LECTURE #6 FOR PHP

Serever-side programming language

Last lecture

Youssef Khaled

PHP SECURITY

- **Input Validation:**

Always validate and sanitize user inputs to prevent SQL injection, cross-site scripting (XSS), and other security vulnerabilities.

```
$username = $_POST['username'];
$safeUsername = htmlspecialchars($username, ENT_QUOTES, 'UTF-8');
```

PHP SECURITY

- **Secure Password Handling:**

Hash user passwords using strong algorithms like bcrypt.

```
$password = password_hash($_POST['password'], PASSWORD_BCRYPT);
```

- **Avoid Register Globals:**

Disable register_globals in your PHP configuration to prevent security risks associated with global variables.

```
// Disable register_globals in php.ini
```

DATETIME FUNCTIONS

- $currentDateTime = new DateTime();
- echo $currentDateTime->format('Y-m-d H:i:s');

C O O K I E S

- **Set Cookie Parameters:**

Define cookie parameters, such as expiration time and path, to enhance security.

```
setcookie("user", "John Doe", time() + 3600, "/");
```

- **Secure and HTTPOnly Flags:**

Use the 'secure' flag for HTTPS-only cookies and the 'HttpOnly' flag to prevent JavaScript access.

```
setcookie("user", "John Doe", time() + 3600, "/", "", true, true);
```

S E S S I O N S

Best Practices:

1. **Regenerate Session ID:**
    1. Regenerate session ID periodically to prevent session fixation attacks.

```
session_regenerate_id(true);
```

```
// Start a session
session_start();
// Set session variables
$_SESSION["username"] = "JohnDoe";
```

HEADERS AND REDIRECTION

- **Prevent Caching:**

Use headers to prevent caching sensitive information.

```
header("Cache-Control: no-store, no-cache, must-revalidate");
```

- **Safe Redirection:**

Validate and sanitize user input before using it in header redirection to prevent open redirects.

```
$redirectURL = filter_var($_GET['redirect'], FILTER_SANITIZE_URL);

header("Location: " . $redirectURL);
```