

회원 관리 -게시판 구축



게시판 만들기

[Home](#)[로그인](#)[회원가입](#)[게시판](#)

게시판 목록

번호	제목	작성일	글쓴이
6	플라스크에 게시판 달기	2021-12-13 06:47:22	강하늘
5	글쓰기 유효성	2021-12-13 04:52:01	김하나
3	최강 추위	2021-12-13 04:35:24	김하나
2	한 해가 저물어간다.	2021-12-12 18:09:15	강하늘
1	겨울속으로	2021-12-12 17:43:42	김하나

[글쓰기](#)

board 테이블 생성

■ member.db에 board 테이블 생성

칼럼 이름	자료형	설명
bno	INTEGER PRIMARY KEY AUTOINCREMENT	글번호
title	TEXT NOT NULL	글제목
content	TEXT NOT NULL	글내용
create_date	TIMESTAMP DATE DEFAULT (datetime('now', 'localtime'))	작성일
mid	FOREIGN KEY(mid) REFERENCES member(mid) ON DELETE CASCADE	글쓴이

board 테이블 생성

▪ member.db에 board 테이블 생성

```
def create_table():
    conn = getconn()
    cur = conn.cursor()
    sql = """
    CREATE TABLE board(
        bno INTEGER PRIMARY KEY AUTOINCREMENT,
        title TEXT NOT NULL,
        content TEXT NOT NULL,
        createDate TIMESTAMP DATE DEFAULT (datetime('now', 'localtime')),
        mid CHAR(5) NOT NULL,
        FOREIGN KEY(mid) REFERENCES member(mid) ON DELETE CASCADE
    );
    """
    cur.execute(sql)
    conn.commit()
    print("board 테이블 생성!")
    conn.close()
```

```
import sqlite3 as sql
```

```
def getconn():
    conn = sql.connect("./member.db")
    return conn
```

게시판 목록

- `/boardlist/` 경로 및 `boardlist()` 함수

```
@app.route('/boardlist/')
def boardlist():
    conn = getconn()
    cur = conn.cursor()
    sql = "SELECT * FROM board ORDER BY bno DESC"
    cur.execute(sql)
    rs = cur.fetchall()
    conn.close()
    return render_template('boardlist.html', rs=rs)
```

게시판 목록

■ boardlist.html

```
<body>
  <div id="container">
    {% include 'navbar.html' %}
    <section>
      <h1>게시판 목록</h1>
      <table class="tbl_board">
        <tr>
          <th>번호</th><th>제목</th><th>작성일</th><th>글쓴이</th>
        </tr>
        {% for i in rs %}
          <tr>
            <td>{{ i[0] }}</td>
            <td><a href="{ url_for('board_view', bno=i[0]) }">{{ i[1] }}</a></td>
            <td>{{ i[3] }}</td>
            <td>{{ i[4] }}</td>
          </tr>
        {% endfor %}
      </table>
```

게시판 목록

▪ boardlist.html

```
        </table>
        <div class="btn_write">
            <input type="button" value="글쓰기" onclick="goWriting()">
        </div>
    </section>
</div>
<script>
    function goWriting(){
        location.href = '/writing/';
    }
</script>
</body>
</html>
```

게시판 스타일

▪ style.css

```
/* boardlist 스타일 */  
section .tbl_board{width: 700px; margin:0 auto; border-collapse: collapse}  
section .tbl_board, th, td{border: 1px solid #ccc; padding: 10px}  
section .tbl_board th{background-color: #eee;}  
section .tbl_board tr td:nth-child(2){text-align: left;}  
section .tbl_board tr a{text-decoration: none;}  
section .tbl_board tr a:hover{text-decoration: underline;}  
section .tbl_board a{color: #000}  
section .btn_write{height:40px; text-align: right; margin: 15px 160px;}  
section .btn_write input{padding: 5px 15px;}
```


게시판 스타일

▪ style.css

```
/* writing 스타일*/
section .writing{width: 600px; margin: 0 auto;}
section .writing input[type='text']{width: 550px; height: 25px;}
section .writing textarea{width: 550px; height: 300px;}
section .writing input[type='button'], input[type='reset']{width: 80px; height: 30px;}

/* board_view 스타일*/
section .tbl_bo_view{width: 600px; margin: 0 auto;}
section .tbl_bo_view, th, td{border: 1px solid #ccc; border-collapse: collapse;}
section .tbl_bo_view th, td{padding: 10px;}
section .tbl_bo_view textarea{width: 550px; height: 300px;}
section .tbl_bo_view input[type='text']{width: 550px; height: 25px;}
section .tbl_bo_view input[type='submit']{width: 70px; height: 35px; border-radius: 3px;}
section .tbl_bo_view input[type='reset']{width: 70px; height: 35px; border-radius: 3px;}
section .tbl_bo_view a{color: #00f;}
```

게시글 상세 보기

게시글 보기

겨울속으로

기온이 많이 내려가고
해도 빨리 저서 지금은 어두워요...

작성자 : 김하나

[목록](#) [수정](#) [삭제](#)

게시글 상세 보기

- `/board_view/<int:bn>/` 경로 및 `board_view()` 함수

```
@app.route('/board_view/<int:bn>/')
def board_view(bn):
    conn = getconn()
    cur = conn.cursor()
    sql = "SELECT * FROM board WHERE bn = %s " % (bn)
    cur.execute(sql)
    rs = cur.fetchone()
    print(rs)
    conn.close()
    return render_template('board_view.html', rs=rs)
```

게시글 상세 보기

▪ board_view.html

```
<div id="container">
  {% include 'navbar.html' %}
  <section>
    <h2>게시글 보기</h2>
    <table class="tbl_bo_view">
      {% if rs %}
      <tr>
        <td><input type="text" name="title" value="{{ rs[1] }}" readonly></td>
      </tr>
      <tr>
        <td>
          <textarea name="content" readonly>{{ rs[2] }}</textarea>
        </td>
      </tr>
      <tr>
        <td><input type="text" name="mid" value="작성자 : {{ rs[4] }}" readonly></td>
      </tr>
    </table>
  </section>
</div>
```

게시글 상세 보기

▪ board_view.html

```
<tr>
  <td colspan="2">
    <a href="{ { url_for('boardlist') } }">목록</a>
    {% if session['userName'] == rs[4] %}
    <a href="{ { url_for('board_edit', bno=rs[0]) } }">수정</a>
    <a onclick="return confirm('정말로 삭제하시겠습니까?')"
      href="{ { url_for('board_del', bno=rs[0]) } }">삭제</a>
    {% endif %}
  </td>
</tr>
{% endif %}
</table>
</section>
```

게시글 쓰기

게시판 글쓰기

글 제목

글 내용

글쓰기

취소

게시글 쓰기

- `/writing/` 경로 및 `writing()` 함수

```
@app.route('/writing/', methods=['GET', 'POST'])
def writing():
    if request.method == "POST":
        title = request.form['title']
        content = request.form['content']
        mid = session.get('userName') #외래키 mid에 글쓴이 저장

        conn = getconn()
        cur = conn.cursor()
        sql = "INSERT INTO board (title, content, mid) VALUES ('%s', '%s', '%s')" \
            % (title, content, mid)
        cur.execute(sql)
        conn.commit()
        conn.close()
        return redirect(url_for('boardlist'))
    else:
        return render_template('writing.html')
```

게시글 쓰기

■ writing.html

```
<div id="container">
  {% if not session['userID'] %}
    <script>
      alert('로그인이 필요합니다.');
```

location.href = '/login/';

```
    </script>
  {% endif %}
  {% include 'navbar.html' %}
  <section>
    <h1>게시판 글쓰기</h1>
    <form action="{% url_for('writing') %}" method="post">
      <table class="tbl_write">
        <tr>
          <td><input type="text" name="title" placeholder="글 제목"></td>
        </tr>
```


게시글 쓰기

■ writing.html

```
<tr>
  <td><textarea name="content" placeholder="글 내용"></textarea></td>
</tr>
<tr>
  <td colspan="2">
    <input type="submit" value="글쓰기">
    <input type="reset" value="취소">
  </td>
</tr>
</table>
</form>
</section>
```

글쓴이 이름 넣기

- 글쓴이 이름이 아이디가 아닌 이름으로 넣기

- ▶ 로그인 – login()함수에서 이름 세션 발급

```
if rs:
    session['userID'] = rs[0]    # 아이디 세션 발급
    session['userName'] = rs[2]  # 이름 세션 발급
    return redirect(url_for('index'))
else:
    error = "아이디나 비밀번호가 일치하지 않습니다."
    return render_template('login.html', error=error)
```

- ▶ 회원 가입 – register()함수에서 이름 세션 발급

```
if rs:
    session['userID'] = rs[0]    # 세션 발급 - mid
    session['userName'] = rs[2]  # 세션 발급 - name
    return redirect(url_for('memberlist'))
```

글쓴이 이름 넣기

- 글쓴이 이름이 아이디가 아닌 이름으로 넣기

navbar.html - 메뉴에 이름 세션 넣기

```
{% if session %}
    <li><a href="{ { url_for('index') } }">Home</a></li>
    <li><a href="{ { url_for('logout') } }">
        ( { { session['userName'] } } 님 ) 로그아웃</a></li>
    <li><a href="{ { url_for('memberlist') } }">회원목록</a></li>
    <li><a href="{ { url_for('boardlist') } }">게시판</a></li>
{% else %}
    <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('boardlist') } }">게시판</a></li>
{% endif %}
```

글쓴이 이름 넣기

- 글쓴이 이름이 아이디가 아닌 이름으로 넣기

board_view.html에서 이름 세션 권한으로 설정

```
<td colspan="2">
  <a href="{{ url_for('boardlist') }}">목록</a>
  {% if session['userName'] == rs[4] %}
  <a href="{{ url_for('board_edit', bno=rs[0]) }}">수정</a>
  <a onclick="return confirm('정말로 삭제하시겠습니까?')"
    href="{{ url_for('board_del', bno=rs[0]) }}">삭제</a>
  {% endif %}
</td>
```

글쓴이 이름 넣기

- 글쓴이 이름이 아이디가 아닌 이름으로 넣기
board_edit() 함수 세션 입력

```
@app.route('/board_edit/<int:bno>/', methods=['GET', 'POST'])
def board_edit(bno):
    if request.method == "POST":
        # 자료 전달 받음
        title = request.form['title']
        content = request.form['content']
        mid = session.get('userName') #자동으로 세션 입력
```

게시글 삭제

127.0.0.1:5000 내용:
정말로 삭제하시겠습니까?

확인

취소

한 해가 저물어간다.

2021년도 이제 얼마 남지 않았네요 .

게시판

게시글 삭제

- `/board_del/<int:bno>/` 경로 및 `board_del()` 함수

```
@app.route('/board_del/<int:bno>/')
def board_del(bno):
    conn = getconn()
    cur = conn.cursor()
    sql = "DELETE FROM board WHERE bno = %s " % (bno)
    cur.execute(sql)
    conn.commit()
    conn.close()
    return redirect(url_for('boardlist'))
```

게시글 수정

게시글 보기

한 해가 저물어간다.

2021년도 이제 얼마 남지 않았네요.

작성자 : 강하늘

저장

취소

게시글 수정

- `/board_edit/<int:bn>/` 경로 및 `board_edit()` 함수

```
@app.route('/board_edit/<int:bn>/', methods=['GET', 'POST'])
def board_edit(bn):
    if request.method == "POST":
        title = request.form['title']
        content = request.form['content']
        mid = session.get('userName') # 외래키 mid에 글쓴이 저장

        conn = getconn()
        cur = conn.cursor()
        sql = "UPDATE board SET title = '%s', content = '%s', mid = '%s' WHERE bn = %s " \
            % (title, content, mid, bn)
        cur.execute(sql)
        conn.commit()
        conn.close()
        return redirect(url_for('board_view', bn=bn))
```

게시글 수정

■ board_edit.html

```
<h2>게시글 보기</h2>
<form action="{{ url_for('board_edit', bno=rs[0]) }}" method="post">
  <table class="tbl_bo_view">
    {% if rs %}
    <tr>
      <td><input type="text" name="title" value="{{ rs[1] }}"></td>
    </tr>
    <tr>
      <td>
        <textarea name="content">{{ rs[2] }}</textarea>
      </td>
    </tr>
    <tr>
      <td><input type="text" name="mid" value="작성자 : {{ rs[4] }}"></td>
    </tr>
    <tr>
      <td colspan="2">
        <input type="submit" value="저장">
        <input type="reset" value="취소">
      </td>
    </tr>
    {% endif %}
  </table>
```

블로그 폼 유효성 검사

127.0.0.1:5000 내용:
제목을 입력해 주세요

확인

글 제목

글 내용

게시판

글쓰기 폼 유효성 검사

■ writing.html 변경

```
<title>게시글 쓰기</title>
<link rel="stylesheet" href="{ { url_for('static', filename='css/style.css') } }">
<script src="{ { url_for('static', filename='js/checkWrite.js') } }"></script>

<form action="{ { url_for('writing') } }" method="post" name="writeForm">
  <table class="tbl_write">
    <tr>
      <td><input type="text" name="title" placeholder="글 제목"></td>
    </tr>
    <tr>
      <td><textarea name="content" placeholder="글 내용"></textarea></td>
    </tr>
    <tr>
      <td colspan="2">
        <input type="button" value="글쓰기" onclick="checkWrite()">
        <input type="reset" value="취소">
      </td>
    </tr>
  </table>
</form>
```

글쓰기 폼 유효성 검사

■ writing.js 만들기

```
function checkWrite(){  
    var form = document.writeForm;  
    var title = form.title.value;  
    var content = form.content.value;  
  
    if(title == ""){  
        alert("제목을 입력해 주세요");  
        form.title.focus();  
        return false;  
    }  
}
```

```
    else if(content == ""){  
        alert("내용을 입력해 주세요.");  
        form.content.focus();  
        return false;  
    }  
    else{  
        form.submit();  
    }  
}
```

게시글 추천수

- 추천수 넣기

게시판 목록

번호	제목	작성일	추천수	글쓴이
5	2021년 12월	2021-12-15 04:41:11	0	구름
4	겨울속으로	2021-12-13 19:16:48	0	김하나
3	커피향	2021-12-13 19:16:07	2	김하나
2	오미크론	2021-12-13 19:12:52	3	구름
1	제목1	2021-12-13 18:14:20	5	today

글쓰기

게시글 추천수

- 추천수 칼럼(필드) 추가 – hit INTEGER

```
sql = ""  
CREATE TABLE board(  
    bno INTEGER PRIMARY KEY AUTOINCREMENT,  
    title TEXT NOT NULL,  
    content TEXT NOT NULL,  
    createDate TIMESTAMP DATE DEFAULT (datetime('now', 'localtime')),  
    hit INTEGER,  
    mid CHAR(5) NOT NULL,  
    FOREIGN KEY(mid) REFERENCES member(mid) ON DELETE CASCADE  
);  
""
```

게시글 추천수

■ 게시물 쓰기 – writing() 함수에서 hit 삽입

```
@app.route('/writing/', methods=['GET', 'POST'])
def writing():
    if request.method == "POST":
        title = request.form['title']
        content = request.form['content']
        mid = session.get('userName') #외래키 mid에 글쓴이 저장
        hit = 0

        conn = getconn()
        cur = conn.cursor()
        sql = "INSERT INTO board (title, content, hit, mid) VALUES ('%s', '%s', %s, '%s')" \
            % (title, content, hit, mid)
        cur.execute(sql)
        conn.commit()
        conn.close()
        return redirect(url_for('boardlist'))
    else:
        return render_template('writing.html')
```


게시글 추천수

- 게시글 보기 – `board_view()` 함수에서 추천수 1증가

```
@app.route('/board_view/<int:bn>/')
def board_view(bno):
    conn = getconn()
    cur = conn.cursor()
    sql = "SELECT * FROM board WHERE bno = %s " % (bno)
    cur.execute(sql)
    rs = cur.fetchone()
    # 추천수 1증가후 업데이트
    hit = rs[4]
    hit = hit + 1
    sql = "UPDATE board SET hit = %s WHERE bno = %s" % (hit, bno)
    cur.execute(sql)
    conn.commit()
    conn.close()
    return render_template('board_view.html', rs=rs)
```

게시글 추천수

■ 게시물 목록 – boardlist()에서 추천수 표시하기

```
<h1>게시판 목록</h1>
<table class="tbl_board">
  <tr>
    <th>번호</th><th>제목</th><th>작성일</th><th>추천수</th><th>글쓴이</th>
  </tr>
  {% for i in rs %}
  <tr>
    <td>{{ i[0] }}</td>
    <td><a href="{ url_for('board_view', bno=i[0]) }}">{{ i[1] }}</a></td>
    <td>{{ i[3] }}</td>
    <td>{{ i[4] }}</td>
    <td>{{ i[5] }}</td>
  </tr>
```