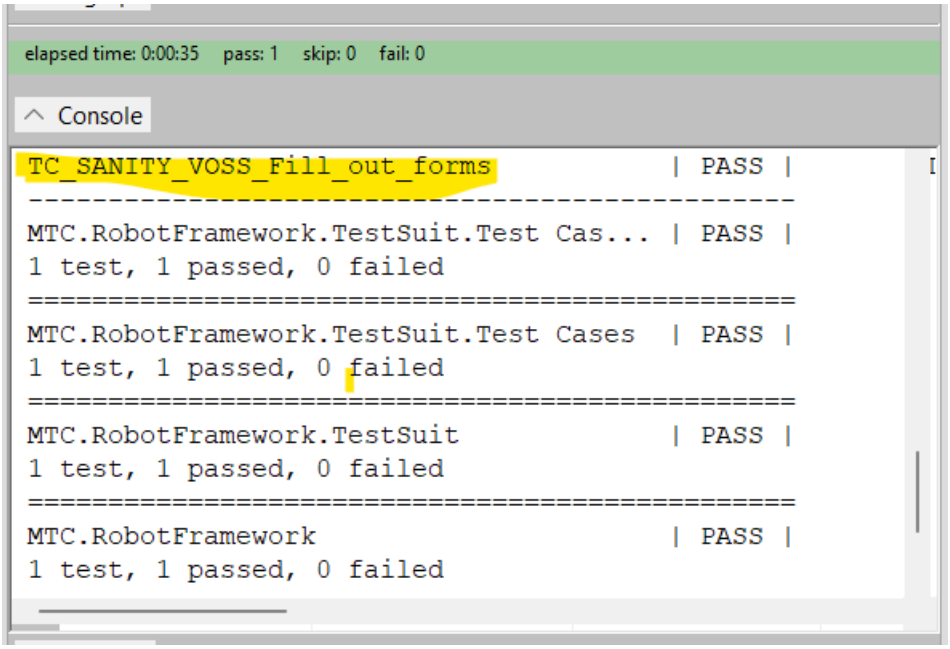


Step 6: Select submit again and results viewed In step 4 should appear again

ULTIMATE QA

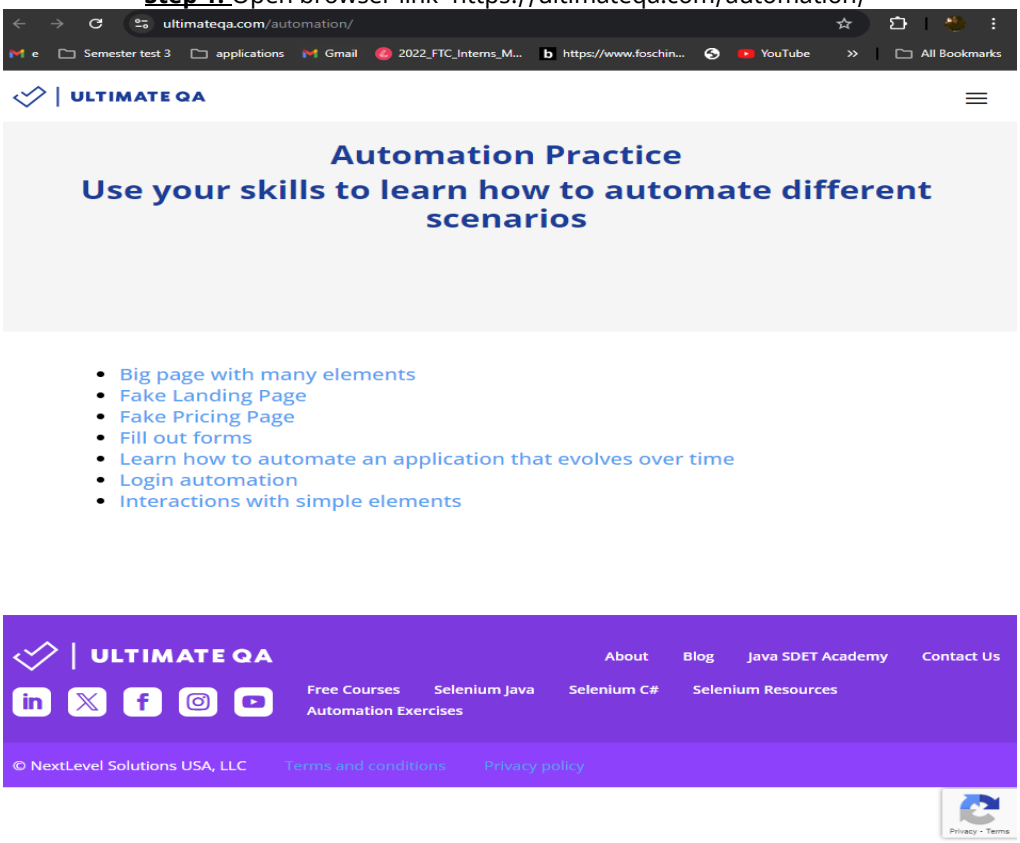
Thanks for contacting us

ROBOTFRAMEWORK RESULTS : TC_SANITY_VOSS_Fill_out_forms **(PASS)**

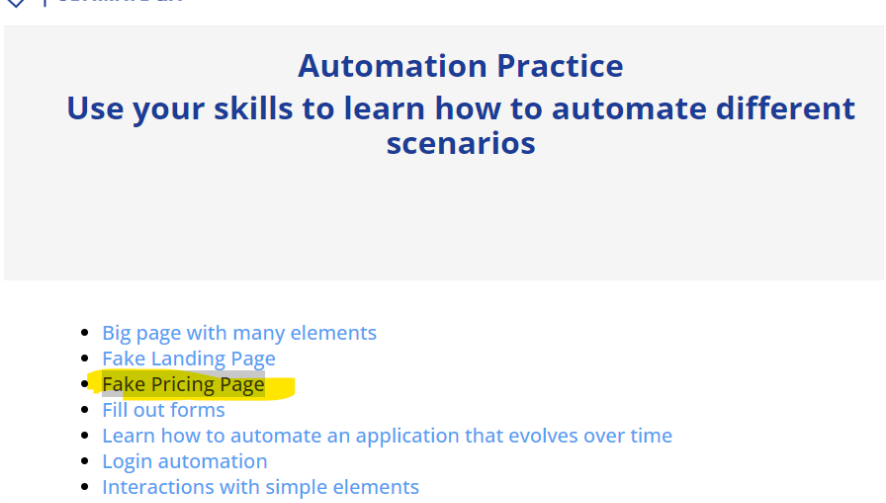


3. Fake Pricing

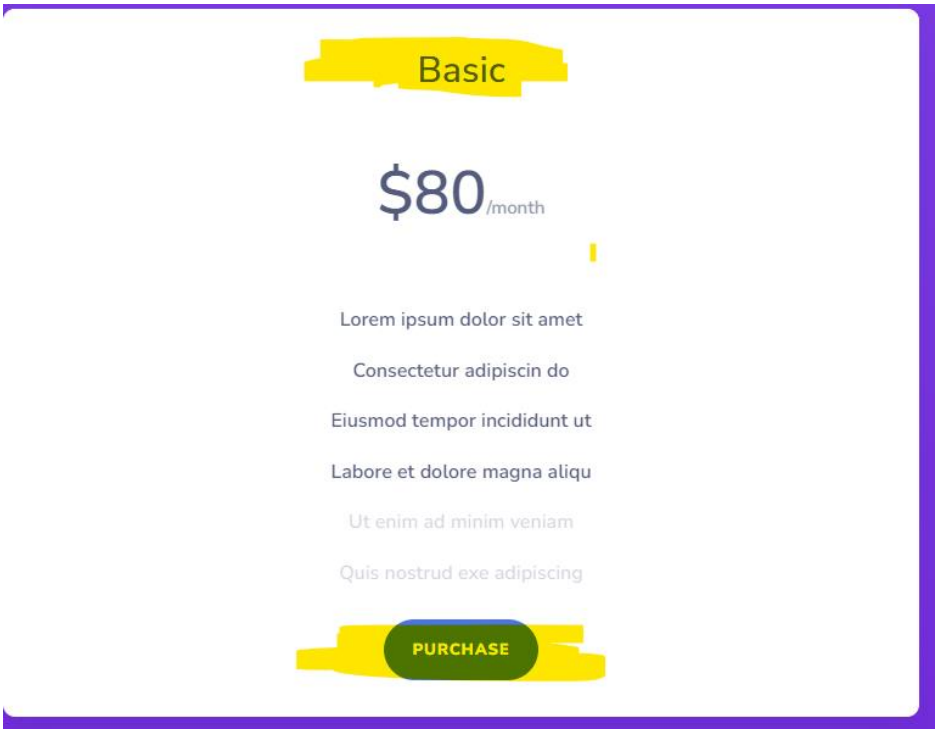
Step 1: Open browser link 'https://ultimateqa.com/automation/'



Step 2: Navigate to and select Fake Pricing Page Link



Step 3 : Navigate to and select 'Purchase' button under 'Basic' Plan.



ROBOTFRAMEWORK RESULTS : TC_VOSS_Solutions_Fake_Pricing **(PASS)**