

Workshop: Petstagram

1. 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

The screenshot shows a web browser window with the URL `localhost:8000/pets/create`. The page has a navigation bar with links: Home, All Pet Photos, Add Pet Photo, Login, and Sign Up. The main content area is titled "Add new pet photo" and contains a form with the following fields:

- Type: A dropdown menu with "dog" selected.
- Name: A text input field.
- Age: A text input field.
- Description: A large text area.
- Image url: A text input field.
- A blue "Create" button at the bottom.

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

The screenshot shows a web browser window with the URL `localhost:8000/pets/details/2`. The page displays the details for a pet named Roxy, a 6-year-old dog. The details include:

- A photo of Roxy and her puppy Henry.
- Metadata: Type: dog, Name: Roxy, Age: 6.
- Description: Roxy playing with her puppy Henry <3.
- Buttons: Edit (green), Delete (red), and a heart icon for likes.
- Likes: 0.
- A comment section with the text "There are no comments for this pet." and a text area for adding a comment, with an "Add comment" button.

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

The screenshot shows a web browser window with the URL `localhost:8000/pets/edit/2`. The page is titled "Edit pet photo" and contains a form with the following fields:

- Type: A dropdown menu with "dog" selected.
- Name: A text input field with "Roxy" entered.
- Age: A text input field with "6" entered.
- Description: A large text area with "Roxy playing with her puppy Henry <3" entered.
- Image url: A text input field with "https://www.sciencemag.org/sites/default/files/styles/article_main_large/" entered.
- A blue "Edit" button at the bottom.

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

The screenshot shows a web browser window with the URL `localhost:8000/pets/delete/2`. The page displays a confirmation dialog with the text "Are you sure you want to delete Roxy?". There are two buttons: "Confirm" (red) and "No, go back" (blue).