|  |
| --- |
| Edinburgh Napier University |
| Advanced Web Technologies (Set09103) |
| Coursework 1 Report: Google’s Doodles online catalogue |
|  |
| **40071697 - Amantle Craviolatti** |
| **10/31/2016** |

|  |
| --- |
|  |

Table of Contents

[Introduction 2](#_Toc465967965)

[Design 2](#_Toc465967966)

[Enhancements 4](#_Toc465967967)

[Evaluation 5](#_Toc465967968)

[Technical 5](#_Toc465967969)

[Personal 5](#_Toc465967970)

[Appendix 6](#_Toc465967971)

[Resources and References 7](#_Toc465967972)

# Introduction

# The web application developed for this coursework is Google’s Doodles; a personalised repository for still and moving alterations of the Google logo to commemorate special events [1] . The catalogue is made of doodles displayed on Google homepages around the world through the months of August to October 2016. The python based application developed within the Flask framework makes use of Google Material design, JQuery Jinja2, Html and CSS to present the collection of selected doodles.

# This report outlines the development process for this web application from design to implementation including evaluations of the development journey and final product as well as recommendations for possible future enhancements

# Design

The design process for the Google’s Doodles web application began with disposable paper mockups of the various web pages of the app with preliminary listing of the desired functionality for each page with numbering based on priority as a loose requirements specification [2] (appendix 1). From the collection of the completed paper mockups, a list of potential server side functions already covered in the practicals (appendix 2) was created which helped to inform the desired url structure for the app. A base directory structure was also created of the web application with space for css and images as the idea for the web application relied on it being image heavy. The full sample size of images to be used across the application was limted to those images appearing on the Google website during August to October 2016. This period also had the advantage of being at the peak of the style and diveristy of Google Doodles since the logo enhancement beacame a regular feature since its initial inception in 1999, while ocurring post the 2016 Olympics, a period during which doodles were largely integrated games as well as enhanced Google logos.Had this period been included, an unnecessary layer of complexity and load in terms of application size and memory would have been placed upon the application, however there is a cosideration that doodles of that time could be included in future enhancements and advancements of the application. Once completed, the initial paper mockup plans were then converted to Html and CSS clickable mockups forming the basis of the second stage of prototyping and further research.

During this second stage of design, research was focused heavily on the differences between web applications and websites to ensure that Google Doodles behaved as expected for an application of its level. Thriough internet searches and usage of websites such as Awww. known from personal experince in industry as reliable for curating current web trends including those of web applications, a short visual review of applications that were highly rated for design took place. This as well as consultation with the current Google archives for their doodles led to the aesthetic design conclusion was to style Google’s Doodles using modiefied Google material Design, while the decissions around functionality design were largely informed by the results of the visual review- namely that applications who’s functions were simple and aligned with the purpose of the web applications fared better than counterparts that were bulkier in features. This too was true of the Google Archive of Doodles which was hard to navigate due to over-use of dynamic elements an example of which being when hovering over a doodle card that pointed to a interactive doodles, you were taken to the specific information page for that doodle rather than the full collection of interactive doodles.

The aesthetic and functionality design decisions are most clearly displayed through the card like structure on the home page of the application which is simple in order to maintain focus on functionality and this proved to be the most beneficial design structure over one leaning toward the traditional standard website structure to showcase the features of the application.

Prior to implementation, the Structure of the application fell in to the following categories, and collection of the sample of Google doodle images and gifs began through manual downloads for the first iteration of the prototype application. These were placed into the earlier decided static folder structure and the url structure was decided around these 3 main elements of the web app:

Personal user collection

General browsing

Latest Doodle

These functionalities then had the following urls nested under them and are reflected largely within the live version of the app:

Personal user collection

            Sign in to  Application

            Upload

All doodles browsing with jpg and gifs

General browsing

            All doodles Jpg only browse

*View individual files (available in the browser)*

Latest Doodle

Static loading latest google doodle from Google archives

# Enhancements

The current version of the Google’s Dooodles application largely fulfills the fitness for purpose of the application, namely that of simple repositiry for Google Doodls with the option of personalising the repository and limted curation through the uploading of files. However the extent to which this is done at present is currently limited due to constraints of time and current ability.However the following list comprises of possible enhancements to current functionality to further achieve to the current purpose of the application and is listed in order of feasibility and ease- known in some cases from attempts to include these during the current iteration of development with unseccessful results, and in outher cases as a reslut of research and are theortically possivle with minimal effort and more time.

|  |  |
| --- | --- |
| Enhancements | Merits |
| Category for popular events such as olympic doodles | Enhancements |
| -search implementation , this was considered but Issues in development:  -Around implementation of Searchwith nested if statements vs impact on efficency of site in serving images for collection of that size, was decided that best kept for larger sample size with database implementation. Deemed unnecessary | Enhancements |
| use of jquery for dynamic loading of images in view image page to load more on scroll, was considered and trialled but deemed unnecessary for size of sample- | Enhancements |
| --Dynamically loading latest google doodle from Google archives-----Surveying the Google site alsogave the idea of dynamic loading of the most recently added doodle as a way to grow the collection available on the web app in future. | Enhancements |
| Recording user activity in session to suggest new images each time | Enhancements |
| Enhancements | Enhancements |
|  |  |

# Evaluation

## Technical

## During development it became clear that certain functions such as that to read and acess Images stored in the static directory would need to be re-used . A simple solution to this was a cobination of using multiple app routing statements and nesting the directory reaing of various files under one condition. This demeonstrated a limited use of DRY technicques, however could stand to be improved by saving this functionality under a standalone function instantiated across multiple routes, or superior to this, future versions would do better to make use of a database whcich could then be queried across the application and would allow simpler implementation of search function within the application.

## Stricter implementation of Software design methodoligies alongsidenprototyping and initial requirements specification would also have better supported development as large amouts of development time were spent searching for optimal and efficeint methods to implemement features such as the reading of images which may have been lessened if research for this was a core task in the requirements checklist during the design stage and specific time was allited to do this.

## Personal

The most challenging parts of the development where in not moving too far ahead of the material already covered in class into more complex territory such as the use of databases, which had not been covered at the time of planning the application. This decision was so as to ensure that the web application did not go too far beyond the coursework scope which would have meant that time was wasted on demonstrating development techniques there is opportunity to attempt in future and to ensure that the understanding of the basic applications of Jinja templates and Python in web applications were known to a high level, while including some of the more advanced functionality common in web applications on the web such as usage of jquery to dynamically load images in a browsing page. Merging the two levels of difficulty proved challenging at points especially in extending the Jinja conditional logic with jquery or learning what python functionality was available in the pre installed libraries and flask, especially in relation to Image rendering when may sources reviewed for this task recommended using PIL or the Python imaging library.

However it was a good exercise I becoming more precise in my search terms and I became well acquainted with Flask after frequent re-reading of the documentation.

# Appendix

# Resources and References

[1] https://www.google.com/doodles/about