

# Experiment - 6

## Open Source Computing

Name : Veeransh Shah  
Reg ID: 221070063

### Aim:

The aim of this project is to develop a user authentication system that includes registration and login functionality using PHP and MySQL. The project also incorporates field validation to ensure data integrity and security.

### Theory:

#### Introduction:

In modern web applications, user authentication plays a crucial role in ensuring that only authorized users can access certain sections of a website. This project involves creating two core components:

Registration Form: Allows users to sign up by providing a username, email, and password.

Login Form: Allows registered users to log in with their credentials.

#### PHP in Web Development:

PHP (Hypertext Preprocessor) is a server-side scripting language widely used for web development. It enables dynamic web pages to interact with databases like MySQL, making it ideal for creating login systems.

#### MySQL Database:

MySQL is a relational database management system that stores data in tables. In this project, user information (username, email, password) will be stored securely in the MySQL database.

#### Password Hashing:

For security reasons, passwords should never be stored as plain text. PHP provides functions like `password_hash()` to securely hash passwords before storing them in the database. During login, the password entered by the user is compared to the stored hashed password using `password_verify()`.

## Validation:

Client-side validation: Generally done using HTML and JavaScript (not covered here but important for enhanced user experience).

Server-side validation: Performed using PHP to ensure valid and safe data is passed to the server, preventing issues like SQL injection and ensuring data integrity.

## Registration Process:

User enters details: username, email, password, and confirm password.

PHP validates the fields for completeness, proper email format, and password match.

If all validations pass, the password is hashed and stored in the database along with the username and email.

## Login Process:

User enters email and password.

PHP verifies the email exists in the database and checks if the password entered matches the stored hashed password using `password_verify()`.

If successful, the user is redirected to the protected section of the website.

## Code:

```
<?php
// Include the database configuration file
include 'config.php';

// Define variables and initialize with empty values
$username = $email = $password = $confirm_password = "";
$username_err = $email_err = $password_err = $confirm_password_err = "";

// Process form data when form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {

    // Validate username
    if (empty(trim($_POST["username"]))) {
        $username_err = "Please enter a username.";
    } else {
```

```

$username = trim($_POST["username"]);
}

// Validate email
if (empty(trim($_POST["email"]))) {
    $email_err = "Please enter an email.";
} elseif (!filter_var(trim($_POST["email"]), FILTER_VALIDATE_EMAIL)) {
    $email_err = "Invalid email format.";
} else {
    $email = trim($_POST["email"]);
}

// Validate password
if (empty(trim($_POST["password"]))) {
    $password_err = "Please enter a password.";
} elseif (strlen(trim($_POST["password"])) < 6) {
    $password_err = "Password must have at least 6 characters.";
} else {
    $password = trim($_POST["password"]);
}

// Validate confirm password
if (empty(trim($_POST["confirm_password"]))) {
    $confirm_password_err = "Please confirm password.";
} else {
    $confirm_password = trim($_POST["confirm_password"]);
    if (empty($password_err) && ($password != $confirm_password)) {
        $confirm_password_err = "Password did not match.";
    }
}

// Check for input errors before inserting in the database
if (empty($username_err) && empty($email_err) && empty($password_err)
&& empty($confirm_password_err)) {

    // Prepare an insert statement
    $sql = "INSERT INTO users (username, email, password) VALUES (?, ?,
?)"

    if ($stmt = mysqli_prepare($link, $sql)) {

```

```

        // Bind variables to the prepared statement as parameters
        mysqli_stmt_bind_param($stmt, "sss", $param_username,
$param_email, $param_password);

        // Set parameters
        $param_username = $username;
        $param_email = $email;
        $param_password = password_hash($password, PASSWORD_DEFAULT);
// Password hash

        // Attempt to execute the prepared statement
        if (mysqli_stmt_execute($stmt)) {
            // Redirect to login page
            header("location: login.php");
        } else {
            echo "Something went wrong. Please try again later.";
        }

        // Close statement
        mysqli_stmt_close($stmt);
    }
}

// Close connection
mysqli_close($link);
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Register</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="container">
        <h2>Register</h2>
        <form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);
?>" method="post">

```

```

        <div>
            <label>Username</label>
            <input type="text" name="username" value="<?php echo
$username; ?>">
            <span class="error"><?php echo $username_err; ?></span>
        </div>
        <div>
            <label>Email</label>
            <input type="email" name="email" value="<?php echo $email;
?>">
            <span class="error"><?php echo $email_err; ?></span>
        </div>
        <div>
            <label>Password</label>
            <input type="password" name="password">
            <span class="error"><?php echo $password_err; ?></span>
        </div>
        <div>
            <label>Confirm Password</label>
            <input type="password" name="confirm_password">
            <span class="error"><?php echo $confirm_password_err;
?></span>
        </div>
        <div>
            <input type="submit" value="Submit">
        </div>
        <p>Already have an Account? <a href="login.php">Login
here</a>.</p>
    </form>
</div>
</body>
</html>

```

```
<?php
```

```

// Include config file
include 'config.php';

// Define variables and initialize with empty values
$email = $password = "";
$email_err = $password_err = "";

// Processing form data when form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {

    // Check if email is empty
    if (empty(trim($_POST["email"]))) {
        $email_err = "Please enter email.";
    } else {
        $email = trim($_POST["email"]);
    }

    // Check if password is empty
    if (empty(trim($_POST["password"]))) {
        $password_err = "Please enter your password.";
    } else {
        $password = trim($_POST["password"]);
    }

    // Validate credentials
    if (empty($email_err) && empty($password_err)) {
        // Prepare a select statement
        $sql = "SELECT id, username, email, password FROM users WHERE email
= ?";

        if ($stmt = mysqli_prepare($link, $sql)) {
            // Bind variables to the prepared statement as parameters
            mysqli_stmt_bind_param($stmt, "s", $param_email);

            // Set parameters
            $param_email = $email;

            // Attempt to execute the prepared statement
            if (mysqli_stmt_execute($stmt)) {
                // Store result

```

```

        mysqli_stmt_store_result($stmt);

        // Check if email exists, if yes then verify password
        if (mysqli_stmt_num_rows($stmt) == 1) {
            // Bind result variables
            mysqli_stmt_bind_result($stmt, $id, $username, $email,
$hashed_password);
            if (mysqli_stmt_fetch($stmt)) {
                if (password_verify($password, $hashed_password)) {
                    // Redirect to dashboard or homepage
                    header("location: welcome.php");
                } else {
                    $password_err = "The password you entered was
not valid.";
                }
            }
        } else {
            $email_err = "No account found with that email.";
        }
    } else {
        echo "Oops! Something went wrong. Please try again later.";
    }

    // Close statement
    mysqli_stmt_close($stmt);
}

}

// Close connection
mysqli_close($link);
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Login</title>
</head>
<body>

```

```
<h2>Login</h2>
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>"
method="post">
    <div>
        <label>Email</label>
        <input type="text" name="email" value="<?php echo $email; ?>">
        <span><?php echo $email_err; ?></span>
    </div>
    <div>
        <label>Password</label>
        <input type="password" name="password">
        <span><?php echo $password_err; ?></span>
    </div>
    <div>
        <input type="submit" value="Submit">
    </div>
</form>
</body>
</html>
```

```
<?php
// Database credentials
define('DB_SERVER', 'localhost');
define('DB_USERNAME', 'root');
define('DB_PASSWORD', '');
define('DB_NAME', 'user_registration');

// Attempt to connect to MySQL database
$link = mysqli_connect(DB_SERVER, DB_USERNAME, DB_PASSWORD, DB_NAME);

// Check connection
if ($link === false) {
    die("ERROR: Could not connect. " . mysqli_connect_error());
}
?>
```



**Output:**