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.

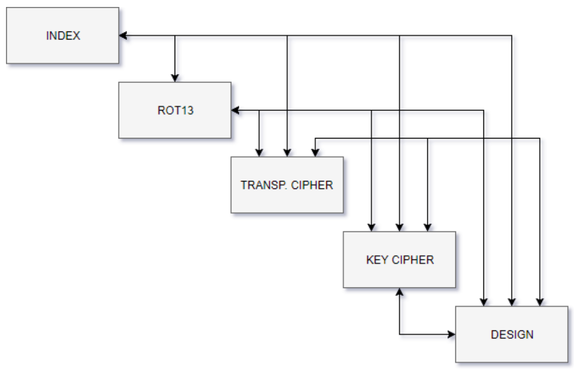


Figure 1: Navigation diagram

This 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'*. The navigation diagram for this is shown on *Figure 2*.

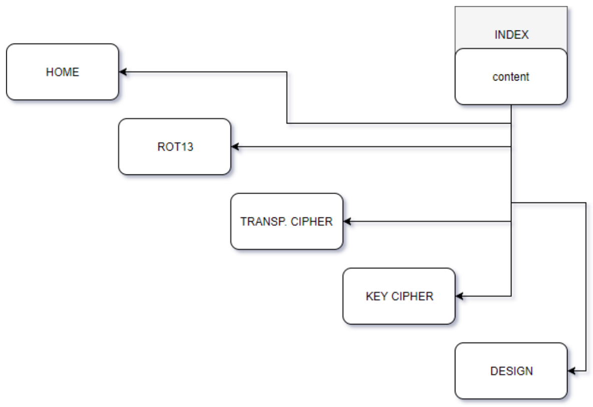


Figure 2: Improved navigation diagram

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 3: Balance between color blindness test and a stop sign.

3 Implementation

JQuery A way to implement the navigational functionality previously mentioned is by using *JQuery*, each individual page will consist of a single *'div'* with all their contents inside. The index page will act as a shell for the different contents. When a navigation bar item is pressed, the contents of the corresponding page will be loaded into the *'content div'*.

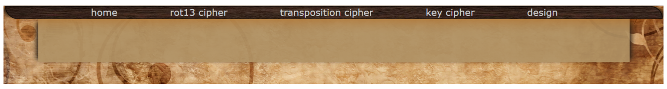
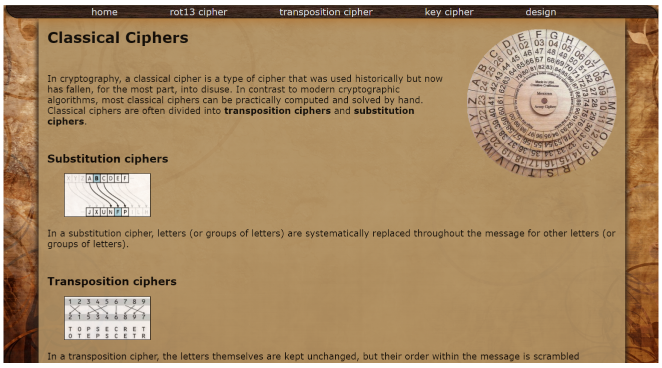


Figure 4: Plain index page with navigation bar and the contents section being empty



Classical Ciphers

In cryptography, a classical cipher is a type of cipher that was used historically but now has fallen, for the most part, into disuse. In contrast to modern cryptographic algorithms, most classical ciphers can be practically computed and solved by hand. Classical ciphers are often divided into **transposition ciphers** and **substitution ciphers**.

Substitution ciphers

In a substitution cipher, letters (or groups of letters) are systematically replaced throughout the message for other letters (or groups of letters).

Transposition ciphers

In a transposition cipher, the letters themselves are kept unchanged, but their order within the message is scrambled

Figure 5: Navigation bar and contents section displaying 'home' content

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.

4 Critical Evaluation

Possible improvements 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. This could be improved by allowing the user to enter various types of characters and symbols.

5 Personal Evaluation

Animation One of the many challenges I have faced during this project is getting the animations to work properly. There's a function for changing the *content div*'s opacity to 0 and increasing the value back to 0.9 over a period of 0.5 seconds which is declared in the *CSS* file using *'transition: 0.5'*

```
function animation_fade()
{
    document.getElementById("content").style.opacity = 0;

    for(var i = 1; i <= 90; i++)
    {
        document.getElementById("content").style.opacity = i↩
/100;
    }
}
```

This function is called as soon as the website loads in, and every time a navigation happens.
When the website loads in the first time, because the opacity of the *content div* is set to 0 in *CSS* and the animation plays fine, but it doesn't seem to change the value when a navigation happens, unless in debug mode. Unfortunately, I was unable to solve this issue, but as far as I understand it is caused by the script being called before the *DOM* is ready, therefore the *getElementById* function is not actually able to access the element.

JQuery To successfully implement the navigational model I have had in mind I had to research *JQuery* and with that came a lot of questions. Although the assignment said not to use additional libraries, it felt like a much better way of navigating a website in terms of performance and structure.

NGINX With the use of *JQuery* the website had to be hosted and the suggested *NGINX* was my choice. After a quick research I turned off *caching* in the browser, yet still a lot of times changes to the documents would not update on the hosted server, not even after several restarts.