ASSIGNMENT

Advanced Ethical Hacking

Submitted by:D.Suresh Xavier

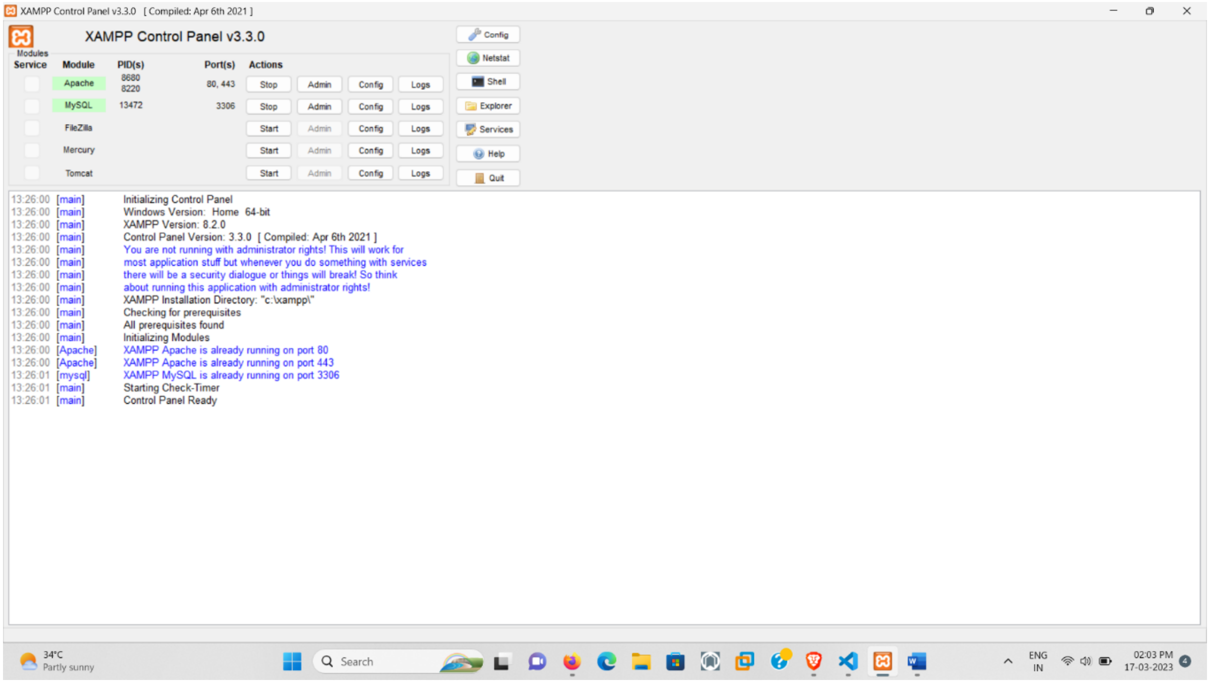
I-M.Sc Cyber Security

1)Create a Small Webpage using PHP:

a)Download and Installing Xampp :

First of all I Download Xampp Apache Distribution that contains MariaDB,PHP,and Perl and Install it in my Windows.

Then in the Xampp I started the Apache and MySQL



Service	Module	PID(s)	Port(s)	Actions			
	Apache	8680 8220	80, 443	Stop	Admin	Config	Logs
	MySQL	13472	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

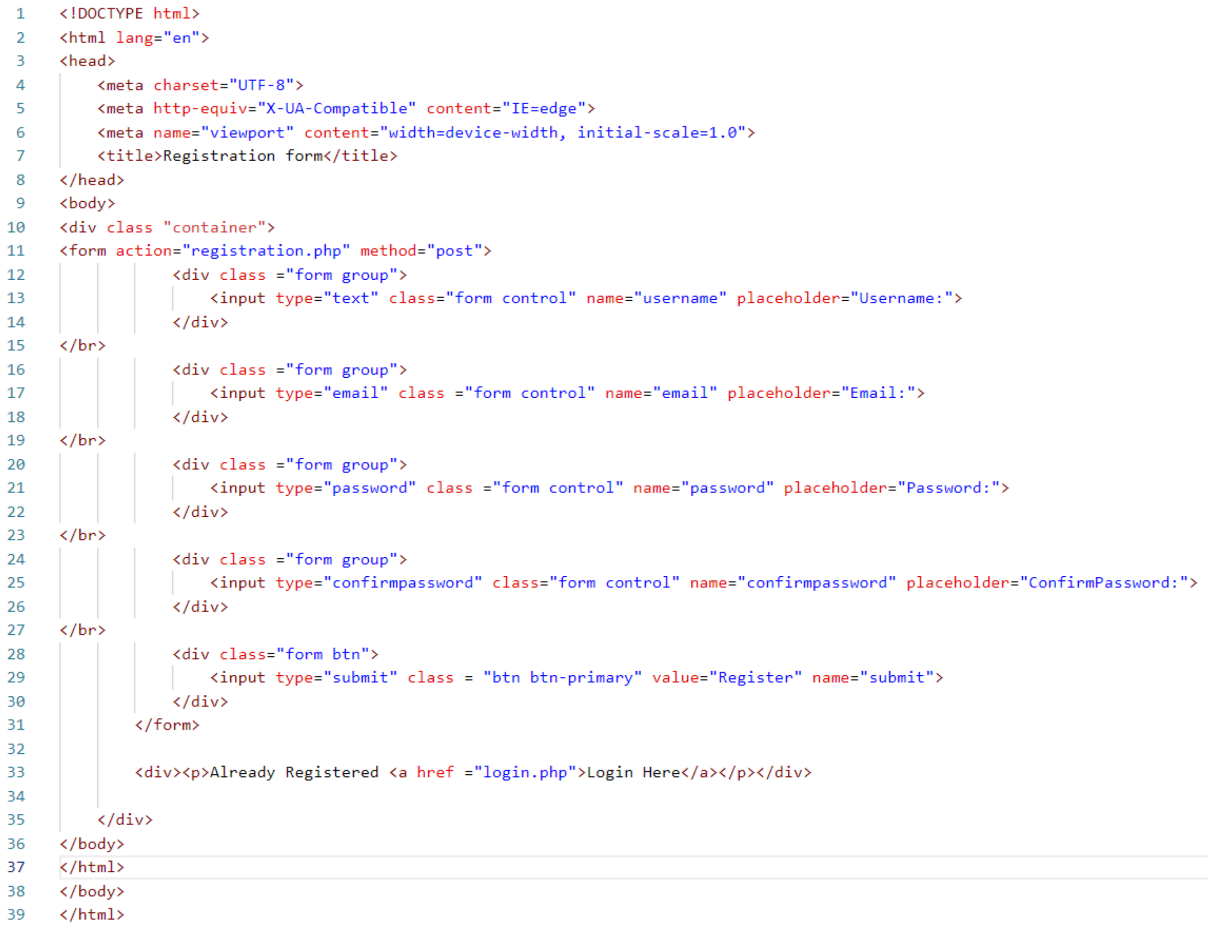
```
13:26:00  [main]     Initializing Control Panel
13:26:00  [main]     Windows Version:  Home  64-bit
13:26:00  [main]     XAMPP Version: 8.2.0
13:26:00  [main]     Control Panel Version: 3.3.0  [ Compiled: Apr 6th 2021 ]
13:26:00  [main]     You are not running with administrator rights! This will work for
13:26:00  [main]     most application stuff but whenever you do something with services
13:26:00  [main]     there will be a security dialogue or things will break! So think
13:26:00  [main]     about running this application with administrator rights!
13:26:00  [main]     XAMPP Installation Directory: "c:\xampp\"
13:26:00  [main]     Checking for prerequisites
13:26:00  [main]     All prerequisites found
13:26:00  [main]     Initializing Modules
13:26:00  [Apache]   XAMPP Apache is already running on port 80
13:26:00  [Apache]   XAMPP Apache is already running on port 443
13:26:01  [mysql]    XAMPP MySQL is already running on port 3306
13:26:01  [main]     Starting Check-Timer
13:26:01  [main]     Control Panel Ready
```

b)Registration Page:

Then I created the new folder namely Login -Register.Then created a new file named registration.php that contains the code for the registration page.

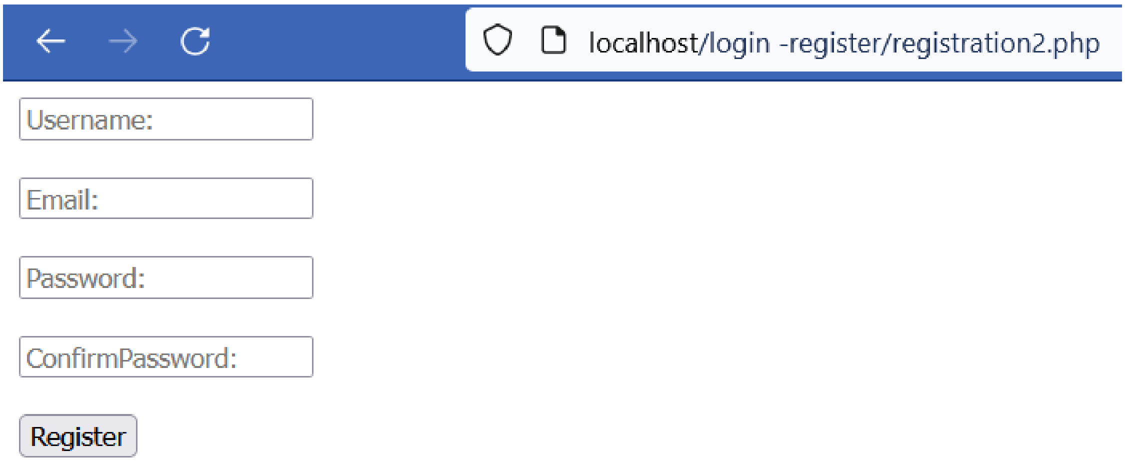
I used Visual studio Code for Coding.

This is the code for simple registration page :



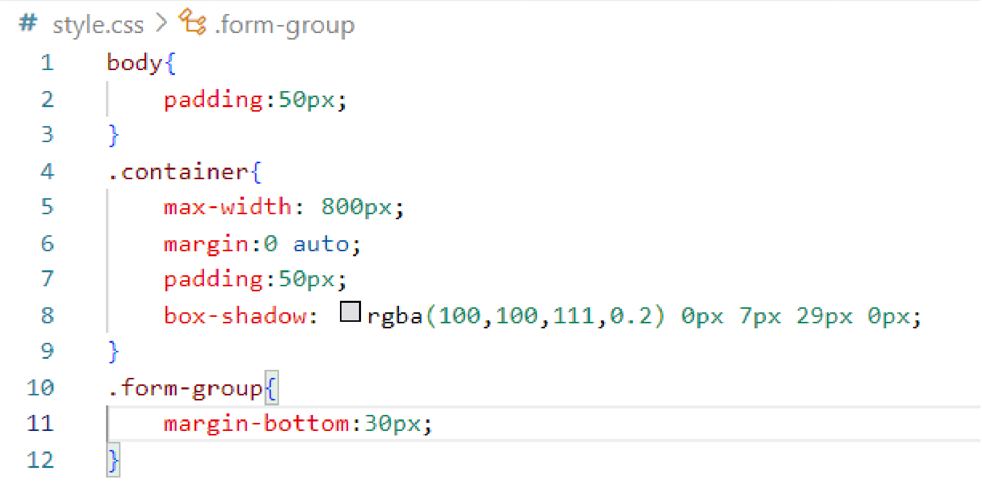
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration form</title>
</head>
<body>
<div class "container">
<form action="registration.php" method="post">
            <div class ="form group">
                <input type="text" class="form control" name="username" placeholder="Username:">
            </div>
</br>
            <div class ="form group">
                <input type="email" class ="form control" name="email" placeholder="Email:">
            </div>
</br>
            <div class ="form group">
                <input type="password" class ="form control" name="password" placeholder="Password:">
            </div>
</br>
            <div class ="form group">
                <input type="confirmpassword" class="form control" name="confirmpassword" placeholder="ConfirmPassword:">
            </div>
</br>
            <div class="form btn">
                <input type="submit" class = "btn btn-primary" value="Register" name="submit">
            </div>
        </form>

        <div><p>Already Registered <a href ="login.php">Login Here</a></p></div>

    </div>
</body>
</html>
</body>
</html>
```

Output:



Then I uses some **CSS** code to get better look of my Registration Page.

This is the **CSS** code that get better look for my registration page.

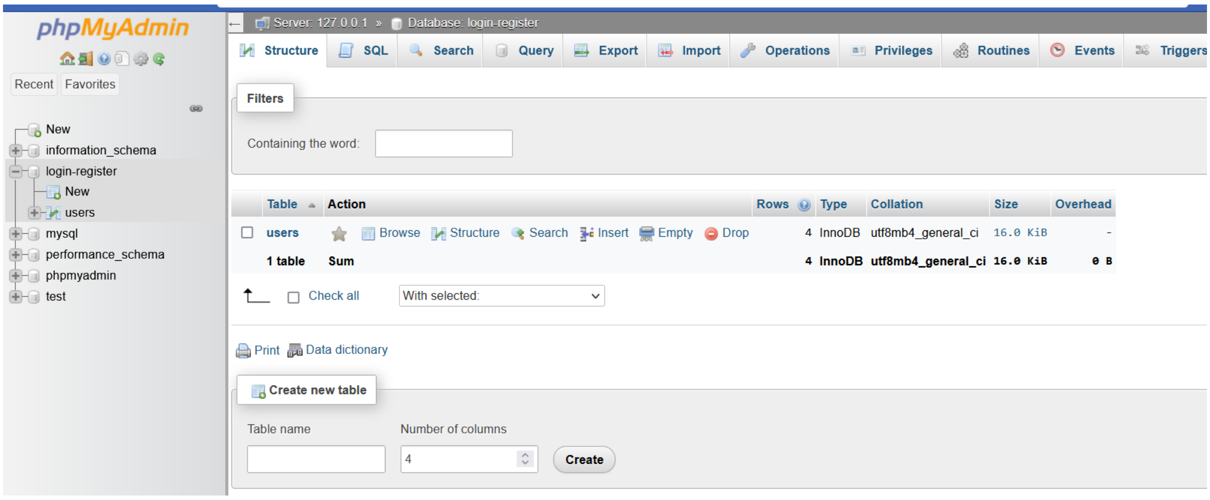


```
body{
    padding:50px;
}
.container{
    max-width: 800px;
    margin:0 auto;
    padding:50px;
    box-shadow: rgba(100,100,111,0.2) 0px 7px 29px 0px;
}
.form-group{
    margin-bottom:30px;
}
```

Then I created a database:

Database-A database is an organized collection of structured information, or data, typically stored electronically in a computer system**.**

Creating a database:



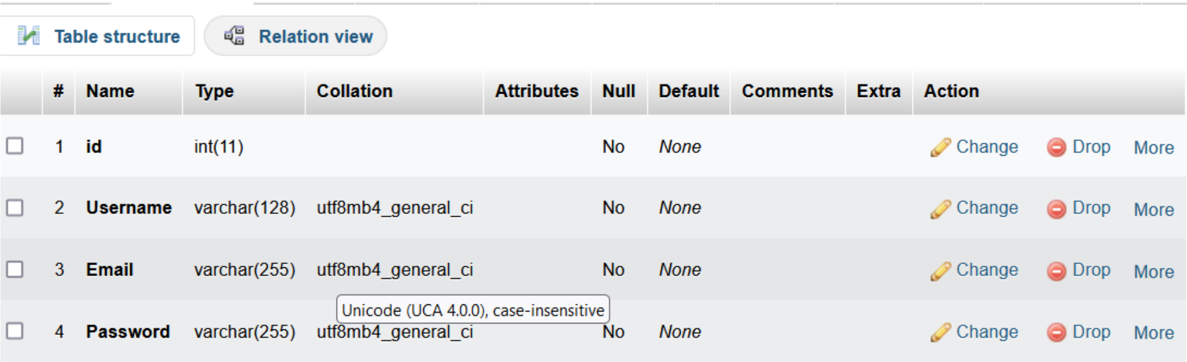
Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
□ users	☆ Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
1 table	Sum	4	InnoDB	utf8mb4_general_ci	16.0 KiB	0 B

Table name:

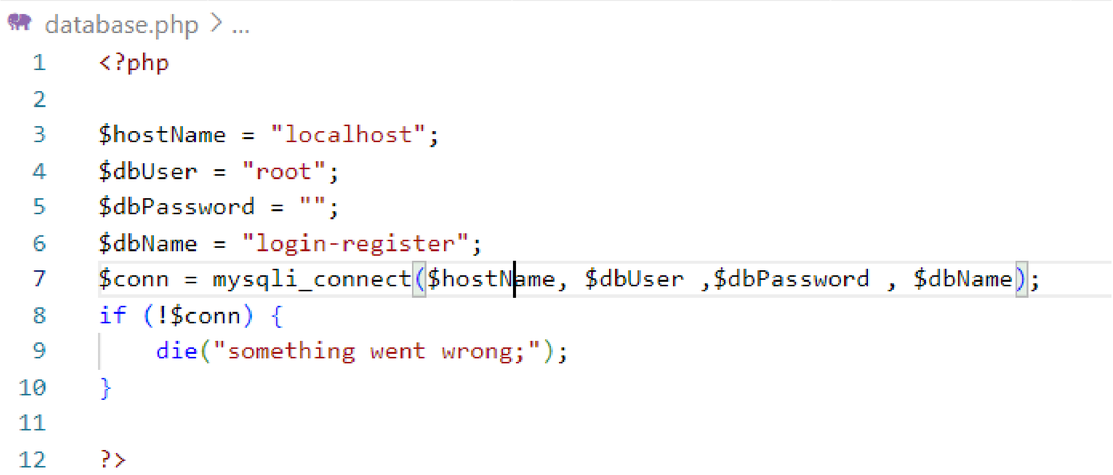
Number of columns: 4

Table Structure :



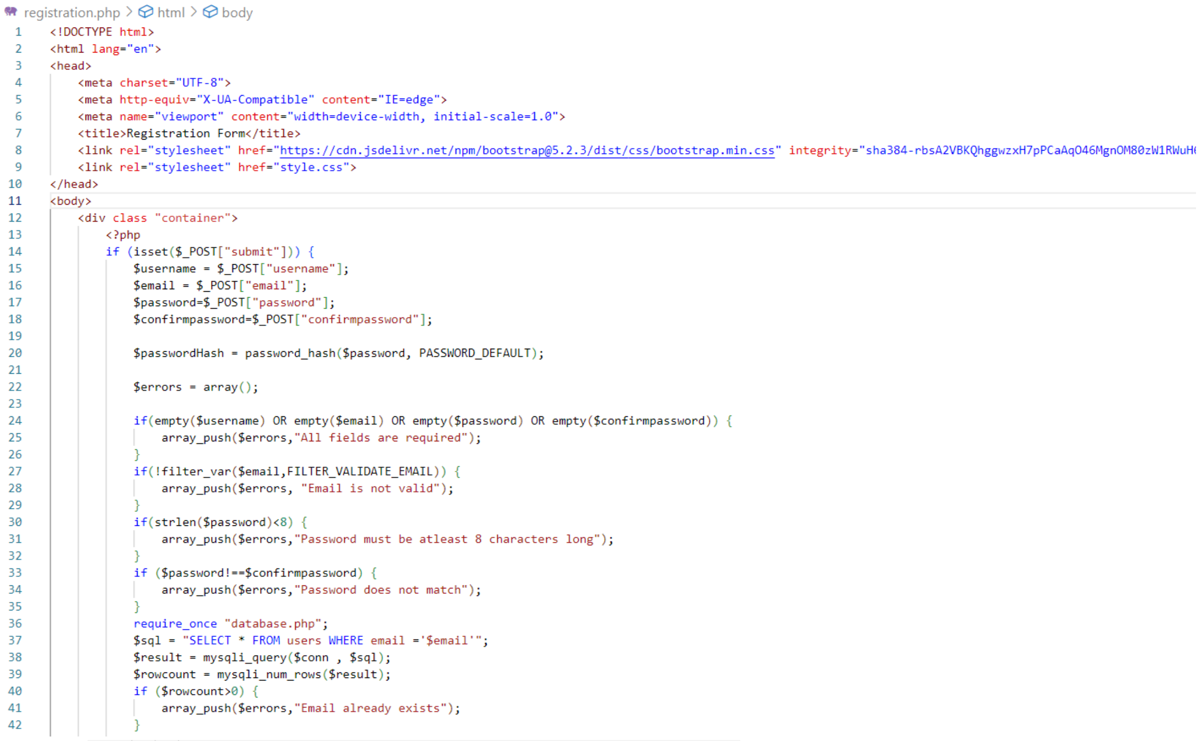
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	**id**	int(11)			No	*None*			Change Drop More
□	2	**Username**	varchar(128)	utf8mb4_general_ci		No	*None*			Change Drop More
□	3	**Email**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More
□	4	**Password**	varchar(255)	utf8mb4_general_ci		No	*None*			Change Drop More

This is the code for **database.php:**



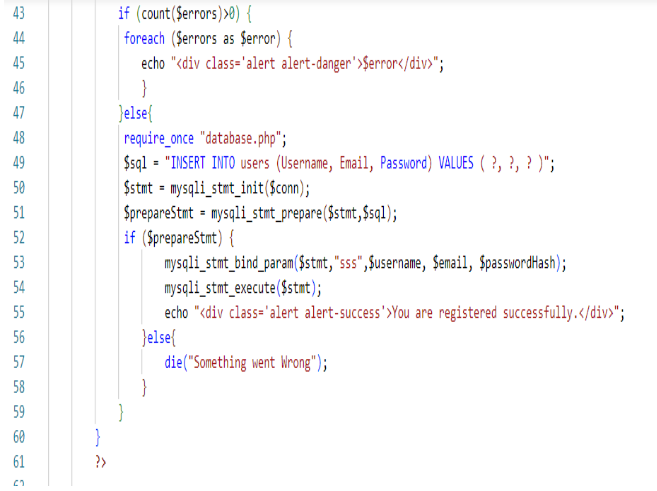
```php
<?php

$hostName = "localhost";
$dbUser = "root";
$dbPassword = "";
$dbName = "login-register";
$conn = mysqli_connect($hostName, $dbUser ,$dbPassword , $dbName);
if (!$conn) {
    die("something went wrong;");
}

?>
```

This is the final code for the registration form:

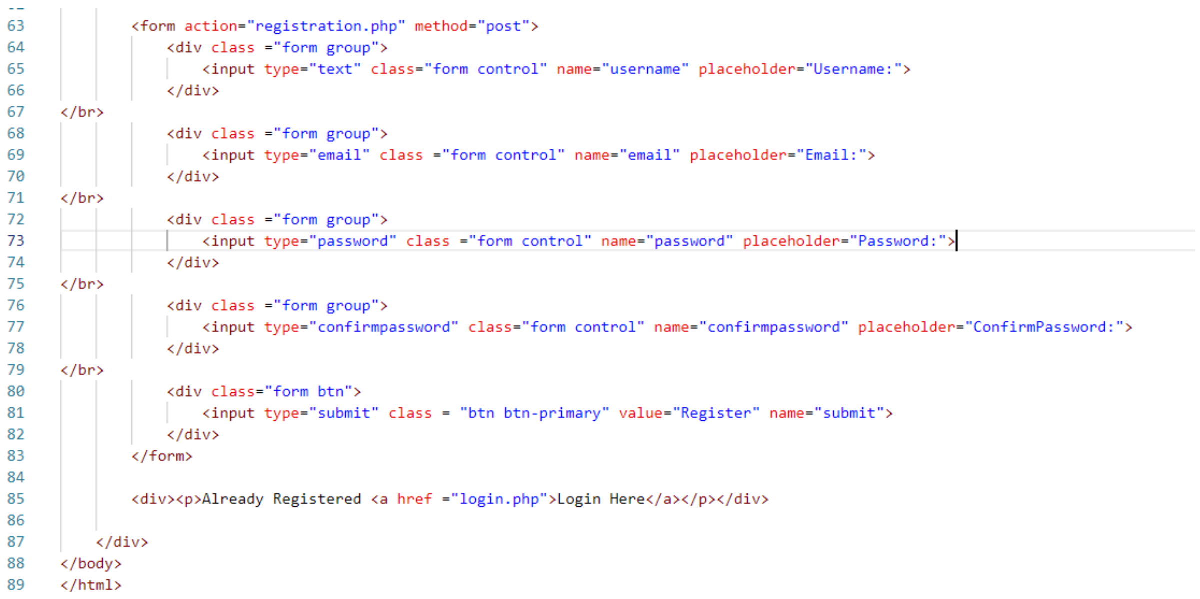


registration.php > html > body

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Registration Form</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class "container">
        <?php
        if (isset($_POST["submit"])) {
            $username = $_POST["username"];
            $email = $_POST["email"];
            $password=$_POST["password"];
            $confirmpassword=$_POST["confirmpassword"];

            $passwordHash = password_hash($password, PASSWORD_DEFAULT);

            $errors = array();

            if(empty($username) OR empty($email) OR empty($password) OR empty($confirmpassword)) {
                array_push($errors,"All fields are required");
            }
            if(!filter_var($email,FILTER_VALIDATE_EMAIL)) {
                array_push($errors, "Email is not valid");
            }
            if(strlen($password)<8) {
                array_push($errors,"Password must be atleast 8 characters long");
            }
            if ($password!==$confirmpassword) {
                array_push($errors,"Password does not match");
            }
            require_once "database.php";
            $sql = "SELECT * FROM users WHERE email ='$email'";
            $result = mysqli_query($conn , $sql);
            $rowcount = mysqli_num_rows($result);
            if ($rowcount>0) {
                array_push($errors,"Email already exists");
            }
```

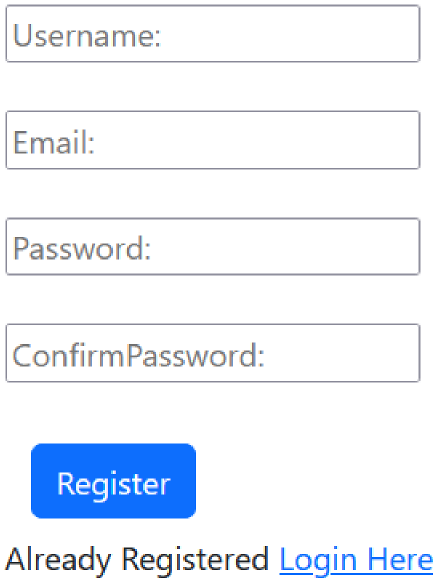


```
            if (count($errors)>0) {
             foreach ($errors as $error) {
                echo "<div class='alert alert-danger'>$error</div>";
                }
            }else{
             require_once "database.php";
             $sql = "INSERT INTO users (Username, Email, Password) VALUES ( ?, ?, ? )";
             $stmt = mysqli_stmt_init($conn);
             $prepareStmt = mysqli_stmt_prepare($stmt,$sql);
             if ($prepareStmt) {
                    mysqli_stmt_bind_param($stmt,"sss",$username, $email, $passwordHash);
                    mysqli_stmt_execute($stmt);
                    echo "<div class='alert alert-success'>You are registered successfully.</div>";
                }else{
                    die("Something went Wrong");
                }
            }
        }
        ?>
```



```
        <form action="registration.php" method="post">
            <div class ="form group">
                <input type="text" class="form control" name="username" placeholder="Username:">
            </div>
</br>
            <div class ="form group">
                <input type="email" class ="form control" name="email" placeholder="Email:">
            </div>
</br>
            <div class ="form group">
                <input type="password" class ="form control" name="password" placeholder="Password:">
            </div>
</br>
            <div class ="form group">
                <input type="confirmpassword" class="form control" name="confirmpassword" placeholder="ConfirmPassword:">
            </div>
</br>
            <div class="form btn">
                <input type="submit" class = "btn btn-primary" value="Register" name="submit">
            </div>
        </form>

        <div><p>Already Registered <a href ="login.php">Login Here</a></p></div>

    </div>
</body>
</html>
```

Output:



Username:

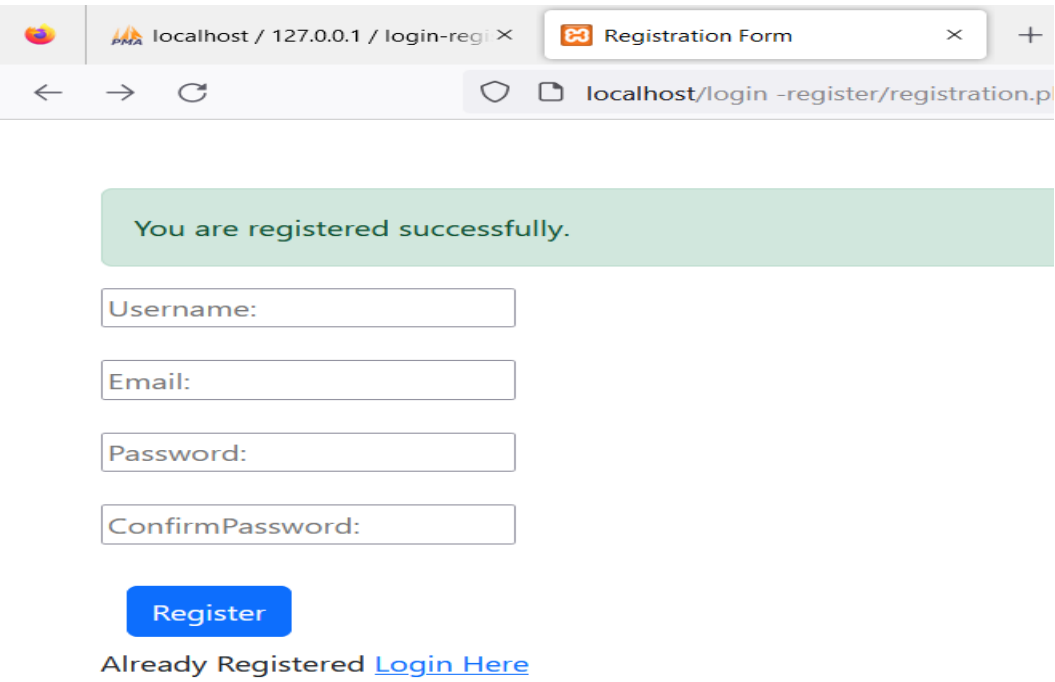
Email:

Password:

ConfirmPassword:

Already Registered Login Here

After the Successful registration it shows the output like this:

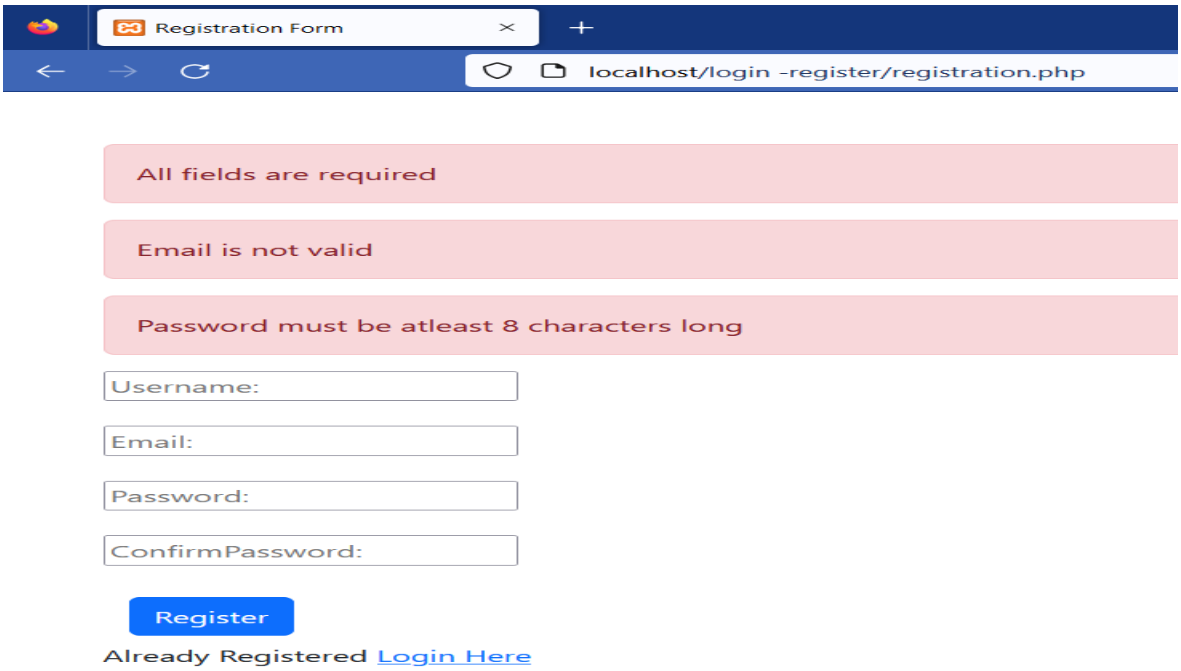

localhost / 127.0.0.1 / login-regi
Registration Form
localhost/login -register/registration.pl

You are registered successfully.

Username:
Email:
Password:
ConfirmPassword:
Register

Already Registered Login Here

- There error will be if we did not fill out the form and click register like this----:



All fields are required

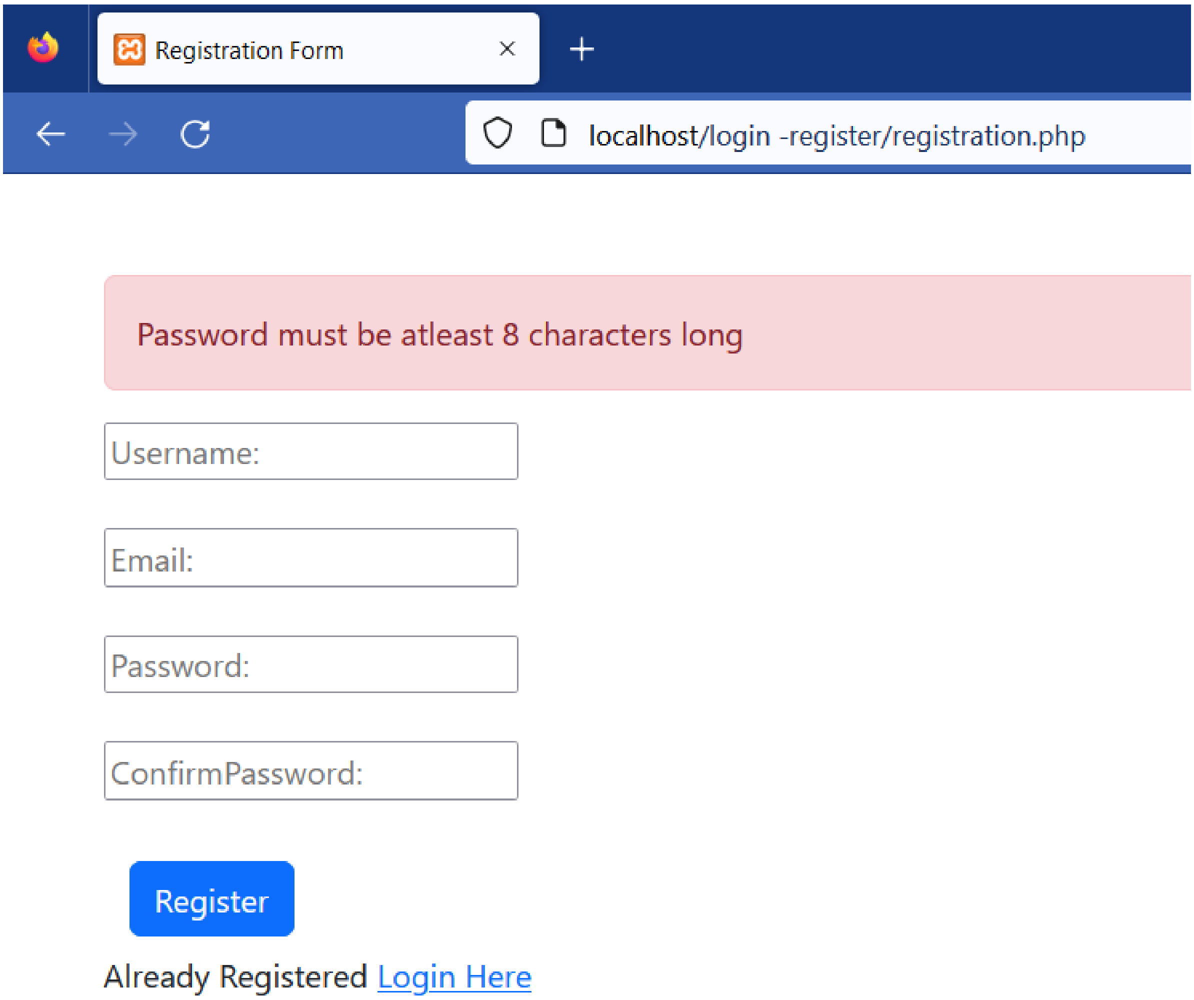
Email is not valid

Password must be atleast 8 characters long

Already Registered Login Here

- In this Registration page the Username, Emails and Passwords that are registered are stored in the Database.

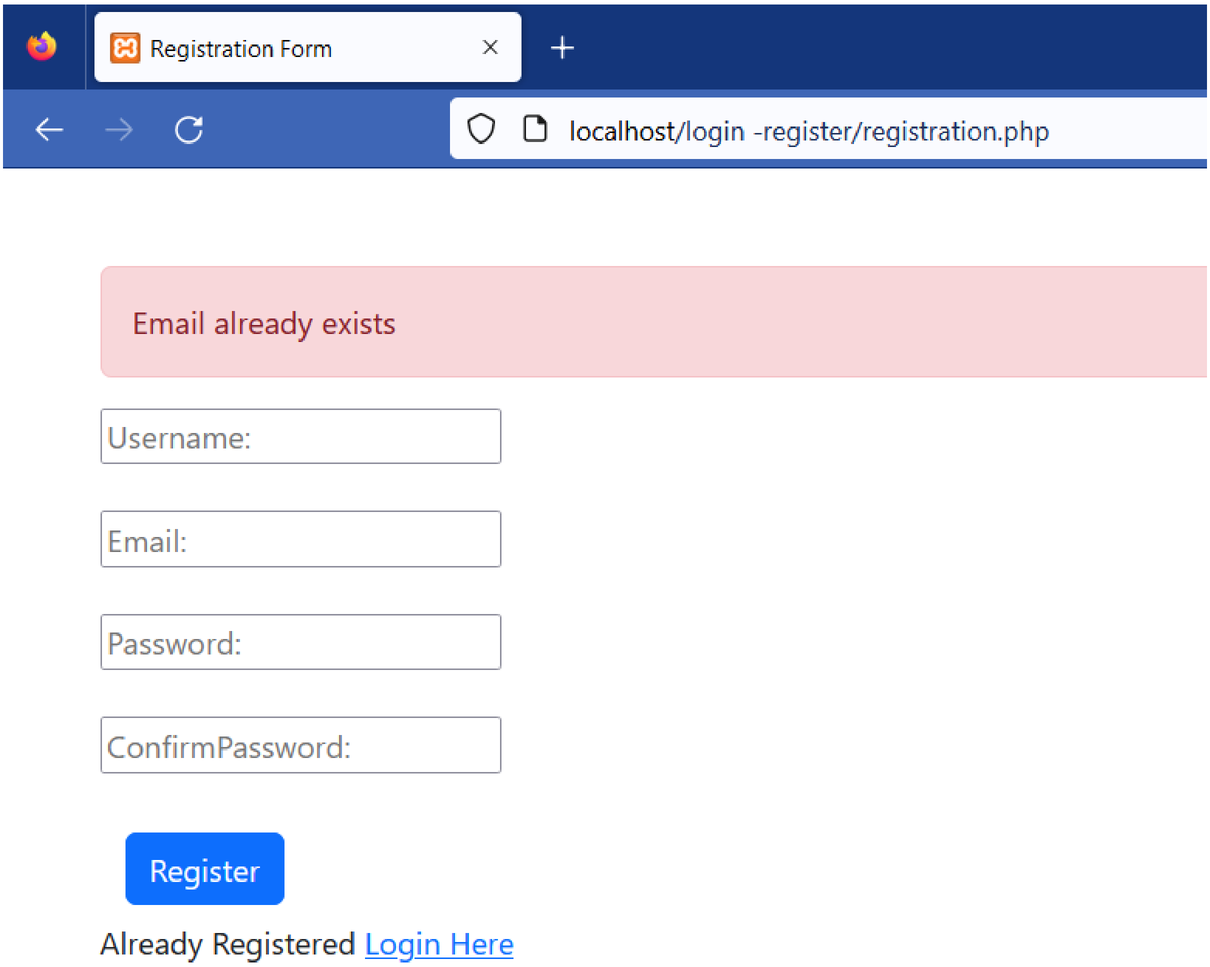
- The code tells that the password must be atleast 8 characters else it will show error like this ---:


Registration Form
localhost/login -register/registration.php

Password must be atleast 8 characters long

Username:
Email:
Password:
ConfirmPassword:
Register
Already Registered Login Here

- And if the Email are already registered and stored in the database it will show error like this---:



Email already exists

Username:

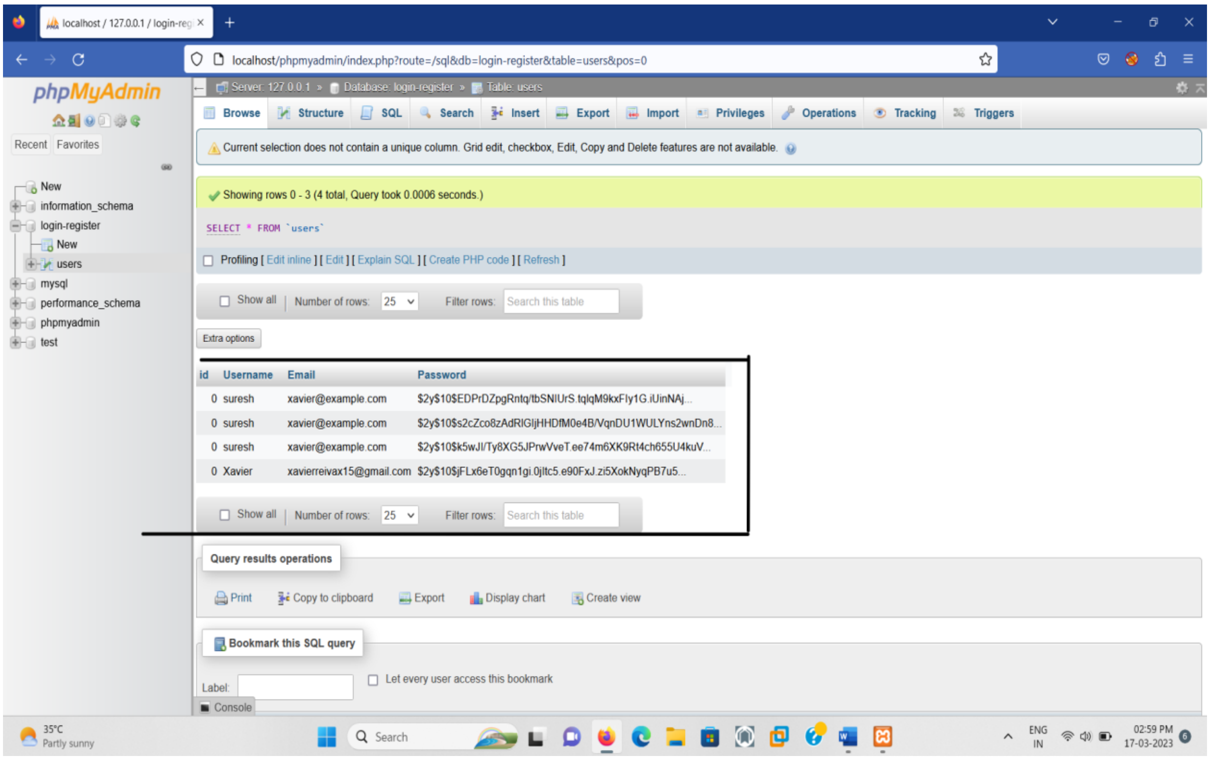
Email:

Password:

ConfirmPassword:

Already Registered Login Here

This are the datas that are stored in the database after the registration:



localhost / 127.0.0.1 / login-reg
localhost/phpmyadmin/index.php?route=/sql&db=login-register&table=users&pos=0
phpMyAdmin
Recent Favorites
New
information_schema
login-register
New
users
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: login-register » Table: users
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 3 (4 total, Query took 0.0006 seconds.)

```
SELECT * FROM `users`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

□ Show all | Number of rows: 25 Filter rows: Search this table

Extra options

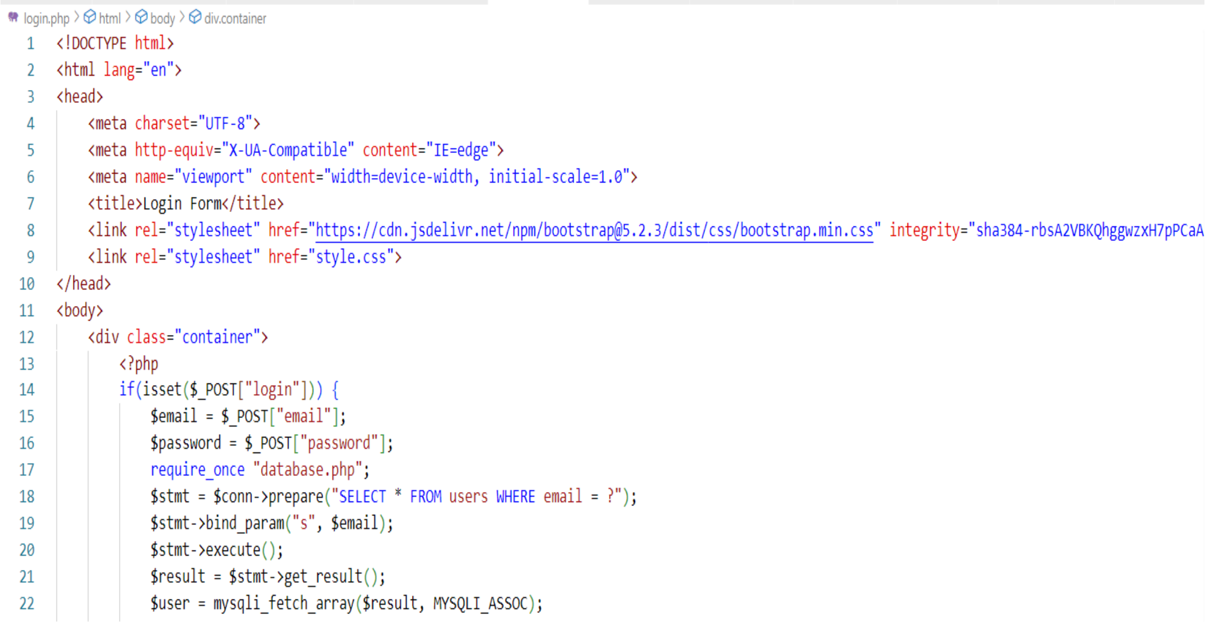
id	Username	Email	Password
0	suresh	xavier@example.com	$2y$10$EDPrDZpgRntq/tbSNIUrS.tqlqM9kxFly1G.iUinNAj...
0	suresh	xavier@example.com	$2y$10$s2cZco8zAdRIGljHHDfM0e4B/VqnDU1WULYns2wnDn8...
0	suresh	xavier@example.com	$2y$10$k5wJI/Ty8XG5JPrwVveT.ee74m6XK9Rt4ch655U4kuV...
0	Xavier	xavierreivax15@gmail.com	$2y$10$jFLx6eT0gqn1gi.0jltc5.e90FxJ.zi5XokNyqPB7u5...

□ Show all | Number of rows: 25 Filter rows: Search this table

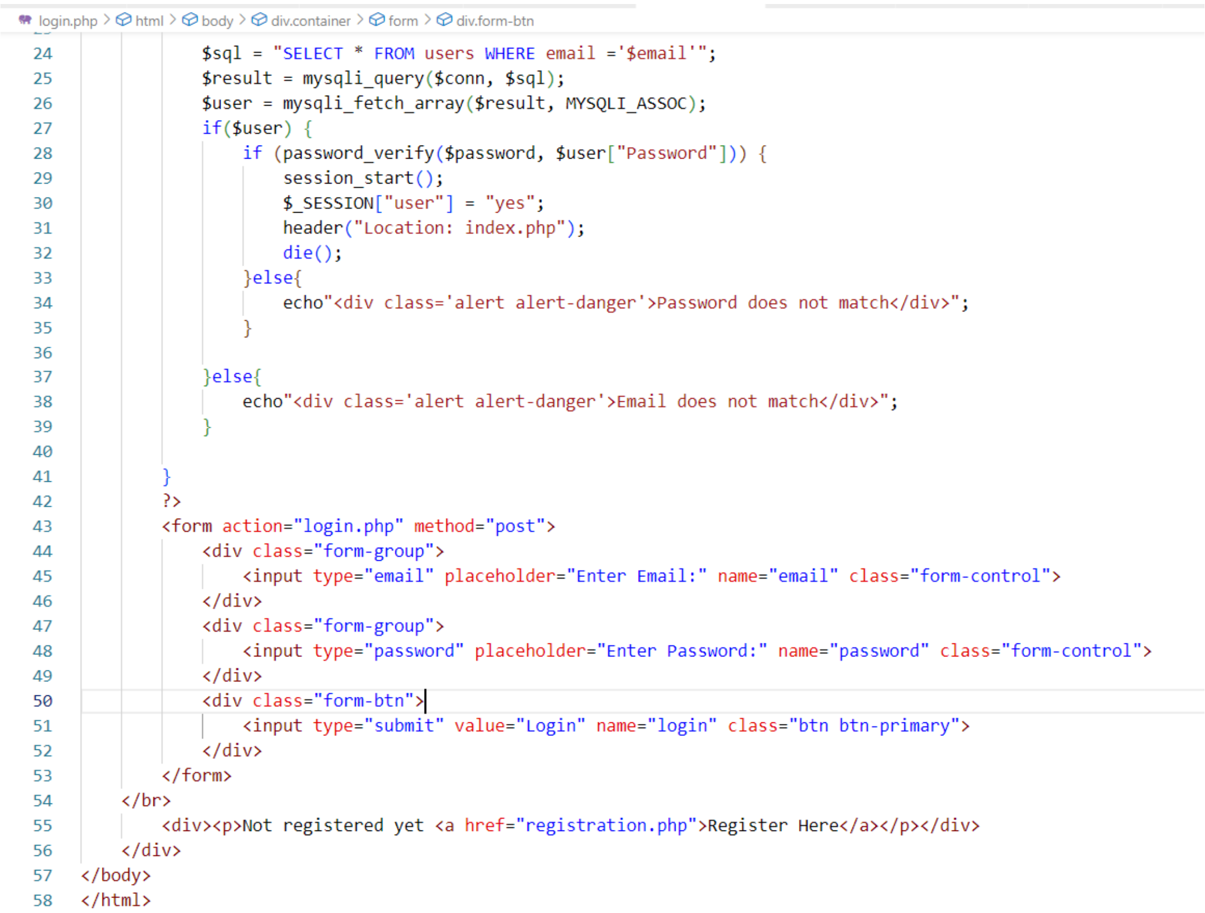
c)Login Page:

Then I created a login page using PHP.

The Code for login page is :

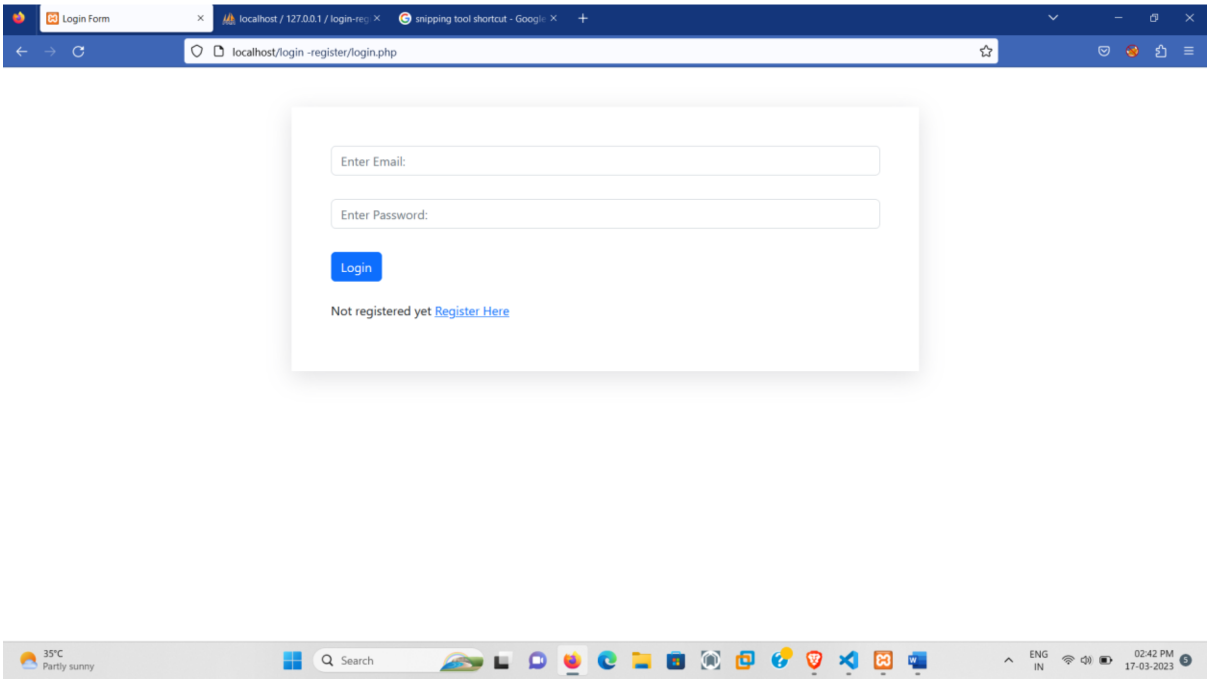


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" integrity="sha384-rbsA2VBKQhggwzxH7pPCaA
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <?php
        if(isset($_POST["login"])) {
            $email = $_POST["email"];
            $password = $_POST["password"];
            require_once "database.php";
            $stmt = $conn->prepare("SELECT * FROM users WHERE email = ?");
            $stmt->bind_param("s", $email);
            $stmt->execute();
            $result = $stmt->get_result();
            $user = mysqli_fetch_array($result, MYSQLI_ASSOC);
```


login.php > html > body > div.container > form > div.form-btn

```
            $sql = "SELECT * FROM users WHERE email ='$email'";
            $result = mysqli_query($conn, $sql);
            $user = mysqli_fetch_array($result, MYSQLI_ASSOC);
            if($user) {
                if (password_verify($password, $user["Password"])) {
                    session_start();
                    $_SESSION["user"] = "yes";
                    header("Location: index.php");
                    die();
                }else{
                    echo"<div class='alert alert-danger'>Password does not match</div>";
                }

            }else{
                echo"<div class='alert alert-danger'>Email does not match</div>";
            }

        }
        ?>
        <form action="login.php" method="post">
            <div class="form-group">
                <input type="email" placeholder="Enter Email:" name="email" class="form-control">
            </div>
            <div class="form-group">
                <input type="password" placeholder="Enter Password:" name="password" class="form-control">
            </div>
            <div class="form-btn">
                <input type="submit" value="Login" name="login" class="btn btn-primary">
            </div>
        </form>
    </br>
        <div><p>Not registered yet <a href="registration.php">Register Here</a></p></div>
    </div>
</body>
</html>
```

Output:


Login Form
localhost / 127.0.0.1 / login-reg
snipping tool shortcut - Google
localhost/login -register/login.php
35°C
Partly sunny
Search
ENG
IN
02:42 PM
17-03-2023

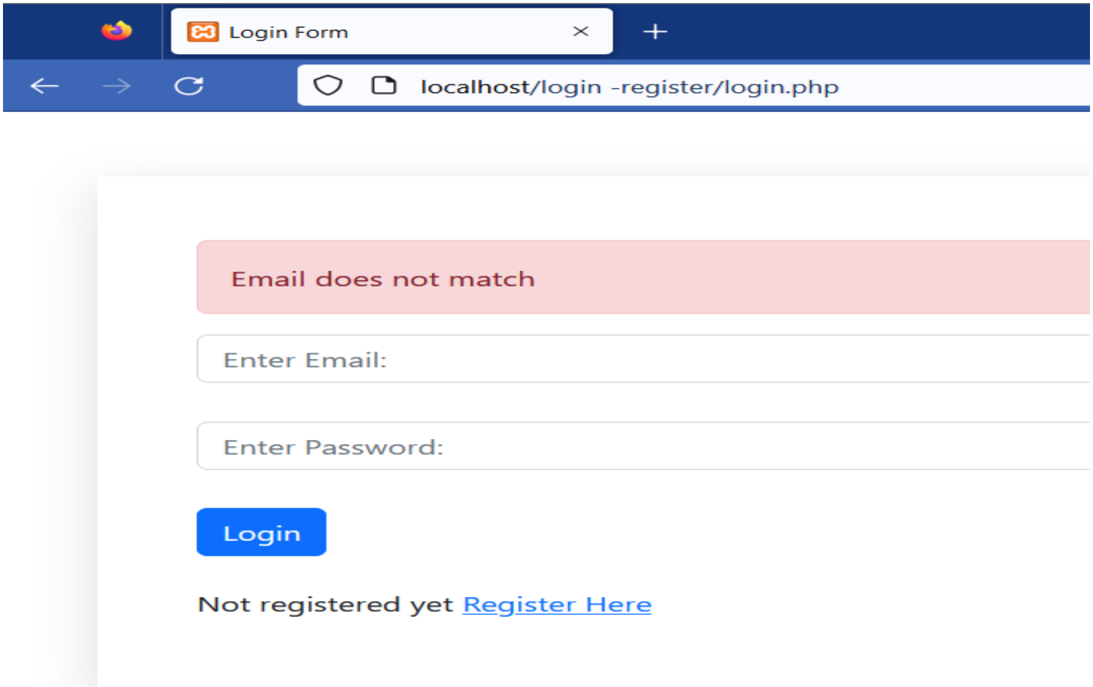
Enter Email:

Enter Password:

Login

Not registered yet Register Here

The error page will be shown if the entered email does not register to log in the webpage---:

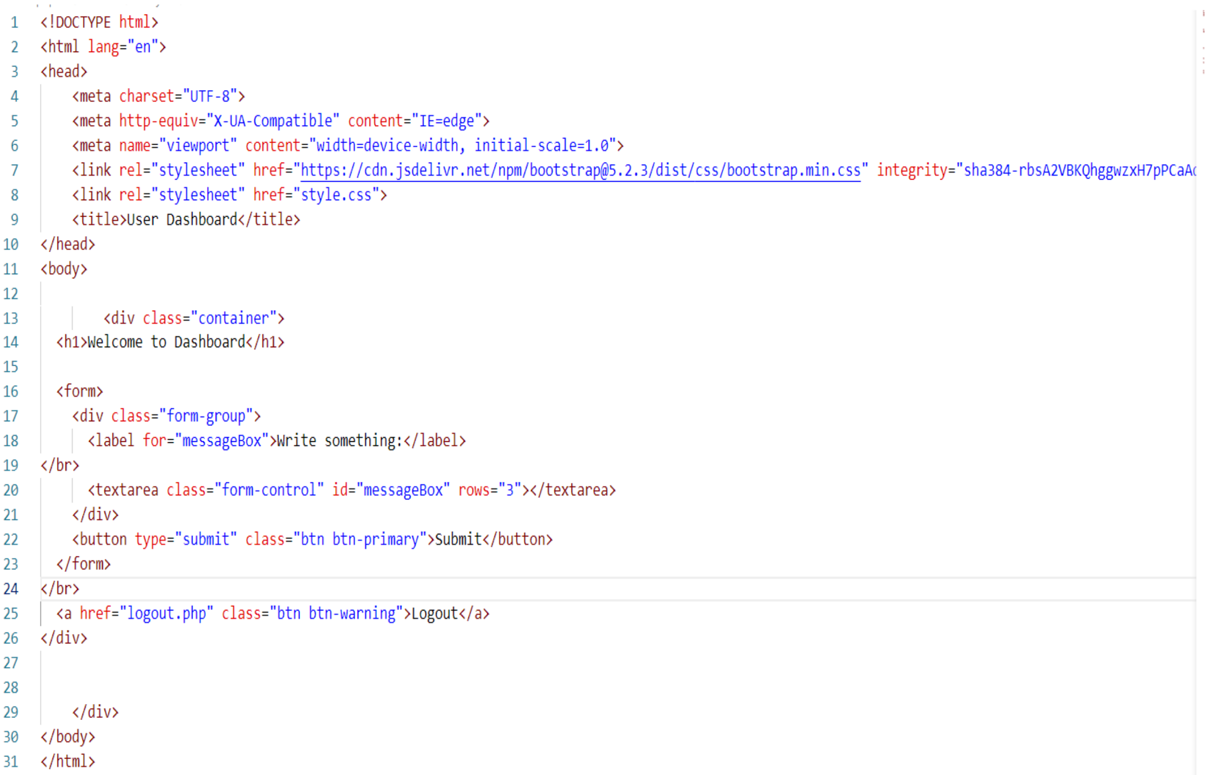

Login Form
localhost/login -register/login.php

Email does not match

Enter Email:
Enter Password:
Login
Not registered yet Register Here

d)Index Page:

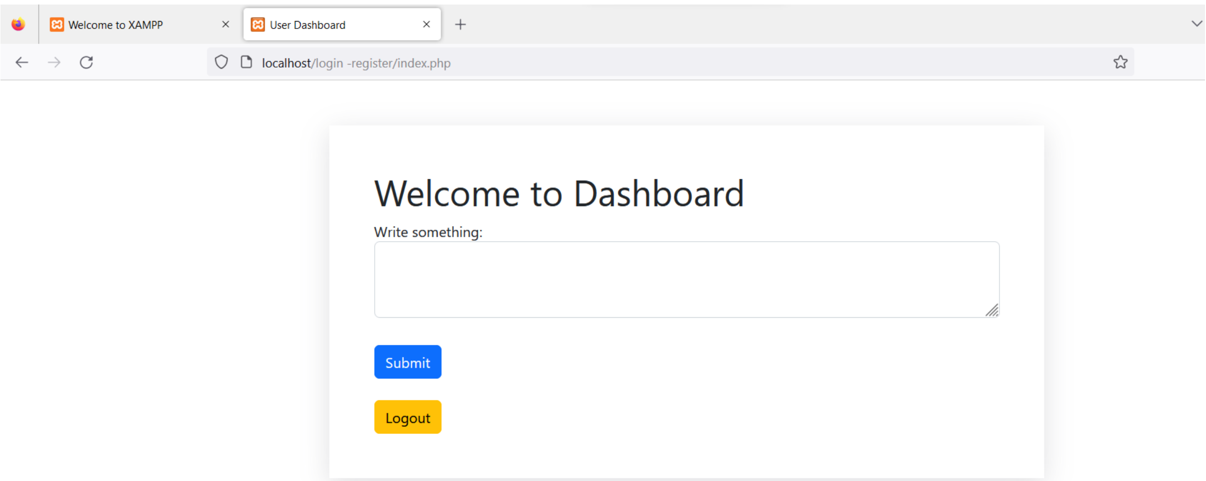
The Code For Index page is:



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" integrity="sha384-rbsA2VBKQhggwzxH7pPCaA
    <link rel="stylesheet" href="style.css">
    <title>User Dashboard</title>
</head>
<body>

        <div class="container">
  <h1>Welcome to Dashboard</h1>

  <form>
    <div class="form-group">
      <label for="messageBox">Write something:</label>
</br>
      <textarea class="form-control" id="messageBox" rows="3"></textarea>
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
  </form>
</br>
  <a href="logout.php" class="btn btn-warning">Logout</a>
</div>


    </div>
</body>
</html>
```

Output:

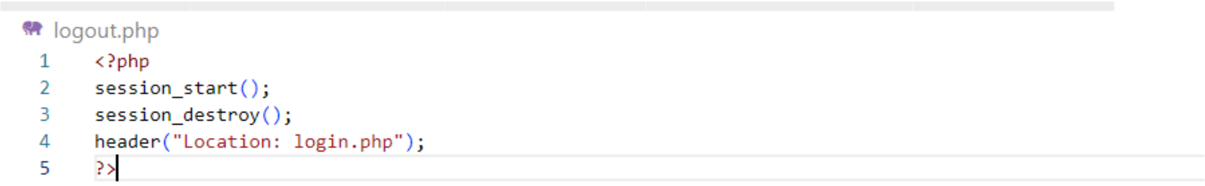
It directing to the dashboard page:



Logout Page:

- When we click the logout button in the dashboard page it will redirect to the login page.

- The code for the logout page is:

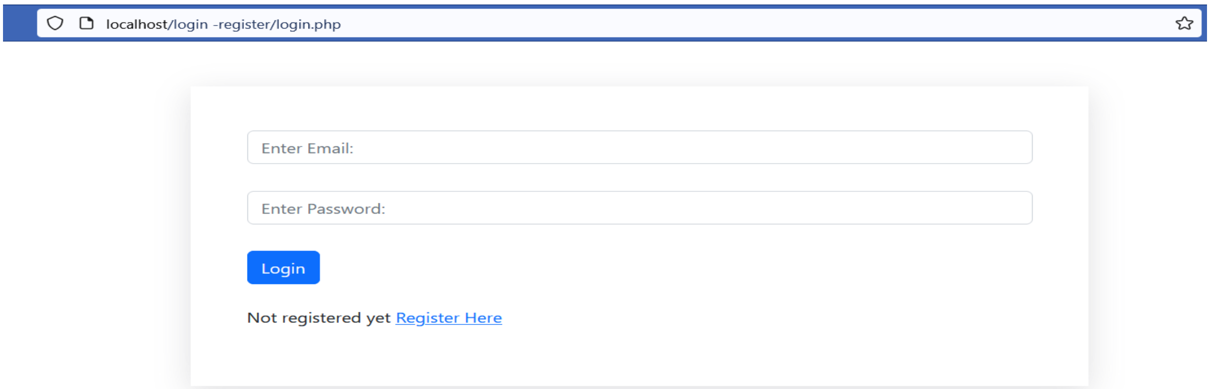


logout.php

```
<?php
session_start();
session_destroy();
header("Location: login.php");
?>
```

Output:

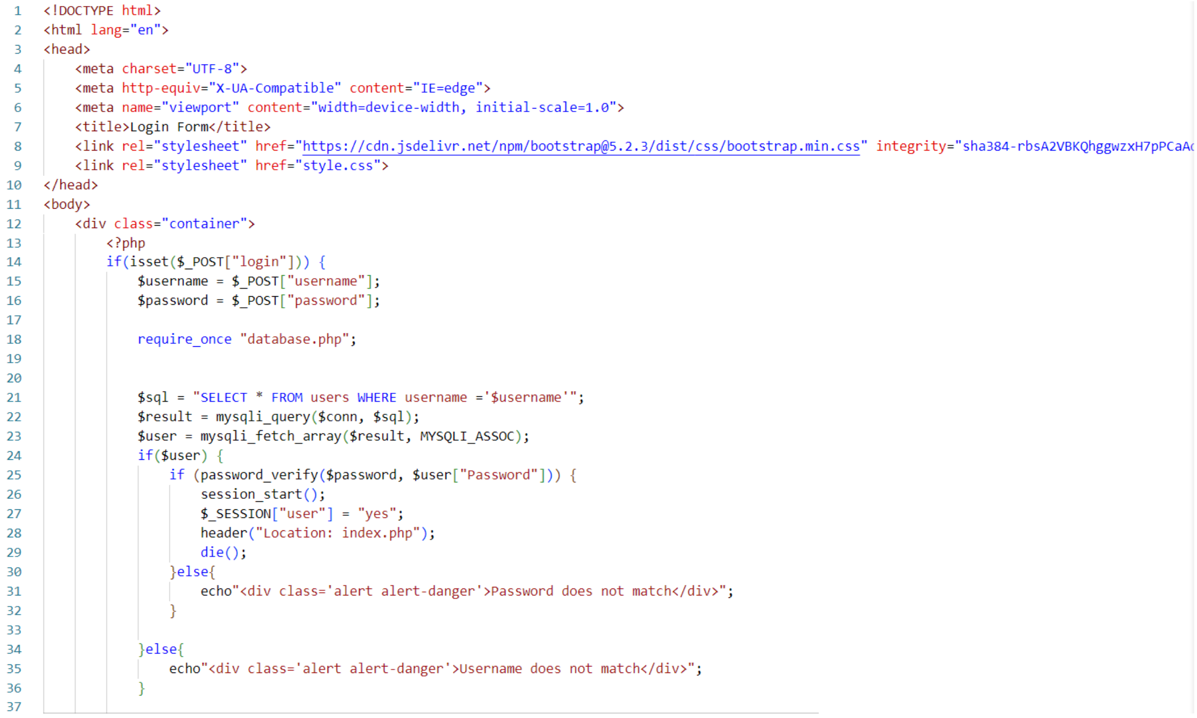
It will be redirect to the Login page :


localhost/login -register/login.php
Enter Email:
Enter Password:
Login
Not registered yet Register Here

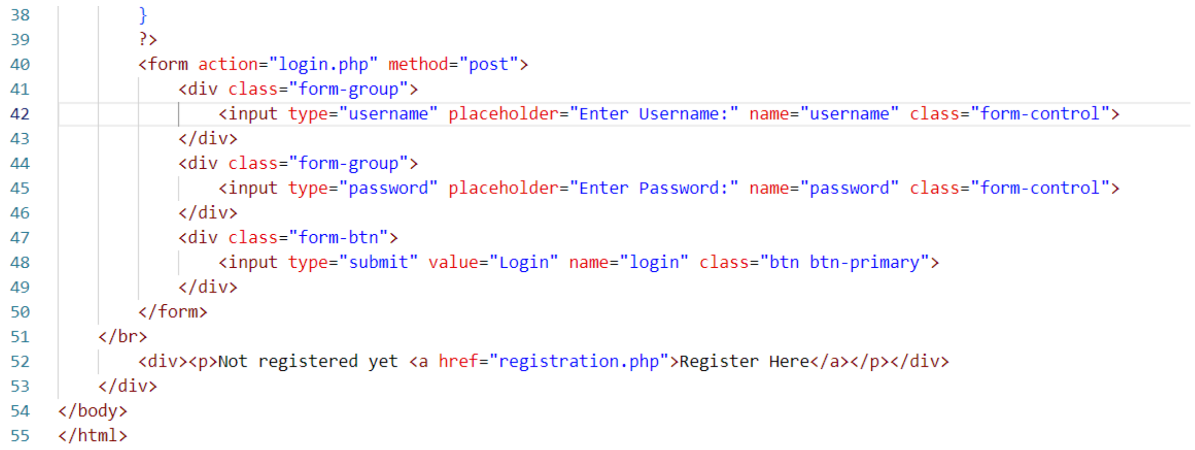
c)SQL Injection:

SQL injection is a code injection technique that might destroy your database. SQL injection is one of the most common web hacking techniques. SQL injection is the placement of malicious code in SQL statements, via web page input.

The code before sql injection--;

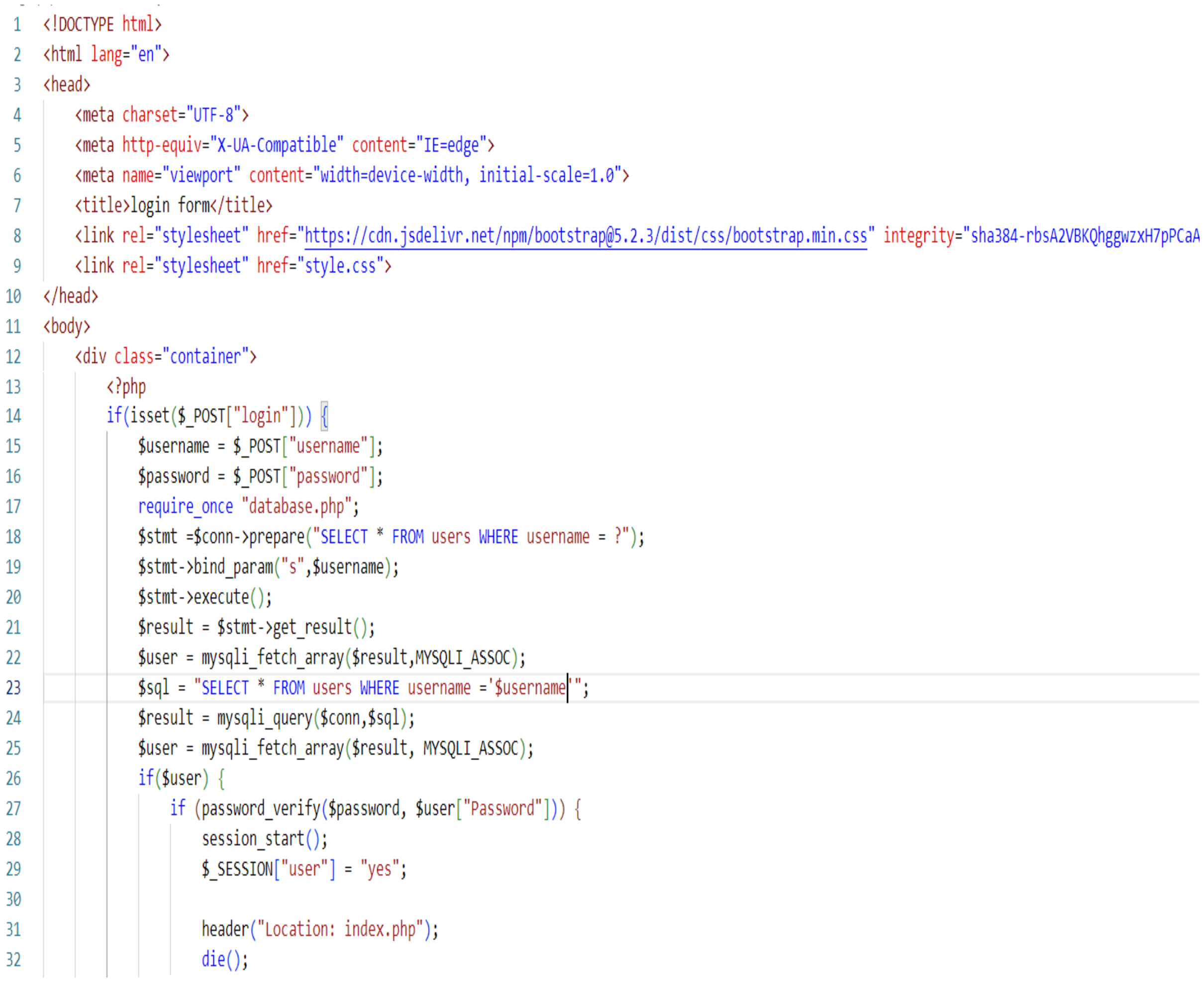


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login Form</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAc
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <?php
        if(isset($_POST["login"])) {
            $username = $_POST["username"];
            $password = $_POST["password"];

            require_once "database.php";


            $sql = "SELECT * FROM users WHERE username ='$username'";
            $result = mysqli_query($conn, $sql);
            $user = mysqli_fetch_array($result, MYSQLI_ASSOC);
            if($user) {
                if (password_verify($password, $user["Password"])) {
                    session_start();
                    $_SESSION["user"] = "yes";
                    header("Location: index.php");
                    die();
                }else{
                    echo"<div class='alert alert-danger'>Password does not match</div>";
                }

            }else{
                echo"<div class='alert alert-danger'>Username does not match</div>";
            }

```

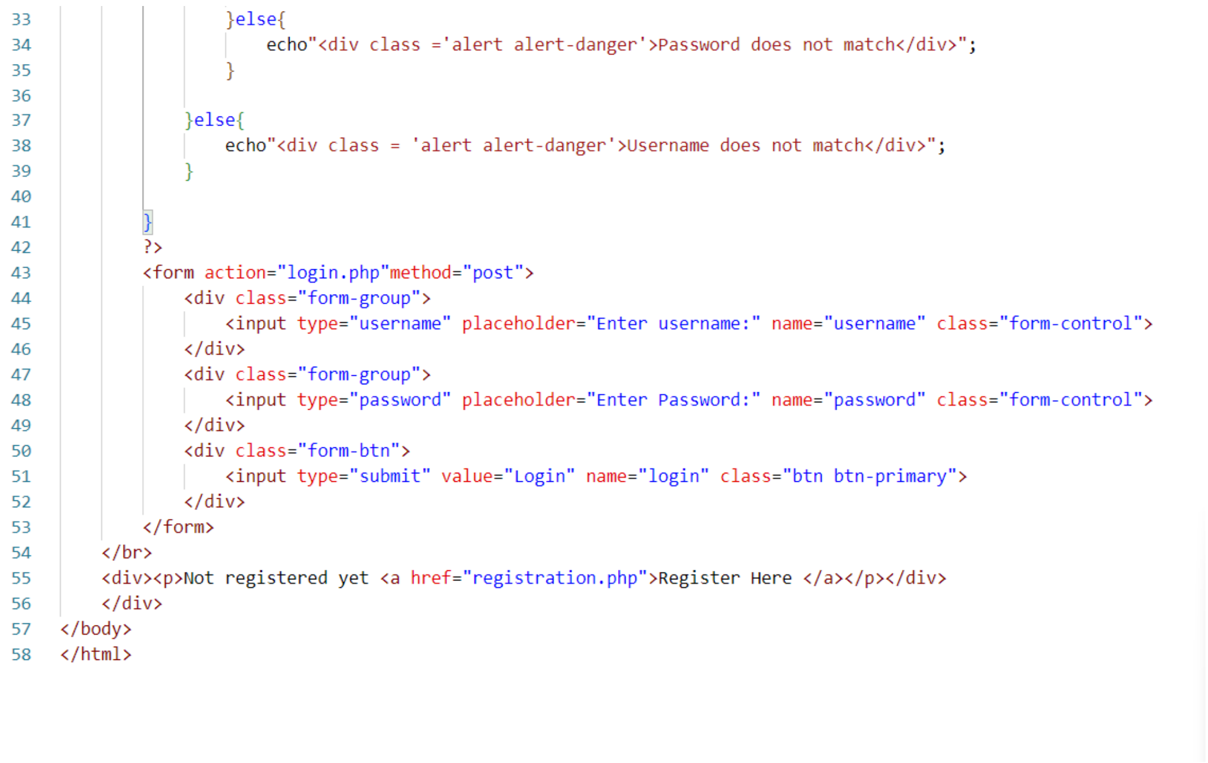


```
        }
        ?>
        <form action="login.php" method="post">
            <div class="form-group">
                <input type="username" placeholder="Enter Username:" name="username" class="form-control">
            </div>
            <div class="form-group">
                <input type="password" placeholder="Enter Password:" name="password" class="form-control">
            </div>
            <div class="form-btn">
                <input type="submit" value="Login" name="login" class="btn btn-primary">
            </div>
        </form>
    </br>
        <div><p>Not registered yet <a href="registration.php">Register Here</a></p></div>
    </div>
</body>
</html>
```

The code after SQL injection---;

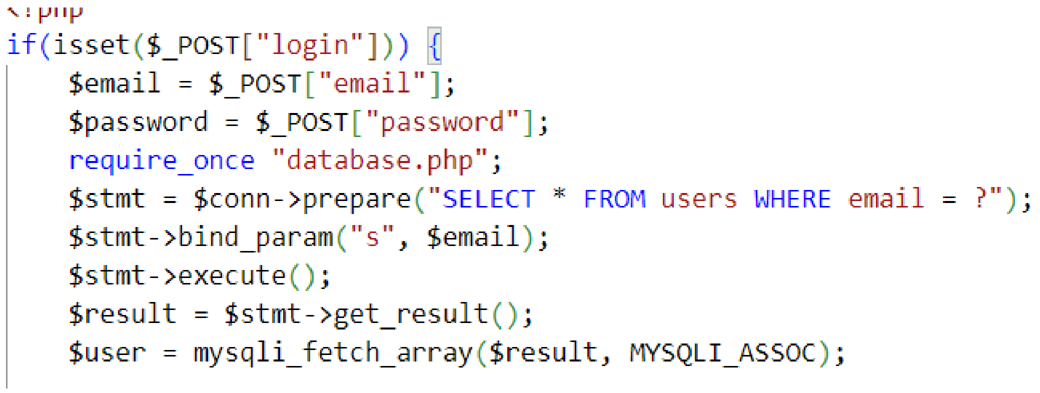


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>login form</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" integrity="sha384-rbsA2VBKQhggwzxH7pPCaA
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <?php
        if(isset($_POST["login"])) {
            $username = $_POST["username"];
            $password = $_POST["password"];
            require_once "database.php";
            $stmt =$conn->prepare("SELECT * FROM users WHERE username = ?");
            $stmt->bind_param("s",$username);
            $stmt->execute();
            $result = $stmt->get_result();
            $user = mysqli_fetch_array($result,MYSQLI_ASSOC);
            $sql = "SELECT * FROM users WHERE username ='$username'";
            $result = mysqli_query($conn,$sql);
            $user = mysqli_fetch_array($result, MYSQLI_ASSOC);
            if($user) {
                if (password_verify($password, $user["Password"])) {
                    session_start();
                    $_SESSION["user"] = "yes";

                    header("Location: index.php");
                    die();
```



```
                }else{
                    echo"<div class ='alert alert-danger'>Password does not match</div>";
                }

            }else{
                echo"<div class = 'alert alert-danger'>Username does not match</div>";
            }

        }
        ?>
        <form action="login.php"method="post">
            <div class="form-group">
                <input type="username" placeholder="Enter username:" name="username" class="form-control">
            </div>
            <div class="form-group">
                <input type="password" placeholder="Enter Password:" name="password" class="form-control">
            </div>
            <div class="form-btn">
                <input type="submit" value="Login" name="login" class="btn btn-primary">
            </div>
        </form>
    </br>
    <div><p>Not registered yet <a href="registration.php">Register Here </a></p></div>
    </div>
</body>
</html>
```

This is the code that secures from sql injection:



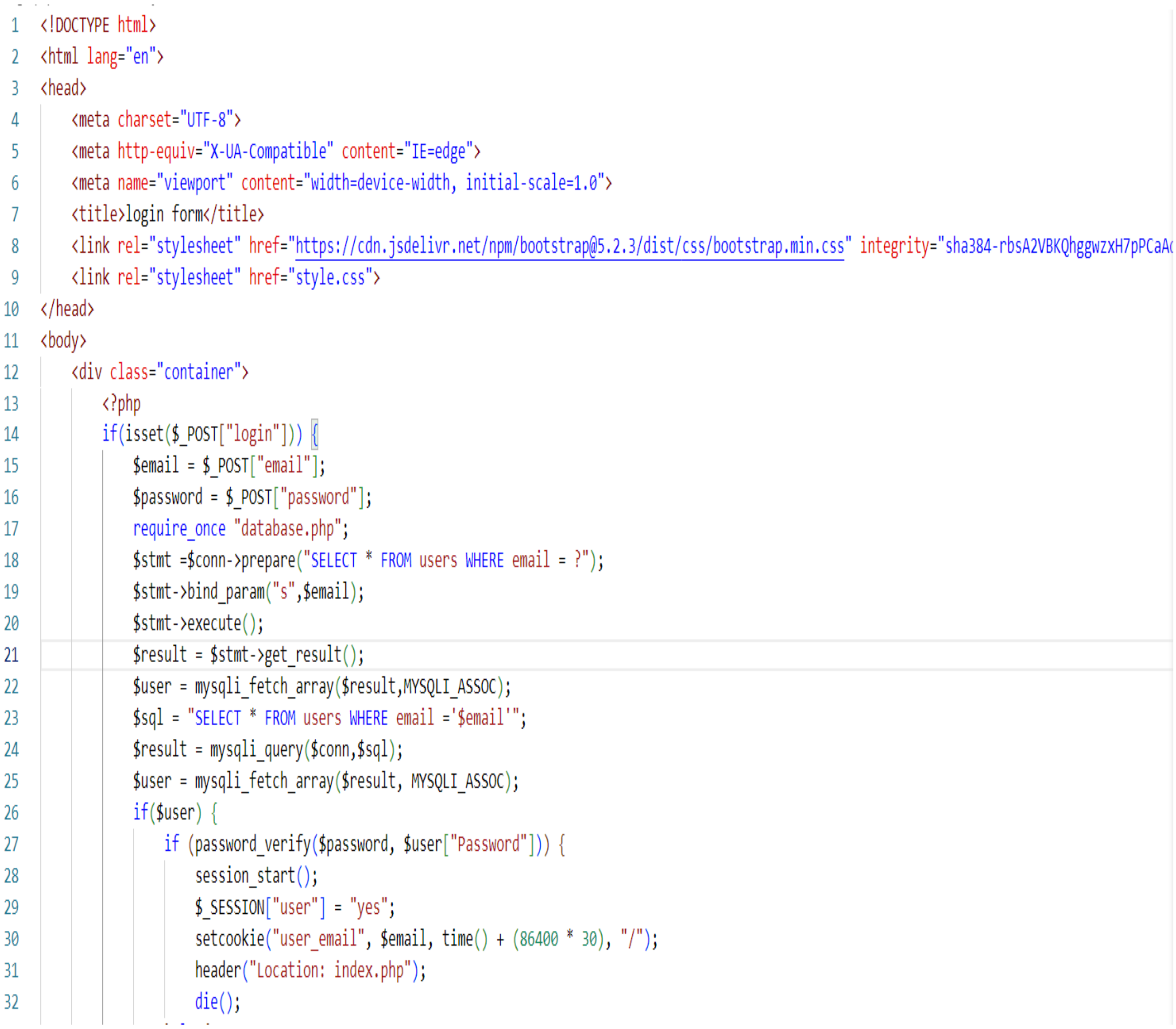
```
<?php
if(isset($_POST["login"])) {
    $email = $_POST["email"];
    $password = $_POST["password"];
    require_once "database.php";
    $stmt = $conn->prepare("SELECT * FROM users WHERE email = ?");
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $result = $stmt->get_result();
    $user = mysqli_fetch_array($result, MYSQLI_ASSOC);
```

In this code bind_param() which helps prevent SQL injection attacks by separating the query structure from the user input.

Session Hijacking :

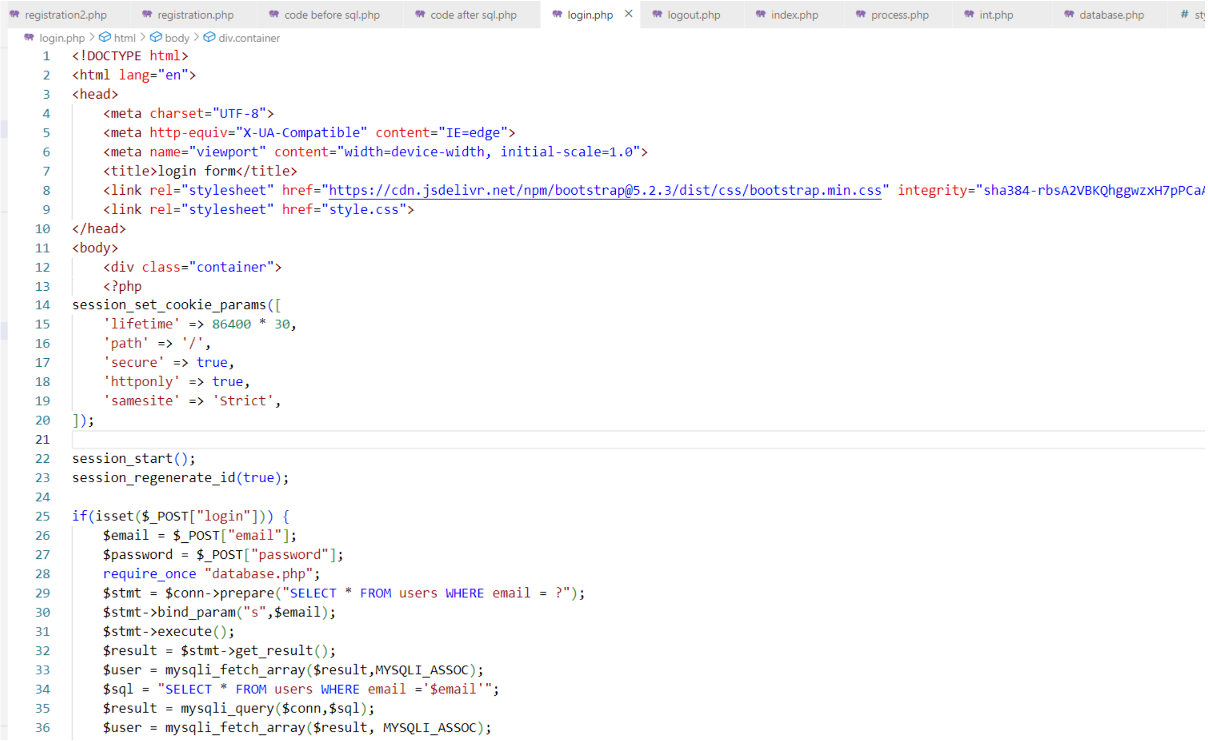
Session hijacking is a technique used by hackers to gain access to a target's computer or online accounts. In a session hijacking attack, a hacker takes control of a user's browsing session to gain access to their personal information and passwords.

The code before session hijacking:

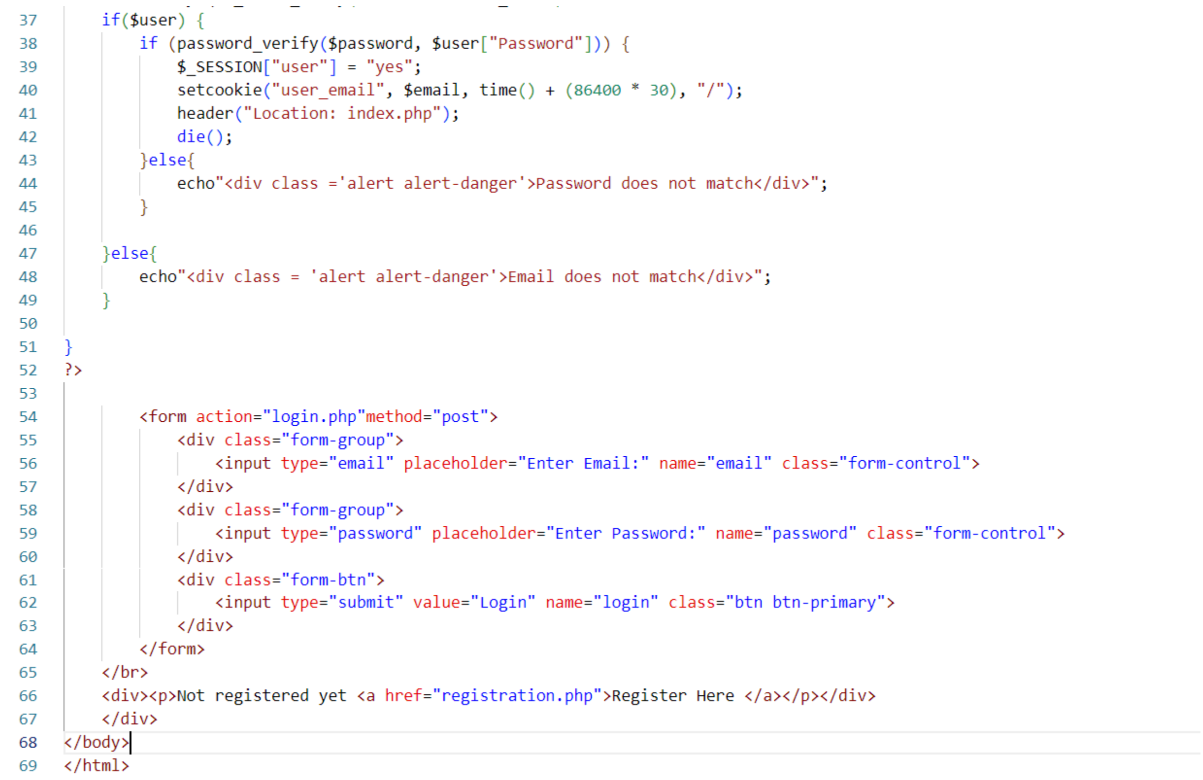


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>login form</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" integrity="sha384-rbsA2VBKQhggwzxH7pPCaA
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <?php
        if(isset($_POST["login"])) {
            $email = $_POST["email"];
            $password = $_POST["password"];
            require_once "database.php";
            $stmt =$conn->prepare("SELECT * FROM users WHERE email = ?");
            $stmt->bind_param("s",$email);
            $stmt->execute();
            $result = $stmt->get_result();
            $user = mysqli_fetch_array($result,MYSQLI_ASSOC);
            $sql = "SELECT * FROM users WHERE email ='$email'";
            $result = mysqli_query($conn,$sql);
            $user = mysqli_fetch_array($result, MYSQLI_ASSOC);
            if($user) {
                if (password_verify($password, $user["Password"])) {
                    session_start();
                    $_SESSION["user"] = "yes";
                    setcookie("user_email", $email, time() + (86400 * 30), "/");
                    header("Location: index.php");
                    die();
```

The Code after the session hijacking to secure from the attack:

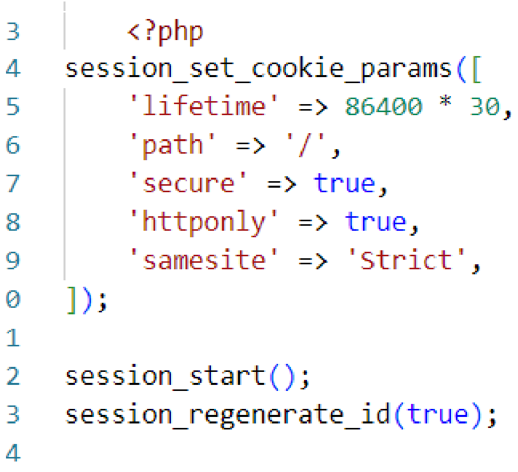


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>login form</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" integrity="sha384-rbsA2VBKQhggwzxH7pPCaA
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
    <?php
session_set_cookie_params([
    'lifetime' => 86400 * 30,
    'path' => '/',
    'secure' => true,
    'httponly' => true,
    'samesite' => 'Strict',
]);

session_start();
session_regenerate_id(true);

if(isset($_POST["login"])) {
    $email = $_POST["email"];
    $password = $_POST["password"];
    require_once "database.php";
    $stmt = $conn->prepare("SELECT * FROM users WHERE email = ?");
    $stmt->bind_param("s",$email);
    $stmt->execute();
    $result = $stmt->get_result();
    $user = mysqli_fetch_array($result,MYSQLI_ASSOC);
    $sql = "SELECT * FROM users WHERE email ='$email'";
    $result = mysqli_query($conn,$sql);
    $user = mysqli_fetch_array($result, MYSQLI_ASSOC);
```



```
    if($user) {
        if (password_verify($password, $user["Password"])) {
            $_SESSION["user"] = "yes";
            setcookie("user_email", $email, time() + (86400 * 30), "/");
            header("Location: index.php");
            die();
        }else{
            echo"<div class ='alert alert-danger'>Password does not match</div>";
        }

    }else{
        echo"<div class = 'alert alert-danger'>Email does not match</div>";
    }

}
?>

        <form action="login.php"method="post">
            <div class="form-group">
                <input type="email" placeholder="Enter Email:" name="email" class="form-control">
            </div>
            <div class="form-group">
                <input type="password" placeholder="Enter Password:" name="password" class="form-control">
            </div>
            <div class="form-btn">
                <input type="submit" value="Login" name="login" class="btn btn-primary">
            </div>
        </form>
    </br>
    <div><p>Not registered yet <a href="registration.php">Register Here </a></p></div>
    </div>
</body>
</html>
```

This line is used to prevent from the session hijacking:



```
    <?php
session_set_cookie_params([
    'lifetime' => 86400 * 30,
    'path' => '/',
    'secure' => true,
    'httponly' => true,
    'samesite' => 'Strict',
]);

session_start();
session_regenerate_id(true);
```

- Set session.cookie will prevent the session cookie from being accessible by JavaScript, which can help mitigate against certain types of attacks.
- Regenerate the session ID on every request: This will ensure that the session ID changes frequently, making it harder for an attacker to hijack a session.

Finally I pushed all my code into the github :

https://github.com/SureshXavier/doherty.git