

# 맛집쓰- ver.대구

팀 - 앓아씨 [이현오, 장광수, 정인혁, 최효정]



# 목차

1

프로젝트 개요

2

팀 소개

3

구성원 소개

4

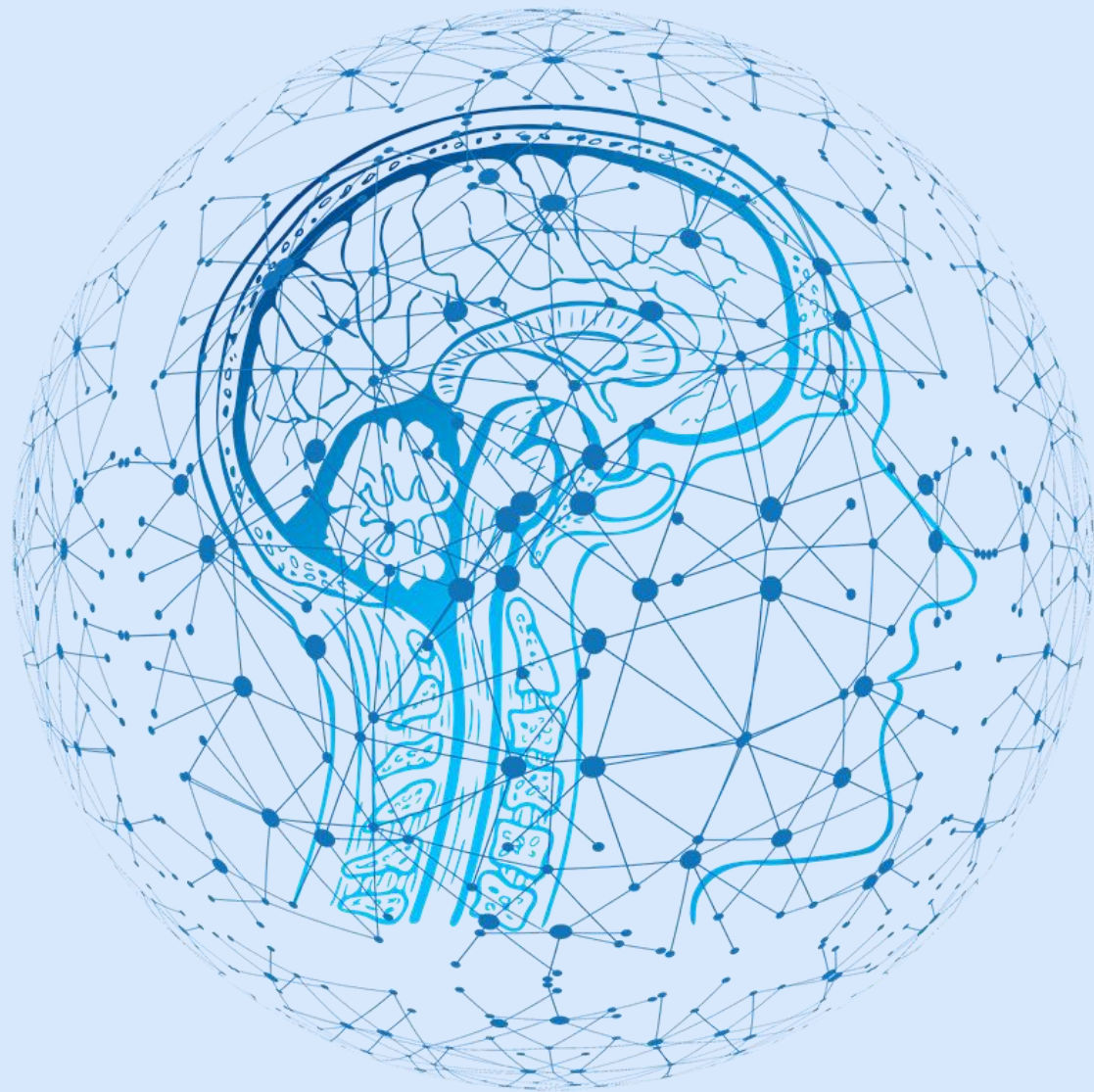
프로젝트 수행절차

5

프로젝트 수행결과

6

자체 평가의견



# 프로젝트 개요

---



# 프로젝트 개요

## 맛집쓰- ver.대구

대구광역시에서 맛집으로 보증하는 데이터만 추출하여  
사용자가 즐겨찾기한 맛집을 개인화하여 관리하기 위함.

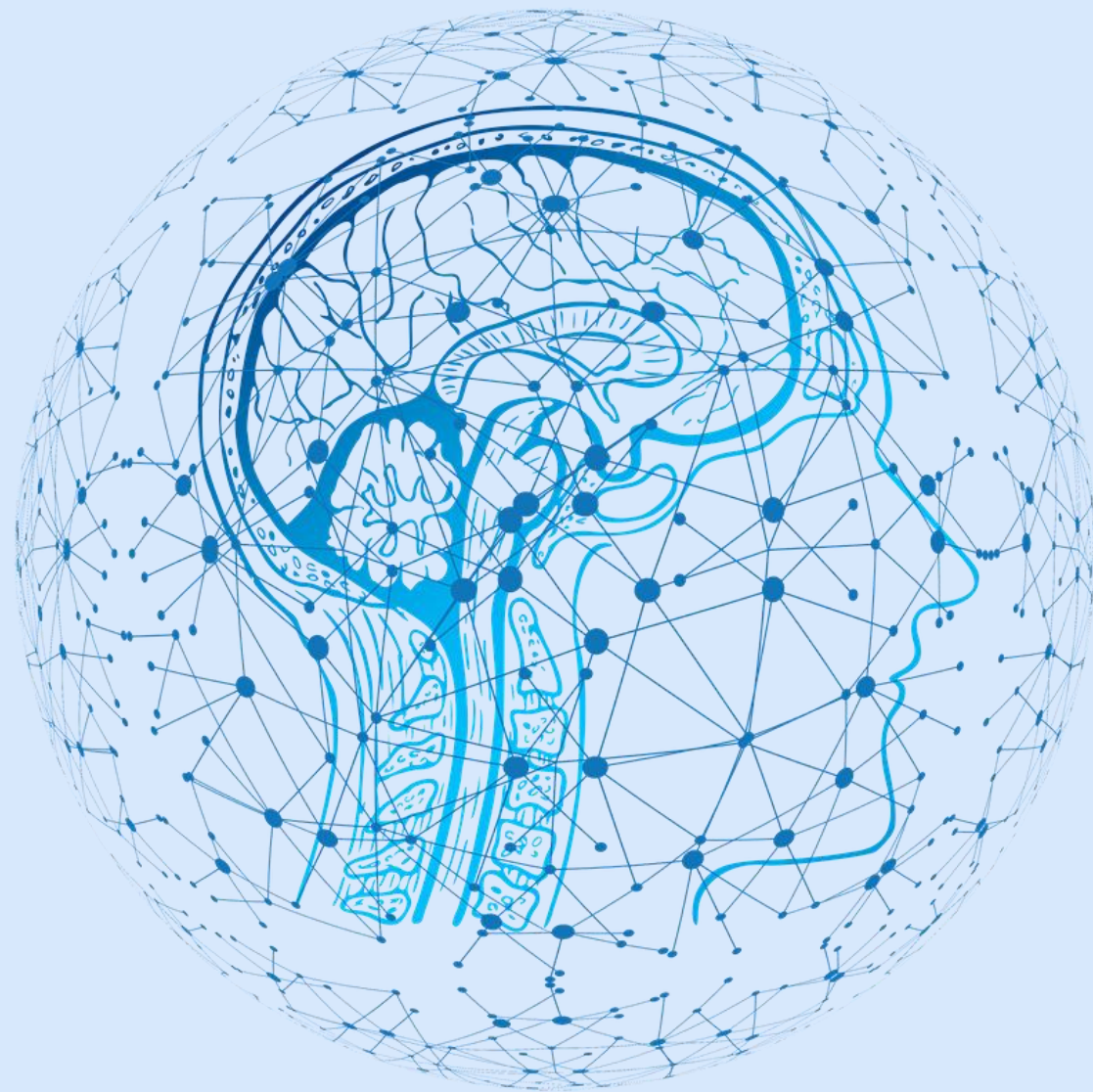
프로젝트 개발환경

Visual studio, mssql



# 팀 소개

---



# 팀 소개

## - 앓 아 씨 -

팀 ' 앓아씨 ' 는 코딩을 공부하고자 모인 4인이 만든 팀으로, 이 팀명은 ' 코딩을 하려면 앓아서 써야 한다 ' 라는 직관적인 아이디어에서 비롯되었습니다. 저희 팀은 코드 작성을 통해 프로젝트를 진행하고, 그 결과물을 통해 지식과 기술을 공유하며 성장하는 목표를 가지고 있습니다.

# 구성원 소개

---



# 구성원 소개



이현오



장광수



정인혁



최효정

공동 설계 및 디버그 확인 작업



# 프로젝트 수행절차

---



# 프로젝트 수행절차

STEP1(11/09~11/10)	STEP2(11/11~11/13)	STEP3(11/14~11/19)	STEP4(11/20~11/22)	STEP5(11/23~11/24)
주제선정 API 지정	맛집API DB연동, 회원가입시스템구축	지도API DB연동 조회 시스템 구축	UI재구성 및 오류수정	PPT 작성 및 발표준비

# 프로젝트 수행절차

## 이용 데이터 소개

### 1. 맛집 API

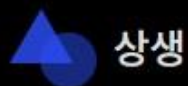
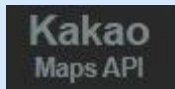


#### JSON 대구광역시\_맛집

대구푸드 홈페이지([www.daegufood.go.kr](http://www.daegufood.go.kr))에 있는 맛집 정보 api 입니다.

주소, 음식카테고리, 음식점명, 연락처, 영업시간, 좌석수, 주차장, 웹사이트, 가능언어, 예약가능여부, 놀이시설여부, 조식여부, 후식여부, 메뉴 등의 항목을 제공합니다.

### 2. 지도 API



#### 상생

인간은 바로 사람과 사람이 서로 기대어 구성됩니다. 상생은 상대의 부정이 아닌 긍정을 통해 이룰 수 있는 세계입니다. 카카오는 상생을 추구합니다. 카카오는 수많은 개발자분들과 함께, 개발자분들에 의해, 개발자분들을 위해 같이 성장하고 싶습니다.

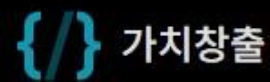
#together, #winwin



#### 생산성

개발자의 생산성 향상은 우리의 중요한 목표 중 하나입니다. 개발자분들 이 앱의 핵심 로직에 좀 더 집중하여, 보다 가치있는 앱들을 창출할 수 있도록 편리한 환경을 제공하는것, 이것이 우리가 추구하는 방향입니다.

#productivity



#### 가치창출

카카오는 새로운 문화를 개척하고 다양한 부가가치를 창출하는 창의적인 개발자분들을 존중합니다. 열린 기술, 열린 서비스, 하지만 정제되고 절제된 소중한 서비스들을 통해, 보다 나은 새로운 세상, 보다 의미있는 사용자 가치를 만들어 나가고 싶습니다.

#value, #create

# 프로젝트 수행절차

## 테이블 명세서

### 테이블 명세서

데이터베이스 명	User_Info		테이블 명	Information		
순번	컬럼 설명	컬럼명	타입 및 길이	NULL	PK	FK
1	유저 아이디	User_ID	varchar(50)	NOT NULL	Yes	No
2	유저 비밀번호	User_PassWord	varchar(50)	NOT NULL	No	No
3	유저 이름	User_Name	varchar(50)	NOT NULL	No	No
4	유저 질문	User_Ques	varchar(50)	NOT NULL	No	No
데이터베이스 명	User_Info		테이블 명	MyPlace		
순번	컬럼 설명	컬럼명	타입 및 길이	NULL	PK	FK
1	유저 아이디	User_ID	varchar(50)	NULL	Yes	No
2	음식점명	BZ_NM	varchar(MAX)	NULL	No	No
3	음식카테고리	FD_CS	varchar(MAX)	NULL	No	No
4	주소	GNG_CS	varchar(MAX)	NULL	No	No
5	연락처	TLNO	varchar(MAX)	NULL	No	No
데이터베이스 명	GoodRestaurant		테이블 명	RestaurantData		
순번	컬럼 설명	컬럼명	타입 및 길이	NULL	PK	FK
1	순번	cnt	int	NULL	No	No
2	고유 번호	OPENDATA_ID	int	NOT NULL	Yes	No
3	주소	GNG_CS	varchar(MAX)	NULL	No	No
4	음식카테고리	FD_CS	varchar(MAX)	NULL	No	No
5	음식점명	BZ_NM	varchar(MAX)	NULL	No	No
6	연락처	TLNO	varchar(MAX)	NULL	No	No
7	영업시간	MBZ_HR	varchar(MAX)	NULL	No	No
8	좌석수	PKPL	varchar(MAX)	NULL	No	No
9	주차장	HP	varchar(MAX)	NULL	No	No
10	웹사이트	BKN_YN	varchar(MAX)	NULL	No	No
11	가능언어	INFN_FCL	varchar(MAX)	NULL	No	No
12	예약가능여부	MNU	varchar(MAX)	NULL	No	No
13	놀이시설여부	SMPL_DESC	varchar(MAX)	NULL	No	No
14	조식여부	SEAT_CNT	varchar(MAX)	NULL	No	No
15	후식여부	SBW	varchar(MAX)	NULL	No	No
16	메뉴	PSB_FRN	varchar(MAX)	NULL	No	No
17	설명	BUS	varchar(MAX)	NULL	No	No
18	지하철 오시는길	BRFT_YN	varchar(MAX)	NULL	No	No
19	버스 오시는길	DSSRT_YN	varchar(MAX)	NULL	No	No

# 프로젝트 수행절차

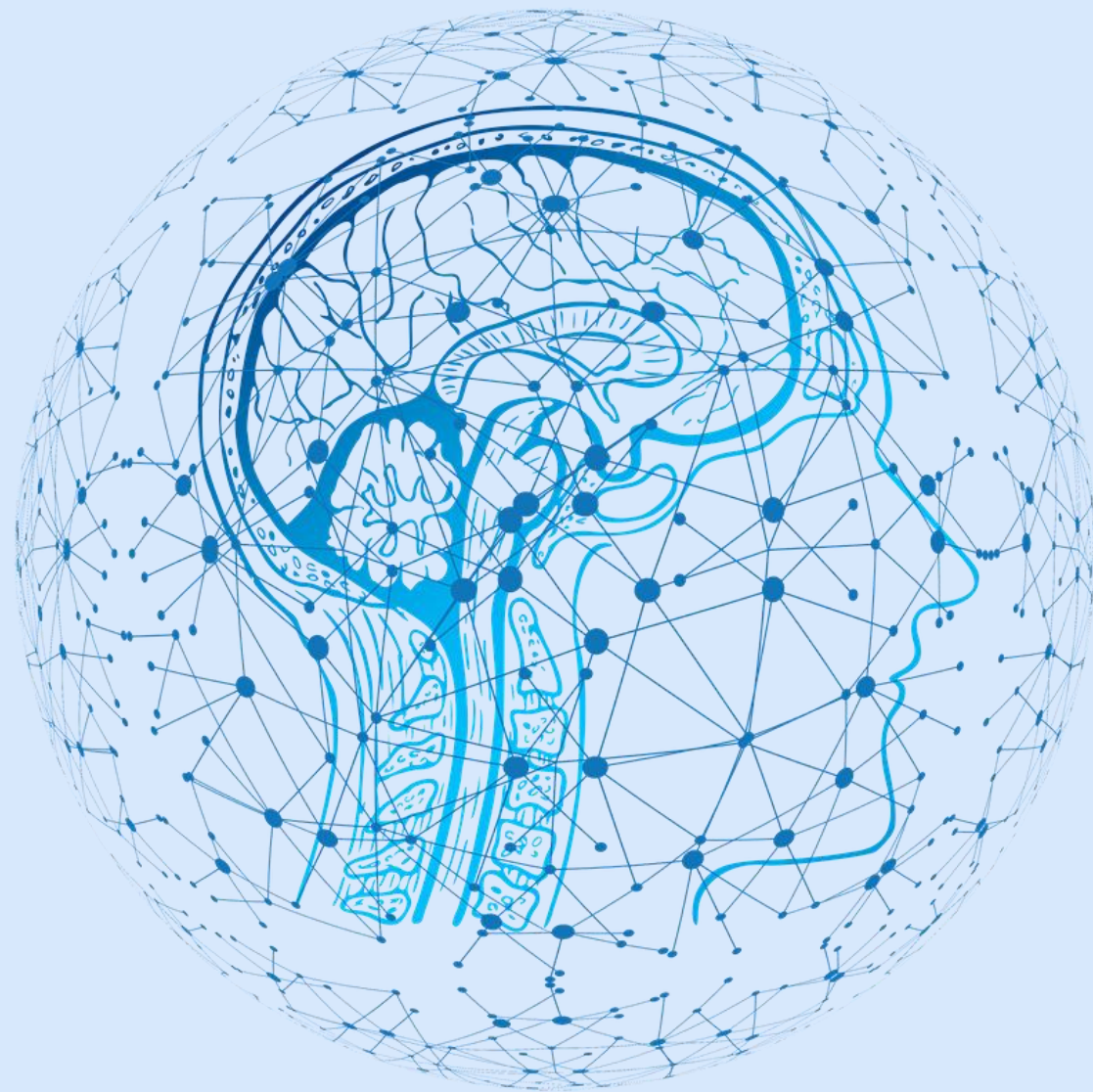
## 요구사항 명세서

요 구 사 항 명 세 서			
No.	기능 요구사항	관리자	내용
RQ-01	로그인 및 회원가입		
	아이디 입력	아이디 입력 텍스트 박스	회원의 아이디를 입력하는 텍스트 박스
	비밀번호 입력	비밀번호 입력 텍스트 박스	회원의 비밀번호를 입력하는 텍스트 박스
	아이디 찾기	아이디 찾기 버튼	아이디 찾기 버튼을 누르면 새로운 폼이 열리면서 아이디를 찾을 수 있다
	비밀번호 찾기	비밀번호 찾기 버튼	비밀번호 찾기 버튼을 누르면 새로운 폼이 열리면서 아이디를 찾을 수 있다
	회원가입	회원가입 버튼	회원추가 버튼을 누르면 새로운 폼이 열리면서 새로운 회원을 등록할 수 있다
RQ-02	맛집 조회		
	각 구별 맛집 조회	지도에 각 구역마다 있는 버튼	버튼을 누르면 새 폼이 뜨면서 그에 해당하는 구 별 맛집이 데이터그리드뷰에 뜨고 카테고리별 분류도 추가.
	즐거찾기 추가	즐거찾기 추가 버튼	해당 맛집 버튼 클릭하면 즐거찾기에 추가하도록
	위치	WebBrowser	웹 브라우저에 해당 맛집 Kakao 지도 연동하여서 뜨게하기
RQ-03	즐거찾기 한 맛집 조회		
	즐거찾기 조회	MyPlace 버튼	로그인 한 ID별 즐거찾기 한 맛집 조회
	즐거찾기 삭제	삭제 버튼	즐거찾기 제거 기능
RQ-04	맛집 데이터 분석		
	데이터 차트	맛집 데이터 분석 버튼	전체 구별 식당 개수 및 구별 맛집 개수 카테고리별 맛집 개수 보이게



# 프로젝트 수행결과

---



# 프로젝트 수행결과

## 1. 로그인 시스템 구축 소스코드

### 1) DBUserHelper.cs

```
public static void ConnectDB()
{
    string dataSource = "local";
    string db = "User_info"; // DB명
    string security = "SSPI";
    conn.ConnectionString =
        $"Data Source={dataSource}; initial Catalog={db};" +
        $"integrated Security={security};" +
        "Timeout=3";

    if (conn.State != ConnectionState.Open)
        conn.Open();
}
```

DB연동 메서드 : ConnectDB

```
public static void RegisterUser(string userId, string password, string username, string userQues)
{
    try
    {
        if (CheckUserIdExists(userId) > 0)
        {
            Console.WriteLine("이미 존재하는 아이디입니다.");
            return;
        }

        ConnectDB();
        string insertQuery = $"INSERT INTO {TABLENAME} (User_ID, User_PassWord, User_Name, User_Ques)" +
            $" VALUES (@UserId, @Password, @Username, @UserQues)";
        using (SqlCommand insertCommand = new SqlCommand(insertQuery, conn))
        {
            insertCommand.Parameters.AddWithValue("@UserId", userId);
            insertCommand.Parameters.AddWithValue("@Password", password);
            insertCommand.Parameters.AddWithValue("@Username", username);
            insertCommand.Parameters.AddWithValue("@UserQues", userQues);
            int rowsAffected = insertCommand.ExecuteNonQuery();

            if (rowsAffected <= 0)
            {
                Console.WriteLine("회원가입에 실패하였습니다.");
            }
        }
    }
    finally
    {
        CloseConnection();
    }
}
```

회원가입 메서드 : RegisterUser

# 프로젝트 수행결과

## 1. 로그인 시스템 구축 소스코드

### 1) DBUserHelper.cs

```
public static int CheckUserIdExists(string userId)
{
    ConnectDB();
    try
    {
        string checkUserQuery = $"SELECT COUNT(*) FROM {TABLENAME} WHERE User_ID = @UserId";
        using (SqlCommand checkUserCommand = new SqlCommand(checkUserQuery, conn))
        {
            checkUserCommand.Parameters.AddWithValue("@UserId", userId);
            return (int)checkUserCommand.ExecuteScalar();
        }
    }
    finally
    {
        CloseConnection();
    }
}
```

중복체크 메서드 : CheckUserIdExists

```
public static string FindUserIdByName(string name)
{
    // DB 연동
    ConnectDB();
    try
    {
        // SQL 쿼리
        string query = $"SELECT User_ID FROM information WHERE User_Name = @UserName";

        // SQL 쿼리 명령 객체 생성
        using (SqlCommand command = new SqlCommand(query, conn))
        {
            // 매개 변수 추가
            command.Parameters.AddWithValue("@UserName", name);

            // 쿼리 실행 및 결과 반환
            object result = command.ExecuteScalar();

            // 결과가 있으면 해당 아이디 반환, 없으면 null 반환
            return result != null ? result.ToString() : null;
        }
    }
    finally { CloseConnection(); }
}
```

ID찾기 메서드 : FindUserIdByName

# 프로젝트 수행결과

## 2. 대구광역시 맛집API DB연동 소스 코드

### 1) DBHelper.cs

```
private static void ConnectDB()
{
    string dataSource = "local";
    string db = "GoodRestaurant"; //DB명
    string security = "SSPI";
    conn.ConnectionString =
        $"Data Source={dataSource}; initial Catalog={db};" +
        $"integrated Security={security};" +
        "Timeout=3";
    conn.Open();
}
```

DB연동 메서드 : ConnectDB

```
public static void selectQuery()
{
    try
    {
        ConnectDB(); //DB 연결
        SqlCommand cmd = new SqlCommand();
        cmd.Connection = conn;
        cmd.CommandText = $"select *from {TABLENAME}";
        da = new SqlDataAdapter(cmd); //쿼리문을 활용하여 데이터 불러오는 것.
        ds = new DataSet();
        da.Fill(ds, TABLENAME); //ds 에 테이블을 채워넣음.
        dt = ds.Tables[0]; //만약 여러개의 테이블을 불러왔다면 그 중 첫번째꺼 갖고 옴.
    }
    catch (Exception)
    {
    }
    finally
    {
        conn.Close(); //DB연결이 잘못되더라도 연결 끊어주는 건 꼭 해줌
    }
}
```

DB데이터 조회 메서드 : selectQuery

# 프로젝트 수행결과

## 2. 대구광역시 맛집API DB연동 소스 코드

### 2) DataManager.cs

```
public static void majipLoad()
{
    matJips.Clear();
    foreach (DataRow item in DBHelper.dt.Rows)
    {
        GoodMatJip Matjips = new GoodMatJip();
        // Matjips.OPENDATA_ID = item["OPENDATA_ID"].ToString();
        Matjips.GNG_CS = item["GNG_CS"].ToString(); // 주소
        Matjips.FD_CS = item["FD_CS"].ToString(); // 카테고리
        Matjips.BZ_NM = item["BZ_NM"].ToString(); // 매장명
        Matjips.MBZ_HR = item["MBZ_HR"].ToString(); // 영업시간
        Matjips.MNU = item["MNU"].ToString().Replace("<br />", " "); //메뉴
        Matjips.TLNO = item["TLNO"].ToString(); // 매장 전화번호
        Matjips.SMPL_DESC = item["SMPL_DESC"].ToString().Replace("<br />", " "); // 매장소개
        Matjips.BKN_YN = item["BKN_YN"].ToString(); // 예약가능여부
        matJips.Add(Matjips);
    }
}
```

맛집 조회 메서드 : majipLoad

```
public static void Gu_Load(string gu)
{
    try
    {
        DBHelper.selectQuery();
        majipLoad();
        //2. LINQ
        var mylist = from c in matJips
                      where c.GNG_CS.Contains(" " + gu + " ")
                      select c;
        SelectMatJips = mylist.ToList<GoodMatJip>();
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
        Console.WriteLine(e.StackTrace);
    }
}
```

구별 조회 메서드 : Gu\_Load



# 프로젝트 수행결과

## 2. 대구광역시 맛집API DB연동 소스 코드

### 2) DataManager.cs

```
public static void Cg_Load(string cg)
{
    try
    {
        DBHelper.selectQuery();
        majipLoad();

        // LINQ 쿼리로 카테고리에 해당하는 데이터 조회
        var mylist = from c in SelectMatJips

                     where c.FD_CS.Contains(cg)

                     select c;
        SelectCategoryMatJips = new List<GoodMatJip>();
        SelectCategoryMatJips = mylist.ToList<GoodMatJip>();
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
        Console.WriteLine(e.StackTrace);
    }
}
```

음식 카테고리별 조회 메서드 : Cg\_Load

# 프로젝트 수행결과

## 3. Kakao 지도API 이용 소스 코드

### 1) KakaoAPI.cs

```
public class KakaoAPI
{
    //주소로 검색
    참조 1개
    public static Locale SelectMap(string text)
    {
        string url = "https://dapi.kakao.com/v2/local/search/address.json";
        string query = $"{url}?analyze_type=similar&page=1&size=10&query={text}";
        string restAPIKey = "799a3031dec6472f3a94b15adf4b9b70";
        string Header = $"KakaoAK {restAPIKey}";
        WebRequest request = WebRequest.Create(query);
        request.Headers.Add("Authorization", Header);

        //응답받기
        WebResponse response = request.GetResponse();
        Stream stream = response.GetResponseStream();
        StreamReader reader = new StreamReader(stream, Encoding.UTF8);
        string json = reader.ReadToEnd();
        JavaScriptSerializer js = new JavaScriptSerializer();
```

```
dynamic dob = js.Deserialize<dynamic>(json);
dynamic docs = dob["documents"][0];

string lname = docs["address_name"];
double x, y;

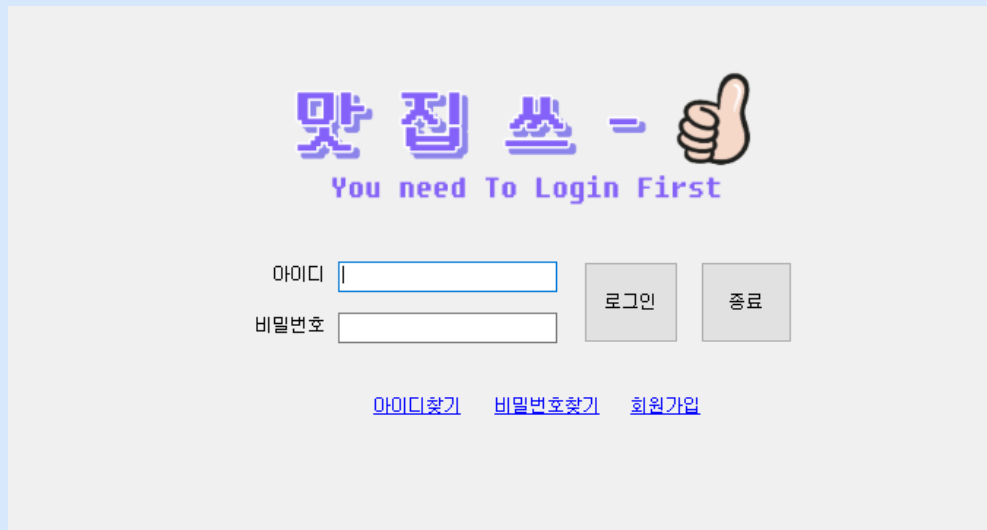
// road_address가 null인 경우에 대한 처리
if (docs["road_address"] != null)
{
    x = double.Parse(docs["road_address"]["x"]);
    y = double.Parse(docs["road_address"]["y"]);
}
else
{
    x = double.Parse(docs["x"]);
    y = double.Parse(docs["y"]);
}
return new Locale(lname, y, x);
```

Kakao 지도API이용 메서드 : SelectMap

# 프로젝트 수행결과

## 4. 애플리케이션 구조

### 1) Login.cs



The login form has a light yellow background. At the top, it features the text '맛집쓰 = 👍' in purple and 'You need To Login First' in blue. Below this, there are two input fields: '아이디' (ID) and '비밀번호' (Password). To the right of the ID field is a '로그인' (Login) button, and to the right of the password field is a '종료' (End) button. At the bottom, there are three links: '아이디찾기' (Find ID), '비밀번호찾기' (Find Password), and '회원가입' (Sign Up).

```
// 아이디 찾기
참조 1개
private void linkLabelID_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    FindID findID = new FindID();
    findID.ShowDialog();
}

// 비밀번호 찾기
참조 1개
private void linkLabelPWD_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    FindPWD findPWD = new FindPWD();
    findPWD.ShowDialog();
}

// 회원가입
참조 1개
private void linkLabelSign_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    Sign_Up sign_Up = new Sign_Up();
    sign_Up.ShowDialog();
}
```

아이디찾기, 비밀번호찾기, 회원가입 별 폼 구성

# 프로젝트 수행결과

## 4. 애플리케이션 구조

### 2) Sign Up.cs

**회원가입**

아이디  중복체크

비밀번호

이름

가장 좋아하는 여행지?

Welcome!!!

```
private void DCheckbox_Click(object sender, EventArgs e)
{
    string userId = SignUpID_textBox.Text;

    if (string.IsNullOrEmpty(userId))
    {
        ShowMessageBox("아이디를 입력하세요.", "경고", MessageBoxIcon.Warning);
        return;
    }

    try
    {
        int userCount = DBUserHelper.CheckUserIdExists(userId);

        if (userCount > 0)
        {
            ShowMessageBox("이미 존재하는 아이디입니다.", "경고", MessageBoxIcon.Warning);
        }
        else
        {
            ShowMessageBox("사용 가능한 아이디입니다.", "알림", MessageBoxIcon.Information);
        }
    }
    catch (Exception ex)
    {
        ShowMessageBox("오류 발생: " + ex.Message, "오류", MessageBoxIcon.Error);
    }
}
```

유저 중복 체크 메서드 : Dcheckboxbutton\_Click

# 프로젝트 수행결과

## 4. 애플리케이션 구조

### 2) Sign Up.cs


**회원가입**

아이디  중복체크

비밀번호

이름

가장 좋아하는 여행지?



```
private void NametextBox_TextChanged(object sender, EventArgs e)
{
    // 입력된 글자 수가 10자를 초과하면 마지막 10자를 자름
    if (SignUpName_textBox.Text.Length > 10)
    {
        SignUpName_textBox.Text = SignUpName_textBox.Text.Substring(0, 10);
        SignUpName_textBox.SelectionStart = SignUpName_textBox.Text.Length;
        SignUpName_textBox.SelectionLength = 0;
    }
}
```

글자 제한 메서드 : NametextBox\_TextChanged



# 프로젝트 수행결과

## 4. 애플리케이션 구조

### 3) FindID.cs



```
private void FNCheckbox_Click(object sender, EventArgs e)
{
    try
    {
        // 아이디 찾기
        string findIdByName = DBUserHelper.FindUserIdByName(FIDName_textBox.Text);
        if (findIdByName != null)
        {
            MessageBox.Show($"아이디 : {findIdByName}");
        }
        else
        {
            MessageBox.Show("아이디를 찾을수 없습니다!");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"오류: {ex.Message}");
    }
}
```

확인 시 체크 메서드 : Fncheckboxbutton\_Click

# 프로젝트 수행결과

## 4. 애플리케이션 구조

### 4) FindPWD.cs

**비밀번호 찾기**

아이디

가장 좋아하는 여행지?

I forgot my password!!



```
private void FNCheckbutton_Click(object sender, EventArgs e)
{
    try
    {
        // 비밀번호 찾기 (질문 기반)
        string findPWD = DBUserHelper.FindUserPWDByQues(FPWDName_textBox.Text,FPWDQ_textBox.Text);

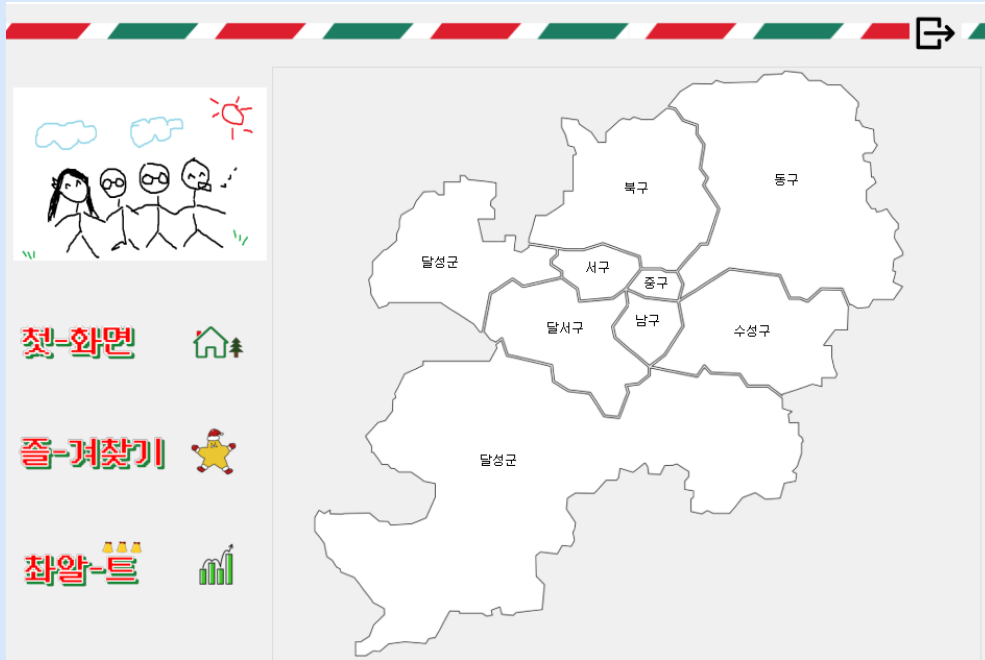
        if (findPWD != null)
        {
            MessageBox.Show($"비밀번호: {findPWD}");
        }
        else
        {
            MessageBox.Show("비밀번호를 찾을 수 없습니다.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"오류: {ex.Message}");
    }
}
```

확인 시 체크 메서드 : Fncheckbutton\_Click

# 프로젝트 수행결과

## 4. 애플리케이션 구조

### 5) MainForm.cs



```
// 동구 버튼 클릭시  
참조 1개  
private void donggu_button_Click(object sender, EventArgs e)  
{  
    DataManager.Gu_Load("동구");  
    matjipdataform.ShowDialog();  
}  
  
// 북구 버튼 클릭시  
참조 1개  
private void bukgu_button_Click(object sender, EventArgs e)  
{  
    DataManager.Gu_Load("북구");  
    matjipdataform.ShowDialog();  
}
```

구 이름 클릭시 -> DataManager의 Gu\_Load메서드 이용

# 프로젝트 수행결과

## 4. 애플리케이션 구조

### 6) MatjipDataForm.cs

The screenshot displays the '맛집쓰' (Matjip) application. On the left, there's a list of restaurants with columns for category, name, address, business hours, menu, and phone number. The selected restaurant is '가송북어' (Gasongbuk-eo). On the right, a detailed view of this restaurant is shown, including its address, phone number, business hours, and a map of its location. The map shows the restaurant is located near '해성빌라' (Haesung Villa) and '보금주택' (Bogum Housing).

```
private void dataGridViewCellClick(object sender, DataGridViewCellEventArgs e)
{
    GoodMatJip m = (sender as DataGridView).CurrentRow.DataBoundItem as GoodMatJip;
    상호명.Text = m.상호명;
    주소.Text = m.주소;
    영업시간.Text = m.영업시간;
    메뉴.Text = m.메뉴;
    매장설명.Text = m.매장설명;
    매장전화번호.Text = m.전화번호;
    카테고리.Text = m.카테고리;
    예약가능여부.Text = m.예약가능여부;
    try
    {
        Locale temp = KakaoAPI.SelectMap(m.주소);
        object[] pos = new object[] { temp.Lat, temp.Lng };
        HtmlDocument hdoc = Majip_webBrowser.Document;
        hdoc.InvokeScript("setCenter", pos);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message + "_" + ex.StackTrace);
    }
}
```

## dataGridViewCellClick 메서드

- 데이터그리드뷰 클릭시 옆 상단 텍스트박스 채우는 기능 및 지도API 이용 위치 조회

# 프로젝트 수행결과

## 4. 애플리케이션 구조

### 6) MatjipDataForm.cs

The screenshot displays the Matjip application interface. On the left, there is a table listing various restaurants with columns for category, name, address, operating hours, menu, and phone number. The table is filtered by 'Category' (한식). On the right, a detailed view of a selected restaurant is shown, including its name, address, phone number, operating hours, menu, and a map of its location. The restaurant selected is '가송북어' (Gasongbuk-eo) located at '대구광역시 동구 방촌동 871-17'.

카테고리	상호명	주소	영업시간	메뉴	전화번호	매
한식	가송북어	대구광역시 동...	10:30 ~ 22:00	모듬찜 : 50,000...	053-984-6755	가
한식	가와집	대구광역시 동...	09:30 ~ 21:30	한정식 10,000...	053-981-0364	가
양식	블루문레스토랑	대구광역시 동...	11:30 ~ 23:00	바다가재와 안...	053-981-8088	19
한식	산중식당	대구광역시 동...	11:30 ~ 21:30	콘드레밥보쌈...	053-982-0077	팔
한식	원조할매두레...	대구광역시 동...	10:30 ~ 20:30	머할 할국수 9...	053-985-5644	속
한식	삼영식당	대구광역시 동...	09:00 ~ 21:00	[채식]매뉴 주문...	053-982-0471	삼
한식	보급신도불이	대구광역시 동...	11:00 ~ 12:00	쌈밥 13,000원 ...	053-983-0666	소
한식	안심농장직영...	대구광역시 동...	09:30 ~ 21:30	물티기(300g) 3...	053-964-0022	안
한식	영화식당	대구광역시 동...	10:30 ~ 21:00	송이도리탕 65...	053-983-6740	영
한식	팔공함한우마살	대구광역시 동...	10:00 ~ 22:00	갈비살(120g) 2...	053-982-3382	팔
한식	천서리원조매...	대구광역시 동...	10:00-18:00	한방수육 25,00...	053-981-6001	천
한식	신암태양칼국수	대구광역시 동...	10:00 ~ 22:00	칼국수 6,500원...	053-951-0321	동
한식	대구가든	대구광역시 동...	11:00 ~ 22:00	맛갈비(250g) 2...	053-964-0677	순
한식	명산한우식육...	대구광역시 동...	11:30 ~ 21:30	한우갈비찜(1...	053-982-1999	10
한식	한정식금마루	대구광역시 동...	10:00 ~ 22:00	금마루 퓨전한...	053-942-0088	마
한식	갈안매운탕식당	대구광역시 동...	11:30 ~ 20:00	갈아매운탕 30...	053-951-0013	소
한식	황토시골집	대구광역시 동...	24시간	오리훈제 43,00...	053-982-2002	시
한식	꼬꼬하우스	대구광역시 동...	12:00 ~ 24:00	튀김통집 10,00...	053-956-7851	7
한식	일경식당	대구광역시 동...	09:30 ~ 23:00	명품왕순대 20...	053-753-4778	일

```
private void MyPlace_button_Click(object sender, EventArgs e)
{
    //현재 로그인한 사용자의 ID를 가져오기
    string userId = DBUserHelper.GetCurrentUserId();

    //입력된 정보 가져오기
    string bz_nm = 상호명.Text;
    string fd_cs = 카테고리.Text;
    string gng_cs = 주소.Text;
    string tlnr = 매장전화번호.Text;

    //MyPlace 등록

    DBMyPlaceHelper.RegisterMyPlace(userId, 상호명.Text, 카테고리.Text, 주소.Text, 매장전화번호.Text);
}
```

즐거찾기 등록 메서드  
MyPlace\_button\_Click



# 프로젝트 수행결과

## 4. 애플리케이션 구조

### 7) Myplace.cs(userControl)



개인별 맛집 즐겨찾기

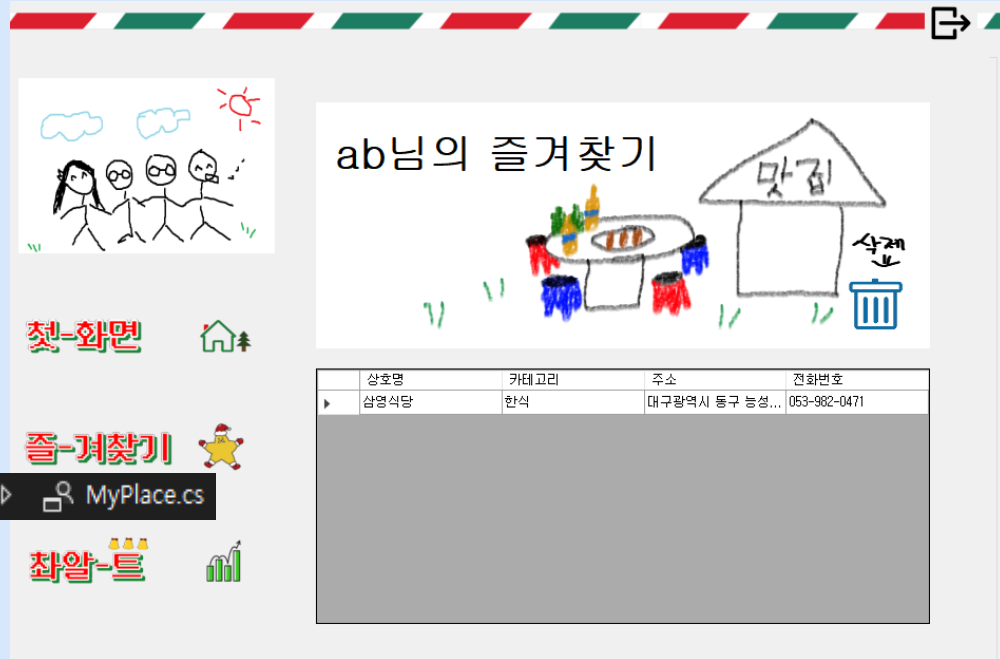
```
public void BindMyPlaceData()
{
    try
    {
        // MyPlace 테이블의 데이터 가져오기
        DataTable myPlaceData = DBMyPlaceHelper.GetMyPlaceData(DBUserHelper.GetCurrentUserId());
        MyPlace_dataGridView.DataSource = null;
        // DataGridView에 데이터 바인딩
        if(myPlaceData.Rows.Count > 0 )
        {
            MyPlace_dataGridView.DataSource = myPlaceData;
            MyPlace_dataGridView.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
            MyPlace_dataGridView.Columns[0].HeaderText = "상호명";
            MyPlace_dataGridView.Columns[1].HeaderText = "카테고리";
            MyPlace_dataGridView.Columns[2].HeaderText = "주소";
            MyPlace_dataGridView.Columns[3].HeaderText = "전화번호";
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show($"데이터를 불러오는 중 오류가 발생했습니다: {ex.Message}", "오류",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

즐거찾기 조회 메서드 : BindMyPlaceDate

# 프로젝트 수행결과

## 4. 애플리케이션 구조

### 7) Myplace.cs(userControl)



개인별 맛집 즐겨찾기

```
private void DeleteSelectedRow()
{
    if (MyPlace_dataGridView.SelectedRows.Count > 0)
    {
        int selectedIndex = MyPlace_dataGridView.SelectedRows[0].Index;
        string userId = DBUserHelper.GetCurrentUserId();
        try
        {
            // 해당 행을 데이터베이스에서도 삭제
            string bz_nm = MyPlace_dataGridView.SelectedRows[0].Cells["BZ_NM"].Value.ToString(); // BZ_NM에 해당하는 셀의 값을
            DBMyPlaceHelper.DeleteMyPlace(userId, bz_nm); // 데이터베이스에서 삭제하는 메소드 호출

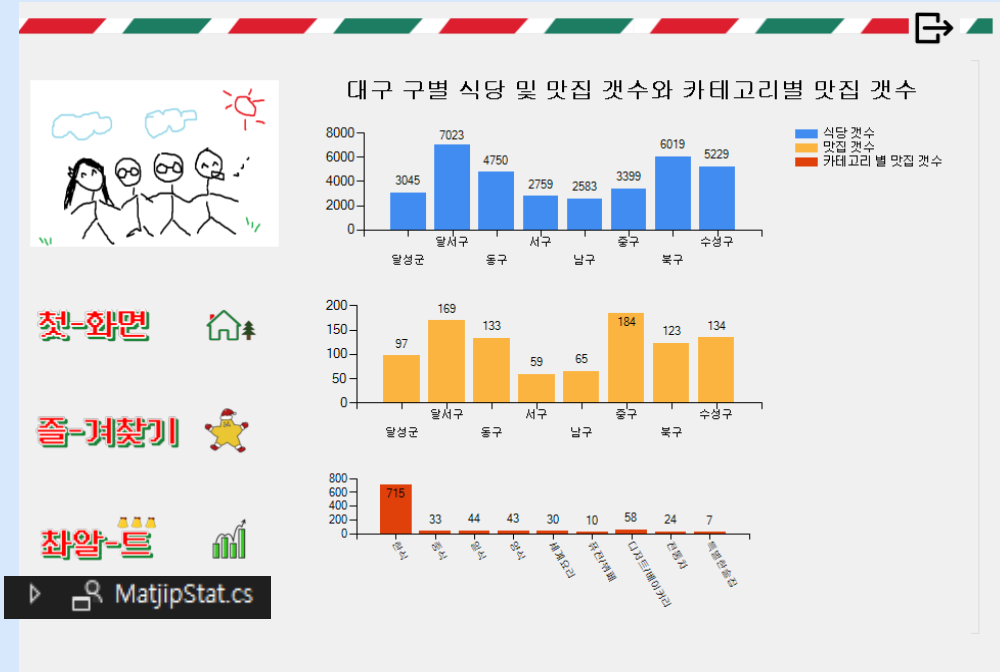
            // DataGridView에서 선택된 행 삭제
            MyPlace_dataGridView.Rows.RemoveAt(selectedIndex);
        }
        catch (Exception ex)
        {
            MessageBox.Show($"삭제 중 오류가 발생했습니다: {ex.Message}", "오류", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

즐거찾기 삭제 메서드 : DeleteSelectedRow

# 프로젝트 수행결과

## 4. 애플리케이션 구조

### 7) MatjipStat.cs(userControl)



```
List<string> x = new List<string> { "달성군", "달서구", "동구", "서구", "남구", "중구", "북구", "수성구" };
List<double> y1 = new List<double> { 3045, 7023, 4750, 2759, 2583, 3399, 6019, 5229 };
List<double> y2 = new List<double> { 97, 169, 133, 59, 65, 184, 123, 134 };
List<string> x1 = new List<string> { "한식", "중식", "일식", "양식", "세계요리", "퓨전/뷔페", "디저트/베이커" };
List<double> y3 = new List<double> { 715, 33, 44, 43, 30, 10, 58, 24, 7 };
```

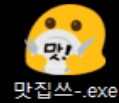
```
Matjip_chart.Titles.Add("대구 구별 식당 및 맛집 갯수와 카테고리별 맛집 갯수");
Matjip_chart.Titles[0].Font = new Font("Arial", 16, FontStyle.Bold);
Matjip_chart.Series[0].Name = "식당 갯수";
Matjip_chart.ChartAreas[0].BackColor = SystemColors.Control;
Matjip_chart.Series[0].Points.DataBindXY(x, y1);
```

```
Matjip_chart.ChartAreas.Add("ChartArea2");
Matjip_chart.Series.Add("맛집 갯수");
Matjip_chart.Series[1].ChartArea = "ChartArea2";
Matjip_chart.ChartAreas[1].BackColor = SystemColors.Control;
Matjip_chart.Series[1].Points.DataBindXY(x, y2);
```

```
Matjip_chart.ChartAreas.Add("ChartArea3");
Matjip_chart.Series.Add("카테고리 별 맛집 갯수");
Matjip_chart.Series[2].ChartArea = "ChartArea3";
Matjip_chart.ChartAreas[2].BackColor = SystemColors.Control;
Matjip_chart.Series[2].Points.DataBindXY(x1, y3);
```

# 프로젝트 수행결과

## 5. 시연 영상

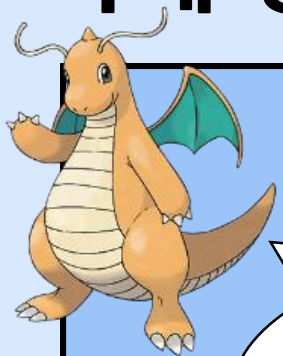


# 자체 평가 의견

---



# 자체 평가 의견



이 현 오

C# 프로젝트를 DB와 처음 연동하여 프로그램을 만들어 보았는데 처음 해보는 프로젝트에 막막함이 있었으나 프로젝트 진행하며 많은 것을 배울 수 있어 좋은 기회였습니다.

장 광 수

프로젝트를 통해 C#의 객체지향적 특성을 활용하여 효율적이고 모듈화된 코드를 작성하는 법에 대한 이해를 높일 수 있었습니다.



정 인 혁

프로젝트를 진행하면서 초반에 주제도 한 번 바꾸고 어떻게 할 지 몰라 막막했는데 그래도 조원들 모두 힘을 합쳐 잘 마무리 지을 수 있어서 좋았습니다!!! 맛집 ㄱㄱ



최 효 정

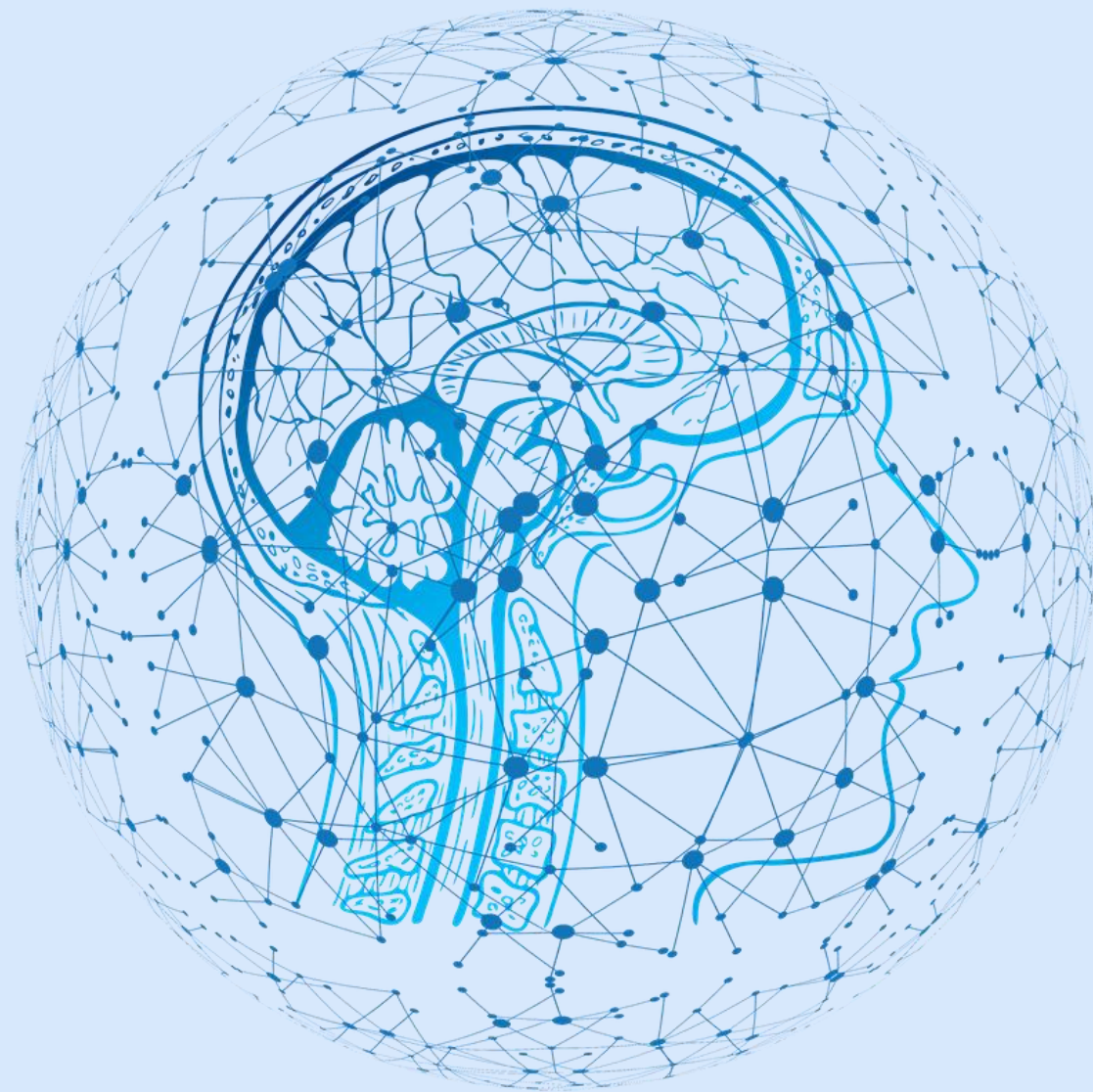
C#숙지가 덜 된 상태로 프로젝트를 시작해 주제선정부터 어려움이 있었지만 그래도 다 같이 노력해 결과물을 만들어 낸 거 같아 뿌듯하다. 프로젝트를 하면서 실력이 는 거 같기도! ㅋㅋㅋ





# 질문 있으신 분?!

---



**감 사 합 니 다**

---