

빅데이터 분석 및 서비스개발 4회차

인터페이스 개발

Angel in Star



Team 천사조



장은영, 김세윤, 김도연, 박세연, 원종현

배경



주제 선정 배경

스타벅스 홈페이지의 다양하고 정돈된 애니메이션 효과를 기반으로 가상의 카페 브랜드 'Angel in Star' 의 웹 페이지 개발



프로젝트 목적

front-end 및 back-end 구현을 통해 인터페이스 개발 및 서버 실행



프로젝트 개요



컨셉

스타벅스 웹사이트를 참고한 새로운
카페 웹사이트 'Angel in Star' 개발



훈련과의 연관성

html, css, js를
이용한 웹페이지 작성
Django를 이용한 서버 실행

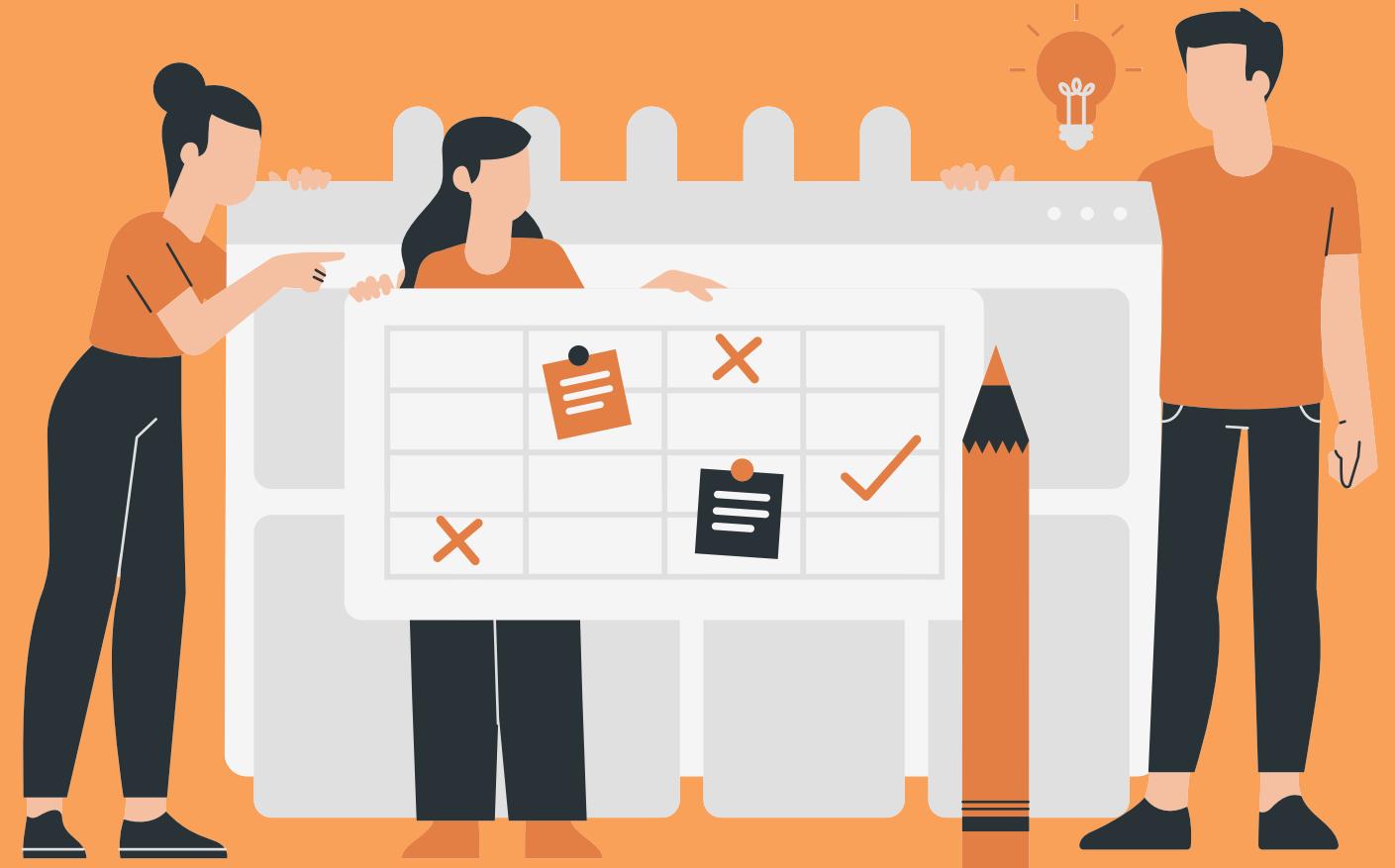


개발 환경

파이썬을 이용한 개발
깃허브를 이용한 팀 협업

프로젝트 기대효과

- ✓ 웹사이트 개발을 위한 업무 수행 숙련도 향상
- ✓ 팀 프로젝트를 통한 협업능력 향상
- ✓ FrontEnd 및 BackEnd 개발 시 업무의 목적 및 진행과정 파악 가능
- ✓ 이론 수업에서 배웠던 내용을 직접 적용해보며 개인 개발능력 향상
- ✓ 프로젝트 과정 중 충돌상황, 오류를 경험하고, 오류 해결능력을 향상



팀 구성 및 역할

팀 구성 및 역할



장은영



products

MENU 상세페이지 스케치 작성
MENU 제품별 상세페이지 관련 html, css, js 작성
Django로 각 제품별 상세페이지 url 연결



김세윤



accounts

로그인, 회원가입 페이지 스케치 작성
로그인, 회원가입 html, css 작성
Django로 각 페이지 url 연결

팀 구성 및 역할



김도연



Store, service, board

이벤트 페이지 스케치 작성
매장 정보(Store), 서비스(Service), 게시판(My Angels) 페이지 html, css, js 구현
Django로 각 페이지 간의 연결 및 DB 생성



박세연



MENU

매장정보 페이지 관련 스케치 작성
홈페이지의 메뉴 페이지 생성 및 관련 html, css, js 구현
Django로 메뉴 페이지의 url 연결



원종현



EVENT

메인 페이지 스케치 작성
메인페이지 html,css,js구현
event 페이지 구현



수행절차 및 방법

스케치파일

미리 스케치파일을 작성해봄으로써
페이지별 구조와 필요한 기능을
파악하여 신속한 개발이 가능했다.



1. 메인페이지

<Header> (Top Menu 고정)
사이트이름(로고형식사진파일) 로그인
메뉴정보 매장정보 고객센터
*****세부메뉴는 숨겨져있음. 메뉴에 마우스를 올리면 세부메뉴가 보인다*****

<배경> 이미지는 고정되어 전체배경을 차지

<slideShow>
이벤트(신메뉴, 할인메뉴 등) 이미지 슬라이드

<메뉴사진과 함께 애니메이션효과>
메뉴 종류1

<메뉴사진과 함께 애니메이션효과>
메뉴 종류2

<메뉴사진과 함께 애니메이션효과>
메뉴 종류3

<메뉴사진과 함께 애니메이션효과>
메뉴 종류

<footer 고정> - 하단 메뉴
하단 서브 메뉴 목록 하단 서브 메뉴 목록 하단 서브 메뉴 목록
- - -
- - -
- - -
- - -

<Footer> (고정)
회사정보소개

개발 일정

미리 개발 일정을 기획하여 계획에 차질없게 체계적으로 진행하였습니다



개발 도구 및 환경

기획



Front-end

HTML



CSS



jQuery
write less, do more.

Back-end



협업



Github

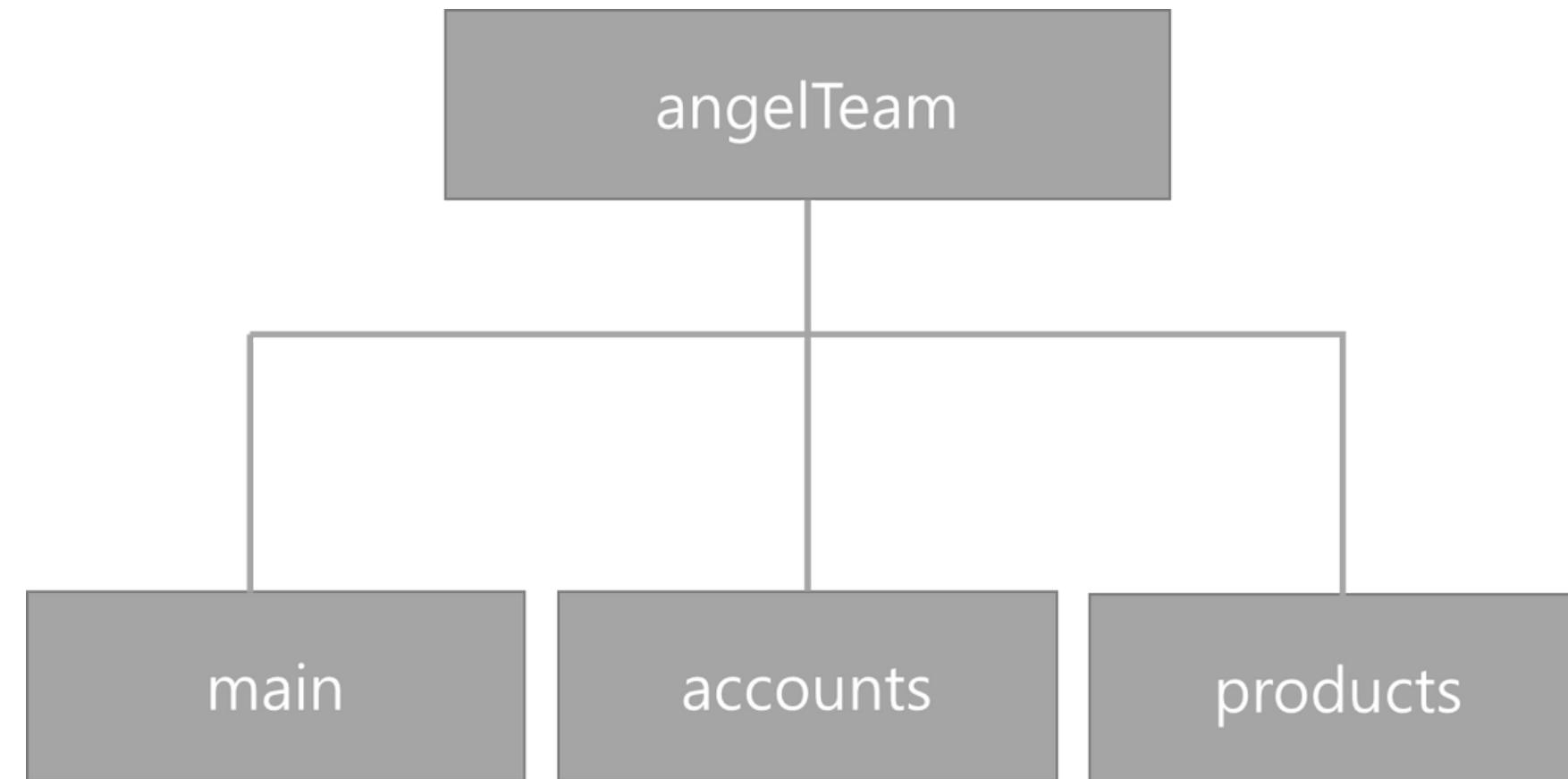
Django 구조도



Django 구조도

Django 프로젝트 angelTeam 내의 3가지 앱으로 구성

- └ angelTeam
 - > .└ angelTeam
 - > .└ main
 - > .└ accounts
 - > .└ products
 - > └ static



Django 구조도

프로젝트의 각 앱에 맞는 html파일을 앱별로 추가함



```
angelTeam
├── angelTeam
│   ├── settings.py
│   └── urls.py
├── main
│   ├── urls.py
│   ├── views.py
│   └── Templates
│       └── main
│           ├── main.html
│           ├── menu.html
│           ├── store.html
│           └── event.html
```

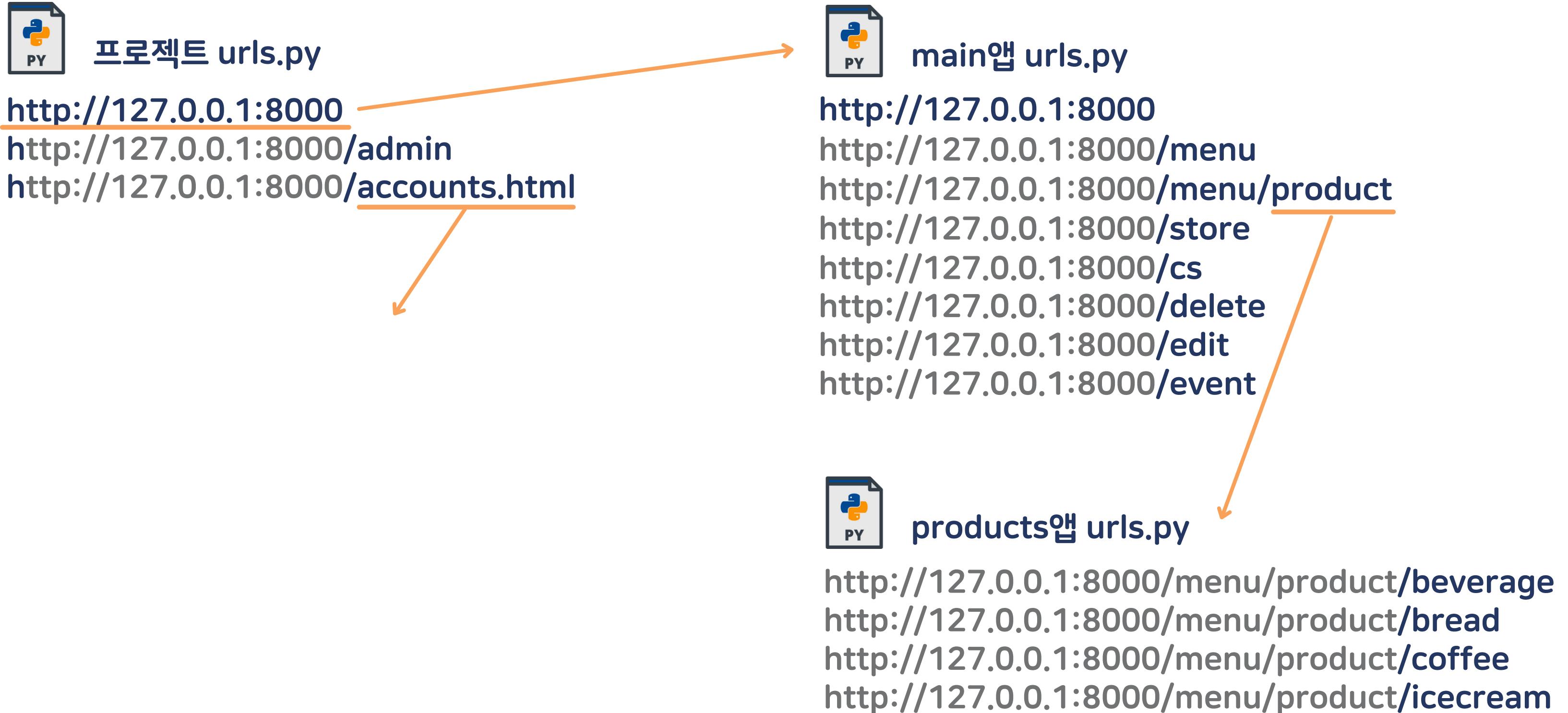
```
products
├── urls.py
└── views.py
└── Templates
    └── products
        ├── beverage
        │   ├── menuDetail_1.html
        │   ├── ...
        │   └── menuDetail_9.html
        ├── bread
        ├── coffee
        └── icecream
```

```
accounts
├── urls.py
└── views.py
└── Templates
    └── accounts
        ├── board.html
        ├── login.html
        ├── signup.html
        ├── update.html
        └── welcome.html
    └── base.html
    └── footer.html
    └── formerrors.html
    └── navbar.html
```

URL 구조도



URL 구조도



프로젝트 urls.py

기본 URL인 `http://127.0.0.1:8000`을 요청하면 main 앱의 `urls.py`에서 처리하도록 연결하고,
기본 URL/ admin, CS, store, accounts, csaccounts 각 URL요청에 알맞게 처리할 수 있도록 연결해줌.

angelTeam\urls.py

```
16  from django.contrib import admin
17  from django.urls import path, include
18
19  from main import views
20
21  urlpatterns = [
22      path('', include('main.urls')),
23      path('admin/', admin.site.urls),
24      path('accounts/', include('accounts.urls'), name='accounts'),
25  ]
```

main 앱

main 앱에서 각 경로별 menu, store, cs, event 페이지를 연결한다.
'menu/product/' url 요청시 products 앱으로 연결한다.

mainUrls.py

```
3  from django.urls import path, include  
4  from . import views  
5  
6  app_name = 'main'  
7  
8  urlpatterns = [  
9      path('', views.main, name="main"),  
10     path('menu/', views.menu, name="menu"),  
11     path('menu/product/', include('products.urls', namespace='products')),  
12     path('store/', views.store, name="store"),  
13     path('cs/', views.cs, name="cs"),  
14     path('cs/', include('accounts.urls'), name="csaccounts"),  
15     path('event/', views.event, name="event"),  
16     path('edit/', include('accounts.urls'), name="edit"),  
17     path('delete', include('accounts.urls'), name="delete"),  
18 ]
```

mainViews.py

```
1  from django.shortcuts import render  
2  from django.http import HttpResponseRedirect  
3  
4  # Create your views here.  
5  
6  def main(request):  
7      return render(request, 'main/main.html')  
8  
9  def menu(request):  
10     return render(request, 'main/menu.html')  
11  
12  def store(request):  
13     return render(request, 'main/store.html')  
14  
15  def cs(request):  
16     return render(request, 'main/CS.html')  
17  
18  def event(request):  
19      return render(request, 'main/event.html')
```

accounts 앱 urls.py

http://127.0.0.1:8000/accounts 요청시 로그인, 로그아웃, 회원가입, 게시판 관련된 URL 구축하고, 각 기능들을 연결해줌

accounts\urls.py

```
1  from django.contrib import admin
2  from django.urls import path, include
3  from django.contrib.auth import views as auth_views
4  from . import views
5
6  app_name = 'accounts'
7
8  urlpatterns = [
9      path('login/', auth_views.LoginView.as_view(template_name='accounts./login.html'), name='login'),
10     path('logout/', auth_views.LogoutView.as_view(), name='logout'),
11     path('signup/', views.signup, name='signup'),
12     path('signup/welcome.html', views.welcome, name='welcome'),
13     path('board.html/', views.board, name="board"),
14     path('board.html/accounts/board.html', views.board, name="acboard"),
15
16     path('<int:pk>', views.boardEdit, name="editboard"),
17     path('accounts/board.html', views.board, name="edboard"),
18     path('/<int:pk>', views.boardDelete, name="delboard"),
19     path('/accounts/board.html', views.board, name="deboard"),
20 ]
```

accounts 앱 views.py

accounts 앱의 urls.py에서 views와 연결된 함수를 구성하여 각 함수별로 기능을 구현한다.

accounts\views.py

```
1  from django.shortcuts import render, redirect
2  from django.contrib.auth import authenticate, login
3  from .forms import UserForm
4  from .forms import *
5  from .models import *
6
7  # Create your views here.
8  def signup(request):
9      if request.method == 'POST':
10          form = UserForm(request.POST)
11          if form.is_valid():
12              form.save()
13              username = form.cleaned_data.get('username')
14              raw_password = form.cleaned_data.get('password1')
15              user = authenticate(username=username, password=raw_password)
16              login(request, user)
17              return redirect('main:main')
18
19      else:
20          form = UserForm() # auth form과 user 모델을 복사해서 만든 사용자계정 클래스
21          # #userform 객체를 사용해서 렌더링할수 있도록 정보를 form이라는 변수를 만들어 저장
22      return render(request, 'accounts/signup.html', {'form':form})
```

```
27  def welcome(request):
28      return render(request, 'accounts/welcome.html')
29
30  # 게시판
31  def board(request):
32      if request.method == 'POST':
33          title = request.POST['title']
34          content = request.POST['content']
35          writer = request.user
36          board = Board(
37              title=title,
38              content=content,
39              writer=writer,
40          )
41          board.save()
42          return redirect('accounts:board')
43      else:
44          boardForm = BoardForm()
45          board = Board.objects.all()
46          context = {
47              'boardForm': boardForm,
48              'board': board,
49          }
50
51      return render(request, 'accounts/board.html', context)
```

product 앱

메뉴 페이지에서 특정 상품 이미지를 클릭하면 해당 상품의 상세정보 페이지로 연결한다.

product\urls.py

```
3   from django.urls import path, include
4   from . import views
5
6   app_name = "products"
7
8   urlpatterns = [
9       path('beverage/1', views.beverage_1, name="beverage1"),
10      path('beverage/2', views.beverage_2, name="beverage2"),
11      path('beverage/3', views.beverage_3, name="beverage3"),
12      path('beverage/4', views.beverage_4, name="beverage4"),
13      path('beverage/5', views.beverage_5, name="beverage5"),
14      path('beverage/6', views.beverage_6, name="beverage6"),
15      path('beverage/7', views.beverage_7, name="beverage7"),
16      path('beverage/8', views.beverage_8, name="beverage8"),
17      path('beverage/9', views.beverage_9, name="beverage9"),
18
19      path('bread/1', views.bread_1, name="bread1"),
20      path('bread/2', views.bread_2, name="bread2"),
21      path('bread/3', views.bread_3, name="bread3"),
22      path('bread/4', views.bread_4, name="bread4"),
23      path('bread/5', views.bread_5, name="bread5"),
```

product\views.py

```
1   from django.shortcuts import render
2
3   # Create your views here.
4
5   # beverage
6   def beverage_1(request):
7       return render(request, 'products/Beverage/menuDetail_1.html')
8
9   def beverage_2(request):
10      return render(request, 'products/Beverage/menuDetail_2.html')
11
12  def beverage_3(request):
13      return render(request, 'products/Beverage/menuDetail_3.html')
14
15  def beverage_4(request):
16      return render(request, 'products/Beverage/menuDetail_4.html')
17
18  def beverage_5(request):
19      return render(request, 'products/Beverage/menuDetail_5.html')
20
21  def beverage_6(request):
22      return render(request, 'products/Beverage/menuDetail_6.html')
```

Static 파일



Static 파일

Static 파일은 크게 CSS, js, image 폴더로 구성됨

스케치에서 구상한 대로 메인, 이벤트, 메뉴, 메뉴 상세정보, 회원가입, 로그인, 매장 정보, 고객센터, 게시판 페이지로 구성함



MAIN 페이지

slide box에 있는 이미지들이 시간이 지나면서 순서대로 나타난다.
양쪽에 있는 next, prev 버튼을 누르면 양방향으로 슬라이드가 움직인다.



slide.js

```
1 //slide-wrap
2 var slideWrapper = document.getElementById('slider-wrap');
3 //current slideIndexition
4 var slideIndex = 0;
5 //items
6 var slides = document.querySelectorAll('#slider-wrap ul li');
7 //number of slides
8 var totalSlides = slides.length;
9 //get the slide width
10 var sliderWidth = slideWrapper.clientWidth;
11 //set width of items
12 slides.forEach(function (element) {
13   element.style.width = sliderWidth + 'px';
14 })
15 //set width to be 'x' times the number of slides
16 var slider = document.querySelector('#slider-wrap ul#slider');
17 slider.style.width = sliderWidth * totalSlides + 'px';
18
19 // next, prev
20 var nextBtn = document.getElementById('next');
21 var prevBtn = document.getElementById('previous');
22 nextBtn.addEventListener('click', function () {
23   plusSlides(1);
24 });
25 prevBtn.addEventListener('click', function () {
26   plusSlides(-1);
27 });
28
29 // hover
30 slideWrapper.addEventListener('mouseover', function () {
31   this.classList.add('active');
32   clearInterval(autoSlider);
33 });
34 slideWrapper.addEventListener('mouseleave', function () {
35   this.classList.remove('active');
36   autoSlider = setInterval(function () {
37     plusSlides(1);
38   }, 3000);
39 });
40
41 function plusSlides(n) {
42   showSlides(slideIndex += n);
43 }
44
45 function currentSlides(n) {
46   showSlides(slideIndex = n);
47 }
```

MAIN 페이지

처음 메인페이지 접속 시 텍스트가 움직이며 나타나고, 마우스를 클릭하면 해당 관련 페이지로 연결되어 있다.



Event 페이지

메인 페이지의 이벤트 사진 또는 텍스트 클릭 시 이벤트 페이지로 이동하게 된다

이벤트사진에 마우스를 올렸을 때 반응을 볼 수 있게끔 hover효과를 적용함.

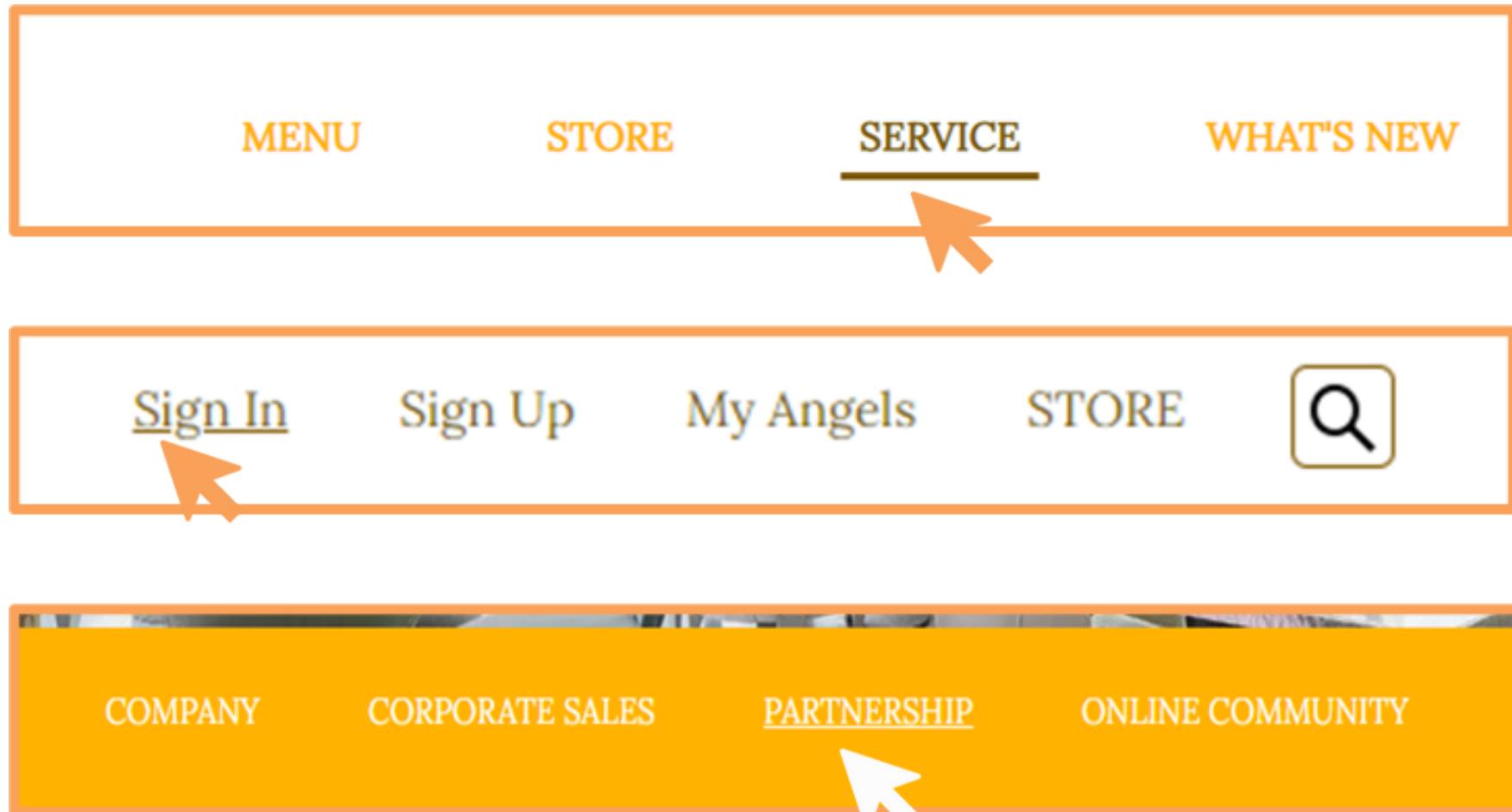
The screenshot illustrates the user interface of the Angel in Star website. On the left, the main homepage features a large pink banner with three event photos: a green iced coffee, a chocolate cake slice, and a floral arrangement. A teal button in the center of the banner says "이 달의 이벤트를 확인하세요!" (Check out this month's events!). An orange arrow points from this button to the top of the right-hand sidebar. The sidebar is titled "WHAT'S NEW" and contains three event cards:

- 번호 3**: **CHOOSE YOUR OWN DRINK** (2022.02.22) - Shows coffee beans and three cups labeled AIS with different drink options: MATCHA LATTE, FLAT WHITE, and ICED COCHA.
- 번호 2**: **월간 구독 서비스** (2022.02.15) - Shows a cup with a sunburst logo and text: "20잔 아메리카노 구독권 15,000원 하루 한 잔 750원의 행복". Below it, a note reads: "Angel in Star의 다양한 상품을 이제 구독 서비스로 만나보세요."
- 번호 1**: **KING SIZE** (2022.02.18) - Shows a large iced coffee with a height scale indicating it's "KING SIZE" compared to a standard "LARGE" cup.

Header , Footer

각 메뉴별 영역에 마우스를 올렸을 때 hover 넣어 해당 반응을 볼 수 있음

header 메인메뉴박스에서는 ::after효과를 텍스트 아래에 밑줄이 중앙에서부터 퍼지는
transform, transition 효과를 적용하여 색다른 반응을 볼 수 있음



common.css

```
101 #mainMenuBox a {  
102     color:#FFB300;  
103     font-weight:bold;  
104     text-decoration:none;  
105     font-size:21px;  
106     position:relative;  
107     padding:10px;  
108 }  
109  
110 #mainMenuBox a::after {  
111     content: " ";  
112     position:absolute;  
113     z-index:5;  
114     bottom:0;  
115     left:50%;  
116     transform:translateX(-50%);  
117     width:0;  
118     height:4px;  
119     background: #805900;  
120     transition: all .5s ease-out;  
121 }  
122  
123 #mainMenuBox a:hover::after {  
124     width:100%;  
125 }  
126  
127 #menuItem a:hover{ color:#805900; }  
128 }
```

Menu 페이지

상품 이미지에 마우스를 올리면 상품 이미지의 크기가 이미지 틀 내에서 서서히 커지는 효과 적용
overflow : hidden 적용함으로써 영역에 초과되는 이미지는 화면에서 보이지 않는다



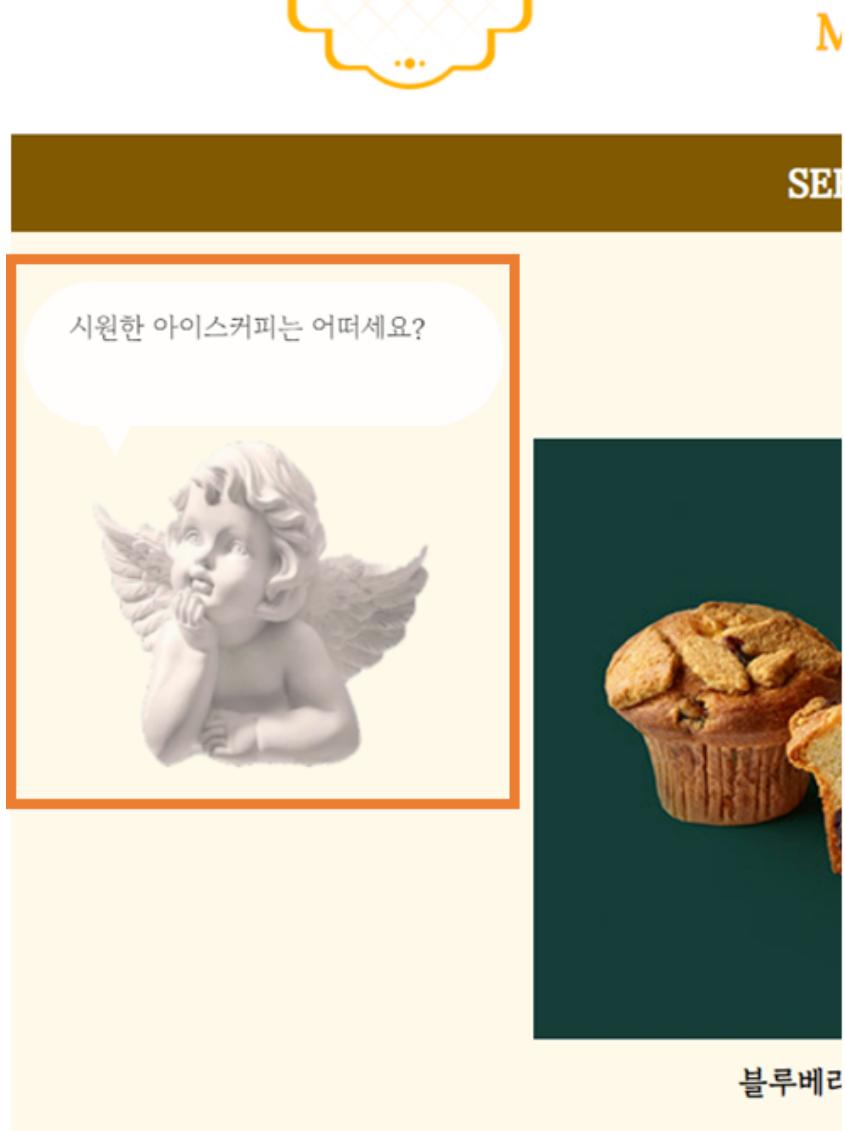
Menu.css

```
72  /*상품*/
73  .product div{
74    display:inline-block;
75    margin:15px;
76    height:250px;
77    width:250px;
78    overflow:hidden;
79    margin-bottom:5px;
80  }

108 .product img:hover{
109   transform: scale(1.1, 1.1);
110   transition:transform 1s;
111 }
```

Menu 페이지

메뉴 페이지 전환 10초 후 왼쪽 상단에 메뉴를 추천해주는 메시지가 서서히 나타남
스크롤 시 함께 이동하며 10초 간격으로 메시지의 내용이 랜덤으로 바뀜



Menu.css

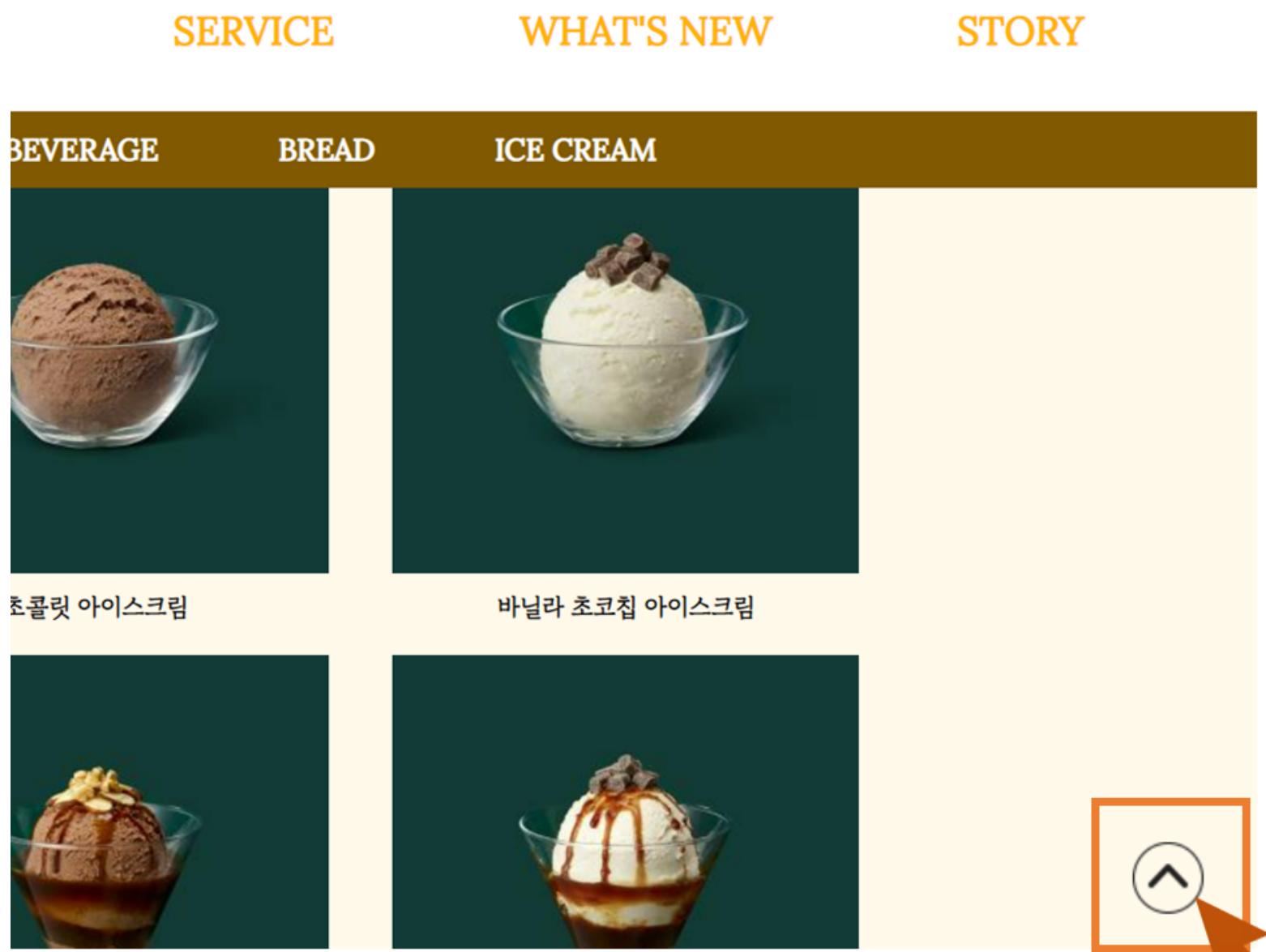
```
113  /*말풍선 css: 메뉴 추천*/
114  .bubble
115  {
116      /*기본설정*/
117      width: 205px;
118      height: 60px;
119      padding: 0px;
120      background: #FFFFFF;
121      -webkit-border-radius: 30px;
122      -moz-border-radius: 30px;
123      border-radius: 30px;
124      /*control*/
125      float:left;
126      position:fixed;
127      z-index:20;
128      opacity:0.9;
129      margin-left:5px;
130      margin-top:40px;
131      font-size:9pt;
132      text-align:left;
133 }
```

Menu.js

```
63  /*말풍선*/
64  $(document).ready(function(){
65      $('.bubble').hide();
66      setInterval(function(){
67          $('.bubble').fadeIn(3000);
68          var msgBox=new Array();
69          msgBox[0] ="시원한 아이스커피는 어떠세요?";
70          msgBox[1] ="동남아가 생각나는 콜드브루를 추천해요!";
71          msgBox[2] ="향긋한 카라멜 마키야또는 어떠신가요?";
72          msgBox[3] ="Angel in Star만의 에스프레소를 맛보세요";
73          msgBox[4] ="오랜만에 상큼한 망고 블렌디드 어떤가요?";
74          msgBox[5] ="인기메뉴 유자 티를 추천해요!";
75          msgBox[6] ="오랜만에 따뜻한 카푸치노가 좋겠군요!";
76          msgBox[7] ="신메뉴 블루베리 머핀을 맛보세요!";
77          msgBox[8] ="오늘의 디저트로 치즈볼을 추천드려요";
78          msgBox[9] ="달콤한 바닐라 초코칩 아포가토는 어떤가요?";
79          msgBox[10] ="시즌한정메뉴 딸기요거트도 있어요";
80
81          var random_num = Math.floor(Math.random()*(msgBox.length));
82          var original_msg = document.getElementById("msg");
83          original_msg.textContent = (msgBox[random_num]);
84      },10000);
85  });
```

전체 페이지 스크롤

화면 최상단에서는 TOP버튼이 나타나지 않고 스크롤 시 메뉴 상단으로 이동하는 move to Top 버튼이 나타남
TOP버튼 클릭 시 화면 맨 위로 이동하고, 화면이 가장 상단에 위치할 경우 버튼이 서서히 사라짐



Menu.html

```
| 241 | <div id="moveToTopBox">
|      
|      id="moveToTop"></div>
```

Common.css

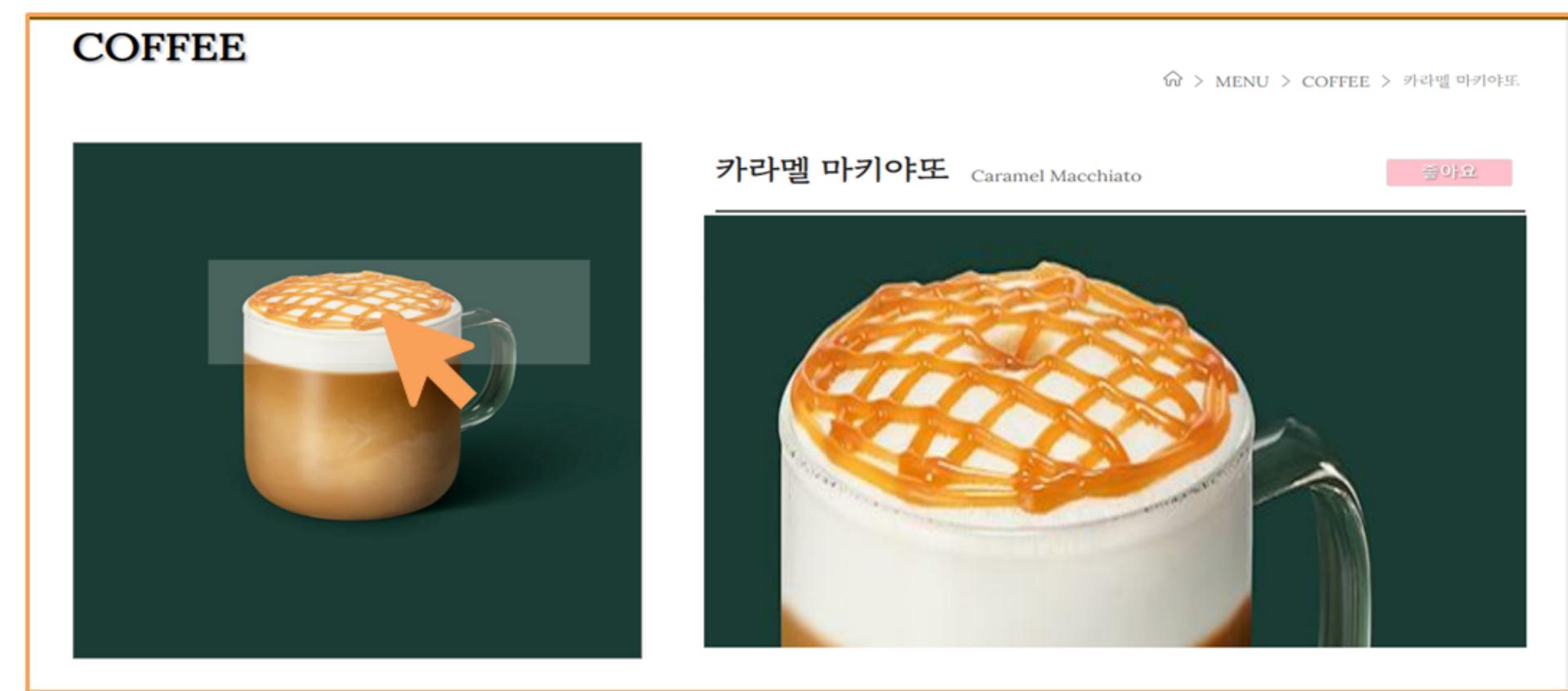
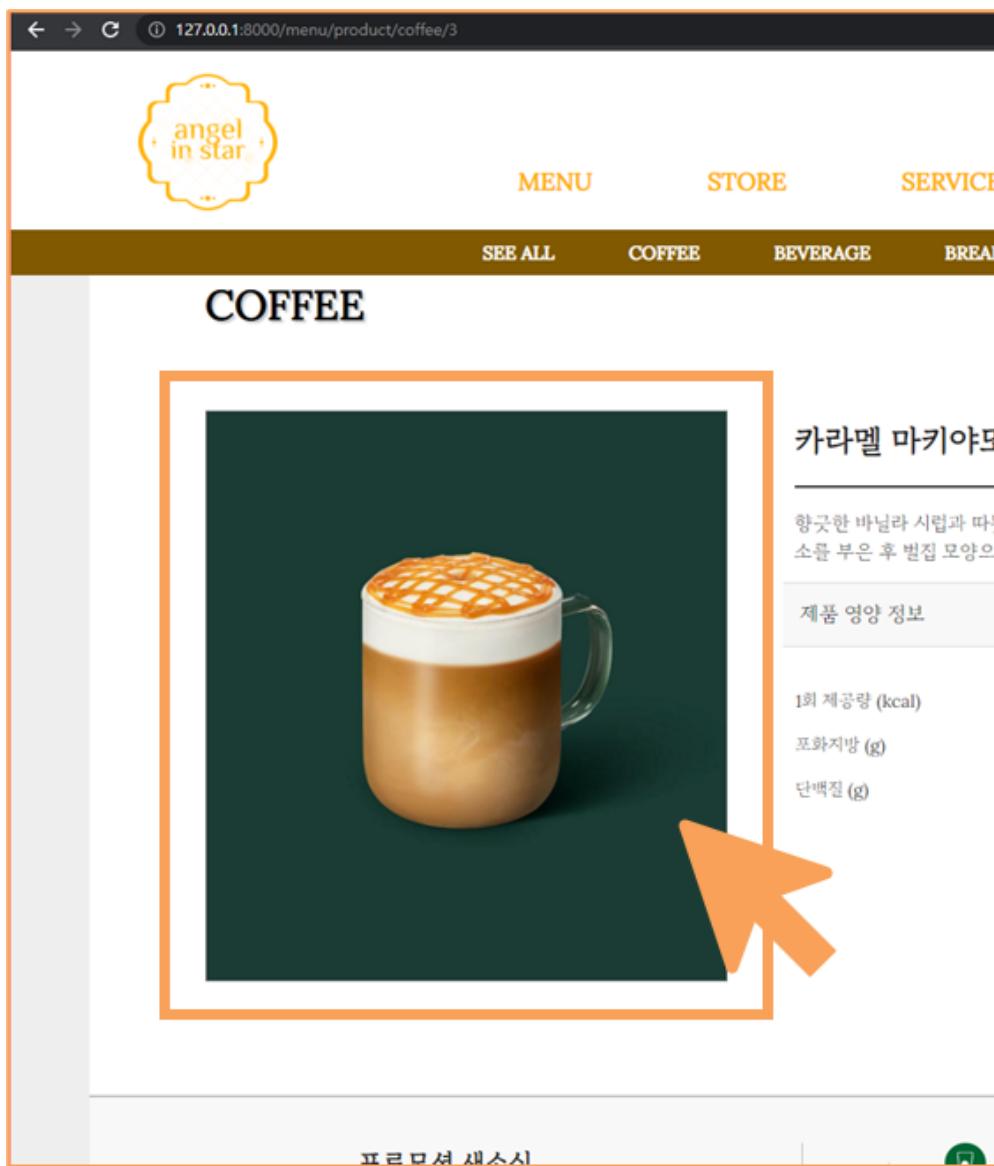
```
168 #moveToTopBox {
169     position:fixed;
170     width:38px;
171     height:38px;
172     bottom:25px;
173     left:94%;
174     border-radius:50%;
175     cursor:pointer;
176 }
177
178 #moveToTopBox img {
179     width:100%;
180     height:100%;
181 }
182
183 #moveToTopBox:hover {
184     transition:0.1s;
185     transform: scale(1.05);
186 }
```

index.js

```
21 $(document).ready(function() {
22     $("#moveToTopBox").hide(); //버튼 최상단에서는 출력하지 않음
23
24     $(window).on('scroll',function(){
25         if($(this).scrollTop() > 500){ // 스크롤탑이 1000보다 클 때
26             $('#moveToTopBox').fadeIn(); //버튼이 fadeIn되면 출력
27         } else { // 1000보다 작을 때
28             $('#moveToTopBox').fadeOut(); //버튼 fadeOut되면 제거
29         }
30     });
31
32     //moveToTop 이미지 클릭시 화면 맨 위로 이동
33     $('#moveToTopBox').on('click',function(){
34         $('html, body').animate({scrollTop:0},500);
35     });
36 });
```

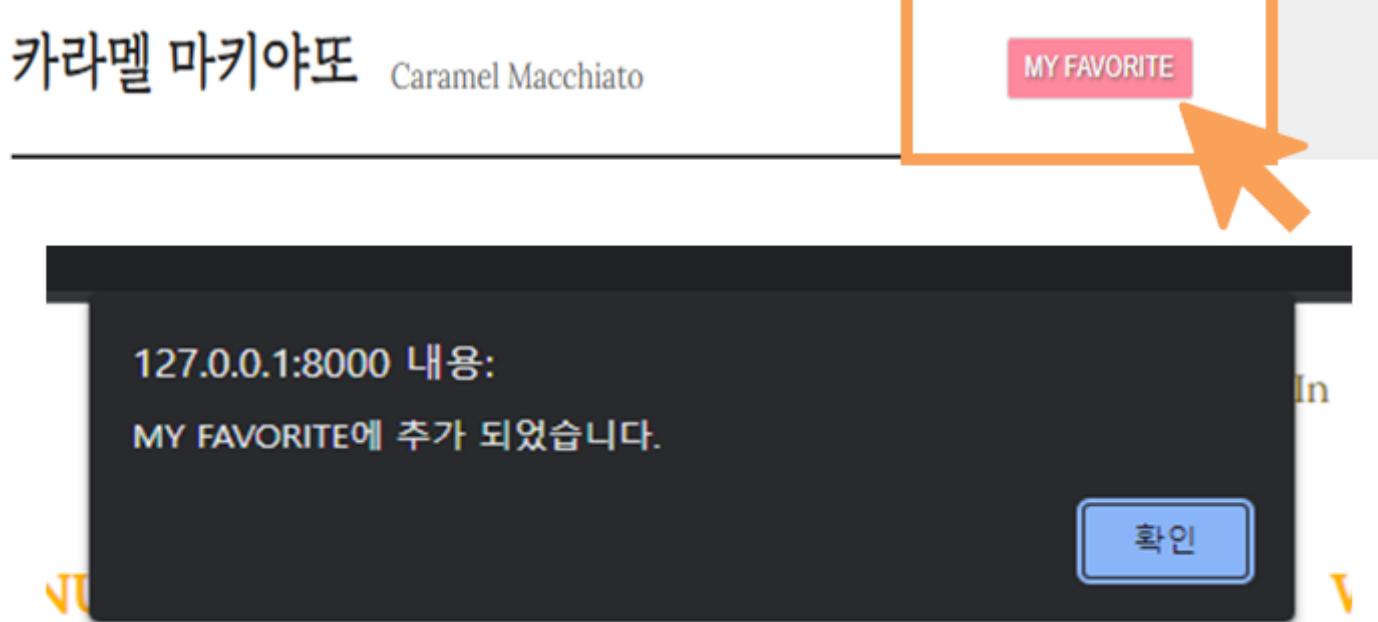
Menu 제품 상세 페이지

제품사진에 마우스를 올리면 마우스기준 영역표시가 나오고 해당 영역에 확대된 제품 사진을 볼 수 있다



Menu 제품 상세 페이지

좋아하는 제품을 선택할 수 있도록 “좋아요” 버튼을 만들고 눌렀을 때
“MY FAVORITE”에 추가되었다는 알림이 나타나고, “좋아요”버튼이 “MY FAVORITE”로 변경된다.

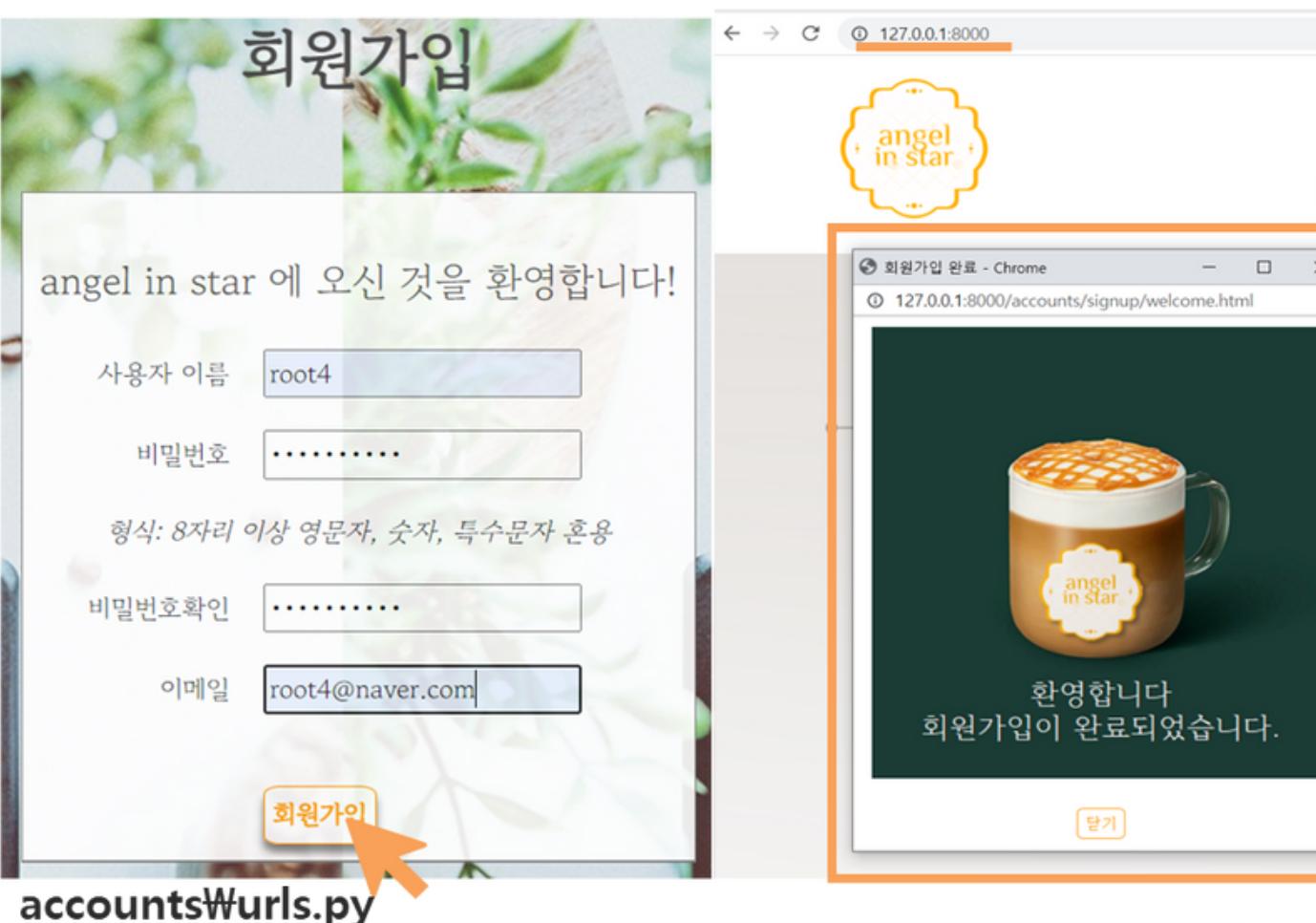


menuDetail.js

```
$document.ready(function () {
    $("button[id='mylike_N']").click(function () {
        if($(this).text() == '좋아요') {
            $('#mylike_N').text('MY FAVORITE').css('font-size','12px');
            $('#mylike_N').css('background','#FE889D');
            alert('MY FAVORITE에 추가 되었습니다.');
        } else {
            $('#mylike_N').text('좋아요').css('font-size','13px');
            $('#mylike_N').css('background','pink');
            alert('MY FAVORITE에서 제외되었습니다.');
        }
    });
});
```

회원가입 페이지

회원가입 버튼을 누르면 팝업창이 뜨면서 메인페이지로 돌아가고,
127.0.0.1:8000/admin 에서 회원가입한 사용자 목록을 확인할 수 있다.



accounts\urls.py

```
6 app_name = 'accounts'  
7  
8 urlpatterns = [  
12     path('signup/welcome.html', views.welcome, name='welcome'),
```

accounts\views.py

```
27 def welcome(request):  
28     return render(request, 'accounts/welcome.html')
```

USERNAME	EMAIL ADDRESS
root	root@google.com
root1	root1@naver.com
root2	root2@naver.com
root3	root3@naver.com
root4	root4@naver.com

angelTeams\urls.py

```
21 urlpatterns = [  
23     path('admin/', admin.site.urls),
```

로그인 페이지

회원가입 후 로그인을 하면 메인페이지로 돌아가며, 오입력 또는 미입력시 에러문구가 나타난다.

로그인

angel in star 에 오신 것을 환영합니다!

사용자 이름

비밀번호

① 127.0.0.1:8000



root1님, 반갑습니다.(Logout)

settings.py

```
128 #로그인 성공 후 이동하는 URL  
129 LOGIN_REDIRECT_URL = '/'  
130 #로그아웃 성공 후 이동하는 URL  
131 LOGOUT_REDIRECT_URL = '/'
```

로그인

Please enter a correct username and password. Note that both fields may be case-sensitive.

angel in star 에 오신 것을 환영합니다!

사용자 이름

비밀번호

login.html

```
9     {% csrf_token %}  
10    {% include "form_errors.html" %}
```

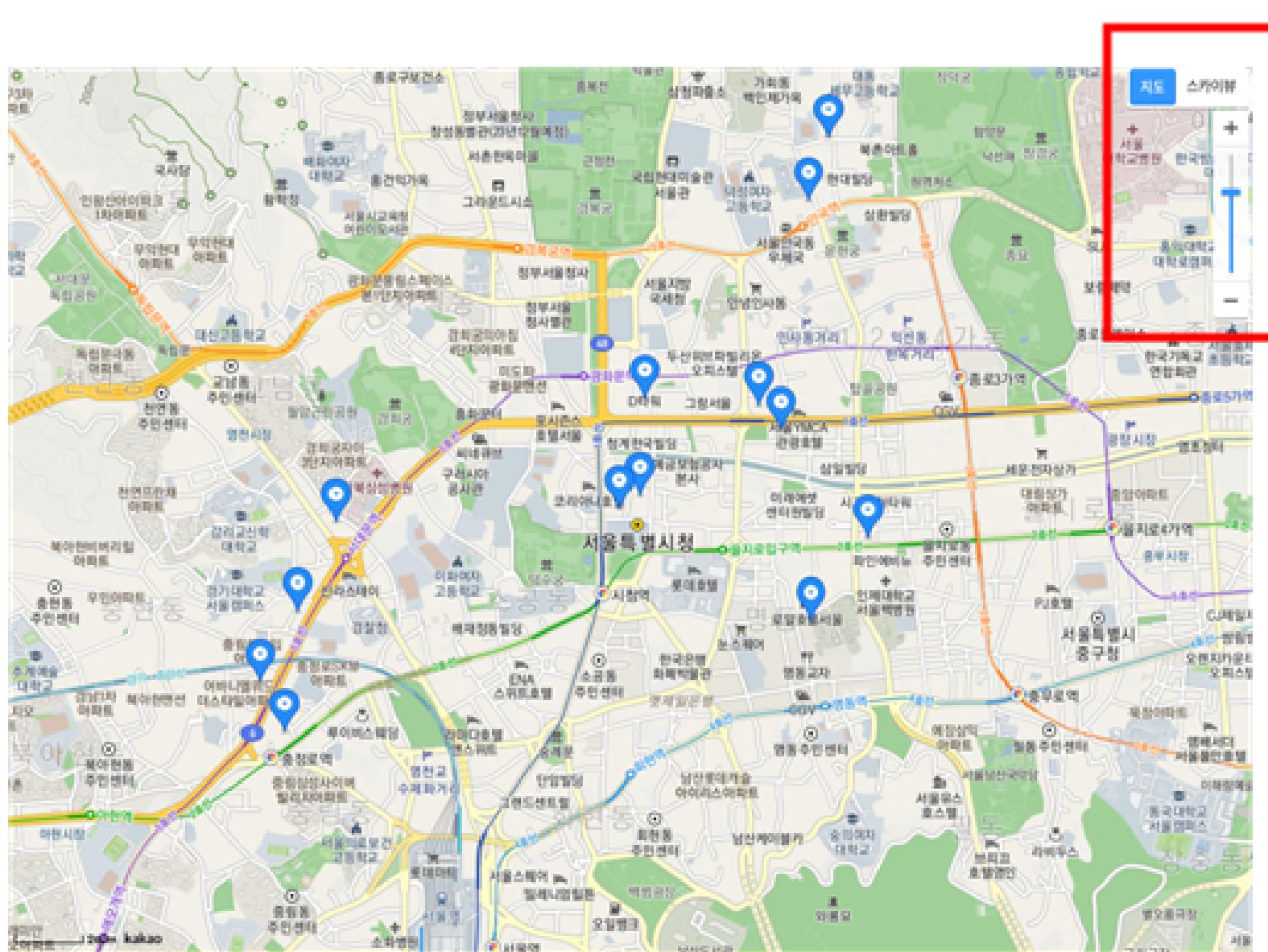
form_errors.html

```
3      {% for error in field.errors %} <!-- 필드 오류를 출력한다. -->  
4          <div class="alert alert-warning">  
10         {% for error in form.non_field_errors %} <!-- 네필드 오류를 출력한다.  
11             <div class="alert alert-warning">
```

Store 페이지

카카오맵 api 를 이용하여 사용자의 조작에 반응하는 반응형 지도 삽입

```
133      <!-- 카카오맵 띄우기 -->
134      <script type="text/javascript"
135          src="//dapi.kakao.com/v2/maps/sdk.js?appkey=18424f3734601e27c03e74e4e2d5c049">
136      </script>
```



지도 확대 및 축소 / 지도 타입 변경 가능하게 구현

```
var map = new kakao.maps.Map(container, options);

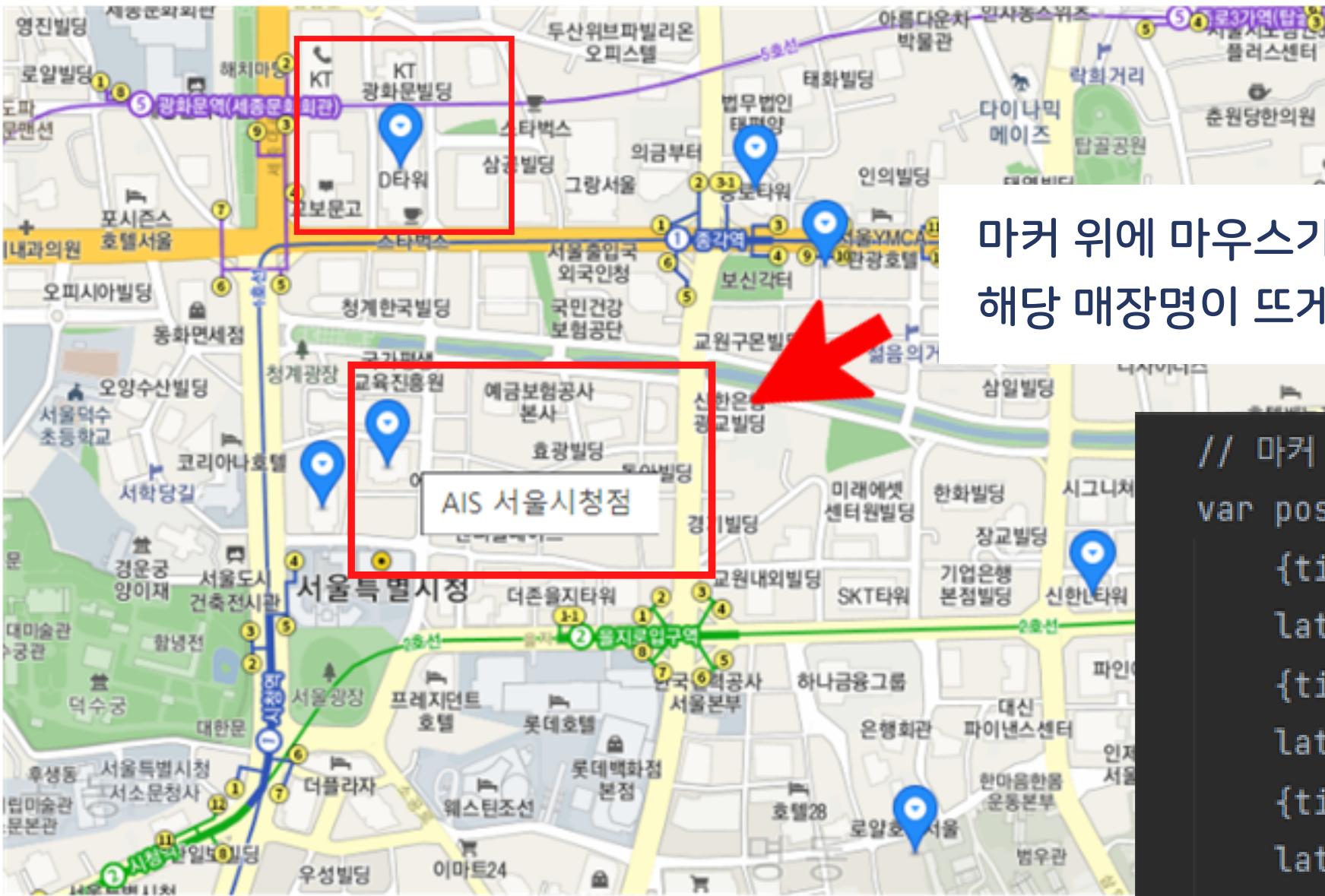
// 지도타입 컨트롤 생성
var mapTypeControl = new kakao.maps.MapTypeControl();

// 지도에 컨트롤 추가
map.addControl(mapTypeControl, kakao.maps.ControlPosition.TOPRIGHT);

// 줌 컨트롤 생성
var zoomControl = new kakao.maps.ZoomControl();
map.addControl(zoomControl, kakao.maps.ControlPosition.RIGHT);
```

Store 페이지

실제 스타벅스 매장의 주소를 차용하여,
가상의 Angel In Star 매장을 만들고 해당 위치마다 마커를 띄움



마커 위에 마우스가 올라가면
해당 매장명이 뜨게끔 js 설정

```
// 마커 표시할 위치
var positions = [
    {title: 'AIS 종각점',
     latlng: new kakao.maps.LatLng(37.5699578826778, 126.984542274763)},
    {title: 'AIS 더종로점',
     latlng: new kakao.maps.LatLng(37.5707451369668, 126.983611410428)},
    {title: 'AIS 광화문점',
     latlng: new kakao.maps.LatLng(37.5709958592808, 126.978914477333)},
    {title: 'AIS 서울시청점',
     latlng: new kakao.maps.LatLng(37.5678548680217, 126.978742292002)},
```

Store 페이지

table 객체를 생성하여

각 매장의 매장명, 주소, 전화번호가 담긴 목록 생성

매장명 :	
매장명	주소
AIS 종각점	서울특별시 종로구 종로 64 (종로2 가) 1522-3232
AIS 더종로점	서울특별시 종로구 종로 51 (종로2 가) 1522-3232
AIS 광화문점	서울특별시 종로구 종로3길 17 (청 진동) D타워 1522-3232
AIS 사청점	서울특별시 중구 을지로 19, 삼성화 재빌딩 1522-3232
AIS 안국점	서울특별시 종로구 북촌로 5-1 1522-3232
AIS 북촌로점	서울특별시 종로구 북촌로4길 23 1522-3232
AIS 명동중앙점	서울특별시 중구 명동길 60 1522-3232
AIS 을지로2가점	서울특별시 중구 삼일대로 358 1522-3232
AIS 서대문역점	서울특별시 종로구 통일로 134 1522-3232
AIS 경기대점	서울특별시 서대문구 충정로 53 1522-3232

매장명 : 종

The screenshot shows a search input field containing the character '종'. A dropdown menu is open below it, displaying two suggestions: 'AIS 종각점' and 'AIS 더종로점'. Both suggestions are followed by their address and phone number.

매장명	주소
AIS 종각점	서울특별시 종로구 종로 64 (종로2 가) 1522-3232
AIS 더종로점	서울특별시 종로구 종로 51 (종로2 가) 1522-3232

매장명 : 종각

The screenshot shows the search results for '종각'. The entry 'AIS 종각점' is highlighted with a red arrow pointing from the previous screenshot. The table displays the store name, address, and phone number for this location.

매장명	주소
AIS 종각점	서울특별시 종로구 종로 64 (종로2 가) 1522-3232

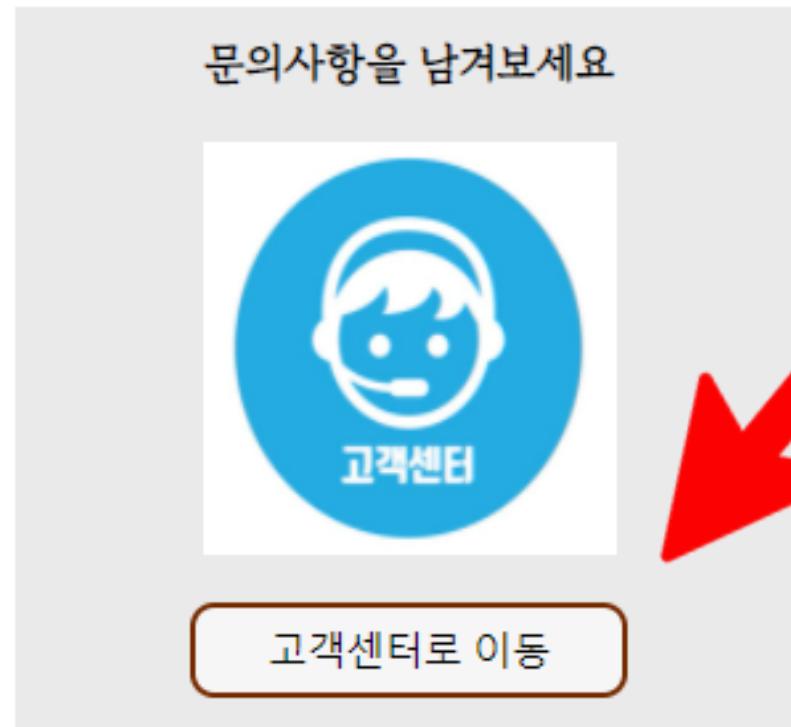
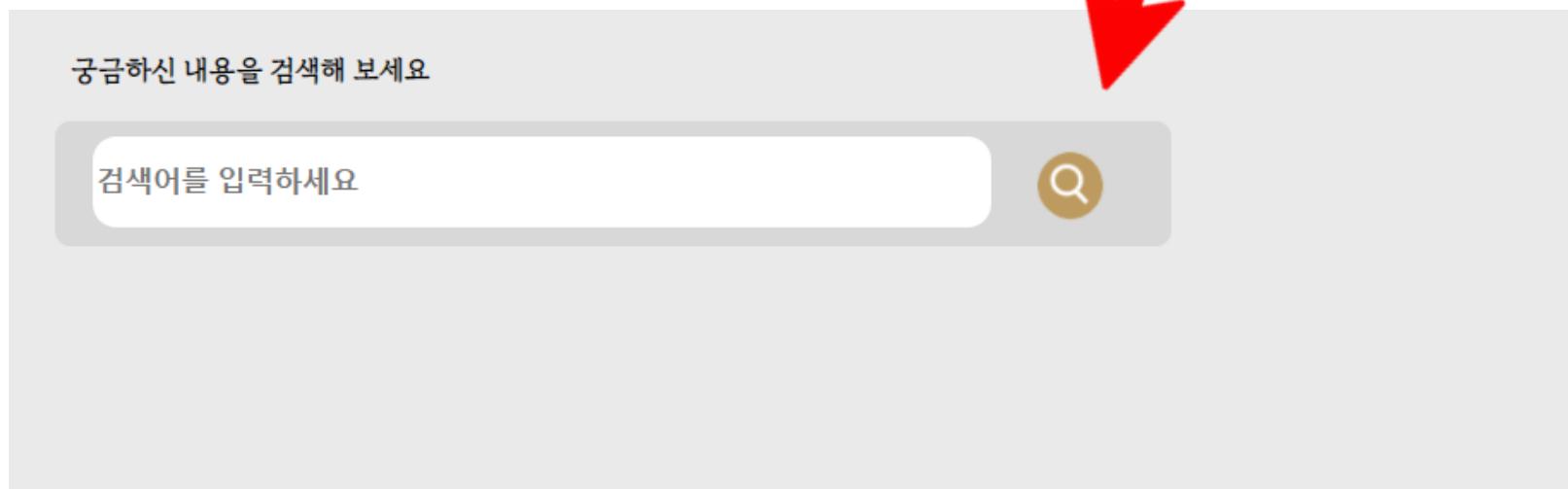
```
18
19 $(document).ready(function() {
20   $("#keyword").keyup(function() {
21     var k = $(this).val();
22     $("#user-table > tbody > tr").hide();
23     var temp = $("#user-table > tbody > tr:contains('" + k + "')");
24
25     $(temp).parent().show();
26   })
27 })
```

keyup.js 를 통해 사용자가 입력하는 글자가 포함
된 매장명만 검색되게끔 구현

Service 페이지

answer.js 를 이용하여

사용자가 검색하고자 하는 검색어를 입력하면, 답변이 뜨는 형식으로 구현



Service 페이지에서
게시판 기능으로 바로 넘어갈 수 있는 버튼 생성 후 url 연결

```
<div id="moveToCommu">
    <h2>문의사항을 남겨보세요</h2>
    
    <button id="moveToCommunity" onclick="location.href='board.html'">고객센터로 이동</button>
</div>
```

Board 페이지

사용자가 직접 불편/불만 사항을 남길 수 있는 board 페이지 구현

-> 로그인을 한 사용자에게만 글 쓰기/수정/삭제 권한을 부여하고, 그렇지 않은 경우에는 로그인 페이지로 연결됨

The screenshot shows a simple web form with a red border around the input fields. It has two text input fields labeled 'Title' and 'Content'. Below them is a blue 'POST' button.

```
# Create your models here.
class Board(models.Model):
    title = models.CharField(max_length=20, null=True)
    content = models.TextField()
    writer = models.ForeignKey(User, on_delete=models.DO_NOTHING, null=True)
```

제목, 내용만 사용자가 작성할 수 있게 하고,
작성자는 로그인 한 아이디로 받아와 저장

번호	제목	내용	작성자	수정하기	삭제하기
1	상품권 관련	문의 드립니다.	user	<button>수정</button>	<button>삭제</button>
2	매장 직원	어제 AIS 광화문점을 방문했는데, 직원 용대에 실망했습니다. 교육 부탁드려요	rla04050	<button>수정</button>	<button>삭제</button>
3	DT 매장	서울 내 DT 매장 있나요??	goc1118	<button>수정</button>	<button>삭제</button>

Board 페이지

작성한 글을 수정하기 위해 '수정' 버튼을 클릭하면,
다시 글을 작성할 수 있도록 url 연결

The screenshot shows a web application interface for a 'Board' page. At the top, there is a navigation bar with links for 'Logout' (goc1118님, 반갑습니다.), 'Sign Up', 'My Angels', 'STORE', and a search icon. Below the navigation is a table listing three posts:

번호	제목	내용	작성자	수정하기	삭제하기
1	상품권 관련	문의 드립니다.	user	수정	삭제
2	매장 직원	어제 AIS 광화문점을 방문했는데, 직원 응대에 실망했습니다. 교육 부탁드려요	rla04050	수정	삭제
3	DT 매장	서울 내 DT 매장 있나요??	goc1118	수정	삭제

A red arrow points from the '수정' (Edit) button in the third post's row to a modal window on the right side of the screen. The modal has fields for 'Title' (Title) containing '매장 관련 문의' and 'Content' (Content) containing '서울 내에 DT 매장 있나요??'. A second red arrow points from the '수정하기' (Edit) button at the bottom right of the modal back to the '수정' button in the post list.

Board 페이지

```
{% extends 'base.html' %}  
{% load static %}  
{% load bootstrap4 %}  
{% block content %}  
  
<link rel="stylesheet" type="text/css" href="{% static 'css/bootstrap.min.css'%}">  
<div class="container" style="max-width:500px;">  
  <form method="POST">  
    {% csrf_token %}  
    {% bootstrap_form boardForm %}  
    <input class="btn btn-dark rounded-pill col-12" type="submit"  
          style="background-color:#EAEAEA; margin-bottom:30px;"  
          value="수정하기" onclick="location.href='/edit/{{board.pk}}/' />  
  </form>  
</div>  
{% endblock %}
```

'수정' 구현 코드

accounts > views.py

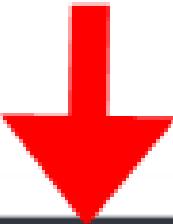
Templates > accounts > update.html

```
def boardEdit(request, pk):  
    board = Board.objects.get(id=pk)  
    if request.method == "POST":  
        board.title = request.POST['title']  
        board.content = request.POST['content']  
        board.writer = request.user  
  
        board.save()  
        return redirect('accounts/board.html')  
  
    else:  
        boardForm = BoardForm  
        return render(request, 'accounts/update.html', {'boardForm':boardForm})
```

Board 페이지

작성한 글을 삭제하기 위해 '삭제' 버튼을 누르면,
해당 글을 삭제하고, 남은 게시글만 목록에 남게끔 구현

번호	제목	내용	작성자	수정하기	삭제하기
1	상품권 관련	문의 드립니다.	user	수정	삭제
2	매장 직원	어제 AIS 광화문점을 방문했는데, 직원 응대에 실망했습니다. 교육 부탁드려요	rla04050	수정	삭제
3	매장 관련 문의	서울 내에 DT 매장 있나요??	goc1118	수정	삭제



번호	제목	내용	작성자	수정하기	삭제하기
1	상품권 관련	문의 드립니다.	user	수정	삭제
2	매장 직원	어제 AIS 광화문점을 방문했는데, 직원 응대에 실망했습니다. 교육 부탁드려요	rla04050	수정	삭제

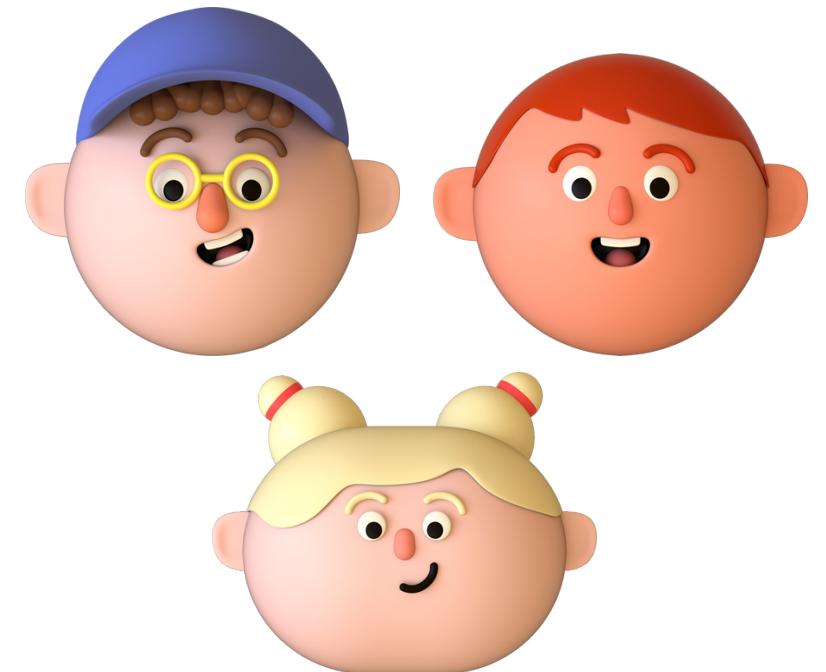
```
def boardDelete(request, pk):
    board = Board.objects.get(id=pk)
    board.delete()
    return redirect('accounts/board.html')
```

-> 삭제 구현 코드
accounts > views.py

수행결과 시연



느낀점



잘한 점

- 훈련 내용 뿐만 아니라 구글링을 통해 배우지 않은 다양한 css, js 효과를 적용함으로써 훈련 내용 이상의 결과물 도출함
- 협업 툴로 깃허브를 이용하여 협업 방법 또한 익힘
- 대부분의 팀원이 책임감을 가지고 프로젝트를 완성함

아쉬운 점

- 깃허브 사용이 미숙하여 오류 발생 시 해결 시간이 오래 걸림
- 담은 상품을 DB에 저장해 보여주는 장바구니 기능을 추가해 구현해보고 싶음

갈등요소 및 해결 방법

- 분담 후 각자의 파트를 수행하는 부분에서 생각한 아이디어가 서로 다르거나 부족한 부분은 카카오톡 채팅이나 줌 미팅을 통한 적극적인 피드백으로 함께 문제를 해결해나감



감사합니다



<https://github.com/dongnee/testpublic>