

PawFect: Online Pet Shop

"The only place to find a true friend"

PRODUCT REPORT

GROUP MEMBERS

OLEKSII MELNYK	U1910039
NODIRA TILLAYEVA	U1910041
JAVOHIR SHOYUSUPOV	U1910056
ROBERT KHAN	U1910059
KOBILJON KOMILOV	U1910092



BY CLOWN9
PREPARED FOR IP GROUP PROJECT

Greetings from Clown9!

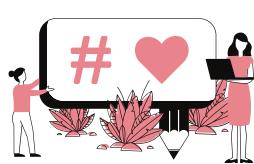
About Us

We are a data-driven, customer-centric web development company that is capable of turning everyday problems into innovative solutions in the form of websites. We are obsessed with empowering businesses to work quicker and better, through **responsive technology**.

Development Team



Oleksii Melnyk
Authentification



Nodira Tillayeva
Frontend/UX/UI



Javohir Shoyusupov
Admin Page Backend



Robert Khan
User Backend



Kobiljon Komilov
Database

PawFect

"PawFect" is a product that was developed and established in the frames of our Internet Programming Group Project. This Project made us think about homeless animals and people who want to adopt pets. We developed this platform to serve as a bridge between owners who want to give away their pets (as they cannot take care of them anymore) and people who want to adopt them.

Target Audience

Our consumers are citizens of Uzbekistan old enough to access internet websites! There are 3 types of people on our platform: people who sell, people who search for pets and admins. The first two categories of our users share the same functionality of our website taking into account that people who want to give away animals might also want to get new pets and vice versa!

The functionality of the Website

- Searching for available pets in your region by different pet categories and other keywords including their breed our other specifications
- Placing announcements about pets that users want to give away
- Ordering pets sends an email to the owners with contacts of a person interested in their pets
- Admins can monitor users post and delete them in case if they post unwanted content
- Users can contact the development team by email on our Contact Us section

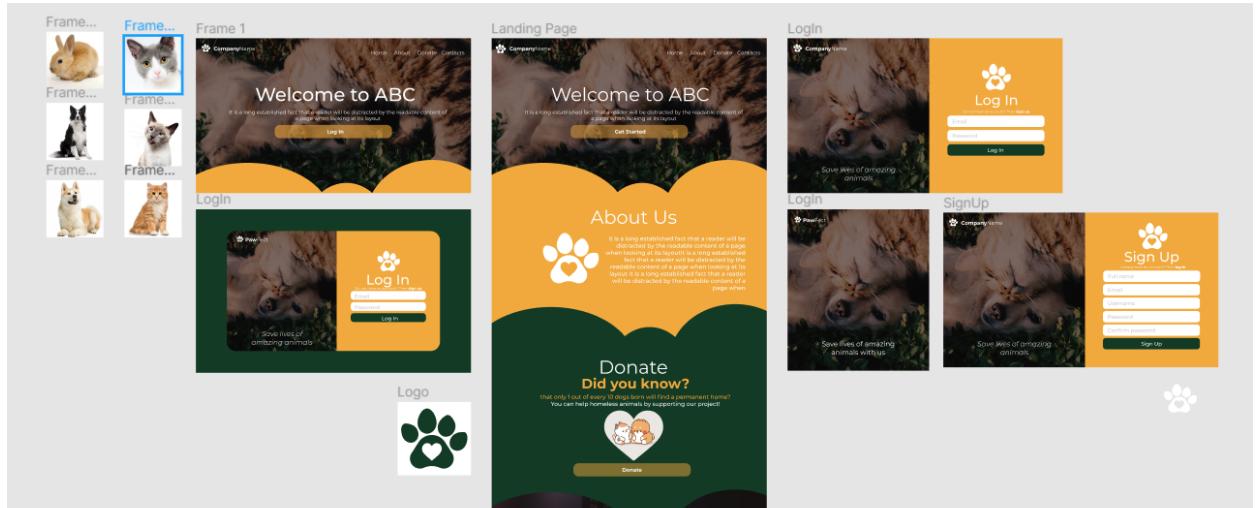
Screenshots and Implementation

Design Step

The whole UX/UI planning and Frontend Development was accomplished using HTML, CSS, Bootstrap (carousels, navbars, search bars, parallax), Font Awesome Icons, Google Fonts

See initial design process:

<https://www.figma.com/file/1M6TGKF7msV2ACZcONej7v/IP-Final-Project?node-id=0%3A1>



Databases on Laravel + Content

Two tables in database were created: one for users info, the second one is for posts. Respectfully 2 Models SiteUsers and SitePosts were created using php artisan make:model NameOfModel -m(to add migration files)

```
ps File Edit View Navigate Code Refactor Run Tools VCS Window Help group-project-clown9\group-project-clown9> Pawfect> database migrations 2021_05_22_073333_create_site_users_table.php
D:\Pawfect\group-project-clown9\group-project-clown9> Pawfect> database migrations 2021_05_22_073333_create_site_users_table.php
Project Files > D:\Vending.blade.php > landing_layout.blade.php > 2021_05_22_073528_create_site_posts_table.php > 2021_05_22_073333_create_site_users_table.php
  > 2014_10_12_000000_create_users_table.php
  > 2014_10_12_100000_create_password_resets_table.php
  > 2019_08_19_000000_create_failed_jobs_table.php
  > 2021_05_22_073333_create_site_users_table.php
  > 2021_05_22_073528_create_site_posts_table.php
  > 2021_05_22_081015_create_stars_table.php
    > seeders
      > CreateSiteUsersTable
        > up()
          public function up()
          {
            Schema::create('site_users', function (Blueprint $table) {
              $table->id();
              ...
            });
          }
        
```

Migrated: 2021_05_22_073333_create_site_users_table (383.82ms)
Migrating: 2021_05_22_073528_create_site_posts_table (894.60ms)
Migrated: 2021_05_22_073528_create_site_posts_table (894.60ms)
Migrating: 2021_05_22_081015_create_stars_table
Migrated: 2021_05_22_081015_create_stars_table (1,439.21ms)

Terminal Local > +
Structure Favorites Event Log
D:\Pawfect\group-project-clown9\PawFect> [] PHP: 5.6 27:40

File Edit View Navigate Code Refactor Run Tools VCS Window Help group-project-clown9 [D:\Pawfect\group-project-clown9]\...database\migrations\2021_05_22_073528_create_site_posts_table.php

D:\Pawfect\group-project-clown9 Pawfect database migrations 2021_05_22_073528_create_site_posts_table.php Add Configuration... Database

Project Files

- D:\Pawfect\group-project-clown9
 - idea
 - Pawfect
 - app
 - Console
 - Exceptions
 - Http
 - Models
 - Providers
 - bootstrap
 - config
 - database
 - factories
 - migrations
 - 2014_10_12_000000_create_users_table.php
 - 2014_10_12_100000_create_password_resets_table.php
 - 2019_08_19_000000_create_failed_jobs_table.php
 - 2021_05_22_073333_create_site_users_table.php
 - 2021_05_22_073528_create_site_posts_table.php
 - 2021_05_22_081015_create_stars_table.php
 - seeders
 - spatie

Terminal Local +

```
Migrated: 2021_05_22_073333_create_site_users_table (383.82ms)
Migrating: 2021_05_22_073528_create_site_posts_table
Migrated: 2021_05_22_073528_create_site_posts_table (894.60ms)
Migrating: 2021_05_22_081015_create_stars_table
Migrated: 2021_05_22_081015_create_stars_table (1,439.21ms)
```

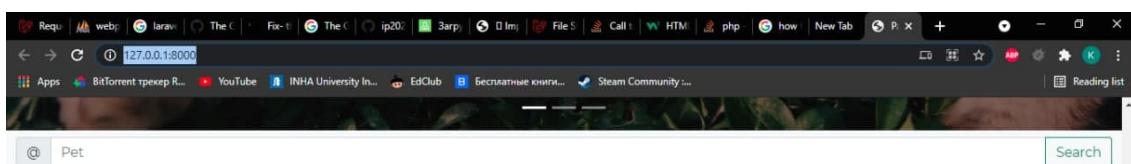
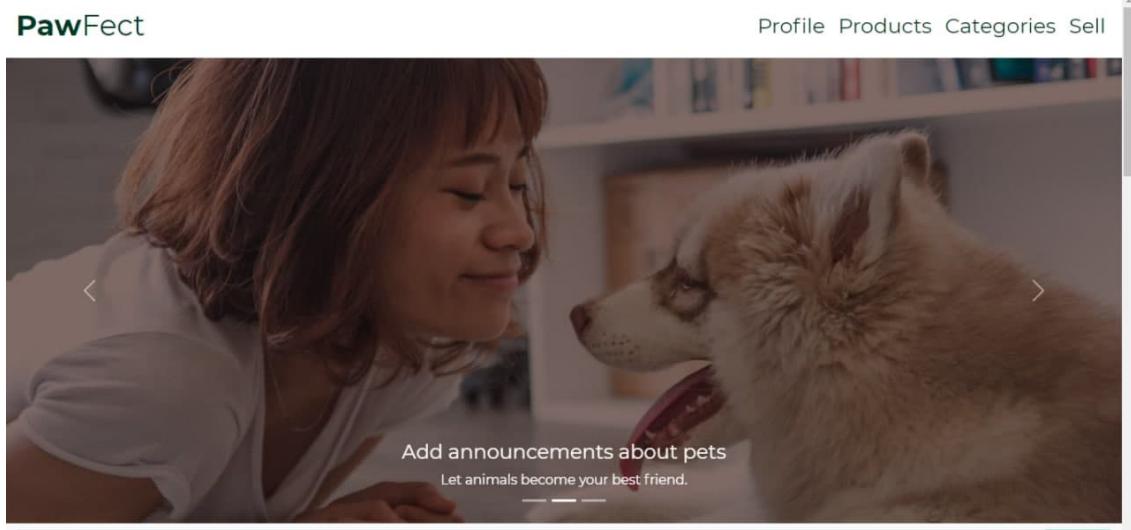
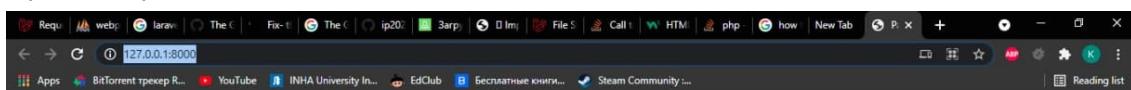
D:\Pawfect\group-project-clown9\PawFect

Event Log PHP 5.6 25:41

To start a debug session, hold Ctrl+Shift and click the link. // Don't show again (42 minutes ago)

All HTML were transitioned to respectful blade.php files.

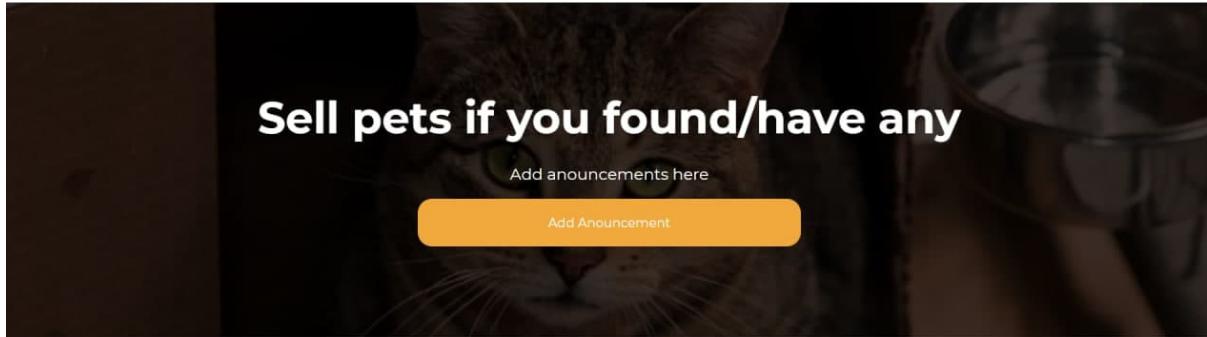
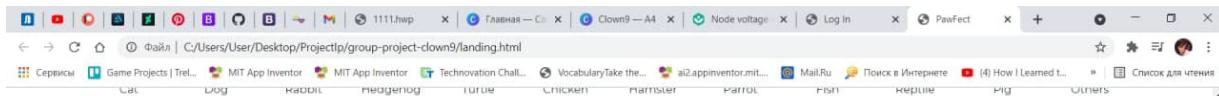
Main page – landing.blade.php were split into layouts according to the static and dynamic parts



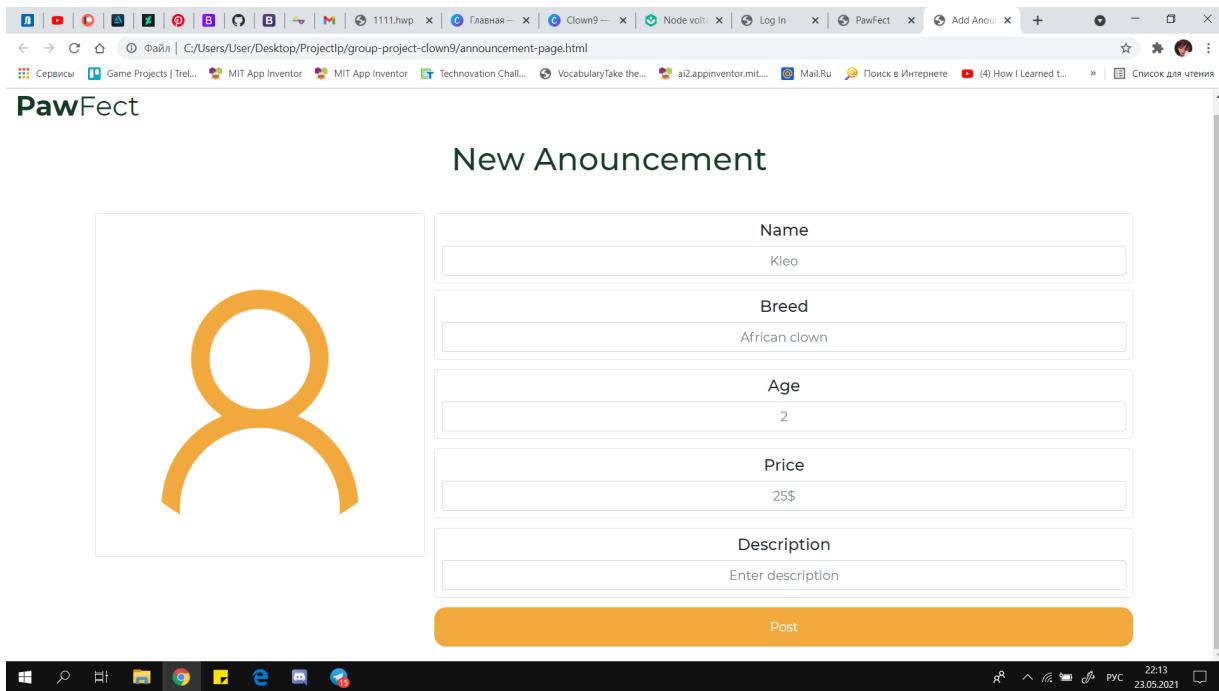
Categories



In main page in the shop section all the created posts are posted. They can be sorted using categories section. Search bar is used to search for post by their name.



Add announcement button will lead to the page



Here new post will be created after filling all the necessary fields

After the post button were pressed new post is added to db and the user is lead to main page

Where he/she can already see their post being published

```

<form action="{{ route('post-form') }}" method="post" enctype="multipart/form-data">
    @csrf
    <div class="container py-2">
        <div class="row py-2">
            <div class="col-1-lg-4">
                <div class="card mb-1 p-2 text-center">
                    <input type="file" name="Image" src="assets/profileAvatar.png" class="img-fluid mb-3" />
                </div>
            </div>
            <div class="col-lg-8">
                <div class="row">
                    <div class="card mb-1 p-2 text-center">
                        <h5> Name </h5>
                        <input type="text" name="Name" placeholder="Kleo" class="form-control text-center" />
                    </div>
                </div>
            </div>
        </div>
    </div>

```

Creating new post is realized using Form

```

Route::name('user.->group(function(){
    Route::view('landing', 'private')->middleware('auth')->name('private');

    Route::get('/login', function(){
        if(Auth::check()){
            return redirect(route('user.private'));
        }
        return view('Auth.login');
    })->name('login');
    Route::get('/Logout', function(){
        Auth::logout();
        return redirect('/login');
    })->name('logout');
    Route::post('/login', [\App\Http\Controllers\RegisterController::class, 'login']);

    Route::get('/Logout', [])->name('logout');
    Route::get('/registration', function (){
        return view('Auth.register');
    })->name('registration');

    Route::post('/registration', [\App\Http\Controllers\RegisterController::class, 'save']);
});

//Route::get('/', [AdminController::class, 'addNewUser']);
Route::post('/admin/search', [AdminController::class, 'showSearch'])->name('searchForAdmin');
Route::get('/admin', [AdminController::class, 'showUsers'])->name('showUsersToAdmin');
Route::post('/admin/delete', [AdminController::class, 'deleteItem']);
Route::get('/admin/not_found', [AdminController::class, 'firstOr']);

Route::get('', [PetsController::class, 'showAllPets'])->name('landing-pets');
Route::post('/products/search', [PetsController::class, 'searchForPet'])->name('searchForPet');
Route::get('/order{id}', [PetsController::class, 'showPostDetail'])->name('post-detail-pets');
Route::get('/admin/details/order{id}', [AdminController::class, 'showDetailsToAdmin'])->name('admin-details');
Route::get('/categories=categories', [PetsController::class, 'sortByCategory'])->name('sort-post');
Route::get('/post', function (){
    return view('announcement-page');
});

Route::post('/postform', action: 'App\Http\Controllers\PostController@submit')->name('post-form');
Route::get('/Landing', [PetsController::class, 'showAllPets']);
Route::get('/displayEditProfile{id}', [App\Http\Controllers\ProfileController::class, 'displayEditProfile']);
Route::post('/editProfile{id}', [App\Http\Controllers\ProfileController::class, 'editProfileFunc'])->name('edit-profile');
Route::get('/editPost', [App\Http\Controllers>EditPostController::class, 'displayEditPost'])->name('edit-post');

//Route::get('/Products.#categories')->name('categories-jump');
})

```

```

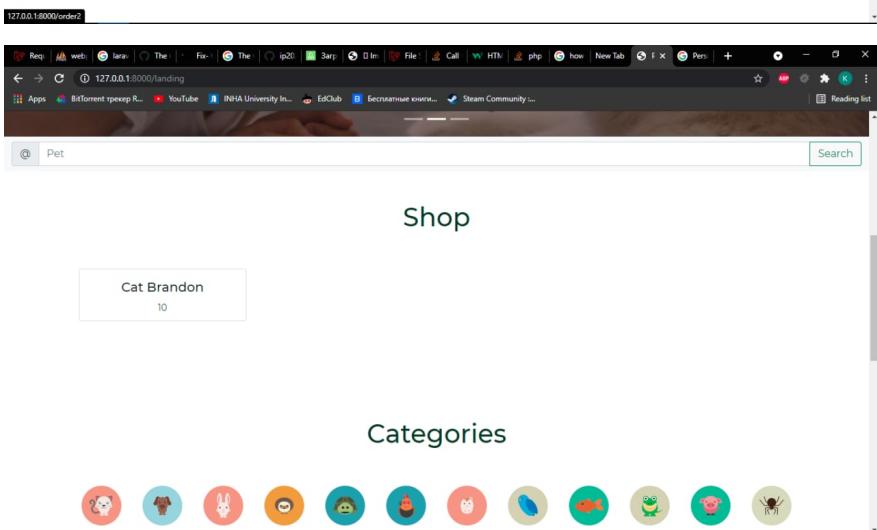
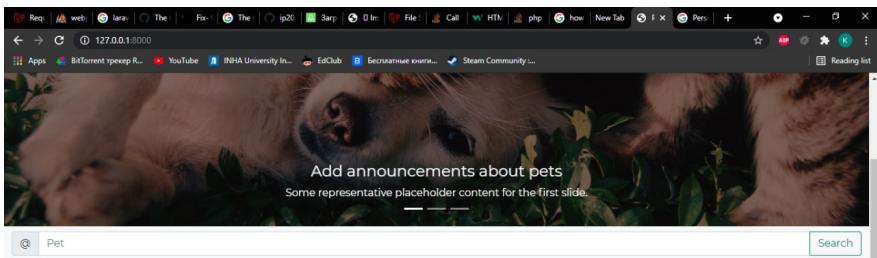
Route::get('/products/search', [PetsController::class, 'searchForPet'])->name('searchForPet');
Route::get('/order{id}', [PetsController::class, 'showPostDetail'])->name('post-detail-pets');
Route::get('/admin/details/order{id}', [AdminController::class, 'showDetailsToAdmin'])->name('admin-details');
Route::get('/categories=categories', [PetsController::class, 'sortByCategory'])->name('sort-post');
Route::get('/post', function (){
    return view('announcement-page');
});

Route::post('/postform', action: 'App\Http\Controllers\PostController@submit')->name('post-form');
Route::get('/Landing', [PetsController::class, 'showAllPets']);
Route::get('/displayEditProfile{id}', [App\Http\Controllers\ProfileController::class, 'displayEditProfile']);
Route::post('/editProfile{id}', [App\Http\Controllers\ProfileController::class, 'editProfileFunc'])->name('edit-profile');
Route::get('/editPost', [App\Http\Controllers>EditPostController::class, 'displayEditPost'])->name('edit-post');

//Route::get('/Products.#categories')->name('categories-jump');
}

```

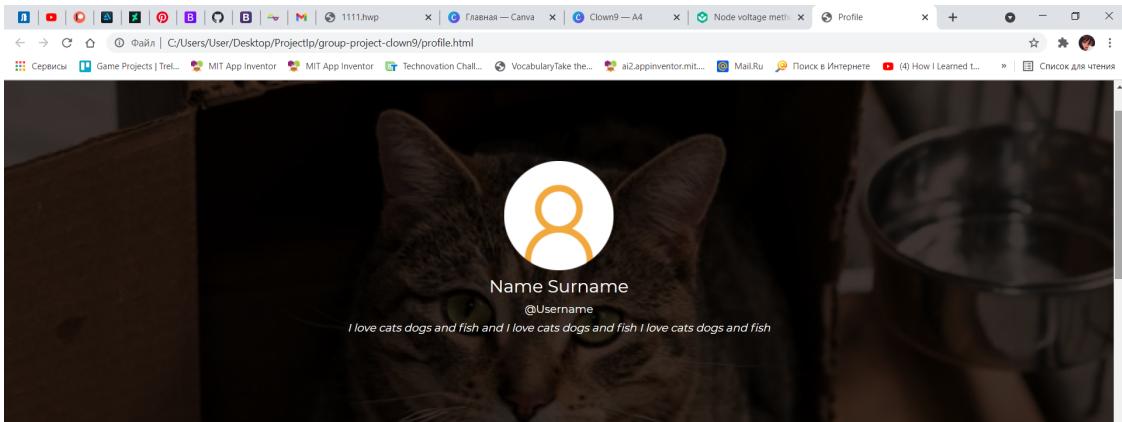
All the transitions are implemented via Route in web.php



When you press the details button more details will pop up where you can see the description as well as the owners mobile number. As it is only the first early version of the platform there is only a phone call option is available but in the future, the options like texting via email will be added

A screenshot of a web browser showing a detailed info page for "Cat Brandon". The page includes a photo placeholder, a "Description" field with the text "I want to sell my cat Brandon, because i cannot longer take care of him due to health issues", a "Breed" field with "Persian", an "Age" field with "2", an "Owner Name" field with "jeremy", an "Owner Number" field with "+998-99-9990000", and a "Date Posted" field. The top navigation bar shows the URL "127.0.0.1:8000/order2" and the title "PawFect".

From every page, user can always return to the main page clicking our logo at the top left corner

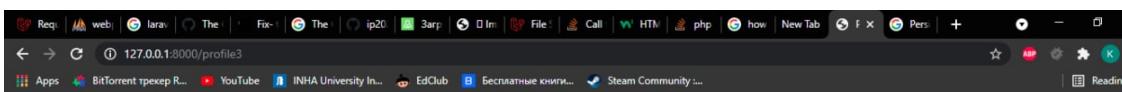


Detailed Information

Email ntillayeva@gmail.com	Phone Number 99893102347234	Date of birth 27/05/2001	Passport Info AB345345	Postal Code 100139	City Tashkent, Uzbekistan
-------------------------------	--------------------------------	-----------------------------	---------------------------	-----------------------	------------------------------



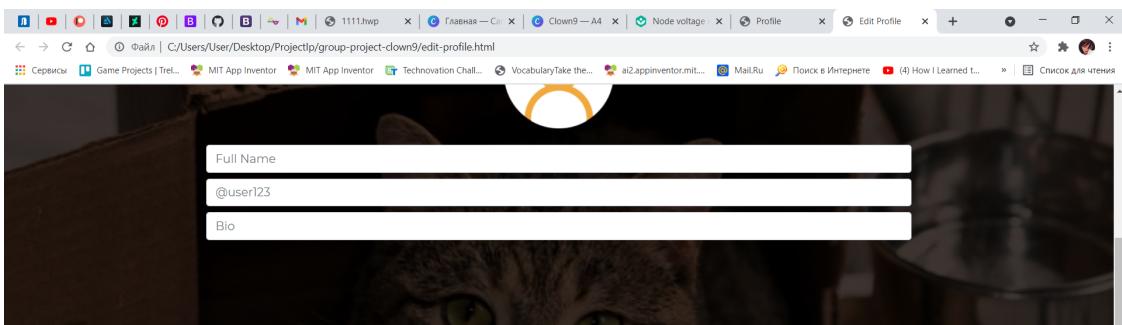
This page contains all the info about current user



Edit Info

This person is selling

Cat Brandon
10



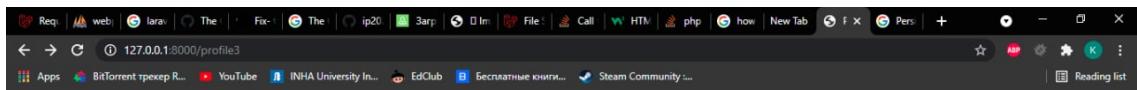
Detailed Information

Email example@email.com	Phone Number 99999999	Date of birth 00/00/0000	Passport Info AA00000	Postal Code 100XXX	City Choose your region ▾
----------------------------	--------------------------	-----------------------------	--------------------------	-----------------------	------------------------------

Save

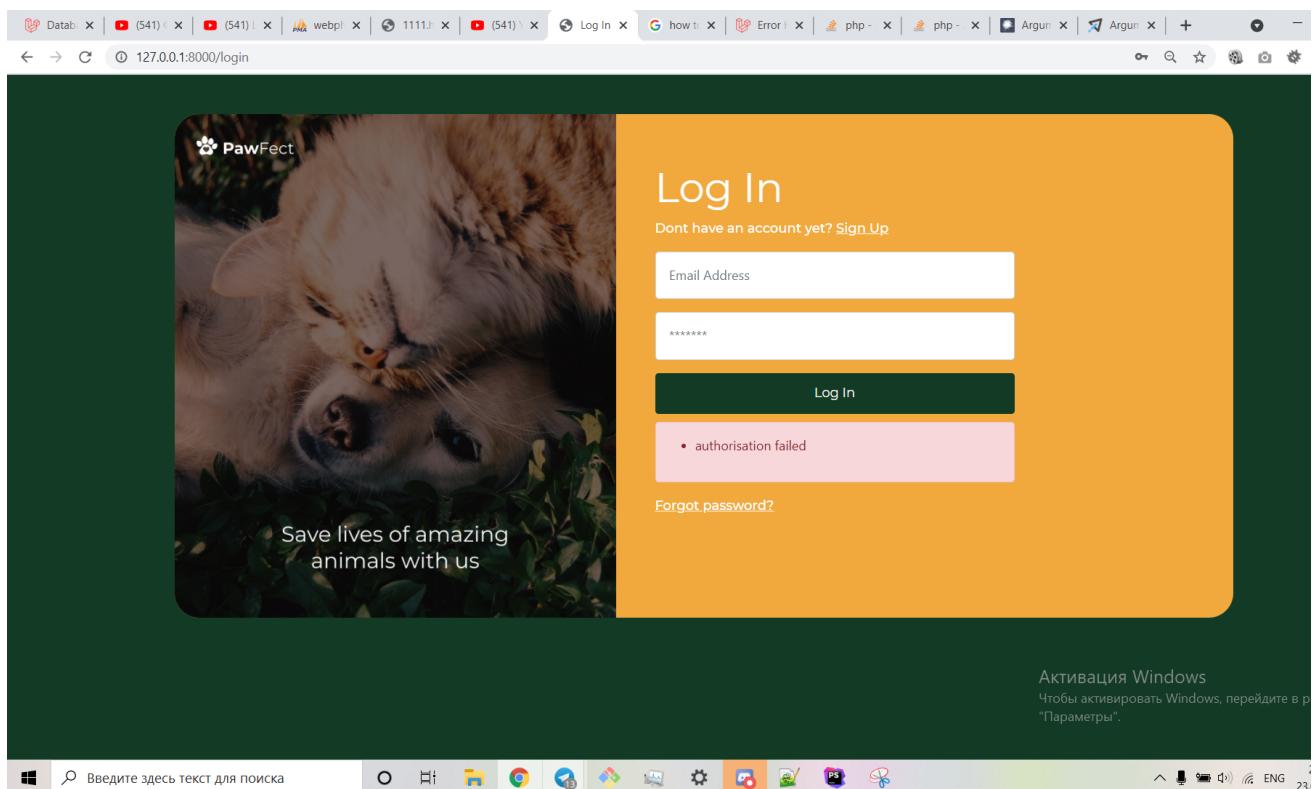
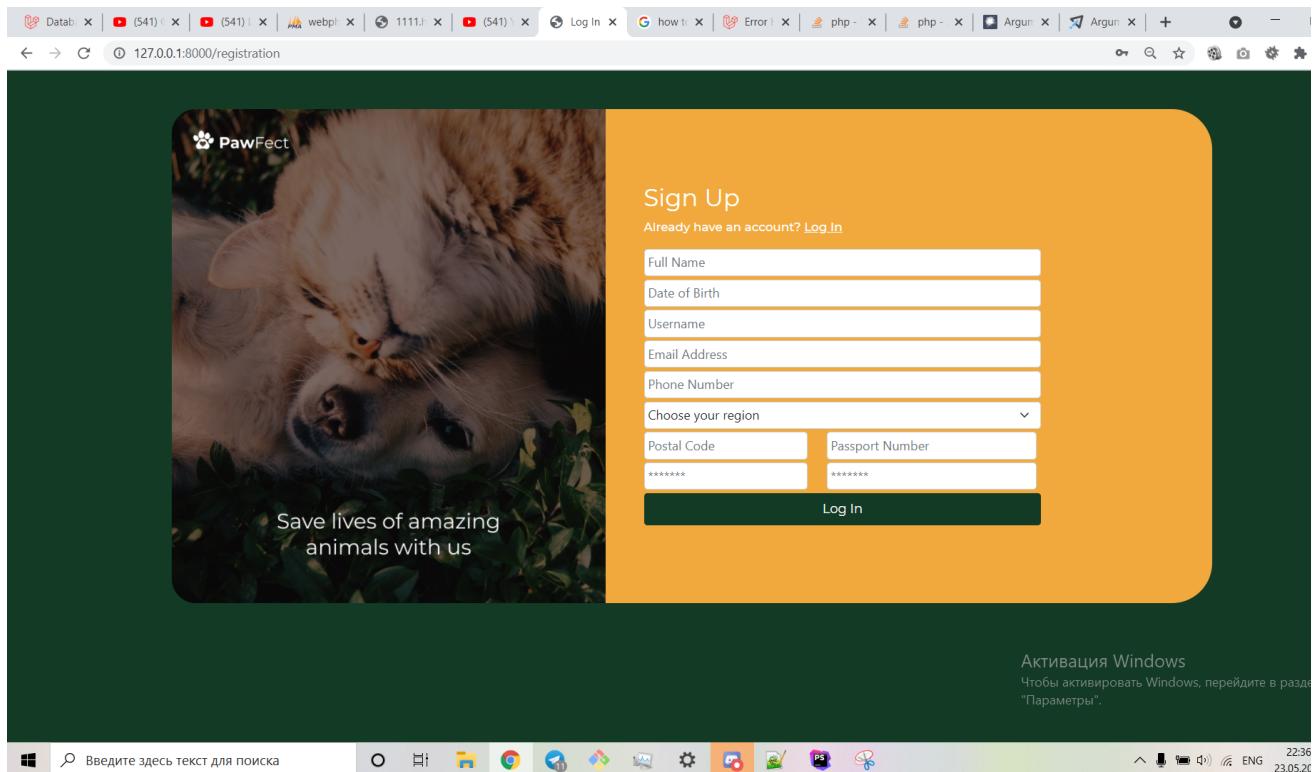


As well as options to edit his personal info

A screenshot of a web browser window showing an edit post form. The address bar shows the URL '127.0.0.1:8000/editPost?'. The form fields are: Name (Kleo), Breed (African clown), Type (Cat), Age (2), Price (25\$), and Description (Enter new description). A 'Save Changes' button is at the bottom.

As well as seeing all the post he/she created and clicking to the details at post change post details

Sign Up and Login



We created a form that accepts all necessary data about user(username password city etc) and stores it into database using auth. The form has laravel validation and also regex validation(for phone, username etc).

Hosting

Website list | databases-auth.000webhost.com | 000webhost File Manager

000webhost powered by HOSTINGER IPawfect > public_html Go Premium

Name ▾ Size Date Permissions

.htaccess 0.2 kB 2021-05-23 15:35:00 -rw-r--r--

/ / tmp

File Manager

Build a Website | databases-auth.000webhost.com | 000webhost File Manager

My Sites Power Store Help 000webhost powered by HOSTINGER Upgrade 20potatom02

ipawfect Current site | Free

Home Tools Set Web Address File Manager Database Manager Email Manager

All websites > ipawfect

zyro
Build a free website with Zyro
Our powerful website builder will get your business online today.
Build a free website

WordPress
Create powerful & beautiful website using WordPress.
Install >

File Manager
Upload your own prebuilt website. For Advanced users.
Upload >

R ENG 23:28 23.05.2021

R ENG 23:29 23.05.2021

Summarizing

Overall, The UX/UI part including the whole Frontend development was done by Nodira Tillayeva(U1910041).

The registration and Authentication part and Hosting was done by Oleksii Melnyk (U1910039).

Admin Page and Search Functions were done by Javokhir Shoyusupov (U1910056). Landing Page backend, Hosting, Category Sorting and displaying all the post were done by Robert Khan (U1910059).

Profiles and Posts Editing and Creation features were developed by Kobiljon Komilov (U1910092).