

E-COMMERCE WEBSITE

[Start page](#)

[Adding to cart](#)

[Adding more to the cart](#)

[Checkout - pay](#)

[Results of a successful purchase](#)

[Admin functionality](#)

[Removing a product as an admin](#)

[Admin - add new product](#)

[Admin - sales and profits](#)

[User Registration](#)

[Summary of requirements and how I met them:](#)

[Ingång för kund med:](#)

[Sessionshantering \(Session management\)](#)

[Shoppingvagn \(Shopping cart\)](#)

[Kassa \(Checkout\)](#)

[Epostlista \(Email list\)](#)

[Kaka för automatisk identifiering \(Cookie for automatic identification\)](#)

[Ingång för innehavare med:](#)

[Lager \(Stock\)](#)

[Kunder \(Customers\)](#)

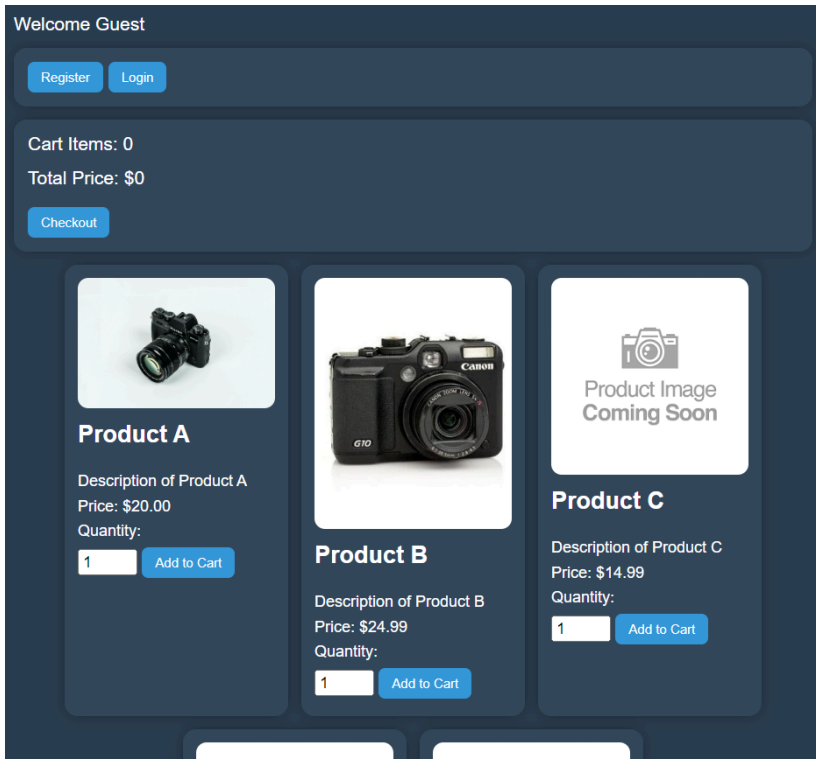
[Omsättning \(Turnover\), Vinst \(Profit\)](#)

[Other details that I made sure to consider](#)

E-COMMERCE WEBSITE

Start page

This is the start-page (index.php). It looks like a standard online-store, with the exception that the cart and its contents is not “hidden” in the upper right corner.



Adding to cart

Try adding a “Product A” to the cart, and notice the session data gets updated.

Session Data:

```
Array
(
    [start_time] => 1705063164
    [cart] => Array
        (
            [1] => 1
        )
)
```

```
<html>
<details class="debug-info">
  <summary>Show Debugging Information</summary>
  <div>
    <h2>Debugging Information</h2>
    <h3>Session Data:</h3>
    <pre><?php print_r($_SESSION); ?></pre>
    <h3>Cookie Data:</h3>
    <pre><?php print_r($_COOKIE); ?></pre>
  </div>
</details>
</html>
```

And the cart will look like this:

Welcome Guest


[Register](#) [Login](#)

Cart Items: 1

Total Price: \$20

Product A x 1 = \$20

[Checkout](#)




Product A

Description of Product A

Price: \$20.00

Quantity: [Add to Cart](#)




Product B

Description of Product B

Price: \$24.99

Quantity: [Add to Cart](#)



Product C

Description of Product C

Price: \$14.99

Quantity: [Add to Cart](#)

(Try refreshing the page, the cart doesn't change, because it is saved in a session variable, and fetched whenever you load the page.. but after 60 minutes the session will run out and your cart will be empty - see session_timeout.php for more details).

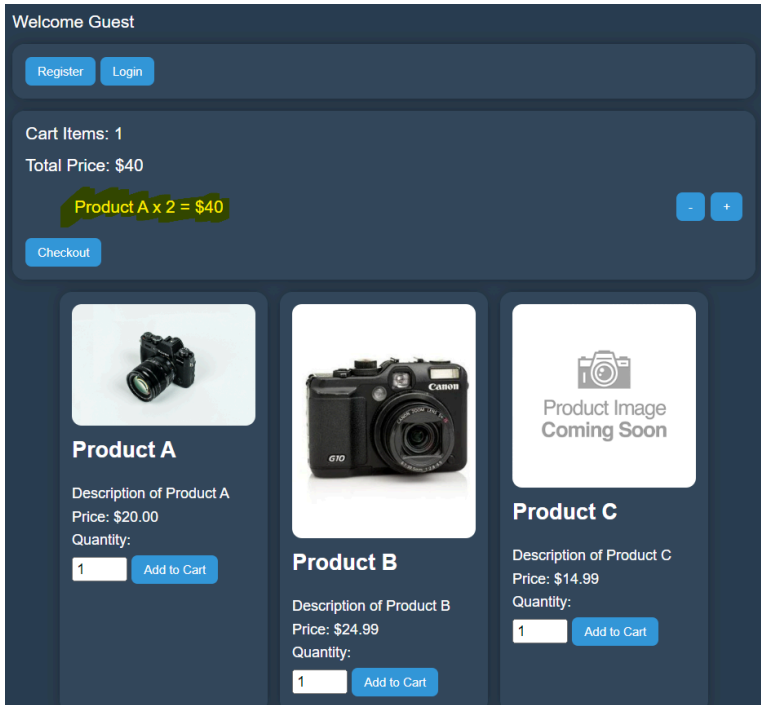
Adding more to the cart

Clicking the '+' will update the session data:

```
Session Data:

Array
(
    [start_time] => 1705063164
    [cart] => Array
        (
            [1] => 2
        )
)
```

Now the cart looks like this. Notice the total price is now updated too.



The total price was calculated by reading the price in the 'products' table, times the amount added to cart (20*2).

| product_id | product_name | product_description | product_image | product_price | product_stock | product_cost |
|------------|--------------|--------------------------|---------------|---------------|---------------|--------------|
| 1 | Product A | Description of Product A | product_a.jpg | 20.00 | 10 | 10.00 |
| 2 | Product B | Description of Product B | product_b.jpg | 24.99 | 10 | 10.00 |
| 3 | Product C | Description of Product C | product_c.jpg | 14.99 | 7010 | 10.00 |
| 10 | Product D | Description of Product D | zdf | 4.00 | 10 | 10.00 |
| 11 | Product E | Description of Product E | a3 | 55.00 | 10 | 10.00 |

Checkout - pay

We click on checkout and this takes us to another page, where the cart is still visible.

Cart

Cart Items: 1
Total Price: \$40.00
Product A x 2 = \$40

Checkout

Checkout

PayPal

Debit or Credit Card

Powered by PayPal

Show Debugging Information

Product Quantities

In order to pay, click “Debit or Credit Card”. We are using a sandbox version of PayPal, it works if you have created a dummy debit card.

Checkout

Debit or Credit Card

Kortnummer
4032 0311 3604 0994

Går ut
04 / 24

Säkerhetskod
822

Omräkningskurs för den här valutaomvandlingen.

USD

SEK

Totalsumma att betala
kr40.00 SEK

Kortutfärdaren fastställer omräkningskursen och vilka avgifter som de kan ta ut. Kontrollera ditt kortutdrag för det slutliga beloppet.

Faktureringsadress

Förnamn
Anton

Efternamn
Liampa

Gatuadress
SomeStreet 1

Adressrad 2

Postnummer
147 50

Ort
Tumba

You can use: 4032 0311 3604 0994
which expires 04/24 and has the code 822

Pick SEK, and then make up some phone number address, name etc.

Results of a successful purchase

The ‘products’ table looks like this now, notice the quantity is 2 less than before:

| product_id | product_name | product_description | product_image | product_price | product_stock | product_cost |
|------------|--------------|--------------------------|---------------|---------------|---------------|--------------|
| 1 | Product A | Description of Product A | product_a.jpg | 20.00 | 8 | 10.00 |
| 2 | Product B | Description of Product B | product_b.jpg | 24.99 | 10 | 10.00 |
| 3 | Product C | Description of Product C | product_c.jpg | 14.99 | 7010 | 10.00 |
| 10 | Product D | Description of Product D | zdf | 4.00 | 10 | 10.00 |
| 11 | Product E | Description of Product E | a3 | 55.00 | 10 | 10.00 |

The “order” table might also be interesting to look at. A new entry has been added.

| order_id | customer_id | order_date | order_total | order_status |
|----------|-------------|---------------------|-------------|--------------|
| 9 | NULL | 2023-08-04 00:58:16 | 89.96 | Pending |
| 10 | NULL | 2023-08-04 01:14:12 | 24.99 | Pending |
| 11 | NULL | 2023-08-04 01:16:20 | 24.99 | Pending |
| 12 | NULL | 2023-08-16 07:21:46 | 0.00 | Pending |
| 13 | NULL | 2023-08-16 07:25:50 | 134.94 | Pending |
| 14 | NULL | 2023-08-16 07:29:23 | 19.99 | Pending |
| 15 | NULL | 2023-08-16 07:50:11 | 39.98 | Pending |
| 16 | NULL | 2023-08-16 16:00:03 | 40.00 | Pending |

And the “orderitems” table looks like the following. We can see that 2 of product_id “1” were sold for 20 dollars each.


| order_item_id | order_id | product_id | quantity | price_at_time_of_purchase | cost_at_time_of_purchase |
|---------------|----------|------------|----------|---------------------------|--------------------------|
| 12 | 9 | 1 | 2 | 19.99 | 0.00 |
| 13 | 10 | 2 | 1 | 24.99 | 0.00 |
| 14 | 11 | 2 | 1 | 24.99 | 0.00 |
| 15 | 13 | 2 | 3 | 24.99 | 0.00 |
| 16 | 13 | 1 | 3 | 19.99 | 0.00 |
| 17 | 14 | 1 | 1 | 19.99 | 0.00 |
| 18 | 15 | 1 | 2 | 19.99 | 0.00 |
| 19 | 16 | 1 | 2 | 20.00 | 10.00 |

Admin functionality

The site can be accessed by an admin like this: <https://localhost/wprog2/9/administration/admin.php>

Enter username: **admin**, password: **admin**.

Now you can add a new product, remove one, see quantity, see profits. Notice that the quantity of product A is 8, because of the previous purchase.




Product A

Description of Product A

Price: \$20.00

Quantity: 8

[Remove Product](#)




Product B

Description of Product B

Price: \$24.99

Quantity: 10

[Remove Product](#)




Product C

Description of Product C

Price: \$14.99

Quantity: 7010

[Remove Product](#)




Product D

Description of Product D

Price: \$4.00

Quantity: 10

[Remove Product](#)



Product E

Description of Product E

Price: \$55.00

Quantity: 10

[Remove Product](#)

Add New Product

Product Name:

Product Description:

Product Image URL:

Product Price:

Product Cost:

Stock:

[Add Product](#)

Email Section

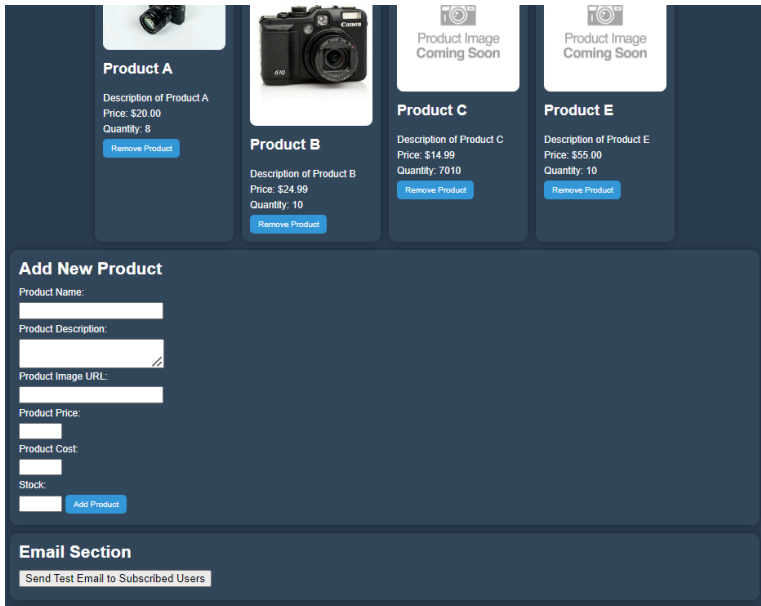
[Send Test Email to Subscribed Users](#)

Sales and Profit

Total Sales: 884.67

Admin - remove product

This is how it looks if we remove Product D. However, we cannot delete a product that is already in 'orderitems'. My thought here is that we don't want to accidentally remove a product that a customer has ordered.

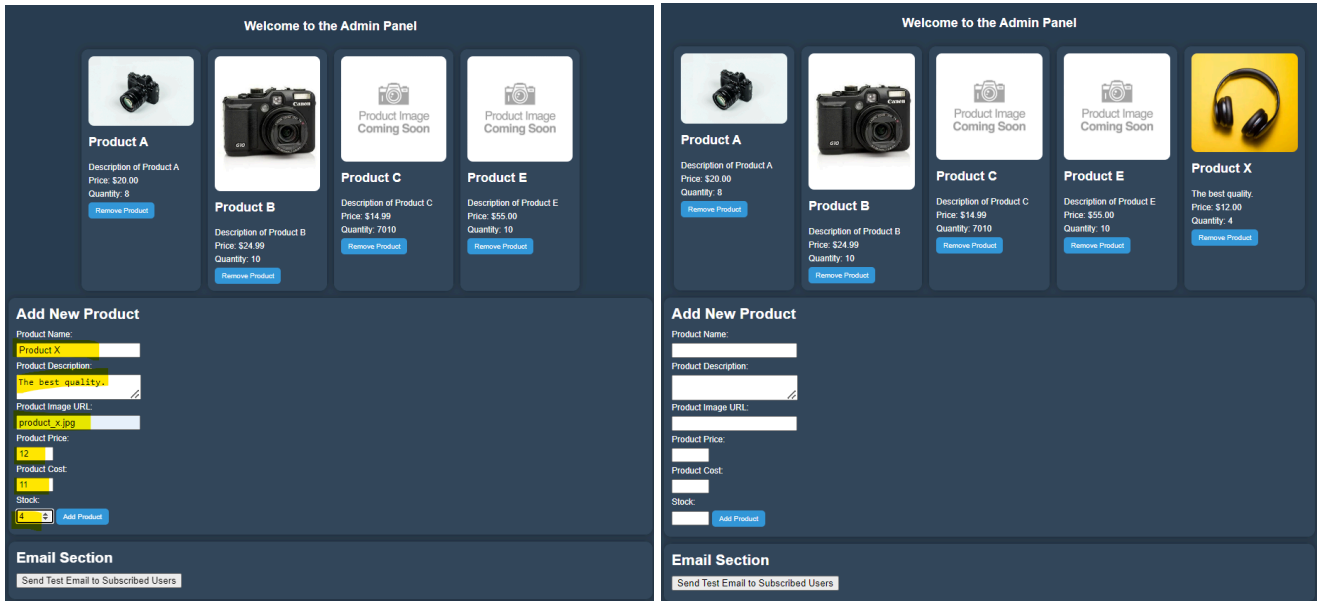


Assuming removal went fine, the removal should also be reflected in the the 'products' table:

| product_id | product_name | product_description | product_image | product_price | product_stock | product_cost |
|------------|--------------|--------------------------|---------------|---------------|---------------|--------------|
| 1 | Product A | Description of Product A | product_a.jpg | 20.00 | 8 | 10.00 |
| 2 | Product B | Description of Product B | product_b.jpg | 24.99 | 10 | 10.00 |
| 3 | Product C | Description of Product C | product_c.jpg | 14.99 | 7010 | 10.00 |
| 11 | Product E | Description of Product E | a3 | 55.00 | 10 | 10.00 |

Admin - add new product

Here you can add a new product, this is self-explanatory. Images need to be already on the server for you to be able to add an image like shown on the image. In a future release, it would be good to add an image upload button here. For now, you can try using “product_x.jpg” in the image url field.



| product_id | product_name | product_description | product_image | product_price | product_stock | product_cost |
|------------|--------------|--------------------------|---------------|---------------|---------------|--------------|
| 1 | Product A | Description of Product A | product_a.jpg | 20.00 | 8 | 10.00 |
| 2 | Product B | Description of Product B | product_b.jpg | 24.99 | 10 | 10.00 |
| 3 | Product C | Description of Product C | product_c.jpg | 14.99 | 7010 | 10.00 |
| 11 | Product E | Description of Product E | a3 | 55.00 | 10 | 10.00 |
| 16 | Product X | The best quality | product_x.jpg | 12.00 | 4 | 11.00 |

TIP: Fill in 2 in the “stock” field. Now you have 2 of these. Try to purchase more than 2 as a customer. It should not allow you to add more than 2 to the cart.

Admin - sales and profits

Look at the sales and profits section below. The example is a bit unrealistic, because how can we have a profit of 865 and cost of only 20. Don't mind this. Pay attention to the fact that we have a profit based on sales minus cost.

Remove Product

Remove Product

Add New Product

Product Name:

Product Description:

Product Image URL:

Product Price:

Product Cost:

Stock:

Add Product

Email Section

Send Test Email to Subscribed Users

Sales and Profit

Total Sales: 884.67

Total Cost: 20.00

Total Profit: 864.67

We go total costs from the 'orderitems' table , and the total sales from the 'orders' table (see images below). Orderitems is the table where products go after a user has gone through the checkout process. All orderitems have a pending status. In the future, it would be good to include Pending/Sent/Received/Unreturnable here. So that unreturnable items are going to count towards the profit, but for example, items that are sent or pending should not yet be counted towards profits.

| order_id | customer_id | order_date | order_total | order_status |
|----------|-------------|---------------------|-------------|--------------|
| 3 | NULL | 2023-08-02 22:41:25 | 74.99 | Pending |
| 4 | NULL | 2023-08-02 22:48:44 | 74.99 | Pending |
| 5 | NULL | 2023-08-04 00:53:53 | 89.96 | Pending |
| 6 | NULL | 2023-08-04 00:57:34 | 89.96 | Pending |
| 7 | NULL | 2023-08-04 00:57:58 | 89.96 | Pending |
| 8 | NULL | 2023-08-04 00:58:07 | 89.96 | Pending |
| 9 | NULL | 2023-08-04 00:58:16 | 89.96 | Pending |
| 10 | NULL | 2023-08-04 01:14:12 | 24.99 | Pending |
| 11 | NULL | 2023-08-04 01:16:20 | 24.99 | Pending |
| 12 | NULL | 2023-08-16 07:21:46 | 0.00 | Pending |
| 13 | NULL | 2023-08-16 07:25:50 | 134.94 | Pending |
| 14 | NULL | 2023-08-16 07:29:23 | 19.99 | Pending |
| 15 | NULL | 2023-08-16 07:50:11 | 39.98 | Pending |
| 16 | NULL | 2023-08-16 16:00:03 | 40.00 | Pending |

| product_id | quantity | price_at_time_of_purchase | cost_at_time_of_purchase |
|------------|----------|---------------------------|--------------------------|
| 2 | 2 | 24.99 | 0.00 |
| 1 | 2 | 19.99 | 0.00 |
| 2 | 2 | 24.99 | 0.00 |
| 1 | 2 | 19.99 | 0.00 |
| 2 | 2 | 24.99 | 0.00 |
| 1 | 2 | 19.99 | 0.00 |
| 2 | 2 | 24.99 | 0.00 |
| 1 | 2 | 19.99 | 0.00 |
| 2 | 2 | 24.99 | 0.00 |
| 1 | 2 | 19.99 | 0.00 |
| 2 | 1 | 24.99 | 0.00 |
| 2 | 1 | 24.99 | 0.00 |
| 2 | 3 | 24.99 | 0.00 |
| 1 | 3 | 19.99 | 0.00 |
| 1 | 1 | 19.99 | 0.00 |
| 1 | 2 | 19.99 | 0.00 |
| 1 | 2 | 20.00 | 10.00 |

User Registration

Before I can demonstrate the email-sending functionality, I will have to register a new user back on the index.php page.

Welcome Guest


Register

Login


Cart Items: 0


Total Price: \$0

Checkout



Product A





Product Image
Coming Soon

Name:

Anton

Email:

Dummy15940@gmail.c...

Password:

.....

Address:

myaddress

Register

customers table: here we can see that the above user managed to register, and by default subscribed to emails is set to true.

| customer_id | customer_name | customer_email | customer_password | customer_address | subscribed_to_emails |
|-------------|---------------|----------------------|--|------------------|----------------------|
| 11 | Anton | Dummy15940@gmail.com | \$2y\$10\$e8zYe5avGonkD2yJRd8SgevVRLWldS.pM5kd3FJGfPY... | myaddress | 1 |

Send email to registered users

For this we need to go back to the admin panel. I've added some fields so we can write a custom email message and subject.

Email Section

Send an email to all subscribed users

Email Subject:

myFirstEmail

Email Body:

this is a message sent to
registered users

Send

It works! here is the result as seen in Gmail:

myFirstEmail Inbox x



Admin <dummy15940@gmail.com>
to Dummy15940 ▾

this is a message sent to
registered users

User login

As we have seen, the user exists in the database and gets the email sent by the admin. But can the user log in? Yes, and his login will be recognized when moving from page to page. Even if the browser is closed, provided cookies aren't cleaned.


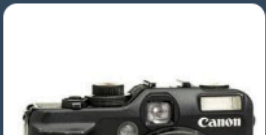

We login from index.php:

Welcome Guest

[Register](#) [Login](#)

Cart Items: 0
Total Price: \$0

[Checkout](#)



Login Form

Email

Password


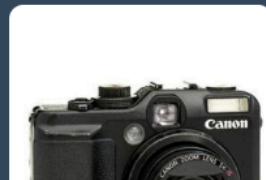

[Login](#)

Welcome Dummy15940@gmail.com

[Register](#) [Login](#)

Cart Items: 0
Total Price: \$0

[Checkout](#)



Product Image
Coming Soon

Logged in user - test cookies

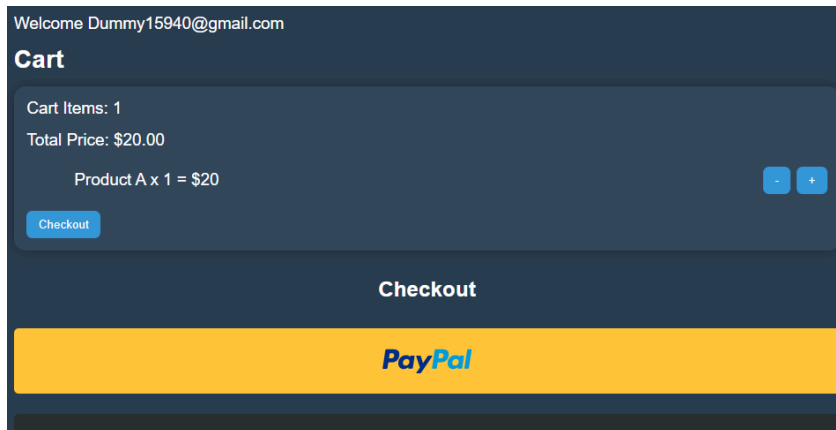
Right after logging in, here is one of the entries in the cookie array
[email] => Dummy15940@gmail.com

And session contains this

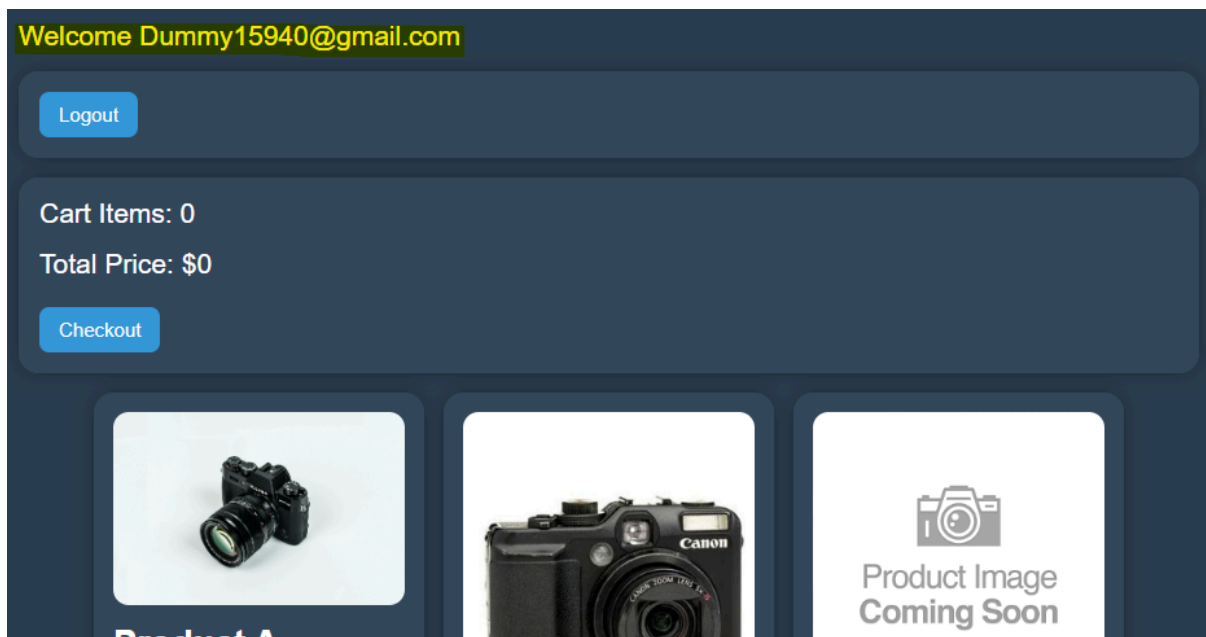
[login_user] => Dummy15940@gmail.com

[customer_id] => 11

Both of these remain when I go to checkout



and when I restart the browser:



However, it is worth noting that the cart will be empty even if it wasn't because its contents were only saved in the session variable. The email of whoever was logged was/is saved in a cookie - that's why we can retain that information.

Summary of requirements and how I met them:

Ingång för kund med:

Sessionshantering (Session management)

I ensure the cart contents are saved when we go from index.php to checkout.php, by using sessions. Also, the session gets cleared (and the contents disappear) after 60 minutes. This is done to keep the servers clean.

Shoppingvagn (Shopping cart)

The shopping cart can be seen on index.php and checkout.php. Right now, the page reloads when you update the cart. This is barely noticeable on firefox, but on brave it is evident. This can be fixed in a future version.

Kassa (Checkout)

I have checkout.php where you can see your shopping cart, and pay for it using PayPal sandbox mode.

Epostlista (Email list)

It is possible to send emails to all registered users in the admin.php panel.

Kaka för automatisk identifiering (Cookie for automatic identification)

If you restart the browser after logging in, you will realize that you remain identified by your email on the site.

Ingång för innehavare med:

Lager (Stock)

By going into site/administration/admin.php you can access info about product stocks, with the ability to add and remove products, specifying their quantity, price, cost etc.

Kunder (Customers)

Regular customers go to index.php

Omsättning (Turnover), Vinst (Profit)

The page admin.php contains a section summarizing your profit and turnover. This can be extended to show more detailed info.

Other details that I made sure to consider

- If not enough of one product, then don't allow adding it to the cart.
- Replaced login and register buttons with a logout button if you are already logged in, for the sake of clarity and minimalism
- Equal margins, and consistent colors, make the site esthetically pleasing.
- Products that do not have an image, have a placeholder image.
- Images that are of different sizes fit into the product containers.
- Flexible layout (try resizing).
- User passwords are not saved as plain text in the database.
- Admin functionality is clearly separated from other code (see folder: administration).
- Admin has own login using http-based authentication.
- Security feature: If you use http, the site simply doesn't work, and tries to redirect you to https.

Possible improvements

- Make the code adhere to OOP principles.
- Separate html and php more, to make the code clearer and separate view from logic.
- When a user makes a purchase, this is directly reflected in the profit that can be seen in the admin panel. In reality, this should not be the case. Items that are purchased and CANNOT be returned are the ones that count towards profit.