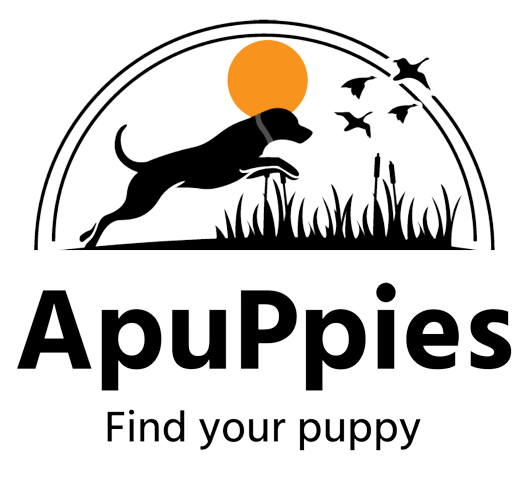


Web application report



realized by

Cléophas FOURNIER

Jean DOUTRIAUX

Tristan Duong

*supervised by*

Daniel MAGO VISTRO

*TI610-TI510M – Advanced Web programming*

Table of content

[I. Gantt chart 3](#_Toc121672128)

[II. Introduction 4](#_Toc121672129)

[III. System design 5](#_Toc121672130)

[Wireframe 5](#_Toc121672131)

[Home page design 5](#_Toc121672132)

[Login page 6](#_Toc121672133)

[Register page 6](#_Toc121672134)

[Balance/wallet of the user 7](#_Toc121672135)

[Buy page of the product 8](#_Toc121672136)

[Entity Relationship Diagram 8](#_Toc121672137)

[Activity diagram 8](#_Toc121672138)

[Login/Register 8](#_Toc121672139)

[Buy animal 9](#_Toc121672140)

[Navigation structure 11](#_Toc121672141)

[IV. Implementation 12](#_Toc121672142)

[Db.php 12](#_Toc121672143)

[Login 12](#_Toc121672144)

[Register 13](#_Toc121672145)

[Animals 14](#_Toc121672146)

[Creating 15](#_Toc121672147)

[Mail 16](#_Toc121672148)

[V. User guidance & sample screens 18](#_Toc121672149)

[Homepage 18](#_Toc121672150)

[Login & Register 18](#_Toc121672151)

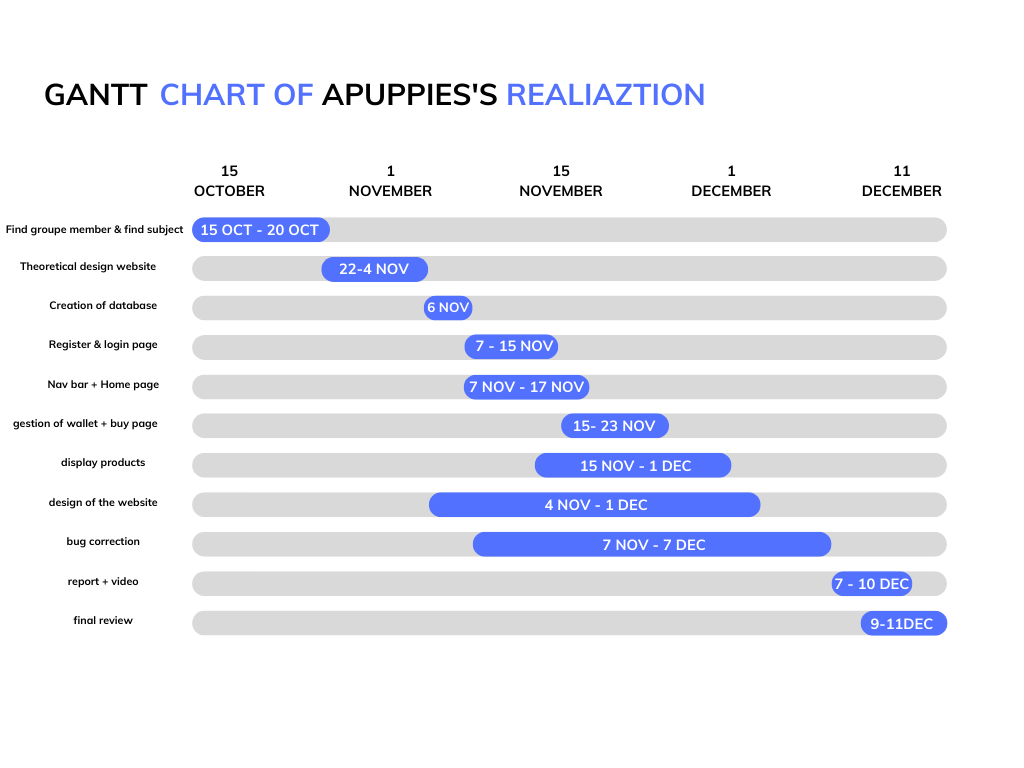
[Buy 19](#_Toc121672152)

[Sell & MyPets 20](#_Toc121672153)

[Mail 21](#_Toc121672154)

[VI. Conclusion 22](#_Toc121672155)

# Gantt chart



realized thanks to the site canva.com, we can visualize the different spots that we realized. this allows to have an overview of the design of the APUPPIES site.

# Introduction

Once the members were found and we had to find a website concept that appealed to us so that we were motivated to achieve it and that suited the expected criteria for the project. At first, we then discussed a lot. Our first idea was to create a social network so that users could post images or sentences and that at the end of each week the best posts would be highlighted. the problem was then that the site was not going to highlight money transactions. it did not fit the criteria. To meet expectations, we decided to create an online store. it suited all points. an account system in the data base. a product system and transactions so that the customer can buy items. for the subject we chose a "pet-shop". for the title we thought of a pun with APU. this is the idea: ApuPpies came to us, so we started to realize it.

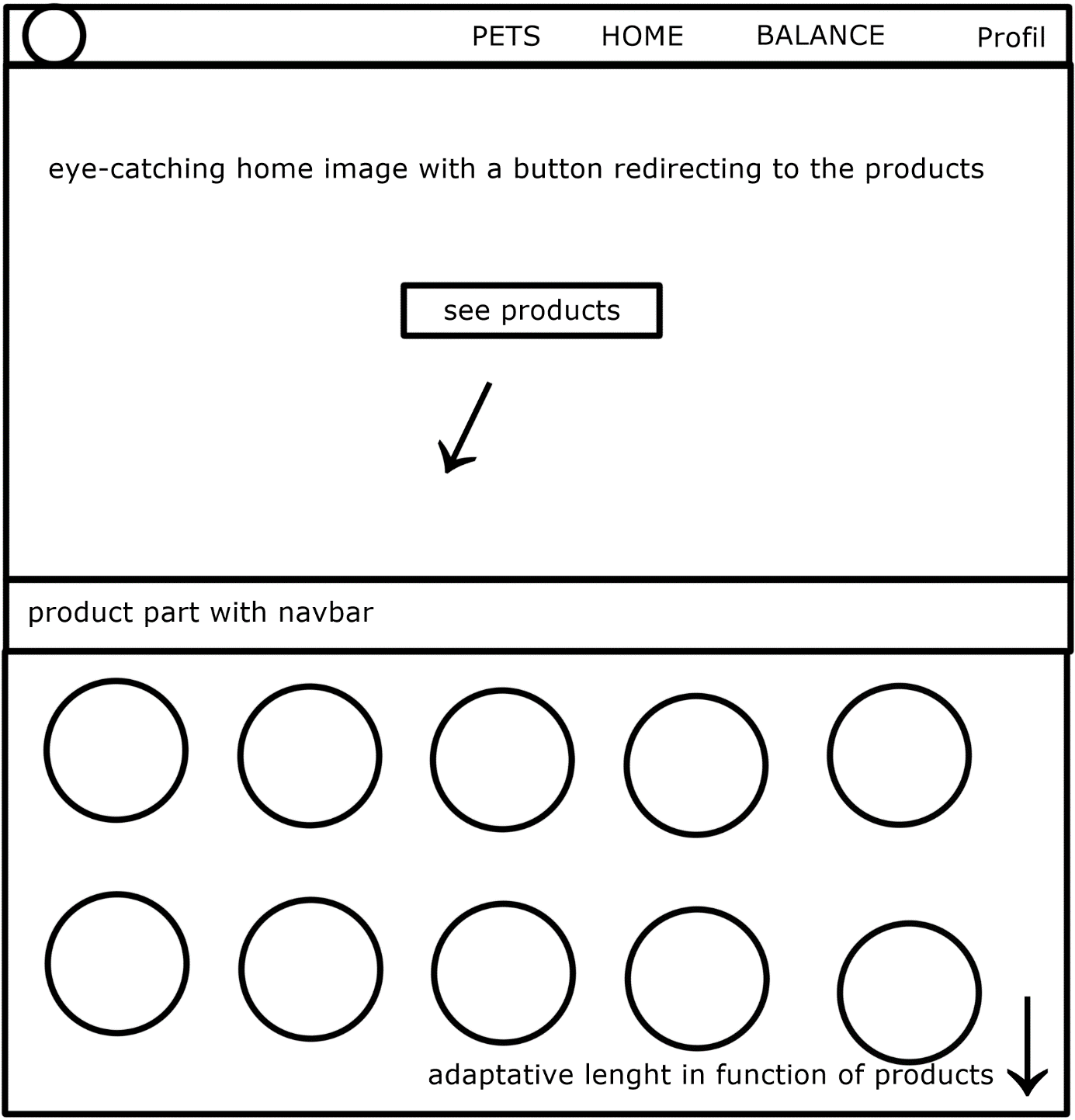
For the design of the site, we thought of making a cute site with colours so that the customer finds the store warm. moreover, we tried to bring an ethical & moral side to the concept. indeed, it was not just a site selling animals. the animals were sold by the customers directly. the site then served as a bridge between the customer and the seller. so that animal management is more organized and regulated. it can also prevent the abandonment of animals.

As we had to create this site with several people, we therefore used Git to be able to work all at the same time. It brought us different synchronization problems but overall, it really accelerated the realization of the site. We also had a problem with the management of our emails, but we will develop that later

# System design

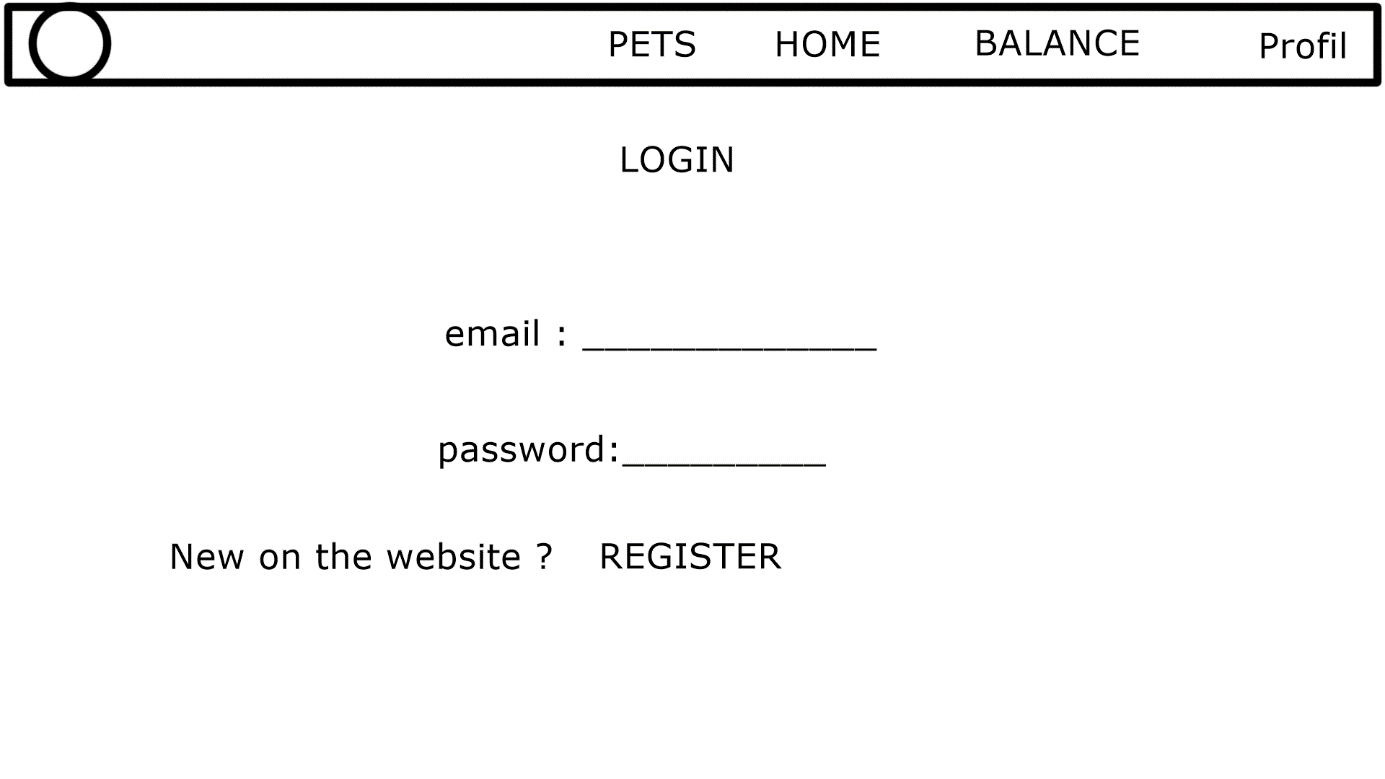
## Wireframe

### Home page design

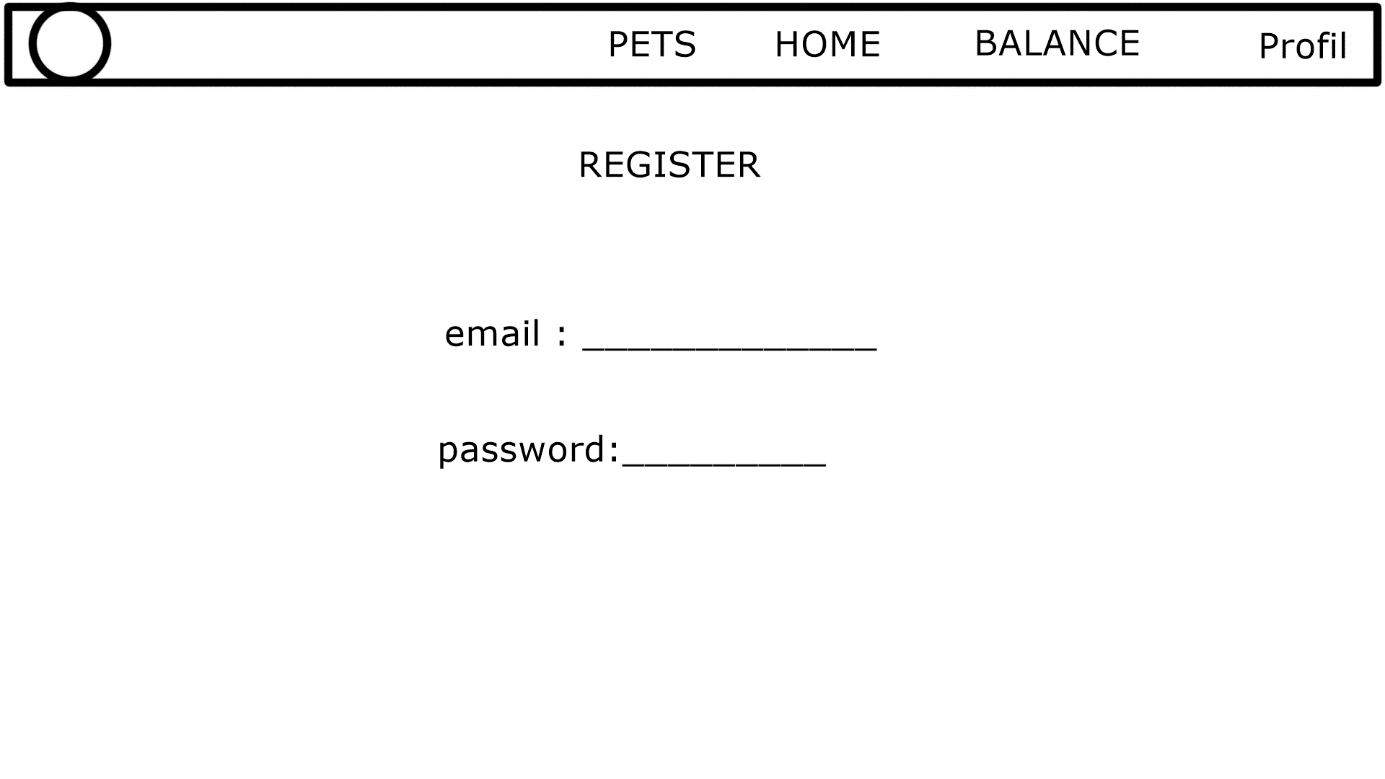


The design of a website's home page is one of the most important steps in the process of creating a website. It serves to present the website in a concise and attractive way for visitors. The home page design should be consistent with the rest of the site and reflect the image and identity of the business or organization. Here we wanted to make a catchy page with an animal so that the customer could directly understand that he was not on the wrong site to find what he was looking for. You can also find a navbar, a very useful tool for navigating through the site. Here we can see two one to navigate in the different parts of the site and another to sort the animals that the customer can see.

### Login page

For the register login pages our goal was to do something very simplistic. we can find just the part for the name and the password. moreover, if the customer does not have an account, he will then be redirected to the register page.

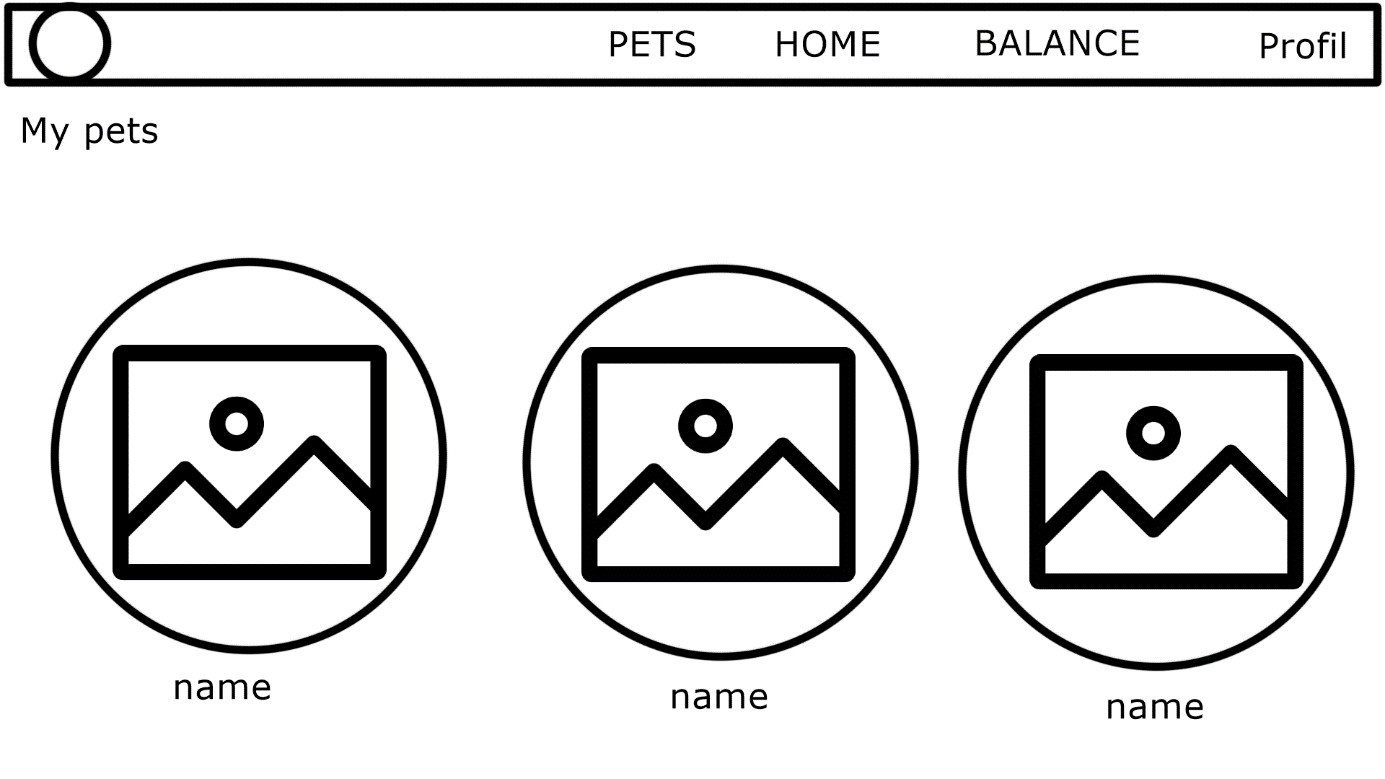
### Register page



like the login page we will use the same design to create the registration page.

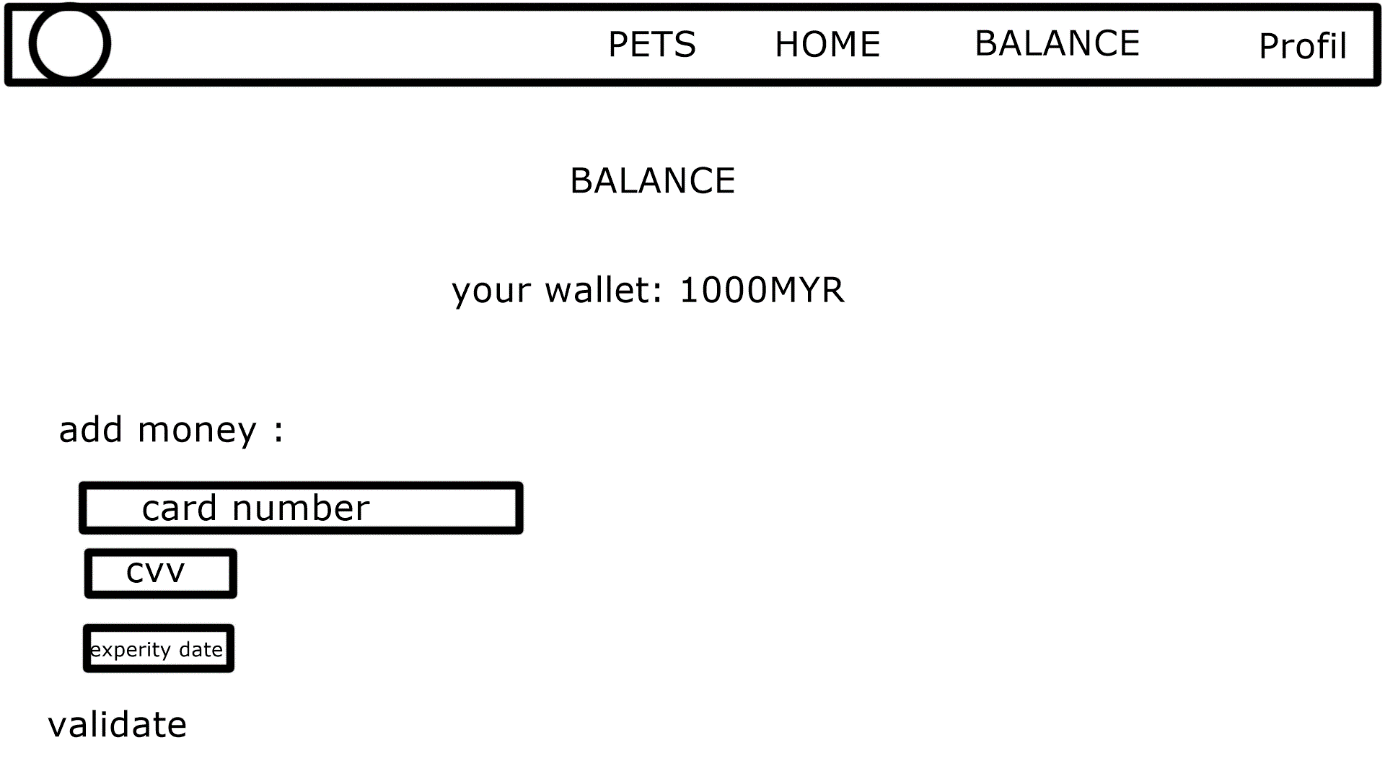
Page with the pets of the user

## My pets



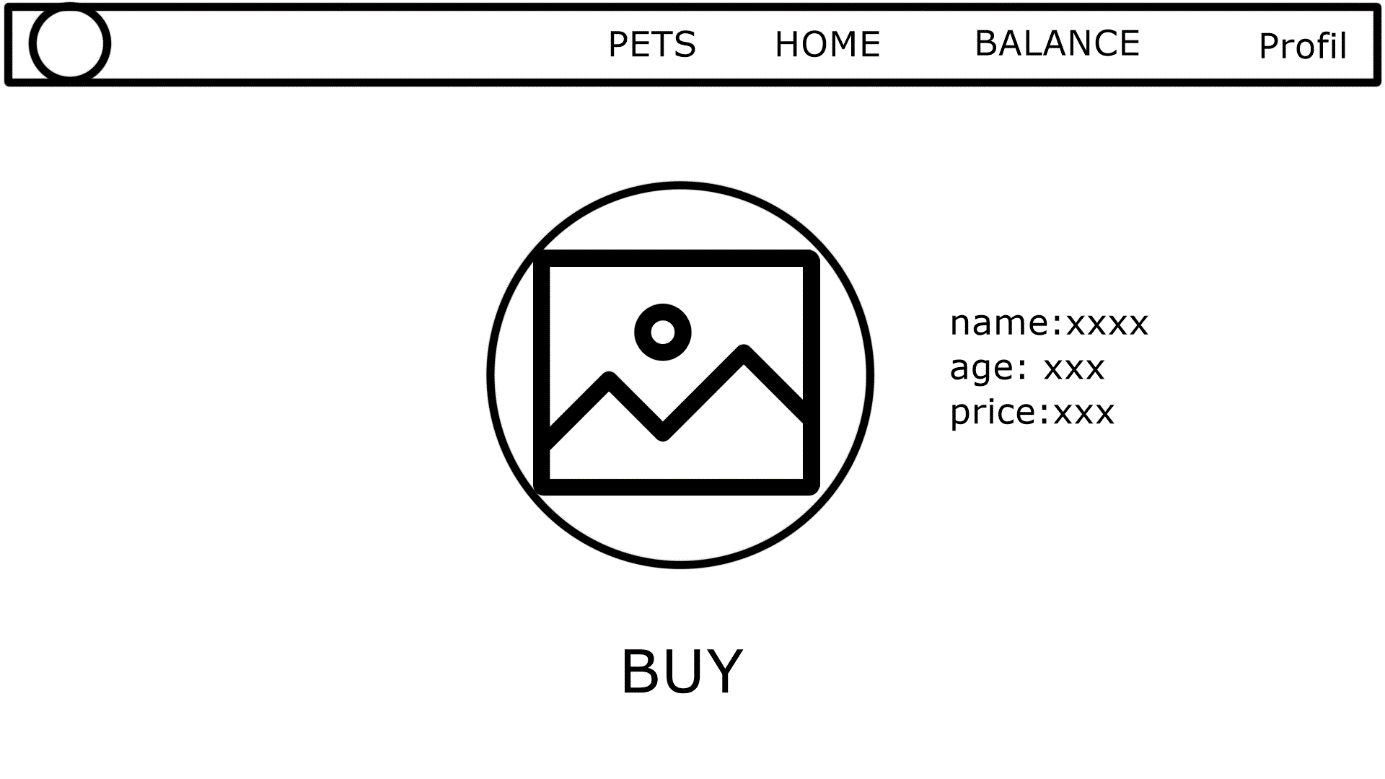
we decided to add a page where the customer can see the animals he has purchased. first it adds functionality to our site and moreover it allows you to see a kind of purchase history.

### Balance/wallet of the user



being an unofficial shop, it was difficult to put a real payment system. we then decided to add a balance to the customer's account. That he could feed with a realistic page.

### Buy page of the product



Finally, to create a good online store, we needed a purchase page. we have, like our site, created a simplistic and understandable buy page for the customer.

## Entity Relationship Diagram

Our database is composed of two entities with their own attributes. The user is having a name, mail & password used for login and a balance to buy pets, note that the mail is used as key attribute. The pet is composed of many attributes to describe them as products, an id which is unique as this is the key attribute and an owner if he’s been bought by a client.

Diagram

Description automatically generated

## Activity diagram

### Login/Register

The user needs to log on the website for this either login or register. Then SQL requests are sent to database and depending on the responses the user can be connected.

Diagram

Description automatically generated

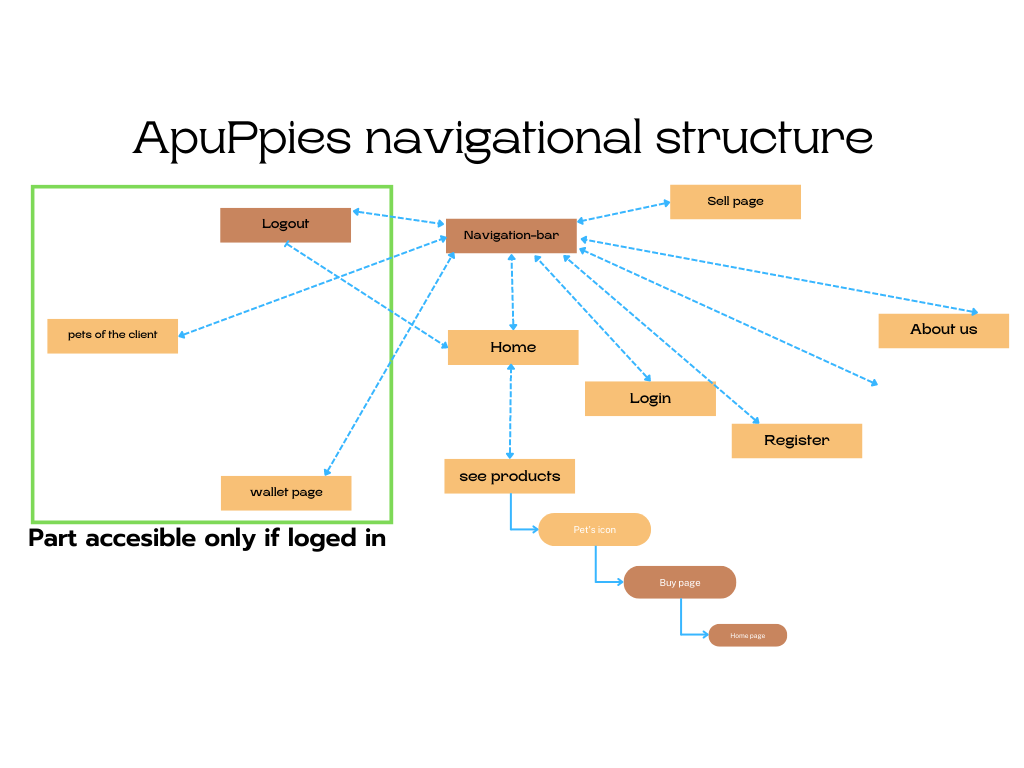
### Buy animal

The user can buy animal if they have sufficient funds. If the database validates this step then we add an owner to the pet and update the user’s balance. After this a mail is sent to the user to confirm the sale of the pet.

Diagram

Description automatically generated

## Navigation structure



# Implementation

## Db.php

Before all our website need a connection to a SQL database to save data and allow users to retrieve their data and actions.

<?php

ini\_set('display\_errors', '1');

$host = "localhost";

$user = "root";

$password = "";

$db = "DATAPETS";

$connexion = mysqli\_connect($host, $user, $password, $db);

if (!$connexion) {

die("Connection failed: " . mysqli\_connect\_error());

}

session\_start();

?>

The code is not that hard it just has to be required on all the other files. We can also notice that we start a session which will allow us to share a common variable between multiple pages. This variable being $\_SESSION[‘keyname’]

## Login

One of the first thing that the user have to do in our website is to either login or register if the user is new to the system. So here is the code snippet for both:

<form class="bonus" action="login.php" method="post" style="

display: flex;

flex-direction: column;

width: 30%;">

<div class="form\_group">

<input class="input" type="text" name="mail" placeholder="eMail">

</div>

<div class="form\_group">

<input class="input" type="text" name="name" placeholder="Name">

</div>

<div class="form\_group">

<input class="input" type="text" name="password" placeholder="Password">

</div>

<input class="border moved2 unset" type="submit" name="submit" value="Login">

</form>

The form is composed of 4 inputs; mail, name, password and the button to submit. The method of the form is POST so we can take all the inputs values from the variable $\_POST[‘keyname’]

<?php

if (isset($\_POST['submit'])) {

$mail = $\_POST['mail'];

$password = md5($\_POST['password']);

$query = "SELECT \* FROM account WHERE password = '$password' AND mail = '$mail'";

$result = mysqli\_query($connexion, $query);

if (mysqli\_num\_rows($result) > 0) {

$\_SESSION['mail'] = $mail;

$\_SESSION['name'] = $\_POST['name'];

echo session\_id();

header("Location: index.php");

} else {

echo "Error you have not been logged in😕<br>You might try another email or password🤔";

}

}

?>

Then when the submit button is hit, we send a SQL request to the account table to check if the user with this mail and password exist. If yes, then we log the user with the $\_SESSION variable and put the user at the homepage. If not, we display to the user that what has been send to the database doesn’t match and ask to try again.

To see the full code if this page here a link is: <https://github.com/cleophass/petshop/blob/master/login.php>

## Register

<form class="bonus" action="registration.php

" method="post" style="display: flex; flex-direction: column; width: 30%">

<div class="form\_group">

<input class="input" type="text" name="mail" placeholder="eMail">

</div>

<div class="form\_group">

<input class="input" type="text" name="name" placeholder="Name">

</div>

<div class="form\_group">

<input class="input" type="password" name="password" placeholder="Password">

</div>

<input class="border moved2 unset" type="submit" name="submit" value="Register">

</form>

<?php

if (isset($\_POST['submit']) && $\_POST['name'] && $\_POST['mail'] && $\_POST['password']) {

$name = $\_POST['name'];

$mail = $\_POST['mail'];

$password = md5($\_POST['password']);

$sql = "SELECT \* FROM account WHERE mail = '$mail'";

$result = mysqli\_query($connexion, $sql);

$user = mysqli\_fetch\_assoc($result);

if ($user) {

echo "User already exist";

echo "<br>

<a href=\"login.php\">Go to Login ⬅️</a>";

} else {

$query = "INSERT INTO account (name, mail, password) VALUES

('$name', '$mail', '$password')";

$result = mysqli\_query($connexion, $query);

if ($result) {

$\_SESSION['name'] = $name;

$\_SESSION['mail'] = $mail;

echo "You have been registered";

echo "<br>

<a href=\"index.php\">Go Home 🏡⬅️</a>";

} else {

echo "Error you have not been registered";

}

}

} else {

echo "Please fill all to register";

}

?>

For the register page it’s almost the same thing but with different cases. First, we check everything is filled up. Then is user already exist we just log him and otherwise we insert him to the database before login him

To see the full code if this page here a link is: <https://github.com/cleophass/petshop/blob/master/registration.php>

## Animals

After being connected to our websites multiple things can be done. The most common one being buying an animal.

This page is composed of only one form which is the buy button. Then depending on the session data many options can append.

<?php

require\_once 'mail.php';

if (isset($\_POST['buy']) && isset($\_SESSION['name'])) {

$name = $\_SESSION['name'];

$mail = $\_SESSION['mail'];

$query = "SELECT \* FROM account WHERE name = '$name' and mail = '$mail'";

$result = mysqli\_query($connexion, $query);

$row = mysqli\_fetch\_assoc($result);

$balance = $row['balance'];

$price = $animal['price'];

if ($balance >= $price) {

$newBalance = $balance - $price;

$query = "UPDATE account SET balance = '$newBalance' WHERE name = '$name'";

$result = mysqli\_query($connexion, $query);

if ($result) {

$\_SESSION['bought'] = true;

$\_SESSION['animal'] = $animal;

sendEmail("Congrats !", $mail, $name);

$query = "UPDATE pets SET owner = '$mail' WHERE id = " . $\_SESSION['petId'];

$result = mysqli\_query($connexion, $query);

// header("Refresh:0");

} else {

echo "Error you have not bought the animal";

}

} else {

echo "<br>You don't have enough money to buy this animal";

}

}

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

$query = "DELETE FROM pets WHERE id = " . $\_SESSION['petId'];

$result = mysqli\_query($connexion, $query);

if ($result) {

echo "<script>location.href='index.php'</script>";

} else {

echo "Error you have not deleted the animal";

}

}

?>

Then the code is just checking if there are sufficient funds to buy the animal. Balance is then updated, and this is when most of the things are happening. When the transaction is done, a mail is sent to the user’s mail which is saved in the $\_SESSION variable. And the name of the owner is added to the pet.

We can remark that the code is also composed of a condition of a delete button. This is because another form is appearing only if the user [admin@apuppies.com](mailto:admin@apuppies.com) is connected he can just remove animals from the catalogue.

The code of the full page is available under the following link: <https://github.com/cleophass/petshop/blob/master/buy.php>

## Creating

The last thing that a user can do in our website now is to sell his own pets. For this we just have a long form with required inputs and a simple SQL request to send it to the database.

<?php

if (isset($\_POST['sell'])) {

$name = $\_POST['name'];

$species = $\_POST['species'];

$race = $\_POST['race'];

$weight = $\_POST['weight'];

$age = $\_POST['age'];

$price = $\_POST['price'];

$sexe = $\_POST['sexe'];

$photo = $\_POST['photo'];

$sql = "INSERT INTO pets values

(NULL, '$name', '$species', '$race', '$weight', '$age', '$price', '$sexe', '$photo', NULL)";

$result = mysqli\_query($connexion, $sql);

if ($result) {

echo "Pet added successfully";

} else {

echo "Error adding pet";

}

}

?>

## Mail

When a client buys a puppy the sendEmail function is execute, we can see that in parameters the function receives the address mail of the client and the name of the puppy.

<?php

    ini\_set('display\_errors', '1');

    require 'includes/PHPMailer.php';

    require 'includes/SMTP.php';

    require 'includes/Exception.php';

    use PHPMailer\PHPMailer\*PHPMailer*;

*function* sendEmail($subject, $to, $name)

    {

        $mail = new *PHPMailer*();

        $mail->isSMTP();

        $mail->Host = "smtp.gmail.com";

        $mail->SMTPAuth = true;

        $mail->SMTPSecure = "tls";

        $mail->Port = "587";

        $mail->Username = "contact.apuppies@gmail.com";

        $mail->Password = "oxiywnbfxtxmypgj";

        $mail->Subject = $subject;

        $mail->setFrom('contact.apuppies@gmail.com');

        $mail->isHTML(true);

        // recuperer le contenue d'un ficher et le mettre dans une variable

        $mail->AddEmbeddedImage("assets/logo.png", "my-attach", "logo.png");

        $mail->Body = '<img alt="PHPMailer" src="cid:my-attach">';

        $mail->Body.= '<h1 style="font-family: Trebuchet MS, Arial, Helvetica, sans-serif;font-size: 40px;text-align: center;color:#003049;">

        Hello '. $\_SESSION['name'] .' from the ApuPuppies team !

    </h1>

    <h2 style="font-family: Trebuchet MS, Arial, Helvetica, sans-serif;font-size: 30px;text-align: center;">

    we congratulate you for your purchase in our shop<br>We hope that you take care of '. $\_SESSION['animal']['name'].'

    </h2>

    <h3 style="font-family: Trebuchet MS, Arial, Helvetica, sans-serif;font-size: 20px;text-align: center;font-style: italic;">if you have a question about your order you can contact us:

    ✉️ : contact.apuppies@gmail.com

    📞 : +60-11-2967-1427

    (This is an automatic message.)

    No refund or exchange is possible for pets purchased on our online store.

    if the animals have abnormal behaviour or reaction,

    contact us, we will redirect you to the best animal behaviour specialists. </h3>';

        // $mail->Body .= file\_get\_contents('mailtemplate.php');

// convert body to html

// convert to html

        // $mail->Body = "Congratulation " . $name . " !!<br>You just bought " . $\_SESSION['animal']['name'] . " 🎉🎉🎉";

        $mail->addAddress($to);

        if ($mail->send()) {

            echo "Check your mails 🤭❗<br>";

        } else {

            echo "Message could not be sent. Mailer Error: ";

        }

        $mail->smtpClose();

    }

    ?>

# User guidance & sample screens

The website is composed of 6 main pages

## Homepage

The first one being the homepage where you can find the catalogue of all pets by scrolling or just by hitting the button ‘See products’. Being sorted or not. All pages are also composed of the navbar which is allowing navigation between them.

A picture containing text, rock, outdoor, mammal

Description automatically generated

A screen shot of a dog

Description automatically generated with low confidence

## Login & Register

First thing to do for the user is to log to the website by either registering or login in with the two following pages

Diagram

Description automatically generated

Diagram

Description automatically generated

## Buy

Then when clicking on an animal from the homepage the buy page is coming, here from an admin view so the pet can be deleted.

A picture containing text, mammal

Description automatically generated

## Sell & MyPets

Now a user can either see their pets or sell some

A picture containing text, dog, mammal, different

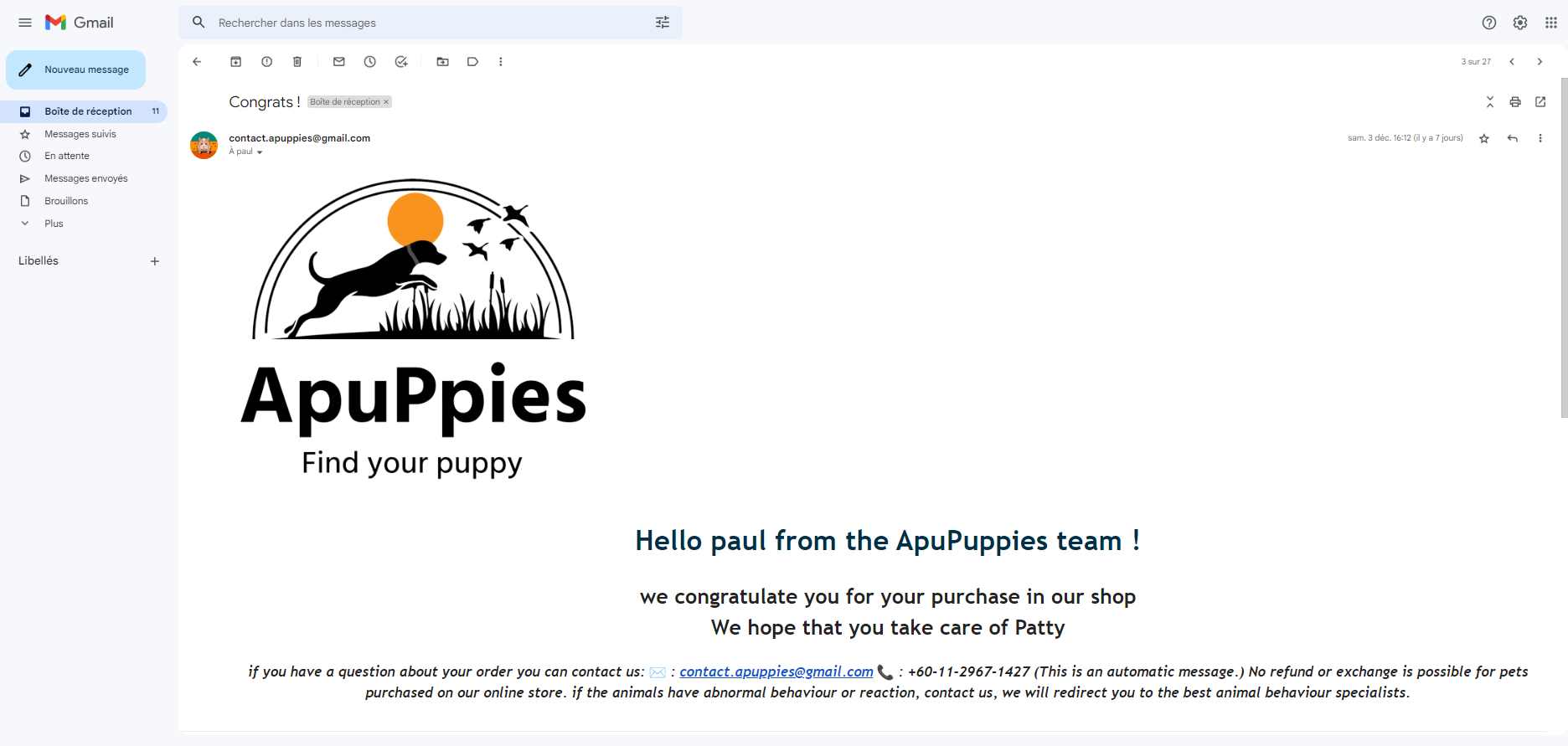
Description automatically generated

Table

Description automatically generated

## Mail

When a user just bought a new pet a mail from the corporate is directly send to him. The mail content the name of the user, the name of the pet and info about apuppies.



# Conclusion

At the end, this project didn’t teach us that much things. It helped us to see how all the lab works can be assembled to construct something concrete.

To conclude, this project has been a big success and we’re proud of the result. We used the technologies of the specifications, php and MySQL.

Though the mail part was close to be a failure due to the hack of our company mail by someone from Morocco. Indeed, due to the use of smtp system we’ve been easy targets for experimented hackers about mails.

First thing that we saw:



The mail sent by the hacker

Graphical user interface, text, application, email

Description automatically generated

Action of Google’s system:

Graphical user interface, text, application, email, Teams

Description automatically generated

Fortunately, we quickly saw it and blocked the smtp system for one day because our mail has been blocked by Google’s security.

The very next day our account was unlocked, we changed password and are now hopping that things like this won’t appear again.

We are confident now about future projects and this one will probably be a base for many future experiences.