**POS system in raw php**

**# Includes Folder**

**-> header.php**

(contain header **<html> to <body>** portion , keep bootstrap cdn and put **navbar.php** link after **starting <body> tag**)

**-> footer.php** ( keep </body> to </html>) and add jQuery cdn and js npm bootstrap link

**-> navbar.php** (opening nav bar in home page or 1st index.php file,)

which contain **login and logout button** *if login* then it will show **user name** with **logout button**)

-> **index.php** (include **header & footer file** from **includes folder**)

In body it contain l**ogin and registration link**

-> **registration.php (form)** && **registration-code.php(database)**

form(action=”registration-code.php” name, email, password, phone,)

Database (emailCheck(), password\_hash($password, PASSWORD\_BCRYPT), redirect('login.php’), redirect(‘registration.php’) )

-> **login.php(form) && login-code.php(database)**

form(action=”login-code.php” email, password)

database(emailCheck(), passwordCheck(),

Maintain session for login : (user information come from **emailCheckFunction()**)

$\_SESSION['loggedIn'] = true;

$\_SESSION['loggedInUser'] = [

'user\_id' => $emailCheck['data']['id'],

'name' => $emailCheck['data']['name'],

'email' => $emailCheck['data']['email'],

'phone' => $emailCheck['data']['phone'],

];

)

**redirect() :** a customize function by me, take 2 parameters url and a message. Assign message in in a $\_SESSION[‘status’] and url pass in header() function which is a php built in function.

// redirect from 1 page to another page with the message(status)

function redirect( $url, $status){

$\_SESSION['status'] = $status;

header('Location: ' . $url);

exit(0);

}

**$\_SESSION : use for login and logout purpose , and show messages purpose**

in login time it contain login situation like :

$\_SESSION['loggedIn'] = true;

And user Information : (in the time of email check, according to email we find user data or info)

$\_SESSION['loggedInUser'] = [

'user\_id' => $emailCheck['data']['id'],

'name' => $emailCheck['data']['name'],

'email' => $emailCheck['data']['email'],

'phone' => $emailCheck['data']['phone'],

]

allertMessage(): using **$\_SESSION['status']** && **$\_SESSION[exception]**

// Display messages or status after any process

function alertMessage(){

if(isset($\_SESSION['status'])){

echo '<div class="alert alert-warning alert-dismissible fade show" role="alert">

<h6>'.$\_SESSION['status'].'</h6>

<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

</div>';

unset($\_SESSION['status']);

}

if (isset($\_SESSION['exception'])) {

echo '<div class="alert alert-danger alert-dismissible fade show" role="alert">

<h6>' . $\_SESSION['exception'] . '</h6>

<button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>

</div>';

unset($\_SESSION['exception']);

}

}

**-> authentication.php :**

**# Admin**

**Admin CRUD :** (for registration(alternative of registration.php))

**Functions :**

**getAll()** : to fetch all data from a table ( $admins = getAll('admins');)  
 **Check rows**: **if(mysqli\_num\_rows($admins)>0)** **Show data :**

<?php foreach($admins as $adminItem): ?>

<p>ID: <?= $adminItem['id']; ?></p>

<p>Name: <?= $adminItem['name']; ?></p>

<p>Email: <?= $adminItem['email']; ?></p>

<a href="admins-edit.php?id=<?= $adminItem['id']; ?>">Edit</a>

<a href="admins-delete.php?id=<?= $adminItem['id']; ?>">Delete</a>

<?php endforeach; ?>

**EDIT :**   
 Form (**action=code.php**, **method=’POST’** name, email, phone)  
**Get edit id from Url :**

**if(isset($\_GET['id'])){**

**$admin\_id = $\_GET['id'];**

**}else{**

**echo '<h5> NO ID given in params!</h5>';**

**return false;**

**}**

Fetch info from database table according to ID and Table :

**$adminData = getById('admins',$admin\_id);**

**getById() :**

function getById($tableName, $id) {

global $conn;

$query = "SELECT \* FROM $table WHERE id = '$id' LIMIT 1";

mysqli\_query($conn, $query);

if (mysqli\_num\_rows($result) == 1){

$row = mysqli\_fetch\_array($result, MYSQLI\_ASSOC);

$response = [

'status' => 200,

'data' => $row,

'message' => 'Record found'

];

return $response;

}

}

**Show value in edit form :**

<form action="code.php" method = 'POST'>

<?php

if(isset($\_GET['id'])){

$admin\_id = $\_GET['id'];

}else{

echo '<h5> NO ID given in params!</h5>';

return false;

}

$adminData = getById('admins',$admin\_id);

if($adminData){

if($adminData['status'] == 200){

?>

<input type="hidden" name = "adminId" value = "<?= $adminData['data']['id']?>">

<input type="text" name = "name" required value = "<?= $adminData['data']['name']?>" >

<input type="email" name = "email" required value = "<?= $adminData['data']['email']?>">

<button type="submit" name = "updateAdmin" >Update</button>

<?php

}

}

?>

</form>

Delete :

Get Delete id from Url :   
Fetch data according to Id from specific table  
Delete the data that fetched :

$paraResultId = checkParamId('id');

if(is\_numeric($paraResultId)){

$adminId = validate($paraResultId);

$admin = getById('admins', $adminId);

if($admin['status'] == 200){

$adminDeleteRes = delete('admins', $adminId);

if($adminDeleteRes){

redirect('admins.php');

}else{

redirect('admins.php');

}

}else{

redirect('admins.php');

}

}

function normalDelete($tableName, $id){

global $conn;

$table = validate($tableName);

$id = validate($id);

// Delete the product from the table

$query = "DELETE FROM $table WHERE id = '$id' LIMIT 1";

$result = mysqli\_query($conn, $query);

if (!$result) {

return false;

}

return true;

}

**checkParamId() :**

function checkParamId($type){

if(isset($\_GET[$type])){

if($\_GET[$type] != ''){

return $\_GET[$type];

}else{

return '<h5>No ID Found</h5>';

}

}else{

return '<h5>No ID Given</h5>';

}

}

Categories CRUD : form(action=’code.php’ , name, description, status, )

Product CRUD : show pictures from photos table according to every product

Order CRUD :

Order View :

Order view Print:

Customer CRUD : form(action=’code.php’ , name, email,phone, status, )

Show pic tures:

<!-----------------Showing Image--------------------->

<td>

<?php

$photos = getPicturesAccordingToIds($itme['id']);

// Check if the query was successful

if (!$photos) {

echo "<h4>Something went wrong.</h4>";

} else {

// Check if there are any photos

if (mysqli\_num\_rows($photos) > 0): ?>

<?php while ($photo = mysqli\_fetch\_assoc($photos)): ?>

<img src="../assets/uploads/products/<?= $photo['file']; ?>"

style="height:50px; width:50px" alt="img">

<?php endwhile; ?>

<?php else: ?>

<h4>No photos found.</h4>

<?php endif;

}

?>

</td>

<!---------------------END Showing Image------------------>

function getPicturesAccordingToIds($id){

global $conn;

$id = validate($id);

$query = "SELECT \* FROM photos WHERE product\_id = $id ";

$result = mysqli\_query($conn, $query);

if (!$result) {

die("Database query failed: " . mysqli\_error($conn));

}

return $result;

}

Store product multiple picture :

<form action="code.php" method = 'POST' enctype="multipart/form-data">

<div class="col-md-4 mb-3">

<label for="">Image \*</label>

<input type="file" name = "images[]" class = "form-control" multiple>

</div>

After insert a product then store images :

$result = insertWithImage('products', $data, $images, 'products' );

$lastInsertId = mysqli\_insert\_id($conn);

if (!empty($images)) {

$imageData = [

'images' => $images,

'product\_id' => $lastInsertId,

'product\_name' => $data['name'], // Assuming 'name' exists in $data

'directory' => $directory,

];

$imageInsertionResult = multipleImageInsert($imageData);

if (!$imageInsertionResult) {

throw new Exception("Failed to insert images.");

}

}

function multipleImageInsert($data){

$allImagesInserted = true;

for( $i=0; $i<count($data['images']['name']); $i++){

$image = [

'image\_name' => $data['images']['name'][$i],

'tmp\_name' => $data['images']['tmp\_name'][$i],

];

// debug(" debug from multipleImageInsert :".$data['images']['name'][$i] ." and ".$data['images']['tmp\_name'][$i] );

$image\_path = imageInsert($image, $data['directory'], $data['product\_name'], $i);

if($image\_path){ // last change

$imageData = [

'product\_id' => $data['product\_id'],

'file' => $image\_path,

// 'created\_at' => date("Y-m-d H:i:s")

];

$result = insert('photos', $imageData);

if (!$result) {

$allImagesInserted = false;

}

}

}

return $allImagesInserted;

}

function imageInsert($image, $directory, $name, $i){

$i += 1;

// debug("image : ".$image['name']);

$path = '../assets/uploads/'.$directory;

$image\_extension = pathinfo($image['image\_name'], PATHINFO\_EXTENSION);

if(!$image\_extension){

return false;

}

$unique\_string = md5(time());

$filename = $i.'-'.$unique\_string.'.'.$image\_extension;

move\_uploaded\_file($image['tmp\_name'], $path."/".$filename);

$final\_image = "assets/uploads/".$directory."/".$filename;

$i += 1;

return $filename;

}

Delete multiple image :

if(!empty($imageToDelete)){

$deleteMultipleImage = deleteMultipleImage($imageToDelete, $directory);

if (!$deleteMultipleImage) {

throw new Exception("Failed to delete images.");

}

}

function deleteMultipleImage($imageToDelete, $directory){

global $conn;

$allDeleted = true; // Flag to check if all images were deleted successfully

foreach($imageToDelete as $imagePath){

// Delete image from directory

$imageDeleteFromDirectory = deleteSingleImageFromDirectory($imagePath,$directory);

if ($imageDeleteFromDirectory) {

// Delete image record from database

$imageDeleteFromDatabaseTable = deleteSingleImageFromDatabaseTable($imagePath, $directory, $conn);

if (!$imageDeleteFromDatabaseTable) {

$allDeleted = false;

}

} else {

$allDeleted = false;

}

}

return $allDeleted;

}

function deleteSingleImageFromDirectory($imagePath,$directory){

$image = '../assets/uploads/'.$directory.'/'.$imagePath;

if(file\_exists($image)){

return unlink($image);

}else{

return false;

}

}

function deleteSingleImageFromDatabaseTable($imagePath,$directory,$conn ){

// $deleteQuery = "DELETE FROM photos WHERE file = '$imagePath'";

// $result = mysqli\_query($conn, $deleteQuery);

// Using a prepared statement to prevent SQL injection

$stmt = $conn->prepare("DELETE FROM photos WHERE file = ?");

$stmt->bind\_param("s", $imagePath);

$result = $stmt->execute();

if (!$result) {

return false;

}

return $result;

}

array\_map():

$numbers = [1, 2, 3, 4, 5];

$squaredNumbers = array\_map(function($num) {

return $num \* $num;

}, $numbers);

print\_r($squaredNumbers);

Output :

Array

(

[0] => 1

[1] => 4

[2] => 9

[3] => 16

[4] => 25

)

$firstNames = ["John", "Jane", "Bob"];

$lastNames = ["Doe", "Smith", "Johnson"];

$fullNames = array\_map(function($firstName, $lastName) {

return $firstName . ' ' . $lastName;

}, $firstNames, $lastNames);

print\_r($fullNames);

.

Array

(

[0] => John Doe

[1] => Jane Smith

[2] => Bob Johnson

)

$strings = ["hello", "world", "php"];

$upperCaseStrings = array\_map('strtoupper', $strings);

print\_r($upperCaseStrings);

Array

(

[0] => HELLO

[1] => WORLD

[2] => PHP

)

`created\_at` date NOT NULL DEFAULT current\_timestamp()