Swinburne University of Technology

School of Software and Electrical Engineering

ASSIGNMENT AND PROJECT COVER SHEET

Subject Code: SWE30003 Unit Title: Software Architectures and Design

Assignment number and title: 3, Design Implementation Due date: 28th May 2023

Tutorial Day and time: Tuesday 9:00 AM Project Group: 8

Tutor: 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
103846676	Tran Viet Thang	Thang
103806557	Phan Thanh Bach	Bach
103487185	Dinh Duc Minh	Minh
103806557	Doan Minh Hieu	Hieu

Marker's comments:

Total Mark:__________________

Extension certification:

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

Signature of Convener:______________________

Contents

I. Introduction

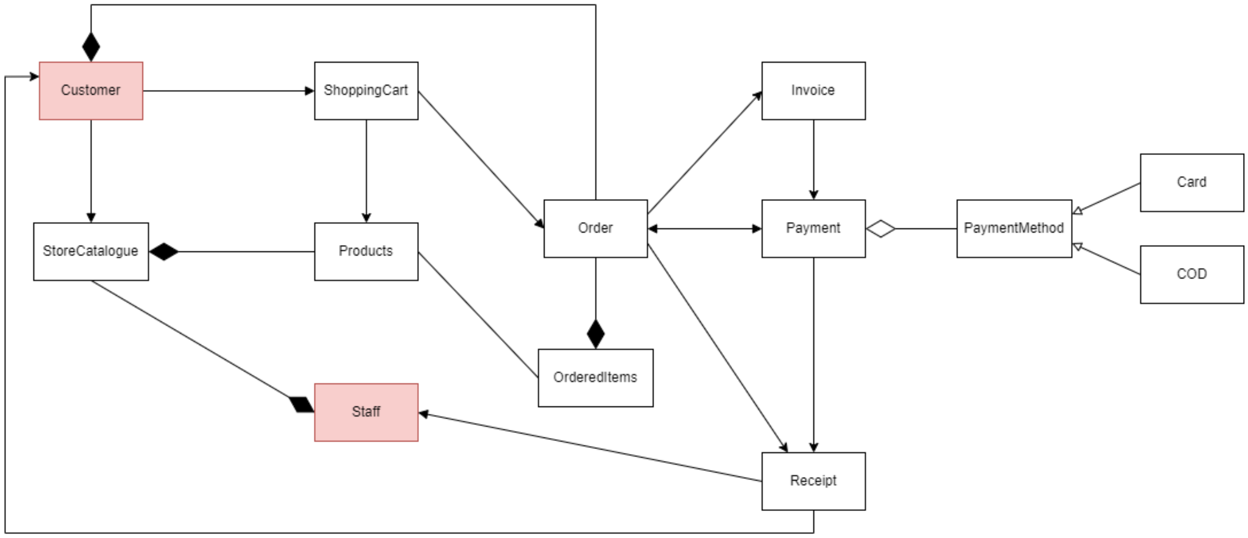
The Online Healthy Foods Store project for All Your Healthy Foods aims to extend the reach of the local food store into the digital realm. Our design focuses on creating a user-friendly platform where customers can conveniently browse, order, and receive healthy products. This initiative addresses the limitations posed by the store's reliance solely on physical operations and seeks to capitalize on the growing trend of online shopping. By leveraging object-oriented design principles, our solution aims to provide a seamless and intuitive shopping experience for customers while streamlining operations for store administrators.

II. Summary of Design Revision

1. Overall changes/non-changes at class level

During the refinement process of the Online Healthy Foods Store System, several adjustments were made to the overall structure of the implementation due to time limitations. These modifications were carefully considered to be removed but not affect the overall structure of the system:

- Not implement classes like Statistics, BestSellingProducts, Goods_sold_over_various_periods and Delivery.
- Change naming convention: Owner -> Staff, OrderLineItems -> OrderedItems.
- Change payment method: Account -> COD.



2. Changes/no-changes to Responsibilities and Collaborators

There are changes were made on CRC Cards based on the feedbacks: only need immediate collaborators:

Customer:

Class Name: Customer Super Class: Person	
Customer is responsible for knowing information about a single real customer.	
Responsibilities	Collaborators
Know the Customer's login information: email, password	NA
Know customer information: name, address and phone number	NA
Know customer order	Order
Know customer delivery status	Delivery

Receipt:

Class Name: Receipt Super Class: -	
A receipt serves as proof of purchase for the customer.	
Responsibilities	Collaborators
Know order ID	Order
Know customer information (name, address, phone number)	Customer
Know order details (products, quantities)	OrderLineItems
Know payment details (amount, method)	Payment
Know payment status (Success or Declined)	Payment
Can email receipt for Customer	Customer

Order:

Class Name: Order Super Class: -	
Essential to managing sales transactions in all aspects of consumer purchases, from the point of sale to the last phases of order monitoring and payment processing.	
Responsibilities	Collaborators
Can automatically generate an Order ID	NA
Knows customer information	Customer
Know customer-ordered items include: Products, Quantity	ShoppingCart

Know Payment status	Payment
Can create an invoice	Invoice
Can create a receipt	Receipt
Can create delivery	Delivery

OrderLineItems:

Class Name: OrderLineItems Super Class: -	
OrderLineItems manages the products included in an order, associating quantities and calculating subtotals.	
Responsibilities	Collaborators
Know customer's order	Order
Know individual products included in an order	Products
Know the quantity of each product	Products
Can calculate subtotal for each product	Products

COD:

Class Name: COD Super Class: PaymentMethod	
User want to pays when package delivered	
Responsibilities	Collaborators
Know amount user need to pay	Order

Other CRC cards such as Statistics, BestSellingProducts (Statistics), Goods_sold_over_various_periods (Statistics), Account (PaymentMethod) and Delivery are not used from this Assignment due to the new implementation design of Assignment 3. The rest of the CRC cards remain the same.

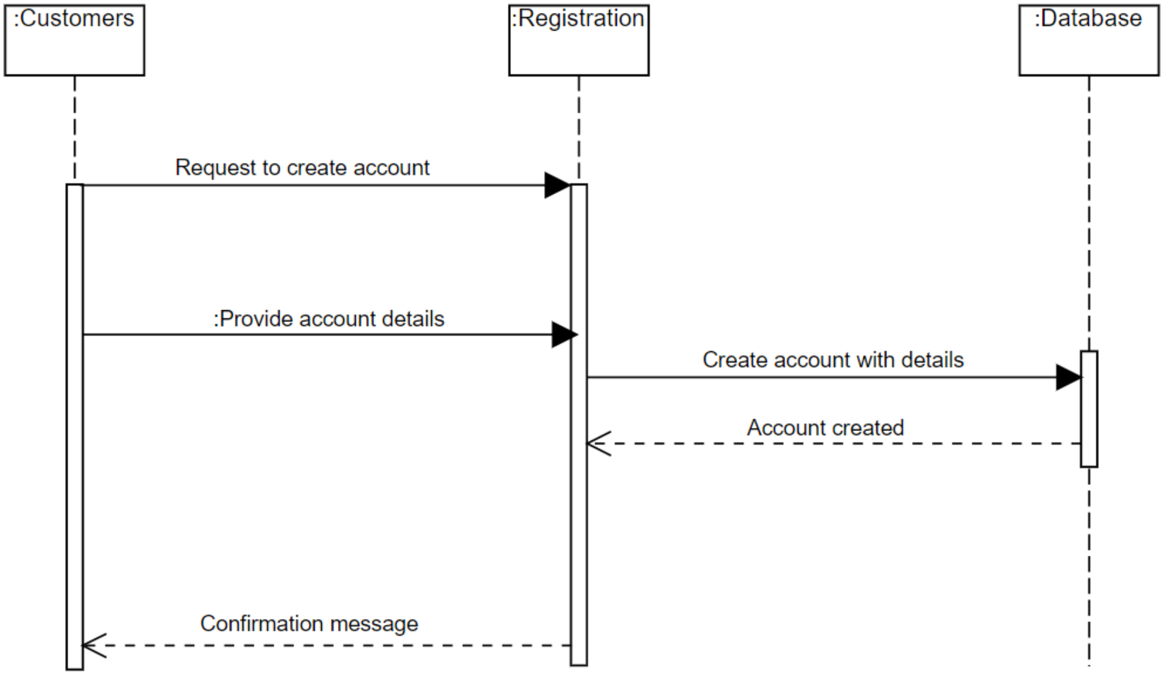
3. Changes/non-changes to dynamic aspects

Based on teacher feedback on interaction scenarios, we have made changes to improve the implementation design:

Customer create an Account:

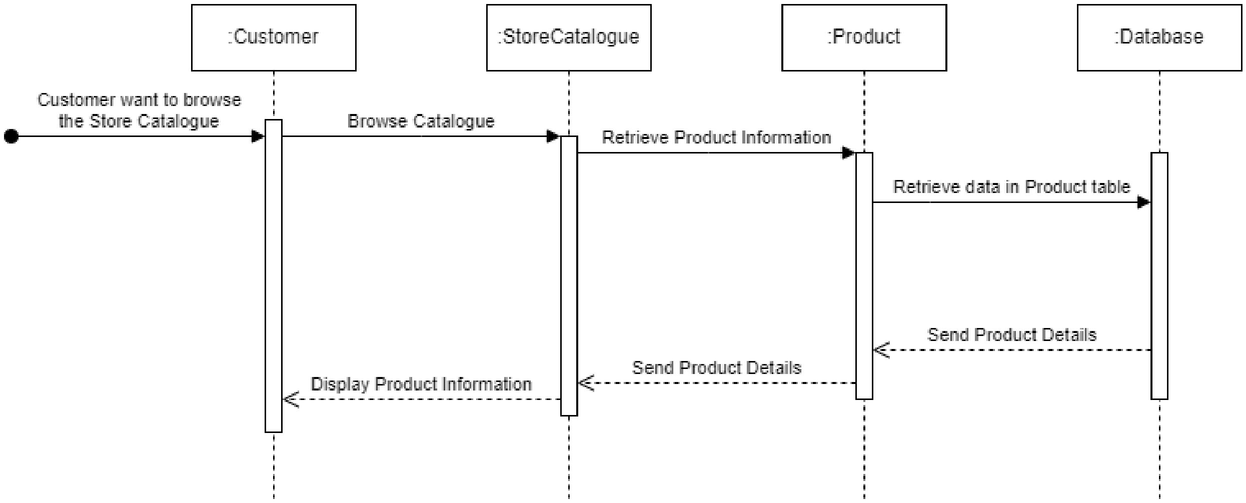
Feedback: Application is not correct, also need to use Objects

To ensure the sequence follows object-oriented programming principles, I have adjusted the class name from Application to Registration.



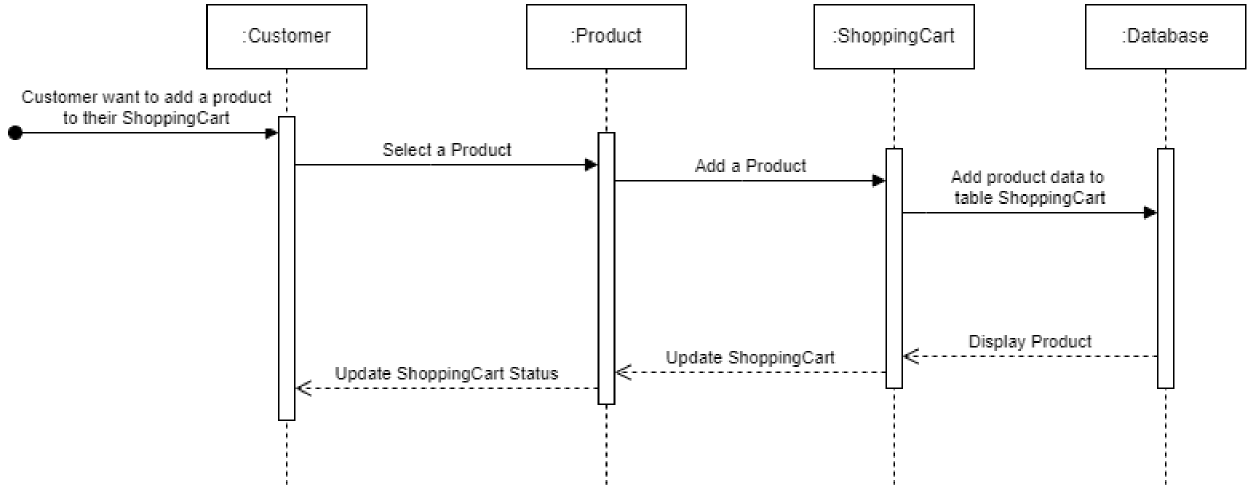
Customer browse the Store Catalogue:

Feedback: need to interact with Repository/Database to retrieve and create Products


:Customer
:StoreCatalogue
:Product
:Database
Customer want to browse the Store Catalogue
Browse Catalogue
Retrieve Product Information
Retrieve data in Product table
Send Product Details
Send Product Details
Display Product Information

Customer manage the Shopping Cart:

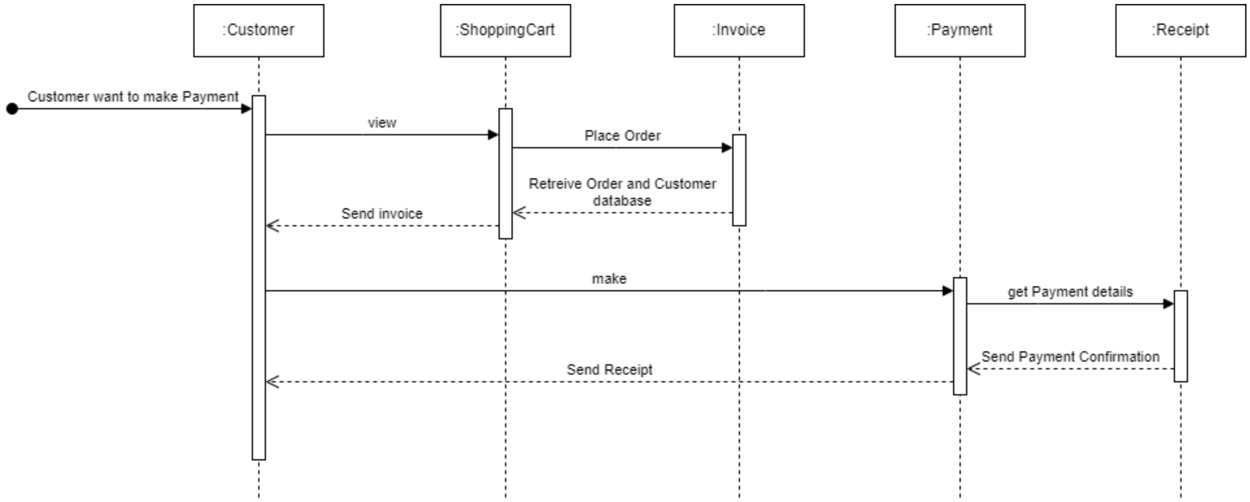
Feedback: “manage cart” message is too general


:Customer
:Product
:ShoppingCart
:Database
Customer want to add a product to their ShoppingCart
Select a Product
Add a Product
Add product data to table ShoppingCart
Display Product
Update ShoppingCart
Update ShoppingCart Status

System generates Invoice and Receipt:

Feedback: Order should not create Receipt, Invoice

I have modified the sequence so that Invoice, Payment, and Receipt interact with each other to create a smooth flow.



The Manage goods packaging and shipment and Provide basic statistics on goods sold interaction scenarios are not used due to the new implementation design of Assignment 3. The rest of the interaction scenarios remain the same.

Additionally, the Delivery Management Process and Sales Statistics Monitoring Process bootstraps are not used due to the new implementation design of Assignment 3. The rest of the bootstraps remain the same.

III. Design Quality

Reflecting on the quality of the initial design from Assignment 2, it's evident that while the design addressed several fundamental aspects of the Online Healthy Foods Store System, there were notable areas for improvement. The initial design adequately addressed the core functionalities of the Online Healthy Foods Store System such as customer browsing Store catalogue, shopping cart management, order processing, payment handling and owner management of the store catalogue. Class responsibilities and collaborations were outlined, providing a clear understanding of the system's architecture.

While the basic structure of the system was outlined in Assignment 2, there was a clear absence of detailed descriptions and implementations for classes related to sales statistics and delivery management. However, it's important to note that these missing components do not significantly impact the overall structure of the project. Unfortunately, due to time limitations, these functionalities could not be implemented.

The initial design facing several errors and deficiencies that required attention. In the UML class diagram, inaccuracies were noted concerning inheritance, particularly with classes like Order and OrderLineItems. Moreover, the relationships between classes such as Invoice, Payment, and Receipt were inaccurately depicted. The CRC cards lacked precision, including collaborators that were not immediate to the class's responsibilities. Additionally, the UML sequence diagrams failed to effectively demonstrate object-oriented programming principles in action, lacking simplicity and clarity. These issues highlighted the need for refinement and improvement in the design to ensure a more accurate implementation of the system.

To address these omissions, errors, and ambiguities, several changes were documented in part II. Summary of Design Revision. This involved refining the UML class diagram to accurately represent inheritance and relationships between classes. Additionally, CRC cards were revised to include only immediate collaborators, providing a clearer understanding of class responsibilities. Furthermore, the UML sequence diagrams were simplified and enhanced to better illustrate object-oriented principles and the flow of interactions within the system. These adjustments were made in an effort to address the shortcomings in the original design and provide a more thorough and cohesive depiction of the software system.

Overall, the changes made to the initial design aimed to enhance its completeness, clarity, and alignment with object-oriented programming principles. By addressing the omissions, errors, and ambiguities identified in the initial design, the revised design sought to create a more comprehensive and robust framework for the Online Healthy Foods Store System, capable of meeting the evolving needs of the business and its users.

Reflecting on our detailed design and implementation journey, we've learned invaluable lessons from our use of different design tools:

- Early Validation and Refinement: We relied on tools like UML diagrams, design objects, and CRC cards to shape our system. Next time, we'll validate our design early with prototypes and tests, tweaking it iteratively. This ensures we catch issues sooner and make our design stronger.

- Better Requirements Gathering: We used tools to capture stakeholder needs effectively. In the future, we'll use UML diagrams and CRC cards more thoroughly to understand user needs upfront. This helps ensure our design matches what users want and what the business needs.
- Flexibility Matters: Design patterns proved instrumental in making our system adaptable to change. Moving forward, we recognize the importance of adhering to these patterns to maintain flexibility in our design. By consistently applying proven design patterns, we can ensure that our system remains agile and responsive to evolving requirements. This proactive stance enables us to easily accommodate new features or modifications, keeping our solution relevant and resilient in dynamic environments.
- Team Collaboration is Key: Our collaborative approach, facilitated by design tools like UML diagrams and CRC cards, significantly enhanced our design process. However, we aim to further emphasize team collaboration in future endeavors. By fostering open communication and knowledge sharing among team members, we can capitalize on diverse perspectives and expertise, enriching the design process and yielding stronger solutions. Leveraging these collaborative practices ensures that everyone is aligned with the design vision, fostering a sense of ownership and collective responsibility for the success of the project.

IV. Implementation and Testing

1. Mapping design to code

a. Index (Home Page):

- Displays a welcome message and introduction to the online healthy food store.
- Includes a banner, main section with information about the store, and a footer.
- Likely the entry point of your website.

b. Account:

- Handles user registration and login functionalities.
- Includes forms for signing up and signing in, along with validation logic in PHP.
- Displays messages based on registration or login status.

c. Store Catalogue:

- Displays a list of products available in the store.
- Includes filtering options by category and a search bar for finding specific products.
- Uses JavaScript for dynamic filtering and searching.

d. Product Detail:

- Displays detailed information about a specific product when clicked from the store catalogue.
- Shows product image, name, price, description, and other details.
- Retrieves product information from a database using PHP.

e. Shopping Cart:

- Shows the items added to the shopping cart.
- Includes functionality to add or remove items and adjust quantities.
- Displays the total price and provides a checkout button.

f. Payment (payment.php):

- This section handles the payment process for orders.
- Users can choose between Cash-on-Delivery (COD) or VISA payment methods.
- If VISA is chosen, additional fields for VISA number, CVC, and due date are displayed.
- Upon submission, the payment details are processed and stored in the database.
- A summary of all the products that the user added to their shopping cart.
- The quantity and price of each product.
- The total price of the order.
- The invoice serves as a confirmation of the user's purchase and provides a detailed breakdown of their order.

g. Receipt (invoice.php):

- This part generates a receipt displaying the user's transaction history.
- It retrieves payment details from the database for the current user and displays them in a table format.

h. Order History (order_history.php):

- This section shows the user's transaction history similar to the receipt but in a different format.
- It fetches payment details from the database and displays them in a tabular format.

i. User Profile (user_profile.php):

- This part displays the user's profile information such as name, email, and date joined.
- It fetches user details from the database and presents them in a structured format.

ii. Stock Adding (stock_adding.php):

- This part of the code allows the user to manage stock levels for products.
- It displays a table with product information including name, category, current stock level, and last import date.
- For each product, there's a form to add additional stock with a quantity input field and an "Add" button.
- Upon submitting the form, the restock.php script is invoked to update the stock levels in the database.

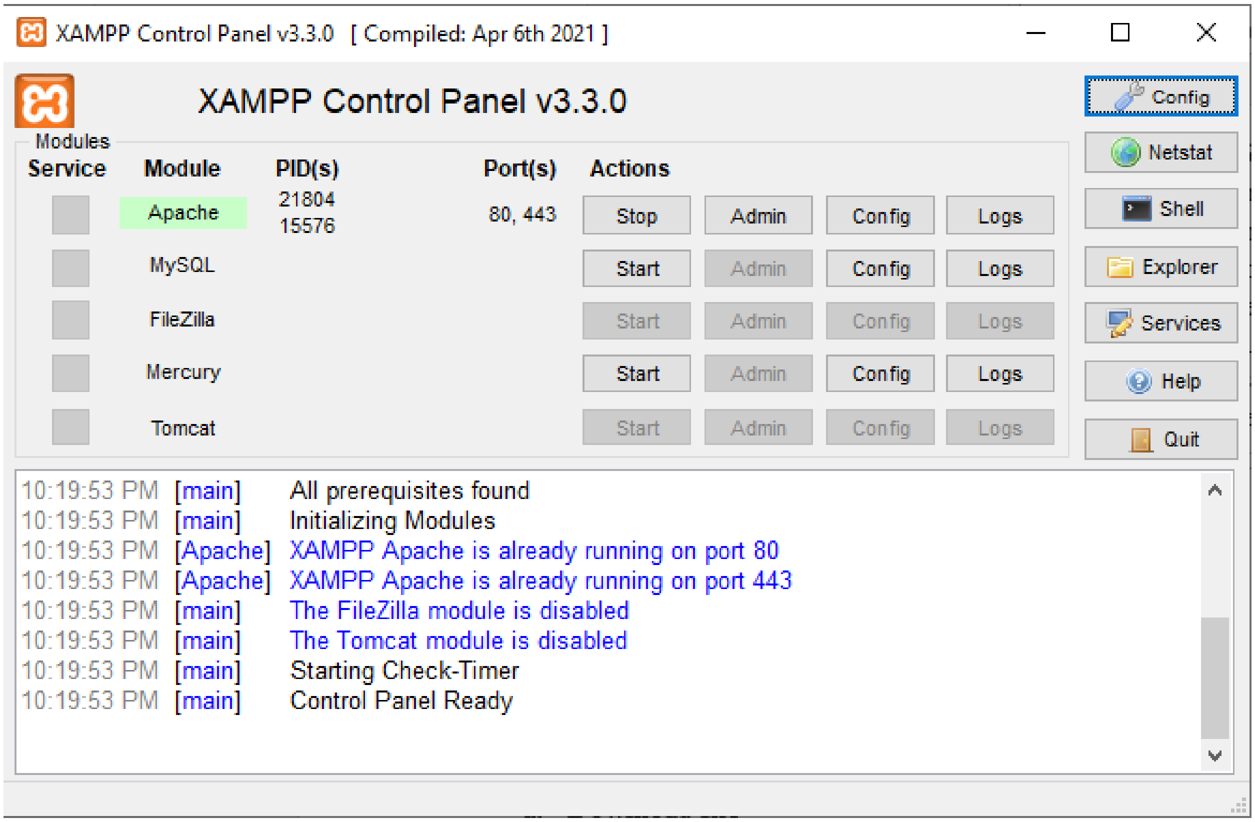
2. Compilation and Execution

a. Download file

b. Extract Files

c. Move the file to ./XAMPP/htdocs/

d. Open XAMPP Control Panel then Start Apache

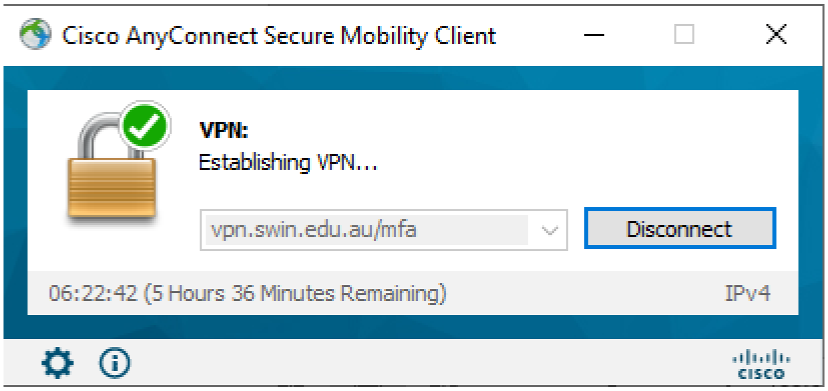


XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]
XAMPP Control Panel v3.3.0
Modules
Config
Netstat
Shell
Explorer
Services
Help
Quit

Service	Module	PID(s)	Port(s)	Actions			
□	Apache	21804 15576	80, 443	Stop	Admin	Config	Logs
□	MySQL			Start	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

```
10:19:53 PM  [main]     All prerequisites found
10:19:53 PM  [main]     Initializing Modules
10:19:53 PM  [Apache]   XAMPP Apache is already running on port 80
10:19:53 PM  [Apache]   XAMPP Apache is already running on port 443
10:19:53 PM  [main]     The FileZilla module is disabled
10:19:53 PM  [main]     The Tomcat module is disabled
10:19:53 PM  [main]     Starting Check-Timer
10:19:53 PM  [main]     Control Panel Ready
```

e. Turn on Cisco AnyConnect Secure Mobility Client to connect to the database



Cisco AnyConnect Secure Mobility Client

VPN:
Establishing VPN...

vpn.swin.edu.au/mfa

06:22:42 (5 Hours 36 Minutes Remaining)

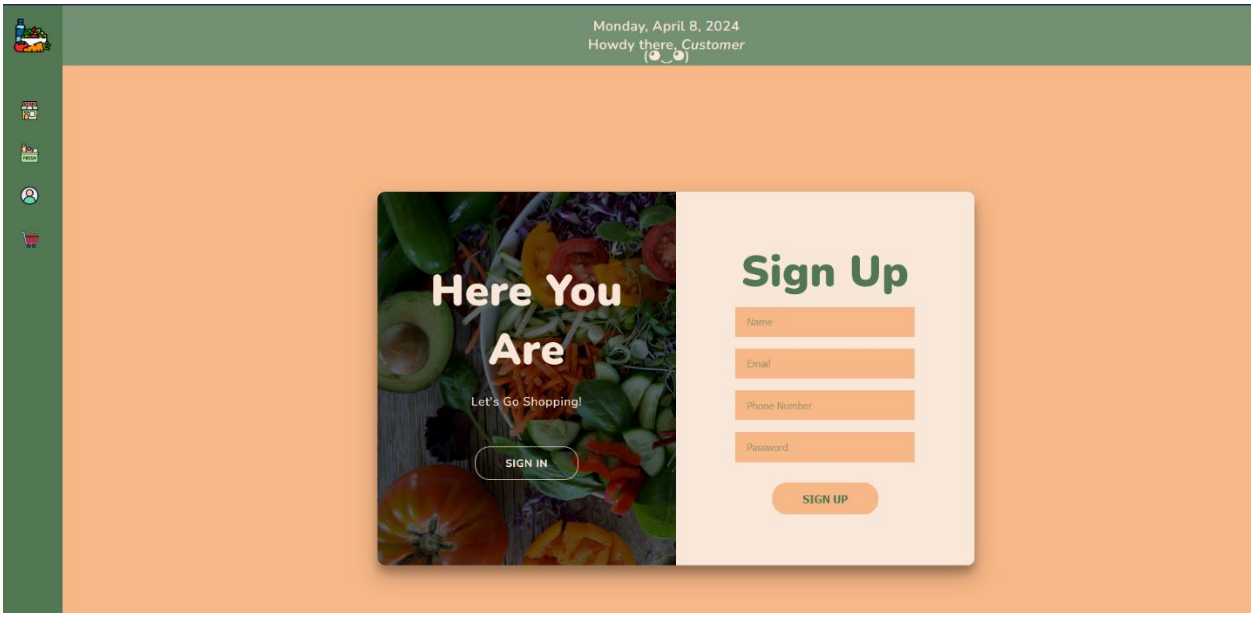
IPv4

d. Open web browser then navigate to

3. Testing

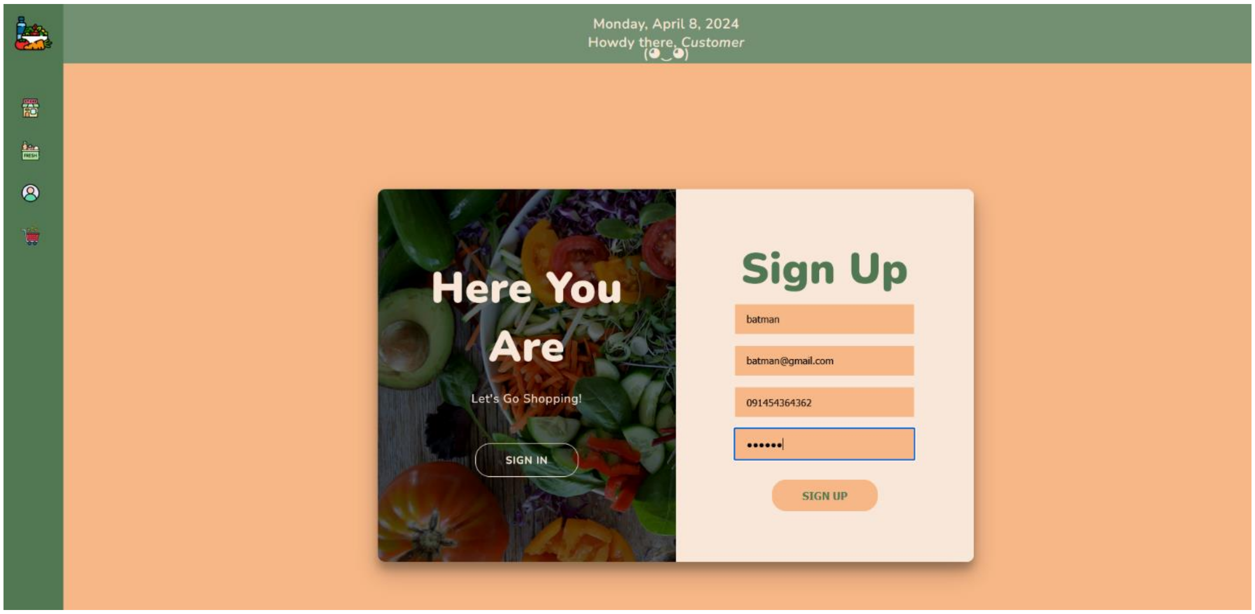
Create customer accounts

Step 1: User click on Sign Up



Monday, April 8, 2024
Howdy there, Customer

Step 2: Fill in information



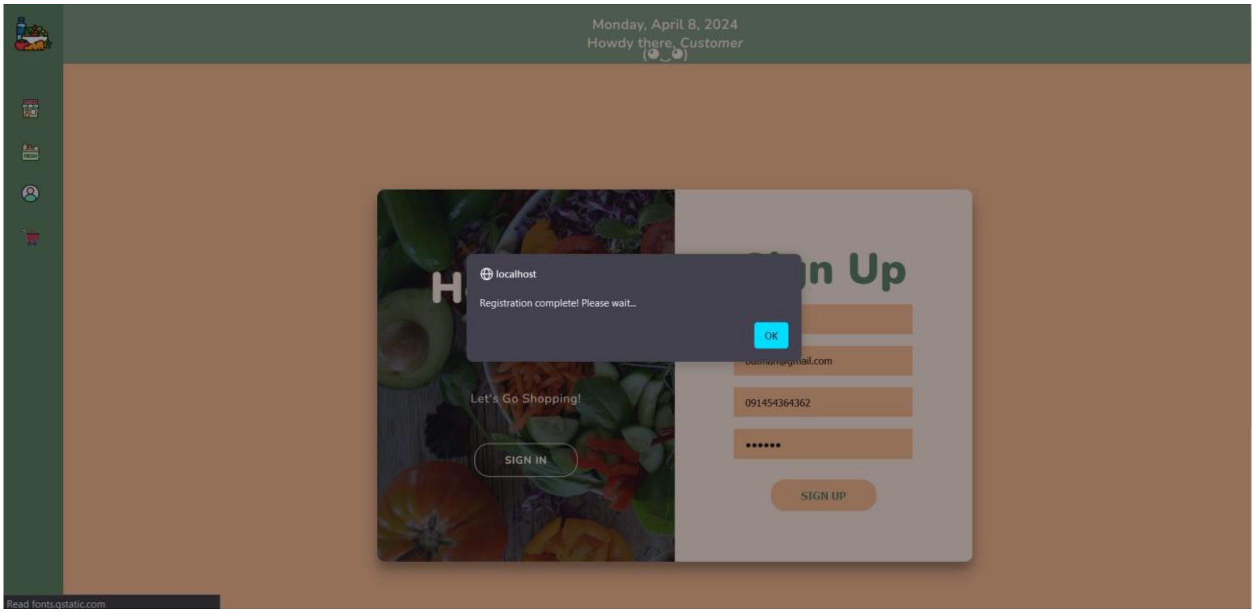
batman

batman@gmail.com

091454364362

••••••

Step 3: Display successfully message



localhost

Registration complete! Please wait...

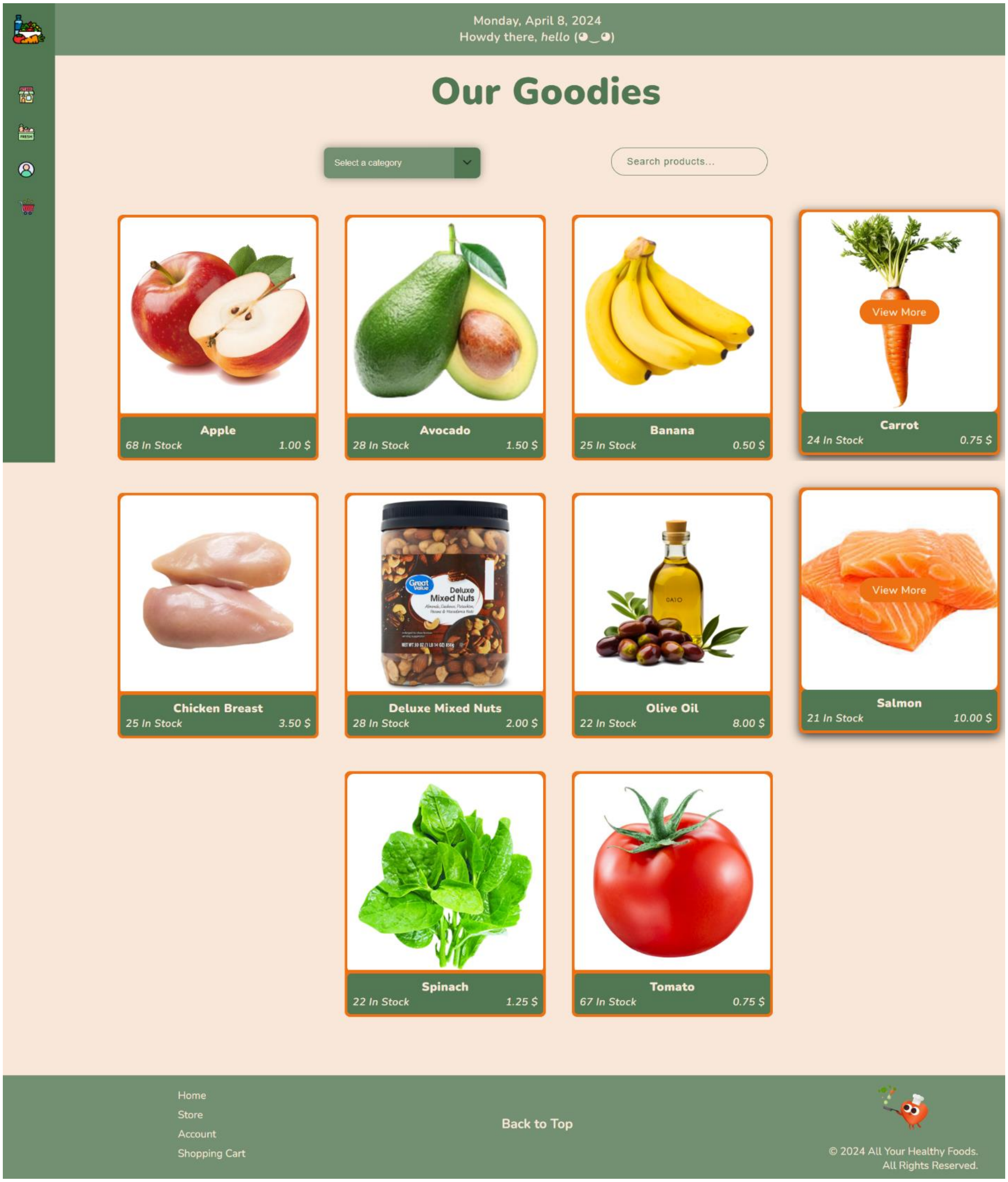
batman@gmail.com

091454364362

••••••

Browse the store catalog

Step 1: Navigate to ‘Market’ page



Monday, April 8, 2024
Howdy there, *hello* (◕‿◕)

Our Goodies

Select a category

Search products...

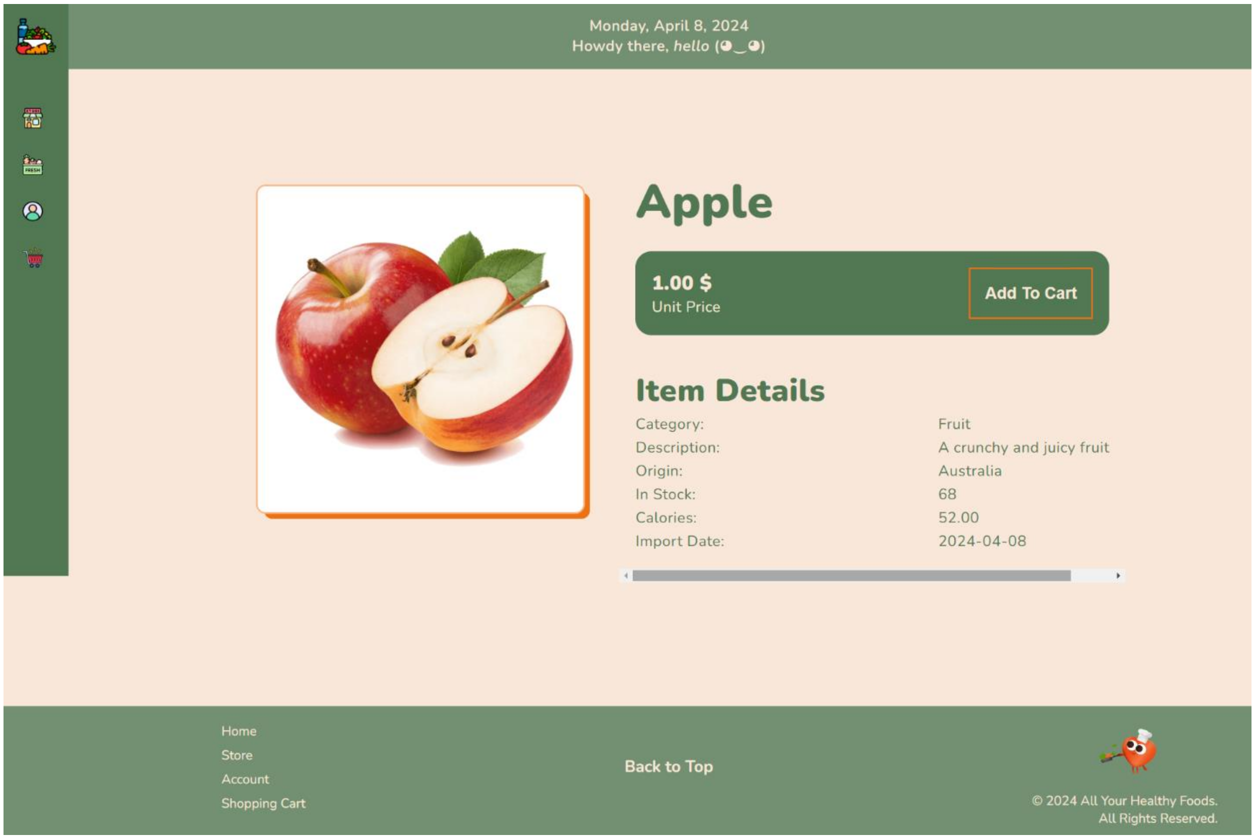
Apple	68 In Stock	1.00 $
Avocado	28 In Stock	1.50 $
Banana	25 In Stock	0.50 $
Carrot	24 In Stock	0.75 $
Chicken Breast	25 In Stock	3.50 $
Deluxe Mixed Nuts	28 In Stock	2.00 $
Olive Oil	22 In Stock	8.00 $
Salmon	21 In Stock	10.00 $
Spinach	22 In Stock	1.25 $
Tomato	67 In Stock	0.75 $

Home
Store
Account
Shopping Cart

Back to Top

© 2024 All Your Healthy Foods.
All Rights Reserved.

Step 2: Explore Product details



Monday, April 8, 2024
Howdy there, *hello* (◕‿◕)

Apple

1.00 $
Unit Price

Item Details

Category:	Fruit
Description:	A crunchy and juicy fruit
Origin:	Australia
In Stock:	68
Calories:	52.00
Import Date:	2024-04-08

Generate invoices and receipts

Step 1: Customer click on ShoppingCart

Step 2: Click ‘Check-out’

Step 3: Customer receives Invoice

Step 4: Make payment

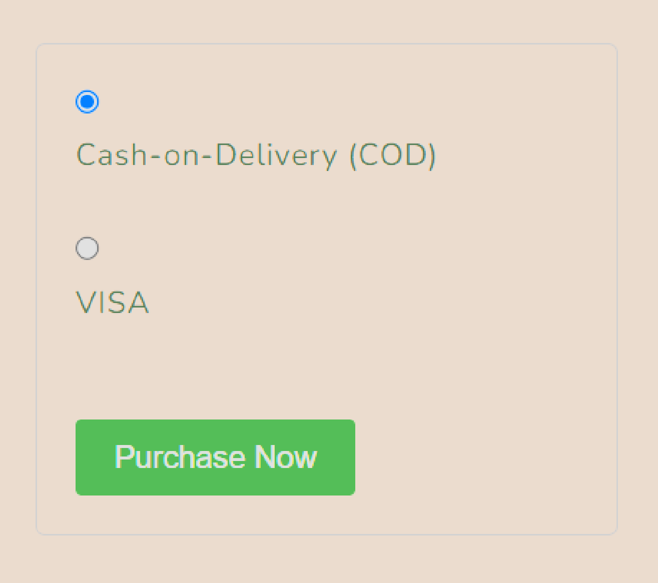
Step 5: Customer receives Receipt

Handle payments

Step 1: After receive Invoice

Step 2: Customer make payment

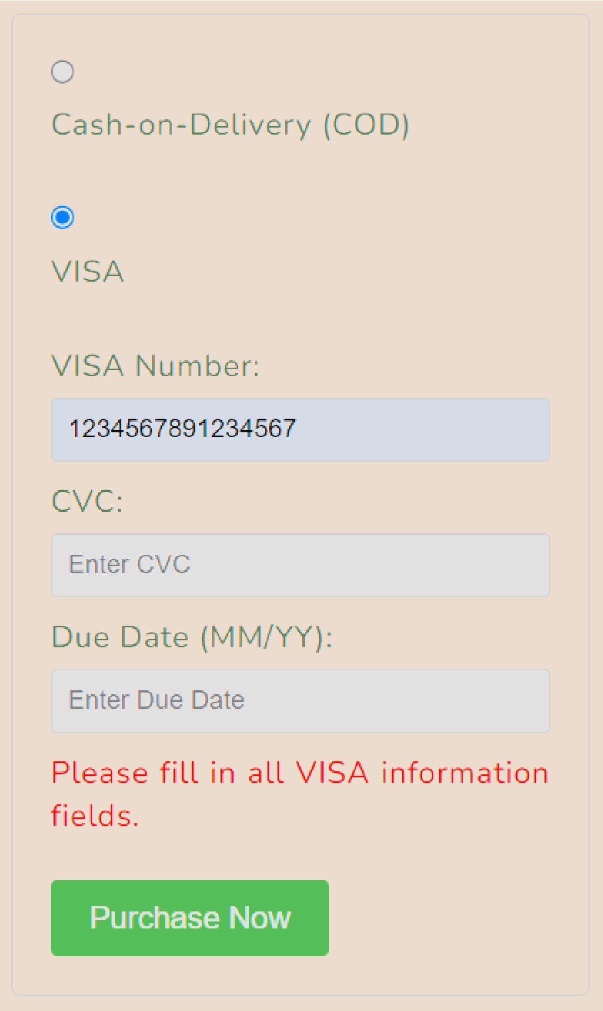
Step 3: Choose a payment method, COD or VISA



● Cash-on-Delivery (COD)

○ VISA

Step 4: If you choose pay by VISA, you must fill in your card information in correct form.



○ Cash-on-Delivery (COD)

● VISA

VISA Number: 1234567891234567

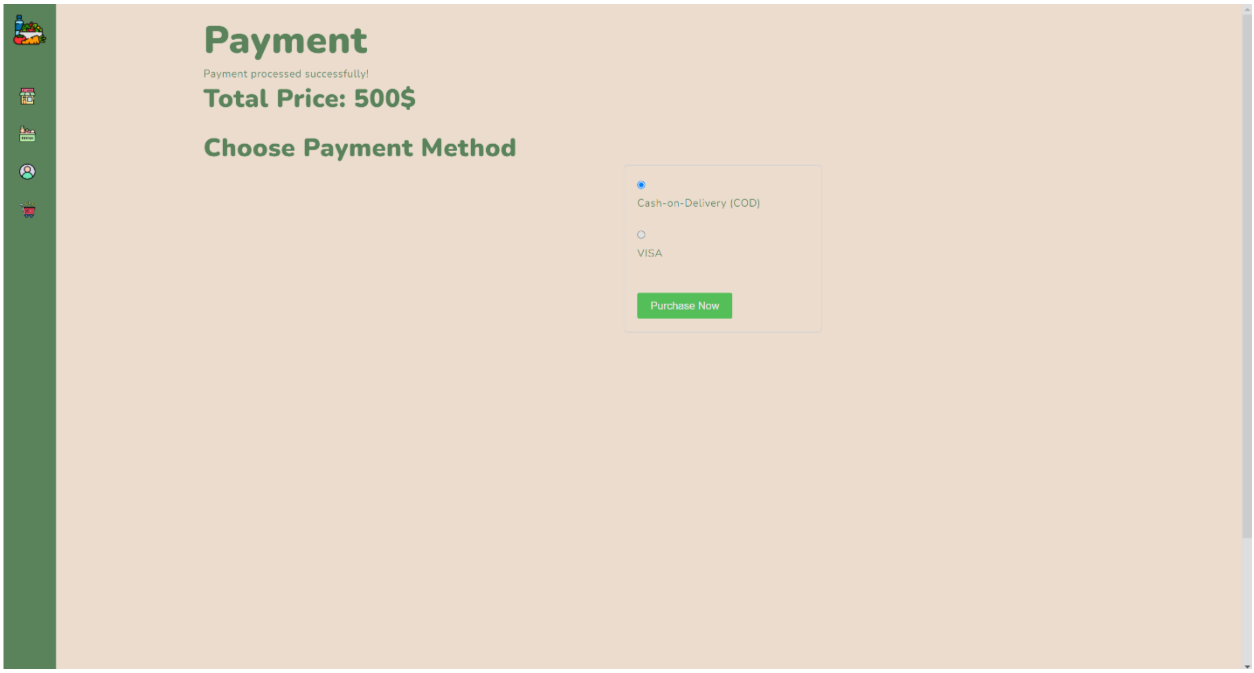
CVC: Enter CVC

Due Date (MM/YY): Enter Due Date

Please fill in all VISA information fields.

Step 4: Click ‘Purchase Now’

Step 5: If the Payment successfully, there will be a confirmation message.



Payment

Payment processed successfully!

Total Price: 500$

Choose Payment Method

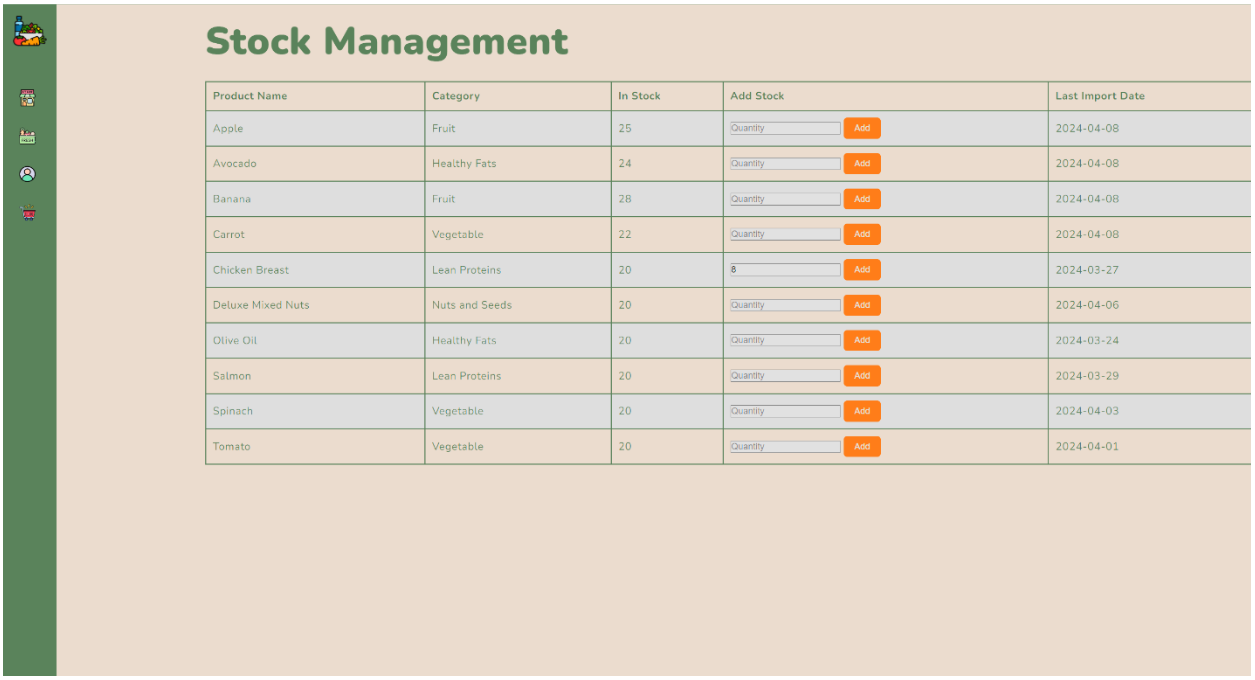
● Cash-on-Delivery (COD)

○ VISA

The owner wants to modify the Store Catalogue

Step 1: Log in as an administrator account

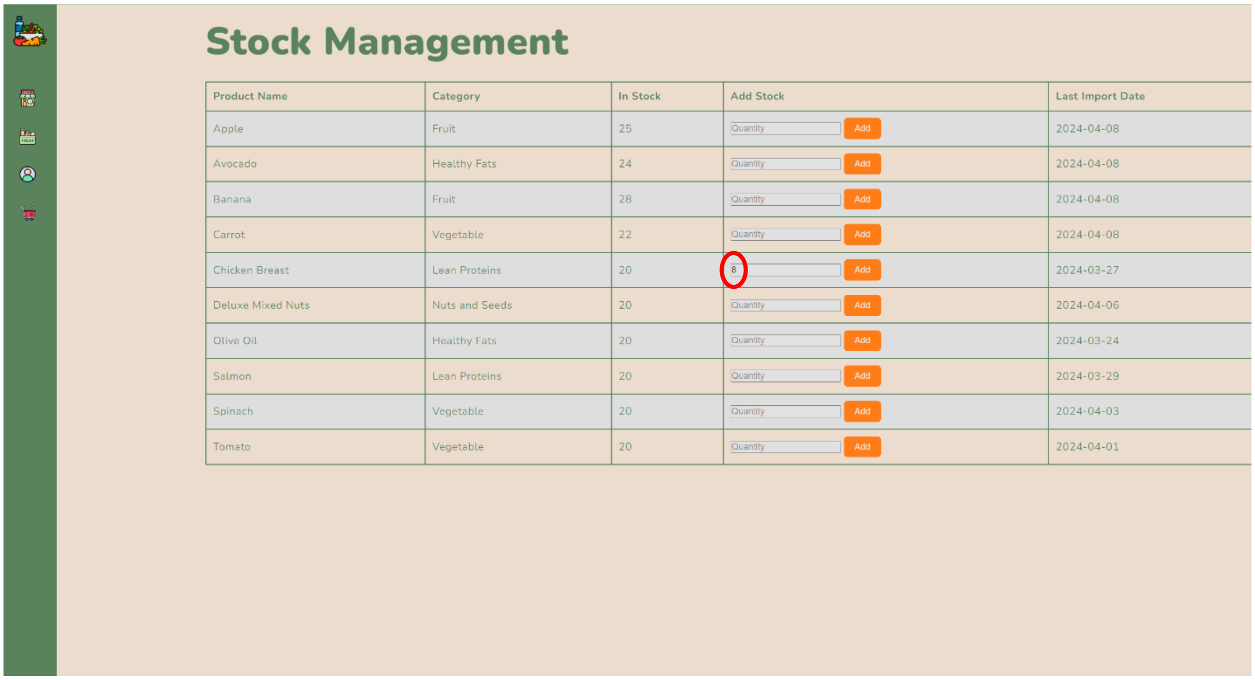
Step 2: Click on ‘Stock Management’ page



Stock Management

Product Name	Category	In Stock	Add Stock	Last Import Date
Apple	Fruit	25	Quantity Add	2024-04-08
Avocado	Healthy Fats	24	Quantity Add	2024-04-08
Banana	Fruit	28	Quantity Add	2024-04-08
Carrot	Vegetable	22	Quantity Add	2024-04-08
Chicken Breast	Lean Proteins	20	8 Add	2024-03-27
Deluxe Mixed Nuts	Nuts and Seeds	20	Quantity Add	2024-04-06
Olive Oil	Healthy Fats	20	Quantity Add	2024-03-24
Salmon	Lean Proteins	20	Quantity Add	2024-03-29
Spinach	Vegetable	20	Quantity Add	2024-04-03
Tomato	Vegetable	20	Quantity Add	2024-04-01

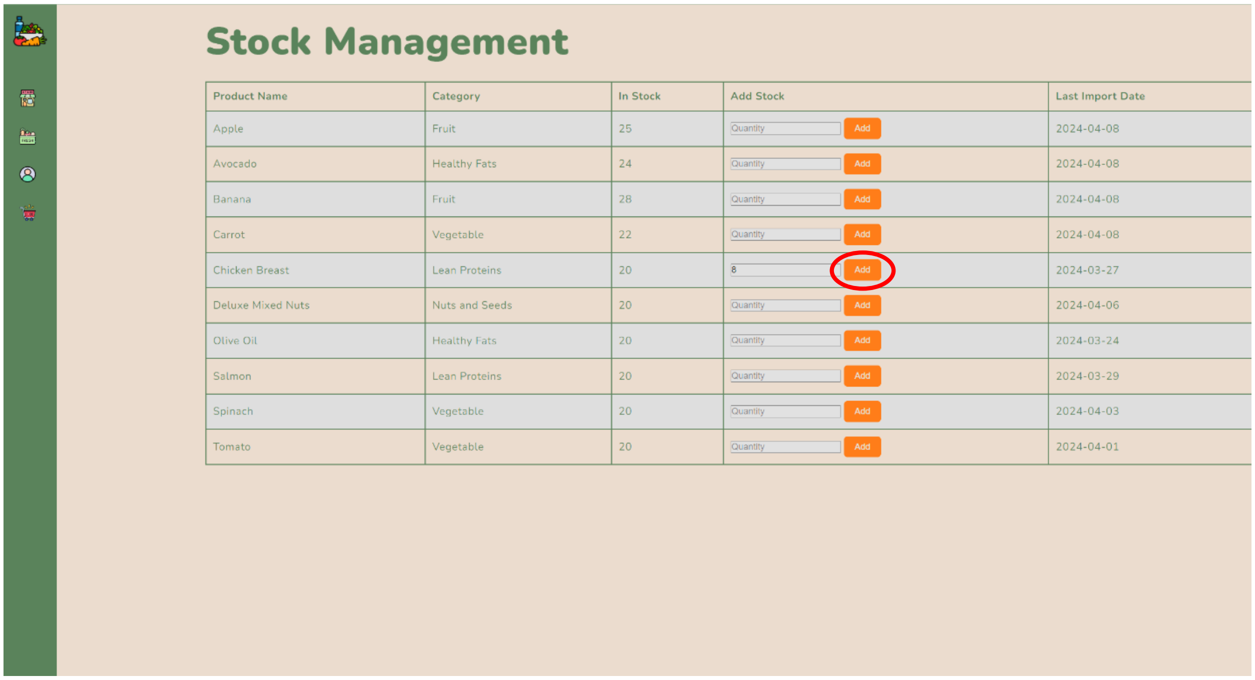
Step 3: Select a Product to add stock then fill in the form



Stock Management

Product Name	Category	In Stock	Add Stock	Last Import Date
Apple	Fruit	25	Quantity Add	2024-04-08
Avocado	Healthy Fats	24	Quantity Add	2024-04-08
Banana	Fruit	28	Quantity Add	2024-04-08
Carrot	Vegetable	22	Quantity Add	2024-04-08
Chicken Breast	Lean Proteins	20	8 Add	2024-03-27
Deluxe Mixed Nuts	Nuts and Seeds	20	Quantity Add	2024-04-06
Olive Oil	Healthy Fats	20	Quantity Add	2024-03-24
Salmon	Lean Proteins	20	Quantity Add	2024-03-29
Spinach	Vegetable	20	Quantity Add	2024-04-03
Tomato	Vegetable	20	Quantity Add	2024-04-01

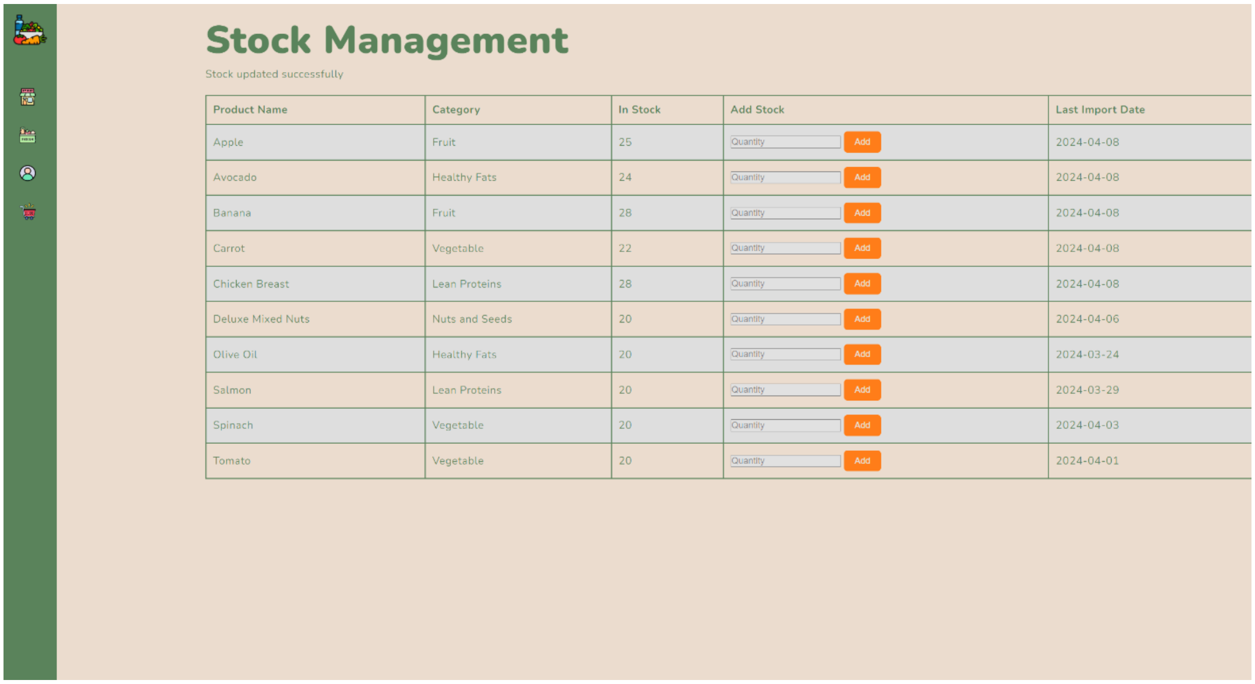
Step 4: Click ‘Add’



Stock Management

Product Name	Category	In Stock	Add Stock	Last Import Date
Apple	Fruit	25	Quantity Add	2024-04-08
Avocado	Healthy Fats	24	Quantity Add	2024-04-08
Banana	Fruit	28	Quantity Add	2024-04-08
Carrot	Vegetable	22	Quantity Add	2024-04-08
Chicken Breast	Lean Proteins	20	8 Add	2024-03-27
Deluxe Mixed Nuts	Nuts and Seeds	20	Quantity Add	2024-04-06
Olive Oil	Healthy Fats	20	Quantity Add	2024-03-24
Salmon	Lean Proteins	20	Quantity Add	2024-03-29
Spinach	Vegetable	20	Quantity Add	2024-04-03
Tomato	Vegetable	20	Quantity Add	2024-04-01

Step 5: There will be a successfully message if the Product update successfully. The ‘In Stock’ and ‘Last Insert Date’ columns will be update



Stock Management

Stock updated successfully

Product Name	Category	In Stock	Add Stock	Last Import Date
Apple	Fruit	25	Quantity Add	2024-04-08
Avocado	Healthy Fats	24	Quantity Add	2024-04-08
Banana	Fruit	28	Quantity Add	2024-04-08
Carrot	Vegetable	22	Quantity Add	2024-04-08
Chicken Breast	Lean Proteins	28	Quantity Add	2024-04-08
Deluxe Mixed Nuts	Nuts and Seeds	20	Quantity Add	2024-04-06
Olive Oil	Healthy Fats	20	Quantity Add	2024-03-24
Salmon	Lean Proteins	20	Quantity Add	2024-03-29
Spinach	Vegetable	20	Quantity Add	2024-04-03
Tomato	Vegetable	20	Quantity Add	2024-04-01

V. References

Szanto, K. (no date) PSR-12: Extended coding style - PHP-fig, PHPFIG Blog. Available at: https://www.php-fig.org/psr/psr-12/ (Accessed: 08 April 2024).

Group 8 (2024). Assignment 3 [Appendix]. Software Architectures and Design (SWE30003). [PDF file].