



A402 : Audiro

삼성 SW 청년 아카데미 서울캠퍼스 8기
공동프로젝트 (6주, 2022/01/09 - 2022/02/17)

포팅 매뉴얼

담당 컨설턴트 : 이상현

이가옥(팀장), 김성환, 김호성, 신승호, 윤소희

1. 프로젝트 기술 스택
 - a. FE
 - b. BE
 - c. IOT
2. 빌드 방법
 - a. 백엔드 빌드 방법
 - b. 프론트엔드 빌드 방법
 - c. 배포 명령어 정리
3. 프로퍼티 정의
 - a. Nginx Default 값 세팅
 - i. ec2에서 세팅 파일로 접근
 - ii. 세팅값 다음과 같이 변경하기
 - b. Git ignore 파일
 - i. aws.yml 파일
4. EC2 세팅
 - a. 세팅을 위한 최신 상태 업데이트
 - b. MySQL 설치
 - c. 추가 세팅을 위한 이동 후 편집
 - d. 확인
5. 외부서비스
6. IoT 포팅 매뉴얼
7. Web 포팅 매뉴얼

1. 프로젝트 기술 스택

A. FE

기술 스택(버전): React 108.2.0, Styled Component, node package manager 0.11.0

사용 툴 : Visual Studio Code 1.70.1

B. BE

기술 스택(버전): Spring boot 2.7.7, MySQL 8.0.30, Nginx 1.18.0, OAuth2, AWS EC2

사용 툴: IntelliJ 2021.2.4, MobaXterm, MySQL 8.0 Commandline Client, JDK zulu8.33.0.1, Postman

C. IOT

기술 스택(버전): PyQt5 5.15.9, Python3 3.10.1, Raspberry Pi 4 Raspbian OS 64bit, requests window : 2.28.1 RPI

사용 툴 : PyCharm 2022.3.2, Qt Designer 4.9.1

2. 빌드 방법

A. 빌드 방법 (공통)

1. mobaXterm에서 Session > SSH 연결 생성해서 EC2에 접속

- 도메인: i8a402.p.ssafy.io
- username은 ubuntu로 설정
- mm에서 받은 pem(키)를 반드시 등록

2. EC2에 Git 설치

- `sudo apt-get install git`

3. 배포할 코드를 EC2로 clone 받아오기

- `git clone repository주소`

4. gradlew 권한 변경

- `chmod +x gradlew` (pull 받고 나서 혹시 권한 없다고 오류가 나면 이 명령어를 실행)

B. 백엔드 API Server + 프론트엔드 빌드 방법

1. /home/ubuntu/app/deploy.sh

- 여기에 있는 `deploy.sh` 를 실행하면 바로 Gitlab에서 풀받아서 빌드와 nohup(백그라운드실행) 가능! (+ nohop.out1 파일을 보면 실행 로그 확인 가능)

- `./deploy.sh`

```
#!/bin/bash

cd ~/app/S08P12A402/backend/audiro/

echo "> Git Pull"
git pull

echo "> 프로젝트 Build 시작"
./gradlew build

echo "> step1 디렉토리 이동"
cd ~/app
```

```

echo "> Build 파일 복사"
cp ./S08P12A402/backend/audiro/build/libs/*.jar ~/app

echo "> 현재 구동중인 애플리케이션 pid 확인"
CURRENT_PID=$(pgrep -f S08P12A402.*.jar)

echo "> 현재 구동중인 애플리케이션 pid: $CURRENT_PID"
if [ -z "$CURRENT_PID" ]; then
    echo "> 현재 구동중인 애플리케이션이 없으므로 종료하지 않습니다."
else
    echo "> kill -15 $CURRENT_PID"
    kill -15 $CURRENT_PID
    sleep 5
fi
echo "> 새 애플리케이션 배포"
JAR_NAME=$(ls -tr ~/app/ | grep jar | tail -n 1)

echo "> JAR Name: $JAR_NAME"

nohup java -jar ~/app/$JAR_NAME 2>&1 &

```

C. 백엔드 Chat Server + 빌드 방법

1. /home/ubuntu/app/deploy-chat.sh

- 여기에 있는 [deploy-chat.sh](#) 를 실행하면 바로 Gitlab에서 풀받아서 빌드와 nohup(백그라운드실행) 가능! (+ nohop.out1 파일을 보면 실행 로그 확인 가능)
- .ldeploy-chat.sh**

```

cd ~/app/S08P12A402/backend/audiro-chat

git pull origin dev

./gradlew build

cd ~/app

cp ./S08P12A402/backend/audiro-chat/build/libs/*.jar ~/app

java -jar ~/app/audiro-chat-0.0.1-SNAPSHOT.jar

```

D. 배포 명령어 정리

1. netstat -tnlp | grep 80

- 80이 들어간 PID 출력

2. ps - ef | grep java

- 현재 실행 중인 서버 pid 확인

3. sudo kill -9 <pid>

- 실행 중인 서버 종료

4. sudo cat /var/log/nginx/error.log

- nginx log 확인하는 방법

5. sudo vi /etc/nginx/sites-available/myapp.conf

- nginx proxy 설정하는 방법

3. 프로퍼티 정의

A. Nginx 값 세팅

```

server {
    listen 80;
    location / {
        root /home/ubuntu/app/S08P12A402/frontend/build;
        index index.html index.htm;
        try_files $uri $uri/ /index.html;
    }
    location /api {
        proxy_pass http://localhost:8080;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
    }
    location /oauth2 {
        proxy_pass http://localhost:8080;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
    }
    location /login/oauth2 {
        proxy_pass http://localhost:8080;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
    }
    location /chat {
        proxy_pass http://localhost:8082;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
    }
    location /ws-stomp{
        proxy_pass http://localhost:8082;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
        proxy_set_header Origin "";
    }
    location /channel{
        proxy_pass http://localhost:8082;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
        proxy_set_header Origin "";
    }
}

```

B. Git Ignore 파일

1. application.yml (api server)

```

server:
    port: 8080
    servlet:
        context-path: /
        encoding:
            charset: UTF-8
            enabled: true
            force: true

spring:
    datasource:
        driver-class-name: com.mysql.cj.jdbc.Driver
        username: root
        url: jdbc:mysql://${MYSQL_HOST:localhost}:3306/audiroDB
        password: '1234'
    jpa:
        database: mysql
        hibernate:
            ddl-auto: update
            naming:
                physical-strategy: org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
            show-sql: true

    profiles:
        include: oauth
    mvc:
        path match:
            matching-strategy: ant_path_matcher
        hidden-method:

```

```

    filter:
      enabled: true #delete method 사용하기 위해서 넣음

notification:
  mattermost:
    enabled: true # mmSender를 사용할 지 여부, false면 알림이 오지 않는다
    webhook-url: https://meeting.ssafy.com/hooks/bxhao3segj8dz86qnqe7zgiuje # 위의 Webhook URL을 기입
    channel: # 기본 설정한 채널이 아닌 다른 채널로 보내고 싶을 때 기입한다
    pretext: # attachments의 상단에 나오게 되는 일반 텍스트 문자
    color: # attachment에 왼쪽 사이드 컬러. default=red
    author-name: # attachment의 상단에 나오는 이름
    author-icon: # author-icon 왼쪽에 나를 아이콘의 url링크
    footer: # attachment에 하단에 나올 부분. default=현재 시간

naver-cloud-sms:
  accessKey: ZwsVKLx41vpVx0bviJVy
  secretKey: DdTysXlOfqyDxdr56YFNhEPY9mp8pd0oh9VY09F
  serviceId: ncp:sms:kr:301300936505:audiro
  senderPhone: 01092039218

```

2. application-oauth.yml

```

spring:
  security:
    oauth2:
      client:
        registration:
          google:
            client-secret: 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
            client-id: 111272373115-5vbj6rckde2ntanmcchogcsan9aislv8.apps.googleusercontent.com
            scope:
              - email
              - profile
          naver:
            client-secret: JJBfQD008A
            client-id: xj6iYQrdmJ4LT_8N0kGu
            scope:
              - name
              - email
              - profile_image
            client-name: Naver
            authorization-grant-type: authorization_code
            redirect-uri: http://i8a402.p.ssafy.io:80/login/oauth2/code/naver
          kakao:
            client-secret: 7S4aXXvgDfDeabHg7sDXE3x8bmEVFs7n
            client-id: 36efccc7ac6a65a752e2994d8568eafb
            scope:
              - profile_nickname
              - profile_image
              - account_email
            client-name: Kakao
            authorization-grant-type: authorization_code
            client-authentication-method: POST
            redirect-uri: http://i8a402.p.ssafy.io:80/login/oauth2/code/kakao
        provider:
          naver:
            authorization-uri: https://nid.naver.com/oauth2.0/authorize
            token-uri: https://nid.naver.com/oauth2.0/token
            user-info-uri: https://openapi.naver.com/v1/nid/me
            user-name-attribute: response
          kakao:
            authorization-uri: https://kauth.kakao.com/oauth/authorize
            token-uri: https://kauth.kakao.com/oauth/token
            user-info-uri: https://kapi.kakao.com/v2/user/me
            user-name-attribute: id

```

3. application.yml (chat server)

```

server:
  port: 8082

spring:
  jpa:
    hiberante:
      ddl-auto: update
      show-sql: true
    database: mysql
  datasource:
    url: jdbc:mysql://$${MYSQL_HOST:localhost}:3306/audiroDB
    username: root
    password: 1234

```

```

driver-class-name: com.mysql.cj.jdbc.Driver
redis:
  host: 127.0.0.1
  port: 6379
mvc:
  path match:
    matching-strategy: ant_path_matcher

```

4. EC2 세팅

A. AWS EC2 DB 세팅

1. 세팅을 위한 최신 상태 업데이트

```
sudo apt-get update
```

2. MySQL 설치

```
sudo apt-get install mysql server
```

3. 추가 세팅을 위한 이동 후 편집

```
cd /etc/mysql/mysql.conf.d
```

```
sudo vi mysql.cnf
```

4. 세팅 값 적용을 위한 재시작

```
sudo service mysql restart
```

5. DB 접속

- 계정 : `sudo mysql -u root -p`
- 비밀번호 : 1234

B. Redis 세팅

1. Ubuntu에 Redis 설치

```

curl -fsSL https://packages.redis.io/gpg | sudo gpg --dearmor -o /usr/share/keyrings/redis-archive-keyring.gpg

echo "deb [signed-by=/usr/share/keyrings/redis-archive-keyring.gpg] https://packages.redis.io/deb $(lsb_release -cs) main" | sudo

sudo apt-get update
sudo apt-get install redis

```

2. Redis server 실행

```
redis-server
```

3. Redis client 실행 (옵션)

```

redis-cli

# server가 정상 작동하는지 궁금한 경우
ping

# pong이라고 대답해줌

# 저장된 모든 key 값 확인
keys *

# 특정 키의 값이 궁금한 경우
hgetall key

```

5. 외부 서비스

1. 카카오

서비스의 회원가입/로그인 기능을 카카오로 진행하였습니다. 서비스 핵심 기능에 집중할 수 있으며, 회원가입/로그인의 다양한 절차를 생략할 수 있고 가입 도중 이탈 비율을 낮출 수 있습니다. 사용한 기능은 '카카오 로그인 : 기존 로그인 여부와 상관 없이 로그인'입니다.

과정은 다음과 같습니다.

1. 애플리케이션 추가

- 앱 이름: Audiro
- 사업자명 :?

2. 도메인 등록

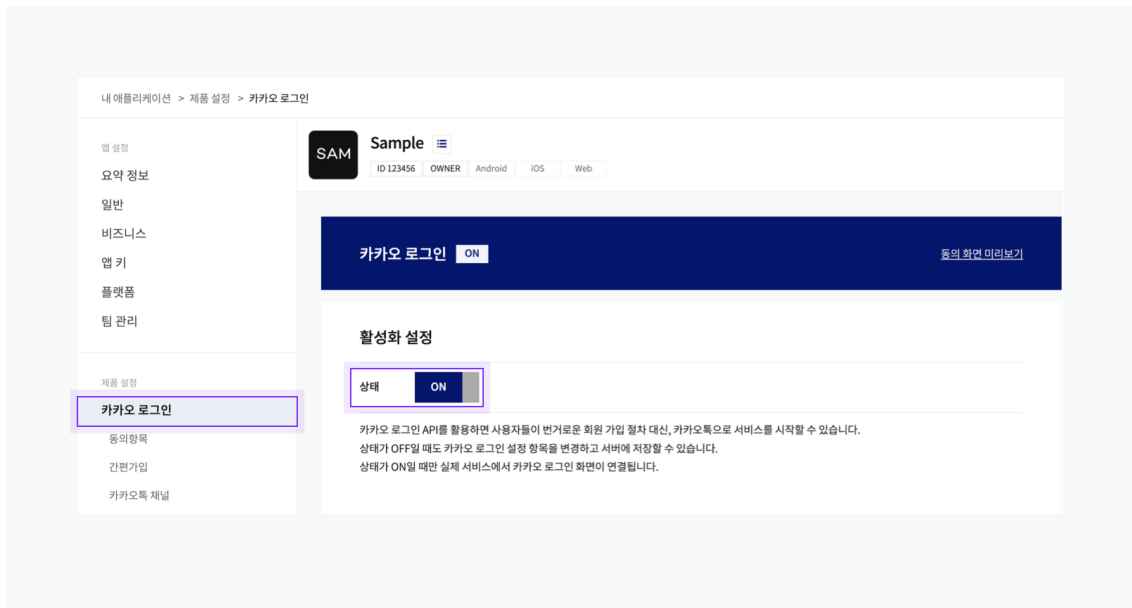
- 앱의 도메인은 <http://i8a402.p.ssafy.io/>입니다

3. Redirect URI 설정

- Redirect URI는 <http://i8a402.p.ssafy.io/home> 입니다. /home 페이지에서 토큰과 로그인 관련 로직을 처리합니다.

4. 로그인 활성화

- 카카오 로그인을 사용하려면 카카오 로그인 활성화 설정의 상태를 ON으로 설정해야 합니다.
-



5. 카카오 로그인 동의항목

- 설정해 둔 동의항목은 프로필 사진, 이메일 입니다.

6. 카카오 인가코드 수신

- KakaoService . java 내 `getAccessToken` 함수와 `getToken` 함수 내 `client_id` 값에 해당 REST API 키를 설정해주어야 합니다.

b. 구글 로그인

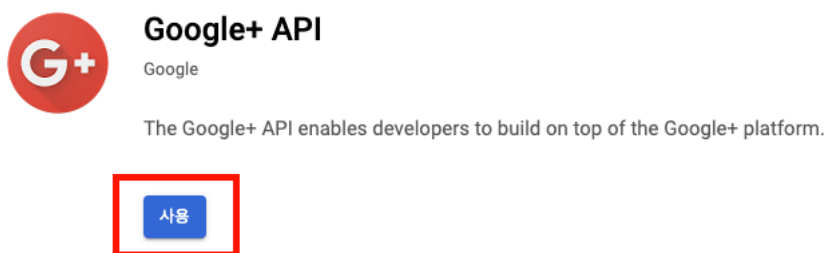
구글 소셜 로그인을 적용하려면 아래의 단계별 방법에 따라 설정합니다.

1. Google API 콘솔 로그인 후 동의

- a. Google API 콘솔에 접속 후 구글 계정으로 로그인합니다. 동의 및 계속하기를 클릭합니다.

2. 라이브러리 추가하기

- a. Google+ API 페이지에서 사용을 클릭합니다.



3. OAuth 동의 화면 구성하기

- a. 사용자 인증정보에서 User Type을 외부로 선택하고 만들기합니다.
- b. 앱 정보 항목에 앱 이름 (audiro) 입력하고 사용자 지원 이메일을 선택합니다.
- c. 앱 도메인 항목에서 애플리케이션 홈페이지, 개인정보처리방침 링크, 애플리케이션 서비스 약관 링크를 작성합니다.

4. 범위 추가하기

- a. 범위 추가 또는 삭제란에서 기본 구글 계정 이메일 주소확인을 선택하고 업데이트 합니다.

5. OAuth 클라이언트 ID 만들기

- a. 애플리케이션 유형을 Chrome웹으로 선택하고 이름 (audiro), 승인된 리디렉션 URI(<http://i8a402.p.ssafy.io/home>)을 입력합니다.

6. 앱 게시하기

a. OAuth 동의 화면에서 앱 게시 버튼을 클릭합니다.

c. 네이버 문자 서비스

네이버 클라우드 플랫폼 서비스에서 제공하는 문자 전송이나 예약 등 및 그 결과를 확인할 수 있는 RESTful API 입니다.URL은 <https://sens.apigw.ntruss.com/sms/v2> 입니다.

참조: <https://api.ncloud-docs.com/docs/ai-application-service-sens-smsv2>

아래의 과정에 따라 설정합니다.

참조 사이트에서 Ncloud platform 인증키 및 signature 생성가이드를 따라 갑니다.

1. 인증키 생성

a. Ncloud 플랫폼 포털의 마이페이지 > 계정 관리 > 인증키 관리로 접속합니다.

b. 인증키 관리 메뉴에서 신규 인증키 생성을 클릭합니다.

c. 발급된 인증키를 보기 하여 복사하여 사용합니다.

2. 헤더 생성

a. 헤더는 세 종류가 있습니다. SMS API의 헤더 설정을 따라갑니다.

3. 호출

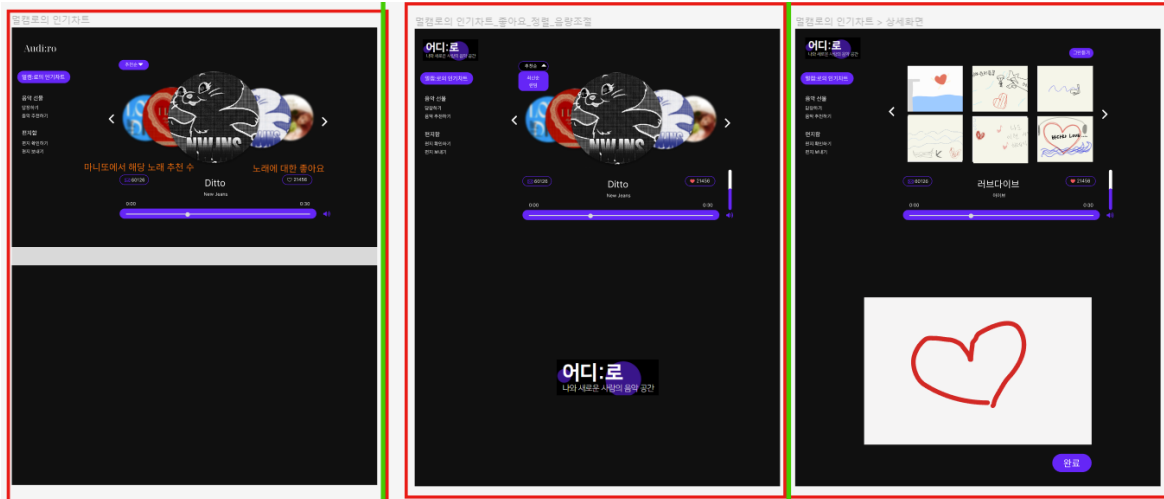
a. 일반적인 RESTFUL api 요청과 같이 위에서 설정한 헤더와 body로 호출합니다.

6. IoT 포팅 매뉴얼

A. 설치 라이브러리 (Qt & Raspberry Pi)

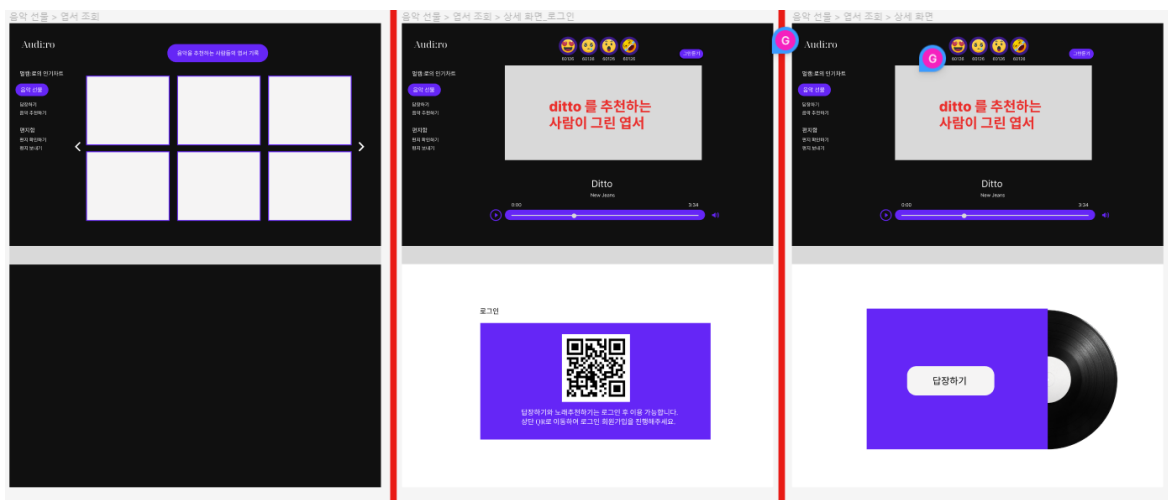
- PyQt5
- PyQt5-Qt5
- PyQt5-sip
- PyQt5Designer
- imageio
- decorator
- pyqt5-plugins
- pyqt5-look
- qt5-applications
- qt5-tools
- requests
- tqdm
- urllib3

B. 화면 별 기능 설명



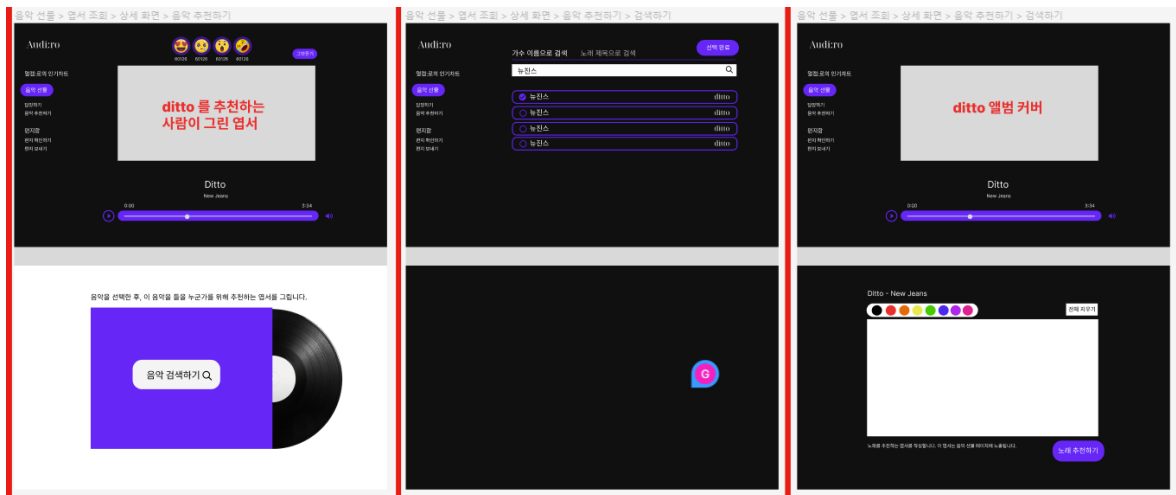
1. 음악 플레이리스트

- 해당 기기에서 가장 많이 추천된 노래 10곡이 표시된다.
- 앨범 아트를 클릭하면 해당 노래가 흘러나오며, 이 음악을 추천한 사람들의 추천 엽서가 보여진다.
- 음악을 들으며 낙서를 할 수 있는 그림판이 나온다.



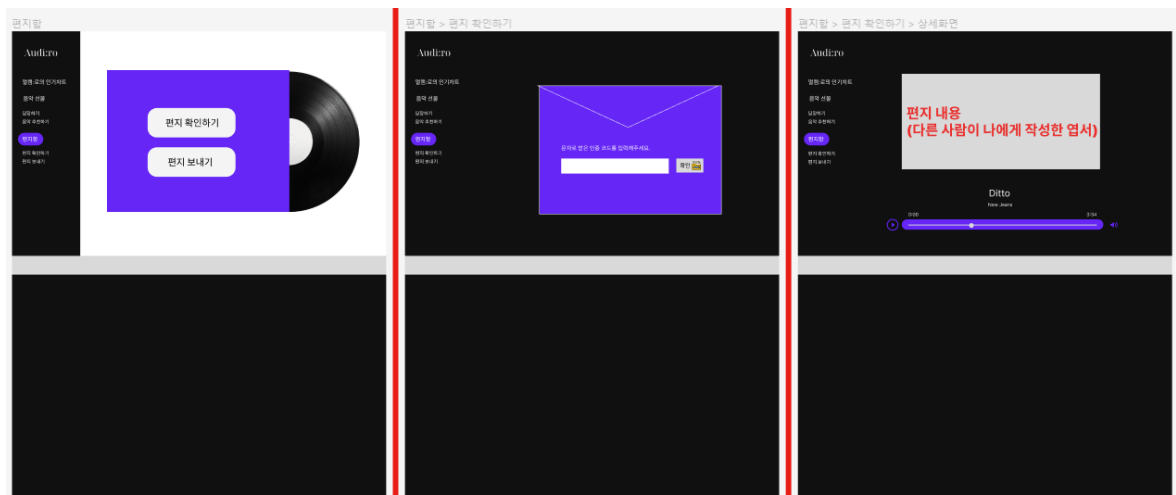
2. 음악 추천1

- 기기에 사람들이 남긴 추천 엽서가 표시된다.
- 엽서를 누르면 어떤 곡을 추천했는지 볼 수 있다.
- 답장 기능으로 연결된다.



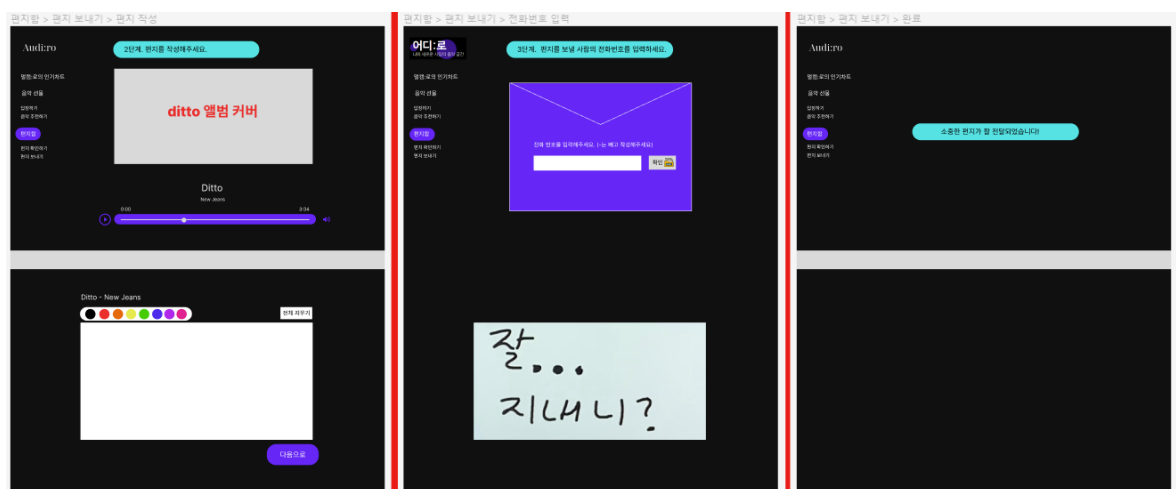
3. 음악 추천2

- 노래를 선택하고, 그림을 그려서 기기에 추천 엽서를 남길 수 있다.



4. 음악 편지 보내기

- 전화번호를 입력한 상대방에게 음악 편지를 남길 수 있다.
- 원하는 비밀번호를 설정해 상대방에게 문자로 발송한다.



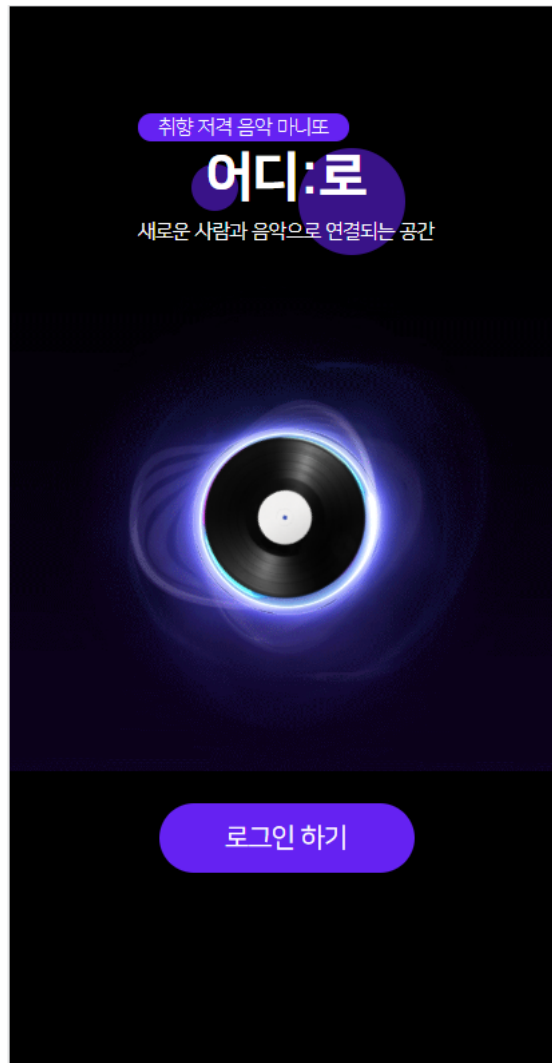
5. 음악 편지 받기

- 문자로 안내 받은 지점에 방문해 비밀번호를 입력하면 나에게 남겨진 음악 편지를 확인할 수 있다.

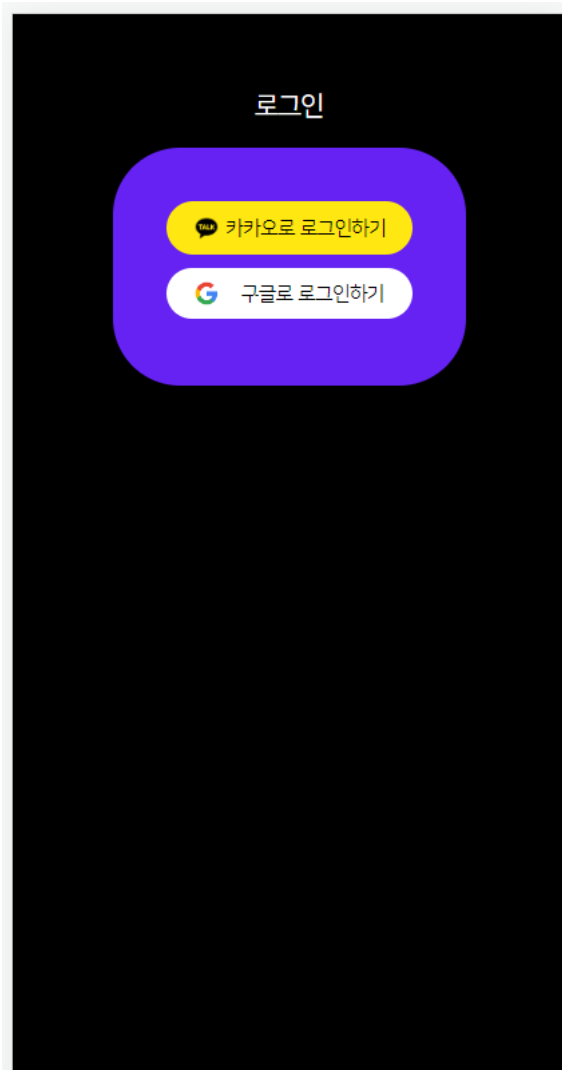
7. WEB 포팅 매뉴얼

로그인 버튼 두 개중 하나(시연에서는 카카오톡으로 로그인하기)를 누릅니다.

1. 웹 접속 및 소셜 로그인하기 , 주소: <http://i8a402.p.ssafy.io/>

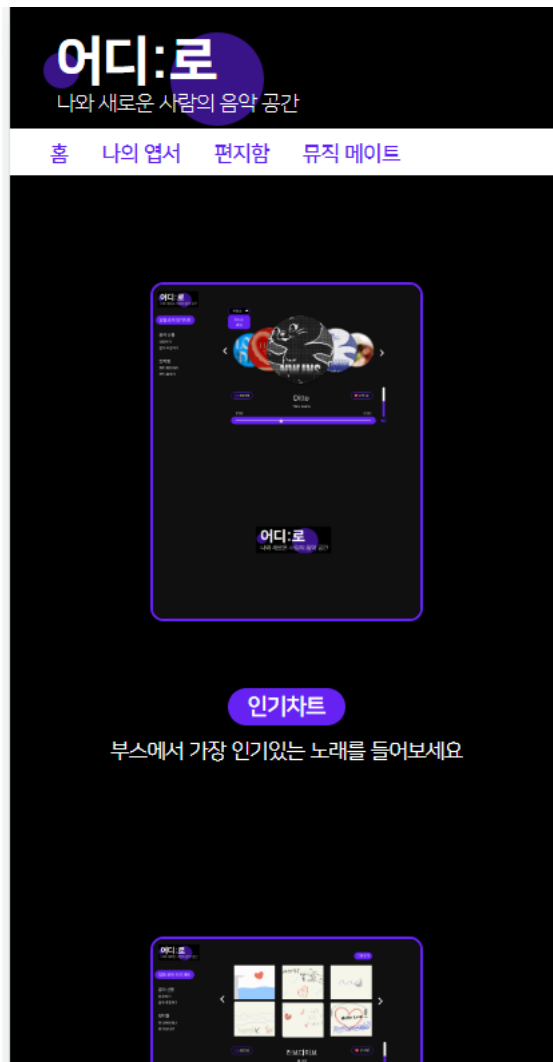


- 로그인 하기를 클릭합니다.
- 두 로그인 버튼 중 하나 (시연에서는 카카오톡)를 클릭

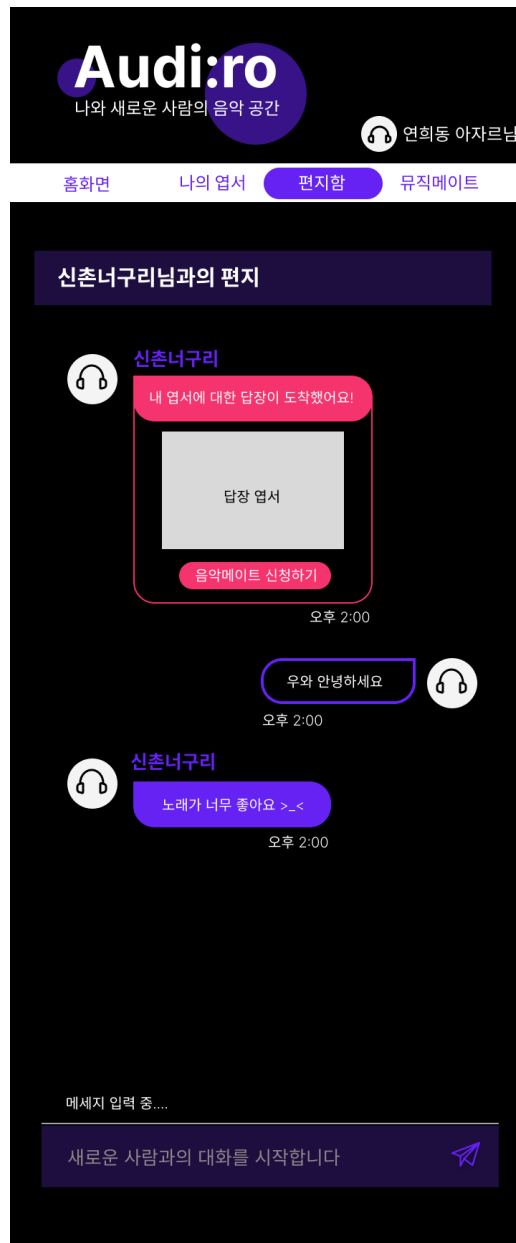


2. 홈화면 - 간단한 서비스 소개

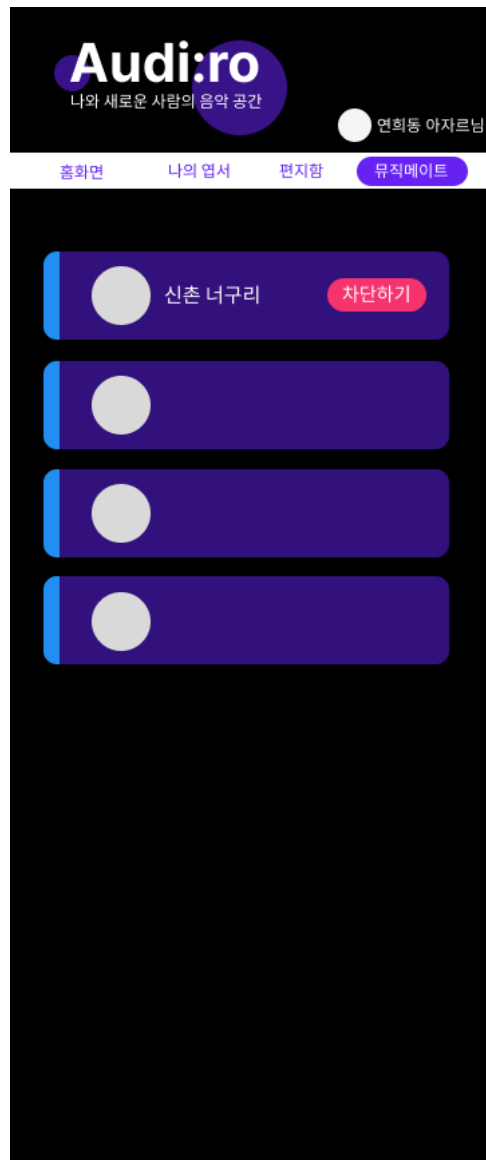
- 기능을 확인합니다.
- 편지함을 클릭합니다.



3. 채팅방에서 답장 온 것 확인



4. 채팅 진행
 - a. 인풋창 활용하여 진행
5. 뮤직메이트 신청하기
 - a. 음악메이트 신청하기 클릭
6. 뮤직메이트 목록보기
 - a. 원하는 사람 눌러서 피드목록으로 접속



7. 피드구경하기

- a. 리스트에서 원하는 것 골라 클릭
- b. 하트나 리액션등을 눌러봄 후 뒤로가기

반가워요, 연희동 아자르님 🙌



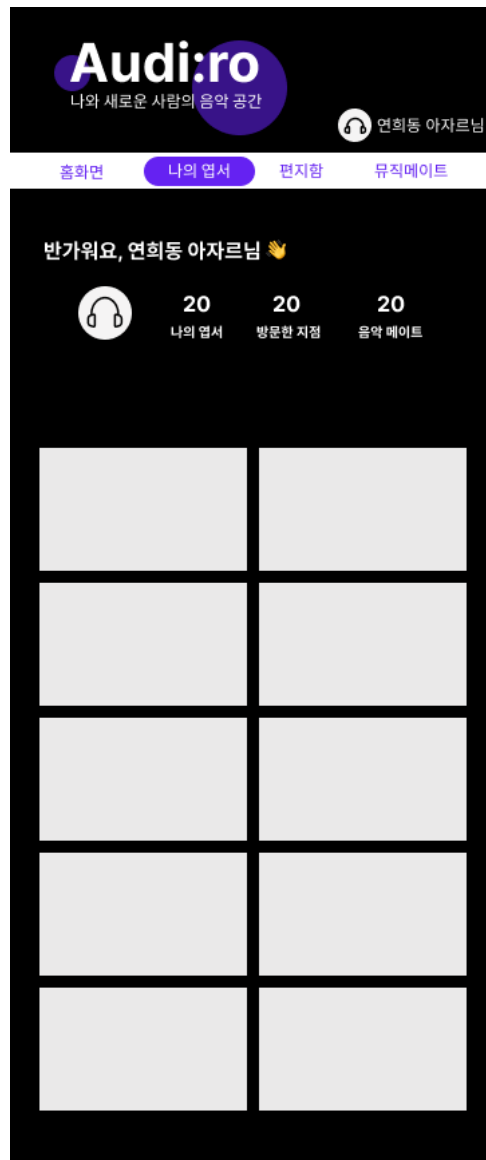
20
나의 업서

20
방문한 지정

20
음악 레이트



8. 음악메이트에서 타겟 삭제하기 버튼 클릭
9. 나의 업서목록
 - a. 원하는 게시물 클릭



10. 게시물 삭제하기

- a. 삭제하기 버튼 클릭



포팅매뉴얼 7