Workshop: Petstagram Full Description

We will develop a complete Django project called **"Petstagram"** throughout this module. The project will cover the following **functionalities**: user **registration**, **login,** and **logout**; each user can **add pets** to their profile and **upload pet photos**; a user can **view all photos** of pets, open **details**, can **like** and **comment** on a photo. Each user can **edit** and **delete their photos** and **pet** information.

You can directly dive into the app here: https://softuni-petstagram.azurewebsites.net/

1. URLs

Home Page: http://127.0.0.1:8000/

Registration Page: http://127.0.0.1:8000/accounts/register/

Login Page**:** http://127.0.0.1:8000/accounts/login/

Profile Details Page: http://127.0.0.1:8000/accounts/profile/<int:pk>/

Profile Edit Page: http://127.0.0.1:8000/accounts/profile/<int:pk>/edit/

Profile Delete Page: http://127.0.0.1:8000/accounts/profile/<int:pk>/delete/

Pet Add Page: http://127.0.0.1:8000/pets/add/

Pet Details Page: http://127.0.0.1:8000/pets/<str:username>/pet/<slug:petname>/

Pet Edit Page: http://127.0.0.1:8000/pets/<str:username>/pet/<slug:petname>/edit/

Pet Delete Page: http://127.0.0.1:8000/pets/<str:username>/pet/<slug:petname>/delete/

Photo Add Page: http://127.0.0.1:8000/photos/add/

Photo Details Page: http://127.0.0.1:8000/photos/<int:pk>/

Photo Edit Page: http://127.0.0.1:8000/photos/<int:pk>/edit/

2. Models

User Profile

- **Username**
- **Password**
- **Email Address**

Note: the user registers with the information provided above (all required)

- **First Name** – it should have **at least 2 chars**, **max - 30 chars**, and should consist **only of letters**
- **Last Name** - it should have **at least 2 chars**, **max - 30 chars**, and should consist **only of letters**
- **Profile Picture** – the user can link their picture using a **URL**
- **Gender** – character field with **choices**, the user can choose one of the following: "**Male**", "**Female**", and "**Do not show**"

Other:

- When the **user has been deleted,** all their **photos**, **pets**, **comments,** and **likes** should be **deleted too**

Pet

The user **must** provide the following information when adding a pet:

- **Name** - it should consist of a **maximum of 30 characters**.
- **Personal Pet Photo** - the user can link a picture using a URL

The user **may** provide the following information when adding a pet:

- **Date of Birth** - pet's day, month, and year of birth

Other:

- **Slug** - a slug automatically generated using the **pet's name and the pet's id, separated by a "-"** (space).
- When a pet is deleted **NONE** of its **photos should be deleted**

Pet's Photo

The user **must** provide the following information when uploading a pet's photo:

- **Photo** - the user can **upload** a picture **from storage**, the **maximum size** of the photo can be **5MB**

The user **may** provide the following information when uploading a pet's photo:

- **Description** - a user can write any description of the photo, it should consist of a **maximum of 300 characters** and a **minimum of 10 characters**
- **Location** - it should consist of a **maximum of 30 characters**
- **Tagged Pets** - the user can tag **none, one, or many of all pets**. There is **no limit** on the number of tagged pets

Other:

- Date of publication - when a picture is **added or edited**, the date of publication is **automatically generated**
- When a **photo is deleted** all its **comments** and **likes** should be **deleted too**

Comment

The user **must** provide the following information when commenting on a picture:

- **Comment Text** - it should consist of a **maximum of 300 characters**

Other:

- **Date and Time of Publication** - when a comment is created (only), the date of publication is **automatically generated**

Like

- Should **connect the user to the photo** so it's known which user which photo has liked

3. Structure

Navigation Bar

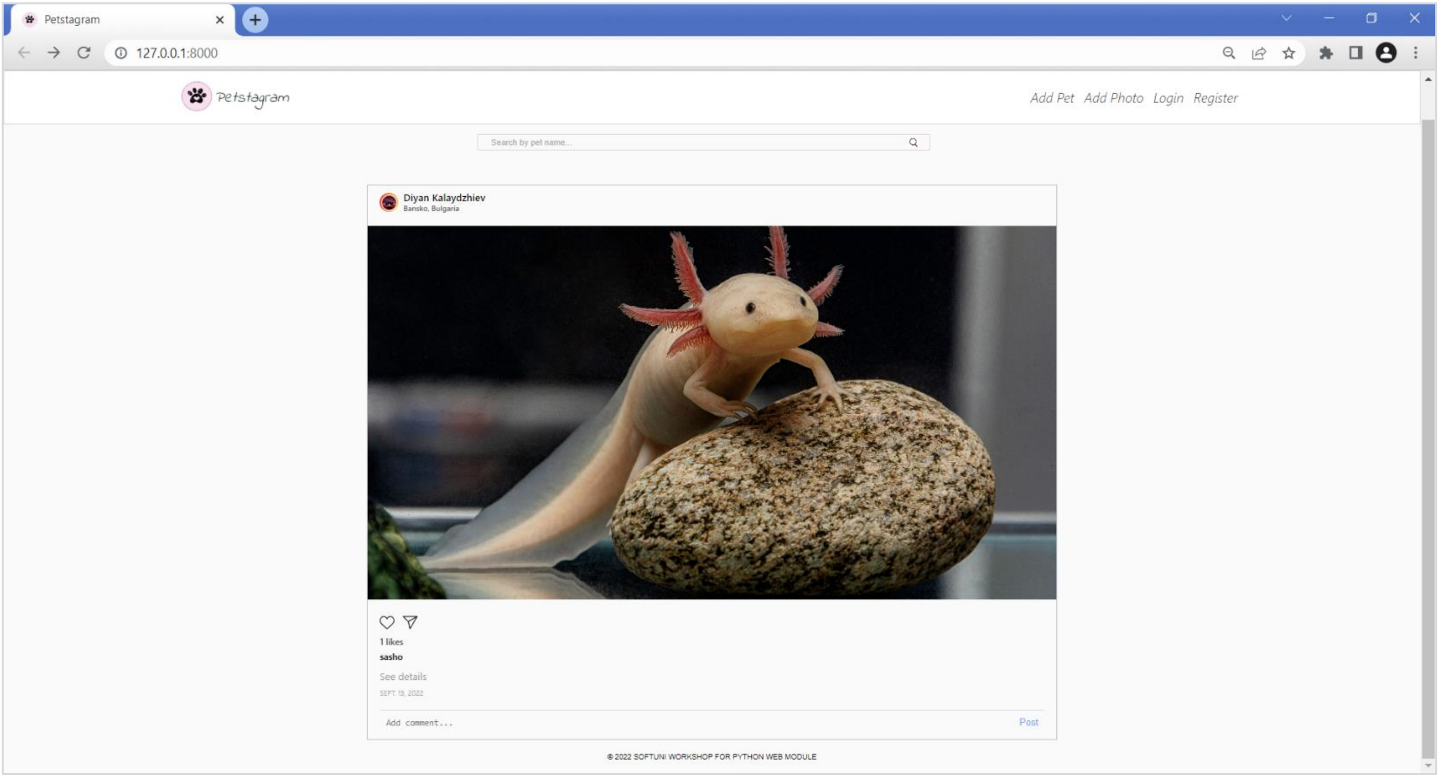
At the top of each page there is a navigation bar that has the following buttons:

- **Petstagram** button that leads to the home page
- **Add Pet** button that leads to the pet add page
- **Add Photo** button that leads to the photo add page
- **Login** button that leads to the login page (visible only to unauthenticated users)
- **Register** button that leads to the register page (visible only to unauthenticated users)
- **Profile** button that leads to the user profile details page (visible only to authenticated users)
- **Logout** button that logs out the logged-in user (visible only to authenticated users)

Home Page

The home page consists of a search bar, where users can **search for pet photos by a pet name**, and pet photos (posts). When there **are pets photos**, the user sees posts, each consisting of:

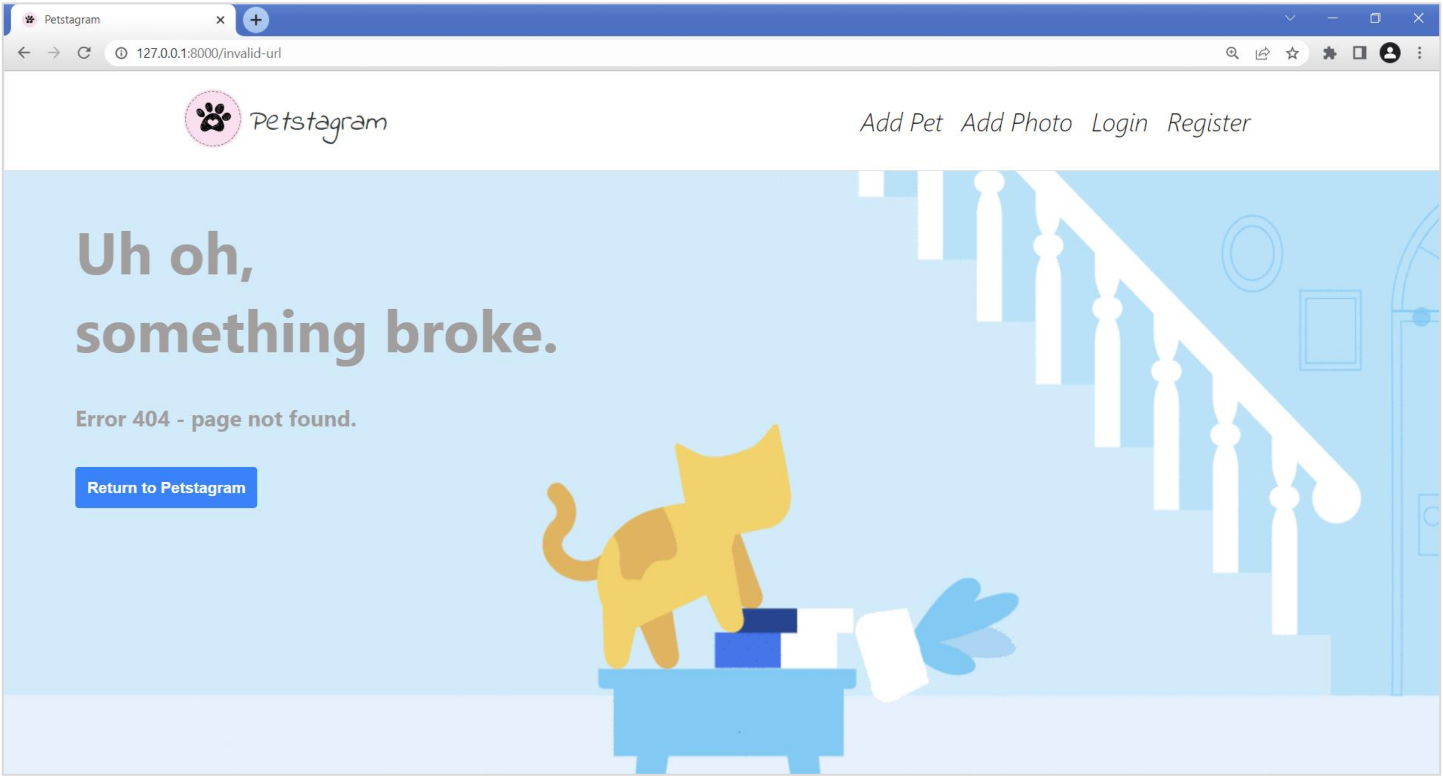
- The owner's **name** (or **username** if first/ last name is **NOT** provided) and their **profile picture**
- The **location** (if one is added)
- The pet **photo**
- **Like** button
  - When the user clicks on the **white heart button** they **like the picture**, and the **heart turns red** - it is considered **one like** and should be updated in the counter (below)
  - When the user **clicks on an already liked picture** (on the read heart button) they **unlike** the picture, and the **heart turns white** - the **one like should be subtracted** from the total count (below)
  - Each user can do the like/unlike cycle an unlimited number of times
- **Share** buttons
  - When the user clicks on the **share button** the **URL in the address bar is copied** into the Clipboard.
- The total **count of likes** for the photo
- The **tagged pets** (if any are added) - if there is **more than one pet** tagged, they must be shown on **different lines**
- "**See Details**" button that leads to the **photo detail page**
- **Date** of publication or edition
- **Comment Form** that users can use to add comments to the photo



404 Page Not Found

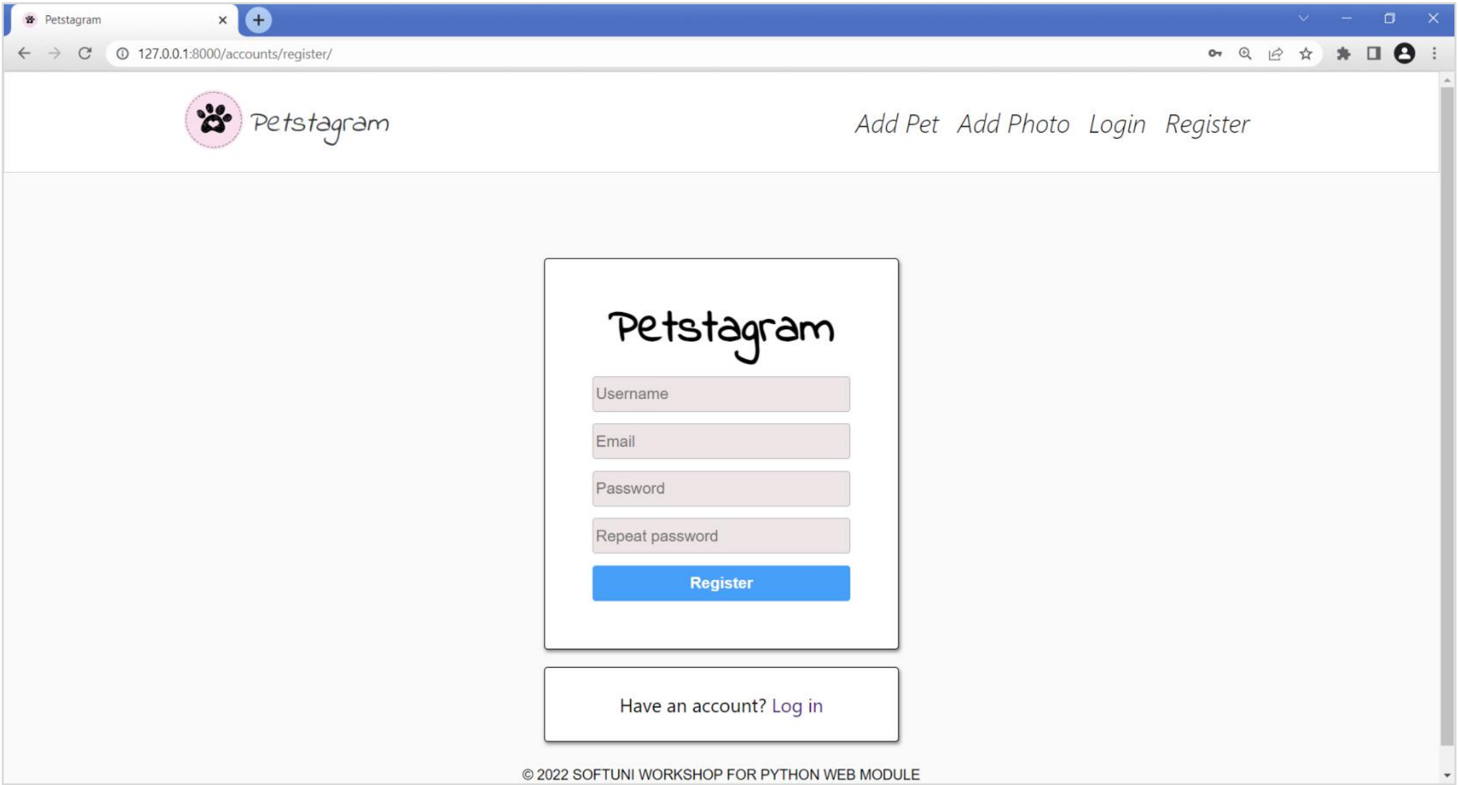
When the user tries to reach an **invalid path**, the app should **load the 404 template**. It consists of:

- Navigation bar
- "**Uh oh, something broke. Error 404 - page not found.**" text
- "**Return to Petstagram**" button which leads to the home page.

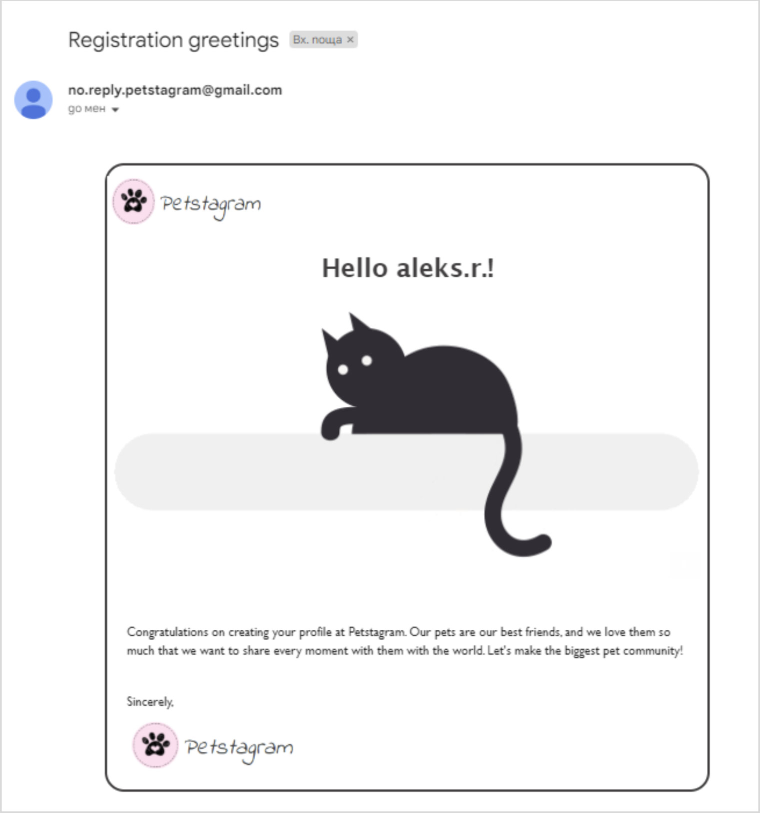


Registration Page

The visible fields when **creating** a profile are "**Username**", "**Email**", and "**Password**" fields. When the button "**Register**" is clicked, the **profile is created**, and it **redirects to the Login page**. There is also a "**Log in**" button that leads to the **Login page**:



When the **profile is created**, the app **automatically sends** a registration greeting **email**:



Registration greetings

no.reply.petstagram@gmail.com
до мен

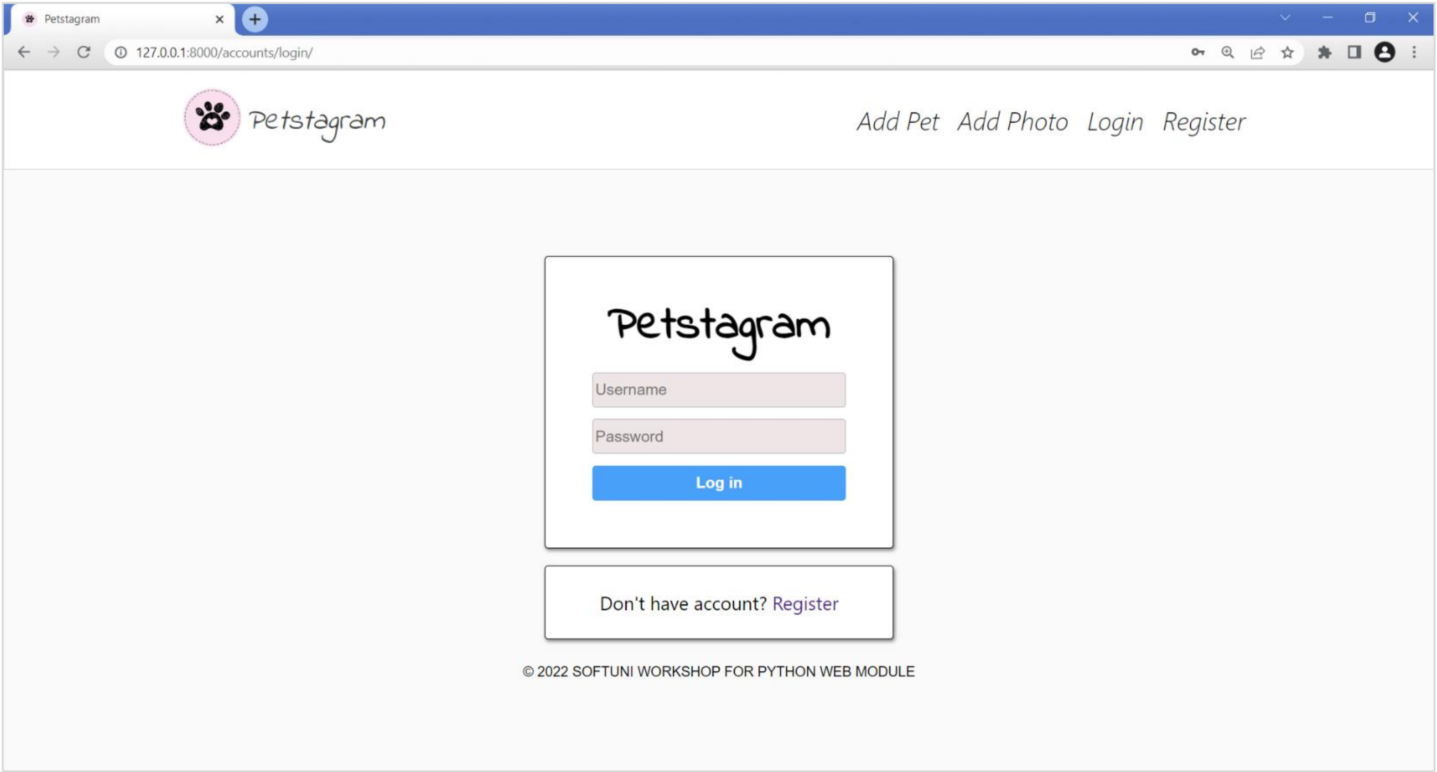
Hello aleks.r.!

Congratulations on creating your profile at Petstagram. Our pets are our best friends, and we love them so much that we want to share every moment with them with the world. Let's make the biggest pet community!

Sincerely,

Login Page

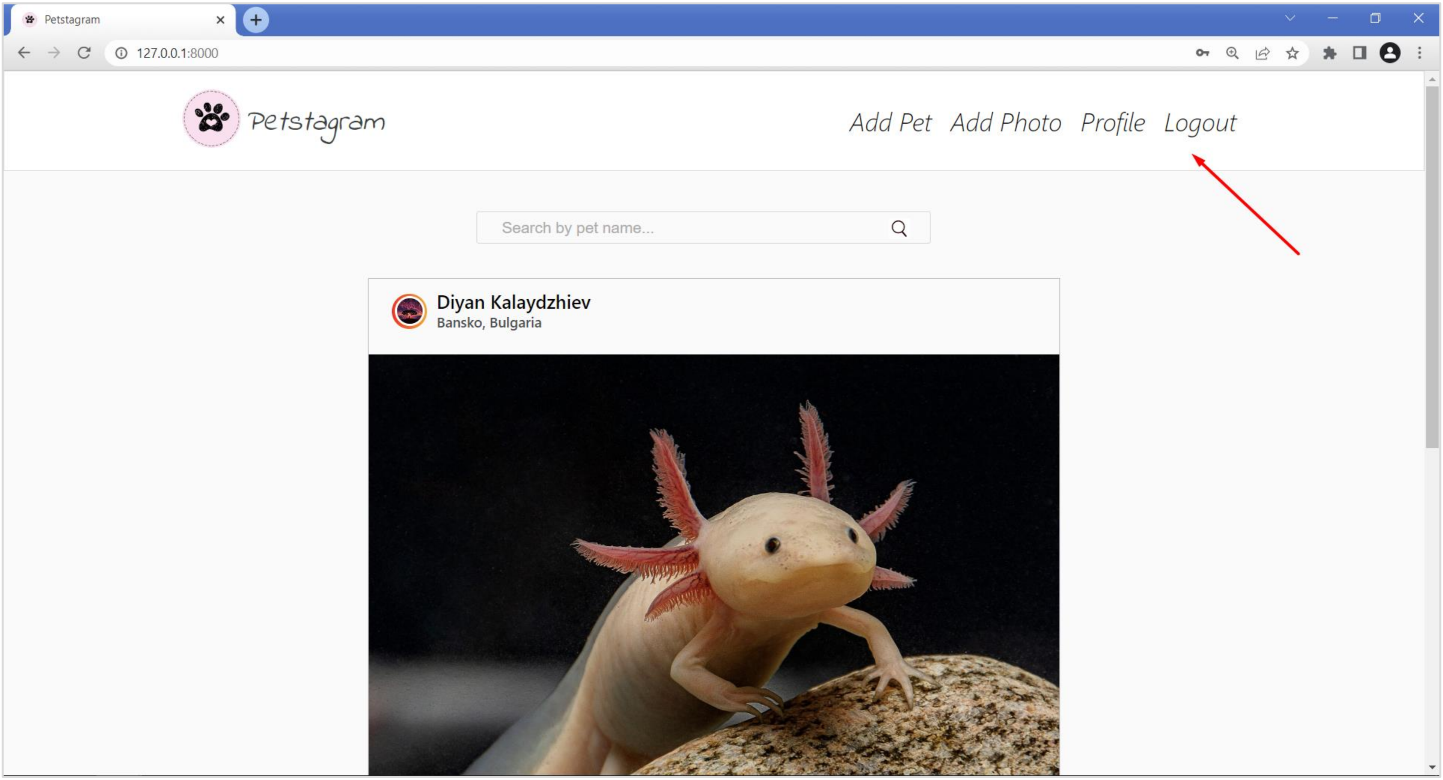
The visible fields when **login** to a profile are "**Username**" and "**Password**". When the button **"Log in"** is clicked, the user is logged in and **redirected to the Home page**. There is also a "**Register**" button that **leads to the register page**:



Don't have account? Register

© 2022 SOFTUNI WORKSHOP FOR PYTHON WEB MODULE

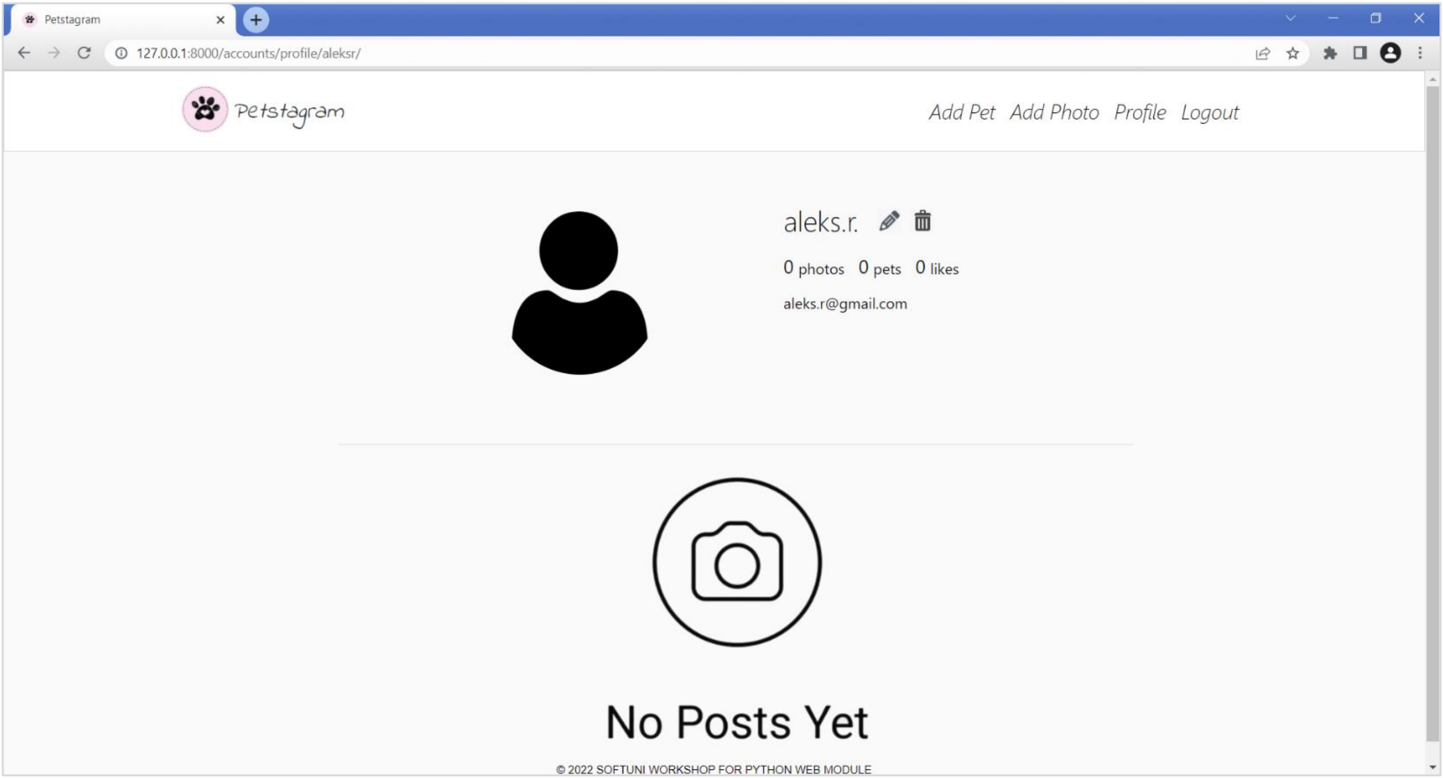
When the **user is logged in**, they can see the buttons "**Profile**" and "**Logout**" in the navigation bar:



Profile Details Page

The profile details page consists of:

- The user profile **picture**; if the user has **NOT** added a profile picture, the **default image** should be shown
- The user **username**
- A **edit** profile button and a **delete** profile button
- The total **number of photos** added from the user
- The total **number of pets** added from the user
- The total **number of likes** collected by the user's photos
- The **email address** of the user
- A "**No Posts Yet**" text when the user has **NOT** uploaded any photos yet; or the photos themselves.



aleks.r.

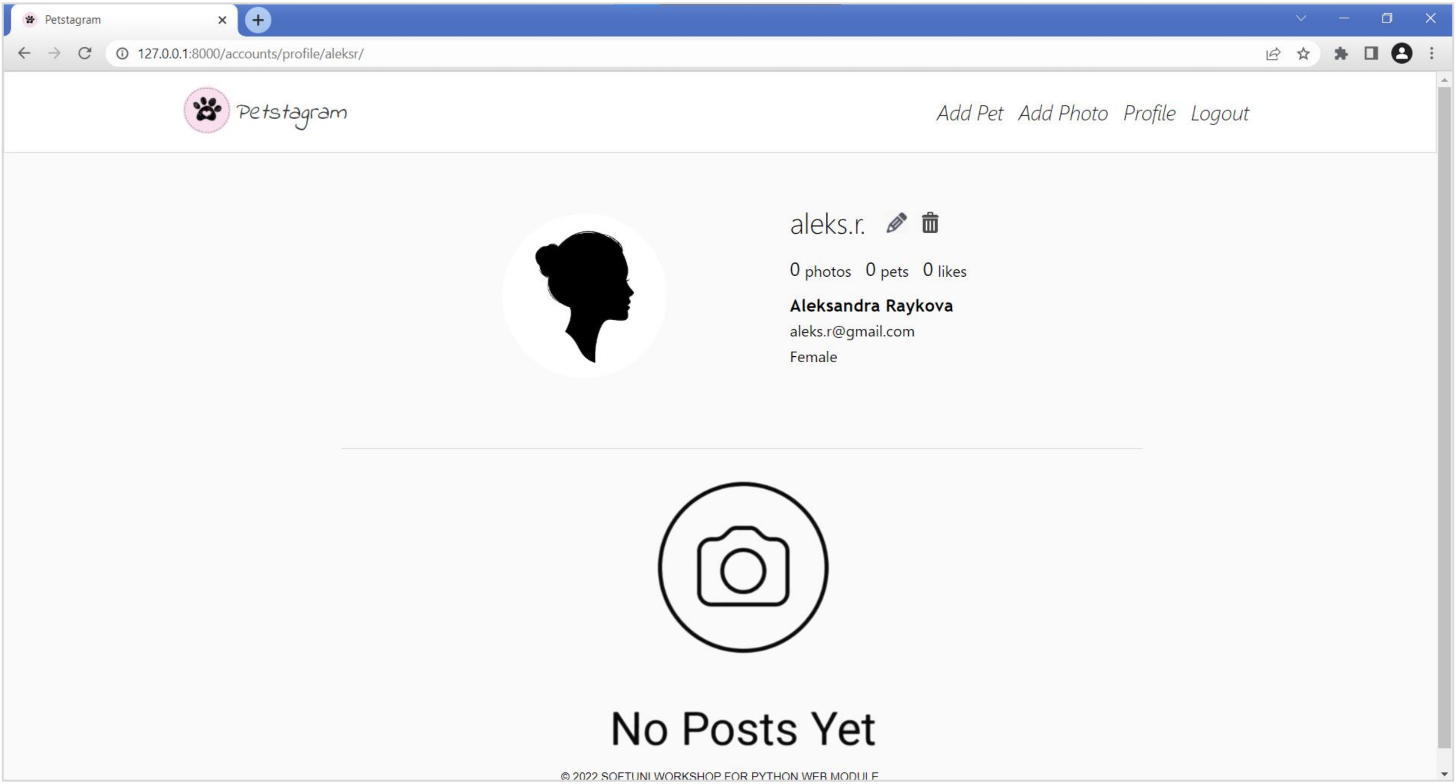
0 photos 0 pets 0 likes

aleks.r@gmail.com

No Posts Yet

© 2022 SOFTUNI WORKSHOP FOR PYTHON WEB MODULE

You can add a **first name** and a **last name**, **gender** ("Male" or "Female"), and a **personal image**:



aleks.r.

0 photos 0 pets 0 likes

Aleksandra Raykova

aleks.r@gmail.com

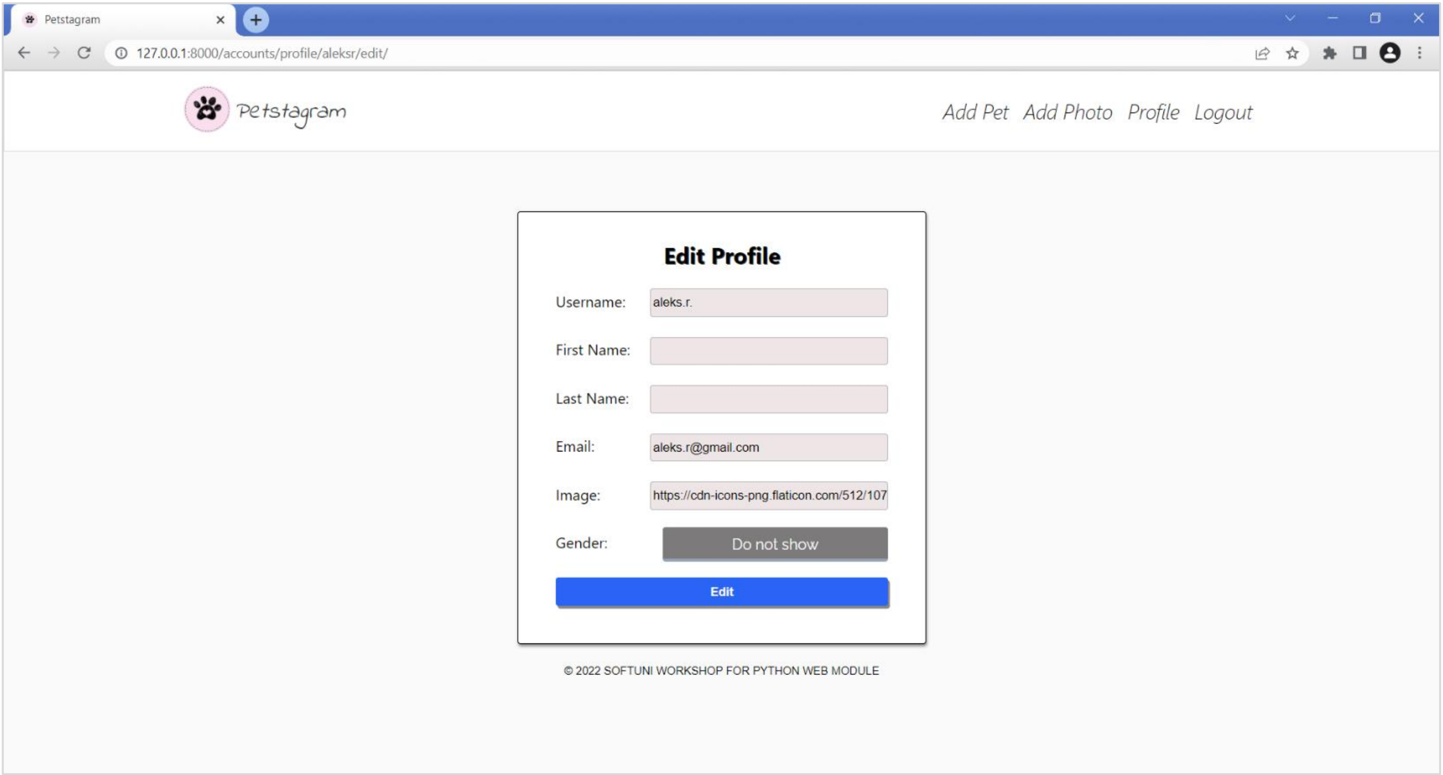
Female

No Posts Yet

© 2022 SOFTUNI WORKSHOP FOR PYTHON WEB MODULE

Profile Edit Page

The visible fields when **editing** the profile are "**Username**" and "**First Name**", "**Last Name**", "**Email**", "**Image**" and "**Gender**". When the button "**Edit**" is clicked, the **profile is edited**, and redirected to the **profile details page**:



Edit Profile

Username: aleks.r.

First Name:

Last Name:

Email: aleks.r@gmail.com

Image: https://cdn-icons-png.flaticon.com/512/107

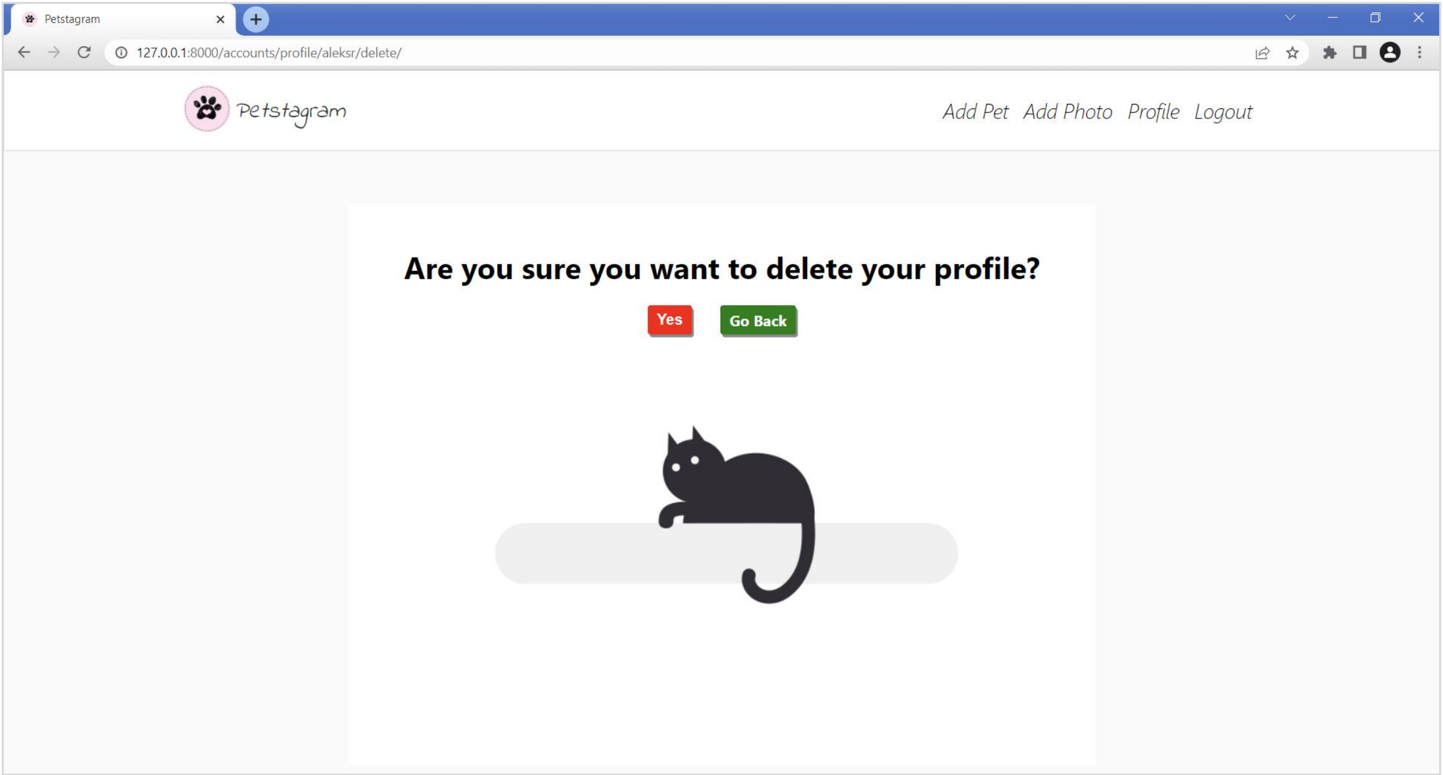
Gender: Do not show

© 2022 SOFTUNI WORKSHOP FOR PYTHON WEB MODULE

Profile Delete Page

On the profile delete page there is a "**Are you sure you want to delete your profile?**" text and a "**Yes**" and " **Go Back**" buttons. If the user clicks on the "**Yes**" button the **profile is deleted**, **and all the user's photos**, **pets**, **comments** and **likes**, and the user is **redirected to the Home page**. If the user clicks on the "**Go back**" button, they are **redirected to the**

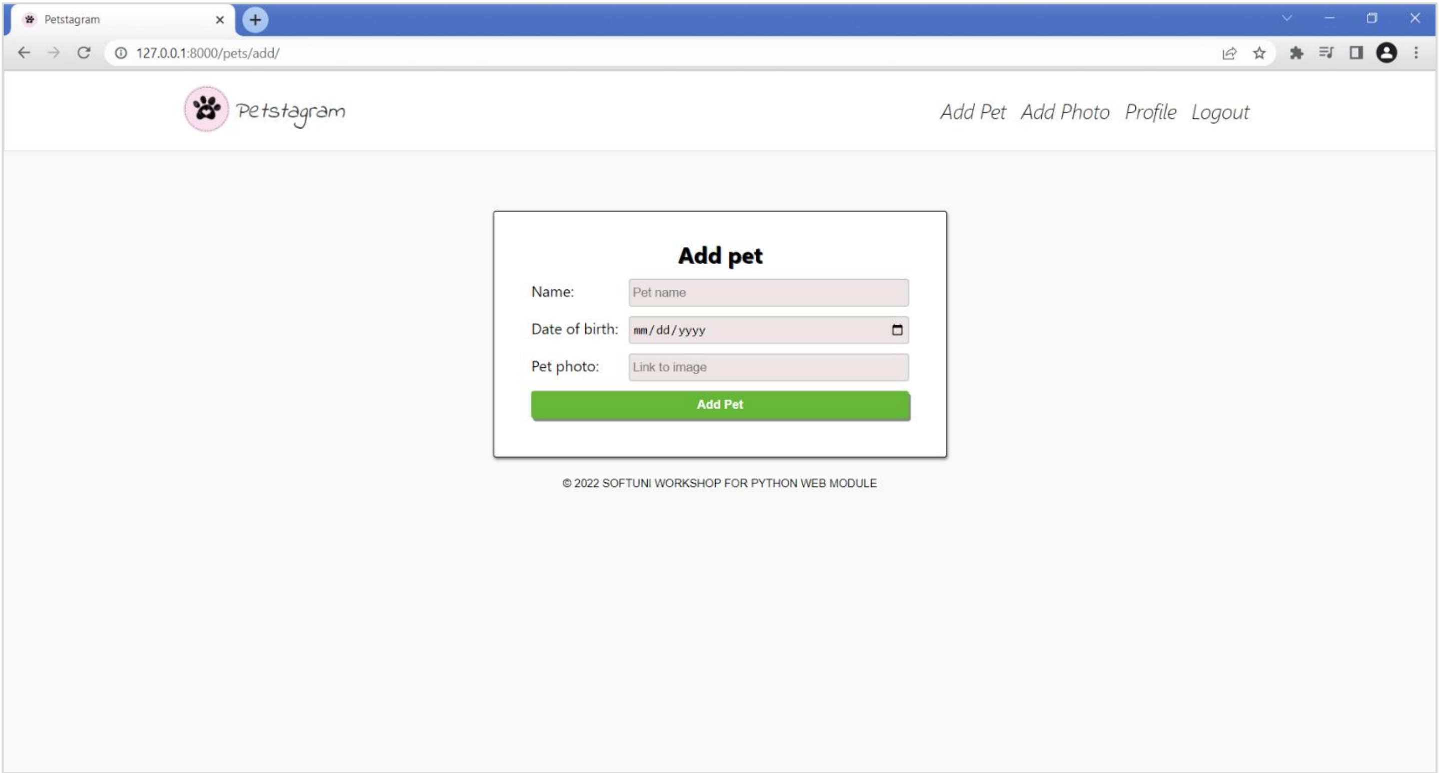
profile details page:



Are you sure you want to delete your profile?

Pet Add Page

The **visible fields** when adding a pet are "**Pet name**", "**Date of birth**", and "**Link to image**":

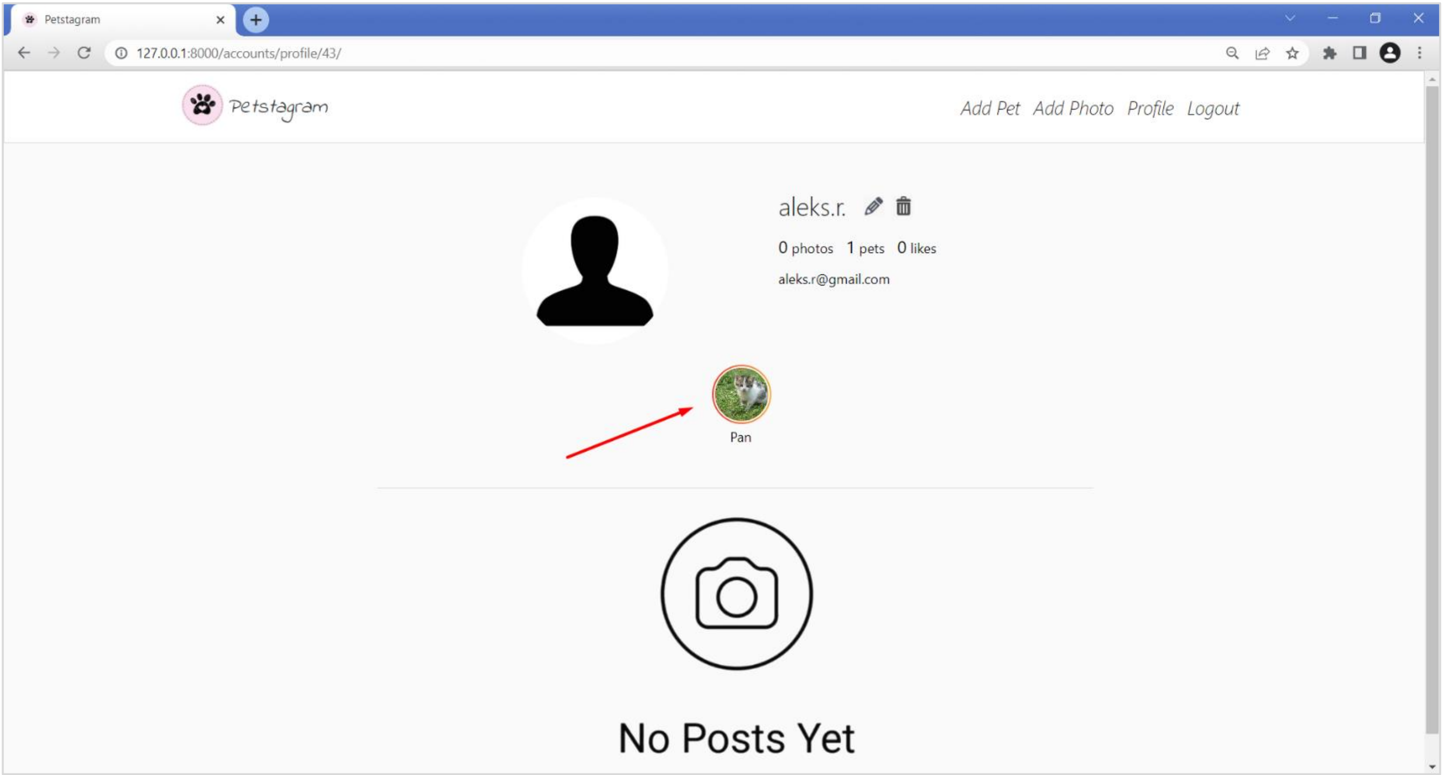


Add pet

Name:	Pet name
Date of birth:	mm/dd/yyyy
Pet photo:	Link to image

© 2022 SOFTUNI WORKSHOP FOR PYTHON WEB MODULE

When the user **successfully adds a pet**, they are **redirected to their profile details page**, where their pets are visible:

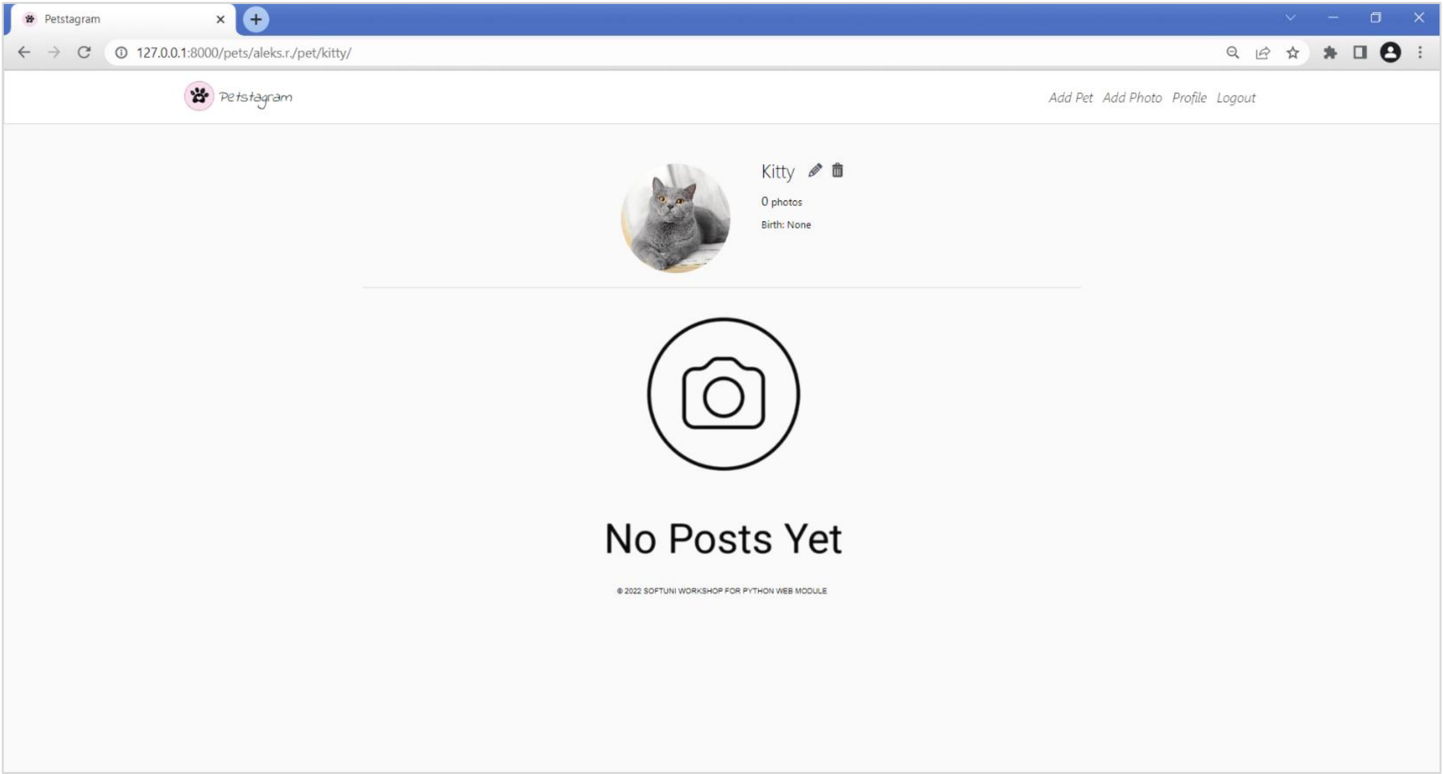


Pet Details Page

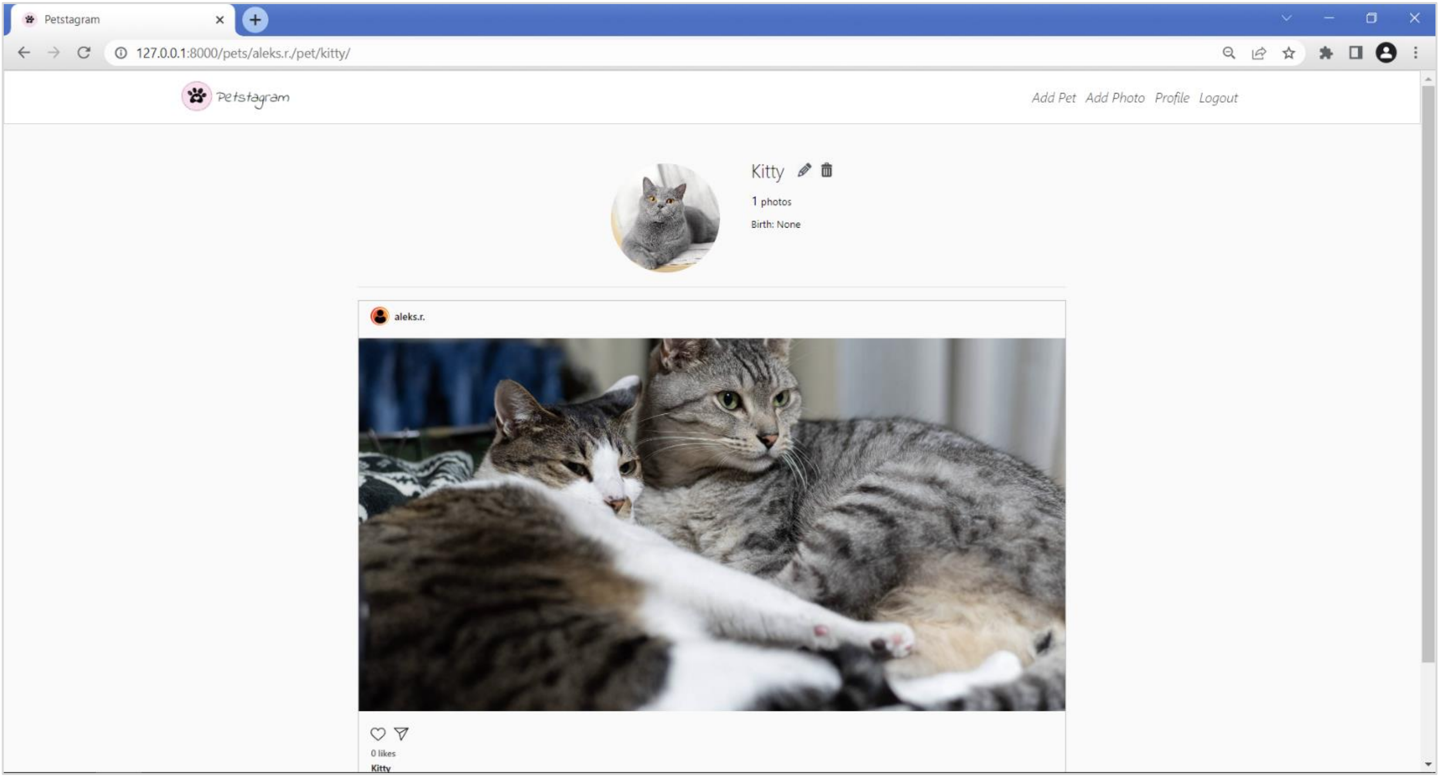
The pet details page consists of:

- The pet **picture**
- The pet **name**
- **Edit** and **delete** buttons (visible only to their owner)
- The total **number of photos** that the pet was tagged on
- The **birthdate** of the pet (if one is added)
- A "**No Posts Yet**" text when the user has **NOT** uploaded any photos yet;

The **edit** button leads to the **pet edit page**. The **delete** leads to the certain **pet delete page**.

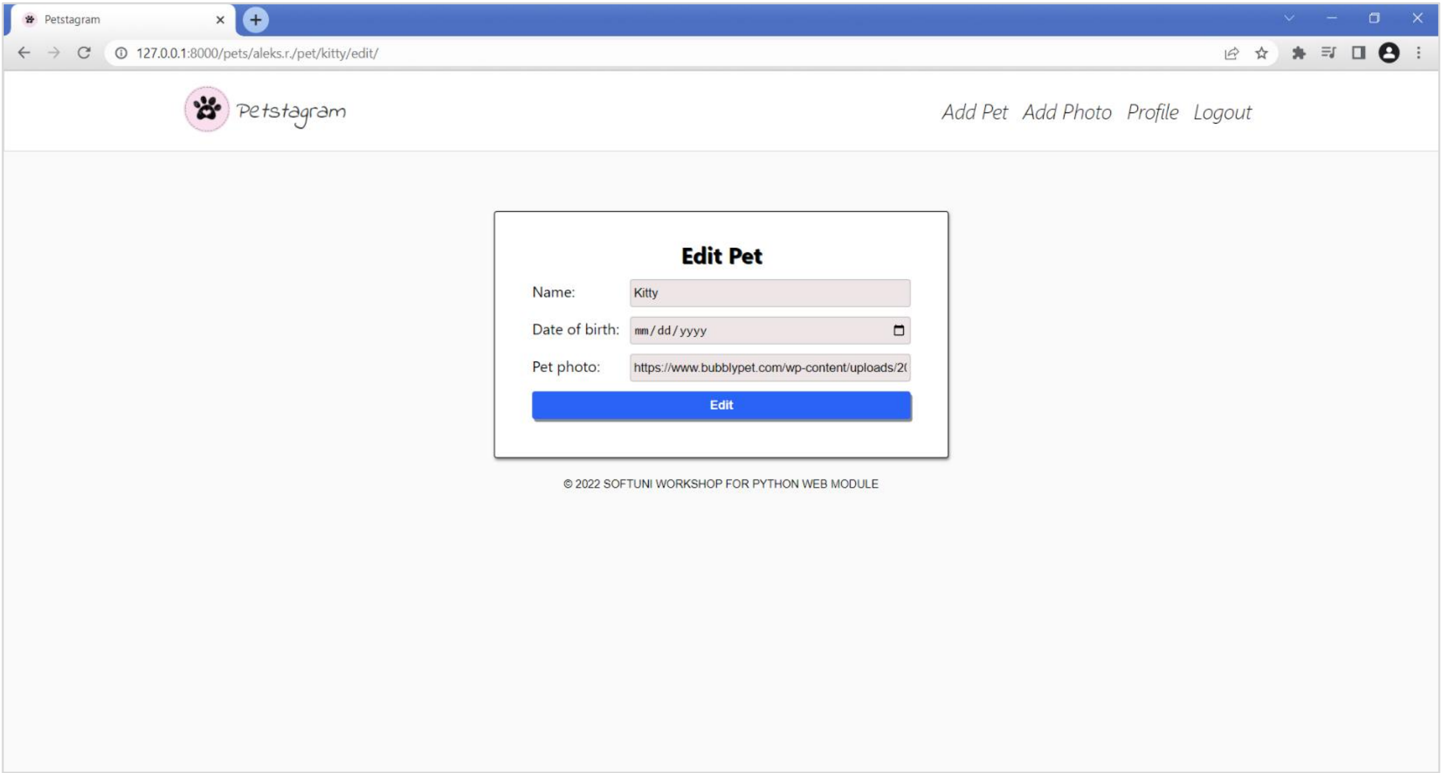


When the pet is **tagged** all **their photos are shown** as posts (same as on the Home page):



Pet Edit Page

The visible fields when editing pet info are "**Pet name**", "**Date of birth**", and "**Link to image**". When you click on the "**Edit**" button, the **pet info is updated**, and the user is redirected to the **pet details** page:



Edit Pet

Name: Kitty

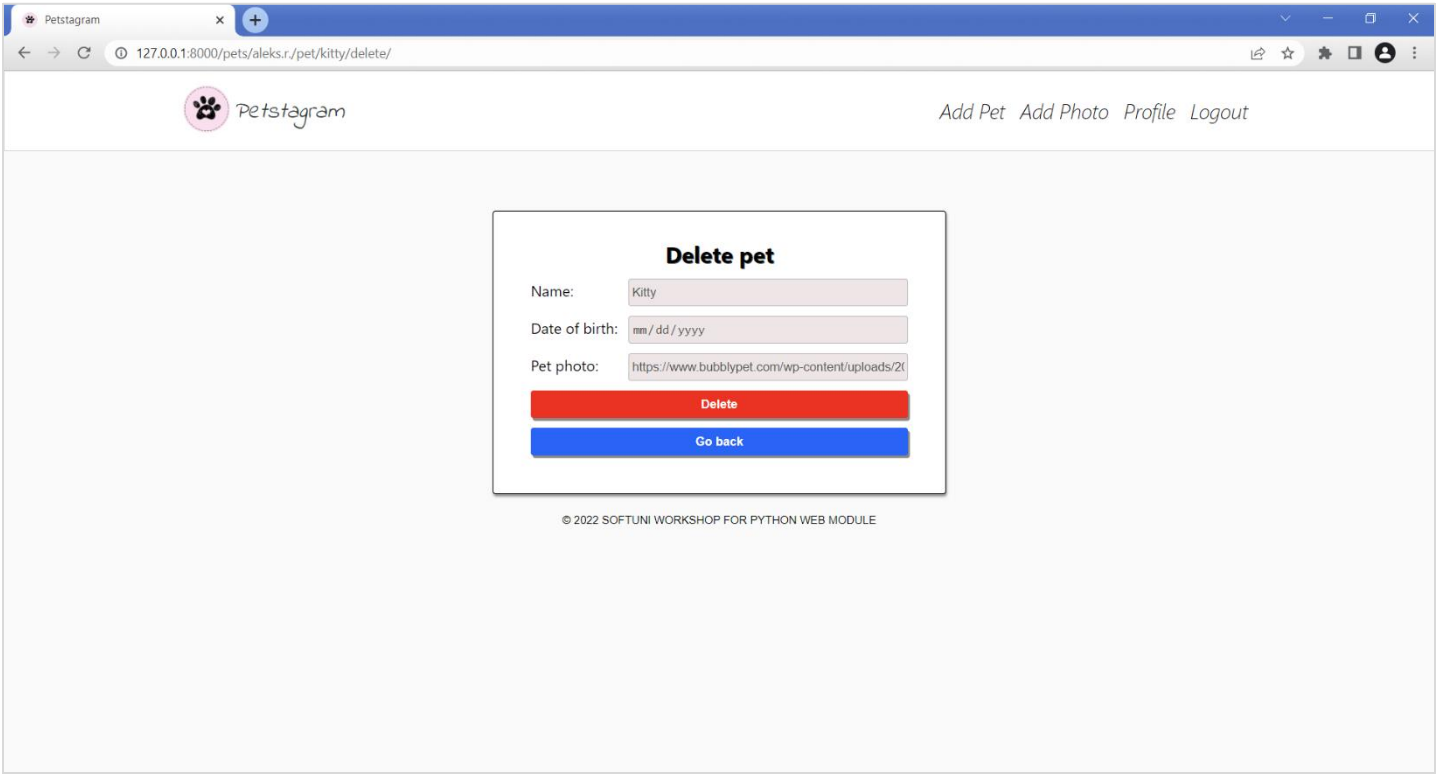
Date of birth: mm/dd/yyyy

Pet photo: https://www.bubblypet.com/wp-content/uploads/2(

© 2022 SOFTUNI WORKSHOP FOR PYTHON WEB MODULE

Pet Delete Page

All pet info fields should be **visible** and **disabled**. If the user clicks on the "**Delete**" button the **pet is deleted** (but all its photos are **NOT** deleted), and the user is **redirected to the profile details page**. If the user clicks on the "**Go back**" button, they are **redirected to the pet details page**:



Delete pet

Name: Kitty

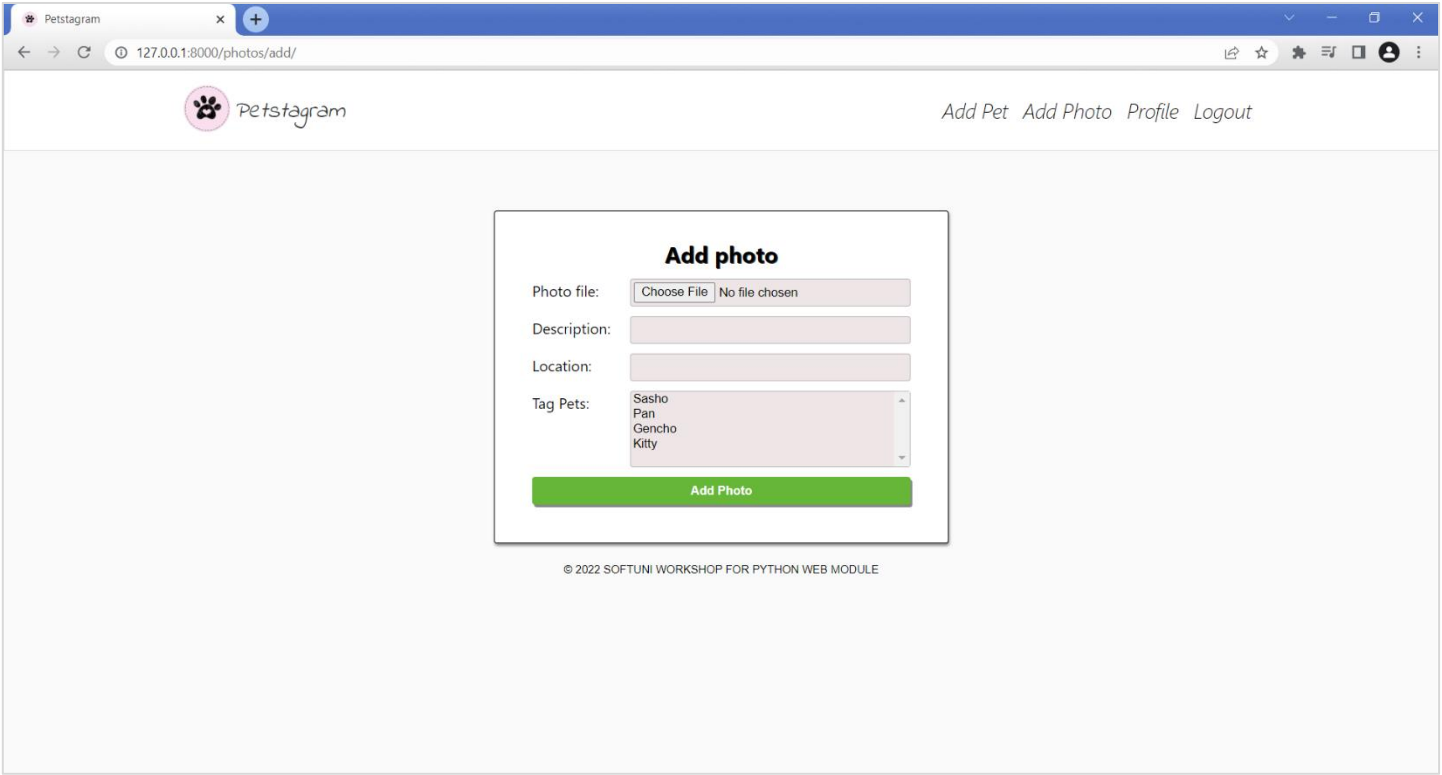
Date of birth: mm / dd / yyyy

Pet photo: https://www.bubblypet.com/wp-content/uploads/2(

© 2022 SOFTUNI WORKSHOP FOR PYTHON WEB MODULE

Photo Add Page

The **visible fields** when adding a new photo are "**Pet Image**", "**Description**", "**Tag Pets**", and "**Location**". When the user successfully adds a photo, they are **redirected to the Home page**:



Add photo

Photo file: Choose File No file chosen

Description:

Location:

Tag Pets: Sasho
Pan
Gencho
Kitty

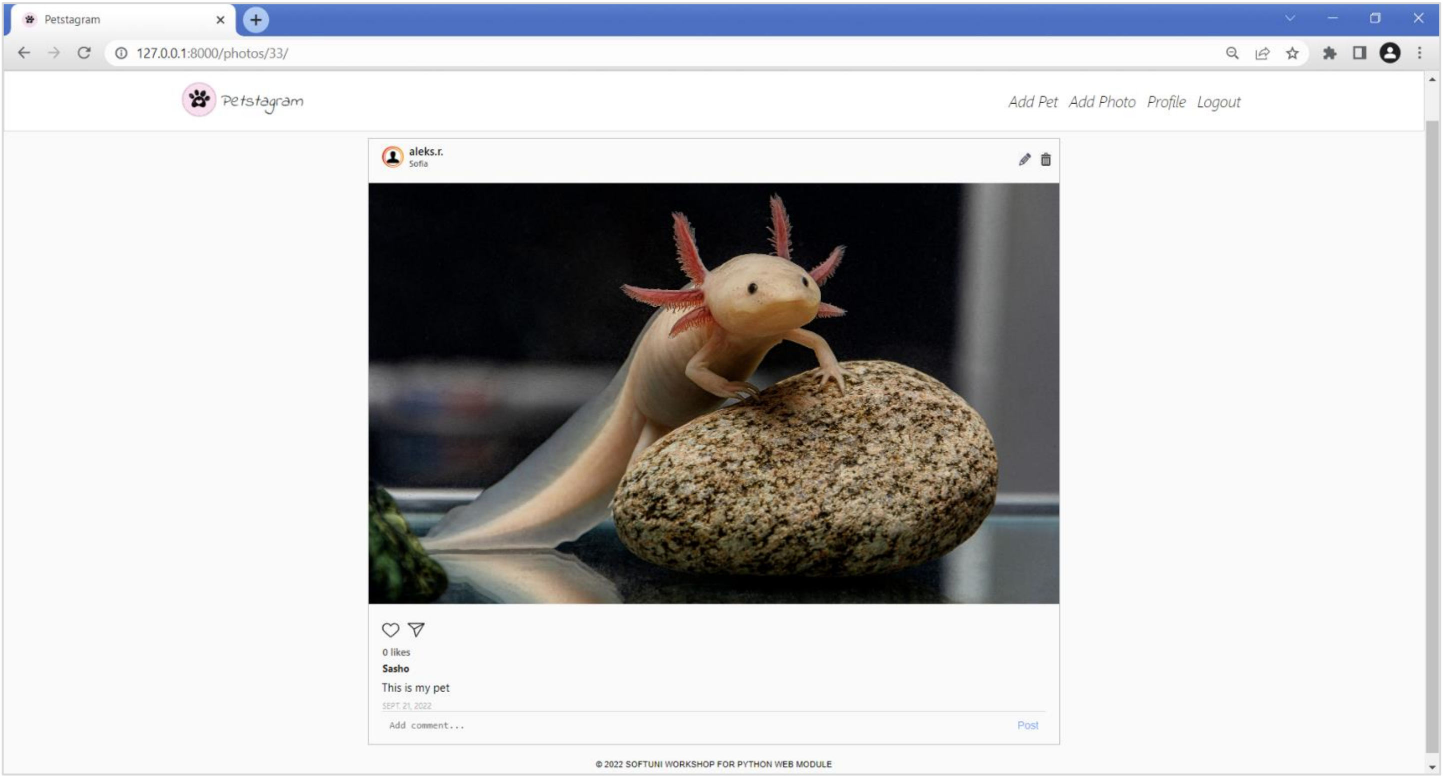
© 2022 SOFTUNI WORKSHOP FOR PYTHON WEB MODULE

Photo Details Page

When there **are pets photos**, the user sees **posts**, each consisting of:

- The **owner's name** (or **username**) and their **profile picture**
- **Edit and delete buttons** (visible only to the owner of the photo)
- The **location** (if one is added)
- The pet **photo**
- **Like** and **Share** buttons
- The total **count of likes** for the photo
- The **tagged pets** (if any are added) - if there is **more than one pet** tagged, they should be must be shown on **different lines**
- Photo **description** (if one is added)
- **Date** of publication (or edition)
- **Comment** from users, **sorted** by the **date and time of publication in descending order** (the last comment should appear first, the first comment should appear last in the comment section)
- **Comment Form** used to add comments to the photo

The **edit** button redirects to the **edit photo page**. The **delete button directly deletes the photo**.

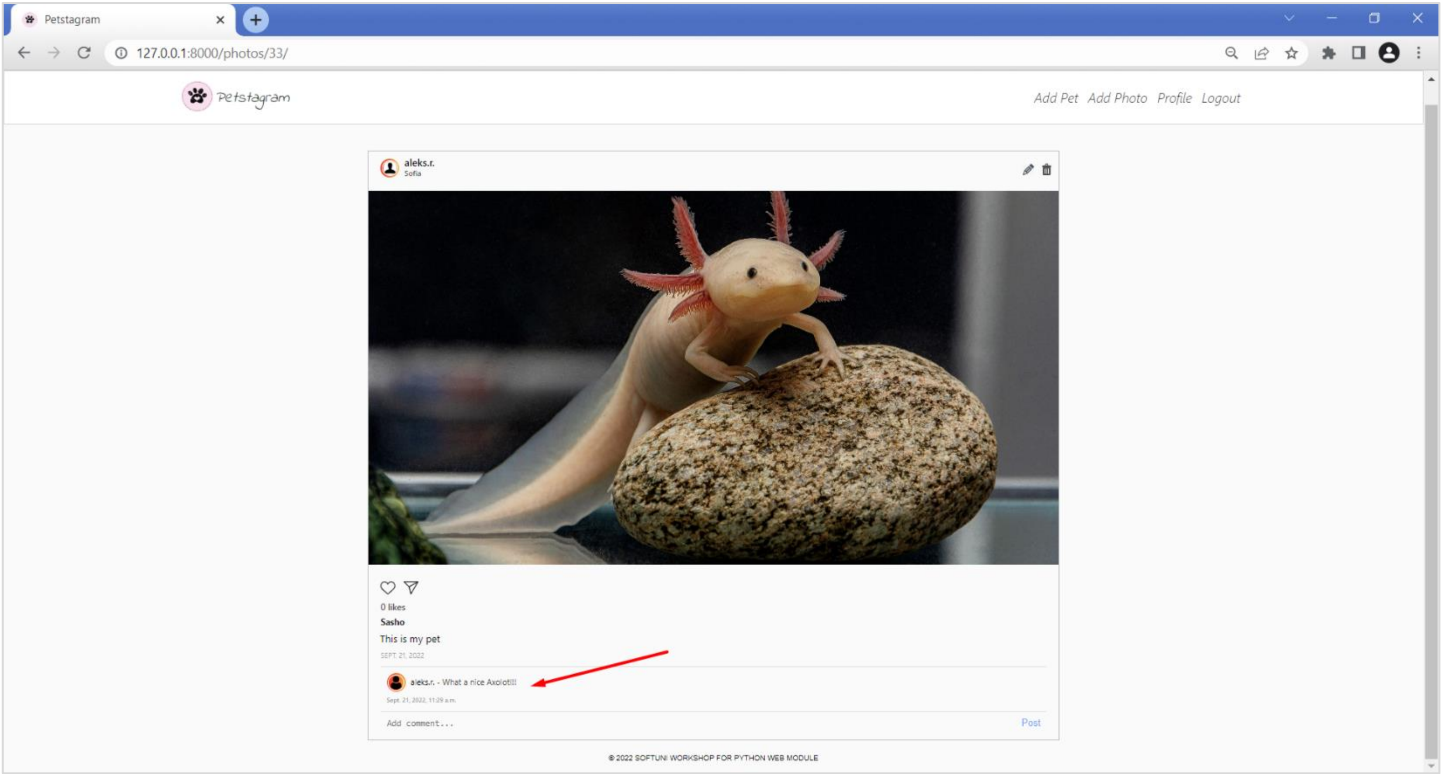


aleks.r.
Sofia

0 likes
Sasho
This is my pet
SEPT. 21, 2022

© 2022 SOFTUNI WORKSHOP FOR PYTHON WEB MODULE

When a **comment is added**, it should be **visible** on the photo details page, **sorted** from the **oldest** (first) **to the latest** (last):



aleks.r.
Sofia

0 likes

Sasho

This is my pet

SEPT. 21, 2022

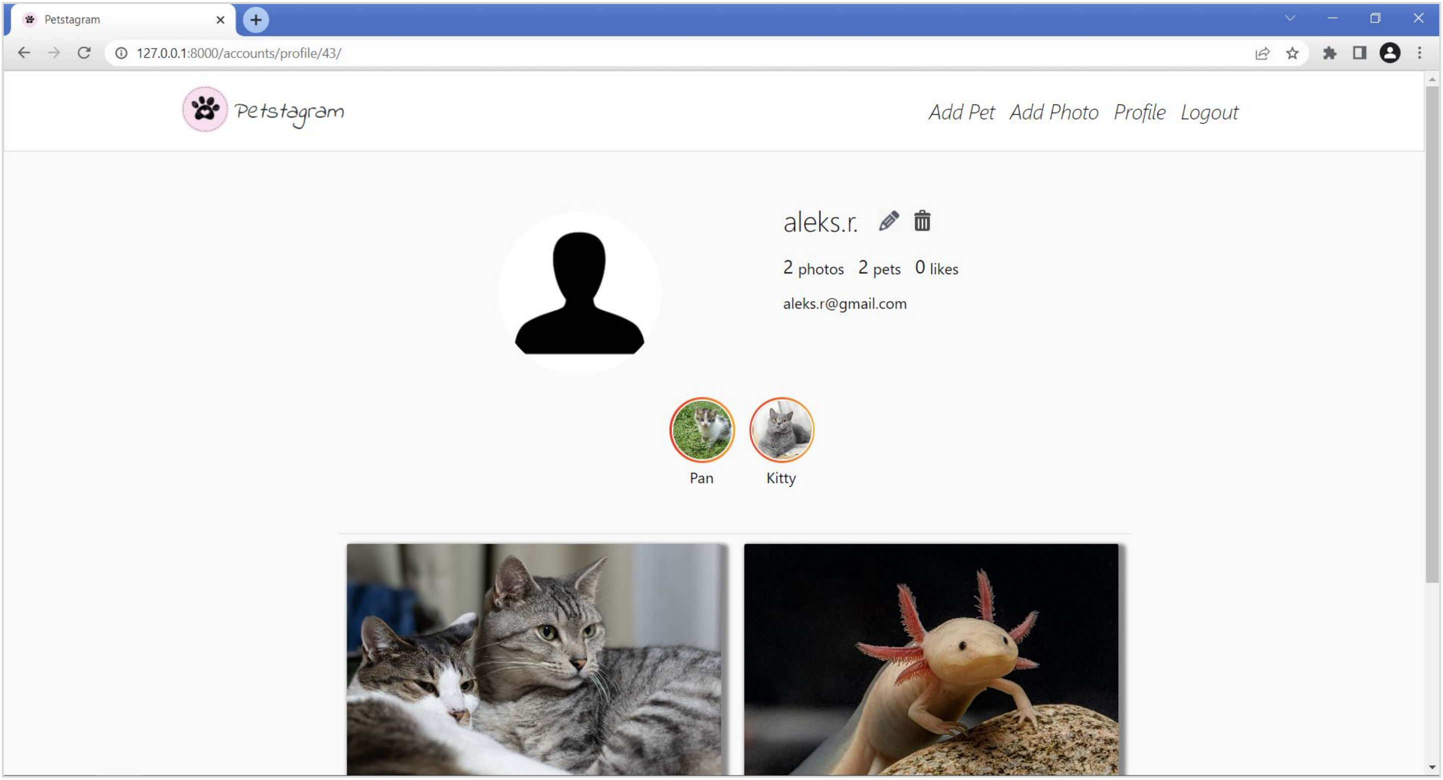
aleks.r. - What a nice Axolotl!!!

Sept. 21, 2022, 11:29 a.m.

Add comment...

© 2022 SOFTUNI WORKSHOP FOR PYTHON WEB MODULE

When there are added photos the **user profile changes** as shown below. Each user can **click on each photo** which leads to the **photo details page**:

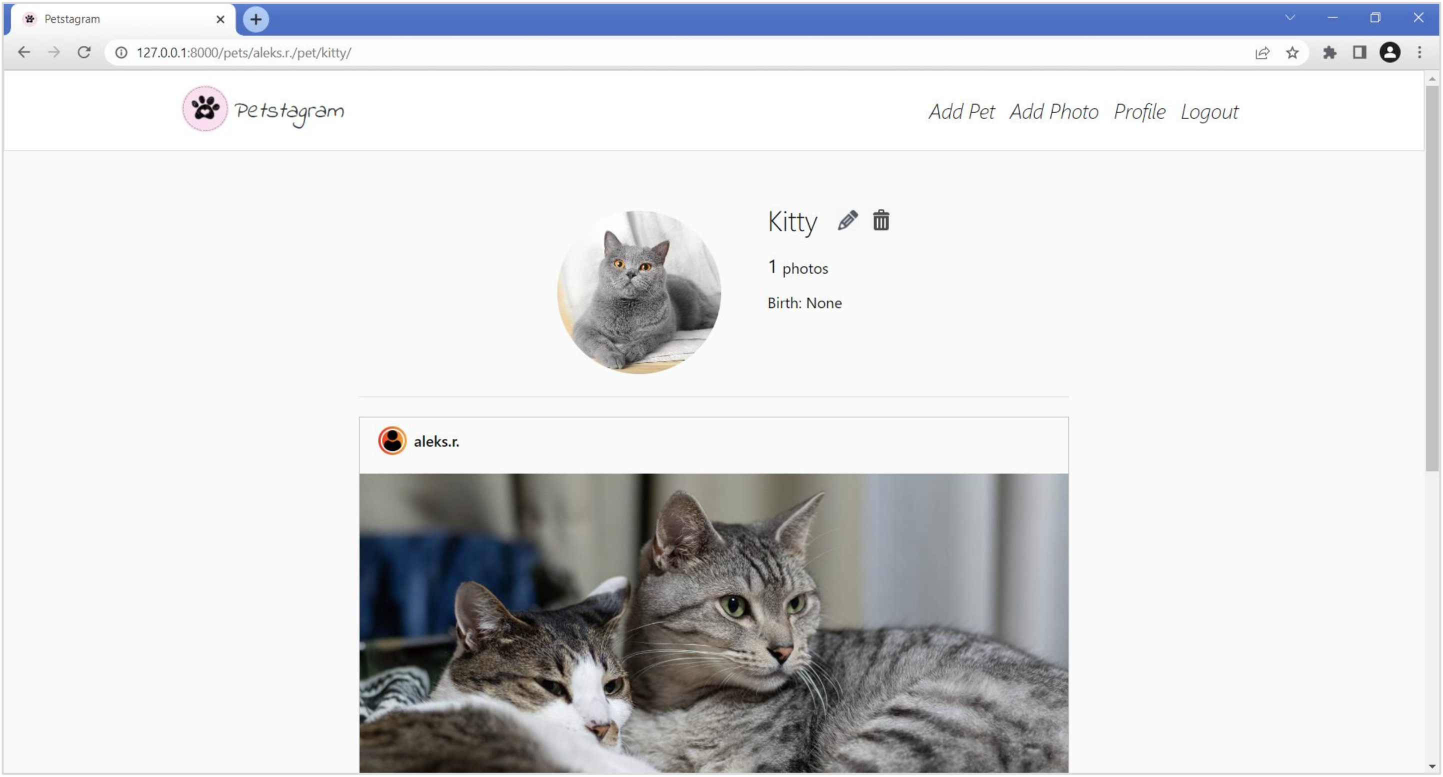


aleks.r.

2 photos 2 pets 0 likes

aleks.r@gmail.com

The pet details page changes as shown below:



Kitty

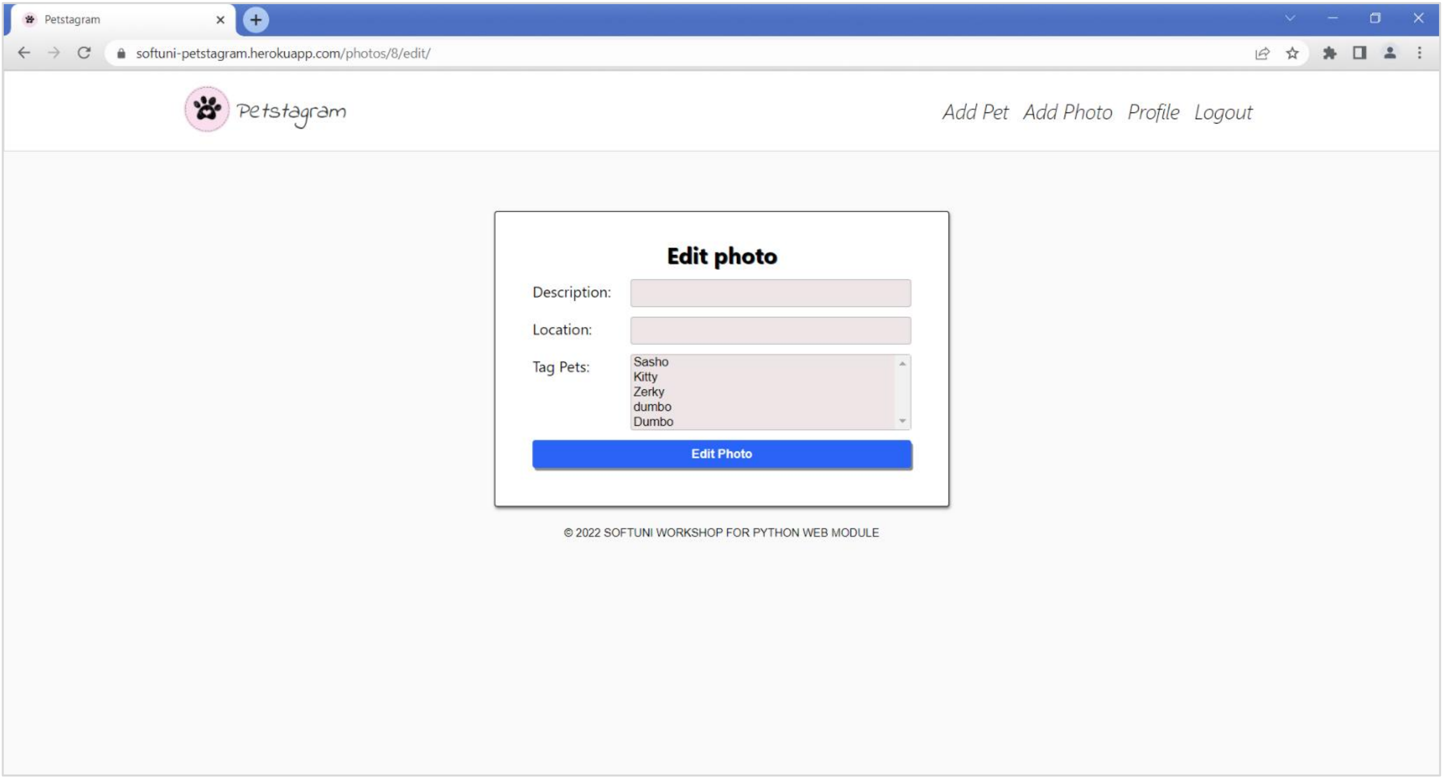
1 photos

Birth: None

aleks.r.

Photo Edit Page

The visible fields when editing a photo are "**Description**", "**Location**" and "**Tag Pets**". The photo is **NOT editable**! When you click on the "**Edit**" button, the **photo info is updated**, and the user is redirected to the **photo details** page:



Edit photo

Description:

Location:

Tag Pets: Sasho
Kitty
Zerky
dumbo
Dumbo

© 2022 SOFTUNI WORKSHOP FOR PYTHON WEB MODULE