A Final Report

19013139 Myeong Su CHOI

Web Programming(006237-004)

Table Of Contents

1. The goal/objective and summary of the project
2. Wireframe on the main page
3. Description of each function

3 – 1. User Authentication

3 – 2. Comment function

3 – 3. Like function

3 – 4. Simple order function

3 – 5. Profile management function

3 – 6. Search function

1. Other function
2. My GitHub repository
3. **The goal/objective and summary of the project**

The goal of the project is to create clothing shopping site.

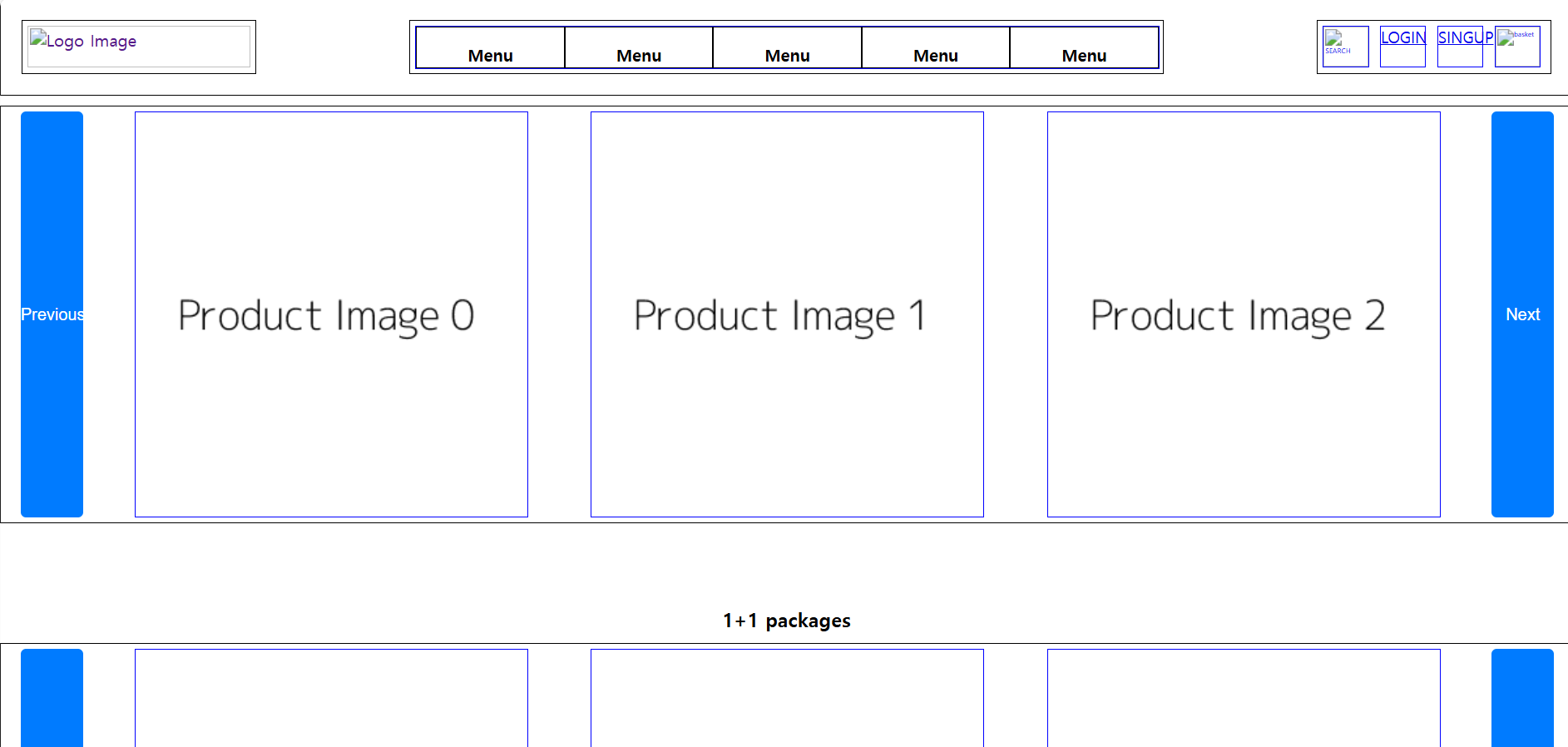
Shopping site has the ability to log in and sign up for membership, click ‘Like’ and purchase products. User can also search for products and add reviews on products. User can find likes and purchased products on user’s profile.

Implement the shopping site function as follows.

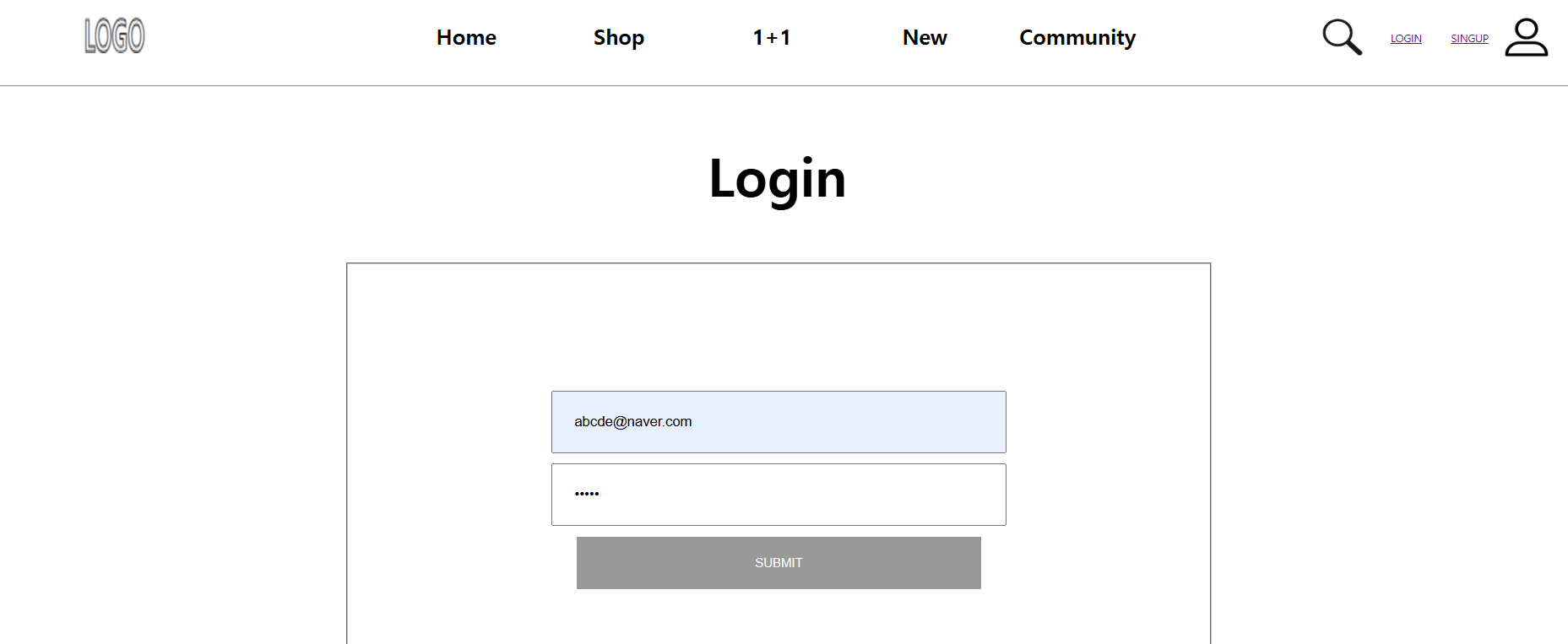
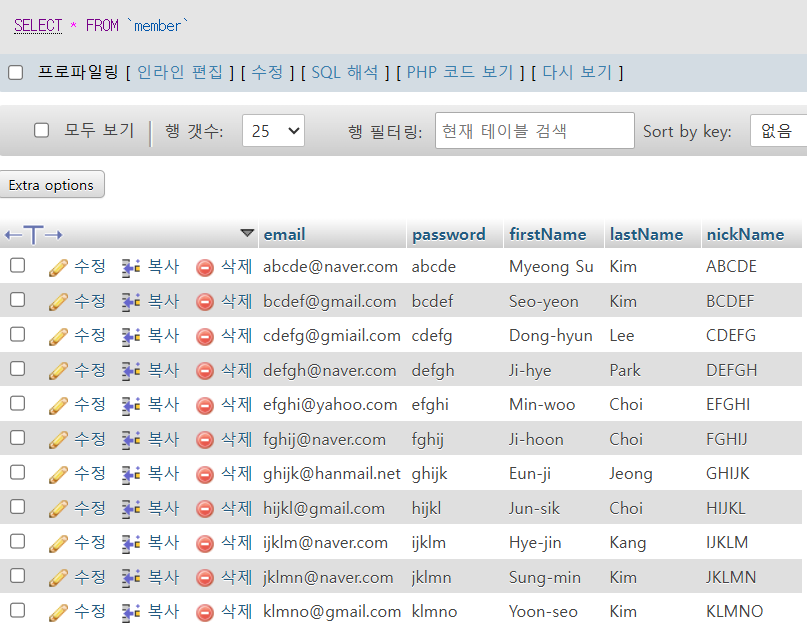
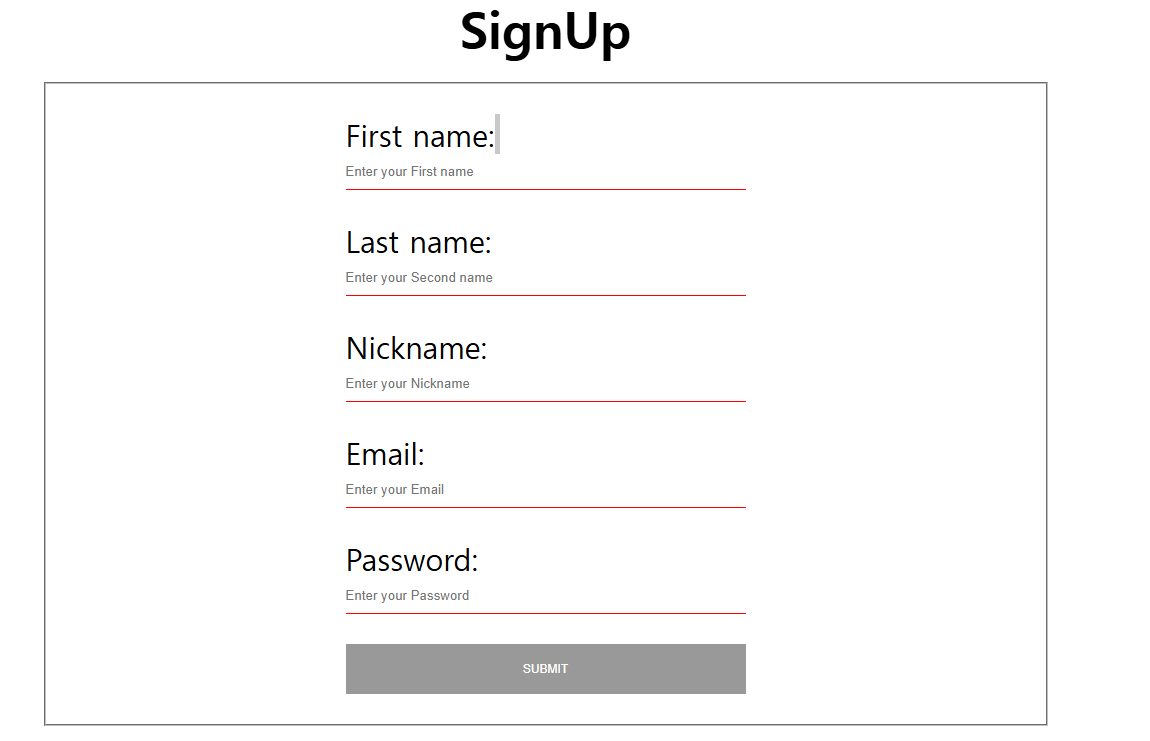
1. User Authentication / It provides login and membership services.
2. Comment function / It provides a function to comment on the product.
3. Like function / It provides the ability to press the Like button for the product.
4. Simpe order function / It provides a simple purchasing function.
5. Profile Management function / It should be able to edit the profile. In the profile, there are products that have been liked and purchased.
6. Search function / It provides the ability to find the product user’s want.
7. ~~It implement responsive design to ensure that the website is mobile-friendly.~~

* Due to lack of time, the following function could not be implemented.

1. **Wireframe on the main page**

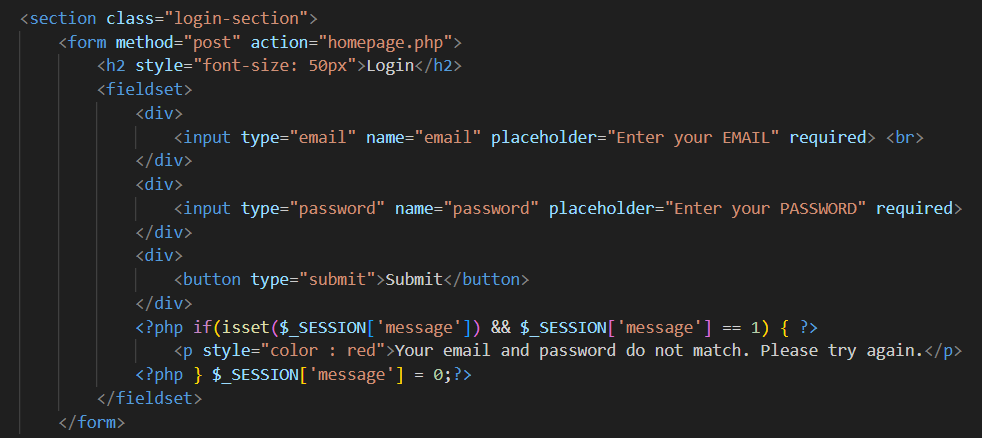
1. **Description of each function**
2. **User Authentication**

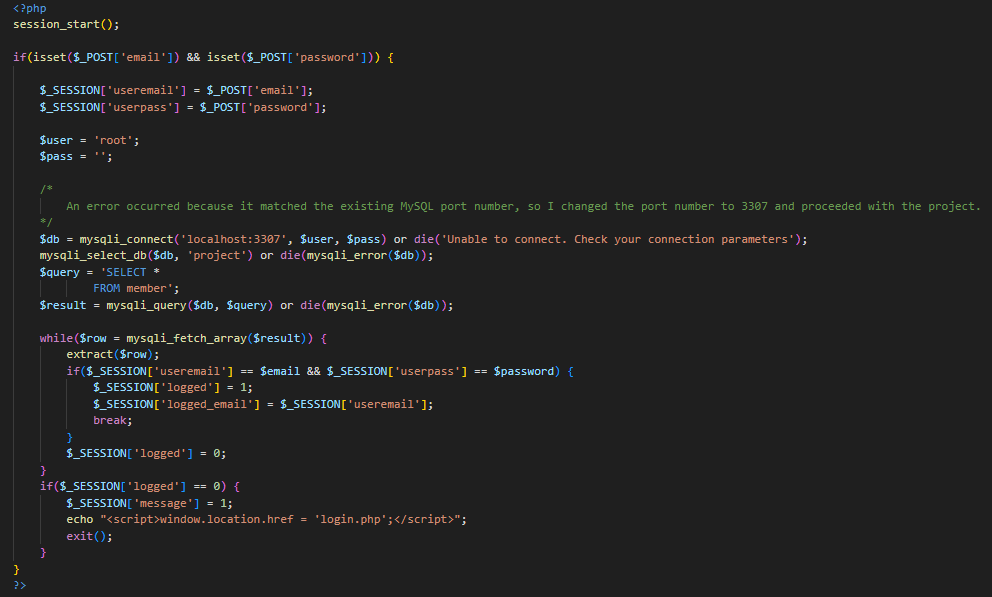
This capture is the values of the login page screen, membership screen and the member table of the project database.

User can log in by logging in with the email and password that exists in the database.

<How does it work?>

~~~~

From the login page to the main page named Homepage.php, pass the information through post method.



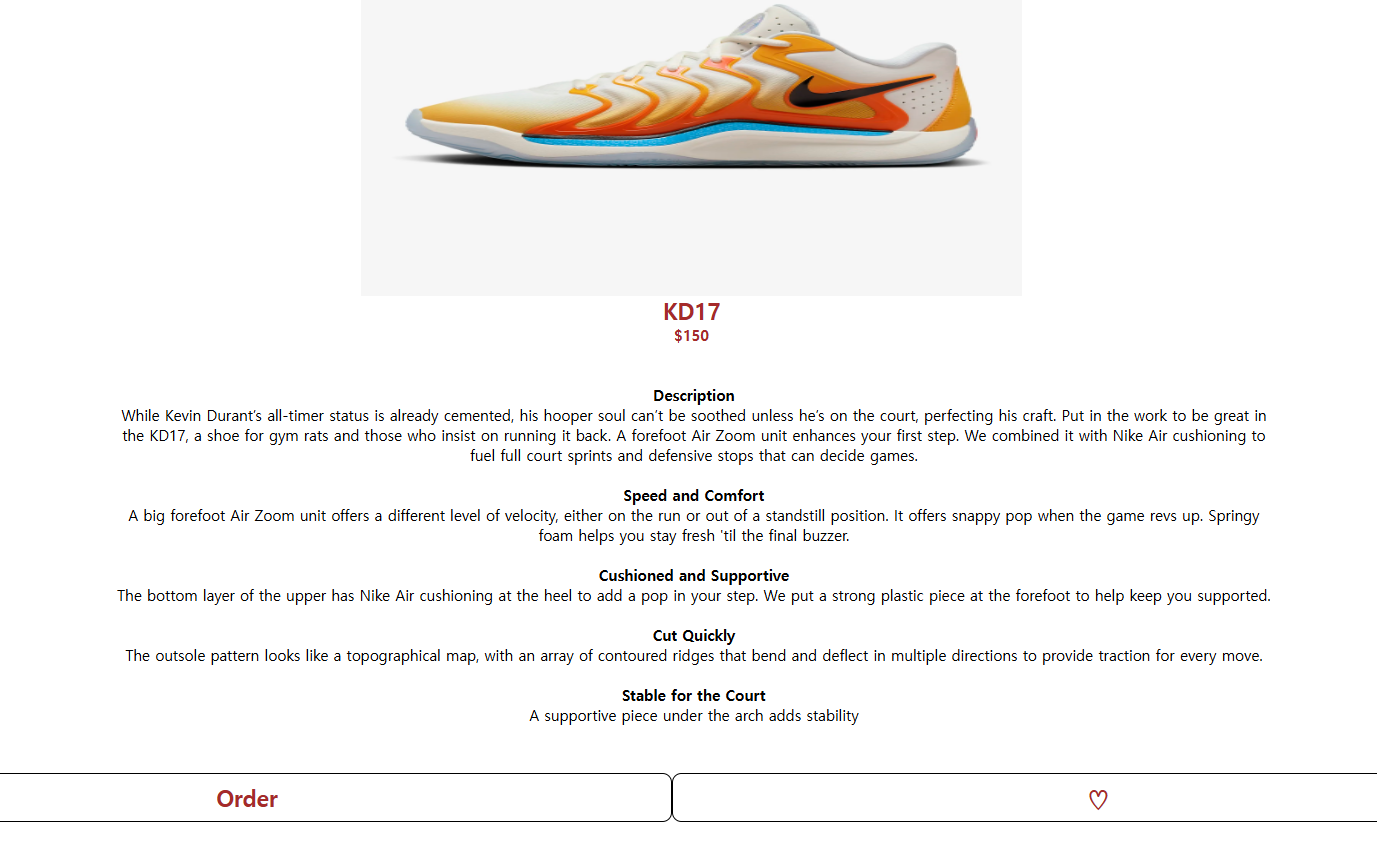
In Homepage.php, it connects to the project database and compares the information received by the post method with the information in the member table to log in if users are a member, and displays the login screen again with a message if users are not member.

It works similarly logic when signing up for membership. However, if membership is successful, new data must be put in the member table.

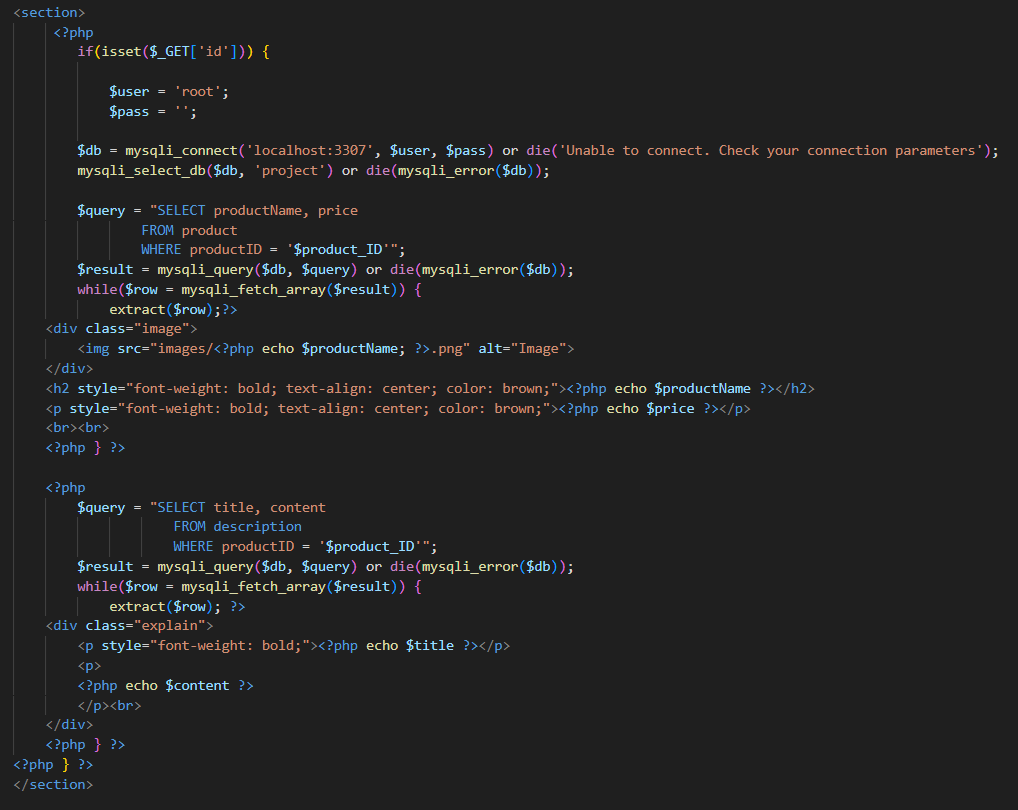
Reference : <https://developer.mozilla.org/en-US/docs/Web/API/Window/location#value>

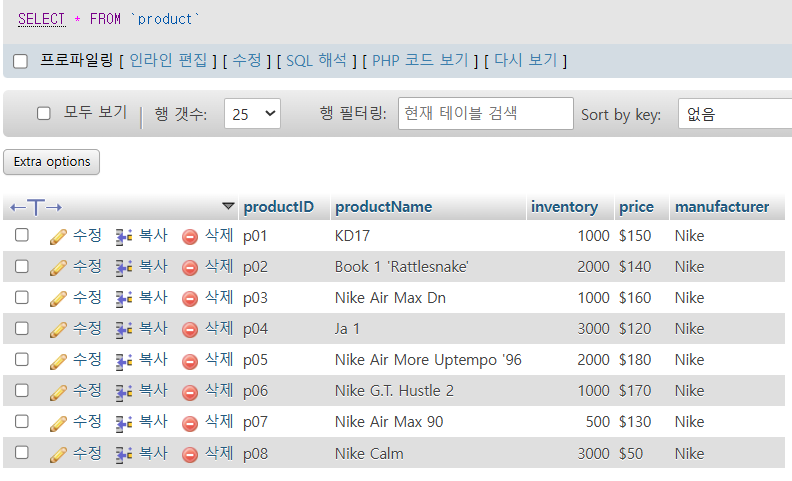
1. **Comment function**

<Product introduction>



<How does it work?>

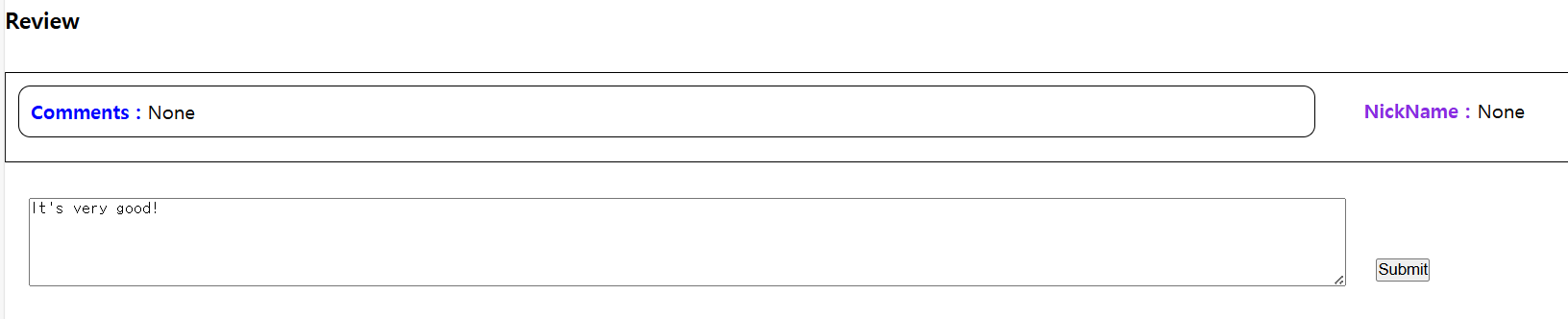


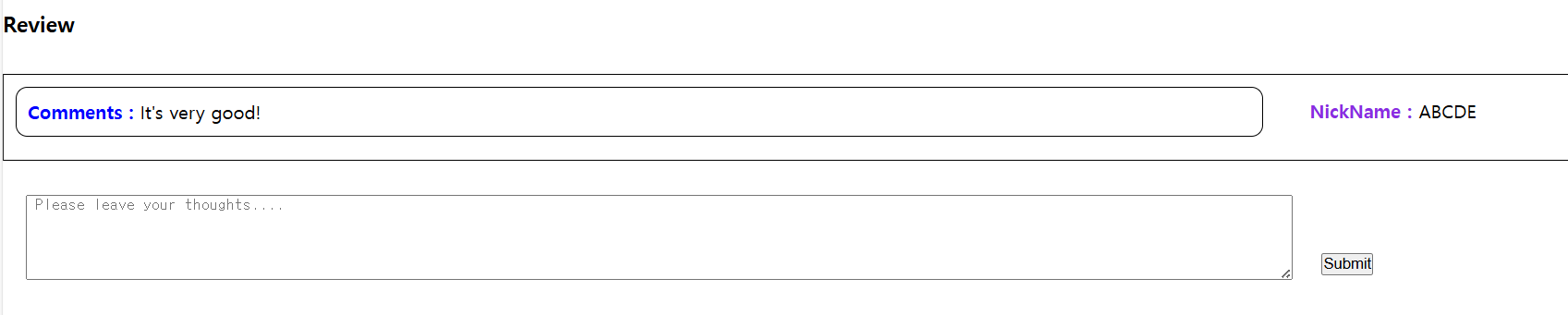


First, when a user clicks on a product, the product number is received as url data, and takes the image, name and price data of the product from the product table and displays it on the screen.

In addition, data are retrieved from the description table where the title and content are stored separately and a description of the product is displayed on the screen.

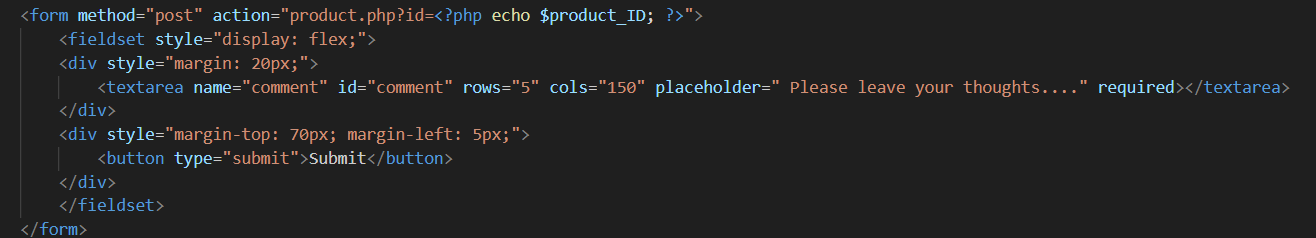
<Comment>





User can see what is displayed on the screen when user enters a new comment.

<How does it work?>



Send data with the product number to url via post method.



If there is any data that has benn transferred to the post method under the name comment, insert the data into the comment table of the project database. If users attempt to comment while not looged in, a warning window called please log in appears.

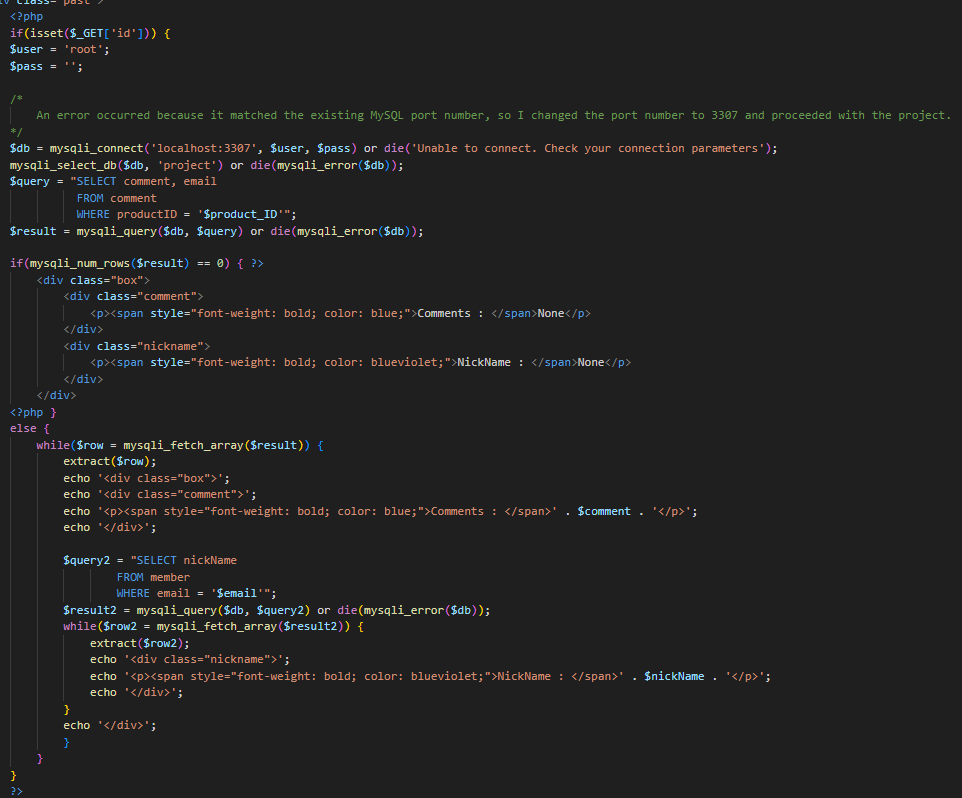
In the following code, the mysqli\_real\_escape\_string method was used so that strings containing ‘ ‘ could also be inserted into comments.

Also, to prevent the problem of adding comments that were previously added once again when you click refresh, use “echo’<meta http-equiv=”refresh” content=”0”>’ exit();” to automatically reload and stop running the php script.

Reference : <https://www.php.net/manual/en/mysqli.real-escape-string.php>

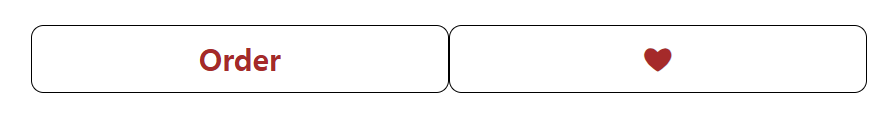
Reference : <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta#attr-http-equiv>





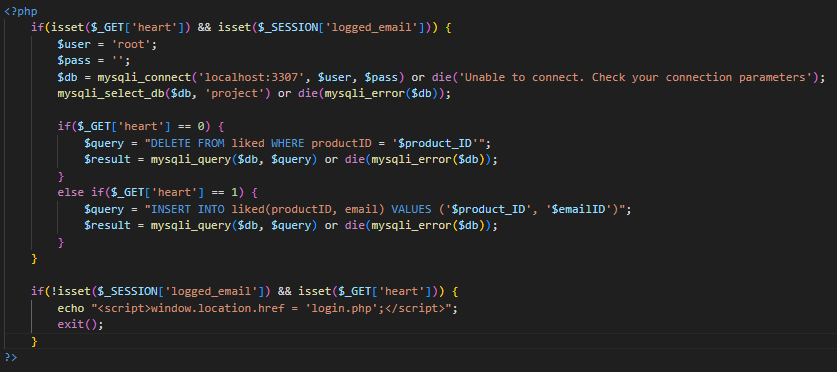
Then, take the data from the comment column and email column from the comment table and display it on the screen. Then, import the nickname value of the email from the member table and display who wrote the comment. If there is no data in the comment table, display None.

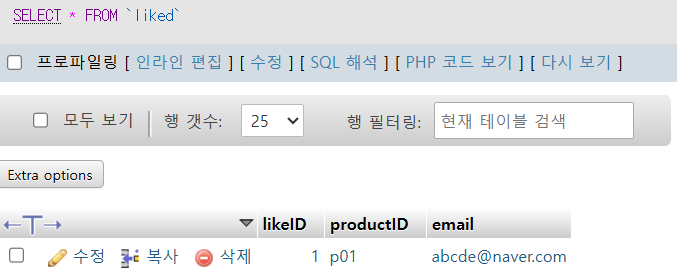
1. **Like function**



If users click on the empty heart in the previous product description image, it changes to a filled heart.

<How does it work?>





The server receives status information about whether a heart is changed to an empty heart(‘like canceled’) or a filled heart(‘like’) by url. If it is canceled, delete the corresponding product number in the like table in the project database. If it is like, add the corresponding product number to the liked table.

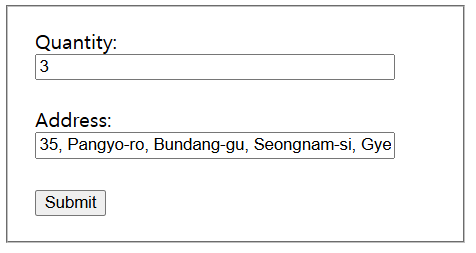
If users click ‘like’ when users are not logged in, set the login page to go.

Reference : <https://developer.mozilla.org/en-US/docs/Web/API/Window/location#value>

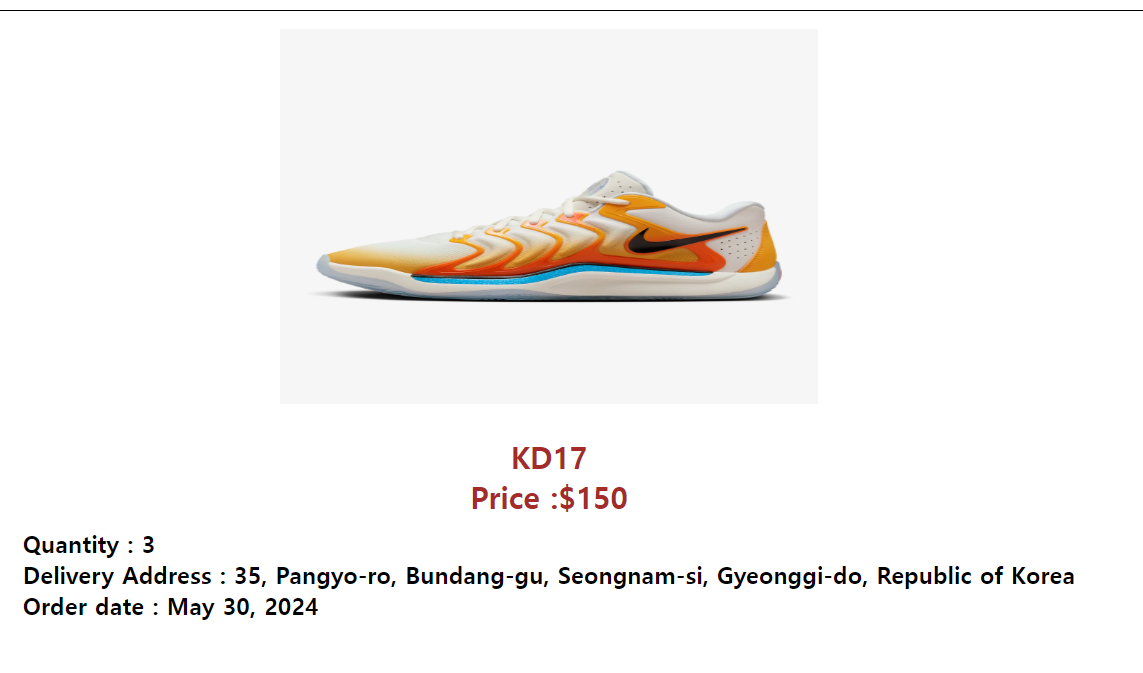


Then, take the products that the currently logged in user has clicked ‘like’ from the database, and make the tmp variable 1 if it matches the product number that is currently clicked, or 0 otherwise. After that, decide whether to display empty or filled hearts on the screen according to the value of the tmp variable.

1. **Simple order function**



When users click the Order button, users will see a screen where you write down the quantity and address.



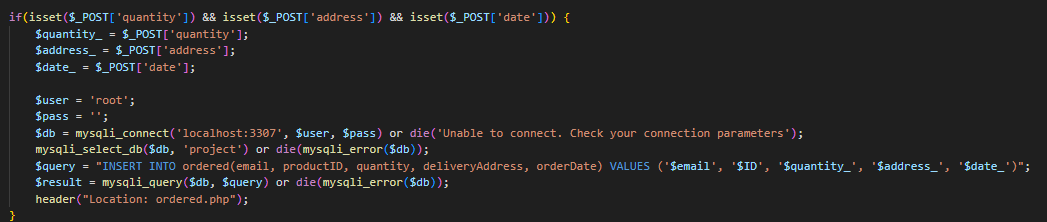
If users click the submit button, users will automatically go to the order history screen in user’s profile and see that the product users ordered has been added to the order history.

<How does it work?>



Send information from order.php to ordered.php using the post method.

Send the current date information using the date\_default\_timezone\_set function along with the quantity and address information.





Add data to the ordered table with information received from order.php .

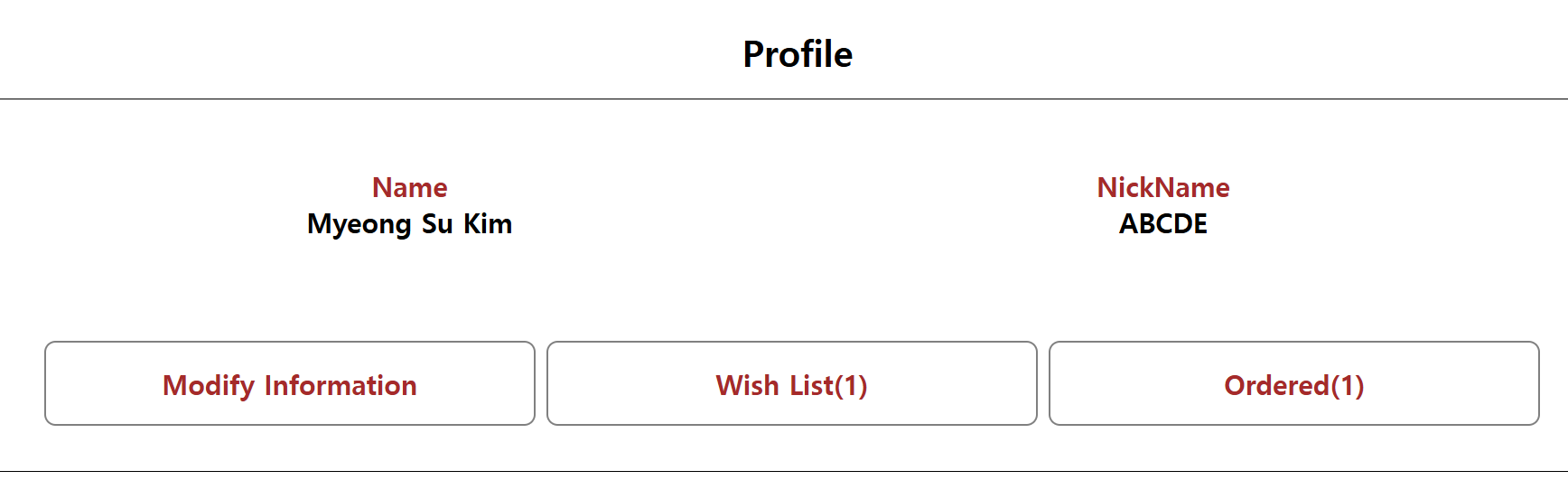
When order.php delivers information, the “header(“Location:ordered.php”)” is used to display the ordered.php screen,. This code prevents server from once again adding a product users recently added even if users refresh it.

Reference : <https://www.php.net/manual/en/function.header.php>

* It’s a shame that I didn’t put in the function for cancellation of the order.

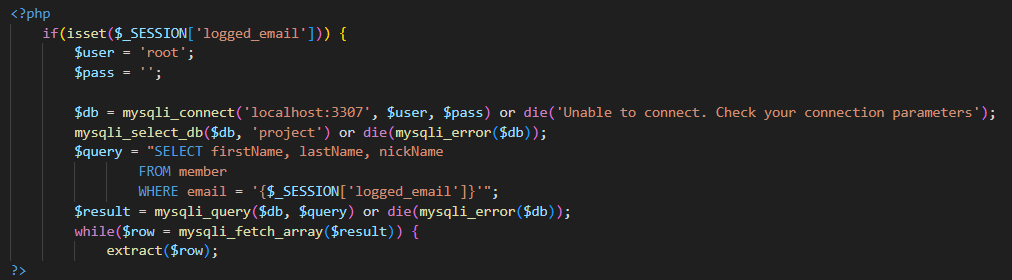
1. **Profile management function**

**<Profile screen>**



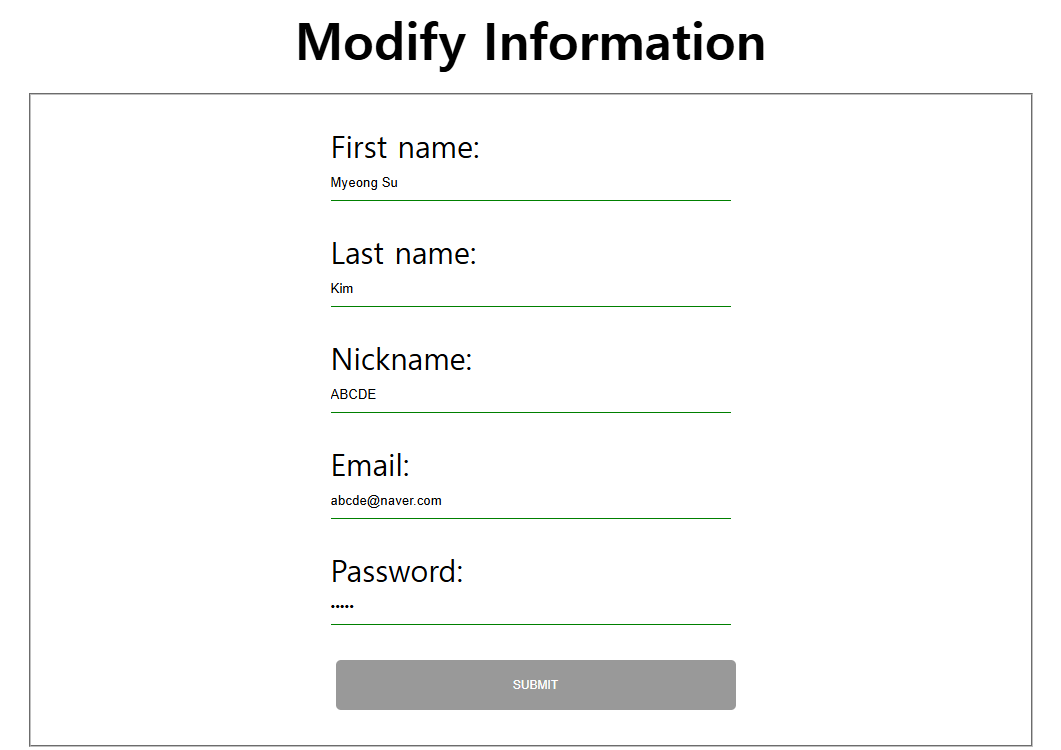
When users press a profile image, it goes to the user’s profile screen. There are buttons to modify user’s information, buttons to see products that they like, and buttons to see products that they have purchased. And it shows the number of products that have pressed likes and the number of products ordered.

<How does it work?>



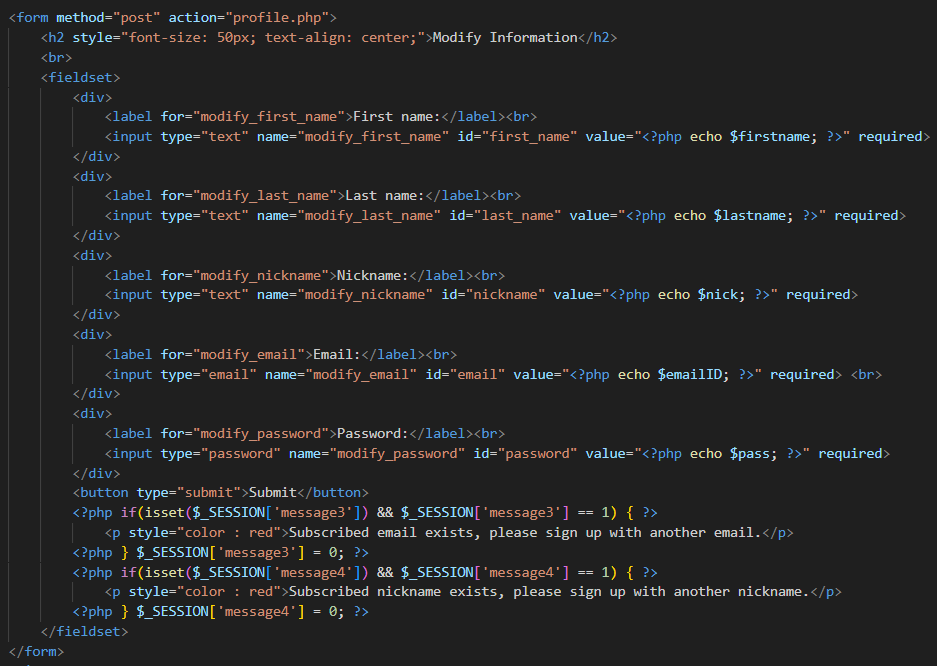
Connect to the project database with the data stored in the session variable ‘logged\_email’, get the firstName, lastName and nickname column values, and display them on the screen.

**<Modify information>**



When users click the modify information button, a window appears to modify the user’s information. Existing information is entered and displayed on the screen. It shows a message that cannot be modified if the email and nickname overlap with existing members.

<How does it works?>



Transfer data to profile.php using the post method.

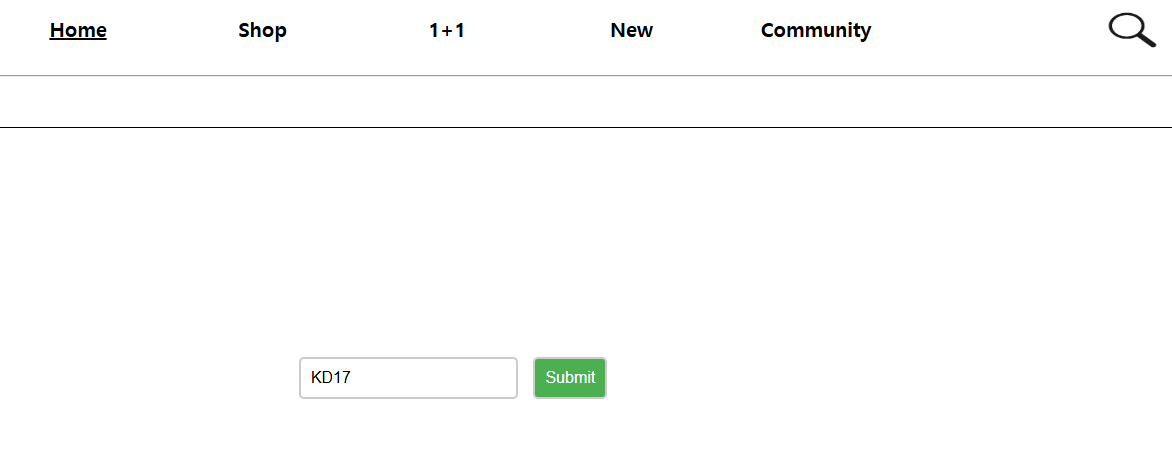


If it does not match existing data, it connects to the database to modify the information. Don’t forget to change the value stored in the session variable ‘logged\_email’ to the newly modified value at the end.

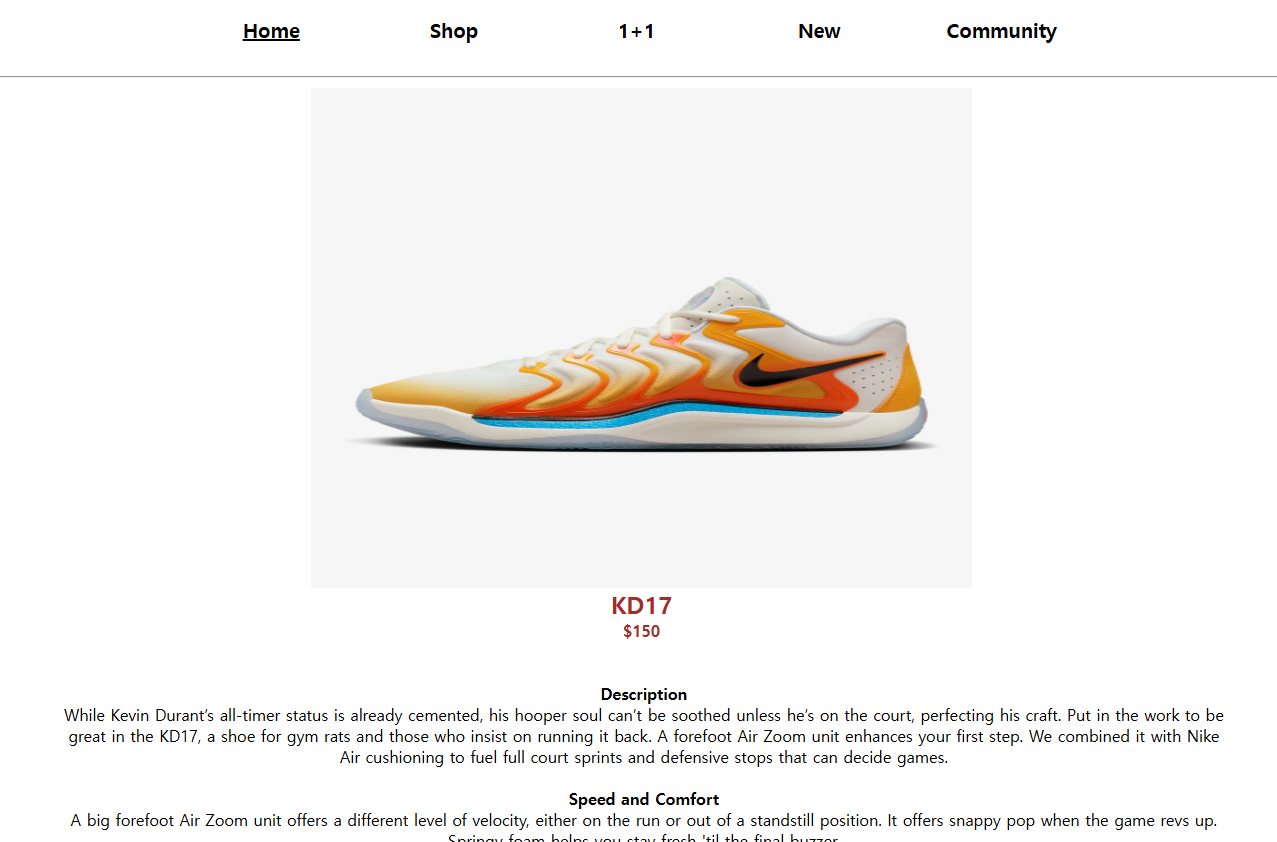
**<Wishlist & Order history>**

It shows the products that the user liked and ordered. It is implemented in simple logic, so I will not explain it.

1. **Search function**



If users press the magnifying glass-shaped image in the image above, users will see the following screen. And if users search for the name of the product users want, users will see the screen of the product.



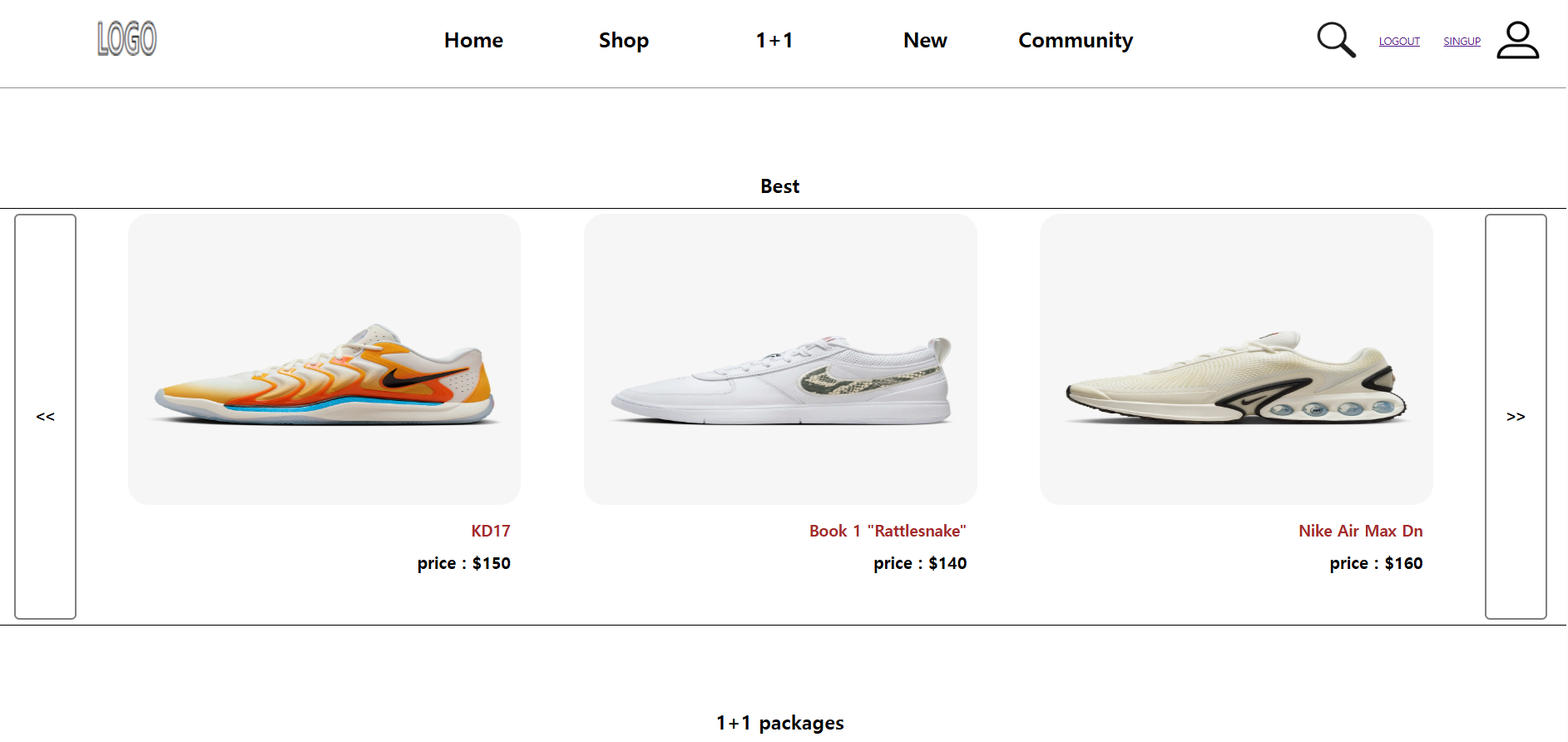
<How does it works?>



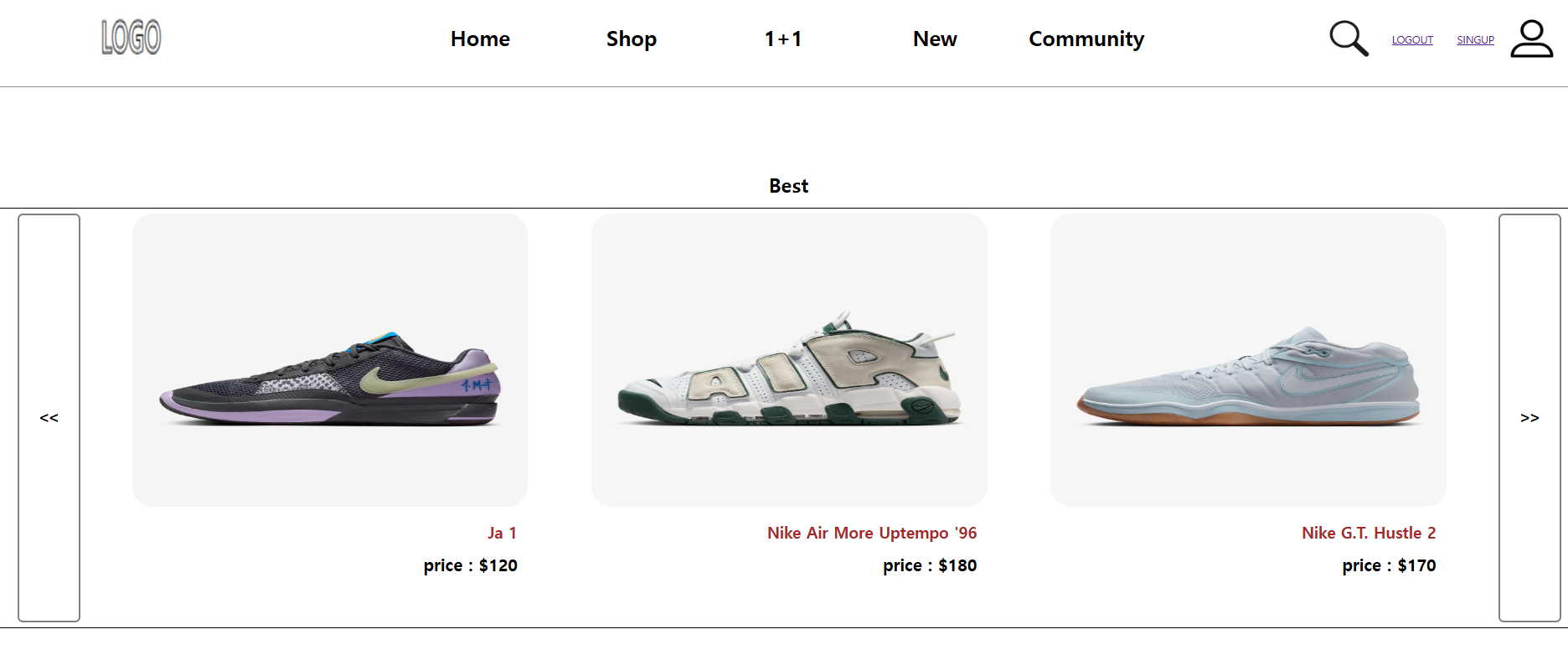
Send data from search.php to searchProduct.php using the post method. If the name of the product to be searched exists in toe product table, send it to product.php including the product number information in the product.php.

If it is a non-existent product name, it shows the search.php screen again with the message that it is a non-existent product.

1. Other function

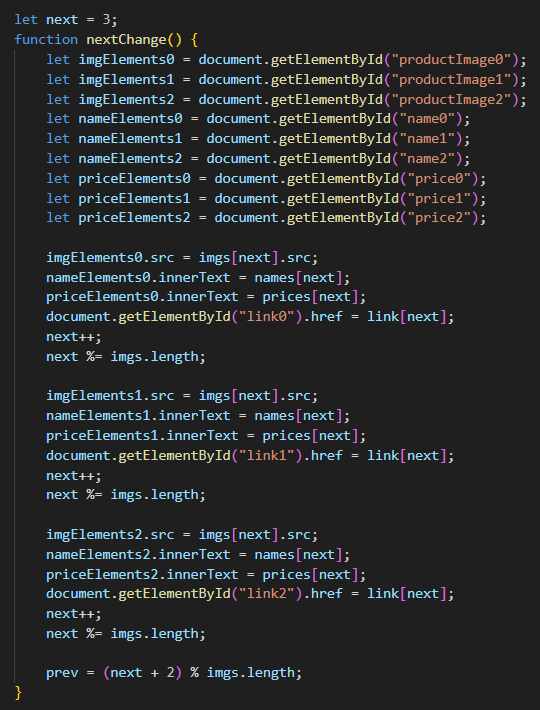


This is the main page of the shopping site. I couldn’t cover all the products, so I added images only of the best products. There are a total of eight products, and when users press the buttons both sides, the following products are displayed three at a time.



* I got the image from the Nike site.

<How does it works?>



I store the image url in the array named files, I store the address of the destination where the data is to be sent in the post method in the array named link, I store the price information of the products in the array named prices. And I use the id to take the data of the products displayed on the screen and put the data of the products to be displayed. The buttons that look at the previous products are implemented with the same logic.

1. My GitHub repository

[withMSCHOI/web\_project (github.com)](https://github.com/withMSCHOI/web_project)