Lara Elkady

Justin Hart

Midori Lynch

Talent Agency

12-09-2021

Database Project Phase III

Source Code: https://github.com/mlynch018/DBProject/

Installation Guide on ReadME.md on Github.

Section 1: Problem Statement

Context

Talent Agencies are companies which provide representation for their clients. Depending on the specific agency these clients could be models, actors or other creative professionals. A large part of how talent agencies market their clients for potential work opportunities is through displaying them on company websites.

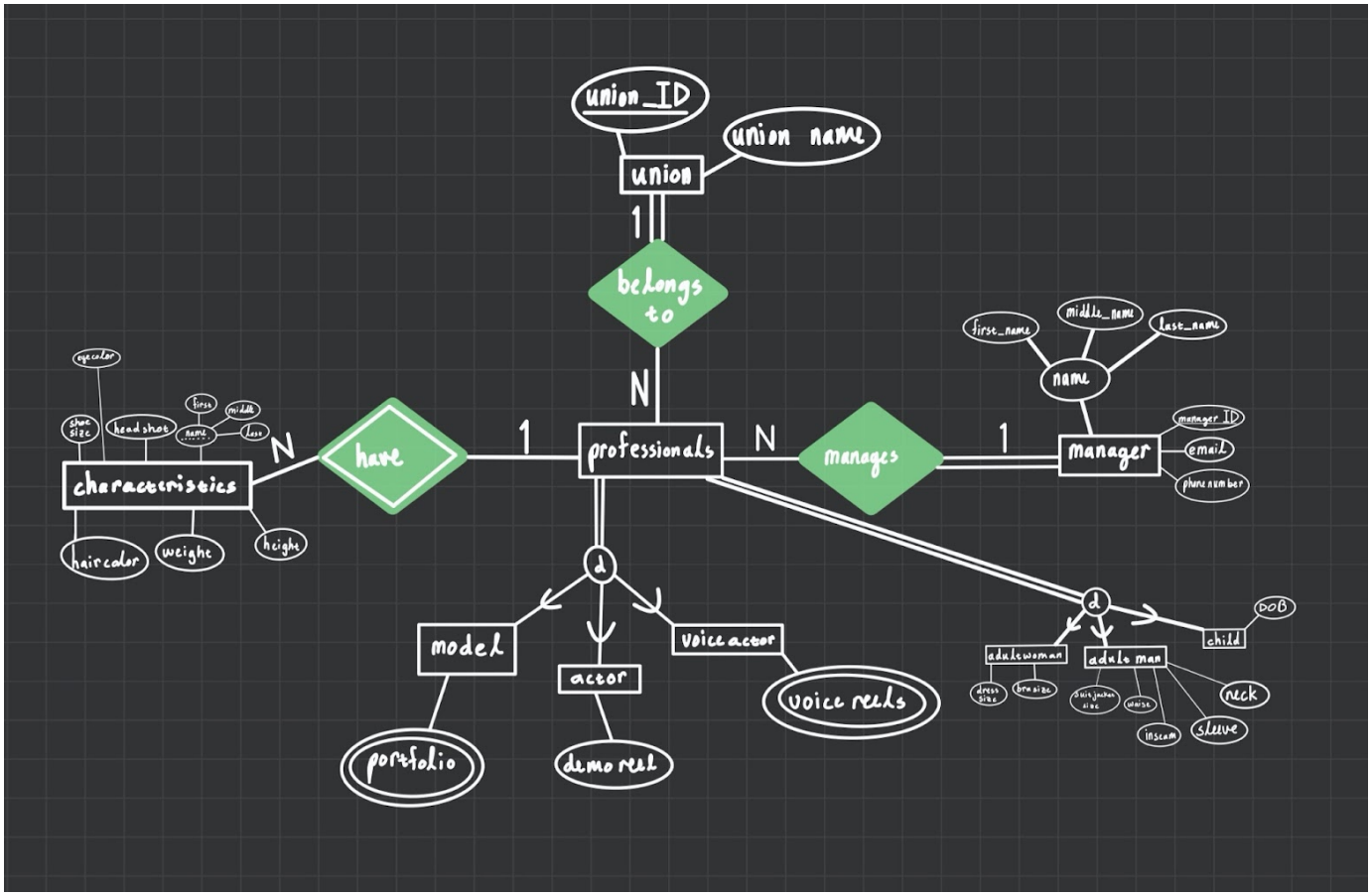
Oftentimes however, these company websites do not fully showcase their clients. In today's competitive talent market, professionals need to showcase their skills through acting reels, vocal demos, modeling portfolios and other supplemental materials. In addition to that, these materials need to be able to be quickly accessed by casting directors. However, talent agency sites usually only have resumes and headshots of their clients. Of course, one can only gauge so much about a professional based on a resume and headshot. This poses a large problem for casting directors who wish to curate a group of actors or models who possess specific skills

or appearances and may have a limited budget of clients to invite to audition for their productions.

Intention

The main goal of this project is to create a web-based database which showcases our talent agency's clients. Because the market for talent is beginning to become more competitive, we not only want to provide resumes and headshots of our clients but also supplemental materials which further market our clients. For actors this means providing a demo reel of previous works or for a model to provide a portfolio of photos. In addition to this we want to not only make the process of finding professionals quick and accessible, we want to make getting into contact with our talent managers an easy process.

Section 2: Conceptual Database Design



As we have described and shown in the ER diagram, we have 3 main types of professionals. The 3 main types of professionals are models, actors and voice actors. Each professional is either an adult woman, adult man or a child. Children are all in a section of their own and not divided by gender. All professionals will have characteristics, belong to a union and have a manager.

One manager can manage multiple professionals, one union can have multiple professionals in it (multiple professionals can belong to the same union) and each professional has one set of characteristics.

Functional Requirements

Our database will need to distinguish between all of our entity types and attributes.

We need to be able to have a distinction between adult males, adult females, and children, with children showing their D.o.B for legal reasons.

We will have to have search parameters to differentiate between all of the professionals by desired attributes.

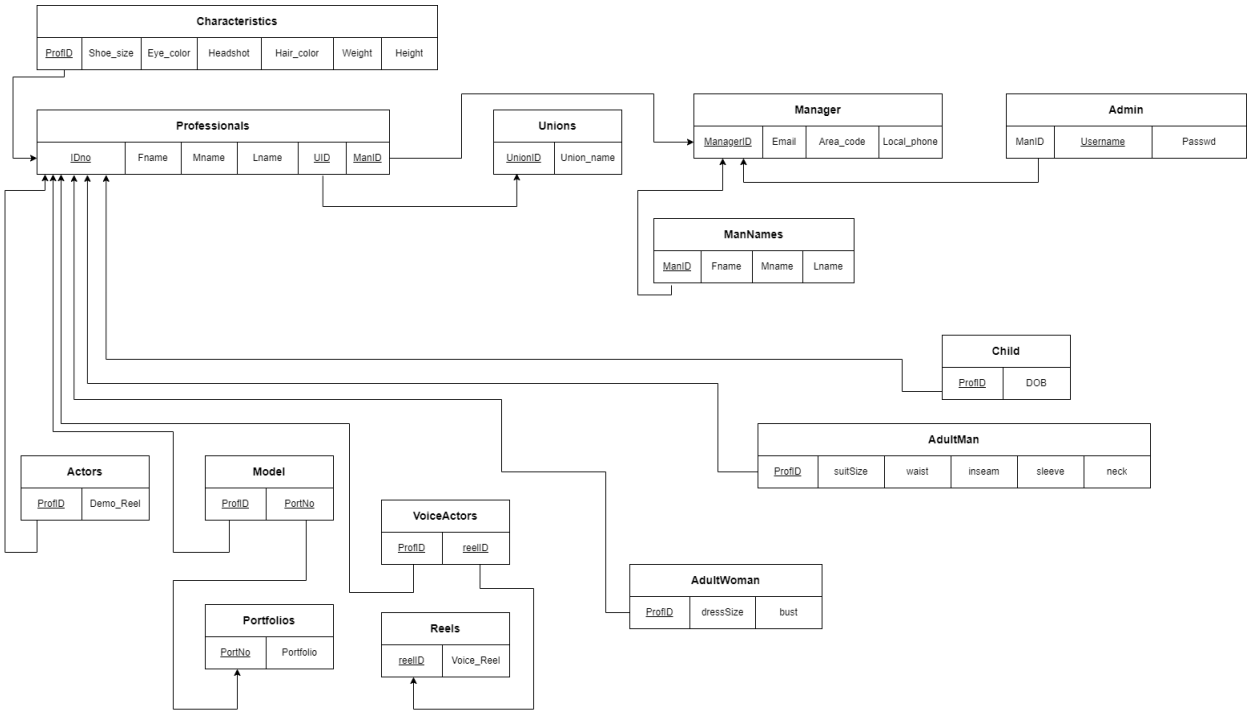
Have attached to each professional a manager email and union name with whom agencies can get in contact with to sort out arrangements.

We need the functionality of adding and removing professionals from the website, as well as editing their page/profile

We have a database stats tab as well that uses aggregate functions to display total number of professions/managers/etc contained in the database.

Section 3: Logical Database Design

Relational Model



Summary Table of Data Types (1 of 3)

Table	Attribute	Type	Constraint
Admin	Username	CHAR(80)	Primary Key
Admin	Passwd	CHAR(80)	NOT NULL
Professionals	IDno	CHAR(80)	Primary Key
Professionals	Fname	CHAR(80)	NOT NULL
Professionals	Mname	CHAR(80)	
Professionals	Lname	CHAR(80)	NOT NULL

Professionals	UID	CHAR(80)	Foriegn Key

Summary Table of Data Types (2 of 3)

Professionals	ManID	CHAR(80)	Foriegn Key
Characteristics	ProfID	CHAR(80)	Foriegn Key
Characteristics	Shoe_size	DOUBLE(3,1)	
Characteristics	Eye_color	CHAR(80)	
Characteristics	Headshot	CHAR(100)	
Characteristics	Hair_color	CHAR(80)	
Characteristics	Weight	INTEGER	
Characteristics	Height	INTEGER[1]	
Unions	UnionID	CHAR(80)	Primary key
Unions	Union_name	CHAR(80)	NOT NULL
Manager	MangerID	CHAR(80)	Primary Key
Manager	Email	CHAR(80)	
Manager	Area_code	INTEGER	
Manager	Local_phone	INTEGER	
ManNames	ManID	CHAR(80)	FORIEGN KEY
ManNames	Fname	CHAR(80)	NOT NULL
ManNames	Mname	CHAR(80)	
ManNames	Lname	CHAR(80)	NOT NULL
Actors	ProfID	CHAR(80)	Foreign Key
Actors	Demo_Reel	CHAR(100)[2]	

[1] Height in inches
[2] Hyperlinks and images were set to varchar(100) due to difficulties with file uploads and linking.

Model	ProfID	CHAR(80)	Unique
Model	PortNo	CHAR(80)	Foriegn Key
Portfolios	PortNo	CHAR(80)	Foriegn Key

Summary Table of Data Types (3 of 3)

Portfolios	Portfolio	CHAR(100)	
VoiceActors	ProfID	CHAR(80)	Foriegn Key
VoiceActors	reelID	CHAR(80)	Foriegn Key
Reels	reelID	CHAR(80)	Primary Key
Reels	Voice_Reels	CHAR(100)	
AdultWoman	ProfID	CHAR(80)	Foriegn Key
AdultWoman	dressSize	INTEGER	
AdultWoman	bust	INTEGER	
AdultMan	ProfID	CHAR(80)	Foreign Key
AdultMan	suitSize	CHAR(80)[3]	
AdultMan	waist	INTEGER	
AdultMan	inseam	INTEGER	
AdultMan	sleeve	INTEGER	
AdultMan	neck	INTEGER	
Child	ProfID	CHAR(80)	Foreign Key
Child	DOB	DATE	NOT NULL
Admin	ManID[4]	CHAR(80)	Foreign Key

[3] Has a letter associated with it. Example: 38R

[4]

Section 4: Application Program Design

SQL Pseudo Code:

//Below is our pseudocode that lays out our general format of what we want our database to contain

Create Manager Table, Prim. Key = ManagerID of type varchar

    Contains Email, zipcode, phone attributes

Create Admin Table, Prim. Key = Admin ID of type varchar

    Contains user, pass

Create ManagersNames Table, Foreign.key = Man ID, of type varchar that refs Manager Table

    Contains Fname, Mname, Lname,

Create Unions Table, Prim. Key = Union ID, of type varchar

    Contains Union_name

Create Professionals Table, Prim. Key = ID of type var char

    Contains, Fname, Mname, Lname, UID and Manager ID as foreign keys,

Create Characteristics Table, Foreign key = ID referencing professionals

    Contains Shoe Size, Eye_color, headshot, hair_color, weight, height

Create Child Table Foreign Key ID

    Contains Date of Birth

Create AdultMan Table Foreign Key ID

    Contains suitSize, waist, inseam, sleeve, neck

Create Actors Table Foreign Key ID

    Contains Demo_reel

Create Portfolios Table Primary key Port ID

Contains portfolio

Create Reels Table Prim. Key ReelID

Contains Voice_reels

Create Model Table Foreign Key ProfID

Contains PortNo Foreign key

Create VoiceActors Table Foreign Key ProfID

Contains ReelID Foreign Key

Insert Temp Values to fill Data table

Website Pseudocode:

To get our website working, we used php in coordination with our SQL Database to display our database. Brief descriptions provided about each file's intended purpose.

Actors.php

Select SQL Query to get actors information

Display actors info

Add.php

Will Add a professional to the database from user input

The code for the delete professional button will also be here

Addmang.php

Will add a manager to the database from user input

Adultman.php

Select SQL Query to get man information

Displays Man Details

Adultwoman.php

Select SQL Query to get adult woman information

Displays Woman Details

Details.php

Select SQL Query to get professionals information

Display details when more information is clicked

Index.php

Main file for the program, handles the displaying of the main home page

Kid.php

Select SQL Query to get Kids information

Display Kids information

Manager.php

Select SQL Query to get Managers information

Display Managers information

Contains a button to send you to a form and add managers

Model.php

Select SQL Query to get Model information

Display Model information

Stats.php - Aggregate Functionality

Select SQL Query and Aggregate functions to get database Stats

Display Database statistics

Update.php

Takes user input to update an existing entry's characteristics in the database

Voices.php

Select SQL Query to get Voice actors information

Displays Voice actors' information

Section 5: User Interface Design and User Manual

Installation Manual:

` Operating System: Windows 10

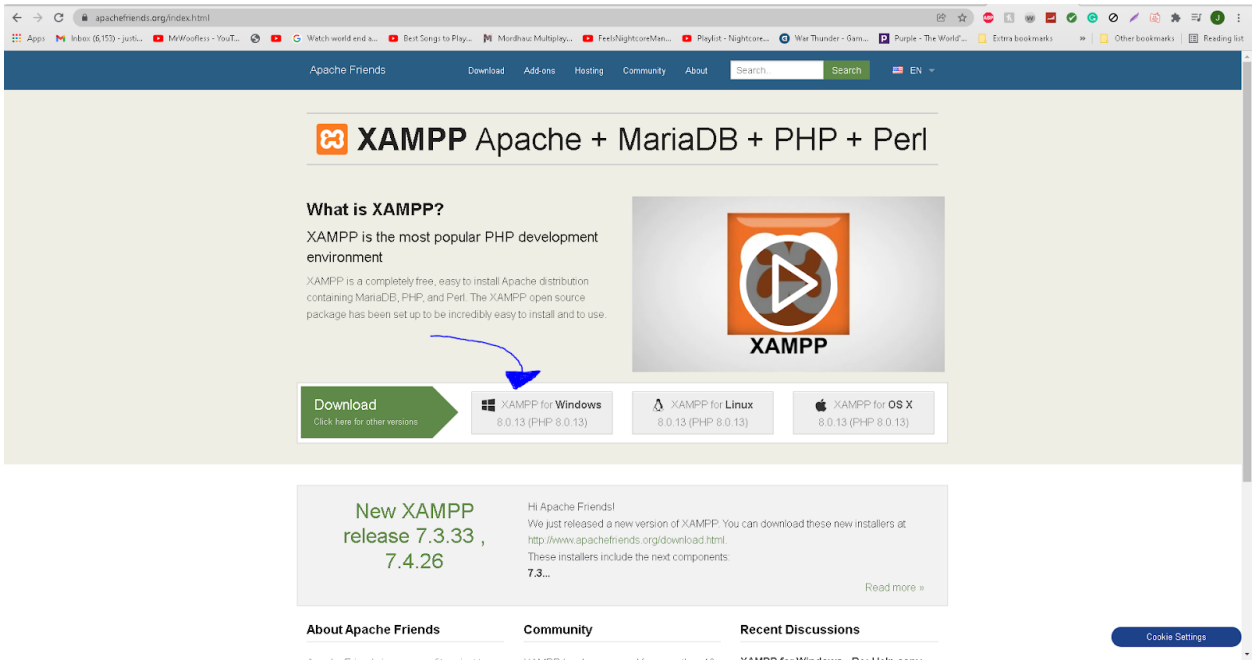
Search Engine: Google Chrome (Note: Microsoft Edge for the most part worked as well but we had issues downloading from git-hub while using it))

In order to run this website, you will need some hosting software as well as mysql. We found it easiest to use XAMPP to connect these components.

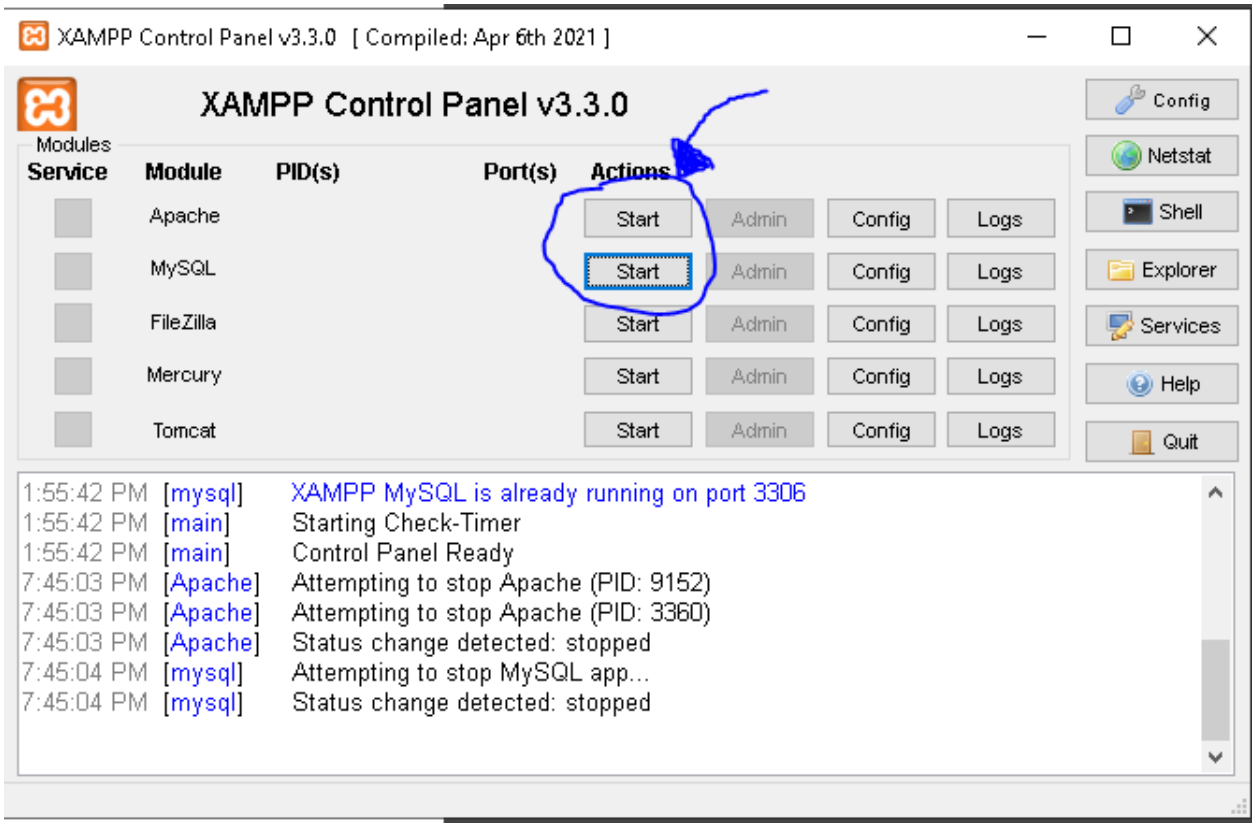
1) We first had to uninstall our existing MySQL workbenches (if you have any downloaded) in order to get this to work. You can download XAMPP using the following link: https://www.apachefriends.org/index.html.

   Just next through the options until you get to where you want your XAMPP file to be saved to, choose a location for it or leave it at the default location

   NOTE: Keep track of where you download your XAMPP to, we'll need to access this file later

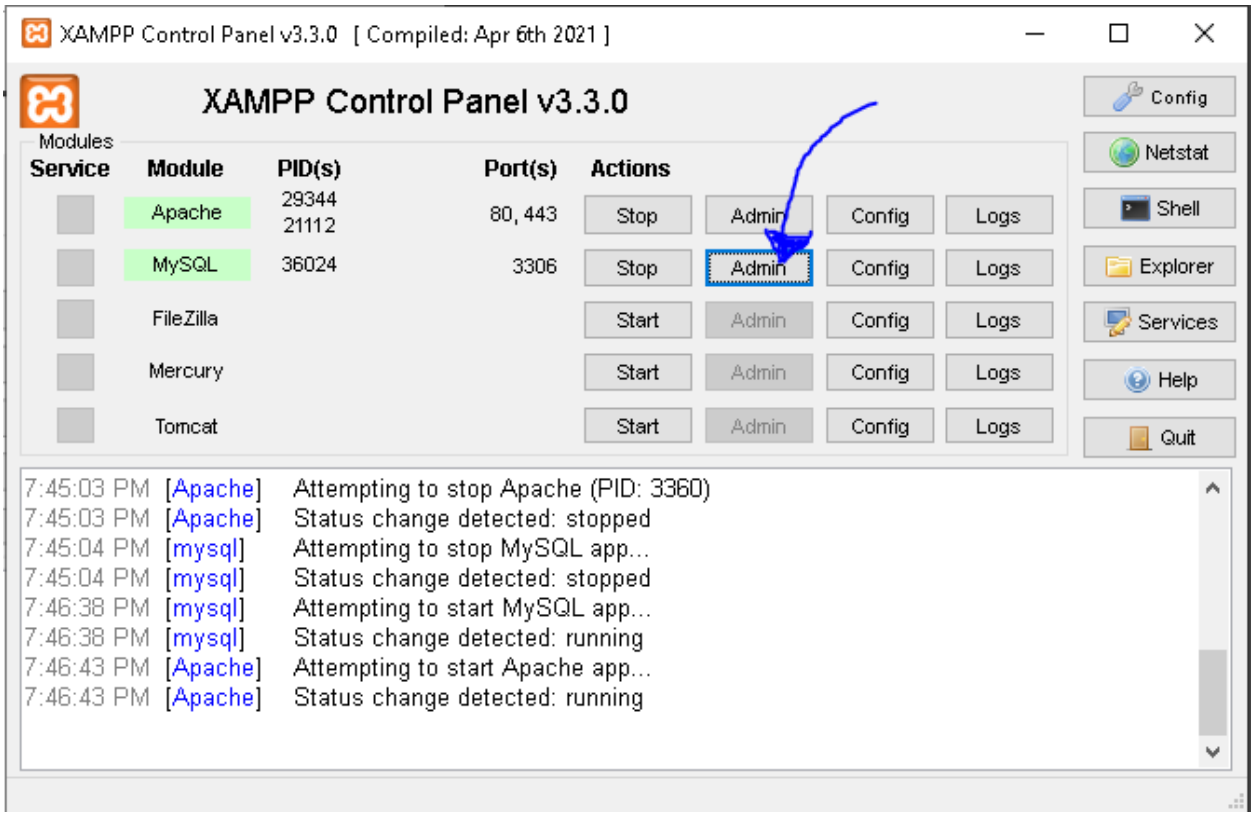


2) Once installed, start MySQL and Apache

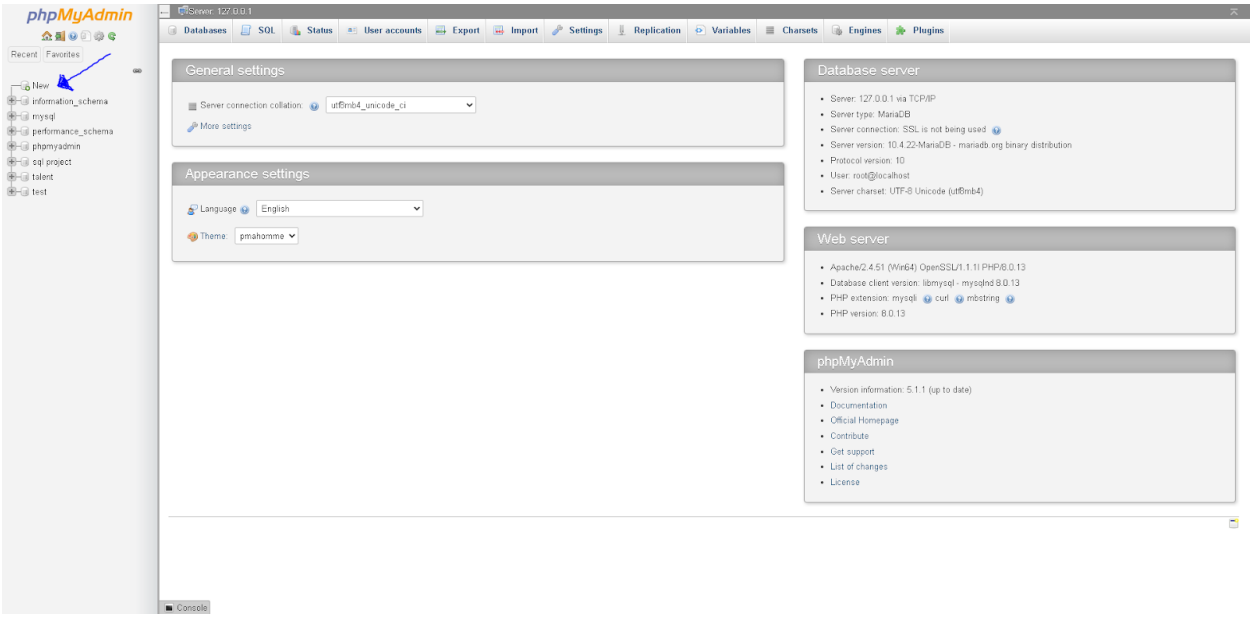


3) Then, click on the admin button for MySql, and this will take you to phpadmin. (If for some reason it doesn't, just type "http://localhost/phpmyadmin/" into your browser)

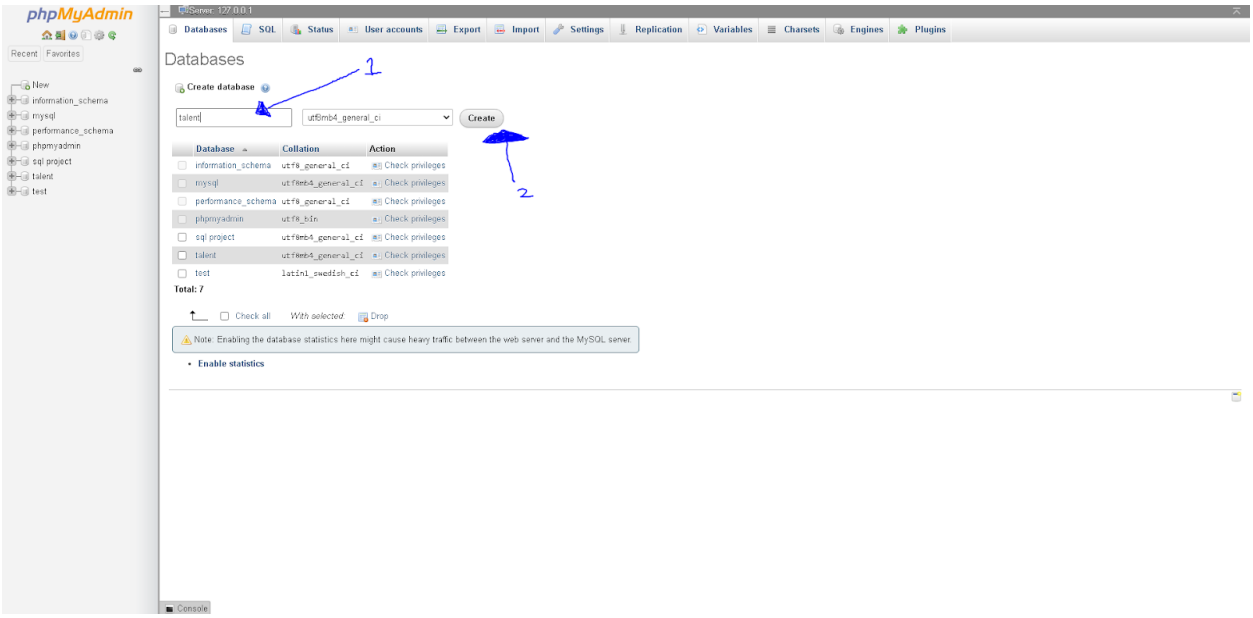
You may get a prompt asking windows for permission, just click allow



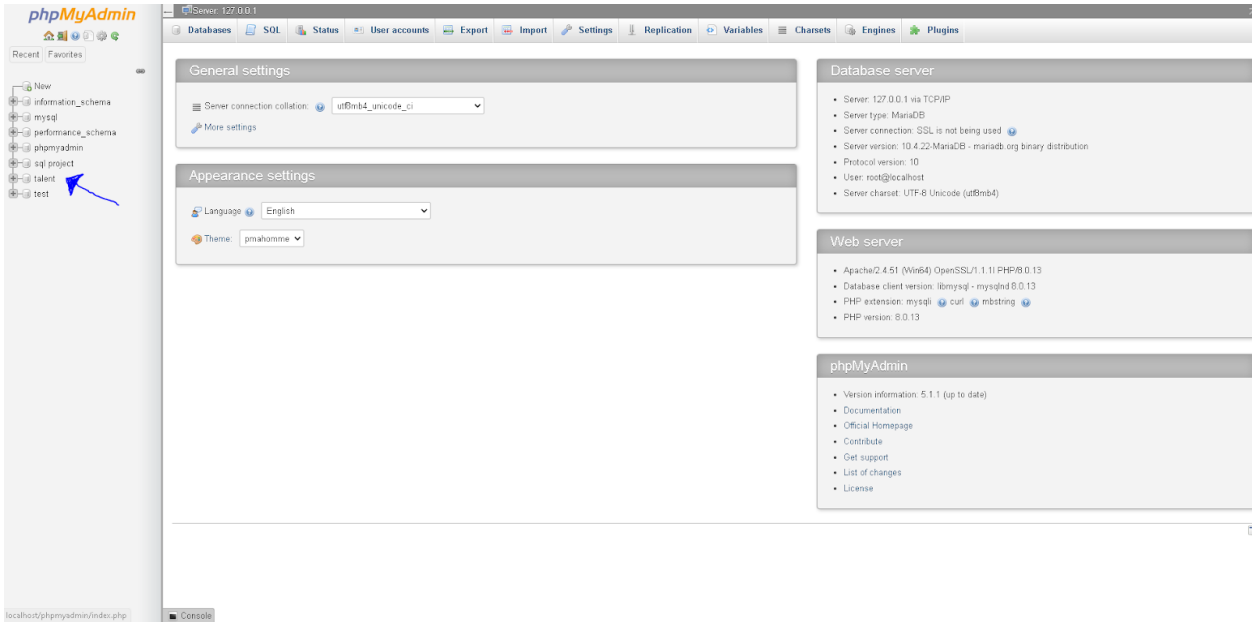
4) Once there, click on new on the left hand side towards the top.



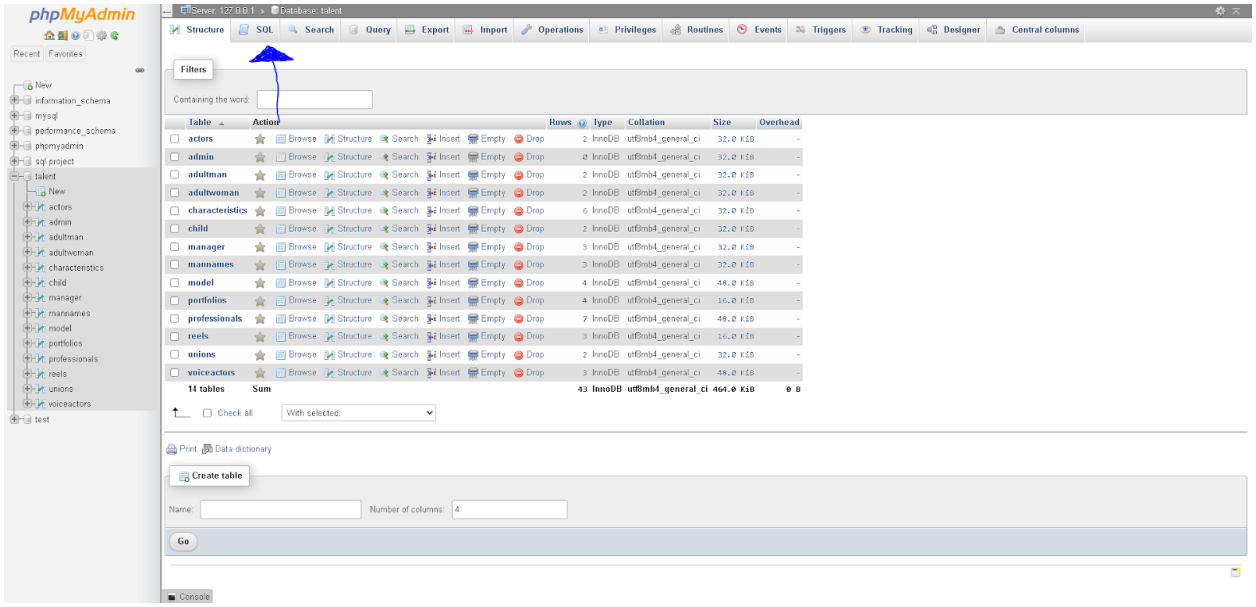
5) Create a new database and make sure the database name is talent.



6) Then, click on talent which should appear in the columns on the left side.

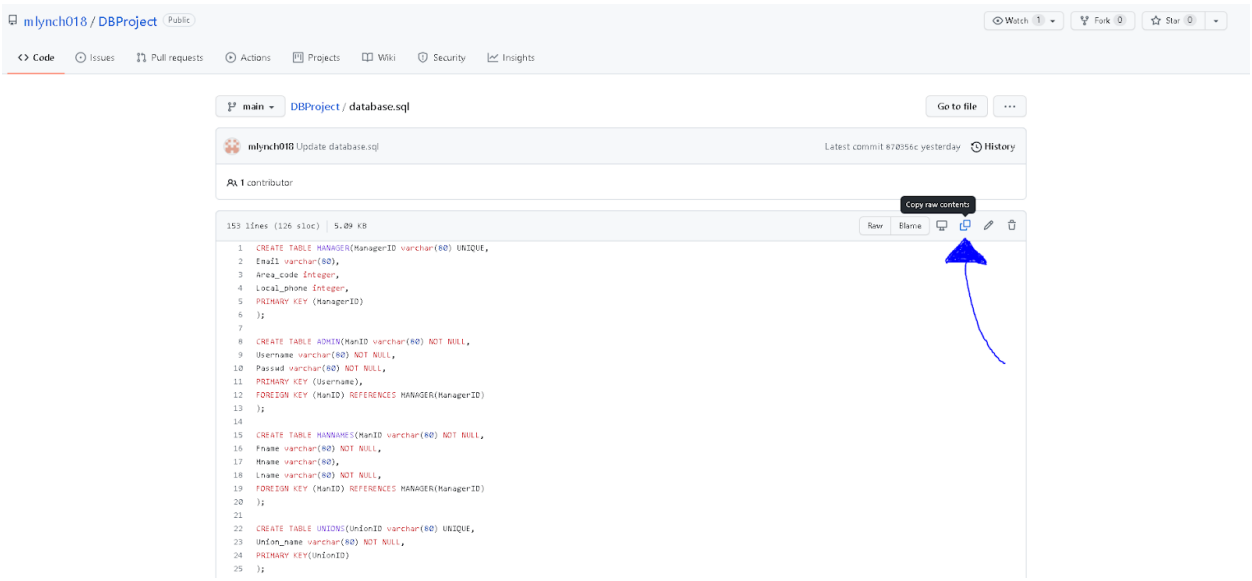


7) Next, click on the SQL tab on the bar near the top

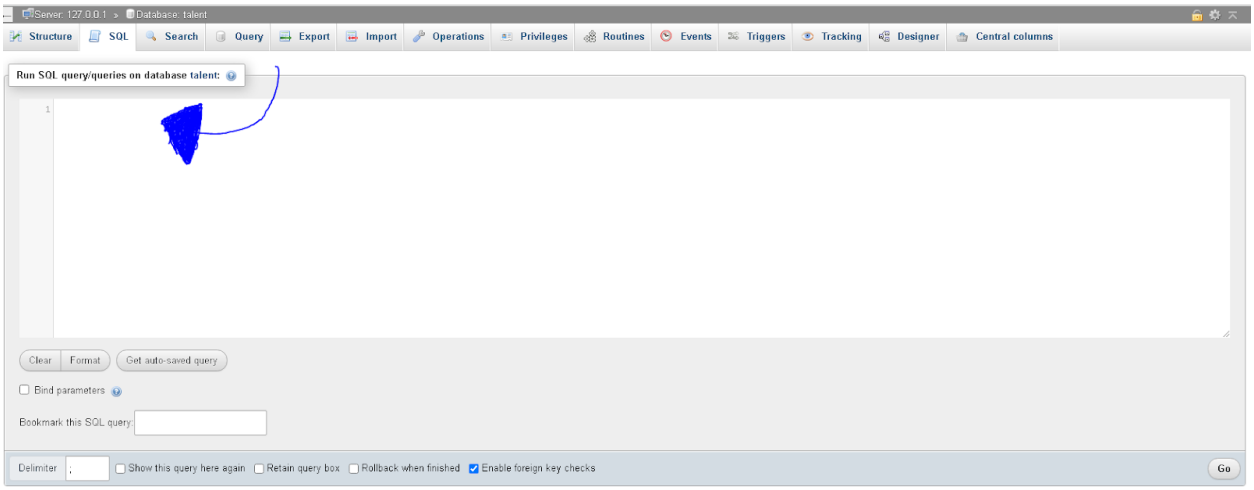


8) Copy all the code from the .sql file on our github link

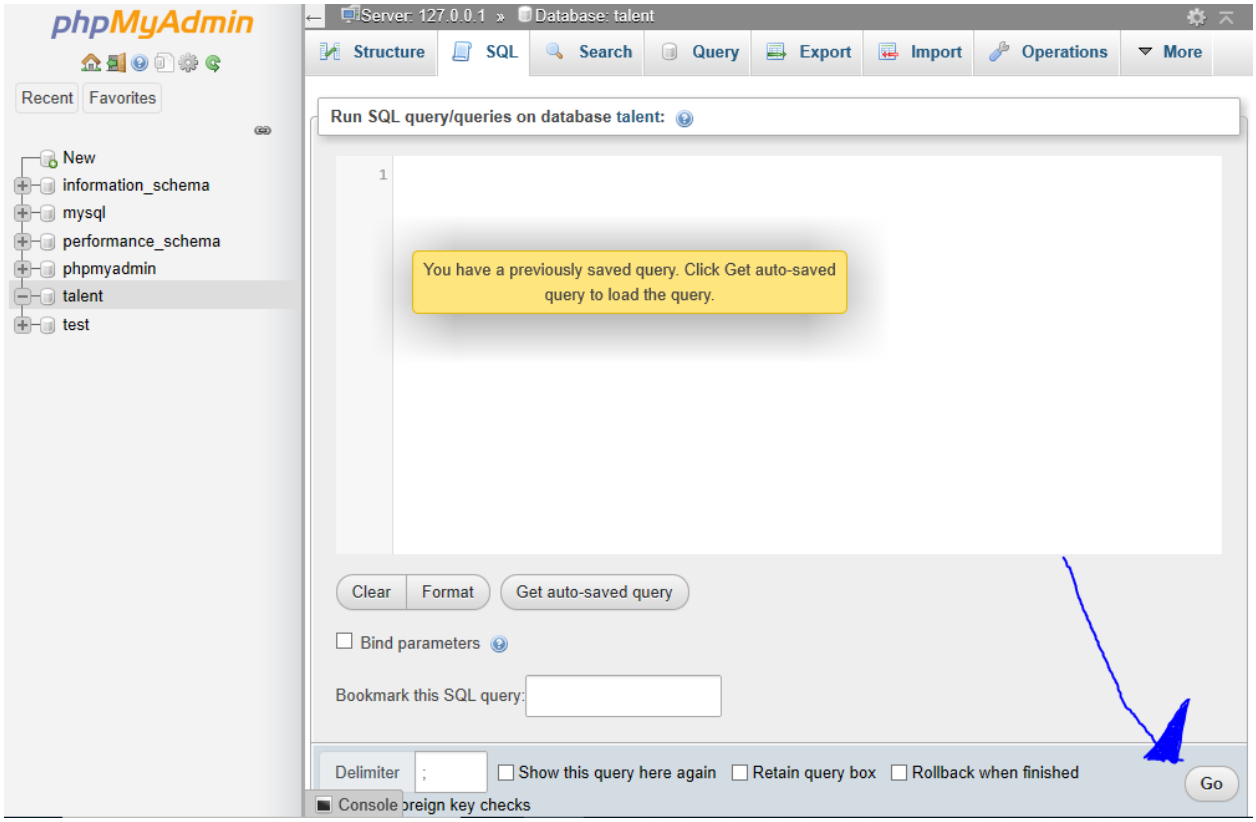
(If for some reason this won't let you paste in the next step, just manually select all of the code using your mouse and copy/paste it that way)



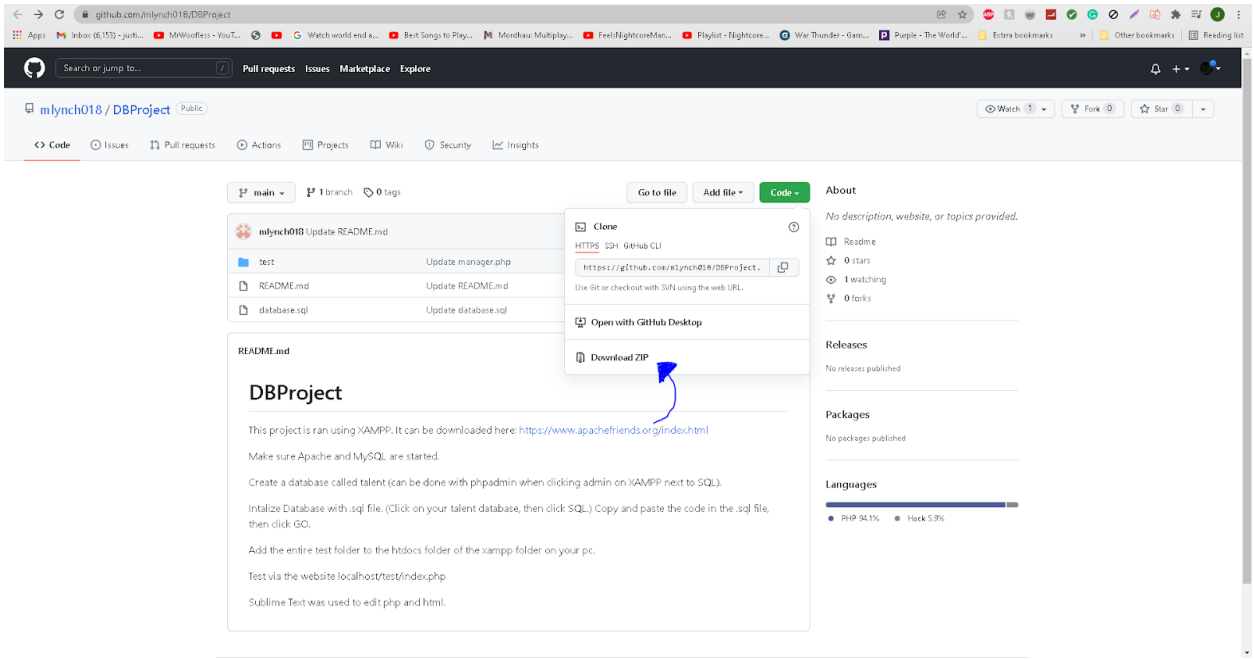
9) Left click once into the center text box, then Ctrl-V to paste the code into the box or right click and paste into the field.



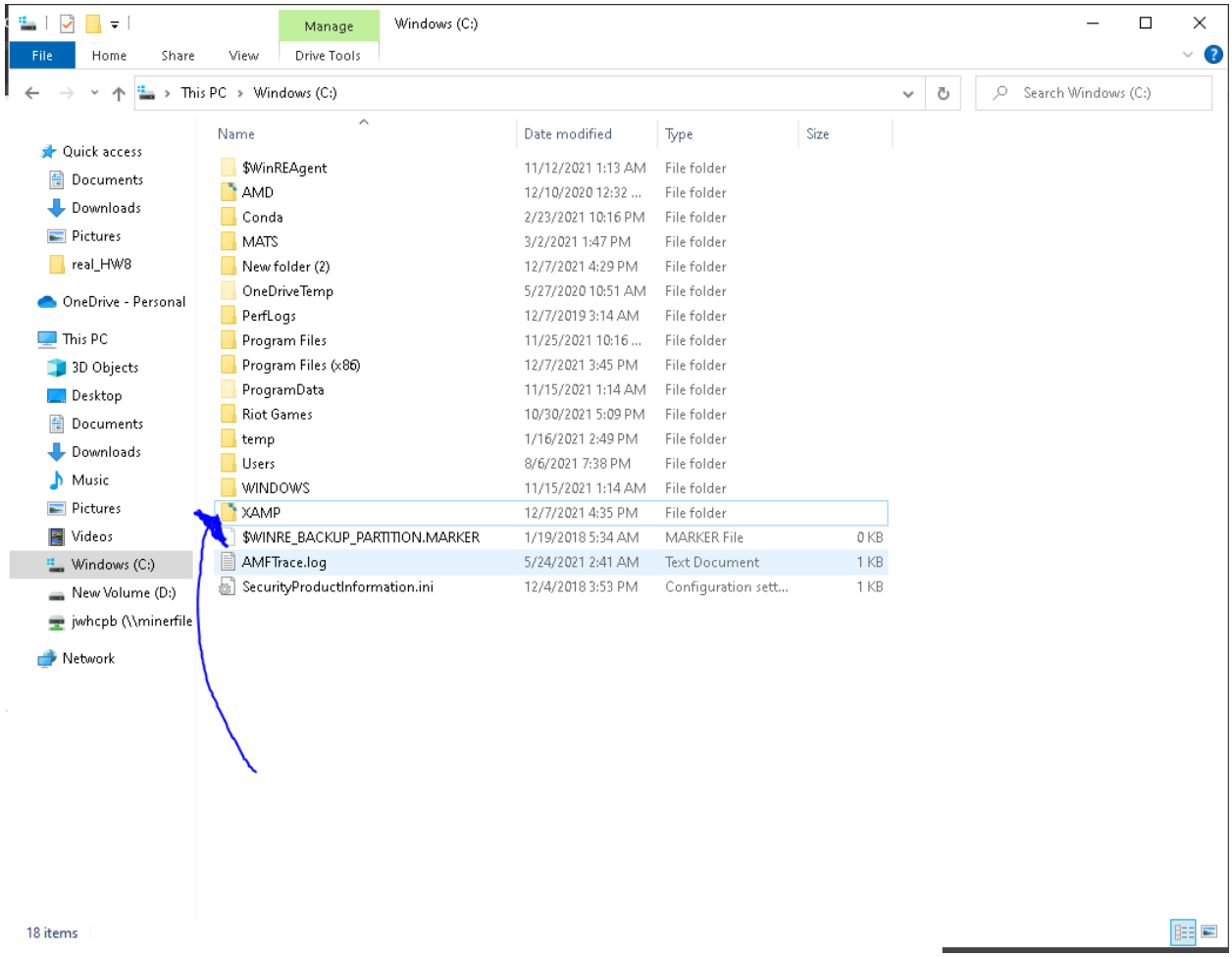
10) After doing so Click the “Go” Button in the bottom right hand corner



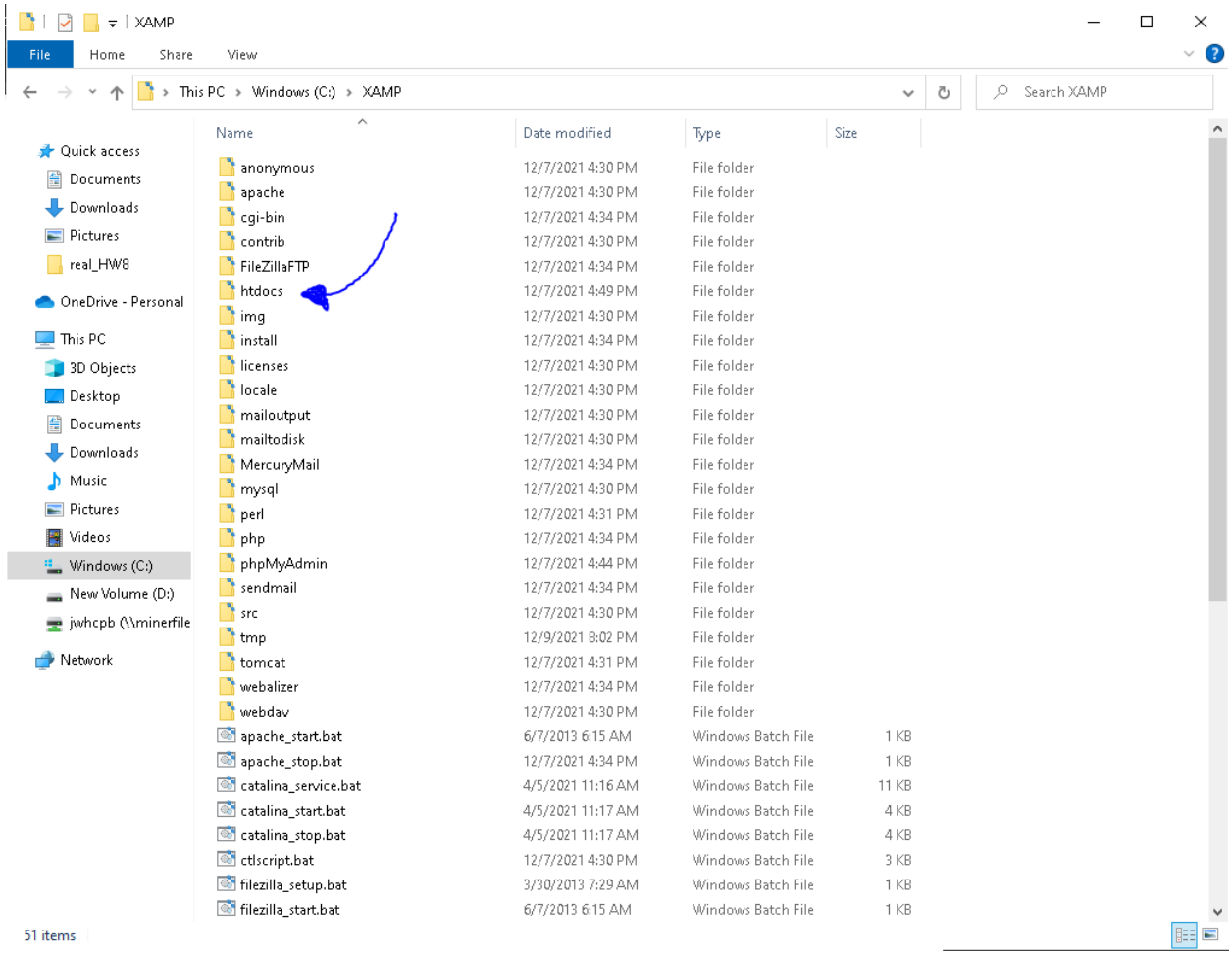
11) Now, you need to get all the php code. Download the entire repository as a zip folder and unzip it.



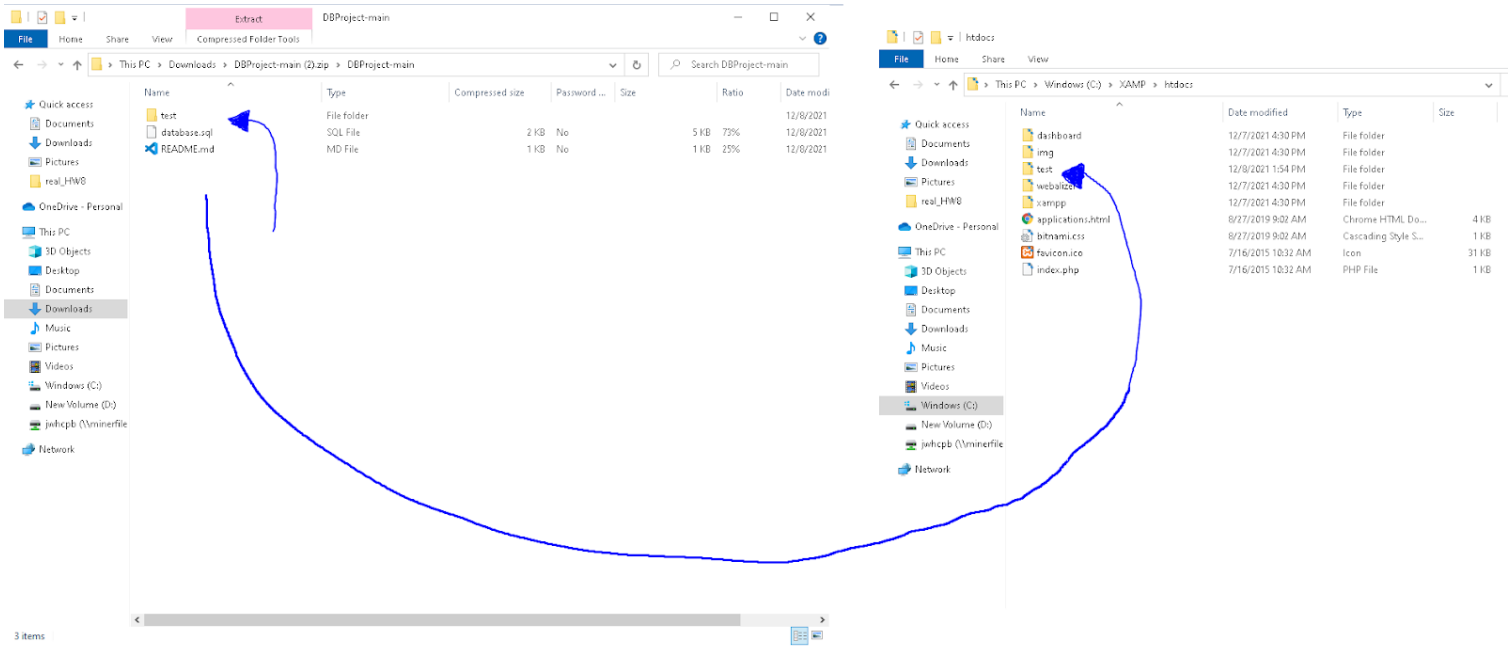
12) Locate your xampp folder on your pc and go into it. It is probably under the c drive. (Or wherever you made your XAMPP file download as mentioned in step 1)



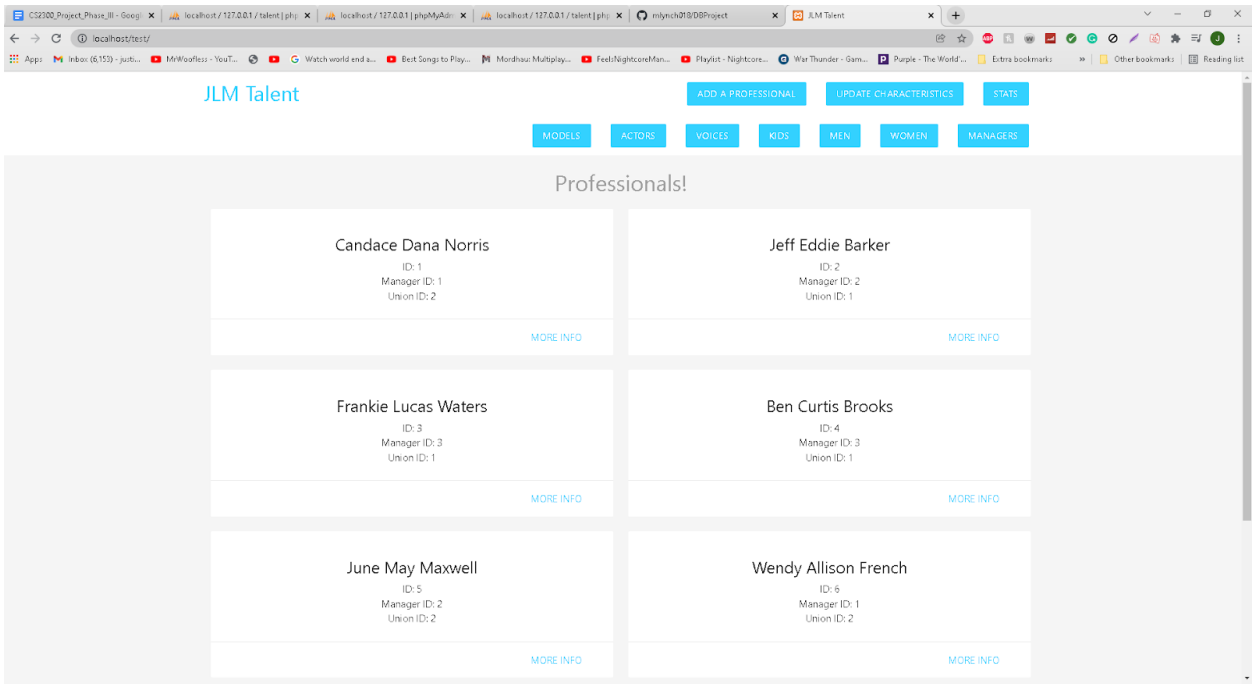
13) Once you are into your Xampp file, Click into htdocs.



14) Once you are in htdocs, copy and paste the entire “test” folder from your unzipped github folder into htdocs.



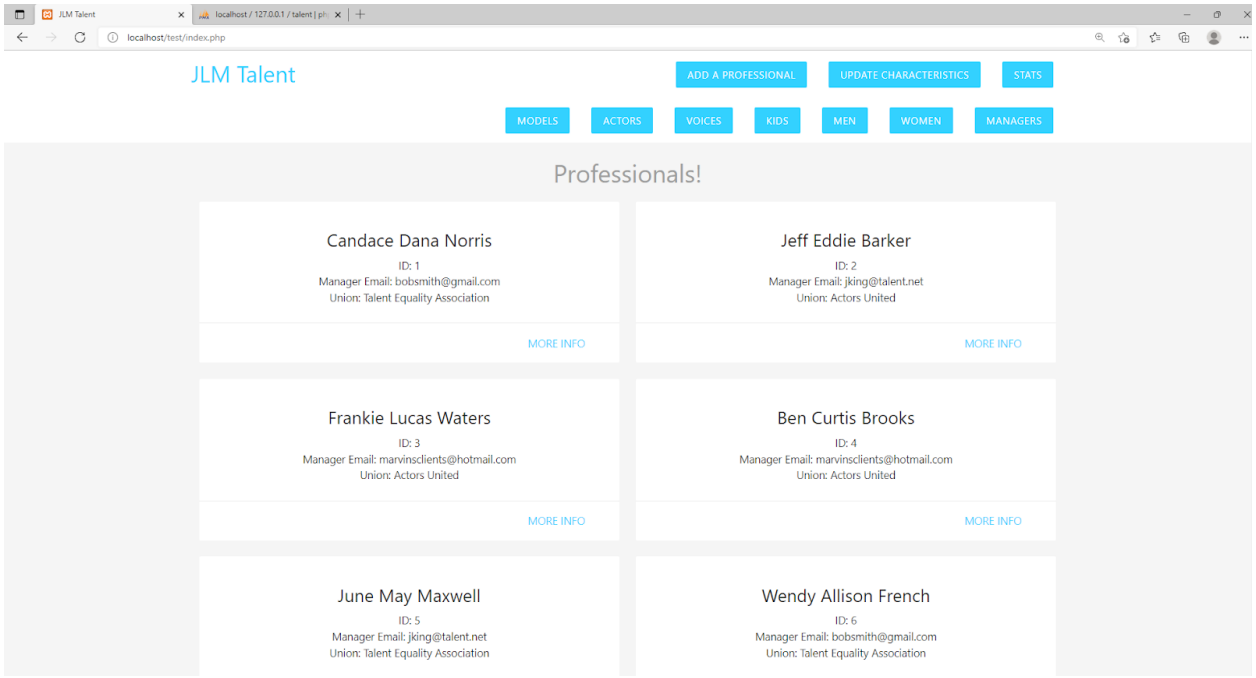
15) You are now ready to test the website. You can do this in a web browser via localhost/test.



User Manual:

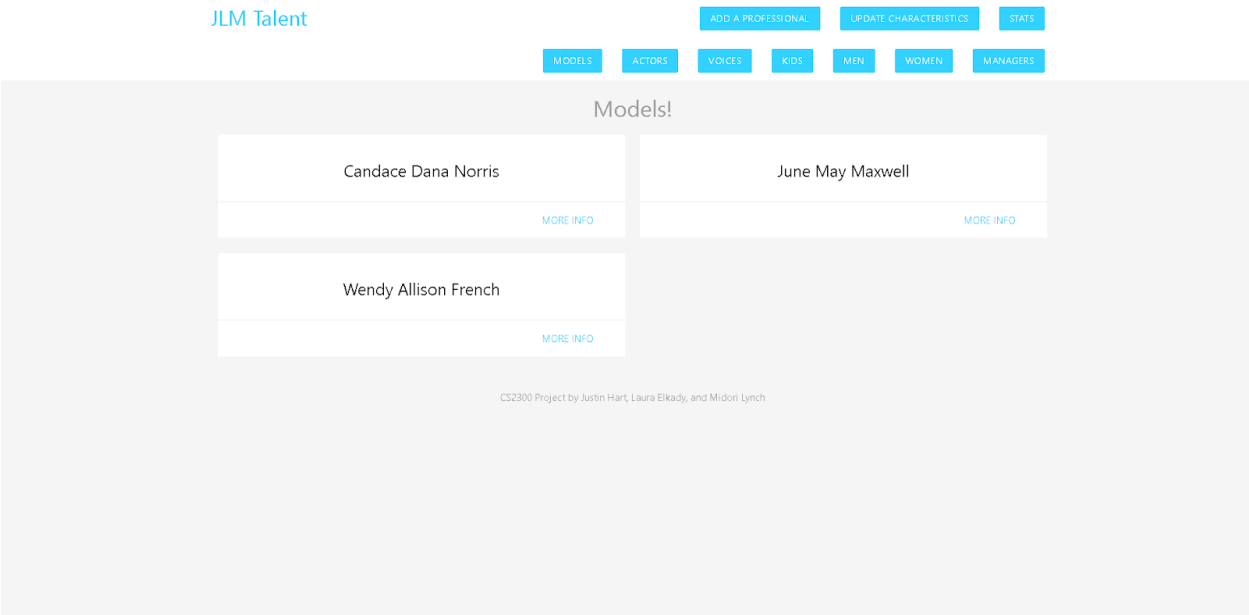
Home Page:

On the homepage you will find many professional profiles and some buttons at the top of the page. Each Button can be clicked on in order to navigate to a different page. Each profile can be selected in order to view more information of the selected professional.

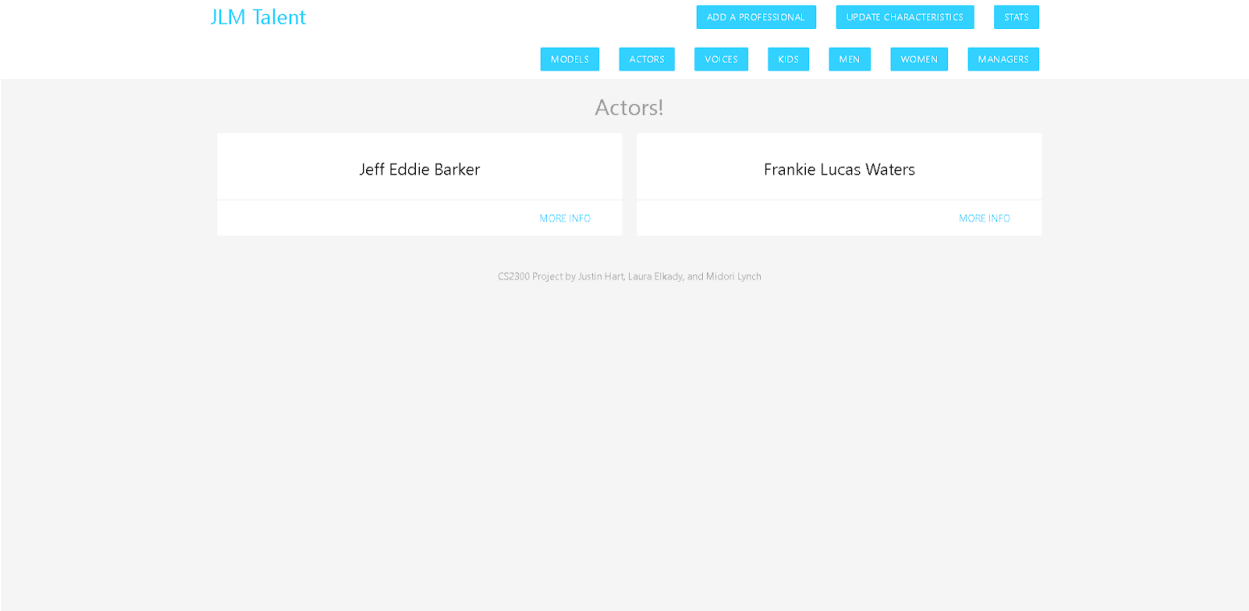

JLM Talent
localhost / 127.0.0.1 / talent | ph
localhost/test/index.php
JLM Talent
ADD A PROFESSIONAL
UPDATE CHARACTERISTICS
STATS
MODELS
ACTORS
VOICES
KIDS
MEN
WOMEN
MANAGERS
Professionals!
Candace Dana Norris
ID: 1
Manager Email: bobsmith@gmail.com
Union: Talent Equality Association
MORE INFO
Jeff Eddie Barker
ID: 2
Manager Email: jking@talent.net
Union: Actors United
MORE INFO
Frankie Lucas Waters
ID: 3
Manager Email: marvinsclients@hotmail.com
Union: Actors United
MORE INFO
Ben Curtis Brooks
ID: 4
Manager Email: marvinsclients@hotmail.com
Union: Actors United
MORE INFO
June May Maxwell
ID: 5
Manager Email: jking@talent.net
Union: Talent Equality Association
Wendy Allison French
ID: 6
Manager Email: bobsmith@gmail.com
Union: Talent Equality Association

Models Page:

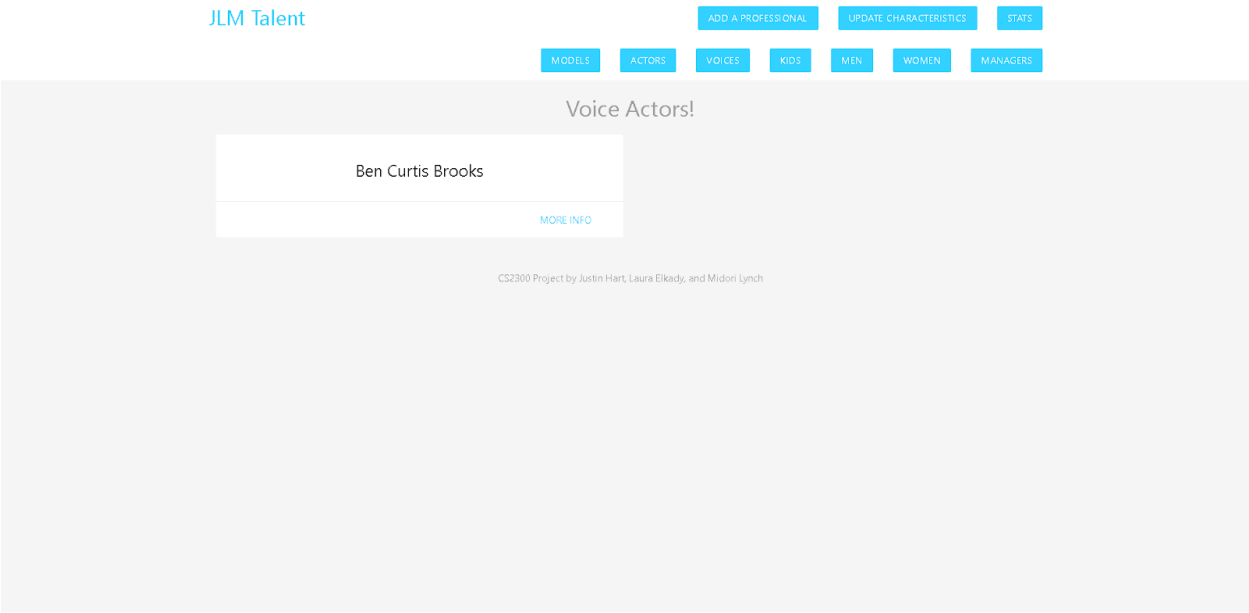
This is the Models page, profiles for professionals who are models can be viewed here. The Actors page and Voices page follow the same format. Select the “More Info” button to view more information on the specific professional.



Actors Page:

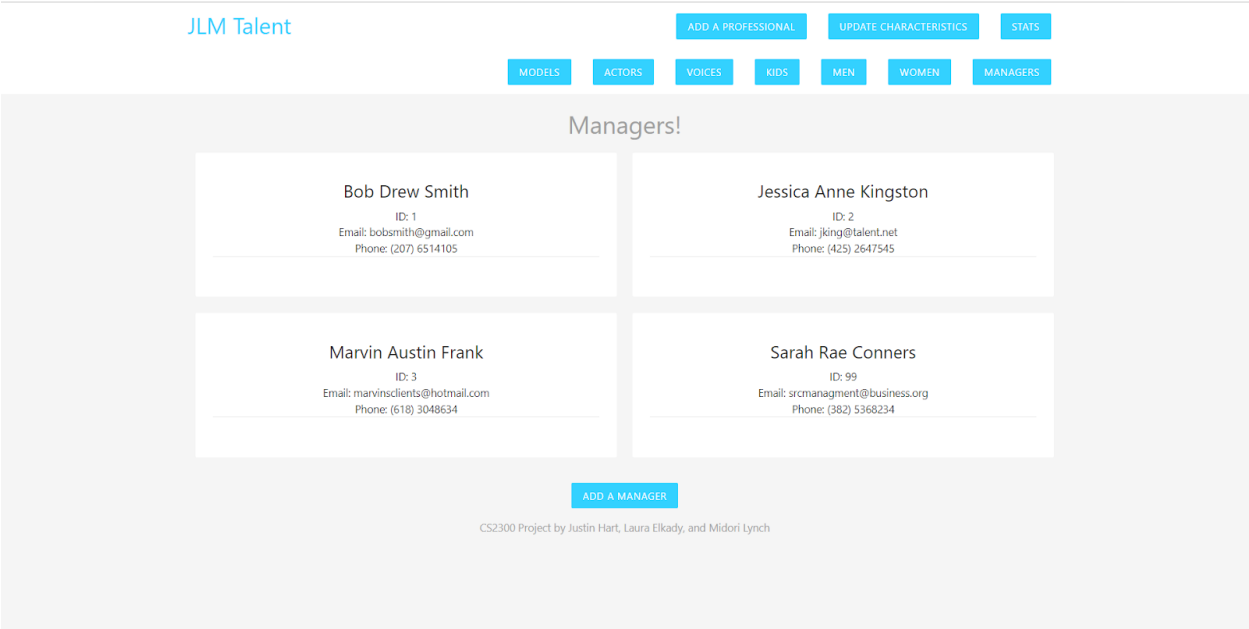


Voices Page:



Managers Page:

This is the managers page, it is a little different from the professionals pages. To add a manager select the “Add a Manager” button.



Update Characteristics Page:

Here is the Update Characteristics page. In order to edit information you must provide the ID number of the professional. Once information is inserted, select the “Update” button.