

慢野

- Slow camping store





눌짝눌짝

- 도메인 주소 : <http://34.47.79.214/>
- 사용자 계정
 - Id : elicer@elice.com
 - Password : 1234
- 관리자 계정
 - Id : admin@elice.com
 - Password : 1234



팀원 소개



프로젝트 과정



평일 오전 10시 **스크럼**



김경래

저도 로컬에서 해봐서

오늘 한번 데이터베이스 연결해서 해봐야될 거 같아요



고의성

회원부분은 계속 수정중이라

이따 시간 내서 머지리퀘스트 해야될 거 같습니

3차 스크럼



날짜

2024년 5월 29일 오전 10:00



속설 추가



댓글 추가

1. API 명세서 작성해두기

2. API 명세서 토대로 ERD 작성하기

- ERD Cloud라는 프로그램 사용
- 같이 공동 작업하면서直观히 편치기
- 그리고 백엔드 코드层面에 피드백 부탁드려기

3차 세팅해두기

백엔드 코드层面

- 데이터베이스 관련 질문드리기
- 네이밍 컨벤션

ZEP 만들고 등

블록체인/암호 가미 있는 데이터베이스 ZEP

제작하고 투표해보면 향후 투표 결과를 투표해보세요. 현실 중간적인 투표입니다.
구성하고 투표할 수 있습니다. 누구나 쉽게 만들고 경험할 수 있는 서비스입니다.

<https://sep.vixplay.yt/w7k>



5차 스크럼



2024년 5월 31일 오전 10:00



속설 추가



댓글 추가

데이터베이스



VM에서

→ VM에 접속하고 나가기 해보기, [이상님께서 MySQL 설치하기](#)

코드 리뷰 날 정하기! (지금 토요일)

→ 목요일 Merge Request, 금요일 오피스아워 피드백

네이밍 컨벤션

- 그냥 ID로만

삭제 - Hard Delete, Soft Delete

- 전천히 만들면서 생각해보기
- 회원이나 주문 같은 정보는 [소프트로 하는게 좋을 것 같음](#)

폐기지 구조

- 도메인별 폐기지 구조



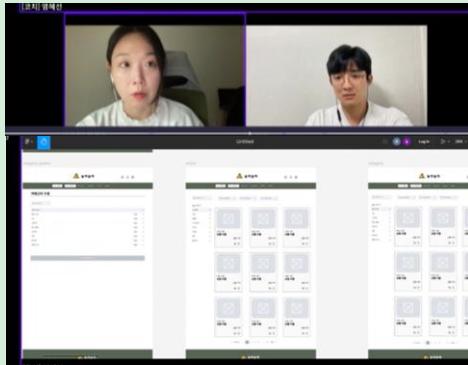


프로젝트 과정

코치님

BACK
이가람

FRONT
염혜선



백엔드 코치님 2

날짜: 2024년 5월 30일 오후 9:20

내이밍 천변선 지끼기

- id → 도메인_id로 옮기자, 그냥 id로 옮기지

+ 관리자는 회원으로 넣고, ROLE 탭을 부여하는 방향으로

- Spring Security 권한 부여 기능을 사용해서 회원 하나의 태이블에서 권한만 다르게 부여해서 구현

데이터타입

- String → VARCHAR 같은 형태로 변경해두기
- 중요하는게 좋을 것 같음

enum 값 → 자바 값으로 들어갈테니. 영여여야 함

- 주가 될 수 있을 때, 태이블에 적용하는게 귀찮으니 자바 클래스로 관리. 태이블에는 제약 안 걸어둬도 됨

created_at, updated_at은 있을때만 추가

삭제는 데이터를 그냥 날릴 수 있는 방법(HARD DELETE), 날리지 않고 쓰레그램 필드를 추가해서 업데이트를 하는 방법(SOFT DELETE) 두 가지



주 3회 오피스아워



매주 금요일 코드리뷰



Garam Lee @emteoh0313 · 1 week ago

p1: rest controller에서의 redirect는 HttpServletResponse.sendRedirect 메서드를 찾아보세요! 그리고 서버에서 리다이렉트를 처리하기 보다는 화면에서 처리하는 것이 더 보편적이에요

Maintainer

p1: @AuthenticationPrincipal이라는 어노테이션을 사용하면 로그인 객체를 주입받아서 쓸 수 있습니다~ 코드양을 좀 줄일 수 있어요

https://wildevelopertrain.tistory.com/324#google_vignette





프로젝트 과정



디버깅
정보 공유

[CLOUD3] 고의성 어제 오후 11:41
저도 지금 그 오류를 겪고 있어서

[CLOUD3] 김경래 어제 오후 11:41
https://velog.io/@pengoose_dev/React-Spring-Boot-Proxy-a-non-empty-string

[CLOUD3] 고의성 어제 오후 11:41
오 감사합니다!!

[CLOUD3] 김경래 어제 오후 11:41
저는 첫번째가 보고 해결하긴 했어요

[CLOUD3] 고의성 어제 오후 11:41
넵 참고할게요!

[CLOUD3] 박원정 어제 오후 11:48
경래님 의성님이 사랑하신대요

[CLOUD3] 고의성 어제 오후 11:48
사랑합니다

[CLOUD3] 박원정 오늘 오전 10:00
<https://mosel.tistory.com/entry/Git-git-pull-시-merge-오류가-날때-강제-git-pull-및-어쓰기-방법>

씹어먹는 블로그

[GIT] git pull 시 merge 오류가 날 때 강제 git pull 및 어쓰기 방법

git pull 시 stash 또는 merge 를 우선 하라고 나올 때, 중요하지 않은 파일 또는 덮어쓰기 해도 무방한 파일일때는 덮어쓰기 하는 것이 빠르다. 1. git pull 받을 목록을 repository 에서 업데이트 git fetch --all 2. git pull 으로 head를 최신으로 가리킨다 git reset --hard origin/master 3. git pull 로 확인 git pull 이렇게 하면 그냥 최신버전 헤드를 가리킴으로써 덮어쓰기와 같은 효과가 된다.



일정 관리
진행 상황 공유



■ 기능	■ HTTP Method	■ API Path	■ 담당자(BE)	■ 진행 상태(BE)	■ 담당자(FE)	■ 진행 상태(FE)
로그인 페이지 이동	GET	/login	고.의성 김.김연지	● 완료	고.의성 김.김연지	● 시작 전
マイ페이지 이동	GET	/mypage	고.의성 김.김연지	● 완료	고.의성	● 시작 전
관리자 페이지 이동	GET	/admin	고.의성 김.김연지	● 완료	고.의성 김.김연지	● 시작 전
회원가입 페이지 이동	GET	/membership	고.의성 김.김연지	● 완료	김.김연지	● 시작 전
회원가입	POST	/membership	고.의성 김.김연지	● 완료	김.김연지	● 시작 전
회원정보 수정	PUT	/mypage/	고.의성 김.김연지	● 완료	고.의성	● 시작 전
회원탈퇴	DELETE	/mypage/	고.의성 김.김연지	● 완료	고.의성	● 시작 전
로그인 요청	POST	/login	고.의성 김.김연지	● 완료	김.김연지	● 시작 전
로그아웃	POST	/logout	고.의성 김.김연지	● 완료	김.김연지	● 시작 전



공식 모임

1차 모임

2차 모임

3차 모임



서비스 기획

- 목표 회원이 상품을 주문할 수 있는 기본적인 기능을 가진 쇼핑몰 개발



온라인으로 간편하게 캠핑 도구를 구입할 수 있는 쇼핑몰이 없을까?



캠핑 관련 장비 구매 가능한 온라인 쇼핑몰 서비스를 구현 해보자!



서비스 기획

- Logo



- Main Color



#586555

페이지 포인트 컬러 통일

- Concept

자연과 더불어 느리게 가는 편안한 분위기의 캠핑 쇼핑몰

- Header & Footer

The image displays two versions of a website's header and footer design. Both versions feature a logo with a yellow tent icon and the Korean text "늘 짹늘 짹".
The top version has a white background. It includes a dark teal horizontal bar at the bottom containing links: "브랜드", "카테고리", "트트.티프", "진남.매트", "피니시", and "리아워".
The bottom version has a dark teal background. It includes a white horizontal bar at the bottom containing links: "박원정", "고의성", "김경래", "김연지", "김이삭", and "조한취". Below this bar, the text "늘 짹늘 짹 총 2024 All rights reserved." is visible.
In the bottom right corner of the dark teal section, there is a QR code.

도메인



회원



주문



상품



카테고리



브랜드



장바구니



기술 스택

Back



Front



커뮤니케이션

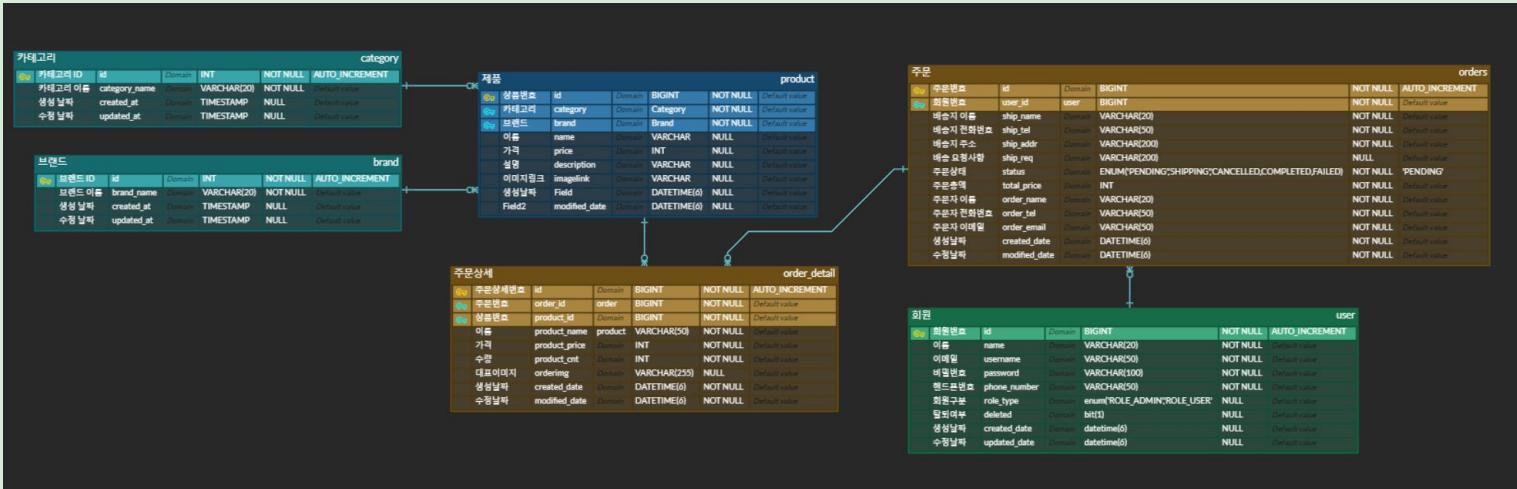


기획



데이터베이스

- ERD



- 구현 방법



API 명세



회원

기능	HTTP Method	API Path	담당자(BE)	진행 상태(BE)	담당자(FE)	진행 상태(FE)
로그인 페이지 이동	GET	/login	고의성	김연지	● 완료	고의성 김 김연지 ● 완료
마이페이지 이동	GET	/mypage	고의성	김연지	● 완료	고의성 ● 완료
관리자 페이지 이동	GET	/admin	고의성	김연지	● 완료	고의성 김 김연지 ● 완료
회원가입 페이지 이동	GET	/membership	고의성	김연지	● 완료	김연지 ● 완료
회원가입	POST	/membership	고의성	김 연지	● 완료	김 연지 ● 완료
회원정보 수정	PUT	/mypage/	고의성	김연지	● 완료	고의성 ● 완료
회원탈퇴	DELETE	/mypage/	고의성	김연지	● 완료	고의성 ● 완료
로그인 요청	POST	/login	고의성	김연지	● 완료	김연지 ● 완료
로그아웃	POST	/logout	고의성	김연지	● 완료	김연지 ● 완료



주문

기능	HTTP Method	API Path	담당자(BE)	진행 상태(BE)	담당자(FE)	진행 상태(FE)
마이페이지 주문 내역 조회	GET	/api/v1/mypage/orders	조한위	● 완료	조한위	● 완료
마이페이지 주문 내역 상세 조회	GET	/api/v1/mypage/orders/{orderId}	조한위	● 완료	조한위	● 완료
마이페이지 주문 정보 수정	PUT	/api/v1/mypage/orders/{orderId}	조한위	● 완료	조한위	● 완료
마이페이지 주문 취소	DELETE	/api/v1/mypage/orders/{orderId}	조한위	● 완료	조한위	● 완료
주문 추가	POST	/orders	조한위	● 완료	조한위	● 완료
주문 성공 페이지 조회	GET	/orders/success	조한위	● 완료	조한위	● 완료
주문 실패 페이지 조회	GET	/orders/failure	조한위	● 완료	조한위	● 완료
관리자 주문 관리 조회	GET	/admin/orders	조한위	● 완료	김이삭	● 완료
관리자 주문 관리 상태별 조회	GET	/admin/orders/status	조한위	● 완료	김이삭	● 완료
관리자 주문 정보 수정	PUT	/admin/orders/{id}/status	조한위	● 완료	김이삭	● 완료
관리자 주문 취소	DELETE	/admin/orders/{id}	조한위	● 완료	김이삭	● 완료



주문

기능	HTTP Method	API Path	담당자(BE)	진행 상태(BE)	담당자(FE)	진행 상태(FE)
상품 추가	POST	/admin/product/create	경래 김	● 완료	경래 김	● 완료
새로 상품 조회	GET	/product/{productId}	경래 김	● 완료	경래 김	● 완료
상품 수정	PUT	/admin/product/update/{productId}	경래 김	● 완료	경래 김	● 완료
상품 삭제	DELETE	/admin/product/delete/{productId}	경래 김	● 완료	경래 김	● 완료
상품 최신 추가 순으로 정렬	GET	/product/latest	경래 김	● 완료	경래 김	● 완료
상품 최신 업데이트 순으로 정렬	GET	/product/modify	경래 김	● 완료	경래 김	● 완료
모든 상품 출력	GET	/product/all	경래 김	● 완료	경래 김	● 완료

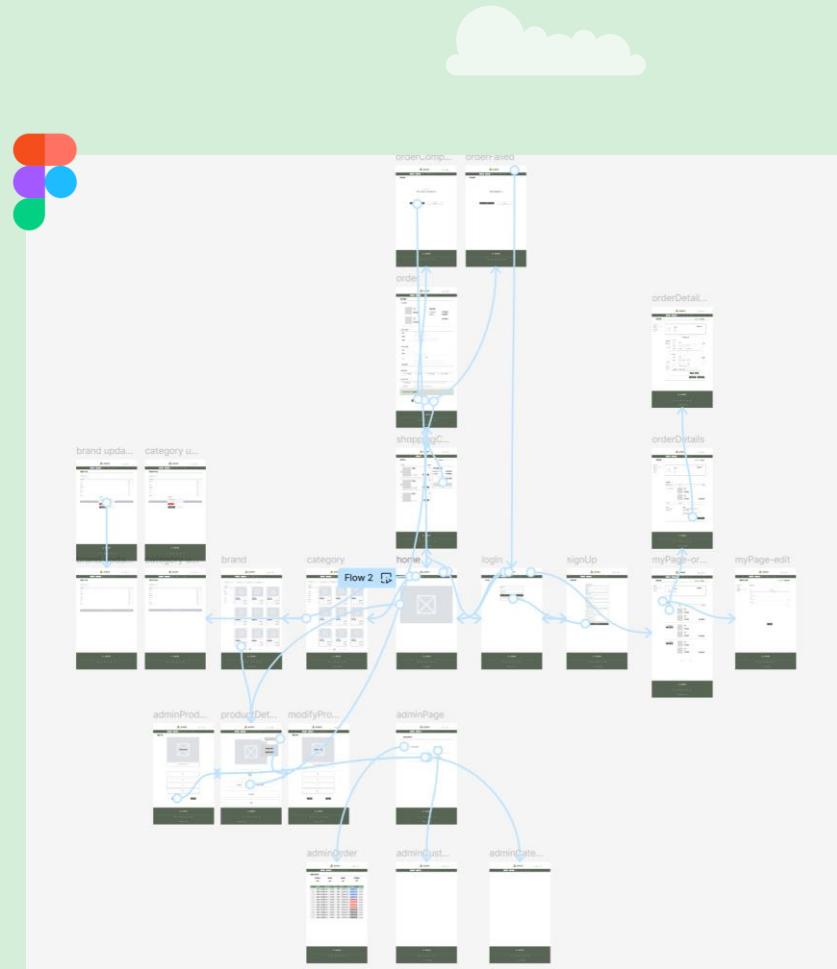
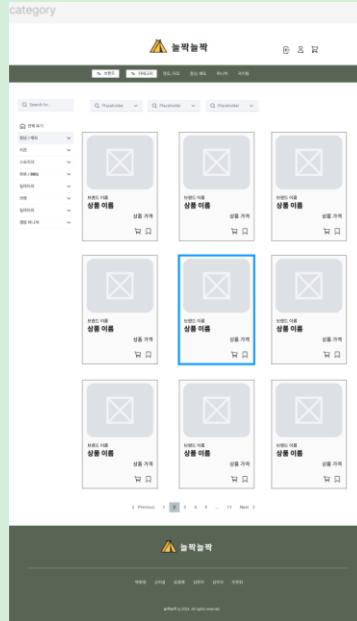
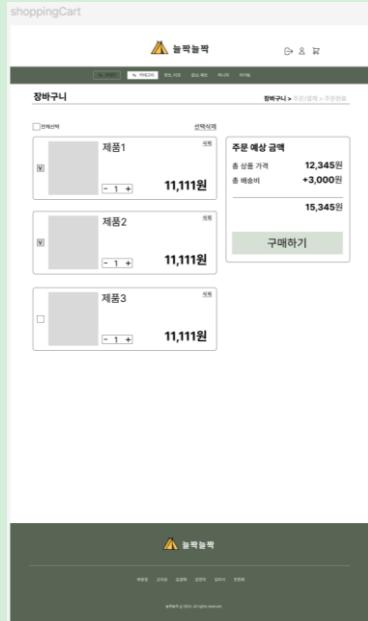


카테고리/브랜드

기능	HTTP Method	API Path	담당자(BE)	진행 상태(BE)	담당자(FE)	진행 상태(FE)
카테고리 별 상품 조회	GET	/category/{categoryId}	박원정	● 완료	박원정	● 완료
특정 상품 조회	GET	/product/{id}	박원정	● 완료	박원정	● 완료
카테고리 수정 화면 이동	GET	/admin/category	박원정	● 완료	박원정	● 완료
카테고리 추가	POST	/admin/category/post	박원정	● 완료	박원정	● 완료
카테고리 수정	PUT	/admin/category/edit/{categoryId}	박원정	● 완료	박원정	● 완료
카테고리 삭제	DELETE	/admin/category/delete/{categoryId}	박원정	● 완료	박원정	● 완료
브랜드	기능	HTTP Method	API Path	담당자(BE)	진행 상태(BE)	담당자(FE)
브랜드 별 상품 조회	GET	/brand/{brandId}	박원정	● 완료	박원정	● 완료
특정 상품 조회	GET	/product/{id}	박원정	● 완료	박원정	● 완료
브랜드 수정 화면 이동	GET	/admin/brand	박원정	● 완료	박원정	● 완료
브랜드 추가	POST	/admin/brand/post	박원정	● 완료	박원정	● 완료
브랜드 수정	PUT	/admin/brand/edit/{brandId}	박원정	● 완료	박원정	● 완료
브랜드 삭제	DELETE	/admin/brand/delete/{brandId}	박원정	● 완료	박원정	● 완료



와이어 프레임



홈페이지

[브랜드](#) [카테고리](#) [매트 / 터프](#) [침낭, 매트](#) [파티차](#) [리뷰들](#)

Field Station
닿는 순간 느껴지는 편안함
STRESS ZERO
필드스테이션
프리미엄 폴딩 카펫매트

제작: 30000원

신상품

모자
페도라스타일의 누 껐식햇
가격: 30000원

의자
다른 의자보다 저렴해요!
가격: 20000원

의자
의자
가격: 50000원

진짜 편해보이는 침낭
제가 지금 풀어가고 싶네요
가격: 55000원

최신 업데이트 상품

진짜 편해보이는 침낭
제가 지금 풀어가고 싶네요
가격: 55000원

모자
페도라스타일의 누 껐식햇
가격: 30000원

의자
다른 의자보다 저렴해요!
가격: 20000원

좁은 비싼 텐트
비싸긴 해도 성능같은 해요.
가격: 30000원

[백화점](#) [고객성](#) [김경래](#) [김연자](#) [김아식](#) [조한위](#)

농작물 2024 All rights reserved.

로그인/회원가입

어떻게 기능하게 만들었을까?!

로그인

이메일 *

비밀번호



로그인

회원가입

회원가입

이름 *

이메일 *

비밀번호



비밀번호 확인



전화번호 *

회원가입



로그인

```
try {
    handleLoginAttempt(loginDto);
    UsernamePasswordAuthenticationToken authToken = new UsernamePasswordAuthenticationToken(loginDto.getUsername(), loginDto.getPassword());
    return authenticationManager.authenticate(authToken);
} catch (UsernameNotFoundException e) {
```

로그인시 토큰 생성 ->
AuthenticationManager에게 토큰 전달, 인증 요청

```
//로그인 성공시 실행하는 메소드 (여기서 JWT를 발급하면 됨)
스 [Cloud3] 박원정 +1
@Override
protected void successfulAuthentication(HttpServletRequest request, HttpServletResponse response, FilterChain chain, Authentication authentication) throws IOException {
    System.out.println("로그인 성공");

    //UserDetailsS
    CustomUserDetails customUserDetails = (CustomUserDetails) authentication.getPrincipal();

    String username = customUserDetails.getUsername();

    Long id = customUserDetails.getId();

    Collection<? extends GrantedAuthority> authorities = authentication.getAuthorities();
    Iterator<? extends GrantedAuthority> iterator = authorities.iterator();
    GrantedAuthority auth = iterator.next();

    String role = auth.getAuthority();

    String token = jwtUtil.createJwt(id, username, role, JWTConfig.EXPIRATION);

    System.out.println(token);

    response.addHeader("Authorization", "Bearer " + token);
}
```

로그인 성공시 JWT 발급
응답 헤더에 토큰 넘겨줌

```
if (!response.ok) {
    const errorData = await response.json();
    // throw new Error(errorData.message);
    throw new Error(`errorCode: ${errorData.errorMessage}`);
}

// 토큰을 응답 헤더에서 가져오기
const token = response.headers.get('Authorization');

// AuthContext의 login 함수 호출
login(token);

if (localStorage.getItem('role') === 'ROLE_ADMIN') {
    navigate('/admin');
} else {
    navigate('/');
}

> localStorage.token
<- 'Bearer eyJhbGciOiJIUzI1NiJ9eyJpZCI6MiwidXNlcj5hbWUiOiJ1c2VyQG5hdmVyb25lST0xfX1vTRViiLCJpYXQiOjE3MTg5NjA1MTAsImV4cCI6MTcxODk2NDExMH0.7B4u_kZhShB1t7U_dxPBYQ0poPs8_nqk'
```

로그인 정상 처리시
토큰 localStorage에 저장



회원가입

- Service

```
# [Cloud3] 김연지
public User membershipProcess(User user){

    if (userRepository.existsByUsername(user.getUsername())) {
        throw new IllegalArgumentException("이미 존재하는 이메일입니다.");
    }

    user.setPassword(bCryptPasswordEncoder.encode(user.getPassword()));

    return userRepository.save(user);
}
```

비밀번호 암호화
-> 실제 데이터베이스에 저장

- Controller

DTO에 담긴 값을 통해 회원가입 예외 발생 -> 해당 에러 return

```
# [Cloud3] 김연지
@PostMapping("/api/v1/membership")
public ResponseEntity<Object> membershipProcess(@RequestBody @Valid MembershipDTO membershipDto, BindingResult bindingResult) {

    Map<String, Object> response = new HashMap<>();
    if (bindingResult.hasErrors()) {
        response.put("success", false);
        response.put("errors", bindingResult.getFieldErrors().stream()
            .collect(Collectors.toMap(
                FieldError::getField,
                FieldError::getDefaultMessage
            )));
        System.out.println(response);
        return ResponseEntity.badRequest().body(response);
    }

    if (userService.isEmailDuplicated(membershipDto.getUsername())) {
        // 이메일 중복 처리
        response.put("success", false);
        response.put("username", "이미 존재하는 이메일 주소입니다.");
        return ResponseEntity.badRequest().body(response);
    }

    try {
        User user = mapper.membershipDtoToUser(membershipDto); // DTO를 Entity로 변환
        userService.membershipProcess(user);
        return new ResponseEntity<>(HttpStatus.CREATED);
    } catch (Exception ex) {
        ExceptionDTO exceptionDTO = new ExceptionDTO("INTERNAL_SERVER_ERROR", "An unexpected error occurred.");
        return new ResponseEntity<>(exceptionDTO, HttpStatus.INTERNAL_SERVER_ERROR);
    }
}
```



관리자 마이페이지 / 회원 마이페이지



токен ROLE localStorage에 저장
-> 접근 가능 페이지 설정

회원 탈퇴



소프트 딜리트로
회원 탈퇴 기능 구현!



회원 탈퇴

- Controller

- Service

```
// 사용자 비밀번호 확인 메소드
// 서비스 메서드 시그니처가 필요한 형식을 맞추도록 확인
1 usage ▲ [Cloud3]+3
public boolean checkPassword(UserDTO user, String inputPassword) {
    if (user == null) {
        throw new IllegalArgumentException("사용자 정보를 찾을 수 없습니다.");
    }

    String storedPassword = user.getPassword();

    log.info("Input Password: {}", inputPassword);
    log.info("Stored Password: {}", storedPassword);
    if (!bcryptPasswordEncoder.matches(inputPassword, storedPassword)) {
        throw new IllegalArgumentException("사용자의 비밀번호 정보가 올바르지 않습니다.");
    }

    return true; // 여기까지 도달하면 비밀번호가 일치
}
```

사용자 비밀번호와 입력 비밀번호 비교
일치 -> return 불일치 -> `IllegalArgumentException`

```
no usages ▲ [Cloud3] 고급설
@PostMapping("/api/v1/checkPasswordForUpdate")
public ResponseEntity<String> checkPasswordForUpdate(@RequestBody @Valid CheckPasswordDTO checkPasswordDTO, @AuthenticationPrincipal CustomUserDetails customUserDetails) {
    return checkPasswordAndProceed(customUserDetails, checkPasswordDTO, successMessage);
}

no usages ▲ [Cloud3] 고급설
@PostMapping("/api/v1/checkPasswordForDelete")
public ResponseEntity<String> checkPasswordForDelete(@RequestBody @Valid CheckPasswordDTO checkPasswordDTO, @AuthenticationPrincipal CustomUserDetails customUserDetails) {
    return checkPasswordAndProceed(customUserDetails, checkPasswordDTO, successMessage);
}
2 usages ▲ [Cloud3]+2
private UserDetails checkPasswordAndProceed(CustomUserDetails customUserDetails, CheckPasswordDTO checkPasswordDTO, String successMessage) {
    if (customUserDetails==null) {
        log.info("check customUserDetails == null");
        return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("사용자 정보를 찾을 수 없습니다.");
    }

    String role = customUserDetails.getAuthorities().iterator().next().getAuthority();
    System.out.println(role);
    System.out.println(checkPasswordDTO);

    String name = customUserDetails.getUsername();
    UserDTO userDTO = userService.findByUsername(name);

    // @GetMapping에서 가져온 비밀번호 정보 사용
    String storedPassword = userDTO.getPassword();

    log.info("Stored Password: {}", storedPassword);

    try {
        boolean passwordsMatch = userService.checkPassword(userDTO, checkPasswordDTO.getPassword());
        if (passwordsMatch) {
            Logger.getLogger(this.getClass().getName()).info( msg="Password check successful for user: " + name);
            return ResponseEntity.ok(successMessage);
        } else {
            return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("비밀번호가 일치하지 않습니다.");
        }
    } catch (IllegalArgumentException e) {
        // 비밀번호가 일치하지 않는 경우 처리
        Logger.getLogger(this.getClass().getName()).info( msg="Password check failed for user: " + name + " - " + e.getMessage());
        return ResponseEntity.status(HttpStatus.UNAUTHORIZED).body("비밀번호가 일치하지 않습니다.");
    } catch (Exception e) {
        // 그 외 예외 처리
        Logger.getLogger(this.getClass().getName()).severe( msg="Error during password check for user: " + name + " - " + e.getMessage());
        return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("비밀번호 확인 중 오류가 발생했습니다.");
    }
}
```

토큰 -> 비밀번호 일치 여부 체크
비밀번호 일치 시 수정 or 탈퇴 이동



주문 생성

주문서

주문 목록

	▶ 최근 판매	2023년 7월 20일
	주문 금액	30,000원
	총 상품 가격	30,000원
	총 배송비	>3,000원
	최종 결제 금액	33,000원
	50,000원 이상 구매시 배송비 무료	

주문자 정보

이름	홍길동
연락처	010-1234-5678
이메일	hong@example.com
주소	서울특별시 종로구 123-12, 201호
도로명	한국 대학교
우편번호	03143

배송지 정보

이름	홍길동
연락처	010-1234-5678
주소	서울특별시 종로구 123-12, 201호
도로명	한국 대학교
우편번호	03143

결제 정보

카카오페이 신용카드 후디온 알리 놀시간 계좌이체

주문자 동의

개인정보 수집 및 이용
주문자 정보(성명, 이메일) / 배송지 정보(이름, 연락처, 주소) / 은행 명의에 따라 8인트 보안
 주문 충돌
주문은 같은 상품을 확인하여서 주문 진행에 불리합니다.

최종 결제 금액: 33,000원

주문하기

```
<Button
    variant="contained"
    color="primary"
    size="large"
    sx={
        backgroundColor: '#586555',
        color: '#ffffff',
        fontWeight: 'bold',
        borderRadius: '8px',
        boxShadow: 'none',
        '&:hover': {
            backgroundColor: '#82957E',
        },
    }
    onClick={handleSubmitOrder}
    disabled={[
        !paymentMethod || !agreementConfirmed.personalInfo || !agreementConfirmed.orderAgreement
    ]}
> 주문하기
</Button>
```

필수사항 미체크시
프론트에서 버튼 비활성화!

```
▲ [Cloud9] 조합된  
@PostMapping("http://orders")  
@PreAuthorize("hasRole('ROLE_USER')")  
public ResponseEntity addOrder(@AuthenticationPrincipal CustomUserDetails userDetails,  
                                @Valid @RequestBody OrderRequest orderRequest, BindingResult bindingResult,  
                                @RequestParam boolean paymentConfirmed,  
                                @RequestParam boolean agreementConfirmed) {  
    if (bindingResult.hasErrors()) {  
        StringBuilder errorMessage = new StringBuilder();  
        bindingResult.getFieldErrors().forEach(error -> {  
            errorMessage.append(error.getDefaultMessage()).append("\n");  
        });  
        return ResponseEntity.status(HttpStatus.BAD_REQUEST).body(errorMessage.toString());  
    }  
  
    if (!paymentConfirmed) {  
        return ResponseEntity.badRequest().body("현재 주문에 체크해주셔요.");  
    }  
  
    if (!agreementConfirmed) {  
        return ResponseEntity.badRequest().body("주문자 동의에 체크해주셔요.");  
    }  
  
    try {  
        Order order = orderService.addOrder(orderRequest);  
        return ResponseEntity.ok(order);  
    } catch (RuntimeException e) {  
        logger.error("Error processing order: " + e);  
        return ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("주문을 처리하는 중에 오류가 발생했습니다.");  
    }  
}
```

백에서도 더블 체크!



주문 목록 조회

マイページ

- 회원정보
- 주문목록
- 회원탈퇴

주문목록

마이페이지 > 주문목록

주문번호	주문일자	주문상세 조회	주문상태	가격	상품준비중
00054	2024. 6. 21.	주문상세 조회	진짜 판매보이는 침낭	110,000 원	취소 완료
00059	2024. 6. 21.	주문상세 조회	의자	23,000 원	상품준비중
00058	2024. 6. 21.	주문상세 조회	음 비캔 텐트	33,000 원	상품준비중

< 1 2 3 4 5 ... 18 >

```
// 마이페이지 주문내역 조회
▲ [Cloud3] 조한휘
@GetMapping(@RequestMapping("api/v1/mypage/orders"))
@PreAuthorize("hasRole('ROLE_USER')")
public ResponseEntity<?> getAllOrdersByUser(@AuthenticationPrincipal CustomUserDetails userDetails) {
    try {
        User user = orderService.findByUsername(userDetails.getUsername());
        List<Order> orders = orderService.getAllOrdersByUserId(user.getId());
        return ResponseEntity.ok(orders);
    } catch (RuntimeException e) {
        return ResponseEntity.status(HttpStatus.NOT_FOUND).body("주문 내역을 찾을 수 없습니다.");
    }
}
```

@AuthenticationPrincipal
로그인 객체 주입하여 조회!

브랜드 조회

블랜드 검색

전체 보기

스노우피크

스프링바

노로디스크

레트로스

노스페이스

네이처하이크

코베아

챌리녹스

아이더

전체 상품

줌 비싼 텐트

진짜 편해보이는 침낭

의자

의자

모자

사이드바로 편리하고
깔끔한 레이아웃!

```
src > layouts > BrandLayout.js
1 import React from 'react';
2 import { Outlet } from 'react-router-dom';
3 import { Grid } from '@mui/material';
4 import BrandSidebar from '../pages/brand/BrandSidebar';
5
6 const BrandLayout = () => {
7   return (
8     <Grid container>
9       <Grid item xs={3}>
10         <BrandSidebar />
11       </Grid>
12       <Grid item xs={9}>
13         <Outlet />
14       </Grid>
15     </Grid>
16   );
17 }
18
19 export default BrandLayout;
```



브랜드 CRUD

```
1 // src/components/Modal.jsx
2 import React from 'react';
3 import './Modal.css';
4 import { Typography } from '@mui/material';
5
6 const Modal = ({ show, title, children, onClose }) => {
7   if (!show) {
8     return null;
9   }
10
11   return (
12     <div className="modal-overlay">
13       <div className="modal">
14         <Typography sx={{ fontWeight: 'semibold', fontSize: '1.2rem', marginBottom: '10px' }>
15           {title}
16         </Typography>
17         {children}
18       </div>
19     </div>
20   );
21 };
22
23 export default Modal;
24
```

```
// src/components/Modal.jsx
import React from 'react';
import './Modal.css';
import { Typography } from '@mui/material';

const Modal = ({ show, title, children, onClose }) => {
  if (!show) {
    return null;
  }

  return (
    <div className="modal-overlay">
      <div className="modal">
        <Typography sx={{ fontWeight: 'semibold', fontSize: '1.2rem', marginBottom: '10px' }}>
          {title}
        </Typography>
        {children}
      </div>
    </div>
  );
}

export default Modal;
```

한 페이지에서 CRUD 구현을 위해
Modal로 처리하여 Component로 통합!



상품 CRUD

모달 품으로
수정, 추가 가능 구현!

제품 목록 - 관리자 화면

ID	이름	가격	설명	브랜드	카테고리	이미지	수정	삭제
1	绾 비싼 텐트	30000	비싸긴 해도 성능값은 해요.	스노우피크	텐트 / 타프		수정	삭제
3	진짜 편해보이는 침낭	55000	제가 지금 들어가고 싶네요	노르디스크	침낭, 매트		수정	삭제
4	의자	50000	의자	스프링바	파니처		수정	삭제
6	의자	20000	다른 의자보다 저렴해요!	스프링바	파니처		수정	삭제
7	모자	30000	페도라스타일의 뉴 애쉬햇	노르디스크	의류		수정	삭제

제품 추가

제품명: 풀 비싼 텐트
가격: 30000
설명: 비싸긴 해도 성능값은 해요.
브랜드: 스노우피크
카테고리: 텐트 / 타프
이미지 링크: <https://idoogen.com/web/product/big/202405/5dad7e0847c50ac228d283>

제품 수정

제품명: 풀 비싼 텐트
가격: 30000
설명: 비싸긴 해도 성능값은 해요.
브랜드: 스노우피크
카테고리: 텐트 / 타프
이미지 링크: <https://idoogen.com/web/product/big/202405/5dad7e0847c50ac228d283>

스프링바

정말 삭제하시겠습니까?

삭제 닫기

장바구니 상세

장바구니

장바구니 > 주문/결제 > 주문완료

프론트에서 localStorage를 이용하여 구현!

주문서

주문 목록

주문자 정보

주문 예상 금액

총 상품 가격 85,000 원
총 배송비 0 원
50,000원 이상 구매시 무료배송

85,000 원

구매하기

전체선택

전체삭제

삭제

삭제

총 비싼 텐트
30,000원

1

진짜 편해보이는 침낭
55,000원

1

총 비싼 텐트
수량: 1
30,000원

진짜 편해보이는 침낭
수량: 1
55,000원

주문 금액
총 상품 가격: 85,000원
총 배송비: 0원

최종 결제 금액: 85,000원
50,000원 이상 구매시 배송비 무료

이용
통길동

트러블 슈팅 1

문제 상황



왜 Spring Boot 프로젝트에서만 SSH 터널링 연결이 안될까요?

원인



설명된 글이 대부분 Maria DB 기준이에요!
MySQL로 설정값을 바꿔봅시다.

해결



VM서버의 MySQL이 사용하는 포트의 방화벽을 해제하였는데도 보안상 연결이 되지 않아
SSH터널링을 통해 해결해야 된다고 합니다. 터미널로 VM서버에 접속할 때
localhost:9091 접근을 SSH 터널링으로 감싸 원격 SSH 서버의 localhost:3036
서비스의 연결에 수행하도록 명령어를 사용하여 접속했습니다. 이 접속을 유지하면서
Spring Boot에서 데이터베이스 접속 주소의 포트를 9091로 변경해줘서 해결했습니다.

트러블 슈팅 2

문제 상황



프로젝트를 VM에서 배포했는데 403 에러가 발생해요!

원인



아마도 CORS 에러 때문인 것 같아요

해결



Spring Boot 프로젝트에 CORS 설정을 해놓아서 로컬에서는 정상작동하였는데
VM서버에 같은 설정으로 올려도 CORS 에러가 발생했습니다. 프론트와 백에서
localhost 주소가 쓰인 곳의 주소를 전부 VM 서버 IP로 바꿔주고 nginx 설정파일에서
'Access-Control_Allow' 관련 헤더들을 설정해주어 해결했습니다!



소감

고의성

프로젝트를 하면서 어려웠던 부분

한 파트를 둘어서 나눠 맡게 되어 진행을 했는데 각자 맡은 부분만 하면 될 거라 생각했던 초반과는 달리 겹치는 부분들이 많아 코딩에 어려움이 많이 생겼고 개인적으로 공부가 많이 부족해 따라가기가 많이 버거워 사실상 포기하기 직전까지 갔던 상황이었다.

어려웠던 부분을 극복한 경험

죄송스럽고 무례가 될 수 있지만 같은 파트를 맡은 연지님께 연지님께서 백엔드 부분을 맡아주시고 내가 프론트엔드 부분을 맡는 게 어떨까 하는 질문을 했는데 연지님께서 내가 반쯤 정신이 나간 걸 아셨는지 그래도 같이 해보자고 하시며 이끌어주셨다. 그 때 했던 잠깐의 소통이 나에게는 큰 힘이 되었고 그 이후로 자주 소통을 하며 문제들을 해결해나갔다.

공부할 때 도움이 되었던 노하우, 꿀팁

GPT를 잘 활용하자

전반적인 소감

소통의 중요성을 뼈저리게 느낀 프로젝트였다. 소통을 통해 맨날도 많이 회복하였고 오히려 코딩에 더 집중할 수 있었던 것 같다. 제주도에 살아서 오프라인 모임을 갖게 되면 참여가 힘들었는데 이번에 큰 맘 먹고 참여해 본 경험도 좋았던 것 같다. 시작때는 부담감이 심했지만 마무리는 엄청난 만족감을 느끼게 되는 것 같다. (사실 팀원들이 나를 버스태워준 덕분인 것 같다)



서비스 시연



Thank you!

Do you have any questions?

CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)

