



STUDENT MANAGEMENT SYSTEM

BSIT21054 - WEB PROGRAMMING

BY: GROUP C

INTRODUCTION

- **A Student Management System (SMS) is a system that can handle all the details about students.**
- **The details include**
 - college details,
 - course details,
 - students' personal details,
 - academic details
 - etc.

- **Extensive information is available at your fingertips through this system.**
- **Viewing student data, managing, subject management, department, class, program and related issues are made simple and easy.**
- **There are custom search capabilities to aid in finding student information and working on student records.**
- **This can make the system easier to navigate and to use maximizing the effectiveness of time and other resources.**
- **Student Management System allows the keeping of personnel data in a form that can be easily accessed and analyse in a consistent way. The Student Management System is an automated version of manual Student Management System.**

OUR OBJECTIVES

- Student Management System is a management information system for education establishments to manage student data.
- It provides capabilities for registering students in courses, Admitting students, tracking student information.
- Ensure data integrity, privacy and security in an open – access environment.

PROJECT CATEGORY

- It is a web-based application.
- Unlike traditional applications, it is accessible anytime, anywhere, via a PC with an internet connection.
- It need lower requirements on the end user system and simplified architecture.

TOOLS AND PLATFORMS

- We have used PHP for back-end, HTML, CSS and JavaScript for the front end of this project
- For saving the data of students we have used MySQL database.
- The entire website executes on the XAMPP server.
- To access the website we can use any type of web browser.

HARDWARE AND SOFTWARE REQUIREMENTS

1. Offline Version

- Web browser
 - Google Chrome
 - Firefox
 - Internet Explorer
- System Requirement
 - OS : Windows 7/8/8.1/10
 - RAM : 2 GB
 - HDD : 250 MB
 - Processor : Intel Dual Core or higher



2. Web Based Version

- Web browser
 - Google Chrome
 - Firefox
 - Internet Explorer
- Internet Connection with good speed.



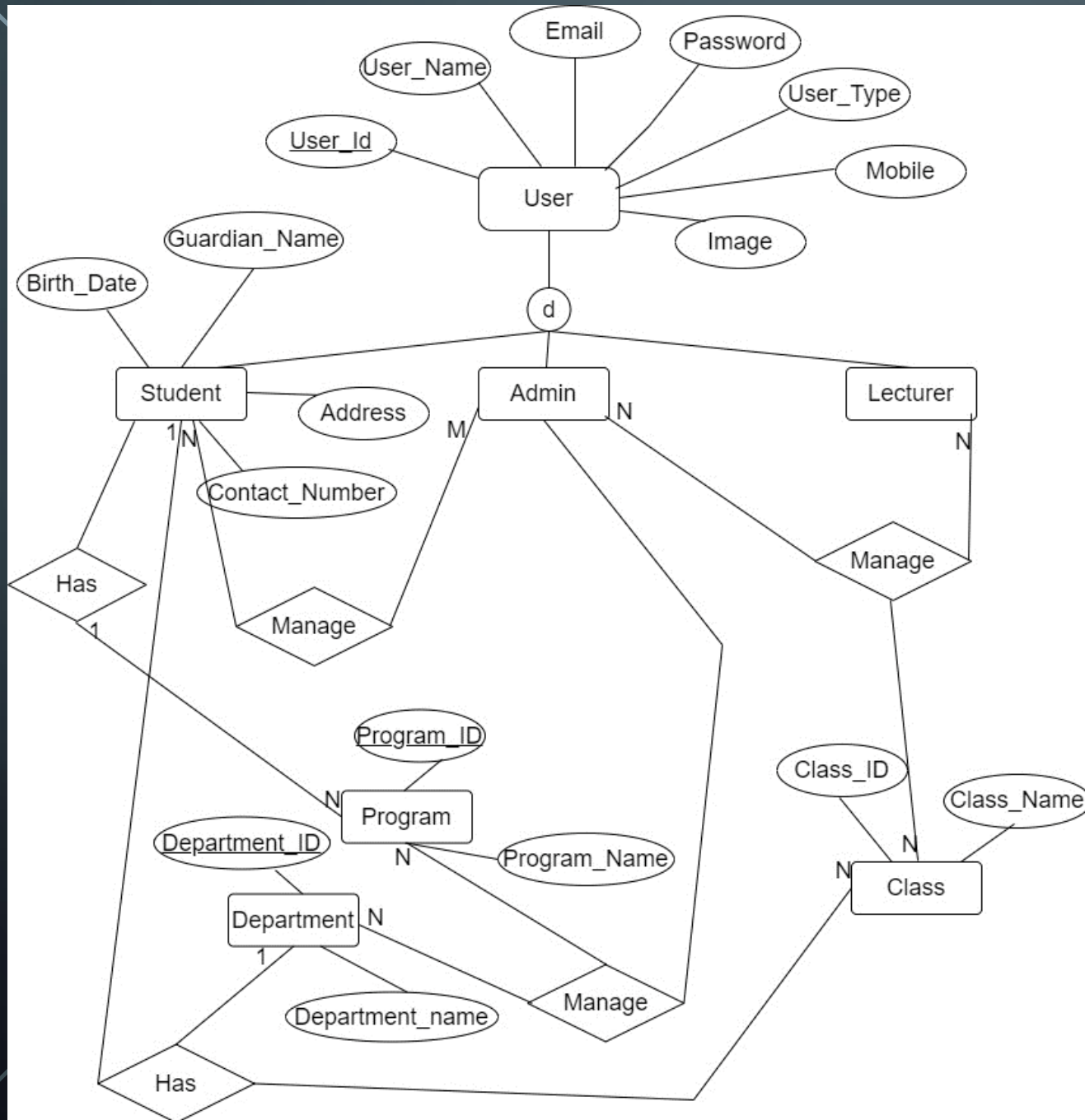
SYSTEM DESIGNS

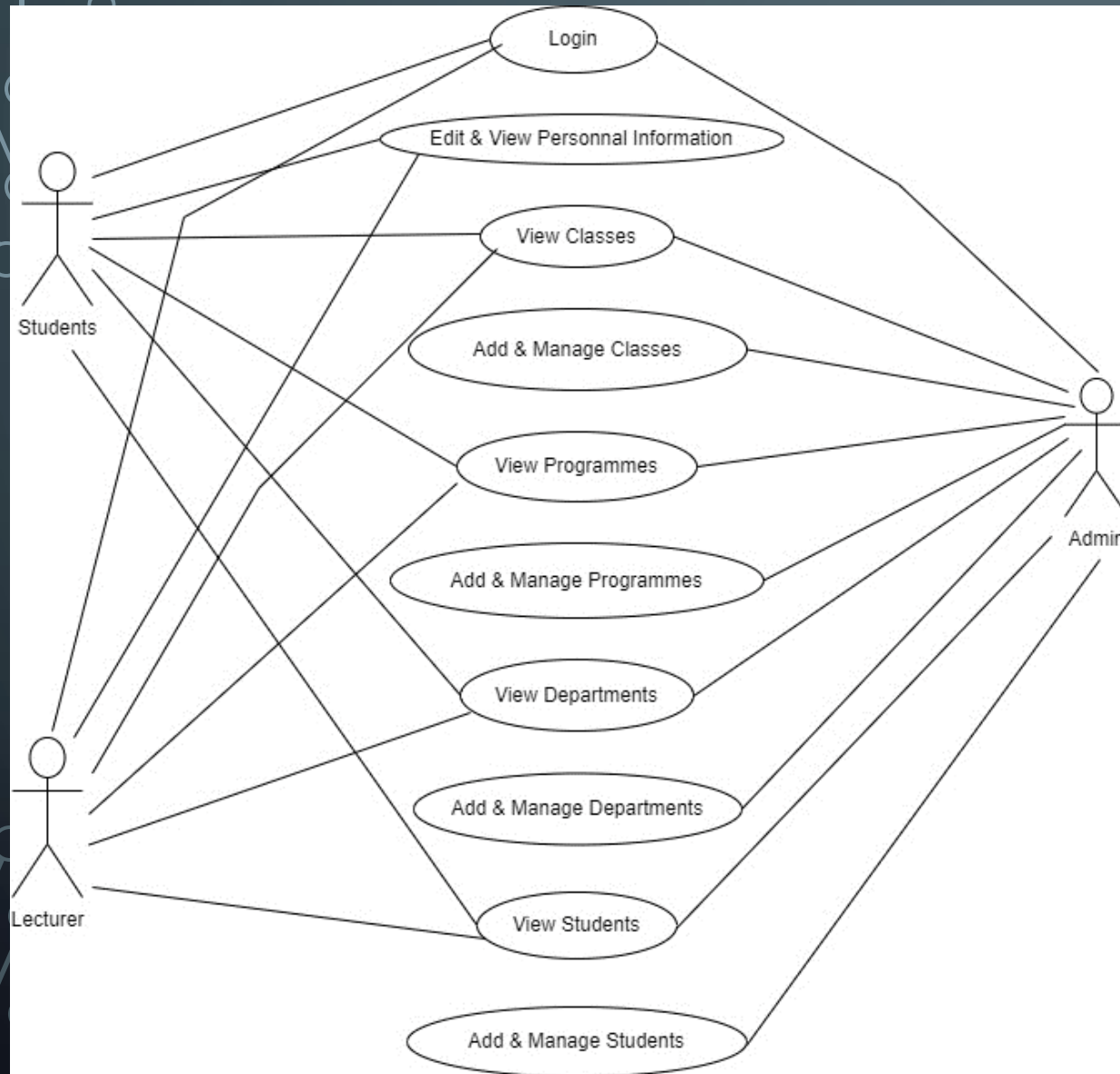
Systems design is the process of defining the architecture, interfaces, and data for a system that satisfies particular requirements. After you have your requirements for your system, the next step is translating them into **technical specifications** so you can construct your system.



ER DIAGRAM

An entity relationship diagram (ERD), also known as an entity relationship model, is a **graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system.**





USE – CASE DIAGRAM

Use-case diagrams describe the high-level functions and scope of a system. These diagrams also identify the interactions between the system and its actors. The use cases and actors in use-case diagrams describe what the system does and how the actors use it, but not how the system operates internally

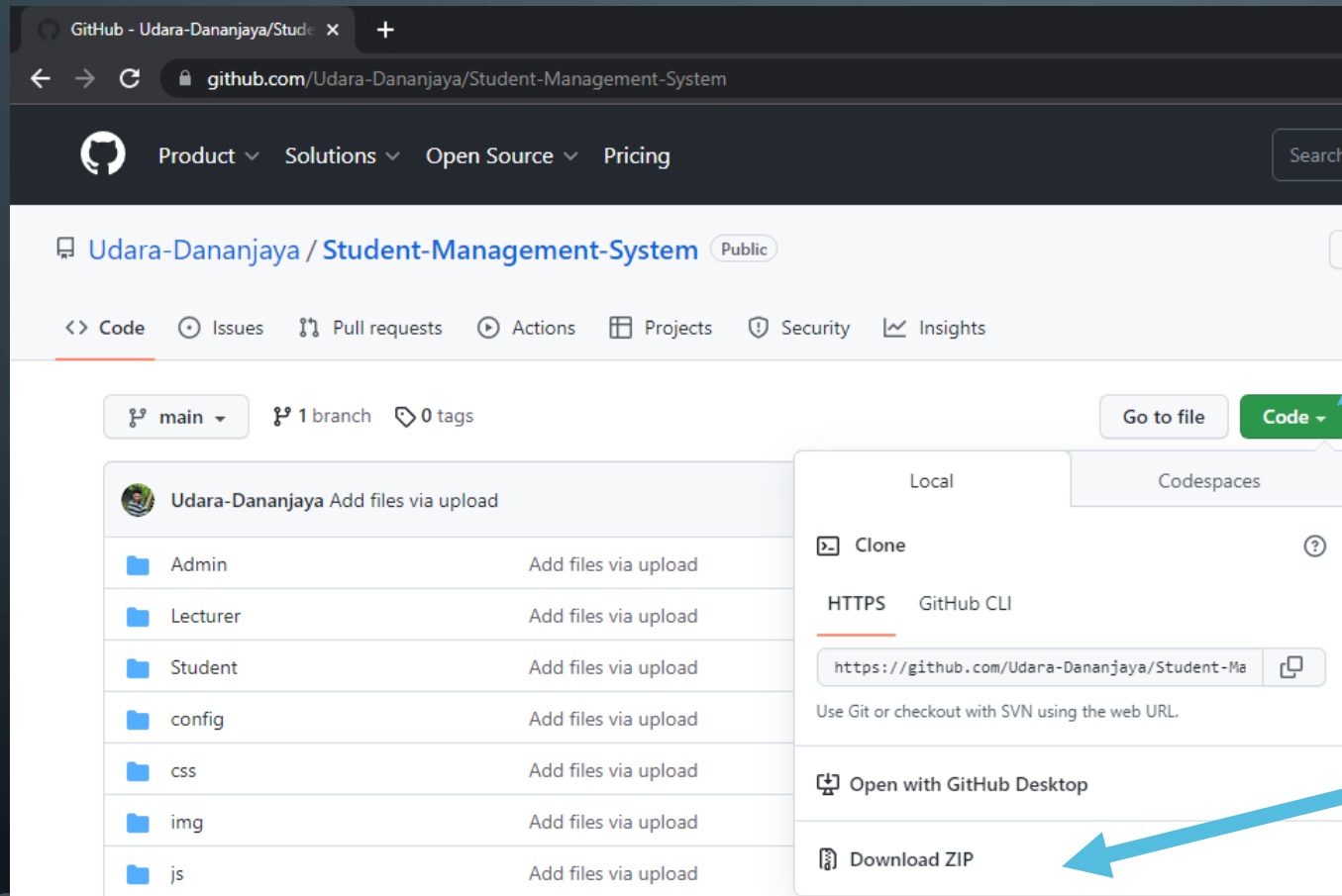
THE PROJECT FLOW

- Step 1:- Making the SRS.
- Step 2:- Coding and development
 - Coding for database
 - Coding for web interfaces
 - Connectivity of Databases
- Step 3:- Testing
 - Functionality Testing
 - Interface Testing
 - Performance Testing
- Implementation & Maintenance

SYSTEM INSTALLATION

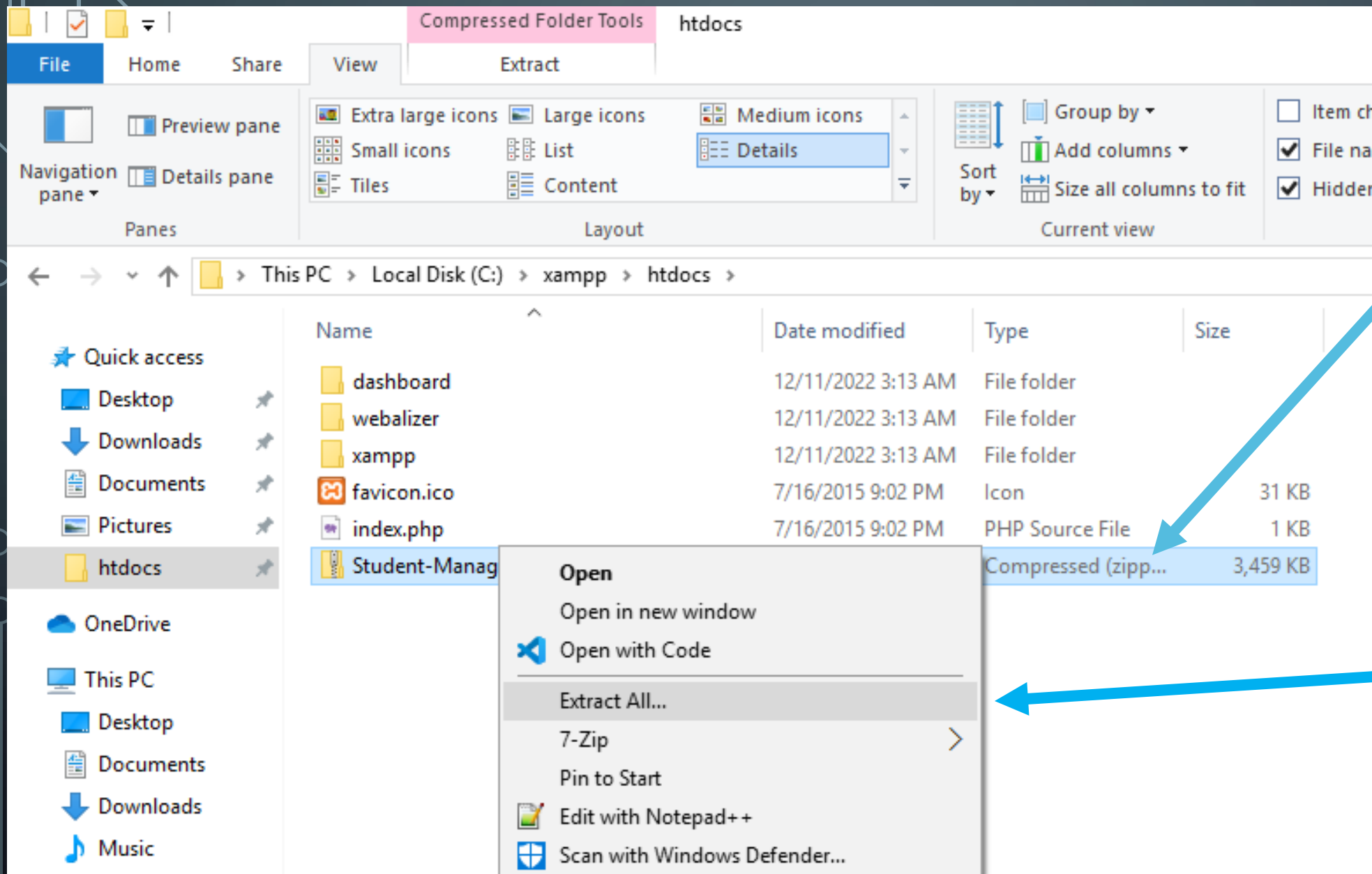
DOWNLOAD FROM GITHUB

- URL : <https://github.com/Udara-Dananjaya/Student-Management-System>



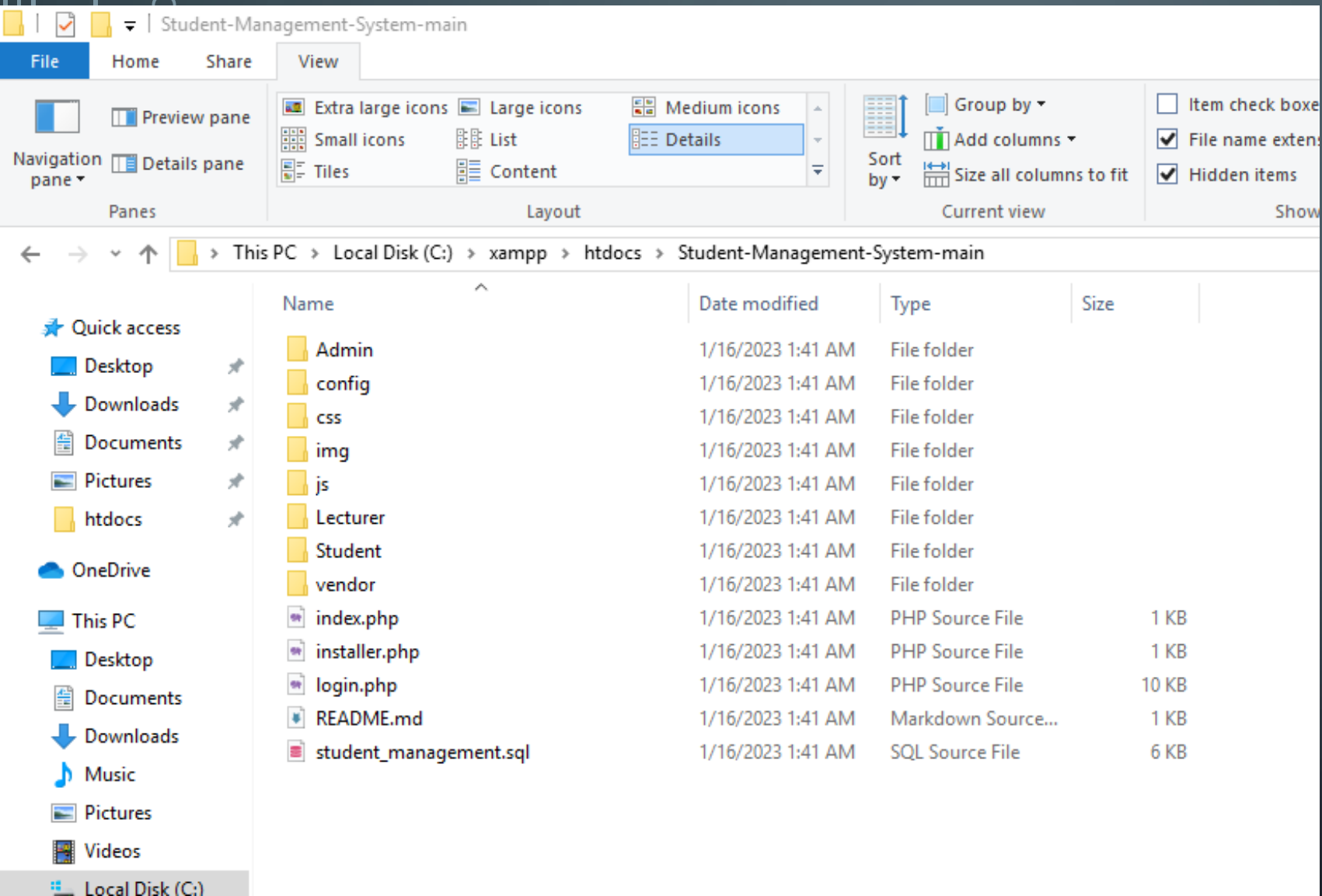
Click On this
“Code”

Click On this
“Download ZIP”



Move Downloaded file to
Xampp htdocs folder

Click "Extract All"

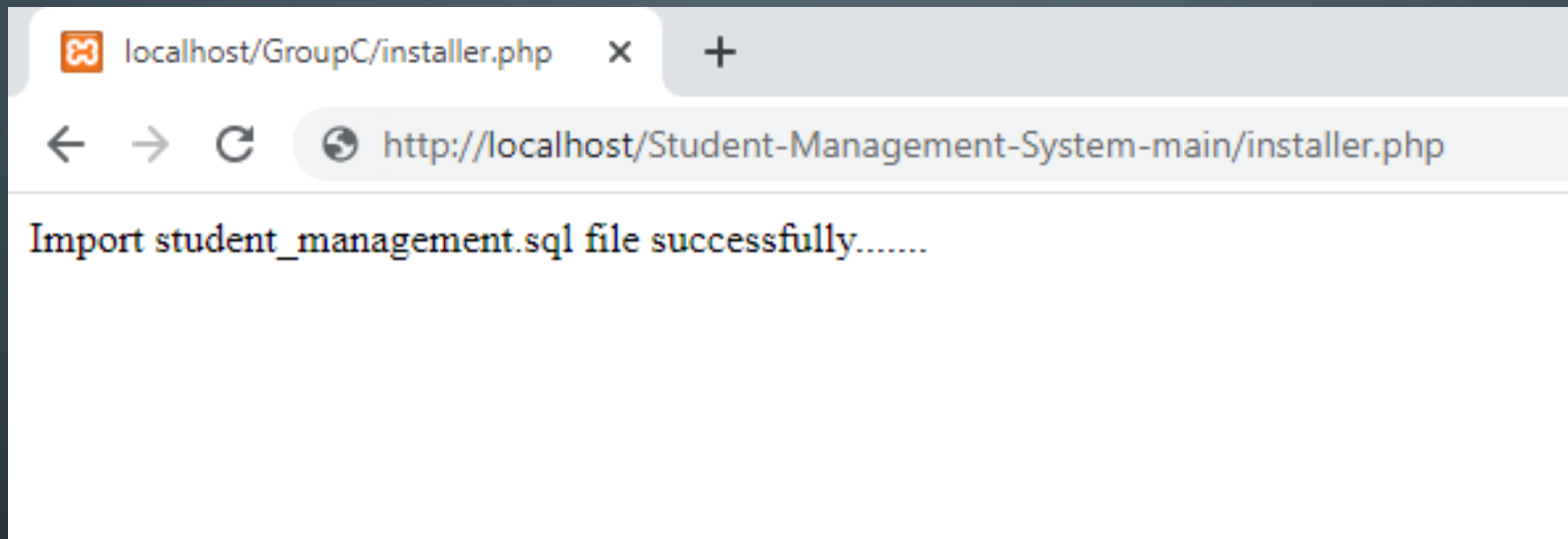


} Open extracted Folder

Now you need to go to this URL.

`http://localhost/< Your Folder Path>/installer.php`

`http://localhost/Student-Management-System-main/installer.php`



You can run our student management system by using this URL.

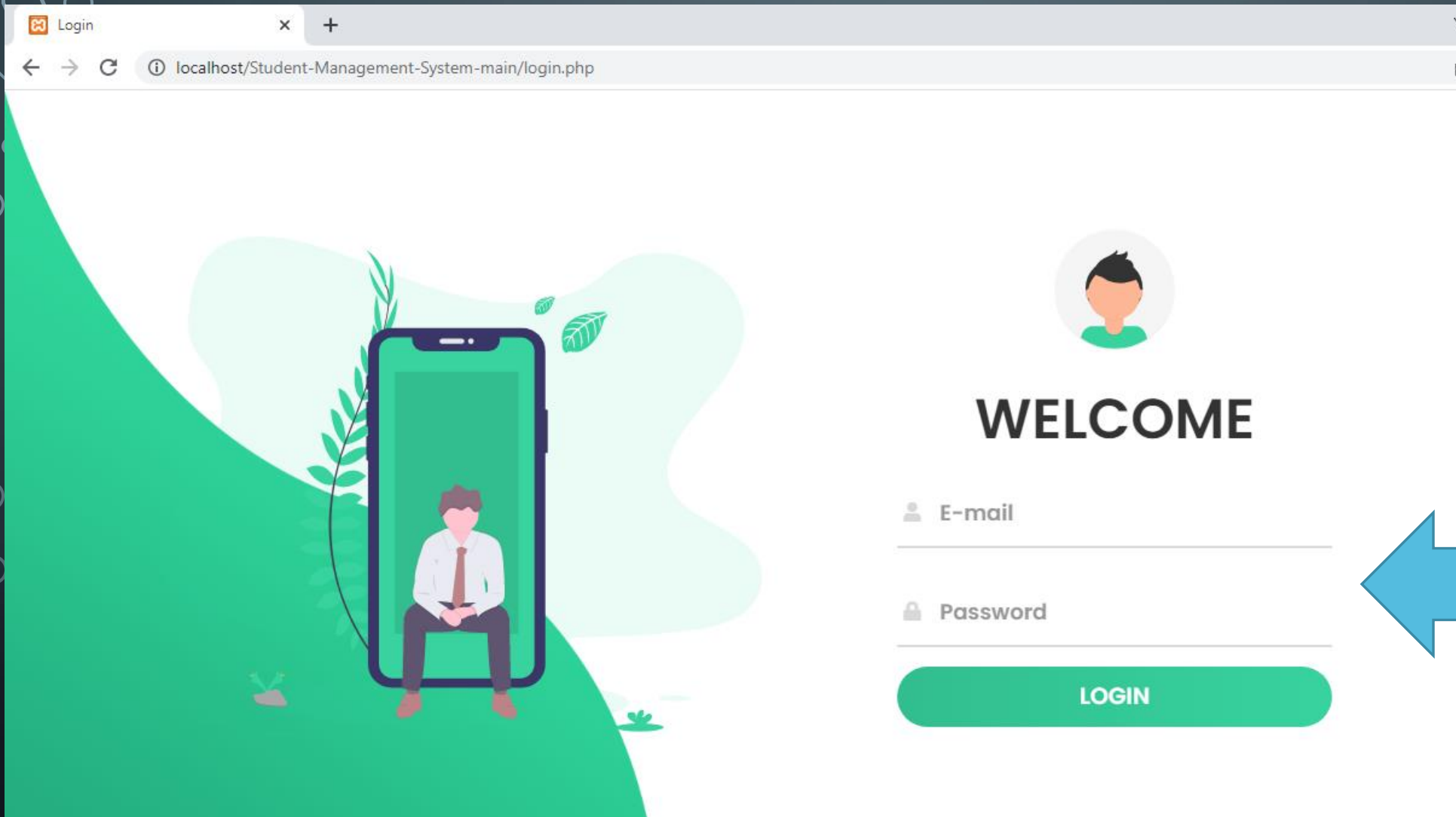
`http://localhost/< Your Folder Path >`

`http://localhost/Student-Management-System-main`

The image features a dark blue background with a subtle gradient. In the corners, there are decorative white line art elements resembling circuit boards or neural network connections, with lines and small circles. The text "SYSTEM CODING EXPLANATION" is centered in a bold, white, sans-serif font.


SYSTEM CODING EXPLANATION


LOGIN INTERFACE




Login


localhost/Student-Management-System-main/login.php





WELCOME

 E-mail

 Password

LOGIN

Login Data

LOGIN CODING

In this system, we store our database connection in one file because once our database username changes, we need to replace it on every page, so we store it in one file and import it to all files.

We use sessions to store

- current username,
- user types,
- etc.

```
login.php x
login.php
1  <?php
2
3  require_once "config/database.php";
4  session_start();
5
6  if (isset($_SESSION['user'])){
7      if($_SESSION['user']=="Admin"){
8          header('Location: Admin/Dashboard');
9      }
10     if($_SESSION['user']=="Lecturer"){
11         header('Location: Lecturer/Dashboard');
12     }
13     if($_SESSION['user']=="Student"){
14         header('Location: Student/Dashboard');
15     }
16 }
```

LOGIN CODING

```
if (isset ($_POST['submit'])) {  
    echo "<script src='js/sweetalert.js'></script>";  
    echo "<br>";  
    if ($stmt = $conn->prepare('SELECT  user_name, password,user_id, user_type FROM user WHERE email = ?')) {  
        $stmt->bind_param('s', $_POST['email']);  
        $stmt->execute();  
        $stmt->store_result();  
        if ($stmt->num_rows > 0) {  
            $stmt->bind_result( $user_name, $password, $user_id, $user_type);  
            $stmt->fetch();  
            $post_password=md5($_POST['password']);  
            if ($post_password== $password) {  
                session_regenerate_id();  
                $_SESSION['Session_Id']=$_POST['email'];  
                $_SESSION['username'] =$user_name;  
                $_SESSION['user_id'] =$user_id;
```

We use prepared statements because they are very useful against SQL injections.


LOGIN CODING

```
if ($post_password== $password) {  
    session_regenerate_id();  
    $_SESSION['Session_Id']=$_POST['email'];  
    $_SESSION['username'] =$user_name;  
    $_SESSION['user_id'] =$user_id;  
    switch ($user_type) {  
        case "Admin":  
            $_SESSION['user']="Admin";  
            header('Location: Admin/Dashboard');  
            break;  
        case "Lecturer":  
            $_SESSION['user']="Lecturer";  
            header('Location: Lecturer/Dashboard');  
            break;  
        case "Student":  
            $_SESSION['user']="Student";  
            header('Location: Student/Dashboard');  
            break;  
    }  
} else {
```

DASHBOARD

Admin Panel - Dashboard

localhost/Student-Management-System-main/Admin/Dashboard



UDARA_DANANJAYA@IT.BCI.LK

Dashboard

DEPARTMENT

Manage Department

Add New Department

PROGRAM

Manage Program

Add New Program

CLASS

Manage Class

Add New Class

LECTURER

Manage Lecturer

Dashboard

4 DEPARTMENTS

11 PROGRAMS

2 CLASSES

0 STUDENTS

Student Data

Show 10 entries

Search:

ID	User Name	E-mail	Mobile No	Class	Birth Date
No data available in table					

Showing 0 to 0 of 0 entries

PreviousNext


DASHBOARD

Dashboard.php

Admin > Dashboard.php

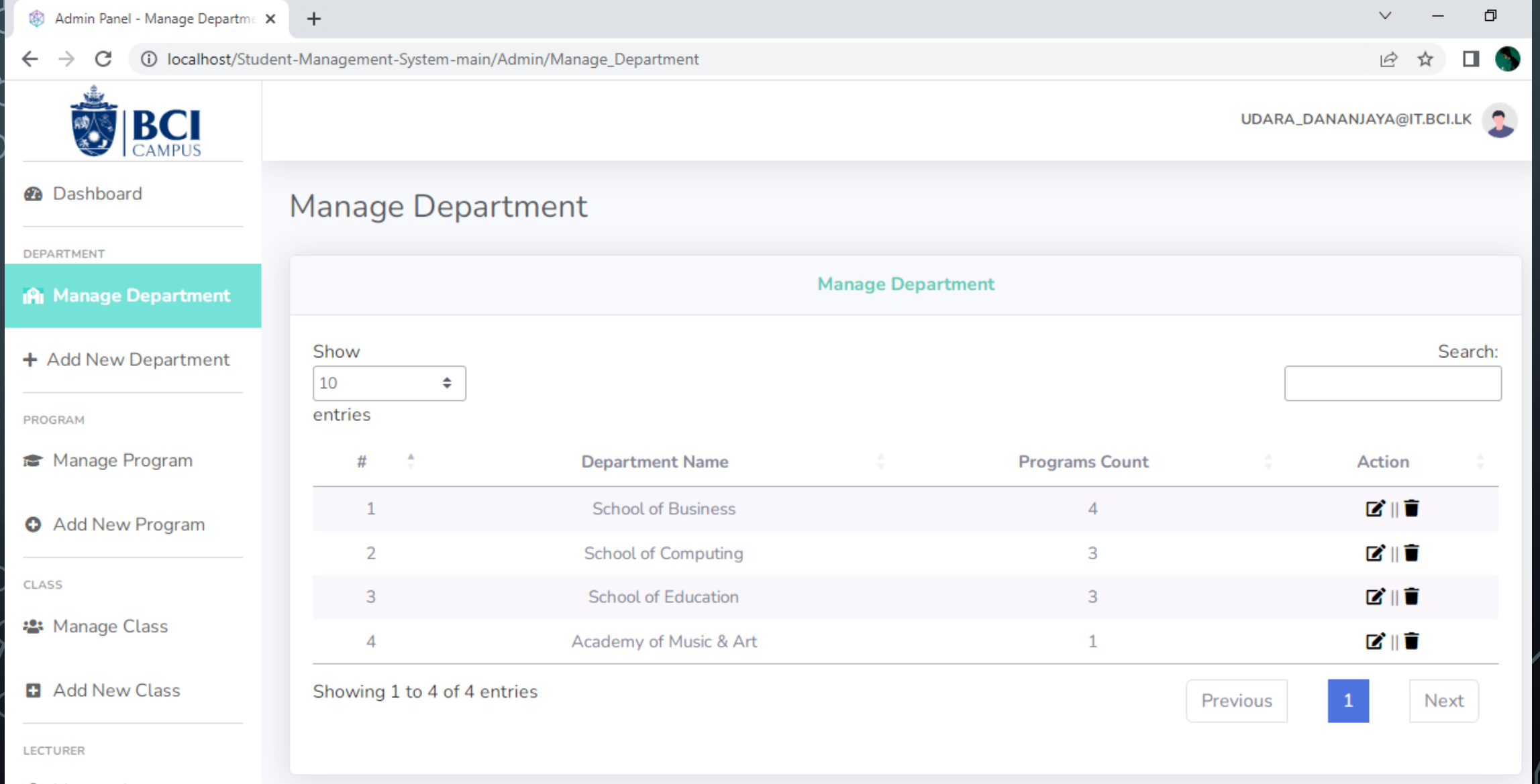
```
1  <?php
2  session_start();
3  if(!isset($_SESSION['Session_Id']))
4  {
5      header('Location: ../login.php');
6  }
7  if(!($_SESSION['user']=="Admin")){
8      header('Location: ../login.php');
9  }
10
11  require_once "../config/database.php";
12  require_once "layout/header.php"; // We use Header And Footer in Separate file
13  require_once "layout/footer.php";
14
15  $sql = "SELECT COUNT(`department_id`) FROM `department`";
16  $result = $conn->query($sql);
17  $row = $result->fetch_assoc();
18  $dep_count= $row["COUNT(`department_id`)"];
19
20  $sql = "SELECT COUNT(`program_id`) FROM `program`";
21  $result = $conn->query($sql);
22  $row = $result->fetch_assoc();
23  $prgm_count= $row["COUNT(`program_id`)"];
24
```

DASHBOARD

Admin >  Dashboard.php

```
38  html_header("Dashboard");
39
40  echo <<<EOT
41
42  <h1 class="h3 mb-4 text-gray-800">Dashboard</h1>
43  <div class="row">
44
45  <div class="col-xl-3 col-md-6 mb-4">
46      <div class="card bg-gradient-card-2 shadow h-100 py-2">
47          <div class="card-body">
48              <div class="row no-gutters align-items-center">
49                  <div class="col mr-2">
50                      <div class="h5 mb-0 font-weight-bold text-center text-white">$dep_count</div>
51                      <div class="text-xs font-weight-bold text-center text-uppercase mb-1 text-white">De
52                  </div>
53                  <div class="col-auto">
54                      <i class="fas fa-calendar fa-2x text-white"></i>
55                  </div>
56              </div>
57          </div>
58      </div>
59  </div>
60
```


MANAGE DEPARTMENT




MANAGE DEPARTMENT

Admin > Manage_Department.php

```
15 $sql = "SELECT `department_name`, `department`.`department_id`, COUNT(`department`.`department_id`) AS 'p_count'
16 FROM `department` LEFT JOIN `program` ON `department`.`department_id` = `program`.`department_id`
17 GROUP BY `department_id`;";
18 $result = $conn->query($sql);
19
20 html_header("Manage_Department");
21
22 if (isset($_GET['DeleteId'])) {
23     $sql = "DELETE FROM `department` WHERE `department_id` = {$_GET['DeleteId']}";
24     $conn->query($sql);
25     echo"<script>window.location.href = 'Manage_Department';</script>";
26 }
27
28
29 echo <<<EOT
30
31 <h1 class="h3 mb-4 text-gray-800">Manage Department</h1>
32
33 <div class="card shadow mb-4">
34 <div class="card-header py-3">
35     <h6 class="m-0 font-weight-bold text-center" style="color:#58cfbb;">Manage Department</h6>
36 </div>
37 <div class="card-body">
38     <table class="stripe" style="text-align:center; white-space:nowrap;font-size: 15px;" id="dataTable" width="
39     <thead>
40     <tr>
```

MANAGE DEPARTMENT

Admin >  Manage_Department.php

```
40 <tr>
41     <th style="text-align: center;">#</th>
42     <th style="text-align: center;">Department Name</th>
43     <th style="text-align: center;">Programs Count</th>
44     <th style="text-align: center;">Action</th>
45 </tr>
46 </thead>
47 <tbody>
48 EOT;
49
50 if ($result->num_rows > 0) {
51     $i=0;
52
53     while($row = $result->fetch_assoc()) {
54
55     $i++;
56         echo <<<EOT
57         <tr>
58         <td>$i</td>
59         <td>{$row["department_name"]}</td>
60         <td>{$row["p_count"]}</td>
61         <td> <a href='Update_Department?EditId={$row["department_id"]} '><i class="fas fa-edit"></i></a> || <a href='M
62         </tr>
63         EOT;
64     }
65     echo"</tbody></table></div></div>";
```

ADDITIONAL FUNCTION

🐘 email.php ×

Admin > controller > 🐘 email.php

```
1  k?php
2  function mailsender( $Mail_Subject,$Mail_Txt,$MAIL_TO ){
3      //SMTP CREDENTIALS
4      $SMTP_SERVER = 'smtp.mailgun.org';
5      $SMTP_USERNAME = "postmaster@sandbox2946eecca5fc4d5682ee5143e8450770.mailgun.org";
6      $SMTP_PASSWORD = "3ed185b718bf130d43d6592a833c5153-c2efc90c-7a7c2a1a";
7
8      $mail_string = "From: <$SMTP_USERNAME>\r\n";
9      $mail_string .= "To: <$MAIL_TO>\r\n";
10     $mail_string .= "Date: " . date('r') . "\r\n";
11     $mail_string .= "Subject: $Mail_Subject\r\n";
12     $mail_string .= "\r\n";
13     $mail_string .= " $Mail_Txt\r\n";
14     $mail_string .= "\r\n";
15
16     $emailFile = fopen("php://temp", 'w+');
17     fwrite($emailFile, "$mail_string");
18     rewind($emailFile);
19     $fstat = fstat($emailFile);
20     $size = $fstat['size'];
21     $ch = curl_init($SMTP_SERVER);
22     curl_setopt($ch, CURLOPT_SAFE_UPLOAD, true);
23     curl_setopt($ch, CURLOPT_MAIL_FROM, "<" . $SMTP_USERNAME . ">");
24     curl_setopt($ch, CURLOPT_MAIL_RCPT, array("<" . $MAIL_TO . ">"));
25     curl_setopt($ch, CURLOPT_USERNAME, $SMTP_USERNAME);
26     curl_setopt($ch, CURLOPT_PASSWORD, $SMTP_PASSWORD);
```

We use this function for sending the username and password. Because we cannot give a username and password. Physically, it is not practical.

```
$sql = "INSERT INTO user (user_name, email, password,user_type,mobile)
VALUES ('${_POST['uname']}', '${_POST['email']}', '$post_pass','Student','${_POST['mobile']}')";
$conn->query($sql);
$user_id = $conn->insert_id;
$sql = "INSERT INTO `student`(`user_id`, `birth_date`, `guardian_name`, `contact_number`, `address`, `class_id`
VALUES ('$user_id','${_POST['birth_date']}', '${_POST['guardian_name']}', '${_POST['contact_number']}', '${_POST[
$conn->query($sql);

$mail_html = " Please Log in to your Account <br> UserName ${_POST['email']} : Password : ${_POST['pass']}";
mailsender("Student Added", " $mail_html", "${_POST['email']}");
echo"<script>window.location.href = 'Manage_Student';</script>";
```

DEPENDENCY

- **Bootstrap** (For Designing)
- **Datatables** (To get advanced features for HTML tables)
- **Fontawesome** (icon library)
- **Jquery** (feature-rich JavaScript library)
- **Sweetalert.js** (customizable JavaScript's popup boxes.)

The background is a dark blue gradient. In the corners, there are decorative white line art elements resembling circuit boards or neural networks, with lines and small circles.

Thank you!

- Megana Ariyaratna
- Udara Aththanayaka
- Enosh Rodrigo