# Workshop: Petstagram

## Part III – The Pet Forms and Comments

### Creating the Comment model

In the **"common"** app create a **model** called **Comment** with the following fields:

* **pet** – foreign key
* **comment** – text field

### Creating forms

In the **"pets" app** create a file called **forms.py** and create a **PetCreateForm**:

**class** CreatePetForm(forms.ModelForm):  
 type = forms.ChoiceField(choices=[(**"dog"**, **"dog"**), (**"cat"**, **"cat"**), (**"parrot"**, **"parrot"**)], required=**True**, widget=forms.Select(  
 attrs={  
 **'class'**: **'form-control'** },  
   
 ))  
 name = forms.CharField(required=**True**, widget=forms.TextInput(  
 attrs={  
 **'class'**: **'form-control'** }  
 ))  
 age = forms.IntegerField(required=**True**, widget=forms.TextInput(  
 attrs={  
 **'class'**: **'form-control'**,  
 **'type'**: **'number'** }  
 ))  
 image\_url = forms.URLField(required=**True**, widget=forms.TextInput(  
 attrs={  
 **'class'**: **'form-control'** }  
 ))  
 description = forms.CharField(required=**True**, widget=forms.Textarea(attrs={  
 **'class'**: **'form-control rounded-2'** }))  
  
 **class** Meta:  
 model = Pet

fields = (**'type'**, **'name'**, **'age'**, **'description'**, **'image\_url'**)

and in the **"common"** **app** create a file called **forms.py** with the **CommentForm**:

**class** CommentForm(forms.Form):  
 comment = forms.CharField(required=**True**, widget=forms.Textarea(attrs={  
 **'class'**: **'form-control rounded-2'** }))  
  
 **class** Meta:  
 model = Comment  
 fields = (**'comment'**)

### New URL's

**Create** the following new **urls** and implement the needed **views** in the **'pets'** app:

* **GET** **'localhost:8000/pets/create'** – open a **form** for pet creation
* **POST** **'localhost:8000/pets/create'** – validate the form, **create** the new pet and **redirect** the user to the **'all pets'** page
* **GET** **'localhost:8000/pets/edit/<int:pk>'** – open an **edit form** for a pet
* **POST** **'localhost:8000/pets/edit/<int:pk>'** – validate the form, **edit the pet** info and **redirect** the user to the **'pet detail'** view of the current pet
* **GET** **'localhost:8000/pets/delete/<int:pk>'** – render the given **'pet-delete.html'**
* **POST** **'localhost:8000/pets/delete/<int:pk>'** – **delete** the pet and **redirect** the user to the **'all pets'** page
* **POST** **'localhost:8000/pets/detail/<int:pk>'** – get the information of the **comment form**, create **new comment** and **redirect** the user to the **details** page again

### Templates

For the new views you are **provided** with the needed **'html'** files. Beware that the **logic** in them is **NOT** implemented. You need to implement it on your own.

### Additions to old templates

**pet-detail.html**

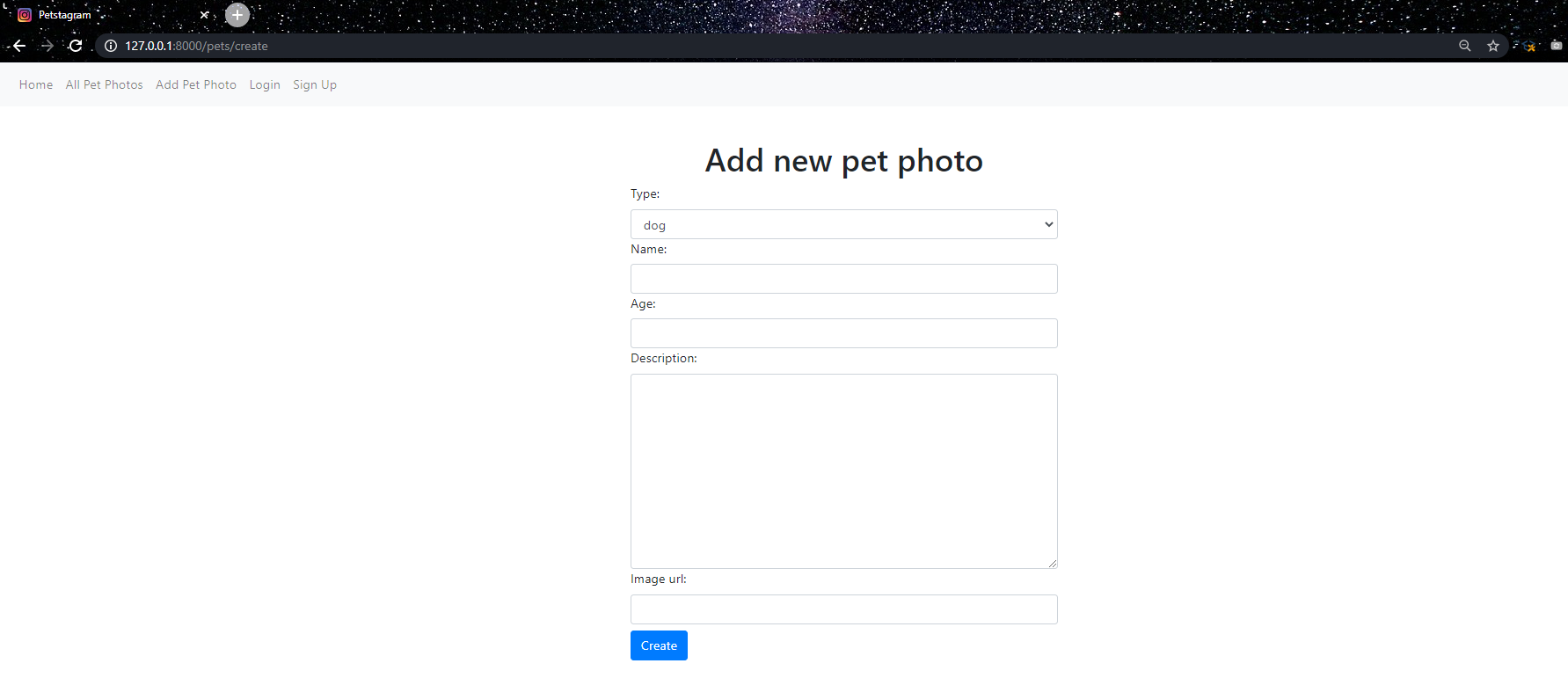
Adding the **'Edit'** and **'Delete'** buttons

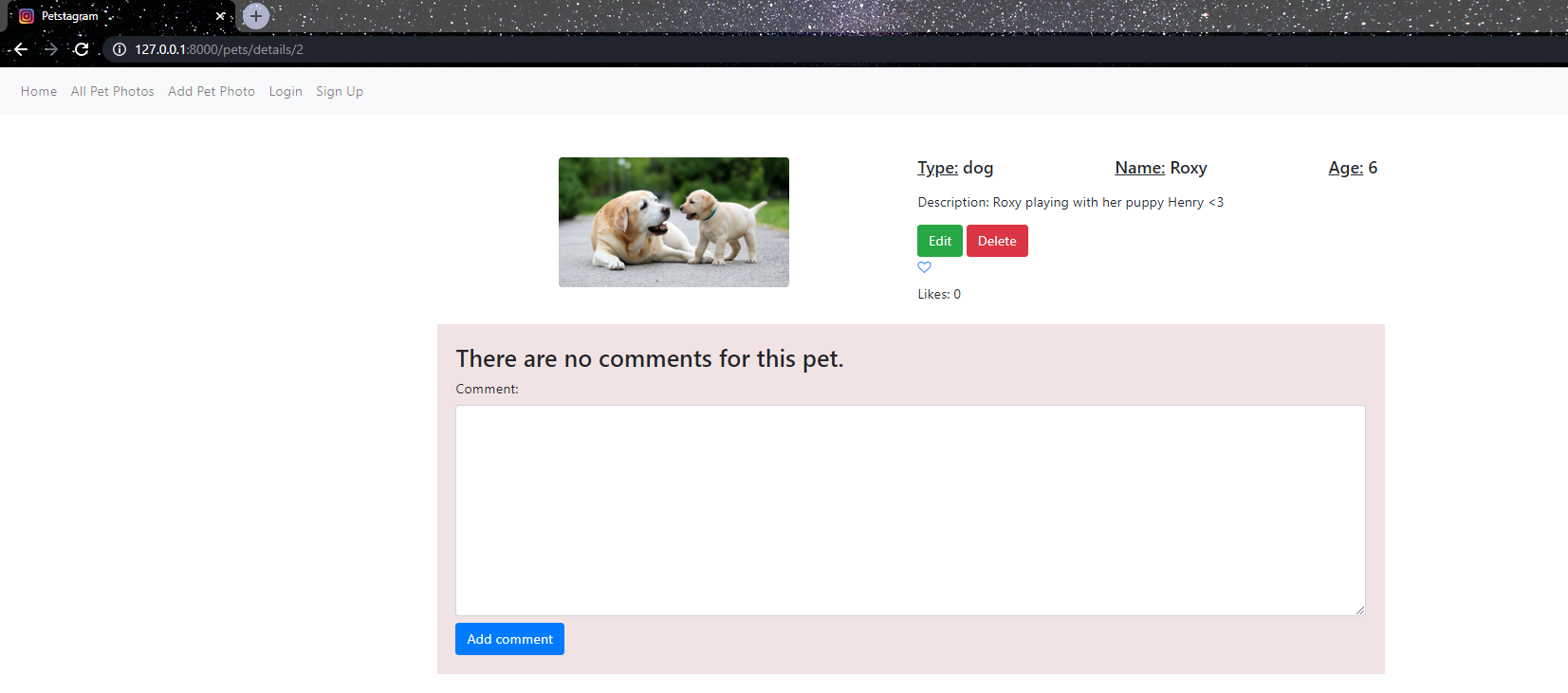
<**a href="url to the edit view" class="btn btn-success"**>Edit</**a**>  
<**a href="url to the delete view" class="btn btn-danger"**>Delete</**a**>

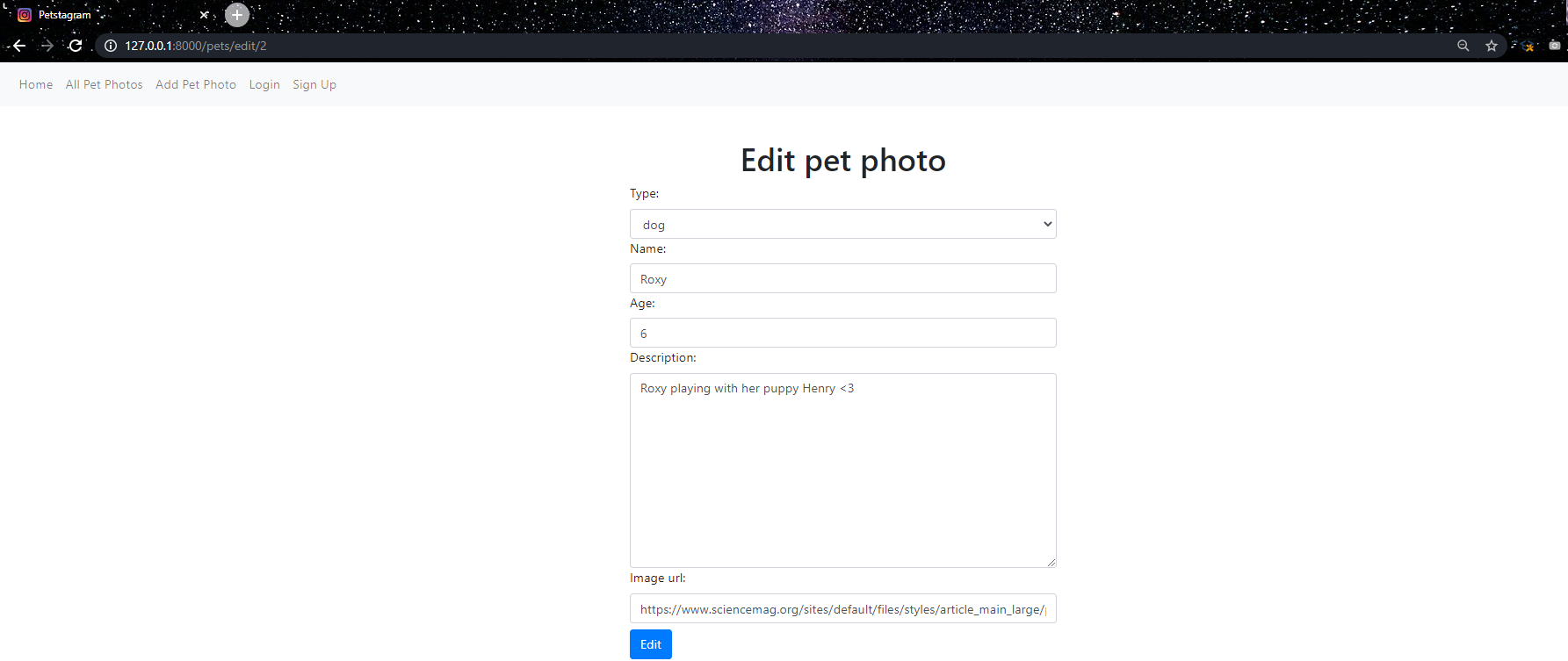
Adding the **comment** div

<**div class="d-block" style="background-color**: **#f1e3e3**; **padding**:2%**"**>  
 if comments  
 for each comment  
 <**div class="box"**>  
 <**h5**><**em**>comment\_content</**em**></**h5**>  
 <**hr**>  
 </**div**>  
 else  
 <**div class="box"**>  
 <**h3**>There are no comments for this pet. </**h3**>  
 </**div**>  
 endif  
 <**div class="box"**>  
 <**form method="post"**>  
 comment\_form  
 <**button class="btn btn-primary mt-2" type="submit"**>Add comment</**button**>  
 </**form**>  
 </**div**>  
</**div**>

### Results

**localhost:8000/pets/create****localhost:8000/pets/details/<int:pk>**

**localhost:8000/pets/edit/<int:pk>**

 **localhost:8000/pets/delete/<int:pk>**

