from django.db import models

from django.shortcuts import render, get\_object\_or\_404

from django.http import HttpResponseRedirect

from django.views.generic import ListView, DetailView, CreateView, UpdateView, DeleteView

from django.contrib.auth.mixins import UserPassesTestMixin, LoginRequiredMixin

from .models import Question, Comment

from .forms import CommentForm

from django.urls import reverse, reverse\_lazy

# Create your views here.

def home(request):

return render(request, 'home.html')

def about(request):

return render(request, 'about.html')

def like\_view(request, pk):

post = get\_object\_or\_404(Question, id=request.POST.get('question\_id'))

liked = False

if post.likes.filter(id=request.user.id).exists():

post.likes.remove(request.user)

liked = False

else:

post.likes.add(request.user)

liked = True

return HttpResponseRedirect(reverse('qesbase:question-detail', args=[str(pk)]))

class QuestionListView(ListView):

model = Question

context\_object\_name= 'questions'

ordering= ['-date\_created']

def get\_context\_data(self, \*\*kwargs):

context = super().get\_context\_data(\*\*kwargs)

search\_input = self.request.GET.get('search-area') or ""

if search\_input:

context['questions'] = context['questions'].filter(title\_\_icontains = search\_input)

context['search\_input'] = search\_input

return context

class QuestionDetailView(DetailView):

model = Question

def get\_context\_data(self, \*args, \*\*kwargs):

context = super(QuestionDetailView, self).get\_context\_data()

something = get\_object\_or\_404(Question, id=self.kwargs['pk'])

total\_likes = something.total\_likes()

liked = False

if something.likes.filter(id=self.request.user.id).exists():

liked = True

context['total\_likes'] = total\_likes

context['liked'] = liked

return context

class QuestionCreateView(LoginRequiredMixin, CreateView):

model = Question

fields=['title','content']

context\_object\_name='question'

def form\_valid(self, form):

form.instance.user = self.request.user

return super().form\_valid(form)

class QuestionUpdateView(UserPassesTestMixin, LoginRequiredMixin, UpdateView):

model = Question

fields=['title','content']

def form\_valid(self, form):

form.instance.user = self.request.user

return super().form\_valid(form)

def test\_func(self):

question= self.get\_object()

if self.request.user == question.user:

return True

return False

class QuestionDeleteView(UserPassesTestMixin, LoginRequiredMixin, DeleteView):

model = Question

context\_object\_name = 'question'

success\_url ="/"

def test\_func(self):

question= self.get\_object()

if self.request.user == question.user:

return True

return False

class CommentDetailView(CreateView):

model = Comment

form\_class = CommentForm

template\_name = 'qesbase/question-detail.html'

def form\_valid(self, form):

form.instance.question\_id = self.kwargs['pk']

return super().form\_valid(form)

success\_url = reverse\_lazy('qesbase:question-detail')

class AddCommentView(CreateView):

model = Comment

form\_class = CommentForm

template\_name = 'qesbase/question-answer.html'

def form\_valid(self, form):

form.instance.question\_id = self.kwargs['pk']

return super().form\_valid(form)

success\_url = reverse\_lazy('qesbase:question-lists')