

Project Progress Report Presentation

32191597 박민규
32191585 박도훈
32222174 서영빈
32223436 이은서



목 차

현재 구현 사항 - 로그인, 회원가입

01

현재 구현 사항 - 유저 프로필

02

현재 구현 사항 - 게시판 및 코멘트

03

현재 구현 사항 - 게시글 등록페이지 / 상세페이지

04

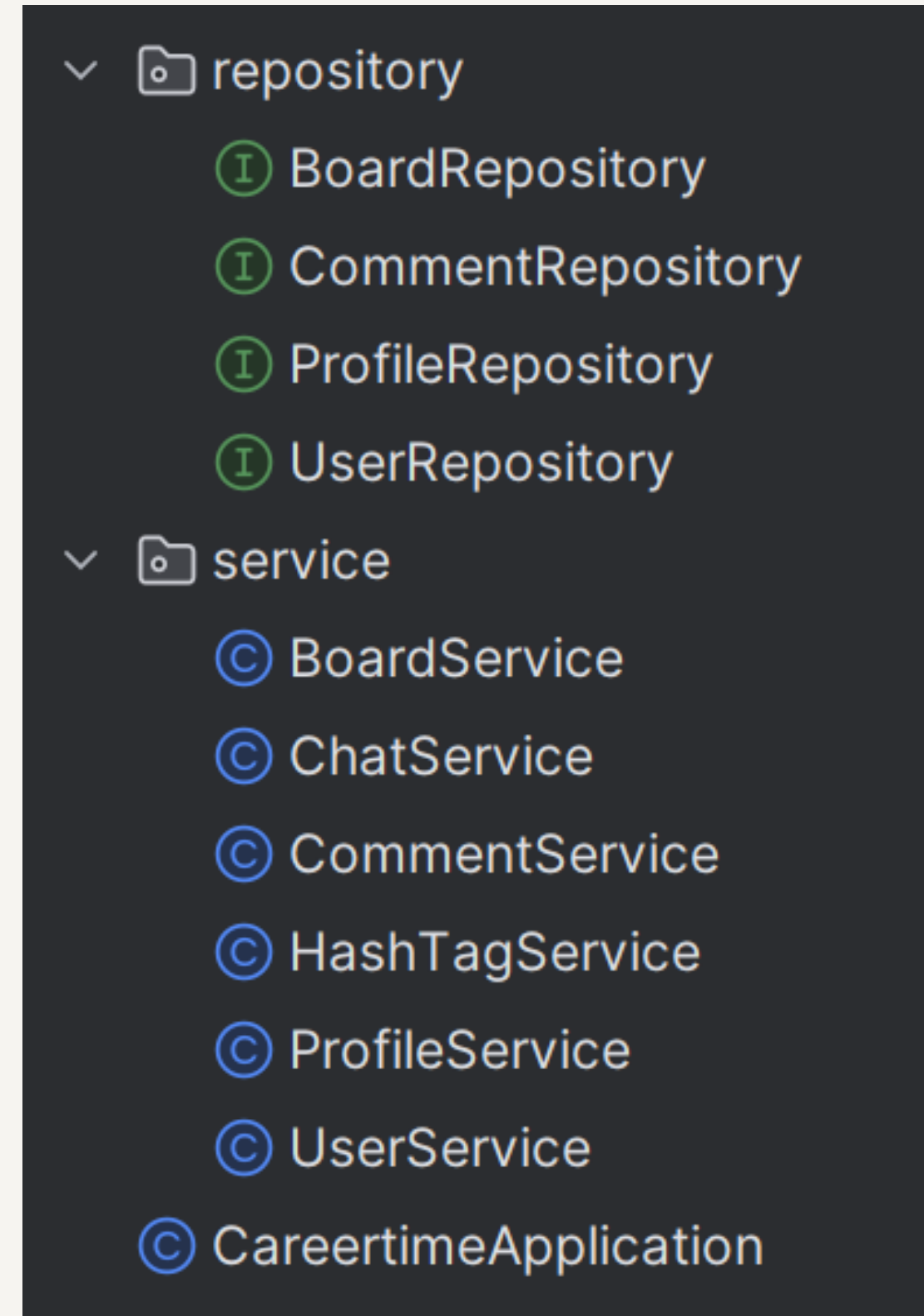
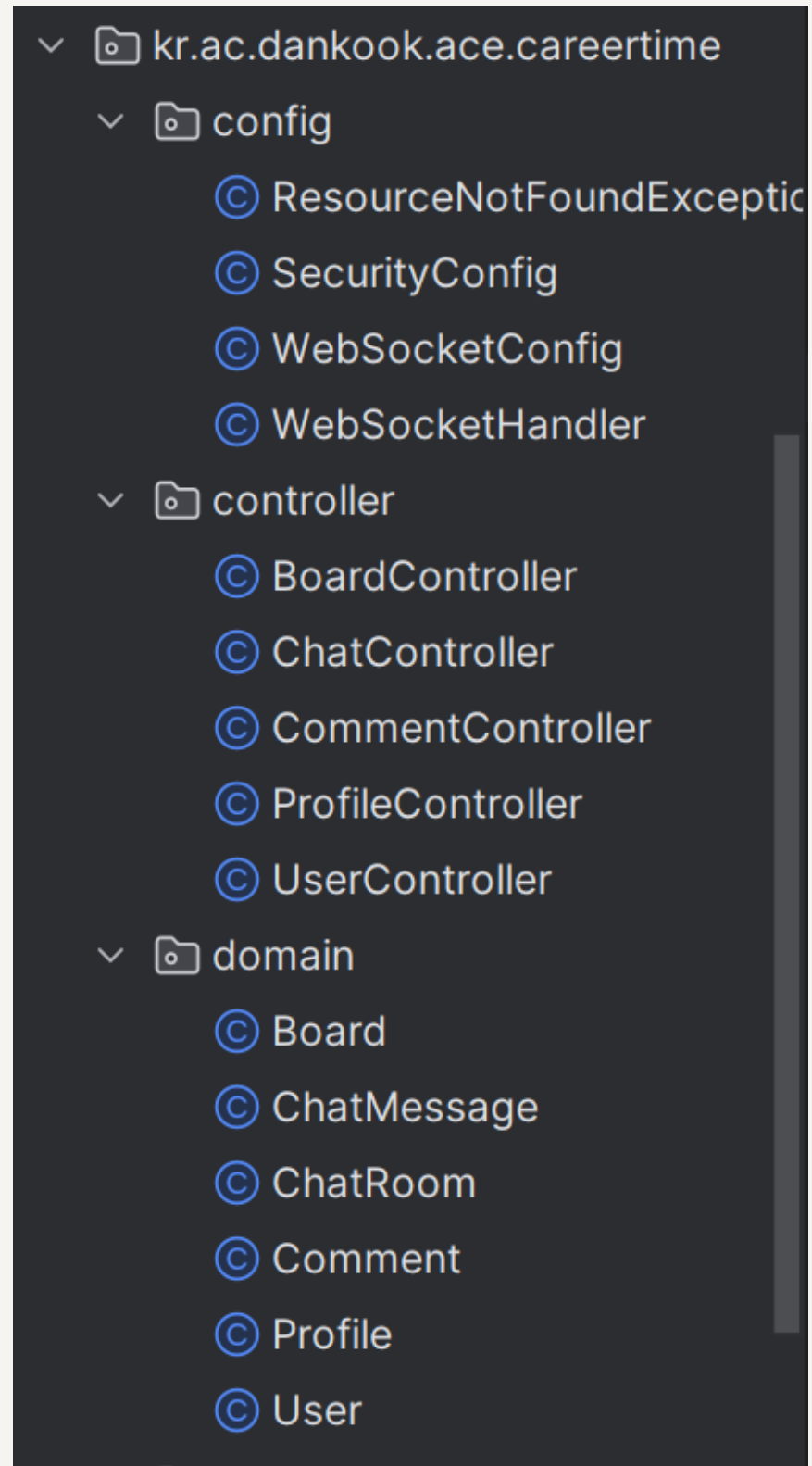
현재 구현 사항 - 마이페이지

05

추후 계획

06

현재 구현 사항 - 로그인, 회원가입



현재 구현 사항 - 로그인, 회원가입

```
@Entity
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "user_id", updatable = false)
    private Long user_id;

    @Column(name = "username", nullable = false, unique = true)
    private String username;

    @Column(name = "password", nullable = false)
    private String password;

    @Column(name = "name")
    private String name;

    @Column(name = "email")
    private String email;

    @Column(name = "user_type", nullable = false)
    private String user_type;

    @Column(name = "points")
    private Long points = 0L;

    @Column(name = "join_date", nullable = false)
    @JsonFormat(pattern = "yyyy-MM-dd HH:mm")
    private LocalDateTime join_date = LocalDateTime.now();
}
```

```
public interface UserRepository extends JpaRepository<User, Long>{
}
```

```
public class UserService {

    private final UserRepository userRepository;

    private final PasswordEncoder passwordEncoder;

    5 usages isasgp
    public User registerUser(String username, String password, String name, String email, String userType) {
        User newUser = new User();
        newUser.setUsername(username);
        newUser.setPassword(passwordEncoder.encode(password));
        newUser.setName(name);
        newUser.setEmail(email);
        newUser.setUser_type(userType);
        return userRepository.save(newUser);
    }

    5 usages isasgp
    public User findUserById(Long id) {
        Optional<User> user = userRepository.findById(id);
        return user.orElse( other: null);
    }
}
```

현재 구현 사항 - 로그인, 회원가입

```
@Configuration
@EnableWebSecurity
@RequiredArgsConstructor
public class SecurityConfig {

    no usages  isasgp

    @Bean
    public SecurityFilterChain filterChain(HttpSecurity http) throws Exception {
        http
            .csrf(AbstractHttpConfigurer::disable)
            .httpBasic(AbstractHttpConfigurer::disable)
            .formLogin(AbstractHttpConfigurer::disable)
            .authorizeHttpRequests((authorize) -> authorize
                .requestMatchers(...patterns: "/signup", "/", "/login", "/api/users/register").permitAll()
                .anyRequest().authenticated())
            .formLogin(formLogin -> formLogin
                .loginPage("/login")
                .defaultSuccessUrl("/home"))
            .logout((logout) -> logout
                .logoutSuccessUrl("/login")
                .invalidateHttpSession(true))
            .sessionManagement(session -> session
                .sessionCreationPolicy(SessionCreationPolicy.STATELESS)
            );
        return http.build();
    }
}
```

```
@RestController
@RequestMapping("/api/users")
@RequiredArgsConstructor
public class UserController {

    6 usages
    private UserService userService;

    no usages  isasgp *
    @PostMapping("/register")
    public ResponseEntity<User> registerUser(@RequestBody User user) {
        User registeredUser = userService.registerUser(user.getUsername(),
                                                    user.getPassword(),
                                                    user.getName(),
                                                    user.getEmail(),
                                                    user.getUser_type());
        return ResponseEntity.ok(registeredUser);
    }
}
```

Run UserServiceTest

✓	UserServiceTest (kr.a 1 sec 633 ms)	
✓	registerUser()	821 ms
✓	updateUser()	236 ms
✓	findUserById()	84 ms
✓	deleteUser()	492 ms

현재 구현 사항 - 유저 프로필

```
public class Profile {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "profile_id")
    private Long profile_id;

    @OneToOne
    @JoinColumn(name = "user_id", nullable = false)
    private User user;

    @Column(name = "company_name")    // 회사 이름 또는 취준
    private String company_name;

    @Column(name = "position")    // 현재 직종 또는 희망 직종
    private String position;

    @Column(name = "introduction")    // 마크 업 언어로 소개 문구 작성
    private String introduction;
}
```

```
public interface ProfileRepository extends JpaRepository<Profile, Long> {
}
```

```
@Service
@RequiredArgsConstructor
public class ProfileService {

    private final ProfileRepository profileRepository;

    2 usages  ⚙ isasgp
    public Profile createProfile(Profile profile) { return profileRepository.save(profile); }











    2 usages  ⚙ isasgp
    public List<Profile> findProfileByUserId(Long userId) {
        return profileRepository.findAll() List<Profile>
            .stream() Stream<Profile>
            .filter(p -> p.getUser().getUser_id().equals(userId))
            .collect(Collectors.toList());
    }
}
```

현재 구현 사항 - 유저 프로필

```
@RestController
@RequestMapping("/api/profiles")
@RequiredArgsConstructor
public class ProfileController {
    private final ProfileService profileService;

    no usages isasgp
    @PostMapping
    public ResponseEntity<Profile> createProfile(@RequestBody Profile profile) {
        Profile createdProfile = profileService.createProfile(profile);
        return new ResponseEntity<>(createdProfile, HttpStatus.CREATED);
    }

    // Get profiles by userId
    no usages isasgp
    @GetMapping("/user/{userId}")
    public ResponseEntity<List<Profile>> getProfilesByUserId(@PathVariable Long userId) {
        List<Profile> profiles = profileService.findProfileByUserId(userId);
        return new ResponseEntity<>(profiles, HttpStatus.OK);
    }
}
```

Run ProfileServiceTest	
         	
✓ ProfileServiceTest (kr.ac.d...	828 ms
✓ updateProfile()	672 ms
✓ deleteProfile()	20 ms
✓ createProfile()	11 ms
✓ findProfileByUserId()	125 ms

현재 구현 사항 - 게시판 및 댓글

```
public class Board {
    @Id // 기본 키임을 선언
    @GeneratedValue(strategy = GenerationType.IDENTITY) // 기본 키를 생성할 때
    private Long post_id;

    @ManyToOne
    @JoinColumn(name = "user_id", nullable = false)
    private User user;

    @Column(name = "title")
    private String title;

    @Column(name = "content")
    private String content;

    @Column(name = "hashtags")
    private String hashtags;

    @Column(name = "post_date") // 게시글의 작성일을 저장하는 컬럼
    @JsonFormat(pattern = "yyyy-MM-dd HH:mm")
    private LocalDateTime post_date;
}
```

```
public class Comment {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "comment_id")
    private Long comment_id;

    @ManyToOne
    @JoinColumn(name = "post_id", nullable = false)
    private Board board;

    @ManyToOne
    @JoinColumn(name = "user_id", nullable = false)
    private User user;

    @Column(name = "comment_text", nullable = false)
    private String comment_text;

    @Column(name = "comment_date", nullable = false)
    @JsonFormat(pattern = "yyyy-MM-dd'T'HH:mm:ss")
    private LocalDateTime comment_date = LocalDateTime.now();
}
```

Run BoardServiceTest		
BoardServiceTest (kr.ac.da 818 ms)		
✓	getBoardById()	637 ms
✓	deleteBoard()	25 ms
✓	findAllBoards()	125 ms
✓	updateBoard()	13 ms
✓	createBoard()	10 ms
✓	findBoardsByTitleAndHasht	8 ms

Run CommentServiceTest		
CommentServiceTest (kr.a 736 ms)		
✓	createComment()	578 ms
✓	findCommentsByBoardId	116 ms
✓	updateComment()	20 ms
✓	deleteComment()	22 ms

현재 구현 사항 - 게시물 등록페이지 / 상세페이지

제목

제목을 입력하세요

해시태그

#해시태그 #입력

본문 내용

B I U H L Link Image Video Table List Table of contents ?

상품 등록

A screenshot of a web browser window. The browser's address bar shows a URL starting with 'http://'. The page title is '본문 내용' (Main Content). The page content includes a title '소개' (Introduction) and a subtitle '자바 백엔드 개발자가 되는 길' (The Path to Becoming a Java Backend Developer). The main text reads: '안녕하세요 제 이름은 박민규 입니다. 현재 저는 000기업에 다니고 있으며 직무는 자바 스프링부트를 사용한 백엔드 개발입니다.' Below this is a list of requirements: '1. 자바 코딩 능력' and '2. 스프링부트 이해도'. A blue button labeled '상품 등록' (Register Product) is at the bottom left.

제목

자바 백엔드 개발 취업 도와드립니다.

해시태그

#해시태그 #입력

#자바 x

#백엔드 x

#스프링 x

본문 내용

B I

소개
안녕하세요 제 이름은 박민규 입니다.
현재 저는 000기업에 다니고 있으며 직무는 자바 스프링부트를 사용한 백엔드 개발입니다.

자바 백엔드 개발자가 되는 길
자바 백엔드 개발자가 되기 위해서는 많은 것들이 필요합니다.
1. 자바 코딩 능력
2. 스프링부트 이해도
3. 등등

상담을 통하여...
저와의 상담을 통해 여러분은 자바 백엔드 개발자가 되기 위한 꿀팁과 더불어

상품 등록

현재 구현 사항 - 게시물 등록페이지 / 상세페이지

커리어타임

[토스뱅크 현직자] 자바 백엔드 개발 진로상담 도와드립니다

#자바 #스프링 #백엔드

상세 설명

후기

반갑습니다. 저는 현재 토스뱅크에서 자바 백엔드 개발자로 근무하고 있습니다. 최근 취업 준비생들을 위한 진로 상담 경험을 공유하고자 이 글을 작성하게 되었습니다.

소개


- 이름: 홍길동
- 경력: 4년차 자바 백엔드 개발자
- 현재 직장: 토스뱅크
- 주요 업무: 결제 시스템, 금융 서비스 백엔드 개발

진로 상담 내용

1. 자바 백엔드 개발자가 되기 위한 길

자바 백엔드 개발자가 되기 위해서는 다음과 같은 기술 스택을 학습하는 것이 좋습니다.

- Java 프로그래밍 언어



Mingyu Park

단국대학교

parkm2ngyu00@naver.com

#스프링 #백엔드 #데이터베이스

프로필 보러가기

구매하기

[토스뱅크 현직자] 자바 백엔드 개발 진로상담 도와드립니다

#자바 #스프링 #백엔드

상세 설명

후기

상품 후기

총 3개의 후기


4 ★★★★★

후기 남기기

★★★★★

후기 내용을 작성해주세요.

제출하기



Mingyu Park

단국대학교

parkm2ngyu00@naver.com

#스프링 #백엔드 #데이터베이스

프로필 보러가기

구매하기

문제 해결을 위한 자바 활용

김철수 ★★★★★★

제품이 매우 좋았습니다!

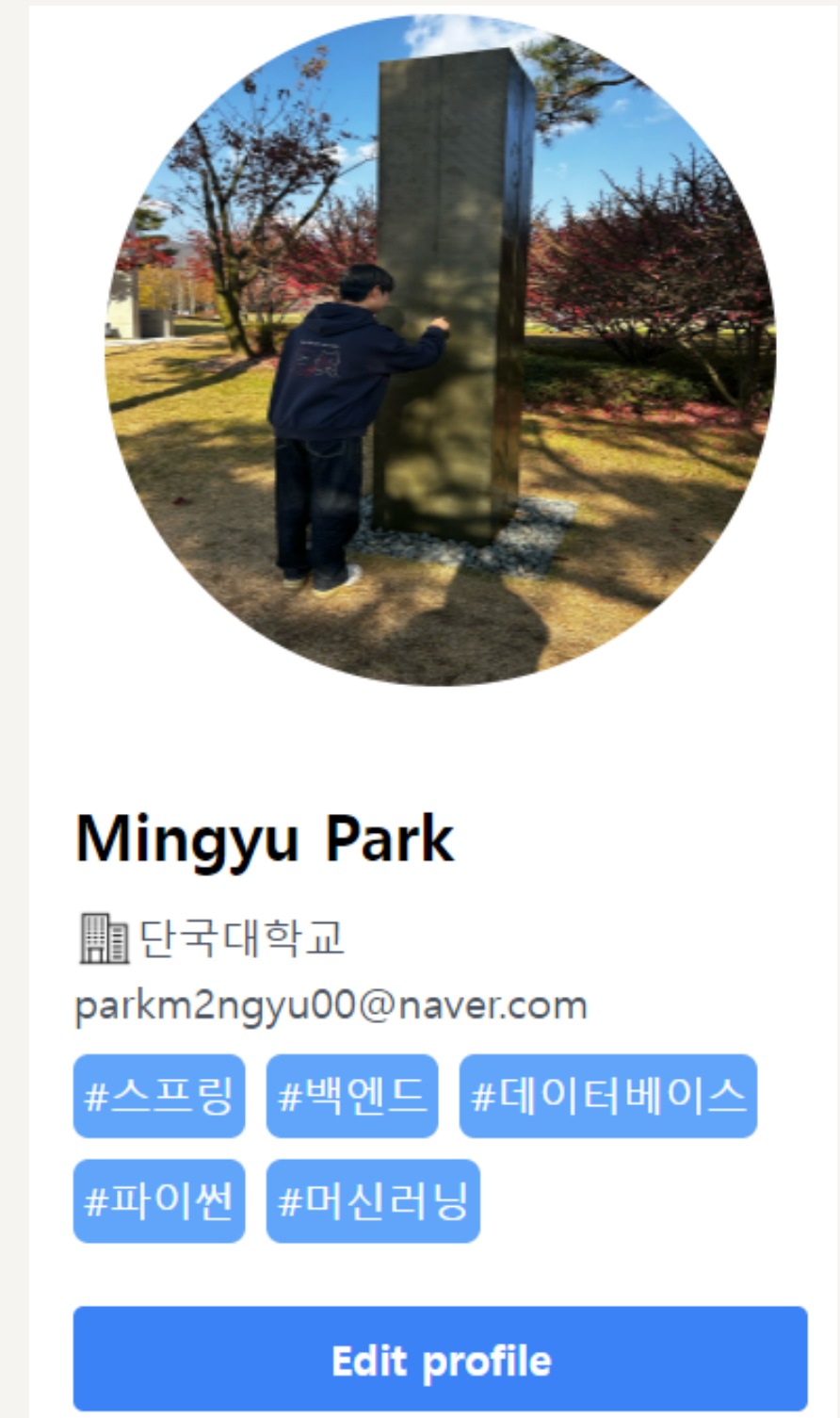
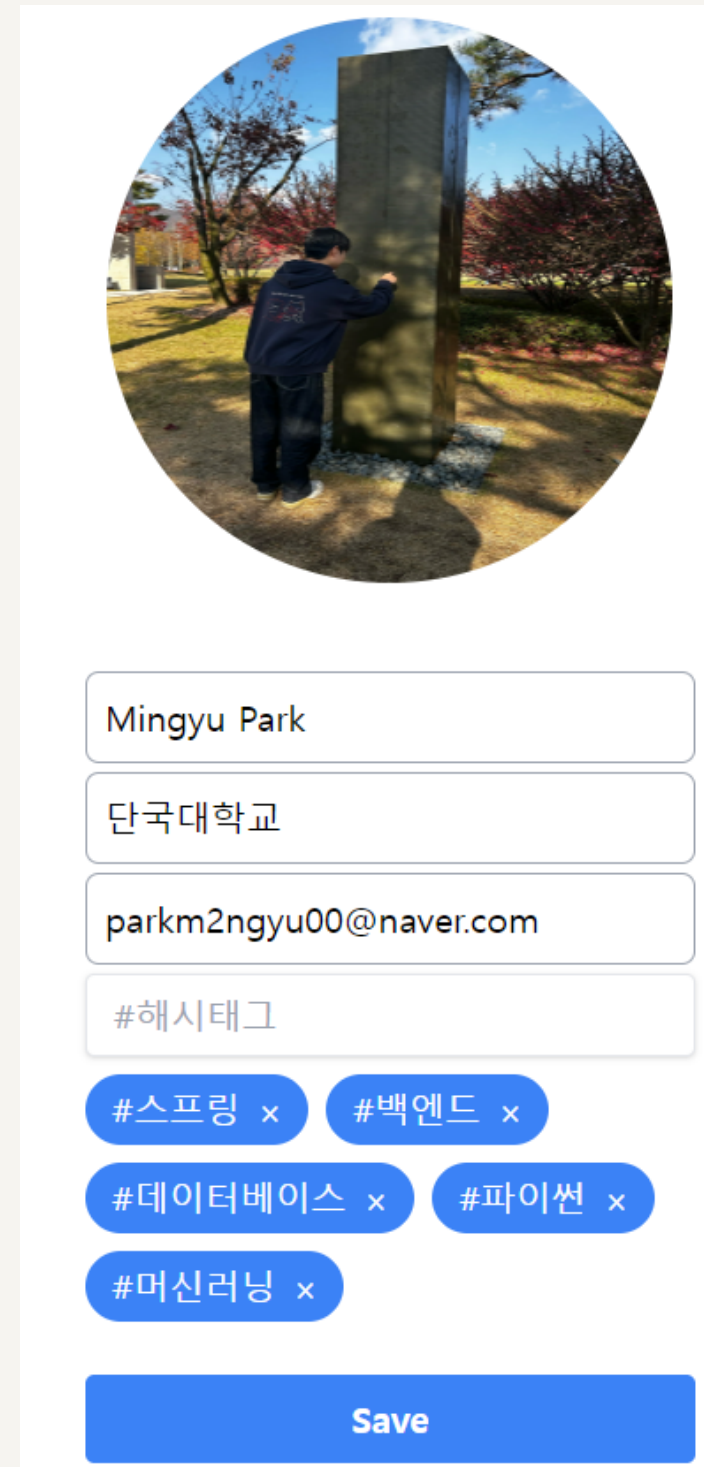
박영희 ★★★★★★

서비스도 친절하고 좋았어요.

이동해 ★★★★★★

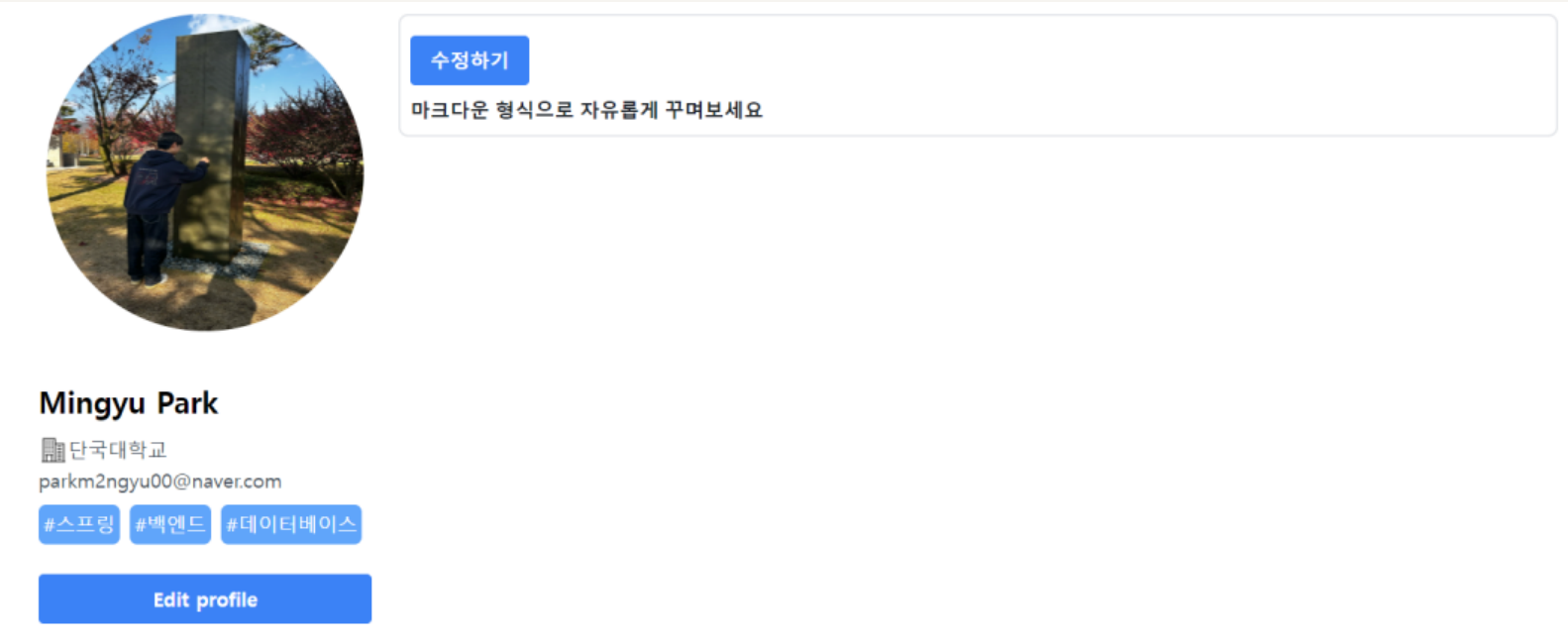
가격이 조금 높은 편이에요.

현재 구현 사항 - 마이페이지



현재 구현 사항 - 마이페이지

커리어타임



문제 해결을 위한 자바 활용



[2021]

- 2021.07 ~ 2021.08 🏆 한국영상자료원 ㄱ 영화 빅데이터 구축·활용 과제 발굴을 위한 아이디어 공모전 ㄱ **장려상**
- 2021.08 ~ 2021.11 🏆 제1회 경소톤(단국대학교) '세대차이 극복을 위한 소통 웹사이트' 프로젝트 **등상**

[2022]

- 2022.08 ~ 2023.01 📖 '개인 맞춤형 음식 추천 알고리즘에 대한 비교 평가 연구' 프로젝트 & 논문 작성 **한국정보통신학회지 등재**

[2023]

- 2022.12 ~ 2023.02 알고리즘 자동 채점 사이트 'CodeChalleng' 프로젝트
- 2023.03 ~ 2023.06 캡스톤디자인 프로젝트 '한대모여'
- 2023.06 ~ 2023.12 고용노동부 '미래내일 일경험 사업' 참가
- 2023.08 ~ 2023.10 '생성형 AI를 활용한 중소기업을 위한 ESG 컨설팅 보고서 자동생성' 프로젝트 with KoData Solution, BizData
- 2023.10 ~ 2023.12 '국립공원공단 대국민 탐방서비스' 프로젝트 with **국립공원공단**

추후 계획

- 게시물 검색 서비스(해시태그 기반, 제목 기반)
- 해시태그 기반 추천서비스
- 구매한 사용자와 게시물 작성자간의 채팅서비스

감사합니다.

