7주차 진행 보고서 세부 사항

2021.04.01~ 2021.04.14 B71116장혜리

코드는 아래 gitnub에서 확인 가능합니다.

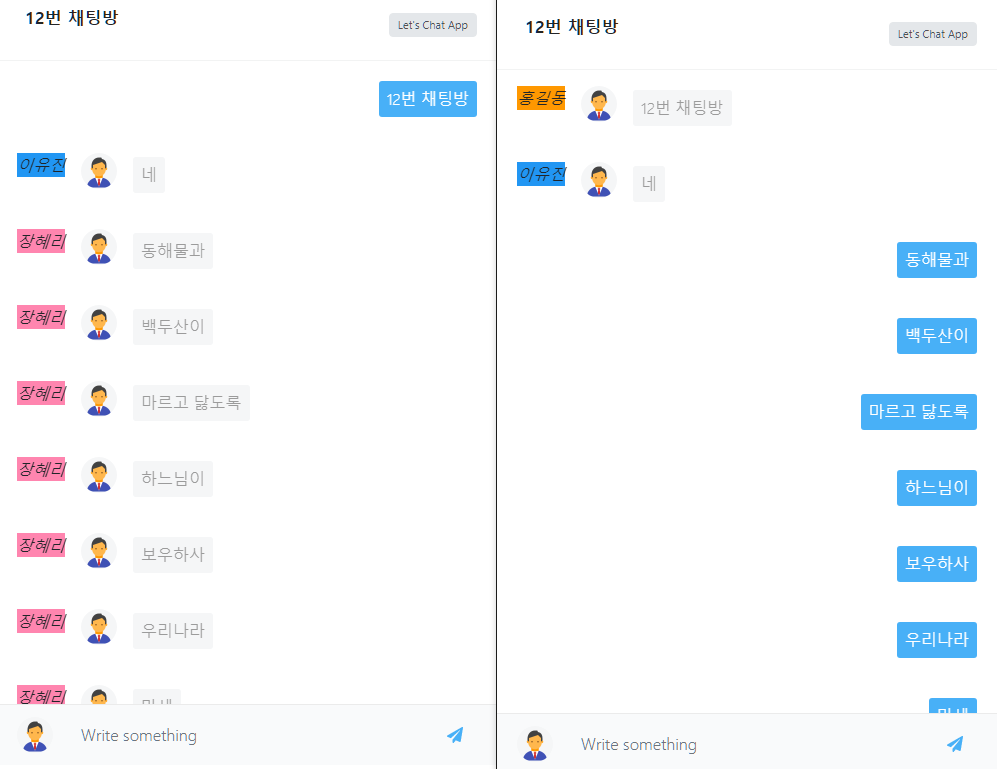
[**[github]MBTI-Lovers 안드로이드**](https://github.com/hyerijang/MBTI-Lovers)

[**[github]MBTI-Lovers-webServer**](https://github.com/hyerijang/MBTI-Lovers-webServer)

# 채팅 구현

## 채팅내역 Firebase realtime DB에 저장

* firebase의 Realtime Database를 이용하여 채팅 기록을 저장하였습니다.
* 서버에서는 소켓 통신으로 전달받은 메시지를 JSON형식으로 저장합니다.
* NoSQL인 파이어베이스는 하나의 채팅방 관련 정보들을 하나의 document로 관리 할 수 있고, 데이터를 다른 객체로 변환할 필요 없이 JSON으로 저장하기 때문에 RDB에서 read 할때보다 속도가 빠릅니다.





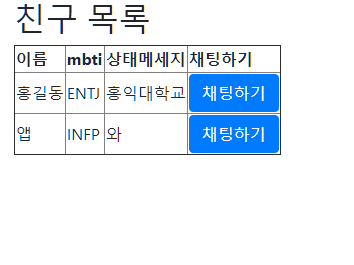
# 채팅 방 및 친구목록 구현

## 친구목록 구현 및 채팅방과 연결

유저 목록에서 친구신청을 하면 친구 목록에 친구가 추가됩니다.

## 채팅 방 생성 및 조회 기능 추가

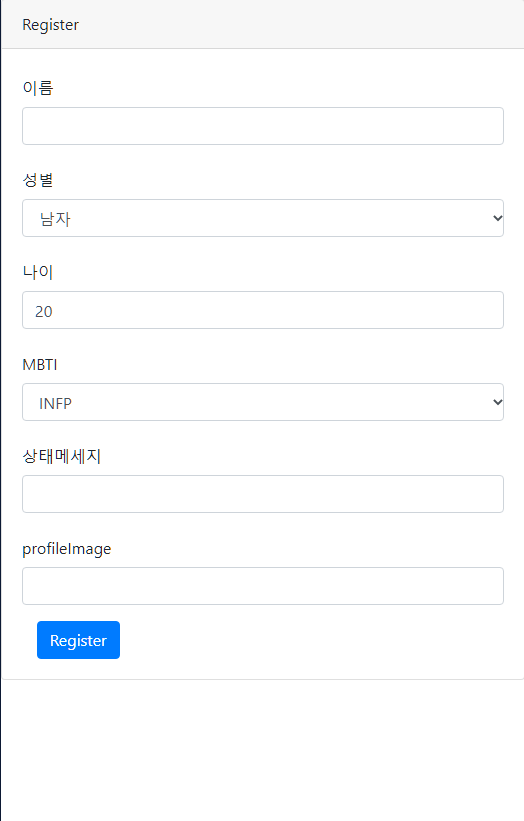
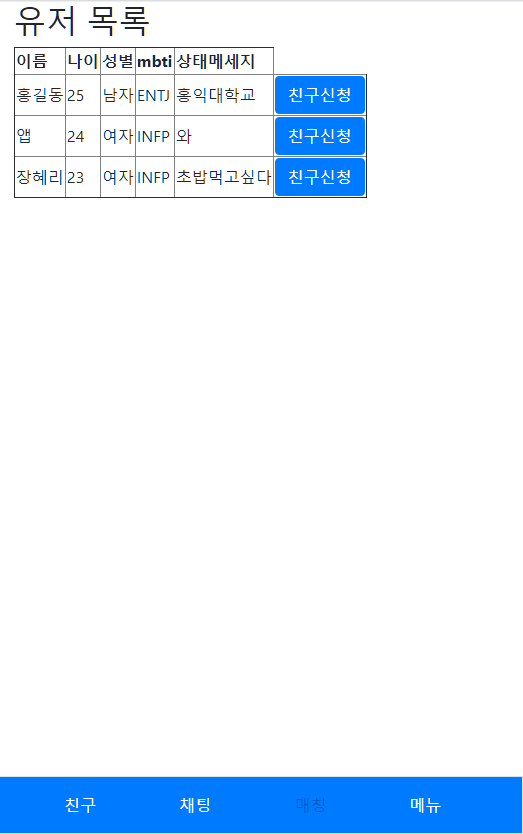
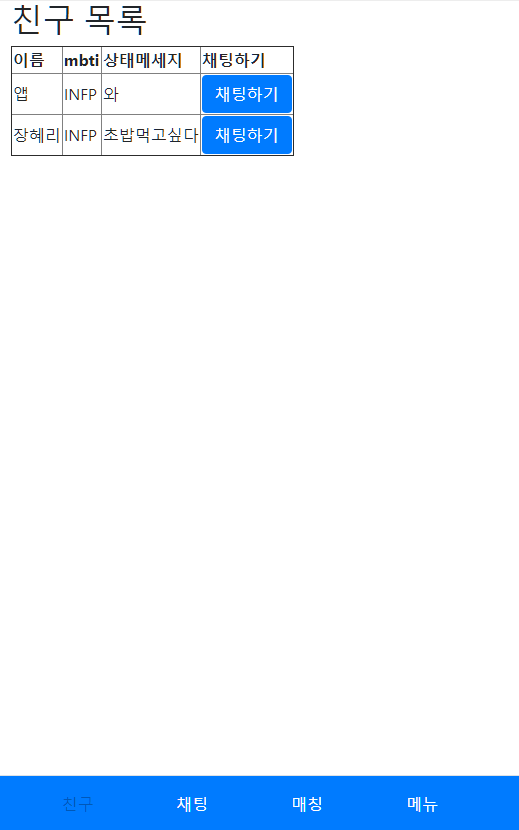
친구 목록을 통해 채팅방에 입장할 수 있습니다.



# 레이아웃 수정

## 전체적인 UX/UI 수정

부트스트랩을 이용하여 임시적으로 화면 UX/UX를 구성하였습니다. 개발 시 편의를 위해 노출시킨 유저 정보는 추후 수정할 예정입니다.



# 로그인 및 인증처리

## 로그인 시 유저 쿠키 저장

로그인 시 유저의 쿠키 정보를 저장하여 재접속, 기타 서비스 이용 시 쿠키에 저장된 UID정보를 활용할 수 있습니다.

## 로그아웃 기능 추가

홈 화면에서 로그아웃 버튼을 누를 시 유저 쿠키 및 세션을 삭제하고 로그아웃 됩니다.

@RequestMapping*(*value = "/logout", method = RequestMethod.*GET)*public String logout*(*HttpServletRequest request,  
 HttpServletResponse response,  
 HttpSession session*)* throws Exception *{* // session에 저장된 login 값  
 Object object = session.getAttribute*(USER\_SESSION)*;  
 if *(*object != null*) {* Member userVO = *(*Member*)* object;  
 // session 정보 삭제  
 session.removeAttribute*(USER\_SESSION)*;  
 // session 초기화  
 session.invalidate*()*;  
 // 로그인 쿠키값  
 Cookie loginCookie = WebUtils.*getCookie(*request, *USER\_COOKIE)*;  
 if *(*loginCookie != null*) {* loginCookie.setPath*(*"/"*)*;  
 // 쿠키 유효기간 0  
 loginCookie.setMaxAge*(*0*)*;  
 // 쿠키 저장  
 response.addCookie*(*loginCookie*)*;  
 *}  
 }* return "redirect:/";  
*}*

## 신규 유저인 경우 유저 프로필 생성 페이지로 이동

기존 프로그램은 유저가 로그인 후 수동으로 유저 프로필 생성 페이지로 이동해야했습니다. 인증 인터셉터와 로그인 인터셉터를 수정하여 신규 유저인 경우 자동으로 유저 프로필 생성 페이지로 이동하게 되었습니다.

기존 유저와 달리 신규유저 정보는 쿠키에 저장되지 않고 세션에 한시적으로 저장되기 때문에 앱를 종료하면 그대로 삭제됩니다. 따라서 회원가입 도중 앱이 종료되더라도 다시 신규회원가입 페이지로 연결됩니다.

### AuthInterceptor.java

@Override  
public boolean preHandle*(*HttpServletRequest request,  
 HttpServletResponse response,  
 Object handler*)* throws Exception *{* HttpSession session = request.getSession*()*;  
  
 String newUserUid = *(*String*)* session.getAttribute*(*LoginController.*NewUserUid)*;  
 if *(*newUserUid != null*) {* Optional*<*Member*>* member = memberService.findOneByUid*(*newUserUid*)*;  
  
 if *(*member.isPresent*()) {* session.setAttribute*(*LoginController.*USER\_SESSION*, member.get*())*;  
 session.removeAttribute*(*LoginController.*NewUserUid)*;  
 *}  
  
 }* if *(*session.getAttribute*(*LoginController.*USER\_SESSION)* == null*) {* // 현재 페이지 저장  
 saveDestination*(*request*)*;  
  
 // 쿠키 유무 확인  
 Cookie loginCookie = WebUtils.*getCookie(*request, LoginController.*USER\_COOKIE)*;  
 if *(*loginCookie != null*) {* Optional*<*Member*>* member = memberService.findOneByUid*(*loginCookie.getValue*())*;  
 if *(*member.isPresent*()) {* session.setAttribute*(*LoginController.*USER\_SESSION*, member.get*())*;  
 return true;  
 *}* else *{* //존재하지 않는 멤버  
 //로그인 쿠키 삭제  
 *logger*.info*(*"존재하지 않는 멤버는 로그인 할 수 없습니다."*)*;  
 loginCookie.setMaxAge*(*0*)*;  
 *}  
 }* response.sendRedirect*(*"/user/login"*)*;  
 return false;  
 *}* return true;  
*}*

### LoginInterceptor.java

//로그인 처리  
@RequestMapping*(*value = "/loginPost", method = RequestMethod.*POST)*public void loginPost*(*LoginVO loginVO, HttpSession session, Model model*) {* //신규 유저인 경우 loginVO에 신규유저 uid 넣음  
 String newUseruid = *(*String*)* session.getAttribute*(NewUserUid)*;  
 if *(*newUseruid != null*)* loginVO.setUid*(*newUseruid*)*;  
  
 Optional*<*Member*>* member = loginService.login*(*loginVO*)*;  
  
 if *(*member.isPresent*()) {* //기존 유저면 접속시간 갱신  
 memberService.editLastConnectTime*(*member.get*())*;  
 //model에 멤버 객체를 currentUser라는 이름의 변수에 저장  
 model.addAttribute*(*"currentUser", member.get*())*;  
 *}* else *{* //신규 가입  
 session.setAttribute*(NewUserUid*, loginVO.getUid*())*;  
 *}  
  
}*

## 참조

[# Spring MVC 게시판 예제 16 - HttpSession을 이용하는 로그인 처리](https://doublesprogramming.tistory.com/211)

[JavaScript LocalStorage 사용 방법과 쿠기와 차이점](JavaScript%20LocalStorage%20사용%20방법과%20쿠기와%20차이점)

[Spring Boot에서 Interceptor 사용하기 | MHLab Blog](https://elfinlas.github.io/2017/12/28/SpringBootInterceptor/)

[부트스트랩4의 메뉴 (Navigation) : 네이버 블로그](https://m.blog.naver.com/PostView.nhn?blogId=pjh445&logNo=221159818328&proxyReferer=https:%2F%2Fwww.google.co.kr%2F)