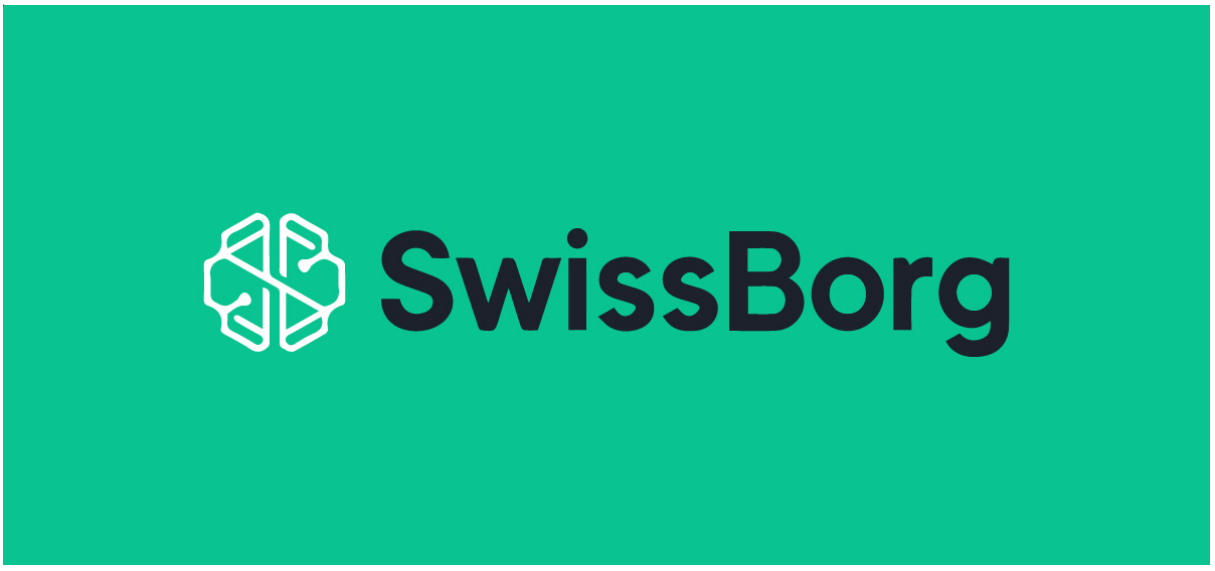


SwissBorg

Exercise 1: Online Factorial Calculator Testing

Overview

The task involved a thorough testing of the online factorial calculator hosted at http://qainterview.pythonanywhere.com/. The objective was to explore the website and the factorial calculator to test the functionality and identify any bugs and UX issues. Additionally, a script was to be written to verify the correctness of the calculator for integer numbers in the range (10, 100).

Testing Methodology

- **Manual Testing**: Conducted an exploratory testing session focusing on the calculator's functionality, UI elements, error handling, and response times.
- **Automated Testing**: Developed a script to automatically test the calculator's functionality for integer values between 10 and 100.

🚀 Test Cases 🚀

1. **Valid Input Test** 🟢
    - **Test Priority**: High
    - **Test Severity**: Medium
    - **Test Data**: 9
    - **Steps**:
        - Enter the test data (9) in the input area.
        - Click on the "Calculate!" button.
        - Verify that the calculator returns the correct factorial value.
2. **Boundary Test** 🟡
    - **Test Priority**: Medium
    - **Test Severity**: High
    - **Test Data**: 10, 100, 1000
    - **Steps**:
        - Enter the first test data (10) in the input area.
        - Click on the "Calculate!" button.
        - Verify that the calculator returns the correct factorial value.
        - Repeat steps 1-3 for the other test data (100, 1000).
3. **Invalid Input Test** 🔴
    - **Test Priority**: High
    - **Test Severity**: Medium
    - **Test Data**: -5, "abc", "!@#", 5-
    - **Steps**:
        - Enter the first test data (-5) in the input area.
        - Click on the "Calculate!" button.
        - Verify that the calculator displays an error message indicating that negative numbers are not supported.
        - Repeat steps 1-3 for the second test data ("abc"), third test data ("!@#") and fourth test data (”5-”).

4. **Performance Test** 🟡
    - **Test Priority**: Medium
    - **Test Severity**: Low
    - **Test Data**: 50
    - **Steps**:
        - Enter the test data (50) in the input area.
        - Click on the "Calculate!" button.
        - Measure the response time of the calculator.
        - Compare the response time against the acceptable limits.
5. **Usability Test** 🔵
    - **Test Priority**: Low
    - **Test Severity**: Low
    - **Test Data**: None (empty input)
    - **Steps**:
        - Leave the input area empty.
        - Click on the "Calculate!" button.
        - Evaluate the overall user experience, including the readability of instructions, ease of inputting values, and clarity of results.
6. **Additional Test Cases**:
    - **Placeholder Text Visibility Test** 🔵
        - **Test Priority**: Low
        - **Test Severity**: Low
        - **Test Data**: None (empty input)
        - **Steps**:
            - Verify that the placeholder text "Enter an integer" is visible when there is no input value or when the input value is deleted.
    - **Negative Input Test** 🟡
        - **Test Priority**: Medium
        - **Test Severity**: High

    - **Test Data**: -5
    - **Steps**:
        - Enter the test data (-5) in the input area.
        - Click on the "Calculate!" button.
        - Verify the result message.
- **Result Message Test** 🟡
    - **Test Priority**: Medium
    - **Test Severity**: Low
    - **Test Data**: 7
    - **Steps**:
        - Enter the test data (7) in the input area.
        - Click on the "Calculate!" button.
        - Verify the result message.
- **No Input Test** 🔴
    - **Test Priority**: High
    - **Test Severity**: High
    - **Test Data**: None (empty input)
    - **Steps**:
        - Click on the "Calculate!" button without entering any input value.
        - Verify that an error message is displayed, stating "Please enter an integer".
- **Invalid Input Characters Test** 🔴
    - **Test Priority**: High
    - **Test Severity**: High
    - **Test Data**: "abc", "!@#"
    - **Steps**:
        - Enter the first test data ("abc") in the input area.
        - Click on the "Calculate!" button.

        - Verify that an error message is displayed, stating "Please enter an integer".
        - Repeat steps 1-3 for the second test data ("!@#").
- **Multiple Digit Input Test** 🔴
    - **Test Priority**: High
    - **Test Severity**: Medium
    - **Test Data**: 7, 26, 123
    - **Steps**:
        - Enter the first test data (7) in the input area.
        - Click on the "Calculate!" button.
        - Verify the result.
        - Repeat steps 1-3 for the second test data (26) and third test data (123).
- **Link Test** 🔵
    - **Test Priority**: Low
    - **Test Severity**: Low
    - **Test Data**: Click on each link: "Terms and Conditions," "Privacy," and "Qxf2 Services"
    - **Steps**:
        - Click on the "Terms and Conditions" link.
        - Verify that the corresponding message or webpage is displayed correctly.
        - Repeat steps 1-2 for the "Privacy" and "Qxf2 Services" links.
        - Additionally, observe that clicking on the "Terms and Conditions" or "Privacy" link opens a new subpage with the corresponding URL extension (e.g., /privacy or /terms).

7. **Larger Number Test** 🔵
    - **Test Priority**: Low
    - **Test Severity**: Low
    - **Test Data**: 22
    - **Steps**:

- Enter the test data (22) in the input area.
- Click on the "Calculate!" button.
- Verify that the result message includes the factorial value and the exponent notation.

For example, the expected message might be "22! = 1.1240007277776077e+21".

8. **Float Input Test** 🔴
   - **Test Priority**: High
   - **Test Severity**: High
   - **Test Data**: 3.14, 2.5, 0.5
   - **Steps**:
     - Enter the first test data (3.14) in the input area.
     - Click on the "Calculate!" button.
     - Verify that an error message is displayed, stating that user needs to enter integer only numbers.
     - Repeat steps 1-3 for the second test data (2.5) and third test data (0.5).

🤖 Automated Test Script 🤖

The script was developed using (specify the programming language and framework used), designed to:

- Loop through a range of integer numbers from 10 to 100.
- For each number, it sends a request to the calculator.
- Verifies the response against the expected factorial value.
- Reports any discrepancies found.

🐞 Bugs 🐞

🐞 Bug: User is unable to use the keyboard Enter key to perform the calculation.

Steps to Reproduce:

1. Insert a number (e.g., 9) in the input area.
2. Press the Enter key on the keyboard.

Expected Result: The factorial calculation should be performed.

Actual Result: Nothing happens when the Enter key is pressed.

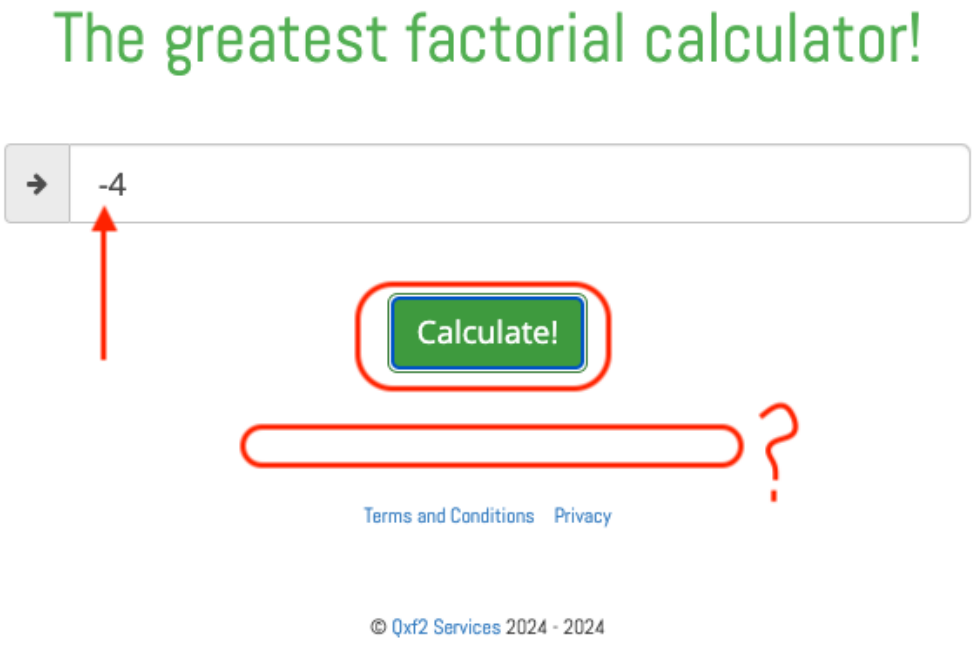
🐞 Bug: No error message is displayed when a negative number is entered.

Steps to Reproduce:

1. Insert a negative number (e.g., -5) in the input area.
2. Click on the "Calculate!" button.

Expected Result: An error message should be displayed, indicating that negative numbers are not supported.

Actual Result: No error message is displayed.



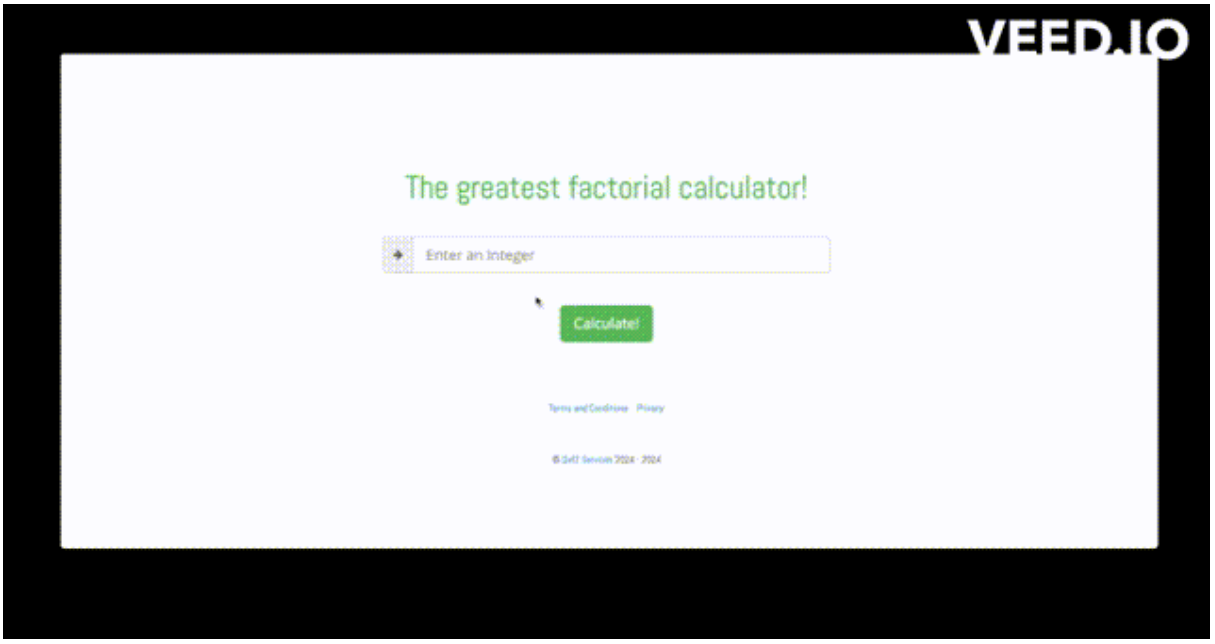
🐞 Bug: No error message is displayed or any action performed when a 4-digit number is entered.

Steps to Reproduce:

1. Insert a 4-digit number (e.g., 1234) in the input area.
2. Click on the "Calculate!" button.

Expected Result: An error message should be displayed, indicating that only numbers within the range of 1 and 170 are going to be calculated.

Actual Result: No error message is displayed.



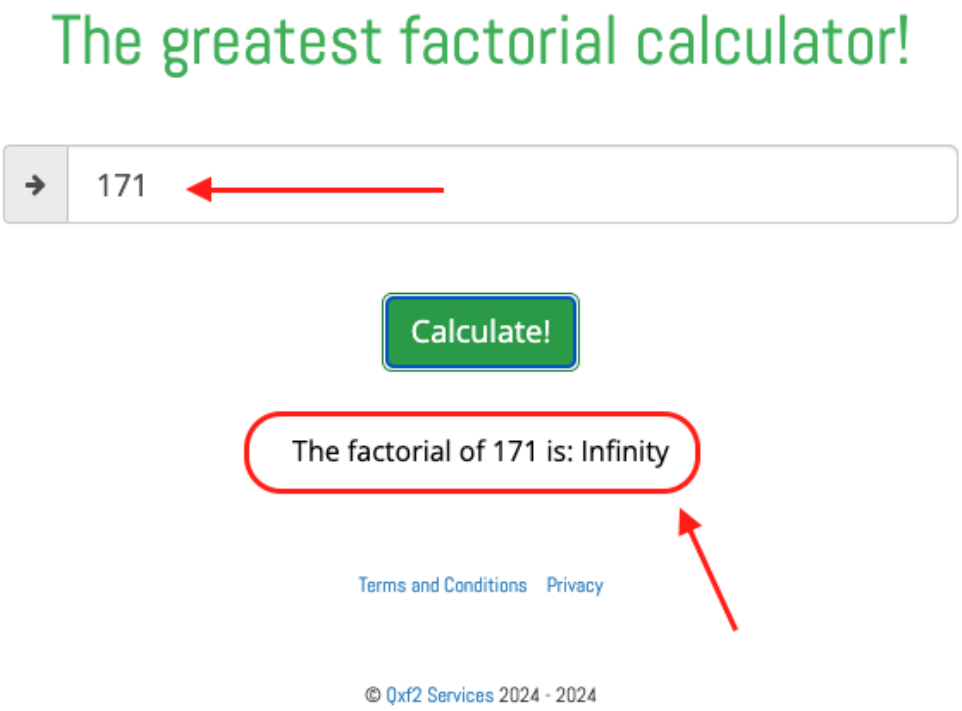
🐞 Bug: Incorrect Error Message for Numbers Larger than 170

Steps to Reproduce:

1. Insert a 3-digit number larger than 170 (e.g., 200) in the input area (don’t enter a 4 digit number).
2. Click on the "Calculate!" button.

Expected Result: An error message should be displayed, indicating that the calculator does not support numbers larger than 170.

Actual Result: The calculator displays an incorrect message, stating that the factorial is "Infinity".



🐞 Bug: Incorrect URL Extensions and Incorrect Message on Newly Opened Pages

Steps to Reproduce:

1. Click on the "Terms and Conditions" link.
2. Verify the URL extension and the message on the newly opened page.

Expected Result: Clicking on the "Terms and Conditions" link should open a new subpage with the URL extension /terms and display the corresponding terms and conditions document.

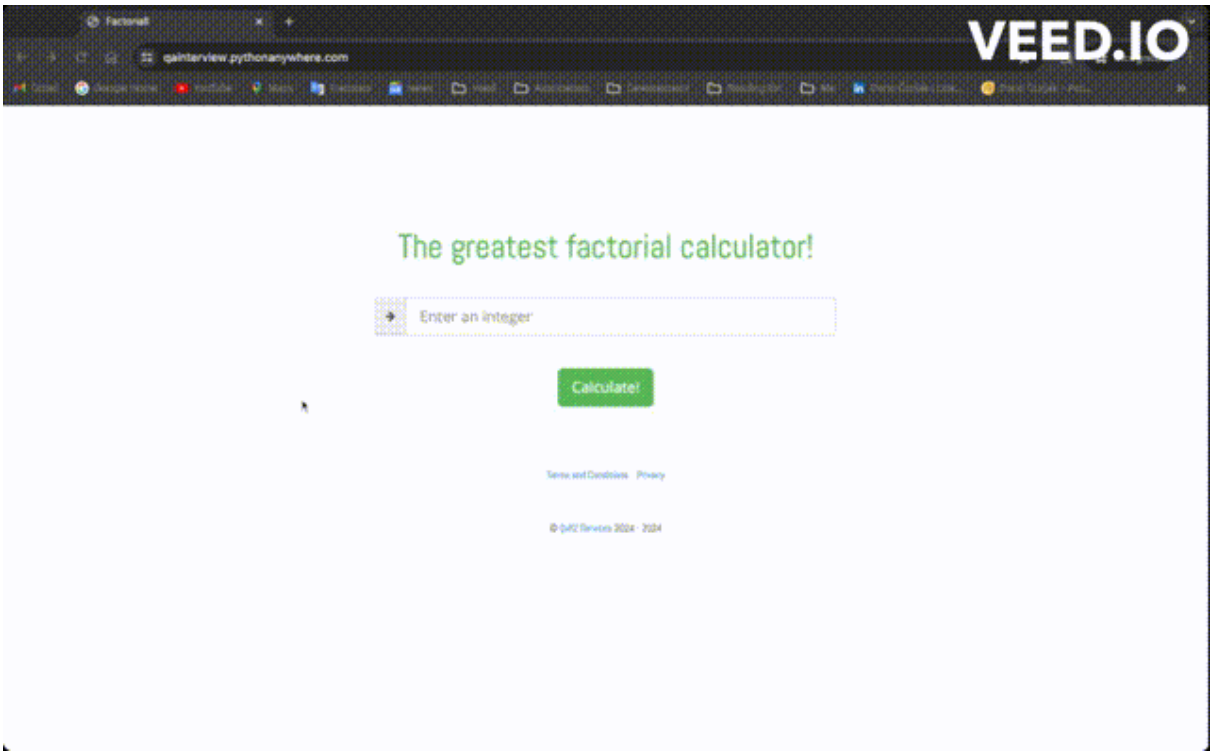
Actual Result: Clicking on the "Terms and Conditions" link opens a subpage with the URL extension /privacy and displays the message "This is the privacy document. We are not yet ready with it. Stay tuned!"

Steps to Reproduce:

1. Click on the "Privacy" link.
2. Verify the URL extension and the message on the newly opened page.

Expected Result: Clicking on the "Privacy" link should open a new subpage with the URL extension /privacy and display the corresponding privacy document.

Actual Result: Clicking on the "Privacy" link opens a new subpage with the URL extension /terms and displays the message "This is the terms and conditions document. We are not yet ready with it. Stay tuned!"

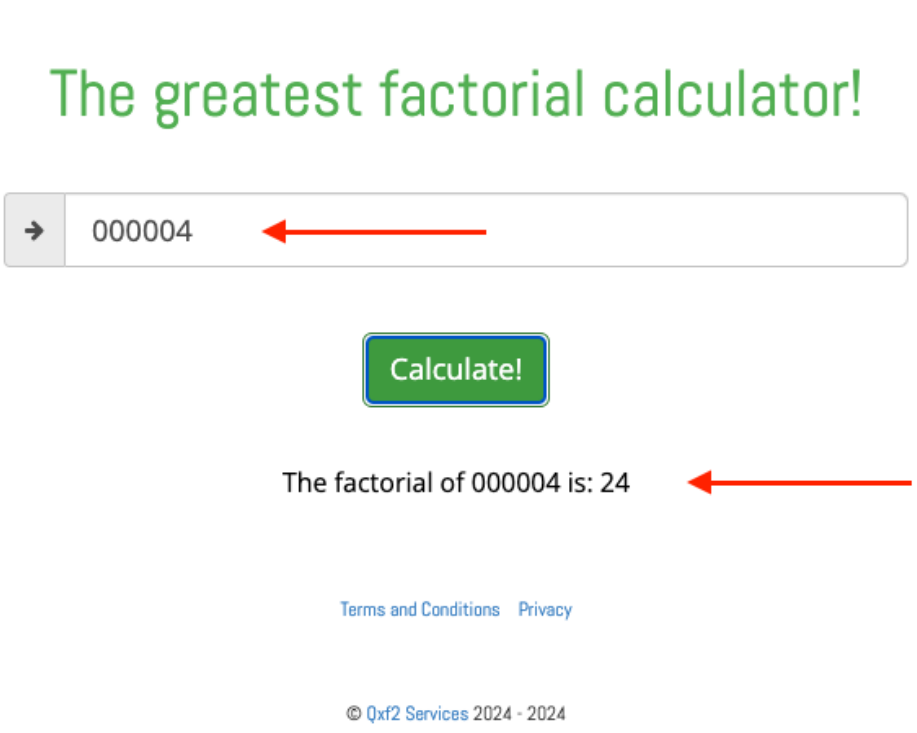


🚀 Improvement Suggestions 🚀

🧰 Issue: The calculator allows redundant prefix data

This one is not necessarily a bug, but more an observation and an opportunity to improve UX experience.

- **Steps to Reproduce**:
  - Insert redundant prefix data (e.g., "+4" or "0000004") in the input area.
  - Click on the "Calculate!" button.
- **Expected Result**: The calculator should reject redundant prefix data and display an error message.
- **Actual Result**: The calculator accepts redundant prefix data and provides the correct factorial value.



This document was created by Dario Ćurjak. 📝✨