**1. Database Setup**

Create a MySQL database with the following tables:

**users table**

sql

Copy code

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

email VARCHAR(100) UNIQUE,

password VARCHAR(255)

);

**products table**

sql

Copy code

CREATE TABLE products (

product\_id INT AUTO\_INCREMENT PRIMARY KEY,

product\_name VARCHAR(100),

price DECIMAL(10, 2),

stock INT

);

**cart table**

sql

Copy code

CREATE TABLE cart (

cart\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

product\_id INT,

quantity INT,

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

**2. Login System**

Ensure the user is logged in before allowing access to the cart.

**Sign In Script (signin.php)**

* Validate user credentials.
* Start a session and store the user's user\_id.

php

Copy code

session\_start();

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

include 'db.php'; // Database connection file

$email = $\_POST['email'];

$password = $\_POST['password'];

$query = "SELECT \* FROM users WHERE email = ?";

$stmt = $conn->prepare($query);

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

$stmt->execute();

$result = $stmt->get\_result();

if ($result->num\_rows > 0) {

$user = $result->fetch\_assoc();

if (password\_verify($password, $user['password'])) {

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

header("Location: products.php"); // Redirect to products page

} else {

echo "Invalid password.";

}

} else {

echo "No account found with this email.";

}

}

**3. Display Products**

Create a page (products.php) to display all products from the database.

php

Copy code

<?php

session\_start();

if (!isset($\_SESSION['user\_id'])) {

header("Location: signin.php");

exit;

}

include 'db.php'; // Database connection

$query = "SELECT \* FROM products";

$result = $conn->query($query);

while ($row = $result->fetch\_assoc()) {

echo "

<div>

<h3>{$row['product\_name']}</h3>

<p>Price: {$row['price']}</p>

<form method='POST' action='add\_to\_cart.php'>

<input type='hidden' name='product\_id' value='{$row['product\_id']}'>

<input type='number' name='quantity' min='1' max='{$row['stock']}' required>

<button type='submit'>Add to Cart</button>

</form>

</div>";

}

?>

**4. Add to Cart**

Create a script (add\_to\_cart.php) to handle adding products to the cart.

php

Copy code

<?php

session\_start();

if (!isset($\_SESSION['user\_id'])) {

header("Location: signin.php");

exit;

}

include 'db.php';

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

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

$product\_id = $\_POST['product\_id'];

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

// Check if the product is already in the cart

$query = "SELECT \* FROM cart WHERE user\_id = ? AND product\_id = ?";

$stmt = $conn->prepare($query);

$stmt->bind\_param("ii", $user\_id, $product\_id);

$stmt->execute();

$result = $stmt->get\_result();

if ($result->num\_rows > 0) {

// Update quantity

$query = "UPDATE cart SET quantity = quantity + ? WHERE user\_id = ? AND product\_id = ?";

$stmt = $conn->prepare($query);

$stmt->bind\_param("iii", $quantity, $user\_id, $product\_id);

$stmt->execute();

} else {

// Add new item to the cart

$query = "INSERT INTO cart (user\_id, product\_id, quantity) VALUES (?, ?, ?)";

$stmt = $conn->prepare($query);

$stmt->bind\_param("iii", $user\_id, $product\_id, $quantity);

$stmt->execute();

}

header("Location: view\_cart.php");

}

?>

**5. View Cart**

Create a page (view\_cart.php) to display the user's cart items.

php

Copy code

<?php

session\_start();

if (!isset($\_SESSION['user\_id'])) {

header("Location: signin.php");

exit;

}

include 'db.php';

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

$query = "

SELECT c.cart\_id, p.product\_name, p.price, c.quantity

FROM cart c

JOIN products p ON c.product\_id = p.product\_id

WHERE c.user\_id = ?";

$stmt = $conn->prepare($query);

$stmt->bind\_param("i", $user\_id);

$stmt->execute();

$result = $stmt->get\_result();

while ($row = $result->fetch\_assoc()) {

$total\_price = $row['price'] \* $row['quantity'];

echo "

<div>

<h3>{$row['product\_name']}</h3>

<p>Price: {$row['price']}</p>

<p>Quantity: {$row['quantity']}</p>

<p>Total: $total\_price</p>

<form method='POST' action='remove\_from\_cart.php'>

<input type='hidden' name='cart\_id' value='{$row['cart\_id']}'>

<button type='submit'>Remove</button>

</form>

</div>";

}

?>

**6. Remove from Cart**

Create a script (remove\_from\_cart.php) to handle removing items from the cart.

php

Copy code

<?php

session\_start();

if (!isset($\_SESSION['user\_id'])) {

header("Location: signin.php");

exit;

}

include 'db.php';

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

$cart\_id = $\_POST['cart\_id'];

$query = "DELETE FROM cart WHERE cart\_id = ?";

$stmt = $conn->prepare($query);

$stmt->bind\_param("i", $cart\_id);

$stmt->execute();

header("Location: view\_cart.php");

}

?>

**7. Test the Functionality**

1. **Sign In** to access the cart.
2. View products on the products.php page.
3. Add products to the cart using add\_to\_cart.php.
4. View your cart on the view\_cart.php page.
5. Remove items from the cart using remove\_from\_cart.php.

Let me know if you need additional features or adjustments! 😊

**1. Checkout Process**

**a. Create an Order Table**

You need a table to store order details when a user checks out.

sql

Copy code

CREATE TABLE orders (

order\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

total\_price DECIMAL(10, 2),

order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

status ENUM('Pending', 'Completed', 'Cancelled') DEFAULT 'Pending',

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);

CREATE TABLE order\_items (

order\_item\_id INT AUTO\_INCREMENT PRIMARY KEY,

order\_id INT,

product\_id INT,

quantity INT,

price DECIMAL(10, 2),

FOREIGN KEY (order\_id) REFERENCES orders(order\_id) ON DELETE CASCADE,

FOREIGN KEY (product\_id) REFERENCES products(product\_id) ON DELETE CASCADE

);

**b. Checkout Page (checkout.php)**

Create a page where users can confirm their cart and place an order.

php

Copy code

<?php

session\_start();

if (!isset($\_SESSION['user\_id'])) {

header("Location: signin.php");

exit;

}

include 'db.php';

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

// Fetch cart items

$query = "

SELECT c.product\_id, p.product\_name, p.price, c.quantity

FROM cart c

JOIN products p ON c.product\_id = p.product\_id

WHERE c.user\_id = ?";

$stmt = $conn->prepare($query);

$stmt->bind\_param("i", $user\_id);

$stmt->execute();

$result = $stmt->get\_result();

$total\_price = 0;

$cart\_items = [];

while ($row = $result->fetch\_assoc()) {

$cart\_items[] = $row;

$total\_price += $row['price'] \* $row['quantity'];

}

// Confirm Order

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

// Insert into orders

$query = "INSERT INTO orders (user\_id, total\_price) VALUES (?, ?)";

$stmt = $conn->prepare($query);

$stmt->bind\_param("id", $user\_id, $total\_price);

$stmt->execute();

$order\_id = $stmt->insert\_id;

// Insert into order\_items

foreach ($cart\_items as $item) {

$query = "INSERT INTO order\_items (order\_id, product\_id, quantity, price) VALUES (?, ?, ?, ?)";

$stmt = $conn->prepare($query);

$stmt->bind\_param("iiid", $order\_id, $item['product\_id'], $item['quantity'], $item['price']);

$stmt->execute();

}

// Clear cart

$query = "DELETE FROM cart WHERE user\_id = ?";

$stmt = $conn->prepare($query);

$stmt->bind\_param("i", $user\_id);

$stmt->execute();

header("Location: orders.php");

}

?>

<h2>Checkout</h2>

<?php foreach ($cart\_items as $item): ?>

<div>

<h3><?= $item['product\_name'] ?></h3>

<p>Price: <?= $item['price'] ?></p>

<p>Quantity: <?= $item['quantity'] ?></p>

</div>

<?php endforeach; ?>

<p>Total Price: <?= $total\_price ?></p>

<form method="POST">

<button type="submit">Place Order</button>

</form>

**2. Order History**

**a. Create an Order History Page (orders.php)**

Show the logged-in user their order history.

php

Copy code

<?php

session\_start();

if (!isset($\_SESSION['user\_id'])) {

header("Location: signin.php");

exit;

}

include 'db.php';

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

$query = "SELECT \* FROM orders WHERE user\_id = ?";

$stmt = $conn->prepare($query);

$stmt->bind\_param("i", $user\_id);

$stmt->execute();

$result = $stmt->get\_result();

while ($row = $result->fetch\_assoc()) {

echo "

<div>

<h3>Order ID: {$row['order\_id']}</h3>

<p>Total Price: {$row['total\_price']}</p>

<p>Status: {$row['status']}</p>

<p>Order Date: {$row['order\_date']}</p>

</div>";

}

?>

**3. Stock Management**

Update the stock of products after an order is placed:

* **In checkout.php**, when inserting order\_items, subtract the purchased quantity from the products table:

php

Copy code

$query = "UPDATE products SET stock = stock - ? WHERE product\_id = ?";

$stmt = $conn->prepare($query);

$stmt->bind\_param("ii", $item['quantity'], $item['product\_id']);

$stmt->execute();

**4. Admin Panel for Order Management**

**a. View All Orders**

Allow admins to view all orders and their statuses. Use a query like:

php

Copy code

SELECT o.order\_id, u.name, o.total\_price, o.order\_date, o.status

FROM orders o

JOIN users u ON o.user\_id = u.user\_id;

**b. Update Order Status**

Add functionality for admins to update the status of orders to "Completed" or "Cancelled."

**5. Optional Features**

* **Email Notifications:** Send an email confirmation when an order is placed.
* **Coupon System:** Add discount codes to reduce total price.
* **Wishlist:** Allow users to save products for future purchase.
* **Delivery Tracking:** Add a column for delivery status in the orders table.

Let me know which step you’d like to dive into, and I’ll guide you through! 😊