Project 1(E-commerce Concepts)

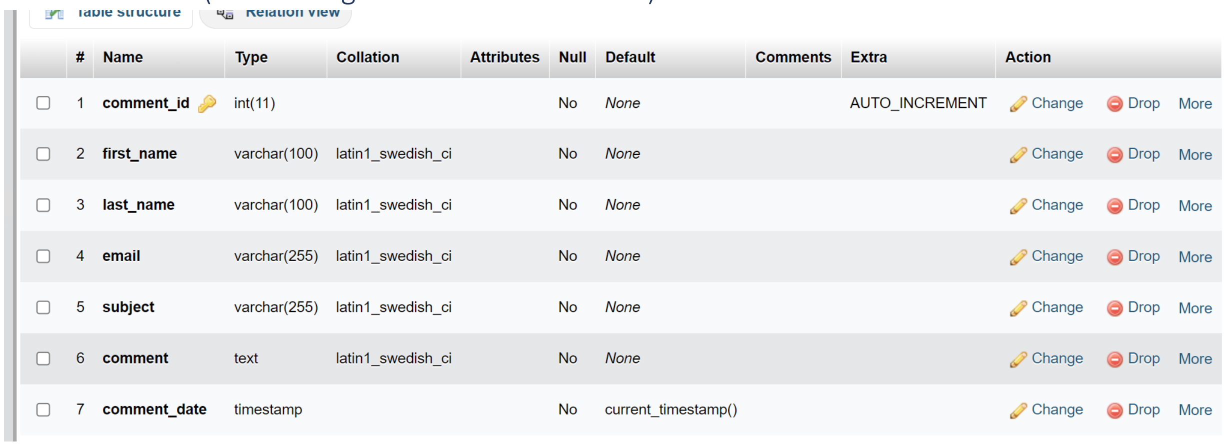
Online Management Systems MEDP-0105-02WBS (23.Fa) S. Taylor
WEB AND INTERACTIVE DEVELOPMENT
Hongxin Ouyang

Oct 6RD 2023

ASSINIBOINE COMMUNITY COLLEGE

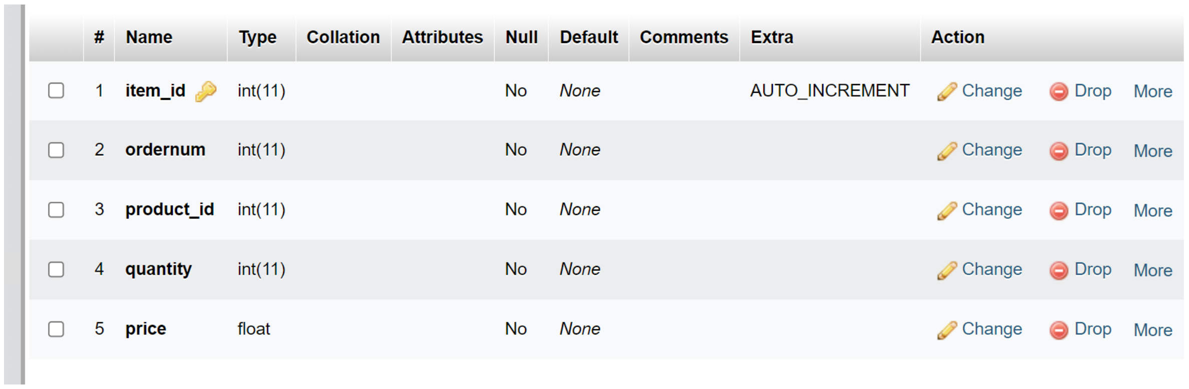
Database Schema

Comment table(for the original comment function)



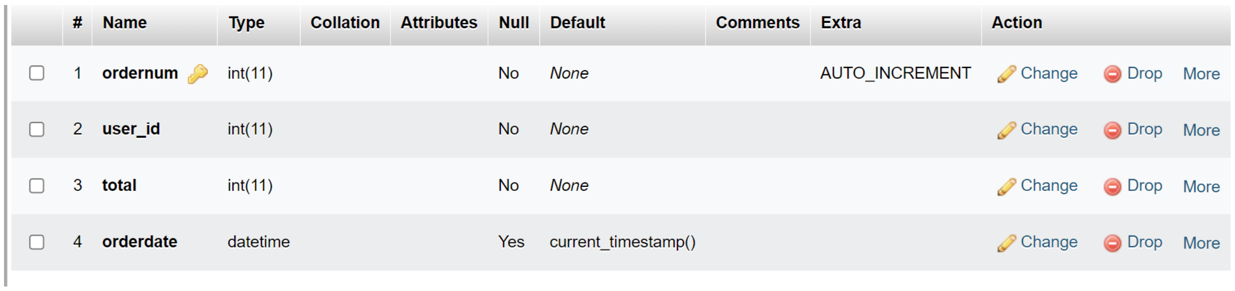
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
☐	1	**comment_id**	int(11)			No	*None*		AUTO_INCREMENT	Change	Drop	More
☐	2	**first_name**	varchar(100)	latin1_swedish_ci		No	*None*			Change	Drop	More
☐	3	**last_name**	varchar(100)	latin1_swedish_ci		No	*None*			Change	Drop	More
☐	4	**email**	varchar(255)	latin1_swedish_ci		No	*None*			Change	Drop	More
☐	5	**subject**	varchar(255)	latin1_swedish_ci		No	*None*			Change	Drop	More
☐	6	**comment**	text	latin1_swedish_ci		No	*None*			Change	Drop	More
☐	7	**comment_date**	timestamp			No	current_timestamp()			Change	Drop	More

Order Item table(items within each order)



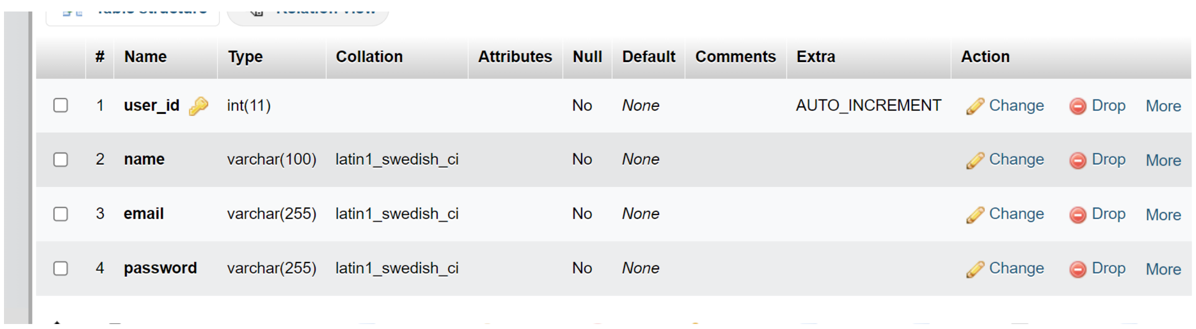
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**item_id** 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**ordernum**	int(11)			No	*None*			Change Drop More
☐	3	**product_id**	int(11)			No	*None*			Change Drop More
☐	4	**quantity**	int(11)			No	*None*			Change Drop More
☐	5	**price**	float			No	*None*			Change Drop More

Order table



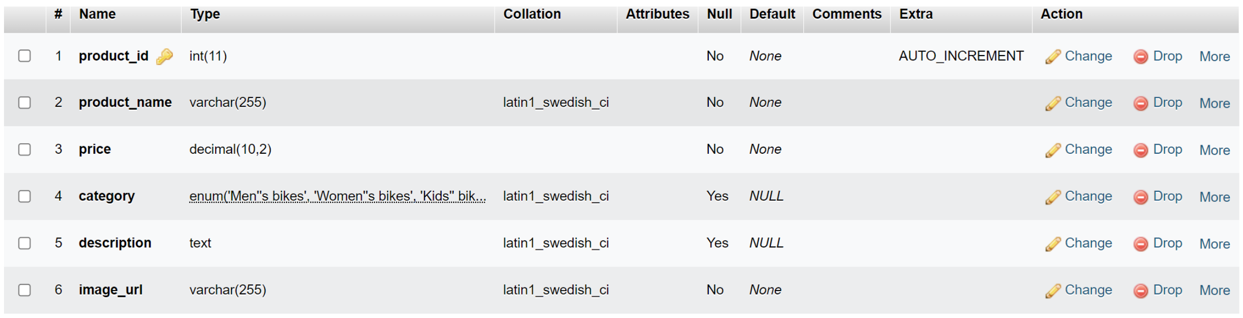
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**ordernum** 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**user_id**	int(11)			No	*None*			Change Drop More
☐	3	**total**	int(11)			No	*None*			Change Drop More
☐	4	**orderdate**	datetime			Yes	current_timestamp()			Change Drop More

User table



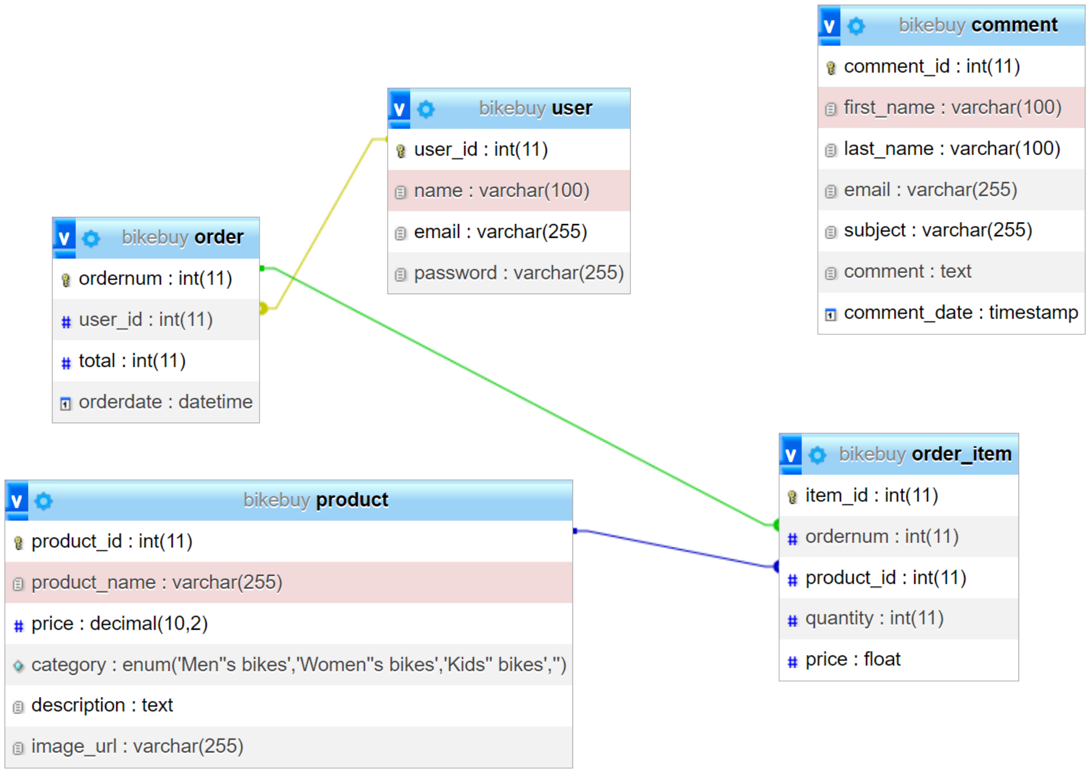
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**user_id** 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**name**	varchar(100)	latin1_swedish_ci		No	*None*			Change Drop More
☐	3	**email**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More
☐	4	**password**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More

Product table



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**product_id** 🔑	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**product_name**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More
☐	3	**price**	decimal(10,2)			No	*None*			Change Drop More
☐	4	**category**	enum('Men''s bikes', 'Women''s bikes', 'Kids'' bik...	latin1_swedish_ci		Yes	*NULL*			Change Drop More
☐	5	**description**	text	latin1_swedish_ci		Yes	*NULL*			Change Drop More
☐	6	**image_url**	varchar(255)	latin1_swedish_ci		No	*None*			Change Drop More

Database Relation



bikebuy user
user_id : int(11)
name : varchar(100)
email : varchar(255)
password : varchar(255)

bikebuy comment
comment_id : int(11)
first_name : varchar(100)
last_name : varchar(100)
email : varchar(255)
subject : varchar(255)
comment : text
comment_date : timestamp

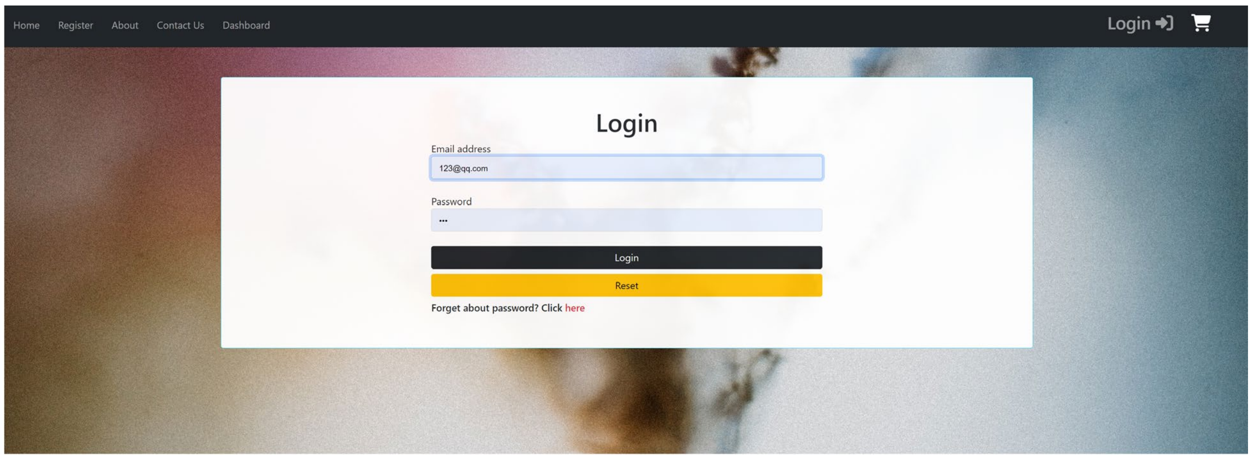
bikebuy order
ordernum : int(11)
user_id : int(11)
total : int(11)
orderdate : datetime

bikebuy order_item
item_id : int(11)
ordernum : int(11)
product_id : int(11)
quantity : int(11)
price : float

bikebuy product
product_id : int(11)
product_name : varchar(255)
price : decimal(10,2)
category : enum('Men''s bikes','Women''s bikes','Kids'' bikes','')
description : text
image_url : varchar(255)

Features:

Login page:



Login

Email address

123@qq.com

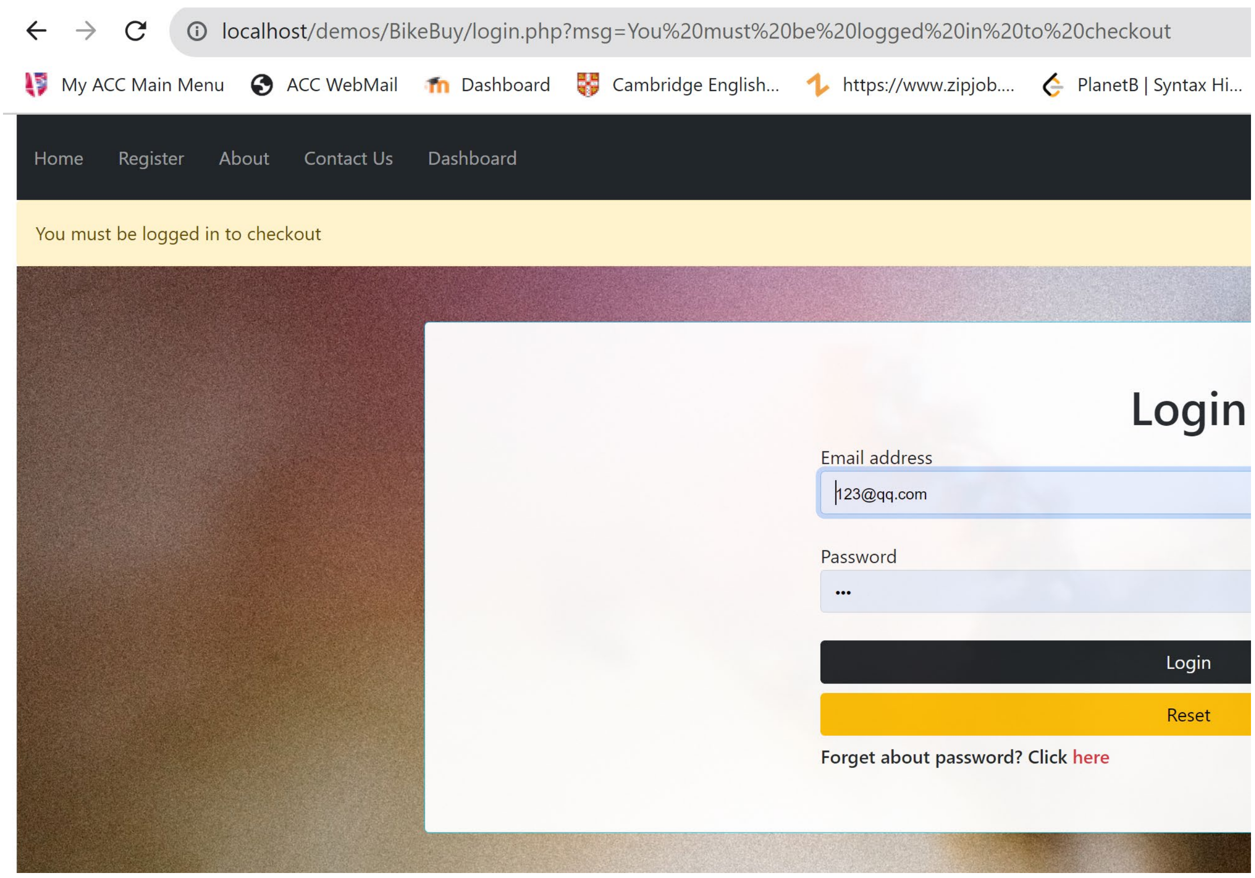
Password

...

Forget about password? Click here

A single login page with basic validation for login purposes.

Message box

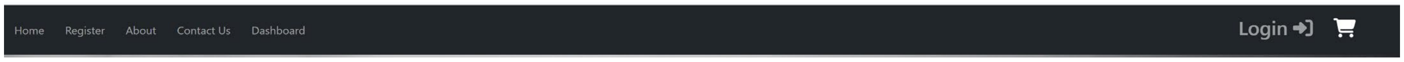


You must be logged in to checkout

Forget about password? Click here

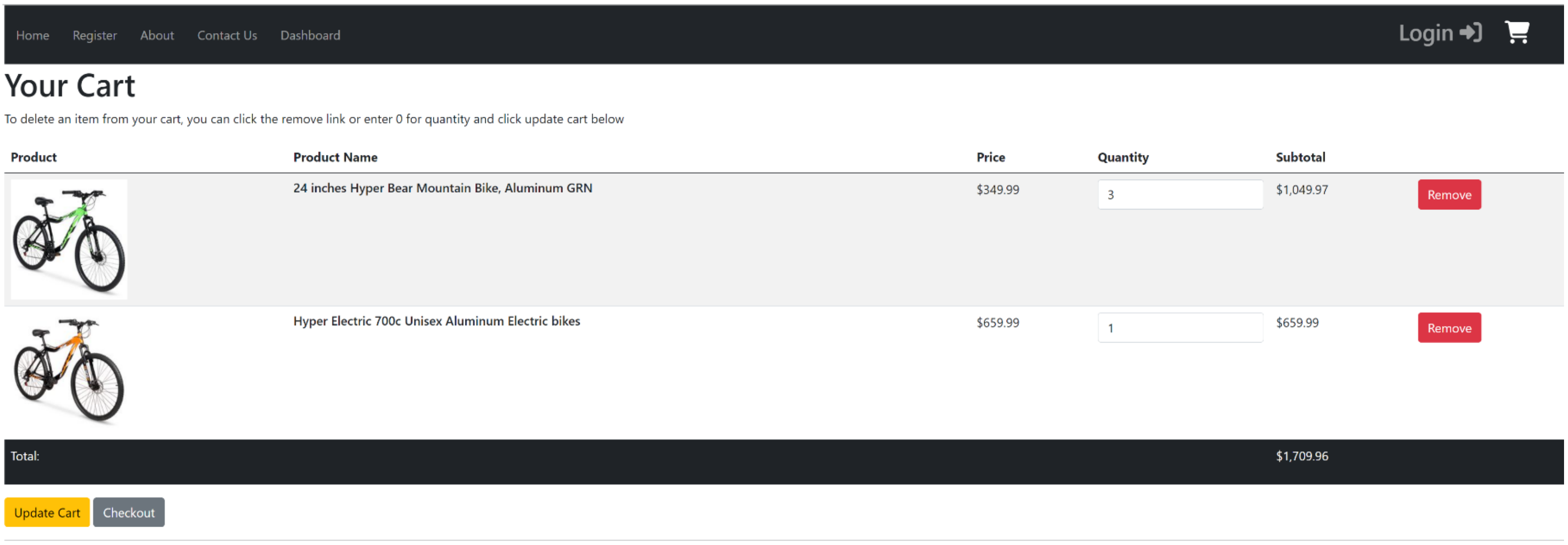
Message will appear at the top of the page(if there's any) as well as in the URL 'msg' dynamic parameter due to the unified message handling mechanism of this project.

Login/Logout and cart at the navigation bar



In the login section, the button will be adjusted to login/logout according to user state

Cart



Your Cart

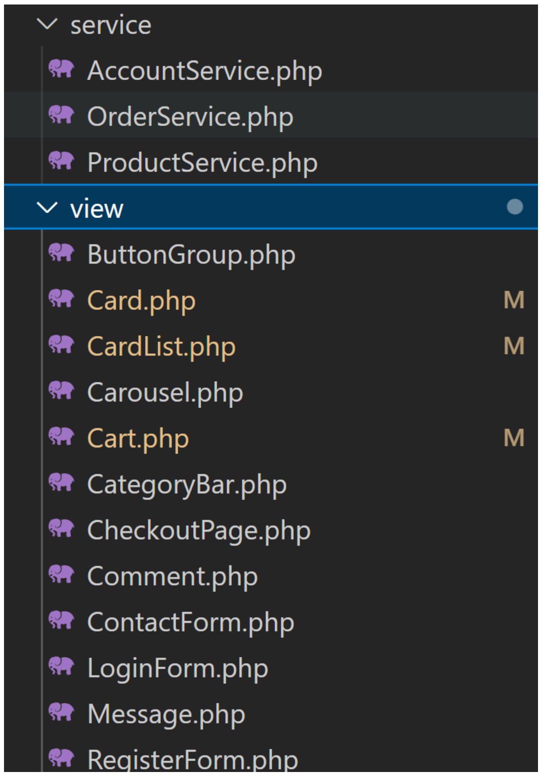
To delete an item from your cart, you can click the remove link or enter 0 for quantity and click update cart below

Product	Product Name	Price	Quantity	Subtotal	
	24 inches Hyper Bear Mountain Bike, Aluminum GRN	$349.99	3	$1,049.97	Remove
	Hyper Electric 700c Unisex Aluminum Electric bikes	$659.99	1	$659.99	Remove
Total:				$1,709.96	

A cart table displaying cart items with editing quantity, remove, and update cart functionality as well as total/subtotal information

Development Features

MVC architecture

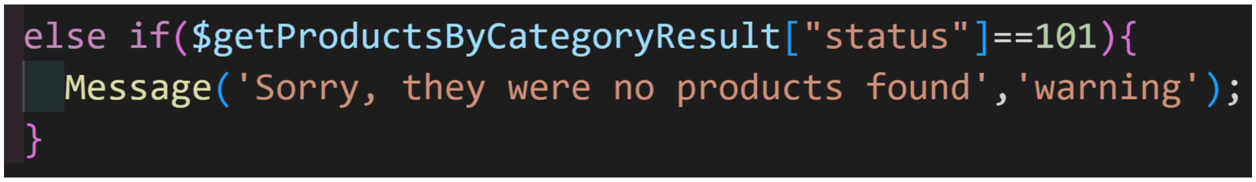


```
service
    AccountService.php
    OrderService.php
    ProductService.php
view
    ButtonGroup.php
    Card.php                M
    CardList.php            M
    Carousel.php
    Cart.php                M
    CategoryBar.php
    CheckoutPage.php
    Comment.php
    ContactForm.php
    LoginForm.php
    Message.php
    RegisterForm.php
```

Components on the webpage are rendered by view functions which are reusable and customizable. Meanwhile, all the database operations are entirely executed in the service layer. This is a good practice for decoupling.

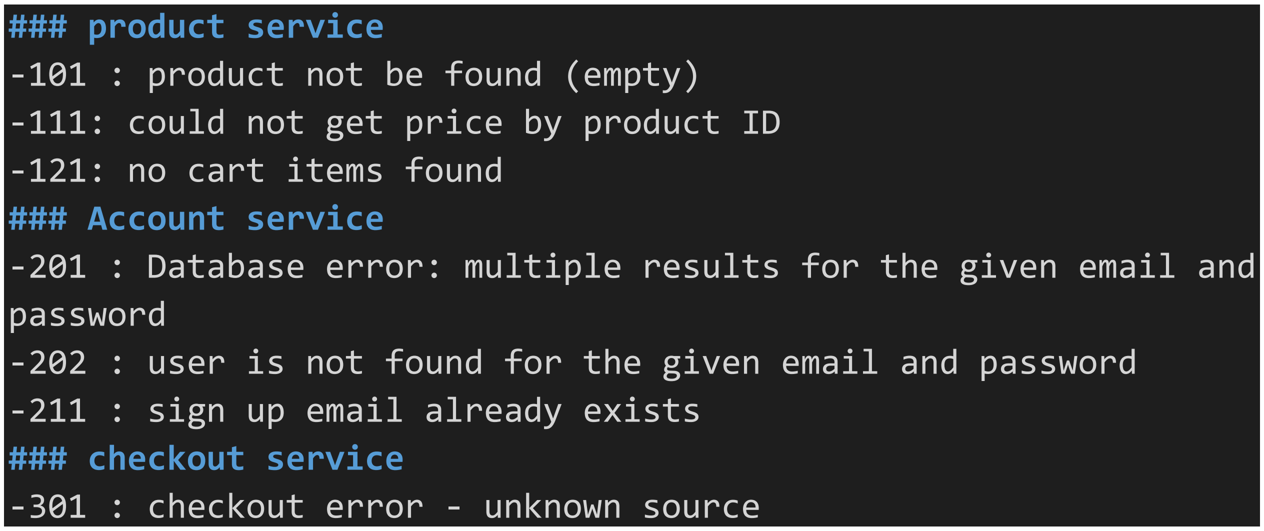
The PHP files in the root folder are considered controller files although I didn't specify explicitly.

Error handling



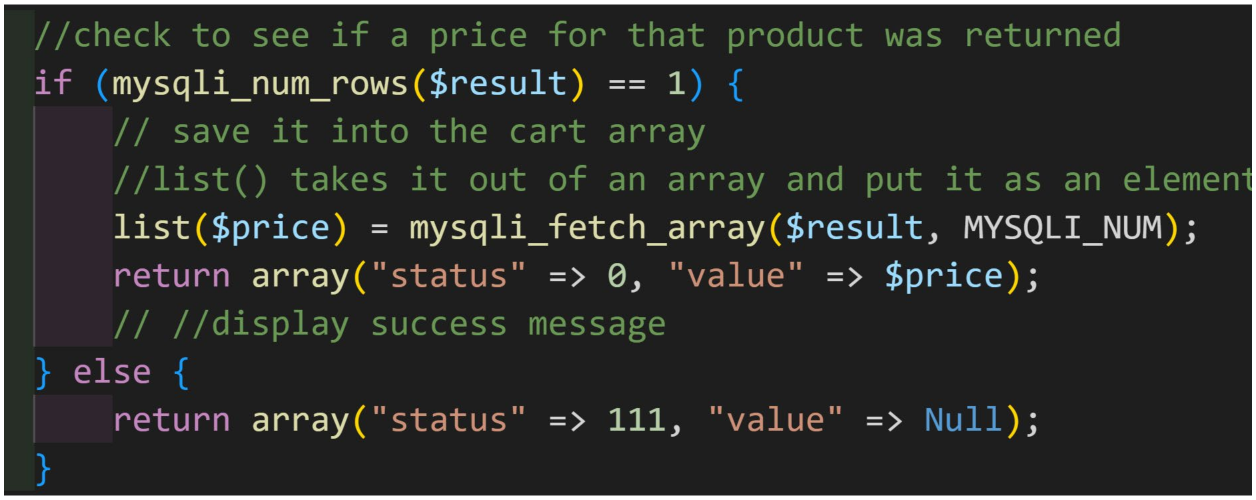
```
else if($getProductsByCategoryResult["status"]==101){
    Message('Sorry, they were no products found','warning');
}
```

All the error messages could be rendered to the front end and it would show up at the top of the webpage. Each error has a unique error code for it.



```
### product service
-101 : product not be found (empty)
-111: could not get price by product ID
-121: no cart items found
### Account service
-201 : Database error: multiple results for the given email and
password
-202 : user is not found for the given email and password
-211 : sign up email already exists
### checkout service
-301 : checkout error - unknown source
```

Each service function returns a status code('0' refers to success) and value


```
//check to see if a price for that product was returned
if (mysqli_num_rows($result) == 1) {
    // save it into the cart array
    //list() takes it out of an array and put it as an element
    list($price) = mysqli_fetch_array($result, MYSQLI_NUM);
    return array("status" => 0, "value" => $price);
    // //display success message
} else {
    return array("status" => 111, "value" => Null);
}
```

Conclusion

what I feel that I did well:

I didn't use any backend framework. However, my project follows controller-view-service architecture so the components are decoupled(it appeared as if I used some framework). I don't need to mess up HTML, PHP, and SQL together. My project is easy to maintain once read through it, easy to debug, and scalable.

what I think I can improve on:

I can use some lightweight frameworks such as vue.js to render dynamic elements on the page, especially the cart page. However, I did not have enough time to do it.

I certainly could refine the details of it and add more features. However, there's no end to this process.

Reference:

Login background:

https://www.pexels.com/photo/plant-on-fire-3357695/

Checkout background

https://www.pexels.com/photo/man-facing-road-1248418/