# W13-011
# Project Proposal

Max Chung

A00743631

COMP 8045 (9 Credits)

# Table of Contents

## Abstract

Online decision making tools allows project managers to choose their plan of action. The currently available tools are hard-coded and do not allow the administrator to easily customize, maintain, and monitor them. This project will create a customizable survey that address these needs.

## Introduction

Community Solutions needs a custom made online decision-making tool to help evaluators determine the evaluation approach they should use based on answering Yes or No questions on questions regarding background factors such as the type of program, evaluator role, and type of participants.

A questionnaire similar to the format used by the ARECCI Ethics Screening Tool needs to be created with customizable sections, questions, and recommendations based on question responses.

The ARECCI Ethics Screening Tool can be accessed from the link below.

http://www.aihealthsolutions.ca/arecci/screening

This project will further enhance my expertise in my chosen specialization because it will allow me to apply my skills in entity-relationship (EER) modeling, data normalization, and creating schemas from EER models. This project will also allow me to do wireframes, visual design, templates, automated cross-browser testing, and API calls, which is vital in web development.

The following images are screen captures taken from the above website and will be used as a reference when creating the questionnaire.

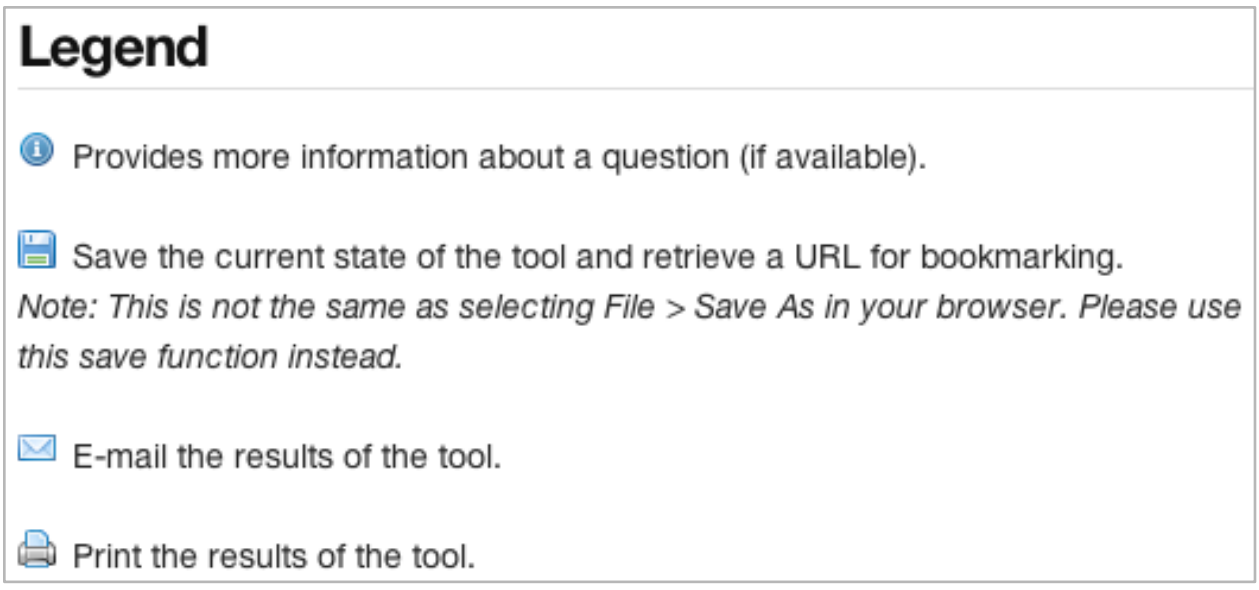



**Figure 1 - Legend shown on the reference website**


**Figure 2 - Question with question details shown on the reference website**

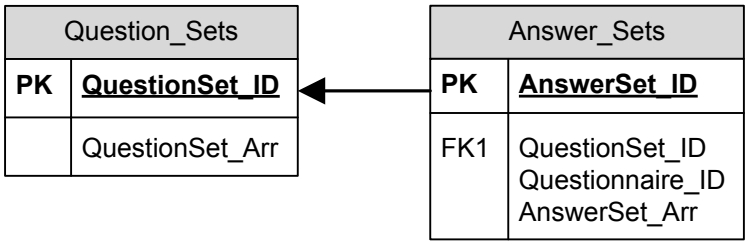

**Figure 3 - Suggestion based on questionnaire responses shown on the reference website**

## Methodology

This project will undergo the following phases:

- Gather requirements
- Create proposal
- Create wireframes
- Do visual design
- Create database schema
- Setup development environment
- Create page templates
- Hookup backend
- Enter initial data
- Testing and bug fixes
- Do automated cross-browser testing
- Do manual browser compatibility checks with beta site
- Deploy on live site
- Create user documentation
- Obtain final client signoff

## Database Schema




### Questions_Sets

QuestionSet_ID (PK AUTO_INCREMENT)

QuestionSet_Arr (LONGTEXT two dimensional array storing questions and descriptions separated by section)

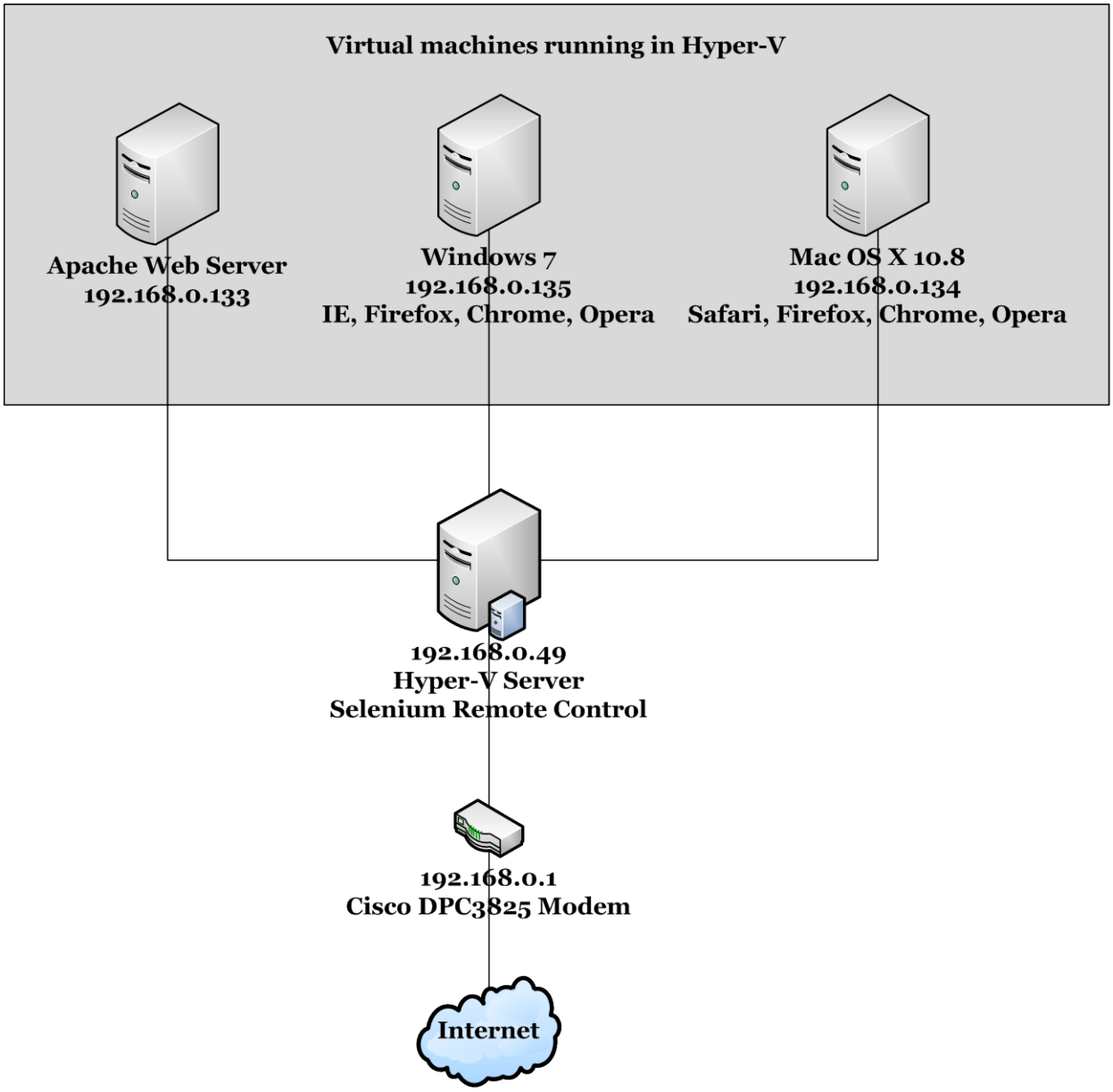
### Answer_Sets

AnswerSet_ID (PK AUTO_INCREMENT)

Questionnaire_ID (md5 of PHP microtime)

Answer_Arr (LONGTEXT Two dimensional array storing question answers separated by section)

**Network Diagram**




## Technologies

The questionnaire and admin site will be coded in HTML and CSS. JavaScript (jQuery) will be used for form validation, dynamically showing each section of the questionnaire, and reordering/deleting questionnaire sections and questions. Google Analytics will be used to gather statistics on the site's users and the data will be presented graphically using Google Charts. FPDF will be used to generate a PDF version of the questionnaire. Email styling will be handled using HTML and CSS. The questionnaire data will be stored in a MySQL database and the data will be written to the database and retrieved using PHP. Everything will be hosted on an Apache web server.

Usability studies will also be conducted using Usabilla and OpenHallway.

The automated cross-browser test case scripts will be created using Selenium IDE by recording the steps taken to complete each task. The website can then be tested concurrently from multiple browsers using Selenium Remote Control.

## Innovation

In this project, I will learn how to perform automated cross-browser testing, perform usability studies, do revision control, and make API calls.

### Usability Testing

In the traditional waterfall development approach, users do not have a chance of give feedback to the developers about a product until the very end when it is difficult and costly to make changes. This results in a product that does not satisfy a user's needs and will likely not be used hence being a waste of time and resources.

Instead of taking the waterfall approach, we should use the agile methodology, where developers can receive feedback from users early and make changes to the software iteratively. It's been found that instead of showing a fully built product to users, conceptual design mock-ups are often sufficient for the users to evaluate whether the product will meet their needs. A mock-up is a rough sketch of how a user interface will look and outlines the workflow. Changes can then be quickly made to the mock-ups according to user feedback and then be tested again. This iterative process results in a product that users will be more satisfied with.

To test the mock-ups, we are going to use Usabilla. In Usabilla, we upload screenshots of the mock-ups and pair each screenshot with a series of questions. Users will then be asked to click on the part of the mock-up that they think satisfies the question. For example, a user may be asked where they will click to get more information regarding a question. Users can also provide comments on individual screens. After multiple users have completed the mock-up testing, a heat map can be generated to show where users clicked to answer the questions to discover problematic areas within the tool. Other feedback that can be collected include rating their satisfaction of the tool, the ease of use, the likelihood of them using the tool again, on a scale of 1 to 10. Usabilla also measures the time it took users to answer each question, and the total amount of time to complete all the questions. After we make changes to the mock-ups, we can ask the users the same set of questions again and compare the statistics of the old and new mock-ups to see if we have effectively address usability issues.

While testing mock-ups give us a good approximation of how user friendly our software might be, we should still conduct a final usability study with the actual tool to confirm this. To test the tool, we are going to use OpenHallway. We will ask the users to complete the entire process from beginning to end. OpenHallway can record every action that the users take, and the amount of time it takes them to complete tasks. At the end, we will ask users to rate the tool again from 1 to 10 on their satisfaction of the tool, ease of use, the likelihood of them using the tool again. Hopefully these set of results will be comparable to those taken from the mock-up tests. Time permitting, should there be any issues that were not address, we will tweak the tool.

### Automated Cross-Browser Testing
As software grow in size, testing each screen of the tool manually becomes infeasible and becomes a major bottleneck when rolling out updates. Automated testing can give developers quick feedback when something breaks so the problem can be tracked down quicker, and reduces the time between testing and deployment.

### Questionnaire Revision Control
Since the survey tool will allow the admin to freely modify the questionnaire questions, sections, categories, and results weighting scheme, questionnaire revision control must be implemented so that when a questionnaire is revised, the previously completed questionnaires will remain correctly stored in the database and always be available for viewing.

### Custom Formatted PDF Creation
After completing the questionnaire, users can request a PDF version of the completed questionnaire along with the recommendations. To make things look tidy, the PDF of the questionnaire will not look like a print out from a web page. Instead, these PDF documents will be generated by a PHP script and the layout can be data can be custom formatted for printing.

### HTML and CSS Styled Emails
Emails requested by the user at completion of the questionnaire containing the questionnaire recommendations and questionnaire permalink will be styled with HTML and CSS to make the email organized and visually appealing.

### Multiple Concurrent Surveys
To increase the usefulness of this tool, support for multiple concurrent surveys will be added.

### Web Site Analytics
To get a better understanding of the how the tool is used and its user base, Google Analytics will be installed to gather data such as page views, completion rates, demographics, and user agents and the results will be displayed graphically with Google Charts.

# Practicum Scope

The scope of this practicum is to create a questionnaire that supports the following features:

## Admin Section

- Secure admin with a login
- Support for creating and editing multiple questionnaires
    - Create or remove questionnaire sections
    - Create or remove questions from the questionnaire
    - Validate sections and questions on change. (using jQuery .change())
    - Reorder questionnaire sections (by drag and drop using jQuery sortable)
    - Reorder questionnaire questions (by drag and drop using jQuery sortable)
    - Customize question category (Color coded)
    - Provide explanation for each question
    - Enable or disable questionnaires
    - Sanitize form inputs
    - Customize questionnaire recommendations (Color coded)
- View statistics gathered by Google Analytics with graphs generated with Google Charts
    - Page views of various surveys, completion rates of various surveys, demographics, and user agents

## Public Section

- View and complete questionnaire
- Show an explanation of the question with a click of a button
- Change the yes or no value of each question
- Add optional notes at the end of each section
- Validate all questionnaire responses before continuing to the next section
- Automatically save form contents on section completion (using jQuery .change())
- Display questionnaire recommendations on questionnaire completion
- Generate permalink for completed questionnaire
- Print questionnaire questions and results to a PDF (custom formatted with FPDF)
- Email summary of questionnaire results with permalink to actual results (formatted and styled with HTML and inline CSS)

## Project Deliverables

- Project proposal
- Wireframes (Admin and public sections)
- Visual mockups (Admin and public sections)
- Database schema
- Static templates
    - Admin section page templates
    - Public section page templates
    - Email template
    - PDF template
- Electronic questionnaire and admin site
- Test Plans & Detailed Test Cases
- User documentations
- Usability Studies Report
- Automated testing scripts

## Expected Outcomes

At end of the project, the admin and public sections of the site and a MySQL database will be hosted on an Apache web server running on Linux. The admin will be able to create and manage questionnaires and view questionnaire statistics for marketing purposes. Visitors will be able to select the questionnaire they want to complete and upon completion view recommendations. They will also be able to print and email questionnaire results and recommendations.

## Schedule

| Task | Description | Est. Hours |
|---|---|---|
| **Project Management** | | |
| Gather requirements | Requirements from client | 5 |
| | Show client proposal | |
| | Clarify questions | |
| Client communications | Get client feedback regarding project deliverables | 20 |
| | Make changes based on feedback | |

| | | |
|---|---|---|
| Create proposal | Outline requirements<br>Describe purpose of project<br>Steps involved to achieve goal<br>Describe how success is measured | 30 |
| **Design** | | |
| Wireframes | Wireframe for admin section and user facing section | 35 |
| Visual design | Determine fonts, layout, and colour scheme for admin and user sections | 35 |
| Database schema | Design ER diagram and schema for MySQL database to store:<br>Questionnaire data as designed by the admin<br>User submitted questionnaire answers | 10 |
| **Development** | | |
| Environment setup | Install Apache web server, PHP, and MySQL | 3 |
| Create schema | Create database tables based on ER diagram | 2 |
| Create templates | Create HTML and CSS framework for admin and user sections | 50 |
| Backend hookup | Connect the front-end UI with the database to make the tool functional | 30 |
| **Usability Testing** | | |
| Initial mock-ups testing | Obtain user feedback early in the process by testing conceptual designs using Usabilla, then make changes to address issues and retest in an iterative fashion for several rounds. | 40 |
| Final testing of actual tool | Obtain user feedback on the final product. If necessary, make tweaks to resolve additional usability problems that remain. | 50 |
| Usability Report | Create report summarizing usability findings and | 5 |

| | | |
|---|---|---|
| | feedback from initial usability tests | |
| **Automated Testing** | | |
| Enter initial data | Enter dummy data into the database | 3 |
| Test Environment Setup | Set up Windows 7 and Mac OS X 10.8 virtual machines in VMware and install Selenium Remote Control on the host machine | 10 |
| Test Case Creation | Create test cases to test the functionality of the admin and user-facing sections of the web site<br>Create scripts in Selenium IDE for the test cases.<br>Revise test cases for later tests | 50 |
| **Manual Testing** | | |
| General Testing | Perform manual testing to ensure the major features are implemented and working. | 15 |
| Browser compatibility | Ensure the admin section works for the latest version of Firefox<br>Ensuring user sections works across the following browsers:<br>Windows<br>Latest versions of Firefox and Chrome, Internet Explorer 7, 8, 9, and 10<br>Mac OS X<br>Latest version of Firefox, Chrome, and Safari | 20 |
| Email client compatibility | Ensure emails sent can be viewed correctly for the following mailboxes:<br>Outlook, Thunderbird, Gmail, Hotmail, and Yahoo | 10 |
| **Bug fixes** | Fix bugs from automated testing, general testing, browser compatibility tests, and email client compatibility tests. | 20 |
| **Live site deployment** | Upload final site onto actual hosting server | 5 |

| | | |
|---|---|---|
| | Do final tests to ensure site functions correctly | |
| **User documentations** | Create documentation to show how to setup and use the admin sections | 15 |
| **Final Report** | Summarize all work done on the project and reflect on possible improvements | 30 |
| **Total Est.** | | **453** |

# Appendix A – Education & Experience

## Education

**Bachelor of Technology** **September 6, 2011 - Present**

- Currently a student in the Bachelor of Technology (BTech) program at BCIT.

**Diploma of Technology** **May 27, 2011**

- Completed the Computer Information Technology (CIT) program at BCIT.

**High School Graduation** **June 3, 2009**

- Graduated from Prince of Wales Secondary School.

## Experience

**Computer Projects Practicum 2** **March 15, 2011 - May 27, 2011**

- Created an employee email form to send emails to selected employees in bulk.
- Created an automatically generated graph to visualize the performance of selected employees over time.
- Retrieves employee email addresses and employee performance statistics from a database.

**Computer Projects Practicum 1** **September 9, 2009 - December 4, 2009**

- Created an online shopping cart for a local pet food company.
- Created a custom form to easily create, update, and categorize store items.
- Created a custom website template to display featured items and recent notices on the home page.

## Specializations

**Database Administration**

- Entity-relationship (EER) modeling
- Database normalization
- Creating schemas from EER models
- Database backup and recovery
- User rights management

**Network Security**

- Configuring remote logging with rsyslog
- Parsing UNIX and Snort logs
- Capturing, reading, and filtering Wireshark packet dumps
- Mapping networks with nmap
- Scanning network computers for vulnerabilities with Nessus
- Creating and testing firewall rules
- Wrapping services with xinetd

**Web Development**

- Installing and configuring Apache, PHP, MySQL, and PHPMyAdmin
- Managing websites remotely via FTP and SSH
- Installing and configuring WordPress sites
- Writing and testing SQL queries, XML, XHTML, HTML, CSS, JavaScript, PHP, and AJAX

# Appendix B - Test Plan

The test environment will consist of two computers. One computer will run Apache and MySQL while the other will run multiple browsers.

## Admin Section

Because only one person will be accessing the admin section of this site, only one browser will be tested (Latest version of Firefox on Windows 7).

An admin should be able to log in and perform the following tasks:

- Create a new questionnaire.
- Create a new section.
- Add questions to each section.
- Assign a category per question.
- Edit a question and category
- Remove a question
- Remove a section
- Reorder questions
- Reorder sections
- Save questionnaire

## User Section

The user should be able to perform the following tasks:

- View the questionnaire that has been created by the admin.
- Fill out the questionnaire
- Enter notes at the end of each section
- View the questionnaire results
- View a saved questionnaire using a permalink.
- View a PDF version of the questionnaire answers and results
- Email the questionnaire results
- Submit feedback

## Browser Testing

Test functionality and layout listed in the user section above for the following browsers:

Windows

Latest versions of Firefox and Chrome, Internet Explorer 7, 8, 9, and 10

Mac OS X

Latest version of Firefox, Chrome, and Safari

### Sample test case

The admin should be able to create a new questionnaire.

| Steps to complete | Expected Outcome |
|---|---|
| 1. Enter the admin site URL and log in using their login info | The admin should be able to successfully login and be redirected to the admin landing page, which should be blank except for a button to create a new questionnaire. |
| 2. Click the button labelled “create questionnaire” | A new questionnaire form should appear on the screen. |
| 3. Enter a name into the questionnaire name field | The contents of the questionnaire name field should be automatically validated and saved onBlur. |
| 4. Click on the create section button | A new jQuery sortable list item should appear at the bottom of the questionnaire form. |
| 5. Name the sections and adds questions to it<br>i. Enter the question, description, and select a category for the question<br>ii. (A list of categories are created previously) | The contents of the section name, question, and description fields should be automatically validated and saved onBlur. |
| 6. Add as many questions as needed | A new jQuery sortable list item should appear at the bottom of the questionnaire form. |
| 7. Add as many new sections and questions as needed. | A new jQuery sortable list item should appear at the bottom of the questionnaire form. |
| 8. Rearrange question order by clicking and dragging on them<br>i. Move questions within the same section by clicking and dragging<br>ii. Move questions into another section by clicking and dragging | The order of the sections and questions should be automatically validated and saved when rearranged. |
| 9. Clicks on the “save” button to save the changes | The contents of the questionnaire form should be saved and the user should be redirected to the admin landing page which should now have a link to the newly created questionnaire. |

The user should be able to complete and submit a questionnaire.

| Steps to complete | Expected Outcome |
|---|---|
| 1. Enter the web site URL and select a questionnaire to complete | The user should be able to select a questionnaire and be redirected to the questionnaire form. |
| 2. Enter a project name into the project name text box | The project name will be validated onBlur. |
| 3. Click on the more information button beside a question | The questionnaire description should appear underneath the question. |
| 4. Click on the yes or no button beside a question | The yes or no buttons should change state and only allow one choice at a time. The questionnaire progress bar will update. |
| 5. Enter some non-ASCII characters in the notes text box | The contents of the notes text box will be validated onBlur and the textbox will have a |

| | |
|---|---|
| | red outline. |
| 6. Click on the "Next section" button without answering all the questions. | The unanswered questions will be highlighted with a red box and the page will scroll up to the first unanswered question. |
| 7. Click on the "Next section" button after answering all the questions in the current section. | The page will scroll down to the next section. |
| 8. Answer all the questionnaire questions and click the "Submit" button. | The user should be redirected to the questionnaire results page. |
| 9. Click on the "Email results" button | A form, overlaid on the current page, will appear to allow the user to type the email address to send the results to. The form will disappear once submitted. |
| 10. Click on the "Save as PDF" button | A PDF of the questionnaire answers and results will be generated and displayed in a separate web browser tab. |
| 11. Click on the "Give feedback" button | A form, overlaid on the current page, will appear to allow the user to enter their feedback and optionally, their email address, if they wish to be contacted afterwards. The form will disappear once submitted. |

## Usability Testing

| **Measures** | **Acceptable Levels** |
|---|---|
| Satisfaction of using the tool on a scale of 1 to 10 | 8/10 |
| Tool's ease of use on a scale of 1 to 10 | 8/10 |
| Likelihood of using the tool again on a scale of 1 to 10 | 7/10 |
| Average time taken to answer each question | 1.5 minutes per question |
| Average time taken to correct an error | 1 minute per question |
| Total amount of time taken to complete the survey | 30 - 50 minutes |
| Number of clicks to complete the survey | 30 - 40 clicks |
| Number of scrolls to complete the survey | No more than 5 scrolls |