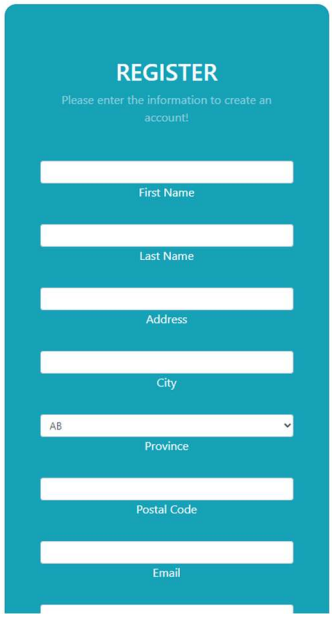
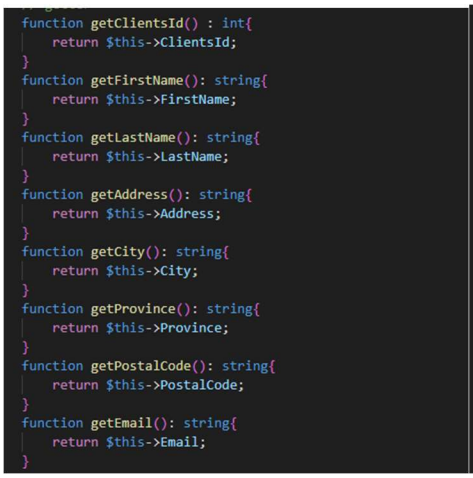
Project Luiz Attias and Oliver Franco

1. Project and entities explained:

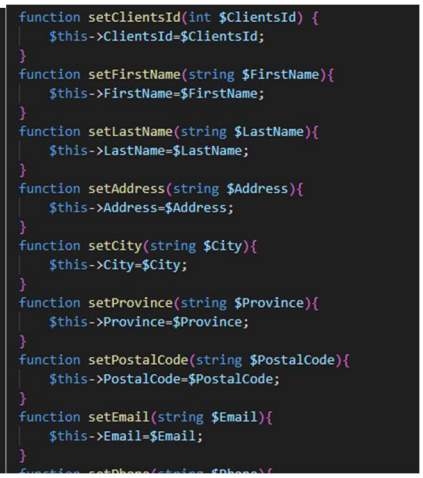
1st entity: Starting with the entities available in our project, Firstly, we have the Clients class entity which is responsible for dealing with the clients that are newly registered to our website.



When the new client chooses to register for our website, the class Clients.class.php will be responsible for reading the existing fields and also setting the new fields when a new user chooses to register. We can see how it is handled in the upcoming images with the getter and setter functions.

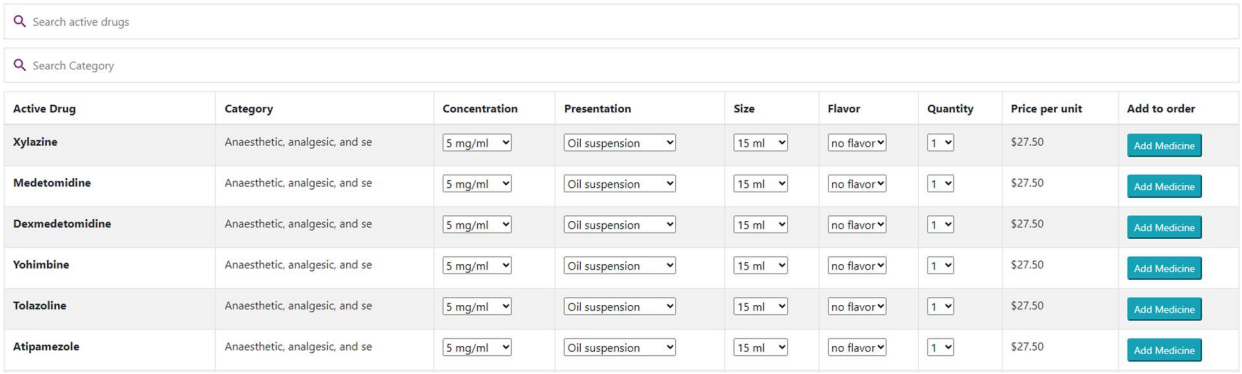


```php
function getClientsId() : int{
    return $this->ClientsId;
}
function getFirstName(): string{
    return $this->FirstName;
}
function getLastName(): string{
    return $this->LastName;
}
function getAddress(): string{
    return $this->Address;
}
function getCity(): string{
    return $this->City;
}
function getProvince(): string{
    return $this->Province;
}
function getPostalCode(): string{
    return $this->PostalCode;
}
function getEmail(): string{
    return $this->Email;
}
```



```php
function setClientsId(int $ClientsId) {
    $this->ClientsId=$ClientsId;
}
function setFirstName(string $FirstName){
    $this->FirstName=$FirstName;
}
function setLastName(string $LastName){
    $this->LastName=$LastName;
}
function setAddress(string $Address){
    $this->Address=$Address;
}
function setCity(string $City){
    $this->City=$City;
}
function setProvince(string $Province){
    $this->Province=$Province;
}
function setPostalCode(string $PostalCode){
    $this->PostalCode=$PostalCode;
}
function setEmail(string $Email){
    $this->Email=$Email;
}
```

2nd entity: Unlike the previous entity where setters were required for new entries, The class medicine.class.php only counts with the getters, since it only requires to reading the medicines that are already available in the database and display them in a table.




```php
class Medicine{

    // attributes
    private $MedicineID = 0;
    private $ActiveDrug ="";
    private $Category= "";
    private $UnitPrice= "";

    // getter
    function getMedicineId() : int{
        return $this->MedicineID;
    }
    function getActiveDrug(): string{
        return $this->ActiveDrug;
    }
    function getCategory(): string{
        return $this->Category;
    }
    function getUnitPrice(): string{
        return $this->UnitPrice;
    }
}

?>
```

3rd and 4th entities: These entities classes order.class.php and ordered_meds.class.php are for dealing with when the user wants to order their medicine, and for this matter, setters are also needed to be implemented since new orders will be placed, therefore new values to be set.

At the Image down below, the ordered_meds.class connects and is responsible to get the total price based on the quantity and size selected. The order.class is responsible for what we are able to see below the table, Where subtotal is calculated based on the GST taxes.

Order # 10001 confirmation:

#	Active Drug	Concentration	Presentation	Size	Flavor	Quantity	Total Price
1	Xylazine	15mg/ml	oil suspension	100ml	chicken	1	$50.50
2	Tolazoline	25mg/ml	ointment	30ml	no flavor	3	$27.85

Subtotal: **$78.35**
PST 0%: **$0.00**
GST 5%: **$3.92**
Total: **$82.27**
***Medications exempt PST in BC**

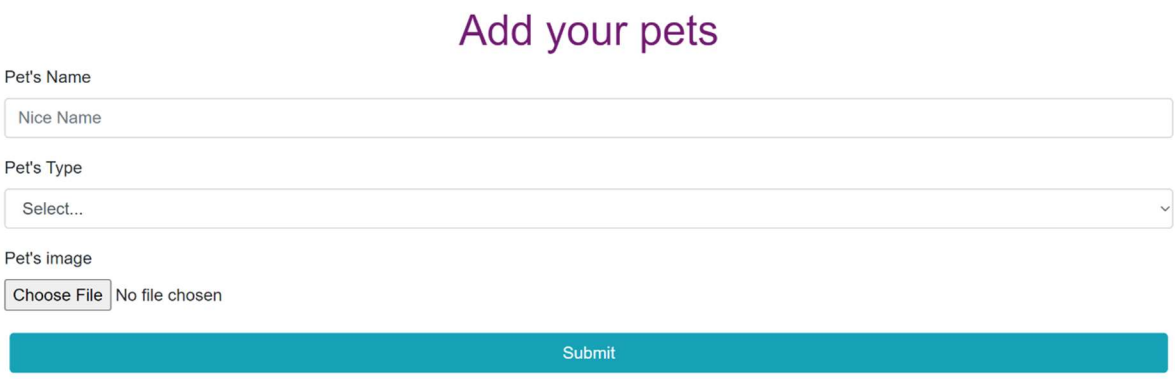
*Add more products from the table below, confirm your order or cancel.

CancelOrder ConfirmOrder

5th entity: The page class is responsible to deal with all the user interface in our website, All the html pages found in our website are implemented in one page class for easier access and management.

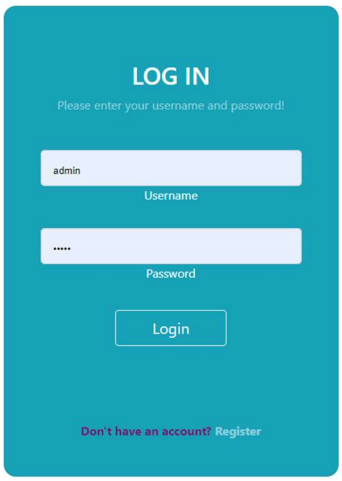
```
class Page  {

    public static $heading = "Veterinary ";
    public static $studentID = "";
    public static $studentName ="Luiz and Oliver";
    public static $member;
    public static $concentrations=["5 mg/ml","10 mg/ml","15 mg/ml","20 mg/ml", "25
    public static $presentations=["Oil suspension","Liquid Aqueous","In Lipoderm Cr
    public static $sizes=["15 ml","30 ml","60 ml","100 ml", "120 ml", "150 ml", "25
    public static $flavors=["no flavor","chicken","bacon","beef", "fish", "salmon",



    static function displayHeader() {
        ?>
                    <!DOCTYPE html>
                <html lang="en">
                <head>
                  <meta charset="UTF-8">
                  <meta name="viewport" content="width=device-width, initial-scale=1.0
                  <title>Pharma-Vet</title>
                  <!-- Add Bootstrap CSS -->
                  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstra
                  <link href="css/medsTableStyles.css" rel="stylesheet">

                </head>
                <body>
                  <!-- Navbar -->
                  <nav class="navbar navbar-expand-md">
                    <a class="navbar-brand" href="#">
```

6th Entity: The pet class is responsible to deal with the pets that the clients are adding, The name of the pet, the type as well as the image.

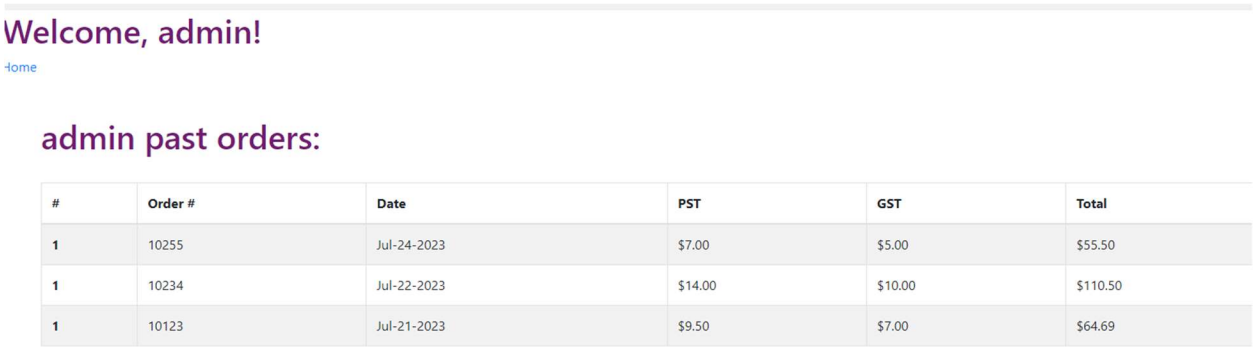


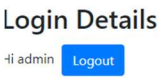

```php
// attributes
private $PetId = 0;
private $Name ="";
private $Type= "";
private $PetPicture= "";
private $Clients_Id= "";

// getter
function getPetId() : int{
    return $this->PetId;
}
function getName(): string{
    return $this->Name;
}
function getType(): string{
    return $this->Type;
}
function getPetPicture(): string{
    return $this->PetPicture;
}
function getClients_Id(): string{
    return $this->Clients_Id;
}
```

8th Entity: User class entity is responsible for the authentication process. The login page. For the administrator's username and password it is "**admin**" for both.

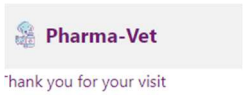


When the user logs in, It is necessary that both the username and password matches the information stored in our database. Whether for the administrative purposes, or new clients that chose to register before. If these conditions are met, When the login Button is clicked, the user will be directed to their profile page, Having the option to see their pets, add new medicines as well as logging out of the session.





And when the logout button is clicked, A message will be displayed and confirming the user that they are logged out of their session.



Group Members: Oliver Franco and Luiz attias

Oliver Franco: Medicine.class.php, Order.class.php, Ordered_Meds.class.php, Page.class.php, VeterinaryPharmacy.sql,ProjectMain.php

List of features I implemented:

Page.class.php
VeterynaryPharmacy.sql
PlaceOrder.php
ListOrders.php
AddPet.php
UserProfile.php

Beside the features implemented in php the logic of the system, and views were implemented. Page.class.php adds value by controlling all the views of the system, called from the controllers and showing all the information retrieved by the controller. PlaceOrder.php is the core of the system, where the user can see the list of the available medicines, can filter with the search bars and can select the options for each the medicine they want to add to the order. The price calculation was made in the Ordered_Meds.class by getting the values selected by the user and generating a price factor that helps calculate the price of each item added to an order. After selecting an item the user can either confirm the order or cancel the order.

In the ListOrders.php we can retrieve the users list of orders they have placed, there they can see the details of each order and also they can edit each order by adding more products, or deleting the order. In the addPet.php users can add pets and also an image from their pets which creates an affinity with the user. Finally in the UserProfile.php The statistics and list of added pets was added. In these two features the info of the users orders and pets is displayed.

Luiz

Attias: Clients.class.php,register.class.php,User.class.php,config.inc.php,ProjectMainBackup.php,Register.php,userProfile.php,userLogout.php,Page.class.php,ValidateRegisterForm.php

List of features I implemented:
-ValidateRegisterForm
-userProfile.php
-userLogout.php
-ProjectMainBackup.php

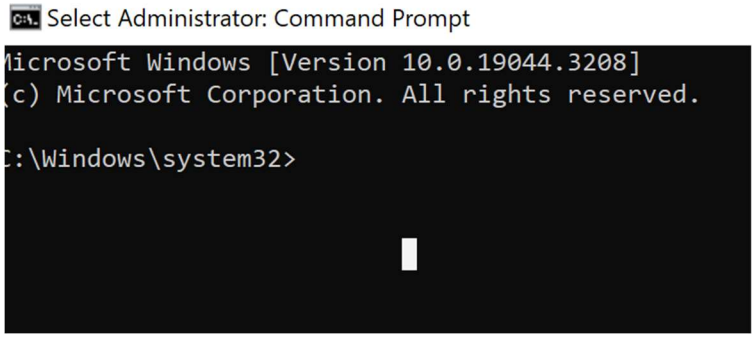
These features add values to the project, such that there the first page(ProjectMainBackup.php) where the user will see the login and register button and therefore they will be able to progress further. In this file, I make sure that neither the username, nor the password are empty so the user will make sure to input a proper data into the fields. It also uses the function verify login in order to make sure that the login is valid and the user can be directed to the userProfile.php page.

In the userProfile.php page the user will keep their own logged in session in order to see the contents related to that specific user logged in. Having the option of a button that allows that specific user to log out of their session and be directed to the userLogout.php page, where the Thank you message will be displayed.

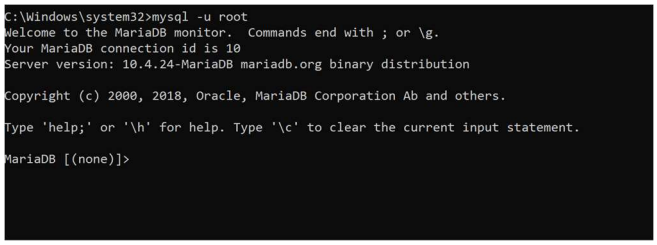
The ValidateRegisterForm.php is a very important page that has all the validations for when a new user is signing up for the website. The fields of the register form need input validation to make sure that the data sent to the database are correct and not missing any requirements from our website. If there are errors containing while the new user is signing up, A box with those errors will be displayed to best assist the user on how they need to update their information in order to be correct and therefore, the system allow their entries.

Installation manual

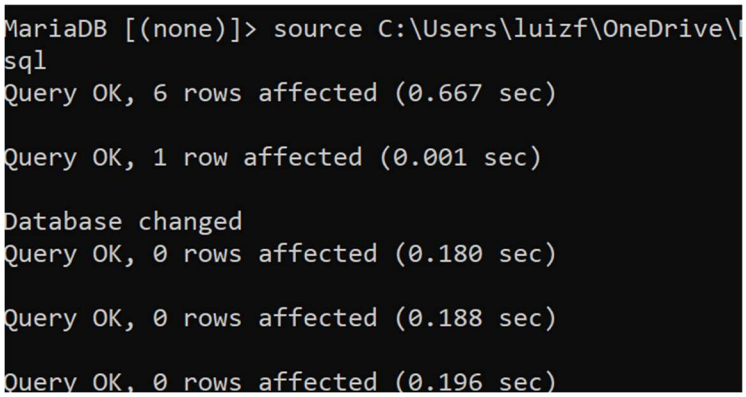
Run command prompt as administrator:



Run the command: “mysql -u root” in oder to be connected to MariaDB



Run the command “source” followed by the path of the SQL in order to create the tables that we have. Make sure to see the rows affected.



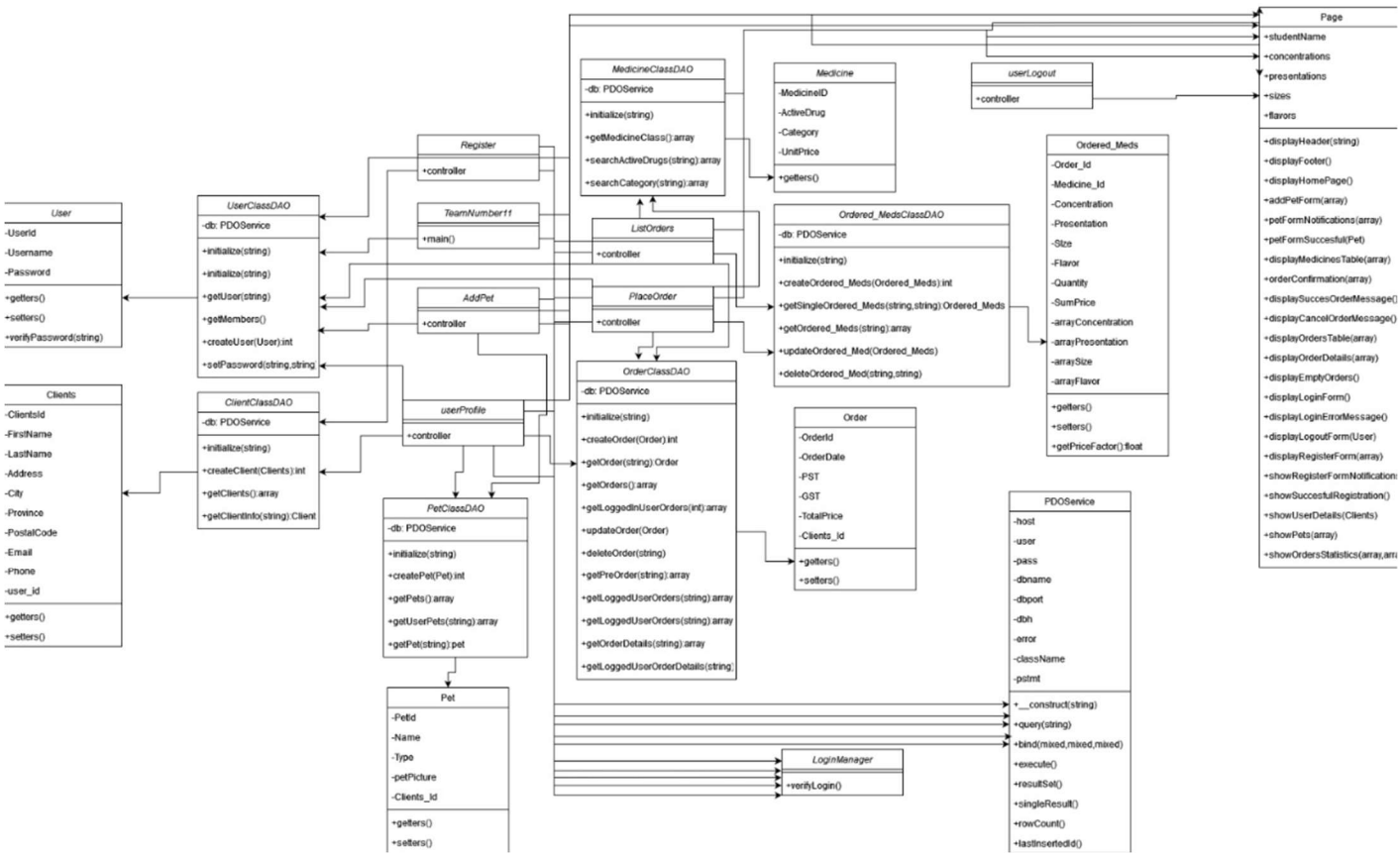
Run the commands show tables; In order to see the tables available in our database.

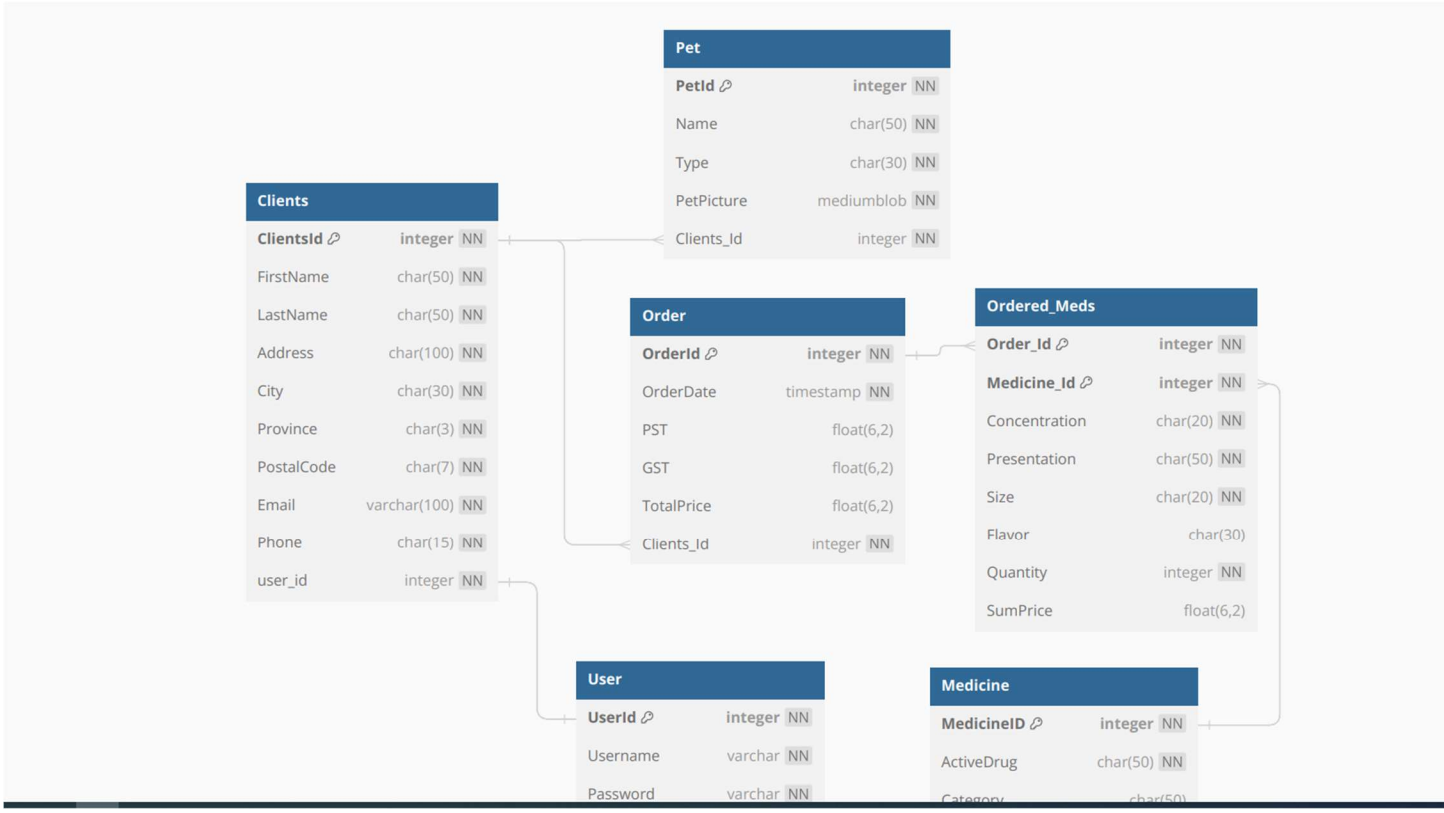
```
MariaDB [vetPharmacy]> show tables;
+-----------------------+
| Tables_in_vetpharmacy |
+-----------------------+
| clients               |
| medicine              |
| order                 |
| ordered_meds          |
| pet                   |
| user                  |
+-----------------------+
6 rows in set (0.001 sec)
```

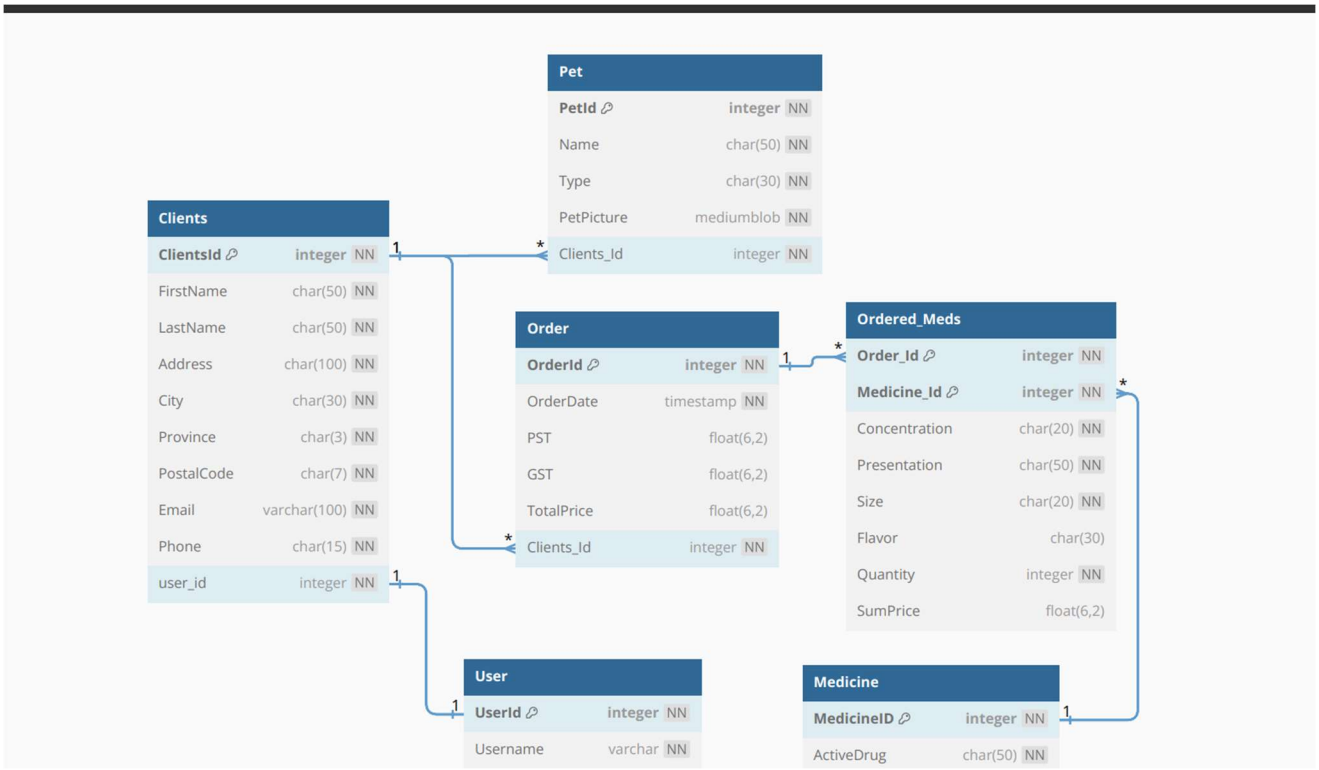
Technologies extension implementation

Search field for searching, uploading image files to the SQL.

2.Class Diagram







3.Meeting minutes

First meeting:

Attendance: Luiz Attias and Oliver Franco

What was Done since Last meeting? Explain the idea of the project and the basic functions to perform in the page. Github was created for the group to use and first commits were made.

Things to do before next meeting:

Oliver: Modify database, merge the veterinary and pet owner entities, create sql file and push it to GitHub.

Luiz: generate structure of project, folders and php files, start implementing to show the medicines displayed after the SQL was updated and push it to GitHub.

Meeting minutes: 1hour and 12 minutes.

Second meeting

Attendance: Luiz Attias and Oliver Franco

What was Done since Last meeting? Implementation of project structure, Page class, Entities, Utilities and configuration page, SQL with all the values were implemented, Displaying data in a table and authentication process.

Oliver: Modified database, merged the veterinary and pet owner entities, created sql file and pushed it to GitHub.

Luiz: generated structure of project, folders and php files, started implementing to show the medicines displayed after the SQL was updated and push it to GitHub.

Working well: Meeting times management are working well and division of processes.

Not working well: Merging the GitHub entities needs improvement.

Things to do before next meeting:

OrderDetailspage will be implemented as well as showing user details in the userProfile Page as well as user logout functions and pages.

Meeting minutes: 1hour and 45 minutes

Third meeting

Attendance: Luiz Attias and Oliver Franco

What was Done since Last meeting?

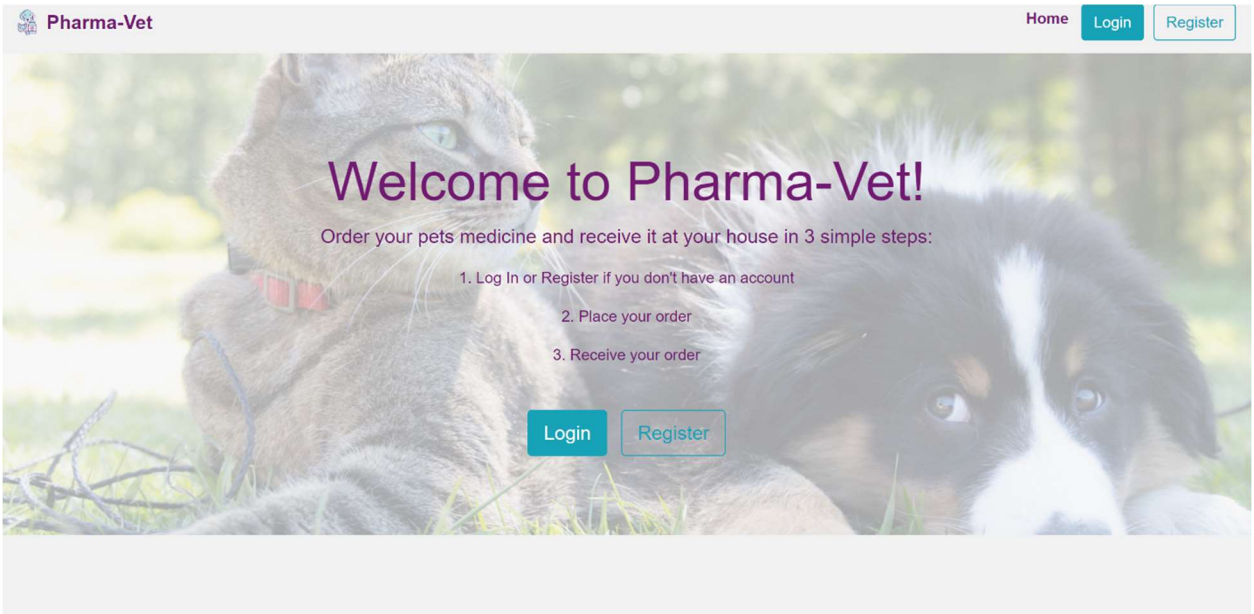
Oliver: Created entities classes to show medicines and pet information. Linked the pages created with the navbar available in the website and authenticated failure for pets entry.

Luiz: Implemented authentication, UserProfile page and user logout. Generated the validate form in order to validate all the fields in the register form for when a new user wants to register to the website.

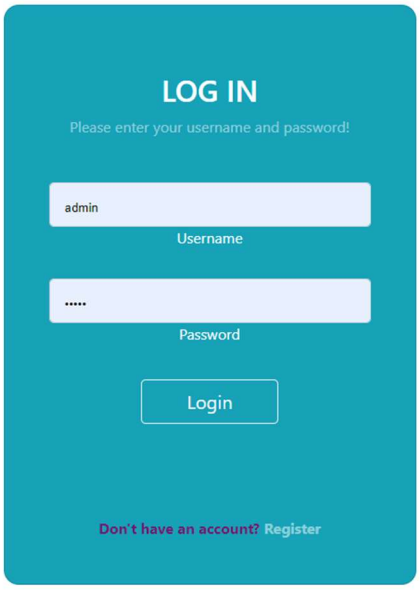
Working well: Division of processes and deadlines management.
Not working well: Still encountering issues while implementing the merge of git branches to the main.
Things to do before next meeting: Finish project final details and do the whole Project documentation.
Meeting minutes: 3hours and 50 minutes

User Manual

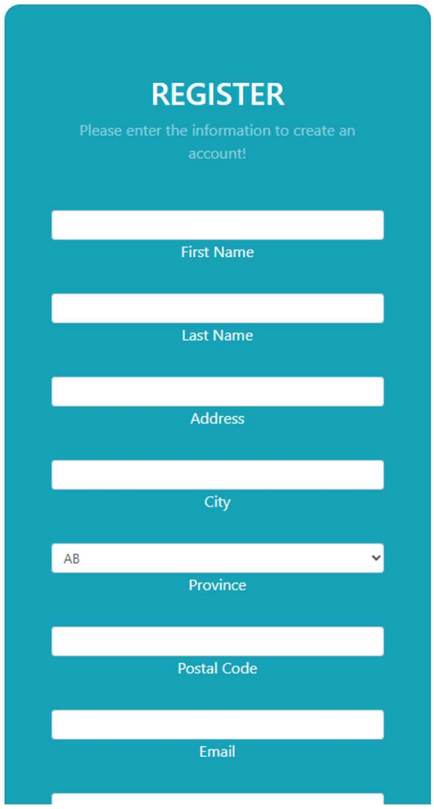
1st The landing page: The user or administrator here have the option to login with existing credentials or register for the first time. For the administrator, The username is admin and password is admin.



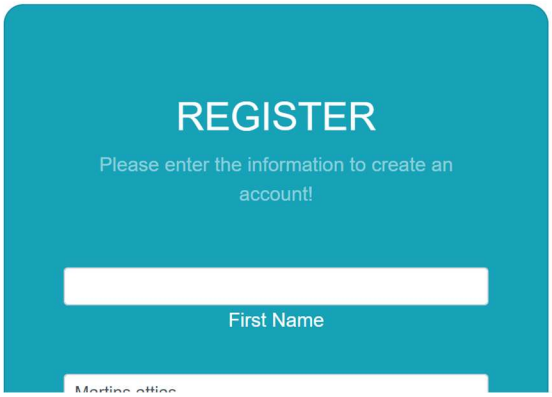
When the user clicks the login button, the login page will be displayed where the user or administrator will be asked to enter their username and password and if they match with our records in the database, they will be directed to the userProfile page.



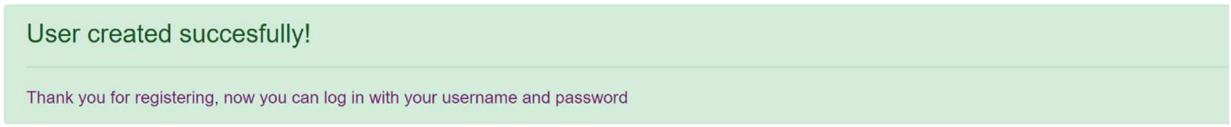
2nd the Register Form: If the user chooses to register for the first time and clicks the register button from the landing page, the register form will be displayed requiring the new user to input their information.



Here they will need to input their first name, last name, address, city, province, postal code, email address, phone number, the username they would like to use and password. If there are errors in the input fields from the user for example, if the first name is empty or the email is not in the correct format or the phone is not a 10 digit phone number, An error message will be displayed in order to let the user know that they need to fix it in order to continue before proceeding in clicking the register button.

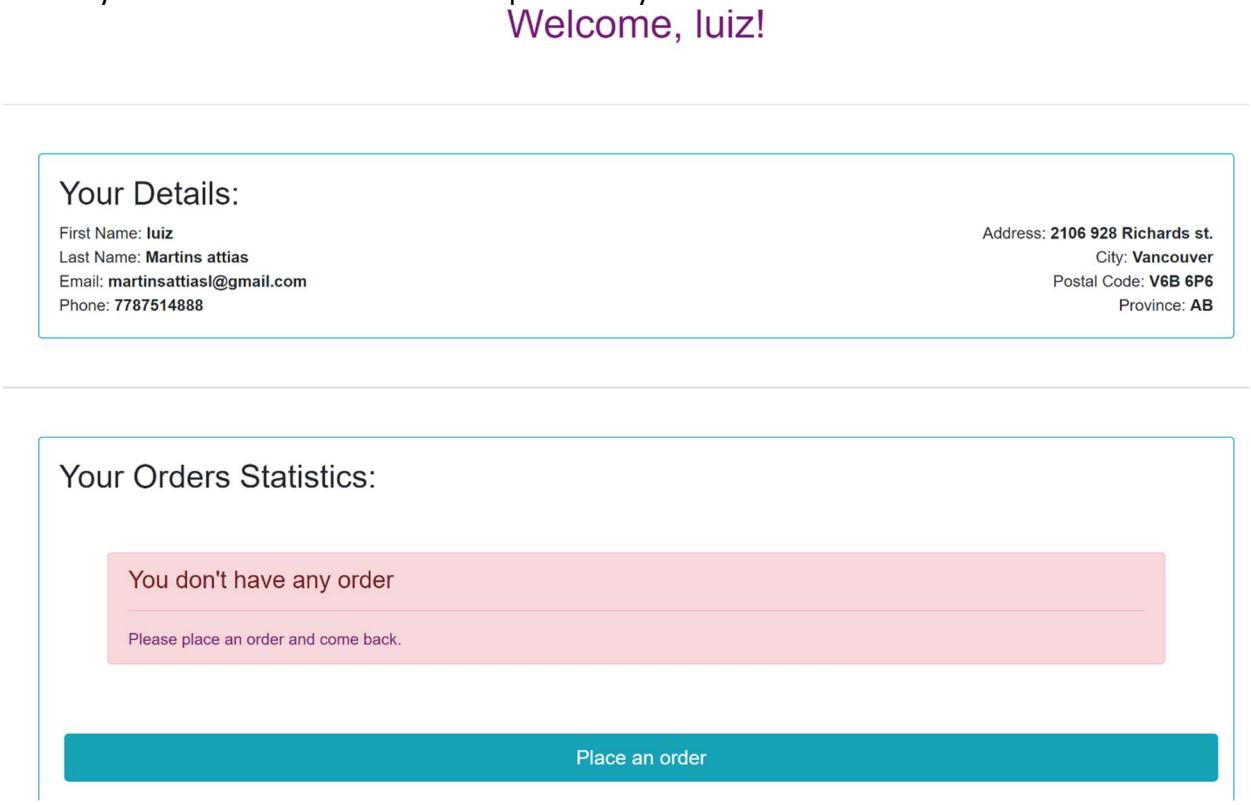


If these conditions are finally met, a successful message is displayed and now the user can login with their new username and password.

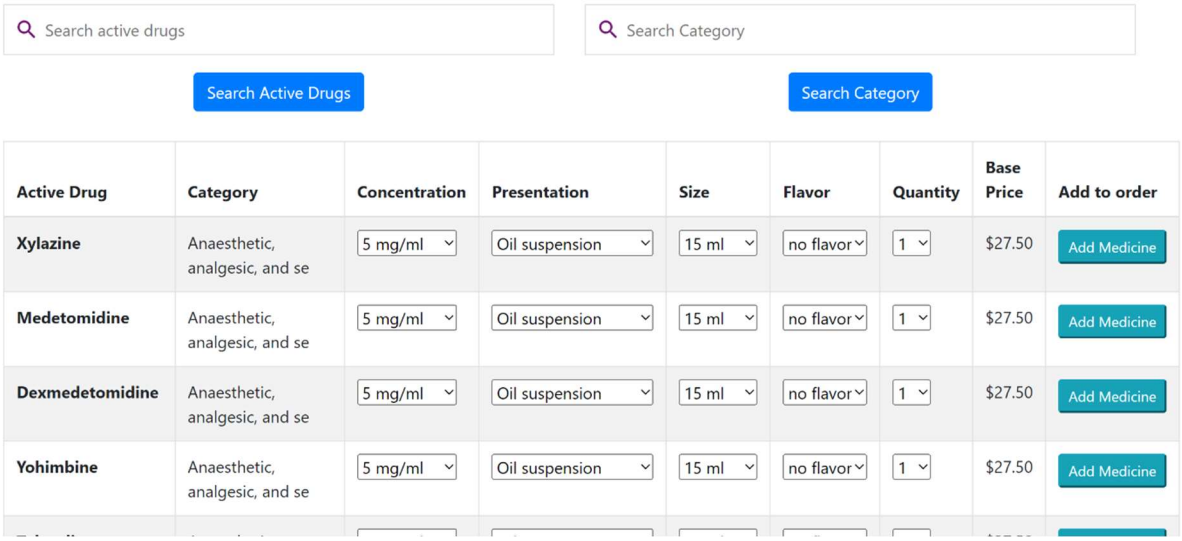


3rd The userProfile page: The user logs in with their information, Now they will be directed to their profile page where their name will be displayed, their information and

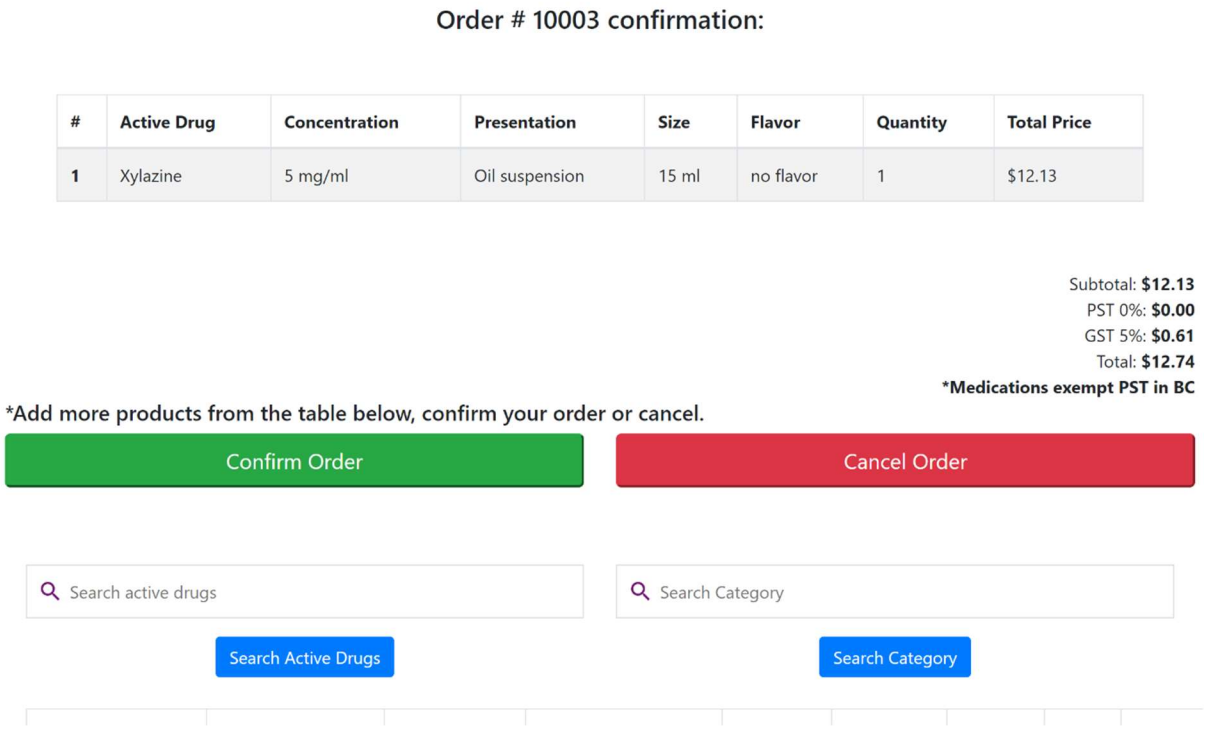
their current orders and current pets. Firstly the user will have no orders and no pets, but they can choose to add orders and pets at any time!



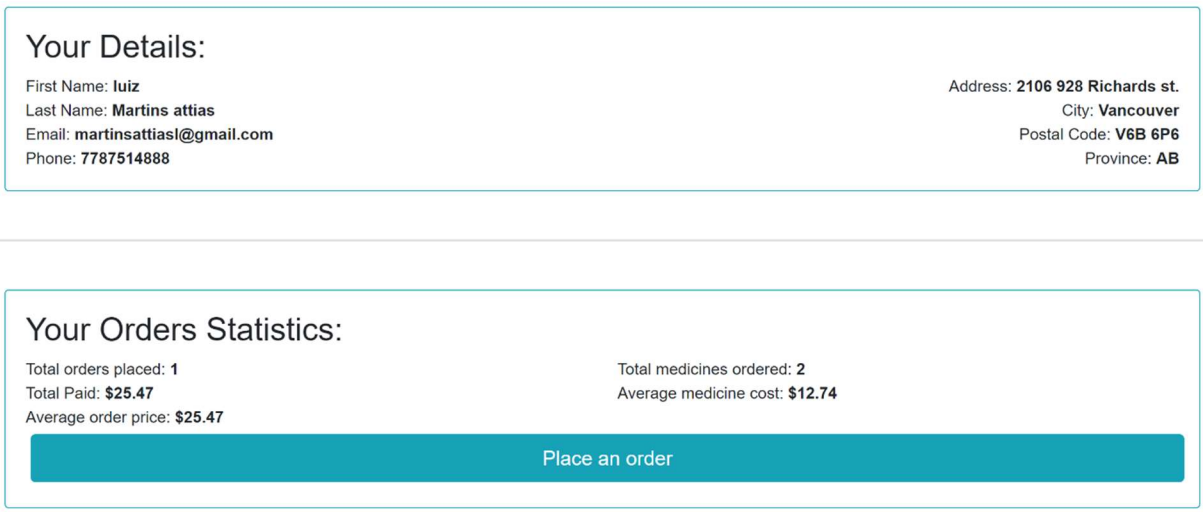
4th The order Page: When the user clicks on the button place an order they will be directed to the order page where they can choose drugs from the list or use the search field to search for a specific drug and thereafter click the button add medicine.



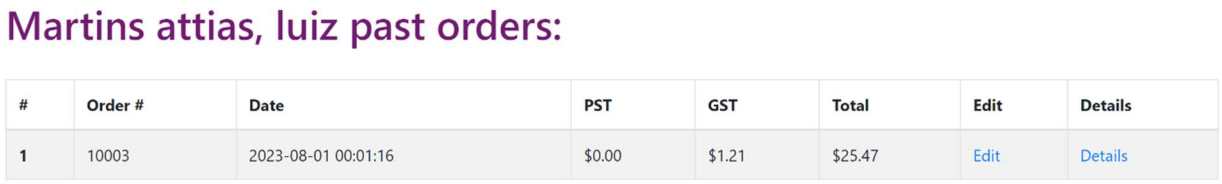
5th The add Medicine: The user clicking on add medicine button, will have the option to place that order, cancel it, or even add more orders if they choose to do so.



When they choose to confirm order, A successful message will be displayed and now the profile will be updated showing the user their new order added.

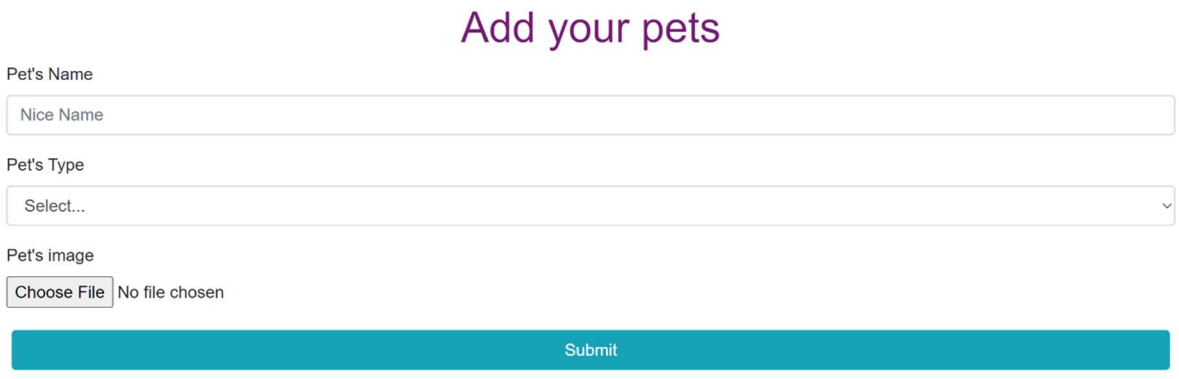


And when the place order is clicked, the orders will be sent to us and displayed in a page showing the users past orders that is available in the navigation bar place called “my orders”.



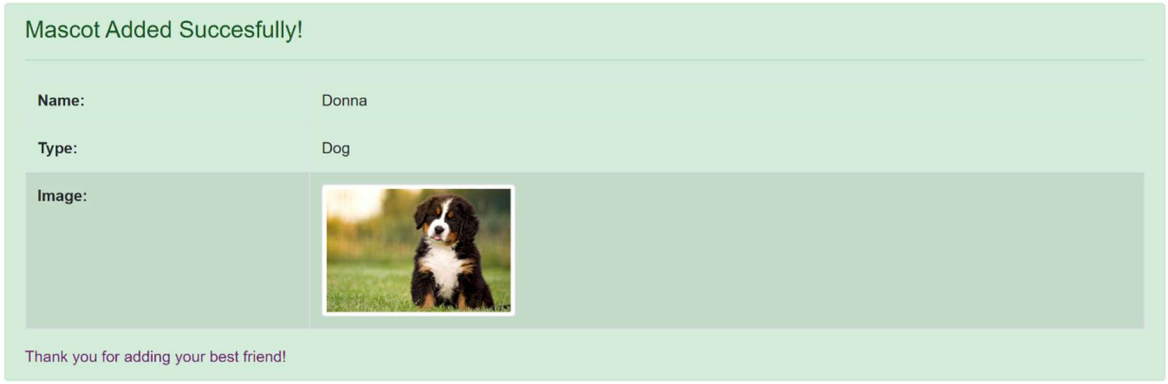
The user here has the option to click edit and therefore if they want to cancel, the order will be removed from their profile.

6th pet’s information: When the user chooses to add their pet the pets page will show with fields for pet’s name, pet’s type and a field for uploading an image of their pet.



If nothing is selected or no image of the pet is uploaded, the error message ill be displayed not letting the user go further until fixed.

If the validation is finally met, The pets information will be displayed in a successful message as well as in your main profile page.



7th Logout: User chooses to log out of their session, A thank you message will be displayed to thank the user for using our service and they can login again or register as a ne user again.

Citations

Stack Overflow - where developers learn, share, & build careers. (n.d.). Stack Overflow. https://stackoverflow.com/

GitHub: Let's build from here. (n.d.). GitHub. https://github.com/

draw.io - free flowchart maker and diagrams online. (n.d.). https://app.diagrams.net/

A free database designer for developers and analysts. (n.d.). https://dbdiagram.io/home/