**1. Overview**

This test plan verifies that the application correctly converts numeric input into its English word equivalent via a Razor Pages web interface.

**2. Test Scope**

* Functional testing of number-to-word logic
* Input validation for invalid entries
* UI behavior and form submission
* Edge case handling

**3. Test Objectives**

* Ensure all valid integers are converted accurately
* Reject invalid inputs gracefully
* Display friendly, readable outputs
* Maintain correct UI responsiveness

**4. Test Environment**

| **Component** | **Details** |
| --- | --- |
| App Framework | ASP.NET Core Razor Pages |
| Runtime | .NET 6 or .NET 7 |
| Testing Mode | Manual |
| Browser | Chrome, Edge, Safari |
| OS | macOS or Windows |

**5. Test Cases**

| **Test ID** | **Test Description** | **Input** | **Expected Output** |
| --- | --- | --- | --- |
| TC01 | Single digit | 5 | five |
| TC02 | Two-digit | 42 | forty-two |
| TC03 | Less than 20 | 17 | seventeen |
| TC04 | Three-digit | 123 | one hundred and twenty-three |
| TC05 | Thousands | 2024 | two thousand and twenty-four |
| TC06 | Millions | 1234567 | one million two hundred and thirty-four thousand five hundred and sixty-seven |
| TC07 | Zero | 0 | Zero. Plain and simple. |
| TC08 | Negative number | -99 | Negative ninety-nine |
| TC09 | Invalid input: letters | abc | [Oops] Umm, that doesn't look like a number. |
| TC10 | Invalid input: special chars | #$% | [Oops] Umm, that doesn't look like a number. |
| TC11 | Empty input | (blank) | [Oops] Umm, that doesn't look like a number. |
| TC12 | Large number | 9999999 | Correct word output (manual verification) |
| TC13 | UI loads correctly | (N/A) | Input field and button visible |
| TC14 | Result shown after submission | Any number | Result appears after form submission |
| TC15 | Trims spaces | " 45 " | forty-five |

**6. Regression Tests**

Repeat all test cases when:

* Adding decimal or currency support
* Changing number formatting logic

**7. Known Limitations**

* No decimal support
* No currency formatting (e.g., “dollars”)
* No handling for billions or larger without extending the logic

**8. Suggestions for Automation**

* Move SayItOutLoud() to a separate service class
* Add **xUnit/NUnit** unit tests
* Automate form submission tests