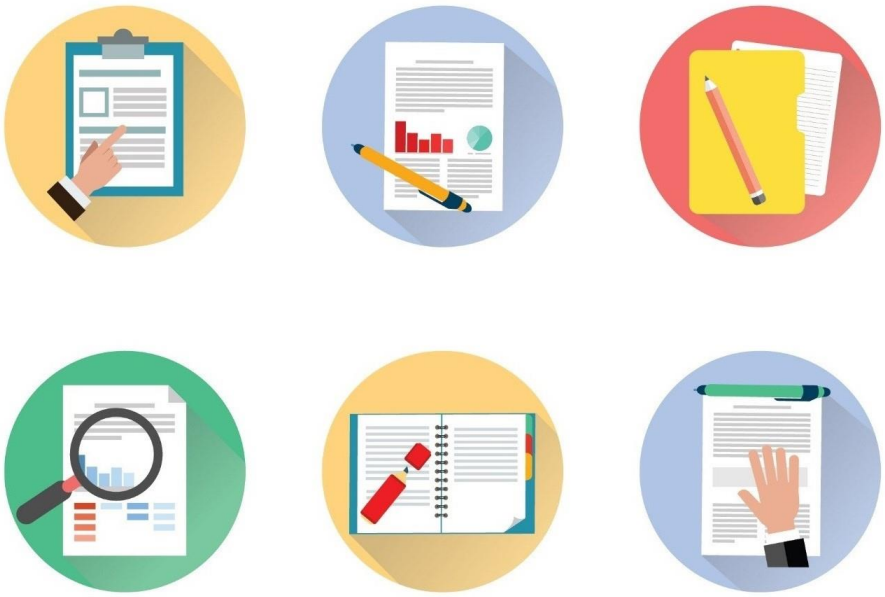
INTERNSHALA
VIRTUAL TRAINING CENTER



# Assignment: IV

**Problem Statement**

In this assignment, you will work on the pages you have created in bootstrap assignment. In bootstrap assignment, you have made HTML pages. To include php, you need to change the extension of web pages to php. So, **change .html to .php.** In this way, web pages will be converted into php.

- Create a new folder named **includes.**
- Create footer.php file. Keep the footer code inside the file. Footer is present in all the web pages. We will include this code inside all the web page using include statement.

```
<footer>
   <div class="container">
      <center>
         <p>Copyright &copy; Lifestyle Store. All Rights Reserved  |  Contact Us: +91 90000
00000</p>
      </center>
   </div>
</footer>
```

- Create header.php file inside includes folder. There are two bootstrap navbar we have used in bootstrap assignment. One is for logged in users, which includes cart, settings, and logout option. Second for logged out users or new users which includes signup and login links. Combine this in one header.

**Logic:**
Session variable will be declared after login or signup. I.e. we can use session to control which links will be shown to which user. We will create email and user_id session variables for logged in users. The code of header.php file will be:

```
<div class="navbar navbar-inverse navbar-fixed-top">
   <div class="container">
      <div class="navbar-header">
         <button type="button" class="navbar-toggle" data-toggle="collapse" data-
target="#myNavbar">
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
         </button>
         <a class="navbar-brand" href="index.php">Lifestyle Store</a>
      </div>
      <div class="collapse navbar-collapse" id="myNavbar">
         <ul class="nav navbar-nav navbar-right">
            <?php
            if (isset($_SESSION['email'])) {
               ?>
               <li><a href = "cart.php"><span class = "glyphicon glyphicon-shopping-
cart"></span> Cart </a></li>
               <li><a href = "settings.php"><span class = "glyphicon glyphicon-user"></span>
Settings</a></li>
               <li><a href = "logout_script.php"><span class = "glyphicon glyphicon-log-
in"></span> Logout</a></li>
               ?>
               <?php
```

```
            } else {
                ?>
                <li><a href="signup.php"><span class="glyphicon glyphicon-user"></span>
Sign Up</a></li>
                <li><a href="login.php"><span class="glyphicon glyphicon-log-in"></span>
Login</a></li>
                    <?php
                }
                ?>
            </ul>
        </div>
    </div>
</div>
```

The difference of links is in navbar-right. If **isset($_SESSION['email']** is true, then the user is logged in. So cart, settings and logout links will be shown. If session is not set, then signup and login links will be shown. As navbar is included in all the web pages, we will add this code in all web pages using include statement.

**Common.php**

Create common.php file inside includes folder which will contain connection variable and session_start() function. We will use require statement to add this file code into all web pages.

**Index.php**

- Use require statement and add common.php file in the first line of the code.
- Index page can be visited by logged out users only. Add the condition that if session variable email is set, redirect the user to products.php page using header function. Code will be like:

```
if (isset($_SESSION['email'])) {
  header('location: products.php');
}
```

- Inside the body tag, remove navbar code. Start the php tag and include header.php file and then close the php tag.
- Remove the footer code. Include footer.php.

**Login.php**

- Add common.php file.
- Login page can be visited only by logged out users.
- To login form, add method POST and action login_submit.php page.
- Include footer.php in the end.

**Login_submit.php**

- Add common.php file.
- Store the login form data into variables.
- Use mysqli_real_escape_string function for security.
- Use MD5 function for password value.
- Write the select query to fetch id and email from the users where email and password are the values entered by the user in the login form.
- If mysqli_num_rows == 0, then there is no user with the email and password in the users table.

- Else use mysqli_fetch_array.
- Initialize email and user_id session variables.
- User header function to redirect the user to products.php page.

**Signup Page**

- Add common.php file.
- This page will not be visited by logged in users i.e. if the session is set, redirect the user to products.php page.
- Remove header and footer code, include php files for the same.
- Use HTML5 validations for the form fields.
- Form data will be send to signup_script.php page using POST method.

**Signup_script.php**

- Add common.php file.
- Use backend validations.
- Use the select query to fetch id of the user whose email is the email entered by the user in the signup form. This means we do not allow duplicate entry of emails.
- If mysqli_num_rows > 0, show error that email id already exists.
- Else write the insert query to insert the new user.
- After mysqli_query function, use mysqli_insert_id($con) function to get the primary key of the new user inserted.
- Initialize session variable as we have done after succesful login.
- Redirect the user to products.php page using header function.

**Check-if-added.php inside includes folder**

- Make a function check_if_added_to_cart($item_id). This function will get user_id from the session.
- Include common.php file inside the function.
- This function will take user id and items id and will check whether there is an entry in users_products table with status 'Added to cart'.

```
SELECT * FROM user_items WHERE item_id='$item_id' AND user_id ='$user_id' and status='Added to cart'
```

- This means the user has already added this product to the cart.
- If mysqli_num_rows >= 1, then return 1. Else return 0.
- This web page will just contain this function.
- This function will be used in products.php page.

**Products Page:**

- Add common.php file.
- Remove navbar code. Include header.php code.
- Include Check-if-added.php page also after including header.php page.
- Now we will change the buy now button code for every item.
- Let us start with the first item.
- Start the PHP tag after paragraph tag which include price for Cannon EOS.
- For logged out users, we will show button with Add to cart value with href="login.php".
- For logged in users, button will be disabled if a particular item is already added to cart by the user. Else show the link to add the item to the cart.
- Else, use another if else condition inside it.

- If condition is, call check_if_added_to_cart(1). For this item, item id is 1. If this function returns 1, add a button which is disabled and contains text Added to cart. Else show the button with href href="cart-add.php?id=1" and text Add to cart.
- We have hardcoded the link. Href attribute will contain cart-add.php?id=1. This id=1 is for first item. For second item id=2 and so on.
- The final button code will be

```
<?php if (!isset($_SESSION['email'])) { ?>
                    <p><a href="login.php" role="button" class="btn btn-primary btn-block">Buy
Now</a></p>
                    <?php
                } else {
                    //We have created a function to check whether this particular product is added
to cart or not.
                    if (check_if_added_to_cart(1)) { //This is same as if(check_if_added_to_cart !=
0)
                        echo '<a href="#" class="btn btn-block btn-success" disabled>Added to
cart</a>';
                    } else {
                        ?>
                        <a href="cart-add.php?id=1" name="add" value="add" class="btn btn-block
btn-primary">Add to cart</a>
                        <?php
                    }
                }
                ?>
```

- This code will be same for every item except the item id's.

**Cart-add.php**

- This page will receive one id in GET method.
- Add common.php file.
- Write the insert query to add items id (from url) and user id from the session and status would be Added to cart.

```
INSERT INTO user_items(user_id, item_id, status) VALUES('$user_id', '$item_id', 'Added to cart')
```

- After mysqli_query function, use header function to redirect the user to products.php page.

**Cart page:**

- Add common.php.
- Add validation, only logged in users can visit this page.
- Use header.php and footer.php
- Get user id from the session.
- Select all the items from users_products of particular user. Use inner join to get the product details of the items added to cart by the user.
- If mysqli_num_rows ==0, show message that "Add items to the cart first!".
- In else part, use while loop.
- In while loop, initialize a variable sum inside the loop and add the price of the items when the loop iterates.
- Initialize id variable inside the loop and append all id's of the items separated by comma.
- Display all the items added to cart by the user.

- Add a link in front of each item <a href='cart-remove.php?id={$row['id']}' class='remove_item_link'> Remove</a>
- Inside single quotes, use {} symbol around the variable to make them count as a variable.
- Last row will be out of the while loop which will contain total and a link to success page with id variable initialized will be passed as GET function.

**Cart-remove.php**

- Add common.php file.
- In URL, use the id and user id is in session.
- Use delete query to delete the row with user id and items id.
- Redirect the user to cart.php page after successful deletion.

**Success.php**

- Add common.php file.
- If email session is not set, redirect the user to index.php page.
- For all item id's present in url, change the status to confirmed including user id in where clause.
- Display success message: Your order is confirmed. Thank you for shopping with us. Click here to purchase any other item.
- Click here in the success message will be a link to products.php page.

**Settings.php**

- Add common.php file.
- If email session is not set, redirect the user to index.php page.
- Make a form with action settings_script.php and method POST. Form will have the fields: old password, new password, retype new password.

**Settings_script.php**

- Add common.php file.
- If email session is not set, redirect the user to index.php page.
- Add backend validations. Hint: you can check with strlen function that whether the length of new password and retype new password fields is same. If not, show an error.
- Use select query to fetch the password stored in the database.
- If password matches, write the update query.
- If not, redirect the user to settings.php page with the error message.

**Logout.php page**

- Start session.
- If email session is not set, redirect the user to index.php page.
- Use Session_destroy function.
- Redirect the user to index.php page.