**PAGE 3**

1. Display all items having active status.

SELECT \* FROM products WHERE item\_status = 'A'

2. Display all users that are inactive.

SELECT \* FROM users WHERE user\_status = 'I'

3. Display the list of order's Descriptive information:

SELECT o.order\_id, o.date\_ordered, u.username, p.item\_name, SUM(o.item\_qty \* p.item\_price) as total\_amount

FROM orders o

JOIN users u

ON o.user\_id = u.user\_id

JOIN products p

ON o.item\_id = p.item\_id

GROUP BY o.order\_id;

4. Display all the order information having "Delivered" status

SELECT o.order\_status, o.order\_id, u.username, p.item\_name, SUM(o.item\_qty \* p.item\_price) as total\_amount, SUM(o.item\_qty) as total\_qty

FROM orders o

JOIN users u

ON o.user\_id = u.user\_id

JOIN products p

ON o.item\_id = p.item\_id

WHERE o.order\_status = 'D'

GROUP BY o.order\_id;

5. Display active items ordered by all active users and has been ordered for the year 2023 or after Dec. 31, 2022 and is still pending

SELECT p.item\_id, p.item\_name, o.item\_qty, u.username, u.user\_status, o.date\_ordered, o.order\_status

FROM orders o

JOIN users u

ON o.user\_id = u.user\_id

JOIN products p

ON o.item\_id = p.item\_id

WHERE u.user\_status = 'A'

AND p.item\_status = 'A'

AND o.order\_status = 'P'

AND o.date\_ordered >= '2023-01-01'

GROUP BY o.order\_id;

6. Write the html/php script that will display the list of Products. a sample output is provided for the users display, you need to display products. put an action button similar to the below image.

<html>

<?php include\_once "conn\_db.php"; ?>

<head>

<meta charset="UTF-8">

<title>List of Products</title>

<link rel="stylesheet" href="css/bootstrap.css">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-12">

<h3>List of Products</h3>

<?php

$product\_list = query($conn, "SELECT item\_id, item\_name, item\_price FROM products WHERE item\_status='A'");

echo "<hr>";

if(isset($\_GET['update\_status'])){

switch($\_GET['update\_status']){

case "success": echo "<div class='alert alert-success'>Product Updated.</div>";

break;

case "failed": echo "<div class='alert alert-danger'>Product Failed to be updated.</div>";

break;

}

}

echo "<hr>";

echo "<table class='table table-bordered'>";

echo "<thead>";

echo "<th>Product Name</th>";

echo "<th>Price</th>";

echo "<th>Action</th>";

echo "<th>Action</th>";

echo "</thead>";

foreach($product\_list as $key => $row){

echo "<tr>";

echo "<td>" . $row['item\_name'] . "</td>";

echo "<td>" . $row['item\_price'] . "</td>";

echo "<td> <a class='btn btn-success' href='update\_product.php?item\_id=". $row['item\_id'] ."'> Update </a> </td>";

echo "<td> <a class='btn btn-danger' href='delete\_product.php?item\_id=". $row['item\_id'] ."'> Delete </a> </td>";

echo "</tr>";

}

echo "</table>";

?>

</div>

</div>

</div>

</body>

<script src="js/bootstrap.js"></script>

</html>

7. BONUS: Write any other Situational Cases that you will encounter with your own Web Systems - related to your Systems Database. Provide similar solution using Display

Situational Case: when a customer wants to view their order history. In this case, the system would need to access the database of all past orders and display the relevant information to the customer.

<html>

<?php include\_once "db\_conn.php"; ?>

<head>

<meta charset="UTF-8">

<title>Order History</title>

<link rel="stylesheet" href="css/bootstrap.css">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-12">

<h3>Order History</h3>

<?php

$customer\_id = $\_SESSION['customer\_id'];

$order\_list = query($conn, "SELECT order\_id, order\_date, order\_total FROM orders WHERE customer\_id=$customer\_id");

echo "<table class='table table-bordered'>";

echo "<thead>";

echo "<th>Order ID</th>";

echo "<th>Date</th>";

echo "<th>Total</th>";

echo "</thead>";

foreach($order\_list as $key => $row){

echo "<tr>";

echo "<td>" . $row['order\_id'] . "</td>";

echo "<td>" . $row['order\_date'] . "</td>";

echo "<td>" . $row['order\_total'] . "</td>";

echo "</tr>";

}

echo "</table>";

?>

</div>

</div>

</div>

</body>

<script src="js/bootstrap.js"></script>

</html>

**PAGE 4**

8. Write the HTML/PHP form interface that will accept the Product information needed, Item Name and Item Price. No need to write the whole html-body, just need the FORM where it accepts the input.

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

<h3>Add Product</h3>

<?php

if(isset($\_GET['new\_record'])){

switch($\_GET['new\_record']){

case "added": echo "<div class='alert alert-success'>User Added.</div>";

break;

case "failed": echo "<div class='alert alert-danger'>User Not Added</div>";

break;

}

}

?>

<form action="add\_product.php" method="post">

<div class="form-group">

<label for="item\_name">Item Name</label>

<input type="text" class="form-control" id="item\_name" name="item\_name" required>

</div>

<div class="form-group mb-3">

<label for="item\_price">Item Price</label>

<input type="number" class="form-control" id="item\_price" name="item\_price" required>

</div>

<button type="submit" class="btn btn-primary">Add Product</button>

</form>

</div>

9. Write the php script that will capture the data that will be submitted be the form you created. the script should be able to perform the Insert into the products table based from the product info needed.

<?php

// add\_product.php

include\_once "conn\_db.php";

if(isset($\_POST['item\_name'])){

$item\_name=$\_POST['item\_name'];

$item\_price=$\_POST['item\_price'];

$table = "products";

$fields = array('item\_name' => $item\_name

, 'item\_price' => $item\_price

);

if(insert($conn, $table, $fields) ){

header("location: product\_list.php?new\_record=added");

exit();

}

else{

header("location: product\_list.php?new\_record=failed");

exit();

}

}

?>

10. BONUS: Write any other Situational Cases that you will encounter with your own Web Systems - related to your Systems Database. Provide similar solution using Insert

**Situational Case**: add a new customer to my database so that I can keep track of their orders and contact information.

<?php

// add\_customer.php

include\_once "conn\_db.php";

if(isset($\_POST['customer\_name'])){

$customer\_name=$\_POST['customer\_name'];

$customer\_email=$\_POST['customer\_email'];

$customer\_phone=$\_POST['customer\_phone'];

$delivery\_address=$\_POST['delivery\_address'];

$table = "customers";

$fields = array('customer\_name' => $customer\_name

, 'customer\_email' => $customer\_email

, 'customer\_phone' => $customer\_phone

, 'delivery\_address' => $delivery\_address

);

if(insert($conn, $table, $fields) ){

header("location: customer\_list.php?new\_record=added");

exit();

}

else{

header("location: customer\_list.php?new\_record=failed");

exit();

}

}

?>

**PAGE 5**

11. Write the html/php script where the link/button "Update" similar from the display in part 1 will lead to, in this case for the list of items. name that script as update\_item.php. it should have a form with the default product details that needs to be updated.

<html>

<head>

<meta charset="UTF-8">

<title>Update Product</title>

<link rel="stylesheet" href="css/bootstrap.css">

</head>

<body>

<div class="container">

<div class="row">

<div class="col-md-12">

<h1>Update Product</h1>

<form action="update\_product.php" method="post">

<input type="hidden" name="item\_id" value="<?php echo $\_GET['item\_id']; ?>">

<div class="form-group">

<label for="item\_name">Item Name</label>

<input type="text" class="form-control" id="item\_name" name="item\_name" value="<?php echo $item\_name; ?>" required>

</div>

<div class="form-group mb-3">

<label for="item\_price">Item Price</label>

<input type="number" class="form-control" id="item\_price" name="item\_price" value="<?php echo $item\_price; ?>" required>

</div>

<button type="submit" class="btn btn-primary">Update Product</button>

</form>

</div>

</div>

</div>

</body>

<script src="js/bootstrap.js"></script>

</html>

12. Write the php script that will accept the submitted data and update the items table using above form details.

<?php

// update\_product.php

include\_once "conn\_db.php";

if(isset($\_POST['item\_id'])){

$table = "products";

$p\_item\_id = $\_POST['item\_id'];

$p\_item\_name = $\_POST['item\_name'];

$p\_item\_price = $\_POST['item\_price'];

$fields = array( 'item\_name' => $p\_item\_name

, 'item\_price' => $p\_item\_price

);

$filter = array( 'item\_id' => $p\_item\_id );

if( update($conn, $table, $fields, $filter )){

header("location: product\_display.php?update\_status=success");

exit();

}

else{

header("location: product\_display.php?update\_status=failed");

exit();

}

}

else{

$item\_id = $\_GET['item\_id'];

$product = query($conn, "SELECT item\_name, item\_price FROM products WHERE item\_id=$item\_id AND item\_status='A'");

if(count($product) == 0){

header("location: product\_display.php");

exit();

}

$item\_name = $product[0]['item\_name'];

$item\_price = $product[0]['item\_price'];

}

?>

13. BONUS: Write any other Situational Cases that you will encounter with your own Web Systems - related to your Systems Database. Provide similar solution using Update.

Situational Case: One of the customers has placed an order for a dish and realized that they need to update the quantity of the item they ordered. They would like to increase the quantity of the dish by 2.

<?php

include\_once "conn\_db.php";

if(isset($\_POST['order\_id'])){

$table = "orders";

$p\_order\_id = $\_POST['order\_id'];

$p\_quantity = $\_POST['quantity'];

// Retrieve order details

$order\_details = query($conn, "SELECT \* FROM orders WHERE order\_id = '$p\_order\_id'");

if($order\_details){

// Calculate updated quantity

$updated\_quantity = $order\_details[0]['quantity'] + $p\_quantity;

// Update quantity in database

$fields = array('quantity' => $updated\_quantity);

$filter = array('order\_id' => $p\_order\_id);

if(update($conn, $table, $fields, $filter)){

// Notify customer of successful update

echo "<div class='alert alert-success'>Order updated successfully.</div>";

}

else{

// Notify customer of failed update

echo "<div class='alert alert-danger'>Failed to update order.</div>";

}

}

else{

// Notify customer that order does not exist

echo "<div class='alert alert-danger'>Order does not exist.</div>";

}

}

?>

**PAGE 6**

14. Write the php script called delete\_item.php where the button in the display will lead to. it should permanently delete the record from the database.

<?php

//delete\_product.php

include\_once "conn\_db.php";

if(isset($\_GET['item\_id'])){

$item\_id = $\_GET['item\_id'];

$params = array($item\_id);

if(query($conn, "DELETE FROM products WHERE item\_id = ?", $params) ){

header("location: product\_display.php?product\_delete=done");

exit();

}

else{

header("location: product\_display.php?product\_delete=failed");

exit();

}

}

?>

15. BONUS: Write any other Situational Cases that you will encounter with your own Web Systems - related to your Systems Database. Provide similar solution using Delete

Situational Case: Suppose a customer made an order, but later decided to cancel it. In this case, we need to delete the order from the database.

<?php

include\_once "conn\_db.php";

if(isset($\_GET['order\_id'])){

if(query($conn, "DELETE FROM orders WHERE order\_id = ?", $params) ){

header("location: order\_history.php?order\_delete=done");

exit();

}

else{

header("location: order\_history.php?order\_delete=failed");

exit();

}

}