



Open API를 활용한 GPS인증 하이브리드앱 개발

**소품샵 투어 : 소담**

# 목차

- 01. 프로젝트 개요
- 02. 프로젝트 팀 구성 및 역할
- 03. 프로젝트 수행 절차 및 방법
- 04. 프로젝트 수행 결과
- 05. 자체 평가 의견



# 프로젝트 개요

- 프로젝트 주제 및 기획의도
- 프로젝트 기술서
- 개발 환경
- 프로젝트 구조
- 활용방안 및 기대 효과



# 01 프로젝트 개요

## ▶ 프로젝트 주제 및 기획 의도 : 소품샵 정의

### 「소품샵」

- 작은 물건들을 판매하는 곳
- 주인의 취향과 판매하는 물품에 따라 가게 분위기가 천차만별



물비늘



사사로운

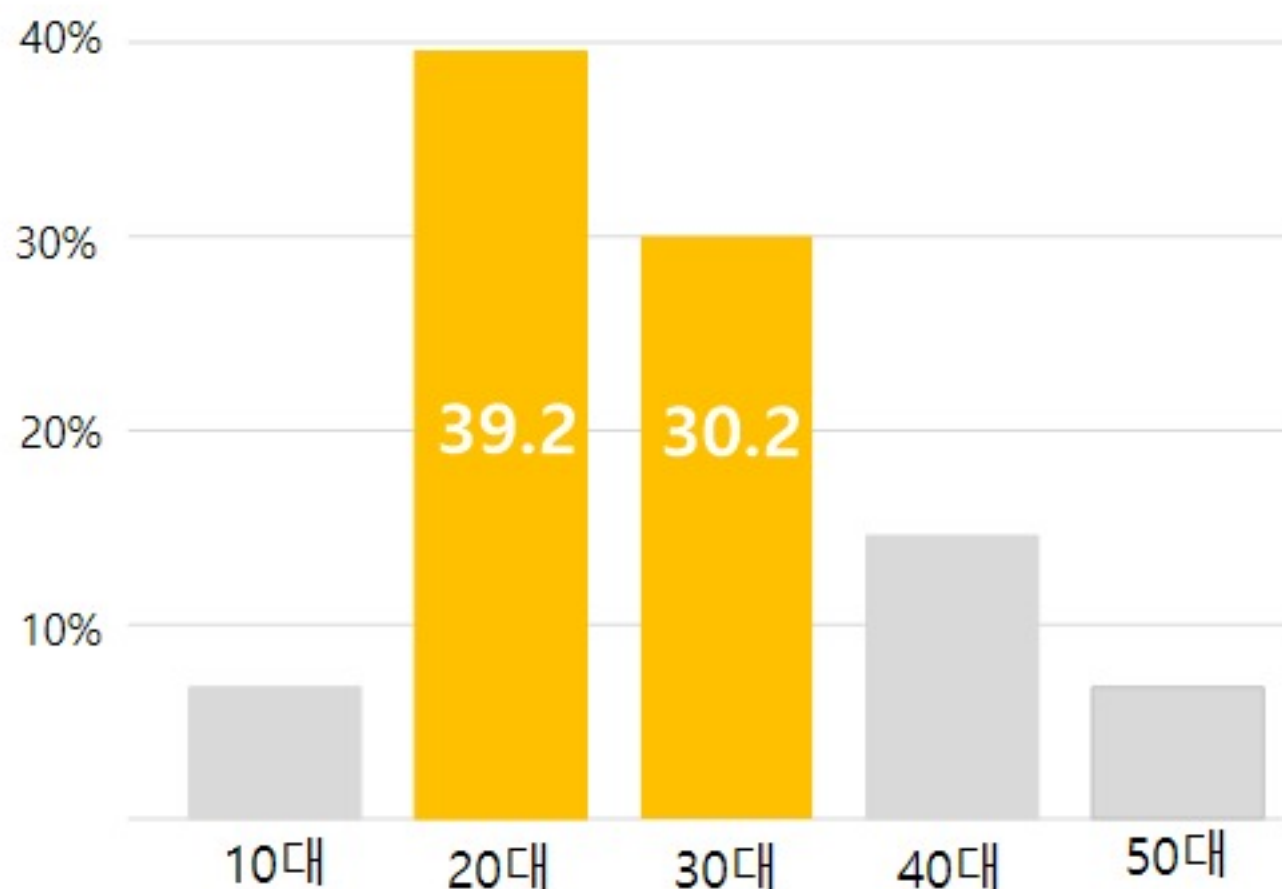


플레이

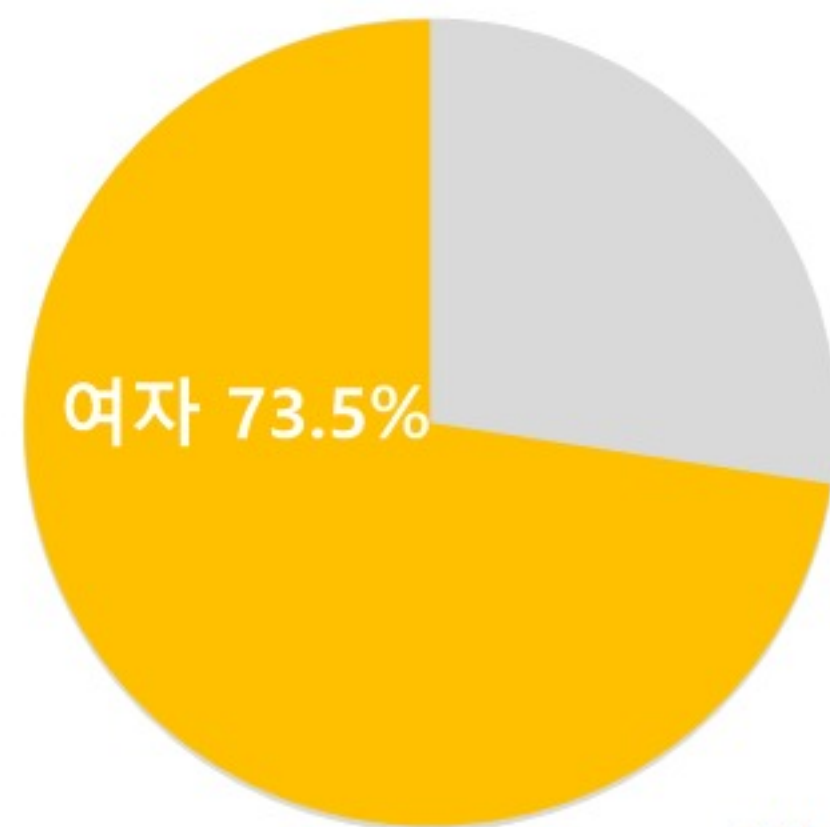
# 01 프로젝트 개요

▶ 프로젝트 주제 및 기획 의도 : 대구 여행 검색 시 검색 비율

「 연령별 검색 비율 」



「 성별 검색 비율 」



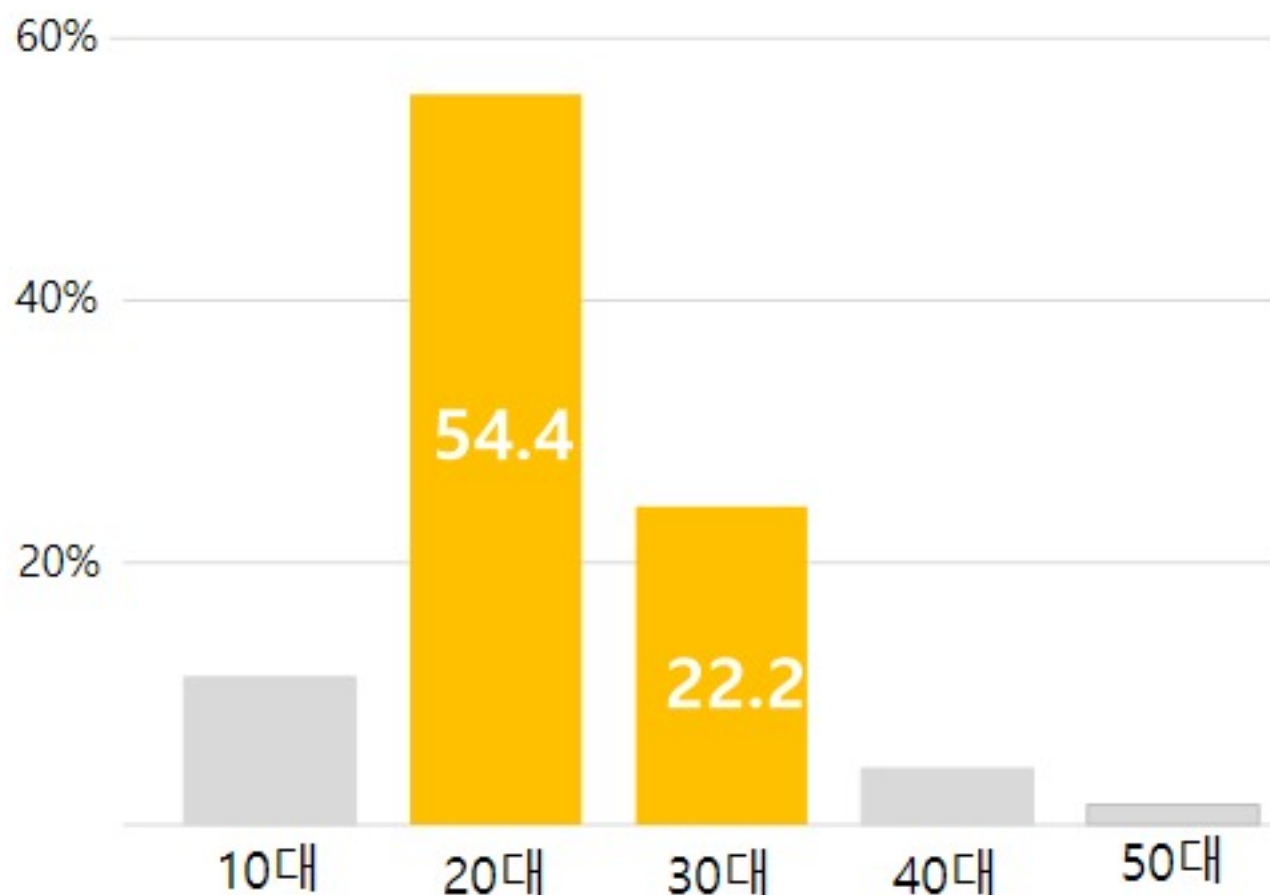
출처: 데이터 분석 툴  
블랙키위



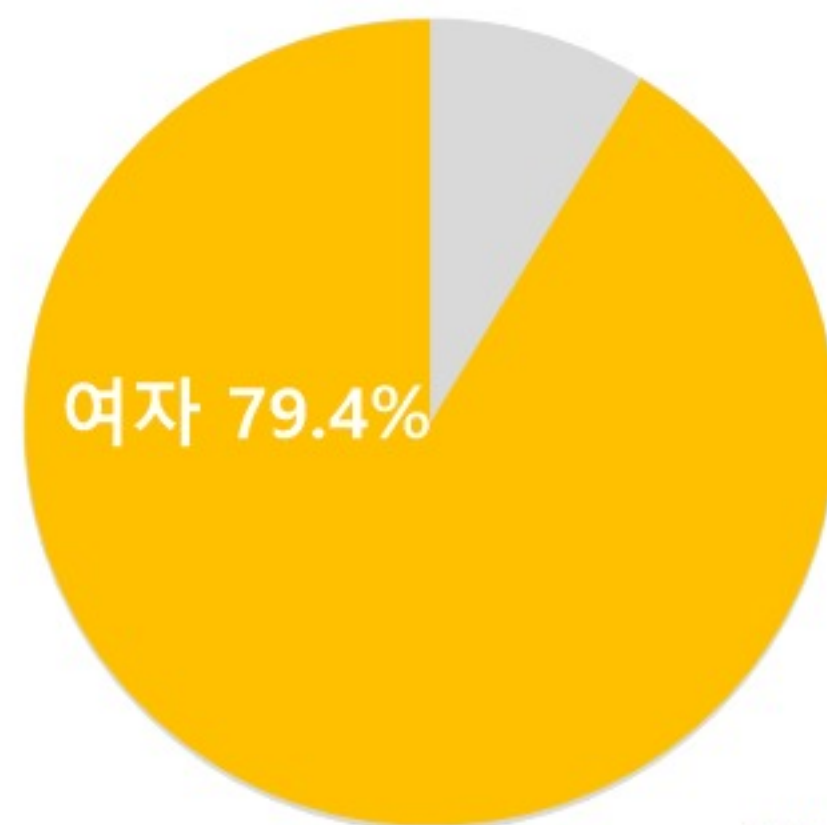
# 01 프로젝트 개요

▶ 프로젝트 주제 및 기획 의도 : 대구 소품샵 검색 시 검색 비율

「 연령별 검색 비율 」



「 성별 검색 비율 」



출처: 데이터 분석 툴  
블랙키위

# 01 프로젝트 개요

▶ 프로젝트 주제 및 기획 의도 : 대구 소품샵 검색 시 검색 비율

「연령별 검색 비율」

60%

40%

20%

10대

20대

30대

40대

50대

- 대구를 여행하려는 사람들에게  
대구 소품샵의 관심도가 높음

- 소품샵이 2·30대 여성-MZ세대  
를 대상으로 하는 상권임

「서버 거래 비율」

출처: 데이터 분석 툴  
블랙키위

# 01 프로젝트 개요

## ▶ 프로젝트 주제 및 기획 의도 : 대구 소품샵 상권의 성장 이유



롤드페인트

- 최근엔 **MZ 세대** 등 젊은 세대를 대상으로 한 점포가 많아지고 있음
- 바이럴 마케팅 등 **SNS 파급력**이 크기 때문

한국경제

김영갑 한양 사이버대학교 교수



# 01 프로젝트 개요

▶ 프로젝트 주제 및 기획 의도 : 유튜브와 인스타그램에서 소품샵의 언급



대구여행 검색 결과 상위 영상(유튜브)



>>>> 대구 데이트 플코스 모음집.jpg 대구 현지인이 추천하는 대구 여행 플코스><  
대구맛집 #대구카페 #대구맛집추천 #대구카페추천 #대구맛집 #대구놀이거리  
#대구편맛집 #대구소품샵 #대구술집 #대구여행 #동성로맛집 #동성로카페

인기 포스트(인스타그램)

# 01 프로젝트 개요

## ▶ 프로젝트 주제 및 기획 의도

### 「 주제 」

GPS인증 하이브리드앱 개발 「대구 소품샵 투어 : 소담」 제작

### 「 기획 의도 」

하이브리드 앱 구현  
스탬프 투어 방식으로 대구 소품샵 방문  
GPS 인증 기술 사용

→ 매체성 회복  
홍보 기능 강화  
간편한 위치파악



# 01 프로젝트 개요

## ▶ 프로젝트 주제 및 기획 의도

현업 이슈	<ul style="list-style-type: none"><li>- '소품샵' 은 대구를 여행하려는 사람들에게 관심이 높은 장소</li><li>- '소품샵' 의 업종이 다양한 분류로 분산되어 있어 정보 수집이 용이하지 않음 (인테리어소품, 종합도소매, 문구팬시용품, 패션잡화, 식료품 등)</li></ul>
문제	<ul style="list-style-type: none"><li>- '소품샵' 정보 검색의 편의성을 높여줄 매개체 부재로 '소품샵' 정보 검색이 불편</li><li>- 알려지지 않은 '소품샵'이나 변화가에서 떨어져 있는 '소품샵'의 홍보 부족</li></ul>



# 01 프로젝트 개요

## ▶ 프로젝트 기술서

### 프로젝트 구현 내용 및 컨셉

- 수정이 간편한 하이브리드 앱으로 매체성 제공
- 대구 소품샵의 홍보를 강화하여 상권의 활성화
- GPS 기술과 QR코드를 이용하여 스탬프 투어 방식으로 쿠폰을 얻을 수 있음

### 훈련 내용 관련 기술

#### [하이브리드 앱 구현]

- node.js를 활용한 앱 제작, 최신 트렌드에 맞는 UX/UI 디자인 구현

#### [Open API를 활용한 GPS 인증 구현]

- 위치 신호를 수신, QR코드로 인증 기술 구현

#### [데이터베이스 설계 및 구현]

- 회원 가입 및 로그인 기능 구현

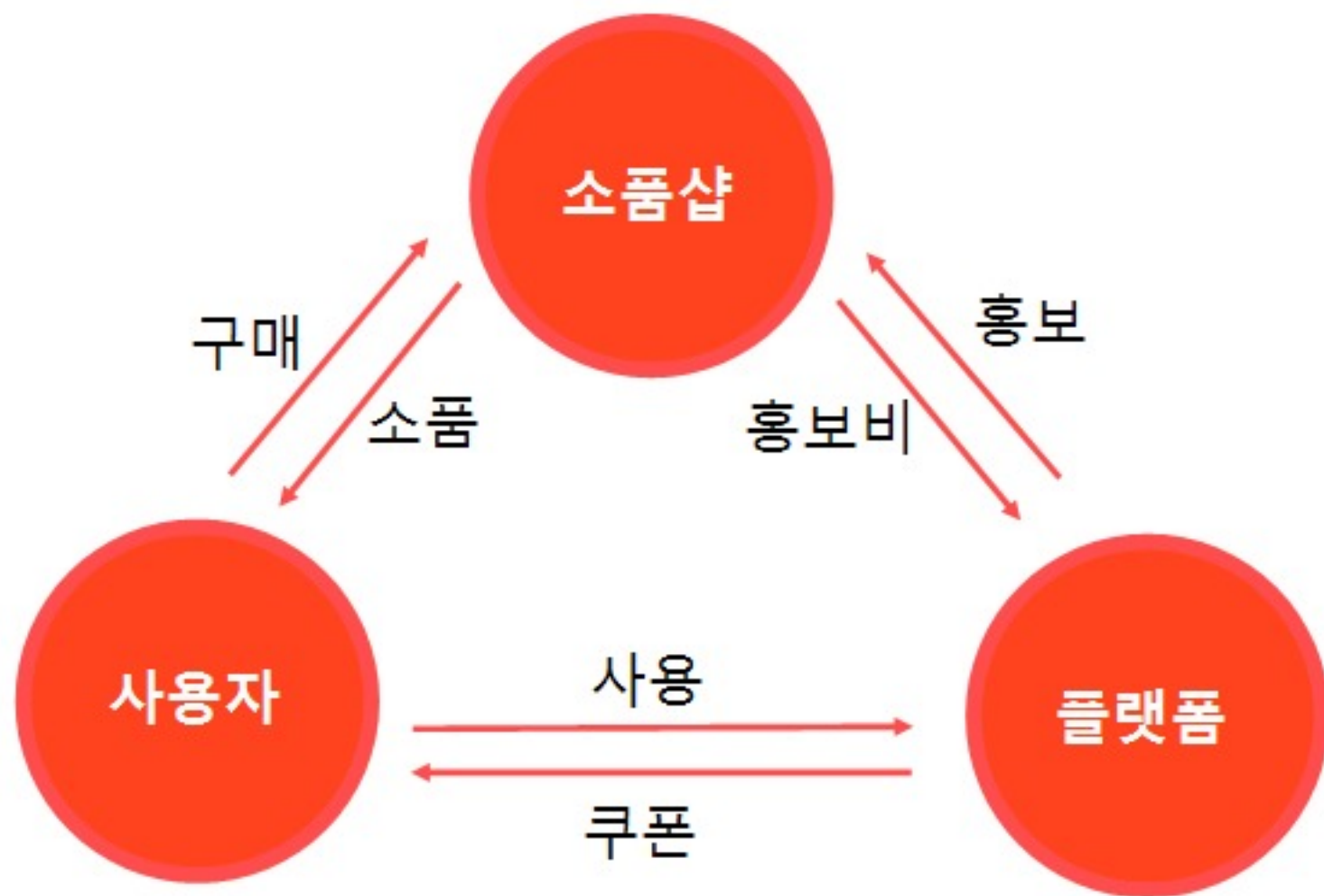
#### [형상관리 도구 활용 및 앱 배포]

- 깃 허브 활용 및 apk 제작

# 01 프로젝트 개요

## ▶ 활용 방안 및 기대 효과

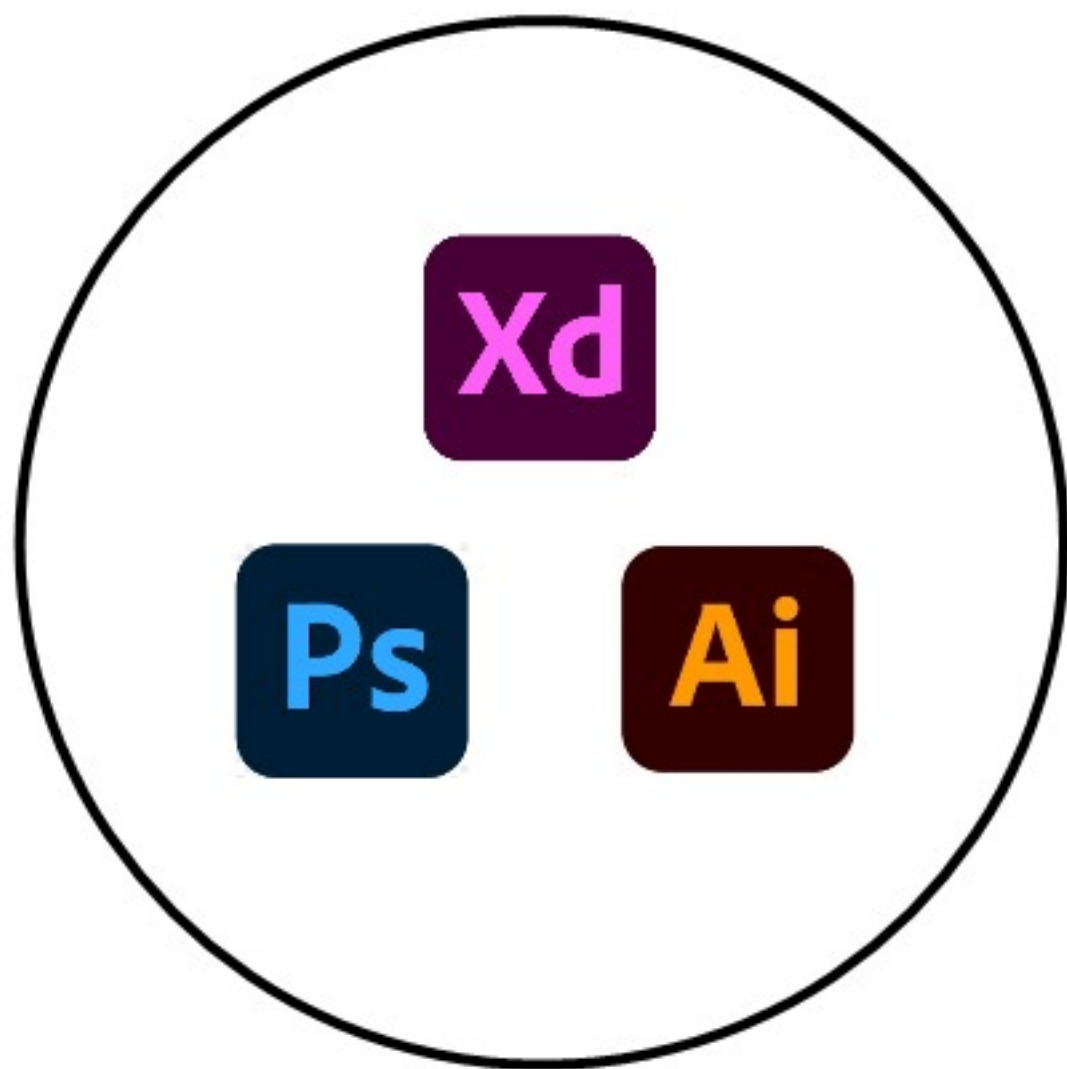
- 1 대구 소품샵을 방문하려는 사람들의  
**정보 편의성 기대**
- 2 가입한 소품샵의 홍보비로  
**플랫폼 수익 증가**
- 3 스탬프 투어 방식으로 받은 쿠폰 사용으로  
대구소품샵에서의 **구매율 증가**



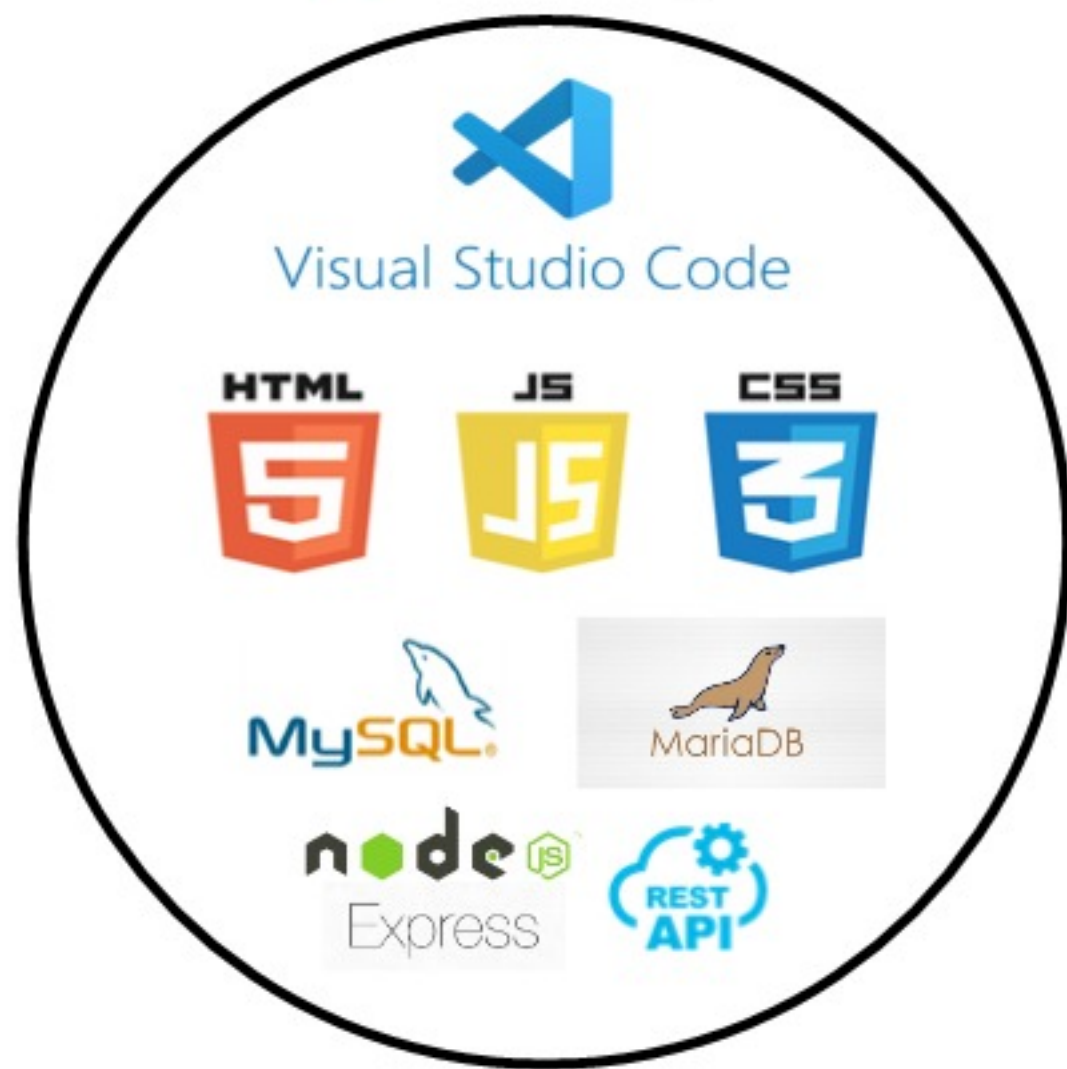
# 01 프로젝트 개요

▶ 개발 환경 (개발 언어 및 툴)

## Design



## Develop





# 01 프로젝트 개요

## ▶ 프로젝트 구조 (워크플로우)





# 프로젝트 팀 구성 및 역할

## 02 프로젝트 팀 구성 및 역할

### ▶ [프로젝트 팀 구성 및 역할]

	정민혜	윤진실	김현정	권지현
역할	웹퍼블리싱 백엔드	디자이너	웹퍼블리싱	기획
담당업무	<ul style="list-style-type: none"><li>- 웹퍼블리싱</li><li>- 백엔드</li><li>- 로그인 , 회원가입 기술 구현</li><li>- 메뉴 제작</li></ul>	<ul style="list-style-type: none"><li>- UXUI 디자인</li><li>- 로고 제작</li><li>- 영상 제작</li></ul>	<ul style="list-style-type: none"><li>- 웹퍼블리싱</li><li>- 백엔드</li><li>- GPS 기술 구현</li><li>- APK 배포</li></ul>	<ul style="list-style-type: none"><li>- 개발보고서 ( PPT ) 제작 및 발표</li><li>- 와이어프레임 제작</li></ul>





# 프로젝트 수행 절차 및 방법

# 03 프로젝트 수행 절차 및 방법



## 프로젝트 수행 절차 및 방법

구분	활 동	2월			비고
		1주	2주	3주	
기획 및 설계	주제 선정 및 벤치마킹	<div></div>			DB 오류와 아직 구현되지 않은 페이지가 있어 계속 수정해 나갈 예정
	워크플로우 및 와이어프레임		<div></div>		
	스토리보드		<div></div>		
UI/UX 디자인	스타일가이드		<div></div>		
	앱 디자인		<div></div>		
	수정 및 보완			<div></div>	
하이브리드 앱	메뉴		<div></div>		
	본 화면 구현		<div></div>		
	수정 및 보완		<div></div>		
API 및 기술 개발	자바스크립트			<div></div>	
	Open API		<div></div>		
	데이터베이스			<div></div>	
	수정 및 보완			<div></div>	



# 프로젝트 수행 결과

- UI/UX 설계
- UI/UX 디자인
- 하이브리드 앱 구현
- GPS API 기술 개발
- 데이터베이스 설계 및 구현
- 형상관리 도구 활용 및 앱 배포
- 시연 동영상



# 04 프로젝트 수행 결과

## ▶ UI/UX 설계 (와이어프레임)

### SPLASH



### LOGIN



### SIGN UP



### HOME



### MAP



## 04 프로젝트 수행 결과

### ▶ UI/UX 설계 (와이어프레임)

QR



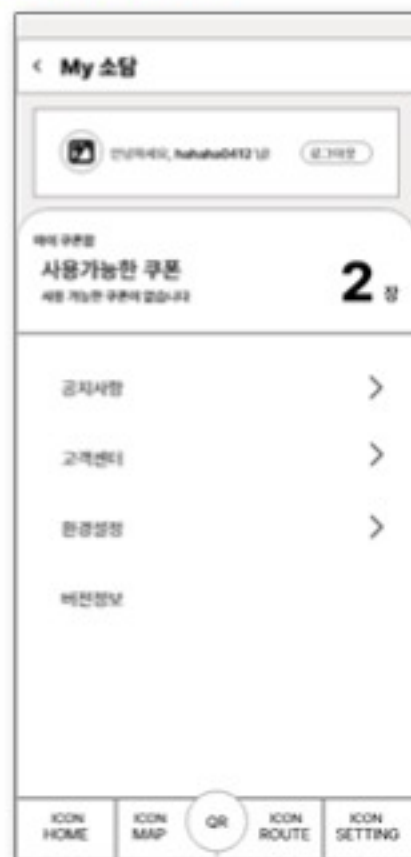
## 04 프로젝트 수행 결과

### ▶ UI/UX 설계 (와이어프레임)

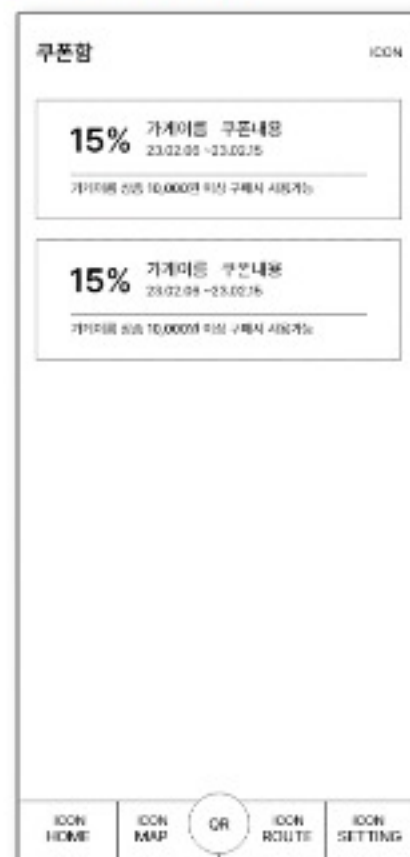
#### ROUTE



#### MYPAGE



#### COUPON





## 04 프로젝트 수행 결과

### ▶ UI/UX 디자인 (스타일 가이드)

#### COLOR

색상



#F55050

#E52538



#CCCCCC

#666666

#333333

#### FONT

폰트

**TT투게더 - 50**

**TT투게더 - 30**

**TT투게더 - 20**

**TT투게더 - 18**

Pretendard - Semibold 20

**Pretendard - Extrabold 16**

Pretendard - Regular 14

Pretendard - Regular 8

## 04 프로젝트 수행 결과

### ▶ UI/UX 디자인 (스타일 가이드)

**LOGO** 로고의 귀여움으로 2·30대 여성의 흥미를 이끌



과일의 **상큼한** 이미지를 차용

**선물상자**에서 영감을 얻음

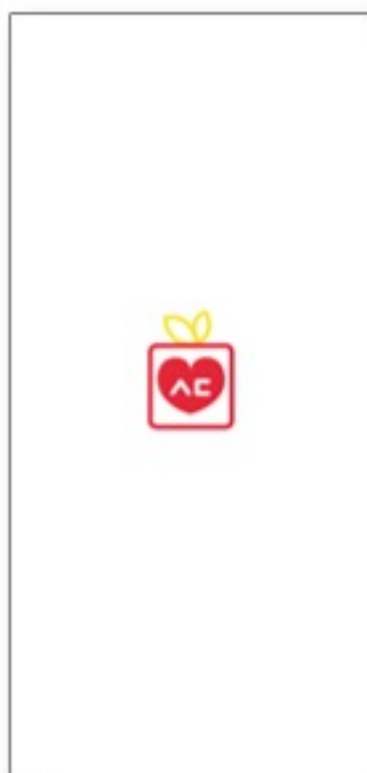
**소담**의 자음

**So: 품을  
Dam 다**

# 04 프로젝트 수행 결과

## ▶ UI/UX 디자인 (프로토타입)

### SPLASH



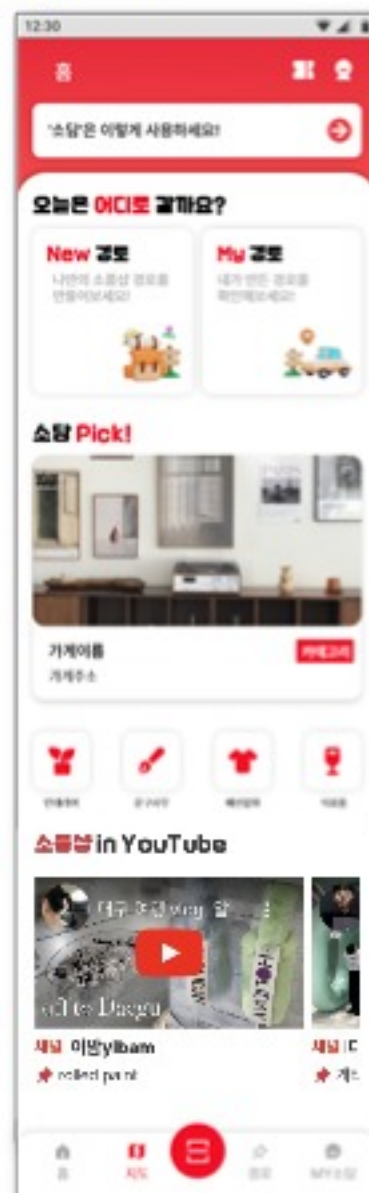
### LOGIN



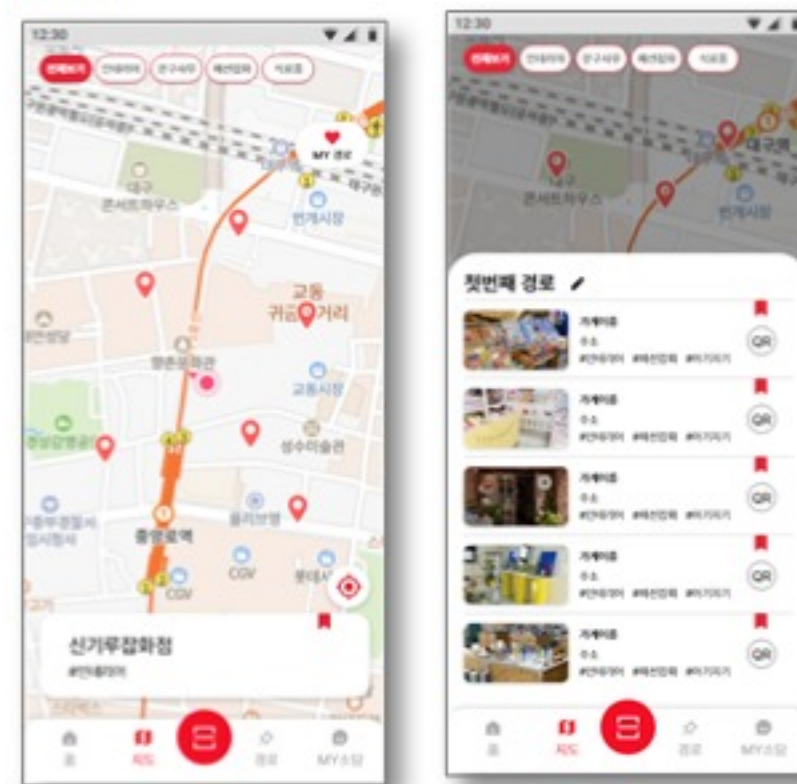
### SIGN UP



### HOME



### MAP

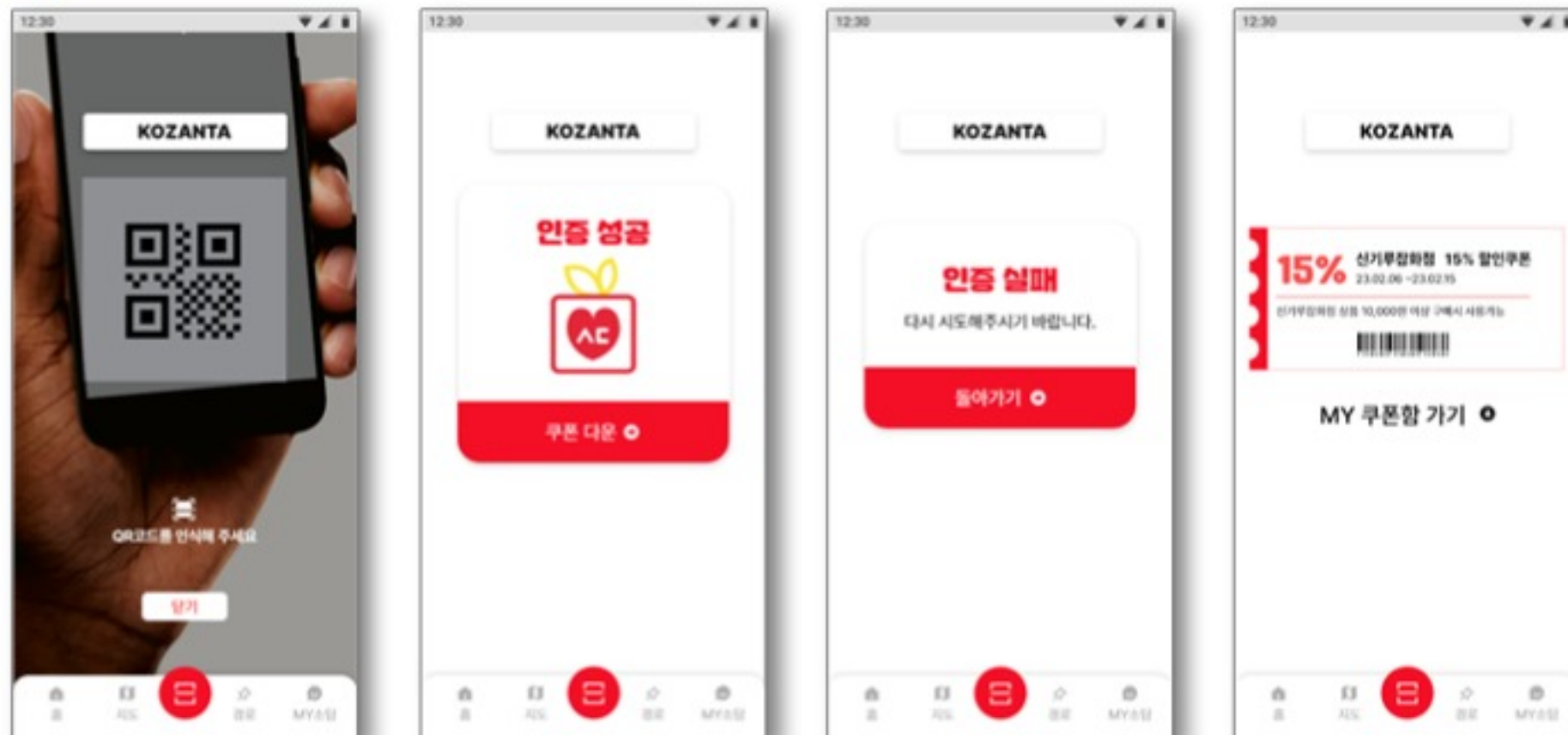




## 04 프로젝트 수행 결과

### ▶ UI/UX 디자인 (목업)

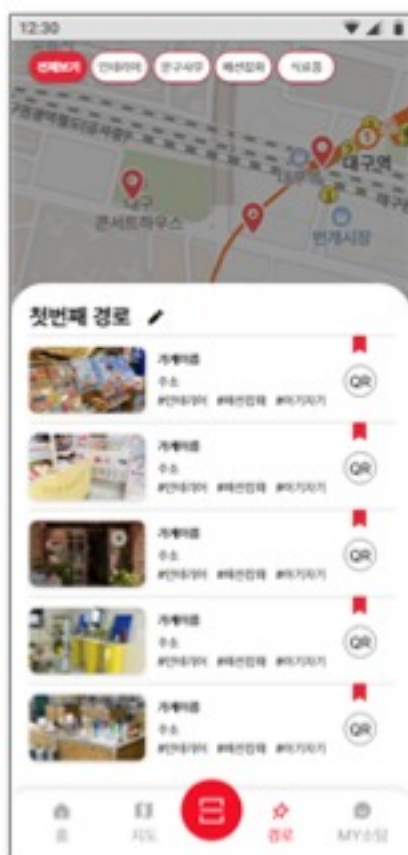
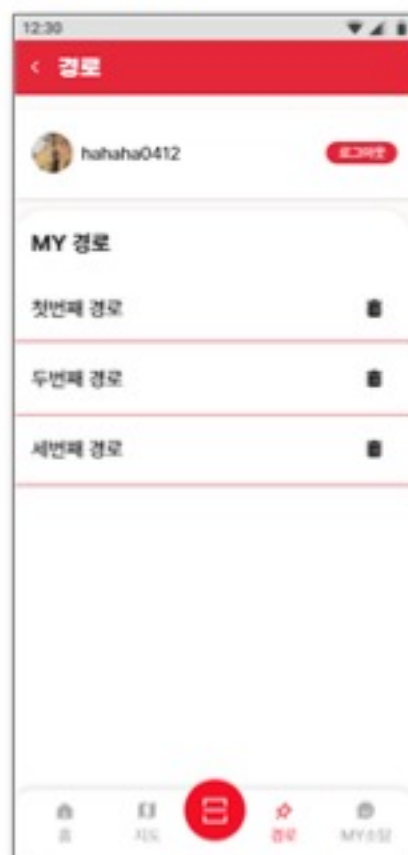
QR



## 04 프로젝트 수행 결과

### ▶ UI/UX 디자인 (목업)

#### ROUTE



#### MYPAGE



#### COUPON



## 04 프로젝트 수행 결과

### ▶ 하이브리드 앱 구현 ( HTML, CSS, JS, Node.js )

-앱 하단 네비게이션 구현

```
const toastMessage = (message) => {
  Swal.mixin({
    toast: true,
    position: "bottom",
    showConfirmButton: false,
    timer: 2000,
  }).fire({ title : `${message}`, icon: "error"})
}

const phoneTypeCheck = () => {
  if(/iPhone|iPad|iPod/i.test(navigator.userAgent)) return "ios"
  else if(/Android/i.test(navigator.userAgent)) return "android"
  else return "pc"
}

const movePage = (e) => {
  const clickUrl = e.currentTarget.dataset.url;
  if (clickUrl === "/qr" && phoneTypeCheck() !== "pc") return window.HybridApp.QR_Btn();
  location.href = clickUrl;
}

const gnbMenu = document.querySelectorAll(".gnbMenu");
gnbMenu.forEach((menu) => {
  const url = window.location.pathname;
  if(menu.dataset.url === url) menu.classList.add("pageOn");
  else menu.addEventListener('click', movePage)
});
```

```
const output = async (code) => {
  const data = await (await fetch("/qr", {
    method: "POST",
    headers: {
      "Content-Type": "application/json",
      "Accept": "application/json"
    },
    body: JSON.stringify({ code }),
  })).json();
  if(data.code === "OK") location.href = "/course";
  else toastMessage(data.message);
}
```





## 04 프로젝트 수행 결과

### ▶ 하이브리드 앱 구현 ( HTML, CSS, JS, Node.js )

-소담 랜덤 픽 구현

```
$(function(){
  var homePickLink = new Array();
  homePickLink[homePickLink.length] = [
    "https://www.instagram.com/getasens/",
    "getasens",
    "게타상스",
    "패션잡화",
    "대구광역시 중구 국채보상로150길 76-12 3층"
  ];
  homePickLink[homePickLink.length] = [
    "https://www.instagram.com/nice_kitchen /",
    "nicekitchen",
    "나이스키친",
    "식료품",
    "대구광역시 중구 동성로5길 41 1층"
  ];
  homePickLink[homePickLink.length] = [
    "https://www.instagram.com/i want to be an oasis/",
    "oasis",
    "신기루잡화점",
    "문구사무",
    "대구광역시 중구 중앙대로 457 2층"
  ];
  homePickLink[homePickLink.length] = [
    "https://www.instagram.com/from.tds/",
    "tds",
    "티디에스",
    "인테리어",
    "대구광역시 남구 양지로14길 7 1층"
  ];
});
```

```
homePickLink[homePickLink.length] = [
  "https://www.instagram.com/official...play//",
  "play",
  "플레이",
  "문구사무",
  "대구광역시 수성구 지범로17길 74 1층"
];

var homePickLinkOrder = Math.floor(Math.random()*(homePickLink.length));
homePickLinkOrder = (isNaN(homePickLinkOrder)) ? 0 : homePickLinkOrder;
$('.homePickLink').attr('href', homePickLink[homePickLinkOrder][0]);
$('.homePickLink img').attr('src', '/client/img/homepick_' + homePickLink[homePickLinkOrder][1] + '.jpg');
$('.homePickLink img').attr('alt', homePickLink[homePickLinkOrder][1]);
$('.homePickName').text(homePickLink[homePickLinkOrder][2]);
$('.homePickCategory').text(homePickLink[homePickLinkOrder][3]);
$('.homePickAddress').text(homePickLink[homePickLinkOrder][4]);
});
```

#### 소담's Pick!



#### 게타상스

대구광역시 중구 국채보상로150길 76-12 3층

패션잡화

- ▶ 앱을 킬 때마다 다른 소품샵을 추천
- ▶ 배열의 인덱스를 랜덤 함수를 써 가게들이 다르게 나오게 함

## 04 프로젝트 수행 결과

### ▶ GPSAPI 기술개발 (Rest API, Json)

```
"use strict";

import mysql from "mysql2";
import "dotenv/config"; // 환경변수 설정

// 설정파일
const config = {
  host: process.env.DB_HOST,
  port: process.env.DB_PORT,
  user: process.env.DB_USER,
  database: process.env.DB_DATABASE,
  password: process.env.DB_PSWORD,
};

// DB 커넥션
const db = mysql.createPool(config).promise();

export default db;
```

- ▲ 1) MySQL에서 받아온 데이터 가져옴
- ▶ 2) 가져온 데이터를 다른 파일에서 쓸 수 있도록 객체를 내보냄

```
import db from '../config/db.js';

export class DbMainRepository {
  async findAllCourse() {
    const QUERY = `SELECT num, name, code, address,
latitude, longitude, category, visited FROM course`;
    return db.execute(QUERY).then((result) => result[0]);
  }
}
```

```
import { DbMainRepository } from "../model/DbMainRepository.js";

export default class MainService {
  // static repository = MemoryMainRepository;
  static repository = new DbMainRepository();

  static async getAllCourse() {
    return await this.repository.findAllCourse();
  }
}
```



## 04 프로젝트 수행 결과

### ▶ GPSAPI 기술개발 (Rest API, Json)

3) 클라이언트 요청을 받아 서버에서 응답을 보냄

```
import MainService from '../service/MainService';

export const getHome = (req, res) => res.render('main/home');
export const getUsed = (req, res) => res.render('main/used');

// controller : 필요요청 보낸다
export const getCourse = async (req, res) => {
  const course = await MainService.getAllCourse();
  res.render('main/course', {course});
}
export const getQr = (req, res) => res.render('main/qr');

export const postQr = (req, res) => {
  const code = req.body.code;
  //code 200
  if (code) return res.status(200).json({ code: 'req_request', message: '요청이 성공함' });
  try {
    //요청을 보낸다
    // 200 성공 400 실패 (요청 실패)
    // 500 서버 오류
    if (code === 'error') throw { status: 400, code: 'not found' };
    return res.status(200).json({ code: 'OK', message: '성공 완료' });
  } catch (e) {
    if (e.code) return res.status(e.status).json({ code: e.code, message: e.message });
    else return res.status(500).json({ code: 'server_error', message: '서버 오류' });
  }
}
```

4) 페이지들을 가져와 그 페이지에 해당하는 내용들을 불러옴

```
import express from 'express';
import { getHome, getMap, getMapData, getQr, getUsed, postQr } from '../controller/mainController';

const mainRouter = express.Router();

mainRouter.get('/', getHome)
mainRouter.get('/used', getUsed)
mainRouter.get('/map', getMap)
mainRouter.get('/map/info', getMapData)

mainRouter.get('/qr', getQr)

mainRouter.post('/qr', postQr)

export default mainRouter;
```

MAP



인테리어



문구사무



패션잡화



식료품

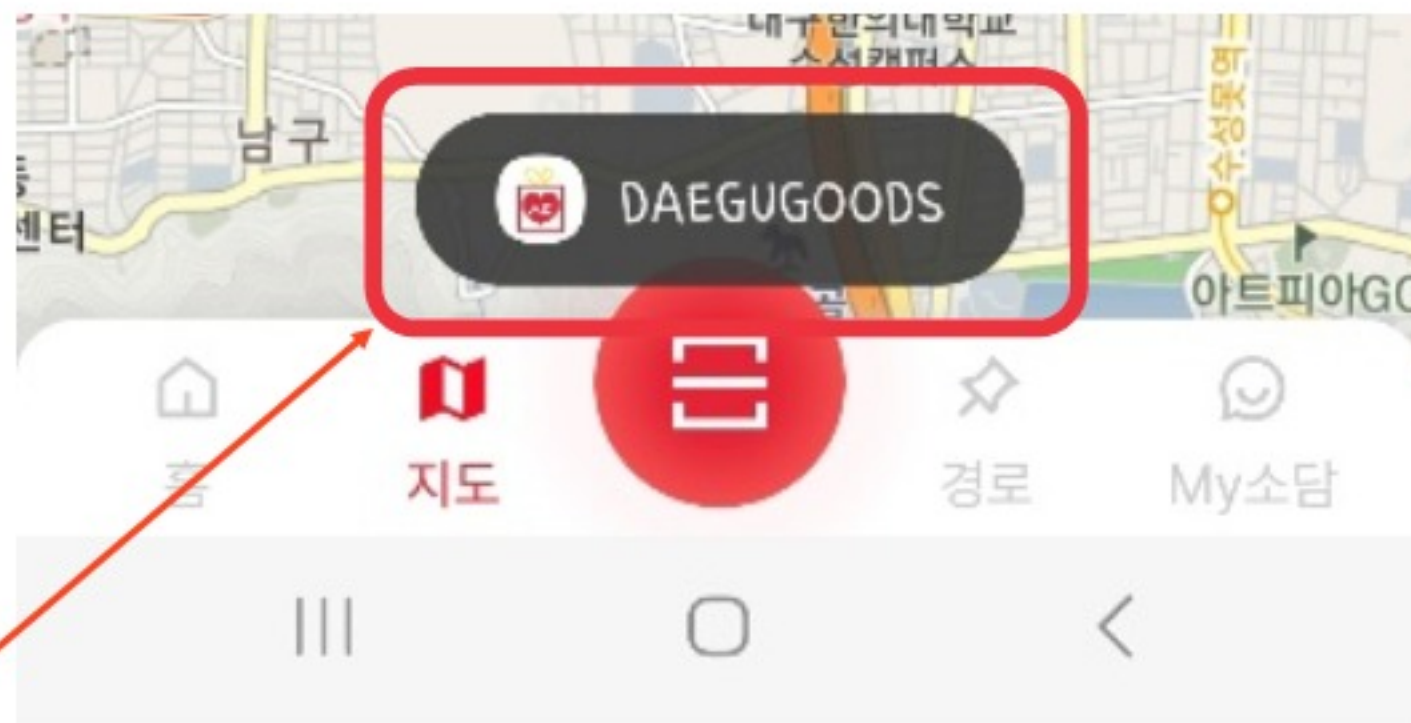
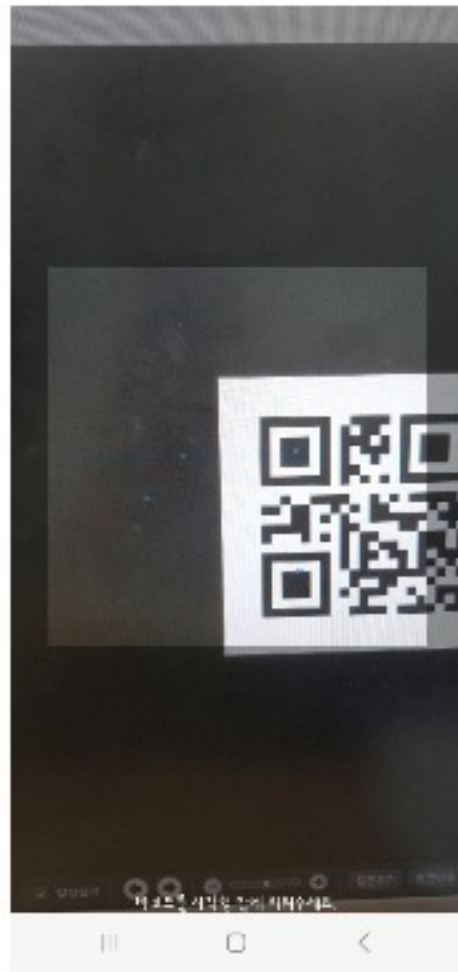




## 04 프로젝트 수행 결과

### ▶ GPSAPI 기술개발 (Rest API, Json)

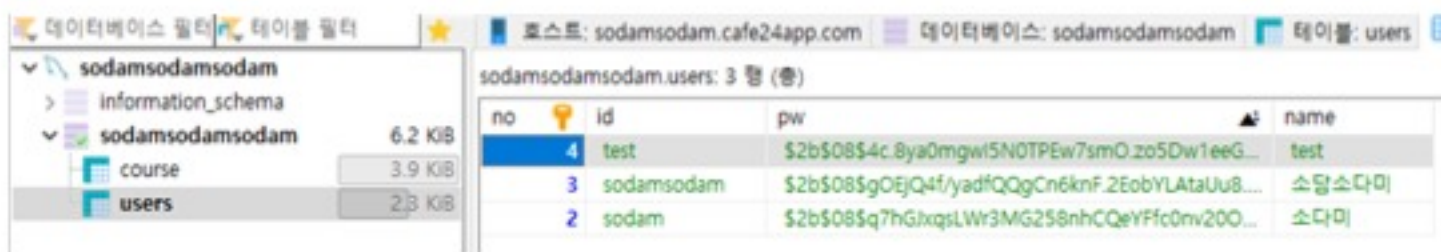
- 궤적 및 위치확인 후 인증 구현



# 04 프로젝트 수행 결과

## ▶ 데이터베이스 설계 및 구현(Node.js, mariaDB)

### 1) 마리아DB에 데이터베이스 생성



no	id	pw	name
4	test	\$2b\$08\$4c.8ya0mgw15N0TPEw7smO.zo5Dw1eeG...	test
3	sodamsodam	\$2b\$08\$gOEjQ4f/yadfQQgCn6knF.2EobYLAtaUu8...	소담소다미
2	sodam	\$2b\$08\$q7hGJxqslWr3MG258nhCQeYFfc0nv200...	소다미

### 2) 카페24를 마리아 db와 연결



### 3) 데이터를 파일로 불러옴

```
"use strict";

import mysql from "mysql2";

const config = {
  host: process.env.DB_HOST,
  port: process.env.DB_PORT,
  user: process.env.DB_USER,
  database: process.env.DB_DATABASE,
  password: process.env.DB_PASSWORD
};

const db = mysql.createPool(config).promise();

export default db;
```

### 4) 가져온 데이터를 다른 파일에서 쓸 수 있도록 객체를 내보냄

```
import db from "../config/db";

export class DbUsersRepository {
  async findById(id) {
    const QUERY = `SELECT no, id, pw, name FROM users WHERE id=?`;
    return db.execute(QUERY, [id]).then((result) => result[0][0]);
  }

  async save({id, pw, name}) {
    const QUERY = `INSERT INTO users (id, pw, name) VALUES (?, ?, ?)`;
    return db.execute(QUERY, [id, pw, name]);
  }
}
```



## 04 프로젝트 수행 결과

### ▶ 데이터베이스 설계 및 구현(Node.js, mariaDB)

- 로그인 및 회원가입 기능 구현

5) 클라이언트 요청을 받아 서버에서 응답을 보냄

```
import { Error } from "../error/error";
import bcrypt from "bcrypt";
import jwt from "jsonwebtoken";
import { DBUsersRepository } from "../models/DBUsersRepository.js";

export class loginService {
  static repository = new DBUsersRepository();
  static jwtSecretKey = process.env.SECRET_KEY;
  static jwtExpire = process.env.JWT_EXPIRE;

  static _createJwtToken(id) {
    return jwt.sign({ id }, this.jwtSecretKey, { expiresIn: this.jwtExpire });
  }

  static async login(id, pw) {
    const user = await this.repository.findById(id);
    if(!user) throw Error.UN_AUTHORIZED;
    const isValidPassword = await bcrypt.compare(pw, user.pw);

    console.log(isValidPassword);
    if(!isValidPassword) throw Error.UN_AUTHORIZED;
    const token = this._createJwtToken(user.id);
    return token;
  }

  static async join({ id, pw, name }) {
    const user = await this.repository.findById(id);
    if(user) throw Error.ID_EXIST;
    const hashedPw = await bcrypt.hash(pw, 8);
    await this.repository.save({id, pw: hashedPw, name});
  }
}
```

6) 페이지들을 가져와 그 페이지에 해당하는 내용들을 불러옴

```
"use strict";
import express from "express";
import { getJoin, getLogin, postJoin, postLogin, postLogout } from "../controller/loginController.js";
import { logoutCheck } from "../middleware/loginCheck.js";
import { joinForm, loginForm } from "../middleware/validation.js";

const userRouter = express.Router();

userRouter.route("/join").get(getJoin).post(joinForm, postJoin);
userRouter.route("/login").get(loginForm, postLogin);
userRouter.route("/logout").post(postLogout);

export default userRouter;
```

7) 로그인, 회원가입 화면 구현 결과

The image shows two mobile application screens for a service called "So:동물 Dam다" (So: Animal Dam-da). Both screens feature a logo at the top consisting of a red heart with a yellow apple inside, and the text "So:동물 Dam다" below it. Underneath the logo is the tagline "- 대구광역시 소상공인 지원 -".

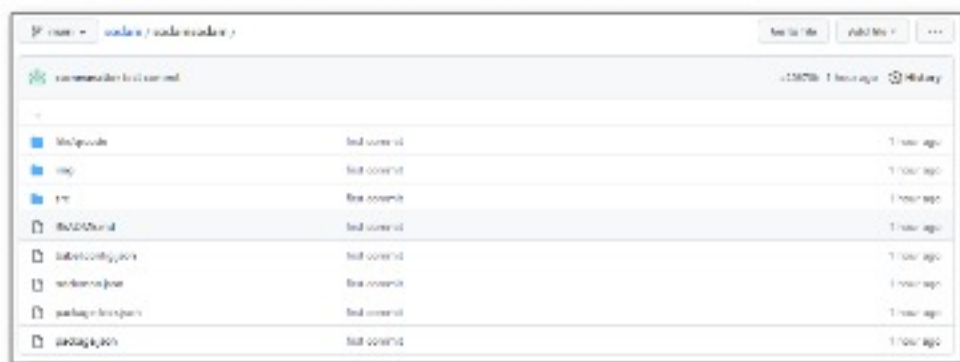
The left screen is the login page. It has two input fields: "이메일을 입력해주세요" (Please enter your email) and "비밀번호를 입력해주세요" (Please enter your password). Below these is a red button labeled "로그인" (Login). At the bottom, there is a link that says "소담이 처음이신가요? 회원가입하기" (Are you a first-time user? Sign up). The bottom navigation bar shows a hamburger menu icon, a home icon, and a back arrow.

The right screen is the sign-up page. It has three input fields: "이메일을 입력해주세요" (Please enter your email), "비밀번호를 입력해주세요" (Please enter your password), and "이름을 입력해주세요" (Please enter your name). Below these is a red button labeled "회원가입" (Sign up). The bottom navigation bar shows a hamburger menu icon, a home icon, and a back arrow.

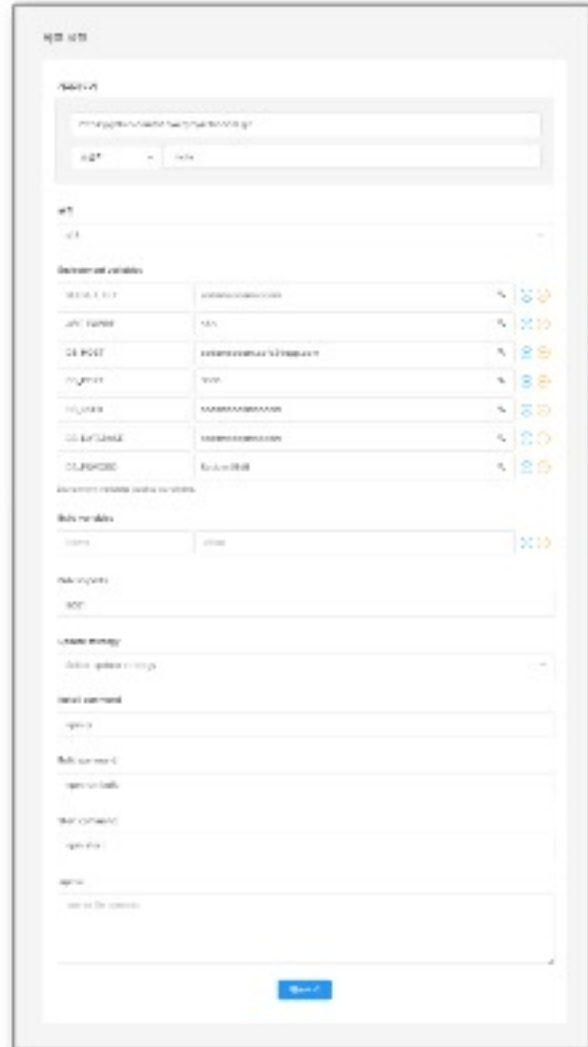
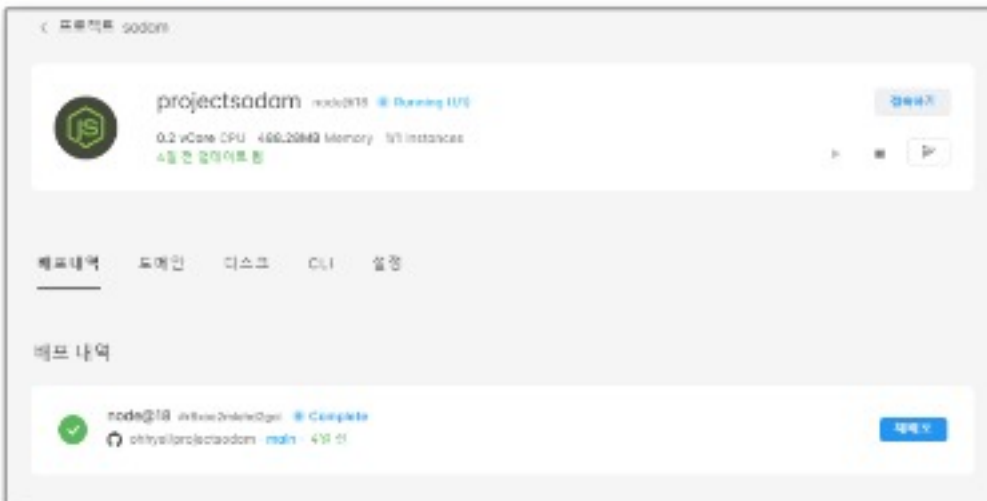
## 04 프로젝트 수행 결과

▶ **형상관리 도구 활용 및 앱 배포( GIT, Cloudtype, Android Studio )**

## GIT >파일공유,호스팅 연결



## Cloudtype > 호스팅





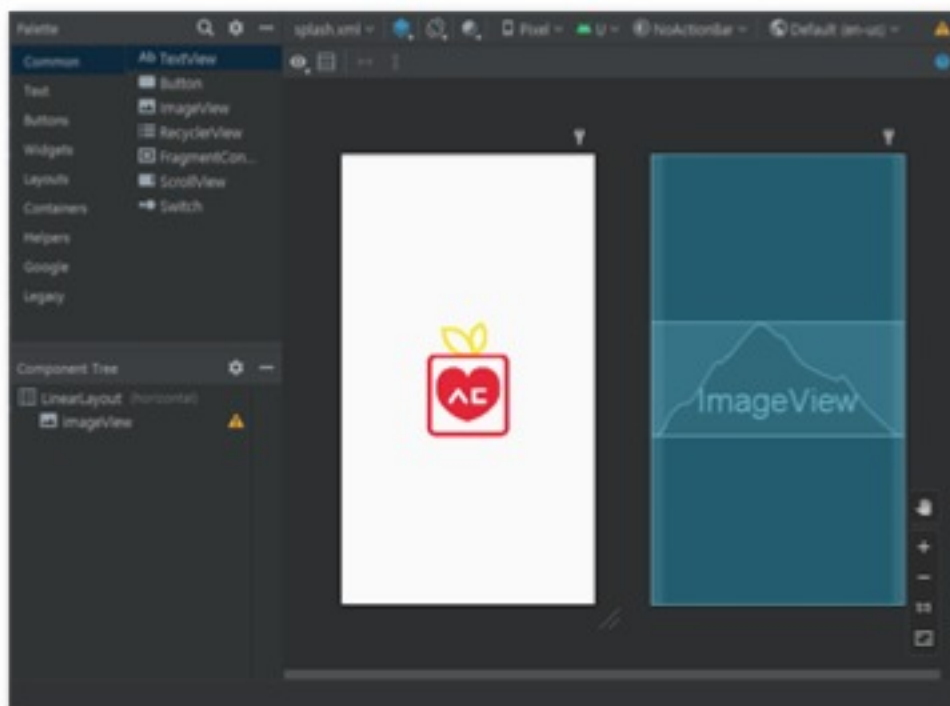
## 04 프로젝트 수행 결과

### ▶ 형상관리 도구 활용 및 앱 배포( GIT, Cloudtype, Android Studio )

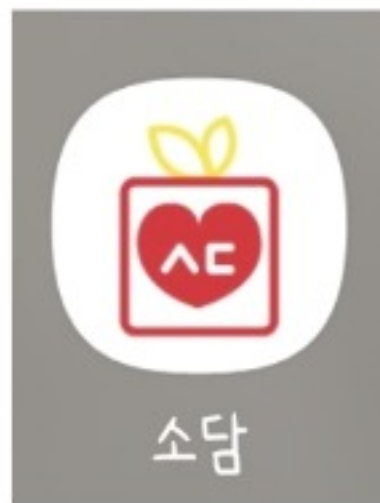
Android Studio > 앱 만들기

```
public static final String INDEX_PAGE =  
" https://port-0-sodamsodam-4uvq2mlegiamip.sel3.cloudtype.app/";
```

Android Studio : 스플래쉬 생성

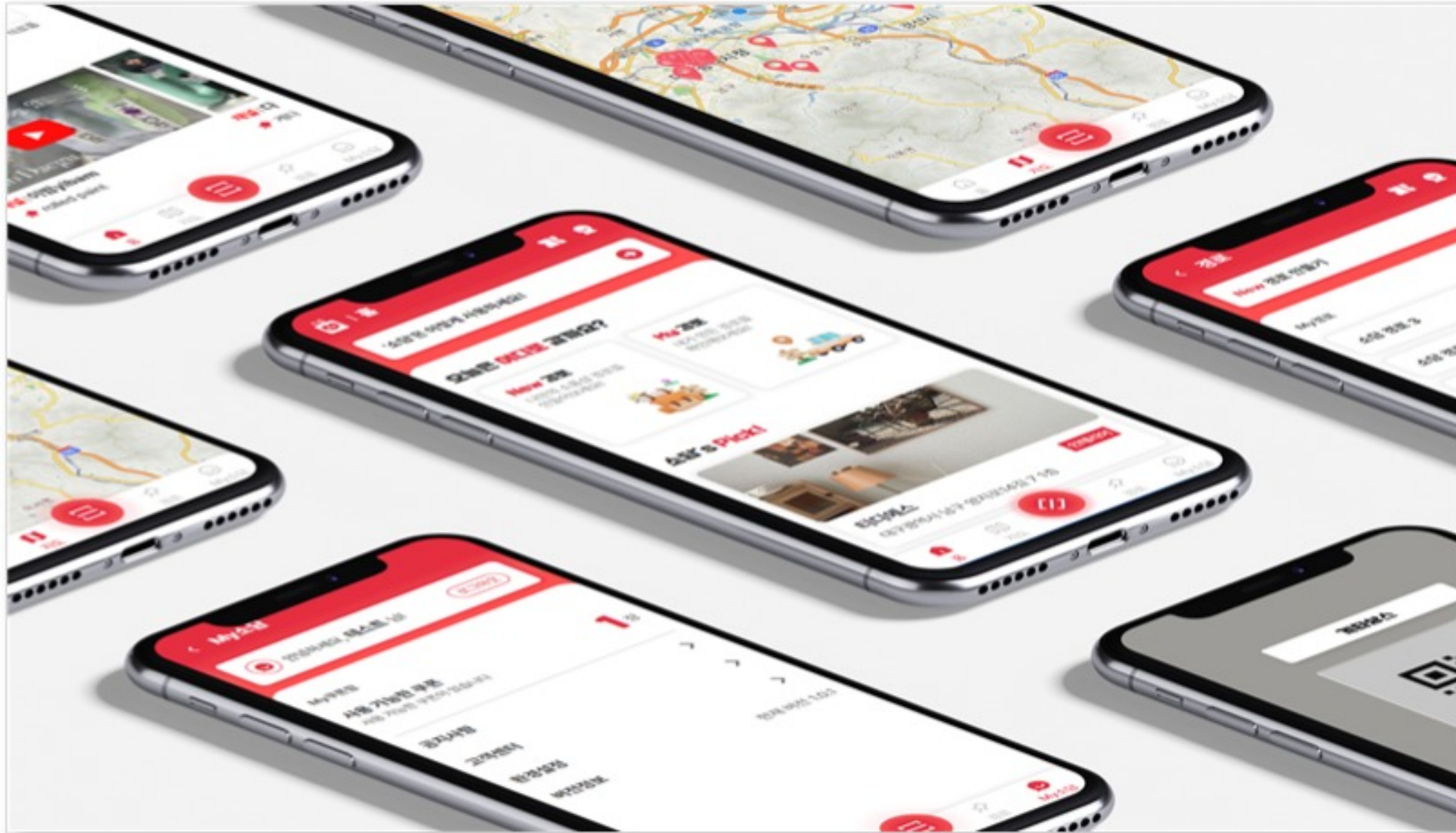


APK 파일 설치



## 04 프로젝트 수행 결과

- ▶ 시현 구현 및 영상(<https://port-0-projectsodam-4uvg2mlegiamip.sel3.cloudtype.app>)  
- Test ID / PW : user1 / 1234





# 자체평가의견



## 05 자체 평가 의견

### ▶ [자체 평가 의견]

정민혜

이번 프로젝트는 가장 짧은 기간 동안 진행한, 가장 어려운 프로젝트였습니다. 특히 DB에서 오류가 많이 발생하였는데 Node.js로 개발하여 클라우드 타입으로 배포하려고 하니 MySQL로 하는 것에는 한계가 있어 마리아 db를 사용하게 되었고 마리아 DB를 연동하기 위해 카페24를 사용하게 되었습니다. 아직도 중간중간 부족한 부분이 있지만 제가 핵심적으로 맡았던 로그인, 회원가입 부분의 DB가 제대로 작동하여 다행이라고 생각합니다.

디자인도 마찬가지로, 네이티브 앱 디자인을 제작한 경험이 있는데 하이브리드 앱의 경우 어쨌든 웹이기 때문에 네이티브 앱처럼 정해진 규격이 있지 않아 신경을 써야 하는 부분이 많았고 API의 경우에도 네이티브 앱에서는 사용 가능한 API 기능이 웹이라 불가능한 경우가 있었기 때문에 그런 점들을 비교하며 많이 배웠다고 생각합니다.

퍼블리셔  
백엔드



## 05 자체 평가 의견

### ▶ [자체 평가 의견]

윤진실	독특한 주제의 어플리케이션 디자인과 로고를 제작할 수 있는 프로젝트였습니다. 테마와 컬러가 확실한 프로젝트여서 디자인적으로 작업하는 데 큰 문제는 없었지만, 기획 과정과 db 연동과 퍼블리싱을 담당해 준 팀원들이 굉장히 많은 힘을 들인 걸 알아 고생하셨다는 말씀 드리고 싶습니다. 짧은 시간이었지만 소담이라는 앱 개발을 위해 고생한 팀원들에게 감사하다는 말씀 드리고 싶습니다.	UXUI 디자이너 영상
김현정	node.js 라는 처음 사용해보는 개발툴로 짧은 기간에 개발하느라 어렵기도 했고 아직 기획한 만큼 완성되진 않았지만 개인적으로 공부도 많이 됐고 GPS API를 활용한 QR 코드 인증이라는 새로운 기술을 익힐 수 있어서 좋았던 프로젝트였습니다.	퍼블리셔
권지현	프로젝트의 기술들을 잘 파악하지 못한 채 기획을 한 점이 아쉬웠습니다. 시간이 많지 않아 QR코드 디자인을 제대로 퍼블리싱 하지 못한 점도 아쉬웠습니다. 하지만 제가 관심있는 주제로 프로젝트를 기획하고 진행하는 일이 즐거웠습니다. 함께 해준 팀원들에게 감사의 인사를 하고 싶습니다.	기획, 발표



**감사합니다**