System Configuration

Little Farmer Supply Chain System

Created By:
Devin Chee Peng
Elaine Ting Yi Ling
Chen Jun Yao
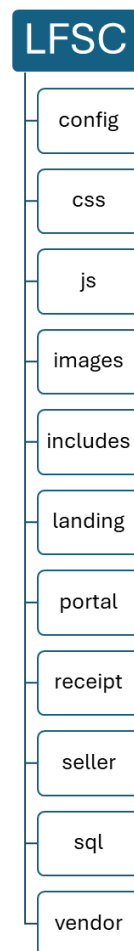Gan Jun Yu
Wong Chang Shan

Contents

1.0 Introduction

This document outlines the steps and procedures required to deploy the Little Farmer e-commerce system into a production environment. It is intended for use by system administrators, IT professionals, and any technical staff involved in the deployment process. By following the guidelines provided here, you will ensure a smooth and efficient transition from development to a fully operational system, ready to serve our community of farmers and customers.
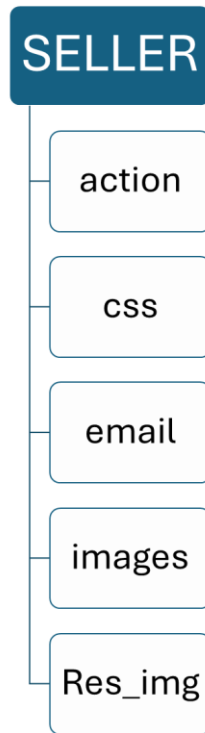
2.0 File Structure

Overall File Structure

This file structure mainly contains files for "Market", "Portal", "Customer Login", and "SQL".

```
LFSC
 ├─ config
 ├─ css
 ├─ js
 ├─ images
 ├─ includes
 ├─ landing
 ├─ portal
 ├─ receipt
 ├─ seller
 ├─ sql
 └─ vendor
```

Seller File Structure

This file structure mainly contains files for "Seller" Website.

```
SELLER
├── action
├── css
├── email
├── images
└── Res_img
```

Admin File Structure

This file structure mainly contains files for "Admin" Website.

```
ADMIN
├── action
├── css
├── email
├── images
└── Res_img
```
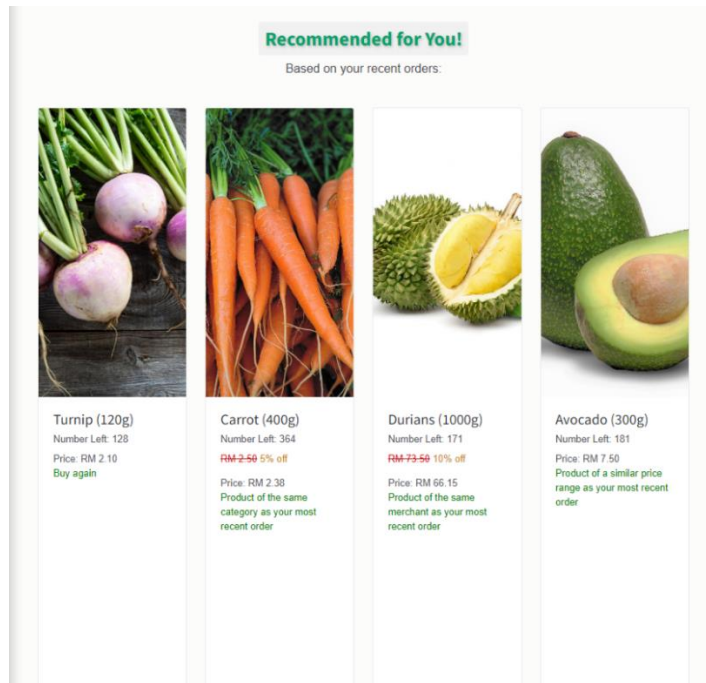
3.0 System Overview

## Storefront

- market.php
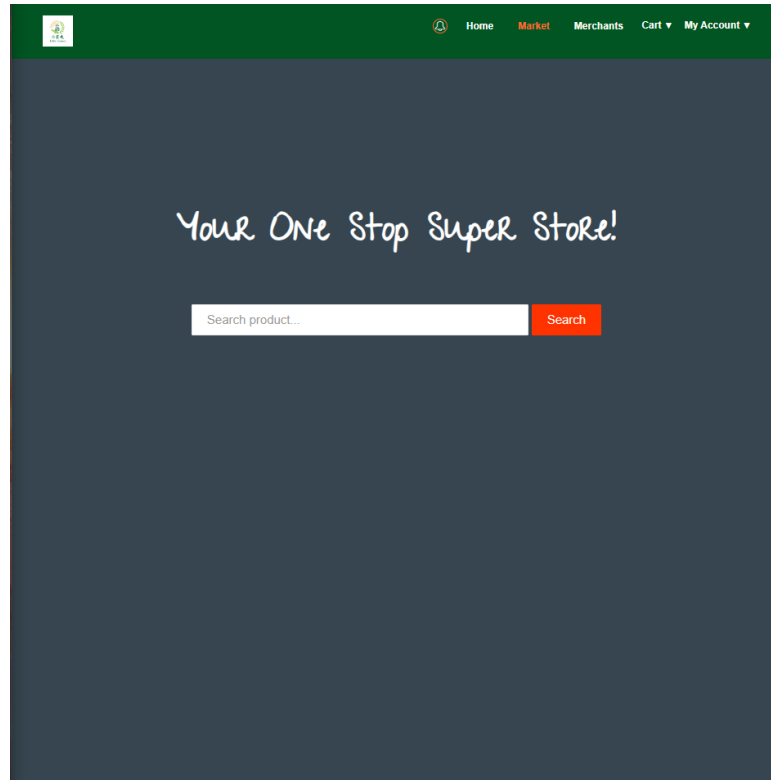
Upon entry to the webpage, the user will be presented with a top navigation bar and a search bar.

Right below the user will be recommended products based on:
1. Previously bought product
2. Product of the same category as your most recent order
3. Product of the same merchant as your most recent order
4. Product of a similar price range as your most recent order
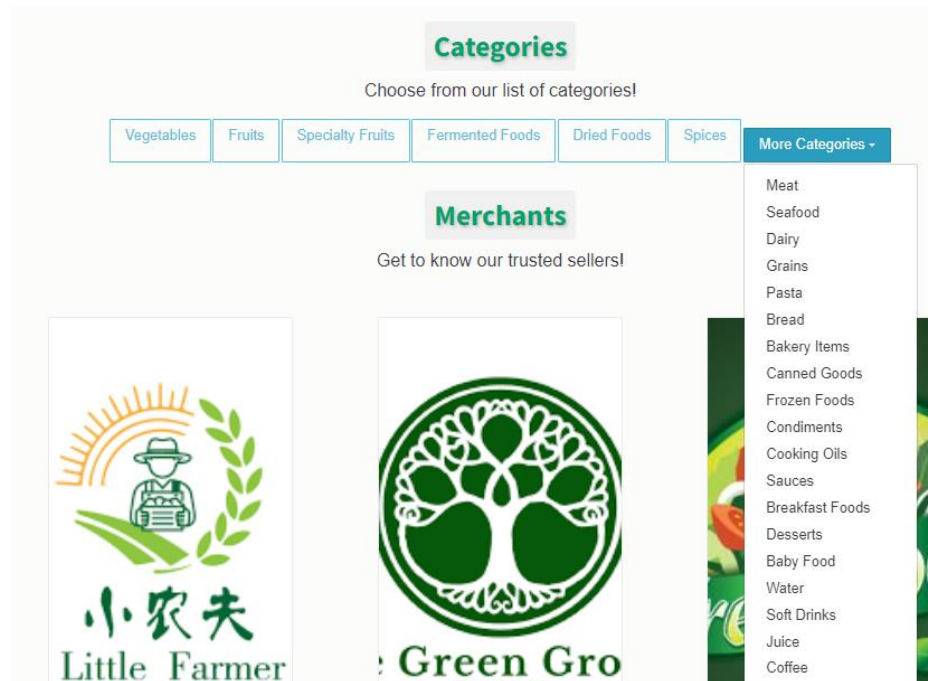
Below that the user will be presented with different product categories, with a drop-down menu that will show more categories.
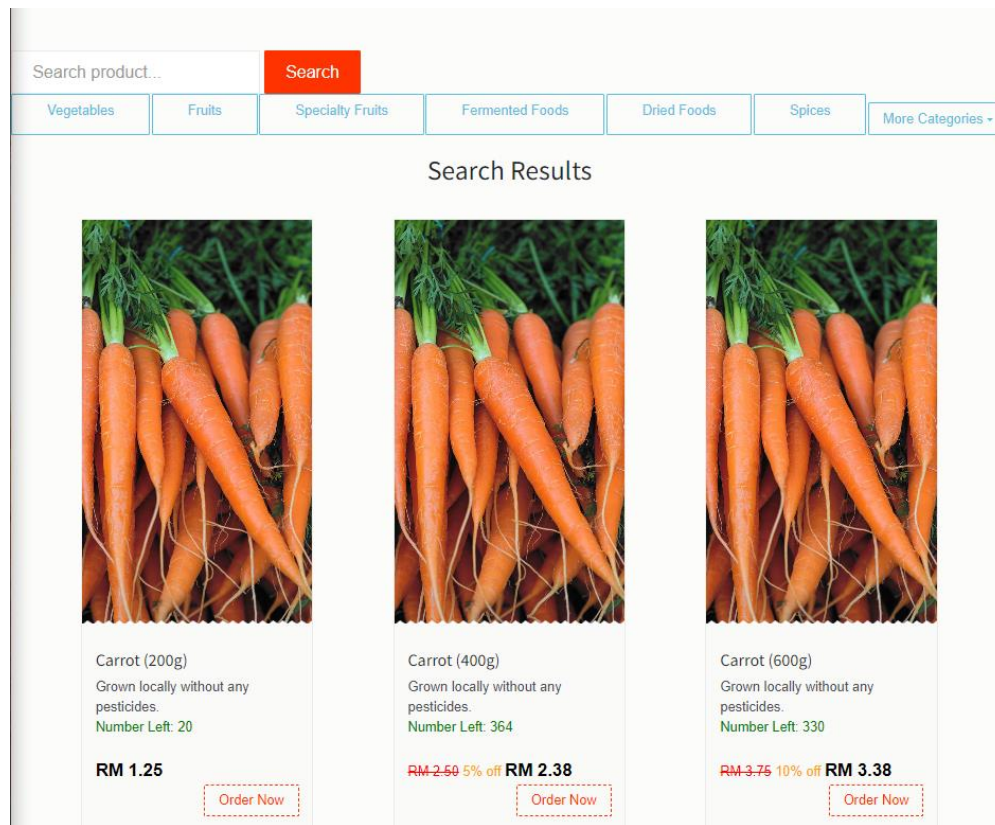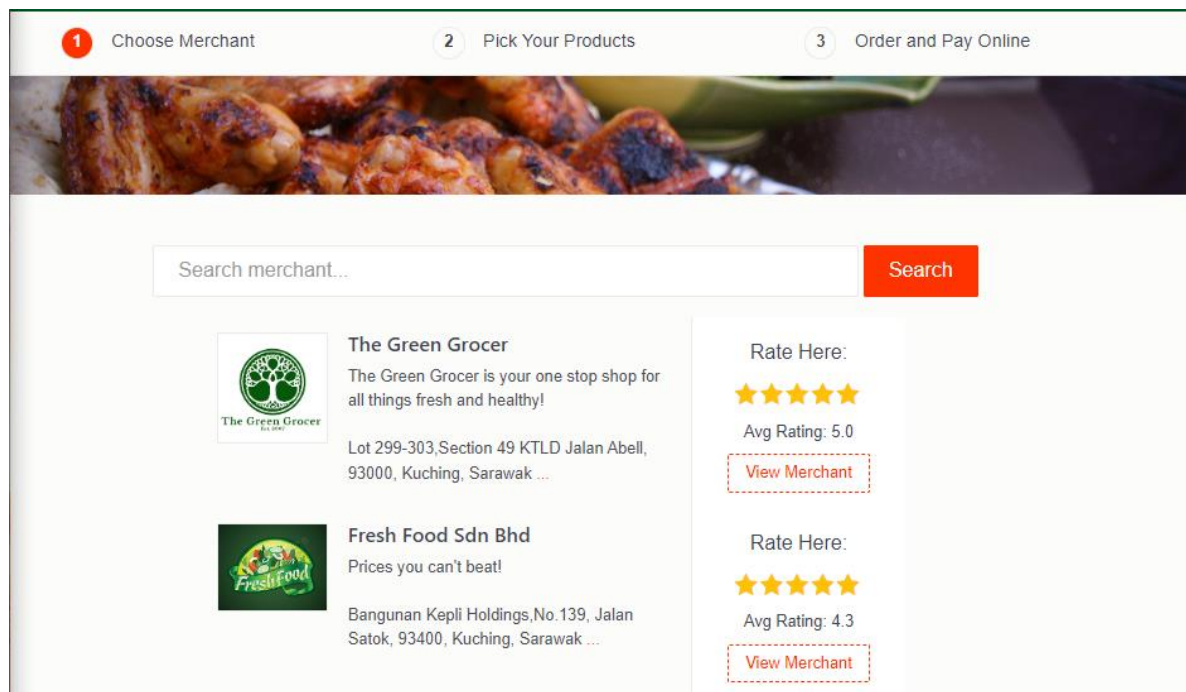

When the categories are clicked, products from the category will be shown (like when the "Vegetables" category is clicked as shown above).
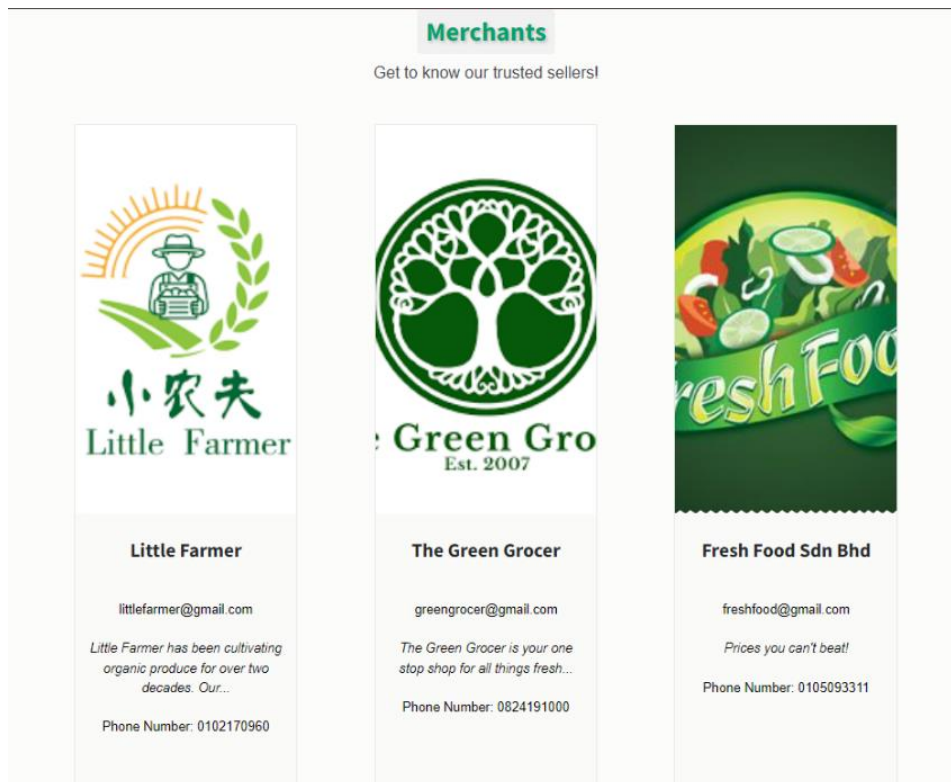
Lastly the top merchants will be shown in the "Merchants" section.

- merchants.php

Upon entering the page, the user will be presented with a searchbar that can be used to search for merchants. Below the merchants present in the store will be presented to the user. Beside each store, the user will be able to rate the merchant. Each user will only be able to rate a merchant once with the new rating overriding the old rating. Below the rating stars, the average rating will be shown.
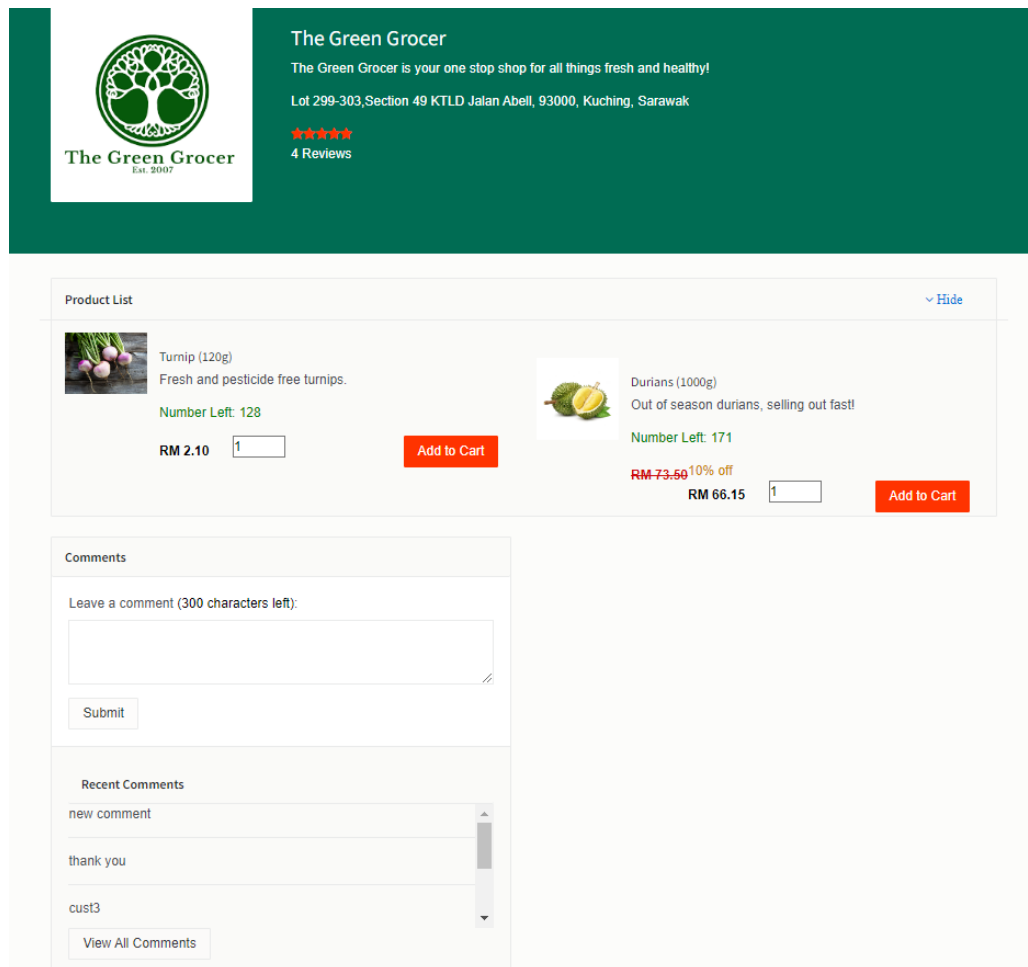
When a store is clicked, the user will view its products. The user can change the quantity of products to be purchased by typing into the text box. Products with "More options" beside the name will allow the user to select a different quantity and price of the products.

Product List

Avocado (300g) **More options**
Healthy fat for healthy dishes!
Number Left: 181
RM 7.50 1 **Add to Cart**

The product list can be hidden to go straight to the comments section. A user can leave comment by typing the comment into the comment box that has a character limit of 300 and obscenity filtering and submit the comment with the button. The user can view all the comments of the merchant by clicking on "View All Comments".

All Comments

Sort By: Newest First

| Customer Name	Posted on
j****	2024-05-06 17:40:07
cust5	
C****	2024-05-06 17:39:47
cust4	
de****	2024-05-06 17:39:31
cust3	
m****	2024-05-06 17:35:33
thank you	
E****	2024-05-06 16:14:37
new comment	

1

The "all_comments.php" page shows the user all comments have a sorting function, user anonymization and pagination.

- Cart
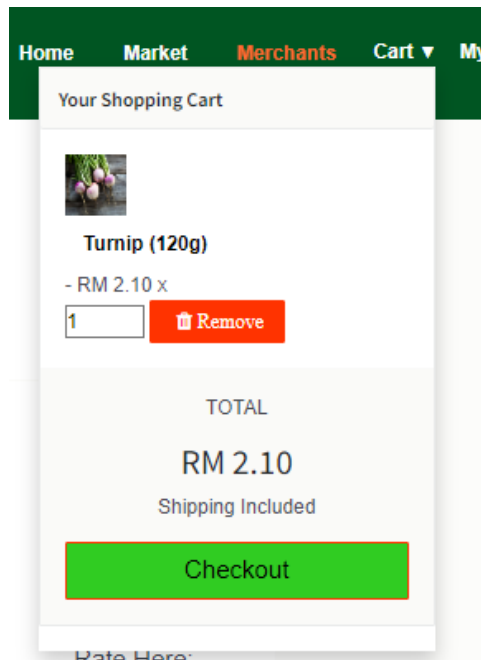
When the user clicks on "Cart", the user will be able to view their cart and remove or edit the quantity of products bought. The user can checkout by clicking "Checkout".

- Notifications
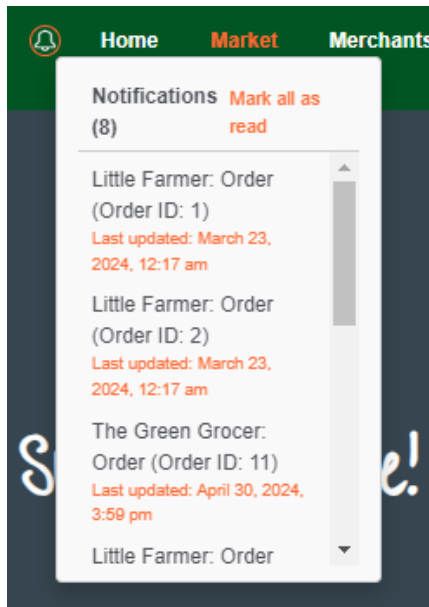
When the user hovers over the bell icon, the user can view orders and order status.

- your_account.php

The user is meant to update account details if necessary and bind their Telegram here.

- your_orders.php

Order No.	Order ID#	Seller	Number of Products	Total Price	Status	Date	Action
1	16	The Green Grocer	1	RM 2.1	Processing	2024-05-13	🗑
2	15	Little Farmer	3	RM 14.74	Processing	2024-05-13	🗑
3	14	The Green Grocer	2	RM 68.25	Processing	2024-05-13	🗑
4	13	Little Farmer	4	RM 30.62	Processing	2024-05-13	🗑
5	12	Little Farmer	3	RM 14.74	Processing	2024-05-13	🗑
6	11	The Green Grocer	1	RM 12.6	Processing	2024-04-30	🗑
7	2	Little Farmer	2	RM 5.98	✅Delivered	2023-12-28	📄
8	1	Little Farmer	2	RM 5.76	✅Delivered	2023-12-28	📄

The user will use this page to view products that were ordered, status of ordered products, cancel orders (the bin icon) or view receipts (the blue icon).

Little Farmer

AB102 Ground Floor Parcel 2586-1-9 Lorong Persiaran Bandar Baru Batu Kawa 3D Batu Kawah New Township Jalan Batu Kawa 93250 Kuching Sarawak

Tel: 0102170960

RECEIPT

RECEIPT NO. : LF512

Date: 2024-05-25 01:31:45

No	Product Name	Quantity	Price
1	Carrot	1	2.38
2	Cabbage	1	3.60

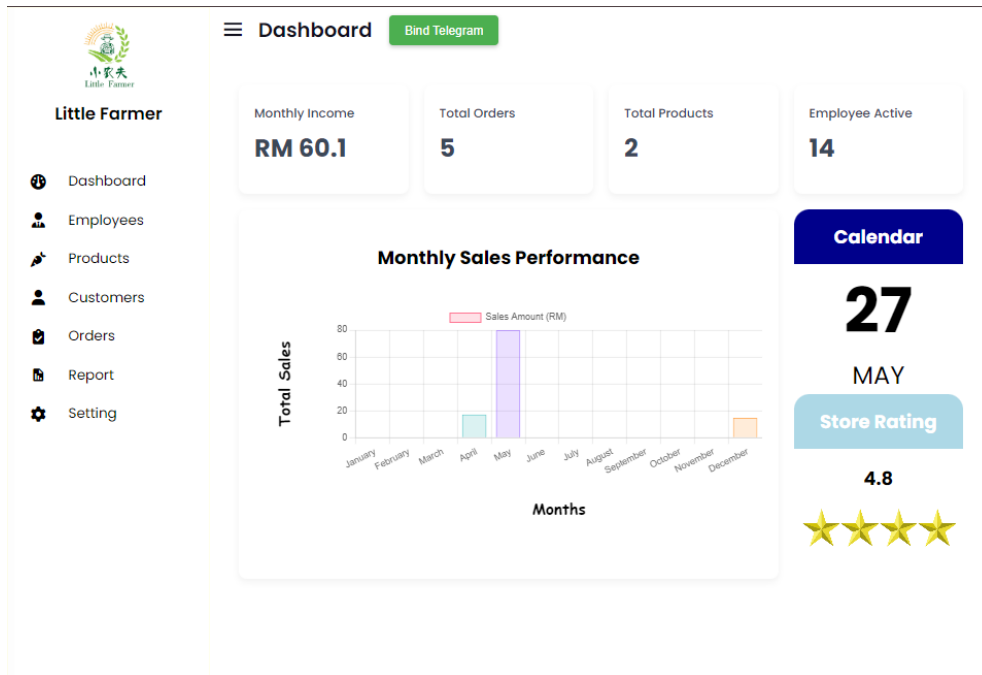**Total Items: 2**   **Total Price: RM 5.98**

Thank you for shopping with us! We appreciate your business.

Print Receipt

This is an example of a receipt.

Seller

- Dashboard.php

This page is meant for the merchant to glance information about their store and bind their telegram.

- employees.php

Employees

			+Add Employee				

Records per page: 5

Search: [Search for names..]

Emp ID#	Name	Gender	Contact Number	Email	Job Title	Status	Action
51231221125018	4	Male	1	6	Accounting	Active	👁 ✏ 🗑
51231221125255	2	Male	1	ryanwong179@gmail.com	Delivery	Active	👁 ✏ 🗑
51231221125305	2	Female	3	4	Delivery	Active	👁 ✏ 🗑
51231221125125	4	Female	5	6	Accounting	Active	👁 ✏ 🗑
51231221125453	333	Female	666	777	Accounting	Active	👁 ✏ 🗑

Showing 1-5 of 14 Records

1 2 3 Next

This page is meant for the merchant to add and manage employees.

- products.php

Products

			+Add Product

Records per page: 5

Search: [Search for names..]

Product Code	Image	Product Name	Description	Status	Action
A0003		Avocado	Healthy fat for healthy dishes!	Active	👁 ✏ 🗑
C0012		Cabbage	Fresh cabbage grown without any pesticides.	Active	👁 ✏ 🗑
C0022		Carrot	Grown locally without any pesticides.	Active	👁 ✏ 🗑
C0032		Fruit Cucumber	Timun for dishes.	Active	👁 ✏ 🗑
A0001		Green Apple	Freshest apples in Malaysia. Sold in packs of 500g.	Active	👁 ✏ 🗑

Showing 1-5 of 6 Records

1 2 Next

This page is meant for merchants to add products to their store.

- customers.php

NO#	Username	Name	Email	Phone Number	Address	Action
1	cust1	Jason Lim	gerdenremseyjr@hotmail.com	0218882102	82 Saradise	📷
2	cust2	Ashley Tan	heyitsashleytanxd@gmail.com	0217719273	1024 Saberkas	📷
3	skyfarm96	Michael Chai	skyangel77@gmail.com	0128789600	Eden 2, MeiMei Road	📷
4	cust5		allianzwierdchamp@gmail.com			📷

Showing 1-4 of 4 Records

The merchant can use this page to view details about the customers that have bought from their store and set custom prices for products for a specific customer.

- cust_comments.php

Sort By: Newest First

Customer Name	Posted on
S****	2024-05-25 22:49:27
great service!	
Sj****	2024-03-24 22:41:52
very nice products!	

The merchant can sort and view anonymized comments from customers about their store here.

- orders.php

☰ Orders

+Add Order

Records per page: 5

Search: [Search for names..]

| Order ID#	Name	Contact Number	Order Date	Payment Type	Status	Action
39	Jason Lim	0218882102	2024-05-25	CASH ON DELIVERY	Active	👁 🗑
53	Jason Lim	0218882102	2024-05-27	CASH ON DELIVERY	Active	👁 🗑
64	William Donald	0123456789	2024-05-27	CASH ON DELIVERY	Active	👁 🗑
68	Peter Tan	0123254370	2024-05-27	CASH ON DELIVERY	Active	👁 🗑
82	Kobe Bryant	0143245367	2024-05-27	CASH ON DELIVERY	Active	👁 🗑

Showing 1-5 of 6 Records

1 2 Next

The merchant can view and fulfil all orders here.

- product_report.php
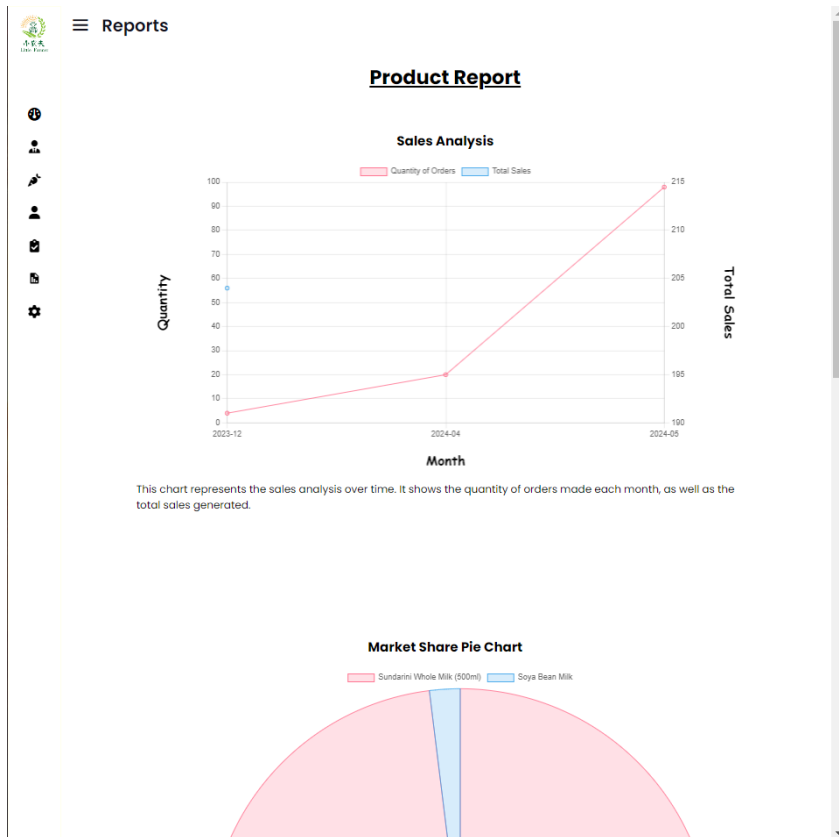
This page is meant for the merchant to get easily digestible information about the financial health of their store through the graphs and summaries that are generated.

- sales_report.php

This page is meant for the merchant to get a detailed sales report for a specified time period.

- setting.php

This page is meant for the merchant to update the store details and to set a new password.

Admin

- dashboard.php

The admin can glance at the overall health of the store.

- employees.php

This page is meant for the admin to add and manage other admins.

- sellers.php

The admin can view all the merchants that are on the platform on this page.

- comment_inspection.php

The admin can see and delete all comments that are on the platform.

- seller_inspection.php

Sellers

Records per page: 5

Search: [Search for names..]

Shop ID#	Title	Email	Phone Number	Action
NO SELLERS UNDER INSPECTION				

Showing 0-0 of 0 Records

The admin can approve new merchants on this page.

- users.php

Customers

[+Add Customer]

Records per page: 5

Search: [Search for names..]

NO#	Username	Name	Email	Phone Number	Location	Action
1	cust1	Jason Lim	gerdenremseyjr@hotmail.com	0218882102	82 Saradise	⊘
2	cust2	Ashley Tan	heyitsashleytanxd@gmail.com	0217719273	1024 Saberkas	⊘
3	cust3	William Donald	willdona@youtube.com	0123456789	YouTube headquarters at the Spring	⊘
4	stephentan44	Stephen Tan	stephentan44@gmail.com	0102170960	547 Lorong 3 Rose Garden 93250 Kuching Sarawak	⊘
5	angrychef	John Divasukarno	john@dailychefshow.net	0134569780	347 CityOne	⊘

Showing 1-5 of 13 Records

[1] [2] [3] [Next]

The merchant can view and ban all the users on the platform.

- products.php

The admin can view and edit all products that are on the platform.

- product_inspection.php

Admin can inspect the newly and modified product, so make sure everything is suitable selling in platform.

- admin_report.php
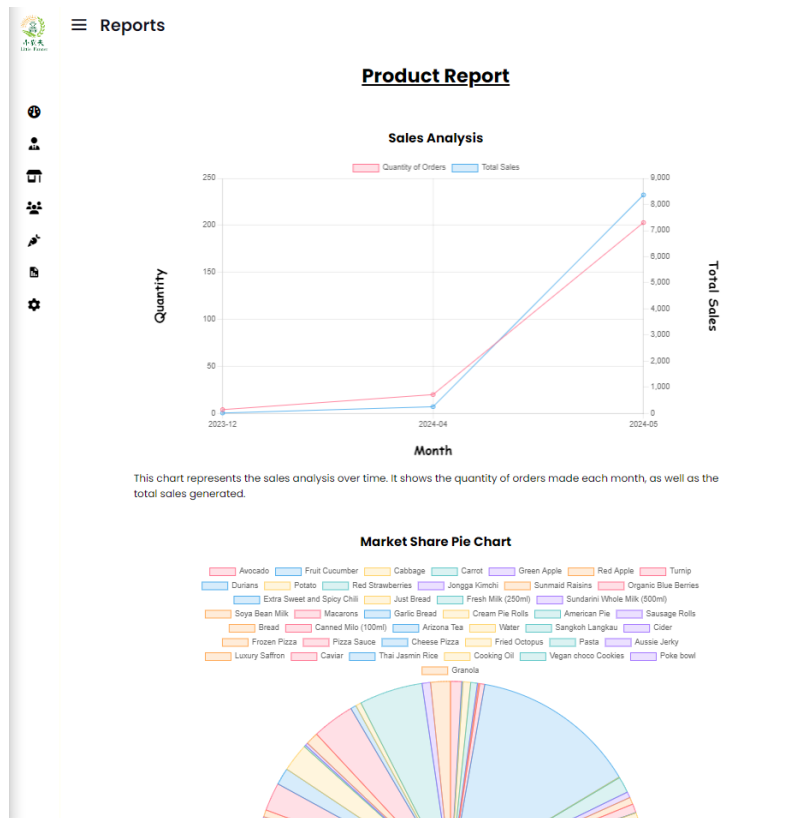
This page gives the admin an overview of the website through graphs and summaries.

- sales_report.php

Start Date: 02/27/2020 　End Date: 05/27/2024 　Search 　Excel

ITEM NO	PRODUCT NAME AND WEIGHT	PRICE (RM)	QUANTITY	TOTAL (RM)	DATE	DETAILS
1	Carrot (400)	2.50	1	2.50	2023-12-28	Details
2	Carrot (400)	2.50	1	2.50	2023-12-28	Details
3	Carrot (400)	2.50	1	2.50	2023-12-28	Details
4	Carrot (600)	3.75	1	3.75	2023-12-28	Details
5	Carrot (600)	3.75	1	3.75	2023-12-28	Details
6	Carrot (600)	3.75	1	3.75	2023-12-28	Details
7	Carrot (400)	2.50	1	2.50	2023-12-28	Details
8	Carrot (400)	2.50	1	2.50	2023-12-28	Details
9	Carrot (400)	2.50	1	2.50	2023-12-28	Details
10	Cabbage (900)	6.00	1	6.00	2023-12-28	Details
11	Cabbage (900)	6.00	1	6.00	2023-12-28	Details
12	Cabbage (900)	6.00	1	6.00	2023-12-28	Details
13	Turnip (120)	2.10	2	4.20	2024-04-20	Details

This page gives the admin a sales report of the entire website.

- settings.php

≡ **Setting**

Personal Info

User Name	admin
Name	Wong
Email	ryan@gmail.com
Phone Number	01151385427

Update Profile

Security

Old Password	
New Password	
Confirm Password	

Update Password

The admin can edit account details here.

Database

17 table(s)

Table	Fields
admin	`adm_id`, `adm_Name`, `username`, `password`, `email`, `contact_num`, `code`, `u_role`, `store`, `date`, `storeStatus`, `chat_id`, `email_token`, `token_expiration`, `email_verified`
categories	`categories_id`, `categories_name`, `categories_active`, `categories_status`
custom_prices	`price_id `, `user_id `, `price `
orders	`order_id`, `order_date`, `client_name`, `client_contact`, `sub_total`, `total_amount`, `paid`, `due`,

	`payment_type`, `order_status`, `user_id`, `order_belong`, `last_updated`, `is_seen`
order_item	`order_item_id`, `order_id`, `priceID`, `quantity`
payment_receipts	`receipt_id`, `order_id`, `receipt_path`, `upload_date`, `status`
product	`product_id`, `productCode`, `product_name`, `product_image`, `descr`, `categories_id`, `owner`, `product_date`, `lowStock`, `status`
restaurant	`rs_id`, `c_id`, `title`, `email`, `phone`, `url`, `o_hr`, `c_hr`, `o_days`, `address`, `image`, `date`, `description`
res_category	`c_id`, `c_name`, `date`
seller_tg_verification	`id`, `adm_id`, `code`, `chatId`, `expiration`
tblemployee	`empNo`, `empID`, `pusername`, `password`, `icNo`, `cempname`, `empgender`, `empcontact`, `empemail`, `empjob`, `code`, `u_role`, `empstore`, `empstatus`, `emailstatus`, `email_token`, `email_verified`
tblprice	`priceNo`, `productID`, `proQuant`, `proWeight`, `proPrice`, `proDisc`
tblverification	`validNo`, `frontImg`, `backImg`, `faceImg`, `imgStatus`, `comment`, `storeID`
tg_verification	`id`, `userId`, `code`, `chatId`, `expiration`
users	`u_id`, `username`, `f_name`, `l_name`, `fullName`, `gender`, `dob`, `email`, `phone`, `password`, `address`, `status`, `date`, `chat_id`, `notifications_enabled`, `email_token`, `token_expiration`, `email_verified`
user_comments	`id`, `user_id`, `res_id`, `comment`, `created_at`
user_ratings	`id`, `user_id`, `res_id`, `rating`

Digital Ocean
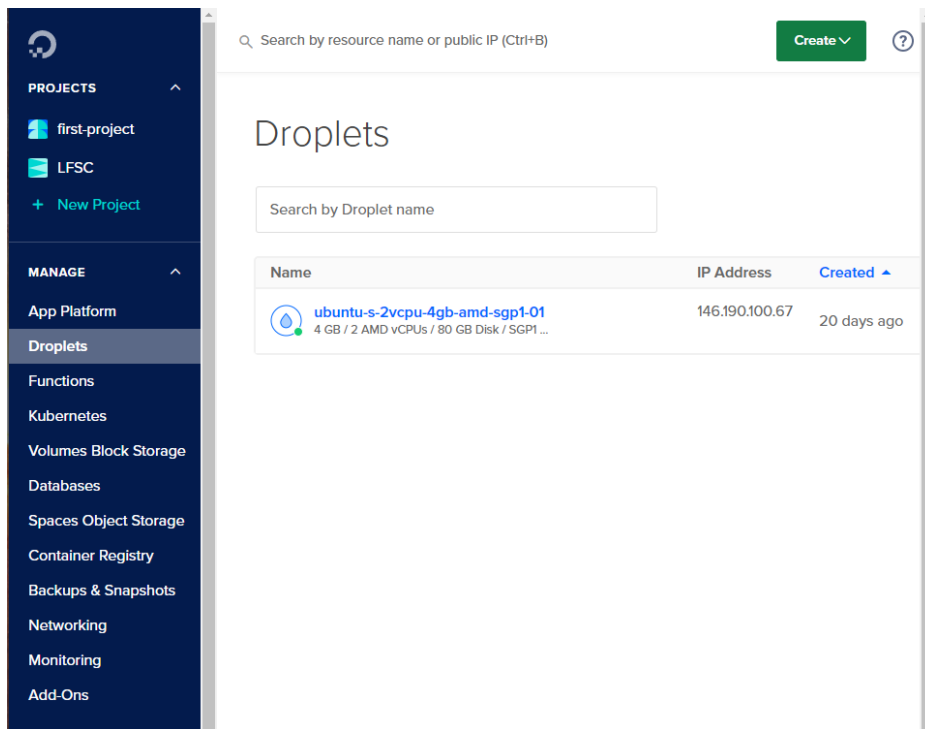
This is where the admin can find a server to deploy.
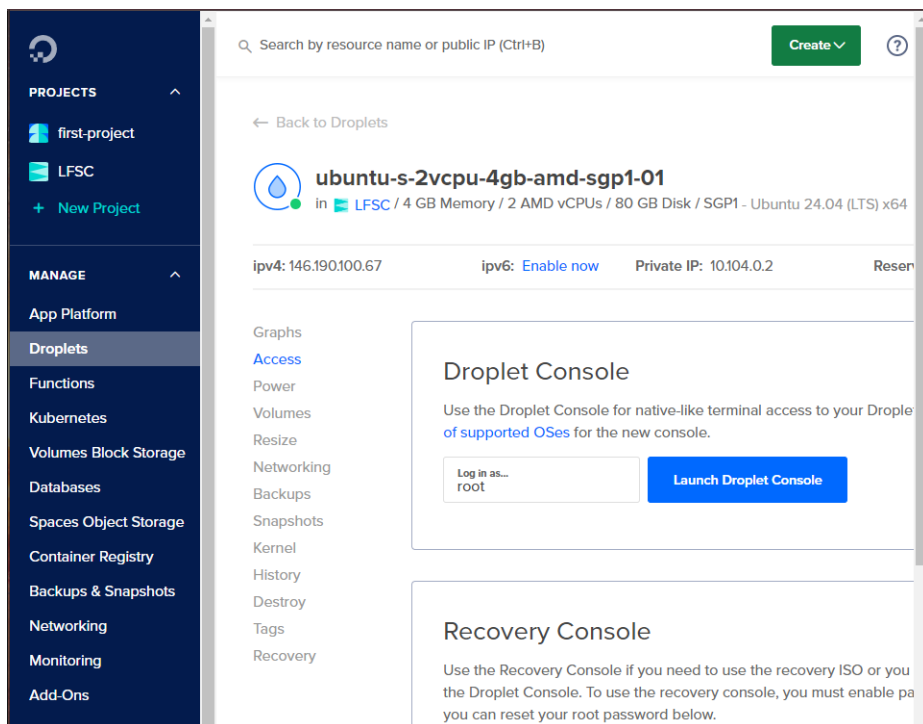
In the "Access" tab, the admin can deploy the droplet console.
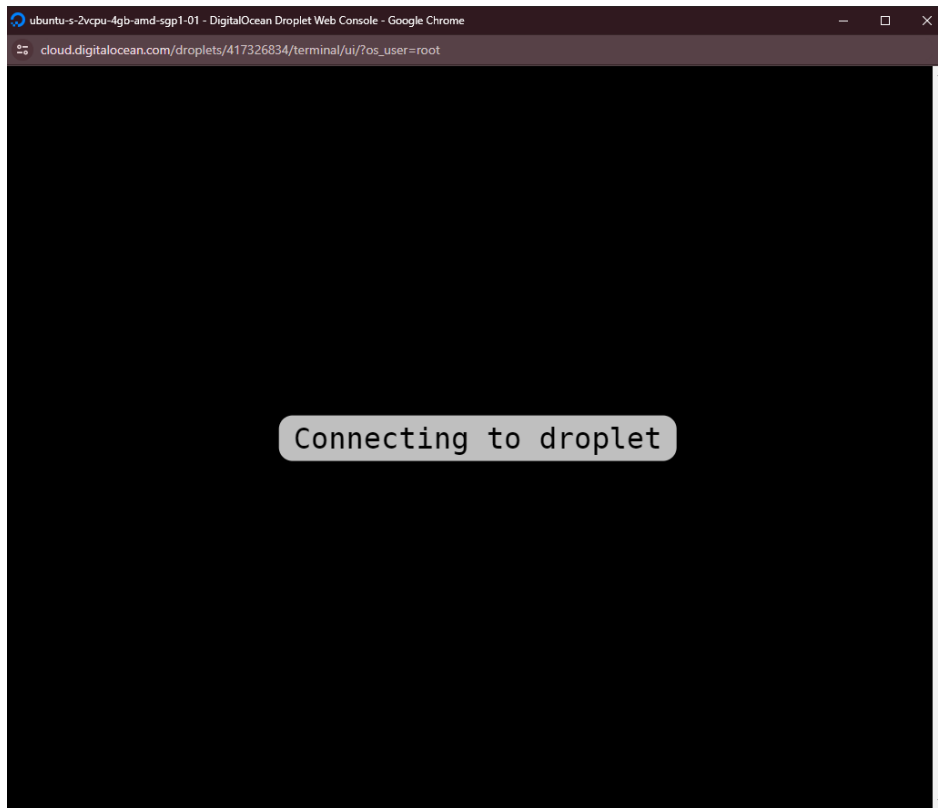
When "Launch Droplet Console" is clicked, the droplet console will be launched, and the server will be running.
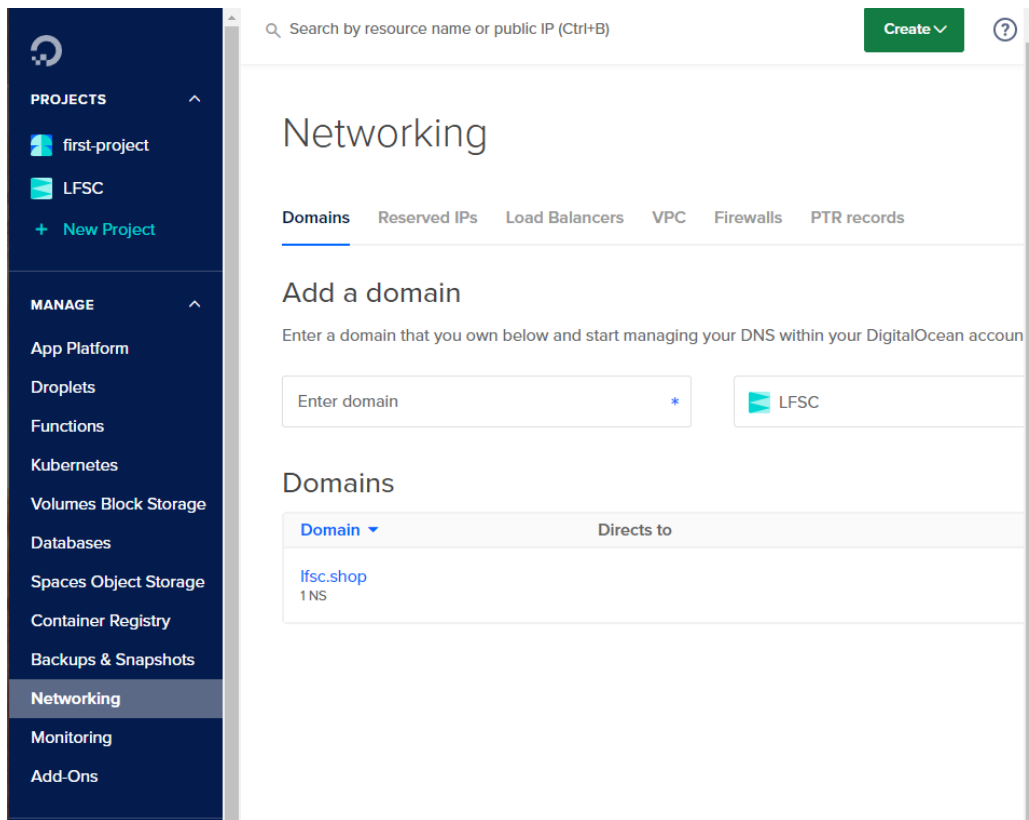
In the "Networking" tab, the admin can set the domain name of the website.

4.0 Installation Procedures

GitHub

- Download GitHub desktop from https://desktop.github.com and login.

localhost

- Download XAMPP from
  https://www.apachefriends.org/download.html and install (use
  default path).
- Open XAMPP control panel to start Apache and MySQL.
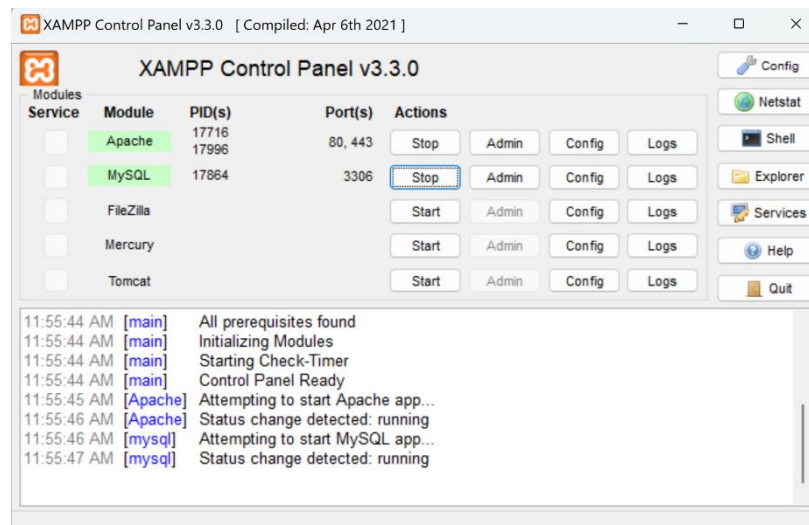
- Clone the repo using File > Clone repository and clone the repository
  (repo) into the **htdocs** folder.
- Type **localhost/phpmyadmin** in your web browser and press enter.
- Create a new schema named **store**.
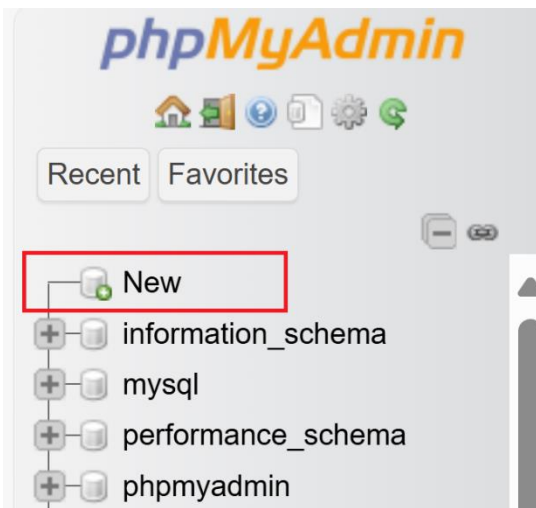
- After creating, click on the newly created schema **store**, and then
  click on import. Navigate to the repo folder > sql > store.sql. E.g.
  C:\XAMPP\htdocs\LFSC\sql\store.sql. After choosing the sql file,
  click on import at the bottom.

- Now, you can access the website on your computer by typing in localhost/{repo name}. E.g. localhost/LFSC. You will automatically be redirected to the index page.
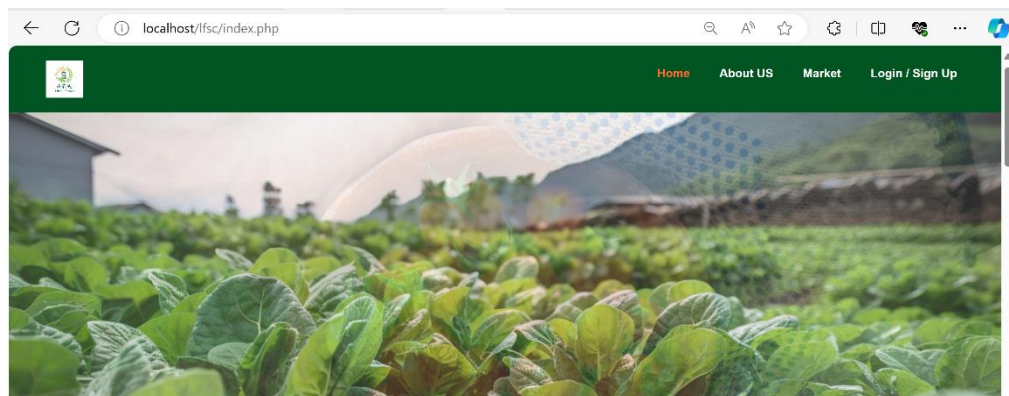
Digital Ocean

1.

Click on "Droplets".

2.

Configure Droplet server then click "Create Droplet". "ubuntu-s-2vcpu-4gb-amd-sgp1-01" was used for the configuration of this server.

3.
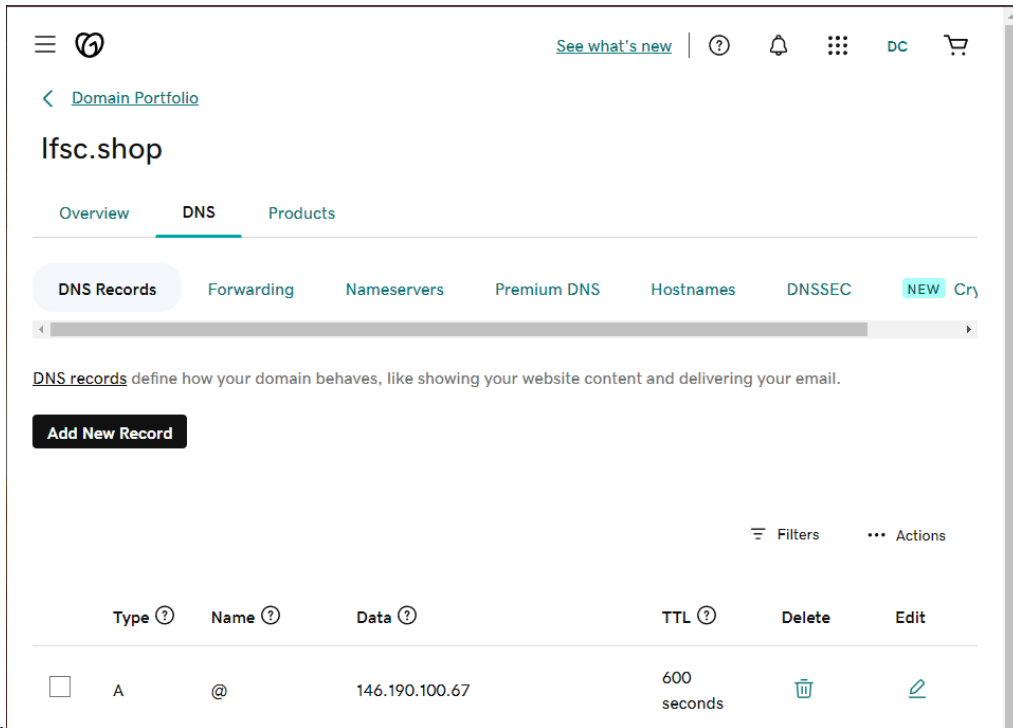
Go to a hosting service provider (GoDaddy in this case) and purchase a domain. Set the domain IP address to the IP address of the Droplet Server.

4.
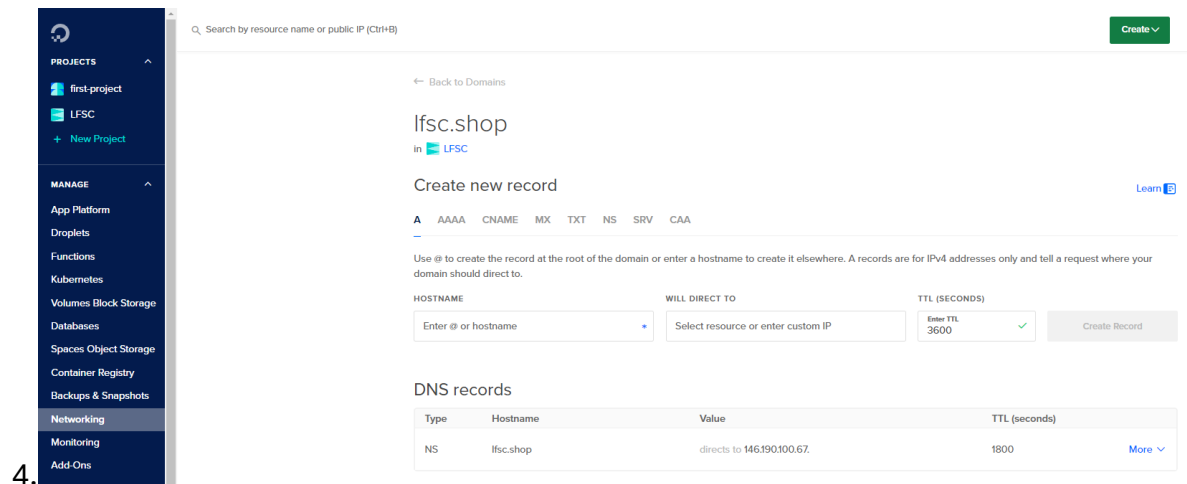
Go to the "Networking" tab and click on "Domains". Then add "lfsc.shop" (or whatever your domain is called).

5.
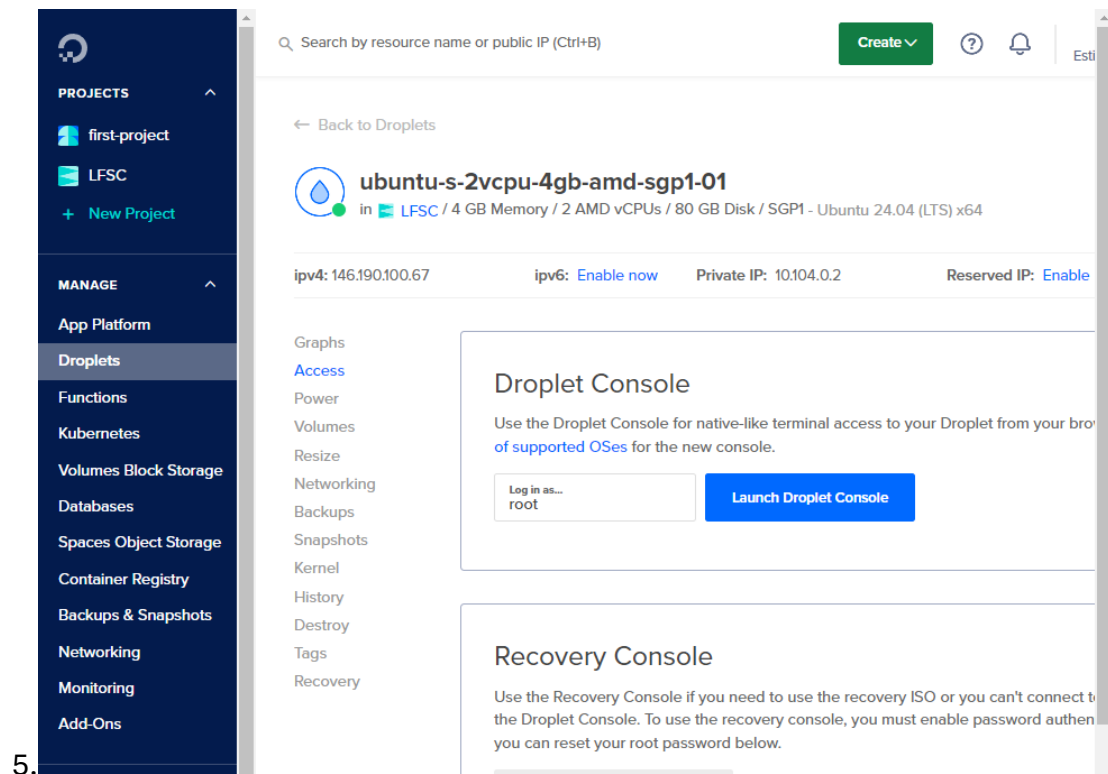
Go into the "Droplet" tab then go into the "Access" sub tab and click on "Launch Droplet Console" to configure server.

6. Update the server and install the necessary tools with the following commands:

- sudo apt-get update
- sudo apt-get install -y nginx
- sudo service nginx start
- sudo apt-cache show php

7. Install the latest php version with the following command:

- sudo apt-get install php8.3-cli php8.3-fpm php8.3-curl php8.3-gd php8.3-mysql php8.3-mbstring
- zip unzip

8. Configure nginx
- Enter the following command: nano /etc/nginx/sites-enabled/default
- Delete all the text in the file with CTRL+K and copy and paste the text below into the file:
  ##

```
# You should look at the following URL's in order to grasp a solid
understanding
# of Nginx configuration files in order to fully unleash the power of
Nginx.
# https://www.nginx.com/resources/wiki/start/
#
https://www.nginx.com/resources/wiki/start/topics/tutorials/config_pitf
alls/
# https://wiki.debian.org/Nginx/DirectoryStructure
#
# In most cases, administrators will remove this file from sites-
enabled/ and
# leave it as reference inside of sites-available where it will
continue to be
# updated by the nginx packaging team.
#
# This file will automatically load configuration files provided by
other
# applications, such as Drupal or Wordpress. These applications will be
made
# available underneath a path with that package name, such as /drupal8.
#
# Please see /usr/share/doc/nginx-doc/examples/ for more detailed
examples.
##

# Default server configuration
#
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        # SSL configuration
        #
        # listen 443 ssl default_server;
        # listen [::]:443 ssl default_server;
        #
        # Note: You should disable gzip for SSL traffic.
        # See: https://bugs.debian.org/773332
        #
        # Read up on ssl_ciphers to ensure a secure configuration.
        # See: https://bugs.debian.org/765782
        #
        # Self signed certs generated by the ssl-cert package
        # Don't use them in a production server!
        #
        # include snippets/snakeoil.conf;

        root /var/www/html/LFSC2 ;

        # Add index.php to the list if you are using PHP
        index index.php index.html index.htm index.nginx-debian.html;

        server_name lfsc.shop www.lfsc.shop;
   location / {
            # First attempt to serve request as file, then
            # as directory, then fall back to displaying a 404.
```

```
                 try_files $uri $uri/ =404;
        }

        # pass PHP scripts to FastCGI server
        #
        location ~ \.php$ {
                include snippets/fastcgi-php.conf;
        #
        #       # With php-fpm (or other unix sockets):
                fastcgi_pass unix:/run/php/php8.3-fpm.sock;
        #       # With php-cgi (or other tcp sockets):
        #       fastcgi_pass 127.0.0.1:9000;
        }

        # deny access to .htaccess files, if Apache's document root
        # concurs with nginx's one
        #
        #location ~ /\.ht {
        #       deny all;
        #}
}


# Virtual Host configuration for example.com
#
# You can move that to a different file under sites-available/ and symlink
that
# to sites-enabled/ to enable it.
#
#server {
#       listen 80;
#       listen [::]:80;
#
#       server_name example.com;
#
#       root /var/www/example.com;
#       index index.html;
#
#       location / {
#               try_files $uri $uri/ =404;
#       }
#}
```
   ▫ Save with CTRL+X

9. Configure lfsc.shop

   ▫ Enter the commands: nano /etc/nginx/sites-enabled/lfsc.shop
   ▫ Delete all the text in the file and copy and paste the text below into the file:
```
server {
        listen 80;
        server_name lfsc.shop www.lfsc.shop;

        root /var/www/html/LFSC2;
        index index.php index.html index.htm;
```

```
        location / {
        try_files $uri $uri/ /index.php$is_args$args;
        }

        location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php8.3-fpm.sock;
        }

        location ~ /\.ht {
        deny all;
        }
   }
```

10. Create a SQL database by entering the following commands:

  - sudo apt update
  - sudo apt install mysql-server
  - mysql
  - CREATE DATABASE store;
  - use store;
  -

11. Create database user profile by typing in the following:

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'adminpass';
GRANT ALL ON store.* TO 'admin'@'localhost';
```

12. Populate database.

  - Copy and paste everything from your local .sql file into the terminal
  - Exit MYSQL with the command "exit;"

13. Copy and paste the following to pull the repository from github:

  - git clone https://github.com/DevinCh33/LFSC2.git

Telegram

Downloading and Installing Telegram

1. Visit the Telegram Website:

   - Go to the [Telegram website](#).

2. Download Telegram:

   - Click on the "Get Telegram for Windows/Mac/Linux" button or select the appropriate version for your operating system.

3. Install Telegram:

   - For Windows:

     1. Open the downloaded .exe file.

     2. Follow the installation instructions by clicking "Next" and "Install".

   - For Mac:

     1. Open the downloaded .dmg file.

     2. Drag the Telegram app to the Applications folder.

   - For Linux:

     1. Follow the instructions provided on the Telegram website for your specific Linux distribution.

4. Launch Telegram:

   - Open the Telegram app from your desktop or applications menu.

5. Set Up Telegram:

   - Enter your phone number and click "Next".
   - Enter the verification code sent to your phone via SMS.
   - Complete your profile setup by entering your name and adding a profile picture if desired.
   -

Setting Up NGROK (Only applicable for localhost testing)

1. Visit the NGROK Website:

- Go to the [NGROK website](#).

2. Sign Up/Log In:

- Sign up for an NGROK account if you don't have one or log in if you already do.

3. Download NGROK:

- Go to the download page and download the version suitable for your operating system.

4. Install NGROK:

- For Windows:

  1. Unzip the downloaded file.

  2. Move the ngrok.exe file to a directory of your choice.

  3. Add the directory to your system's PATH environment variable if you want to run ngrok from any command prompt.

- For Mac/Linux:

  1. Open a terminal window.

  2. Unzip the downloaded file with unzip /path/to/ngrok.zip.

  3. Move the ngrok binary to a directory in your PATH with mv /path/to/ngrok /usr/local/bin.

5. Authenticate Your NGROK Account:

- In your terminal or command prompt, run the command provided in your NGROK dashboard to authenticate your account. It will look something like this:

  ngrok authtoken YOUR_AUTH_TOKEN

5.0 Configuration Details

a. localhost
   - Download a code editor such as [Visual Studio Code](#) (VS Code).
   - Open the repo folder in VS Code (File > Open Folder).
   - Have GitHub Desktop opened.

- You may make changes to the code as you wish.
- Review your changes with GitHub Desktop.
- If satisfied, write some details about your changes and commit.

b. Digital Ocean (Devin)
c. GoDaddy (Devin)
d. Telegram

Configuring NGROK

1. Start NGROK:

- Open a terminal or command prompt.

    - Press Win + R, type cmd, and press Enter.

- Use the cd Command to Change Directory.

    - cd path\to\directory

- Start NGROK by typing the command:

    - ngrok http 80
    - Replace 80 with the port number your local server is running on.

2. Check the NGROK Dashboard:
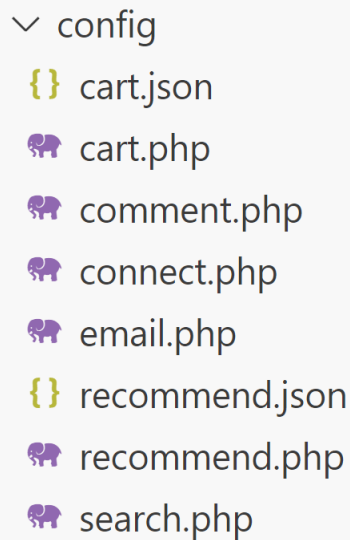
- Once NGROK is running, you will see a dashboard with the status of your tunnels.
- Note the public URL (e.g., http://abcd1234.ngrok-free.io) provided by NGROK.
- Replace the new URL with the **BOLD** URL below:
  - Customer bot webhook:
  https://api.telegram.org/bot6857843252:AAHa267pqAKyWAimgH52Vi4NYOt1FdsMu0A/setWebhook?url=https://**667f-180-73-156-227**.ngrok-free.app/{repo name}/telegram_bot.php
  - Seller bot webhook:
      https://api.telegram.org/bot6871950229:AAGGsiFvspL7UdThz8DyupfCVcYFUAfIaxw/setWebhook?url=https://**667f-180-73-156-227**.ngrok-free.app/{repo name}/seller/seller_telegram_bot.php

3. Start interacting with the Telegram bot to see if there is any response from it.

- Customer bot link: https://t.me/lfsc_buyer_bot
- Seller bot link: https://t.me/lfsc_seller_bot

e. Configuration Files
- There are a few configuration files in the system.

  ∨ config
  {} cart.json
  🐘 cart.php
  🐘 comment.php
  🐘 connect.php
  🐘 email.php
  {} recommend.json
  🐘 recommend.php
  🐘 search.php

- connect.php: Mainly includes the details (such as username, password and schema name) required to access the database using **mysqli**. Note that there is separate **connect.php** files for seller and admin as well.
- email.php: Requires an email username, password and more details about the mail server to send an email.
- search.php: $max is the maximum number of products listed for a search.
- recommend.php:
- $messageRec are the messages that are shown in the four product recommendations (refer to the system overview above).
- Products that have a price of +- range (as indicated in $price) compared to the last ordered product's price, will be recommended.
- $recommendMerchantsBasedOnRating can either be **avg**, **sum**, or **mul**. This will determine which six merchants get recommended based on their rating.
- **Avg: Only considers their average rating.** This will allow new merchants to be shown here easily, as it is likely for a new merchant to get a 5.0 rating easily, while a well-respected merchant might only have a 4.8 average rating.
- **sum**: Usually, the more ratings a merchant has, the higher its score will be. This might allow some low rating merchants to be recommended.
- **mul:** By considering the average rating and the number of ratings, this will push merchants with many positive ratings to the top. However, it is likely that the

recommended merchants list will not change much, providing less opportunity for new merchants to be recommended.

- recommend.json: Best to only put four products to guide new customers in what to search for in the search bar.
- cart.json:
- debug: For use in development, this can be set to **true** to see some debugging messages relating to the cart in the browser's console.
- enableQuantityCheck: Makes sure that the quantity of product that the customer wants to add to cart is not higher than the number available.
- cart.php:
- $exactPriceCheck: It is possible to use a browser's Inspect Element tool to change the pricing of products charged. Setting this to **true** will combat that.
- $roomForError denotes the maximum difference it can allow, or it will reject the product being added into the cart. The recommended is 0.00999, which is less than a cent difference (so no price change is allowed) but will allow for some error in the case of imperfect computing.
- $divideMinPrice: If $exactPriceCheck is disabled, the room for error is based on this value. It can be set to 999999 for no price checking.
- $divideMaxStock: Same concept of combating Inspect Element, but this time on the number of available stock. One should not be able to buy more than the number of stock that is actually available. It can also be set to 999999 for no stock checking.
- $refreshBeforeCheck: Refreshes the product list that is used for price and stock checking. Recommended to be set to **false**.
- Note: Do not use the Seller's custom prices feature if there is price checking involved.
- comment.php: $swearWords is the list of swear words that are blaklisted. If a comment contains these words, the comment will not be posted.

6.0 Security Configuration

Implementation in Code

- The provided code snippet secures access by checking for an admin session variable `$_SESSION['adm_id']`. If this variable isn't set, indicating the user isn't logged in as an admin, the script redirects them to a specified URL using `header()` to prevent further code execution. The `connect.php` file is included to establish a database connection, essential for subsequent operations. This setup ensures only

authenticated admins can access the page, enhancing security by preventing unauthorized access and protecting sensitive information. Proper session management with `session_start()` is assumed at the script's beginning.

Code:

```php
<?php

include("../../../config/connect.php"); // connection to database

if(!$_SESSION['adm_id']) {
   header('location:'.$inv_url);
}
?>
```

- The code enhances security by securely handling user input during registration or login. It retrieves the email from the POST request and hashes the password using `password_hash` with bcrypt, making passwords resistant to attacks. This ensures that even if the database is compromised, actual passwords are not easily retrievable. Additionally, sanitizing and validating the email input prevents XSS and ensures it is valid. This approach protects sensitive data and maintains the integrity of user credentials.

Code:

```php
$email = $_POST['email'];
$password = password_hash($_POST['txtPass'], PASSWORD_DEFAULT); // Default is
bcrypt
```

- To create an account, the PHP script ensures secure authentication by establishing a database connection (`connect.php`) and including `checks.php` for added security measures like input validation. Upon form submission (`$_POST['submit']`), it securely compares the hashed password (`password_verify($password, $row['password'])`) and checks email verification (`$row['email_verified']`) before granting access. Session management (`$_SESSION`) maintains user authentication and handles critical tasks such as price validation (`PricesCheck`). After successful authentication, users are redirected (`header("refresh:1;url=market.php")`) to `market.php`, enhancing security by ensuring only authenticated users can access sensitive functionalities.
- Using if-else conditions in this manner ensures that the script securely validates user credentials, manages sessions effectively, and provides appropriate feedback to users based on

their login status and verification status. This structured approach helps maintain system security and user trust while handling sensitive information securely.

Code:

```php
<?php
include("config/connect.php"); // connection to database
include("includes/checks.php"); // for PWS dictionary

if (isset($_SESSION["user_id"])) // if already logged in
{
    header("refresh:0;url=market.php"); // redirect to market.php page
}

    if (isset($_POST['submit'])) {
        $username = $_POST['username'];
        $password = $_POST['password'];

        if (!empty($_POST['submit'])) {
            // Update the query to also select the email_verified field
            $loginquery = "SELECT u_id, password, email_verified FROM users
WHERE username='$username'";
            $result = mysqli_query($db, $loginquery);
            $row = mysqli_fetch_array($result);

            // First check if the user is found and then if the email is
verified
            if ($row && password_verify($password, $row['password'])) {
                if ($row['email_verified']) {
                    $_SESSION['user_id'] = $row['u_id'];
                    $_SESSION['loginStatus'] = true;
                    $_SESSION['pricesCheck'] = new PricesCheck();
                    $_SESSION['pricesCheck']->Refresh($db);

                    header("refresh:1;url=market.php");
                } else {
                    $message = "You must verify your email to log in.";
                }
            } else {
                $message = "Invalid Username or Password!";
            }
        }
    }

    ?>
```

- To reset password, the PHP script ensures security by including `connect.php` for a secure database link. It validates the token (`$_GET['token']`) and checks its validity (`token_expiration > NOW()`) to prevent unauthorized access. Password updates (`$_POST['updatePassword']`) use `password_hash()` for secure encryption and parameterized queries (`mysqli_real_escape_string()`) to mitigate SQL injection risks.

Feedback (`$message`) informs users of successful or failed password updates, enhancing transparency. Integration of `vendor/autoload.php` implies additional security measures from external libraries. Overall, these practices secure password reset functionality, safeguarding user data and maintaining system integrity.

Code:

```php
<?php
include("config/connect.php");
require 'vendor/autoload.php';

$message = '';
$redirect = false;
if (isset($_GET['token'])) {
    $token = $_GET['token'];

    // Check if the token exists and is not expired
    $query = "SELECT * FROM users WHERE email_token = '$token' AND
token_expiration > NOW()";
    $result = mysqli_query($db, $query);

    if (mysqli_num_rows($result) > 0) {
        $user = mysqli_fetch_assoc($result);

        if (isset($_POST['updatePassword'])) {
            $newPassword = mysqli_real_escape_string($db,
$_POST['password']);
            $hashedPassword = password_hash($newPassword, PASSWORD_DEFAULT);

            $updateQuery = "UPDATE users SET password = '$hashedPassword',
email_token = NULL, token_expiration = NULL WHERE email_token = '$token'";
            if (mysqli_query($db, $updateQuery)) {
                $message = "<div class='alert alert-success'
role='alert'>Your password has been updated successfully. You will be
redirected back to the login page in 3 seconds.</div>";
                $redirect = true;
            } else {
                $message = "<div class='alert alert-danger'
role='alert'>Failed to update your password. Please try again.</div>";
            }
        }
    } else {
        $message = "<div class='alert alert-danger' role='alert'>Invalid or
expired token!</div>";
    }
} else {
    $message = "<div class='alert alert-danger' role='alert'>No token
provided!</div>";
```

```
}
          ?>
```

7.0 Troubleshooting

Explain error message:

- If a customer adds something to cart, and after a refresh or during checkout and at least one product is missing, it means the price detection system rejected a product added. This is due to the customer using Inspect Element to cheat. It is also likely that there is a genuine stock mismatch due to the system. If so, this can be remedied by increasing the $divideMaxStock. If custom prices are not working, $exactPriceCheck should be set to **false** and $divideMinPrice should be set to a much higher value. Alternatively, custom prices should be disabled.
- If a customer cannot post a comment (nothing happens), it means the swear filter blocked the comment. Try checking the $swearWords variable.

8.0 Configuration templates

Digital Ocean

1. Ensure that the connect.php in the seller folder has the following configuration:

```php
<?php
session_start(); // Temp session
error_reporting(0); // Hide undefined index errors
date_default_timezone_set("Asia/Kuala_Lumpur");

// Main connection file for both admin & front end
$servername = "localhost"; // server
$username = "admin"; // username
$password = "adminpass"; // password
$dbname = "store"; // database
$inv_url = "http://lfsc.shop/seller/";

// Create connection
$db = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$db)
{
    die("Connection failed: " . mysqli_connect_error());
}
```

2. Ensure that the connect.php in the admin folder has the following configuration:
```php
<?php
```

```
session_start(); // Temp session
error_reporting(0); // Hide undefined index errors
date_default_timezone_set("Asia/Kuala_Lumpur");

// Main connection file for both admin & front end
$servername = "localhost"; // server
$username = "admin"; // username
$password = "adminpass"; // password
$dbname = "store";   // database
$inv_url = "http://lfsc.shop/seller/";

// Create connection
$db = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$db)
{
    die("Connection failed: " . mysqli_connect_error());
}
```

3. Ensure that the InfoProduct.php in the seller folder has the following configuration:
```php
<?php
include('../connect.php');

$valid = false;
$check = false;

$formDataArray = array();
    foreach ($_POST as $key => $value) {
        $formDataArray[$key] = $value;
    }

$act = $formDataArray['act'];

$date = date("Y-m-d");
$proID = $formDataArray['proID'];

$productID = $formDataArray['proID'];
$productCode = $formDataArray['productCode'];
$productName = $formDataArray['proName'];
$productDescription = $formDataArray['proDescr'];
$productCategory = $formDataArray['proCat'];
$productStatus = $formDataArray['proStatus'];
$productStockAlert = $formDataArray['txtStock'];
$store = $formDataArray['storeID'];

// Extract quantity, weight, price and discount values from the arrays
$quantityValues = $formDataArray['quantity'];
$weightValues = $formDataArray['weight'];
$priceValues = $formDataArray['price'];
$priceNo = $formDataArray['priceNo'];
$discountValues = $formDataArray['discount'];

$uploadDirectory = "../images/product/";
$fileDirectory = "http:/lfsc.shop/seller/images/product/";
$fileName = uniqid() . '.' . pathinfo($_FILES['proImage']['name'],
PATHINFO_EXTENSION);
```

```
$imageTmpName = $_FILES['proImage']['tmp_name'];
$uploadFile = $uploadDirectory.$fileName;
move_uploaded_file($imageTmpName, $uploadFile);
$urlImage = $fileDirectory . $fileName;
```