Coursework Report

Richard Borbely
40283185@napier.ac.uk
Edinburgh Napier University - Web Technologies (SET08101)

1 Introduction

This paper describes the various stages and challenges of designing a website using ***HTML***, ***CSS*** and ***Javascript***.
The website consists of a home page with general information on **classical ciphers**, additional pages for each cipher included and a design page. Cipher pages include a brief description and a text area to code and decode messages using the algorithms implemented in Javascript.
Ciphers included in this project are; the well known *ROT13*, a *substitutional key cipher* and a simple *transpositional cipher*.

2 Software Design

The Plan. Create the necessary documents for the project;

- index.html
- design.html
- ciphername.html for each cipher

Once the documents are in place, the index page should have a basic look and a navigation bar implemented, linking the items with the corresponding pages. The navigational model is shown on *Figure 1*.

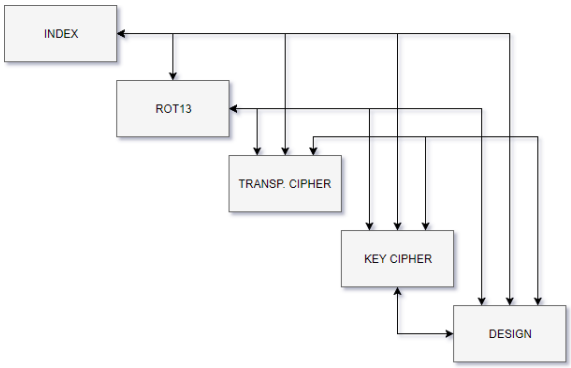

Figure 1: Navigation diagram

Each page is able to access any of the other pages through the navigation bar.

A single external CSS file is going to be attached to the created pages, which gives the ability to easily manipulate divisions around, and customise various design properties in an organised manner. The choice of colour palette is an important step, the information on the pages must be easily readable, but still blend in quite well with the context.

Figure 2: Balanced colour palette use

3 Implementation

Navigation bar The navigation bar was implemented using an unordered list, it's elements are *floated* to the left so that it would display as a horizontal list. The choice of background and the use of shadows bring the whole bar 'out' of the page, making it clearly visible for the user.

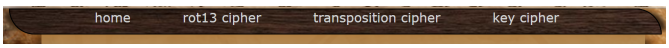

Figure 3: Navigation bar with menu items

Ciphers For the the ciphers, the use of *'textarea'* blocks are perfect for both input and output. As for the functionality a single external JavaScript file is attached, that contains the algorithms for coding and decoding messages, and also other non-cipher related functions such as animations.

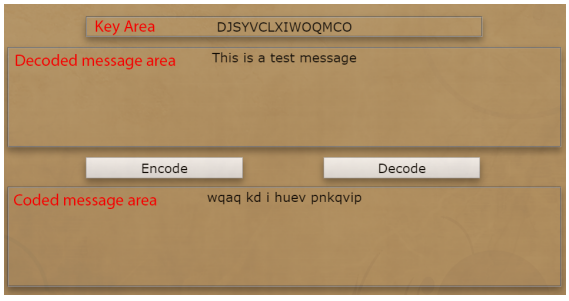

Figure 4: Input, Output and Key Input area for a cipher

Advertisements Advertisements appear on the right side of the pages. These advertisements can be customized by providing an image that is going to be displayed, and a link to navigate to if the image is clicked. By default there are 6 advertisements that are cycled around every 8 seconds, each pointing to a valid link. Two of the same will never appear on a page.

Figure 5: Dynamic advertisement area

4 Possible improvements

Algorithms Improving the algorithms would definitely be the next step for this project. They are currently only able to process letters of the alphabet *(a-z, A-Z)*, separated by spaces. If a number or a special character is entered, the user will not be given an error message and the output could be misunderstood. This could be improved by allowing the user to enter various types of characters and have the algorithms simply ignore them.

Navigation The currently implemented navigational model could be optimized further as parts of the website *(background, navigation bar, contents area)* are static on every page. Navigating to another page will force the browser to load the same elements repeatedly. Therefore a more optimal way of getting around the website would be to have an index page with a *'content div'* and load every other page inside that *'div'*. This would greatly improve the performance and provide a better structure for the website.

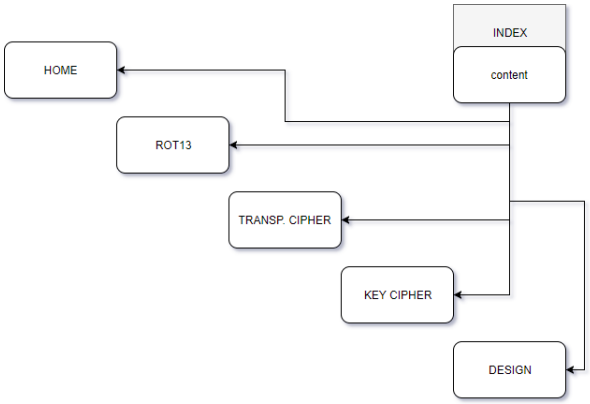

Figure 6: Improved navigation diagram

5 Personal Evaluation

Animations Each page has a *'content div'* with a fade animation implemented. The opacity of the element is set to 0 by default in *CSS*. Whenever a page is finished loading in, the *'animation_fade'* function is called which will increase the element's opacity over a period of 0.5 seconds, which is defined in *CSS* using *transition: .5s;*

```
1  function animation_fade()
2  {
3     for(var i = 1; i <= 90; i++)
4     {
5        document.getElementById("content").style.opacity = i/100;
6     }
7  }
8
```

On page load, there's also function calls for the Dynamic Advertisements, function *'cycle_ads'* will call *'ad_change'* every 8 seconds. The advertisement names and their associated links are defined inside the JavaScript file, where the name equals to the file name of the image that needs to be displayed.

```
1  function cycle_ads()
2  {
3     setInterval(ad_change, 8000);
4  }
5
6  function ad_change()
7  {
8  var randomAD = getRnd(0, ad_names.length - 1);
9  //set AD1 element image
10 //set AD1 element link
11
12 var randomAD2 = getRnd(0, ad_names.length - 1);
13 while(randomAD == randomAD2){
14    randomAD2 = getRnd(0, ad_names.length - 1);
15 }
16 //set AD2 element image
17 //set AD2 element link
18 }
19
```

To make sure that Ad's do not repeat on a page the second random number is going to be generated until it differs from the first one.