

## **Structure of my website files and folders**

Starting off with the structure of my website files, there is an index file which contains a header and links to the other pages. The other webpages, such as “login”, “Events”, “About”, “signedin”, “signedup” and “signup”, are contained within the “pages” folder. These files allow the user to do certain actions, such as view the events available and create and view their own account. I have used PHP and HTML to create these webpages. JS was also built into the pages itself, therefore, it does not have any unique files. There are pages with unique folders that contain certain PHP files which allows for the user to redirect themselves towards certain pages. The other folder contains a mix of JSON, PHP and CSS. The purpose of this folder is for the functionality of the page folders. Every function that is needed for the page is stored within these files. We also have CSS files which allow us to decorate the websites and make it easy to navigate. When you enter the website, you can access other pages, such as the log in page or event page, by using the header file.

## **Summary of the website**

From the structure of the websites, we can go in depth what each page does. Starting with the pages folder. Before beginning, we have a header function in all the webpages. The header function allows the user to redirect to certain pages. In the header function presents 4 buttons if they are not logged in, “Home”, “First”, “signup” page allows the user to create an account in the system, so they can book events and like them. While signing up there is some validations checks such as checking if it is an Aston email, checking if the password is 6 characters long, checking if the phone number is valid and checking if the email has the right length for string. After signing up they are needed to “login” page which allows them to login into the system. The login page is a validation system check if the user checks for who they are. After successfully logging in the website they can use the events page. The Events page allows the user to filter on what type of event they want to attend. There is Art, Sports and history pages. These pages all have html and php code and they are similar. They have two buttons which are like button and book appointment. The book appointment allows the user to book an appointment with the event, the like button allows the user to show interest within the event. In my other folder we have “AllEvents”, “dbconnect”, “Eventfunctions”, “logout”, “register”, “signedout”, and “myfunctions”. Starting with the dbconnect this allows user to connect to the database with the configuration of hostname, username, password, and the database name to phpmyadmin. Eventfunctions relate towards the AllEvents. Event functions has the functions which relate to the events such as liking events, checking if the user liked a certain event, check if a user certain signed up and booking towards appointment for the event. The register file allows takes in the information from the signup page and checks if the information is valid. The register file uses validation functions from the myfunctions file. The myfunction file has functions which help indicate

whether the user has inputted the correct information or not. It has certain functions such as "PhoneNumberExists" which checks if the phone number already exists in the database, "CollegeIDExists" checks if the college id exists in the database, "CollegeIDInvalid", this checks if collegeid has the right amount of numbers and checks if what has been inputted is numbers.

"CreateAccount" is a function which is used when creating an account and it's used in the register file. CreateAccount function takes the validated inputs and runs them through an sql statement which inserts them into the database.

"LoginAccount" is used for logging the account and starting a session. When loginaccount is valid it allows the user to book appointments without loginaccount been valid user can't like an event or book an appointment for the event.

### **Database Structure and Functionality**

My Database structure I have one database which has 4 tables.

"AstonEvents", "EventsJoined", "EventsLiked", "StudentUsers" are tables which are in the database and used throughout the website. AstonEvents is the table which has information for each event such as name, description, time and date ectr. "EventsJoined" is the table which has information on who has booked an appointment on for a certain event. It uses University ID to identity the person and Event ID to identify which event it is. StudentUsers is table which contains all the information on the users who registered.

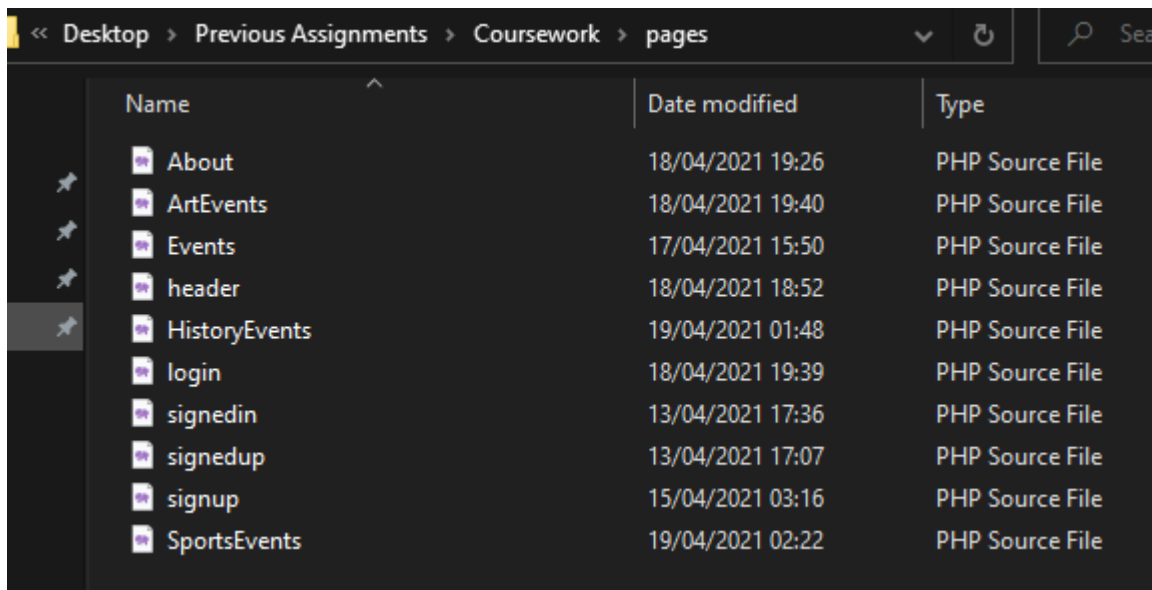
StudentUsers takes in University ID, Email, Password and Phone Number.

Each column has a purpose throughout the database for example university id is used in eventsliked and eventsjoined tables to identity if the person, email and userpassword are used to login the website. To make this database I have used phpmyadmin to create the whole tables and it allowed me to create relations between the tables. The foreign keys "CollegeID" that are relations between the tables. All tables that are in relation of finding the EventID and UniversityId to find if something exists. The main functionality of using tables in my website is to allow to save information about the users liking and what they have booked and it's also check if certain information exists; if it doesn't the user is allowed to create it for example creating an account or liking an event.

### **Cons of the website**

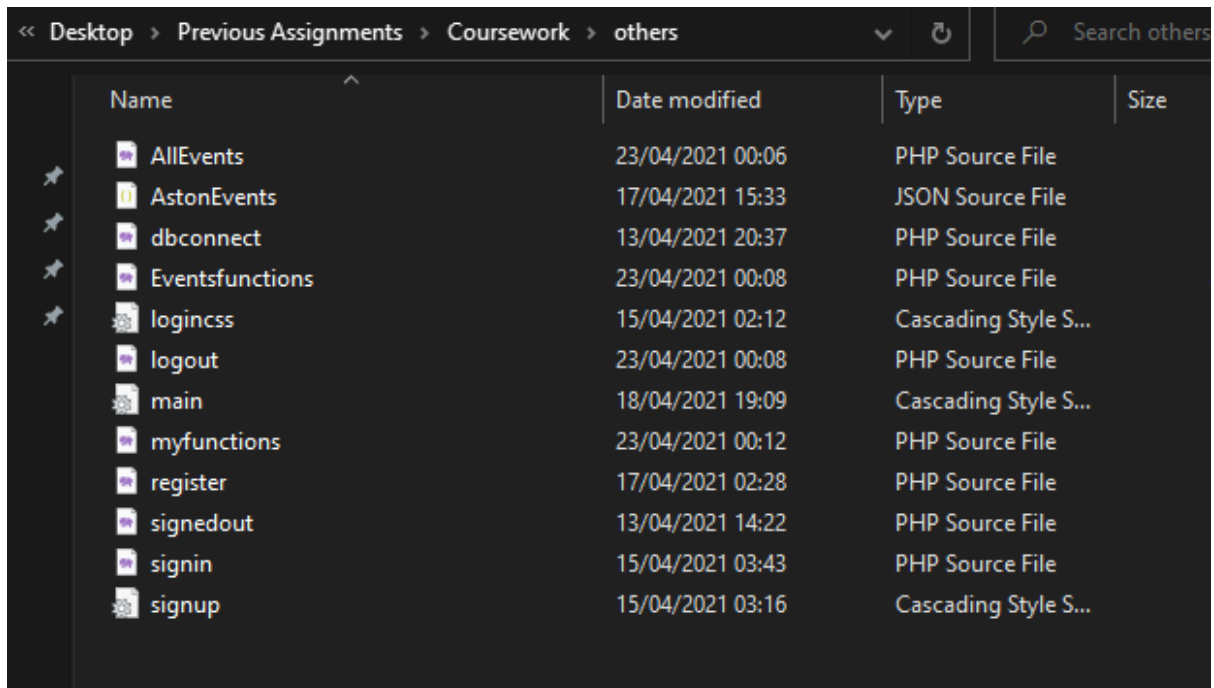
When making the websites there was a few issues that i had to get across. First issue was the directory issue. When using header file it created issues because when redirecting to a different folder it can't redirect to other files which are attached to the header. The way I fixed this issue is by putting all the files in the same folder. Second issue I had while making the website was making the website compatible with mobile but I have fixed that issue due to the learning through bootstrap and using the meta tag. However, when using the meta tag its not 100% fully compatible with mobile as there is some issues with it such as events are not properly sized.

### File purpose



Name	Date modified	Type
About	18/04/2021 19:26	PHP Source File
ArtEvents	18/04/2021 19:40	PHP Source File
Events	17/04/2021 15:50	PHP Source File
header	18/04/2021 18:52	PHP Source File
HistoryEvents	19/04/2021 01:48	PHP Source File
login	18/04/2021 19:39	PHP Source File
signedin	13/04/2021 17:36	PHP Source File
signedup	13/04/2021 17:07	PHP Source File
signup	15/04/2021 03:16	PHP Source File
SportsEvents	19/04/2021 02:22	PHP Source File

For these files it allows them to present themselves when the user is using the website. Events page will lead the user to certain events such as art event, history event ect.



Name	Date modified	Type	Size
AllEvents	23/04/2021 00:06	PHP Source File	
AstonEvents	17/04/2021 15:33	JSON Source File	
dbconnect	13/04/2021 20:37	PHP Source File	
Eventsfunctions	23/04/2021 00:08	PHP Source File	
logincss	15/04/2021 02:12	Cascading Style S...	
logout	23/04/2021 00:08	PHP Source File	
main	18/04/2021 19:09	Cascading Style S...	
myfunctions	23/04/2021 00:12	PHP Source File	
register	17/04/2021 02:28	PHP Source File	
signedout	13/04/2021 14:22	PHP Source File	
signin	15/04/2021 03:43	PHP Source File	
signup	15/04/2021 03:16	Cascading Style S...	

In the others folder we have our functionality such as login, register and dbconnect.

Dbconnect allows php to connect to the database using mysqli.

AllEvents file holds the functionality for the events page checking if the user liked the page or has already registered for an event and to like and book for event.

Myfunctions file has the functionality of the register and it checks if the user has right validity.

Register file gets the inputs and runs the functions to see if they are correct.

The css files decorate the page correctly.

AbdulRahman Alian

Sighin file is the login file which gets the inputs and validates them to check if any inputs has been inputted.