

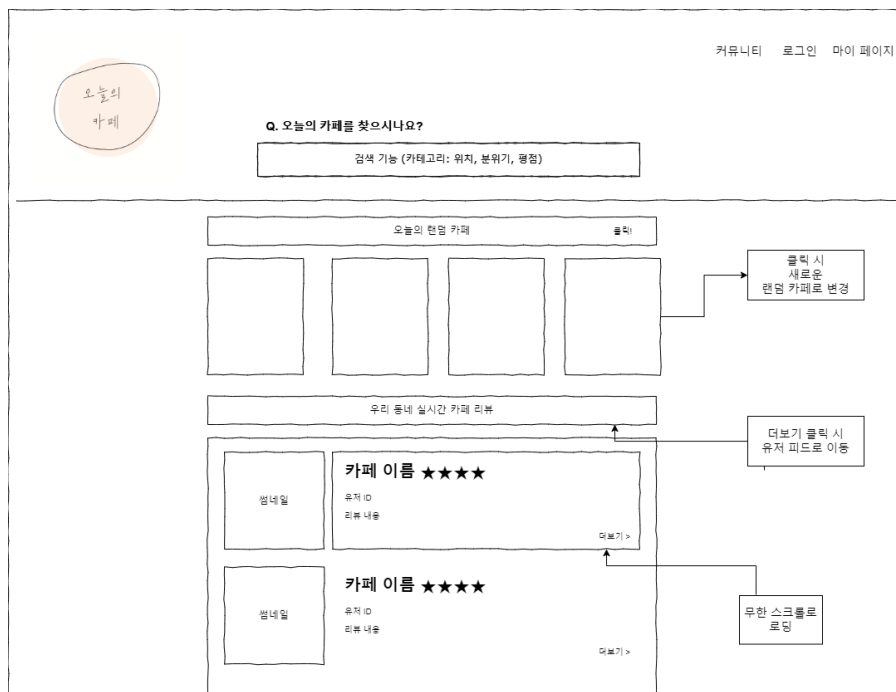
# 내일배움캠프 Node.js A반 10조 - 오늘의 카페 프로젝트 최종 결과물

## ▼ 프로젝트 개요

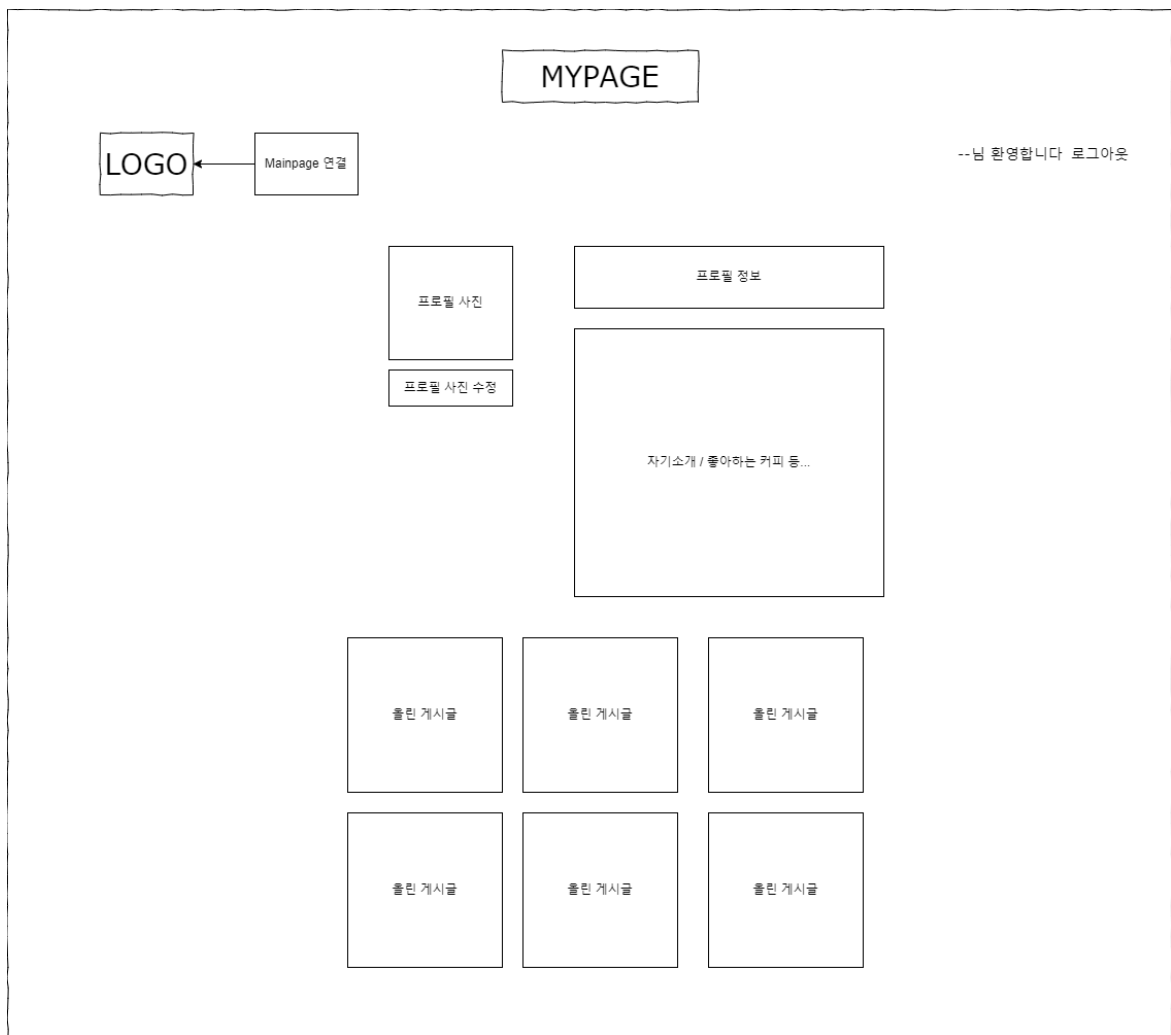
1. 프로젝트 명: 오늘의 카페
2. 프로젝트 목적 및 기능
  - 카페를 좋아하는 사람들을 위한 오늘의 카페를 추천해주는 사이트 및 커뮤니티
4. 총 개발 인원: 5명 (노드 팀조(node 10조): 조해빈(팀장), 김택환, 우태현, 장동규, 황민주)
5. 개발 기간: 2022.12.02~2022.12.08 (총 7일)

## ▼ 와이어프레임

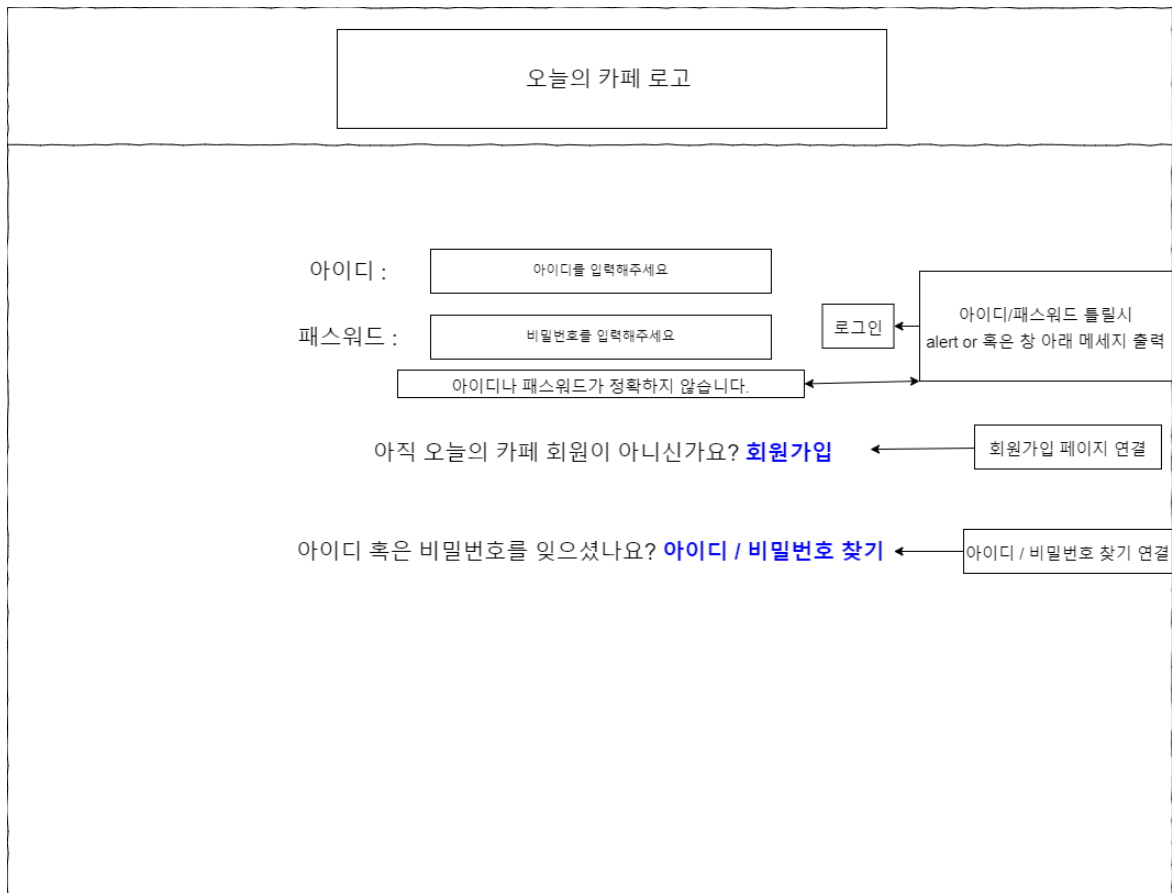
- 메인 페이지



- 마이 페이지



- 로그인 페이지



- 회원가입 페이지

# 회원가입

\* 필수

\* 이름

이름

\* 아이디

아이디

중복 확인

\* 비밀번호

비밀번호

\* 비밀번호 확인

비밀번호 확인

일치 / 틀릴지

\* 이메일

이메일

인증

연락처

연락처

좋아하는 원두 종류

좋아하는 원두 종류

mbti

mbti

가입하기

alert: 가입이 완료 되었습니다.  
 메인페이지로 이동

- 게시판 목록

# Daily Cafe

information only they know

원하시는 정보를 검색해보세요.

MENU

Question

번호	제목	글쓴이	날 짜	조 회
3	1월에 분위기 좋은 카페 있나요?	글쓴이	12. 2	44
2	윈도를 추천받는 이유에 대해 알고싶어요	글쓴이	11. 20	21
1	게시글의 제목 표시	글쓴이	11. 14	12

<< < 1 > >>

수정하기 글쓰기

## • 게시물 수정

"프로필 이미지"

"회원정보"

"제목을 입력해주세요"

"내용을 입력해주세요"

SUBMIT

"최초 글쓰기 버튼 click 시"

"프로필 이미지"

"회원정보"

"제목을 입력해주세요"

"내용을 입력해주세요"

MODIFY

"수정 버튼 click 시"

### ▼ 역할 분담

- 조해빈: 마이 페이지
- 우태현: 메인 페이지
- 황민주: 로그인
- 김택환: 회원가입
- 장동규: 게시판

♥다들 서로 도우며...합시다♥

### ▼ 글꼴 및 색상 (색상 코드: #FDF0E6 #EDDBC9 #595151 #362E30 #F4E5D)

```
/*카페24 고운밤*/
@font-face {
  font-family: 'Cafe24Oneprettypnight';
  src: url('https://cdn.jsdelivr.net/gh/projectnoonnu/noonfonts_twelve@1.1/Cafe24Oneprettypnight.woff') format('woff');
  font-weight: normal;
  font-style: normal;
}
```

### ▼ API 명세서

기능	Method	URL	Request type	Response
날씨 불러오기	GET	api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={API key}	-	{ "coord":{"lon":12 {"temp":283.23,"fe {"type":1,"id":8105
카페 지도 불러오기	GET	dapi.kakao.com/v2/maps/sdk.js?appkey=APIKEY&libraries=services,clusterer,drawing	-	window.kakao=wi {},window.daum.r 8",e.onload=t,e.or {for(;c[0];)c.shift() {ROADMAP:"220! SKYVIEW_VERS {ROADVIEW_AJ/ n,r="https:"=loca c=o.onloadcallbac drawing:r+"//t1.da ([^&]*)/gi,function( {a&&document.wr
책 검색 API		<a href="https://dapi.kakao.com/v3/search/book?target=title?Authorization: config.bookapikey">https://dapi.kakao.com/v3/search/book?target=title?</a> Authorization: config.bookapikey		{ "documents": [{ "ai 하는 독자를 위해 가 컴퓨터 내부에서 어 매", "thumbnail": "h Dive", "translators'
회원가입	POST	/post/users	{ "user": "user", "pw": "pw", "name": "name", "email": "email", "desc": "desc" }	success: function
마이 페이지 수정	POST	/post/mypages	{ "name": "name", "id": "id", "desc": "desc" }	{ 'msg': '수정 완료' }
마이 페이지 불러오기	GET	/mypages	-	{ 'msg': 'user_list' }
게시판 목록 조회	GET	/post	{ "name": "name", "pass": "pass", "title": "title" "content": "content" }	
게시물 조회	GET	/post/content	{ "name": "name", "pass": "pass", "title": "title" "content": "content" }	
게시물 등록	POST	/write	{ "name": "name", "pass": "pass", "title": "title" "content": "content" }	localhost:5000내

기능	Method	URL	Request type	Response
게시판 수정	PUT	/post/edit	{ "name": "name", "pass": "pass", "title": "title" "content": "content" }	localhost:5000내돈
게시판 삭제	DELETE	/post/delete	{ "name": "name", "pass": "pass", "title": "title" "content": "content" }	localhost:5000내돈

#### ▼ DB 설계

##### • users

컬럼	데이터 타입	제약 조건	설명
id	INT	PRIMARY KEY, AUTO INCREMENT	번호
email	VARCHAR(50)	UNIQUE	사용자 이메일(아이디)
password	VARCHAR(500)	NOT NULL	사용자 비밀번호
name	VARCHAR(40)	NOT NULL	사용자 이름
desc	VARCHAR(500)		사용자 자기소개
img	VARCHAR(50)		
upload_time	TIMESTAMP	NOT NULL	

##### • board

컬럼	데이터 타입	제약 조건	설명
id	INT	PRIMARY KEY, AUTO INCREMENT, NOT NULL	게시물 번호
name	VARCHAR(20)	NULL	글쓴이
title	VARCHAR(70)	NULL	게시글 제목
content	text	NULL	게시글 내용
wdate	TIMESTAMP	NULL DEFAULT CURRENT_TIMESTAMP	게시물 생성 시간
view	int	NULL DEFAULT "0"	조회수

##### • cafelist

컬럼	데이터 타입	제약 조건	설명
idnumber	INT	FOREIGN KEY, NOT NULL	번호
shopname	VARCHAR(100)		점포명
telnumber	VARCHAR(45)		점포 전화번호
address	VARCHAR(100)		(구)지번주소
newaddress	VARCHAR(100)		(신)도로명주소
changed_at	VARCHAR(45)		정보 업데이트 시간(오픈데이터에서 제공)

#### ▼ 폴더 구조

static>css	static>js	templates	app.py	app.route	담당
main.css response.css	apikey.js kakaobook.js kakaomap.js openweather.js searchdata.js timenow.js	index.html response.html	main.py	/ (메인 페이지) /searchdata (검색 결과)	태현
edit_page.css my_page.css	edit_page.js my_page.js	edit_page.html my_page.html	my_page.py	/my_page /edit_page /users/<id> (GET) /users/<id> (POST) /upload (GET, POST)	해빈

static>css	static>js	templates	app.py	app.route	담당
style.css	-	content.html delete.html edit.html editError.html editsuccess.html post.html write.html	app.py	/post /post/content/<id> /post/edit/<id>(GET,POST) /post/delete/<id> /post/delete/success/<id> /write(GET,POST)	동규
login.css	-	login.html	main.py	/login /logout	민주
signup.css	signup.js	signup.html	app.py	/signup /post/users	택환

#### ▼ 사용 기술 스택

- Front : HTML, CSS, Javascript, Bootstrap
- Back : JavaScript, Python, Flask
- Database: MySQL
- Server : Windows

#### ▼ 작업 히스토리

- 2022.12.02
  - 프로젝트 SA, 와이어프레임 작성, 역할 분담
- 2022.12.05
  - 메인 페이지: 현재 시간, 지역 날씨 불러오기, 카페 검색 결과 기능 구현 완료
  - 마이 페이지: 마이 페이지 화면 구현 완료
  - 로그인 페이지: 로그인 기능 화면 구현
  - 회원가입 페이지: 회원가입 입력 폼 화면 구현 완료
  - 게시판 페이지: 게시판 리스트 화면 구현 완료
- 2022.12.06
  - 메인 페이지: 카페 지도 불러오기 기능 구현 완료, 책 검색 기능 구현 완료
  - 마이 페이지: 회원 이름, 소개 수정 구현 완료
- 2022.12.07
  - 메인 페이지: 카페 검색 기능 페이지네이션 1차 완료
  - 마이 페이지: 이름, 소개 수정 기능 구현 완료
  - 회원가입 페이지: 비밀번호 암호화 작업 및 회원 정보 DB 저장 완료
- 2022.12.08
  - 팀원 작업물 병합 및 충돌 해결
  - 메인 페이지: 카페 검색 기능 페이지네이션 최종 완료
  - 마이 페이지: 이름, 소개 수정 구현 완료
  - 로그인: 로그인 여부에 따른 내비게이션 노출 변경 완료(세션 관리)
  - 게시판 페이지: 글 작성/수정/삭제 기능 완료, 게시글 목록 페이지네이션 추가 완료
  - logging 시스템: 구현 완료
- 2022.12.09
  - 마이 페이지: 회원 이미지 수정 완료
  - 프로젝트 종료

#### ▼ 회고

- 개발을 진행하면서 어려웠던 점과 해결한 사항



- 서버 사이드 페이지네이션 구현
- 회원 프로필 이미지 업로드 후 보여주기
- 로그인 세션 관리

## ▼ 코드 소개

### ▼ 민주

```
if request.method == 'POST':
    # 이메일, 패스워드 받아서 암호화
    email = request.form['email']
    password = request.form['password'].encode('utf-8')

    # DB에 저장된 이메일 값 조회
    cur = conn.cursor(pymysql.cursors.DictCursor)
    cur.execute('SELECT * FROM users WHERE email = %s', (email,))
    user = cur.fetchone()
    cur.close()
    print(user)
    print(password)

    # 암호화된 비밀번호와 조회한 비밀번호를 비교
    if user is not None:
        if bcrypt.hashpw(password, user['password'].encode('utf-8')) == user['password'].encode('utf-8'):
            # 세션에 이메일, id 저장
            session['email'] = user['email']
            session['id'] = user['id']
            print(session.get('email'))
            sql = "SELECT * FROM cafelist ORDER BY rand() LIMIT 4"
            with conn:
                with conn.cursor() as cur:
                    cur.execute(sql)
                    result = cur.fetchall()
                    for data in result:
                        print(data)
            return render_template('index.html', result=result)
        # 비밀번호 틀렸거나 DB가 없을 때 예외 처리
        else:
            msg = '이메일 또는 비밀번호를 확인해주세요'
            return render_template('login.html', msg=msg)
    else:
        msg = '이메일 또는 비밀번호를 확인해주세요'
        return render_template('login.html', msg=msg)

else:
    return render_template('login.html')
```

```
{% if session.get('email') is not none %}
<nav>
  <span class="spanmenu"><a href="/signup">회원가입</a></span>
  <span class="spanmenu"><a href="/logout">로그아웃</a></span>
  <span class="spanmenu"><a href="/my_page">마이 페이지</a></span>
  <span class="spanmenu"><a href="/post">커뮤니티</a></span>
</nav>
{% else %}
<nav>
  <span class="spanmenu"><a href="/signup">회원가입</a></span>
  <span class="spanmenu"><a href="/login">로그인</a></span>
  <span class="spanmenu"><a href="/my_page">마이 페이지</a></span>
  <span class="spanmenu"><a href="/post">커뮤니티</a></span>
</nav>
{% endif %}
```

### ▼ 소감

- 회원가입과 로그인 기능을 다른 사람이 작성하고, 처음에는 소통 없이 각자 개발하다 보니 나중에 싱크를 맞추는 게 어려웠습니다. 다음에는 어떤 방식으로 접근할지 처음부터 협의해서 하면 더 효율적으로 할 수 있을 것 같습니다.
- 기초가 없고 시간이 없다 보니 이미 구현되어 있는 코드를 많이 참고했는데, 공부할 때 직접 하나씩 써보는 연습, 그리고 에러를 만났을 때 디버깅하는 연습이 많이 필요하다고 느꼈습니다.
- 에러의 에러의 에러를 만나도 항상 웃음을 잃지 않고 포기하지 않고 함께 머리 싸매고 고민한 저희 팀원들, 팀장님 그리고 저희 조 일처럼 같이 고민해주신 대원분들 감사합니다. 노드쉽조 영원히 함.께.해 🍷

## ▼ 택환

### ▼ html

```
<div class="signUp" style="text-align:center">
  <button id="signUpButton" onclick="signUpCheck()">가입하기</button>
</div>
```

### ▼ script

```
$.ajax({
  url: '/post/users',
  type: 'POST',
  data: {'email':email, 'name':name, 'password':password},
  success: function () {
    window.location.href = "/login";
  }
})
```

### ▼ python

```
@app.route('/post/users', methods=['GET', 'POST'])
def user_post():
    conn = pymysql.connect(host=f'{host}',
                           user=f'{mysqluser}',
                           password=f'{pwd}',
                           db=f'{mysqlldb}',
                           charset='utf8')
    cur = conn.cursor()
    email = request.form['email']
    pass_word = request.form['password']
    b = bcrypt.hashpw(pass_word.encode('utf-8'), bcrypt.gensalt())
    name_re = request.form['name']
    sql = "INSERT INTO users(email, password, name) VALUES(%s,%s,%s)"
    with conn:
        with conn.cursor() as cur:
            cur.execute(sql, (email, b, name_re))
            conn.commit()
    return render_template('login.html')
```

개인적으로 회원가입기능을 작성했었는데 실력이 너무 부족하여서 어떻게 할까 막막했지만 구글링으로 찾아보고 구현해보고 갖가지 오류들을 만나면서 굉장히 성장 할 수 있었습니다. 서로가 서로의 코드를 보고 도움을 주는것들이 보기가 좋았고 무엇보다 팀 분위기가 너무 좋아서 협업해서 잘 해나갈 수 있었습니다.

물론.. 발표 직전까지 코드 수정하느라 너무 고생했지만 진짜 끝까지 버텨준 팀장님과 팀원들에게 너무 감사드립니다❤️

## ▼ 해빈

```
def allowed_file(filename):
    return '.' in filename and \
           filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS
@app.route("/upload", methods=['POST'])
def upload():
    db = pymysql.connect(host=f'{host}',
                        user=f'{mysqluser}',
                        password=f'{pwd}',
                        db=f'{mysqlldb}',
                        charset='utf8')
    cur = db.cursor(pymysql.cursors.DictCursor)
    # 저장 시간 저장
    now = datetime.now()
    # 이메일 세션 저장
    id = session['id']
    # method = post 이면
    if request.method == 'POST':
        # input 타입의 태그 이름을 files에 저장
        files = request.files.getlist('files[]')
        for file in files:
            # 만약 파일이 {'png', 'jpg', 'jpeg'} 이와 같은 확장자와 일치하면
            if file and allowed_file(file.filename):
                # 어려워
                filename = secure_filename(file.filename)
                #이건 더
                file.save(os.path.join(app.config['UPLOAD_FOLDER'], filename))
                # users 테이블 안에 있는 세션 저장한 email의 img, upload_time의 컬럼 값을 변경
```

```

        sql = f'UPDATE users SET img="{filename}",upload_time="{now}" WHERE id="{id}";'
        # sql 쿼리문 실행
        cur.execute(sql)
        # 커밋 및 연결 종료
        db.commit()
        db.close()

        # alert 띄워주기
        flash('사진이 성공적으로 업로드 되었어요!')
    # edit 페이지로 돌아가기
    return redirect('/edit_page')

```

```

$(document).ready(function () {
    console.log(1)
    // const id = 1;
    $.ajax({
        type: "GET",
        url: `/users`,
        data: {},
        //response 정제 밝히기
        success: function (response) {
            console.log(2)
            const rows = response["users"]
            let name = rows['name']
            let desc = rows["desc"]
            let img = rows["img"]
            let temp_html = `<div class="whole">
                                <div class="myprofile">
<!-- action = 폼을 전송할 서버 쪽 스크립트 파일을 지정 / name = 폼을 식별하기 위한 이름 지정 / method = 폼을 서버에 전송할 http 메소드를 정함 -->
<!-- class = 중복 가능 , id = 중복 불가능 name = 폼 통신을 하기 위한 장치? -->
                                <form method="post" action="/upload" enctype="multipart/form-data" class="form-inline
                                <div class="form-group">
<!-- type = 태그 모양을 변경 / name = 태그 이름 / onchange = 미리보기 함수 / placeholder = 태그 입력 값에 대한 힌트 -->
                                <input type="file" name="files[]" id="fileInput" onchange="test()" class="form-control">
<!-- 이미지 경로를 static 폴더 안의 img 폴더의 변수 선언을 한 값? -->
                                
<!-- submit 버튼을 누르면 에러메시지가 웹 브라우저에 출력 -->
                                <input type="submit" name="submit" class="btn btn-success" value="UPLOAD"/>
                                </form>
                                </div>
<!--
                                method = post , action = users의 id 컬럼 값-->
                                <form method="post" action="/users">
                                <div class="d1">
                                    <div class="a1">아이디</div>
                                    <div class="form-floating mb-3" style="flex-grow: 2; ">
                                        <input type="name" class="form-control" id="name" name="name" placeholder="${name}">
                                    </div>
                                </div>
                                <div class="d2">
                                    <div class="a2">소개</div>
                                    <div class="form-floating mb-3" style="flex-grow: 2; height: 10rem;">
                                        <input type="id" class="form-control" id="id" name="desc" placeholder="${desc}">
                                    </div>
                                </div>
                                <a href="/my_page">
                                    <button style="margin-left: 15rem"><input type="submit" value="수정하기"></button>
                                </a>
                                <a href="/my_page">
                                    버튼 타입이 버튼일 경우 무반응-->
                                    <button type="button" class="btn btn-dark" style="margin-left: 1rem">닫기</button>
                                </a>
                                </form>
                                </div>`
            $('.myinfo').append(temp_html)
        }
    })
})

```

😊 이번 프로젝트는 아는게 없이 시작한 프로젝트 이다 보니 팀원들과의 소통이 정말 중요했던 것 같습니다. 서로 모르는 것을 질문하고 본인의 기능이 아님에도 불구하고 본인의 일처럼 적극적으로 코드에 대한 서포트를 해주셔서 포기하지 않고 끝까지 할 수 있었던 것 같습니다. 저를 믿고 따라와주신 팀원 분들께도 정말 감사드리고, 다른 조원 분들이 많이 도움을 주셔서 성공할 수 있었습니다. 협력을 통해 원하는 결과를 얻을 수 있다는 좋은 깨달음을 얻은 프로젝트였습니다. 10조 최고♥



## ▼ 동규

```
게시판 목록 html

<div class="board_wrap">
  <div class="board_title">
    <strong>Daily Cafe</strong>
    <p>information only they know</p>
  </div>
  <div class="post_list">
    <table id="example" class="table table-striped table-bordered" style="width:100%">
      <thead class="top">
        <tr class="nums">
          <td class="num">번호</td>
          <td class="title">제목</td>
          <td class="wirter">글쓴이</td>
          <td class="wdate">작성일</td>
          <td class="view">조회</td>
        </tr>
      </thead>
      <tbody>
        {% for row in result %}
        <tr class="nums">
          <td class="num"> {{ row.id }}</td>
          <td class="title"> <a style="color:#110957;" href="/post/content/{{ row.id }}">{{ row.title }}</a>
          <td class="wirter"> {{ row.name }}</td>
          <td class="wdate"> {{ row.wdate }}</td>
          <td class="view"> {{ row.view }}</td>
          <td>{{row.telnumber}}</td>-->
          <td>{{row.address}}</td>-->
          <td>{{row.newaddress}}</td>-->
        </tr>
        {% endfor %}
      </tbody>
    </table>
  </div>
</div>
</div>

<script>
  $(document).ready(function () {
    $('#example').DataTable({
      "aLengthMenu": [[5, 10, 15, 20, 50, -1], [5, 10, 15, 20, 50, "All"]],
      "iDisplayLength": 5
    });
  });
</script>

python 부분

(아래는 게시판 영역이며 세션에서 정보를 확인하여 실행(목록뿐만아니라 글쓰기, 수정도 동일))

</script>@app.route('/post')
# board테이블의 게시판 제목리스트 역순으로 출력
def post():
    if 'email' in session:
        username = session['email']
    else:
        username = None

    conn = connectsql()
    cursor = conn.cursor(pymysql.cursors.DictCursor)
    query = "SELECT id, name, title, wdate, view FROM board ORDER BY id DESC" # ORDER BY 컬럼명 DESC : 역순출력, ASC : 순차출력
    cursor.execute(query)
    result = cursor.fetchall()
    # post_list = cursor.fetchall()
```

```

    cursor.close()
    conn.close()

    return render_template('post.html', result=result, logininfo=username)

```

개인적으로 많이 어렵고 막막했지만 옆에 있는 훌륭한 팀원들 덕분에 잘 따라갈수 있었습니다!!

저도 언젠가 캐리 할수있도록 실력을 키우겠습니다.

10조 우리 팀원들 앞으로도 지금처럼 포기하지말고 끝까지 열심히 합시다!!!!!!

#### ▼ 태현

```

//페이지네이션 JAVASCRIPT
//pagination-body를 클릭 event가 발생했을 경우 e를 매개변수로 받는다
document.getElementById("pagination-body").addEventListener('click', function (e) {
    //
    if (e.target.tagName !== 'A') return;
    //preventDefault 란?
    // a 태그나 submit 태그는 누르게 되면 href 를 통해 이동하거나 , 창이 새로고침하여 실행됩니다.
    // preventDefault 를 통해 이러한 동작을 막아줄 수 있습니다.
    // 주로 사용되는 경우는
    // 1. a 태그를 눌렀을때도 href 링크로 이동하지 않게 할 경우
    // 2. form 안에 submit 역할을 하는 버튼을 눌렀어도 새로 실행하지 않게 하고싶을 경우 (submit은 작동됨)
    e.preventDefault();
    if (e.target.href) {
        const getUrlParams = () => {
            var params = {};
            //패턴매칭, gi = replace gi g발생한 모든 패턴에서 전역검색 i대소문자 구분안함
            e.target.href.replace(/[?&]+(?:[^\&]+)=([^\&]*)/gi,
                function (str, key, value) {
                    params[key] = value;
                }
            );
            return params;
        }
        params = getUrlParams();
        params.search_word = decodeURI(params.search_word);
        console.log(params)
        const search_word = params.search_word;
        const page = e.target.text;

        $.ajax({
            type: "GET",
            url: `/searchdata?search_word=${search_word}&page=${page}`,
            // data: dataString,
            cache: false,
            beforeSend: function (html) {
                document.getElementById("insert_search").innerHTML = '';
                $("#flash").show();
                $("#searchword").show();
                $(".searchword").html(search_word);
                $("#flash").html(' Loading Results...');
            },
            success: function (html) {
                $("#insert_search").show();
                $("#insert_search").append(html.data);
                $("#flash").hide();
            }
        });
    }
});

```

```

//페이지네이션 HTML
<script src="{{ url_for('static', filename='js/pagination.js') }}"></script>

<div id="pagination-body">
    <table id="example" class="table table-striped table-bordered" style="width:100%">
        <thead>
            {{ pagination.info }}
        <tr>
            <td>점포명</td>
            <td>전화번호</td>
            <td>(구)지번주소</td>
            <td>도로명주소</td>
        </tr>
        </thead>
        <tbody>
            {% for row in data_lists %}
            <tr>
                <td> {{row.shopname}}</td>
            </tr>
            </tbody>
        </table>
    </div>

```

```

        <td>{{row.telnumber}}</td>
        <td>{{row.address}}</td>
        <td>{{row.newaddress}}</td>
    </tr>
    {% endfor %}
</tbody>
</table>
{{ pagination.links }}
</div>

```

```

#페이지네이션 PYTHON
@app.route("/searchdata", methods=["POST", "GET"])
def searchdata():
    conn = pymysql.connect(host=f'{host}',
                           user=f'{mysqluser}',
                           password=f'{pwd}',
                           db=f'{mysqldb}',
                           charset='utf8')

    if request.method == 'GET':
        search_word = request.args['search_word']
        pages = 0
        if (request.args.getlist('page')):
            pages = (int(request.args.getlist('page')[0]) - 1) * 10
            print('page', pages)
        with conn:
            with conn.cursor(pymysql.cursors.DictCursor) as curs:
                per_page = 10
                page, _, offset = get_page_args(per_page=per_page)
                # db = pymysql.connect(host='localhost', user='root', db='spartagram', password='12345678', charset='utf8')
                curs.execute("SELECT * from cafelist WHERE shopname LIKE '%{}%' ORDER BY idnumber DESC;".format(search_word
d))

                all_count = len(curs.fetchall())
                print(all_count)
                # '{}'을 쓰면서 %s로 변수를 받을께 있다!? = f스트링이 답이다
                sql = f"SELECT * from cafelist WHERE shopname LIKE '{search_word}' ORDER BY idnumber DESC LIMIT {per_pag
e) OFFSET {pages};"
                print('sql', sql);
                curs.execute(sql)
                data_list = curs.fetchall()
                print(data_list)
                pagination = Pagination(page=page, per_page=per_page, total=all_count, record_name='searchdata',
                                       css_framework='foundation', bs_version=5, prev_label="<<", next_label=">>")
                print("this is pagination", pagination)
                check = False
                return jsonify({'data': render_template('response.html', data_lists=data_list, pagination=pagination)})

```

마지막까지..... 우여곡절이 많았지만 어쨌든 해결되고 잘 돌아가면 된겁니다!!!!!!! 끝까지 포기 안하고 한 우리 모두 칭찬해~..

각자의 위치에서 잘 해내주셔서 너무 편했습니다 여러분.....

마치 포르투갈을 1:2로 이긴 대한민국 대표팀과 같은 역전승과 같은 느낌이었습니다!!!!!!!

중요한건 꺾이지 않는 마음.....♥

