Create table

CREATE TABLE user (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

email VARCHAR(255) NOT NULL UNIQUE,

fname VARCHAR(100) NOT NULL,

lname VARCHAR(100) NOT NULL,

phone VARCHAR(15),

user\_type ENUM('admin', 'customer', 'rider') NOT NULL,

image VARCHAR(255),

password VARCHAR(255) NOT NULL

);

CREATE TABLE admin (

admin\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

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

);

CREATE TABLE customer (

customer\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT,

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

);

CREATE TABLE cart (

cart\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT,

FOREIGN KEY (customer\_id) REFERENCES customer(customer\_id) ON DELETE CASCADE

);

CREATE TABLE cart\_item (

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

qty INT NOT NULL,

cart\_id INT,

product\_id INT,

FOREIGN KEY (cart\_id) REFERENCES cart(cart\_id) ON DELETE CASCADE,

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

);

CREATE TABLE category (

category\_id INT AUTO\_INCREMENT PRIMARY KEY,

category\_name VARCHAR(255) NOT NULL

);

CREATE TABLE customer\_address (

address\_id INT AUTO\_INCREMENT PRIMARY KEY,

full\_address TEXT NOT NULL,

city\_corporation VARCHAR(255),

upazila VARCHAR(255),

postal\_code VARCHAR(10),

customer\_id INT,

FOREIGN KEY (customer\_id) REFERENCES customer(customer\_id) ON DELETE CASCADE

);

CREATE TABLE delivery (

delivery\_id INT AUTO\_INCREMENT PRIMARY KEY,

actual\_delivery\_date DATE,

delivery\_status ENUM('pending', 'shipped', 'delivered') DEFAULT 'pending',

tracking\_number VARCHAR(100),

rider\_id INT,

order\_id INT,

FOREIGN KEY (rider\_id) REFERENCES rider(rider\_id) ON DELETE CASCADE,

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

);

CREATE TABLE gift\_card (

gift\_card\_id INT AUTO\_INCREMENT PRIMARY KEY,

gift\_amount DECIMAL(10, 2),

user\_id INT,

personal\_note TEXT,

reciever\_mail VARCHAR(255),

sender\_mail VARCHAR(255),

generated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

used\_status BOOLEAN DEFAULT FALSE,

payment\_id INT,

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

FOREIGN KEY (payment\_id) REFERENCES payment(payment\_id) ON DELETE CASCADE

);

CREATE TABLE gift\_card\_transaction (

transaction\_id INT AUTO\_INCREMENT PRIMARY KEY,

transaction\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

used\_amount DECIMAL(10, 2),

order\_id INT,

gift\_card\_id INT,

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

FOREIGN KEY (gift\_card\_id) REFERENCES gift\_card(gift\_card\_id) ON DELETE CASCADE

);

CREATE TABLE orders (

order\_id INT AUTO\_INCREMENT PRIMARY KEY,

order\_date TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

total\_amount DECIMAL(10, 2),

user\_id INT,

payment\_id INT,

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

FOREIGN KEY (payment\_id) REFERENCES payment(payment\_id) ON DELETE CASCADE

);

CREATE TABLE payment (

payment\_id INT AUTO\_INCREMENT PRIMARY KEY,

payment\_method VARCHAR(100),

total\_amount DECIMAL(10, 2),

transaction\_id VARCHAR(100),

order\_id INT,

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

);

CREATE TABLE product (

product\_id INT AUTO\_INCREMENT PRIMARY KEY,

product\_name VARCHAR(255),

product\_price DECIMAL(10, 2),

description TEXT,

image VARCHAR(255),

stock\_qty INT,

active\_status BOOLEAN DEFAULT TRUE,

category\_id INT,

FOREIGN KEY (category\_id) REFERENCES category(category\_id) ON DELETE CASCADE

);

CREATE TABLE product\_audit (

audit\_id INT AUTO\_INCREMENT PRIMARY KEY,

product\_id INT,

timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

admin\_id INT,

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

FOREIGN KEY (admin\_id) REFERENCES admin(admin\_id) ON DELETE CASCADE

);

CREATE TABLE rider (

rider\_id INT AUTO\_INCREMENT PRIMARY KEY,

rider\_phone VARCHAR(15),

rider\_active\_status BOOLEAN DEFAULT TRUE,

max\_delivery INT DEFAULT 0,

user\_id INT,

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

);

CREATE TABLE rider\_preferred\_area (

rider\_id INT,

city\_corporation VARCHAR(255),

upazila VARCHAR(255),

FOREIGN KEY (rider\_id) REFERENCES rider(rider\_id) ON DELETE CASCADE

);

CREATE TABLE wishlist (

wishlist\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT,

product\_id INT,

FOREIGN KEY (customer\_id) REFERENCES customer(customer\_id) ON DELETE CASCADE,

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

);

1. Tea Was Once Used as Currency

In ancient China, tea was so valuable that it was used as currency. During the Tang Dynasty (618–907 AD), compressed tea bricks were used for trade and tax payments. These tea bricks were especially popular in Tibet, Mongolia, and Central Asia, where they were exchanged for horses, silk, and other goods. Even soldiers were sometimes paid in tea instead of money. The practice lasted for centuries until loose-leaf tea became more common. Today, while tea is no longer a form of currency, its economic and cultural value remains strong worldwide.

2. Bangladeshis Love Their Tea—Even More Than Water!

In Bangladesh, tea is more than just a drink—it’s a way of life. Many Bangladeshis consume more tea than water daily, making it a central part of their routine. Whether in a village, a city café, or a roadside tea stall (*cha-er dokan*), tea is always available. The most common tea in Bangladesh is black tea with milk and sugar, but herbal and spiced varieties are also gaining popularity. Tea stalls serve as social hubs, where people gather to chat, debate, or just relax. This deep connection to tea makes Bangladesh one of the most tea-loving nations in the world.

3. The British Once Tried to Keep Tea a Secret

When the British first started importing tea from China in the 17th century, they treated it as a highly guarded secret. At first, tea was considered a luxury product, only affordable to the wealthy. To maintain their monopoly, the British did not reveal how tea was cultivated. However, in the 19th century, a Scottish botanist named Robert Fortune disguised himself as a Chinese merchant and smuggled tea plants and seeds to India. This act led to the establishment of mass tea plantations in India and later Bangladesh, changing the global tea industry forever.

8. In Russia, Tea Was Once Served with Jam

In Russia, traditional tea-drinking often includes a spoonful of jam! Instead of adding sugar, Russians would take a sip of strong black tea and then eat a bit of sweet fruit jam to balance the bitterness. This practice, called "tea with jam", is still common today, especially in winter. Some even mix jam directly into their tea for a unique fruity flavor. It’s a delicious way to enjoy tea, and maybe you can try it too!

6. Tea Can Glow in the Dark

Some types of tea, like tonic-flavored herbal teas, can glow under blacklight! This happens because they contain quinine, a natural compound found in tonic water. When exposed to ultraviolet (UV) light, quinine glows blue. Though regular tea won’t glow in the dark, certain herbal blends with citrus or flower extracts can create a faint neon effect. Imagine sipping on a glowing tea at night—it’s like a magic potion in a cup!

<?php

// user\_login.php

// Handles user login and password reset

include 'db\_connection.php';

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

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

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

if (isset($\_POST['reset\_password'])) {

// Check if email exists

$stmt = $conn->prepare("SELECT \* FROM user WHERE email = ?");

$stmt->execute([$email]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($user) {

$new\_password = password\_hash($\_POST['new\_password'], PASSWORD\_BCRYPT);

$update\_stmt = $conn->prepare("UPDATE user SET password = ? WHERE email = ?");

$update\_stmt->execute([$new\_password, $email]);

echo "Password updated successfully!";

} else {

echo "Email not found!";

}

} else {

// Normal login process

$stmt = $conn->prepare("SELECT \* FROM user WHERE email = ?");

$stmt->execute([$email]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

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

session\_start();

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

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

header("Location: dashboard.php");

} else {

echo "Invalid credentials!";

}

}

}

?>

<?php

// admin\_add\_or\_update\_product.php

// Handles adding, updating, and deleting products

include 'db\_connection.php';

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

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

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

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

$stock\_qty = $\_POST['stock\_qty'];

$active\_status = $\_POST['active\_status'];

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

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

// Update product

$stmt = $conn->prepare("UPDATE product SET product\_name=?, product\_price=?, description=?, stock\_qty=?, active\_status=?, category\_id=? WHERE product\_id=?");

$stmt->execute([$product\_name, $product\_price, $description, $stock\_qty, $active\_status, $category\_id, $\_POST['product\_id']]);

} else {

// Add new product

$stmt = $conn->prepare("INSERT INTO product (product\_name, product\_price, description, stock\_qty, active\_status, category\_id) VALUES (?, ?, ?, ?, ?, ?)");

$stmt->execute([$product\_name, $product\_price, $description, $stock\_qty, $active\_status, $category\_id]);

}

}

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

// Delete product

$stmt = $conn->prepare("DELETE FROM product WHERE product\_id = ?");

$stmt->execute([$\_GET['delete\_id']]);

}

?>

<?php

// admin\_add\_or\_update\_category.php

// Handles adding, updating, and deleting categories

include 'db\_connection.php';

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

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

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

// Update category

$stmt = $conn->prepare("UPDATE category SET category\_name=? WHERE category\_id=?");

$stmt->execute([$category\_name, $\_POST['category\_id']]);

} else {

// Add new category

$stmt = $conn->prepare("INSERT INTO category (category\_name) VALUES (?)");

$stmt->execute([$category\_name]);

}

}

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

// Delete category

$stmt = $conn->prepare("DELETE FROM category WHERE category\_id = ?");

$stmt->execute([$\_GET['delete\_id']]);

}

?>

<?php

// buy\_gift\_card.php

// Handles purchasing a gift card and sending email

include 'db\_connection.php';

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

$gift\_amount = $\_POST['gift\_amount'];

$sender\_mail = $\_POST['sender\_mail'];

$receiver\_mail = $\_POST['receiver\_mail'];

$personal\_note = $\_POST['personal\_note'];

$stmt = $conn->prepare("INSERT INTO gift\_card (gift\_amount, sender\_mail, receiver\_mail, personal\_note, generated\_at, used\_status) VALUES (?, ?, ?, ?, NOW(), 0)");

$stmt->execute([$gift\_amount, $sender\_mail, $receiver\_mail, $personal\_note]);

$gift\_card\_id = $conn->lastInsertId();

// Send email (mocked, use mail() or PHPMailer)

echo "Gift Card #$gift\_card\_id sent to $receiver\_mail!";

}

?>

<?php

// payment.php

// Handles payment processing and first-order discount

include 'db\_connection.php';

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

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

$payment\_method = $\_POST['payment\_method'];

$total\_amount = $\_POST['total\_amount'];

$gift\_card\_id = $\_POST['gift\_card\_id'] ?? null;

// Check first-order discount

$stmt = $conn->prepare("SELECT COUNT(\*) FROM orders WHERE user\_id = ?");

$stmt->execute([$user\_id]);

$order\_count = $stmt->fetchColumn();

if ($order\_count == 0) {

$total\_amount \*= 0.9; // 10% discount

}

if ($gift\_card\_id) {

$stmt = $conn->prepare("SELECT gift\_amount FROM gift\_card WHERE gift\_card\_id = ? AND used\_status = 0");

$stmt->execute([$gift\_card\_id]);

$gift\_card = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($gift\_card) {

$total\_amount -= $gift\_card['gift\_amount'];

$stmt = $conn->prepare("UPDATE gift\_card SET used\_status = 1 WHERE gift\_card\_id = ?");

$stmt->execute([$gift\_card\_id]);

}

}

// Insert payment record

$stmt = $conn->prepare("INSERT INTO payment (payment\_method, total\_amount, payment\_status) VALUES (?, ?, 'pending')");

$stmt->execute([$payment\_method, $total\_amount]);

echo "Payment recorded successfully!";

}

?>

<?php

// user\_login.php

// Handles user login and password reset

include 'db\_connection.php';

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

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

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

if (isset($\_POST['reset\_password'])) {

// Check if email exists

$stmt = $conn->prepare("SELECT \* FROM user WHERE email = ?");

$stmt->execute([$email]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($user) {

$new\_password = password\_hash($\_POST['new\_password'], PASSWORD\_BCRYPT);

$update\_stmt = $conn->prepare("UPDATE user SET password = ? WHERE email = ?");

$update\_stmt->execute([$new\_password, $email]);

echo "Password updated successfully!";

} else {

echo "Email not found!";

}

} else {

// Normal login process

$stmt = $conn->prepare("SELECT \* FROM user WHERE email = ?");

$stmt->execute([$email]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

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

session\_start();

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

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

header("Location: dashboard.php");

} else {

echo "Invalid credentials!";

}

}

}

?>

<?php

// admin\_add\_or\_update\_product.php

// Handles adding, updating, and deleting products

include 'db\_connection.php';

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

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

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

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

$stock\_qty = $\_POST['stock\_qty'];

$active\_status = $\_POST['active\_status'];

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

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

// Update product

$stmt = $conn->prepare("UPDATE product SET product\_name=?, product\_price=?, description=?, stock\_qty=?, active\_status=?, category\_id=? WHERE product\_id=?");

$stmt->execute([$product\_name, $product\_price, $description, $stock\_qty, $active\_status, $category\_id, $\_POST['product\_id']]);

} else {

// Add new product

$stmt = $conn->prepare("INSERT INTO product (product\_name, product\_price, description, stock\_qty, active\_status, category\_id) VALUES (?, ?, ?, ?, ?, ?)");

$stmt->execute([$product\_name, $product\_price, $description, $stock\_qty, $active\_status, $category\_id]);

}

}

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

// Delete product

$stmt = $conn->prepare("DELETE FROM product WHERE product\_id = ?");

$stmt->execute([$\_GET['delete\_id']]);

}

?>

<?php

// user\_login.php

// Handles user login and password reset

include 'db\_connection.php';

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

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

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

if (isset($\_POST['reset\_password'])) {

// Check if email exists

$stmt = $conn->prepare("SELECT \* FROM user WHERE email = ?");

$stmt->execute([$email]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($user) {

$new\_password = password\_hash($\_POST['new\_password'], PASSWORD\_BCRYPT);

$update\_stmt = $conn->prepare("UPDATE user SET password = ? WHERE email = ?");

$update\_stmt->execute([$new\_password, $email]);

echo "Password updated successfully!";

} else {

echo "Email not found!";

}

} else {

// Normal login process

$stmt = $conn->prepare("SELECT \* FROM user WHERE email = ?");

$stmt->execute([$email]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

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

session\_start();

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

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

header("Location: dashboard.php");

} else {

echo "Invalid credentials!";

}

}

}

?>

<?php

// admin\_add\_or\_update\_product.php

// Handles adding, updating, and deleting products

include 'db\_connection.php';

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

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

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

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

$stock\_qty = $\_POST['stock\_qty'];

$active\_status = $\_POST['active\_status'];

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

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

// Update product

$stmt = $conn->prepare("UPDATE product SET product\_name=?, product\_price=?, description=?, stock\_qty=?, active\_status=?, category\_id=? WHERE product\_id=?");

$stmt->execute([$product\_name, $product\_price, $description, $stock\_qty, $active\_status, $category\_id, $\_POST['product\_id']]);

} else {

// Add new product

$stmt = $conn->prepare("INSERT INTO product (product\_name, product\_price, description, stock\_qty, active\_status, category\_id) VALUES (?, ?, ?, ?, ?, ?)");

$stmt->execute([$product\_name, $product\_price, $description, $stock\_qty, $active\_status, $category\_id]);

}

}

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

// Delete product

$stmt = $conn->prepare("DELETE FROM product WHERE product\_id = ?");

$stmt->execute([$\_GET['delete\_id']]);

}

?>

<?php

// admin\_add\_or\_update\_category.php

// Handles adding, updating, and deleting categories

include 'db\_connection.php';

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

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

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

// Update category

$stmt = $conn->prepare("UPDATE category SET category\_name=? WHERE category\_id=?");

$stmt->execute([$category\_name, $\_POST['category\_id']]);

} else {

// Add new category

$stmt = $conn->prepare("INSERT INTO category (category\_name) VALUES (?)");

$stmt->execute([$category\_name]);

}

}

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

// Delete category

$stmt = $conn->prepare("DELETE FROM category WHERE category\_id = ?");

$stmt->execute([$\_GET['delete\_id']]);

}

?>

<?php

// buy\_gift\_card.php

// Handles purchasing a gift card and sending email

include 'db\_connection.php';

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

$gift\_amount = $\_POST['gift\_amount'];

$sender\_mail = $\_POST['sender\_mail'];

$receiver\_mail = $\_POST['receiver\_mail'];

$personal\_note = $\_POST['personal\_note'];

$stmt = $conn->prepare("INSERT INTO gift\_card (gift\_amount, sender\_mail, receiver\_mail, personal\_note, generated\_at, used\_status) VALUES (?, ?, ?, ?, NOW(), 0)");

$stmt->execute([$gift\_amount, $sender\_mail, $receiver\_mail, $personal\_note]);

$gift\_card\_id = $conn->lastInsertId();

// Send email (mocked, use mail() or PHPMailer)

echo "Gift Card #$gift\_card\_id sent to $receiver\_mail!";

}

?>

<?php

// payment.php

// Handles payment processing and first-order discount

include 'db\_connection.php';

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

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

$payment\_method = $\_POST['payment\_method'];

$total\_amount = $\_POST['total\_amount'];

$gift\_card\_id = $\_POST['gift\_card\_id'] ?? null;

// Check first-order discount

$stmt = $conn->prepare("SELECT COUNT(\*) FROM orders WHERE user\_id = ?");

$stmt->execute([$user\_id]);

$order\_count = $stmt->fetchColumn();

if ($order\_count == 0) {

$total\_amount \*= 0.9; // 10% discount

}

if ($gift\_card\_id) {

$stmt = $conn->prepare("SELECT gift\_amount FROM gift\_card WHERE gift\_card\_id = ? AND used\_status = 0");

$stmt->execute([$gift\_card\_id]);

$gift\_card = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($gift\_card) {

$total\_amount -= $gift\_card['gift\_amount'];

$stmt = $conn->prepare("UPDATE gift\_card SET used\_status = 1 WHERE gift\_card\_id = ?");

$stmt->execute([$gift\_card\_id]);

}

}

// Insert payment record

$stmt = $conn->prepare("INSERT INTO payment (payment\_method, total\_amount, payment\_status) VALUES (?, ?, 'pending')");

$stmt->execute([$payment\_method, $total\_amount]);

echo "Payment recorded successfully!";

}

?>

<?php

// admin\_update\_order.php

// Handles order updates by the admin

include 'db\_connection.php';

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

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

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

if ($status == 'shipped') {

$stmt = $conn->prepare("UPDATE orders SET delivery\_status = 'shipped' WHERE order\_id = ?");

$stmt->execute([$order\_id]);

}

}

?>

<?php

// user\_sign\_up.php

// Handles user registration

include 'db\_connection.php';

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

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

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

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

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

$password = password\_hash($\_POST['password'], PASSWORD\_BCRYPT);

$user\_type = $\_POST['user\_type'];

$image = null;

if (!empty($\_FILES['image']['name'])) {

$image = file\_get\_contents($\_FILES['image']['tmp\_name']);

}

$stmt = $conn->prepare("INSERT INTO user (email, fname, lname, phone, user\_type, image, password) VALUES (?, ?, ?, ?, ?, ?, ?)");

$stmt->execute([$email, $fname, $lname, $phone, $user\_type, $image, $password]);

echo "User registered successfully!";

}

?>

<?php

// admin\_update\_order.php

// Handles order updates by the admin

include 'db\_connection.php';

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

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

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

if ($status == 'shipped') {

$stmt = $conn->prepare("UPDATE orders SET delivery\_status = 'shipped' WHERE order\_id = ?");

$stmt->execute([$order\_id]);

// Assign rider automatically

$stmt = $conn->prepare("SELECT customer\_address.city\_corporation, customer\_address.upazila FROM orders

JOIN customer\_address ON orders.user\_id = customer\_address.customer\_id

WHERE orders.order\_id = ?");

$stmt->execute([$order\_id]);

$address = $stmt->fetch(PDO::FETCH\_ASSOC);

$stmt = $conn->prepare("SELECT rider\_id FROM rider\_preferred\_area

JOIN rider ON rider\_preferred\_area.rider\_id = rider.rider\_id

WHERE city\_corporation = ? AND upazila = ?

AND rider\_active\_status = 1 AND (SELECT COUNT(\*) FROM delivery WHERE rider\_id = rider.rider\_id AND delivery\_status != 'delivered') < 4

LIMIT 1");

$stmt->execute([$address['city\_corporation'], $address['upazila']]);

$rider = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($rider) {

$stmt = $conn->prepare("UPDATE delivery SET rider\_id = ?, delivery\_status = 'shipped' WHERE order\_id = ?");

$stmt->execute([$rider['rider\_id'], $order\_id]);

echo "Rider assigned successfully!";

} else {

echo "No available riders in this area.";

}

}

}

?>

<?php

// rider\_update\_delivery.php

// Handles delivery status updates by riders

include 'db\_connection.php';

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

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

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

if ($status == 'delivered') {

$stmt = $conn->prepare("UPDATE delivery SET delivery\_status = 'delivered' WHERE delivery\_id = ?");

$stmt->execute([$delivery\_id]);

echo "Order delivered successfully!";

}

}

?>

another question, i want to know how to solve this issue or assess this issue using html, css, js, php and sql. I want to show the default home page when anyone enters the website, but when they sign-up they automatically get logged in, and for the sign-in the same. but suppose when they login, their email is checked with their user\_type from the database and later on password, so suppose when a user logs in and from the database it sees the user\_type to be admin, it shows a loading page, and then loads the panel especially designed for admins, and when a user logs in and from the database it sees the user\_type to be customer, it shows a loading page, and then loads the panel especially designed for customers, and for the riders same thing as well

<?php

// user\_login.php

include 'db\_connection.php';

session\_start();

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

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

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

$stmt = $conn->prepare("SELECT \* FROM user WHERE email = ?");

$stmt->execute([$email]);

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

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

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

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

echo json\_encode(["status" => "success", "user\_type" => $user['user\_type']]);

} else {

echo json\_encode(["status" => "error", "message" => "Invalid credentials!"]);

}

}

?>

<?php

// user\_sign\_up.php

include 'db\_connection.php';

session\_start();

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

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

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

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

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

$password = password\_hash($\_POST['password'], PASSWORD\_BCRYPT);

$user\_type = $\_POST['user\_type'];

$stmt = $conn->prepare("INSERT INTO user (email, fname, lname, phone, user\_type, password) VALUES (?, ?, ?, ?, ?, ?)");

$stmt->execute([$email, $fname, $lname, $phone, $user\_type, $password]);

$user\_id = $conn->lastInsertId(); // Get newly inserted user ID

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

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

echo json\_encode(["status" => "success", "user\_type" => $user\_type]);

}

?>

document.addEventListener("DOMContentLoaded", function () {

const loginForm = document.getElementById("loginForm");

if (loginForm) {

loginForm.addEventListener("submit", function (e) {

e.preventDefault();

let formData = new FormData(loginForm);

fetch("user\_login.php", {

method: "POST",

body: formData,

})

.then(response => response.json())

.then(data => {

if (data.status === "success") {

showLoadingScreen();

setTimeout(() => {

if (data.user\_type === "admin") {

window.location.href = "admin\_dashboard.php";

} else if (data.user\_type === "customer") {

window.location.href = "customer\_dashboard.php";

} else if (data.user\_type === "rider") {

window.location.href = "rider\_dashboard.php";

}

}, 2000); // Simulate loading

} else {

alert(data.message);

}

})

.catch(error => console.log(error));

});

}

});

function showLoadingScreen() {

document.body.innerHTML = `

<div style="display:flex; align-items:center; justify-content:center; height:100vh; font-size:24px;">

Loading your dashboard...

</div>

`;

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>Cha Kahben - Home</title>

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

</head>

<body>

<h1>Welcome to Cha Kahben</h1>

<form id="loginForm">

<input type="email" name="email" placeholder="Email" required>

<input type="password" name="password" placeholder="Password" required>

<button type="submit">Login</button>

</form>

</body>

</html>

and also when they want to place an order, they will get an notification, that sign-in to place an order and they get redirected to the index.html page basically home page and the user icon from the navigation bar will show a transform-scale(1.2) for 2 seconds and a different background and color to the user icon at the same time.

<button id="placeOrderBtn">Place Order</button>

document.addEventListener("DOMContentLoaded", function () {

const placeOrderBtn = document.getElementById("placeOrderBtn");

if (placeOrderBtn) {

placeOrderBtn.addEventListener("click", function (e) {

e.preventDefault();

fetch("check\_auth.php")

.then(response => response.json())

.then(data => {

if (data.authenticated) {

window.location.href = "checkout.php"; // Proceed to checkout

} else {

// Store the redirect trigger in localStorage

localStorage.setItem("redirectAfterLogin", "true");

// Redirect to home page

window.location.href = "index.html";

}

})

.catch(error => console.log(error));

});

}

// Check if user was redirected after attempting to place an order

if (localStorage.getItem("redirectAfterLogin") === "true") {

localStorage.removeItem("redirectAfterLogin"); // Clear the flag

const userIcon = document.getElementById("userIcon");

if (userIcon) {

userIcon.classList.add("highlight-user-icon");

setTimeout(() => {

userIcon.classList.remove("highlight-user-icon");

}, 2000);

}

}

});

<?php

session\_start();

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

echo json\_encode(["authenticated" => true]);

} else {

echo json\_encode(["authenticated" => false]);

}

?>

.highlight-user-icon {

transform: scale(1.2);

background-color: #ffcc00; /\* Yellow background \*/

color: #000; /\* Black text \*/

transition: all 0.5s ease-in-out;

}

<!DOCTYPE html>

<html lang="en">

<head>

<title>Cha Kahben - Home</title>

<link rel="stylesheet" href="style.css">

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

</head>

<body>

<nav>

<ul>

<li><a href="menu.html">Menu</a></li>

<li><a href="about.html">About</a></li>

<li><a href="contact.html">Contact</a></li>

<li id="userIcon"><a href="user\_profile.php">User</a></li>

</ul>

</nav>

<h1>Welcome to Cha Kahben</h1>

</body>

</html>

//product description

<?php

include('db.php');

$sql = "SELECT \* FROM products WHERE active\_status = 1";

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

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Our Products</title>

<link rel="stylesheet" href="styles.css">

</head>

<body>

<h1>Our Products</h1>

<input type="text" id="searchBox" placeholder="Search for products..." onkeyup="searchProducts()">

<div class="product-list" id="listProduct">

<?php while ($row = $result->fetch\_assoc()): ?>

<div class="product">

<img src="images/<?php echo $row['image']; ?>" alt="<?php echo $row['name']; ?>">

<h2><?php echo $row['name']; ?></h2>

<p><?php echo $row['description']; ?></p>

<div class="price"><?php echo $row['price']; ?> BDT</div>

<div class="stock">

<?php echo ($row['stock\_qty'] > 0) ? "In Stock" : "<span class='out-of-stock'>Out of Stock</span>"; ?>

</div>

</div>

<?php endwhile; ?>

</div>

<script>

function searchProducts() {

let input = document.getElementById("searchBox").value.toLowerCase();

let products = document.querySelectorAll(".product");

products.forEach(product => {

let name = product.querySelector("h2").textContent.toLowerCase();

product.style.display = name.includes(input) ? "block" : "none";

});

}

</script>

</body>

</html>

<?php $conn->close(); ?>

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

margin: 0;

padding: 0;

}

h1 {

text-align: center;

margin: 20px 0;

}

input#searchBox {

width: 300px;

margin: 10px auto;

display: block;

padding: 10px;

font-size: 16px;

}

.product-list {

display: flex;

flex-wrap: wrap;

justify-content: center;

gap: 20px;

}

.product {

background-color: white;

padding: 20px;

border-radius: 10px;

width: 250px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

text-align: center;

}

.product img {

width: 100%;

height: auto;

border-radius: 10px;

}

.product h2 {

font-size: 18px;

margin: 10px 0;

}

.product .price {

font-size: 20px;

font-weight: bold;

margin: 10px 0;

}

.product .stock {

font-size: 14px;

color: green;

}

.product .out-of-stock {

color: red;

}

.product p {

font-size: 14px;

color: #555;

}