BRAC UNIVERSITY

Inspiring Excellence

CSE370: Database Systems
Project Report
Project Title: QuickMeds - Drugstore

Group No: 1, CSE370 Lab Section: 04, Fall 2022		
ID	**Name**	**Contribution**
20101197	Abir Ahmmed Bhuiyan	Cart, PaymentGateway, Order views, Product add/update page, Project Structure
20201039	Sanjida Tasnim	Login, Registration, Home, Nav, Search
20201086	Namreen Shaiyaz	Admin dashboard, Order table, Customer table
20201098	Khandaker Maisha Tanzim	Database, UserProfile/Update Profile

CSE370_LabSection04_GroupNo01_20101197_20201039_20201086_20201098.pdf

Table of Contents

Introduction

Electronic commerce, or e-commerce, refers to the trading of commercial goods and services through the internet in an online platform. The purpose of this project is to create an online pharmacy that can be used to fulfill the medical needs of individuals in the country. It has been developed with the intention to reach people in need, and to implement a functioning database system that can work behind the project to cater to a large population.

- Ecommerce
- Online pharmaceutical
- Developed using the LAMP stack (i.e Linux Apache MariaDB PHP)

Why We Have Intended to Build This System

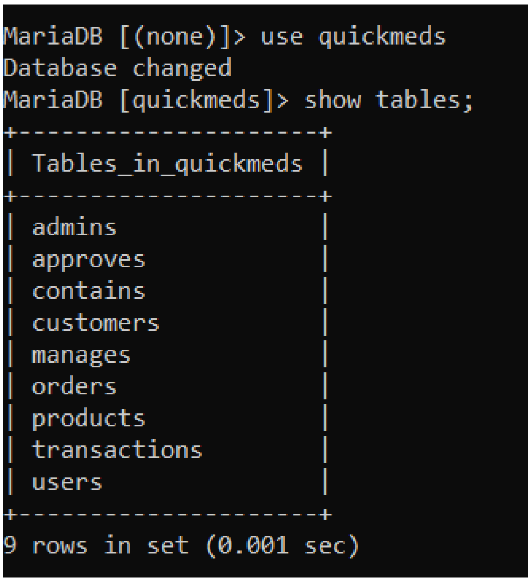
The primary intention behind this project is to understand the principles of an e-commerce platform and the implementation of the database system. However, the scope of this project goes beyond that. We have tried to develop a system that is realistic and helpful for the community with hopes for future development. Since it is not possible for many people to physically buy their necessary medications, this project aims to create a convenient system that delivers them to their doorsteps.

- How db works
- With intentions to further improve and modify
- To work on something that is realistic etc etc → that can be helpful for community

Project Description

This project uses LAMP stack. This means that it uses a software bundle that is made up of the Linux operating system; the Apache web server; MySQL/MariaDB as the database server; and PHP as the main backend programming language. Additionally, the front uses HTML, CSS, and bootstrap which is a free open-source frontend CSS library. Furthermore, we have included `php router` an open source project to give MVC architecture to our codebase and to make it facile in terms of implementation.

Now, the database behind this includes- 6 tables with 3 relations as shown below:



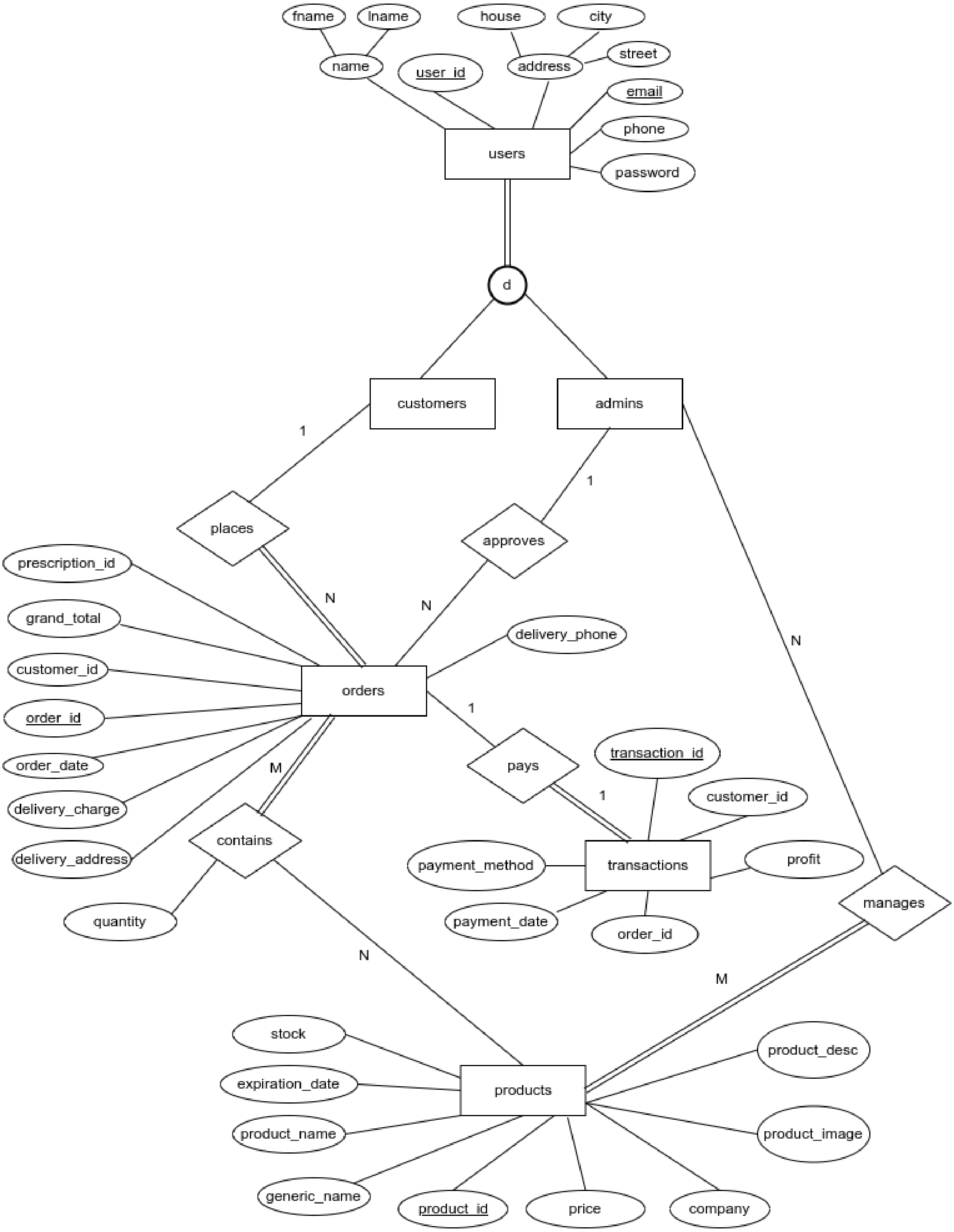
```
MariaDB [(none)]> use quickmeds
Database changed
MariaDB [quickmeds]> show tables;
+---------------------+
| Tables_in_quickmeds |
+---------------------+
| admins              |
| approves            |
| contains            |
| customers           |
| manages             |
| orders              |
| products            |
| transactions        |
| users               |
+---------------------+
9 rows in set (0.001 sec)
```

Project Features

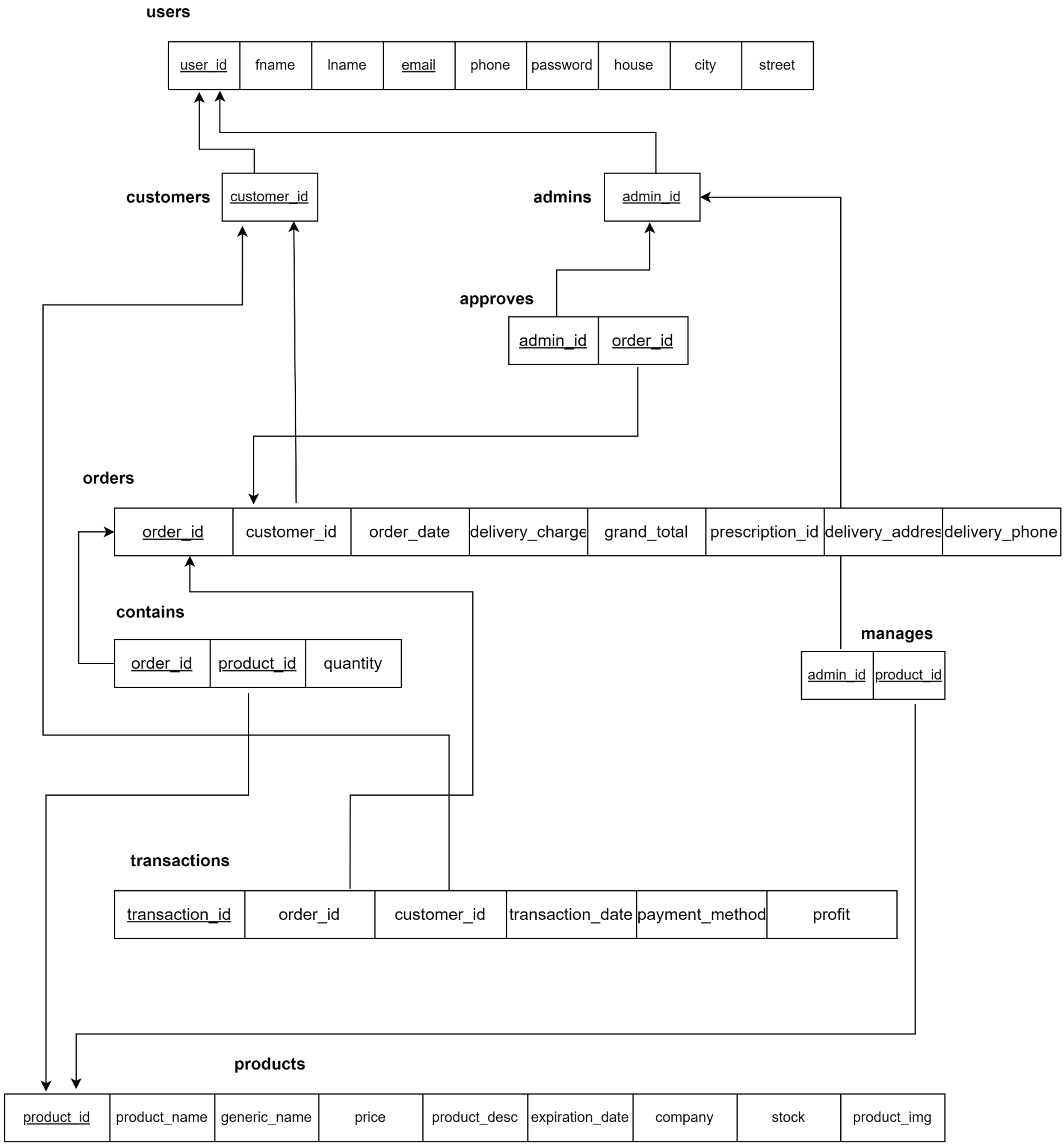
The project includes the following features to allow the project to function seamlessly:

- **Login** - to allow users and admins to access the website based on their individual roles.
- **Registration** - to allow a new user to create their own account before purchase.
- **User panel** - to provide an interface to the users, when they access the website or when they try to make a purchase.
- **Admin panel** - to allow database admin to modify the database, manage resources, and access control over users.
- **Shop** - to display products to users and to
- **Shopping cart** - to keep track of user's selected items in a user session until the order is placed.
- **Search filter** - to filter the user's or admin's required search based on specific criteria.
- **Transaction with payment gateway** - to allow users to place an order and make payment through an online portal.

ER/EER Diagram



Schema Diagram



users

user_id	fname	lname	email	phone	password	house	city	street

customers

customer_id

admins

admin_id

approves

admin_id	order_id

orders

order_id	customer_id	order_date	delivery_charge	grand_total	prescription_id	delivery_addres	delivery_phone

contains

order_id	product_id	quantity

manages

admin_id	product_id

transactions

transaction_id	order_id	customer_id	transaction_date	payment_method	profit

products

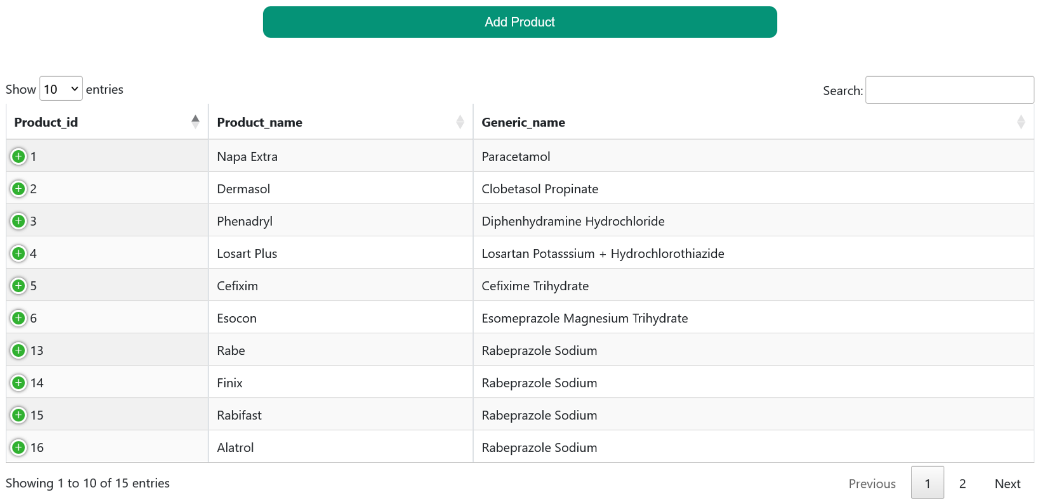
product_id	product_name	generic_name	price	product_desc	expiration_date	company	stock	product_img

Frontend Development

<Briefly discuss about Frontend Development and add Screenshots
by mentioning Individual Contributions>

Contribution of ID: 20101197, Name: Abir Ahmmed Bhuiyan

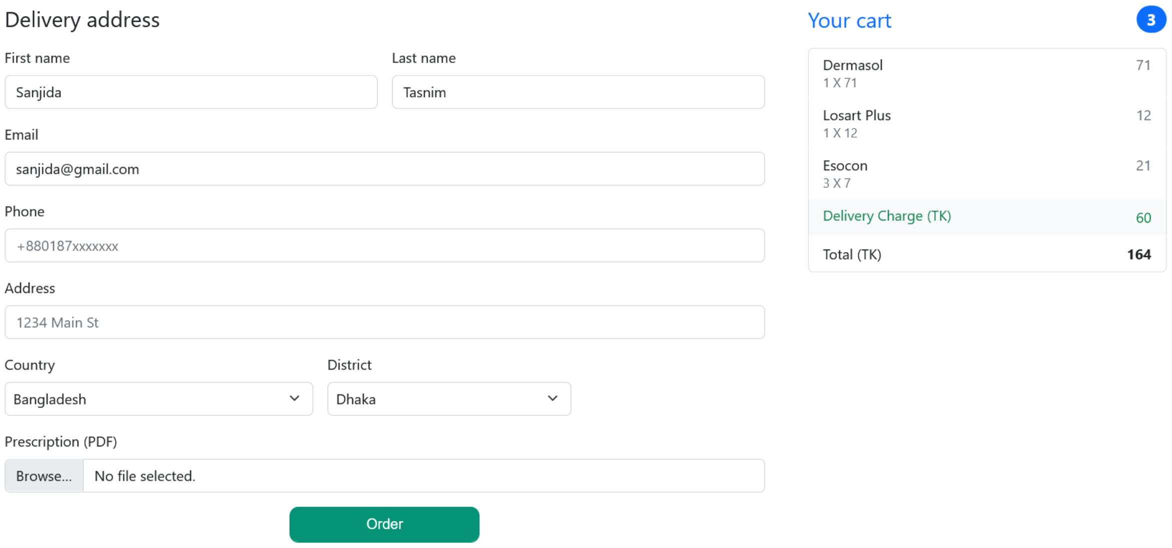
1. Product page [admin side] - Products and corresponding data are read from the database and are displayed in a table.



Product_id	Product_name	Generic_name
⊕ 1	Napa Extra	Paracetamol
⊕ 2	Dermasol	Clobetasol Propinate
⊕ 3	Phenadryl	Diphenhydramine Hydrochloride
⊕ 4	Losart Plus	Losartan Potasssium + Hydrochlorothiazide
⊕ 5	Cefixim	Cefixime Trihydrate
⊕ 6	Esocon	Esomeprazole Magnesium Trihydrate
⊕ 13	Rabe	Rabeprazole Sodium
⊕ 14	Finix	Rabeprazole Sodium
⊕ 15	Rabifast	Rabeprazole Sodium
⊕ 16	Alatrol	Rabeprazole Sodium

Showing 1 to 10 of 15 entries

2. Checkout page - billing page to show users details of their shopping cart and to take or confirm their personal information before an order is placed.



Delivery address

First name: Sanjida

Last name: Tasnim

Email: sanjida@gmail.com

Phone: +880187xxxxxxx

Address: 1234 Main St

Country: Bangladesh

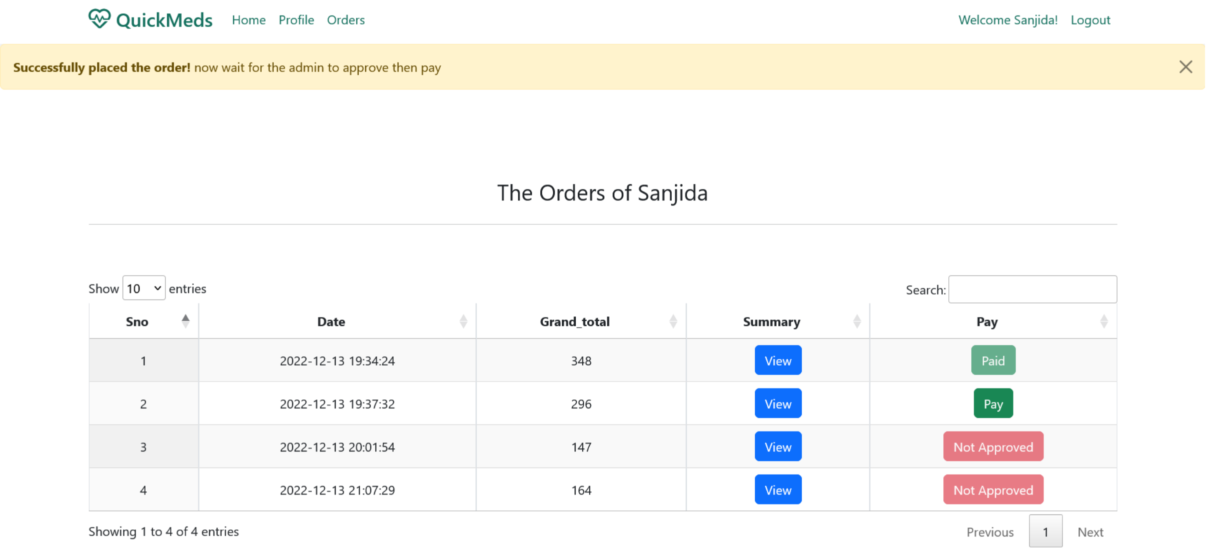
District: Dhaka

Prescription (PDF): Browse... No file selected.

Your cart 3

Dermasol 1 X 71	71
Losart Plus 1 X 12	12
Esocon 3 X 7	21
Delivery Charge (TK)	60
Total (TK)	**164**

3. Order page [user side] - used to show the details of the user's past transactions also enables the approved orders to be paid.



Successfully placed the order! now wait for the admin to approve then pay

The Orders of Sanjida

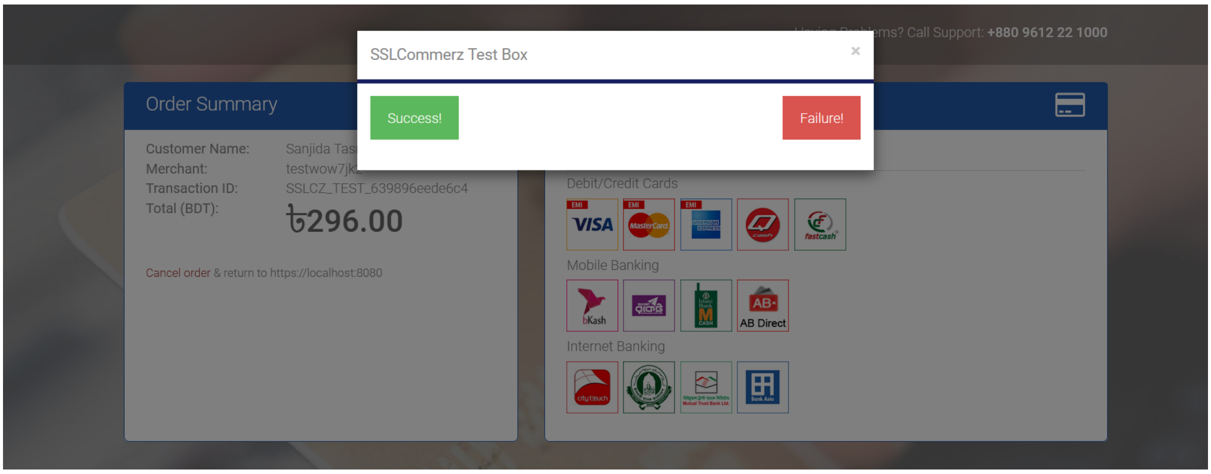
Show 10 entries

Search:

Sno	Date	Grand_total	Summary	Pay
1	2022-12-13 19:34:24	348	View	Paid
2	2022-12-13 19:37:32	296	View	Pay
3	2022-12-13 20:01:54	147	View	Not Approved
4	2022-12-13 21:07:29	164	View	Not Approved

Showing 1 to 4 of 4 entries

4. Transaction- to connect to an online payment gateway (i.e SSLCOMMERZ)



SSLCommerz Test Box

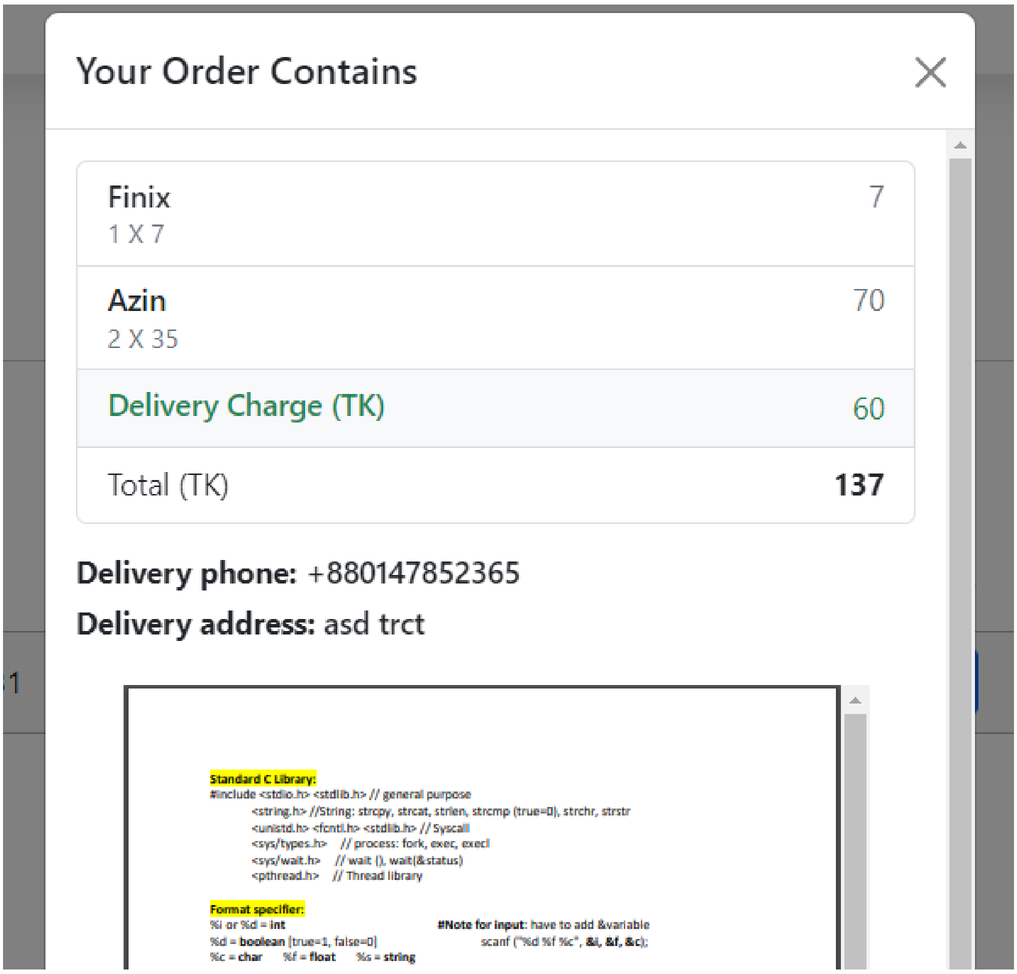
Having Problems? Call Support: **+880 9612 22 1000**

Order Summary

Customer Name: Sanjida Tas
Merchant: testwow7jk
Transaction ID: SSLCZ_TEST_639896eede6c4
Total (BDT): ৳296.00

Cancel order & return to https://localhost:8080

5. View pages [user/admin] - This is for viewing the ordered product including the prescription if it exists.



Your Order Contains

Finix 1 X 7	7
Azin 2 X 35	70
Delivery Charge (TK)	60
Total (TK)	**137**

Delivery phone: +880147852365

Delivery address: asd trct

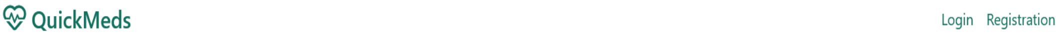
```
Standard C Library:
#include <stdio.h> <stdlib.h> // general purpose
        <string.h> //String: strcpy, strcat, strlen, strcmp (true=0), strchr, strstr
        <unistd.h> <fcntl.h> <stdlib.h> // Syscall
        <sys/types.h>    // process: fork, exec, execl
        <sys/wait.h>    // wait (), wait(&status)
        <pthread.h>    // Thread library

Format specifier:
%i or %d = int                          #Note for input: have to add &variable
%d = boolean [true=1, false=0]               scanf ("%d %f %c", &i, &f, &c);
%c = char    %f = float    %s = string
```

Contribution of ID: 20201039, Name: Sanjida Tasnim

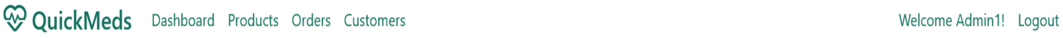
All of the following frontend features have been developed using HTML, CSS, and bootstrap.

1. Navbar - Create and customize the navbar in three different views for users with no session, admins, and logged-in users with a session of their own.

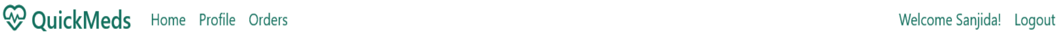
A. Default navbar - navigation view for users that are not logged in.



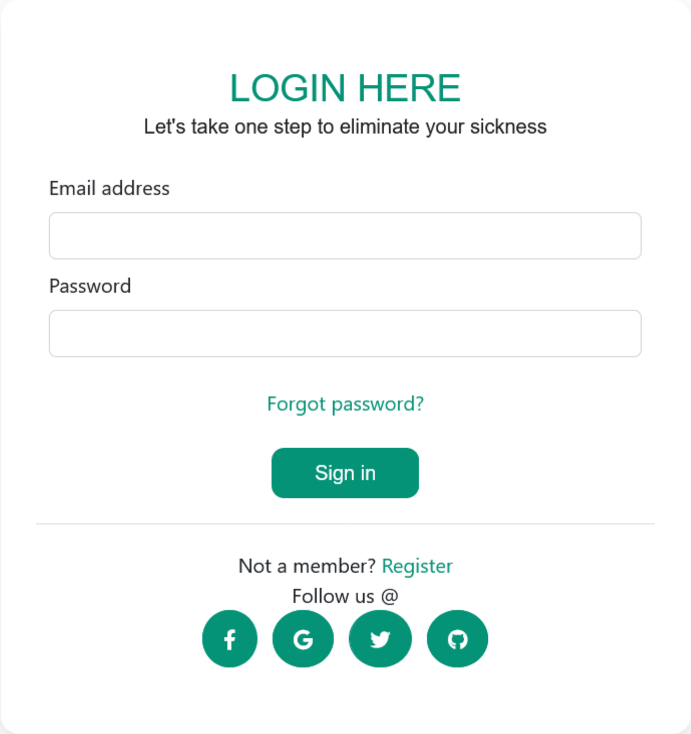
B. Admin navbar - for admins to navigate through the admin dashboard and shop.



C. User navbar - for logged-in users.



2. User/Admin Login - a login interface to provide the email and password required for logging in.



LOGIN HERE

Let's take one step to eliminate your sickness

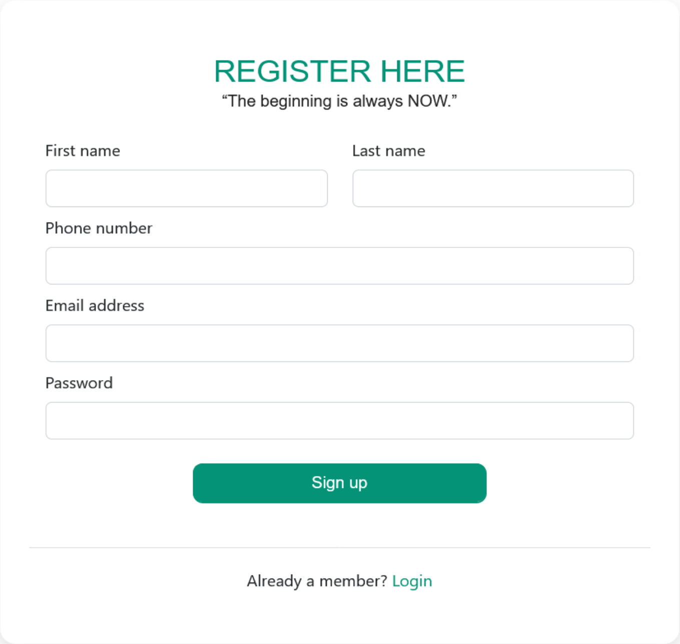
Email address

Password

Not a member? Register

Follow us @

3. Registration page - for new users to make their own accounts.



REGISTER HERE

“The beginning is always NOW.”

First name

Last name

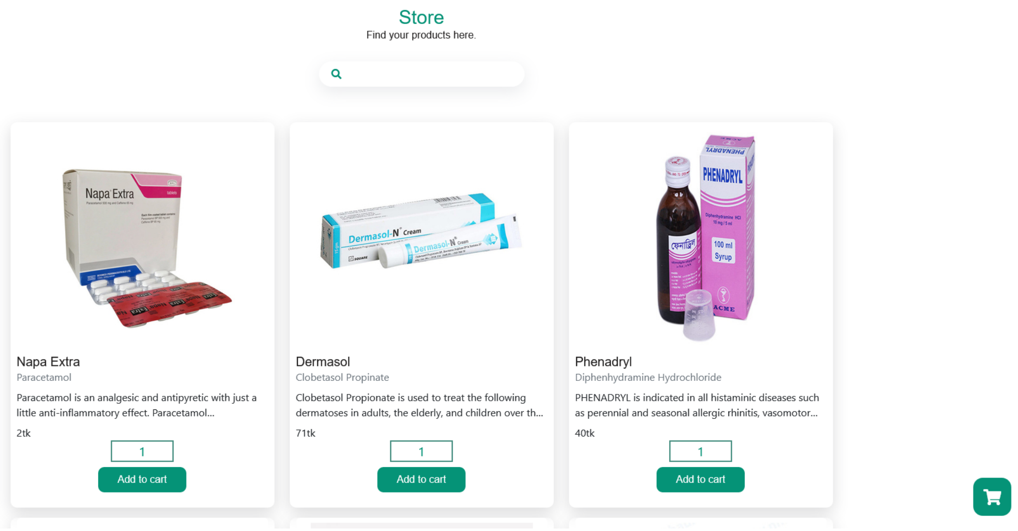
Phone number

Email address

Password

Already a member? Login

4. Home page with banner, shop display, shopping cart, and product search (Code with Amir, 2021) - for users to access.



Store

Find your products here.

Napa Extra

Paracetamol

Paracetamol is an analgesic and antipyretic with just a little anti-inflammatory effect. Paracetamol...

2tk

Dermasol

Clobetasol Propinate

Clobetasol Propionate is used to treat the following dermatoses in adults, the elderly, and children over th...

71tk

Phenadryl

Diphenhydramine Hydrochloride

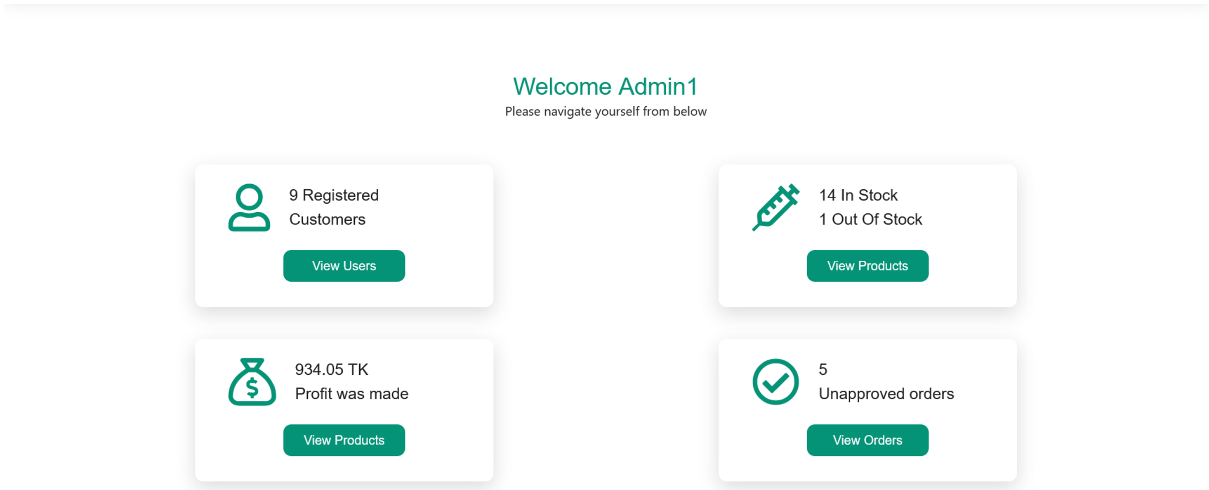
PHENADRYL is indicated in all histaminic diseases such as perennial and seasonal allergic rhinitis, vasomotor...

40tk

Contribution of ID: 20201086, Name: Namreen Shaiyaz

The frontend was implemented using a combination of HTML, CSS, and Bootstrap.

1. Admin dashboard - for the admin to get a brief summary of the most important information, and also includes buttons that correspond to the appropriate pages.



Welcome Admin1

Please navigate yourself from below

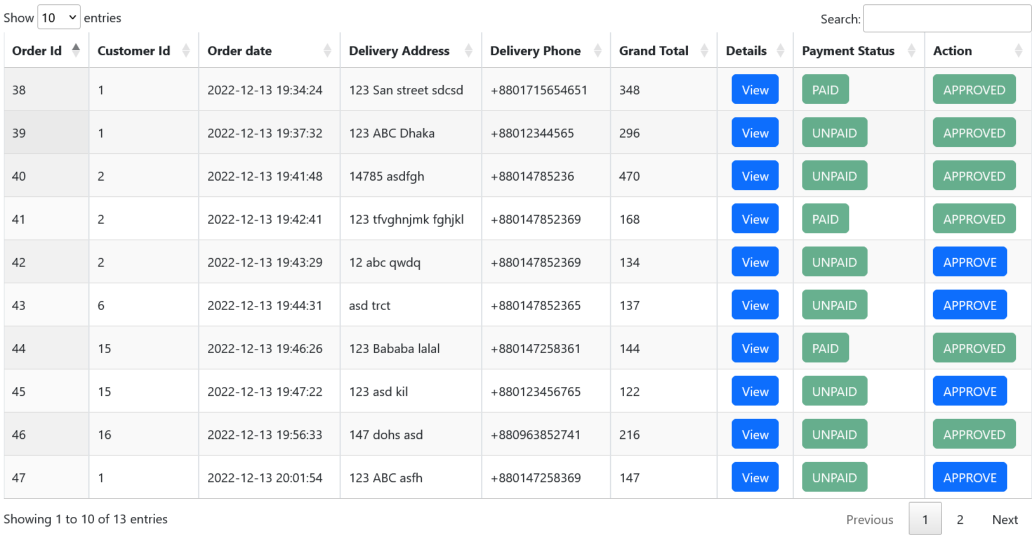
9 Registered Customers

14 In Stock
1 Out Of Stock

934.05 TK
Profit was made

5
Unapproved orders

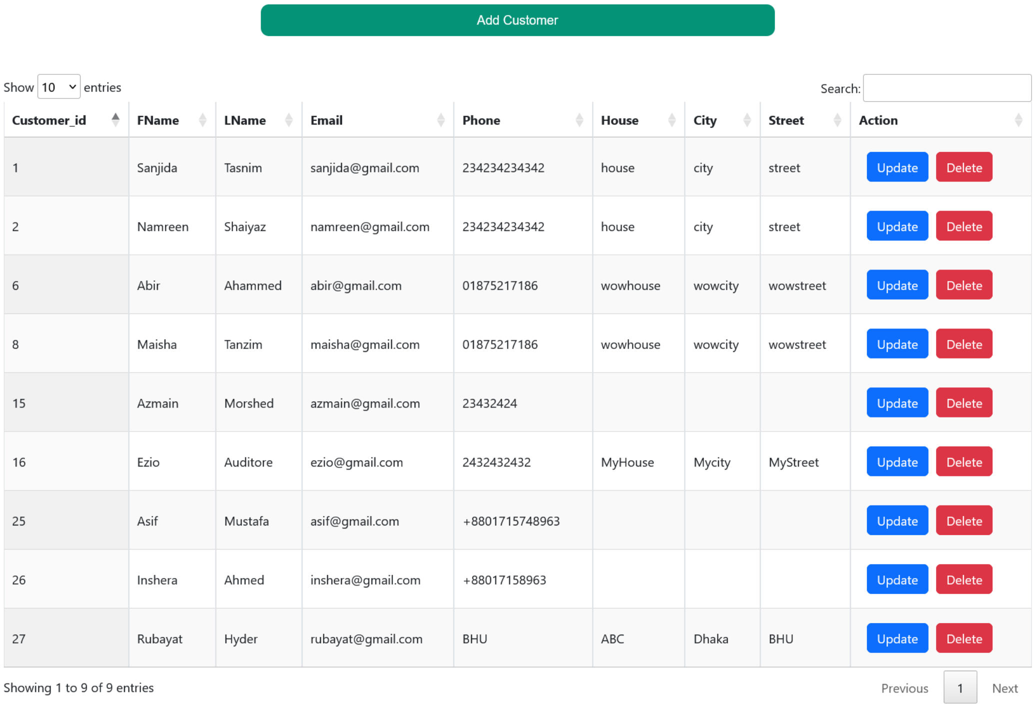
2. Order page - allows the admin to check the current and update the current state of the orders.



Order Id	Customer Id	Order date	Delivery Address	Delivery Phone	Grand Total	Details	Payment Status	Action
38	1	2022-12-13 19:34:24	123 San street sdcsd	+8801715654651	348	View	PAID	APPROVED
39	1	2022-12-13 19:37:32	123 ABC Dhaka	+88012344565	296	View	UNPAID	APPROVED
40	2	2022-12-13 19:41:48	14785 asdfgh	+88014785236	470	View	UNPAID	APPROVED
41	2	2022-12-13 19:42:41	123 tfvghnjmk fghjkl	+880147852369	168	View	PAID	APPROVED
42	2	2022-12-13 19:43:29	12 abc qwdq	+880147852369	134	View	UNPAID	APPROVE
43	6	2022-12-13 19:44:31	asd trct	+880147852365	137	View	UNPAID	APPROVE
44	15	2022-12-13 19:46:26	123 Bababa lalal	+880147258361	144	View	PAID	APPROVED
45	15	2022-12-13 19:47:22	123 asd kil	+880123456765	122	View	UNPAID	APPROVE
46	16	2022-12-13 19:56:33	147 dohs asd	+880963852741	216	View	UNPAID	APPROVED
47	1	2022-12-13 20:01:54	123 ABC asfh	+880147258369	147	View	UNPAID	APPROVE

Showing 1 to 10 of 13 entries

3. Customer page - contains a list of all the customers, along with options to add new users, update the information of existing ones, and delete them.



Customer_id	FName	LName	Email	Phone	House	City	Street	Action
1	Sanjida	Tasnim	sanjida@gmail.com	234234234342	house	city	street	Update Delete
2	Namreen	Shaiyaz	namreen@gmail.com	234234234342	house	city	street	Update Delete
6	Abir	Ahammed	abir@gmail.com	01875217186	wowhouse	wowcity	wowstreet	Update Delete
8	Maisha	Tanzim	maisha@gmail.com	01875217186	wowhouse	wowcity	wowstreet	Update Delete
15	Azmain	Morshed	azmain@gmail.com	23432424				Update Delete
16	Ezio	Auditore	ezio@gmail.com	2432432432	MyHouse	Mycity	MyStreet	Update Delete
25	Asif	Mustafa	asif@gmail.com	+8801715748963				Update Delete
26	Inshera	Ahmed	inshera@gmail.com	+88017158963				Update Delete
27	Rubayat	Hyder	rubayat@gmail.com	BHU	ABC	Dhaka	BHU	Update Delete

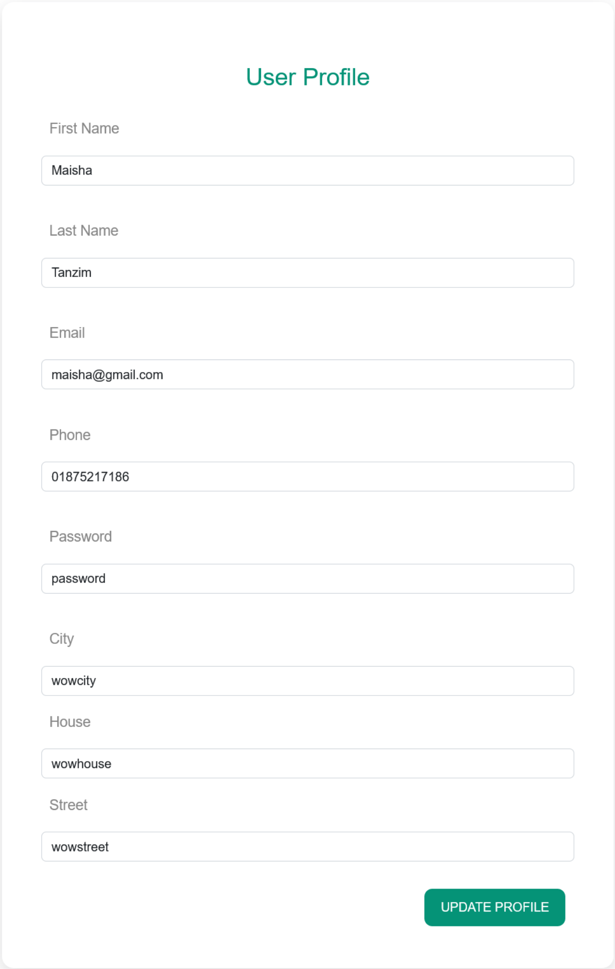
Showing 1 to 9 of 9 entries

Contribution of ID: 20201098, Name: Khandaker Maisha Tanzim

User profile page

The following was developed using HTML, CSS and Bootstrap

1. User profile page/ Update user information



User Profile

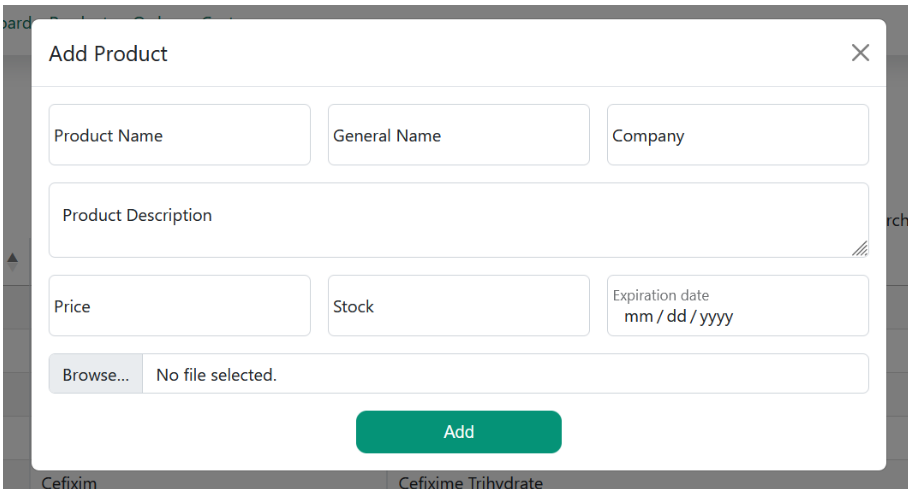
First Name	Maisha
Last Name	Tanzim
Email	maisha@gmail.com
Phone	01875217186
Password	password
City	wowcity
House	wowhouse
Street	wowstreet

Backend Development

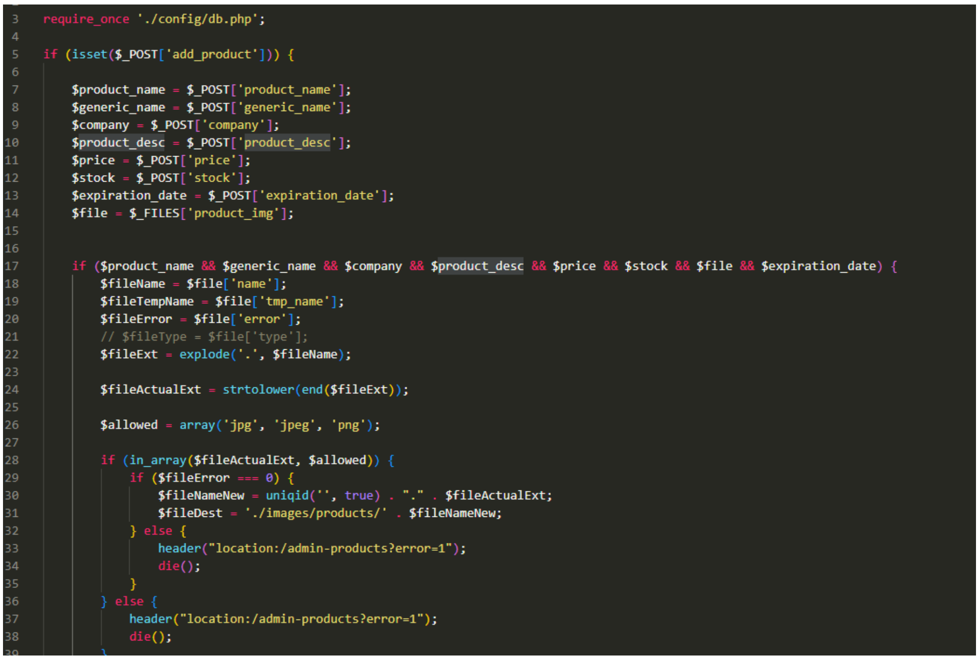
<Briefly discuss about Backend Development and add Screenshots
by mentioning Individual Contributions>

Contribution of ID: 20101197, Name: Abir Ahmmed Bhuiyan

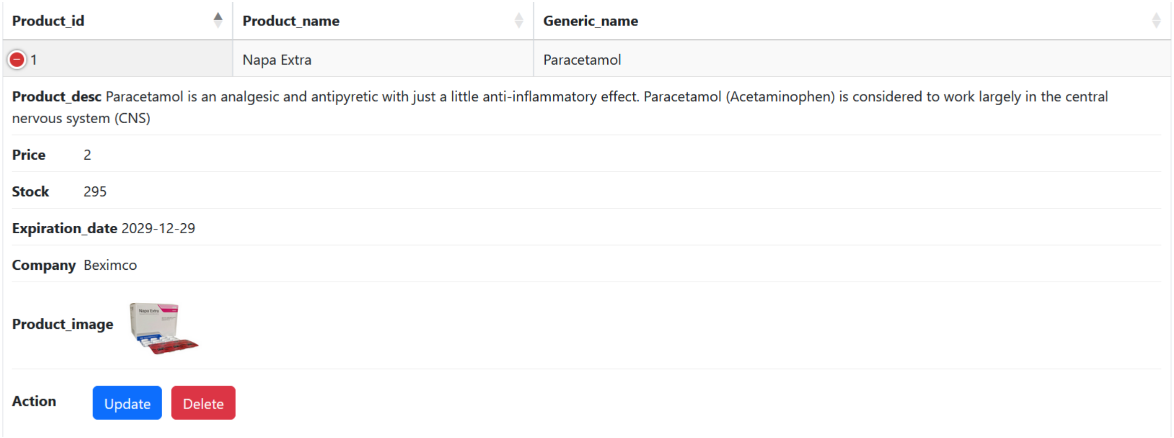
1. **Admin product page** - Add product in admin side opens a modal that asks for the product details after submitting those it goes to the /admin-add-product route as post.



There we tried to parse those information and also tried to save the included image and recorded all the information in the db. Also the same thing goes for updation and deletion of the products.



```php
require_once './config/db.php';

if (isset($_POST['add_product'])) {

    $product_name = $_POST['product_name'];
    $generic_name = $_POST['generic_name'];
    $company = $_POST['company'];
    $product_desc = $_POST['product_desc'];
    $price = $_POST['price'];
    $stock = $_POST['stock'];
    $expiration_date = $_POST['expiration_date'];
    $file = $_FILES['product_img'];


    if ($product_name && $generic_name && $company && $product_desc && $price && $stock && $file && $expiration_date) {
        $fileName = $file['name'];
        $fileTempName = $file['tmp_name'];
        $fileError = $file['error'];
        // $fileType = $file['type'];
        $fileExt = explode('.', $fileName);

        $fileActualExt = strtolower(end($fileExt));

        $allowed = array('jpg', 'jpeg', 'png');

        if (in_array($fileActualExt, $allowed)) {
            if ($fileError === 0) {
                $fileNameNew = uniqid('', true) . "." . $fileActualExt;
                $fileDest = './images/products/' . $fileNameNew;
            } else {
                header("location:/admin-products?error=1");
                die();
            }
        } else {
            header("location:/admin-products?error=1");
            die();
```



Product_id	Product_name	Generic_name
1	Napa Extra	Paracetamol

Product_desc Paracetamol is an analgesic and antipyretic with just a little anti-inflammatory effect. Paracetamol (Acetaminophen) is considered to work largely in the central nervous system (CNS)

Price 2

Stock 295

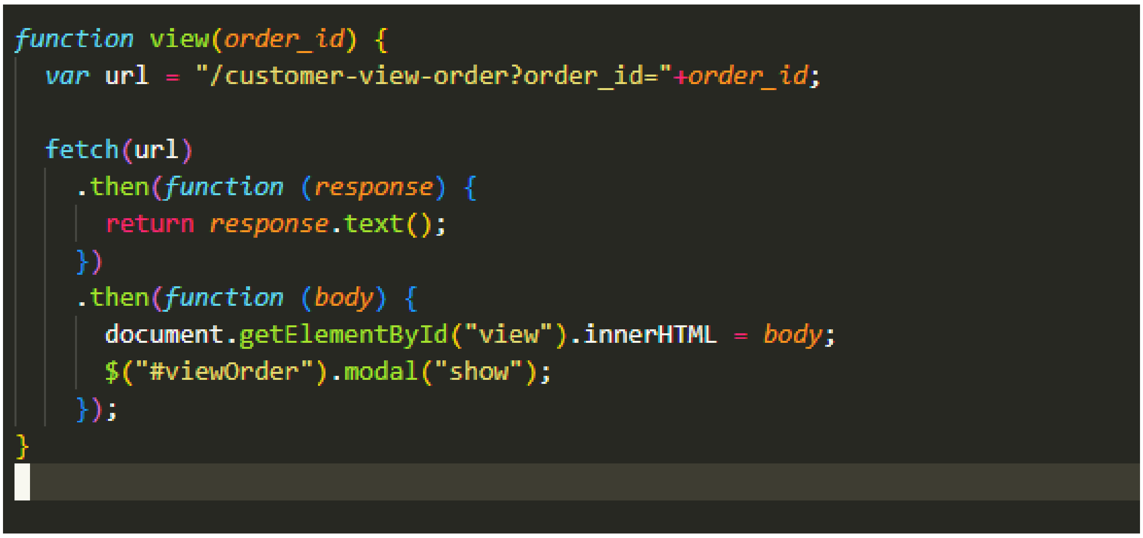
Expiration_date 2029-12-29

Company Beximco

Product_image

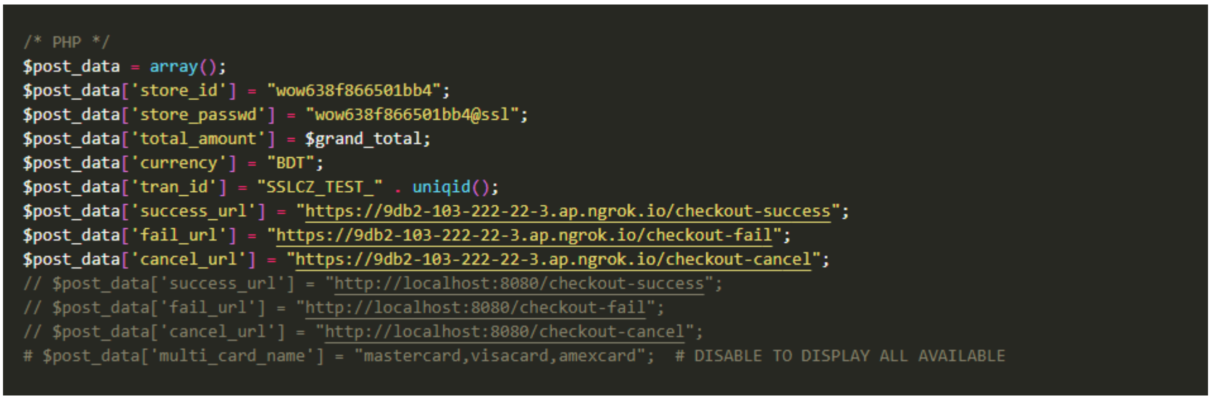
Action

2. **Display user's orders details from both admin and user side:** It was the most difficult part in my defense because I was unsure how to work it out. So, what I did I bind on-click attribute to each view button and made a function called view(id) that takes order_id as parameters and makes a fetch request to the server and the server runs the php code including the queries and sends html code and then the same function toggles the modal on.



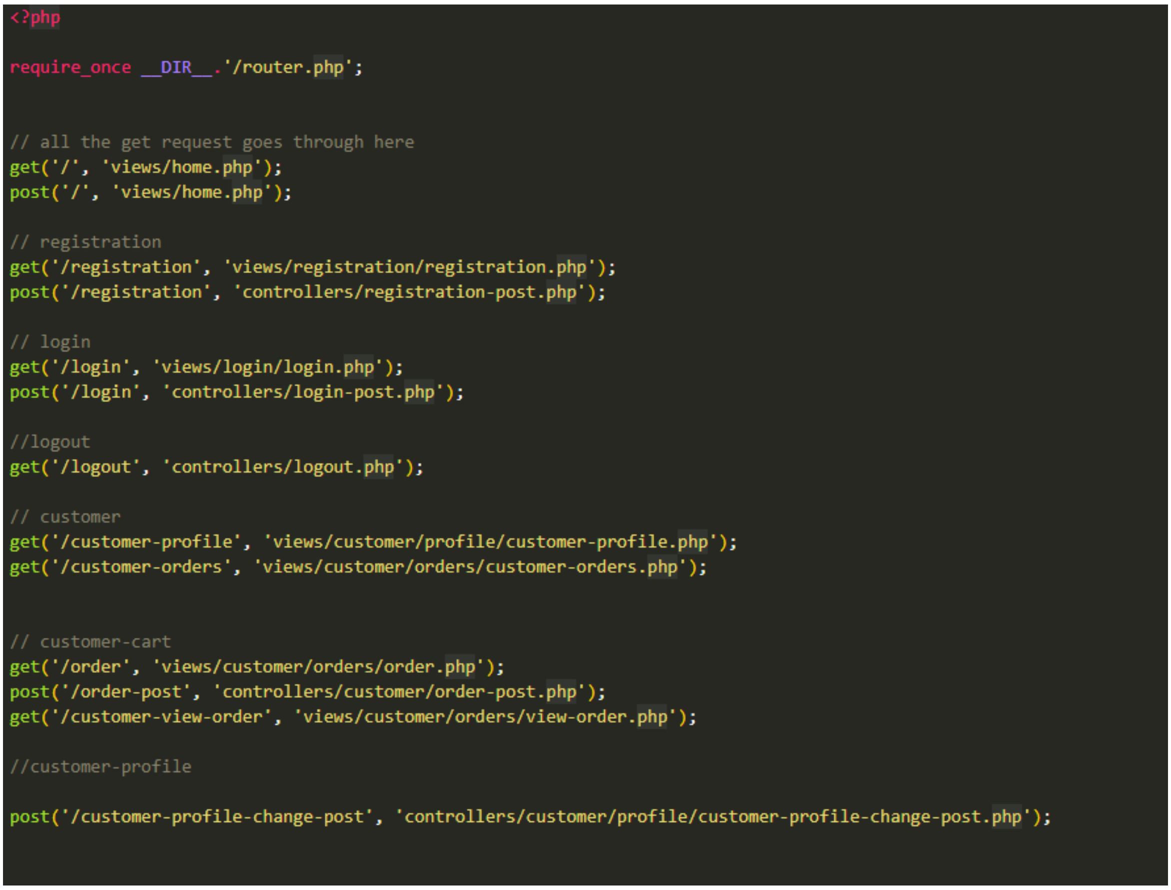
```
function view(order_id) {
    var url = "/customer-view-order?order_id="+order_id;

    fetch(url)
      .then(function (response) {
        return response.text();
      })
      .then(function (body) {
        document.getElementById("view").innerHTML = body;
        $("#viewOrder").modal("show");
      });
}
```

3. **Transaction - payment gateway integration:** It was done using SSLCOMMERZ and in this project we used a sandbox to ensure the transactions. It was done using only 4 files but the major ones are the checkout.php and the success.php files. The code has been taken from the SSLCOMMERZ documentation for developers. We had to create a sandbox account to do this job and included the credentials in the checkout.php file.



```php
/* PHP */
$post_data = array();
$post_data['store_id'] = "wow638f866501bb4";
$post_data['store_passwd'] = "wow638f866501bb4@ssl";
$post_data['total_amount'] = $grand_total;
$post_data['currency'] = "BDT";
$post_data['tran_id'] = "SSLCZ_TEST_" . uniqid();
$post_data['success_url'] = "https://9db2-103-222-22-3.ap.ngrok.io/checkout-success";
$post_data['fail_url'] = "https://9db2-103-222-22-3.ap.ngrok.io/checkout-fail";
$post_data['cancel_url'] = "https://9db2-103-222-22-3.ap.ngrok.io/checkout-cancel";
// $post_data['success_url'] = "http://localhost:8080/checkout-success";
// $post_data['fail_url'] = "http://localhost:8080/checkout-fail";
// $post_data['cancel_url'] = "http://localhost:8080/checkout-cancel";
# $post_data['multi_card_name'] = "mastercard,visacard,amexcard";  # DISABLE TO DISPLAY ALL AVAILABLE
```

4. **The code base:** We tried to follow the most efficient code base architecture also known as MVC. We did not use any frameworks instead we used an open source project called php router to ensure separation of concerns as much as possible. Moreover, we used an index.php page to index all the possible routes that exist in the project.

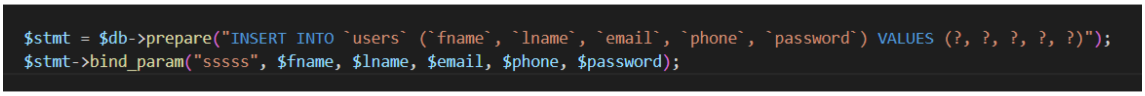


```php
<?php

require_once __DIR__.'/router.php';


// all the get request goes through here
get('/', 'views/home.php');
post('/', 'views/home.php');

// registration
get('/registration', 'views/registration/registration.php');
post('/registration', 'controllers/registration-post.php');

// login
get('/login', 'views/login/login.php');
post('/login', 'controllers/login-post.php');

//logout
get('/logout', 'controllers/logout.php');

// customer
get('/customer-profile', 'views/customer/profile/customer-profile.php');
get('/customer-orders', 'views/customer/orders/customer-orders.php');


// customer-cart
get('/order', 'views/customer/orders/order.php');
post('/order-post', 'controllers/customer/order-post.php');
get('/customer-view-order', 'views/customer/orders/view-order.php');

//customer-profile

post('/customer-profile-change-post', 'controllers/customer/profile/customer-profile-change-post.php');
```

Contribution of ID: 20201039, Name: Sanjida Tasnim

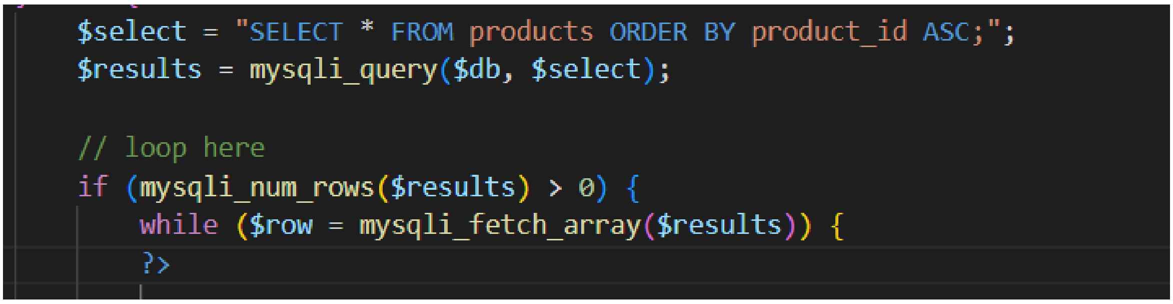
1. User login - the user's email and password are collected through a form which on submission redirects to a login-post page where the input values are checked, empty, inputs, non-registered emails, and incorrect passwords are rejected. If it passes the user validation check, the user's id and first name are put into the session to create a session under the user.

2. Registration - the user's basic information such as first name, last name, email phone, and passwords are taken on registration. Empty fields are rejected. Once all the fields are filled, submit the registration is completed by adding the provided input into the user table of the database and the user is redirected to the login page.



```
$stmt = $db->prepare("INSERT INTO `users` (`fname`, `lname`, `email`, `phone`, `password`) VALUES (?, ?, ?, ?, ?)");
$stmt->bind_param("sssss", $fname, $lname, $email, $phone, $password);
```

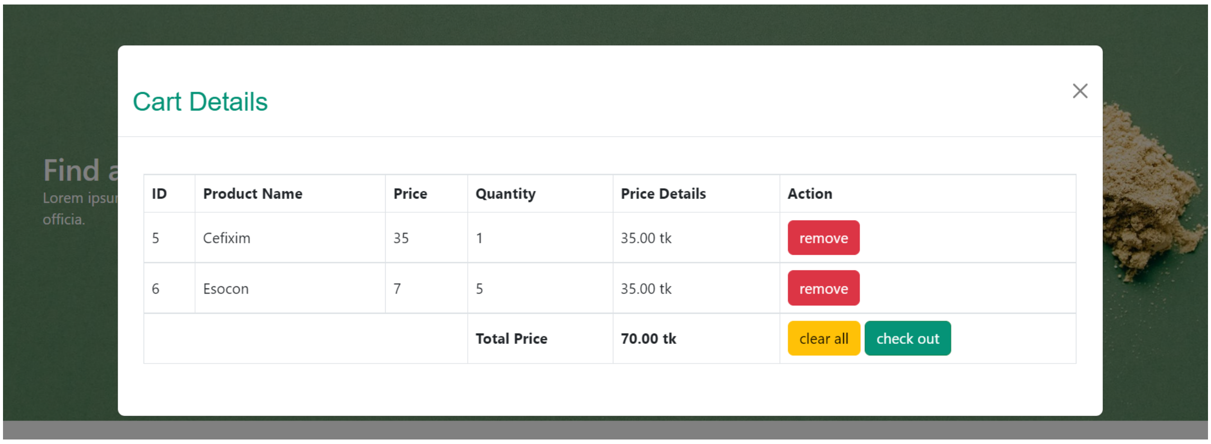
3. Shop - the products and all the necessary information about the products are taken from the product table read from and displayed on cards in the shop. Each product card includes an add-to-cart button and quantity input to place the user's desired orders (Daily Tuition, 2017).



```
$select = "SELECT * FROM products ORDER BY product_id ASC;";
$results = mysqli_query($db, $select);

// loop here
if (mysqli_num_rows($results) > 0) {
    while ($row = mysqli_fetch_array($results)) {
    ?>
```

4. Shopping cart - on click, the 'add-to-cart' button proceeds to store the product id, name, price, and quantity in the user session. These values are then used to be displayed in the cart. Other action buttons in the cart are added to
    a. Remove - deletes a single item from the cart (that is from the session)
    b. Clear all - deletes all the items in the cart (from the session)
    c. Check out - redirects the users to the checkout page for the order to be placed.

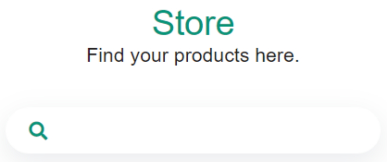
   (Programming With Dickson, 2021)

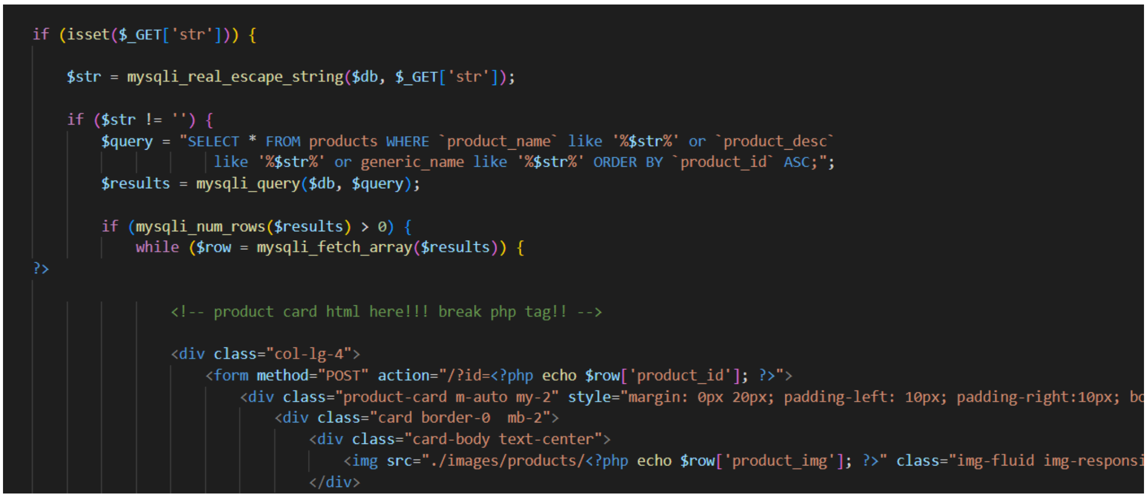


Cart Details

ID	Product Name	Price	Quantity	Price Details	Action
5	Cefixim	35	1	35.00 tk	remove
6	Esocon	7	5	35.00 tk	remove
			Total Price	70.00 tk	clear all check out

5. Search shop - is implemented using the SQL query based on user search value. The user input is compared with the product name, product description, and generic medicine name to find the most relevant search result. The retrieved values from the search are then displayed below.



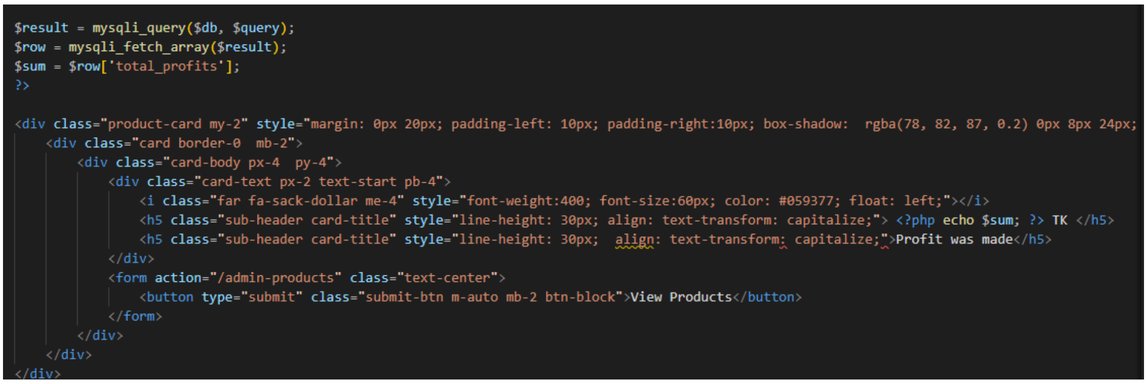


```
if (isset($_GET['str'])) {

    $str = mysqli_real_escape_string($db, $_GET['str']);

    if ($str != '') {
        $query = "SELECT * FROM products WHERE `product_name` like '%$str%' or `product_desc`
                    like '%$str%' or generic_name like '%$str%' ORDER BY `product_id` ASC;";
        $results = mysqli_query($db, $query);

        if (mysqli_num_rows($results) > 0) {
            while ($row = mysqli_fetch_array($results)) {
?>

                <!-- product card html here!!! break php tag!! -->

                <div class="col-lg-4">
                    <form method="POST" action="/?id=<?php echo $row['product_id']; ?>">
                        <div class="product-card m-auto my-2" style="margin: 0px 20px; padding-left: 10px; padding-right:10px; bo
                            <div class="card border-0  mb-2">
                                <div class="card-body text-center">
                                    <img src="./images/products/<?php echo $row['product_img']; ?>" class="img-fluid img-responsi
                                </div>
```

Contribution of ID: 20201086, Name: Namreen Shaiyaz

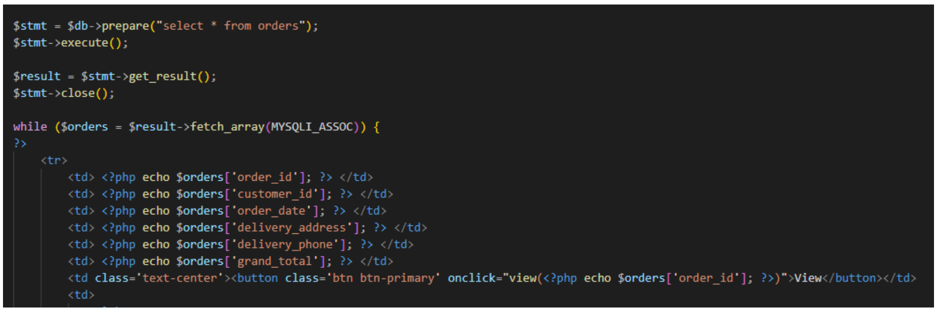
1. Admin dashboard: The dashboard has four different cards to display four different pieces of information. All the data on them has been fetched using their corresponding MySQL queries. For registered users, it is checked how many entries there are under the customers table. The number of items in stock or out of stock is found by checking if any of their stocks are 0. The profit information is calculated from the sum of all the profits from the profits table. The status of order approval is checked using

the existence of the order in the approves table - if it is not in the table, then it has not been approved, and appears on the dashboard.

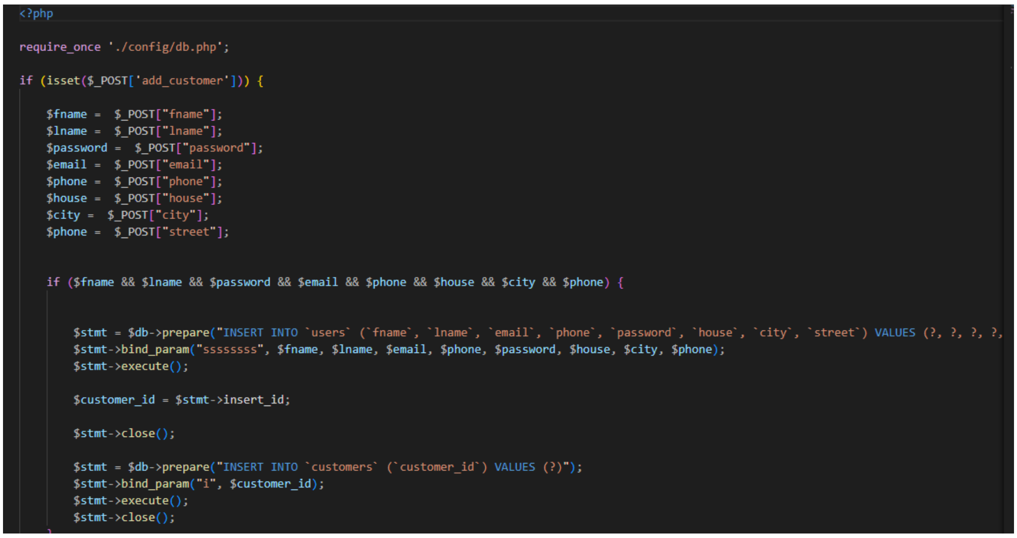


```
$result = mysqli_query($db, $query);
$row = mysqli_fetch_array($result);
$sum = $row['total_profits'];
?>

<div class="product-card my-2" style="margin: 0px 20px; padding-left: 10px; padding-right:10px; box-shadow:  rgba(78, 82, 87, 0.2) 0px 8px 24px;
    <div class="card border-0  mb-2">
        <div class="card-body px-4  py-4">
            <div class="card-text px-2 text-start pb-4">
                <i class="far fa-sack-dollar me-4" style="font-weight:400; font-size:60px; color: #059377; float: left;"></i>
                <h5 class="sub-header card-title" style="line-height: 30px; align: text-transform: capitalize;"> <?php echo $sum; ?> TK </h5>
                <h5 class="sub-header card-title" style="line-height: 30px;  align: text-transform: capitalize;">Profit was made</h5>
            </div>
            <form action="/admin-products" class="text-center">
                <button type="submit" class="submit-btn m-auto mb-2 btn-block">View Products</button>
            </form>
        </div>
    </div>
</div>
```

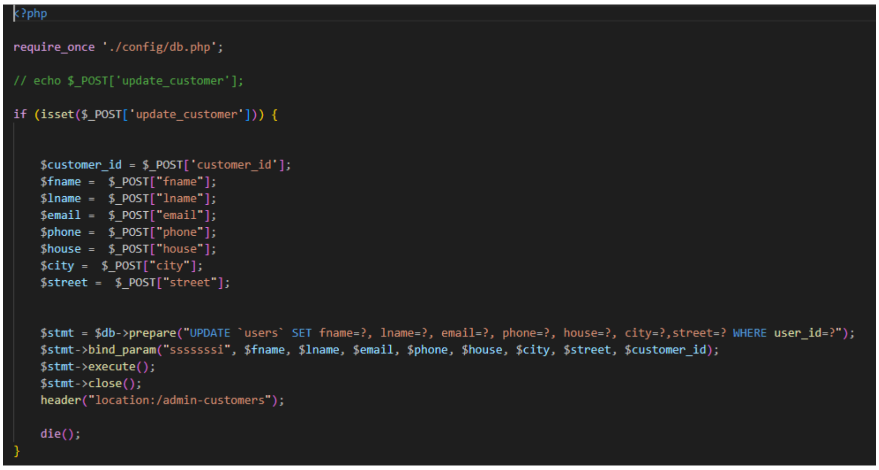
2. Customer table: The customers have been obtained by using an inner join on the customers and users tables. A loop is run through the query to display all the required details. The add user button at the top redirects to a page that allows the admin to input the necessary information, which is then added to the database. The update user button works similarly by redirecting to another page which can be used to update their individual information, and the changes are made on the database. The delete button deletes the customer directly from the user table.



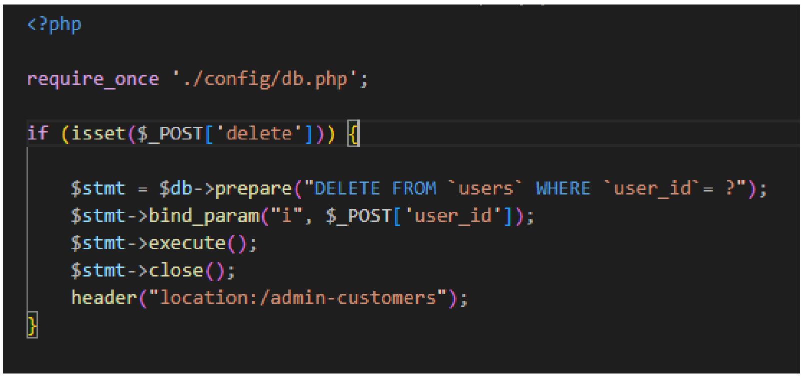
```
$stmt = $db->prepare("select * from orders");
$stmt->execute();

$result = $stmt->get_result();
$stmt->close();

while ($orders = $result->fetch_array(MYSQLI_ASSOC)) {
?>
    <tr>
        <td> <?php echo $orders['order_id']; ?> </td>
        <td> <?php echo $orders['customer_id']; ?> </td>
        <td> <?php echo $orders['order_date']; ?> </td>
        <td> <?php echo $orders['delivery_address']; ?> </td>
        <td> <?php echo $orders['delivery_phone']; ?> </td>
        <td> <?php echo $orders['grand_total']; ?> </td>
        <td class='text-center'><button class='btn btn-primary' onclick="view(<?php echo $orders['order_id']; ?>)">View</button></td>
        <td>
```



```
<?php

require_once './config/db.php';

if (isset($_POST['add_customer'])) {

    $fname =  $_POST["fname"];
    $lname =  $_POST["lname"];
    $password =  $_POST["password"];
    $email =  $_POST["email"];
    $phone =  $_POST["phone"];
    $house =  $_POST["house"];
    $city =  $_POST["city"];
    $phone =  $_POST["street"];


    if ($fname && $lname && $password && $email && $phone && $house && $city && $phone) {


        $stmt = $db->prepare("INSERT INTO `users` (`fname`, `lname`, `email`, `phone`, `password`, `house`, `city`, `street`) VALUES (?, ?, ?, ?,
        $stmt->bind_param("ssssssss", $fname, $lname, $email, $phone, $password, $house, $city, $phone);
        $stmt->execute();

        $customer_id = $stmt->insert_id;

        $stmt->close();

        $stmt = $db->prepare("INSERT INTO `customers` (`customer_id`) VALUES (?)");
        $stmt->bind_param("i", $customer_id);
        $stmt->execute();
        $stmt->close();
```

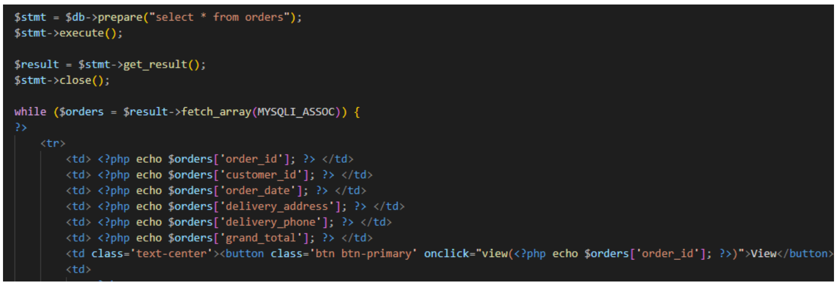


```php
<?php

require_once './config/db.php';

// echo $_POST['update_customer'];

if (isset($_POST['update_customer'])) {


    $customer_id = $_POST['customer_id'];
    $fname =  $_POST["fname"];
    $lname =  $_POST["lname"];
    $email =  $_POST["email"];
    $phone =  $_POST["phone"];
    $house =  $_POST["house"];
    $city =  $_POST["city"];
    $street =  $_POST["street"];


    $stmt = $db->prepare("UPDATE `users` SET fname=?, lname=?, email=?, phone=?, house=?, city=?,street=? WHERE user_id=?");
    $stmt->bind_param("sssssssi", $fname, $lname, $email, $phone, $house, $city, $street, $customer_id);
    $stmt->execute();
    $stmt->close();
    header("location:/admin-customers");

    die();
}
```

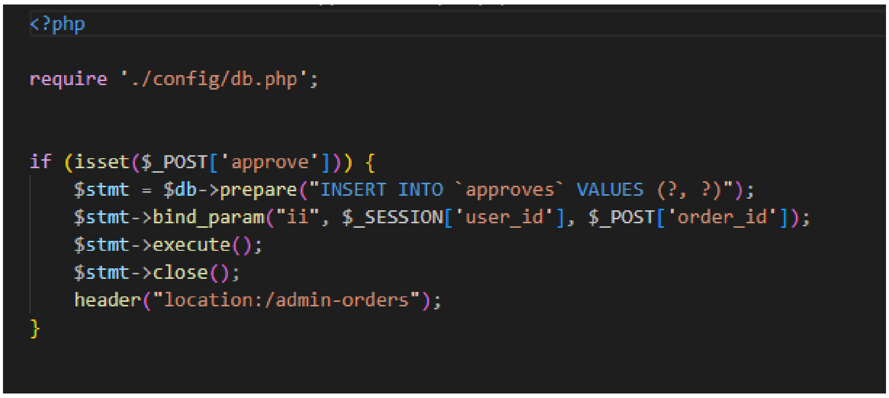


```php
<?php

require_once './config/db.php';

if (isset($_POST['delete'])) {

    $stmt = $db->prepare("DELETE FROM `users` WHERE `user_id`= ?");
    $stmt->bind_param("i", $_POST['user_id']);
    $stmt->execute();
    $stmt->close();
    header("location:/admin-customers");
}
```

3. Orders table: The orders have been obtained from the orders table, and a loop is run to display the required information. The payment status is found by checking for its existence in the transactions table. If the transaction has been completed, it shows that it has been paid, or it remains unpaid. The approved button under the action table adds the order to the approve table, so the order is verified, and it can be processed.



```php
$stmt = $db->prepare("select * from orders");
$stmt->execute();

$result = $stmt->get_result();
$stmt->close();

while ($orders = $result->fetch_array(MYSQLI_ASSOC)) {
?>
    <tr>
        <td> <?php echo $orders['order_id']; ?> </td>
        <td> <?php echo $orders['customer_id']; ?> </td>
        <td> <?php echo $orders['order_date']; ?> </td>
        <td> <?php echo $orders['delivery_address']; ?> </td>
        <td> <?php echo $orders['delivery_phone']; ?> </td>
        <td> <?php echo $orders['grand_total']; ?> </td>
        <td class='text-center'><button class='btn btn-primary' onclick="view(<?php echo $orders['order_id']; ?>)">View</button>
        <td>
```



```php
<?php

require './config/db.php';


if (isset($_POST['approve'])) {
    $stmt = $db->prepare("INSERT INTO `approves` VALUES (?, ?)");
    $stmt->bind_param("ii", $_SESSION['user_id'], $_POST['order_id']);
    $stmt->execute();
    $stmt->close();
    header("location:/admin-orders");
}
```

Contribution of ID: 20201098, Name: Khandaker Maisha Tanzim

- Database: The database has been created as per the schema provided in page 6. It includes the following tables:

    1. Users Table: contains users' basic information, along with a unique user ID which branches into separate tables for customers and admins.

    2. Customers Table: which uses user ID from Users table as foreign key to distinguish all customers.

    3. Admins Table: which uses user ID from Users table as foreign key to distinguish all admins.

    4. Products Table: contains products' details such as, name, price,available stock etc., along with a unique product ID.

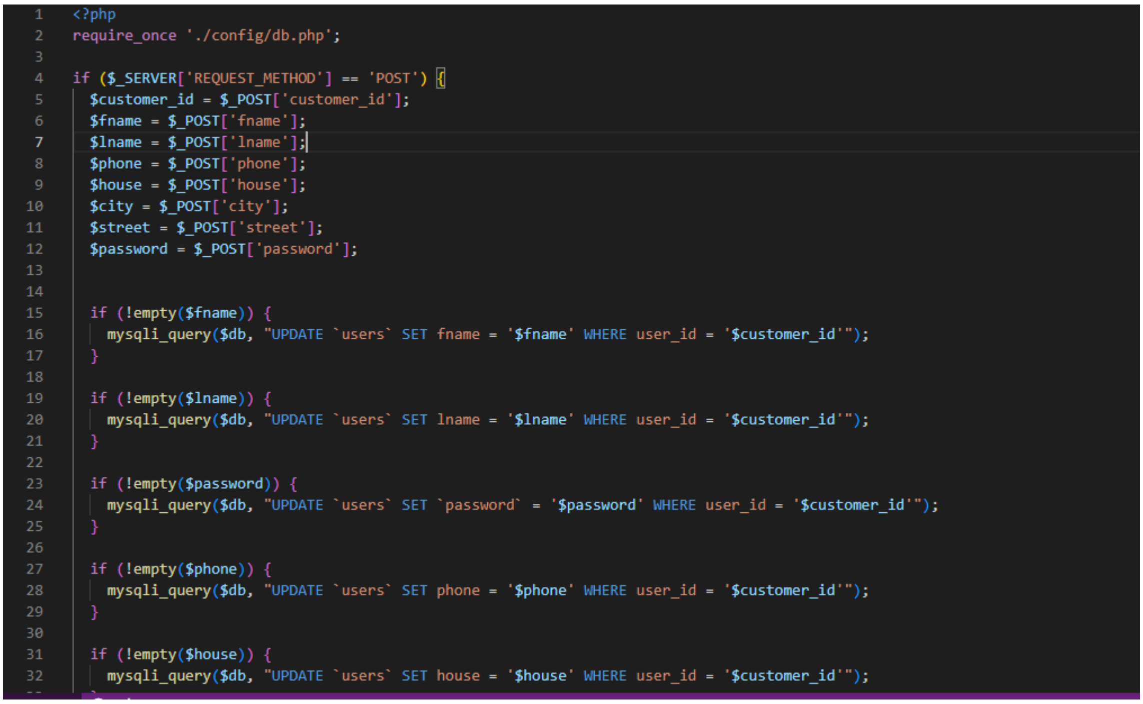
    5. Manages table: contains user ID of the admin(foreign key from admins table) and product ID of the product(foreign key from products table) which is managed/updated by the particular admin.. Using cross-referencing, order ID and product ID together are the superkey.

    6. Orders Table: contains order details such as order date, total, delivery address etc.c, along with a unique order ID. It also uses the user ID of the customer who places the order from the Customers table, as a foreign key using foreign-key approach.
    7. Contains table: contains the order ID (foreign key from orders table), product ID (foreign key from products table) and quantity of the particular product. Using cross-referencing, order ID and product ID together are the superkey.

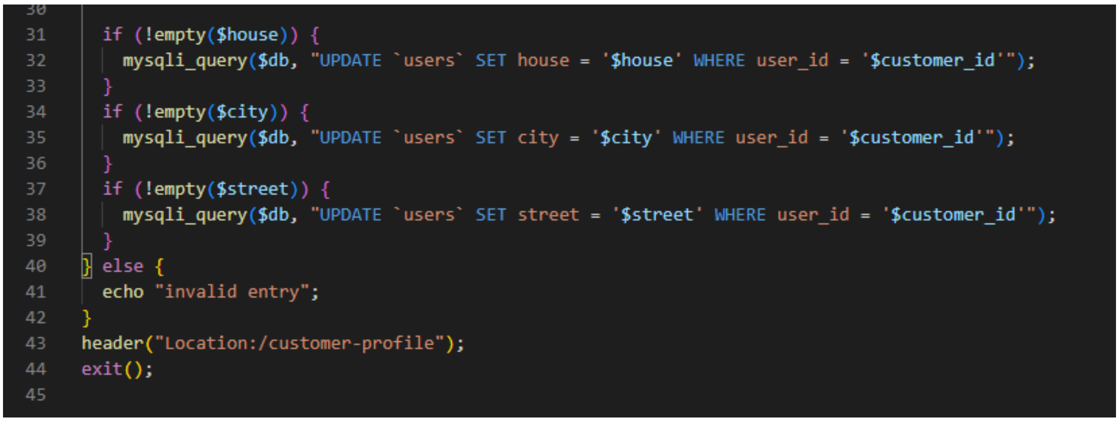
    8. Approves tables: contains user ID of admin, as admin ID, and order ID (foreign key from Admins table and orders table respectively) to show which admin approves which particular order. Using cross-referencing, order ID and admin ID together are the superkey.

    9. Transactions table: contains payment details such as transaction date, payment method etc., along with a unique transaction ID. It also uses the user ID of the customer who places the order from the Customers table and the order ID of the order from the Orders table, as a foreign keys using foreign-key approach

- User Profile: User's basic details have been collected/fetched from a login session. Details are displayed in form. An if-condition checks if any fields in the form are filled. If yes, the details are then updated into the user table (into the database) for that particular user. Updates are done using mysql query (UPDATE table_name SET column_name = new_value WHERE column_name = value_from_condition).



```php
<?php
require_once './config/db.php';

if ($_SERVER['REQUEST_METHOD'] == 'POST') {
  $customer_id = $_POST['customer_id'];
  $fname = $_POST['fname'];
  $lname = $_POST['lname'];
  $phone = $_POST['phone'];
  $house = $_POST['house'];
  $city = $_POST['city'];
  $street = $_POST['street'];
  $password = $_POST['password'];


  if (!empty($fname)) {
    mysqli_query($db, "UPDATE `users` SET fname = '$fname' WHERE user_id = '$customer_id'");
  }

  if (!empty($lname)) {
    mysqli_query($db, "UPDATE `users` SET lname = '$lname' WHERE user_id = '$customer_id'");
  }

  if (!empty($password)) {
    mysqli_query($db, "UPDATE `users` SET `password` = '$password' WHERE user_id = '$customer_id'");
  }

  if (!empty($phone)) {
    mysqli_query($db, "UPDATE `users` SET phone = '$phone' WHERE user_id = '$customer_id'");
  }

  if (!empty($house)) {
    mysqli_query($db, "UPDATE `users` SET house = '$house' WHERE user_id = '$customer_id'");
```



```php
  if (!empty($house)) {
    mysqli_query($db, "UPDATE `users` SET house = '$house' WHERE user_id = '$customer_id'");
  }
  if (!empty($city)) {
    mysqli_query($db, "UPDATE `users` SET city = '$city' WHERE user_id = '$customer_id'");
  }
  if (!empty($street)) {
    mysqli_query($db, "UPDATE `users` SET street = '$street' WHERE user_id = '$customer_id'");
  }
} else {
  echo "invalid entry";
}
header("Location:/customer-profile");
exit();

```

Future Plan

Despite the many obstacles and struggles faced to develop the project, we have hopes to further improve it. We would like to increase the versatility of the existing features such as - improving the search option by making a search filter that can sort the products based on use, need, price, popularity, brand, and more.

Additionally, new and innovative features could be added. Such as-

- AI chat bot to help users find the product they are looking for
- Prescription verification
- Promo Code and discount options
- Cash On Delivery option
- Separate form/page for password change (More secure)

Furthermore, in its current state the model is not capable of handling more realistic options needed for business development. Hence, the system would benefit from a rating method or comment option so that users can share their concerns with admins. Additionally, the current state of the inventory management system is not in the most optimal state.

Moreover, we can develop more on the existing system like,

- Making the project more realistic
- Enhanced security system
- Real payment gateway
- Enriched exception handling
- Integration of business model

Conclusion

Medical services are crucial to ensure the well-being of a community. However, it is not very accessible yet for many people. Hopefully this project will be able to bridge this gap and allow people to purchase their medicines with ease. In case of the business owner(s), this project will help to keep track of inventory, profit and customer demands. Although more improvements can be applied, an efficient database is faster to implement and also it requires less amount of time and finance to set up which provides both business owners and customers easy usage and accessibility.

References

Code with Amir. (2021, Jul 9). How to create Animated Search Bar using HTML and CSS | Website Search Box. https://www.youtube.com/watch?v=AmdIfgxMqY8

Programming With Dickson. (2021, Feb 16). How To Create Simple Shopping Cart Using PHP And MySQL https://www.youtube.com/watch?v=YloyMFPJyV4&t=181s

Daily Tuition. (2017, Dec 1). Create Shopping Cart Using PHP and Mysql https://www.youtube.com/watch?v=IO5ezsURqyg&t=1091s