

2020-2 캡스톤 디자인

<최종보고서>



Professor:

박상오 교수님

Team Name:

20144727 장동재

20161863 김지혜

20172051 강유경

1. 팀 소개

- 팀 이름 : 10조
- 프로젝트 주제 : 소규모 화상 취미 클래스
- 이름 : We Do With You

2. 프로젝트 소개

A. 개발 배경

뉴스 > 경제 > 금융

주52시간 근무제後... 직장인 소비시간대 2시간 앞당겨졌다

남건우 기자 입력 2019-06-18 03:00 수정 2019-06-18 10:35



‘워라벨 업종’ 카드결제 분석해보니

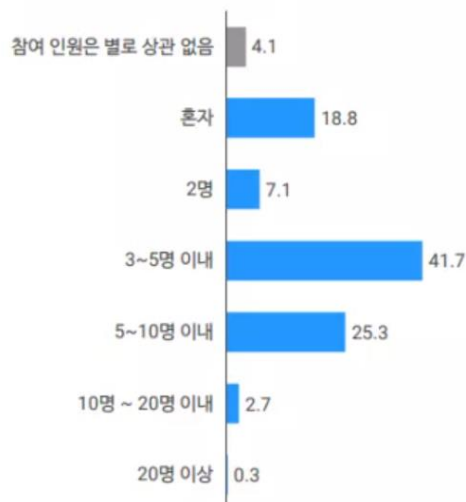


<http://www.donga.com/news/article/all/20190617/96038604/1>

'코로나 집콕' 30·40대로 미어터진다...온라인 '취미 플랫폼'

입력 2020.04.11 10:31 | 수정 2020.04.11 22:33

향후 유료로 취미생활/자기계발 분야 참여시 선호하는 인원



최근 주 52시간제 도입으로 인한 워라벨 문화, 여러 개의 직업을 갖는 'N잡' 트렌드 등으로 인해 취

미 활동이 강화되는 시대가 도래했다. 게다가 코로나라는 사태로 인해, 늘어난 여가시간을 집에서 새로운 취미로 보내는 사람들 역시 늘어나고 있는 추세이다. 이에 따라 원데이 클래스, 온라인 클래스 등이 많이 열리고 있다. 하지만 그만큼 온라인 클래스의 필요성이 늘어난 데에 비해, 이용자가 적극적으로 수업에 참여할 수 있는 참여형 강의는 찾아보기 쉽지 않다. 또한, 설문조사 업체 오픈서베이의 2019년도 발표에 따르면, 향후 유료 취미 생활 클래스의 선호 인원을 개인이나 단체보다는 소규모 집단을 선호한다는 발표가 있다. 이에 아이디어를 얻어, '화상'으로 '소규모' 인원의 학생들과 '함께' 취미 클래스를 여는 서비스를 만들어보기로 결정했다.

B. 프로젝트 소개

프로젝트 WDWY는 화상으로 소규모 인원의 학생들이 취미 클래스를 수강하고, 학생들 간 소통하며 보다 제대로, 즐겁게 취미 활동시간을 보낼 수 있는 시스템이다. 접근성을 고려하여 앱 서비스를 개발할 예정이다.

강의는 기존에 존재하는 화상 채팅 플랫폼을 이용할 계획이다. 이미 충분히 괜찮은 서비스(Google Hangout)가 제공되고 있기 때문에, 우리가 그 서비스보다 차별화된 화상 채팅 시스템을 만들 필요는 없다는 판단이 들었다. 강의 시간 전에 화상 채팅 프로그램 행아웃의 링크가 제공되고, 그 링크를 통해 바로 강의에 참여할 수 있다. 이는 우리학교 eclass learning x 대시보드의 zoom 카테고리에서 아이디어를 얻었다.

이 서비스는 '화상'과 '소규모'에 중점을 둔 만큼, 수강생들의 적극적인 참여를 유도한다. 특히, 소규모 수업이기에, 학생들 간의 교류를 통해 보다 클래스의 효율성을 극대화할 수 있게끔 도운다. 화상으로 모두 수업에 참여하고, 과목에 따라 서로 화면을 공유하는 등 학생들이 수업을 얼마나 잘 따라오고, 어떻게 하고 있는지를 확인할 수 있다. 정규 클래스 시간 외에도 소모임 등을 통해 서로 어떻게 배운 바를 활용하는지 확인하고 소통하며 서로에게 도움을 줄 수 있다.

수강자들 간의 교류를 더 이끌어내고자 소모임 활동을 적극 권장할 계획이다. 서비스 내에 소모임 일정을 정할 수 있는 시스템을 줄 예정이다. 또한, 보다 원활한 교류를 위해, 강의자와 이용자 모두 이용 가능한 커뮤니티를 만들어 학생들 간 의견이나 자료 공유를 편리하게 도와줄 것이다.

이 서비스는 수강자뿐만 아니라 강의자에게도 보다 편리하게 강의를 진행할 수 있는 서비스를 제공한다. 강의자들은 웹 사이트를 통해 클래스를 개설하고 관리할 수 있는 권한이 주어진다. 수업 시간에 학생 출석 체크가 가능하며, 신규인지 기존 학생인지 확인도 가능하다. 또한 강의 계획, 공지사항과 자료 등을 올릴 수 있는 강의자 전용 게시판이 주어진다.

강의를 수강한 이용자는 해당 강의에 대한 피드백을 남길 수 있으며, 이 평가는 다른 이용자들이 강의를 선택하는 데에 도움을 주게 될 것이다.

C. 프로젝트 목표

이 프로젝트의 목표는 이용자가 알찬 여가시간을 보낼 수 있도록 도와주는 것이다. 이 목표는 크게 두 가지로 나뉜다. 첫 번째로는, 수강자가 만족스러운 수업을 할 수 있도록 하는 것이다. 이 만족에는 금전적인 만족 역시 포함된다. 많은 유료 온라인 클래스들 중에, 좀 더 적극적인 참여와 교류로 인한 만족감과 실력 상승을 이끌어주어 돈이 아깝지 않은, 가치 있는 서비스를 제공하고 싶다. 만족스러운 수강으로 알찬 여가시간을 보내고, 취미생활을 즐겁게 해주는 것이 궁극적인 목표이다.

두 번째로는, 강의자 역시 만족스러운 수업을 할 수 있도록 도와주는 것이다.

조선비즈

"임금은 줄고 시간은 늘고"... 코로나에 직장인들 투잡 대신 '1.5잡' 뜬다

직장인들 사이에 '1.5잡'이 대세다. 주52시간제 이후 유연근무제가 도입돼 근무시간이 줄어든 데다 올해 코로나19 사태가 겹치면서 임금까지 감소 ...
3일 전



이 기사에서 알 수 있듯이, 많은 사람들이 새로운 취미생활을 배우기만 할 뿐만 아니라, 자신이 자신 있어 하는 분야를 집에서 매체를 통해 가르치기도 한다. 새로운 직업을 갖는 것은 쉬운 일이 아닌 만큼, 제대로 된 동기와 보상이 필요하다. 이 서비스를 통해 가르치는 입장에서도 피드백을 받아 스스로에게도 도움이 될 수 있고, 학생들의 참여 덕에 기존의 다른 한방향 클래스보다 더 적극적으로 수업에 임하게 될 수 있다. 결국 강의자도 더 만족스럽고 뿌듯하게 여가시간을 보낼 수 있게 되는 것이다.

3. 개발 및 구현 내용

A. 개발 환경



We Do With You 은 Android를 지원하는 앱으로 개발될 예정이다. 따라서 가장 보편적으로 쓰이는 Android Studio를 사용하기로 하였다.

APA의 경우 제품명, 제품유형, 가격, 제품사진, 사이트주소 등의 정보를 각 상품별로 저장할 공간이 필요하므로 이를 위해 MySQL을 사용하기로 하였으며, MySQL과 Android Studio간의 연결을 위해 Filezilla 서버와 php를 사용하기로 하였다.

B. 화상 회의 프로그램 선택 기준

행아웃, 스카이프, ZOOM 3가지 프로그램을 비교한 뒤 가장 적합하다고 판단되는 프로그램을 사용하기로 하였다.

행아웃 : 어플 이용시에는 배터리 소모가 심하다는 단점이 있지만, 계정 없이, 10명까지 무료로 회의에 참가할 수 있다. 또한, 구글 캘린더와 연동이 되어 '화상회의 추가' 버튼만 누르면 행아웃 접속 링크가 생성된다.

스카이프 : 참가자 모두 스카이프 계정이 있어야하며, 이용자가 많아 안정성 이슈(불안정한 연결)가 있다. 또한, 보안 강화와 첨부 용량 증가 시 별도로 요금이 부과된다.

ZOOM : 회의 초대자 외에는 줌 계정이 없어도, 링크만으로 바로 회의 참가 가능하지만, 1:1 이외의 회의는 40분 이후 별도로 요금이 부과된다는 단점이 있다.

=> Google Hangout은 무료로 사용할 수 있고, 간단하게 입장할 수 있고, 회의 추가 기능으로 소모임 활동 역시 편리하게 할 수 있다. 따라서 Google Hangout이 소규모 화상 클래스에 가장 적합하다고 판단하였다.

4. 차별성 및 기대 효과

현재 시중에 나와있는 대표적인 온라인 교육 플랫폼은 크게 세가지가 있다.

1. 클래스101
2. 솜씨당
3. 탈잉

CLASS101

첫째, '클래스101'은 국내 최고규모의 온라인 교육 플랫폼이다. 그만큼 많은 종류의 강의를 있지만 거의 모든 강의를 녹화 강의로 진행되기 때문에 우리가 기획한 'We Do With You'(이하 WDWY)의 장점으로 볼 수 있는 실시간 피드백이 이루어질 수 없다.



둘째, '숨씨당'은 이전까지 녹화 강의만을 진행했지만, 최근 라이브 클래스를 개설했다. 하지만 한 클래스 수강생이 100명이 넘는 대형 강의로 이루어지고 있어, 역시 실시간 피드백이 이루어지기는 힘들다. 또한 대부분의 강의를 하루동안 한시적으로 열리는 '원데이클래스' 방식이므로 'WDWY'와 방향성이 다르다.



셋째, '탈잉' 역시 라이브 클래스가 존재하지만, 현재 개설되어있는 라이브클래스는 5가지 밖에 찾아볼 수 없고, 모두 '원데이클래스' 방식이다.



그리고 서비스가 필요한 고객과 서비스를 제공하는 숨은 고수를 쉽고 빠르게 연결해드리는 전문가 매칭 서비스인 숨고가 있는데 이 서비스의 경우 고객과 판매자를 연결해주는 기능만 할 뿐 그 후일 클래스 진행에 대해서는 관여하지 않는다. 또한 내가 선택한 카테고리에 대한 5개정도의 견적서만 보여줌으로 선택의 폭이 넓지 않다.

	숨고	CLASS 101	숨씨당	탈잉	WDWY
강의 개설		○	○	○	○
라이브 강의			○	○	○
소규모	○			○	○
지속성					○
소통					○

프로젝트 소개에서 설명한 'WDWY'의 장점은

1. 수강생들의 만족도를 최대한으로 높일 수 있도록 소규모 클래스로 진행한다.
2. 수강생들의 만족도가 높은 장기간 강의이다.
3. 실시간으로 소규모 클래스로 진행되기 때문에 수강생과 강의진행자 간의 소통 및 피드백이 원활하게 진행될 수 있다.
4. 수강생들끼리의 교류가 가능하다.

이다.

따라서 위 네가지 플랫폼과 비교했을 때, 각각의 단점들을 보완할만한 플랫폼이 'WDWY'가 될 수 있을 것이라 생각한다.

5.구현 기능

A. 안드로이드 스튜디오



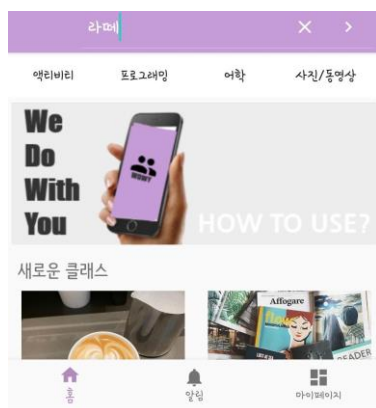
메인페이지는 검색창, 상단바, 배너, 새로운 클래스, 하단바로 구성되어 있다.

카테고리는 horizontalscrollview를 이용하며, 각 카테고리 클릭시 카테고리에 해당하는 클래스를 모아놓은 페이지로 이동한다. 카테고리는 액티비티, 어학, 프로그래밍, 음악, 미술, 사진 및 동영상, 수공예로 구성되어 있다.

배너는 어플상에서 진행되는 각종 이벤트 및 안내사항을 홍보하는 기능을 한다.

최신 업데이트 클래스는 가장 최근에 등록된 클래스를 보여주는 기능을 하며 클릭 시 클래스 상세보기 페이지로 이동한다. 각 데이터는 강의명,강의 요일 및 시간을 화면상에 나타내준다.GridLayoutManager을 이용하여 한줄에 2개에 데이터가 들어가도록 이루어진 세로형 리스트뷰가 되도록 설정하였다.

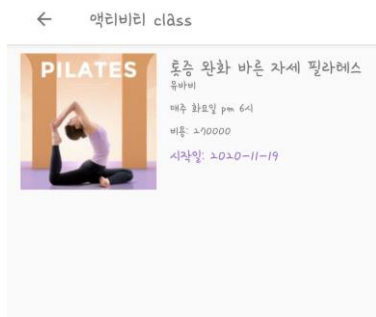
메인페이지의 하단바는 홈, 알림, 마이페이지로 구성되어있으며 navigation bar를 이용해 구현할 예정이다.



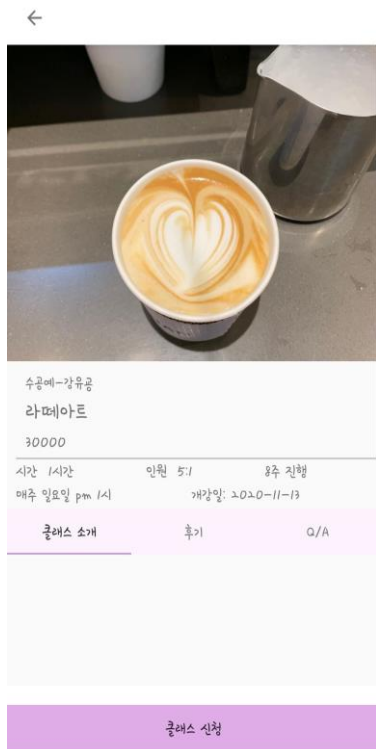
상단바에 위치해있는 강의 검색 기능은 검색어를 db의 user_name,class_name필드와 비교하여 동일한 단어가 들어갈 경우 결과창에 해당 클래스를 띄워주며, 검색 버튼 클릭시 검색 바가 뜨고 그곳에 검색어를 입력하면 된다. 이 기능은 searchview를 이용하여 구현하였으며 검색 버튼을 클릭하면 "검색어를 입력하세요"라는 문구가 뜨고, 검색창에 검색어를 입력하면 해당 검색어를 인텐트를 이용하여 homefragment 클래스에서 search 클래스로 넘겨준다.



search 클래스에서 인텐트를 넘겨받아 검색어를 확인하고 그 검색어를 search_class.php에서는 user name과 class name 2개의 필드 중 다음 검색어와 일치하는 문자열을 갖고 있는 데이터가 있는지 검색한 뒤 그 결과 전체를 보여준다. 그 결과는 json형식으로 내보내준 뒤 리스트뷰에 추가하여 결과를 나타내어준다.



카테고리 클래스에서는 인텐트를 넘겨받아 카테고리를 그 카테고리에 해당하는 결과를 select 하여 result로 보내준다. 이에 해당하는 결과는 다음과 같이 확인할 수 있다.

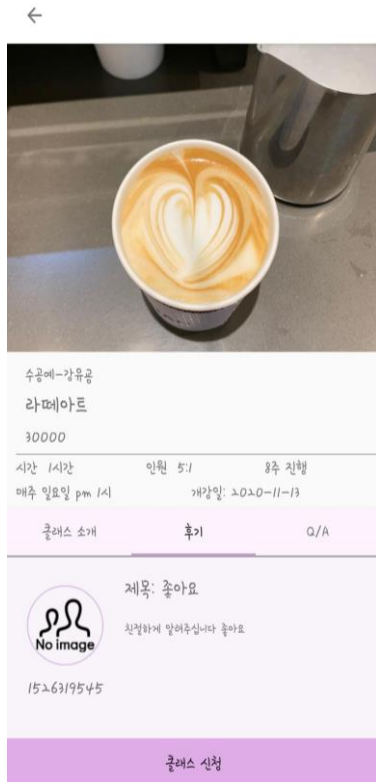


상세보기 페이지에서는 상단에 클래스의 이름, 비용, 대표 이미지 등의 정보를 나타내도록 하였다.

페이지 중간에는 탭 레이아웃과 프레임 레이아웃을 이용하여 강의 내용, Q/A, 후기를 나타내는 탭을 제작하였다. 프래그먼트 3개를 제작하여 각 탭을 클릭할 시 해당하는 프래그먼트로 이동하도록 설정해두었으며, 아무런 탭 버튼을 클릭하지 않은 상황에서는 기본 페이지로 클래스 소개 프래그먼트에 해당하는 fragment1 이 보이도록 설정해두었다.

클래스 소개 프래그먼트의 경우 만약 이미지가 존재하지 않을 경우 미리 ftp서버에 저장해놓은 커밍순 이미지를 띄우도록 설정해두었다.

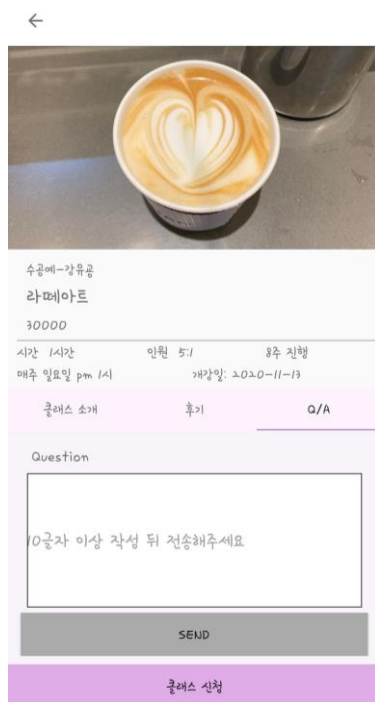
후기 페이지는 해당 강의를 수강한 학생들의 후기를 모든 사용자가



확인할 수 있도록 만든 페이지이다.


후기 프래그먼트의 경우 details.java 페이지에 해당하는 classno1 데이터를 넘겨받은 뒤 해당 데이터를 php로 보내 해당 클래스의 클래스 소개 이미지를 찾고, review board 테이블과 user테이블을 inner join 한 뒤, 그 결과값에 포함되는 후기 데이터들을 받아오도록 하였다. 후기 데이터로는 후기 제목, 후기 내용, 유저 아이디, 유저 이미지를 받아오도록 하였고 유저 이미지가 존재하지 않는 경우 미리 ftp 서버에 저장해놓은 이미지를 결과로 받아오도록 설정해두다. 다음과 같이 받아온 결과는 RecyclerView의 한 줄에 보여줄 데이터를 클래스로 선언하는 item_fragment2.java,

ArrayList에 있는 item_fragment2 클래스 타입의 데이터를 RecyclerView에 보여주는 처리를 하는 Myadapter_fragment2를 만든 뒤 json 형식으로 받은 데이터인 결과를 리스트뷰에 추가하여 결과를 나타내어준다.



q/a 페이지는 해당 강의에 대하여 궁금한 점이 있는 사용자들이 질문을 할 수 있으며, 다른 사용자들이 한 질문과 그에 대해 강의자가 남긴 답변을 찾아볼 수 있는 페이지이다.

우선 질문 작성란은 질문을 빈칸에 작성한 뒤 send 버튼을 누르면 해당 질문의 데이터가 question_board로 저장되고, 그와 동시에 listview를 갱신하여 작성된 질문을 보여준다. 빈 공백이 질문으로 작성되는 것을 방지하기 위해 질문이 최소 10글자 이상인 경우에만 전송 가능하도록 하였습니다.



오직 스피킹 수업만 진행되나요? reading이나 writing 수업의 경우는 같이 이루어지지 않나요?
 yumi

네 맞습니다. 스피킹만 합니다.



교재가 따로 필요한가요?
 yumi

네 준비 후 수업에 일해주세요

클래스 신청

하단의 리스트뷰는 강의에 대한 사용자의 질문과 그에 대해 강의자가 남긴 답변으로 구성하였고, 아직 강의자가 답변을 남기지 않은 경우 '등록된 답변이 없습니다' 라고 뜨도록 설정하였다. 리스트뷰 구성 방법은 후기 프래그먼트의 리스트뷰와 동일하게 설정하였다.

<

결제하기

신청정보

클래스명: Speaking English

강의자: 장동재

결제수단

☒ 인앱결제


결제 금액

30000원

결제하기

클래스 신청 클릭 시 다음과 같이 결제하기 페이지로 이동한다. 결제 페이지에는 다음과 같이 신청한 클래스의 클래스명, 강의자, 결제금액이 표시된다.

Google Play



테스트결제 ₩1,000

테스트 결제

테스트 주문이므로 청구되지 않습니다.

세금 포함

원클릭 구매

클래스 신청 클릭 시 다음과 같이 결제하기 페이지로 이동한다. 결제 페이지에는 다음과 같이 신청한 클래스의 클래스명, 강의자, 결제금액이 표시된다.

결제하기 클릭 시 다음과 같이 결제창이 뜨며 원클릭 구매 클릭 시 결제 완료된다. 결제 완료와 동시에 바로 수강 목록에 내가 결제한 클래스가 추가 되며 마이페이지에서 바로 확인 가능하다.



Q/A 답변 페이지는 사용자들이 상세보기 페이지의 q/a페이지에 남긴 질문에 대해 강의자가 답변을 하는 페이지이다. 마이페이지에서 q/a 답변을 클릭하면 다음과 같이 현재 로그인 된 이용자의 강의 클래스에 입력된 모든 질문에 대한 리스트가 뜨도록 하였다. 우선 user_id가 로그인한 아이디인 클래스들에 대한 질문 목록을 sql문으로 받아와 그 중 답변란이 comment_content가 비어있는 데이터만 가져오도록 하였다.

하단의 리스트뷰 클릭시 상단의 Q로 표시된 부분의 옆에 선택된 질문이 나타나도록 하였다.

질문 클릭 후 답변을 작성한 뒤 답변을 전달 버튼을 클릭하면 해당 답변이 db에 저장되도록 하였다.

클래스가 선택되지 않은 상태에서 답변이 전송되는 것을 방지하기 위해 선택된 질문의 길이가 1이상인 경우만 답변

전달이 진행되도록 하였으며 만약 1 미만인 경우 "질문을 선택해주세요"라는 toast 메시지가 뜨도록 설정하였다.

또한 답변이 제대로 작성되지 않은 상태에서 전송되는 것을 방지하기 위해 작성된 답변의 길이가 10 이상인 경우만 답변 전달이 진행되도록 하였으며 만약 10 미만인 경우 "최소 10글자 이상 작성 뒤 전송해주세요" 라는 메시지가 뜨도록 설정하였다.



후기 페이지는 로그인 된 이용자의 수강 클래스를 나타내고 그 중 후기를 작성하지 않은 강의에 대해 후기 작성을 할 수 있도록 하는 페이지이다. 마이페이지에서 후기 작성을 클릭하면 다음과 같이 현재 로그인 된 이용자의 수강 클래스의 리스트가 뜨도록 하였다. 우선 user_id가 로그인한 아이디인 클래스 목록을 sql문으로 받아와 그 중 후기 작성 유무를 확인할 수 있는 review 필드가 비어있는 데이터만 가져오도록 하였다. 다음과 같이 받아온 결과는 q/a 답변 페이지와 동일한 형식을 이용하여 리스트뷰에 나타나도록 하였다.

답변 전달 후 답변 작성이 완료된 질문은 리스트뷰에 있을 필요가 없으므로 리스트뷰를 초기화시킨 뒤 답변 작성이 완료된 질문을 제외한 질문 리스트를 다시 불러오도록 하였다.

하단의 리스트뷰 클릭시 상단의 '선택강의'라 적혀있는 부분의 옆에 선택된 질문이 나타나도록 하였다.

클래스 클릭 후 후기를 작성한 뒤 후기 전달 버튼을 클릭하면 해당 후기가 db에 저장되도록 하였다. 클래스가 선택되지 않은 상태에

서 후기가 전송되는 것을 방지하기 위해 선택된 클래스의 길이가 1이상인 경우만 후기 전달이 진행되도록 하였으며 만약 1 미만인 경우 "클래스를 선택해주세요"라는 toast 메시지가 뜨도록 설정하였다. 또한 제목이 제대로 작성되지 않은 상태에서 전송되는 것을 방지하기 위해 작성된 제목의 길이가 3 이상인 경우만 답변 전달이 진행되도록 하였으며 만약 3 미만인 경우 "최소 3글자 이상 작성 뒤 전송해주세요" 라는 메시지가 뜨도록 설정하였다. 후기내용 또한 이와 동일한 방식을 적용하였다. 후기 전달 후 후기 작성이 완료된 클래스는 리스트뷰에 있을 필요가 없으므로 리스트뷰를 초기화시킨 뒤 후기 작성이 완료된 클래스를 제외한 클래스 리스트를 다시 불러오도록 하였다.



알림페이지는 사용자의 수강 목록에 있는 강의 목록을 기반으로 채워진다. 현재 시간으로부터 강의 종료시까지 남은 강의를 모두 표시한다.

표시되는 정보는 강의가 있는 날짜, 요일, 시간과 강의명이고 클릭시 해당 강의 수강자 페이지로 이동한다.

데이터베이스의 user 테이블에 있는 alarm_onff 속성을 통해 통해 강의목록을 불러오도록 하거나 불러오지 않도록 할 수 있다.



마이페이지는 사용자의 수강 목록, 강의 목록, 회원정보, 설정 등을 확인할 수 있는 페이지이다. 회원정보 및 설정을 누르게 되면 각각 해당하는 페이지로 이동하게 된다.

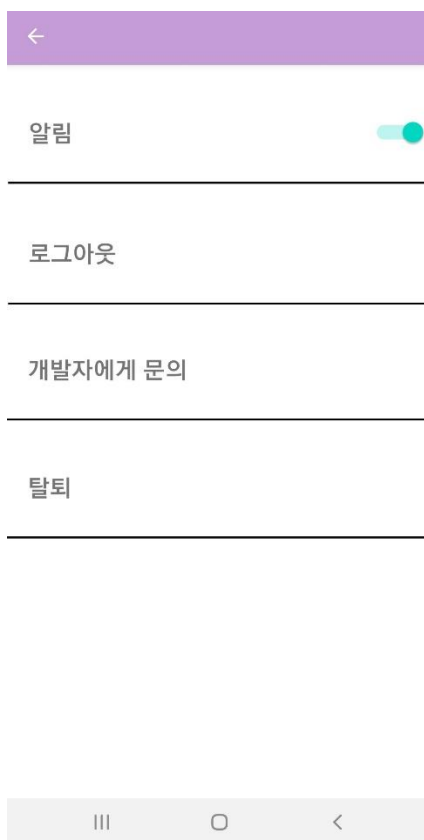
수강 목록의 경우, 데이터베이스의 sugang_class 테이블에서 사용자의 id값과 일치하는 데이터의 class_no2를 얻는다. 이후 class 테이블에서 class_no2와 일치하는 모든 데이터들을 받아온 뒤, listview와 scrollview를 통해 나타낸다. 표시되는 정보는 강의 요일, 시간, 강의명이고 클릭 시 해당 강의 수강자 페이지로 이동한다.

강의 목록의 경우, 데이터베이스의 class 테이블에서 사용자의 id값과 일치하는 데이터를 모두 가져온 뒤, listview와 scrollview를 통해 나타낸다. 표시되는 정보는 강의 요일, 시간, 강의명이고 클릭 시 해당 강의 강의자 페이지로 이동한다.



회원정보 페이지는 마이페이지의 회원 정보 버튼 클릭 시 이동한다. 회원 정보 페이지는 사용자의 id, 이름 그리고 등록된 이미지를 보여준다. 마이페이지에서 id 값을 회원정보페이지로 넘겨 주어 user 테이블에서 id값과 일치하는 데이터 정보를 가져온다.

기존 진행 계획은 회원정보 수정페이지였지만, 로그인을 카카오 로그인 api로 구현하게 되면서 회원 정보 수정이 아닌 확인하는 용도의 페이지로 방향을 바꾸었다.



설정페이지에서는 알림 설정, 개발자에게 문의 페이지로 이동, 로그아웃, 탈퇴를 진행할 수 있다.

알림은 switch 버튼으로 만들어 off일 시, user 테이블의 alarm_onoff를 0으로 변경하고, on일 시 1로 변경한다.

개발자에게 문의 버튼을 클릭하면 개발자에게 문의 페이지로 이동한다.

로그아웃을 클릭하면, 확인 alert 메시지가 뜨고 확인 클릭 시 로그아웃을 진행한다. 로그아웃 시 sqlite에 저장되어 있는 사용자 정보를 삭제하므로 다음번 로그인 시 간편로그인으로 진행할 수 없다.

탈퇴는 가르치는 강의를 있을 경우, "강의가 남아있어 탈퇴를 진행할 수 없다"라는 토스트 메시지가 뜨고, 탈퇴를 진행할 수 없다. 강의를 없을 경우, 비밀번호를 입력하면 데이터베이스에서 사용자 id와 일치하는 데이터를 모두 삭제한다.

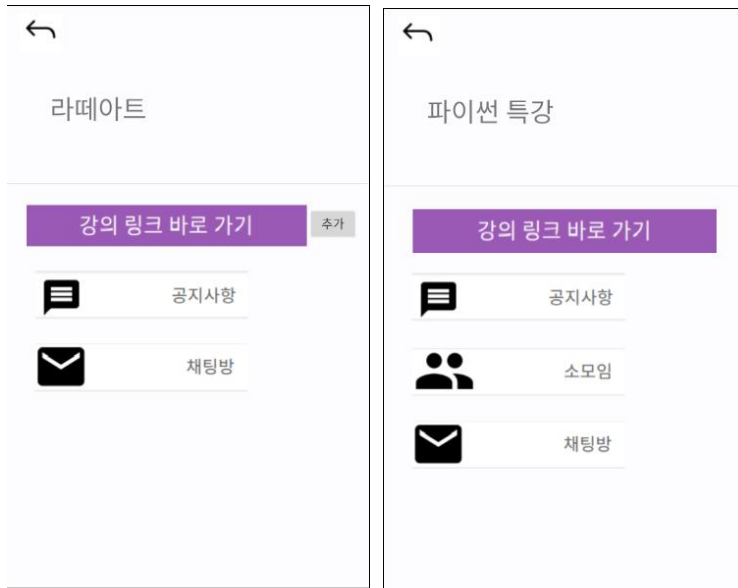
개발자에게 문의 페이지에서는 질문 제목과 내용, 답변을 받을 이메일을 작성한 뒤 보내기를 클릭하면, 데이터베이스 내 developer_QA 테이블에 정보가 저장된다.

placeholder를 통해 입력해야할 내용을 설명했다.

채팅방 페이지는 파이어베이스 Realtime database를 이용해 구현했다.

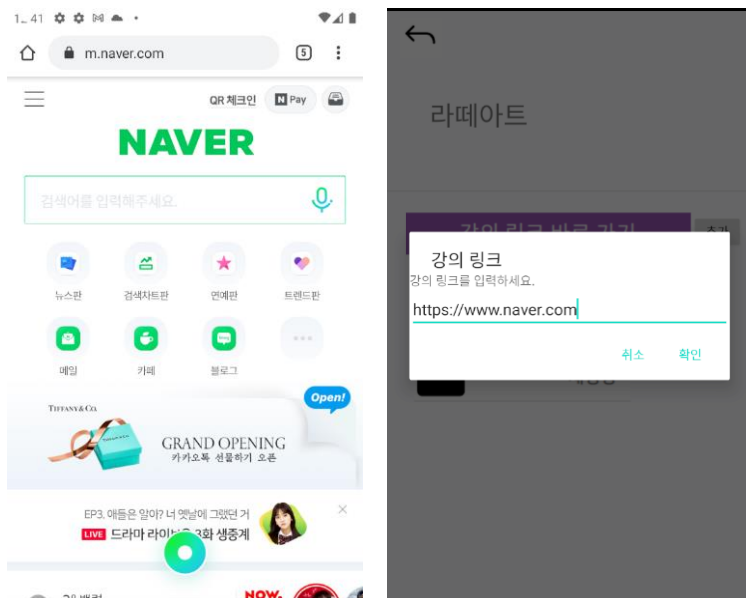
채팅방은 강의 고유 값인 class_no2에 따라 독립적으로 존재해야하므로, 참조객체주소를 class_no2로 지정했다. 메시지 입력란에 메시지를 입력하면 데이터베이스로 사용자 id, 이름, 사진, 시간, 날짜를 전송한다.

날짜를 나타내기 위해 현재 보낸 메시지의 날짜와 가장 최근의 메시지의 날짜가 다를 경우, 현재 날짜를 함께 포함하고 있는 레이아웃을 통해 메시지를 나타낸다. 날짜가 같을 경우 메시지만 표시해준다.



왼쪽은 강의자 페이지, 오른쪽은 수강자 페이지로, 마이페이지에서 클래스를 선택 시 각 클래스에 맞는 페이지가 뜨도록 하였다.

강의자 페이지의 경우, 강의 링크를 추가(수정)할 수 있으며, 채팅방 버튼은 제공되지 않는다. 반대로 수강자 페이지의 경우, 강의 링크 추가를 제외한 강의링크 바로가기, 공지사항, 소모임, 채팅방 기능이 제공된다.



강의 링크 바로가기 버튼을 누를 경우 db에 저장되어 있는 링크 페이지가 새로 뜨며, 링크 추가 버튼을 누를 경우 강의 링크를 입력할 수 있으며, 확인을 누르면 db에 해당 클래스의 링크가 업데이트된다.

←

공지사항

글쓰기

라떼아트 준비물입니다. 강유공

○○○○○○○ 강유공

notice for latteart 강유공

←

제목을 입력하세요

작성

내용을 작성하세요.

←

라떼아트 준비물입니다.

우유, 스팀기, 피쳐를 준비하세요.

강의자의 공지사항 기능 경우엔, 글 열람과 작성이 가능하다. 우선 글의 목록을 확인할 수 있는데, 등록된 순대로 제목과 글쓴이가 출력되도록 하였다. 글쓰기 버튼을 누르면 작성 페이지가 뜨고, 제목과 내용을 작성하면 db에 입력 순대로 공지글이 저장된다.

글 열람의 경우, db에서 클릭한 제목과 일치하는 공지글을 찾은 뒤 해당 글을 출력하도록 하였다.

←

소모임

● TUESDAY : pm 9 ●

+

이번주 소모임 가능한 날짜를 입력해주세요!

←

월 화 수 목 금 토 일

13	가능				가능	가능
15					가능	
17				가능	가능	가능
19					가능	
21		가능	가능	가능	가능	

소모임 버튼을 누를 시에는, 이번주 소모임 요일과 시간이 뜬다. 이 시간은 해당 클래스를 듣는 수강생들이 '가능'으로 설정한 일정 중 한 하나를 골라 임의로 일정을 정해주는 기능으로, db에서 수강생들의 스케줄을 불러와 가능한 시간대를 고르는 방식으로 진행된다.

밑에 + 버튼을 누를 시에 오른쪽 페이지가 뜨며, 표에 맞는 요일과 시간대를 toggle button을 통해 설

정할 수 있다. 이 스케줄 정보는 매 클릭 시마다 db에 업데이트된다. 모든 일정은 요일별로 길이가 5인 string으로 이루어져있으며, 가능한 시간대이면 1, 아니면 0으로 저장되어있다. 예를 들어 해당 페이지의 월요일 일정은 가능 불 불 불 불이므로, db에 월요일 스케줄은 10000으로 저장되어있는 것이다.



로그인은 카카오로그인을 통해 회원가입 없이 SNS 로그인을 할 수 있도록 하였다. Oauth api와 kakao SDK를 이용하여 구현했으며, 카카오로그인을 클릭 시 카카오 계정으로 로그인할 수 있는 창이 뜬다. 카카오 로그인을 한 이력이 있는 경우, id가 내부 db인 sqlite에 저장되어, 카카오 계정 입력 없이 간편로그인을 할 수 있다. 카카오 로그인 이력이 없거나 로그아웃 된 상태라면, 이러한 간편로그인은 불가능하도록 설정했다.

B. php

-1)search_class.php

```
$res = mysqli_query($conn, "SELECT * FROM class
WHERE user_name LIKE '%$search%' or class_name LIKE '%$search%'");
```

```
array_push($result,
    array('class_name' =>$row["class_name"],
        'class_no' =>$row["class_no"],
        'class_no2' =>$row["class_no2"],
        'class_image' =>$row["class_image"],
        'user_name' =>$row["user_name"],
        'class_time' =>"매주 ".$day1.$day2.$day3.$day4.$day5.$day6.$day7. $row["class_ap"]." ". $row["class_hour"]."시",
        'class_price' => "비용: ".$row["class_price"],
        'class_start' => "시작일: ".$row["class_start"]
    ));
```

search_class.php에서는 user name과 class name 2개의 필드 중 다음 검색어와 일치하는 문자열을 갖고 있는 데이터가 있는지 검색한 뒤 그 결과 전체를 보여준다.

검색 결과는 json형식으로 보내주고, 검색어의 결과에 해당하는 강의 list 띄워준다. 강의자명, 강의명, 강의 비용, 강의 요일 및 시간, 강의 시작일을 나타내 준다.

-2)category_class.php

```
$res = mysqli_query($conn, "SELECT * FROM class
WHERE class_category LIKE '%$search%' order by class_no2 desc");
```

category_class.php에서는 각 카테고리별 클래스를 최신등록순으로 나타낸다. 검색 결과는 json형식으로 보내주고, 카테고리에 해당하는 강의 list 띄워준다. 강의자명, 강의명, 강의 비용, 강의 요일 및 시간, 강의 시작일을 나타내 준다.

-3)update_class.php

```
$res = mysqli_query($conn, "SELECT * FROM class order by class_no2 desc limit 4");
```

update_class.php에서는 최신 데이터 4개만 가져오도록 설정하였다.

-4)class_data1.php

```
$res = mysqli_query($conn, "SELECT * FROM class
WHERE class_no2='$class_no2'");
```

```
array_push($result,
    array('class_name' =>$row["class_name"],
        'class_no' =>$row["class_no"],
        'class_no2' =>$row["class_no2"],
        'class_image' =>$row["class_image"],
        'user_name' =>$row["user_name"],
        'class_time' =>"매주 ".$day1.$day2.$day3.$day4.$day5.$day6.$day7. $row["class_ap"]." ". $row["class_hour"]."시",
        'class_price' => $row["class_price"],
        'class_start' => "개강일: ".$row["class_start"],
        'class_category' => $row["class_category"],
        'class_count'=>$row["class_count"]."주 진행",
        'class_runningtime'=>"시간 ".$row["class_runningtime"]."시간",
        'class_max_student'=>"인원 ".$row["class_max_student"].":1"
    ));
```

class_data1.php은 불러와야 하는 페이지의class_no2를 받아와 이와 동일한 class_no2를 갖는 데이터를 가져오도록 하였고, 그에 대한 결과값은 json형식으로 다음과 같이 내보내주도록 하였다.

-5)classdata2.php

```
$res = mysqli_query($conn, "SELECT * FROM class
WHERE class_no2='$class_no2'");

$result = array();
while($row = mysqli_fetch_array($res)){
    $image=$row["info_image1"];
    if( empty($row["info_image1"])){$image="http://hyper0616.dothome.co.kr/image/commingsoon.PNG";}
    array_push($result,
        array('info_image1' =>$image
        ));
}
```

class_data2.php는 불러와야 하는 페이지의 class_no2를 받아와 당 클래스의 클래스 소개 이미지를 찾고, 만약 이미지가 존재하지 않을 경우 미리 ftp서버에 저장해놓은 커밍순 이미지를 띄우도록 설정해 두었다.

-6)review.php

```
$res = mysqli_query($conn, "SELECT * FROM review_board
INNER JOIN user ON review_board.review_id=user.id
WHERE class_no='$class_no' order by board_no desc");

$result = array();
while($row = mysqli_fetch_array($res)){
    $image=$row["image"];
    if( empty($row["info_image1"])){$image="http://hyper0616.dothome.co.kr/image/no_image.PNG";}
    array_push($result,
        array('user_id' =>$row["user_id"],
        'board_contents' =>$row["board_contents"],
        'comment_contents' =>$row["comment_contents"],
        'user_image' =>$image
        ));
}
```

review.php의 경우 review board 테이블과 user테이블을 inner join 한 뒤, 그 결과값에 포함되는 후기 데이터들을 받아오도록 하였다. 후기 데이터로는 후기 제목, 후기 내용, 유저 아이디, 유저 이미지를 받아오도록 하였고 유저 이미지가 존재하지 않는 경우 미리 ftp서버에 저장해놓은 이미지를 결과로 받아오도록 설정해두었다.

-7)qa.php

```

$res = mysqli_query($conn, "SELECT * FROM question_board
    INNER JOIN user ON question_board.question_id=user.id
    WHERE class_no='$class_no' order by board_no desc");

$result = array();
while($row = mysqli_fetch_array($res)){
    $image=$row["image"];
    $comment_contents=$row["comment_contents"];
    if( empty($row["info_image1"])){$image="http://hyper0616.dothome.co.kr/image/no_image.PNG";}
    if( empty($row["comment_contents"])){$comment_contents="등록된 답변이 없습니다";}
    array_push($result,
        array('user_id' =>$row["question_id"],
            'board_contents' =>$row["board_contents"],
            'comment_contents' =>$comment_contents,
            'user_image' =>$image
        ));
}

```

qa.php의 경우 qa board 테이블과 user테이블을 inner join 한 뒤, 그 결과값에 포함되는 질문 데이터들을 받아오도록 하였다. 질문 데이터로는 질문 내용, 질문 아이디, 질문 이미지를 받아오도록 하였고 유저 이미지가 존재하지 않는 경우 미리 ftp서버에 저장해놓은 이미지를 결과로 받아오도록 설정해 두었다. 질문 밑에는 그에 대해 강의자가 남긴 답변이 같이 구성되도록 설정하였는데 아직 강의자가 답변을 남기지 않은 경우 '등록된 답변이 없습니다' 라고 뜨도록 설정하였다.

-8)qasend.php

```

$query="SELECT board_no FROM question_board ORDER BY board_no desc limit 1";
$result = mysqli_query($conn, $query);
if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        $total_rows=$row["board_no"];
    }
    $new_rows=(int)$total_rows +1;

    $sql = "INSERT INTO question_board (board_contents,class_no,question_id,board_no)
    VALUES ('$num1','$text1','$text2','$new_rows')";
}

```

qasend.php의 경우 질문 내용과 질문자 아이디,질문 대상 강의 no를 앱에서 받아와 db에 저장하는 과정이다. 이 과정에서 현재 question_board에 있는 board_no 중 가장 숫자가 높은 값을 받아와 이 값에 +1한 값을 board_no으로 부여하도록 설정하였다.

-9)review_list.php

```

$res = mysqli_query($conn,
    "SELECT * FROM class
    INNER JOIN sugang_class ON class.class_no2=sugang_class.class_no2
    WHERE sugang_class.user_id='$login'

```

```

while($row = mysqli_fetch_array($res)){
    if( empty($row["review"])){
        array_push($result,
        array(
            'class_no' =>$row["class_no"],
            'class_image' =>$row["class_image"],
            'class_name' =>$row["class_name"]
        ));
    }
}

```

review_list.php는 우선 user_id가 로그인한 아이디인 클래스들의 목록에 접근한 뒤 review가 비어있는 데이터만 가져오도록 설정하였다.

-10)review_send.php

```

$query="SELECT board_no FROM review_board ORDER BY board_no desc limit 1";
$result = mysqli_query($conn, $query);
if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        $total_rows=$row["board_no"];
    }
}
$new_rows=(int)$total_rows +1;

```

```

$sql = "INSERT INTO review_board
(board_no, class_no, review_id, board_contents,comment_contents)
VALUES ('$new_rows','$class_no','$review_id','$board_contents','$comment_contents')";

if ($conn->query($sql) === TRUE) {
    $sql2="UPDATE sugang_class SET review='1' WHERE class_no='$class_no' and user_id='$review_id'";
    if($conn->query($sql2) === TRUE)
        {echo "1";}
}

```

review_Send.php에서는 후기에 대한 새로운 board_no를 만들어주기 위해 현재 review_board에 있는 데이터 중 board_no값이 가장 큰 데이터 값의 넘버를 뽑아왔고 여기에 1을 더하여 새로 입력할 후기 데이터의 board_no이 되도록 작성하였다. 이 값을 포함하여 db에 후기 데이터를 전송한 뒤 후기 데이터 전송이 완료되면 이와 함께 sugang_class의 review 부분을 1로 수정하고 이 모든 작업이 완료되면 1을 안드로이드로 전송하여 db 전달이 성공적으로 완료되었음을 앱상에서 확인 가능하도록 하였다.

-11)qa_answer.php

```

$res = mysqli_query($conn, "SELECT * FROM class
INNER JOIN question_board ON class.class_no=question_board.class_no
WHERE user_id='$login' order by user_id desc");//user_id가 로그인한

```

```

$result = array(); // 'class_image' =>$image,
while($row = mysqli_fetch_array($res)){
    if( empty($row["comment_contents"])){
        array_push($result,
            array( 'class_qa'=>$row["board_contents"],
                'user_name' =>$row["question_id"],
                'board_no' =>$row["board_no"],
                'class_image' =>$row["class_image"],
                'class_name' =>$row["class_name"]
            ));
    }
}

```

qa_answer.php는 우선 user_id가 로그인한 아이디인 클래스들의 강의 목록에 해당하는 질문 리스트에 접근한 뒤 답변란이 비어있는 데이터만 가져오도록 설정하였다.

-12)qa_answer_send.php

```

$sql = "UPDATE question_board SET comment_contents='$answer' WHERE board_no='$no'";
if ($conn->query($sql) === TRUE) {
    echo "1";
} else {
    echo "Error: " . $sql . " ";
    . $conn->error;
}

```

qa_answer_send.php는 답변 전송시 해당 질문의 답변란에 저장되도록 하였다.

-13)pay.php

```

$query="SELECT sugang_no FROM sugang_class ORDER BY sugang_no desc limit 1";
$result = mysqli_query($conn, $query);
if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        $total_rows=$row["sugang_no"];
    }
}
$new_rows=(int)$total_rows +1;

$sql = "INSERT INTO sugang_class (sugang_no,class_no,class_no2,user_id)
VALUES ('$new_rows','$num1','$num2','$text2')";

```

pay.php는 user가 클래스 구매 완료시 구매한 클래스를 수강 목록에 추가해주는 역할을 한다. 현재 sugang_class에 있는 데이터 중 sugang_no값이 가장 큰 데이터 값의 넘버를 뽑아왔고 여기에 1을 더 하여 새로 입력할 후기 데이터의 sugang_no이 되도록 작성하였다.

-14)learn_class.php


```

if ($class_count > 0) {
    array_push($result,
        array(
            'class_name' => $row["class_name"],
            'class_no2' => $row["class_no2"],
            'class_time' => "매주 " . $day1 . $day2 . $day3 . $day4 . $day5 . $day6 . $day7 . $row["class_ap"] . " " . $row["class_hour"] . "시"
        ));
}

```

learn_class.php는 마이페이지에서 user의 수강 목록을 불러오는 역할을 한다. user_id 값을 받아 sugang_class 테이블에서 일치하는 class_no2를 받아온 뒤, class 테이블의 class_no2에 해당하는 모든 데이터를 불러온다. class_count가 0이 되면 강의가 종료된 것을 의미하므로 더 이상 나타낼 필요가 없기 때문에 class_count > 0 이라는 조건을 추가했다.

-15)teach_class.php

```

if ($class_count > 0) {
    array_push($result,
        array(
            'class_name' => $row["class_name"],
            'class_no2' => $row["class_no2"],
            'class_time' => "매주 " . $day1 . $day2 . $day3 . $day4 . $day5 . $day6 . $day7 . $row["class_ap"] . " " . $row["class_hour"] . "시"
        ));
}

```

teach_class.php는 마이페이지에서 user의 강의 목록을 불러오는 역할을 한다. user_id 값을 받아 class 테이블에서 일치하는 모든 데이터를 불러온다. class_count가 0이 되면 강의가 종료된 것을 의미하므로 더 이상 나타낼 필요가 없기 때문에 class_count > 0 이라는 조건을 추가했다.

-16)send_to_developer.php

```

$sql = "insert into developer_QA (user_id, user_email, question_title, question_contents) values ($user_id, $email, $title, $content)";

if ($conn->query($sql) == TRUE) {
    echo "1";
} else {
    echo "Error: " . $sql . $conn->error;
}

```

send_to_developer.php는 개발자에게 문의 페이지에서 작성한 질문을 데이터베이스 내 developer_QA 테이블에 저장하는 역할을 한다. 테이블에 데이터 추가를 성공하면 1을 반환하고, 실패하면 에러 메시지를 반환한다.

-17)modify_alarm.php

```

if ($alarm_ == 1) {
    $sql = "update user set alarm_onoff = 0 where id = '$user_id_'";

    if ($conn->query($sql) === TRUE) {
        echo "1";
    } else {
        echo "Error: " . $sql . $conn->error;
    }
} else if ($alarm_ == 0) {
    $sql = "update user set alarm_onoff = 1 where id = '$user_id_'";

    if ($conn->query($sql) === TRUE) {
        echo "1";
    } else {
        echo "Error: " . $sql . $conn->error;
    }
}
}

```

modify_alarm.php는 설정페이지에서 알림 스위치 버튼을 눌렀을 때, user 테이블의 alarm_onoff 속성 값을 변경해주는 역할을 한다. 기존 alarm_onoff 값이 1이라면 0으로 변경하고, 0이라면 1로 변경한다. 변경이 성공적으로 되면 1을 반환하고, 실패 시 에러메세지를 반환한다.

-18) get_user_info.php

```

$res = mysqli_query($conn, query: "select user.name, user.image, user.password, user.alarm_onoff from user where user.id = '$userId'");

$result = array();

while($row = mysqli_fetch_array($res)){

    array_push($result,
        array('name' =>$row["name"],
            'image' =>$row["image"],
            'password' =>$row["password"],
            'alarm' =>$row["alarm_onoff"]
        ));
}

```

get_user_info.php는 회원정보페이지, 설정페이지 등에서 user_id를 통한 사용자의 정보를 받아오기 위한 php이다. 받아온 user_id 값을 통해 user 테이블에서 일치하는 데이터를 찾고 이름, 이미지 주소, 비밀번호, 알림 값을 가져온다.

-19) get_alarm_list.php

```

// class_count 빼줄 주 --> 수업한 주차 수
$cha_q = floor( value: $past_day / 7);
// 남은 기간 주차 수
$q = floor( value: ($total_day - $past_day) / 7);
// 남은 기간 일 수
$r = ($total_day - $past_day) % 7;
// class_count를 수업한 주차만큼 빼줌
$class_count -= $cha_q;

```

```
// 날짜 순으로 정렬
$result = arr_sort($result, key: 'class_time');
```

get_alarm_list는 알람페이지에서 남아있는 강의목록을 불러오기 위해 사용된다. 먼저 class 테이블에서 class_count 값과 class_start를 가져온다. 강의를 진행한 주차를 cha_q로 나타내고, 남은 주차를 q로 나타냈다. r은 남은 일 수이다. 그런 뒤에 class_count에서 수업이 진행된 만큼(cha_q) 빼준 뒤, class_count가 0이 될 때까지 result 배열에 넣어준다. 모든 강의를 result 배열에 추가했다면, 강의 목록을 시간순으로 맞춰주기 위해 class_time을 기준으로 정렬을 진행한 뒤 반환한다.

-20)delete_user_info.php

```
if ($conn->query($sql1) === TRUE && $conn->query($sql2) === TRUE && $conn->query($sql3) === TRUE &&
    $conn->query($sql4) === TRUE && $conn->query($sql5) === TRUE && $conn->query($sql6) === TRUE &&
    $conn->query($sql7) === TRUE && $conn->query($sql8) === TRUE && $conn->query($sql9) === TRUE) {
    echo "1";
} else {
    echo "Error: " . $conn->error;
}
```

delete_user_info.php는 회원 탈퇴를 눌렀을 때, 데이터베이스 내 user_id와 일치하는 모든 정보를 삭제하기 위해 사용된다. 데이터베이스 내 모든 테이블에서 받아온 user_id 값을 삭제하는데 성공했다면 1을 반환하고 아닐 시 에러 메시지를 반환한다

-21)can_delete.php

```
if (empty($result)) {
    echo "1";
} else {
    echo "0";
}
```

can_delete.php는 회원 탈퇴를 눌렀을 때, 사용자가 가르치는 강의가 있는지 없는지 확인하기 위한 php이다. 앞서 말한 teach_class.php 와 같이 class_count를 구한 뒤, class_count가 0 이상인 강의를 result 배열에 넣는다. 마지막으로 result 배열에 아무것도 없다면 1을 반환하고, 원소가 존재하면 0을 반환한다.

-22)get_classname.php

```
$result = mysqli_query($conn, "SELECT * FROM class WHERE class_no2='$class_no2'");
if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        echo $row["class_name"];
    }
}
```

이전 페이지에서 넘겨받은 class_no2 값을 통해, db에서 class_no2가 일치하는 클래스의 클래스명을 불러온다.

-23)get_class_link.php

```
$result = mysqli_query($conn, "SELECT * FROM sugang_class WHERE class_no2='$class_no2'");  
if (mysqli_num_rows($result) > 0) {  
    while($row = mysqli_fetch_assoc($result)) {  
        echo $row["class_link"] ;  
        break;  
    }  
}
```

클래스 명을 가져오는 코드와 마찬가지로, class_no2가 일치하는 클래스의 링크를 가져온다.

-24)input_class_link.php

```
$result = "UPDATE sugang_class SET class_link = '$class_link' WHERE class_no2='$class_no2'";
```

강의 페이지에서 강의 링크 추가 버튼을 누를 시, class_no2가 일치하는 클래스의 링크에 입력한 링크를 업데이트 해주는 코드이다.

-25)add_notice.php

```
$result = mysqli_query($conn,  
    "INSERT INTO notice (notice_title, user_id,  
        notice_contents, class_no2, class_no, name)  
    VALUES ('$notice_title', '$notice_writer',  
        '$notice_contents', '$class_no2', '$class_no2', '$user_name')");
```

공지사항 테이블에 작성한 공지를 추가하는 코드이다. 이는 강의자가 공지사항을 작성할 때 이루어진다. 이전 페이지에서 받아온 class_no2 값과, 내부 db에서 받아온 id를 통해, 어떤 클래스의 공지사항이며, 누가 작성자인지 명시할 수 있다.

-26)get_user_name.php

```
$result = mysqli_query($conn, "SELECT * FROM user WHERE id='$userId'");
```

위에서 언급한 작성자를 명시하기 위해, 내부 db에 저장된 id를 통해 해당 유저(강의자)의 이름을 불러오는 코드이다.

-27) get_notice.php

```
$res = mysqli_query($conn, "SELECT * FROM notice
WHERE notice.class_no='$class_no'
ORDER BY notice.notice_no");

$result = array();
while($row = mysqli_fetch_array($res)){
    array_push($result,
        array('title' =>$row["notice_title"],
            'writer' =>$row["name"]
        ));
}
echo json_encode(array("result"=>$result));
```

db에서 class_no가 일치하는 클래스의 모든 공지사항 제목과 글쓴이를 불러오는 코드이다. 이는 공지사항 페이지에서, 공지사항 제목과 작성자 목록을 출력하는 recyclerview에 이용된다. 이때, 제목과 작성자를 한 번에 불러와서 배열 형식으로 사용하기 위해, 두 값을 json 형식으로 출력하도록 하였다.

-28) get_notice_content.php

```
$res = mysqli_query($conn, "SELECT * FROM notice
WHERE notice_title = '$title'");

if (mysqli_num_rows($res) > 0) {
    while($row = mysqli_fetch_assoc($res)) {
        echo $row["notice_contents"];
    }
}
```

공지사항 목록에서 제목을 클릭할 경우, 해당 제목과 일치하는 공지사항의 내용을 db에서 찾아 출력한다.

-29) get_somoim_date.php

```
$res = mysqli_query($conn, "SELECT * FROM somoim
WHERE somoim.class_no='$class_no'
ORDER BY somoim.somoim_no");

$result = array();
while($row = mysqli_fetch_array($res)){
    array_push($result,
        array('day' =>$row["somoim_day"],
            'ap' =>$row["somoim_ap"],
            'hour' =>$row["somoim_hour"],
        ));
}
echo json_encode(array("result"=>$result));
```

정해진 소모임의 일정을 불러와 출력하는 코드이다. 굳이 배열 형식을 사용할 필요는 없었지만, 소모임 일정 개수를 정하기 전, 초기에 작성한 코드를 사용하여 배열 형식으로 이루어져있다.

-30)get_schedule.php

```
$res = mysqli_query($conn, "SELECT * FROM schedule
WHERE class_no2='$class_no2_' AND user_id='$userId_'");
//WHERE schedule.class_no=4 AND schedule.user_id=1524618196");

$result = array();
while($row = mysqli_fetch_array($res)){
    array_push($result,
        array('mon_' =>$row["user_schedule_mon"],
            'tue_' =>$row["user_schedule_tue"],
            'wed_' =>$row["user_schedule_wed"],
            'thu_' =>$row["user_schedule_thu"],
            'fri_' =>$row["user_schedule_fri"],
            'sat_' =>$row["user_schedule_sat"],
            'sun_' =>$row["user_schedule_sun"]
        ));
}
echo json_encode(array("result"=>$result));
```

이 코드는 소모임 일정을 정할 때 사용되는 코드이다. 각 소모임 일정은 모든 요일 다섯 시간대로 이루어져 있다. 각 요일은 다섯시간대의 가능/불가능 여부를 0과 1로 나타내는 string으로 표현된다. 이 php 코드를 통해, 해당 유저의, 모든 요일의 스케줄을 길이가 5인 string 7개로 불러올 수 있다. 배열 형식으로 사용하기 위해 json 형식으로 출력하며, 불러온 값은 후에 android studio에서 char 형식으로 변형하여 토글 버튼 클릭 시 0과 1이 변환되도록 하였다.

-31)push_요일.php

```
$result = "UPDATE schedule SET user_schedule_mon = '$schedule_'
WHERE class_no2='$class_no2_' AND user_id='$userId_'";
```

(Mon~sun 모든 요일이 있으며, 구조는 다 동일함.)

위에서 언급한 대로, 유저가 각 시간대의 토글 버튼을 클릭할 때마다 변경되는 스케줄의 값은 각 요일 별로 업데이트 하도록 하였다. 가능한 경우 1로, 불가능한 경우 0으로 저장되는 길이 5의 string 값이 업데이트 된다.

-32)get_users.php

```
$res = mysqli_query($conn, "SELECT * FROM schedule
WHERE class_no2='$class_no2_'");

if (mysqli_num_rows($res) > 0) {
    while($row = mysqli_fetch_assoc($res)) {
        echo $row["user_id"]. " ";
    }
}
```

소모임 날짜를 정하기 위해, class_no2에 해당 클래스를 수강하는 모든 학생들의 목록을 불러오는 코드이다. 이 코드를 통해 모든 수강자들의 아이디를 알 수 있고, 이 아이디를 이용하여, 각자의 스케줄을 위에서 언급한 get_schedule.php를 이용하여 불러올 수 있다.

-33)save_somoim.php

```
$sql = "UPDATE somoim SET somoim_day = '$yoil', somoim_hour = '$time'
      WHERE class_no2 = '$class_no2'";
```

위에서 이야기한대로 불러온 해당 클래스의 모든 수강생들의 스케줄을 토대로 정해진 소모임 일정을, 해당 클래스의 소모임 날짜로 업데이트한다.

-34)update_log.php

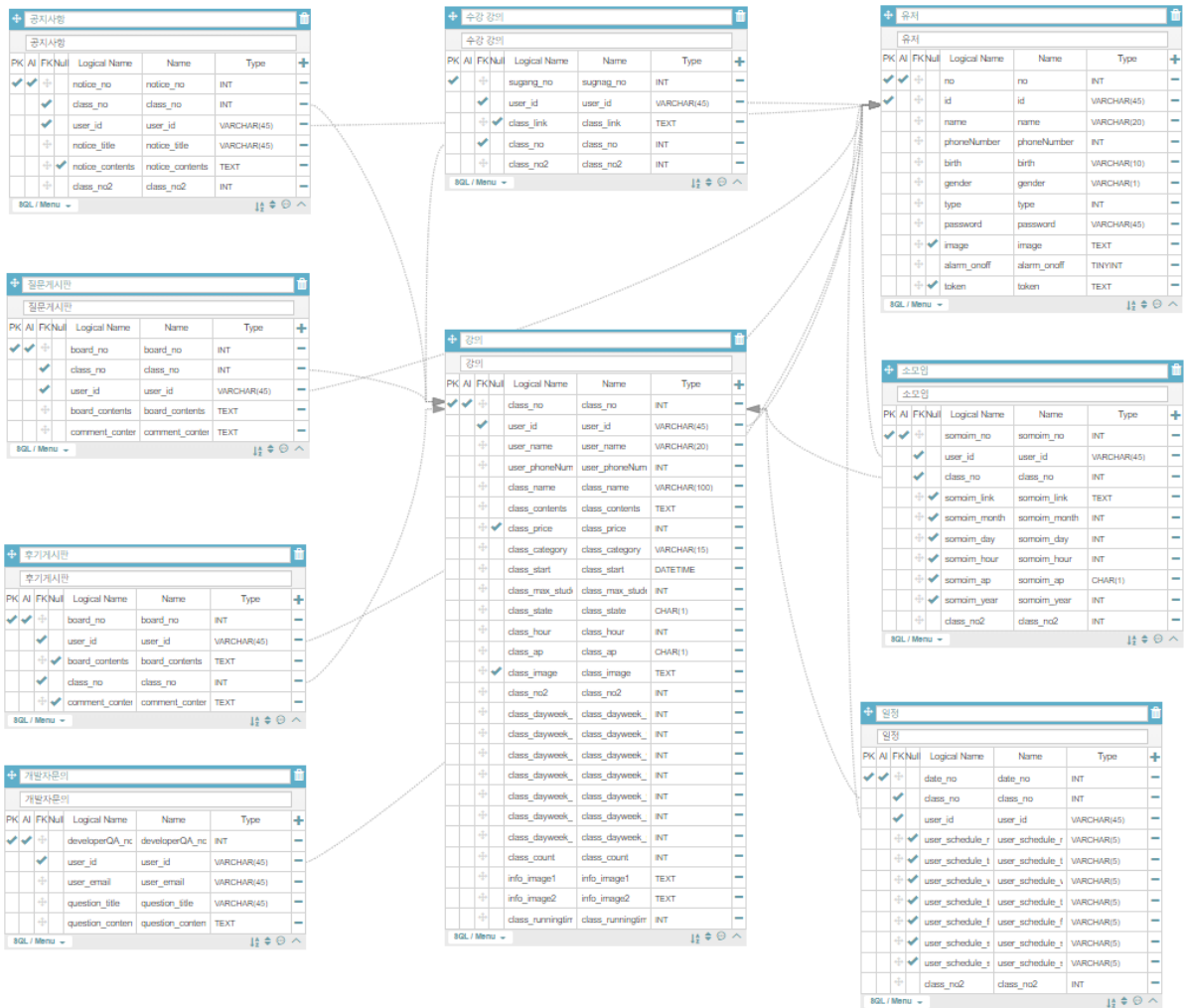
```
$res = mysqli_query($conn, "SELECT * FROM user where id = '$user_id'");
if (mysqli_num_rows($res) > 0) {
    while($row = mysqli_fetch_assoc($res)) {
        echo $row["user_id"]."//";
    }
}
else {
    $sql = "INSERT INTO user (id, name, image, phoneNumber, birth, gender, type, password, token)
    VALUES ('$user_id','$nickname_', '$user_image', '', '', '', 0, '', NULL)";

    if ($conn->query($sql) === TRUE) {
        echo "1";
    }
    else{
        echo "FAIL: ". $sql . $conn->error;;
    }
}
```

로그인할 때 이용되는 코드이다. 로그인한 정보(입력된 아이디)가 db에 존재하지 않는 경우, 해당 유저를 db에 추가한다. 이때 저장되는 정보는, 카카오에서 받아오는 정보들로, 이용자의 이름과 프로필 사진을 받아온다.

이미 존재하는 아이디일 경우에는 따로 정보를 추가하지 않고, 아이디만 불러온다.

C. MySQL



데이터베이스는 user, class, sugang_class, notice, qa_board, review_board, somoim, schedule, developer_QA => 총 9개의 테이블로 구성된다.

6. 주간 진행 계획

		PLAN														
		1주차	2주차	3주차	4주차	5주차	6주차	7주차	8주차	9주차	10주차	11주차	12주차	13주차	14주차	15주차
공통	제안서 작성								중 간 고 사						최 종 발 표	최 종 보 고 서
	제안서 수정/보완															
	수정 및 발표준비															
정동재	DB 설계 및 구축 +DB 데이터 정리															
	워크플로우 제작															
	마이페이지 구현															
	회원정보페이지 구현															
	설정 페이지 구현															
	알림 페이지 구현															
	개발자에 문의 페이지 구현															
	채팅방 페이지 구현															
김지혜	안드로이드 UI 설계 : 홈, 클 래스 신청															
	서버 안드로이드 연동															
	프로토타입 제작															
	메인페이지 + 강의 검색 구 현															
	강의 카테고리별 페이지 구 현															
	강의 상세보기 페이지 구현															
	Q/A 답변 페이지 구현															
	후기 작성 페이지 구현															
	구매 및 결제 페이지 구현															
강유경	안드로이드 UI 설계 : 마이페 이지, 클래스															
	각 페이지 개발 상세설명 제 작															
	카카오 api 연동 및 로그인 구현															
	수강자, 강의자 클래스페이 지 구현															
	공지사항페이지 + 공지사항 작성 구현															
	소모임 페이지 구현															

7. github

apk: <https://github.com/jihye-kim11/wdwy2020/blob/main/10-1.apk>

app: https://github.com/jihye-kim11/wdwy2020/tree/main/wdwy_1130

php: <https://github.com/jihye-kim11/wdwy2020/tree/main/php>

db: <https://github.com/jihye-kim11/wdwy2020/tree/main/sql>