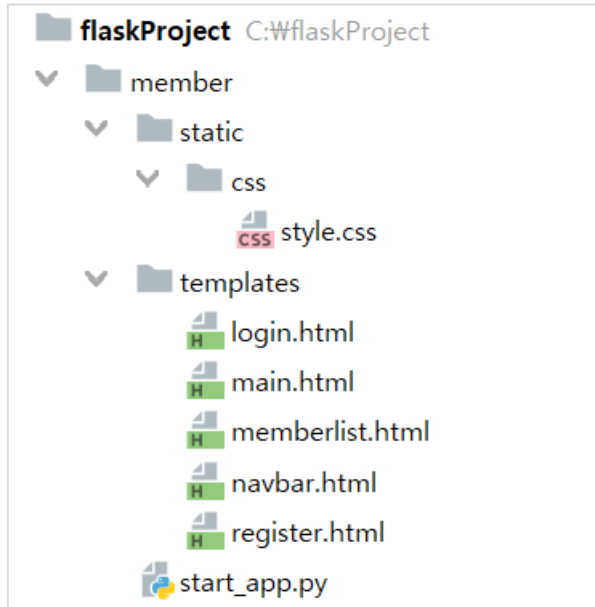


2장. 회원 관리 사이트 제작



회원 관리 사이트



Home | 로그인 | 회원가입 | 회원목록

로그인

아이디 :

패스워드 :

로그인

Home | 로그인 | 회원가입 | 회원목록

회원 가입

아이디

비밀번호

이름

나이

등록일 연도-월-일

가입

취소

회원 관리 사이트

▪ main 페이지

Home | 로그인 | 회원가입 | 회원목록

안녕하세요~ 회원 관리 사이트입니다.

```
from flask import Flask, render_template, request, redirect, session, url_for
import sqlite3

app = Flask(__name__)
app.secret_key = "abcde"  # 암호키 발행해야함

@app.route("/")
def main():
    if "userID" in session:  # login()에서 세션 발급
        return render_template("main.html", username=session.get("userID"), login = True)
    else:
        return render_template('main.html', login = False)
```

회원 관리 사이트

main 페이지

```
<head>
  <meta charset="UTF-8">
  <title>메인 페이지</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
</head>
<body>
  <nav>
    <a href="{{ url_for('main') }}"> Home | </a>
    {% if login %}
      <a href="{{ url_for('logout') }}">({{ username }}님) 로그아웃 | </a>
    {% else %}
      <a href="{{ url_for('login') }}"> 로그인 | </a>
    {% endif %}
    <a href="{{ url_for('register') }}"> 회원가입 | </a>
    <a href="{{ url_for('memberlist') }}"> 회원목록 </a>
  </nav>
  <h2>안녕하세요~ 회원 관리 사이트입니다.</h2>
  
</body>
```

회원 관리 사이트

■ Css 스타일

```
nav a{text-decoration: none; color: #000;}
```

```
/* memberlist 스타일 */
```

```
.tbl_list, th, td{border: 1px solid #ccc; border-collapse: collapse;}
```

```
.tbl_list th, td{padding: 15px; text-align: center}
```

```
/* member_view 스타일 */
```

```
.tbl_view, th, td{border: 1px solid #ccc; border-collapse: collapse; }
```

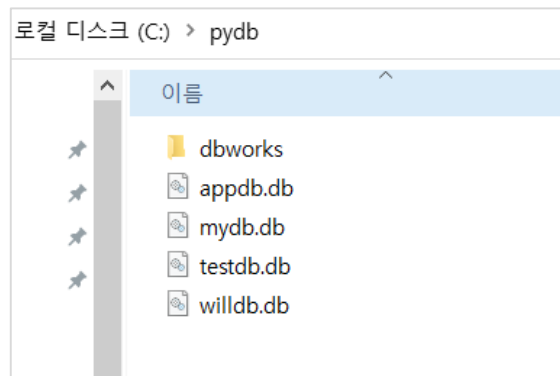
```
.tbl_view th, td{width: 120px; padding: 15px; text-align: left}
```

```
.tbl_view #align{text-align: center;}
```

회원 관리 사이트

▪ Sqlite3 데이터 베이스

```
sqlite3.connect("c:/pydb/testdb.db")
```



```
-- member 테이블 --  
CREATE TABLE member (  
    memberid CHAR(5) PRIMARY KEY,  
    passwd CHAR(8) NOT NULL,  
    name TEXT NOT NULL,  
    age INTEGER,  
    regdate DATETIME DEFAULT CURRENT_TIMESTAMP  
);
```

회원 관리 사이트

register 페이지

Home | 로그인 | 회원가입 | 회원목록


회원 가입

아이디

비밀번호

이름

나이

등록일 

```
<h2>회원 가입</h2>
<form action="{{ url_for('register') }}" method="post">
  <p>
    <label>아이디 </label>
    <input type="text" name="memberid">
  </p>
  <p>
    <label>비밀번호 </label>
    <input type="password" name="passwd">
  </p>
  <p>
    <label>이름 </label>
    <input type="text" name="name">
  </p>
  <p>
    <label>나이 </label>
    <input type="text" name="age">
  </p>
  <p>
    <label>등록일 </label>
    <input type="date" name="regDate">
  </p>
  <input type="submit" value="가입">
</form>
```

회원 관리 사이트

▪ register 페이지

```
@app.route("/register", methods = ['GET', 'POST'])
def register():
    if request.method == 'POST':
        id = request.form['memberid']
        pwd = request.form['passwd']
        name = request.form['name']
        age = request.form['age']
        regDate = request.form['regDate']

        conn = sqlite3.connect("c:/pydb/testdb.db")
        cur = conn.cursor()
        sql = "INSERT INTO member VALUES ('%s', '%s', '%s', '%s', '%s')" % (id, pwd, name, age, regDate)
        cur.execute(sql)
        conn.commit()
        conn.close()
        return redirect(url_for('memberlist'))
    else:
        return render_template("register.html")
```


회원 관리 사이트

▪ memberlist 페이지

Home | 로그인 | 회원가입 | 회원목록 | 회원수정

회원 목록

번호	아이디	패스워드	이름	나이	가입일
1	10001	a1234567	이강인	21	2021-09-28
2	10002	b1234567	안산	20	2021-09-28
3	1003	c11112222	김연아	31	2021-09-01
4	10003	c11112222	추신수	40	2021-09-01

회원 관리 사이트

▪ memberlist 페이지

```
@app.route("/memberlist")
def memberlist():
    conn = sqlite3.connect("c:/pydb/testdb.db")
    cur = conn.cursor()
    sql = "SELECT * FROM member"
    cur.execute(sql)
    rs = cur.fetchall()
    conn.close()
    return render_template('memberlist.html', rs=rs)
```

회원 관리 사이트

▪ memberlist 페이지

```
{% include 'navbar.html' %}
<h3>회원 목록</h3>
<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>
```

회원 관리 사이트

- navbar.html을 외부 파일로 포함하기

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

```
<nav>
  <a href="{% url_for('main') %}"> Home | </a>
  {% if login %}
    <a href="{% url_for('logout') %}">({{ username }}님) 로그아웃 | </a>
  {% else %}
    <a href="{% url_for('login') %}"> 로그인 | </a>
  {% endif %}
  <a href="{% url_for('register') %}"> 회원가입 | </a>
  <a href="{% url_for('memberlist') %}"> 회원목록 </a>
</nav>
```

```
<body>
  {% include 'navbar.html' %}
  <h2>안녕하세요~ 회원 관리 사이트입니다.</h2>
</body>
```

회원 관리 사이트

▪ login 페이지

[Home](#) | [로그인](#) | [회원가입](#) | [회원목록](#)

로그인

아이디 :

패스워드 :

[Home](#) | [로그인](#) | [회원가입](#) | [회원목록](#)

로그인

아이디나 비밀번호가 일치하지 않습니다.

아이디 :

패스워드 :

[Home](#) | [\(10001님\) 로그아웃](#) | [회원가입](#) | [회원목록](#)

안녕하세요~ 회원 관리 사이트입니다.

회원 관리 사이트

login 페이지

```
@app.route("/login", methods = ['GET', 'POST'])
def login():
    if request.method == 'POST':
        id = request.form['memberid']
        pwd = request.form['passwd']

        conn = sqlite3.connect("c:/pydb/testdb.db")
        cur = conn.cursor()
        sql = "SELECT * FROM member WHERE memberid = '%s' AND passwd = '%s' " % (id, pwd)
        cur.execute(sql)
        rs = cur.fetchone()
        if rs:
            session['userID'] = id #세션 발급
            return redirect(url_for('main'))
        else:
            error = "아이디나 비밀번호가 일치하지 않습니다."
            return render_template("login.html", error=error)
    else:
        return render_template("login.html")
```

app.secret_key = "abcde" #암호키 발행해야함

회원 관리 사이트

■ 비밀키 설정

```
app.config.update(  
    TESTING=True,  
    SECRET_KEY=b'_5#y2L"F4Q8z\n\xec]/'  
)
```

Q.

Why does Flask force us to set this secret_key property?

> **secret key 설정 값을 Flask가 설정토록 강제하는 이유가 무엇인가?**

A.

Anything that requires encryption (for safe-keeping against tampering by attackers) requires the secret key to be set. For just Flask itself, that 'anything' is the Session object

> **공격자로부터의 방어를 위한 암호화를 요구하는 어떤 것이든지 비밀키 값이 세팅되도록 요구된다. 예를 들어 Flask의 경우, 앞서 언급한 '어떤 것이든지'는 Session 객체이다.**

회원 관리 사이트

▪ login 페이지

```
<h2>로그인</h2>
<form action="{ url_for('login') }" method="post">
    {% if error %}
    {{error}}
    {% endif %}
    <p><label>아이디 : </label>
        <input type="text" name="memberid"></p>
    <p><label>패스워드 : </label>
        <input type="password" name="passwd"></p>
        <input type="submit" value="로그인">
</form>
```


회원 관리 사이트

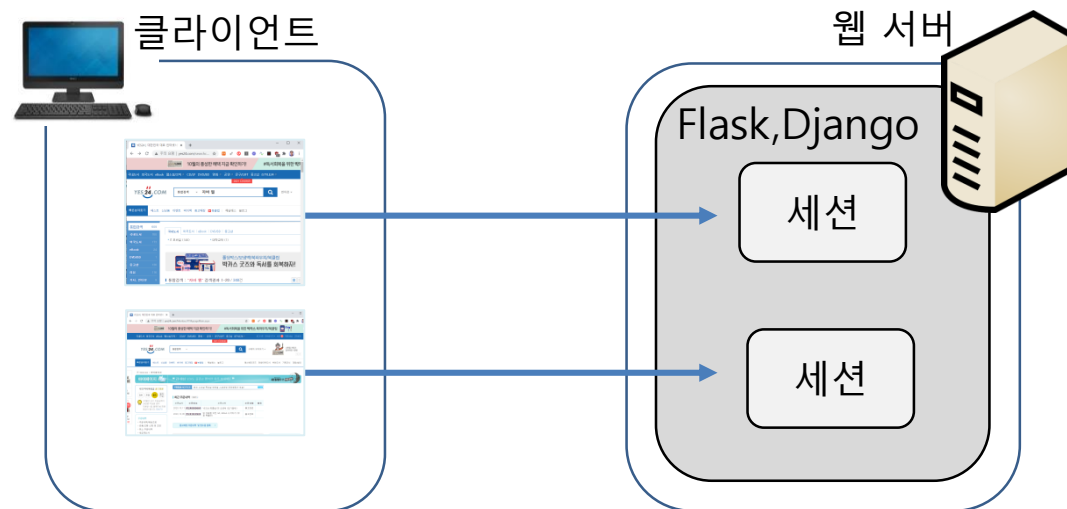
▪ logout 페이지

```
@app.route('/logout')
def logout():
    session.pop("userID") # 세션 삭제
    return redirect(url_for('main'))
```

세션(session)

세션(session)

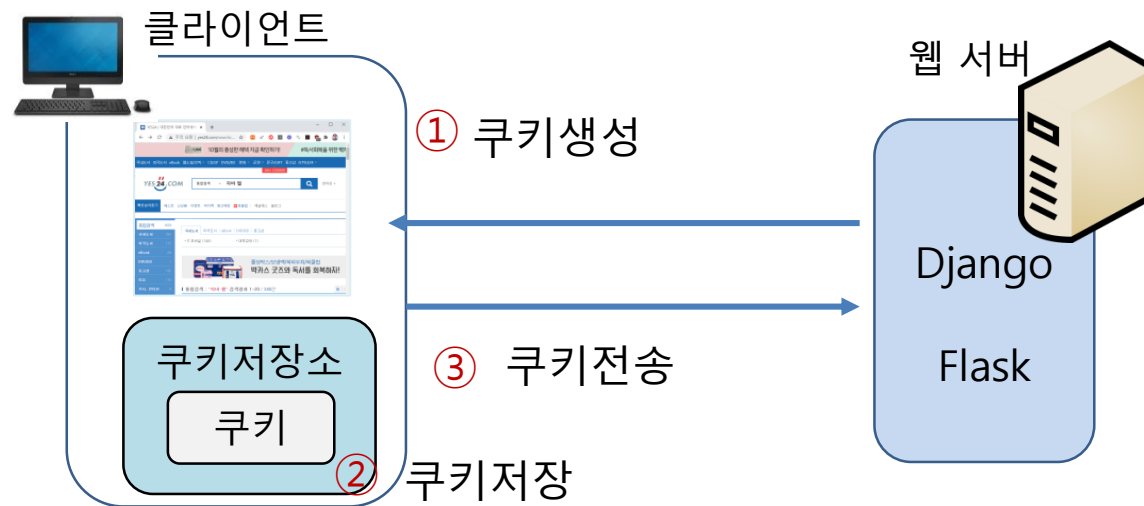
- 클라이언트와 웹 서버 간의 상태를 지속적으로 유지하는 방법을 말한다.
- 사용자 인증을 통해 특정 페이지를 사용할 수 있도록 권한 상태 유지.
- 예) 웹 쇼핑몰 - 장바구니나 주문 처리와 같은 회원 전용 페이지 로그인 후 다른 웹 페이지에 갔다가 돌아와도 로그인 상태 유지됨
- 세션은 오직 웹 서버에 존재하는 객체로 웹 브라우저마다 **하나씩** 존재하므로 브라우저를 닫기 전까지 웹 페이지를 이동하더라도 사용자 정보가 유지된다.



쿠키(cookie)

쿠키(cookie)

- 클라이언트와 웹 서버간의 상태를 지속적으로 유지하는 방법으로 쿠키와 세션이 있다.
- 쿠키는 세션과 달리 상태 정보를 웹 서버가 아닌 클라이언트에 저장한다.
예) 어떤 웹 사이트를 처음 방문한 사용자가 로그인 인증을 하고 나면 아이디와 비밀번호를 기록한 쿠키가 만들어지고, 그 다음부터 사용자가 그 사이트에 접속하면 별도의 절차를 거치지 않고 쉽게 접속할 수 있다.
- 쿠키는 클라이언트의 일정 폴더에 정보를 저장하므로 웹 서버의 부하를 줄일 수 있으나 개인 정보 기록이 남기 때문에 보안에 문제가 있다.



Session & Cookie

◆ 세션과 쿠키 확인

개발자도구 > Network

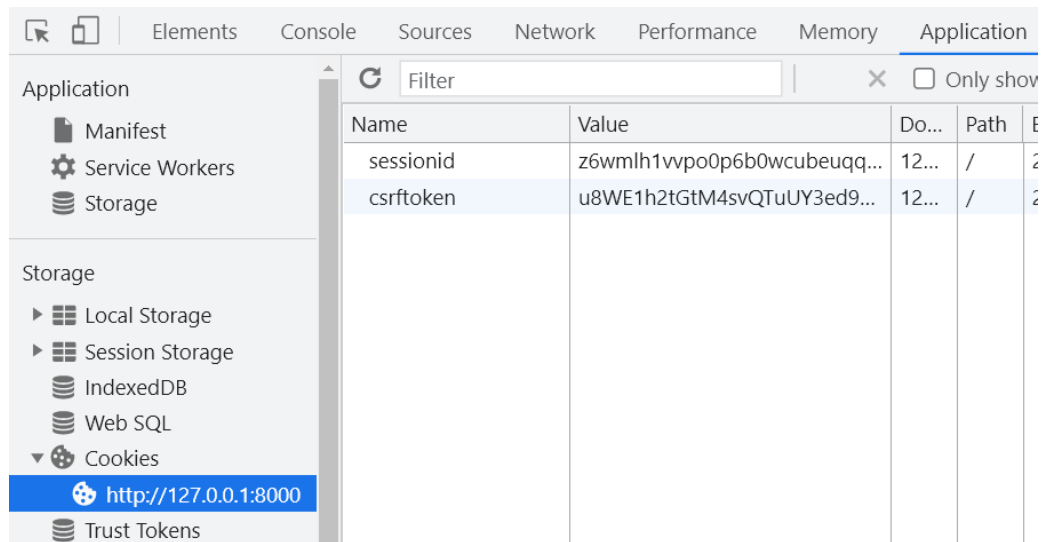
The screenshot shows the Chrome DevTools Network tab. The top bar includes tabs for Elements, Console, Sources, Network, Performance, Memory, Application, Security, and Lighthouse. The Network tab is active, displaying a list of requests. The first request is selected, showing its details in the right pane. The request is to 127.0.0.1 for style.css. The request headers are expanded, showing the following information:

- Accept:** text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exch
- Accept-Encoding:** gzip, deflate, br
- Accept-Language:** ko,en;q=0.9,ja;q=0.8,en-US;q=0.7,ko-KR;q=0.6
- Cache-Control:** max-age=0
- Connection:** keep-alive
- Cookie:** csrftoken=HT00n67IxYLWw5SRZFHEbZOu8M9KA1Hqi2gEXu78ydQ0p7WrzNyI7UFktQDjNk1X; session=eyJ1c2VySUQiOiIxMDAwMiJ9.YVI6Fg.hjW3
- Host:** 127.0.0.1:5000
- Referer:** http://127.0.0.1:5000/login

Session & Cookie

◆ 세션과 쿠키 확인

개발자도구 > Application



Session

◆ navbar.html 변경

로그인전

Home | 로그인 | 회원가입 | 게시판 |

안녕하세요~ 회원 관리 사이트입니다.

로그인후

Home | (10002님) 로그아웃 | 회원목록 | 게시판 |

안녕하세요~ 회원 관리 사이트입니다.

Session

◆ navbar.html 변경

```
<nav>
    {% if login %}
        <a href="{{ url_for('main') }}"> Home | </a>
        <a href="{{ url_for('logout') }}">({{ username }}님) 로그아웃 | </a>
        <a href="{{ url_for('memberlist') }}"> 회원목록 | </a>
        <a href="#"> 게시판 | </a>
    {% else %}
        <a href="{{ url_for('main') }}"> Home | </a>
        <a href="{{ url_for('login') }}"> 로그인 | </a>
        <a href="{{ url_for('register') }}"> 회원가입 | </a>
        <a href="#"> 게시판 | </a>
    {% endif %}
</nav>
```

Session

◆ memberlist 세션 발급


```
@app.route("/memberlist")
def memberlist():
    conn = getconn()
    cur = conn.cursor()
    sql = "SELECT * FROM member"
    cur.execute(sql)
    rs = cur.fetchall()
    conn.close()
    if "userID" in session:
        return render_template('memberlist.html', username=session.get("userID"), login = True, rs=rs)
    else:
        return render_template('memberlist.html', login = False)
```


회원 관리 사이트

■ 회원 상세 페이지

Home | (10002님) 로그아웃 | 회원목록 | 게시판 |

회원 정보

아이디	<input type="text" value="20001"/>
패스워드	<input type="password" value="....."/>
이름	<input type="text" value="홍부"/>
나이	<input type="text" value="35"/>
가입일	<input type="text" value="2021-09-29"/> 

[목록](#)

회원 관리 사이트

■ 회원 상세 페이지

```
@app.route('/member_view/<string:id>')
def member_view(id):
    conn = getconn()
    cur = conn.cursor()
    sql = "SELECT * FROM member WHERE memberid = '%s' " % (id)
    cur.execute(sql)
    rs = cur.fetchone()
    print(rs)
    conn.close()
    if "userID" in session:
        return render_template('member_view.html', username=session.get("userID"), login = True, rs=rs)
    else:
        return render_template('member_view.html', login = False)
```

← memberid를 경로로
설정
id를 매개변수로 전달

회원 관리 사이트

▪ member_view.html

```
{% include 'navbar.html' %}
<h3>회원 정보</h3>
<table class="tbl_view">
  {% if rs %}
  <tr>
    <td>아이디</td>
    <td><input type="text" name="memberid" value="{{ rs[0] }}"></td>
  </tr>
  <tr>
    <td>패스워드</td>
    <td><input type="password" name="passwd" value="{{ rs[1] }}"></td>
  </tr>
  <tr>
    <td>이름</td>
    <td><input type="text" name="name" value="{{ rs[2] }}"></td>
  </tr>
  <tr>
    <td>나이</td>
    <td><input type="text" name="age" value="{{ rs[3] }}"></td>
  </tr>
```

회원 관리 사이트

▪ member_view.html


```
<tr>
  <td>가입일</td>
  <td><input type="date" name="regdate" value="{{ rs[4] }}"></td>
</tr>
<tr>
  <td colspan="2" id="align">
    <a href="{{ url_for('memberlist') }}">목록</a>
    {% if session['userID'] == rs[0] %}
    <a onclick="return confirm('정말로 탈퇴하겠습니까?')"
      href="{{ url_for('member_delete', id=rs[0]) }}">탈퇴</a>
    {% endif %}
  </td>
</tr>
{% endif %}
</table>
```

회원 탈퇴

◆ 목록보기와 탈퇴 메뉴

Home | (10002님) 로그아웃 | 회원목록 | 게시판 |

회원 정보

아이디	<input type="text" value="10002"/>
패스워드	<input type="password" value="....."/>
이름	<input type="text" value="안산"/>
나이	<input type="text" value="20"/>
가입일	<input type="text" value="2021-09-28"/> 
목록 탈퇴	

127.0.0.1:5000 내용:
정말로 탈퇴하겠습니까?

확인

취소

회원 탈퇴

◆ 탈퇴 함수 – member_delete

```
@app.route('/member_delete/<string:id>')
def member_delete(id):
    conn = getconn()
    cur = conn.cursor()
    sql = "DELETE FROM member WHERE memberid = '%s' " % (id)
    cur.execute(sql)
    conn.commit()
    conn.close()
    return redirect(url_for('memberlist'))
```