# Test Plan

## Introduction

The project covers a web server routine which involves taking a numerical input and outputs it’s corresponding word output. It uses streamlit for the web server routine and a string mapping algorithm which is stored in the helpers.py

## In-scope

1. All the UI related testing that ensures the smooth access and functionality of the website.
2. Sanity of the number to word conversion logic
3. Latency testing
4. User Experience
5. Regression Testing
6. Compatibility Testing

## Out of scope

1. Load testing
2. Performance Testing
3. UI-based automation
4. Unit Testing
5. Other sorts of non-functional testing

## Environment

1. The application must be tested on a pyton3 environment which is locally hosted on the users machine.
2. This web app can also be deployed on a server but that is out-of-scope of the project so we will be testing on the local environment

## Tools

1. Git
2. Github
3. Python3
4. Internet connection
5. Windows/Linux/MacOS
6. Streamlit (Python Package)
7. Web browser (Edge/Safari/Chrome/Firefox)
8. Test Management Tools (TestRail)
9. Defect tracking tools (Jira)

## Assumptions

It’s assumed that the user has followed the steps in the github readme to setup their environment and that environment is up and running at the time of testing.

## Expected Risk

1. Not using defect tracking tools can lead to confusion about the bug lifecycle of any bug raised during testing, it can be hard to update the status of open bugs manually while not using a specialised tool such as Jira.
2. Not having a test management tool can make it difficult to maintain a regression suite.
3. Unavailability of environment and resources can lead to delays in testing.

## Risk Mitigation

1. Using a tool like excel or spreadsheet/google sheets that allows the user to change/update the status of a bug in a columnar method can make it easier to see the overall status of open bugs in the unavailability of ticketing tools.
2. Using certain tools like sheets can help if multiple people are creating/executing test cases to maintain the latest test run status.
3. Ensuring that the environment is healthy/available at the time of sanity/functional testing.

## Entry Criteria

1. All P0 test cases must be enumerated by the test engineer and executed by the developer just before handover.
2. All implementation related to requirements is closed and test documents are ready and reviewed.
3. Test data should be ready.
4. The handover build/application should be ready.
5. Pull requests if any should be checked and done.
6. The necessary resources to test the application should be ready.

## Exit Criteria

1. All test cases should be executed.
2. All test cases should be passed except the ones that have been agreed upon.
3. There should be no blocker/major bugs open at the time of sign off.
4. The final build is stable.

## Sanity

### P0 test cases

These test cases are to be executed at the time of handover.

1. The application should load when the localhost url opens
2. The number input should only be numerical input and does not allow alphabetical input
3. The heading should show number to text
4. Does not allow special characters
5. The increment button increments number by 1
6. The decrement button subtracts number by 1
7. The numeric input in the text box is converted into words and displayed clearly below

This and other test functional areas should be tested to ensure the sanity of the application is good

## Latency testing

1. When the application prompts the user to convert the number the latency should not be extremely high.
2. The latency of the increment and decrement buttons should be stable and not high.
3. Other actions such as reloading the page should not take long to render the page.

## User Experience

Overall user experience should be smooth and should not provide any hurdles to user when using.

## Regression Testing

For any future releases/updates in the application, a P0 test suite should be referred to and executed to ensure there are no regressions.

## Compatibility Testing

This can be tested on different OS: Windows, Mac, Linux, and different browsers such as Chrome, Firefox, and Safari.

## Useful Links

1. Link to Repository: <https://github.com/sudislife/TechOne>
2. Test Suite: (TBD)
3. Defect Tracking: (TBD)