Beat4Beat

Robbie McNeil
40296075@live.napier.ac.uk
Edinburgh Napier University - Advanced Web Tech (Set09103)

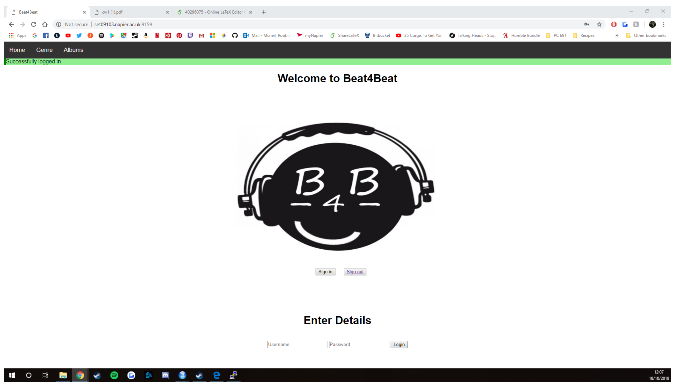
Abstract

This report is a brief view into the development of my Flask web app, Beat4Beat, a music based app which contains information on various albums and their artists. I will explore the design, what enhancements I would make given more time, as well as evaluations of the final web-app and myself.

Keywords – Python, Flask, HTML, CSS, Javascript, Web App, SSH, Linux

1 Introduction

Beat4Beat was created with the intention of making a music encyclopedia of sorts where users would be able to come to discover new music or artists which they might want to listen to. I decided upon this idea as I often use music reviewing websites and videos as a way of discovering new music that I otherwise wouldn't hear and thought it would be fun to give my own opinions on some of my own favourite albums. When you open up Beat4Beat you are greeted at the homepage, as seen in figure 1, this has little functionality, and is intended to just greet the user and act as a log-in/log-out page if they have an account. Initially the username and password textboxes are hidden along with the Login button, users can press the Sign in button below the logo in order to reveal the Login features[1]. Once the user signs in or out they are displayed a corresponding message at the top of the screen informing them that they have logged in/out.

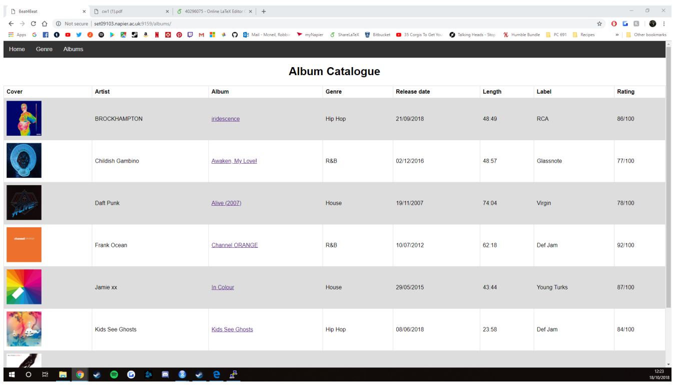

Successfully logged in

Welcome to Beat4Beat

Enter Details

Figure 1: **Beat4Beat Home screen** - showing a log in message and displaying log in features

From the home screen the user can navigate to one of three pages: Genre, Artist or Albums. The Albums page, shown in figure 2, features a table listing all of the albums reviewed on Beat4Beat, users can easily find the artist, album name, what genre the album falls into, the release date, the album length in minutes, the label that the album was released under and the rating as well as the album's cover art. Each album title is a hyperlink and the user can click the album name to be taken straight to that album's review page, which includes most of the same information as the table (in more detail) but also includes a review and an embedded spotify player. This was implemented as a way for users to quickly and easily find what they want if they already know what they might be looking for.

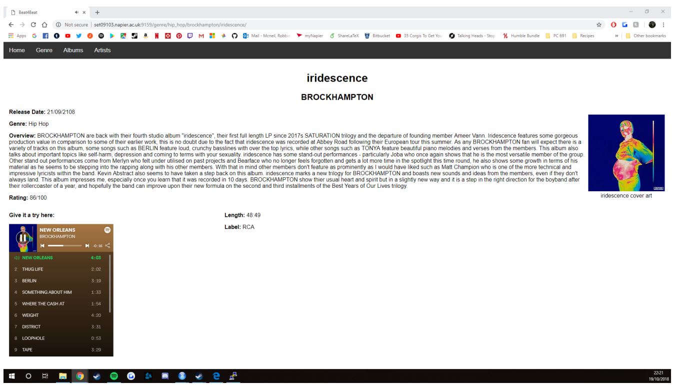

Album Catalogue

Cover	Artist	Album	Genre	Release date	Length	Label	Rating
	BROCKHAMPTON	iridescence	Hip Hop	21/09/2018	48:49	RCA	86/100
	Childish Gambino	Awaken, My Love!	R&B	02/12/2016	48:57	Glassnote	77/100
	Daft Punk	Alive (2007)	House	19/11/2007	74:04	Virgin	78/100
	Frank Ocean	Channel ORANGE	R&B	10/07/2012	62:18	Def Jam	92/100
	Jamie xx	In Colour	House	29/05/2015	43:44	Young Turks	87/100
	Kids See Ghosts	Kids See Ghosts	Hip Hop	08/06/2018	23:58	Def Jam	84/100

Figure 2: **Album Catalogue**

The Artist page shows a grid of all the Artists on the web-app, each artist has a given profile picture and their name underneath. Each row in the grid has 4 artists, and clicking on an artist's picture will direct you to that artist's profile page which includes a photo of the artist, a brief overview of their musical career and the artist's Discography. The Discography section includes hyperlinks to any albums which have been reviewed on Beat4Beat, clinking the link will take the user to that review.
For users who may not know what or who they're looking for they can use the genre page. Although the genre page also ultimately leads to the same review pages as the album table and artist page, the user is first given a choice of genres to choose from (currently hip-hop, house, indie rock or R&B). Once the user picks their desired genre they are taken to that genre's page which displays images and names of different artists which fall under that genre category. Users can click on an artist they find interesting or want to know more about and they are taken to that artist's profile page, just like in the Artist page. From here the user can explore that artist's profile and their albums.
The album and artist pages take up most of the web app, with eight artists included and each artist having one album

reviewed, if I had time to complete all of the eight included artist's albums then the number of album pages would increase to twenty five.


Beat4Beat
Not secure | set09103.napier.ac.uk:9159/genre/hip_hop/brockhampton/iridescence/
Home
Genre
Albums
Artists

iridescence

BROCKHAMPTON

Release Date: 21/09/2108

Genre: Hip Hop

Overview: BROCKHAMPTON are back with their fourth studio album "iridescence", their first full length LP since 2017s SATURATION trilogy and the departure of founding member Ameer Vann. Iridescence features some gorgeous production value in comparison to some of their earlier work, this is no doubt due to the fact that iridescence was recorded at Abbey Road following their European tour this summer. As any BROCKHAMPTON fan will expect there is a variety of tracks on this album, some songs such as BERLIN feature loud, crunchy basslines with over the top lyrics, while other songs such as TONYA feature beautiful piano melodies and verses from the members. This album also talks about important topics like self-harm, depression and coming to terms with your sexuality. iridescence has some stand-out performances - particularly Joba who once again shows that he is the most versatile member of the group. Other stand out performances come from Merlyn who felt under utilised on past projects and Bearface who no longer feels forgotten and gets a lot more time in the spotlight this time round, he also shows some growth in terms of his material as he seems to be stepping into the rapping along with his other members. With that in mind other members don't feature as prominently as I would have liked such as Matt Champion who is one of the more technical and impressive lyricists within the band. Kevin Abstract also seems to have taken a step back on this album. iridescence marks a new trilogy for BROCKHAMPTON and boasts new sounds and ideas from the members, even if they don't always land. This album impresses me, especially once you learn that it was recorded in 10 days. BROCKHAMPTON show thier usual heart and spirit but in a slightly new way and it is a step in the right direction for the boyband after their rollercoaster of a year, and hopefully the band can improve upon their new formula on the second and third installments of the Best Years of Our Lives trilogy.

iridescence cover art

Rating: 86/100

Give it a try here:

Length: 48:49

Label: RCA

	NEW ORLEANS	4:03
2	THUG LIFE	2:02
3	BERLIN	3:19
4	SOMETHING ABOUT HIM	1:33
5	WHERE THE CASH AT	1:54
6	WEIGHT	4:20
7	DISTRICT	3:31
8	LOOPHOLE	0:53
9	TAPE	3:29

Figure 3: **Example of an album review**

2 Design

I wanted to design my web app in a way that was fairly intuitive to use and users could automatically pick it up and find something they were looking for quickly and with ease. The URL hierarchy was designed with this in mind, from the homepage (the '/' route) you can navigate to 3 areas, '/genre/','/artist/' and '/album/'. From these starting out points the user can find their way around the web app using the following structure:

- /genre/[genre]/[artist]/[album]/
- /artist/[artist]/
- /album/[album]/

where [genre], [artist] and [album] are the desired queries respectively. I decided to make this the general structure so that anybody who wants to quickly find a potential artist or album can easily do so as long as they remember this basic rule. If a person wanted to search for Queen for example they would just have to type in the URL + '/artist/queen/'. This makes it easy for anybody to quickly search for any album or artist straight from the URL bar, without needing to even be on the web app. As for the /genre/ route, I knew that some users don't know exactly what they're looking for at first, and are simply looking for something new that they haven't heard of. I decided to split the search into genre, artist then album as when people ask what kind of music you like you generally reply with a genre or multiple genres, as most people have a broad music taste. Then you may give an example of a particular artist within that genre and then you would give particular albums or songs. Because users will already know what genres they enjoy this will make pinpointing new music and artists they enjoy much easier and quicker than if they just had an album section or an artist section.

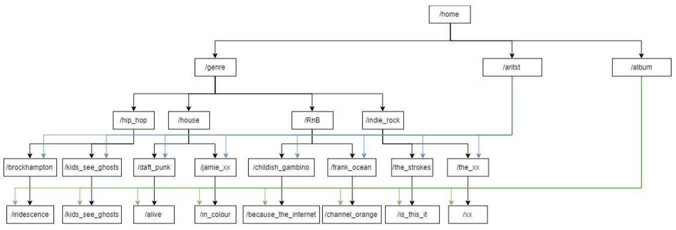


Figure 4: **Beat4Beat URL Hierarchy**- this map shows a complete list of possible URLs using the routes which are currently in the web app

The web app features a number of templates which are used by the various pages to generate HTML layouts. All templates inherit from a template called 'header.html' which is used to give every route a similar feel by keeping a consistent layout throughout the web app. header.html contains the code which generates the top navigation bar and also includes code for a number of other features used in the various pages. This includes making a table[2], aligning images next to each other[3], placing captions under images, and the log-in/log-out message border. This is not only useful for keeping things consistent but also speeds up development as many routes have the same layout but different content. Using templates allows you to just plug the content of a page into a layout without worrying about inconsistencies between related pages.
I wanted to include pictures within the web-app to make the app seem a bit more lively and to add some colour into the mix as without the images there would be little to no colour on any of the pages. I felt it was important to include album cover art (as seen in figures 2 and 3) as this is a big part of an album, but also makes album pages more distinguishable from one another. I also wanted to include pictures of the artists in their respected pages, not only because once again it adds a bit of colour and personality, but a name alone isn't a lot to go off of when choosing to listen to a artist you've never heard of before. Giving users an image of an artist may be the difference between them listening to an artist and not. The choice to add the embedded spotify player which plays small clips from each song came very late in development, however I feel like it adds something a bit extra to the table and gives the user something else to do.

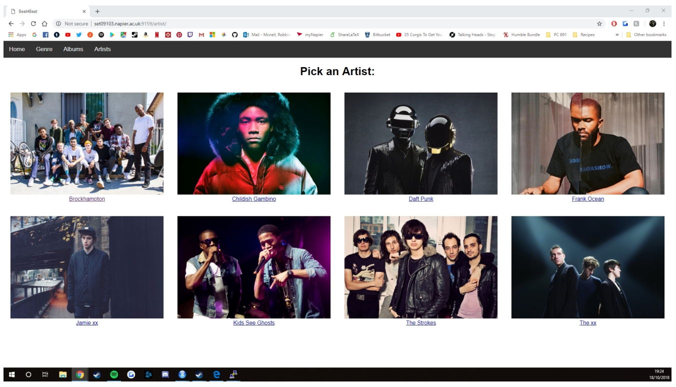
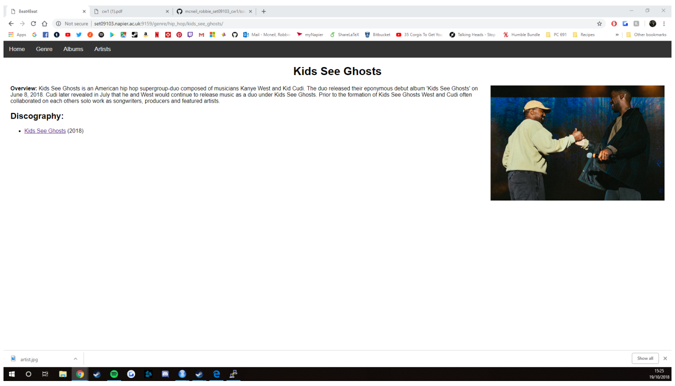


Figure 5: **Artist page** - this shows the use of artist pictures in the web app

3 Enhancements

As I said in the introduction if I had more time to work on Beat4Beat I could increase the number of album reviews included in the web app as well as branch out into different genres and artists. This would be a fairly easy thing to do as all albums and artist pages are derived from templates meaning that there is no design element to consider, all that would need to be done is find the correct images and album/artist information. I believe this would make the web app feel less bare bones, because at the moment there is not much content to browse, and a simple change such as adding more album reviews would make Beat4Beat less empty.
On top of this I would like to give the web app more personality by adding to the layout/design of the app routes. A change in font, an adjustment in layout and perhaps a touch of colour could make a big difference. In places the design may feel too minimalistic, with a majority of the site being in black and white and many routes having a lot of empty space. For instance, figure 3 shows the review for an album named 'iridescence', but to the right of the length and label there is a lot of space that isn't being used, this is the case with most album pages. To fix this issue I might use Bootstrap to add a simple and consistent style to Beat4Beat and add a larger margin along either side of the contents. This way I could potentially add a bit of colour to the side of the app and force all the contents of the page to be closer together - which would take up more space within the page, making it appear less empty.
An easy implementation I could add would be links to other websites outside of Beat4Beat. ratings could be linked to the metacritic page which they are based off of or artist pages could have links to websites, merchandise stores, articles, wikipedia pages etc. Artist pages are definitely bare and could be improved upon. One idea could be to add an album gallery rather than have just one photo of an artist or perhaps an embedded video of their most popular song. Of all the pages the artist profiles feel like they could use a shake up as they feature quite a small amount of data, and some artists such as Kids See Ghosts (see figure 6 below) are so new onto the scene that there is very little to write about, so adding extra content could improve these artist's profiles massively.


Beat4Beat
cw1 (1).pdf
mcneil_robbie_set09103_cw1/so
Not secure set09103.napier.ac.uk:9159/genre/hip_hop/kids_see_ghosts/
Home
Genre
Albums
Artists

Kids See Ghosts

Overview: Kids See Ghosts is an American hip hop supergroup-duo composed of musicians Kanye West and Kid Cudi. The duo released their eponymous debut album 'Kids See Ghosts' on June 8, 2018. Cudi later revealed in July that he and West would continue to release music as a duo under Kids See Ghosts. Prior to the formation of Kids See Ghosts West and Cudi often collaborated on each others solo work as songwriters, producers and featured artists.

Discography:

- Kids See Ghosts (2018)

Figure 6: **Kids See Ghost artist profile**

Other than the design there are a few additions I would have liked to make, for instance Initially I would have liked to implement a fully functioning user registration form, however, in order to do this you need to install the flask-login extension [4] (which I was not able to do due to the coursework specification). As well as this I would have liked to be able to add a comment section to album pages that would allow the user to post a comment, discuss and give their own opinion on albums, making Beat4Beat more forum-like. The reason I would like to implement this is because it would add an extra bit of interactivity to the web-app, which may keep users interested and more likely to return if they feel like they can have their say.
Finally, a final feature I would like to improve upon is the table in the album catalogue page (as seen in figure 2). At the minute the table is static and ordered alphabetically by artist. As it is there isn't much wrong with this, however, if there was hundreds of artists or albums included then searching for a specific album could prove to be problematic. If the user was able to sort by all the categories in the table (name, artist, genre, release date, length, label and rating) by ascending or descending then this could fix this problem. Another option would be to just add a a search bar into the navigation bar that would allow users to quickly find any album or artist they want, both these features are common on many music streaming services such as Apple music and Spotify and I feel would be a great addition to my web app.

4 Critical Evaluation

Although I feel like there are many areas I could improve upon with Beat4Beat there are many areas I feel work well. In order to evaluate my web app I will go page by page and discuss any features I feel work well and any features I feel don't work well.
I'll start off with the homepage which I feel has room for improvement. The homepage includes the sign in button which when clicked shows or hides the sign in form. The sign in function works but not as I hoped, the application checks the username and password textboxes once the login button has been pressed, if the username and password are not correct then an error message is shown. Otherwise a green banner is displayed at the top of the screen letting the user know they are logged in. If the sign out button is pressed then a red banner is shown along the top of the screen and the user is informed they have been signed out. The problem with this is there is no way to add any new users and the one user which is able to sign into Beat4Beat isn't really a 'user'. Signing in has no real effect on the web app and adds no extra functionality and hence might feel completely unnecessary.
The /genre/, /genre/[genre]/ and /artist/ routes all have a similar structure, where each genre or artist is represented by a picture and a name directly below it. In all of these routes both the picture and the name are links to the next page, I felt as if this was a small but welcome addition, as it gives the user a greater area to click on. These pages are pretty simple and self explanatory, they could be improved by having more genres and artists to choose from, but they all work as intended. I made sure that when making these pages all images where the same aspect ratio so they could sit nicely next to each other and all the images and text would line up next to each other, this gives the web app a much cleaner and polished look.

When making the /album/ route I wanted to give the user as much information on each album as possible, so I decided a table would be most appropriate as I used a spreadsheet in excel to keep all of the album data in one place and it worked well for me. At first I didn't include the album cover art in the table but this made the page quite boring, there was no images, it was just text. Adding the cover art was definitely a big improvement for this page. Not only that but originally not only were the albums hyperlinks but all the artists and genres were hyperlinks too. I removed these as hyperlinks as I believed it took away from the focus of the page (and I already had pages for artists and genres, so having the hyperlinks here seemed redundant). Although it can be argued that taking these hyperlinks away takes away functionality of this page overall I feel the page is better because of it, and if I did not take these hyperlinks away then the /genre/ and /artist/ pages may be obsolete. On the whole I feel like the /album/ page is one of the best on Beat4Beat as it has images, plenty of different kinds information about the albums on the web app, and there isn't much white space where there is nothing going on like there is on some other pages. Despite the fact that the artist profiles do as they are intended to do I would have liked for them to at least have a bit more functionality as they're just a bit plain as they are.
When designing the album review page I wanted to include a few things, namely the album cover art, a track list, a review, a rating and various other pieces of information. I decided to place the image on the far right of the screen with a caption underneath explaining what the image was for those unfamiliar. I am happy with how the album pages turned out, despite the fact that these pages have little for the user to do, they are meant to be information page and were not designed with having a huge amount of functionality in mind. I later decided to add embedded links to spotify and remove the track list, this way the user can get a brief snippet of each song, this was a simple addition to put in as all the work is pretty much done when you request the embed code from spotify. It also adds a bit more functionality for the user.

5 Personal Evaluation

As I went into this coursework I was slightly worried at first because I have a very limited knowledge of HTML and little to no experience with CSS, Javascript and Python. My main concern was design, I didn't feel confident in my ability to use HTML, CSS and Javascript in order to create good looking layouts. Because of this I felt it necessary to start as early as possible.
Firstly I needed a way for users to quickly navigate back up the web app without editing the URL or pressing the back button in their browser. I decided on a navigation bar along the top of the screen, as I've used these in the past when creating other applications and they're usually very effective. Luckily I found code for a navigation bar on w3schools[5], however, this method required a fair bit of code and typing the code out by hand in every layout file would have taken way too much time. To get around this I used a method called template inheritance from Chapter 7 in the module workbook to create a header.html file that allowed me to create one header that other layout files inherit from, meaning every layout page can have the same code in it's head tag. This allowed me to write out the navigation bar code only once but include it on every page.
I started by making the album pages, as they make up a large majority of the web app. I followed chapter 7 from the module workbook to create a simple layout. In order to create the right-aligned cover art with cation I placed code in the header.html file that would take a paragraph with the id 'imgwithcap' float on the right of the screen and any image which was placed within that paragraph would be centered above the caption. To create the track list I used the technique in chapter 7, section 7.0.3 of the workbook to loop data from the python file into the template file to create a numbered track list (as I mentioned earlier however this feature is no longer present). Development of these album pages were fairly easy, but the first major problem I ran into was the fact that the album pages were taking an extremely long time to complete due to the fact that I was being extremely meticulous when writing my reviews and opinions of the albums. In my original planning I had set out to do twenty albums, however early in development I realized that if I was going to meet the deadline this number was completely unrealistic, and so I trimmed down the number of albums and decided I would return to them after development if I had any spare time. I also decided to be less particular with how I was writing the reviews as the reviews themselves were not what I was going to be marked on.
When making the /genre/[genre]/[artist]/ routes I was unable to find a way to have an artist's discography printed out in a list with hyperlinks to corresponding albums. I could have made a list similar to the track list, however, I wanted to have listed albums link to their album page. Because I couldn't find a way around this I knew I would have to create a separate template for each artist which would be painstakingly slow to code by hand and would probably lead to some inconsistencies in layouts. To get around this I made one general layout for the artist profiles then copy and pasted that file for each artist. So I had 8 artist layout files that where all exactly the same, then I started adding artist info to each one until I had 8 unique but cohesive artist profiles. Although this may not have been the most graceful way around my problem, I did find a solution and it did work, despite the method being quite tedious.
Another challenge I had to overcome was adding different borders around flash messages. I figured out how to implement the flash messages initially from Chapter 8 in the module workbook. Once the user logged into Beat4Beat they received a message in a green border saying they had logged in. Upon logging out the user received a message in the same border telling them they had logged out. What I wanted to do was add different colours to these messages, with signing in having a green background and singing out having a red background. I played around for a while and tried to find a solution but could not figure it out, luckily I found some documentation on Flashing with categories[6].
Outside of HTML, CSS, Javascript and Python I took the time to learn about git and how to use it properly. Previously I haven't used git as I found it confusing and didn't quite understand how it worked. In the past it seemed like more of a hindrance because I could never figure out how to push or pull anything, however this time around I fully embraced git. At the time of writing this I have made over 30

commits and even made more than one branch in order to create the log in feature. The log in feature was somewhat of a risk I decided to take as I wasn't sure of how well (or if) it would work and I didn't want it mess up any existing code, hence the decision to create a new branch. I'm still not perfect with git and did struggle here and there using it, especially when trying to merge branches however I was always able to find a solution and feel way more confident when using git now.
Overall I feel like I performed well with this coursework. I did not feel overly confident with any of the languages I was using going into the task but I feel as if I have made huge progress with HTML and Python. My web app turned out more or less the way I envisioned when planning and I learned numerous things throughout the coursework. If I had to do the task again I would spend more time on the layout of the pages and look into Bootstrap.

References

[1] "Html hide/show an element." `https://www.w3schools.com/howto/howto_js_toggle_hide_show.asp`.

[2] "Html table tag." `https://www.w3schools.com/html/html_tables.asp`.

[3] "How to align images next to each other using css." `https://www.w3schools.com/howto/howto_css_images_side_by_side.asp`.

[4] "Flask login documentation." `https://flask-login.readthedocs.io/en/latest/`.

[5] "How to create a navigation bar using css." `https://www.w3schools.com/howto/howto_js_topnav.asp`.

[6] "How to create flash categories." `http://flask.pocoo.org/docs/1.0/patterns/flashing/#message-flashing-pattern`.