



الجامعة الافتراضية السورية
SYRIAN VIRTUAL UNIVERSITY

تقرير وظيفة برمجة مواقع 2

موقع متجر الكتروني

Website: https://abubakr-alsheikh.infinityfreeapp.com/	
الطالب	الصف
abubakr_203648	C3

المحتويات

1	مقدمة	2
1.1	نظرة عامة على المشروع	2
1.2	التقنيات المستخدمة	2
2	تصميم المشروع	3
2.1	بنية الموقع	3
2.2	تصميم قاعدة البيانات	4
2.3	تصميم واجهة المستخدم (UI)	7
3	التنفيذ	8
3.1	الصفحات الأساسية	8
3.2	هيكل وتنظيم الكود	13
3.3	تفاعل مع قاعدة البيانات	13
3.4	الوظائف الديناميكية	15
3.5	معالجة الأخطاء والأمان	16
4	التحديات والحلول	17
5	الخاتمة	18
5.1	ملخص المشروع	18
5.2	نتائج التعلم	19
5.3	تحسينات المستقبل	19

1 مقدّمة

1.1 نظرة عامة على المشروع

يهدف هذا المشروع إلى عرض تطوير موقع للتجارة الإلكترونية ديناميكي باستخدام مجموعة من تقنيات تطوير الواجهات الأمامية والخلفية. يعمل الموقع كمتجر إلكتروني فعال، على الرغم من أنه يعرض منتجات وهمية، وذلك لتلبية متطلبات هذه المهمة. يعرض الموقع الخصائص الأساسية لمنصة التجارة الإلكترونية النموذجية، مما يُمكن المستخدمين من تصفّح المنتجات وإدارة سلة التسوق ومحاكاة عملية تقديم الطلبات ومراجعتها.

تشمل الوظائف الرئيسية التي تم تنفيذها داخل الموقع ما يلي:

- **المنتجات الديناميكي:** يقدّم الموقع للمستخدمين منتجات يتم إنشاؤه ديناميكيًا، حيث يتم سحب معلومات المنتج مباشرةً من قاعدة بيانات MySQL.
- **سلة التسوق التفاعلية:** يمكن للمستخدمين إضافة المنتجات إلى سلة التسوق الخاصة بهم وضبط الكميات وإزالة العناصر ديناميكيًا دون الحاجة إلى إعادة تحميل الصفحات، مما يُحسّن تجربة المستخدم بشكل عام.
- **مصادقة المستخدم الآمنة:** تم تطبيق نظام تسجيل دخول وتسجيل مستخدم قوي، حيث يستخدم تجزئة كلمة المرور لضمان الأمان وتمكين المستخدمين من إدارة تفاصيل حساباتهم.
- **عملية دفع مبسطة:** ترشد عملية الدفع المستخدمين خلال عملية توفير معلومات الشحن و مراجعة ملخص الطلب قبل إتمام عملية الشراء.
- **إدارة الطلبات:** يمكن للمستخدمين الوصول إلى سجل طلباتهم ومراجعتها، مما يوفر سجلًا لمعاملاتهم السابقة.

1.2 التقنيات المستخدمة

يستخدم موقع التجارة الإلكترونية مجموعة متنوعة من التقنيات لتحقيق وظائفه وتصميمه المرئي:

- **الواجهة الأمامية: (Front-end)**
 - **HTML:** توفر الأساس الهيكلي للموقع.
 - **CSS:** تُضفي طابعًا على مظهر الموقع، مما يعزز جاذبيته البصرية.
 - **JavaScript:** تنفذ التفاعلات الديناميكية ووظائف AJAX لتجربة مستخدم سلسة.
 - **Bootstrap:** إطار عمل للواجهة الأمامية يضمن الاستجابة وتخطيطًا متسقًا عبر مختلف الأجهزة.
- **الواجهة الخلفية: (Back-end)**
 - **PHP:** لغة برمجة من جانب الخادم مسؤولة عن معالجة طلبات المستخدم والتفاعل مع قاعدة البيانات وإنشاء محتوى ديناميكي.
 - **قاعدة البيانات:**
 - **MySQL:** نظام إدارة قواعد البيانات العلائقية المستخدم لتخزين وإدارة معلومات المنتج وحسابات المستخدم وتفاصيل الطلب.

2 تصميم المشروع

2.1 بنية الموقع

تم تصميم موقع التجارة الإلكترونية ليوفر تدفقًا واضحًا ومنطقيًا للمستخدم، يوجه العملاء خلال التصفح، والاختيار، والشراء، وإدارة الطلبات. فيما يلي مخطط لإ صفحات الموقع الرئيسية وطريقة ارتباطها ببعضها البعض:

1. الصفحة الرئيسية: (index.php)

- تُعد الصفحة الرئيسية هي الصفحة الأولى التي يزورها الزوار.
- تُعرض مجموعة من المنتجات المميزة، يتم استرجاعها بشكل ديناميكي من قاعدة البيانات، مما يُشجع المستخدمين على استكشاف الموقع.
- تُوفر شريط التنقل الأساسي مع روابط لل أقسام الرئيسية: المنتجات، السلة، الحساب (إذا كان المستخدم قد سجل دخوله)، تسجيل الدخول/التسجيل.

2. صفحة المنتجات: (products.php)

- تُقدم قائمة شاملة لجميع المنتجات المتاحة، يتم استرجاعها بشكل ديناميكي من قاعدة البيانات.
- تستخدم تنسيقًا شبكيًا لعرض المنتجات بشكل فعال.
- يشتمل كل إعلان عن منتج على صورة مصغرة، اسم المنتج، وصفًا مختصرًا، السعر، وزر "إضافة إلى السلة".
- تُوفر روابط لصفحة تفاصيل المنتج (product-details.php) للحصول على معلومات إضافية.

3. صفحة تفاصيل المنتج: (product-details.php)

- تُعرض معلومات تفصيلية عن منتج معين، يتم استرجاعها بشكل ديناميكي من قاعدة البيانات بناءً على معرف المنتج المرسل من خلال عنوان URL.
- تُضمّن صورًا للمنتج، وصفًا شاملاً، السعر، معلومات المخزون المتاحة، وزر "إضافة إلى السلة" للشراء.

4. صفحة السلة: (cart.php)

- تُقدم ملخصًا لسلة تسوق المستخدم، وتُدرج المنتجات المختارة، وكمياتها، وتكاليفها الفردية والإجمالية.
- تُتيح للمستخدمين تحديث كميات المنتجات أو إزالة منتجات بعينها. يتم تحديث السلة بشكل ديناميكي باستخدام AJAX.
- تُوفر زر "متابعة إلى عملية الدفع"، الذي يُوجّه المستخدمين إلى صفحة الدفع.

5. صفحة الدفع: (checkout.php)

- تُرشد المستخدمين خلال عملية إتمام الطلب.
- تُجمع معلومات الشحن من خلال نموذج، وتُضمن تقديم الحقول المطلوبة.
- تُقدم ملخصًا تفصيليًا للطلب للمراجعة، بما في ذلك تكاليف الشحن (إن وجدت) والإجمالي النهائي.
- عند إرسال النموذج بنجاح، تُعالج الطلب، وتُحدث المخزون، وتؤكد عملية الشراء.

6. صفحة الحساب: (account.php)

- يمكن الوصول إليها للمستخدمين الذين قاموا بتسجيل الدخول.
- تُعرض سجل طلبات المستخدم، وتُقدم تفاصيل مثل تاريخ الطلب، المبلغ الإجمالي، وحالة الطلب.

○ قد تشمل خيارات لإدارة تفاصيل الحساب (مثل تحديث عنوان الشحن).

7. صفحات تسجيل الدخول/التسجيل: (login.php, register.php)

- تُسهّل مصادقة المستخدم وإنشاء الحسابات.
- تُتيح صفحة تسجيل الدخول للمستخدمين الحاليين الوصول إلى حساباتهم.
- تُتيح صفحة التسجيل للمستخدمين الجدد إنشاء حساب، وتُجمع المعلومات الضرورية مثل اسم المستخدم، البريد الإلكتروني، وكلمة المرور.

ترتبط هذه الصفحات ببعضها البعض من خلال مجموعة من الروابط التشعبية في شريط التنقل وأزرار تُوجّه المستخدمين بناءً على تصرفاتهم (مثل "إضافة إلى السلة"، "متابعة إلى عملية الدفع"). يضمن هذا النهج المُنظّم تجربة سهلة الاستخدام، ويُتيح للعملاء التنقل عبر موقع التجارة الإلكترونية بشكل بديهي وإكمال مشترياتهم بكفاءة.

2.2 تصميم قاعدة البيانات

يعتمد موقع التجارة الإلكترونية على قاعدة بيانات علائقية MySQL لتخزين وإدارة بياناته. يتم تنظيم قاعدة البيانات في أربعة جداول رئيسية، تم تصميمها بعناية للحفاظ على سلامة البيانات وتسهيل استرجاعها بكفاءة. فيما يلي الجداول وعلاقاتها:

1. جدول products:

الوصف	نوع البيانات	العمود
المفتاح الأساسي: معرف فريد لكل منتج	INT AUTO_INCREMENT	id
اسم المنتج	VARCHAR(255)	name
وصف مفصل للمنتج	TEXT	description
سعر المنتج	DECIMAL(10, 2)	price
مسار الملف أو عنوان URL لصورة المنتج	VARCHAR(255)	image
علامة تشير إلى ما إذا كان المنتج مميزاً (1 أو 0)	TINYINT(1)	featured
عدد الوحدات المتوفرة في المخزون للمنتج	INT	stock

2. جدول users:

الوصف	نوع البيانات	العمود
المفتاح الأساسي: معرف فريد لكل حساب مستخدم	INT AUTO_INCREMENT	id

username	VARCHAR(255)	اسم مستخدم فريد لتسجيل دخول المستخدم
email	VARCHAR(255)	عنوان بريد إلكتروني فريد مرتبط بحساب المستخدم
password_hash	VARCHAR(255)	كلمة مرور مشفرة للتخزين الآمن والمصادقة

3. جدول orders:

الوصف	نوع البيانات	العمود
المفتاح الأساسي: معرف فريد لكل طلب	INT AUTO_INCREMENT	id
مفتاح أجنبي: يرتبط بالمستخدم الذي قام بتقديم الطلب (من users)	INT	user_id
الطابع الزمني الذي يشير إلى وقت تقديم الطلب	TIMESTAMP	order_date
المبلغ الإجمالي للطلب	DECIMAL(10, 2)	total
اسم مستلم الشحنة	VARCHAR(255)	shipping_name
عنوان الشحن للطلب	VARCHAR(255)	shipping_address

4. جدول order_items:

الوصف	نوع البيانات	العمود
المفتاح الأساسي: معرف فريد لكل عنصر في الطلب	INT AUTO_INCREMENT	id
مفتاح أجنبي: يرتبط بالطلب الذي ينتمي إليه العنصر (من orders)	INT	order_id
مفتاح أجنبي: يرتبط بالمنتج الذي يتم طلبه (من products)	INT	product_id
عدد وحدات المنتج المطلوبة	INT	quantity
سعر المنتج وقت الطلب (لسجلات تاريخية دقيقة)	DECIMAL(10, 2)	price

العلاقات بين الجداول:

- واحد إلى متعدد (**users إلى orders**) : يمكن أن يكون للمستخدم طلبات متعددة، لكن كل طلب ينتمي إلى مستخدم واحد فقط. يتم تأسيس هذه العلاقة باستخدام المفتاح الأجنبي `user_id` في جدول `orders`.
- واحد إلى متعدد (**orders إلى order_items**) : يمكن أن يحتوي الطلب على عناصر طلب متعددة، لكن كل عنصر طلب ينتمي إلى طلب واحد فقط. يتم تحقيق ذلك باستخدام المفتاح الأجنبي `order_id` في جدول `order_items`.
- واحد إلى متعدد (**products إلى order_items**) : يمكن أن يكون المنتج جزءًا من عناصر طلب متعددة، لكن كل عنصر طلب يشير إلى منتج محدد. يؤسس المفتاح الأجنبي `product_id` في جدول `order_items` هذه العلاقة.

كود SQL لإنشاء الجداول:

-- Products Table

```
CREATE TABLE `products` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `name` VARCHAR(255) NOT NULL,
  `description` TEXT,
  `price` DECIMAL(10, 2) NOT NULL,
  `image` VARCHAR(255),
  `featured` TINYINT(1) DEFAULT 0, -- 1 for featured, 0 for not featured
  `stock` INT DEFAULT 0
);
```

-- Users Table

```
CREATE TABLE `users` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `username` VARCHAR(255) NOT NULL UNIQUE,
  `email` VARCHAR(255) NOT NULL UNIQUE,
  `password_hash` VARCHAR(255) NOT NULL
);
```

-- Orders Table

```
CREATE TABLE `orders` (
  `id` INT AUTO_INCREMENT PRIMARY KEY,
  `user_id` INT,
```

```
`order_date` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
`total` DECIMAL(10, 2) NOT NULL,
`shipping_name` VARCHAR(255),
`shipping_address` VARCHAR(255),
FOREIGN KEY (`user_id`) REFERENCES `users`(`id`)
);
```

-- Order Items Table

```
CREATE TABLE `order_items` (
`id` INT AUTO_INCREMENT PRIMARY KEY,
`order_id` INT,
`product_id` INT,
`quantity` INT NOT NULL,
`price` DECIMAL(10, 2) NOT NULL,
FOREIGN KEY (`order_id`) REFERENCES `orders`(`id`),
FOREIGN KEY (`product_id`) REFERENCES `products`(`id`)
);
```

2.3 تصميم واجهة المستخدم (UI)

يُركز موقع التجارة الإلكترونية على تقديم واجهة مستخدم نظيفة وعصرية وسهلة الاستخدام، ويهدف إلى توفير تجربة تسوق ممتعة وكفاءة. وتشمل جوانب تصميم واجهة المستخدم الرئيسية ما يلي:

- **التوافق مع جميع الأجهزة:** تم بناء الموقع باستخدام إطار عمل Bootstrap، مما يضمن تخطيطاً متسقاً وعرضاً مثاليًا على مجموعة متنوعة من الأجهزة، من أجهزة الكمبيوتر المكتبية إلى الأجهزة اللوحية والهواتف المحمولة. يتم استخدام نظام الشبكة في Bootstrap وخصائص الاستجابة للتكيف مع المحتوى ديناميكيًا مع أحجام شاشة مختلفة.
- **الجاذبية البصرية:** يتم استخدام لوحة ألوان بسيطة وجذابة بصريًا في جميع أنحاء الموقع، مما يخلق جمالية متناغمة وممتعة. تُستخدم صور المنتجات عالية الجودة لتعزيز العرض البصري للمنتجات.
- **التنقل:** يوجد شريط تنقل بارز وسهل الفهم في جميع الصفحات، مما يتيح للمستخدمين الوصول بسهولة إلى أقسام الموقع المختلفة. تُستخدم تسميات واضحة وموجزة لروابط التنقل، مما يسهل على المستخدمين العثور على ما يبحثون عنه.
- **عرض المنتجات:** يتم عرض المنتجات في تخطيط شبكي منظم بشكل جيد في صفحة المنتجات، مما يوفر نظرة عامة جذابة بصريًا. تتضمن كل قائمة منتج معلومات أساسية ودعوة واضحة للعمل ("أضف إلى السلة").

- **العناصر التفاعلية:** تساهم العناصر التفاعلية، مثل تحديث السلة الديناميكي باستخدام AJAX وإشعارات toast ، في توفير تجربة مستخدم أكثر إثارة للاهتمام واستجابة. تُستخدم مؤشرات التحميل لإظهار إجراءات المعالجة، مما يوفر ملاحظات بصرية للمستخدمين.
- **إمكانية الوصول:** تُبذل الجهود لضمان إمكانية الوصول إلى الموقع لمجموعة واسعة من المستخدمين. على سبيل المثال، يتم توفير نص بديل مناسب للصور لمساعدة مستخدمي قارئ الشاشة.
- بشكل عام، يركز تصميم واجهة المستخدم على الوضوح والبساطة وسهولة الاستخدام، مما يخلق بيئة متجر عبر الإنترنت تشجع على الاستكشاف واكتشاف المنتج و عملية شراء سلسة.

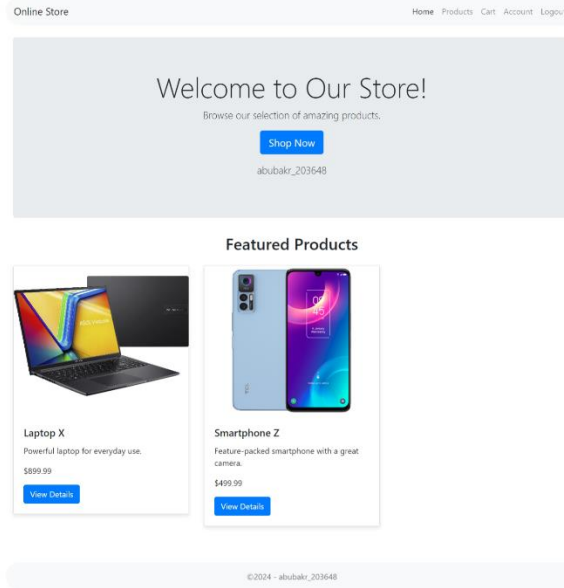
3 التنفيذ

3.1 الصفحات الأساسية:

يُدمج موقع التجارة الإلكترونية هذا ميزات أساسية لتوفير تجربة تسوق شاملة عبر الإنترنت:

1. الصفحة الرئيسية:

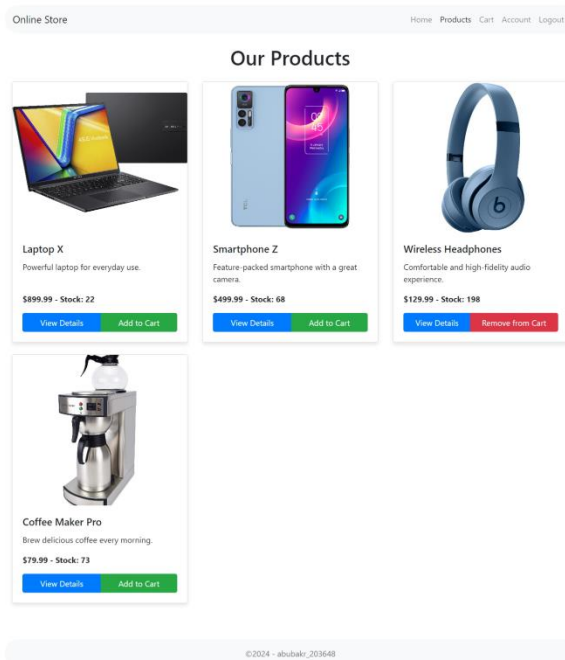
- **عرض المنتجات المميزة:** تُستعرض الصفحة الرئيسية مجموعة مختارة من المنتجات المميزة التي يتم استردادها ديناميكياً من قاعدة البيانات. يتم تحقيق ذلك من خلال الاستعلام عن جدول products بحثاً عن الإدخالات التي تكون فيها قيمة عمود featured مساوياً لـ 1 (صحيح).
- **تنفيذ التنقل:** يُدرج شريط تنقل بارز في الصفحة الرئيسية، ويوفر روابط واضحة للوصول إلى صفحات رئيسية أخرى (المنتجات، السلة، الحساب، تسجيل الدخول/التسجيل). يضمن هذا شريط التنقل سهولة الوصول إلى أقسام مختلفة من الموقع، ويشجع المستخدمين على الاستكشاف.



2. قائمة المنتجات:

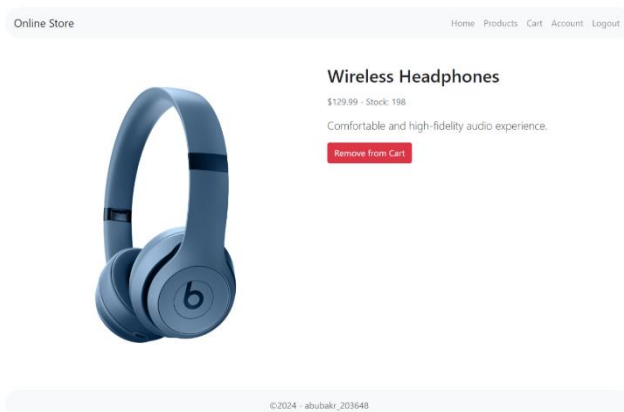
- **التحميل الديناميكي:** تُحمل صفحة "قائمة المنتجات" وتُعرض المنتجات ديناميكياً من قاعدة البيانات. يقوم سكريبت PHP بالاستعلام عن جدول products واسترجاع المعلومات ذات الصلة (الاسم، الصورة، السعر، إلخ). يتم

استخدام هذه البيانات بعد ذلك لتوليد قوائم المنتجات ديناميكياً داخل تخطيط شبكي HTML ، مما يضمن تحديث الموقع بشكل مستمر بمحتوى قاعدة البيانات.



3. تفاصيل المنتج:

- عرض المعلومات المحددة :عند النقر على قائمة منتج، يتم توجيه المستخدمين إلى صفحة "تفاصيل المنتج". تسترد هذه الصفحة معلومات المنتج المحددة من قاعدة البيانات بناءً على معرف المنتج الذي يتم تمريره عبر عنوان URL. يستخدم سكريبت PHP عبارة مُعدة (لأمن) لاستعلام عن جدول products باستخدام المعرف المُعطى، ثم يتم عرض التفاصيل المُسترجعة.
- منطق إضافة المنتج إلى السلة :يسمح زر "أضف إلى السلة" الموجود في صفحة "تفاصيل المنتج" للمستخدمين بإضافة المنتج المحدد إلى سلة التسوق الخاصة بهم. يُطلق هذا الإجراء طلب AJAX إلى سكريبت جانب الخادم (add_to_cart.php)، الذي يتعامل مع إضافة المنتج إلى السلة، وتحديث كميات السلة، وتقديم ملاحظات للمستخدم.




4. سلة التسوق:

- إدارة عناصر السلة: توفر صفحة "السلة" وظائف لإضافة، إزالة، وتحديث عناصر السلة. يتم التعامل مع هذه الإجراءات من خلال طلبات AJAX إلى الخادم، مما يضمن تجربة مستخدم سلسة.
- عرض الإجماليات: تحسب سلة التسوق وتُعرض إجمالي المبلغ، وتكاليف الشحن (إن وجدت)، وإجمالي المبلغ النهائي للطلب. يتم تحديث هذه الإجماليات في الوقت الفعلي عندما يُعدّل المستخدمون محتويات سلة التسوق الخاصة بهم.

Online Store Home Products Cart Account Logout

Your Shopping Cart

Product	Price	Quantity	Subtotal	Actions
 Wireless Headphones	\$129.99	<input type="text" value="1"/>	\$129.99	Remove

Order Summary

Subtotal:	\$129.99
Shipping:	\$0.00
Total:	\$129.99

[Proceed to Checkout](#)

©2024 - abubakr_203648

5. مصادقة المستخدم:

- تسجيل وتسجيل دخول المستخدمين: يتم تنفيذ مصادقة المستخدم وتسجيل الدخول الآمن باستخدام PHP و MySQL. تقوم عملية التسجيل بتجزئة كلمات مرور المستخدم باستخدام bcrypt قبل تخزينها في قاعدة البيانات، مما يضيف طبقة من الأمان.
- التعامل مع كلمات المرور بشكل آمن: لا يتم تخزين كلمات المرور أبدًا في نص عادي؛ يتم تجزئتها باستخدام خوارزمية تجزئة قوية (bcrypt) لحماية البيانات الحساسة للمستخدم.

Online Store Home Products Cart Login Register

Login

Username:

Password:

[Login](#)

If you don't have account [Register](#)

©2024 - abubakr_203648

Online Store Home Products Cart Login Register

Register

Username:

Email:

Password:

Confirm Password:

[Register](#)

If you have an account [Login](#)

©2024 - abubakr_203648

id	username	email	password_hash
1	asdf	asdf1@asdf.com	\$2y\$10\$uSfRUzi.FlusbQAbUml0F.zrtQZhRDHXjEmw6JPwam...
4	Abubakr	abubakr_203648@onlinestore.com	\$2y\$10\$23vWcocOfouwfbOFQ3wNrOdFYRYnlok2jVvDCujr6wz...

6. عملية الدفع:

- جمع تفاصيل الشحن: تستخدم صفحة الدفع نموذجًا لجمع معلومات الشحن للمستخدم، بما في ذلك الاسم، العنوان، وأي تفاصيل أخرى ضرورية.
- إتمام الطلبات: عند إرسال النموذج بنجاح، تقوم عملية الدفع بإدراج سجل طلب جديد في جدول orders، مع العناصر المُقابلة للطلب في جدول order_items.
- تحديث المخزون: بعد إتمام الطلب، يتم تقليل المخزون المتاح للمنتجات المشتراة بشكلٍ مناسب في جدول products لمخزون المُحدَّث.

Online Store Home Products Cart Account Logout

Checkout

Shipping Address

Full Name

Address

Place Order

Your Cart 1

Wireless Headphones

Quantity: 1

\$129.99

Subtotal

\$129.99

Shipping

\$0.00

Total

\$129.99

©2024 - abubakr_203648

7. تأكيد الطلب:

- عرض تفاصيل الطلب: بعد عملية شراء ناجحة، يتم إعادة توجيه المستخدمين إلى صفحة تأكيد الطلب. تُعرض هذه الصفحة ملخصًا للطلب، بما في ذلك رقم الطلب، العناصر المشتراة، عنوان الشحن، وإجمالي المبلغ المدفوع. يعمل تأكيد الطلب كسجل للعملية للمستخدم.
- تأكيد عملية شراء ناجحة: تقدم صفحة تأكيد الطلب ملاحظات واضحة للمستخدم، مؤكدةً أن الطلب قد تم معالجته بنجاح. قد تتضمن أيضًا معلومات إضافية، مثل أوقات التسليم المُقدَّرة أو تعليمات تتبع الطلب.

Online Store Home Products Cart Account Logout

Order Confirmation

Order Details

Product	Price	Quantity	Subtotal
Wireless Headphones	\$129.99	1	\$129.99
Total:			\$129.99

Shipping Address

Full Name: Alipoo

Address: al-jamiliyeh

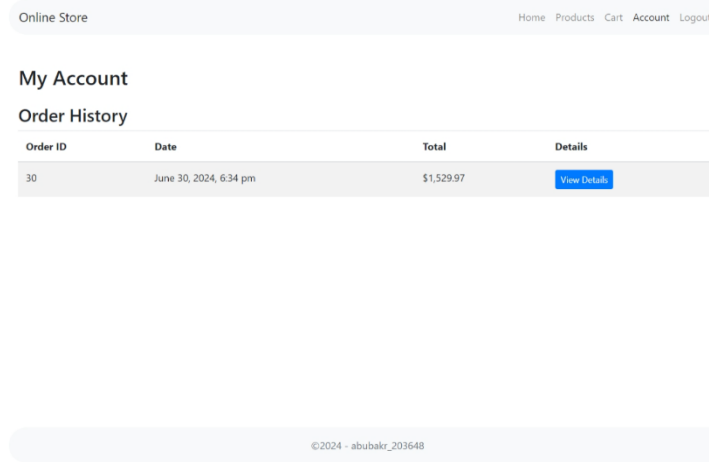
Continue Shopping

View your orders

©2024 - abubakr_203648

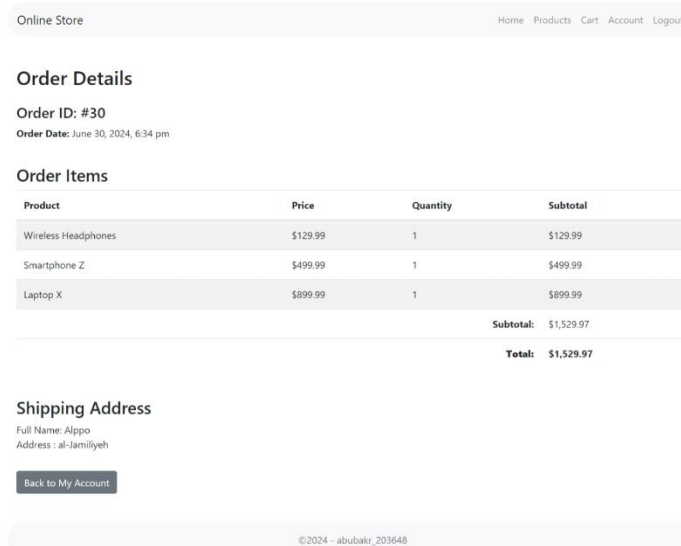
8. صفحة الحساب:

- عرض سجل الطلبات: تعرض صفحة الحساب، المتاحة للمستخدمين المسجلين فقط، سجلاً شاملاً لطلبات المستخدم السابقة. يتضمن هذا التاريخ تفاصيل أساسية لكل طلب، مثل تاريخ الطلب والمبلغ الإجمالي ورابط لعرض مزيد من المعلومات التفصيلية في صفحة تفاصيل الطلب.
- إدارة الطلبات: يمكن للمستخدمين مراجعة محتويات طلباتهم السابقة، وربما يكون لديهم خيارات لإعادة طباعة الفواتير أو الاتصال بدعم العملاء بخصوص طلبات معينة.



9. صفحة تفاصيل الطلب:

- تفكيك تفصيلي للطلب: توفر هذه الصفحة عرضاً تفصيلياً لطلب معين يتم اختياره من قبل المستخدم من صفحته. تقدم قائمة مفصلة بالمنتجات المشتراة، والكميات، والأسعار، والتكلفة الإجمالية للطلب، بما في ذلك رسوم الشحن (إن وجدت).
- معلومات الشحن والفواتير: تعرض الصفحة أيضاً عنوان الشحن وتفاصيل الفواتير المستخدمة للطلب، مما يسمح للمستخدمين بتأكيد صحة المعلومات أو تحديثها لعمليات الشراء المستقبلية.



3.2 هيكل وتنظيم الكود

يُعد الحفاظ على قاعدة كود نظيفة ومنظمة أمرًا ضروريًا لنجاح وصيانة أي تطبيق ويب على المدى الطويل. يركز موقع التجارة الإلكترونية هذا على إمكانية إعادة استخدام الكود والتنظيم المنطقي من خلال الممارسات التالية:

- **دليل التضمينات المركزي (includes/):** مكونات شائعة الاستخدام، مثل:
 - **اتصال قاعدة البيانات (database.php):** يُغطي منطق اتصال قاعدة البيانات لتجنب التكرار عبر الملفات. يُعرّف هذا الملف فئة Database مع طرق لإنشاء اتصال بقاعدة البيانات، وتنفيذ الاستعلامات، واسترجاع النتائج.
 - **الرأس (header.php):** يحتوي على لغة HTML للرأس الخاص بالموقع، بما في ذلك التنقل، ويُدرج في كل صفحة للحفاظ على التناسق.
 - **القدم (footer.php):** يحتوي على لغة HTML لقدم الموقع، بما في ذلك معلومات حقوق النشر، ويُدرج في كل صفحة.
- **استخدام عبارات تضمين PHP:** تُستخدم عبارة include على نطاق واسع لدمج الرأس والقدم والمكونات المشتركة الأخرى في ملفات PHP الفردية. يُعزز هذا إعادة استخدام الكود ويُسهل تحديث العناصر عالميًا.
- **فصل الاهتمامات:** تتم هيكلة قاعدة الكود لفصل المخاوف المختلفة، مثل منطق العرض (HTML)، (CSS)، والمنطق التجاري (PHP)، ووصول البيانات (تفاعلات قاعدة البيانات). يُعزز هذا الفصل وضوح الكود وصيانتته.
- **أسماء الملفات والمتغيرات ذات المغزى:** تُستخدم اتفاقيات التسمية الوصفية والمتسقة للملفات والدوال والمتغيرات، مما يحسّن قابلية قراءة الكود وفهمها.
- **التعليقات والتوثيق:** تُضاف تعليقات ذات صلة إلى الكود لشرح المنطق المعقد أو غرض أقسام الكود المحددة. على الرغم من عدم إدراجها في هذا المثال، يمكن إنشاء توثيق مفصل لتقديم نظرة عامة على هيكل المشروع ووظائفه.

مثال للتطبيق :

```
// index.php
<?php
    include "includes/header.php";
    // ... PHP logic for the homepage
    include "includes/footer.php";
?>
```

في هذا المثال، index.php يُدرج الرأس والقدم من دليل includes/، مما يضمن مظهرًا متناسقًا في جميع أنحاء الموقع. يُحتوى المنطق الخاص بصفحة البداية داخل index.php، مما يحافظ على تنظيم الكود وإدارته.

من خلال اتباع هذه الممارسات، تظل قاعدة كود موقع التجارة الإلكترونية منظمة بشكل جيد، مما يسهل فهمها وصيانتها وتوسيعها بميزات أو تعديلات جديدة في المستقبل.

3.3 تفاعل مع قاعدة البيانات

يعتمد موقع التجارة الإلكترونية بشكل كبير على التفاعل السلس مع قاعدة بيانات MySQL لاسترجاع معلومات المنتج وإدارة حسابات المستخدمين ومعالجة الطلبات والمزيد. توفر لغة PHP الأدوات اللازمة لإنشاء هذا الاتصال وتنفيذ عمليات قاعدة البيانات بأمان وكفاءة.

1. اتصال قاعدة البيانات:

- يوجد ملف PHP مخصص (includes/database.php) يُخزن منطق اتصال قاعدة البيانات. يُسهّل هذا النهج إدارة وتحديث تكوين الاتصال.
- يُستخدم ملحق PDO كائنات بيانات PHP لضمان الأمان والمرونة. يوفر PDO واجهة متسقة للتفاعل مع أنظمة قواعد البيانات المختلفة.

```
class Database {

    public static function connect() {
        if (!isset(self::$conn)) {
            try {
                self::$conn = new PDO(
                    "mysql:host=" . self::$servername . ";dbname=" .
self::$dbname,
                    self::$username,
                    self::$password
                );
                self::$conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
            } catch (PDOException $e) {
                echo "Connection failed: " . $e->getMessage();
            }
        }
        return self::$conn;
    }
}
```

2. تنفيذ الاستعلامات واسترجاع البيانات:

- توفر فئة Database في database.php طريقة (مثل query()) لتنفيذ استعلامات SQL واسترجاع النتائج.

```
class Database {

    public static function query($sql, $params = []) {
        try {
            $stmt = self::connect()->prepare($sql);
            $stmt->execute($params);
            return $stmt->fetchAll(PDO::FETCH_ASSOC);
        } catch (PDOException $e) {
            // error handling
        }
    }
}
```

3. منع حقن SQL باستخدام عبارات مُعدة:

- تعد العبارات المُعدة ضرورية لمنع ثغرات حقن SQL.
- في عبارة مُعدة، يُرسل استعلام SQL إلى خادم قاعدة البيانات ويتم/عده دون أي بيانات من جانب المستخدم.
- يتم استخدام علامات وضع مكان (عادةً ما يتم تمثيلها بعلامة استفهام (?) للأجزاء من الاستعلام التي ستستقبل إدخال المستخدم.
- ثم يقوم خادم قاعدة البيانات بتحليل وإعداد الاستعلام للتنفيذ، ولكنه لا يُنفذه بعد.
- يُنفصل عن ذلك، يتم إرسال القيم من جانب المستخدم إلى خادم قاعدة البيانات، يقوم الخادم بتعويض هذه القيم بأمان في علامات وضع مكان في العبارة المُعدة قبل التنفيذ.

مثال (استرجاع تفاصيل المنتج):

```
$product = Database::query("SELECT * FROM products WHERE id = ?",  
[$productId]);  
  
if ($product) {  
    // Access product data safely:  
    echo $product[0]['name'];  
}
```

يضمن هذا الفصل بين بنية استعلام SQL وبيانات المستخدم عدم إمكانية حقن شفرة ضارة في الاستعلام، ما يجعل التطبيق أكثر أمانًا بكثير.

من خلال تجميع منطق قاعدة البيانات واستخدام PDO وتطبيق العبارات المُعدة بشكل متنسق، يقوم موقع التجارة الإلكترونية بإنشاء إطار عمل آمن وفعال للتفاعل مع قاعدة البيانات، ويحمي من الثغرات ويضمن سلامة البيانات.

3.4 الوظائف الديناميكية

يُدمج موقع التجارة الإلكترونية عناصر ديناميكية لتحسين تجربة المستخدم وتوفير واجهة مستخدم أكثر استجابة وجاذبية. تُقلل هذه العناصر من إعادة تحميل الصفحات وتوفر ملاحظات فورية على إجراءات المستخدم:

1. AJAX لتحديثات العربية:

يلعب AJAX دورًا حاسمًا في إنشاء تجربة تسوق سلسلة وتفاعلية للعربية. إليك كيفية تنفيذه:

- **إجراء إضافة إلى العربية:** عندما ينقر المستخدم على زر "إضافة إلى العربية" على صفحة تفاصيل المنتج، يتم إرسال طلب AJAX إلى البرنامج النصي على جانب الخادم (add_to_cart.php) يحمل هذا الطلب معرف المنتج والإجراء المطلوب (إضافة أو إزالة).
- **معالجة جانب الخادم:** يعالج البرنامج النصي على جانب الخادم الطلب:
 - يضيف المنتج إلى عربة جلسة المستخدم.
 - إذا كان المنتج موجودًا بالفعل في العربية، يتم تحديث كميته.
 - يعيد استجابة JSON إلى JavaScript على جانب العميل، مما يشير إلى النجاح أو الفشل ويوفر معلومات محدثة عن العربية (مثل إجمالي العناصر، المجموع الفرعي).

- تحديث جانب العميل: تُعالج شفرة JavaScript التي أرسلت طلب AJAX الاستجابة:

- تُحدث عرض العربية (مثل رمز العربية، محتويات صفحة العربية) دون إعادة تحميل الصفحة بأكملها.
- يمكنها اختيارياً تقديم ملاحظات بصرية للمستخدم، مثل إشعار بأن العنصر قد تم إضافته إلى العربية.

2. عناصر ديناميكية أخرى:

- إشعارات التوست: تُعد التوست إشعارات خفيفة الوزن ومؤقتة تُعرض على الموقع الإلكتروني لتوفير ملاحظات للمستخدم. على سبيل المثال، يؤدي إضافة منتج إلى العربية إلى إطلاق توست "نجاح"، بينما قد يُعرض توست "خطأ" عند حدوث خطأ. تعزز هذه الدلائل البصرية تجربة المستخدم من خلال تأكيد الإجراءات أو الإشارة إلى المشكلات.
- أيقونات التحميل: تُستخدم أيقونات التحميل للإشارة إلى أن عملية ما جارية، مثل إضافة عنصر إلى العربية أو تحديث كميات العربية. يوفر عرض أيقونة التحميل ملاحظات بصرية للمستخدم، ويمنع الارتباك ويشير إلى أن الموقع الإلكتروني يُعالج طلبه. عندما تكتمل الإجراءات، تختفي أيقونة التحميل.

مثال: إضافة منتج إلى العربية باستخدام AJAX

```
$('.add-to-cart').click(function() {
    const productId = $(this).data('product-id');

    $.ajax({
        url: 'add_to_cart.php',
        method: 'POST',
        data: { productId: productId, action: 'add' }, // Pass data to the server
        dataType: 'json', // Expect JSON response
        success: function(response) {
            if (response.success) {
                // Update cart count display
                // Display success toast
            } else {
                // Display error message (e.g., out of stock)
            }
        },
        error: function() {
            // Handle AJAX errors, display a generic error message
        }
    });
});
```

تساهم هذه الوظائف الديناميكية بشكل كبير في تجربة مستخدم إيجابية واستجابة على موقع التجارة الإلكترونية.

3.5 معالجة الأخطاء والأمان

يُعدّ الأمان ومعالجة الأخطاء أمراً بالغ الأهمية لأي تطبيق ويب، خاصةً التطبيقات التي تتعامل مع بيانات المستخدم الحساسة والمعاملات المالية. يتخذ موقع التجارة الإلكترونية هذا العديد من الإجراءات لمواجهة الثغرات الأمنية الشائعة وتقديم تجربة آمنة وموثوقة للمستخدمين:

1. منع حقن SQL :

- **العبارات المُعدّة:** تستخدم عبارات SQL المُعدّة لجميع استعلامات قاعدة البيانات. تُفصل هذه التقنية بين بنية استعلام SQL والقيم الفعلية، مما يُمنع حقن الكود الضار. تُعامل القيم كبيانات، وليس كجزء من الاستعلام نفسه، مما يُحظر بفعالية هجمات حقن SQL.
- **التحقق من صحة الإدخال وتطهيره:** قبل استخدام إدخال المستخدم في استعلامات قاعدة البيانات، يتم التحقق من صحة البيانات وتطهيرها بعناية. يتضمن ذلك التأكد من أن البيانات تتوافق مع التنسيق المتوقع (مثل التأكد من أن عناوين البريد الإلكتروني تتبع تنسيقًا صحيحًا) وإزالة أي أحرف ضارة.

2. معالجة كلمة المرور الآمنة:

- **تجزئة كلمة المرور:** لا تُخزن كلمات مرور المستخدمين أبدًا كبيانات نصية عادية. بدلاً من ذلك، يتم تجزئتها باستخدام خوارزمية تجزئة قوية وغير قابلة للرجوع، مثل bcrypt. هذا يعني أنه حتى إذا تم اختراق قاعدة البيانات، فلن يمكن استرجاع كلمات المرور الفعلية.

3. معالجة الأخطاء:

- **رسائل:** إذا فشلت عملية الطلب بسبب نقص المخزون، يتم عرض رسالة واضحة للمستخدم توضح المشكلة.
- **تسجيل الأخطاء:** يتم تسجيل الأخطاء أيضًا في ملف لغرض تصحيح الأخطاء وتتبعها. يساعد ذلك في تحديد وإصلاح أي مشكلات متكررة أو ثغرات أمنية محتملة.
- **تجنب الكشف عن المعلومات الحساسة:** يُفترض تجنب الكشف عن معلومات حساسة للغاية في رسائل الخطأ. على سبيل المثال، يتم عرض رسالة خطأ عامة بدلاً من رمز خطأ محدد، مما يُمنع المهاجمين من الحصول على رؤى حول عمل النظام الداخلي.

4 التحديات والحلول

تضمن تطوير موقع التجارة الإلكترونية هذا العديد من التحديات التقنية، مما أتاح فرصًا قيمة للتعلم ورؤى حول أفضل ممارسات تطوير الويب. فيما يلي بعض التحديات الرئيسية التي واجهتها والحلول التي تم تنفيذها للتغلب عليها:

التحدي الأول: تنفيذ تقنية AJAX لوظائف سلة التسوق

المشكلة:

كان تحديث سلة التسوق ديناميكيًا دون الحاجة إلى إعادة تحميل الصفحة أمرًا صعبًا في البداية. تطلب فهم كيفية تنظيم طلبات AJAX ، وإرسال البيانات إلى الخادم، ومعالجة الطلبات في PHP ، وتحديث الصفحة ديناميكيًا باستخدام JavaScript بحثًا وتجربة متعمقين.

الحل:

- **البحث الشامل:** تمت استشارة البرامج التعليمية عبر الإنترنت، والوثائق، ونماذج التعليمات البرمجية حول تقنية AJAX ولغة PHP لفهم الأساسيات.
- **التنفيذ خطوة بخطوة:** قمت بتقسيم تنفيذ AJAX إلى خطوات أصغر قابلة للإدارة: إنشاء طلب AJAX ، والتعامل معه في برنامج PHP منفصل، وإرسال رد إلى العميل، وتحديث الصفحة ديناميكيًا بناءً على الرد.
- **التصحيح والاختبار:** كانت أدوات مطوري البرامج في المتصفح (وخاصة علامة تبويب الشبكة) لا تقدر بثمن لمراقبة طلبات AJAX ، وتحليل الردود، وتصحيح أي مشكلات.

التحدي الثاني: ضمان سلامة البيانات أثناء عملية الدفع

المشكلة:

تتضمن عملية الدفع تحديثات متعددة لقاعدة البيانات (إنشاء طلب، وإدخال عناصر الطلب، وتحديث مخزون المنتجات). كان ضمان حدوث هذه التحديثات بشكل متسق والحفاظ على سلامة البيانات، حتى في حالة حدوث أخطاء، أمرًا بالغ الأهمية.

الحل:

- **معاملات قاعدة البيانات:** تم تنفيذ معاملات MySQL لتجميع عمليات قاعدة البيانات المتعددة ضمن عملية دفع واحدة في وحدة عمل واحدة. وقد ضمن ذلك تطبيق جميع التحديثات بنجاح، أو في حالة حدوث خطأ، تتم إعادة جميع المعاملة، مما يمنع حدوث تناقضات في البيانات.

التحدي الثالث: إنشاء تصميم متجاوب

المشكلة:

كان تصميم موقع ويب يتكيف بسلاسة مع أحجام الشاشات والأجهزة المختلفة أمرًا ضروريًا لتجربة مستخدم مثالية.

الحل:

- **الاستفادة من Bootstrap:** تم اختيار إطار عمل Bootstrap لنظام الشبكة القوي وفئات الاستجابة. قامت هذه الميزات بتبسيط عملية إنشاء تخطيط مرّن يتم تعديله تلقائيًا ليناسب عروض مختلفة، مما يضمن مظهرًا وشعورًا متسقًا عبر الأجهزة.

التحدي الرابع: التخزين الآمن لكلمات المرور

المشكلة:

يُعد تخزين كلمات مرور المستخدمين بشكل آمن أمرًا بالغ الأهمية لحماية حساباتهم.

الحل:

- **تنفيذ تجزئة كلمات المرور:** لا يتم تخزين كلمات المرور كنص عادي في قاعدة البيانات أبدًا. بدلاً من ذلك، يتم استخدام خوارزمية التجزئة bcrypt لإنشاء تجزئة آمنة أحادية الاتجاه لكلمة المرور. يضمن ذلك بقاء كلمات المرور الفعلية محمية حتى في حالة اختراق قاعدة البيانات.

النهج العام لحل المشكلات:

طوال عملية التطوير، تم تبني نهج منظم لحل المشكلات:

1. **تحديد المشكلة:** حدد المشكلة أو التحدي بوضوح.
2. **البحث وجمع المعلومات:** استشر الموارد عبر الإنترنت، والوثائق، والمجتمعات للحصول على حلول محتملة وأفضل الممارسات.
3. **تقسيم المشكلة:** قسّم المشكلات المعقدة إلى مكونات أصغر وأكثر قابلية للإدارة.
4. **التجربة والاختبار:** جرب مناهج مختلفة، وقم بتجربة التعليمات البرمجية للعثور على الحل الأكثر فعالية.
5. **توثيق الحل:** وثّق التنفيذ المختار بوضوح للمساعدة في الصيانة والفهم في المستقبل.

5 الخاتمة

5.1 ملخص المشروع:

نجح هذا المشروع في تقديم موقع إلكتروني تجاري ديناميكي، وفي تحقيق هدفه الرئيسي وهو عرض مفاهيم وتقنيات التجارة الإلكترونية الرئيسية. من خلال دمج لغات HTML و CSS و JavaScript و PHP و MySQL، يوفر الموقع منصة للمستخدمين لتصفح كتالوج المنتجات، وإدارة سلة التسوق، والتسجيل والدخول بأمان، وإكمال عملية الدفع، ومراجعة سجلات الطلبات.

5.2 نتائج التعلم:

وفر تطوير موقع التجارة الإلكتروني هذا تجربة تعليمية قيمة، ووسع مهاراتي في جوانب مختلفة من تطوير الويب:

- **التطوير متعدد الطبقات:** عزز المشروع فهمي للطبيعة المترابطة لتقنيات الواجهة الأمامية والواجهة الخلفية، وكيفية عملها معًا لإنشاء تطبيقات ويب ديناميكية.
- **دمج PHP و MySQL:** عمقت معرفتي باستخدام PHP للتعامل مع قاعدة بيانات MySQL، بما في ذلك الاتصال بقاعدة البيانات، وكتابة الاستعلامات، واسترجاع البيانات، وتنفيذ عمليات (CRUD إنشاء، وقراءة، وتحديث، وحذف).
- **AJAX والمحتوى الديناميكي:** قدم المشروع تجربة عملية في استخدام AJAX لإنشاء تجربة مستخدم أكثر سلاسة، لا سيما في تحديث سلة التسوق ديناميكيًا دون الحاجة إلى إعادة تحميل الصفحة بأكملها.
- **ممارسات الأمان:** أصبحت على دراية أكبر بمشكلات أمان تطبيقات الويب ونفذت تدابير لمنع حقن SQL، و ثغرات XSS، وضمان تخزين كلمات المرور بأمان باستخدام تقنيات التجزئة.
- **حل المشكلات وإصلاح الأخطاء:** قدم المشروع العديد من الفرص لتعزيز مهارات حل المشكلات وإصلاح الأخطاء، باستخدام أدوات مطور المتصفح و نهج منهجي لتحديد و حل المشكلات.

5.3 تحسينات المستقبل:

على الرغم من أن التنفيذ الحالي يوفر أساسًا متينًا، فإن هناك العديد من التحسينات التي يمكن أن تعزز وظائف الموقع وتجربة المستخدم:

- **تعليقات وتقييمات المستخدم:** ستضيف إضافة نظام لترك المستخدمين التعليقات والتقييمات للمنتجات دليلًا اجتماعيًا قويًا وسيساعد في اتخاذ قرارات الشراء.
 - **البحث المتقدم والتصفية:** سيؤدي تنفيذ قدرات بحث أكثر تقدمًا، بما في ذلك المرشحات لفئات المنتجات و نطاقات الأسعار و سمات أخرى، إلى تحسين اكتشاف المنتجات.
 - **لوحة تحكم المسؤول:** سيؤدي تطوير واجهة إدارية إلى تمكين إدارة فعالة للمنتجات و الطلبات و حسابات المستخدمين و بيانات الموقع الأخرى.
- عمل مشروع التجارة الإلكتروني هذا كمعلم هام في رحلة تطوير الويب الخاصة بي، و عزز مهاراتي الموجودة و عرّفتني على تقنيات و ممارسات جديدة يمكنني البناء عليها في المشاريع المستقبلية.