



# What's in My cart?

위치기반 식료품 쇼핑대행 서비스

# 목차



01

프로젝트 소개 및 참여멤버

02

개발 스택

03

Web 기획 스토리보드

04

주요기능 분석

05

배포



# 프로젝트 소개

# About Project



## What's in My Cart?

“ 코로나 19로 직접 접촉이 꺼려지는 환경에서 실시간 배송 서비스들이 많이 생기는 시기에 'E-mart, 홈플러스 등의 대형 오프라인 마트에서 어플리케이션을 이용하여 실시간 배송 서비스를 제공하면 어떨까?' 라는 생각으로 제작하게 되었습니다. ”



### Client

위치기반으로 주변 마트의 물품을 구매하는 고객으로서 Shopper의 Delivery서비스를 제공받는 대상입니다.



### Shopper

Client에게 배달서비스를 제공해주는 대상입니다. Client가 지불한 Delivery Cost를 받고 배달 서비스를 대행해줍니다.



### Admin

Client에게 QnA를 통해 웹사이트 관련 문제를 해결하고, DashBoard를 통해 다양한 통계치를 활용합니다.



# Contributors



안경은

✉ anruddms8124@gmail.com

모든 위치기반 서비스 구현  
AWS RDS, EC2 등 환경 구축  
및 배포 담당  
Git 병합 및 프로젝트 관리



신다혜

✉ alrean@naver.com

장바구니, 위시리스트,  
주문서 관리 및 매칭,  
쇼퍼 평점, 포인트, 결제구현



남윤서

✉ namys0114@gmail.com

상품 데이터  
크롤링 및 관리  
Shop, Q&A Page  
기능 구현



염가희

✉ ykh4933@naver.com

로그인, 회원가입  
스프링시큐리티  
채팅기능  
Admin Page





# 개발 환경





# Web 기획 스토리보드

# Storyboard - Client

주소 입력시 하단에 관련 주소들이 출력 (API 사용)

What's in my cart?  
편리한 배달 서비스를 이용하세요!

🔍 Address or Zipcode

계속하기

로그인하러가기

button 클릭시 입력한 주소의 좌표를 session에 보관한다

로그인창 이동

(로그인성공)  
메인 화면으로 이동

What's in My Cart?

What's in My Cart? 00일 환영합니다. 9000points Logout

WM 로고 Home Shop MyPage Contact

대한민국 서울특별시 서초구 강남대로 근처에 있는 마트

|                        |                      |                     |                         |
|------------------------|----------------------|---------------------|-------------------------|
| <br>이마트세종시가지점<br>0.7km | <br>홈플러스도곡점<br>0.9km | <br>CU명파동점<br>0.9km | <br>GS25신도곡오성점<br>0.3km |
| <br>롯데수유점<br>1.7km     | <br>미니샵<br>0.7km     |                     |                         |

첫화면에서 입력한 주소

본인 위치기반 지역내 마켓 보여주기  
지점, 거리 등을 표기

마트별 카테고리 이동

WM 로그인  
로그인에 필요한 이메일과 비밀번호를 입력하세요

UserEmail

Password

로그인

구글 로그인

계정이 없으신가요? 회원가입

비밀번호를 잊으셨나요? 비밀번호 찾기

유효성 검사 후,  
성공시 Security 상  
태로 로그인

구글 로그인  
API 사용

회원가입창으로 이동

비밀번호 찾기로 이동

체크박스)체크X -> 일반회원 가입  
체크O -> 쇼퍼로 가입

무료로 WM에 가입해보세요!

☒ shopper로 회원가입

이름 입력

UserEmail

Password

이메일로 회원가입

구글로 회원가입

이미 계정이 있으신가요? 로그인

What's in my cart?  
비밀번호를 잊으셨나요?  
이메일을 입력하시면 초기화 비밀번호를 알려드립니다.

EmailAddress

비밀번호

로그인하러가기

회원가입하러가기





위시리스트 추가

장바구니 추가

크롤링한 아이템들 foreach로 정렬



# Storyboard - Client

마트페이지로 이동

## 1.계정관리

My info

이름

이메일

주소

장바구니 상품 count 표기

장바구니 상품 정보를  
foreach로 정렬

비밀번호 입력후 탈퇴

비밀번호 입력후 수정,  
이메일은 readonly

## 2.장바구니

총 1개의 상품

item Photo

amount  
바나멜라 프레쉬 바나나 (총)  
장바구니에서 보기 2,900원

The total amount of  
물품 가격 2,900원  
배송비 5000   
총 결제금액 7,900원

장바구니 테이블에서 삭제

장바구니로 이동

결제대기로 이동

## 3.주문서(주문내역, 결제대기, 과거주문서)

My Order List

1번 주문서  
아직 shopper와 매칭되지 않았습니다

2번 주문서  
☒ shopper@naver.com님과 매칭 되었습니다

if 문 조건)

주문서 매칭X -> 매칭되지 않았습니다!

주문서 매칭O -> ~님과 매칭되었습니다!

쇼퍼의 위치와 주문서의 클라이언트가 선택한  
마켓 거리를 가까운 순서대로 보여줌

채팅창으로 이동

주문하신 상품이 없으시네요!

주문서가 null일때  
표시되는 화면

Order List (결제대기/과거)

1번 주문서  
Tot 7,900원

주문서가 존재할때,  
주문서를 표시



Sidebar 내용 클릭시 Body부분이 바뀜

Sidebar 내용 클릭시 Body부분이 바뀜



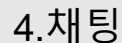
상세주소+요청사항을 DB에 저장

## 결제 API를 통해 구현

- checkBox 클릭시, 포인트를 구매금액에서 차감

- 세션에서 주소 호출

#### 4.우|시|리|스트

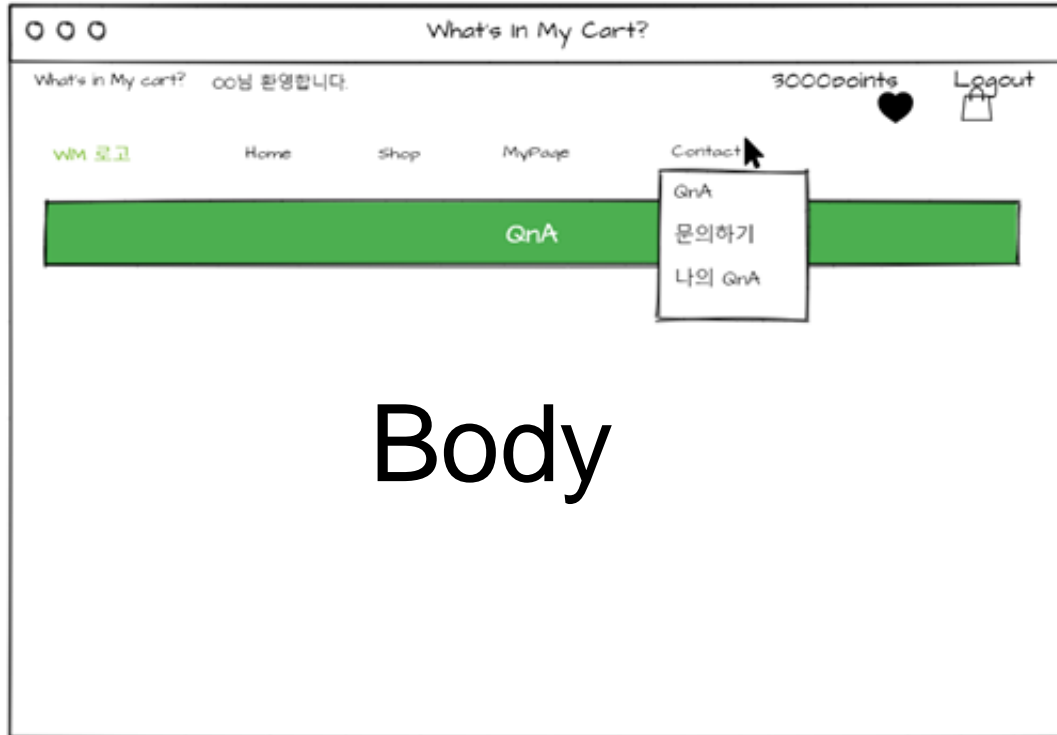
주문서별 채팅방 생성

채팅 Text입력하는 곳

과거 채팅내용 DB에  
저장하여 호출

위시리스트 아이টে을 DB에  
서 호출 foreach로 뿌림

# Storyboard - Client



Body

## QnA & 나의 QnA

| No | title | name | date    |
|----|-------|------|---------|
| 1  | 문의합니다 | 김회원  | 2008-12 |

제목 Q search 문의하기

Dropbox  
제목, 이름

제목, 이름 기준 QnA  
검색

문의하기

Leave Message

name email

제목

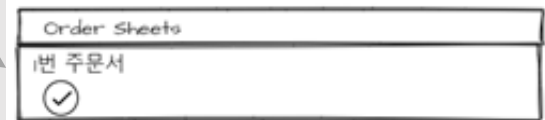
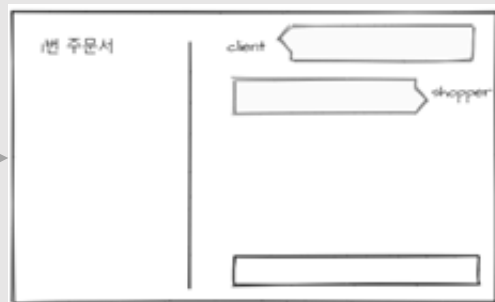
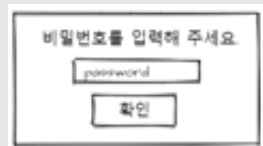
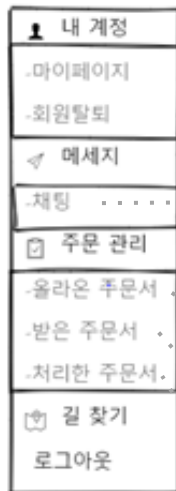
문의 사항을 입력하세요

문의하기

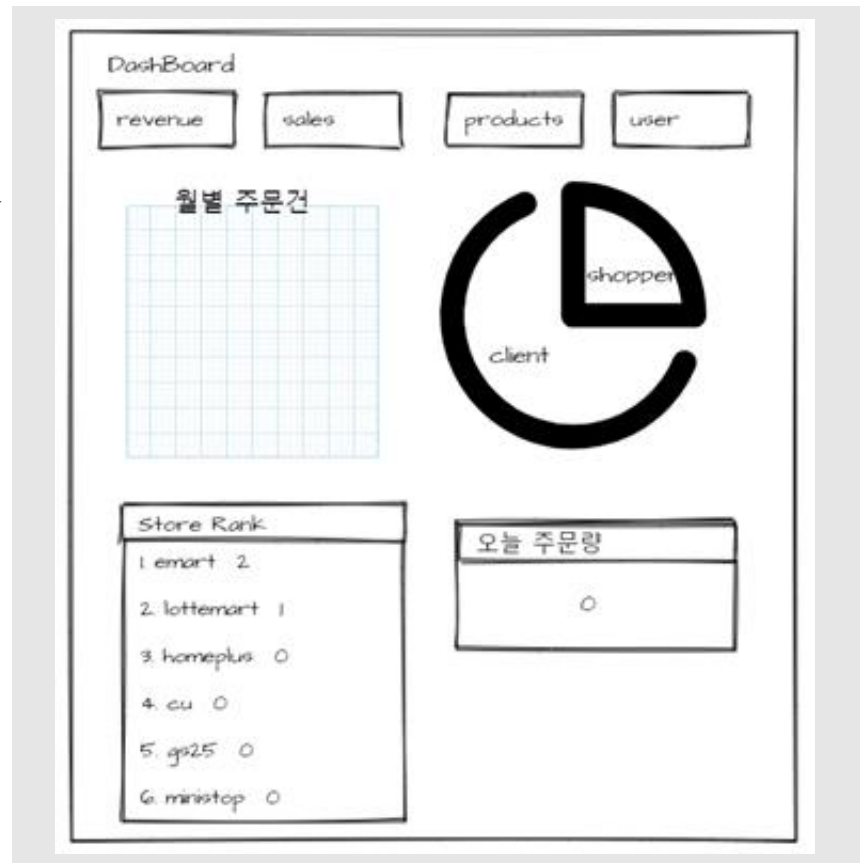
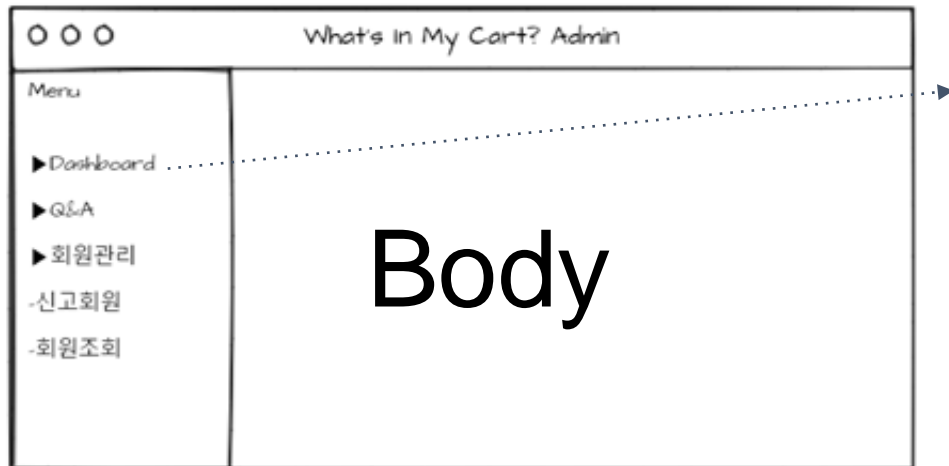
# Storyboard - Shopper



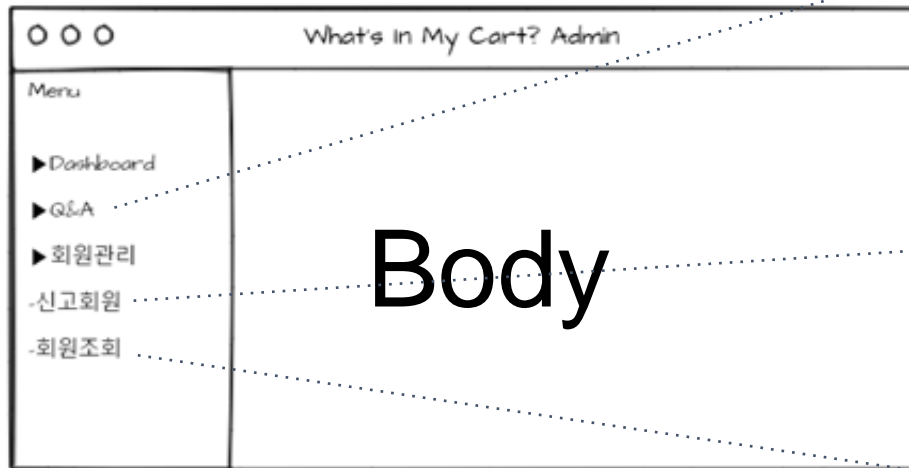
Sidebar dropBox



# Storyboard - Admin



# Storyboard - Admin



## QnA

Q&A 관리

| No | title | name | date     |
|----|-------|------|----------|
| 1  | 문의합니다 | 김회원  | 20/08/12 |

제목 Q search

## 신고회원

회원 신고 내역

Q search 이메일

| 번호 | 신고자 | 신고당한 회원 | 내용 | 날짜 | 처리버튼 |
|----|-----|---------|----|----|------|
|----|-----|---------|----|----|------|

## 회원조회

회원조회

Q search 이메일

| Email | Name | Address | Role | Enable | Setting |
|-------|------|---------|------|--------|---------|
|-------|------|---------|------|--------|---------|

이메일 혹은 아이디로 검색

회원 정지 유무    회원 정지 취소 버튼



## 주요기능 분석





# 주요기능

- 스프링 시큐리티로 로그인, 회원가입 및 불량회원 관리 ✓
- 사용자(Client) 위치 기반 마트 정보 ✓
- 사용자(Shopper) 위치 기반 주문서 리스트 ✓
- STOMP를 이용한 1 : 1 채팅 ✓
- 마트별 상품에 대한 크롤링 ✓
- 장바구니 담기 기능 ✓
- 주문서 제출 및 결제 기능 ✓
- Chart.js를 이용한 Admin Dashboard ✓



# 우편번호 검색/로그인/회원가입

What's in my cart?

편리한 배달 서비스를 이용하세요!

78

- 수정구 산성동 782 대한민국 경기도 성남시
- 수정구 단대동 78 대한민국 경기도 성남시
- 수정구 산성동 780 대한민국 경기도 성남시
- 중원구 은행동 782 대한민국 경기도 성남시
- 중원구 금광1동 자혜로 78-2 대한민국 경기도 성남시

powered by Google

What's in my cart?

비밀번호를 잊으셨나요?

이메일을 입력하시면  
초기화 비밀번호를 알려드립니다.

Email Address

비밀번호 초기화

로그인하러 가기

회원가입하러 가기

WM 로그인

로그인에 필요한 이메일과 비밀번호를 입력하세요.

Email Address

Password(min 6 characters)

로그인

or

구글 로그인

계정이 없으신가요? 회원가입

비밀번호를 잊으셨나요? 비밀번호 찾기

무료로 WM에 가입해보세요!

Shopper로 회원가입

Name

Email Address

Password(min 6 characters)

이메일로 회원가입

or

구글로 회원가입

이미 계정이 있으신가요? 로그인

웹 사이트 접속 시 제일 먼저 우편번호를 입력받고  
(위치기반에 서비스에 필요한 정보)  
로그인 및 회원가입을 진행한다.

## [주요 로직 및 고민]

- ✓ Name, Email, Pwd 각각 Validation Checking
- ✓ 비밀번호 암호화 (BcryptPasswordEncoder)
- ✓ 주소 입력 창 Google 주소 Api사용(자동완성)
- ✓ 비밀번호 찾기 : JavaMailSender
- ✓ 소셜 로그인 : 기존 회원인지 확인 후 update or save한다.

## [소스 코드]

- ✓ Javascript

<https://github.com/angyeongun/Web-project/blob/master/src/main/resources/static/js/app/signUp.js>

- ✓ Controller

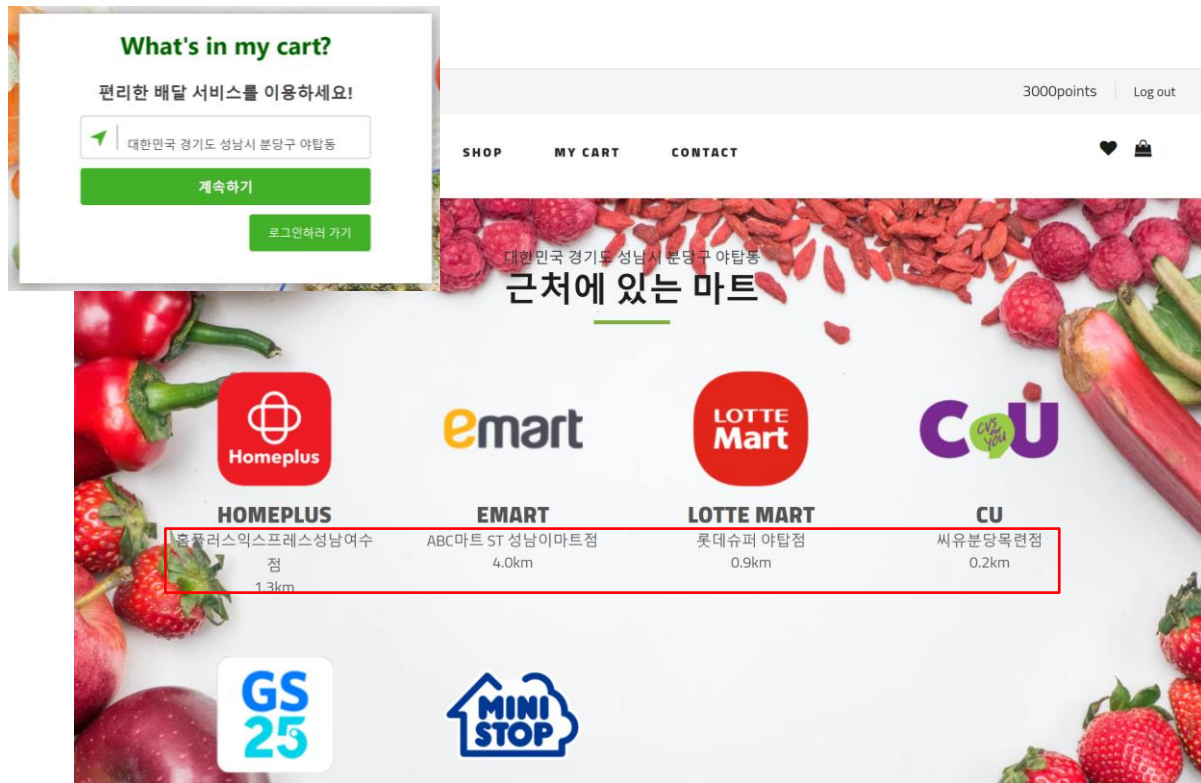
<https://github.com/angyeongun/Web-project/blob/master/src/main/java/com/cartrapido/main/web/MemberController.java>

- ✓ Service

<https://github.com/angyeongun/Web-project/blob/master/src/main/java/com/cartrapido/main/service/MemberService.java>

# Google Maps Places API 이용 사용자(Client) 위치 기반 마트 정보

클라이언트가 처음에 입력한  
우편번호(배송주소) 주변의 가장 가까운  
마트 정보를 나타낸다.



## [주요 로직 및 고민]

- ✓ 특정 위치로부터 마트까지 떨어져 있는 거리 계산
- ✓ 정확히 잡지 못하는 마트 이름  
(ex) ABC마트 ST 성남이마트점 등을  
정규표현식으로 제외시키기
- ✓ 렌더링 속도에 대한 고민(로직의 배치문제)

## [소스 코드]

- ✓ 해당 페이지 HTML

<https://github.com/angyeongun/Web-project/blob/master/src/main/resources/templates/clientWebBody/clientMain.html>

- ✓ 주요 로직 Javascript

<https://github.com/angyeongun/Web-project/blob/master/src/main/resources/static/js/ap/p/distanceApi.js>

# Google Maps Places API 이용 사용자(Shopper) 위치 기반 주문서 리스트

| Order Sheets   |  |
|--|--|
| <b>1번 주문서</b><br>고객의 마트가 현재위치로부터 2.2Km 떨어져 있음<br>By anruddms8124@gmail.com<br>✓  |  |
| <b>4번 주문서</b><br>고객의 마트가 현재위치로부터 8.5Km 떨어져 있음<br>By anruddms8124@gmail.com<br>✓  |  |
| <b>3번 주문서</b><br>고객의 마트가 현재위치로부터 12.8Km 떨어져 있음<br>By anruddms8124@gmail.com<br>✓ |  |
| <b>5번 주문서</b><br>고객의 마트가 현재위치로부터 16.1Km 떨어져 있음<br>By anruddms8124@gmail.com      |  |

쇼퍼의 현재위치(CurrentLocation)로 부터 클라이언트가 선택한 마켓간의 위치를 계산하여 가까운 순서대로 보여준다.

이 서비스는 궁극적으로 위치가 가까운 클라이언트와 쇼퍼끼리 매칭될 가능성이 높여 서비스의 회전률을 높이는 기능을 한다.

## [주요 로직 및 고민]

- ✓ 렌더링 속도에 대한 고민(로직의 배치 문제)
- ✓ 클라이언트가 여러 개의 마켓에서 상품을 골랐을 경우, 그 마켓들 중 쇼퍼에게 더 가까운 마켓과의 거리를 보여준다.

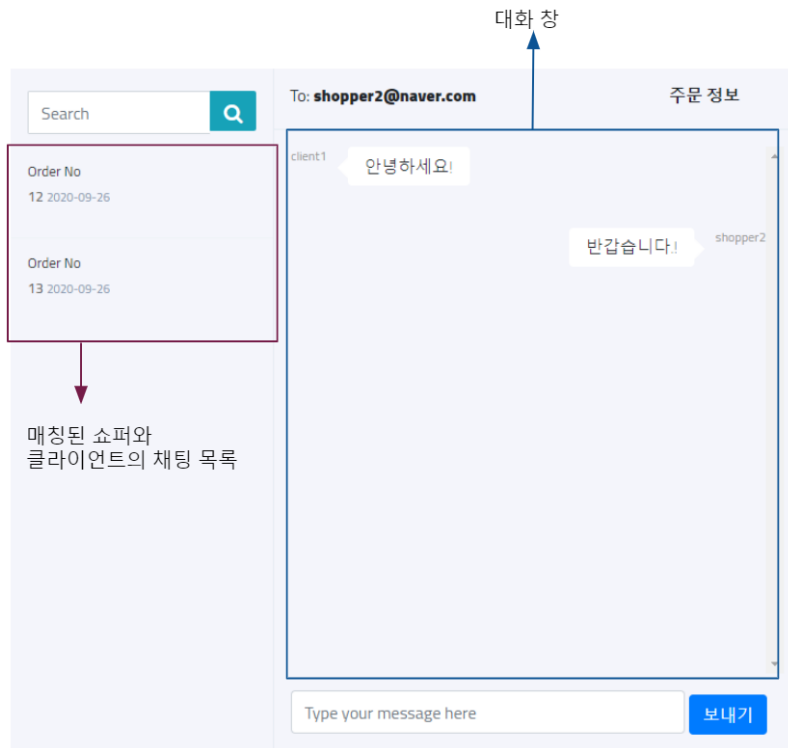
## [소스 코드]

✓ 해당 페이지 HTML  
<https://github.com/angyeongun/Web-project/blob/master/src/main/resources/templates/shopperWebBody/shopperList/orderSheetList.html>

✓ **Controller** - @GetMapping("/orderSheetList")  
<https://github.com/angyeongun/Web-project/blob/master/src/main/java/com/cartrapido/main/web/ShopperController.java>

✓ **주요 로직 Javascript**  
<https://github.com/angyeongun/Web-project/blob/master/src/main/resources/static/js/app/currentLocationApi.js>

# STOMP를 이용한 1 : 1 채팅



채팅기능을 통해 클라이언트와 쇼퍼가 소통을 하여 의견을 조율할 수 있다.

## [주요 로직 및 고민]

- ✓ Shopper와 Client간 커뮤니케이션 가능
- ✓ Shopper가 주문서 클릭했을때, 채팅창 생성
- ✓ Client가 배송 완료를 누르면 채팅방 삭제

## [소스 코드]

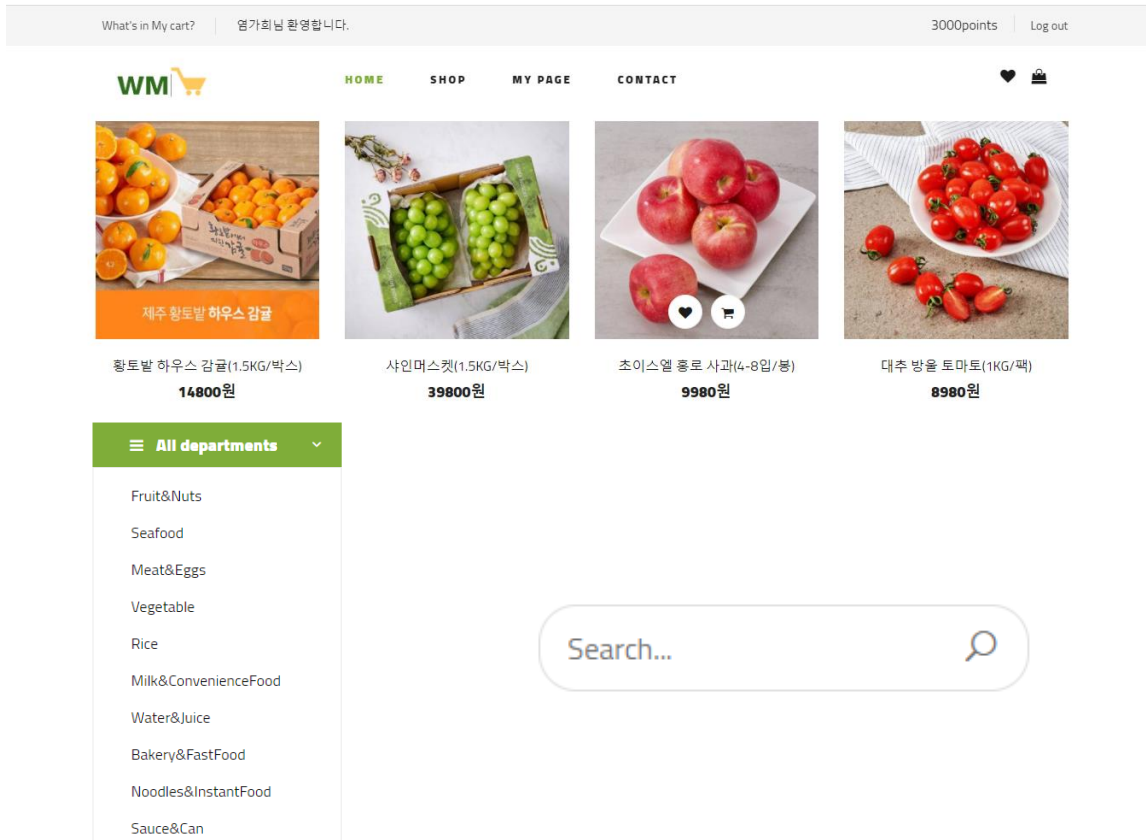
### ✓ 채팅

<https://github.com/angyeongun/Web-project/tree/master/src/main/java/com/cartrapido/main/chat>

### ✓ Javascript

<https://github.com/angyeongun/Web-project/blob/master/src/main/resources/static/js/chat/chatRoomList.js>

# 마트별 상품에 대한 크롤링



상품정보를 크롤링 하여 상품정보를 조회할 수 있도록 한다.

## [주요 로직 및 고민]

- ✓ 마트별 상품 가져온다.
- ✓ 상품 클릭시 상품 상세보기 가능
- ✓ 카테고리별 물건 볼 수 있음
- ✓ 카테고리에서 물건 검색 가능
- ✓ 크롤링 데이터 및 서버관리에 대한 고민

## [소스 코드]

- ✓ 해당 페이지 HTML

<https://github.com/angyeongeun/Web-project/blob/master/src/main/resources/templates/clientWebBody/mart.html>

- ✓ 크롤링 Controller

<https://github.com/angyeongeun/Web-project/blob/master/src/main/java/com/cartrapido/main/web/ClientController.java>

- ✓ 크롤링 Service

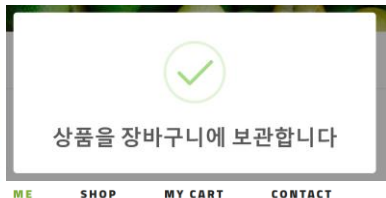
<https://github.com/angyeongeun/Web-project/tree/master/src/main/java/com/cartrapido/main/service/crawlingService>

# 장바구니 담기 기능



[피코크] 딸기 그대로 10g

2480원



## What's in My cart?

### My cart

내정보

장바구니

위시리스트

주문 내역

결제 대기

지난 결제내역

### 장바구니

총 1개의 상품



emart  
[피코크] 딸기 그대로 10g  
[장바구니에서 빼기](#)

2480

- 1 +

결제 내역

상품 가격 2480 원

서비스 비용 5000 [edit](#)

총 합계 7480 원

주문서 제출

장을 볼 마트 상품 페이지에서 상품을 장바구니에 담는 기능. 마이페이지나 장바구니 모양 클릭을 통해 본인이 담은 상품 목록 확인 가능

### [주요 로직 및 고민]

- ✓ 물건 담기 및 찜하기 가능
- ✓ 장바구니에서 바로 주문서 제출 후 결제 기능 구현

### [소스 코드]

✓ 해당 페이지 HTML

<https://github.com/angyeongun/Web-project/blob/master/src/main/resources/templates/clientWebBody/shoppingCart.html>

✓ Controller


<https://github.com/angyeongun/Web-project/blob/master/src/main/java/com/cartrapido/main/web/ClientController.java>

✓ Service

<https://github.com/angyeongun/Web-project/blob/master/src/main/java/com/cartrapido/main/service/CartService.java>

# 주문서 제출 및 결제 기능

1개의 품목을 주문하셨습니다!



emart

[피코크] 딸기 그대로 10g

2480원 \* 1개

☐ 포인트 사용 0

적용

3000 points

| 상품가격  | 서비스 가격 | 총 합계  |
|-------|--------|-------|
| 2480원 | 5000원  | 7480원 |

주소

야탑

상세주소

추가 요청사항

추가 요청사항을 기입하세요.

☐ (선택 사항)

택 사항) 소퍼가 대체 상품으로 구매할 경우, 구매 지역을 마일리지에서 차감하는데 동의합니다.

저장하기

주문서 삭제

카카오페이 결제하기

신용카드 결제하기

임시 결제버튼

장바구니에 담은 상품의 구입 및 배송 대행을 위해  
주문서를 발주한다.  
결제가 완료되면 주문서는 쇼퍼가 확인 가능하다

## [주요 로직 및 고민]

- ✓ 주문서의 정보를 쇼퍼 페이지로 넘기기
- ✓ API를 이용한 결제 기능 구현
- ✓ 클라이언트의 대체 상품에 대한 고민을 마일리지 차감  
방식으로 해결

## [소스 코드]

### ✓ 해당 페이지 HTML

<https://github.com/angyeongun/Web-project/blob/master/src/main/resources/templates/clientWebBody/clientView/payMyOrder.html>

### ✓ Controller

<https://github.com/angyeongun/Web-project/blob/master/src/main/java/com/cartrapido/main/web/ClientController.java>

### ✓ Service

<https://github.com/angyeongun/Web-project/blob/master/src/main/java/com/cartrapido/main/service/OrderSheetService.java>



# Chart.js를 이용한 Admin Dashboard

현재 판매현황과 유저수를 보여주는 관리자 대시보드  
[주요 로직 및 고민]

- ✓ Chart.js를 이용해 현재 판매현황과 유저수를 보여주기
- ✓ 월별 주문량 관리
- ✓ 수익률은 배달료의 5%를 가져간다고 가정
- ✓ 서비스 비용은 얼마정도가 적절한가?
- ✓ 어떤 지표들을 중심으로 측정하는게 적합한가?

## [소스 코드]

- ✓ 해당 페이지 HTML

<https://github.com/angyeongun/Web-project/tree/master/src/main/resources/templates/admin>

- ✓ Controller

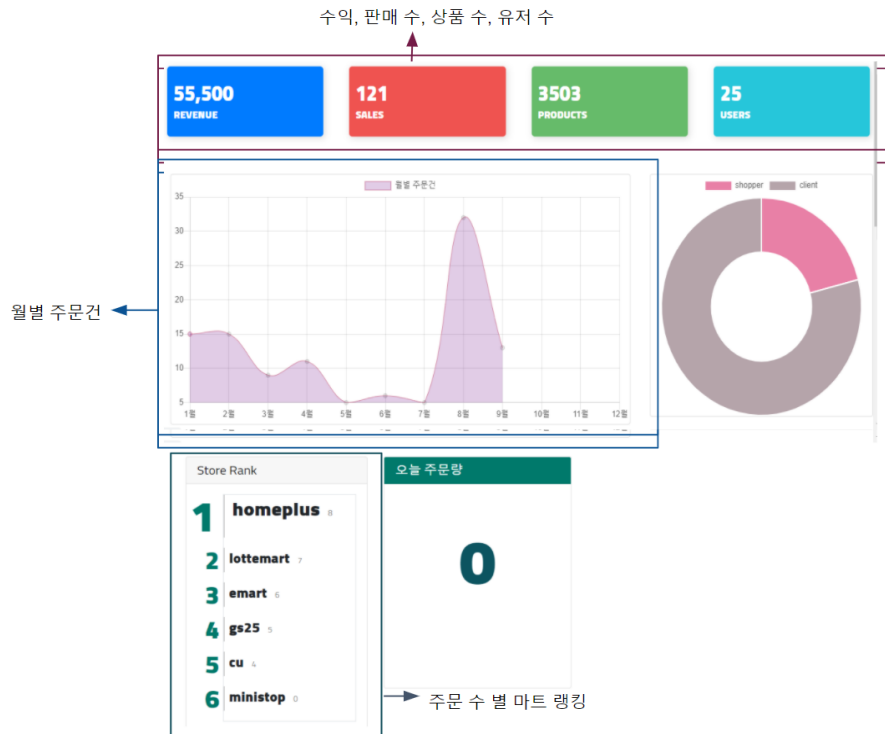
<https://github.com/angyeongun/Web-project/blob/master/src/main/java/com/cartrapido/main/web/AdminController.java>

- ✓ Service

<https://github.com/angyeongun/Web-project/blob/master/src/main/java/com/cartrapido/main/service/OrderNumHistoryService.java>

- ✓ DashBoard Javascript

<https://github.com/yeongahui/Web-project/blob/master/src/main/resources/static/js/admin/dashboard.js>

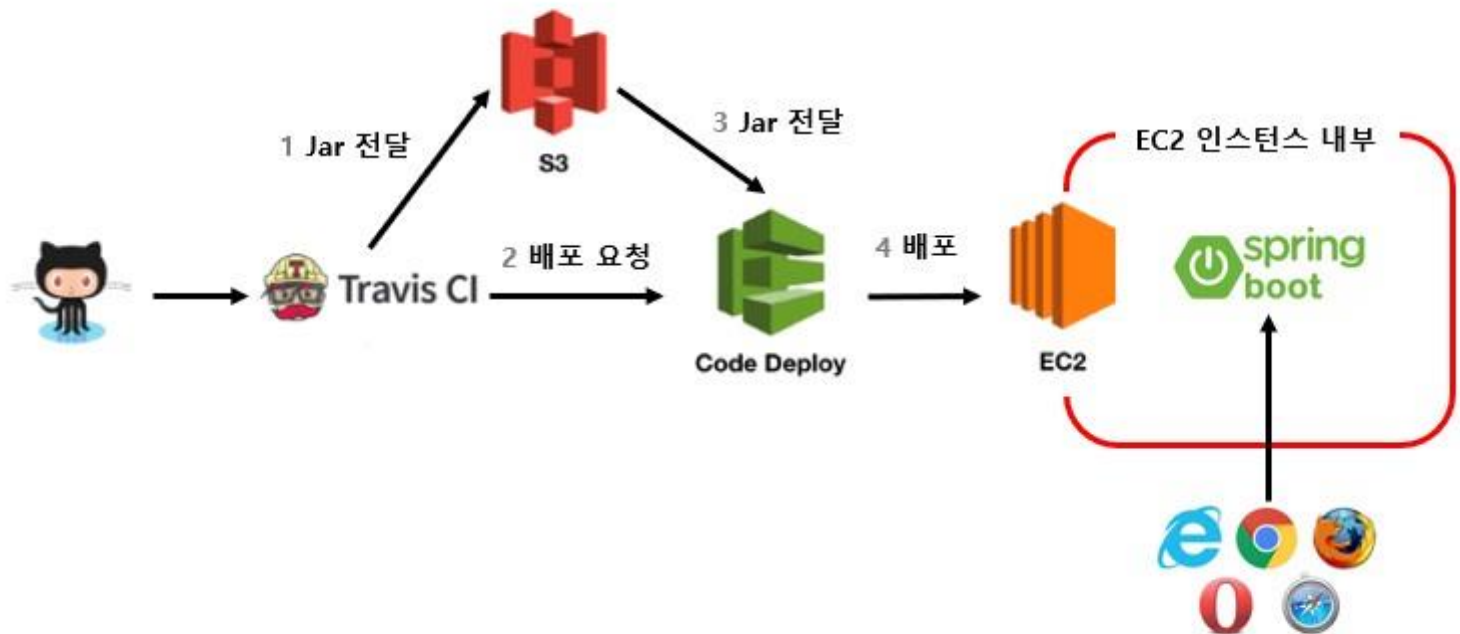




배포

# Travis CI 배포 자동화 / 연동 구조

배포 서버 : <http://ec2-3-34-159-150.ap-northeast-2.compute.amazonaws.com:8080>



**Thank you**

