Swinburne University of Technology

School of Science, Computing and Engineering Technologies

ASSIGNMENT AND PROJECT COVER SHEET

Subject Code: SWE30003 Unit Title: Software Architectures and Design

Assignment number and title: 3, Design Implementation Due date: 5th April 2024

Tutorial Day and Time: Tuesday 9:00AM Original Project Group: 9

Lecturer: Prof Jun Han Tutor: Ms. Mandeep Dhindsa

To be completed as this is a group assignment:

We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgment is made explicitly in the text, nor has any part been written for us by another person.

ID Number	Name	Signature
103803891	Tran Quoc Dung	Dung
103843282	Pham Hoang Duong	Duong
103806531	Nguyen Thai Son	NTSon
103792724	Nguyen Manh Duc	ManhDuc

Marker's comments:

Total Mark:__________________

Extension certification:

This assignment has been given an extension and is now due on __________________________

Signature of Convener:_______________________

Table of Contents

Introduction

This assignment builds upon our initial object-oriented design for an online healthy food store website. We'll take a deep dive, refining the design for direct implementation. User interaction will be facilitated through UI elements, and an object-oriented database will be designed. Any adjustments made during this process will be meticulously documented. To ensure the website functions as intended, we'll establish a comprehensive testing strategy with detailed test cases. These test cases will validate the website's functionalities and guarantee a smooth user experience.

1. Design Revision

	Assignment 2	**Assignment 3**
Problem Analysis	Summarize the goals, assumptions, simplification, along with noting the discarded class	- The goals, assumptions and simplifications are quite clear. However, some adjustments are still needed to be more comprehensive. - However, we decided not to discard the Cart class, which we thought is not necessary.
Candidate Class Lists	Customer, Staffs, Product, Order, Invoice, Receipt, Payment, Shipping, Database, Account, Menu	Discarding Menu class, and adding Cart class, while keeping the maintained classes.
UML Diagram	Illustrating how the classes in our webpage are connected, using 4 types of relation: *Directional*, *Aggregation*, *Composition* and *General*.	- Our group decided to simplify the UML diagram with only directional relation, which display the process of a customer that accessing the webpage. - Also adding a Cart class into the UML for updating.

CRC Cards	Containing list of 11 classes	Adding Cart, discarding Menu as Navigation Pane, and adjusting the connections between several classes.
Design Heuristics	Maintaining the total of 9 design heuristics (keeping the original wordings)	
Design Patterns	3 types of patterns, with total of 5 unique patterns	- Truly applying Factory Method Pattern to the Generalization hierarchy, which belong to Account class. - Adjust to applying Builder Pattern to Product Class instead of Order class, based on feedback - Adjust to utilizing Composite Pattern to design the class for Invoice – Payment classes.
Bootstrap Process	Nothing changes much, except adding Cart class.	
Verifications	Adjusting Verification 1 & 4 (Step 3 & 7)	

(Observe full Assignment 2 in Appendices)

2. Detailed Design

UX/UI Design:

- Model-View-Controller (MVC) Pattern: Separate the application logic (Model - Classes), user interface, and user interaction handling (Controller - server-side scripts).
- Views: Design user interfaces for the following functionalities:
    - Product Browsing: Allow users to search and filter products by category, name, or dietary needs. Display product information with clear images, descriptions, prices.
    - Shopping Cart Management: Enable users to add, remove, and update quantities of products in their cart. Display a clear overview of the cart contents, including product information, subtotal, and potential discounts.
    - Checkout Process: Guide users through a streamlined checkout process, including:
    - Login or Account Creation: Allow users to log in to existing accounts or create new accounts.
    - Shipping Information: Collect user's shipping address and offer options for different delivery methods with corresponding costs.
    - Payment Details: Provide a secure interface for users to enter payment information (credit card, etc.) with proper validation checks.
    - Order Confirmation: Display a clear summary of the order details (products, prices, shipping, total) for final confirmation before payment.
    - Order History: Allow users to view their past orders, including order status, tracking information (if applicable), and the option to reorder items.

Database Design:

1. Entity-Relationship Diagram (ERD): Create an ERD to represent the relationships between tables (classes) in the database. Each class will map to a corresponding table with relevant attributes as columns.
2. Data Types: Define appropriate data types for each attribute in the database tables based on the information they store (e.g., strings for names, integers for quantities).
3. Relationships: Establish relationships between tables using primary and foreign keys. For example, the `Order` table might have a foreign key referencing the `Customer` table's primary key to link orders to specific customers.

Bootstrap and Verification Process Diagrams

- Bootstrap Process Diagram: This diagram could illustrate the steps a customer takes while browsing products, adding items to their cart, and proceeding to checkout.
- Verification Process Diagrams: These diagrams could depict specific verification steps within the application, such as: Manage the shopping cart, create invoices and receipts for

customers, handle payment securely and record statistics of goods sold based on receipts over various periods.

Reflection on Initial Design:

+) Strengths of the Initial Design:

- Identified Core Classes: The initial design effectively identified the core classes needed for the online store, including Customer, Staff, Product, Order, Invoice, Receipt, Payment, Shipping, Database and Account.
- Outlined Functionalities: It provided a high-level overview of the functionalities required in the online store.
- Class Relationships: The initial design included a UML class diagram that depicted the basic relationships between the classes.

+) Missing Aspects of the Initial Design:

- Class Details: The design lacked details on class attributes, methods, data validation, error handling, and database interactions.
- UI Design: There was no consideration for the user interface (UI) design, which is crucial for user interaction with the online store.

Ambiguity and Interpretation: The initial design, while providing a good starting point, required some interpretation during the detailed design phase. This involved defining specific attributes and methods for each class, designing the user interface elements.

Justification for Changes:

Class attributes and methods define functionalities, UI design guides user interaction, and database design facilitates data storage and retrieval in an object-oriented manner.

Lessons Learned:

- Initial Design Importance: While high-level design provides a starting point, detailed design refines it for practical implementation.
- Iterative Process: Design can evolve during development as specific requirements become clearer.
- Object-Oriented Principles: Applying MVC and ERD ensures a well-structured, maintainable system.

Future Design Approach:

For future high-level designs, a more detailed breakdown of class functionalities with potential attributes and methods would be beneficial. Additionally, considering basic UI mockups and an outline of database schema early on can lead to a more comprehensive initial design.

3. Design Quality

Design Heuristics

Inheritance: Common data and behaviors should reside high in the hierarchy for efficient reuse.
Encapsulation: Data should be encapsulated within classes with controlled access.

- Classes should avoid excessive object composition.
- Utilize class-level variables/methods instead of global.
- Derived classes shouldn't override inherited methods with empty implementations.

Abstraction: Base classes should ideally be abstract, focusing on defining core functionalities.
Avoiding God Classes: Strive for smaller, well-defined classes with specific responsibilities.
Reusability: Design for code that can be easily adapted and reused in different contexts.
Action vs. Object: Differentiate between actions (verbs) and objects (nouns) when creating classes.

Design Patterns

Factory Method Pattern: Enables creation of different product types based on categories.
Builder Pattern: Facilitates constructing complex objects like orders or invoices with flexibility.
Observer Pattern: Allows notifying customers about order status changes.
Composite Pattern: Represents product categories with a hierarchical structure.
Model-View-Controller (MVC) Pattern: Separates data (Model), presentation (View), and business logic (Controller) for maintainability.
Bootstrap process:
- Clear user flow: The process clearly defines the steps a customer takes to browse, select products, purchase them, and receive their order. This provides a user-centric perspective on the system's functionality.
- Class functionality integration: The process highlights how different classes interact to facilitate each step. This reinforces the understanding of how the overall system works together.

- Focus on security: The inclusion of secure payment handling demonstrates an awareness of critical security measures in e-commerce transactions.

4. Implementation

4.1. Mapping Design to Code

4.1.1. Explanation of translation to code

Although our project designation followed Object-Oriented Design (OOD), we decided not to develop the website with Object-Oriented Programming for some purpose.

- When translating the Class diagram into real Class in the code, the transition is detailed as following:
  - Staff or Admin is the class that has a lot of control with the ability to see how many customers are there, different accounts existing, orders placed. Staff can also register for new staff account, delete old ones.
  - Menu contains all products available and their details for the user to see to place orders on the products there.
  - Customers (users) class represents the account for the users that need to be registered to perform functions like taking orders, adding to cart, the account can be tracked by Staff.
  - Product class containing product information such as name, associated images, price and is added in through Staff and can be seen by Customers (Users) through the menu.
  - Orders class is the main way for customers to order Product from the menu of the store, product that is ordered will appear in cart first before orders are processed. Orders can be tracked by Staff.
  - Cart class contains orders after they are placed by the users that after which can be processed to check out to get the product.
- When creating the database in the backend, it is done so:
  - The database is created with 4 different tables: cart, orders, users, products, admin.
  - Each row of Cart contains attributes such as id, product id, user id, product cost, product image and product quantity.
  - Each row of Orders contains attributes such as id, user id, date, total price, total products, payment method, address and email.
  - Each row of Users contains attributes such as id, name, email, address, and password.
  - Each row of Products contains attributes such as id, name, category, product cost and image.
  - Each row of Admin contains attributes such as id, password, and name (this password is encrypted)
- When adding some UI into the webpage, these are considered:

- There’s 2 separated UI, one for the users which is the main webpage, and the other is for the admin which can’t be accessed without logging in with an appropriate admin account and both have a login, register page and profile page that they can change their account name, password, etc.
- Admin dashboard has 5 main pages: home, product, orders, admin, and users with home linking to the other pages and they show details about existing admin, users account, orders placed with product allowing the admin to add products into the store.
- Store dashboard has 4 main pages: home, about, menu and orders with menu displaying most of the products available or added in the admin dashboard and orders show orders made within that user account.

4.1.2. Consideration of Coding Standards

We use some coding standards as following to make sure all developers contributing the project make unified code:

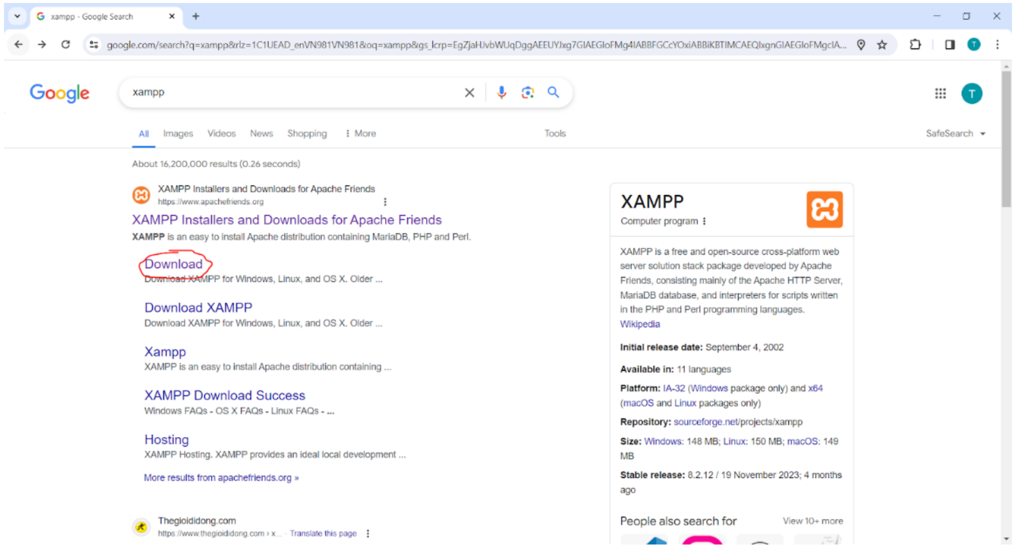
- Big containers like “<div>” tag should be separated from each other with an empty line using “space” to better manage each component.
- Indentation of the code should start from the furthest left and with each block that comes after but inside the previous block be indented a bit to the right (applied mostly to nested code)
- Variable name that is made up of more than 1 word is concatenated with “_”
- HTML tag with attributes containing value inside double quotations, PHP code use single quotations instead for all types (string, echoing HTML elements, etc.)
- Comments added for HTML elements in PHP file to know which is corresponding to what in the website.
- use PDO object instead of MySQL for database connectivity for better way to separate query called and variable passed into query.
- Website headers and footers is separated into a different PHP file and is called whenever needed to reduce code redundancy, same applies for PHP file to connect into database.

4.2. Compilation and Execution

4.2.1. Setup and Running Instructions

Setting up XAMPP for accessing localhost back-end website.

Our code uses PHP - a server-sided language that integrates with MySQL database, therefore, a local server needs to be installed to run the code.
First, search for XAMPP on Google then click “Download” from this result.
(Download link in References)



xampp - Google Search

google.com/search?q=xampp&rlz=1C1UEAD_enVN981VN981&oq=xampp&gs_lcrp=EgZjaHJvbWUqDggAEEUYJxg7GIAEGIoFMg4IABBFGCcYOxiABBiKBTIMCAEQIxgnGIAEGIoFMgcIA...

xampp

All Images Videos News Shopping More Tools SafeSearch

About 16,200,000 results (0.26 seconds)

XAMPP Installers and Downloads for Apache Friends
https://www.apachefriends.org

XAMPP Installers and Downloads for Apache Friends
XAMPP is an easy to install Apache distribution containing MariaDB, PHP and Perl.

Download
Download XAMPP for Windows, Linux, and OS X. Older ...

Download XAMPP
Download XAMPP for Windows, Linux, and OS X. Older ...

Xampp
XAMPP is an easy to install Apache distribution containing ...

XAMPP Download Success
Windows FAQs - OS X FAQs - Linux FAQs - ...

Hosting
XAMPP Hosting. XAMPP provides an ideal local development ...

More results from apachefriends.org »

Thegioididong.com
https://www.thegioididong.com › x... · Translate this page

XAMPP
Computer program

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages. Wikipedia

Initial release date: September 4, 2002

Available in: 11 languages

Platform: IA-32 (Windows package only) and x64 (macOS and Linux packages only)

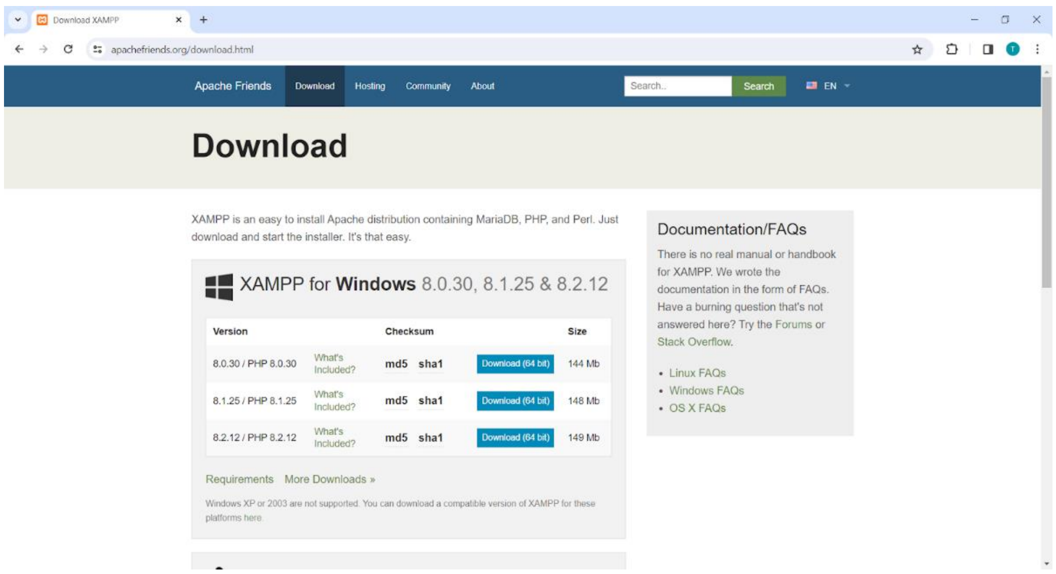
Repository: sourceforge.net/projects/xampp

Size: Windows: 148 MB; Linux: 150 MB; macOS: 149 MB

Stable release: 8.2.12 / 19 November 2023; 4 months ago

People also search for View 10+ more

XAMPP comes with PHP when installed so choose the PHP version.



Download XAMPP

apachefriends.org/download.html

Apache Friends Download Hosting Community About Search EN

Download

XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy.

XAMPP for **Windows** 8.0.30, 8.1.25 & 8.2.12

Version		Checksum			Size
8.0.30 / PHP 8.0.30	What's Included?	md5	sha1	Download (64 bit)	144 Mb
8.1.25 / PHP 8.1.25	What's Included?	md5	sha1	Download (64 bit)	148 Mb
8.2.12 / PHP 8.2.12	What's Included?	md5	sha1	Download (64 bit)	149 Mb

Requirements More Downloads »

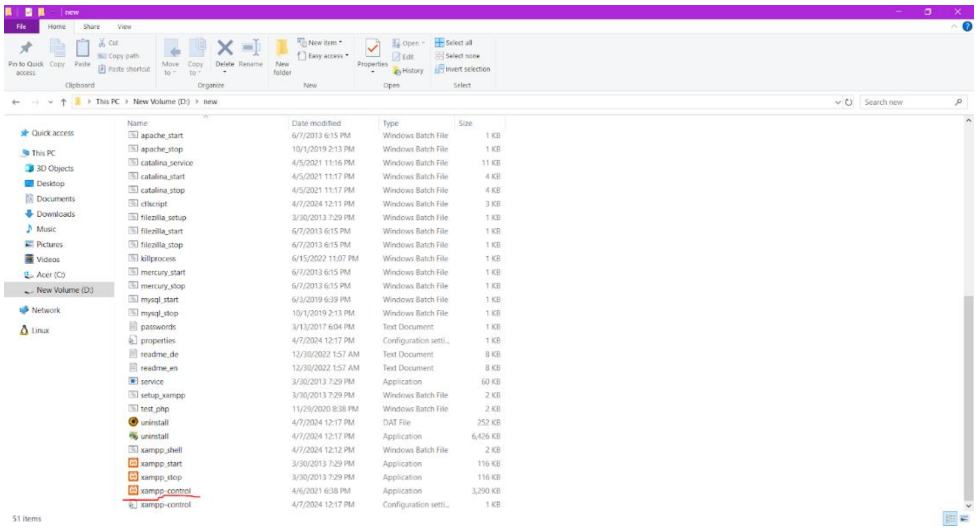
Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms here.

Documentation/FAQs

There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or Stack Overflow.

- Linux FAQs
- Windows FAQs
- OS X FAQs

In the Download tab from File Explorer, choose the XAMPP installer that is downloaded. Once inside, click “Next” and choose your file path then keep clicking until XAMPP is being installed. Once finished, go into the path that you installed XAMPP on, and find “xampp-control” and click on it to launch the XAMPP control panel.

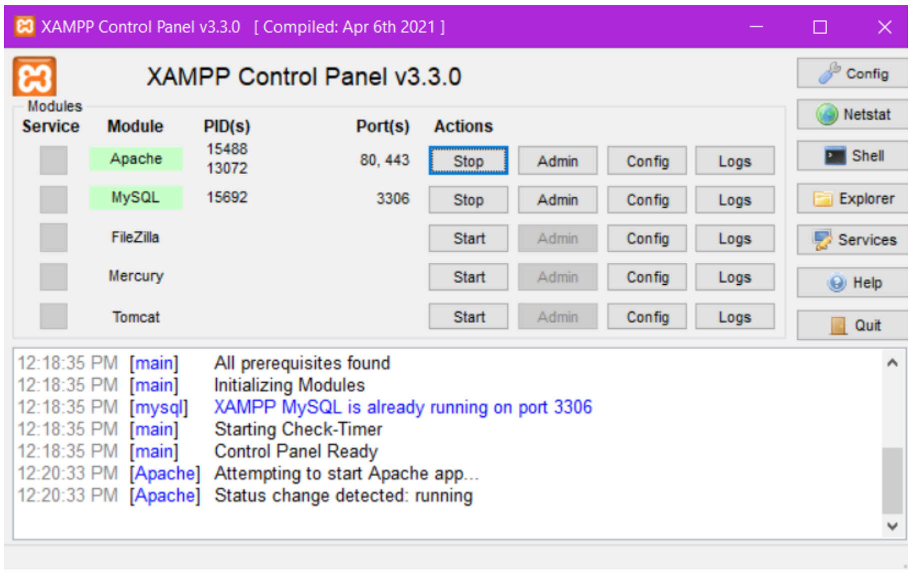


This PC > New Volume (D:) > new

Name	Date modified	Type	Size
apache_start	6/7/2013 6:15 PM	Windows Batch File	1 KB
apache_stop	10/1/2019 2:13 PM	Windows Batch File	1 KB
catalina_service	4/5/2021 11:16 PM	Windows Batch File	11 KB
catalina_start	4/5/2021 11:17 PM	Windows Batch File	4 KB
catalina_stop	4/5/2021 11:17 PM	Windows Batch File	4 KB
ctlscript	4/7/2024 12:11 PM	Windows Batch File	3 KB
filezilla_setup	3/30/2013 7:29 PM	Windows Batch File	1 KB
filezilla_start	6/7/2013 6:15 PM	Windows Batch File	1 KB
filezilla_stop	6/7/2013 6:15 PM	Windows Batch File	1 KB
killprocess	6/15/2022 11:07 PM	Windows Batch File	1 KB
mercury_start	6/7/2013 6:15 PM	Windows Batch File	1 KB
mercury_stop	6/7/2013 6:15 PM	Windows Batch File	1 KB
mysql_start	6/3/2019 6:39 PM	Windows Batch File	1 KB
mysql_stop	10/1/2019 2:13 PM	Windows Batch File	1 KB
passwords	3/13/2017 6:04 PM	Text Document	1 KB
properties	4/7/2024 12:17 PM	Configuration setti...	1 KB
readme_de	12/30/2022 1:57 AM	Text Document	8 KB
readme_en	12/30/2022 1:57 AM	Text Document	8 KB
service	3/30/2013 7:29 PM	Application	60 KB
setup_xampp	3/30/2013 7:29 PM	Windows Batch File	2 KB
test_php	11/29/2020 8:38 PM	Windows Batch File	2 KB
uninstall	4/7/2024 12:17 PM	DAT File	252 KB
uninstall	4/7/2024 12:17 PM	Application	6,426 KB
xampp_shell	4/7/2024 12:12 PM	Windows Batch File	2 KB
xampp_start	3/30/2013 7:29 PM	Application	116 KB
xampp_stop	3/30/2013 7:29 PM	Application	116 KB
xampp-control	4/6/2021 6:38 PM	Application	3,290 KB
xampp-control	4/7/2024 12:17 PM	Configuration setti...	1 KB

51 items

Once the control panel is opened, activate “Apache” and “MySQL” to run the website and connect to the database.



XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

Service	Module	PID(s)	Port(s)	Actions			
	Apache	15488 13072	80, 443	Stop	Admin	Config	Logs
	MySQL	15692	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
12:18:35 PM  [main]     All prerequisites found
12:18:35 PM  [main]     Initializing Modules
12:18:35 PM  [mysql]    XAMPP MySQL is already running on port 3306
12:18:35 PM  [main]     Starting Check-Timer
12:18:35 PM  [main]     Control Panel Ready
12:20:33 PM  [Apache]   Attempting to start Apache app...
12:20:33 PM  [Apache]   Status change detected: running
```

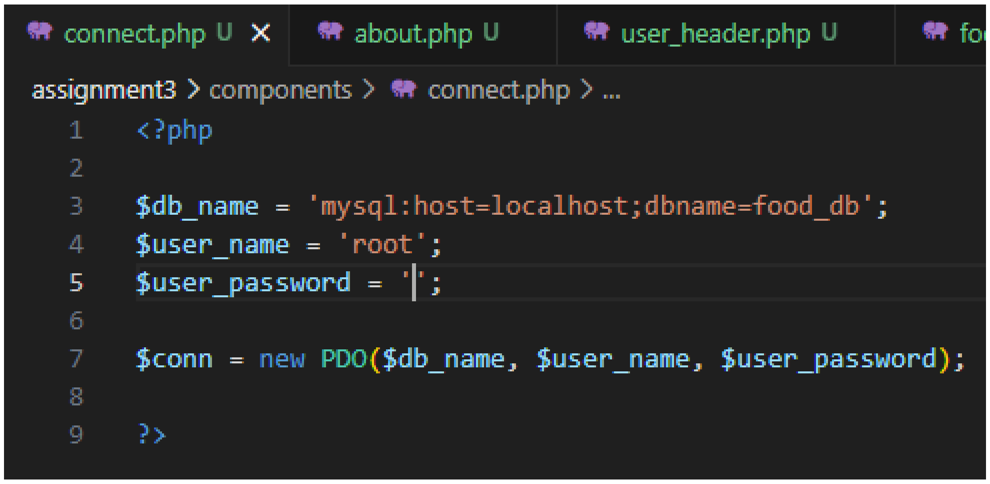
Go back to XAMPP installation path and choose *htdocs* folder and create a new folder to add in the code from our website (the folder of our team is “assgiment3coding” that will contain the code)

Accessing the Website

Step 1: Opening file connect.php (Location path: assignment3coding/components/connect.php)

Step 2: Adjusting username and password of your localhost phpMyAdmin.

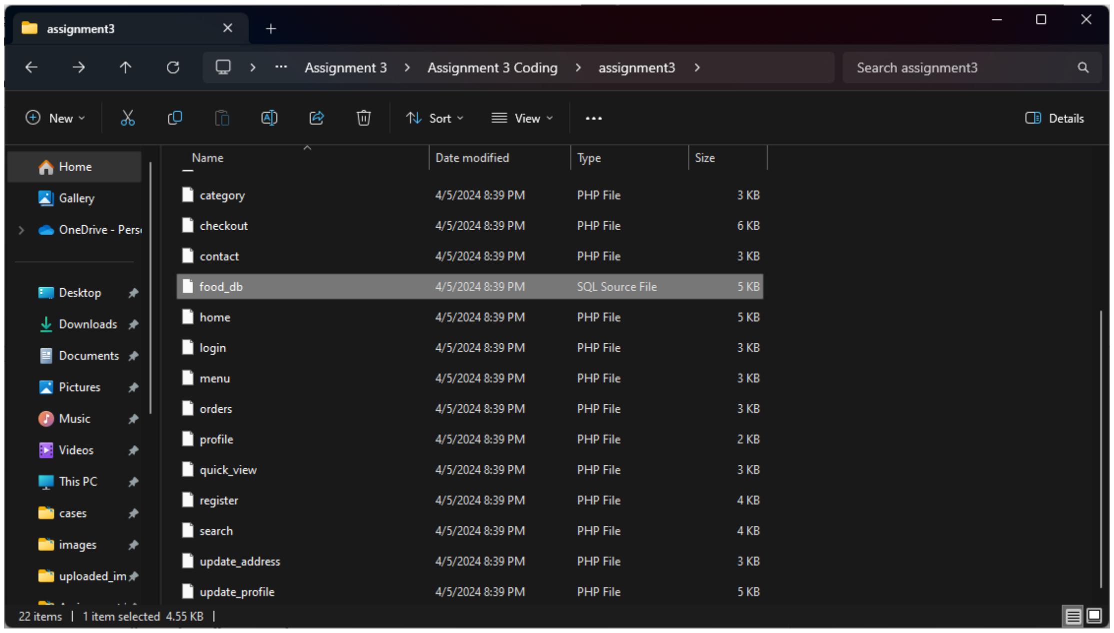
(since no password is created, keep as following like in the screen)



```php
<?php

$db_name = 'mysql:host=localhost;dbname=food_db';
$user_name = 'root';
$user_password = '';

$conn = new PDO($db_name, $user_name, $user_password);

?>
```

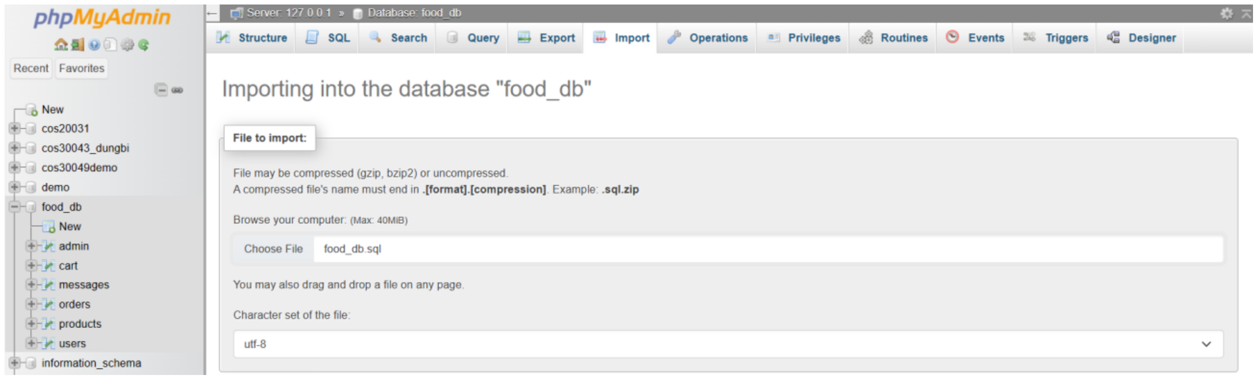
NOTE: You can see your username and password of localhost phpMyAdmin in the file config.inc.php (Location path: xampp/phpMyAdmin/config.inc.php)

Step 3: Searching for SQL file *food_db* to import into localhost phpMyAdmin.



Name	Date modified	Type	Size
category	4/5/2024 8:39 PM	PHP File	3 KB
checkout	4/5/2024 8:39 PM	PHP File	6 KB
contact	4/5/2024 8:39 PM	PHP File	3 KB
food_db	4/5/2024 8:39 PM	SQL Source File	5 KB
home	4/5/2024 8:39 PM	PHP File	5 KB
login	4/5/2024 8:39 PM	PHP File	3 KB
menu	4/5/2024 8:39 PM	PHP File	3 KB
orders	4/5/2024 8:39 PM	PHP File	3 KB
profile	4/5/2024 8:39 PM	PHP File	2 KB
quick_view	4/5/2024 8:39 PM	PHP File	3 KB
register	4/5/2024 8:39 PM	PHP File	4 KB
search	4/5/2024 8:39 PM	PHP File	4 KB
update_address	4/5/2024 8:39 PM	PHP File	3 KB
update_profile	4/5/2024 8:39 PM	PHP File	5 KB

Step 4: Create a new database named “*food_db*”, then import the file.



Importing into the database "food_db"

File to import:

File may be compressed (gzip, bzip2) or uncompressed.
A compressed file's name must end in **.[format].[compression]**. Example: **.sql.zip**

Browse your computer: (Max: 40MiB)

Choose File food_db.sql

You may also drag and drop a file on any page.

Character set of the file:

utf-8

Step 5: Now, the website is ready and to be accessed, you click on an URL in your browser as following: “localhost/SWE30003_assignment3_coding” to access main website.

Step 6:
To observe **customer’s perspective**, access the PHP files inside the folder “htdocs/SWE30003_assignment3_coding”.
URL example: localhost/SWE30003_assignment3_coding/home.php

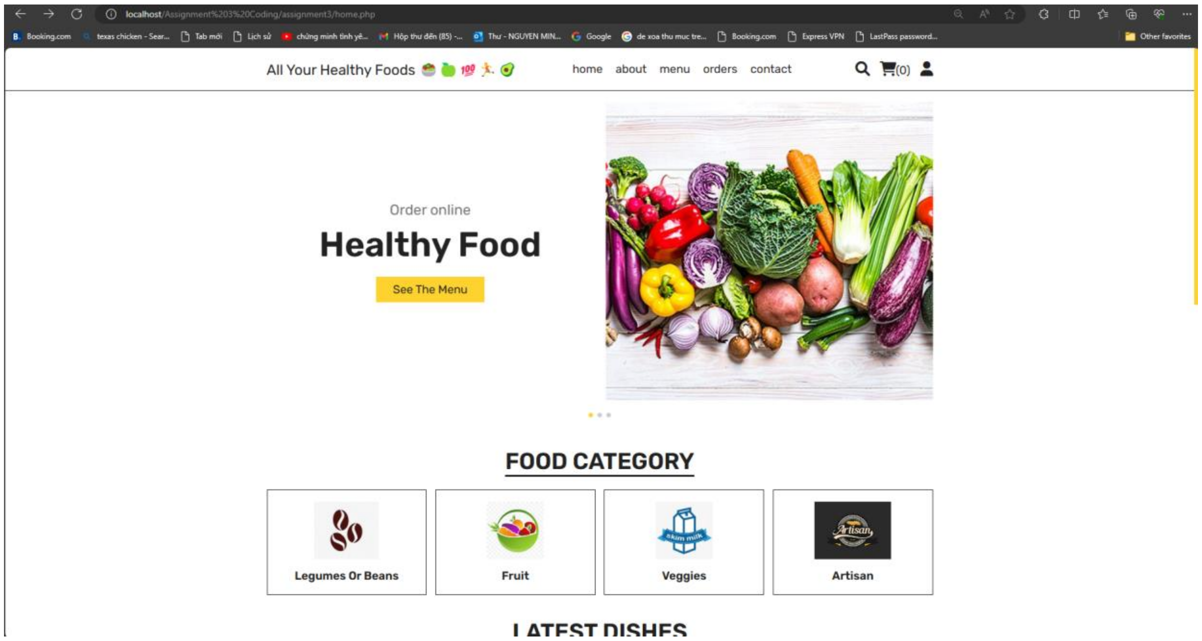
To observe **staffs, developers’ perspective**, access the PHP files inside the folder “htdocs/SWE30003_assignment3_coding/admin”.
URL example: localhost/SWE30003_assignment3_coding/admin/dashboard.php

4.2.2. Compilation and Execution Process

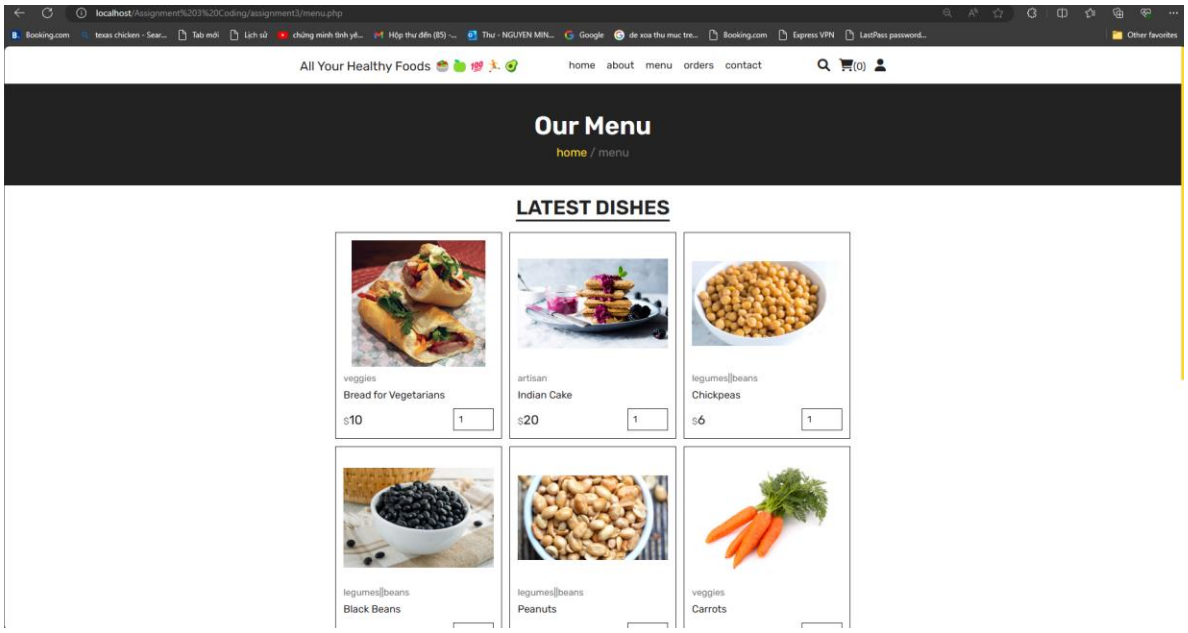
4.2.2.1. ONLINE CUSTOMERS's VIEWPOINT

Scenario 1: Accessing informative pages.

Accessing Home page (*home.php*)



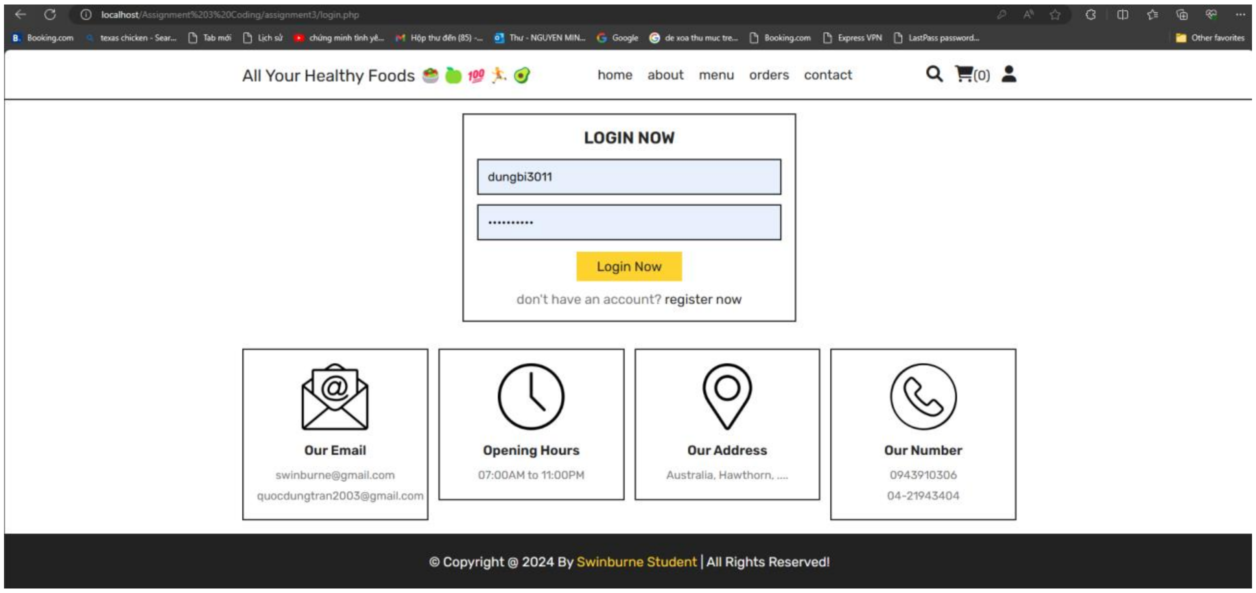
About (*about.php*) and Product page (*menu.php*) can be accessed easily without logging in.



Scenario 2: Observing the Orders.

To access Order page, which display all the Purchased Order, users must login.

Before logging in:


localhost/Assignment%203%20Coding/assignment3/login.php

All Your Healthy Foods

home about menu orders contact (0)

LOGIN NOW

dungbi3011

••••••••••

Login Now

don't have an account? register now

Our Email
swinburne@gmail.com
quocdungtran2003@gmail.com

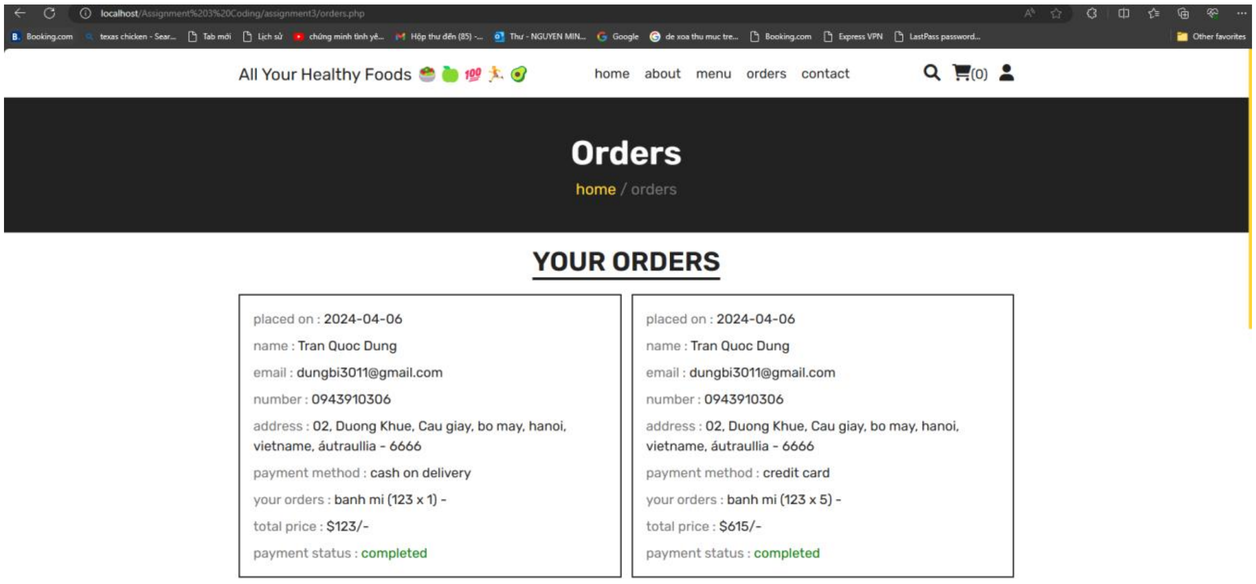
Opening Hours
07:00AM to 11:00PM

Our Address
Australia, Hawthorn, ...

Our Number
0943910306
04-21943404

© Copyright @ 2024 By Swinburne Student | All Rights Reserved!

After logging in:


localhost/Assignment%203%20Coding/assignment3/orders.php

All Your Healthy Foods

home about menu orders contact (0)

Orders

home / orders

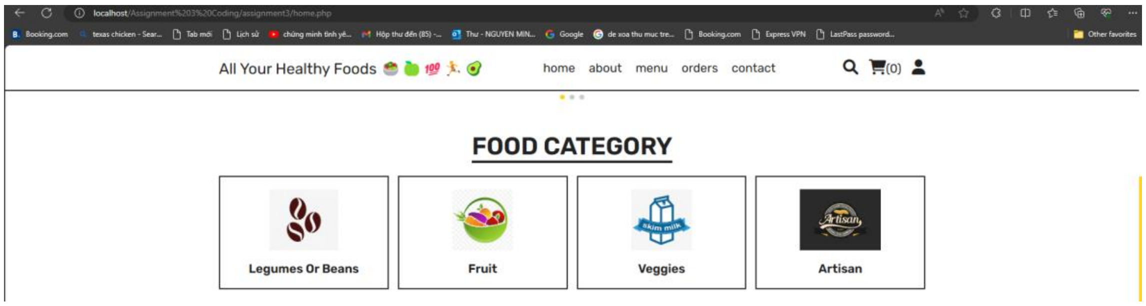
YOUR ORDERS

placed on : 2024-04-06
name : Tran Quoc Dung
email : dungbi3011@gmail.com
number : 0943910306
address : 02, Duong Khue, Cau giay, bo may, hanoi, vietname, áutraullia - 6666
payment method : cash on delivery
your orders : banh mi (123 x 1) -
total price : $123/-
payment status : completed

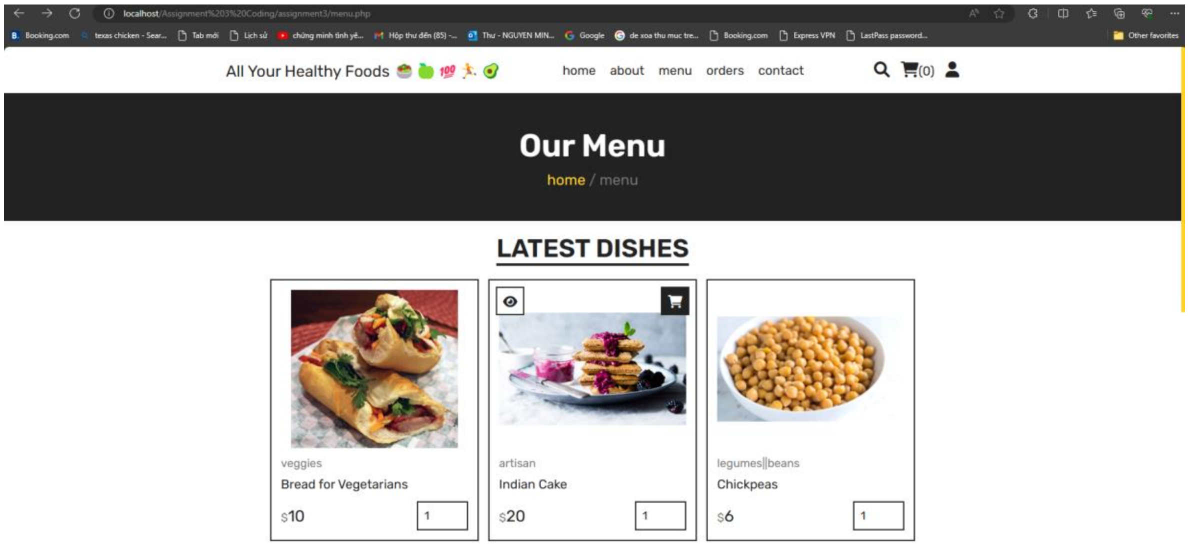
placed on : 2024-04-06
name : Tran Quoc Dung
email : dungbi3011@gmail.com
number : 0943910306
address : 02, Duong Khue, Cau giay, bo may, hanoi, vietname, áutraullia - 6666
payment method : credit card
your orders : banh mi (123 x 5) -
total price : $615/-
payment status : completed

Scenario 3: Start purchasing products.

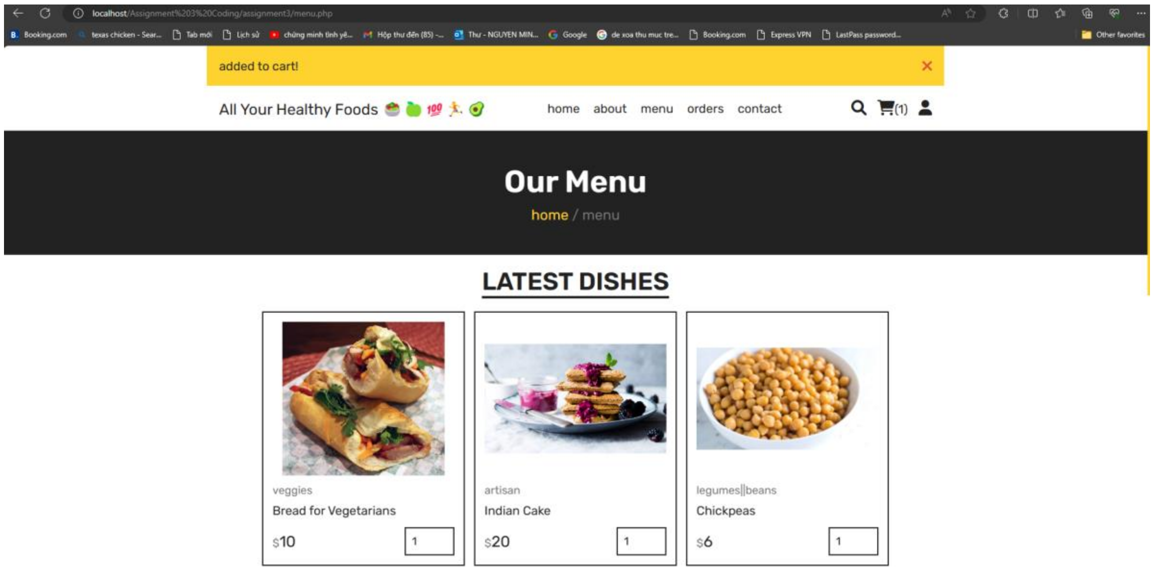
Directly accessing Product page (*menu.php*), or through Categories in Home page.



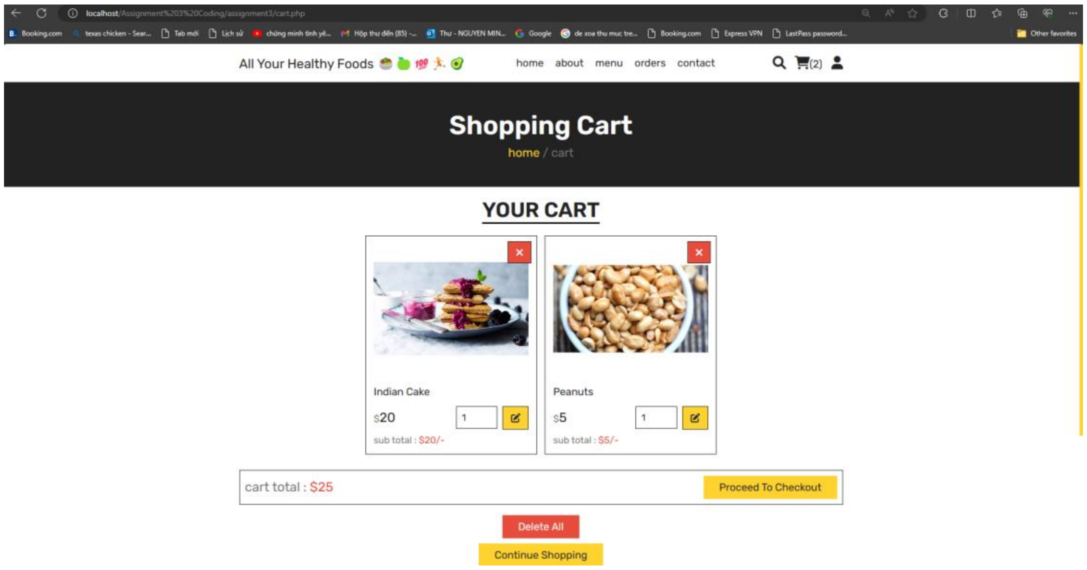
"Eye" logo to display more information about a product; "Cart" logo to add a product to Cart.



"Add to Cart" alert message & Product added to Cart.



Press “Continue Shopping” to add more products to Cart, and the price is updated.



localhost/Assignment%203%20Coding/assignment3/cart.php
All Your Healthy Foods
home about menu orders contact
(2)

Shopping Cart

home / cart

YOUR CART

Indian Cake

$20 | 1

sub total : $20/-

Peanuts

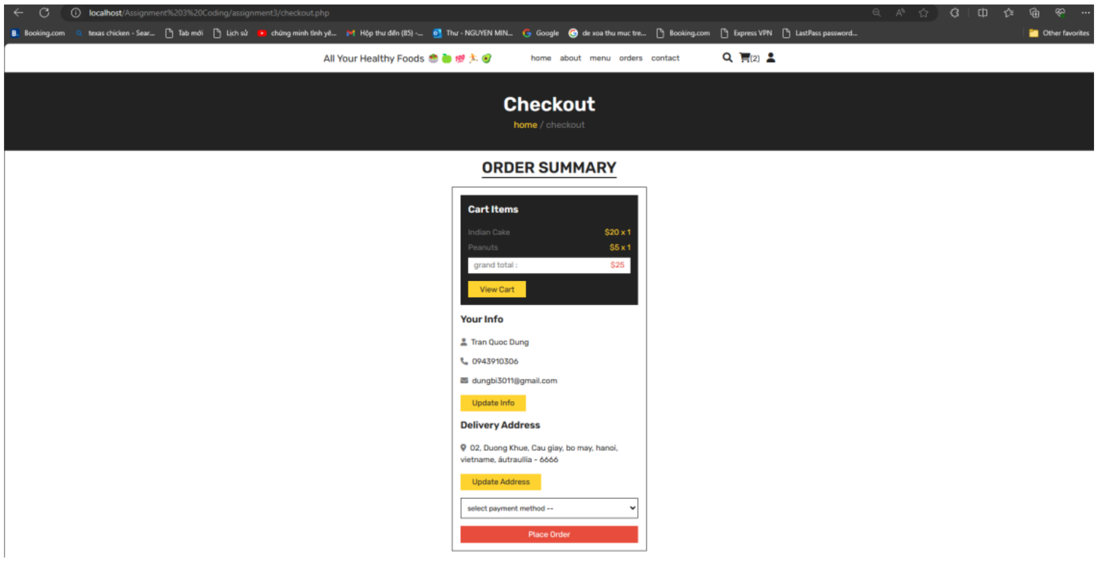
$5 | 1

sub total : $5/-

cart total : $25

Proceed To Checkout
Delete All
Continue Shopping

Webpages redirect to Checkout page (*checkout.php*) for confirming Order.



localhost/Assignment%203%20Coding/assignment3/checkout.php
All Your Healthy Foods
home about menu orders contact
(2)

Checkout

home / checkout

ORDER SUMMARY

Cart Items

Indian Cake $20 x 1

Peanuts $5 x 1

grand total : $25

Your Info

Tran Quoc Dung

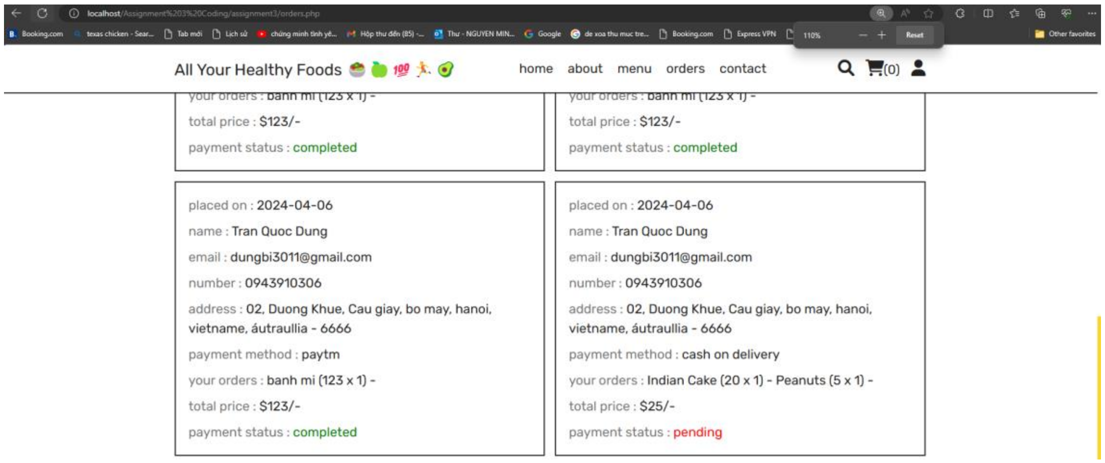
0943910306

dungbi3011@gmail.com

Delivery Address

02, Duong Khue, Cau giay, bo may, hanoi, vietname, áutraullia - 6666

Save Order in Order page (*order.php*), including Payment status.



localhost/Assignment%203%20Coding/assignment3/orders.php
110%
Reset
All Your Healthy Foods
home about menu orders contact
(0)

your orders : banh mi (123 x 1) -
total price : $123/-
payment status : completed

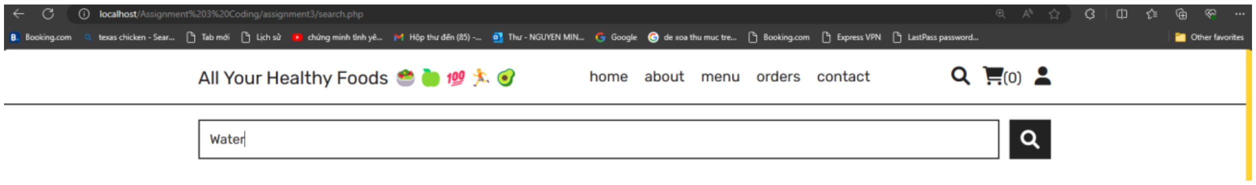
your orders : banh mi (123 x 1) -
total price : $123/-
payment status : completed

placed on : 2024-04-06
name : Tran Quoc Dung
email : dungbi3011@gmail.com
number : 0943910306
address : 02, Duong Khue, Cau giay, bo may, hanoi, vietname, áutraullia - 6666
payment method : paytm
your orders : banh mi (123 x 1) -
total price : $123/-
payment status : completed

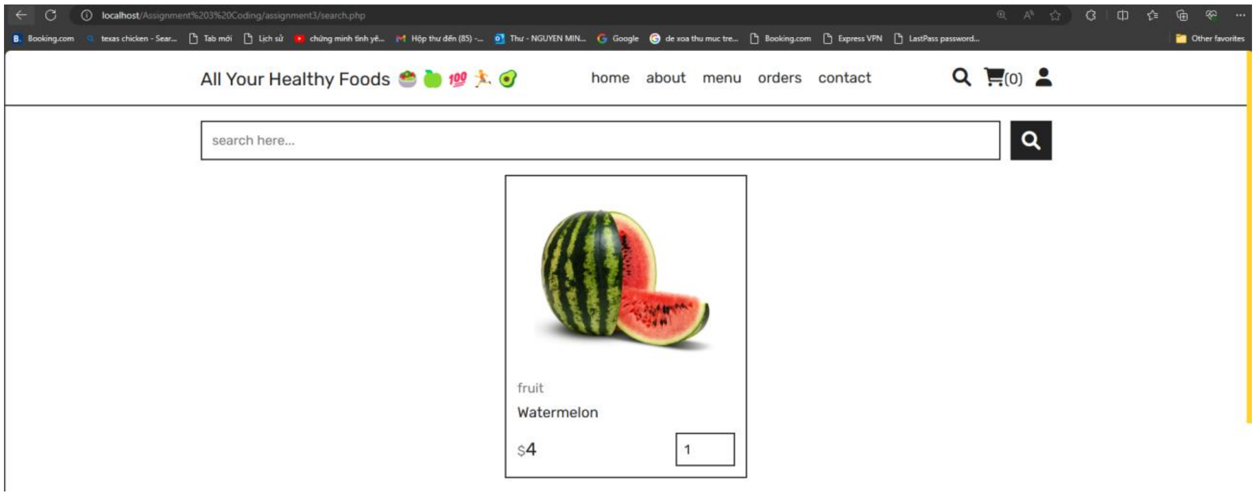
placed on : 2024-04-06
name : Tran Quoc Dung
email : dungbi3011@gmail.com
number : 0943910306
address : 02, Duong Khue, Cau giay, bo may, hanoi, vietname, áutraullia - 6666
payment method : cash on delivery
your orders : Indian Cake (20 x 1) - Peanuts (5 x 1) -
total price : $25/-
payment status : pending

Scenario 4: Searching for Products

Pressing "*Search*" logo to type in product name you want to search.


localhost/Assignment%203%20Coding/assignment3/search.php
All Your Healthy Foods
home about menu orders contact
(0)
Water

Seeing the results.

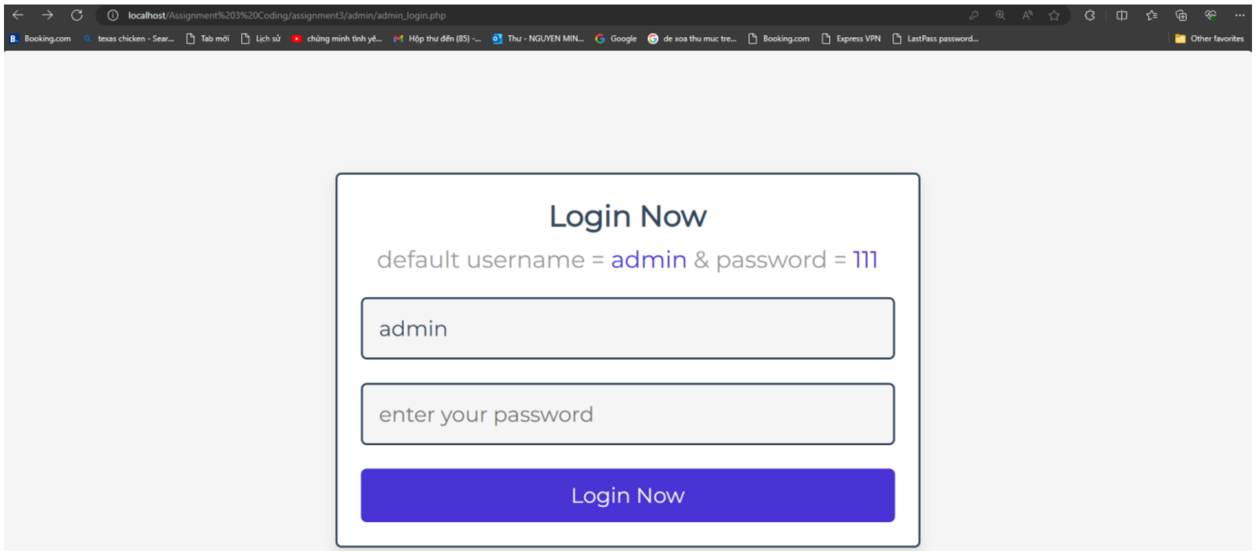

localhost/Assignment%203%20Coding/assignment3/search.php
All Your Healthy Foods
home about menu orders contact
(0)
search here...
fruit
Watermelon
$4
1

4.2.2.2. STAFFS, DEVELOPERS' VIEWPOINT

Currently, we have not worked on how to discriminate between Staff login and Customer login, since this aspect might be out of scope. However, some ideas can also be developed such as using unique code to be verified as Staffs, Developers; or privately authorizing some accounts. Therefore, the webpage only displays how the Staff accounts contribute to the online presence of All Your Healthy Foods.

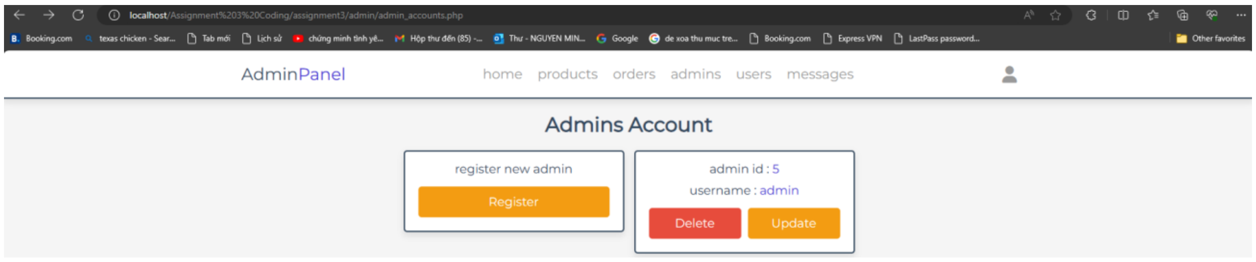
Scenario 1: Login as Staffs, Admin to observe Dashboard, Statistics, etc. (folder "*admin*")

Logging in as admin (location path: SWE30003_assignment3_coding/admin/admin_login.php)


localhost/Assignment%203%20Coding/assignment3/admin/admin_login.php
Login Now
default username = admin & password = 111
admin
enter your password
Login Now

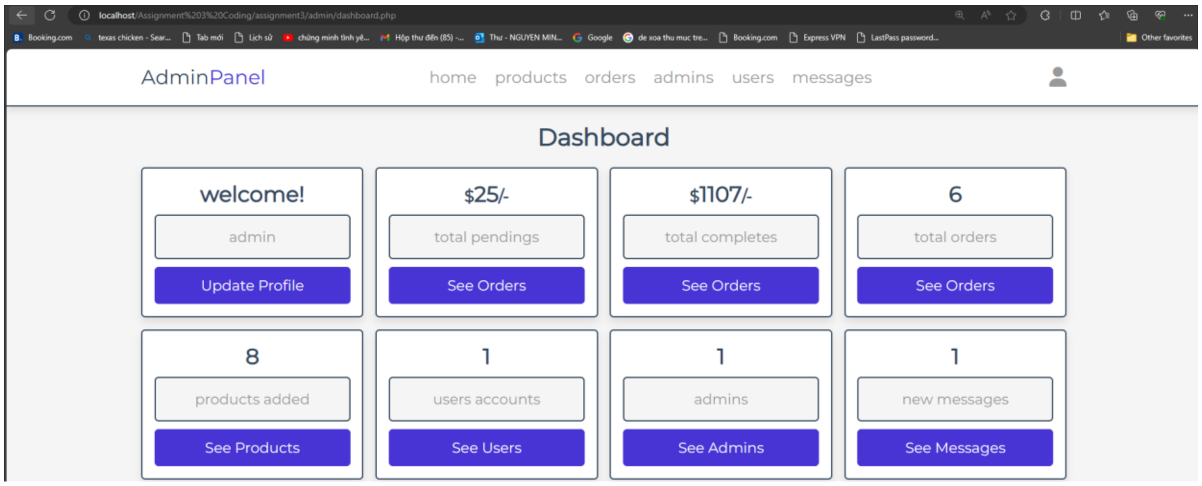
Admin can signup for new account, or update their information in Account page

(location path: SWE30003_assignment3_coding/admin/login_accounts.php)

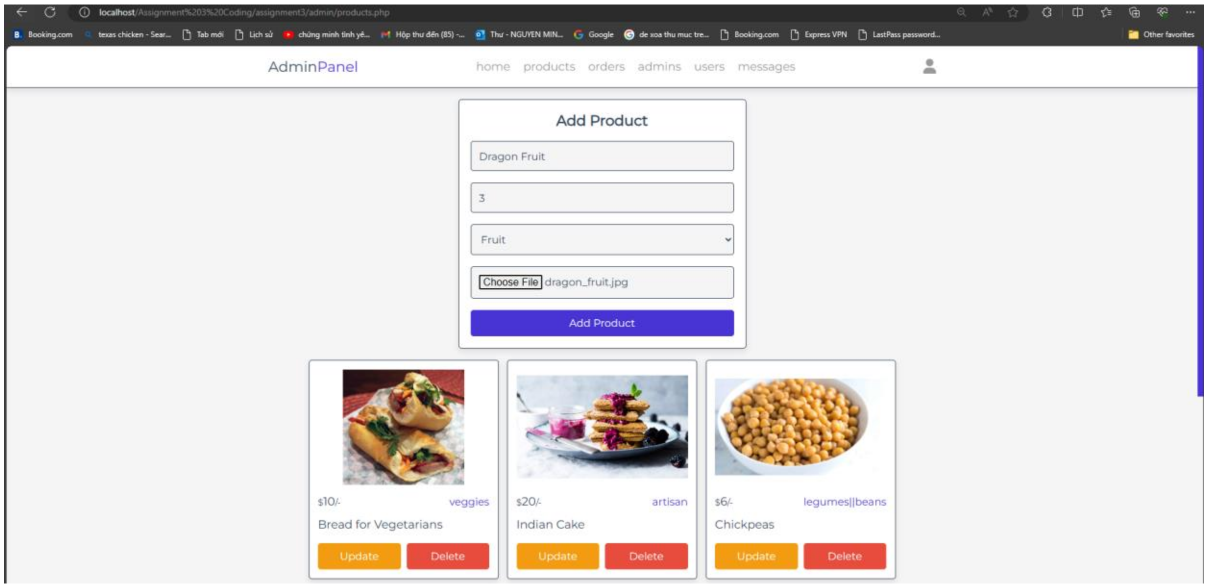


Scenario 2: Observing all the records in the Dashboard page.

Dashboard includes sold statistics, products, orders, customer accounts and feedback.



Scenario 3: Updating Products in Product page.



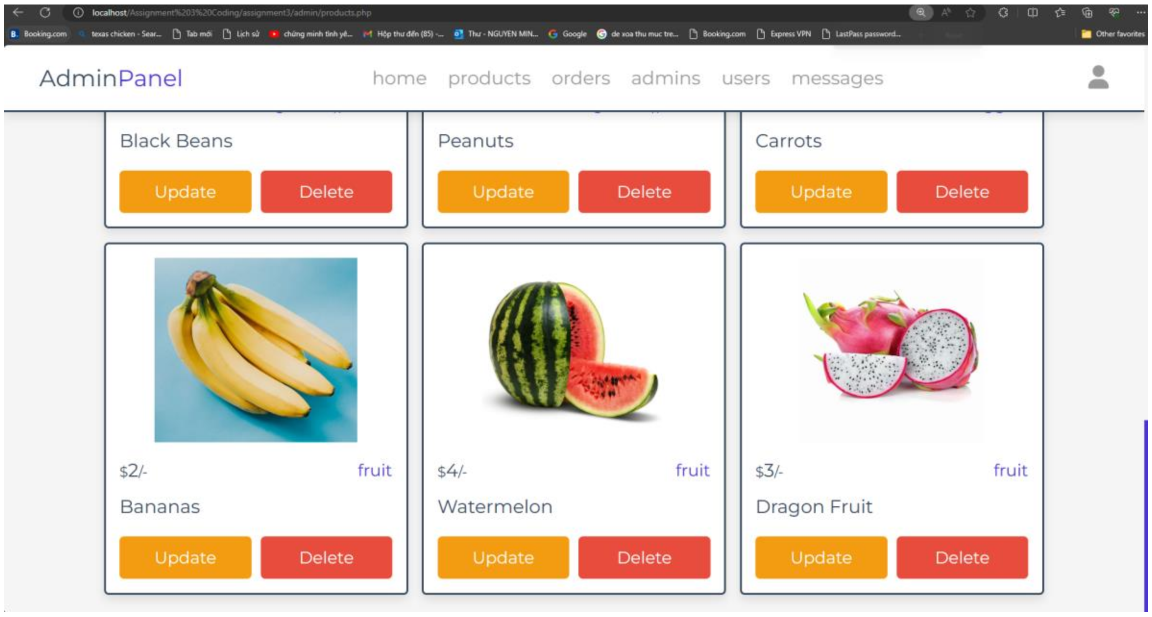
Dragon Fruit
3
Fruit
dragon_fruit.jpg

$10/- veggies	$20/- artisan	$6/- legumes||beans
Bread for Vegetarians	Indian Cake	Chickpeas

Manage and adding products.

NOTE: Adding, choosing images to folder named "uploaded_img".

```
<img src="../uploaded_img/<?= $fetch_products['image']; ?>" alt="">
```


localhost/Assignment%203%20Coding/assignment3/admin/products.php
AdminPanel
home
products
orders
admins
users
messages
Update
Delete

Black Beans

Peanuts

Carrots

$2/- fruit

Bananas

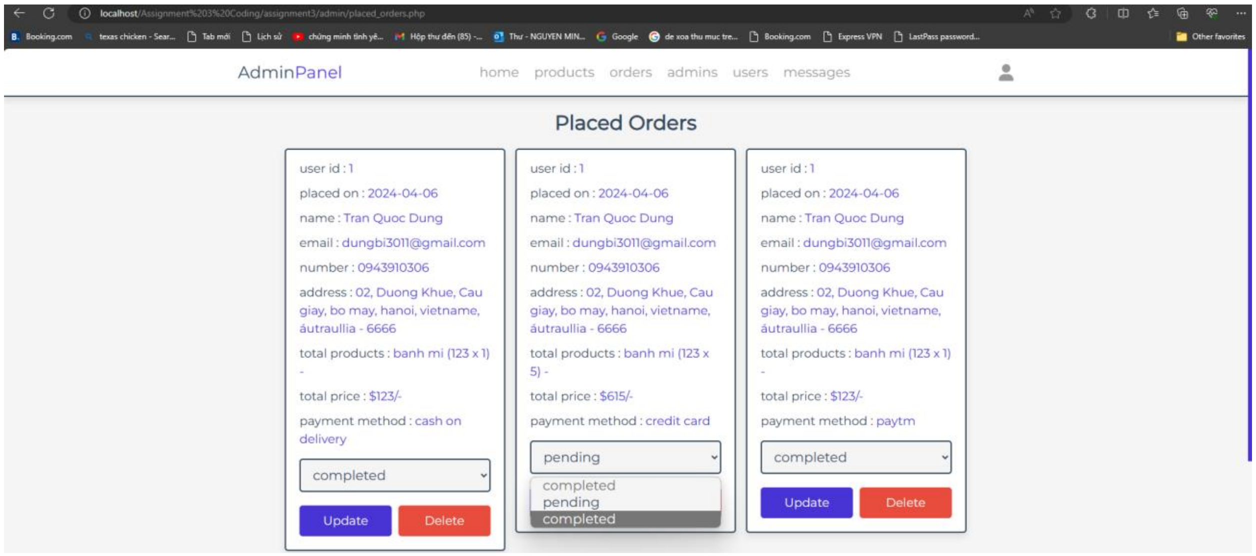
$4/- fruit

Watermelon

$3/- fruit

Dragon Fruit

Scenario 4: Observing every order & Updating order status.


localhost/Assignment%203%20Coding/assignment3/admin/placed_orders.php
AdminPanel
home
products
orders
admins
users
messages
Update
Delete

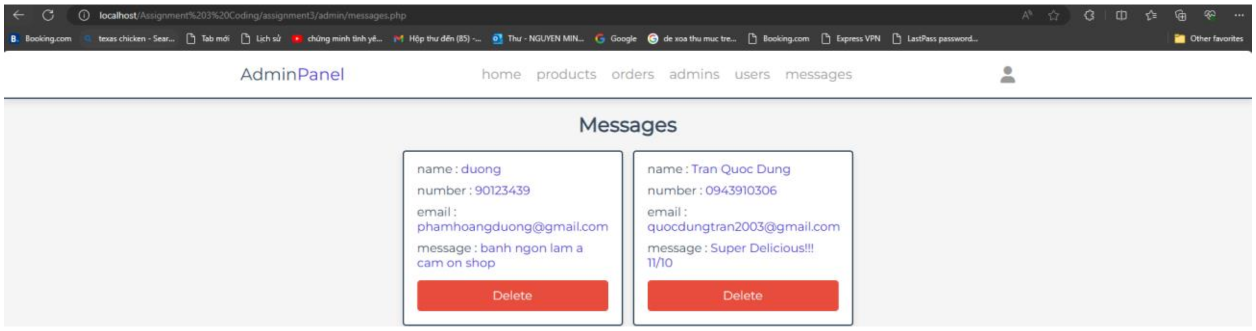
Placed Orders

user id : 1
placed on : 2024-04-06
name : Tran Quoc Dung
email : dungbi3011@gmail.com
number : 0943910306
address : 02, Duong Khue, Cau giay, bo may, hanoi, vietname, âutraullia - 6666
total products : banh mi (123 x 1) -
total price : $123/-
payment method : cash on delivery
completed

user id : 1
placed on : 2024-04-06
name : Tran Quoc Dung
email : dungbi3011@gmail.com
number : 0943910306
address : 02, Duong Khue, Cau giay, bo may, hanoi, vietname, âutraullia - 6666
total products : banh mi (123 x 5) -
total price : $615/-
payment method : credit card
pending
completed
pending
completed

user id : 1
placed on : 2024-04-06
name : Tran Quoc Dung
email : dungbi3011@gmail.com
number : 0943910306
address : 02, Duong Khue, Cau giay, bo may, hanoi, vietname, âutraullia - 6666
total products : banh mi (123 x 1) -
total price : $123/-
payment method : paytm
completed

Scenario 5: Observing Feedback from customers.


localhost/Assignment%203%20Coding/assignment3/admin/messages.php
AdminPanel
home
products
orders
admins
users
messages
Delete

Messages

name : duong
number : 90123439
email : phamhoangduong@gmail.com
message : banh ngon lam a cam on shop

name : Tran Quoc Dung
number : 0943910306
email : quocdungtran2003@gmail.com
message : Super Delicious!!! 11/10

5. Testing

5.1. Overview of Testing Strategy

To complete the Testing Procedures of the program, at least one testing methodology is required. Therefore, our team decided to opt for **Metamorphic Testing** as our Testing strategy.

Metamorphic Testing is a software-testing technique, used for creating and generating related test cases, and alleviating the test oracle problem by defining relations between inputs & outputs. The definition of test oracle is a mechanism or procedure, aims to compare the outputs with the test cases of the program, thereby evaluating the accuracy of the program. During implementing the **Metamorphic Testing**, **Metamorphic Relations** are the equally compulsory concepts required.

Metamorphic Relations are inseparable properties for testing the program implementation during these circumstances, which encompasses various related inputs and their outputs. When metamorphic relations are identified, related inputs can be newly created based on the this, thereby generating new test cases.

We can start implementing **Metamorphic Testing** by doing these steps:

- Executing several random test cases.

- Identify the properties of the problem, therefore identifying the metamorphic relations.

- Generating new test cases based on the original test case by utilizing the identified metamorphic relations.

- Verifying the metamorphic relations based on the computed outputs.

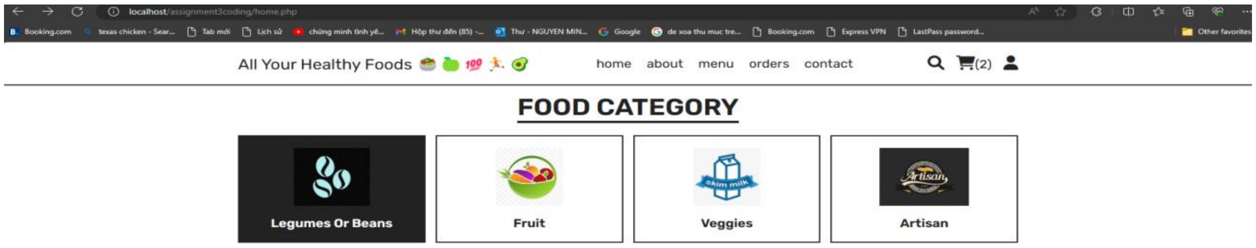
(References: *SWE30009 Testing Methodologies*)

5.2. Test Cases and Procedures

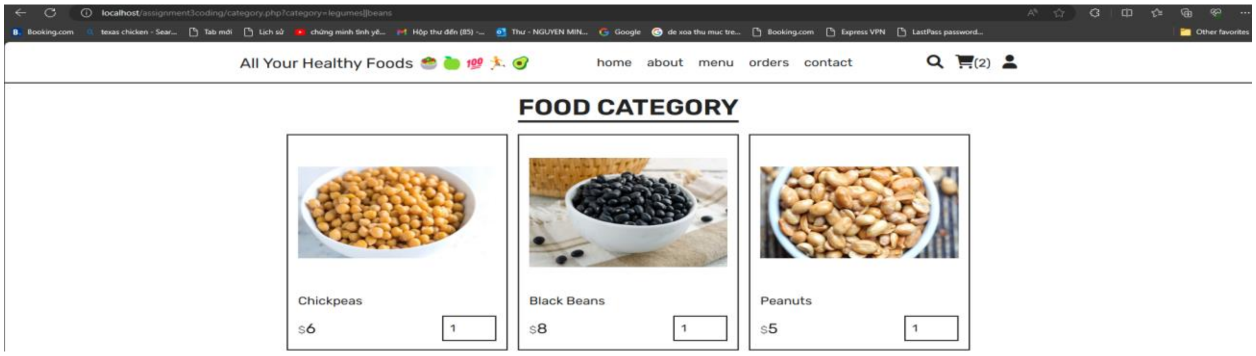
Test case 1: Filtering accurate types of healthy food.

Metamorphic Relation: Different categories lead to different display of Healthy Foods. Whereby, input 1.2 is selected based on input 1.1, to check for the accuration of categories filtering.

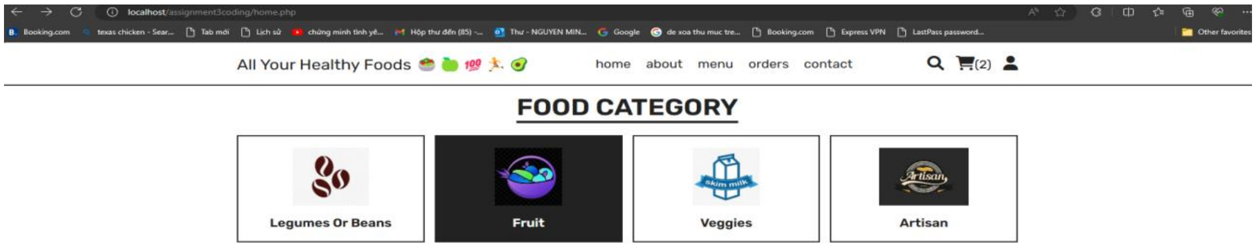
Input 1.1: “Legumes or Beans” Category.



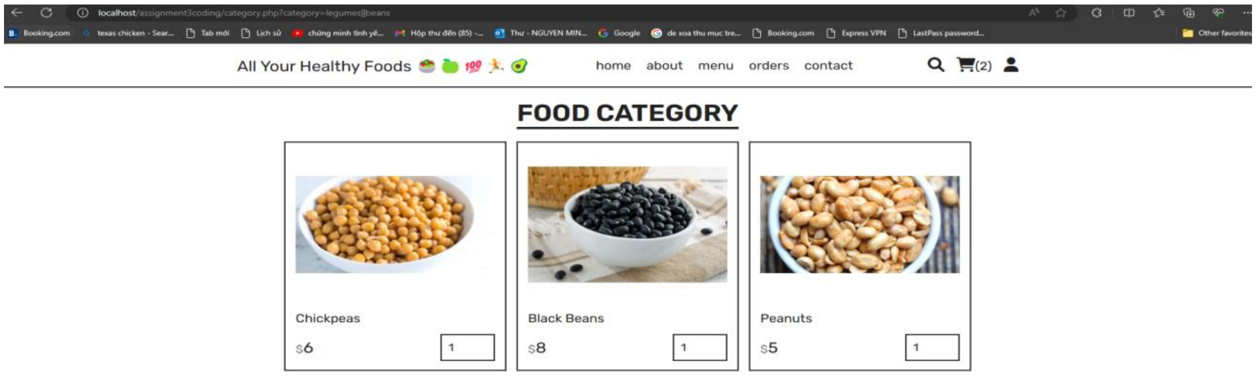
Output 1.1: Display only healthy foods with Category of “Legumes or Beans”.



Input 1.2: “Fruit” Category.



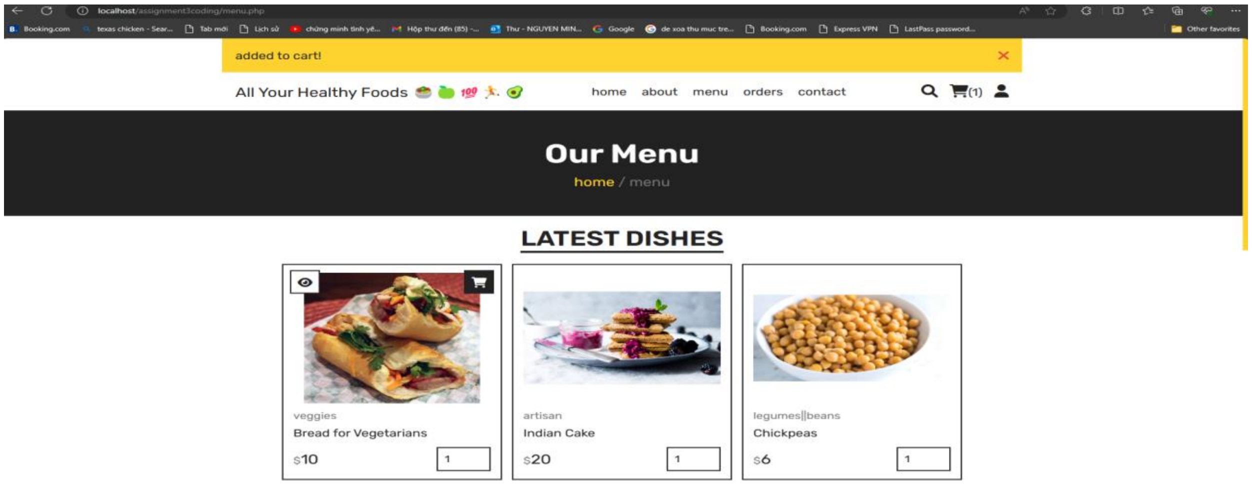
Output 1.2: Display only healthy foods with Category of “Fruit”.



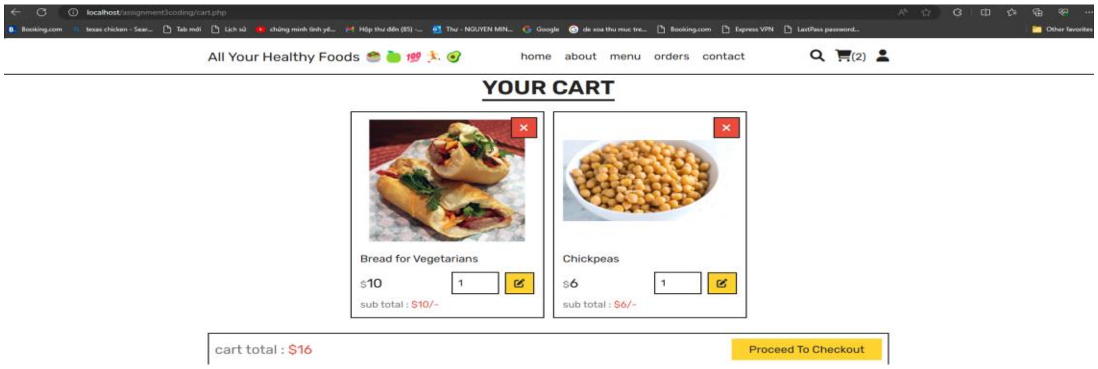
Test case 2: Cart Checking

Metamorphic Relation: Different or duplicated products add to cart will lead to the difference in the product name & quantity in the Cart. Whereby, input 2.2 is generated from input 2.1 by selecting duplicated Watermelon.

Input 2.1: “Bread for Vegetarians” and “Chickpeas” are added to Cart.



Output 2.1: Cart displayed respective products with accurate quantity & price.



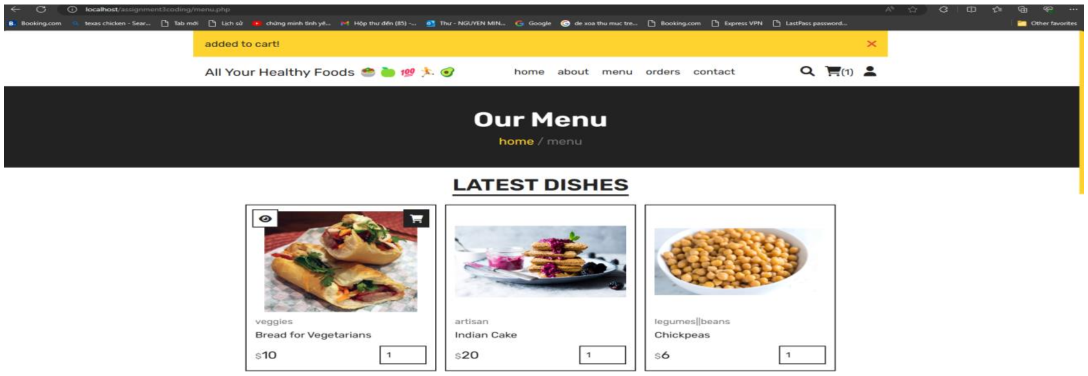
YOUR CART

Bread for Vegetarians
$10 1
sub total : $10/-

Chickpeas
$6 1
sub total : $6/-

cart total : $16

Input 2.2: “Bread for Vegetarians” and “Chickpeas” are added to Cart, while “Bread for Vegetarians” is added to Cart twice manually.



added to cart!

Our Menu

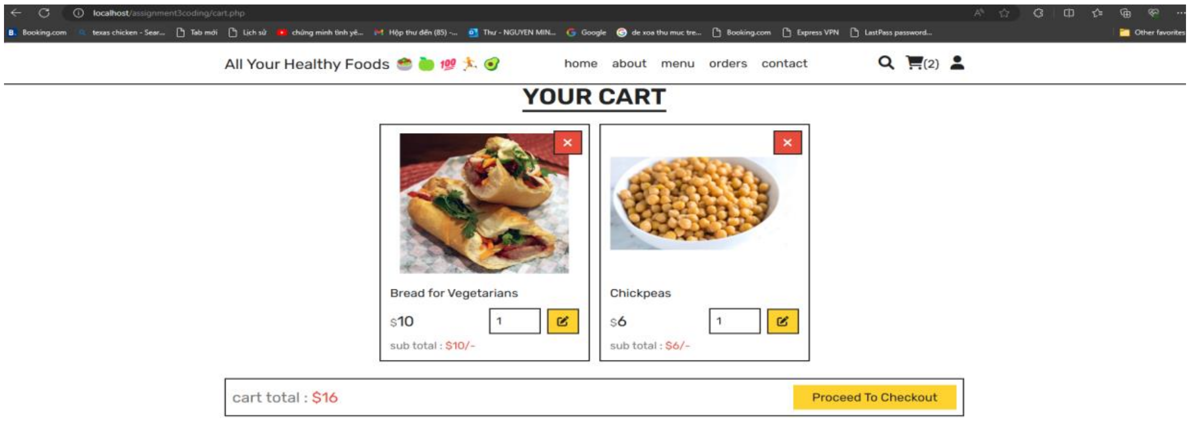
LATEST DISHES

veggies
Bread for Vegetarians
$10 1

artisan
Indian Cake
$20 1

legumes||beans
Chickpeas
$6 1

Output 2.2: Cart displayed respective products with accurate quantity & price.



YOUR CART

Bread for Vegetarians
$10 1
sub total : $10/-

Chickpeas
$6 1
sub total : $6/-

cart total : $16

The quantity and price should increase since “Bread for Vegetarians” is added to Cart twice, while the output illustrates the remained unchanged quantity and price.

5.3. Validation of Correct Functionality (PASS/FAIL)

Test case 1: Filtering accurate types of healthy food:

PASS

As the result of the two different inputs display the accurate respective products based on the category selected, this function works accurately.

Test case 2: Cart Checking

FAIL

Although the function of Cart implements correctly at displaying selected products to be added and calculating total price, when the users added the same product to the Cart twice, the quantity should increase to 2, and the price should be increased by adding the price of “Bread for Vegetarians”.

However, the price and quantity remained unchanged, therefore this function does not work accurately, and failed the requirements.

Conclusion

The Online Healthy Foods Store for All Your Healthy Foods has been designed and implemented using object-oriented principles and best practices. The design phase involved creating a high-level overview, followed by a detailed design that included UI mockups and an outline of the database schema. The code is implemented in PHP and utilizes a MySQL database for data storage. The website provides functionalities for users to browse products, add them to a cart, checkout, and manage their order history. Additionally, an admin panel allows for product management, user account management, and order tracking.

References

- XAMPP Download: https://www.apachefriends.org/download.html
- SWE30009 Testing Methodologies:

Project Report.pdf

Appendix

Assignment 2.pdf

Full PDF references in .zip folder