Test Plan: Number to Words Converter

1. Objectives

The purpose of this test plan is to validate the functionality, accuracy, and robustness of the Number to Words Converter application. Key objectives include:

- Verifying the accurate conversion of decimal numbers into English words formatted as currency.
- Ensuring correct handling of various edge cases, including large numbers, singular/plural distinctions, and invalid input.
- Confirming stability, error handling, and output formatting.

2. Scope

This test plan covers:

- **Unit Testing**: Testing individual functions within NumberToWordsService.
- **Integration Testing**: Ensuring proper interactions between different components (e.g., front end and back end).
- **Error Handling**: Checking that invalid inputs produce appropriate errors.

3. Test Approach

3.1 Unit Testing

Unit tests focus on the ConvertNumberToWords method in NumberToWordsService, verifying correct output for a range of inputs, including valid, invalid, edge cases, and boundary values.

3.2 Integration Testing

Integration testing ensures that the entire application workflow, from user input through output generation, functions correctly. These tests confirm the end-to-end functionality via manual verification on the user interface, accessed via the browser.

3.3 Error Handling and Edge Cases

Special tests focus on invalid inputs, handling of boundary cases, and formatting accuracy, ensuring robustness against potential user input errors.

4. Test Scenarios

Test ID	Test Scenario	Input	Expected Output	Result
1	Whole number without cents	250	TWO HUNDRED AND FIFTY DOLLARS	PASS/FAIL
2	Single digit dollar amount	7	SEVEN DOLLARS	PASS/FAIL
3	Tens place only	40	FORTY DOLLARS	PASS/FAIL

4	Exact hundred	500	FIVE HUNDRED DOLLARS	PASS/FAIL
5	Exact thousand	1000	ONE THOUSAND DOLLARS	PASS/FAIL
6	Small dollar amount with cents	1.25	ONE DOLLAR AND TWENTY-FIVE CENTS	PASS/FAIL
7	Very large number	9876543210	NINE BILLION EIGHT HUNDRED AND SEVENTY-SIX MILLION...	PASS/FAIL
8	Single cent	0.01	ZERO DOLLARS AND ONE CENT	PASS/FAIL
9	Dollar amount with max two decimal places	123456789.99	ONE HUNDRED AND TWENTY-THREE MILLION... AND NINETY-NINE CENTS	PASS/FAIL
10	More than two decimal places	123.456	Throws ArgumentException	PASS/FAIL
11	Negative dollar amount	-100	Throws ArgumentOutOfRangeException	PASS/FAIL

5. Testing Tools and Environment

- **Tools**: xUnit (for automated unit testing), Visual Studio or Visual Studio Code
- **Environment**: .NET 6 SDK, local development environment
- **Access**: Run the application via https://localhost: 5171 (or specified local URL)

6. Test Execution

Unit Test Execution

1. Use xUnit to run all tests in NumberToWordsServiceTests.cs.
2. Validate that each test produces the expected output as per the scenarios.

Integration Test Execution

1. Start the application and navigate to the specified localhost URL.
2. Enter test values in the application UI and verify the displayed output.
3. Confirm results against the expected outputs for each scenario.

Error Handling Execution

1. Input invalid values (e.g., numbers with more than two decimal places, negative values).
2. Confirm that the application throws the appropriate exceptions and displays error messages.