

조용현 포트폴리오

목차

1. 활용 가능한 언어 소개
2. C언어 2차원 배열을 활용한 주차 관리 프로그램
3. C# Winform을 활용한 주차 관리 프로그램 v2.0
4. Python 데이터 분석 “내 님 찾기” 프로그램
5. K-Digital Hackathon 본선 진출 “가는 날 눈치 게임” 프로젝트 기획
6. JSP와 MySQL을 활용한 K-Digital Hackathon ‘WarmPlace’ 프로젝트 제작

1. 커리큘럼 및 활용 가능한 언어

2021년 3월

-프로그래밍 입문

-C를 활용하여 프로그래밍과 알고리즘에 대한 기본적인 개념 학습.

-C#언어 학습과 비주얼 스튜디오 툴을 활용한 프로그램 제작

2021년 4월

-Python을 활용한 CSV 데이터 분석 및 기초 언어 학습

2021년 5월

-Oracle, JAVA 데이터베이스의 기본 개념 및 이클립스 툴을 활용한 JAVA언어 학습과 데이터베이스 연동, 객체지향 언어의 개념 학습

2021년 6월

-K-Digital 해커톤 참가 '가는 날 눈치 게임' 프로젝트

2021년 7월

-HTML, JSP, JAVA를 활용한 웹 개발

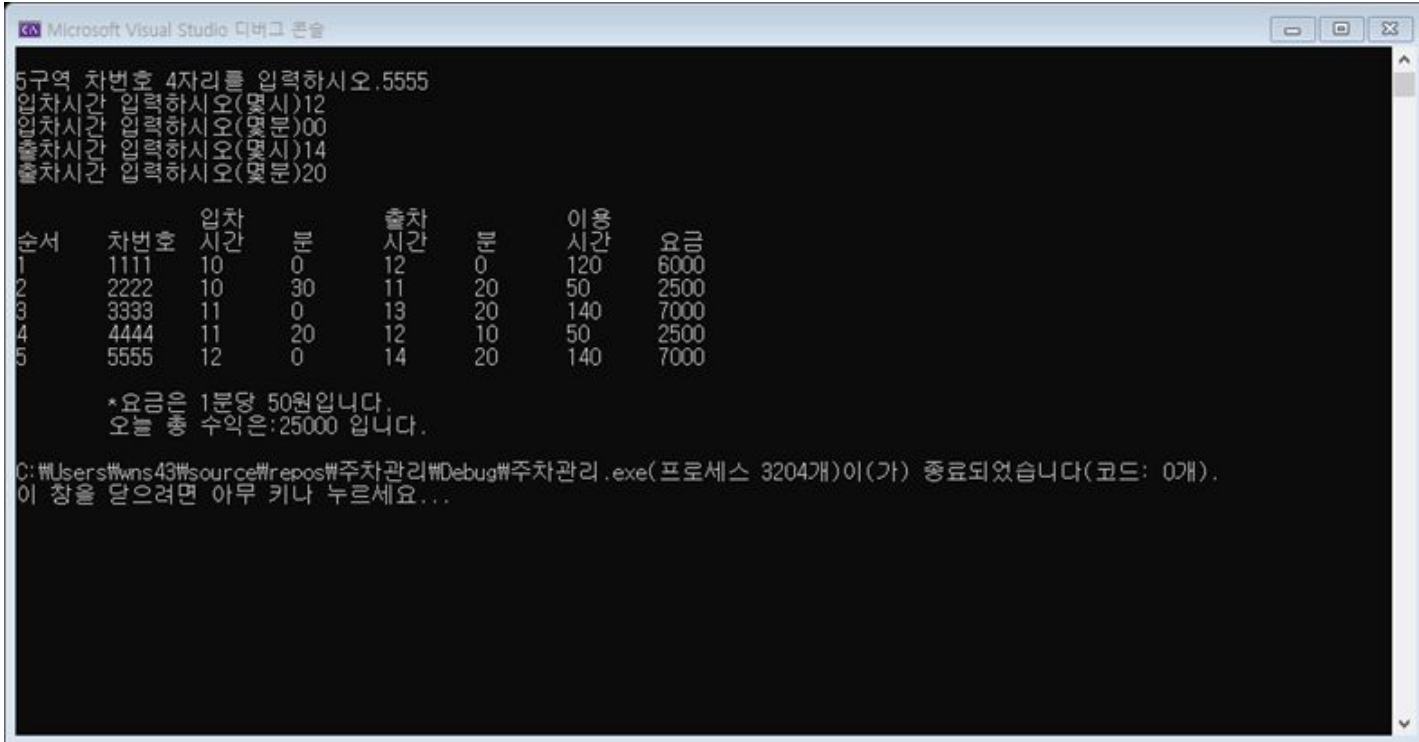
2. C언어 2차원 배열을 활용한 주차 관리 프로그램

소개

- 5대의 주차공간을 가정하고 2차원 배열에 차번호, 입차시간, 출차시간을 입력하여 자동으로 주차시간과 총 비용을 계산해 주는 프로그램 제작
- 기능을 함수화하여 메인 함수에서 불러오는 방식의 절차지향 프로그램의 코딩방식에 따라 제작

github URL : <https://github.com/DrillND/Portfolio>

2-1 주차 관리 프로그램 실행 화면



```
Microsoft Visual Studio 디버그 콘솔

5구역 차번호 4자리를 입력하시오 .5555
입차시간 입력하시오(몇시)12
입차시간 입력하시오(몇분)00
출차시간 입력하시오(몇시)14
출차시간 입력하시오(몇분)20

순서   차번호   입차   출차   이용   요금
      번호   시간   시간   시간
1      1111    10     12     0     120    6000
2      2222    10    30    11     20     50    2500
3      3333    11     0     13     20    140    7000
4      4444    11    20    12     10     50    2500
5      5555    12     0     14     20    140    7000

*요금은 1분당 50원입니다.
오늘 총 수익은 :25000 입니다.

C:\Users\hwns43\source\repos\주차관리\Debug\주차관리.exe(프로세스 3204개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

2-2 주차관리 프로그램 주요 코드

```
#include <stdio.h> //주차관리시스템

int i, j;
int result;
int arr[5][10];

int main(void)
{
    printf("<일일주차관리기록부>\n\n");
    input();
    menu();
    order();
    output();
    total();

    printf("\n");
    printf("Wt*요금은 1분당 50원입니다.\n");
    printf("Wt오늘 총 수익은:%d 입니다.\n", result);

    error();

    return 0;
}
```

메인 함수에 각각의
기능들을 함수화하여
유지보수가 쉽게 구성

2-2 주차관리 프로그램 주요 코드

```
int input()    //입력값
{
    for (i = 0; i < 5; i++)
    {
        carnum();
        startnum();
        startnum2();
        endnum();
        endnum2();

        printf("Wn");
    }
}
```

입력값이라는 기능 안에
함수화를 통해 기능의 추가와
수정이 용이하게 제작

2-2 주차관리 프로그램 주요 코드

```
int startnum()
{
    while (1)
    {
        printf("입차시간 입력하시오(몇시)");
        scanf_s("%d", &arr[i][2]);
        if (arr[i][2] > 24 || arr[i][2] < 0)
            printf("잘못입력하셨습니다\n");
        if (arr[i][2] <= 24)
            break;
    }
}
```

함수 중 입차 시간 입력 함수

- if함수를 활용하여 알맞은 조건의 시간을 배열에 입력

3. C# Winform을 활용한 주차 관리 프로그램 v2.0

소개

- 실제 유료주차장에서 사용 할 프로그램을 가정하고 입차 시 차번호와 입차시간, 여러가지 조건을 입력하고 출차 시에 조건들을 입력하면 자동으로 주차시간과 총 비용을 계산해 주는 프로그램 제작

github URL : <https://github.com/DrillND/Portfolio>

3-1 주차 관리 프로그램 v2.0 실행화면

Form1

주차 관리

자 번호 입력

1117

입차 시간

오후

4

23

자 종

슈퍼카(분당1만원)

☐ 할인 쿠폰

등록

현재 차량 현황

삭제

자 번호

1117

출차 시간

오후

10

10

출차

☐ 현재 시간으로 계산(미완성)

삭제

| 자 번호 | 자 종 | 입차 시간 | 할인 쿠폰 여부 |
|--------|------------|-----------|----------|
| ▶ 1111 | 일반(분당500원) | 오전 1 : 2 | 할인 없음 |
| 1112 | 일반(분당500원) | 오전 3 : 12 | 10% 할인 |
| 1113 | 일반(분당500원) | 오전 4 : 22 | 할인 없음 |
| 1114 | 일반(분당500원) | 오후 1 : 12 | 할인 없음 |
| 1115 | 슈퍼카(분당1만원) | 오후 1 : 22 | 할인 없음 |
| 1116 | 경차(분당300원) | 오후 2 : 23 | 10% 할인 |

| 자 번호 | 자 종 | 입차 시간 | 출차 시간 | 경과 시간 | 할인 쿠폰 여부 | 총 비용 |
|--------|-------------|-----------|------------|-------|----------|----------|
| ▶ 1117 | 슈퍼카(분당1만... | 오후 4 : 23 | 오후 10 : 10 | 347 분 | 할인 없음 | 3470000원 |

정산하기

3-2 주차 관리 프로그램 v2.0 주요코드

```
private void button1_Click(object sender, EventArgs e)
{
    //등록버튼
    if (textBox1.Text != "" && textBox4.Text != "시" && textBox5.Text != "분" && comboBox1.Text != "")
    {
        //차량 번호가 중복되지 않으면 1 중복되면 2
        int flag = 1;
        foreach (var sam in dataGridView1.Rows) //foreach 문과 DataGridView 복습
        {
            DataGridViewRow dataGridViewRow = sam as DataGridViewRow;
            if (dataGridViewRow.Cells[0].Value.ToString() == textBox1.Text)
            {
                flag = 2;
                MessageBox.Show("중복되는 차량 번호 입니다.");
                break;
            }
        }
        if (flag == 1)
        {
            if (checkBox1.Checked)
            {
                {
                    dataGridView1.Rows.Add(textBox1.Text, comboBox1.Text, comboBox2.Text + " " + textBox4.Text + " : " + textBox5.Text, "10% 할인");
                }
            }
            else
            {
                dataGridView1.Rows.Add(textBox1.Text, comboBox1.Text, comboBox2.Text + " " + textBox4.Text + " : " + textBox5.Text, "할인 없음");
            }
        }
    }
}
```

등록버튼을
클릭시 텍스트
박스에 입력된
값(차량번호,
입차시간 등)
들이 데이터
그리드 뷰에
조건에 맞게
입력된다.

3-2 주차 관리 프로그램 v2.0 주요코드

```

if (textBox6.Text != "" && textBox8.Text != "시" && textBox7.Text != "분")
{
    if (checkBox2.Checked)
    {
        foreach (var item in dataGridView1.Rows)
        {
            DataGridViewRow dataGridViewRow = item as DataGridViewRow;

            if (dataGridViewRow.Cells[0].Value.ToString() == textBox6.Text)
            {
                dataGridView1.Rows.Remove(dataGridViewRow);
                dataGridView3.Rows.Add(dataGridViewRow.Cells[0].Value, dataGridViewRow.Cells[1].Value, dataGridViewRow.Cells[2].Value,
                    outtimeamp + " " + outtime1 + ":" + outtime2, dataGridViewRow.Cells[3].Value);
            }
        }
    }
}
else
{
    var a = int.Parse(textBox4.Text);
    var b = int.Parse(textBox8.Text);
    if (comboBox2.Text == "오후")
    {
        a = (a + 12) * 60;
    }
    else
    {
        a = a * 60;
    }
    if (comboBox3.Text == "오후")
    {

```

이어지는 등록버튼 코드

조건문을 활용하여
조건을 모두 열거하여

데이터그리드 뷰에 입력

(개발 초창기라 변수와 버튼이름이 제멋대로이다.)

3-2 주차 관리 프로그램 v2.0 주요코드

```
int x = int.Parse(textBox5.Text);
int y = int.Parse(textBox7.Text);
int outcome = (b + y) - (a + x);
if (outcome < 0)
{
    MessageBox.Show("출차 시간을 확인하세요!");
}
else
{
    foreach (var item in dataGridView1.Rows)
    {
        DataGridViewRow dataGridViewRow = item as DataGridViewRow;

        if (dataGridViewRow.Cells[0].Value.ToString() == textBox6.Text)
        {
            {
                if (dataGridViewRow.Cells[1].Value.ToString() == "경차(분당300원)" && dataGridViewRow.Cells[3].Value.ToString() == "10% 할인")
                {
                    dataGridView1.Rows.Remove(dataGridViewRow);
                    dataGridView3.Rows.Add(dataGridViewRow.Cells[0].Value, dataGridViewRow.Cells[1].Value, dataGridViewRow.Cells[2].Value, comboBox3.Text);
                }
                if (dataGridViewRow.Cells[1].Value.ToString() == "경차(분당300원)" && dataGridViewRow.Cells[3].Value.ToString() == "할인 없음")
                {
                    dataGridView1.Rows.Remove(dataGridViewRow);
                    dataGridView3.Rows.Add(dataGridViewRow.Cells[0].Value, dataGridViewRow.Cells[1].Value, dataGridViewRow.Cells[2].Value, comboBox3.Text);
                }
            }
            {
                if (dataGridViewRow.Cells[1].Value.ToString() == "일반(분당500원)" && dataGridViewRow.Cells[3].Value.ToString() == "10% 할인")
                {
                    dataGridView1.Rows.Remove(dataGridViewRow);
                    dataGridView3.Rows.Add(dataGridViewRow.Cells[0].Value, dataGridViewRow.Cells[1].Value, dataGridViewRow.Cells[2].Value, comboBox3.Text);
                }
            }
        }
    }
}
```

출차 버튼을 클릭
시 입차 버튼을
통해 입력되었던
첫번째 그리드뷰에
데이터는 삭제되고
새로운 그리드뷰로
출차되는 자료로
등록

4. Python ‘내 님 찾기’ 프로그램

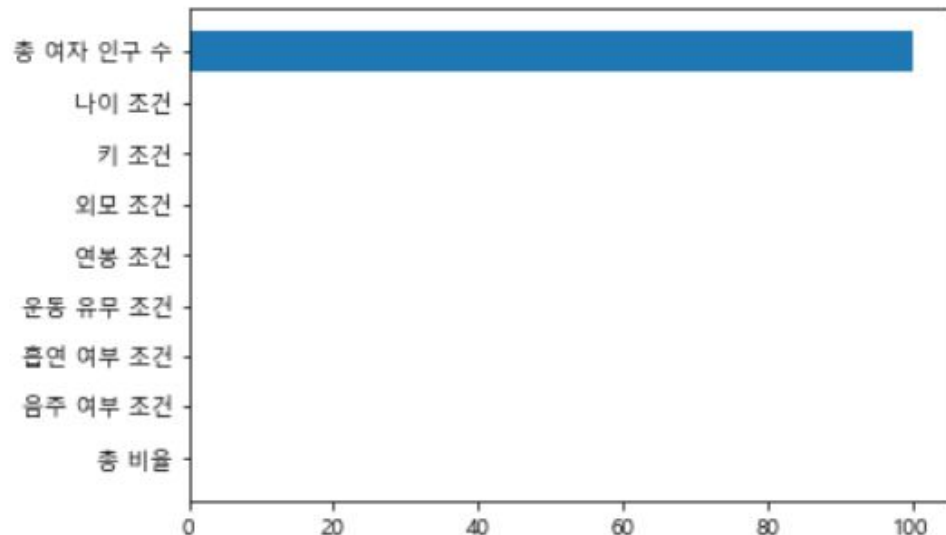
소개

- 병무청, 통계청 등 신뢰할 만한 출처에서 가져온 **CSV**파일을 활용하고 대구지역의 연령별 미혼자 수, 키, 연봉, 흡연 및 음주 여부 등을 조합하여 사용자가 원하는 조건을 입력하면 그에 맞는 조건의 사람을 만날 수 있는 확률을 계산해 주는 프로그램

github URL : <https://github.com/DrillND/Portfolio>

4-1 '내 님 찾기' 프로그램 실행 화면

당신의 성별은 무엇입니까? 1.남자 2.여자 :1



희망하는 배우자 여성의 나이 대를 선택하세요.

몇 살 부터(20살 이상) :

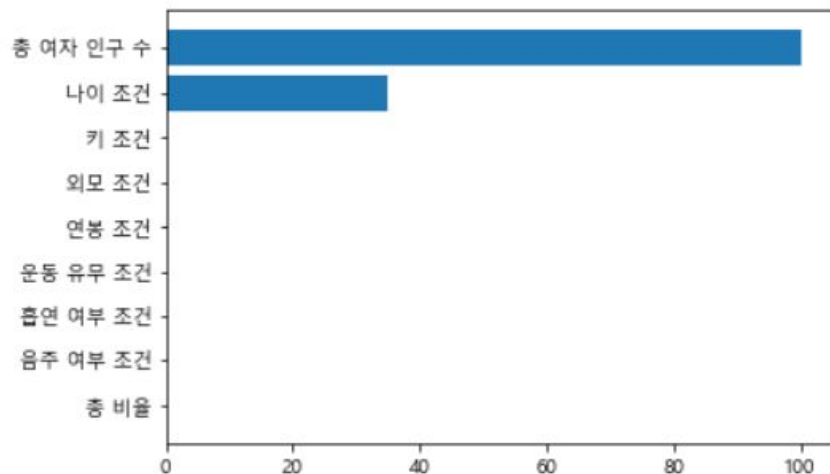
4-1 '내 님 찾기' 프로그램 실행 화면

희망하는 배우자 여성의 나이 대를 선택하세요.

몇 살 부터(20살 이상) : 25

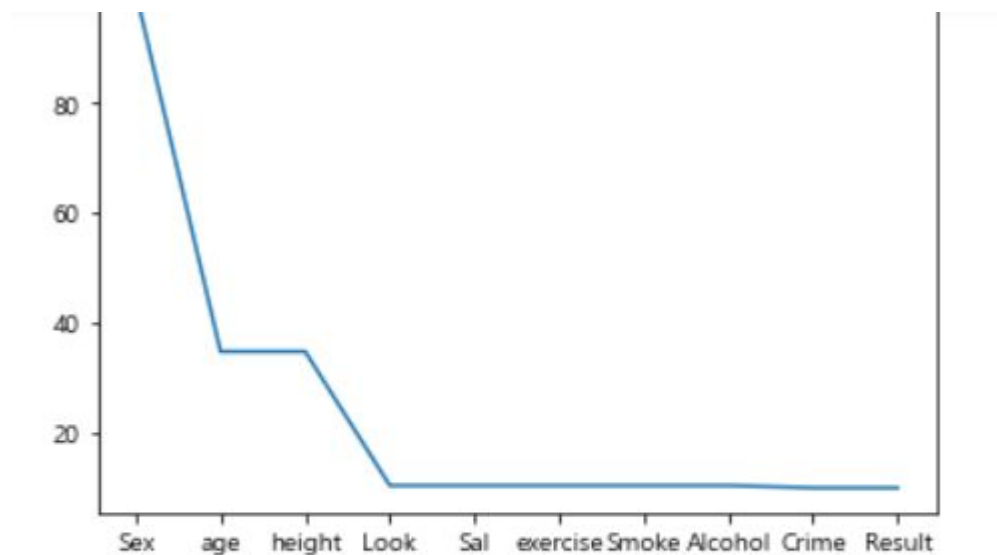
몇 살 까지 : 35

선택한 나이의 미혼자 비율 34.79373510189765 %



희망하는 배우자 여성의 키를 입력하세요.(상관없다면 0 입력) :

4-1 '내 님 찾기' 프로그램 실행 화면 최종 결과



대구광역시 총 여자 수 : 1,213,338

당신이 원하는 조건을 만족할 확률 10.020595709346523 %

당신이 원하는 조건을 만족하는 여성의 수 : 121583.7 명

4-2 '내 님 찾기' 프로그램 주요 코드

```
import csv #나0/대#1122
f=open('marriage.csv',encoding='cp949')
data=csv.reader(f)

for i in data:
    if '대구' in i[0]:
        if '15' in i[1]:
            a=i[8:9]
            w_a=i[13:14]
        if '20' in i[1]:
            b=i[8:9]
            w_b=i[13:14]
        if '25' in i[1]:
            c=i[8:9]
            w_c=i[13:14]
        if '30' in i[1]:
            d=i[8:9]
            w_d=i[13:14]
        if '35' in i[1]:
            e=i[8:9]
            w_e=i[13:14]
```

CSV 파일을 import하고
데이터들을 조건에 맞추어
배열, 딕셔너리, 튜플에
입력시킨다.

4-2 '내 님 찾기' 프로그램 주요 코드

```
def smoke(self):  
    s = int(input("배우자의 흡연 여부 1.흡연자였으면 좋겠다. 2.비흡연자였으면 좋겠다. 3.상관없음 : "))  
    if s == 1:  
        return list_dic4[0]  
        print(list_dic4[0])  
    if s == 2:  
        return list_dic4[1]  
        print(list_dic4[1])  
    if s == 3:  
        return 100  
        print(100)
```

필요한 조건들은 정렬된 데이터를 함수화 하여 제작.

4-2 ‘내 님 찾기’ 프로그램 주요 코드

```
%matplotlib inline  
  
import matplotlib.pyplot as plt  
import numpy as np
```

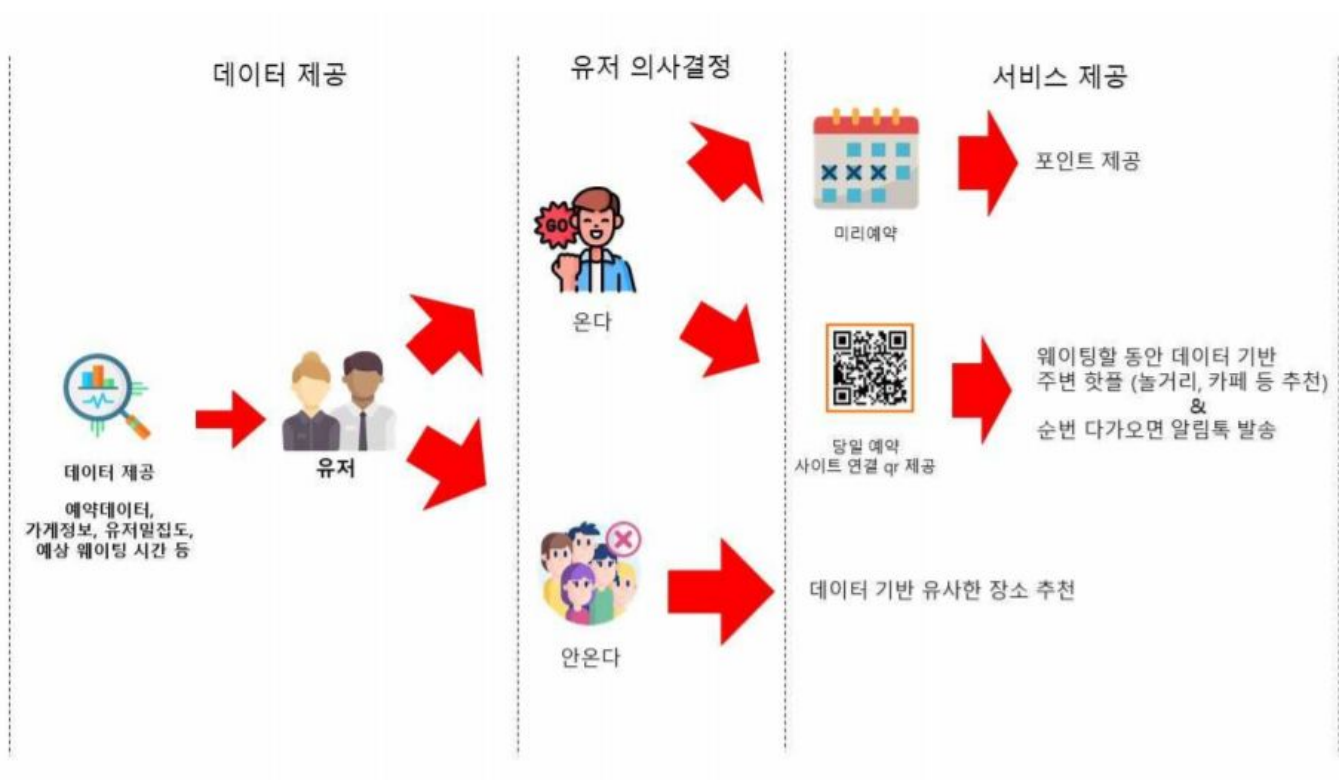
matplotlib과 numpy를
이용하여 그래프를
삽입. 데이터를
시각적으로 표현

5. K-Digital Hackathon 본선 ‘가는 날 눈치 게임’ 프로젝트

소개

- 한국판 뉴딜의 실현을 돕는 서비스나 앱 개발을 주제로 한 고용노동부 해커톤 대회 본선 진출 작품
- 판데믹 사태로 인해 인파가 몰리는 상황을 피해야 하는 보건 상황과 인파가 붐비는 곳을 피하고 싶은 관광객의 수요를 일치시켜 줄 수 있는 서비스.
- 사용자들의 예약 데이터를 활용하여 실시간 인파를 예측 제공하는 서비스
사용자는 서비스를 통해 사람들이 많은 장소를 피하고 비교적 안전하면서 유명한 장소를 찾아 갈 수 있다.

5-1 '가는 날 눈치 게임' 구성도



5-2 ‘가는 날 눈치게임’ 소프트웨어 아키텍처



5-4 웹 어플리케이션 서버 & 데이터 베이스 구현 계획

WAS

-가게 추천을 위한 로직

- 1) 데이터베이스에 저장된 가게테이블에서 가게좌표를 받아와서 사용자 거리의 일정범위 내 (예 100M이내)에 있는 가게를 가져온다.
- 2) 데이터베이스에 저장된 소비자가 원하는 키워드를 가진 가게 리스트를 가져온다.
- 3) JAVA 코드를 활용해 FOR반복문으로 받아온 객체를 비교하여 일정범위 내에 있는 가게 중 소비자가 원하는 키워드를 가진 가게들을 추려 웹 서버에 넘겨준다.



DB
AWS

| + STORE_T | | | | | | | |
|--------------|----|----|------|--------------|------------|-------------|-------------|
| 가게 테이블 | | | | | | | |
| PK | AI | FK | Null | Logical Name | Name | Type | + |
| ✓ | ✓ | + | | 가게ID | STORE_ID | INT | — |
| | | + | ✓ | 가게이름 | STORE_NAME | VARCHAR(45) | — |
| | | + | ✓ | 가게좌표 | STORE_GPS | VARCHAR(45) | — |
| | | + | ✓ | 가게대기번호 | STORE_WNUM | VARCHAR(45) | — |
| SQL / Menu ▾ | | | | | | | ⌕ (X) ⌂ ↶ ↷ |

| KEYWORD_ID_T | | | | | | |
|--------------|----|----|------|--------------|------------|------|
| 키워드 테이블 | | | | | | |
| PK | AI | FK | Null | Logical Name | Name | Type |
| ✓ | + | | | 키워드ID | KEYWORD_ID | INT |
| | | ✓ | | 가게ID | STORE_ID | INT |

SQL / Menu

| KEYWORD_INFO_T | | | | | | |
|--------------------|----|----|------|--------------|--------------|-------------|
| Table logical name | | | | | | |
| PK | AI | FK | Null | Logical Name | Name | Type |
| ✓ | + | | | 키워드정보 | KEYWORD_INFO | VARCHAR(45) |
| | | ✓ | | 키워드ID | KEYWORD_ID | INT |

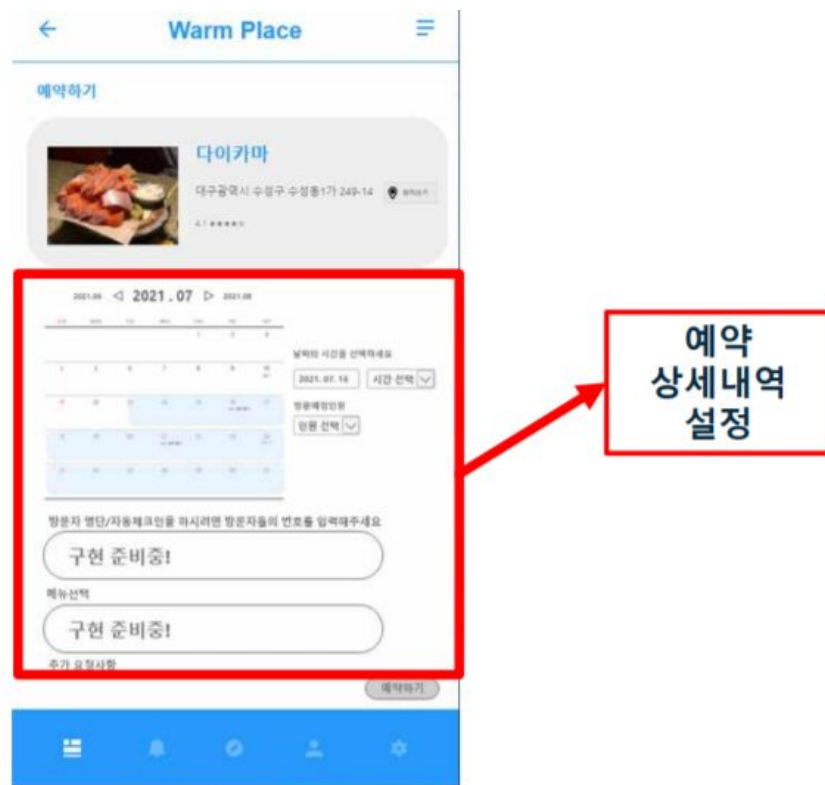
SQL / Menu

가게 좌표를 구글지도나 네이버지도 API를 통해 데이터 수집,
한 가게는 여러 개의 키워드를 참조

5-5 서비스 메인 화면 디자인



5-6 서비스 예약 기능 디자인



5-7 서비스 당일 예약 기능 디자인



전화번호
입력시
당일예약 진행

웨이팅 동안
이용가능한 주변
장소 추천

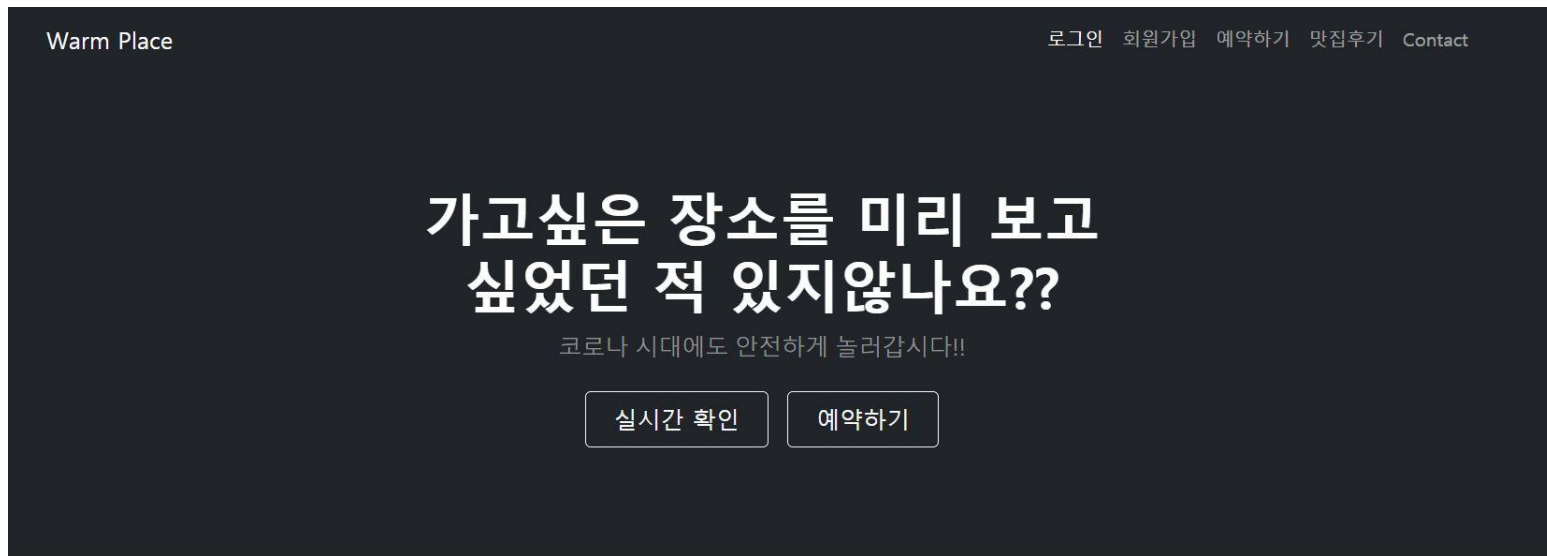
6. JSP와 MySQL을 활용한 Hackathon 'WarmPlace' 프로젝트 제작

소개

- K-Digital Hackathon 대회에서 기획했던 프로젝트를 JSP와 MySQL을 활용하여 웹 서비스 제작
- 사람들이 몰리는 장소를 피하면서 유명한 장소를 찾아가는 '가는 날 눈치게임'의 주제에 맞추어 'HOT Place'보다 조금 덜 사람들이 몰리는 'Warm Place'로 이름 변경

github URL : <https://github.com/DrillND/Portfolio>

6-1 WarmPlace 주요 페이지 화면



실시간 장소확인



Featured title



Featured title

6-2 WarmPlace 주요 페이지 화면

Warm Place

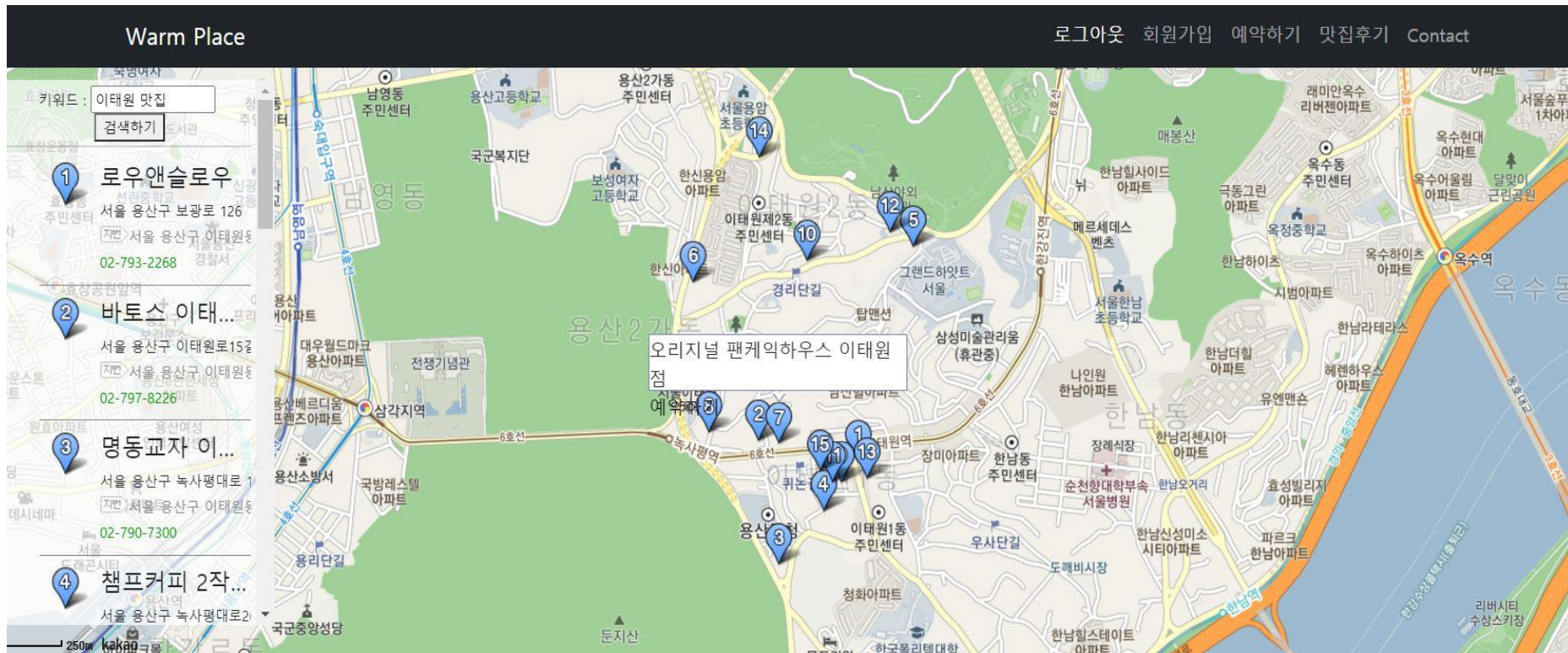
로그인 화면

아이디

비밀번호

로그인

6-3 WarmPlace 주요 페이지 화면



6-4 WarmPlace 주요 코드

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta charset="utf-8" />
6 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
7 <meta name="description" content="" />
8 <meta name="author" content="" />
9 <title>Warm Place_main</title>
10 <!-- Favicon-->
11 <link rel="icon" type="image/x-icon" href="assets/favicon.ico" />
12 <!-- Bootstrap icons-->
13 <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-icons.css" rel="stylesheet" />
14 <!-- Core theme CSS (includes Bootstrap)-->
15 <link href="css/styles.css" rel="stylesheet" />
16 </head>
17 <body>
18 <%
19 // 현재 세션 상태를 체크한다
20 String userID = null;
21 if (session.getAttribute("userID") != null) {
22     userID = (String) session.getAttribute("userID");
23 }
24 %>
25
```

부트스트랩

테마를 활용하여

웹페이지 구성

6-5 WarmPlace 주요 코드

```
public int login(String userID, String userPassword) {
    String sql = "select userPassword from userjoin where userID = ?";
    try {
        // 결과값 받아오기
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt = conn.prepareStatement(sql); // sql 쿼리문을 대기 시킨다
        pstmt.setString(1, userID); // 첫번째 '?'에 매개변수로 받은 'userID'를 대입
        rs = pstmt.executeQuery(); // 쿼리를 실행한 결과를 rs에 저장
        if (rs.next()) {
            if (rs.getString(1).equals(userPassword)) {
                return 1; // 로그인 성공
            } else {
                return 0; // 비밀번호 틀림
            }
        }
        return -1; // 아이디 없음
    } catch (Exception e) {
        e.printStackTrace();
    }
    return -2; // 오류
}
```

로그인 기능

DB의 Userjoin

테이블에 저장된

UserID컬럼의 항목과

입력된 UserID를

비교하여 같은 것의

비밀번호를 추출,

입력된 비밀번호와

비교하여 로그인

여부를 판단한다.

6-5 WarmPlace 주요 코드

```
// 글쓰기
public int write(String Title, String userID, String Content) {
    String sql = "insert into bbs values(?, ?, ?, ?, ?, ?)";
    try {
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, getNext());
        pstmt.setString(2, Title);
        pstmt.setString(3, userID);
        pstmt.setString(4, getDate());
        pstmt.setString(5, Content);
        pstmt.setInt(6, 1); // 글의 유효번호
        return pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return -1; // 데이터베이스 오류
}
```

맛집 후기의 글쓰기 기능

write.jsp에서 입력 받은
게시글의 제목, 사용자의
ID, 내용을
데이터베이스에 저장하는
DAO.JAVA 코드



YONG HYEON CHO

Tel) 010 - 7101 -8392

E-Mail) nnewvoice9@gmail.com

GitHub URL) <https://github.com/DrillND>