

Responsive Web PORTFOLIO

반응형 웹 포트폴리오

2021-11 ~ 2021-12
김준영

Responsive Web

목차

● 0. 프로젝트 설명

- 프로젝트 목적
- 프로젝트 소개

○ 1. 개발 프로그램

- 사용 프로그램

● 2. 계획

- 개발 예상 기간
- 초기 구상도

● 3. 개발 및 구현

- 웹 구현
- 프로그램 기능 구현
- 웹 디자인

○ 4. 후기

- 작업중 힘들었던 부분
- 추가 구현 & 개발 사항

Responsive Web

0. 프로젝트 설명

Responsive 반응하다.

반응형 웹 디자인이란

반응형 웹 디자인은 다양한 **디바이스, 창, 화면 크기**에서 원활하게 렌더링되는 웹 페이지를 제작을 말합니다.

웹 사이트를 반응형으로 설계하면, 어떤 디바이스에서 어떤 방식으로 접속하든지 웹 사이트의 **일관성을 유지**할 수 있습니다.

좋은 반응형 웹 디자인은 데스크탑과 모바일이 **단순히 화면 크기만 다른 것이 아니라** 둘을 이용하는 **소비자의 태도도 다르다**는 점을 인식하고 이를 반영한 디자인입니다.

Responsive Web Project

반응형 웹 을 구현하기 위해 **CSS** 파트와 **JSP** 부분에 대해 공부 했었고,
그 과정중 국비 교육 과정중 배웠던 **게시판**을 응용해서 여러 방면으로 보여줄 수 있는
프로그램을 개발하기로 생각했습니다.

그에 따라 **반응형 웹 디자인**을 적용 시킨 **블로그** 웹을 개발하기로 했습니다.
전체적인 프로그램의 형태는 게시판 형태를 참고하여 구현 하였으며, 이에 블로그
기능을 넣어주기 위해 조회수에 따른 메인 페이지의 변화등과 가독성등을 수정하며
제작하였습니다.

Responsive Web

1. 개발 프로그램

01



Spring

자바 플랫폼을 위한 오픈소스
애플리케이션 프레임워크

02



Javascript

웹페이지에 적용할 수 있게 하는
스크립트 혹은 프로그래밍 언어

03



Java

객체 지향적 프로그래밍 언어

Responsive Web

1. 개발 프로그램

04



Jsp

웹 브라우저에 돌려주는 서버
사이드 스크립트 언어

05



Mabatis

자바 퍼시스턴스 프레임워크
xml서술자를 sql문으로
객체 연결해주는 언어

06



Oracle XE

오라클 데이터베이스

Responsive Web

1. 개발 프로그램



Dbeaver

SQL 클라이언트이자
데이터베이스 관리 도구



pivotal tc Server Developer Edition

spring framework 로 지원되는
server

Responsive Web

2. 계획

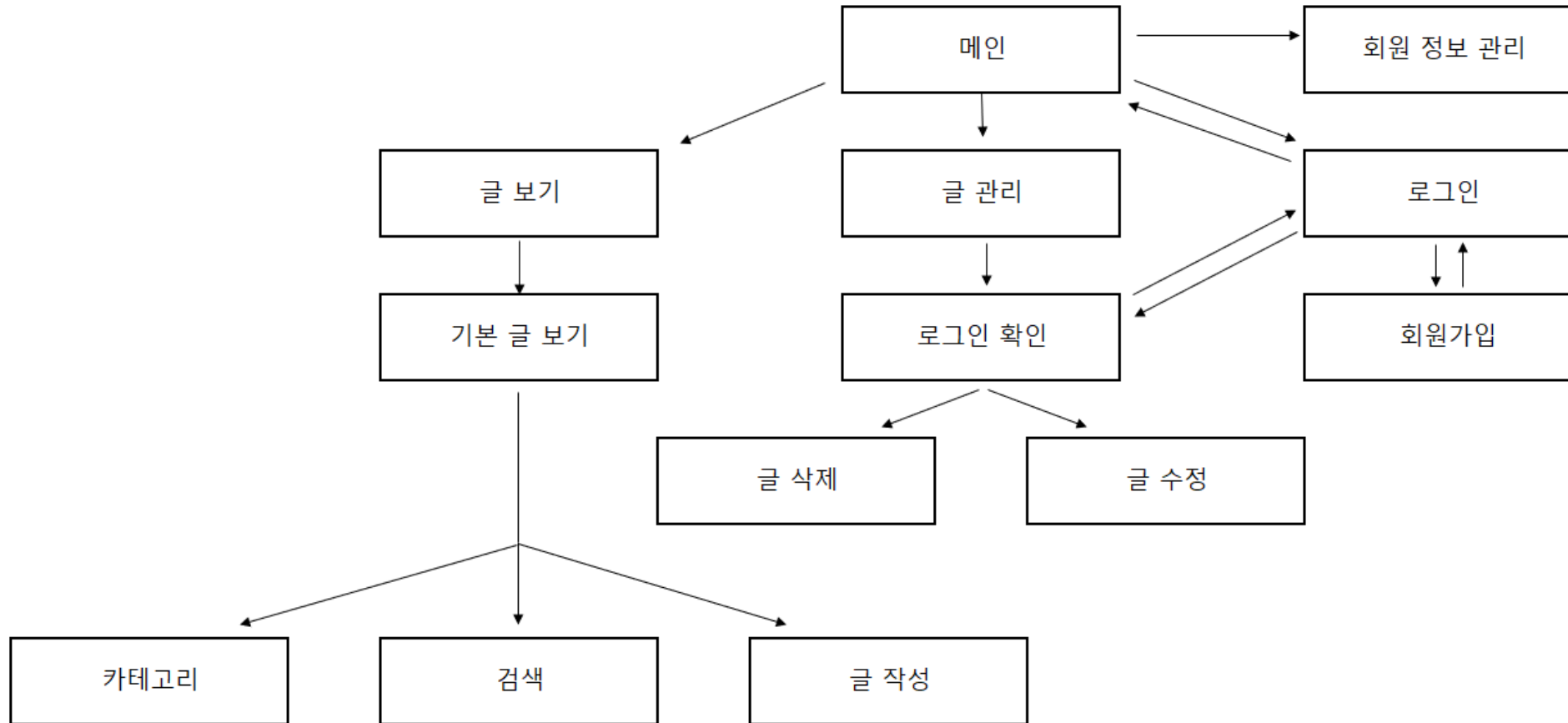
예상 소요 기간

작업 예정일 : 2021년 11월 29일 ~ 2021년 12월 10일

- 약 2주 정도 소요 예상



초기 구상도



Main

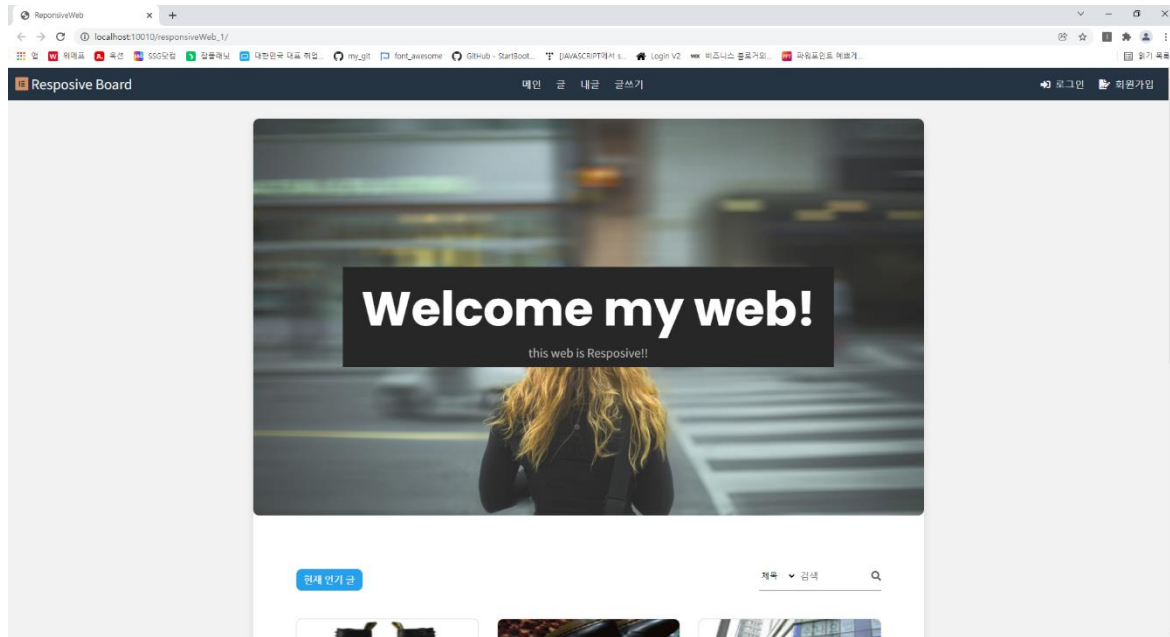
- 첫 시작 페이지.
- nav 메뉴 구현 과 로그인 정보에 따른 로그인 또는 로그아웃과 내정보 기능구현.
- 조회수 를 기준으로 가장 많은 6개의 글을 보여주는 기능
- 검색 기능
- 가장 하단의 더보기 버튼 클릭시 글보기 메뉴로 이동.

Responsive Web

3. 개발 및 구현

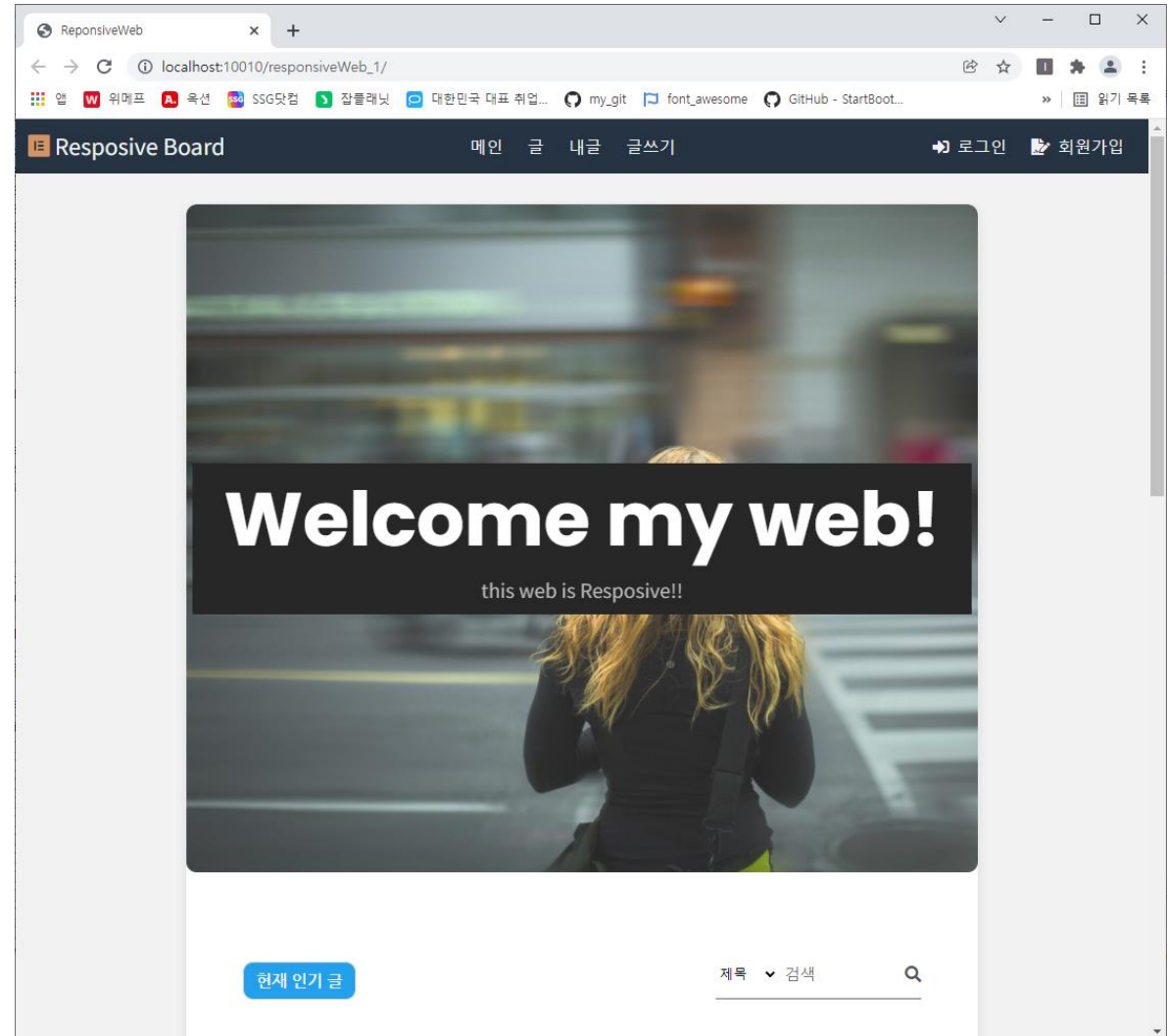
Main

윈도우 크기 최대 -> 1250px 이하.



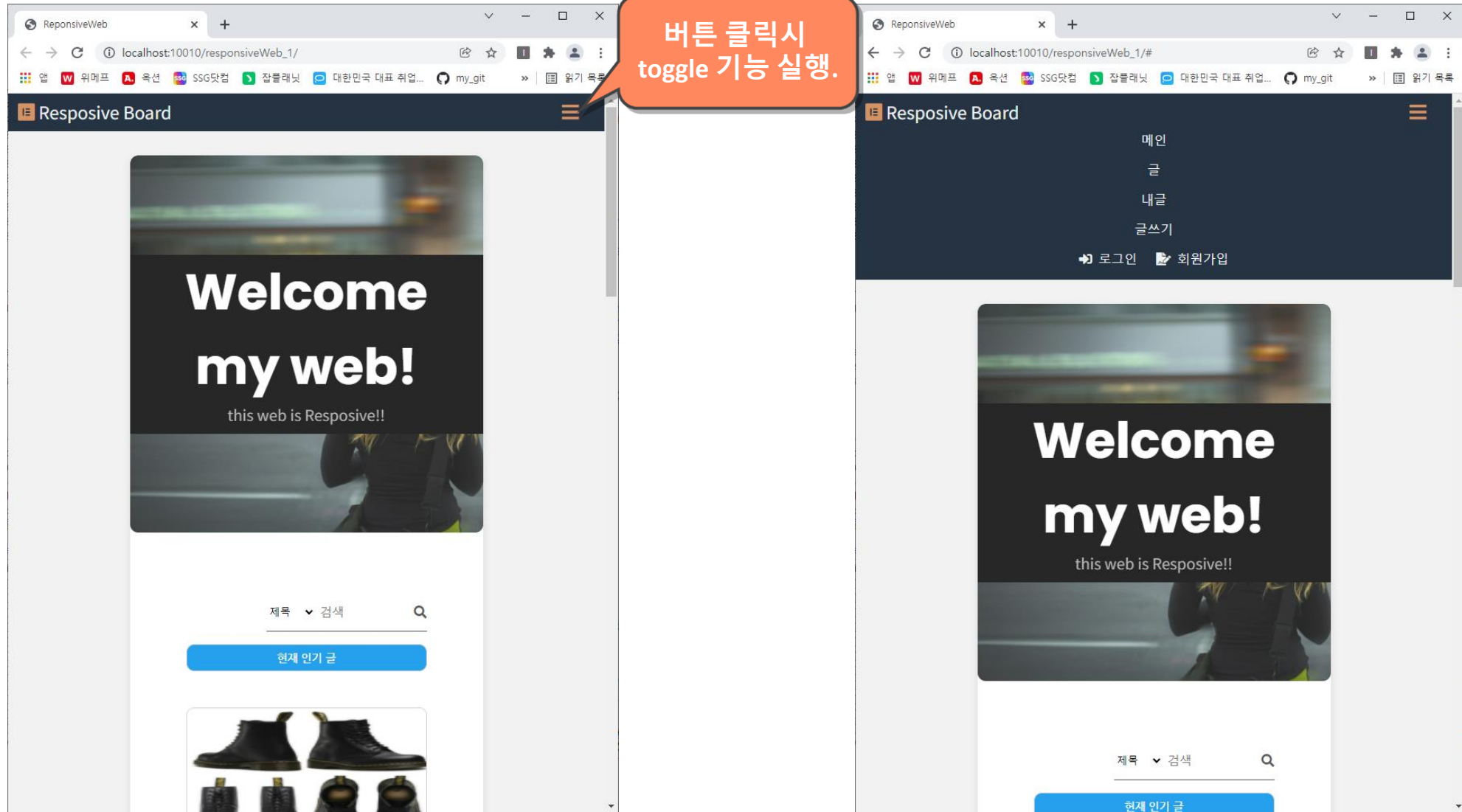
Css 의 **@media** 기능을 이용해 **윈도우** 크기가 **1250px** 또는 **768px** 보다 작아 진다면 전체적인 비율과 크기를조정.

768px 이하시 **javascript** 를 이용해 버튼 클릭시 표시하는 **드롭바** 형태로 조정.



Responsive Web

3. 개발 및 구현



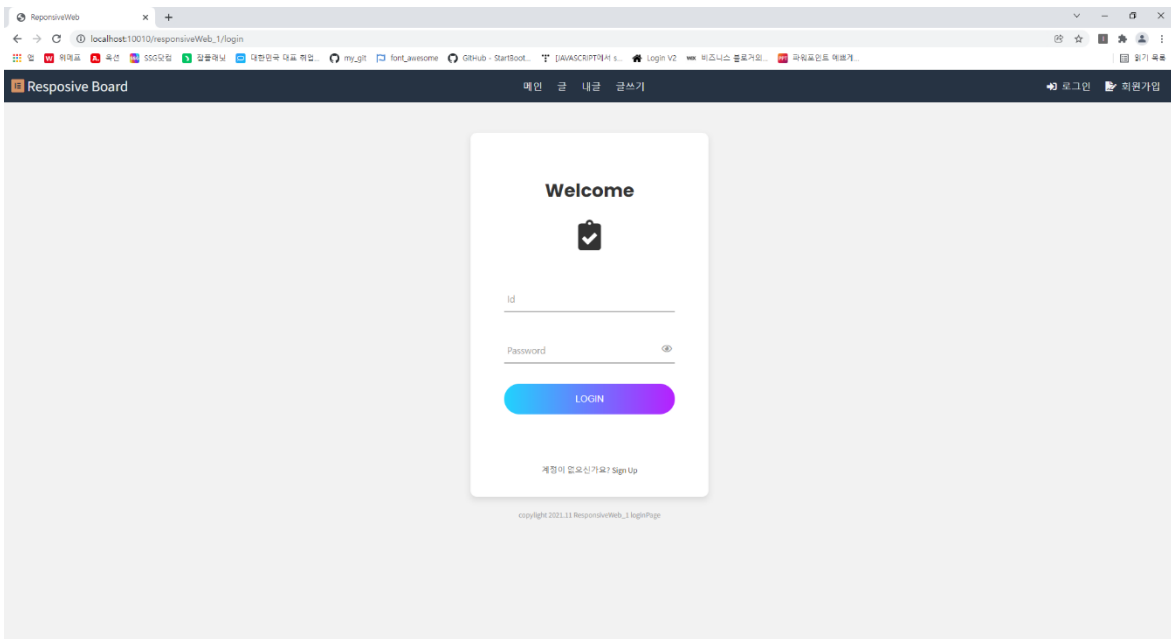
Login & SignUp

- 전체적인 크기를 조정해 두어 어떤 크기의 윈도우라도 호환 되는 사이즈로 설정.
- **js** 파일에서 **blur** 시 **addClass** 와 **removeClass** 와 **css** 의 **transition** 등 을 조정해 동적인 로그인 페이지 구현.
- 하단의 a 태그를 이용해 컨트롤러로 요청 전송후 페이지 이동이 아닌 **toggle** 기능을 응용해 부화 없이 바로 창을 이동 할수 있도록 설정.
- 로그인 또는 회원 가입 **실패시** 입력창 아래 span 태그에 **메시지** 전송.

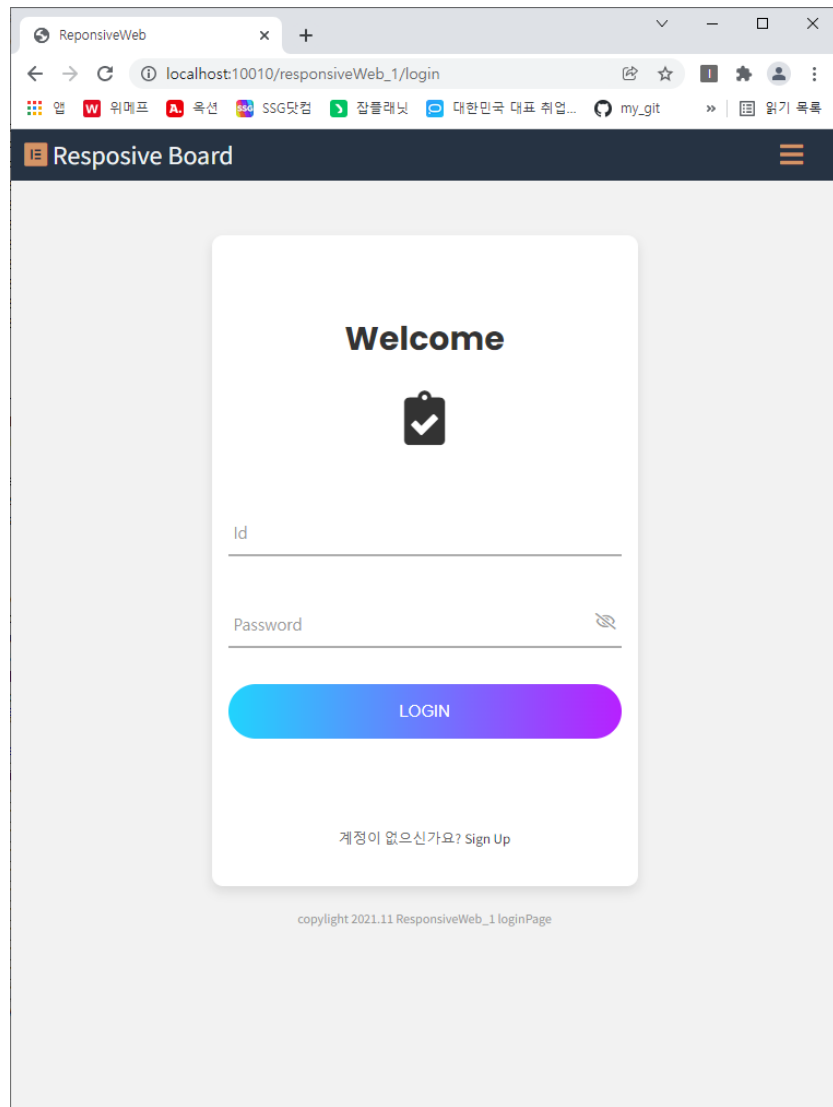
Responsive Web

3. 개발 및 구현

Login



전체적인 크기가 조정되어 있으므로 윈도우의 크기가 **768px** 이하로 내려갈때 **input 태그**의 길이와 **submit 버튼**의 길이를 조정.



Responsive Web

3. 개발 및 구현

Login input 이벤트

Mockup of a login form. It features a 'Welcome' header with a checkmark icon, followed by 'Id' and 'Password' input fields. A 'LOGIN' button is positioned below the fields. At the bottom, there is a link that says '계정이 없으신가요? Sign Up'. The footer contains the text 'copyright 2021.11 ResponsiveWeb_1 loginPage'.

Mockup of a login form. The 'Id' input field is highlighted with a blue and purple gradient border, indicating it is the current focus. The 'Password' field and 'LOGIN' button are visible below it. The footer text is 'copyright 2021.11 ResponsiveWeb_1 loginPage'.

css 의 focus

Mockup of a login form. The 'Id' input field contains the text 'user1' and is highlighted with a blue and purple gradient border, indicating it is the current focus. The 'Password' field and 'LOGIN' button are visible below it. The footer text is 'copyright 2021.11 ResponsiveWeb_1 loginPage'.


Js 의 blur + css 의 focus

Responsive Web

3. 개발 및 구현


Login input 이벤트

Welcome



Id
user1

Password




LOGIN

계정이 없으신가요? Sign Up

copyright 2021.11 ResponsiveWeb_1 loginPage


Toggle 버튼 클릭시

Welcome



Id
user1

Password
0000



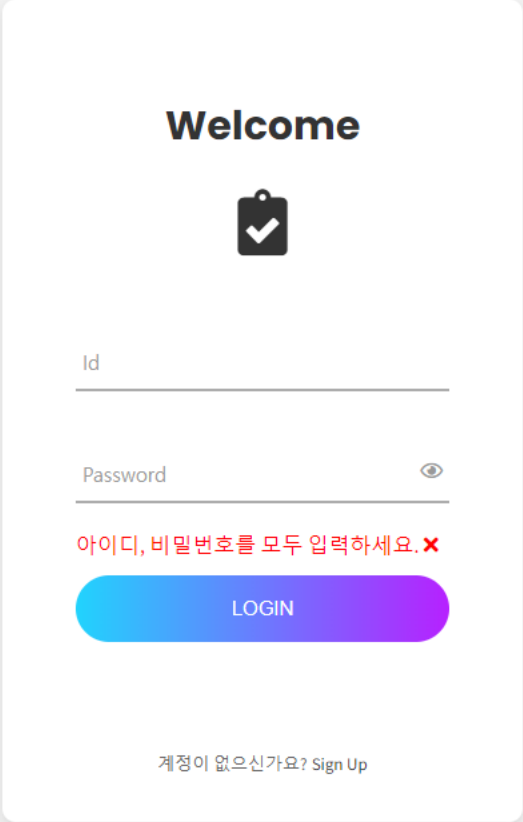
LOGIN

계정이 없으신가요? Sign Up

copyright 2021.11 ResponsiveWeb_1 loginPage

onclick 이벤트
addclass, removeclass 이벤트
비밀번호 표시

Login submit



The image shows a login form with the following elements:

- Welcome** header with a checkmark icon.
- Id** input field.
- Password** input field with a toggle eye icon.
- Red error message: **아이디, 비밀번호를 모두 입력하세요. ✕**
- LOGIN** button with a blue-to-purple gradient.
- Footer text: **계정이 없으신가요? Sign Up**
- Copyright notice at the bottom: **copyright 2021.11 ResponsiveWeb_1 loginPage**

- 로그인 클릭시 form 태그에서 **homecontroller** 로 Id 와 password 등의 데이터를 전송.
- 이후 **DAO 클래스**에서 메소드 실행.
-> **xml** 에서 실행할 **jquery** 문 실행.
- 로그인 정보가 없거나 틀리다면 msg 를 **request** 영역에 전송.
- 로그인 정보가 있다면 **session** 영역에 **로그인 정보** 를 저장 후 메인으로 전송.
- 코드는 com.project.dao 의 **MybatisDAO.java** 와 **user.xml** 체크.

Content

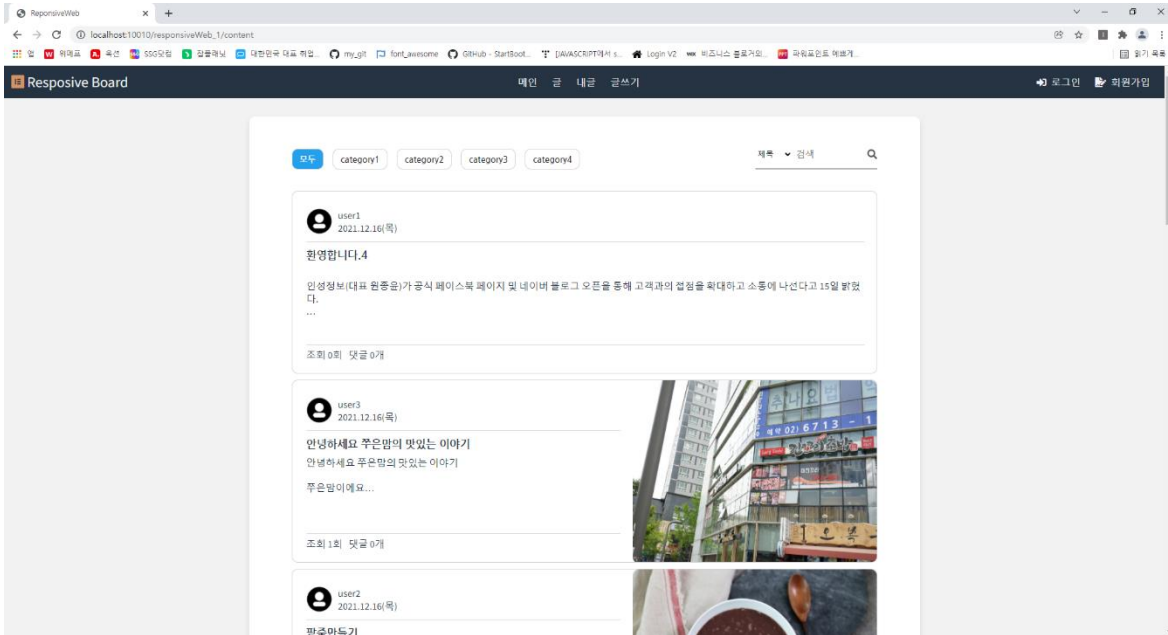
- 컨트롤러에서 요청시 **DB**에서 글들을 얻어와서 **request** 영역에 객체를 저장하여 jsp 에 하나씩 읽어오는 처리를 수행.
- 기본적인 **페이지의 글 개수**는 **10개** 씩으로 지정되어 있지만, 이후 보이는 글의 형태와 크기 등을 조절할 수 있도록 개선 예정.
- 기본적으로는 **모든 글**을 불러오지만, **카테고리** 와 **검색** 기능을 사용해 요청한다면 조금더 많은 양의 request 영역에 저장된 정보를 읽어와야 하므로 이후 작업들에 더 많은 request 요청이 요구됨.
- 하단에 **페이지 이동 버튼** 기능 구현

Responsive Web

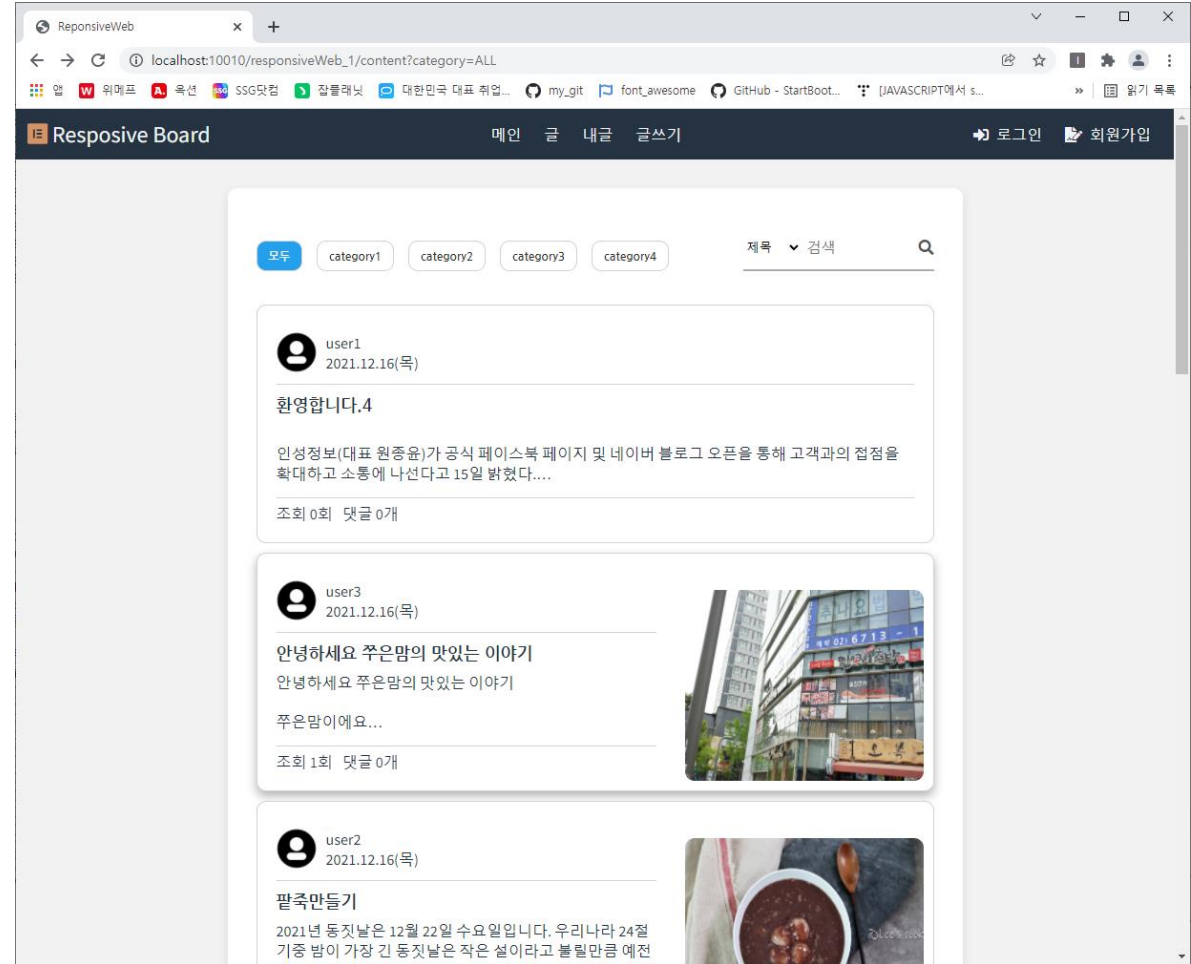
3. 개발 및 구현

Content

윈도우 크기 최대 -> 1250px 이하.



윈도우의 크기가 **1250px, 768px** 이하로 내려갈 때 **전체적인 크기**와 **이미지 크기, 폰트 사이즈** 등을 조정해 **동적인 이미지** 표현.



Responsive Web

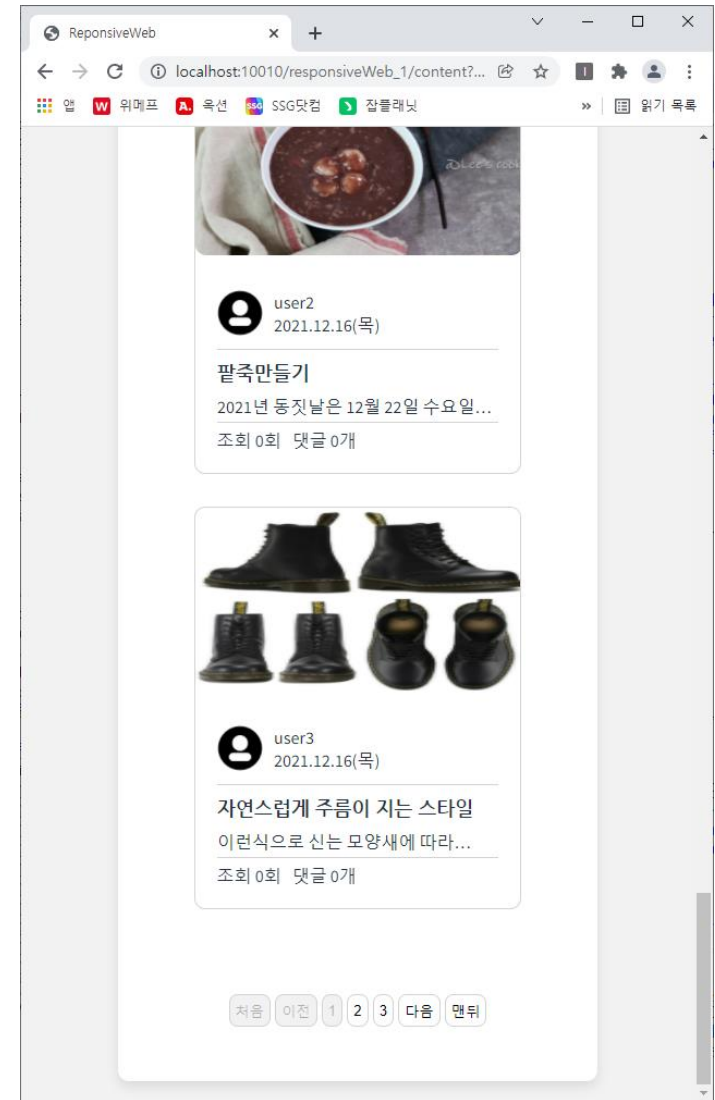
3. 개발 및 구현

Content

윈도우 크기 768px 이하.

- 윈도우의 크기가 1250px, 768px 이하로 내려갈 때 **전체적인 크기**와 **이미지 크기**, **폰트 사이즈** 등을 조정해 **동적인 이미지** 표현.

- 전체적인 크기가 줄어들어 버튼을 모두 표현 하지 못하므로 버튼의 **개수를 조정**



Content 검색 기능

```
<select id="searchCount" parameterType="java.util.HashMap" resultType="int">
    select count(*) from CONTENTS where ${select} LIKE '%' || #{text} || '%' and category like '%' || #{category} || '%'
</select>

<select id="searchContents" parameterType="java.util.HashMap" resultType="contentsVO">
    select * from (
        select rownum rnum, GG.* from (
            select * from CONTENTS where ${select} LIKE '%' || #{text} || '%' and category like '%' || #{category} || '%' order by idx desc
        ) GG
    ) where rnum between #{startNo} and #{endNo}
</select>
```

- 검색 또는 카테고리 선택으로 검색 요청시 컨트롤러에서 request 영역의 값을 읽어와 content 와 비슷하지만 조금더 여러 정보를 넣어 검색.
- 카테고리 기능과 검색 기능이 서로 다르게 적용되는 문제가 있어 여러 가지 구현을 해보았고, 최종적으로 Like % % 문으로 서로 동시에 요청을 처리하는 방법으로 구현되었다.
- 검색후 카테고리 설정 또는 카테고리 선택후 그 안에서 검색하는 등의 복합적인 기능도 수행 가능.

ContentView

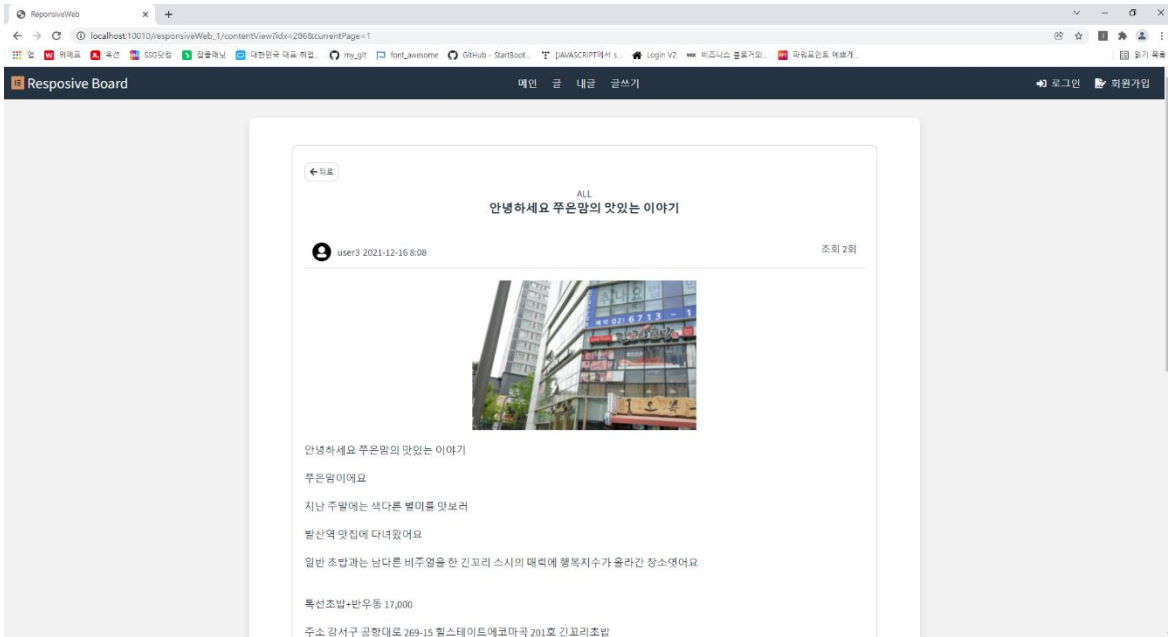
- 컨트롤러에서 해당 글에 해당하는 **idx**를 받아오고, 이후 **idx**에 해당하는 **글 한건**과 **댓글들**을 DB에서 얻어와 **request 영역**에 **저장**하고, jsp 에서 해당 글의 정보를 **표시**한다.
- 글 한건을 읽어올 때 먼저 **조회수**를 올리는 **update sql**을 수행한 이후 해당글을 읽어오는 작업이 요구된다.
- 글 한건을 읽어와서 보여줄 때 글 한건의 **모든 정보**를 읽어오고 **이전 페이지**로 돌아갈 수 있도록 여러 정보를 **request 영역**에 저장하므로 이후 작업들에 그 정보들을 계속해서 넣어줘야 되는 불편함이 있다.
- **댓글 달기 기능**은 로그인 이 되어 있다면 **활성화** 되고 로그인이 되어 있지 않다면 **비활성화** 된다.
- 댓글에 댓글을 다는 **대댓글 기능**을 구현 하기 위해 초기 comment 테이블에서 여러 추가사항을 적용했다.

Responsive Web

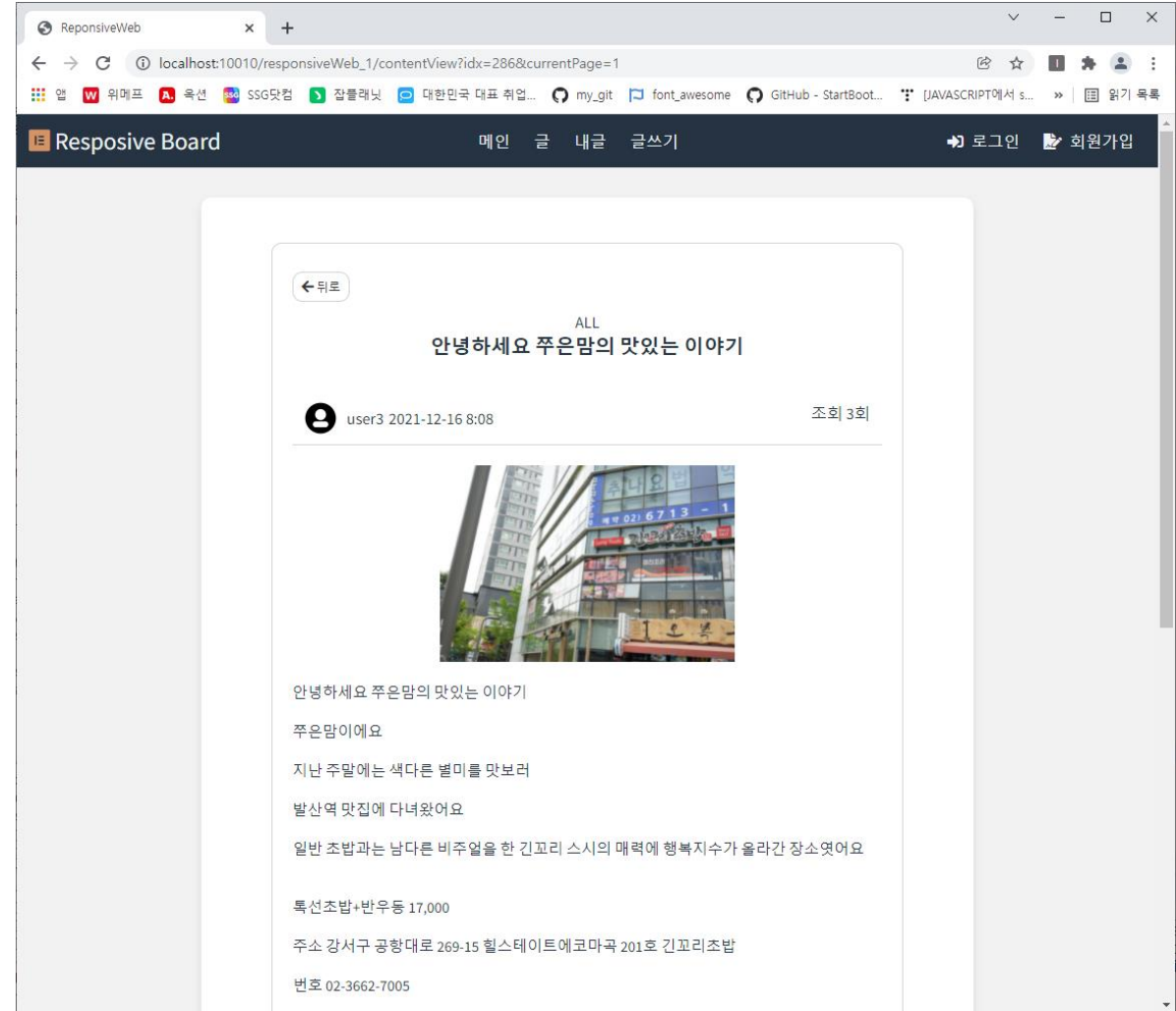
3. 개발 및 구현

ContentView

윈도우 크기 최대 -> 1250px 이하.



윈도우의 크기가 **1250px, 768px** 이하로 내려갈 때 **전체적인 크기**와 **이미지 크기, 폰트 사이즈** 등을 조정해 적당한 사이즈를 맞추었다.

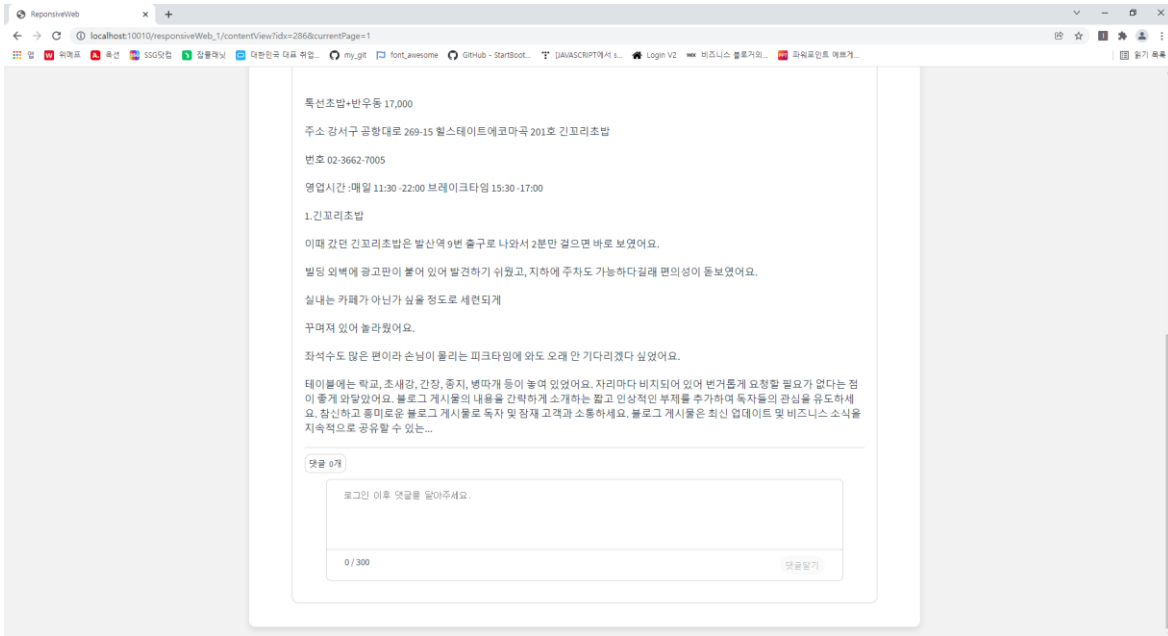


Responsive Web

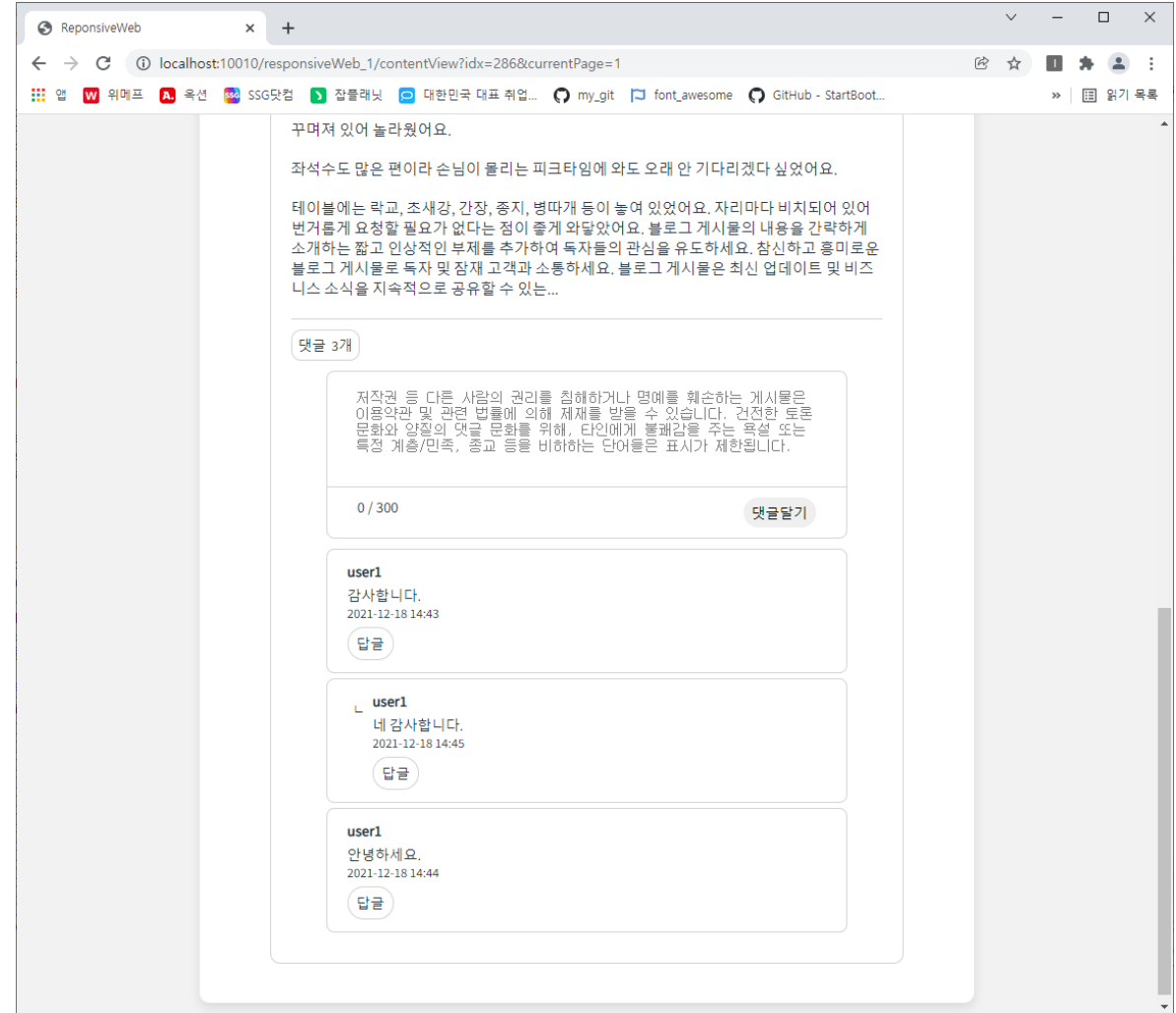
3. 개발 및 구현

ContentView

윈도우 크기 최대 -> 1250px 이하.



윈도우의 크기가 **1250px, 768px** 이하로 내려갈 때 **전체적인 크기**와 **이미지 크기, 폰트 사이즈** 등을 조정해 적당한 사이즈를 맞추었다.



UserStatus

- DB의 **User 테이블**에서 session 에 저장된 id 와 비밀번호 와 일치하는 데이터를 **select sql**문을 사용해 **유저 정보**를 얻어온다.
- 아이디와 비밀번호 에 **Lock**을 걸어 부분적으로 감춘다.
- 이미지와 닉네임, 비밀번호 를 수정할 수 있도록 **수정 페이지**로 이동을 요청한다.
- 닉네임 과 이미지는 USER 테이블 이외에도 **contents** 와 **comments** 에서 도 사용되므로 USER 테이블에서 **변경하기 전** 원래 닉네임으로 설정된 모든 글과 댓글을 불러오고 변경하려는 닉네임과 이미지로 먼저 **update** 를 한후, USER 정보를 **update** 하는 순서가 필요하다.

매우 중요

Responsive Web

3. 개발 및 구현

UserStatus 닉네임 수정

```
AbstractApplicationContext ctx = new GenericXmlApplicationContext("classpath:applicationCTX.xml");
UserVO userVO = ctx.getBean("userVO", UserVO.class);
userVO = mapper.selectById(id);

String update_nickname = request.getParameter("update_nickname").trim();
String password_check = request.getParameter("password_check").trim();

if(update_nickname == null || update_nickname.equals("") ||
    password_check == null || password_check.equals("")) {
    model.addAttribute("msg", "닉네임 또는 비밀번호를 모두 입력하세요. <i class=\"fas fa-times\"></i>");
    return "updateNickname";
} else if(!password.equals(password_check)){
    model.addAttribute("msg", "현재 비밀번호가 일치하지 않습니다. <i class=\"fas fa-times\"></i>");
    return "updateNickname";
} else if (nickname.equals(update_nickname)) {
    model.addAttribute("msg", "현재 닉네임과 동일합니다. <i class=\"fas fa-times\"></i>");
    return "updateNickname";
}

HashMap<String, Object> hmap = new HashMap<String, Object>();
hmap.put("id", id);
hmap.put("nickname", nickname);
hmap.put("update_nickname", update_nickname);

int count = mapper.selectNicknameCount(hmap);

if(count > 0) {
    model.addAttribute("msg", "중복된 닉네임이 존재합니다. <i class=\"fas fa-times\"></i>");
    return "updateNickname";
}
}
```

nickname



현재 닉네임

user1

새로운 닉네임

비밀번호




닉네임 수정

UserStatus 닉네임 수정

```
mapper.updateContentNickname(hmap);  
mapper.updateCommentNickname(hmap);  
  
mapper.updateNickname(hmap);  
session.setAttribute("nickname", update_nickname);  
model.addAttribute("pass", "닉네임 변경이 완료 되었습니다.");  
  
return "updateNickname";
```


- 닉네임이 현재와 **동일** 하거나, **중복된 닉네임**일 경우 변경할 수 없다.
- 중복 되지 않는 닉네임의 경우 **이전 닉네임**으로 작성한 글과 댓글을 모두 **변경하려는 닉네임**으로 교체하는 **update sql** 을 사용해 변경해 준다.
- 이후 USER 테이블의 내 정보를 변경해주고 **session**에 변경된 **새로운 nickname**으로 수정해 준다.

nickname



현재 닉네임
user1

새로운 닉네임

비밀번호 


닉네임 수정

copyright 2021.11 ResponsiveWeb_1 loginPage

UserStatus 비밀번호 수정


- 현재 비밀번호가 session 에 저장된 비밀번호와 일치하지 않거나,
새로운 비밀번호가 **현재 비밀번호**와 **동일** 하거나,
새로운 비밀번호가 **비밀번호 체크**와 동일 하지 않으면 변경할 수 없다.
- 모든 조건을 통과한 **비밀번호**의 경우 **update sql** 을 사용해
변경 해준다. 이후 USER 테이블의 내 정보를 변경해주고 **session**에 변경된
새로운 password로 수정해 준다.

password



현재 비밀번호

새로운 비밀번호

비밀번호 체크 

비밀번호 수정

copyright 2021.11 ResponsiveWeb_1 loginPage

작업하며

처음 작업 예상 시간보다 더 늦춰지긴 했지만 여러 자잘한 수치등과 방향성 등을 수정하며 처음 적었던 코드들의 문제점 등을 파악하며 수정 하다 보니 부족했던 부분과 발전할 수 있는 부분들을 파악할 수 있어 재밌게 작업 했습니다.

학원에 공부 하는 시간 만큼 집중적으로 작업하며, 주말도 작업할 수 있었다면 기간 안에 더 나은 작업을 할 수 있었을 것 같은 마음이 있습니다.

하지만 이것저것 적용해보고 다 작성해 놓았던 코드들을 전부 지우고 다시 적용해보고, 하는 작업들에 힘들기도 했지만 새로운 것 도 많이 알았고 재밌었던 것 같습니다.

개선 사항

블로그를 메인으로 잡고 작업하다 보니 이미지와 형태등을 중점적으로 파악하고 작업 했어야 했지만, 이미지를 적용하는 부분에 있어 미흡했던 것 같아 부족하다는 생각이 많이 들었고, 이부분에 대해 조금씩 배우며 개선해야 된다는 걸 알게 되었습니다.

그외로 CSS 적용을 위해 class 를 부분별로 다 다르게 적용해 중복 되는 속성 들이 생각보다 많았고, 따로 만들어야 하는 코드들도 부분적으로 많이 보게 되었습니다. 이러한 코드들은 1개의 class 가 아닌 2개 이상의 class 들 또는 id 로 구성해 중복되는 것은 첫번째로, 중복되면 안되는 것들은 id 나 다른 class 로 적용 하는 작업이 필요하да걸 알게되었습니다.

Jsp 에서 영역을 나눌때 의미없는 <div> 태그들 보다는 의미있는 태그들을 사용해서 작업하는 것이 더 괜찮다는걸 배웠고, 그에 따라 작업했던 코드중 div 보다는 더 나은 태그들을 적용하는 것 이 필요하다는 것 또한 알게되었습니다.

개선 사항

현역에서 바로바로 사용되는 디자인 에 대해서 현재 무지한 상태이기에 괜찬아 보이는 디자인 보다는 내가 하고 싶은 디자인을 채택하고 한것 같아 나중에 천천히 모양과 디자인 , 색감등을 수정해서 보완하는 작업이 필요합니다.

그 외에도 여러 문제가 있지만 하나하나 적기엔 너무 부족한 작업이라 천천히 고쳐나가려 합니다.

감사합니다.

Responsive Web PORTFOLIO END

https://github.com/Kimjunyoung111/ResponsiveWeb_project