



SOFE 3700U: Data Management Systems

Final Report

Supervised By: Dr. Khalid A. Hafeez

Project Title:

University Database Systems

Group Members and IDs

Rayyan Mohammed (100752351)

Daniyal Khan (100750029)

Mohammad Khalid (100746801)

Yousif Sarmad (100746887)

Charles Olagunju (100749818)

Table of Contents

Introduction	3
Background and Motivation	4
System Design	5
Observations and Analysis	10
Conclusion	11
References	12
Appendices	16
Contribution Matrix	

Introduction

The goal of our website is for universities to increase efficiency in having their data stored and retrieved in a categorized manner. Our motivation to do this database system is due to having used very similar systems throughout high school and university to select courses or update personal information. We find it very intriguing and are very interested in learning more about how they work on a data storage level.

To begin, we will plan out this project by putting our thoughts on numerous distinct diagrams. To help define our project and ideas, we employ diagrams like related schema and E-R diagrams, as well as sample data and views. The front end of our system will be created using HTML, CSS, and JavaScript, and the back end will be created using PHP, XML, or JSON, to connect the front end to the databases. Overall, these processes will aid in the creation and shaping of a university database.

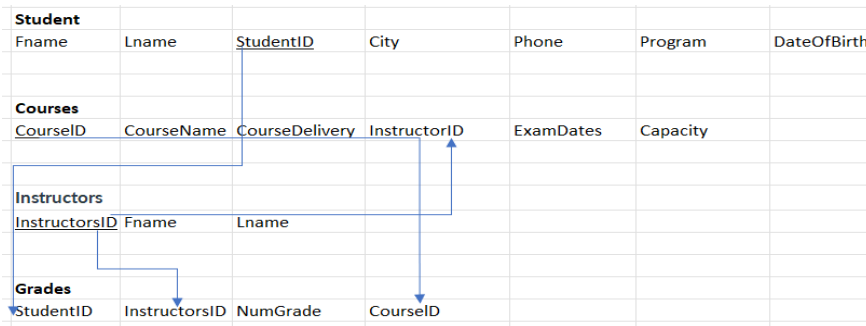
Background and Motivation

Our project will focus on the area of creating a university database system. This database will be designed in such a way that it will take care of all the needs of major universities, so they are able to use our system to catalog and systemize all courses and students' information. Our system will be more extensive than a typical university database and will provide information to students and staff to make educated decisions. This will make our system more efficient and user friendly so that universities would prefer our system over others.

Our motivation to take on this database system is due to similar systems being used over the years of our education. The previous systems were not user friendly, and it was hard to register and get information about the courses selected. Some related work we are basing our project on is our university's database on MyCampus. The components we would be recreating are the transcripts and the course availability. Our database will differ from the university's database because it will include more options and information about the courses. Some examples of the information we would add are midterm dates and course deliverables. We will also improve the interface of the database, which will make it easier for the user while interacting with the database. This will make our system differ from other systems that are available.

System Design

The very first step we had to take was to visualize our system. We started off with a relational schema to create our tables for our database and identified all of our primary keys and foreign keys within all the tables.



The next step was to create our views needed for the database. We began with a schema for our University Database System, and then creating SQL commands for our required tables: Student, Instructor, Courses, and Grades (Project Phase II).

After our tables were made, we had to populate them with sample data. We created our own tables in excel with our own names for simplicity of query search and gave randomized information to each of our attributes.

Student						
StudentID	Fname	Lname	City	Phone	DateOfBirth	Program
10001	Rayyan	Mohammed	Toronto	416-875-2145	2001-10-23	Engineering
10002	Daniyal	Khan	Toronto	647-854-7865	2001-03-26	Business
10003	Charles	Olagunju	Pickering	209-451-8765	2001-07-24	Health Science
10004	Yousif	Sarmad	Oshawa	905-324-7851	1992-01-01	Engineering
10005	Zaem	Khalid	Oshawa	905-789-3214	1990-05-12	Business
10006	John	Doe	Mississauga	209-147-8532	1983-12-15	Nursing

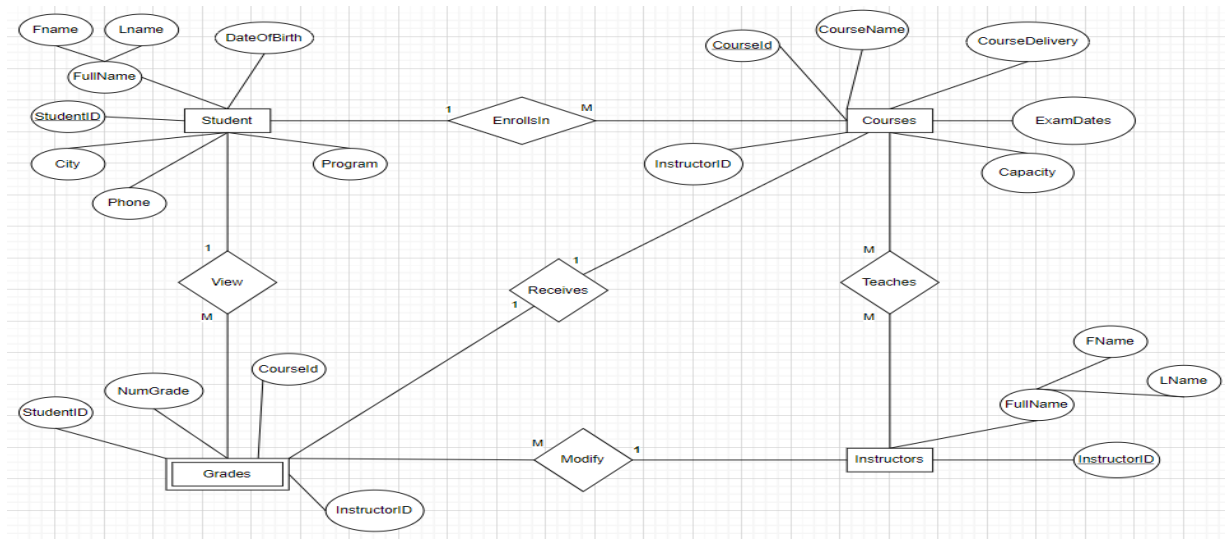
Courses					
CourseID	CourseName	CourseDelivery	Capacity	ExamDates	InstructorID
ENG3200	Intro to Engineering	In Person	140	2021-12-15	20001
ENG3850	Engineering Economics	In Person	140	2021-12-12	20001
BUS2840	Accounting Principles	Hybrid	200	2021-12-13	20002
BUS1470	Intro to Business	Online	500	2021-12-09	20002
HSCI4800	Life Science Advanced	Hybrid	200	2021-12-12	20004
NUR1000	Nursing Principles	In Person	100	2021-12-15	20006
HSCI1000	Intro to Life Science	Online	500	2021-12-05	20003
NUR4000	Long Term Care	Hybrid	200	2021-12-07	20005

Instructors		
InstructorID	Fname	Lname
20001	Khalid	Hafeez
20002	Usman	Aziz
20003	Farhan	Mohammed
20004	David	Suzuki
20005	LeBron	James
20006	Franklin	Torgbo

Grades			
StudentID	CourseID	InstructorID	NumGrade
10001	ENG3200	20001	83
10001	ENG3850	20001	74
10002	BUS2840	20002	91
10002	BUS1470	20002	89
10003	HSCI4800	20004	84
10003	HSCI1000	20006	62
10004	ENG3200	20001	53
10004	ENG3850	20001	66
10005	BUS2840	20002	41
10005	BUS1470	20002	86
10006	NUR1000	20006	33
10006	NUR4000	20005	27

In addition to the relational schema, tables, and ER Diagram, we also created 10 views that a user may find helpful when searching for their information in the University Database Systems (Project Phase II).

Lastly, once all of that was completed, we moved on to creating our ER Diagram, to get a good visual understanding of our system and to visualize the connections between each of our tables and their attributes.



Once our foundation was set for our University Database System, it was time to create the website and implement the database. For our website, we used XAMPP which acts as a local host server to run our system. Then we had to transfer our database from MYSQL Workbench to PhpMyAdmin, making sure our naming conventions for the tables and schema were consistent. Next, we imported/transferred our sample data from the excel sheet into the appropriate tables.

Now that our database was implemented, we had to start coding our website. The first page of our website is a login/registration page. For the login component, we used a FORM that has a POST method and has an ACTION set to the file "login.php" which contains our php code for the login. This allows our website to interact with the database and check if the user inputted the correct information to login. For our login.php file, we first needed to establish a connection to our database with our appropriate username, password, and schema name. Next, we got the information the user inputted in our login FORM and used a SELECT statement, which checks if the studentID and firstname match the information from our student table. Based on if the input and the data matches, an if/else statement allows the user to either be directed to the next page or taken back to the login page with a pop alert displaying "Please try again".

```

<div id="box">
  <form action="login.php" method="post">
    <div id="login">
      <h1>Welcome Please Login</h1>

      <label>First Name:</label>
      <input type="text" name="firstname" required><br><br>
      <label>StudentID:</label>
      <input type="text" name="studentid" required><br><br>
      <button type="submit">Login</button>
    </div>
  </form>

```

```

session_start();

$con = mysqli_connect('localhost','root','');
mysqli_select_db($con,'FinalProject');

$studentid = $_POST['studentid'];
$firstname = $_POST['firstname'];

$$ = " select * from Student where StudentID = '$studentid' && FName = '$firstname'";
$result = mysqli_query($con, $$);

$num = mysqli_num_rows($result);

if($num == 1){
  $_SESSION['firstname'] = $firstname;
  $_SESSION['studentid'] = $studentid;
  header('location:home.php');
}
else{
  echo "<script>
  alert ('Sorry you have entered the wrong information. Please try again');
  window.location.href = 'data.html';
  </script>";
}

```

The next component we implemented was for registration. We again used a FORM that contained a POST method, but this time the ACTION was set to our “registration.php” code file. This component will directly interact with our database by storing and saving the users information into the Students table. The code follows the same steps as the login component; however, it only checks the users studentID with a SELECT statement to make sure the id isn’t already taken. This is because studentID’s are unique to each person, making them a primary key. If the id was already taken, the website directs you back to the login/registration page with another pop alert displaying “StudentID is already taken, please try again”. If the id isn’t taken, it uses an INSERT INTO statement which stores the user information into the Students table. The user will be alerted that their registration was successful, directing them back to the login page.

```

<form action="registration.php" method="post">
  <div id="sign">
    <h1>Registration</h1>

    <label>StudentID:</label>
    <input type="text" name="studentid" required><br><br>
    <label>First Name:</label>
    <input type="text" name="firstname" required><br><br>
    <label>Last Name:</label>
    <input type="text" name="lastname" required><br><br>
    <label>City:</label>
    <input type="text" name="city" required><br><br>
    <label>Phone:</label>
    <input type="text" name="phone" required><br><br>
    <label>DateOfBirth:</label>
    <input type="text" name="dateofbirth" required><br><br>
    <label>Program:</label>
    <input type="text" name="program" required><br><br>
    <button type="submit">Sign Up</button>
  </div>
</form>

```

```

<?php
session_start();

$con = mysqli_connect('localhost','root','');
mysqli_select_db($con,'FinalProject');

$studentid = $_POST['studentid'];
$firstname = $_POST['firstname'];
$lastname = $_POST['lastname'];
$city = $_POST['city'];
$phone = $_POST['phone'];
$dateofbirth = $_POST['dateofbirth'];
$program = $_POST['program'];

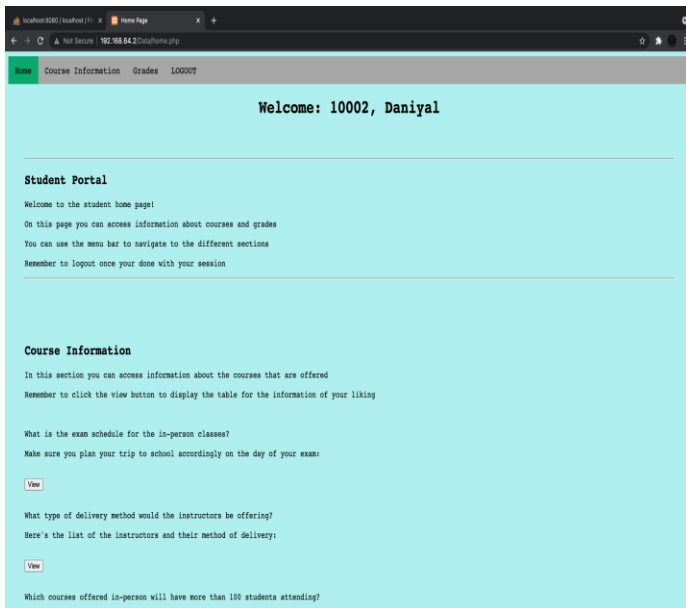
$$ = " select * from Student where StudentID = '$studentid'";
$result = mysqli_query($con, $$);

$num = mysqli_num_rows($result);

if($num == 1){
  echo "<script>
  alert ('Sorry you the studentid you have entered is already taken. Please try again');
  window.location.href = 'data.html';
  </script>";
}
else{
  $reg = " insert into Student(StudentID, FName, Lname, City, Phone, DateOfBirth, Program) values ('$studentid', '$firstname',
  mysqli_query($con, $reg);
  echo "<script>
  alert ('Registration Successfull. Please login');
  window.location.href = 'data.html';
  </script>";
}

```

Once you login into the website, you're taken to a student home page. This page is an informational page, containing commonly asked questions from students with their respective answers from the university. On this page, we executed our 10 views from phase two of our project. This page contains a navigational menu, which takes you to specific sections based on the information you want, such as course information or grades. When you want to see a particular view, you click on the view button, which will display the information either in a table or sentences.



```
<form method="POST">

<div id="CourseInformation">

<h2>Course Information</h2>
<p>In this section you can access information about the courses that are offered</p>
<p>Remember to click the view button to display the table for the information of your liking</p><br>

<div id="view5">
<p>What is the exam schedule for the in-person classes? </p>
<p>Make sure you plan your trip to school accordingly on the day of your exam: </p>
<?php
if(isset($_POST['button5'])) {
    require('view5.php');
}
?>
<br>
<input type="submit" name="button5" value="View"/> <br> <br>
</div>

<div id="view9">
<p>What type of delivery method would the instructors be offering? </p>
<p>Here's the list of the instructors and their method of delivery: </p>
<?php
if(isset($_POST['button9'])) {
    require('view9.php');
}
?>
<br>
<input type="submit" name="button9" value="View"/> <br> <br>
</div>
```

Now this report will go over a few of the views and how they are implemented. The first thing we had to do was establish a connection to the database. Then, we had to execute SQL statements depending on the views. For example, one view that displays information in a table, is a view that allows the students to see the exam schedule for in person classes. It gives information about in-person courses, ordering the information based on the exam dates. To do this, the information is placed in a result variable. If the result for the information has more than 0 rows, it displays the results in a table. An example of a view that displays information using sentences is a view that tells a student if they made the dean's list. It gives information about the courses the student took and if their grade for the course is above an 80. We once again used a result variable which will check if the rows of information are greater than 0. Since the information for this view doesn't include many rows, we displayed the information using a sentence instead of a table.

Course Information

In this section you can access information about the courses that are offered

Remember to click the view button to display the table for the information of your liking

What is the exam schedule for the in-person classes?

Make sure you plan your trip to school accordingly on the day of your exam:

CourseID	CourseName	CourseDelivery	Capacity	ExamDates	InstructorID
ENG3850	Engineering Economics	In Person	140	2021-12-12 00:00:00	20001
ENG3200	Intro to Engineering	In Person	140	2021-12-15 00:00:00	20001
NUR1000	Nursing Principles	In Person	100	2021-12-15 00:00:00	20006

View

```
#!/usr/bin/php
$dbhost = "localhost";
$dbuser = "root";
$dbpass = "";
$db = "FinalProject";

$conn = mysqli_connect($dbhost, $dbuser, $dbpass, $db);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT * FROM Courses WHERE CourseDelivery = 'In Person' ORDER BY ExamDates";
$result = mysqli_query($conn, $sql);

if ($result->num_rows > 0) {
    echo "<table><tr><th>CourseID</th><th>CourseName</th><th>CourseDelivery</th><th>Capacity</th><th>ExamDates</th><th>InstructorID</th></tr>";
    // output data of each row
    while($row = mysqli_fetch_assoc()) {
        echo "<tr><td>". $row["CourseID"]. "</td><td>". $row["CourseName"]. "</td><td>". $row["CourseDelivery"]. "</td><td>". $row["Capacity"]. "</td><td>". $row["ExamDates"]. "</td><td>". $row["InstructorID"]. "</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}

mysqli_close($conn);
?>
```

How can I see if I am on the dean's list?

The dean's list has students who obtain an average of a grade A and higher

Here's the courses where you got a grade of A and higher:

The student: 10001 for the course: ENG3200 got a grade of: 83

View

```
#!/usr/bin/php
$dbhost = "localhost";
$dbuser = "root";
$dbpass = "";
$db = "FinalProject";

$conn = mysqli_connect($dbhost, $dbuser, $dbpass, $db);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT Grades.StudentID, NumGrade, CourseID
FROM Grades
WHERE NumGrade >= 80
AND StudentID IN (SELECT StudentID
FROM Student
WHERE StudentID = 10001)";

$result = mysqli_query($conn, $sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc()) {
        echo "The student: " . $row["StudentID"]. " for the course: " . $row["CourseID"]. " got a grade of: " . $row["NumGrade"]. "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);
?>
```

Observations and Analysis

Our project is catered toward communicating data between clients, which in this case is the school and its students and the faculty. This will allow both parties to make the most informed decision that is necessary to protect their interests. By establishing surveys for students and faculty alike, we can create multiple data tables that will provide answers to questions that users have. For example, a potential use of our project can be for the dean of a program to use it to find the students that have excelled courses or multiple courses so they can be eligible for awards and scholarships like the “Dean’s List”.

We can see our University Database system being used by multiple universities as it provides an easy, quick, and convenient way to answer both student and faculty questions alike. Many students are bound to have questions about their university, as well as the courses they are taking, their grades, how they compare with the rest of their program, etc. With our system, we can provide all the answers to those questions, without having the need for students or faculty to delve into further research.

Some obstacles our design can face will be the physical limitations of the database, conflicts when creating keys and relationships. The physical limitation of the system will be based on the amount of space that the server is allocated and the amount of traffic it will experience. The conflict between keys can also occur if primary keys and foreign keys are not clearly defined, causing errors to occur in the database program, resulting in the program crashing. However, our design has vast potential once these obstacles are overcome, as we can expand the front end to large amounts and create many queries for any possible scenario. If the design is in use for large amounts of time, the massive amount of data will be cleanly categorized and made into tables to address any scenario both students and faculty can think of.

Conclusion

Throughout the Project, we had quite a few implications here and there but nothing that wasn't achievable. While we were making our ER Diagram, we had to think with a broad view to make an application that would be able to support all faculty at a university, with useful views that would provide students and faculty convenience to easily access readily available information.

Our first implication would be creating our tables and properly choosing distinct primary keys in each table, as well as identifying foreign keys in SQL. After our tables were created, we had some difficulty creating our five custom views needed for the website. The problem arose when trying to think of convenient and useful views that would be beneficial to all students and faculty alike.

In addition, when creating the web page, implementing the views onto it was tedious work as we had to iterate through the same steps repeatedly for each view. Our last concern was the visual appeal of the website. We came across some difficulties with the CSS of the website and the formatting, but nothing too troublesome for us to handle.

Some future work that we can add onto the system to improve it would be a more appealing login and home page for the users. Visuals that would represent their university, and have external links to more relevant information about their university, courses, school life, resources, etc.

Throughout this project, we all grasped a stronger understanding of MYSQL and database systems. We learned how to create specific views for our database that would provide easy access to convenient and useful information. We also had a beneficial refresher on HTML and PHP, as we had a few struggles in the beginning since it has been a lengthy period since we had touched upon it. Overall, this project strengthened our knowledge, and we are confident we can create a more complex and sophisticated database with the knowledge we have acquired.

References

“HTML introduction,” Introduction to HTML. [Online]. Available: https://www.w3schools.com/html/html_intro.asp.

“How to make login & registration form in PHP and MySQL, Create Signin & Signup Page,” YouTube, 15-Aug-2018. [Online]. Available: <https://www.youtube.com/watch?v=NXAHkqiIepc&list=LL&index=1&t=853s>. [Accessed: 28-Nov-2021].

“How to download and install xampp on mac OS 2021,” YouTube, 25-Aug-2020. [Online]. Available: <https://www.youtube.com/watch?v=Zo5gGr0DWhg&t=425s>. [Accessed: 28-Nov-2021].

B. Ayodeji, “How to connect mysql with PHP,” Medium, 26-Aug-2018. [Online]. Available: <https://medium.com/@bolajiyodeji/how-to-connect-mysql-with-php-f99d4ec1eff3>. [Accessed: 28-Nov-2021].

A. Ahmad, N. H. and P. Jain, “Including database connection file in PHP,” Stack Overflow, 01-Feb-1962. [Online]. Available: <https://stackoverflow.com/questions/20787386/including-database-connection-file-in-php/20787414>. [Accessed: 28-Nov-2021].

Prasanth, Y. Prajapati, M. A. Zafar, and E. L. Ramos, “Display alert message and redirect after click on accept,” Stack Overflow, 01-Oct-1960. [Online]. Available: <https://stackoverflow.com/questions/11869662/display-alert-message-and-redirect-after-click-on-accept/11869779>. [Accessed: 28-Nov-2021].

I. Ali, “How to connect mysql databases with PHP websites,” The Official Cloudways Blog, 24-Sep-2021. [Online]. Available: <https://www.cloudways.com/blog/connect-mysql-with-php/>. [Accessed: 28-Nov-2021].

PHP MySQL Select Data. [Online]. Available: https://www.w3schools.com/php/php_mysql_select.asp. [Accessed: 28-Nov-2021].

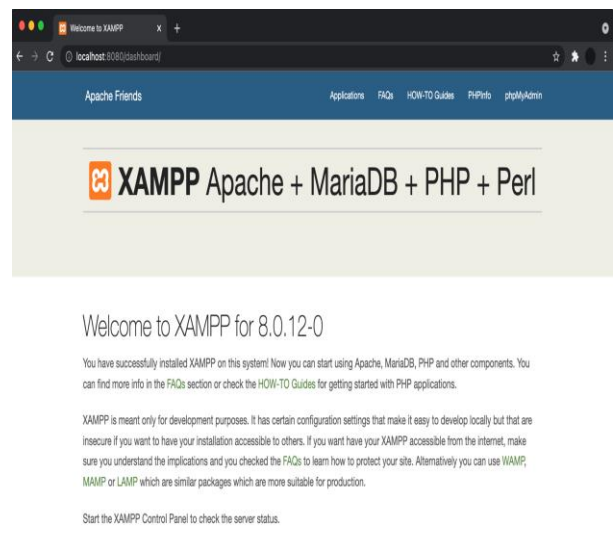
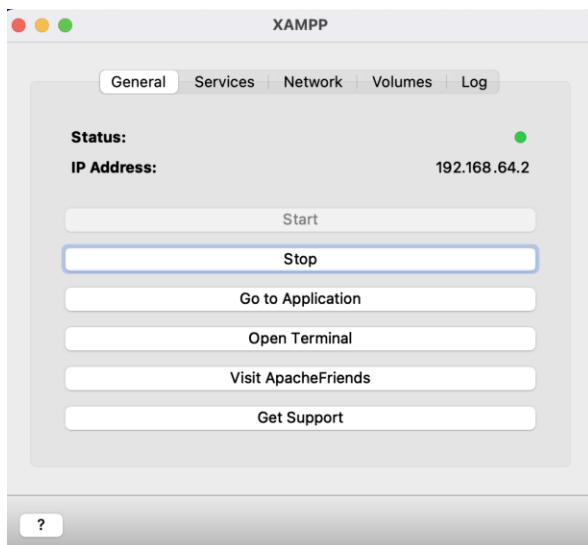
“How to call PHP function on the click of a button ?,” GeeksforGeeks, 23-Aug-2019. [Online]. Available: <https://www.geeksforgeeks.org/how-to-call-php-function-on-the-click-of-a-button/>. [Accessed: 28-Nov-2021].

Appendices

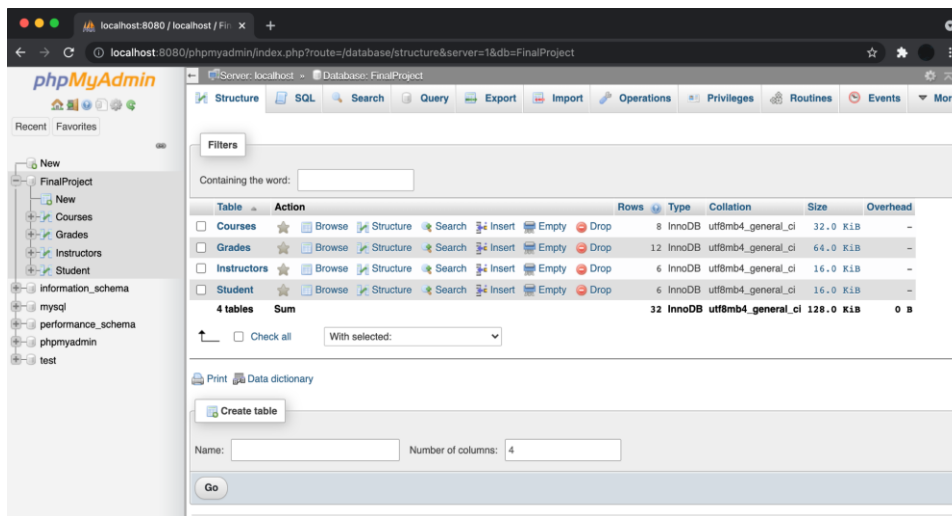
GITHUB LINK: <https://github.com/Rayyan1023/Data-Management-Systems-Final-Project>

Installation of University Database System:

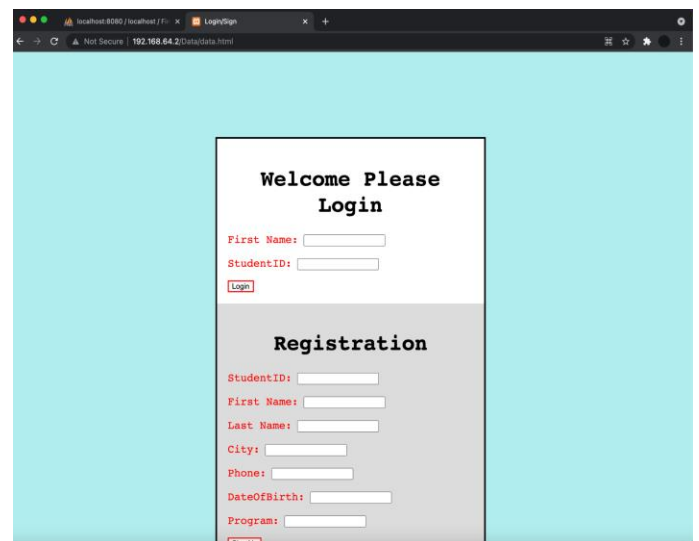
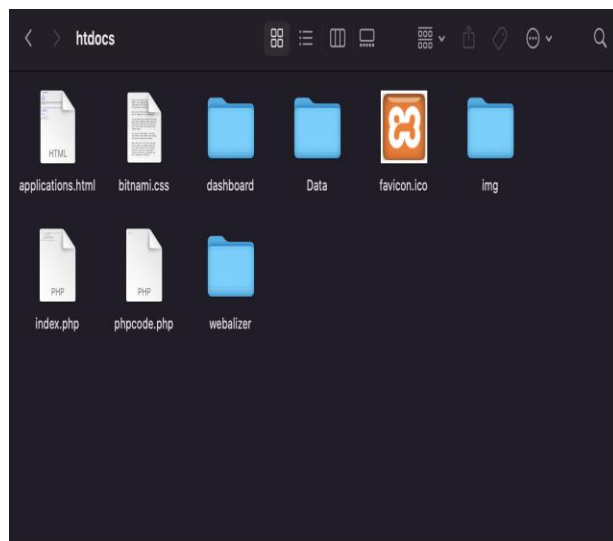
Step 1: Download and install XAMPP onto your device. Once it's installed, start the application, and check to see if all services like (MySQL, ProFTPD, and Apache) are running in the services tab. Next you need to enable the network connection to localhost:8080 in the network tab. Finally, you need to mount the directory in the volume tab. To check if XAMPP is running, the status will be green in the general tab and when you open the XAMPP website in the browser of your choice by typing localhost:8080 you will be met with a welcome page.



Step 2: Create and implement the database using PhpMyadmin. For this step you need to make sure you follow all the appropriate naming conventions used for the schema and tables. When logging into PhpMyadmin the username needs to be root and you should keep the password blank. Then you create the schema naming it FinalProject. Next you need to create the tables Student, Courses, Grades, and Instructors. After creating the tables, you need to add the columns based on the information in the relational schema. Make sure you follow phase two of the report to create the primary and foreign keys for the tables. Finally, you need to import and transfer the sample data from the excel sheet provided into the database.



Step 3: Download the code and run the website. First you need to download the code from our GitHub, the folder should be called Data. Next you need to locate the htdocs folder in the Iampmp directory. Once you find it you need to place the code folder into the htdocs folder. To run the project, you need to type the IP address given to you at the general tab of the XAMPP application. The default one is 192.168.64.2 and then you need to write the file you're trying to run in the htdocs folder. For our project it should look like this: <http://192.168.64.2/Data/data.html>. Now you have successfully completed all the steps to install and run or project.



Project Phase II: University Database System: List of views and descriptions:

View 1 - Joins Students, Grades and Courses table together while displaying StudentID, NumGrade and CourseID Points.

View 2: Displays any student who has a higher grade than the lowest grade for student “10001” which is 74

View 3: Displays the Capacity, InstructorID and CourseName for the instructors who have a higher total capacity than instructor “20001” which is 380.

View 4: Displays Left, Right Join and Union to emulate a Full Join in MySQL Workbench. Displays course delivery and last name by combining Courses table and Instructors table.

View 5: Displays emulation of an Intercept in MySQL Workbench by using IN. Displays StudentID, NumGrade and CourseID where the student with id “10001” got a higher grade than 80.

View 6: Displays the Grades table for the students who have a grade higher than 50. It orders the table by the StudentID. This view is useful to our project because it helps us to see which students are passing the courses.

View 7: Gets information from the Student table and if they live in the city Toronto or Mississauga the data is ordered by their first name. This view is key to our project because it allows us to see which students need to commute to classes.

View 8: This view gets information from the Courses table and displays the in-person classes with a capacity greater than 100. This view is useful because we can see which classes are in person and how many are attending with new covid laws and restrictions.

View 9: Checks the total capacity under the Courses table where the classes are online. It’s useful to see which classes are online and determine the number of students attending.

View 10: Gets information from the Courses table and displays the in-person classes, ordering them by the exam dates. This view is useful for students that need to go in on the dates published.

Contribution Matrix

Names	Tasks Completed	Contribution Percentage
Rayyan Mohammed	Contributed to Phase 1, Phase 2, and Phase 3. As well as the final report	100
Daniyal Khan	Contributed to Phase 1, Phase 2, and Phase 3. As well as the final report	100
Mohammad Khalid	Contributed to Phase 1, Phase 2, and Phase 3. As well as the final report	100
Yousif Sarmad	Contributed to Phase 1, Phase 2, and Phase 3. As well as the final report	100
Charles Olagunju	Contributed to Phase 1 and 3.	50

We recorded our project timeline and task completion in our discord group chat.

Sep 27 - This was our first meeting where we introduced ourselves and decided on a project topic. Zaeem was selected as a team leader, and he sent an email to the TA including our names and project topic.

Oct 15 - We met on a call and worked together on the project proposal. Everyone was assigned a paragraph. We gave everyone a one-day deadline to complete the paragraph.

Oct 16 - We all met on call again and reviewed our proposal by editing it and discussing how we would implement the database. Zaeem submitted the proposal at the end of our meeting.

Oct 28 - On this day we met and divided the tasks for phase II of our project. We gave a deadline of 5 days to complete the tasks

Task distribution:

Yousif oversaw the Relational Schema.

Rayyan oversaw the Sample Data.

Daniyal oversaw the Views.

Zaeem oversaw the ER Diagram.

When anyone needed help with a task all of us were there to help.

Nov 2 - This was the deadline to complete the tasks. We met on a call and discussed any issues we faced during our tasks. We presented all our findings and inputted the tasks into our phase II doc. We were given 2 days to look over and edit the doc.

Nov 4 - We all joined a call, did a final review of the doc and Zaeem submitted phase II. We also discussed the task distribution for phase III of the project. We set a hard deadline of Nov. 20 and had two weekly meetings where we discussed and worked on the project together.

Daniyal and Zaeem oversaw completing the code.

Yousif and Rayyan oversaw the presentation.

We distributed the sections of the final report equally.

Nov 20 - We all met on call where we discussed any challenges and issues we faced while completing phase III of the project. We made a deadline till Nov. 27 where we can edit the final report. We also met and rehearsed the presentation.

Nov 28 - This was our final meeting where we did a final review and submitted phase III of the report