PROYECTO FINAL PYTHON

Para crear una aplicación de reserva de alojamiento en Python, puedes seguir estos pasos generales:

1. Crear una clase de alojamiento que tenga atributos como nombre, dirección, número de habitaciones, etc.
2. Crear una función de reserva que permita al usuario seleccionar una fecha y reservar una habitación en el alojamiento. Esta función debería actualizar el estado de la habitación (disponible o no) y almacenar la información de la reserva.
3. Crear una vista de calendario que muestre los días disponibles en verde y los días reservados en rojo. Esta vista debería mostrar la información de la reserva almacenada en el paso 2.
4. Crear una interfaz de usuario para permitir al usuario interactuar con la aplicación, como seleccionar un alojamiento, ver los detalles del alojamiento y reservar una habitación.

Puedes usar un framework de Python para el desarrollo de aplicaciones web como Flask o Django para crear la interfaz de usuario y manejar las solicitudes HTTP. También puedes usar una librería de calendario como calendar o icalendar para crear la vista de calendario.

detallemos paso por paso, comencemos por el paso 2, creemos la funcion de reserva

Para crear la función de reserva, puedes seguir estos pasos específicos:

1. Crear una función llamada "reservar" que tome como parámetros el objeto de alojamiento y la fecha de reserva.
2. En la función, verificar si hay habitaciones disponibles en el alojamiento en la fecha especificada. Si no hay habitaciones disponibles, devolver un mensaje de error.
3. Si hay habitaciones disponibles, actualizar el estado de la habitación a "no disponible" y almacenar la información de la reserva (fecha, nombre del huésped, etc.) en una lista o diccionario.
4. Devolver un mensaje de confirmación de la reserva.

Aqui te dejo un ejemplo de una posible implementación de la función de reserva en Python:

Copy codeclass Alojamiento:

def \_\_init\_\_(self, nombre, direccion, num\_habitaciones):

self.nombre = nombre

self.direccion = direccion

self.num\_habitaciones = num\_habitaciones

self.habitaciones\_disponibles = num\_habitaciones

self.reservas = []

def reservar(self, fecha, huesped):

if self.habitaciones\_disponibles == 0:

return "Lo siento, no hay habitaciones disponibles para esa fecha."

self.habitaciones\_disponibles -= 1

reserva = {"fecha": fecha, "huesped": huesped}

self.reservas.append(reserva)

return "Reserva realizada con éxito para el día {}.".format(fecha)

En este ejemplo, la clase Alojamiento tiene atributos como nombre, dirección y número de habitaciones. Al momento de crear un objeto de esta clase, se inicializan estos atributos y se crea una lista vacía de reservas. El método reservar verifica si hay habitaciones disponibles en la fecha especificada y, si es así, actualiza el estado de la habitación y agrega la reserva a la lista de reservas.

en la reserva deberiamos contemplar fecha de entrada y fecha de salida ya que los alquileres suelen ser de muchos dias

Sí, tienes razón. En un sistema de reservas de alojamiento, es común que las reservas se realicen por un período de tiempo específico (por ejemplo, desde el día de llegada hasta el día de salida).

Para contemplar las fechas de entrada y salida en el sistema de reservas, puedes hacer los siguientes cambios en la función reservar:

1. Modificar la función para que tome dos parámetros adicionales: fecha\_entrada y fecha\_salida.
2. Verificar si hay habitaciones disponibles en el alojamiento para el período de tiempo especificado (desde fecha\_entrada hasta fecha\_salida).
3. Actualizar el número de habitaciones disponibles para el período de tiempo especificado.
4. Almacenar la información de la reserva en una lista o diccionario. Esta información debería incluir las fechas de entrada y salida, así como el nombre del huésped y otros detalles relevantes.

Aqui te muestro un ejemplo de cómo podría quedar la función reservar con fechas de entrada y salida:

Copy codeclass Alojamiento:

def \_\_init\_\_(self, nombre, direccion, num\_habitaciones):

self.nombre = nombre

self.direccion = direccion

self.num\_habitaciones = num\_habitaciones

self.habitaciones\_disponibles = num\_habitaciones

self.reservas = []

def reservar(self, fecha\_entrada, fecha\_salida, huesped):

dias\_reservados = (fecha\_salida - fecha\_entrada).days

if self.habitaciones\_disponibles < dias\_reservados:

return "Lo siento, no hay habitaciones disponibles para ese período."

self.habitaciones\_disponibles -= dias\_reservados

reserva = {"fecha\_entrada": fecha\_entrada, "fecha\_salida": fecha\_salida, "huesped": huesped}

self.reservas.append(reserva)

return "Reserva realizada con éxito para el período {} - {}.".format(fecha\_entrada, fecha\_salida)

En este ejemplo, la función reservar toma tres parámetros: fecha\_entrada, fecha\_salida y huesped. Se utiliza la diferencia entre fecha\_salida y fecha\_entrada para calcular los días que se van a reservar, y se actualiza el número de habitaciones dispon

django-admin startproject ProyectoSMA . desde concols creo el proyecto, el punto al final sirve para evitar crear otra carpeta

python manage.py migrate

python manage.py startapp Appsma

en settings pongo el nombre de la app

creamos modelos

makemigrations

migrate

agregamos información a los modelos

en admin agrego

from django.contrib import admin

from Appsma.models import \*

# Register your models here.

admin.site.register(Huesped)

admin.site.register(Alojamientos)

admin.site.register(Vehiculos)

creo superusuario para gestionar datos desde el admin de django