

Default = 꾸준함

주제환 입니다.



Profile

Skills

Experience

Plan



이름 : 주세환(JOO SE HWAN)

생년월일 : 1990.06.27

E-mail : j4615545@gmail.com

GitHub : <https://github.com/JOO-1.0000-Zi>

평생교육진흥원 멀티디미어학과 졸업

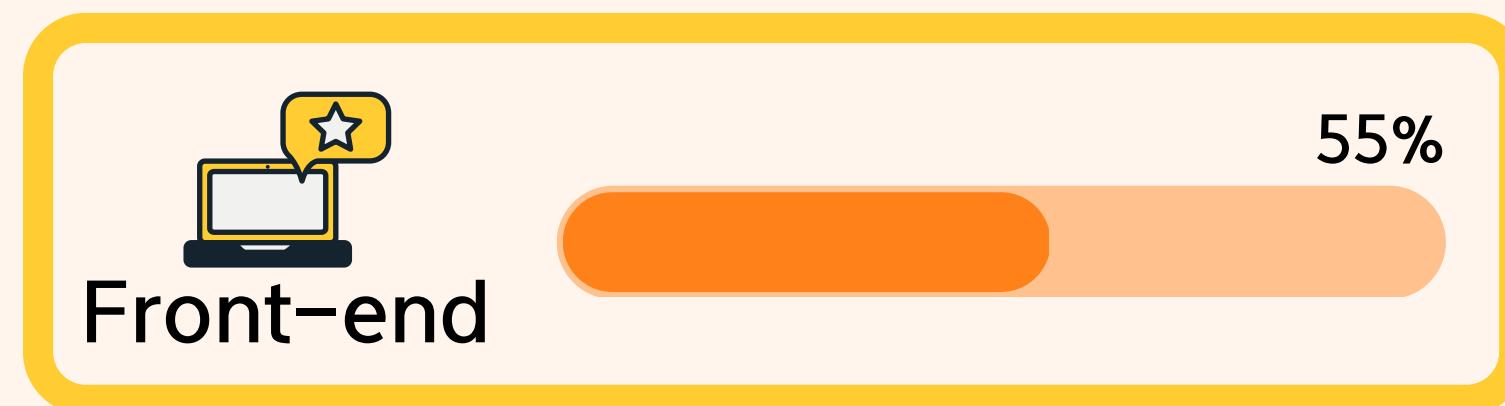
멀티캠퍼스 KDT 실무 맞춤형 풀스택 개발 교육 수료

Profile

Skills

Experience

Plan



MySQL

Git

Windows

Max

Notion

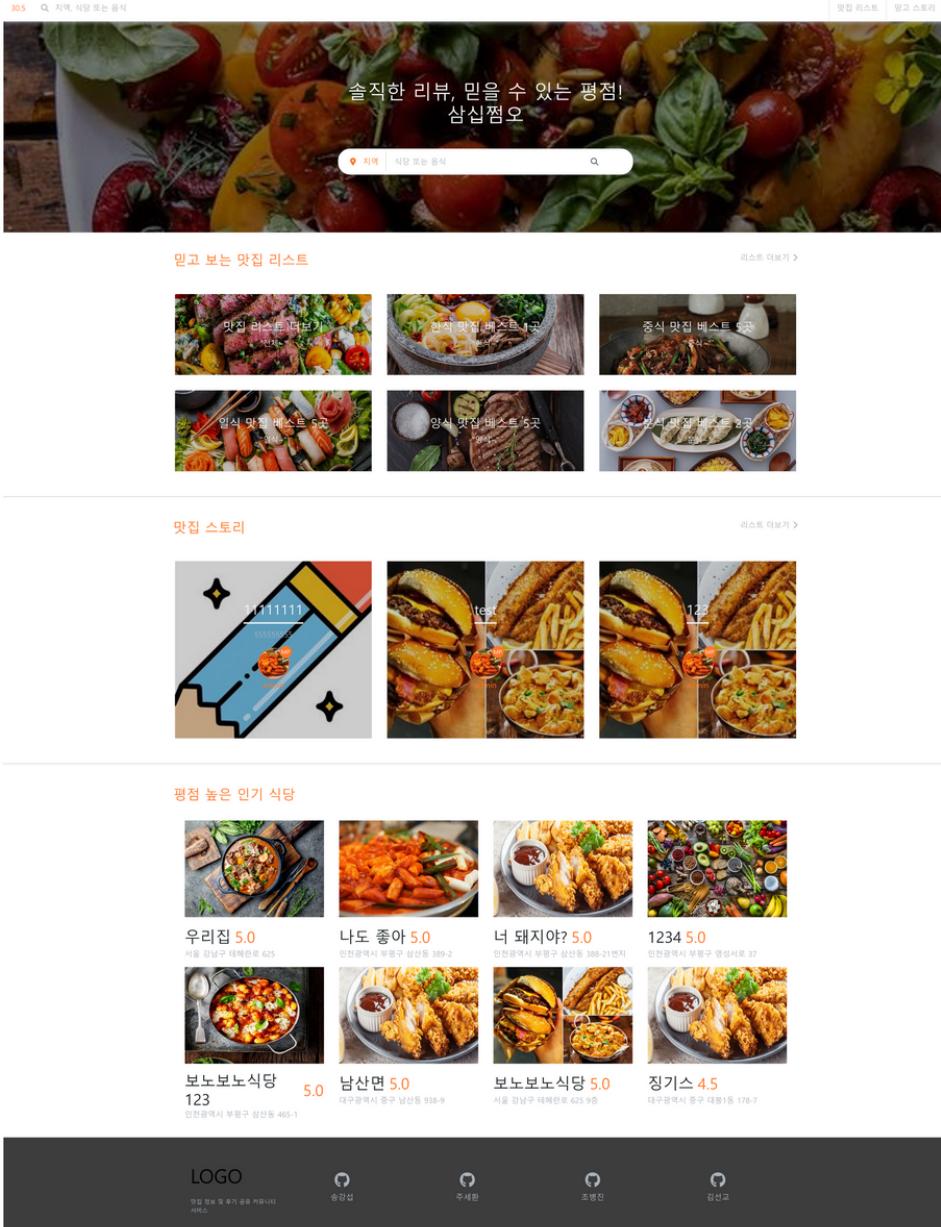
Jira

HTML

CSS

VSCode

프로젝트 30.5



기획 - 맛집 정보 및 후기 공유 커뮤니티 서비스



일정 - 2022.10.31 ~ 2022.11.07



기술 -



GitHub - Project 30.5



역할 - 초기 가상환경 설치 및 Setting 기본 설정 공유

- 식당 App 생성 및 Model 생성 및 DB 구축
- Django 내장 AUTH_USER를 활용하여 회원 App 생성
- Bootstrap를 활용하여 식당 등록, 회원가입 폼 디자인
- 프로젝트 발표

프로젝트 30.5

The screenshot shows a web-based form for creating a new restaurant entry. The form fields include:

- 상호명 (Name): A text input field.
- 주소 (Address): A text input field.
- 전화번호 (Phone Number): A text input field.
- 기본 가격대 (Default Price Range): A dropdown menu with the option "기본 가격대".
- 영업 시간 (Business Hours): A text input field.
- 휴무일 (Holidays): A text input field.
- 태그 (Tags): A text input field.

At the bottom right of the form area, there is a large orange button labeled "추가하기" (Add).

식당 Create

- restaurants의 Model과 Form을 활용하여 Create 구현
 - 태그 부분을 ManyToManyField로 생성하여 향후 검색 기능을 통해 검색 될 수 있도록 구현

```
class Restaurant(models.Model):
    name = models.CharField(max_length=80)
    address = models.CharField(max_length=80)
    shop_number = models.CharField(max_length=80)
    between_pay = models.CharField(max_length=80)
    opening_time = models.CharField(max_length=80)
    break_day = models.CharField(max_length=80)
    subtext = models.CharField(max_length=140)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)
    hits = models.IntegerField(default=0)
    tags = models.ManyToManyField("restaurants.Tag", related_name="restaurants")
```

프로젝트 30.5

The screenshot shows a web application interface for managing restaurants. At the top, there's a navigation bar with a search bar containing '30.5' and placeholder text '지역, 식당 또는 음식'. To the right of the search bar are three icons: a menu, a user profile, and a person icon.

The main content area has a title '내 점포 등록 하기' (Register New Store). Below the title are several input fields:

- 상호명**: A text input field with placeholder '상호명'.
- 주소**: A text input field with placeholder '주소'.
- 전화번호**: A text input field with placeholder '전화번호'.
- 기본 가격대**: A text input field with placeholder '기본 가격대'.
- 영업 시간**: A text input field with placeholder '영업 시간'.
- 휴무일**: A text input field with placeholder '휴무일'.
- 태그**: A text input field with placeholder '태그'.

At the bottom left is a dark button labeled '추가하기' (Add).

식당 Create

- Bootstrap5의 폼을 활용하여 기본 디자인을 구성 함
 - method를 POST로 데이터 변환을 하여 넘겨 주기 때문에 `csrf_token`을 함께 적용

```
% extends 'base.html' %
% load django_bootstrap5 %
% block title %}내 점포 등록{% endblock title %

% block body %
<div class='>
  <h2 class='text-center mt-4'>내 점포 등록 하기</h2>
  <div class='container mt-5'>
    <hr>
    <form action="" method="POST" enctype="multipart/form-data">
      {% csrf_token %}
      {% bootstrap_form restaurants_form %}
      <input class='btn btn-dark my-3' type="submit" name="" id="" value='추가하기'>
    </form>
  </div>
</div>
% endblock body %}
```

프로젝트 30.5

남산면 4.7

4 3 0

주소	대구광역시 중구 남산동 938-9
전화번호	053-568-9090
가격대	만 원 이하
영업시간	11:00 ~ 21:00
쉬는시간	음식점에 직접 문의해주세요.
태그 #	서울대 샤대 대학가 숨은맛집
음식점 소개	"0"

식당 Read

- Model을 import해서 저장된 데이터를 활용
 - 해당 항목의 데이터를 정해진 위치 값에 노출 시킨다
 - Bootstrap의 Table을 이용해서 디자인 함

```
<div class="detail-info">
  <table>
    <tbody>
      <p></p>
      <tr>
        <th>주소</th>
        <td id="address">{{ restaurant.address }}</td>
      </tr>
      <tr>
        <th>전화 번 호</th>
        <td>{{ restaurant.shop_number }}</td>
      </tr>
      <tr>
        <th>가격 대</th>
        <td>{{ restaurant.between_pay }}</td>
      </tr>
```

프로젝트 30.5

[수정 하기](#)

상호명

남산면

주소

대구광역시 중구 남산동 938-9

전화번호

053-568-9090

기본 가격대

만 원 이하

영업 시간

11:00 ~ 21:00

휴무일

음식점에 직접 문의해주세요.

태그

[<Tag: 서울대>, <Tag: 샤대>, <Tag: 대학가>, <Tag: 숨은맛집>]

[수정하기](#)

식당 Update

- 식당의 기본 정보를 수정 할 수 있도록 구현
 - import한 Form의 수정하려고 하는 식당의 기존 데이터 값이 자동으로 입력 되도록 instance를 사용

```
def update(request, pk):
    restaurant = Restaurant.objects.get(pk=pk)
    if request.method == "POST":
        restaurants_form = RestaurantsForm(
            request.POST, request.FILES, instance=restaurant
        )
        if restaurants_form.is_valid():
            restaurant.tags.all().delete()
            tags = restaurants_form.cleaned_data["tags"].split(",")
            for tag in tags:
                ...
            restaurants = restaurants_form.save(commit=False)
            restaurants.save()
            return redirect("restaurants:detail", pk)
    else:
        restaurants_form = RestaurantsForm(instance=restaurant)

    context = {
        "restaurants_form": restaurants_form,
    }
    return render(request, "restaurants/update.html", context=context)
```

프로젝트 30.5

징기스 4.5

③ 3 편집 2 리뷰쓰기 ★ 0



주소 대구광역시 중구 대봉1동 178-7

전화번호 053-425-8592

가격대 1만 원대

영업시간 17:00 ~ 02:00

쉬는시간 음식점에 직접 문의해주세요.

태그 #

음식점 소개 "0"

업데이트 : 2023-02-09

식당 찜하기(좋아요) 기능

- 식당 정보 우측 상단에 별 모양을 찜하기(좋아요)버튼으로 구성
 - JavaScript를 활용하여 비동기로 구현
- 찜하기 한 식당은 마이페이지에서 모아서 볼 수 있다

The screenshot shows a restaurant profile page. At the top right, there is a user icon labeled 'admin' with a checked checkbox and a dropdown arrow. Below it, there are statistics: '리뷰 6' (Reviews 6), '팔로우 2' (Followers 2), and '팔로워 2' (Following 2). A circular thumbnail image of a dish is displayed. Below the profile area, there is a section titled '찜한 식당' (Favorited Restaurants) with three small images of food: a burger, a fish fillet with fries, and a bowl of soup.

프로젝트 30.5

찜하기 JavaScript Code

```
<!-- CDN axios -->
<script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
<script>
  const forms = document.querySelectorAll('.like-forms')
  const csrftoken = document
    .querySelector('[name=csrfmiddlewaretoken]')
    .value;

  forms.forEach((form) => {
    form.addEventListener('submit', function (event) {
      event.preventDefault();
      const restaurantId = event
        .target
        .dataset
        .restaurantId
      axios({
        method: 'post',
        url: `http://127.0.0.1:8000/restaurants/${restaurantId}/like/`,
        headers: {
          'X-CSRFToken': csrftoken
        }, // csrf token
      })
    })
  })
}
```

```
.then((response) => {
  console.log(response.data)
  const isLiked = response.data.is_liked
  const likeBtn = document.querySelector(`#like-${restaurantId}`)
  const starBtn = document.querySelector(`#star-${restaurantId}`)
  const likeCountTag = document.querySelector('#star-count')
  const likeCount = response.data.like_count
  likeCountTag.innerText = likeCount
  if (isLiked === true) {
    starBtn
      .classList
      .remove('fa-regular')
    starBtn
      .classList
      .add('fa-solid')
    starBtn.style.color = 'var(--main-color)';
  } else {
    starBtn
      .classList
      .remove('fa-solid')
    starBtn
      .classList
      .add('fa-regular')
    starBtn.style.color = 'var(--gray-color)';
  }
}.catch((error) => {
  console.log(error.response)
})
```

프로젝트 30.5

KPT 회고록

- Keep

- 궁금했던 기능을 직접 문서도 찾아보고 맨땅에 해당 하는 식으로 작업을 해보면서 다양한 코드를 반복하고 배울 수 있었고, 처음 4명이라는 인원이 팀을 이루어 처음 보는 사람과 어색함을 이겨내고 프로젝트를 하기 위해 서로 배려하면서 내가 모르는 부분을 남에게 이야기를 해야지 프로젝트를 성공적으로 완료 할 수 있다는 것을 배웠습니다.

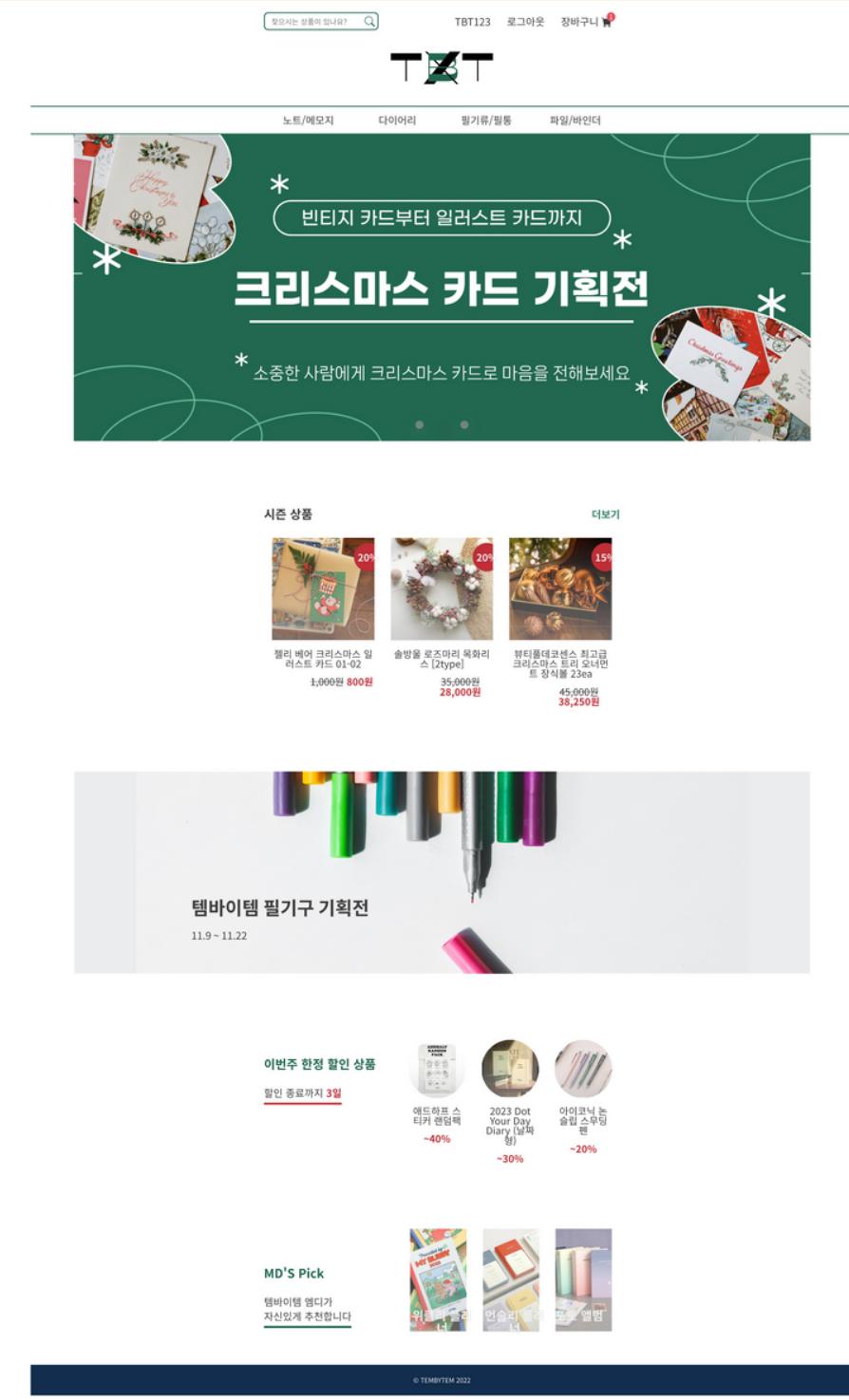
- Problem

- 시간이 처음에는 1주일이 길게 느껴졌으나 프로젝트를 하면서 기능을 개발하고 다양한 문서들을 찾아 가면서 하다 보니 원하는 기능을 제대로 구현을 못한 것들도 있고 하고 싶었던 기능을 포기 해야 하는 경우도 많았습니다 기간 안에 원하는 기능들을 모두 넣어서 프로젝트를 완성 하는 것이 쉽지 안아서 아쉬웠습니다.

- Try

- 교육에서 배우지 못한 기능을 만들려고 하다 보니 다양한 문서와 검색 통해서 코드를 가져와서 사용을 하려고 했으나 막상 가져와도 활용하는 것이 쉽지 않아서 새로운 기능은 팀원들과 다 함께 고민을 하면 구현을 했다. 개발자는 코드를 복사 + 붙여 넣기 하는 것도 능력이라는 것을 배웠습니다.

프로젝트 TBT



- 기획 - 문구 판매 및 상품 후기 제공 서비스**
- 일정 - 2022.11.09 ~ 2022.11.22**
- 기술 -**
- GitHub - Project TBT**
- 역할 - Django 내장 AUTH_USER를 활용하여 회원 App 생성**
 - 유저 팔로워/팔로잉 기능 구현
 - 상품 찜하기 기능 구현(비동기) / 작성한 리뷰 리스트
 - 쿠키를 활용한 장바구니 기능 제작
 - 총괄 팀장 (업무 분담 및 전체 일정 관리)

프로젝트 TBT

로그인 하기'."/>

찾으시는 상품이 있나요?

admin 로그아웃 상품 등록 장바구니

TBT

노트/메모지 다이어리 필기류/필통 파일/바인더

회원가입

아이디

닉네임

비밀번호

비밀번호 확인

이메일

가입하기

이미 아이디가 있으신가요? [로그인](#) 하기

© TEMBYTEM 2022

사용자 회원가입(Create)

- Django 내장 AUTH_USER를 통해 구현
 - Password 복잡도, 중복 체크 기능 까지 구현을 함
 - get_User_model을 import 해서 활용

```
class CustomCreationUserForm(UserCreationForm):
    class Meta:
        model = get_user_model()
        fields = ["username", "nickname", "email", "profile_image"]

    labels = {
        "username": "이름",
        "nickname": "닉네임",
        "email": "이메일",
        "profile_image": "프로필 사진",
    }
```

프로젝트 TBT

The screenshot shows the TBT app's user profile interface. At the top, there is a search bar and navigation links for '회원가입', '로그인', and '장바구니'. The main header 'TBT' is displayed above a navigation bar with tabs: '노트/메모지', '다이어리', '필기류/필통', and '파일/바인더'. Below this, the title '플렉스 님의 마이페이지' is shown. A user profile card for '플렉스 님' displays 1 follower and 0 following, with a '리뷰 목록' button. The '최근 리뷰' section lists a single review for '뷰티풀데코센스 최고급 크리스마스 트리 오너먼트 장식 봄 23ea' with a rating of ★5 and the comment 'good'. The 'Wish List' section shows four items: '2023 에브리데이 다이어리' (12,000원), '2023 마이 버디 다이어리' (11,000원), '아이코닉 2023 일상집 다이어리' (13,800원), and '[세트구성] 워너디스 A5 6공 스티커 바인더 + 양면 속지 40매' (12,000원). The bottom of the screen includes a footer with the text '© TEMBYTEM 2021'.

사용자 마이페이지(Read)

- 사용자 팔로우 기능 구현
- 작성 리뷰 리스트 구현
 - 더보기 탭을 통해서 전체 리뷰 확인 가능
- 짐한 상품 리스트 구현
 - 더보기 탭을 통해 전체 리스트 볼 수 있음
 - 더보기 리스트에서 짐하기 선택 삭제 바로 가능

The screenshot shows the 'Wish List' section of the TBT app. It displays four items from the previous screenshot with a red '삭제하기' (Delete) button in the top right corner. The items are: '2023 에브리데이 다이어리' (12,000원), '2023 마이 버디 다이어리' (11,000원), '아이코닉 2023 일상집 다이어리' (13,800원), and '[세트구성] 워너디스 A5 6공 스티커 바인더 + 양면 속지 40매' (12,000원).

프로젝트 TBT

회원 정보 수정

닉네임

플렉스

이름

길동

성

터

이메일

123@123.com

프로필 이미지

[사진수정하기](#)

파일 선택 선택된 파일 없음

[비밀번호 변경](#)[기본 정보 수정](#)[이전페이지로 돌아가기](#)

사용자 정보 수정(Update)

- 사용자의 정보를 수정 할 수 있도록 구현
 - ID는 고정 값이라 제외 하고 수정 할 수 있도록 하였음
 - 닉네임, 이름, 이메일, 프로필 사진을 수정 할 수 있다
 - 비밀번호 변경 또한 가능 하도록 구현 하였음

비밀번호 수정하기

기존 비밀번호

새 비밀번호

새 비밀번호 확인

[비밀번호 수정](#)[이전페이지로 돌아가기](#)

프로젝트 TBT

The screenshot shows a user interface for a project named 'TBT'. At the top, there is a search bar with placeholder text '찾으시는 상품이 있나요?' and a magnifying glass icon. To the right of the search bar are links for 'TBT123' (user profile), '로그아웃' (Logout), and '장바구니' (Cart) with a red notification badge showing '1'. Below the header is a logo consisting of the letters 'TBT' where the 'B' is partially obscured by a green square.

Below the header, there are four navigation tabs: '노트/메모지', '다이어리', '필기류/필통', and '파일/바인더'. Under the '다이어리' tab, a section titled '플렉스 님의 장바구니' (Flex's Cart) is displayed. It lists a single item: '아이코닉 2023 일상집 다이어리' (Iconic 2023 Daily Life Diary). The item details are: 20% off, price 11,040원 (original 13,800원), quantity 4, total price 44,160원, and shipping cost 2,500원. A delete icon is also present. Below the item summary, it says '총 1 개의 상품을 선택하셨습니다.' (You selected 1 product). At the bottom, a summary table shows the breakdown: 선택 상품 금액 (Selected Product Price) 44,160원 + 배송비 (Shipping Cost) 2,500원 = 총 결제 예상 금액 (Total Expected Payment) 46,660원. A green '주문 하기' (Place Order) button is located at the bottom right.

At the very bottom of the screenshot, there is a dark footer bar with the text '© TEMBYTEM 2022'.

사용자 장바구니

- 쿠키 기능을 활용한 장바구니 기능 구현

```
def add_cart(request, product_pk):
    if request.method == "POST":
        item_quantity = request.POST.get("buy-mount")
        product = Product.objects.get(pk=product_pk)

        try:
            cart = Cart.objects.get(cart_id=_cart_id(request))
        except Cart.DoesNotExist:
            cart = Cart.objects.create(cart_id=_cart_id(request))
            cart.save()

        try:
            cart_item = CartItem.objects.get(product=product, cart=cart)
            cart_item.quantity += int(item_quantity)
            cart_item.save()
        except CartItem.DoesNotExist:
            cart_item = CartItem.objects.create(product=product, quantity=1, cart=cart)
            cart_item.save()

        item_count = CartItem.objects.get(cart=cart.pk, product=product)
        item_count.quantity = item_quantity
        item_count.save()

    return redirect("cart:cart_detail")
```

프로젝트 TBT

KPT 회고록

- Keep

- 6명이 팀원이 되면서 백엔드와 프론트엔드 인원이 1대 1로 비율이 맞아서 User, Product, Review 파트를 나눠 진행하면서 좀 더 서로 빠른 피드백을 통해 바로 수정을 하면서 각 부분에 맞는 개발을 할 수 있는 점이 재미있었고 프론트엔드를 맡은 팀원 분들이 너무 잘하셔서 초반에 피그마를 완성하고 진행을 해보니 각 페이지에 맞는 기능을 만들기 쉬웠습니다.

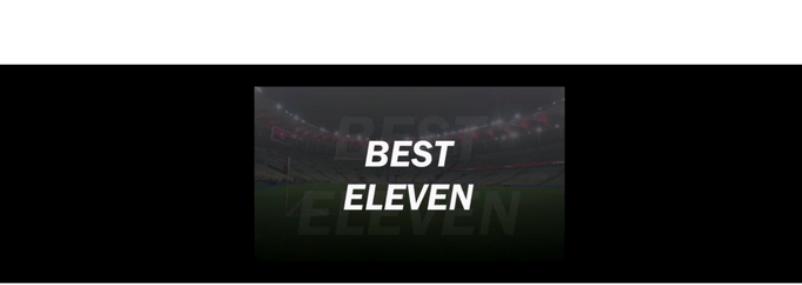
- Problem

- 작업을 하면서 프론트엔드 분들의 요청을 빠르게 처리하고 싶었으나 실력이 차이나는 부분이 발생을 하면서 기능을 구현하는 부분에서 시간이 부족했고 9~6의 작업 시간 별도로 저녁에 좀 더 기능을 구현을 했으며, 시간을 더 투자를 해서 초반에 구성한 피그마 페이지에 맞게 기능을 최대한 적용 할 수 있어서 뿌듯했습니다.

- Try

- 새로운 장바구니 기능을 구현을 하기 위해 자료 검색과 정리하는 부분이 대략 2일 정도 걸리면서 내가 기능을 구현 할 수 있을까 하는 걱정이 됐지만 코드를 작성하고 계속 수정을 해가면서 4일 차에 완성을 할 수 있었고 수량 변경 하는 부분은 도움을 받아서 마무리 할 수 있었습니다.

프로젝트 Best11



대한민국

FIFA 월드컵 2018 | 김석 편집부 번역

역대 월드컵 성적

2022 - 개최 국가: 142회 (승 1무 2패)
2018 - 개최 국가: 142회 (승 1무 2패)
2014 - 개최 국가: 142회 (승 1무 2패)
2010 - 개최 국가: 142회 (승 1무 2패)
2006 - 개최 국가: 142회 (승 1무 1패)
2002 - 개최 대한민국/일본: 42회 (승 2무 2패)

카타르 월드컵 기록

나이트 토토월드컵 대회 기록 | 대회전 보러가기
2022-12-06 대한민국 1:4 코스타리카
2022-12-03 대한민국 2:1 포르투갈
2022-11-28 대한민국 2:1 가나
2022-11-24 대한민국 0:0 우루과이

베스트 일레븐

기량 있는 선수들을 위한 전문적인 플랫폼입니다.

피설피드 경기일정 축구규칙 미니게임

프로필 1
프로필 2
프로필 3
프로필 4
프로필 5
프로필 6

국가대표

2022년 11월 12일 대표팀 | FIFA 카카오 월드컵 명단

ALL	FW	MF	DF	GK
정우영	1999-09-20 180cm / 70kg FC 콘카카페디아	조규성	1998-01-25 180cm / 70kg 전북현대모비스	황희조

베스트 일레븐

BEST11 축구클럽
프로젝트 기획서

간접전 이수경 일간주 주제판 차희경

© 베스트일레븐 2022. All Rights Reserved.

기획 - 월드컵 국가대표 정보를 팬의 정보로 공유하는 서비스

일정 - 2022.11.24 ~ 2022.12.14

기술 -

서비스 URL - [Project Best11](#)

역할 - 국가대표 선수 DB 생성(DB Browser 활용)

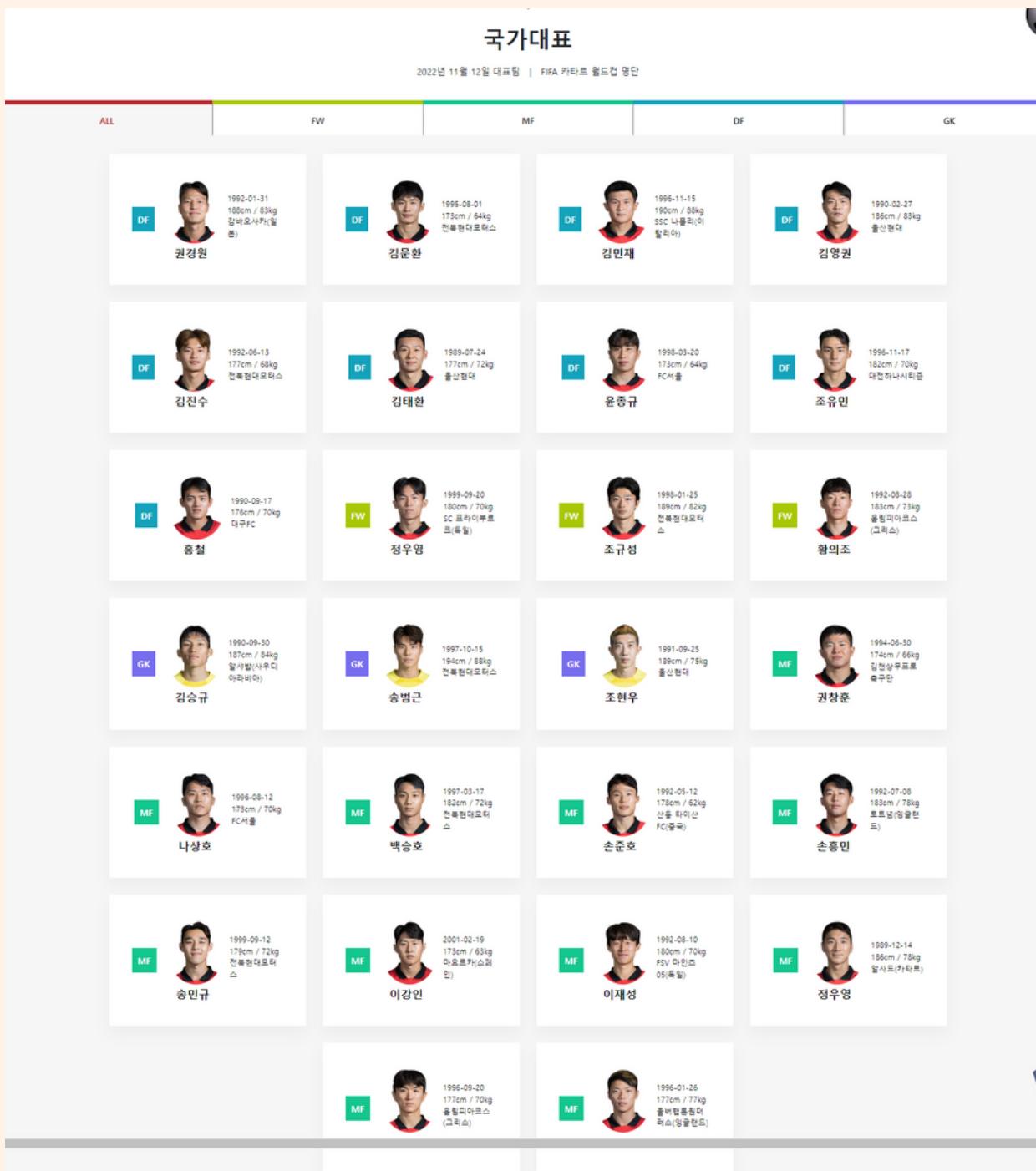
- 응원많은 선수 랭킹 구현(annotation 활용)

- 댓글 좋아요순, 최신순 정렬 구현

- 피셜(댓글) 신고하기 기능 구현(hidden 활용)

- 비동기(선수 응원하기, 댓글 좋아요, 팔로우) 구현

프로젝트 Best11



국가대표 선수단 정보 생성(Create)

- 국가대표 선수단의 정보를 생성하기 위해서 한번만 등록을 하면 별도 수정이나 변동이 적기 때문에 csv 파일로 선수단의 필요한 정보를 생성하고 DB Browser를 통해 직접 서비스 DB에 업로드 하는 방식을 사용 함

프로젝트 Best11

베스트 일레븐

가장 많은 응원을 받은 11명의 선수들입니다



응원하기(좋아요) 많은 순서대로 노출

- 국가대표 선수 중 팬들이 응원하기(좋아요)가 많이 누적된 순서로 노출 될 수 있도록 `annotate().order_by()`를 사용 정렬 하여 값을 넘겨 줌

```
like_players = Players.objects.annotate(like_count=Count("fans")).order_by(
    "-like_count"
)[:11]
```

- 몇 명의 팬이 응원을 하고 있는지 카운트를 해서 순위를 쉽게 알 수 있도록 보여줌

```
{{ player.fans.count }}
```

프로젝트 Best11

21개의 피셜

선수에 대해 알고 있는 정보를 공유해 주세요! 등록

최신순 좋아요순

koreaaaaaa ⚡ 0 | 2022.12.14 | 신고

조규성은 k리그 득점왕입니다.
득점왕답게 가나전에서 멀티골을 기록하였습니다.

soccer ⚡ 8 | 2022.12.02 | 신고

Beemo ⚡ 8 | 2022.12.05 (수정됨) | 수정 | 삭제
헤딩 경합과 위치선정이 좋아요

병만이 ⚡ 4 | 2022.12.02 (수정됨) | 신고
오늘 경기 화이팅....

피셜(댓글)을 최신순/좋아요순 필터링

- 피셜을 쉽게 보기 위한 기능으로 최신순과 좋아요순으로 볼 수 있도록 백엔드 팀원과 함께 구현 하였음

```
<div class='comment-tabs'>
  <input type="radio" name="sort" id="toggle-like" checked="checked"/>
  <label for="toggle-like">
    <span class="text-title" style="font-size: 1.1rem;">좋아요순</span>
  </label>
  <input type="radio" name="sort" id="toggle-date"/>
  <label for="toggle-date">
    <span class="text-title" style="font-size: 1.2rem;">최신순</span>
  </label>
```

- css "comment-tabs"를 활용하여 기능 구현 함

```
.comment-tabs {
  height: 100%;
  font-size: 1.5rem;
}
```

```
.comment-tab {
  height: 100%;
  width: 100%;
  display: none;
  padding: 4rem 1rem 1rem;
}
```

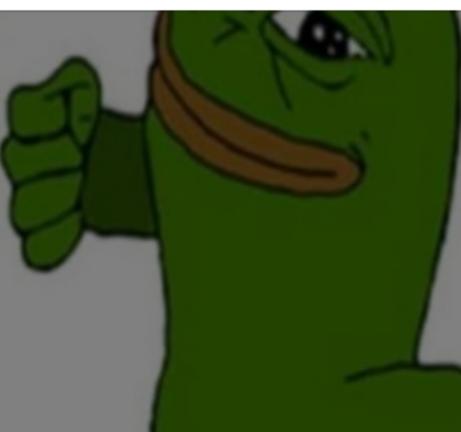
프로젝트 Best11

마리애비 ⚡ 3 | 2022.12.03 | 신고

이전에는 몰랐는데 헤딩경합이 아주좋고 이번 월드컵에서 전방에서 정말 많은 활동량으로 팀에 엄청난 헌신을 한 선수

해당 피셜을 신고하시겠어요?
(신고 후 해당 피셜은 숨김 처리 됩니다.)

[취소](#) [신고하기](#)



kmk4162 ⚡ 3 | 2022.12.02 | 신고

EPL 가보자~~

gustjd000 ⚡ 2 | 2022.12.02 | 신고

잘 생겨따

방배동최수종ㅋㅋ ⚡ 2 | 2022.12.03 | 신고

ㅎㅇ 잘생긴 사람끼리 잘 통할거같네

포르투갈전2골넣을조규성 ⚡ 1 | 2022.12.02 | 신고

마리애비

⚡ 3 | 2022.12.03 | 신고

이전에는 몰랐는데 헤딩경합이 아주좋고 이번 월드컵에서 전방에서 정말 많은 활동량으로 팀에 엄청난 헌신을 한 선수

kmk4162

⚡ 3 | 2022.12.02 | 신고

EPL 가보자~~

gustjd000

⚡ 2 | 2022.12.02 | 신고

잘 생겨따

방배동최수종ㅋㅋ

⚡ 2 | 2022.12.03 | 신고

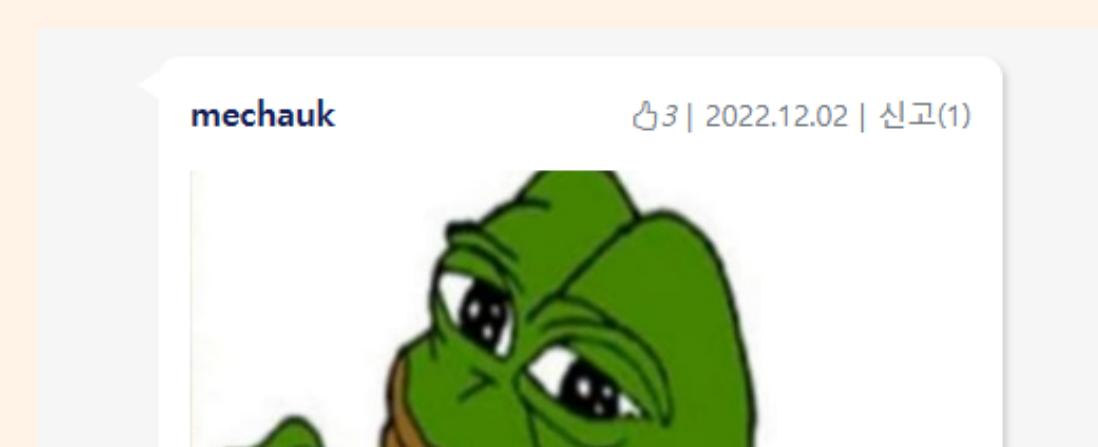
ㅎㅇ 잘생긴 사람끼리 잘 통할거같네

포르투갈전2골넣을조규성

⚡ 1 | 2022.12.02 | 신고

피셜(댓글)의 신고 기능 추가 구현

- 피셜의 내용이 자극적 이거나 사용자가 원하지 않는 내용의 피셜일 경우 신고하기를 클릭 하면 hidden 기능을 통해 숨김 처리 되도록 구현을 하였음
- 신고한 피셜은 나에게만 숨김 처리가 되며 신고하지 않은 사용자에게는 피셜의 신고 개수를 표시 함



프로젝트 Best11

내가 신고한 피셜



총 2개

작성 선수
조규성

뇌피셜 내용
좋아요!!! 헤트트릭 가자!~!!

신고 취소



조규성



신고한 피셜 모아보기 / 삭제

- 향후 상황에 따라서 신고한 피셜을 다시 보거나 신고를 취소해야 할 수 있는 상황을 고려하여 마이페이지 -> 신고한 피셜로 들어오면 모아서 볼 수 있고 바로 취소 번튼을 통해서 신고를 취소 할 수 있도록 구현함
- 피셜의 신고 개수가 5개를 초과하면 모든 사용자에게 노출되지 않도록 if문의 분기 처리를 통해서 구현 함

```
<!-- 피셜 조건문 신고한 피셜이 있으면 안보이도록 -->
{% if block_dict|get_item:comment.pk < 6 or block_dict|get_item:comment.pk == none %}
    <!-- 신고가 5개 초과하면 모든 사용자 숨김처리 -->
    {% if request.user.pk in block_list and comment.pk in block_comment %}
```

프로젝트 Best11

KPT 회고록

- Keep

- 기존의 배운 기능을 기초로 가져가면서 새로운 DB Browser 프로그램을 통한 DB 업로드 방법과 annotate의 order_by + 슬라이싱 되는 부분까지 사용을 해보고, 구글링 하지 않고 JavaScript를 통해 신고하기 기능을 구현을 했다는 점에 많이 뿌듯했고 새로운 프로그램과 기능을 구현하는 재미를 느꼈습니다.

- Problem

- 초반에 응원하기의 버튼 색상과 아이콘 변경 이렇게 두 가지 항목에 대해서 변경하여 효과를 주려고 하였으나 버튼 내부에서 버튼과 아이콘 동시에 값을 변경하는 부분이 안된다는 것을 배웠습니다 그래서 다른 방법으로 수정을 하여 진행을 했습니다.
- 피셜을 분기 처리하면서 코드가 너무 길어지고 중복 해서 발생하는 코드를 어떻게 정리 하는 부분은 공부가 필요하다고 생각 했습니다.

- Try

- 중복 되는 코드를 어떠한 방식으로 줄일 수 있는지 코드를 직접 수정해 보고 다른 방법으로 반복 되는 코드를 구현 할 수 있는지 공부의 필요성을 느꼈습니다.

앞으로의 계획

1. GitHub를 지속적으로 관리한다.
2. BEST11 팀원들과 계속 프로젝트를 발전 시켜 나간다.
3. 현재 진행 중인 알고리즘 공부(문제 풀이) 모임을 꾸준히 진행 한다.
4. 취업 전까지 스터디 팀원들과 토이 프로젝트를 진행 한다.
5. 부족하다고 느끼고 있는 AWS S3, CSS, Python을 추가로 공부한다(Udemy)
6. 공부하는 내용을 기술 블로그(개인 채널)를 개설하여 공부하는 내용을 정리한다.
7. 포트폴리오를 계속 업데이트 시킨다
8. 취업 후에도 1~ 6번을 우선 순위를 두고 업그레이드 한다.