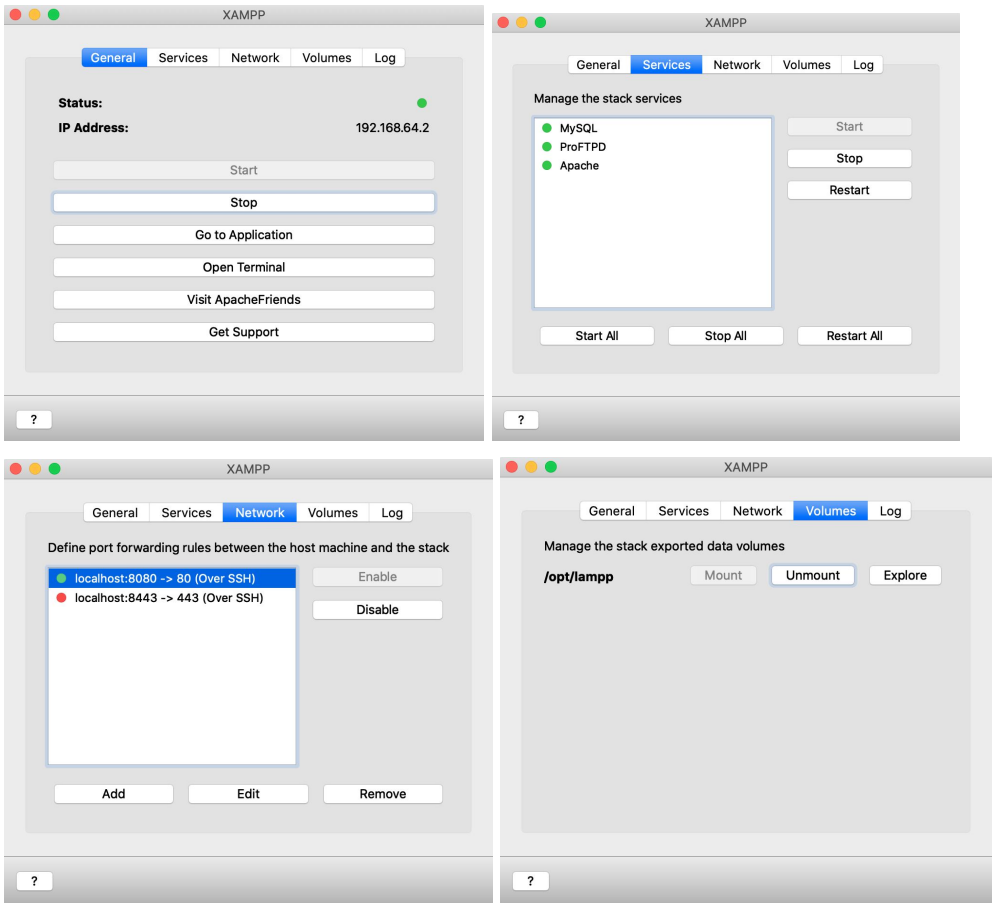
Implementation Documentation

For this project, we have used programming languages PHP and JavaScript, and markup language HTML with style sheet language CSS. The integrated development environment and tools needed for this project are Sublime Text, cross-platform web server Xampp, and phpMyAdmin software tool for handling MySQL databases.

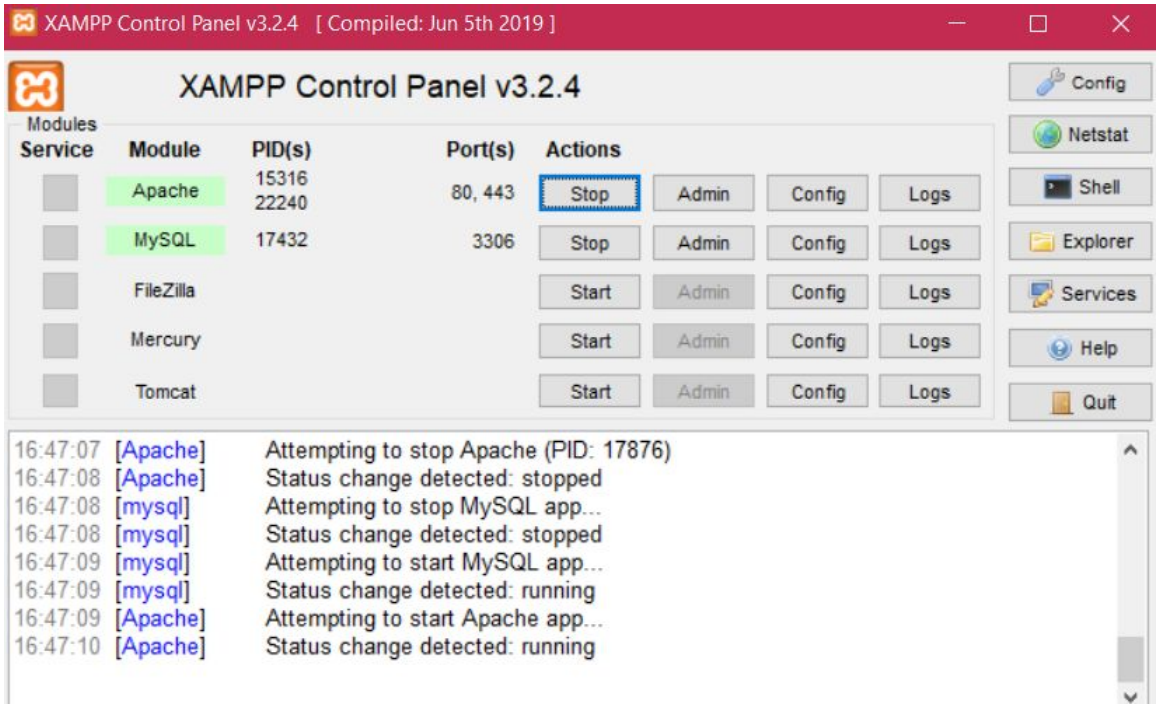
Set Up

To begin developing our application, we needed to find a platform that would allow us to link out front-end and back-end development code with a database. Xampp would allow us to do this, so one firstly needs to download it through https://www.apachefriends.org/index.html .

Check that all the features are running correctly as shown below (for Mac).



When it comes to Windows OS, after you are done with the installation, if not opened automatically, go to /xampp folder where you will find the controller as xampp-control.exe.



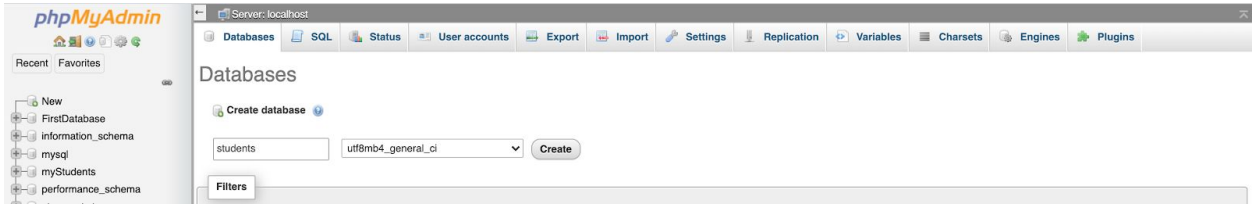
Here make sure Apache and MySQL are set to start.

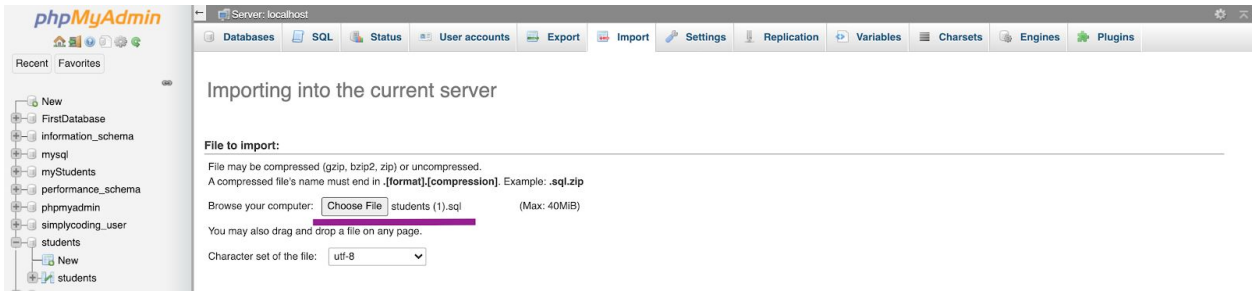
Now open http://localhost:8080/phpmyadmin/. Important part is to add a user “studentdb” with password “timephp” to phpmyAdmin so that the files can connect with the database through localhost correctly.

An example of why this is important is that we are connecting to the database in phpmyAdmin multiple times as below and it needs to be under this credentials. (Another way to be to change all of the credentials according to each local host, which we found to be a more inconvenient approach.)

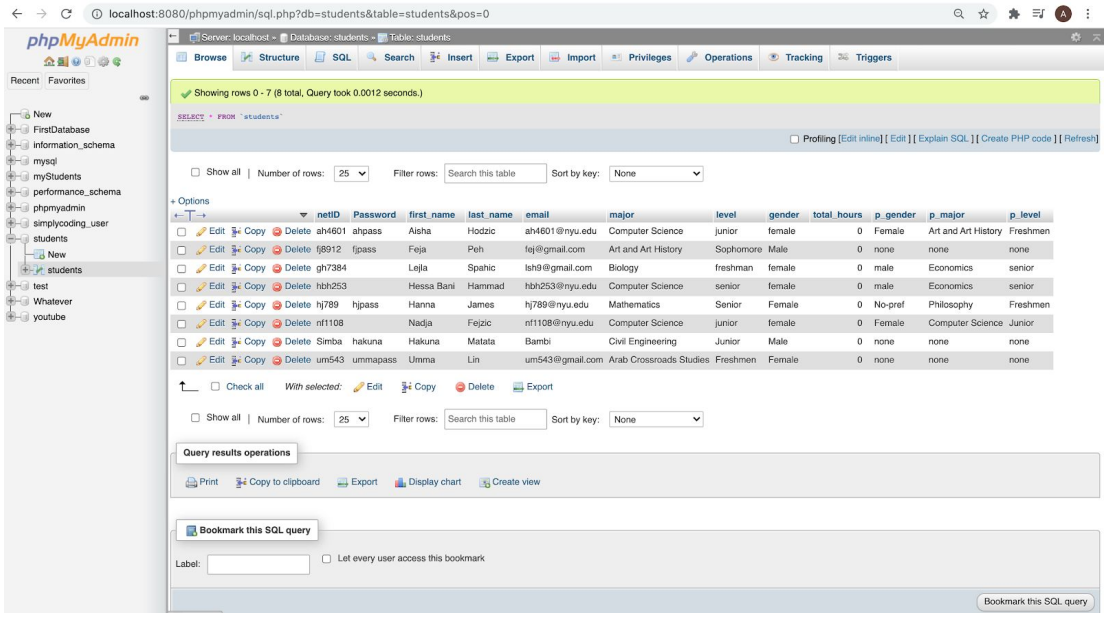
```php
$dbhst = "localhost";
$dbnme = "students";
$bdusr = "studentdb";
$dbpws = "timephp";


$conn = new PDO('mysql:host='.$dbhst.';dbname='.$dbnme, $bdusr, $dbpws);
```

In order to import our database, you’ll firstly have to go to pypmyAdmin,click “New”, write “students”, click Create.
Then you can import the students.sql that we have provided.

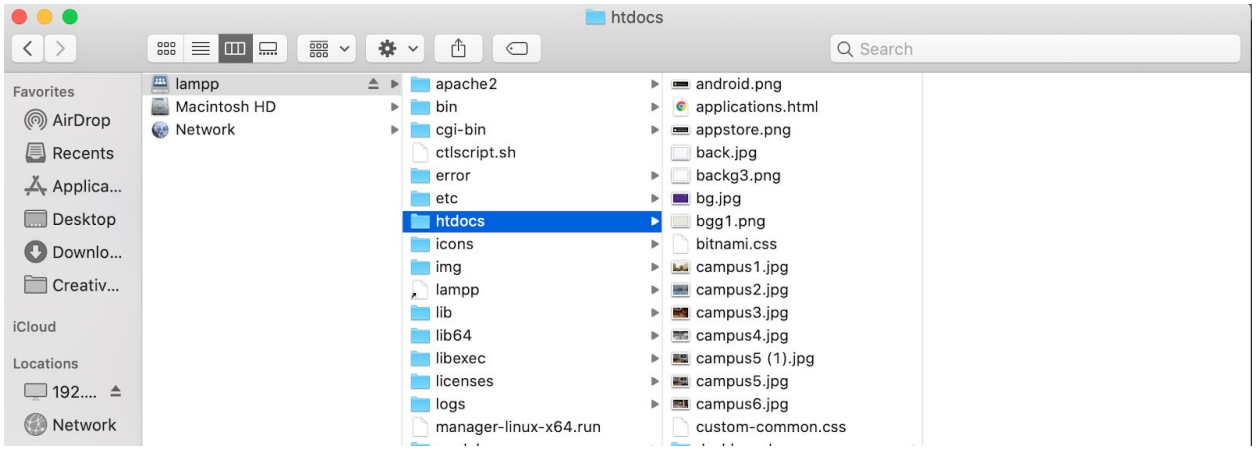




You should now be able to see that the table has loaded looking like this.



Now, place <u>all files (</u>including photos) in -> lampp/htdocs. The path is also shown below.



In your browser, type http://localhost:8080/StudyBuddy.html and enjoy our application!

Code: Behind the scenes

The features such as titles, forms, photos of each page are displayed through HTML. Every HTML page is connected to a CSS page that determines the needed styles and design. As our main features of the product are based on the forms, either through registration, login and deciding preferences, we will explain that process of communicating to the database through the forms. The HTMLs pages are then later connected to the PHP code, mostly through forms. The following form from the main.html:
<form id="testForm" action="insert.php" method="post">
is connected to insert.php, meaning that all the answers given in the form will be handled in the PHP called insert.php. Each php page connects to the database in order to set or update the variables. An example of storing form answers named “j_username” and “j_password” is below.

```
$statusnetID = mysqli_real_escape_string($link, $_REQUEST['j_username']);
$loginpassword = mysqli_real_escape_string($link, $_REQUEST['j_password']);
```

The insert.php reads the database first, then reads all the answers from the form, stores them into variables, and places them into the database if needed. We might choose to extract data from the database, through SELECT - FROM - WHERE function, insert new data into database through INSERT INTO (database) VALUES (values) function, or update just some of our variables through the UPDATE function.

```
$sql = " SELECT netID, Password,status FROM students WHERE netID='$loginnetID'";
```

All the work can be checked in the database, which reflects the successful integration of HTMLs, CSSs and PHPs documents.

Another application of integration was done to achieve the successful matching results. The user is able to indicate their study buddy preferences, upon which those preferences where compared to the major, gender and year of all students in the database in order to find either a full match (satisfying all preferences) or a nice match (satisfies at least one indicated preference).

```
// Comparing answers
$sql = " SELECT netID, first_name, last_name, major,email,status FROM students WHERE major='$
  pref_major' and level='$pref_level' and gender='$pref_gender' and status='Active' ";
$result = $conn->query($sql);

if ($result) {
  // output data of each row
  while($row = $result->fetch(PDO::FETCH_ASSOC)) {

    echo '<div class="mySlides fade">';

    if (strcmp($itsmyNetID, $row["netID"]) != 0) {
    echo "<br> Email: ". $row["email"]. " - Name: ". $row["first_name"]. " " . $row["last_name"] . "<
      br>";
    }
    echo '</div>';

  }
} else {
  echo "0 100% match results";
}
```

This code refers to finding a 100% match. First we read the data from the database and the compare them with the given preferences. Lastly, we'll display the results with the email and name so that the user can contact their match. The matching based on at least one preference is done through a similar way, but by using a different truth table value due to the different requirement of the functionality.

Technical Expertise

We are fully aware that there might be better choices regarding databases, platforms and tools used for this application, however, given the time constraint and technical background, we believe that these are the best possible solutions. In a very short time period, we have been able to self-teach ourselves everything about PHP, HTML and CSS, making and handling databases from scratch which we haven't worked with before. Rather than being considered just a lack of technical knowledge due to the no prior experience regarding the application development, we believe that this project shows the immensely fast-learning process, communication, team work and our capabilities in grasping knowledge and coding skills to meet the product requirements, time deadline, as well as given and personal expectations.