

Task 1

Test the buggy application on URL http://goalsec.com/software_quality/

Incorrect format of output

Steps: Enter "eee" as first name and "eee" as last name. Press "Submit" button.

Expected: "You are called eee eee. You are 144cm long and your weight is 78kg."

Actual: "Welcome eee eee. You are 144cm long and your weight is 78kg."

Empty first name or last name

Steps: Enter "whatever" as first name and leave last name blank (or vice versa).

Expected: No output (enter non-empty First Name and Last Name)

Actual: "Welcome a . You are 158cm long and your weight is 64kg."

Unescaped output and XSS

Steps: Enter characters that should be converted to special html entities, for example:

"whatever" or possible XSS vulnerability:

"<script>alert('Whoops'); window.location.href='http://is.muni.cz'</script>"

Expected: See "Welcome <a>Whoops . You are 158cm long and your weight is 64kg."

Actual: Formatted text, or alert, or redirect, etc.

What other testing methodologies would you use in order to verify this application and find more bugs?

Acceptance testing, Selenium test suit with various edge case inputs, so we can test whether the application conforms to specification.

What would you like to know about this (or similar application) if you were about to test them in real life? Write at least 3 reasonable questions (which are not answered in the specification) you would like to ask PM and/or DEV prior testing.

I would inquire about the following topics since the application is input based

1. Input characters - only alphabetic characters, spaces, period for names, other characters do not make sense.
2. Input limitation - maximum number of characters - server side + client side.
3. Input escaping - trim the input, escape html special entities with `htmlspecialchars()` function, is it necessary to use GET method.

Task 2

Your team is about to implement one of the top 20 biggest world's eshops selling to most of the world countries (assume the list of countries is given).

By Google's ACC method, identify at least:

- 7 attributes which you think are the most relevant for this product to be successful.
 1. security,
 2. speed
 3. ease of use
 4. localization
 5. accessibility
 6. multiplatform
 7. modularization
- 11 components which are (almost) necessary for the product to be successful.
 1. shopping cart
 2. promotion codes/royalty system
 3. suggested products
 4. product list
 5. product search
 6. product overview
 7. product reviews/rating/feedback
 8. checkout/cashier
 9. customer support
 10. shipping tracking
 11. promotional/sales/deals newsletter
- 3 capabilities for one combination of attribute and component.
 - Speed + product list
 1. Infinite scrolling
 2. Real time filtering
 3. Multiple views for clarity, faster scrolling, wider devices, etc.
- 10 other capabilities which should be tested.
 1. Add item to shopping cart
 2. Remember shopping cart content
 3. Filters' results correctness
 4. Sales/deals calculation of product's price
 5. Applying promo code - calculation of product's price
 6. Invalid/expired promo code
 7. Adding review on product's page
 8. Checkout process with payment
 9. Customer support tickets
 10. Sending email to customer

- 3 test cases for one capability.
 1. User adds promotional code “PV260” on product checkout page to product “Product #1” with price of \$100, a message appears saying “Promotional code PV260 was applied for 20% discount.”, the product’s price is now \$80.
 2. User adds promotional code “PV2601” on product checkout page to product “Product #1” with price of \$100, a message appears saying “Promotional code PV2601 is invalid.”, the product’s price remains at \$100.
 3. User adds promotional code “PV2602” on product checkout page to product “Product #1” with price of \$100, a message appears saying “Promotional code PV2602 has already expired.”, the product’s remains at \$100.

Identify six of the most risky places (product risks) of this product.

1. Safety of checkout - credit card payment spoofing/fishing/etc.
 - User can be charged with more money, credit card can be emptied out
2. Sensitive information when paying - not storing customer’s credit card security code
 - Respect customers privacy, prevent leaks, laws, ...
3. Sensitive information about user - use encryption when storing passwords
 - Same reasons as mentioned above (2.)
4. Privacy laws - option to opt out of newsletter, showing information about cookie policy
 - European union regulations, customers get annoyed when they are spammed often
5. Chargebacks/disputes - have invoices, confirmation and well written terms and conditions so users cannot dispute purchase for no reason
 - Meaning eshop would have to pay a fee for each dispute/chargeback that is lost, loss of reputation and customers
6. Product safety - only trusted people can set prices & description, user’s products don’t get mixed up with other users
 - Reputation, fraud, customer loss, ...

Why do you think those are the most risky places? Give at least 2 arguments.

Mentioned in sublist above.

Task 3

Identify at least 5 different components (visible to all Internet users) of Wikipedia.
page, login/sign up, languages, portals, content editing/creation

For each of these components, write a test case in Selenium.
Test cases are located in Github repository.

From those 5 automated tests, make at least 1 negative test case and at least 1 positive test case.