

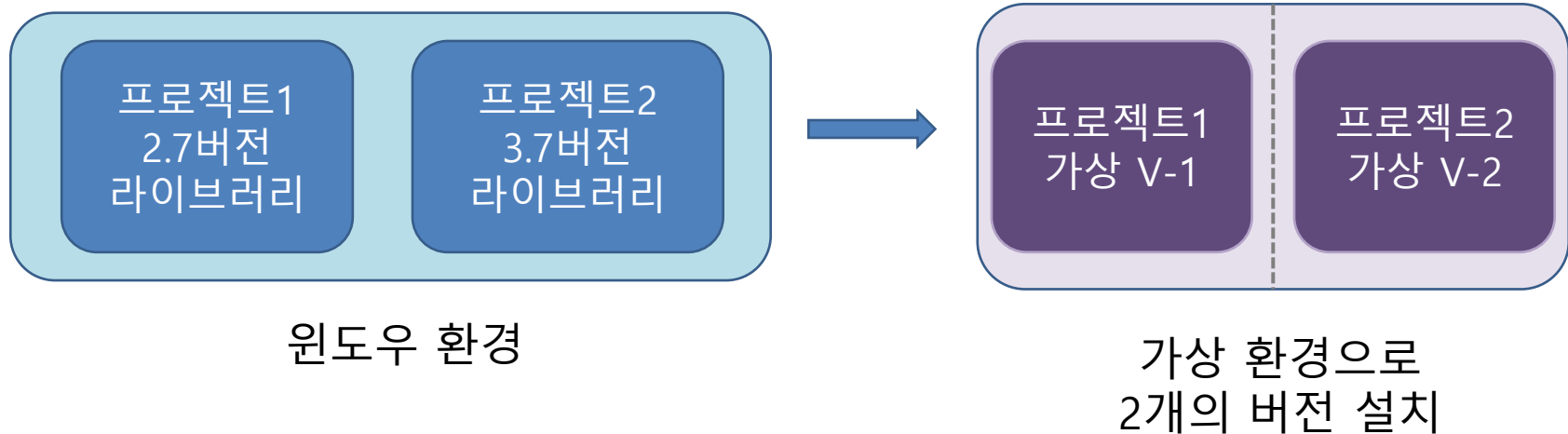
# 회원 관리 사이트 1



# 가상화(venv)

## ❖ 가상화

파이썬 가상 환경은 파이썬 프로젝트를 진행할 때 독립된 환경을 제공한다.  
파이썬 버전이나 라이브러리 버전이 다른 경우 하나의 데스크탑에서 개발할 경우 문제가 발생할 수 있다.



# 가상화 모듈 venv 설치

## ❖ Flask 개발 환경 구축

1. venvs 디렉터리와 하위에 myflask 디렉터리를 생성
2. 파이썬 모듈 중 venv를 현재 폴더에 설치(끝에 한칸 띄고 점)

```
C:\venvs\myflask>python -m venv .
```

3. 가상 환경에 진입하기

```
C:\venvs\myflask\Scripts>activate  
(myflask) C:\venvs\myflask\Scripts>
```

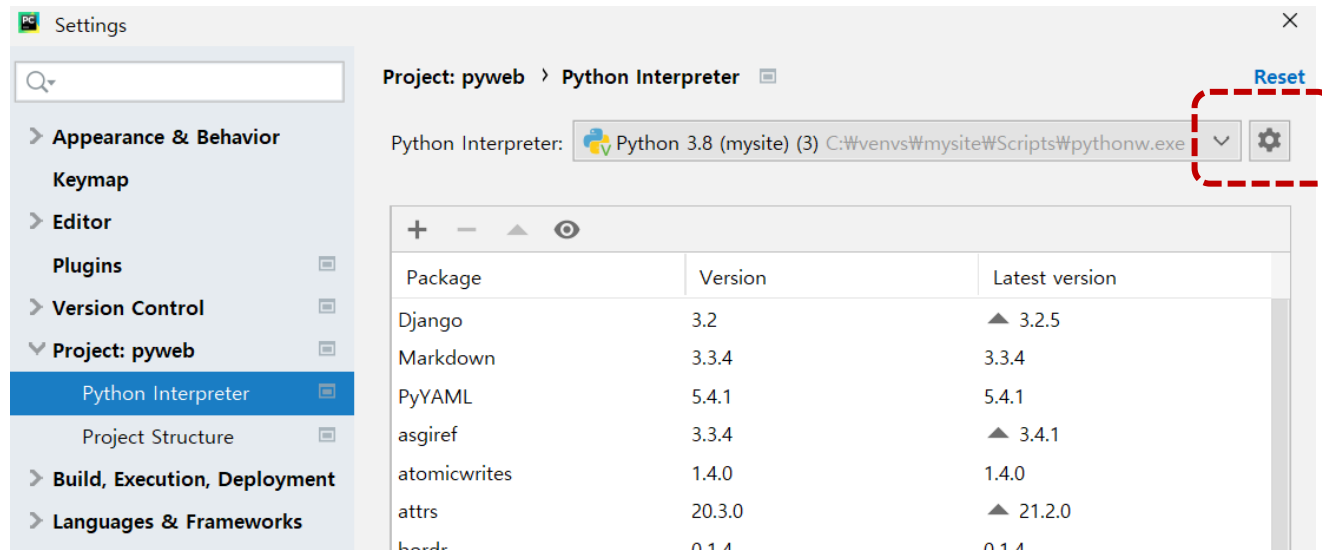
4. flask 모듈 설치하기

```
(myflask) C:\venvs\myflask\Scripts>pip install flask
```

# 파이썬 인터프리터 설정

## ❖ 파이썬 인터프리터 설정

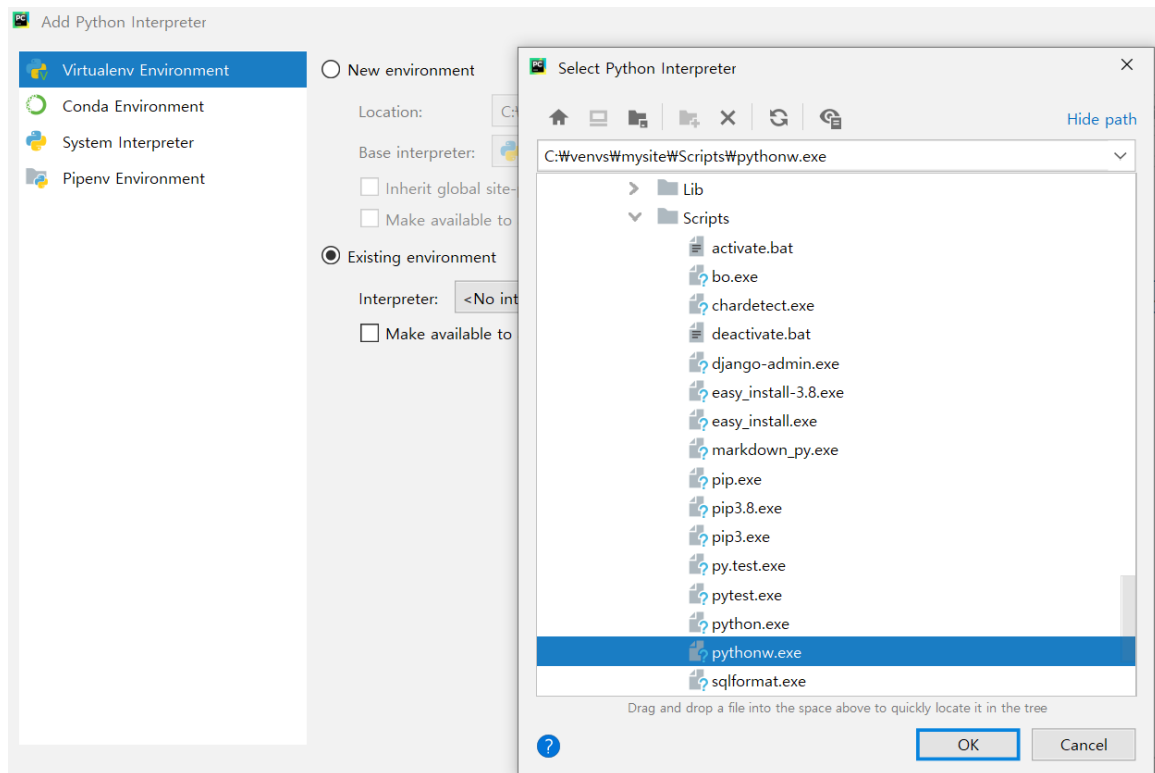
1. File > Open > webproject/member 선택
2. 파이썬 인터프리터 위치를 가상 환경 위치로 바꾸기  
[File-Settings] > [Project:myflask – Project Interpreter]  
오른쪽 위 설정 > add



# 파이썬 인터프리터 설정

## ❖ 파이썬 인터프리터 설정

[ Existing environment ] – venvs/myflask/Scripts/python.exe 선택



# 회원 관리 사이트

[Home](#)

[로그인](#)

[회원가입](#)

[회원목록](#)

안녕하세요~ 회원 커뮤니티 사이트입니다.



오전 5:55:26

# app.py 만들기

❖ app.py – 127.0.0.1:5000 접속



A screenshot of a web browser window. The address bar shows the URL '127.0.0.1:5000'. The main content area displays the text 'Welcome~ 방문해주셔서 감사합니다.' in a large, bold, black font.

```
from flask import Flask

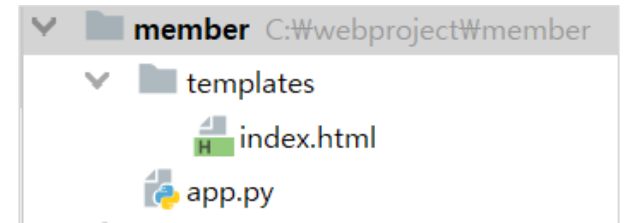
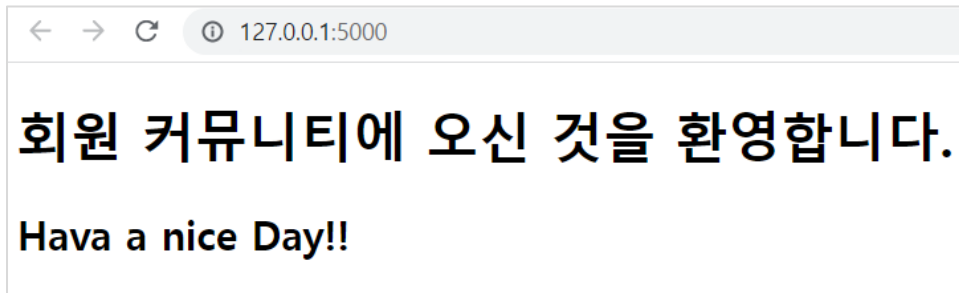
app = Flask(__name__)

@app.route('/')
def index():
    return "<h1>Welcome~ 방문해주셔서 감사합니다.</h1>"

app.run(debug=True)
```

# 템플릿 디렉터리 생성

## ❖ templates 디렉터리 생성 – index.html 만들기



```
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def index():
    #return "<h1>Welcome~ 방문해주셔서 감사합니다.</h1>"
    return render_template('index.html')
```



# index.html

```
<title>Welcome~ mysite!!</title>
<link rel="stylesheet" href="{ { url_for('static', filename='css/style.css') } }">
<link rel="shortcut icon" href="data:image/x-icon; " type="image/x-icon">
</head>
<body>
  <div id="container">
    <header>
      <nav>
        <ul>
          <li><a href="{ { url_for('index') } }">Home</a></li>
          <li><a href="#">로그인</a></li>
          <li><a href="#">회원가입</a></li>
          <li><a href="#">게시판</a></li>
        </ul>
      </nav>
    </header>
```

# index.html

```
<section>
  <h1>안녕하세요~ 회원 커뮤니티 사이트입니다.</h1>
  </p>
</section>
</div>

<script src="{{ url_for('static', filename='js/main_js.js') }}"></script>
</body>
</html>
```

# style.css

## ❖ static 디렉터리 생성 – style.css 만들기

```
/* 전체 스타일 */
*{margin: 0; padding: 0}
#container{width: 1000px; margin: 10px auto;}
header{height: 100px;}
nav ul{list-style: none; text-align: right; margin-right: 150px;}
nav ul li{display: inline-block; margin: 10px;
    border-top: 3px solid #333; padding: 10px}
nav ul li a{text-decoration: none; color: #000;}
nav ul li:hover{background-color: #eee}

/* index 페이지 스타일 */
section{text-align: center}
section h1{font-size: 2em; margin-bottom: 20px}
section img{margin-bottom: 10px;}

/* memberlist */
section table{width: 600px; margin: 0 auto; border-collapse: collapse}
section table, th, td{border: 1px solid #ccc; padding: 10px}
section table th{background-color: #eee;}
section table a{color: #000}
```

# style.css

```
/* member_view 스타일 */
section .tbl_view{width: 450px;}
section .tbl_view input{width: 200px; height: 30px;}
section .tbl_view a{width: 200px; height: 30px; color:#00f}

/* register */
section .join{width: 550px; margin: 0 auto;}
section .join fieldset{border:1px solid #555; border-radius: 3px}
section .join ul{list-style: none; padding: 10px;}
section .join ul li{margin: 15px;}
section .join ul li label{width: 120px; float: left; text-align: right}
section .join ul li input[type='text'], input[type='password']{width: 250px;
    height: 30px; border:1px solid #555; border-radius: 3px}
section .join ul li input[type='button'], input[type='reset']{width: 100px;
    height: 30px; border:1px solid #555; border-radius: 3px}
```

# main\_js.js

## ❖ main\_js.js 만들기

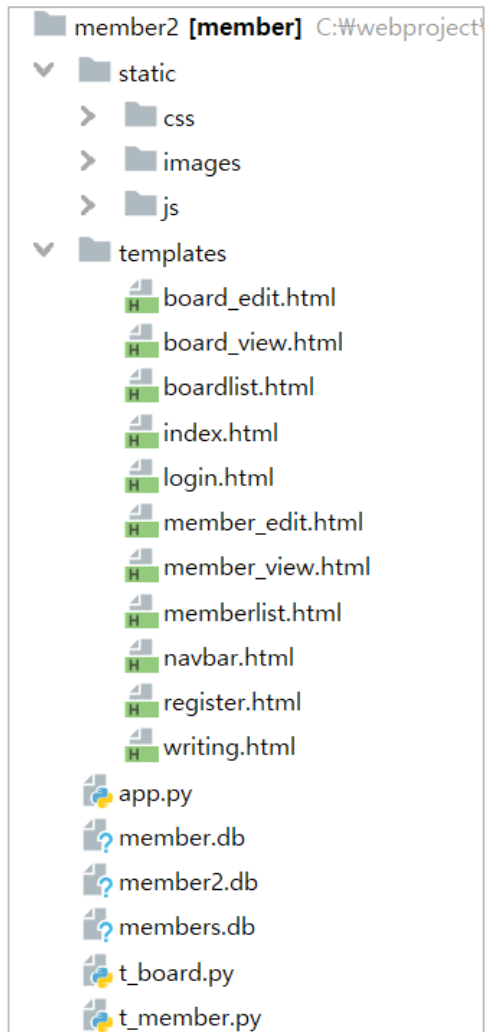
```
pic = document.getElementById('pic')
pic.onmouseover = changePic;
pic.onmouseout = originPic;

function changePic(){
    pic.src = "../static/images/coffee-pink.jpg";
}
function originPic(){
    pic.src = "../static/images/coffee-blue.jpg";
}

// 디지털 시계
setInterval(myWatch, 1000);
function myWatch(){
    var date = new Date();
    var now = date.toLocaleTimeString();
    document.getElementById("demo").innerHTML = now;
}
```

# main\_js.js

## ❖ 플라스크 프레임워크 계층 구조도



# member 데이터베이스

member.db에 member 테이블 생성

파일이름 : t\_member.py

칼럼 이름	자료형	설명
mid	CHAR(5) PRIMARY KEY	회원번호
passwd	CHAR(8) NOT NULL	비밀번호
name	TEXT NOT NULL	이름
age	INTEGER	나이
regDate	TIMESTAMP DATE DEFAULT (datetime('now', 'localtime'))	작성일

# member 테이블

## member 테이블 생성

```
def create_table():
    conn = getconn()
    cur = conn.cursor()
    sql = """
    CREATE TABLE member(
        mid CHAR(5) PRIMARY KEY,
        passwd CHAR(8) NOT NULL,
        name TEXT NOT NULL,
        age INTEGER,
        regDate TIMESTAMP DATE DEFAULT (datetime('now', 'localtime'))
    )
    """
    cur.execute(sql)
    conn.commit()
    print("member table 생성")
    conn.close()
```

```
import sqlite3
```

```
def getconn():
    conn = sqlite3.connect('./member.db')
    return conn
```

한국시간 적용 : [https://www.sqlite.org/lang\\_datefunc.html](https://www.sqlite.org/lang_datefunc.html)



# 파이썬으로 테이블 만들기

## member 테이블에 자료 추가

```
def insert_member():  
    conn = getconn()  
    cur = conn.cursor()  
    sql = "INSERT INTO member (mid, passwd, name, age) VALUES (?, ?, ?, ?)"  
    cur.execute(sql, ('10002', 'm1234568', '팔쥐', 17))  
    conn.commit()  
    print("회원 추가")  
    conn.close()
```

regDate(등록일)는 자동 입력되므로 명시하지 않아야 함.

# member 테이블

## member 테이블에 자료 검색 – 전체 검색

```
def select_member():  
    conn = getconn()  
    cur = conn.cursor()  
    sql = "SELECT * FROM member"  
    cur.execute(sql)  
    rs = cur.fetchall()  
    for i in rs:  
        print(i)  
    conn.close()
```

```
('10001', 'm1234567', '콩쥐', 17, '2021-12-06 07:13:18')  
( '10002', 'm1234568', '팥쥐', 17, '2021-12-06 08:22:10')
```

# 회원 목록 페이지

## ▪ memberlist 페이지

### 회원 목록

번호	아이디	패스워드	이름	나이	가입일
1	<u>bts12</u>	m123456@	BTS	31	2022-02-17 04:51:47
2	<u>smile</u>	m123456@	박미소	35	2022-02-14 22:25:03
3	<u>cloud</u>	m123456@	구름	100	2021-12-13 19:12:23
4	<u>today.</u>	m123456@	김하나	21	2021-12-13 18:12:45

# 회원 목록 페이지

## ▪ memberlist.html

```
{% include 'navbar.html' %}
<section>
  <h1>회원 목록</h1>
  <table>
    <tr>
      <th>번호</th><th>아이디</th><th>패스워드</th><th>이름</th><th>나이</th><th>가입일</th>
    </tr>
    {% for i in rs %}
      <tr>
        <td>{{ loop.index }}</td>
        <td><a href="{% url_for('member_view', id=i[0]) %}">{{ i[0] }}</a></td>
        <td>{{ i[1] }}</td>
        <td>{{ i[2] }}</td>
        <td>{{ i[3] }}</td>
        <td>{{ i[4] }}</td>
      </tr>
    {% endfor %}
  </table>
</section>
```

## 회원 목록 페이지

- url 경로 설정 – /memberlist/ 및 memberlist() 함수

```
@app.route('/memberlist/')
def memberlist():    # 회원 목록
    conn = getconn()
    cur = conn.cursor()
    sql = "SELECT * FROM member ORDER BY regDate DESC" #가입일 기준 내림차순
    cur.execute(sql)
    rs = cur.fetchall()
    conn.close()

    return render_template('memberlist.html', rs=rs)
```

# navbar.html 만들기

- navbar.html을 외부 파일로 포함하기

```
{% include 'navbar.html' %}
```

```
<header>
  <nav>
    <ul>
      <li><a href="{% url_for('index') %}">Home</a></li>
      <li><a href="{% url_for('login') %}">로그인</a></li>
      <li><a href="{% url_for('register') %}">회원가입</a></li>
      <li><a href="{% url_for('memberlist') %}">회원목록</a></li>
    </ul>
  </nav>
</header>
```

# 회원 상세 페이지

## ■ 회원 상세 페이지

회원 정보	
아이디	<input type="text" value="today"/>
비밀번호	<input type="password" value="....."/>
이름	<input type="text" value="김하나"/>
나이	<input type="text" value="21"/>
가입일	<input type="text" value="2021-12-13 18:12:45"/>
<a href="#">목록</a>	

## 회원 상세 페이지

- url 경로 설정 - /member\_view/<string:id>/

```
<td>{{ loop.index }}</td>  
<td><a href="{{ url_for('member_view', id=i[0]) }}">{{ i[0] }}</a></td>  
<td>{{ i[1] }}</td>
```

```
@app.route('/member_view/<string:id>/')  
def member_view(id): #회원 상세 - mid를 전달받음  
    conn = getconn()  
    cur = conn.cursor()  
    sql = "SELECT * FROM member WHERE mid = '%s' " % (id)  
    cur.execute(sql)  
    rs = cur.fetchone()  
    conn.close()  
    return render_template('member_view.html', rs=rs)
```

mid를 경로로 설정  
id를 매개변수로 전달



# 회원 상세 페이지

## ▪ member\_view.html

```
{% include 'navbar.html' %}
<section>
<h1>회원 정보</h1>
  <table class="tbl_view">
    <colgroup>
      <col style="background-color: #eee">
      <col>
    </colgroup>
    {% if rs %}
    <tr>
      <td>아이디</td>
      <td><input type="text" name="mid" value="{{ rs[0] }}" readonly></td>
    </tr>
    <tr>
      <td>비밀번호</td>
      <td><input type="password" name="passwd" value="{{ rs[1] }}" readonly></td>
    </tr>
```

# 회원 상세 페이지

## ▪ member\_view.html

```
<tr>
  <td>이름</td>
  <td><input type="text" name="name" value="{{ rs[2] }}" readonly></td>
</tr>
<tr>
  <td>나이</td>
  <td><input type="text" name="age" value="{{ rs[3] }}" readonly></td>
</tr>
<tr>
  <td>가입일</td>
  <td><input type="text" name="regDate" value="{{ rs[4] }}" readonly></td>
</tr>
<tr>
  <td colspan="2" style="background-color: #fff">
    <a href="{{ url_for('memberlist') }}" >목록</a>
  </td>
</tr>
{% endif %}
</table>
</section>
```

# 회원 가입 페이지

- register 페이지

## 회원 가입

아이디	<input type="text" value="5자만 입력해주세요"/>
비밀번호	<input type="password" value="4자에서 8자리까지 입력해주세요"/>
비밀번호 확인	<input type="password"/>
이름	<input type="text" value="필수 입력 항목입니다."/>
나이	<input type="text"/>
<input type="button" value="가입"/> <input type="button" value="취소"/>	

# 회원 가입 페이지

## ▪ register.html 페이지

```
<section>
<h1>회원 가입</h1>
  <form action="" method="post" class="join" name="regForm">
    <fieldset>
      <ul>
        <li>
          <label for="mid">아이디 </label>
          <input type="text" id="mid" name="mid"
            placeholder="5자만 입력해주세요">
        </li>
        <li>
          <label for="passwd1">비밀번호 </label>
          <input type="password" id="passwd1" name="passwd"
            placeholder="4자에서 8자리까지 입력해주세요">
        </li>
        <li>
          <label for="passwd2">비밀번호 확인</label>
          <input type="password" id="passwd2" name="passwd_confirm">
        </li>
      </ul>
    </fieldset>
  </form>
</section>
```

# 회원 가입 페이지

## ▪ register.html 페이지

```
<li>
  <label for="name">이름 </label>
  <input type="text" id="name" name="name"
    placeholder="필수 입력 항목입니다.">
</li>
<li>
  <label for="age">나이 </label>
  <input type="text" id="age" name="age">
</li>
<li>
  <input type="submit" value="가입">
  <input type="reset" value="취소">
</li>
</ul>
</fieldset>
```

# 회원 가입 페이지

- url 경로 설정 – /register/ , methods 리스트 – [ 'GET' , 'POST' ]

```
@app.route('/register/', methods = ['GET', 'POST'])
def register():
    if request.method == "POST":
        # 데이터 수집
        id = request.form['mid']
        pwd = request.form['passwd']
        name = request.form['name']
        age = request.form['age']
        # DB 처리
        conn = getconn() #db 연결
        cur = conn.cursor()
        sql = "INSERT INTO member (mid, passwd, name, age) VALUES " \
              "( '%s', '%s', '%s', %s )" % (id, pwd, name, age)
        cur.execute(sql)
        conn.commit()
        return redirect(url_for('memberlist'))
    else:
        return render_template('register.html')
```