Foundation

Kevin Falconer
40172641@napier.ac.uk
Edinburgh Napier University - Advanced Web Technologies (SET09103)

1 Introduction

The foundation clothing application is an essential tool for anyone with a keen interest in streetwear. The application allows the user view a variety of different types of clothing from popular streetwear brands such as Nike, Adidas and Patagonia.

Once the user firsts visit the web application, the home page will be displayed. The home page includes the featured brand, and the latest product's which have been added to the web application. Alongside this is a navigation bar which is the key method of maneuvering through the variety of web pages on the web application, which include web pages for each item in the web application, a list of brands, and a list of product types. The user is also able to give feedback regarding the website through the use of the contact us functionality. With each user feedback being sent to the administrator.

The web application also allows a login functionality for the administrator of the website. Once logged in, the administrator can add, amend, or remove any of the items on the web application provided they have the correct ID number for a product. Information relating to all of the meta data regarding to each product listed on the web application is also viewable in the administrator page

2 Design

2.1 Planning

This coursework task had an extra level of complexity as it relied on having to brainstorm an idea from nothing. This added a level of pressure during the brainstorming process, as I did not want to potentially spend too long thinking of a concept, which could in turn leave me less time to implement that idea. However I was able to think of a core concept quite quickly. The concept was to create a clothing website. However, once I was settled on the idea, I had two contrasting ideas on how to implement a clothing website. The first idea was to create a clone of web applications such as ASOS, and Size, in which the user can maneuver through a collection of items from various different brands. The other idea was to implement a website similar to a mobile application called Depop, which allows users to sign up, and add their own items to the website to sell. I decided to go for the first option, with the hope that I could also implement a Depop/Marketplace functionality once the core web application had been created. Once settled on the concept, before looking at the sort of functionalities I could implement, I researched similar web applications such as ASOS, END, and Size, so I could get an idea of how their web applications were laid out, and so that I had a benchmark to compare my web application to once the coursework task had been completed. The decision to do research prior to even creating draft mockups for the website was an incredibly helpful process. I noticed that each of website's had a core structure and hierarchy to their web applications which consisted of web pages for a home page, which included new and featured products, webpages for each brand, type of product and individual items. From the home page, a navigation bar was the key method of maneuvering through the web application, so I made a note of this, and any other potential functionalities.

2.2 Design

Once I had completed my research, I started by planning out the URL hierarchy for the web application, and began designing mockups for the key page's. When designing each mock up, alongside designing how I wanted the web page to look, I also included annotations, detailing the potential functionalities which could be implemented on that webpage. This proved to be extremely helpful as it meant that prior to writing any code, I had a clear idea of how I wanted the web application to look, and how I wanted it to function.

2.3 Implementation

2.4 SQLite3 Database

When I began programming, I ensured that the implementation of a working database was one of the first tasks which I completed, as the database would contain all of the items which would be displayed on the web application. I started by using the workbook example to create a test database, using SQLite3. The database only a few column's, and rows, in which data could be input into the database using a line of code which contained a simple SQL command and then use another SQL command to retrieve and display the information, just to make sure the database was up and running. Using a combination of materials found online on how to input variables into an SQL execute command alongside looking at my python code from the previous coursework task, in which I had an example of using form's and a POST method to write user feedback to a text file locally, I was able to implement addition of rows into the database from user input. After getting this aspect of the functionality with the database working, I created another iteration of the database which included an extra row for an ID number of a specific product, and a row for a date. I created a variable for the ID number, so it would be a randomly generated number between 1000 and 100000 when the POST method was activated. A variable for the date was also created, which would retrieve the current date and time, and these values would

be added to the database alongside the form data. Thanks to the experience I had during the first coursework task of matching a URL variable to a specific value in a JSON file, the implementation of the webpage for a specific item using the ID number was straightforward. The final version of the database was not implemented until around a week before submission however this database was very similar to the second iteration except that it contained more columns for the inclusion of more meta data regarding as such as the product name and colour. I also implemented a second database on the web application which was used to store user feedback. This feedback could then be viewed through logging into the administrator view.

2.5 Flask Login

A login functionality was something I was also looking to implement on the coursework task. Initially I had planned to allow users to register, and have their credentials stored to a database so they could eventually log in, however after I realized I would not be able to implement the marketplace functionality due to time constraints, I settled on having a login functionality however only for an administrator. The administrator would then be able to add, amend, and remove items from the website. Once the core functionality for addition was implemented, I was able to tweak the example so it would work for the amend, and remove functionality.

2.6 Iteration

Although I done a considerable amount of planning and designing before the implementation phase of the web application, I found that during this coursework, the process of creating the web application was much more iterative than the first coursework task. I had the core backend functionalities of the web application such as the administrator login, and the ability to add new items to the database completed around a week before the submission date. This gave me more than enough time to try get some user feedback on what I had already implemented, and then return to the design phase to plan out new feature's. Examples of this include the administrator also being able to amend, and remove itemâĂŹs alongside adding new items, and the administrator all products page. The idea for the all products page, which displays all of the products, and it's meta data listed on the website, came from user feedback in which the user said they would have preferred an easier method of finding the ID number of an item, which at that time could only be accessed by maneuvering to the item page, and taking the ID number from the URL.

3 Enhancements

3.1 Login and Marketplace

The one key enhancement which I would make to the web application would be the inclusion of a login, and register functionality for new users. The functionality would allow users to login, if they had an existing account, or register if they did not. The login functionality would then lead to onto other enhancements such as the marketplace functionality which I was initially looking to implement when I initially designed the web application. In the marketplace, logged in users could view, and interact with products listed on the web application from other user's and add their own products, making the web application much more like a social media platform, alongside being a clothing web application. Users could also add certain products they liked to a wish list, a functionality provided by web applications such as ASOS, and Size.

3.2 Checkout

Another key enhancement I would make to the website would be the inclusion of a functioning shop and checkout system. A checkout functionality was another idea which I looked into when I was designing the website, however after trying to research methods of implementing this functionality, I was unable to find anything useful, or even any example's in flask on online forums, or other website's.

3.3 General Enhancements

Other enhancements I would make to the web application would be adding additional CSS, including brand logos, bootstrap functionalities such as an image carousel which would display more than one image of a particular product, and using the python library for form methods. WTForms was something which I looked into late during the implementation process and I had already implemented the form methods so that user input could be add, amend and remove items from the database. I decided not to implement the library as I did not want to potentially start work using the library and then encounter errors, and at the time I still had aspects of the web application which needed to be fixed. However, the library's functionality for input validation is something which is far superior to the simple JavaScript which I used for the web application.

4 Critical Evaluation

The features of the web application which I feel perform the best is the administrator view as a whole. A lot of the key functionalities in web application are accessed through the admin login page, and although I would have liked to implement a login functionality which would allow the registration of new user's, I was pleased that I could implement the web application so the admin user is able to add, amend and remove items. However one aspect of the add, and amend functionalities which I feel perform poorly, and could be improved is in relation input validation on each form. Currently, the input validation restricts users from submitting a POST with blank values, however that's the real extent of the validation. I was also looking to implement JavaScript which would restrict the user so that only .JPG files could be added to the web application, however despite finding various example's on online forum's, I was not able to get any of the example's to work, which I found to be extremely frustrating, despite only being a small aspect of the overall web application.

5 Personal Evaluation

Overall, I am pleased with the overall outcome of the web application, and my performance during the duration of the coursework work. I felt that during the implementation of the first coursework task, that I started far too late which meant I had little opportunity for feedback on the idea, and sort of features to implement, and it also meant I could not resolve certain error before the deadline. So, I made sure when attempting this coursework that I started early enough in which I had plenty of time to resolve errors, and get feedback during lab sessions, and from user's who tested the web application.

However, the experience I had of implementing the previous coursework task using JSON, was a massive help when it came to implementing this coursework. Although the python syntax was different when comparing retrieving values from a JSON file compared to a database, the core concept of checking if the URL variable matched a row in the database, and then displaying the relevant information if it found a match was a huge aspect of both coursework tasks.

References

Stack Overflow:https://stackoverflow.com/

W3 Schools:w3schools.com

Flask:http://flask.pocoo.org/

Size:https://www.size.co.uk/