

1차 프로젝트

MySQL을 활용한 MVC1패턴 숙박예약사이트 구현

21.01.27~21.02.25

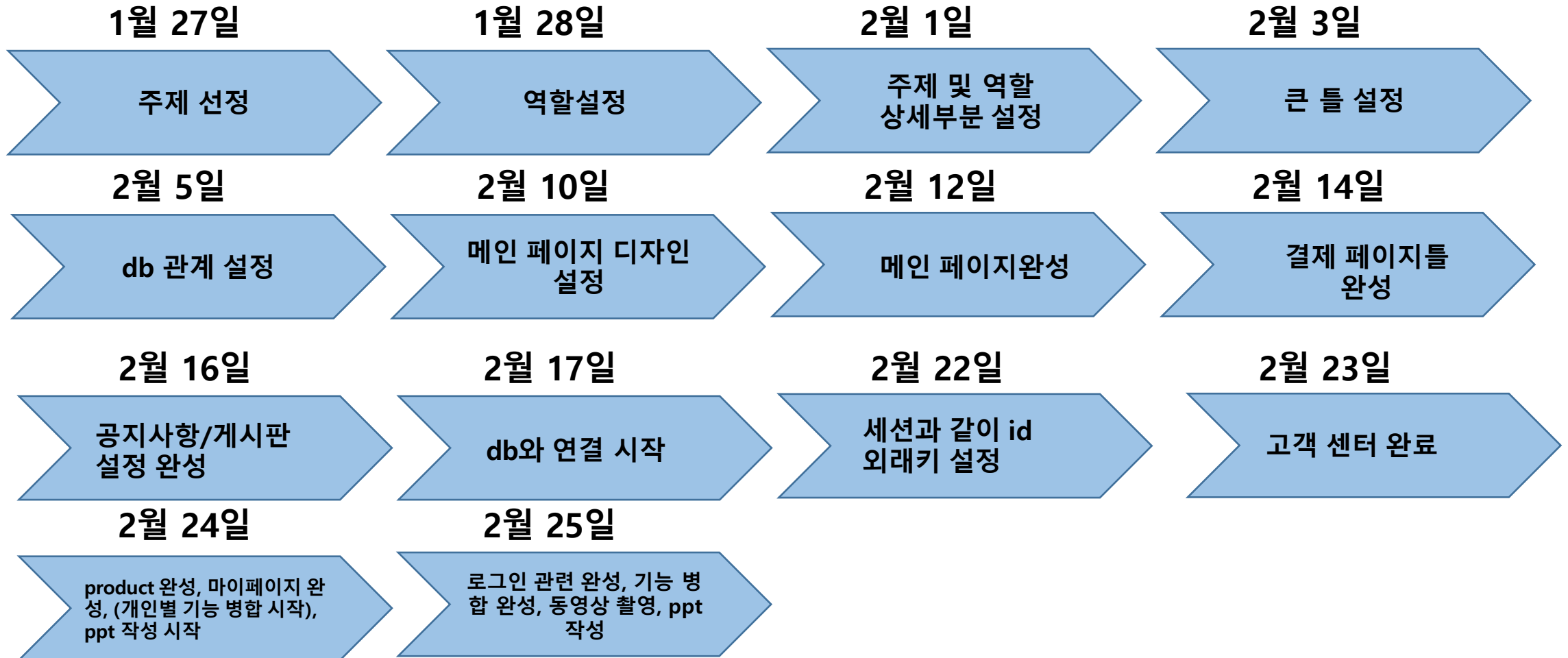
회원가입
로그인
회원정보관리

원예진

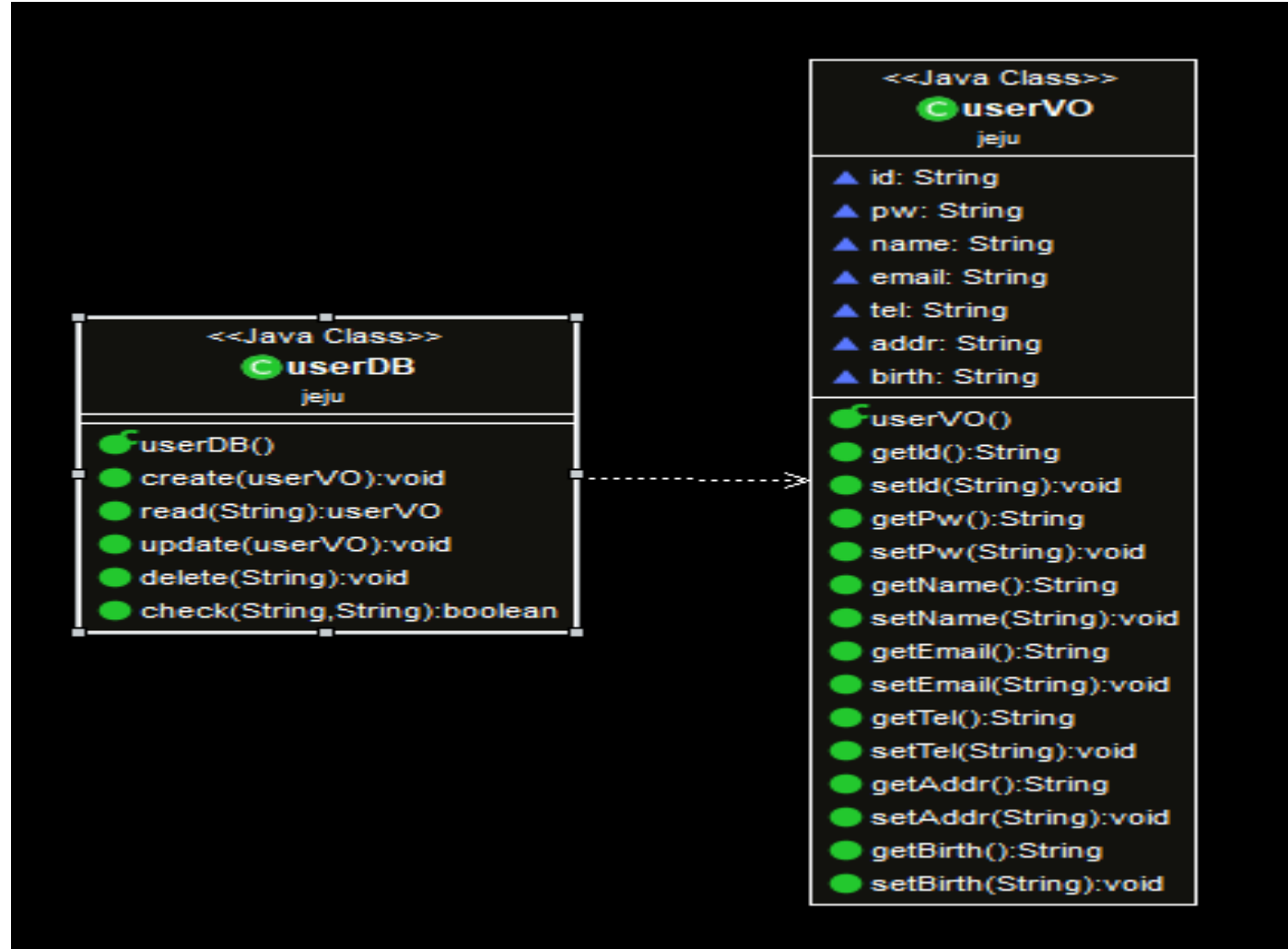
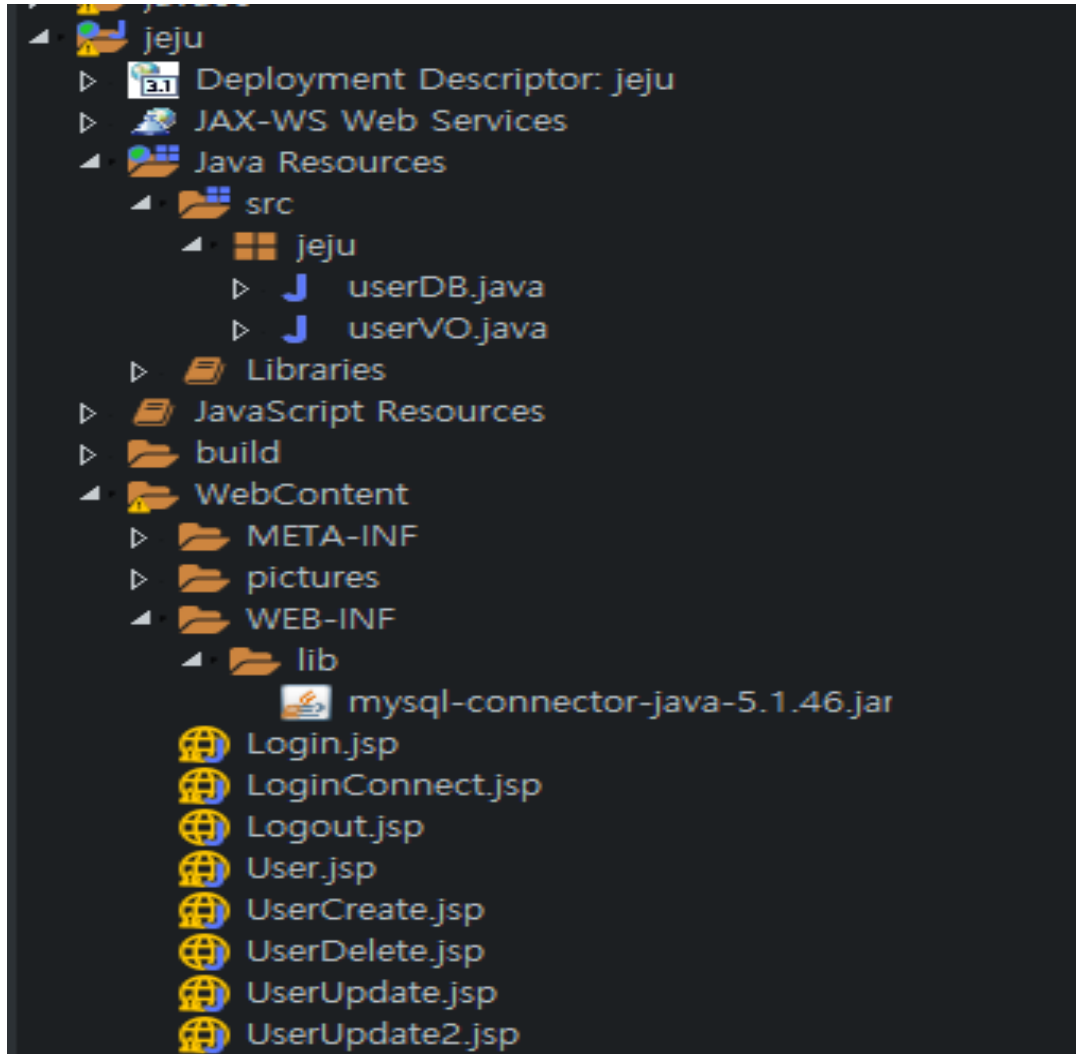
I. 개요



I. 개발과정 > 개발기간



III. 메뉴별 소개 : 프로젝트트리, UML



I. 개요

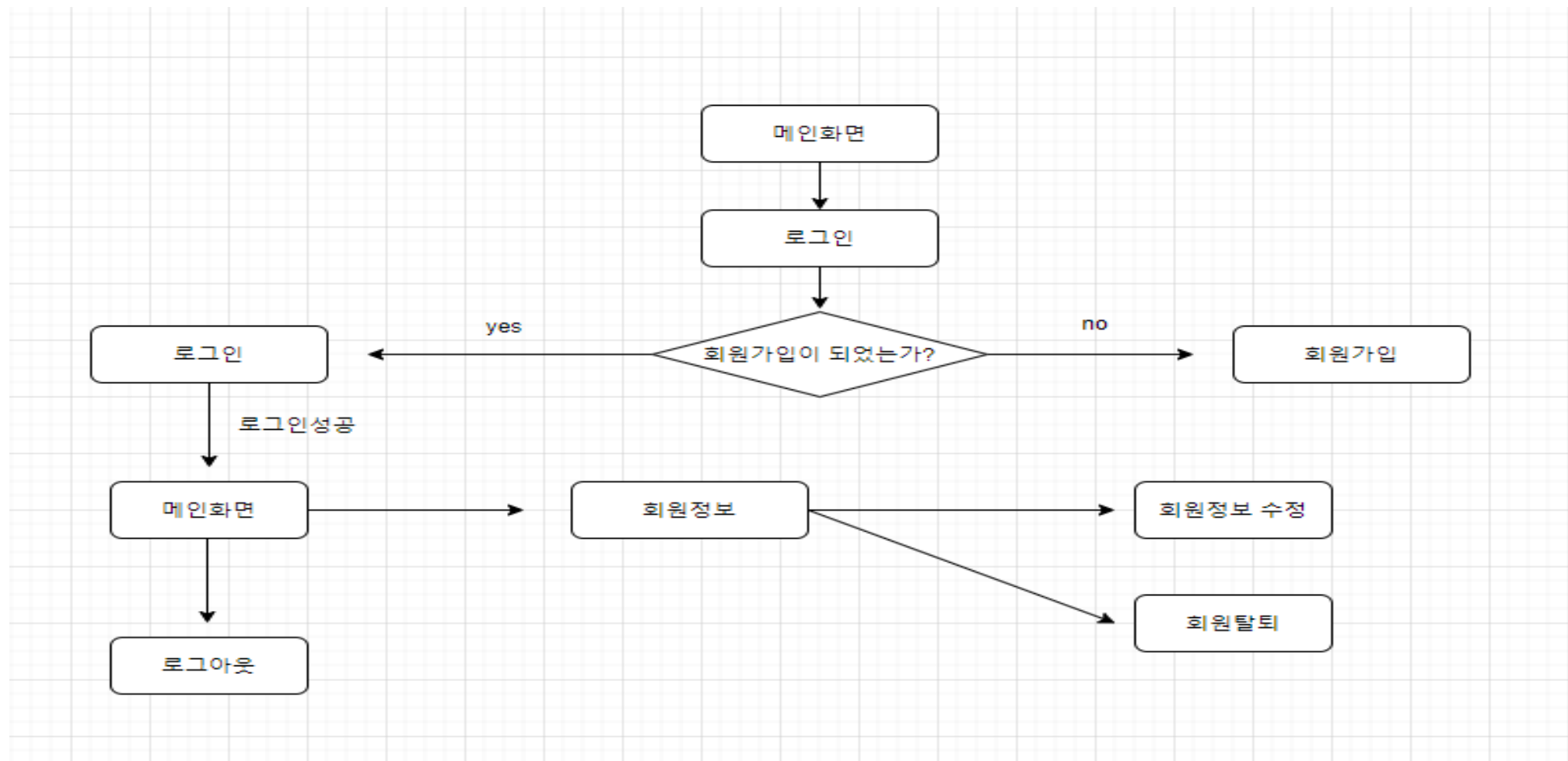
개발 과정 >> 제약조건 : user_information

컬럼 명	Data Type	제약 조건	설명
id	Varchar(100)	Not Null Primary Key	아이디
pw	Varchar(50)	Not Null	패스워드
name	Varchar(20)	Not Null	이름
tel	Varchar(20)	Not Null	전화번호
email	Varchar(200)	Not Null	이메일
addr	Varchar(30)	Not Null	주소
birth	Varchar(15)	Not Null	생년월일

II. 시스템 구성도



III. 메뉴별 소개 : 순서도



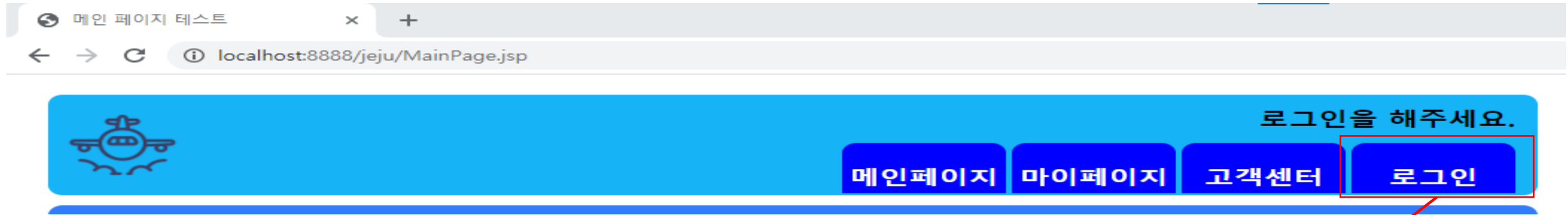
구현



21.02.14~ 21.02.25

회원가입 | 로그인 | 회원정보관리

III. 메뉴별 소개 : 로그인



메인화면의 로그인 메뉴
로그인이 완료되지 않아 잡혀있는 id 세션이
없을 때 보이는 “로그인”

```
<% if( id == null) {    %>
<!-- 세션을 이용해 로그인이 되지 않았을때 로그인을 하기위해서 로그인페이지로 연결되는 데이터셀을 생성 -->
    <a class="menuLink" href="Login.jsp">로그인</a>
<%} else {%>
    <!-- 로그인이 성공 했을경우 로그아웃을 할 수 있게하는 로그아웃으로 연결되는 셀을 생성 -->
    <a class="menuLink" href="UserUpdate.jsp">회원 정보</a>
<%} %>
</td>
```

III. 메뉴별 소개 : 로그인

Form으로 입력값들을 받아주는 input창들과 submit해주는 버튼을 묶어준다.

```
<form action = "LoginConnect.jsp"> <!-- 입력값 LoginConnect로 전송 -->

    <button>LOGIN</button><br> <!-- 버튼 타입 기본 submit -->
    <br>
    <br>
    <br>
</form>
```

두 개의 input창의 입력값을 데이터를 처리해주는 LoginConnect.jsp로 넘겨준다.

```
<br>
<font size = "5em" color = "black" >
<input name = "id" placeholder = "아이디" type="text" required style="text-align:center;width:250px; height:30px; letter-spacing: 1px"><br>
<!--아이디인풋창 => placeholder(인풋창에 글씨써놓고 입력값이 생기면 글씨 사라지는 것), required(입력값이 없으면 전송이 안됨. 밑에 경고창띄우기), style(순서대로/중앙정렬/ input창 너비/높이/글자간 간격) -->
<input name = "pw" type = "password" placeholder = "비밀번호" type="text" required style="text-align:center; width:250px; height:30px; letter-spacing: 1px"><br>
<!-- 패스워드 인풋창 => type을 패스워드로 설정해 입력시 어떤 글씨를 치고있는지 보이지 않게함 -->
</font>
```

III. 메뉴별 소개 : 로그인

로그인 입력값 처리해주는 LoginConnect.jsp

```
4
5 <%
6
7     String id = request.getParameter("id"); //Login.jsp에서 입력된 id 변수값으로 넣기
8     String pw = request.getParameter("pw"); //Login.jsp에서 입력된 pw 변수값으로 넣기
9
10    userDB db = new userDB(); //데이터베이스 처리할 도구 가져오기
11    boolean result = db.check(id, pw); //check 메서드 사용해서 id,pw가 일치한지 확인
12
13    if(result){ //id,pw 일치하여 result 값이 true일때
14        session.setAttribute("id", id); //로그인 성공시 세션 생성
15        response.sendRedirect("MainPage.jsp"); //로그인완료시 메인페이지로 보내줌
16    }
17
18
19 %>
```

입력된 데이터를 매개변수로 받아 dao객체를 만들고 dao안의
Check메서드를 실행시켜 id,pw가 일치한지 확인

III. 메뉴별 소개 : 로그인

UserDB.java(DAO)의 check메서드

```
public boolean check(String id, String pw) throws Exception { //id,pw 입력값을 가져와 check 메서드 실행. 반환값은 boolean으로 결과값이 true나 false.
    Class.forName("com.mysql.jdbc.Driver");
    System.out.println("1. 커넥터 사용 설정 성공. <br>");

    String url = "jdbc:mysql://localhost:3306/jeju?useUnicode=true&characterEncoding=utf8";

    Connection con = DriverManager.getConnection(url, "root", "1234");
    System.out.println("2. db연결성공. <br>");

    String sql = "select * from user_information where id = ? and pw = ?"; //입력된 데이터의 id 와 pw가 일치하는것을 select
    PreparedStatement ps = con.prepareStatement(sql); // sql문 인식
    ps.setString(1, id);
    ps.setString(2, pw);
    System.out.println("3. sql문을 만들기 성공. <br>");

    ResultSet rs = ps.executeQuery();
    System.out.println("4. SQL문을 MySQL서버로 전송 성공. <br>");

    boolean result = false; // result라는 boolean변수 생성 및 값 초기화.

    if (rs.next()) { // rs.next()로 찾는 데이터값들이 있으면
        result = true; //result값을 true로 입력
    }
    return result; //result 반환
} // check
```

입력된 데이터가 일치하는지 확인하는 sql문 생성.
Select사용.

Select의 결과 유무를 보여주는 result변수로 반환

boolean 으로 결과값 반환한다.

III. 메뉴별 소개 : 로그인

```

4
5<%
6
7    String id = request.getParameter("id");    //Login.jsp에서 입력된 id 변수값으로 넣기
8    String pw = request.getParameter("pw");    //Login.jsp에서 입력된 pw 변수값으로 넣기
9
10   userDB db = new userDB();    //데이터베이스 처리할 도구 가져오기
11   boolean result = db.check(id, pw);    //check 메서드 사용해서 id,pw가 일치한지 확인
12
13   if(result){    //id,pw 일치하여 result 값이 true일때
14       session.setAttribute("id", id);    //로그인 성공시 세션 생성
15       response.sendRedirect("MainPage.jsp");    //로그인완료시 메인페이지로 보내줌
16   }
17
18
19 %>

```

로그인 성공시 입력된 id로 세션을 생성

로그인 완료 후 메인페이지로 보내줌

상단에 로그인 정보가 입력되고, 로그아웃 버튼과 회원정보 메뉴가 생긴 것을 확인할 수 있다.



III. 메뉴별 소개 : 회원가입

LOGIN

User.jsp 회원가입창으로
연결해주는 버튼

회원이 아니세요?

[메인페이지](#)
[마이페이지](#)
[고객센터](#)
[로그인](#)

회원가입

*전체 항목 모두 필수 항목입니다.

아이디

패스워드

이름

핸드폰번호

E-mail

주소

생년월일

```

<form action = "User.jsp"> <!-- 회원가입창 연결을 위한 form -->
회원가입?
<button>회원가입</button> <!-- 회원가입 버튼 -->
<br>
</form> <!-- 회원가입 form닫기 -->
    
```

Input 입력값 총 7개 받기
Id, pw, name, tel, email, addr, birth
하나의 폼 안에 입력창들과 값 전송을 위한
버튼을 묶어둔다.

III. 메뉴별 소개 : 회원가입

```
<form action = "UserCreate.jsp" > <!-- 하단 입력데이터를 UserCreate.jsp 로 전송해주기 위한 form -->
<b><font size = "4em" color = "black" > <!-- font 설정 시작 -->
<table align="center"> <!-- 회원가입 테이블 생성 / 테이블 위치 가운데 정렬 -->
  *전체 항목 모두 필수 항목입니다. <br> <!-- 전체항목을 입력하지않으면 회원가입창이 넘어가지 않을 것임을 보여줌 -->
```

```
<tr>
<br> <!-- 입력창과 알림문구 사이의 공백을 띄우기 위함 -->
</tr>
<tr>
```

전체 항목 필수항목으로 required 속성값 추가
Lowercase -> 전체문자 소문자로 변환시켜주는 것

```
<td>
아이디
</td>
<td>
```

```
<input name = "id" type="text" required style="text-align:center; width:200px; height:30px; letter-spacing: 1px; text-transform: lowercase"> <br>
<!-- id 인풋창. required 필수항목, style 가운데정렬, input창의 너비, 높이, 글자간 간격, 입력된 글자를 소문자로 자동변환 해주는 것 -->
<!-- 회원가입시 소문자, 대문자 차이로 데이터 처리할때 오류가 생기는것을 없게하기 위함 -->
</td>
```

```
</tr>
<tr>
<td>
패스워드
</td>
```

```
<td>
<input name = "pw" type="text" required style="text-align:center; width:200px; height:30px; letter-spacing: 1px; text-transform: lowercase"><br>
<!-- pw 인풋창. required 필수항목, style 가운데정렬, input창의 너비, 높이, 글자간 간격, 입력된 글자를 소문자로 자동변환 해주는 것 -->
<!-- 로그인 시 패스워드 입력창에 글씨가 안보이게 설정해두었음 로그인 오류를 줄이기 위해 소문자로 자동변환 -->
```

```
</td>
</tr>
```

III. 메뉴별 소개 : 회원가입

```
<td>
    핸드폰번호
</td>
<td>
    <input name = "tel" type="text" maxlength = "11" required style="text-align:center; width:200px; height:30px; letter-spacing: 1px"><br>
    <!-- maxlength = 글씨수 제한을 두는것. 핸드폰번호는 11자리로 11개의 숫자만 입력가능 -->
</td>
</tr>
```

```
<td>
    생년월일
</td>
<td>
    <input name = "birth" type="text" maxlength = "8" required style="text-align:center; width:200px; height:30px; letter-spacing: 1px"><br>
    <!-- maxlength = 생년월일 8글자만 입력가능 -->
</td>
</tr>
```

글씨수 제한 maxlength
핸드폰번호 : 11자리
생년월일 : 8자리

이 밖의 데이터들은 required 속성만 추가

III. 메뉴별 소개 : 회원가입

회원가입

***전체 항목 모두 필수 항목입니다.**

아이디	won
패스워드	123
이름	yejin
핸드폰번호	01012345678
E-mail	dfsdf@naver.com
주소	서울특별시
생년월일	19970402

데이터 입력

회원가입



```
<%
/* 입력값 변수에 넣기 */
String id = request.getParameter("id");
String pw = request.getParameter("pw");
String name = request.getParameter("name");
String email = request.getParameter("email");
String tel = request.getParameter("tel");
String addr = request.getParameter("addr");
String birth = request.getParameter("birth");

/* 데이터 처리해줄 틀 만들기 */
userDB db = new userDB();

/* 데이터를 옮길 가방 만들기 */
userVO bag = new userVO();

/* 가방에 데이터 넣기 */
bag.setId(id);
bag.setPw(pw);
bag.setName(name);
bag.setTel(tel);
bag.setEmail(email);
bag.setAddr(addr);
bag.setBirth(birth);

/* 데이터 넣은 가방을 create메서드로 데이터 처리하기 */
db.create(bag);
%
```

데이터 처리

VO생성 후
VO에 데이터를 넣어
create
메서드 실행

user_information							
	id	pw	name	tel	email	addr	birth
1	won	123	yejin	01012345678	dfsdf@naver.com	서울특별시	19970402



DB 데이터 insert

III. 메뉴별 소개 : 회원가입

Create 메서드 (UserDb.java)

```

9 public class userDB {
10
11     public void create(userVO bag) throws Exception { //입력받은 데이터를 가방에 넣어서 주기때문에 userVO bag이 입력값
12
13         Class.forName("com.mysql.jdbc.Driver");
14         System.out.println("1. 커넥터 사용 설정 성공.");
15
16         String url = "jdbc:mysql://localhost:3306/jeju?useUnicode=true&characterEncoding=utf8";
17
18         Connection con = DriverManager.getConnection(url, "root", "1234");
19         System.out.println("2. db연결성공.");
20
21         String sql = "insert into user_information values(?,?,?,?,?,?,?)"; //user_information 테이블에 7개의 값들을 insert
22         PreparedStatement ps = con.prepareStatement(sql); // sql문 안착
23
24         ps.setString(1, bag.getId()); //user_information에 설정되어있는 칼럼순서대로 insert //가방에서 id를 꺼내 꺼낸값을 value로 insert시킴
25         ps.setString(2, bag.getPw()); //가방에서 pw를 꺼내 꺼낸값을 value로 insert시킴
26         ps.setString(3, bag.getName()); //가방에서 name를 꺼내 꺼낸값을 value로 insert시킴
27         ps.setString(4, bag.getTel()); //가방에서 tel를 꺼내 꺼낸값을 value로 insert시킴
28         ps.setString(5, bag.getEmail()); //가방에서 email를 꺼내 꺼낸값을 value로 insert시킴
29         ps.setString(6, bag.getAddr()); //가방에서 addr를 꺼내 꺼낸값을 value로 insert시킴
30         ps.setString(7, bag.getBirth()); //가방에서 birth를 꺼내 꺼낸값을 value로 insert시킴
31         System.out.println("3. sql문을 만들기 성공.");
32
33         ps.executeUpdate();
34         System.out.println("4. SQL문을 mySQL서버로 전송 성공.");
35
36     } // create
37

```

VO에서 데이터를 꺼내 sql문에 적용

III. 메뉴별 소개 : 회원정보 수정

won님 환영합니다.로그아웃

메인페이지 마이페이지 고객센터 **회원 정보**

회원정보 수정 및 탈퇴

*전체 항목 모두 필수 항목입니다.

아이디	won
패스워드	
이름	yejin
핸드폰번호	01012345678
E-mail	dfsdf@naver.com
주소	서울특별시
생년월일	19970402

회원정보 수정 회원탈퇴

로그인이 완료되어
잡혀있는 id 세션이 있을 때
보이는
“회원정보”메뉴

```
<% if( id == null) {    %>
<!-- 세션을 이용해 로그인이 되지 않았을때 로그인을 하기위해서 로그인페이지로 연결되는 데이터셀을 생성 -->
    <a class="menuLink" href="Login.jsp">로그인</a>
<%} else {%>
<!-- 로그인이 성공 했을경우 로그아웃을 할 수 있게하는 로그아웃으로 연결되는 셀을 생성 -->
    <a class="menuLink" href="UserUpdate.jsp">회원 정보</a>
<%} %>
</td>
```

III. 메뉴별 소개 : 회원정보 수정

회원정보 수정 및 탈퇴

*전체 항목 모두 필수 항목입니다.

아이디	won
패스워드	
이름	yejin
핸드폰번호	01012345678
E-mail	dfsdf@naver.com
주소	서울특별시
생년월일	19970402
회원정보 수정	회원탈퇴

```
<%
userDB db = new userDB(); /* 데이터처리해줄 틀 생성 */
String id = (String)session.getAttribute("id"); /* id세션값을 변수에 넣어주기 */
userVO bag = db.read(id); /* read메서드 실행 */

%>
```

회원정보 수정시 value로 기존 데이터를
입력받아두기 위해 read 메서드 실행

세션으로 잡힌 id를 변수로 넣고 변수를 read
메서드로 실행

III. 메뉴별 소개 : 회원정보 수정

```
public userVO read(String id) throws Exception { //입력값을 id로 해서 read 메서드 실행시킬 것. 반환값은 userVO로 가방에 넣어 반환.

    Class.forName("com.mysql.jdbc.Driver");
    System.out.println("1. 커넥터 사용 설정 성공. <br>");

    String url = "jdbc:mysql://localhost:3306/jeju?useUnicode=true&characterEncoding=utf8";
    Connection con = DriverManager.getConnection(url, "root", "1234");
    System.out.println("2. db연결 성공. <br>");

    String sql = "select * from user_information where id = ?"; //id가 ?인 데이터를 user_information 테이블에서 검색
    PreparedStatement ps = con.prepareStatement(sql);
    // sql인식하게 만들어주는 것 (java sql)
    ps.setString(1, id); //id는 입력값으로 ?자리에 입력
    System.out.println("3. sql문을 만들기 성공. <br>");

    ResultSet rs = ps.executeQuery();
    //read는 executeupdate 아니고 query임
    System.out.println("4. SQL문을 mySQL서버로 전송 성공. <br>");

    // 1. 검색결과가 있는지 체크해야한다.
    //System.out.println(rs.next()); //true = 있다, false = 없다.

    // 2. 만약에 결과가 있으면, 테이블에서 원하는 데이터 항목을 꺼내온다.
```

String id를 넣어
read메서드를
실행시키고
반환값은 userVO

검색한 데이터가
존재하는지 확인하는
ResultSet 생성

아이디를 조건으로 하는 user_information 테이블의 데이터를 검색(select)

III. 메뉴별 소개 : 회원정보 수정

```
// 1. 검색결과가 있는지 체크해야한다.
//System.out.println(rs.next()); //true = 있다, false = 없다.

// 2. 만약에 결과가 있으면, 테이블에서 원하는 데이터 항목을 꺼내온다.

userVO bag = new userVO(); //데이터를 담을 가방 생성
if (rs.next()) { //rs.next값이 true(있다)일 경우
    String id2 = rs.getString(1); //데이터값을 변수에 넣어줌
    String pw = rs.getString(2);
    String name = rs.getString(3);
    String tel = rs.getString(4);
    String email = rs.getString(5);
    String addr = rs.getString(6);
    String birth = rs.getString(7);
    bag.setId(id2); //변수로 넣어 가져온 데이터값들을 가방에 넣어줌
    bag.setPw(pw);
    bag.setName(name);
    bag.setTel(tel);
    bag.setEmail(email);
    bag.setAddr(addr);
    bag.setBirth(birth);
}
return bag; //가방으로 결과값 반환
```

User DB.java

```
</td>
<td>
    <input name = "id" type="text" value = <%=bag.getId()%>
    <!-- 기본입력값을 read로 불러와 value로 넣어줌. id는 고유값으로 수정이 불가능
</td>
>
```

UserUpdate.jsp

Value값으로 VO에 있는 데이터를 꺼내온다.

데이터가 true(있다)일 경우 데이터 값을 변수에 넣어
VO에 넣어
결과값을 반환

회원정보 수정 및 탈퇴

*전체 항목 모두 필수 항목입니다.

아이디	won
패스워드	
이름	yejin
핸드폰번호	01012345678
E-mail	dfsdf@naver.com
주소	서울특별시
생년월일	19970402

회원정보 수정

회원탈퇴

III. 메뉴별 소개 : 회원정보 수정

회원가입과 동일한 형태로 입력된 데이터들을 Form으로 묶어 데이터를 처리해주는 jsp로 전송

회원정보 수정 및 탈퇴

*전체 항목 모두 필수 항목입니다.

아이디	won
패스워드	
이름	yejin
핸드폰번호	01012345678
E-mail	dfsdf@naver.com
주소	서울특별시
생년월일	19970402

회원정보 수정

회원탈퇴

```

5
6 <%
7   userDB db = new userDB();
8   /* userupdate에서 입력된 수정값들을 변수로 받음 */
9   String id = request.getParameter("id");
10  String pw = request.getParameter("pw");
11  String name = request.getParameter("name");
12  String email = request.getParameter("email");
13  String tel = request.getParameter("tel");
14  String addr = request.getParameter("addr");
15  String birth = request.getParameter("birth");
16
17  /* 가방생성 */
18  userVO bag = new userVO();
19
20  /* 수정데이터를 가방에 넣기 */
21  bag.setId(id);
22  bag.setPw(pw);
23  bag.setName(name);
24  bag.setTel(tel);
25  bag.setEmail(email);
26  bag.setAddr(addr);
27  bag.setBirth(birth);
28
29  /* 가방으로 update메서드 실행 */
30  db.update(bag);
31
32 %>
  
```

UserUpdate2.jsp

III. 메뉴별 소개 : 회원정보 수정

```
public void update(userVO bag) throws Exception { //입력값이 데이터가 모여있는 가방으로 update메서드 실행
```

```
Class.forName("com.mysql.jdbc.Driver");
System.out.println("1. 커넥터 사용 설정 성공. <br>");

String url = "jdbc:mysql://localhost:3306/jeju?useUnicode=true&characterEncoding=utf8";
Connection con = DriverManager.getConnection(url, "root", "1234");
System.out.println("2. db연결 성공. <br>");
```

```
String sql = "update user_information set pw=?, name=?, tel=?, email=?, addr=?, birth=? where id=?";
PreparedStatement ps = con.prepareStatement(sql);
```

```
ps.setString(1, bag.getPw()); //왼쪽부터 순서대로 가방에서 데이터 꺼내와 입력
ps.setString(2, bag.getName());
ps.setString(3, bag.getTel());
ps.setString(4, bag.getEmail());
ps.setString(5, bag.getAddr());
ps.setString(6, bag.getBirth());
ps.setString(7, bag.getId());
System.out.println("3. sql문을 만들기 성공. <br>");
```

```
ps.executeUpdate();
System.out.println("4. SQL문을 mySQL서버로 전송 성공. <br>");
```

User DB.java

```
}//update
```

	id	pw	name	tel	email	addr	birth
1	won	1234	yejin	01012345678	dfsdf@naver.com	서울특별시	19970402

기존데이터에 값을 변경해주는 sql문 생성.

id는 고유한 값으로 변경하지않고 조건으로 사용한다.

비밀번호를 123 -> 1234 변경

회원정보 수정 및 탈퇴

*전체 항목 모두 필수 항목입니다.

아이디	won
패스워드	1234
이름	yejin
핸드폰번호	01012345678
E-mail	dfsdf@naver.com
주소	서울특별시
생년월일	19970402
회원정보 수정	회원탈퇴

III. 메뉴별 소개 : 회원정보 수정

```
<meta charset="UTF-8">
<title>회원정보 수정 및 탈퇴</title>
<!-- 스크립트로 회원정보 수정이 완료되었다는 알림창 띄우기 -->
<script type="text/javascript">
  alert("회원정보 수정이 완료되었습니다.")
</script>
</head>
```

회원정보 수정 완료시
자바스크립트로 알림창을 띄운다.

localhost:8888 내용:

회원정보 수정이 완료되었습니다.

확인

III. 메뉴별 소개 : 회원탈퇴

회원정보 수정 및 탈퇴

*전체 항목 모두 필수 항목입니다.

아이디	won
패스워드	
이름	yejin
핸드폰번호	01012345678
E-mail	dfsdf@naver.com
주소	서울특별시
생년월일	19970402

회원정보 수정

회원탈퇴

```
</td>
<td>
  <a href = "UserDelete.jsp">
    <button type = "button" >회원탈퇴</button>
  </a>
</td>
```

UserUpdate2.jsp

```
<%
  userDB db = new userDB(); /* db처리해줄 도구 생성 */
  String id = (String)session.getAttribute("id"); /* id세션받아서 id변수에 넣어주기 */
  db.delete(id); /* delete메서드에 id를 넣어 실행 */

  session.invalidate(); /* delete 완료 후 세션 끊기 */

  response.sendRedirect("MainPage.jsp"); /* 메인페이지로 화면 전환 */
%>
```

UserDelete.jsp

세션을 받아 delete메서드를 실행

메서드 실행 후 세션을 끊고
메인페이지로 화면을 전환해 준다.

III. 메뉴별 소개 : 회원탈퇴

```
public void delete(String id) throws Exception { //입력값을 id로 받아 delete메서드 실행

    Class.forName("com.mysql.jdbc.Driver");
    System.out.println("1. 커넥터 사용 설정 성공. <br>");

    String url = "jdbc:mysql://localhost:3306/jeju?useUnicode=true&characterEncoding=utf8";

    Connection con = DriverManager.getConnection(url, "root", "1234");
    System.out.println("2. db연결성공. <br>");

    String sql = "delete from user_information where id = ?"; //id가 조건인 데이터를 delete
    PreparedStatement ps = con.prepareStatement(sql); // sql문 인식

    ps.setString(1, id); // id조건에 입력 id 넣어주기

    System.out.println("3. sql문을 만들기 성공. <br>");

    ps.executeUpdate();
    System.out.println("4. SQL문을 mySQL서버로 전송 성공. <br>");

} // delete
```

Id를 조건으로 user_information테이블에서 데이터를 찾아 삭제하는 sql

IV. 소감문 : 원예진

프로그래밍에 대한 지식이 전무해 인생 처음으로 진행한 프로젝트였습니다. 브라우저, 서버, 데이터베이스와의 관계를 생각해보고 데이터 처리가 어떠한 순서로 진행되는지 고려해 하나의 웹페이지를 만들어내는 작업을 진행했습니다.

코드 상으로는 문제가 없어 보여 실행시켜보면 데이터베이스의 칼럼명이 자바의 함수명과 같아 변수를 설정하는 과정에서 계속 오류가 났습니다. 결국에는 모든 파일의 관련 변수명을 수정하고 나서야 오류가 해결되었습니다. 혼자였으면 어떠한 원인으로 생긴 오류인지 찾아내지 못했을 것인데 팀원들과 협력해서 머리를 맞대고 고민을 해보니 풀리지 않던 문제도 해결이 되었고 그 과정에서 다른 팀원들에게 많이 배울 수 있어 좋은 경험 이었습니다.

Javascript, jsp, java 등등 만든 파일에 따라서 코드가 조금씩 변경되어 어려웠고 특히 jsp같은 경우는 html과 java코드가 한 파일 안에서 모두 사용할 수 있어 구조를 이해하는데 어려움을 겪기도 했습니다. 1차 프로젝트를 완료한 지금 이 시점에선 jsp, java 는 코드를 짜는 방법을 터득했지만 아직 javascript같은 동적인 처리를 해주는 코드가 들어오면 많이 어려워하는 것 같습니다.

DAO파일을 만들어보면서 자바가 왜 객체지향언어인지, 메서드와 객체는 뭔지 등 배웠던 지식들이 더 와 닿았습니다. 객체지향이라는 말을 이해하는데까지 꽤 오랜시간이 걸렸는데 프로젝트를 진행하면서 경험의 중요성을 더욱 깨달은 것 같습니다.

다음 프로젝트에서는 더 다양하고 심화된 코드들을 다채롭게 사용해보고 아직 미숙한 DB부분을 더 공부해 어떤 데이터가 들어오던 그 처리를 잘 해보고싶다는 생각이 들었습니다.