Web Programming

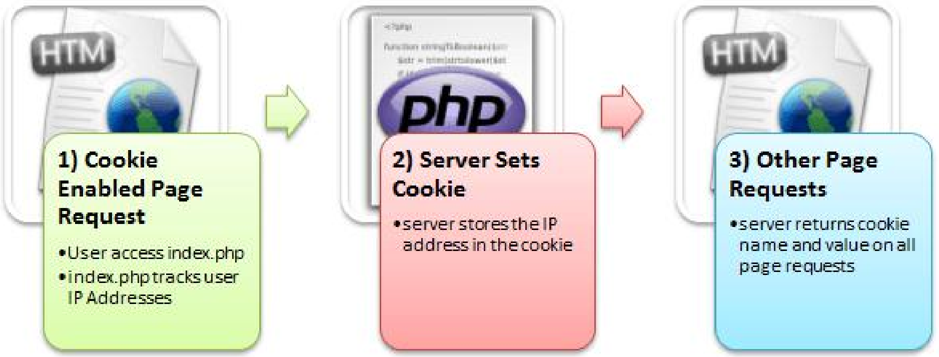
#9 Cookie & Session

Herman Kabetta, M.T.

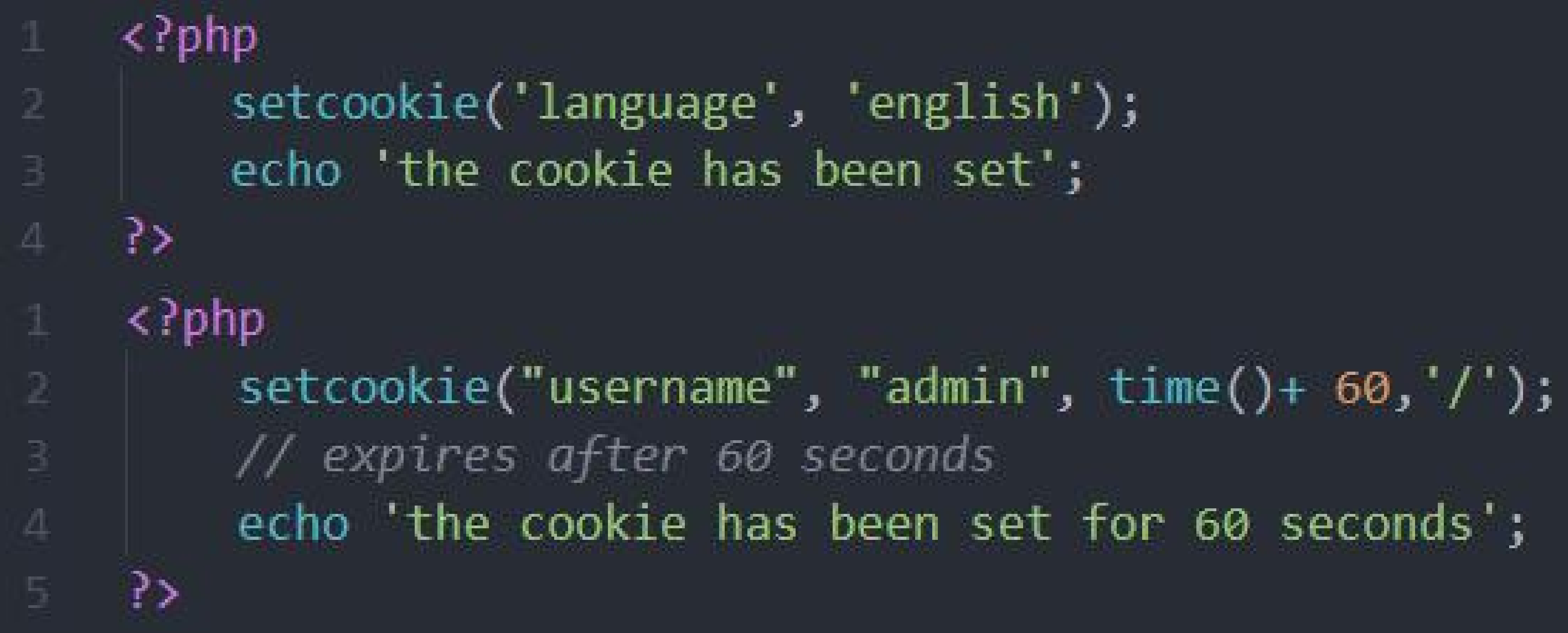
hermanka.github.io

What is Cookie?

- Small file with the maximum size of 4KB that the web server stores on the client computer.
- HTTP is a stateless protocol; cookies allow us to track the state of the application.
- Personalizing the user experience using cookies.
- A cookie can only be read from the domain that it has been issued from.
- Cookie must be started before any HTML tags.

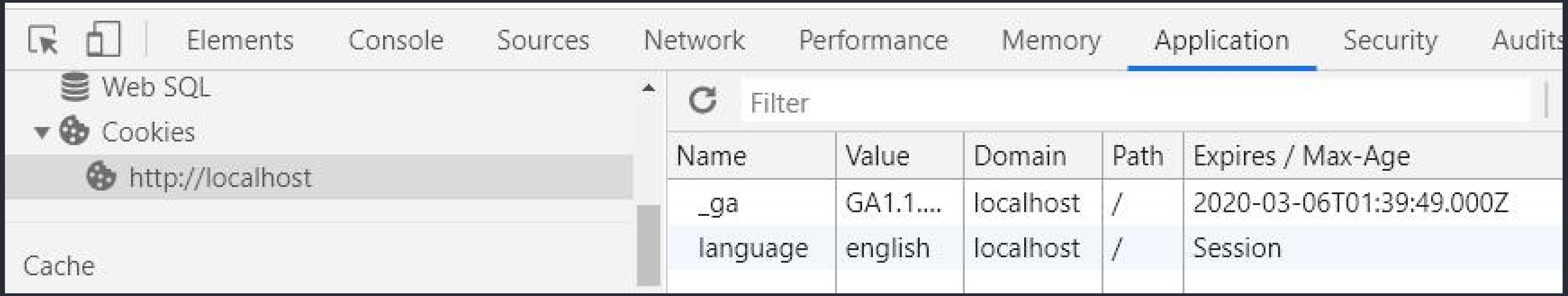

HTM
1) Cookie Enabled Page Request
•User access index.php
•index.php tracks user IP Addresses
php
2) Server Sets Cookie
•server stores the IP address in the cookie
HTM
3) Other Page Requests
•server returns cookie name and value on all page requests

Creating Cookies



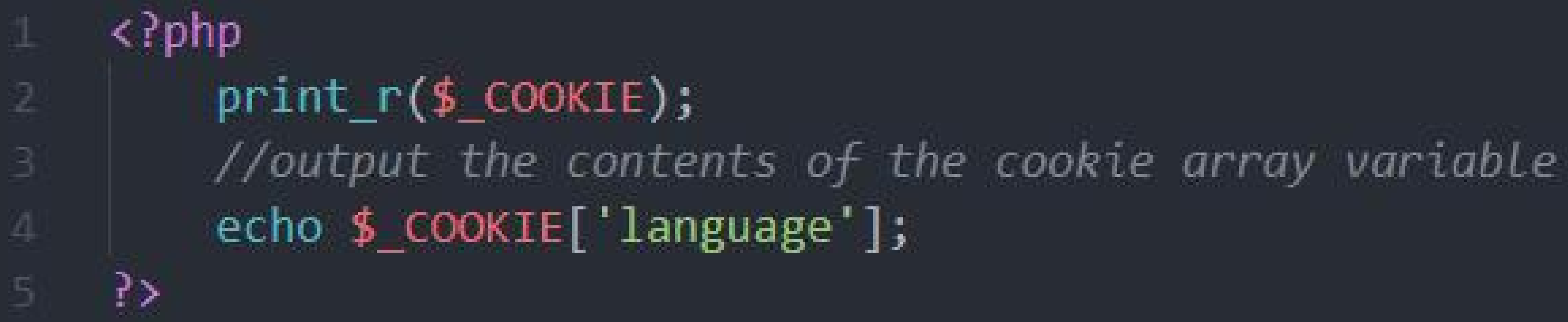
```php
<?php
    setcookie('language', 'english');
    echo 'the cookie has been set';
?>
```

```php
<?php
    setcookie("username", "admin", time()+ 60,'/');
    // expires after 60 seconds
    echo 'the cookie has been set for 60 seconds';
?>
```



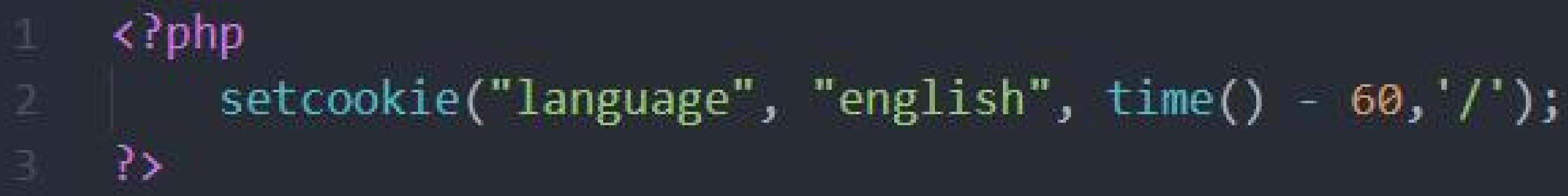
Name	Value	Domain	Path	Expires / Max-Age
_ga	GA1.1....	localhost	/	2020-03-06T01:39:49.000Z
language	english	localhost	/	Session

Read & Delete Cookies



```php
<?php
    print_r($_COOKIE);
    //output the contents of the cookie array variable
    echo $_COOKIE['language'];
?>
```

Set the expiry time to a time that has already passed to delete



```php
<?php
    setcookie("language", "english", time() - 60,'/');
?>
```

Cookies (All Params)

```
setcookie(name, value, expire, path, domain, security);
```

Here is the detail of all the arguments −

- **Name** − This sets the name of the cookie and is stored in an environment variable called HTTP_COOKIE_VARS. This variable is used while accessing cookies.
- **Value** − This sets the value of the named variable and is the content that you actually want to store.
- **Expiry** − This specify a future time in seconds since 00:00:00 GMT on 1st Jan 1970. After this time cookie will become inaccessible. If this parameter is not set then cookie will automatically expire when the Web Browser is closed.
- **Path** − This specifies the directories for which the cookie is valid. A single forward slash character permits the cookie to be valid for all directories.
- **Domain** − This can be used to specify the domain name in very large domains and must contain at least two periods to be valid. All cookies are only valid for the host and domain which created them.
- **Security** − This can be set to 1 to specify that the cookie should only be sent by secure transmission using HTTPS otherwise set to 0 which mean cookie can be sent by regular HTTP.

Securing Cookies

```php
<?php

    function signing_checksum($string) {
        $salt = "r4j4H03t4n"; // makes process hard to guess
        return hash('sha1', $string . $salt);
    }

    function sign_string($string) {
        return $string . '--' . signing_checksum($string);
    }

    function signed_string_is_valid($signed_string) {
        $array = explode('--', $signed_string);
        // if not 2 parts it is malformed or not signed
        if(count($array) != 2) { return false; }

        $new_checksum = signing_checksum($array[0]);
        return ($new_checksum === $array[1]);
    }

    setcookie("language", sign_string("english"));
    echo "cookie has been set";

?>
```

```php
echo signed_string_is_valid($_COOKIE['language']);
```

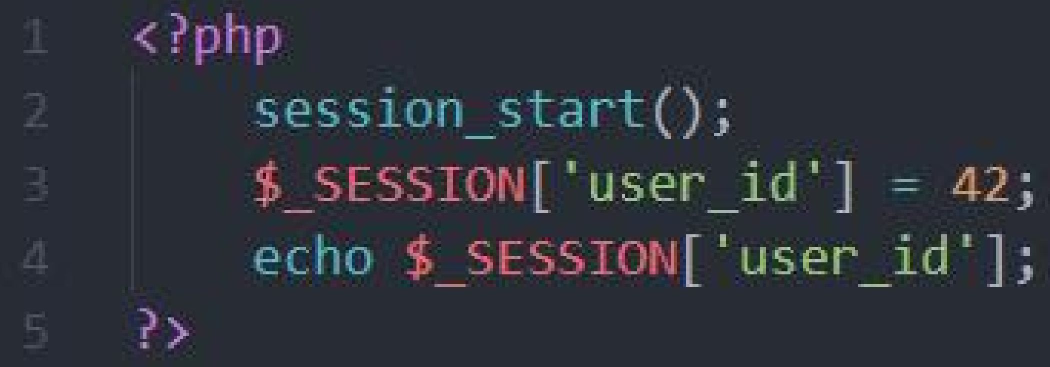
Session

- A session is a global variable stored on the server.
- Each session is assigned a unique id which is used to retrieve stored values.
- Whenever a session is created, a cookie containing the unique session id is stored on the user's computer and returned with every request to the server. If the client browser does not support cookies, the unique php session id is displayed in the URL
- Sessions have the capacity to store relatively large data compared to cookies.
- The session values are automatically deleted when the browser is closed. If you want to store the values permanently, then you should store them in the database.

Session

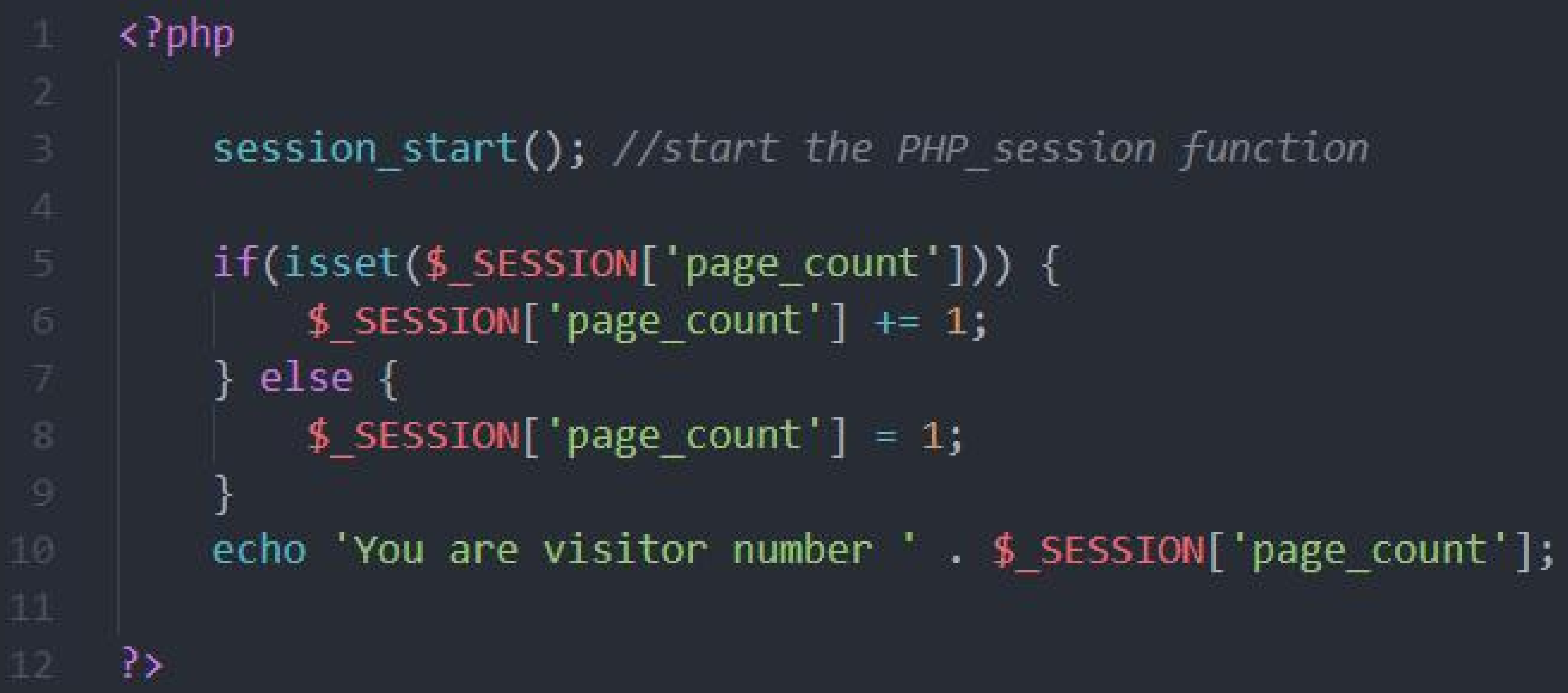
- Just like the $_COOKIE array variable, session variables are stored in the $_SESSION array variable.
- Just like cookies, the session must be started before any HTML tags.
- Store important information such as the user id more securely on the server where malicious users cannot temper with them.
- Alternative to cookies on browsers that do not support cookies.
- Store global variables in an efficient and more secure way compared to passing them in the URL
- Good for developing an application such as a shopping cart that has to temporary store information with a capacity larger than 4KB.

Creating Session



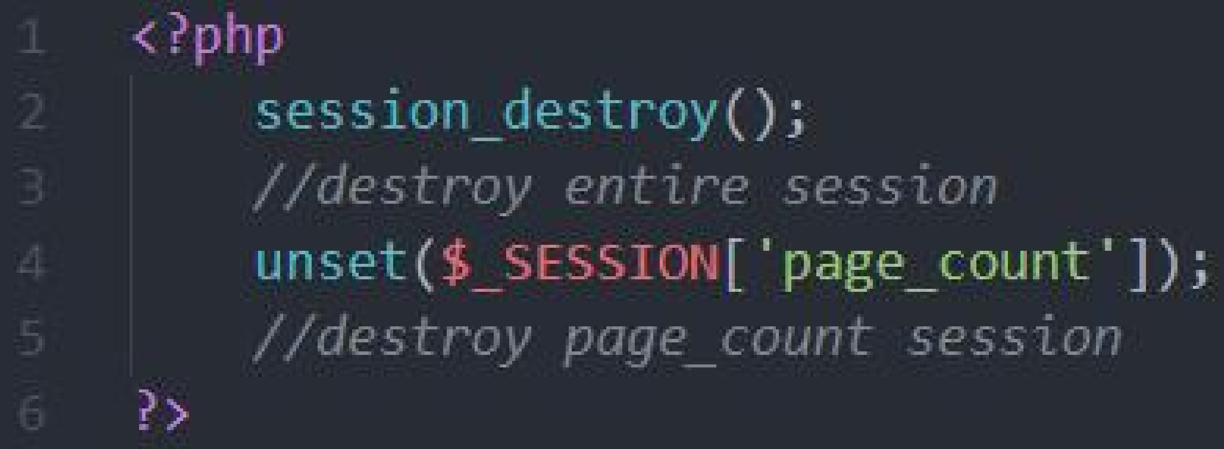
```php
<?php
    session_start();
    $_SESSION['user_id'] = 42;
    echo $_SESSION['user_id'];
?>
```

Creating Session



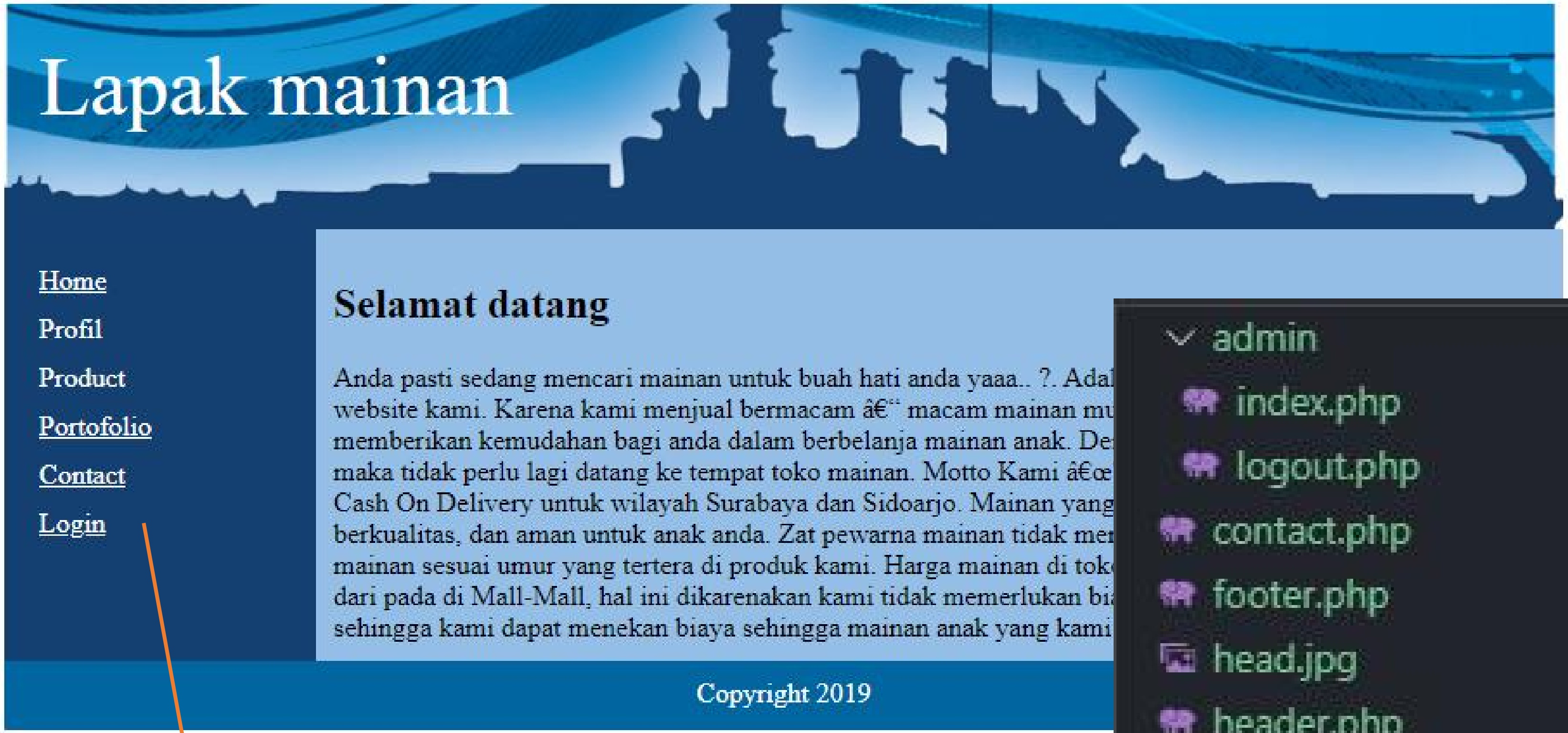
```php
<?php

    session_start(); //start the PHP_session function

    if(isset($_SESSION['page_count'])) {
        $_SESSION['page_count'] += 1;
    } else {
        $_SESSION['page_count'] = 1;
    }
    echo 'You are visitor number ' . $_SESSION['page_count'];

?>
```

Destroying Session



```php
<?php
    session_destroy();
    //destroy entire session
    unset($_SESSION['page_count']);
    //destroy page_count session
?>
```

Login Form Using Session



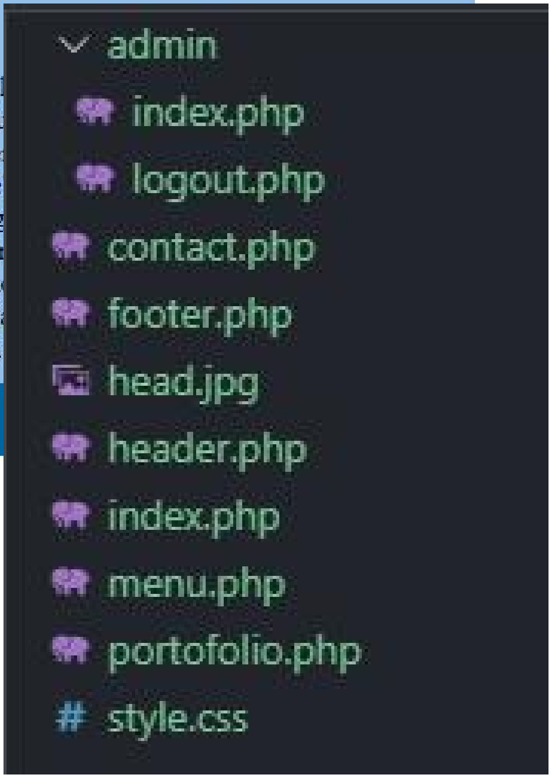
Lapak mainan

Home
Profil
Product
Portofolio
Contact
Login

Selamat datang

Anda pasti sedang mencari mainan untuk buah hati anda yaaa.. ?. Adal
website kami. Karena kami menjual bermacam â€“ macam mainan mu
memberikan kemudahan bagi anda dalam berbelanja mainan anak. De
maka tidak perlu lagi datang ke tempat toko mainan. Motto Kami â€œ
Cash On Delivery untuk wilayah Surabaya dan Sidoarjo. Mainan yang
berkualitas, dan aman untuk anak anda. Zat pewarna mainan tidak me
mainan sesuai umur yang tertera di produk kami. Harga mainan di tok
dari pada di Mall-Mall, hal ini dikarenakan kami tidak memerlukan bi
sehingga kami dapat menekan biaya sehingga mainan anak yang kami

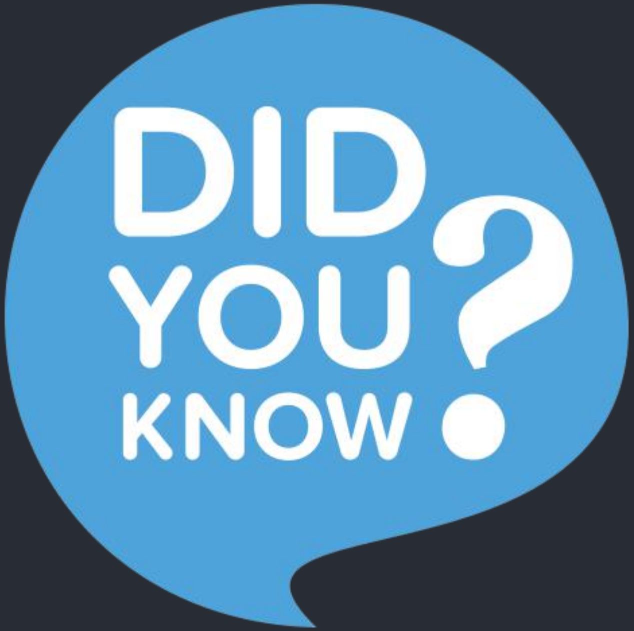
Copyright 2019



```
admin
    index.php
    logout.php
contact.php
footer.php
head.jpg
header.php
index.php
menu.php
portofolio.php
style.css
```

Make a login link to admin directory

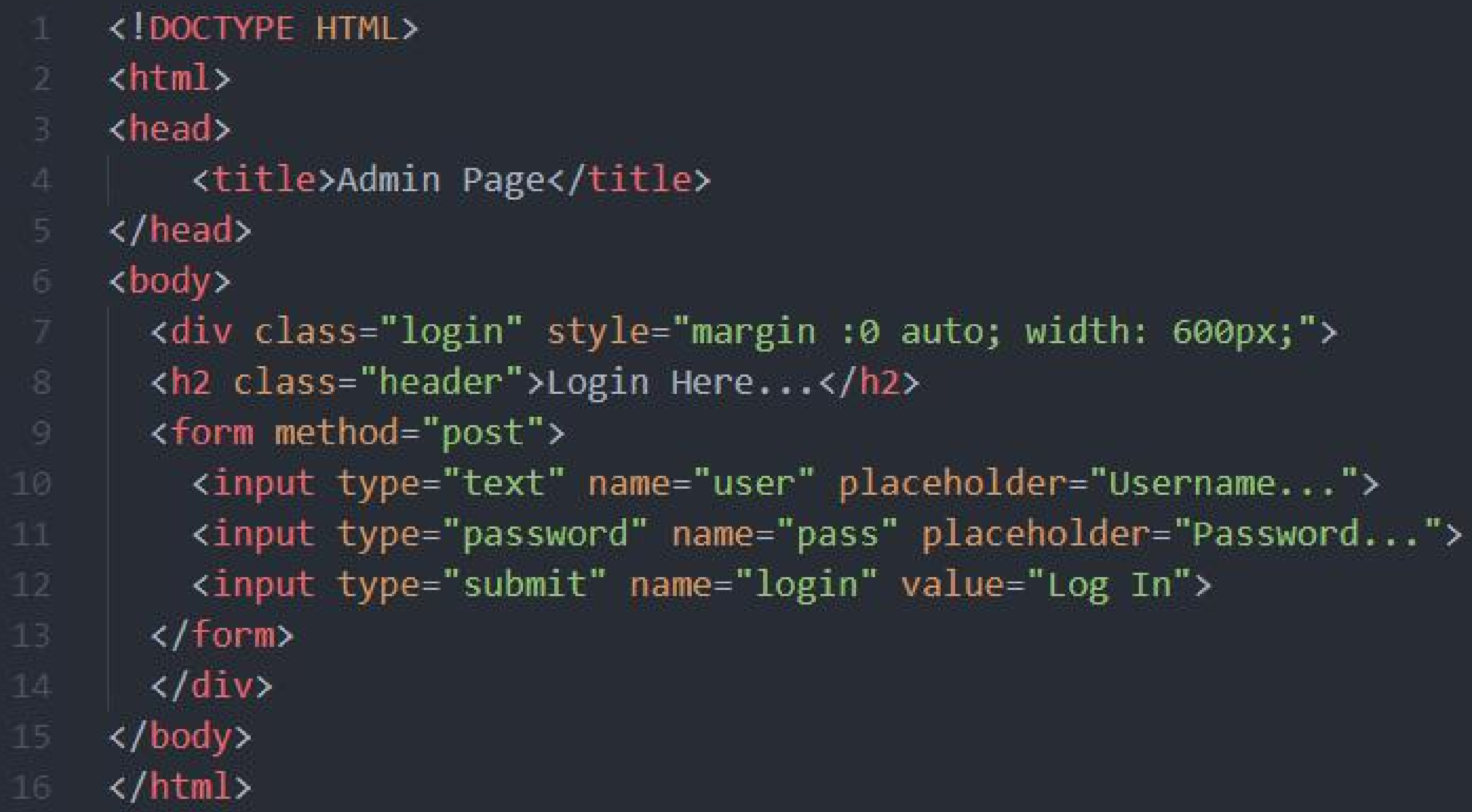
<a href="admin">Login</a>



The **index.php** is a PHP file that is the entry point of any website or application.

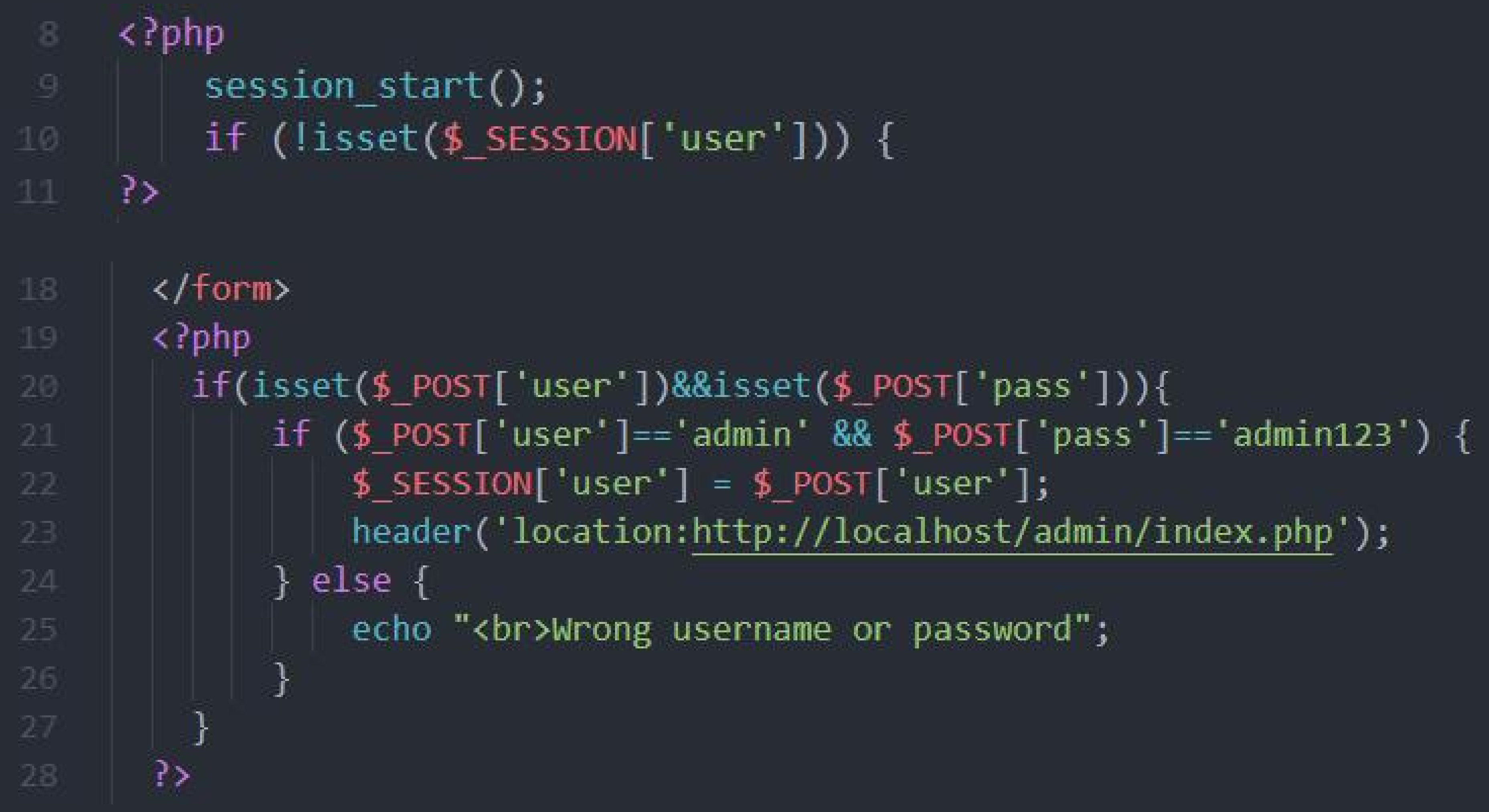
```
http://localhost/goodapp == http://localhost/goodapp/index.php
```

Login Form Using Session



```html
<!DOCTYPE HTML>
<html>
<head>
    <title>Admin Page</title>
</head>
<body>
  <div class="login" style="margin :0 auto; width: 600px;">
  <h2 class="header">Login Here...</h2>
  <form method="post">
    <input type="text" name="user" placeholder="Username...">
    <input type="password" name="pass" placeholder="Password...">
    <input type="submit" name="login" value="Log In">
  </form>
  </div>
</body>
</html>
```

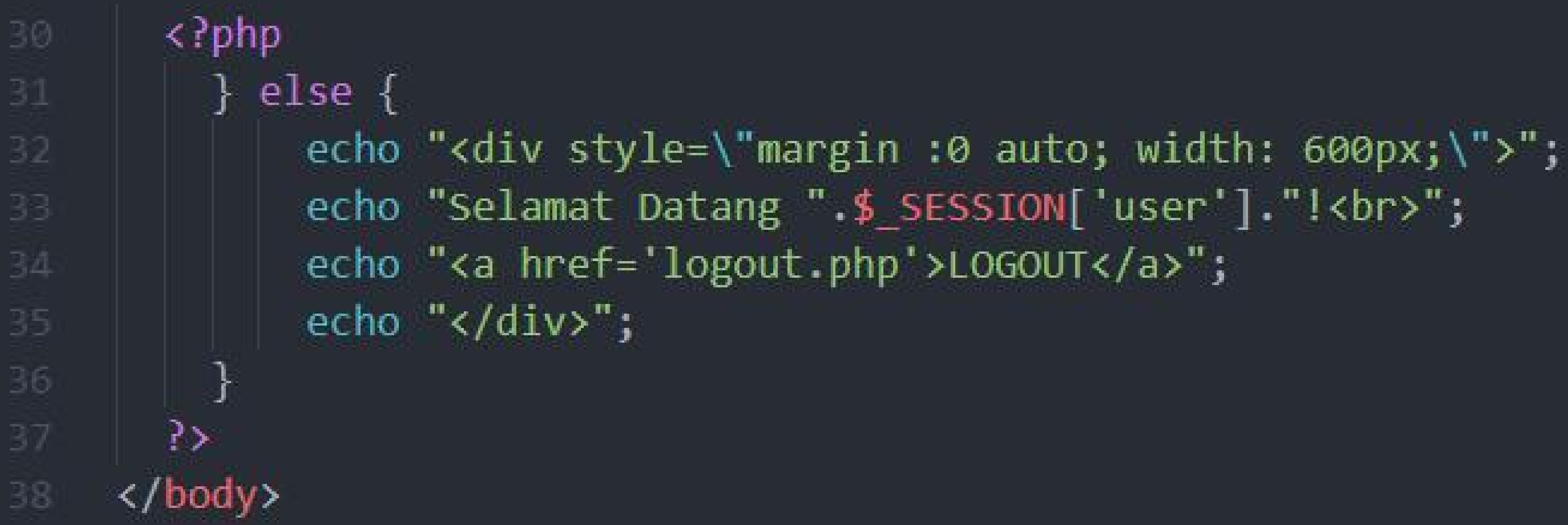
Login Form Using Session



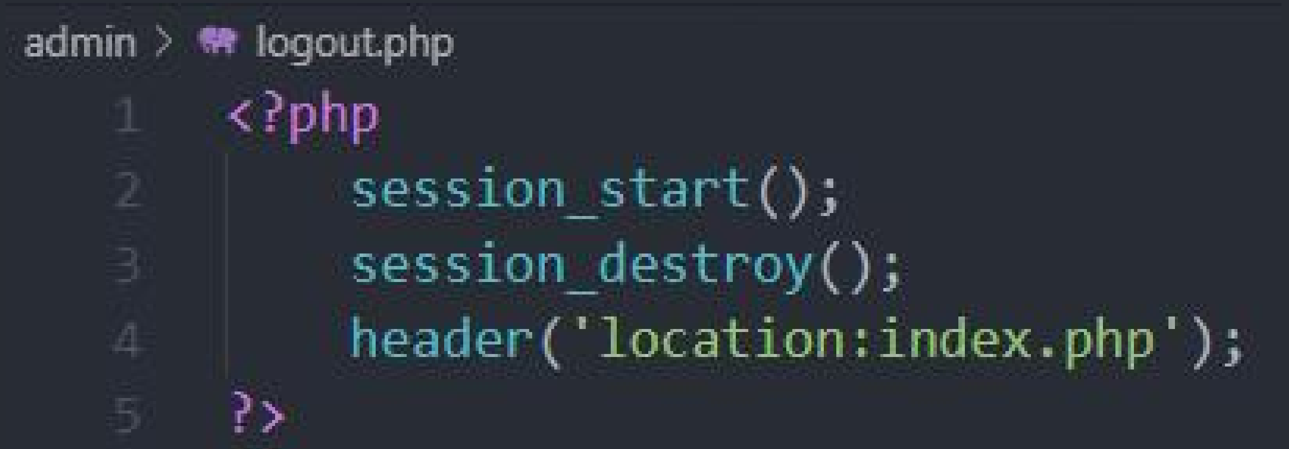
```php
<?php
        session_start();
        if (!isset($_SESSION['user'])) {
?>
```

```php
    </form>
    <?php
      if(isset($_POST['user'])&&isset($_POST['pass'])){
            if ($_POST['user']=='admin' && $_POST['pass']=='admin123') {
                $_SESSION['user'] = $_POST['user'];
                header('location:http://localhost/admin/index.php');
            } else {
                echo "<br>Wrong username or password";
            }
      }
    ?>
```

Login Form Using Session



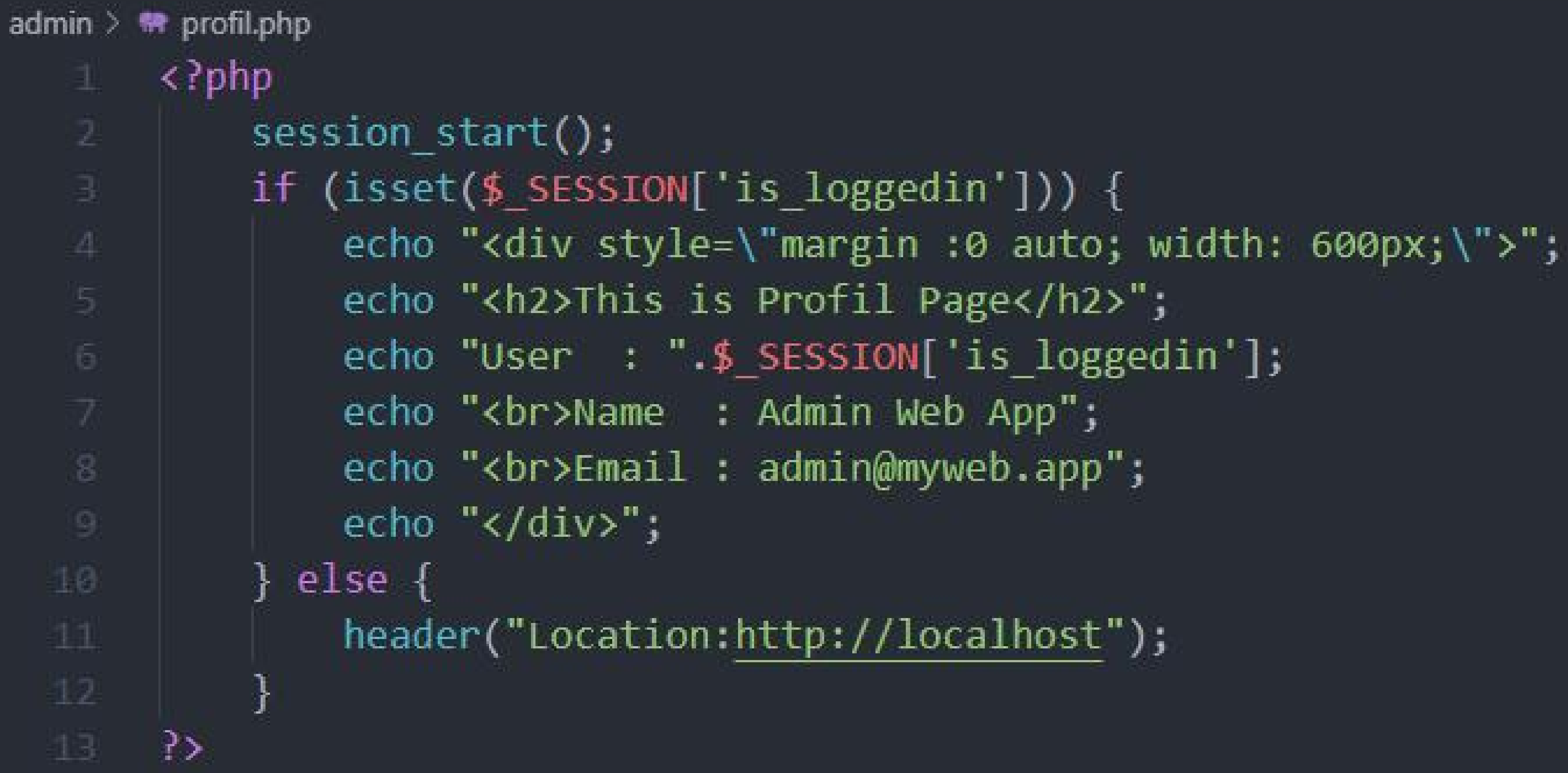
```
<?php
  } else {
    echo "<div style=\"margin :0 auto; width: 600px;\">";
    echo "Selamat Datang ".$_SESSION['user']."!<br>";
    echo "<a href='logout.php'>LOGOUT</a>";
    echo "</div>";
  }
?>
</body>
```



admin > logout.php

```
<?php
    session_start();
    session_destroy();
    header('location:index.php');
?>
```

Protect Page Using Session



admin > profil.php

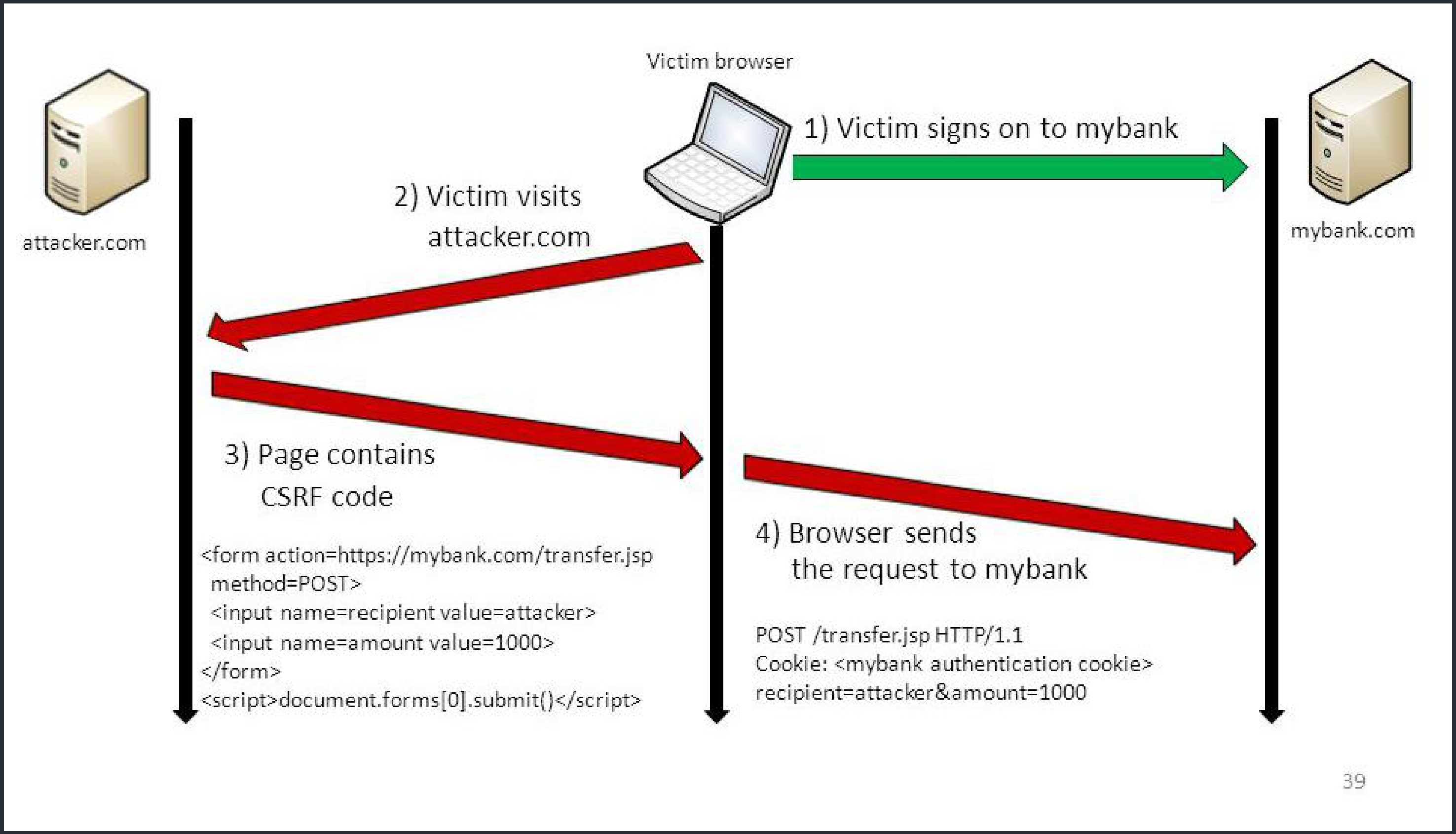
```php
<?php
    session_start();
    if (isset($_SESSION['is_loggedin'])) {
        echo "<div style=\"margin :0 auto; width: 600px;\">";
        echo "<h2>This is Profil Page</h2>";
        echo "User  : ".$_SESSION['is_loggedin'];
        echo "<br>Name  : Admin Web App";
        echo "<br>Email : admin@myweb.app";
        echo "</div>";
    } else {
        header("Location:http://localhost");
    }
?>
```

Prevent CSRF Attack

Cross Site Request Forgery

"Cross-Site": originates on one site but performs an action on another

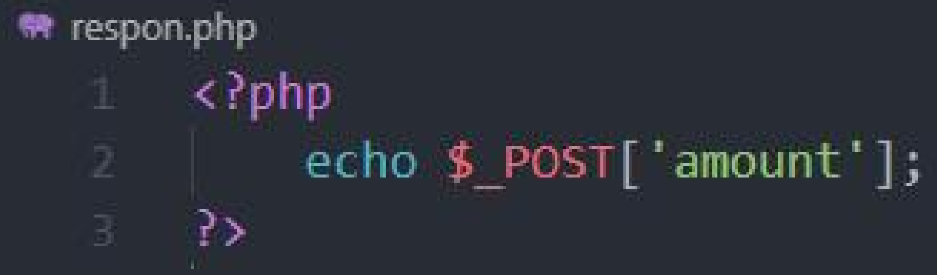
"Request Forgery": it is not a genuine user request


Victim browser
attacker.com
mybank.com
1) Victim signs on to mybank
2) Victim visits attacker.com
3) Page contains CSRF code
4) Browser sends the request to mybank

```
<form action=https://mybank.com/transfer.jsp
 method=POST>
 <input name=recipient value=attacker>
 <input name=amount value=1000>
</form>
<script>document.forms[0].submit()</script>
```

```
POST /transfer.jsp HTTP/1.1
Cookie: <mybank authentication cookie>
recipient=attacker&amount=1000
```

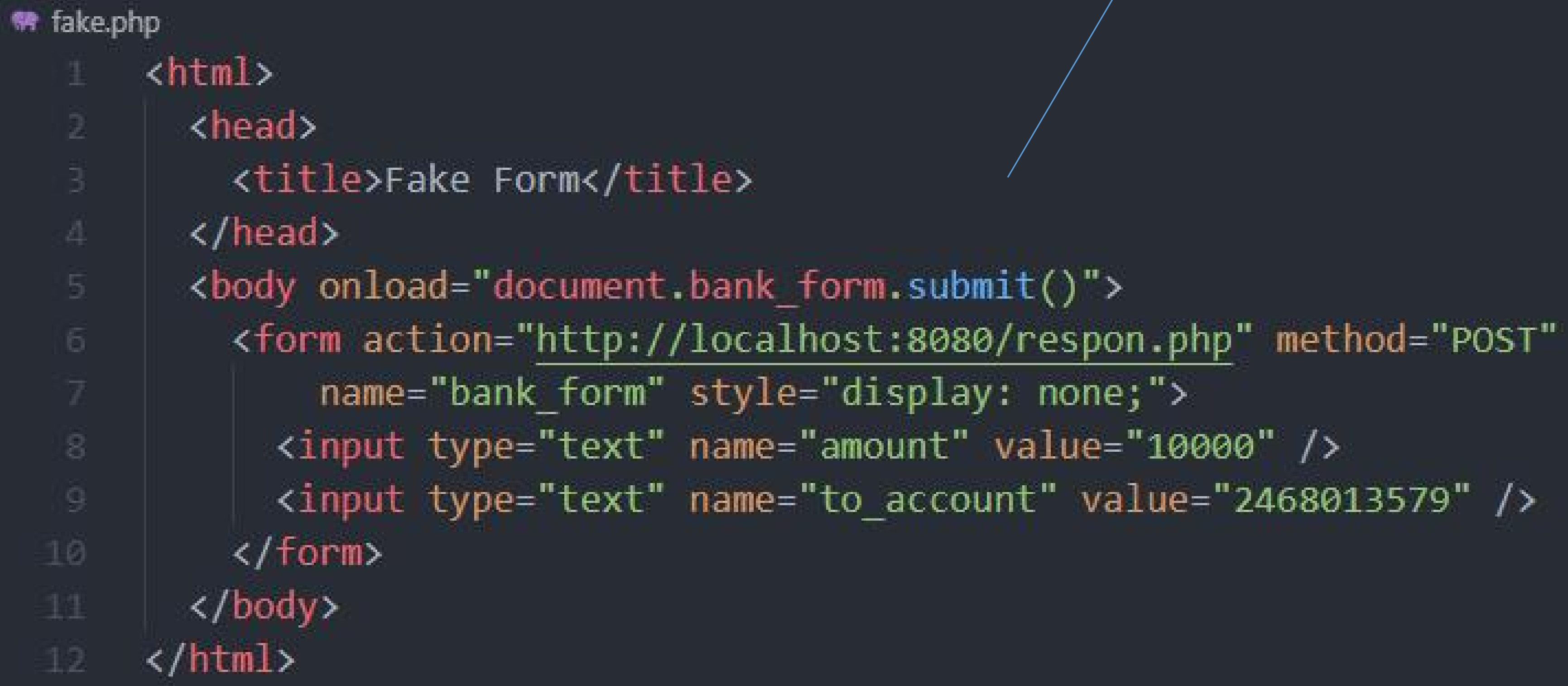
Prevent CSRF Attack



respon.php

```php
<?php
    echo $_POST['amount'];
?>
```

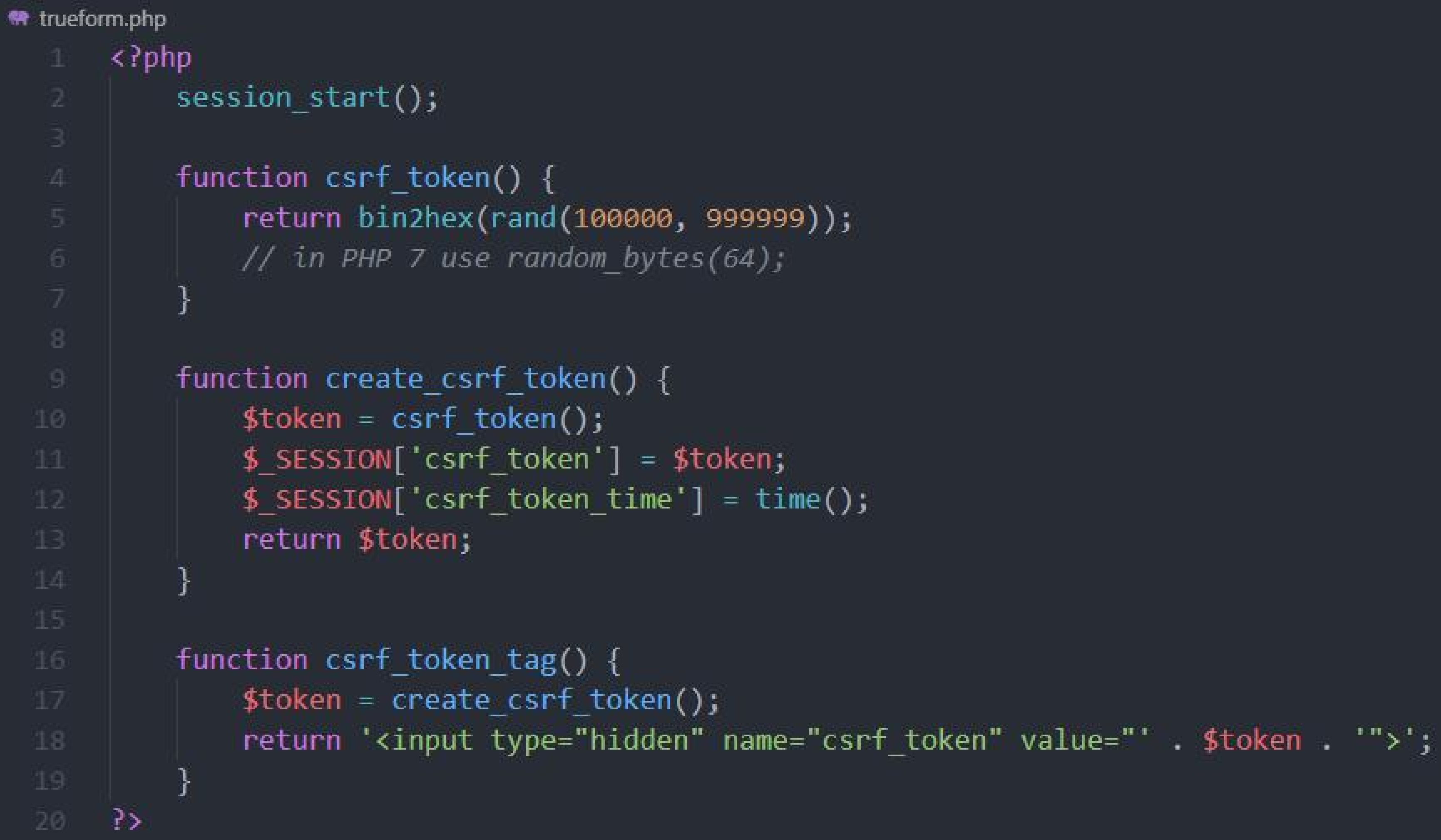
Different Server



fake.php

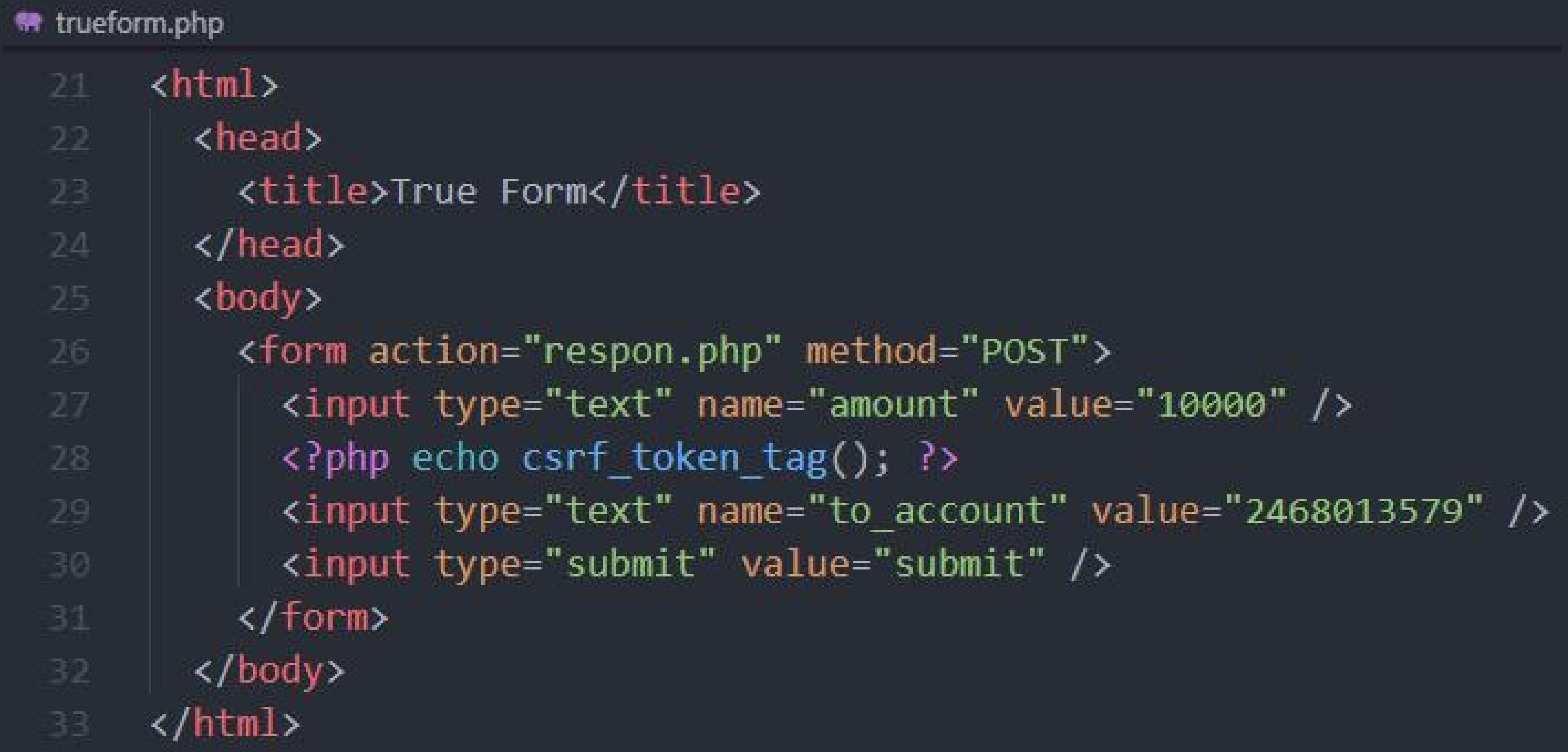
```html
<html>
  <head>
    <title>Fake Form</title>
  </head>
  <body onload="document.bank_form.submit()">
    <form action="http://localhost:8080/respon.php" method="POST"
        name="bank_form" style="display: none;">
      <input type="text" name="amount" value="10000" />
      <input type="text" name="to_account" value="2468013579" />
    </form>
  </body>
</html>
```

Prevent CSRF Attack


trueform.php

```php
<?php
    session_start();

    function csrf_token() {
        return bin2hex(rand(100000, 999999));
        // in PHP 7 use random_bytes(64);
    }

    function create_csrf_token() {
        $token = csrf_token();
        $_SESSION['csrf_token'] = $token;
        $_SESSION['csrf_token_time'] = time();
        return $token;
    }

    function csrf_token_tag() {
        $token = create_csrf_token();
        return '<input type="hidden" name="csrf_token" value="' . $token . '">';
    }
?>
```

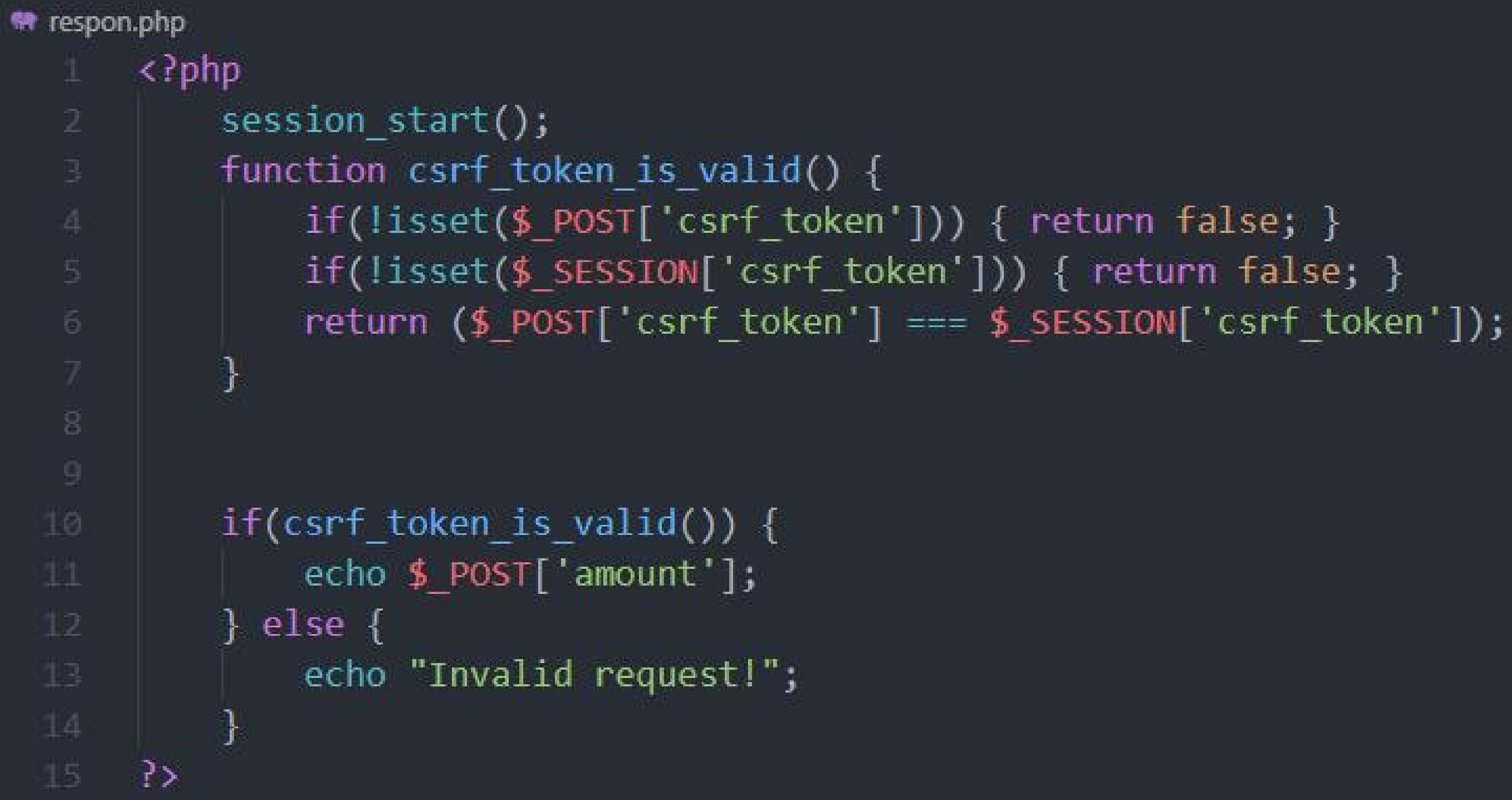
Prevent CSRF Attack



trueform.php

```
<html>
  <head>
    <title>True Form</title>
  </head>
  <body>
    <form action="respon.php" method="POST">
      <input type="text" name="amount" value="10000" />
      <?php echo csrf_token_tag(); ?>
      <input type="text" name="to_account" value="2468013579" />
      <input type="submit" value="submit" />
    </form>
  </body>
</html>
```

Prevent CSRF Attack



respon.php

```php
<?php
    session_start();
    function csrf_token_is_valid() {
        if(!isset($_POST['csrf_token'])) { return false; }
        if(!isset($_SESSION['csrf_token'])) { return false; }
        return ($_POST['csrf_token'] === $_SESSION['csrf_token']);
    }


    if(csrf_token_is_valid()) {
        echo $_POST['amount'];
    } else {
        echo "Invalid request!";
    }
?>
```