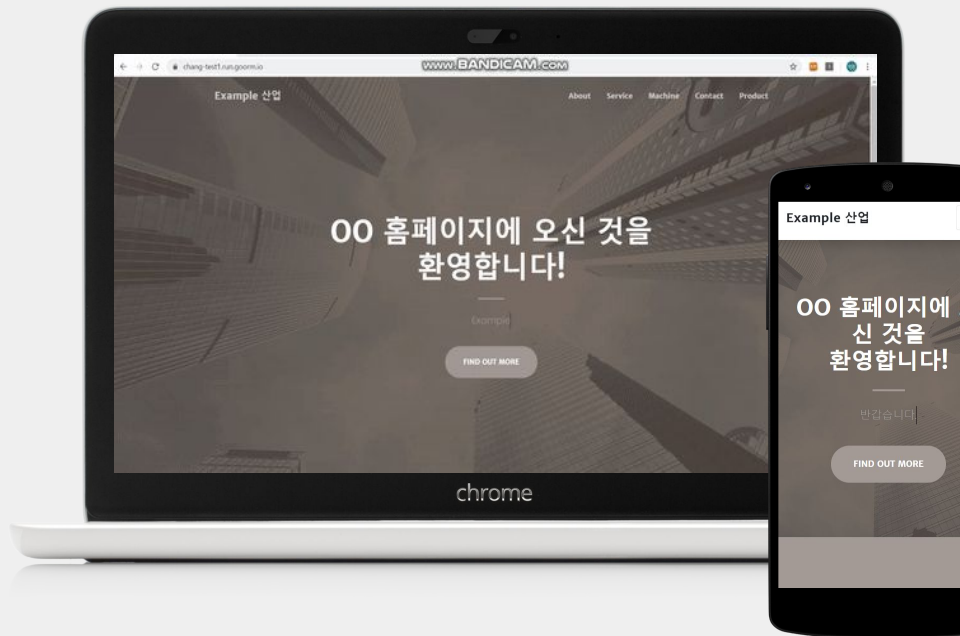


소프트웨어 공학 팀프로젝트 최종 발표

갓성비 팀

- 김영환 (201511795)
- 남승철 (201511809)
- 문정민 (201511812)
- 박은혜 (201713834)
- 심세은 (201713886)



Outline

✓ 프로젝트 개요

✓ 진행과정

✓ 설명 및 시연

✓ 자체평가

목차로 돌아가기

프로젝트 개요

- 세부 목차

- 프로젝트 선정이유
- 고객의 요구사항
- 기초자료 수집
- 요구 분석
- 요약

프로젝트 선정 이유

지인의 회사 홍보사이트

지인의 회사에서 기존에 사용중이었던 오래된 홍보 페이지를 최신 트렌드의 모습으로 새롭게 단장하고자 하였습니다.

추가로 기존의 기능에서 몇가지 추가하고자 하는 기능이 있으셨고, 이를 구현하고자 우리팀에 의뢰를 해주셨습니다.

고객의 요구 사항

요구 내용

- 제작된 지 오래된 홈페이지를 트렌드에 맞게 개선
- 메인 페이지에서 가장 먼저 회사 전경, 소개를 보여주기
- 회사 소개의 시각적 강조
- 기성품 판매 기능
- 주문 및 문의 시, 메일 알림 기능
- 관리자 권한으로 사이트 내부정보 수정가능



기초자료 수집

기존 사이트에 대한 자체분석

고객이 요구한 사항중 제품의 디자인적인 면의 개선이 있었기 때문에 그러한 면에서의 개선점을 찾았고,

추가로 구현 가능한 기능이 무엇이 있을까 기존의 사이트 분석을 시작하였습니다.

Main Page / 전부 사진으로 UI구성

1. 전부 이미지로 UI를 구성

- 사이트 최초 접속시 받는 데이터가 크다.

2. 사이트에서 다음 사이트로의 이동시간

3. 개발자 없이 사이트에 대한 수정이 불가능

4. 사용하지 않는 메뉴버튼

5. 모바일 기능 미지원



Product Page / 세분화 되지 않은 구조

1. 세분화 되지 않은 구조

2. 원본 사진 미리보기

- 받아오는 데이터가 클

3. 개발자 없이 사이트에 대한 수정이 불가능

4. 상세보기 페이지 미지원

5. 모바일 기능 미지원



문서화 (Google Docs 이용)

문서작성 : Google Docs에 동시에 작성 및 수정

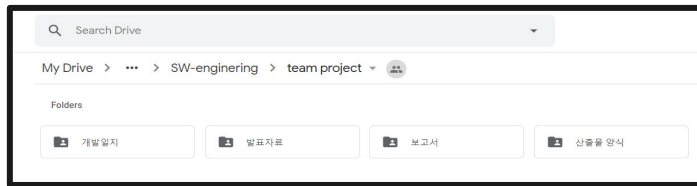
- Project Proposal (제안서)
- Story Card (스토리 카드)
- 현장조사

협업도구 사용 : Trello

- Trello (협업도구)

세부목차로
이동

문서 사진



프로젝트 제안서		
1. 프로젝트 명 : 웹 사이트 리뉴얼 프로젝트		
2. 프로젝트 팀과 명단		
가. 팀 명 : 갯성비		
나. 팀 원(팀장은 * 표시) 및 역할		
이름	학 번	역할
*김영환	201511795	PM 및 CTO
남승철	201511809	Backend 팀장, 기술 명세
문정민	201511812	Backend 팀원, 단위 테스트 담당
심세은	201713886	Frontend 팀장, 요구사항 명세
박은혜	201713834	FrontEnd 팀원, UI디자인

온라인 회의 (HangOut 이용)

HangOut이라는 Google 서비스를 이용하여
온라인으로 회의를 진행.

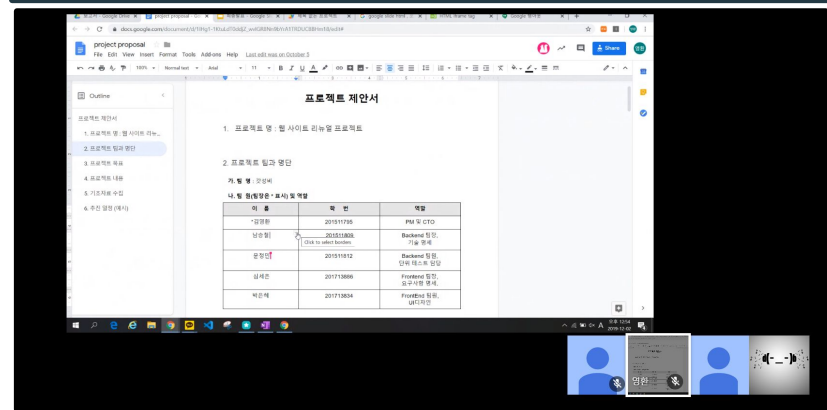
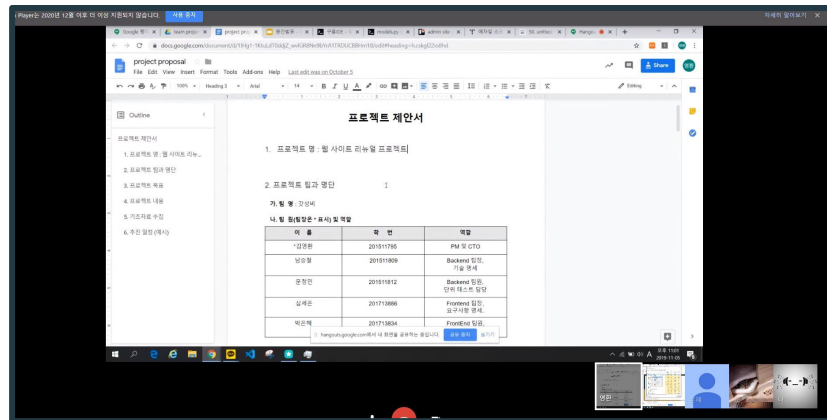
회의 안건 : 기존 사이트 분석

회의 방식 : HangOut 이용

- 화면 공유 + 그룹통화 가능

문서작성 : Google Docs에 동시에 작성 및 수정

행아웃 사진 (온라인 회의)

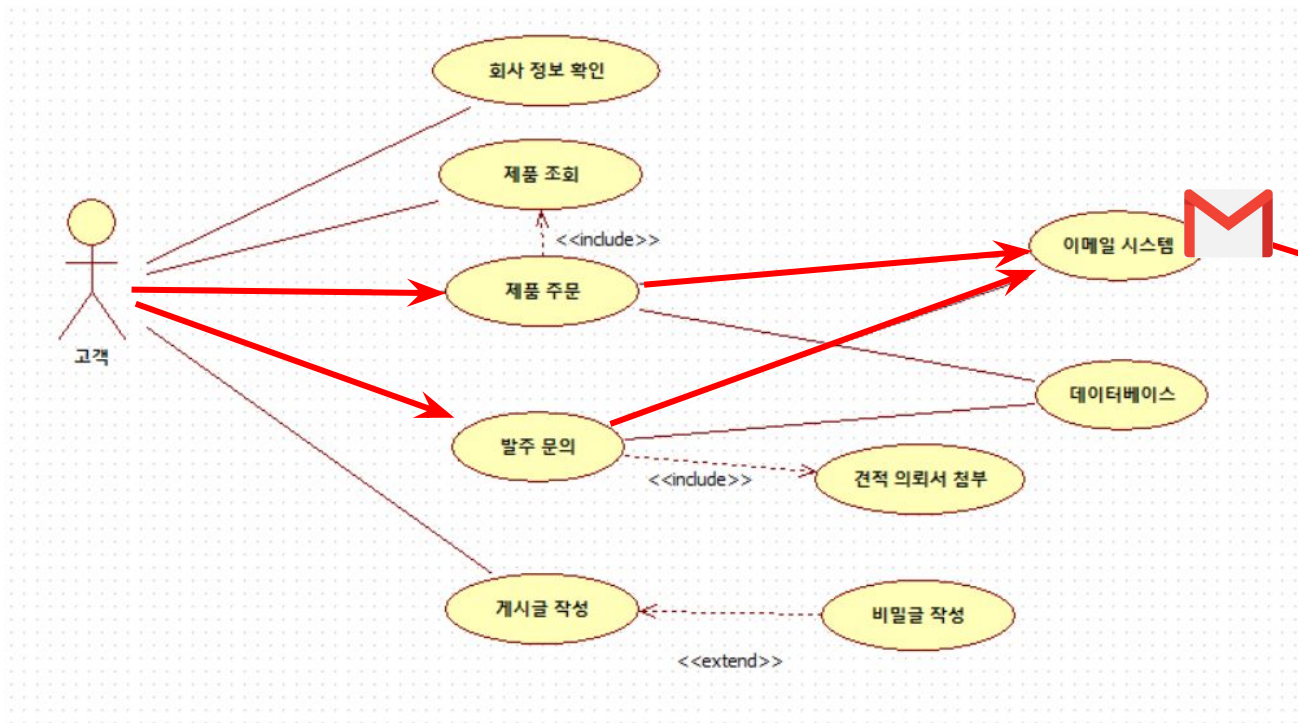


요구분석

우선순위에 따른 요구사항

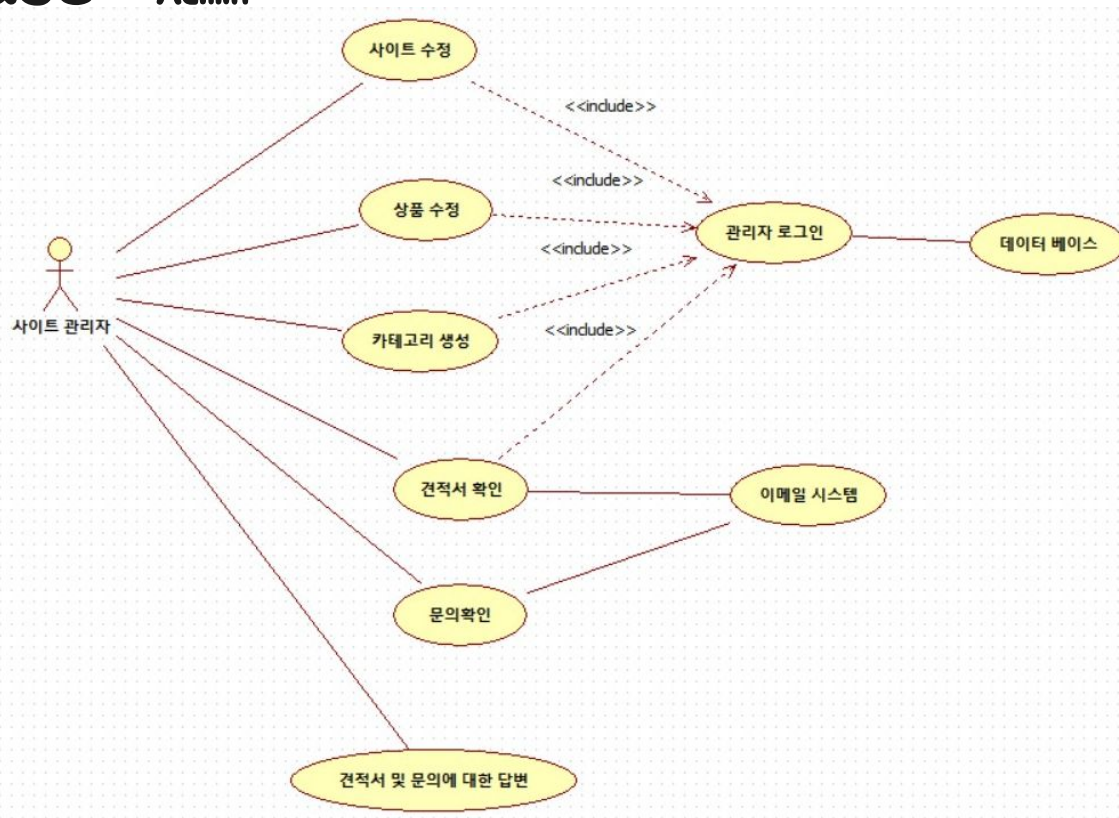
1. 홈페이지를 트렌드에 맞게 개선
2. 기성품 판매
3. 관리자 권한으로 사이트 사진 및 글 작성
4. 주문 및 문의 시, 메일 알림 기능
5. 메인 페이지에서 가장 먼저 회사 전경, 소개를 보여주기
6. 회사소개의 시각적 강조

Use Case - User

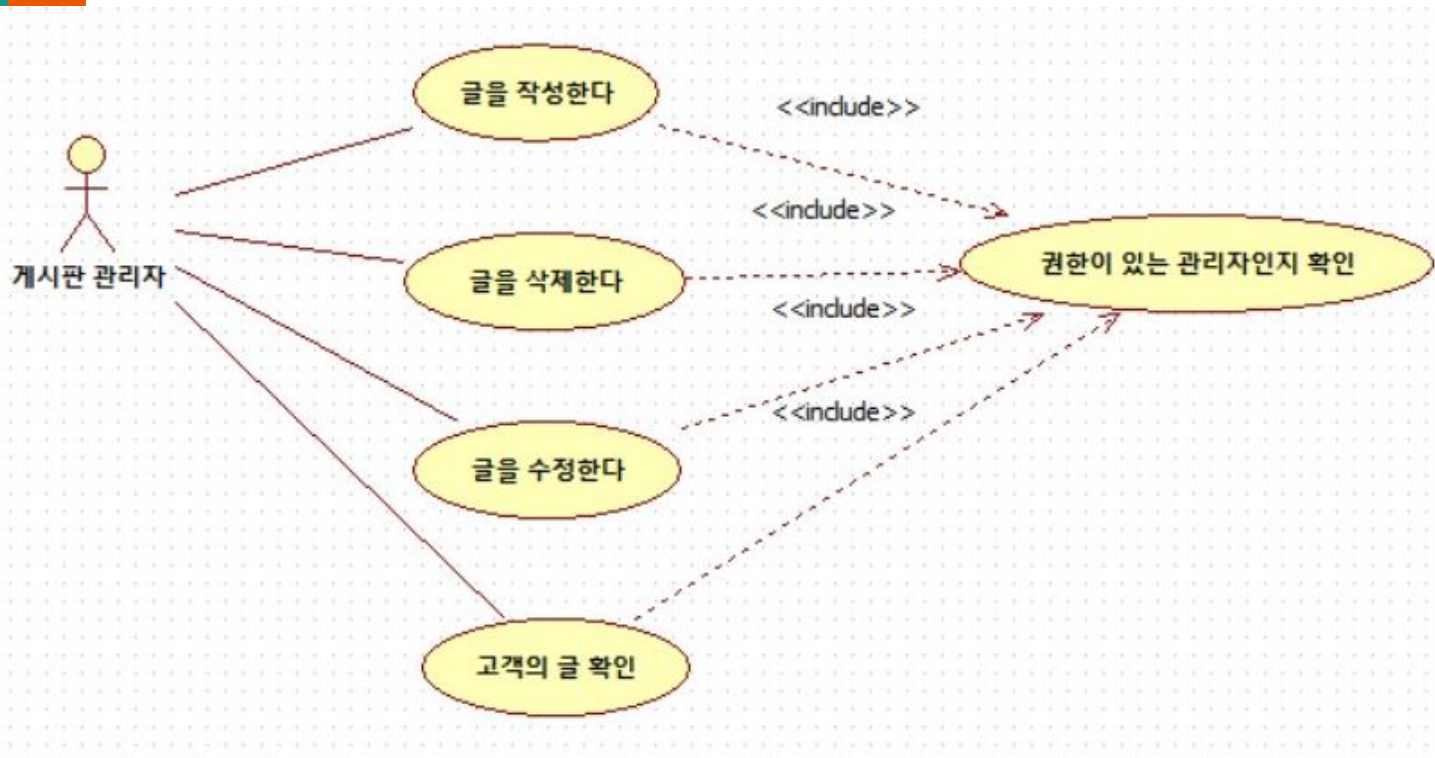


관리자

Use Case - Admin



Use Case - Admin



목차로 돌아가기

진행과정

- 세부 목차

- 공학기법 선정
- 개발환경
- 기초문서 작업
- 제품 백로그 설정
- 스프린트 백로그
- 형상관리 및 TEST

공학기법 선정

웹 서비스 개발, 애자일 개발

웹 서비스 개발은 V모델과 같이 검증 절차와 문서화에 비용이 많이 드는 개발론을 사용하기가 어렵습니다.

추가로 시간이 지남에 따라 변화되는 고객의 요구사항을 받아들이기엔 한계가 있다고 판단 하였습니다.

이러한 이유로 애자일 개발론과 재사용 컴퍼넌트 방법을 채택하였고, 계획기반 방법과 점증적 방법을 적절하게 사용하기 위해 스크럼 기법을 채택하였습니다.

고객의 요구사항을 달성시마다 고객의 평가를 받기위해 프로토타이핑 기법 채택하였습니다.

공학기법

가. 팀 명 : 갯성비

나. 팀 원(팀장은 * 표시) 및 역할

이름	학번	역할
*김영환	201511795	PM
남승철	201511809	Backend 팀장, 기술 명세
문정민	201511812	Backend 팀원, 단위 테스트 담당
심세은	201713886	Frontend 팀장, 요구사항 명세.
박은혜	201713834	FrontEnd 팀원, UI디자인

스크립 주기

- 전체 회의 : 매주 1회
 - 각 팀별 개발 진척상황 확인
 - 소스 병합
 - 요구조사 재반영
- 팀별회의
 - 팀장 지시 아래 필요시 마다 온/오프라인 회의

테스트주도 개발(TDD)

- Python에서 지원되는 단위 테스트 모듈사용
 - unittest 모듈의 사용

개발 환경



📖 사용된 기술 및 모듈

- 구름 IDE
 - 클라우드 서버에서 개발
 - 서버 자동 배포
- Python
 - 3.6.8 버전
- Django
 - 2.2.4 버전
- Bootstrap
 - Frontend framework
- jQuery & vanilla JS
 - 동적 처리
- Db sqlite3
 - Django 내장 database



기초 문서작업

온/오프라인 활동에서 도출된 결과들

최대한 팀원들의 프로젝트 참여의 효율성을 증대시키기 위해서 오프라인 회의만 고수하지 않았습니다.

오프라인 회의를 최대한 활용하여 회의 결과나 각자에게 부여한 조사항목들을 Cloud 문서에 기록하였습니다.

추가적으로 DB설계와 같이 문서작업이 필요하겠다고 판단된 부분에 대해서는 UML등을 활용해 문서화 작업을 실시하였습니다.

오프라인 회의 - 8303 이용



온라인 회의 - HangOut 이용

Flash Player는 2020년 12월 이후 더 이상 지원되지 않습니다. [사용 중지](#) 자세히 알아보기 X

Google Docs: project proposal

Outline

- 프로젝트 제안서
 - 1. 프로젝트 명: 웹사이트 리뉴얼...
 - 2. 프로젝트 팀과 명단
 - 3. 프로젝트 목표
 - 4. 프로젝트 내용
 - 5. 기조자료 수집
 - 6. 추진 일정 (예시)

1. 프로젝트 명 : 웹사이트 리뉴얼 프로젝트

2. 프로젝트 팀과 명단

가. 팀 명 : 갯성비

나. 팀 원(팀장은 * 표시) 및 역할

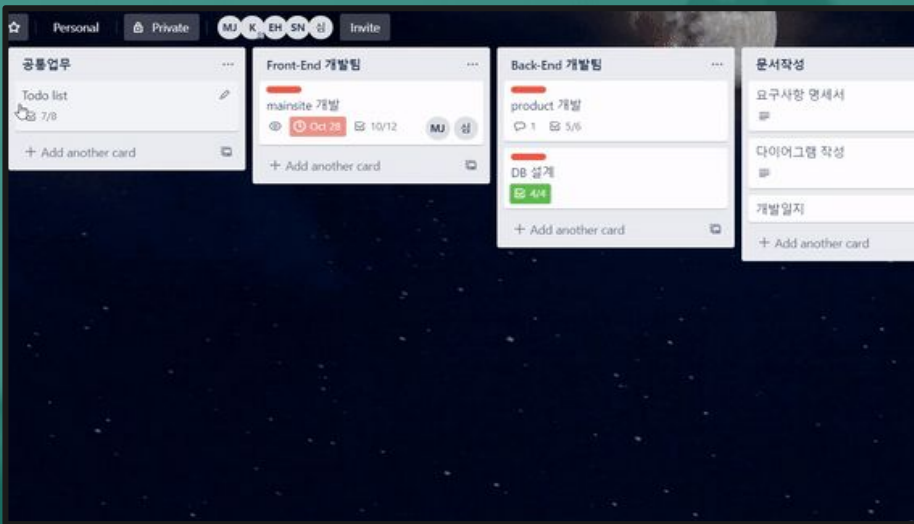
이름	학번	역할
*김영환	201511795	PM 및 CTO
남승철	201511809	Backend 팀장, 기술 명세
문정민	201511812	Backend 팀원, 단위 테스트 담당
심세은	201713886	Frontend 팀장, 요구사항 명세
박은혜	201713834	FrontEnd 팀원

hangouts.google.com에서 내 화면을 공유하는 중입니다. [공유 중지](#) [숨기기](#)

오류 1101
2019-11-05

영환

제품 백로그 설정



팀프로젝트 협업 도구 사용

팀프로젝트에 대한 진행 사항과 각자의 임무를 명확히 보여주기 위해서 Trello라는 협업도구를 사용하였습니다.

현재 진행하는 프로젝트의 요구분석 내용을 토대로 시나리오 카드를 작성하였습니다.

그 후 시나리오 카드에 따라 가장 먼저 진행해야 하는 순서로 프로젝트 진행흐름을 설정하였습니다.

스토리 및 카드

📖 스토리카드

일단 전반적으로 현재의 사이트보다 디자인적으로 개선이 되었으면 좋겠습니다.

메인 페이지에 들어왔을 때 가장먼저 회사의 전경과 회사 소개가 나왔으면 좋겠습니다.
그리고 회사 소개부분이 시각적으로 강조가 되었으면 좋을것 같습니다.

그 다음 다른 타사 사이트와 같이 메인페이지에 CEO 인사말이 들어가 있으면 좋겠습니다. 추가로 협력사에대한 리스트가 화면에 나왔으면 좋겠어요.

그리고 중간에는 회사의 기술을 대표할 수 있는 공간을 마련해주었으면 합니다. 가령 핵심 기계와 같은 사진을 넣어주셨으면 합니다.

추가로 기존에는 기성품 판매없이 사용자가 직접 제품을 설계한거에만 주문을 받았는데, 기성품에 대한 주문이 가능한 형식이 있었으면 합니다. 뿐만 아니라 주문시 DB에 저장은 물론, 그때그때 메일로도 받아보고 싶습니다.

기존의 페이지에서는 사이트에 올라가 있는 사진들을 수정한다거나 하기가 힘들었는데, 수정이 가능했으면 합니다.

또 메인사이트 내부의 CEO인사말이나 소개글도 수정이 가능했으면 해요.

+ 제안사항

- Board App
- 게시판 기능을 만들고 카테고리를 관리자가 생성함으로써 원하는 게시판을 만들고 때에따라 여러가지 용도로 사용하자.
 - ex) 이벤트용 게시판, 설문, 입사지원 자소서 등..]

📖 메인사이트

	요구사항	기능적내용
header	배경을 전경사진으로	배경이미지 <ul style="list-style-type: none">● Css로 처리<ul style="list-style-type: none">○ Opacity : 배경을 어둡게처리● 회사 소개를 배경위에 없기
	회사 소개를 시각적으로 강조	
Section : about	CEO 소개란	About DB 생성 <ul style="list-style-type: none">● CEO 소개 Column 생성 Intro DB 생성 ORM을 이용해 디비에서 불러와 html에 추가
Section: services	회사의 주요 서비스 소개	Service DB 생성 <ul style="list-style-type: none">● Service Column 생성 Contact DB 생성 <ul style="list-style-type: none">● Pax● Office● Call● Email ORM을 이용해 디비에서 불러와 html에 추가
Section: machine	회사의 주요 기계	항후에 바뀌지 않을것을 감안하여 고정된 사진으로 사용

스토리 및 카드

프로덕트 사이트

	요구사항	기능적내용
navbar	서랍형 형식으로 좌측에 배치	상품의 정보를 유저가 보기쉽게 디자인 <ul style="list-style-type: none"> 서랍형 navbar 사용
	제품 종류 별	product_Category DB 생성 <ul style="list-style-type: none"> Category : 상품 군 이름
Section : products	관련상품 진열	Product DB생성 <ul style="list-style-type: none"> Name : 이름 Description : 설명 Price : 가격 Photo : 사진이름 <p>관련 Image 파일 관리</p> <ul style="list-style-type: none"> Media 폴더에 따로 모아서 관리 이미지처리 : Pillow 라이브러리 사용
	기성 상품별 발주기능 메일알림 추가	Order DB 생성 <ul style="list-style-type: none"> Email : 사용자 이메일 Subject : 요청 제목 Order_count : 주문량 Description : 요청내용 Created_at : DB 생성시간 <p>메일전송 함수 및 DB를 동일한 것으로 사용</p> <p>BackEnd 에서 하나의 이메일 계정으로 일괄처리 전송</p>
	사용자 주문제작 발주기능 메일알림 추가	Order DB 사용 <p>메일전송 함수 및 DB를 동일한 것으로 사용</p> <p>BackEnd 에서 하나의 이메일 계정으로 일괄처리 전송</p>

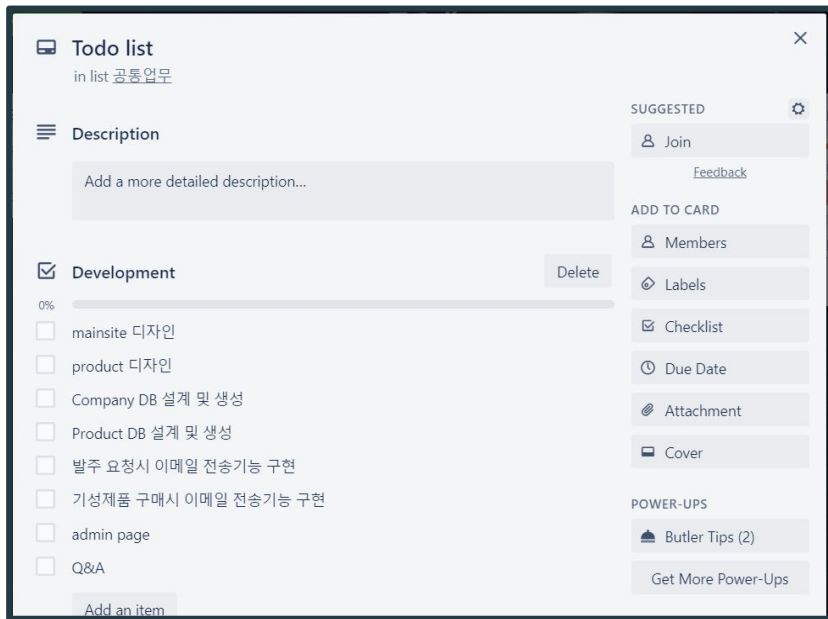
게시판 사이트

	추가사항	기능적내용
Board App	게시판 카테고리 생성	Product template을 재사용 (Extends 이용)
	관리자만 보기 기능	
	게시글 수정 및 삭제	
		Adaptor 패턴 이용 (interface 를 이용하여 다른 뷰를 보여줌)

관리자 사이트

	요구사항	기능적내용
Admin App	관리자 로그인	Django Admin app을 재사용컴포넌트로 사용
	관리자 사이트	
	사이트 관련 DB 접근	
	관리자 계정 생성	
	관리자 Template <ul style="list-style-type: none"> 깔끔한 디자인, mobile, desktop 호환성 	
		Interface를 이용해 Customizing

제품 백 로그 - trello 이용



업무도구인 Trello를 이용

- 우선순위별 배치
- 요구 삽입, 수정, 삭제가 가능



스프린트 백로그 설정

1주 단위로 온/오프라인 회의진행

2주 주기로 회의(sprint) 하는것이 적절하지만,
프로젝트 기한에 따라 1주 주기로 회의를
진행하였습니다.

매주 회의시마다 현재 진행척도를 확인하고 팀별로
작성한 소스를 병합 및 평가하는 과정을 거쳤습니다.

그리고 매 회의 마지막에는 1주일 단위로 가능한
업무 즉 스프린트 백로그를 생성하고 각자가 현재
진행중이던 업무량을 감안하여 새롭게 생성된
스프린트 백로그를 가져가는 시간을 가졌습니다.

ChangWon

문서작성

요구사항 명세서

다이어그램 작성

개발일지

+ Add another card

mainsite

in list Front-End 개발팀 (1주차 스프린트 11.1 ~ 11.8)

Description

Add a more detailed description...

☒ 배경사진을 전경사진으로 - 문정민 - 2019.11.1까지

Hide completed items

Delete

100%

☒ 배경이미지 css☒ 배경-어둡게 처리하기☒ 배경위에 회사소개 내용 없기

Add an item

☒ 회사의 주요서비스 소개 - 심세은 - 2019.11.1까지

Hide completed items

Delete

100%

☒ SERVICE-DB 에서 불러온 DATA 를 HTML에 표시

Add an item

☒ CEO 소개란 - 심세은 - 2019.11.1까지

Hide completed items

Delete

SUGGESTED



Join

[Feedback](#)

ADD TO CARD



Members



Labels



Checklist



Due Date



Attachment



Cover

POWER-UPS



Butler Tips (2)

[Get More Power-Ups](#)

ACTIONS



Move



Copy



Make Template



Watch

Butler (2 Tips) ... Show Menu

Back-End 개발팀 (1주차 스프린트 11.1 ~ 11.8)

mainsite

8/8 SN

product

8/8

admin site

3/3

+ Add another card

Home

Boards

Search

ChangWon

☆

Personal

Private

주차 스프린트

...

Front-End 개발팀 (3주차 스프린트 11.22 ~ 11. 29)

mainsite

6/6

product

5/5

+ Add another card

mainsite

in list Back-End 개발팀(1주차 스프린트 11.1 ~ 11.8)

MEMBERS

SN

+

Description

Add a more detailed description...

ceo소개란 - 박은혜 - 2019.11.1까지

Hide completed items

Delete

100%

about db 생성

ceo소개-column 생성

intro db 생성

db에서 내용을 불러와 프론트엔드로 넘겨주기

Add an item

회사의 주소서비스 소개 - 남승철 - 2019.11.1까지

Hide completed items

Delete

100%

service db 생성

contact db 생성

SUGGESTED

Join

Feedback

ADD TO CARD

Members

Labels

Checklist

Due Date

Attachment

Cover

POWER-UPS

Butler Tips (2)

Get More Power-Ups

ACTIONS

Move

Copy

Make Template

Watch

세부목차로

Butler (2 Tips)

Show Menu

스프린트

...

+ Add another list

SN



형상관리 및 TEST

Test 목록

Product DB

Mainsite DB

Board DB

Product Views.py

Mainsite Views.py

Board Views.py

형상관리 - Github를 이용

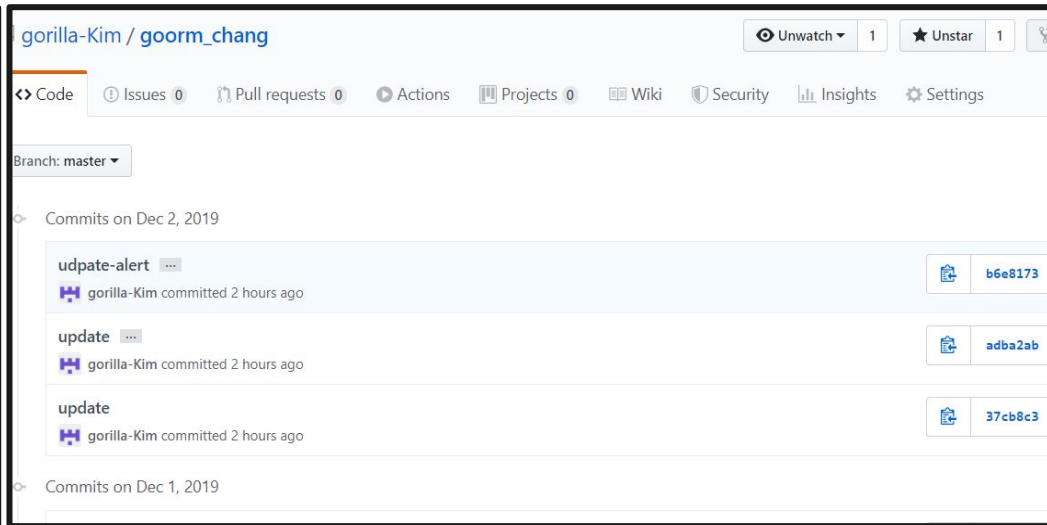
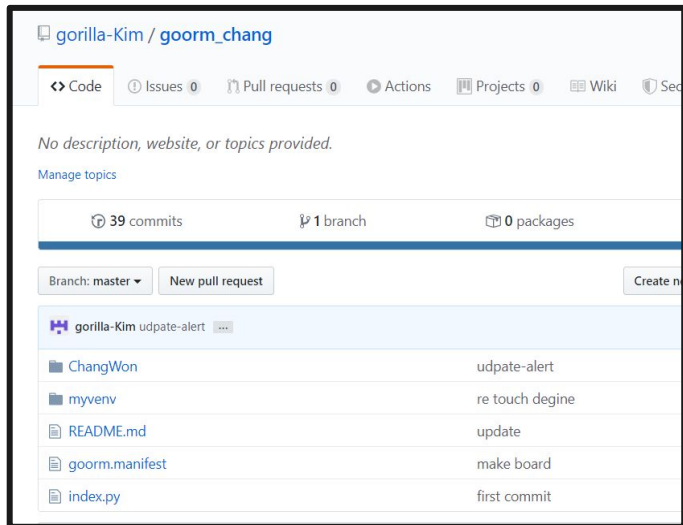
- **사용 서비스 : Github**

- 이유 : 가장 보편적이고, 한 계정당 하나의 Private repository를 지원해줌
- 팀 내 Github 서비스 유경험자 2명이상

- **사용방법**

- 각자 repository를 만들고 수정사항이 발생시 PM의 repository에 병합
- 병합 이후 각자 pm repository를 clone후 작업 재실시 (매 스프린트/회의 시 수행)
- **Brunch 미사용 이유**
 - 경험부족
 - 팀원 모두가 Git을 배우고 시작하기에 시간적 여유가 없었음

형상관리 - Github를 이용 (Click)



Test

Django 에서 지원해주는 unittest인 test.py 를 사용함

DB:

1. assertEquals() 함수를 사용하여 가상 db 에서 insert, select 를 check 함

View.py:

1. client.[get / post] 로 response 객체 생성
2. 이동 할 templates로 rendering or redirect

Error_handler

1. 서버 내부에서 에러 핸들러를 사용

단위테스트 결과->

```
# =====
#                      에러 핸들러 사용법 -김영환
# =====
# parameter (아래 참고)
# request : request,
# activate : 실행모드(True=error생성, False=에러출력 및 삭제),
# msg : message
# =====

# error message handler function -김영환
def error_handler(request, active, msg):
    err_msg = ""
    if request.session.get('error', False) and not active:
        err_msg = request.session['error']
        del request.session['error']
    elif active == True:
        request.session['error'] = msg
    return err_msg

root@goorm:~/workspace/sw_project/ChangWon(master)# python manage.py test

Board App view.py test

Mainsite App view.py test
Mainsite App view.py test success

Product App view.py test
Product App view.py test success
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
Board App test
Board App test success
-----공지사항 접속-----
-----메인페이지 접속-----
.testssubject
Mainsite App test
Mainsite App test success
....Product App test
Product App test success
./workspace/sw_project/ChangWon/products/views.py:47: UnorderedObjectListWarning: Pagination may yield inconsistent results with an unordered object_list: <class 'products.models.Product'> QuerySet.
paginator = Paginator(products, 3)
..
-----
Ran 11 tests in 0.366s

OK
Destroying test database for alias 'default'...
```


Test 결과 보고서

Views.py 단위테스트

Models.py (db) 테스트

Views.py

APP	테스트 항목	세부사항	결과
mainsite	def main	메인페이지 렌더링	ok
product	def main	메인페이지렌더링	ok
product	def order	발주 기능 메일 전송 발주내역 디비 저장	ok
product	def product_view	제품 목록 페이지 렌더링	ok
board	def main	게시판 렌더링	ok
board	def create_board	게시글 작성 디비 저장	ok
board	def read_board	디비에 저장된 게시글 읽어와서 표시	ok
board	def update_board	게시글 수정 기능	ok
board	def delete_board	게시글 삭제 기능	ok
board	def update_page	게시글 업데이트 form 렌더링	ok

Models.py

APP	테스트 항목(컬럼)	세부사항	결과
mainsite	Intro About Service Contact Portfolio	Insert select	ok
product	KindOfProduct Product Order	Insert select	ok
board	Board_Category Board	Insert select	ok

설명 및 시연

- 세부 목차

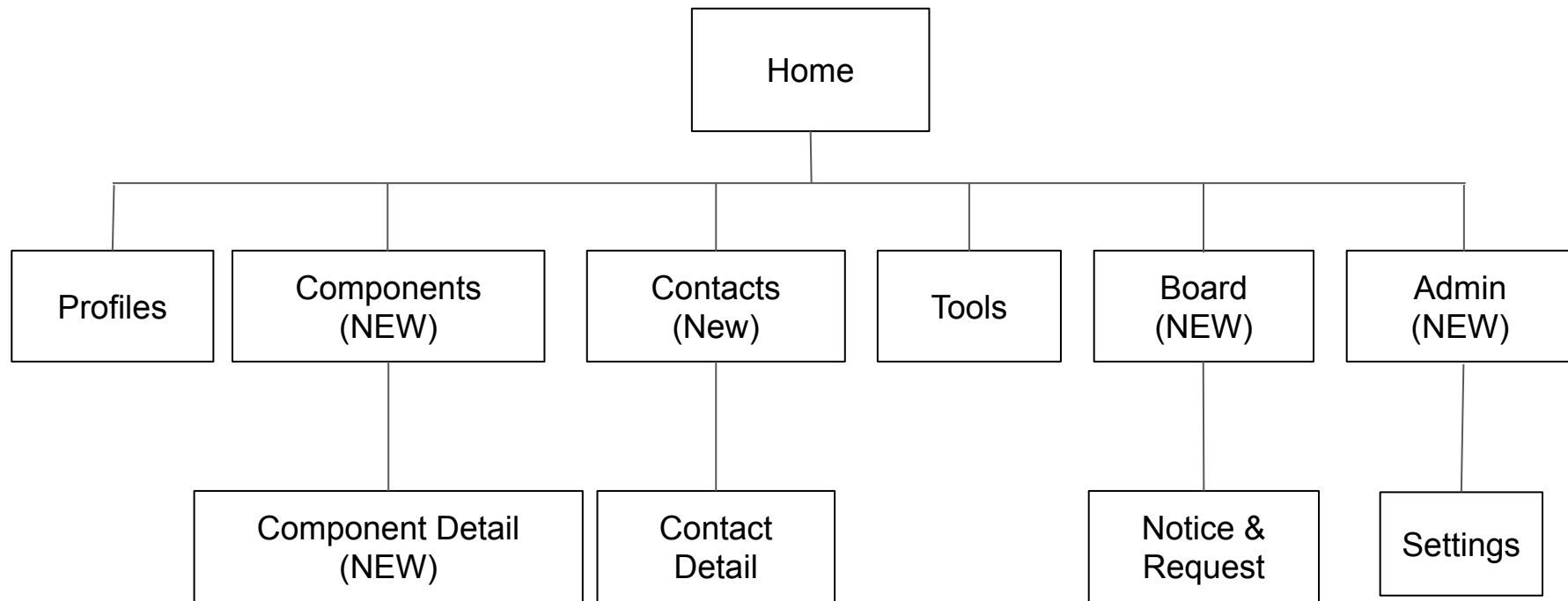
- 프로젝트 살펴보기
- Page별 설명
 - 주요기능 설명
- 영상
- 사이트

프로젝트 살펴보기

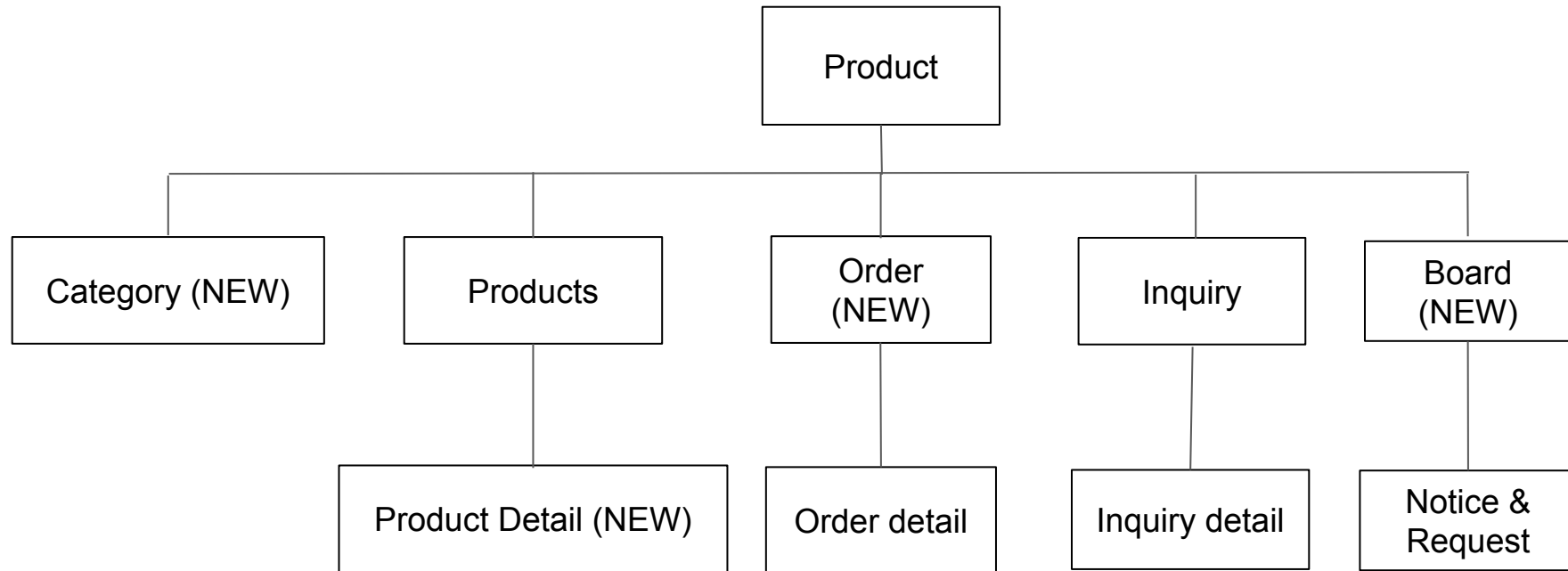
Test 목록

- 프로젝트 package, app
- 프로젝트 사용 패턴(아키텍처 패턴)
- Team(프론트/ 백)별 활동
 - 상세활동
 - 코드설명

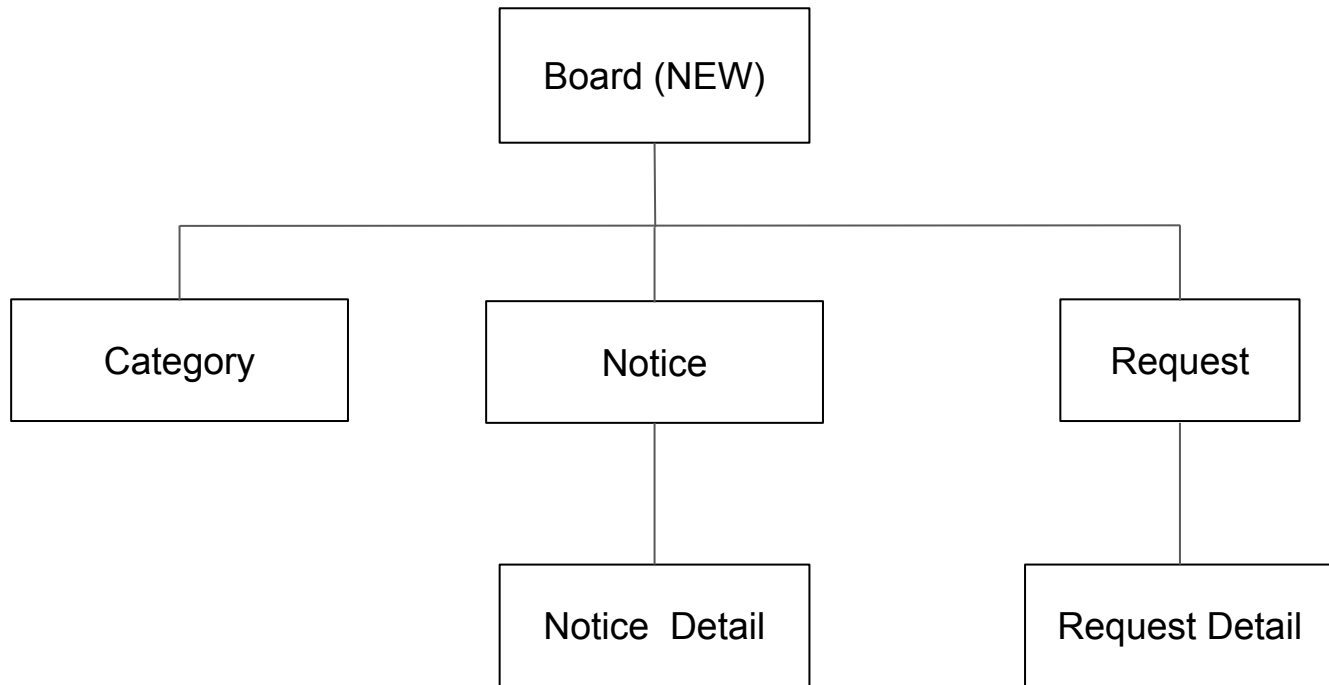
전체 Page의 계층적 구조



Product Page의 계층적 구조

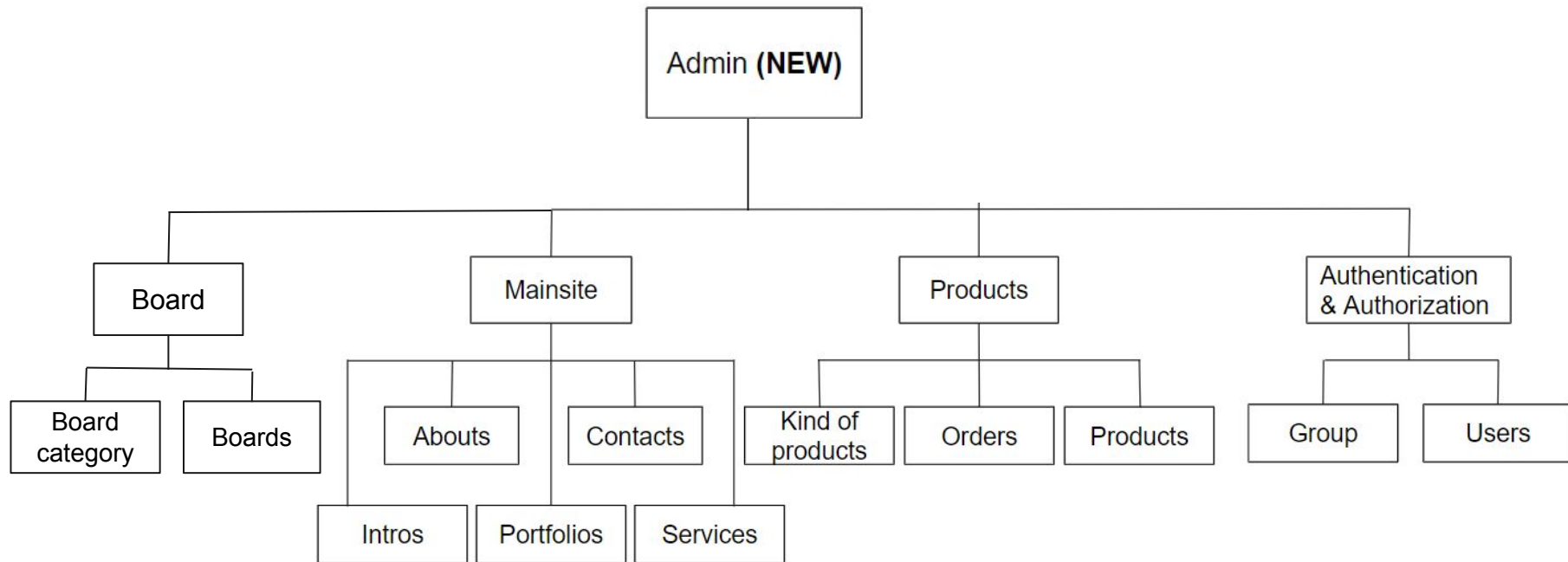


Board Page의 계층적 구조



Admin Page의 계층적 구조

세부목차로
이동

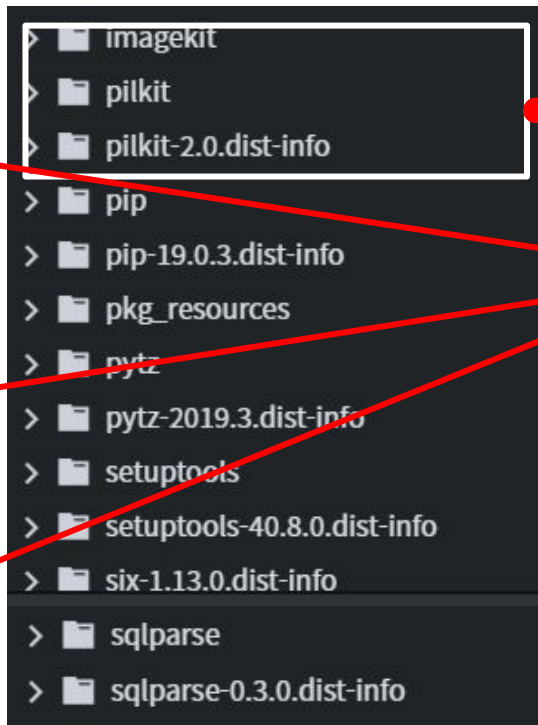
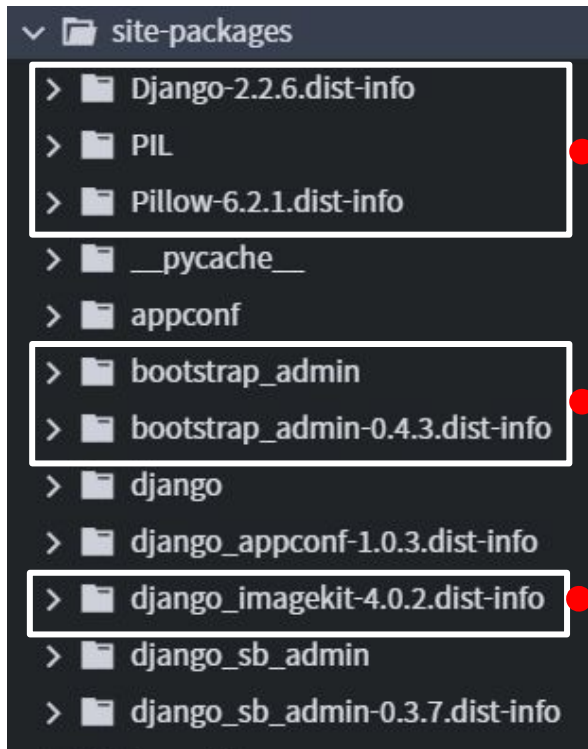




프로젝트 Package & App

프로젝트 Package & App

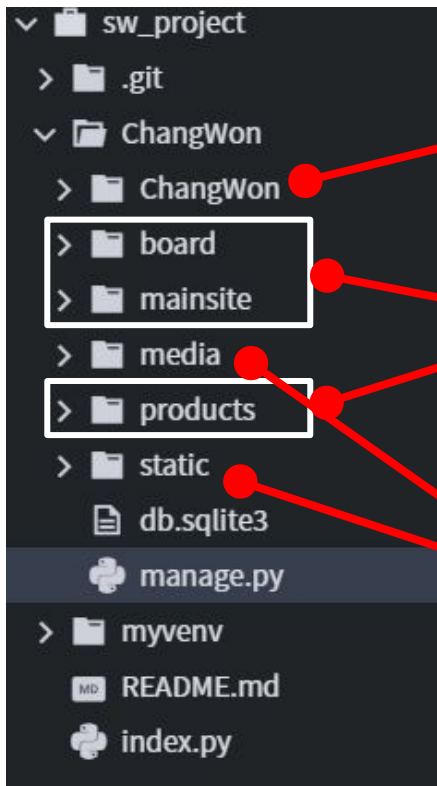
세부목차로
이동



추가로 설치한 패키지

- **이미지 처리**
 - PIL
 - Pillow
 - Pilkit
 - Imagekit
- **Admin customizing**
 - bootstrap-admin

프로젝트 Package & App



projectApp

- Default로 생성 됨

새롭게 생성한 App

- 주요 기능들을 역할에 따라 App으로 분리

파일관리를 위한 폴더

- 고정된 파일
 - Static
- 동적으로 바뀌는 파일
 - media

Front Team

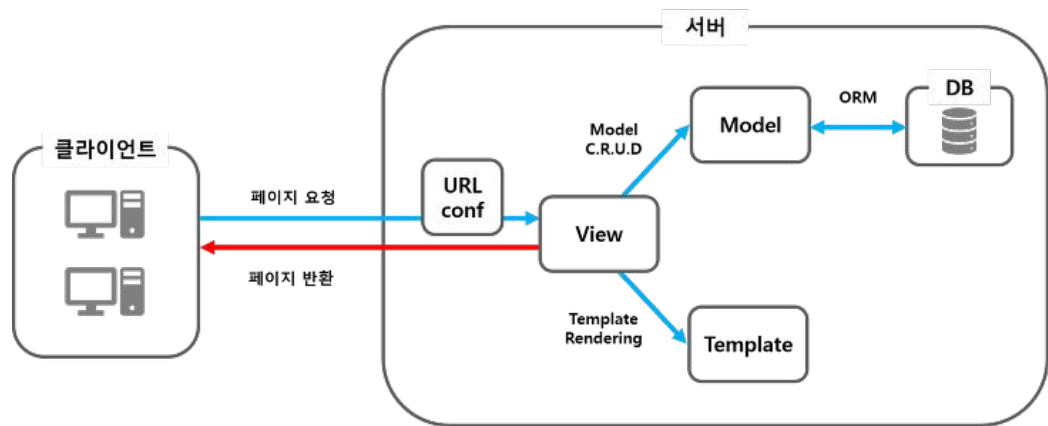
Back Team



프로젝트 주 사용 패턴

프로젝트 사용패턴 (MTV)

세부목차로
이동



장고의 MTV패턴은 자바 웹 프로그래밍의 MVC패턴 방식과 거의 동일한 개념으로, 웹프로그래밍 영역을 3가지 개념으로 나눠서 개발하는 방식이다.

- 테이블을 정의하는 모델(Model),
- 사용자가 보게 될 화면의 모습을 정의하는 템플릿(Template),
- 애플리케이션의 제어 흐름 및 처리 로직을 정의하는 뷰(View)로

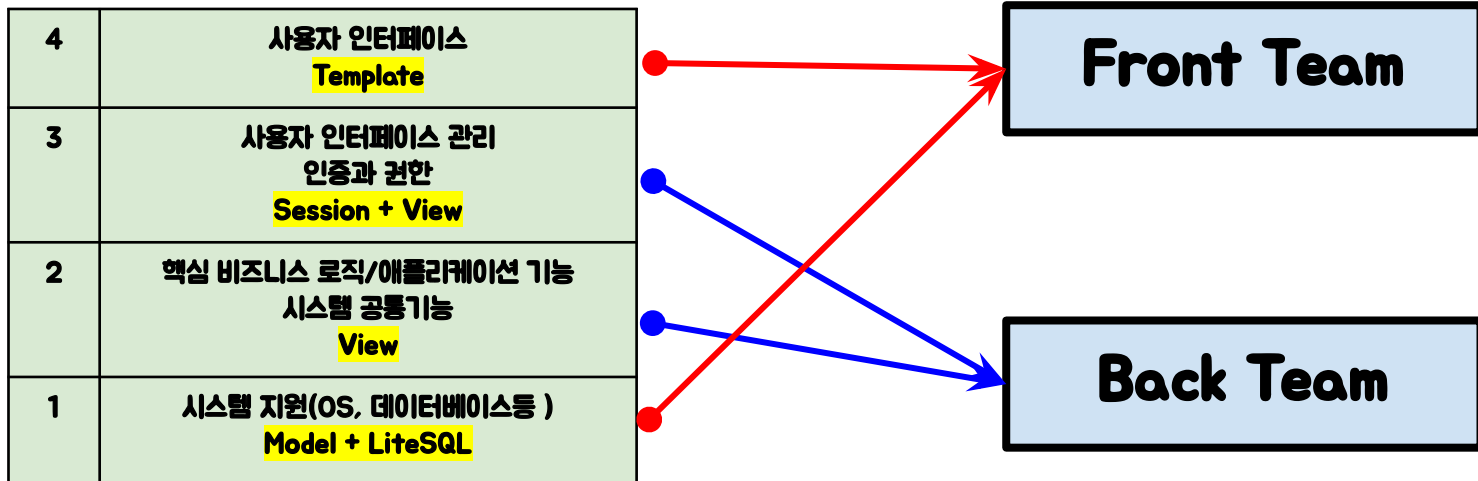
📖 4계층 아키텍처

4	사용자 인터페이스 Template
3	사용자 인터페이스 관리 인증과 권한 Session + View
2	핵심 비즈니스 로직/애플리케이션 기능 시스템 공통기능 View
1	시스템 지원(OS, 데이터베이스 등) Model + LiteSQL

프로젝트 사용패턴 (계층)

세부목차로
이동

📄 4계층 아키텍처



프로젝트 사용패턴 (저장소)

세부목차로
이동

4계층 아키텍처

4	사용자 인터페이스 Template
3	사용자 인터페이스 관리 인증과 권한 Session + View
2	핵심 비즈니스 로직/애플리케이션 기능 시스템 공통기능 View
1	시스템 자원(OS, 데이터베이스 등) Model + LiteSQL

ORM 과 Models.py를 이용해 하나의 DB에 저장하고 필요시
꺼내어 사용

ORM이란

Object Relational Mapping, 객체-관계 매핑

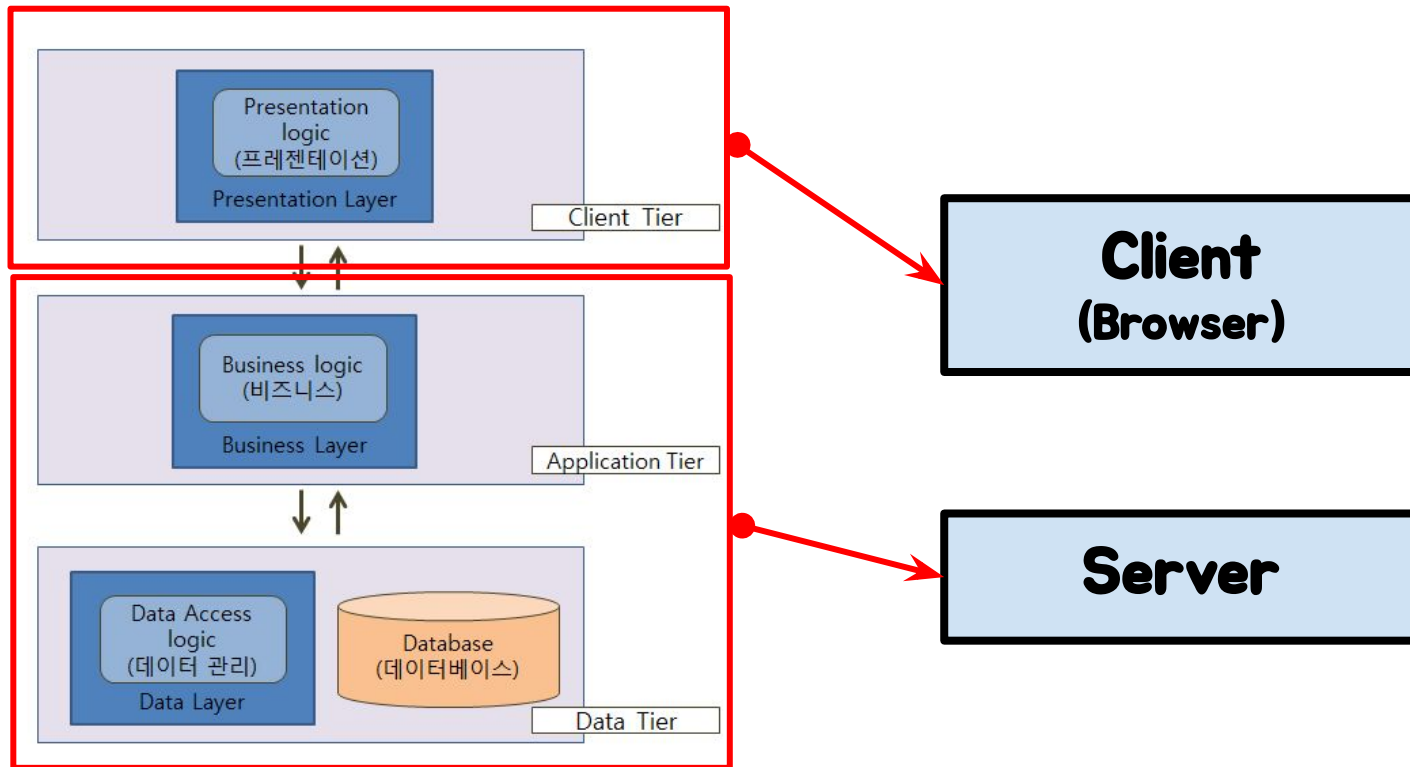
객체와 관계형 데이터베이스의 데이터를 자동으로 매핑(연결)
해주는 것을 말한다.

객체 지향 프로그래밍은 클래스를 사용하고, 관계형
데이터베이스는 테이블을 사용한다.

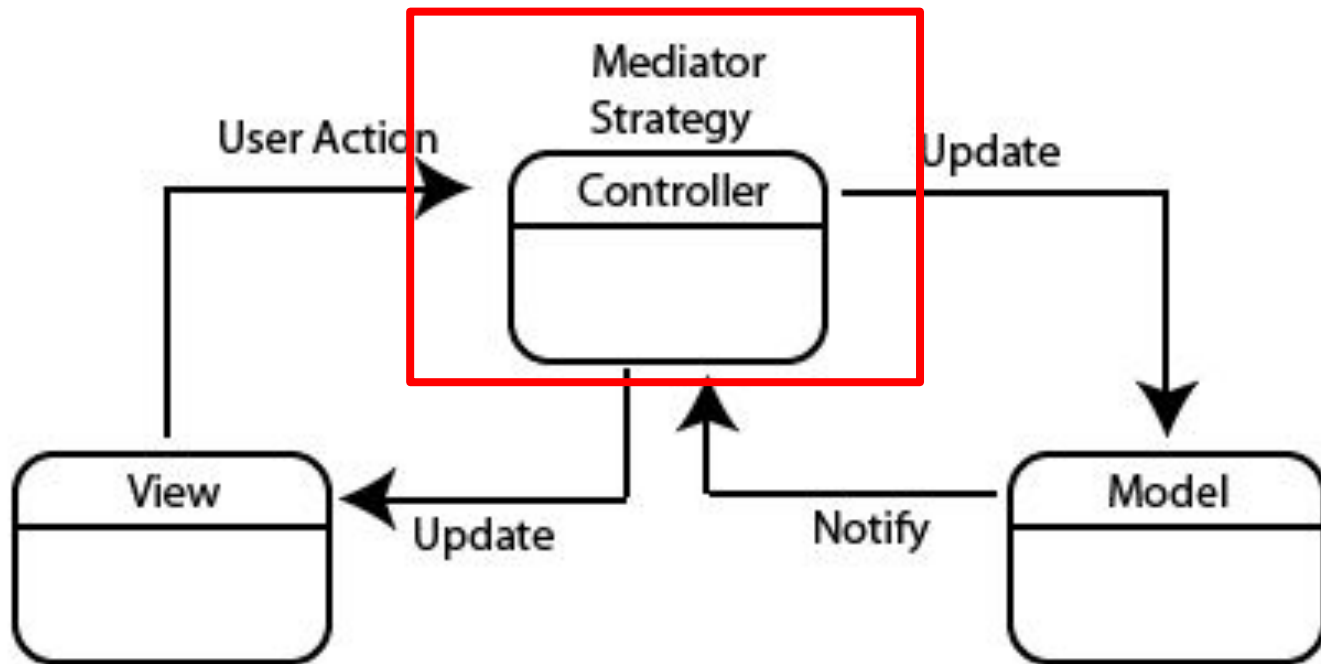
객체 모델과 관계형 모델 간에 불일치가 존재한다.

- ORM을 통해 객체 간의 관계를 바탕으로 SQL을 자동으로
생성하여 불일치를 해결한다.

프로젝트 사용패턴 (클라이언트 서버)



프로젝트 사용패턴 (미디어이터 서버)





Team(Front/Bacck) 별 활동

Front Team

작업 내역

1. Index.html => 기존 레이아웃 명칭 및 구성을 회사에 맞게 수정
2. Product.html => Product를 위한 레이아웃 구성 추가
3. Creative.css => 레이아웃 색상 변경을 위한 root 추가 및 각 레이아웃의 색상 수도 변경
4. Custom.css => 프로젝트 사이드바 색 수정(기본 색 및 마우스 올렸을 때 반응색), 리스트 색 변경, 포트폴리오 부분 이미지 조절, css 타이핑 효과, 에러 핸들러

	전체 라인 수	작성 및 수정 라인 수
index.html	287줄	52줄
Product.html	42줄	42줄
Creative.css	9764줄	113줄
Custom.css	289줄	48줄

Front Team

세부목차로
이동

	김영환	심세은	문정민
index.html	 환경 구축		 레이아웃 명칭 수정
Product.html	 환경 구축	 레이아웃 스타일 수정	 레이아웃 명칭 수정
Creative.css	 타이핑효과  경고창  이미지 조절 추가  이미지 위치 수정	 사이드바  토글바  리스트 색 수정  버튼 색 수정  토글 마우스 올렸을 시 색상 수정  박스 그림자 수정  메인 배경 수정	 주로 사용 할 색상 추가  이메일 링크 색상 수정  버튼 부분 색상 및 효과 수정  버튼 마우스 올렸을 시 색상 및 효과 수정  중간 줄바꿈 이미지 색상 수정  네브바 폰트색 수정  네브바 마우스 올렸을 시 색상 수정  CEO인사말 부분 박스 색 수정  기계 박스 색 수정
Custom.css	 포트폴리오  이미지사이즈 조절  타이핑 효과 추가  경고창 추가	부분  사이드바  올렸을 시 효과 추가  리스트 색 수정	마우스  주로 사용 할 색상 추가

Back Team

Back Team

세부목차로
이동

	views.py		
	Function cnt	전체 라인 수	작성한 라인 수
Main	1개	28줄	28줄
Product	4개	130줄	130줄
Board	7개	192줄	192줄

Back Team

세부목차로
이동

	admin.py		
	Function cnt	전체 라인 수	작성한 라인 수
Main	0개	17줄	17줄
Product	0개	7줄	7줄
Board	0개	4줄	4줄

Back Team

세부목차로
이동

	test.py		
	Function cnt	전체 라인 수	작성한 라인 수
Main	6개	91줄	91줄
Product	4개	94줄	94줄
Board	7개	133줄	133줄

Back Team

세부목차로
이동

	작업내역		
	김영환 ❌ 환경구축 ❌ ErrorHandler 함수	남승철	박은혜
Main		📌 메인함수 작성 📌 test case 작성	📌 read함수 작성
Product	📌 사진 업로드시 thumbnail 이미지 자동 생성 기능 작성 📌 pagination 구현	📌 발주 기능 📌 메일 전송 기능 📌 test case 작성	📌 발주시 주문서 DB저장 📌 create함수 작성
Board	📌 create함수 작성 📌 read함수 작성 📌 pagination 구현	📌 test case 작성	📌 게시물 작성시 DB저장 📌 update함수 작성 📌 delete함수 작성
admin	📌 admin.py이용해서 admin app 기능 커스터마이징	📌 admin navbar 기능추가	

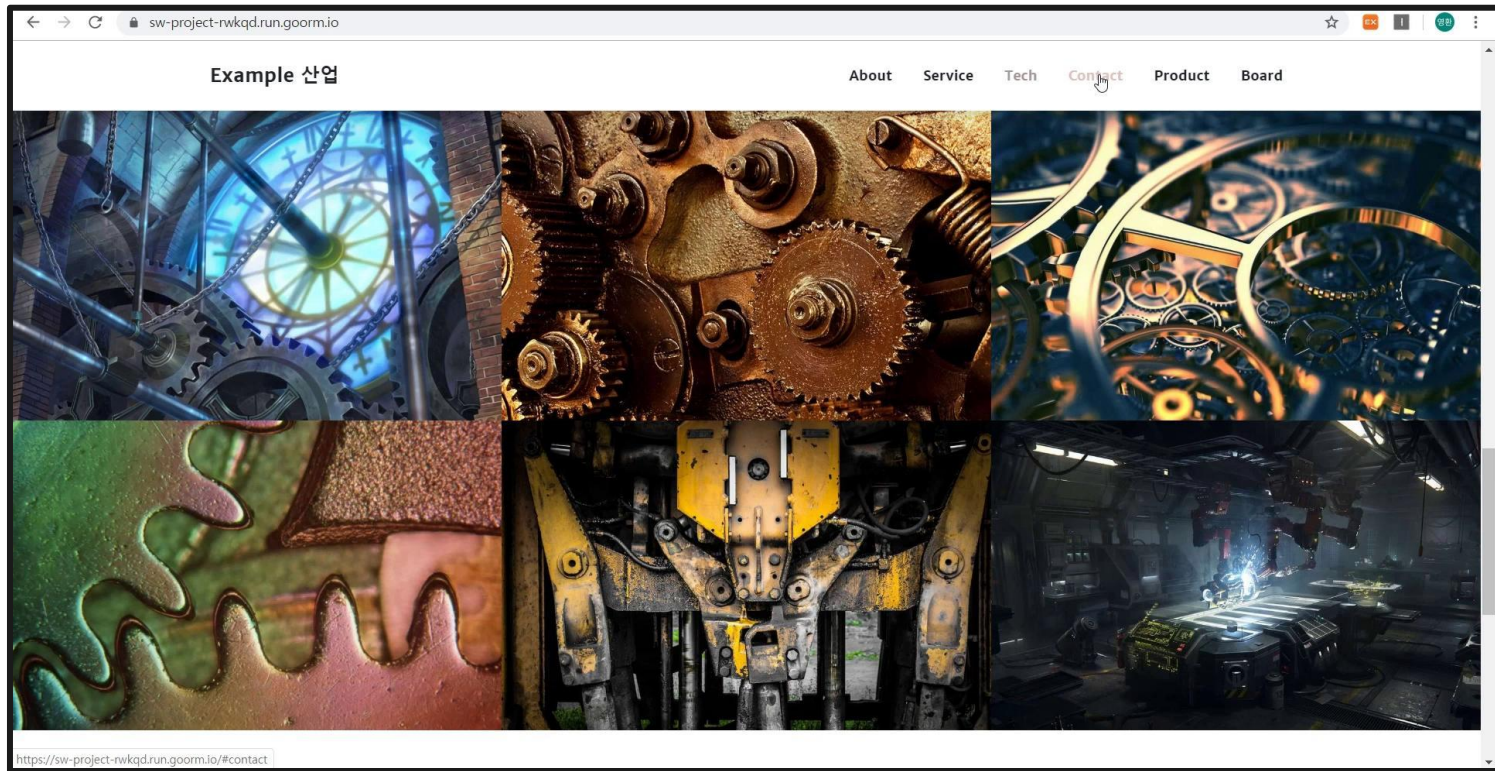




시연 영상

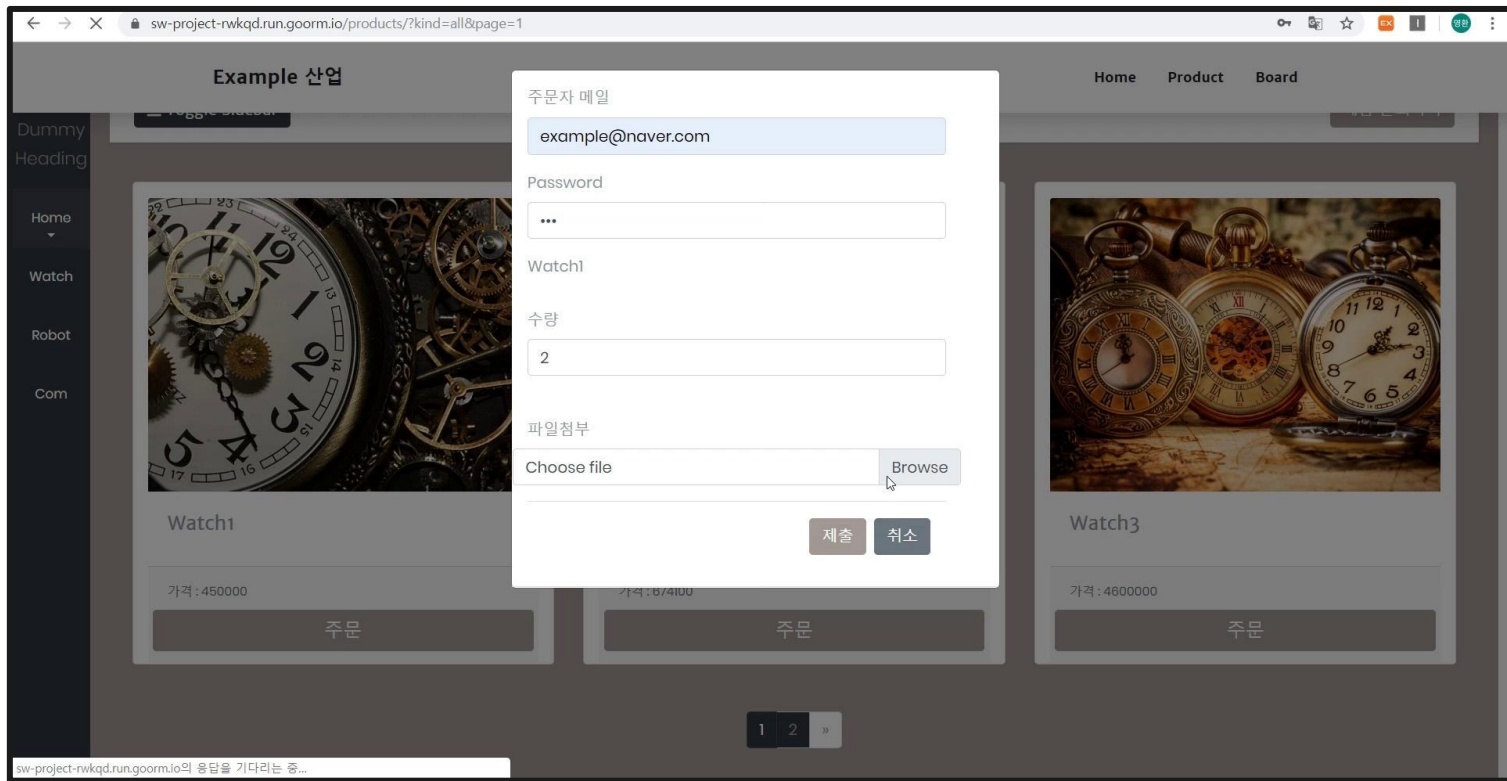
User 시나리오

세부목차로
이동



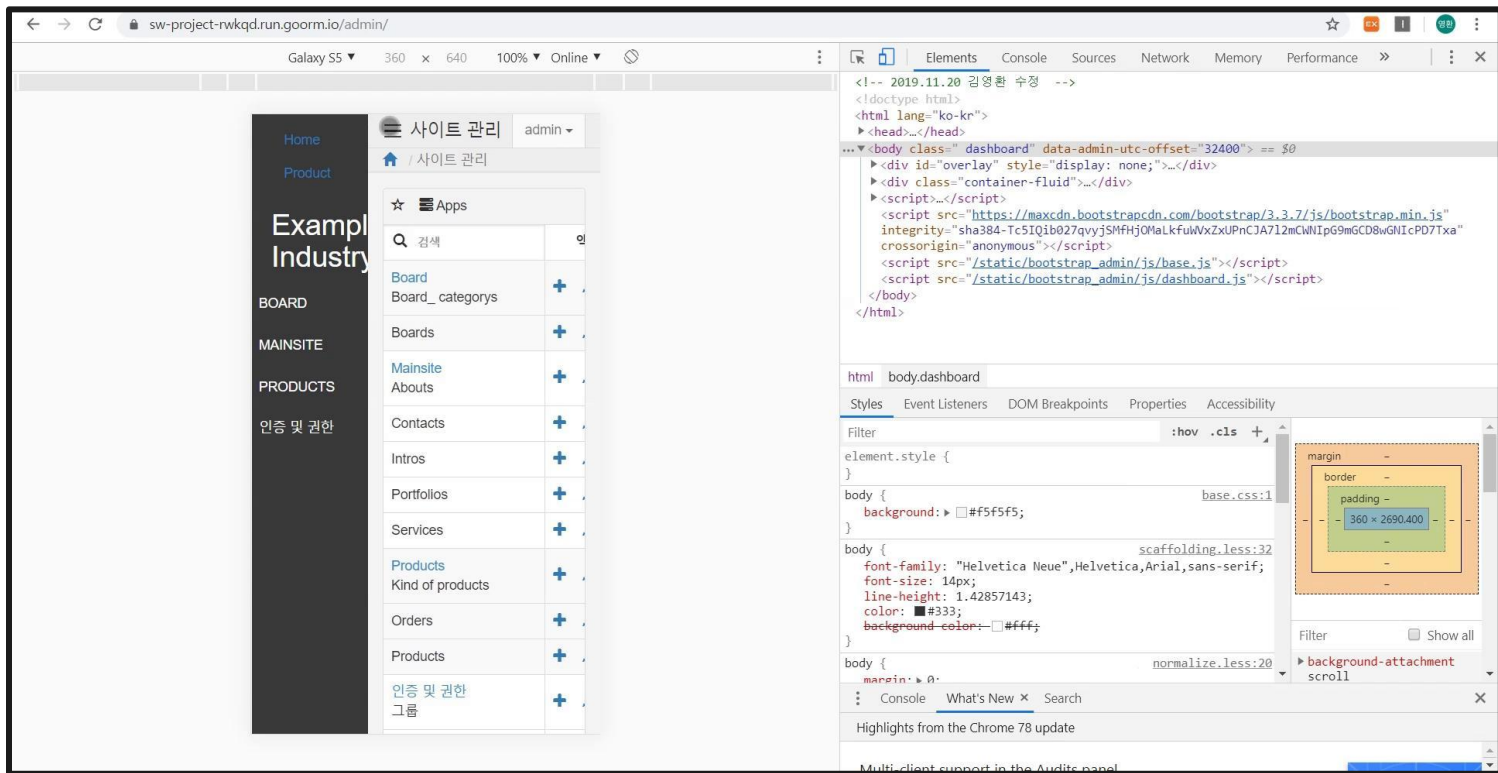
User 시나리오

세부목차로
이동



Admin 시나리오

세부목차로
이동



목차로 돌아가기

자체평가

자체평가 기준

→ 요구자의 요구사항을 만족하였는가?

◆ 전반적인 요구자의 요구사항들은 만족을 하였고, 추가적으로 팀이 지원 할 수 있는 부분들에 대해서 지원함.

→ 요구자가 원하는 형태로 만들어졌는가?

◆ 요구자가 생각했던 부분과 약간 다른 부분들이 있긴 하였지만 지속적인 소통을 통하여 원하는 형태로 맞춰옴.

→ 제작하는데 어느 정도의 시간과 노력이 소비되었는가?

◆ 시간이 촉박한 탓에 적은 시간에 최대의 노력을 투자하여 작성을 함.

→ 제작자가 보았을 때 완성도가 높은가?

◆ 기능적인 부분에서는 어느 정도 구현이 되었으나 시간적인 부분이 충분하다면 조금 더 사용자가 편하게 사용할 수 있게 만들 수 있을 텐데 이런 부분에서 조금 아쉬움이 있음.

목차로 돌아가기

Questions?