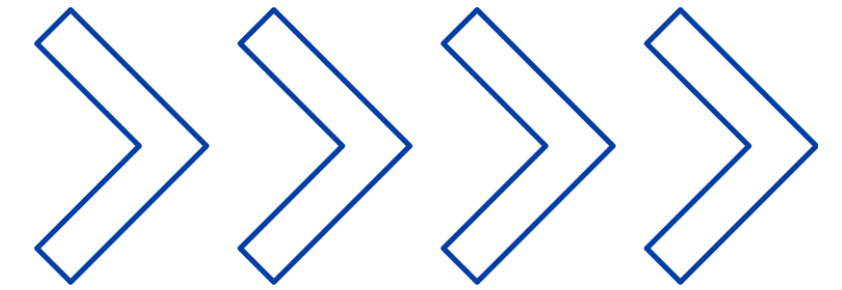


# OPEN APP STORE

JSP PROJECT



# 목차



**01** CONCEPT

**02** 프로젝트 일정

**03** E-R Diagram

**04** SQL Model

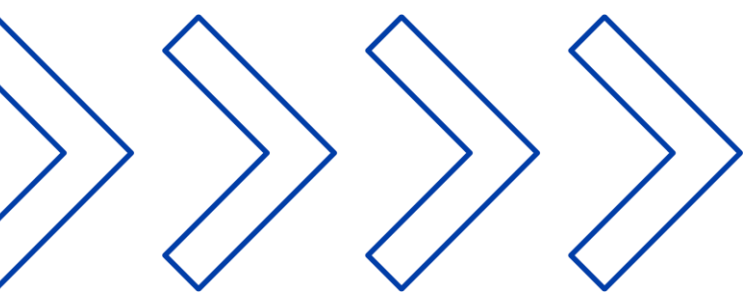
**05** 파일명세서

**06** CODE

**07** 실행결과

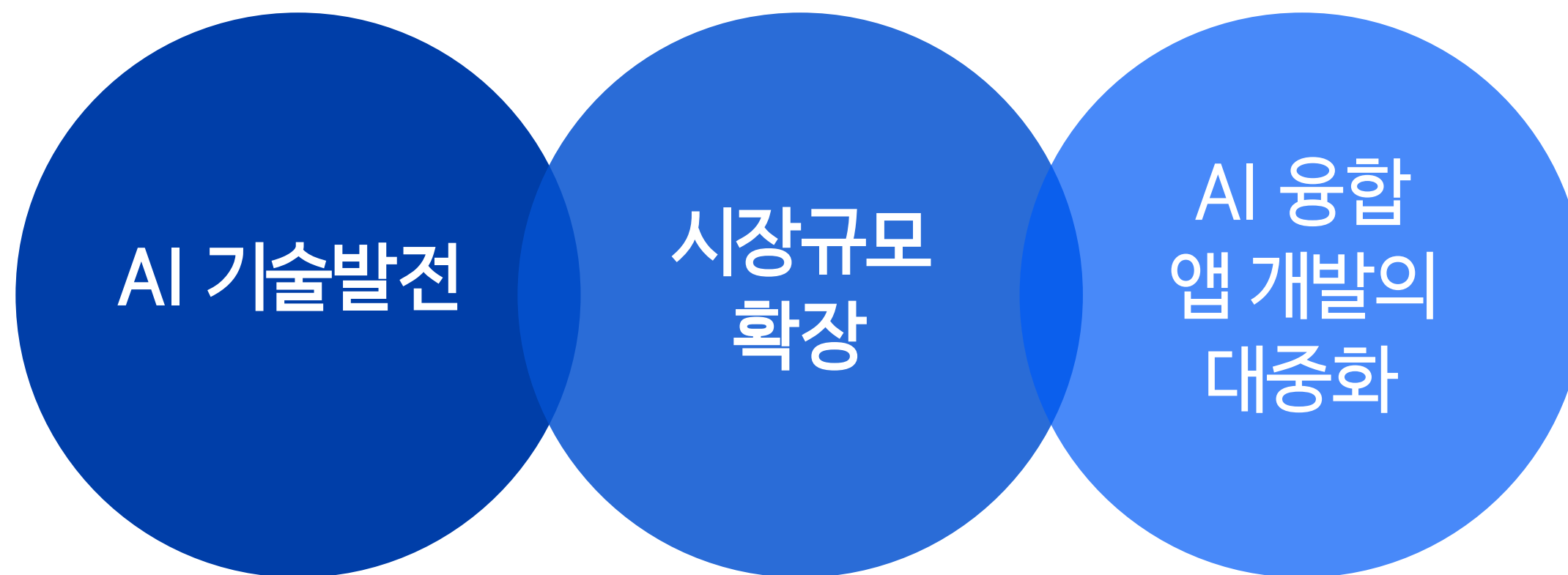
**08** Evaluation

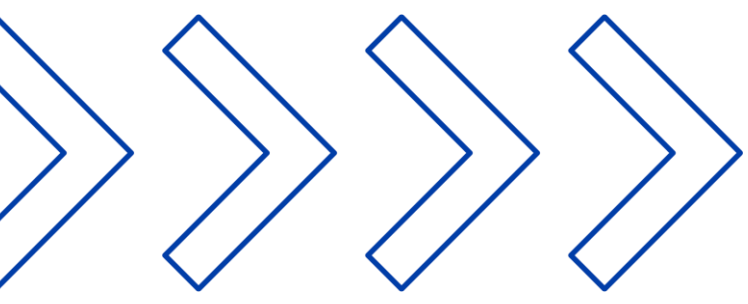
**09** Q & A



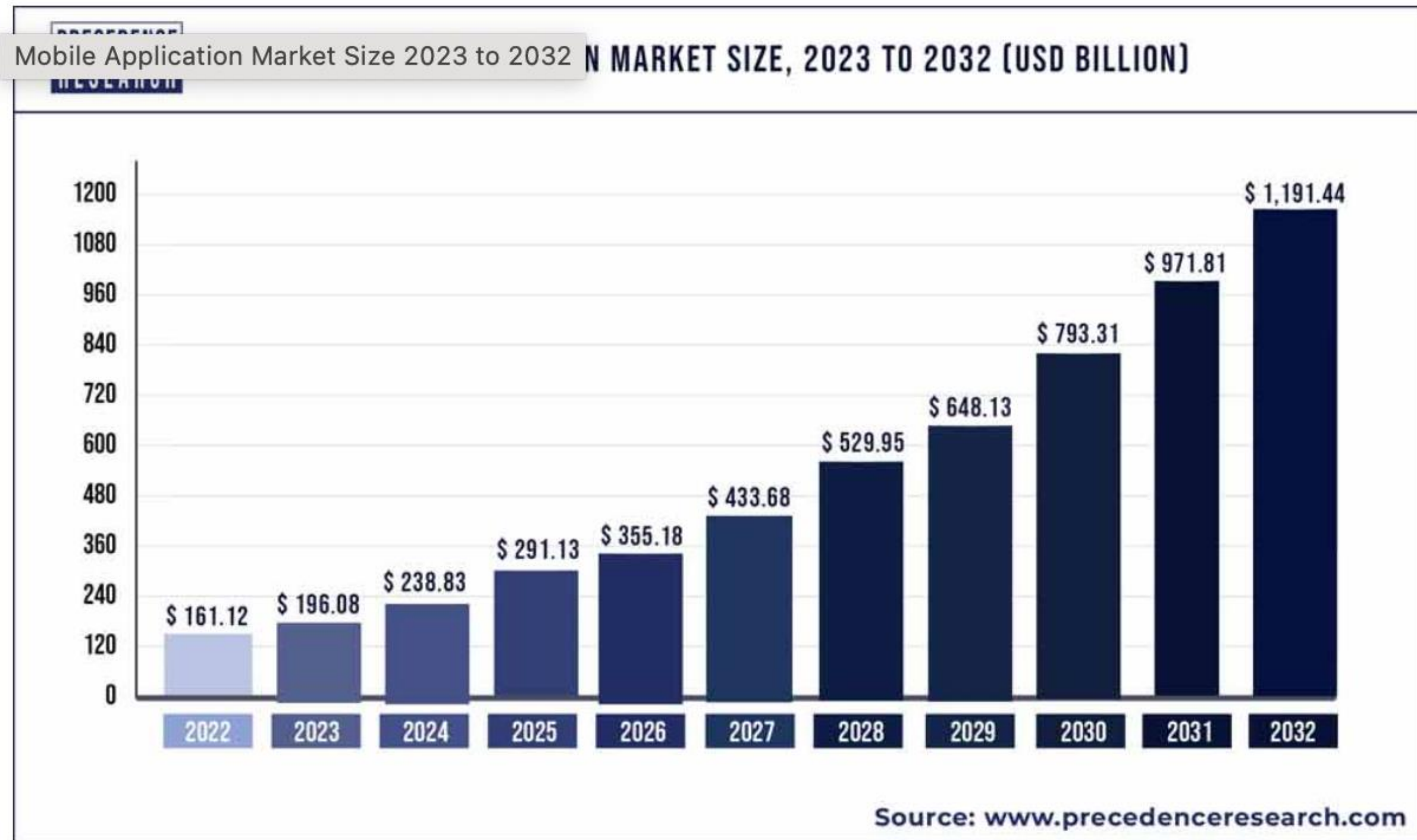
# CONCEPT

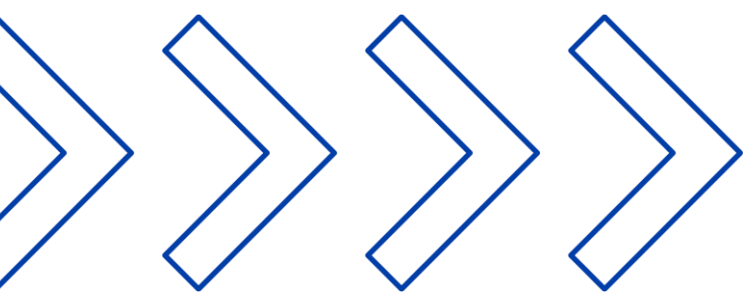
인공지능 기술의 발전으로 인한 앱 개발의 대중화 및 시장 규모의 확장으로 인한  
다양한 앱스토어 확충이 필요할 것으로 전망



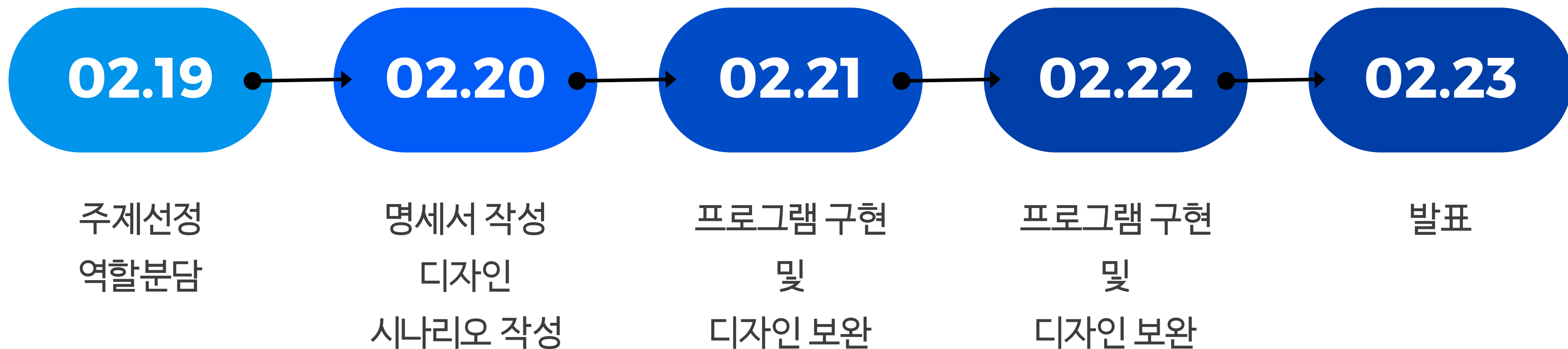


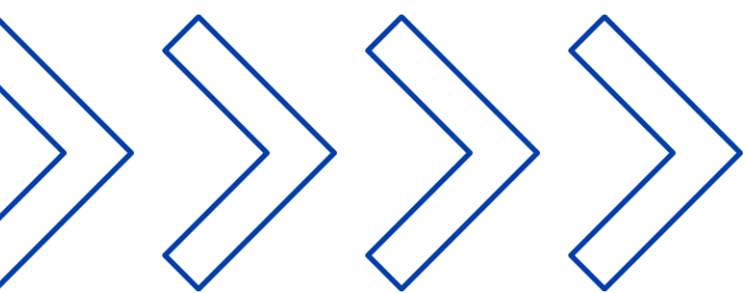
# CONCEPT



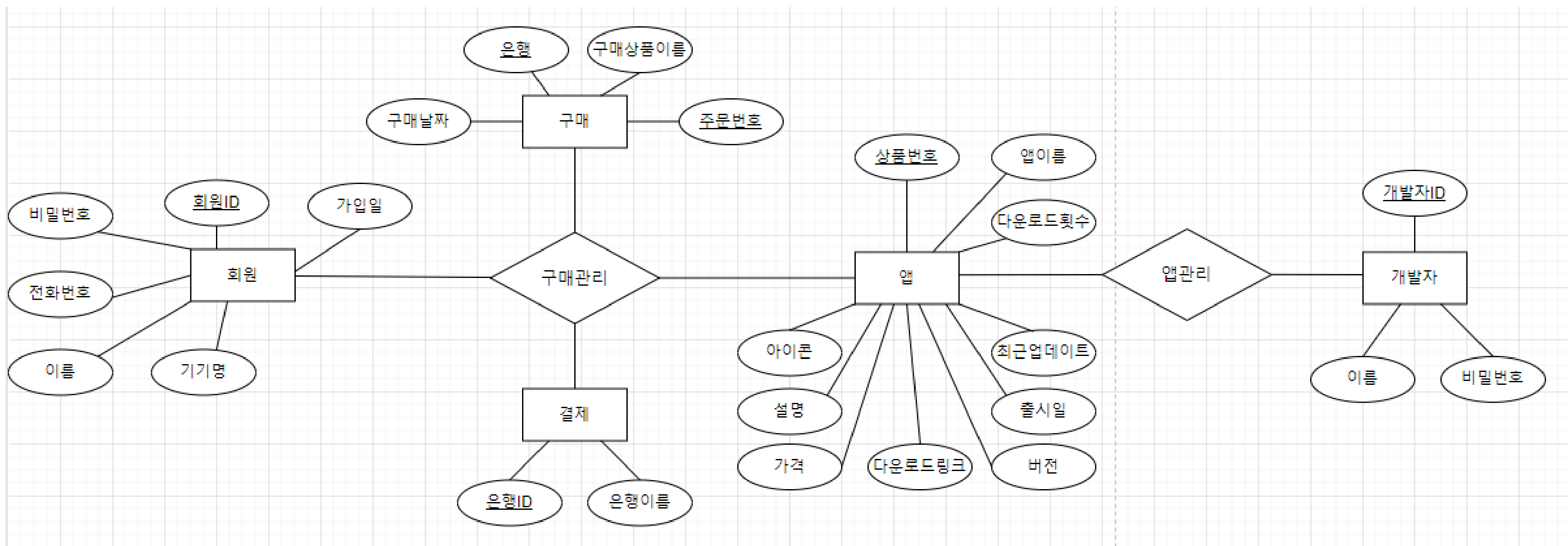


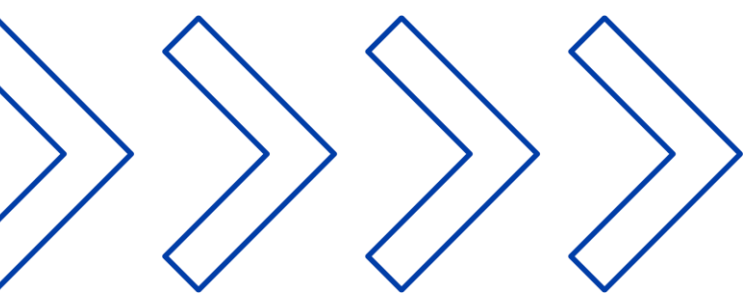
# 프로젝트 일정



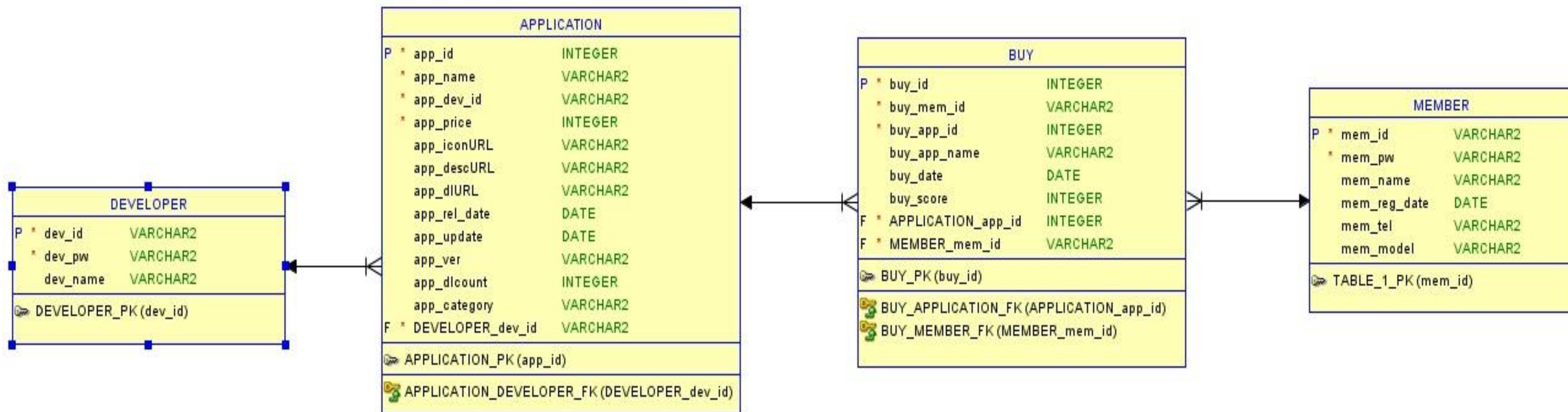


# E-R Diagram





# SQL Model





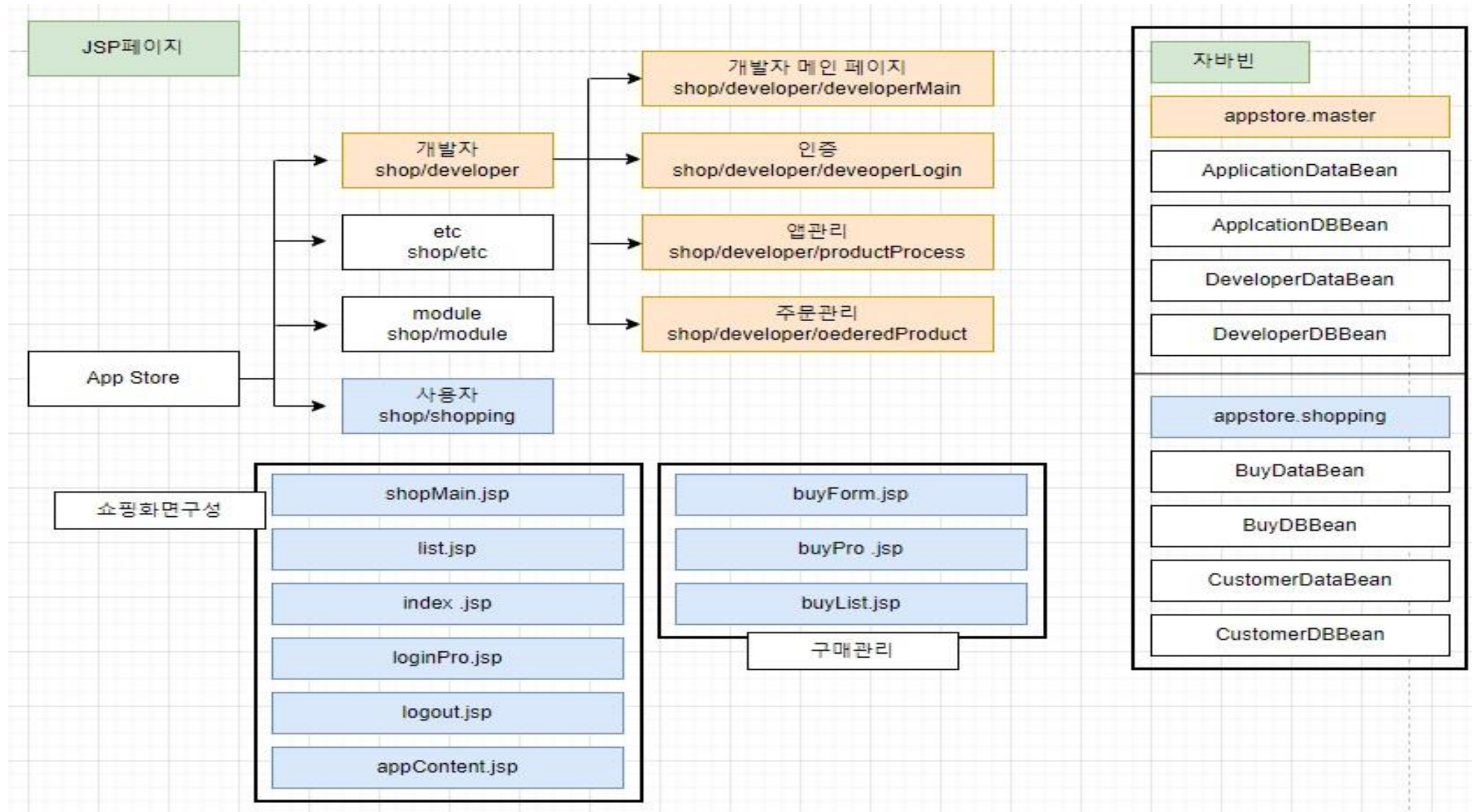
# 테이블

```
- create table developer (  
  dev_id varchar(20) not null primary key,  
  dev_pw varchar(20) not null,  
  dev_name varchar(20) not null  
);  
  
- create table member (  
  mem_id varchar(20) not null primary key,  
  mem_pw varchar(20) not null,  
  mem_name varchar(20) not null,  
  mem_reg_date date not null,  
  mem_tel varchar(20) not null,  
  mem_model varchar(20) not null  
);  
  
- create table application (  
  app_id int not null primary key auto_increment,  
  app_name varchar(20) not null,  
  app_dev_id varchar(20) not null,  
  app_price int not null,  
  app_iconURL varchar(200) not null,  
  app_descURL varchar(200) not null,  
  app_dlURL varchar(200) not null,  
  app_rel_date date not null,  
  app_update date not null,  
  app_ver varchar(20) not null,  
  app_dlcount int not null,  
  app_category varchar(20) not null  
);
```

```
- create table buy (  
  buy_id bigint not null primary key auto_increment,  
  buy_mem_id varchar(20) not null,  
  buy_app_id int not null,  
  buy_app_name varchar(20) not null,  
  buy_date date not null,  
  buy_bank varchar(20) not null,  
  buy_score int  
);  
  
- create table bank (  
  bank_id varchar(20) not null primary key,  
  bank_name varchar(20) not null  
);  
  
insert into bank values  
('sh', '신한은행'),  
('db', '대구은행'),  
('kb', '국민은행');
```

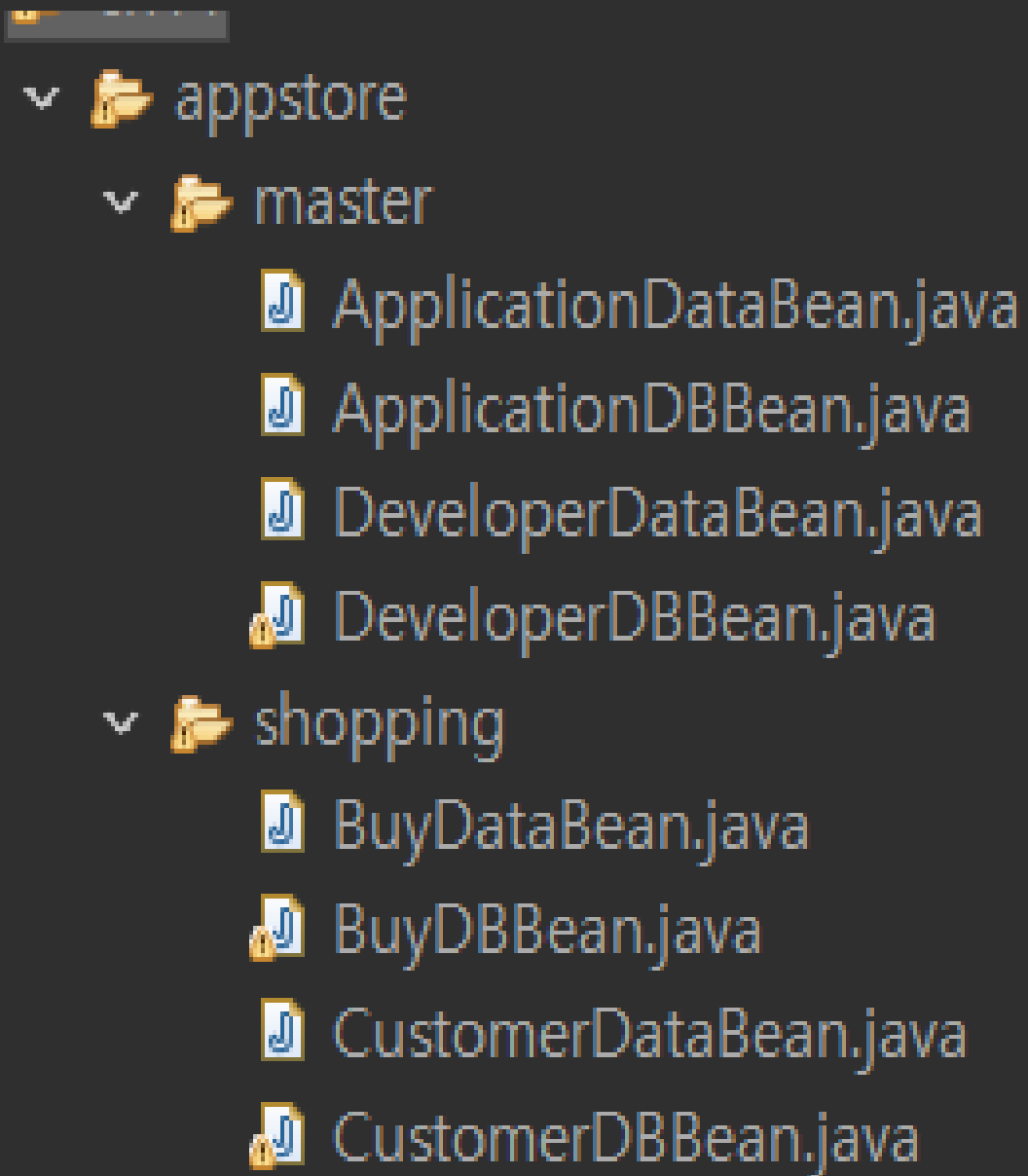


# 파일명세서



# JAVA CODE

## 자바빈



```
1 package ch14.appstore.master;
2
3 public class ApplicationDataBean {
4     private int app_id;
5     private String app_name;
6     private String app_dev_id;
7     private int app_price;
8     private String app_iconURL;
9     private String app_descURL;
10    private String app_dlURL;
11    private String app_rel_date;
12    private String app_update;
13    private String app_ver;
14    private int app_dlcount;
15    private String app_category;
16
17    public int getApp_id() {
18        return app_id;
19    }
20    public void setApp_id(int app_id) {
21        this.app_id = app_id;
22    }
23    public String getApp_name() {
24        return app_name;
25    }
26    public void setApp_name(String app_name) {
27        this.app_name = app_name;
28    }
29    public String getApp_dev_id() {
30        return app_dev_id;
31    }
32    public void setApp_dev_id(String app_dev_id) {
33        this.app_dev_id = app_dev_id;
34    }
35    public int getApp_price() {
36        return app_price;
37    }
38    public void setApp_price(int app_price) {
39        this.app_price = app_price;
40    }
41    public String getApp_descURL() {
42        return app_descURL;
43    }
44    public void setApp_descURL(String app_descURL) {
45        this.app_descURL = app_descURL;
46    }
47    public String getApp_rel_date() {
48        return app_rel_date;
49    }
50    public void setApp_rel_date(String app_rel_date) {
51        this.app_rel_date = app_rel_date;
52    }
53    public String getApp_update() {
54        return app_update;
55    }
56    public void setApp_update(String app_update) {
57        this.app_update = app_update;
58    }
59    public String getApp_ver() {
60        return app_ver;
61    }
62    public void setApp_ver(String app_ver) {
63        this.app_ver = app_ver;
64    }
65    public int getApp_dlcount() {
66        return app_dlcount;
67    }
68    public void setApp_dlcount(int app_dlcount) {
69        this.app_dlcount = app_dlcount;
70    }
71    public String getApp_category() {
72        return app_category;
73    }
74    public void setApp_category(String app_category) {
75        this.app_category = app_category;
76    }
77 }
```

```
1 package ch14.appstore.master;
2
3 import java.sql.Connection;
4
5
6
7
8
9
10
11
12
13 public class ApplicationDBBean {
14
15     private static ApplicationDBBean instance = new ApplicationDBBean();
16
17     public static ApplicationDBBean getInstance() {
18         return instance;
19     }
20
21     private ApplicationDBBean() {
22
23     }
24
25     // 커넥션풀로부터 커넥션객체를 얻어내는 메소드
26     private Connection getConnection() throws Exception {
27         Context initCtx = new InitialContext();
28         Context envCtx = (Context) initCtx.lookup("java:comp/env");
29         DataSource ds = (DataSource) envCtx.lookup("jdbc/basicjsp");
30         return ds.getConnection();
31     }
32 }
```

# JAVA CODE

```
--
32 // 개발자 인증 메소드
33 public int developerCheck(String id, String passwd) throws Exception {
34     Connection conn = null;
35     PreparedStatement pstmt = null;
36     ResultSet rs = null;
37     String dbpasswd = "";
38     int x = -1;
39
40     try {
41         conn = getConnection();
42
43         pstmt = conn.prepareStatement("select dev_pw from developer where dev_id = ? ");
44         pstmt.setString(1, id);
45
46         rs = pstmt.executeQuery();
47
48         if (rs.next()) {
49             dbpasswd = rs.getString("dev_pw");
50             if (dbpasswd.equals(passwd))
51                 x = 1; // 인증 성공
52             else
53                 x = 0; // 비밀번호 틀림
54         } else
55             x = -1; // 해당 아이디 없음
56
57     } catch (Exception ex) {
58         ex.printStackTrace();
59     } finally {
60         if (rs != null)
61             try {
62                 rs.close();
63             } catch (SQLException ex) {
64             }
65         if (pstmt != null)
66             try {
67                 pstmt.close();
68             } catch (SQLException ex) {
69             }
70         if (conn != null)
71             try {
72                 conn.close();
73             } catch (SQLException ex) {
74             }
75     }
76     return x;
77 }
78
```

```
--
79 // 앱 등록 메소드
80 public void insertApp(ApplicationDataBean app) throws Exception {
81     Connection conn = null;
82     PreparedStatement pstmt = null;
83
84     try {
85         conn = getConnection();
86
87         pstmt = conn.prepareStatement("insert into application values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
88         pstmt.setInt(1, app.getApp_id());
89         pstmt.setString(2, app.getApp_name());
90         pstmt.setString(3, app.getApp_dev_id());
91         pstmt.setInt(4, app.getApp_price());
92         pstmt.setString(5, app.getApp_iconURL());
93         pstmt.setString(6, app.getApp_descURL());
94         pstmt.setString(7, app.getApp_dlURL());
95         pstmt.setString(8, app.getApp_rel_date());
96         pstmt.setString(9, app.getApp_update());
97         pstmt.setString(10, app.getApp_ver());
98         pstmt.setInt(11, app.getApp_dlcount());
99         pstmt.setString(12, app.getApp_category());
100
101         pstmt.executeUpdate();
102
103     } catch (Exception ex) {
104         ex.printStackTrace();
105     } finally {
106         if (pstmt != null)
107             try {
108                 pstmt.close();
109             } catch (SQLException ex) {
110             }
111         if (conn != null)
112             try {
113                 conn.close();
114             } catch (SQLException ex) {
115             }
116     }
117 }
118
```



# JAVA CODE

```
366 // app_id에 해당하는 앱의 정보를 얻어내는 메소드로
367 // 등록된 앱을 수정하기 위해 수정폼으로 읽어들이기 위한 메소드
368 public ApplicationDataBean getApp(int app_id) throws Exception {
369     Connection conn = null;
370     PreparedStatement pstmt = null;
371     ResultSet rs = null;
372     ApplicationDataBean app = null;
373
374     try {
375         conn = getConnection();
376
377         pstmt = conn.prepareStatement("select * from application where app_id = ?");
378         pstmt.setInt(1, app_id);
379
380         rs = pstmt.executeQuery();
381
382         if (rs.next()) {
383             app = new ApplicationDataBean();
384
385             app.setApp_name(rs.getString("app_name"));
386             app.setApp_dev_id(rs.getString("app_dev_id"));
387             app.setApp_price(rs.getInt("app_price"));
388             app.setApp_iconURL(rs.getString("app_iconURL"));
389             app.setApp_descURL(rs.getString("app_descURL"));
390             app.setApp_dlURL(rs.getString("app_dlURL"));
391             app.setApp_rel_date(rs.getString("app_rel_date"));
392             app.setApp_update(rs.getString("app_update"));
393             app.setApp_ver(rs.getString("app_ver"));
394             app.setApp_dlcount(rs.getInt("app_dlcount"));
395             app.setApp_category(rs.getString("app_category"));
396         }
397     } catch (Exception ex) {
398         ex.printStackTrace();
399     } finally {
400         if (rs != null)
401             try {
402                 rs.close();
403             } catch (SQLException ex) {
404             }
405         if (pstmt != null)
406             try {
407                 pstmt.close();
408             } catch (SQLException ex) {
409             }
410         if (conn != null)
411             try {
412                 conn.close();
413             } catch (SQLException ex) {
414             }
415     }
```

```
416
417
418 // 등록된 앱의 정보를 수정시 사용하는 메소드
419 public void updateApp(ApplicationDataBean app, int app_id) throws Exception {
420     Connection conn = null;
421     PreparedStatement pstmt = null;
422     String sql;
423
424     try {
425         conn = getConnection();
426
427         sql = "update application set app_name=?, app_dev_id=?, app_price=?";
428         sql += ", app_iconURL=?, app_descURL=?, app_dlURL=?, app_rel_date=?";
429         sql += ", app_update=?, app_ver=?, app_dlcount=?, app_category=?";
430         sql += " where app_id=?";
431
432         pstmt = conn.prepareStatement(sql);
433
434         pstmt.setString(1, app.getApp_name());
435         pstmt.setString(2, app.getApp_dev_id());
436         pstmt.setInt(3, app.getApp_price());
437         pstmt.setString(4, app.getApp_iconURL());
438         pstmt.setString(5, app.getApp_descURL());
439         pstmt.setString(6, app.getApp_dlURL());
440         pstmt.setString(7, app.getApp_rel_date());
441         pstmt.setString(8, app.getApp_update());
442         pstmt.setString(9, app.getApp_ver());
443         pstmt.setInt(10, app.getApp_dlcount());
444         pstmt.setString(11, app.getApp_category());
445         pstmt.setInt(12, app_id);
446
447         pstmt.executeUpdate();
448
449     } catch (Exception ex) {
450         ex.printStackTrace();
451     } finally {
452         if (pstmt != null)
453             try {
454                 pstmt.close();
455             } catch (SQLException ex) {
456             }
457     }
```

# JAVA CODE

```
465
466 // app_id에 해당하는 앱의 정보를 삭제시 사용하는 메소드
467 public void deleteApp(int app_id) throws Exception {
468     Connection conn = null;
469     PreparedStatement pstmt = null;
470     ResultSet rs = null;
471
472     try {
473         conn = getConnection();
474
475         pstmt = conn.prepareStatement("delete from application where app_id=?");
476         pstmt.setInt(1, app_id);
477
478         pstmt.executeUpdate();
479
480     } catch (Exception ex) {
```

```
479
480     } catch (Exception ex) {
481         ex.printStackTrace();
482     } finally {
483         if (rs != null)
484             try {
485                 rs.close();
486             } catch (SQLException ex) {
487             }
488         if (pstmt != null)
489             try {
490                 pstmt.close();
491             } catch (SQLException ex) {
492             }
493         if (conn != null)
494             try {
495                 conn.close();
496             } catch (SQLException ex) {
497             }
498     }
499 }
500 }
```



# JSP CODE

레이아웃

> TileSave

▼ layout

- ≡ footer.jsp
- ≡ header.jsp
- ≡ login\_modal.jsp
- ≡ sidebar\_end.jsp
- ≡ sidebar\_shop\_end.jsp
- ≡ sidebar\_shop.jsp
- ≡ sidebar.jsp

webapps / studyjsp / cartshop / shopping / = shopmain.jsp

```
1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2      pageEncoding="UTF-8"%>
3
4  <%@ include file="../../etc/color.jspf"%>
5  <%@ include file="/layout/header.jsp" %>
6  <%@ include file="/layout/sidebar_shop.jsp"%>
7  <jsp:include page="introList.jsp" flush="false" />
8  <%@ include file="/layout/sidebar_shop_end.jsp"%>
9  <%@ include file="/layout/footer.jsp" %>
10
```



# JSP CODE

```
<html>
<head>
<title>AppStore</title>
<script type="text/javascript" src="../../../etc/script.js"></script>
<script src="https://cdn.tailwindcss.com"></script>
<link href="https://cdnjs.cloudflare.com/ajax/libs/flowbite/2.3.0/flowbite.min.css" rel="stylesheet">
</head>
<body>
```

```
<nav class="bg-white dark:bg-gray-900 fixed w-full z-20 top-0 start-0 border-b border-gray-200 dark:border-gray-700">
  <div class="max-w-screen-xl flex flex-wrap items-center justify-between mx-auto p-4">
    <a href="/StudyJSP/ch14shop/index.jsp" class="flex items-center space-x-3 rtl:space-x-reverse">
      
      <span class="self-center text-2xl font-semibold whitespace-nowrap dark:text-white">AppStore</span>
    </a>
    <div class="flex md:order-2 space-x-3 md:space-x-0 rtl:space-x-reverse">
```

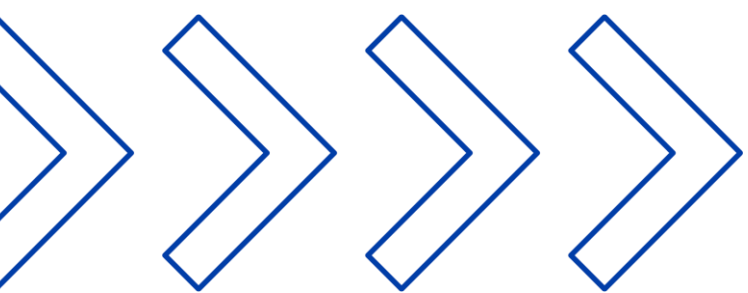
pps / StudyJSP / layout / = footer.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<script src="https://cdnjs.cloudflare.com/ajax/libs/flowbite/2.3.0/flowbite.min.js"></script>
</body>
</html>
```

# JSP CODE 뷰컨트롤러

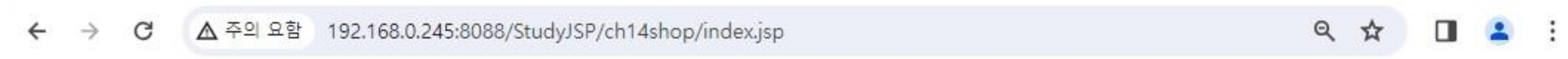
▼ shopping
≡ dmemDeleteForm.jsp
≡ dmemDeletePro.jsp
≡ dmemRegisterForm.jsp
≡ dmemRegisterPro.jsp
≡ dmemUpdateForm.jsp
≡ dmemUpdatePro.jsp
≡ introList.jsp
≡ list.jsp
≡ loginForm.jsp
≡ loginPro.jsp
≡ logout.jsp
≡ memDeleteForm.jsp
≡ memDeletePro.jsp
≡ memRegisterForm.jsp
≡ memRegisterPro.jsp
≡ memRegisterSelect.jsp

```
<%  
String mem_id = request.getParameter("mem_id");  
String mem_pw = request.getParameter("mem_pw");  
String mem_name = request.getParameter("mem_name");  
String mem_tel = request.getParameter("mem_tel");  
String mem_model = request.getParameter("mem_model");  
  
mem.setMem_id(mem_id);  
mem.setMem_pw(mem_pw);  
mem.setMem_name(mem_name);  
mem.setMem_tel(mem_tel);  
mem.setMem_model(mem_model);  
  
CustomerDBBean memProcess = CustomerDBBean.getInstance();  
memProcess.insertMember(mem);  
  
%>  
  
<script>  
    alert("회원가입이 완료되었습니다.");  
    location.href = "shopMain.jsp";  
</script>
```



# INDEX PAGE

<http://192.168.0.245:8088/StudyJSP/ch14shop/index.jsp>



 Open AppStore

[Home](#) [Store](#) [Developer](#) [Contact](#)

[로그인](#)

## 오픈소스가 스타트업을 구한다

순수 오픈소스로 만들어진 프리웨어 애플리케이션으로 쉽고 간편하고 알차게  
귀사의 첫 사업을 시작하세요.

[시작하기](#)

# Login Page

← → ↻ 주의 요함 192.168.0.245:8088/StudyJSP/ch14shop/developer/logon/developerLoginForm.jsp



[Home](#) [Store](#) [Developer](#) [Contact](#)

## 개발자 회원 로그인

아이디

아이디를 입력하세요.

비밀번호

\*\*\*\*\*

☐ Remember me

[Forgot password?](#)

로그인

회원가입

또는 일반 회원 로그인

← → ↻ 주의 요함 192.168.0.245:8088/StudyJSP/ch14shop/shopping/loginForm.jsp ☆ □ 👤 ⋮



[Home](#) [Store](#) [Developer](#) [Contact](#)

로그인

## 일반 회원 로그인

아이디

아이디를 입력하세요.

비밀번호

\*\*\*\*\*

☐ Remember me

[Forgot password?](#)

로그인

회원가입

또는 앱 개발자 로그인



# 일반회원

Open AppStore

HomeStoreDeveloperContact로그인

Open AppStore

일반 회원가입

아이디

아이디를 입력하세요.

이름

이름을 입력하세요.

전화번호

전화번호를 입력하세요.

비밀번호

\*\*\*\*\*

스마트폰 기종

스마트폰 기종을 입력하세요.

가입하기

이미 가입 하셨나요? 로그인 하러가기

Open AppStore

HomeStoreDeveloperContact로그인

← 메인으로

전체목록보기

11

사진

3

게임

3

운동


3

요리

2

신작 소개


사진 분류의 신작목록: 더보기



잡스의작품1

개발자 : jobs@apple.com

가격 : 1000원




빌의추억1

개발자 : bill@ms.com

가격 : 1000원


게임 분류의 신작목록: 더보기



잡스의추억2

개발자 : jobs@apple.com

가격 : 2000원



빌의추억2

개발자 : bill@ms.com

가격 : 1300원

# 개발자 페이지

AppStore										
Home Store Developer Contact										
로그아웃										
관리자 메인화면										
상품등록										
상품수정/삭제										
전체구매목록 확인 3										
로그아웃										
개발자 정보수정										
번호	앱 분류	앱 이름	가격	개발자	출시일	업데이트일	버전	다운로드횟수	수정	삭제
1	100	잡스의 작품1	1000	jobs@apple.com	2024-02-01	2024-02-13	1.01	3	수정	삭제
2	200	잡스의 작품2	2000	jobs@apple.com	2023-06-04	2024-02-07	1.01	0	수정	삭제
3	300	잡스의 작품3	3000	jobs@apple.com	2024-01-29	2024-02-07	1.01	2	수정	삭제
4	100	잡스의 추억1	1500	jobs@apple.com	2024-01-29	2024-02-07	1.01	0	수정	삭제
5	200	잡스의 추억2	2000	jobs@apple.com	2024-02-07	2024-02-08	1.01	2	수정	삭제
6	300	잡스의 추억3	1500	jobs@apple.com	2024-02-05	2024-02-06	1.01	0	수정	삭제

AppStore					
Home Store Developer Contact					
로그아웃					
관리자 메인화면					
상품등록					
상품수정/삭제					
전체구매목록 확인 3					
로그아웃					
번호	구매자ID	구매 앱	구매 일시	결제 계좌	별점
10	abc101@naver.com	1	2024-02-22	db 대구은행	0
12	abc101@naver.com	6	2024-02-22	db 대구은행	0
14	abc101@naver.com	3	2024-02-22	db 대구은행	0

# Evaluation

## 이상목

JSP를 활용한 웹 개발 프로젝트를 통해, 전통적인 서버 사이드 동적 콘텐츠 생성의 장점을 직접 경험할 수 있었습니다. JSP의 태그 라이브러리와 쉬운 데이터 접근 방식은 백엔드 로직을 웹 페이지에 효율적으로 통합하는 데 큰 도움이 되었습니다.

## 안승국

코딩보다는 디자인에 더 자신있다고 자부했었지만, CSS 레이아웃은 집에서 편하게 작업할 때는 괜찮았지만, 막상 밖에서 팀 프로젝트로 할 때는 기한에 대한 압박감 때문인지, 팀원들과 맞지 않는 소스 파일 때문인지, 좀 처럼 집에서 때 처럼 만족스러운 결과를 만들어내지 못했다. 그래서 내가 잘 할 수 있는 유일한 작업은 이미지 생성 뿐이었다. 특히 아이콘을 만들 때는 집에서나 밖에서나 재밌었는데, CSS 디자인은 기한에 맞춰야 한다는 압박감과 팀원들과는 달라도 너무 다른 HTML 구조로 인한 소스 호환성, 이전에 써본 적이 없는 새로운 라이브러리에 대한 접근 난이도(필요로 하는 템플릿이 하필 유료) 탓에 부담감과 난이도가 내 한계점을 넘어버려서 큰 벽을 느꼈고, 디자이너로서의 자질이 부족함을 느껴 그 점이 매우 아쉬웠다.

## 이종운

자바빈과 커넥션풀을 이용하여 DB와 웹페이지를 연동하는 작업을 많이 연습할 수 있었습니다. 기능을 구현한 후 CSS를 이용하여 여러 레이아웃을 적용함으로써 매우 다양한 페이지를 만들 수 있음을 알게 되었습니다. 자바를 원활히 다룰 수 있어서 필요한 메소드를 만들거나 다루는데에 큰 어려움은 없었습니다. 프로젝트 기간이 짧아서 아쉬웠습니다..

## 최재희

팀원들의 배려로 수업시간에 완전히 이해하지 못했던 커넥션풀에 대해서 더 공부할 수 있어서 감사하다. JSP뿐만 아니라 이전 수업했던 분야에 대해서 복습을 할 수 있어서 더 값진 시간이었다. 다만 프로젝트 기간이 짧아 코딩에 적극적으로 참여하지 못해서 아쉬움이 남는다..

# OPEN APP STORE

## Q & A





감사합니다