SOFTWARE ENGINEERING LEC/LAB
MIDTERM EXAM
“REGISTER, LOGIN, LOGOUT AND WHO DID IT?”

NAME: Villanueva, Marithei H.
SECTION: UCOS 3-1

INSTRUCTIONS:

- Go back to your activity, **YOUR BUSINESS IDEAS AS SEEN WITH PDO**, what you need to do for your midterm exam is to add a **USER ENTITY** to the system you created. The user should have columns storing his personal information. For example, a user may have the first_name, last_name, address, age, etc columns for the users table. No limits on the columns that need to be filled.

- The user should be able to **LOGIN, REGISTER, AND LOGOUT** from the system and keep in mind that we should be able to keep track of that certain user who made an insertion, deletion, and updating of a record stored in the database.

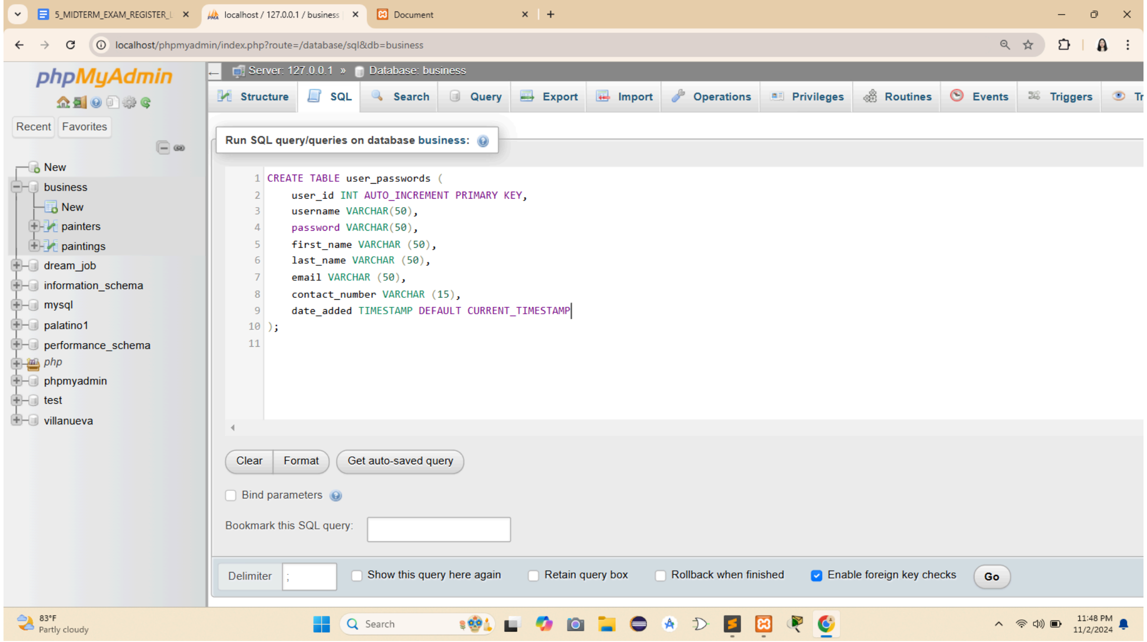
- For example if your chosen business is a web development, we should be able to **IDENTIFY THAT USER** who **CREATED** a project, **UPDATED** a project, and the **LAST TIMESTAMP** when the project was updated. So for your tables, you must add additional columns, added_by and last_updated.

- Please keep in mind that a user should **NOT BE ABLE** to access the homepage without logging in. If a user goes inside index.php and he’s not logged in, he should be redirected to the login page.

- Don’t forget to give what’s being asked from the items below.

1. How did you implement the **User entity** to the system you created? Give **the complete steps below** with some screenshots of the steps you took.

STEP 1: I created a new table called “user_passwords”.



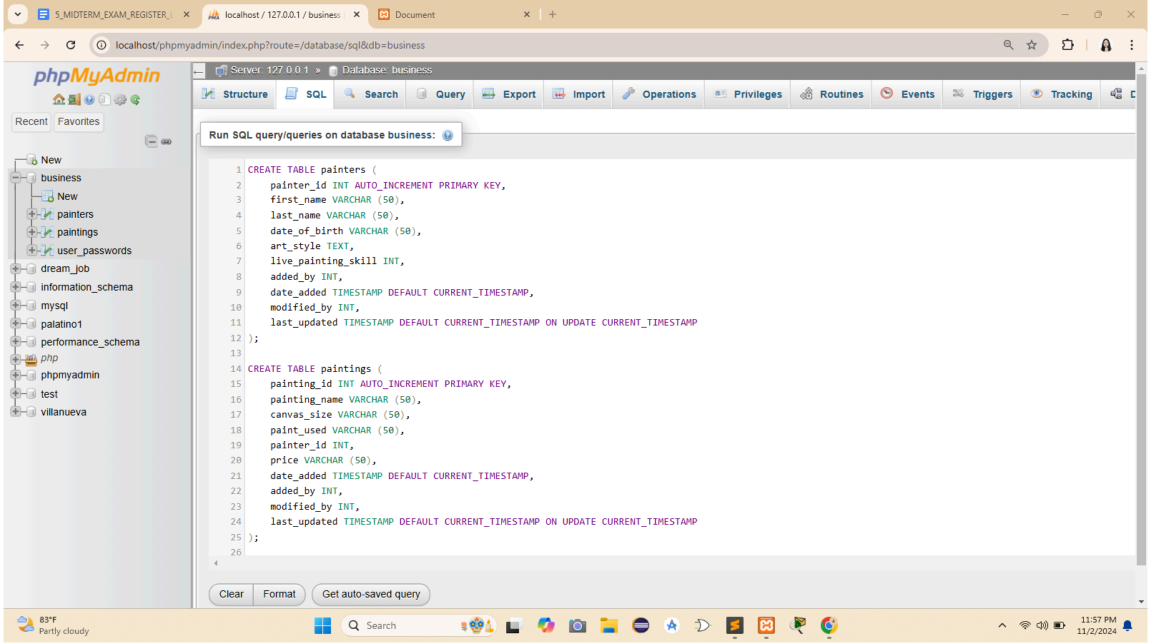
```
CREATE TABLE user_passwords (
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50),
    password VARCHAR(50),
    first_name VARCHAR (50),
    last_name VARCHAR (50),
    email VARCHAR (50),
    contact_number VARCHAR (15),
    date_added TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

□ Bind parameters

Delimiter ;

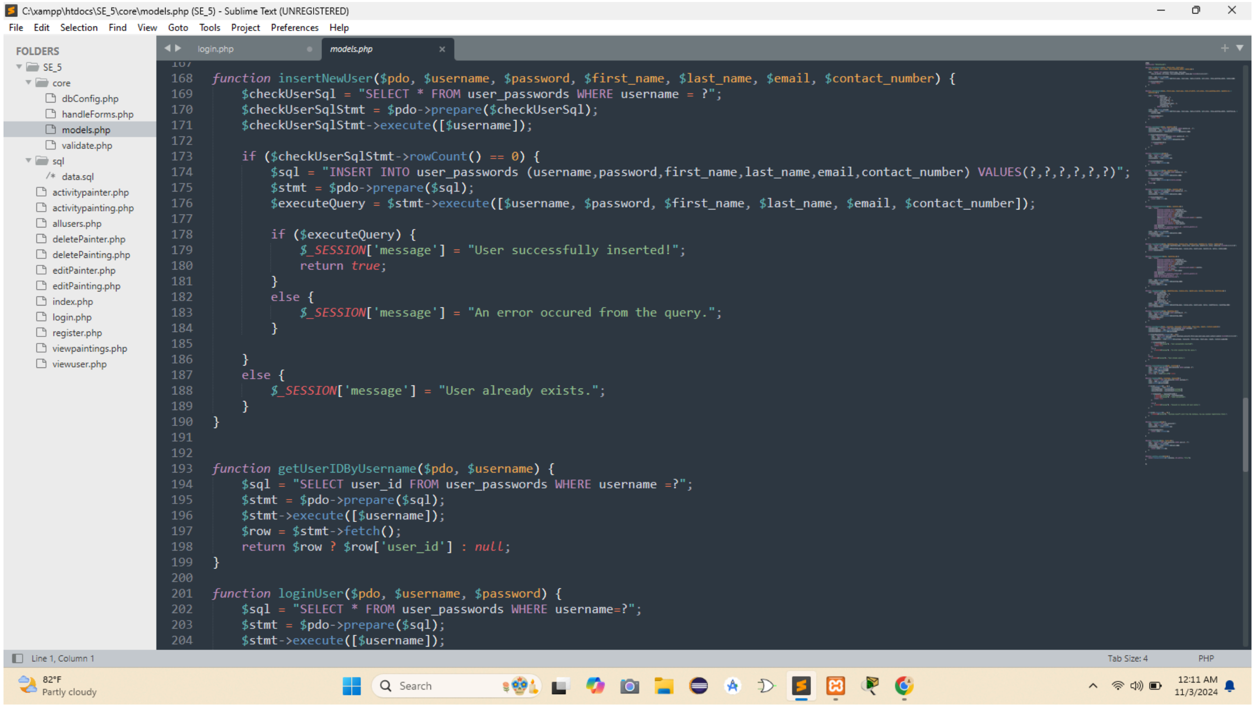
□ Show this query here again □ Retain query box □ Rollback when finished ☑ Enable foreign key checks

STEP 2: I updated the existing tables to include "added_by" to track the user who added the painter or painting in the system, "modified_by" to indicate who edited it, and "last_updated" to show the most recent timestamps of modifications.

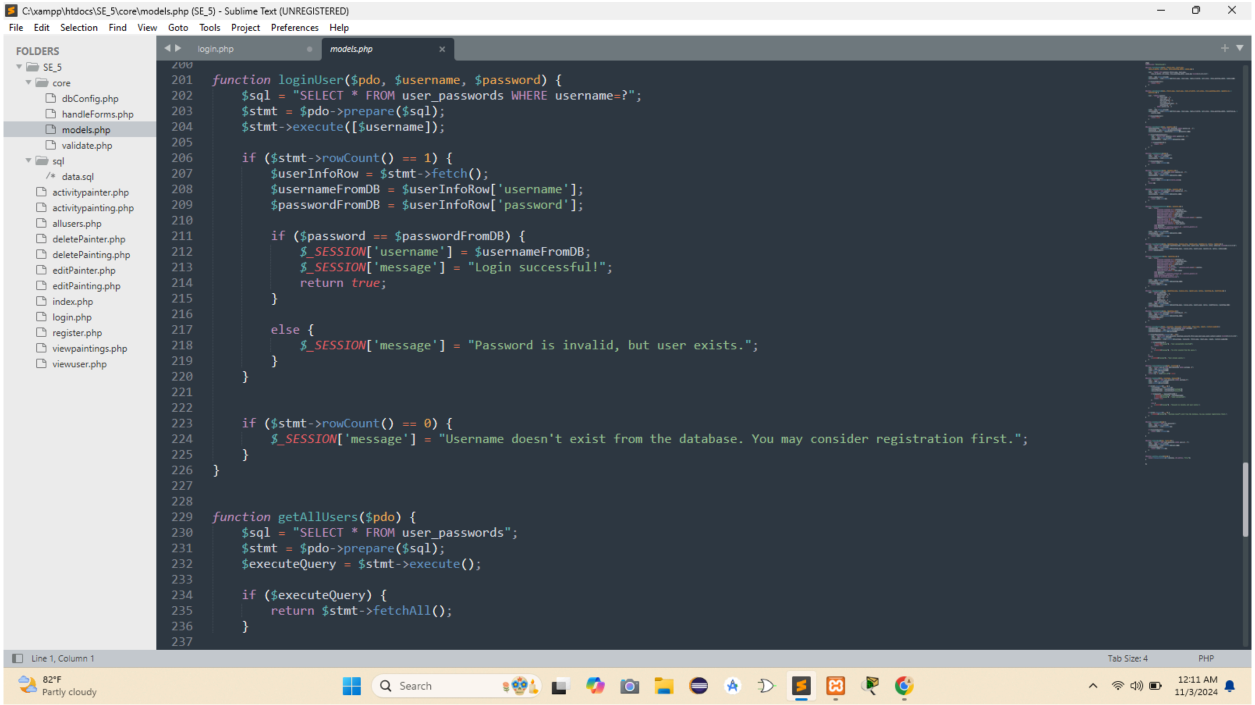


```
CREATE TABLE painters (
    painter_id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR (50),
    last_name VARCHAR (50),
    date_of_birth VARCHAR (50),
    art_style TEXT,
    live_painting_skill INT,
    added_by INT,
    date_added TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    modified_by INT,
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);

CREATE TABLE paintings (
    painting_id INT AUTO_INCREMENT PRIMARY KEY,
    painting_name VARCHAR (50),
    canvas_size VARCHAR (50),
    paint_used VARCHAR (50),
    painter_id INT,
    price VARCHAR (50),
    date_added TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    added_by INT,
    modified_by INT,
    last_updated TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
);
```

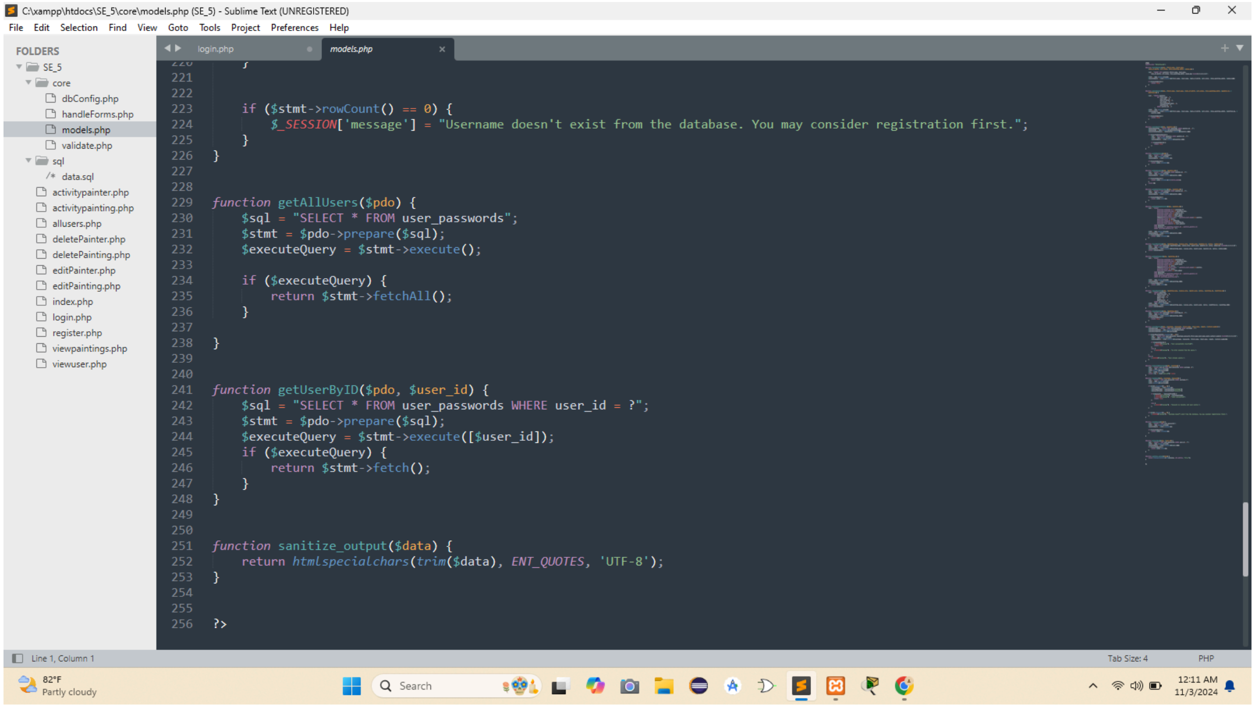
STEP 3: I declared new functions in models.php to handle user registration and login.



```php
function insertNewUser($pdo, $username, $password, $first_name, $last_name, $email, $contact_number) {
    $checkUserSql = "SELECT * FROM user_passwords WHERE username = ?";
    $checkUserSqlStmt = $pdo->prepare($checkUserSql);
    $checkUserSqlStmt->execute([$username]);

    if ($checkUserSqlStmt->rowCount() == 0) {
        $sql = "INSERT INTO user_passwords (username,password,first_name,last_name,email,contact_number) VALUES(?,?,?,?,?,?)";
        $stmt = $pdo->prepare($sql);
        $executeQuery = $stmt->execute([$username, $password, $first_name, $last_name, $email, $contact_number]);

        if ($executeQuery) {
            $_SESSION['message'] = "User successfully inserted!";
            return true;
        }
        else {
            $_SESSION['message'] = "An error occured from the query.";
        }

    }
    else {
        $_SESSION['message'] = "User already exists.";
    }
}


function getUserIDByUsername($pdo, $username) {
    $sql = "SELECT user_id FROM user_passwords WHERE username =?";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$username]);
    $row = $stmt->fetch();
    return $row ? $row['user_id'] : null;
}

function loginUser($pdo, $username, $password) {
    $sql = "SELECT * FROM user_passwords WHERE username=?";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$username]);
```

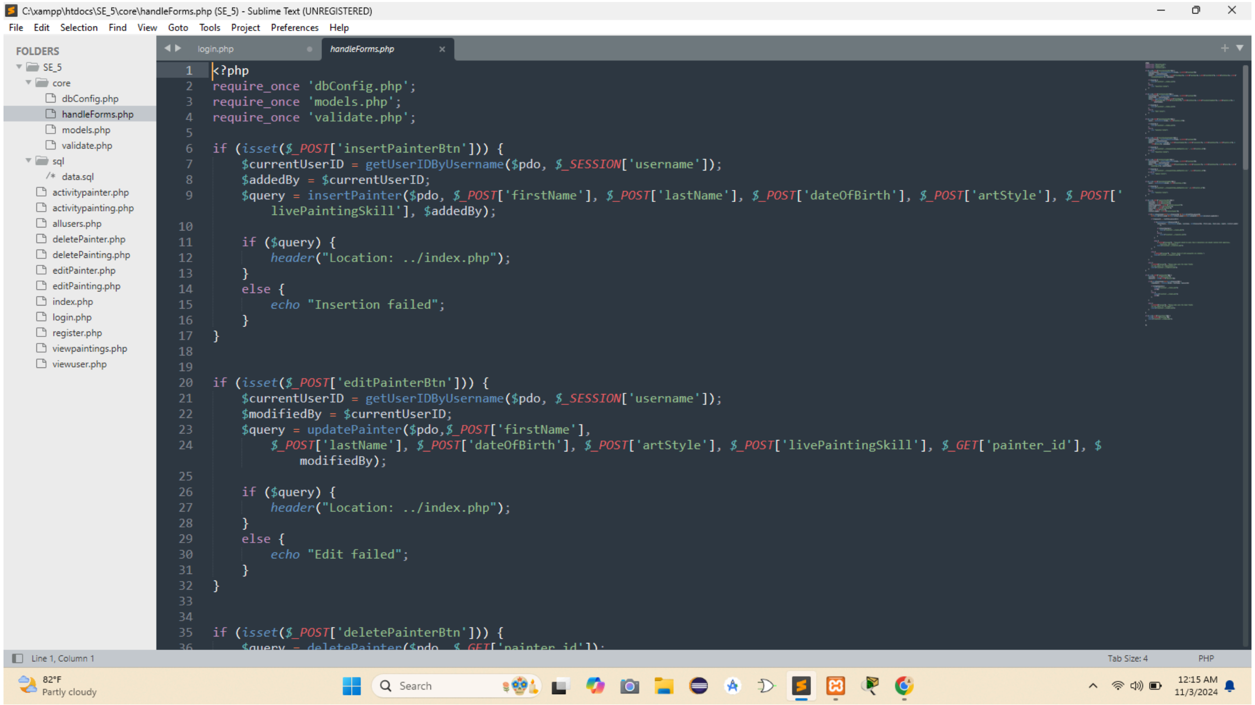


```php
function loginUser($pdo, $username, $password) {
    $sql = "SELECT * FROM user_passwords WHERE username=?";
    $stmt = $pdo->prepare($sql);
    $stmt->execute([$username]);

    if ($stmt->rowCount() == 1) {
        $userInfoRow = $stmt->fetch();
        $usernameFromDB = $userInfoRow['username'];
        $passwordFromDB = $userInfoRow['password'];

        if ($password == $passwordFromDB) {
            $_SESSION['username'] = $usernameFromDB;
            $_SESSION['message'] = "Login successful!";
            return true;
        }

        else {
            $_SESSION['message'] = "Password is invalid, but user exists.";
        }
    }


    if ($stmt->rowCount() == 0) {
        $_SESSION['message'] = "Username doesn't exist from the database. You may consider registration first.";
    }
}


function getAllUsers($pdo) {
    $sql = "SELECT * FROM user_passwords";
    $stmt = $pdo->prepare($sql);
    $executeQuery = $stmt->execute();

    if ($executeQuery) {
        return $stmt->fetchAll();
    }

```

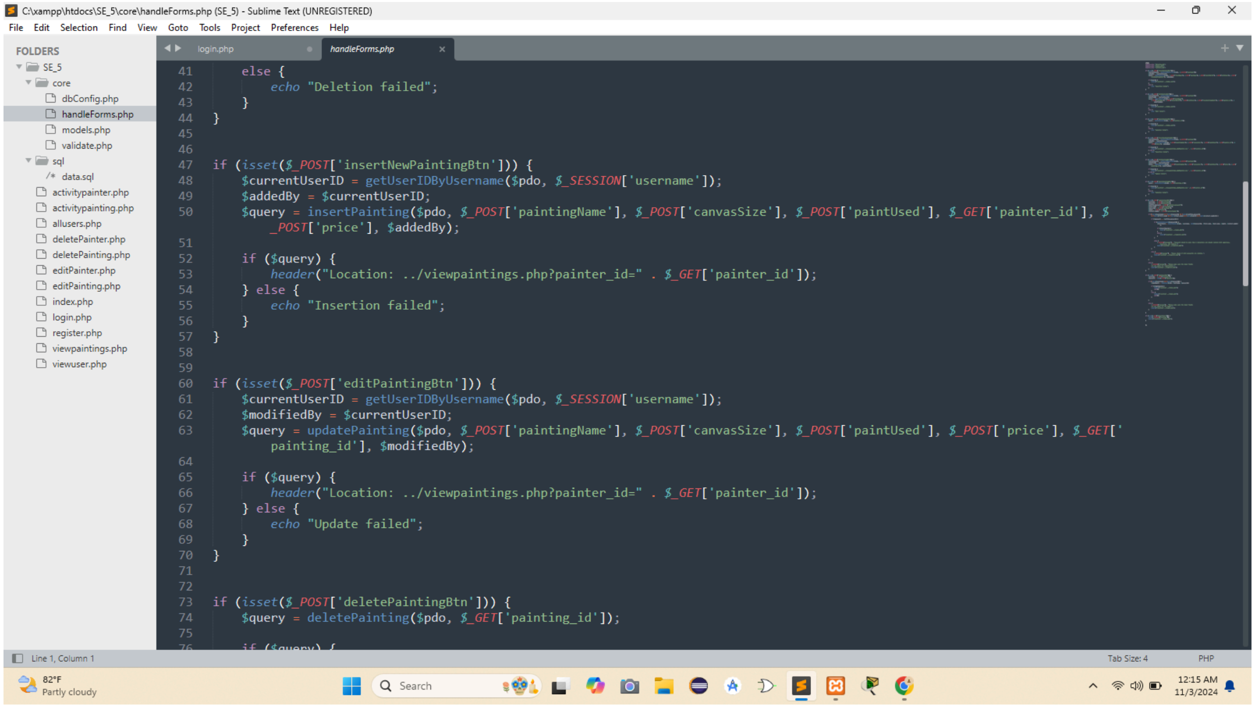


```php
    }


    if ($stmt->rowCount() == 0) {
        $_SESSION['message'] = "Username doesn't exist from the database. You may consider registration first.";
    }
}


function getAllUsers($pdo) {
    $sql = "SELECT * FROM user_passwords";
    $stmt = $pdo->prepare($sql);
    $executeQuery = $stmt->execute();

    if ($executeQuery) {
        return $stmt->fetchAll();
    }

}


function getUserByID($pdo, $user_id) {
    $sql = "SELECT * FROM user_passwords WHERE user_id = ?";
    $stmt = $pdo->prepare($sql);
    $executeQuery = $stmt->execute([$user_id]);
    if ($executeQuery) {
        return $stmt->fetch();
    }
}


function sanitize_output($data) {
    return htmlspecialchars(trim($data), ENT_QUOTES, 'UTF-8');
}


?>
```

STEP 4: I modified my handleForms to include the “added_by” and “modified_by” attributes from my previous implementation.

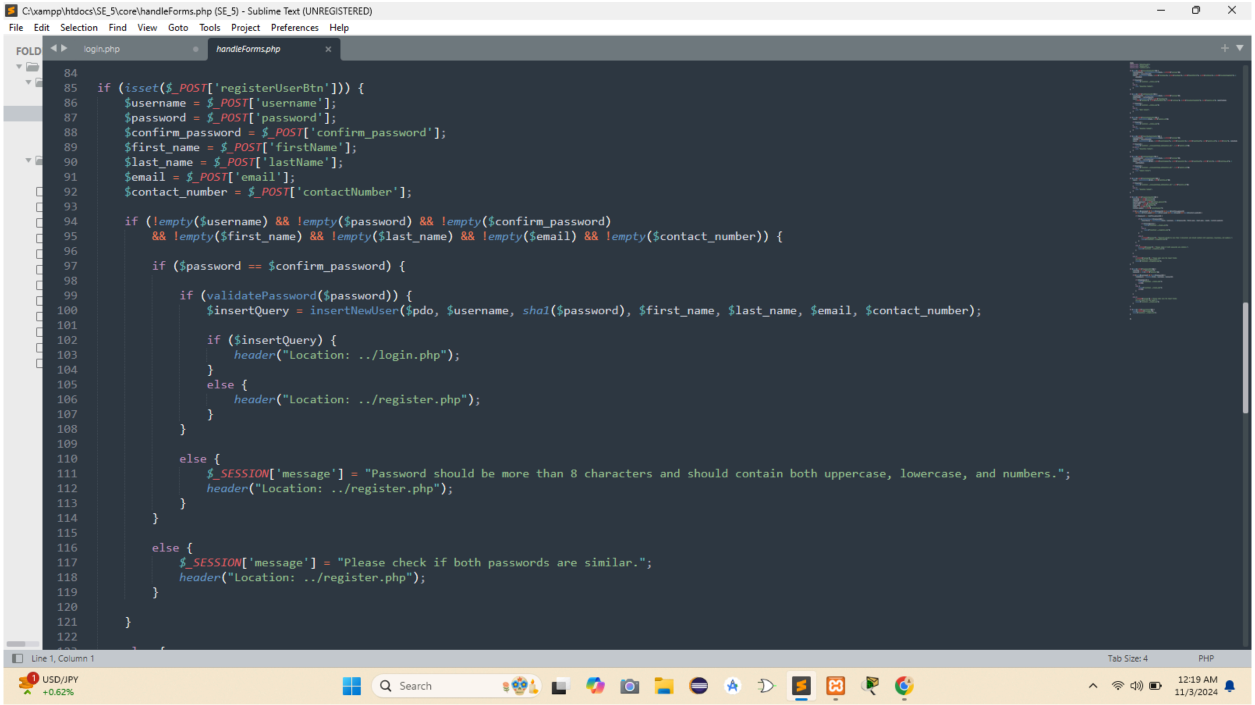


```php
<?php
require_once 'dbConfig.php';
require_once 'models.php';
require_once 'validate.php';

if (isset($_POST['insertPainterBtn'])) {
    $currentUserID = getUserIDByUsername($pdo, $_SESSION['username']);
    $addedBy = $currentUserID;
    $query = insertPainter($pdo, $_POST['firstName'], $_POST['lastName'], $_POST['dateOfBirth'], $_POST['artStyle'], $_POST['
        livePaintingSkill'], $addedBy);

    if ($query) {
        header("Location: ../index.php");
    }
    else {
        echo "Insertion failed";
    }
}


if (isset($_POST['editPainterBtn'])) {
    $currentUserID = getUserIDByUsername($pdo, $_SESSION['username']);
    $modifiedBy = $currentUserID;
    $query = updatePainter($pdo,$_POST['firstName'],
        $_POST['lastName'], $_POST['dateOfBirth'], $_POST['artStyle'], $_POST['livePaintingSkill'], $_GET['painter_id'], $
            modifiedBy);

    if ($query) {
        header("Location: ../index.php");
    }
    else {
        echo "Edit failed";
    }
}


if (isset($_POST['deletePainterBtn'])) {
    $query = deletePainter($pdo, $_GET['painter_id']);
```

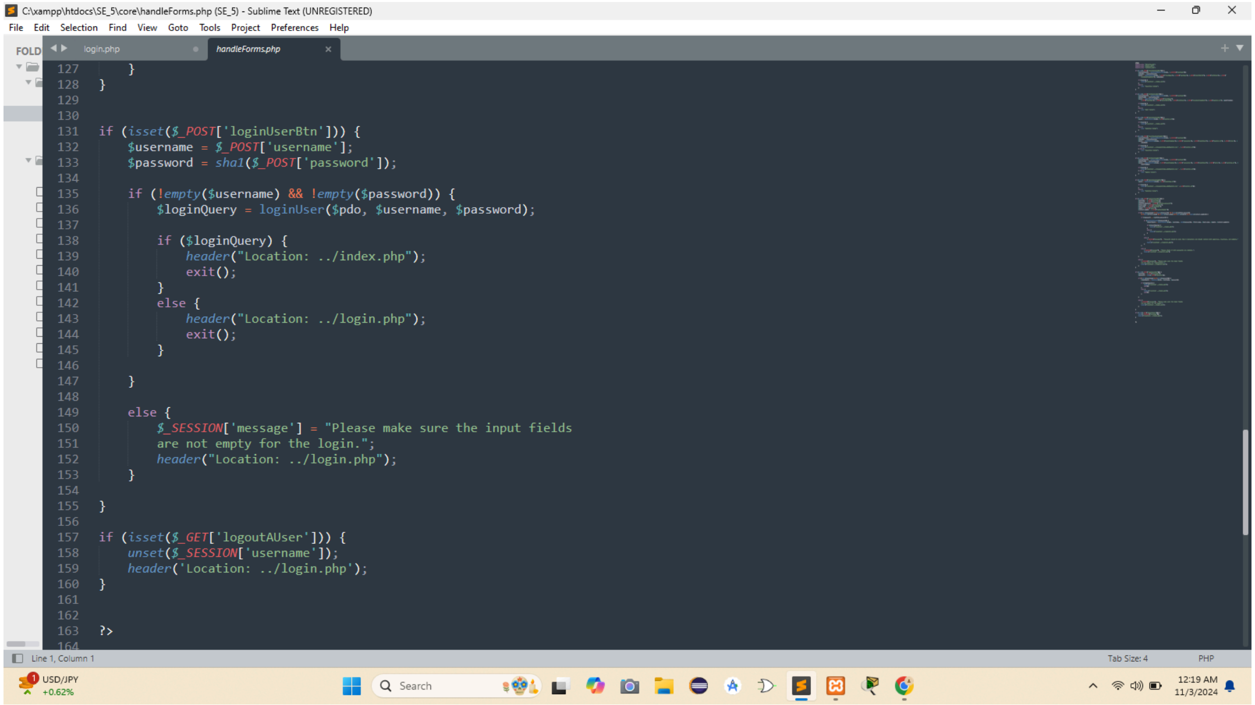


```php
    else {
        echo "Deletion failed";
    }
}


if (isset($_POST['insertNewPaintingBtn'])) {
    $currentUserID = getUserIDByUsername($pdo, $_SESSION['username']);
    $addedBy = $currentUserID;
    $query = insertPainting($pdo, $_POST['paintingName'], $_POST['canvasSize'], $_POST['paintUsed'], $_GET['painter_id'], $
        _POST['price'], $addedBy);

    if ($query) {
        header("Location: ../viewpaintings.php?painter_id=" . $_GET['painter_id']);
    } else {
        echo "Insertion failed";
    }
}


if (isset($_POST['editPaintingBtn'])) {
    $currentUserID = getUserIDByUsername($pdo, $_SESSION['username']);
    $modifiedBy = $currentUserID;
    $query = updatePainting($pdo, $_POST['paintingName'], $_POST['canvasSize'], $_POST['paintUsed'], $_POST['price'], $_GET['
        painting_id'], $modifiedBy);

    if ($query) {
        header("Location: ../viewpaintings.php?painter_id=" . $_GET['painter_id']);
    } else {
        echo "Update failed";
    }
}


if (isset($_POST['deletePaintingBtn'])) {
    $query = deletePainting($pdo, $_GET['painting_id']);

    if ($query) {
```

STEP 5: I also added the registration and login buttons in my handleForms file.


C:\xampp\htdocs\SE_5\core\handleForms.php (SE_5) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLD
login.php
handleForms.php
Line 1, Column 1
Tab Size: 4
PHP
USD/JPY
+0.62%
Search
12:19 AM
11/3/2024

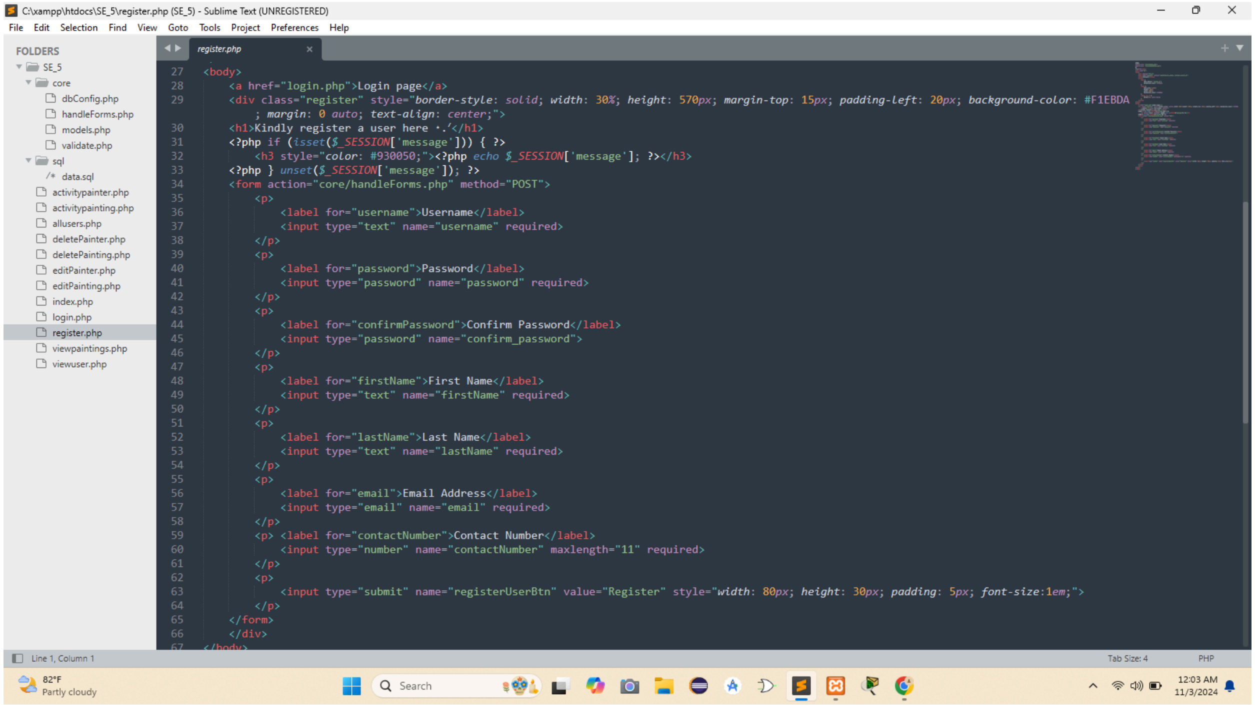
```php

if (isset($_POST['registerUserBtn'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];
    $confirm_password = $_POST['confirm_password'];
    $first_name = $_POST['firstName'];
    $last_name = $_POST['lastName'];
    $email = $_POST['email'];
    $contact_number = $_POST['contactNumber'];

    if (!empty($username) && !empty($password) && !empty($confirm_password)
        && !empty($first_name) && !empty($last_name) && !empty($email) && !empty($contact_number)) {

        if ($password == $confirm_password) {

            if (validatePassword($password)) {
                $insertQuery = insertNewUser($pdo, $username, sha1($password), $first_name, $last_name, $email, $contact_number);

                if ($insertQuery) {
                    header("Location: ../login.php");
                }
                else {
                    header("Location: ../register.php");
                }
            }

            else {
                $_SESSION['message'] = "Password should be more than 8 characters and should contain both uppercase, lowercase, and numbers.";
                header("Location: ../register.php");
            }
        }

        else {
            $_SESSION['message'] = "Please check if both passwords are similar.";
            header("Location: ../register.php");
        }

    }

```



C:\xampp\htdocs\SE_5\core\handleForms.php (SE_5) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
login.php
handleForms.php
Line 1, Column 1
Tab Size: 4
PHP
USD/JPY +0.62%
Search
12:19 AM 11/3/2024

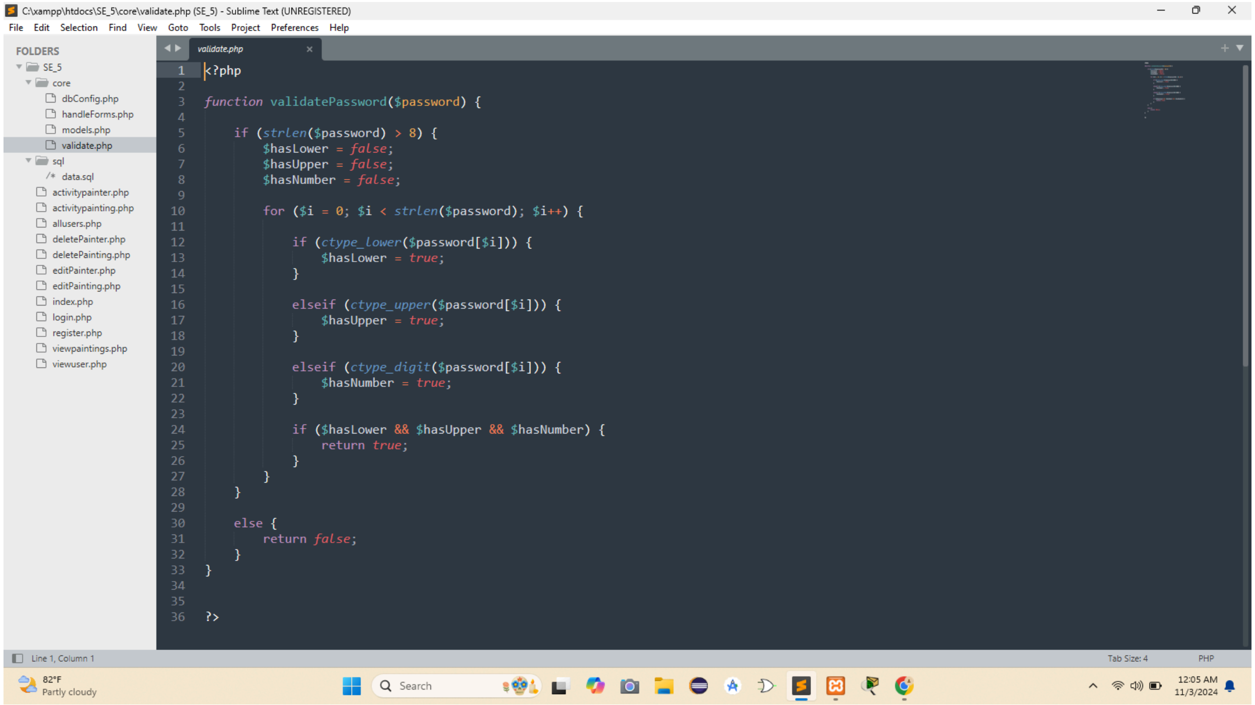
```php
    }
}


if (isset($_POST['loginUserBtn'])) {
    $username = $_POST['username'];
    $password = sha1($_POST['password']);

    if (!empty($username) && !empty($password)) {
        $loginQuery = loginUser($pdo, $username, $password);

        if ($loginQuery) {
            header("Location: ../index.php");
            exit();
        }
        else {
            header("Location: ../login.php");
            exit();
        }

    }

    else {
        $_SESSION['message'] = "Please make sure the input fields
        are not empty for the login.";
        header("Location: ../login.php");
    }

}

if (isset($_GET['logoutAUser'])) {
    unset($_SESSION['username']);
    header('Location: ../login.php');
}



?>
```

STEP 6: I created a PHP file named “register” for user registration.



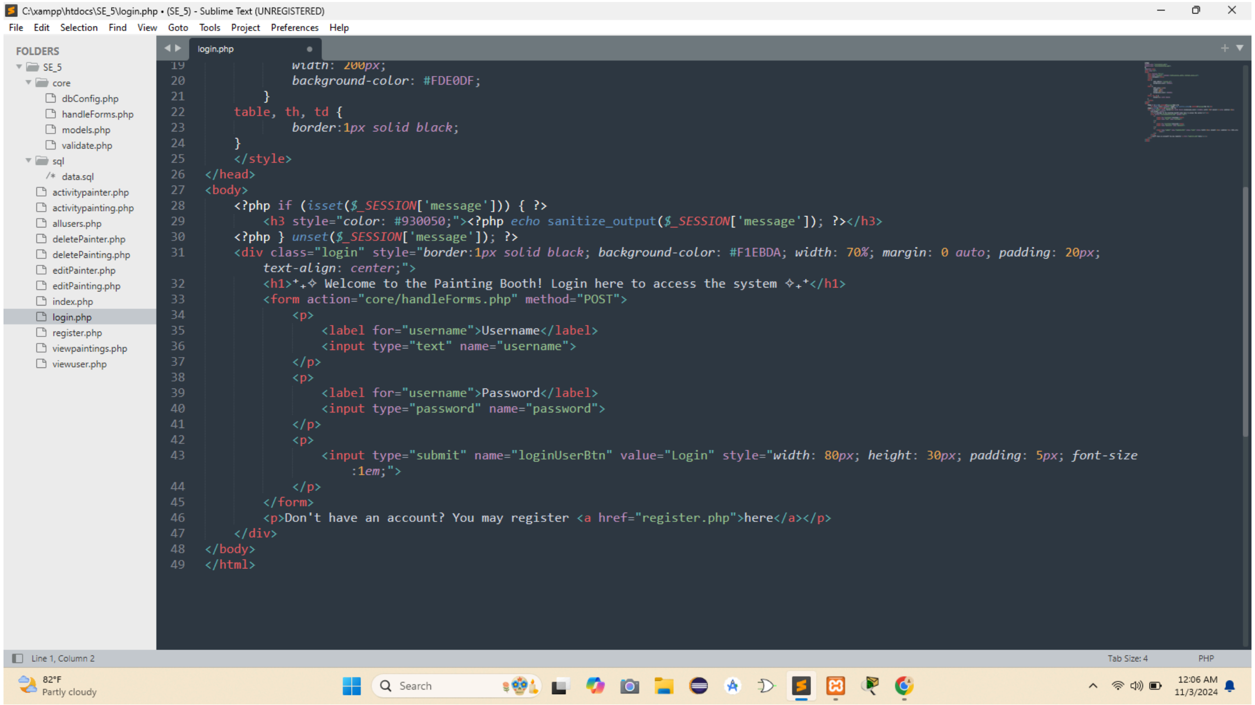
```html
<body>
    <a href="login.php">Login page</a>
    <div class="register" style="border-style: solid; width: 30%; height: 570px; margin-top: 15px; padding-left: 20px; background-color: #F1EBDA
        ; margin: 0 auto; text-align: center;">
    <h1>Kindly register a user here ·.’</h1>
    <?php if (isset($_SESSION['message'])) { ?>
        <h3 style="color: #930050;"><?php echo $_SESSION['message']; ?></h3>
    <?php } unset($_SESSION['message']); ?>
    <form action="core/handleForms.php" method="POST">
        <p>
            <label for="username">Username</label>
            <input type="text" name="username" required>
        </p>
        <p>
            <label for="password">Password</label>
            <input type="password" name="password" required>
        </p>
        <p>
            <label for="confirmPassword">Confirm Password</label>
            <input type="password" name="confirm_password">
        </p>
        <p>
            <label for="firstName">First Name</label>
            <input type="text" name="firstName" required>
        </p>
        <p>
            <label for="lastName">Last Name</label>
            <input type="text" name="lastName" required>
        </p>
        <p>
            <label for="email">Email Address</label>
            <input type="email" name="email" required>
        </p>
        <p> <label for="contactNumber">Contact Number</label>
            <input type="number" name="contactNumber" maxlength="11" required>
        </p>
        <p>
            <input type="submit" name="registerUserBtn" value="Register" style="width: 80px; height: 30px; padding: 5px; font-size:1em;">
        </p>
    </form>
    </div>
</body>
```

STEP 7: I created a PHP file named “validate” to check if the password is weak or strong.



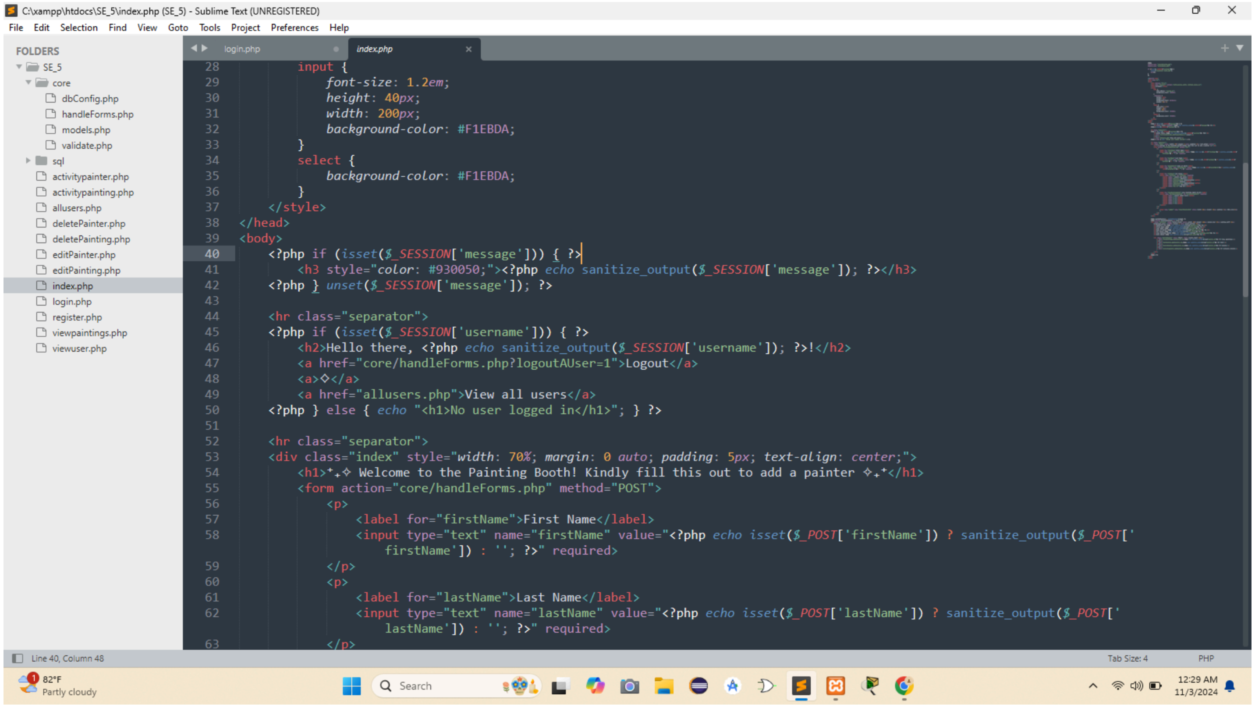
```php
<?php

function validatePassword($password) {

    if (strlen($password) > 8) {
        $hasLower = false;
        $hasUpper = false;
        $hasNumber = false;

        for ($i = 0; $i < strlen($password); $i++) {

            if (ctype_lower($password[$i])) {
                $hasLower = true;
            }

            elseif (ctype_upper($password[$i])) {
                $hasUpper = true;
            }

            elseif (ctype_digit($password[$i])) {
                $hasNumber = true;
            }

            if ($hasLower && $hasUpper && $hasNumber) {
                return true;
            }
        }
    }

    else {
        return false;
    }
}


?>
```

STEP 8: I created another PHP file named “login” for users to access the system.



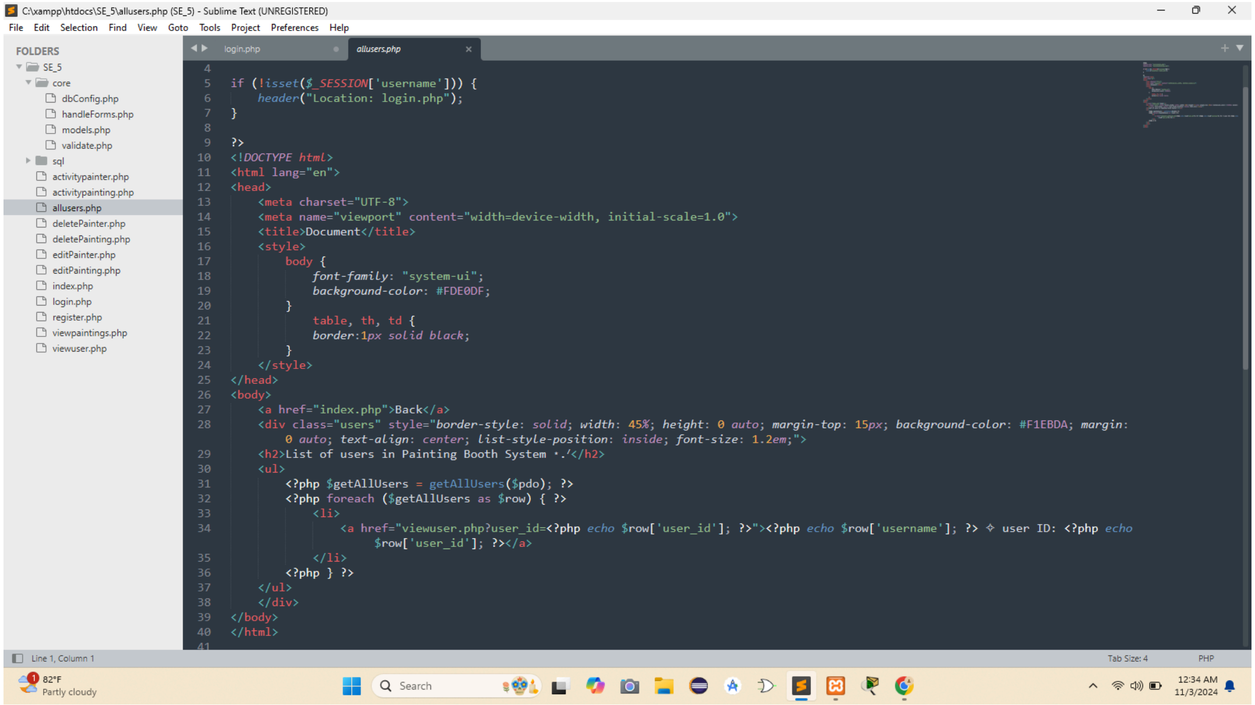
```html
            width: 200px;
            background-color: #FDE0DF;
        }
    table, th, td {
            border:1px solid black;
    }
    </style>
</head>
<body>
    <?php if (isset($_SESSION['message'])) { ?>
        <h3 style="color: #930050;"><?php echo sanitize_output($_SESSION['message']); ?></h3>
    <?php } unset($_SESSION['message']); ?>
    <div class="login" style="border:1px solid black; background-color: #F1EBDA; width: 70%; margin: 0 auto; padding: 20px;
        text-align: center;">
        <h1>⁺₊✧ Welcome to the Painting Booth! Login here to access the system ✧₊⁺</h1>
        <form action="core/handleForms.php" method="POST">
            <p>
                <label for="username">Username</label>
                <input type="text" name="username">
            </p>
            <p>
                <label for="username">Password</label>
                <input type="password" name="password">
            </p>
            <p>
                <input type="submit" name="loginUserBtn" value="Login" style="width: 80px; height: 30px; padding: 5px; font-size
                    :1em;">
            </p>
        </form>
        <p>Don't have an account? You may register <a href="register.php">here</a></p>
    </div>
</body>
</html>
```

STEP 9: I inserted code in my index file to check for successful logins and display the session message.



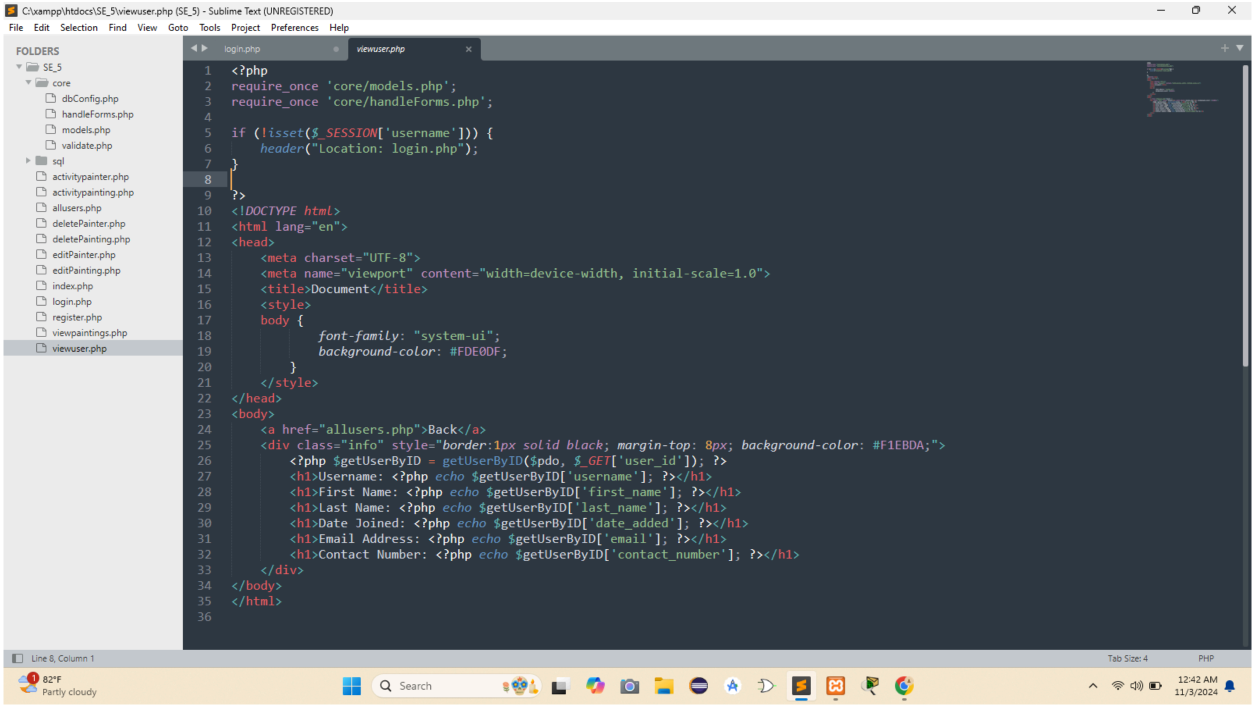
```
        input {
            font-size: 1.2em;
            height: 40px;
            width: 200px;
            background-color: #F1EBDA;
        }
        select {
            background-color: #F1EBDA;
        }
    </style>
</head>
<body>
    <?php if (isset($_SESSION['message'])) { ?>
        <h3 style="color: #930050;"><?php echo sanitize_output($_SESSION['message']); ?></h3>
    <?php } unset($_SESSION['message']); ?>

    <hr class="separator">
    <?php if (isset($_SESSION['username'])) { ?>
        <h2>Hello there, <?php echo sanitize_output($_SESSION['username']); ?>!</h2>
        <a href="core/handleForms.php?logoutAUser=1">Logout</a>
        <a>✧</a>
        <a href="allusers.php">View all users</a>
    <?php } else { echo "<h1>No user logged in</h1>"; } ?>

    <hr class="separator">
    <div class="index" style="width: 70%; margin: 0 auto; padding: 5px; text-align: center;">
        <h1>⁺₊✧ Welcome to the Painting Booth! Kindly fill this out to add a painter ✧₊⁺</h1>
        <form action="core/handleForms.php" method="POST">
            <p>
                <label for="firstName">First Name</label>
                <input type="text" name="firstName" value="<?php echo isset($_POST['firstName']) ? sanitize_output($_POST['firstName']) : ''; ?>" required>
            </p>
            <p>
                <label for="lastName">Last Name</label>
                <input type="text" name="lastName" value="<?php echo isset($_POST['lastName']) ? sanitize_output($_POST['lastName']) : ''; ?>" required>
            </p>
```

STEP 10: On the index page, there is an option to view users, so I created another PHP file named "all_users" that stores the list of users authorized to access and manage the system.



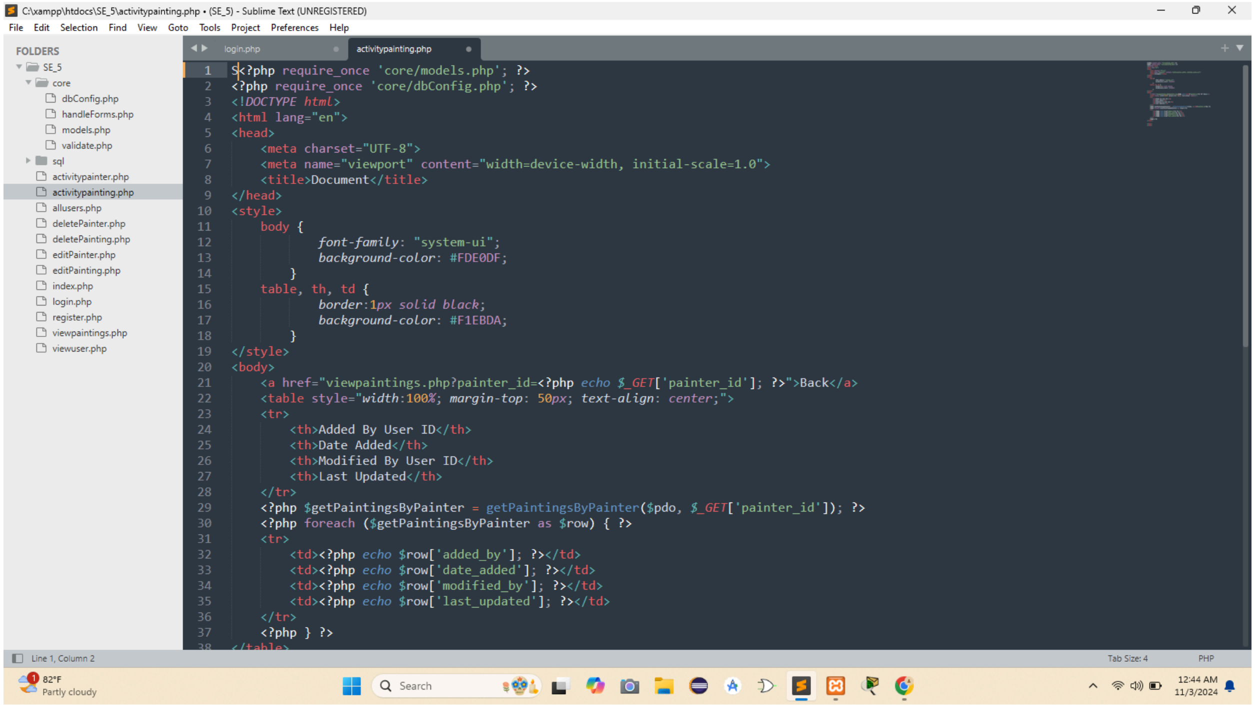
```

if (!isset($_SESSION['username'])) {
    header("Location: login.php");
}

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
        body {
            font-family: "system-ui";
            background-color: #FDE0DF;
        }
            table, th, td {
            border:1px solid black;
        }
    </style>
</head>
<body>
    <a href="index.php">Back</a>
    <div class="users" style="border-style: solid; width: 45%; height: 0 auto; margin-top: 15px; background-color: #F1EBDA; margin: 0 auto; text-align: center; list-style-position: inside; font-size: 1.2em;">
    <h2>List of users in Painting Booth System ⋆.'</h2>
    <ul>
        <?php $getAllUsers = getAllUsers($pdo); ?>
        <?php foreach ($getAllUsers as $row) { ?>
            <li>
                <a href="viewuser.php?user_id=<?php echo $row['user_id']; ?>"><?php echo $row['username']; ?> ✧ user ID: <?php echo $row['user_id']; ?></a>
            </li>
        <?php } ?>
    </ul>
    </div>
</body>
</html>
```

STEP 11: Each user in the "allusers" file is clickable, directing to the "viewuser" file I created to show the user's information.

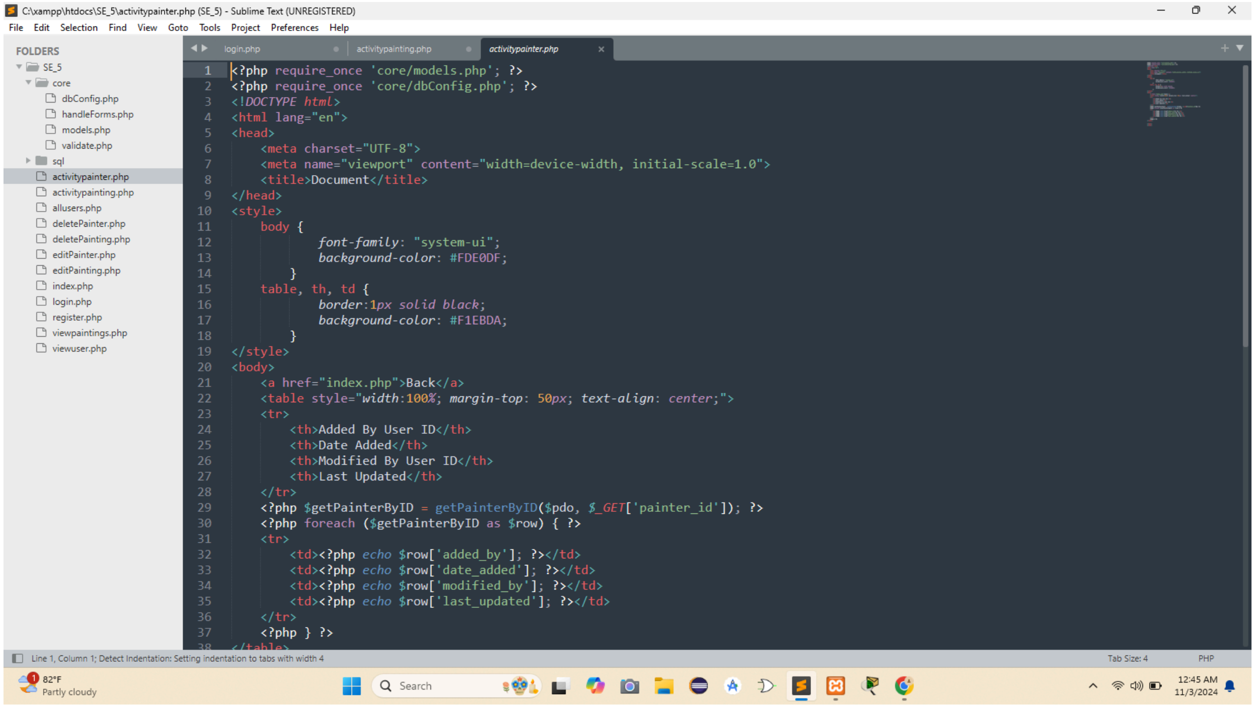


```php
<?php
require_once 'core/models.php';
require_once 'core/handleForms.php';

if (!isset($_SESSION['username'])) {
    header("Location: login.php");
}

?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
    body {
            font-family: "system-ui";
            background-color: #FDE0DF;
        }
    </style>
</head>
<body>
    <a href="allusers.php">Back</a>
    <div class="info" style="border:1px solid black; margin-top: 8px; background-color: #F1EBDA;">
        <?php $getUserByID = getUserByID($pdo, $_GET['user_id']); ?>
        <h1>Username: <?php echo $getUserByID['username']; ?></h1>
        <h1>First Name: <?php echo $getUserByID['first_name']; ?></h1>
        <h1>Last Name: <?php echo $getUserByID['last_name']; ?></h1>
        <h1>Date Joined: <?php echo $getUserByID['date_added']; ?></h1>
        <h1>Email Address: <?php echo $getUserByID['email']; ?></h1>
        <h1>Contact Number: <?php echo $getUserByID['contact_number']; ?></h1>
    </div>
</body>
</html>
```

STEP 12: I created two new files named "activitypainting" and "activitypainter" to view the activity details that occurred for each specific record.

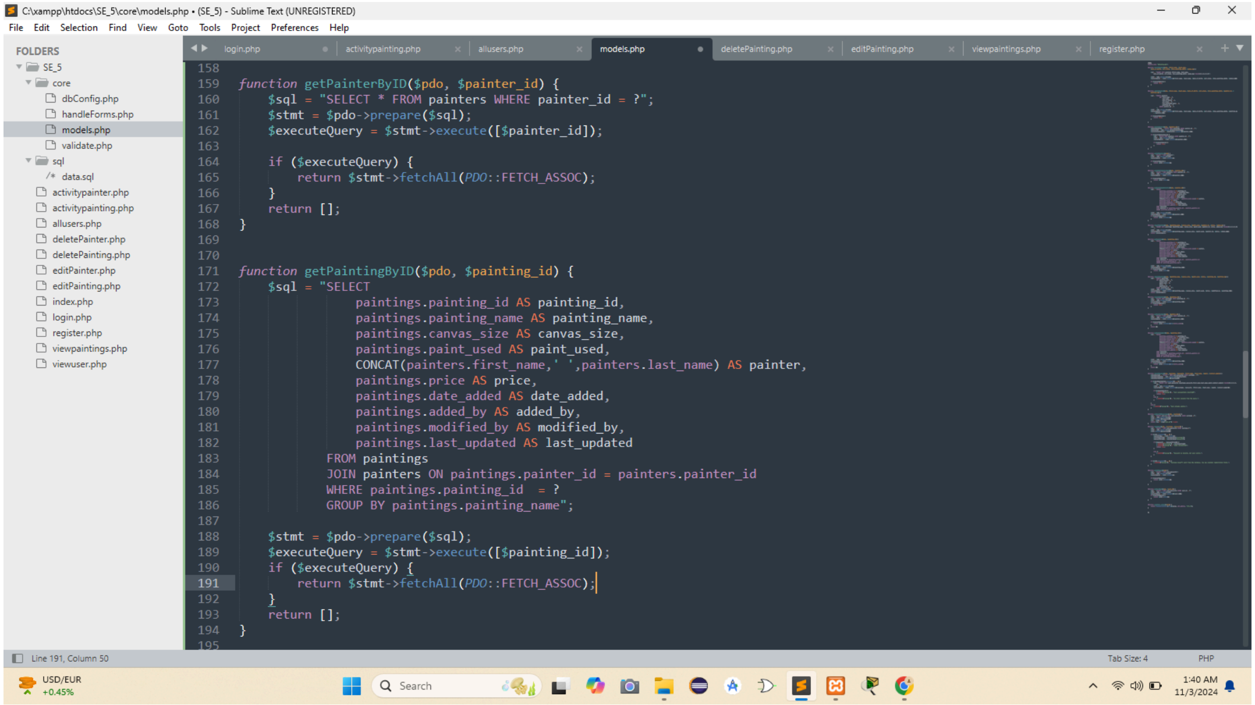


```php
S<?php require_once 'core/models.php'; ?>
<?php require_once 'core/dbConfig.php'; ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<style>
    body {
            font-family: "system-ui";
            background-color: #FDE0DF;
        }
    table, th, td {
            border:1px solid black;
            background-color: #F1EBDA;
        }
</style>
<body>
    <a href="viewpaintings.php?painter_id=<?php echo $_GET['painter_id']; ?>">Back</a>
    <table style="width:100%; margin-top: 50px; text-align: center;">
    <tr>
        <th>Added By User ID</th>
        <th>Date Added</th>
        <th>Modified By User ID</th>
        <th>Last Updated</th>
    </tr>
    <?php $getPaintingsByPainter = getPaintingsByPainter($pdo, $_GET['painter_id']); ?>
    <?php foreach ($getPaintingsByPainter as $row) { ?>
    <tr>
        <td><?php echo $row['added_by']; ?></td>
        <td><?php echo $row['date_added']; ?></td>
        <td><?php echo $row['modified_by']; ?></td>
        <td><?php echo $row['last_updated']; ?></td>
    </tr>
    <?php } ?>
</table>
```



```
<?php require_once 'core/models.php'; ?>
<?php require_once 'core/dbConfig.php'; ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<style>
    body {
            font-family: "system-ui";
            background-color: #FDE0DF;
        }
    table, th, td {
            border:1px solid black;
            background-color: #F1EBDA;
        }
</style>
<body>
    <a href="index.php">Back</a>
    <table style="width:100%; margin-top: 50px; text-align: center;">
    <tr>
        <th>Added By User ID</th>
        <th>Date Added</th>
        <th>Modified By User ID</th>
        <th>Last Updated</th>
    </tr>
    <?php $getPainterByID = getPainterByID($pdo, $_GET['painter_id']); ?>
    <?php foreach ($getPainterByID as $row) { ?>
    <tr>
        <td><?php echo $row['added_by']; ?></td>
        <td><?php echo $row['date_added']; ?></td>
        <td><?php echo $row['modified_by']; ?></td>
        <td><?php echo $row['last_updated']; ?></td>
    </tr>
    <?php } ?>
</table>
```

STEP 13: Lastly, I created another function to handle empty rows, as some records have not yet been edited, resulting in the "modified_by" field being empty.



```

function getPainterByID($pdo, $painter_id) {
    $sql = "SELECT * FROM painters WHERE painter_id = ?";
    $stmt = $pdo->prepare($sql);
    $executeQuery = $stmt->execute([$painter_id]);

    if ($executeQuery) {
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
    return [];
}


function getPaintingByID($pdo, $painting_id) {
    $sql = "SELECT
                paintings.painting_id AS painting_id,
                paintings.painting_name AS painting_name,
                paintings.canvas_size AS canvas_size,
                paintings.paint_used AS paint_used,
                CONCAT(painters.first_name,' ',painters.last_name) AS painter,
                paintings.price AS price,
                paintings.date_added AS date_added,
                paintings.added_by AS added_by,
                paintings.modified_by AS modified_by,
                paintings.last_updated AS last_updated
            FROM paintings
            JOIN painters ON paintings.painter_id = painters.painter_id
            WHERE paintings.painting_id  = ?
            GROUP BY paintings.painting_name";

    $stmt = $pdo->prepare($sql);
    $executeQuery = $stmt->execute([$painting_id]);
    if ($executeQuery) {
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
    return [];
}

```

2. Show a screen-recorded demonstration of the system you created and paste its **Google Drive link** below. This should include **registration**, **logging in**, **inserting**, **updating**, **deleting** records from the system and **logging out**. Don't forget to set its visibility to **anyone with link.**

https://drive.google.com/file/d/1khGcwrad-bKHJBVkHenZLeTpC2vS-cYP/view?usp=sharing

3. Paste **GitHub** link of project below.

https://github.com/mystheirious/business_login/tree/main/SE_5