

40122332

SET09103

Report

Introduction

The task that was set was to design, implement and evaluate a prototype of a web application of an online catalogue using the learning environment that has been taught throughout the module. The subject of the catalogue was breeds of rabbits.

The application has a menu bar that allows the user to sort by different defining characteristics of a rabbit e.g. ear type and colour. Once the user has selected a characteristic to search they then select one of the different options of that characteristic e.g. if the user selects origin then they could select Germany or Belgium. There is also a search bar that can be used to look for specific rabbits by their name. This will display a list of all the rabbits that fit the users search and then allow them to select a rabbit from the results that have been given. The root page of the application displays all the rabbits in the database so that a casual user would be able to simply look through the rabbits from the database in a leisurely fashion.

Design

I wanted my design to be simple and easy to use by the user and also not to have too many different routes. I just wanted a few routes that will interact with each other and display appropriate information depending on what the user has selected from the navigational bar.

There is also a search bar. The search bar can be used by the user to search for rabbits by their name. The search bar takes you to a page that displays all the results from the search which the user can then select and go to the rabbit page that displays all of the rabbits information. The search bar is located on the navigation bar.

The home page has a navigation bar that allows the user to make a choice from a selection of sort options for the rabbits in the database. The sort options have two different ways in which they are displayed.

The first way in which the sort option is displayed is through the use of drop down boxes. The drop down boxes were incredibly useful for the sort options that only had few set defined options to sort from e.g. the sort option “ear type” only had two options to sort from “lopped” and “unlopped”. The reason the drop boxes are incredibly useful is because the drop boxes allowed the user to go straight to the page that displayed the sorted rabbits instead of going through a separate page thus cutting out a middle man.

The second way in which the sort options are displayed is through the use of links. These links go to another page where the options to sort from are shown. These options would also act as links that would go to the next page that would display all the rabbits that correspond to options selected.

Below is a diagram that displays the different ways in which the user can get from the main route to the Rabbit they are searching for.

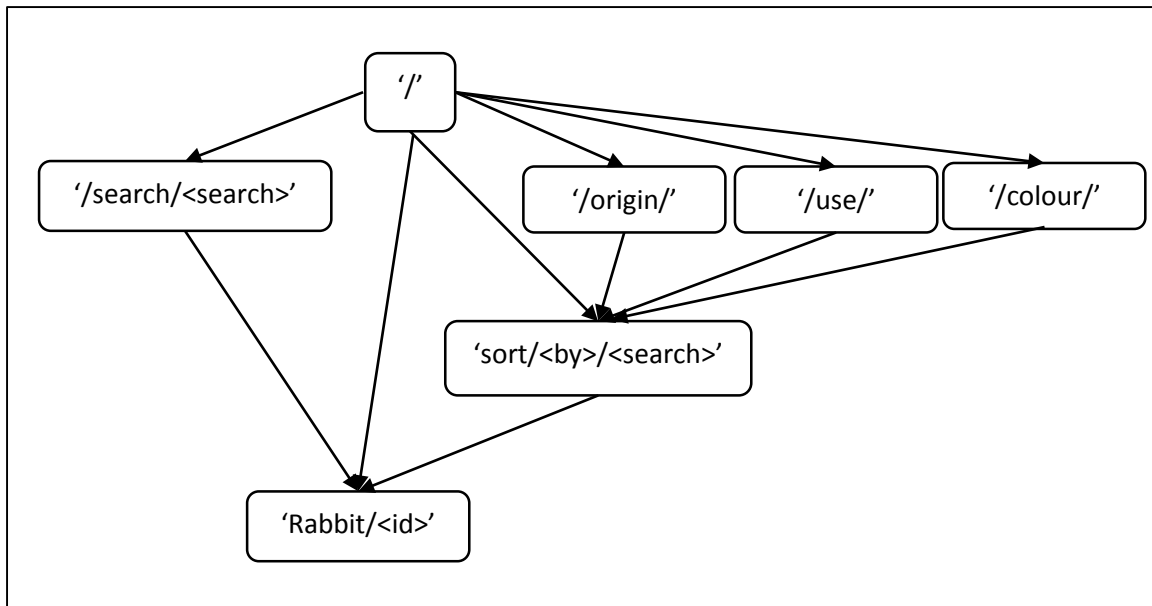


Diagram showing the different paths to the rabbit page

Enhancements

The first enhancement would be to the search function to allow users to search for more than just the Rabbits name. I would also add in a google maps API in to the origin page to allow the user to have a visual representation of where the rabbit came from e.g. If the rabbit was from Germany then the map would have an icon on showing where Germany is.

Added in more features such where a user may be able to purchase a specific breed of rabbit or even a list of trusted breeders. The google maps API could also be connected with breeders list and display within a certain country or region. I would also like to add in globalisation so that that the application can be viewed in different countries and languages. This would be useful as it would all the application to reach wider audience and it would also make better use of the proposed google maps API as that would allow it to display breeders from all around the world

A login system would also be a useful addition to the application. The login system would not be a mandatory and would still allow the casual user to access the main features of the page. The login system would allow users to act as a member allowing them access to certain parts of the application that would not normally be available to the casual user.

A user gallery would also be implemented allowing members to share images of Rabbits to public board. The reason as this would be member exclusive would be to monitor the images that are being posted to the application so that nothing unrelated or obscene would be posted onto the web application. The images that are posted to the application would be logged with the information showing who posted the image the time it was posted and the URL for the images that way if something was posted that shouldn't be then the culprit can be identified. Another feature that could be added to the gallery is a system that would do an images search to see if the image was taken from anywhere else on the internet and also check its legal status to make sure that we are not displaying a copyrighted image to the application.

Another feature that would be useful would be the implantation of cookies. The cookies would be used to change the header on the top left of the page so that instead of always saying "Welcome to Rabbit Dictionary" it would switch to just "Rabbit Dictionary" after the user had left the welcome page for the first time. The cookies would also be useful if the login system was added because if done in the correct way it would allow for the application to personalise some of the pages for the user e.g. instead of just saying "Welcome to Rabbit Dictionary" the header could display "Welcome <users name> to the Rabbit Dictionary".

Better testing would also be a huge benefit for the application as it would make sure that everything would run smoothly and that everything was working the way that it should. Also adding in some more logging would help to identify potential errors within the application.

The application would also benefit from an improved error handling. At the moment the application handler only has error handling for a 404 page not found error. Adding in error handling for a 500 error just in case something went wrong with database or a 503 error in case the database needs to be updated.

Pagination would also be a useful feature to add in. This would divide up the data into different pages so that the user could easily look through the data rather than spending a lot of time scrolling down to find what they are looking for.

Critical evaluation

One problem I found with the application is that if the user clicks on the links in the navigation bar too much it can cause the 404 error message to appear and sometimes it does not redirect to the home page which is what the error handler is supposed to do.

The logging that is implemented works to an extent as it logs certain information such as what rabbit the user is looking. The one problem I found with the logging is that whenever the application logs when a user returns to the home page it also does the log for the error handler. This can be an issue as from looking at the log it would appear that the application was constantly have 404 errors, which could cause a lot of confusion to anyone who reads the log.

Another problem is that the database and select statements are very simple so it makes it difficult to store a lot of information about certain characteristics e.g. a rabbit could have multiple colours and uses but due to the limitations of the database you can only assign one. This could be resolved by using better by using multiple databases to store the extra data.

Another issue I found was that some of the pages have a lot of white space which makes the pages look a bit empty and not very stimulating visually.

The search function works but it only allows the user to search for the rabbit's name. This could be a bit of an inconvenience for the user if they wanted to search by anything other than the rabbit's name.

The main positive is that the application is easy to navigate allowing the user to access the information they are looking for quickly and without much difficulty. This is possible through the use of the navigation bar that is located at the top of every page.

Personal Evaluation

The first obstacle I had was trying to figure out how I was going to store the data that would be displayed in the application. I decided that the best way for me to store the data was to use a data base. Since I was not sure how to implement a database into my application I had to do some research into the matter. I eventually found a website that gave step by step instructions on how to implement a database. I finally managed to get the database up a running which allowed me continue with rest of my application

One of the other problems I was having was with the CSS. At first I decided to do the CSS myself but I started to have problems when it came to doing the drop down boxes. I found that I was spending a lot of time working on the CSS and if I continued using my own CSS then I probably would not have gotten it done in time and the application would look messy. I resolved this by deciding to use Bootstrap instead of my own CSS. Bootstrap allowed me to make drop down boxes that worked and also over all gave my application a nicer more polished look to it.

Another problem I had was with deciding how to set up the navigation bar. At first I had decided to have all of my options as links that would link to a page where the user would then select the option to sort by. This became a design problem because it would lead certain pages like the size only having a couple of options to select form and made the page look very bare. I then decided that in order to avoid this I would add in drop down boxes on the navigation that way the user would completely cut out the option selection page and go straight to the search results.

The most important lesson I learnt from this project was to spend more time planning out the project so that I know what I'm going to be doing when it comes to writing the code. Also figuring out what features are going to be in the project at the beginning instead of adding them in further along the line. I found myself jumping straight into the code which resulted in me doing a lot of edits latter on once I figured out a more effective way to write the code.

Resources

<http://flask.pocoo.org/docs/0.10/tutorial/schema/> (used to create and implement the database)

<http://www.w3schools.com/bootstrap/> (used to help make bootstrap work in the application)

https://en.wikipedia.org/wiki/List_of_rabbit_breeds (used for the database entries and also used for images)

<http://www.rabbitbreeds.org/index.php> (used as research into already existing rabbit catalogues)