

Create your WEB Application with MVC paradigm and OPP – PAP support

Tutorial 9

In this tutorial we will insert some validation areas. All users when made the register automatically receive the role costumers. But for some projects it is important define more than one role.

But first we need to define a few things before introduce the admin validation parts.

First we need to check if our user is logged-in and if is not, send him to the login page.

So, to achieve this goal, please insert the following code:

```
function check_login($redirect = false, $allowed = array())
{
    if (!isset($_SESSION['user_url'])) {
        if ($redirect) {
            header("Location: " . ROOT . "login");
            exit;
        }
        return false;
    }

    $db = Database::getInstance();
    $url = $_SESSION['user_url'];

    $query = "SELECT * FROM users WHERE url_address = :url LIMIT 1";
    $result = $db->read($query, array('url' => $url));

    if (!$result) {
        if ($redirect) {
            header("Location: " . ROOT . "login");
            exit;
        }
        return false;
    }

    $user = $result[0];
    if (!empty($allowed) && !in_array($user->role, $allowed)) {
        if ($redirect) {
            header("Location: " . ROOT . "login");
            exit;
        }
        return false;
    }

    return $user;
}
```

Let's debug this code.

First Checks if the `$_SESSION['user_url']` variable is set, which typically stores the URL address of the logged-in user. If not set redirects the user to the login page, otherwise, it returns false and the user is log in.

Next, it creates a database instance and retrieves the URL address of the user from the session. After that, it checks the result. If the logged-in user has admin permissions, certain additional parts of the application are allowed to be visited. Otherwise, the user is restricted from accessing these parts.

Now let's test our function:

First you need to go to your home controller and put this code:

This will call the function that you have created to validate if the user is logged-in.

Next you will test if everything is ok, and made a see you name in front end of your application, and this is the proof that the user exists and is logged-in.

```
public function index()
{
    $User = $this->load_model('User');
    $user_data = $User->check_login(true);

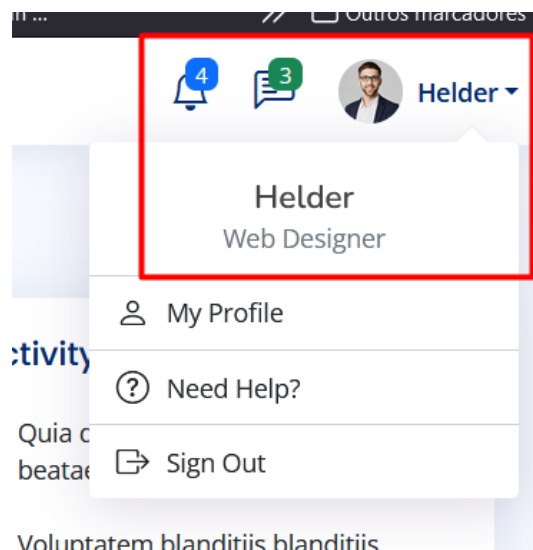
    //validate if the user is really log in
    if(is_object($user_data))
    {
        $data['user_data'] = $user_data;
        //show($data['user_data']);
    }

    //$this->title = 'Admin - Dashboard';
    $data['page_title'] = "Home - Dashboard";
    //Rota onde esta a view que vai carregar
    $this->view("index", $data);
}
```

Please put this line in a space, like the example:

```
</li><!-- End Messages Nav -->
<li class="nav-item dropdown pe-3">
    <a class="nav-link nav-profile d-flex align-items-center pe-0" href="#" data-bs-toggle="dropdown">
        
        <span class="d-none d-md-block dropdown-toggle ps-2"><?php echo $data['user_data']->name; ?></span>
    </a><!-- End Profile Image Icon -->
    <ul class="dropdown-menu dropdown-menu-end dropdown-menu-arrow profile">
        <li class="dropdown-header">
            <h6><?php echo $data['user_data']->name; ?></h6>
            <span>Web Designer</span>
        </li>
        <li>
            <hr class="dropdown-divider">
        </li>
    </ul>
</li>
```

And in front end you should see the name of the authenticated person:



Now you can call your fields database anywhere.

OK, now you want to logout your account.

So to do that, please create a new function in user.class.php, before the check_login() function.

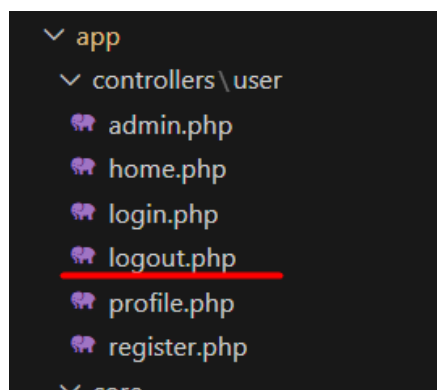
Insert this code:

```
//Function to logout
public function logout()
{
    if(isset($_SESSION['user_url']))
    {
        unset($_SESSION['user_url']);
    }

    header("Location: " . ROOT . "login");
    die;
}
```

In this code, you will do the unset for your user_url, and automatically when this function is call, the authenticated user pass to not loggedin.

Please create a new controller in your controller folder:

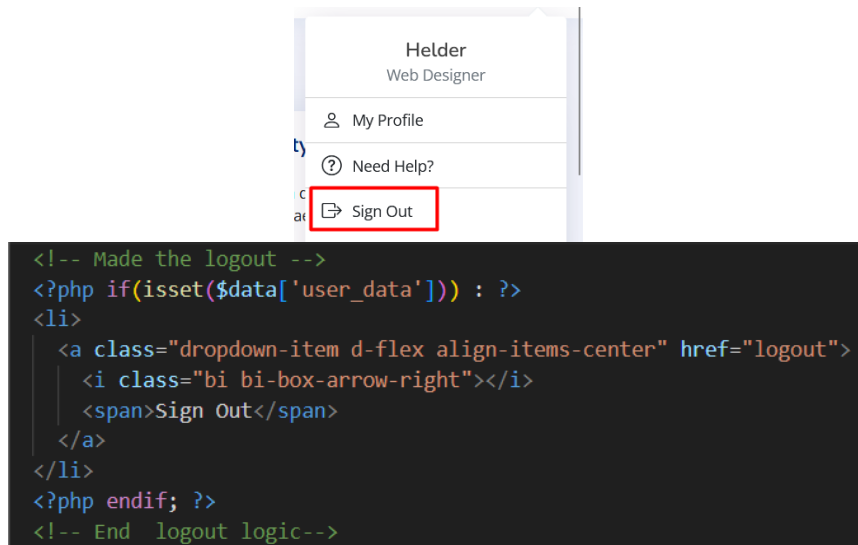


And insert this code inside:

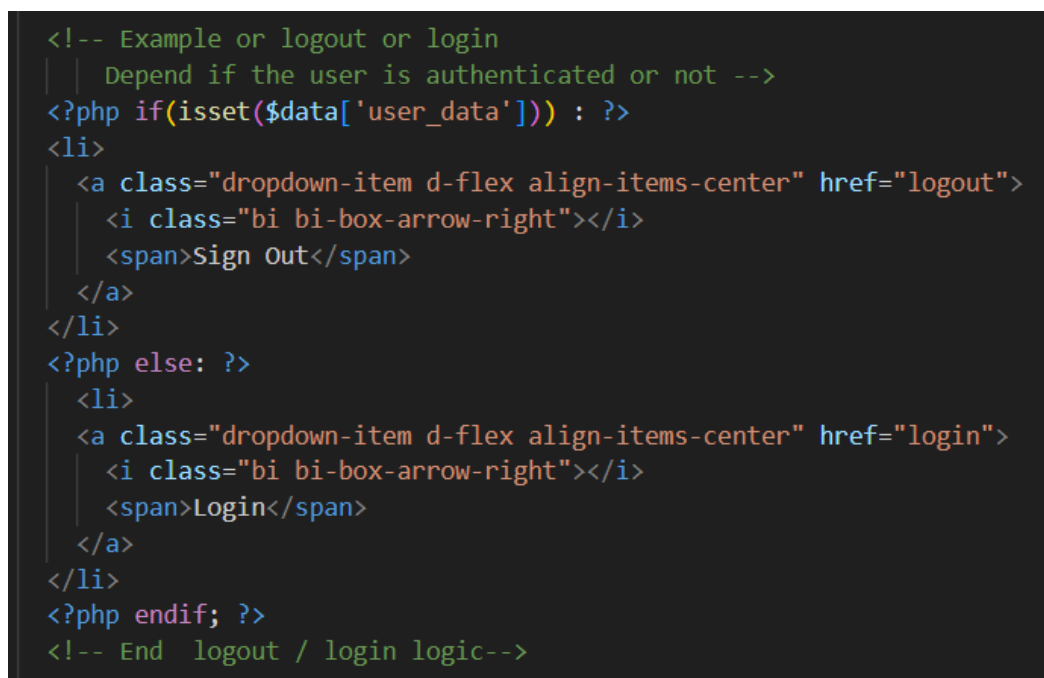
```
eshop > app > controllers > user > 🐘 logout.php
1  <?php
2
3  class Logout extends Controller
4  {
5      //Public default metodo index, mesmo que
6      public function index()
7      {
8          $User = $this->load_model("User");
9          $User->logout();
10
11      }
12 }
```

Next, you will adapt your header view to call this controller and do the unset user:

This example is if you have a separated button in your front end.



But imagine that your header is the same when you do the log in our log out, then you need to create a if statement to show one option, or another like this example:



With this example you can change buttons, if you are login, or logout. You will use the code that you need.

Next tutorial you will create a section for your administrator.

END Tutorial 9

Reference: **PHP Ecommerce website development | Create user login | MVC OOP - Quick programming**

Template: **NICEADMIN, design by BOOTSTRAP**