Web Development – Mr. Turner

Project – Mad Libs

**Project Overview**

Mad Libs is a game where the player is given a series of word types (nouns, adjective, verb, etc…). With only knowledge of the types, the player makes up a word for each space. When the player is done making up words, a story appears, with his or her words filled into the appropriate spots, making a, usually, nonsense story.

**The Page**

Using a form, you are going to create a Mad Libs web site.

Included with this spec sheet is a file called ***story.js***. This file contains two pieces of information:

* A DELIMITER, which is used to identify the word types in the stories.
* A STORIES array, which contains 3 stories (but feel free to make up your own if you like.

Each story in the array is short, but contains a number of word types for the user to play with. Each word type is surrounded by the DELIMITER (a \*\*).

There are no spaces in any of the word types.

On your ***index*** page, create a form.

Choose a random story from the array. You will need to know which story you chose for the destination page, so make sure to keep track of its index in the array. The best way to transfer this information to the destination page is by creating a hidden text box in your form. Just create an input box and make its type *“hidden”* instead of *“text”*. Once you’ve determined which story you’re working with, store its index in the hidden text box. It will transfer to your destination page with the rest of your data.

Go through the story and create a text box for each of the word types listed. Make sure you also output the word type itself so the user knows what to type into the box.

Once the user has typed in all of the requested words, the destination page will load.

On your ***destination*** page, you should display the story.

In order to properly display the story, you will need to read it again from the STORIES array in story.js. You will find the proper index in the query data (assuming you wrote it to the hidden text box properly).

Replace all instances of the word types, with the information entered by the user on the previous page.

Create a link back to the ***index*** page so that user can try again with a different random story.

***Both pages must include CSS. These pages should look finished and have an intuitive interface.***

**Programming Requirements**

|  |
| --- |
| * You must code your own HTML and CSS. Use of a drag and drop interface or the usage of code downloaded from the internet is not permitted. |
| * Use of deprecated code is not permitted. Use an HTML 5 reference as your guide. |
| * Your home page must be named *index.html*. |
| * You must comment your name into the top of every page. |
| * Your code must be structured in a consistent and legible manner |
| * Your text must be organized through the use of sectioning tags (div, span, p, h1 – h6). Tables are acceptable for data but not for page design. |
| * Your interface must be smooth and easy to figure out for a client. |
| * Your pages must be formatted using CSS. |