

likelions-backend-8th 주.D.부 🙏
(주문은 DM으로 부탁드립니다 🙏)

Custom Master

목차



프로젝트 개요

개발 동기, 기획의도



프로젝트 상세

구조도, ERD, 기술설명



프로젝트 시연

시연 영상, 화면



진행 프로세스

개발일정, 개발 환경



프로젝트 개요

팀원 소개 / 개발 동기 / 기획 의도

팀원 소개

양동화(팀장)



- 인증
- 로그인 기능
- 매장 페이지

김슬찬



- 상점 관리
- 상품 관리
- 주문하기

박지수



- 주문 관리
- 리뷰
- CI/CD
- PPT 제작

장수빈



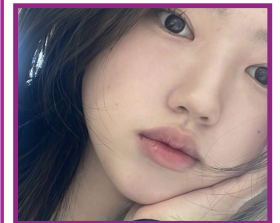
결제

김세인



리뷰

오시은



리뷰

개발 동기

커스텀 제품 구매 시 SNS의 DM(Direct Message)을 이용

이 방식에 친숙하지 않은 사람은 주문하기 어려움

보다 쉬운 주문으로 많은 사람들이 커스텀 제품에 입문, 소비

커스텀 제품의 진입 장벽 낮춤

기획 의도

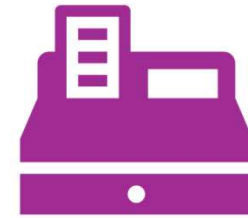
커스텀 제품을 예약하는 플랫폼 사이트



소비자

특정 매장에 Custom Order를 요청

제품을 구매한 소비자는 구매한 매장에 대한
리뷰 등록이 가능



매장

판매하는 상품을 등록 가능

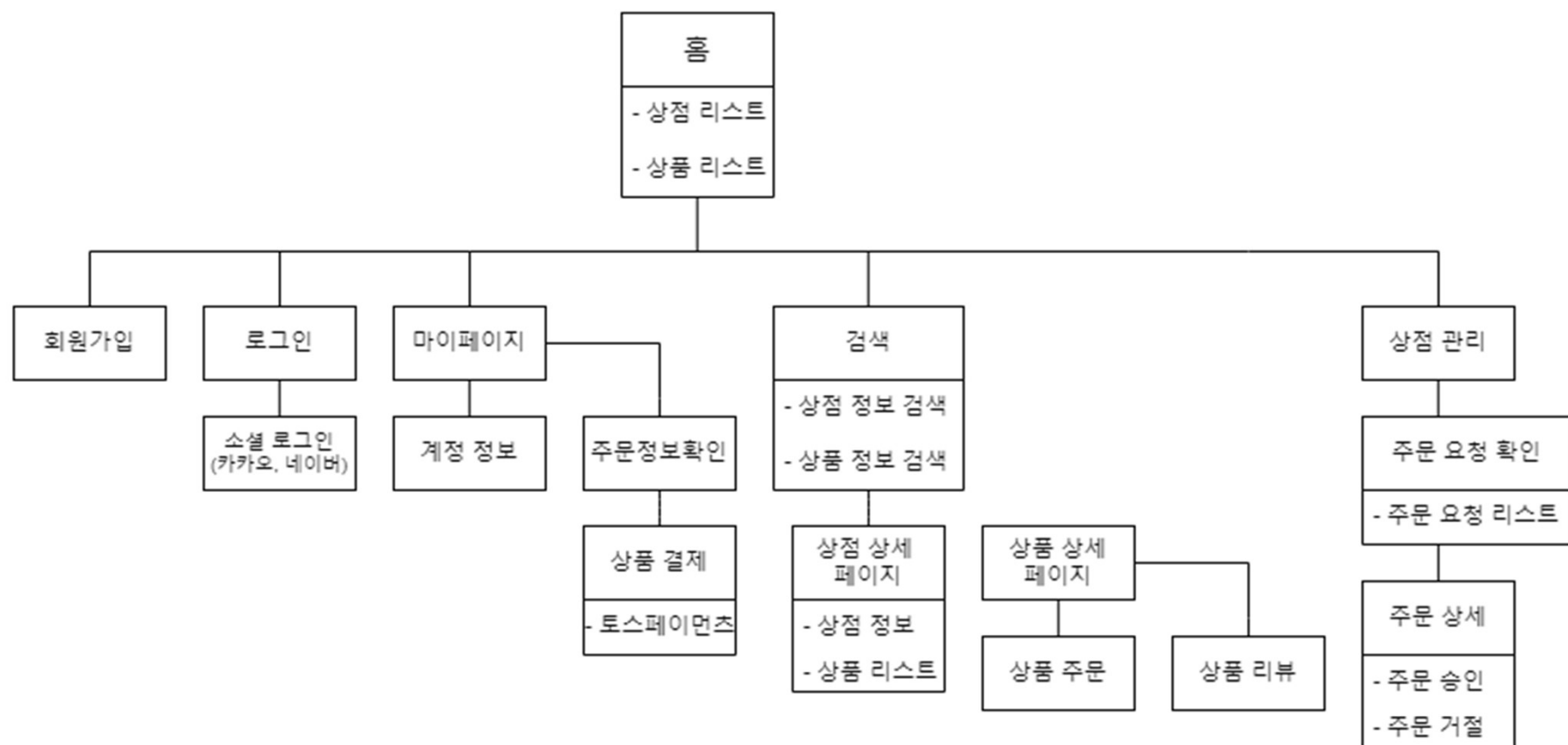
요청된 주문을 확인, 적정 금액을 설정해
승인 또는 거절이 가능



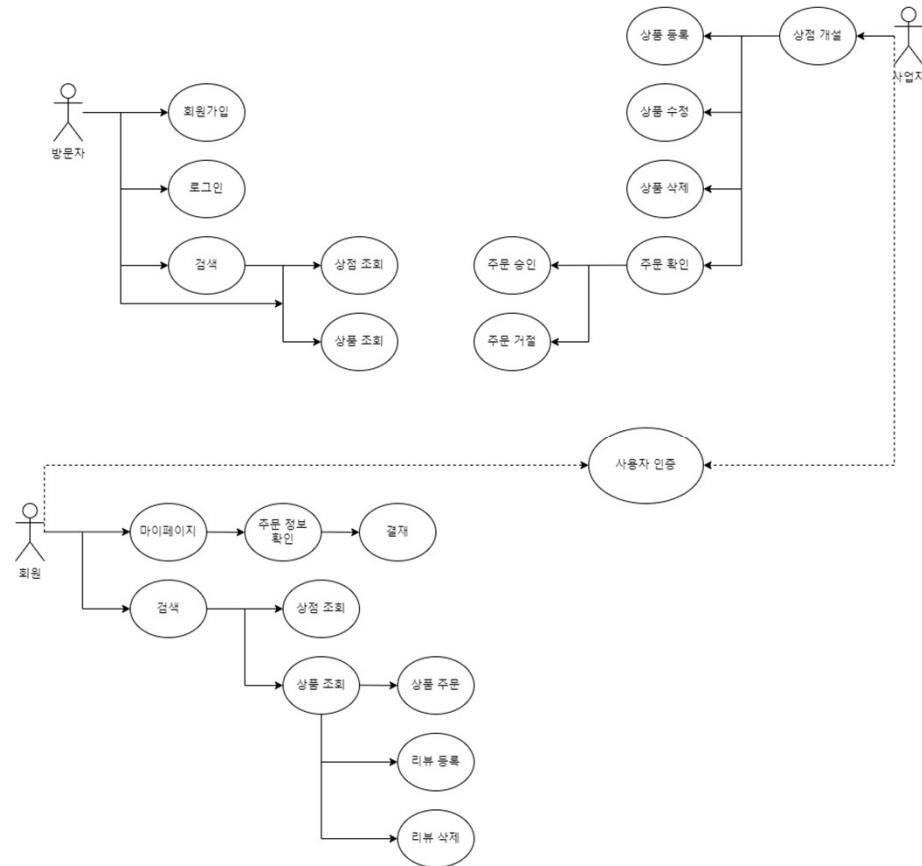
프로젝트 상세

구조도 / ERD / 기술설명

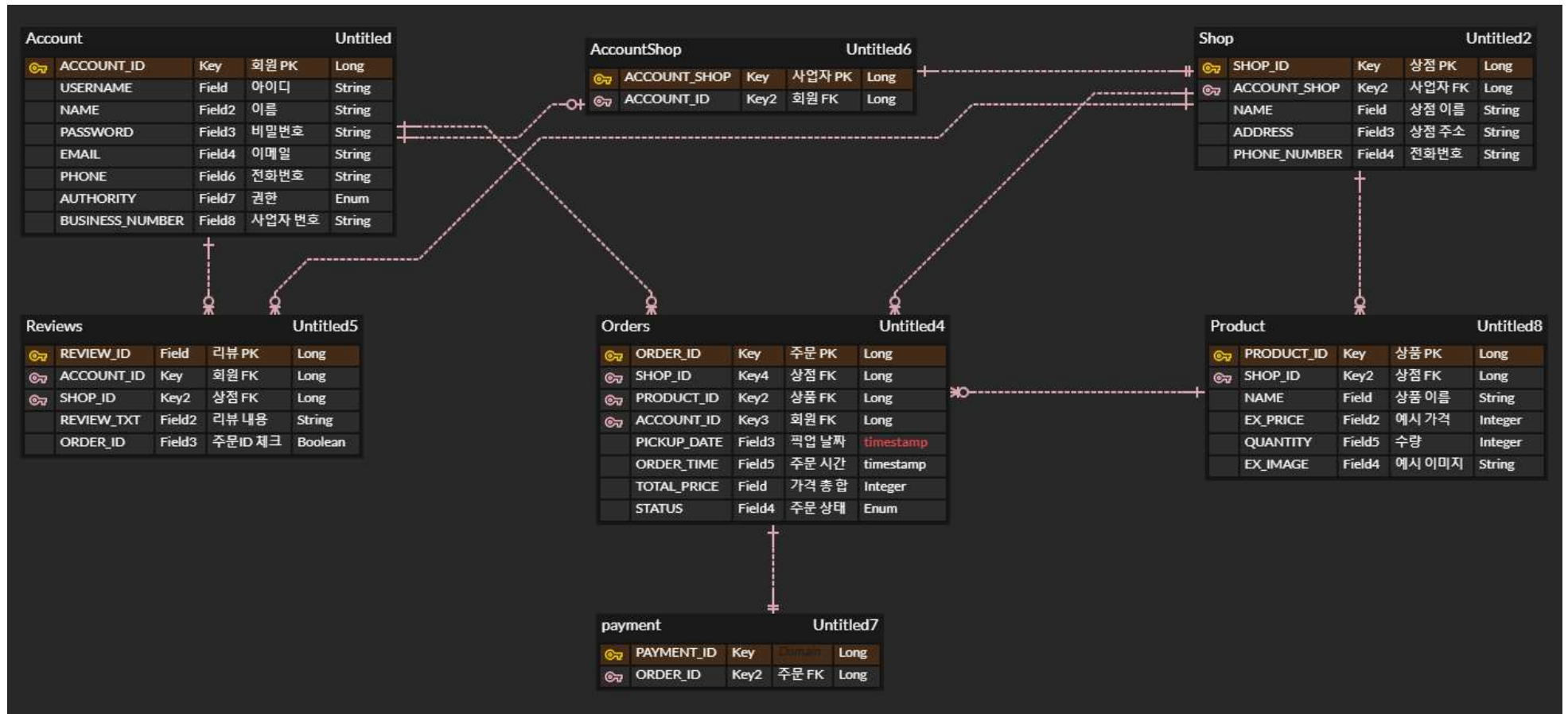
구조도



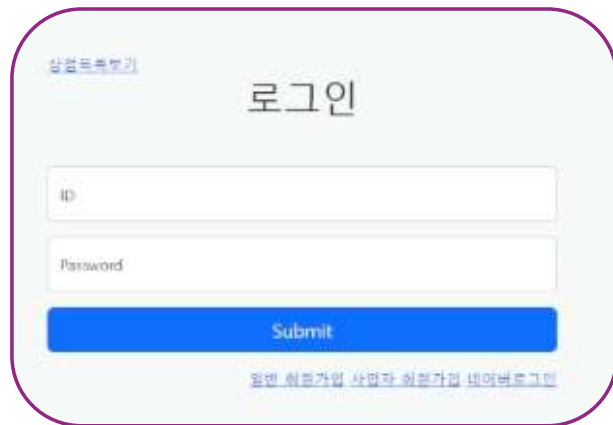
Usecase Diagram



ERD



기술 설명 - 인증하기



A login form titled '로그인' (Login). It features a header with a link '상업프로젝트' and a sub-header '로그인'. Below the header are two input fields: 'ID' and 'Password'. A blue 'Submit' button is positioned below the password field. At the bottom, there is a link '회원가입' (Sign Up) and a note '회원가입 시 비밀번호를 꼭 입력해주세요' (Please enter a password when signing up).

로그인



An OAuth login form. It includes a warning message at the top: '공통 PC에서 사용하지는 경우 보안을 위해 서비스 이용 후, 네이비게이션도 반드시 로그아웃해 주세요.' (Please log out after using the service for security if you are not using a common PC). The form has input fields for '아이디' (ID) and '비밀번호' (Password). There is a checkbox for '로그인 상태 유지' (Keep login state) and a toggle for '자동로그인' (Auto login). A large green '로그인' (Login) button is in the center. At the bottom, there are links for '비밀번호 찾기' (Find password), '아이디 찾기' (Find ID), and '회원가입' (Sign up).

OAuth



An email authentication form. It shows a notification at the top: 'localhost:8080 내용: 메일보내기 성공' (localhost:8080 content: Email sending success) with a '확인' (Check) button. Below this is a '로그인' (Login) section with a blue '메일보내기' (Send email) button. Underneath is a 'Code' input field and a blue '코드 제출하기' (Submit code) button. At the bottom right, there is a link '프로젝트 가기' (Go to project).

메일인증

Log in

Token 발급

- 로그인 시 두 종류의 토큰 발급
- Access token 은 로컬 저장소에, Refresh token은 쿠키에 저장

```
JwtResponseDto accessTokenDto = tokenService.issueAccess(userDetails);  
JwtResponseDto refreshTokenDto = tokenService.issueRefresh(userDetails, accessTokenDto.getAccessToken());  
Cookie cookie = new Cookie( name: "CMTOKEN", refreshTokenDto.getRefreshToken());  
cookie.setMaxAge(24 * 60 * 60 * 2);  
cookie.setPath("/"); // 쿠키의 경로 설정  
cookie.setDomain("localhost");  
cookie.setSecure(false);  
cookie.setHttpOnly(true);  
response.addCookie(cookie);  
return ResponseEntity.ok().body(refreshTokenDto);
```

OAuth

- 최초 로그인 시 회원 생성 후 로그인
- 로그인 시 일반 로그인과 마찬가지로 token 발급

NAVER

이메일 로그인 | 비밀번호 로그인 | QR코드

네이버에 로그인하여 커스텀마스터 서비스를
이용하실 수 있습니다.

공용 PC에서 사용하시는 경우 보안을 위해 서비스 이용 후
네이버에서도 반드시 로그아웃해 주세요.

이메일

비밀번호

☐ 로그인 상태 유지 ☒ 비밀번호 찾기

로그인

비밀번호 찾기 | 아이디 찾기 | 회원가입

프로필 관리

내 프로필

- 프로필 페이지로 이동 가능하다.
- 프로필 페이지에서 간단한 프로필을 확인할 수 있다.

메일인증

- 내 프로필에서 메일인증이 가능하다.
- 메일 인증을 통해 유저 권한을 올릴 수 있다.



기술 설명 - 주문 요청하기

Wilkinson Group

기본 정보

ID: 5

이름: Wilkinson Group

주소: 30588 Cordelia Parkway

전화번호: 409-426-4821

상품 추가하기

상품 목록

ID: 43

이름: [simvastatin](#)

가격: 63466

주문 삭제 주문하기

상점 상세에서 주문하기

simvastatin 주문하기

이름:

전화번호:

픽업 날짜와 시간:

요구 사항:

예시 이미지: speaker.png

주문하기

주문 요청

- 품 작성

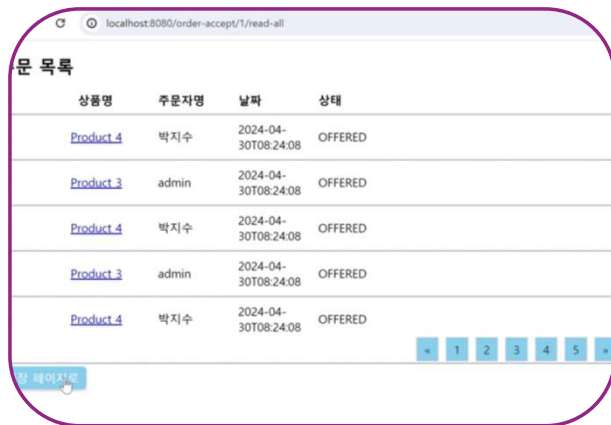
귀님의 주문 목록입니다.

상품명	매장명	날짜	상
Povidone-iodine	Mraz Group	2024-05-01T15:02:09.133	C
simvastatin	Wilkinson Group	2024-05-01T14:55:58.081	C

주문 확인

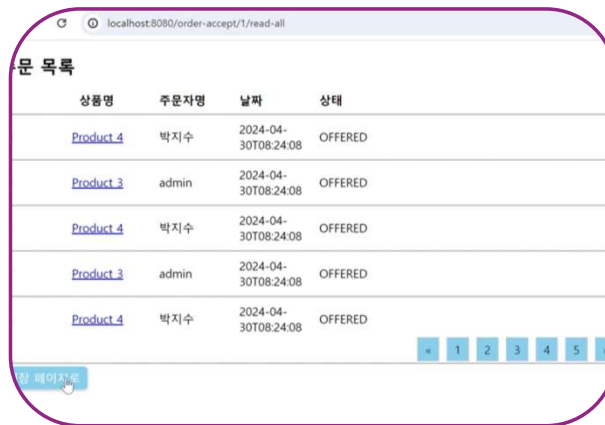
- 주문 목록에서 확인

기술 설명 - 주문 관리하기



상품명	주문자명	날짜	상태
Product 4	박지수	2024-04-30T08:24:08	OFFERED
Product 3	admin	2024-04-30T08:24:08	OFFERED
Product 4	박지수	2024-04-30T08:24:08	OFFERED
Product 3	admin	2024-04-30T08:24:08	OFFERED
Product 4	박지수	2024-04-30T08:24:08	OFFERED
Product 3	admin	2024-04-30T08:24:08	OFFERED

주문 목록 확인



상품명	주문자명	날짜	상태
Product 4	박지수	2024-04-30T08:24:08	OFFERED
Product 3	admin	2024-04-30T08:24:08	OFFERED
Product 4	박지수	2024-04-30T08:24:08	OFFERED
Product 3	admin	2024-04-30T08:24:08	OFFERED
Product 4	박지수	2024-04-30T08:24:08	OFFERED
Product 3	admin	2024-04-30T08:24:08	OFFERED

Pagination 적용



주문 상세

제품명: Product 4 단가: 1500

주문자: 박지수

주문 시각: 2024-04-30T08:24:08 픽업 날짜: 2024-04-30T08:24:08

주문 상태: OFFERED [주문 거절](#)

확정 금액을 입력하세요.

요청 사항

요청 사항이 없습니다.

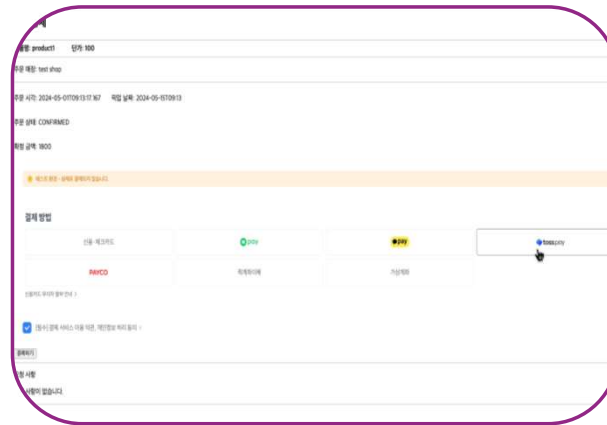
주문 승낙 / 거절

기술 설명 - 결제

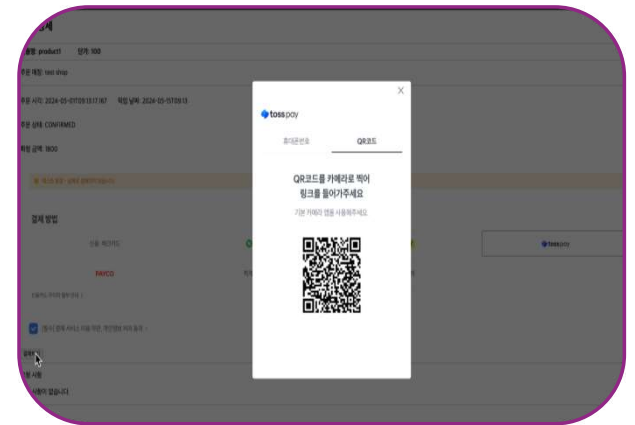
ang님의 주문 목록입니다.

ID	상품명	매장명	날짜	상태
3	product1	test shop	2024-05-01T09:13:17.167	CONFIRMED
1800				
	결제하기			
2	product1	test shop	2024-05-01T10:26:13.787	PAID
1	product1	test shop	2024-05-01T10:22:57.857	PAID

CONFIRMED만 결제

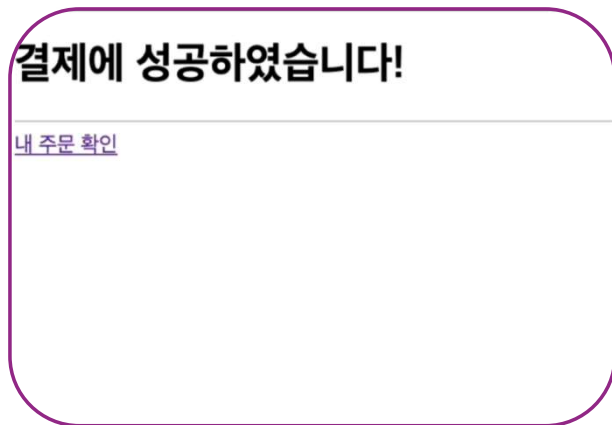


상세 주문에서 결제

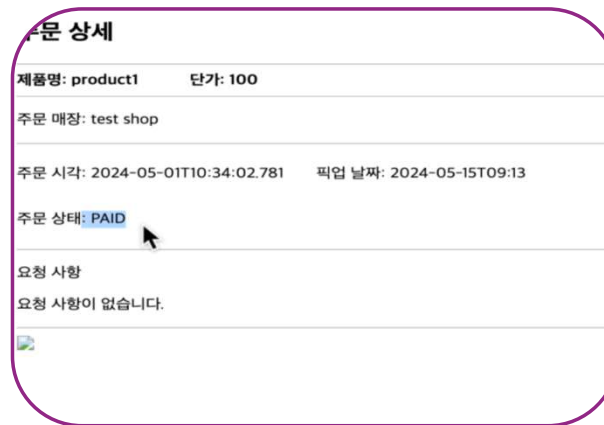


결제 진행

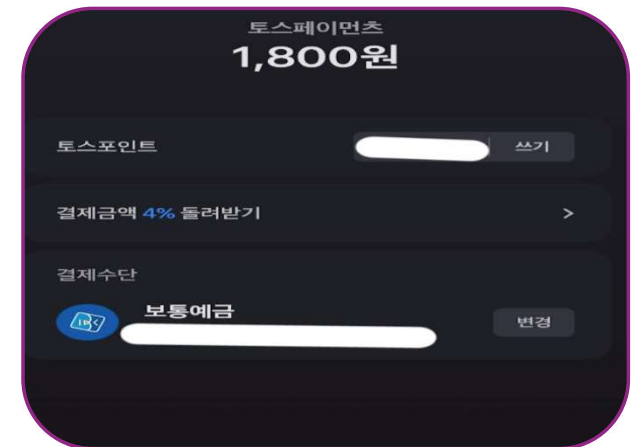
기술 설명 - 결제



결제 성공



주문 상태 변경
• CONFIRMED -> PAID



결제 내역

결제하기 – TROUBLE SHOOTING

인증되지 않은 키(Client, Secret)

- Toss 내부 이슈 문제
- 계약전에는 docs 공개키로만 결제 위젯이 연동되도록 변경되었다.
- 결제 페이지의 clientKey와 application.yaml의 secretKey 변경



위젯 불러오기

- 기존의 variantKey를 'AGREEMENT'를 'DEFAULT'로 변경

```
const paymentMethodsWidget = paymentWidget.renderPaymentMethods('#payment-method', {
  value: [`${ord.totalPrice}`], // 확정 금액을 value로 사용
  currency: 'KRW',
  country: 'KR'
}, { variantKey: 'DEFAULT' });
const paymentAgreement = paymentWidget.renderAgreement('#agreement', { variantKey: 'DEFAULT' });
const paymentRequestButton = document.getElementById('payment-request-button');
paymentRequestButton.addEventListener('click', () => {
  try {
```

기술 설명 - 리뷰 CRUD

테스트56
44

2
테스트56
22

1
테스트56
1

리뷰 전체 리스트

등록하세요.

로그인한 작성자 이름으로 리뷰가 등록됩니다.

리뷰 내용 입력하세요

리뷰 작성

리뷰 등록하기

리뷰 상세보기

작성자: 테스트56

수정하기

리뷰 삭제

33

Back

리뷰 상세보기

CI/CD - EC2 이용해 Docker Image 생성

Dockerfile 설정

- 배포는 dev 브랜치에서 테스트
- Jar 파일을 실행할 수 있도록 Dockerfile을 작성한다.

EC2 설정

- Dockerfile이 추가되어 있는 브랜치를 clone한다.
- Docker 이미지를 생성한다.
- Docker가 설치되어 있어야 한다.

Docker Image 실행

- 해당 컴퓨터에 설치된 이미지를 확인 가능하다.
- Docker Image를 실행한다.

```
# Gradle 빌드를 수행하고, JAR 파일을 적절한 위치로 이동합니다.
```

```
RUN ./gradlew bootJar && mv build/libs/*.jar app.jar
```

새로운 스테이지

```
✓ FROM eclipse-temurin:17-jre
```

```
WORKDIR /app
```

```
COPY --from=build /app/app.jar .
```

```
CMD ["java", "-jar", "app.jar"]
```

EXPOSE 8080

- docker 이미지 생성 후 실행

```
docker build -t <소문자 이름>
```

[illegible]

- 도커 이미지를 실행한다.

```
Bash
docker run --rm -p 8080:8080 <이미지 이름>
```

```

ubuntu@ip-10-0-1-10:~/Custom-Master$ sudo docker run --rm -p 8080:8080 custom-master
Standard Commons Logging discovery in action with spring-jcl: please remove commons-logging.jar from classpath in order
to avoid potential conflicts

  ____ _
 / ___ \| | | |
/ /___ \| |_| |
 \___ \|____|_|_|
  | | | |
  | |_| |
  |_____|

:: Spring Boot ::
               (v3.2.4)

2024-04-29T02:32:11.948Z INFO 1 -- [ main ] o.s.d.w.embedded.TomcatWebServer : Tomcat started on port
8080 (http with context path '/')
2024-04-29T02:32:11.973Z INFO 1 -- [ main ] j.custom-master.CustomMasterApplication : Started CustomMasterAp
plication in 28.481 seconds, forcesset running for 22.806s

```

CI/CD – TROUBLE SHOOTING

Docker 문제일 경우

- 도커는 컨테이너 내부에서 설정이 필요하다.
- docker 실행 시 -p 태그를 사용했는지 확인이 필요하다.
- 앞서 -p 를 사용해서 해당 문제는 아닌 것 같다.

ec2 문제일 경우

- ec2 보안 그룹을 설정하지 않으면 오류가 발생한다.
- 인바운드 규칙 설정이 필요하다.
- 8080포트에 대한 사용자 지정 TCP 인바운드 규칙을 추가했다.

The image shows two screenshots from the AWS IAM console. The top screenshot displays the 'Inbound Rules' configuration for a security group. It lists three rules: 'sgr-0488f51d3ff2d6e2' (User-defined TCP, port 8080), 'sgr-08aaa860fa3fad1f1' (SSH, port 22), and 'sgr-08ed28b19d342e50f' (User-defined TCP, port 8080). Each rule has a 'Source' field with a search icon and a '0.0.0.0/0' restriction. A 'Rules Added' button is at the bottom. The bottom screenshot shows a '로그인' (Login) page with fields for 'ID' and 'Password', a 'Submit' button, and links for '일반 회원가입' and '사업자 회원가입'.

보안 그룹 규칙 ID	유형	프로토콜	포트 범위	소스	설명 - 선택 사항	삭제
sgr-0488f51d3ff2d6e2	사용자 지정 TCP	TCP	8080	사용...	0.0.0.0/0 X	
sgr-08aaa860fa3fad1f1	SSH	TCP	22	사용...	0.0.0.0/0 X	
sgr-08ed28b19d342e50f	사용자 지정 TCP	TCP	8080	사용...	0.0.0.0/0 X	

규칙 추가

7.239:8080/account/login

로그인

ID

Password

Submit

[일반 회원가입](#) [사업자 회원가입](#)

CI/CD – GitHub Actions 사용

GitHub Action Workflow 추가

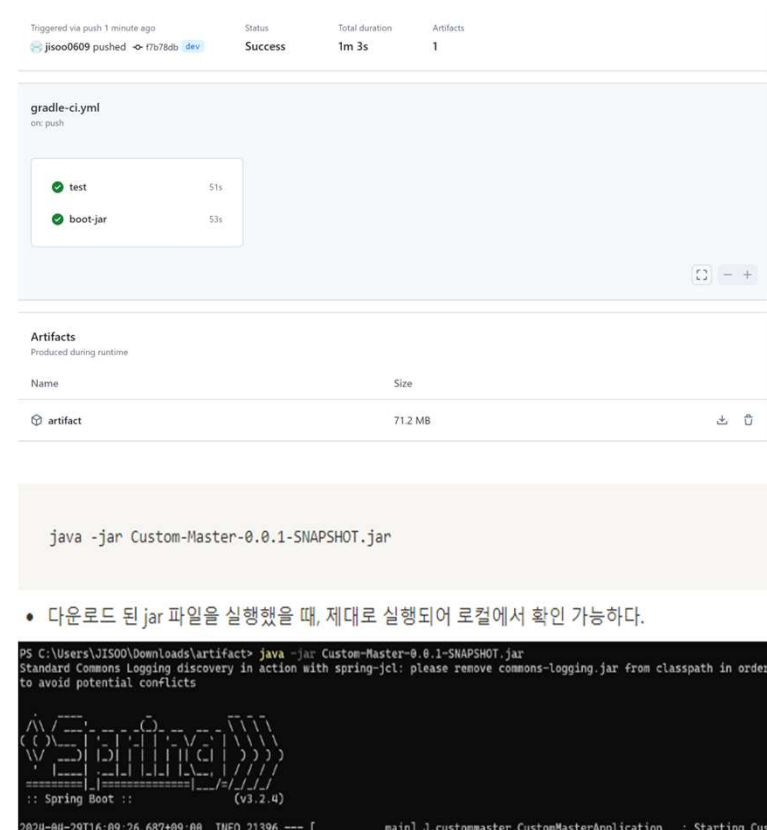
- .github/workflow 추가

gradle-ci.yml 작성

- jobs test 추가
- boot-jar 추가

GitHub Action 확인

- 설정이 제대로 실행되는지 확인한다.
- 생성된 artifact를 다운로드 받은 후 실행한다.



Triggered via push 1 minute ago
js00609 pushed · f7b78db · dev

Status: Success
Total duration: 1m 3s
Artifacts: 1

gradle-ci.yml
on: push

- test 51s
- boot-jar 53s

Artifacts
Produced during runtime

Name	Size
artifact	71.2 MB

java -jar Custom-Master-0.0.1-SNAPSHOT.jar

- 다운로드 된 jar 파일을 실행했을 때, 제대로 실행되어 로컬에서 확인 가능하다.

```
PS C:\Users\JIS00\Downloads\artifact> java -jar Custom-Master-0.0.1-SNAPSHOT.jar
Standard Commons Logging discovery in action with spring-jcl: please remove commons-logging.jar from classpath in order
to avoid potential conflicts

Spring Boot (v3.2.4)
2024-08-29T16:09:26.697+09:00 INFO 21396 --- [main] J.custommaster.CustomMasterApplication : Starting Cust
```

CI/CD – AWS CodeDeploy

AWS Setting

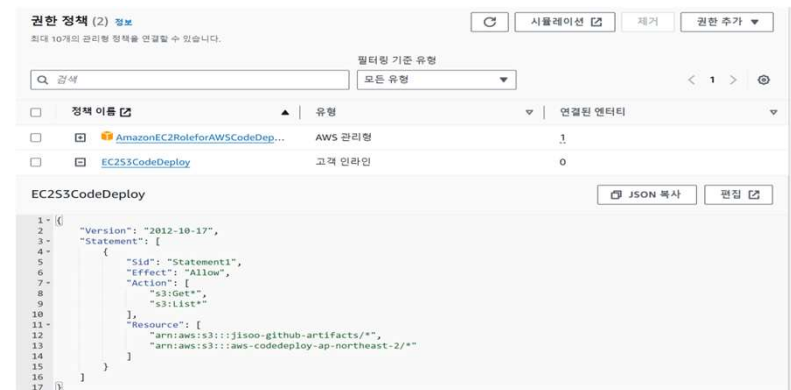
- S3에 ZIP 업로드
- appspec.yml
- scripts/application_start.sh 포함

EC2 Setting

- 권한 부여
- 서버에 CodeDeploy Agent 설치

CodeDeploy 설정

- 배포 그룹으로 묶어 배포



d-RKFSDMKY4

배포 상태

인스턴스에 애플리케이션 설치 중

1/1개의 인스턴스가 업데이트됨 ✔ 성공 100%

CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
7a76f50975e2	jisoo0609/custom-master:latest	80/tcp	/__cacert_entrypoin_	2 minutes ago	Up 2 minutes	0.0.0.0:8080->80
	:::8080->8080/tcp	custom-master-cd				

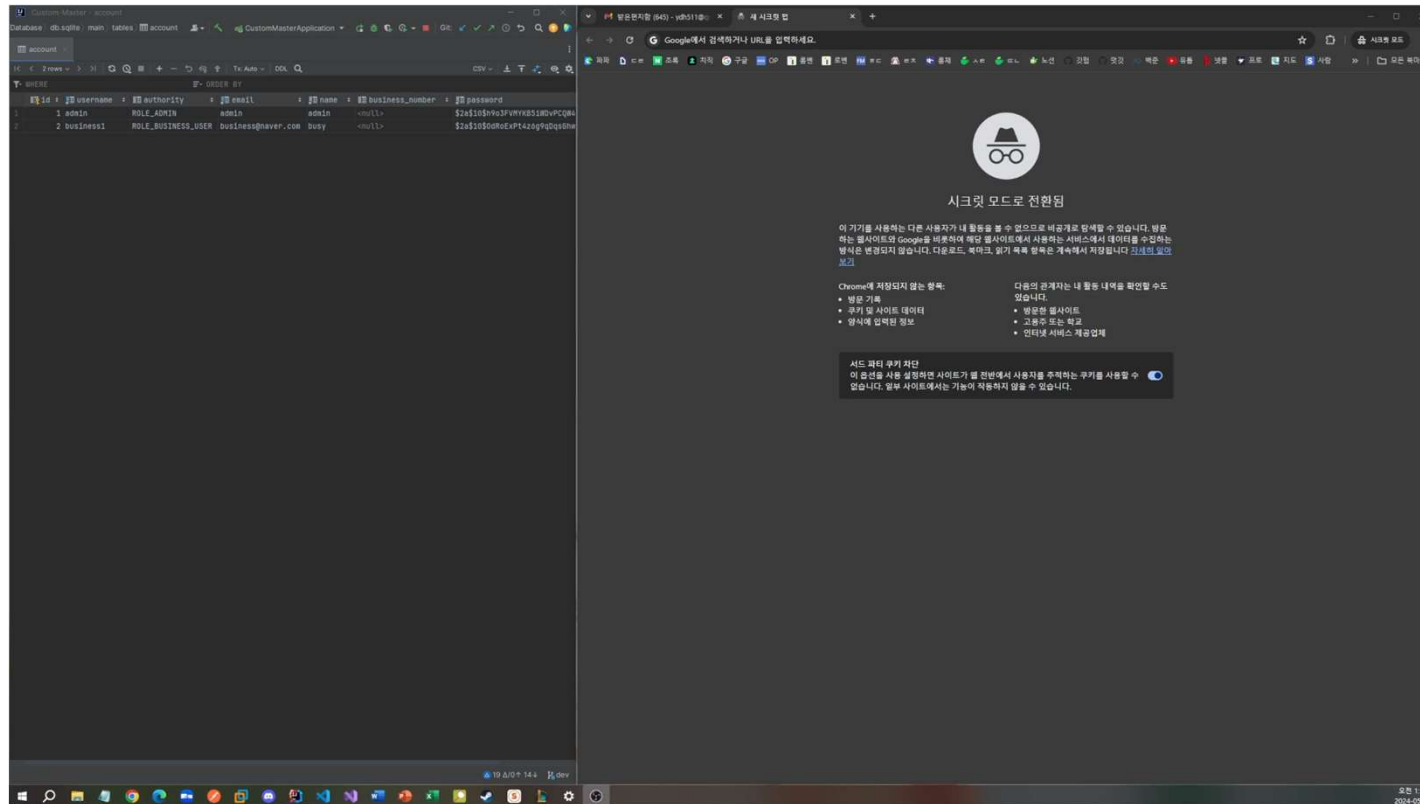
<http://43.203.226.154:8080/shop>



프로젝트 시연

시연 영상

시연 영상





진행 프로세스

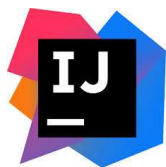
개발 일정 / 개발 환경

개발 일정

	월	04월																							5월
	주차	22주차						23주차						24주차						25주차					
	일정	담당	4/11	4/12	4/13	4/14	4/15	4/16	4/17	4/18	4/19	4/20	4/21	4/22	4/23	4/24	4/25	4/26	4/27	4/28	4/29	4/30	5/1	5/2	5/3
실제 진행 TASK	주제 선정	팀 전체	3																						
	DB 설계 및 계획	팀 전체	1	3																					
	아이디어톤 - PPT	박지수		1	1	1	1	2	3																
	로그인 기능 구현, 페이지제작	양동화		1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	3		
	매장 페이지 구현 - 매장 정보, 주문 요청 / 요청확인	양동화		1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	3		
	주문하기 - 고객	김솔찬		1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	3		
	주문 관리하기 - 주문 승인, 거절	박지수		1	1	1	2	2	2	3															
	결제	장수빈		1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	3			
	리뷰 구현	박지수/김세인/오시은		1	1	1	1	2	2	2	3														
	배포	박지수										1	1	1	1	1	1	1	1	1	1	2	2	2	3
	페이지 별 기능 시연	팀 전체																							3
	시연 영상 준비	팀 전체																			1	1	2	3	
	최종 PPT 발표자료 준비	박지수																			1	1	2	3	
																					1	2	2	3	

개발 환경

개발 및 협업 툴



Backend



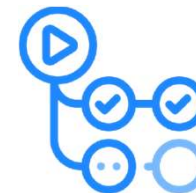
Security



DB



배포



API



감사합니다 🙏

GitHub - <https://github.com/likelion-backend-8th-pj2-team4-JuDBu/Custom-Master>

배포 페이지 - <http://43.203.226.154/shop>