FACULTY OF ENGINEERING

SCHOOL OF COMPUTING

SEMESTER 2/20212022

SECV1223 – WEB PROGRAMMING

SECTION 09

PROJECT DOCUMENTATION

STUDENTS' INDUSTRIAL TRAINING MANAGEMENT SYSTEM

LECTURER: DR NUR ZURAIFAH SYAZRAH BINTI OTHMAN

GROUP NO: 3

TAY WEI JIAN	A20EC0159
TEE NG ZIKANG	A20EC0161
TEOH YEE XIAN	A20EC0164
THAM CHUAN YEW	A20EC0166
CHIA JEE LIANG	A20EE0029

DETAILS OF THE SYSTEM

File name of login/landing page: index.php

File name of configuration file: config.php

Name of database: industrialtrainingdatabase

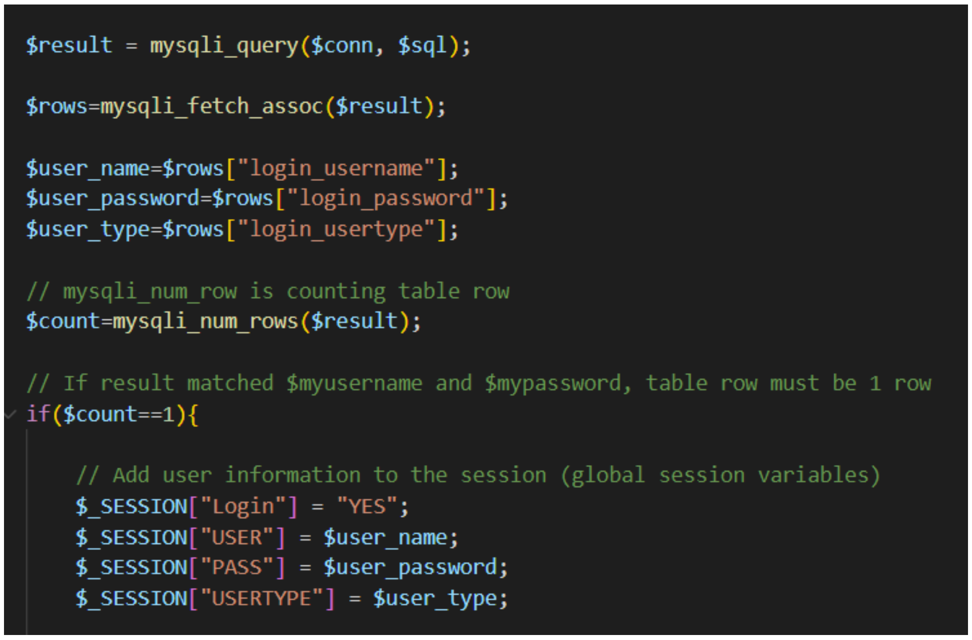
LOGIN DETAILS

Table 1: Login Combination Table

Username	Password	User Level
admin	admin	1
admin0921	admin0921#	1
admin132	#admin132	1
coor123	#coor123	2
st001coor	#st123	2
st003coor	#iwoq1	2
bobby123	$bobby	3
murshid	#qe12	3
nafeesa0111	*nafeesa	3

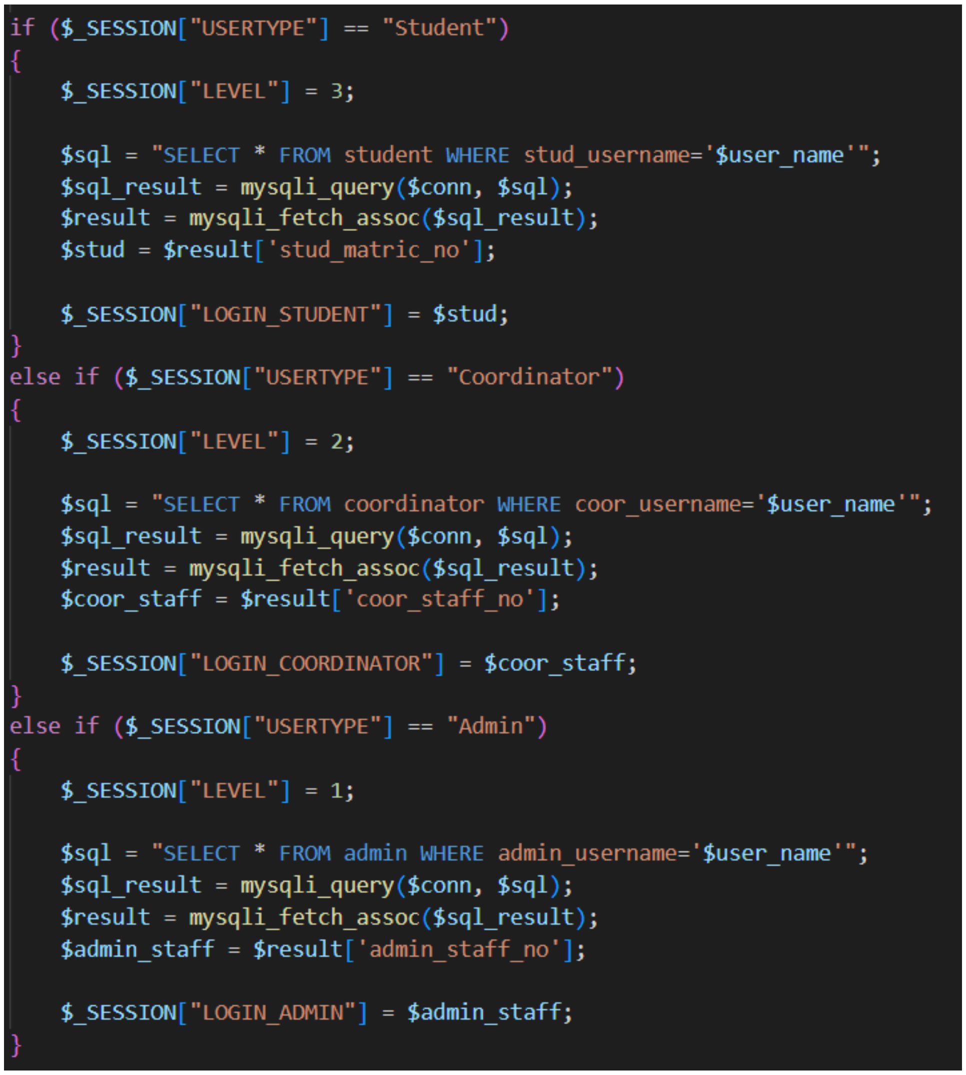
USAGE OF SESSION

The usage of session in this project allows the login user data to be passed from one page to another page. Session is a global variable that is stored on the server for later use and the session information is temporary and deleted after the login user leaves the website. The session used in the project allows the login user data such as login username, login password, login user type, login user level and the login user's staff number or matric number based on the login user type. The declaration of the session variable is performed in check_login.php as shown in the two figures below:



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

$rows=mysqli_fetch_assoc($result);

$user_name=$rows["login_username"];
$user_password=$rows["login_password"];
$user_type=$rows["login_usertype"];

// mysqli_num_row is counting table row
$count=mysqli_num_rows($result);

// If result matched $myusername and $mypassword, table row must be 1 row
if($count==1){

    // Add user information to the session (global session variables)
    $_SESSION["Login"] = "YES";
    $_SESSION["USER"] = $user_name;
    $_SESSION["PASS"] = $user_password;
    $_SESSION["USERTYPE"] = $user_type;
```

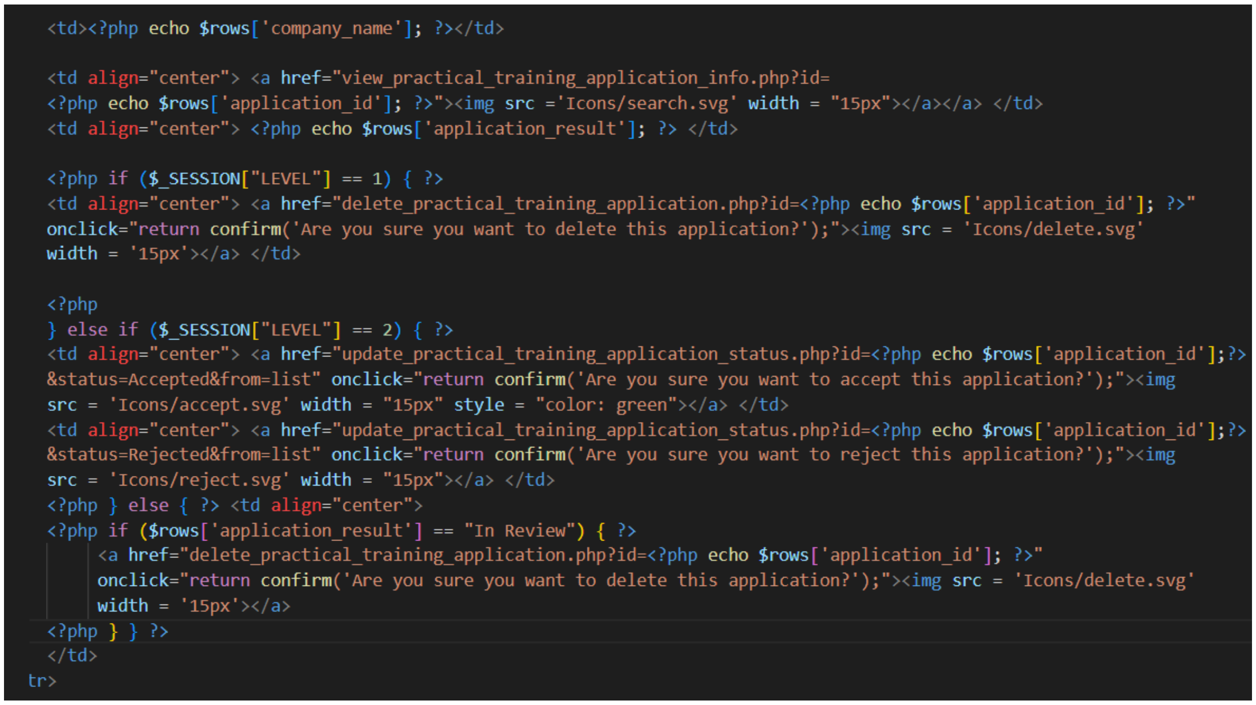
Figure 1: Setting the Values for Global Session Variables such as "Login", "USER", "PASS" and "USERTYPE"



```php
if ($_SESSION["USERTYPE"] == "Student")
{
    $_SESSION["LEVEL"] = 3;

    $sql = "SELECT * FROM student WHERE stud_username='$user_name'";
    $sql_result = mysqli_query($conn, $sql);
    $result = mysqli_fetch_assoc($sql_result);
    $stud = $result['stud_matric_no'];

    $_SESSION["LOGIN_STUDENT"] = $stud;
}
else if ($_SESSION["USERTYPE"] == "Coordinator")
{
    $_SESSION["LEVEL"] = 2;

    $sql = "SELECT * FROM coordinator WHERE coor_username='$user_name'";
    $sql_result = mysqli_query($conn, $sql);
    $result = mysqli_fetch_assoc($sql_result);
    $coor_staff = $result['coor_staff_no'];

    $_SESSION["LOGIN_COORDINATOR"] = $coor_staff;
}
else if ($_SESSION["USERTYPE"] == "Admin")
{
    $_SESSION["LEVEL"] = 1;

    $sql = "SELECT * FROM admin WHERE admin_username='$user_name'";
    $sql_result = mysqli_query($conn, $sql);
    $result = mysqli_fetch_assoc($sql_result);
    $admin_staff = $result['admin_staff_no'];

    $_SESSION["LOGIN_ADMIN"] = $admin_staff;
}
```

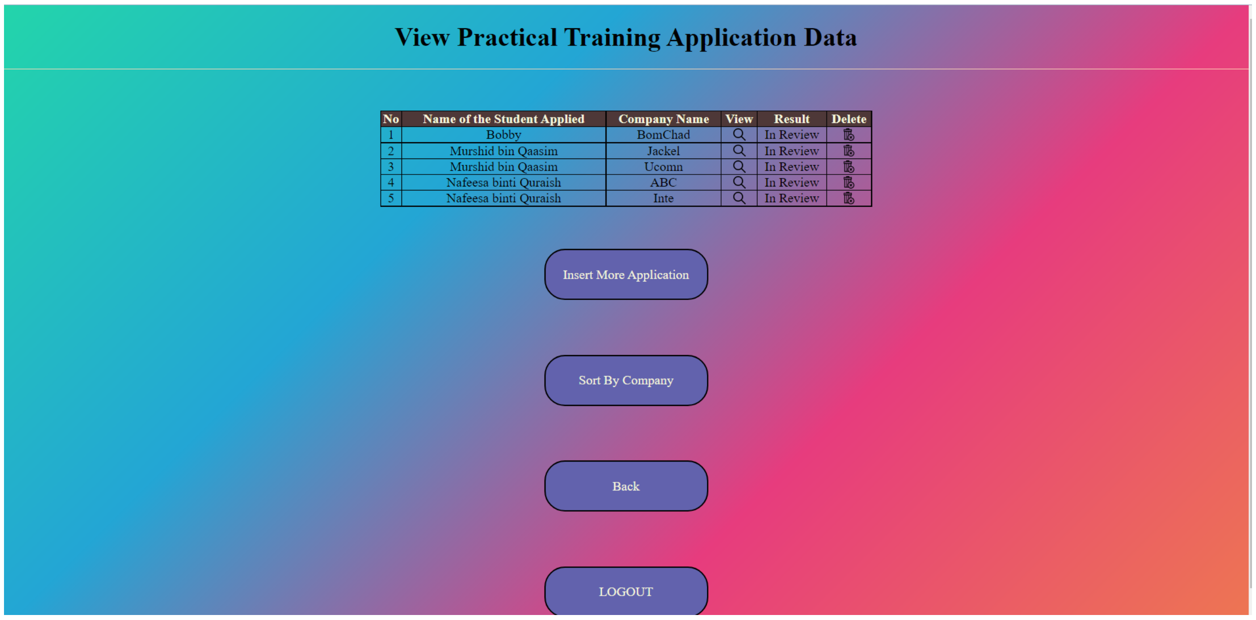
Figure 2: Setting the Values for Global Session Variables such as "LEVEL", "LOGIN_STUDENT", "LOGIN_COORDINATOR" and "LOGIN_ADMIN" based on the Login User Type

The session variables are allowed to be used by other pages in the website for user level checking to display different contents to the user based on the login user level. For example, the delete function is only available for the users with admin type or student type (only the application with "In Review" status) whereas the accept and reject function only can be used by the users with coordinator type when viewing the practical training application. The code and the page related to this usage can be shown in the figures below:



```
  <td><?php echo $rows['company_name']; ?></td>

  <td align="center"> <a href="view_practical_training_application_info.php?id=
  <?php echo $rows['application_id']; ?>"><img src ='Icons/search.svg' width = "15px"></a></a> </td>
  <td align="center"> <?php echo $rows['application_result']; ?> </td>

  <?php if ($_SESSION["LEVEL"] == 1) { ?>
  <td align="center"> <a href="delete_practical_training_application.php?id=<?php echo $rows['application_id']; ?>"
  onclick="return confirm('Are you sure you want to delete this application?');"><img src = 'Icons/delete.svg'
  width = '15px'></a> </td>

  <?php
  } else if ($_SESSION["LEVEL"] == 2) { ?>
  <td align="center"> <a href="update_practical_training_application_status.php?id=<?php echo $rows['application_id'];?>
  &status=Accepted&from=list" onclick="return confirm('Are you sure you want to accept this application?');"><img
  src = 'Icons/accept.svg' width = "15px" style = "color: green"></a> </td>
  <td align="center"> <a href="update_practical_training_application_status.php?id=<?php echo $rows['application_id'];?>
  &status=Rejected&from=list" onclick="return confirm('Are you sure you want to reject this application?');"><img
  src = 'Icons/reject.svg' width = "15px"></a> </td>
  <?php } else { ?> <td align="center">
  <?php if ($rows['application_result'] == "In Review") { ?>
      <a href="delete_practical_training_application.php?id=<?php echo $rows['application_id']; ?>"
      onclick="return confirm('Are you sure you want to delete this application?');"><img src = 'Icons/delete.svg'
      width = '15px'></a>
  <?php } } ?>
  </td>
tr>
```

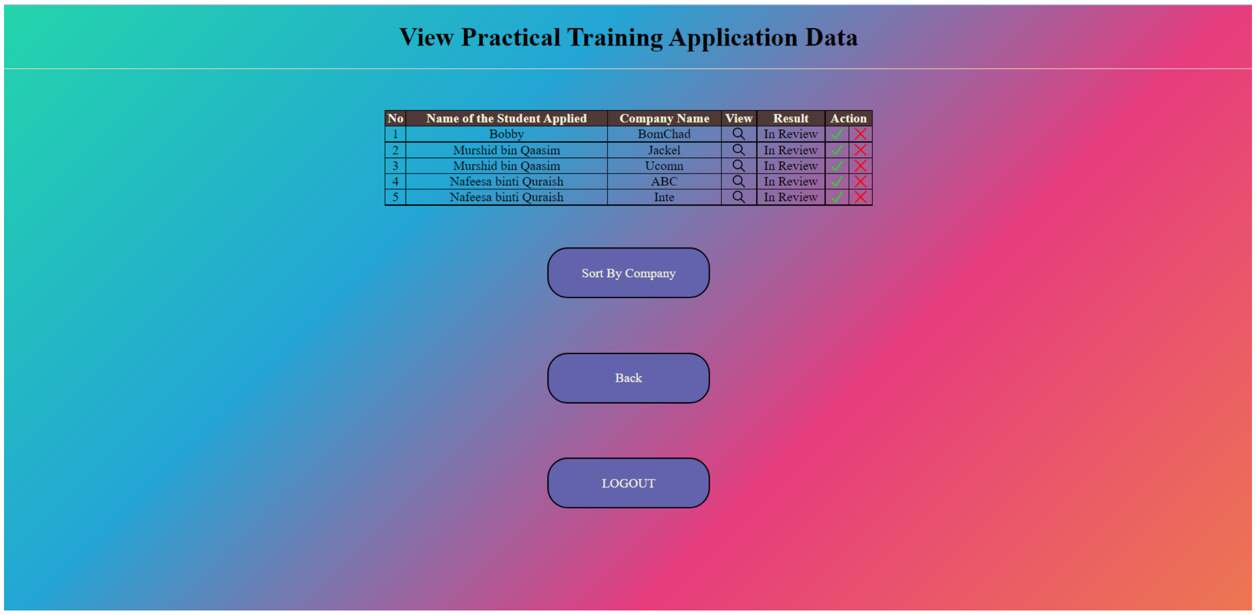
Figure 3: Code of Displaying Different Content in Viewing Practical Training Application



View Practical Training Application Data

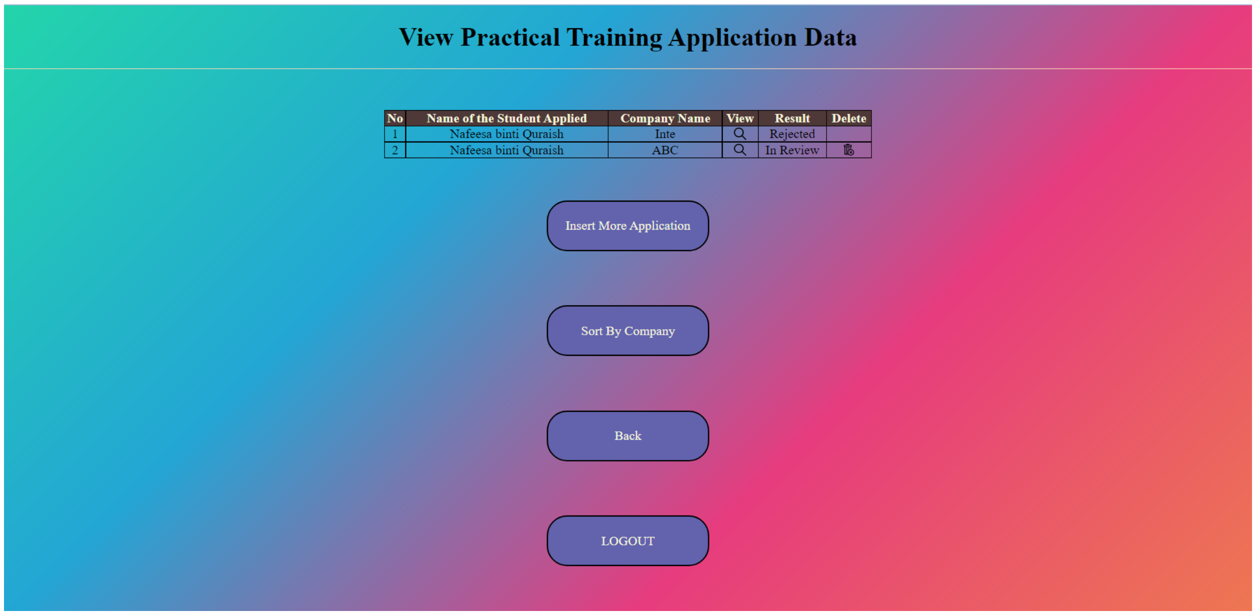
No	Name of the Student Applied	Company Name	View	Result	Delete
1	Bobby	BomChad	🔍	In Review	🗑
2	Murshid bin Qaasim	Jackel	🔍	In Review	🗑
3	Murshid bin Qaasim	Ucomn	🔍	In Review	🗑
4	Nafeesa binti Quraish	ABC	🔍	In Review	🗑
5	Nafeesa binti Quraish	Inte	🔍	In Review	🗑

Figure 4: Viewing Practical Training Application Page (Admin User)


View Practical Training Application Data

No	Name of the Student Applied	Company Name	View	Result	Action	
1	Bobby	BomChad	🔍	In Review	✓	✗
2	Murshid bin Qaasim	Jackel	🔍	In Review	✓	✗
3	Murshid bin Qaasim	Ucomn	🔍	In Review	✓	✗
4	Nafeesa binti Quraish	ABC	🔍	In Review	✓	✗
5	Nafeesa binti Quraish	Inte	🔍	In Review	✓	✗

Figure 5: Viewing Practical Training Application Page (Coordinator User)


View Practical Training Application Data

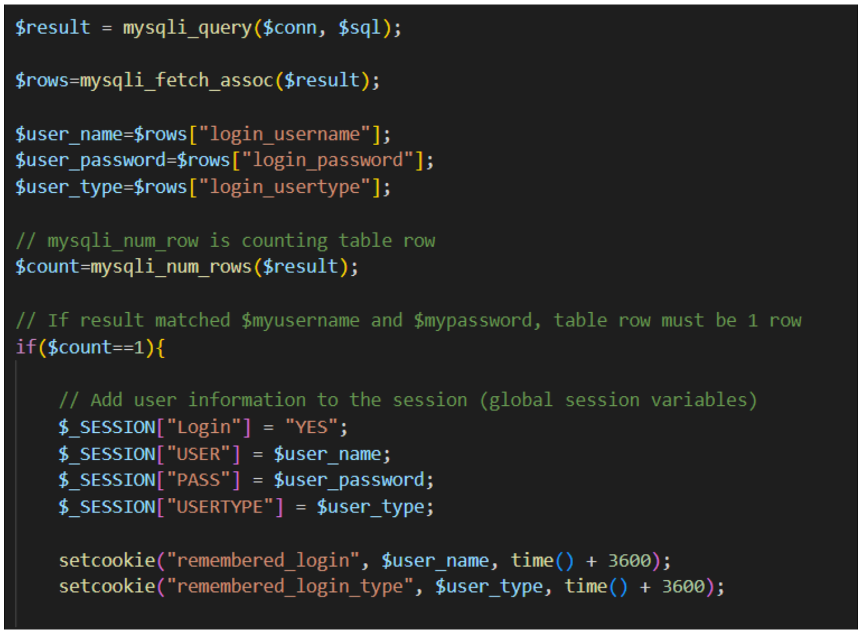
No	Name of the Student Applied	Company Name	View	Result	Delete
1	Nafeesa binti Quraish	Inte	🔍	Rejected	
2	Nafeesa binti Quraish	ABC	🔍	In Review	🗑

Figure 6: Viewing Practical Training Application Page (Student User)

The global session variables are also used in other pages to check the user level to ensure the user logged in has the privilege to see the content within the page as well as displaying the appropriate content and functions for the user to view and use based on the user level.

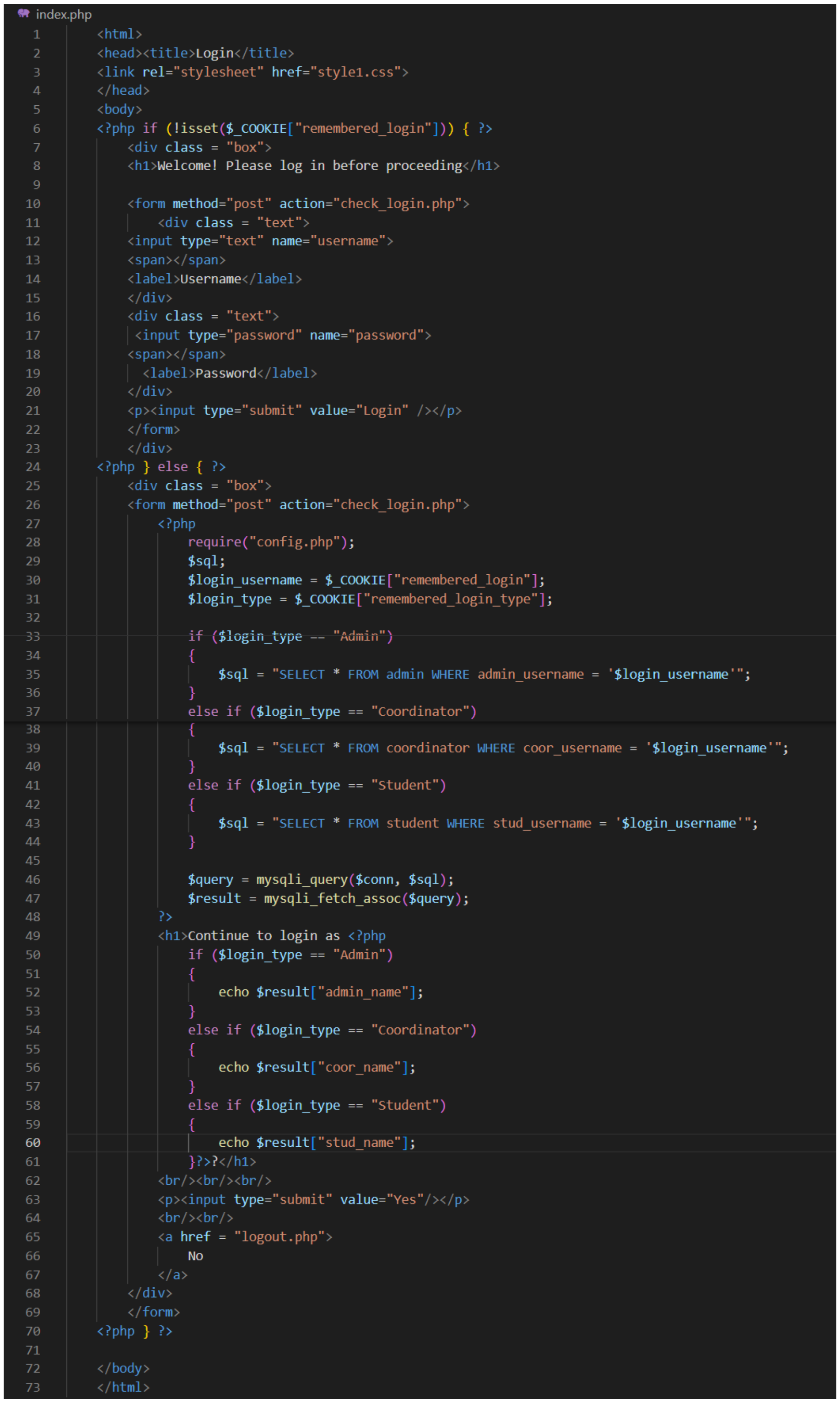
USAGE OF COOKIE

The cookie used in this project is used to record the login user information if the user closes the browser without going through logout. The cookie is set when the user first logged in into the system and with an hour active before the cookie expires. The implementation of cookie in recording login user information allows the user to continue login even if the browser is closed before the expiration time. The code and page related to the implementation of cookie in the project can be shown in the figures below:



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

$rows=mysqli_fetch_assoc($result);

$user_name=$rows["login_username"];
$user_password=$rows["login_password"];
$user_type=$rows["login_usertype"];

// mysqli_num_row is counting table row
$count=mysqli_num_rows($result);

// If result matched $myusername and $mypassword, table row must be 1 row
if($count==1){

    // Add user information to the session (global session variables)
    $_SESSION["Login"] = "YES";
    $_SESSION["USER"] = $user_name;
    $_SESSION["PASS"] = $user_password;
    $_SESSION["USERTYPE"] = $user_type;

    setcookie("remembered_login", $user_name, time() + 3600);
    setcookie("remembered_login_type", $user_type, time() + 3600);
```

Figure 7: Code to Set Cookie when the User Logged In



```
index.php
<html>
<head><title>Login</title>
<link rel="stylesheet" href="style1.css">
</head>
<body>
<?php if (!isset($_COOKIE["remembered_login"])) { ?>
    <div class = "box">
    <h1>Welcome! Please log in before proceeding</h1>

    <form method="post" action="check_login.php">
        <div class = "text">
    <input type="text" name="username">
    <span></span>
    <label>Username</label>
    </div>
    <div class = "text">
     <input type="password" name="password">
    <span></span>
      <label>Password</label>
    </div>
    <p><input type="submit" value="Login" /></p>
    </form>
    </div>
<?php } else { ?>
    <div class = "box">
    <form method="post" action="check_login.php">
        <?php
            require("config.php");
            $sql;
            $login_username = $_COOKIE["remembered_login"];
            $login_type = $_COOKIE["remembered_login_type"];

            if ($login_type == "Admin")
            {
                $sql = "SELECT * FROM admin WHERE admin_username = '$login_username'";
            }
            else if ($login_type == "Coordinator")
            {
                $sql = "SELECT * FROM coordinator WHERE coor_username = '$login_username'";
            }
            else if ($login_type == "Student")
            {
                $sql = "SELECT * FROM student WHERE stud_username = '$login_username'";
            }

            $query = mysqli_query($conn, $sql);
            $result = mysqli_fetch_assoc($query);
        ?>
        <h1>Continue to login as <?php
            if ($login_type == "Admin")
            {
                echo $result["admin_name"];
            }
            else if ($login_type == "Coordinator")
            {
                echo $result["coor_name"];
            }
            else if ($login_type == "Student")
            {
                echo $result["stud_name"];
            }?>?</h1>
        <br/><br/><br/>
        <p><input type="submit" value="Yes"/></p>
        <br/><br/>
        <a href = "logout.php">
            No
        </a>
    </div>
    </form>
<?php } ?>

</body>
</html>
```

Figure 8: Code in Displaying Different Login Page when Cookie is Set and not Expired

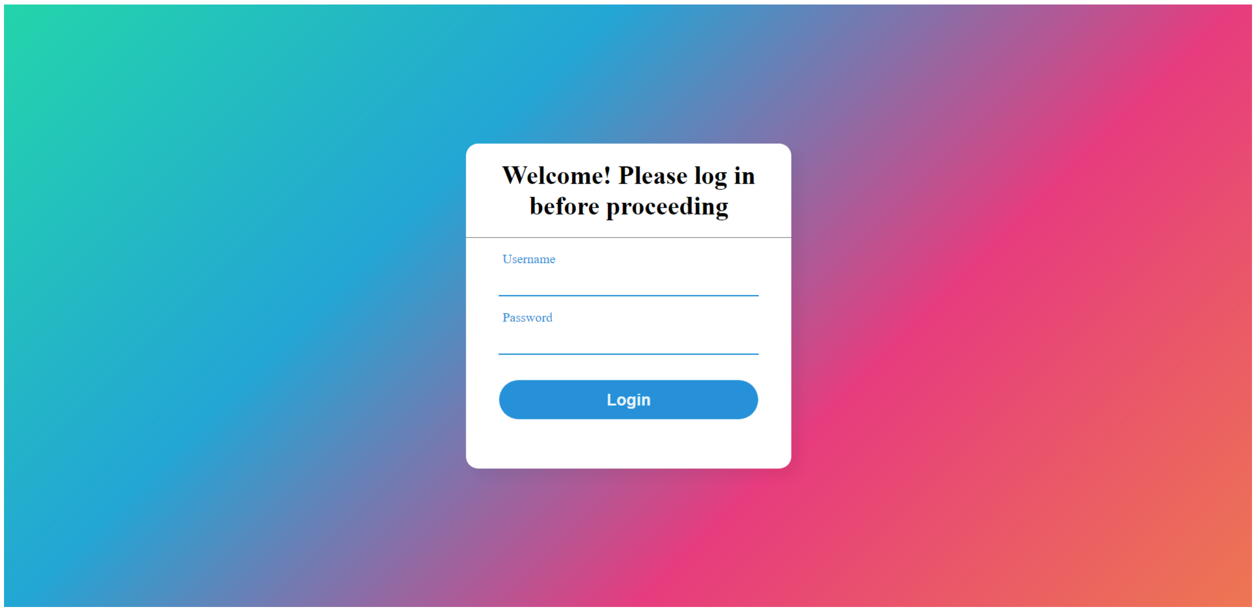


Figure 9: Normal Login Page if the Cookie is not Set or is Expired

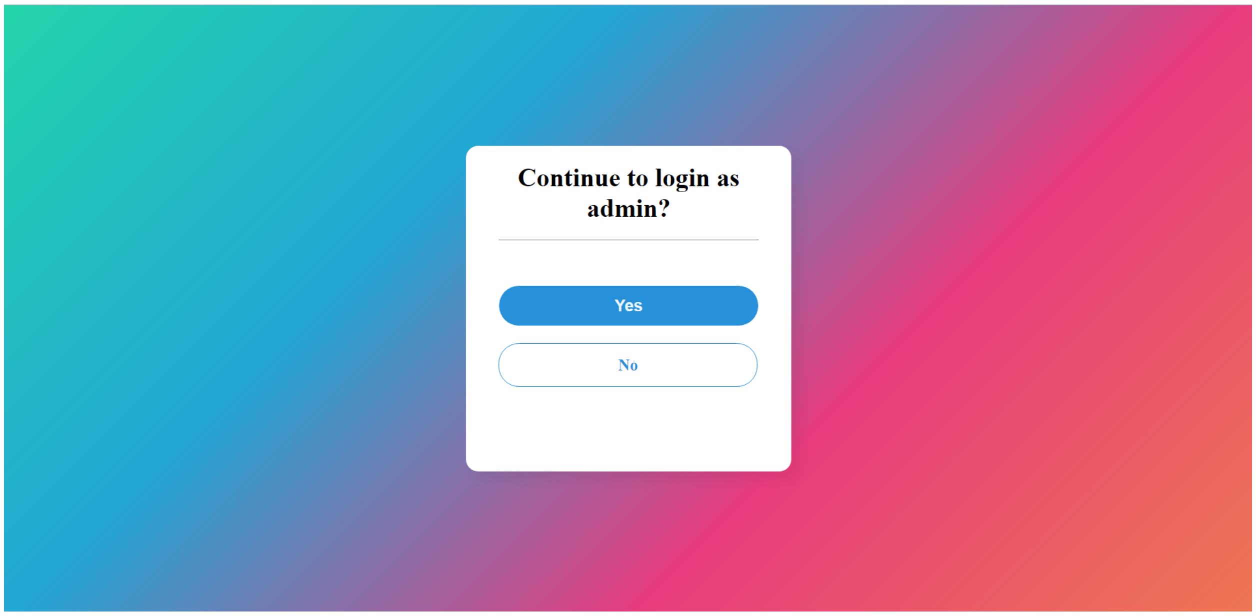
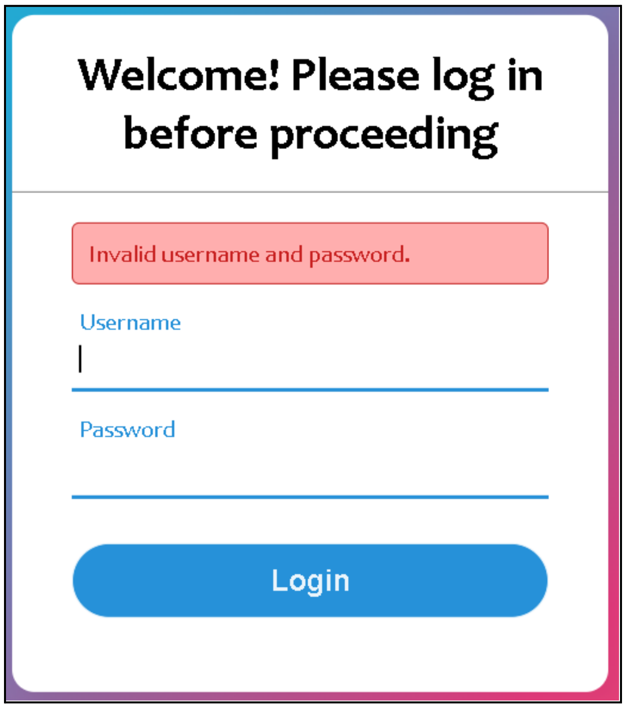


Figure 10: Login Page when Cookie is Set and not Expired

INPUT VALIDATION

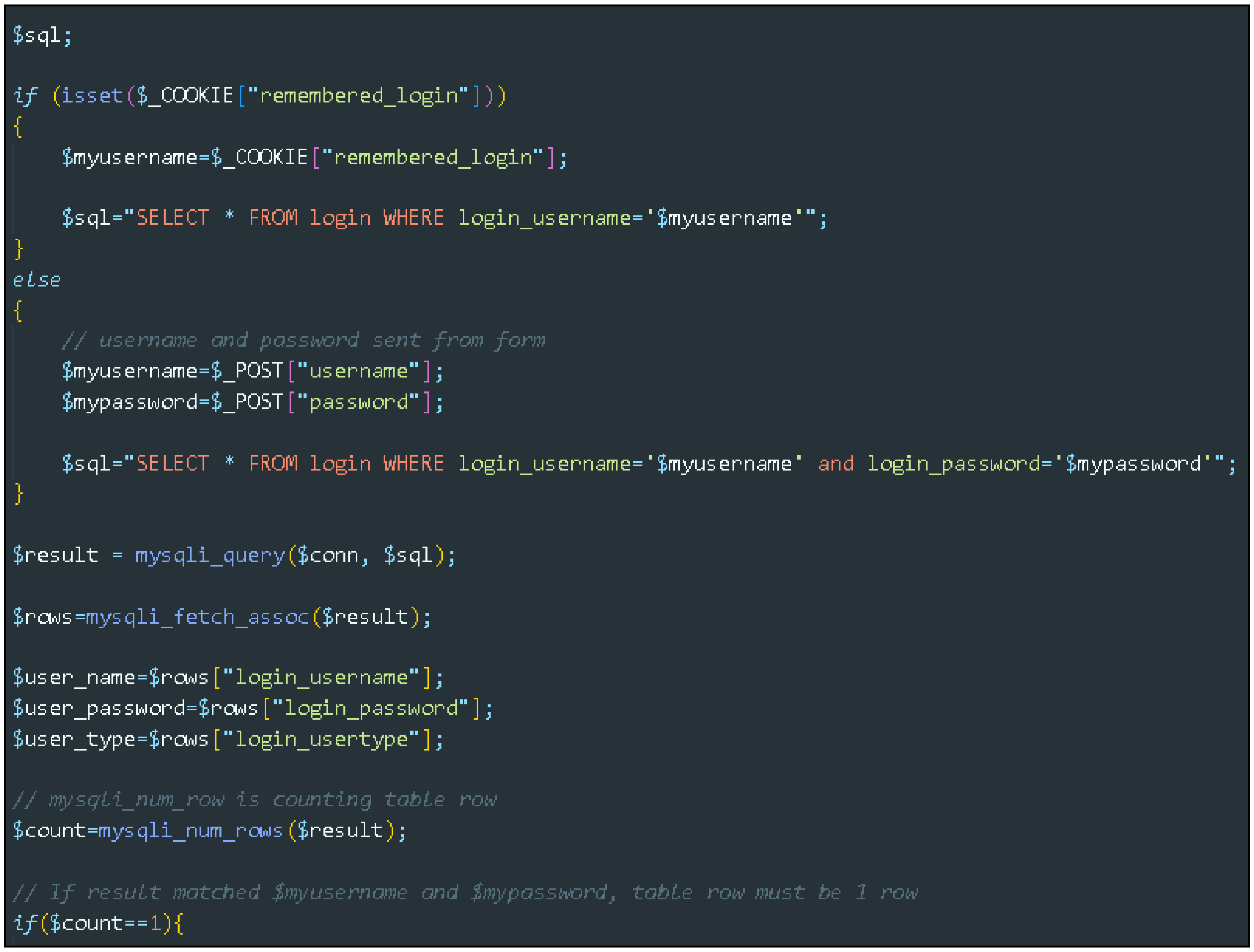
This system contains a lot of validation from login to field format. All of these are done to remind users whenever they input or press some unexpected input. By having these proper validations, it makes our interface more user friendly.

Figure 11 shows the validation of login credentials where the system will check for the matching username and password. If any of the inputs (i.e. username or password) are not matching with the username and password in our database, then the error message will be displayed. The source code of the validation of login credentials is shown in Figure 12 and 13.


Welcome! Please log in before proceeding

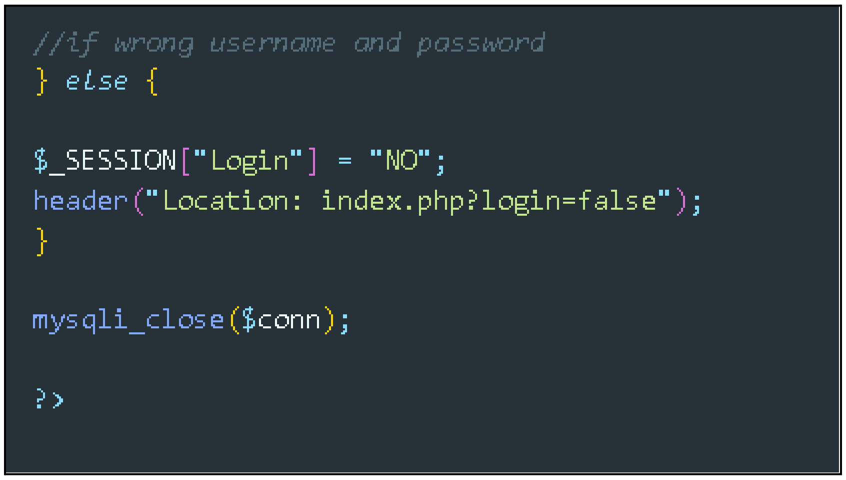
Invalid username and password.

Figure 11: Invalid Username and Password Error Message



```php
$sql;

if (isset($_COOKIE["remembered_login"]))
{
    $myusername=$_COOKIE["remembered_login"];

    $sql="SELECT * FROM login WHERE login_username='$myusername'";
}
else
{
    // username and password sent from form
    $myusername=$_POST["username"];
    $mypassword=$_POST["password"];

    $sql="SELECT * FROM login WHERE login_username='$myusername' and login_password='$mypassword'";
}

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

$rows=mysqli_fetch_assoc($result);

$user_name=$rows["login_username"];
$user_password=$rows["login_password"];
$user_type=$rows["login_usertype"];

// mysqli_num_row is counting table row
$count=mysqli_num_rows($result);

// If result matched $myusername and $mypassword, table row must be 1 row
if($count==1){
```

Figure 12: Source Code for Validation of Login Credentials (1st)

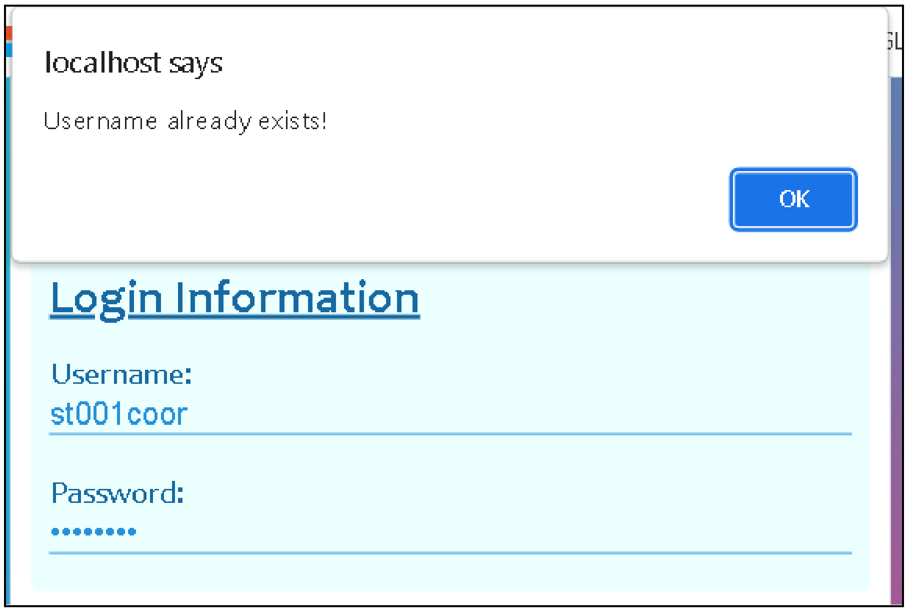


```php
//if wrong username and password
} else {

$_SESSION["Login"] = "NO";
header("Location: index.php?login=false");
}

mysqli_close($conn);

?>
```

Figure 13: Source Code for Validation of Login Credentials (2nd)

After successfully login to the system, the user can use different functionalities. For example, the admin can manage his/her own data, as well as data for other admins, coordinators, and students. Each user will constantly encounter different kinds of error messages if they input wrongly when creating or updating data. For example, when a forgetful admin wants to create a new student that they already did a few minutes ago, then an error message will pop out to remind the admin that the student already exists in the database.

Figure 14, 15, 16, 17 and 18 shows the error message regarding username, staff number, ic, email and phone number respectively. Note that all the examples shown in Figure 14 to 18 is the interface of creating a new coordinator only. However, all these error messages that prompt out when user input similar credentials are also shown in create new student page and create new administrator page, edit admin information, edit coordinator information and edit student information.


localhost says

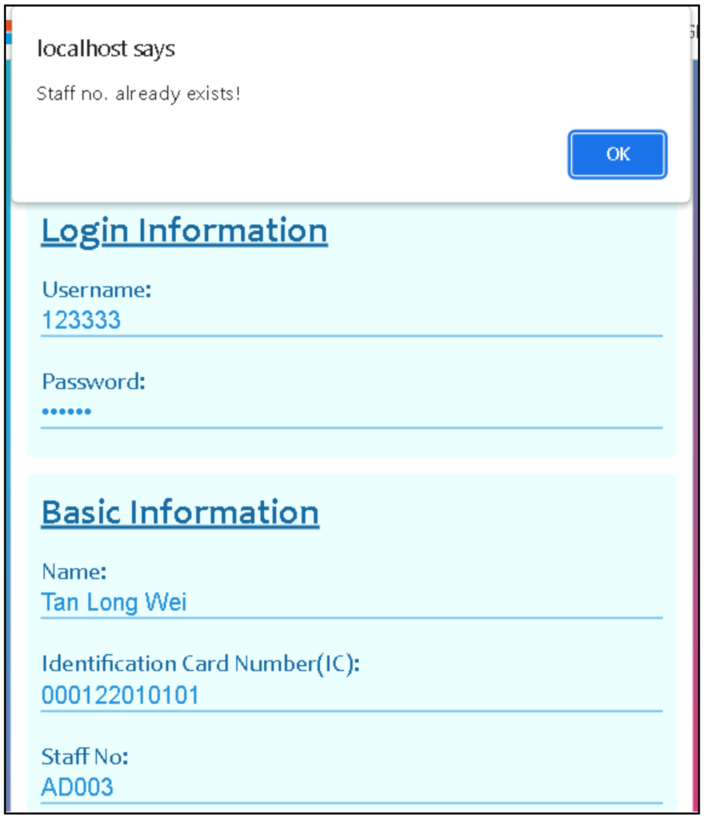
Username already exists!

Login Information

Username:
st001coor

Password:
••••••••

Figure 14: Username Already Exists Error Message


localhost says

Staff no. already exists!

Login Information

Username:
123333

Password:
••••••

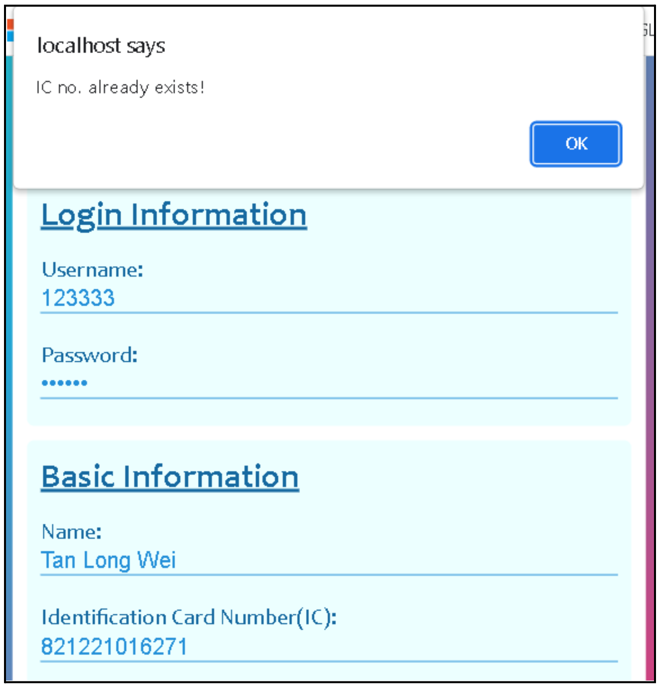
Basic Information

Name:
Tan Long Wei

Identification Card Number(IC):
000122010101

Staff No:
AD003

Figure 15: Staff Number Already Exists Error Message


localhost says

IC no. already exists!

Login Information

Username:
123333

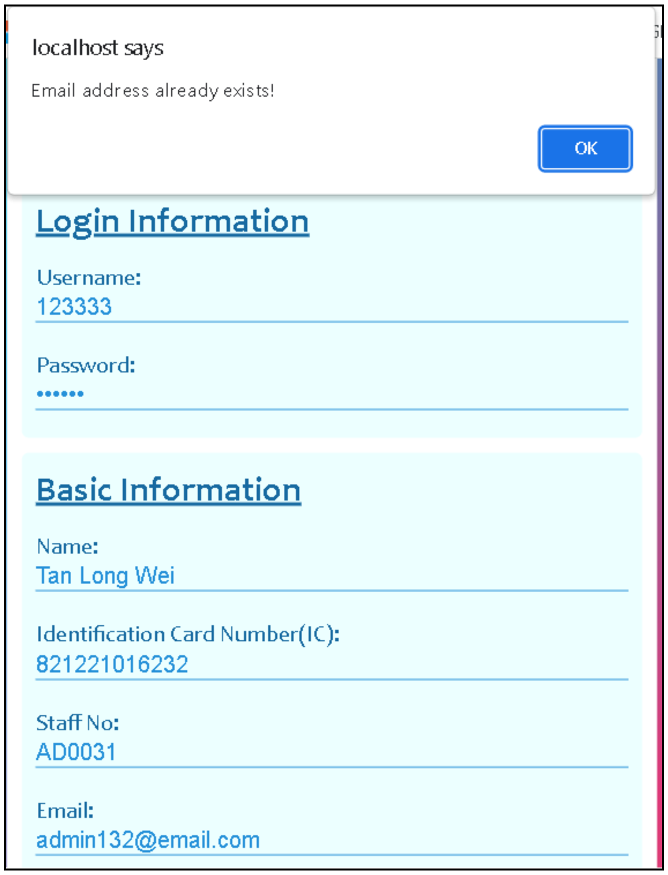
Password:
••••••

Basic Information

Name:
Tan Long Wei

Identification Card Number(IC):
821221016271

Figure 16: IC Number Already Exists Error Message



localhost says

Email address already exists!

Login Information

Username:
123333

Password:
••••••

Basic Information

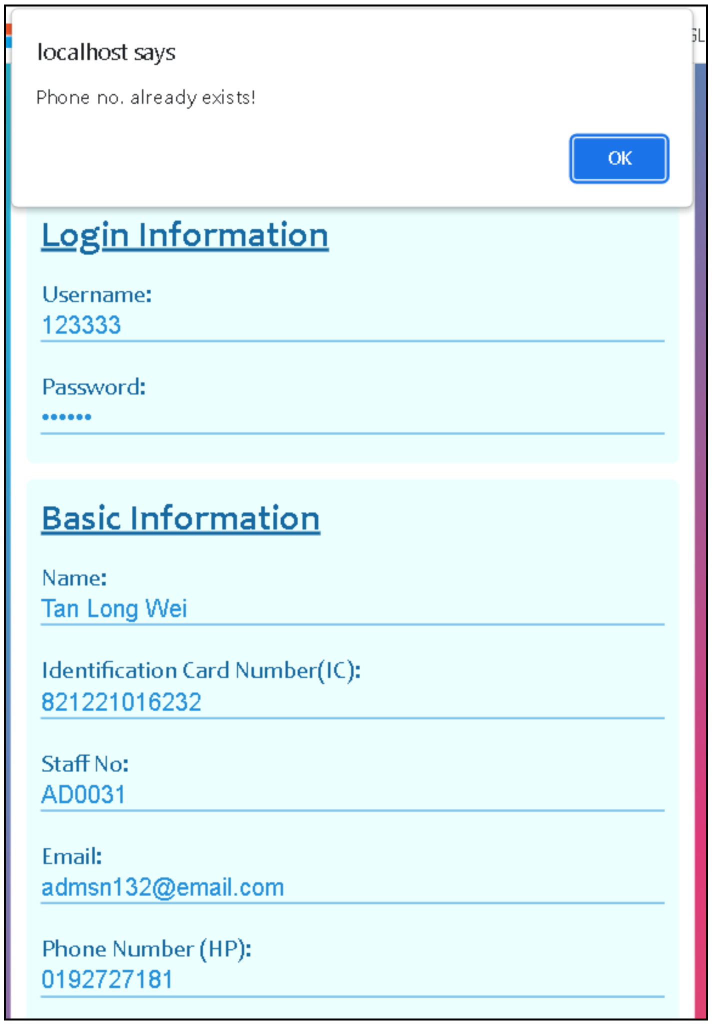
Name:
Tan Long Wei

Identification Card Number(IC):
821221016232

Staff No:
AD0031

Email:
admin132@email.com

Figure 17: Email Address Already Exists Error Message


localhost says

Phone no. already exists!

Login Information

Username:
123333

Password:
••••••

Basic Information

Name:
Tan Long Wei

Identification Card Number(IC):
821221016232

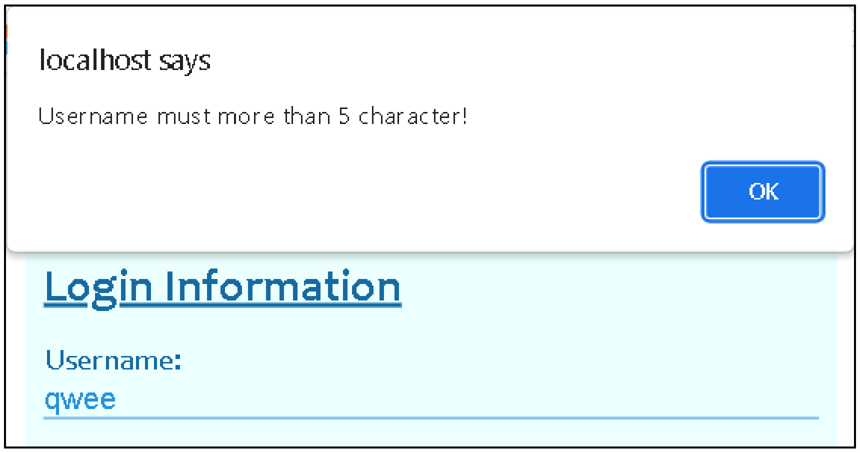
Staff No:
AD0031

Email:
admsn132@email.com

Phone Number (HP):
0192727181

Figure 18: Phone Number Already Exists Error Message

Besides checking for similar data, the system also checks if the user input a valid format. Figure 19, 20, 21, 22, 23 and 24 shows the error message regarding the invalid format of username, password, name, email, phone number and address respectively. Note that all the example shows in Figure 19 to 24 are the interface of create new coordinator only. However, all these error message that prompt out when user input invalid format is also shown in create new student page, create new administrator page, edit admin information, edit coordinator information and edit student information. For creating or editing the training session application, only the email and phone number format will be checked as only these two have certain format to follow.

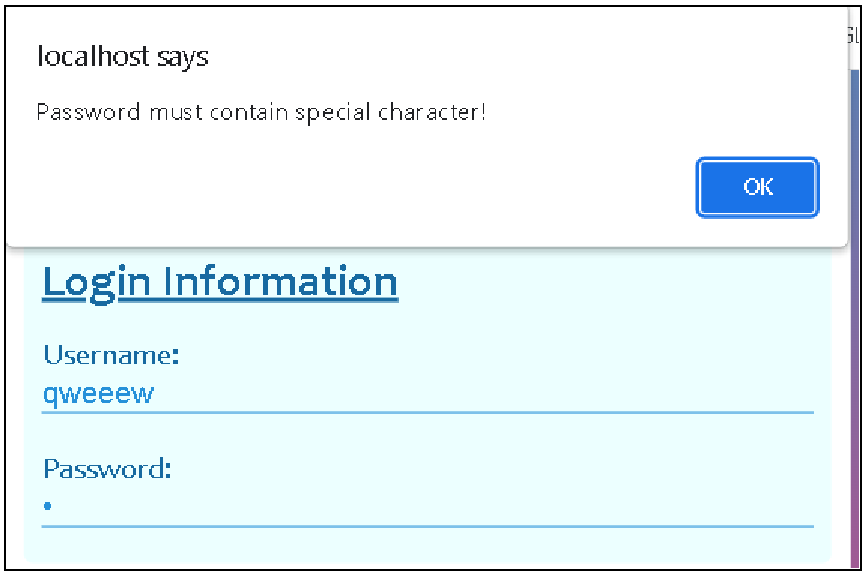

localhost says

Username must more than 5 character!

Login Information

Username:
qwee

Figure 19: Invalid Username Format Error Message


localhost says

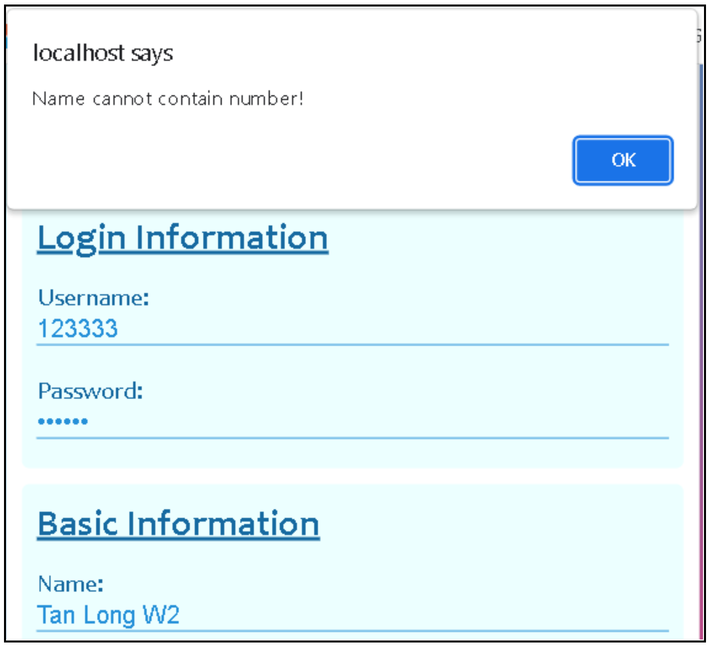
Password must contain special character!

Login Information

Username:
qweeew

Password:
•

Figure 20: Invalid Password Format Error Message


localhost says

Name cannot contain number!

Login Information

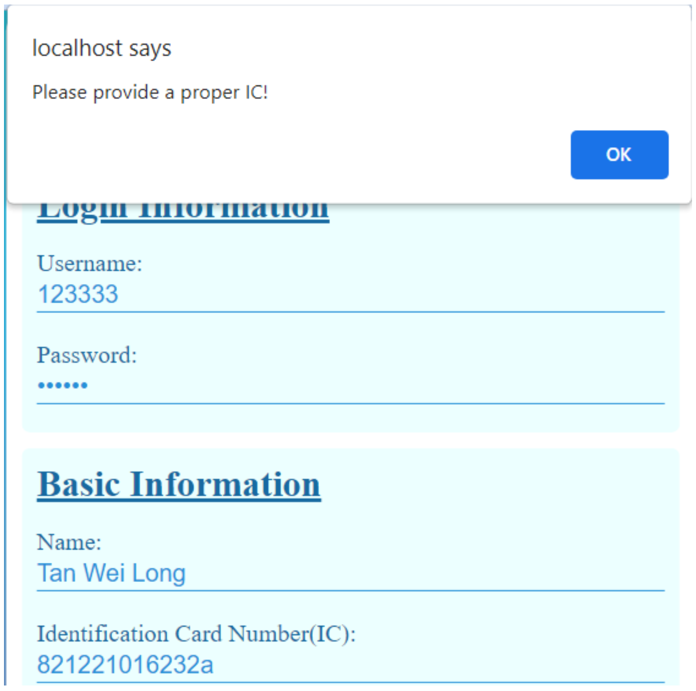
Username:
123333

Password:
••••••

Basic Information

Name:
Tan Long W2

Figure 21: Invalid Name Format Error Message


localhost says

Please provide a proper IC!

Login Information

Username:
123333

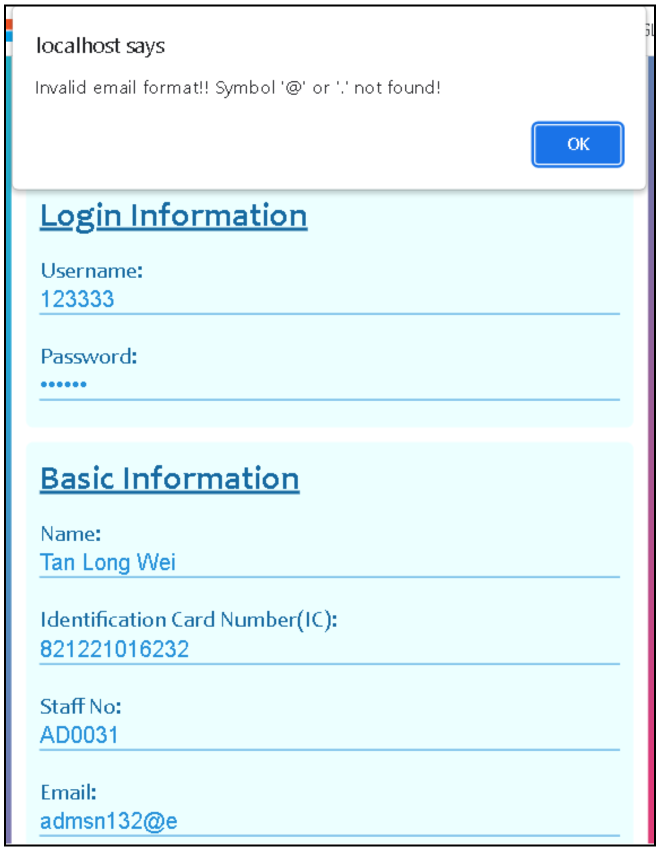
Password:
••••••

Basic Information

Name:
Tan Wei Long

Identification Card Number(IC):
821221016232a

Figure 22: Invalid IC Format Error Message


localhost says

Invalid email format!! Symbol '@' or '.' not found!

Login Information

Username:
123333

Password:
••••••

Basic Information

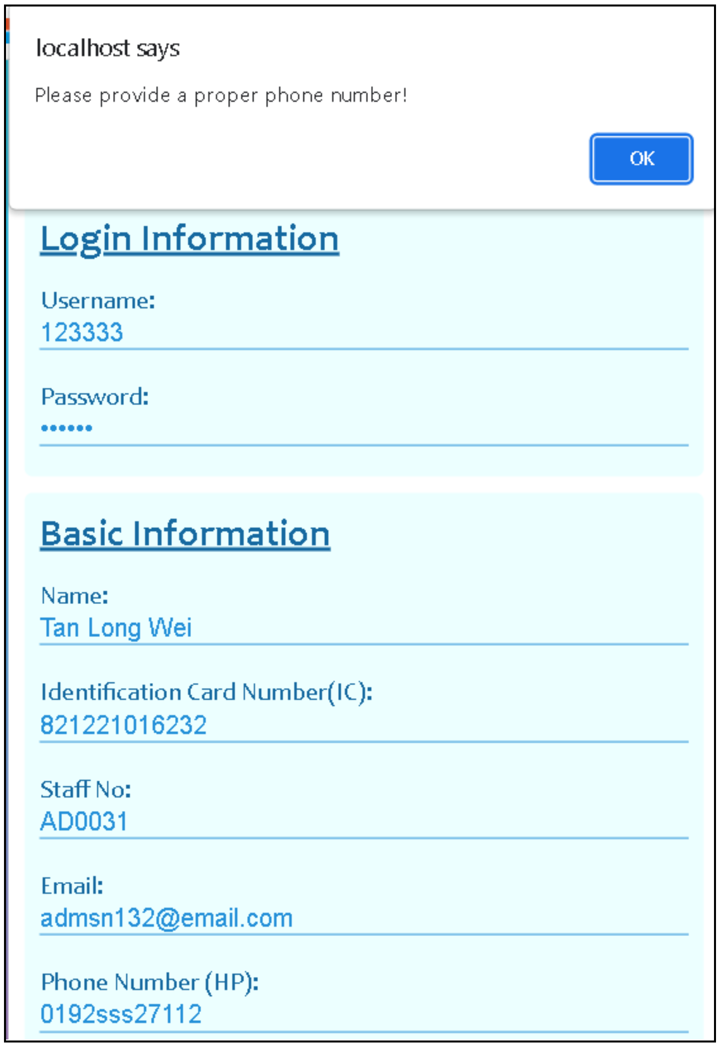
Name:
Tan Long Wei

Identification Card Number(IC):
821221016232

Staff No:
AD0031

Email:
admsn132@e

Figure 23: Invalid Email Format Error Message



localhost says

Please provide a proper phone number!

OK

Login Information

Username:
123333

Password:
••••••

Basic Information

Name:
Tan Long Wei

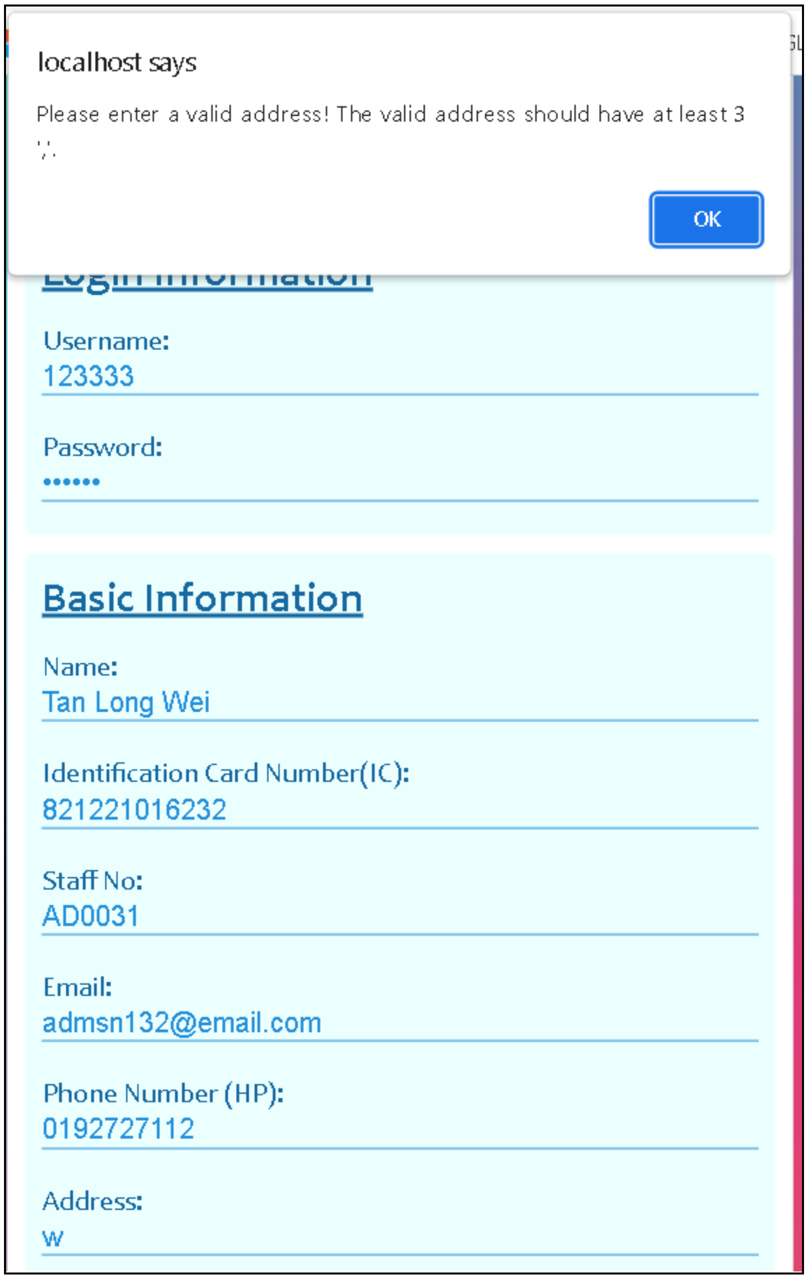
Identification Card Number(IC):
821221016232

Staff No:
AD0031

Email:
admsn132@email.com

Phone Number (HP):
0192sss27112

Figure 24: Invalid Phone Number Format


localhost says

Please enter a valid address! The valid address should have at least 3 ','.

Username:
123333

Password:
••••••

Basic Information

Name:
Tan Long Wei

Identification Card Number(IC):
821221016232

Staff No:
AD0031

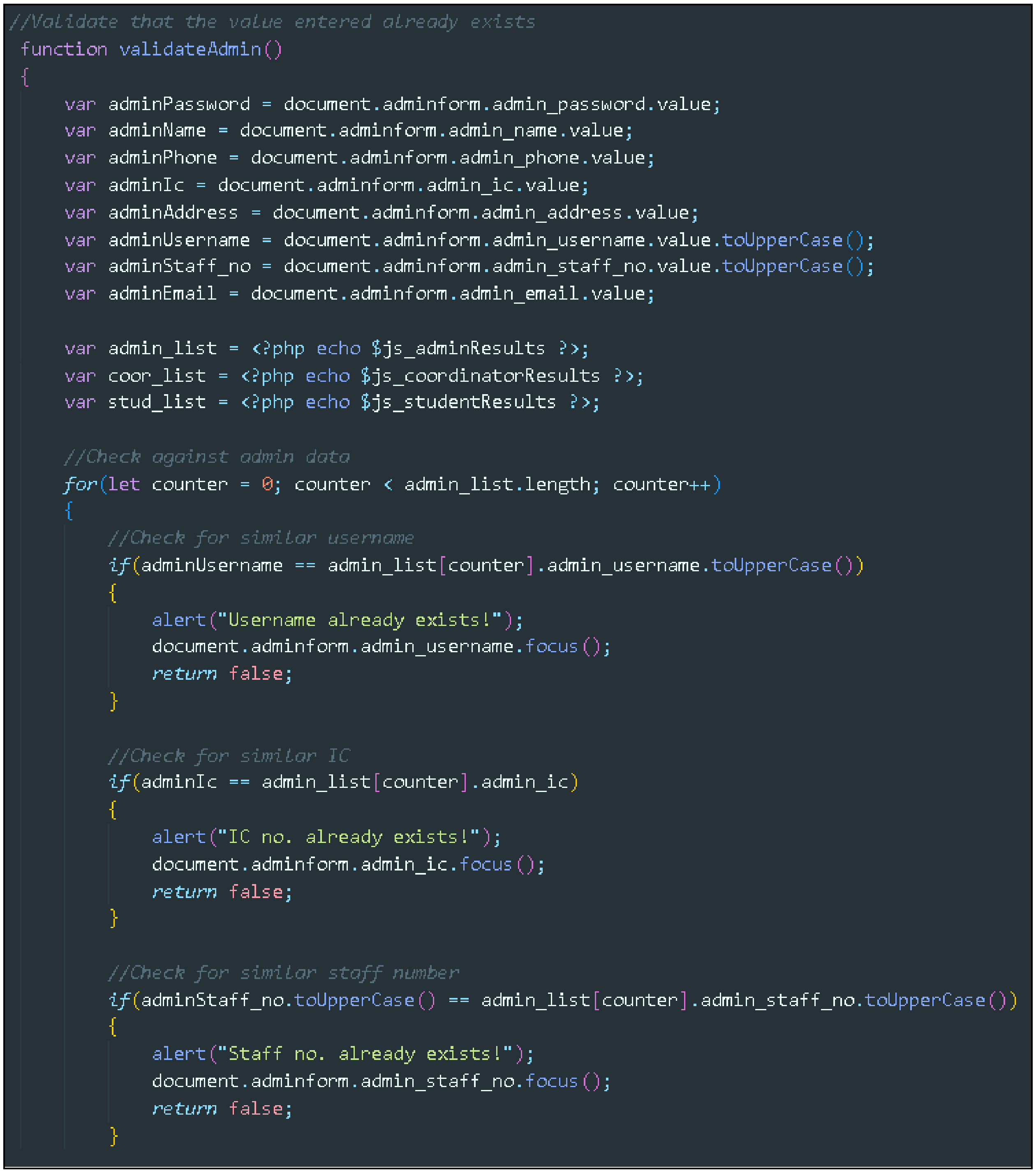
Email:
admsn132@email.com

Phone Number (HP):
0192727112

Address:
w

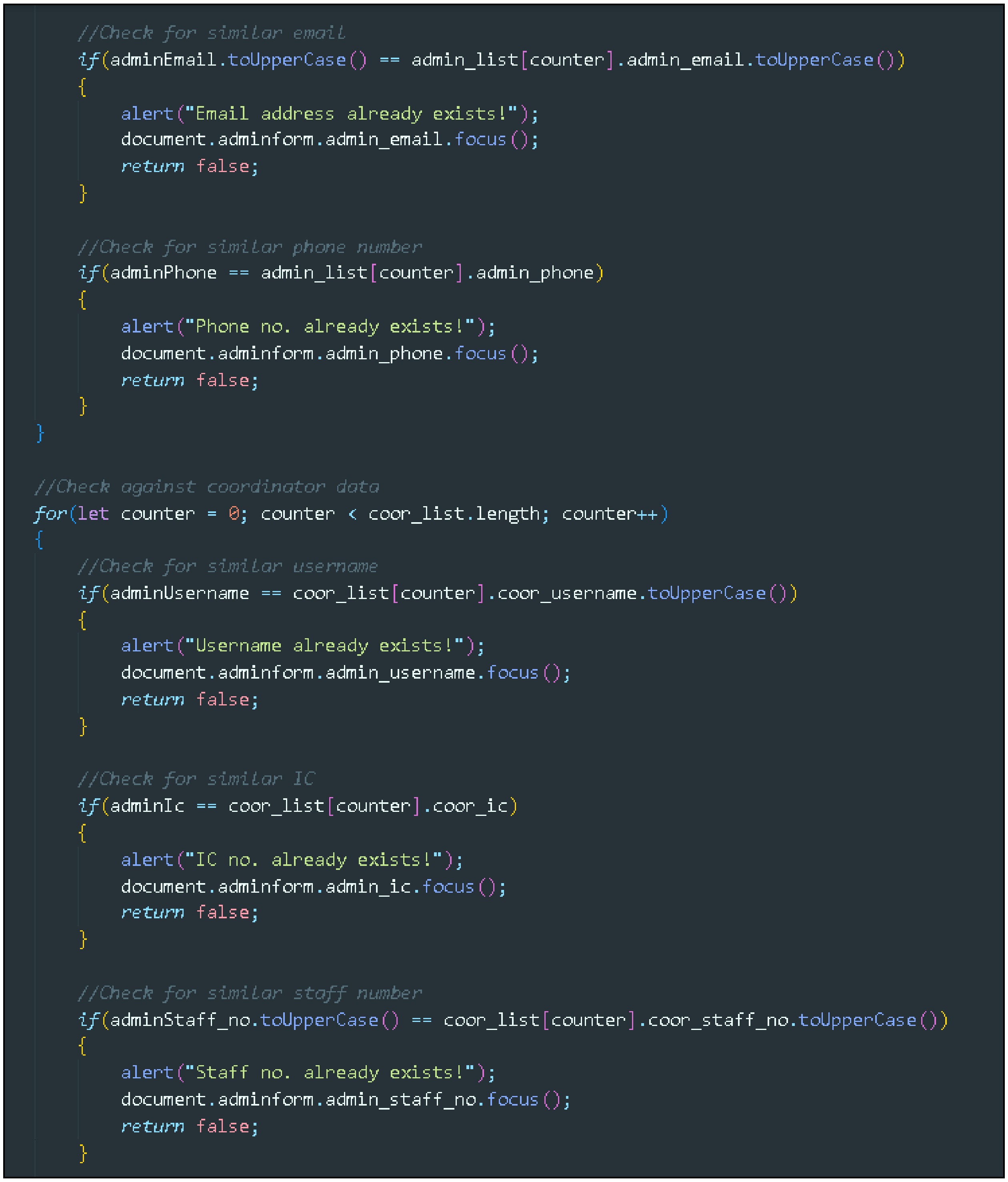
Figure 25: Invalid Email Format

Figure 26, 27, 28, 29 and 30 shows the source code of validating the same input as well as the format. When creating a new user or updating data, the inputted data will be checked against existing admin, coordinator, and student data to prevent duplication of existing data. For example, if the inputted username already exists in the databases, a message will be prompted to the user to use another username. The new username will be checked according to valid format as well, such as it must be more than 5 characters, or else another message will be prompted.



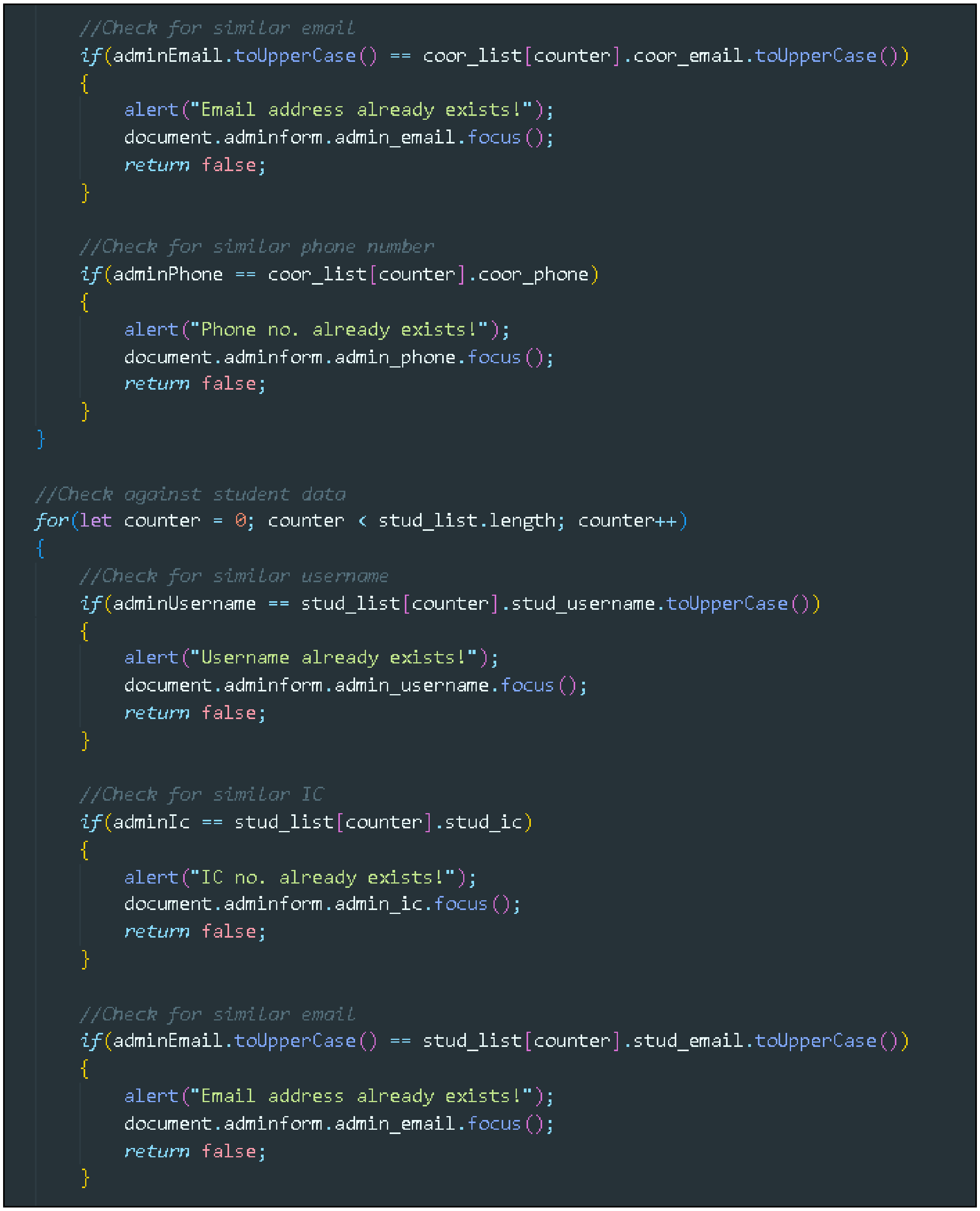
```
//Validate that the value entered already exists
 function validateAdmin()
 {
     var adminPassword = document.adminform.admin_password.value;
     var adminName = document.adminform.admin_name.value;
     var adminPhone = document.adminform.admin_phone.value;
     var adminIc = document.adminform.admin_ic.value;
     var adminAddress = document.adminform.admin_address.value;
     var adminUsername = document.adminform.admin_username.value.toUpperCase();
     var adminStaff_no = document.adminform.admin_staff_no.value.toUpperCase();
     var adminEmail = document.adminform.admin_email.value;

     var admin_list = <?php echo $js_adminResults ?>;
     var coor_list = <?php echo $js_coordinatorResults ?>;
     var stud_list = <?php echo $js_studentResults ?>;

     //Check against admin data
     for(let counter = 0; counter < admin_list.length; counter++)
     {
         //Check for similar username
         if(adminUsername == admin_list[counter].admin_username.toUpperCase())
         {
             alert("Username already exists!");
             document.adminform.admin_username.focus();
             return false;
         }

         //Check for similar IC
         if(adminIc == admin_list[counter].admin_ic)
         {
             alert("IC no. already exists!");
             document.adminform.admin_ic.focus();
             return false;
         }

         //Check for similar staff number
         if(adminStaff_no.toUpperCase() == admin_list[counter].admin_staff_no.toUpperCase())
         {
             alert("Staff no. already exists!");
             document.adminform.admin_staff_no.focus();
             return false;
         }
```

Figure 26: Source Code for Validating Similar Input of Admin Data



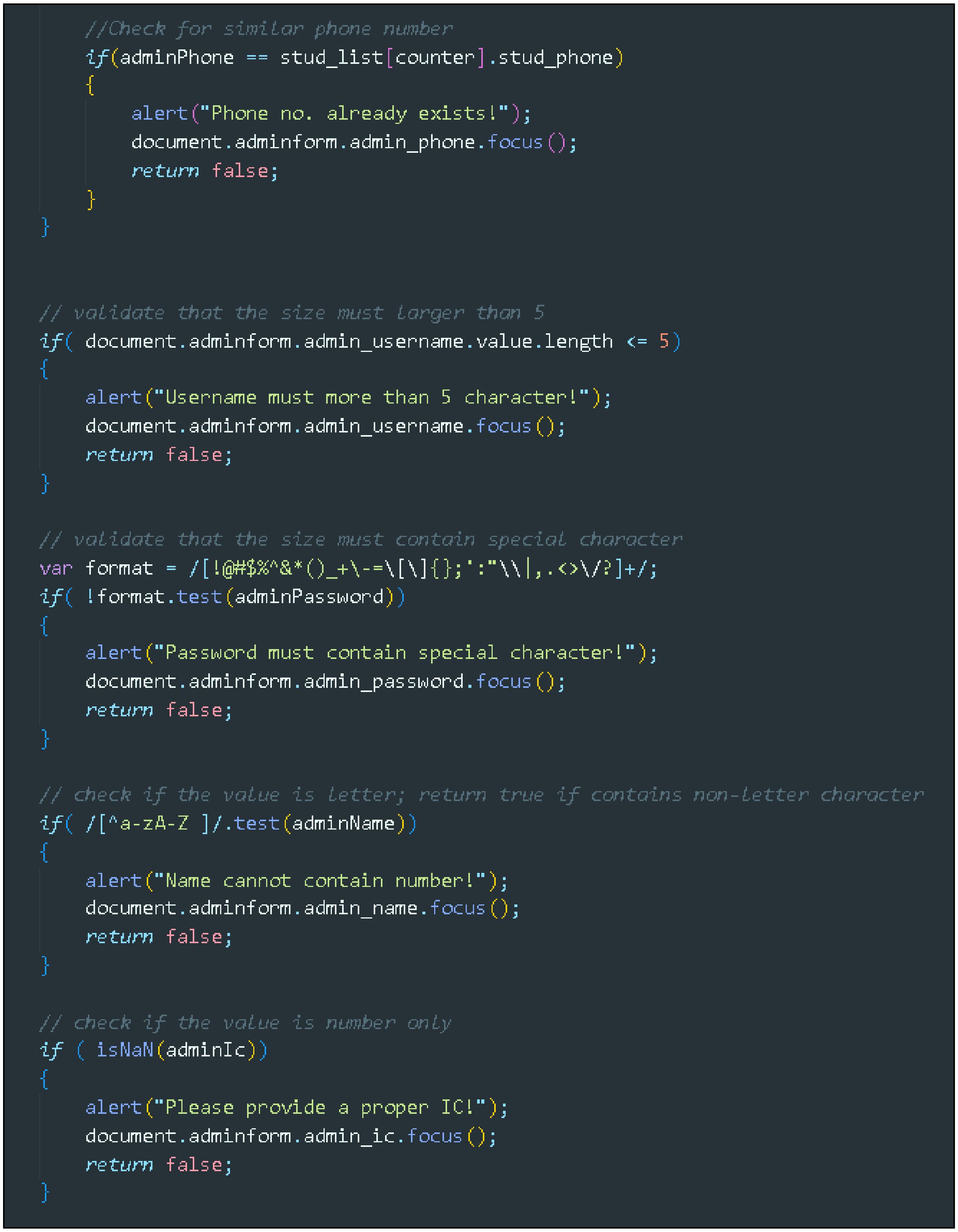
```
        //Check for similar email
        if(adminEmail.toUpperCase() == admin_list[counter].admin_email.toUpperCase())
        {
            alert("Email address already exists!");
            document.adminform.admin_email.focus();
            return false;
        }

        //Check for similar phone number
        if(adminPhone == admin_list[counter].admin_phone)
        {
            alert("Phone no. already exists!");
            document.adminform.admin_phone.focus();
            return false;
        }
    }

    //Check against coordinator data
    for(let counter = 0; counter < coor_list.length; counter++)
    {
        //Check for similar username
        if(adminUsername == coor_list[counter].coor_username.toUpperCase())
        {
            alert("Username already exists!");
            document.adminform.admin_username.focus();
            return false;
        }

        //Check for similar IC
        if(adminIc == coor_list[counter].coor_ic)
        {
            alert("IC no. already exists!");
            document.adminform.admin_ic.focus();
            return false;
        }

        //Check for similar staff number
        if(adminStaff_no.toUpperCase() == coor_list[counter].coor_staff_no.toUpperCase())
        {
            alert("Staff no. already exists!");
            document.adminform.admin_staff_no.focus();
            return false;
        }
```

Figure 27: Source Code for Validating Similar Input of Coordinator Data



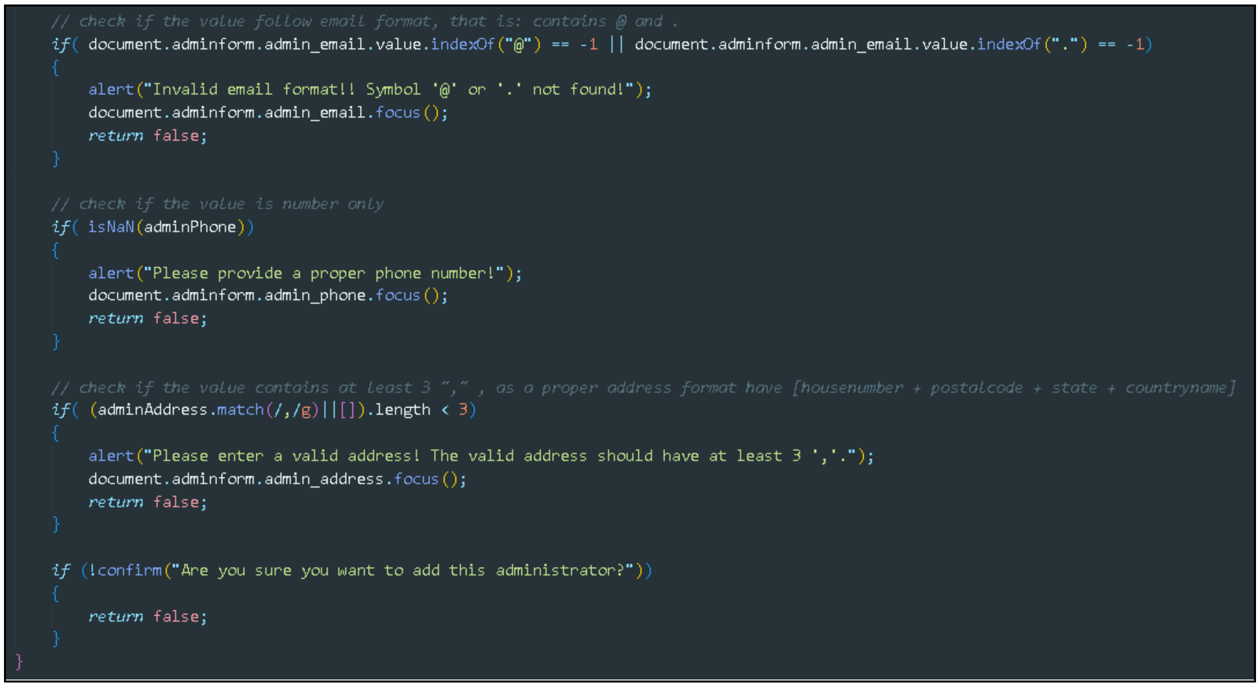
```
        //Check for similar email
        if(adminEmail.toUpperCase() == coor_list[counter].coor_email.toUpperCase())
        {
            alert("Email address already exists!");
            document.adminform.admin_email.focus();
            return false;
        }

        //Check for similar phone number
        if(adminPhone == coor_list[counter].coor_phone)
        {
            alert("Phone no. already exists!");
            document.adminform.admin_phone.focus();
            return false;
        }
    }

    //Check against student data
    for(let counter = 0; counter < stud_list.length; counter++)
    {
        //Check for similar username
        if(adminUsername == stud_list[counter].stud_username.toUpperCase())
        {
            alert("Username already exists!");
            document.adminform.admin_username.focus();
            return false;
        }

        //Check for similar IC
        if(adminIc == stud_list[counter].stud_ic)
        {
            alert("IC no. already exists!");
            document.adminform.admin_ic.focus();
            return false;
        }

        //Check for similar email
        if(adminEmail.toUpperCase() == stud_list[counter].stud_email.toUpperCase())
        {
            alert("Email address already exists!");
            document.adminform.admin_email.focus();
            return false;
        }
```

Figure 28: Source Code for Validation of Student Data



```
        //Check for similar phone number
        if(adminPhone == stud_list[counter].stud_phone)
        {
            alert("Phone no. already exists!");
            document.adminform.admin_phone.focus();
            return false;
        }
    }


    // validate that the size must larger than 5
    if( document.adminform.admin_username.value.length <= 5)
    {
        alert("Username must more than 5 character!");
        document.adminform.admin_username.focus();
        return false;
    }

    // validate that the size must contain special character
    var format = /[!@#$%^&*()_+\-=\[\]{};':"\\|,.<>\/?]+/;
    if( !format.test(adminPassword))
    {
        alert("Password must contain special character!");
        document.adminform.admin_password.focus();
        return false;
    }

    // check if the value is letter; return true if contains non-letter character
    if( /[^a-zA-Z ]/.test(adminName))
    {
        alert("Name cannot contain number!");
        document.adminform.admin_name.focus();
        return false;
    }

    // check if the value is number only
    if ( isNaN(adminIc))
    {
        alert("Please provide a proper IC!");
        document.adminform.admin_ic.focus();
        return false;
    }
```

Figure 29: Source Code for Validation of Format (1st)

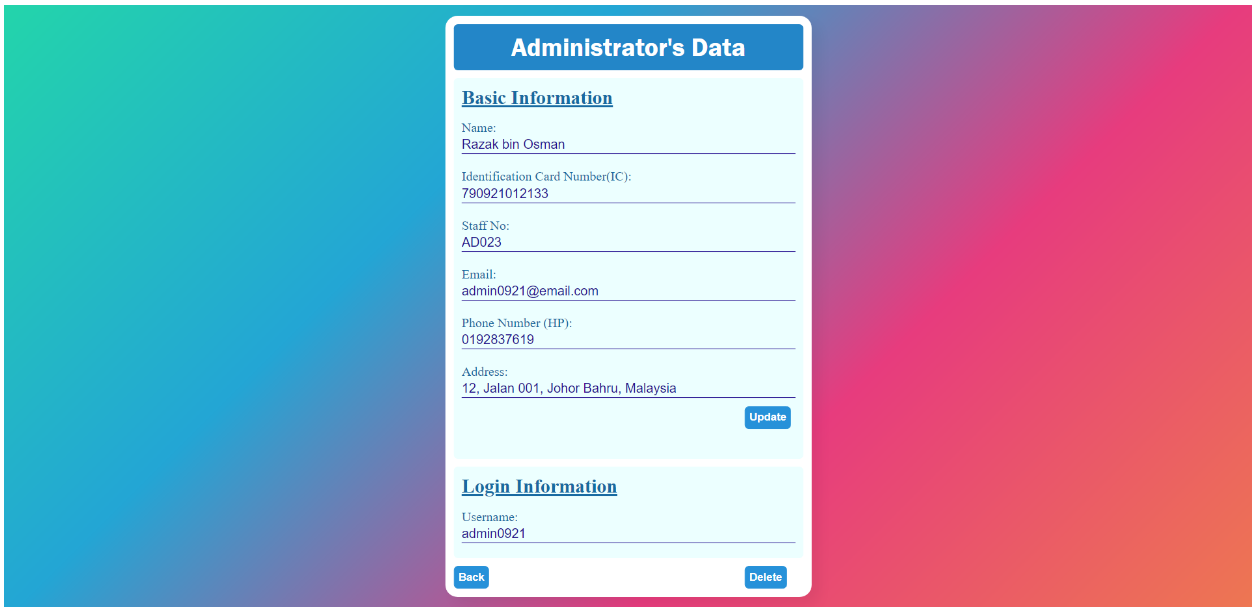


```
    // check if the value follow email format, that is: contains @ and .
    if( document.adminform.admin_email.value.indexOf("@") == -1 || document.adminform.admin_email.value.indexOf(".") == -1)
    {
        alert("Invalid email format!! Symbol '@' or '.' not found!");
        document.adminform.admin_email.focus();
        return false;
    }

    // check if the value is number only
    if( isNaN(adminPhone))
    {
        alert("Please provide a proper phone number!");
        document.adminform.admin_phone.focus();
        return false;
    }

    // check if the value contains at least 3 ",", as a proper address format have [housenumber + postalcode + state + countryname]
    if( (adminAddress.match(/,/g)||[]).length < 3)
    {
        alert("Please enter a valid address! The valid address should have at least 3 ','.");
        document.adminform.admin_address.focus();
        return false;
    }

    if (!confirm("Are you sure you want to add this administrator?"))
    {
        return false;
    }
}
```

Figure 30: Source Code for Validation of Format (2nd)

DYNAMIC HTML

One of the dynamic HTML applied in this project is the adding of another button as well as changing the function of the current button by using JavaScript. For example, the “Update” button when the admin type of users viewing the data of an administrator will allow the user to start editing modifiable information by changing the related input fields’ readonly attribute to false. Besides that, the existing “Update” button will become the “Confirm” button which allows the user to submit the changed data to modify the existing data of the administrator whereas a new button will be displayed to allow the users to cancel the changes made and display back the existing administrator’s data.



Administrator's Data

Basic Information

Name:
Razak bin Osman

Identification Card Number(IC):
790921012133

Staff No:
AD023

Email:
admin0921@email.com

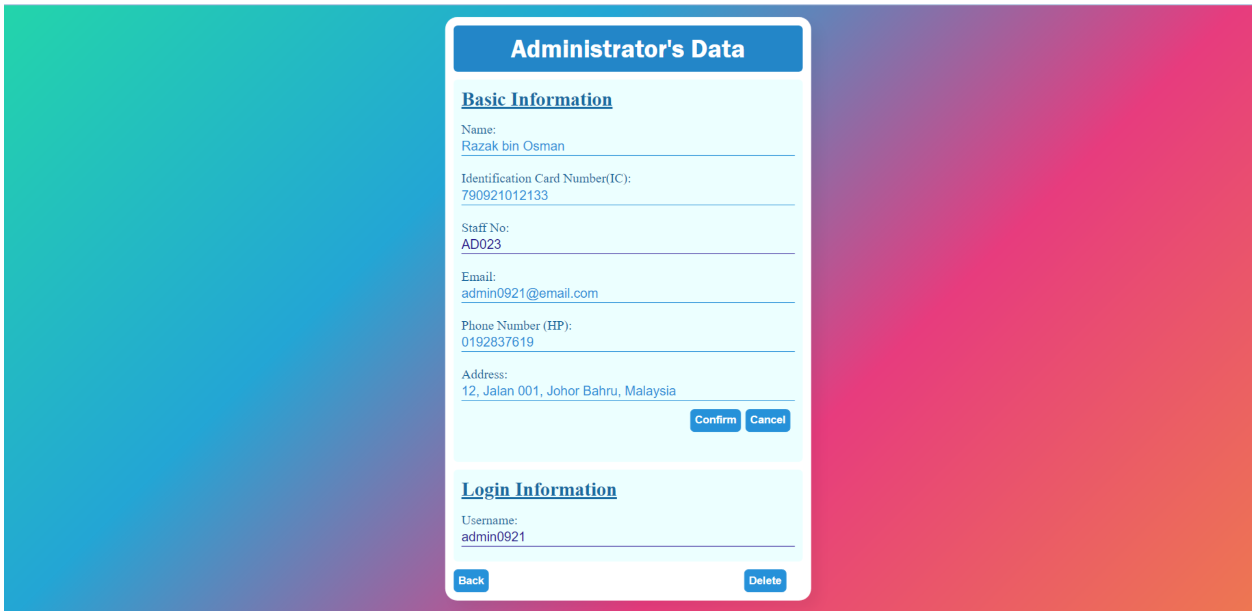
Phone Number (HP):
0192837619

Address:
12, Jalan 001, Johor Bahru, Malaysia

Login Information

Username:
admin0921

Figure 31: Viewing Administrator's Data Page (Before Clicking "Update" Button)



Administrator's Data

Basic Information

Name:
Razak bin Osman

Identification Card Number(IC):
790921012133

Staff No:
AD023

Email:
admin0921@email.com

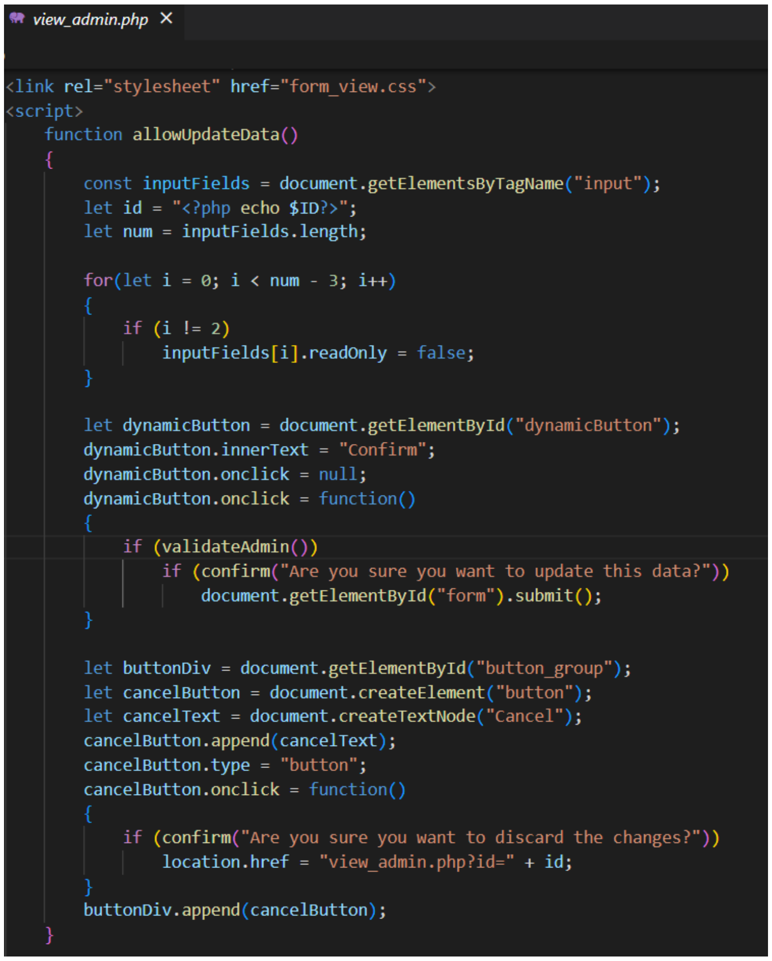
Phone Number (HP):
0192837619

Address:
12, Jalan 001, Johor Bahru, Malaysia

Login Information

Username:
admin0921

Figure 32: Viewing Administrator's Data Page (After Clicking "Update" Button)

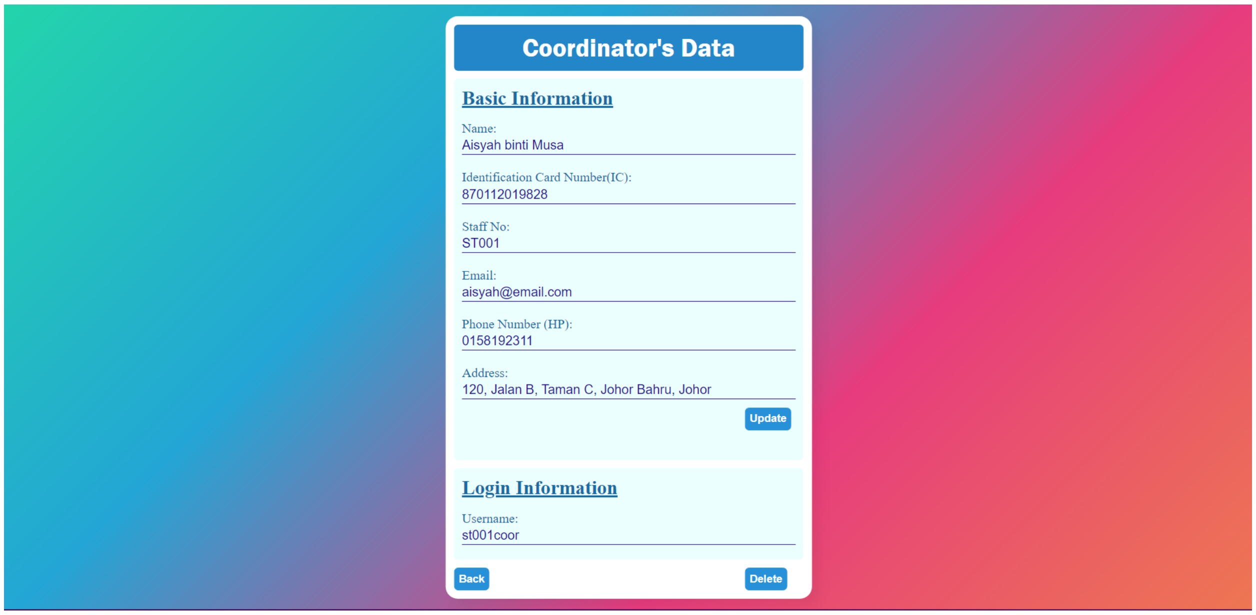


view_admin.php

```
<link rel="stylesheet" href="form_view.css">
<script>
    function allowUpdateData()
    {
        const inputFields = document.getElementsByTagName("input");
        let id = "<?php echo $ID?>";
        let num = inputFields.length;

        for(let i = 0; i < num - 3; i++)
        {
            if (i != 2)
                inputFields[i].readOnly = false;
        }

        let dynamicButton = document.getElementById("dynamicButton");
        dynamicButton.innerText = "Confirm";
        dynamicButton.onclick = null;
        dynamicButton.onclick = function()
        {
            if (validateAdmin())
                if (confirm("Are you sure you want to update this data?"))
                    document.getElementById("form").submit();
        }

        let buttonDiv = document.getElementById("button_group");
        let cancelButton = document.createElement("button");
        let cancelText = document.createTextNode("Cancel");
        cancelButton.append(cancelText);
        cancelButton.type = "button";
        cancelButton.onclick = function()
        {
            if (confirm("Are you sure you want to discard the changes?"))
                location.href = "view_admin.php?id=" + id;
        }
        buttonDiv.append(cancelButton);
    }
```

Figure 33: JavaScript that Implements the Dynamic HTML in view_admin.php

This dynamic HTML also is applied in other pages such as viewing a coordinator, a student or a practical training application by an admin type of user. The code that is used to implement the dynamic HTML and the page displayed are similar to the figures shown above where the data displayed in the page is either the coordinator's, student's or practical training application's data. The dynamic HTML applied on other pages can be shown in the figures below:



Coordinator's Data

Basic Information

Name:
Aisyah binti Musa

Identification Card Number(IC):
870112019828

Staff No:
ST001

Email:
aisyah@email.com

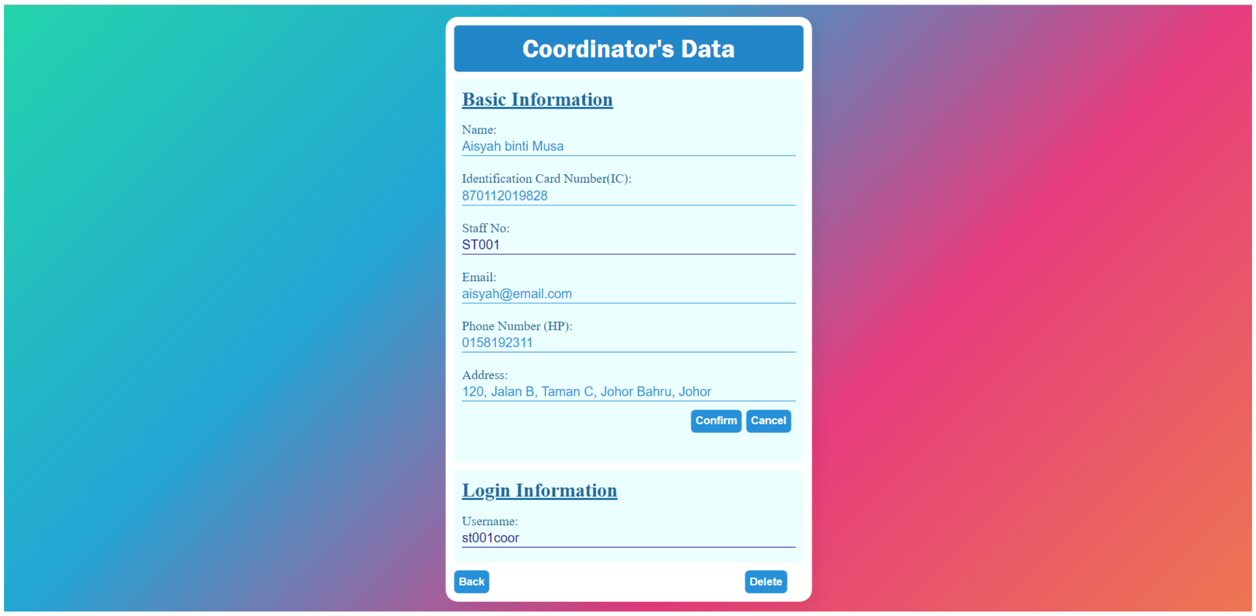
Phone Number (HP):
0158192311

Address:
120, Jalan B, Taman C, Johor Bahru, Johor

Login Information

Username:
st001coor

Figure 34: Viewing Coordinator's Data Page (Before Clicking "Update" Button)



Coordinator's Data

Basic Information

Name:
Aisyah binti Musa

Identification Card Number(IC):
870112019828

Staff No:
ST001

Email:
aisyah@email.com

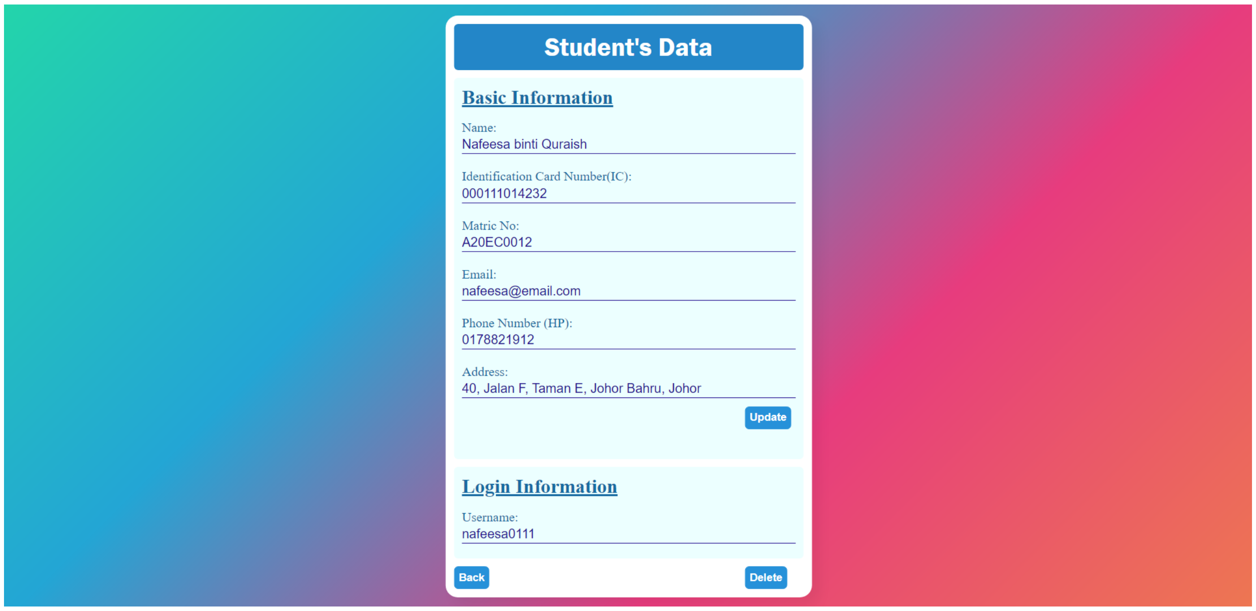
Phone Number (HP):
0158192311

Address:
120, Jalan B, Taman C, Johor Bahru, Johor

Login Information

Username:
st001coor

Figure 35: Viewing Coordinator's Data Page (After Clicking "Update" Button)



Student's Data

Basic Information

Name:
Nafeesa binti Quraish

Identification Card Number(IC):
000111014232

Matric No:
A20EC0012

Email:
nafeesa@email.com

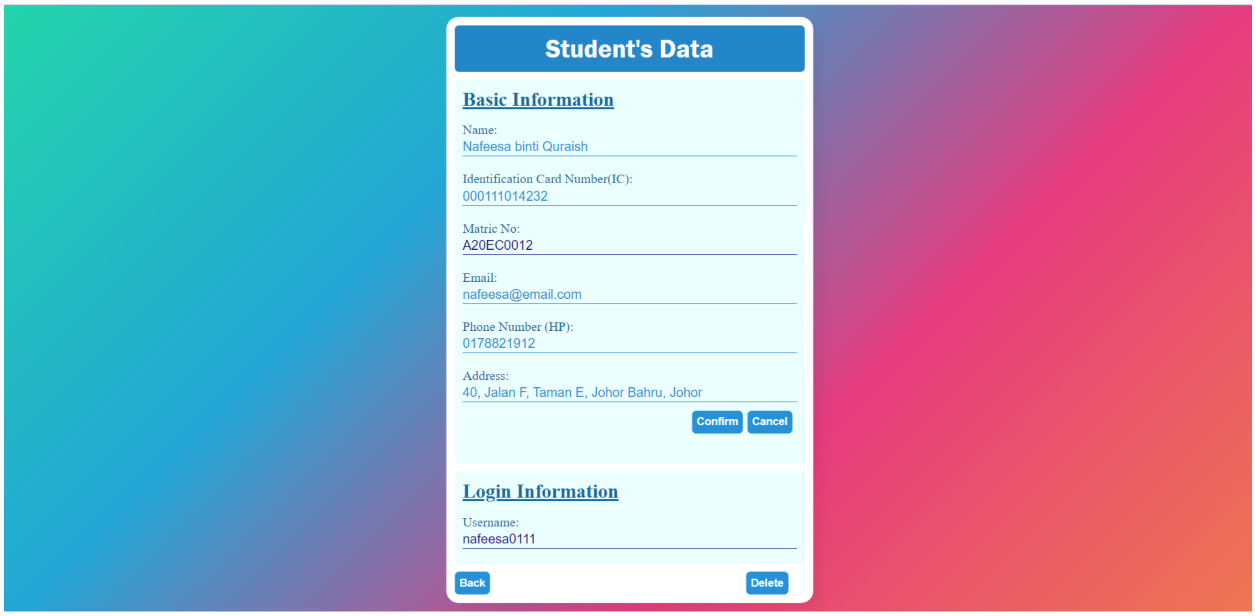
Phone Number (HP):
0178821912

Address:
40, Jalan F, Taman E, Johor Bahru, Johor

Login Information

Username:
nafeesa0111

Figure 36: Viewing Student's Data Page (Before Clicking "Update" Button)



Student's Data

Basic Information

Name:
Nafeesa binti Quraish

Identification Card Number(IC):
000111014232

Matric No:
A20EC0012

Email:
nafeesa@email.com

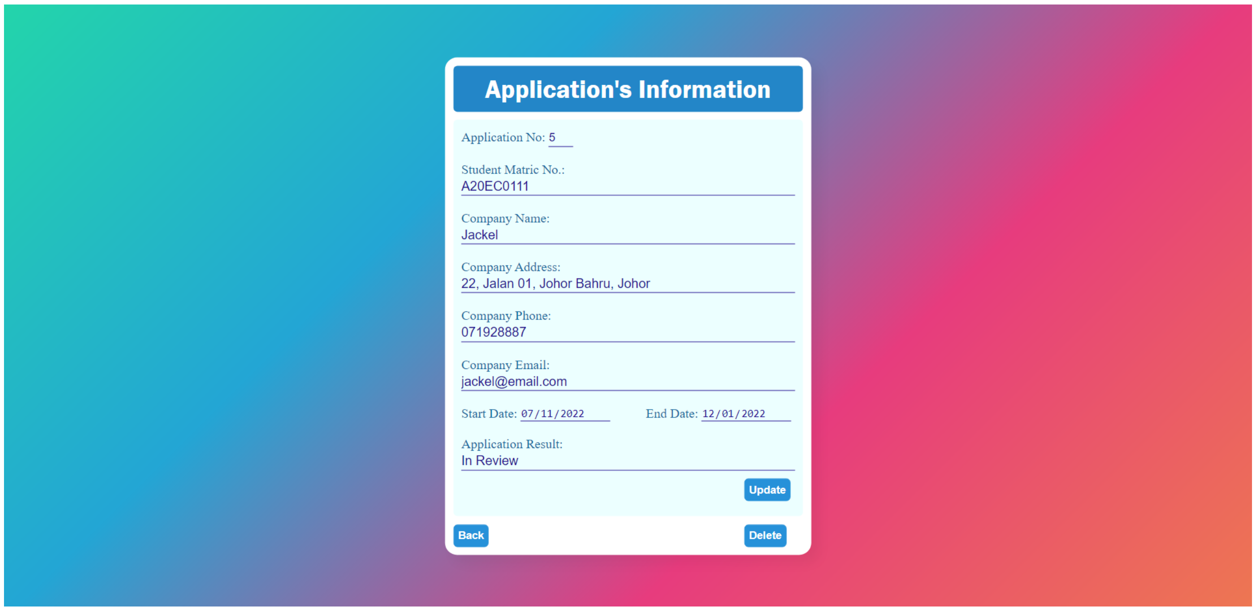
Phone Number (HP):
0178821912

Address:
40, Jalan F, Taman E, Johor Bahru, Johor

Login Information

Username:
nafeesa0111

Figure 37: Viewing Student's Data Page (After Clicking "Update" Button)


Application's Information

Application No: 5

Student Matric No.:
A20EC0111

Company Name:
Jackel

Company Address:
22, Jalan 01, Johor Bahru, Johor

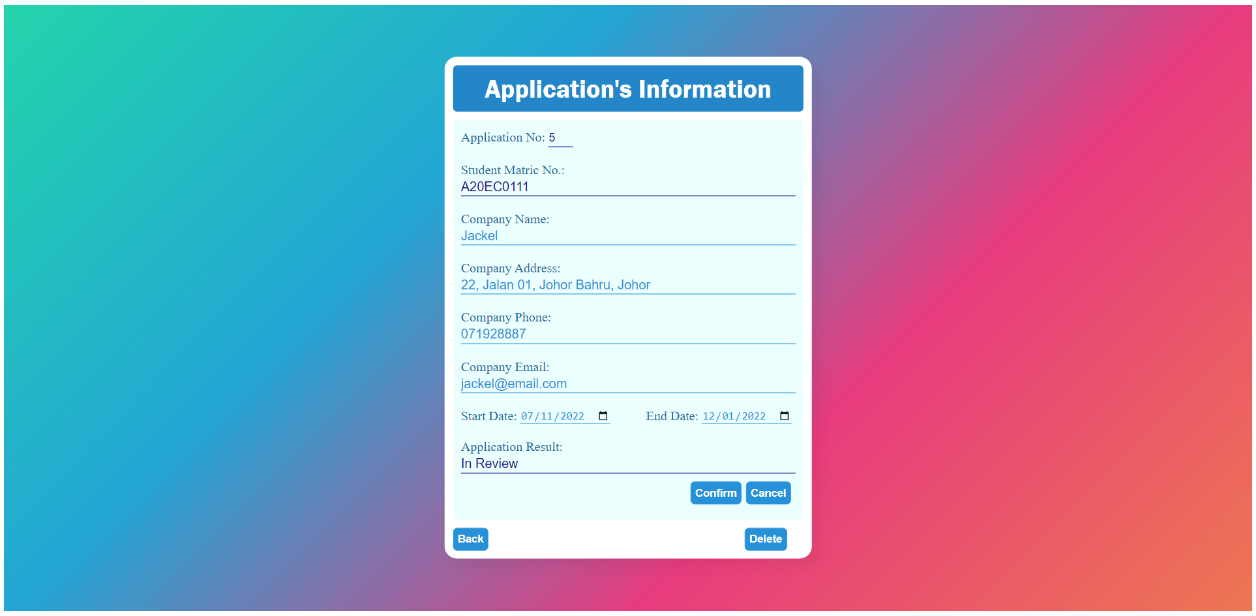
Company Phone:
071928887

Company Email:
jackel@email.com

Start Date: 07/11/2022 End Date: 12/01/2022

Application Result:
In Review

Figure 38: Viewing Application's Data Page (Before Clicking "Update" Button)


Application's Information

Application No: 5

Student Matric No.:
A20EC0111

Company Name:
Jackel

Company Address:
22, Jalan 01, Johor Bahru, Johor

Company Phone:
071928887

Company Email:
jackel@email.com

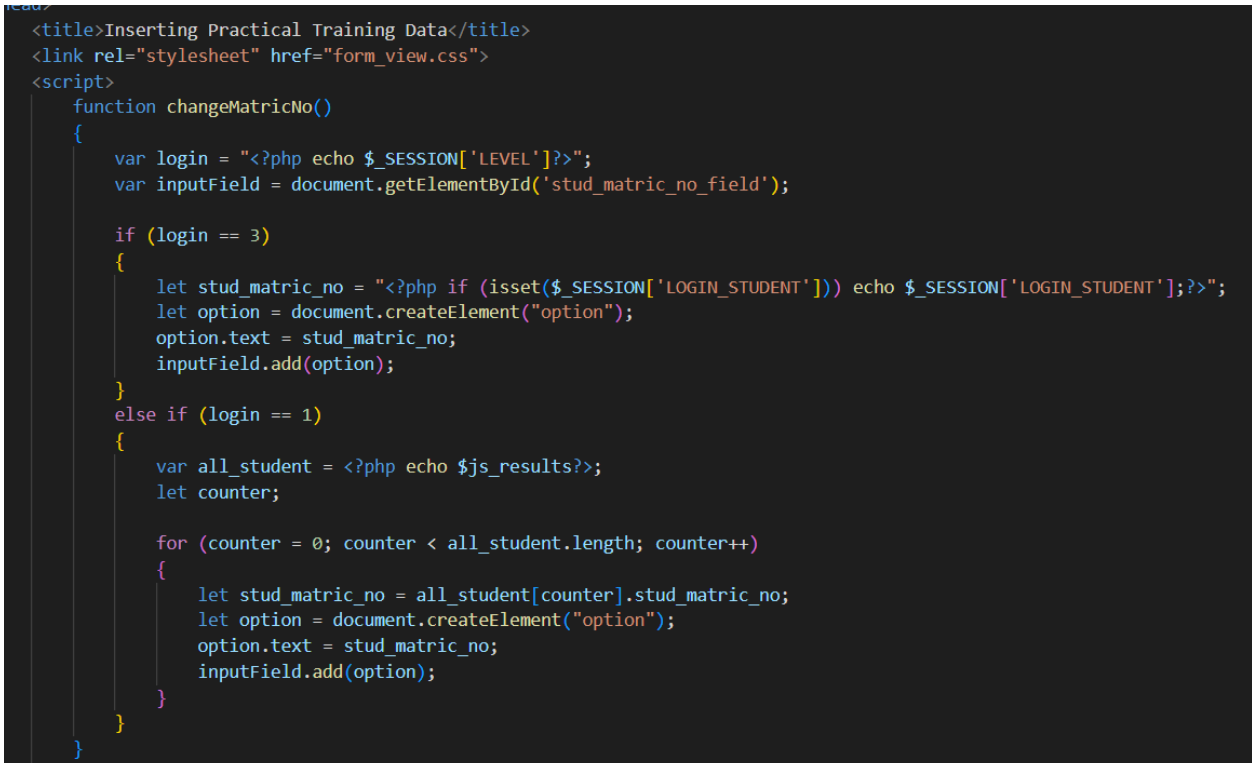
Start Date: 07/11/2022 End Date: 12/01/2022

Application Result:
In Review

Figure 39: Viewing Application's Data Page (After Clicking "Update" Button)

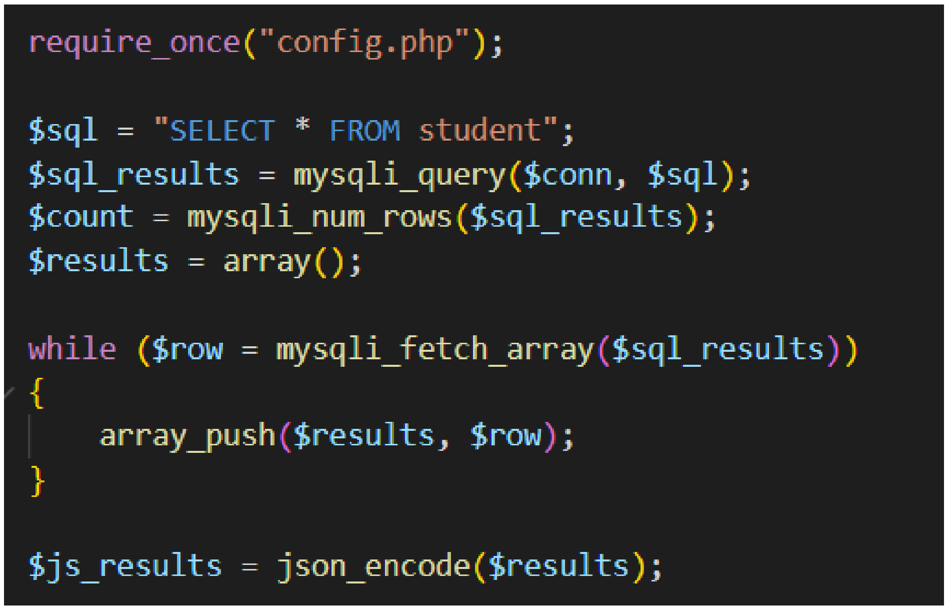
Another dynamic HTML applied in the project is by adding different options for the user based on the login user level when inserting a new practical training application. For the admin type of user, the user can select all students whereas the student type of users can only select their matric number when applying for a training session. The implementation is a combination of using JavaScript to alter the number of selections and PHP to fetch the data from the database.

The code of implementing this dynamic HTML and the page related to this dynamic HTML can be shown in the figures below:



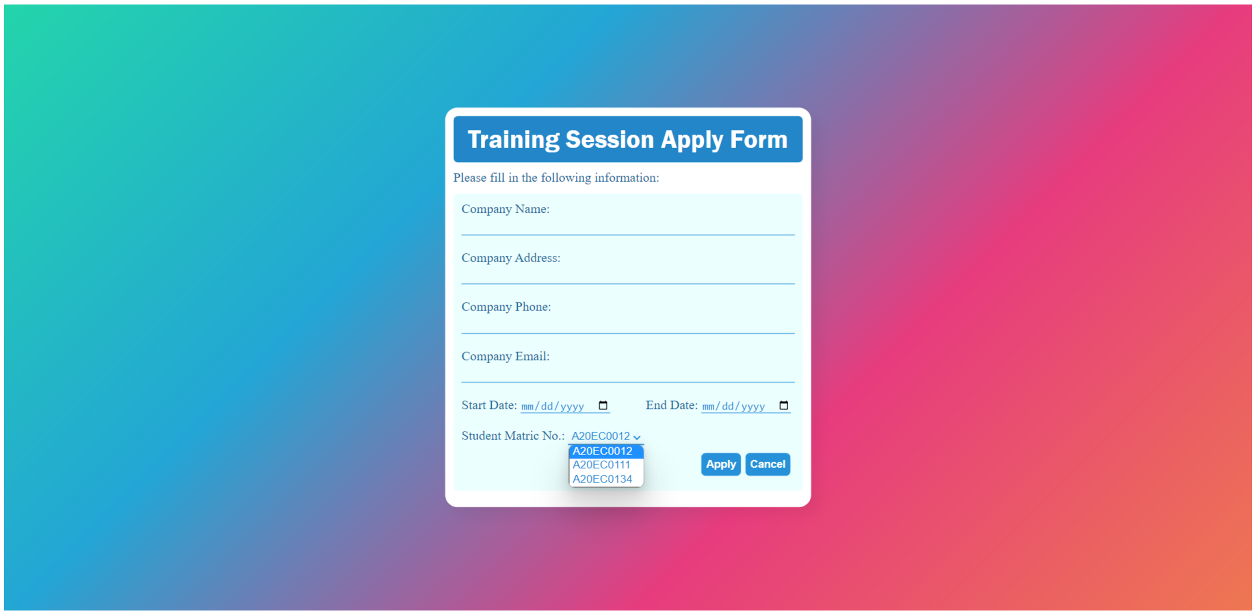
```
    <title>Inserting Practical Training Data</title>
    <link rel="stylesheet" href="form_view.css">
    <script>
        function changeMatricNo()
        {
            var login = "<?php echo $_SESSION['LEVEL']?>";
            var inputField = document.getElementById('stud_matric_no_field');

            if (login == 3)
            {
                let stud_matric_no = "<?php if (isset($_SESSION['LOGIN_STUDENT'])) echo $_SESSION['LOGIN_STUDENT'];?>";
                let option = document.createElement("option");
                option.text = stud_matric_no;
                inputField.add(option);
            }
            else if (login == 1)
            {
                var all_student = <?php echo $js_results?>;
                let counter;

                for (counter = 0; counter < all_student.length; counter++)
                {
                    let stud_matric_no = all_student[counter].stud_matric_no;
                    let option = document.createElement("option");
                    option.text = stud_matric_no;
                    inputField.add(option);
                }
            }
        }
```

Figure 40: JavaScript to Change the Selection of Student Matric Number based on Logged In User Type



```
require_once("config.php");

$sql = "SELECT * FROM student";
$sql_results = mysqli_query($conn, $sql);
$count = mysqli_num_rows($sql_results);
$results = array();

while ($row = mysqli_fetch_array($sql_results))
{
    array_push($results, $row);
}

$js_results = json_encode($results);
```

Figure 41: PHP to Fetch Data to be Used in JavaScript


Training Session Apply Form

Please fill in the following information:

Company Name:

Company Address:

Company Phone:

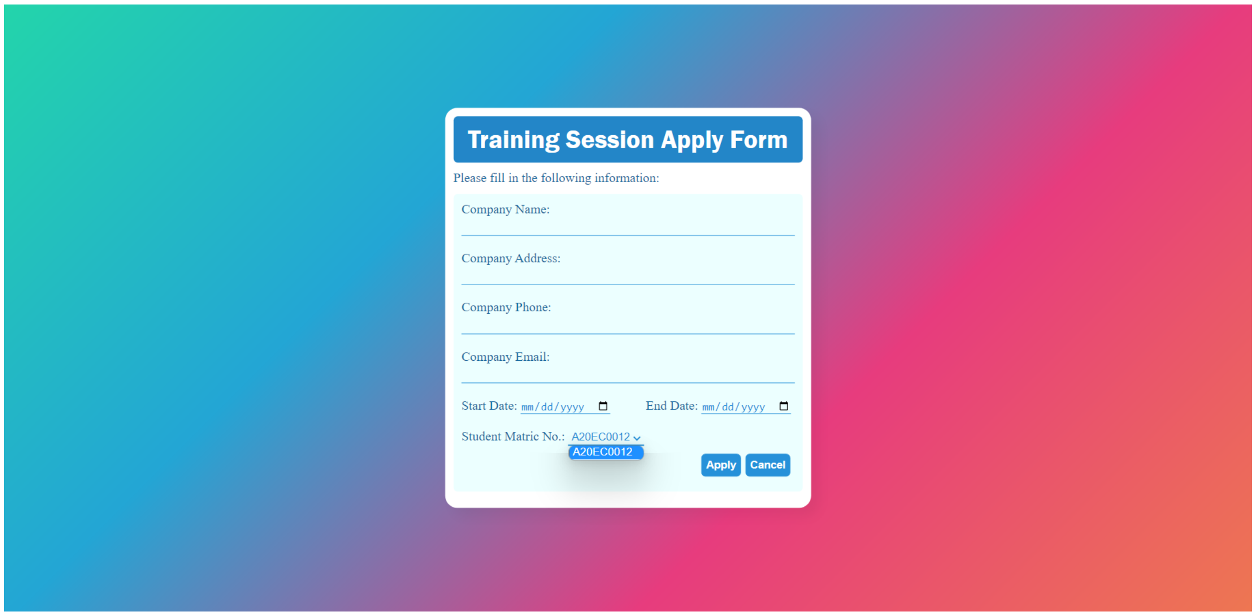
Company Email:

Start Date: mm/dd/yyyy End Date: mm/dd/yyyy

Student Matric No.: A20EC0012

A20EC0012
A20EC0111
A20EC0134

Figure 42: Selection of Student Matric Number when Applying Training Session (Admin User)


Training Session Apply Form

Please fill in the following information:

Company Name:

Company Address:

Company Phone:

Company Email:

Start Date: mm/dd/yyyy End Date: mm/dd/yyyy

Student Matric No.: A20EC0012

A20EC0012

Figure 43: Selection of Student Matric Number when Applying Training Session (Student User)