



약쟁이가 되고 싶은 당신! 주목하세요

- 01 프로젝트 개요
- 02 팀 구성원 역할 소개
- 03 프로젝트 수행 절차
- 04 프로젝트 수행 결과
- 05 자체 평가 의견

01

프로젝트 개요

1. 프로젝트 개요

최근 자꾸만 늘어가는 유행성 질병(코로나, 독감)등
확진자의 수가 늘어남에 따라 병원, 약국에 가는 것 마
저 불안해하는 사람이 늘어나고 있어 이에 따라 최신
트랜드는 직접 매장 방문이 아닌 주문이라는 것에 컨셉
을 둬 근처 약국에서 물품을 떼와 배송해주는 주문 플
랫폼을 만들어보자는 취지 하에 제작을하게 되었음.

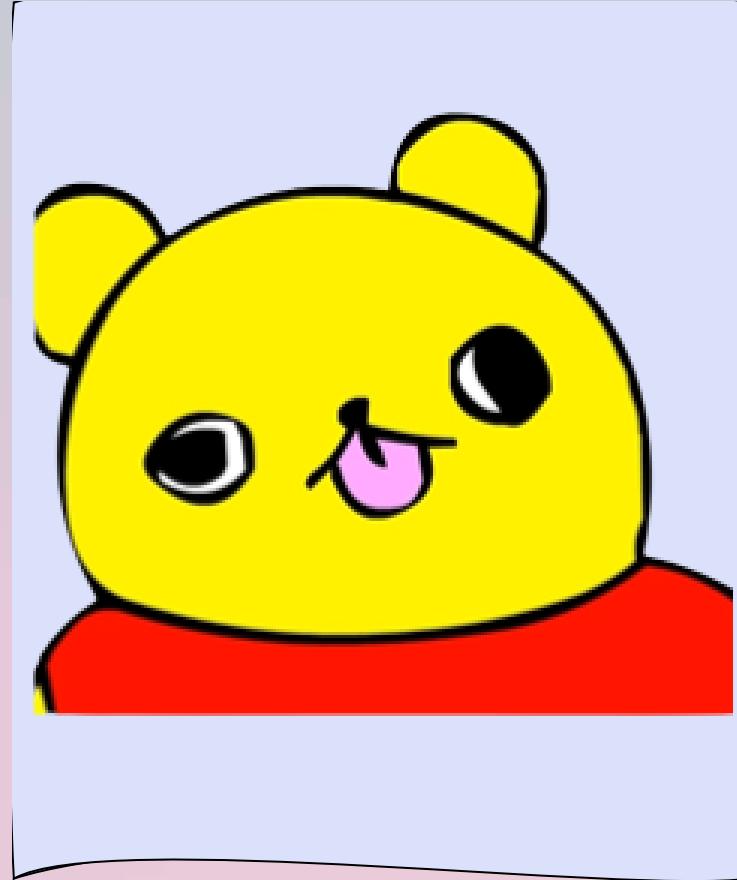
빨리 나아요



02

팀 구성원 역할 소개

2. 팀 구성원 역할 소개



조장 권병진

- 코드 총괄



조원 신예원

- 주문수정 코드



조원 백정윤

- FORM 디자인



조원 홍주환

- 자료수집 및 삽화 조사

03

프로젝트 수행 절차

3. 프로젝트 수행절차 프로젝트 일정



3. 프로젝트 수행절차 테이블 명세서

Medicine 테이블			
논리이름	물리이름	데이터 타입	ALLOW NULL
회사	Company	VARCHAR(50)	
제품명	Name	VARCHAR(50)	
증상	Sick	VARCHAR(50)	
가격	Price	VARCHAR(50)	

User 테이블			
논리이름	물리이름	데이터 타입	ALLOW NULL
id	ID	varchar(50)	
주소	Addr	varchar(50)	
이름	Name	varchar(50)	
비밀번호	Password	varchar(50)	
가입날짜	JoinDate	date	

Customer 테이블			
논리이름	물리이름	데이터 타입	ALLOW NULL
이름	Name	varchar(50)	
배송지	Addr	varchar(50)	
제품명	Mname	varchar(50)	
가격	Price	varchar(50)	

- 200개 이상의 약품 정보를 DB에 저장
- 조회 메뉴에서 사용자가 약품을 조회 할 때 DB에 저장된 데이터들이 출력됨

- 회원가입 시 회원 정보를 DB에 저장
- 개인정보 수정에서 비밀번호, 이름, 주소를 변경할 수 있음

- 주문 시 고객정보를 DB에 저장
- 주문정보 수정에서 이름과 배송지를 변경 할 수 있음

3. 프로젝트 수행절차

DB 커넥션 소스코드

```
public class DBConnect
{
    private const string ConnectionUser = "Data Source=(local);Initial Catalog=User;Integrated Security=True";
    private const string ConnectionMedicine = "Data Source=(local);Initial Catalog=Medicine;Integrated Security=True";
    private const string ConnectionPocket = "Data Source=(local);Initial Catalog=Customer;Integrated Security=True";

    참조 1개
    public static bool RegisterUser(string id, string password, string name, string address)
    {
        try
        {
            using (SqlConnection connection = new SqlConnection(ConnectionUser))
            {
                connection.Open();

                string query = "INSERT INTO [user] (id, password, name, addr) VALUES (@id, @password, @name, @address)";
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    command.Parameters.AddWithValue("@id", id);
                    command.Parameters.AddWithValue("@password", password);
                    command.Parameters.AddWithValue("@name", name);
                    command.Parameters.AddWithValue("@address", address);

                    int rowsAffected = command.ExecuteNonQuery();

                    return rowsAffected > 0;
                }
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
            return false;
        }
    }

    참조 1개
    public static bool UpdateUser(string id, string password, string name, string address)
    {
        try
        {
            using (SqlConnection connection = new SqlConnection(ConnectionUser))
            {
                connection.Open();

                string query = "update [User] set password=@password,name=@name,addr=@address where id=@id";
                using (SqlCommand command = new SqlCommand(query, connection))
                {

```

- 후에 설명할 customer DB의 insert문과 동일

3. 프로젝트 수행절차

DB 커넥션 소스코드

```
        command.Parameters.AddWithValue("@password", password);
        command.Parameters.AddWithValue("@name", name);
        command.Parameters.AddWithValue("@address", address);
        command.Parameters.AddWithValue("@id", id);

        int rowsAffected = command.ExecuteNonQuery();

        return rowsAffected > 0;
    }
}

catch (Exception ex)
{
    Console.WriteLine($"Error: {ex.Message}");
    return false;
}
}

참조 1개
public static bool UpdateOrder(string name, string addr, string Mname, string Price, string num)
{
    try
    {
        using (SqlConnection connection = new SqlConnection(ConnectionPocket))
        {
            connection.Open();

            string query = "update [Customer] set name=@name,addr=@addr where num=@num";
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                command.Parameters.AddWithValue("@name", name);
                command.Parameters.AddWithValue("@addr", addr);
                command.Parameters.AddWithValue("@Mname", Mname);
                command.Parameters.AddWithValue("@Price", Price);
                command.Parameters.AddWithValue("@num", num);
                int rowsAffected = command.ExecuteNonQuery();

                return rowsAffected > 0;
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
        return false;
    }
}
```

- orderRepair를 통해 주문 내용을 변경할 때, update문을 통해 customer DB의 데이터를 변경시켜준다.

3. 프로젝트 수행절차 DB 커넥션 소스코드

```
public static bool RegisterPocket(string name, string addr, string mname, string price, string num)

try
{
    using (SqlConnection connection = new SqlConnection(ConnectionPocket))
    {
        connection.Open();

        string query = "INSERT INTO [Customer] (name, addr, mname, price,num) VALUES (@name, @addr, @mname, @price, @n
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@name", name);
            command.Parameters.AddWithValue("@addr", addr);
            command.Parameters.AddWithValue("@mname", mname);
            command.Parameters.AddWithValue("@price", price);
            command.Parameters.AddWithValue("@num", num);

            int rowsAffected = command.ExecuteNonQuery();

            return rowsAffected > 0;
        }
    }
}
catch (Exception ex)
{
    Console.WriteLine($"Error: {ex.Message}");
    return false;
}

조 0개
private bool CheckId(string id, string password)

try
{
    using (SqlConnection connection = new SqlConnection(ConnectionUser))
    {
        connection.Open();

        string query = "SELECT COUNT(*) FROM [User] WHERE id = @id AND password = @password";
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            command.Parameters.AddWithValue("@id", id);
            command.Parameters.AddWithValue("@password", password);

            int count = (int)command.ExecuteScalar();
        }
    }
}
```

- search폼을 통해 제품을 선택하고 개인정보를 입력해 구매할 경우 customer DB에 insert문을 이용해 데이터를 저장해준다.

3. 프로젝트 수행절차 회원가입 Form

QMS
회원가입

ID
 중복체크

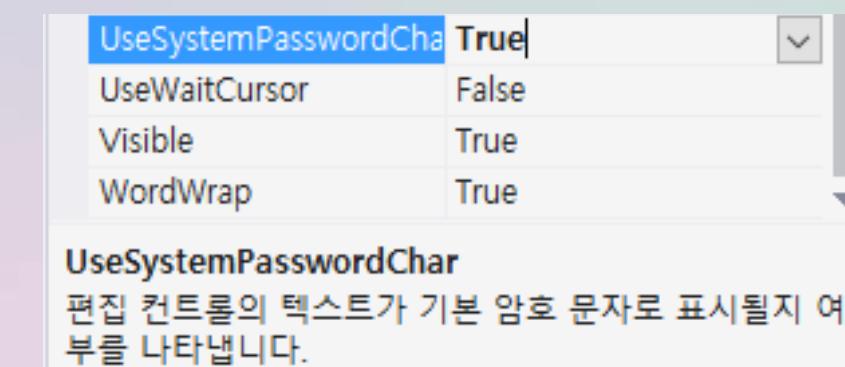
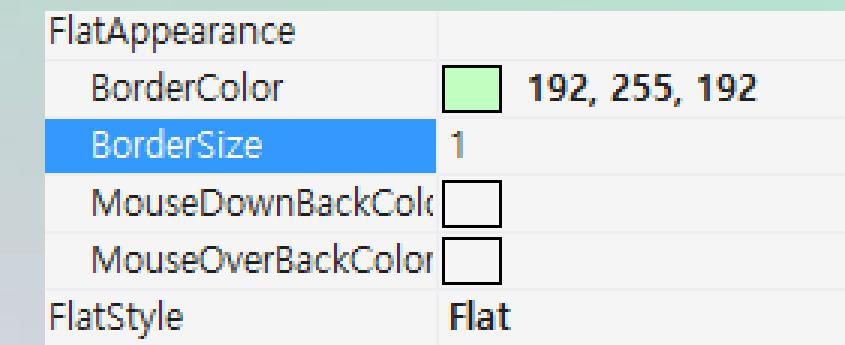
PassWord

PassWord Check

Name

Address

취소 **가입완료**



- 중복체크 버튼이 눈에 띠기 쉽게 버튼 굽기와 색상 조정
- 비밀번호 암호화 하기 위해 텍스트 박스에 기본 암호 문자 설정
- 회원가입 시 입력했던 값들은 DB의 USER 테이블에 저장되며 이미 가입된 ID를 중복체크 할 수 있고, PASSWORD를 다르게 입력했을 시 가입이 불가능함.

3. 프로젝트 수행절차 회원가입 소스코드

```
private void 중복체크_Click(object sender, EventArgs e)
{
    string id = textBox1.Text;

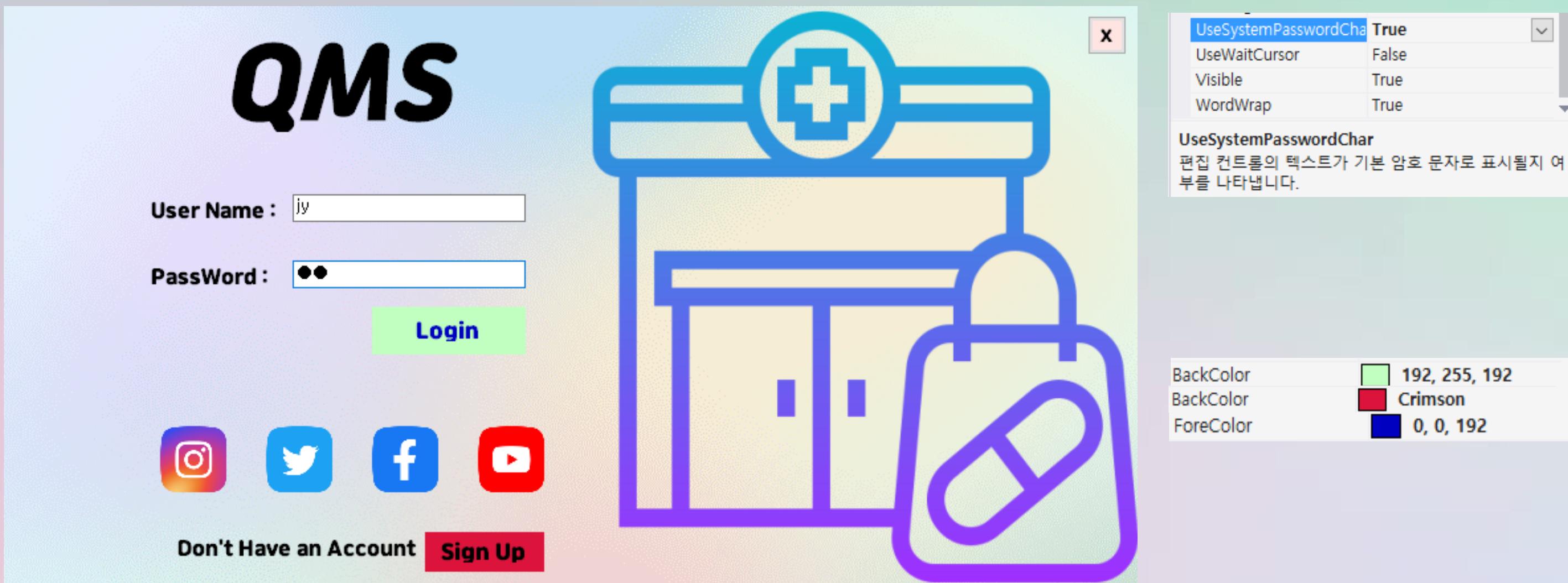
    bool loginSuccess = CheckId(id);

    if (loginSuccess)
    {
        MessageBox.Show("이미 있는 아이디입니다.");
        try
        {
        }
        catch (SqlException x)
        {
            MessageBox.Show(x.Message);
        }
    }
    else
    {
        MessageBox.Show("사용 가능합니다.");
    }
}
```

```
private bool CheckId(string id)
{
    try
    {
        using (SqlConnection conn = new SqlConnection(ConnectionString))
        {
            conn.Open();
            string query = "SELECT COUNT(*) FROM [USER] WHERE id = @id";
            using (SqlCommand command = new SqlCommand(query, conn))
            {
                command.Parameters.AddWithValue("@id", id);
                int count = (int)command.ExecuteScalar();
                return count > 0;
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error : {ex.Message}");
        return false;
    }
}
```

- userDB에 입력받은 ID와 같은 내용이 있다면 ID는 고유키 임으로 사용불가능한 ID임을 알려준다.

3. 프로젝트 수행절차 로그인 Form



- DB와 연동하고 USER 테이블에 저장된 ID와 PASSWORD를 조회 후 값을 입력받아 로그인 실행

- 비밀번호 암호화 하기위해 텍스트 박스에 기본 암호 문자 설정
- 버튼 및 폰트 색상은 가시성에 좋게 설정

3. 프로젝트 수행절차 로그인 소스코드

```
private void 로그인_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "")
    {
        MessageBox.Show("아이디를 입력하세요.");
    }
    else if (textBox2.Text == "")
    {
        MessageBox.Show("비밀번호를 입력하세요.");
    }
    else
    {
        string id = textBox1.Text;
        string password = textBox2.Text;

        bool loginSuccess = CheckLogin(id, password);

        if (loginSuccess)
        {
            MessageBox.Show("로그인 성공!");
            Hide();
            new Title().ShowDialog();
        }
        else
        {
            MessageBox.Show("로그인 실패. 아이디 또는 비밀번호를 확인하세요.");
        }
    }
}
```

```
private bool CheckLogin(string id, string password)
{
    try
    {
        using (SqlConnection connection = new SqlConnection(ConnectionString))
        {
            connection.Open();

            string query = "SELECT COUNT(*) FROM [User] WHERE id = @id AND password = @password";
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                command.Parameters.AddWithValue("@id", id);
                command.Parameters.AddWithValue("@password", password);

                int count = (int)command.ExecuteScalar();

                return count > 0;
            }
        }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
        return false;
    }
}
```

- 입력받은 id와 pw를 매개변수로 CheckLogin메서드로 전달해 user DB에 입력되어 있는 내용을 조회하고 일치하는 내용이 있다면 로그인에 성공하게 된다.
(전과 동일)

3. 프로젝트 수행절차 타이틀 Form



BackColor Transparent



- 버튼 이미지 뒷배경은 투명화 처리

- 타이틀 Form에서 버튼을 누르게 되면 해당 기능 Form으로 이동함

3. 프로젝트 수행절차 타이틀 소스코드

```
private void Exit_Click(object sender, EventArgs e)
{
    Application.Exit();
}

참조 1개
private void 검색_Click(object sender, EventArgs e)
{
    Hide();
    new Search().ShowDialog();
}

참조 1개
private void 개인정보수정_Click(object sender, EventArgs e)
{
    Hide();
    new MyInfoUpdate().ShowDialog();
}
참조 0개
private void 배송정보수정_Click(object sender, EventArgs e)
{
    Hide();
    new OrderRepair().ShowDialog();
}

참조 1개
private void pictureBox3_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("http://www.kb.or.kr");
}

참조 1개
private void pictureBox2_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("http://www.kb.or.kr");
}

참조 1개
private void pictureBox1_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("http://www.kb.or.kr");
}

조 1개
private void Map_Click(object sender, EventArgs e)
{
    Hide();
    new Map().ShowDialog();
}

조 1개
private void OrderR_Click(object sender, EventArgs e)
{
    Hide();
    new OrderRepair().ShowDialog();
}

조 1개
private void LogOut_Click(object sender, EventArgs e)
{
    Hide();
    new Login().ShowDialog();
}

조 1개
private void Chart_Click(object sender, EventArgs e)
{
    Hide();
    new Chart().ShowDialog();
}
```

- 각 버튼에 할당된 Form을 화면에 출력해준다.
- 마찬가지로 각 픽쳐박스에 할당된 주소값을 pc별 기본브라우저를 통해 열어준다.

3. 프로젝트 수행절차 타이틀 소스코드

```
using WMPLib; // Windows 미디어 플레이어 라이브러리
```

```
참조 1개
private void 재생_Click(object sender, EventArgs e)
{
    mp.URL = @"C:\Users\Administrator\Downloads\CSharp팀플\CSharp팀플\C#TeamProject\C#TeamProject\C#TeamProject\Resources\hello.mp3";
    mp.controls.play();
}

참조 1개
private void 음소거_Click(object sender, EventArgs e)
{
    mp.controls.stop();
}
```

- 배경음악 재생을 위해 windows.media 참조를 추가하고 new windowsmediaplayer를 mp로 만들어 사용하였다.

3. 프로젝트 수행절차 제품조회 Form

구매

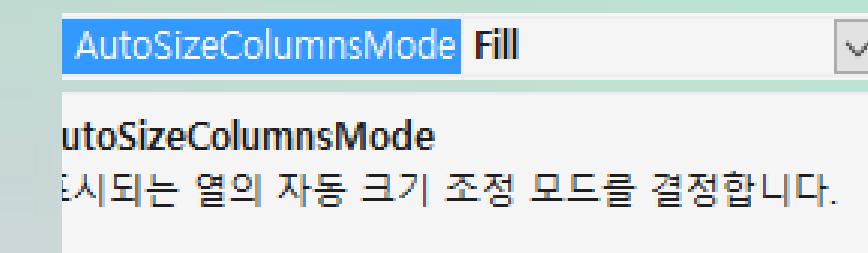
제품명 받는 이 주문번호 구매하기

가격 주소

보기

	Company	Name	Sick	Price
▶	(주)마더스제약	벤조날정(트리메부...)	식도역류 및 열공예...	2000
	(주)마더스제약	자이드캡슐(니푸록...	급성 세균성 설사	3000
	대우제약(주)	로파미드캡슐[로페...	급성설사, 만성설사	2000
	동화약품(주)	헬민200연질캡슐[...	간기능장애의 보조 ...	2000
	삼천당제약(주)	삼천당산화마그네...	위·심이지장궤양, ...	3000
	태극제약(주)	아비나파스타[트리...	만성 박리성 치은염...	4000
	동화약품(주)	활명수	식욕감퇴(식욕부진)...	2000
	신신제약(주)	신신티눈고[살리실...	티눈, 뜯(굳은살), ...	1000
	삼진제약(주)	아네모정	위산과다, 속쓰림, ...	3000
	동아제약(주)	타치온정50밀리그...	약물증독, 자가증독	4000
	(주)보령	겔포스현탁액[인산...	위·심이지장 궤양, ...	5000
	일약약품(주)	일약노이겔현탁액[...	위산과다 위·심이...	2000

아직까지 무서워?
나약은 더 무서워!



- DataGridView 조회 시 데이터 정보들이 나타나면 표 오른쪽에 공백이 생기는 것을 방지하기 위해 Fill 설정
- DataGridView에 약품 정보를 조회하여 사용자가 증상에 맞는 약품을 선택하고 개인정보를 입력해 구매할 수 있음

3. 프로젝트 수행절차 제품조회 소스코드

```
public partial class Search : Form
{
    private const string ConnectionMedicine = "Data Source=(local);Initial Catalog=Medicine;Integrated Security=True";
    private const string ConnectionPocket = "Data Source=(local);Initial Catalog=Customer;Integrated Security=True";
    참조 5개
    public Search()
    {
        InitializeComponent();
        try
        {
            using(SqlConnection con = new SqlConnection(ConnectionMedicine))
            {
                con.Open();
                string sql = "select * from [Medicine]";
                SqlCommand cmd = new SqlCommand(sql, con);

                SqlDataAdapter da = new SqlDataAdapter();
                da.SelectCommand = cmd;

                DataSet ds = new DataSet();

                DataTable dt = ds.Tables["Medicine"];
                da.Fill(ds, "t1");
                dataGridView1.DataSource = ds.Tables["t1"];
                con.Close();
            }
        }
        catch (SqlException x)
        {
            MessageBox.Show(x.Message);
        }
    }
}
```

```
{
    if (tbname.Text == "")
    {
        MessageBox.Show("이름을 입력하세요.");
    }
    else if (tbnum.Text == "")
    {
        MessageBox.Show("주문 번호를 할당하세요.");
    }
    else if (tbaddr.Text == "")
    {
        MessageBox.Show("주소를 입력하세요.");
    }
    else
    {
        string name = tbname.Text;
        string addr = tbaddr.Text;
        string mname = tbmname.Text;
        string price = tbprice.Text;
        string num = tbnum.Text;

        bool registrationSuccess = DBConnect.RegisterPocket(name, addr, mname, price, num);

        if (registrationSuccess)
        {
            MessageBox.Show("주문이 완료되었습니다.");
            ClearForm();
        }
        else
        {
            MessageBox.Show("주문에 실패했습니다.");
        }
    }
}
```

- connection@@를 생성해 DB에 연결하는 코드를 작성 그리드뷰에 출력하기 위한 코드

- 제품 조회 후 구매를 위한 예외처리와 DB에 넣어줄 코드
DBConnect의 메서드를 호출해 사용한다.

3. 프로젝트 수행절차 개인정보 조회 및 수정 Form

QMS

로그아웃

Title

Search

Map

Order repair

Chart

ID: **jy**

비밀번호: **●●**

조회

내정보 변경

비밀번호

이름

주소

변경하기

	Id	PassWord	Name	Addr
▶	jy	25	정윤	구미

UseSystemPasswordCha True

- 비밀번호 텍스트 박스 암호화



3. 프로젝트 수행절차 개인정보 조회 및 수정 Form

QMS

로그아웃

Title

Search

Map

Order repair

Chart

내정보 변경

ID: jy

비밀번호: ●●

조회

내정보 변경

비밀번호:

이름:

주소:

변경하기

	Id	PassWord	Name	Addr
▶	jy	25	정윤	구미

시작되기 전에 끊어내십시오

© 씨或多의약품안전처

- 조회 성공시 Form

QMS

로그아웃

Title

Search

Map

Order repair

Chart

내정보 변경

ID: jy

비밀번호: ●●

조회

내정보 변경

비밀번호:

이름:

주소:

변경하기

조회 실패.
아이디 혹은 비밀번호를 확인하세요.

확인

시작되기 전에 끊어내십시오

© 씨或多의약품안전처

- 조회 실패시 Form

3. 프로젝트 수행절차 개인정보 조회 및 수정 소스코드

```
void 조회_Click(object sender, Event  
ng id = textBox4.Text;  
ng password = textBox5.Text;  
  
loginSuccess = CheckId(id, password);  
  
loginSuccess)  
  
MessageBox.Show("조회 성공!");
```

- Login에서 사용했던 메서드를 재활용해 입력받은 id와 pw를 활용해 조회

```
try  
{  
    using (SqlConnection con = new SqlConnection(ConnectionString))  
    {  
        con.Open();  
        string sql = $"select * from [User] where id='{id}'";  
        SqlCommand cmd = new SqlCommand(sql, con);  
  
        SqlDataAdapter da = new SqlDataAdapter();  
        da.SelectCommand = cmd;  
  
        DataSet ds = new DataSet();  
  
        DataTable dt = ds.Tables["User"];  
        da.Fill(ds, "t1");  
        dataGridView1.DataSource = ds.Tables["t1"];  
        con.Close();  
    }  
}  
catch (SqlException x)  
{  
    MessageBox.Show(x.Message);  
}  
  
MessageBox.Show("조회 실패.\n아이디 혹은 비밀번호를 확인하세요.");
```

- 앞서 조회한 내용을 토대로 데이터그리드뷰에 띄워주는 코드

```
CheckId(string id, string password)  
  
{  
    SqlConnection connection = new SqlConnection(ConnectionString);  
    connection.Open();  
  
    string query = "SELECT COUNT(*) FROM [User] WHERE id = @id AND pass  
    word = @password";  
    SqlCommand command = new SqlCommand(query, connection);  
  
    command.Parameters.AddWithValue("@id", id);  
    command.Parameters.AddWithValue("@password", password);  
  
    int count = (int)command.ExecuteScalar();  
  
    return count > 0;  
  
}  
  
exception ex)  
{  
    Console.WriteLine($"Error: {ex.Message}");  
}
```

3. 프로젝트 수행절차 주문조회 및 수정 Form

The screenshot shows the QMS application interface. On the left, there is a sidebar with icons for Title, Search, MyInfo Update, Map, and Chart. The main area has two overlapping windows:

- 주문 조회 (Order Inquiry) Window:** Contains fields for '이름' (Name) with value '정윤' and '배송지' (Delivery Address) with value '구미'. A red box highlights the '조회' (Inquiry) button.
- 주문 수정 (Order Modification) Window:** Contains fields for '이름' (Name), '배송지' (Delivery Address), '제품' (Product), '가격' (Price), and '주문번호' (Order Number). A red box highlights the '수정' (Modify) button.

Below these windows is a table with columns: Name, Addr, Mname, and Price. One row is selected, showing '정윤' in the Name column and '구미' in the Addr column. The table also contains a header row with column names: Name, Addr, Mname, and Price.

	Name	Addr	Mname	Price
▶	정윤	구미	벤즈날정(트리메부틴말...	2000
*				

At the bottom of the application window, there is a dark banner with the text "시작하는 순간
중독도 시작됩니다" (The moment it starts,
addiction begins).

- DB에 저장된 고객의 이름과 주소 정보를 불러들여 조회 해주고 예외처리 실패시 에러메세지를 출력함

3. 프로젝트 수행절차 주문조회 및 수정 Form

QMS

로그아웃

주문 조회

주문 수정

이름 정윤

배송지 구미

조회

수정

Title

Search

MyInfo Update

Map

Chart

주문 조회

주문 수정

이름 정윤

배송지 대구

제품

가격

주문번호

조회

수정

Name	Addr	Mname	Price
▶ 정윤	구미	벤즈남정(트리메부틴말...	2000
*			

시작하는 순간
중독도 시작됩니다

The screenshot shows the QMS application interface. On the left, there's a sidebar with icons for Title, Search, MyInfo Update, Map, and Chart. The main area has two tabs: '주문 조회' (Order Inquiry) and '주문 수정' (Order Modification). In the '주문 조회' tab, there are input fields for '이름' (Name) and '배송지' (Delivery Address), both containing the value '정윤'. A red box highlights the '조회' (Search) button. Below this is a table with columns 'Name', 'Addr', 'Mname', and 'Price'. A row is selected, showing '▶ 정윤' in the Name column, '구미' in the Addr column, '벤즈남정(트리메부틴말...' in the Mname column, and '2000' in the Price column. A red box highlights the '수정' (Modify) button. At the bottom, there's a banner with the text '시작하는 순간 중독도 시작됩니다'.

- 주문 조회 성공시 Form

QMS

로그아웃

주문 조회

주문 수정

이름 정윤

배송지 대구

제품

가격

주문번호

조회

수정

Title

Search

MyInfo Update

Map

Chart

조회 실패.
아이디 혹은 비밀번호를 확인하세요.

확인

시작하는 순간
중독도 시작됩니다

The screenshot shows the QMS application interface. It's similar to the first one but with an error message. A modal window is open with the text '조회 실패. 아이디 혹은 비밀번호를 확인하세요.' (Search failed. Please check your ID or password.) and a blue '확인' (Confirm) button. The rest of the interface is identical to the first screenshot, including the sidebar and the banner at the bottom.

- 주문 조회 성공시 Form

3. 프로젝트 수행절차 주문조회 및 수정 소스코드

```
try
{
    using (SqlConnection con = new SqlConnection(ConnectionString))
    {
        con.Open();
        string sql = $"select * from [Customer] where name='{name}' AND Addr='{addr}'";
        SqlCommand cmd = new SqlCommand(sql, con);

        SqlDataAdapter da = new SqlDataAdapter();
        da.SelectCommand = cmd;

        DataSet ds = new DataSet();

        DataTable dt = ds.Tables["Customer"];
        da.Fill(ds, "t1");
        dataGridView1.DataSource = ds.Tables["t1"];
        con.Close();
    }
    catch (SqlException x)
    {
        MessageBox.Show(x.Message);
    }
}
else
{
    MessageBox.Show("조회 실패.\n아이디 혹은 비밀번호를 확인하세요.");
}
```

```
private bool CheckOrder(string name, string addr)
{
    try
    {
        using (SqlConnection connection = new SqlConnection(ConnectionString))
        {
            connection.Open();

            string query = "SELECT COUNT(*) FROM [Customer] WHERE name = @name AND addr = @addr";
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                command.Parameters.AddWithValue("@name", name);
                command.Parameters.AddWithValue("@addr", addr);

                int count = (int)command.ExecuteScalar();

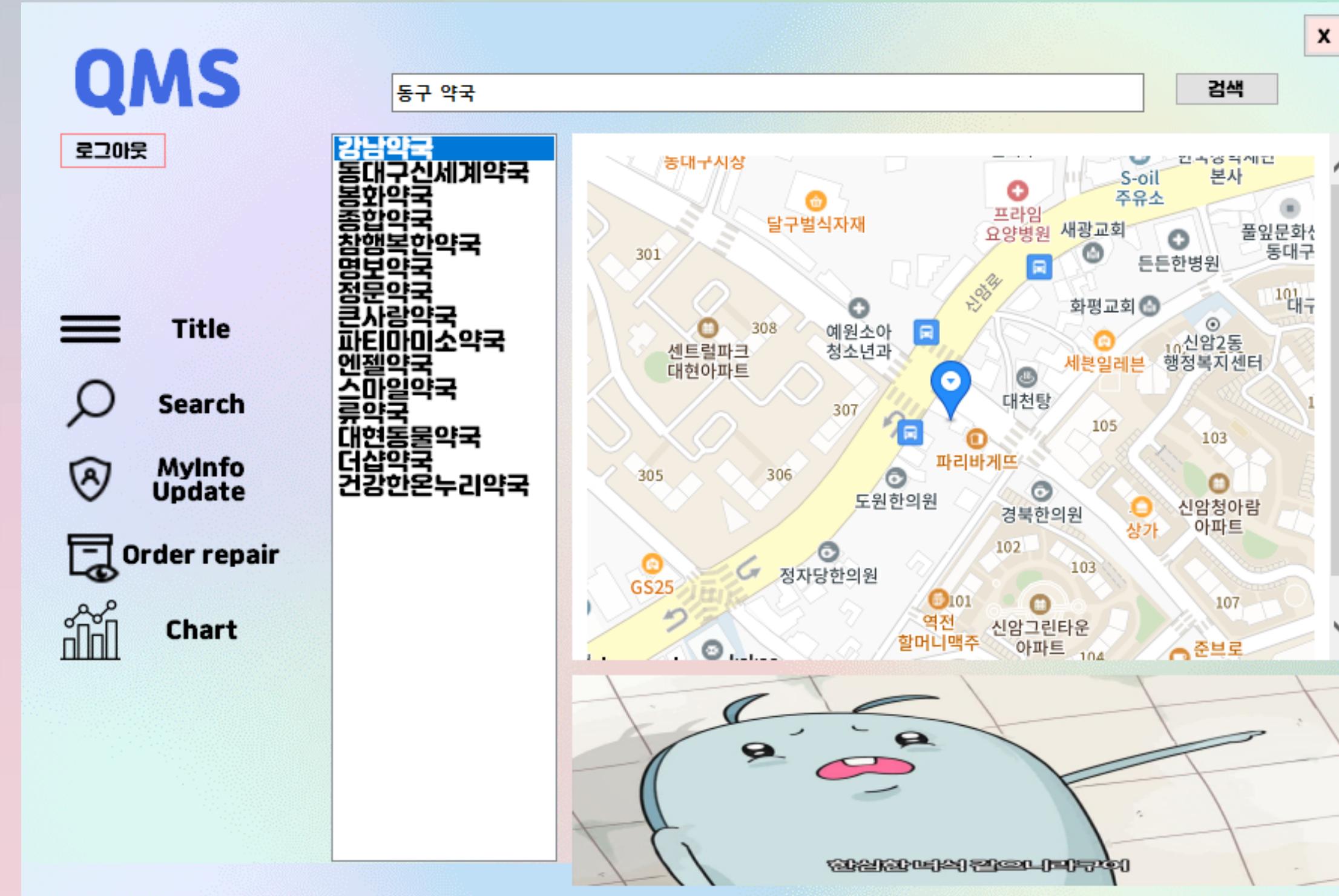
                return count > 0;
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
        return false;
    }
}
```

- 조회 창에서 입력받은 이름, 주소를 통해 customer DB에서 동일한 내용이 있는지 여부 체크

- 변경 창에서 입력받은 이름, 주소를 통해 customer DB의 내용 수정

3. 프로젝트 수행절차

MAP Form



- 카카오 API를 활용하여 MAP Form 구현

3. 프로젝트 수행절차 MAP 소스코드

```
public class Locale
{
    참조 2개
    public string Name { get; set; }
    참조 2개
    public double Lat { get; set; }
    참조 2개
    public double Lng { get; set; }

    참조 1개
    public Locale(string name, double lat, double lng)
    {
        Name = name;
        Lat = lat;
        Lng = lng;
    }

    참조 0개
    public override string ToString()
    {
        return Name;
    }
}
```

```
private void LocationSearch_Click(object sender, EventArgs e)
{
    List<Locale> locales = KaKaoAPI.Search(textBox1.Text);
    listBox1.Items.Clear();
    foreach (Locale item in locales)
        listBox1.Items.Add(item);
}

참조 1개
private void textBox1_KeyUp(object sender, KeyEventArgs e)
{
    if(e.KeyCode == Keys.Enter)
        LocationSearch.PerformClick();
}

참조 1개
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listBox1.SelectedIndex == -1)
        return;
    Locale ml = listBox1.SelectedItem as Locale;
    object[] pos = new object[] { ml.Lat, ml.Lng };
    HtmlDocument hdoc = webBrowser1.Document;
    hdoc.InvokeScript("setCenter", pos);
}

참조 1개
private void Logout_Click(object sender, EventArgs e)
{
    ClearForm();
    Hide();
    now.Login();
}
```

- Map Form에서 조회될 약국명과 위치 좌표가 선언된 클래스.

- 해당 지역 약국을 입력하면 응답한 값들이 나오고 Enter키를 누르면 클릭이 강제호출됨.

3. 프로젝트 수행절차

MAP 소스코드

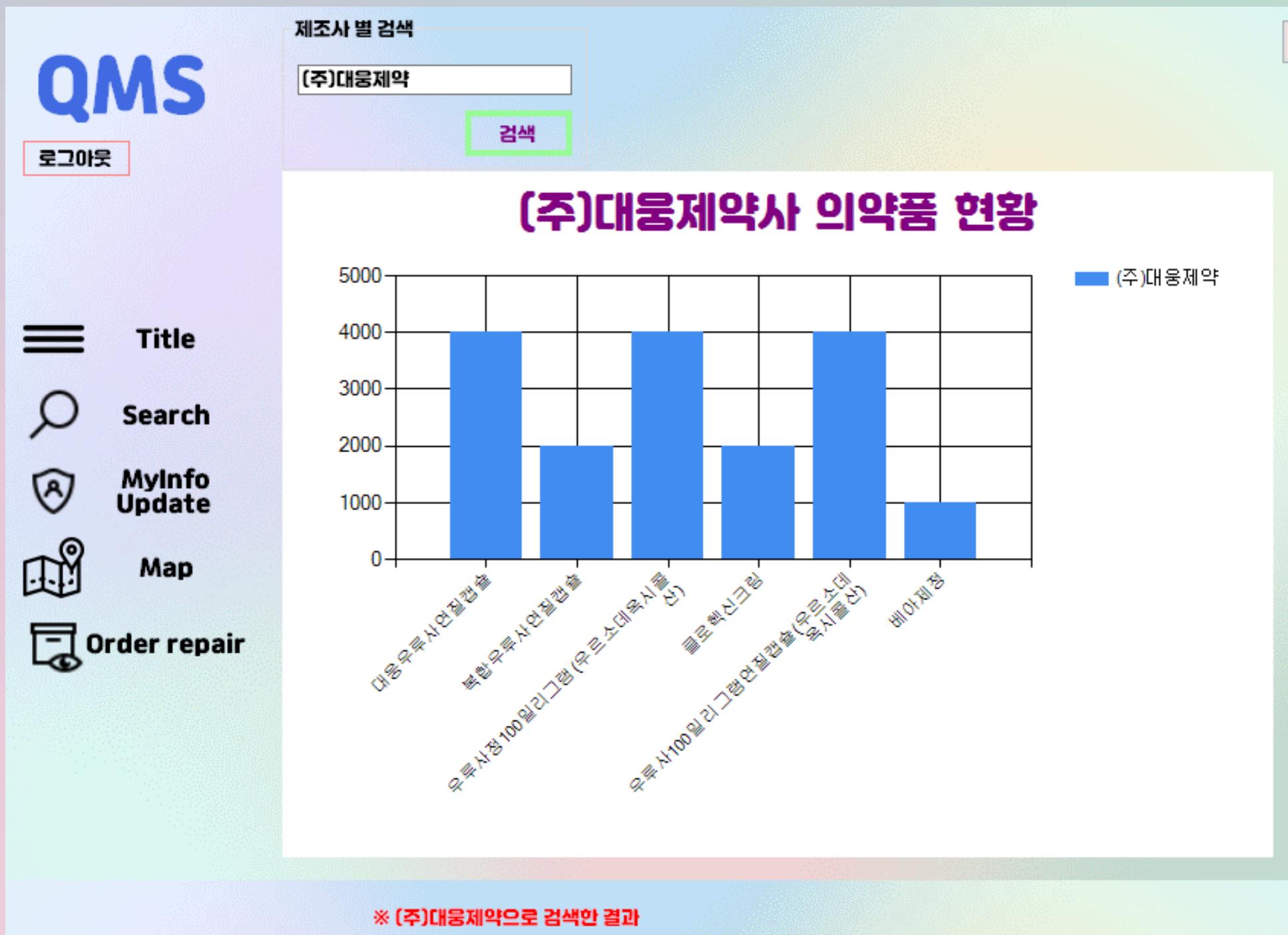
```
public class KaKaoAPI
{
    참조 1개
    public static List<Locale> Search(string text)
    {
        List<Locale> list = new List<Locale>();
        string url = "https://dapi.kakao.com/v2/local/search/keyword.json";
        string query = $"{url}?query={text}";
        string restAPIKey = "2f23516a0b28443cf35d2893103b6f7";
        string Header = $"KakaoAK {restAPIKey}";
        WebRequest request = WebRequest.Create(query);
        request.Headers.Add("Authorization", Header);

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

- 카카오 맵 API 주소를 URL로 호출하고
요청을 API에 보내 응답받아 지도를 출력함

3. 프로젝트 수행절차

Chart Form



- 제약 회사 별 검색을 통해
제품 별 가격으로 차트를 표현

3. 프로젝트 수행절차

Chart 소스코드

```
private void Chart_Load(object sender, EventArgs e)
{
    using (SqlConnection conn = new SqlConnection(ConnectionMedicine))
    {
        conn.Open();

        string query = "SELECT Company, COUNT(*) AS CompanyCount From Medicine GROUP BY Company";
        SqlDataAdapter da = new SqlDataAdapter(query, conn);
        DataSet ds = new DataSet();
        da.Fill(ds, "Medicine");

        Chart chart1 = new Chart();
        chart1.Size = new System.Drawing.Size(650, 450);
        chart1.Titles.Add("의약품 종류 현황");
        chart1.Titles[0].Font = new System.Drawing.Font("여기어때 잘난체 OTF", 20);
        chart1.Titles[0].ForeColor = System.Drawing.Color.Purple;
        chart1.Series.Clear();
        chart1.Series.Add("제약사");
        chart1.Series.Add("제품 수");
        chart1.Series["제약사"].XValueMember = "Company";
        chart1.Series["제약사"].YValueMembers = "CompanyCount";
        chart1.DataSource = ds.Tables["Medicine"];

        chart1.ChartAreas[0].AxisX.LabelStyle.Angle = -45;

        this.Controls.Add(chart1);
    }
}
```

```
private void CompanySearch_Click(object sender, EventArgs e)
{
    string al = textBox1.Text;
    알림.Text = $"※ {al}으로 검색한 결과";
    using (SqlConnection conn = new SqlConnection(ConnectionMedicine))
    {
        conn.Open();

        string query = $"SELECT * From Medicine where Company = @Company";
        SqlDataAdapter da = new SqlDataAdapter(query, conn);
        da.SelectCommand.Parameters.AddWithValue("@Company", al);

        DataSet ds = new DataSet();
        da.Fill(ds, "Medicine");

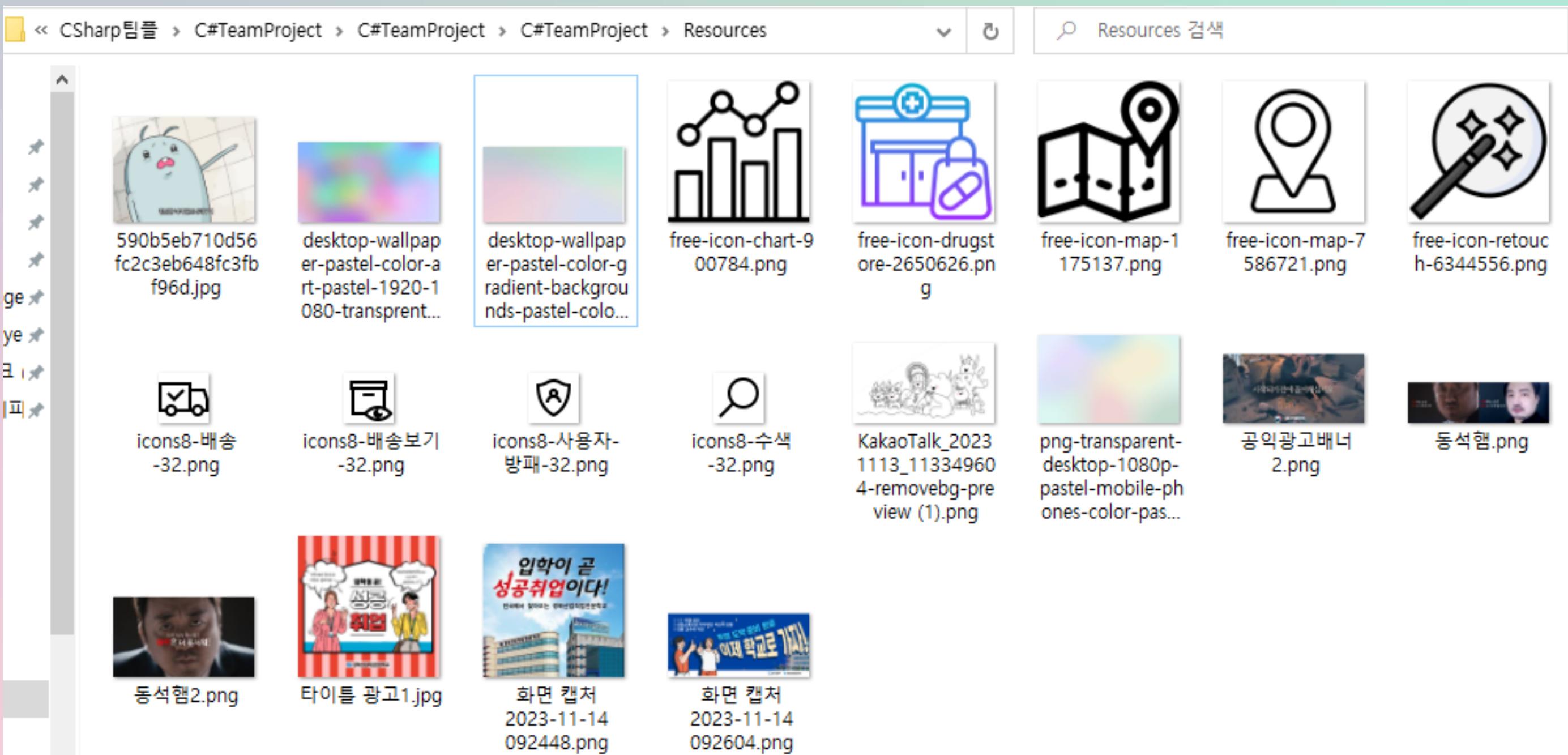
        chart1.Series.Clear();
        chart1.Titles.Clear();
        chart1.Size = new System.Drawing.Size(650, 450);
        chart1.Titles.Add($"{al}사 의약품 현황");
        chart1.Titles[0].Font = new System.Drawing.Font("여기어때 잘난체 OTF", 20);
        chart1.Titles[0].ForeColor = System.Drawing.Color.Purple;
        chart1.Series.Add($"{al}");
        chart1.Series[$"{al}"].XValueMember = "Name";
        chart1.Series[$"{al}"].YValueMembers = "Price";
        chart1.DataSource = ds.Tables["Medicine"];

        chart1.ChartAreas[0].AxisX.LabelStyle.Angle = -45;

        this.Controls.Add(chart1);
    }
}
```

- CHART1에 대한 타이틀, 시리즈, 시리즈에 대한 X,Y값을 지정 후 차트에 표기해준다.
- LABEL앵글을 -45도를 줘 대각선으로 텍스트를 표기하여 후에 데이터가지가 많아졌을 때, 서로 꼬이는 일이 생기지 않도록 예방해주었다.
- 입력받은 검색어(회사이름)을 통해 DB에 같은 회사명의 제품을 셈해주고 가격별로 차트에 나열해준다.

3. 프로젝트 수행절차 리소스 파일



- Form에 사용한 이미지를 리소스 파일에 보관하여 여러 컴퓨터에서도 이미지를 옮기지 않아도 오류없이 Form 실행 가능

04

프로젝트 결과

4. 프로젝트 결과

