AUDIO TOUR GUIDE APPLICATICATION

John Ivatt

Table of contents

Problem identification

There are currently no guides though Faversham that are constantly available apart from paper guides that you can pick up. But I believe they don't have the same emotion that a voice tour would have. So, I believe that building a voice tour web app for the town of Faversham would be unique and appealing to people who are new to Faversham as the town is very historical. This can increase tourism in the town and help local business by bringing in more people to visit the town and try the voice tour.

The features of this voice tour application would be required for this solution would be hardware capable of having an internet connection, a internet browser, gather location and play audio. The problem lends itself to computational solution because it's a lot cheaper to host a website audio tour then pay someone to do multiple tours a month. To describe it in its simplest form it will guide you on a map of historical locations in Faversham and will play audio talking about the history of the location.

Stakeholder

The clients and demographic for this web application would be mobile phone users who are also tourists. The amount of impact this web application can have on the towns tourism depends on how much it is advertised and promoted by the stake holders but it also depends on how many people are familiar with phones as well as how many people are able to use phones and have mobile data to interact with the web application.

The stakeholders for this software represent almost all tourist who want to know more about a town they are visiting but also a town, but there are also people who already live in the town and may want to go for a day out and explore the history of the town.

This being a very expandable web application means that it could be added to other places for audio tours about interesting locations. Another example of a good location would be Conyer as it has historical sites such as a brick works and a Victorian era waste site. Doing these for place would increase tourist and increase customers to restaurants such as "The Ship Inn" as they all have a vast history.

Why it is suited to a computational solution

I believe this issue lends itself to a computational method of finding and implements a solution for these reasons. The solution can be software based where all it needs is location data an internet connection and audio output. This will be able to run on a computer simply because that in this environment the location data needs to be updated every couple of seconds. I believe there is not a cheaper solution then using a computer as a guide would cost a lot more money.

The aspects of the solution that would be mainly controlled via gps data gathered and processed by the computer of where they are and where they are compared to the location to play audio. So depending on their location it will start playing audio for the correct location.

Computational methods that the solution lends itself to:

Problem recognition:

The overall issue is finding a way to process and then compare gps location to the location of the audio however the underlying problem is gathering the gps location from the gps hat. When this is overcome the solution is just comparing the data gathered to the data of the audio locations. If there are multiple locations I will have to have multiple locations for it to compare to and I believe I will need a list of list of gps locations for one area due to gps not always being accurate. So for this I may also add a code or qr code system where a person can input a code shown at a location to play audio.

Problem decomposition

This problem can be decomposed into a set of smaller steps. This is an initial idea of what these steps are:

1. Using the gps hat to gather location data.
2. Compare gps location to location of audio in database.
3. if location is correct play correct audio file.
4. check if they have entered another audio location and play that file instead

When this is done, the process will be completed, and if it is done correctly and the gps data is correct I will not need to add a backup system such as a code for the audio or a qr code. As every couple of seconds, it will update gps location to make sure the correct audio is playing.

Divide and conquer

These smaller steps maybe challenging on their own but are very doable with the correct research. Solving each step on their own and combining them into a modular program and finding out what works together to solve the problem.

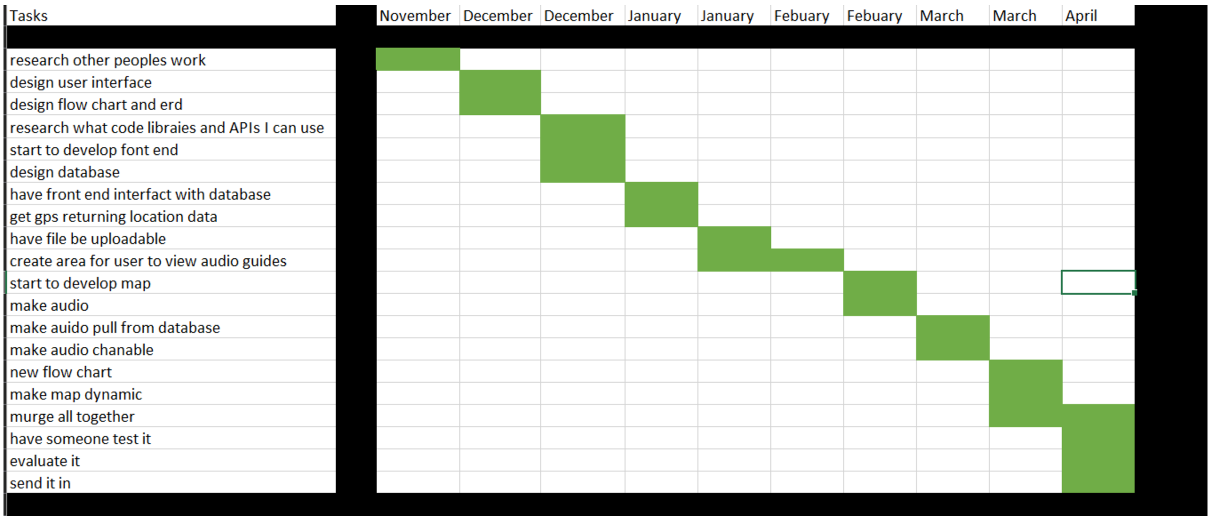
Abstraction

When gathering the GPS location data. The location data here we will need almost all of the data although we may be able to cut off a decimal here and there to reduce the amount of data sent to the program. This will be input to the software.

The output of the program will be audio and potentially the devices location in proximity to where the audio guide is on a map. This can lead to another way to learn and have guides around historical sites. I believe this will impact the way people visit historical sites. This can be implemented by companies and councils. There can be local companies who sponsor the. Production of these so that we can recommend them on the walk

MoSCoW Prioritisation			
Project Name:	Voice map using html JavaScript and python		
Project Manager:	John Ivatt		
Project Description:	A description of the project, or work package to be undertaken.		
	My project is to use a raspberry pi which will gather gps location to find its location and if it finds that the location is correct it will play an audio track stored in a database.		
Limitations:	Time:	Budget:	Resource:
	deadline end of March 2022 **4-6 hours a week**	**£120**	Raspberry **Pi** **Gps pi hat** **IDE**
M(ust have):	What must be delivered, i.e. it is essential for this phase?		
	Gather information from a GPS hat on a raspberry pi and play audio from a database when the correct data from the gps hat is gathered and have a GUI.		

S(hould have):	What should be delivered as a high priority but not essential?
	Shows the location of the pi and how close the destination is on a website that uses backend python to process.
C(ould have):	What could be delivered if there was available time / budget / resource?
	A backup for if the GPS isn't working such as having a QR code to scan using a camera on the pi. Have more locations that play different audio
W(ould have):	What would be have if time / budget / resource was unlimited?
	Hosted on a website and be a web app that phone users can use that gathers their location from phone.

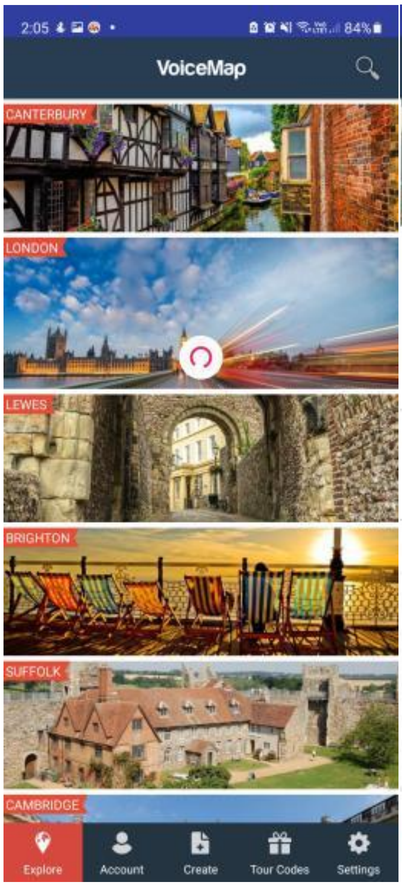


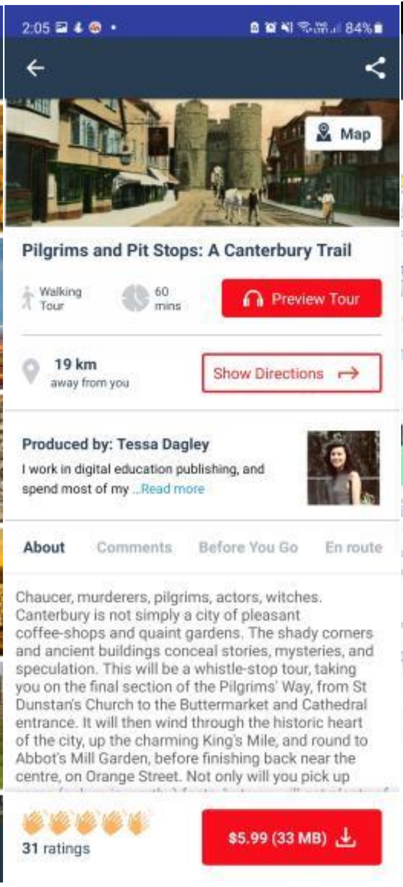
Tasks	November	December	December	January	January	Febuary	Febuary	March	March	April
research other peoples work	🟩									
design user interface		🟩								
design flow chart and erd		🟩								
research what code libraies and APIs I can use			🟩							
start to develop font end			🟩							
design database			🟩							
have front end interfact with database				🟩						
get gps returning location data				🟩						
have file be uploadable					🟩					
create area for user to view audio guides					🟩	🟩				
start to develop map							🟩			
make audio							🟩			
make auido pull from database								🟩		
make audio chanable								🟩		
new flow chart									🟩	
make map dynamic									🟩	
murge all together									🟩	🟩
have someone test it										🟩
evaluate it										🟩
send it in										🟩

Research

Existing similar solutions

Voice map





Pilgrims and Pit Stops: A Canterbury Trail

Walking Tour | 60 mins | Preview Tour

19 km away from you | Show Directions

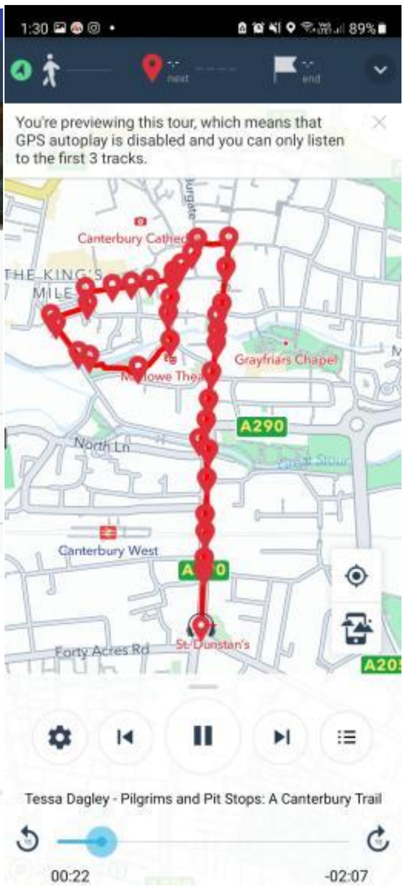
Produced by: Tessa Dagley

I work in digital education publishing, and spend most of my ...Read more

About | Comments | Before You Go | En route

Chaucer, murderers, pilgrims, actors, witches. Canterbury is not simply a city of pleasant coffee-shops and quaint gardens. The shady corners and ancient buildings conceal stories, mysteries, and speculation. This will be a whistle-stop tour, taking you on the final section of the Pilgrims' Way, from St Dunstan's Church to the Buttermarket and Cathedral entrance. It will then wind through the historic heart of the city, up the charming King's Mile, and round to Abbot's Mill Garden, before finishing back near the centre, on Orange Street. Not only will you pick up

31 ratings | $5.99 (33 MB)



You're previewing this tour, which means that GPS autoplay is disabled and you can only listen to the first 3 tracks.

Tessa Dagley - Pilgrims and Pit Stops: A Canterbury Trail

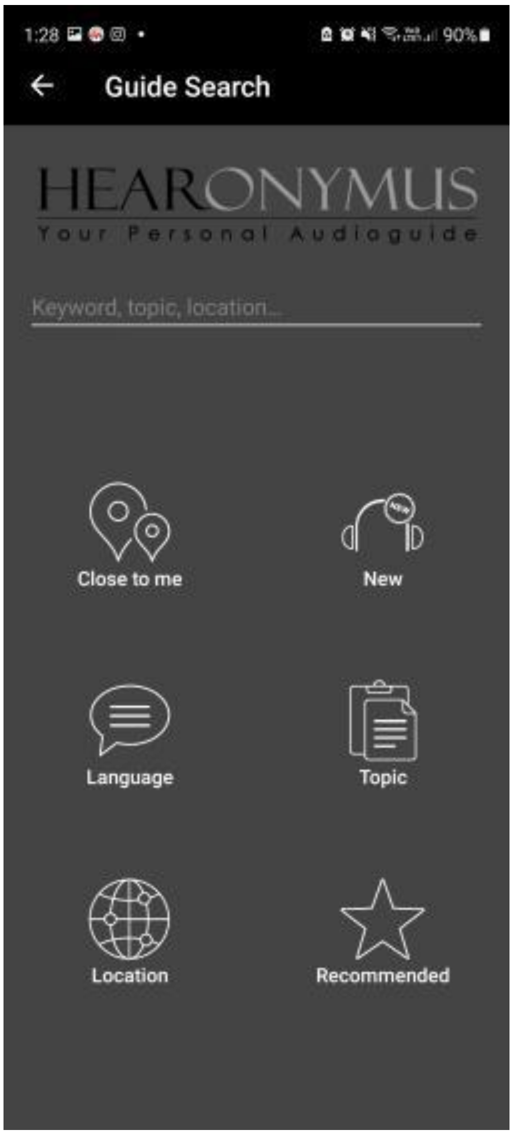
00:22 -02:07

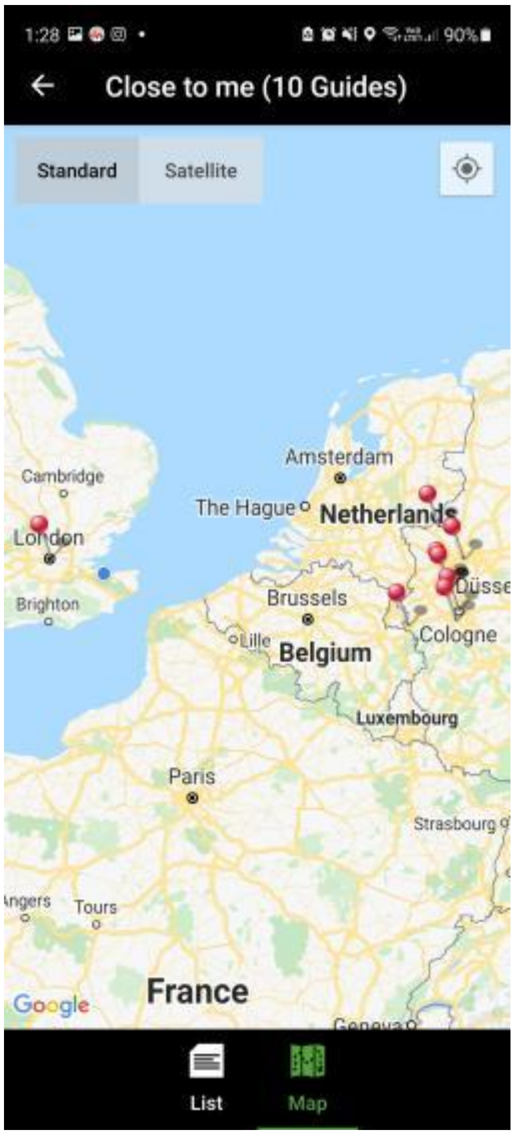
An similar existing solution is VoiceMap, VoiceMap is one of the most popular audio tour apps due to its modern style and crowed sourced audio guides it is free to use and look at but guides can cost money depends on who uploaded it as some can be free while others are paid for such as the one in the images. The design of the app is informative and yet not over crowded which I believe is perfect. The audio tours show a guided map of where you need to go as well as points of interest that will let you track of where you are which is great for keeping pace. The user interface of the audio is also unique but also intuitive to the point where you will not even have to think about what you're looking at such as the back or forward 10 seconds as well as the settings icon, the whole layout in almost perfect.

Parts that I can use

The concept of VoiceMap is very similar to my solution, however my solution will have more of an emphasis on the location of the person. The map window on the guide was very useful to display where you are meant to go. Although my guide would have better directions similar to google maps due to mine being much more location based than audio based. However I believe the audio controls that are used in VoiceMap are very good being simple and intuitive and does not get in the way to much. The audio control and map is something I would like to carry forward into my solution.

Hearonymous





Hearonymus is an audio tour guide app that focuses on indoor tours. The minimalist deign is great for getting to the point and I think showing a map of audio tours is a great way to show where they are and how close they are to you so you could plan out a day of an audio tour based tour of locations. I dislike the fact that you are not able to upload your own content and that there isn't a trail of the audio tours, so you will not know the quality till you buy it which could be disappointing.

Parts that I can apply to my solution is using a map to show where the audio tours are because I believe that showing them in a map makes it easier to plan which ones you may want to listen to in a certain order. Another part of Hearonyums I like is the minimalist design on the home screen which will make coding easier.

There were many other audio tour applications suggested by Alterativeto.net however it seems all of them have been abandoned or redesigned into being a leisure and travel blog.

Hardware

Touch Screen are a type of peripheral that allows input of touch and output of an image/video. Touch on the screen can be registered as mouse input meaning that it can be used instead of a mouse. Both Windows and Linux have touch support so it could be used with either operating systems. They are becoming very popular in laptops and all in one computers.

I can apply this to my solution to my solution as they provide a similar environment to a mobile screen this will also make it easier to move while on the go as I would not need to carry a mouse to the location.

GPS is a type of location data that can be gathered in many ways. A USB GPS dongle will give a computer GPS Location data on where the dongle is which can be used to track the computer world wide. Its performance depends on where it is as if it is wrapped in tinfoil in a building it will be hard to gather location data due to it not being able to gather data.

I will be able to use a GPS dongle to gather GPS location of the device. I can then display the location of the device and also use this location data to verify the location and then start playing the correct audio.

Similar projects that ask/give for advice with code

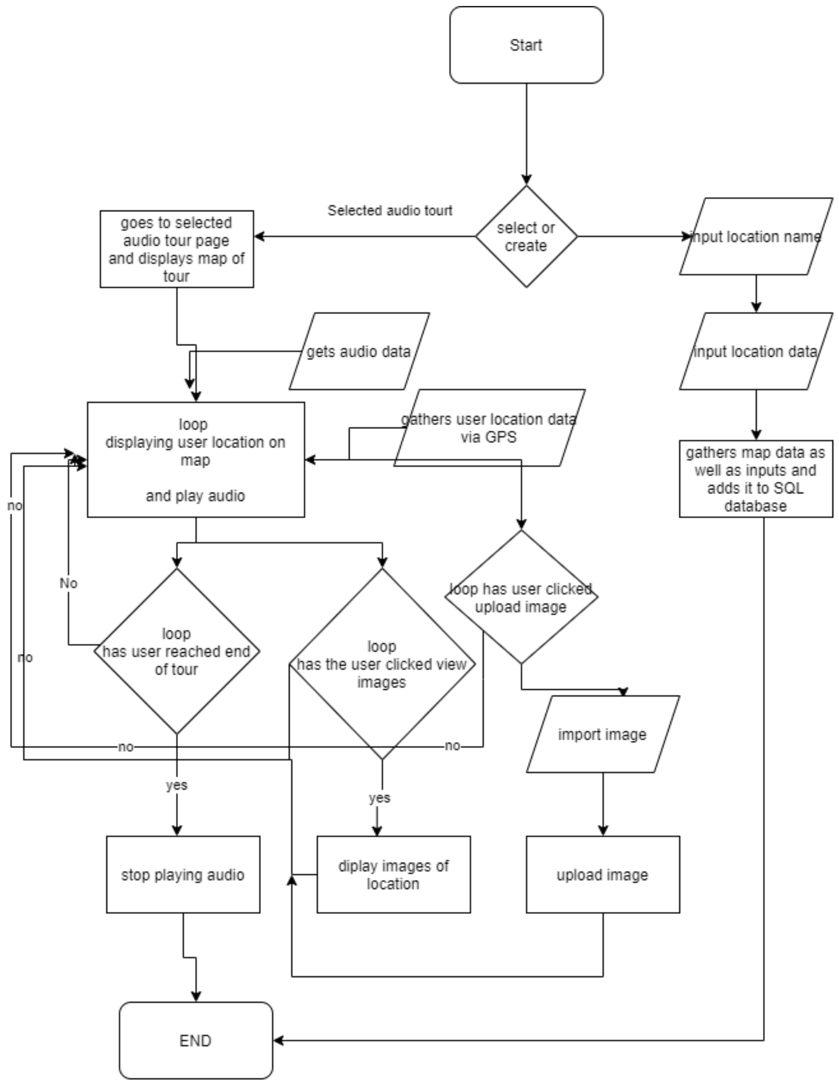
One similar project that asked for advice was how to gather location data on python(stack overflow). The most agreed upon advice on how to gather location data was geocoder which gathers IP data which gives a general idea of location via IP but I believe that wont be accurate enough another piece of advice given was to use IPinfo or IPstack which once again only gathers IP information which is not accurate enough for my project.

Towards data science recently published an article about “Simple GPS data visualization using Python and Open Street Maps”. In this article describes how to show the GPS data on a map. I believe this will be perfect for displaying the location of audio tours as well as a route to follow for the audio tour.

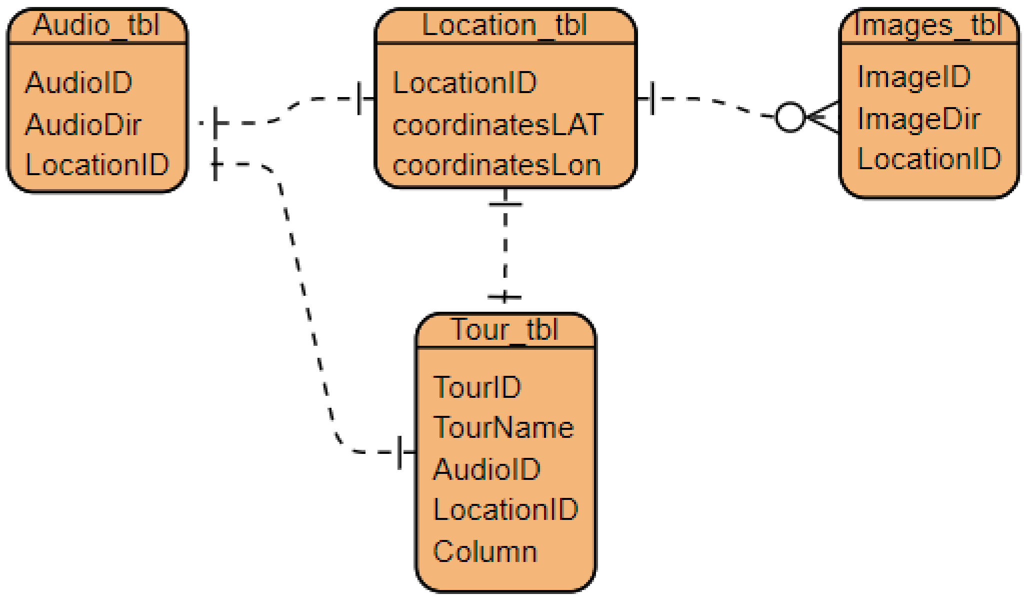
Conclusion of research

In conclusion to my research, I believe that there are many factors of my research that will prove vital to the success of my project. The similar applications available made me see the positives and negatives of what customers may want as well as what I want my application to be like. When looking at coding such as GPS data Visualization it made me think about how I can use it myself and how I could implement it in a way that could auto generate a route though the audio tour.

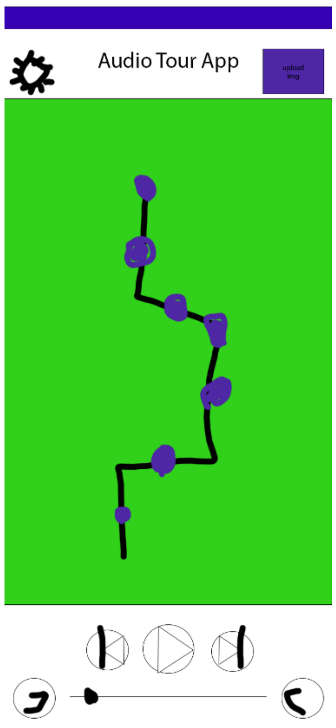
First design of Flowchart and ERD


Start
select or create
Selected audio tourt
goes to selected audio tour page and displays map of tour
input location name
input location data
gathers map data as well as inputs and adds it to SQL database
gets audio data
gathers user location data via GPS
loop displaying user location on map
and play audio
loop has user reached end of tour
loop has the user clicked view images
loop has user clicked upload image
import image
No
no
no
no
no
yes
yes
stop playing audio
diplay images of location
upload image
END

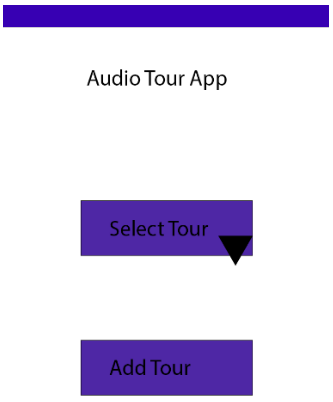
This is a flowchart to show how the data will flow within my program and what processes it will use as well as the data inputs. First it starts with a main page with an option to either create an audio tour or select a previously made tour. If the user selects a premade audio tour from the drop down list they will be presented with that audio tours map as well as audio controls. From there location data will be gathered and the users location will be displayed on the map. From now on it will repeatedly check for user inputs such as new location as well as if they have paused or skipped in the audio. If the user selects Create, they will be taken to a page where they can import location data as well as audio and then the program.



In my ERD I show how the data will be moved within my Relational Database. My Images_tbl has the ID of each image as well as the directory that it is stored in. It is then linked to the Location_tbl via a many to one relation as there can be many images for one location. The location table Includes both the Latitude and Longitude Coordinates. As well as an ID that correlates to them both. The Audio_tbl pulls LocationID from the Location_tbl. The Audio_tbl has the directory as well as an ID for the Location. Finally, the Tour_tbl pulls data from every table together as a main table to pull from when accessing the database.



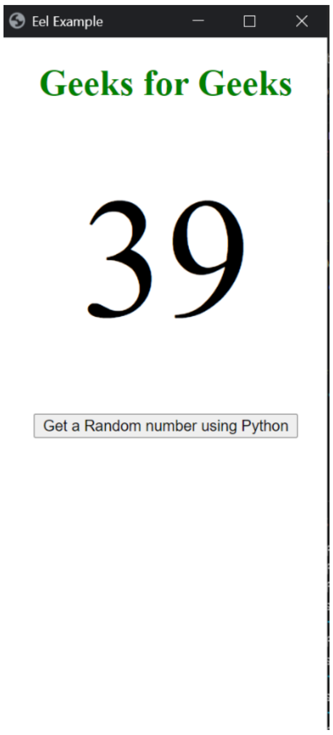
This is my design for the user interface for the user will be using an audio tour it will show them locations of interests as well as images if the location is clicked on. The settings tab will have things such as reset location as well as reloading the page. There is also an option to upload images for the locations of interest on this guide.



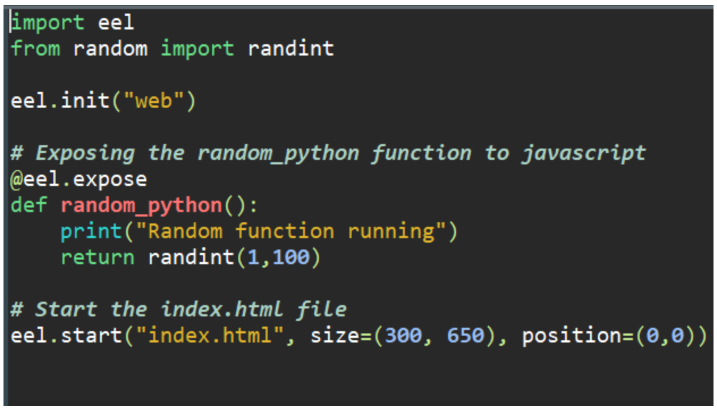
Here will be the main page/index page as it will let the user select what they will want to do may it be selecting a premade tour or creating their own. If they click on select tour it will show a drop down menu of names of tours as well as locations.

This is the create page where you can upload the location as well as the file. After creating this I have realised that I am missing a name input box which will be added with the next iteration. Once these details have been added they will be processed into an SQL Database.

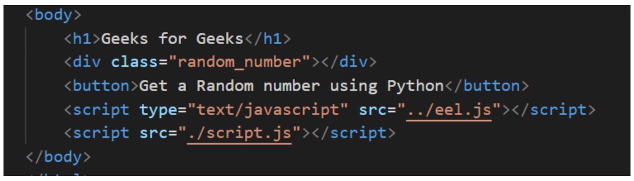
Developing the solution:



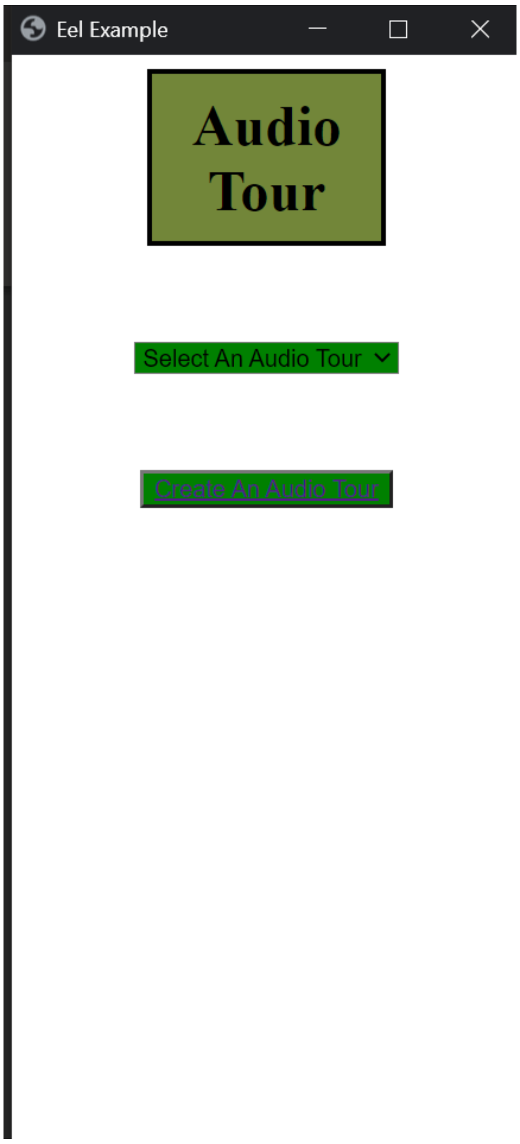
My first iteration of python code I got the frame for eels and file directory from an article on GeeksForGeeks.org published in 2021 and then to get the correct size of the page to make it look like a mobile phone I used a line of code from a git hub response by a user named Zeerti.



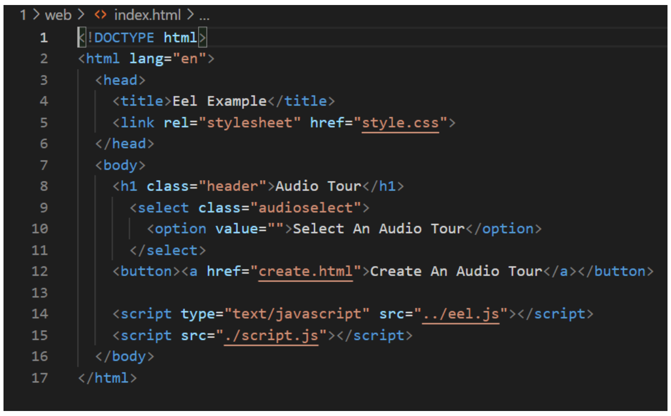
```python
import eel
from random import randint

eel.init("web")

# Exposing the random_python function to javascript
@eel.expose
def random_python():
    print("Random function running")
    return randint(1,100)

# Start the index.html file
eel.start("index.html", size=(300, 650), position=(0,0))
```



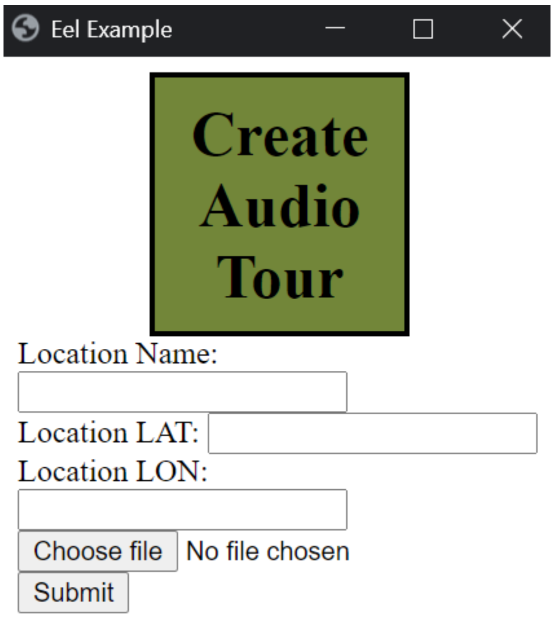
```
<body>
    <h1>Geeks for Geeks</h1>
    <div class="random_number"></div>
    <button>Get a Random number using Python</button>
    <script type="text/javascript" src="../eel.js"></script>
    <script src="./script.js"></script>
</body>
```

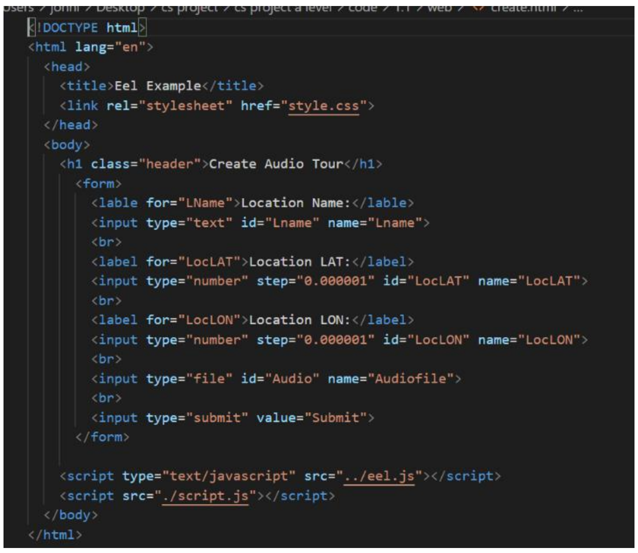


From here I changed the UI and created a user interface using HTML. With a drop down menu which will be built upon on a later iteration. The Create An Audio Tour button then takes the user towards the Creation page for Audio tours.



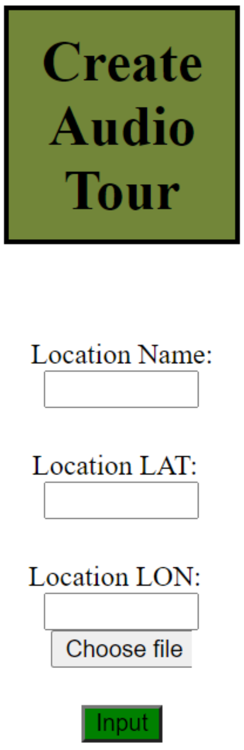
```
1 > web > <> index.html > ...
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Eel Example</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1 class="header">Audio Tour</h1>
      <select class="audioselect">
        <option value="">Select An Audio Tour</option>
      </select>
    <button><a href="create.html">Create An Audio Tour</a></button>

    <script type="text/javascript" src="../eel.js"></script>
    <script src="./script.js"></script>
  </body>
</html>
```

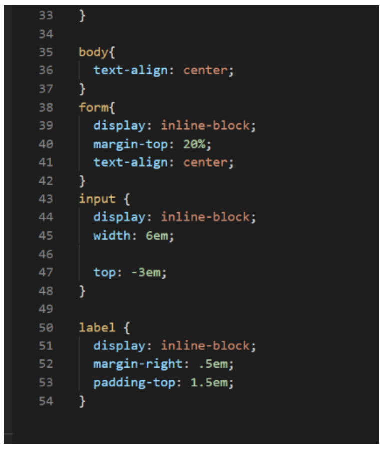




```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Eel Example</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1 class="header">Create Audio Tour</h1>
      <form>
        <lable for="LName">Location Name:</lable>
        <input type="text" id="Lname" name="Lname">
        <br>
        <label for="LocLAT">Location LAT:</label>
        <input type="number" step="0.000001" id="LocLAT" name="LocLAT">
        <br>
        <label for="LocLON">Location LON:</label>
        <input type="number" step="0.000001" id="LocLON" name="LocLON">
        <br>
        <input type="file" id="Audio" name="Audiofile">
        <br>
        <input type="submit" value="Submit">
      </form>

    <script type="text/javascript" src="../eel.js"></script>
    <script src="./script.js"></script>
  </body>
</html>
```

On the creation page I created multiple input boxes for the location name as well as the latitude and longitude of my location using HTML tags. I also created a file submit button for the Audio.





```
}

body{
  text-align: center;
}
form{
  display: inline-block;
  margin-top: 20%;
  text-align: center;
}
input {
  display: inline-block;
  width: 6em;

  top: -3em;
}

label {
  display: inline-block;
  margin-right: .5em;
  padding-top: 1.5em;
}
```

Then I centred the input boxes and had the input names above them using CSS Code.

I have created a relational database that I will be passing data to via SQL Statements I have tried to follow normalisation rules. Here is the main table that will be used for the tours.

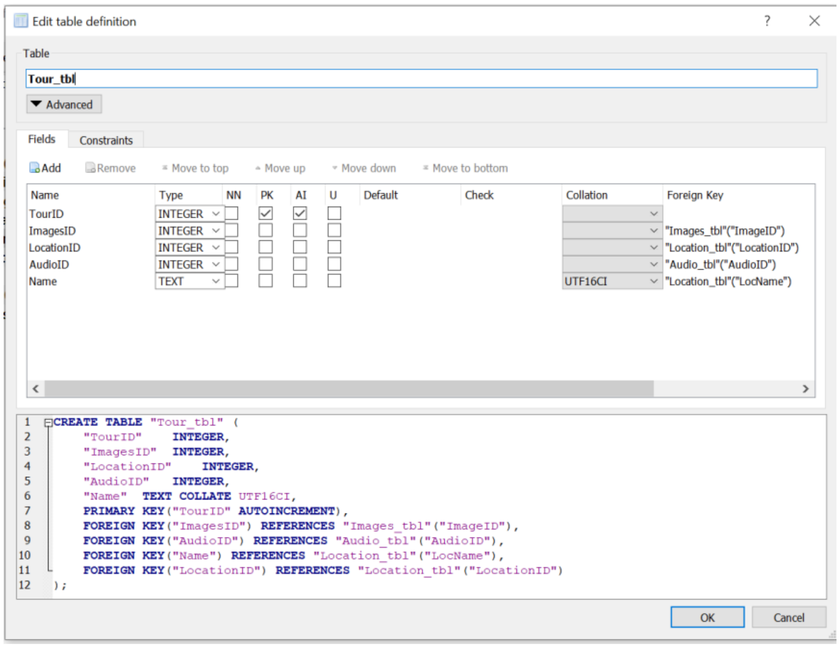
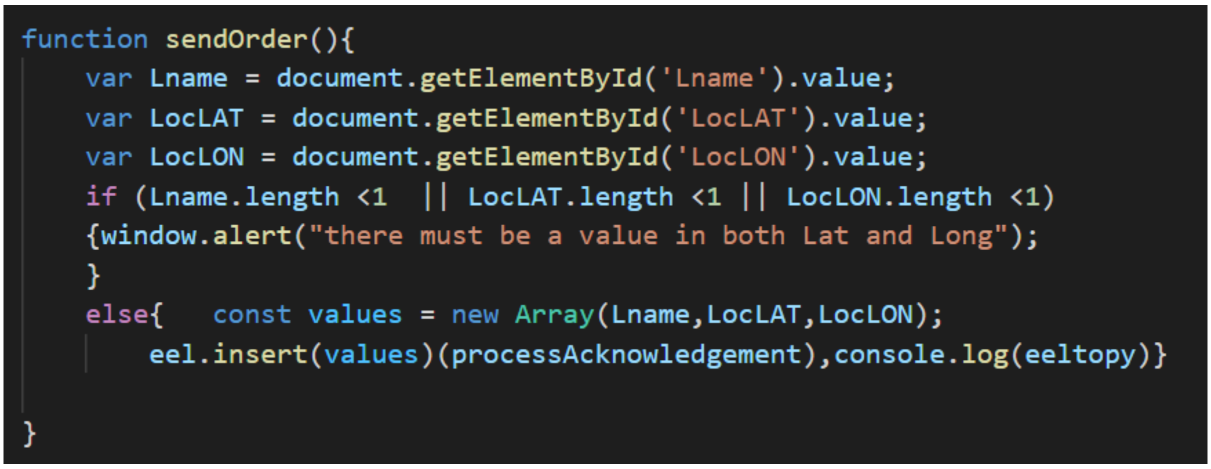


Table: Tour_tbl

Name	Type	NN	PK	AI	U	Default	Check	Collation	Foreign Key
TourID	INTEGER	☐	✓	✓	☐				
ImagesID	INTEGER	☐	☐	☐	☐				"Images_tbl"("ImageID")
LocationID	INTEGER	☐	☐	☐	☐				"Location_tbl"("LocationID")
AudioID	INTEGER	☐	☐	☐	☐				"Audio_tbl"("AudioID")
Name	TEXT	☐	☐	☐	☐			UTF16CI	"Location_tbl"("LocName")

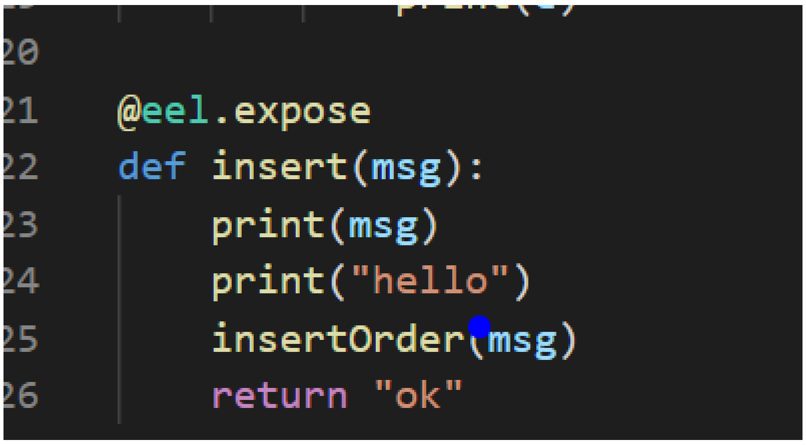
```
CREATE TABLE "Tour_tbl" (
	"TourID"	INTEGER,
	"ImagesID"	INTEGER,
	"LocationID"	INTEGER,
	"AudioID"	INTEGER,
	"Name"	TEXT COLLATE UTF16CI,
	PRIMARY KEY("TourID" AUTOINCREMENT),
	FOREIGN KEY("ImagesID") REFERENCES "Images_tbl"("ImageID"),
	FOREIGN KEY("AudioID") REFERENCES "Audio_tbl"("AudioID"),
	FOREIGN KEY("Name") REFERENCES "Location_tbl"("LocName"),
	FOREIGN KEY("LocationID") REFERENCES "Location_tbl"("LocationID")
);
```

From there I also created a JavaScript function which pulls data from the HTML input form and then converts it to an array and also has data validation that made it so the user had to fill in the input boxes.



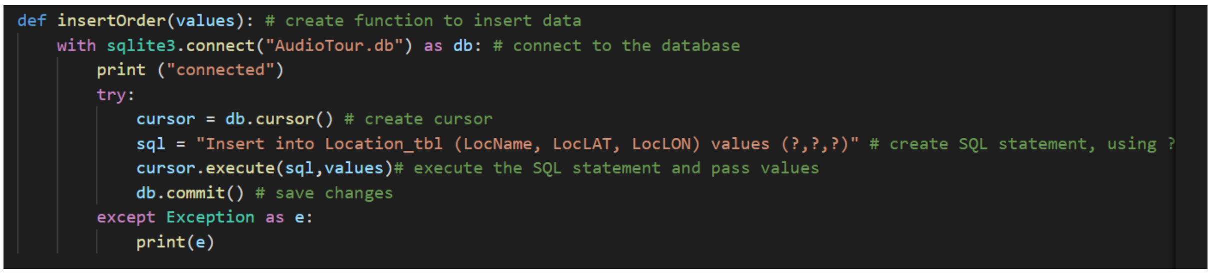
```
function sendOrder(){
    var Lname = document.getElementById('Lname').value;
    var LocLAT = document.getElementById('LocLAT').value;
    var LocLON = document.getElementById('LocLON').value;
    if (Lname.length <1  || LocLAT.length <1 || LocLON.length <1)
    {window.alert("there must be a value in both Lat and Long");
    }
    else{   const values = new Array(Lname,LocLAT,LocLON);
        eel.insert(values)(processAcknowledgement),console.log(eeltopy)}

}
```

From here I calls a python function called insert via eels(the app framework I'm using) which then prints the array in python to show me that eels has received the array. At the end of the function it calls upon called insertOrder.



```

@eel.expose
def insert(msg):
    print(msg)
    print("hello")
    insertOrder(msg)
    return "ok"
```

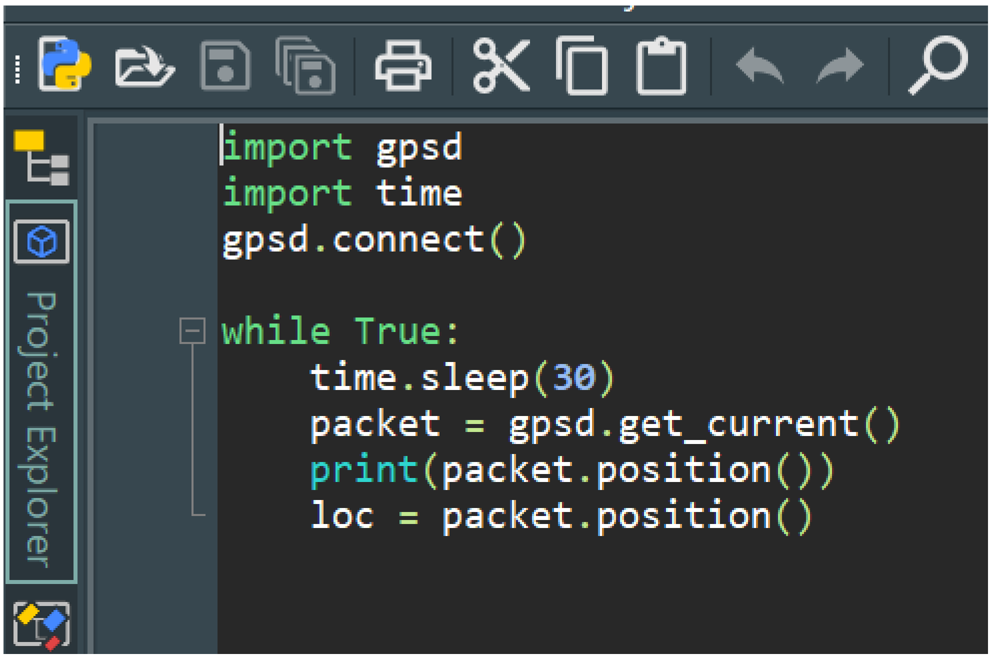
The insertOrder function uses SQLight3 in order to be able to do SQL statements and queries. The function was from Pythonschool.net which was then modified to be able to take 3 variables instead of 2 which then appends to my Location table.



```
def insertOrder(values): # create function to insert data
    with sqlite3.connect("AudioTour.db") as db: # connect to the database
        print ("connected")
        try:
            cursor = db.cursor() # create cursor
            sql = "Insert into Location_tbl (LocName, LocLAT, LocLON) values (?,?,?)" # create SQL statement, using ?
            cursor.execute(sql,values)# execute the SQL statement and pass values
            db.commit() # save changes
        except Exception as e:
            print(e)
```

Getting location data from GPS

For my location tracking and verification I used a usb GPS device which will give me location data of the device for me I'm using a raspberry pi as my device due to its portability and being able to modify it to my needs. To get the GPS working I first downloaded dfrobot Linux drivers by cloning their GitHub repository and following their instructions. From there I would sudo ls -1 /dev which would output a 2d array of devices and the ports from there I was able to find out which port the GPS was connected to. After that I would install gpsd using sudo apt-get install gpsd dpsd-clients which installs more gps drivers that will later on be used in python3. I then changed a file in gpsd so it directs to the correct port and runs at start up. After that I rebooted my pi so GPSD would run in the background. I then tried to run xgps as that shows me a grid of what satellites are locked on to my GPS. However it turned out I was missing some drivers/dependencies for XGPS so I then installed python-gi-cairo. After that xgps then worked. I then finished install the final part of gpsd with pip3 install gpsd-py3. After that I imported gpsd in python and then connected to the gps with gpsd.connect() and to the location I would do packet=gpsd.get_current() which gets me the current position and then to print the location I would use print(packet.position()). This gpsd code was from the GPSD client github.



```
import gpsd
import time
gpsd.connect()

while True:
    time.sleep(30)
    packet = gpsd.get_current()
    print(packet.position())
    loc = packet.position()
```

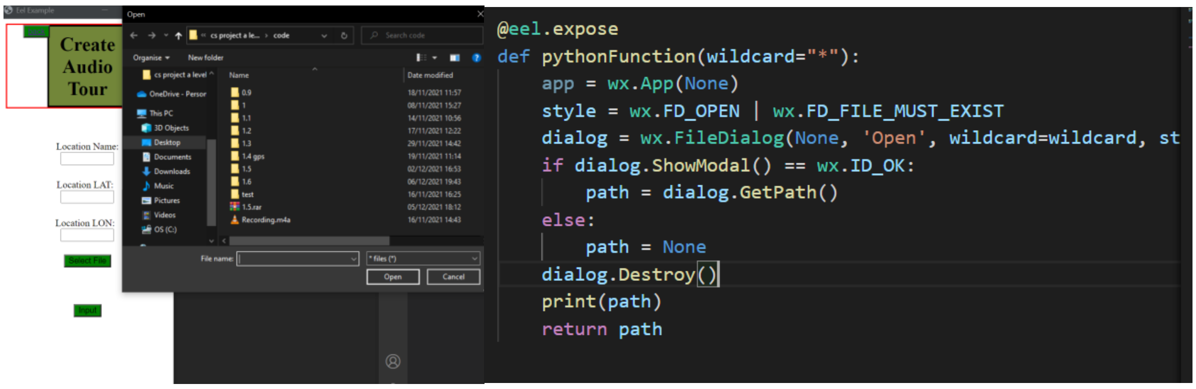
Second iteration

Code changes

In my second iteration I have added a working audio file upload feature which lets people upload a “file” when it really uploads the directory and sends that over to python to interpret and from there I will use an SQL statement to add it to a database which currently not implemented but will be next.

When the user selects a file to choose it will run a JavaScript function which in turn calls upon an eels function in python that creates a new window for the file directory to be uploaded using a python library called wx. After that it is then processed in python and forwarded to java script.

This code was from a stack over flow response by the user Nootaku.



```
@eel.expose
def pythonFunction(wildcard="*"):
    app = wx.App(None)
    style = wx.FD_OPEN | wx.FD_FILE_MUST_EXIST
    dialog = wx.FileDialog(None, 'Open', wildcard=wildcard, st
    if dialog.ShowModal() == wx.ID_OK:
        path = dialog.GetPath()
    else:
        path = None
    dialog.Destroy()
    print(path)
    return path
```

Database ERD

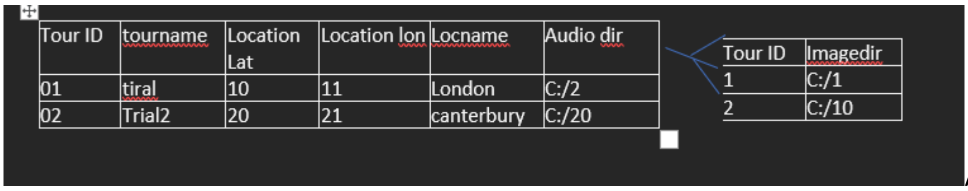
I have also remade my ERD and my database and am now following to normalisation 1, 2 and 3. I have decided to follow the rules of normalisation 3 to stop I started my making an example of my data that will be used shown below.

tourname	Location Lat/lon	Locname	Audio dir	Imagedir
trial	10 ,11	London	C:/2	C:/1
Trial2	20 ,21	canterbury	C:/20	C:/10
trial	10 ,11	London	C:/2	C;/20

From there I followed n1 which are each table cell should contain a single value and each record needs to be unique.

tourname	Location Lat	Location lon	Locname	Audio dir	Imagedir
trial	10	11	London	C:/2	C:/1
Trial2	20	21	canterbury	C:/20	C:/10
trial	10	11	London	C:/2	C;/20

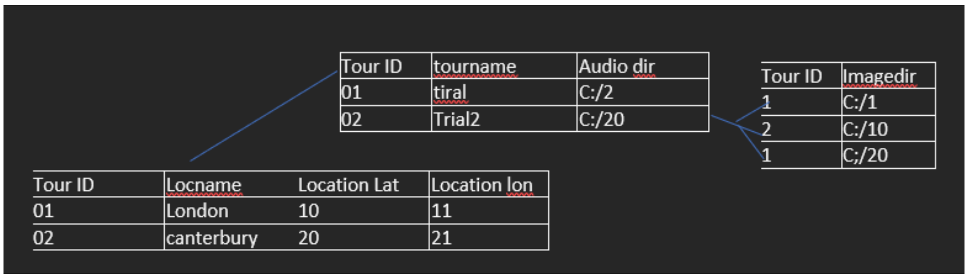
After this I followed n2 rules which is be in N1 and that there is a single column primary key that does not functionally depend on any subset of candidate key relation.



Tour ID	tourname	Location Lat	Location lon	Locname	Audio dir
01	tiral	10	11	London	C:/2
02	Trial2	20	21	canterbury	C:/20

Tour ID	Imagedir
1	C:/1
2	C:/10

And finally N3 requires N2 to be met and that the database has no transitive functional dependencies which means that tables often need to be split again.



Tour ID	tourname	Audio dir
01	tiral	C:/2
02	Trial2	C:/20

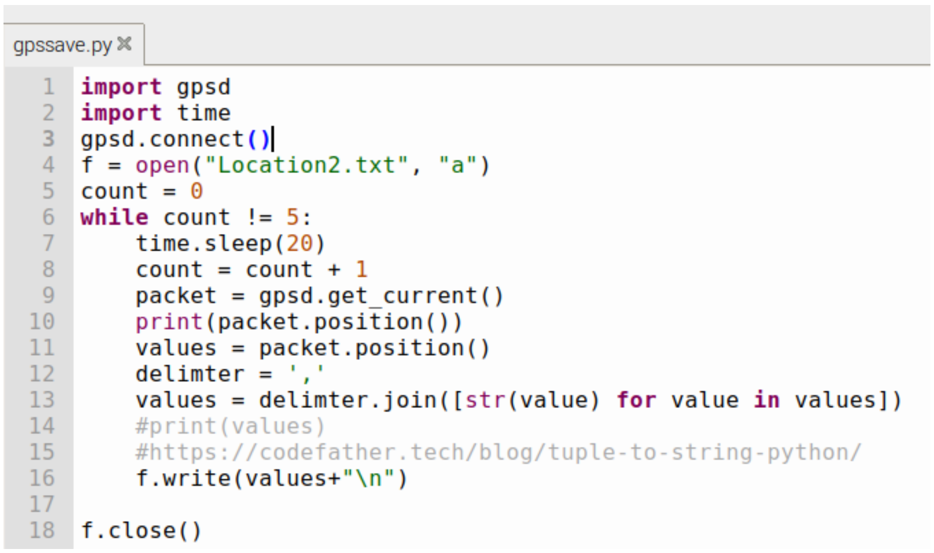
Tour ID	Imagedir
1	C:/1
2	C:/10
1	C;/20

Tour ID	Locname	Location Lat	Location lon
01	London	10	11
02	canterbury	20	21

To get to this I followed an article on normalisation on a website named Guru99 and the article was by a person named Richard peterson and a youtbe video by a channel named craig'n'dave about reaching N3 normalisation.

GPS location gathering and testing.

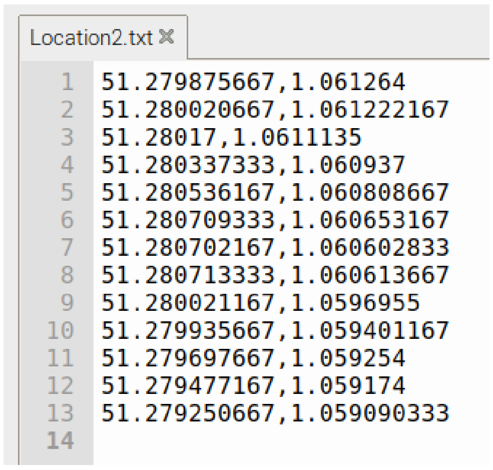
As seen in my previouse iteration I am using GPS data as an example for general location data that my application will interptit to display user location and the location of audio tours. To start with I tried setting up my gps on a Linix based virual matchine but I had issues passing through the gps to the virtual machine so instead I tried Linux laptop which is a T420 Think Pad was running Lubuntu (a lightwight version of ubuntu) when trying XGPS on my laptop I kept running into issues mostly with dependcies.



gpssave.py

```python
import gpsd
import time
gpsd.connect()
f = open("Location2.txt", "a")
count = 0
while count != 5:
    time.sleep(20)
    count = count + 1
    packet = gpsd.get_current()
    print(packet.position())
    values = packet.position()
    delimter = ','
    values = delimter.join([str(value) for value in values])
    #print(values)
    #https://codefather.tech/blog/tuple-to-string-python/
    f.write(values+"\n")

f.close()
```

This code gathers location data via gpsd which has a sleep to give the chnace for the GPS to lock onto the satalities, it then returns a tuple of floats which then gets turned into a string via code from. This string is then appened to a text file callled Location2.txt, below is an example of the text file after the code has ran.



Location2.txt

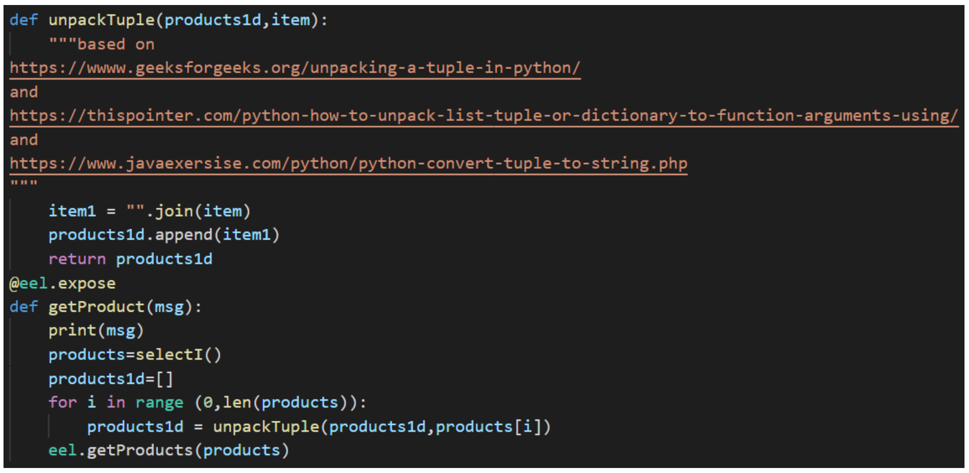
```
51.279875667,1.061264
51.280020667,1.061222167
51.28017,1.0611135
51.280337333,1.060937
51.280536167,1.060808667
51.280709333,1.060653167
51.280702167,1.060602833
51.280713333,1.060613667
51.280021167,1.0596955
51.279935667,1.059401167
51.279697667,1.059254
51.279477167,1.059174
51.279250667,1.059090333

```

When testing my application I realised there was no back button to swap between pages. To fix this on the create audio tour I created a button which redirects to the index page.

Third iteration

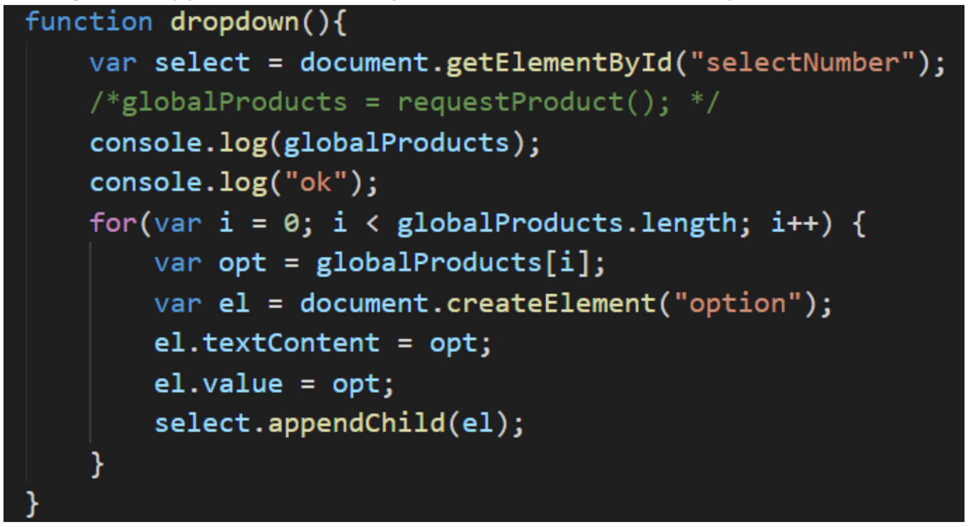
Populating a dropdown from a database

Ive decided to create a dropdown menu for usesrs to select an auido guide from because it means a user will not have to update the index file every time an audio tour is created. This will automate it meaning that a admin will not have to change it meaning It will save time and money. To start I needed to look at what datastructures I will use and what ones I should focus on. To start, I first looked at what data strucuture I have being retrived from my database using an SQL statemtent which was an array of tuples. To start maiplulating the data, I had to unpack the tuple due to tuples being immutable and static. To unpack them I followed many guides however the most useful one was made by javaexercvise which suggests using .join as in takes itertable object and turns them into a string and appends them to an array which is pulled via into javascript via eels.

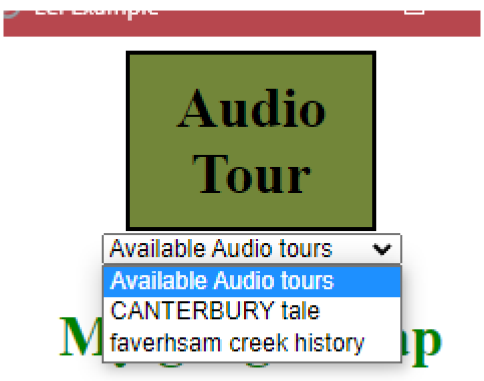


```
def unpackTuple(products1d,item):
    """based on
https://wwww.geeksforgeeks.org/unpacking-a-tuple-in-python/
and
https://thispointer.com/python-how-to-unpack-list-tuple-or-dictionary-to-function-arguments-using/
and
https://www.javaexersise.com/python/python-convert-tuple-to-string.php
"""
    item1 = "".join(item)
    products1d.append(item1)
    return products1d
@eel.expose
def getProduct(msg):
    print(msg)
    products=selectI()
    products1d=[]
    for i in range (0,len(products)):
        products1d = unpackTuple(products1d,products[i])
    eel.getProducts(products)
```

It gets made into a drop down menu though javascript code based on code from a stack overflow thread. The comment was made by a user named Tomerikoo who said that the array must be looped through and appended to the object which would be the drop down.

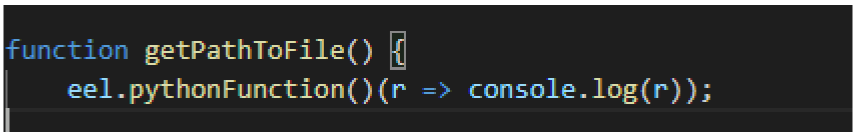


```
function dropdown(){
    var select = document.getElementById("selectNumber");
    /*globalProducts = requestProduct(); */
    console.log(globalProducts);
    console.log("ok");
    for(var i = 0; i < globalProducts.length; i++) {
        var opt = globalProducts[i];
        var el = document.createElement("option");
        el.textContent = opt;
        el.value = opt;
        select.appendChild(el);
    }
}
```

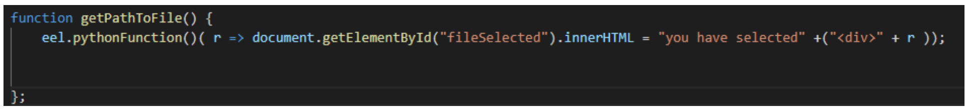


Inputting data to database.

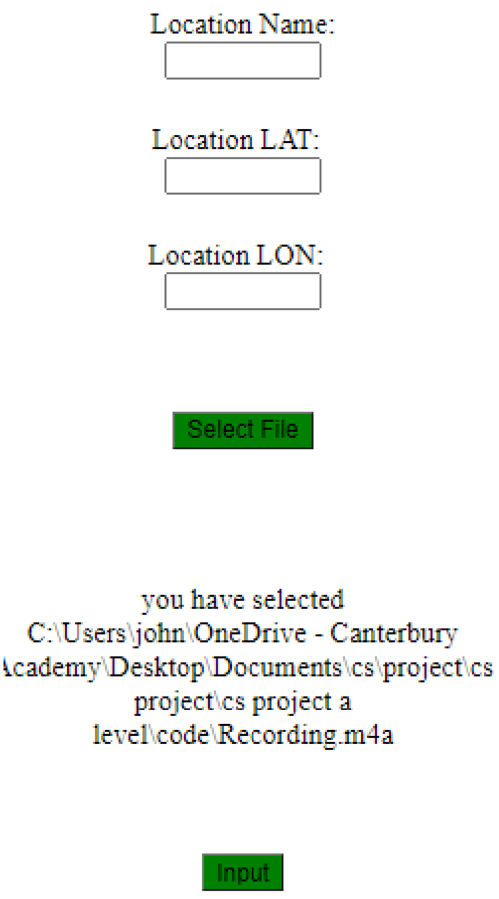
At this point I decided to test my code which lead to me to realise that users cannot see the file they have selected, to fix this I changed consle log which would post the string to the terminal to instaed get element by id as it would let me print it directly to the webpage.



```
function getPathToFile() {
    eel.pythonFunction()(r => console.log(r));
```

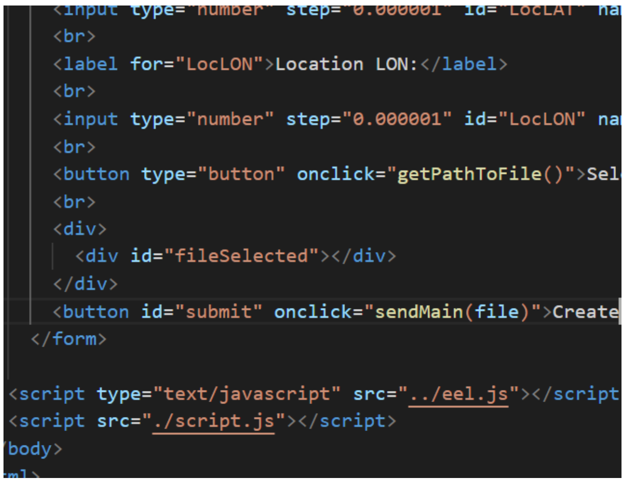


```
function getPathToFile() {
    eel.pythonFunction()( r => document.getElementById("fileSelected").innerHTML = "you have selected" +("<div>" + r ));


};
```



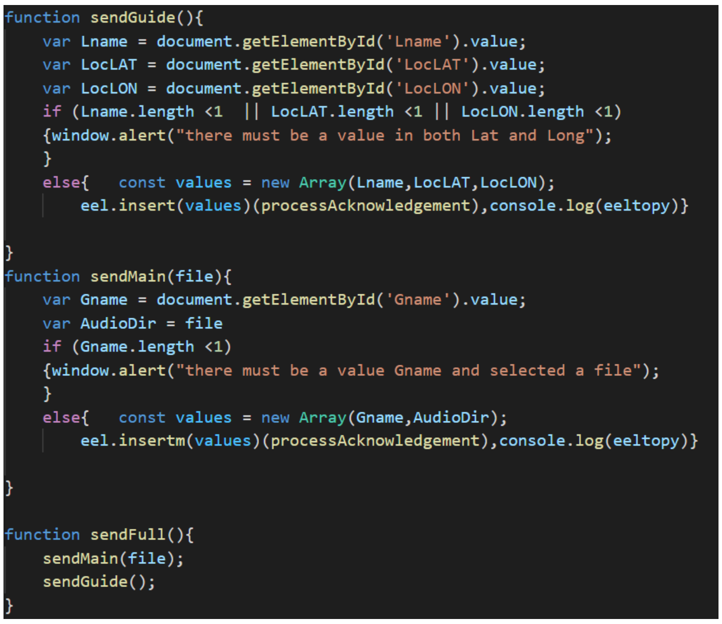
you have selected
C:\Users\john\OneDrive - Canterbury
Academy\Desktop\Documents\cs\project\cs
project\cs project a
level\code\Recording.m4a

From here I made it so when you click input which has changed to be “create” it sends a string for the file location to eels which gets input to a database using SQLight3.



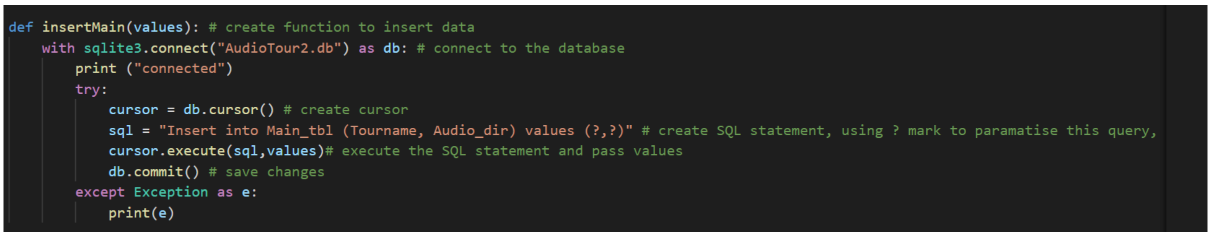
```
    <input type="number" step="0.000001" id="LocLAT" na
    <br>
    <label for="LocLON">Location LON:</label>
    <br>
    <input type="number" step="0.000001" id="LocLON" na
    <br>
    <button type="button" onclick="getPathToFile()">Sel
    <br>
    <div>
      <div id="fileSelected"></div>
    </div>
    <button id="submit" onclick="sendMain(file)">Create
  </form>

<script type="text/javascript" src="../eel.js"></script
<script src="./script.js"></script>
/body>
ml>
```

When the user clicks the button with the ID of submit it runs 2 functions.



```
function sendGuide(){
    var Lname = document.getElementById('Lname').value;
    var LocLAT = document.getElementById('LocLAT').value;
    var LocLON = document.getElementById('LocLON').value;
    if (Lname.length <1  || LocLAT.length <1 || LocLON.length <1)
    {window.alert("there must be a value in both Lat and Long");
    }
    else{   const values = new Array(Lname,LocLAT,LocLON);
        eel.insert(values)(processAcknowledgement),console.log(eeltopy)}

}
function sendMain(file){
    var Gname = document.getElementById('Gname').value;
    var AudioDir = file
    if (Gname.length <1)
    {window.alert("there must be a value Gname and selected a file");
    }
    else{   const values = new Array(Gname,AudioDir);
        eel.insertm(values)(processAcknowledgement),console.log(eeltopy)}

}

function sendFull(){
    sendMain(file);
    sendGuide();
}
```

As shown above the function calls 2 other functions which both send an array to eels and both use validation that has a window alert if the data is not correct.

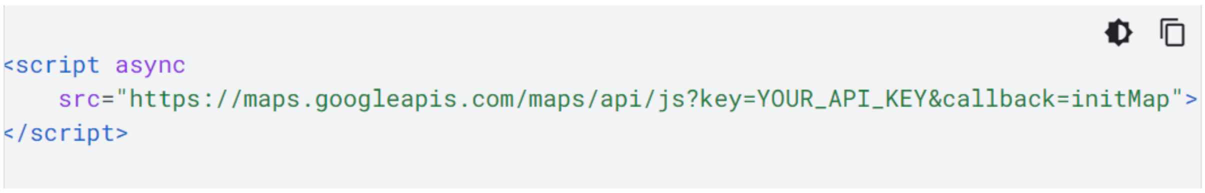


```
def insertMain(values): # create function to insert data
    with sqlite3.connect("AudioTour2.db") as db: # connect to the database
        print ("connected")
        try:
            cursor = db.cursor() # create cursor
            sql = "Insert into Main_tbl (Tourname, Audio_dir) values (?,?)" # create SQL statement, using ? mark to paramatise this query,
            cursor.execute(sql,values)# execute the SQL statement and pass values
            db.commit() # save changes
        except Exception as e:
            print(e)
```

Forth iteration

Showing user location on map

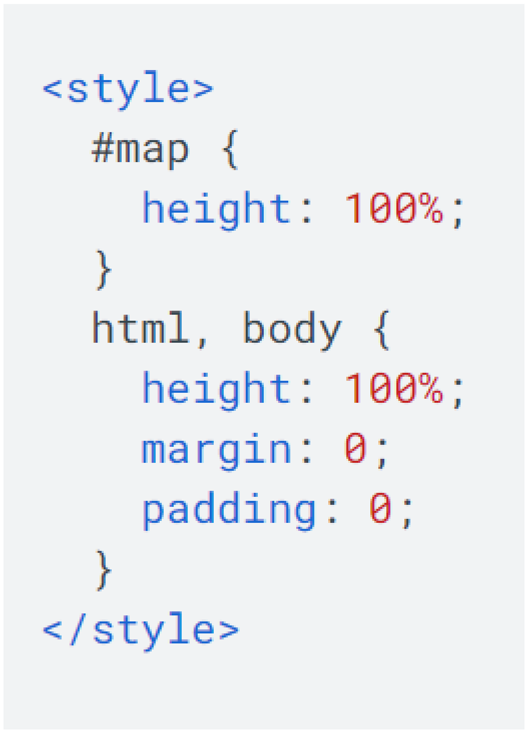
I started by thinking about what map software there is, open streetmap, google maps, maps.me, bing maps. I have decided to go with google maps as it has the most support and documentation esplally for the API. The google maps API uses Coridiantes to show the location of an area and my GPS sends out Courdinates which is perfect to display user locations. To start I got a Google maps api key which you can get for free if you have under a certain amount of requests a day. You can get the key from Google Cloud Platform which requires a Gmail account. Bur from there you can pull the APi onto a webpage using javascript using the example code on the documentation.



```
<script async
    src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap">
</script>
```

This code above sets up the API and lets you enter your API Key.

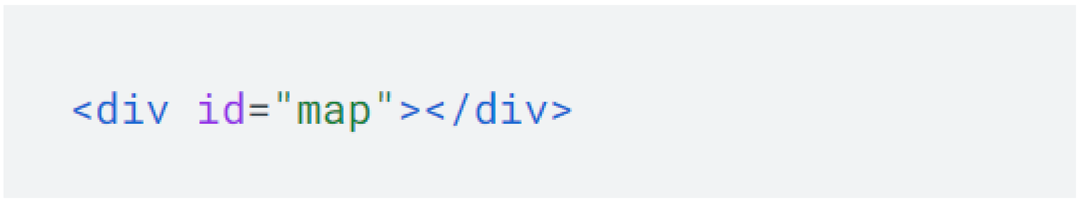
To view the map you have to set it up in css.



```
<style>
  #map {
    height: 100%;
  }
  html, body {
    height: 100%;
    margin: 0;
    padding: 0;
  }
</style>
```

Here you can see it uses # to set up an ID for the map and sets up the dimentions of it using css.

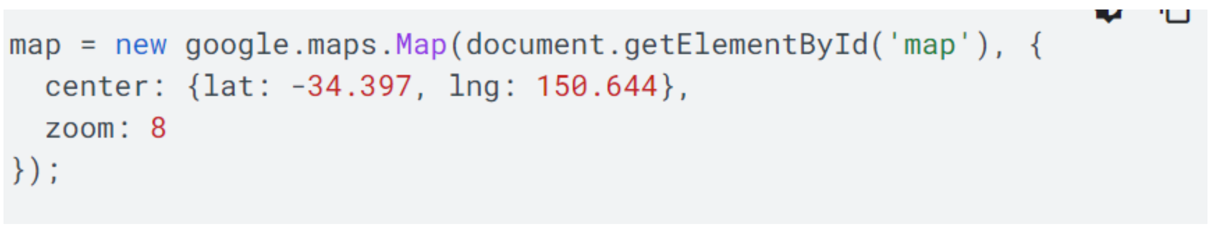
After this I followed the documentation which ues a div to set up the api to view



```
<div id="map"></div>
```

This is used to display the map on the web page and because it is a div it means we can place It where we like using css we can also make its values dynamic by using % signs to make it only fir a certain amount of the screen no matter its size.

The final thing the map needs to work is optiopns set up this is shown in the doumentation as 2 variables being reqired, one being center and the other is the zoom. The way to set th is up is within javascript as shown below.



```
map = new google.maps.Map(document.getElementById('map'), {
  center: {lat: -34.397, lng: 150.644},
  zoom: 8
});
```

The center varaible is where the map is centred at a location using latiturde and longitured couridates which is perfect for my audio guide app as my gps outputs latitude and longitude courdinates. The zoom level is based of how close in as often tou will want a street level instead of the entire world. The zoom levels go from 1-20 and an example list is shown on the documentation and goes as follows:

1: world

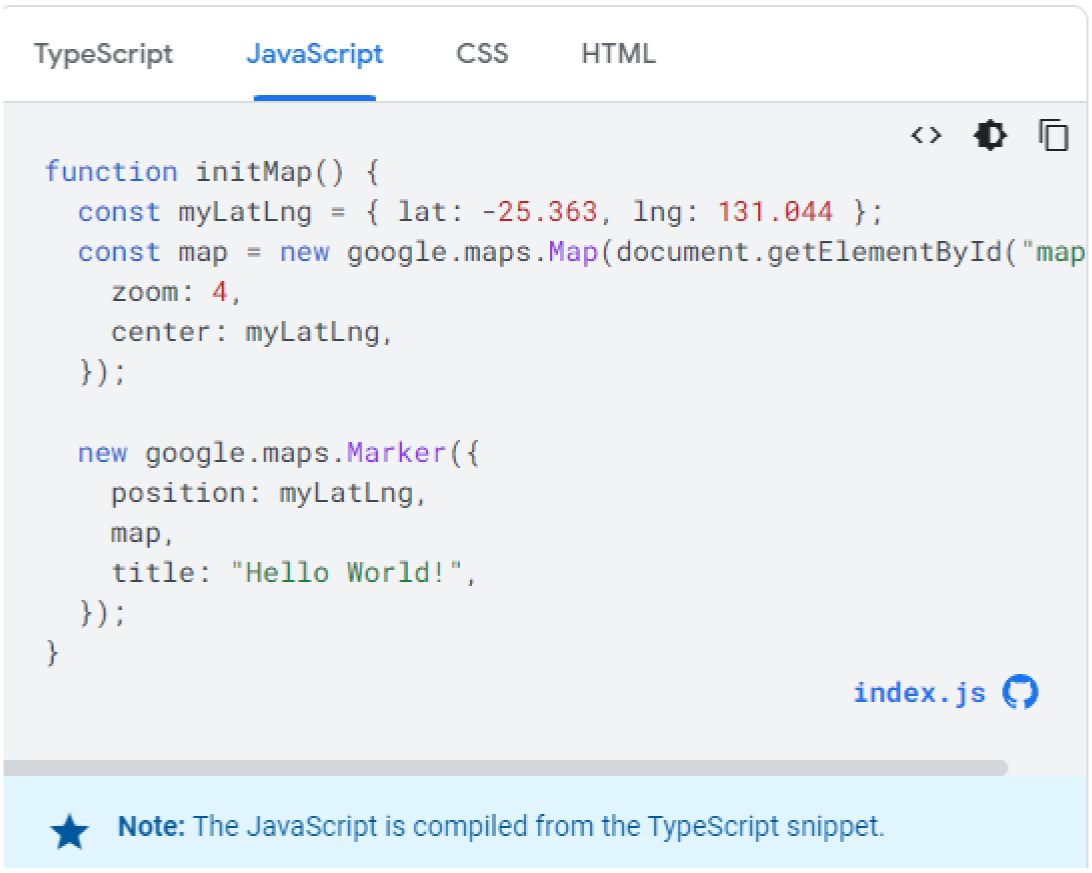
5: Landmass/contient

10: City

15: Streets

20: Buildings

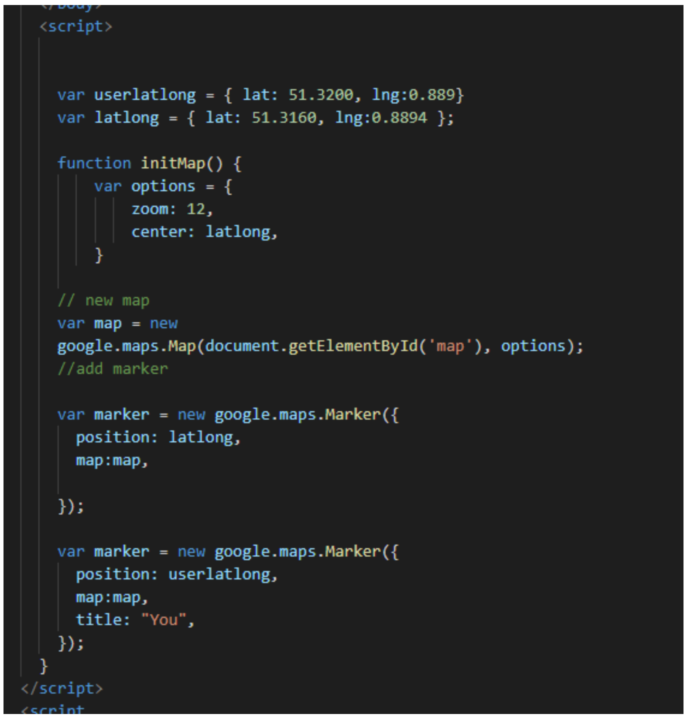
To display the users location I will be updating a custom marker and the center screen this weill be decided by 2 variables, one being the screen center and one being the user so that if the user wants to look else where they can by moving the map.



```
function initMap() {
  const myLatLng = { lat: -25.363, lng: 131.044 };
  const map = new google.maps.Map(document.getElementById("map
    zoom: 4,
    center: myLatLng,
  });

  new google.maps.Marker({
    position: myLatLng,
    map,
    title: "Hello World!",
  });
}
```

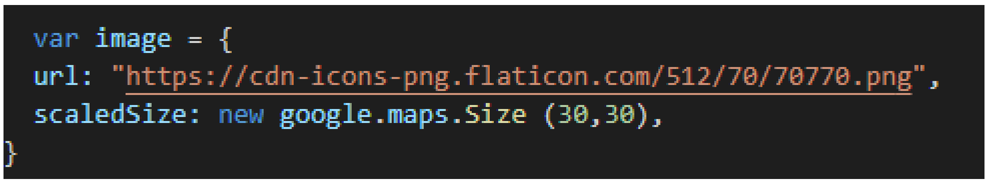
Note: The JavaScript is compiled from the TypeScript snippet.

Below is how I used the code

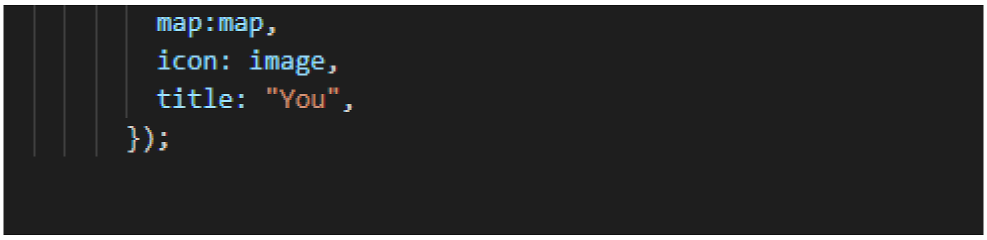


```
<script>


    var userlatlong = { lat: 51.3200, lng:0.889}
    var latlong = { lat: 51.3160, lng:0.8894 };

    function initMap() {
        var options = {
            zoom: 12,
            center: latlong,
        }

    // new map
    var map = new
    google.maps.Map(document.getElementById('map'), options);
    //add marker

    var marker = new google.maps.Marker({
      position: latlong,
      map:map,

    });

    var marker = new google.maps.Marker({
      position: userlatlong,
      map:map,
      title: "You",
    });
  }
</script>
```

From here I wanted to add a custom image icon for the user which was made by creating a variabled and assigning it a source for an image. After that, I can assign



```
  var image = {
  url: "https://cdn-icons-png.flaticon.com/512/70/70770.png",
  scaledSize: new google.maps.Size (30,30),
}
```

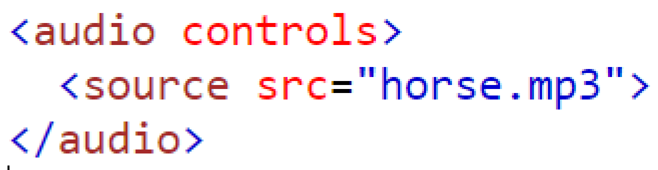


```
            map:map,
            icon: image,
            title: "You",
          });
```

Making the audio directory dynamic

In javescript I have decided to make the audio directory dynamic which allows me to change the audio as I need, for example If I wanted to change the audio from Recording1.mp3 to Recording2.mp3 I would be able to by changing a string which has the path for the audio within it.

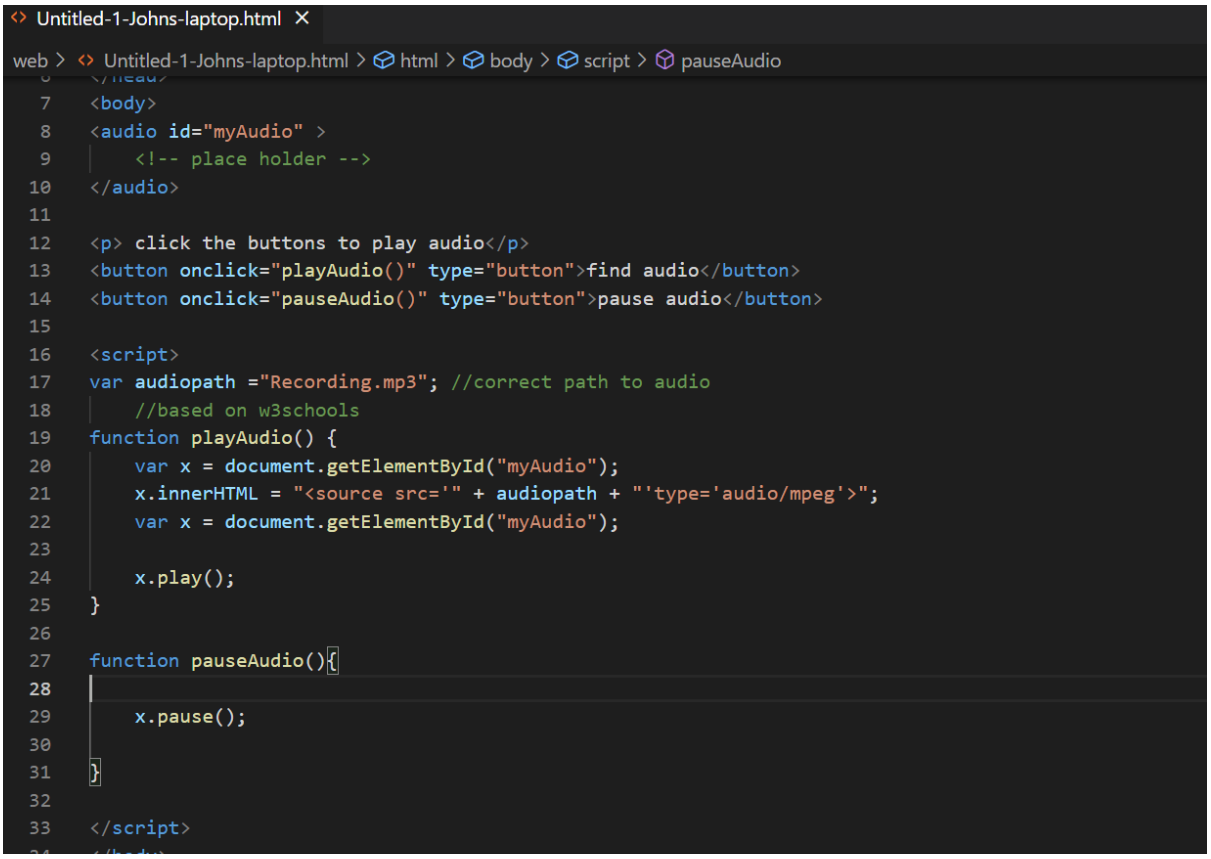
I started by looking at the html audio tag which allows a person to listen to audio of a file within the browser.



```
<audio controls>
  <source src="horse.mp3">
</audio>
```

here is an example how the audio tag and how to use it in html from w3schools. The Audio allows it to be set up in the audio format while the controls alows the creator to set options up such as stopping downaload as well as allowing the user to pause,play and set playback speed. With in the audio tag we can set it to auto play audio which I belive I will need o satisfy the user as I belive the user will need to be able to rewind audio as needed or even pause it. This will allow the user to control the audio as they need.

To make the auido directory dynamic I would need to make a a variable of the file path for the auidio file which we can do by making the file path a string. From here I would add an ID to the auido so that the javascript can interact with it. From there I have created a function which will create an audio source and then apend it between the audio tags. This is done by using document.getElementbyid now that we have the ellement we can change it. For example, appending a source to auido which is done bt using innerHTML which allows us to edit html though javascript. Here I have used it to add a source to the audio tag by updating the variable within it.

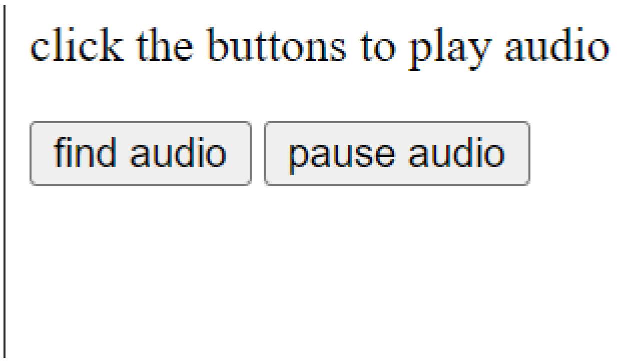


Untitled-1-Johns-laptop.html

web > Untitled-1-Johns-laptop.html > html > body > script > pauseAudio

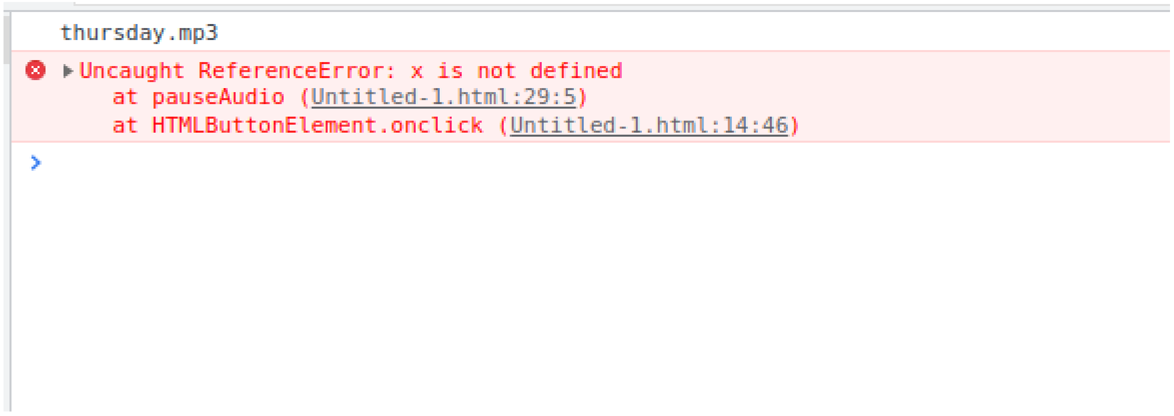
```
<body>
<audio id="myAudio" >
    <!-- place holder -->
</audio>

<p> click the buttons to play audio</p>
<button onclick="playAudio()" type="button">find audio</button>
<button onclick="pauseAudio()" type="button">pause audio</button>

<script>
var audiopath ="Recording.mp3"; //correct path to audio
    //based on w3schools
function playAudio() {
    var x = document.getElementById("myAudio");
    x.innerHTML = "<source src='" + audiopath + "'type='audio/mpeg'>";
    var x = document.getElementById("myAudio");

    x.play();
}

function pauseAudio(){


    x.pause();

}

</script>
```

This is how it looks on the webpage.



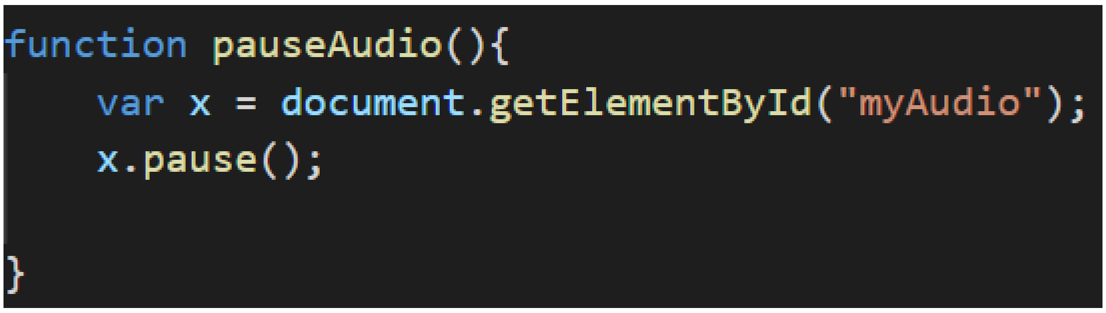
click the buttons to play audio

From here I tired the pause button which would come up with this error in the log.



```
thursday.mp3
▶Uncaught ReferenceError: x is not defined
    at pauseAudio (Untitled-1.html:29:5)
    at HTMLButtonElement.onclick (Untitled-1.html:14:46)
>
```

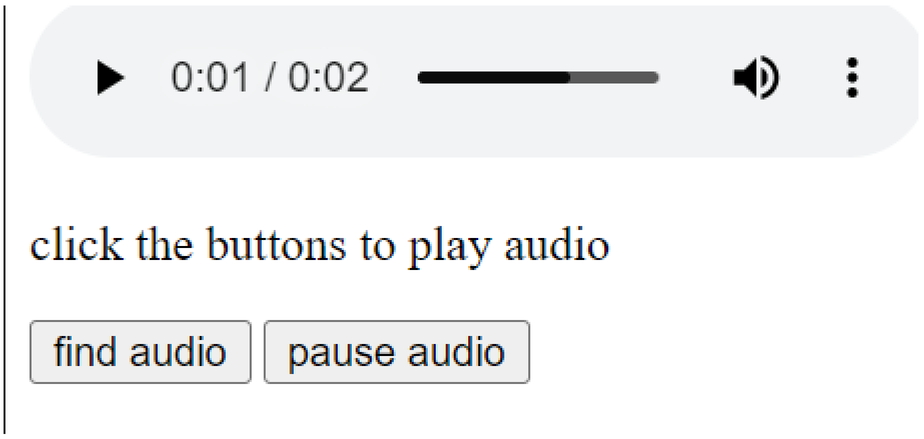
So it seems as if the function is not able to grab the varible which has the audio for. So to fix this I implemented a line of code which gets audio via document.getElementByID and the assignes to a varible which can be maipulated.



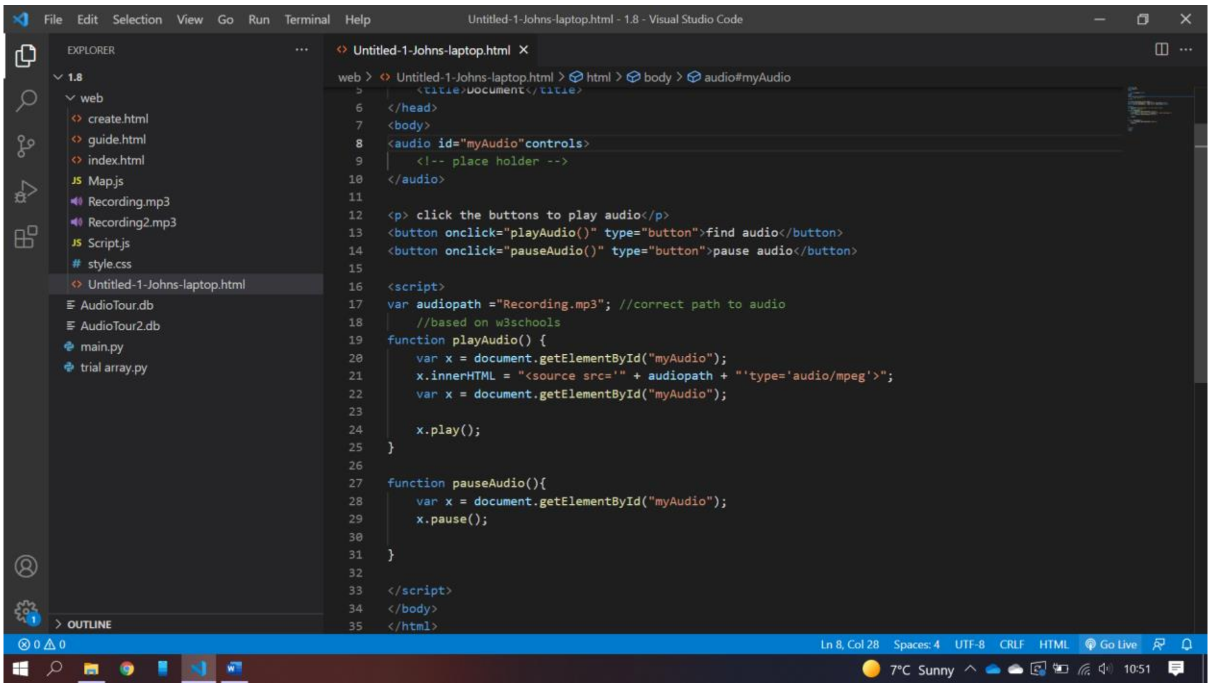
```
function pauseAudio(){
    var x = document.getElementById("myAudio");
    x.pause();

}
```

The pause button now works to pause the audio.

From here i reasied I have not tried setting up audio controlls with in the audio tag so from here I tried addeding controls which resulted in this.



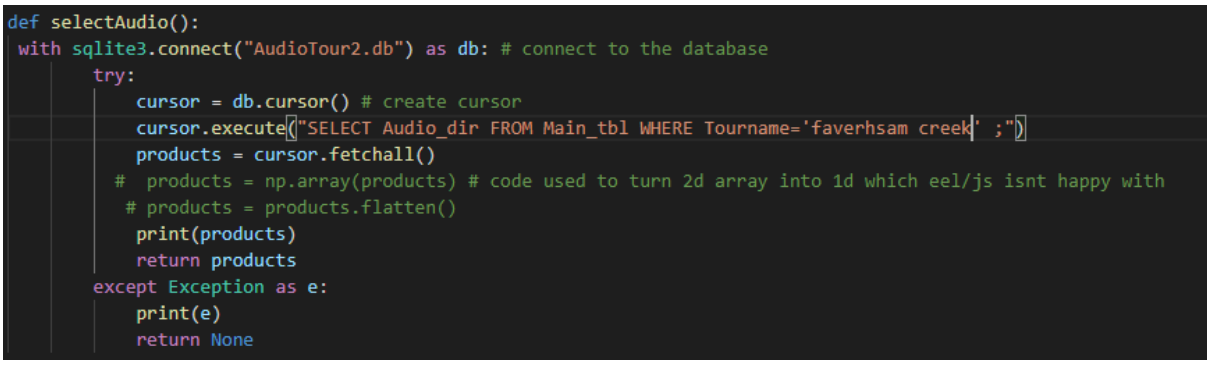
I belive this works a lot better than the pause audio button and instead of a play audio button it will now be a find audio button which could be turned into a function that starts palying the audio. Below is the code for this.


File Edit Selection View Go Run Terminal Help
Untitled-1-Johns-laptop.html - 1.8 - Visual Studio Code
EXPLORER
1.8
web
create.html
guide.html
index.html
Map.js
Recording.mp3
Recording2.mp3
Script.js
style.css
Untitled-1-Johns-laptop.html
AudioTour.db
AudioTour2.db
main.py
trial array.py
OUTLINE
web > Untitled-1-Johns-laptop.html > html > body > audio#myAudio
Ln 8, Col 28 Spaces: 4 UTF-8 CRLF HTML Go Live
7°C Sunny 10:51

```html
    <title>Document</title>
</head>
<body>
<audio id="myAudio"controls>
    <!-- place holder -->
</audio>

<p> click the buttons to play audio</p>
<button onclick="playAudio()" type="button">find audio</button>
<button onclick="pauseAudio()" type="button">pause audio</button>

<script>
var audiopath ="Recording.mp3"; //correct path to audio
    //based on w3schools
function playAudio() {
    var x = document.getElementById("myAudio");
    x.innerHTML = "<source src='" + audiopath + "'type='audio/mpeg'>";
    var x = document.getElementById("myAudio");

    x.play();
}

function pauseAudio(){
    var x = document.getElementById("myAudio");
    x.pause();

}

</script>
</body>
</html>
```

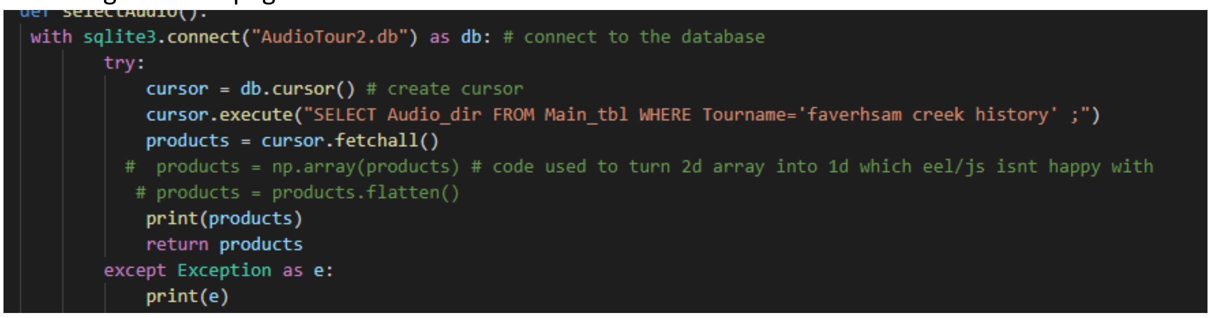
Pulling path from a database

From here I need to implement the auido to the main page and then get the auido from a databse via an SQL statement. To do this is used a function I used previouly to get data from a database via an SQL satement. I have then changed the SQL stament to get data from a different section of the database.

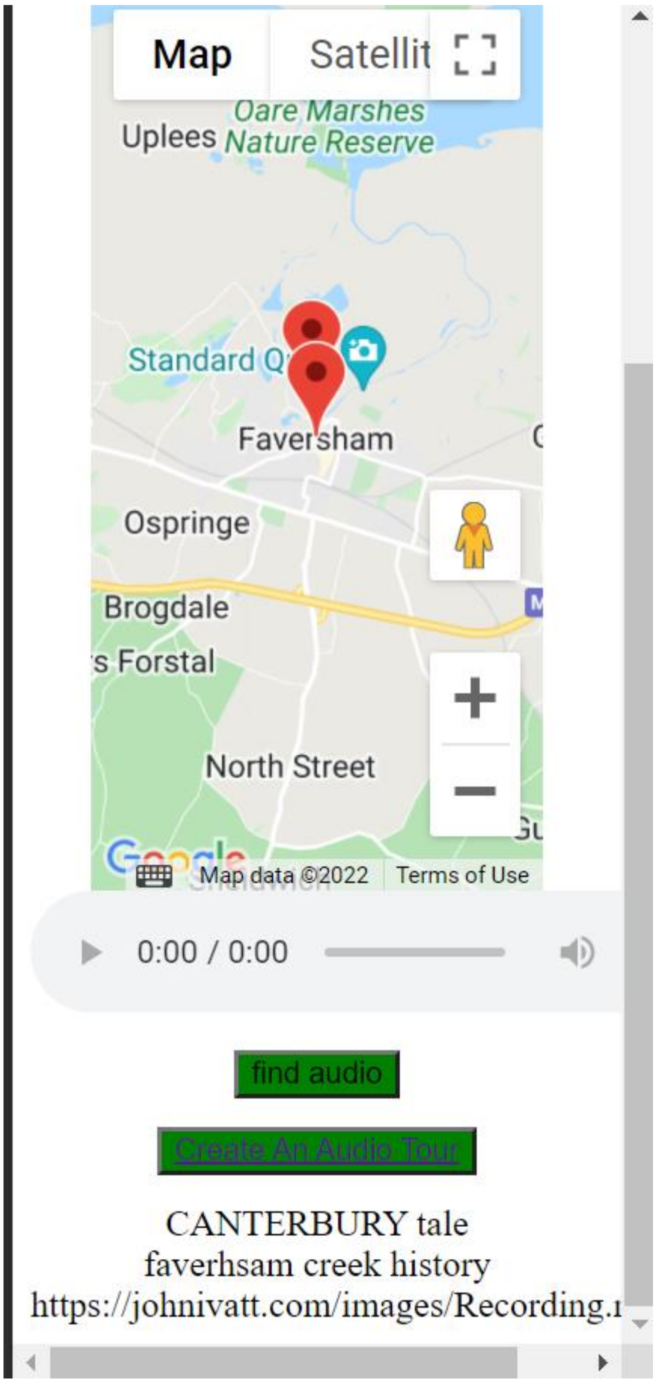


```python
def selectAudio():
 with sqlite3.connect("AudioTour2.db") as db: # connect to the database
        try:
            cursor = db.cursor() # create cursor
            cursor.execute("SELECT Audio_dir FROM Main_tbl WHERE Tourname='faverhsam creek' ;")
            products = cursor.fetchall()
          #  products = np.array(products) # code used to turn 2d array into 1d which eel/js isnt happy with
           # products = products.flatten()
            print(products)
            return products
        except Exception as e:
            print(e)
            return None
```

From here I have tried getting the Data from an SQL database and pass it though. For some reason the GUI would hold up the SQL stament so it would only print it once I closed the GUI which lead me and my teacher to think it could be a threading isuse which would be a big problem. After worring that this was a limitation it turns out I got the SQL statement wrong. From here I was able to pass the string onto the page of the html file.



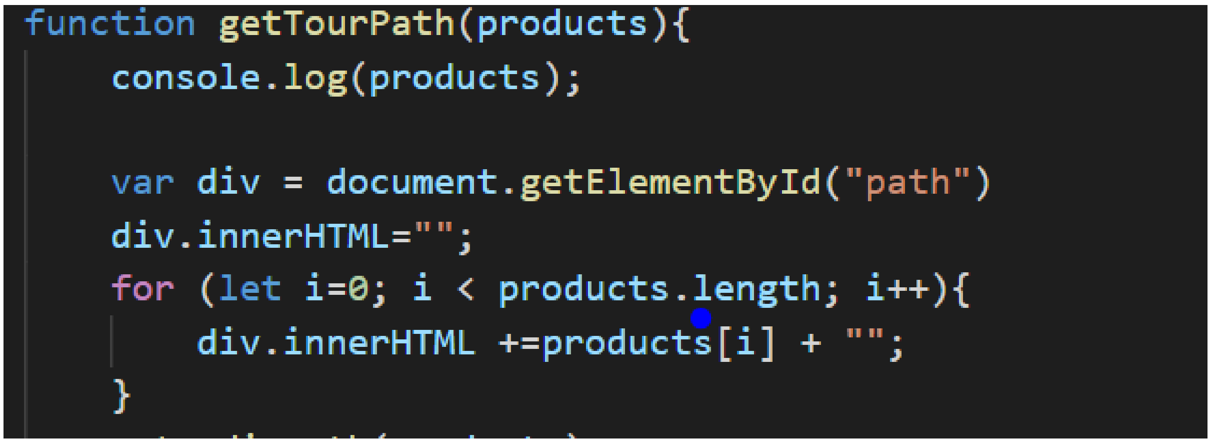
```
def selectAudio():
 with sqlite3.connect("AudioTour2.db") as db: # connect to the database
        try:
            cursor = db.cursor() # create cursor
            cursor.execute("SELECT Audio_dir FROM Main_tbl WHERE Tourname='faverhsam creek history' ;")
            products = cursor.fetchall()
          #  products = np.array(products) # code used to turn 2d array into 1d which eel/js isnt happy with
           # products = products.flatten()
            print(products)
            return products
        except Exception as e:
            print(e)
```



find audio

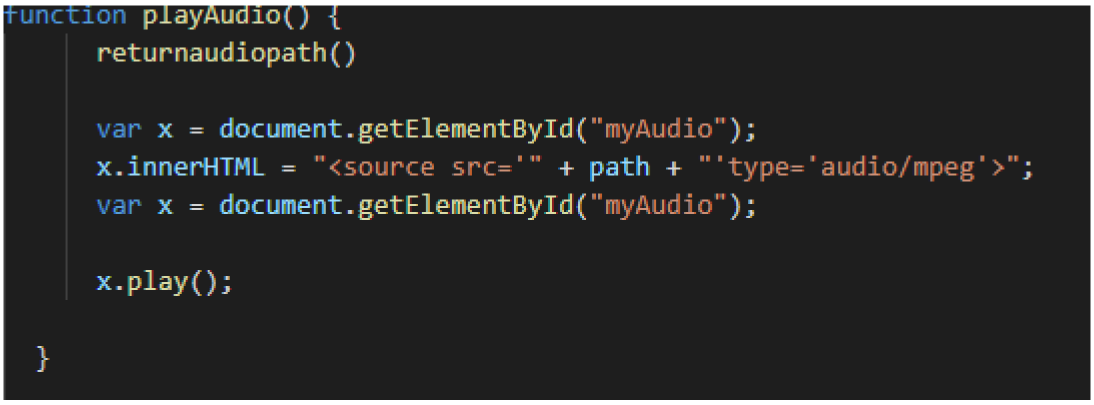
Create An Audio Tour

CANTERBURY tale
faverhsam creek history
https://johnivatt.com/images/Recording.r



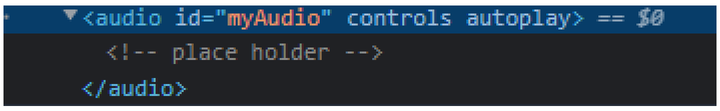
```
function getTourPath(products){
    console.log(products);

    var div = document.getElementById("path")
    div.innerHTML="";
    for (let i=0; i < products.length; i++){
        div.innerHTML +=products[i] + "";
    }
```

Looking at the code above theis is the code that writes the path onto the webapge for it to be ripped later on. After this this we pull the page by using document.getElementById and then assiging it to a variable. From here I asssign the vatible using inner html with a new string og “<source src’” then I concatinate the path via + path and then I add +”’type=’Audio/mped’>” which then is written to the page when the line is finished.

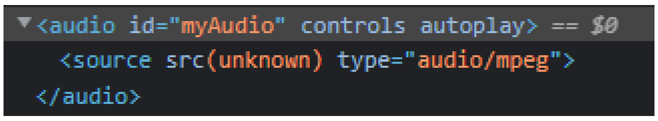


```
function playAudio() {
    returnaudiopath()

    var x = document.getElementById("myAudio");
    x.innerHTML = "<source src='" + path + "'type='audio/mpeg'>";
    var x = document.getElementById("myAudio");

    x.play();

}
```

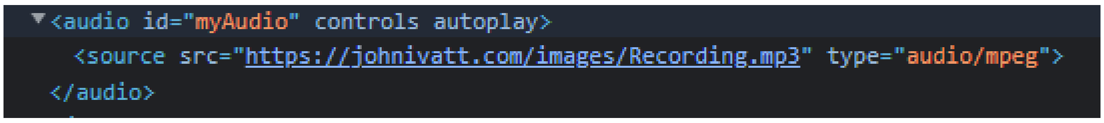
After this it was working but it requires 2 clicks this is due to the fact on the first click it would write the the sauce tag for the audio but not the path. An example of this is below.



```
▼<audio id="myAudio" controls autoplay> == $0
   <!-- place holder -->
 </audio>
```



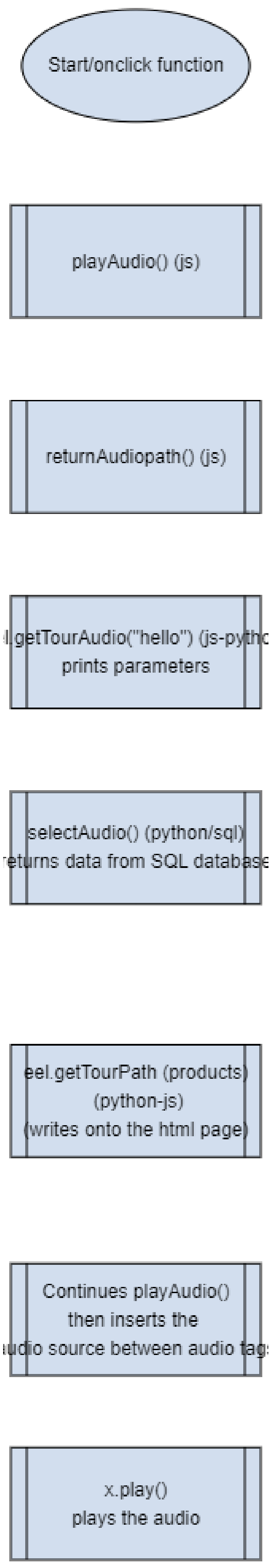
```
▼<audio id="myAudio" controls autoplay> == $0
   <source src(unknown) type="audio/mpeg">
 </audio>
```



```
▼<audio id="myAudio" controls autoplay>
   <source src="https://johnivatt.com/images/Recording.mp3" type="audio/mpeg">
 </audio>
```

The first image is before the click which shows nothing is written between the tags.

On the second click we can see the source tags have been implemented but there is no path on the audio which leads me to belive that on the 2nd click something else happening or crashing for it to be inserted. As finnaly it srtarts to play the audio. To attemt to fix this, I tried to run the function 2x when the button is clicked which didn’t work either. From here I decided to trace the function using a flow chart.

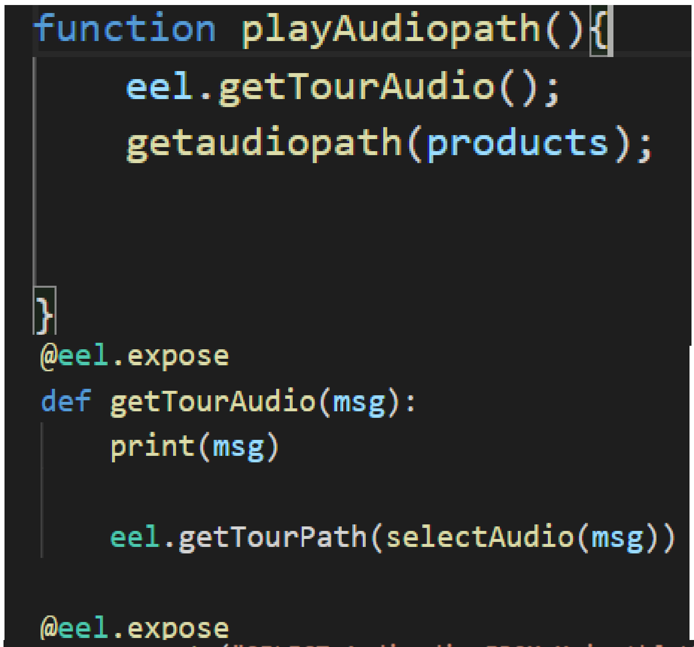

Start/onclick function
playAudio() (js)
returnAudiopath() (js)
l.getTourAudio("hello") (js-pytho
prints parameters
selectAudio() (python/sql)
eturns data from SQL databas
eel.getTourPath (products)
(python-js)
(writes onto the html page)
Continues playAudio()
then inserts the
udio source between audio tag
x.play()
plays the audio

For this to start we must we must click the button which then would start playing the audio tour.

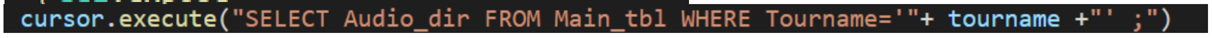
Here is where it starts to interact with python to start the SQL statement

This is the function that gets the SQL data.

From the flowchart above we can see where the datagoes and where it comes from and how it is used. To fix the issue of the function having to be ran 2x I have moved some of the functions around and will now reduce the amunt of function needed. While reducing the number of functions I will also be doing decompistion to find what ones I need. To start fixing the issue of having to click the button 2x I looked through the flowchat and started working out what is needed as well as what was not needed. Once the onclick event is it starts a function it then starts the eel function in python which then starts a function that reads data from an SQL statement. The SQL statement retrieves the data from a database and then prints and returns the audio path the SQL statement consisits of Select as we are reading data then the colum name frim the tabke where the name of the tour is equal to the variable that we are passing though the peramiter as we are making it dynamic to take many inputs. I will talk more about making it dynamic later on.

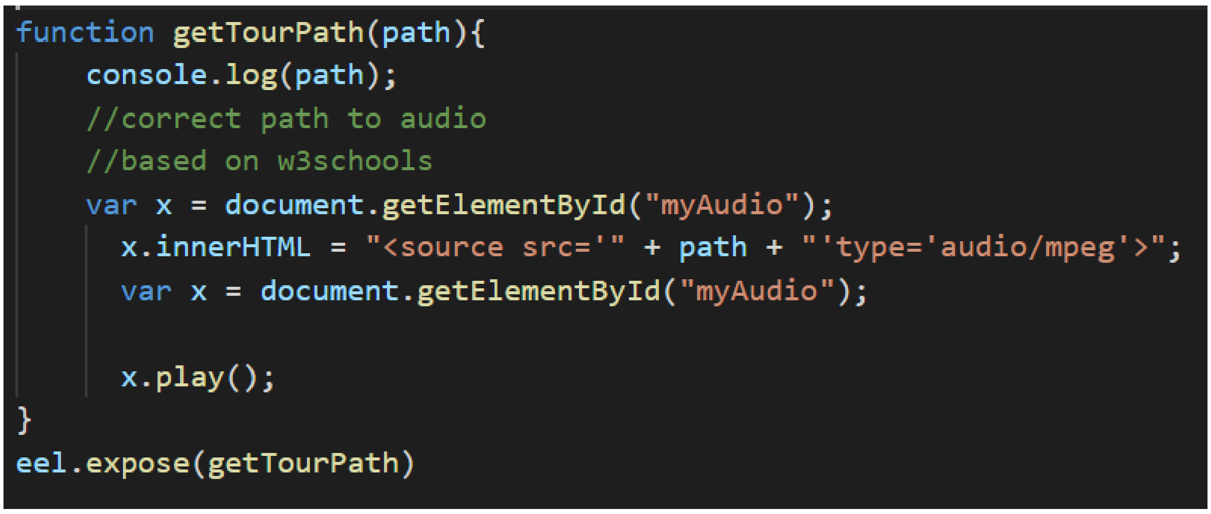


```
function playAudiopath(){
    eel.getTourAudio();
    getaudiopath(products);


}
@eel.expose
def getTourAudio(msg):
    print(msg)

    eel.getTourPath(selectAudio(msg))

@eel.expose
```



```
cursor.execute("SELECT Audio_dir FROM Main_tbl WHERE Tourname='"+ tourname +"' ;")
```

Below is the final part of the audio code where it is written into the pagr with xinner html from code with w3shools and is plated with x.play(); which is an inbuild javascript fuction that can control audio and video.

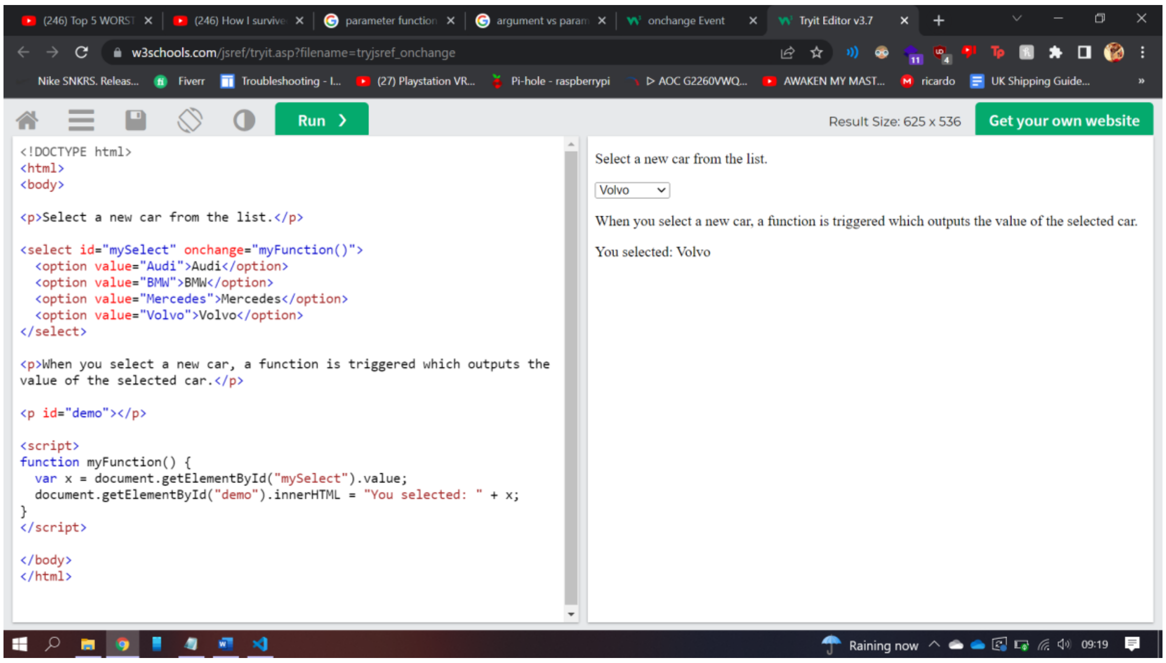


```
function getTourPath(path){
    console.log(path);
    //correct path to audio
    //based on w3schools
    var x = document.getElementById("myAudio");
      x.innerHTML = "<source src='" + path + "'type='audio/mpeg'>";
      var x = document.getElementById("myAudio");

      x.play();
}
eel.expose(getTourPath)
```

Making Audio Dynamic

To make the audio dyncamic I started by thinking aboit what parts need to change to make it dynamic annd as we saw above the main function that I had to change was the SQL statement which I made dynamic via concatinging a 1 string and a varibe and anthother string to make a full SQL Statement. From here it was just adding peramiters and arguemnts to the functions. The most difficaalt part of this was making it so the user could create an argument. So to go back to the bain of my code the drop down menu I have used an onchange event which can make a function run to check what the selection box currently is after it has changed to do this I looked at w3schools example code of it and then chnaged it into what I needed it to do. In w3schools example it uses a static load of values while mine is dynamic which made me wonder if it was an issue but thnkfully it is not as mine adds these options using an onload once pulled from a database. For w3schools example once selected it is written onto the page. While with my code the on selectAudio for me updates a global varaible which then is used as a argument when calling other functions such as the SQL statement as the stament uses the Audio Tour name to select the audio.

W3schools example



```html
<!DOCTYPE html>
<html>
<body>

<p>Select a new car from the list.</p>

<select id="mySelect" onchange="myFunction()">
  <option value="Audi">Audi</option>
  <option value="BMW">BMW</option>
  <option value="Mercedes">Mercedes</option>
  <option value="Volvo">Volvo</option>
</select>

<p>When you select a new car, a function is triggered which outputs the
value of the selected car.</p>

<p id="demo"></p>

<script>
function myFunction() {
  var x = document.getElementById("mySelect").value;
  document.getElementById("demo").innerHTML = "You selected: " + x;
}
</script>

</body>
</html>
```

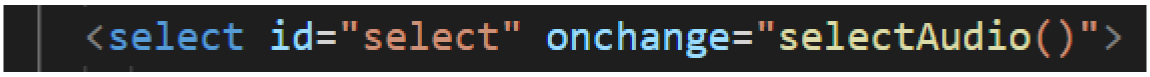
Select a new car from the list.

Volvo

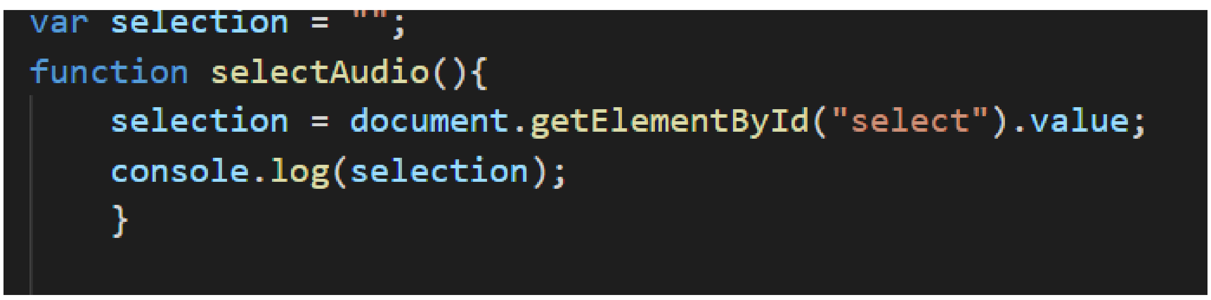
When you select a new car, a function is triggered which outputs the value of the selected car.

You selected: Volvo

My work

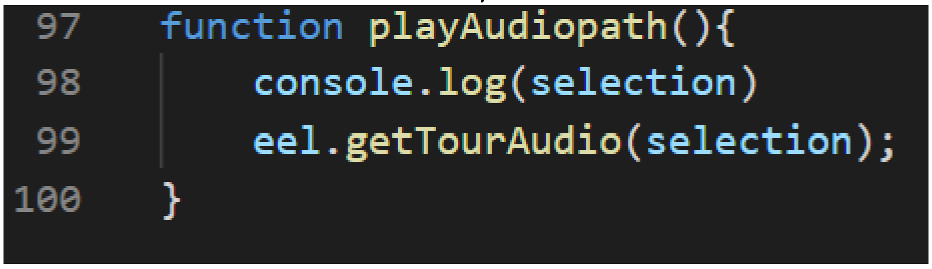


```html
<select id="select" onchange="selectAudio()">
```



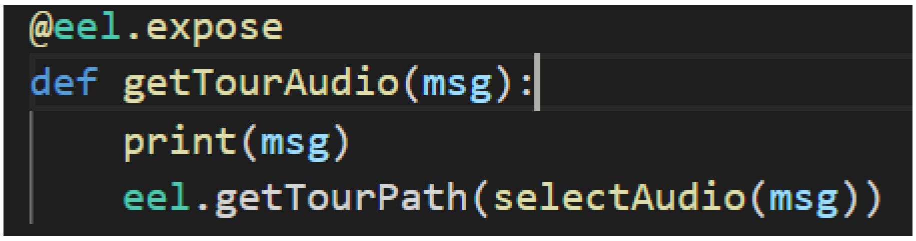
```javascript
var selection = "";
function selectAudio(){
    selection = document.getElementById("select").value;
    console.log(selection);
    }
```

Assigning the varariable outside the function makes it a global varable which then lets me use it in other functuions to be maipulated as well as used as information for what the user has selected. Which will be used to make the Audio dynamic.

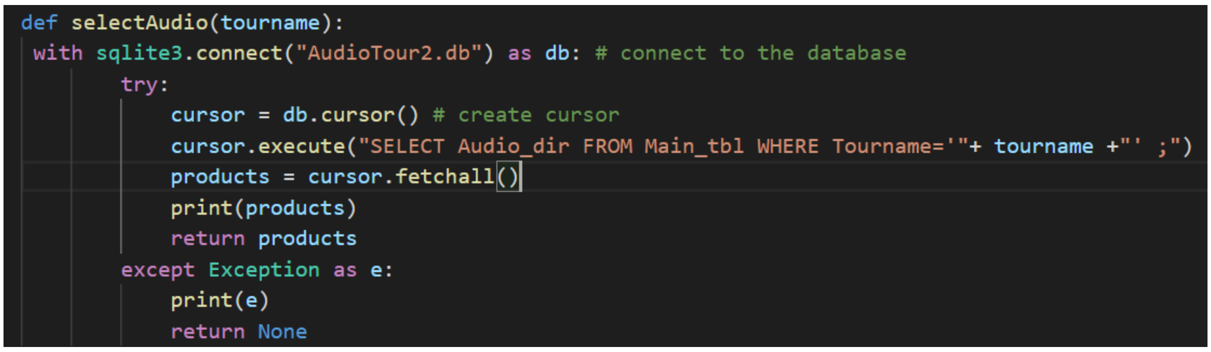


```
function playAudiopath(){
    console.log(selection)
    eel.getTourAudio(selection);
}
```

From here we start a console.log for error checking and then from there we run eel.getTourAudio with the argument of slection which is what the user selected from the drop down at the top for. From here a python function. A function that then acesses a databse via an SQL statement also uses a arugument passed though as a peramiter to the eels function which includes the name of the audio for the database to get the direcotry.

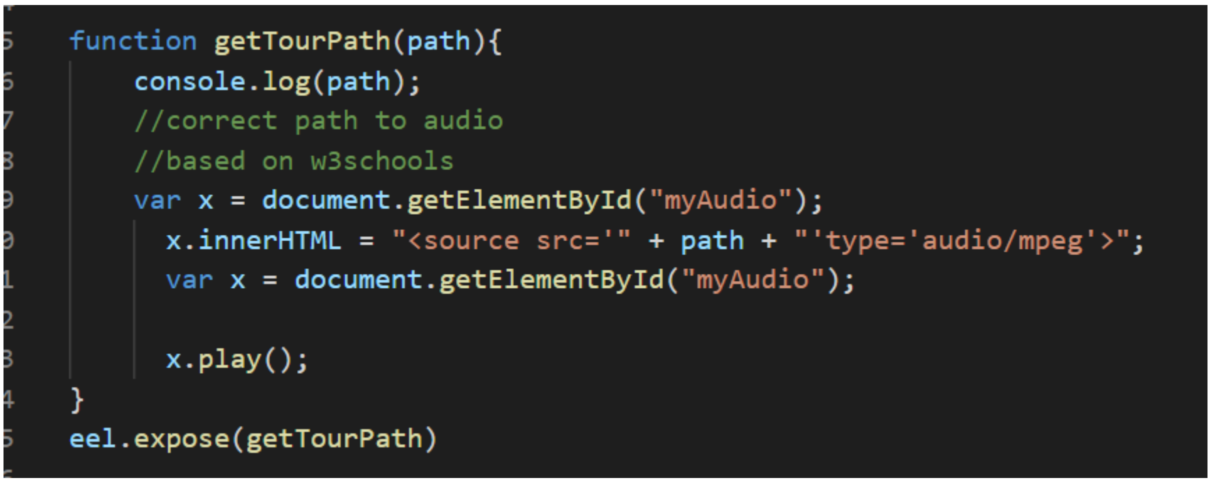


```
@eel.expose
def getTourAudio(msg):
    print(msg)
    eel.getTourPath(selectAudio(msg))
```



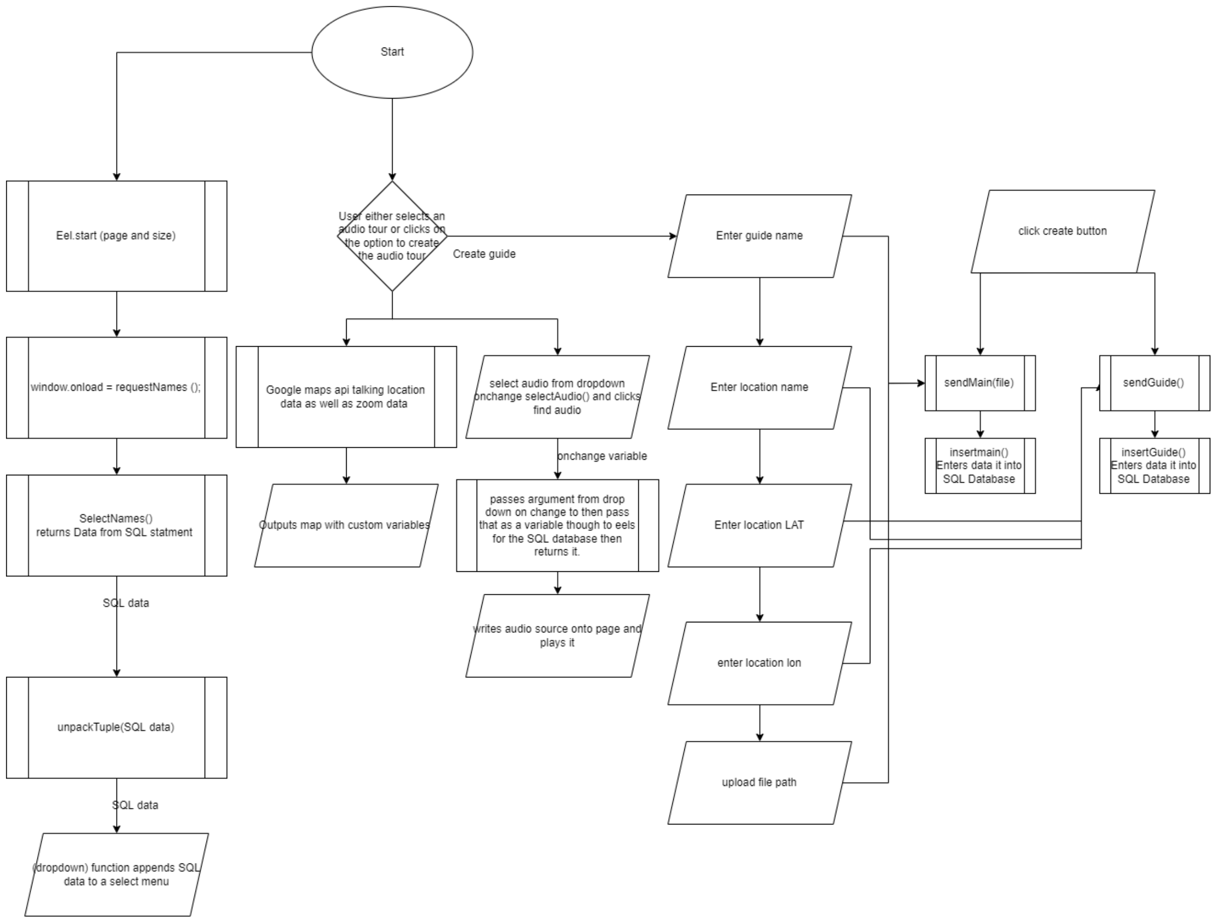
```
def selectAudio(tourname):
    with sqlite3.connect("AudioTour2.db") as db: # connect to the database
        try:
            cursor = db.cursor() # create cursor
            cursor.execute("SELECT Audio_dir FROM Main_tbl WHERE Tourname='"+ tourname +"' ;")
            products = cursor.fetchall()
            print(products)
            return products
        except Exception as e:
            print(e)
            return None
```

As we can see once again the peramiter that includes the tours name for the function is used to tell the SQL statement where to look for the audio directory which is then returned to javascript as an argument though an eels function. This argument passed through as a peramiter which will be the directory/path for the location of the audio.

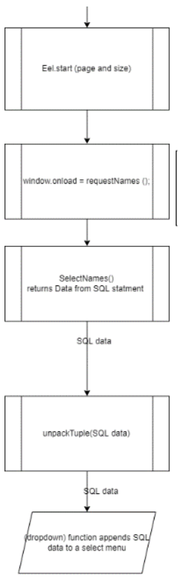


```
function getTourPath(path){
    console.log(path);
    //correct path to audio
    //based on w3schools
    var x = document.getElementById("myAudio");
      x.innerHTML = "<source src='" + path + "'type='audio/mpeg'>";
      var x = document.getElementById("myAudio");

      x.play();
}
eel.expose(getTourPath)
```

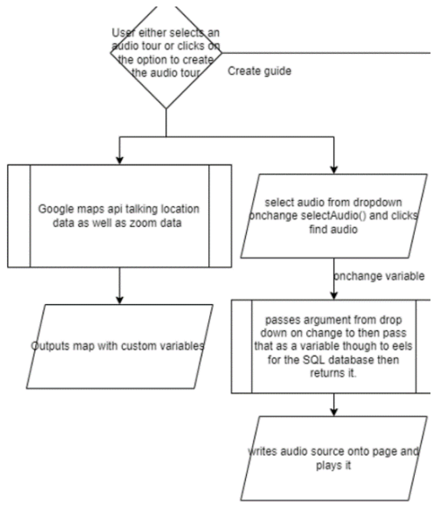
Updated flow chart for most current itteration.


Start
Eel.start (page and size)
window.onload = requestNames ();
SelectNames()
returns Data from SQL statment
SQL data
unpackTuple(SQL data)
SQL data
(dropdown) function appends SQL data to a select menu
User either selects an audio tour or clicks on the option to create the audio tour
Create guide
Google maps api talking location data as well as zoom data
Outputs map with custom variables
select audio from dropdown onchange selectAudio() and clicks find audio
onchange variable
passes argument from drop down on change to then pass that as a variable though to eels for the SQL database then returns it.
writes audio source onto page and plays it
Enter guide name
Enter location name
Enter location LAT
enter location lon
upload file path
click create button
sendMain(file)
insertmain()
Enters data it into SQL Database
sendGuide()
insertGuide()
Enters data it into SQL Database

As the project has gotten so far along I belive it is time to redo the flowchart to show how it has changed. I belive this maybe how it will finnaly look for the project due to time constraints and how little there is much do to left. From here I will go though and describe each main function of the dataflow diagram from left to right.

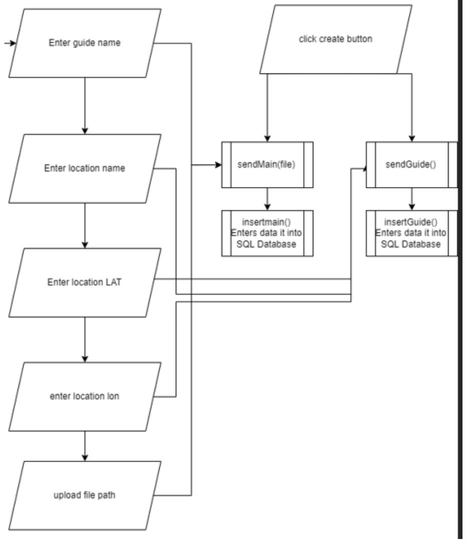

Eel.start (page and size)
window.onload = requestNames ();
SelectNames()
returns Data from SQL statment
SQL data
unpackTuple(SQL data)
SQL data
(dropdown) function appends SQL data to a select menu

This is the set of functions which creates the drop down it starts with the the eels.start which inualizes the page with the peramiters of what browser it is using as well as what page it starts on, the position and the size of said browser. From here it runs an onload script though javascipt which starts to request the data to create the dropdown from data gathered from a databse gathered via SQL. This data is then unpacked after it is gathred via the SelectNames function which gatheres the SQL data. After it is unpacked it is then returned back to javascript as a peramiter to be turned into an a dropdown using javascript.


User either selects an audio tour or clicks on the option to create the audio tour
Create guide
Google maps api talking location data as well as zoom data
Outputs map with custom variables
select audio from dropdown onchange selectAudio() and clicks find audio
onchange variable
passes argument from drop down on change to then pass that as a variable though to eels for the SQL database then returns it.
writes audio source onto page and plays it

If the user decides to not click on the create guide and the user selects an audio tour out of the drop down.

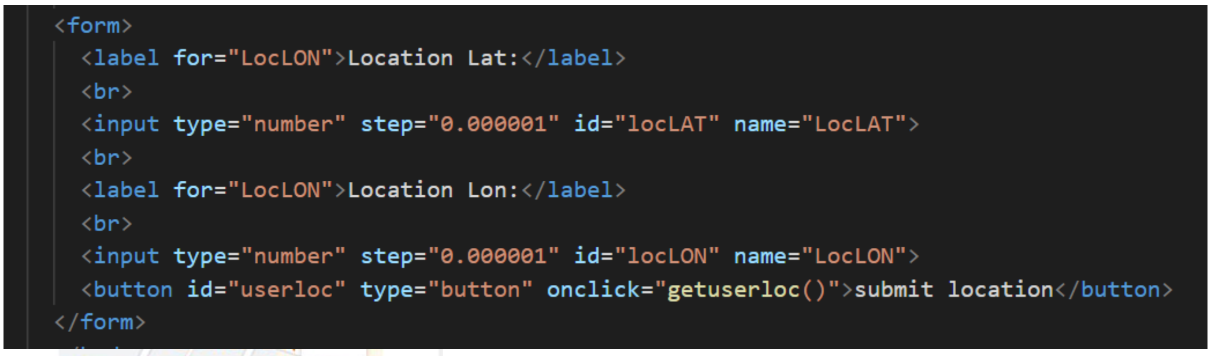
This dropdown lets the user select what audio they want to listen to as well as the location for an audio tour on the map. Once selected the location for the audio tour is presented to the google maps api. The source for the audio is also selected via the drop down menu and is passed though eels and then used as a peramiter to get data from an SQL statement and is then written into the HTML via Javascript.



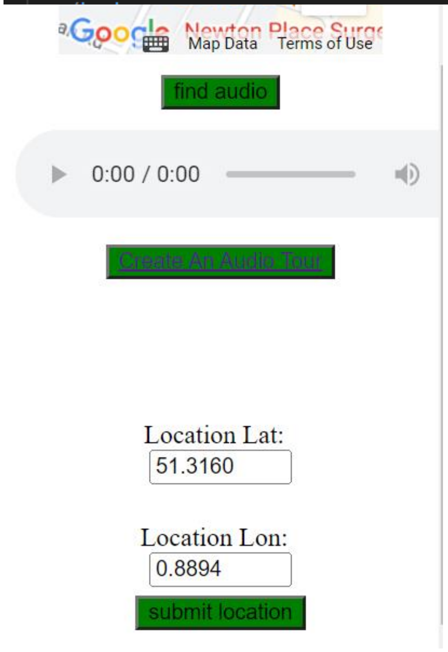
If the user selected the create guide button they would be taken to a page with 54 input boxes and a button to upload file/web directory. All this is then gathred via get element via id and then put into an SQL database to be pulled from later.

Making google maps api dynamic

To make google maps dynamic Ive decided to start by making it so the user can update their location via entering their latitude and logituitde into 2 different boxes which are then used as a varoble for the center of the map and as a varible for the users location which is based apon the center of the map. I looked at how ived used input boxes previously and have decided that a form would be the best cource of action to gather the data inputed via a user. Below we can see the HTML code I have used to create the from. There is also an image of how it looks on the eels page.



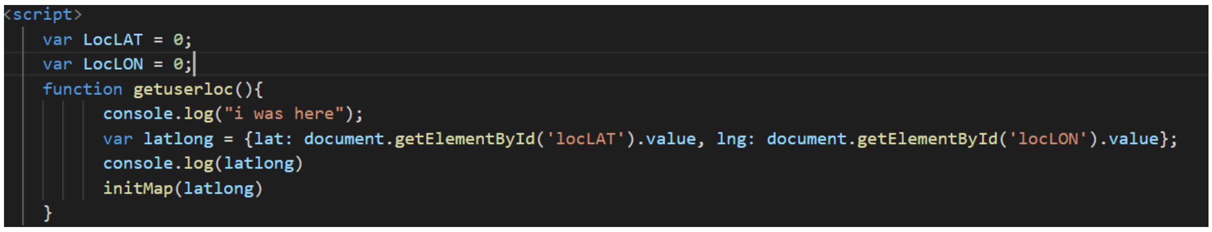
```
<form>
  <label for="LocLON">Location Lat:</label>
  <br>
  <input type="number" step="0.000001" id="locLAT" name="LocLAT">
  <br>
  <label for="LocLON">Location Lon:</label>
  <br>
  <input type="number" step="0.000001" id="locLON" name="LocLON">
  <button id="userloc" type="button" onclick="getuserloc()">submit location</button>
</form>
```



Location Lat:
51.3160

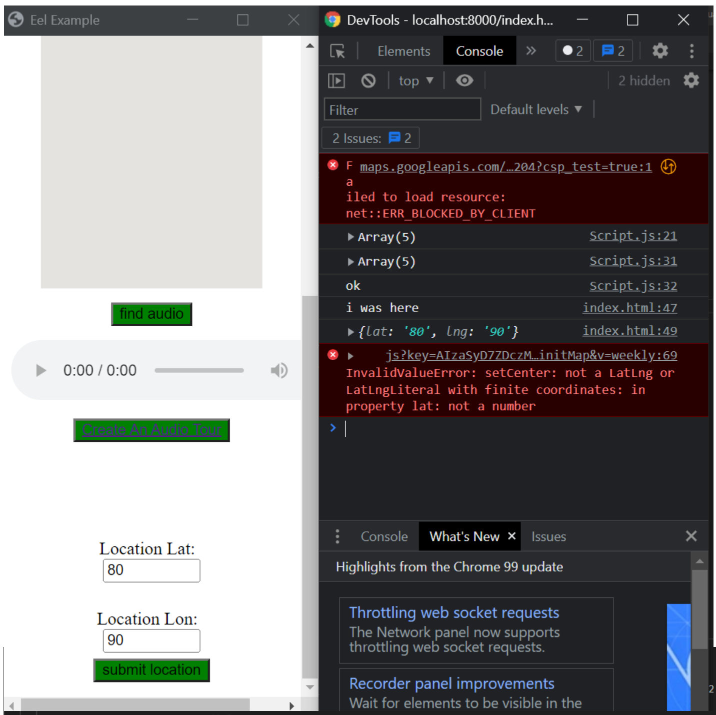
Location Lon:
0.8894

As we can see there is now an input box for the users latitude and longitude and a submit button. Both input boxes have titles so the user knows where to input each courdinate so that they do not enter them the wrong way round. Each input box is set up so that the user can only enter a float as that is what longtuide and latuide is.



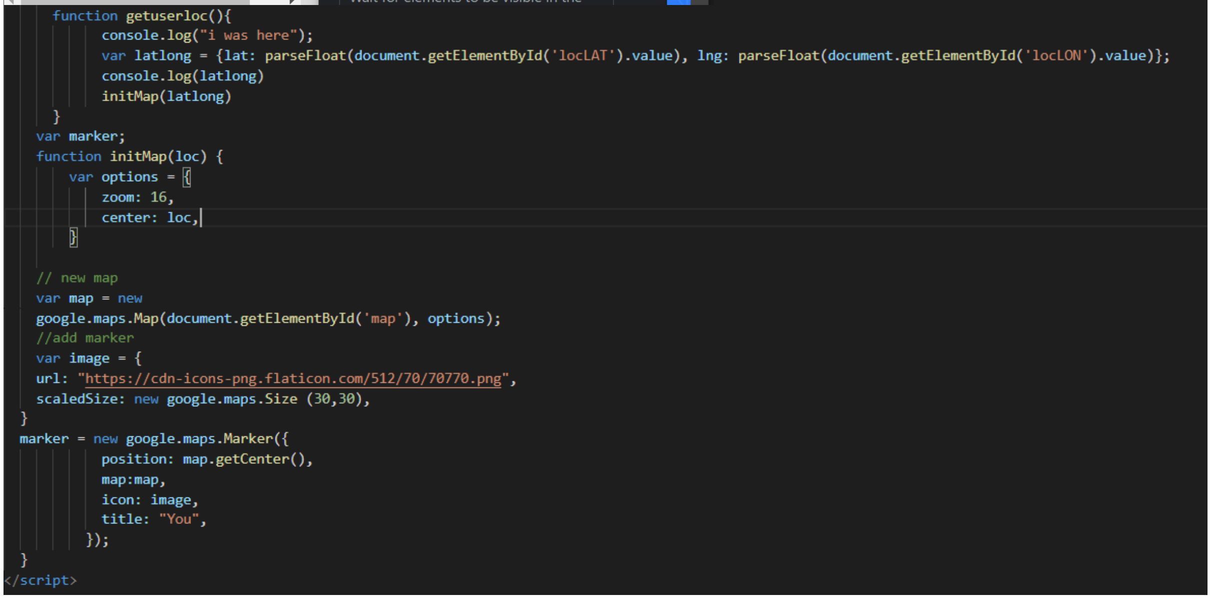
```
<script>
    var LocLAT = 0;
    var LocLON = 0;
    function getuserloc(){
            console.log("i was here");
            var latlong = {lat: document.getElementById('locLAT').value, lng: document.getElementById('locLON').value};
            console.log(latlong)
            initMap(latlong)
    }
```

To start the javascript code I assign 2 global varibles one being LocLAT and LocLON which will be used to assign the Audio tour location later on but for now the function we are looking at is called getuserloc() which creates a Dictionary and assigns to to a varible called latlong. The first key is lat which the value for is gathred iva a getElementById('locLAT') which returns the data the user entred unto the input box with the id locLAT this is then repeared for LocLON but with the key being lng and the id for the getElementById being locLON


Eel Example
find audio
0:00 / 0:00
Create An Audio Tour
Location Lat:
80
Location Lon:
90
submit location
DevTools - localhost:8000/index.h...
Elements
Console
top
2 hidden
Filter
Default levels
2 Issues: 2
Console
What's New
Issues
Highlights from the Chrome 99 update
Throttling web socket requests
The Network panel now supports throttling web socket requests.
Recorder panel improvements
Wait for elements to be visible in the

```
F maps.googleapis.com/…204?csp_test=true:1
a
iled to load resource:
net::ERR_BLOCKED_BY_CLIENT
▸ Array(5)                                  Script.js:21
▸ Array(5)                                  Script.js:31
ok                                          Script.js:32
i was here                                  index.html:47
▸ {lat: '80', lng: '90'}                    index.html:49
▸    js?key=AIzaSyD7ZDczM…initMap&v=weekly:69
InvalidValueError: setCenter: not a LatLng or
LatLngLiteral with finite coordinates: in
property lat: not a number
```

However once this data was pulled in to the API I would get an error of it not being a float which made me confused as the user can only enter a float. To fix this I first looked it up online which lead to a stack overflow thread of someone with the same issue and the fix was to make sure the input is a float to do this I used parseFloat() which is a function built itnto javascipt which can turn a value into a float. W3schools has



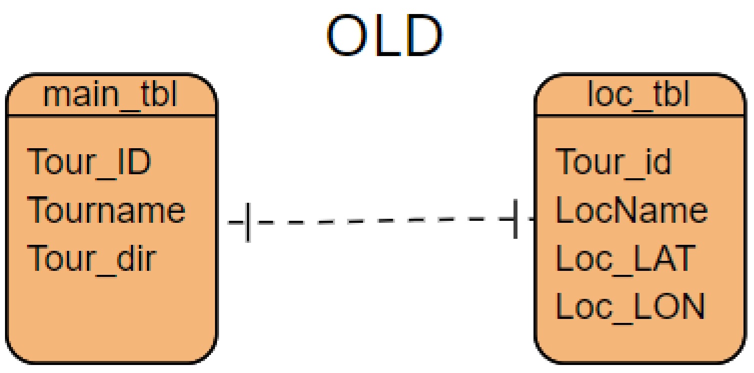
```
    function getuserloc(){
            console.log("i was here");
            var latlong = {lat: parseFloat(document.getElementById('locLAT').value), lng: parseFloat(document.getElementById('locLON').value)};
            console.log(latlong)
            initMap(latlong)
    }
    var marker;
    function initMap(loc) {
        var options = {
            zoom: 16,
            center: loc,
        }

    // new map
    var map = new
    google.maps.Map(document.getElementById('map'), options);
    //add marker
    var image = {
    url: "https://cdn-icons-png.flaticon.com/512/70/70770.png",
    scaledSize: new google.maps.Size (30,30),
  }
  marker = new google.maps.Marker({
            position: map.getCenter(),
            map:map,
            icon: image,
            title: "You",
          });
  }
</script>
```

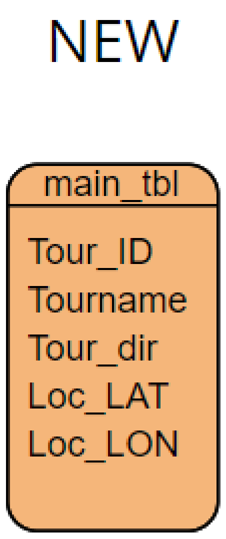
many great examples of how it can be used which helped me greatly here as it helped me understand how to use it correctly.

Once the parseFloat() function has been added to both variables it now works correctly and the user can input their latitude and longitud and get taken to their location on the map. With a cusotom image of their marker. Custom markers are done with the new google.map.Marker attribute which allows a new marker to be created we then have to give it values such as where it is postioned and for the user I have selected their maker to be at the center of the map. So that the map moves with them so instead of the user location being the marker. The center of the map is the location of the user. To get the center of the map I used a function called map.getCenter() which I found with in the google maps api doumentation and found its use here to be perfect.

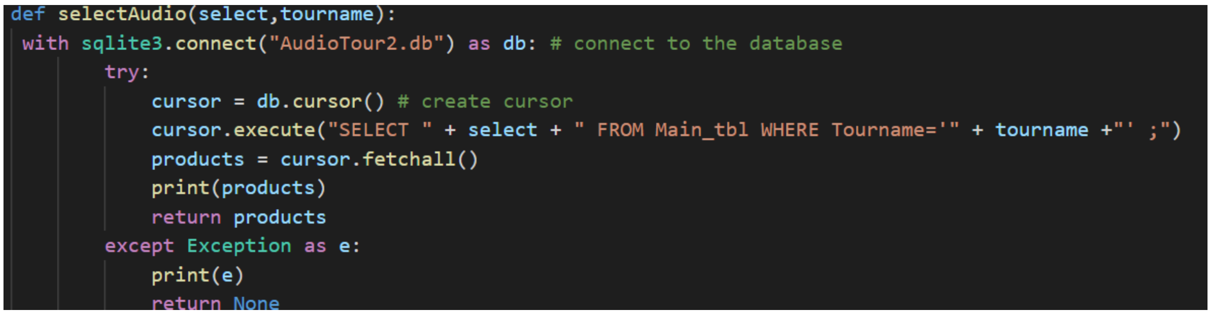
Making location of guide dynamic

To start I redesigned my database where previously it was 2 separate tables I murged Into one as I belive this would make it easier to handle. Below is an ERD of the databases and how ive modified them.





As show beside and above we can see that I have halved the number of tables and reduced the number collums due to one being unnessacery. For me this made the data easier to handle as everything was in the same table. It was not to difficault to move the data over as I could just copy and paste the data I wanted from one table to the other. From here I just have to pull data from this database. Instead of writing a new function I made an older one I used and manipluated it to be able to take more SQl statements and still return the correct data. An image of the new dynamic SQL statement is below as we can see it concanitantes strings and the peramiters of the function to create an SQL statement.



```
def selectAudio(select,tourname):
 with sqlite3.connect("AudioTour2.db") as db: # connect to the database
        try:
            cursor = db.cursor() # create cursor
            cursor.execute("SELECT " + select + " FROM Main_tbl WHERE Tourname='" + tourname +"' ;")
            products = cursor.fetchall()
            print(products)
            return products
        except Exception as e:
            print(e)
            return None
```

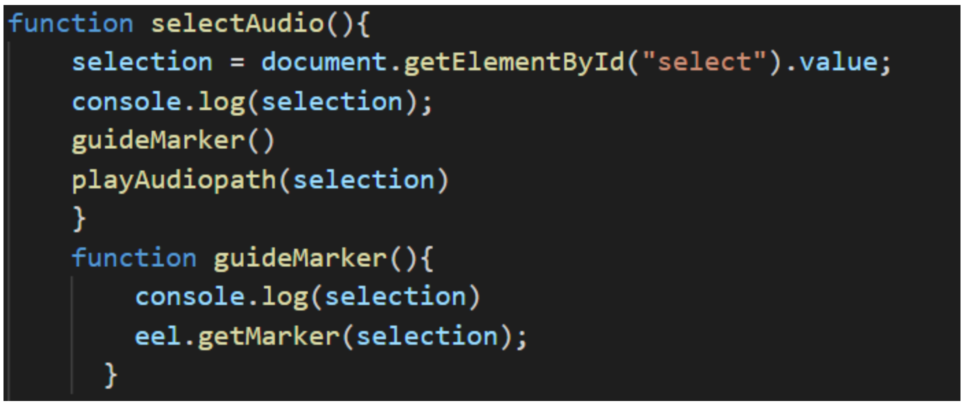
Final iteration (unit testing and merging)

Unit testing

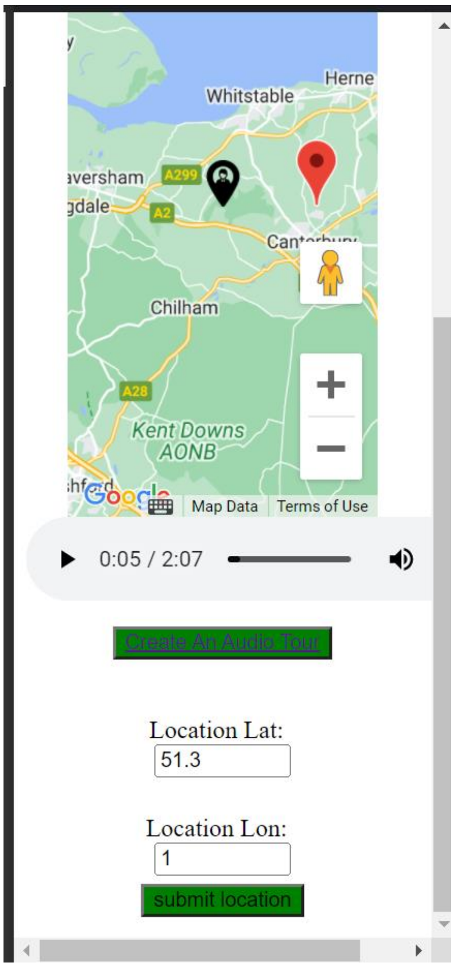
With my project coming towards its deadline ive decided its time to unit test each major part and murge it all toegther. I started via testing to see if the correct audio wold play depeding on my selection which it did. So my dynamic audio worked. Did the marker for the guide change depening on the guide I clicked. Yes, yes it does change postion depending on the section. The map also moves depening on where the user enters their lcoation. Which means all tests have passed on the index page. The next tests are on the create page where the data entred gets written to the database which it does so now we to murge them.

Murging

To murge the code I started to remove the onclick buttons and instead run them under other functions. An example of this would be me moving the playAudiopath function and the guide Marker fucntion undeneath the selectAudio function. This means it pulls the data from the auido and the guide after the user has selcted from the drop down which means I have been able to get rid of 2 buttons which were unnessasy this makes the app look cleaner and requires less user input for the same result. Below ive post an image of the murged functions.



```
function selectAudio(){
    selection = document.getElementById("select").value;
    console.log(selection);
    guideMarker()
    playAudiopath(selection)
    }
    function guideMarker(){
        console.log(selection)
        eel.getMarker(selection);
    }
```



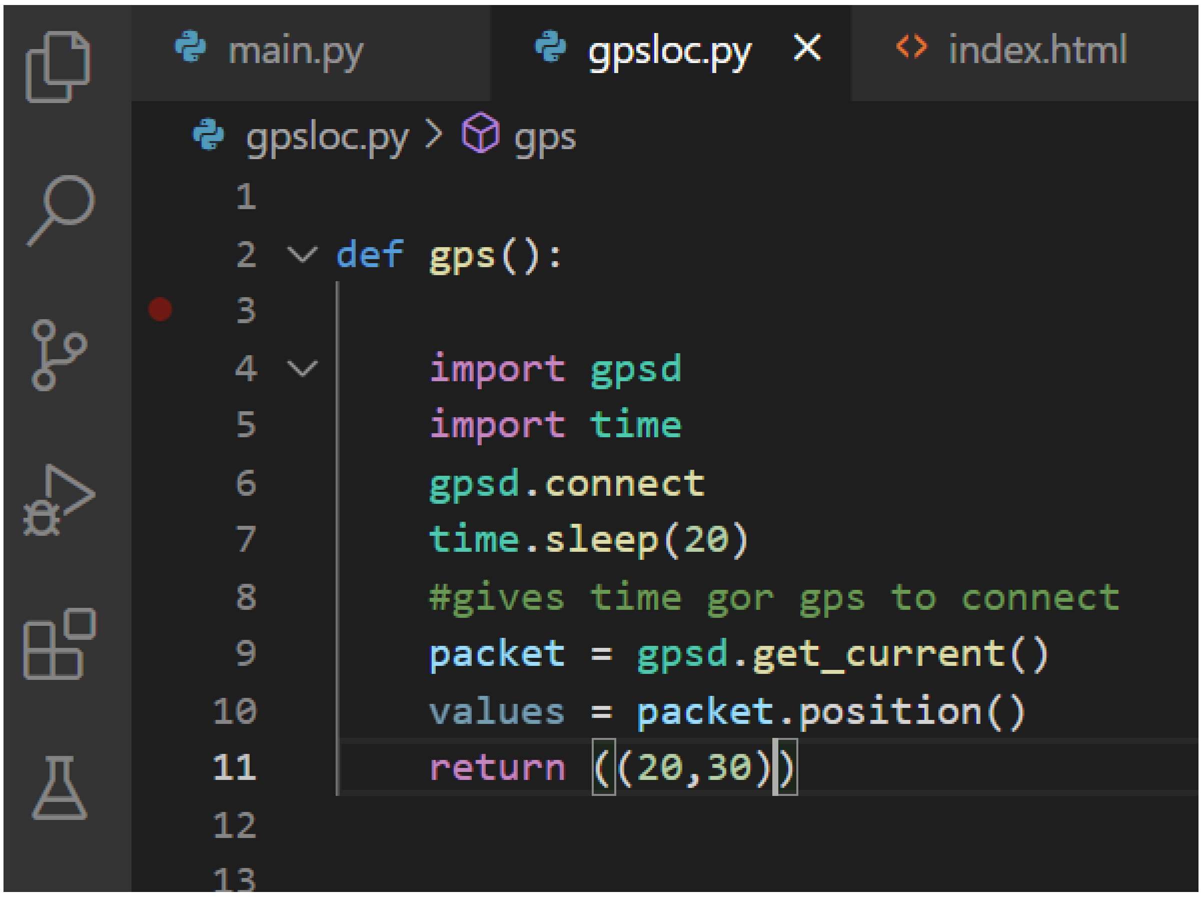
Location Lat: 51.3

Location Lon: 1

As we can see that there are now only 2 buttons, one to create an audio tour and one to supmit your location to initalise the map and start the audio. Which I think makes it a lot easier for the user as less options is more often better esspicaly for the app im devloping. I have also made it so that all the boxes are closer together and so it fits on the screen nicer. I have done this via manipulating css code to reduce the amount of space between the boses to pus then all closer togther which also makes it eaiser to screen shot.

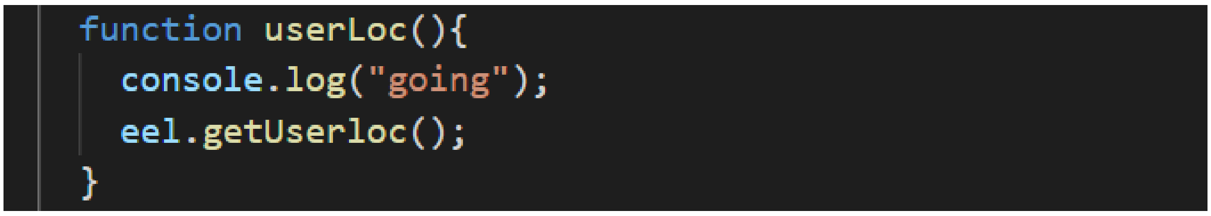
Adding gps version

I started via creating a fucntion in another file and then importating it. As you may notice it is simmilar to the orginal GPSD code I used and that it is because it worked however when testing I didn't want to wait for the gps so instead I just returned premade values.

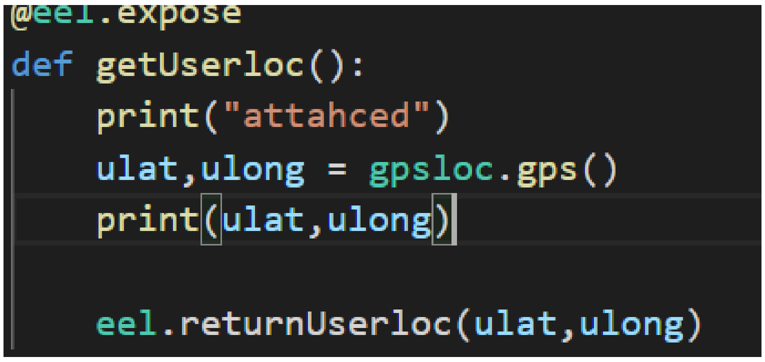

main.py gpsloc.py index.html
gpsloc.py > gps

```python

def gps():

    import gpsd
    import time
    gpsd.connect
    time.sleep(20)
    #gives time gor gps to connect
    packet = gpsd.get_current()
    values = packet.position()
    return ((20,30))


```

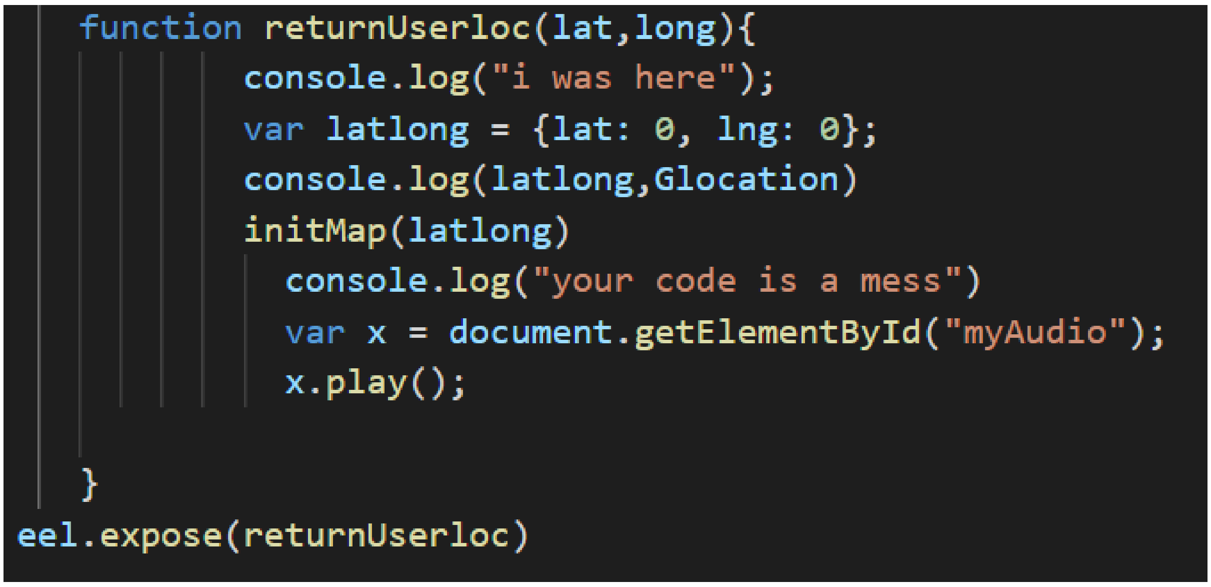
From here I then had to edit previouse functions that would pull data from intput boxes to now calling an eels function that calls gpsd.


```javascript
function userLoc(){
  console.log("going");
  eel.getUserloc();
}
```

The gps returns a tuple of floats and then gets assigned to 2 varibles to unpack the tuple and assign assigns them to separate variable and then pass them through as an argument to javascipt though eels


```python
@eel.expose
def getUserloc():
    print("attahced")
    ulat,ulong = gpsloc.gps()
    print(ulat,ulong)

    eel.returnUserloc(ulat,ulong)
```

Once passed back to this function all I had to do was expose it to eels via eel.expose and it worked



```
  function returnUserloc(lat,long){
          console.log("i was here");
          var latlong = {lat: 0, lng: 0};
          console.log(latlong,Glocation)
          initMap(latlong)
            console.log("your code is a mess")
            var x = document.getElementById("myAudio");
            x.play();

  }
eel.expose(returnUserloc)
```

Creating a read me file.

I started via looking pat how must people do it and It seems people use either txt or md files. Ive decided to go with MD as I like the formating it can give me, I used a website called makeareadme.com which helps people delvop a read me file which I delopved via looking at what I installed and set up at the start such as GPSD and what I pip installed. After that this was the result.

Audio Tour Guide

About

this is an audio tour app that uses eels, wxPython, GPSD and google maps API. the google maps api key is included so do not worry about generating your own nor do you need a GPS as there are 2 versions 1 with GPS one with out.

eel is a Python library for creating a front end for python using HTML, CSS and JavaScript. wxPython is a Python library that is a cross platform GUI toolkit.

Installation

Use the package manager pip to install wx and eels and GPSD if you're using a gps.

```
pip install eel wxPython
```

GPSD Installation and Set-up (GPS with Linux only)

```
sudo apt-get install gpsd dpsd-clients

sudo nano /etc/default/gpsd
```

then change replace the ttyXXX with the com port your computer is using

stty -F /dev/ttyXXX ispeed 4800 && cat </dev/ttyXXX

```
sudo apt install python-gi-cairo
```

then run xgps to see of your gps is working if it doesn't connect you may be out of line of site or you need to redo previous steps

```
pip3 install gpsd-py3
```

This is the python library for GPSD which allows us to pull data from the gps very easily That should be done

Usage for non GPS

```
start via selecting tour guide,
then input user location ,
via imputing latitude and longitude into input boxes
then submit location and enjoy.
```

With GPS

```
start via selecting tour guide,
then click get my location button and enjoy.
```

To make the headers you can use 1 hash like so # while to make sub headers you can use 2 hashes like so ##. Then to create the black boxes with the code they need for example pash or python you use ``` with the language then ``` to close it for example ```python (then code here)```.

3rd party Testing

These were the questions for the people who were testing my code

Which version did you test, GPS with linux or windows/linux without gps?

Were you able to view all guides deom the drop down?

were you able to select a guide from the drop down?

once you entred your location were you ablke to view the location of your self?

were you able to view the location of the guide?

Did it play the Audio of the Tour?

once you clicked the create audio tour would it take you to a new page to enter details of the auido tour?

when you clicked the file button did it open a windows that allowed you to browse your own files?

once all of your deatils were ented and clicked on the create button were you able to view your details in the database?

if no details were entred would it come up with a question asking you to enter them?

First it was tested by a student in another class who goes via the namer gigi here are his test results.

Test with classmate

Which version did you test, GPS with linux or windows/linux without gps?

windows without gps

Were you able to view all guides deom the drop down?

yes i was able to view them

were you able to select a guide from the drop down?

yes i was able to

once you entred your location were you ablke to view the location of your self?

yes i was and there was a marker to show me

were you able to view the location of the guide?

yes but i had to zoom out to see it

Did it play the Audio of the Tour?

yes it would play the audio of the tour i selected

once you clicked the create audio tour would it take you to a new page to enter details of the auido tour?

yes it did take me to a new page where i was able to enter the details of the auido tour

when you clicked the file button did it open a windows that allowed you to browse your own files?

yes a file browser did open up that let me look at my own files.

once all of your deatils were ented and clicked on the create button were you able to view your details in the database?

yes i was

if no details were entred would it come up with a question asking you to enter them?

yes it would come up with a notification telling me to enter details

at the end he stated that it would always play audio no matter what his location was.

Evaluation

What has been a success

The application can play audio, the application can display user location as well as location of tour. The application can let users upload their own content and location to the auido guide app.

What has not been a scussess.

It is a local application so all content user uploads is only acessable to them which limits the community delopment of these tours.

Changes I would make if I was remaking it.

Instead of making a localised app using eels I would make a web application which allows it to be used on any device by anyone no matter if they have a gps or what os they are running. This will also allow for user genrated content to be spread between as it would be on a MySQL database which allows the data to be read and uploaded over the web to anyone who has acess it.

Maintence and further delopment.

If I were to further develop my application I would have to push updates via geithub and emails current users to update to the most recent verison and to move their data from old to the new which I belive isnt very sustaiable. To fix this I would have to move over the data to a mySQL server to let more people interact and for it to aud automaticly move data over.

Bibliography

VoiceMap available at: https://Voicemao.me

Hearonymus available at: https://hearonymus.com

Alternativeto available at: https://Alternativeto.net

Stack Overflow, “how to access current location of any user using python”. Available at: https://stackoverflow.com/questions/24906833/how-to-access-current-location-of-any-user-using-python

IPInfo available at: https://ipinfo.io

Ipstack available at: https://ipstack.com

Towards Data Science, “Simple GPS data Visualization using Python and Open Street Maps”. Available at: https://towardsdatascience.com/simple-gps-data-visualization-using-python-and-open-street-maps-50f992e9b676

GPSD github https://github.com/MartijnBraam/gpsd-py3

Pythonschools.net http://pythonschool.net/databases/inserting-data/

Dfrobot.com https://wiki.dfrobot.com/USB_GPS_Receiver_SKU_TEL0137

Geeks for geeks eel https://www.geeksforgeeks.org/create-html-user-interface-using-eel-in-python/

Screen size https://github.com/ChrisKnott/Eel/issues/238

Stack overflow Nootaku. avalible here:

https://stackoverflow.com/users/10964544/nootaku

Craig’n’dave Normalisation to N3. avalible here:

https://www.youtube.com/watch?v=UY3zc9G_YZo

Normalisation in DBMS by Richard peterson, 2021. avalible here:

https://www.guru99.com/database-normalization.html#4

Code father tuple to string. avalible here:

https://codefather.tech/blog/tuple-to-string-python/

Stack over flow drop down. Avalible here:

https://stackoverflow.com/questions/9895082/javascript-populate-drop-down-list-with-array

Google maps api. Availble here:

https://developers.google.com/maps/documentation/javascript/overview#maps_map_simple-javascript

Adding markers. Availble here:

https://developers.google.com/maps/documentation/javascript/adding-a-google-map

w3schools mp3. Avalible here:

https://www.w3schools.com/html/html5_audio.asp

w3schools onChange. Avalible here:

https://www.w3schools.com/jsref/event_onchange.asp

Google maps api marker size

https://stackoverflow.com/questions/15096461/resize-google-maps-marker-icon-image

stack overflow invalid value error google maps api. avalible here:

https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_parsefloat

string to float

https://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_parsefloat

makeareadme

https://www.makeareadme.com/

Appendix / Code dump

Main.py

```
#all imports
import eel,sqlite3
import wx
import gpsloc
e = "fail"
eel.init("web")

# Start the index.html file
eel.start("index.html", size=(300, 650), position=(0,0), block=False)



#SQL ststment to insert data to database
def insertMain(values): # create function to insert data
    with sqlite3.connect("AudioTour2.db") as db: # connect to the database
        print ("connected")
        try:
            cursor = db.cursor() # create cursor
            sql = "Insert into Main_tbl (Tourname, Audio_dir, Loc_LAT,
Loc_LON) values (?,?,?,?)" # create SQL statement, using ? mark to paramatise
this query, so you can pass any value into this function
            cursor.execute(sql,values)# execute the SQL statement and pass
values
            db.commit() # save changes
        except Exception as e:
            print(e)
#eel function to pass though
@eel.expose
def insertm(msg):
    print(msg)
    print("inseting main")
    insertMain(msg)
    return "ok"

#get names for drop down
def selectNames():
 with sqlite3.connect("AudioTour2.db") as db: # connect to the database
        try:
            cursor = db.cursor() # create cursor
            cursor.execute("SELECT Tourname FROM Main_tbl;")
            products = cursor.fetchall()
            print(products)
            return products
        except Exception as e:
            print(e)
            return None
#Dynamic SQL statment for gathering data from table
def selectAudio(select,tourname):
 with sqlite3.connect("AudioTour2.db") as db: # connect to the database
        try:
            cursor = db.cursor() # create cursor
            cursor.execute("SELECT " + select + " FROM Main_tbl WHERE 
Tourname='" + tourname +"' ;")
            products = cursor.fetchall()
            print(products)
            return products
        except Exception as e:
            print(e)
            return None

#tuple unpacking for dropdown
def unpackTuple(products1d,item):
    """based on 
https://wwww.geeksforgeeks.org/unpacking-a-tuple-in-python/
and 
https://thispointer.com/python-how-to-unpack-list-tuple-or-dictionary-to-
function-arguments-using/
and 
https://www.javaexersise.com/python/python-convert-tuple-to-string.php
"""
    item1 = "".join(item)
    products1d.append(item1)
    return products1d
#unapcking data for dropdown
@eel.expose
def getProduct(msg):
    print(msg)
    products=selectNames()
    products1d=[]
    for i in range (0,len(products)):
        products1d = unpackTuple(products1d,products[i])
    eel.returnNames(products)

#function to return souce/path for audio tour
@eel.expose
def getTourAudio(msg):
    print(msg)
    eel.getTourPath(selectAudio("Audio_dir ",msg))
#function for uploading path
@eel.expose
def pythonFunction(wildcard="*"):
    app = wx.App(None)
    style = wx.FD_OPEN | wx.FD_FILE_MUST_EXIST
    dialog = wx.FileDialog(None, 'Open', wildcard=wildcard, style=style)
    if dialog.ShowModal() == wx.ID_OK:
        path = dialog.GetPath()
    else:
        path = None
    dialog.Destroy()
    print(path)
    return path
#function for returning guide marker location to javascipt though eels.
@eel.expose
def getMarker(msg):
    print(msg)
    longlat=selectAudio("Loc_LAT,Loc_LON", msg)
    longlat =longlat[0]
    lat,long = longlat
    eel.returnMarker(lat,long)

    eel.returnUserloc(ulat,ulong)

while True:
    eel.sleep(2.0)
```

Python for gps to eels within main.py

```
#function that gets user location data though gps
@eel.expose
def getUserloc():
    print("attahced")
    ulat,ulong = gpsloc.gps()
    print(ulat,)
```

gpsloc.py

```
def gps():

    import gpsd
    import time
    gpsd.connect
    time.sleep(20)
    #gives time gor gps to connect
    packet = gpsd.get_current()
    values = packet.position()
    return (values)
```

Scripts.js

```javascript
window.onload = requestNames(); /*calls function to create dropdown*/

//assigns global variables
var globalProducts = new Array();
var file = "";
var selection = "";
//fucntion for when user selects audio which gethers audio path as guide
location
function selectAudio(){
    selection = document.getElementById("select").value;
    console.log(selection);
    guideMarker()
    playAudiopath(selection)
    }
    //function that gets guide marker
    function guideMarker(){
        console.log(selection)
        eel.getMarker(selection);
      }

function requestNames(){
    eel.getProduct("hello");

    //calls eels
}

//retuns name for dropdown from python
function returnNames(products){
    console.log(products);
    globalProducts = products;

    dropdown();

}
eel.expose(returnNames)
//creates the dropdown table using array called globalProducts
function dropdown(){
    var select = document.getElementById("select");
    console.log(globalProducts);
    console.log("ok");
    for(var i = 0; i < globalProducts.length; i++) {
        var opt = globalProducts[i];
        var el = document.createElement("option");
        el.textContent = opt;
        el.value = opt;
        select.appendChild(el);
    }
}





//sends dat over from create page to python to be put into an SQL database via
eels.
function sendMain(file){
    var Gname = document.getElementById('Gname').value;
    var AudioDir = file;
    var LocLAT = document.getElementById('LocLAT').value;
    var LocLON = document.getElementById('LocLON').value;
    if (Gname.length <1)
    {window.alert("there must be a value Gname and selected a file");
    }
    else if (LocLAT.length <1 || LocLON.length <1)
    {window.alert("there must be a value in both Lat and Long");
    }
    else{   const values = new Array(Gname,AudioDir,LocLAT,LocLON);
        eel.insertm(values)(processAcknowledgement)}

}


//fucntion for making sure we have the file and writing it to screen for user
to see.
function processAcknowledgement(r){
    console.log(r);
    file  =  document.getElementById("fileSelected").innerHTML=r;
    console.log(file)

}


function getPathToFile() {
    eel.pythonFunction()(processAcknowledgement)
}

//writes audio path to html
function getTourPath(path){
    console.log(path);
    //correct path to audio
    //based on w3schools
    var x = document.getElementById("myAudio");
      x.innerHTML = "<source src='" + path + "'type='audio/mpeg'>";
```

```
}
eel.expose(getTourPath)
//starts funtion to gather audio path data
function playAudiopath(){
    console.log(selection)
    eel.getTourAudio(selection);
}
```

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Eel Example</title>
    <link rel="stylesheet" href="style.css">
    <script type="text/javascript" src="eel.js"></script>
    <script src="./Script.js"></script>


  </head>
  <body>
    <h1 class="header">Audio Tour</h1>
  <select id="select" onchange="selectAudio()">
      <option>Select Audio tour</option>
  </select>
  <h1>Guide</h1>
  <div id="map"></div>
  <div id="audioguide">
  <audio id="myAudio"controls>
    <!-- place holder -->
  </audio>
    <button ><a href="create.html">Create An Audio Tour</a></button>
    <!--used to check array results-->
  <form>
    <label for="LocLON">Location Lat:</label>
    <br>
    <input type="number" step="0.000001" id="locLAT" name="LocLAT">
    <br>
    <label for="LocLON">Location Lon:</label>
    <br>
    <input type="number" step="0.000001" id="locLON" name="LocLON">
    <button id="userloc" type="button" onclick="getuserloc()">submit
location</button>

  </form>

  </body>
  <script>
   var marker;
   var Glocation;
   var user;
   var arraylong = new Array;
    function guideMarker(){
      console.log(selection)
      eel.getMarker(selection);
    }

    function returnMarker(lat,long){
    console.log(lat,long)

    Glocation = { lat: lat, lng: long};
    }
    eel.expose(returnMarker)

    function getuserloc(){
            console.log("i was here");
            var latlong = {lat:
parseFloat(document.getElementById('locLAT').value), lng:
parseFloat(document.getElementById('locLON').value)};
            console.log(latlong,Glocation)
            initMap(latlong)
              console.log("your code is a mess")
              var x = document.getElementById("myAudio");
              x.play();

    }

    function initMap(loc) {
        var options = {
            zoom: 16,
            center: loc,
        }

    // new map
    var map = new
    google.maps.Map(document.getElementById('map'), options);
    //add marker

    var image = {
    url: "https://cdn-icons-png.flaticon.com/512/70/70770.png",
    scaledSize: new google.maps.Size (30,30),
  }

  user = new google.maps.Marker({
      position: Glocation,
      map:map,
      title: "audio guide location",
  });
  marker = new google.maps.Marker({
            position: map.getCenter(),
            map:map,
            icon: image,
            title: "You",
          });


  }
</script>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyD7ZDczMEJbu4P2300eRFxHg
OFE23qidmQ&callback=initMap&v=weekly"
async>
</script>

</html>
```

Index.html for gps

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Eel Example</title>
    <link rel="stylesheet" href="style.css">
    <script type="text/javascript" src="eel.js"></script>
    <script src="./Script.js"></script>


  </head>
  <body>
    <h1 class="header">Audio Tour</h1>
  <select id="select" onchange="selectAudio()">
      <option>Select Audio tour</option>
  </select>
  <h1>Guide</h1>
  <div id="map"></div>
  <div id="audioguide">
  <audio id="myAudio"controls>
    <!-- place holder -->
  </audio>
    <button ><a href="create.html">Create An Audio Tour</a></button>
    <!--used to check array results-->
    <button id="userloc" type="button" onclick="userL1oc()">submit
location</button>

  </body>
  <script>
   var marker;
   var Glocation;
   var user;
   var arraylong = new Array;
    function guideMarker(){
      console.log(selection)
      eel.getMarker(selection);
    }

    function returnMarker(lat,long){
    console.log(lat,long)

    Glocation = { lat: lat, lng: long};
    }
    eel.expose(returnMarker)

    function userLoc(){
      console.log("going");
      eel.getUserloc();
    }
    function returnUserloc(lat,long){
            console.log("i was here");
            var latlong = {lat: 0, lng: 0};
            console.log(latlong,Glocation)
            initMap(latlong)
              console.log("your code is a mess")
              var x = document.getElementById("myAudio");
              x.play();

    }
 eel.expose(returnUserloc)

    function initMap(loc) {
        var options = {
            zoom: 16,
            center: loc,
        }

    // new map
    var map = new
    google.maps.Map(document.getElementById('map'), options);
    //add marker

    var image = {
    url: "https://cdn-icons-png.flaticon.com/512/70/70770.png",
    scaledSize: new google.maps.Size (30,30),
  }
```

```
  user = new google.maps.Marker({
      position: Glocation,
      map:map,
      title: "audio guide location",
  });


  marker = new google.maps.Marker({
            position: map.getCenter(),
            map:map,
            icon: image,
            title: "You",
          });


  }
</script>
<script
src="https://maps.googleapis.com/maps/api/js?key=AIzaSyD7ZDczMEJbu4P2300eRFxHg
OFE23qidmQ&callback=initMap&v=weekly"
async>
</script>

</html>
```

Style.css for both versions

```
h1{
  color: green;
  text-align: center;
}


.head{
  display: inline-block;

  border: solid 2px red;
}
button{
  width: auto;
  display: block;
  background-color: green ;
  margin: 5% auto;
}

.header{
  background-color: #728639 ;
  color: black;
  margin: auto;
  width: 40%;
  border: 3px solid black;
  padding: 10px;
}

#back{
  float:left;
  margin-left: 10%;
  margin-top: 0px;
  margin-bottom: 0px;
  margin-right: 0px;

}
.header h1{
  color: black;
  text-align:center ;
  margin-left: 50%;
  margin-top: 0px;
  float:left
}

.audioselect{
  width: auto;
  display: block;
  background-color: green ;
  margin: 20% auto;
}
.create{

  margin-bottom: 30%;
}

body{
  text-align: center;
}
form{
  display: inline-block;

  text-align: center;
}
.createform{
  margin-top: 20%;
}
input {
  display: inline-block;
  width: 6em;
  top: -3em;
}

label {
  display: inline-block;
  margin-right: .5em;
  padding-top: 1.5em;
}


#map{
    margin: auto;
    height: 400px;
    width: 200px;
}
```