**최종 결과 보고서**

**그룹 운동 커뮤니티 어플리케이션**

**(DIETA)**

Ver. 2

2019.06.03

한국외국어대학교

융복합소프트웨어학과

2팀(Make U)

**문서 정보**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 구분 | 소속 | 성명 | 날짜 | 서명 |
| 작성자 | 한국외국어대학교 | 최진용 | 2019-06-02 |  |
| 한국외국어대학교 | 이정훈 | 2019-06-02 |  |
| 한국외국어대학교 | 윤지영 | 2019-06-02 |  |
| 한국외국어대학교 | 김영인 | 2019-06-02 |  |
| 한국외국어대학교 | 정진원 | 2019-06-02 |  |
| 검토자 | 한국외국어대학교 | 이정훈 | 2019-06-03 |  |
| 사용자 |  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 승인자 | 한국외국어대학교 | 홍진표 |  |  |

머리말

본 문서는 장고 프레임워크로 제작할 운동 커뮤니티 웹 어플리케이션 서비스 소개와 목적을 설명하고, 본 시스템을 구축하기 위한 시스템 상세 설계를 기술한다.

목차

[**1.개요 5**](#_Toc8674160)

[1.1 서비스 정의 5](#_Toc8674161)

[1.2 문서의 목적 5](#_Toc8674162)

[**2. 시스템 구성요소 및 핵심기능 6**](#_Toc8674163)

2.1 시스템 구성요소…………......................................................................................................................................... 6

2.2 핵심기능…………………………………………………………………………………………………………………………………..7

[**3. 소프트웨어, DB, 서버 설계 8**](#_Toc8674165)

[3.1 인터페이스 어플리케이션 sequence diagram 8](#_Toc8674166)

[3.2 인터페이스 어플리케이션 설계 9](#_Toc8674167)

[3.3 인터페이스 어플리케이션 설계 목록](#_Toc8674168) 9

## 3.4 데이터베이스 설계………………………………………………………………………………………………………………..10

3.5 서버 개설 및 배포……….……………………………………………………………………………………………………….10

[**4. 프로젝트 구현 및 결과 12**](#_Toc8674169)

[4.1 디바이스를 활용한 운동량 측정 및 칼로리 정보 전송 12](#_Toc8674171)

[4.2 운동 진행 상황 따른 팝업 메시지 동기부여 16](#_Toc8674173)

## 4.3 운동모임 개설 및 투표…………..…………………………………………………………………………………………….18

## 4.4 게시판 작성…………………………………………………………………………………………………………………………..19

4.5 기타 구현(로그인, 회원가입, 그룹 경쟁 등)………………………………………………………………………..24

[**5. 결론 27**](#_Toc8674175)

1.개요

1.1 서비스 정의

▶ 자사가 서비스하는 ‘Dieta’는 GPS정보를 활용하여 그룹 매칭을 지원하고, 단체 운동에서 활동량을 기준으로 게임 랭킹 시스템을 적용한다. 이를 통해 함께 하는 다이어트의 즐거움과 지속성을 도모할 수 있는 환경을 제공한다.

1.2 문서의 목적

▶ 본 문서는 자바 프로그래밍 기반 운동 커뮤니티 ‘Dieta’ 어플리케이션 개발 사업에 관련해 시스템, 기능. 성능. 인터페이스. 데이터, 테스트, 보안, 품질, 제약, 관리 및 지원 등 개발관련 요구사항을 토대로 설계 시 필요한 상세설계 부분에 대하여 중점적으로 작성한다.

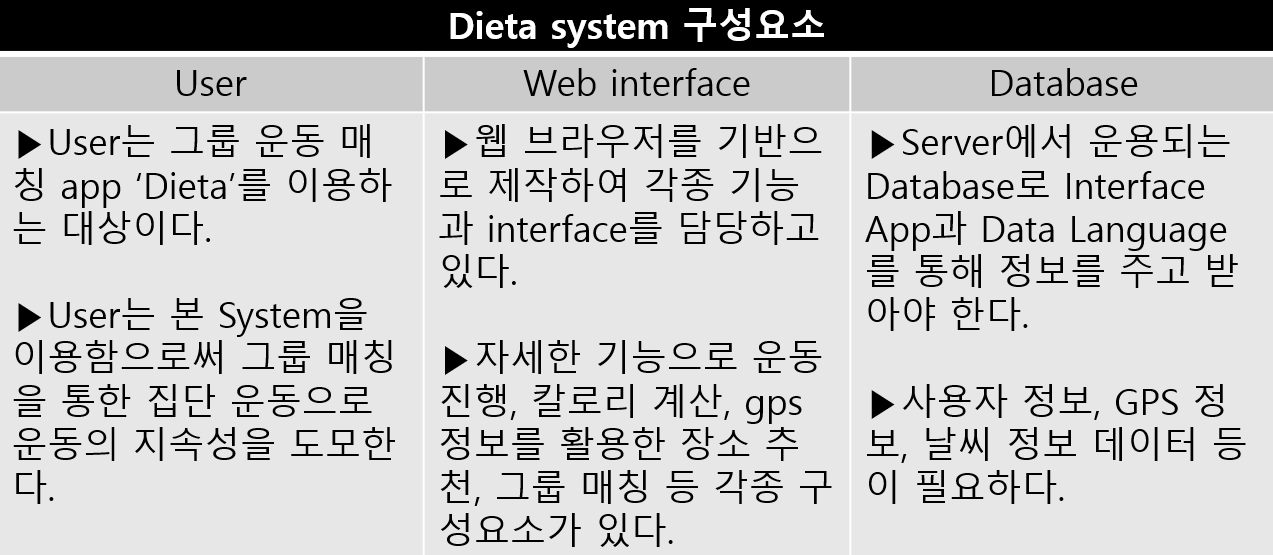
2. 시스템 구성요소 및 핵심기능

2.1 시스템 구성요소

본 System의 구성요소에는 User, web Interface, Database가 포함된다.



아래에 구성요소별로 정리한 표가 있다.



2.2 핵심 기능

1. **게시판 및 캘린더 기능**

- 다이어트 웹 기반 커뮤니티로서의 기능 수행하기 위함.

- 그룹 운동 날짜를 선정하기 위함

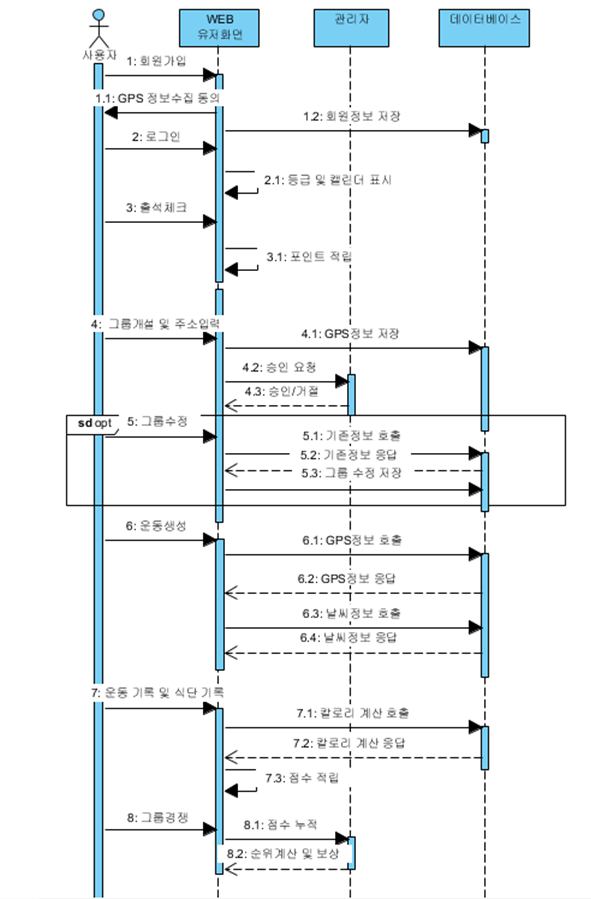
1. **날씨예보 api를 활용한 의사결정 지원**

- 사용자 운동 모임 개설 시 날씨 기반한 적합한 실내/실외 활동 선택 가능

1. **칼로리 측정을 통한 운동 평가 기능**
2. **운동 진행상황 별 팝업 메시지를 통한 동기부여**
3. **포인트 누적을 통한 그룹 경쟁 기능**

3. 소프트웨어, DB, 서버 설계

3.1 인터페이스 어플리케이션 sequence diagram



3.2 인터페이스 어플리케이션 설계

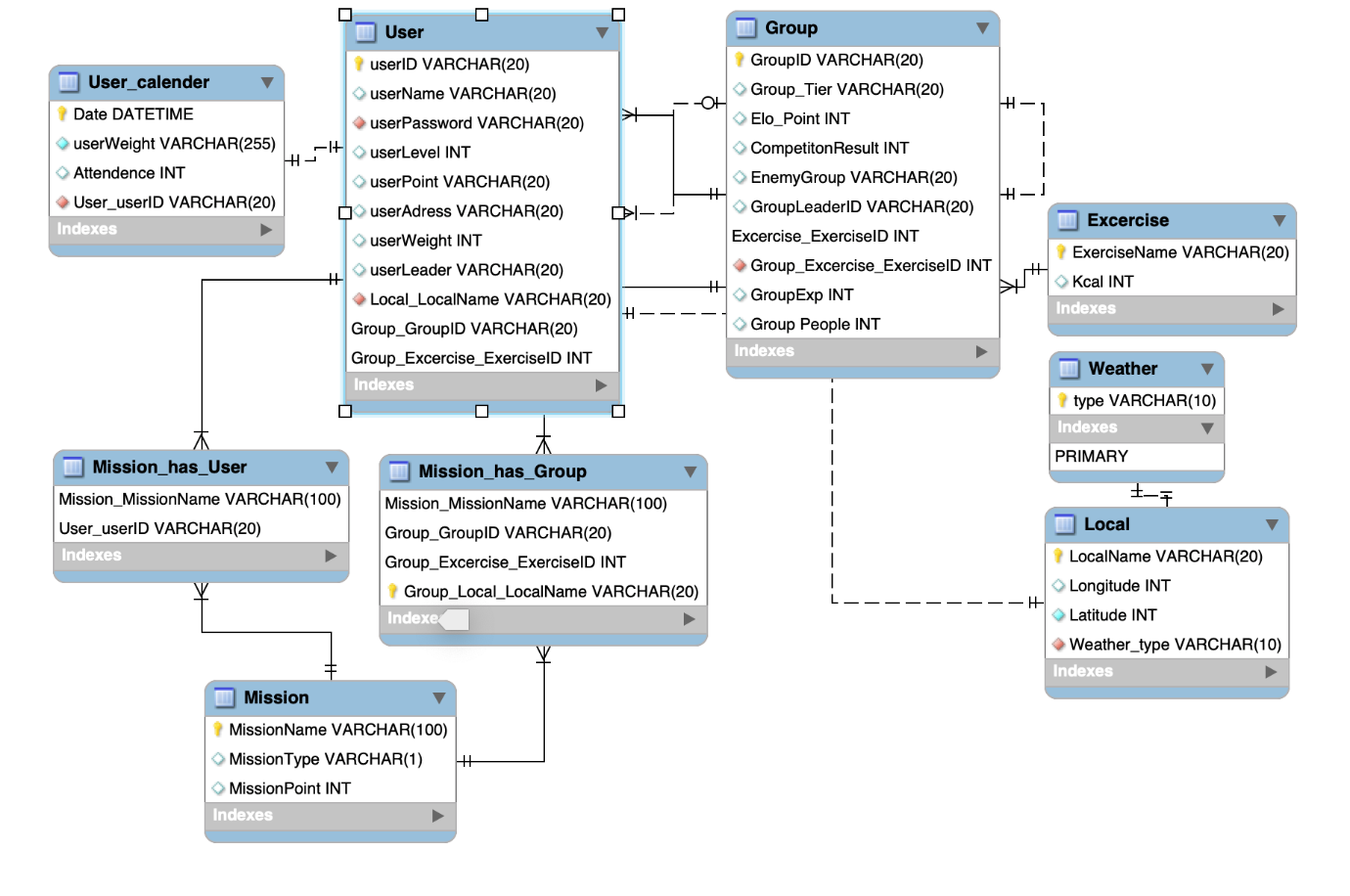
- 회원이 디바이스(안드로이드 스마트폰, 스마트 워치 등)를 갖고 운동 시 DIETA 웹 어플리케이션을 통해 Tracking된 운동 종목 및 칼로리 정보가 기록된다. 이때 Google Fit에서 제공하는 REST API를 활용한다. 또 운동 진행 상황에 따라 목표 팝업 메시지를 통해 일정 목표량 완수를 위한 동기부여를 지원한다.

- DIETA 웹 어플리케이션은 회원들에게 커뮤니티 활동을 지원한다. 회원은 모바일 혹인 웹을 통해 로그인하여 운동 개설 및 참여, 게시판 글 작성 또한 캘린더를 활용해 운동 일정을 잡을 수 있다. 특히, 운동 개설 시 입력된 위치정보에 따라 날씨 정보가 나타나 운동 개설 시 참고할 수 있다.

3.3 인터페이스 어플리케이션 설계 목록

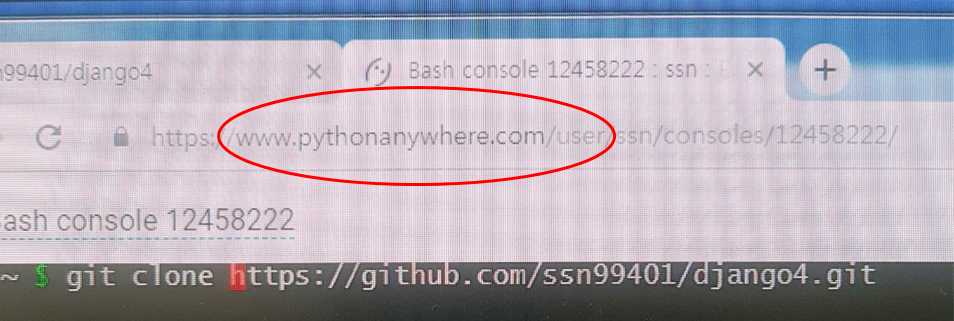
|  |  |  |  |
| --- | --- | --- | --- |
| 상세유형 | 요구사항 명 | 상세설계 ID | 비고 |
| 인터페이스 부 상세설계 내역 (SFD-I) | 홈 메인 화면 제공 | SFR-S-1 |  |
| 회원가입 기능 제공 | SFR-S-2 |
| 로그인 기능 제공 | SFR-S-3 |
| 로그아웃 기능 제공 | SFR-S-4 |
| 게시판 글 목록 기능 제공 | SFR-S-5 |
| 게시판 글 상세보기 기능 제공 | SFR-S-6 |
| 게시판 글 작성 기능 제공 | SFR-S-7 |
| 캘린더 기능 제공 | SFR-S-8 |
| 캘린더 전, 후 달 이동 기능 제공 | SFR-S-9 |
| 캘린더 이벤트 추가 기능 제공 | SFR-S-10 |
| 회원 그룹 만들기 기능 제공 | SFR-S-11 |
| 그룹 회원 추가 기능 제공 | SFR-S-12 |
| 운동 모임 개설 기능 제공 | SFR-S-13 |
| 날씨예보 api 활용 의사결정 지원 | SFR-S-14 |
| 칼로리 측정을 통한 운동 평가 기능 | SFR-S-15 |
| 운동진행 상황 별 팝업 메시지 기능 | SFR-S-16 |
| 포인트 누적을 통한 그룹 경쟁 기능 | SFR-S-17 |

3.4 데이터베이스 설계



- DIETA에 가입한 회원 정보는 데이터베이스에 적재된다. 더불어 입력된 주소 데이터에 맞게 저장된 GPS 및 날씨 정보 데이터베이스에서 호출된다. 디바이스를 통해 측정된 운동 데이터의 경우 운동 이름, 소모 칼로리 등 테이블 COLUMN에 맞게 적재된다.

3.5 서버 개설 및 배포

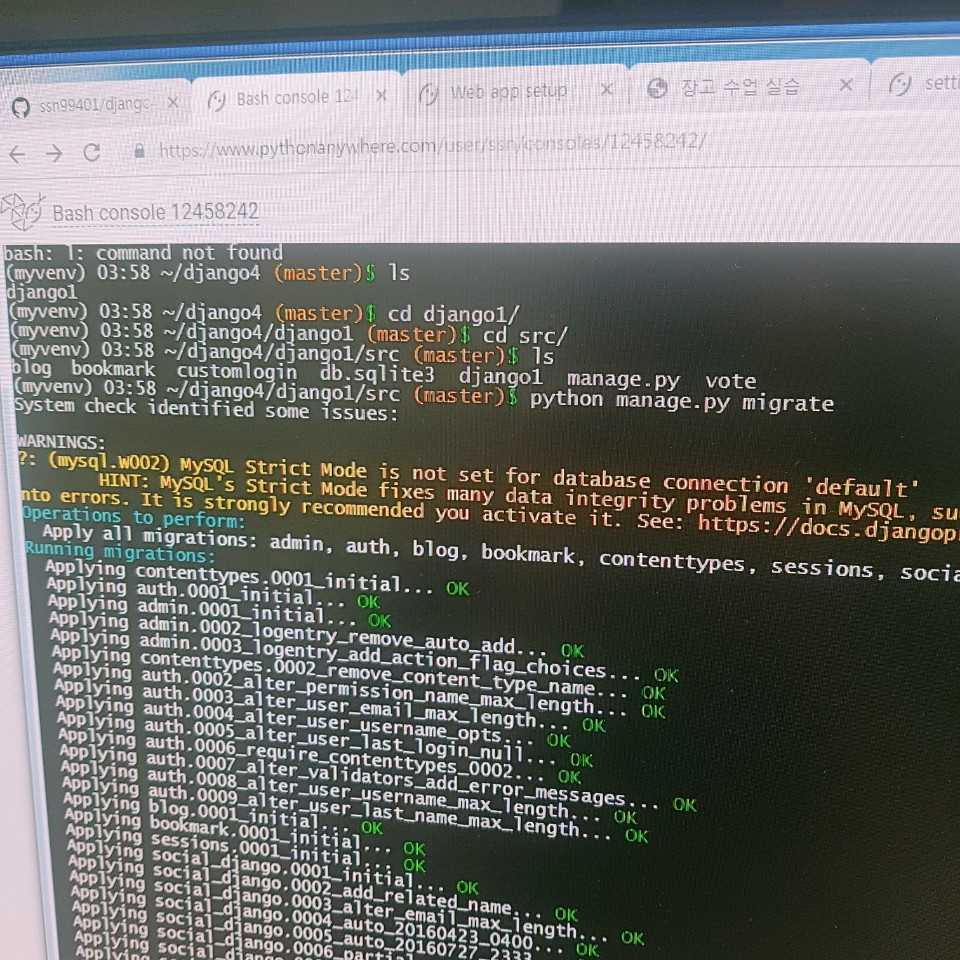


- pythonanywhere을 통해 서버와 연결한다.

- Git에 있는 파일을 가져오고 가상환경을 만들어준다.

- pip install을 통해 적합한 가상환경을 생성한 뒤, migrate를 해준다.

- mysql과 연결시키고 pythonanywhere에서 접근 가능하도록 setting.py를 변경한다.



DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'ssn$django4',

'USER':'ssn',

'PASSWORD':'duddlsrla7',

'HOST':'ssn.mysql.pythonanywhere-services.com',

}

}

ALLOWED\_HOSTS = ['.pythonanywhere.com']

web을 생성해준뒤 가상환경을 적용시키고 reload 시킨다.

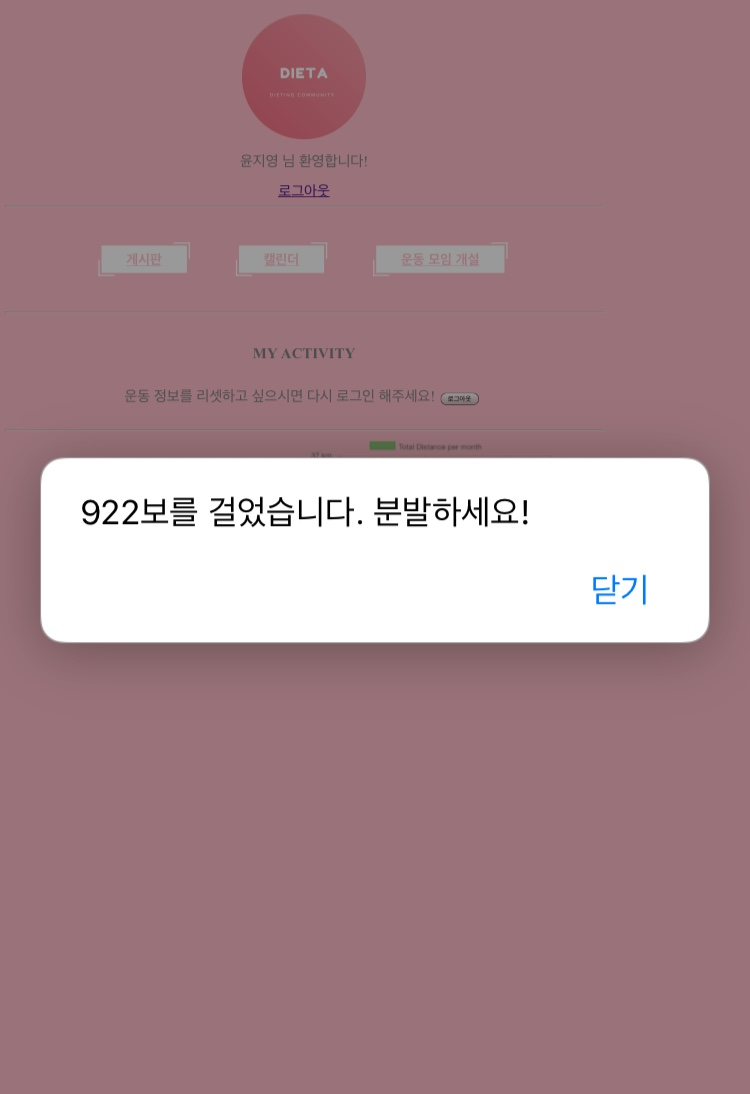
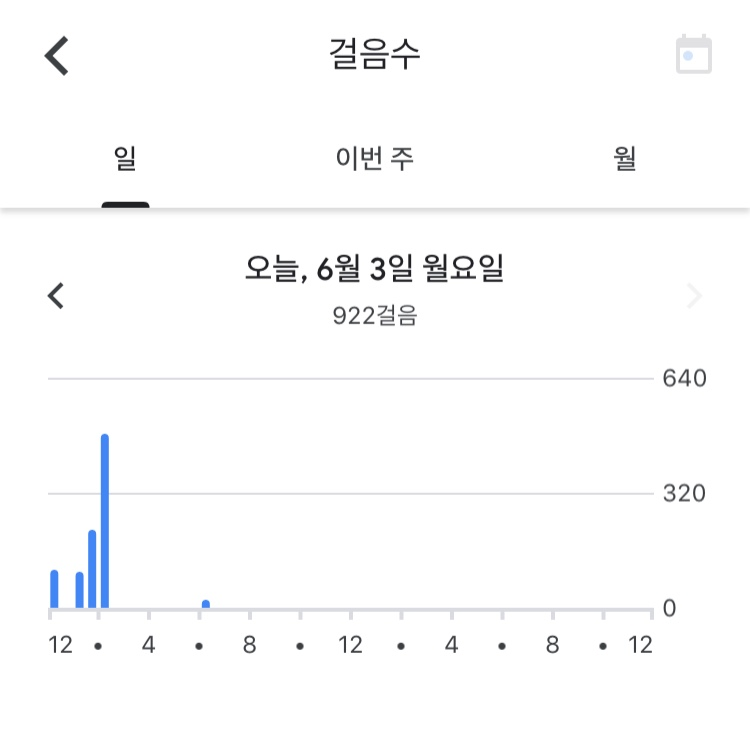
4. 프로젝트 구현 및 결과

4.1 디바이스를 활용한 운동량 측정 및 칼로리 정보 전송

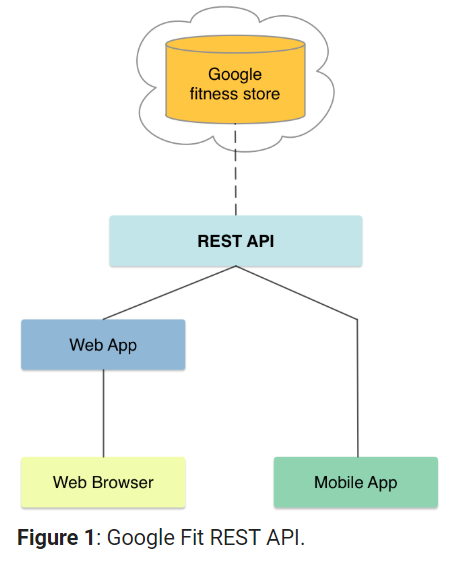
- 디바이스(안드로이드 스마트폰, 스마트 워치)를 통해 실시간으로 운동량이 측정된다. 이때 Google fit에서 제공하는 REST API를 활용한다.



- 구글 계정으로 연동하여 로그인 시 구글 피트니스 어플과 연동한 정보를 받을 수 있다.



- 구글 핏(Google fit)에서 제공된 걸음 수 정보



- Google fit 어플은 fitbit 웨어러블 기기와 연동이 가능하다.



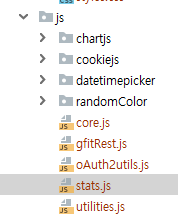
- Google Fitness API 사용 ( Google Fit 어플)



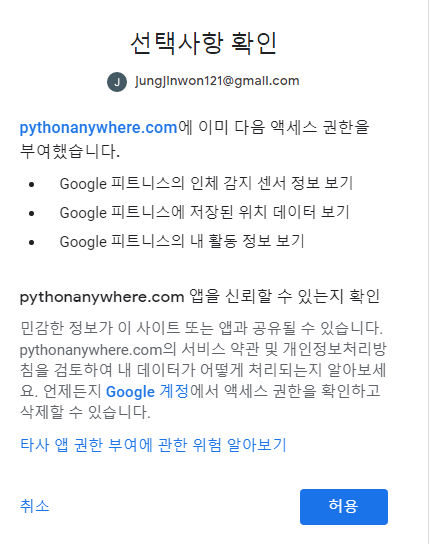
- 필요한 암호, 아이디 ReDIRECT 페이지



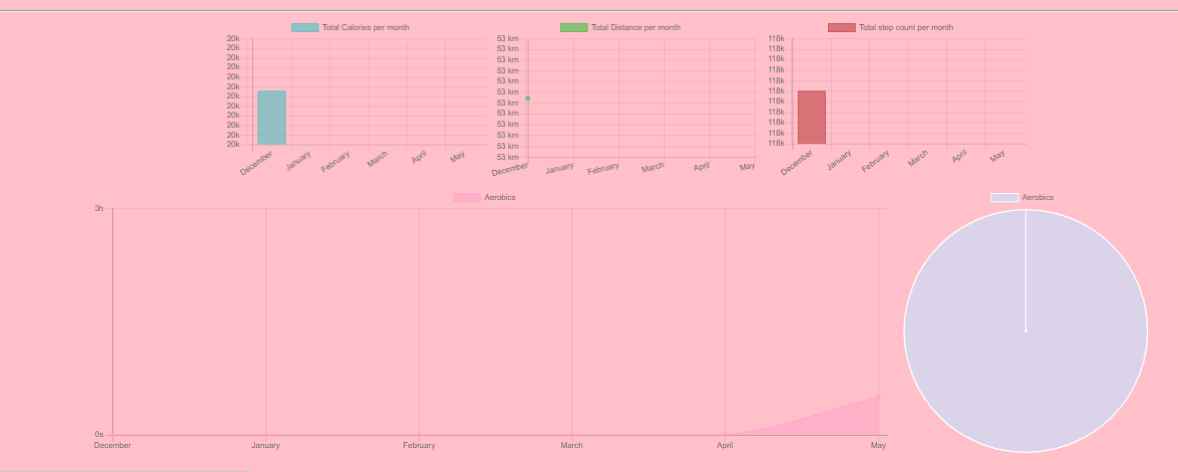
- Scopes fitness로 어플리케이션내 운동정보 읽어옴



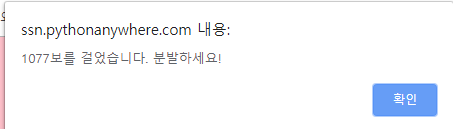
- Html 파일 이해 쉽게 js파일 여러 개 만듦.



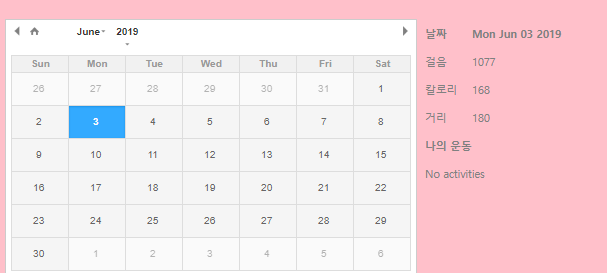
- 데이터 가져오기 버튼을 누르면 다음과 같은 배너가 뜨고 액세스 권한을 받습니다.



- 시각적인 효과를 더 주기 위해 월별 운동데이터, 그래프 그림 Chart.js에서 여러 형태의 운동량 표출 : 아직 부족한 운동량으로 독려 메시지 송출



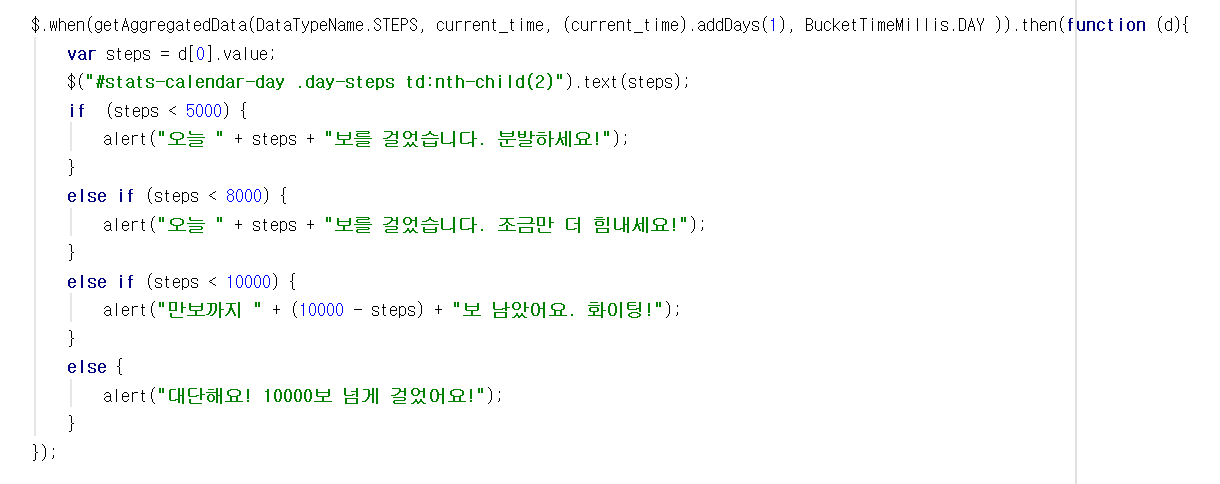
- 운동 자극을 위해 걸음수 데이터를 가져오면 배너를 표출한다.

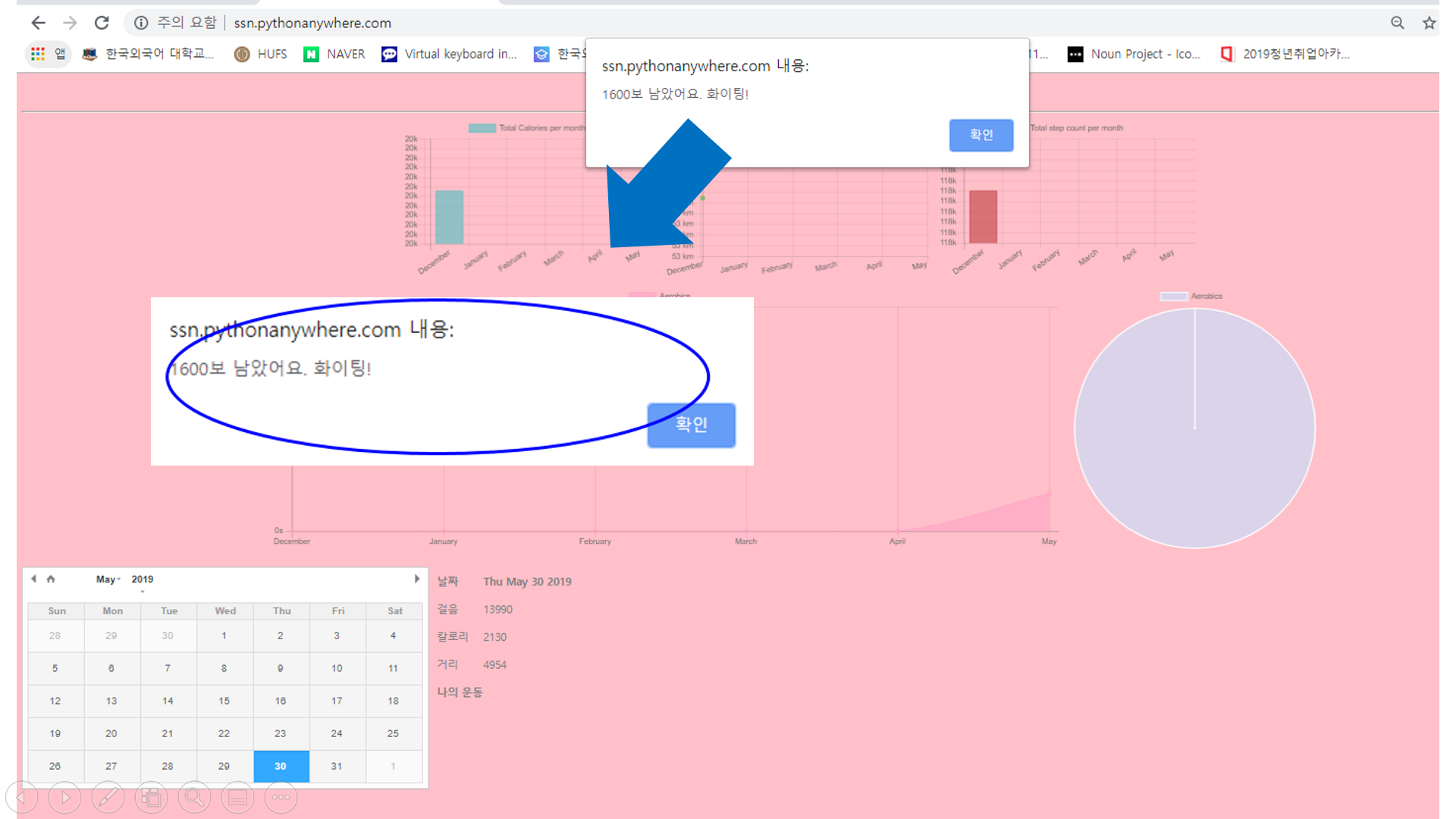


- 그 전 데이터의 걸음, 칼로리도 가져올 수 있다.

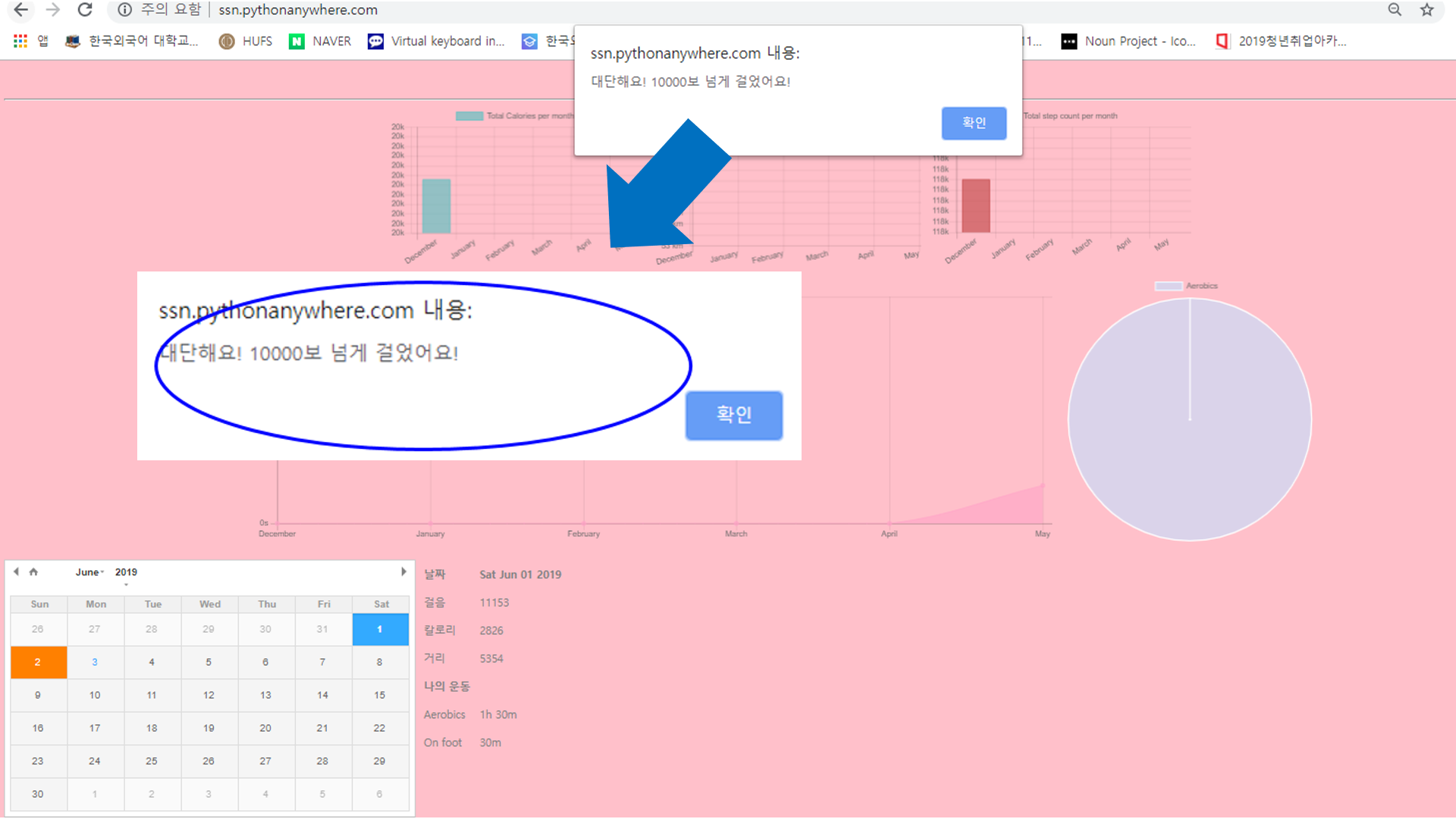
4.2 운동 진행 상황 별 팝업 메시지 동기부여

- 디바이스(안드로이드 스마트폰, 스마트 워치)를 통해 측정된 운동량에 따라 팝업 메시지를 띄어 동기부여를 한다. 한 예로, 걸음 수가 8,000보 이상 걸으면 ‘10,000보까지 몇 보가 남았습니다’라는 팝업을 띄워 10,000보를 채울 수 있도록 동기부여를 한다. 그리고 10,000보 이상 걸었을 경우 칭찬의 메시지를 띄어 보상 효과를 제공한다.





▶목표 운동량 대비 현재 운동량과 부족 시 독려



▶목표 운동량 달성 시 달성 메시지를 통한 보상 효과

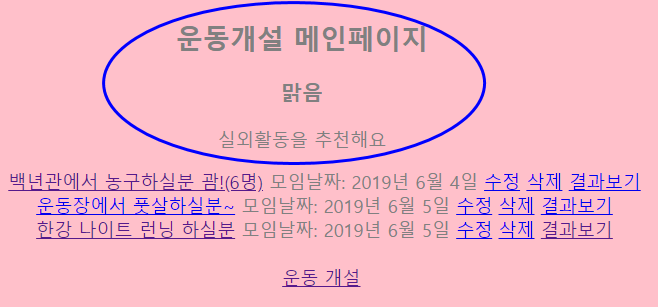
4.3 운동모임 개설 및 투표

- 운동 개설 시 사용자 위치를 중심으로 날씨 정보를 호출하여 사용자에게 제공한다. 회원은 위 정보를 의사결정 시 참고할 수 있다.

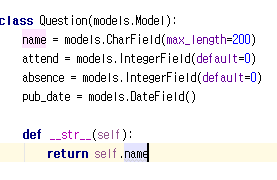
- 운동목록이 나타난다.

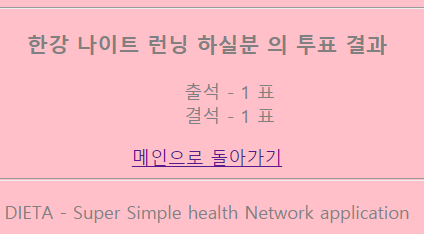
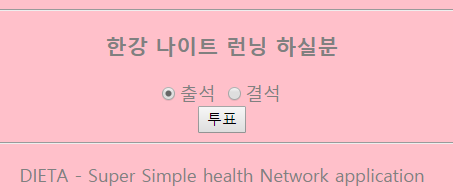






- 운동을 개설하고 투표를 통해 참여의사를 나타낸다.





**▶ 운동 개설 후 투표 기능**

4.4 게시판 작성

- 다이어트 웹 기반 커뮤니티로서의 기능을 수행하기 위함

- 캘린더를 제공하여 그룹 운동 날짜를 선정 및 계획 가능

1. **게시판 글 목록 기능 제공**

그룹 사용자가 글 목록을 볼 수 있다.

▶ from django.views.generic.list import ListView

ListView(특정 모델클래스의 객체 목록을 다루는 기능이 있는 제네릭 뷰)를 이용해 html에 Post객체를 넘겨준다. ListView에서 paginate\_by변수를 이용해 목록에 몇 개의 객체를 출력할지 설정할 수 있다.

▶{% if page\_obj.has\_previous %}

{% if page\_obj.has\_next %}

html에서 함수를 이용해 이전 페이지와 다음페이지 버튼을 만들어준다.

▶<table width=*"90%"*>

<tr>

<th>글번호</th>

<th>카테고리</th>

<th>제목</th>

<th>작성자</th>

<th>작성일</th>

</tr>

{% for obj in objs %}

<tr>

<th>{{ obj.id }}</th>

<th>{{ obj.c }}</th>

<th><a href=*"{% url 'blog:detail' obj.id %}"*>{{ obj.title }}</a></th>

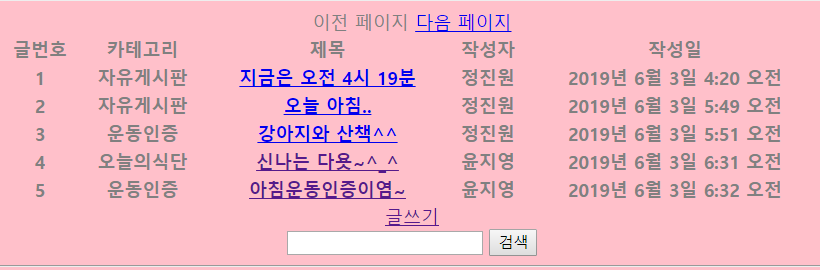
<th>{{ obj.u }}</th>

<th>{{ obj.pub\_date }}</th>

</tr>

{% endfor %}

그리고 table태그로 모델클래스 안의 값을 나열해준다



1. **게시판 글 상세보기 기능 제공**

그룹 사용자가 목록의 글을 읽을 수 있다.

▶from django.views.generic.detail import DetailView

DetailView(데이터베이스에 특정 모델클래스에서 한 개의 객체를 추출하는 제네릭 뷰)를 views.py에 작성해준다.

▶<a href=*"{% url 'blog:detail' obj.id %}"*>{{ obj.title }}</a>

목록부분의 html에서 a태그를 이용해 해당 객체의 id를

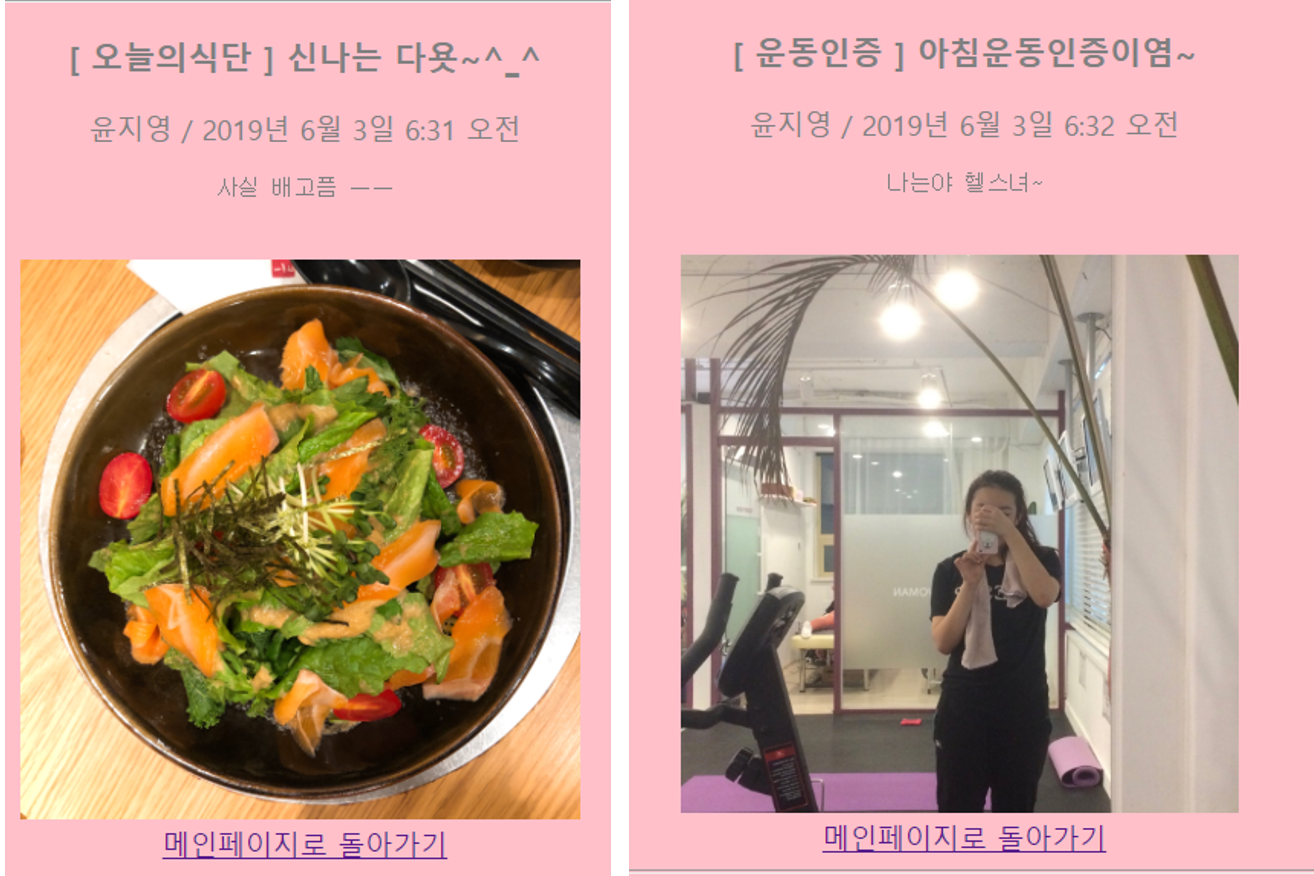
path(*'<int:pk>/'*, Detail.as\_view(), name=*'detail'*)

pk변수로 views.py의 DetailView로 전해주고 html파일로 뿌려준다.

▶<a download href=*"{{ f.f.url }}"*>{{ f.file.name }}</a>

<img src=*"{{ f.i.url }}"* width=*"40%"*>

전해진 Post객체의 정보 중 파일은 a태그를 이용해 다운로드를 할 수 있도록 해주고 이미지는 img태그를 이용해 띄워준다.



1. **게시판 글 작성 기능 제공**

그룹 사용자가 카테고리를 선택해 글과 그림을 올릴 수 있다

▶from django.views.generic.edit import FormView

FormView(폼클래스를 기반으로 사용자에게 입력공간을 제공하고 사용자가 입력한 데이터르 바탕으로 객체를 생성해주는 뷰)를 이용해 views.py에 작성해준다,

▶files = forms.FileField(label=*'첨부파일'*, required=False)

images = forms.ImageField(label=*'이미지'*, required=False)

forms.py에 FormView 클래스와 연동할 폼클래스를 만들어준다. FileField, ImageField를 이용해서 첨부파일과 이미지파일을 올릴 수 있도록 해준다.

(모델클래스에 파일과 이미지 모델클래스를 만들어 주었다.)

이 때, 이미지파일 이용을 위해 pip install Pillow을 해줘야한다

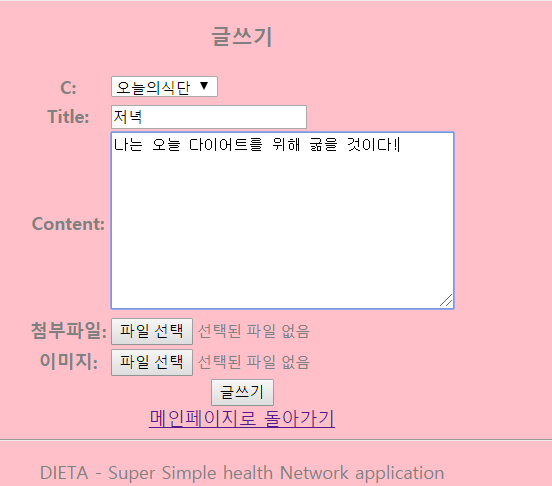
▶def **form\_valid**(*self*, form):

p = form.save(commit=False)

p.u = *self*.request.user

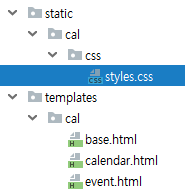
p.save()

폼의 사용자 입력이 유효한 경우, 저장한다.

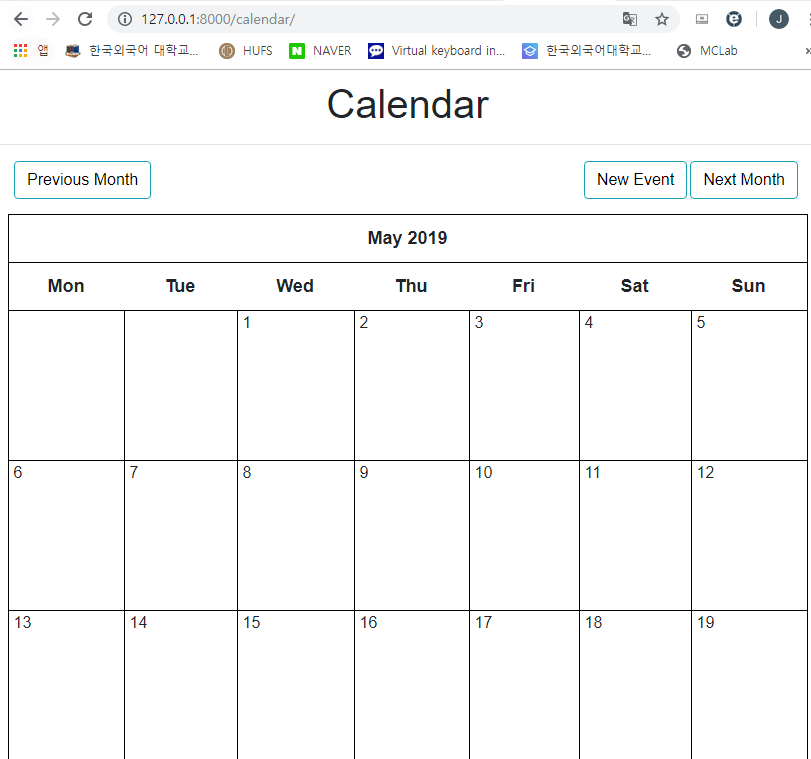


1. **캘린더 기능 제공**

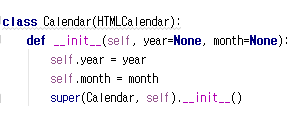
**1. Calendar 디렉토리**



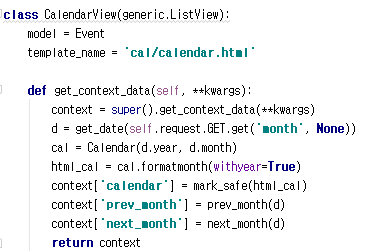
**2. Calendar 화면**



**▶ Utils.py에 Calendar 클래스 정의**



**▶** **Utils.py에 정의된 calendar 클래스 views.py에서 사용**



4.5 기타 설계(로그인, 회원가입, 그룹 경쟁 등)

1. **회원가입 기능 제공**

사용자가 이 엡에 가입할 수 있도록 한다

▶*'django.contrib.auth'*,

장고에 원래 설치되어 있는 앱을 이용해서 회원 가입 기능을 만들어준다.

▶ from django.contrib.auth.models import User

모델은 만들어져 있는 user 모델 클래스를 이용하고 forms.py로 입력창을 만들어준다.

▶ widgets = {*'password'* : forms.PasswordInput()}

이 때 비밀번호를 설정할 때, 위젯을 이용해 비밀번호를 보호한다.

▶views.py에서는 사용자에게 회원가입폼 객체 생성 및 html파일을 전달해주고, 사용자가 채워 넣은 정보가 유효한지 확인해주는 if문을 통과하면

User.objects.create\_user을 통해 유저를 추가해준다.



1. **로그인 기능 제공**

사용자가 로그인을 할 수 있다.

▶class **SigninForm**(forms.models.ModelForm):

class **Meta**:

model = User

fields = [*'username'*, *'password'*]

widgets = {*'password'* : forms.PasswordInput()}

Forms.py에 입력폼을 만들고 views.py에서 사용자가 입력한 값을 받는다

id = request.POST.get(*'username'*)

pw = request.POST.get(*'password'*)

▶u = authenticate(username=id,password=pw)

데이터베이스에 입력정보가 일치하면

login(request,u)

로그인을 해준다.



1. **로그아웃 기능 제공**

사용자가 로그아웃을 할 수 있다.

▶Views.py에 클라이언트의 User정보를 지워주는 함수를 사용해준다.

logout(request)



5. 결론



DIETA 앱을 통해

- 사용자가 [www.dieta.com](http://www.dieta.com)으로 Web을 통해 접속하면 Dieta가 지원하는 기능을 사용할 수 있다.

- 회원가입 후 로그인을 하면 출석체크를 하게 된다. 이때 개인 포인트가 적립되며 추후에 제휴된 제품을 구매할 때 포인트를 쓸 수 있다.

- User가 그룹 생성을 요청하면 관리자가 그룹을 생성한다.

- User가 운동 목록을 만들면 회원들이 참여 투표를 한다. 그리고 참여 현황을 알 수 있게 된다. 사용자 위치를 중심으로 날씨 정보와 연동해 사용자가 카테고리에 지정된 운동을 선택 시 도움을 준다.

- 사용자는 게시판을 들어가서 커뮤니티 활동을 할 수 있다.

- 사용자는 별도의 개설된 링크를 통해 운동에 참여할 수 있고, 디바이스를 통해 실시간으로 운동 정보가 기록된다.