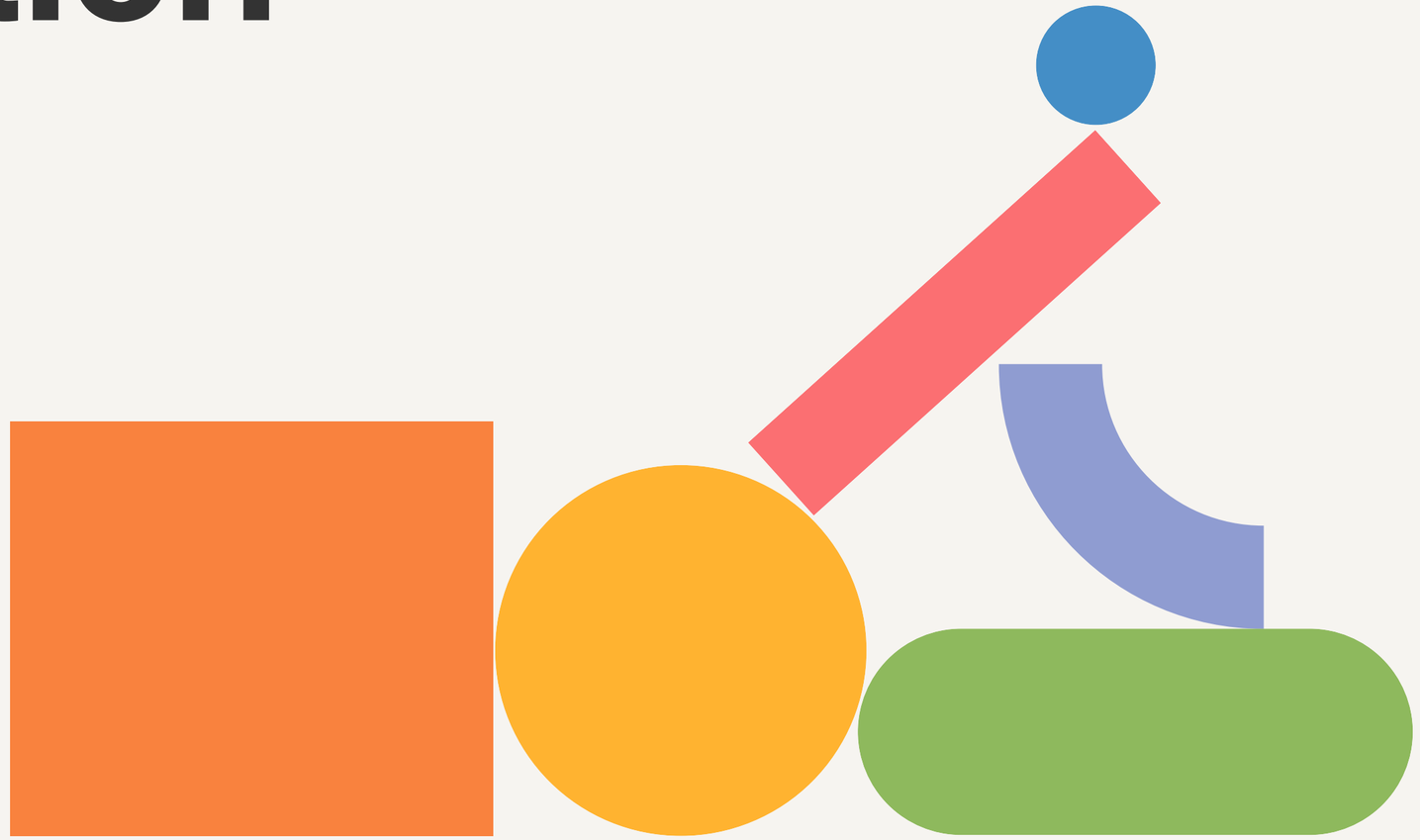
Project Midterm Presentation

32191597 박민규
32191585 박도훈
32222174 서영빈
32223436 이은서



문제해결을위한자바활용 2분반

목 차

문제해결을위한자바활용

프로젝트 주제

연봉

사내 분위기

복지

프로젝트 주제

JOBKOREA

Jobplanet

blind

기업 리뷰 작성해야 열람 가능(직장인만 가능)

기업 리뷰를 볼 순 있으나 제한적

멤버쉽 가입 등 유료

정작 알고싶은 기업의 정보는 없음

프로젝트 주제

네이버 expert 벤치마킹

지식iN expert

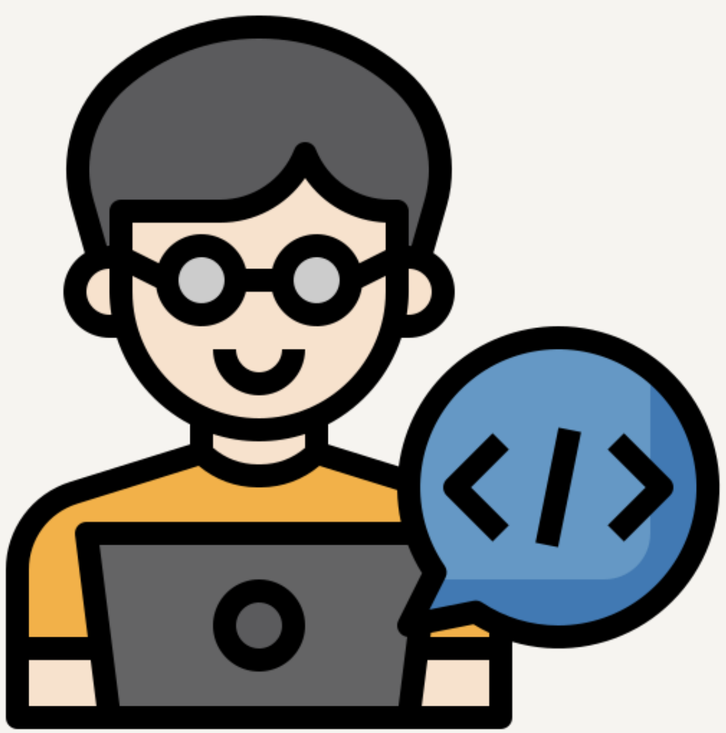
취업에 도움되는 서비스 개발

프로젝트 주제

상담을 받고자 하는 고객이 비용 지불

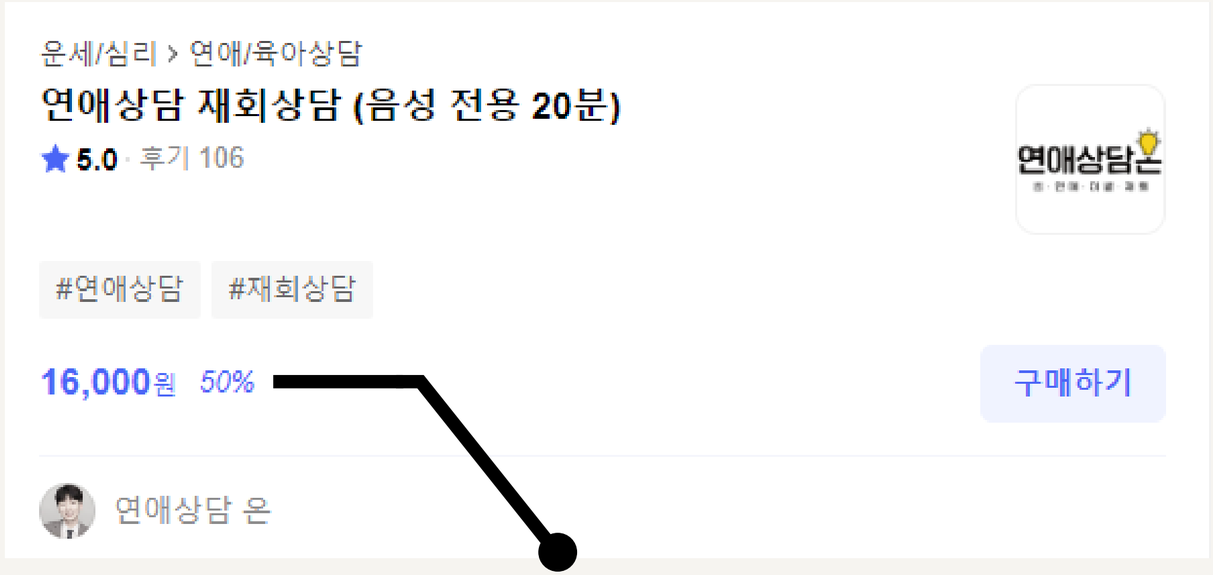
전문가와 1:1 채팅

프로젝트 주제



1:1 매칭

주요 기능



현직자는 상품 등록 가능

취준생은 마음에 드는 상품 선택

※ 웹 상에서 쓸 수 있는 포인트로 구현예정

주요 기능

1 : 1 채팅 기능

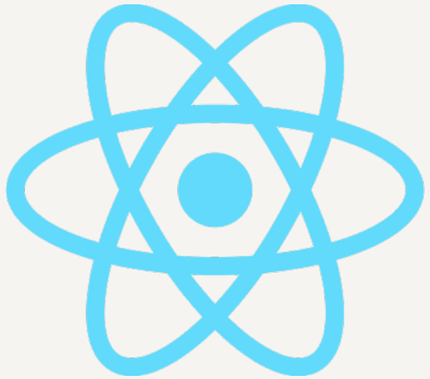
주요 기능

해시태그 기반 추천 기능

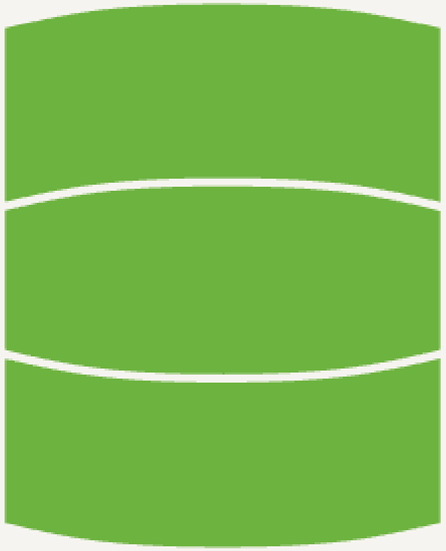
상품의 해시태그와 취준생 관심분야 비교

벡터화 후 코사인 유사도 기반

사용 기술



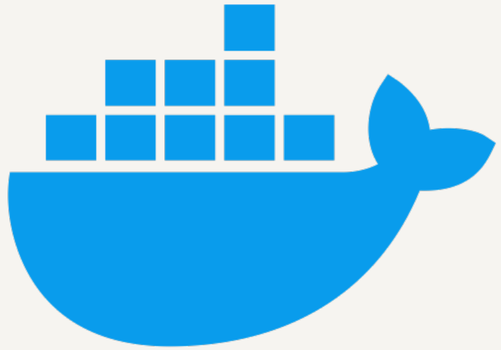
사용 기술



Spring Data JPA

Spring Security

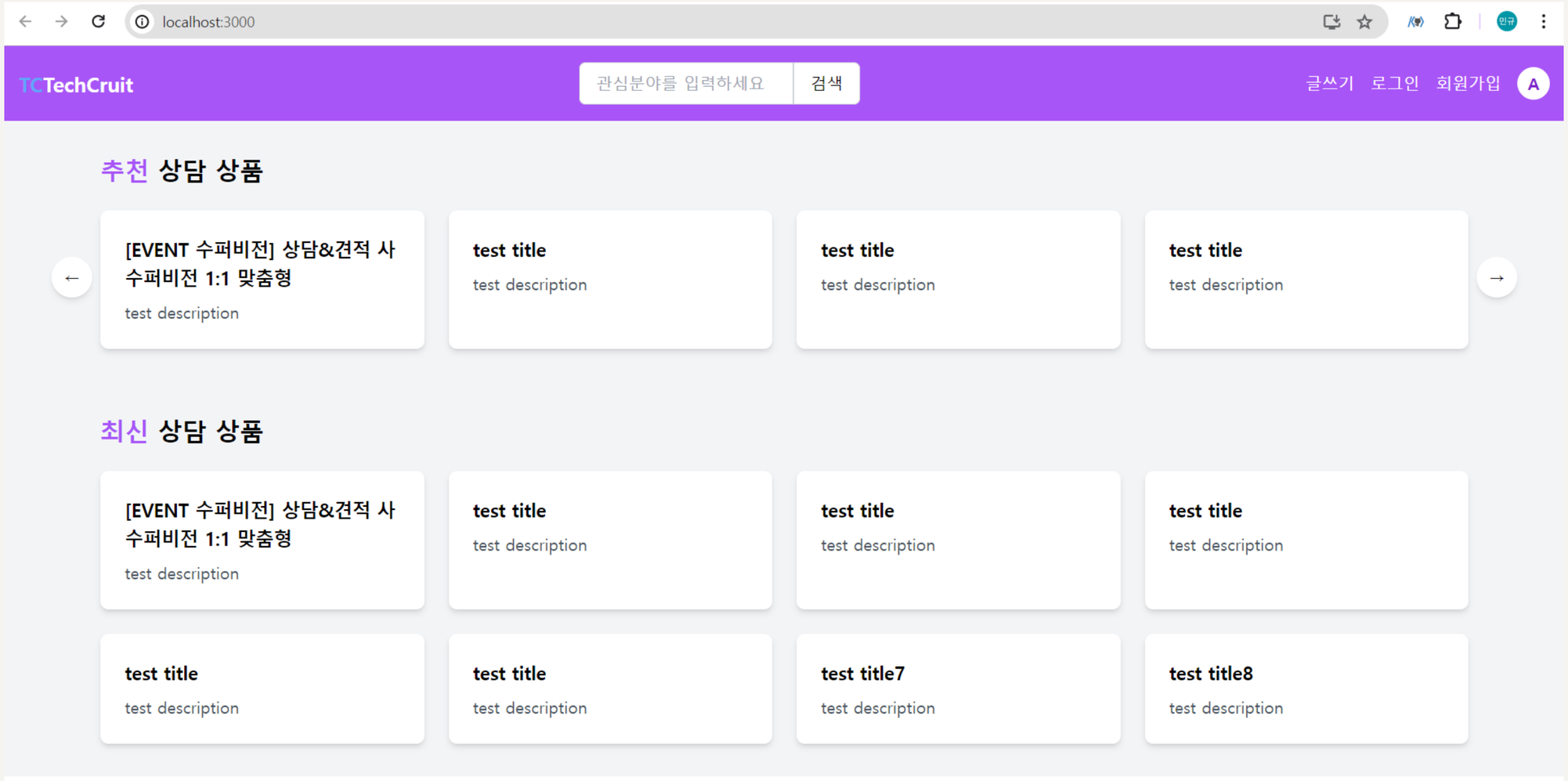
Kafka



Docker

Chapter 4

진행 상황 - UI

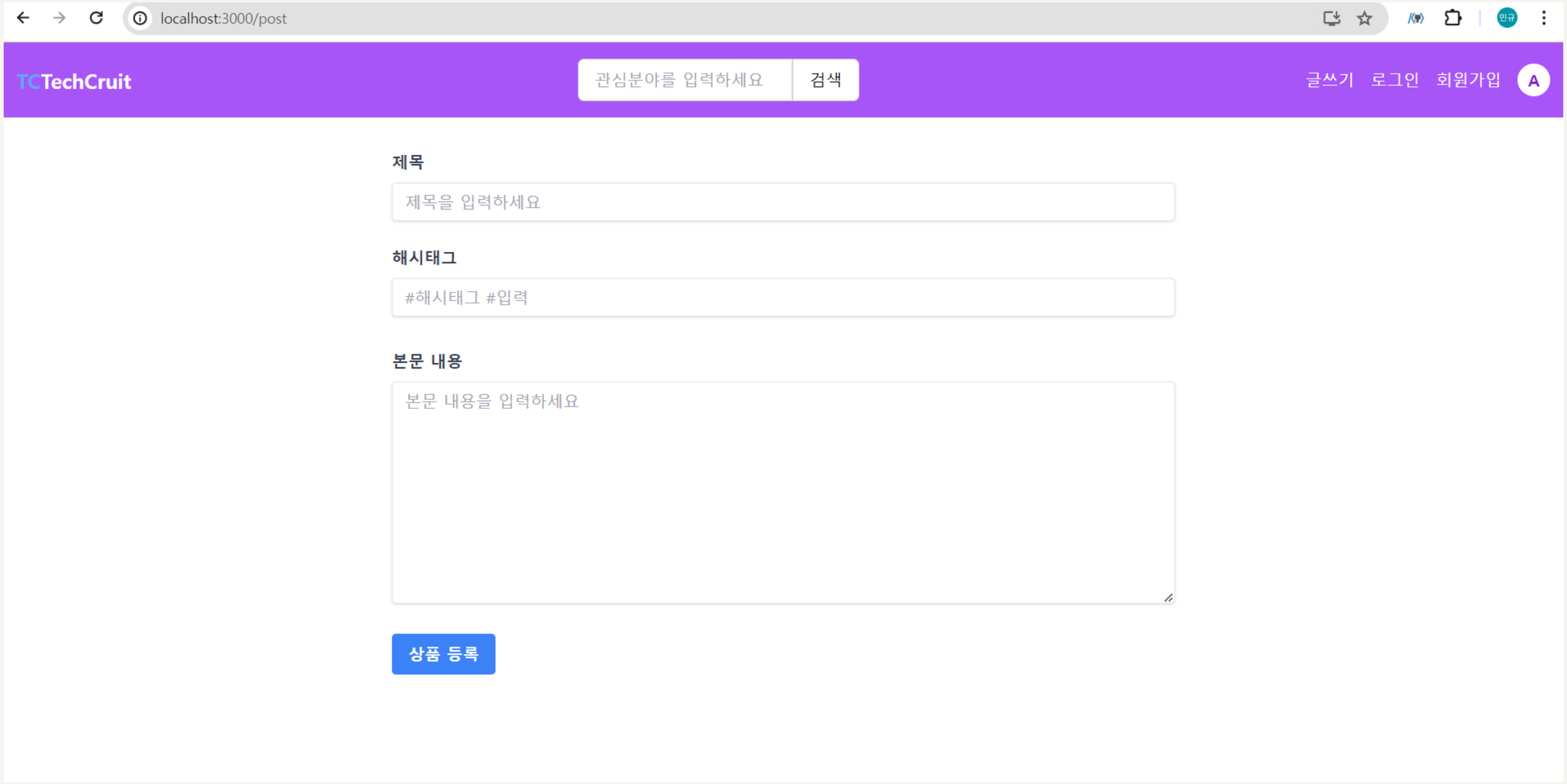


커리어타임

문제해결을위한자바활용

Chapter 4

진행 상황 - UI



진행 상황 - UI


TCTechCruit
관심분야를 입력하세요
검색
글쓰기
로그인
회원가입
A
제목
백엔드 개발자 취업 도와드립니다
해시태그
#해시태그 #입력
#백엔드 ×
#취업 ×
#스프링 ×
본문 내용
저는 현재 카X오에서 백엔드 개발자로 일하고 있는 ....
상품 등록
content: "저는 현재 카X오에서 백엔드 개발자로 일하고 있는
hashtags: Array(3)
0: "#백엔드"
1: "#취업"
2: "#스프링"
length: 3
[[Prototype]]: Array(0)
title: "백엔드 개발자 취업 도와드립니다"

진행 상황 - UI

```
return (
  <>
    <Header isExpert={isExpert}></Header>
    <div className="max-w-3xl mx-auto py-8">
      <div className="mb-6">
```

```
const Header = ({ isExpert }) => {
  return (
    <header className="bg-purple-500
      <nav className="container mx-aut
        <a href="/" className="text-
```

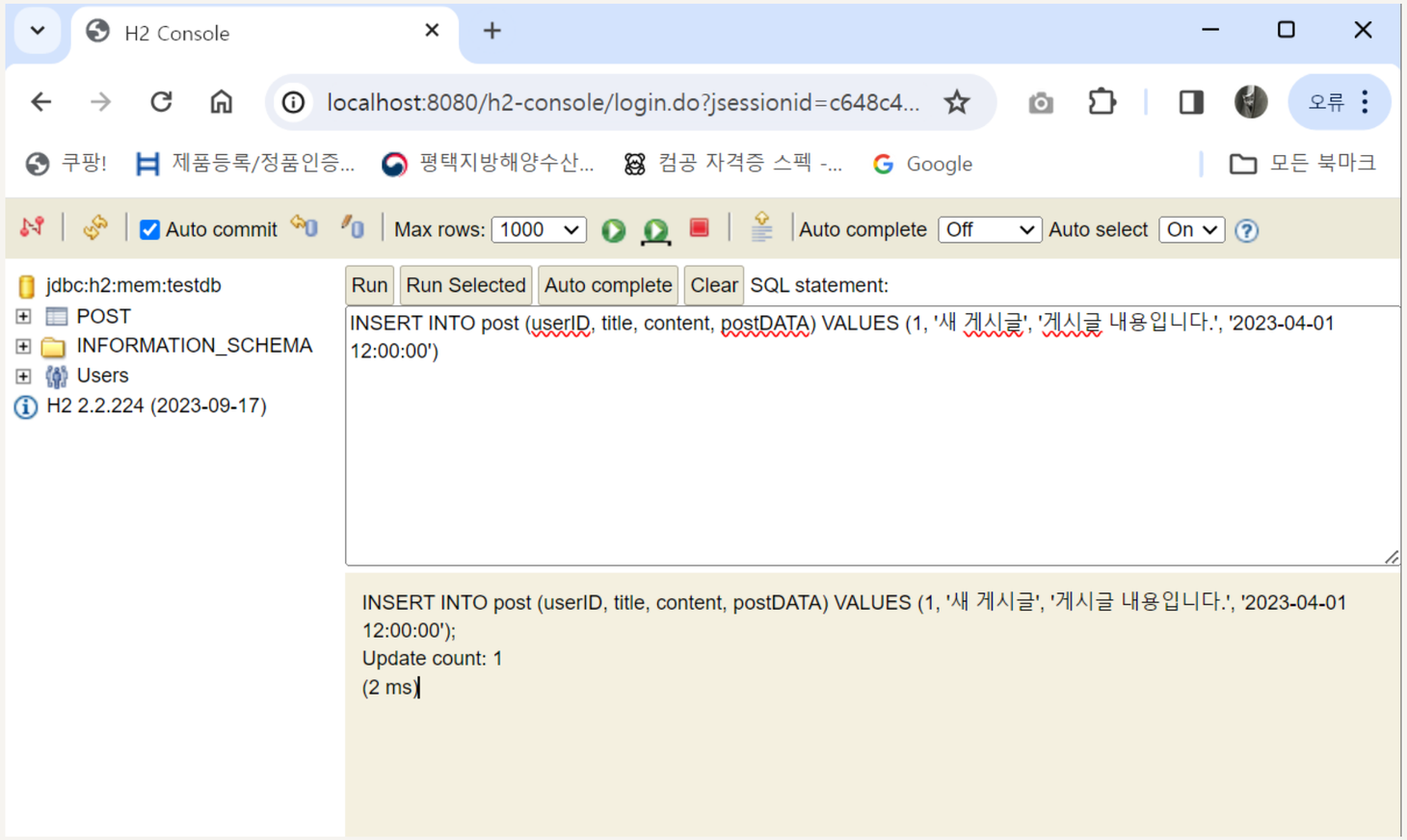
```
{isExpert ? (
  <Link
    className="text-white hover:t
    to="/post"
    state={{ isExpert: isExpert }}
  >
    글쓰기
  </Link>
) : (
  <></>
)}
```

로그인 회원가입 A

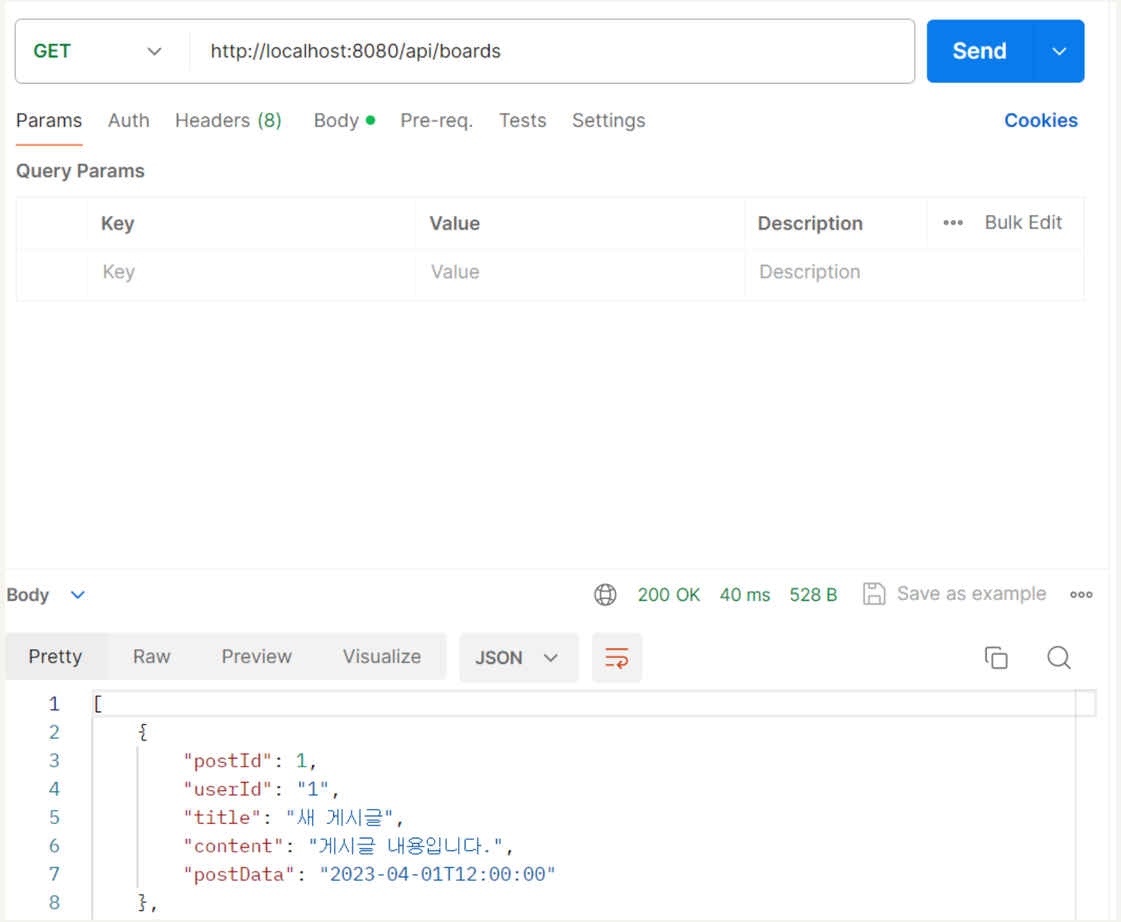
Expert로 로그인 한 경우에만 보임

진행 상황 - 상품 CRUD

H2 CONSOLE & POSTMAN 으로 테스트



진행 상황 - 상품 CRUD




```java
@GetMapping("/boards") // 모든 게시글 조회
public List<Board> findAllBoards() {
    return boardService.findAllBoards();
}
```

BoardController.java

```java
// 모든 게시글 조회
public List<Board> findAllBoards() {
    return boardRepository.findAll(); // 모든 게시글 조회
}
```

BoardService.java

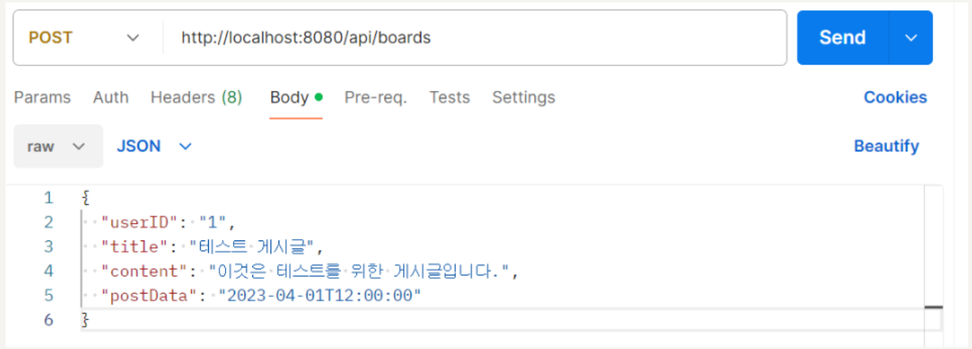
```java
@Repository // 리포지토리 계층을 선언, 스프링이 자동으로 구현체를 생성
public interface BoardRepository extends JpaRepository<Board, Integer> {
}
```

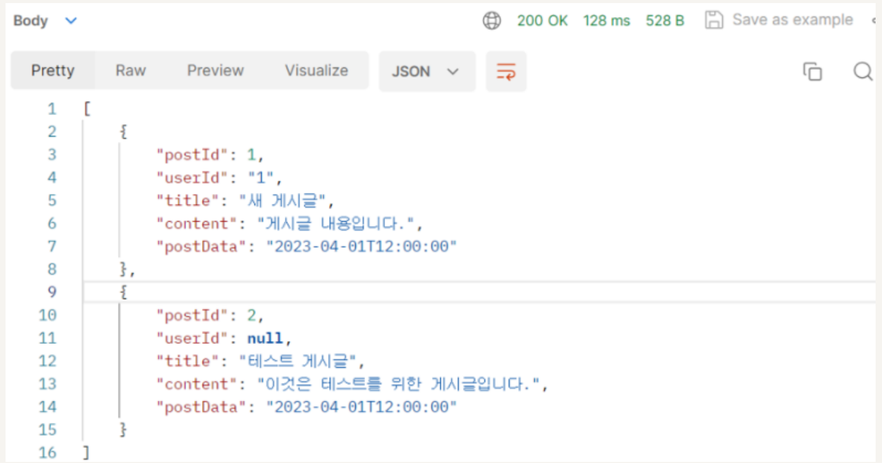
BoardRepository.java

커리어타임

문제해결을위한자바활용

진행 상황 - 상품 CRUD






```java
@PostMapping("/boards") // 게시글 생성
public Board createBoard(@RequestBody Board board) {
    return boardService.createBoard(board);
}
```

BoardController.java

```java
// 게시글 생성
public Board createBoard(Board board) {
    return boardRepository.save(board); // 게시글 저장
}
```

BoardService.java

```java
@Repository // 리포지토리 계층을 선언, 스프링이 자동으로 구현체를 생성
public interface BoardRepository extends JpaRepository<Board, Integer> {
}
```

BoardRepository.java

진행 상황 - 상품 CRUD

```java
@PutMapping("/boards/{id}") // 게시글 업데이트
public ResponseEntity<Board> updateBoard(@PathVariable Integer id, @RequestBody Board boardDetails) {
    return boardService.updateBoard(id, boardDetails);
}

@DeleteMapping("/boards/{id}") // 게시글 삭제
public ResponseEntity<Map<String, Boolean>> deleteBoard(@PathVariable Integer id) {
    return boardService.deleteBoard(id);
}
```

BoardController.java

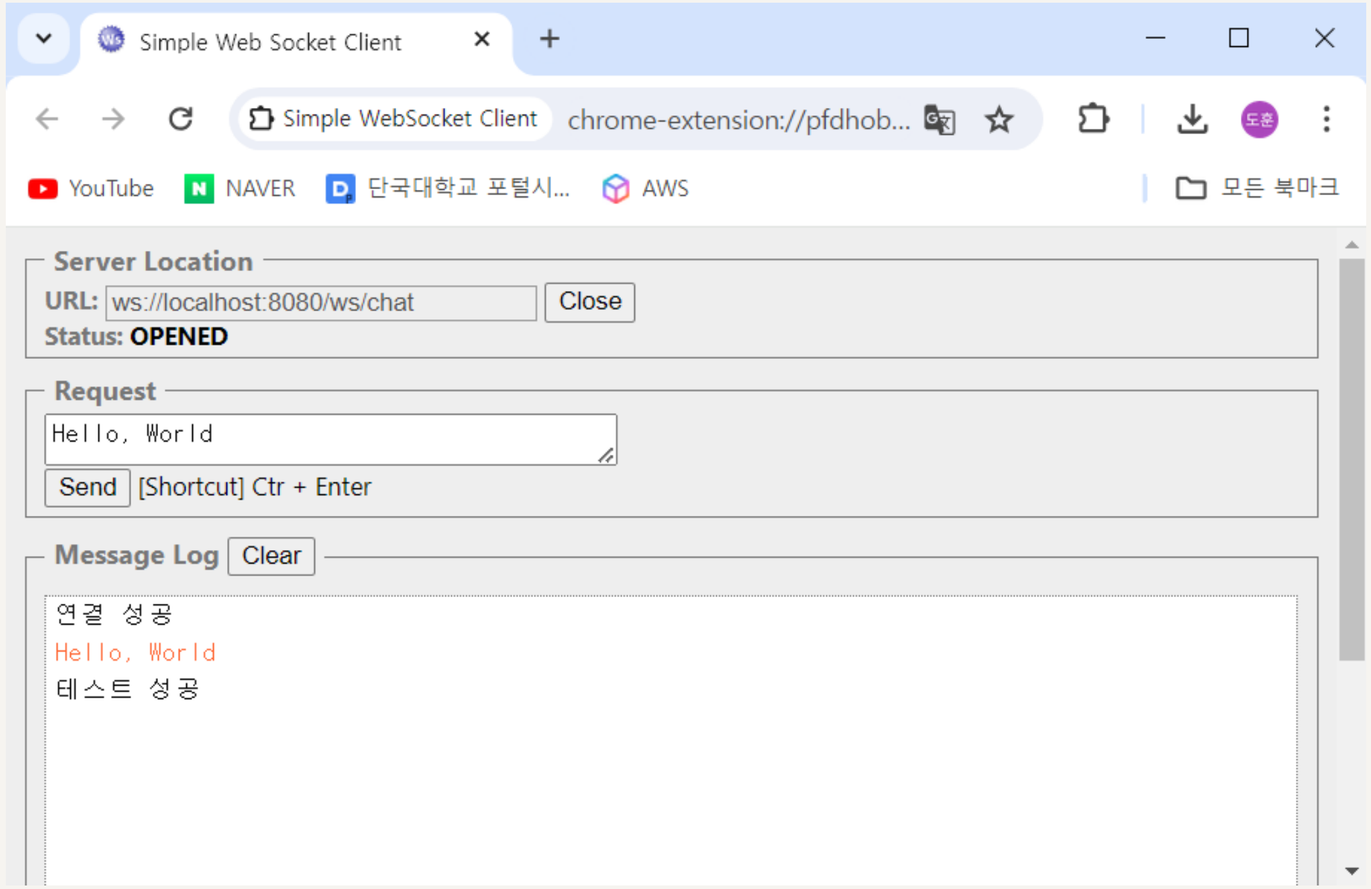
```java
// 게시글 업데이트
public ResponseEntity<Board> updateBoard(Integer id, Board boardDetails) {
    Board board = boardRepository.findById(id)
            .orElseThrow(() -> new ResourceNotFoundException("Board not exist with id :" + id));
    board.setTitle(boardDetails.getTitle());
    board.setContent(boardDetails.getContent());
    Board updatedBoard = boardRepository.save(board);
    return ResponseEntity.ok(updatedBoard);
}

// 게시글 삭제
public ResponseEntity<Map<String, Boolean>> deleteBoard(Integer id) {
    Board board = boardRepository.findById(id)
            .orElseThrow(() -> new ResourceNotFoundException("Board not exist with id :" + id));
    boardRepository.delete(board);
    Map<String, Boolean> response = new HashMap<>();
    response.put("deleted", Boolean.TRUE);
    return ResponseEntity.ok(response);
}
```

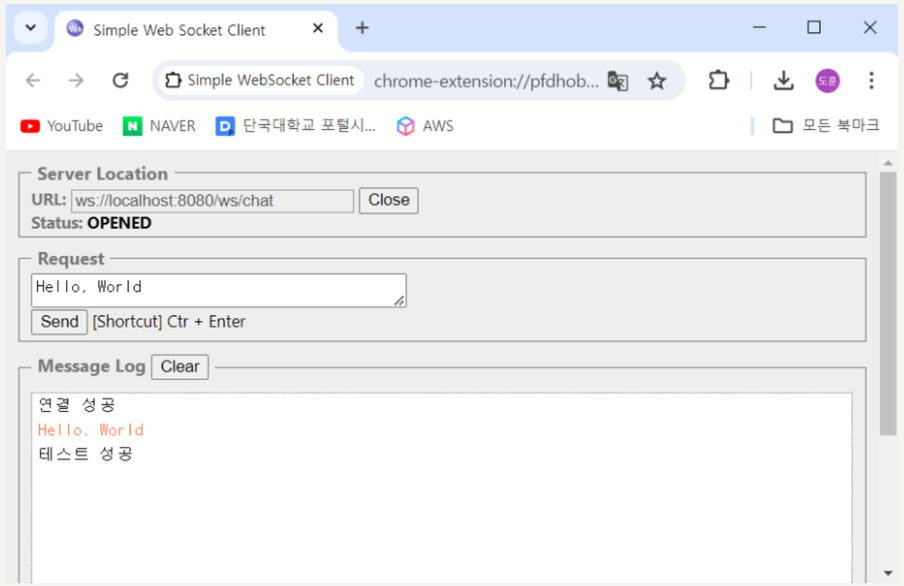
BoardService.java

진행 상황 - 1:1 채팅

simple websocket client 로 테스트



진행 상황 - 1:1 채팅



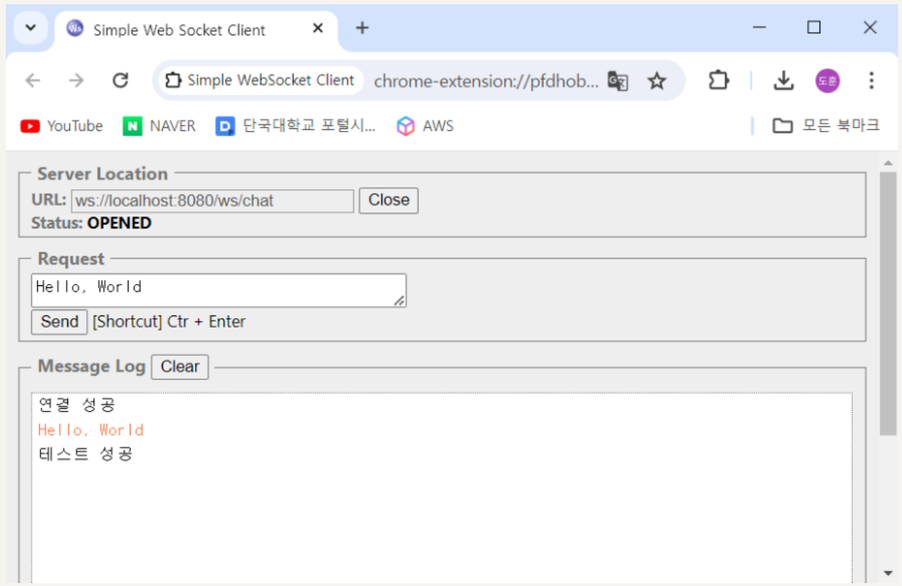

```java
@Override
public void handleTextMessage(WebSocketSession session, TextMessage message) throws Exception
    TextMessage successMessage = new TextMessage("테스트 성공");
    session.sendMessage(successMessage);
```

WebSocketHandler.java

```java
@Override
public void registerWebSocketHandlers(WebSocketHandlerRegistry registry) {
    registry.addHandler(customWebSocketHandler, "ws/chat").setAllowedOrigins("*");
}
```

WebSocketConfig.java

진행 상황 - 1:1 채팅




```java
@Component
public class Producer {     // 메세지를 전송
    private final KafkaTemplate<String, String> kafkaTemplate;
```

Producer.java

```java
@Component
public class Consumer {     // 메세지를 소비
    private KafkaConsumer<String, String> kafkaConsumer = null;
```

Consumer.java

```java
@Bean
public ProducerFactory<String, String> producerFactory() {  // kafka producer 설정
    Map<String, Object> configProps = new HashMap<>();
    configProps.put(ProducerConfig.BOOTSTRAP_SERVERS_CONFIG, bootstrapServer);
    configProps.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG, keySerializer);
    configProps.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG, valueSerializer);
    return new DefaultKafkaProducerFactory<>(configProps);
}
```

KafkaProducerConfig.java

감사합니다.