Programming a music loaning and catalogue website

Adam Bilkus

Candidate Number: 3085

Centre Number: 12290

Table of Contents

[1-Analysis 3](#_Toc21939113)

[1.1-Introduction 3](#_Toc21939114)

[1.1.1-Background 3](#_Toc21939115)

[1.1.2-Project Scope 3](#_Toc21939116)

[1.1.3- The Client/Supervisor 5](#_Toc21939117)

[1.1.4- The Prospective Users/Audience 5](#_Toc21939118)

[1.2-Investigation 6](#_Toc21939119)

[Existing Websites 6](#_Toc21939120)

[Questionnaire 7](#_Toc21939121)

[Prototyping 11](#_Toc21939122)

[Neural Network/Suggestions Algorithm 12](#_Toc21939123)

[1.3-Constraints/Hardware 16](#_Toc21939124)

[Hardware Constraints 16](#_Toc21939125)

[Software Constraints 16](#_Toc21939126)

[User’s Knowledge of Information Technology 17](#_Toc21939127)

[Access rights 17](#_Toc21939128)

[1.4 Limitations 17](#_Toc21939129)

[1.5-Objectives 18](#_Toc21939130)

[General Objectives 18](#_Toc21939131)

[Specific Objectives 18](#_Toc21939132)

[Optional Objectives (for if I have time) 19](#_Toc21939133)

[1.6-Proposed Solution 19](#_Toc21939134)

[Privacy Policy 20](#_Toc21939135)

# 1-Analysis

## 1.1-Introduction

### 1.1.1-Background

I will be creating a music loan website for Barnet Education Arts Trust. This will be based off library loaning online systems such as Barnet Libraries. Library Management systems are a way of keeping track of stock in a library. The first one was created over 20 years ago and hasn’t changed much.

Library management systems have been based off of stock tracking websites and let you know where items are and when they are going to be returned or if it is a shop when the stock is going to be delivered.

When I have finished coding my website, I will test it by giving it to my orchestra conductor to use for his music and test it for all of his bands. I will also test it across different operating systems and different devices.

I will be investigating how to send automatic emails from a server, how to record music, how to get ISBN numbers, possibly how to scan barcodes to get the code from it.

### 1.1.2-Project Scope

I will aim to finish this project over the course of several months. This table gives me a rough outline of every task I am going to need to complete, the time it will take, and the date it needs to be done by

|  |  |  |  |
| --- | --- | --- | --- |
| # | Task | Duration | Due Date |
| 1 | Introduction | 2 |  |
| 2 | Investigation | 2 |  |
| 3 | Constrains, requirements, limitations | 2 |  |
| 4 | Objectives | 1 |  |
| 5 | Proposed Solution | 1 |  |
| 6 | Finish Analysis |  | 13/10/2019 but add final changes before final deadline |
| 7 |  |  |  |
| 8 | Data Flow Diagrams | 3 |  |
| 9 | Data Dictionary | 1 |  |
| 10 | Design the database + Normalisation | 10 |  |
| 11 | Create Class Diagrams and class definitions | 1 |  |
| 12 | Pseudocode |  |  |
| 13 | Plan Security Measures |  |  |
| 14 | Plan test strategy |  |  |
| 15 | Finish Documented Design |  | 13/10/2019 added in a few changes along the way |
| 16 | Implement login system | 1 |  |
| 17 | Implement Reservation System | 2 |  |
| 18 | Implement Borrow System | 6 |  |
| 19 | Implement Returning System | 3 |  |
| 20 | Implement Suggestions Page | 10 |  |
| 21 | Implement AI and neural network on Google Cloud | 7 |  |
| 22 | Finish Implementation |  | 09/01/2020 after due to mocks |
| 23 | Test system on different OSs | 7 |  |
| 24 | Test using invalid Inputs | 1 |  |
| 25 | Get feedback from client | 1 |  |
| 26 | Finish Testing |  | 01/02/2020 after due to mocks |
| 27 | Did I meet all my objectives? | 1 |  |
| 28 | Go through feedback on game | 1 |  |
| 29 | How could it be improved | 1 |  |
| 30 | What could have been done differently | 1 |  |
| 31 | What was easy/hard? | 1 |  |
| 32 | Finish Evaluation |  | 11/02/2020 after due to mocks |
| 33 | Hand IN Whole Thing |  | 28/02/2020 |

### 1.1.3- The Client/Supervisor

My clients are Barnet Education Arts Trust and my dad. I have regular meetings with my dad and will meet when I can with a represented member of Barnet Education Arts Trust. Barnet Education Arts Trust are a music charity for the London Borough of Barnet. They run most music schools and lessons in the borough. As part of leading music centres/schools they have a music office. The music office is where all the music is kept when not on loan to conductors at these centres/schools. They don’t have an online system, at the moment, of keeping track of the music so are constantly losing music. For this reason, they have employed me to create this system for them.

### 1.1.4- The Prospective Users/Audience

The website is particularly made for Barnet Education Arts Trust, although it could be used with a few edits for any other music office. It will be designed in an easy way to be edited for whichever company will want to use it.

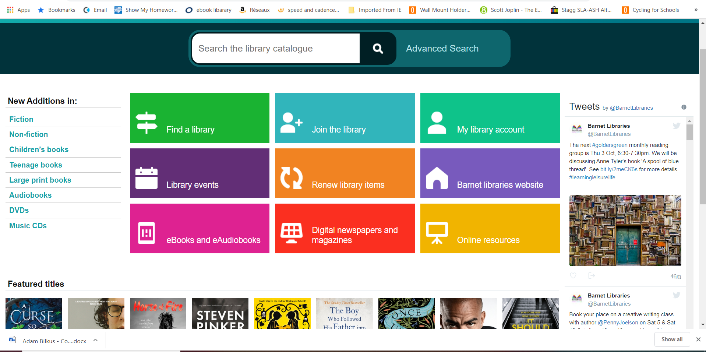
To achieve this, I will use OOP to create objects that can be edited and an admin page that is easy to use. This should make editing easy. It should also make it easy to maintain.

## 1.2-Investigation

### Existing Websites

1. **Barnet Library Page**

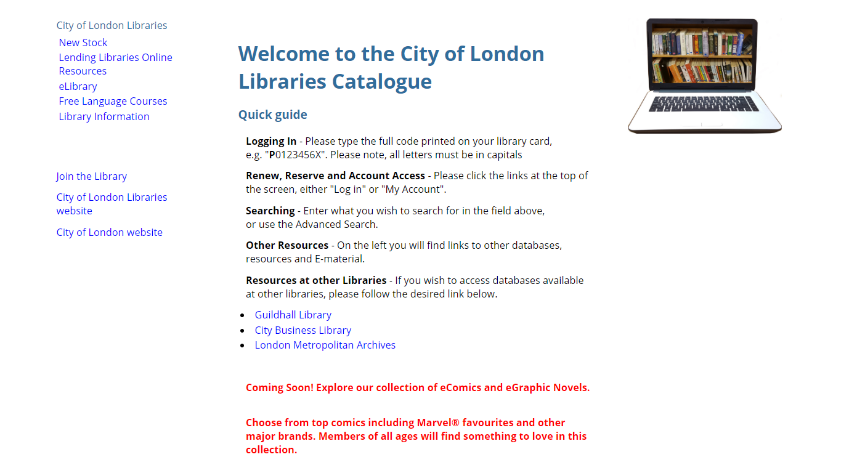
Barnet Libraries Website is the website for the catalogue of books in all the Barnet libraries. The website allows you to renew books and to see all of the books in the libraries, but you can’t reserve books off of it. They show who it is made for, what type of thing it is, new additions.



|  |  |
| --- | --- |
| Pros | Cons |
| -Clear buttons  -Easy to login and renew music  -Shows the new additions for different things | -No way to reserve stock  -Doesn’t show the genres as part of new things. |

In my project, I will replicate the clear buttons and showing new music. I will add a way to see all books. However, I will need to add a way to reserve and return music and show all the genres.

1. **City of London Library Page**



The City of London Library Page is where members of the City of London Library can reserve or renew music. You can also view all the books. It has queues for deciding who gets the books when they reserve it in order. It also has filters so you can filter by author, by year released, by genre, by whether non-fiction or fiction etc.

|  |  |
| --- | --- |
| Pros | Cons |
| -Clear buttons  -Easy to login and renew music  -Can renew Books  -Can reserve books  -Nice Welcome Page  -Multiple Filters | -Doesn’t show the new additions for different things |

I am going to try to copy everything but add in a way of showing all the new music with filters

### Questionnaire

I wrote a questionnaire to find out exactly what my users thought of the current manual loaning system as well as what else they would want from a new online system.

A screenshot of a cell phone

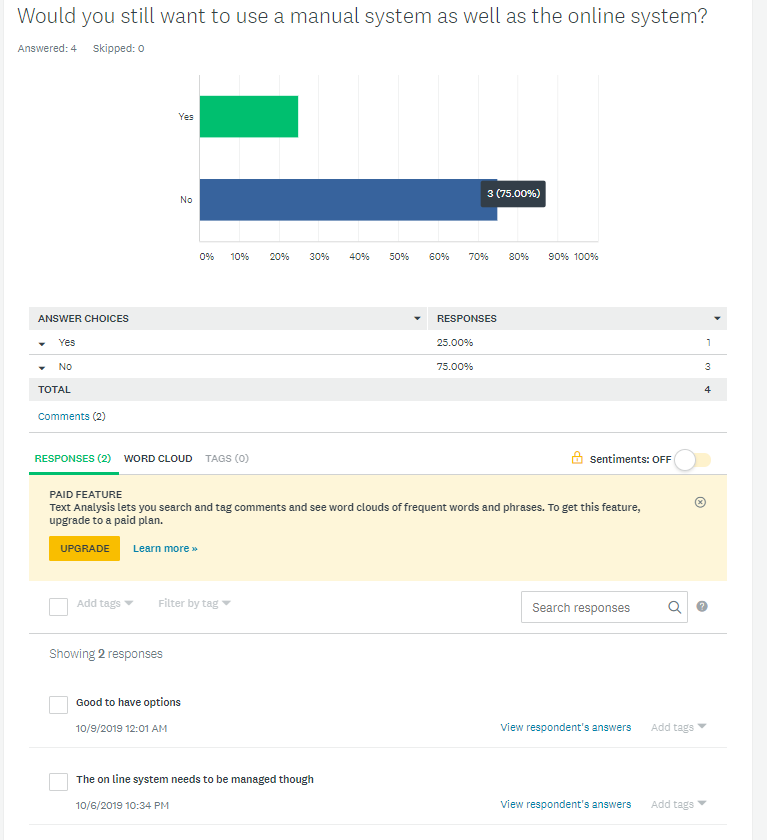
Description automatically generated

All of my users want an online catalogue and reservation system to replace the textbook that is currently in use.

A screenshot of a cell phone

Description automatically generated

Most of my users think that the catalogue part of the website is more important than the reservation system with 2/3 wanting the catalogue so that they can actually see what is available. The problem they have at the moment is having to go in hoping that there is a piece of music, but it may not be there when they arrive. You can’t have the reservation system without the catalogue, so it is the most important part. I will still need the reservation system as everyone in the previous question asked for it and have said it is very useful. Also, this is what will make my project A-Level Standard. The other parts will make the user’s use easier like other library systems.



Most of my users say that they wouldn’t want the manual system. This could be due to getting in the way or being absolutely rubbish. One user wants to still have the option of the manual system. This means that I will need to allow admins/librarians to input a user that is not themselves in the music being reserved and having to have someone go through the book everyday/week to add any music being reserved so that other people who want to reserve music online so that they can see what is still available.

A screenshot of a social media post

Description automatically generated

When I asked what else is needed my users all asked for a way to approximate standard of piece. This will mean looking into the way music is graded and finding out how to grade it. It also means editing the database so that it includes more information on the grading. One other thing asked for is a way for conductors to talk about the music. This will mean having a paragraph box that is allowed to be empty, but I will need to make sure only 1 user can edit this box at a time and set up a queue system.

A screenshot of a computer

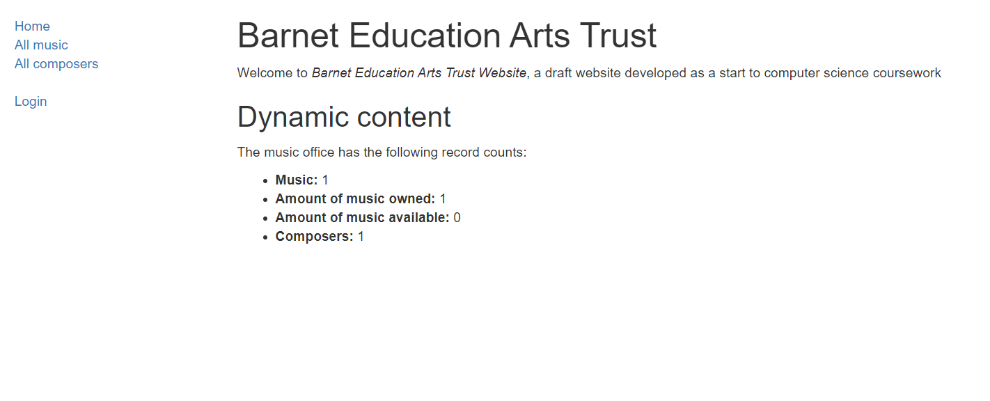
Description automatically generated

One user has said that the current system meets their needs meaning that all they need is a way of seeing current music and letting people know where the music is. They did say though that an online system that can be accessed remotely would be more convenient for them. The other 2 users have said that the system is not good for them. They want a catalogue that is kept up to date about where music is. They also want the capabilities/age recommendations for each piece.

### Prototyping

Over the summer I programmed a neural network in tflearn and TensorFlow to familiarise myself with them and how to create the different layers needed for a neural network. This isn’t doing what I need it to do for the final project, but it was just me learning. It is based off of the tutorial series starting with <https://www.youtube.com/watch?v=3zeg7H6cAJw>.

I also programmed a prototype of my website. This let me familiarise myself with Django. This is what it looked like:



In Django you have to create everything yourself except for the admin page which is created for you. You can edit this page to show you everything. You can use an automatically created view, but this is useless. I did this originally while learning. Each page and each button is based around views. These views can do everything. Pages can be created in views but you would have to pass information from the form to the view so you might as well do it in the view. Filtering is done in a form as it is easier to send the information back. You send information to a html page that will then pass information back to the view.

I have also written a very basic suggestions algorithm that only works on a csv file which I won’t be using. It looks at a user’s previous reviews and finds the most recent one. It then looks at the review and looks for user’s who have given the same review as you and will return you those user’s other reviewed pieces of music. In the final edition it will look at only the current user’s good reviews and then look for users who have similar reviews and return their best reviewed music unless you have borrowed/reserved/cancelled that piece of music before.

Here it is:

import csv

import sys

import random

number = 1

csv\_file = csv.reader(open('reviews.csv', "r"), delimiter=",")

print("Debug")

x=[]

for row in csv\_file:

x.append(row)

def get\_user\_reviews(x):

reviews = []

for i in x:

if i[2] == str(number):

reviews.append(i)

return reviews

def date\_check(reviews):

biggestdate = "01/01/0001"

reviewToUse = []

print(biggestdate)

print()

print()

print()

print(reviews)

for i in reviews:

print("Date Check")

print(i)

print("Test")

if i[3]>=biggestdate:

print(biggestdate)

print(i[3])

print()

print()

biggestdate = i[3]

reviewToUse.append(i)

return reviewToUse

def review\_check(reviewToUse):

print(reviewToUse)

stars = reviewToUse[0][3]

print(stars)

return stars

def music\_check(reviewToUse):

music = reviewToUse[0][1]

print(music + "Hello")

return music

def other\_reviews(number, stars, music, x):

otherReviews = []

choices = []

for i in x:

print(i)

print(str(stars))

print(str(number))

if i[1] == str(music) and i[3] == str(stars) and i[2]!= str(number):

otherReviews.append(i)

choices.append(i[0])

print("Hello")

return otherReviews, choices

def suggest(number, x):

reviews = get\_user\_reviews(x)

reviewToUse = date\_check(reviews)

stars = review\_check(reviewToUse)

music = music\_check(reviewToUse)

otherReviews, choices = other\_reviews(number ,stars, music, x)

print(otherReviews)

FinalChoices = []

if len(choices) > 0 and len(choices) <= 5:

for i in choices:

FinalChoices.append(i)

elif len(choices) > 5:

for i in range(0,5):

choice = random.choice(choices)

choices.remove(choice)

FinalChoices.append(choice)

for i in otherReviews:

for j in FinalChoices:

if i[0] == j:

print(i)

suggest(number, x)

The final edition can work straight on the database and is way shorter as it just uses a few queries rather than having to search through every single bit of the csv. It also links in to a view.

### Neural Network/Suggestions Algorithm

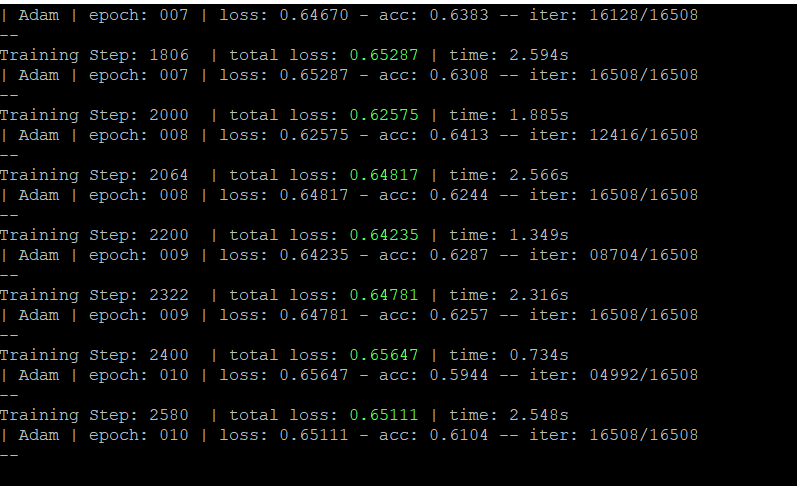
My current Neural Networks algorithms uses different levels. The first layer is the input layer. I then have fully connected levels. In this every neural is connected to every neuron in the previous layer. This uses a lot of memory and computation compared to convolutional layers where each neuron is only connected to a few nearby neurons. On the other hand, the fully connected layers are easier to program. You then train based off random data which over time keeps being used with a dropout rate where only useful data gets stored until it is only perfect and won’t be over trained. This is a possibility for my final AI but probably won’t be what I can use as I won’t have the information to use for it.

Instead of using the neural networks algorithm I learnt over the Summer holidays, I will use a basic suggestions algorithm like what is used by Amazon and Netflix. There are 2 types of suggestions-based algorithms. One is called content-based filtering which is where you look at people have been recently been getting out and recommend music based off of it. The other is collaborative filtering systems which make recommendations based on user interactions. This would be when you see what other people got when they got out what you had.

I will first be building the content-based filtering system as this version doesn’t need the information from other users, so I only need one user. Once a few people are using the system and I have a bit more information stored I can then implement a collaborative filtering system so that I recommend based off of what other people have done.

For this suggestions I will first build a basic AI for content filtering using a tutorial from <https://towardsdatascience.com>

The current neural network program running looks as follows:



The code itself looks like this:

A screenshot of a computer

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

## 1.3-Constraints/Hardware

### Hardware Constraints

Server

I am building a website so it will need to be able to run on a normal server but one that can handle a bit of AI. This means that it shouldn’t use too much RAM or need a too powerful CPU. The specs needed are listed below.

|  |  |
| --- | --- |
| CPU | Intel Xeon |
| Graphics Card | Red Hat, Inc. QXL paravirtual graphic card |
| RAM | 100mb |

PC/Device connecting to website

You will need any machine that can handle a chrome page and has an internet connection.

### Software Constraints

I want to be able to run this on any machine so all they will need is a way of accessing the internet such as Google Chrome or Firefox. I will need to be able to test on different operating systems so I will need to borrow a Chromebook and an iPhone. I have a Windows Phone and a way of running Windows and Mac OS. Eventually, I want to be able to create an app that will be able to run on any Android Device able to run Android Marshmallow or after and on an IOS device running IOS 10 and above. This means that you are getting most devices.

### User’s Knowledge of Information Technology

My users will come from a range of IT backgrounds and may have little to no experience with computers. Therefore, I am aiming to make the website as accessible as possible. To do this, I will think carefully about UI design, making menus and buttons as clear as possible (e.g. using icons and large text). I will also include a chatbot eventually that will deal with any questions about how to use the website.

### Access rights

My users will have different permissions – librarians can: add/remove users; change reserved to borrowed then to returned. Admins can add/remove books, assign people user permissions and edit databases. Users can just see music and reserve available music. If I have time, I may add the option to have other user types.

## 1.4 Limitations

**Areas not touched / areas considered for future**

In the future there are other features I could add including having a link to a YouTube video for each set of music, but I don’t have the time for it and don’t have the expertise to be able to do it at this time. This would mean that the website would constantly be having to check YouTube for the best link and the most watched of the correct video and check the link it has to make sure that it is still working. This would slow the system down and require more RAM which I don’t have the money to upgrade.

I could also include a way for people to recommend music by making links from YouTube suggestions, but it takes a lot of time which I don’t have the time for. It would also mean creating an extra AI to work in the background and training this AI on top of the other would take a lot of resources which I don’t currently have.

## 1.5-Objectives

### General Objectives

My general objectives involve making sure that the website functions correctly, allows logins, allows reservations, allows librarians to change reserved to borrowed, allows returns and moves between pages correctly.

-The website functions as planned. The website has different pages with different links to other pages. The website allows reservations from previously created users. The users have a set group. These users will be able to see different pages depending on the group that they are part of.

-Users are able to be registered by the admin

-When the new user is registered it is hashed and sent to the database.

-The details are stored in an SQLite database

-Details of every music and how they are linked are saved in the database

-Details of every conductor is stored in the database

-Users are able to view the catalogue of music and composers

-Users are able to view the available music and reserve it

-Users can see their reserved/loaned music

-Different types of users.

-Visitors have to login

-Non-members can see all music and all composers but not reserve anything

-Members can do everything that a non-member can do but can also reserve only for themselves and see their suggestions

-Librarians can do everything a member can but can also reserve for others, allow users to actually borrow music, renew users music and can return music.

-Admins can see everything that a librarian can but can also edit the database completely and see all logs.

As part of the user types I will need permissions which will be outlined in design

### Specific Objectives

-The system should have a login page that allows you to login

-The system should have a main page with links to other pages such as:

-Login/Logout

-If it is login just give you a way to login as well as a way to get a new password

-If it is logout tells you that you are no longer logged

-Catalogue

-Has all music displaying the details including all the music instances. Also has a link to the composer

-Composers

-Displays all information about composer and links to music composed.

-Available

-My Reserved

-My Borrowed

### Optional Objectives (for if I have time)

1. Phone app to allow reservations to occur more easily as well as phone reminders about reservations and bookings
2. AI that is specific to each user and learns their normal music and gives them suggestions as they go along
3. Hard coded (or possibly AI based) chatbot to answer simple questions about how to use the reservation system
4. Calendaring
5. Suggestions algorithm that allows users to see what other users have given good reviews to based off of the fact that they gave a similar good review to the same piece of music
6. Filtering using JQuery. I chose this as it was the easiest to do in the time I had. I could have used Django-filter but it would have taken a long time and in the time constraints I couldn’t. JQuery allows easy lists similar to excel that will give me the filtering.
7. Adding in a separate user database to show that I could remove my user database and use BEAT’s instead.

## 1.6-Proposed Solution

I will be using a programming language called Python because it is a language I know fairly well, and it is a language with plenty of documentation. This will mean that I can easily learn the things I don’t know from reading about them and following tutorials. I tried to learn PHP from scratch in context to my project but couldn’t find anything.

I will be using a web framework called Django (<https://www.djangoproject.com/> or <https://github.com/django/django>). This library handles the backend of website building in Python. This includes talking to an SQLite3 database without the programmer having to know the precise SQL to do so. It also handles the running of the webpage and the admin functions. It also has extra parts do handle emails and creating of forms and the actual creation of the database. This means that you only have to know one programming language. I will however be using HTML so that I can use bootstrap (<https://getbootstrap.com/>) to make the website look nice. Django doing most things will allow me to focus more on complicated algorithms and building my AI for my suggestions page (and hopefully my chatbot). I tried using Flask but it has very limited resources and doesn’t provide you with much of a framework.

To implement my AI, I will be using a neural network library called TensorFlow and tflearn. These contain basic functions for creating layers for the network to learn from. This also allows it to learn off of itself to get better every time. Using this library will free up time working on my own functions to create layers as well as being able to not have to get it to teach it every time from new data. This will also allow it to function quicker and I can link it to the Google Cloud to do all the learning for me.

From my research, it is more suitable to run AI on a server where you get a GPU you can use rather than trying to host it on your machine which will never have a powerful enough GPU. Also, neural networks are the better way for me to go as they learn off of lists and are easy enough to program, they also take up far less space compared to deep learning. Deep learning would create more links but takes more time which I don’t have.

To run the server, I have chosen to run a virtual Ubuntu server on Digital Ocean. They offer a $5 tier and GitHub provides a free $50 as a student. For the web server I am just using Django. I have some familiarity with Apache but there is no need for it as Django can do everything. Eventually I would have to use Apache for the final running of the server but as it is still in development I will not be using it.

If I have time to create a phone app it will first be on android as I only have a windows laptop so can only run android studio. For this I would have to learn Kotlin. This is a java-based language made specifically for Android devices.

I would need to buy an iOS device and learn Swift to be able to create an app for apple watch and iPhone/iPad. This is a C based language aimed to work on just Apple Devices and only be written in MacOS/Linux.

For the database I am currently using SQLite3 as it is built into Django. It is an easy way to use python and databases without having to write hard SQL queries as instead you write 2 line queries which Django turns into sqlite3 queries for you. You also create readable models which have methods in them that are callable. I will for the moment create my own user model but in the future I can either import BEAT’s user table into my own or as they are using wordpress there is an easy API called django-wordpress that allows you to integrate WordPress’s user table into Django.

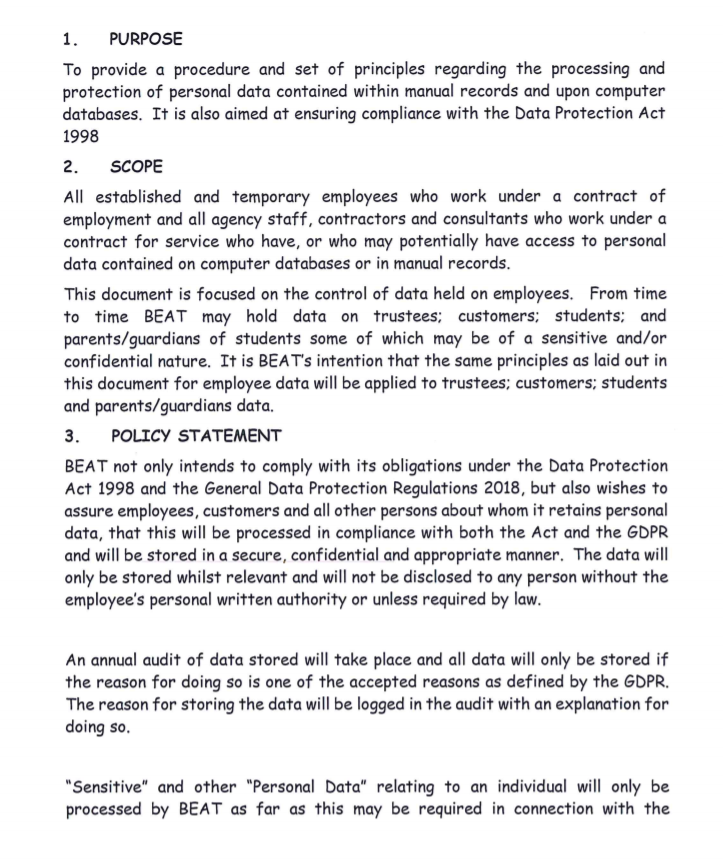
### Privacy Policy

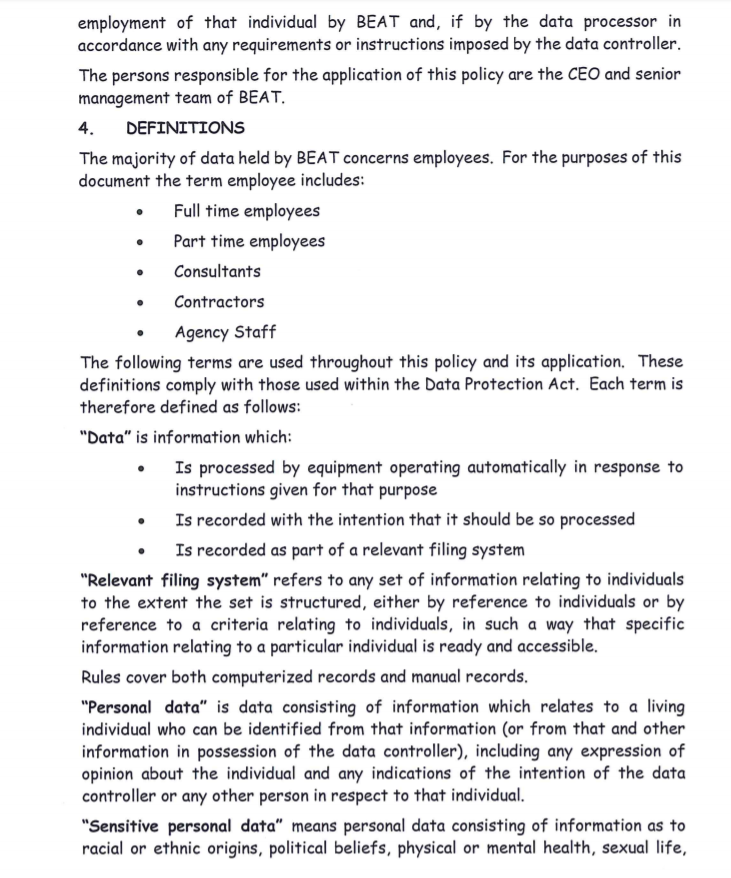
I am building a website and of 25th of May 2018 the GDPR laws came into force. This is the General Data Protection Regulation Act. It tells you what you can store and how long you should store it for. This may disappear in the future depending on Brexit but currently it is still necessary. My client currently has a privacy policy for their other websites so I will take this and update it.

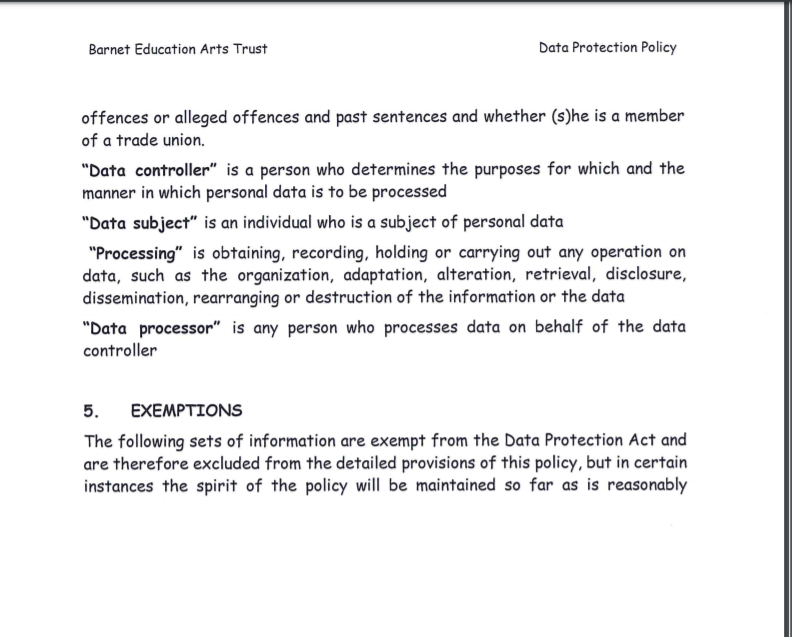
The things I need to include are:

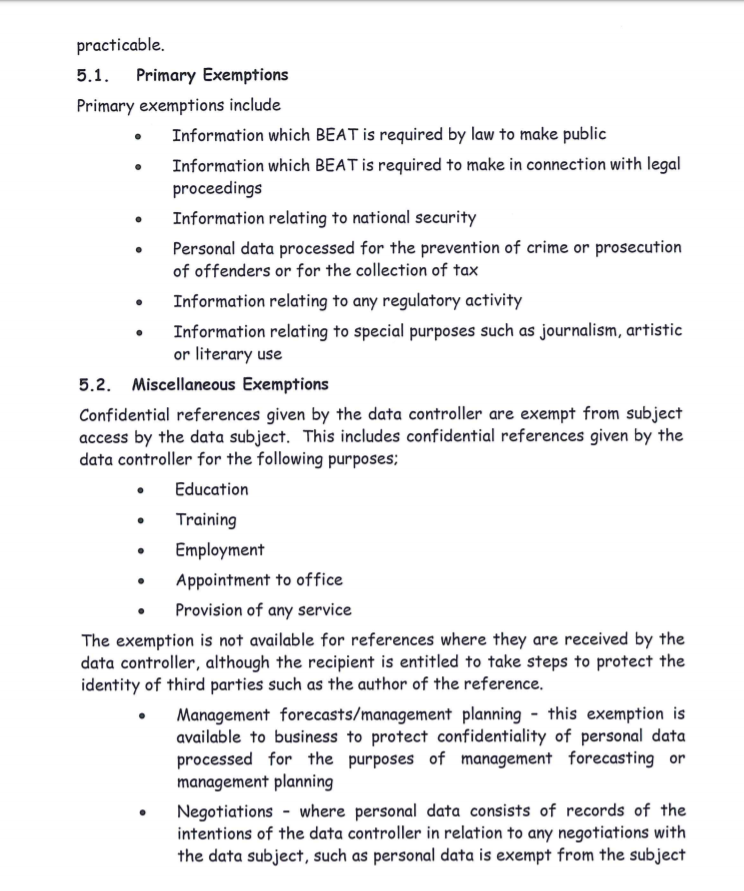
* What personal information I am collecting
* Why and how I am collecting it, and whether it’s mandatory
* What I am using the data for
* Who controls and processes the data
* What I retain, and where I store the data
* How I can keep the data secure
* Which third party companies can access the data
* Whether I am using cookies
* Whether I am using the data in automated decision-making
* Your users’ rights regarding their personal information

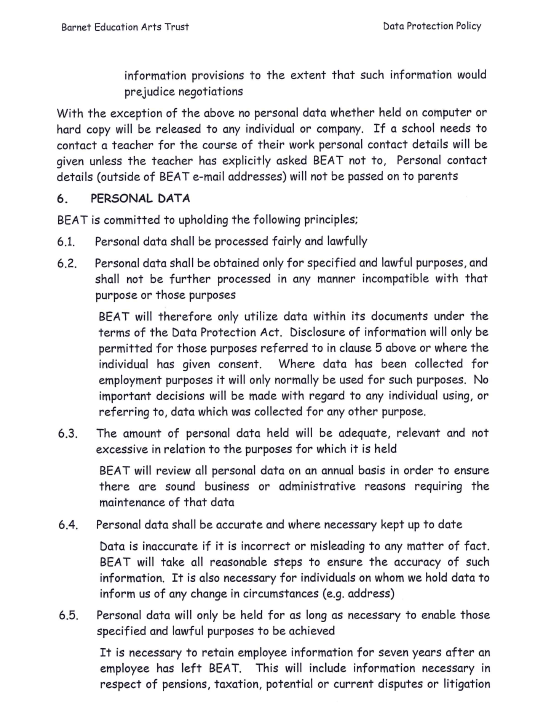
#### Current BEAT Privacy Policy

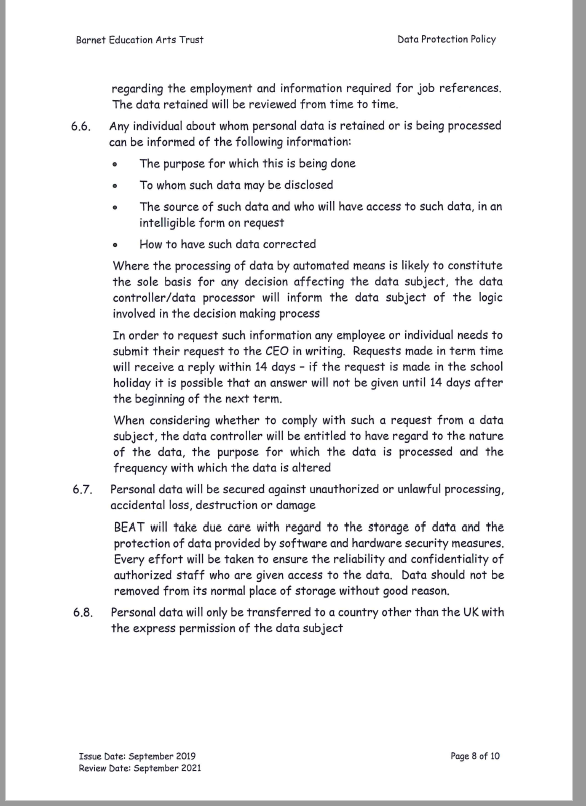


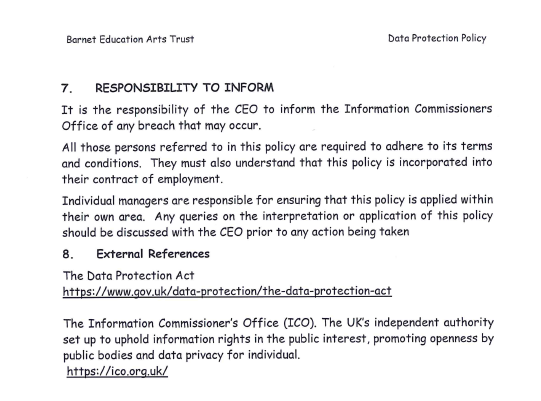












#### Bits to be Added to the Privacy Policy

**Using Your Data for Automated Decision-Making**

We collect information about what you take out to be able to make automated decisions of what else you might want to reserve and what to suggest you get out next. If you don’t want us using this information just let us know by email at admin@gshs.tech