

Test Cases (FRONT END)

1. Input Validation

Test Cases:

1.1 Scenario: When the user enters a non-numeric value (e.g., "A", "B").

The screenshot shows a web form with a title "input Number" and a "HISTO" button. The input field is empty. Below the input field is a purple "SUBMIT" button. A white error message box is displayed, stating "You can't put alphabet in input number". The message box has a blue "닫기" (Close) button in the bottom right corner.

1.2 Scenario: When the user enters special characters (e.g., "@", "#", "\$").

The screenshot shows a web form with a title "input Number" and a "HISTO" button. The input field is empty. Below the input field is a purple "SUBMIT" button. A white error message box is displayed, stating "You can't put alphabet in input number". The message box has a blue "닫기" (Close) button in the bottom right corner.

1.3 Scenario: When the user enters a number with multiple decimal points (e.g., "123.45.67").

The screenshot shows a web form with a title "input Number" and a "HISTO" button. The input field contains the text "123.4.44". Below the input field is a purple "SUBMIT" button. A white error message box is displayed, stating "You got one dot in your input text". The message box has a blue "닫기" (Close) button in the bottom right corner.

1.4 Scenario: When the user enters a number with more than 3 decimal places (e.g., "123.456").

The screenshot shows a web form with a title "input Number" and a "HISTO" button. The input field contains the text "123.45". Below the input field is a purple "SUBMIT" button. A white error message box is displayed, stating "Only two decimal places are not allowed". The message box has a blue "닫기" (Close) button in the bottom right corner.

1.5 Scenario: When the user enters a number with leading zeros (e.g., "00123.45").

The screenshot shows a web form with a title "input Number" and a "HISTO" button. The input field contains the text "00123.45". Below the input field is a purple "SUBMIT" button. Below the "SUBMIT" button is a section titled "RESULT" which displays the text "ONE HUNDRED AND TWENTY-THREE DOLLARS AND FORTY-FIVE CENTS".

1.6 Scenario: When the user enters a dot as the last character (e.g., "123.").

A screenshot of a web form with a text input field containing "123.". Below the input is a purple "SUBMIT" button. A white error message box is displayed over the button, containing the text "should remove . at the end". At the bottom of the form, the text "ONE HUNDRED AND T" is visible on the left, and a blue "닫기" (Close) button is on the right.

1.7 Scenario: When the user enters an empty value (e.g., "").

A screenshot of a web form with an empty text input field. Below the input is a purple "SUBMIT" button. A white error message box is displayed over the button, containing the text "Enter the Number!". At the bottom of the form, the text "ONE HUNDRED AND T" is visible on the left, and a blue "닫기" (Close) button is on the right.

1.8 Scenario: When the user pastes an invalid value.

A screenshot of a web form with a text input field containing a single vertical bar "|". Below the input is a purple "SUBMIT" button. A white error message box is displayed over the button, containing the text "You can't put alphabet in input number". At the bottom of the form, the text "ONE HUNDRED AND T" is visible on the left, and a blue "닫기" (Close) button is on the right.

Expected Behavior:

1. The system should validate and sanitize the input.
2. The submit button will be disabled or a warning message will be shown.

2. Range Limits:

Test Cases:

- 2.1 Scenario: When the user enters an extremely large number (more than 30 digits).
- 2.2 Scenario: When the user enters a very small decimal number (e.g., "0.01").
- 2.3 Scenario: When the user enters zero.
- 2.4 Scenario: When the user enters negative numbers.

Expected Behavior:

For scenario 2.1, the system should return a warning message.

A screenshot of a web form with a text input field filled with 30 '2' characters. Below the input is a purple "SUBMIT" button. A white error message box is displayed over the button, containing the text "Too big number". At the bottom of the form, the text "ONE HUNDRED AND T" is visible on the left, and a blue "닫기" (Close) button is on the right.

For scenario 2.2, the system should return "ZERO DOLLARS AND ONE CENT."

SUBMIT

RESULT

ZERO DOLLAR AND ONE CENT

For scenario 2.3, the system should return a warning message.

0

SUBMIT

0 is ZERO

닫기

For scenario 2.4, the system should return "You can't input letters as the number" because negative numbers are not allowed.

|

SUBMIT

You can't put alphabet in input number

닫기

3. Output Format:

Test Cases:

3.1 Scenario: When the input is a decimal number without a whole number part (e.g., ".45").

3.2 Scenario: When the input is a decimal number without a decimal part (e.g., "123.").

Expected Behavior:

For scenario 3.1, the system should return "should remove the dot at the start."

.

SUBMIT

. at the start should be removed

닫기

For scenario 3.2, the system should return "should remove the dot at the end."

123.

SUBMIT

should remove . at the end

닫기

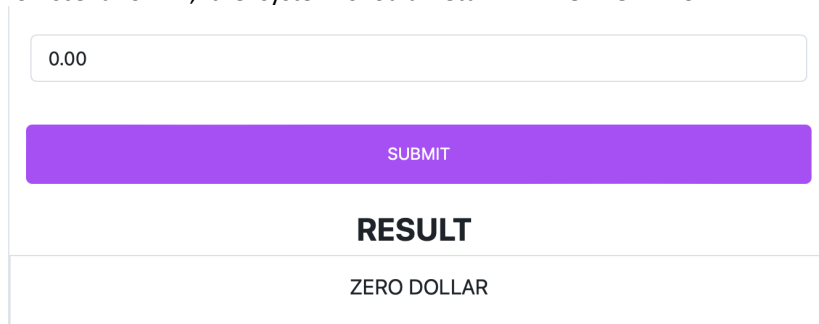
4. Edge Cases:

Test Cases:

4.1 Scenario: When the input consists of only decimal places without any significant digits (e.g., "0.00").

Expected Behavior:

For scenario 4.1, the system should return "ZERO DOLLARS."



0.00

SUBMIT

RESULT

ZERO DOLLAR

5. Usability:

Test Cases:

5.1 Scenario: Ensure the input box is easily identifiable.

5.2 Scenario: Ensure there's a clear call to action (e.g., a "SUBMIT" button).

5.3 Scenario: Ensure error messages are clear and helpful.

6. History:

Test Cases:

6.1 Scenario: After submitting valid input, ensure the input appears in the history field.

6.2 Scenario: After multiple valid inputs, ensure all are recorded in the history field in the correct order.

6.3 Scenario: Submitting invalid input should not be added to the history field.

7. Click History:

Test Cases:

7.1 Scenario: Click on an entry in the history field. The clicked entry should populate the input field.

7.2 Scenario: After populating the input field from history, clicking submit should display the same result as shown in the history.

8. Clear Button:

Test Cases:

8.1 Scenario: Click the clear button when the history is populated. The history field should be cleared.

8.2 Scenario: Click the clear button when the history is empty. Nothing should happen.

8.3 Scenario: After clearing history, enter and submit a valid number. The number should appear in the history, and previously cleared entries should not reappear.

9. Responsive Design:

Test Cases:

9.1. Test on different browsers (Chrome, Firefox, Safari) to ensure cross-browser compatibility.