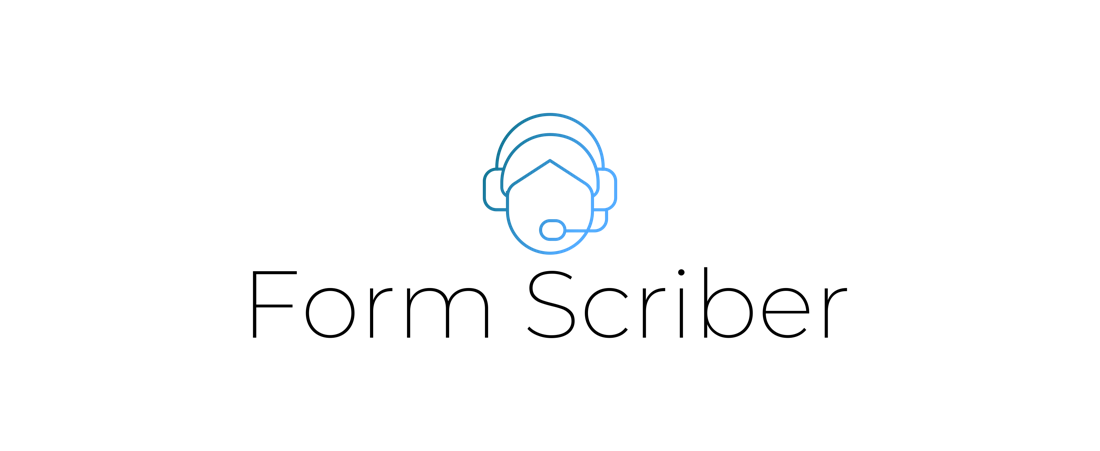
Form Scriber Test Report: Dialogflow Team



Version 1.0

Stefon Williams

Caleb Crickette

Thomas Barton

Eugene Kim

Nicholas Ballo

Abdul Kamara

Sohail Sobhani

Geoffrey Dean

Jean-nae Dedrick

Anthony Cedeno

SWEN 670 Software Engineering Project

University of Maryland Global Campus

**REVISION HISTORY**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description** |
| 3/30/2021 | 1.0 | Initial Test Report |

**Table of Contents**

[1. Introduction 5](#_Toc67935823)

[1.1 Purpose 5](#_Toc67935824)

[1.2 Application Overview 5](#_Toc67935825)

[1.3 Testing Strategy 5](#_Toc67935826)

[2.1. Unit 6](#_Toc67935827)

[2.2. Integration 6](#_Toc67935828)

[2.3. Regression 6](#_Toc67935829)

[2.4. End-to-End 6](#_Toc67935830)

[2.5. Website Performance Testing 6](#_Toc67935831)

[3. Items not tested 7](#_Toc67935832)

[4. Metrics and Testing 7](#_Toc67935833)

[4.1 Manual Test Suite 7](#_Toc67935834)

[Test Case 01: Form Scriber initiates with a new form to fill out. 8](#_Toc67935835)

[Test Case 02: Form Scriber identifies a google document to fill out 9](#_Toc67935836)

[Test Case 03: Form Scriber identifies an invalid URL 10](#_Toc67935837)

[Test Case 04: Form Scriber checks if google doc has appropriate share settings. 11](#_Toc67935838)

[Test Case 05: User provides a value for an unfilled field in the specified form 12](#_Toc67935839)

[Test Case 06: User provides a non-existing field in the specified form 13](#_Toc67935840)

[Test Case 07: User requests to print a specified form 14](#_Toc67935841)

[Test Case 08: Attempt to download a generated Google Doc file from a browser 15](#_Toc67935842)

[Test Case 09: User issues a request to start over. 16](#_Toc67935843)

[Test Case 10: Validate if Form Scriber is able to process a newly specified form after a start over command. 17](#_Toc67935844)

[Note: The succeeding test cases pertain to the form scriber web user interface. 19](#_Toc67935845)

[Test Case 11: Shared Google Drive Folder and Docs 19](#_Toc67935846)

[Test Case 12: Un-shared Google Drive Folder and Docs. 21](#_Toc67935847)

[Test Case 13: Sharing a Doc with valid fields 22](#_Toc67935848)

[Test Case 14: test FAQ contextual searching 23](#_Toc67935849)

[5. Defects Identified 24](#_Toc67935850)

[6. Exit Criteria 25](#_Toc67935851)

[7. Website Performance Report 26](#_Toc67935852)

[8. Conclusion/Sign Off 28](#_Toc67935853)

# Introduction

## Purpose

This document is being presented as part of the Quality Control requirements for the application: Form Scriber. The aim is to present the application’s testing activities, along with supplementary tesing information for formscriber.com’s website performance. Like any test report, the goal of this document is also to present fundamental testing metrics based on executed test cases.

In this document, we will delve into introductory aspects like getting an overview of the Form Scriber, its testing strategy, testing limitations and its association to common types of testing. After which, the performed test cases, along with results are presented. Defects and corresponding corrections are presented next. Lastly, a fundamental presentation of metrics derived from the tests performed.

## 1.2 Application Overview

The Form Scriber application project was created by Dr. Mir Mohammed Assadullah. This project is otherwise known as the class’ Capstone Project. The Dialogflow team, Mobile team, and DevSecOps teams were created to collaborate and complete the project. The prospective customers that may use the product delivered by this project are any general professional whom fills out forms after interacting with clients and/or customers.

The main activities of the Form Scriber application are the following, where the last bullet point is the desired output: *a properly-filled form*:

* To have any professional create a customized form according to a specified format in Google Docs.
* To use and fill out the customized Google Doc form through voice or text inputs.
* To be able to acquire help and guidance in using Form Scriber.
* To store and print the filled-out Google Doc form.

## 1.3 Testing Strategy

This document will delve around the three activities above and aims to present results based on provided and pre-defined test cases that aims to test adherence to specific application requirements.   
  
 **1.4 Testing Limitations**

There is a feature of Form Scriber that offers flexibility through form creation and customization. This feature entails that fields also need to be dynamically-defined. Due to the nature of this feature, the following limitations in input validation exist, and may be considered as future considerations for further development:

* All inputs cannot be tested, including all potential inputs derived from machine learning
* All the combination of inputs cannot be tested.
* We are unable to test all possible paths.
* The following specific input properties such as:
  + Input being required or mandatory
  + Expected Text Format
    - For example, the way a field will appear in the form cannot be validated and predefined:   
      Hence, if a person says *5 feet and 8 inches for* height, and height is a custom/dynamic field, there’s no way to specify that the text format in the form will be 5’ 8”.
  + Order of fields in form.
  + Establish standardized / precise field values
    - For example, in a web form, fields that require uniform values use drop down lists for user selection. This can actually be achieved through further setup on intent, but could not be tested in Form Scriber’s dynamic form creation feature.

1. **The Form Scriber & Types of Testing**

## Unit

Because of limited time the Dialogflow development team wrote unit tests to cover Google DOC parsing that is one of the main functions of the system. These unit tests provide 35% code coverage.

## Integration

Integration Testing will involve testing all developed REST services in tandem with the GCP Dialogflow service. The Dialogflow team has focused on integration testing wherein corresponding test cases are executed manually.

## Regression

Identified defects are reported to development team and are evaluated for correction. When an identified defect has been corrected, regression testing is performed for validation. Therefore, the corresponding test case that associates with a defect is executed once again, and results are reported again as well.

## End-to-End

End-to-end testing will not be performed by the Dialogflow team and will not be part of this Test Report. The mobile team will embed the Form Scriber to the mobile user interface they are developing. Hence, all test cases that will be presented in the succeeding section will, in essence, function the same when the system is embedded. The mobile team, will present test cases that demonstrate an embedded Form Scriber behaves as expected in a mobile app.

## Website Performance Testing

One of Dialogflow team’s product is to provide a web user interface provide help access to users, and aid giving intuitive user experience with Google Drive integration. Hence, to provide this service with optimum performance, the team uses gtmetrix.com to gain substantial report for formsriber.com.

.

# Items not tested

The following items are not in scope to be testing by the Dialog Flow team.

* Network Security
* System deployment
* Load and stress testing

# Metrics and Testing

## 4.1 Manual Test Suite

The following test cases are performed via Dialogflow team’s web-based User Interface: http://www.formscriber.com

| **Manual Test Suite Execution Details** | |
| --- | --- |
| Test Started | 3/28/2021 |
| Test End | 3/28/2021 |
| Test Conductors | Anthony Cedeno and Sohail Sobhani |
| Number of Test Cases | 14 |
| Number of failures | 1 |

Table 4: Manual Test Suite Execution:

## Test Case 01: Form Scriber initiates with a new form to fill out.

|  |  |
| --- | --- |
| **Description:** | When Form Scriber is initiated, it is always the very first step to fill-out a form and make it ready for processing. |
| **Actor:** | A user will speak or type a command. |
| **Trigger** | The user will activate the mic button and say “start a new document”. Triggered also by typing “start a new document” and pressing <Enter> |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. |
| **Post Condition** | System will state a welcome note and prompt to take a predefined Google Doc URL. |
| **Result** | Passed |

## Test Case 02: Form Scriber identifies a google document to fill out

|  |  |
| --- | --- |
| **Description:** | Based on guidelines in creating a document template, Form Scriber will take a Google Document form. |
| **Actor:** | The user will paste the URL |
| **Trigger** | The user presses <Enter> after pasting the Google Document URL |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. * System state: Post Condition of Test Case 01. |
| **Post Condition** | Dialogflow will recognize user’s intent and respond appropriately or send request to webhook depending on the message.  In this test case, Form Scriber is able to recognize a well-formed Google Doc, and has identified fields that need to be filled.    **Google Doc Status**: Notice how unfilled fields: last-name and date-of-birth were identified. |
| **Result** | Passed |

## Test Case 03: Form Scriber identifies an invalid URL

|  |  |
| --- | --- |
| **Description:** | Based on guidelines in creating a document template, Form Scriber will take a Google Document form. |
| **Actor:** | The user will paste the URL that is invalid  (e.g. in this particular test, test.com was entered) |
| **Trigger** | The user presses <Enter> after pasting the Google Document URL |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. |
| **Post Condition** | Dialogflow will recognize user’s intent and respond appropriately or send request to webhook depending on the message.  In this test case, the URL entered is not a valid Google Document URL, as it first checks the domain part of the URL. |
| **Result** | Passed |

## Test Case 04: Form Scriber checks if google doc has appropriate share settings.

|  |  |
| --- | --- |
| **Description:** | Based on guidelines in creating a document template, Form Scriber will take a Google Document form. The Google Document is not shared to [formbot@form-bot-1577a.iam.gserviceaccount.com](mailto:formbot@form-bot-1577a.iam.gserviceaccount.com) |
| **Actor:** | The user will paste the Google Document URL |
| **Trigger** | The user presses <Enter> after pasting the Google Document URL |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. |
| **Post Condition** | Dialogflow will recognize user’s intent and respond appropriately or send request to webhook depending on the message.  In this test case, Form Scriber raised an error but it’s not appropriate to the situation. |
| **Result** | Passed |

## 

## Test Case 05: User provides a value for an unfilled field in the specified form

|  |  |
| --- | --- |
| **Description:** | User will provide an existing unfilled field. |
| **Actor:** | The user says or types the form field and value as it gets prompted by the system. |
| **Trigger** | The user presses <Enter> or completes each voice command. |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. * System state: Post Condition of Test Case 02: Form Scriber identifies a google document to fill out |
| **Post Condition** | Dialogflow will recognize user’s intent and respond appropriately or send request to webhook depending on the message.  In this test case, user states date of birth will be filled, system asks for it, then user states its value.  **Google Doc Status:** Date of Birth filled with user’s entry |
| **Result** | Passed |

## Test Case 06: User provides a non-existing field in the specified form

|  |  |
| --- | --- |
| **Description:** | User will provide a non-existing field. |
| **Actor:** | The user says or types the form filed to fill out. |
| **Trigger** | The user presses <Enter> or completes each voice command. |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. * System state: Post Condition of Test Case 02: Form Scriber identifies a google document to fill out |
| **Post Condition** | Dialogflow will recognize user’s intent and respond appropriately or send request to webhook depending on the message.  In this case, Form Scriber somehow recognizes that a non-existing field, it should have raised that the field is non-existent.  **Google Doc:** Notice Height is not an existing field |
| **Result** | Failed |

## Test Case 07: User requests to print a specified form

|  |  |
| --- | --- |
| **Description:** | User will issue a command to print a specified Google Document Form. |
| **Actor:** | The user says or types the command *print*. |
| **Trigger** | The user presses <Enter> or completes the voice command. |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. * System state: Post Condition of Test Case 02: Form Scriber identifies a google document to fill out |
| **Post Condition** | Dialogflow will recognize user’s intent and respond appropriately or send request to webhook depending on the message.  In this test case, Form Scriber recognized the print command, and issues a link of the PDF version of the specified Google Doc Form. |
| **Result** | Passed |

## Test Case 08: Attempt to download a generated Google Doc file from a browser

|  |  |
| --- | --- |
| **Description:** | User will cut and paste the URL of a generated Google Doc file in PDF format. |
| **Actor:** | User will cut and paste the URL of a generated Google Doc file in PDF format. |
| **Trigger** | The user presses <Enter> in the browser address bar. |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. * System state: Post Condition of Test Case 08: user prints a form |
| **Post Condition** | Dialogflow will recognize user’s intent and respond appropriately or send request to webhook depending on the message.  In this test case, user will cut and paste the URL of the Google Document in a browser’s address bar.    As seen below, the PDF file loaded and is ready to be printed. |
| **Result** | Passed |

## Test Case 09: User issues a request to start over.

|  |  |
| --- | --- |
| **Description:** | User issues a command to start over. This enables user to start off a new session and indicate a new Google Document Form to fill out. |
| **Actor:** | User type of say “Start Over” |
| **Trigger** | The user presses <Enter> when typing, or just stops speaking when using voice commands. |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. * System state: any state. |
| **Post Condition** | Dialogflow will recognize user’s intent and respond appropriately or send request to webhook depending on the message.  In this test case, user issues a command to start over and it prompted that it is ready to process an entirely new form. |
| **Result** | Passed |

## Test Case 10: Validate if Form Scriber is able to process a newly specified form after a start over command.

|  |  |
| --- | --- |
| **Description:** | Based on guidelines in creating a document template, Form Scriber will take a new Google Document form. |
| **Actor:** | The user will paste the URL |
| **Trigger** | The user presses <Enter> after pasting the Google Document URL |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. * System state: Post Condition of Test Case 08, user issues a request to start over. |
| **Post Condition** | Dialogflow will recognize user’s intent and respond appropriately or send request to webhook depending on the message.  In this test case, Form Scriber is able to recognize a well-formed Google Doc, and has identified fields that need to be filled.    **Google Doc Status**: Notice that this is a completely new form with new fields: Gender and Age. |
| **Result** | Passed |

## Note: The succeeding test cases pertain to the form scriber web user interface.

## Test Case 11: Shared Google Drive Folder and Docs

|  |  |
| --- | --- |
| **Description:** | A user enters a Google Driver folder URL that has a document and it’s shared with Form Scriber. |
| **Actor:** | Navigate to Google Drive.  Create a new folder.  Right click on the folder and click share.  Add Form Scriber email as editor.  Create a new DOC in the folder.  Go to “https://formscriber.com”.  Go to “Form Scriber” Page.  Enter your Google Drive Folder URL and click ‘submit. |
| **Trigger** | The user presses <Enter> after pasting the Google Drive URL |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. * System state: Post Condition of Test Case 08, user issues a request to start over. |
| **Post Condition** | List of Templates are displayed. |
| **Result** | Passed |

## Test Case 12: Un-shared Google Drive Folder and Docs.

|  |  |
| --- | --- |
| **Description:** | A user has a Google Driver folder that has a document and it’s NOT shared with Form Scriber. |
| **Actor:** | Navigate to Google Drive.  Create a new folder.  Create a new DOC in the folder.  Go to “https://formscriber.com”.  Go to “Form Scriber” Page.  Enter your Google Drive Folder URL and click ‘submit. |
| **Trigger** | The user presses <Enter> after pasting the Google Drive URL |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. |
| **Post Condition** | An empty list should be displayed. |
| **Result** | Passed |

## Test Case 13: Sharing a Doc with valid fields

|  |  |
| --- | --- |
| **Description:** | A user has a Google Drive folder that has a document and it’s shared with Form Scriber. |
| **Actor:** | Go to “https://formscriber.com”.  Go to “Form Scriber” Page.  Enter the Google Drive Folder URL that contains the document  Enter the Google Drive Folder URL and click ‘submit.  Find the document in the list and select on UTL button.  In the Form-Bot chat say “Start a new report”.  After bot response, paste the URL to the document and press enter |
| **Trigger** | The user types or says “Start a New Report” |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. |
| **Post Condition** | Form-Bot should respond back with success message and a list of 4 fields that found on the document. |
| **Result** | Passed |

## Test Case 14: test FAQ contextual searching

|  |  |
| --- | --- |
| **Description:** | Test if a user can search the formsriber.com’s FAQs through keywords. |
| **Actor:** | User types a keyword to search in FAQs. |
| **Trigger** | The user types a keyword. This function executes as user types (event is on-key-press) |
| **Precondition:** | The system should have:   * Internet Access * FormScriber.com loaded with Dialogflow chat window ready for input. |
| **Post Condition** | Form Scriber shall load the FAQ that matches the keyword.   In this example, the user types print and the How-to for printing a form appears. |
| **Result** | Passed |

# Defects Identified

**Definition of a defect**  
A *defect* is defined as a system error that doesn't allow the intended action to be completed, and are evaluated in terms of its severity.

In this particular test report, the defects identified have severity level of low because the result of their existence does not interrupt the normal execution of Form Scriber. Table 6 below describes in detail the description of each defect identified as well as the current status of each defect.  
  
Table 5 below lists all defects by severity that were found during testing of Form Scriber.

|  | **Critical** | **Medium** | **Low** | **Total** |
| --- | --- | --- | --- | --- |
| **Identified** | 0 | 0 | 2 | 2 |
| **Resolved** | 0 | 0 | 1 | 1 |
| **Total Open** | 0 | 0 | 1 | 1 |

Table 5: Defect Matrix

Table 5.1 below, describes the defect categorization in this Test Report.

|  |  |
| --- | --- |
| **Severity Level of Defects** | |
| **Critical** | Errors that result in complete failure and interrupts the normal operation of the system. |
| **Medium** | Errors that result in a temporary system failure, but other functionalities remain operable |
| **Low** | An expected action did not happen but system remains operable. For example, when an error is raised but an incorrect error message is presented. |

Table 5.1: Severity Level of Defects

Table 6 below, provides the description of each defect identified, resolution, as well as the current status of each defect.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Severity** | **Description** | **Regression Testing** **Performed?** | **Current Status** |
| 1 | Low | Test Case 04: When detecting a Google Document share setting, Form Scriber raised an error but it’s not appropriate to the situation.  Resolution:  *Regression Testing Result: Passed* Form Scriber now raises the following error that the system has no write access to the specified Google Document. | Yes | Resolved |
| 2 | Low | Test Case 06: User provides a non-existing field in the specified form.   Form Scriber somehow recognizes that a non-existing field, it should have raised that the field is non-existent.  Resolution:  Pending, Development team evaluated that the minor defect needs correction due to time constraints. | No | Open |

Table 6: Defect Description and Status

# Exit Criteria

The completion of testing is determined by the following conditions:

* Testing of all items in scope has been completed
* All test cases have been executed
* All defects have been reported to Development team for evaluation and correction
* All minor defects have been documented
* No additional test cases are identified by the team

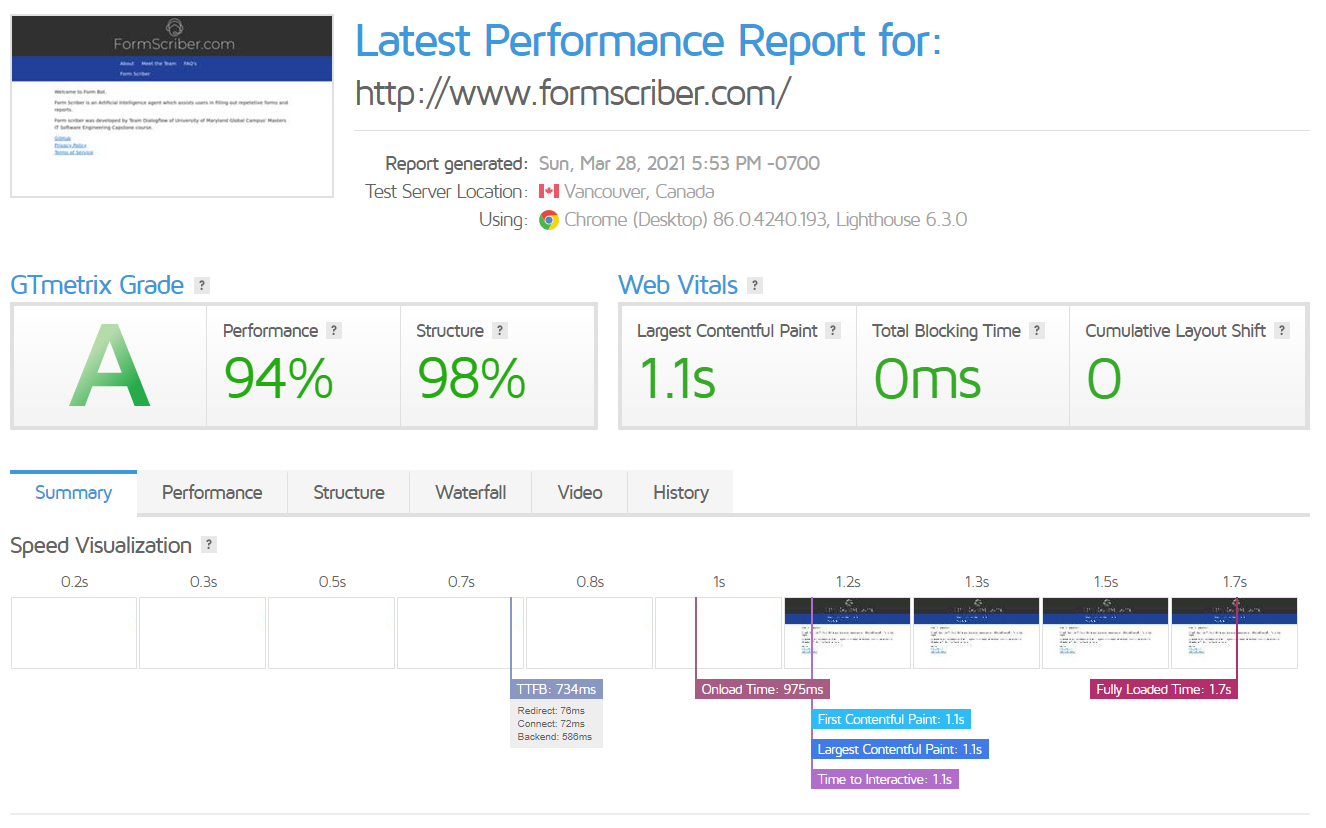
# 7. Website Performance Report

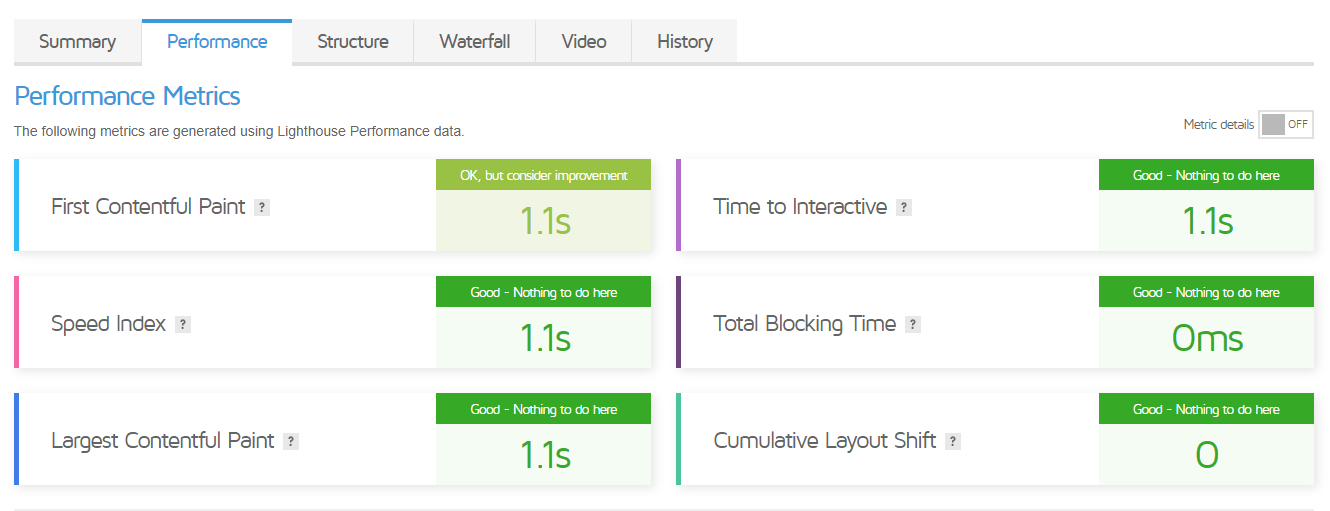
Dialogflow team’s website, formscriber.com, has undergone a website performance test from gtmetrix.com and obtained a grade of A from gtmetrix.com. According to the website, this grade is a weighted average of the two new percentage-based scores, namely:

* The Performance Score (70%)
* The Structure Score (30%)

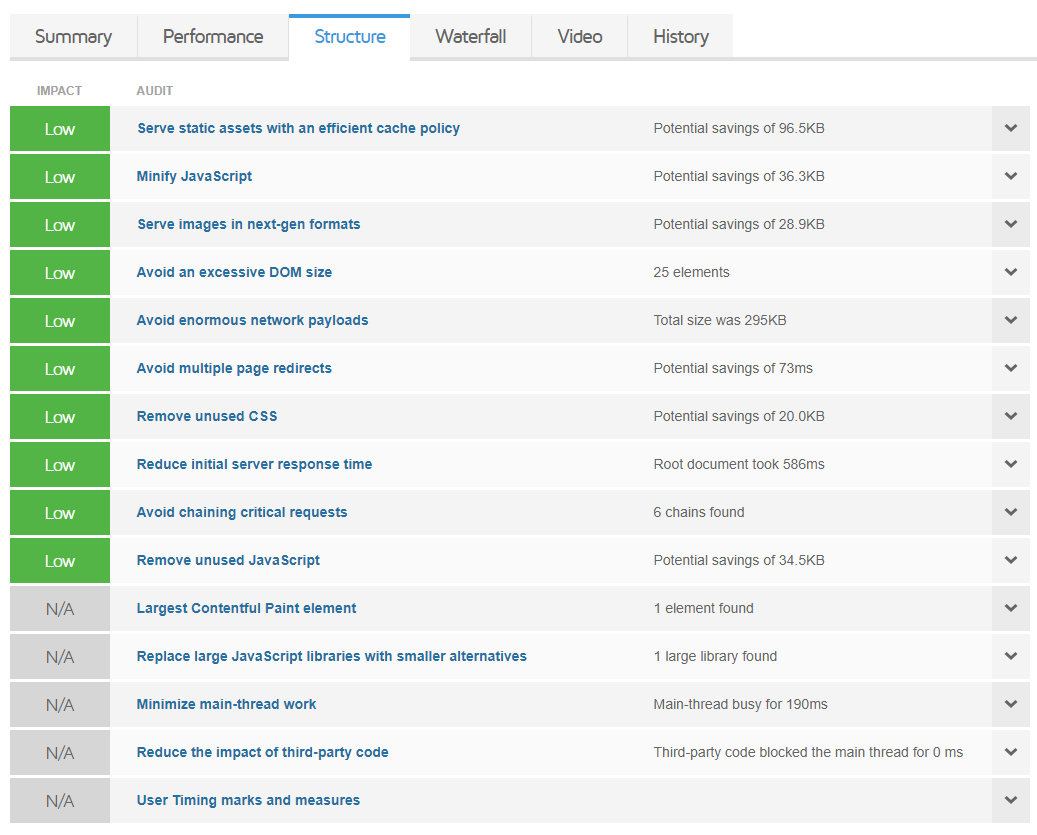
These scores are best practices established by Google to help build websites for optimal front-end performance. Each audit is assessed based on the website’s adherence to them and ordered by the most likely impact to your page's performance. (gtmetrix.com, n.d.)

Figures 1 to 3 represents the test reports obtained from gtmetrix.com

  
*Figure 1*. formscriber.com website performance summary (Source: gtmetrix.com)

**Performance Metrics Details**  


*Figure 2*. formscriber.com website performance metrics (Source: gtmetrix.com)

**Structure Metrics Details**  


*Figure 3*. formscriber.com website structure metrics (Source: gtmetrix.com)

# 8. Conclusion/Sign Off

| **Name** | **Signature** | **Date** |
| --- | --- | --- |
| Professor  Dr. Mir Assadullah |  |  |
| Form Scriber Dialog Team Project Manager  Stefon Williams |  |  |

Table 7: Sign Off