

# 주제 : whYmile

(정보공유에 초점을 둔 간단한 챗봇 시스템과 게시판 JSP)

---



---

# Contents

---

01

## Motivation for Selection

- 정보 공유
- 신의 탑
- 에밀리

02

## Benchmarking

- 심심이
- 에밀리
- 게시판

03

## Concept

- 간접적인 정보 공유
- 직접적인 정보 공유

04

## System Screen

- 메인 페이지
- whYmile
- 게시판

05

## UI Navigation

- 페이지 흐름

06

## ER-Diagram

- 테이블
- MySQL

07

## Result

08

## ※ Setting※

- 페이지 경로 설정
  - 업로드 파일 설정
-

---

# **Motivation for Selection**

---

# Motivation for Selection



## 선정동기

해당 웹툰 내용 중 중학생 때(2012년)부터 인상 깊게 받아들이고 생각한 부분이 있다.

그것이 바로 “에밀리”라는 것인데, 웹툰 내용 중에서 “에밀리”는 흔히 우리가 아는 심심이, 이루다, 시리, 빅스비 와 같은 인공지능, 챗봇 프로그램이다.

웹툰에서 “에밀리”는 정보를 공유할 수 있는 프로그램으로 초점을 두고 다뤄진다.

허황된 정보를 공유할 수도 있지만 때로는 그런 정보를 믿고 따르거나 정확한 정보를 얻을 수도 있는 이런 프로그램에 감명을 받아 한번 사용해 보고 싶다는 생각을 한 바이다.

---

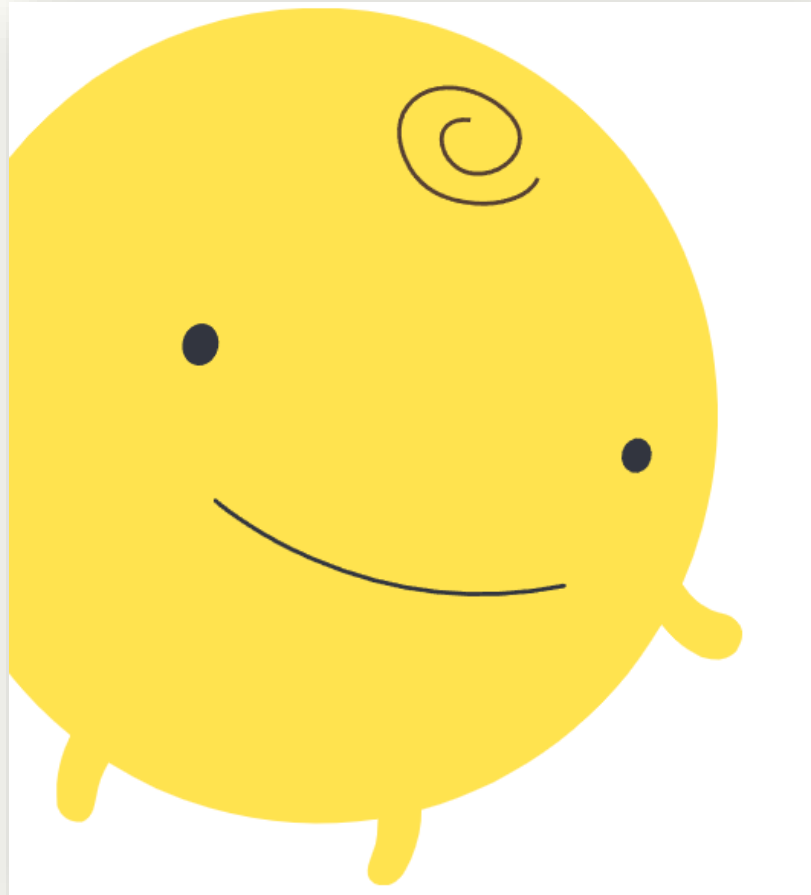
# Benchmarking

---

---

# Benchmarking

---



## 벤치마킹

이번 프로젝트에서 벤치마킹할 사이트는 심심이와 신의 탑(네이버 웹툰)내용이다.

심심이(<https://www.simsimi.com>)의 대화식 질문/답변 시스템과 웹툰 내용에서 나오는 에밀리의 질문/답변 시스템을 참고하여 프로젝트를 진행할 예정이다.

여기서 사용자끼리 직접 정보를 공유할 수 있는 부분을 추가하여 차별점을 두고자 한다.

---

**Concept**

---

# Concept



## 컨셉트

이번 JSP프로젝트는 짧은 기간 동안 진행하는 프로젝트로 적당한 규모의 프로젝트를 설계하는 것이 중요하다.

따라서 이번 프로젝트인 나의 웹 시스템 개발은 에밀리와 같은 “정보공유”에 초점을 둔 시스템이다.

머신 러닝이나 딥 러닝과 같은 자가 학습 시스템은 도입하지 않고, 사용자들끼리 질문에 대한 답변을 하여 데이터 베이스에 저장하는 시스템이다.



---

# Concept

---

## 컨셉트

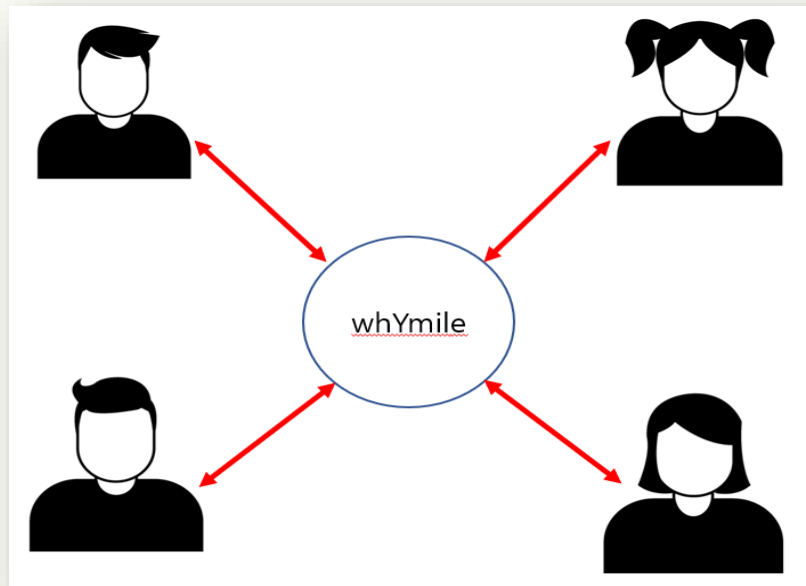
학생 본인의 이름인 김연욱 가운데 글자인 연에서 Y를 뺀 Ymile -> whYmile 라는 시스템 이름이다.

사용자가 와이밀리에게 질문하면 와이밀리는 해당 질문에 대한 키워드와 관련된 정보를 데이터 베이스에서 랜덤하게 찾아 답변해주는 시스템이다.

만약 질문에 대한 키워드와 관련된 정보가 없다면 해당 질문은 와이밀리가 저장하고, 사용자들에게 반대로 질문하여 답변을 얻고 그 답변을 저장하는 시스템을 구축할 것이다.

연속적인 대화의 형태가 아닌 한 번의 질문과 답변으로 이루어지는 간단한 챗봇 시스템이다.

---



---

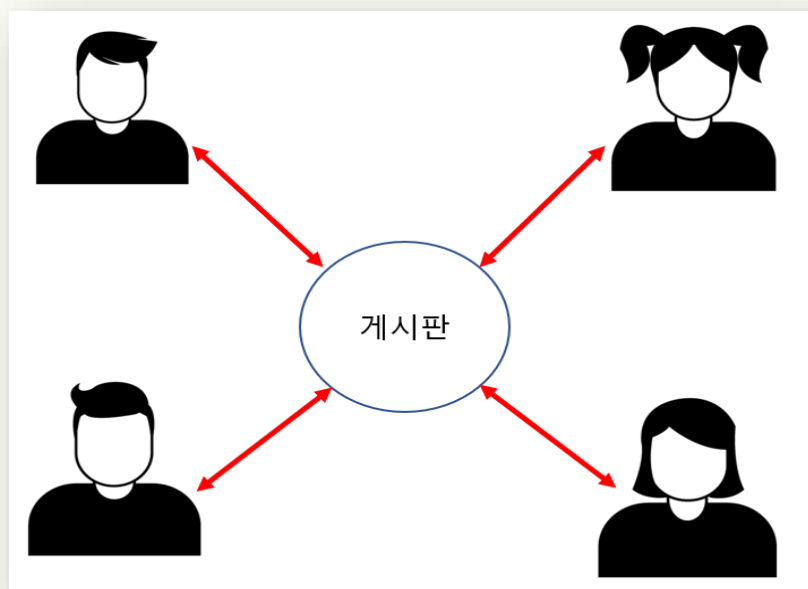
# Concept

---

## 컨셉트

추가적으로 게시판 기능을 넣어 사용자들이  
와이밀리를 거치지 않고도 정보를 공유할 수 있는  
시스템 또한 추가하려 한다.

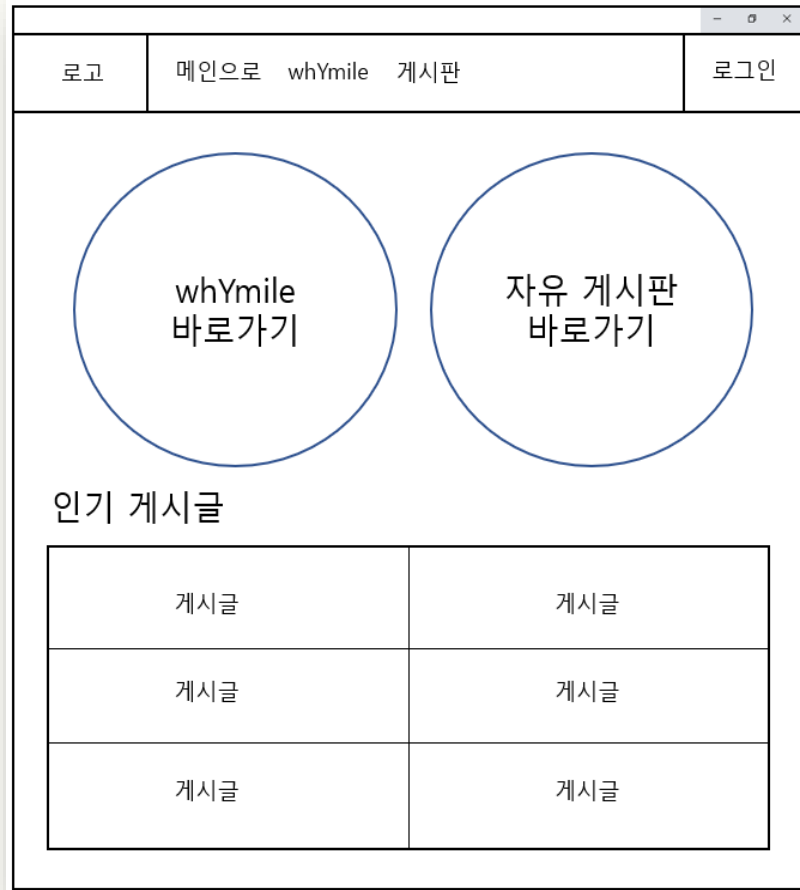
이때 이미지도 같이 공유할 수 있다.



---

# System Screen

# System Screen



## 메인 페이지

- 회원 로그인 후 whYmile를 이용하러 가기를 통해 와이밀리를 사용할 수 있다.
- 상단 메뉴바의 로그인 부분은 비로그인 시 회원가입과 로그인 기능을 로그인 시 회원정보수정과 로그아웃 기능을 넣을 것이다.

# System Screen

The screenshot shows a web browser window with the title bar containing minus, maximize, and close buttons. The browser's address bar shows the URL '메인으로 whYmile 게시판' and a '로그인' button on the right. The main content area is a chat interface with a large text box at the top containing the prompt '질문을 해주세요.' and a '로그인' button. Below this, a series of buttons represent the chatbot's responses: '오늘 뭐 먹지?', '갈비탕', '치킨 추천해 줘', '뿌링클', '고마워요.', and '대화종료'. At the bottom, there is a text input field labeled '질문할 내용 입력' and a '질문하기' button.

로그	메인으로 whYmile 게시판	로그인
----	------------------	-----

질문을 해주세요.

로그인

오늘 뭐 먹지?

갈비탕

치킨 추천해 줘

뿌링클

고마워요.

대화종료

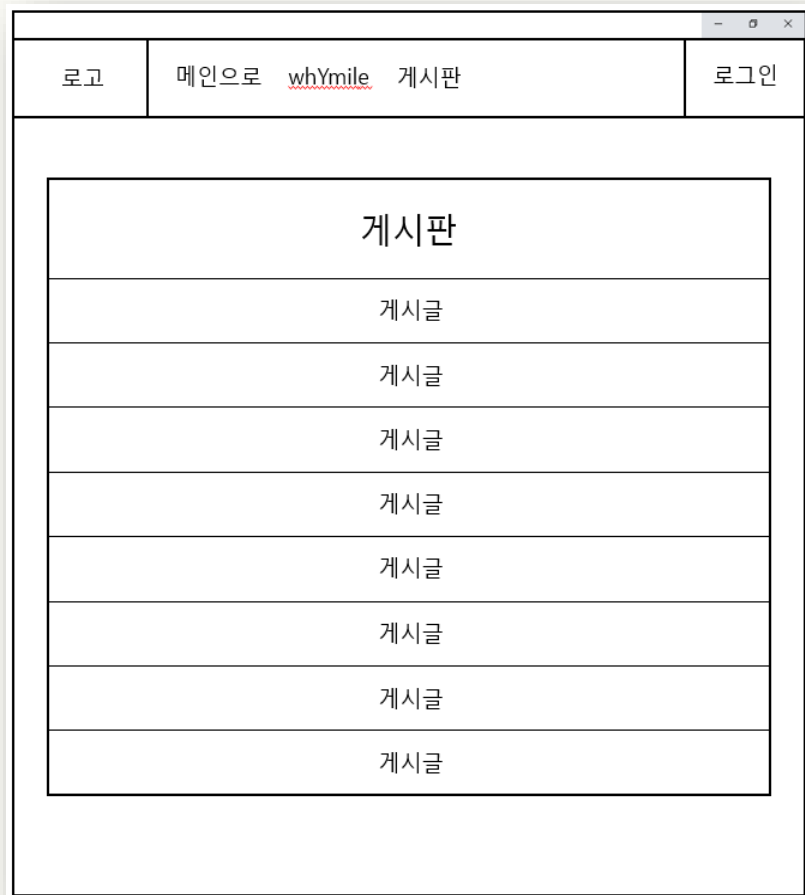
질문할 내용 입력

질문하기

## whYmile

- 로그인 후 이용할 수 있는 챗봇 기능이다.
- 먼저 사용자가 질문을 하며, 해당 질문에 대한 답변을 와이밀리가 한다.
- 그 다음 와이밀리가 질문을 하고 사용자가 답변을 하는 한 사이클의 시스템이다.
- 사이클이 한 번 돌면 챗봇 대화를 종료한다.

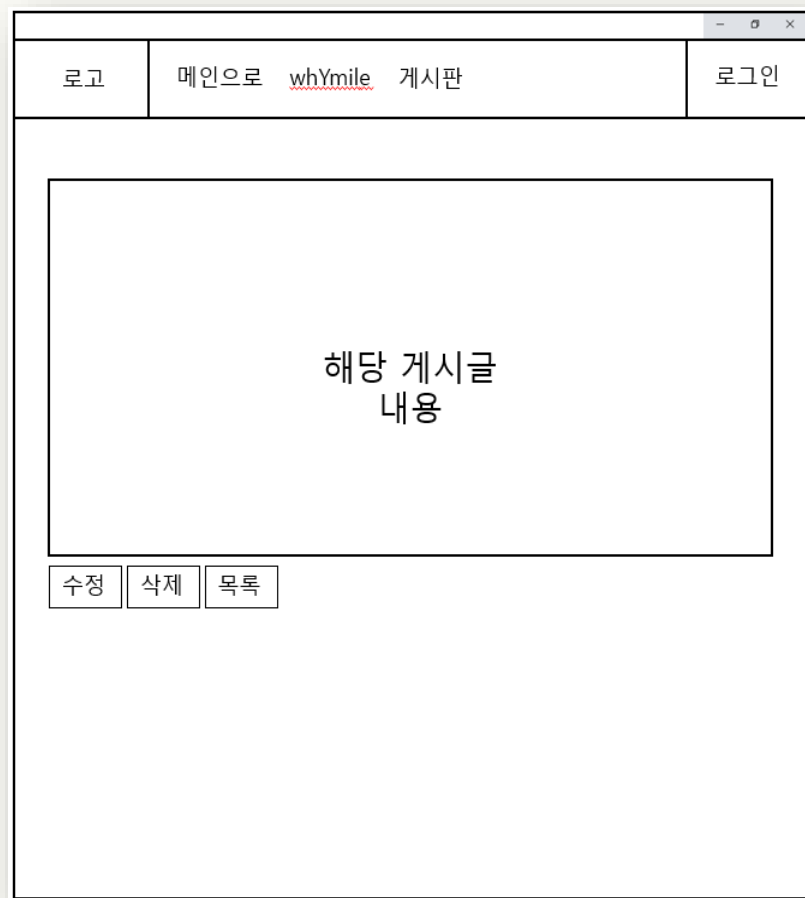
# System Screen



## 게시판

- 글쓰기를 통하여 게시글을 등록할 수 있다.
- 글 보는 것은 비로그인 유저도 볼 수 있게 하며,
- 글쓰기 기능은 로그인 유저만 사용할 수 있도록 한다.

# System Screen



## 게시글 내용

- 게시글을 보는 기능은 비로그인 유저도 가능하며, 이 때 자기 자신이 쓴 글이라면 수정과 삭제를 가능하게 한다.
- 그리고 해당 게시글 조회수 기능을 넣어 몇 명의 사용자가 봤는지 알 수 있도록 한다.

# System Screen

로고    메인으로 whYmile    게시판    로그인

### 회원가입 화면

아이디

비밀번호

비밀번호 확인

이름

이메일

남자    여자

회원가입

## 회원가입 화면

- 회원가입 화면이다.
- 이 기능을 통해 유저 아이디를 생성할 수 있다.



# System Screen

로그인 화면

아이디

비밀번호

로그인

## 로그인 화면

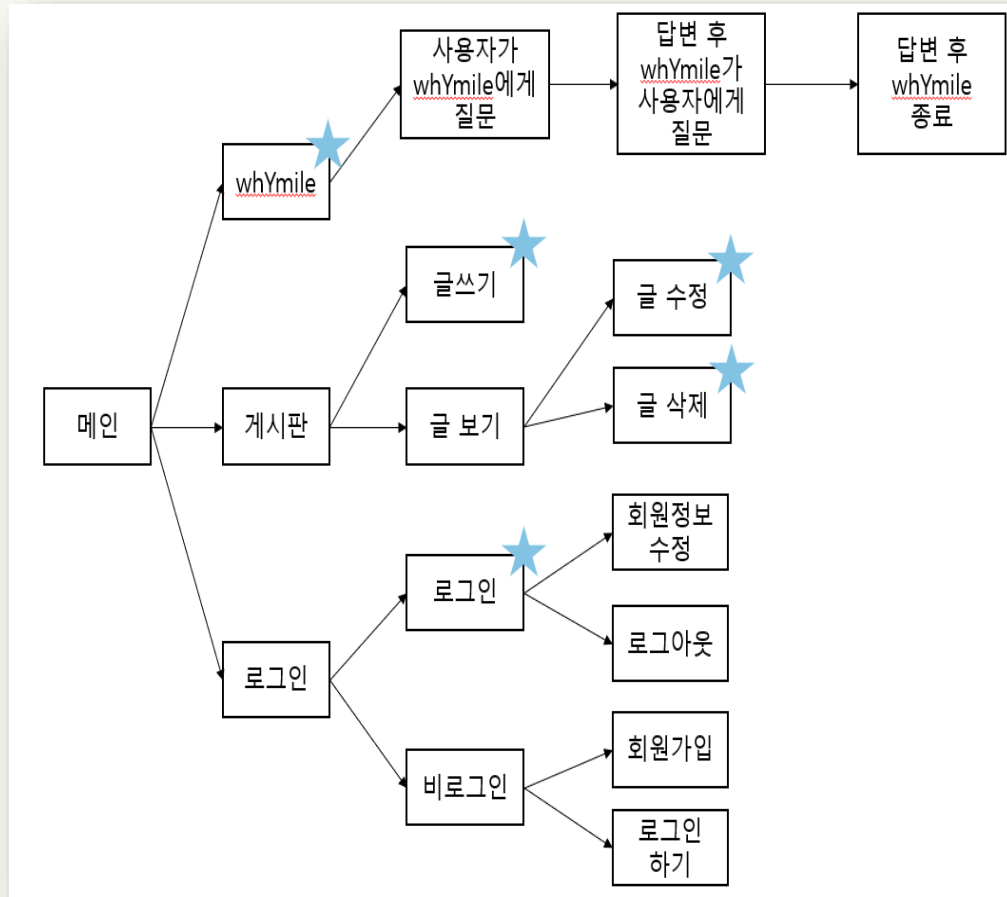
- 로그인 화면이다.
- 회원가입된 아이디와 비밀번호로 로그인을 할 수 있다.

---

# UI Navigation

---

# UI Navigation



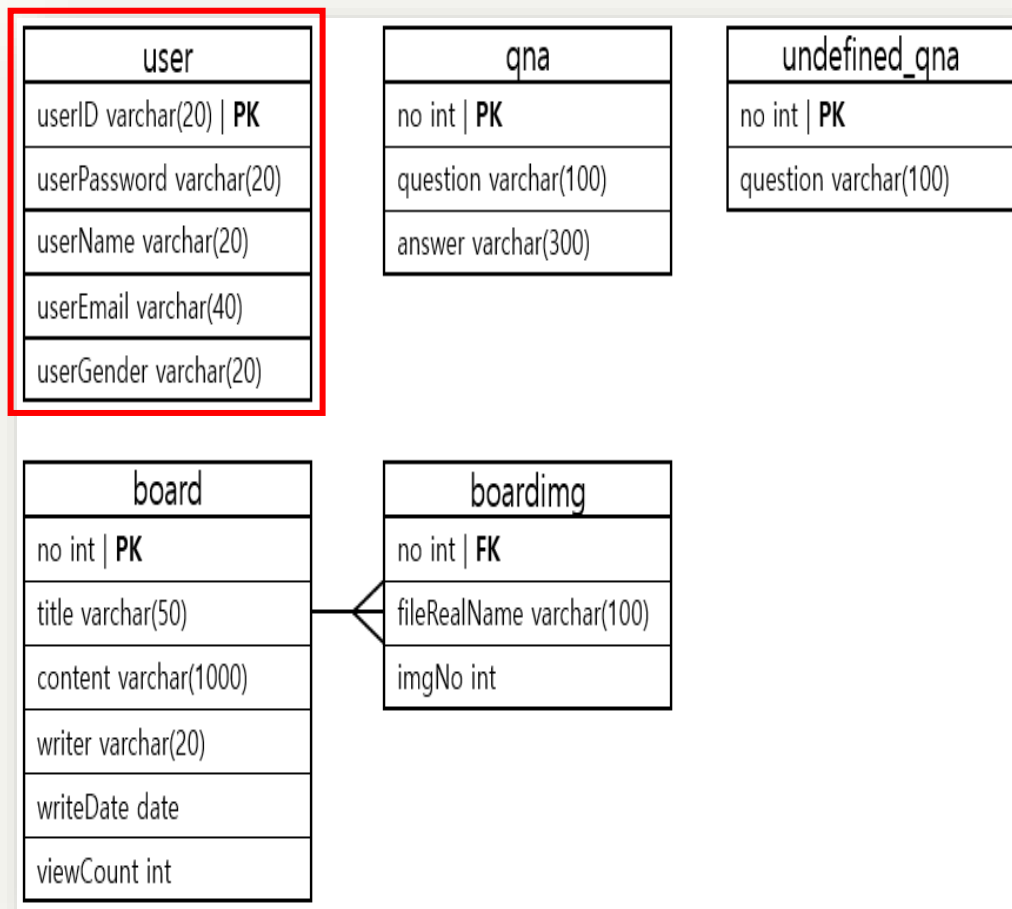
## 페이지 흐름

- 크게 세 가지로 분류 후 좀 더 세분화 한 페이지 흐름이다.
- 하늘색 별표는 로그인이 필요한 기능이다.
- 하늘색 별표 뒤로 연결된 기능들은 모두 로그인 세션이 필요한 기능이다.
- 각 기능들은 Form과 Action 즉 양식과 처리로 나누어 기능들을 보여주고 처리하게 된다.

---

# ER-Diagram

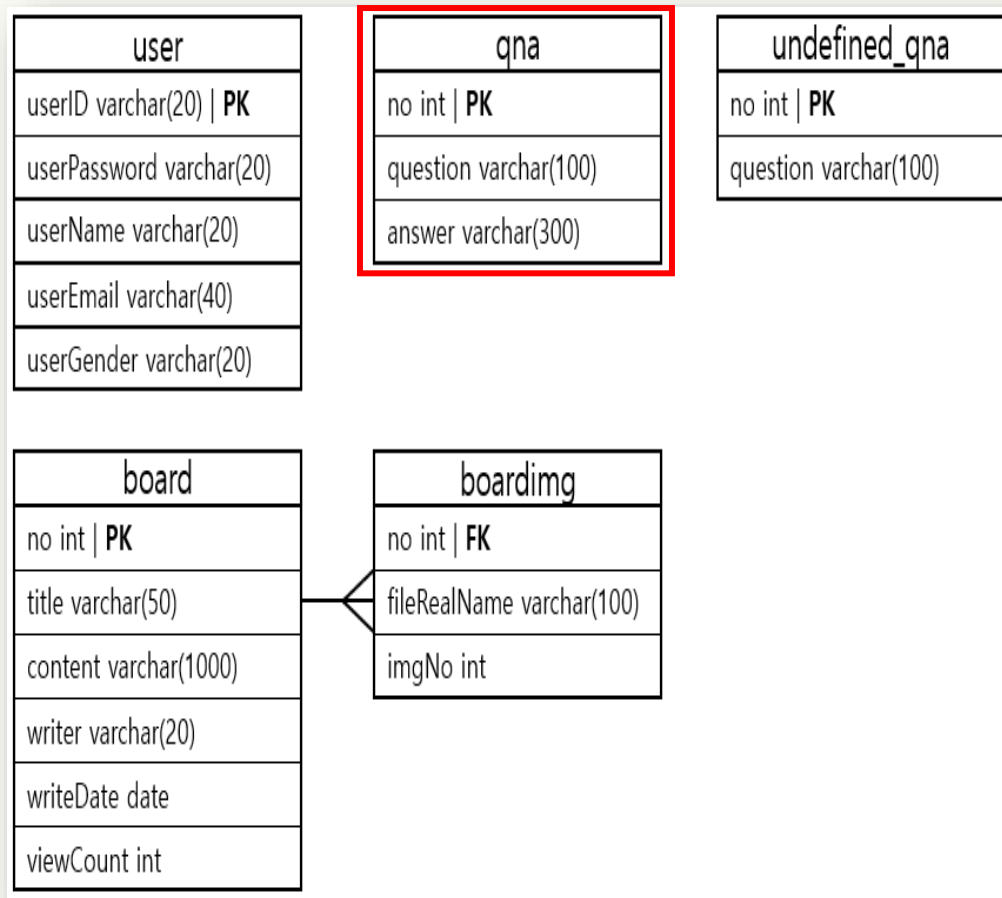
# ER-Diagram



## user 테이블

- userID 필드는 Primary Key로 아이디를 저장하는 가변길이의 문자열 타입이다.
- userPassword 필드는 비밀번호를 저장하는 가변길이의 문자열 필드 타입이다.
- userName 필드는 사용자 이름을 저장하는 가변길이의 문자열 필드 타입이다.
- userEmail 필드는 이메일을 저장하는 가변길이의 문자열 필드 타입이다.
- userGender 필드는 성별을 저장하는 가변길이의 문자열 필드 타입이다.

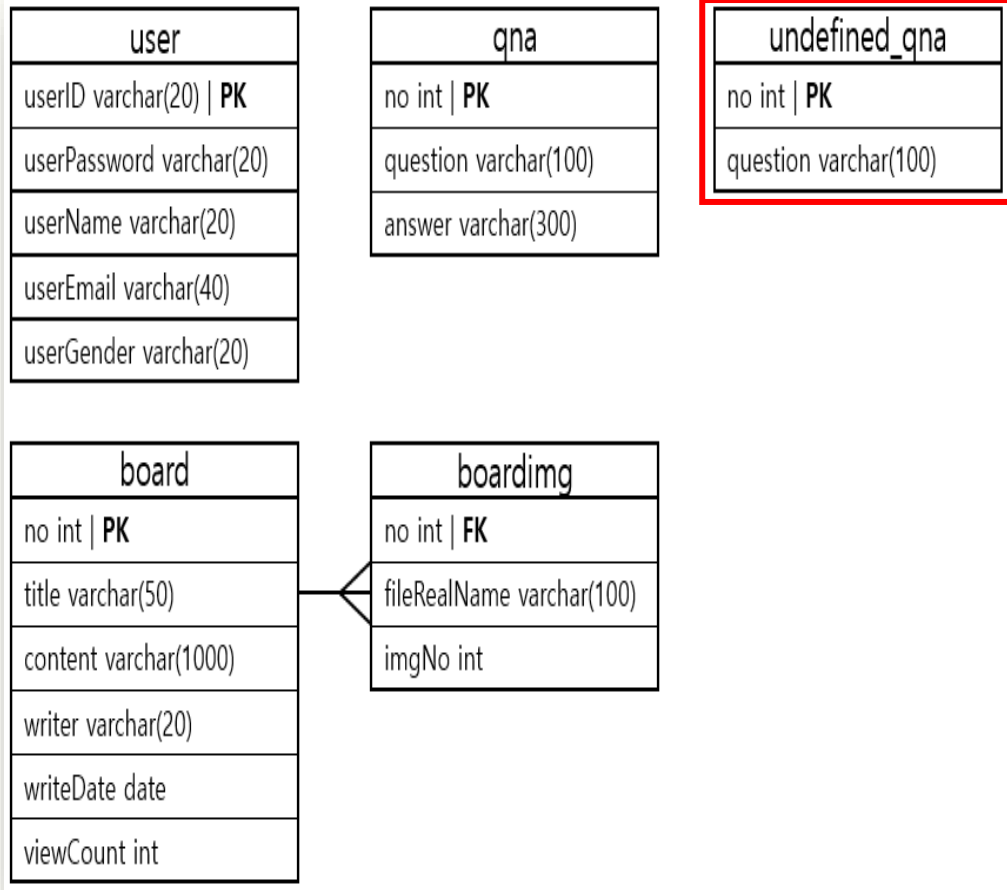
# ER-Diagram



## qna 테이블

- no 필드는 Primary Key로 질문과 답변 레코드의 고유 번호를 저장하는 정수형 타입이다.
- question 필드는 질문을 저장하는 가변길이의 문자열 타입이다.
- answer 필드는 답변을 저장하는 가변길이의 문자열 타입이다.
- 와이밀리가 하는 질문에 대한 답을 사용자가 했을 경우 qna테이블에 레코드를 추가한다.
- 추후 사용자가 관련된 질문을 했을 시 레코드를 검사 후 답변을 추출하여 와이밀리가 답변한다.

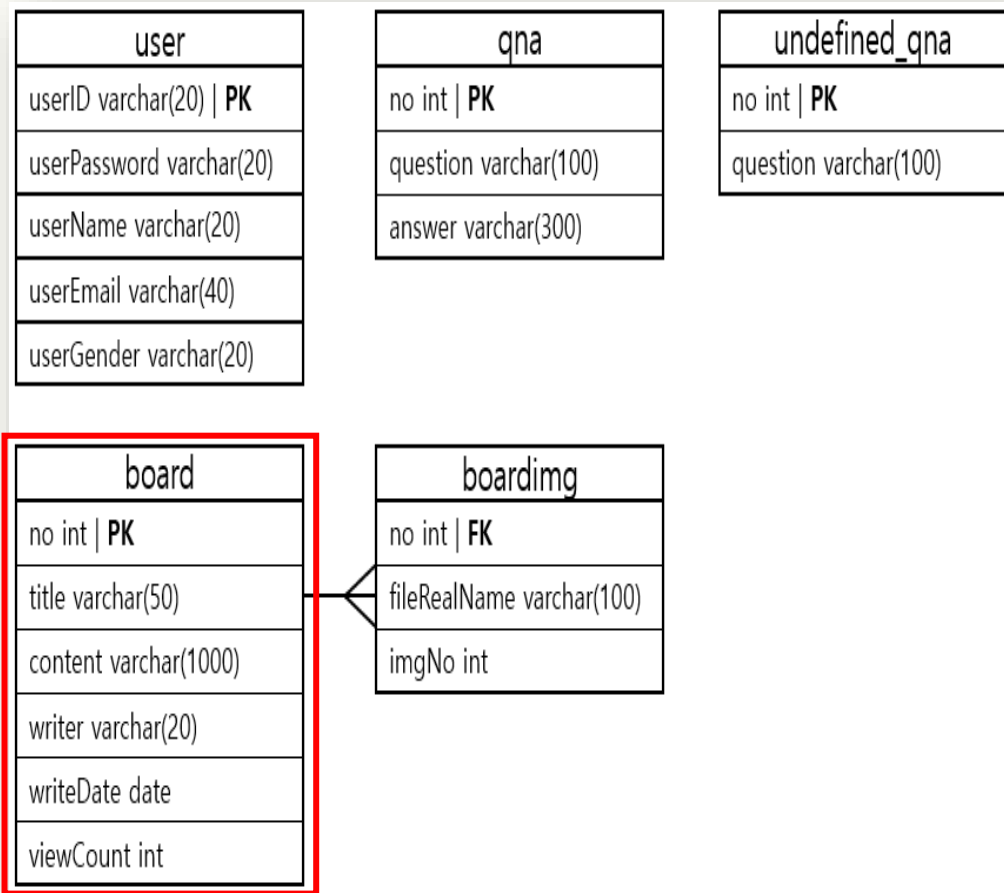
# ER-Diagram



## undefined\_qna 테이블

- no 필드는 Primary Key로 답변이 없는 질문만 있는 레코드의 고유 번호를 저장하는 정수형 타입이다.
  - question 필드는 질문을 저장하는 가변길이의 문자열 타입이다.
  - 와이밀리가 사용자에게 하는 수 많은 질문 테이블
  - 이 테이블 레코드는 여태 사용자들이 와이밀리에 게 했던 질문들이 담겨있다.
- \* 해당 테이블은 질문만 있는 테이블이며, 챗봇 내에서 레코드를 삭제하지 않는다.

# ER-Diagram

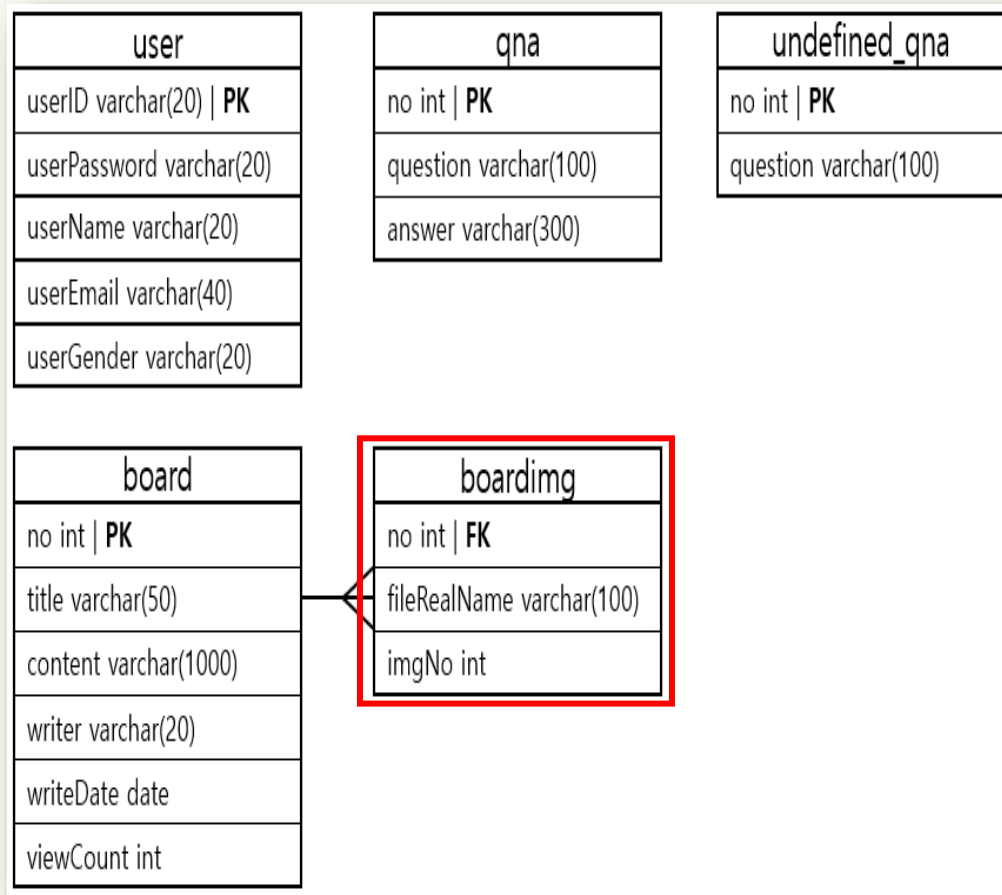


## board 테이블

- no 필드는 Primary Key로 게시글의 고유 번호를 저장하는 정수형 타입이다.
- title 필드는 게시글 제목을 저장하는 가변길이의 문자열 타입이다.
- content 필드는 게시글 내용을 저장하는 가변길이의 문자열 타입이다.
- writer 필드는 작성자를 저장하는 가변길이의 문자열 타입이다.
- writeDate 필드는 작성된 날짜를 저장하는 date타입이다
- viewCount 필드는 조회수를 저장하는 정수형 타입이다.



# ER-Diagram



## boardimg 테이블

- no 필드는 Foreign Key로 게시글의 번호를 저장하는 정수형 타입이다. (제약조건을 고려하여 서버DB에서는 따로 FK를 설정하지 않았음.)
- fileRealName 필드는 이미지 파일명을 저장하는 가변길이의 문자열 타입이다.
- imgNo 필드는 몇 번째 이미지인지 저장하는 정수형 타입이다.

---

# ER-Diagram

---

```
mysql> desc user;
```

Field	Type	Null	Key	Default	Extra
userID	varchar(20)	YES		NULL	
userPassword	varchar(20)	YES		NULL	
userName	varchar(20)	YES		NULL	
userEmail	varchar(40)	YES		NULL	
userGender	varchar(2)	YES		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> desc qna;
```

Field	Type	Null	Key	Default	Extra
no	int	NO	PRI	NULL	
question	varchar(100)	YES		NULL	
answer	varchar(300)	YES		NULL	

```
3 rows in set (0.00 sec)
```

```
mysql> desc undefined_qna;
```

Field	Type	Null	Key	Default	Extra
no	int	NO	PRI	NULL	
question	varchar(100)	YES		NULL	

```
2 rows in set (0.00 sec)
```

## MySQL

- user 테이블
- qna 테이블
- undefined\_qna 테이블

---

# ER-Diagram

---

## MySQL

- board 테이블

```
mysql> desc board;
```

Field	Type	Null	Key	Default	Extra
no	int	NO	PRI	NULL	
title	varchar(50)	YES		NULL	
content	varchar(1000)	YES		NULL	
writer	varchar(20)	YES		NULL	
writeDate	date	YES		NULL	
viewCount	int	YES		NULL	

```
6 rows in set (0.00 sec)
```

```
mysql> desc boardimg;
```

Field	Type	Null	Key	Default	Extra
no	int	YES		NULL	
fileRealName	varchar(100)	YES		NULL	
imgNo	int	YES		NULL	

```
3 rows in set (0.00 sec)
```

- boardimg 테이블
-

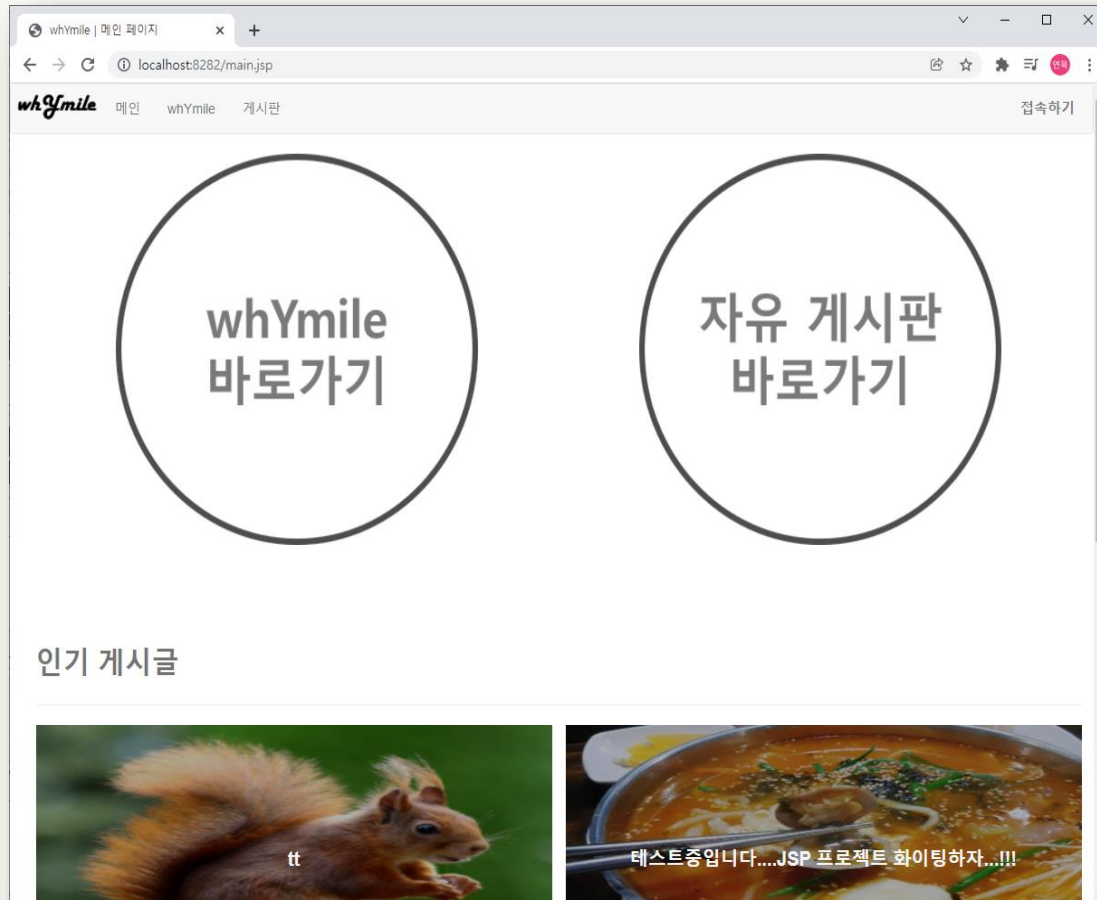
---

A dark gray rectangular box with the word "Result" centered inside in a white serif font.

**Result**

---

# Result



## 메인 페이지

- 메인 페이지는 whYmile 바로가기, 자유 게시판 바로가기, 인기 게시글이 있다.
- 바로가기는 해당 기능을 이용할 수 있는 페이지로 이동한다.
- 인기 게시글을 클릭하면 해당 게시글을 볼 수 있는 페이지로 넘어가게 된다.

# Result

whYmile | 회원가입

← → ↺ ⓘ localhost:8282/user/joinForm.jsp

whYmile 메인 whYmile 게시판 접속하기

### 회원가입 화면

아이디

비밀번호

비밀번호 확인

이름

이메일

남자 여자

회원가입

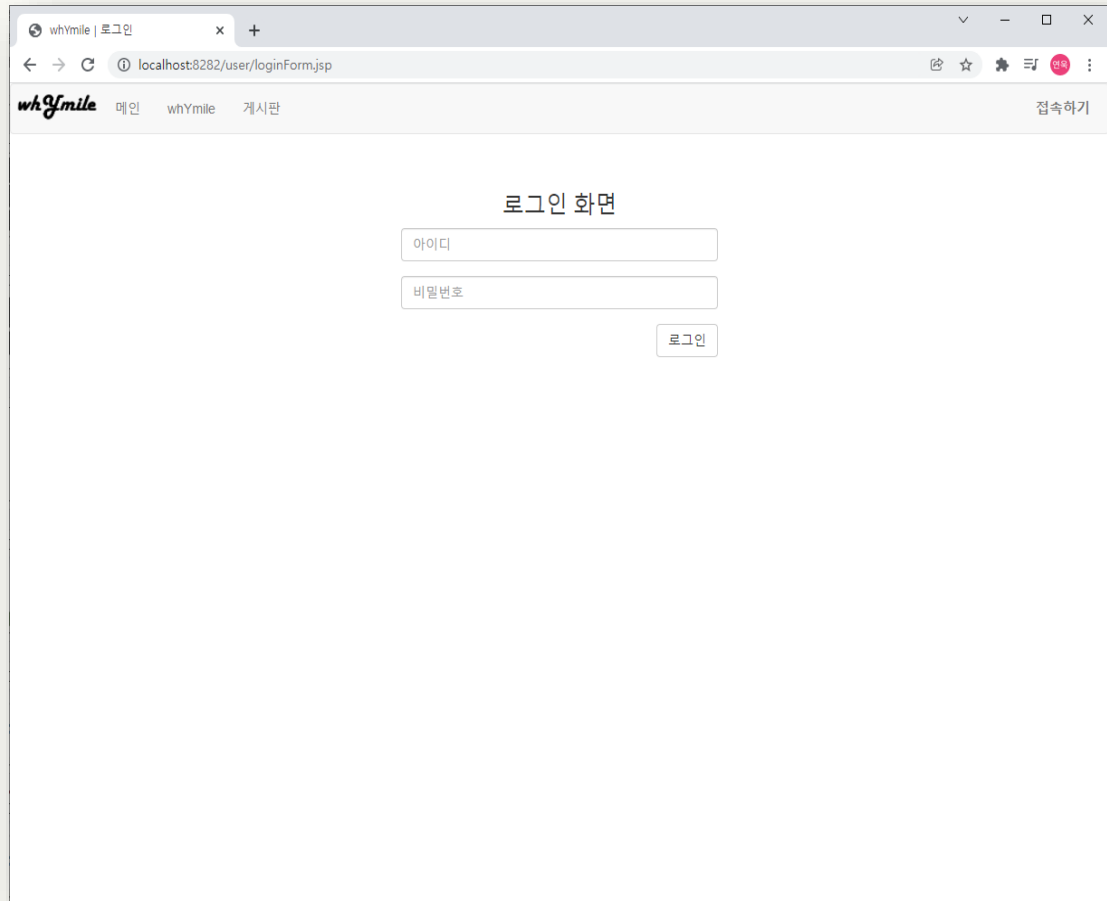
## 회원가입

- 회원가입에서 입력한 정보는 user 테이블에 저장된다.
- 이 때 이미 생성된 아이디가 있다면 해당 아이디로 회원이 생성되지 않는다.

---

# Result

---



## 로그인

- 회원가입에서 생성한 정보를 가지고 아이디와 비밀번호를 입력해서 로그인을 하는 화면이다.
- 아이디 : 201844045 / 비밀번호 : dusdnr123
- 위 정보의 계정을 하나 만들었다.

# Result

whyMile | 회원정보 수정

localhost:8282/user/updateForm.jsp

whyMile 메인 whyMile 게시판 201844045

### 회원정보 수정

201844045

현재 비밀번호

수정할 비밀번호

수정할 비밀번호 확인

김연욱

201844045@lfc.ac.kr

남자 여자

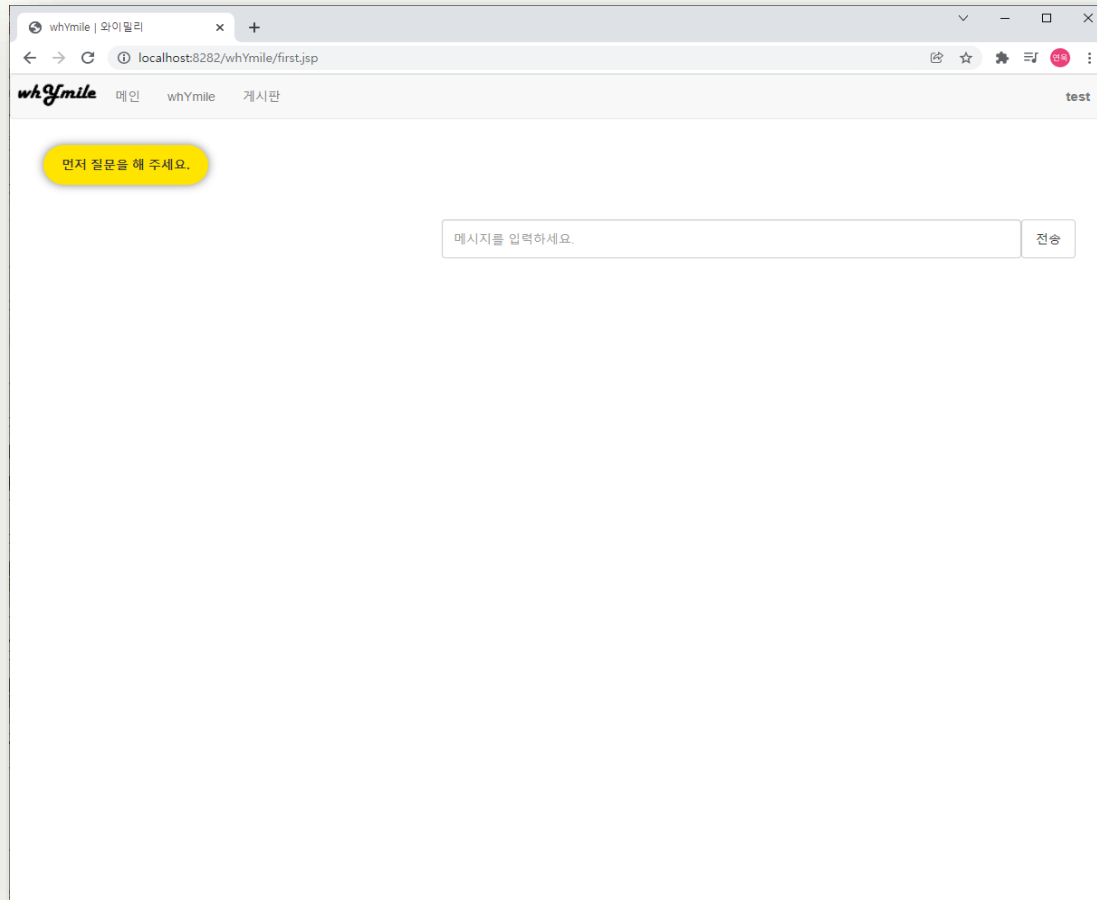
정보수정

## 회원정보 수정

- 로그인한 정보를 수정하는 페이지이다.
- 맨 위는 아이디로 변경할 수 없는 칸이다.
- 정보를 수정하기 위해서는 현재 비밀번호를 입력해야 한다.
- 만약 이름이나 이메일 또는 성별을 바꾸고자 하면, 수정할 비밀번호와 수정할 비밀번호 확인 필드는 입력을 안해주어도 된다.



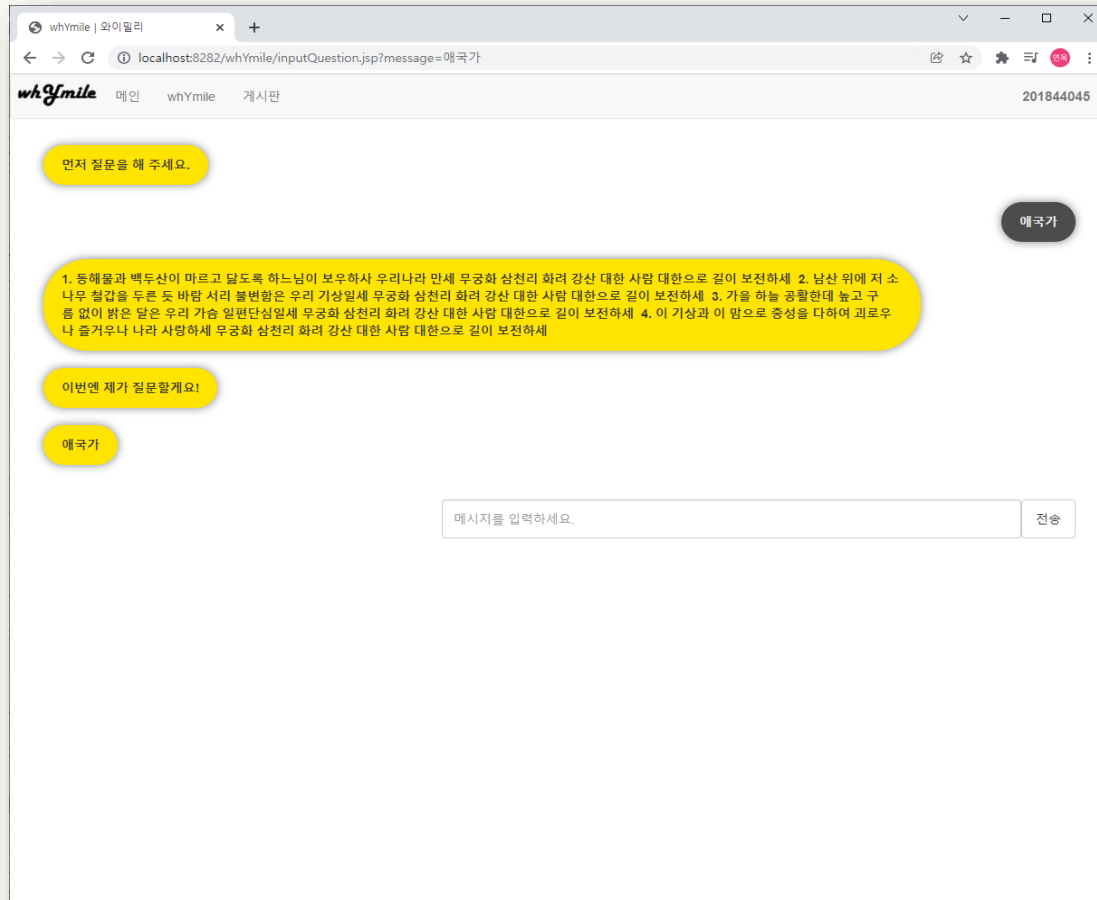
# Result



## whYmile first

- 상단 navbar에서의 “whYmile” 혹은 메인 페이지에서의 “whYmile 바로가기” 클릭 시 이동되는 페이지 이다.
- 와이밀리가 “먼저 질문을 해 주세요.” 라는 말을 하며 사용자는 질문을 하면 된다.
- 이 기능은 로그인 시에만 사용 가능한 기능 이다.

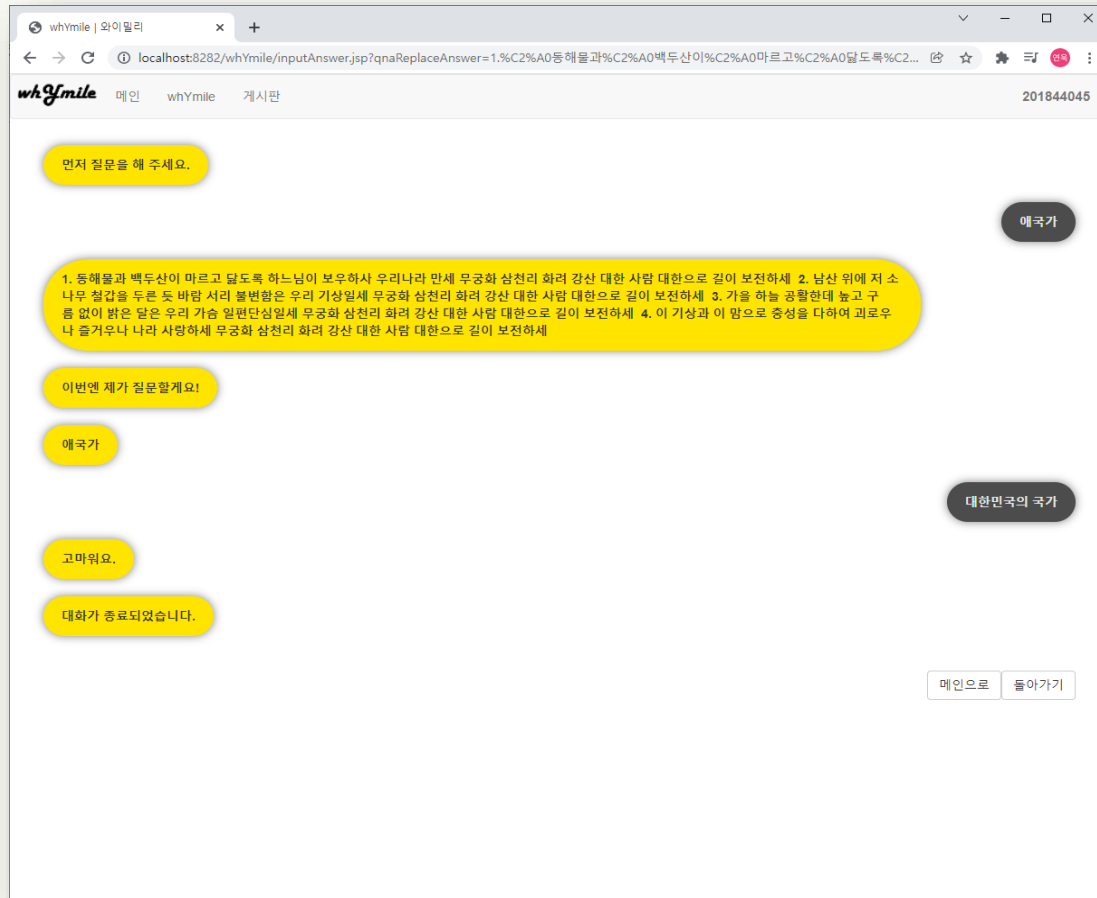
# Result



## whYmile second

- 사용자가 애국가라는 메시지를 입력하였고, 이와 관련된 답변을 와이밀리가 해준 화면이다.
- 그리고 다음에는 와이밀리가 사용자에게 질문을 한다.
- 이 때 질문 내용은 이전에 사용자들이 와이밀리에게 질문했던 질문 내용이며, 이렇게 하여 답변을 얻으면 qna테이블에 추가하는 형태이다.
- 그렇게 여러가지 형태의 정보를 모으고 보여주는 것이 와이밀리이다.

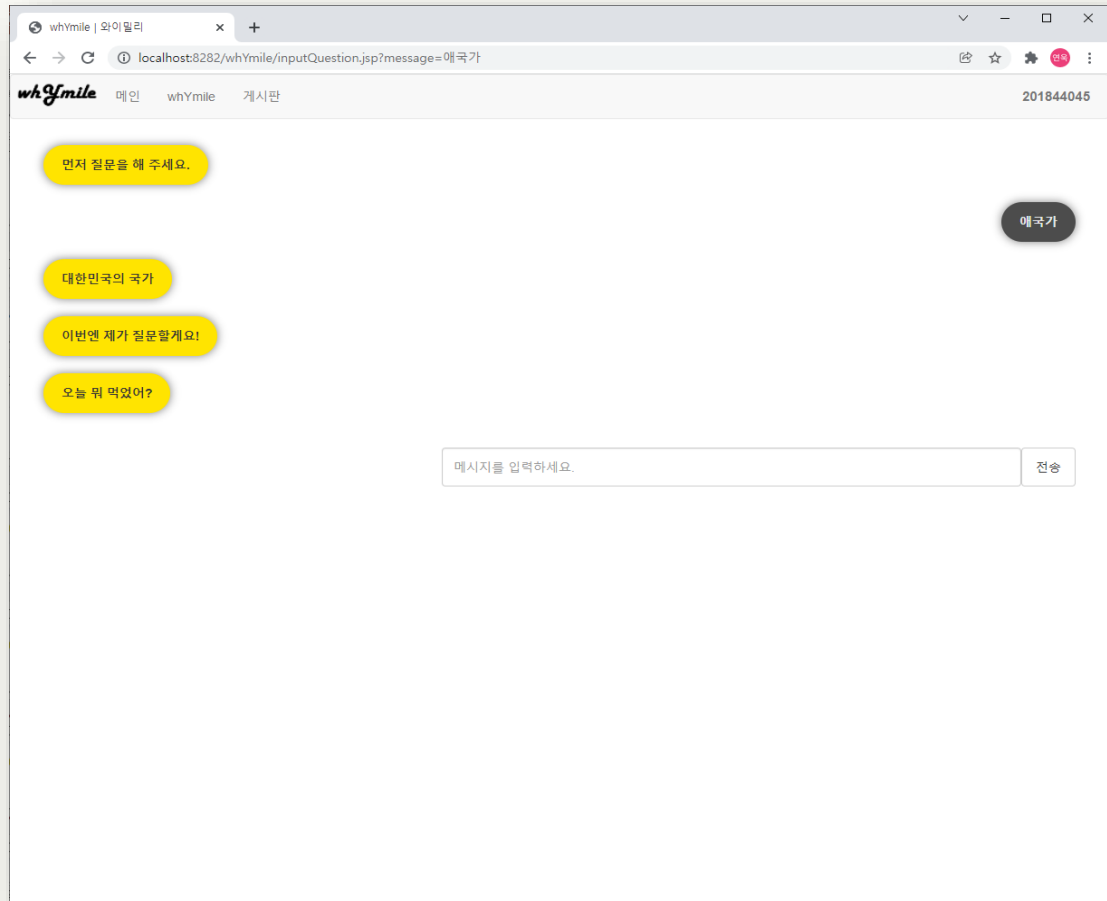
# Result



## whYmile third

- 사용자가 와이밀리 질문에 대한 답변을 하면 대화가 종료된다.
- 메인으로 버튼은 메인 페이지로, 돌아가기 버튼은 whYmile의 first페이지로 돌아가게 되어 다시 질문하고 답 할 수 있는 페이지로 가게 된다.

# Result



## whYmile 다른 결과

- 방금 입력한 애국가에 대한 답변이 사용자가 질문했을 때 나오는 화면이다.
- 이렇게 애국가와 관련된 수 많은 답변 중 랜덤하게 추출하며 답변해 주는 시스템이다.
- 입력 값을 어절로 나누어 제일 많이 연관된 값들 중 랜덤하게 추출하는 시스템이다.
- 이 시스템은 사용자가 어떤 답변을 저장하느냐에 따라 정확한 정보를 얻을 수도 있고, 엉뚱한 답변을 얻을 수도 있기에 신뢰보다는 재미의 챗봇 시스템이다.

---

# Result

---

## whYmile 챗봇 흐름 설명

1. 사용자가 와이밀리에게 질문을 한다.
    - 1.1. 사용자가 한 질문이 undefined\_qna테이블에 없을 경우 레코드 추가한다.
    - 1.2. 사용자가 한 질문이 undefined\_qna테이블에 있을 경우 레코드 추가하지 않는다.
  2. 와이밀리가 질문에 대한 답변을 한다.
    - 2.1. 해당 질문과 관련된 레코드를 qna테이블에서 찾는다.  
(어절로 나누어서 레코드의 질문이 사용자의 질문과 똑같거나 어절이 1개 차이가 나는 질문이 있는 레코드를 랜덤 수를 이용하여 추출함)
      - 2.1.1. 해당 질문에 대한 답변이 추출되었으면 와이밀리가 답변한다.
      - 2.1.2. 해당 질문에 대한 답변이 없다면 “저도 잘 모르겠어요.”라고 답변한다.
  3. 와이밀리가 사용자에게 질문한다.
    - 3.1. undefined\_qna테이블에 있는 질문을 랜덤 수를 이용하여 추출해서 질문한다.
      - 3.1.1. undefined\_qna테이블에 질문이 없다면 “질문할 내용이 없어요.”라고 한다. (이 경우는 사용자의 입력이 한 번도 이루어지지 않았을 때만 나옴)
-

---

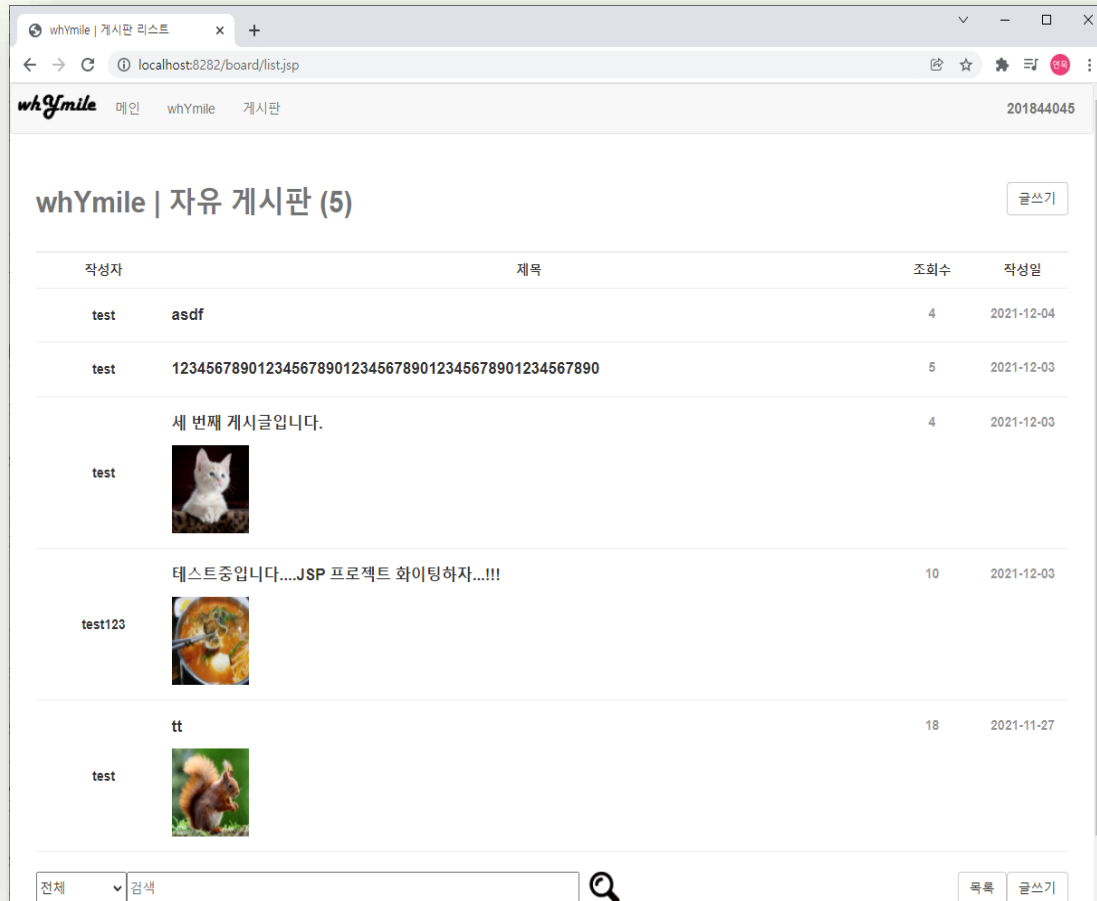
# Result

---

## whYmile 챗봇 흐름 설명

4. 사용자가 와이밀리 질문에 대해 답변한다.
  - 4.1. 와이밀리 질문과 사용자 답변을 합친 레코드를 qna테이블에 저장한다.
5. 와이밀리가 종료된다.
  - 5.1. “고마워요.”라는 메시지 이후 “대화가 종료되었습니다.”라는 메시지를 띄운다.
  - 5.2. 돌아가기 버튼과 메인으로 버튼을 보여준다.

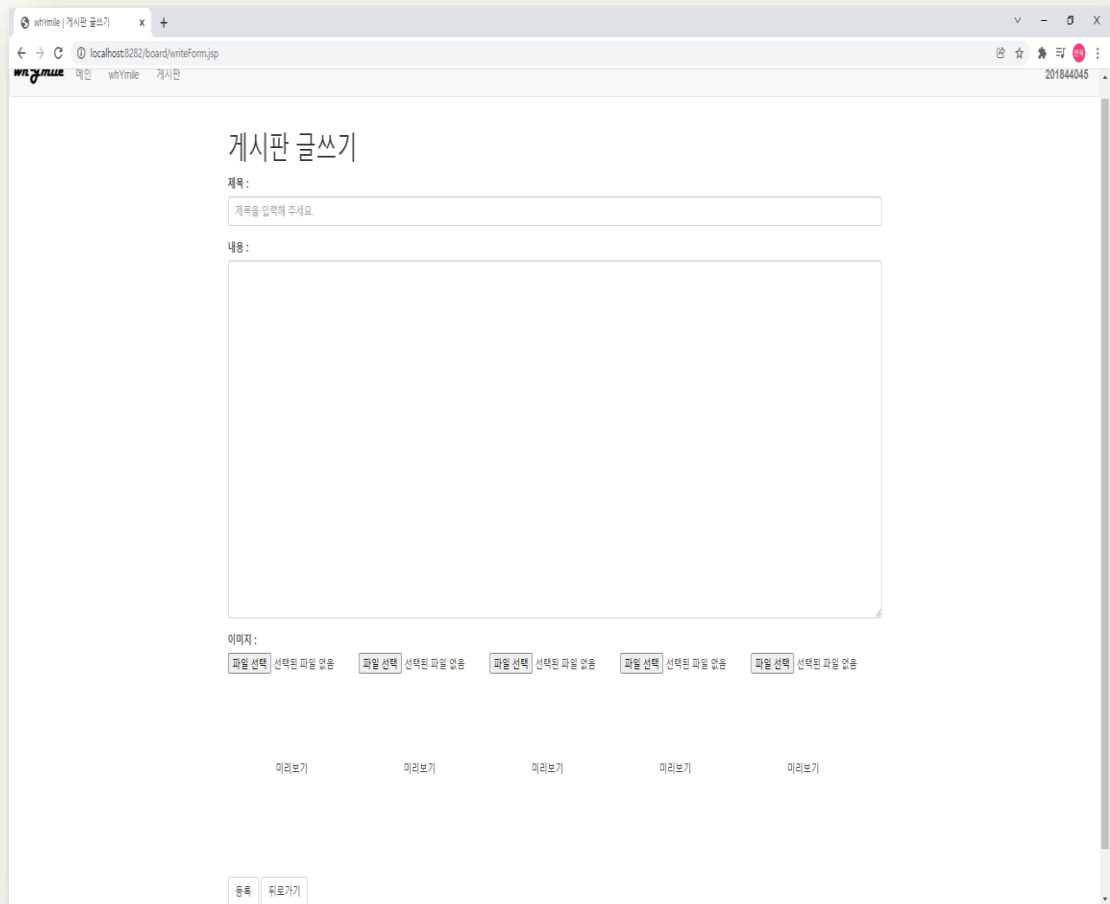
# Result



## 게시판

- 상단 navbar에서의 “게시판”이나 메인 페이지에서의 “자유 게시판 바로가기”를 클릭 시 해당 페이지로 이동하게 된다.
- 현재 작성된 게시글을 볼 수 있으며, 한 페이지당 15개의 게시글을 볼 수 있다. (15개가 넘어가면 우측에 다음과 이전 버튼을 생성하게 된다.)
- 게시글 작성 시 이미지를 등록하면 등록된 이미지를 썸네일처럼 보여준다.
- 그리고 하단에 검색 부분에서 드롭 다운을 통해 검색하고자 하는 부분을 검색할 수 있다.

# Result



## 글쓰기

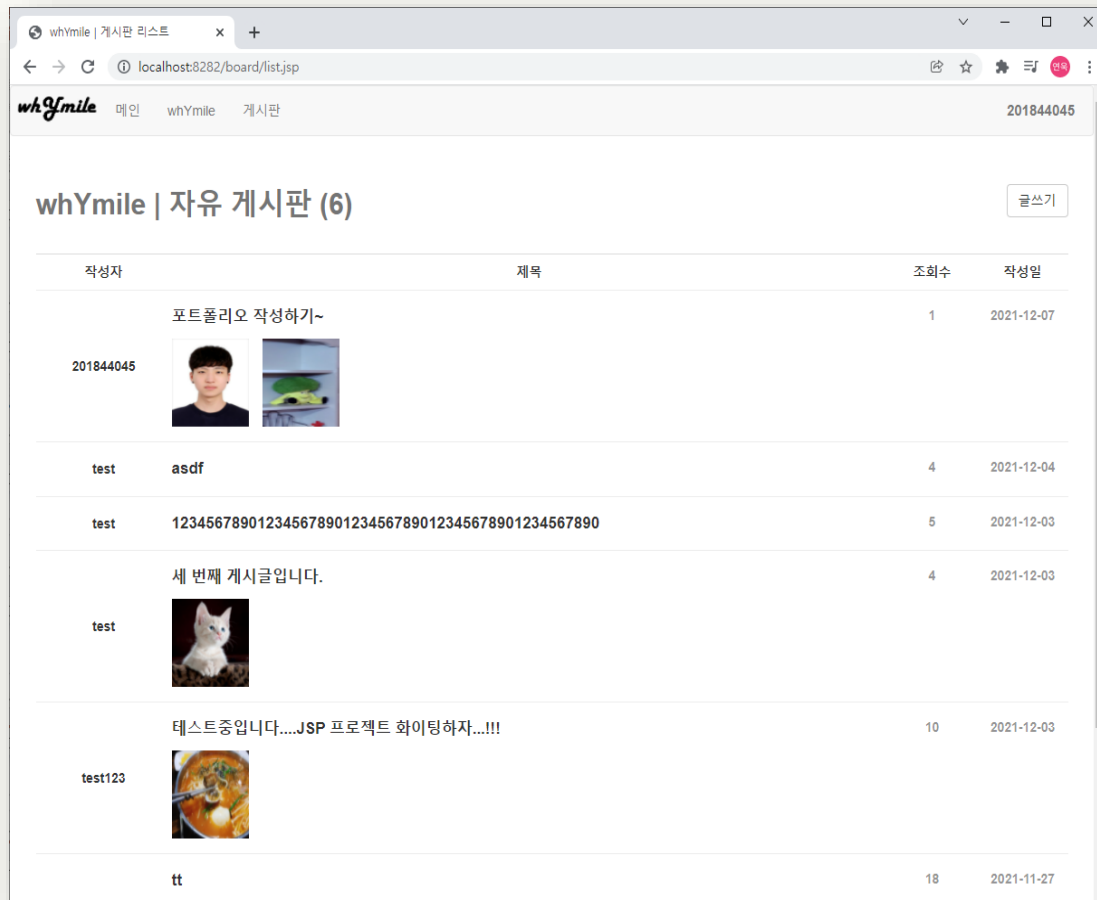
- 제목, 내용, 이미지를 선택하여 글 작성이 가능한 페이지이다.
- 이미지는 총 5개 업로드가 가능하며, 몇 번째에 파일을 올려도 보여질 때는 정상적으로 처음부터 순서대로 보여지게 된다.
- 이 때 이미지 파일 선택 부분에서 사용자가 임의로 pdf파일, exe파일 등 다른 파일을 업로드 할 수 있으므로 해당 부분에 대한 문제 또한 잡아 주었다.
- 이 기능은 로그인 시에만 사용 가능하다.



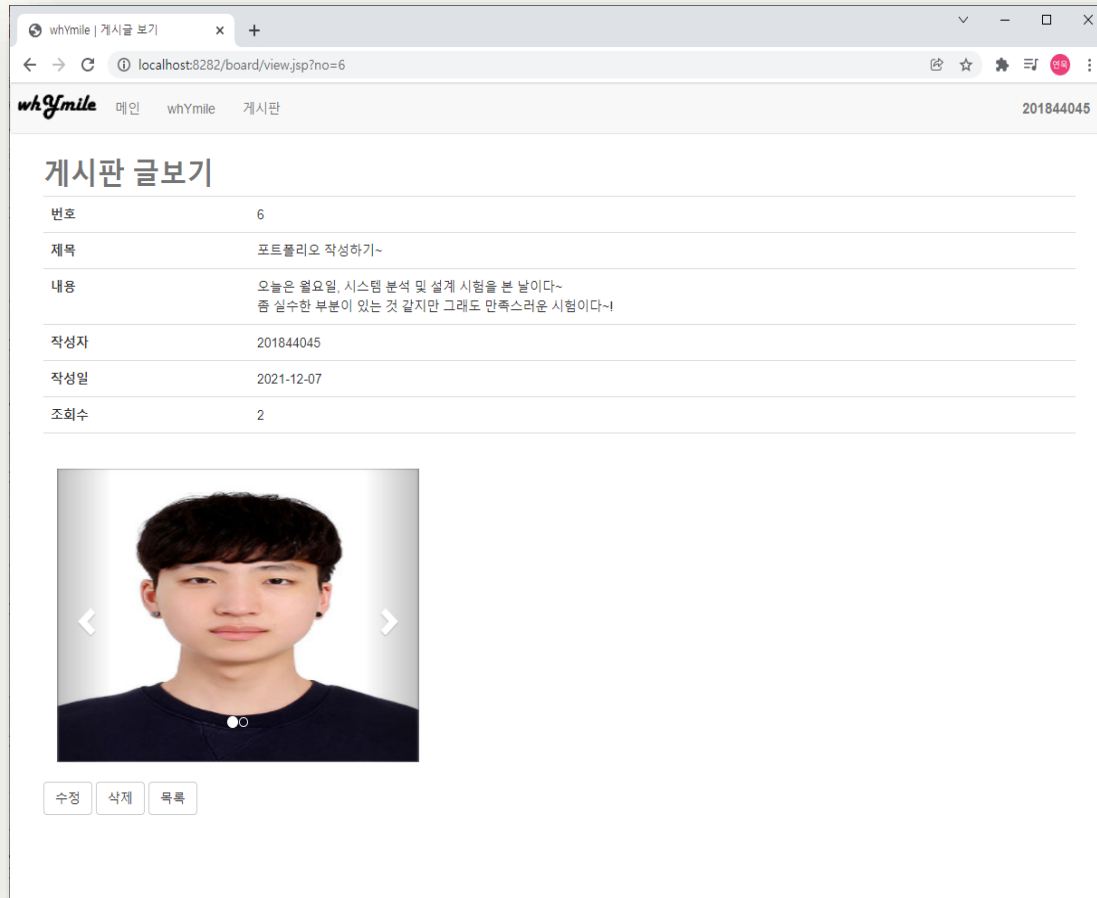
# Result

## 글쓰기 후

- 글을 작성한 후 자유 게시판 부분이다.
- 이미지는 2,3번째 칸에 업로드를 하였고, 그 두 장이 성공적으로 불러와 짐을 볼 수 있다.



# Result



## 글 보기

- 방금 작성한 글을 보는 화면이다.
- 글 작성시 입력한 제목과 내용이 정상적으로 DB에서 불러와 짐을 볼 수 있다.
- 조회수는 로그인 상태에서 게시글을 볼 때만 올라가게 코딩을 한 상태로, 1에서 1이 증가한 2가 된 것을 볼 수 있다.
- 이미지는 carousel을 이용하여 업로드한 이미지 개수만큼 자동으로 넘어가거나 수동으로 넘겨서 이미지를 볼 수 있다.
- 이 때 작성자와 로그인한 사용자 세션이 같으면 수정과 삭제 버튼이 보인다.

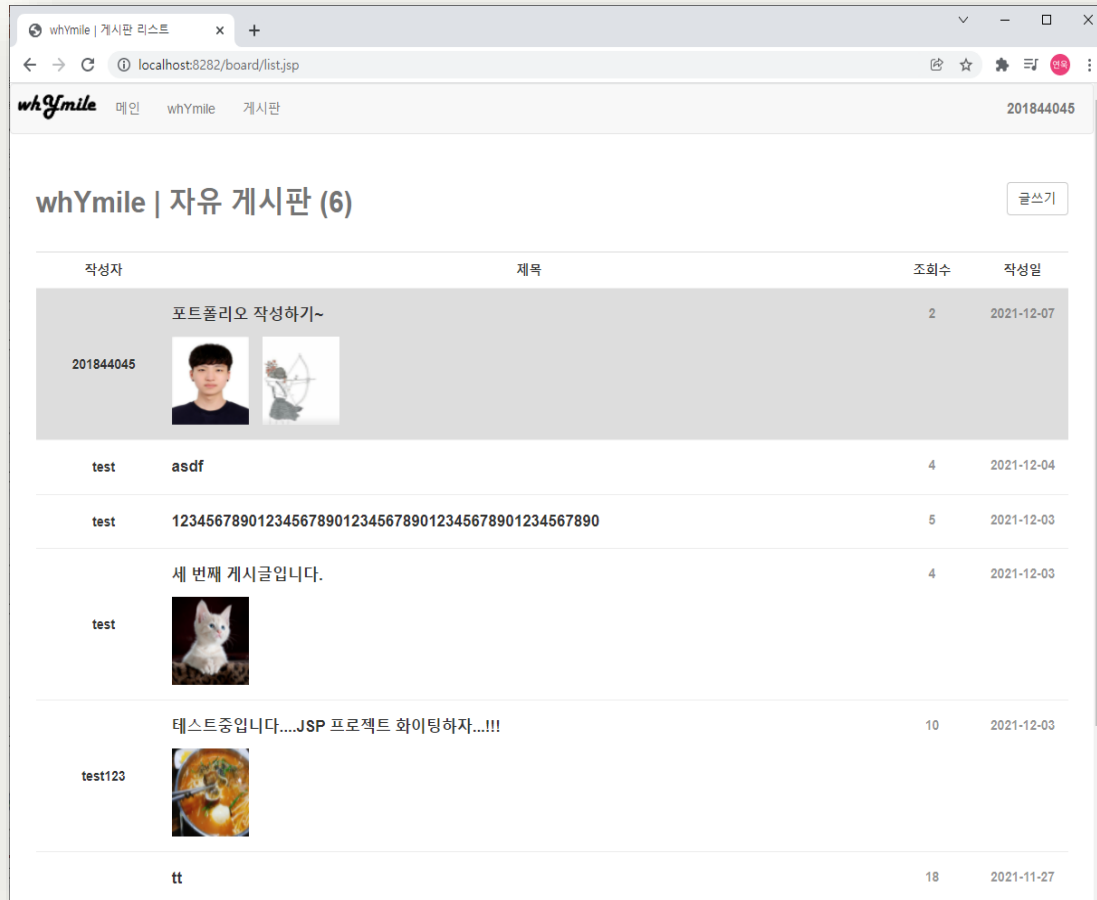
# Result

## 글 수정

- 방금 작성한 글을 수정하는 화면이다.
- 작성한 내용 그대로 불러와지며, 이미지 또한 업로드한 부분에 미리보기 대신 올라와 짐을 볼 수 있다.

The screenshot shows a web browser window with the address bar displaying 'localhost:8282/board/updateForm.jsp?no=6'. The page title is '게시판 글 수정하기' (Edit Post). The form includes fields for '번호 : 6' (Number: 6), '제목 : 포트폴리오 작성하기~' (Title: Portfolio Writing~), and '내용 : 오늘은 월요일, 시스템 분석 및 설계 시험을 본 날이다~ 좀 실수한 부분이 있는 것 같지만 그래도 만족스러운 시험이다~!' (Content: Today is Monday, I took the system analysis and design exam~ I made some mistakes but I'm satisfied with the exam~!). Below the content field is an '이미지 : ' (Image:) section with four file selection buttons labeled '파일 선택' and '선택된 파일 없음'. The first two buttons have preview images: a person's face and a green object on a shelf. The last two buttons have empty preview areas. At the bottom, the '작성자 : 201844045' (Author: 201844045) is displayed, along with '수정' (Edit) and '뒤로가기' (Back) buttons.

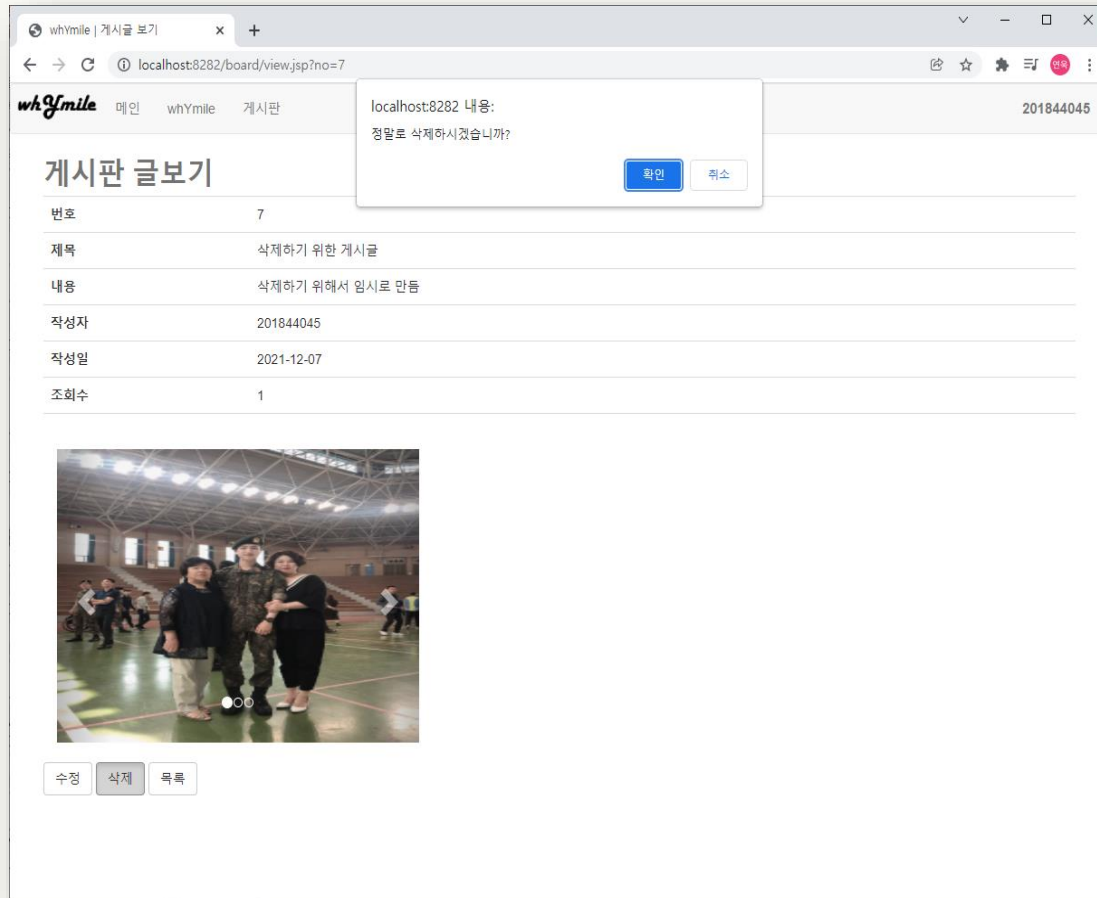
# Result



## 글 수정 후

- 세 번째 이미지 칸에 있던 브로콜리를 여자 아이가 장미 활을 쏘는 이미지로 수정한 결과이다.
- 로컬에서도 성공적으로 업로드 된 이미지가 삭제되며, DB에서도 정상적으로 처리가 되어 출력이 되는 것을 볼 수 있다.

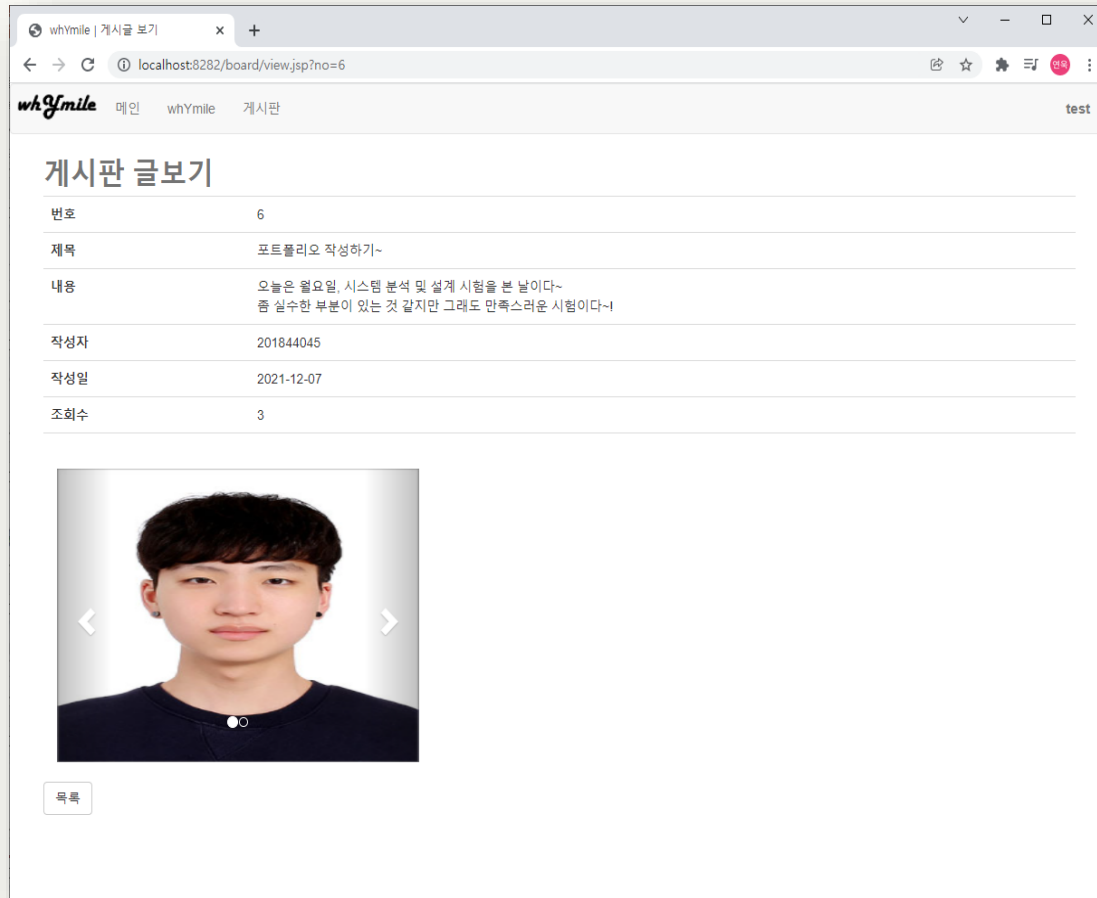
# Result



## 글 삭제

- 해당 글의 삭제 버튼을 누르면 알림 창이 뜨게 되며, 확인을 누르면 업로드 되었던 이미지, DB 레코드들이 전부 삭제 된다.
- 이렇게 게시글이 삭제가 되면 모든 게시글의 번호들이 다시 정렬되게 된다.
- 현재 7번을 지우면 그 이후 번호들은 -1을 하게 된 번호로 수정되며, 게시글의 이미지 테이블 번호 또한 모두 수정되게 된다.

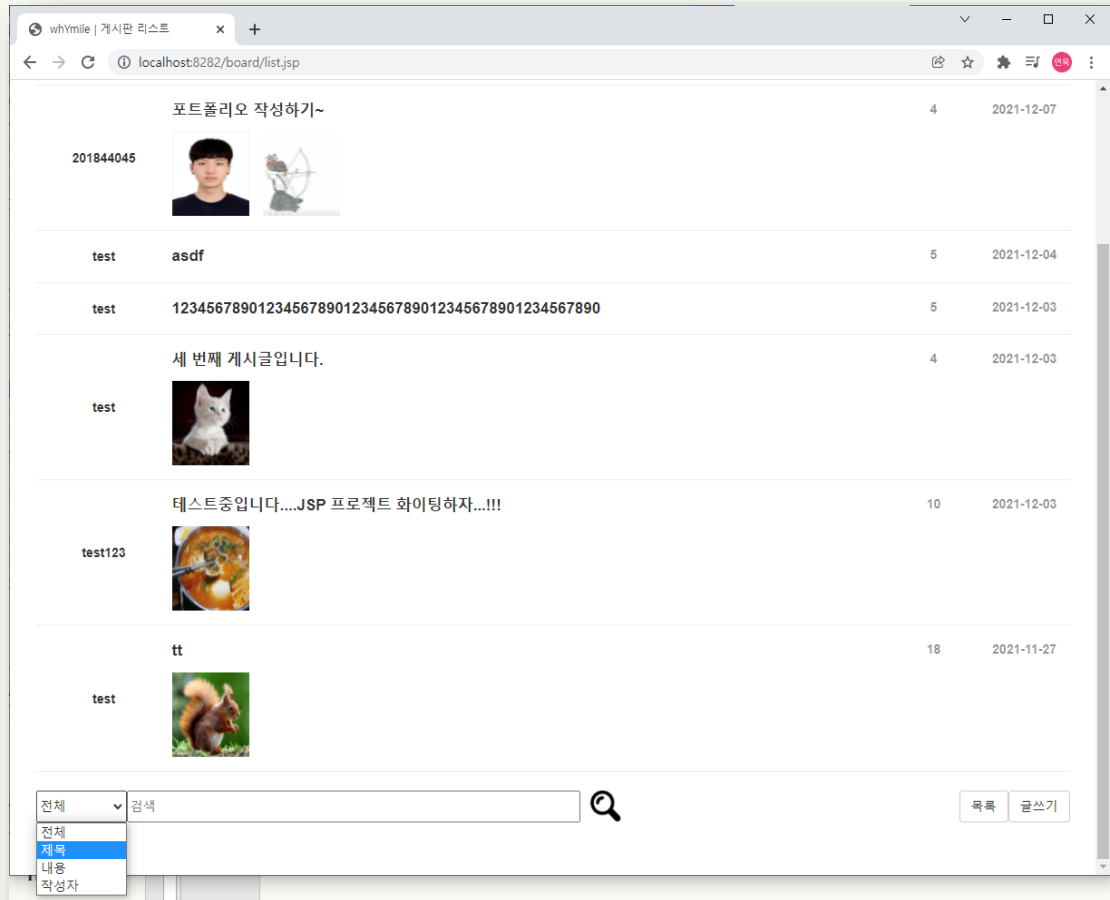
# Result



## 글 보기 버튼

- test계정으로 바꾸어 로그인을 한 상태이다.
- 해당 게시물 작성자는 201844045이므로 현재 로그인 된 세션과 작성자가 일치하지 않기 때문에 수정과 삭제 버튼이 없는 것을 볼 수 있다.
- 만약 사용자가 임의로 처리 페이지로 넘길 수 있기에 Action페이지에도 다 예외를 처리해 주었다.

# Result



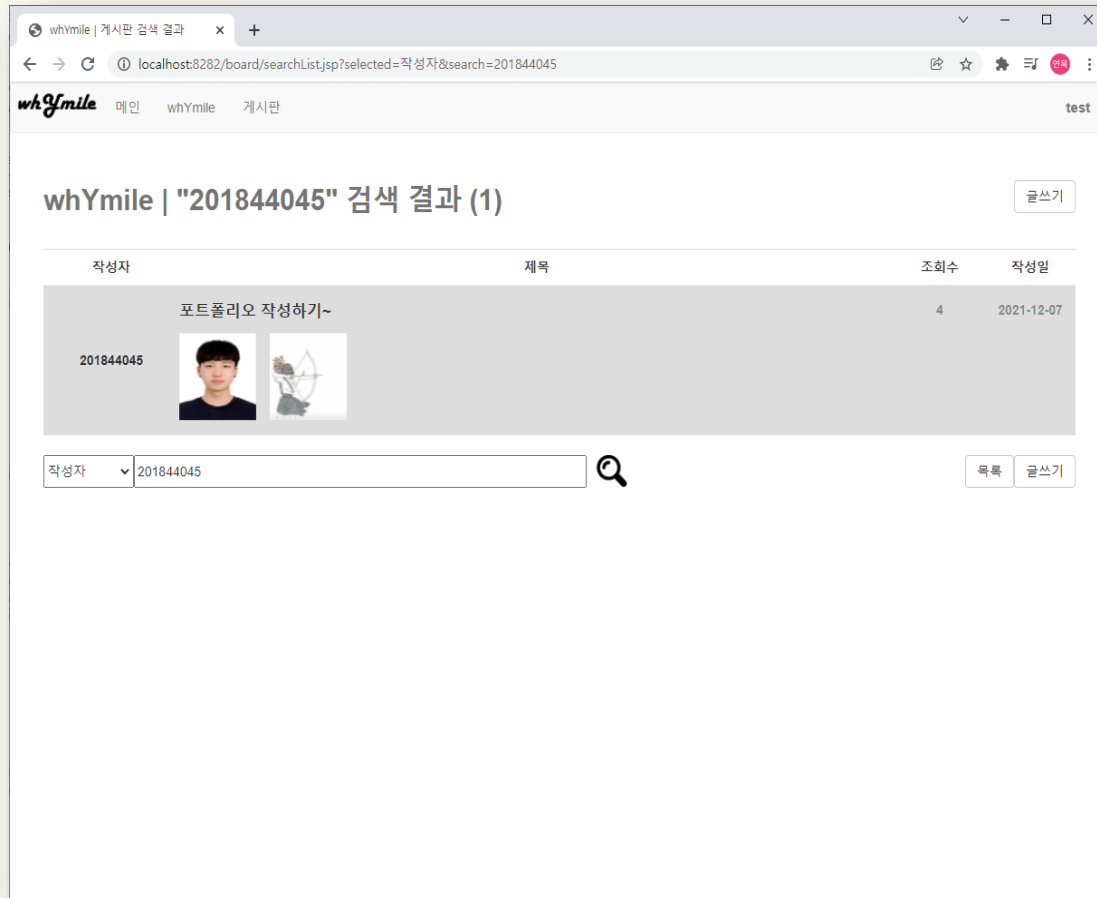
## 글 검색

- 글 검색 기능에는 드롭다운 메뉴를 총 4가지를 주었다.
- 전체를 선택하고 검색하면 제목, 내용, 작성자에서 입력한 검색 값과 연관되면 뜨게 된다.
- 입력 후 엔터키나 돋보기 버튼을 클릭하면 검색을 하게 된다.
- 이 때 입력 값이 없는 경우를 예외처리 해주어 “검색창에 입력을 해 주세요.” 라는 알림 창이 뜨게 해 두었다.

# Result

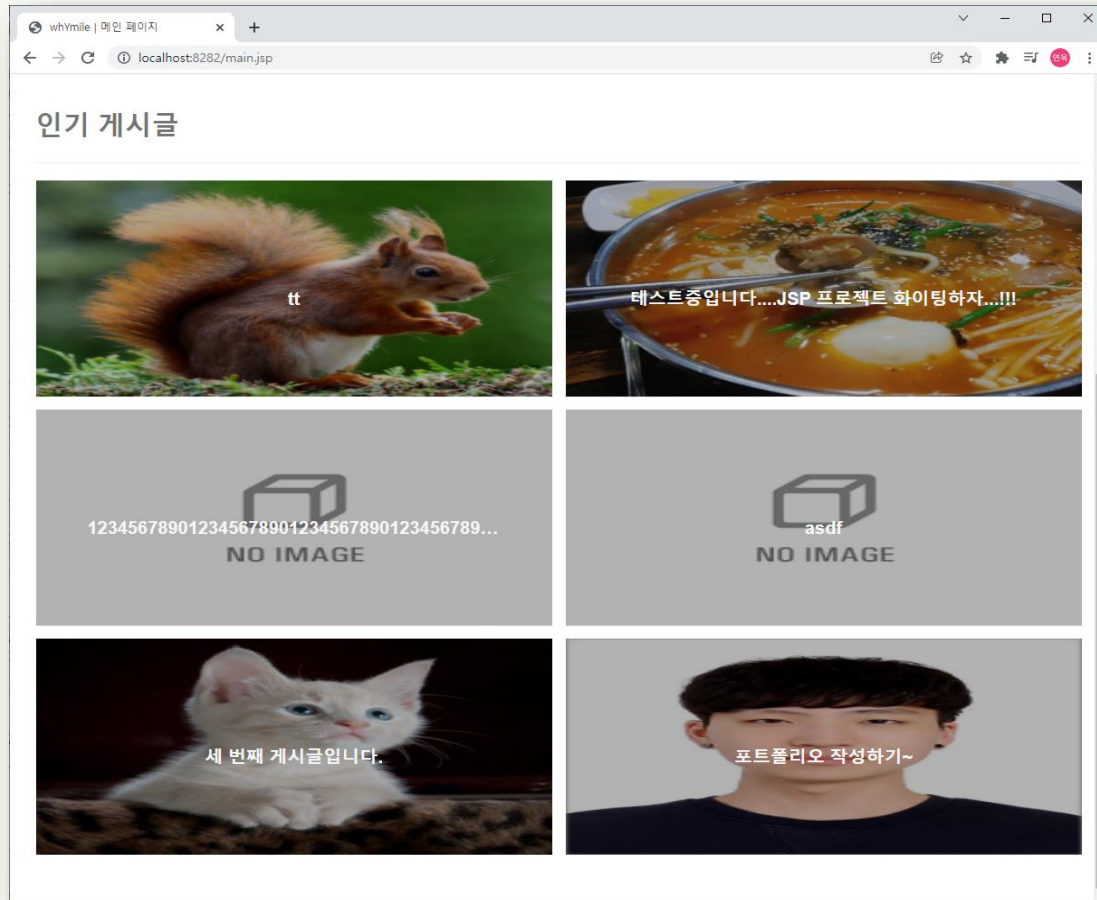
## 글 검색 결과

- 검색 시 선택한 드롭다운 메뉴와 입력 값을 그대로 검색창에 다시 띄워 주게 되며, 해당 검색 내용과 관련된 게시글을 보여준다.





# Result



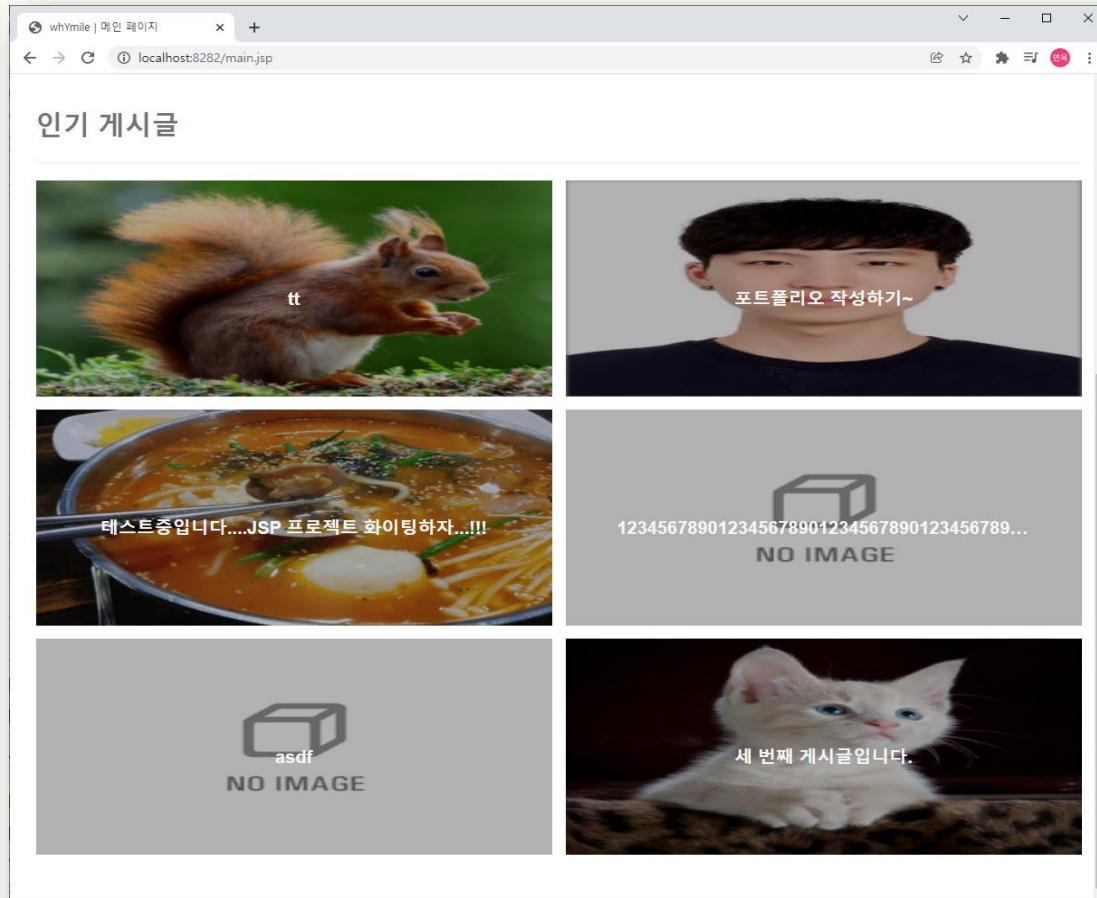
## 인기 게시물

- 현재 게시물 중 조회수가 제일 많은 순서대로 나열되게 된다.
- 게시물 작성시 이미지를 업로드 했을 시 제일 앞 순서에 넣은 이미지를 썸네일처럼 가져오게 된다.
- 만약 이미지가 없다면 NO IMAGE라는 이미지를 썸네일로 사용한다.
- 그리고 그 위에는 게시물의 제목을 흰색 글씨로 나타내 주었다.

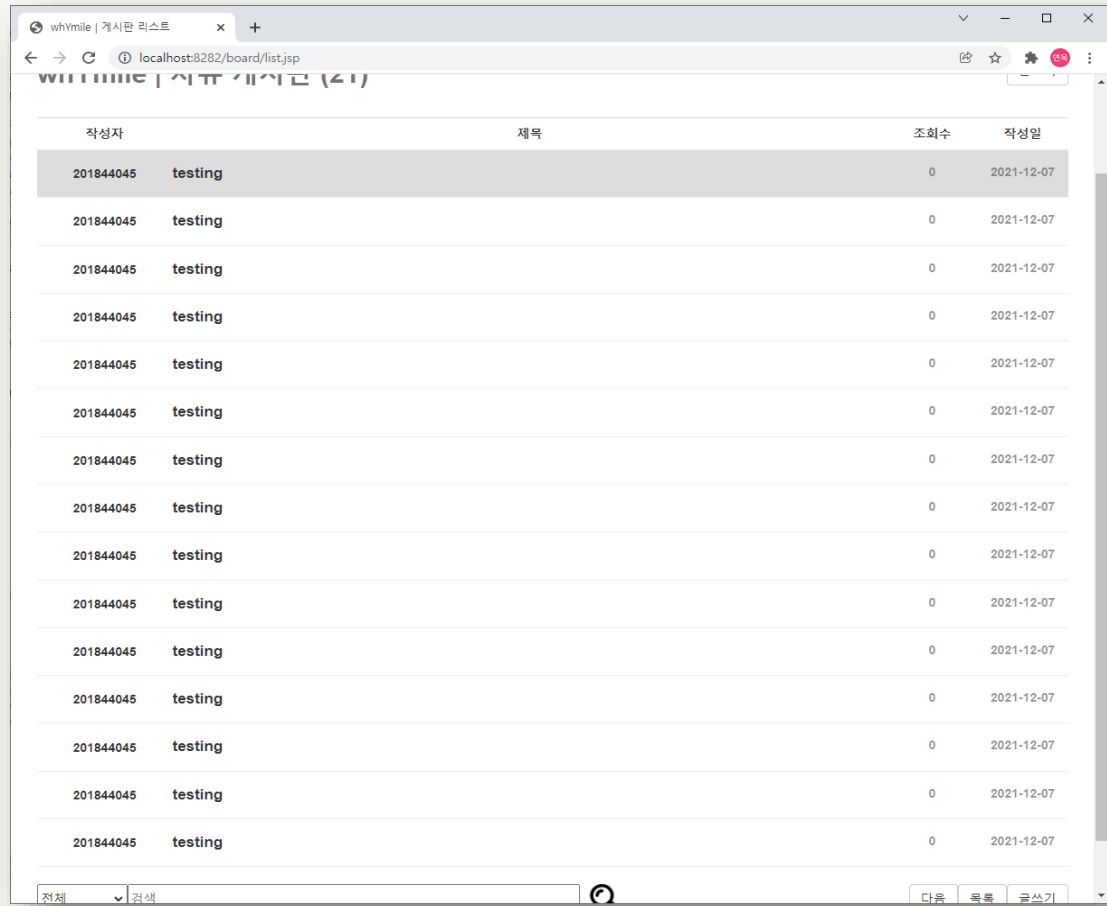
# Result

## 인기 게시물 변동

- 원래 6번째에 있던 “포트폴리오 작성하기~”라는 제목의 게시글의 조회수가 16으로 올랐다.
- 모든 게시물 중 조회수가 2등이기 때문에 2번째 칸으로 옮겨서 보여지는 것을 볼 수 있다.



# Result



작성자	제목	조회수	작성일
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07

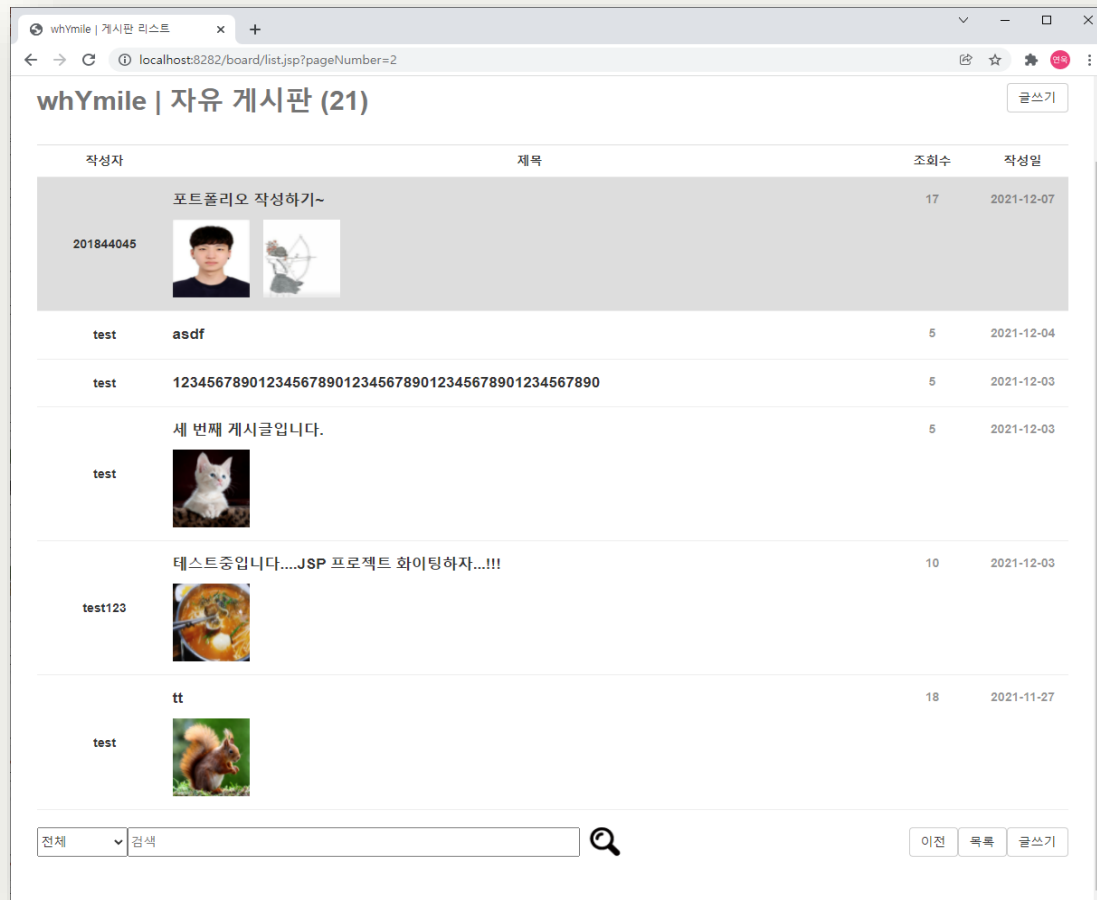
## 게시판 다음 버튼

- 임의로 게시글을 21개까지 작성하였다.
- 한 페이지에서는 15개의 게시글만 보이게 하며, 그 다음 게시글 부터는 다음 버튼을 눌러 다음 페이지로 이동되었을 때 보여지게 된다.
- 자유 게시판은 모든 글을 보여주기 때문에 모든 게시 글에 대하여 내림차순을 해주어 페이지의 게시 글들을 컨트롤 해주면 된다.

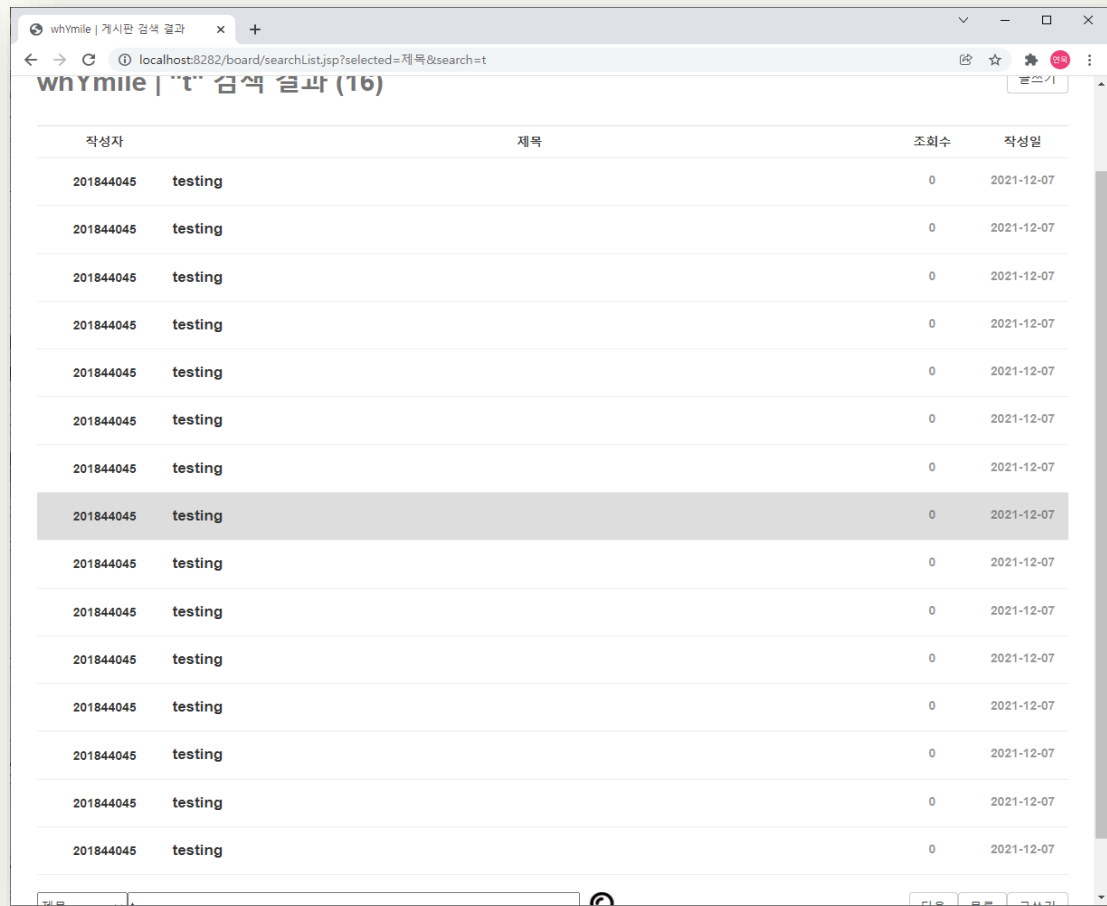
# Result

## 게시판 이전 버튼

- 이전 버튼이다.



# Result

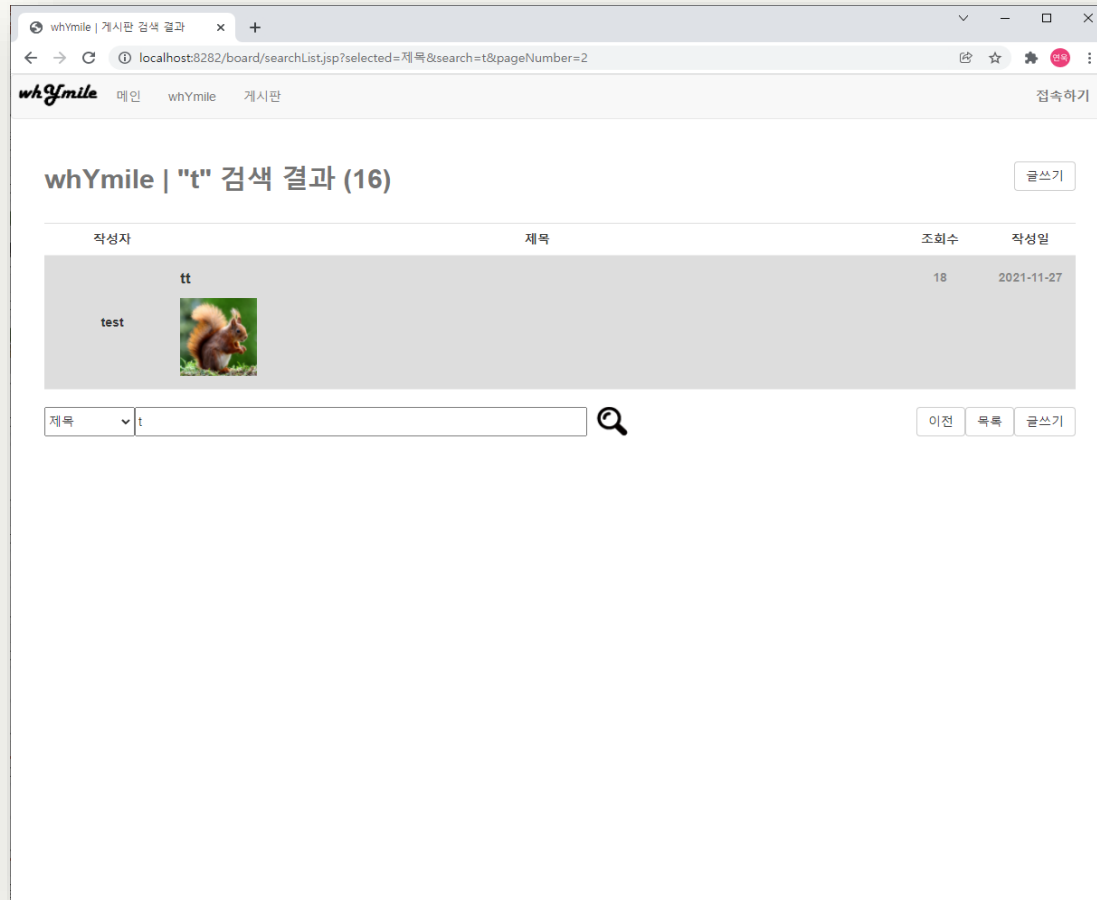


작성자	제목	조회수	작성일
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07
201844045	testing	0	2021-12-07

## 검색 시 다음 버튼

- 검색 했을 때에는 모든 게시글이 아니기 때문에 검색된 데이터만을 가지고 페이지의 게시 글들을 컨트롤 해주어야 한다.
- 현재는 제목에 t가 들어간 게시 글들을 검색한 경우이다.
- 우측 하단에 보면 다음 버튼이 정상적으로 나타남을 볼 수 있다.

# Result



## 검색 시 이전 버튼

- 검색 페이지에서 마지막 페이지라면 이전 버튼만을 활성화 시킨다.
- 다음, 이전 버튼들은 게시글 개수에 따라 아예 활성화가 되지 않거나, 하나만 활성화가 되거나, 둘 다 동시에 활성화가 되는 경우가 있다. (게시글이 31개일 경우 총 3페이지)

---

※Setting※

---

---

교수님 제가 이번 프로젝트를 진행하면서 Web Module Path를 손을 봐서 경로에 프로젝트명이 들어가지 않은 채 페이지 이동을 하게 됩니다

저의 포트폴리오 Result단계에서 주소를 보시면 프로젝트([http://localhost:8282/201844045\\_final/~~~](http://localhost:8282/201844045_final/~~~))와 달리 프로젝트명이 들어가지 않는 것을 확인하실 수 있으십니다..!

혹시 저의 코드를 돌려보신다면 교수님께서 경로설정을 살짝 만져 주셔야 합니다..!

많이 번거로우시겠지만 양해 부탁드립니다.....ㅠㅠㅠㅠㅠㅠ

(이걸 안해주면 이미지 업로드와 경로가 제대로 되지 않습니다....

제가 데스크 탑으로 작업한 뒤 노트북으로 파일을 옮겨 실행하는 과정에서 설정에 대한 문제가 발생하는 것을 확인하였습니다.. 중간고사 작업물도 이미지 업로드가 되지 않았을 까 걱정입니다...)

경로설정을 하는 법은 제가 뒤에 작성해 놓겠습니다..!!

---



# 코드 실행 전 설정

2번을 안 하면 워크스페이스 폴더에 이미지가 안 들어갑니다!  
(C:\Users\ccoma\eclipse-workspace\.metadata\plugins\org.eclipse.wst.server.core\tmp4\wtpwebapps\201844045\_final\img/) 에만 이미지가 올라가게 되어 war파일 생성 시 이미 업로드한 이미지가 들어가지 않게 됩니다...

그 다음 Server Locations에서 Use Tomcat installation (takes control of Tomcat installation)를 체크해 주시고, 그 아래 Server Options에서 Serve modules without publishing을 체크해 주십시오!

3 이후 Modules에 들어가 주십시오!

1 제 코드가 돌아갈 Tomcat서버를 만들어 주십시오!

# 코드 실행 전 설정

The screenshot shows the Eclipse IDE interface. On the left is the Project Explorer showing a project named '201844045\_final'. The main editor area displays the 'Web Modules' configuration dialog. A red box highlights the 'Add Web Module...' button in the top right corner of the dialog. A red arrow points from this button to a text box on the right side of the image.

이후 저의 프로젝트를 Add Web Module...을 통해 추가한 뒤 Edit...을 눌러 주십시오!!

# 코드 실행 전 설정

2 이렇게 / 한 개만 있으면 됩니다!

1 그 다음 Path부분을 / 하나만 남겨 주시고 OK해주시면 됩니다!  
(/201844045\_final -> /)  
그리고 해당 톰캣 서버로 제 프로젝트를 실행하게 되면 잘 돌아갈 겁니다..!!

Path	Document Base	Module	Auto Reload
/	201844045_final	201844045_final	Enabled

Document base: 201844045\_final  
Path: /  
☒ Auto reloading enabled

---

**Thanks !**

---