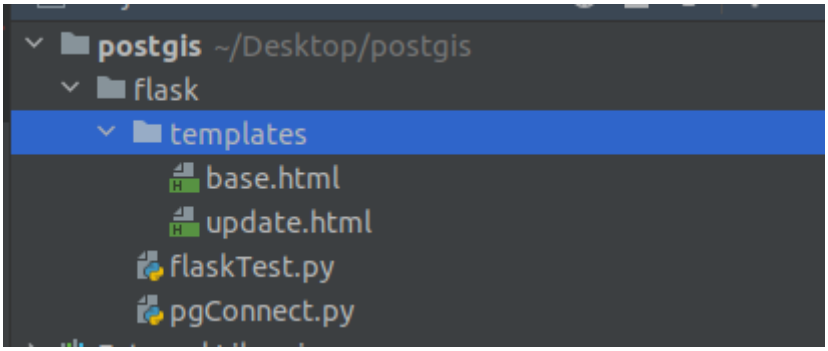
1.目录结构

主要就是网页文件需要放在template目录下（这是flask的特性）

目录结构如下图：



2.如何连接数据库?

我们定义了一个pgconnect.py的脚本文件用于连接数据库

通psycopg2这个库可以实现连接（pip install psycopg2即可下载）

主要的实现内容：

（1） 使用psycopg2.connect方法对数据库进行连接（这里需要先确保我们的数据库服务是打开并且是可连接的，apt方式安装的直接psql -u postgres即可开启服务，源码编译的使用pg_ctl start开启服务，使用psql -h 127.0.0.1 -U 用户名 -d 数据库 可以检查数据库是否可连接）

主要报错：1.用户名密码错误，改密码就行，问题不大

2.用户权限错误，修改用户连接的配置文件，我的示例如下：（为了防止出现权限不够，我直接全部信任了）

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD

# "local" is for Unix domain socket connections only
local   all             all                                     trust
# IPv4 local connections:
host    all             all             127.0.0.1/32            trust
# IPv6 local connections:
host    all             all             ::1/128                 trust
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication     all                                     trust
host    replication     all             127.0.0.1/32            trust
host    replication     all             ::1/128                 trust
host    all             all             0.0.0.0/0               trust

host    replication     replica         0.0.0.0/0               trust # 流复制用
户密码验证登录
host    all             postgres        0.0.0.0/0               trust # 拒绝超级
用户从网络登录
```

接下来没问题就是连接了

（1）确定端口无误（第二次实验修改了端口数值，如果改了这里对应修改就行）

（2）确定ip地址，在虚拟机下可以用指令查看到（这里的127.0.0.1是回环地址，我直接在虚拟机下运行的python，可以直接用这个地址，如果是window下运行连接到虚拟机服务，则需要查看ip地址）

（3）用户名以及你想操作的数据库名

(C:\Users\鹤年\AppData\Roaming\Typora\typora-user-images\image-20211117151303157.png)

```python
#更新临时禁飞区表
import psycopg2

def insertGeo(Customer_ID,Last_Name,First_Name,Address,City,State,Zip,Phone,Email_Address,Credit_Card_Details,Birth_Date,Exercise_History,Fit
    conn = psycopg2.connect(database="test",user="postgres",password="10086",host="127.0.0.1",port="5432")
    cur = conn.cursor()

    sql="insert into onlineworkout(Customer_ID,Last_Name,First_Name,Address,City,State,Zip,Phone,Email_Address,Credit_Card_Details,Birth_Date
    print(sql)
    cur.execute(sql)
    conn.commit()
    print('更新成功')
    # http://127.0.0.1:5000/api/updateNoFly/insert?start_time='2018-01-01'&end_time='2018-01-01'&reg_name='www'&reg_radius=1&valid_time='1-1'
# insertGeo('2018-01-01','2018-01-01','www',1,'1-1','1','sds2','{"type":"MultiPolygon","coordinates":[[[[114.160603606931,22.2858883847479],[
```

连接无误后，我们生成一个sql指令的字符串（如上图，名叫sql），然后用execute方法，执行的效果就相当于你在数据库下输入这个指令了

3.与前端的交互

既然我们可以通过这个脚本直接对数据库执行指令了，那么问题就是我们怎么获取到这些指令了（也就是在网页上对数据库增删改查）

我们需要的是什么？（对用户来说，我们输入的是红框里面的这些数据）（ps.因为在脚本里只有插入还没有实现建表指令，所以需要先在自己的数据库下建立这张表先）

	A	B	C	D	E	F	G	H	I	J	K	L	M	
1	Customer ID	Last Name	First Name	Address	City	State	Zip	Phone	Email Address	Credit Card Details	Birth Date	Exercise Histor	Fitness Level	Medical Hea
2	101	Dickerson	Allen	138 Woodlawn Ave	Seattle	WA	98119	(206)256-0097	AED@zoom.net	4536	1986/4/10	Frequent	Very Fit	Very Health
3	102	Faber	Dale	121 Chaucer Lane	Bronx	NY	10463	(212)549-3324	Oldie@brandywine.net	7624	1958/7/12	Moderate	Moderately Fit	Somewhat He
4	103	Hearn	Arthur	26 Julie Court	Media	PA	19063	(610)543-7611	Ahearn@aol.com	1196	1975/8/15	Frequent	Unfit	Somewhat He
5	104	Lavelle	Shirley	4001 Birch Street	Peekskill	NY	10566	(914)736-5512	Shirl121@hotmail.com	3746	1960/9/25	Frequent	Very Fit	Somewhat He
6	105	Nelson	Janice	23 Geneva Blvd	Piscataway	NJ	08854	(210)469-3541	NelsJ@hotmail.com	2984	1948/12/25	None	Unfit	Unheathry
7	106	Schwartz	Byron	2 Waverly Rd	Deep River	CT	09776	(203)536-0954	BS599@aol.com	0912	1989/2/2	Frequent	Moderately Fit	Very Health
8	107	Sunzar	Sam	103 Chadd Rd	Owings Mills	MD	21117	(301)762-1298	55SS@zoom.net	7864	1975/3/16	Moderate	Unfit	Unheathry
9	108	Turner	Cynthia	1502 Valley Stream Lane	Salt Lake Ci	UT	84109	(801)675-9812	CT@brandywine.net	3562	1992/5/6	Frequent	Very Fit	Somewhat He
10	109	Trapp	John	220 E Main Street	Scotch Plain	NJ	07076	(201)757-8876	Trapp334@hotmail.com	8475	1967/11/26	Moderate	Moderately Fit	Very Health
11	110	Wills	Billy	25 Brown Lane	Centerville	OH	45459	(513)435-9123	BillyW@aol.com	0911	1951/7/1	None	Moderately Fit	Somewhat He
12														

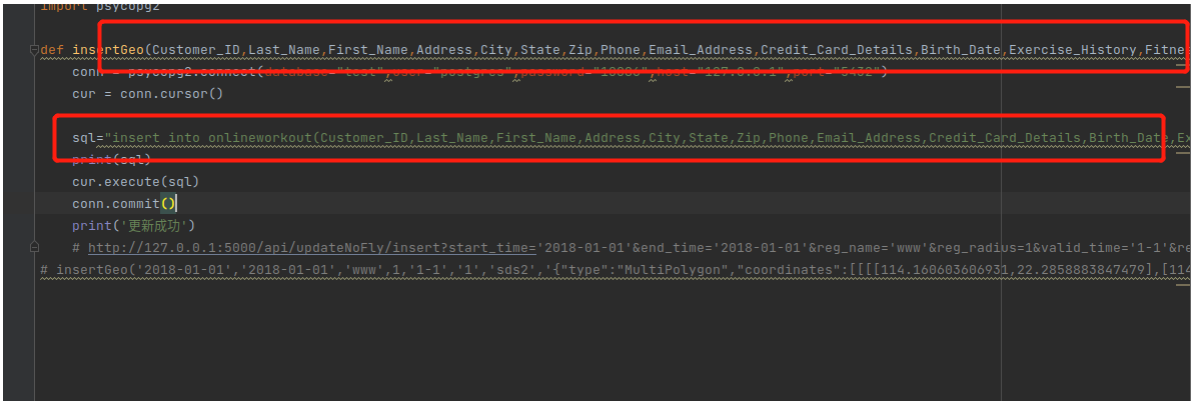
我们首先通过html的post请求获取到用户的这些输入(update.html)

```
        <tr>
            <td>Customer_ID：</td>
            <td><input type="text" placeholder="101" name="Customer_ID" value="101"/></td>
        </tr>
        <tr>
            <td>Last_Name：</td>
            <td><input type="text" placeholder="Dickerson" name="Last_Name" value="Dickerson"/></t
        </tr>
        <tr>
            <td>First_Name：</td>
            <td><input type="text" placeholder="Allen" name="First_Name" value="Allen"/></td>
        </tr>
        <tr>
            <td>Address：</td>
            <td><input type="text" placeholder="138 Woodlawn Ave" name="Address" value="138 Woodla
        </tr>
        <tr>
            <td>City：</td>
            <td><input type="text" placeholder="Seattle" name="City" value="Seattle"/></td>
        </tr>
        <tr>
            <td>State：</td>
            <td><input type="text" placeholder="WA" name="State" value="WA"/></td>
        </tr>
        <tr>
            <td>Zip：</td>
            <td><input type="text" placeholder="98119" name="Zip" value="98119"/></td>
        </tr>
        <tr>
            <td>Phone：</td>
            <td><input type="text" placeholder="(206)256-0097" name="(206)256-0097" /></td>
```

通过flask的交互功能从html里获取到这些数据（flasktest.py)

```
    return render_template('update.html')
else:
    Customer_ID = request.form.get('Customer_ID')#post请求。获取模版语言中输入框输入的值
    Last_Name = request.form.get('Last_Name')
    First_Name = request.form.get('First_Name')
    Address = request.form.get('Address')
    City = request.form.get('City')
    State = request.form.get('State')
    Zip = request.form.get('Zip')
    Phone = request.form.get('Phone')
    Email_Address = request.form.get('Email_Address')
    Credit_Card_Details = request.form.get('Credit_Card_Details')
    Birth_Date = request.form.get('Birth_Date')
    Exercise_History = request.form.get('Exercise_History')
    Fitness_Level = request.form.get('Fitness_Level')
    Medical_Heath = request.form.get('Medical_Heath')
    Personal_Trainer_ID = request.form.get('Personal_Trainer_ID')
    PT_Last_Name = request.form.get('PT_Last_Name')
    PT_First_Name = request.form.get('PT_First_Name')
    PT_Address = request.form.get('PT_Address')
    PT_City = request.form.get('PT_City')
    PT_Phone = request.form.get('PT_Phone')
    PT_State = request.form.get('PT_State')
    PT_Zip = request.form.get('PT_Zip')
    PT_Email = request.form.get('PT_Email')
    Level = request.form.get('Level')
    Price_Per_Month = request.form.get('Price_Per_Month')
    pgconnect.insertDeo(Customer_ID,Last_Name,First_Name,Address,City,State,Zip,Phone,Email_Addre
    return "{'success':'true','mess':'更新成功'}"
```

将获取到的这些数据作为参数传进insert方法转成sql语句并执行(pgconnect.py)



至此我们便实现了简单的交互