**Tool**

Technical Guide

Ian Field

**Abstract**

Technical guide for tool.

Table of Contents

1 Executive Summary 3

2 Requirements Analysis 4

3 Design 5

4 Content 6

5 Implementation 7

5.1 Database 7

5.1.1 Data model 7

5.1.2 Relational Model 7

# 1 Executive Summary

This tool has been created to be a flexible platform for gathering result sets from public participation of content. For example, a block of text can be presented on the web page, beneath that a series of check boxes will be shown with values associated with the text. This could be used as an assessment platform to analyze human input or reaction.

The tool supports multiple languages with a few elements of user customization throughout. By editing the text-based files it is possible to provide a limited customization of the site.

# 2 Requirements Analysis

The correct and clear display of the corpus articles is paramount as this is the main focus of the project. This should be immediately followed by the related keywords of the article in a clear and easy to select format. There should also be a way of quickly selecting none of the keywords, should this be the visitor’s response.

With the nature of the project – results of multiple algorithms or chance for additional keywords, it is natural for some keywords to be duplicated. The visitor should not be able to notice that there are multiple keywords, which are the same for any given article, nor should they be able to determine which keyword result set they are from.

The number of keywords for each algorithm or chance group is 5, each of these has a possible length of 50 characters, this is expected to be enough, however, should it not be the case then the tool supports this, however the visual outcome of such an action on the web page may not be desirable – care has been taken to mitigate such effects.

The database should record enough information about each visitor’s selections in order to determine and relate back to which article they are referring, as well as supporting the merged keywords as they are presented to the user, but they are two separate fields in the database.

# 3 Design

It was decided that the overall background and colour scheme should be high-contrast and promote openness to help to engage visitors and reduce the number of website hits which do not yield any further data.

The aim of the site is also to maximize the likelihood of repeated, honest participants in the survey or other data gathering. With this reduces the possibilities of filtering spam or malicious entries through the use of something like a CAPCHA (“***C****ompletely****A****utomated****P****ublic****T****uring test to tell****C****omputers and* ***H****umans****A****part”*), which essentially would present the user with an obscured word to type in. This was felt to delay the process or rapid result gathering that the site is intended for so the database results should be used with caution.

With this smooth, modern looking design was created utilizing the background colour #67C8FF, CSS 3’s support for round-cornered borders, the content background contrasts this by being white, with the articles themselves having a black colour and having an easy to read web-safe font determined by the visitor’s browser’s capabilities.

The site’s textual content supports the customization based on user needs. The sections, which are customizable, are potentially limitless, however those detailed within the user manual are as follows:

Navigation links, Website title pages, Home page title, About page title, About page body.

It is expected that the home page content will be populated from a user created database, however two example entries have been included for user testing and tool exploration.

# 4 Content

This section details the construction of the web pages and the location of various site components. The DB folder contains the SQL files needed to set up the database. The file names are fairly self-explanatory. The Docs folder contains this document and a user manual. The html folder is the contents of the site for use within the public html folder of the webserver.

|  |  |
| --- | --- |
| Index.php | The initial entry point for the website |
| Navigation\_bar.php | The navigation bar links, included on all pages |
| Process\_entry.php | The form handler, which processes the POST from JavaScript and then updates the database with the user’s options |
| Footer.php | The footer which is included on each page of the website |
| About/index.php | The about page for the website |
| Config/about.php | The text file to populate the about page with. |
| Config/locales.php | The file, which contains the php variables for the link text and the home page title text. This is to support a different language, however it is assumed that the text used will be html compliant (e.g. Romanised). |
| Content/privacy.php | The privacy policy should it be required. |
| Content/terms.php | The terms and conditions page, should it be required. |
| Css/reset.css | The reset Stylesheet to hopefully reduce cross-site scripting. |
| Css/style.css | The actual Stylesheet used for the web page. |
| Db\_conn/closedb.php | Close the opened database collection, just used for a shortcut. |
| Db\_conn/config.php | The configuration file for the database connection, including the database, username, password and location. |
| Db\_conn/opendb.php | Open the database connection specified within the configuration file. Used for a shortcut in the php code. |
| Javascript/form.js | This is the main event handler and posting handler of the database. It utilises JQuery to perform the POST operation for the form.  The submission is prevented should no user interaction be made to the page; this is to reduce the number of spam results for later analysis.  The selection choices are maintained even with the selection of ‘none of the above’, however they are not updated in the database unless ‘select from below’ is selected. |

# 5 Implementation

## 5.1 Database

The database platform used is MySQL due to php’s native support and no-cost availability.

The corpus title in the example files has a limited field size of 100, the body has a limit of 2500 characters and each of the options has a 50-character limit. These can be changed as appropriate either through an administration tool or within the SQL script. For more detailed information on the fields in the tables please see *DB > create.sql*.

The Selections table is designed to require the least space, and allow for very fast queries due to operating on a Boolean data type as it is expected to have a large number of rows over the experimental period.

The option groups in the selections table are Booleans, which are intended to directly correspond to the entries table options. The selection time and the *user\_ip* address are automatically populated to facilitate the customized filtering of the result set should a particular *ip\_address* or time slot be identified as spam when mining the data.

There is a relation between the two tables via the *entry\_id* key for more complex SQL queries. Both *entry\_id* and *selection\_id* are automatically incremented, and as such do not require calculation upon an INSERT function.

Within the folder named DB with some example SQL scripts. Including adding and removing constraints on tables, along with a single script to set up an example database with two entries populated.

### 5.1.1 Data model

Entries (entry\_id, corpus\_title, corpus\_body, a1, a2, a3, a4, a5, b1, b2, b3, b4, b5, c1, c2, c3, c4, c5)

Selections (selection\_id, entry\_id, user\_ip, selection\_time, a1, a2, a3, a4, a5, b1, b2, b3, b4, b5, c1, c2, c3, c4, c5)

### 5.1.2 Relational Model

**Model:** Tool

**Relation** - *Entries*

*entry\_id*: Integer > 0

corpus\_title: String **NOT NULL**

*corpus\_body*: String **NOT NULL**

*a1*: String **NOT NULL**

*a2*: String **NOT NULL**

*a3*: String **NOT NULL**

*a4*: String **NOT NULL**

*a5*: String **NOT NULL**

*b1*: String **NOT NULL**

*b2*: String **NOT NULL**

*b3*: String **NOT NULL**

*b4*: String **NOT NULL**

*b5*: String **NOT NULL**

*c1*: String **NOT NULL**

*c2*: String **NOT NULL**

*c3*: String **NOT NULL**

*c4*: String **NOT NULL**

*c5*: String **NOT NULL**

**Primary Key** (*entry\_id*)

**Relation** - *Selections*

*selection\_id*: Integer > 0

*entry\_id*: Integer > 0

*user\_ip: String* **NOT NULL**

*selection\_time:* DateTime **NOT NULL**

*a1*: Boolean **NOT NULL**

*a2*: Boolean **NOT NULL**

*a3*: Boolean **NOT NULL**

*a4*: Boolean **NOT NULL**

*a5*: Boolean **NOT NULL**

*b1*: Boolean **NOT NULL**

*b2*: Boolean **NOT NULL**

*b3*: Boolean **NOT NULL**

*b4*: Boolean **NOT NULL**

*b5*: Boolean **NOT NULL**

*c1*: Boolean **NOT NULL**

*c2*: Boolean **NOT NULL**

*c3*: Boolean **NOT NULL**

*c4*: Boolean **NOT NULL**

*c5*: Boolean **NOT NULL**

**Primary Key** (*selection\_id*)

**Foreign Key** (*entry\_id*) **References** *Entries*