**1. Requirements and Database Design**

**1.1 Expanding Customer Base**

* Goal: Develop a web shop application to reach a broader customer base.

**1.2 Database Design**

* Tables:
  + users (id, username, password, email, role)
  + products (id, name, description, price, image)
  + orders (id, user\_id, total\_price, created\_at)
  + order\_items (id, order\_id, product\_id, quantity, price)

**1.3 Time Estimation**

* Planning and Design: 2 days
* Database Design: 1 day
* Function Implementation: 5 days
* Testing and Debugging: 2 days
* Documentation: 1 day
* Total: approx. 11 days

**2. Sketch and GUI Design**

* Sketch: Draw a simple user interface for the web shop.
* GUI: Simple, user-friendly design without absolute positioning.

**3. Product Listing and Management**

* List of products with name, image, and price.
* Detailed view of a product with all properties.

**4. Admin Area**

* Admin login-logout.
* CRUD (Create, Read, Update, Delete) operations for products.
* Statistics on the most and least sold products.

**5. Security Measures**

* Prevent SQL injection using prepared statements.
* Password hashing (e.g., with password\_hash).
* Session management for admin login.
* Do not use root in the database, create a new user.

Checkout.php   
 **Login**: The script assumes the user is logged in because it fetches the user\_id from $\_SESSION['id']. However, it does not explicitly check if the user is logged in ($\_SESSION['id'] existence). Ensure that you have a mechanism to ensure the user is authenticated before proceeding with the checkout process.

 **Warenkorb (Cart)**: The script fetches cart items ($cartId from $\_GET['id']) and processes them. It calculates the total amount based on the items in the cart.

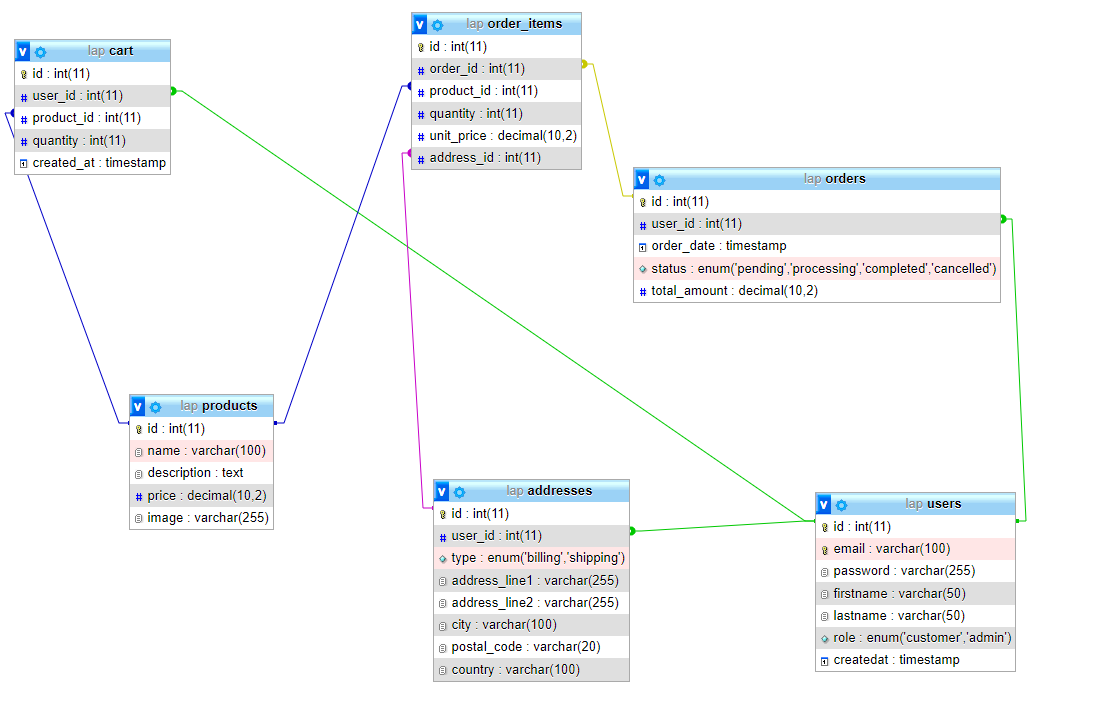
 **Bestellvorgang (Order Process)**:

* It validates and processes the address information ($\_POST data).
* It inserts orders into the database (orders table) and associates them with the user (user\_id).
* It inserts address information (addresses table) for shipping.
* It inserts order items (order\_items table) and removes them from the cart (cart table).
* It uses a transaction ($conn->begin\_transaction() and $conn->commit()) to ensure data consistency.

 **Rechnungserstellung und E-Mail-Versand (Invoice Generation and Email Sending)**:

* Generates an invoice number using generateInvoiceNumber() function.
* Sends an invoice email (sendInvoiceEmail() function) to the customer's provided email address ($\_POST['email']).

 **Error Handling**: The script uses try-catch blocks to handle exceptions during database transactions ($conn->rollback() on failure).



public function getLeastOrderedProducts()

    {

        $stmt = *$this*->conn->prepare("

            SELECT p.id, p.name, COUNT(oi.product\_id) AS order\_count

            FROM products p

            LEFT JOIN order\_items oi ON p.id = oi.product\_id

            GROUP BY p.id, p.name

            ORDER BY order\_count ASC

            LIMIT 5

        ");

        $stmt->execute();

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

        $products = $result->fetch\_all(MYSQLI\_ASSOC);

        $stmt->close();

        return $products;

    }

 public function getOrderHistoryLastFourWeeks()

    {

        $stmt = *$this*->conn->prepare("

          SELECT DATE\_FORMAT(o.order\_date, '%Y-%m-%d') AS order\_date, COUNT(\*) AS order\_count

          FROM orders o

          WHERE o.order\_date >= DATE\_SUB(NOW(), INTERVAL 4 WEEK)

          GROUP BY order\_date

          ORDER BY order\_date ASC

        ");

        $stmt->execute();

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

        $orderHistory = $result->fetch\_all(MYSQLI\_ASSOC);

        $stmt->close();

        return $orderHistory;

    }

 public function getMostOrderedProducts()

    {

        $stmt = *$this*->conn->prepare("

            SELECT p.id, p.name, COUNT(oi.product\_id) AS order\_count

            FROM products p

            LEFT JOIN order\_items oi ON p.id = oi.product\_id

            GROUP BY p.id, p.name

            ORDER BY order\_count DESC

            LIMIT 5

        ");

        $stmt->execute();

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

        $products = $result->fetch\_all(MYSQLI\_ASSOC);

        $stmt->close();

        return $products;

    }