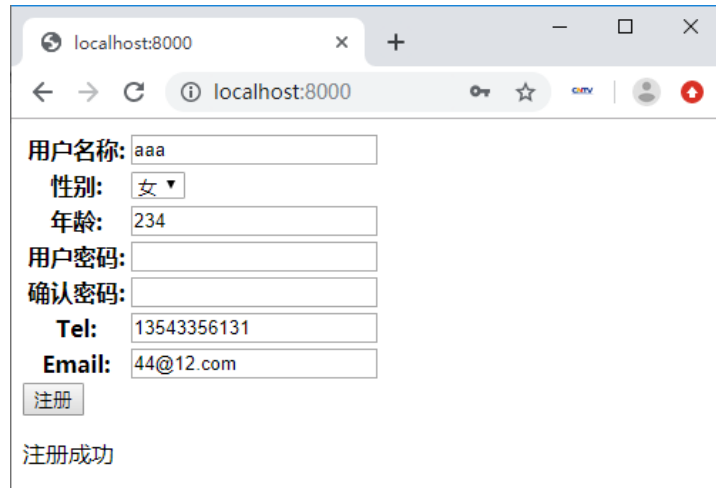


表单 综合实训

一、项目：用户注册



localhost:8000

← → ↻ ⓘ localhost:8000

用户名称:

性别:

年龄:

用户密码:

确认密码:

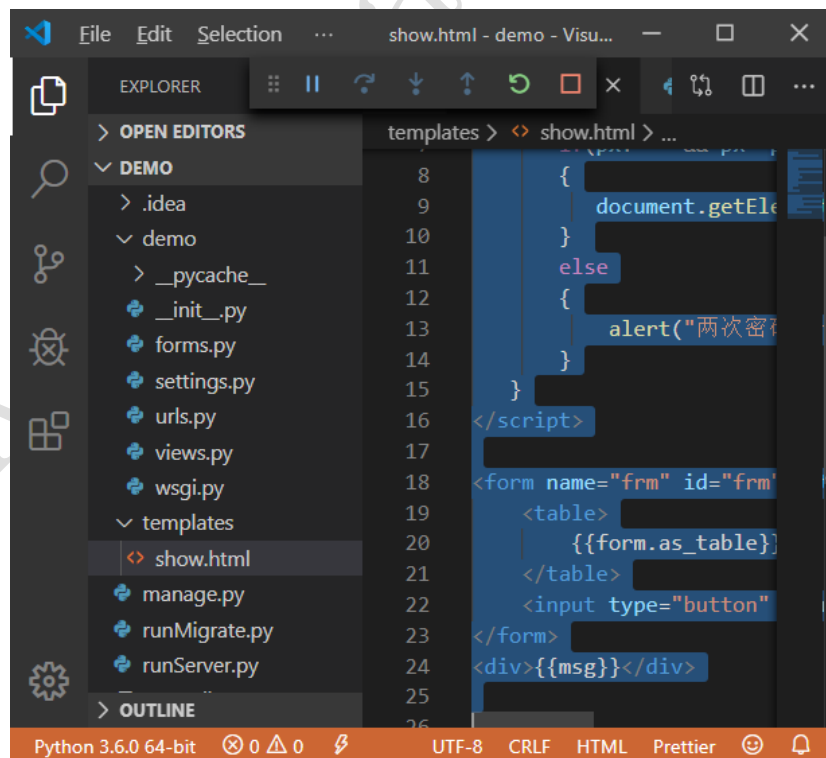
Tel:

Email:

注册成功

注册的用户写到数据库 `users.db`

二、程序设计



```
File Edit Selection ... show.html - demo - Visu...
EXPLORER
OPEN EDITORS
DEMO
  .idea
  demo
    __pycache__
    __init__.py
    forms.py
    settings.py
    urls.py
    views.py
    wsgi.py
  templates
    show.html
  manage.py
  runMigrate.py
  runServer.py
OUTLINE
Python 3.6.0 64-bit 0 0 UTF-8 CRLF HTML Prettier

8
9 {
10 document.getEle
11 }
12 else
13 {
14 alert("两次密
15 }
16 }
17 </script>
18 <form name="frm" id="frm
19 <table>
20 {{form.as_table}}
21 </table>
22 <input type="button"
23 </form>
24 <div>{{msg}}</div>
25
26
```

目录结构

(*) demo\urls.py

```
from django.conf.urls import url
import demo.views

urlpatterns = [
    url(r'^$', person.views.show),
]
```

(*) demo/forms.py

```
from django import forms
from django.core.validators import RegexValidator
from django.core.exceptions import ValidationError
import re

def tel_validate(value):
    tel = re.compile(r'^(13[0-9]|15[012356789]|17[678]|18[0-9]|14[57])[0-9]{8}$')
    if not tel.match(value):
        raise ValidationError('手机号码格式错误')

class RegisterForm(forms.Form):
    SEX=[("M", "男"), ("F", "女")]
    user=forms.CharField(label="用户名称", required=True, max_length=8)
    sex=forms.ChoiceField(label="性别", choices=SEX, initial="M")
    age=forms.IntegerField(label="年龄", validators=[RegexValidator(r"\d{1,2}", "年龄是 1-2 位数值")])
    pwd=forms.CharField(label="用户密码", required=True, max_length=8, widget=forms.PasswordInput(attrs={"id": "pwd"}))
    pwdConfirm=forms.CharField(label="确认密码", required=True, widget=forms.PasswordInput(attrs={"id": "pwdConfirm"}), max_length=8)
    tel=forms.CharField(label="Tel", required=False, max_length=11, validators=[tel_validate])
    email=forms.EmailField(label="Email", required=False)
```

(*) demo/views.py

```
from django.shortcuts import render
from demo.forms import RegisterForm
```

```
import sqlite3

class UserDB:
    def initialize(self):
        try:
            self.cursor.execute("drop table users")
        except:
            pass
        self.cursor.execute("create table users (user varchar(16) primary key,pwd varchar(16),sex varchar(16),age int,tel varchar(16),email varchar(64))")

    def open(self):
        self.con=sqlite3.connect("users.db")
        self.cursor=self.con.cursor()
        #self.initialize()

    def close(self):
        self.con.commit()
        self.con.close()

    def insert(self,user,pwd,sex,age,tel,email):
        flag=True
        try:
            sql="insert into users (user,pwd,sex,age,tel,email) values (?,?,?,?,?,?)"
            self.cursor.execute(sql,[user,pwd,sex,age,tel,email])
        except Exception as err:
            print(err)
            flag=False
        return flag

def show(request):
    msg=""
    if request.method=="POST":
        form=RegisterForm(request.POST)
        if(form.is_valid):
            user=request.POST.get("user")
            sex=request.POST.get("sex")
            age=request.POST.get("age","0")
            age=int(age)
            pwd=request.POST.get("pwd")
            tel=request.POST.get("tel")
            email=request.POST.get("email")
```

```

        DB=UserDB()
        DB.open()
        if(DB.insert(user,pwd,sex,age,tel,email)):
            msg="注册成功"
        else:
            msg="注册失败"
        DB.close()
    else:
        form=RegisterForm()
    return render(request,"show.html",locals())

```

(*) templates\show.html

```

<script>
    function trySubmit()
    {
        var px=document.getElementById("pwd").value;
        var py=document.getElementById("pwdConfirm").value;
        if(px!="" && px==py)
        {
            document.getElementById("frm").submit();
        }
        else
        {
            alert("两次密码不一致!"); return false;
        }
    }
</script>

<form name="frm" id="frm" action="" method="post">
    <table>
        {{form.as_table}}
    </table>
    <input type="button" value="注册" onclick="trySubmit()">
</form>
<div>{{msg}}</div>

```

```

TEMPLATES = [
    {

```

```
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [os.path.join(BASE_DIR, "templates")],
'APP_DIRS': True,
'OPTIONS': {
    'context_processors': [
        'django.template.context_processors.debug',
        'django.template.context_processors.request',
        'django.contrib.auth.context_processors.auth',
        'django.contrib.messages.context_processors.messages',
    ],
},
],
```

R. J. Huang 2022