ABP - EJERCICIO INDIVIDUAL 7.7

Guido Saavedra

REPO GITHUB: https://github.com/BOKG/MOD7IND

En documento Word comenta, de forma detallada, de qué forma participan las siguientes aplicaciones
preinstaladas en tu proyecto.
- django.contrib.admin
- django.contrib.auth
- django.contrib.sessions

Las aplicaciones preinstaladas en Django, como django.contrib.admin, django.contrib.auth y django.contrib.sessions, desempeñan roles clave en un proyecto Django. Aquí te explico brevemente la función de cada una de ellas:

django.contrib.admin: Esta aplicación proporciona una interfaz de administración basada en la web que permite gestionar y administrar los modelos de la base de datos de manera sencilla. Con esta aplicación, puedes crear, leer, actualizar y eliminar registros de modelos sin tener que escribir código adicional. El panel de administración de Django es altamente personalizable y se puede ajustar para adaptarse a las necesidades específicas de tu proyecto.

django.contrib.auth: Esta aplicación es responsable de la autenticación y autorización de usuarios en tu aplicación web. Proporciona modelos y vistas para gestionar usuarios, grupos y permisos. Además, incluye características como el inicio de sesión, cierre de sesión, registro de usuarios, cambio de contraseñas y restablecimiento de contraseñas olvidadas. También ofrece decoradores y funciones útiles para proteger vistas y limitar el acceso a usuarios autenticados.

django.contrib.sessions: Esta aplicación se encarga de gestionar las sesiones de los usuarios en tu aplicación. Permite almacenar y recuperar datos de sesión, lo que es útil para mantener el estado de la sesión del usuario a medida que navega por tu sitio web. Django utiliza una clave de sesión única almacenada en una cookie del navegador del usuario para identificar y asociar los datos de sesión. Esto permite recordar la información del usuario, como las preferencias o el carrito de compras, entre diferentes solicitudes HTTP.

Aplicacion de dichos modulos en mi aplicacion.

django.contrib.admin:

```python
from django.contrib import admin
from .models import Task, Label

# Register your models here.
admin.site.register(Task)
admin.site.register(Label)
```

django.contrib.auth:

```python
from django.shortcuts import get_object_or_404, redirect
from django.views.generic import ListView
from django.contrib.auth.mixins import LoginRequiredMixin
from django.views.generic.edit import CreateView, UpdateView
from django.urls import reverse_lazy
from django.contrib.auth.decorators import login_required
```

```python
class CreateTask(LoginRequiredMixin, CreateView):

    model = Task
    form_class = TaskForm
    template_name = 'tareas/crear_tarea.html'
    success_url = reverse_lazy('home')

    def form_valid(self, form):
        form.instance.user = self.request.user
        return super().form_valid(form)


class DetailsTask(LoginRequiredMixin, UpdateView):
    model = Task
    template_name = 'tareas/tarea_detalles.html'
    context_object_name = 'tarea'
    fields = ['status']

    def get_queryset(self):
        queryset = super().get_queryset()
        return queryset.filter(id=self.kwargs['pk'])

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['tarea'] = self.get_queryset().first()
        return context

    def post(self, request, *args, **kwargs):
        tarea = self.get_queryset().first()
        tarea.status = request.POST.get('status')
        tarea.status = 'C'
        tarea.save()
        return redirect('home')

class EditTask(LoginRequiredMixin, UpdateView):
    model = Task
    form_class = TaskForm
    template_name = 'tareas/editar_tarea.html'
    success_url = reverse_lazy('home')

    def get_queryset(self):
        queryset = super().get_queryset()
        return queryset.filter(id=self.kwargs['pk'])
```

django.contrib.sessions:
Actualmente este modulo no lo tengo aplicado