

★ JUGAN  COFFEE ★

주간 커피

{ 리액트 & 스프링부트 기반 X 커피 구독 서비스 }

프로젝트 주제 및 동기

- **프로젝트 주제**

리액트 & 스프링 부트 기반

커피 원두 구독 서비스 & 클라이언트 서비스

- **프로젝트 동기**

1. 코로나 이후 홈카페의 수요 증가
2. 온라인 판매에 의한 소상공인 상생 가능 방안
3. 온라인 구독 경제 활성화

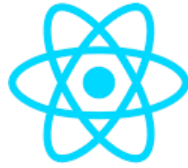
프로젝트 개요

사용기술

통신



프론트엔드 프레임워크 / 라이브러리



백엔드 프레임워크 / 라이브러리



Spring Data JPA

클라우드 서버



Amazon EC2



Amazon RDS

사용 언어



데이터베이스



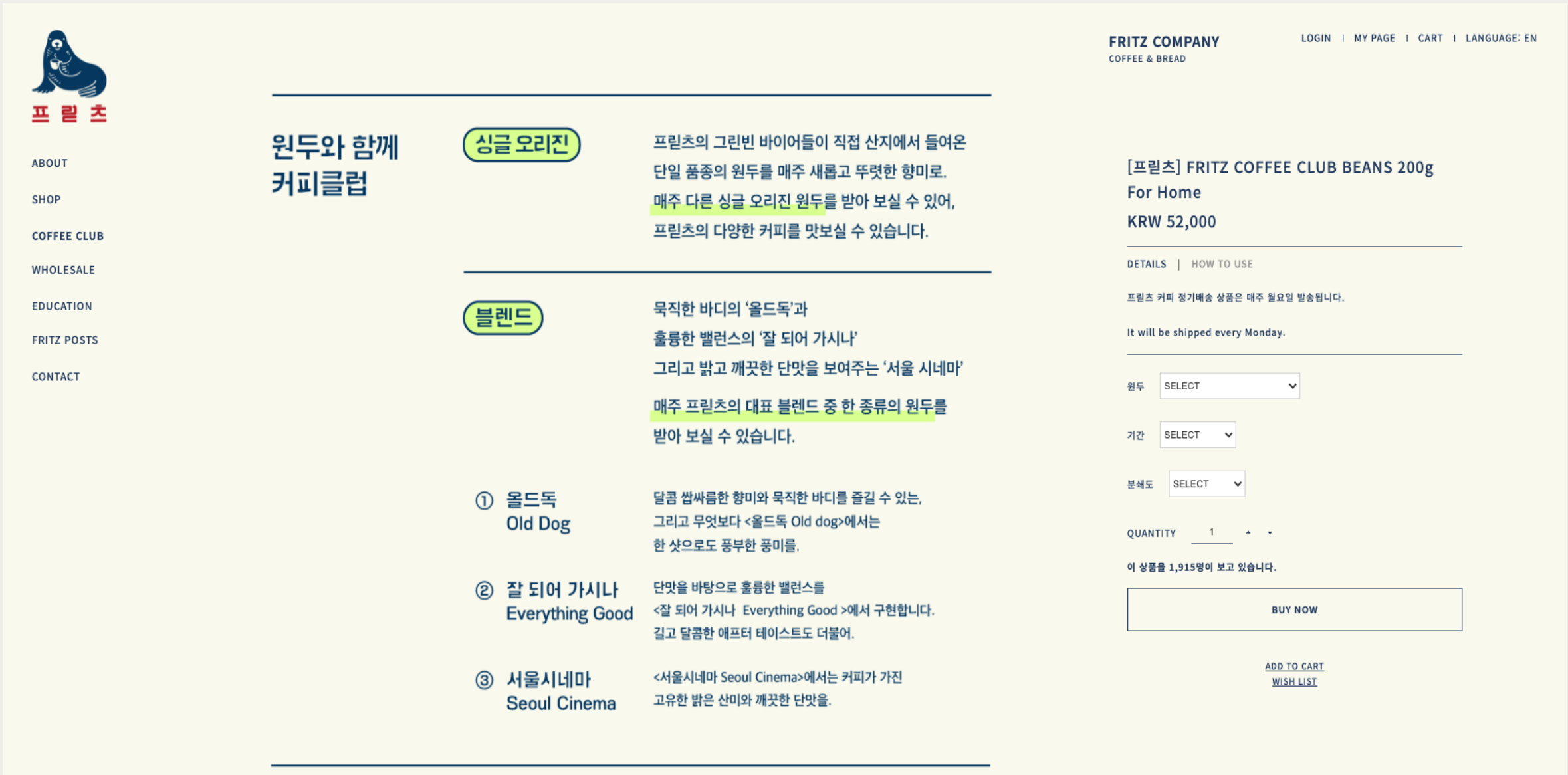
PostgreSQL

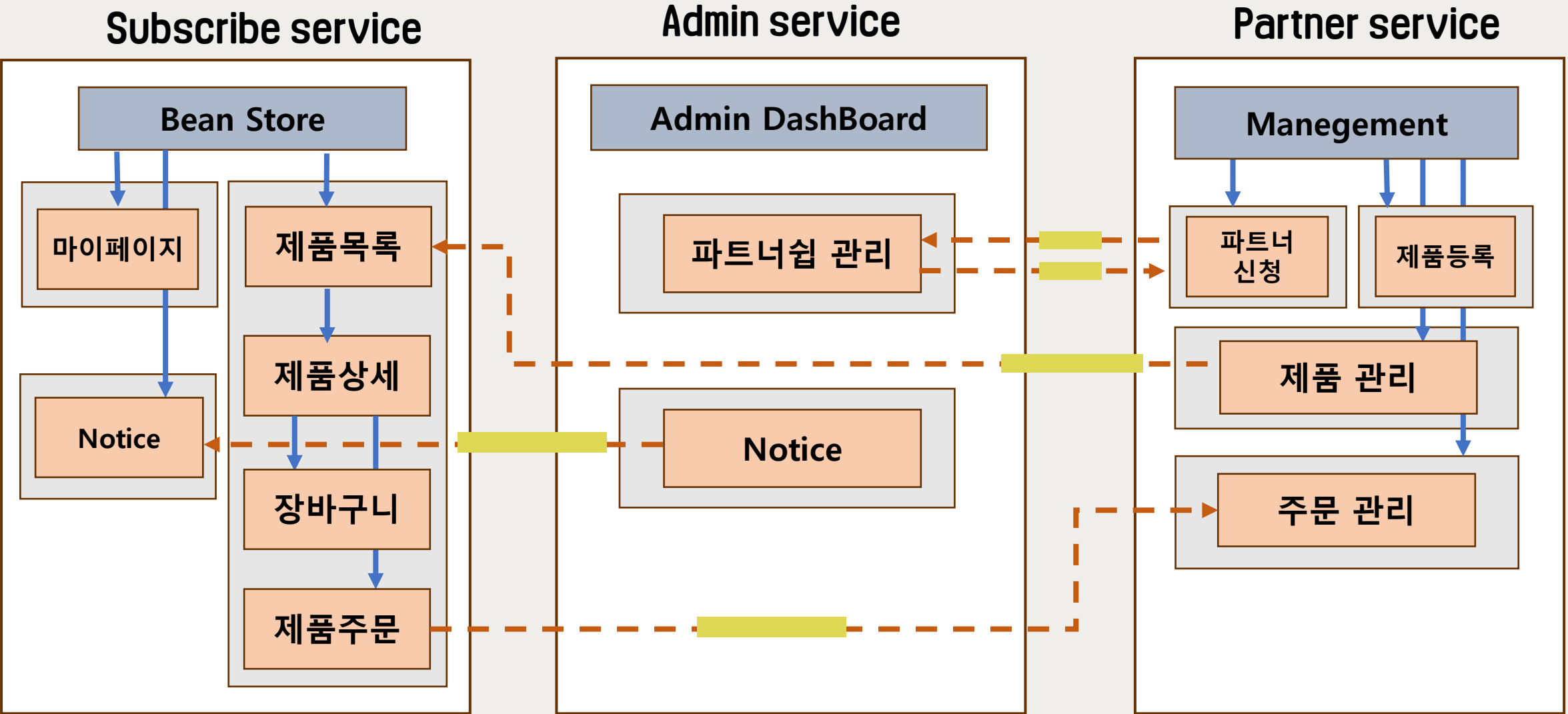
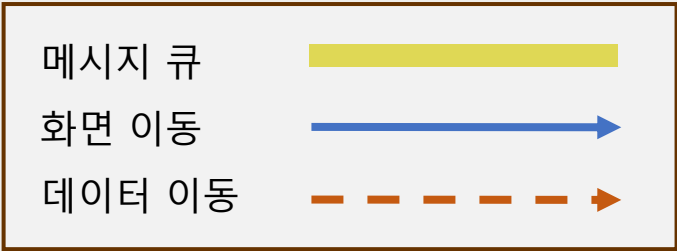
프로젝트 개요 사례분석 - [코케]

- 원두 구독 서비스를 운영하는 기업을 모델링 하여 프로젝트를 구상했습니다.
- 클라이언트 서비스가 구현해야 할 부분을 조사하여 반영하였습니다.



- 원두를 판매하는 로스터리의 홈페이지를 참고하여 UI를 설계했습니다.
- 원두 구독(판매) 시 필요한 데이터를 수집하였습니다.





프로젝트 일정	1주차	2주차	3주차	4주차	5주차	6주차	7주차	8주차
기획								
분석 및 설계								
단위 서비스 구현								
단위 테스트								
통합 테스트 및 배포								
산출물 정리								

프로젝트 개요 프로젝트 역할 분배



Project Leader

이주은

구독자 서비스

협업 툴 관리 (Notion)
프론트엔드, 백엔드
UI 디자인



Developer

명재윤

파트너 서비스



Developer

김준호

어드민 서비스

BeanStore Service			
BeanStore Home	홈페이지	BeanStore 시작화면 - 상단 : 네비게이션바를 보여준다. - 메인 : 간단한 브랜드 소개, 시작하기 [배경이미지]를 보여준다.	이주은
	네비게이션	네비게이션바 - 네비게이션바 카테고리를 표시한다. [제품 목록], [마이페이지], [contact us]	
Notice	Notice	Notice 를 클릭하면 Notice 상세화면이 나온다 - 공지사항 - 어드민에서 작성한 공지사항을 볼 수 있다, 테이블 형태로 보여진다. - 페이지를 넘길 수 있다.	
제품목록	제품목록	제품 목록버튼 클릭시 목록 화면으로 이동한다. - 제품 목록을 조회한다. - 4x4의 [그리드 스타일]로 보여준다. - 각 제품의 사진밑에 이름과 가격 등을 보여준다.	
제품상세	제품 상세	제품을 클릭하면 해당 제품정보 및 구독 옵션 선택 창으로 이동한다. - 제품정보 [제품사진, 원두정보, 판매자 정보, 좋아요버튼]을 보여준다. - 구독옵션 [가격, 원두용량선택, 구독기간 선택, 분쇄도 선택, 수량, 총 결제금액]을 보여준다.	
구독 상세	구독 신청	구독 신청 버튼 클릭시 구독신청 상세 화면으로 이동한다. - [입력 폼 스타일]로 화면에 보여준다. - 주문정보 확인, 이름, 카드정보, 배송지, 전화번호를 입력할수 있는 화면을 보여준다.	
마이페이지	마이페이지	마이페이지 버튼 클릭시 해당 화면으로 이동한다. - 현재 내 주문정보 확인 [테이블 형태] - 테이블 각 행 마다 주문한 제품의 사진, 제품명, 결제금액, 주문요청결과를 확인할 수 있다. - 해당 주문정보를 클릭하면 내가 주문한 제품의 주문옵션을 확인할 수 있다.	

WBS

[illegible]

분석 및 설계 테스트케이스 명세서

- 사용자의 관점에서 이벤트의 흐름, 선행조건, 예상결과를 시나리오로 정리했습니다.

1. 테스트 케이스(Test/Use Case)

▣ 대괄호: Entity Object, 실제 데이터 속성을 가지고 있는 객체, 주로 테이블과 매핑된 엔티티/리포지터리 객체

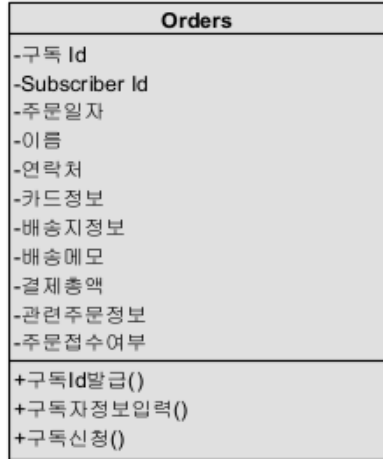
▣ 중괄호: Control Object(시스템), 요청 및 응답에 대한 흐름을 처리하는 객체, 주로 API 컨트롤러/서비스 객체

▣ 소괄호: Boundary Object: 사용자와 시스템 간에 경계를 처리하는 객체, 주로 UI객체

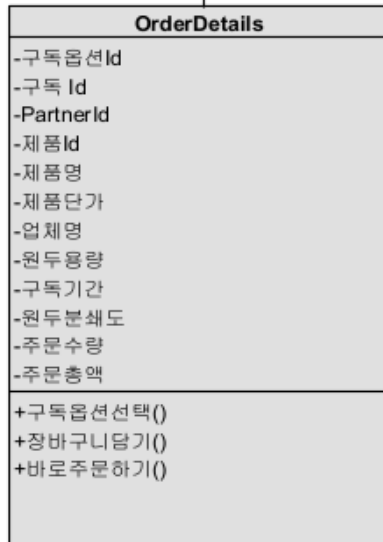
No	케이스 명	액터	이벤트 흐름	선행조건	예상결과
1.	홈페이지 접속	subscriber	1. [구독자]가 (Beanstore Home)에 접속 2. 상단에 (제품목록), (마이페이지), (Contact US) (장바구니) 항목이 있는 (네비게이션바) 출력 3. (네비게이션바) 하단에 (브랜드 소개), (배경이미지) 출력	홈화면을 (Beanstore Home)으로 지정해야함	(네비게이션바)를 통해 [구독자]가 페이지 이동을 할 수 있음
2.	제품목록 조회	subscriber	1. [구독자]가 네비게이션바의 (제품목록) 메뉴를 클릭 2. (제품목록) 화면으로 이동 3. (제품목록시스템)이 전체 [제품정보] 목록을 출력 4. 모든 [제품정보] 목록을 (4X4 그리드 형태)로 출력하고, 각 그리드에 (가격), (제품명), (제품사진), (업체명) 출력	[제품정보] DB테이블이 있어야함 [Partner]와 메시지 큐로 통신할 서버가 있어야함	모든 [제품정보] 목록을 (4X4 그리드 형태)로 출력 (가격), (원두이름), (제품사진), (업체명) 등 요약한 정보를 출력
3.	제품상세 조회	subscriber	1. [구독자]가 (제품목록) 중 한 제품을 클릭 2. (제품상세) 화면으로 이동 3. 화면의 왼쪽에 (제품사진), (제품명), (원두정보), (파트너정보), (하트아이콘)을 출력 4. 화면 오른쪽에 (구독옵션) 선택 창을 출력함 4-1. [구독자]는 (가격), (원두용량), (구독기간), (분쇄도), (수량)을 선택할 수 있고, 선택 사항에 따라 (총 결제금액)을 보여줌 [구독자]는 (구독옵션) 항목을 모두 선택하고 (구독하기) (장바구니) 버튼 클릭	[제품정보] DB테이블이 있어야함 (제품상세시스템)이 전체 [제품정보] 중 1개의 [제품정보]를 id로 조회해서 출력함 [구독자]는 (구독옵션) 항목을 모두 선택해야 다음 화면으로 넘어갈 수 있음	왼편에 한 건의 [제품정보]를 품 형식으로 출력 오른편에 (구독옵션) 선택 품 출력 (구독옵션)창은 오른쪽에 고정, (제품정보)는 왼쪽에서 스크롤 가능 (총 결제금액) 옆에 (구독하기) (장바구니) 버튼 출력 [구독자]가 옵션 선택 후 (구독하기) 클릭 시 (구독신청)으로 이동 (장바구니) 클릭 시 (장바구니시스템)이 해당 [주문정보]를 (장바구니)에 추가 (장바구니)를 클릭 시 (장바구니로 이동하시겠습니까 y/n)

				메시지 출력	
4.	장바구니	subscriber	1. {구독자}가 {네비게이션바}의 {장바구니}클릭 1-1. 혹은 {제품상세}의 {장바구니} -> {장바구니 이동} 클릭 시 {장바구니} 화면 출력 2. {장바구니시스템}이 {제품상세}에서 {장바구니}에 담았던 모든 {주문목록}을 테이블 형식으로 조회 3. 테이블 한 행에 {제품명} {수량} {가격} {업체명} {배송비}와 {주문하기} 버튼 출력 4. {전체주문} {선택주문} 버튼이 있고, {체크박스}를 클릭해서 원하는 목록만 선택 주문 가능	{장바구니시스템}에 들어갈 {주문정보} 임시데이터 테이블 있어야함 최소 한 개 이상의 {주문정보}를 {장바구니시스템}에 추가해야함	{장바구니}에 제품이 없으면 {제품없음} 메시지 표시 테이블 형식으로 {장바구니}에 담은 {주문목록} 표시 {체크박스}를 클릭해서 주문 별 선택 가능 {전체주문} {선택주문} 버튼으로 {구독신청시스템}에 주문 건 수 전송
5.	구독신청	subscriber	1. {제품상세}에서 {구독자}가 {구독옵션} 항목을 모두 선택하고 {구독하기} 버튼 클릭 1-1. 혹은 {장바구니} 하단 {구독하기} 버튼 클릭 2. {구독옵션}에서 선택한 옵션을 테이블 한 행에 출력 3. {주문자정보입력폼}을 {구독신청}에 출력 3-1. {구독자}는 {주문자정보입력폼}에 {연락처} {이름} {카드정보} {배송지정보} {배송메모}를 입력함 3-2. {구독자}가 {주문자정보}를 {주문자정보입력폼}에 입력 완료 4. {구독자}가 {구독} 버튼 클릭 5. {구독신청시스템}에서 위 정보를 {주문정보} DB에 추가. 6. 페이지 이동하여 {주문완료} 메시지를 보여줌	{구독자}는 {구독옵션}과 {주문자정보입력폼} 모두 기입해야함 {주문정보} DB테이블이 있어야함 {Partner}/{Admin}와 메시지큐로 통신할 서버가 있어야함 {구독신청시스템}이 구독을 진행하는 {구독자}에게 임의의 {구독id} 발급	{구독신청시스템}의 DB에 {구독자}가 등록한 {주문정보}의 데이터가 들어감 {Partner}/{Admin}에게 각각 {주문정보}를 보내줌 페이지 이동하여 {주문완료} 메시지를 보여줌
6.	마이페이지	subscriber	1. {구독자}가 {마이페이지} 메뉴 클릭 2. {마이페이지시스템}이 모든 {주문정보} 항목을 테이블로 출력 3. {주문정보} 테이블 한 행에 {제품사진} {제품명} {결제가격} {구독기간}을 출력함	{구독신청시스템}에서 저장한 {주문정보} DB테이블이 있어야함 {주문정보} 한 행을 클릭하면 {주문상세}로 넘어가도록 링크 연결	{마이페이지}목록에 {마이페이지시스템}이 전체 {주문정보}를 테이블 형식으로 화면에 출력
7.	주문상세	subscriber	1. {구독자}가 {마이페이지}에서 {주문정보}테이블 중 한 행을 클릭 2. {주문상세시스템}이 1건의 {주문정보}를 {주문id}로 조회해서 {주문상세} 화면에 출력 3. {주문상세} 화면에 테이블 형식으로 {제품사진} {제품명} {수량} {결제가격} {구독기간} {배송지정보}를 출력 3-1. {배송지정보}는 {수령인} {연락처} {배송지} {배송메모} 포함	{구독신청시스템}에서 저장한 {주문정보} DB테이블이 있어야함 {주문정보} DB테이블에 {주문id}가 포함 되어야함	{구독자}가 입력한 {주문정보} 중 1건을 {주문상세}에서 보여줌

주문 정보



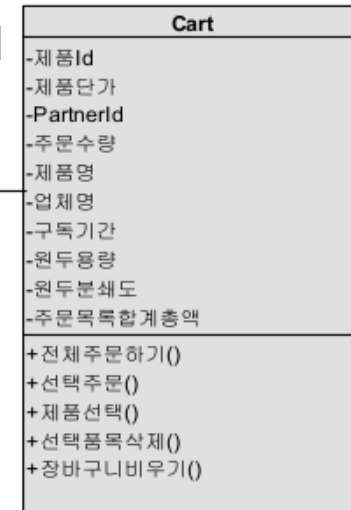
옵션 정보



제품 정보

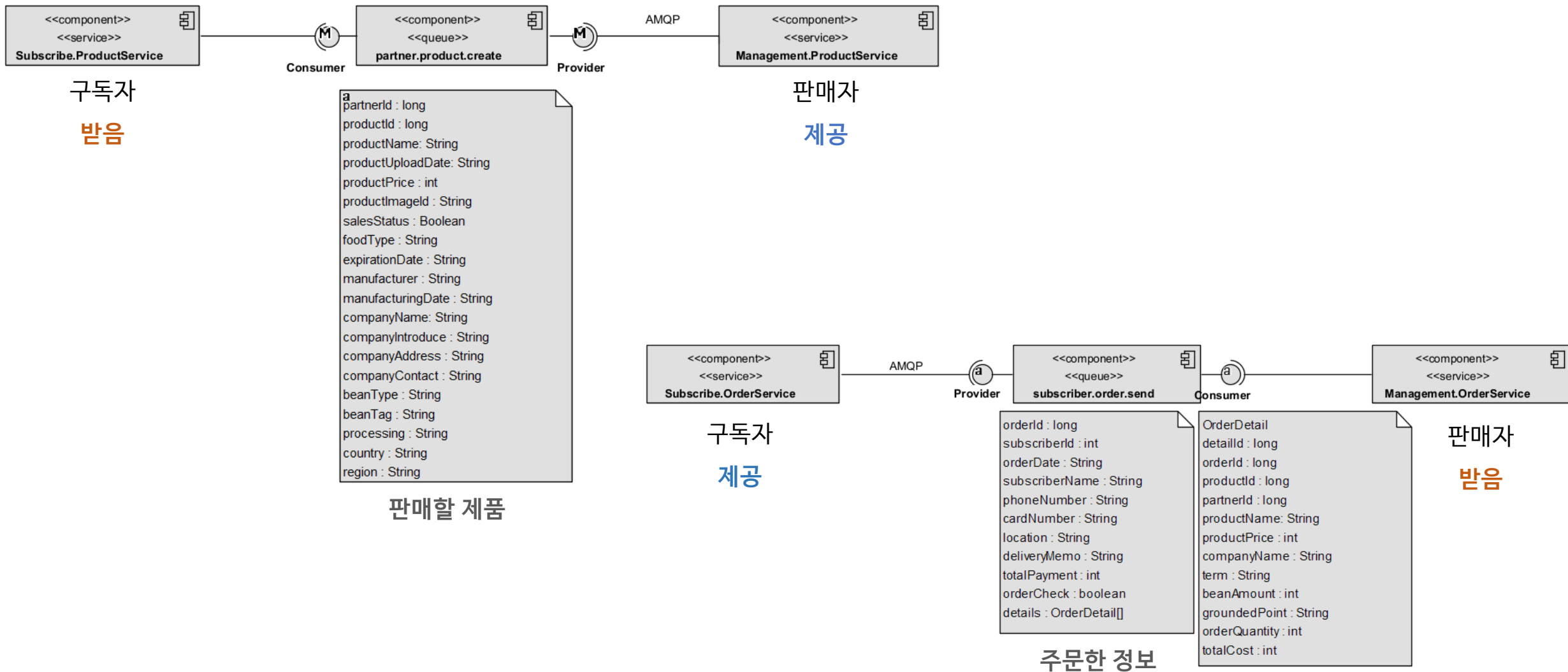


장바구니

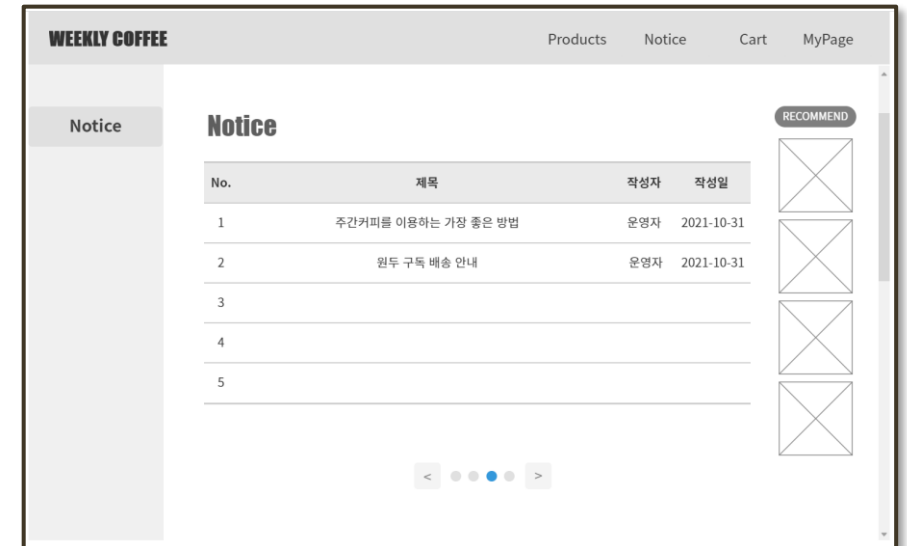
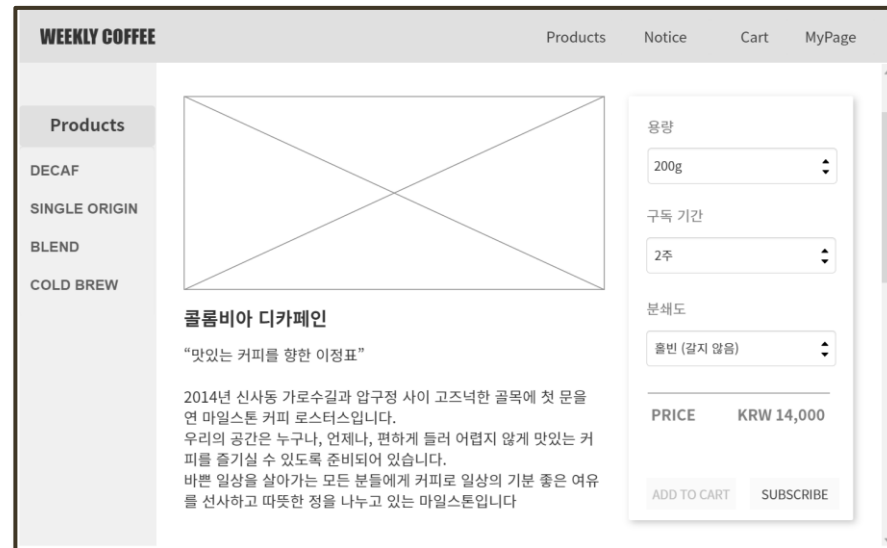
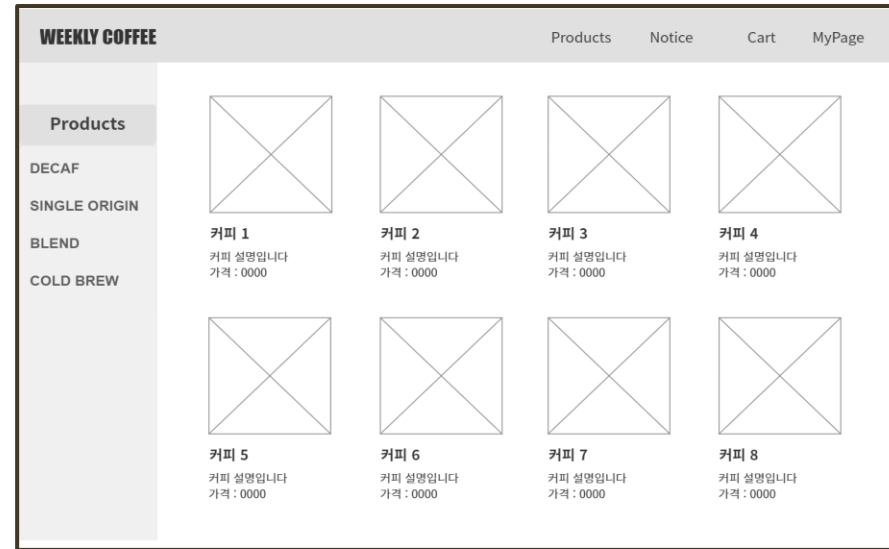
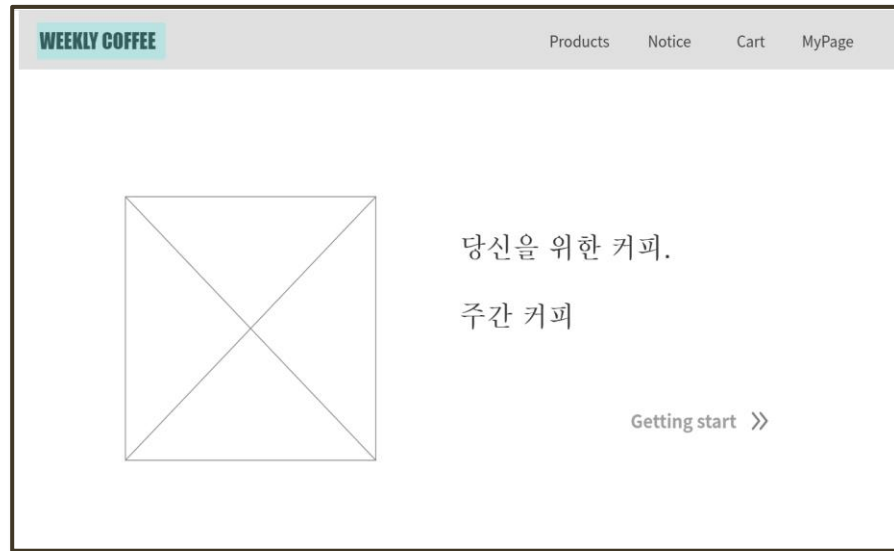


- 프로젝트에 필수적인 객체 요소를 Visual Paradigm을 사용하여 표현했습니다.

- 각 서비스 간에 AMQP를 통해 주고 받을 데이터를 설계하였습니다.



- 카카오 오븐을 활용하여 주간커피 UI 프로토타입을 제작했습니다. (사진을 클릭하면 카카오 오븐 링크로 연결됩니다.)



- 카카오 오븐을 활용하여 주간커피 UI 프로토타입을 제작했습니다. (사진을 클릭하면 카카오 오븐 링크로 연결됩니다.)

WEEKLY COFFEE SUBSCRIBE 배송지 정보

수령인

주소 우편번호

휴대전화

배송메모

☐ [콜롬비아 디카페인] 상글오리진, 200g, 2주, 홀빈 - 1 +

START SUBSCRIBE

WEEKLY COFFEE Cart

CART

선택삭제 전체삭제

<input checked="" type="checkbox"/>	<input type="checkbox"/>	[콜롬비아 디카페인] 상글오리진, 200g, 2주, 홀빈	KRW 14,000	- 1 +	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	[콜롬비아 디카페인] 상글오리진, 200g, 2주, 홀빈	KRW 14,000	- 1 +	<input type="checkbox"/>
<input checked="" type="checkbox"/>	<input type="checkbox"/>	[콜롬비아 디카페인] 상글오리진, 200g, 2주, 홀빈	KRW 14,000	- 1 +	<input type="checkbox"/>

RECOMMEND

Order total KRW 14,000

선택 상품 주문 전체 상품 주문

WEEKLY COFFEE MyPage

MyPage

주문일자	주문내역	수량	구매금액
2021-10-31	[콜롬비아 디카페인] 상글오리진, 200g, 2주, 홀빈	1	14,000 ₩

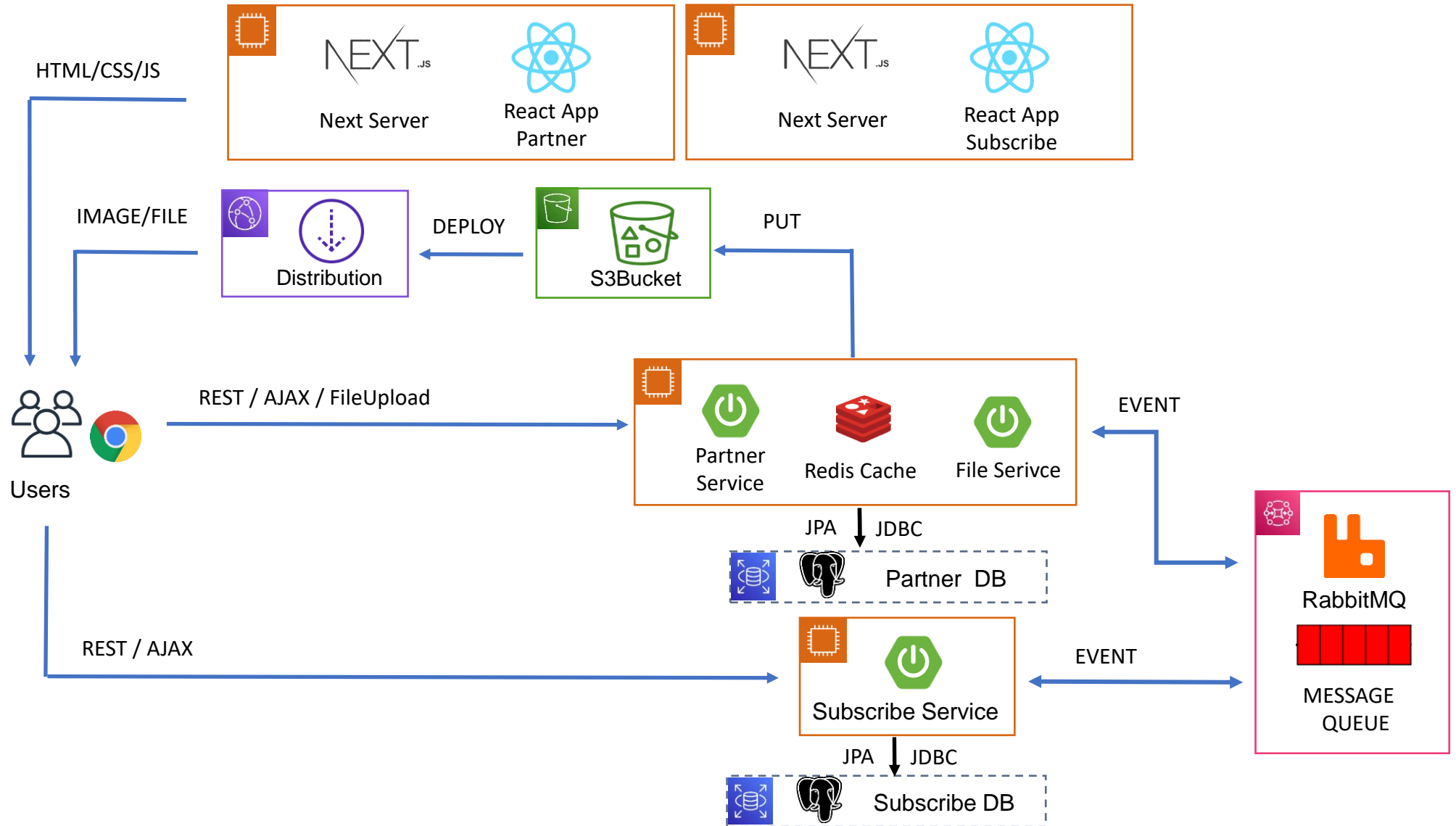
RECOMMEND

< . . . >

KAKAO OVEN

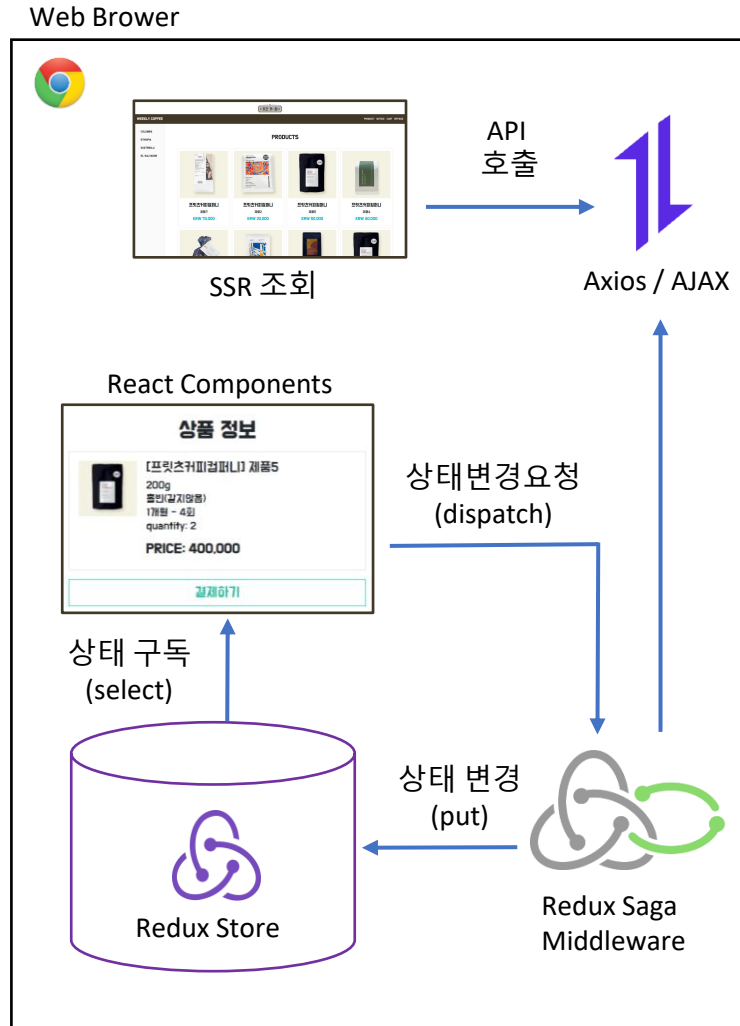


팀 아키텍처

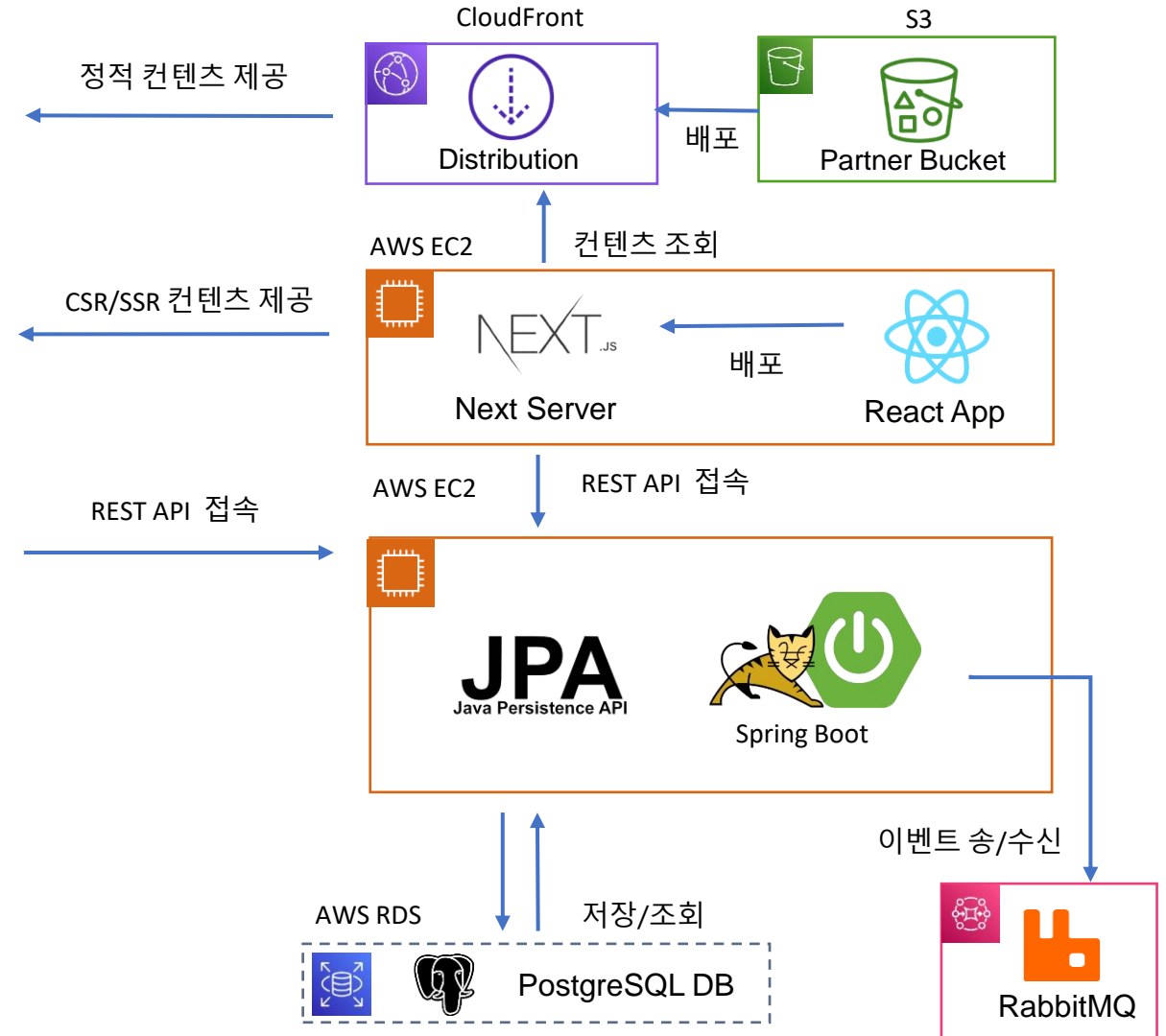


Subscribe 서비스 아키텍처

Subscribe Client



Subscribe Server



프로젝트 구현

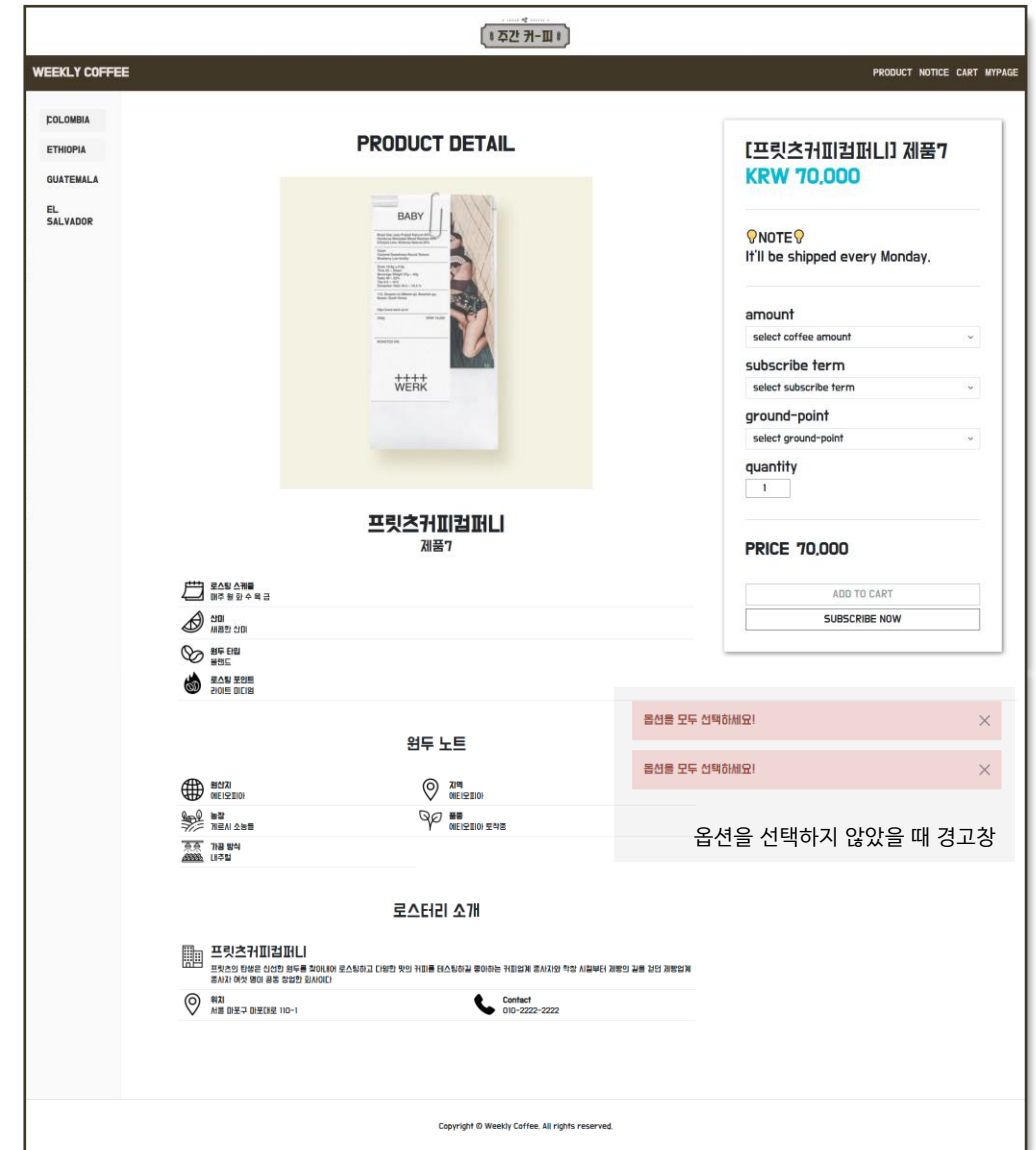
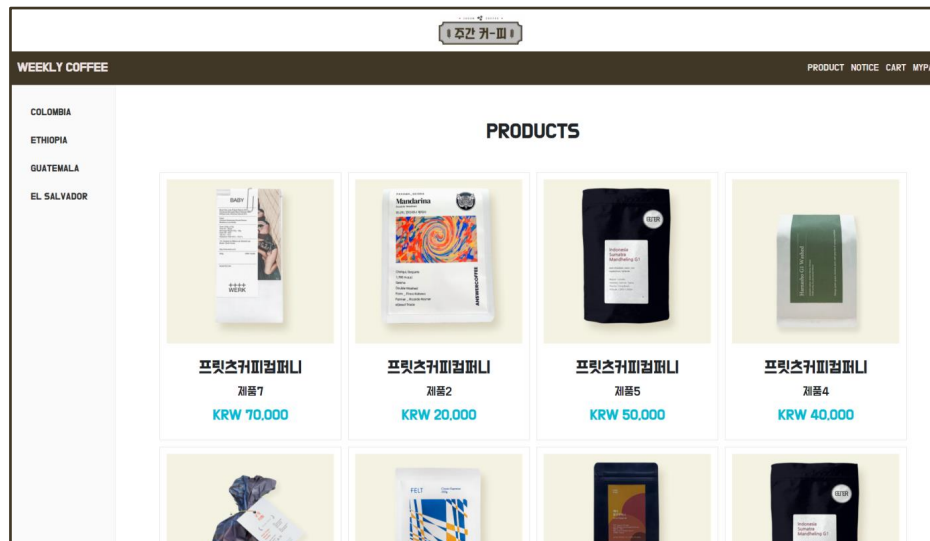
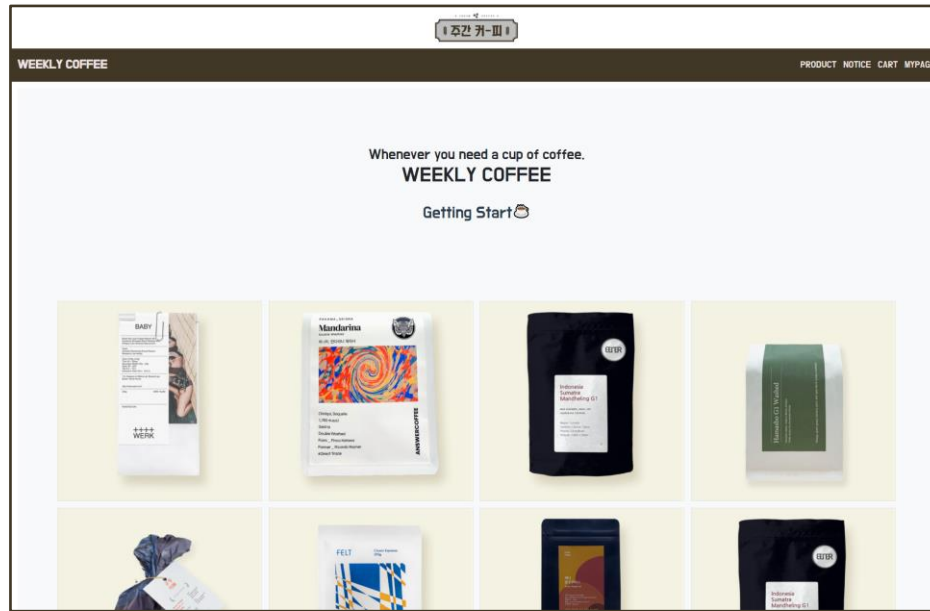
Subscribe Client

UI 구현

- UI 라이브러리를 사용하여 화면을 구현하였습니다.



UI style



UI 구현 • UI 라이브러리를 사용하여 화면을 구현하였습니다.



UI style

PRODUCT

배송지 정보

받으시는 분
이름

휴대전화
0000

주소
주소
상세주소

배송 메모
배송메모

결제 정보

주문금액
배송비

총 결제 금액

KRW 960,000
KRW 2,500
KRW 962,500

상품 정보

[프릿츠커피컴퍼니] 제품7
200g
출반(갈지않음)
17개월 ~ 4회
quantity: 1
PRICE: 560,000

[프릿츠커피컴퍼니] 제품5
200g
출반(갈지않음)
17개월 ~ 4회
quantity: 2
PRICE: 400,000

결제하기

클릭하면 주문 화면으로 이동하고
장바구니에 담긴 옵션을 우측에 보여줍니다.

SHOPPING CART

SELECT	ITEM	NAME	PRICE	QUANTITY	OPTION	REMOVE
<input checked="" type="checkbox"/>		[프릿츠커피컴퍼니] 제품5	400,000	2	200g 출반(갈지않음) 17개월 ~ 4회	×
<input checked="" type="checkbox"/>		[프릿츠커피컴퍼니] 제품7	560,000	1	400g 출반(갈지않음) 17개월 ~ 4회	×
전체삭제		Order Total	960,000			

선택 상품 주문

전체 상품 주문

RECOMMEND

click!

UI 구현

- UI 라이브러리를 사용하여 화면을 구현하였습니다.



MY PAGE					
주문일자	제품	제품정보	판매자	상품구매금액	주문처리상태
2021-12-02		제품2	프릿츠커피컴퍼니	80,000	15
2021-12-02		제품13. 외 2건	프릿츠커피컴퍼니	704,000	14
2021-12-02		제품5	프릿츠커피컴퍼니	400,000	13
2021-12-01		제품5	프릿츠커피컴퍼니	600,000	8
2021-12-1		제품5	프릿츠커피컴퍼니	600,000	8

RECOMMEND



주문 한 건을 클릭하면
주문 상세 내역으로 이동합니다.

한 주문에 제품이 여럿이면 **외 0건**으로 표시됩니다.

주문 상세 정보

주문번호	상품	상품정보	판매자	옵션	금액(수량)	진행상태
20211202014		제품13	프릿츠커피컴퍼니	600g 디지 1개량 ~ 4외	264,000 1개	접수완료
20211202014		제품3	프릿츠커피컴퍼니	200g 프릿츠프레스 1개량 ~ 4외	120,000 1개	접수완료
20211202014		제품4	프릿츠커피컴퍼니	400g 에스프레소 1개량 ~ 4외	320,000 1개	접수완료
				상품금액: 704,000	배송비: 2,500	합계: KRW 706,500

배송지 정보

주문일자	수령인	주소	연락처	배송명도
2021-12-02	JANE LEE	253, Cheongsu-ro205-1006	01067412771	부재 시 문 앞

목록

RECOMMEND



Next.js 공통 컴포넌트 사용

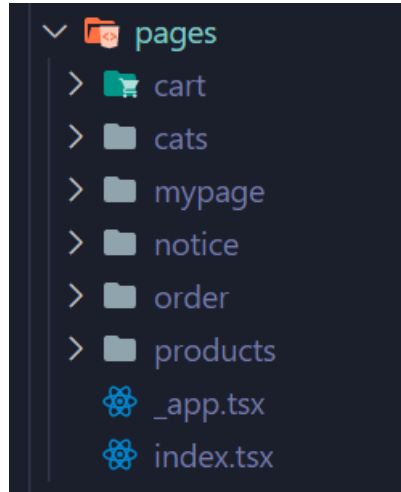
Next.js Layout



```
function MyApp({ Component, pageProps }: AppProps) {  
  return (  
    <Layout>  
      <Provider store={store}>  
        <Component {...pageProps} />  
      </Provider>  
    </Layout>  
  );  
}
```

- 최상위 컴포넌트인 app.tsx에 Layout 컴포넌트를 import 해서 공통 컴포넌트로 사용하였습니다.
- Layout의 각 부분을 컴포넌트로 만들고 import 해서 레이아웃의 틀을 구현하였습니다.
- pageProps에 따라 application 페이지가 교체 됩니다. (Single-Page-Application)

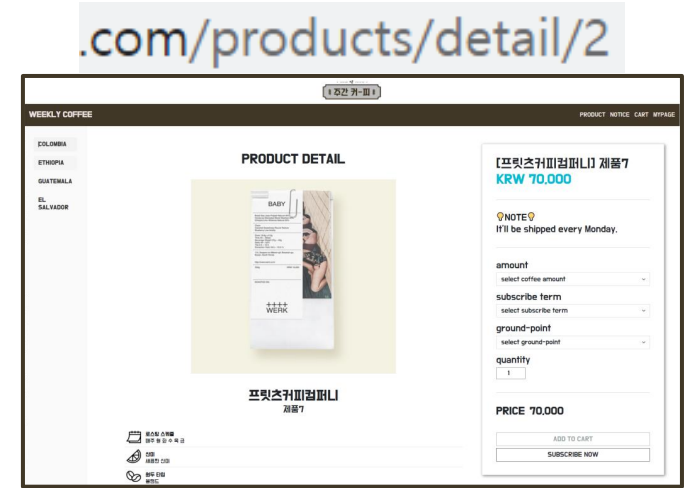
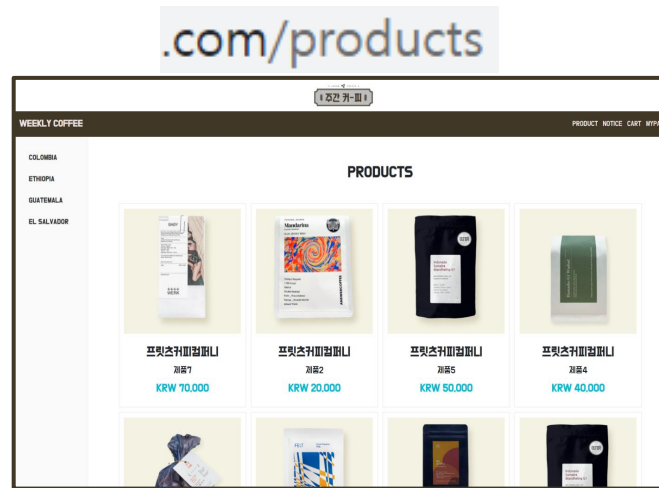
Next.js 프레임워크의 파일 기반 Routing 사용 (컴포넌트 이동)



Pages 폴더의 path 와 useRouter 로
컴포넌트 이동을 했습니다.

```
const router = useRouter();

onClick={() => {
  // id값을 묻고 이동해야함
  router.push(`/products/detail/${item.productId}`);
}}
```



각 product의 id 값을 가지고
해당 product detail로 이동합니다

React hooks 를 사용하여 함수형 컴포넌트를 구현했습니다.

- useRef 코드와 화면에서 Ref input값을 받아오는 부분입니다.

```
const amountInput = useRef<HTMLSelectElement>(null);
const substermInput = useRef<HTMLSelectElement>(null);
const groundpointInput = useRef<HTMLSelectElement>(null);
const quantityInput = useRef<HTMLInputElement>(null);
```

- useEffect 에 dependency를 설정하여 데이터가 추가되어야 화면이 전환되도록 했습니다.

```
useEffect(() => {
  isAddcomplete && router.push("/mypage");
}, [isAddcomplete, router]);
```

- useState에 초기값을 설정해준 코드입니다.

```
const [addTotal, setAddTotal] = useState(0);
const [amount, setAmount] = useState("");
const [term, setTerm] = useState("");
```

[프릿츠커피컴퍼니] 제품2
KRW 20,000

💡NOTE💡

It'll be shipped every Monday.

amount

select coffee amount

subscribe term

select subscribe term

ground-point

select ground-point

quantity

1

PRICE 20,000

ADD TO CART

SUBSCRIBE NOW

Redux-toolkit을 사용해 글로벌 state를 관리 하였습니다.

Subscribe Component (주문 생성)

배송지 정보

받는이는 본

JANE LEE

휴대전화

01067412771

주소

253, Cheongs-ro

205-1006

배송 방법

선택

결제 정보

주문금액

KRW 80,000

배송비

KRW 2,500

총 결제 금액

KRW 82,500

상품 정보

[프린트초커피컬러] 제품2

200g

다지

1개량 - 4단

quantity: 1

PRICE: 80,000

구매하기

click!

Mypage Component (주문 조회)

PRODUCT NOTICE CART MYPA

RECOMMEND

MY PAGE

주문일자	제품	제품정보	판매자	상품구매금액	주문처리상태
2021-12-02		제품2	프린트초커피컬러	80,000	15
2021-12-02		제품13, 외 2건	프린트초커피컬러	704,000	14
2021-12-02		제품5	프린트초커피컬러	400,000	13
2021-12-01		제품5	프린트초커피컬러	600,000	8
2021-12-1		제품5	프린트초커피컬러	600,000	8

Redux - toolkit
Redux - Saga

- State에 주문이 추가됨

WEEKLY COFFEE

State

Action

State

Diff

Trace

Test

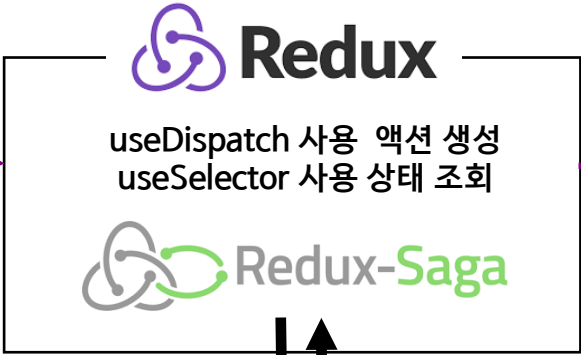
Tree

Chart

Raw

```

▶ product (pin): { data: [], isFetched: false }
▶ cartItem (pin): { data: [], isFetched: false, isAddCompleted: true
▼ subscribe (pin)
▶ data (pin): [{...}, {...}, {...}, {...}, ...]
isFetched (pin): true
totalPages (pin): 2
page (pin): 0
    
```



API 호출 응답



Subscribe Server

state change

Swagger를 사용하여 서버의 API를 시각화 했습니다.



SubscribeService^{1.0}

[Base URL : localhost:8080/]
<http://localhost:8080/v2/api-docs>

API List of SubscribeService Application

포트폴리오 프로젝트

product-controllerProduct Controller

GET

/products

getProducts

GET

/products/{id}

getProduct

GET

/products/country/{countryName}

getProductContainsCountry

subscribe-controllerSubscribe Controller

POST

/subscribes

requestSubscribe

GET

/subscribes/{id}

getSubscribe

GET

/subscribes/paging/{subscriberId}

getPagingSubscribes

프로젝트 구현

서비스 통합 구현

RabbitMQ를 활용해 서비스 간 데이터를 주고받도록 구현 했습니다.

판매 제품 메시지큐
Partner → Subscribe



프로젝트 구현

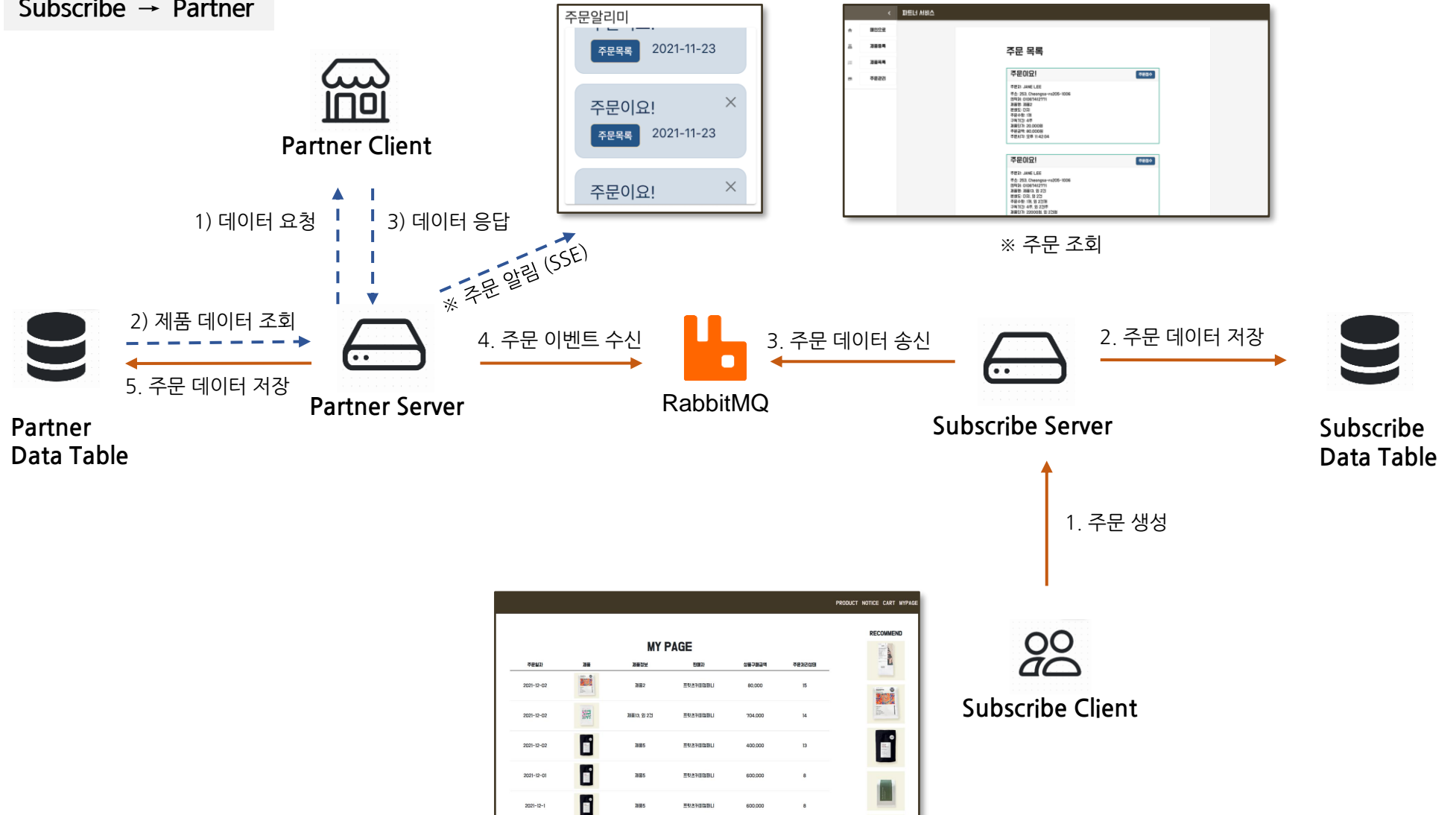
서비스 통합 구현

RabbitMQ를 활용해 서비스 간 데이터를 주고받도록 구현 했습니다.

주문 메시지큐

Subscribe → Partner

Subscribe에서 추가한 주문 데이터를 Partner 서버에서도 출력합니다.

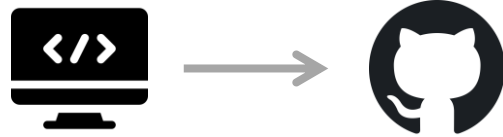




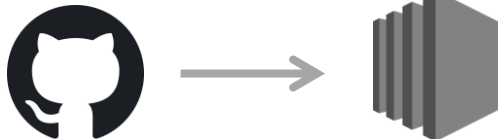
AWS EC2 인스턴스 서버에 node js 를 설치하여 배포 하였습니다.



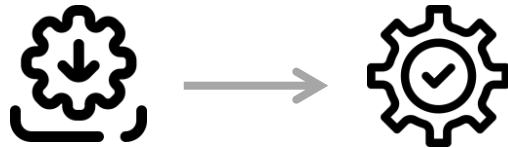
Front-end 배포



1. 개발 작업을 마치고 git commit 을 통해 소스코드 업데이트



2. EC2 인스턴스 서버에 git clone (git pull)을 통해 소스 최신화



3. 서버에서 npm install / npm run build / npm start

```
ubuntu@ip-172-31-2-11:~/weekly-coffee$ ls -al
total 2056
drwxrwxr-x 7 ubuntu ubuntu 4096 Nov 28 08:47 .
drwxr-xr-x 7 ubuntu ubuntu 4096 Nov 22 01:40 ..
-rw-rw-r-- 1 ubuntu ubuntu 82 Nov 22 01:30 .eslintrc.json
drwxrwxr-x 8 ubuntu ubuntu 4096 Nov 28 09:51 .git
-rw-rw-r-- 1 ubuntu ubuntu 414 Nov 22 01:30 .gitignore
-rw-rw-r-- 1 ubuntu ubuntu 12530 Nov 28 08:19 README.md
-rw-rw-r-- 1 ubuntu ubuntu 86 Nov 22 01:30 es2-myworkspace.bat
-rw-rw-r-- 1 ubuntu ubuntu 92 Nov 22 01:38 package-lock.json
drwxrwxr-x 12 ubuntu ubuntu 4096 Nov 24 14:46 subscribe
drwxrwxr-x 3 ubuntu ubuntu 4096 Nov 22 01:30 subscribe-server
drwxrwxr-x 3 ubuntu ubuntu 4096 Nov 24 14:46 참고 파일 reference files
-rw-rw-r-- 1 ubuntu ubuntu 2043281 Nov 28 08:47 포트폴리오_주간커피_이주는.pptx
drwxrwxr-x 3 ubuntu ubuntu 4096 Nov 22 01:30 프로젝트 설계 project design
ubuntu@ip-172-31-2-11:~/weekly-coffee$
```

```
up to date, audited 550 packages in 2s

89 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

> subscribe@0.1.0 build
> next build

info - Loaded env from /home/ubuntu/weekly-coffee/subscribe/.env.production
info - Checking validity of types...
```



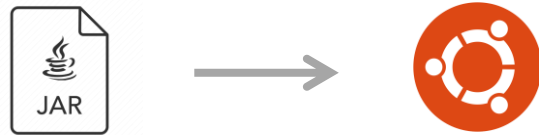
AWS EC2 인스턴스 서버에 node js 를 설치하여 배포 하였습니다.



Back-end 배포



1. 서버 경로에서 gradlew bootjar 입력, jar 파일 생성



2. Terminal에서 jar 파일 우분투 서버로 전송 (scp)

```
scp -i "path/path/filename.pem" -r filename.jar ubuntu@배포주소:서버저장경로
```

**NO
HangUp**

3. 백그라운드에 jar 파일 배포 → 백그라운드 서버 프로세스 유지

```
nohup java -jar [name-of-jar-file].jar &
```

- Java application 실행 여부 확인

```
ps -ef | grep java  
[pid]
```

- Java application 종료

```
kill -9 [pid]
```

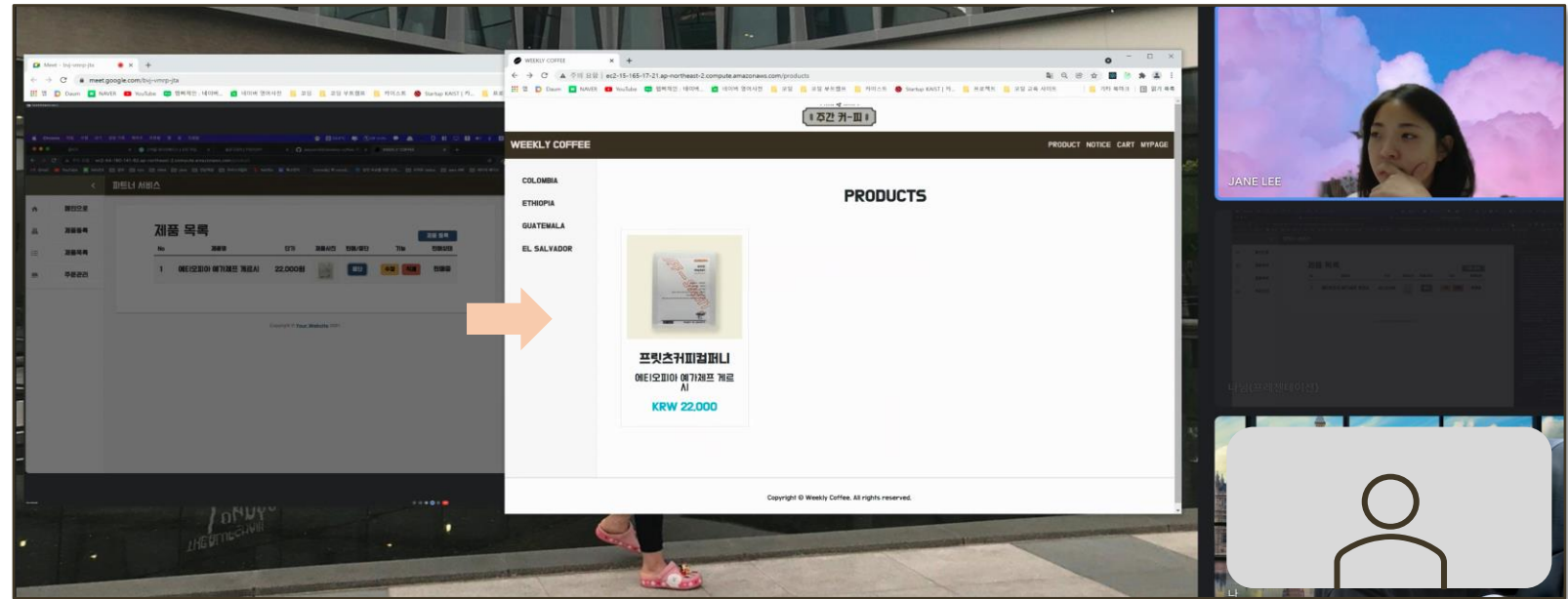
통합 운영 테스트

About Us

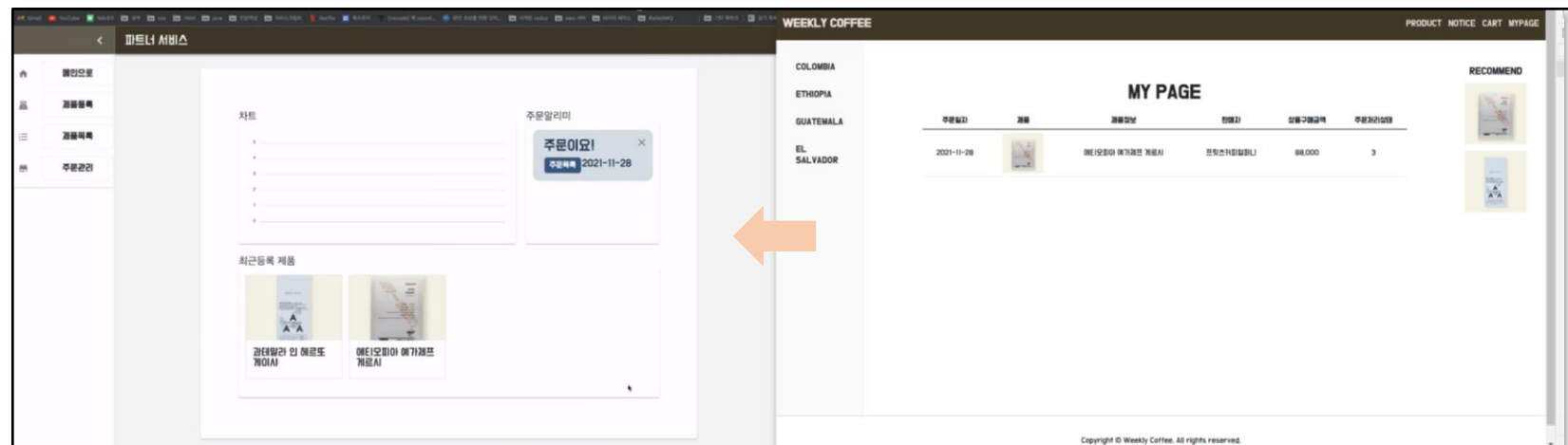


구글 미팅을 활용해 배포 서버에서 메시지 큐가 상대 서버에 잘 전달 되는지 테스트 하였습니다.

Partner → Subscribe



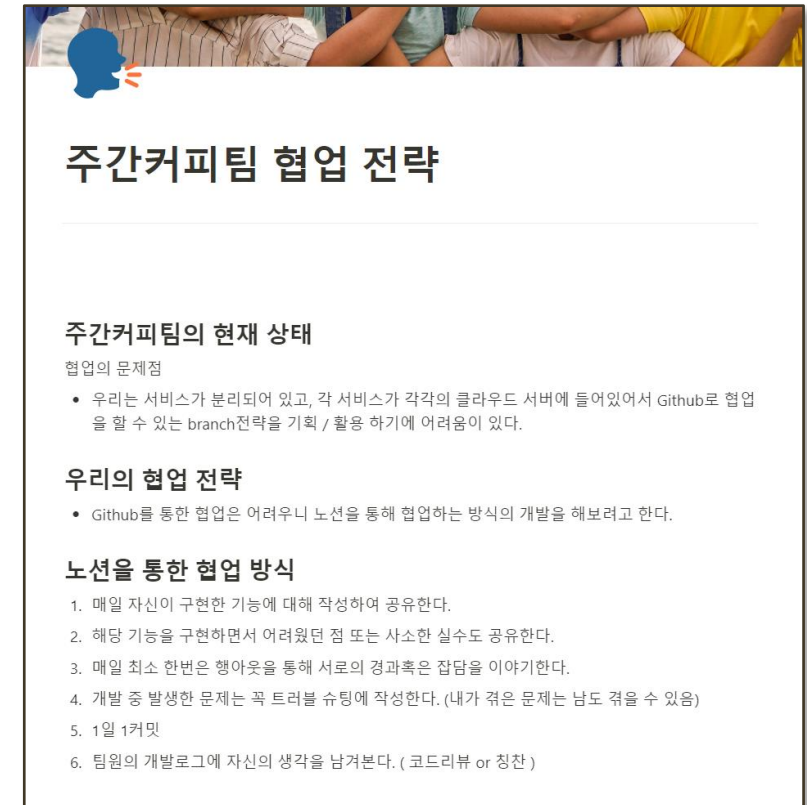
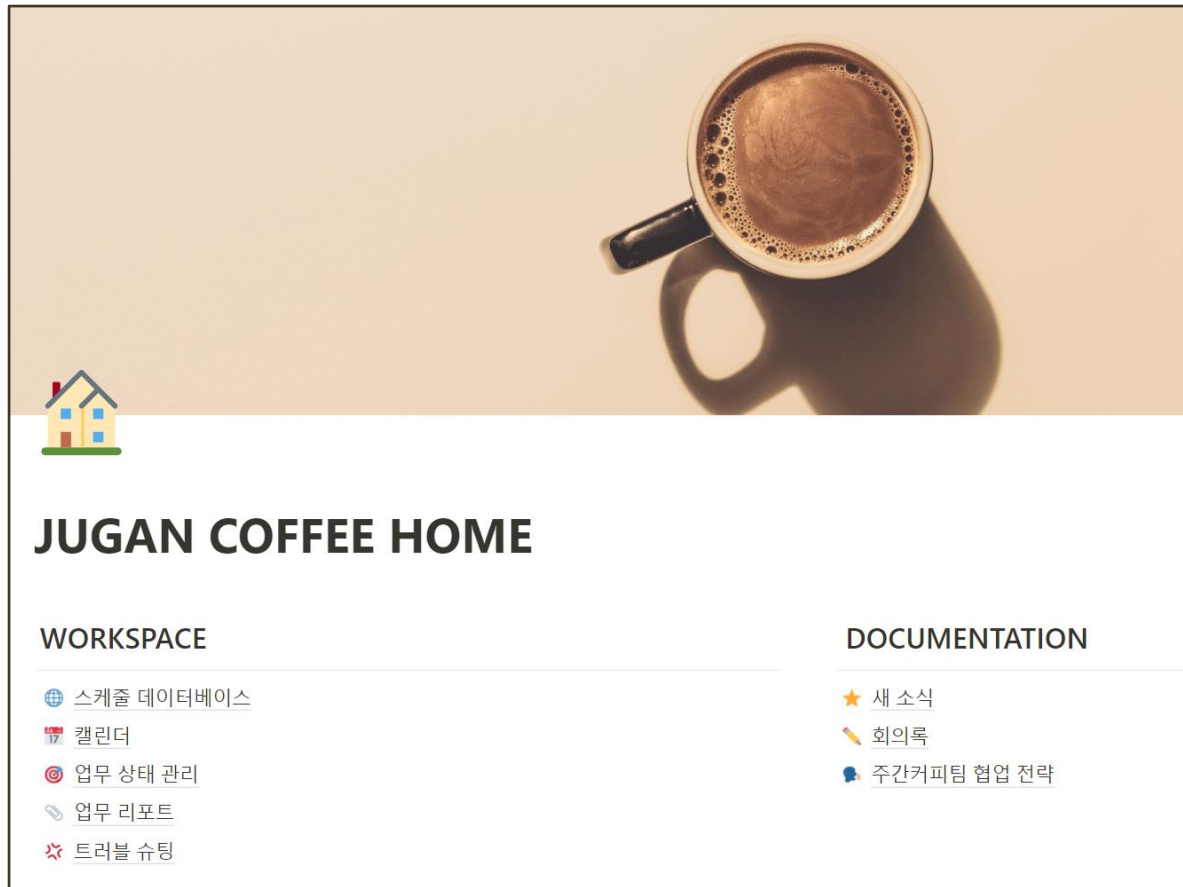
Subscribe → Partner





Notion 협업 툴 활용, 프로젝트 진척도 관리

- 구성원이 적극적으로 Notion을 활용할 수 있도록 홈페이지를 작성하고 사용법을 공유했습니다.



Notion 협업 툴 활용, 프로젝트 진척도 관리

- 개발 중 겪은 문제를 기록하여 서로의 어려움과 해결책을 공유했습니다.
- Notion을 활용한 결과, 효율적으로 프로젝트 진척도 관리를 할 수 있었습니다.

트러블 슈팅

💡 개발 중 겪은 문제를 공유하는 곳입니다.

📄 (양식)[이름] 트러블 슈팅 제목

📄 (양식)[이름] 트러블 슈팅 제목

2021-10-27

📄 [주은] git add . 에러 (LF will be replaced by CRLF)

2021-11-1

📄 [재윤] DB연동중 한 실수

📄 [재윤] PostgreSQL예약어 관련 에러

📄 [재윤] 중복 에러

📄 [재윤] 객체 관계 맵핑

2021-11-2

📄 [재윤] SQL Order키워드 예약어 관련 에러

📄 [주은] Next.js & 타입스크립트 스타일 적용



👤 담당자 별 업무

📅 담당자 별 업무입니다.

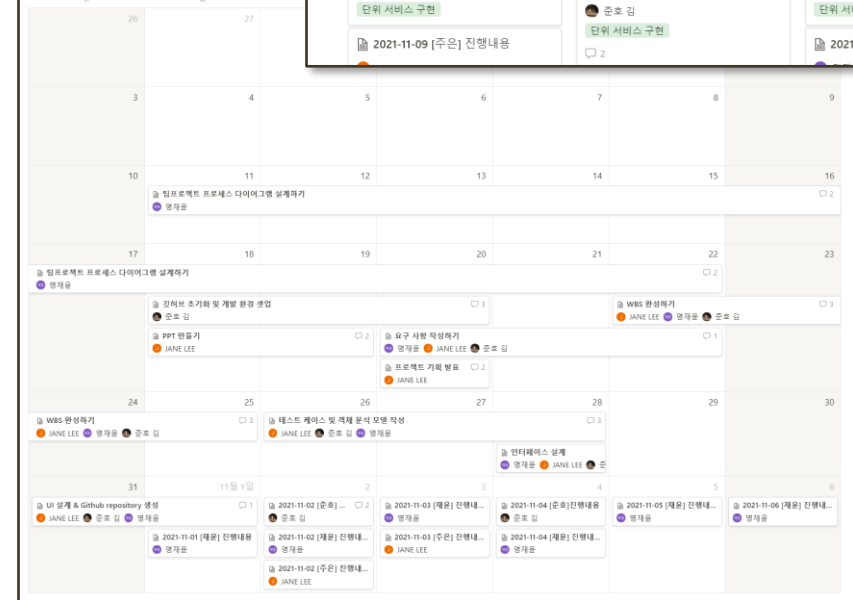
📊 스케줄 데이터베이스 + 보기 추가

JANE LEE 13	준호 김 8	명재윤 23
📄 요구 사항 작성하기 명재윤 JANE LEE 준호 김 분석 및 설계 1	📄 요구 사항 작성하기 명재윤 JANE LEE 준호 김 분석 및 설계 1	📄 요구 사항 작성하기 명재윤 JANE LEE 준호 김 분석 및 설계 1
📄 WBS 완성하기 JANE LEE 명재윤 준호 김 분석 및 설계 3	📄 WBS 완성하기 JANE LEE 명재윤 준호 김 분석 및 설계 3	📄 WBS 완성하기 JANE LEE 명재윤 준호 김 분석 및 설계 3
📄 2021-11-17 [주은] 진행내용 JANE LEE 단위 서비스 구현	📄 깃허브 초기화 및 개발 환경 셋업 준호 김 기획	📄 2021-11-22 [재윤] 진행내용 명재윤 단위 서비스 구현
📄 2021-11-16 [주은] 진행내용 JANE LEE 단위 서비스 구현	📄 2021-11-04 [준호] 진행내용 준호 김 단위 서비스 구현	📄 2021-11-21 [재윤] 진행내용 명재윤 단위 서비스 구현
📄 2021-11-15 [주은] 진행내용 JANE LEE 단위 서비스 구현	📄 2021-11-02 [준호] 진행내용 준호 김 단위 서비스 구현	📄 2021-11-20 [재윤] 진행내용 명재윤 단위 서비스 구현
📄 2021-11-09 [주은] 진행내용	📄 2021-11-02 [준호] 진행내용 준호 김 단위 서비스 구현	📄 2021-11-19 [재윤] 진행내용

월간 업무

📊 스케줄 데이터베이스 + 보기 추가

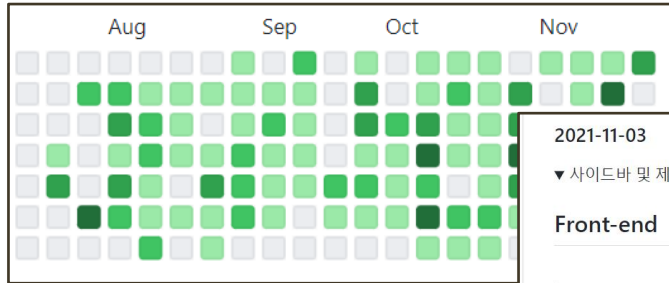
2021년 10월





Github repository의 Readme 파일 활용

- 마크다운 문법으로 readme 파일을 작성하여 프로젝트 진행 과정을 기록했습니다.



2021-11-03

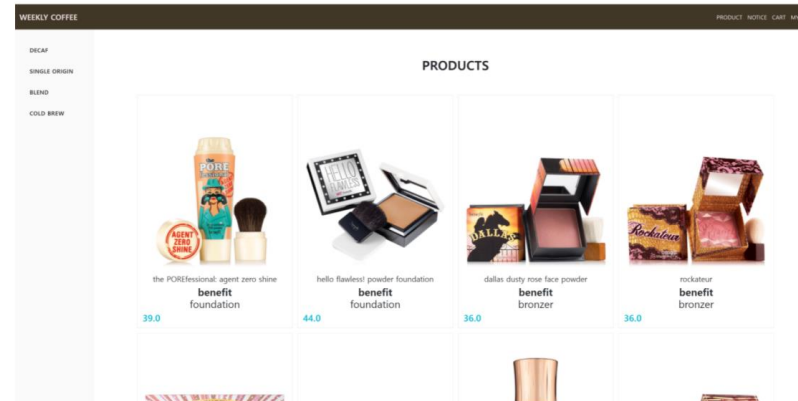
▼ 사이드바 및 제품목록 그리드 구현 [👉click]

Front-end

- Github commit
<https://github.com/happyjane210/weekly-coffee/commit/1912320b8a43f501b10bc21c8cd2af977acd5abb>
- Develop Issue
<https://www.notion.so/2021-11-03-d56fcd92d142f9b61657e95a4ce045>

진행 내용

- 원편 사이드바를 구현했습니다.
- 각 컴포넌트에 import 했던 상단 nav바를 app.tsx 한 곳에 추가해 전역으로 적용되도록 변경하였습니다.
- 상품목록 (products)를 구현했습니다.
 - 그리드 구현을 위해 화장품 오픈 API를 사용해서 제품 목록을 구현했습니다.
 - Next.js 에서 SSR(ServerSideRendering) 방식을 통해 오픈 API 데이터를 화면으로 가져오는 방법을 익혔습니다



참고 자료

- <http://makeup-api.herokuapp.com/api/v1/products.json?brand=dior>

☰ README.md

☕ weekly-coffee 주간커피 ☕

여러 로스터리 사업장과 소비자를 연결하는,
원두 정기 구독 서비스를 제공합니다.

💡 담당 개발 서비스 - 구독 서비스 (client) 👤

다양한 원두를 소비자가 원하는 옵션에 맞춰 정기 구독 할 수 있는 구독 서비스를 개발 합니다

- 판매 제품 등록
- 주문 관리
- 마이페이지, 공지사항, 장바구니

🔄 개발 진행 기록 🕒

2021-11-01

▶ 개발 환경 구성 [👉click]

2021-11-02

▶ 홈페이지 구현 [👉click]

2021-11-03

▶ 사이드바 및 제품목록 그리드 구현 [👉click]



PROJECT LINK BIO



<https://github.com/happyjane210/weekly-coffee>



<https://www.notion.so/JUGAN-COFFEE-HOME-11c388513ab74fd4bda5e6478c22abc8>

KAKAO
OVEN

<https://ovenapp.io/view/nEe8JaGECVCmiyLLlkkmkgamjo0M14R/>



<http://ec2-15-165-17-21.ap-northeast-2.compute.amazonaws.com/>

★ JUGAN  COFFEE ★

