Proposed SpikeBee Web testing procedure- 2020.

This is a set of recommended tests and review that will ensure that our site is fully functional, as well as secure and as user friendly as we can make it. The following categories of test will be performed:

1) Functionality Testing

2) Usability testing

3) Interface testing

4) Compatibility testing

5) Performance testing

6) Security testing

1. **Functionality testing**

   This is the testing of "mechanics" of the website, this will verify that our site is free of broken links, exceptions etc. Even if it's a quick fix, any bugs found should be added to our MS To-do list and assigned to oneself if you are capable of addressing it.

   Check out all the links:

   - Test the outgoing links from all the pages. This is just the test of the "registration link "we create.
   - Test all internal links. Every combination should be tested: Page A to page B,C,D. Page B to page A, C, D etc.
   - Test links are used to send emails to admin or other users from web pages. Verify Contact page.

   Test forms on all pages:

   - Wrong inputs in the forms to the fields in the forms I.e. wrong datatype.
   - Field validation, check with #$% characters, numbers , negatives in int fields etc.
   - Verify Error message for wrong input
   - Verify Optional and Mandatory fields
   - Verify email account verification works.

2. **Usability Testing**

Usability testing is the process by which the human-computer interaction characteristics of a system are measured, and weaknesses are identified for correction.

• Ease of learning

• Navigation

- Subjective user satisfaction

- General appearance

Test for Navigation:

Navigation means how a user surfs the web pages, different controls like buttons, boxes or how the user uses the links on the pages to surf different pages.

Usability Testing includes the following:

- The website should be easy to use.
- The instructions provided should be very clear.
- Check if the instructions provided are perfect to satisfy its purpose.
- The main menu should be provided on each page.
- It should be consistent enough.

Content Checking:

Content should be logical and easy to understand. Check for spelling, grammar, punctuation errors.

Third Party

I propose that we get a spouse or friend to test drive the site from both the perspective of a Vendor, and as a parent.

Get the individual to imagine themselves as a Vendor and create a program, edit it and the delete it.

Get the individual to imagine themselves as a Parent and search a program.

In both cases ask the individual to record moments when the flow of their usage of the site stopped because they could not instinctively find the next step.

We will then address the unclear processes (Hopefully by simply adding further instruction text!)

3. **Interface Testing**

Once the site is deployed, the server-side interface should be tested. This can be done by verifying that the communication is done properly. Compatibility of the server with software, hardware, network, and the database should be tested.

The main interfaces are:

- Web server and application interface
- Application and Database server interface (If applicable to us).

Check if all the interactions between these servers are executed and errors are handled properly. If the database or web server returns an error message for any query by the application server then the application server should catch and display these error messages appropriately to the users.

Check what happens if the user interrupts any transaction in-between? Check what happens if the connection to the webserver is reset in between?

4. Compatibility Testing

Once the site is deployed, perform a full run through (CRUD for Vendor) of the site using MS Edge, MS Internet explorer, Chrome, Firefox and Safari.

Perform a full run through (CRUD for Vendor) of the site using both Apple and Android mobile browsers.

5. Performance testing

Once deployed we will test how the site behaves and how long it takes to change pages, do searches etc. Once there is a larger amount of content in the database this will be more meaningful. The creation of this content will be part of the testing process. This may also be a test of server performance if we eventually utilize an external DB server.

6. Security testing

Since we are using ASP.NET Core individual user accounts for maintaining the secure parts of the site, there may be little to test as far as security. A basic attempt to type secure page names into the browser should be done.