

## Setting Up the Database and Local Clone:

### 1. Access phpMyAdmin:

- Open a web browser and go to <http://localhost/phpmyadmin>.

### 2. Create a New Database:

- Click on "Databases" in the top menu, then click "Create database."
- Name it (e.g., "ecommerce\_db") and choose "utf8\_general\_ci" as the collation.

### 3. Executing Database Scripts:

- In the "ecommerce\_db" database, select the "SQL" tab.

-- Table: Users

```
CREATE TABLE Users (  
    user_id INT PRIMARY KEY AUTO_INCREMENT,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50) NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,  
    password VARCHAR(255) NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

-- Table: Products

```
CREATE TABLE Products (  
    product_id INT PRIMARY KEY AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    description TEXT,  
    price DECIMAL(10, 2) NOT NULL,  
    stock_quantity INT NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

-- Table: Carts

```
CREATE TABLE Carts (  
    cart_id INT PRIMARY KEY AUTO_INCREMENT,  
    user_id INT NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (user_id) REFERENCES Users(user_id)  
);
```

-- Table: CartItems

```
CREATE TABLE CartItems (  
    cart_item_id INT PRIMARY KEY AUTO_INCREMENT,  
    cart_id INT NOT NULL,  
    product_id INT NOT NULL,  
    quantity INT NOT NULL,  
    FOREIGN KEY (cart_id) REFERENCES Carts(cart_id),  
    FOREIGN KEY (product_id) REFERENCES Products(product_id)  
);
```

-- Table: Orders

```
CREATE TABLE Orders (  
    order_id INT PRIMARY KEY AUTO_INCREMENT,  
    user_id INT NOT NULL,  
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    total_amount DECIMAL(10, 2) NOT NULL,  
    status ENUM('pending', 'shipped', 'delivered') DEFAULT 'pending',  
    FOREIGN KEY (user_id) REFERENCES Users(user_id)  
);
```

-- Table: OrderItems

```
CREATE TABLE OrderItems (  
    order_item_id INT PRIMARY KEY AUTO_INCREMENT,  
    order_id INT NOT NULL,  
    product_id INT NOT NULL,  
    quantity INT NOT NULL,  
    item_price DECIMAL(10, 2) NOT NULL,  
    total_price DECIMAL(10, 2) NOT NULL,  
    FOREIGN KEY (order_id) REFERENCES Orders(order_id),  
    FOREIGN KEY (product_id) REFERENCES Products(product_id)  
);
```

### *Cloning the Repository and Creating Branches:*

#### 1. Cloning the Repository:

- Run the following command to clone the repository:

```
git clone https://github.com/Adjanour/E-CommerceX
```

- Navigate into the cloned repository directory using `E-CommerceX`.

#### 2. Create a Development (Dev) Branch:

- Create a `dev` branch to serve as the staging area for integrating new features:

```
git checkout -b dev
```

### 3. Create Feature Branches:

- For each task or feature, create a separate branch:

`git checkout -b feature/feature-name`

### Creating Pull Requests:

#### 1. Work on Your Feature:

- Make changes to the code related to your feature within your feature branch.

#### 2. Commit and Push Changes:

- Commit your changes:

`git add .`

`git commit -m "Your commit message"`

`git push origin feature-name`

#### 3. Create a Pull Request:

- Go to the GitHub repository and click the "Pull Requests" tab.
- Click the "New pull request" button.

#### 4. Select Branches:

- Set the base branch to `dev` and the compare branch to your feature branch.
- Review the changes that will be merged.

## 5. Write a Description:

- Describe the purpose of your changes in the pull request description.
- Mention any relevant issues or context.

## 6. Request Reviewers:

- Request team members to review your pull request.
- Click "Create pull request."

## 7. Review and Discuss:

- Team members review your changes and provide feedback in the comments section.
- Address any feedback or suggested changes.

## 8. Merge the Pull Request:

- Once approved and conflicts resolved, merge your pull request into the `dev` branch.
- Click the "Merge pull request" button.