pep8ci.herokuapp.com



CI Python Linter

Project urls.py

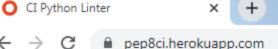
```
"""little_learner_lab_logs URL Configuration
3 The `urlpatterns` list routes URLs to views. For more information please see:
       https://docs.djangoproject.com/en/3.2/topics/http/urls/
   Examples:
   Function views
       1. Add an import: from my_app import views
       Add a URL to urlpatterns: path('', views.home, name='home')
   Class-based views
       1. Add an import: from other_app.views import Home
       Add a URL to urlpatterns: path('', Home.as_view(), name='home')
   Including another URLconf
       1. Import the include() function: from django.urls import include, path
       Add a URL to urlpatterns: path('blog/', include('blog.urls'))
   ....
   from django.contrib import admin
   from django.urls import path, include
   urlpatterns = [
       path('admin/', admin.site.urls),
       path('summernote/', include('django_summernote.urls')),
       path('', include('logs.urls'), name='logs'),
       path('profiles/', include('profiles.urls'), name='profiles'),
       path('accounts/', include('allauth.urls')),
   handler404 = 'little learner lab logs.views.handler404'
```

Setting





Results



CI Python Linter

Profiles models.py

Setting

Results

```
''' Profile Model Imports '''
 from django.db import models
from django.contrib.auth.models import User
from django.db.models.signals import post_save
 from django.dispatch import receiver
 from django.core.exceptions import ObjectDoesNotExist
 from cloudinary.models import CloudinaryField
 # Profile Model
 class Profile(models.Model):
      Profile model
      Extends user model
      user = models.OneToOneField(
          User,
          on_delete=models.CASCADE,
          blank=True,
          null=True)
      profile_image = CloudinaryField(
          'image',
          default='../default_profile_afxozd',
          blank=True
      brief_bio = models.TextField()
      def __str__(self):
          To return the user's username object as a string
          return f'Profile for {self.user.username}'
 @receiver(post_save, sender=User)
```

CI Python Linter

Profiles forms.py

Setting

Results

```
from django import forms
    from django.forms import ModelForm, FileInput
    from django.contrib.auth.models import User
4 from .models import Profile
   from crispy_forms.helper import FormHelper
from crispy_forms.layout import Layout, Div, Field, Button, Submit, HTML
    from crispy_forms.bootstrap import FormActions
10 -
    class ProfileForm(ModelForm):
        Create form class based on Profile model
        Reference for the code in the link below:
        https://github.com/Code-Institute-Submissions/andy-guttridge-song-mates
        def __init__(self, *args, **kwargs):
             Init form using crispy forms FormHelper for form layout
            super(ProfileForm, self).__init__(*args, **kwargs)
             self.helper = FormHelper(self)
             self.helper.form_tag = False
             self.helper.form_class = 'form-horizontal'
             self.helper.field_class = 'col-md-12'
             self.helper.layout = Layout(
                 HTML('<div class = "text-start">'),
                Field('brief_bio', css_class='bkgnd-color-1 txt-color-2'),
                 Field('profile_image', css_class='bkgnd-color-1 txt-color-2'),
                 # Approach to using a HTML helper class to display an image from
                 # the data base in the form from
                 # https://stackoverflow.com/questions/21076248/imagefield-preview-in-crispy-forms-layer
                 Div(
                     Div('Username: '),
                     Div('Brief-Bio'),
                 ),
```

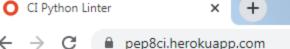
CI Python Linter

Profiles views.py

Setting

Results

```
from django.shortcuts import render, redirect, HttpResponseRedirect, get_object_or_404 # NOQA
   from django.urls import reverse lazy
   from django.views import View
4 from django.utils.decorators import method decorator
5 from django.template import RequestContext
6 from django.contrib import messages
   from django.contrib.auth.decorators import login_required
   from .forms import ProfileForm
   from .models import Profile
    class ProfileAccount(View):
        User profile and account view.
        Redirects to login page if user not authenticated.
16 -
        Credits for the code:
        https://github.com/Code-Institute-Submissions/andy-guttridge-song-mates
        @method_decorator(login_required)
        def get(self, request, *args, **kwargs):
20 -
            if not Profile.objects.filter(user=request.user).exists():
                profile = Profile(user=request.user)
                profile.save()
            # Retrieve profile from database, create form from it and
            profile_queryset = Profile.objects.filter(user=request.user)
            get object or 404(profile queryset)
            profile = profile_queryset.first()
            profile_form = ProfileForm(instance=profile)
            return render(
                request,
                'edit_profile.html',
                     'profile': profile,
                    'form': profile_form
```





CI Python Linter

Profiles urls.py

```
from . import views
from django.urls import path
urlpatterns = [
    path('edit-profile/', views.ProfileAccount.as_view(), name='edit_profile'),
    path('user-delete/', views.UserDelete.as_view(), name='user_delete'),
    path('update-profile/', views.UpdateProfile.as_view(),
         name='update profile'),
```

Setting





Results

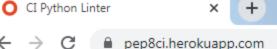
CI Python Linter

Logs models.py

Setting

Results

```
from django.db import models
from django.contrib.auth.models import User
from cloudinary.models import CloudinaryField
from django.urls import reverse
from django.template.defaultfilters import slugify
STATUS = ((0, "Draft"), (1, "Published"))
class Category(models.Model):
    Category Model linked to each post
    category_title = models.CharField(max_length=80, blank=False)
    slug = models.SlugField(max_length=200, unique=True)
    class Meta:
        verbose_name = "category"
        verbose_name_plural = "categories"
    def __str__(self):
        return self.category_title
class Post(models.Model):
    Model for lab log posts
    title = models.CharField(max_length=200, unique=True)
    author = models.ForeignKey(
        User,
        on delete=models.CASCADE,
        related_name="log_posts",
        blank=True,
```



CI Python Linter

Logs forms.py

Setting

Results

```
from django import forms
from django_summernote.widgets import SummernoteWidget
from .models import Post, Comment
class UploadFileForm(forms.Form):
    To upload images
    title = forms.CharField(max_length=100)
    file = forms.FileField()
    class Meta:
        model = Post
       fields = ['image']
class PostForm(forms.ModelForm):
    Django form to add lab log posts
    class Meta:
        # Get post model and choose which fields to display
        model = Post
        fields = ('title', 'author', 'description', 'items_required', 'steps_to_perform', 'category'
            , 'image',) # noga: E50
        widgets = {
            'title': forms.TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Title'}),
            'author': forms.TextInput(attrs={
                'class': 'form-control',
                'value': '',
                'id': 'author_field',
```



CI Python Linter

Logs urls.py

```
from . import views
from django.urls import path
from .views import PostDetail, AllPosts, PostLike, SharedPostsByUsers, DeletePost # noqa: E501

urlpatterns = [
    path('', views.PostList.as_view(), name='home'),
    path('search/', views.PostSearch, name='search'),
    path('lab_logs/', AllPosts.as_view(), name='lab-logs'),
    path('my_page/', views.SharedPostsByUsers.as_view(), name='my-page'),
    path('log_details/<slug:slug>', PostDetail.as_view(), name='log-details'),
    path('like/<slug:slug>', views.PostLike.as_view(), name='post-like'),
    path('add_logs/', views.CreatePost, name='add-logs'),
    path('edit/<slug:slug>/', views.EditPost, name='edit-logs'),
    path('delete/<slug:slug>/', DeletePost.as_view(), name='delete-logs'),
    ]

17
```

Setting





Results



class PostDetail(View):

WE

34 -

CI Python Linter

from django.shortcuts import render, get_object_or_404, reverse, redirect from django.contrib.auth.decorators import login_required from django.views import generic, View from django.http import HttpResponseRedirect from django.contrib import messages 6 from django.utils.text import slugify from django.contrib.auth.mixins import LoginRequiredMixin from django.db.models import Q from django.views.generic import ListView from .models import Post from .forms import PostForm, CommentForm, UploadFileForm, EditForm class PostList(generic.ListView): ''' Class to show list of posts''' model = Post queryset = Post.objects.filter(status=1).order_by('-created_on') template name = 'index.html' paginate by = 3class AllPosts(generic.ListView): to get all the lab log posts, and display 12 posts per page model = Post queryset = Post.objects.filter(status=1).order_by('created_on') # noqa: E501 template name = 'lab logs.html' paginate_by = 12

''' Class to show selected post in detail view '''
def get(self, request, slug, *args, **kwargs):

queryset = Post.objects.filter(status=1)
nost = get object or 404(queryset, slug=slug)

Logs views.py



Setting

