Assignment: Cat Database CRUD

In this assignment you will build a web based application to track data about cats (i.e. Tabby, Persian, Maine Coon, Siamese, Manx, etc.). We will only allow logged in users to track cats.

Resources

There are several resources you might find useful:

- Recorded lectures, sample code and chapters from www.dj4e.com

- The sample CRUD code that we covered in class and used in previous assignments.

  https://github.com/csev/dj4e-samples/tree/master/autos

Sample Implementation

You can experiment with a reference implementation at:

https://crud.dj4e.com/cats

The login information is as follows:

```
Account: dj4e-crud
Password: dj4e_nn_!
```

The 'nn' is a 2-digit number that by now, you should be able to easily guess.

Specifications / Tasks

Here are some general specifications for this assignment:

- Use the Django-provided features for login and log out just as in the provided sample code.
- The auto-grader-required **meta** tag must be in the head area for all of the pages for this assignment.
- This can be added as a new application to your **mysite** project. You do not have to remove existing applications, simply add a new **cats** application. Activate any virtual environment you need (if any) and go into your `django_projects` folder and start a new application in your `mysite` project (this project already should have 'hello' and 'autos' applications from previous assignments):

```
workon django4                  # or django3 needed
cd ~/django_projects/mysite
python manage.py startapp cats
```

- Edit the **django_projects/mysite/mysite/settings.py** to update the INSTALLED_APPS:

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'home.apps.HomeConfig',
    'autos.apps.AutosConfig',
    'cats.apps.CatsConfig',    <---- Add this
]
```

- Edit the **cats/models.py** file to add Cat and Breed models as shown below with a foreign key from Cat to Breed.

```
from django.db import models
from django.core.validators import MinLengthValidator

class Breed(models.Model):
    name = models.CharField(
            max_length=200,
            validators=[MinLengthValidator(2, "Breed must be greater than 1 character")]
    )

    def __str__(self):
        return self.name

class Cat(models.Model):
    nickname = models.CharField(
            max_length=200,
            validators=[MinLengthValidator(2, "Nickname must be greater than 1 character")]
    )
    weight = models.PositiveIntegerField()
    foods = models.CharField(max_length=300)
    breed = models.ForeignKey('Breed', on_delete=models.CASCADE, null=False)

    def __str__(self):
        return self.nickname
```

- Run the commands to perform the migrations.
- Add a link to **django_projects/mysite/home/templates/home/main.html** that has the text for the top-level page.

```
    <ul>
    <li><a href="/autos">Autos CRUD</a>
    <li><a href="/cats">Cats CRUD</a>
    <ul>
```

- You should add a route to your **django_projects/mysite/mysite/urls.py** as follows:

```
urlpatterns = [
        path('', include('home.urls')),
        path('admin/', admin.site.urls),
        path('accounts/', include('django.contrib.auth.urls')),
        path('autos/', include('autos.urls')),
        path('cats/', include('cats.urls')),
    ]
```

- Create the **cats/urls.py** file to add routes for the list, edit, and delete pages for both cats and breeds. You do not need to change the **main** or **lookup** urls in **cats/urls.py** -

  You should change the 'name=' values and class name on the paths from the sample application so you don't conflict with the 'autos' application:

```
    urlpatterns = [
        path('', views.CatList.as_view(), name='all'),
        path('main/create/', views.CatCreate.as_view(), name='cat_create'),
        path('main/<int:pk>/update/', views.CatUpdate.as_view(), name='cat_update'),
        ...
    ]
```

- Edit the **cats/views.py** file to add/edit views for the list, edit, and delete pages for both cats and breeds.
- Add the appropriate templates to **cats/templates** following the naming conventions for the templates.
- If you have not already done so, create the necessary templates in **home/templates/registration** to support the login / log out views.
- Edit **cats/admin.py** to add the Cat and Breed models to the Django administration interface.
- If you have not already done so, create a superuser so you can test the admin interface and log in to the application.

Hand-Testing Your Application

While it might be tempting to edit all your code following the above instructions and immediately send it to the autograder, you usually can save a lot of time by running a quick hand-check of your application. Errors are much easier to see in a browser than in the autograder. Do all these steps:

- Log in to your application
- Go to the main page to list all the cats
- Add a new breed
- View all the breeds
- Update a breed
- Add a new cat
- View all the cats
- Update a cat

- Delete a cat
- View all the breeds
- Delete a breed

You should be able to do this hand test in about a minute once you do it a few times.

Once you are confident that all the views are working without error, then you can send it to the autograder.

Using the Autograder

This assignment will be automatically graded. You will have unlimited attempts in the autograder until the deadline for submission. Your web server will need an Internet-accessible URL so you can submit it for autograding. You can do this either using PythonAnywhere.

Please see the process for handing in the assignment at the end of this document.

If the autograder complains about a missing "dj4e" meta tag, add or edit it in your **home/templates/base_bootstrap.html** file:

```
<meta name="dj4e" content="--provided-by-autograder--">
```

If the autograder does not find the tag, it will run all the tests but will not treat the grade as official.

If the autograder complains about a missing "dj4e-code" meta tag, add or edit it in your **home/templates/base_bootstrap.html** file:

```
<meta name="dj4e-code" content="42169769777706">
```

If you are adding an application to an existing Django project that you have already run through the autograder, you probably already have good value for both meta tags in your template file.

What To Hand In

This assignment will be autograded by a link that you will be provided with in the LMS system. When you launch the autograder, it will prompt for a web-accessible URL where it can access your web application.

Provided by: www.dj4e.com