SDET TEST (Adi Darmawan)

FAQ and HOW TO

- How long is the assignment supposed to take? *The assignment has been designed to be finished in 1-3 days. but it might take longer to get to know these frameworks and practices.*
- For the non-code test, please use Google doc or sheet, and provide us the link.
- Do I have to submit to **GIT** repository? ***YES**, as part of good development practice, we expect you to do frequent commits with proper git commit messages. So we will verify them in the evaluation phase. (Please use **PRIVATE GITHUB** repository) or otherwise submit with attachment in email.*
- Do I have to write test for code that I have written? ***YES**, as part of good development practice, we expect you to write a unit test that cover negative and positive cases.*
- **When you have completed this assignment, please send us your repository link and please make sure we can access it thoroughly.**

[QA Basic]
1. Create any possible test case for refrigerator!
2. Mention any elements that should be present on a Register and Sign In page and its functionality!
3. Create any possible test case for Register and Sigin in!
4. Mention any possible front-end and back-end issue on Register and Sigin in!
5. For Register and Sign in feature, mention any security issue that might be occur!

[Answer]
1. **Functionality :**
   a. Check the cooling of each setting.
   b. Check if the light is turned off when the door is closed.
   c. Check if the light is turned on when the door is opened.
   d. Check if there are specific settings for specific types of food.
   e. Verify the setting of the refrigerator.
   f. Check the cooling of each setting.

   Usability :
   a. Check if the door is easy to open.
   b. Check if the interior is easy to clean.
   c. Verify the hinges of the door.

   Performance :
   a. Check if the refrigerator can handle the voltage variation.

b. Check at what voltage level the fridge stops functioning.

c. Check if there are any voltage fluctuations when the door is open.

2. **Elements and functionality on Register Page :**

    a. Username field → the username field for user logins into page after registering.

    b. Name field → data for that user.

    c. Email field → confirm the registration and data for login.

    d. Password field → security.

    e. User agreement checkbox → user agrees to the applicable police after registering.

    f. Submit button → send data to server.

    g. Login button → if the user has an account at that page, the user can go to the login page easily.

    h. Hide/unhide button on password field → user can see the password.

    i. Placeholders on each field → user can fill the data easily.

    j. Error Message → user can find out if there is an input error.

    k. Labels on each field → user can find out which fields are being filled easily.

    Elements and functionality on Sign in Page :

    a. Email field → input for login.

    b. Password Field → input for login and security.

    c. Submit Button → send data to server.

    d. Forgot Password Button → if the user forgets the password, the user can reset the password easily.

    e. Hide/unhide button on password field → user can see the password.

    f. Placeholders on each field → user can fill the data easily.

    g. Error Message → user can find out if there is an input error.

    h. Labels on each field → user can find out which fields are being filled easily.

3. **Test case for register page :**

    Functional :

    a. Verify that all the specified fields are present on the page.

    b. Verify that the required/mandatory fields are marked.

    c. Verify the register page if both of the fields are blank and the submit button is clicked, warning should be displayed.

    d. Verify that entering blank space on mandatory fields leads to the validation error.

    e. Verify that all fields have a valid placeholder.

    f. Verify that clicking on the submit button after entering all the mandatory fields, submits the data to the server.

    g. Verify that clicking on the submit button by leaving optional fields, submits the data to the server without any validation error.

h. Verify that system generates a validation message when entering existing username/ email.
i. Verify that the validation of the email field by entering incorrect email id.
j. Verify that the "terms and conditions" checkbox is unselected by default (depends on business logic).
k. Verify that the password and confirm password are the same or not.
l. Check validation on numeric fields by entering alphabets and special characters.

Non-Functional :
a. Verify that the password is encrypted when entered.
b. Verify the implementation of SSL certificate.

Test case for Sign in page :
Functional :
a. Verify if a user will be able to login with valid data.
b. Verify if a user cannot login with a valid username/email and an invalid password.
c. Verify if a user cannot login with invalid data.
d. Verify the login page if both of the fields are blank and the submit button is clicked.
e. Verify the message for invalid login.
f. Verify the 'Forgot Password' functionality.
g. Verify if the data in the password field is either visible as asterisk or bullet signs.
h. Verify if the user is able to login with a new password only after the user has changed the password.

Non-Functional :
a. Verify if a user cannot the characters more than the specified range in each field.
b. Verify the login page by pressing 'Back Button' of the browser. It should not allow you to enter into system once you log out.
c. Verify the timeout functionality of the login session.
d. Verify the login page against SQL injection attack.
e. Verify the implementation of SSL certificate.

4. Front-end and back-end issue on Register and Sign in page :
   a. No field validation.
   b. No autofocus on the field.
   c. No crosslink between Login/Sign up.
   d. The page that never finished loading after submit..

5. Security issue on register and sign in feature :
   a. Descriptive error messages.
   b. Weak password requirements.
   c. Password complexity that's not enforced.

d. Allowing unlimited login attempts.
e. Storing password as plaintext.
f. Not regenerating season.