



[스파르타코딩클럽] 웹개발 플러스 - 4주차



매 주차 강의자료 시작에 PDF파일을 올려두었어요!

▼ PDF 파일

▼ 단축키 모음

▼ 새로고침

- `F5`

▼ 저장

- Windows: `Ctrl` + `S`
- macOS: `command` + `S`

▼ 전체선택

- Windows: `Ctrl` + `A`
- macOS: `command` + `A`

▼ 잘라내기

- Windows: `Ctrl` + `X`
- macOS: `command` + `X`

▼ 콘솔창 켜기

- `shift` + `enter`

▼ 코드정렬

- Windows: `Ctrl` + `Alt` + `L`
- macOS: `option` + `command` + `L`

▼ 들여쓰기

- `Tab`
- 들여쓰기 취소: `Shift` + `Tab`

▼ 주석

- Windows: `Ctrl` + `/`
- macOS: `command` + `/`

[수업 목표]

1. Bulma로 예쁜 웹사이트를 만든다.
2. 로그인 기능을 구현할 수 있다.

[목차]

- [01. 4주차 이번주에 배울 것](#)
- [02. Bulma로 웹사이트 꾸미기](#)
- [03. 회원가입 기능](#)
- [04. 로그인 기능](#)
- [05. 프로젝트 4: Sweeter](#)
- [06. 프로젝트 세팅](#)
- [07. 로그인 & 회원가입 페이지 모습 만들기](#)

08. 회원가입 페이지 기능 만들기
09. 로그인 페이지 기능 만들기
10. 메인 페이지 모습 만들기
11. 메인 페이지 기능 만들기 - 포스팅
12. 메인 페이지 기능 만들기 - 좋아요
13. 프로필 페이지 모습 만들기 - 전체
14. 프로필 페이지 모습 만들기 - 프로필 수정
15. 프로필 페이지 기능 만들기
16. 4주차 끝 & 숙제 설명
17. 4주차 숙제 답안 코드



모든 토글을 열고 닫는 단축키

Windows : **Ctrl** + **alt** + **t**

Mac : **⌘** + **⌥** + **t**

01. 4주차 이번주에 배울 것

▼ 보안, 로그인, 암호화

- 웹플러스 수업에서 일기장, 단어장, 맛집지도 등 혼자만을 위한 서비스들을 만들어왔지만, 사실 주소만 알면 누구나 들어와서 내용을 보고 추가/수정/삭제가 가능했죠?
- 이를 방지하기 위해 비밀번호를 입력해야만 내용을 볼 수 있도록 만들 수도 있을 것입니다.
- 하지만 만약 여러 사람이 한 서비스를 이용해야한다면? 전체에게 공개된 내용과 개인이 볼 수 있는 내용이 다르다면? 이럴 때는 로그인 기능이 필요하겠죠.

▼ 이번주에 만들 웹서비스 구경하기: 스위트

▼ [코드스니펫] - Sweeter 보러 가기

<http://spartacodingclub.shop/wp/sweeter>

02. Bulma로 웹사이트 꾸미기

▼ 1) Bulma란?



우리가 써왔던 Bootstrap과 비슷하게, 미리 정해진 모습의 클래스를 가져다 쓸 수 있는 무료 CSS 프레임워크입니다.

• Bootstrap과 비교 ([공식문서 참조](#))

- Bootstrap은 jQuery를 써서 웹사이트에서의 상호작용을 쉽게 구현할 수 있는 반면, Bulma는 순수한 CSS 프레임워크이기 때문에 기능을 직접 구현해야하지만 대신 더 자유롭게 커스터마이징이 가능해요.
- Bootstrap은 커뮤니티가 커서 테마나 플러그인 등이 개발이 많이 되어있고 질문에 대한 답이나 예시 등을 찾기 쉽고, Bulma는 문법이 직관적이고 Flexbox 등 최신기술을 많이 써요!

▼ 2) Bulma 공식문서 살펴보기

- Bootstrap과 마찬가지로 Bulma도 [공식문서](#)에 각 컴포넌트의 묘사와 예시가 잘 정리되어 있어요.

▼ [코드스니펫] - Bulma 공식문서 링크

<https://bulma.io/documentation/>

▼ 3) Bulma 써보기

▼ 코딩을 시작하려면 우선 새 파이썬 프로젝트부터 만들어야겠죠?

- File > New Project...에 가서 project04 폴더 열기
- Project Interpreter에서 가상환경에 필요한 패키지 설치하기(flask , pyjwt , pymongo)
- Template Language를 Jinja2로 설정하기
- project04 폴더 안에 templates, static 폴더 만들기
- templates 폴더 안에 prac_bulma.html 파일 만들기

▼ [코드스니펫] - prac_bulma.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <title>Bulma 쓰는 법 연습하기</title>

    <!-- Bulma CSS 링크 -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">

  </head>
  <body>

  </body>
</html>
```

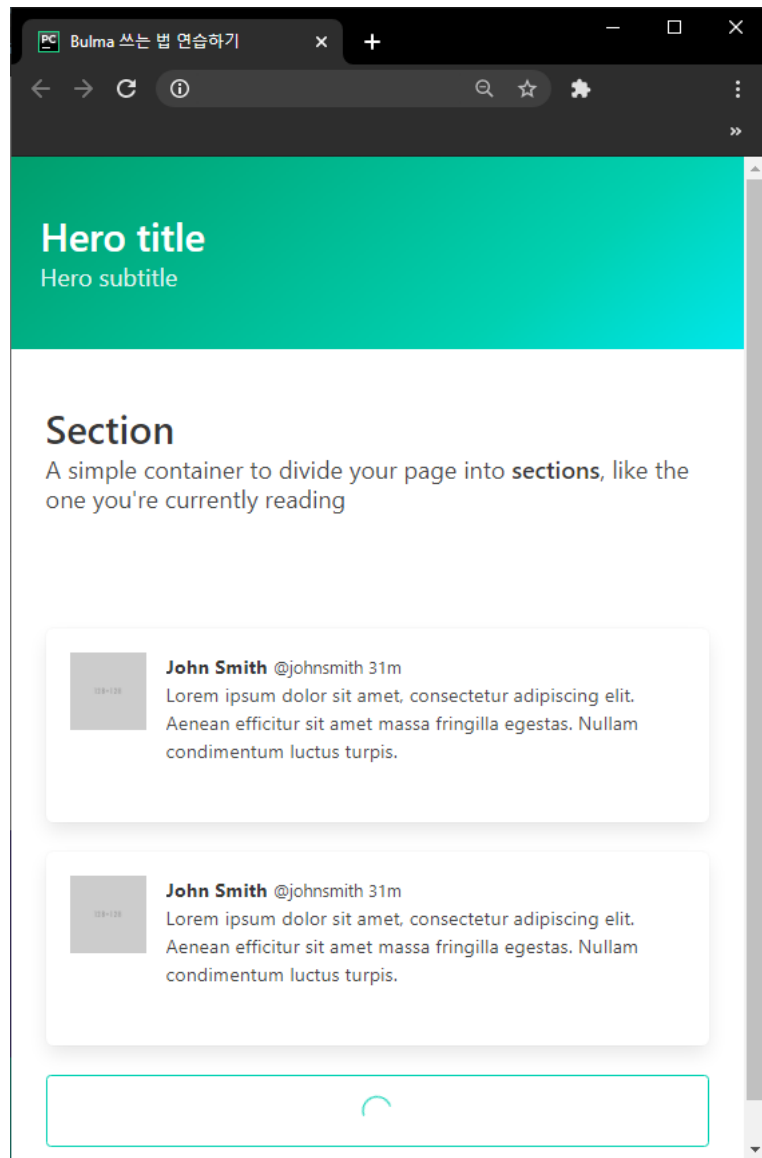
▼ bulma를 쓰기 위해서는 딱 한 줄만 추가해주면 됩니다! prac_bulma.html 파일에 이미 포함되어 있어요.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
```

▼ 직접 해보기



공식 문서에서 각 요소들을 복사해 아래와 같은 화면을 만들어볼까요?



- 잠깐 Bootstrap과 비교를 해볼까요? Bulma는 직관적으로 태그명과 같은 클래스명을 쓰고 각 기능을 모듈화하여 적용하고 싶은 CSS를 하나 씩 엮는 방식으로 추구합니다.

- Bootstrap

```
<button class="btn btn-outline-primary btn-lg">Primary</button>
```

- Bulma

```
<button class="button is-primary is-outlined is-large">Primary</button>
```

▼ Hero 배너

`hero`는 화면 전체 너비를 채우는 배너 클래스입니다. 자식인 `hero-body` 안에 내용을 넣어줄 수 있어요. 클래스를 중첩해서 색과 그라디언트, 높이 등을 바꿔줄 수 있습니다.

▼ [코드스니펫] - hero

```
<section class="hero is-primary is-bold is-medium">
  <div class="hero-body">
    <div class="container">
      <h1 class="title">
        Hero title
```

```

        </h1>
        <h2 class="subtitle">
            Hero subtitle
        </h2>
    </div>
</div>
</section>

```

▼ Section

body 태그 안에 바로 넣어서 구역을 나누어줄 때 씁니다.

▼ [코드스니펫] - section

```

<section class="section">
  <div class="container">
    <h1 class="title">Section</h1>
    <h2 class="subtitle">
      A simple container to divide your page into <strong>sections</strong>, like the one you're
      currently reading
    </h2>
  </div>
</section>

```

- Bulma의 클래스에 CSS를 추가해줄 수도 있습니다.

▼ [코드스니펫] - Bulma CSS 추가하기

```

.section {
  width: 600px;
  max-width: 100vw;
  margin: auto;
}

```

▼ Box와 Media

box 는 테두리와 그림자 등을 이용하여 만든 카드이고, **media** 는 스위터에서 많이 쓰이는 이미지와 글의 조합입니다.

▼ [코드스니펫] - box

```

<div class="box">
  <article class="media">
    <div class="media-left">
      <figure class="image is-64x64">
        
      </figure>
    </div>
    <div class="media-content">
      <div class="content">
        <p>
          <strong>John Smith</strong> <small>@johnsmith</small> <small>31m</small>
          <br>
          Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean efficitur sit
          amet massa fringilla egestas. Nullam condimentum luctus turpis.
        </p>
      </div>
      <nav class="level is-mobile">
        <div class="level-left">
          <a class="level-item" aria-label="reply">
            <span class="icon is-small">
              <i class="fas fa-reply" aria-hidden="true"></i>
            </span>
          </a>
          <a class="level-item" aria-label="retweet">
            <span class="icon is-small">
              <i class="fas fa-retweet" aria-hidden="true"></i>
            </span>
          </a>
          <a class="level-item" aria-label="like">
            <span class="icon is-small">
              <i class="fas fa-heart" aria-hidden="true"></i>
            </span>
          </a>
        </div>
      </nav>
    </div>
  </article>
</div>

```

▼ Button

▼ [코드스니펫] - button

```
<button class="button is-primary is-outlined is-large is-fullwidth is-loading"></button>
```

03. 회원가입 기능

▼ 4) 해시함수란?



해시함수란, 알고리즘의 한 종류로서 임의의 데이터를 입력 받아 항상 고정된 길이의 임의의 값으로 변환해주는 함수를 말합니다.

- 우리가 회원가입에 사용할 해시함수 SHA256은 어떤 길이의 입력값을 넣어도 항상 256바이트의 결과값이 나옵니다!
- 추가적으로 동일한 입력값은 항상 같은 결과값이 나오고, 입력값은 조금이라도 달라지면 완전히 다른 값이 나오게 됩니다. 그리고 결과값을 통해 입력값을 알아내는 것이 불가능하다는 세 가지 특징이 있습니다!

▼ 5) 플라스크 서버에서 회원가입 기능 구현하기

- 로그인 기능을 구현하기 위해서는 우선 로그인 페이지, 회원가입 페이지, 그리고 로그인 이후 볼 수 있는 페이지 등 최소 세 페이지가 필요하겠죠?

▼ 파일 준비하기

▼ [코드스니펫] - app.py

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

app = Flask(__name__)

from pymongo import MongoClient

client = MongoClient('mongodb://3.34.44.93', 27017, username="sparta", password="woowa")
db = client.dbsparta_plus_week4

# JWT 토큰을 만들 때 필요한 비밀문자열입니다. 아무거나 입력해도 괜찮습니다.
# 이 문자열은 서버만 알고있기 때문에, 내 서버에서만 토큰을 인코딩(=만들기)/디코딩(=풀기) 할 수 있습니다.
SECRET_KEY = 'SPARTA'

# JWT 패키지를 사용합니다. (설치해야할 패키지 이름: PyJWT)
import jwt

# 토큰에 만료시간을 줘야하기 때문에, datetime 모듈도 사용합니다.
import datetime

# 회원가입 시엔, 비밀번호를 암호화하여 DB에 저장해두는 게 좋습니다.
# 그렇지 않으면, 개발자(=나)가 회원들의 비밀번호를 볼 수 있으니까요.^^;
import hashlib

#####
## HTML을 주는 부분 ##
#####
@app.route('/')
def home():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        user_info = db.user.find_one({"id": payload['id']})
        return render_template('index.html', nickname=user_info["nick"])
    except jwt.ExpiredSignatureError:
        return redirect(url_for("login", msg="로그인 시간이 만료되었습니다."))
    except jwt.exceptions.DecodeError:
        return redirect(url_for("login", msg="로그인 정보가 존재하지 않습니다."))

@app.route('/login')
def login():
    msg = request.args.get("msg")
```

```

        return render_template('login.html', msg=msg)

@app.route('/register')
def register():
    return render_template('register.html')

#####
##   로그인을 위한 API           ##
#####

# [회원가입 API]
# id, pw, nickname을 받아서, mongoDB에 저장합니다.
# 저장하기 전에, pw를 sha256 방법(=단방향 암호화. 풀어볼 수 없음)으로 암호화해서 저장합니다.
@app.route('/api/register', methods=['POST'])
def api_register():
    id_receive = request.form['id_give']
    pw_receive = request.form['pw_give']
    nickname_receive = request.form['nickname_give']

    pw_hash = hashlib.sha256(pw_receive.encode('utf-8')).hexdigest()

    db.user.insert_one({'id': id_receive, 'pw': pw_hash, 'nick': nickname_receive})

    return jsonify({'result': 'success'})

# [로그인 API]
# id, pw를 받아서 맞춰보고, 토큰을 만들어 발급합니다.
@app.route('/api/login', methods=['POST'])
def api_login():
    id_receive = request.form['id_give']
    pw_receive = request.form['pw_give']

    # 회원가입 때와 같은 방법으로 pw를 암호화합니다.
    pw_hash = hashlib.sha256(pw_receive.encode('utf-8')).hexdigest()

    # id, 암호화된pw를 가지고 해당 유저를 찾습니다.
    result = db.user.find_one({'id': id_receive, 'pw': pw_hash})

    # 찾으면 JWT 토큰을 만들어 발급합니다.
    if result is not None:
        # JWT 토큰에는, payload와 시크릿키가 필요합니다.
        # 시크릿키가 있어야 토큰을 디코딩(=풀기) 해서 payload 값을 볼 수 있습니다.
        # 아래에선 id와 exp를 담았습니다. 즉, JWT 토큰을 풀면 유저ID 값을 알 수 있습니다.
        # exp에는 만료시간을 넣어줍니다. 만료시간이 지나면, 시크릿키로 토큰을 풀 때 만료되었다고 에러가 납니다.
        payload = {
            'id': id_receive,
            'exp': datetime.datetime.utcnow() + datetime.timedelta(seconds=5)
        }
        token = jwt.encode(payload, SECRET_KEY, algorithm='HS256').decode('utf-8')

        # token을 줍니다.
        return jsonify({'result': 'success', 'token': token})
    # 찾지 못하면
    else:
        return jsonify({'result': 'fail', 'msg': '아이디/비밀번호가 일치하지 않습니다.'})

# [유저 정보 확인 API]
# 로그인된 유저만 call 할 수 있는 API입니다.
# 유효한 토큰을 줬아 올바른 결과를 얻어갈 수 있습니다.
# (그렇지 않으면 남의 장바구니라든가, 정보를 누구나 볼 수 있겠죠?)
@app.route('/api/nick', methods=['GET'])
def api_valid():
    token_receive = request.cookies.get('mytoken')

    # try / catch 문?
    # try 아래를 실행했다가, 에러가 있으면 except 구문으로 가란 얘기입니다.

    try:
        # token을 시크릿키로 디코딩합니다.
        # 보실 수 있도록 payload를 print 해두었습니다. 우리가 로그인 시 넣은 그 payload와 같은 것이 나옵니다.
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        print(payload)

        # payload 안에 id가 들어있습니다. 이 id로 유저정보를 찾습니다.
        # 여기에선 그 예로 닉네임을 보내주겠습니다.
        userinfo = db.user.find_one({'id': payload['id']}, {'_id': 0})
        return jsonify({'result': 'success', 'nickname': userinfo['nick']})
    except jwt.ExpiredSignatureError:
        # 위를 실행했는데 만료시간이 지났으면 에러가 납니다.
        return jsonify({'result': 'fail', 'msg': '로그인 시간이 만료되었습니다.'})
    except jwt.exceptions.DecodeError:
        return jsonify({'result': 'fail', 'msg': '로그인 정보가 존재하지 않습니다.'})

```

```
if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)
```

▼ [코드스니펫] - index.html

```
<!doctype html>
<html lang="en">
  <head>

    <!-- Webpage Title -->
    <title>Hello, world!</title>

    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bulma CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.js"></script>

    <script>
      // 로그아웃은 내가 가지고 있는 토큰만 쿠키에서 없애면 됩니다.
      function logout(){
        $.removeCookie('mytoken');
        alert('로그아웃!')
        window.location.href='/login'
      }
    </script>

  </head>
  <body>
    <p>
      <h1 class="title">로그인하고 5초 동안만 볼 수 있는 페이지입니다.</h1>
      <h1 class="subtitle">계속 새로그침 해보세요</h1>
    </p>
    <h5 class="subtitle">나의 닉네임은: {{nickname}}</h5>
    <button class="button is-danger" onclick="logout()">로그아웃하기</button>
  </body>
</html>
```

▼ [코드스니펫] - login.html

```
<!doctype html>
<html lang="en">
  <head>

    <!-- Webpage Title -->
    <title>Hello, world!</title>

    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bulma CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.js"></script>

    <script>
      {% if msg %}
        alert("{ { msg } }")
      {% endif %}
      // ['쿠키'라는 개념에 대해 알아봅시다]
      // 로그인을 구현하면, 반드시 쿠키라는 개념을 사용합니다.
      // 페이지에 관계없이 브라우저에 임시로 저장되는 정보입니다. 키:밸류 형태(딕셔너리 형태)로 저장됩니다.
      // 쿠키가 있기 때문에, 한번 로그인하면 네이버에서 다시 로그인할 필요가 없는 것입니다.
      // 브라우저를 닫으면 자동 삭제되게 하거나, 일정 시간이 지나면 삭제되게 할 수 있습니다.
      function login() {
        $.ajax({
          type: "POST",
          url: "/api/login",
          data: {id_give: $('#userid').val(), pw_give: $('#userpw').val()},
```



```

        success: function (response) {
            if (response['result'] == 'success') {
                // 로그인 성공으로 되면, 토큰을 받아옵니다.
                // 이 토큰을 mytoken이라는 키 값으로 쿠키에 저장합니다.
                $.cookie('mytoken', response['token']);

                alert('로그인 완료!')
                window.location.href = '/'
            } else {
                // 로그인 안되면 에러메시지를 띄웁니다.
                alert(response['msg'])
            }
        }
    })
}

</script>

</head>
<body>
    <div class="section has-text-centered">
        <h1 class="title">로그인 페이지</h1>
        <div class="container" style="width:60%">
            <div class="field is-horizontal">
                <div class="field-label is-normal">
                    <label class="label" for="userid">ID</label>
                </div>
                <div class="field-body">
                    <div class="field">
                        <div class="control">
                            <input type="text" class="input" id="userid" aria-describedby="emailHelp"
                                placeholder="My ID">
                        </div>
                    </div>
                </div>
            </div>
            <div class="field is-horizontal">
                <div class="field-label is-normal">
                    <label class="label" for="userpw">PW</label>
                </div>
                <div class="field-body">
                    <div class="field">
                        <div class="control">
                            <input type="password" class="input" id="userpw" placeholder="My Password">
                        </div>
                    </div>
                </div>
            </div>
            <button class="button is-primary" onclick="login()">로그인</button>
        </div>
    </div>
</body>
</html>

```

▼ [코드스니펫] - register.html

```

<!doctype html>
<html lang="en">
    <head>

        <!-- Webpage Title -->
        <title>Hello, world!</title>

        <!-- Required meta tags -->
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

        <!-- Bulma CSS -->
        <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
        <!-- JS -->
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
        <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.js"></script>

        <script>
            // 간단한 회원가입 함수입니다.
            // 아이디, 비밀번호, 닉네임을 받아 DB에 저장합니다.
            function register() {
                $.ajax({
                    type: "POST",
                    url: "/api/register",
                    data: {
                        id_give: $('#userid').val(),
                        pw_give: $('#userpw').val(),
                        nickname_give: $('#usernick').val()
                    }
                })
            }
        </script>
    </head>
    <body>
        <div class="section has-text-centered">
            <h1 class="title">회원가입</h1>
            <div class="container" style="width:60%">
                <div class="field is-horizontal">
                    <div class="field-label is-normal">
                        <label class="label" for="userid">아이디</label>
                    </div>
                    <div class="field-body">
                        <div class="field">
                            <div class="control">
                                <input type="text" class="input" id="userid" placeholder="아이디">
                            </div>
                        </div>
                    </div>
                </div>
                <div class="field is-horizontal">
                    <div class="field-label is-normal">
                        <label class="label" for="userpw">비밀번호</label>
                    </div>
                    <div class="field-body">
                        <div class="field">
                            <div class="control">
                                <input type="password" class="input" id="userpw" placeholder="비밀번호">
                            </div>
                        </div>
                    </div>
                </div>
                <div class="field is-horizontal">
                    <div class="field-label is-normal">
                        <label class="label" for="usernick">닉네임</label>
                    </div>
                    <div class="field-body">
                        <div class="field">
                            <div class="control">
                                <input type="text" class="input" id="usernick" placeholder="닉네임">
                            </div>
                        </div>
                    </div>
                </div>
                <button class="button is-primary" onclick="register()">회원가입</button>
            </div>
        </div>
    </body>
</html>

```

```

    },
    success: function (response) {
        if (response['result'] == 'success') {
            alert('회원가입이 완료되었습니다.')
            window.location.href = '/login'
        } else {
            alert(response['msg'])
        }
    }
})
}

</script>
</head>
<body>
    <div class="section has-text-centered">
        <h1 class="title">회원가입 페이지</h1>
        <div class="container" style="width:60%">
            <div class="field is-horizontal">
                <div class="field-label is-normal">
                    <label class="label" for="userid">ID</label>
                </div>
                <div class="field-body">
                    <div class="field">
                        <div class="control">
                            <input type="text" class="input" id="userid" aria-describedby="emailHelp"
                                placeholder="My ID">
                        </div>
                    </div>
                </div>
            </div>
            <div class="field is-horizontal">
                <div class="field-label is-normal">
                    <label class="label" for="userpw">PW</label>
                </div>
                <div class="field-body">
                    <div class="field">
                        <div class="control">
                            <input type="password" class="input" id="userpw" placeholder="My Password">
                        </div>
                    </div>
                </div>
            </div>
            <div class="field is-horizontal">
                <div class="field-label is-normal">
                    <label class="label" for="usernick">NICKNAME</label>
                </div>
                <div class="field-body">
                    <div class="field">
                        <div class="control">
                            <input type="text" class="input" id="usernick" placeholder="My Nickname">
                        </div>
                    </div>
                </div>
            </div>
            <button class="button is-primary" onclick="register()">회원가입</button>
        </div>
    </div>
</body>
</html>

```

- 회원가입 시, 입력 정보를 DB에 저장합니다. 비밀번호는 암호화하여 저장합니다.
- 회원가입이 제대로 완료되었다면, Robo3T로 확인해 봅시다! (입력된 패스워드가 암호화 되었죠?)

▼ (1) ObjectId("5fbf881142d05dd647721790")	{ 4 fields }	Object
_id	ObjectId("5fbf881142d05dd647721790")	ObjectId
id	ernybugi	String
pw	594958850b56fe4fbde4c60fbd7f034b5e328509...	String
nick	어니부기	String

04. 로그인 기능

▼ 6) JWT란?



JSON Web Token의 줄임말로, JSON 객체를 사용해 정보를 안정성 있게 전달하는 웹표준이에요!

- 예를 들어, 로그인 기능을 생각해 보면 사용자가 로그인하면 서버에서 회원임을 인증하는 토큰을 넘겨줌으로써 이후 회원만 접근할 수 있는 서비스 영역에서 신분을 확인하는 데 쓰일 수 있습니다.

- [더 깊게 알아보기](#)

▼ 7) 플라스크 서버에서 로그인 기능 구현하기

- 로그인 시, 비밀번호를 같은 방법으로 암호화한 후, DB에서 해당 아이디와 비밀번호를 갖는 회원이 있는지 찾습니다. 회원 정보가 없는 경우 실패 메시지를 보내고, 찾은 경우 아이디와 토큰 만료 시간을 저장하는 토큰을 만들어 넘겨줍니다.
- 로그인 성공 메시지를 받으면 건네받은 토큰을 쿠키로 저장하여 만료되기 전까지 갖고 있으면서, API 요청을 보낼 때마다 회원임을 확인받습니다.
- 로그아웃 시 해당 토큰을 삭제합니다.

05. 프로젝트 4: Sweeter

▼ 8) 문제 분석 - 완성작부터 보기: [스위터](#)

▼ [코드스니펫] - Sweeter 보러 가기

```
http://spartacodingclub.shop/wp/sweeter
```

▼ 9) API 설계하기



Sweeter의 각 페이지에서는 많은 일들이 일어납니다! 한 번 정리해볼까요?

▼ 로그인 & 회원가입 페이지

- 기본 화면으로 로그인 화면 보이기
- '회원가입하기' 버튼을 클릭하면 회원가입 화면으로 바뀌기
- '취소' 버튼을 클릭하면 로그인 화면으로 돌아오기
- 회원가입
 - 아이디 & 비밀번호 형식 확인
 - 아이디 중복 확인
 - DB에 아이디와 비밀번호 저장하여 회원가입 & 로그인 화면으로 전환
- 로그인
 - 아이디 & 비밀번호 입력 확인
 - 서버로 POST 요청을 보내 가입 정보가 존재하는지 확인
 - 회원일 경우 토큰 부여

▼ 메인 페이지

- 모든 사람의 포스트를 시간 역순으로 보여주기
- 각 포스트에 좋아요/좋아요 취소 가능
 - 좋아요 누른 포스트는 찬 하트로 보여주기
- 포스팅 칸에 내 프로필 사진 보여주기
 - 프로필 사진 누르면 프로필 페이지로 이동

4. 포스팅 칸 클릭하면 포스팅 모달 띄우기

- 포스팅하기 버튼 클릭하면 포스트 DB에 저장
- 새로고침하여 포스트 목록 다시 띄우기

▼ 프로필 페이지

1. 해당 사용자의 포스트만 시간 역순으로 보여주기
2. 내 프로필이라면 프로필 수정 & 로그아웃 버튼 보여주기
3. 내 프로필일 때만 포스팅 칸 보여주기
4. 프로필 수정 버튼 클릭하면 프로필 수정 모달 보여주기
 - 기존의 저장되어 있는 값 보여주기
 - 수정 시 DB에 업데이트하고 새로고침해서 변경사항 적용
5. 로그아웃 버튼 클릭하면 토큰 삭제하고 로그인 페이지로 이동

06. 프로젝트 세팅

▼ 10) 프로젝트 준비 - app.py 준비하기

▼ [코드스니펫] - app.py

```
from pymongo import MongoClient
import jwt
import datetime
import hashlib
from flask import Flask, render_template, jsonify, request, redirect, url_for
from werkzeug.utils import secure_filename
from datetime import datetime, timedelta

app = Flask(__name__)
app.config["TEMPLATES_AUTO_RELOAD"] = True
app.config['UPLOAD_FOLDER'] = './static/profile_pics'

SECRET_KEY = 'SPARTA'

client = MongoClient('내AWS아이피', 27017, username="아이디", password="비밀번호")
db = client.dbsparta_plus_week4

@app.route('/')
def home():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])

        return render_template('index.html')
    except jwt.ExpiredSignatureError:
        return redirect(url_for("login", msg="로그인 시간이 만료되었습니다."))
    except jwt.exceptions.DecodeError:
        return redirect(url_for("login", msg="로그인 정보가 존재하지 않습니다."))

@app.route('/login')
def login():
    msg = request.args.get("msg")
    return render_template('login.html', msg=msg)

@app.route('/user/<username>')
def user(username):
    # 각 사용자의 프로필과 글을 모아볼 수 있는 공간
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        status = (username == payload["id"]) # 내 프로필이면 True, 다른 사람 프로필 페이지면 False

        user_info = db.users.find_one({"username": username}, {"_id": False})
        return render_template('user.html', user_info=user_info, status=status)
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))
```

```

@app.route('/sign_in', methods=['POST'])
def sign_in():
    # 로그인
    return jsonify({'result': 'success'})

@app.route('/sign_up/save', methods=['POST'])
def sign_up():
    # 회원가입
    username_receive = request.form['username_give']
    password_receive = request.form['password_give']
    password_hash = hashlib.sha256(password_receive.encode('utf-8')).hexdigest()
    # DB에 저장
    return jsonify({'result': 'success'})

@app.route('/sign_up/check_dup', methods=['POST'])
def check_dup():
    # ID 중복확인
    return jsonify({'result': 'success'})

@app.route('/update_profile', methods=['POST'])
def save_img():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        # 프로필 업데이트
        return jsonify({"result": "success", 'msg': '프로필을 업데이트했습니다.'})
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

@app.route('/posting', methods=['POST'])
def posting():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        # 포스팅하기
        return jsonify({"result": "success", 'msg': '포스팅 성공'})
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

@app.route("/get_posts", methods=['GET'])
def get_posts():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        # 포스팅 목록 받아오기
        return jsonify({"result": "success", "msg": "포스팅을 가져왔습니다."})
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

@app.route('/update_like', methods=['POST'])
def update_like():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        # 좋아요 수 변경
        return jsonify({"result": "success", 'msg': 'updated'})
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)

```

▼ 10) 프로젝트 준비 - index.html, login.html, user.html 준비하기

▼ [코드스니펫] - index.html

```

<!doctype html>
<html lang="en">
  <head>

    <!-- Webpage Title -->
    <title>Home | SWEETER</title>

    <!-- Required meta tags -->
    <meta charset="utf-8">

```

```

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<!-- Bulma CSS -->
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
<!-- Font Awesome CSS -->
<link href="//maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

<!-- JS -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.js"></script>

<script>
</script>

</head>
<body>
</body>
</html>

```

▼ [코드스니펫] - login.html

```

<!doctype html>
<html lang="en">
<head>

  <!-- Webpage Title -->
  <title>Log In | SWEETER</title>

  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bulma CSS -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
  <!-- Font Awesome CSS -->
  <link href="//maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

  <!-- JS -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.js"></script>

  <script>
    // {% if msg %}
    //   alert("{ { msg } }")
    // {% endif %}
    function sign_in() {
      $.ajax({
        type: "POST",
        url: "/sign_in",
        data: {
          username_give: "",
          password_give: ""
        },
        success: function (response) {
          if (response['result'] == 'success') {
            $.cookie('mytoken', response['token'], {path: '/'});

            alert('로그인 완료!')
            window.location.href = '/'
          } else {
            // 로그인에 안되면 에러메시지를 띄웁니다.
            alert(response['msg'])
          }
        }
      })
    }

    function sign_up() {
      let username = ""
      let password = ""
      let password2 = ""

      $.ajax({
        type: "POST",
        url: "/sign_up/save",
        data: {
          username_give: username,
          password_give: password
        },
        success: function (response) {
          alert("회원가입을 축하드립니다!")
          window.location.replace("/login")
        }
      })
    }
  </script>

```

```

    });

    }

</script>

</head>
<body>

</body>
</html>

```

▼ [코드스니펫] - user.html

```

<!doctype html>
<html lang="en">
  <head>

    <!-- Webpage Title -->
    <title>{{ user_info.name }} | SWEETER</title>

    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bulma CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
    <!-- Font Awesome CSS -->
    <link href="//maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.js"></script>

    <script>
      function sign_out() {
        $.removeCookie('mytoken', {path: '/'});
        alert('로그아웃!')
        window.location.href = "/login"
      }
    </script>

  </head>
  <body>
    {% if status %}
      내 프로필 페이지
    {% else %}
      다른 사람 프로필 페이지
    {% endif %}
  </body>
</html>

```

▼ 11) 프로젝트 준비 - 로고 이미지, 프로필 기본 이미지 준비하기

- 로고는 static 폴더 안에, 프로필 이미지는 static 폴더 안에 profile_pics 폴더를 만들어서 저장해주세요. 사용자가 업로드할 프로필 사진들도 이 폴더 안에 넣어줄 거예요!

▼ [코드스니펫] - logo.png

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/87fdadda-45d1-4299-884f-811e5773c840/logo.png>

▼ [코드스니펫] - profile_placeholder.png

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/bf5ec7c8-2844-4037-aafc-e0633cf85159/profile_placeholder.png

07. 로그인 & 회원가입 페이지 모습 만들기

▼ 12) 모습 만들기



다른 페이지들은 로그인을 해야만 볼 수 있기 때문에 우선 로그인 페이지부터 구현을 해봅시다!

▼ 배너 만들기

- Bulma의 hero 클래스를 이용하여 배너를 만들어볼까요?

▼ [코드스니펫] - 배너 HTML

```
<section class="hero is-white">
  <div class="hero-body has-text-centered" style="padding-bottom:1rem;margin:auto;">
    <h1 class="title is-sparta">SWEETER</h1>
    <h3 class="subtitle is-sparta">세상을 달달하게</h3>
  </div>
</section>
```

▼ [코드스니펫] - 배너 CSS

```
body {
  background-color: RGBA(232, 52, 78, 0.2);
  min-height: 100vh;
}

.section {
  padding: 1rem 1.5rem;
  max-width: 750px;
  margin: auto;
}

.title {
  font-weight: 800;
  font-size: 5rem;
}

.subtitle {
  font-size: 2rem;
}

.is-sparta {
  color: #e8344e !important;
}
```

- 구글 웹폰트도 적용해서 예쁘게 꾸며봅시다!

▼ [코드스니펫] 구글 웹폰트 링크

```
https://fonts.google.com/?subset=korean
```

▼ 로그인 박스 만들기

▼ [코드스니펫] - 로그인 박스 HTML

```
<section class="section">
  <div class="container">
    <div class="box" style="max-width: 480px;margin:auto">
      <article class="media">
        <div class="media-content">
          <div class="content">
            <div class="field has-addons">
              <div class="control has-icons-left" style="width:100%">
                <input id="input-username" class="input" type="text" placeholder="아이디">
                <span class="icon is-small is-left"><i class="fa fa-user"></i></span>
              </div>
              <div id="btn-check-dup" class="control">
                <button class="button is-sparta" onclick="check_dup()">중복확인</button>
              </div>
            </div>
          </div>
          <p id="help-id" class="help">아이디는 2-10자의 영문과 숫자와 일부 특수문자(. _ -)만 입력 가능합니다.</p>
```



```

        <div class="field">
            <div class="control has-icons-left">
                <input id="input-password" class="input" type="password" placeholder="비밀번호">
                <span class="icon is-small is-left"><i class="fa fa-lock"></i></span>
            </div>
            <p id="help-password" class="help">영문과 숫자 조합의 8-20자의 비밀번호를 설정해주세요. 특수문자(!@#%&^&*)
        </div>

    </div>
    <div id="div-sign-in-or-up" class="has-text-centered">
        <nav class="level is-mobile">
            <button class="level-item button is-sparta" onclick="sign_in()">
                로그인
            </button>

        </nav>
        <hr>
        <h4 class="mb-3">아직 회원이 아니라면</h4>
        <nav class="level is-mobile">

            <button class="level-item button is-sparta is-outlined"
                onclick="toggle_sign_up()">
                회원가입하기
            </button>
        </nav>
    </div>

    <div id="sign-up-box">
        <div class="mb-5">
            <div class="field">
                <div class="control has-icons-left" style="width:100%">
                    <input id="input-password2" class="input" type="password"
                        placeholder="비밀번호 재입력">
                    <span class="icon is-small is-left"><i class="fa fa-lock"></i></span>
                </div>
                <p id="help-password2" class="help">비밀번호를 다시 한 번 입력해주세요.</p>
            </div>
        </div>
        <nav class="level is-mobile">
            <button class="level-item button is-sparta" onclick="sign_up()">
                회원가입
            </button>
            <button class="level-item button is-sparta is-outlined" onclick="toggle_sign_up()">
                취소
            </button>
        </nav>
    </div>

</div>
</article>
</div>
</section>

```

▼ [코드스니펫] - 로그인 박스 CSS

```

.button.is-sparta {
    background-color: #e8344e;
    border-color: transparent;
    color: #fff !important;
}

.button.is-sparta.is-outlined {
    background-color: transparent;
    border-color: #e8344e;
    color: #e8344e !important;
}

.help {
    color: gray;
}

```

- 우선 로그인/회원가입 상관 없이 모든 요소를 숨기지 않고 보였을 때의 모습을 볼까요?

SWEETER

세상을 달달하게

- 우선 모든 요소를 넣은 다음에, 로그인할 때는 파란색 요소들을 숨기고, 회원가입할 때는 초록색 요소들을 숨기면 되겠군요! 주황색 도움말 요소들도 숨겨져있다가 회원가입할 때 나타나야합니다.

▼ 13) 로그인/회원가입 토글 기능 만들기

- 이제 로그인 중인지, 회원가입 중인지 상황에 맞게 각 요소들을 숨겼다, 드러냈다 하는 기능을 만들어봅시다.
- Bulma에서는 `is-hidden` 이라는 클래스를 이용해서 요소를 숨길 수 있습니다. CSS로는 아래와 같이 정의되어있어요.

```
.is-hidden {
  display: none!important;
}
```

- 이 클래스를 로그인 화면에서 숨겨야하는 요소들에 붙여주세요.
- 이제 숨겨져 있으면 드러내고, 드러나 있으면 숨겨주는 함수를 만들어야겠죠? 우선 `sign-up-box` div에 적용시켜보겠습니다.

```
function toggle_sign_up() {
  if ($("#sign-up-box").hasClass("is-hidden")) {
    $("#sign-up-box").removeClass("is-hidden")
  } else {
    $("#sign-up-box").addClass("is-hidden")
  }
}
```

- jQuery에는 이것을 더 간단하게 도와주는 함수가 있는데요, 바로 `toggleClass()` 입니다.

```
function toggle_sign_up() {
  $("#sign-up-box").toggleClass("is-hidden")
}
```

```
}
```

- 이렇게 한 번에 토글할 수 있는 함수를 만들어 **회원가입하기** 버튼과 **취소** 버튼에 연결해주면 끝!

▼ [코드스니펫] - toggle_sign_up

```
function toggle_sign_up() {  
  $("#sign-up-box").toggleClass("is-hidden")  
  $("#div-sign-in-or-up").toggleClass("is-hidden")  
  $("#btn-check-dup").toggleClass("is-hidden")  
  $("#help-id").toggleClass("is-hidden")  
  $("#help-password").toggleClass("is-hidden")  
  $("#help-password2").toggleClass("is-hidden")  
}
```

08. 회원가입 페이지 기능 만들기

▼ 14) 회원가입 기능 만들기

- 회원가입할 때는 입력 받은 값들이 형식에 맞는지 우선 확인해야겠죠?
 - 아이디: 영문과 숫자, 일부 특수문자(._-)만 사용 가능, 2-10자 길이. 영문 무조건 포함
 - 비밀번호: 영문, 숫자는 1개 씩 무조건 포함, 일부 특수문자 사용 가능, 8-20자 길이
 - 비밀번호 확인: 비밀번호와 일치
- 이렇게 복잡한 조건을 확인할 때는 '정규표현식(Regular Expressions)'을 이용하여 비교하는 것이 좋습니다. 형식을 확인하여 결과를 참/거짓으로 반환하는 함수를 정의하면 편리하겠죠?

▼ [코드스니펫] - 아이디, 비밀번호 정규표현식

```
function is_nickname(asValue) {  
  var regExp = /^(?=[a-zA-Z])[a-zA-Z0-9_]{2,10}$/;  
  return regExp.test(asValue);  
}  
  
function is_password(asValue) {  
  var regExp = /^(?=[a-zA-Z])(?=.*[a-zA-Z])[0-9a-zA-Z!@#%&^*]{8,20}$/;  
  return regExp.test(asValue);  
}
```

- 그리고 아이디는 다른 사람과 겹치면 안되기 때문에 중복확인을 해주어야겠죠? 서버로 POST 요청을 보내 아이디가 존재하는지 확인해주세요.

▼ [코드스니펫] - 아이디 중복확인 클라이언트

```
function check_dup() {  
  let username = $("#input-username").val()  
  console.log(username)  
  if (username == "") {  
    $("#help-id").text("아이디를 입력해주세요.").removeClass("is-safe").addClass("is-danger")  
    $("#input-username").focus()  
    return;  
  }  
  if (!is_nickname(username)) {  
    $("#help-id").text("아이디의 형식을 확인해주세요. 영문과 숫자, 일부 특수문자(._-) 사용 가능. 2-10자 길이").removeClass("is-safe")  
    $("#input-username").focus()  
    return;  
  }  
  $("#help-id").addClass("is-loading")  
  $.ajax({  
    type: "POST",  
    url: "/sign_up/check_dup",  
    data: {  
      username_give: username  
    },  
    success: function (response) {  
      if (response["exists"]) {  
        $("#help-id").text("이미 존재하는 아이디입니다.").removeClass("is-safe").addClass("is-danger")  
        $("#input-username").focus()  
      } else {  

```

```

        $("#help-id").text("사용할 수 있는 아이디입니다.").removeClass("is-danger").addClass("is-success")
    }
    $("#help-id").removeClass("is-loading")

    }
    });
}

```

▼ [코드스니펫] - 아이디 중복확인 서버

```

@app.route('/sign_up/check_dup', methods=['POST'])
def check_dup():
    username_receive = request.form['username_give']
    exists = bool(db.users.find_one({"username": username_receive}))
    return jsonify({'result': 'success', 'exists': exists})

```

- 이제 이 조건들을 만족할 때만 회원가입 POST 요청을 보내도록 함수를 짜면 끝!

▼ [코드스니펫] - 회원가입 클라이언트

```

function sign_up() {
    let username = $("#input-username").val()
    let password = $("#input-password").val()
    let password2 = $("#input-password2").val()
    console.log(username, password, password2)

    if ($("#help-id").hasClass("is-danger")) {
        alert("아이디를 다시 확인해주세요.")
        return;
    } else if (!$("#help-id").hasClass("is-success")) {
        alert("아이디 중복확인을 해주세요.")
        return;
    }

    if (password == "") {
        $("#help-password").text("비밀번호를 입력해주세요.").removeClass("is-safe").addClass("is-danger")
        $("#input-password").focus()
        return;
    } else if (!is_password(password)) {
        $("#help-password").text("비밀번호의 형식을 확인해주세요. 영문과 숫자 필수 포함, 특수문자(!@#%&*) 사용가능 8-20자").removeClass("is-safe").addClass("is-danger")
        $("#input-password").focus()
        return
    } else {
        $("#help-password").text("사용할 수 있는 비밀번호입니다.").removeClass("is-danger").addClass("is-success")
    }

    if (password2 == "") {
        $("#help-password2").text("비밀번호를 입력해주세요.").removeClass("is-safe").addClass("is-danger")
        $("#input-password2").focus()
        return;
    } else if (password2 != password) {
        $("#help-password2").text("비밀번호가 일치하지 않습니다.").removeClass("is-safe").addClass("is-danger")
        $("#input-password2").focus()
        return;
    } else {
        $("#help-password2").text("비밀번호가 일치합니다.").removeClass("is-danger").addClass("is-success")
    }

    $.ajax({
        type: "POST",
        url: "/sign_up/save",
        data: {
            username_give: username,
            password_give: password
        },
        success: function (response) {
            alert("회원이입을 축하드립니다!")
            window.location.replace("/login")
        }
    });
}

```

▼ [코드스니펫] - 회원가입 서버

```

@app.route('/sign_up/save', methods=['POST'])
def sign_up():
    username_receive = request.form['username_give']
    password_receive = request.form['password_give']

```

```
password_hash = hashlib.sha256(password_receive.encode('utf-8')).hexdigest()
doc = {
    "username": username_receive,          # 아이디
    "password": password_hash,             # 비밀번호
    "profile_name": username_receive,      # 프로필 이름 기본값은 아이디
    "profile_pic": "",                     # 프로필 사진 파일 이름
    "profile_pic_real": "profile_pics/profile_placeholder.png", # 프로필 사진 기본 이미지
    "profile_info": ""                     # 프로필 한 마디
}
db.users.insert_one(doc)
return jsonify({'result': 'success'})
```

09. 로그인 페이지 기능 만들기

▼ 15) 로그인 기능 만들기

- 로그인 입력값 확인은 훨씬 간단합니다. 값을 입력했는지만 확인하고 바로 로그인 POST 요청을 보내주세요.

▼ [코드스니펫] - 로그인 클라이언트

```
function sign_in() {
    let username = $("#input-username").val()
    let password = $("#input-password").val()

    if (username == "") {
        $("#help-id-login").text("아이디를 입력해주세요.")
        $("#input-username").focus()
        return;
    } else {
        $("#help-id-login").text("")
    }

    if (password == "") {
        $("#help-password-login").text("비밀번호를 입력해주세요.")
        $("#input-password").focus()
        return;
    } else {
        $("#help-password-login").text("")
    }

    $.ajax({
        type: "POST",
        url: "/sign_in",
        data: {
            username_give: username,
            password_give: password
        },
        success: function (response) {
            if (response['result'] == 'success') {
                $.cookie('mytoken', response['token'], {path: '/'});
                window.location.replace("/")
            } else {
                alert(response['msg'])
            }
        }
    });
}
```

▼ [코드스니펫] - 로그인 서버

```
@app.route('/sign_in', methods=['POST'])
def sign_in():
    # 로그인
    username_receive = request.form['username_give']
    password_receive = request.form['password_give']

    pw_hash = hashlib.sha256(password_receive.encode('utf-8')).hexdigest()
    result = db.users.find_one({'username': username_receive, 'password': pw_hash})

    if result is not None:
        payload = {
            'id': username_receive,
            'exp': datetime.utcnow() + timedelta(seconds=60 * 60 * 24) # 로그인 24시간 유지
        }
        token = jwt.encode(payload, SECRET_KEY, algorithm='HS256').decode('utf-8')

        return jsonify({'result': 'success', 'token': token})
    # 찾지 못하면
```

```
else:
    return jsonify({'result': 'fail', 'msg': '아이디/비밀번호가 일치하지 않습니다.'})
```



위의 기능들을 테스트하면서 개발하려면 한 아이디로 로그인해서 다른 사용자의 프로필 페이지를 봐야할 때가 있으니 테스트 용 아이디를 2개 이상 만들어주세요 ㅎㅎ

10. 메인 페이지 모습 만들기

▼ 16) CSS 파일 분리하기

- CSS 내용 중 다른 페이지에서도 쓸만한 것들은 static 폴더 안에 mystyle.css 파일을 만들어 옮겨주시고, head에는 링크를 달아주세요. 전체적으로 통일감 있는 웹사이트를 만들 수 있겠죠?

▼ [코드스니펫] - CSS 파일 링크

```
<link href="{ url_for('static', filename='mystyle.css') }}" rel="stylesheet">
```

▼ [코드스니펫] - CSS 파일 내용

```
body {
    background-color: RGBA(232, 52, 78, 0.2);
    min-height: 100vh;
    padding-top: 3.25rem;
}

.section {
    padding: 1rem 1.5rem;
    max-width: 750px;
    margin: auto;
}

.is-sparta {
    color: #e8344e !important;
}

.button.is-sparta {
    background-color: #e8344e;
    border-color: transparent;
    color: #fff !important;
}

.button.is-sparta.is-outlined {
    background-color: transparent;
    border-color: #e8344e;
    color: #e8344e !important;
}

.modal-content {
    width: 600px;
    max-width: 80%;
}

input::-webkit-calendar-picker-indicator {
    display: none;
}

.image img {
    object-fit: cover;
    width: 100%;
    height: 100%;
}
```



Bulma의 클래스들을 커스터마이징해서 쓰기 때문에 Bulma의 링크보다 아래에 넣어주어야합니다.

▼ 17) 네비게이션 바 만들기

- 로고를 클릭하면 메인 페이지로 갈 수 있는 내비게이션 바를 만들어봅시다.

▼ [코드스니펫] - 내비게이션 바

```
<nav class="navbar is-fixed-top is-white" role="navigation" aria-label="main navigation">
  <div class="navbar-brand">
    <a class="navbar-item" href="/">
      
      <strong class="is-sparta"
        style="font-family: 'Stylish', sans-serif;font-size: larger;">SWEETER</strong>
    </a>
  </div>
</nav>
```

▼ 18) 포스팅 칸 만들기

- 클릭하면 글을 적을 수 있는 팝업이 뜨는 포스팅 칸입니다.

▼ [코드스니펫] - 포스팅 칸

```
<section class="section">
  <article class="media">
    <figure class="media-left" style="align-self: center">
      <a class="image is-32x32" href="#">
        
      </a>
    </figure>
    <div class="media-content">
      <div class="field">
        <p class="control">
          <input id="input-post" class="input is-rounded" placeholder="무슨 생각을 하고 계신가요?"
            onclick="$('#modal-post').addClass('is-active')">
          </p>
        </div>
      </div>
    </article>
  </section>
```

- 앞의 이미지를 사용자의 프로필 사진으로 넣어주기 위해서는 서버에서 사용자 정보를 같이 보내주어야 합니다.

▼ [코드스니펫] - 사용자 정보 보내주기

```
user_info = db.users.find_one({"username": payload["id"]})
return render_template('index.html', user_info=user_info)
```

```
@app.route('/')
def home():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        user_info = db.users.find_one({"username": payload["id"]})
        return render_template('index.html', user_info=user_info)
    except jwt.ExpiredSignatureError:
        return redirect(url_for("login", msg="로그인 시간이 만료되었습니다."))
    except jwt.exceptions.DecodeError:
        return redirect(url_for("login", msg="로그인 정보가 존재하지 않습니다."))
```

- 이제 이 정보를 이용해 이미지를 바꿔줄 수 있습니다. 이미지를 클릭하면 사용자 프로필로 가도록 링크도 바꾸어 주었습니다.

▼ [코드스니펫] - href

```
href="/user/{{ user_info.username }}"
```

▼ [코드스니펫] - src

```
src="{{ url_for('static', filename=user_info.profile_pic_real) }}"
```

```
<a class="image is-32x32" href="/user/{{ user_info.username }}">
  
</a>
```

▼ 19) 포스팅 모달 만들기

- 실제로 글을 적을 수 있는 포스팅 모달은 우선 모습을 만들고 숨겨놓았다가 포스팅 칸을 클릭하면 나타납니다. 바깥 배경 영역이나 X표, 취소 버튼을 클릭하면 사라집니다. 나타나고 사라지는 것은 `is-active` 클래스를 이용해 제어가 가능합니다.

▼ [코드스니펫] - 포스팅 모달

```
<div class="modal" id="modal-post">
  <div class="modal-background" onclick='$("#modal-post").removeClass("is-active")'></div>
  <div class="modal-content">
    <div class="box">
      <article class="media">
        <div class="media-content">
          <div class="field">
            <p class="control">
              <textarea id="textarea-post" class="textarea"
                placeholder="무슨 생각을 하고 계신가요?"></textarea>
            </p>
          </div>
          <nav class="level is-mobile">
            <div class="level-left">
              </div>
            <div class="level-right">
              <div class="level-item">
                <a class="button is-sparta" onclick="post()">포스팅하기</a>
              </div>
              <div class="level-item">
                <a class="button is-sparta is-outlined"
                  onclick='$("#modal-post").removeClass("is-active")'>취소</a>
              </div>
            </div>
          </nav>
        </div>
      </article>
    </div>
    <button class="modal-close is-large" aria-label="close"
      onclick='$("#modal-post").removeClass("is-active")'></button>
  </div>
</div>
```

▼ 20) 포스팅 카드 만들기

- Bulma의 `box` 와 `media` 클래스를 이용하면 예쁜 카드를 쉽게 만들 수 있습니다.

▼ [코드스니펫] - 포스팅 카드

```
<section class="section">
  <div id="post-box" class="container">
    <div class="box">
      <article class="media">
        <div class="media-left">
          <a class="image is-64x64" href="#">
            <img class="is-rounded"
              src={{ url_for("static", filename="profile_pics/profile_placeholder.png") }} alt="Image">
          </a>
        </div>
        <div class="media-content">
          <div class="content">
            <p>
              <strong>홍길동</strong> <small>@username</small> <small>10분 전</small>
              <br>
              글을 적는 칸
            </p>
          </div>
          <nav class="level is-mobile">
            <div class="level-left">
              <a class="level-item is-sparta" aria-label="heart" onclick="toggle_like('', 'heart')">
                <span class="icon is-small"><i class="fa fa-heart" aria-hidden="true"></i></span>&nbsp;</a>
            </div>
          </nav>
        </div>
      </article>
    </div>
  </div>
</section>
```


11. 메인 페이지 기능 만들기 - 포스팅

▼ 21) 포스팅 기능 만들기

- 글과 현재 시각을 문자열로 받아 POST 요청을 보내고, 저장에 성공하면 모달을 닫고 새로고침해줍니다.

▼ [코드스니펫] - 포스팅 함수

```
function post() {
  let comment = $("#textarea-post").val()
  let today = new Date().toISOString()
  $.ajax({
    type: "POST",
    url: "/posting",
    data: {
      comment_give: comment,
      date_give: today
    },
    success: function (response) {
      $("#modal-post").removeClass("is-active")
      window.location.reload()
    }
  })
}
```

- 서버에서는 글과 현재 시각을 받아 로그인한 사용자의 정보로부터 아이디, 이름, 프로필 사진을 같이 저장합니다.

▼ [코드스니펫] - 포스팅 API

```
user_info = db.users.find_one({"username": payload["id"]})
comment_receive = request.form["comment_give"]
date_receive = request.form["date_give"]
doc = {
  "username": user_info["username"],
  "profile_name": user_info["profile_name"],
  "profile_pic_real": user_info["profile_pic_real"],
  "comment": comment_receive,
  "date": date_receive
}
db.posts.insert_one(doc)
```

▼ 22) 포스팅 카드 띄우는 기능 만들기

- 포스트를 저장했으니 이번에는 받아와봅시다.
- 서버에서는 DB에서 최근 20개의 포스트를 받아와 리스트로 넘겨줍니다. 나중에 좋아요 기능을 쓸 때 각 포스트를 구분하기 위해서 MongoDB가 자동으로 만들어주는 `_id` 값을 이용할 것인데, `ObjectId` 라는 자료형이라 문자열로 변환해주어야합니다.

▼ [코드스니펫] - 문자열로 변환하기

```
posts = list(db.posts.find({}).sort("date", -1).limit(20))
for post in posts:
  post["_id"] = str(post["_id"])
```

- 클라이언트에서는 각 포스트를 카드로 만들어줍니다. 기존에 있던 카드는 다 지우고 새로 만들어서 담벼락에 붙여줍니다.

▼ [코드스니펫] - 포스팅 카드 만들기

```
function get_posts() {
  $("#post-box").empty()
  $.ajax({
    type: "GET",
    url: "/get_posts",
    data: {},
    success: function (response) {
      if (response["result"] == "success") {
        let posts = response["posts"]
        for (let i = 0; i < posts.length; i++) {
          let post = posts[i]
          let time_post = new Date(post["date"])
          let html_temp = `<div class="box" id="${post["_id"]}">`
        }
      }
    }
  })
}
```

```

<article class="media">
  <div class="media-left">
    <a class="image is-64x64" href="/user/${post['username']}">
      
    </a>
  </div>
  <div class="media-content">
    <div class="content">
      <p>
        <strong>${post['profile_name']}</strong> <small>@${post['username']}<br>
        ${post['comment']}
      </p>
    </div>
    <nav class="level is-mobile">
      <div class="level-left">
        <a class="level-item is-sparta" aria-label="heart" onclick="toggl
          <span class="icon is-small"><i class="fa fa-heart"
            aria-hidden="true"></i></span>
        </a>
      </div>
    </nav>
  </div>
</article>
</div>`
$("#post-box").append(html_temp)
  }
}
})
}

```

- 이 `get_posts()` 함수가 페이지가 로딩되었을 때, 실행되게 하면 되겠죠?

▼ [코드스니펫] - `get_posts` 실행하기

```

$(document).ready(function () {
  get_posts()
})

```

▼ 23) 포스팅 시간 나타내기

- 이번에는 포스팅한 지 얼마나 되었는지 보여주는 기능을 만들어봅시다.
- 자바스크립트의 `Date` 오브젝트 간의 빼기의 결과는 밀리초로 주어집니다.

```

function time2str(date) {
  let today = new Date()
  let time = (today - date) / 1000 / 60 // 분

  return parseInt(time) + "분 전"
}

```

- 60분이 넘어가는 경우에는 시간으로 나타내봅시다.

```

function time2str(date) {
  let today = new Date()
  let time = (today - date) / 1000 / 60 // 분

  if (time < 60) {
    return parseInt(time) + "분 전"
  }
  time = time / 60 // 시간
  return parseInt(time) + "시간 전"
}

```

- 24시간이 넘어가는 경우에는 일수로 나타내볼까요?

```

function time2str(date) {
  let today = new Date()
  let time = (today - date) / 1000 / 60 // 분

  if (time < 60) {

```

```

    return parseInt(time) + "분 전"
  }
  time = time / 60 // 시간
  if (time < 24) {
    return parseInt(time) + "시간 전"
  }
  time = time / 24
  return parseInt(time) + "일 전"
}

```

- 7일 이상일 때에는 날짜로 보여주도록 하겠습니다.

▼ [코드스니펫] - 포스팅 시간 나타내기

```

function time2str(date) {
  let today = new Date()
  let time = (today - date) / 1000 / 60 // 분

  if (time < 60) {
    return parseInt(time) + "분 전"
  }
  time = time / 60 // 시간
  if (time < 24) {
    return parseInt(time) + "시간 전"
  }
  time = time / 24
  if (time < 7) {
    return parseInt(time) + "일 전"
  }
  return `${date.getFullYear()}년 ${date.getMonth() + 1}월 ${date.getDate()}일`
}

```

- 이것을 `get_posts()` 함수 안에 각 포스팅 카드에 포스팅 시각 대신 넣어주면 되겠죠?

▼ [코드스니펫] - time2str(time_post)

```

let time_before = time2str(time_post)

```

12. 메인 페이지 기능 만들기 - 좋아요

▼ 24) 좋아요/좋아요 취소 기능 만들기

▼ 서버

- 우선 서버 쪽 기능을 먼저 생각해봅시다. 하트를 누르면 1) 어떤 포스트를 2) 누가 눌렀고 3) 좋아요인지 좋아요 취소인지를 알아야겠죠? 속제로 만들 다른 반응들(★, 👍)을 생각하면 어느 아이콘을 눌렀는지도 알아야겠네요.
- DB에 저장할 때는 1) 누가 2) 어떤 포스트에 3) 어떤 반응을 남겼는지 세 정보만 넣으면 되고, 좋아요인지, 취소인지에 따라 해당 도큐먼트를 `insert_one()` 을 할지 `delete_one()` 을 할지 결정해주어야합니다.

```

if action_receive == "like":
  db.likes.insert_one(doc)
else:
  db.likes.delete_one(doc)

```

- 좋아요 컬렉션을 업데이트한 이후에는 해당 포스트에 해당 타입의 반응이 몇 개인지를 세서 보내주어야합니다.

```

count = db.likes.count_documents({"post_id": post_id_receive, "type": type_receive})

```

▼ [코드스니펫] - 좋아요 업데이트 API

```

user_info = db.users.find_one({"username": payload["id"]})
post_id_receive = request.form["post_id_give"]
type_receive = request.form["type_give"]
action_receive = request.form["action_give"]
doc = {
  "post_id": post_id_receive,
  "username": user_info["username"],

```

```

    "type": type_receive
  }
  if action_receive == "like":
    db.likes.insert_one(doc)
  else:
    db.likes.delete_one(doc)
  count = db.likes.count_documents({"post_id": post_id_receive, "type": type_receive})
  return jsonify({"result": "success", 'msg': 'updated', "count": count})

```

▼ 클라이언트

- API에서 요구하는 데이터가 사용자 정보, 포스트 아이디, 좋아요/좋아요 취소, 아이콘 종류입니다.
- 여기에서 하트를 누른 사람의 정보는 로그인 정보에서 받아왔으므로 나머지 3개만 데이터로 보내주면 됩니다.
- 좋아요인지, 좋아요 취소인지는 아이콘의 클래스가 `fa-heart` 인지 `fa-heart-o` 인지로 알 수 있습니다.
- 업데이트에 성공하면 아이콘의 클래스를 바꾸고 좋아요 숫자도 업데이트해줍니다.

▼ [코드스니펫] - 좋아요 업데이트 함수 클라이언트

```

function toggle_like(post_id, type) {
  console.log(post_id, type)
  let $a_like = $('#${post_id} a[aria-label='heart']')
  let $i_like = $a_like.find("i")
  if ($i_like.hasClass("fa-heart")) {
    $.ajax({
      type: "POST",
      url: "/update_like",
      data: {
        post_id_give: post_id,
        type_give: type,
        action_give: "unlike"
      },
      success: function (response) {
        console.log("unlike")
        $i_like.addClass("fa-heart-o").removeClass("fa-heart")
        $a_like.find("span.like-num").text(response["count"])
      }
    })
  } else {
    $.ajax({
      type: "POST",
      url: "/update_like",
      data: {
        post_id_give: post_id,
        type_give: type,
        action_give: "like"
      },
      success: function (response) {
        console.log("like")
        $i_like.addClass("fa-heart").removeClass("fa-heart-o")
        $a_like.find("span.like-num").text(response["count"])
      }
    })
  }
}

```

▼ 25) 좋아요 숫자 표시하기

- ▼ 이제 좋아요 기능이 생겼으니 포스팅 카드를 만들 때도 좋아요 개수를 제대로 입력해주도록 합니다.

- 우선 서버에서 포스트 목록을 보내줄 때 그 포스트에 달린 하트가 몇 개인지, 내가 단 하트도 있는지 같이 세어 보내줍니다.

```

for post in posts:
  post["_id"] = str(post["_id"])
  post["count_heart"] = db.likes.count_documents({"post_id": post["_id"], "type": "heart"})
  post["heart_by_me"] = bool(db.likes.find_one({"post_id": post["_id"], "type": "heart", "username": payload['id']}))

```

- 클라이언트에서는 이 정보를 받아 찬 하트(`fa-heart`)를 보여줄 것인지, 빈 하트(`fa-heart-o`)를 보여줄 것인지 결정합니다.

```
let class_heart = ""
if (post["heart_by_me"]) {
  class_heart = "fa-heart"
} else {
  class_heart = "fa-heart-o"
}
```

- 이것을 '조건부 삼항 연산자(ternary operator)'를 쓰면 한 줄로 나타낼 수 있어요!

```
let class_heart = post['heart_by_me'] ? "fa-heart" : "fa-heart-o"
```

조건부 삼항 연산자는 다음과 같은 구조로 써주면 됩니다.

```
변수 = 조건 ? 참일 때 값 : 거짓일 때 값
```

- 이 정보를 `html_temp`를 만들 때 하트 개수와 함께 넣어주면 끝!

```
<span class="icon is-small"><i class="fa ${class_heart}" aria-hidden="true"></i></span>
  &nbsp;
  <span class="like-num">${post["count_heart"]}</span>
```

▼ 좋아요 숫자도 형식을 조금 바꿔볼까요?

- 우선 10,000개가 넘으면 '12k'처럼 정수+k 형식으로 만들어줍니다.
- 500개가 넘으면 '0.5k'처럼 소숫점 아래 한 자리 수에서 반올림해줍니다.
- 좋아요 수가 0개일 때는 숫자를 적지 않습니다.
- 작은 숫자는 그대로 적습니다.

▼ [코드스니펫] 좋아요 숫자 형식

```
function num2str(count) {
  if (count > 10000) {
    return parseInt(count / 1000) + "k"
  }
  if (count > 500) {
    return parseInt(count / 100) / 10 + "k"
  }
  if (count == 0) {
    return ""
  }
  return count
}
```

- `get_posts()` 와 `toggle_like()` 안에 넣어줍니다.

```
// get_posts()
<span class="icon is-small"><i class="fa ${class_heart}" aria-hidden="true"></i></span>
  &nbsp;
  <span class="like-num">${num2str(post["count_heart"])}</span>

// toggle_like()
$a_like.find("span.like-num").text(num2str(response["count"]))
```

▼ 26) JS 파일 분리하기

- 여러 페이지에 걸쳐 쓰이는 CSS 내용을 `mystyle.css` 파일에 적어 공유할 수 있듯, 자바스크립트 코드도 파일을 분리하여 다른 html에서 임포트해올 수 있습니다.
- 포스팅 관련 함수들은 프로필 페이지에서도 쓰이므로 JS 파일로 분리하여봅시다.
- static 폴더 안에 `myjs.js` 파일을 만들고 함수들을 복사해 넣습니다.
- HTML 안에는 아래와 같이 임포트해옵니다.

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

⚠ jQuery의 함수들을 사용하기 때문에 jQuery 임포트하는 코드보다 아래에 넣어주어야합니다.

13. 프로필 페이지 모습 만들기 - 전체

▼ 27) 틀 만들기

- 프로필 페이지의 모습은 메인 페이지와 아주 비슷하기 때문에 우선 복사해오겠습니다. mystyle.css와 myjs.js 파일 임포트하는 것을 잊지 마세요!

▼ [코드스니펫] - 프로필 페이지 템플릿

```
<body class="has-navbar-fixed-top">
  <nav class="navbar is-fixed-top is-white" role="navigation" aria-label="main navigation">
    <div class="navbar-brand">
      <a class="navbar-item" href="/">
        
        <strong class="is-sparta" style="font-family: 'Stylish', sans-serif;font-size: larger;">SWEETER</strong>
      </a>
    </div>
  </nav>
  <section class="section">
    <article class="media">
      <figure class="media-left" style="align-self: center">
        <a class="image is-32x32" href="/user/{{ user_info.username }}">
          
        </a>
      </figure>
      <div class="media-content">
        <div class="field">
          <p class="control">
            <input id="input-post" class="input is-rounded" placeholder="무슨 생각을 하고 계신가요?"
              onclick="$("#modal-post").addClass("is-active")">
          </p>
        </div>
      </div>
    </article>
    <div class="modal" id="modal-post">
      <div class="modal-background" onclick="$("#modal-post").removeClass("is-active")"></div>
      <div class="modal-content">
        <div class="box">
          <article class="media">
            <div class="media-content">
              <div class="field">
                <p class="control">
                  <textarea id="textarea-post" class="textarea"
                    placeholder="무슨 생각을 하고 계신가요?"></textarea>
                </p>
              </div>
            </div>
            <nav class="level is-mobile">
              <div class="level-left">
                </div>
              <div class="level-right">
                <div class="level-item">
                  <a class="button is-sparta" onclick="post()">포스팅하기</a>
                </div>
                <div class="level-item">
                  <a class="button is-sparta is-outlined"
                    onclick="$("#modal-post").removeClass("is-active")">취소</a>
                </div>
              </div>
            </nav>
          </div>
        </div>
      </div>
      <div>
        <button class="modal-close is-large" aria-label="close"
          onclick="$("#modal-post").removeClass("is-active")"></button>
      </div>
    </div>
  </section>
</section>
```

```

<div id="post-box" class="container">
  <div class="box">
    <article class="media">
      <div class="media-left">
        <a class="image is-64x64" href="#">
          <img class="is-rounded"
            src={{ url_for("static", filename="profile_pics/profile_placeholder.png") }} alt="Image">
        </a>
      </div>
      <div class="media-content">
        <div class="content">
          <p>
            <strong>홍길동</strong> <small>@username</small> <small>10분 전</small>
            <br>
            글을 적는 칸
          </p>
        </div>
        <nav class="level is-mobile">
          <div class="level-left">
            <a class="level-item is-sparta" aria-label="heart"
              onclick="toggle_like(' ', 'heart')">
              <span class="icon is-small"><i class="fa fa-heart"
                aria-hidden="true"></i></span>&nbsp;<span
                class="like-num">2.7k</span>
            </a>
          </div>
        </nav>
      </div>
    </article>
  </div>
</div>
</section>
</body>

```

- 페이지가 로딩되고 나면 포스팅 카드들을 띄워줍니다.

```

$(document).ready(function () {
  get_posts()
})

```

▼ 28) 프로필 영역 만들기

- 프로필 페이지에서는 각 사용자의 프로필이 보여야겠죠! `hero` 클래스와 `media` 클래스를 이용해 만들어보겠습니다.

▼ [코드스니펫] - 프로필 영역

```

<section class="hero is-white">
  <div class="hero-body" style="padding-bottom:1rem;margin:auto;min-width: 400px">
    <article class="media">
      <figure class="media-left" style="align-self: center">
        <a class="image is-96x96" href="#">
          
        </a>
      </figure>
      <div class="media-content">
        <div class="content">
          <p>
            <strong>{{ user_info.profile_name }}</strong> <small>@{{ user_info.username }}</small>
            <br>
            {{ user_info.profile_info }}
          </p>
        </div>
      </div>
    </article>
  </div>
</section>

```

14. 프로필 페이지 모습 만들기 - 프로필 수정

▼ 29) 프로필 수정 & 로그아웃 버튼 만들기

- 내 프로필 페이지에 들어갔을 때에는 프로필 수정과 로그아웃 버튼이 보여야합니다.

▼ [코드스니펫] - 내 프로필 페이지 버튼 추가

```
<nav id="btms-me" class="level is-mobile" style="margin-top:2rem">
  <a class="button level-item has-text-centered is-sparta" aria-label="edit"
    onclick='$("#modal-edit").addClass("is-active")'>
    프로필 수정<span class="icon is-small"><i class="fa fa-pencil"
      aria-hidden="true"></i></span>
    </a>

  <a class="button level-item has-text-centered is-sparta is-outlined" aria-label="logout"
    onclick="sign_out()">
    로그아웃<span class="icon is-small"><i class="fa fa-sign-out"
      aria-hidden="true"></i></span>
    </a>
</nav>
```

- 로그아웃 기능은 이미 프로필 시작 코드에 다 들어있었죠? 로그아웃 버튼을 클릭하면 토큰을 삭제하고 로그인 페이지로 이동하면 끝!

```
function sign_out() {
  $.removeCookie('mytoken', {path: '/'});
  alert('로그아웃!')
  window.location.href = "/login"
}
```

▼ 30) 프로필 수정 모달 만들기

- 프로필 수정 버튼을 누르면 나오는 모달입니다.

▼ [코드스니펫] - 프로필 수정 모달

```
<div class="modal" id="modal-edit">
  <div class="modal-background" onclick='$("#modal-edit").removeClass("is-active")'></div>
  <div class="modal-content">
    <div class="box">
      <article class="media">
        <div class="media-content">
          <div class="field">
            <label class="label" for="input-name">이름</label>

            <p class="control">

              <input id="input-name" class="input"
                placeholder="홍길동" value="{ { user_info.profile_name } }">

            </p>
          </div>
          <div class="field">
            <label class="label" for="input-pic">프로필 사진</label>

            <div class="control is-expanded">
              <div class="file has-name">
                <label class="file-label" style="width:100%">
                  <input id="input-pic" class="file-input" type="file"
                    name="resume">
                  <span class="file-cta"><span class="file-icon"><i
                    class="fa fa-upload"></i></span>
                  <span class="file-label">파일 선택</span>
                </span>
                <span id="file-name" class="file-name"
                  style="width:100%;max-width:100%">{ { user_info.profile_pic } }</span>
              </label>
            </div>
          </div>
        </div>
      </div>
      <div class="field">
        <label class="label" for="textarea-about">나는 누구?</label>

        <p class="control">

          <textarea id="textarea-about" class="textarea"
            placeholder="자기소개하기">{ { user_info.profile_info } }</textarea>

        </p>
      </div>
    <div class="level is-mobile">
      <div class="level-left">

      </div>
      <div class="level-right">
```



```

        <div class="level-item">
          <a class="button is-sparta" onclick="update_profile()">업데이트</a>
        </div>
        <div class="level-item">
          <a class="button is-sparta is-outlined"
            onclick='$("#modal-edit").removeClass("is-active")'>취소</a>
        </div>
      </div>
    </nav>
  </div>
</article>
</div>
<div>
  <button class="modal-close is-large" aria-label="close"
    onclick='$("#modal-edit").removeClass("is-active")'></button>
</div>

```

15. 프로필 페이지 기능 만들기

▼ 31) 내 프로필에서만 프로필 수정기능 보이게 하기

- 프로필 수정 & 로그아웃 버튼은 내 프로필에 들어갔을 때만 보여야겠죠? 서버에서 보내준 `status` 파라미터를 이용해 내 프로필 일 때만 해당 부분을 그리도록 jinja2 문법을 씁니다.

```

{% if status %}

<nav id="btns-me" class="level is-mobile" ...>
<div class="modal" id="modal-edit" ...>

{% endif %}

```

- 글을 적는 포스팅 칸과 모달도 내 프로필에서만 보이게 해줍니다.

```

{% if status %}

<section id="section-post" class="section" ...>

{% endif %}

```

▼ 32) 프로필 수정 기능 만들기

- 프로필 수정 모달에서 이름을 바꾸거나 새 프로필 사진을 업로드하는 경우 파일을 받아 저장해주어야합니다. 프로필 업데이트 후 에는 페이지를 새로고침하여 다시 정보를 받아옵니다.

▼ [코드스니펫] - 프로필 수정 클라이언트

```

function update_profile() {
  let name = $('#input-name').val()
  let file = $('#input-pic')[0].files[0]
  let about = $('#textarea-about').val()
  let form_data = new FormData()
  form_data.append("file_give", file)
  form_data.append("name_give", name)
  form_data.append("about_give", about)
  console.log(name, file, about, form_data)

  $.ajax({
    type: "POST",
    url: "/update_profile",
    data: form_data,
    cache: false,
    contentType: false,
    processData: false,
    success: function (response) {
      if (response["result"] == "success") {
        alert(response["msg"])
        window.location.reload()
      }
    }
  });
}

```

▼ [코드스니펫] - 프로필 수정 서버

```
@app.route('/update_profile', methods=['POST'])
def save_img():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        username = payload["id"]
        name_receive = request.form["name_give"]
        about_receive = request.form["about_give"]
        new_doc = {
            "profile_name": name_receive,
            "profile_info": about_receive
        }
        if 'file_give' in request.files:
            file = request.files["file_give"]
            filename = secure_filename(file.filename)
            extension = filename.split(".")[-1]
            file_path = f"profile_pics/{username}.{extension}"
            file.save("./static/"+file_path)
            new_doc["profile_pic"] = filename
            new_doc["profile_pic_real"] = file_path
        db.users.update_one({'username': payload['id']}, {'$set': new_doc})
        return jsonify({"result": "success", 'msg': '프로필을 업데이트했습니다.'})
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))
```

▼ 33) 해당 사용자 글만 보이게 하기

- 포스팅 카드들 중에 해당 사용자 글만 보여주게 해보겠습니다.
- 아까 만든 `get_posts()` 함수에 `username` 을 변수로 받도록 바꿔볼까요?

```
function get_posts(username) {
    if (username==undefined) {
        username=""
    }
    $("#post-box").empty()
    $.ajax({
        type: "GET",
        url: `/get_posts?username_give=${username}`,
        data: {},
        success: function (response) {
            if (response["result"] == "success") {
                ...
            }
        }
    })
}
```

- 이제 서버 쪽에서 username을 받아 해당 사용자의 글만 가져오도록 바꿔봅시다.

```
@app.route("/get_posts", methods=['GET'])
def get_posts():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        username_receive = request.args.get("username_give")
        if username_receive=="":
            posts = list(db.posts.find({}).sort("date", -1).limit(20))
        else:
            posts = list(db.posts.find({"username":username_receive}).sort("date", -1).limit(20))
        for post in posts:
            post["_id"] = str(post["_id"])
            post["count_heart"] = db.likes.count_documents({"post_id": post["_id"], "type": "heart"})
            post["heart_by_me"] = bool(db.likes.find_one({"post_id": post["_id"], "type": "heart", "username": my_username}))
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))
```

▼ 34) og태그, favicon 넣기

- 마지막 웹서비스도 Open Graph 태그와 favicon을 넣어서 완성해줍니다. ogimg는 로그인하면 배너를 스크린샷을 찍고 favicon은 아래 파일을 다운 받아 static 폴더에 넣어줄게요.

▼ [코드스니펫] - favicon

```
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8ba29765-3ffb-42d4-bff1-aa1e68a
e60e9/favicon.ico
```

- HTML 파일들의 head에 링크를 첨부합니다. 내용은 각 페이지에 맞게 바꿔주어야겠죠?

▼ [코드스니펫] - og태그와 favicon 링크

```
<link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
<link rel="icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
<meta property="og:title" content="스위터 - 세상을 달달하게"/>
<meta property="og:description" content="mini project for Web Plus"/>
<meta property="og:image" content="{{ url_for('static', filename='ogimg.png') }}" />
```

▼ 35) 전체 완성 코드

▼ app.py

```
from pymongo import MongoClient
import jwt
import datetime
import hashlib
from flask import Flask, render_template, jsonify, request, redirect, url_for
from werkzeug.utils import secure_filename
from datetime import datetime, timedelta

app = Flask(__name__)
app.config["TEMPLATES_AUTO_RELOAD"] = True
app.config['UPLOAD_FOLDER'] = './static/profile_pics'

SECRET_KEY = 'SPARTA'

client = MongoClient('18.188.71.218', 27017, username="sparta", password="qwer1234")
db = client.dbsparta_plus_week4

@app.route('/')
def home():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        user_info = db.users.find_one({"username": payload["id"]})
        return render_template('index.html', user_info=user_info)
    except jwt.ExpiredSignatureError:
        return redirect(url_for("login", msg="로그인 시간이 만료되었습니다."))
    except jwt.exceptions.DecodeError:
        return redirect(url_for("login", msg="로그인 정보가 존재하지 않습니다."))

@app.route('/login')
def login():
    msg = request.args.get("msg")
    return render_template('login.html', msg=msg)

@app.route('/user/<username>')
def user(username):
    # 각 사용자의 프로필과 글을 모아볼 수 있는 공간
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        status = (username == payload["id"]) # 내 프로필이면 True, 다른 사람 프로필 페이지면 False

        user_info = db.users.find_one({"username": username}, {"_id": False})
        return render_template('user.html', user_info=user_info, status=status)
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

@app.route('/sign_in', methods=['POST'])
def sign_in():
    # 로그인
    username_receive = request.form['username_give']
    password_receive = request.form['password_give']

    pw_hash = hashlib.sha256(password_receive.encode('utf-8')).hexdigest()
    result = db.users.find_one({'username': username_receive, 'password': pw_hash})
```

```

if result is not None:
    payload = {
        'id': username_receive,
        'exp': datetime.utcnow() + timedelta(seconds=60 * 60 * 24) # 로그인 24시간 유지
    }
    token = jwt.encode(payload, SECRET_KEY, algorithm='HS256').decode('utf-8')

    return jsonify({'result': 'success', 'token': token})
# 찾지 못하면
else:
    return jsonify({'result': 'fail', 'msg': '아이디/비밀번호가 일치하지 않습니다.'})

@app.route('/sign_up/save', methods=['POST'])
def sign_up():
    username_receive = request.form['username_give']
    password_receive = request.form['password_give']
    password_hash = hashlib.sha256(password_receive.encode('utf-8')).hexdigest()
    doc = {
        "username": username_receive,
        "password": password_hash,
        "profile_name": username_receive,
        "profile_pic": "",
        "profile_pic_real": "profile_pics/profile_placeholder.png",
        "profile_info": ""
    }
    db.users.insert_one(doc)
    return jsonify({'result': 'success'})

@app.route('/sign_up/check_dup', methods=['POST'])
def check_dup():
    username_receive = request.form['username_give']
    exists = bool(db.users.find_one({"username": username_receive}))
    # print(value_receive, type_receive, exists)
    return jsonify({'result': 'success', 'exists': exists})

@app.route('/update_profile', methods=['POST'])
def save_img():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        username = payload["id"]
        name_receive = request.form["name_give"]
        about_receive = request.form["about_give"]
        new_doc = {
            "profile_name": name_receive,
            "profile_info": about_receive
        }
        if 'file_give' in request.files:
            file = request.files["file_give"]
            filename = secure_filename(file.filename)
            extension = filename.split(".")[1]
            file_path = f"profile_pics/{username}.{extension}"
            file.save("./static/"+file_path)
            new_doc["profile_pic"] = filename
            new_doc["profile_pic_real"] = file_path
        db.users.update_one({'username': payload['id']}, {'$set': new_doc})
        return jsonify({"result": "success", 'msg': '프로필을 업데이트했습니다.'})
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

@app.route('/posting', methods=['POST'])
def posting():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        # 포스팅하기
        user_info = db.users.find_one({"username": payload["id"]})
        comment_receive = request.form["comment_give"]
        date_receive = request.form["date_give"]
        print(type(date_receive))
        doc = {
            "username": user_info["username"],
            "profile_name": user_info["profile_name"],
            "profile_pic_real": user_info["profile_pic_real"],
            "comment": comment_receive,
            "date": date_receive
        }
        db.posts.insert_one(doc)
        return jsonify({"result": "success", 'msg': '포스팅 성공'})
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

```

```
@app.route("/get_posts", methods=['GET'])
def get_posts():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        username_receive = request.args.get("username_give")
        if username_receive=="":
            posts = list(db.posts.find({}).sort("date", -1).limit(20))
        else:
            posts = list(db.posts.find({"username":username_receive}).sort("date", -1).limit(20))
        for post in posts:
            post["_id"] = str(post["_id"])
            post["count_heart"] = db.likes.count_documents({"post_id": post["_id"], "type": "heart"})
            post["heart_by_me"] = bool(db.likes.find_one({"post_id": post["_id"], "type": "heart", "username": my_username}))
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

@app.route('/update_like', methods=['POST'])
def update_like():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        # 좋아요 수 변경
        user_info = db.users.find_one({"username": payload["id"]})
        post_id_receive = request.form["post_id_give"]
        type_receive = request.form["type_give"]
        action_receive = request.form["action_give"]
        doc = {
            "post_id": post_id_receive,
            "username": user_info["username"],
            "type": type_receive
        }
        if action_receive == "like":
            db.likes.insert_one(doc)
        else:
            db.likes.delete_one(doc)
        count = db.likes.count_documents({"post_id": post_id_receive, "type": type_receive})
        print(count)
        return jsonify({"result": "success", 'msg': 'updated', "count": count})
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)
```

▼ login.html

```
<!doctype html>
<html lang="en">
  <head>

    <!-- Webpage Title -->
    <title>Log In | SWEETER</title>
    <link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
    <link rel="icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
    <meta property="og:title" content="스위터 - 세상을 달달하게"/>
    <meta property="og:description" content="mini project for Web Plus"/>
    <meta property="og:image" content="{{ url_for('static', filename='ogimg.png') }}" />
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bulma CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
    <!-- Font Awesome CSS -->
    <link href="//maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Gamja+Flower&family=Stylish&display=swap" rel="stylesheet">
    <link href="{{ url_for('static', filename='mystyle.css') }}" rel="stylesheet">

    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.js"></script>

  </head>
  <body>
    <div class="container">
      <div class="row">
        <div class="col">
          <div class="card">
            <div class="card-header">
              <h3>로그인</h3>
            </div>
            <div class="card-body">
              <div class="form">
                <div class="form-group">
                  <input type="text" value="이름" />
                </div>
                <div class="form-group">
                  <input type="password" value="비밀번호" />
                </div>
                <div class="form-group">
                  <input type="button" value="로그인" />
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>

<style>
  .title {
    font-weight: 800;
    font-size: 5rem;
  }
</style>
```

```

        font-family: 'Stylish', sans-serif;
    }

    .subtitle {
        font-family: 'Gamja Flower', cursive;
        font-size: 2rem;
    }

    .help {
        color: gray;
    }
}
</style>
<script>
    // {% if msg %}
    //     alert("{ msg }")
    // {% endif %}

    function toggle_sign_up() {
        $("#sign-up-box").toggleClass("is-hidden")
        $("#div-sign-in-or-up").toggleClass("is-hidden")
        $("#btn-check-dup").toggleClass("is-hidden")
        $("#help-id").toggleClass("is-hidden")
        $("#help-password").toggleClass("is-hidden")
        $("#help-password2").toggleClass("is-hidden")
    }

    function is_nickname(asValue) {
        var regExp = /^(?=[a-zA-Z])[-a-zA-Z0-9_]{2,10}$/;
        return regExp.test(asValue);
    }

    function is_password(asValue) {
        var regExp = /^(?=[a-zA-Z])(?=.*[a-zA-Z])[0-9a-zA-Z!@#%&*]{8,20}$/;
        return regExp.test(asValue);
    }

    function check_dup() {
        let username = $("#input-username").val()
        console.log(username)
        if (username == "") {
            $("#help-id").text("아이디를 입력해주세요.").removeClass("is-safe").addClass("is-danger")
            $("#input-username").focus()
            return;
        }
        if (!is_nickname(username)) {
            $("#help-id").text("아이디의 형식을 확인해주세요. 영문과 숫자, 일부 특수문자(._-) 사용 가능. 2-10자 길이").removeClass("is-safe").addClass("is-danger")
            $("#input-username").focus()
            return;
        }
        $("#help-id").addClass("is-loading")
        $.ajax({
            type: "POST",
            url: "/sign_up/check_dup",
            data: {
                username_give: username
            },
            success: function (response) {

                if (response["exists"]) {
                    $("#help-id").text("이미 존재하는 아이디입니다.").removeClass("is-safe").addClass("is-danger")
                    $("#input-username").focus()
                } else {
                    $("#help-id").text("사용할 수 있는 아이디입니다.").removeClass("is-danger").addClass("is-success")
                }
                $("#help-id").removeClass("is-loading")
            }
        })
    }

    function sign_up() {
        let username = $("#input-username").val()
        let password = $("#input-password").val()
        let password2 = $("#input-password2").val()
        console.log(username, password, password2)

        if ($("#help-id").hasClass("is-danger")) {
            alert("아이디를 다시 확인해주세요.")
            return;
        } else if (!$("#help-id").hasClass("is-success")) {
            alert("아이디 중복확인을 해주세요.")
            return;
        }

        if (password == "") {

```

```

        $("#help-password").text("비밀번호를 입력해주세요.").removeClass("is-safe").addClass("is-danger")
        $("#input-password").focus()
        return;
    } else if (!is_password(password)) {
        $("#help-password").text("비밀번호의 형식을 확인해주세요. 영문과 숫자 필수 포함, 특수문자(!@#$%^&*) 사용가능 8-20자").rem
        $("#input-password").focus()
        return
    } else {
        $("#help-password").text("사용할 수 있는 비밀번호입니다.").removeClass("is-danger").addClass("is-success")
    }
    if (password2 == "") {
        $("#help-password2").text("비밀번호를 입력해주세요.").removeClass("is-safe").addClass("is-danger")
        $("#input-password2").focus()
        return;
    } else if (password2 != password) {
        $("#help-password2").text("비밀번호가 일치하지 않습니다.").removeClass("is-safe").addClass("is-danger")
        $("#input-password2").focus()
        return;
    } else {
        $("#help-password2").text("비밀번호가 일치합니다.").removeClass("is-danger").addClass("is-success")
    }
    $.ajax({
        type: "POST",
        url: "/sign_up/save",
        data: {
            username_give: username,
            password_give: password
        },
        success: function (response) {
            alert("회원가입을 축하드립니다!")
            window.location.replace("/login")
        }
    });
}

function sign_in() {
    let username = $("#input-username").val()
    let password = $("#input-password").val()

    if (username == "") {
        $("#help-id-login").text("아이디를 입력해주세요.")
        $("#input-username").focus()
        return;
    } else {
        $("#help-id-login").text("")
    }

    if (password == "") {
        $("#help-password-login").text("비밀번호를 입력해주세요.")
        $("#input-password").focus()
        return;
    } else {
        $("#help-password-login").text("")
    }
    $.ajax({
        type: "POST",
        url: "/sign_in",
        data: {
            username_give: username,
            password_give: password
        },
        success: function (response) {
            if (response['result'] == 'success') {
                $.cookie('mytoken', response['token'], {path: '/'});
                window.location.replace("/")
            } else {
                alert(response['msg'])
            }
        }
    });
}

```

</script>

</head>

<body>

<section class="hero is-white">

<div class="hero-body has-text-centered" style="padding-bottom:1rem;margin:auto;">

<h1 class="title is-sparta">SWEETER</h1>

<h3 class="subtitle is-sparta">세상을 달달하게</h3>

</div>

</section>

<section class="section">

<div class="container">

<div class="box" style="max-width: 480px;margin:auto">

```

<article class="media">
  <div class="media-content">
    <div class="content">
      <div class="field has-addons">
        <div class="control has-icons-left" style="width:100%">
          <input id="input-username" class="input" type="text" placeholder="아이디">
          <span class="icon is-small is-left"><i class="fa fa-user"></i></span>
        </div>
        <div id="btn-check-dup" class="control is-hidden">
          <button class="button is-sparta" onclick="check_dup()">중복확인</button>
        </div>
      </div>
      <p id="help-id" class="help is-hidden">아이디는 2-10자의 영문과 숫자와 일부 특수문자(._-)만 입력 가능합니다.
      <p id="help-id-login" class="help is-danger"></p>

      <div class="field">
        <div class="control has-icons-left">
          <input id="input-password" class="input" type="password" placeholder="비밀번호">
          <span class="icon is-small is-left"><i class="fa fa-lock"></i></span>
        </div>
        <p id="help-password" class="help is-hidden">영문과 숫자 조합의 8-20자의 비밀번호를 설정해주세요.
        특수문자(!@#$$%^&*)도 사용 가능합니다.</p>
        <p id="help-password-login" class="help is-danger"></p>
      </div>

    </div>
    <div id="div-sign-in-or-up" class="has-text-centered">
      <nav class="level is-mobile">
        <button class="level-item button is-sparta" onclick="sign_in()">
          로그인
        </button>

      </nav>
      <hr>
      <h4 class="mb-3">아직 회원이 아니라면</h4>
      <nav class="level is-mobile">
        <button class="level-item button is-sparta is-outlined"
          onclick="toggle_sign_up()">
          회원가입하기
        </button>
      </nav>
    </div>

    <div id="sign-up-box" class="is-hidden">
      <div class="mb-5">
        <div class="field">
          <div class="control has-icons-left" style="width:100%">
            <input id="input-password2" class="input" type="password"
              placeholder="비밀번호 재입력">
            <span class="icon is-small is-left"><i class="fa fa-lock"></i></span>
          </div>
          <p id="help-password2" class="help is-hidden">비밀번호를 다시 한 번 입력해주세요.</p>
        </div>
      </div>
      <nav class="level is-mobile">
        <button class="level-item button is-sparta" onclick="sign_up()">
          회원가입
        </button>
        <button class="level-item button is-sparta is-outlined" onclick="toggle_sign_up()">
          취소
        </button>
      </nav>
    </div>

  </div>
</article>
</div>

</div>
</section>

</body>
</html>

```

▼ index.html

```

<!doctype html>
<html lang="en">

```



```

<head>

  <!-- Webpage Title -->
  <title>Home | SWEETER</title>
  <link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
  <link rel="icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
  <meta property="og:title" content="스위터 - 세상을 달달하게"/>
  <meta property="og:description" content="mini project for Web Plus"/>
  <meta property="og:image" content="{{ url_for('static', filename='ogimg.png') }}" />
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <!-- Bulma CSS -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
  <!-- Font Awesome CSS -->
  <link href="//maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=Gamja+Flower&family=Stylish&display=swap" rel="stylesheet">
  <link href="{{ url_for('static', filename='mystyle.css') }}" rel="stylesheet">

  <!-- JS -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.js"></script>
  <script src="{{ url_for('static', filename='myjs.js') }}"></script>

  <script>
    $(document).ready(function () {
      get_posts()
    })
  </script>

</head>
<body class="has-navbar-fixed-top">
  <nav class="navbar is-fixed-top is-white" role="navigation" aria-label="main navigation">
    <div class="navbar-brand">
      <a class="navbar-item" href="/">
        
        <strong class="is-sparta" style="font-family: 'Stylish', sans-serif;font-size: larger;">SWEETER</strong>
      </a>
    </div>
  </nav>
  <section class="section">
    <article class="media">
      <figure class="media-left" style="align-self: center">
        <a class="image is-32x32" href="/user/{{ user_info.username }}">
          
        </a>
      </figure>
      <div class="media-content">
        <div class="field">
          <p class="control">
            <input id="input-post" class="input is-rounded" placeholder="무슨 생각을 하고 계신가요?"
              onclick='$("#modal-post").addClass("is-active")'>
          </p>
        </div>
      </div>
    </article>
    <div class="modal id="modal-post">
      <div class="modal-background" onclick='$("#modal-post").removeClass("is-active")'></div>
      <div class="modal-content">
        <div class="box">
          <article class="media">
            <div class="media-content">
              <div class="field">
                <p class="control">
                  <textarea id="textarea-post" class="textarea"
                    placeholder="무슨 생각을 하고 계신가요?"></textarea>
                </p>
              </div>
              <nav class="level is-mobile">
                <div class="level-left">
                  </div>
                <div class="level-right">
                  <div class="level-item">
                    <a class="button is-sparta" onclick="post()">포스팅하기</a>
                  </div>
                  <div class="level-item">
                    <a class="button is-sparta is-outlined"
                      onclick='$("#modal-post").removeClass("is-active")'>취소</a>
                  </div>
                </div>
              </nav>
            </div>
          </article>
        </div>
      </div>
    </div>
  </div>

```

```

        </nav>
      </div>
    </article>
  </div>
</div>
<button class="modal-close is-large" aria-label="close"
  onclick='$("#modal-post").removeClass("is-active")'></button>
</div>

</section>
<section class="section">
  <div id="post-box" class="container">
    <div class="box">
      <article class="media">
        <div class="media-left">
          <a class="image is-64x64" href="#">
            <img class="is-rounded"
              src={{ url_for("static", filename="profile_pics/profile_placeholder.png") }} alt="Image":
          </a>
        </div>
        <div class="media-content">
          <div class="content">
            <p>
              <strong>홍길동</strong> <small>@username</small> <small>10분 전</small>
              <br>
              글을 적는 칸
            </p>
          </div>
          <nav class="level is-mobile">
            <div class="level-left">
              <a class="level-item is-sparta" aria-label="heart"
                onclick="toggle_like('', 'heart')">
                <span class="icon is-small"><i class="fa fa-heart"
                  aria-hidden="true"></i></span>&nbsp;<span
                  class="like-num">2.7k</span>
              </a>
            </div>
          </nav>
        </div>
      </article>
    </div>
  </div>
</section>
</body>
</html>

```

▼ user.html

```

<!doctype html>
<html lang="en">
  <head>

    <!-- Webpage Title -->
    <title>{{ user_info.profile_name }} | SWEETER</title>
    <link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
    <link rel="icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
    <meta property="og:title" content="스위터 - 세상을 달달하게"/>
    <meta property="og:description" content="mini project for Web Plus"/>
    <meta property="og:image" content="{{ url_for('static', filename='ogimg.png') }}" />
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bulma CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
    <!-- Font Awesome CSS -->
    <link href="//maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Gamja+Flower&family=Stylish&display=swap" rel="stylesheet">
    <link href="{{ url_for('static', filename='mystyle.css') }}" rel="stylesheet">

    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.js"></script>
    <script src="{{ url_for('static', filename='myjs.js') }}"></script>

    <script>
      $(document).ready(function () {
        get_posts("{{ user_info.username }}")
      })
    </script>

```

```

function sign_out() {
    $.removeCookie('mytoken', {path: '/'});
    alert('로그아웃!')
    window.location.href = "/login"
}

function update_profile() {
    let name = $('#input-name').val()
    let file = $('#input-pic')[0].files[0]
    let about = $('#textarea-about').val()
    let form_data = new FormData()
    form_data.append("file_give", file)
    form_data.append("name_give", name)
    form_data.append("about_give", about)
    console.log(name, file, about, form_data)

    $.ajax({
        type: "POST",
        url: "/update_profile",
        data: form_data,
        cache: false,
        contentType: false,
        processData: false,
        success: function (response) {
            if (response["result"] == "success") {
                alert(response["msg"])
                window.location.reload()
            }
        }
    });
}
}
</script>

</head>
<body class="has-navbar-fixed-top">
    <nav class="navbar is-fixed-top is-white" role="navigation" aria-label="main navigation">
        <div class="navbar-brand">
            <a class="navbar-item" href="/">
                
                <strong class="is-sparta" style="font-family: 'Stylish', sans-serif;font-size: larger;">SWEETER</strong>
            </a>
        </div>
    </nav>

    <section class="hero is-white">
        <div class="hero-body" style="padding-bottom:1rem;margin:auto;min-width: 400px">
            <article class="media">

                <figure class="media-left" style="align-self: center">
                    <a class="image is-96x96" href="#">
                        
                    </a>
                </figure>
                <div class="media-content">
                    <div class="content">
                        <p>
                            <strong>{{ user_info.profile_name }}</strong> <small>@{{ user_info.username }}</small>
                            <br>
                            {{ user_info.profile_info }}
                        </p>
                    </div>
                </div>
            </article>
            {% if status %}
                <nav id="btns-me" class="level is-mobile" style="margin-top:2rem">
                    <a class="button level-item has-text-centered is-sparta" aria-label="edit"
                        onclick='$("#modal-edit").addClass("is-active")'>
                        프로필 수정<span class="icon is-small"><i class="fa fa-pencil"
                            aria-hidden="true"></i></span>
                    </a>

                    <a class="button level-item has-text-centered is-sparta is-outlined" aria-label="logout"
                        onclick="sign_out()">
                        로그아웃<span class="icon is-small"><i class="fa fa-sign-out"
                            aria-hidden="true"></i></span>
                    </a>
                </nav>

                <div class="modal" id="modal-edit">
                    <div class="modal-background" onclick='$("#modal-edit").removeClass("is-active")'></div>

```

```

<div class="modal-content">
  <div class="box">
    <article class="media">
      <div class="media-content">
        <div class="field">
          <label class="label" for="input-name">이름</label>

          <p class="control">

            <input id="input-name" class="input"
              placeholder="홍길동" value="{{ user_info.profile_name }}">

          </p>
        </div>
        <div class="field">
          <label class="label" for="input-pic">프로필 사진</label>

          <div class="control is-expanded">
            <div class="file has-name">
              <label class="file-label" style="width:100%">
                <input id="input-pic" class="file-input" type="file"
                  name="resume">
                <span class="file-cta"><span class="file-icon"><i
                  class="fa fa-upload"></i></span>
                <span class="file-label">파일 선택</span>
              </span>
              <span id="file-name" class="file-name"
                style="width:100%;max-width:100%">{{ user_info.profile_pic }}</span>
            </div>
          </div>
        </div>
        <div class="field">
          <label class="label" for="textarea-about">나는 누구?</label>

          <p class="control">

            <textarea id="textarea-about" class="textarea"
              placeholder="자기소개하기">{{ user_info.profile_info }}</textarea>

          </p>
        </div>
        <nav class="level is-mobile">
          <div class="level-left">

          </div>
          <div class="level-right">
            <div class="level-item">
              <a class="button is-sparta" onclick="update_profile()">업데이트</a>
            </div>
            <div class="level-item">
              <a class="button is-sparta is-outlined"
                onclick='$("#modal-edit").removeClass("is-active")'>취소</a>
            </div>
          </div>
        </nav>
      </div>
    </article>
  </div>
  <button class="modal-close is-large" aria-label="close"
    onclick='$("#modal-edit").removeClass("is-active")'></button>
</div>
{% endif %}
</div>
</section>
{% if status %}
  <section id="section-post" class="section">
    <article class="media">
      <figure class="media-left" style="align-self: center">
        <a class="image is-32x32" href="/user/{{ user_info.username }}">
          
        </a>
      </figure>
      <div class="media-content">
        <div class="field">
          <p class="control">
            <input id="input-post" class="input is-rounded" placeholder="무슨 생각을 하고 계신가요?"
              onclick='$("#modal-post").addClass("is-active")'>
          </p>
        </div>
      </div>
    </article>
    <div class="modal" id="modal-post">
      <div class="modal-background" onclick='$("#modal-post").removeClass("is-active")'></div>
      <div class="modal-content">
        <div class="box">
          <article class="media">

```

```

        <div class="media-content">
            <div class="field">
                <p class="control">
                    <textarea id="textarea-post" class="textarea"
                        placeholder="무슨 생각을 하고 계신가요?"></textarea>
                </p>
            </div>
            <nav class="level is-mobile">
                <div class="level-left">

                    </div>
                <div class="level-right">
                    <div class="level-item">
                        <a class="button is-sparta" onclick="post()">포스팅하기</a>
                    </div>
                    <div class="level-item">
                        <a class="button is-sparta is-outlined"
                            onclick='$("#modal-post").removeClass("is-active")'>취소</a>
                    </div>
                </div>
            </nav>
        </div>
    </article>
</div>
<div>
    <button class="modal-close is-large" aria-label="close"
        onclick='$("#modal-post").removeClass("is-active")'></button>
</div>

</section>
{% endif %}
<section class="section">
    <div id="post-box" class="container">

    </div>
</section>

</body>
</html>

```

▼ mystyle.css

```

body {
    background-color: RGBA(232, 52, 78, 0.2);
    min-height: 100vh;
}

.section {
    padding: 1rem 1.5rem;
    max-width: 750px;
    margin: auto;
}

.is-sparta {
    color: #e8344e !important;
}

.button.is-sparta {
    background-color: #e8344e;
    border-color: transparent;
    color: #fff !important;
}

.button.is-sparta.is-outlined {
    background-color: transparent;
    border-color: #e8344e;
    color: #e8344e !important;
}

.modal-content {
    width: 600px;
    max-width: 80%;
}

input::-webkit-calendar-picker-indicator {
    display: none;
}

.image img {
    object-fit: cover;
    width: 100%;
    height: 100%;
}

```

▼ myjs.js

```
function post() {
    let comment = $("#textarea-post").val()
    let today = new Date().toISOString()
    $.ajax({
        type: "POST",
        url: "/posting",
        data: {
            comment_give: comment,
            date_give: today
        },
        success: function (response) {
            $("#modal-post").removeClass("is-active")
            window.location.reload()
        }
    })
}

function get_posts(username) {
    if (username == undefined) {
        username = ""
    }
    $("#post-box").empty()
    $.ajax({
        type: "GET",
        url: `/get_posts?username_give=${username}`,
        data: {},
        success: function (response) {
            if (response["result"] == "success") {
                let posts = response["posts"]
                for (let i = 0; i < posts.length; i++) {
                    let post = posts[i]
                    let time_post = new Date(post["date"])
                    let time_before = time2str(time_post)

                    let class_heart = post['heart_by_me'] ? "fa-heart": "fa-heart-o"

                    let html_temp = `<div class="box" id="${post["_id"]}">
                        <article class="media">
                            <div class="media-left">
                                <a class="image is-64x64" href="/user/${post['username']}">
                                    
                                </a>
                            </div>
                            <div class="media-content">
                                <div class="content">
                                    <p>
                                        <strong>${post['profile_name']}</strong> <small>@${post['username']}</small> <br>
                                        ${post['comment']}
                                    </p>
                                </div>
                                <nav class="level is-mobile">
                                    <div class="level-left">
                                        <a class="level-item is-sparta" aria-label="heart" onclick="toggle_li"
                                            <span class="icon is-small"><i class="fa ${class_heart}"
                                                aria-hidden="true"></i></span>&nbs
                                        </a>
                                    </div>
                                </nav>
                            </div>
                        </article>
                    </div>`
                    $("#post-box").append(html_temp)
                }
            }
        }
    })
}

function time2str(date) {
    let today = new Date()
    let time = (today - date) / 1000 / 60 // 분

    if (time < 60) {
        return parseInt(time) + "분 전"
    }
    time = time / 60 // 시간
    if (time < 24) {
        return parseInt(time) + "시간 전"
    }
    time = time / 24
}
```

```

    if (time < 7) {
        return parseInt(time) + "일 전"
    }
    return `${date.getFullYear()}년 ${date.getMonth() + 1}월 ${date.getDate()}일`
}

function num2str(count) {
    if (count > 10000) {
        return parseInt(count / 1000) + "k"
    }
    if (count > 500) {
        return parseInt(count / 100) / 10 + "k"
    }
    if (count == 0) {
        return ""
    }
    return count
}

function toggle_like(post_id, type) {
    console.log(post_id, type)
    let $a_like = $('#${post_id} a[aria-label='${type}']')
    let $i_like = $a_like.find("i")
    let class_s = {"heart": "fa-heart", "star": "fa-star", "like": "fa-thumbs-up"}
    let class_o = {"heart": "fa-heart-o", "star": "fa-star-o", "like": "fa-thumbs-o-up"}
    if ($i_like.hasClass(class_s[type])) {
        $.ajax({
            type: "POST",
            url: "/update_like",
            data: {
                post_id_give: post_id,
                type_give: type,
                action_give: "unlike"
            },
            success: function (response) {
                console.log("unlike")
                $i_like.addClass(class_o[type]).removeClass(class_s[type])
                $a_like.find("span.like-num").text(num2str(response["count"]))
            }
        })
    } else {
        $.ajax({
            type: "POST",
            url: "/update_like",
            data: {
                post_id_give: post_id,
                type_give: type,
                action_give: "like"
            },
            success: function (response) {
                console.log("like")
                $i_like.addClass(class_s[type]).removeClass(class_o[type])
                $a_like.find("span.like-num").text(num2str(response["count"]))
            }
        })
    }
}
}

```

16. 4주차 끝 & 숙제 설명

하트♥ 뿐만 아니라 별★, 좋아요👍도 달 수 있게 만들어볼까요?

- 각 포스팅 카드에 더 다양한 반응을 남길 수 있도록 해봅시다!



김서진 @sfelixkim 3일 전
사이트 만들기 너무 어려워 ㅠㅠ

♡ 1 ★ 2 👍 1

▼ 힌트

- app.py에서 포스트 정보를 보낼 때 각 포스트마다 하트, 별, 좋아요가 몇 개인지, 내가 누른 것은 있는지 세 주어야 합니다.
- myjs.js 파일에서 `get_posts()` 로 각 카드를 만들 때 이 정보를 이용하여 카드를 만들어줍니다.

아이콘 클래스 정보 : fa-heart-o, fa-heart, fa-star, fa-star-o, fa-thumbs-up, fa-thumbs-o-up

- 좋아요 수 업데이트를 할 때 `toggle_like()` 에 반응 유형에 따라 값을 잘 저장하고 바꿔주어야겠죠!

17. 4주차 숙제 답안 코드

▼ [코드스니펫] - 4주차 숙제 답안 코드

전체 코드

(app.py랑 myjs.js만 바꿈)

▼ app.py

```
from pymongo import MongoClient
import jwt
import datetime
import hashlib
from flask import Flask, render_template, jsonify, request, redirect, url_for
from werkzeug.utils import secure_filename
from datetime import datetime, timedelta

app = Flask(__name__)
app.config["TEMPLATES_AUTO_RELOAD"] = True
app.config['UPLOAD_FOLDER'] = './static/profile_pics'

SECRET_KEY = 'SPARTA'

client = MongoClient('내AWS아이피', 27017, username="아이디", password="비밀번호")
db = client.dbsparta_plus_week4

@app.route('/')
def home():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        user_info = db.users.find_one({"username": payload["id"]})
        return render_template('index.html', user_info=user_info)
    except jwt.ExpiredSignatureError:
        return redirect(url_for("login", msg="로그인 시간이 만료되었습니다."))
    except jwt.exceptions.DecodeError:
        return redirect(url_for("login", msg="로그인 정보가 존재하지 않습니다."))

@app.route('/login')
def login():
    msg = request.args.get("msg")
    return render_template('login.html', msg=msg)

@app.route('/user/<username>')
def user(username):
    # 각 사용자의 프로필과 글을 모아볼 수 있는 공간
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        status = (username == payload["id"]) # 내 프로필이면 True, 다른 사람 프로필 페이지면 False

        user_info = db.users.find_one({"username": username}, {"_id": False})
        return render_template('user.html', user_info=user_info, status=status)
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

@app.route('/sign_in', methods=['POST'])
def sign_in():
    # 로그인
    username_receive = request.form['username_give']
    password_receive = request.form['password_give']

    pw_hash = hashlib.sha256(password_receive.encode('utf-8')).hexdigest()
```



```

result = db.users.find_one({'username': username_receive, 'password': pw_hash})

if result is not None:
    payload = {
        'id': username_receive,
        'exp': datetime.utcnow() + timedelta(seconds=60 * 60 * 24) # 로그인 24시간 유지
    }
    token = jwt.encode(payload, SECRET_KEY, algorithm='HS256').decode('utf-8')

    return jsonify({'result': 'success', 'token': token})
# 찾기 못하면
else:
    return jsonify({'result': 'fail', 'msg': '아이디/비밀번호가 일치하지 않습니다.'})

@app.route('/sign_up/save', methods=['POST'])
def sign_up():
    username_receive = request.form['username_give']
    password_receive = request.form['password_give']
    password_hash = hashlib.sha256(password_receive.encode('utf-8')).hexdigest()
    doc = {
        "username": username_receive,
        "password": password_hash,
        "profile_name": username_receive,
        "profile_pic": "",
        "profile_pic_real": "profile_pics/profile_placeholder.png",
        "profile_info": ""
    }
    db.users.insert_one(doc)
    return jsonify({'result': 'success'})

@app.route('/sign_up/check_dup', methods=['POST'])
def check_dup():
    username_receive = request.form['username_give']
    exists = bool(db.users.find_one({'username': username_receive}))
    # print(value_receive, type_receive, exists)
    return jsonify({'result': 'success', 'exists': exists})

@app.route('/update_profile', methods=['POST'])
def save_img():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        username = payload["id"]
        name_receive = request.form["name_give"]
        about_receive = request.form["about_give"]
        new_doc = {
            "profile_name": name_receive,
            "profile_info": about_receive
        }
        if 'file_give' in request.files:
            file = request.files["file_give"]
            filename = secure_filename(file.filename)
            extension = filename.split(".")[1]
            file_path = f"profile_pics/{username}.{extension}"
            file.save("./static/"+file_path)
            new_doc["profile_pic"] = filename
            new_doc["profile_pic_real"] = file_path
        db.users.update_one({'username': payload['id']}, {'$set': new_doc})
        return jsonify({"result": "success", 'msg': '프로필을 업데이트했습니다.'})
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

@app.route('/posting', methods=['POST'])
def posting():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        # 포스팅하기
        user_info = db.users.find_one({'username': payload["id"]})
        comment_receive = request.form["comment_give"]
        date_receive = request.form["date_give"]
        print(type(date_receive))
        doc = {
            "username": user_info["username"],
            "profile_name": user_info["profile_name"],
            "profile_pic_real": user_info["profile_pic_real"],
            "comment": comment_receive,
            "date": date_receive
        }
        db.posts.insert_one(doc)
        return jsonify({"result": "success", 'msg': '포스팅 성공'})
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):

```

```

        return redirect(url_for("home"))

@app.route("/get_posts", methods=['GET'])
def get_posts():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        my_username = payload["id"]
        username_receive = request.args.get("username_give")
        if username_receive=="":
            posts = list(db.posts.find({}).sort("date", -1).limit(20))
        else:
            posts = list(db.posts.find({"username":username_receive}).sort("date", -1).limit(20))

        for post in posts:
            post["_id"] = str(post["_id"])

            post["count_heart"] = db.likes.count_documents({"post_id": post["_id"], "type": "heart"})
            post["heart_by_me"] = bool(db.likes.find_one({"post_id": post["_id"], "type": "heart", "username": my_username}))

            post["count_star"] = db.likes.count_documents({"post_id": post["_id"], "type": "star"})
            post["star_by_me"] = bool(db.likes.find_one({"post_id": post["_id"], "type": "star", "username": my_username}))

            post["count_like"] = db.likes.count_documents({"post_id": post["_id"], "type": "like"})
            post["like_by_me"] = bool(db.likes.find_one({"post_id": post["_id"], "type": "like", "username": my_username}))

        return jsonify({"result": "success", "msg": "포스팅을 가져왔습니다.", "posts": posts})
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

@app.route('/update_like', methods=['POST'])
def update_like():
    token_receive = request.cookies.get('mytoken')
    try:
        payload = jwt.decode(token_receive, SECRET_KEY, algorithms=['HS256'])
        # 좋아요 수 변경
        user_info = db.users.find_one({"username": payload["id"]})
        post_id_receive = request.form["post_id_give"]
        type_receive = request.form["type_give"]
        action_receive = request.form["action_give"]
        doc = {
            "post_id": post_id_receive,
            "username": user_info["username"],
            "type": type_receive
        }
        if action_receive == "like":
            db.likes.insert_one(doc)
        else:
            db.likes.delete_one(doc)
        count = db.likes.count_documents({"post_id": post_id_receive, "type": type_receive})
        print(count)
        return jsonify({"result": "success", 'msg': 'updated', "count": count})
    except (jwt.ExpiredSignatureError, jwt.exceptions.DecodeError):
        return redirect(url_for("home"))

if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)

```

▼ login.html

```

<!doctype html>
<html lang="en">
  <head>

    <!-- Webpage Title -->
    <title>Log In | SWEETER</title>
    <link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
    <link rel="icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
    <meta property="og:title" content="스위터 - 세상을 달달하게"/>
    <meta property="og:description" content="mini project for Web Plus"/>
    <meta property="og:image" content="{{ url_for('static', filename='ogimg.png') }}" />
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bulma CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
    <!-- Font Awesome CSS -->
    <link href="//maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

    <link rel="preconnect" href="https://fonts.gstatic.com">

```

```

<link href="https://fonts.googleapis.com/css2?family=Gamja+Flower&family=Stylish&display=swap" rel="stylesheet">
<link href="{ url_for('static', filename='mystyle.css') }}" rel="stylesheet">

<!-- JS -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.js"></script>

<style>

    .title {
        font-weight: 800;
        font-size: 5rem;
        font-family: 'Stylish', sans-serif;
    }

    .subtitle {
        font-family: 'Gamja Flower', cursive;
        font-size: 2rem;
    }

    .help {
        color: gray;
    }
</style>
<script>
    // {% if msg %}
    //     alert("{ msg }")
    // {% endif %}

    function toggle_sign_up() {
        $("#sign-up-box").toggleClass("is-hidden")
        $("#div-sign-in-or-up").toggleClass("is-hidden")
        $("#btn-check-dup").toggleClass("is-hidden")
        $("#help-id").toggleClass("is-hidden")
        $("#help-password").toggleClass("is-hidden")
        $("#help-password2").toggleClass("is-hidden")
    }

    function is_nickname(asValue) {
        var regexp = /^[^?=.*[a-zA-Z]][-a-zA-Z0-9_]{2,10}$/;
        return regexp.test(asValue);
    }

    function is_password(asValue) {
        var regexp = /^[^?=.*\d](?=.*[a-zA-Z])[0-9a-zA-Z!@#%&*]{8,20}$/;
        return regexp.test(asValue);
    }

    function check_dup() {
        let username = $("#input-username").val()
        console.log(username)
        if (username == "") {
            $("#help-id").text("아이디를 입력해주세요.").removeClass("is-safe").addClass("is-danger")
            $("#input-username").focus()
            return;
        }
        if (!is_nickname(username)) {
            $("#help-id").text("아이디의 형식을 확인해주세요. 영문과 숫자, 일부 특수문자(._-) 사용 가능. 2-10자 길이").removeClass("is-safe").addClass("is-danger")
            $("#input-username").focus()
            return;
        }
        $("#help-id").addClass("is-loading")
        $.ajax({
            type: "POST",
            url: "/sign_up/check_dup",
            data: {
                username_give: username
            },
            success: function (response) {

                if (response["exists"]) {
                    $("#help-id").text("이미 존재하는 아이디입니다.").removeClass("is-safe").addClass("is-danger")
                    $("#input-username").focus()
                } else {
                    $("#help-id").text("사용할 수 있는 아이디입니다.").removeClass("is-danger").addClass("is-success")
                }
                $("#help-id").removeClass("is-loading")
            }
        });
    }

    function sign_up() {
        let username = $("#input-username").val()

```

```

let password = $("#input-password").val()
let password2 = $("#input-password2").val()
console.log(username, password, password2)

if ($("#help-id").hasClass("is-danger")) {
    alert("아이디를 다시 확인해주세요.")
    return;
} else if (!$("#help-id").hasClass("is-success")) {
    alert("아이디 중복확인을 해주세요.")
    return;
}

if (password == "") {
    $("#help-password").text("비밀번호를 입력해주세요.").removeClass("is-safe").addClass("is-danger")
    $("#input-password").focus()
    return;
} else if (!is_password(password)) {
    $("#help-password").text("비밀번호의 형식을 확인해주세요. 영문과 숫자 필수 포함, 특수문자(!@#%&* ) 사용가능 8-20자").rem
    $("#input-password").focus()
    return
} else {
    $("#help-password").text("사용할 수 있는 비밀번호입니다.").removeClass("is-danger").addClass("is-success")
}

if (password2 == "") {
    $("#help-password2").text("비밀번호를 입력해주세요.").removeClass("is-safe").addClass("is-danger")
    $("#input-password2").focus()
    return;
} else if (password2 != password) {
    $("#help-password2").text("비밀번호가 일치하지 않습니다.").removeClass("is-safe").addClass("is-danger")
    $("#input-password2").focus()
    return;
} else {
    $("#help-password2").text("비밀번호가 일치합니다.").removeClass("is-danger").addClass("is-success")
}

$.ajax({
    type: "POST",
    url: "/sign_up/save",
    data: {
        username_give: username,
        password_give: password
    },
    success: function (response) {
        alert("회원가입을 축하드립니다!")
        window.location.replace("/login")
    }
});

}

function sign_in() {
    let username = $("#input-username").val()
    let password = $("#input-password").val()

    if (username == "") {
        $("#help-id-login").text("아이디를 입력해주세요.")
        $("#input-username").focus()
        return;
    } else {
        $("#help-id-login").text("")
    }

    if (password == "") {
        $("#help-password-login").text("비밀번호를 입력해주세요.")
        $("#input-password").focus()
        return;
    } else {
        $("#help-password-login").text("")
    }

    $.ajax({
        type: "POST",
        url: "/sign_in",
        data: {
            username_give: username,
            password_give: password
        },
        success: function (response) {
            if (response['result'] == 'success') {
                $.cookie('mytoken', response['token'], {path: '/'});
                window.location.replace("/")
            } else {
                alert(response['msg'])
            }
        }
    })
};
}

```

```

</script>

</head>
<body>
  <section class="hero is-white">
    <div class="hero-body has-text-centered" style="padding-bottom:1rem;margin:auto;">
      <h1 class="title is-sparta">SWEETER</h1>
      <h3 class="subtitle is-sparta">세상을 달달하게</h3>
    </div>
  </section>
  <section class="section">
    <div class="container">
      <div class="box" style="max-width: 480px;margin:auto">
        <article class="media">
          <div class="media-content">
            <div class="content">
              <div class="field has-addons">
                <div class="control has-icons-left" style="width:100%">
                  <input id="input-username" class="input" type="text" placeholder="아이디">
                  <span class="icon is-small is-left"><i class="fa fa-user"></i></span>
                </div>
                <div id="btn-check-dup" class="control is-hidden">
                  <button class="button is-sparta" onclick="check_dup()">중복확인</button>
                </div>
              </div>
              <p id="help-id" class="help is-hidden">아이디는 2-10자의 영문과 숫자와 일부 특수문자(._-)만 입력 가능합니다.</p>
              <p id="help-id-login" class="help is-danger"></p>
            </div>
            <div class="field">
              <div class="control has-icons-left">
                <input id="input-password" class="input" type="password" placeholder="비밀번호">
                <span class="icon is-small is-left"><i class="fa fa-lock"></i></span>
              </div>
              <p id="help-password" class="help is-hidden">영문과 숫자 조합의 8-20자의 비밀번호를 설정해주세요. 특수문자(!@#$%^&*)도 사용 가능합니다.</p>
              <p id="help-password-login" class="help is-danger"></p>
            </div>
          </div>
          <div id="div-sign-in-or-up" class="has-text-centered">
            <nav class="level is-mobile">
              <button class="level-item button is-sparta" onclick="sign_in()">로그인</button>
            </nav>
            <hr>
            <h4 class="mb-3">아직 회원이 아니라면</h4>
            <nav class="level is-mobile">
              <button class="level-item button is-sparta is-outlined" onclick="toggle_sign_up()">회원가입하기</button>
            </nav>
          </div>
          <div id="sign-up-box" class="is-hidden">
            <div class="mb-5">
              <div class="field">
                <div class="control has-icons-left" style="width:100%">
                  <input id="input-password2" class="input" type="password" placeholder="비밀번호 재입력">
                  <span class="icon is-small is-left"><i class="fa fa-lock"></i></span>
                </div>
                <p id="help-password2" class="help is-hidden">비밀번호를 다시 한 번 입력해주세요.</p>
              </div>
            </div>
            <nav class="level is-mobile">
              <button class="level-item button is-sparta" onclick="sign_up()">회원가입</button>
              <button class="level-item button is-sparta is-outlined" onclick="toggle_sign_up()">취소</button>
            </nav>
          </div>
        </article>
      </div>
    </div>
  </div>

```

```

        </div>
    </section>

    </body>
</html>

```

▼ index.html

```

<!doctype html>
<html lang="en">
  <head>

    <!-- Webpage Title -->
    <title>Home | SWEETER</title>
    <link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
    <link rel="icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
    <meta property="og:title" content="스위터 - 세상을 달달하게"/>
    <meta property="og:description" content="mini project for Web Plus"/>
    <meta property="og:image" content="{{ url_for('static', filename='ogimg.png') }}" />
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bulma CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
    <!-- Font Awesome CSS -->
    <link href="//maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

    <link rel="preconnect" href="https://fonts.gstatic.com">
    <link href="https://fonts.googleapis.com/css2?family=Gamja+Flower&family=Stylish&display=swap" rel="stylesheet">
    <link href="{{ url_for('static', filename='mystyle.css') }}" rel="stylesheet">

    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.js"></script>
    <script src="{{ url_for('static', filename='myjs.js') }}"></script>

    <script>
      $(document).ready(function () {
        get_posts()
      })
    </script>

  </head>
  <body class="has-navbar-fixed-top">
    <nav class="navbar is-fixed-top is-white" role="navigation" aria-label="main navigation">
      <div class="navbar-brand">
        <a class="navbar-item" href="/">
          
          <strong class="is-sparta" style="font-family: 'Stylish', sans-serif;font-size: larger;">SWEETER</strong>
        </a>
      </div>
    </nav>
    <section class="section">
      <article class="media">
        <figure class="media-left" style="align-self: center">
          <a class="image is-32x32" href="/user/{{ user_info.username }}">
            
          </a>
        </figure>
        <div class="media-content">
          <div class="field">
            <p class="control">
              <input id="input-post" class="input is-rounded" placeholder="무슨 생각을 하고 계신가요?"
                onclick="$("#modal-post").addClass("is-active")'>
            </p>
          </div>
        </div>
      </article>
      <div class="modal" id="modal-post">
        <div class="modal-background" onclick="$("#modal-post").removeClass("is-active")'></div>
        <div class="modal-content">
          <div class="box">
            <article class="media">
              <div class="media-content">
                <div class="field">
                  <p class="control">
                    <textarea id="textarea-post" class="textarea"
                      placeholder="무슨 생각을 하고 계신가요?"></textarea>
                  </p>

```

```

        </div>
        <nav class="level is-mobile">
          <div class="level-left">

          </div>
          <div class="level-right">
            <div class="level-item">
              <a class="button is-sparta" onclick="post()">포스팅하기</a>
            </div>
            <div class="level-item">
              <a class="button is-sparta is-outlined"
                onclick='{$("#modal-post").removeClass("is-active")}'>취소</a>
            </div>
          </div>
        </nav>
      </div>
    </article>
  </div>
  <div>
    <button class="modal-close is-large" aria-label="close"
      onclick='{$("#modal-post").removeClass("is-active")}'></button>
  </div>
</section>
<section class="section">
  <div id="post-box" class="container">
    <div class="box">
      <article class="media">
        <div class="media-left">
          <a class="image is-64x64" href="#">
            <img class="is-rounded"
              src={{ url_for("static", filename="profile_pics/profile_placeholder.png") }} alt="Image":
          </a>
        </div>
        <div class="media-content">
          <div class="content">
            <p>
              <strong>홍길동</strong> <small>@username</small> <small>10분 전</small>
              <br>
              글을 적는 칸
            </p>
          </div>
          <nav class="level is-mobile">
            <div class="level-left">
              <a class="level-item is-sparta" aria-label="heart"
                onclick="toggle_like(' ', 'heart')">
                <span class="icon is-small"><i class="fa fa-heart"
                  aria-hidden="true"></i></span>&nbsp;<span
                  class="like-num">2.7k</span>
              </a>
            </div>
          </nav>
        </div>
      </article>
    </div>
  </section>
</body>
</html>

```

▼ user.html

```

<!doctype html>
<html lang="en">
  <head>

    <!-- Webpage Title -->
    <title>{{ user_info.profile_name }} | SWEETER</title>
    <link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
    <link rel="icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
    <meta property="og:title" content="스위터 - 세상을 달달하게"/>
    <meta property="og:description" content="mini project for Web Plus"/>
    <meta property="og:image" content="{{ url_for('static', filename='ogimg.png') }}" />
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bulma CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bulma@0.9.1/css/bulma.min.css">
    <!-- Font Awesome CSS -->
    <link href="//maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

```

```

<link rel="preconnect" href="https://fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css2?family=Gamja+Flower&family=Stylish&display=swap" rel="stylesheet">
<link href="{{ url_for('static', filename='mystyle.css') }}" rel="stylesheet">

<!-- JS -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-cookie/1.4.1/jquery.cookie.js"></script>
<script src="{{ url_for('static', filename='myjs.js') }}"></script>

<script>
    $(document).ready(function () {
        get_posts("{{ user_info.username }}")
    })

    function update_profile() {
        let name = $('#input-name').val()
        let file = $('#input-pic')[0].files[0]
        let about = $("#textarea-about").val()
        let form_data = new FormData()
        form_data.append("file_give", file)
        form_data.append("name_give", name)
        form_data.append("about_give", about)
        console.log(name, file, about, form_data)

        $.ajax({
            type: "POST",
            url: "/update_profile",
            data: form_data,
            cache: false,
            contentType: false,
            processData: false,
            success: function (response) {
                if (response["result"] == "success") {
                    alert(response["msg"])
                    window.location.reload()
                }
            }
        });
    }

    function sign_out() {
        $.removeCookie('mytoken', {path: '/'});
        alert('로그아웃!')
        window.location.href = "/login"
    }
</script>

</head>
<body class="has-navbar-fixed-top">
<nav class="navbar is-fixed-top is-white" role="navigation" aria-label="main navigation">
    <div class="navbar-brand">
        <a class="navbar-item" href="/">
            
            <strong class="is-sparta" style="font-family: 'Stylish', sans-serif;font-size: larger;">SWEETER</strong>
        </a>
    </div>
</nav>

<section class="hero is-white">
    <div class="hero-body" style="padding-bottom:1rem;margin:auto;min-width: 400px">
        <article class="media">

            <figure class="media-left" style="align-self: center">
                <a class="image is-96x96" href="#">
                    
                </a>
            </figure>
            <div class="media-content">
                <div class="content">
                    <p>
                        <strong>{{ user_info.profile_name }}</strong> <small>@{{ user_info.username }}</small>
                        <br>
                        {{ user_info.profile_info }}
                    </p>
                </div>
            </div>
        </article>
        {% if status %}
        <nav id="btns-me" class="level is-mobile" style="margin-top:2rem">
            <a class="button level-item has-text-centered is-sparta" aria-label="edit">

```



```

        <div class="media-content">
            <div class="field">
                <p class="control">
                    <input id="input-post" class="input is-rounded" placeholder="무슨 생각을 하고 계신가요?"
                        onclick='$("#modal-post").addClass("is-active")'>
                </p>
            </div>
        </div>
    </article>
    <div class="modal" id="modal-post">
        <div class="modal-background" onclick='$("#modal-post").removeClass("is-active")'></div>
        <div class="modal-content">
            <div class="box">
                <article class="media">
                    <div class="media-content">
                        <div class="field">
                            <p class="control">
                                <textarea id="textarea-post" class="textarea"
                                    placeholder="무슨 생각을 하고 계신가요?"></textarea>
                            </p>
                        </div>
                    </div>
                    <nav class="level is-mobile">
                        <div class="level-left">
                            </div>
                        <div class="level-right">
                            <div class="level-item">
                                <a class="button is-spata" onclick="post()">포스팅하기</a>
                            </div>
                            <div class="level-item">
                                <a class="button is-spata is-outlined"
                                    onclick='$("#modal-post").removeClass("is-active")'>취소</a>
                            </div>
                        </div>
                    </nav>
                </div>
            </div>
            </div>
            <div>
                <button class="modal-close is-large" aria-label="close"
                    onclick='$("#modal-post").removeClass("is-active")'></button>
            </div>
        </div>
    </section>
    {% endif %}
    <section class="section">
        <div id="post-box" class="container">
            </div>
        </section>
    </body>
</html>

```

▼ mystyle.css

```

body {
    background-color: RGBA(232, 52, 78, 0.2);
    min-height: 100vh;
}

.section {
    padding: 1rem 1.5rem;
    max-width: 750px;
    margin: auto;
}

.is-spata {
    color: #e8344e !important;
}

.button.is-spata {
    background-color: #e8344e;
    border-color: transparent;
    color: #fff !important;
}

.button.is-spata.is-outlined {
    background-color: transparent;
    border-color: #e8344e;
    color: #e8344e !important;
}

.modal-content {

```

```

        width: 600px;
        max-width: 80%;
    }

    input::-webkit-calendar-picker-indicator {
        display: none;
    }

    .image img {
        object-fit: cover;
        width: 100%;
        height: 100%;
    }

```

▼ myjs.js

```

function post() {
    let comment = $("#textarea-post").val()
    let today = new Date().toISOString()
    $.ajax({
        type: "POST",
        url: "/posting",
        data: {
            comment_give: comment,
            date_give: today
        },
        success: function (response) {
            $("#modal-post").removeClass("is-active")
            window.location.reload()
        }
    })
}

function get_posts(username) {
    if (username == undefined) {
        username = ""
    }
    $("#post-box").empty()
    $.ajax({
        type: "GET",
        url: `/get_posts?username_give=${username}`,
        data: {},
        success: function (response) {
            if (response["result"] == "success") {
                let posts = response["posts"]
                for (let i = 0; i < posts.length; i++) {
                    let post = posts[i]
                    let time_post = new Date(post["date"])
                    let time_before = time2str(time_post)

                    let class_heart = post['heart_by_me'] ? "fa-heart": "fa-heart-o"
                    let class_star = post['star_by_me'] ? "fa-star": "fa-star-o"
                    let class_like = post['like_by_me'] ? "fa-thumbs-up" : "fa-thumbs-o-up"

                    let html_temp = `

<article class="media">
                            <div class="media-left">
                                <a class="image is-64x64" href="/user/${post['username']}">
                                    
                                </a>
                            </div>
                            <div class="media-content">
                                <div class="content">
                                    <p>
                                        <strong>${post['profile_name']}</strong> <small>@${post['username']}</small>
                                        <br>
                                        ${post['comment']}
                                    </p>
                                </div>
                                <nav class="level is-mobile">
                                    <div class="level-left">
                                        <a class="level-item is-sparta" aria-label="heart" onclick="toggle_li
                                            <span class="icon is-small"><i class="fa ${class_heart}"
                                                aria-hidden="true"></i></span>&nbsp;
                                        </a>
                                        <a class="level-item is-sparta" aria-label="star" onclick="toggle_li
                                            <span class="icon is-small"><i class="fa ${class_star}"
                                                aria-hidden="true"></i></span>&nbsp;
                                        </a>
                                        <a class="level-item is-sparta" aria-label="like" onclick="toggle_li
                                            <span class="icon is-small"><i class="fa ${class_like}"
                                                aria-hidden="true"></i></span>&nbsp;
                                        </a>
                                    </div>
                                </nav>
                            </div>
                        </article>
                    `
                }
            }
        }
    })
}


```

```

        </div>

        </nav>
    </div>
</article>
</div>`
    $("#post-box").append(html_temp)
}
}
})
}

function time2str(date) {
    let today = new Date()
    let time = (today - date) / 1000 / 60 // 분

    if (time < 60) {
        return parseInt(time) + "분 전"
    }
    time = time / 60 // 시간
    if (time < 24) {
        return parseInt(time) + "시간 전"
    }
    time = time / 24
    if (time < 7) {
        return parseInt(time) + "일 전"
    }
    return `${date.getFullYear()}년 ${date.getMonth() + 1}월 ${date.getDate()}일`
}

function num2str(count) {
    if (count > 10000) {
        return parseInt(count / 1000) + "k"
    }
    if (count > 500) {
        return parseInt(count / 100) / 10 + "k"
    }
    if (count == 0) {
        return ""
    }
    return count
}

function toggle_like(post_id, type) {
    console.log(post_id, type)
    let $a_like = $('#${post_id} a[aria-label='${type}']')
    let $i_like = $a_like.find("i")
    let class_s = {"heart": "fa-heart", "star": "fa-star", "like": "fa-thumbs-up"}
    let class_o = {"heart": "fa-heart-o", "star": "fa-star-o", "like": "fa-thumbs-o-up"}
    if ($i_like.hasClass(class_s[type])) {
        $.ajax({
            type: "POST",
            url: "/update_like",
            data: {
                post_id_give: post_id,
                type_give: type,
                action_give: "unlike"
            },
            success: function (response) {
                console.log("unlike")
                $i_like.addClass(class_o[type]).removeClass(class_s[type])
                $a_like.find("span.like-num").text(num2str(response["count"]))
            }
        })
    } else {
        $.ajax({
            type: "POST",
            url: "/update_like",
            data: {
                post_id_give: post_id,
                type_give: type,
                action_give: "like"
            },
            success: function (response) {
                console.log("like")
                $i_like.addClass(class_s[type]).removeClass(class_o[type])
                $a_like.find("span.like-num").text(num2str(response["count"]))
            }
        })
    }
}
}

```

