Advanced Web Technologies Coursework Part 2

Site URL

http://webtech-2223-05.napier.ac.uk:5000/

Differences between my initial plan and my final implementation.

In my initial plan for my website as described in part one of my report I talked about various features I aimed to include as well as the theme and base design of how I wanted my functioning website to look. In this report I will outlining the differences in the original plan and final product as well as the challenges and accomplishments I encountered along the way.

My idea for this assignment was to create a functioning social media page which allows multiple different users to write and create short messages which are then posted to a wall or 'feed' and remain there for the other users to view. I wanted to allow the users to be able to include a personal name and location beside each of their posts so that every user is aware of who created each message. Each user would be able to have an account as such with their information which they submitted and a profile picture which they chose from a selection of images and any post created would be linked to their account. I wanted these accounts to be basic and easily changed so there was no need for setting passwords as that would make the user experience unnecessarily more difficult. This was the baseline idea for my website, and I was able to complete all of these features and allow my website to function accordingly. However, there were additional features I mentioned in my previous report which I decided against or left out due to various reasons I will now explain.

List of features I originally planned to implement:

- *Be able to create a post of text of up to 140 characters.*

  A character limit is necessary to prevent a wall flooded with paragraphs of text. The idea of my website is to be able share and read short updates instead of long blog entries. Too much text can be overbearing.

  This idea was implemented but slightly changed, however. Through testing I realised that 140 characters was excessive and so I therefore reduced the limit to 100 characters.

- *The ability to view previous posts.*

  Previous posts from other users and the current user will be present on the homepage and will remain there indefinitely. This allows users to scroll down, and view posts created at an earlier date.

  I was able to fully implement this feature.

- *Register basic personal details such as name, age, and location.*

  This will allow other users to identify who created each post as well as view some simple information about that person.

  I was also able to fully implement this idea however I removed the feature to add a user's age. I could have easily included this however I thought it was unnecessary and gave the website a much more wholesome feel without it.

- *Store basic user information for 24 hours and automatically clear it afterwards.*

  To prevent security risks, personal information will only be stored for 24 hours after it is submitted. Each user will have to re-enter their information to make another post after 24 hours of it being in the system. This will also help reduce storage space being used over time.

  I was not able to implement this feature but due to time constraints however, I incorporated a URL that would clear the session data if they wished to. I didn't, however, include a button to access the clear session page and so it can only be accessed if the user knows to type '/clear' into the URL.

- *Enable users to reset and clear their information using a button.*

  If a user wishes to change their information or simply clear it from the system, they will be able to click a button which will reset it. They will of course have to re-enter their name, age and location before creating another post. Posts will always show the information that was entered along with that post at that time instead of updating with newly entered information.

  The ability to clear the user's data at the click of a button wasn't fully implemented due to time constraints however the user can change their information at any time using the menu and if they wish to remain anonymous, they can simply not submit anything into the name and location fields and it will clear their previous information and replace it with nothing if they so wish to do so.

- *Choose a profile picture out of a selection of images.*

  The user will be able to choose a small profile picture to go alongside their post. There will be a selection of around 20 images that the user will be able to choose from. This will allow for more personalization and make the page look more colourful and inviting.

  I was able to fully implement this feature into my menu and I am happy with the result of this.

- *A timestamp will also be created at the time of the post creation and will be displayed next to the post.*

  This will allow users to be able to determine when each post was created. This will help to organize posts and can be helpful if you are searching for a post created at a specific time; you will simply be able to scroll and look for the correct timestamp.

  This function was not added due to time constraints. All the created posts will simply display that they were posted 'Just now…' whereas in fact the function doesn't work. As

I originally planned to add this feature, I have the template for it added in my HTML template but as I wasn't able to complete this function in my Flask code the placeholder for the function remains without use.

- *A seamless scrolling effect when reading previous posts.*
  This will serve many benefits such as allowing the user to load only what is necessary at one time and help with loading times. It will also mean the user isn't overwhelmed with having their homepage be extremely long with all the previous posts being displayed at once; this would also make the scroll bar substantially harder to use and create a less friendly user interactive experience.

  In a way this feature exists but not in the way I originally described. At the current moment the page will grow with each added post and the user will be able to scroll down to the first post without accessing another page. However, if there are lots of posts it can cause slightly longer loading times which is what I originally intended to avoid. I instead wanted a 'load more' bar at the footer of the page which would load 10 or so older posts at a time. Instead, I was only able to load all the posts at once. This was a complex feature and an additional one that had low priority as it doesn't change much of the website's functionality however if I had much more time, I would have attempted to include this. If my website was much more large scale, then this is something I would realistically need to implement otherwise loading hundreds or thousands of posts at once could cause detrimental damage to the usability of the website.

Finally, there was one feature I decided to add which I did not originally discuss in my previous report. As I was developing the feed of posts for my main page, I noticed that it would make the website seem a lot more sophisticated if the user's personal posts were displayed on side of the screen and other user's posts on the other side. I was only able to partly implement this unfortunately and it would be one of the first things added given different circumstances.

Features I would add or improve to enhance, given more time.

There were several features I wanted to add or improve upon during the development of my website which were limited due to the time I had. Some of these features have already been discussed above but I wish to expand upon them in this section. There were four main areas in which I would have liked to improve upon given more time:

Firstly, throughout the development of my website I encountered numerous bugs which costed valuable time to correct or find a way to work around. Most of these bugs were corrected but some remain as I didn't have the time to expend upon them fully. One of the main noticeable bugs is seen when navigating from the menu screen to the index page. When a user creates a post, navigates to the menu screen, and returns it seems as though all of the previously made posts have disappeared until the user creates a new post. When the user creates a new post all of the previously created posts return without issue, but it is quite a large defect with the website. The root to this issue is with the lists I use to store the posts and not being retrieved until the user presses the submit post button. Given more time I would wish to change the way these posts are stored by potentially using a database with SQLite. I believe that would scale the website to become a lot more robust and also allow for all of the posts to be stored and made visible even after the server is restarted.

Incorporating this would also next lead me to want to scale the site in other ways such as creating a seamless scrolling effect to load 10 or so posts at a time instead of having all posts load at once. This would allow the site to be fit for much heavier use and accommodate to lots more users creating lots more posts without significantly decreasing the website's performance in loading times.

Continuing with the theme of scaling my website; in its current state there is no working post timestamp function involved. At the current time user's posts are temporary as they are solely stored in lists which reset when the server resets and so they are not present for long. If I were to incorporate a much more permanent approach to the website, I would want each post to have a timestamp next to it so that we know when it was created. I believe this wouldn't be a very difficult task and I believe given an extra day I could make this happen, however due to time constraints I opted to leave it out.

Finally, one final feature that I wasn't fully able to implement was having one side of the news feed feature the personal user's posts and the other side feature other user's posts. Currently all new posts are created on the right-hand side of the screen but given more time I would segregate the different user's posts and have them appear on opposite sides of the screen much like an instant messaging screen. This would be one of the very first added features that I would implement given more time.

The challenges I faced and achievements I made during this assignment.

During the development of my website, I spent hours of valuable time working around and fixing bugs which caused me to have to skip on features I originally planned to implement. Learning how to deal with these bugs however, taught me a great deal about Flask and Python. I found that when I encountered a bug or error, I went away to research the cause and solution of it and experienced a large amount of self-learning which was required to solve the problem. At the beginning of development, I was slow to learn the solutions to my problems and lacked the understanding of the root of the problem. As I continued closer to the end of my development however, I notice that I was much quicker in understanding why the error was occurring and therefore was much better equipped to implementing a fix or work around. I believe I learnt the most about the fundamentals of Flask and Python during the researching of the problems that I was having and was taught the most during those times, as frustrating as they were…

Overall, I am very pleased with the outcome of my website given the time that I had, and I am much more skilled in the programming languages that I used to create them now than I was at the beginning. I believe I am fully capable of building bigger and more sophisticated Flask deployment apps or websites with the skills I have now, and I have been given a better understanding of the career I want to take in the future because of it which is a very useful achievement for myself. I hope to use the skills I have learnt during the past few months and especially with the experience gained from the development of this website to further my career in the future and open many more doors for me…