fXAVIER UNIVERSITY – ATENEO DE CAGAYAN

COLLEGE OF COMPUTER STUDIES
DEPARTMENT OF INFORMATION TECHNOLOGY

Web Application for Student Information Management

A Project Presented to the Faculty of the
Department of Information Technology
College of Computer Studies

In Partial Fulfillment of the Requirement for the Course
ITCC 14 A – Integrative Programming and Technologies

Submitted by:

Niznik Marie C. Alamban
Rey Christian C. Llamas
Kathleen G. Pacanut

ITCC 14

Submitted to:

Sir Jessie Christopher Lagrosas

September 10, 2020

Table of Contents

Chapter 1
Introduction

1.1 Background of the Study

The Office of Student Affairs, also known as OSA, is a division of services and support for college students to enhance their growth and development. The office is in charge of providing services and support for students outside of the classroom. They are also in charge of advising student volunteers to plan, organize, implement, and evaluate programs; supervise major campus events, including safety and security, crowd control, and compliance with contractual obligations and University policies.

OSA utilizes a system which can update and monitor the record of students and various transactions or activities done by the students for use of the University facilities, lost and found items, and violations they have made. This paper discusses a web application system that will allow OSA to continuously view, add, and edit student information and communicate with the registrar's office to verify the status of a student. The purpose of verifying the status of the student is to check if the student is indeed currently enrolled in the University or not. Once this is verified, the office then proceeds with processing a request for a certificate of good moral character and recording violations done by students. More details about the web application will be discussed in the succeeding parts of the paper.

1.2 Conceptual Framework

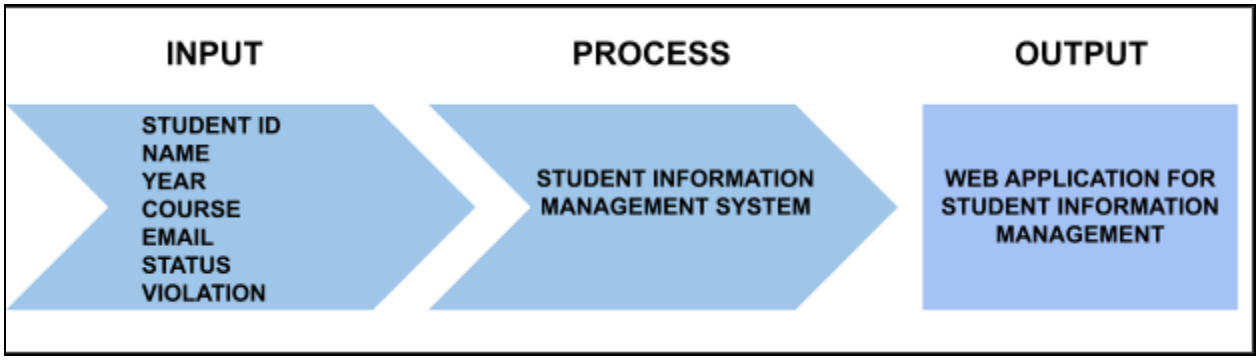

Figure 1. Conceptual Framework of the Proposed Web Application

The web application for student information management is developed by the third year Information Technology students of Xavier University - Ateneo de Cagayan as part of the requirements for the subject ITCC 14 - Integrative Programming and Technologies. The web application is designed to aid the Office of Student Affairs to have a more efficient process of viewing and updating student records and allowing the registrar's office to communicate with the web application to identify students with violations while the office, on the other hand, will ask for verification for the status of a student.

1.3 Statement of the Problem

The Office of Student Affairs has been operating in the University to provide students the proper services that they need as well as to guide their formation during their stay in the University. This is to maintain an atmosphere which is good for the personal development and studies of the students and to facilitate an active and exciting life in the school. They follow a system in order to proceed with their day to day operation. However, there are still parts of the system that are still in need of improvement. In line with this, our proposed web application aims to solve the following problems:

1. Absence of real time updates from the registrar's office.
2. Inefficient method of updating and monitoring students' records.

1.4 General Objectives

This project seeks to accomplish the following objectives:

- Create a web application to monitor the records of the students for the Office of Student Affairs.
- Apply the skill sets acquired from ITCC14:Integrative Programming and Technologies.
- Implement the algorithms, data, and the structural design of the overall web application.

1.4.1 Specific Objectives

This project seeks to accomplish the following objectives:

- To develop, design and implement a web app system that can handle and manage the records of the students for the Office of the Students Affairs.
- To design, develop and implement a web app system that can be accessed by the Office of Student Affairs and Registrar's Office.
- Provide real time monitoring of the records of the students.

1.4.1.1 Requirements Determination

It is essential for this project to determine the requirements that would satisfy the needs of the users. We have observed how the Office of Student Affairs conducts their transactions and we identified the areas that are in need of improvement in order to make their day to day operations more efficient. The user requirements we have determined will be the basis for the web application we will be creating.

In order to gather more information we needed for this project, we also asked a student assistant who works at the office to answer some questions that would help us determine the needs of the office. These are the questions we asked and the corresponding answers by the student assistant:

1. How does the Office of Student Affairs function?
    - *The office is in charge of processing the Good Moral Certificate of students. We are also in charge of insurance and permits as well as recording student withdrawals, violations, and ID.*

2. For recording the violations of students, what are the data or information that the office needs?
    - *Name, year and program, violation, date*

3. Is the office also in charge of monitoring if project heads are already cleared of their projects?
    - *The Student Activities and Leadership Development (SACDEV) Office is in charge of monitoring the status of the project heads. OSA is only in charge of checking the project heads' status during the signing of clearance.*

4. During the signing of clearance, how will the office know if a project head has been cleared by the SACDEV office?
    - *We utilize a shared google sheet for real time monitoring.*

5. What does the Office of Student Affairs use for keeping the records of students and updating them whenever violations are made?
    - *The office uses google sheets.*

6. What do you think is needed by the office to have a more efficient way of monitoring student records?
    - *Another alternative to google sheets that can easily handle a lot of data. At the same time, it should also be user friendly.*

1.4.1.2 System Analysis

Currently, the Office of Student Affairs keeps the student records and monitors them as well through google sheets. They do not use any web application or software for the records. Although google sheets still allow them to function for their day to day operations, it is difficult to ensure accuracy through the use of it because the design of the spreadsheet makes it prone for users to commit errors. The fact that the office has a lot of data to handle due to the huge amount of students in the university means that they need a web application that would allow the data they handle to be visible and organized. The advantages of creating a very own web application for the Office of Student Affairs is that the application could be designed according to the school's branding and its features and functions could be designed in a way where they are truly relevant for the office. Through this, the office will have a better means of handling student transactions and data. Thus, resulting in a more efficient day to day operations.

1.4.1.3 System Design

The usage of google sheets as a means for the Office of Student Affairs to monitor and update student information is important for the office to function effectively in providing services to the students and to the university. The following are the data that the office inputs in the spreadsheets: Name, year and program, violation, date.

The disadvantage of the existing system of the office is that it is highly possible for the users to commit an error in inputting data to the spreadsheets. It could also be time consuming because of having multiple spreadsheets. Having all the students' data integrated in one place allows for an improved visibility and real time updates.

1.4.1.4 System Implementation

To implement this project successfully, the website will utilize various programming languages such as HTML, CSS, PHP, and Javascript.

The following data structures and algorithms were used to implement the program:

- HTML for tagging text files to achieve font, color, graphic, and hyperlink effects.
- CSS to format the layout of our web application.
- Javascript for the interactive capability of the web system.
- SQL serves as the database and to record the information of the students.
- PHP to format the web application better.

1.4.1.5 Testing and Evaluation

Testing and evaluation must be done in order to determine if the proposed web application has achieved the expected output and has satisfied the needs of the users. Furthermore, it will help in the correction of errors for the proposed web application.

The following web application testing techniques can be conducted to ensure its quality:

1. Functional Testing
   This is done to check if the web application works as the user intended for and fulfills the requirements recorded in the developer's documentation. The following tasks should be done: Test UI workflows, test hyperlinks, validate input fields, test cookies, and validate HTML. CSS and XPath.

2. Usability Testing
   Usability testing is done to check the user experience while navigating through the web application. While the traditional testing is carried out by a developer, designer or project manager, a feedback from the end user is also done to avoid any bias.

3. Interface Testing
   Interface testing is important to ensure a great user experience. Application, web server, and database are mainly the three areas that a tester should target.

4. Compatibility Testing
   Compatibility testing confirms the compatibility of the website design across different browsers and also on a variety of devices. It conducts device compatibility testing and browser compatibility testing.

5. Performance Testing
   Performance testing aims to verify the application response time and throughput under various load conditions.

1.4.1.6 System Installation

System Installation will take place after the web application has been tested, evaluated, and confirmed to be trusted and reliable. This requires the preparation of the system for installation to be used by the Office of the Students Affairs.

1.5 Originality/Significance of the Study

The web application was designed to update and monitor students' records and the violations they have done, if there is any. This project will benefit the following:

To the Staff of the Office of Student Affairs. With the use of the web application, they can continuously view, update, and delete information; and communicate with the registrar's office to verify a student's status. Thus, allowing them to have more efficient day to day operations.

To the Students. The students will be accommodated faster in the Office of Student Affairs in case they have any requests or transactions to do in the office. Their data will also be handled more accurately since the proposed web application's front end is designed to avoid less errors in inputting data.

To the Future Researchers/Developers. This paper can also benefit future researchers and developers who wish to create this kind of web application. They will be able to use this paper as a basis or reference to their research and figure out how to broaden the study even more and improve it.

1.6 Scope and Limitation

The web application is created for the admins, staff and the student assistants to monitor and update (add, edit, and delete) the students' records in the Office of Student Affairs. The web application is only limited to these functionalities. Furthermore, the web application can ensure that the authorized persons are the only ones who can use the app. It will also allow the office to

connect with an API from the registrar's office which will check the status of a student. These functionalities of the web application can help the staff be more productive with their work. The prime reason why the developers aim to develop and implement a web application that will handle the students' records is because the Office of Student Affairs only uses papers and google sheets to keep the records of the students.

- **Administrator, Student-In-Charge, Staff** - These are the only authorized users that are allowed to access the web application. A username and password will be asked in the login page for authentication. The administrator is in charge of monitoring and maintaining the web application while the student-in-charge, and staff of the Office of Student Affairs can access and utilize the functionalities of the web application. In addition, the web application is only limited to the use of one account. So, this means that all of the aforementioned users will only be able to see the same contents of the web application.

Data for Students' Records

- Name List
- Year
- Course
- Email
- Status
- Violation Report

Security

- Password is hidden
- Anyone in the Office of Student Affairs can access it with the same username and password.

Other limitations of the Web Application

- It is not yet capable of showing whether a student handling a project in his organization is cleared of his project at the Student Activities & Leadership Development Office (SACDEV).
- The web application must have different user types and their accessibility to the functions of the web application should depend on their user type.

1.7 Definition of Terms

Google Sheets - it is a spreadsheet program which is a software created by Google Services.
Real time - this is the actual time where the action takes place or the event occurs.
Records - this is used to store or to keep track of the person's information, conduct or performance.
System - a method where things should work together to make this procedure work.
Transactions - an act of buying, selling, or a deal with something.

User Types - classify users, and may also confer roles that grant access to protected areas of the website.
Web application - it is an computer application software where it uses the web browsers to perform

Chapter 2
Construction Phase

2.1 Requirements Determination

Requirements determination was conducted to make the desired web application that will aid the end users. The researchers were able to achieve the following requirements:

- **Monitoring**
  - The client would be able to monitor the students' records and edit their information if needed. In case of violations made by the students, the users can update the student's record. The users can also access the students' data from the registrars' office. If a student is listed in the database of the registrar's office, then that student is currently enrolled in the university.
- **Data/Information Privacy**
  - Personal Information of the students are protected in compliance with the Data Protection Act.
- **Security**
  - Authorized person can login through the web app using the given username and password.
- **Operational**
  - The web app can run on different browsers given that the PC is connected to the internet.

2.2 System Analysis

The proposed web application provides better service for the staff, teachers, and student-in-charge in the Office of Student Affairs. The main purpose of creating the proposed system is to monitor the students' records. Through this, they will be able to utilize a better designed application and commit less errors in inputting data. They only need to login to the web app to keep track of the students' records. The web app's database is safely stored and is regularly updated and maintained. Only authorized persons are capable of using the proposed web app system with the given username and password.

2.3 System Design

The design of the system consists of the following:

- **API**
  - It allows the web application to communicate with the registrar's application to check if a student is listed in their database. If the student's name can be found in their database, then that student is currently enrolled in the university.
- **Monitor/Edit Records**
  - The user can view, add, edit, and delete the information of the students.
- **User**
  - The web app has a user authentication to avoid unauthorized persons from accessing the information of the students.

2.4 System Implementation

The code snippets shown below will enable the users of the web application to view, add, delete and update the students' information as well as connect to the registrar's office database to check whether a student is indeed currently enrolled in the university or not.

Figure 2. add_students.php

```php
<?php
include 'db_connect.php';
$pdo = pdo_connect_mysql();
$msg = '';
// Check if POST data is not empty
if (!empty($_POST)) {
    // Post data not empty insert a new record
    // Set-up the variables that are going to be inserted, we must check if the POST variables exist if not we can default them to blank
    $student_id = isset($_POST['student_id']) && !empty($_POST['student_id']) && $_POST['student_id'] != 'auto' ? $_POST['student_id'] : NULL;
    // Check if POST variable "name" exists, if not default the value to blank, basically the same for all variables
    // $student_id = isset($_POST['student_id']) ? $_POST['student_id'] : NULL;
    $name = isset($_POST['name']) ? $_POST['name'] : '';
    $email = isset($_POST['email']) ? $_POST['email'] : '';
    $year = isset($_POST['year']) ? $_POST['year'] : '';
    $is_enrolled = isset($_POST['is_enrolled']) ? $_POST['is_enrolled'] : '';
    $course = isset($_POST['course']) ? $_POST['course'] : '';
    $violation = isset($_POST['violation']) ? $_POST['violation'] : '';
    // Insert new record into the contacts table
    $stmt = $pdo->prepare('INSERT INTO students VALUES (?, ?, ?, ?, ?, ?, ?)');
    $ret = $stmt->execute([$student_id, $name, $email, $year, $course, $is_enrolled, $violation]);
    // Output message
    // var_dump($ret);
    if($ret == 'true'){
        echo "<script>
        alert('Successfully Added Student');
        window.location.href='get_students.php';
        </script>";
    }else{
        echo "<script>
        alert('Please try Again later!');
        </script>";
    }

```

```
}
?>


<!DOCTYPE html>
<html>
<head>
path/to/font-awesome/css/font-awesome.min.css
<link rel = "stylesheet" href = "assets/style.css">
<!-- <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"> -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> Xavier University - Office of Student Affairs </title>
</head>
    <div class = "header_2">
    <img src = "assets/XU Logo.png" id="leftImageXU" />

    </div>
<div class="footer">
  <p> <footer>&copy;2020</footer> </p>
  <p> <footer>ALL RIGHTS RESERVED</footer> </p>
</div>
</div>
<body>

</body>

<br>
<br>
<br>
<br>
<br>
<h1 style="text-align:center">STUDENT RECORDS</h1>
<hr>

<div class="container">
<form action="add_students.php" method="post">
        <label for="id">ID</label>
        <input type="text" name="student_id"  id="student_id" required>
        <label for="name">Name</label>
        <input type="text" name="name"  id="name" required>
        <label for="email">Email</label>
        <input type="text" name="email"  id="email" required>
        <label for="Year">Year</label>
        <input type="text" name="year"  id="year" required>
        <label for="Course">Course</label>
        <input type="text" name="course"  id="course" required>
        <label for="Is Enrolled?">Is Enrolled?</label>
        <select style="width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;" name="is_enrolled" id="is_enrolled">
            <option value=" ">----- PLEASE SELECT -----</option>
            <option value="Yes">Yes</option>
            <option value="None">Not yet</option>
        </select>
        <!-- <input type="text" name="is_enrolled" placeholder="1" value="<?=$student['is_enrolled']?>" id="is_enrolled"> -->
        <label for="Violation">Violation</label>
        <div class="select_inp">
        <select style="width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;" name="violation" id="violation">
            <option value=" ">----- PLEASE SELECT -----</option>

            <option value="Yes">Yes</option>
            <option value="None">None</option>
        </select>
        </div>
        <!-- <input type="text" name="violation" placeholder="1" value="<?=$student['violation']?>" id="violation"> -->
        <input type="submit" value="Add">
    </form>
    <?php if ($msg): ?>
    <p><?=$msg?></p>
    <?php endif; ?>
</div>

</body>
</html>
```

Figure 3. api.php

```php
<?php


header("Content-Type:application/json");

include 'db_connect.php';
$pdo = pdo_connect_mysql();

$stmt = $pdo->prepare('SELECT * FROM students');
$stmt->execute();
$students = $stmt->FetchAll(PDO::FETCH_ASSOC);

echo json_encode($students);

?>
```

Figure 4. db_connect.php

```php
<?php
function pdo_connect_mysql() {
$servername = "localhost";
$username = "root";
$password = "";

try {
  $conn = new PDO("mysql:host=$servername;dbname=student_db", $username, $password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
  // echo "Connected successfully";
  return $conn;
} catch(PDOException $e) {
  echo "Connection failed: " . $e->getMessage();
}
}
?>
```

Figure 5. delete_students.php

```php
<?php
include 'db_connect.php';
$pdo = pdo_connect_mysql();
$x;
if(isset($_GET['student_id']))
{
  $student_id=$_GET['student_id'];
  $stmt=$pdo->prepare("delete from students where student_id='$student_id'");
  $stmt->execute();
  if($stmt)
    {
        echo "<script>
        // alert('Successfully Deleted!');
        window.location.href='get_students.php';
        </script>";
    }
}

?>
```

Figure 6. get_students.php

```php
<?php
include 'db_connect.php';
$pdo = pdo_connect_mysql();
$page = isset($_GET['page']) && is_numeric($_GET['page']) ? (int)$_GET['page'] : 1;
$records_per_page = 5;


$stmt = $pdo->prepare('SELECT * FROM students');
// $stmt = $pdo->prepare('SELECT * FROM students ORDER BY id LIMIT :current_page, :record_per_page');
// $stmt->bindValue(':current_page', ($page-1)*$records_per_page, PDO::PARAM_INT);
// $stmt->bindValue(':record_per_page', $records_per_page, PDO::PARAM_INT);
$stmt->execute();
// Fetch the records so we can display them in our template.
$students = $stmt->fetchAll(PDO::FETCH_ASSOC);

?>

<!DOCTYPE html>
<html>
<head>
<link rel = "stylesheet" href = "assets/style.css">
<link rel = "stylesheet" href = "assets/font-awesome/css/font-awesome.min.css">
<!-- <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"> -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> Xavier University - Office of Student Affairs </title>
</head>
    <div class = "header_2">
    <img src = "assets/XU Logo.png" id="leftImageXU">
    <a href="rest_con.php" class="button" style="color: white;
    padding-left: 50px;
    padding-bottom: 20px;"> Registrar's Student Data </a>
    <a href="http://localhost/final_proj/" class="button" style="color: white;
    padding-left: 1250px;
   padding-bottom: 20px;" >LOG OUT</a>

   </div>

<div class="footer">
  <p> <footer>&copy;2020</footer> </p>
  <p> <footer>ALL RIGHTS RESERVED</footer> </p>
</div>
</div>
<body>

</body>

<br>
<br>
<br>
<br>
<br>
<h1 style="text-align:center; ">STUDENT RECORDS</h1>
<div style="text-align: right;">
<a style="background-color: white;
  color: black;
  border: 2px solid #555555;  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
  " id="add_buttn" href="add_students.php" type="submit" ><i class="fa fa-plus-square" style="font-size:35px;color:black"></i></a>
  </div>
<hr>
<div class="container">
```

```
<table>
        <thead>
            <tr>
                <td>Student ID</td>
                <td>Name</td>
                <td>Year</td>
                <td>Course</td>
                <td>Email</td>
                <td>Enrolled</td>
                <td>Violation</td>
                <td>Action</td>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($students as $student): ?>
            <tr>
                <td><?=$student['student_id']?></td>
                <td><?=$student['name']?></td>
                <td><?=$student['year']?></td>
                <td><?=$student['course']?></td>
                <td><?=$student['email']?></td>
                <td><?=$student['is_enrolled']?></td>
                <td><?=$student['violation']?></td>
                <td>
                <a href="update_students.php?student_id=<?=$student['student_id']?>"><i class="fa fa-pencil-square-o" style="font-size:35px;color:black"></i></a>
                <a href="delete_students.php?student_id=<?=$student['student_id']?>" onclick="return confirm('Are you sure to delete this student?')"><i class="fa fa
            </td>
            </tr>
            <?php endforeach; ?>
        </tbody>
    </table>
```

```
</body>
</html>

<script>
function confirmationDelete(anchor)
{
    var conf = confirm('Are you sure want to delete this record?');
    if(conf){
        window.location=anchor.attr("href");
    }else{
        window.location="get_students.php";
    }
}
 </script>
```

Figure 7. index.php

```php
<?php
include 'db_connect.php';
// include 'assets/style.css';

$pdo = pdo_connect_mysql();
$page = isset($_GET['page']) && is_numeric($_GET['page']) ? (int)$_GET['page'] : 1;
$records_per_page = 5;

// var_dump($pdo);
$stmt = $pdo->prepare('SELECT * FROM students');
// $stmt = $pdo->prepare('SELECT * FROM students ORDER BY id LIMIT :current_page, :record_per_page');
// $stmt->bindValue(':current_page', ($page-1)*$records_per_page, PDO::PARAM_INT);
// $stmt->bindValue(':record_per_page', $records_per_page, PDO::PARAM_INT);
$stmt->execute();
// Fetch the records so we can display them in our template.
$contacts = $stmt->fetchAll(PDO::FETCH_ASSOC);



?>

<!-- <?php foreach ($contacts as $contact): ?>
<h1><?=$contact['student_id']?></h1>
<?php endforeach; ?> -->

<!DOCTYPE html>
<html>
<head>
<link rel = "stylesheet" href = "assets/style.css">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> Xavier University - Office of Student Affairs </title>
</head>
<div class = "header">
```

```
<img src = "assets/XU Logo.png" id="leftImageXU" />
        <div class="login-container">
                <form action="login.php" method="post">
                        <input type="text" placeholder="Username" name="username" >
                        <input type="password" placeholder="Password" name="password" >
            <input type="submit" value="Login">
            <!-- <a id="log_a" href="get_students.php" type="submit" value="Login">Login</a> -->
                </form>
        </div>
</div>
<!-- <div class = "header2"> -->
<!-- <body bgcolor = "#FFFFFF"> -->
<div class="cont">
    <img src = "assets/XU1.jpg" id="centerImageXU"/>
</div>
<div class="footer">
  <p> <footer>&copy;2020</footer> </p>
  <p> <footer>ALL RIGHTS RESERVED</footer> </p>
</div>
</div>
</body>
</html>
<!--
<!DOCTYPE html>
<html lang="en">
<head>
<title>CSS Template</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel = "stylesheet" href = "assets/styles.css">
</head>
<body>
    <div class="fixed-header">
        <div class="container">
            <nav>
                <a href="#">Home</a>
                <a href="#">About</a>
                <a href="#">Products</a>
                <a href="#">Services</a>
                <a href="#">Contact Us</a>
            </nav>
        </div>
    </div>
    <div class="container">
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper
        <p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at matti
    </div>
    <div class="fixed-footer">
        <div class="container">Copyright &copy; 2016 Your Company</div>
    </div>
</body>
</body>
</html> -->
```

Figure 8. login.php

```php
<?php
include 'db_connect.php';
$pdo = pdo_connect_mysql();

// var_dump($_GET['student_id']);
if (!empty($_POST)) {
        // This part is similar to the create.php, but instead we update a record and not insert
        $username = isset($_POST['username']) ? $_POST['username'] : '';
        $password = isset($_POST['password']) ? $_POST['password'] : '';
        $stmt = $pdo->prepare('SELECT * FROM users WHERE username = ? AND password = ?');
        $stmt->execute([$_POST['username'],$_POST['password']]);
        $student = $stmt->fetch(PDO::FETCH_ASSOC);
        if($student != null){
            echo "<script>
            alert('Hi $student[name]');
            window.location.href='get_students.php';
            </script>";
        }else{
            echo "<script>
            alert('No user found, Try again.');
            window.location.href='index.php';
            </script>";
        }
        // if($ret){
        //     echo "<script type='text/javascript'>alert('false nigga');</script>";
        // }else{
        // echo "<script type='text/javascript'>alert('$msg');</script>";
        // }
    }
    // Get the contact from the contacts table
//     $stmt = $pdo->prepare('SELECT * FROM students WHERE student_id = ?');
//     $stmt->execute([$_GET['student_id']]);
//     $student = $stmt->fetch(PDO::FETCH_ASSOC);
```

```php


// $stmt = $pdo->prepare('SELECT * FROM students');
// $stmt->execute();
// $students = $stmt->fetchAll(PDO::FETCH_ASSOC);

?>
<!-- <h1> <?php echo $student; ?></h1> -->
```

Figure 9. rest_con.php

```php
<?php

$response = json_decode(file_get_contents('http://localhost/reg/api_new.php'));
  // $stud = json_encode($response);
 // $result = '1';

// foreach($response as $res):
// echo $res;
// endforeach;
// var_dump($response);
?>



<!DOCTYPE html>
<html>
<head>
<link rel = "stylesheet" href = "assets/style.css">
<link rel = "stylesheet" href = "assets/font-awesome/css/font-awesome.min.css">
<!-- <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"> -->
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> Xavier University - Office of Student Affairs </title>
</head>
    <div class = "header_2">
    <img src = "assets/XU Logo.png" id="leftImageXU" />
    <a href="get_students.php" class="button" style="color: white;
    padding-left: 50px;
    padding-bottom: 20px;"> Go Back to our Database</a>

    </div>
<div class="footer">
  <p> <footer>&copy;2020</footer> </p>
  <p> <footer>ALL RIGHTS RESERVED</footer> </p>
</div>
</div>
<body>

</body>

<br>
<br>
<br>
<br>
<br>
<h1 style="text-align:center; ">REGISTRAR'S STUDENT RECORDS</h1>
<div style="text-align: right;">
<a style="background-color: white;
  color: black;
  border: 2px solid #555555;  padding: 15px 32px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
  font-size: 16px;
  margin: 4px 2px;
  cursor: pointer;
  " id="add_buttn" href="add_students.php" type="submit" ><i class="fa fa-plus-square" style="font-size:35px;color:black"></i></a>
  </div>
<hr>
<div class="container">
<table>
        <thead>
            <tr>
                <td>Student ID</td>
                <td>Name</td>
                <td>Gender</td>
```

```
                <td>Year and Course</td>
                <td>Contact no.</td>
            </tr>
        </thead>
        <tbody>
            <?php foreach ($response as $student): ?>
            <tr>
                <td><?=$student->Id?></td>
                <td><?=$student->name?></td>
                <td><?=$student->gender?></td>
                <td><?=$student->year_course?></td>
                <td><?=$student->contact?></td>
            </tr>
            <?php endforeach; ?>
        </tbody>
    </table>


</body>
</html>
```

Figure 10. update_students.php

```
<?php
include 'db_connect.php';
$pdo = pdo_connect_mysql();
$page = isset($_GET['page']) && is_numeric($_GET['page']) ? (int)$_GET['page'] : 1;
$records_per_page = 5;
$msg = '';

// var_dump($_GET['student_id']);
if (isset($_GET['student_id'])) {
    if (!empty($_POST)) {
        // This part is similar to the create.php, but instead we update a record and not insert
        $student_id = isset($_POST['student_id']) ? $_POST['student_id'] : NULL;
        $name = isset($_POST['name']) ? $_POST['name'] : '';
        $email = isset($_POST['email']) ? $_POST['email'] : '';
        $year = isset($_POST['year']) ? $_POST['year'] : '';
        $is_enrolled = isset($_POST['is_enrolled']) ? $_POST['is_enrolled'] : '';
        $course = isset($_POST['course']) ? $_POST['course'] : '';
        $violation = isset($_POST['violation']) ? $_POST['violation'] : '';
        $stmt = $pdo->prepare('UPDATE students SET student_id = ?, name = ?, email = ?, year = ?, course = ?, is_enrolled = ?, violation = ? WHERE student_id = ?');
        $ret = $stmt->execute([$student_id, $name, $email, $year, $course, $is_enrolled, $violation, $_GET['student_id']]);
        $msg = 'Updated Successfully!';
        echo "<script>
        alert('$msg');
        window.location.href='get_students.php';
        </script>";
        // if($ret){
        //     echo "<script type='text/javascript'>alert('False nigga');</script>";
        // }else{
        // echo "<script type='text/javascript'>alert('$msg');</script>";
        // }
    }
    // Get the contact from the contacts table
    $stmt = $pdo->prepare('SELECT * FROM students WHERE student_id = ?');
```

```
    $stmt->execute([$_GET['student_id']]);
    $student = $stmt->fetch(PDO::FETCH_ASSOC);
} else {
    exit('No ID specified!');
}

$stmt = $pdo->prepare('SELECT * FROM students');
$stmt->execute();
$students = $stmt->fetchAll(PDO::FETCH_ASSOC);

?>

<!DOCTYPE html>
<html>
<head>
<link rel = "stylesheet" href = "assets/style.css">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title> Xavier University - Office of Student Affairs </title>
</head>
    <div class = "header_2">
    <img src = "assets/XU Logo.png" id="leftImageXU" />

    </div>
<div class="footer">
  <p> <footer>&copy;2020</footer> </p>
  <p> <footer>ALL RIGHTS RESERVED</footer> </p>
</div>
</div>
<body>

</body>

<br>
<br>
<br>
<br>
<br>
<h1 style="text-align:center">STUDENT RECORDS</h1>
<hr>
<div class="container">
<form action="update_students.php?student_id=<?=$student['student_id']?>" method="post">
        <label for="id">ID</label>
        <input type="text" name="student_id"  value="<?=$student['student_id']?>" id="student_id">
        <label for="name">Name</label>
        <input type="text" name="name"  value="<?=$student['name']?>" id="name">
        <label for="email">Email</label>
        <input type="text" name="email"  value="<?=$student['email']?>" id="email">
        <label for="Year">Year</label>
        <input type="text" name="year"  value="<?=$student['year']?>" id="year">
        <label for="Course">Course</label>
        <input type="text" name="course"  value="<?=$student['course']?>" id="course">
        <label for="Is Enrolled?">Is Enrolled?</label>
        <select style="width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;" name="is_enrolled" id="is_enrolled">
            <option value="<?=$student['is_enrolled']?>" selected><?=$student['is_enrolled']?></option>
            <option value=" ">----- PLEASE SELECT -----</option>
            <option value="Yes">Yes</option>
            <option value="None">Not yet</option>
        </select>
        <!-- <input type="text" name="is_enrolled" placeholder="1" value="<?=$student['is_enrolled']?>" id="is_enrolled"> -->
        <label for="Violation">Violation</label>
```

```
100         <div class="select_inp">
101         <select style="width: 100%;
102    padding: 12px 20px;
103    margin: 8px 0;
104    display: inline-block;
105    border: 1px solid #ccc;
106    box-sizing: border-box;" name="violation" id="violation">
107              <option value="<?=$student['violation']?>" selected><?=$student['violation']?></option>
108              <option value=" ">----- PLEASE SELECT -----</option>
109              <option value="Yes">Yes</option>
110              <option value="None">None</option>
111         </select>        </div>
112         <!-- <input type="text" name="violation" placeholder="1" value="<?=$student['violation']?>" id="violation"> -->
113         <input type="submit" value="Update">
114      </form>
115  </div>
116
117  </body>
118  </html>
```

2.5 Testing and Evaluation

The researchers were able to test the web application but unfortunately, was not able to allow the Office of Student Affairs to test the functionalities of the proposed web application. After it was tested, here are the observations of the researchers:

Strengths

- The web application has a user authentication to protect the application's information from unauthorized persons.
- The web application can perform the following functionalities: view, add, edit, and delete student information, and allows connection to the registrar's office database to check the students' status.
- The users can easily navigate through the web application. Its design is simple and the users can easily understand its features and functionalities.

Weaknesses

- The web application does not have other user types. Anyone from the Office of Student Affairs with access to the web application has similar accessibility to its features and functions.
- The web application still lacks the search and sort functions which are also essential for the users to easily view information and save time.

2.6 System Installation

The web application was tested and evaluated if it was able to meet the needs of the Office of Student Affairs. For the application to function, the researchers used PHP which is a widely used general-purpose scripting language that is suited for web development and can be embedded in HTML. Moreover, the researchers also utilized phpmyadmin for the database of the web application. Phpmyadmin is a free software tool used to handle the administration of MySQL over the web. For setting up PHP, the researchers made use of XAMPP. It is a free, easy to install Apache distribution containing MariaDB, PHP, and Perl.

The following pictures below are the front-end and back-end design of the web application.

Front-end

Figure 11 . Login Page

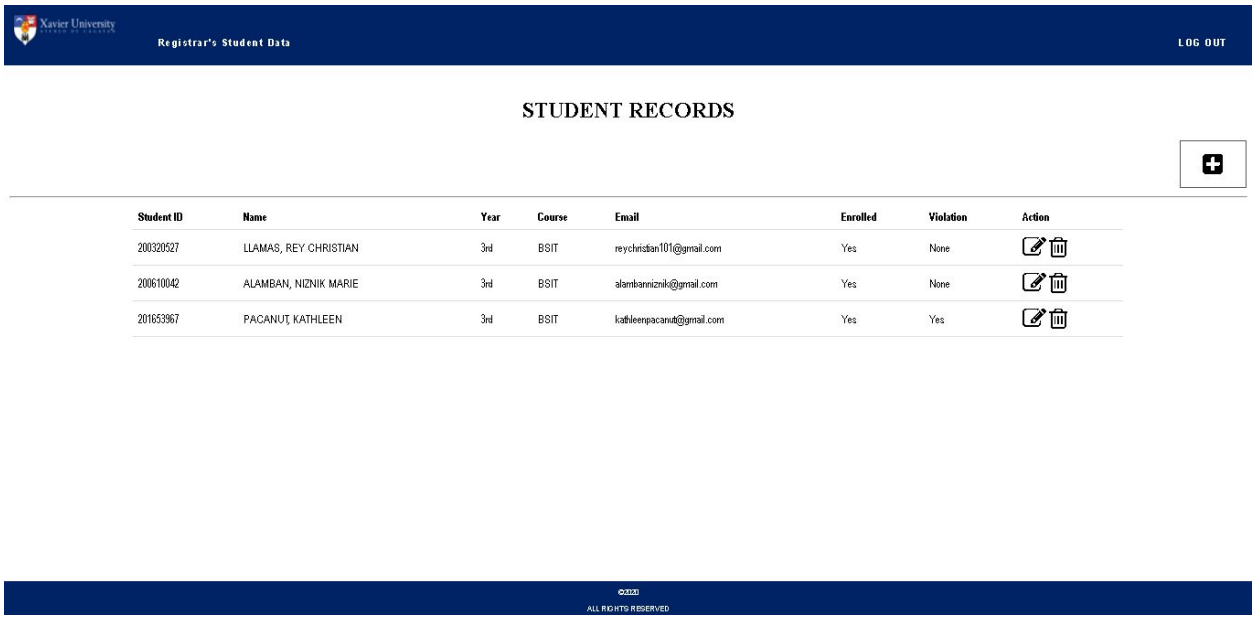

Figure 12 . Home Page

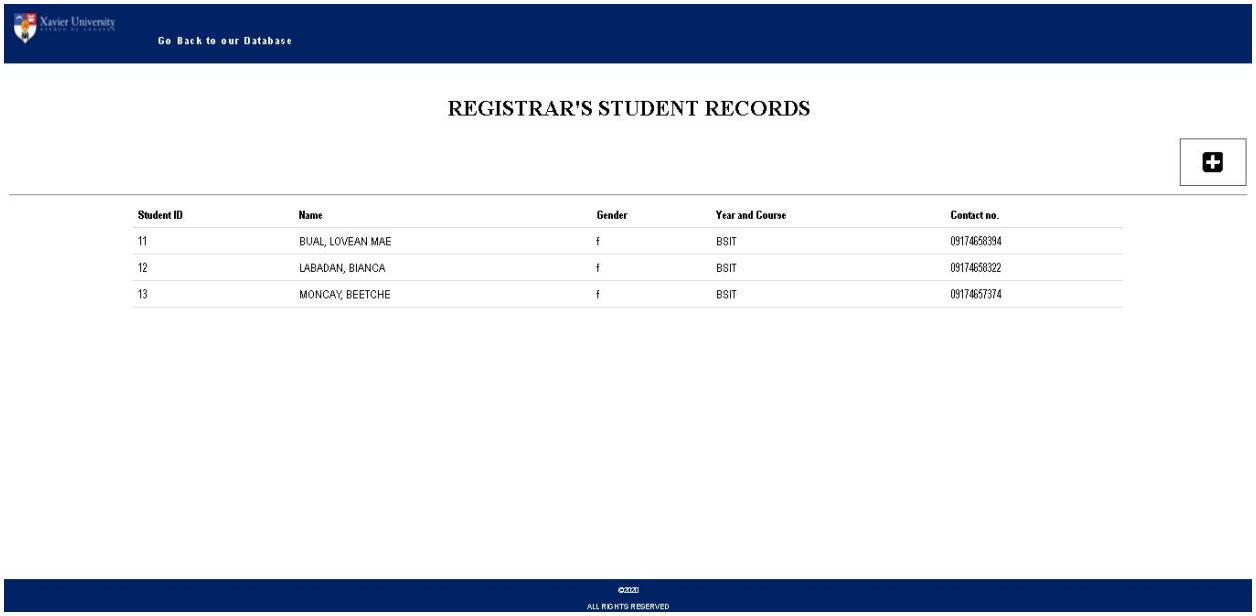

Figure 13 . Registrar's Student Data

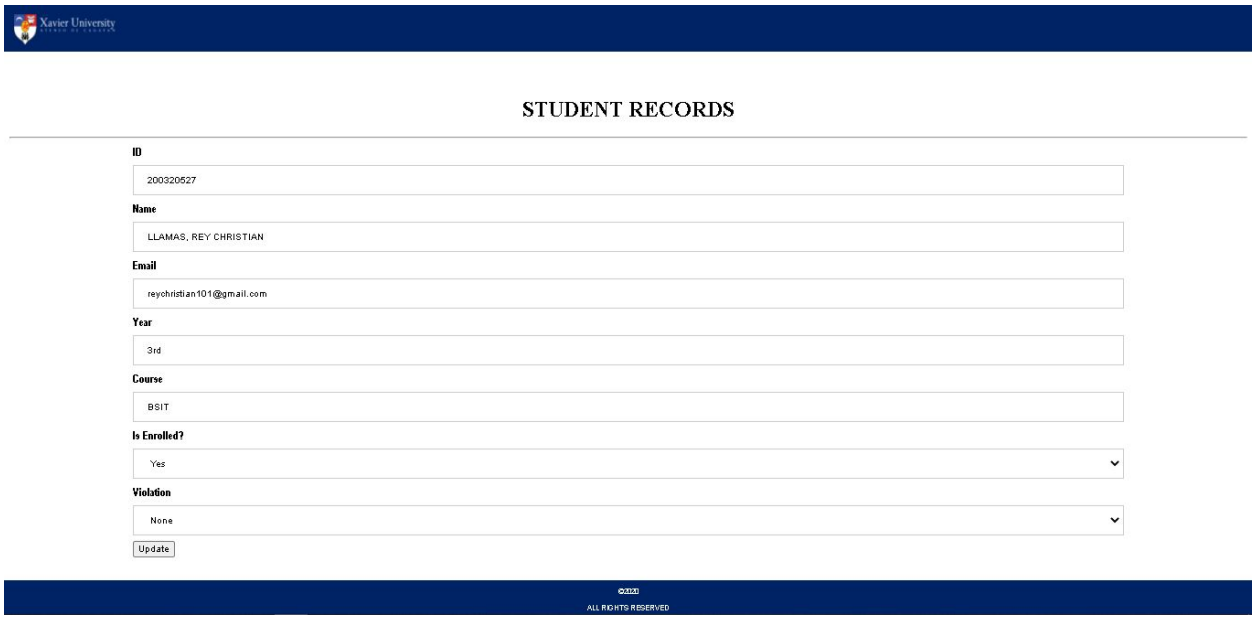

Figure 14 . Edit Page

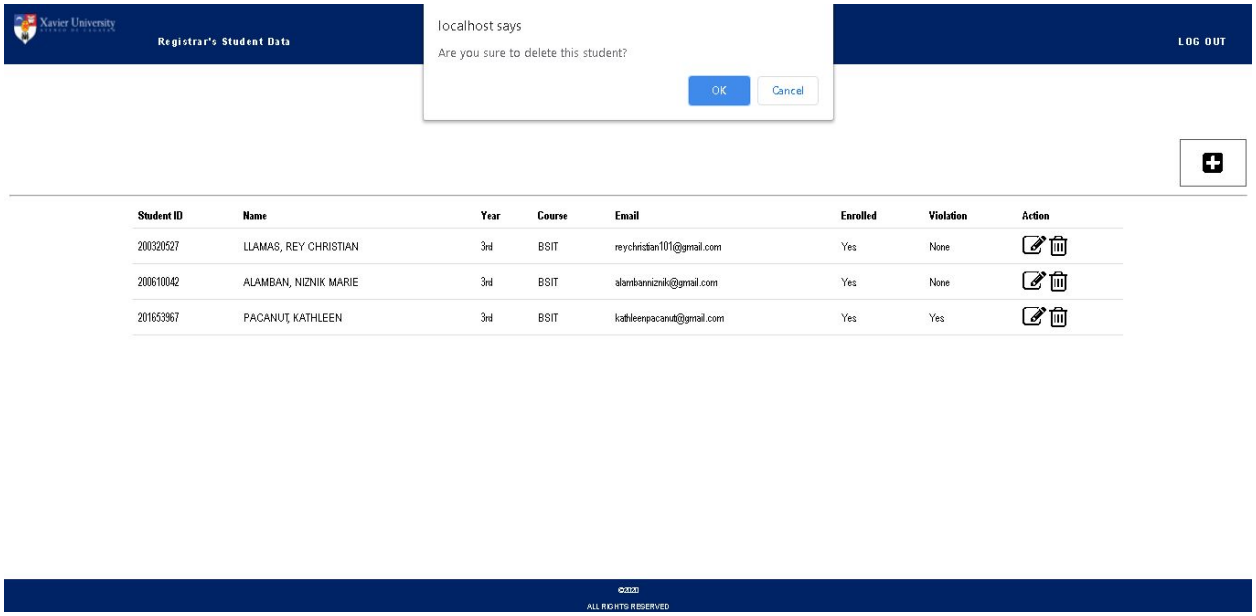

Figure 15. Delete Page

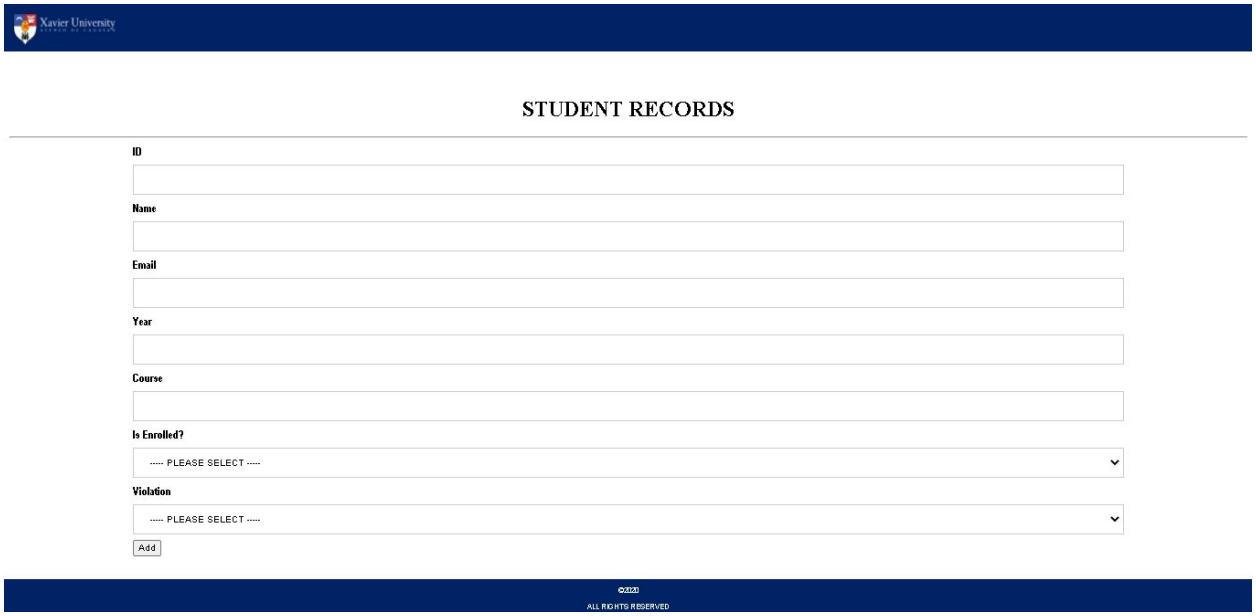

Figure 16. Add Student Page

Back-end

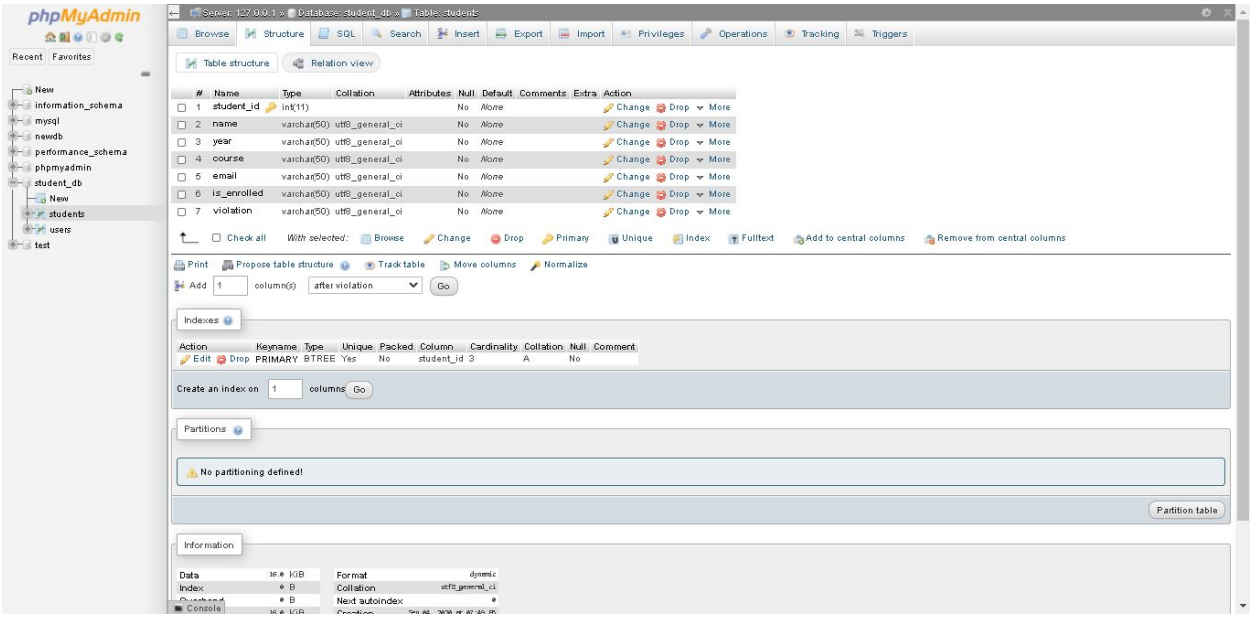

Figure 17. Web Application Database

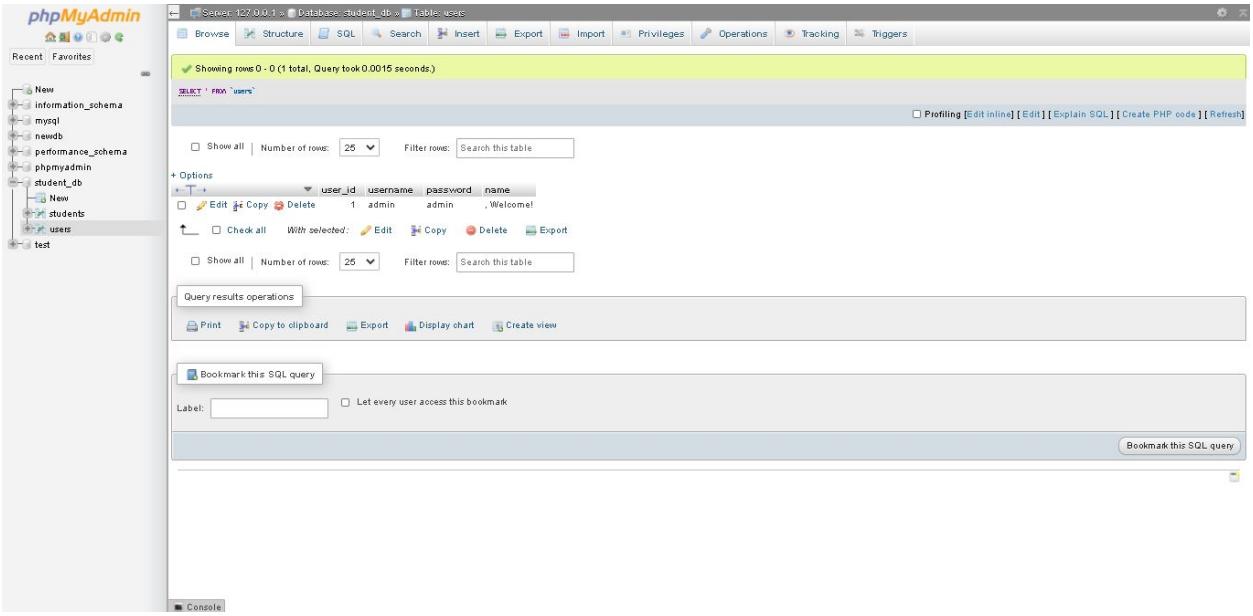

Figure 18. Login Database

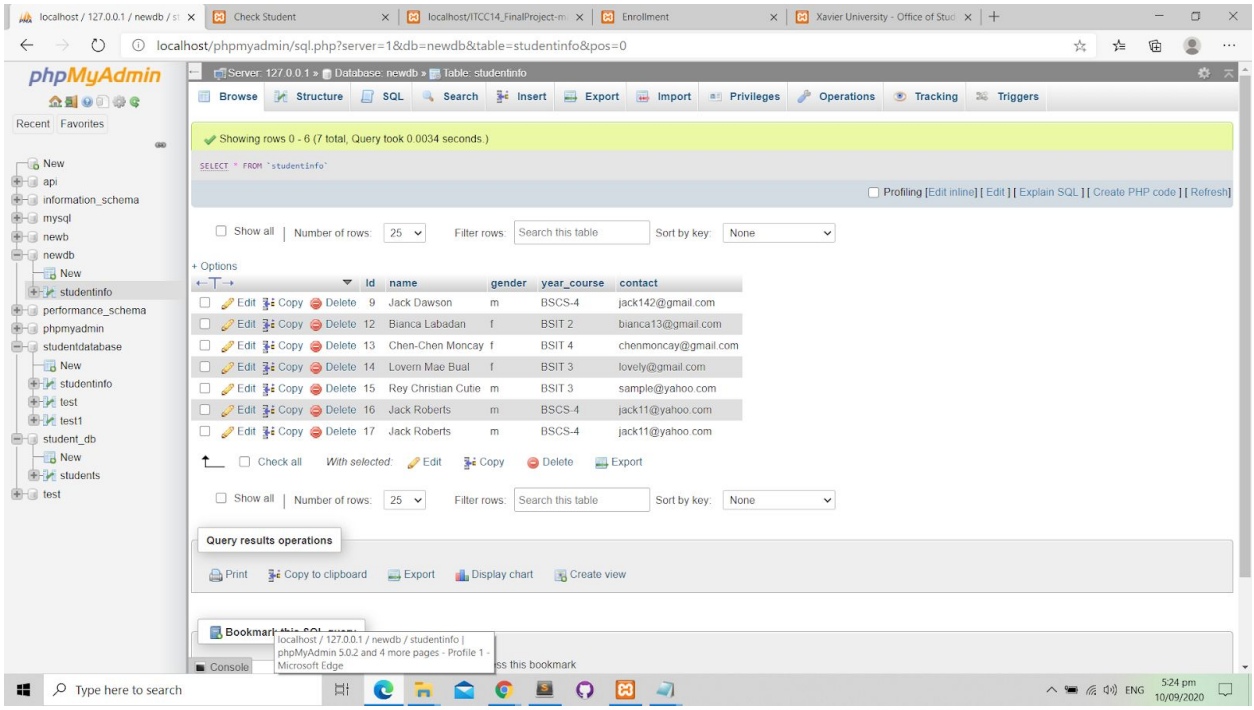

Figure 19. Registrar's Office Database

Chapter 3
Conclusion and Recommendation

3.1 Conclusion

In conclusion, the researchers were able to achieve most of the objectives stated in chapter one of the paper. They were able to develop a web application for the Office of Student Affairs through the use of the skills they have acquired in ITCC 14 - Integrative Programming and Technologies. The web application has the following functionalities: add, edit, and delete student information, and check the status of a student from the registrar's office database whether he or she is currently enrolled in the university. In addition, it also has a user authentication to ensure that unauthorized persons will not be able to access the information inside the web application. The purpose of creating the web application is to aid the Office of Student Affairs in updating student records in case there are any violations committed and also to update their information in case of any changes. Moreover, when students request for a Certificate of Good Moral as well as handling other transactions, the office will need to verify from the registrar's office whether the student is indeed currently enrolled in the university. Through these functionalities, the Office of Student Affairs will no longer have to waste a lot of papers and use google sheets for managing the data of the students and monitoring them.

When we asked a user to test the web application, they found the design to be simple and its functionalities to be understandable. However, due to some limitations, the web application still lacks a few features which are also essential for users to have a great user experience. The web application lacks the search and sort functions which would have saved so much time for the users especially if they are handling a huge amount of data. This function is very important to enable a more efficient process in their day to day operations. The web application also does not have other user types which has restrictions on the accessibility to some parts of the web application. With this, more research, practice, and tests should be made in the future to further improve the web application and ensure that it will enable an efficient day to day operations.

3.2 Recommendation

To the Office of Student Affairs. It is recommended for the office to utilize and maximize the functionalities of this kind of web application because it can help them save resources such as papers and enable a more efficient day to day operations. Furthermore, it is important for them to be fully knowledgeable of the web application's process and capabilities so that less errors will be made. Regular tests must also be done in case the web application is in need of any improvements.

To the Future Researchers. It is recommended that further research, practice, and tests be done in order to create a better web application and ensure that it satisfies the needs of the end users. Specifically, the researchers recommend that the search and sort functions should be added to the web application because they can surely save time and will allow the office to provide a better service to the students.

Bibliography

(2020). Retrieved 10 September 2020, from https://www.apachefriends.org/index.html.

17. User Types. Oasis-open.org. (2020). Retrieved 10 September 2020, from https://www.oasis-open.org/khelp/members/user_help/html/user_types.html.

Agarwal, M. (2020). *How to Perform Web Application Testing Effectively?*. Learn Programming and Software Testing. Retrieved 10 September 2020, from https://www.techbeamers.com/web-application-testing/.

contributors, p. (2020). *phpMyAdmin*. phpMyAdmin. Retrieved 10 September 2020, from https://www.phpmyadmin.net/.

PHP: Hypertext Preprocessor. Php.net. (2020). Retrieved 10 September 2020, from https://www.php.net/.