





## 산학협력 프로젝트 결과보고서

본 보고서를 SW중심대학 실무형 산학협력 프로젝트 결과보고서로 제출합니다.  
(프로젝트과제명 : Django를 이용한 플랫폼 웹 에디팅 플랫폼 개발)

2021. 07. 15.

참여기업 :	세미콜론즈	이재호	
참여학생 :	정보컴퓨터공학부	김정윤(여)	
	산업공학과	김태윤(남)	
	산업공학과	서정빈(남)	
참여교수 :	소프트웨어교육센터	육동철	(인)
참여인원 :	학부생 03 명,    기업체 : 01 명,    교수 : 01 명,    총 05명		

부산대학교 소프트웨어교육센터장 귀하

## 산학협력 프로젝트 결과보고서(10페이지 이내)

과제명	Django를 이용한 플랫폼 웹 에디팅 플랫폼 개발			
수행기간	2021.03.15. ~ 2021.07.15.		책임교수명	육동철
협력기관명	세미콜론즈		기관담당자명	이재호
참여학생	이름	학번	팀명	따따상 - 2
	팀장: 서정빈	201727525		
	팀원: 김정운	201824450		
	팀원: 김태운	201727518		
참여인원	참여교수 1명(육동철), 학부생 3명(서정빈, 김정운, 김태운), 협력기관 1명(이재호)			

### 추진 배경 및 목표

<망고보드>나 <미리캔버스>와 유사한 웹 에디팅 플랫폼 제작을 진행하였다.

먼저 '상세 페이지'의 경우 미리 제작된 템플릿을 수정하는 앞선 플랫폼들과 달리 수정이나 참고를 원하는 쇼핑몰 페이지의 URL을 입력하고 해당 페이지를 그대로 가져와, 자신의 취향대로 상세 페이지를 수정 및 저장할 수 있는 것을 목표로 한다.

'마케팅 페이지'의 경우 순위 조회 기능을 통해 스토어명 혹은 키워드 입력시 해당 입력값을 포함한 쇼핑몰 판매량 순서대로 정렬하며, 키워드 분석 기능을 통해 입력된 키워드와 관련된 쇼핑몰 페이지 수, 경쟁률 등을 표시하는 것을 목표로 한다.

이를 통해 사용자들은 인기있는 상세페이지가 어떤 형식을 가지고 있는지 확인할 수 있으며, 이를 바탕으로 자신만의 페이지 구축을 쉽게 할 수 있도록 보조한다.

### 수행 결과

URL 입력 혹은 HTML파일 업로드 시 사진 영역과 글 영역을 인식하여 수정을 가능할 수 있게 전처리하는 과정은 따따상-1의 OpenCV팀에서 맡아 진행하였다.

'상세 페이지'의 경우, 입력된 URL이나 HTML 파일을 에디팅 창으로 불러와 각 요소들을 수정할 수 있게 하여야 한다. 하나의 페이지에서 두 개의 HTML 파일을 불러와야 하기 때문에 정식 서버로 올리기 전 node.js를 활용하여 로컬 서버를 생성 후 로컬 서버에서 필요한 기능들을 추가한 후에 정식 서버로 업로드 하였다. 초반에는 글씨 및 그림 수정을 위해 'Canvas' 프로그램을 사용했는데 다른 HTML 파일의 업로드 및 기능의 사용이 어려워 HTML 자체를 즉시 수정 후 바로 반영하는 형식으로 바꿔 수행하였다.

'상세 페이지'의 기능에는 URL 찾기, 이미지 추가, 텍스트 추가, 페이지 다운로드 총 4개의 기능을 추가하였다. 그리고 이는 상세페이지의 어떤 부분에 있더라도 사용이 가능해야 했기 때문에 좌측 메뉴바를 지정하여 드래그에 따라 함께 이동할 수 있도록 하였다. 좌측 메뉴바의 경우 클릭된 기능의 내용만 표시하도록 하기 위해 자바스크립트의 hide와 show기능을 사용하였다.

URL 찾기 기능의 경우, 서버에서 한 페이지에 두개의 html을 올릴 수 있도록 하기 위해 load기능을 사용하여 수정을 원하는 HTML의 이름을 입력 후 버튼 클릭 시 자동으로 해당 페이지를 띄울 수 있게 하였다. 아직 OpenCV와의 연동이 제대로 이루어 지지 않아 임의로 띄울 HTML 파일을 설정 한 후 사용하였다.

이미지 추가 기능의 경우, input의 type을 file로 설정하고 multiple 기능을 통해 다중 이미지 업로드를 가능하게 하였고, fileUploadAction 함수를 생성하여 업로드할 이미지를 선택하면 해당 함수를 실행하도록 설정하였다.

처음에는 이 추가된 이미지들의 경로를 그대로 부여한 후 업로드 전 썸네일로 표시될 수 있도록 시도하였는데, 해당 파일의 경로를 제대로 받아오지 못해 썸네일에는 표시가 되었지만 에디팅 페이지로 업로드 하였을 경우 대체이미지로 나타나게 되었다.

그래서 해당 이미지를 FileReader를 통해 해당 이미지를 데이터로 읽고, setAttribute를 통해 해당 이미지의 경로를 event.target.result를 통해 데이터에 대한 새 경로를 추가하였다.

여러 이미지의 업로드를 가능하게 하기 위해 sel\_files라는 배열과 index=0이라는 변수를 생성하였고 각각의 단

일 파일들을 push한 후에 index를 하나씩 증가시켜 다중 이미지 파일들을 표시할 수 있게 하였다.

기존 파일들의 크기가 큰 경우를 대비하여 각 이미지들의 너비를 100px로 고정시켰으며, for문을 통해 만약 선택된 파일의 타입이 image가 아닐 경우 알림창을 띄우도록 하였다. 각 업로드된 이미지들은 구분할 수 있게 id와 class를 부여하였고, reader.readAsDataURL을 통해 각 파일들을 데이터로 읽어 에디팅 페이지에서도 대체이미지가 아닌 원래 이미지를 표시할 수 있게 설정하였다.

텍스트 추가 기능의 경우 먼저 글씨체 수정을 위해 굵게, 기울임, 밑줄, 글자 크기, 글자 색, 글자 배경색 등에는 각 id를 부여하여 개별 stylesheet를 만들어 각각의 아이디를 통해 해당 기능에 맞게끔 설정하였으며 한글이나 워드에서와 같이 글씨 정렬을 위해 왼쪽 정렬, 가운데 정렬, 오른쪽 정렬과 같은 총 3개의 버튼을 추가하였다.

이또한 stylesheet를 통해 각 기능에 알맞게 설정하였다. 자바스크립트를 활용하여 클릭된 혹은 드래그된 텍스트에 focus를 설정하고 각 기능의 버튼을 클릭 시 글씨가 사용자의 의도에 맞게 변경될 수 있도록 설정하였다.

또한, 기존에 있는 텍스트를 제외한 다른 추가텍스트 입력을 위해 textarea를 넣어 추가텍스트 입력 완료 후 해당 텍스트를 드래그 하여 에디팅 페이지에 드롭하면 그대로 반영되는 형식으로 기능을 추가하였다.

이미지나 글씨의 위치 변경, 사이즈 조정 및 추가된 이미지 및 텍스트를 에디팅 페이지에 추가하고 변경할 수 있도록 하기 위해 jQuery를 사용하였다. 사용한 jQuery의 버전은 1.12.4를 사용하였다.

먼저 에디팅 페이지의 그림 및 글씨, 그리고 추가된 이미지 혹은 글씨의 위치 이동을 자유롭게 하기 위해 jQuery의 draggable 기능을 활용하였고, 업로드된 이미지 혹은 추가 이미지의 사이즈를 조절하기 위해 resizable 기능을 활용하였다.

처음에는 먼저 나타나 있는 글씨 혹은 이미지가 무조건적으로 그 다음에 나타난 이미지 혹은 글씨 밑으로 들어갔다. 일반적으로 글씨는 그림 위에 위치하기 때문에 z-index를 활용하여 무조건 글씨는 앞으로, 이미지는 뒤로 갈 수 있도록 설정하였다.

또한, 에디팅 페이지의 경우 수정이 일어나야 하기 때문에 해당 페이지 자체를 contenteditable="true"로 설정하여 추가 텍스트 기능을 이용하지 않더라도 기존 글씨 수정 후 엔터를 치면 자동으로 p 태그가 할당되어 글씨를 추가할 수 있게끔 설정하였다.

추가로 업로드된 이미지를 썸네일에 띄우고 해당 이미지를 에디팅 페이지에만 추가할 수 있도록 하기위해 추가된 이미지에는 draggable 기능을, 에디팅 페이지에는 droppable기능을 추가하였고, 추가된 이미지의 위치이동 및 사이즈 조정을 위해 class를 부여하였다. 또한 추가된 이미지의 제거를 원하는 경우 더블 클릭시 해당 이미지가 삭제되는 기능도 추가하였다.

위 기능들을 추가하였음에도 위치 조정은 가능하지만 사이즈 조정은 제대로 일어나지 못했다. 뿐만 아니라 추가된 이미지는 에디팅 페이지에서 벗어난 위치로 이동이 안되었지만 기존 페이지에 있던 텍스트나 이미지의 경우에는 에디팅 페이지에서 벗어난 위치더라도 이동이 가능하게 되었다. 로컬 서버에서는 기존에 설정한 css의 적용이 잘되는 모습을 보여주었는데 OpenCV와 병합한 후 정식 서버로 업로드 하였을 경우 Django에서 설정해 놓은 css와 충돌이 일어났기 때문인지 기존 css의 설정이 제대로 실행되지 않았다.

웹 에디팅 페이지에 이어, 전반적인 웹 페이지 틀과 기본 구조는 django 프레임워크를 이용하여 진행하였다. 파이썬 기반의 django 특성과 opencv 실행시 용이한 언어가 파이썬으로 귀결되어 최종적으로 선정하게 되었다. 기본적인 로그인 및 회원가입 기능과 사용자 별 링크와 이미지를 기입받아 저장하는 데이터베이스의 구축 및 부가적인 마케팅 페이지 기능을 백엔드에서 진행하게 되었다.

장고 상에서 모델은 'fcuser', 'changeurl', 'blog' 3개로 구성하였다.

'fcuser'에서는 email과 password의 필드를 두어 기본적인 회원관리 기능을 우선적으로 수행하게 하였다.

해당 폴더 아래 decorators.py파일을 생성하여 권한설정 기능을 도모하였다. login\_required라는 함수를 만들면서 요청한 유저가 authenticated하지않은경우에는 로그인한 사용자만 이용할 수 있다는 내용으로 메시지를 전달하게 설정하였다.

유사하게 logout\_message\_required라는 함수를 만들었는데 이는 이미 로그인한 사용자의 회원가입 및 로그인을 막게 설정함으로써 같은 작업이 잘못 여러번 전달되지 않도록 방지하고자 하였다.

회원가입과 로그인 관련하여서는 각각 RegisterForm, LoginForm의 이름으로 class 기반 개발방식으로 진행함으로써 생산성 및 상속을 통한 활용성을 향상시키고자 하였다. 각 폼 모두 내용이 기입되지 않을 경우 예러 메시지를 전달하도록 인자를 설정하였다. 또한 비밀번호를 비교 및 암호화하기 위하여 장고 내장 check\_password, make\_password를 불러와서 진행하였다.

추가로 비밀번호 관련 설정 시, 입력받은 비밀번호(password)와 다시 입력받은 비밀번호(re\_password)가 서로 틀리지 않게 해주는 기능을 만들기위해 super().clean()을 이용하여 clean이란 이름의 함수를 정의하였다.

그 외로, 페이지 연결 및 이동을 위한 함수들을 url 설정을 하였고 받은 세션을 없애고 홈페이지로 redirect 시키는 원리의 로그아웃 기능과 약관동의 체크사항들(agreement1, agreement2)이 모두 체크되어야 회원가입 페이지로 진행이 되도록하는 약관동의 class도 작업을 해주었다.

'changeurl'에서는 fcuser의 user를 외래키로 받아서 회원 별 기입한 링크와 첨부한 이미지를 관리하고자

get\_url, input\_image 필드를 추가해주었다.

이 때 작업전반 모델내용에 수정소요가 있을 때 마다 생기는 오류들을 처리하고자 할때는 필드인자에 null 및 blank 값을 True로 임시로 설정해주면서 해결하고자 하였다.

그리고 이미지 파일 등의 미디어 파일 제어를 위해 일련의 기본설정 변경 작업도 진행하게 되었다. views.py에서는 user = Fcuser.objects.get(email=request.session.get('user')) 와 같이 로그인 한 유저의 세션을 갖고오고, 추후 서술할 마케팅 페이지관련 내용을 웹에 전달하는 내용들을 수행하게 되었다. 작업이 성공적으로 이루어질 경우 /result/ 페이지로 {'input\_image' : input\_image}을 전달하여 redirect 설정을 하였다.

웹 상에서의 페이지 이동 시에는 같은 changeurl/ 이하에서 작업하고자 show, search, analyze, keyword 이름으로 urls.py에 추가설정을 해주었다.

'blog'에서는 사용자가 최종 파일을 다운로드 할 수 있도록, 파일 업로드 및 다운로드 기능을 구현하였다.

파일 다운로드 관련 작업에서는 임의로 설정을 바꾸어야 할 부분들이 있었기에 쉽지않은 점이 있었다. download 함수를 만들면서 파일경로를 settings.MEDIA\_ROOT로 만들어서 지정해주었고 content\_type을 application/adminupload라 설정해주면서 개발을 진행하게 되었다.

media파일들을 관리하고자 urls.py에서는 설정하고자하는 path에 + static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)를 추가설정해주었다. 또한 settings.py에서 디버그 상황일 때 urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_STATIC\_ROOT)조치를 해주었다.

settings.py에서 파일업로드 및 다운로드 기능을 구현하고자 추가한 설정은 다음과 같다.  
STATIC\_URL = '/static/', STATICFILES\_DIRS = [os.path.join(BASE\_DIR, 'static')], MEDIA\_URL = "/fordown/"  
MEDIA\_ROOT = os.path.join(BASE\_DIR, 'fordown\_cdn')

이후 마케팅 페이지는 프로젝트의 목적이 상세 페이지의 품질이 소비적 행동까지 영향을 미치면서, 유사한 상품 중 실적이 좋은 상세페이지를 벤치마킹하는 방식으로 이용할 수 있었다.

뿐만 아니라 마케팅 페이지란 이용자에게 상품 관련 정보들을 제공해주고 기존 온라인 상에서 어떤 상품이 우세에 있는지 그리고 그러한 상품들의 경쟁력, 키워드 등의 정보를 가져올 수 있는 기능들을 구현하였는데 일단 첫 번째로 상품명들 검색하면 네이버 쇼핑 사이트를 통해 100개의 상품들을 순위별로 가져올수 있는 페이지를 제작하였고 이는 네이버가 자체적으로 제공하는 개발자 api(Application Programming Interface)를 통해 관련된 정보를 가져왔다.

api를 이용하기 위해선 client id 와 client secret 키를 직접 발급 받은 후 urllib.request 및 parse 모듈을 이용하여 웹 페이지 상에서 검색한 내용의 정보들을 가져오는 식으로 진행하였다.

api를 통해 정상적으로 정보가 받아지면 json 형태의 dictionary와 유사하게 이용할 수 있는 구조인데 여기에서 최저가, 카테고리, 사진 및 링크, 관련 연관 카테고리등을 페이지에 표현하였다. 그리고 총 존재하는 상품들의 개수를 input 키워드를 통해 나타내었다.

두 번째로는 datalab이라고 별칭한 페이지인데 이것도 앞서 네이버 쇼핑에서 제공하는 api와 같이 네이버 광고 관련 페이지에서 제공하는 api를 이용해 정보를 가져오게 되었다.

코딩 방식은 크게 다르지 않지만 네이버 쇼핑에서 정보를 가져올때와 다른 점은 class signature 과 get\_header 라는 함수들을 이용해 사이트에 맞는 형식으로 메시지를 변경하는 식의 과정이 필요했고 contents 타입과 timestamp 라는 속성을 적절하게 적용시켜 주어 api를 사이트에 요청하는 식으로 진행하였다.

이후 과정에서는 api 안에 다양한 정보가 전달되었지만 모두 영어로 적혀있었기 때문에 이를 pandas 의 dataframe을 이용해 전체적으로 바꿔주는 과정으로 진행하였고 결론적으로 표 형식으로 사용자에게 제공해주려 했기 때문에 관련 데이터를 정제해주는 과정이 필요했다.

이런 과정을 끝내고 html 즉 페이지의 데이터를 보여주는 과정에 있어서 render 라는 함수에 df 항목을 전달해주는 방식이었는데 첫 번째로 dataframe을 html 형식으로 변경해주는 to\_html() 함수를 유용하게 사용할 수 있었고 django의 template 코드 즉, render 함수로 뿌려준 정보들을 웹 페이지에 표시해주는 형식을 통해 불러왔지만 정보가 깨지는 등의 문제점이 들어났다.

어떤 부분에 오류가 있었는지 시행착오 시간이 꽤 있었지만, 결론적으로는 { df | safe } 라는 형식으로 safe 필터를 적용해 줌에 따라 html 형식으로 표가 잘 나타나는 것을 확인할 수 있었다.

총 9개의 정보를 가져왔고 연관 키워드와 검색수 그리고 클릭률 및 광고 관련 노출 광고수들을 수치상으로 나타내었다.

마지막 세번째페이지는 앞선 내용과 공통되는 부분이 있지만 사용자가 페이지를 만들 때 기존 상품들의 태그, 연관검색어 및 내용들을 가져와서 순위별로 보여주는 기능들을 추가한 페이지를 제작하였다.

하단에 첨부한 사진에서도 알수 있듯이 추천 카테고리나 상품명들 제공하여 사용자 편의성을 제공하는 형식이었는데 이러한 카테고리 및 상품명은 기존의 상품 카테고리를 기존에 사용했던 네이버 쇼핑 api 로 가져왔고 이

렇게 가져온 데이터를 분석하여 어떤 키워드들이 가장 많이 포함되었는지 빈도 분석 과정을 거쳐 나타내는 형식으로 진행하였다. 여기서 counter 라는 모듈을 이용해서 딕셔너리 형태로 데이터를 넣고 반복문을 이용하여 단어가 나온 수를 늘리는 형식을 이용했다.

이러한 과정들에서 처음에는 beautifulsoup를 이용해 데이터를 가져오려 했지만 데이터의 개수가 5개로 한정되는 문제를 해결할 수 없어서 절충안으로 api를 사용하였고 또한 원래 기업체에서 요구한 사항중 tag 라는 키워드도 추출하여 추천 형태로 제공하려고 하였지만 이는 기존 페이지에서 관련 쇼핑 페이지로 이동해 정보를 가져오고 다시 돌아가 반복하는 형태로 selenium 이라는 기능을 통해 가져오는 방식을 구상하였다.

하지만 네이버 쇼핑 플랫폼의 특성상 네이버 스마트 스토어 그리고 여러 쇼핑몰 사이트가 있기 때문에 tag 내용을 가져오는 find\_all 및 select 함수의 parameter를 적용시켜 주는 부분에 어려움이 있었고 최종적으로는 반영하지 못하여 이러한 점을 보완함이 필요한 것을 느꼈다.

또한 지속적으로 빈도 분석을 시행하면서 앞서 언급했던 counter 라는 기능을 반복적으로 사용하였는데 코드 재사용성이나 기능적인 부분에서 함수 및 클래스를 추가적으로 이용해 객체 지향적인 프로그래밍을 했으면 조금 더 깔끔한 구성이 되지 않았을까 하는 아쉬움이 남는다.

## 과제 성과 및 향후 계획

한 페이지에 다중 HTML 업로드 및 텍스트와 이미지 위치 수정과 사이즈 수정이 가능하고 추가 이미지를 여러개 선택 후 썸네일로 미리 볼 수 있다. 또한 추가된 이미지 및 텍스트를 다른 페이지로 드래그 & 드랍하여 해당 페이지 내에서도 수정이 가능하다.

앞서 제시했듯이 에디팅 페이지를 벗어난 위치 이동, 추가된 이미지의 사이즈 조정 및 Django에서 설정한 css와의 충돌등 자잘하게 발생한 문제들을 해결하고, 다중 이미지 선택 시 드래그하면 추가된 이미지들이 한꺼번에 이동되는 모습을 보였는데 이를 개별적으로 선택하여 이동 및 조정이 가능하게끔 설정하여 프로젝트의 완성도를 높여야 한다.

백엔드 부분에서는 기본적인 웹 페이지 구성요소들을 충족하도록 제작 완료하였고 추가로 관련된 마케팅 페이지를 제작함으로써 사용자의 편의성을 제공할 수 있는 웹 서비스를 제작하였다.

하지만 추천 관련 마케팅 페이지에서 tag 관련 기능은 추가적인 작업으로 보완작업이 필요하고 전반적인 UI 요소들도 조금의 개선이 필요하다고 판단된다.

그리고 로그인 한 사용자의 세션을 잘 유지시키면서 회원 별 가입받은 url과 이미지파일이 잘 전달되는지의 데이터베이스 관련 파트들에 대한 지속적인 관심경주와 심층적인 학습이 필요할 것이다.

## 관련 사진

### 1. 웹 화면

A Dena

상세페이지 만들기

마케팅 페이지

로그인 회원가입 고객센터

"잘 팔리는 상세페이지의 비밀"  
상세페이지에 심리학이 적용되면 벌어지는 일



'사진'과 '텍스트'만 수정하면 멋진 나만의 상세페이지 완성

의류

뷰티

잡화

식품

여행

유아

가구



### 2. 링크 입력받기 및 첨부한 사진 미리보기

링크를 기입해주세요

url 입력

<https://plato.pusan.ac.kr/>

변환

파일 선택

조던.PNG



다운로드 받기

home

### 3. 마케팅 페이지 - 키워드에 따른 데이터 정보

검색할 키워드를 입력하세요									분석
데이터 정보입니다.									
연관키워드	월간검색수_PC	월간검색수_모바일	월평균클릭수_PC	월평균클릭수_모바일	월평균클릭률_PC	월평균클릭률_모바일	월평균노출광고수	경쟁정도	
0 영양제	6850	27900	35.8	303.3	0.56	1.16	15	높음	
1 농약	3300	10200	31.6	242.3	1.02	2.56	15	높음	
2 건강식품	3740	6640	30.7	59.3	0.87	0.95	15	높음	
3 여성영양제	410	1130	1.5	13.7	0.39	1.30	15	높음	
4 캡마그	250	1350	6.9	51.1	2.87	4.13	14	높음	

### 4. 마케팅 페이지(2) - 상품 순위 관련

검색 키워드

검색

■ 총 검색 개수: 5470339 개

■ 페이지 1위



락토핏 생유산균 골드 2g x 50포

- 최저가 : 7620원 , 카테고리: 식품 > 건강식품 > 영양제  
- 구매 링크: <https://search.shopping.naver.com/gate.nhn?id=13016571849>  
- 브랜드 : 락토핏

### 5. 마케팅페이지(3) - 카테고리, 연관검색어 관련

#### 영양제

##### 추천 카테고리

식품, 건강식품, 영양제,

##### 추천 상품명

30캡슐, 60캡슐, 500mg, 오메가3, 프로바이오틱스

#### 상품명 단어

네이버 쇼핑에서 **영양제**(으)로 검색시 상위 페이지 안에서 상품명에 가장 많이 사용된 단어 순위입니다 (상위1~40위)

1. 30캡슐, 60캡슐, 500mg, 오메가3, 프로바이오틱스

#### 카테고리

네이버 쇼핑에서 **영양제**(으)로 검색시 상위 노출되는 카테고리입니다.

식품, 건강식품, 영양제,

#### 연관 검색어

네이버 쇼핑에서 **영양제** (으)로 검색시 검색창에 제공되는 연관검색어입니다.

1. 여성영양제, 포스파티딜세린, 어린이영양제, 마그네슘, 징코민, 갱년기영양제, 질유산균, 루테인,