- You can refer to demo: http://vader.kean.edu/CPS3740
- Project URL : http://obi.kean.edu/~xxxx/CPS3740 where **xxxx** is your **obi** login ID in this requirement.

(1) Database table:

For data insertion, we should create the **Order_xxxx table in CPS3740_xxxx** database instance with these fields**:**

1. **oid**: the order id field in your **Order_xxxx** table. It should be the primary key and auto increment.
2. **order_datetime**: the datetime when the order is placed by the customer.
3. **order_qty**: a positive integer.
4. **cid**: a foreign key reference to the primary key in CPS3740.Customers table.
5. **pid**: a foreign key reference to the primary key in CPS3740.Products table.
6. All values cannot be NULL, empty, 0 and negative.

(2) Web programs

1. Make the **index.html** file. The web page should display the followings. Please refer to **Figure 1**.
   - 1.1 Your name
   - 1.2 A link "View all customers" to call **view_customers.php** and
   - 1.3 A search function "Search product names" to call **search_products.php** to search products based on the input keyword.
   - 1.4 The login and password input boxes, a submit button for customers to login to their home page.

2. **Authentication**: A program named "**login.php**" be called to do the following functions after the **Submit** button is clicked. The program name must be **login.php**.
   - 2.1 The customer login and password must be read from the table **CPS3740.Customers**. Login and password cannot be hard coded in the program.
   - 2.2 If the **login does not exist in the database**, please display an error message "**Login XXXX doesn't exist in the database**" **and exit the program**.
   - 2.3 If the login exists in the database, but password doesn't match the record in database, please display an error message "**login xxxx exists, but password not matches.**" **and exit the program**.
   - 2.4 All other main items should be available only when the login and password match the record in the database which is considered login successfully.

3. User home page. After the user login successfully, the display the followings as shown in Figure 2.
   - 3.1 Display the customer's name, birthday and age, Address (street, city, zipcode) and IP address.
   - 3.2 Display the logged in customer's image and display the image under the Address.
   - 3.3 Display a message "You are (NOT) from Kean domain." under the IP address, if the customer (NOT) login from Kean domain (10.*.*.*) or (**131.125.*.***). You can use **lynx** command on obi.kean.edu to test the Kean domain.
   - 3.4 On the customer home page, add **additional 3** links - "**Logout**", "**Order Product**", and "**View, change, cancel my order history**".

4. **Order product** - After the customer clicks on the "**Order Product**" link on their home page, a program named "**order_product.php**" should be called to do the following functions. Please refer to Figure 3.
   - 4.1 The product's information should be shown in HTML <TABLE> format with headers.
   - 4.2 Display one row per product, showing the product ID, name, price, available quantity, vendor ID, and customer order quantity.
   - 4.3 Each product has its own input box for the customer to enter the quantity and a button to place order. The quantity box should be required to prevent the customer not entering any value to place an order.

5. **Place order**: After the customer enter the quantity and place the order, a program named "**place_order.php**" should be called to the following functions.
   - 5.1 If the input quantity is not an integer or is a negative number, please present the following message: "The order quantity must be a positive integer. The order has not been successfully placed." This order should NOT be inserted into your Order table.
   - 5.2 If the input quantity is a positive number and its value exceeds the available quantity of the corresponding product, please display a message "There is only *yy* quantity available. The order is not successfully". This order should NOT be inserted into your Order table.

5.3 If the input quantity does not have the errors in requirement 5.1 & 5.2, the order should be added into **your Order table**. All values cannot be NULL, empty, 0 and negative.

6.   **View order:** After the customer clicks the "**View, change, cancel my order history**" link on their home page, a program named "**view_order.php**" should be called to the following functions. Please refer to Figure 4.
   6.1 Display all the orders belonging to the currently logged-in customer in HTML <TABLE> format, complete with appropriate headers.
   6.2 Display one row per order, showing the order ID, Product name, price, available quantity, vendor ID, and customer order quantity.
   6.3 Each order has its own link for the customer to cancel that order.
   6.4 Each order has its own input box for the customer to enter the quantity and a button to change the order's quantity. The quantity box should be required to prevent the customer not entering any value to place an order.

7.   **Cancel order:** - After customer clicks the "**Cancel order**" **link** on their view order page, a program named "**cancel_order.php**" should be called to do the following functions. Please refer to Figure 4.
   7.1 If the order id belongs to the logined customer, that order should be removed from your **Order_xxxx** table, and a message "**Successfully deleted the order yy!**" should be displayed where yy is the order id.
   7.2 If the order id not exists, or not belongs to the logined customer, an error message "**The order id yy for customer id zz not exists, the order cannot be canceled.**" should be displayed where yy is the order id and zz is the customer id.

8.   **Change order**: After customer enters the quantity and clicks the "**Change quantity**" **button** on their view order page, a program named "**change_order.php**" should be called to do the following functions. Refer to Figure 4.
   8.1 If the order id belongs to the logined customer and the quantity value is not a positive integer (refer to 5.1), an error message "Please enter a positive integer.", and the quantity should NOT be changed in your Order table.
   8.2 If the new quantity is a positive number and its value exceeds the available quantity of the corresponding product, please display a message "There is only yy quantity available. The change is not successfully.", and the quantity should NOT be changed in your Order table.
   8.3 If the order id belongs to the logined customer and the quantity value does not have the errors in requirement 5.1 & 5.2, the order quantity should be changed to the new value in your **Order_xxxx** table, and a message "**Successfully changed the order yy!**" should be displayed where yy is the order id.
   8.4 If the order id not exists, or not belongs to the logined customer, an error message "The order id yy for customer id zz not exists, the order cannot be canceled." should be displayed where yy is the order id and zz is the customer id. No record should be changed.

Welcome to YOUR_NAME CPS3740 project.

View all customers
Search product name (enter * to show all products):
[   ] [Search]

Login ID: [huang   ]
Password: [•••] [Submit]

Figure 1. Class exercise - The index.html for project home page.

IP:67.69.23.120
You are not from Kean University.
Welcome customer: Mary1 Lee
Gender: M
DOB: 2000-01-01, age: 23
Address: 3 King St., Union, NJ, 07083

Logout
Order Product
View, change, cancel my order history

Figure 2. Class exercise - Customer home page

pid	Name	Price	Available QTY	vid	QTY to order
2001	iPod	200	11	1002	Place order
2002	chair	25	13	1001	Place order
2003	table	30	7	1001	Place order
2004	Bell	10	9	1003	Place order

Figure 3. Class exercise - Order product page. But students need to implement program - place_order.php.

order id	product name	order qty	date time	actions
3	iPod	4	2023-08-03 01:53:20	Cancel order Change quantity
4	iPod	4	2023-08-03 01:54:02	Cancel order Change quantity
5	iPod	4	2023-08-03 01:55:40	Cancel order Change quantity
2	chair	4	2023-08-03 01:03:38	Cancel order Change quantity

Figure 4. View order page should show the orders only placed the logined customer. Students need to implement 3 programs - view_ordr.php, cancel_order, change_order.php.