

Test Report of Reverse Text Generator

Contents

1. Purpose	2
2. Application Overview	2
3. Testing Scope	2
4. Metrics	2
5. Types of testing performed	3
6. Test Environment & Tools	3
7. Lessons Learnt	3
8. Recommendations	3
9. Best Practices	4
10. Exit Criteria	4
11. Conclusion/Sign Off	4
12. Definitions, Acronyms, and Abbreviations	4

1. Purpose

This document explains the various activities performed as part of Testing of 'Reverse Text Generator'.

2. Application Overview

'Reverse Text Generator' is a web based function which provides 3 methods to convert the input text into another format . Each function will read the content from the "Input" text-field, convert into a new format and then finally print the output in the "Output" text-field.

"Reverse Text" will reverse the entire text string.

"Reverse Wording" will reverse the entire word order without reversing the word lettering

"Flip the sentence" will reverse the word order of each text line.

3. Testing Scope

a) In Scope

Automated Basic Functional Testing for the following functions are in Scope of Testing

- Reverse Text
- Reverse Wording
- Flip the sentence

b) Out of Scope

User Experience Testing was not done for this application.

Security Testing was not done for this application.

Performance Testing was not done for this application.

4. Metrics

d) No. of test cases planned vs executed

e) No. of test cases passed/failed

Test cases planned	Test cases executed	TCs Pass	Tcs Failed
51	51	12	39

f) No. of defects identified and their Status & Severity

Defect description	Sevierity	Visibility	Priority
1. Reverse text return empty string	most severe	everyone will see	must fix
2. Reverse wording add extra at the beginning of the sentence	most severe	everyone will see	must fix
3. Flip sentence doesn't support multiple sentences	most severe	everyone will see	must fix
4. Flip one sentence of string with only one word returns empty string	most severe	most people will see	must fix
5. Flip one sentence of string with odd number of words will omit the middle word	most severe	most people will see	must fix

5. Types of testing performed

a) Smoke Testing

This testing was to make sure the major functionalities are working fine, Build can be accepted and Testing can start.

b) Functionality Testing

- Different kinds of scenarios were tested to make sure various input datas to the application works as intended without any errors.

c) Automation of Basic Functionality Testing

- User Python language and Selenium framework to automate Basic Functionality Testing

6. Test Environment & Tools

- a) Application to Test: FusionSportQA_testproject.html
- b) Testing Environment: Mac OS 10.13, Chrome 62
- c) Testing Tools: Python 2.7, Eclipse, Selenium
- d) Guide to setup test environment: <http://selenium-python.readthedocs.io/installation.html>

7. Lessons Learnt

N/A

8. Recommendations

- a) Need to improve encoding support in the automation suite.
- b) Need to add different encoding testing in the automation suite.

9. Best Practices

- a) **Automated basic functionalities testing of this application.**

10. Exit Criteria

- a) All test cases should be executed - **Yes**
- b) All defects in Critical, Major, Medium severity should be verified and closed - **No**
- c) Any open defects in trivial severity - **No**

11. Conclusion/Sign Off

Many major defects exist, this application is just a draft, should not go live until exit criteria met.

12. Definitions, Acronyms, and Abbreviations

N/A