

Website Coursework 2 Report

Lewis Jamieson

40273123@live.napier.ac.uk

Edinburgh Napier University - Advanced Web Technologies (SET09103)

1 Title

Team Chat - Gaming Central

2 Introduction

Team Chat is a video game themed message board website with a navigation bar, which gives the user the ability to login(Figure 4), register(Figure 5), logout and edit their profile. The website catalogs discussion threads (Figure 2). It is based on message boards like Reddit. The user has the ability to look at what other people have posted, post messages on each board, register an account, login with an account and logout. Posting on each messageboard is done by a textarea input and a button. In profile users can change their user-name, password and avatar. Profile editing is done through text boxes, an image input and a button (Figure 6). There is also a custom 404 error page.

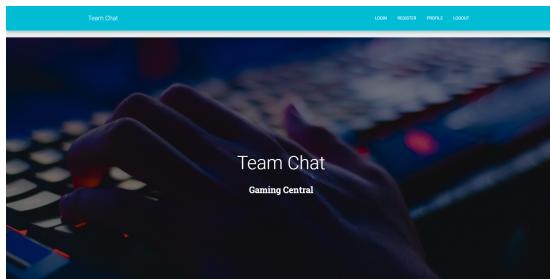


Figure 1: Home Screenshot - A screenshot of the home page showing the navbar

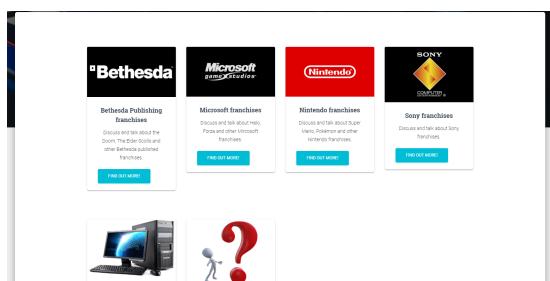


Figure 2: Home Screenshot 2 - A screenshot of the home page showing the message board discussion threads

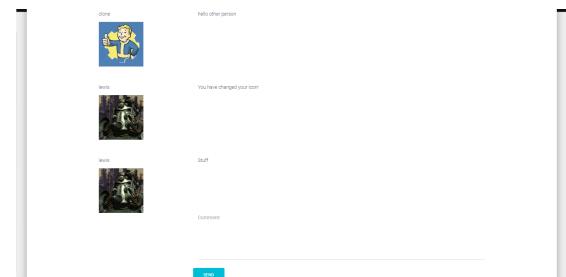


Figure 3: Message board Screenshot 2 - A screenshot of the other message board showing multiple users and textarea input

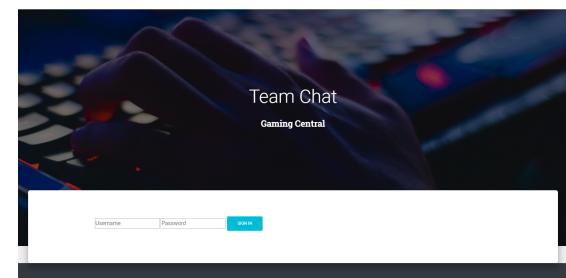


Figure 4: Login Screenshot - A screenshot of the login page

3 Design

The website is linked together by a free version of a bootstrap called material kit made by Creative Tim. [1] The website has a navigation bar which links to the pages for home, login, register, profile and the logout button(Figure 1). The navigation bar can be accessed from every directory. From the home page, users go through each topic. Selecting Bethesda brings the user to the Bethesda specific board. Microsoft brings the user to the Microsoft specific board. Nintendo brings the user to the Nintendo specific board. Sony brings the user to the Sony specific board. PC and indie brings the user to the PC and indie specific board. Other brings the user to the nonspecific, off-topic board. When clicking on a board they are redirected to the board where messages are stored in a database, this was done so that the messages users input are stored and do not disappear when the website is closed. In each message the message body, user name, group in which the board is and the avatar are stored. This is done so that when the specific message board is opened the messages show in the specific format as shown in (Figure 9). Each user's user names, passwords and avatars are also stored in a database. Selecting Register brings the user to the registration page where users can create an account by inputting a user name and password, which is then in-

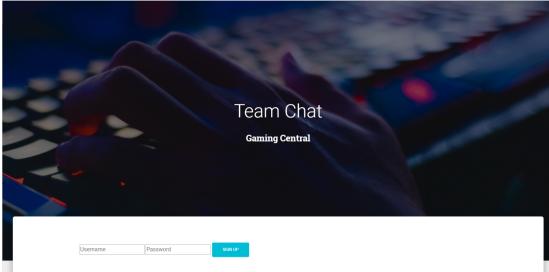


Figure 5: Register Screenshot - A screenshot of the register page

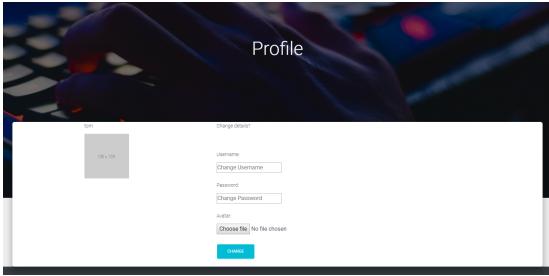


Figure 6: Profile Screenshot - A screenshot of the profile page showing the text boxes, image uploader and submit button

serted into the database. If the name is already taken, the user will be told. After registering users will be redirected to the login page. Selecting Login brings the user to the login page where users can log into an account by inputting a user name and password, which is then search against the database. If the user tries to access a directory which does not exist they will be redirected to a custom 404 HTML page, which has the navigation bar allowing the user to go back to the homepage without clicking the go back button in their browser (Figure 7).

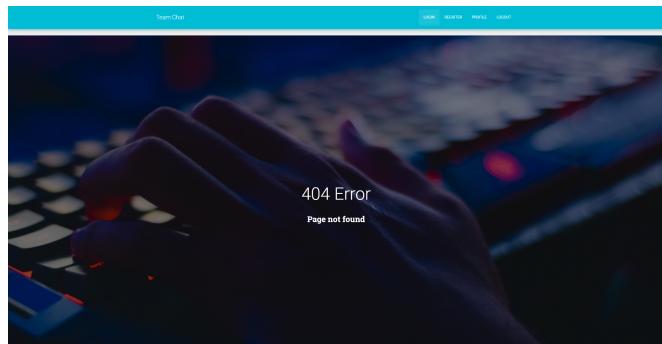


Figure 7: 404 Screenshot - A screenshot of the website's 404 error

4 Enhancements

The website could be improved. If the user had the ability to create new threads more users would use it because people might want to talk about different things within the topic of the games. For example when accessing the Nintendo topic users would be greeted with a table of entries and for example have access to a Super Mario series entry and a Legend Of Zelda entry, etc. Implementing this would be easier if the

Hierarchy

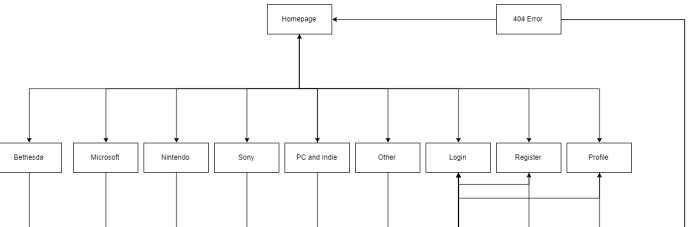


Figure 8: Hierarchy Sketch - A sketch of the website's hierarchy

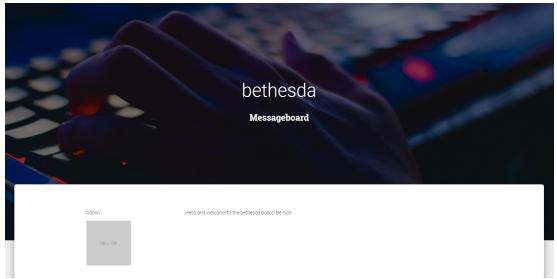


Figure 9: Message board Screenshot- A screenshot of the Bethesda message board

website was changed, so that it used JSON format files and to edit the table there would be an edit mode which would probably use Javascript. User would be able to add their own entries, but the ability to delete would be restricted to admins. Each change would be sent to the python, saved on the JSON files which the directories to would be saved in a database. Having a like/dislike for each message in the top corner would have allowed logged in visitors to vote whether or not they liked or disliked about the message. This would increase user activity on the site as there would be more to do than just messaging each other. Adding an image or a logo to the website would make the website look more appealing which would make people more likely to use the website. If the navigation bar showed the user avatar in the top right corner when the current user is logged in, users would be more likely to use the site as it is clearly shown that they are logged in. This is unlike as it is currently as the user has to try to input a message or access the account page to see if they are signed in. Giving the user the ability to input images in their messages would be a good enhancement as it would give more context to a users message. Having individual pages would be good as currently the user posts could go on infinitely, possibly using up a lot of data. Having pages so that only twenty are stored on each page would prevent this issue and improve navigation.

5 Critical Evaluation

The home and sub-directories are shown using "home.html", the data itself is stored on the Python file "CWsite.py" and the database file "database.db". Data from the database is sent to the HTML using parameter passing into a for loop [2]. It worked out well because if parameter passing was not used, it would not be able to use data stored on the

database and permanently storing messages would probably have been impossible. For passing data from the python file, it would have to manage multiple HTML files for each message board and if a link was add to the navigation bar that change would have to happen to every HTML. Using the parameters to pass information from the database forms the message boards. Each message board consists of the data being pass and put into a easier to read form using grids [3]. It was a good idea as it makes the page more visually appealing and combined with the for loop automates the HTML design process. The user avatar is shown in the message to prevent the image being larger than the message, the width and height are set to fixed values in the HTML [4]. The Profile page uses two text forms, one image and a button. The first text form is to change the user's name, it is used in the file input naming process. The second text form is for changing the password, so if they want to make their password more secure, they have an option. The file input is for uploading an image to change the avatar (Figure 10). Changing these details are done using the sql UPDATE function [5] and the current user session, which holds the current user's name for updating where user is equal to the current user. The button is for submitting the text forms and image. The request flask object was used to get the input forms from the HTML to Python. When submitting the image, it is saved as a "png" file and the name is used as apart of the name of the file. An if statement checks wherever or not a request to change user name and password or change avatar, so users don't have to change all of their details.

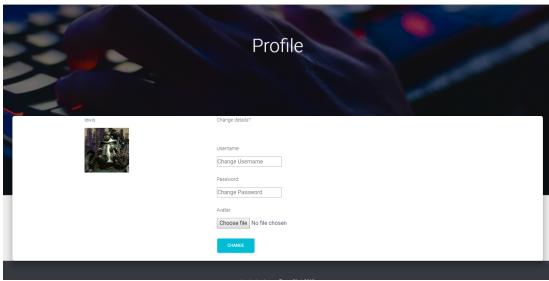


Figure 10: Profile Screenshot - A screenshot of the profile page showing a changed avatar

6 Personal Evaluation

In reflection, the most difficult issue I had was the database as getting variables sent from the user were not strings as I initially thought, I overcame this issue where I had found online that the input from a HTML is not declared and should be referenced as a "?" type. I had issues with time management if I had better time keeping I would have probably implemented the ability to add images to messages. A burn-down chart would have improved my time keeping as I would know if I was behind schedule or not. I also had issues with the sessions. The issue itself was that if they were not declared before being used in an if statement the website would crash and declared is not the same as none in flask. I was using if statements to prevent non-registered users from inputting messages and accessing the profile, also I was storing the user's name in a session, so that I can reference it when

require. I overcame this issue using try and except[6] to catch the error and prevent the site from crashing. I had learned from the first coursework to start with the bootstrap this time, so I did not have the annoying issue of changing each HTML. Thanks to the background image [7], the website is not as previous coursework. Overall I would say this shows a better understanding of website design than the previous website, thanks to the sessions and the database.

References

- [1] CreativeTim, “<https://www.creativetim.com/product/material-kit>.”
- [2] tutorialspoint, “https://www.tutorialspoint.com/flask/flask_sqlite.html”
- [3] w3schools, “https://www.w3schools.com/bootstrap4/bootstrap_grids.asp”
- [4] w3schools, “https://www.w3schools.com/html/html_images.asp”
- [5] tutorialspoint, “https://www.tutorialspoint.com/sqlite/sqlite_python.html”
- [6] pythonforbeginners, “<https://www.pythonforbeginners.com/error-handling/python-try-and-except>.”
- [7] S. Kumar, “<https://www.pexels.com/photo/photo-of-person-typing-on-computer-keyboard-735911/>.”

7 Appendices

Images were obtain from Various Sources:

Somosxbox: <https://www.somosxbox.com/bethesda-adquiere-battlecry-y-forma-un-nuevo-estudio-en-texas-774300>

Digitaltrends: <https://www.digitaltrends.com/gaming/new-studio-black-tusk-making-microsoft/>

Wikipedia: <https://ru.wikipedia.org/wiki/>

Globalmedia-it: <https://globalmedia-it.com/sony-computer-entertainment-adquirira-una-seleccion-de-activos-de-onlive/>

Zdnet: <https://www.zdnet.com/article/its-time-to-stop-obsessing-over-pc-sales/>