

JSP와 서블릿

(Web Programming)

웹 프로젝트 설정

• JSP와 서블릿이란?

- 서블릿(servlet)이란 자바 클래스를 기반으로 하는 웹 애플리케이션 프로그래밍 기술이다.
- JSP(Java Server Page)도 자바를 기반으로 하는 웹 애플리케이션 기술로 HTML 문서 안에 자바 코드가 삽입되는 구조를 갖는다.

• 톰캣 서버등록 및 실행

- 1) 이클립스 메뉴에서 [Window]-[Show View]-[Servers] 메뉴를 선택한다.
- 2) 이클립스 [Servers] 탭을 더블클릭한 후 톰캣 서버의 버전과 프로그램 위치를 설정한다.
- 3) 이클립스 [Servers] 탭에서 실행버튼을 클릭한 후 브라우저에 아래 주소로 접속한다.

```
http://localhost:8080
```

▪ 톰캣 포트 충돌인 경우

```
netstat -ano|find "8088"  
taskkill /f /pid 18064
```

- 외부의 이미지, 자바스크립트, CSS를 인식하기 위해 [Servers]의 web.xml에 아래 내용을 추가한다.

```
[Servers] web.xml  
  
<servlet-mapping>  
  <servlet-name>default</servlet-name>  
  <url-pattern>/</url-pattern>  
  <url-pattern>*.css</url-pattern>  
  <url-pattern>*.png</url-pattern>  
  <url-pattern>*.gif</url-pattern>  
  <url-pattern>*.js</url-pattern>  
  <url-pattern>*.jpg</url-pattern>  
  <url-pattern>*.svg</url-pattern>  
  <url-pattern>*.ico</url-pattern>  
</servlet-mapping>
```

상품관리 프로그램작성

• 데이터베이스 생성

- 데이터베이스, 테이블을 생성하고 Sample 데이터를 입력한다.

데이터베이스, 사용자 생성

```
create database webdb;  
create user web identified by 'pass';  
drop user web;  
grant all privileges on webdb.* to web@'%';
```

상품 테이블 생성

```
use webdb  
create table products(  
  code int auto_increment primary key,  
  name varchar(100) not null,  
  price int default 0,  
  rdate datetime default now()  
);
```

상품 Sample 데이터 입력

```
insert into products(code, name, price) values(100, '비스코프 냉장고', 30000000);
insert into products(name,price) values('오브제 세탁기', 25000000);
insert into products(name,price) values('오브제 스타일러', 15000000);
```

• 프로젝트 생성 및 설정

- 1) [File]-[Dynamic Web Project] 메뉴에서 상품관리 프로젝트(ex01)를 생성한다.
- 2) [Servers]에서 톰캣서버를 더블클릭 한 후 [Modules] 탭에서 프로젝트를 등록한 후 path를 /로 변경한다.

[src/main/java]-[controller] HomeController.java

```
@WebServlet("/")
public class HomeController extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setAttribute("pageName", "about.jsp");
        RequestDispatcher dis=request.getRequestDispatcher("home.jsp");
        dis.forward(request, response);
    }
}
```

[src]-[main]-[webapp] home.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<html>
<head>
    <title>상품관리프로그램</title>
    <!-- CSS only -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65 "
        crossorigin="anonymous">
    <!-- JavaScript Bundle with Popper -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-kenU1KFdBLE4zVF0s0G1M5b4hcxpyD9F7jL+jjXkk+Q2h455rYXK/7HAuoII+0I4 "
        crossorigin="anonymous"></script>
    <link rel="stylesheet" href="/css/style.css"/>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col">
                <jsp:include page="header.jsp"/>
                <jsp:include page="${pageName}"/>
                <jsp:include page="footer.jsp"/>
            </div>
        </div>
    </div>
</body>
</html>
```

[src]-[main]-[webapp] about.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">회사소개</h1>
    </div>
</div>
```

[src]-[main]-[webapp] header.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="row">
    <div class="col">
        
        <div>
            <a href="/">Home</a>
            <a href="/pro/list">상품목록</a>
            <a href="/pro/insert">상품등록</a>
        </div>
    </div>
    <hr>
</div>
</div>
```

[src]-[main]-[webapp] footer.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="row my-5">
    <div class="col">
        <hr>
        <h3 class="text-center mb-5">Copyright 2023. 인천아카데미 홍길동 all rights reserved.</h3>
    </div>
</div>
```

[src]-[main]-[webapp]-[css] style.css

```
@import url('https://fonts.googleapis.com/css2?family=Gugi&display=swap');
* {
    font-family: 'Gugi', cursive;
}
a {
    margin-right: 20px;
}
@media screen and (max-width: 960px) {
    #img_header {
        display: none;
    }
}
```

- 상품목록, 상품등록 페이지를 위한 ProductController를 작성한다.

[src/main/java]-[controller] ProductController.java

```
@WebServlet(value= {"/pro/list", "/pro/insert"})
public class ProductController extends HttpServlet {
    private static final long serialVersionUID = 1L;
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
        switch(request.getServletPath()) {
            case "/pro/list":
                request.setAttribute("pageName", "/pro/list.jsp");
                dis.forward(request, response);
                break;
            case "/pro/insert":
                request.setAttribute("pageName", "/pro/insert.jsp");
                dis.forward(request, response);
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[pro] list.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="row my-5 justify-content-center">
    <div class="col-md-6">
        <h1 class="text-center mb-5">상품목록</h1>
    </div>
</div>
```

[src]-[main]-[webapp]-[pro] insert.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="row my-5 justify-content-center">
    <div class="col-md-6">
        <h1 class="text-center mb-5">상품등록</h1>
    </div>
</div>
```

• 상품목록 출력하기

- Mysql 라이브러리를 [src]-[main]-[webapp]-[WEB-INF]-[lib]에 mysql-connector-java-x.x.x.-bin.jar 파일을 복사한다.

[src/main/java]-[model] Database.java

```
public class Database {
    public static Connection CON;
    static {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            CON=DriverManager.getConnection("jdbc:mysql://localhost:3306/webdb", "web", "pass");
            System.out.println("접속성공");
        } catch (Exception e) {
            System.out.println("접속실패:" + e.toString());
        }
    }
}
```

[src/main/java]-[model] ProductVO.java

```
import java.util.Date;
public class ProductVO {
    private int code;
    private String name;
    private int price;
    private Date rdate;
    public int getCode() {
        return code;
    }
    public void setCode(int code) {
        this.code = code;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    //setter(), getter(), toString() 메서드를 generate한다.
}
```

[src/main/java]-[model] ProductDAO.java

```
package model;
import java.sql.*;
import java.util.*;

public class ProductDAO {
    //상품목록
    public ArrayList<ProductVO> list(){
        ArrayList<ProductVO> array=new ArrayList<>();
        try {
            String sql="select * from products order by code desc";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                ProductVO vo=new ProductVO();
                vo.setCode(rs.getInt("code"));
                vo.setName(rs.getString("name"));
                vo.setPrice(rs.getInt("price"));
                vo.setRdate(rs.getTimestamp("rdate"));
                array.add(vo);
            }
        } catch(Exception e) {
            System.out.println("상품목록:" + e.toString());
        }
        return array;
    }
}
```

- ProductController에 아래 내용을 추가한다.

[src/main/java]-[controller] ProductController.java

```
package controller;
import model.*;

@WebServlet(value= {"/pro/list", "/pro/insert"})
public class ProductController extends HttpServlet {
    private static final long serialVersionUID = 1L;
    ProductDAO dao=new ProductDAO();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
        switch(request.getServletPath()) {
            case "/pro/list":
                request.setAttribute("array", dao.list());
                request.setAttribute("pageName", "/pro/list.jsp");
                dis.forward(request, response);
                break;
            ...
        }
    }
}
```

- JSTL(JSP Standard Tag Library)
- JSTL을 이용해 간단한 프로그램 로직의 구사, 다른 JSP페이지 호출, 날짜/시간/숫자의 포맷, 데이터베이스로의 입력/수정/삭제/조회, XML 문서의 처리, 문자열을 처리하는 함수 호출 기능들이 가능하다.
- JSTL을 설치하려면 JSTL라이브러리 파일을 다운로드 받은 후 특정 폴더에 아래와 같이 복사해 둔다.
 - 1) <http://jakarta.apache.org/>에서 jstl.jar와 standard.jar 파일을 다운로드한다.
 - 2) [webContent]-[WEB-INF]-[lib] 폴더에 위 두 개의 파일을 복사해 둔다.

```
[src]-[main]-[webapp]-[pro] list.jsp
```

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<div class="row my-5">
  <div class="col">
    <h1 class="text-center mb-5">상품목록</h1>
    <table class="table table-striped">
      <c:forEach items="${array}" var="vo">
        <tr>
          <td>${vo.code}</td>
          <td>${vo.name}</td>
          <td><fmt:formatNumber value="${vo.price}" pattern="#,###"/></td>
          <td><fmt:formatDate value="${vo.rdate}" pattern="yyyy-MM-dd HH:mm:ss"/></td>
        </tr>
      </c:forEach>
    </table>
  </div>
</div>
```

• 상품등록 하기

- 상품등록 DAO와 상품등록 JSP 페이지, Controller를 작성한다.

```
[src]-[main]-[webapp]-[pro] insert.jsp
```

```
...
<form class="card p-3" method="post" name="frm">
  <div class="input-group my-2">
    <span class="input-group-text">상품명</span>
    <input class="form-control" name="name">
  </div>
  <div class="input-group my-2">
    <span class="input-group-text">상품가격</span>
    <input class="form-control" name="price">
  </div>
  <div class="text-center mt-3">
    <input type="submit" value="상품등록" class="btn btn-primary">
    <input type="reset" value="상품취소" class="btn btn-secondary">
  </div>
</form>
<script>
  $(frm).on("submit", function(e){
    e.preventDefault();
    let name=$(frm.name).val();
    let price=$(frm.price).val();
    if(name==""){
      alert("이름을 입력하세요!");
      $(frm.name).focus();
    }else if(price==" || price.replace(/[0-9]/g,"")){
      alert("가격을 숫자로 입력하세요!");
      $(frm.price).val("");
      $(frm.price).focus();
    }else{
      if(confirm("새로운 상품을 등록하실래요?")) frm.submit();
    }
  });
</script>
```

```
[src/main/java]-[model] ProductDAO.java
```

```
public class ProductDAO {
    public void insert(ProductVO vo) { //상품등록
        try {
            String sql="insert into products(name, price) values(?, ?)";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, vo.getName());
            ps.setInt(2, vo.getPrice());
            ps.execute();
        } catch (Exception e) {
            System.out.println("상품등록:" + e.toString());
        }
    }
    ...
}
```

```
[src/main/java]-[controller] ProductController.java
```

```
@WebServlet(value= {"/pro/list", "/pro/insert"})
public class ProductController extends HttpServlet {
    ...
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        switch(request.getServletPath()) {
            case "/pro/insert":
                String name=request.getParameter("name");
                String price=request.getParameter("price");
                ProductVO vo=new ProductVO();
                vo.setName(name);
                vo.setPrice(Integer.parseInt(price));
                dao.insert(vo);
                response.sendRedirect("/pro/list");
                break;
        }
    }
}
```

• 상품정보 출력하기

- 상품정보 출력 DAO와 상품정보 출력 JSP 페이지, Controller를 작성한다.

```
[src]-[main]-[webapp]-[pro] read.jsp
```

```
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<div class="row my-5 justify-content-center">
    <div class="col-md-6">
        <h1 class="text-center mb-3">상품정보</h1>
        <div class="card p-3">
            <h5>상품코드: ${vo.code}</h5>
            <h5>상품명: ${vo.name}</h5>
            <h5>상품가격: <fmt:formatNumber value="${vo.price}" pattern="#,###원"/></h5>
            <h5>상품수정일: <fmt:formatDate value="${vo.rdate}" pattern="yyyy-MM-dd HH:mm:ss"/></h5>
        </div>
        <div class="text-center my-3">
            <button class="btn btn-primary">수정</button>
            <button class="btn btn-danger">삭제</button>
        </div>
    </div>
</div>
</div>
```


[src/main/java]-[model] ProductDAO.java

```
public class ProductDAO {
    //상품정보
    public ProductVO read(String code) {
        ProductVO vo=new ProductVO();
        try {
            String sql="select * from products where code=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setInt(1, Integer.parseInt(code));
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                vo.setCode(rs.getInt("code"));
                vo.setName(rs.getString("name"));
                vo.setPrice(rs.getInt("price"));
                vo.setRdate(rs.getTimestamp("rdate"));
            }
        }catch(Exception e) {
            System.out.println("상품정보:" + e.toString());
        }
        return vo;
    }
    ...
}
```

[src/main/java]-[controller] ProductController.java

```
@WebServlet(value= {"/pro/list", "/pro/insert", "/pro/read"})
public class ProductController extends HttpServlet {
    private static final long serialVersionUID = 1L;
    ProductDAO dao=new ProductDAO();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
        switch(request.getServletPath()) {
            case "/pro/read":
                String code=request.getParameter("code");
                request.setAttribute("vo", dao.read(code));
                request.setAttribute("pageName", "/pro/read.jsp");
                dis.forward(request, response);
                break;
            ...
        }
    }
}
```

• 상품정보 삭제하기

- 상품정보 출력페이지에 아래 내용을 추가하고 삭제 DAO와 Controller를 작성한다.

[src]-[main]-[webapp]-[pro] read.jsp

```
<script>
    const code="${vo.code}";
    //삭제버튼을 클릭한 경우
    $(".btn-danger").on("click", function(){
        if(confirm(code + "번 상품을 삭제하시겠습니까?")){
            location.href="/pro/delete?code=" + code; //삭제로이동
        }
    });
</script>
```

```
[src/main/java]-[model] ProductDAO.java
```

```
package model;
import java.sql.*;
import java.util.*;

public class ProductDAO {
    ...
    //상품삭제
    public void delete(String code) {
        try {
            String sql="delete from products where code=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, code);
            ps.execute();
        }catch(Exception e) {
            System.out.println("상품삭제:" + e.toString());
        }
    }
}
```

```
[src/main/java]-[controller] ProductController.java
```

```
@WebServlet(value= {"/pro/list", "/pro/insert", "/pro/read", "/pro/delete"})
public class ProductController extends HttpServlet {
    private static final long serialVersionUID = 1L;
    ProductDAO dao=new ProductDAO();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            ...
            case "/pro/delete":
                code=request.getParameter("code");
                //System.out.println("delete code....." + code);
                dao.delete(code);
                response.sendRedirect("/pro/list");
                break;
        }
    }
}
```

- **상품정보 수정하기**

- 상품정보 출력페이지에 아래 내용을 추가하고 수정페이지, 수정DAO, 수정 Controller를 작성한다.

```
[src]-[main]-[webapp]-[pro] read.jsp
```

```
<script>
    const code="${vo.code}";
    //수정버튼을 클릭한 경우
    $(".btn-primary").on("click", function(){
        location.href="/pro/update?code=" + code;
    });

    //삭제버튼을 클릭한 경우
    $(".btn-danger").on("click", function(){
        if(confirm(code + "번 상품을 삭제하실래요?")){
            location.href="/pro/delete?code=" + code;
        }
    });
</script>
```

[src]-[main]-[webapp]-[pro] update.jsp

```
<div class="row my-5 justify-content-center">
  <div class="col-md-6">
    <h1 class="text-center mb-3">상품수정</h1>
    <form class="card p-3" method="post" name="frm">
      <div class="input-group my-2">
        <span class="input-group-text">상품코드</span>
        <input class="form-control" name="code" value="${vo.code}" readonly>
      </div>
      <div class="input-group my-2">
        <span class="input-group-text">상품명</span>
        <input class="form-control" name="name" value="${vo.name}">
      </div>
      <div class="input-group my-2">
        <span class="input-group-text">상품가격</span>
        <input class="form-control" name="price" value="${vo.price}">
      </div>
      <div class="text-center mt-3">
        <input type="submit" value="상품수정" class="btn btn-primary">
        <input type="reset" value="상품취소" class="btn btn-secondary">
      </div>
    </form>
  </div>
</div>
<script>
  $(frm).on("submit", function(e){
    e.preventDefault();
    let name=$(frm.name).val();
    let price=$(frm.price).val();
    if(name==""){
      alert("이름을 입력하세요!");
      $(frm.name).focus();
    }else if(price==" || price.replace(/[0-9]/g,"")){
      alert("가격을 숫자로 입력하세요!");
      $(frm.price).val("");
      $(frm.price).focus();
    }else{
      if(confirm("상품 정보를 수정하실래요?")) frm.submit();
    }
  });
</script>
```

[src/main/java]-[model] ProductDAO.java

```
public class ProductDAO {
  ...
  //상품수정
  public void update(ProductVO vo) {
    try {
      String sql="update products set name=?,price=?,rdate=now() where code=?";
      PreparedStatement ps=Database.CON.prepareStatement(sql);
      ps.setString(1, vo.getName());
      ps.setInt(2, vo.getPrice());
      ps.setInt(3, vo.getCode());
      ps.execute();
    }catch(Exception e) {
      System.out.println("상품수정:" + e.toString());
    }
  }
}
```

```
[src/main/java]-[controller] ProductController.java
```

```
import model.*;
```

```
@WebServlet(value= {"/pro/list", "/pro/insert", "/pro/read", "/pro/delete", "/pro/update"})
```

```
public class ProductController extends HttpServlet {
```

```
    private static final long serialVersionUID = 1L;
```

```
    ProductDAO dao=new ProductDAO();
```

```
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
```

```
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
```

```
        switch(request.getServletPath()) {
```

```
        case "/pro/list":
```

```
            ...
```

```
        case "/pro/insert":
```

```
            request.setAttribute("pageName", "/pro/insert.jsp");
```

```
            dis.forward(request, response);
```

```
            break;
```

```
        case "/pro/read":
```

```
            ...
```

```
            request.setAttribute("vo", dao.read(code));
```

```
            request.setAttribute("pageName", "/pro/read.jsp");
```

```
            dis.forward(request, response);
```

```
            break;
```

```
        case "/pro/delete":
```

```
            code=request.getParameter("code");
```

```
            dao.delete(code);
```

```
            response.sendRedirect("/pro/list");
```

```
            break;
```

```
        case "/pro/update":
```

```
            code=request.getParameter("code");
```

```
            request.setAttribute("vo", dao.read(code));
```

```
            request.setAttribute("pageName", "/pro/update.jsp");
```

```
            dis.forward(request, response);
```

```
            break;
```

```
    }
```

```
}
```

```
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
```

```
        request.setCharacterEncoding("UTF-8");
```

```
        switch(request.getServletPath()) {
```

```
        case "/pro/insert":
```

```
            ...
```

```
            dao.insert(vo);
```

```
            response.sendRedirect("/pro/list");
```

```
            break;
```

```
        case "/pro/update":
```

```
            String code=request.getParameter("code");
```

```
            name=request.getParameter("name");
```

```
            price=request.getParameter("price");
```

```
            vo=new ProductVO();
```

```
            vo.setCode(Integer.parseInt(code));
```

```
            vo.setName(name);
```

```
            vo.setPrice(Integer.parseInt(price));
```

```
            //System.out.println(vo.toString());
```

```
            dao.update(vo);
```

```
            response.sendRedirect("/pro/read?code=" + code);
```

```
            break;
```

```
    }
```

```
}
```

```
}
```

- 상품목록 페이징 기능
- 상품목록에 페이징 기능을 추가하기 위한 DAO, Controller, JSP 페이지를 추가한다.

[src]-[main]-[webapp]-[pro] update.jsp

```
public class ProductDAO {
    //상품목록
    public ArrayList<ProductVO> list(int page){
        ArrayList<ProductVO> array=new ArrayList<>();
        try {
            String sql="select * from products order by code desc limit ?, 5";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setInt(1, (page-1)*5);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                ProductVO vo=new ProductVO();
                ...
                array.add(vo);
            }
        }catch(Exception e) {
            System.out.println("상품목록:" + e.toString());
        }
        return array;
    }

    //전체상품수
    public int total() {
        int total=0;
        try {
            String sql="select count(*) cnt from products";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                total=rs.getInt("cnt");
            }
        }catch(Exception e) {
            System.out.println("상품전체수:" + e.toString());
        }
        return total;
    }
}
```

[src/main/java]-[controller] ProductController.java

```
public class ProductController extends HttpServlet {
    ...
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
        case "/pro/list":
            int page=request.getParameter("page")==null ? 1 : Integer.parseInt(request.getParameter("page"));
            request.setAttribute("page", page);
            request.setAttribute("total", dao.total());
            request.setAttribute("array", dao.list(page));
            request.setAttribute("pageName", "/pro/list.jsp");
            dis.forward(request, response);
            break;
        ...
    }
    ...
}
```

[src]-[main]-[webapp]-[pro] list.jsp

```
<div class="row my-5">
  <div class="col">
    <h1 class="text-center mb-5">상품목록</h1>
    ...
    <div class="text-center">
      <button class="btn btn-primary" id="prev">이전</button>
      <span class="mx-2" id="page">0</span>
      <button class="btn btn-primary" id="next">다음</button>
    </div>
  </div>
</div>
</div>
<script>
  let page="${page}";
  let total="${total}";
  let last = Math.ceil(total/5);
  $("#page").html(page + "/" + last);

  if(page==1) $("#prev").attr("disabled", true) else $("#prev").attr("disabled", false);
  if(page==last) $("#next").attr("disabled", true) else $("#next").attr("disabled", false);

  $("#prev").on("click", function(){
    page--;
    location.href="/pro/list?page=" + page;
  });
  $("#next").on("click", function(){
    page++;
    location.href="/pro/list?page=" + page;
  });
</script>
```

- 상품목록 페이지징 기능(Ajax)

- JSON Object 생성을 위한 gson-x.x.x.jar, json-simple-x.x.x.jar 파일을 [src]-[main]-[webapp]-[WEB-INF]-[lib] 폴더로 복사한다.
- 상품목록 Controller와 JSP 페이지를 수정한다.

[src/main/java]-[controller] ProductController.java

```
@WebServlet(value= {"/pro/list", "/pro/insert", "/pro/read", "/pro/delete", "/pro/update", "/pro/list.json"})
public class ProductController extends HttpServlet {
  ...
  protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out= response.getWriter();
    RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
    switch(request.getServletPath()) {
      case "/pro/list.json":
        int page=request.getParameter("page")==null ? 1 : Integer.parseInt(request.getParameter("page"));
        JSONObject jObject = new JSONObject();
        Gson gson=new Gson();
        jObject.put("total", dao.total());
        jObject.put("items", gson.toJson(dao.list(page)));
        out.print(jObject);
        break;
      ...
    }
    ...
  }
}
```

[src]-[main]-[webapp]-[pro] list.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="row my-5 justify-content-center">
  <div class="col">
    <h1 class="text-center mb-5">상품목록</h1>
    <div id="div_product"></div>
    <div class="text-center mt-3" id="btn-group">
      <button class="btn btn-primary" id="prev">이전</button>
      <span class="mx-3" id="page">1/10</span>
      <button class="btn btn-primary" id="next">다음</button>
    </div>
  </div>
</div>
<!-- 상품목록 템플릿 -->
<script id="temp_product" type="text/x-handlebars.template">
  <table class="table">
    {{#each items}}
    <tr>
      <td class="code">{{code}}</td>
      <td><input value="{{name}}" class="name"></td>
      <td>{{price}}</td>
      <td>{{rdate}}</td>
    </tr>
    {{/each}}
  </table>
</script>
<script>
  let page=1;
  getList();

  $("#next").on("click", function(){
    page++;
    getList();
  });
  $("#prev").on("click", function(){
    page--;
    getList();
  });

  function getList() {
    $.ajax({
      type:"get",
      url:"/pro/list.json",
      data:{page: page},
      dataType:"json",
      success:function(data){
        const temp=Handlebars.compile($("#temp_product").html());
        const html=temp(data);
        $("#div_product").html(html);

        const last=Math.ceil(data.total/5);
        $("#page").html(page + "/" + last);
        if(page==1) $("#prev").attr("disabled", true) else $("#prev").attr("disabled", false)
        if(page==last) $("#next").attr("disabled", true) else $("#next").attr("disabled", false);
        if(data.total==0) $("#btn-group").hide() else $("#btn-group").show();
      }
    });
  }
</script>
```

- 상품목록 검색기능

[src/main/java]-[model] ProductDAO.java

```
public class ProductDAO {
    //상품수 출력
    String sql="select count(*) cnt from products where name like ?";
    PreparedStatement ps=Database.CON.prepareStatement(sql);
    ps.setString(1, "%" + query + "%");
    ...
    //상품목록
    String sql="select * from products where name like ? order by code desc limit ?, 5";
    PreparedStatement ps=Database.CON.prepareStatement(sql);
    ps.setString(1, "%" + query + "%");
    ps.setInt(2, (page-1)*5);
    ...
}
```

[src/main/java]-[controller] ProductController.java

```
switch(request.getServletPath()) {
    case "/pro/list.json":
        int page=request.getParameter("page")==null ? 1 : Integer.parseInt(request.getParameter("page"));
        String query=request.getParameter("query")==null ? "" : request.getParameter("query");
        ...
        jsonObject.put("items", gson.toJson(dao.list(page)));
        out.print(jsonObject);
        break;
}
```

[src]-[main]-[webapp]-[pro] list.jsp

```
<div class="row">
    <form class="col-6" name="frm">
        <div class="input-group">
            <input class="form-control" placeholder="검색어" name="query">
            <button class="btn btn-primary">검색</button>
        </div>
    </form>
</div>
<script>
    ...
    let query="";
    $(frm).on("submit", function(e){
        e.preventDefault();
        page=1;
        query=$(frm.query).val();
        getList();
    });

    function getList(){
        $.ajax({
            type:"get",
            url:"/pro/list.json",
            data:{ "page": page, "query":query },
            dataType:"json",
            success:function(data){ ... }
        });
    }
</script>
```


- 상품목록 등록, 수정, 삭제 기능(Ajax)
- 상품목록 출력(Pagination)을 위한 Controller, JSP 페이지를 작성한다.

[src/main/java]-[controller] ProductController.java

```
@WebServlet(value={"/pro/list", "/pro/list.json", "/pro/total"})
public class ProductController extends HttpServlet {
    ProductDAO dao=new ProductDAO();
    SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    DecimalFormat df=new DecimalFormat("#,###");

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
        switch(request.getServletPath()) {
            case "/pro/list":
                request.setAttribute("pageName", "/pro/list.jsp");
                dis.forward(request, response);
                break;
            case "/pro/list.json": //pro/list.json?page=1 &query=냉장고
                int page=Integer.parseInt(request.getParameter("page"));
                String query=request.getParameter("query");
                ArrayList<ProductVO> array=dao.list(page, query);
                JSONArray jArray=new JSONArray();
                for(ProductVO vo:array) {
                    JSONObject obj=new JSONObject();
                    obj.put("code", vo.getCode());
                    obj.put("name", vo.getName());
                    obj.put("price", vo.getPrice());
                    obj.put("fprice", df.format(vo.getPrice()));
                    obj.put("fdate", sdf.format(vo.getRdate()));
                    jArray.add(obj);
                }
                out.print(jArray);
                break;
            case "/pro/total":
                query=request.getParameter("query");
                out.print(dao.total(query));
                break;
        }
    }
}
```

- home.jsp 페이지에 Pagination 라이브러리를 추가한다.

[src]-[main]-[webapp] home.js

```
<head>
    <title>상품관리프로그램</title>
    ...
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="xxx" >
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="x"></script>
    <script type="text/javascript"src="//dapi.kakao.com/v2/maps/sdk.js?appkey=424c8eabdcea01747b41c162ae95dd3d"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/twbs-pagination/1.4.2/jquery.twbsPagination.min.js"></script>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.3/font/bootstrap-icons.css">
    <link rel="stylesheet" href="/css/style.css"/>
</head>
```

[src]-[main]-[webapp]-[pro] list.jsp

```
<div class="row my-5">
  ...
  <div id="div_product"></div>
  <div id="pagination" class="pagination justify-content-center mt-5"></div>
</div>
...
<script>
  let page=1;
  let query=$(frm.query).val();

  $(frm).on("submit", function(e){
    e.preventDefault();
    query=$(frm.query).val();
    page=1;
    getTotal();
  });

  function getList() {
    ...
  }

  getTotal();
  function getTotal() {
    $.ajax({
      type:"get",
      url:"/pro/total",
      data:{query:query},
      success:function(data){
        if(data==0){
          $("#div_product").html("<h3 class='text-center my-5'>상품이 존재하지 않습니다.</h3>");
        }else{
          const totalPages=Math.ceil(data/5);
          $("#pagination").twbsPagination("changeTotalPages", totalPages, page);
        }
        if(data > 5) {
          $("#pagination").show();
        }else {
          $("#pagination").hide();
        }
      }
    });
  }

  $('#pagination').twbsPagination({
    totalPages:1, //총 페이지 번호 수
    visiblePages: 5, // 하단에서 한 번에 보여 지는 페이지 번호 수
    startPage : 1, // 시작 시 표시되는 현재 페이지
    initiateStartPageClick: false, // 플러그인이 시작 시 페이지 버튼 클릭 여부 (default : true)
    first : '<i class="bi bi-skip-start"></i>', // 페이지네이션 버튼 중 처음으로 돌아가는 버튼에 쓰여 있는 텍스트
    prev : '<i class="bi bi-caret-left"></i>', // 이전 페이지 버튼에 쓰여 있는 텍스트
    next : '<i class="bi bi-caret-right"></i>', // 다음 페이지 버튼에 쓰여 있는 텍스트
    last : '<i class="bi bi-skip-end"></i>', // 페이지네이션 버튼 중 마지막으로 가는 버튼에 쓰여 있는 텍스트
    onPageClick: function (event, clickPage) {
      page=clickPage;
      getList();
    }
  });
</script>
```

- 상품등록을 위한 Controller와 JSP 페이지를 작성한다.

```
[src/main/java]-[controller] ProductController.java
```

```
@WebServlet(value={"/pro/list", "/pro/list.json", "/pro/total", "/pro/insert"})
...
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    switch(request.getServletPath()) {
    case "/pro/insert":
        ProductVO vo=new ProductVO();
        vo.setName(request.getParameter("name"));
        vo.setPrice(Integer.parseInt(request.getParameter("price")));
        dao.insert(vo);
        break;
    }
}
```

```
[src]-[main]-[webapp]-[pro] insert.jsp
```

```
<div class="row justify-content-center">
    <div class="col-md-8">
        <h1 class="text-center">상품등록</h1>
        <form name="frm1" class="card p-3">
            <div class="input-group mb-3">
                <span class="input-group-text">상품명</span>
                <input name="name" class="form-control">
            </div>
            <div class="input-group mb-3">
                <span class="input-group-text">상품가격</span>
                <input name="price" class="form-control">
            </div>
            <div class="text-center">
                <input type="submit" value="상품등록" class="btn btn-primary">
                <input type="reset" value="등록취소" class="btn btn-secondary">
            </div>
        </form>
    </div>
</div>
<script>
    $(frm1).on("submit", function(e) {
        e.preventDefault();
        const name=$(frm1.name).val();
        const price=$(frm1.price).val();
        if(name==" " || price.replace(/[0-9]/g, "")){
            alert("상품정보를 입력하세요!");
        }else{
            if(confirm("새로운 상품을 등록하시겠습니까?")){
                $.ajax({
                    type:"post",
                    url:"/pro/insert",
                    data:{"name":name, "price":price},
                    success:function(){
                        $(frm1.name).val("");
                        $(frm1.price).val("");
                        $(frm1.name).focus();
                        getTotal();
                    }
                });
            }
        }
    });
</script>
```

```
[src]-[main]-[webapp]-[pro] list.jsp
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<jsp:include page="insert.jsp"/>
...
```

- 상품삭제를 위한 Controller와 목록 페이지를 수정한다.

```
[src/main/java]-[controller] ProductController.java
```

```
@WebServlet(value={"/pro/list", "/pro/list.json", "/pro/total", "/pro/insert", "/pro/delete"})
public class ProductController extends HttpServlet {
    ...
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            ...
            case "/pro/delete":
                int code=Integer.parseInt(request.getParameter("code"));
                System.out.println("삭제할코드:" + code);
                dao.delete(code);
                break;
        }
    }
}
```

```
[src]-[main]-[webapp]-[pro] list.jsp
```

```
...
<!-- 상품목록 템플릿 -->
<script id="temp_product" type="text/x-handlebars-template">
    <table class="table">
        {{#each .}}
        <tr>
            <td class="code">{{code}}</td>
            <td class="name">{{name}}</td>
            <td>{{fprice}}</td>
            <td>{{fdate}}</td>
            <td><button class="btn btn-danger btn-sm" code="{{code}}">삭제</td>
        </tr>
        {{/each}}
    </table>
</script>
<script>
    ...
    //삭제버튼을 클릭한 경우
    $("#div_product").on("click", ".btn-danger", function(){
        const code=$(this).attr("code");
        const name=$(this).parent().parent().find("name").text();
        if(confirm(name + " 상품을 삭제하실래요?")){
            $.ajax({
                type:"post",
                url:"/pro/delete",
                data:{"code":code},
                success:function(){
                    alert("삭제완료!");
                    getTotal();
                }
            });
        }
    });
</script>
```

- 상품수정을 위한 Controller와 목록 페이지를 수정한다.

[src/main/java]-[controller] ProductController.java

```
@WebServlet(value={"/pro/list", "/pro/list.json", "/pro/total", "/pro/insert", "/pro/delete", "/pro/update"})
...
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    switch(request.getServletPath()) {
        case "/pro/update":
            vo=new ProductVO();
            vo.setCode(Integer.parseInt(request.getParameter("code")));
            vo.setName(request.getParameter("name"));
            vo.setPrice(Integer.parseInt(request.getParameter("price")));
            dao.update(vo);
            break;
    }
}
```

[src]-[main]-[webapp]-[pro] list.jsp

```
<!-- 상품목록 템플릿 -->
<tr>
    <td class="code">{{code}}</td>
    <td><input value="{{name}}" class="name"></td>
    <td><input value="{{price}}" class="price" oninput="isNumber(this)"><span class="fprice">{{fprice}}</span></td>
    <td>{{fdate}}</td>
    <td><button class="btn btn-danger btn-sm" code="{{code}}">삭제</td>
    <td><button class="btn btn-success btn-sm">수정</button></td>
</tr>
</script>
...
//숫자인 경우에만 입력
function isNumber(item){
    item.value=item.value.replace(/[^0-9]/g,'');
}

//가격이 바뀔경우
$("#div_product").on("keydown", ".price", function(e){
    const fprice=$(this).parent().find(".fprice");
    const price=$(this).val();
    fprice.html(price.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ','));
});

//수정버튼을 클릭한 경우
$("#div_product").on("click", ".btn-success", function(){
    const row=$(this).parent().parent();
    const code=row.find(".code").html();
    const name=row.find(".name").val();
    const price=row.find(".price").val();
    if(confirm("상품정보를 수정하실래요?")){
        $.ajax({
            type:"post",
            url:"/pro/update",
            data:{code:code, name:name, price:price},
            success:function(){
                alert("수정완료!");
            }
        });
    }
});
});
</script>
```

• 카카오 검색 API 와 Ajax 활용

• 카카오 지역검색 API 예제

- 1) 카카오 개발자 센터 페이지(<https://developers.kakao.com/>)로 이동한다.
- 2) 내 애플리케이션에서 새로운 애플리케이션을 등록하여 API Key를 발급 받는다.
- 3) 지도 사용을 위해 [플랫폼]-[Web]-[사이트 도메인]에 <http://localhost:8080>를 등록한다.

- home.jsp 페이지에 json 데이터 출력을 위한 handlebars CDN 주소를 추가한다.

```
[src]-[main]-[webapp] home.jsp
```

```
<head>
  <script src="http://code.jquery.com/jquery-1.9.1.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/3.0.1/handlebars.js"></script>
  ...
</head>
```

```
[src]-[main]-[webapp]-[local] list.jsp
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="row my-5">
  <div class="col">
    <h1 class="text-center mb-5">지역검색</h1>
    <div id="div_local"></div>
  </div>
</div>

<!-- 지역검색목록 템플릿 -->
<script id="temp_local" type="text/x-handlebars-template">
  <table class="table">
    <tr>
      <td>{{id}}</td>
      <td class="place">{{place_name}}</td>
      <td>{{address_name}}</td>
      <td class="phone">{{phone}}</td>
    </tr>
  </table>
</script>

<script>
  getList();
  function getList() {
    $.ajax({
      type:"get",
      url:"https://dapi.kakao.com/v2/local/search/keyword.json", //로컬-REST API-키워드로 장소 검색하기
      headers:{"Authorization":"KakaoAK b80880fbde422de3fd9b4a4e67c9bb54"}, //REST API Key
      dataType:"json",
      data:{query:"인천 버거킹"},
      success:function(data) {
        //console.log(data);
        const temp=Handlebars.compile($("#temp_local").html());
        const html=temp(data);
        $("#div_local").html(html);
      }
    });
  }
</script>
```

▪ 페이지 이동 및 검색 기능 추가

[src]-[main]-[webapp]-[local] list.jsp

```
...
<div class="row justify-content-end mb-3">
  <form class="col-6 col-md-4" name="frm">
    <div class="input-group">
      <input class="form-control" name="query" value="인천 버거킹">
      <button class="btn btn-primary">검색</button>
    </div>
  </form>
</div>
<hr>
<div id="div_local"></div>
<div class="text-center my-3">
  <button class="btn btn-primary" id="prev">이전</button>
  <span class="mx-3" id="page">1/100</span>
  <button class="btn btn-primary" id="next">다음</button>
</div>
...
<!-- 지역검색목록 템플릿 -->
<script id="temp_local" type="text/x-handlebars-template">
  ...
</script>
<script>
  let page=1;
  getList();

  $("#next").on("click", function(){
    page++;
    getList();
  });
  $("#prev").on("click", function(){
    page--;
    getList();
  });

  $(frm).on("submit", function(e){
    e.preventDefault();
    page=1;
    getList();
  });

  function getList(){
    let query=$(frm.query).val();
    $.ajax({
      type:"get",
      url:"https://dapi.kakao.com/v2/local/search/keyword.json",
      headers:{"Authorization":"KakaoAK b80880fbde422de3fd9b4a4e67c9bb54"},
      dataType:"json",
      data:{query:query, size:5, page:page},
      success:function(data){
        if(page==1) $("#prev").attr("disabled", true) else $("#prev").attr("disabled", false);
        const last=Math.ceil(data.meta.pageable_count/5);
        if(data.meta.is_end) $("#next").attr("disabled", true) else $("#next").attr("disabled", false);
        $("#page").html(page + "/" + last);
      }
    });
  }
</script>
```

• 카카오 지도 API

- [내 애플리케이션]-[앱 키]-[JavaScript 키]를 home.jsp에 추가한다.

```
[src]-[main]-[webapp] home.jsp
```

```
<head>
...
<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=xxxxxx"></script>
</head>
```

```
[src]-[main]-[webapp]-[local] search.jsp
```

```
...
<!-- 지도 출력 -->
<div class="row">
  <div class="col">
    <div id="map" style="width:100%;height:400px;display:none;" class="card"></div>
  </div>
</div>
<script id="temp_local" type="text/x-handlebars-template">
  <table class="table">
    <tr>
      <td>{{id}}</td>
      <td class="place">{{place_name}}</td>
      <td>{{address_name}}</td>
      <td class="phone">{{phone}}</td>
      <td><button class="btn btn-success btn-sm" x="{{x}}" y="{{y}}">위치</button></td>
    </tr>
  </table>
</script>
<script>
  ...
  $("#div_local").on("click", ".btn-success", function() { //위치 버튼을 클릭한 경우
    $("#map").show();
    const x=$(this).attr("x");
    const y=$(this).attr("y");
    var container = document.getElementById('map');
    var options = {
      center: new kakao.maps.LatLng(y, x),
      level: 3
    };
    var map = new kakao.maps.Map(container, options);

    var markerPosition = new kakao.maps.LatLng(y, x); // 마커가 표시될 위치입니다
    var marker = new kakao.maps.Marker({ // 마커를 생성합니다
      position: markerPosition
    });
    marker.setMap(map); // 마커가 지도 위에 표시되도록 설정합니다

    let row = $(this).parent().parent();
    let place = row.find(".place").text();
    let phone = row.find(".phone").text();
    var str = "<div style='padding:5px;font-size:12px;'>" + place + "<br>" + phone + "</div>";
    var info=new kakao.maps.InfoWindow({ content:str });
    kakao.maps.event.addListener(marker, "mouseover", function() {
      info.open(map, marker);
    });
    kakao.maps.event.addListener(marker, "mouseout", function() {
      info.close(map, marker);
    });
  });
</script>
```


- 로컬 테이블에 검색 데이터 저장하기

- 저장할 항목을 전체선택 해지하는 기능을 추가한다.

```
[src]-[main]-[webapp]-[local] search.jsp
```

```
...
<div class="row justify-content-end mb-3">
  <form class="col-6 col-md-4" name="frm">
    <div class="input-group">
      <select class="form-select me-2" name="city">
        <option>서울</option>
        <option selected>인천</option>
        <option>경기</option>
        <option>부산</option>
      </select>
      <input class="form-control" name="query" value="버거킹">
      <button class="btn btn-primary">검색</button>
    </div>
  </form>
  <div id="div_local"></div>
</div>
<!-- 지역검색목록 템플릿 -->
<script id="temp_local" type="text/x-handlebars-template">
  <table class="table">
    <tr>
      <td colspan="6">
        <input type="checkbox" class="me-3" id="all">
        <button class="btn btn-primary btn-sm" id="btn-save">선택저장</button>
      <td>
    </td>
    </tr>
    <tr>
      <td colspan="6">
        {{#each documents}}
        <tr class="local">
          <td><input type="checkbox" class="chk"></td>
          ...
        </tr>
        {{/each}}
      </td>
    </tr>
  </table>
</script>
<script>
  ...
  //전체 체크박스를 클릭한 경우
  $("#div_local").on("click", "#all", function(){
    if($(this).is(":checked")) $("#div_local .chk").prop("checked", true);
    else $("#div_local .chk").prop("checked", false);
  });

  //각행의 체크박스를 클릭한 경우
  $("#div_local").on("click", ".chk", function(){
    let all=$("#div_local .chk").length; //전체 체크박스 갯수
    let chk=$("#div_local .chk:checked").length; //체크된 체크박스 갯수
    if(all==chk) $("#div_local #all").prop("checked", true);
    else $("#div_local #all").prop("checked", false);
  });

  function getList(){
    let query=$(frm.city).val() + $(frm.query).val();
    ...
  }
</script>
```

- 로컬 데이터 저장을 위한 VO, DAO, Controller를 작성한다.

local 테이블 생성

```
create table local(
    seq int auto_increment primary key,
    id varchar(20) not null,
    name varchar(200) not null,
    address varchar(500),
    phone varchar(30),
    url varchar(200),
    x varchar(100),
    y varchar(100),
    rdate datetime default now()
);
```

[src/main/java]-[domain] LocalVO.java

```
public class LocalVO {
    private int seq;
    private String id;
    private String name;
    private String address;
    private String phone;
    private String url;
    private String x;
    private String y;
    private Date rdate;

    //setter(), getter(), toString() 메서드를 generate한다.
}
```

[src/main/java]-[domain] LocalDAO.java

```
package model;
import java.sql.*;

public class LocalDAO {
    //검색한 로컬 데이터 저장
    public void insert(LocalVO vo) {
        try {
            String sql="select * from local where id=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, vo.getId());
            ResultSet rs=ps.executeQuery();
            if(!rs.next()) {
                sql="insert into local(id, name, address, phone, url, x, y) values(?, ?, ?, ?, ?, ?, ?)";
                ps=Database.CON.prepareStatement(sql);
                ps.setString(1, vo.getId());
                ps.setString(2, vo.getName());
                ps.setString(3, vo.getAddress());
                ps.setString(4, vo.getPhone());
                ps.setString(5, vo.getUrl());
                ps.setString(6, vo.getX());
                ps.setString(7, vo.getY());
                ps.execute();
            }
        } catch (Exception e) {
            System.out.println("로컬데이터저장:" + e.toString());
        }
    }
}
```

[src/main/java]-[controller] LocalController.java

```
@WebServlet(value={"/local/search", "/local/insert"})
public class LocalController extends HttpServlet {
    private static final long serialVersionUID = 1L;
    LocalDAO dao=new LocalDAO();

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setAttribute("pageName", "/local/search.jsp");
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
        dis.forward(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        switch(req.getServletPath()) {
            case "/local/insert":
                LocalVO vo=new LocalVO();
                vo.setId(req.getParameter("id"));
                vo.setName(req.getParameter("place_name"));
                vo.setAddress(req.getParameter("address_name"));
                vo.setPhone(req.getParameter("phone"));
                vo.setUrl(req.getParameter("place_url"));
                vo.setX(req.getParameter("x"));
                vo.setY(req.getParameter("y"));
                //System.out.println(vo.toString());
                dao.insert(vo);
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[local] search.jsp

```
<!-- 지역검색목록 템플릿 -->
<script id="temp_local" type="text/x-handlebars-template">
    <table class="table">
        <tr>
            <td colspan="6">
                <input type="checkbox" class="me-3" id="all">
                <button class="btn btn-primary btn-sm" id="btn-save">선택저장</button>
            <td>
        </tr>
        <tr>
            <td colspan="6">{{#each documents}}
            <tr class="local" local="{{toString @this}}">
                <td><input type="checkbox" class="chk"></td>
                <td>{{id}}</td>
                <td class="place">{{place_name}}</td>
                <td>{{address_name}}</td>
                <td class="phone">{{phone}}</td>
                <td><button class="btn btn-success btn-sm" x="{{x}}" y="{{y}}">위치</button></td>
            </tr>
            </td>
        </tr>
        </table>
    </script>
    <script>
        Handlebars.registerHelper("toString", function(local){
            return JSON.stringify(local);
        });
    </script>
```

[src]-[main]-[webapp]-[local] search.jsp

```
<script>
...
//선택 저장버튼을 클릭했을때
$("#div_local").on("click", "#btn-save", function(){
    let chk=$("#div_local .chk:checked").length;
    if(chk==0){
        alert("저장할 항목을 선택하세요!");
    }else{
        if(confirm(chk + "개 항목을 저장하실래요?")) {
            $("#div_local .chk:checked").each(function(){
                let row=$(this).parent().parent();
                let data=JSON.parse(row.attr("local"));
                //console.log(data);
                $.ajax({
                    type:"post",
                    url:"/local/insert",
                    data: data,
                    success:function(){ }
                });
            });
            alert("등록이 완료되었습니다.");
        }else {
            alert("등록이 취소되었습니다.");
        }
        $("#div_local .chk").prop("checked", false);
        $("#div_local #all").prop("checked", false);
    }
});
</script>
```

• 카카오 도서 검색 API

- 도서검색 페이지를 작성한다.

[src]-[main]-[webapp]-[book] search.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">도서검색</h1>
        <div class="row mb-3">
            <form name="frm" class="col-6 col-md-4 col-lg-3">
                <div class="input-group">
                    <input class="form-control" value="깜냥" name="query">
                    <button class="btn btn-primary">검색</button>
                </div>
            </form>
        </div>
        <hr>
        <div id="div_book"></div>
        <div class="text-center">
            <button class="btn btn-primary" id="prev">이전</button>
            <span class="mx-3" id="page">1/10</span>
            <button class="btn btn-primary" id="next">다음</button>
        </div>
    </div>
</div>
```

[src]-[main]-[webapp]-[book] search.jsp

```
<!-- 도서목록 템플릿 -->
<script id="temp_book" type="text/x-handlebars-template">
  <table class="table">
    <tr>
      <td colspan=5>
        <input type="checkbox" id="all" class="me-3">
        <button class="btn btn-primary btn-sm" id="btn-save">선택한항목 저장</button>
      </td>
    </tr>
    <tr>
      <td colspan=5>
        <div class="list-group">
          <div class="list-item">
            <div class="checkbox">
              <input type="checkbox" class="chk" book="{{toString @this}}"/>
            </div>
            <div class="img">
              
            </div>
            <div class="title">
              <div class="text">{{title}}</div>
            </div>
            <div class="price">
              <div class="text">{{fmtPrice price}}</div>
            </div>
            <div class="authors">
              <div class="text">{{authors}}</div>
            </div>
          </div>
        </div>
      </td>
    </tr>
  </table>
</script>
<script>
  Handlebars.registerHelper("fmtPrice", function(price){
    return price.toString().replace(/B(?=(\d{3})+(?!\d))/g, ",") + "원"
  });

  Handlebars.registerHelper("printImage", function(image){
    if(image) return image;
    else return "http://via.placeholder.com/120x174";
  });

  Handlebars.registerHelper("toString", function(book){
    return JSON.stringify(book);
  });
</script>
<script>
  let page=1;
  let query=$(frm.query).val();
  getList();
  function getList() {
    $.ajax({
      type:"get",
      url:"https://dapi.kakao.com/v3/search/book?target=title",
      data:{query:query, size:50, page:page},
      headers:{"Authorization":"KakaoAK b80880fbde422de3fd9b4a4e67c9bb54"},
      dataType:"json",
      success:function(data){
        //console.log(data);
        const temp=Handlebars.compile($("#temp_book").html());
        const html=temp(data);
        $("#div_book").html(html);
        let last = Math.ceil(data.meta.pageable_count/50);
        $("#page").html(page + "/" + last);
        if(page==1) $("#prev").attr("disabled",true) else $("#prev").attr("disabled",false);
        if(data.meta.is_end) $("#next").attr("disabled",true) else $("#next").attr("disabled",false);
      }
    });
  }
</script>
```

[src]-[main]-[webapp]-[book] search.jsp

```
<script>
    $(frm).on("submit", function(e){
        e.preventDefault();
        page=1;
        query=$(frm.query).val();
        getList();
    });

    $("#next").on("click", function(){
        page++;
        getList();
    });

    $("#prev").on("click", function(){
        page--;
        getList();
    });

    //전체 체크박스를 클릭한 경우
    $("#div_book").on("click", "#all", function(){
        if($(this).is(":checked")) $("#div_book .chk").prop("checked", true)
        else $("#div_book .chk").prop("checked", false);
    });

    //각행의 체크박스를 클릭한 경우
    $("#div_book").on("click", ".chk", function(){
        const all=$("#div_book .chk").length;
        const chk=$("#div_book .chk:checked").length;
        if(all==chk) $("#div_book #all").prop("checked", true)
        else $("#div_book #all").prop("checked", false);
    });
</script>
```

- 도서 이미지를 클릭하면 도서정보가 출력되는 모달창을 작성한다.

[src]-[main]-[webapp]-[book] search.jsp

```
<style>
    #div_book img {
        cursor: pointer;
    }
</style>
<!-- 도서목록 템플릿 -->
<script id="temp_book" type="text/x-handlebars-template">
    <table class="table">
        {{#each documents}}
            ...
            <jsp:include page="modal.jsp"/>
        {{/each}}
    </table>
</script>
<script>
    //각행의 이미지를 클릭한 경우
    $("#div_book").on("click", "img", function(){
        const index=$(this).attr("index");
        $("#modal" + index).modal("show");
    });
</script>
```

[src]-[main]-[webapp]-[book] modal.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!-- Modal -->
<div class="modal fade" id="modal{{@index}}" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1"
aria-labelledby="staticBackdropLabel" aria-hidden="true">
  <div class="modal-dialog modal-lg">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="staticBackdropLabel">도서정보</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <div class="row">
          <div class="col-4 card p-3 me-3">
            
          </div>
          <div class="col card p-3 me-3">
            <h5>제목: {{title}}</h5>
            <h5>가격: {{fmtPrice price}}</h5>
            <h5>출판사: {{publisher}}</h5>
            <h5>저자: {{authors}}</h5>
            <h5>ISBN: {{isbn}}</h5>
            <h5><a href="{{url}}">사이트이동</a></h5>
          </div>
        </div>
        <hr>
        <div>
          <p class="card p-3">{{contents}}</p>
        </div>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
```

- 도서정보 저장을 위한 VO, DAO, Controller를 작성한다.

[src/main/java]-[domain] BookVO.java

```
package model;
import java.util.Date;

public class BookVO {
    private int seq;
    private String title;
    private String thumbnail;
    private int price;
    private String pdate;
    private String authors;
    private String url;
    private String isbn;
    private String contents;
    private Date rdate;
    private String publisher;

    //setter(), getter(), toString() 메서드를 generate한다.
}
```

도서정보 테이블 생성

```
create table books(  
    seq int auto_increment primary key,  
    title varchar(200) not null,  
    thumbnail varchar(200),  
    price int default 0,  
    pdate varchar(20),  
    authors varchar(100),  
    url varchar(200),  
    isbn varchar(100),  
    contents text,  
    rdate datetime default now(),  
    publisher varchar(100)  
);
```

[src/main/java]-[domain] BookDAO.java

```
public class BookDAO {  
    public void insert(BookVO vo) {  
        try {  
            String sql="select * from books where isbn=?";  
            PreparedStatement ps1=Database.CON.prepareStatement(sql);  
            ps1.setString(1, vo.getIsbn());  
            ResultSet rs=ps1.executeQuery();  
            if(!rs.next()) {  
                String sql="insert into books(isbn,title,thumbnail,price,authors,url,contents,publisher) values(?,?,?,?,?,?,?,?)";  
                PreparedStatement ps=Database.CON.prepareStatement(sql);  
                ps.setString(1, vo.getIsbn());  
                ps.setString(2, vo.getTitle());  
                ...  
                ps.execute();  
            }  
        } catch (Exception e) {  
            System.out.println("도서등록:" + e.toString());  
        }  
    }  
}
```

[src]-[main]-[webapp]-[book] search.jsp

```
$("#div_book").on("click", "#btn-save", function(){ //선택 저장버튼을 클릭한 경우  
    const chk=$("#div_book .chk:checked").length;  
    if(chk==0){  
        alert("저장할 항목을 선택하세요!");  
    }else{  
        if(confirm("선택한 항목을 저장하실래요?")) {  
            $("#div_book .chk:checked").each(function(){ //선택한 항목 저장  
                const book=JSON.parse($(this).attr("book"));  
                $.ajax({  
                    type:"post",  
                    url:"/book/insert",  
                    data:book,  
                    success:function(){  
                        });  
                    });  
                alert("저장이 완료되었습니다.");  
            }else{  
                alert("저장이 취소되었습니다.")  
            }  
            $("#div_book .chk").prop("checked", false);  
            $("#div_book #all").prop("checked", false);  
        }  
    }  
});
```


- 학사관리 프로그램

- 교수관리 프로그램

- 교수목록 출력을 위한 DAO, Controller, JSP 페이지를 작성한다.

```
[src/main/java]-[domain] ProVO.java
```

```
public class ProVO {
    private String pcode;
    private String pname;
    private String dept;
    private String hiredate;
    private String title;
    private int salary;

    //setter(), getter(), toString() 메서드를 generate한다.
}
```

```
[src/main/java]-[domain] ProDAO.java
```

```
public class ProDAO {
    public ArrayList<ProVO> list(int page, String query, String key) { //교수목록
        ArrayList<ProVO> array=new ArrayList<ProVO>();
        try {
            String sql = "select * from professors where " + key + " like ? limit ?, 5";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, "%" + query + "%");
            ps.setInt(2, (page-1)*5);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                ProVO vo=new ProVO();
                vo.setPcode(rs.getString("pcode"));
                vo.setPname(rs.getString("pname"));
                vo.setDept(rs.getString("dept"));
                vo.setHiredate(rs.getString("hiredate"));
                vo.setTitle(rs.getString("title"));
                vo.setSalary(rs.getInt("salary"));
                array.add(vo);
            }
        }catch(Exception e) {
            System.out.println("교수목록:" + e.toString());
        }
        return array;
    }

    public int total(String query, String key) { //검색수
        int total=0;
        try {
            String sql ="select count(*) total from professors where " + key + " like ?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, "%" + query + "%");
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                total=rs.getInt("total");
            }
        }catch(Exception e) {
            System.out.println("검색수:" + e.toString());
        }
        return total;
    }
}
```

[src/main/java]-[controller] ProfessorController.java

```
@WebServlet(value={"/pro/list", "/pro/list.json", "/pro/total"})
public class ProfessorController extends HttpServlet {
    DecimalFormat df=new DecimalFormat("#,###월");
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
        switch(request.getServletPath()) {
            case "/pro/list":
                request.setAttribute("pageName", "/pro/list.jsp");
                dis.forward(request, response);
                break;
            case "/pro/list.json":
                String key=request.getParameter("key");
                String query=request.getParameter("query");
                int page=Integer.parseInt(request.getParameter("page"));
                JSONArray jArray=new JSONArray();
                for(ProVO vo:dao.list(page, query, key)) {
                    JSONObject obj=new JSONObject();
                    obj.put("pcode", vo.getPcode());
                    obj.put("pname", vo.getPname());
                    obj.put("dept", vo.getDept());
                    obj.put("hiredate", vo.getHiredate());
                    obj.put("salary", df.format(vo.getSalary()));
                    obj.put("title", vo.getTitle());
                    jArray.add(obj);
                }
                out.print(jArray);
                break;
            case "/pro/total":
                key=request.getParameter("key");
                query=request.getParameter("query");
                out.print(dao.total(query, key));
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[pro] list.jsp ①

```
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">교수목록</h1>
        <div class="row">
            <form class="col-6" name="frm">
                <div class="input-group">
                    <select class="form-select" name="key">
                        <option value="pcode">교수번호</option>
                        <option value="pname" selected>교수이름</option>
                        <option value="dept">교수학과</option>
                    </select>&nbsp;  
                    <input class="form-control" placeholder="검색어" name="query">
                    <input type="submit" value="검색" class="btn btn-primary">
                </div>
            </form>
            <div id="div_pro"></div>
            <div id="pagination" class="pagination justify-content-center mt-5"></div>
        </div>
    </div>
</div>
```

```

<script id="temp_pro" type="text/x-handlebars-template">
  <table class="table">
    {{#each .}}
      <tr>
        ...
      </tr>
    {{/each}}
  </table>
</script>
<script>
  let page=1;
  let query="";
  let key=$(frm.key).val();

  $(frm).on("submit", function(e) {
    e.preventDefault();
    key=$(frm.key).val();
    query=$(frm.query).val();
    page=1;
    getTotal();
  });

  getTotal();
  function getTotal() {
    $.ajax({
      type:"get",
      url:"/pro/total",
      data:{key:key, query:query},
      success:function(data){
        if(data==0){
          $("#div_product").html("<h3 class='text-center my-5'>검색내용이 존재하지 않습니다.</h3>");
        }else{
          const totalPages=Math.ceil(data/5);
          $("#pagination").twbsPagination("changeTotalPages", totalPages, page);
        }
        if(data > 5){
          $("#pagination").show();
        }else{
          $("#pagination").hide();
        }
      }
    });
  }

  function getList() {
    $.ajax({
      type:"get",
      url: "/pro/list.json",
      data: {page: page, key:key, query:query},
      dataType:"json",
      success:function(data){ ... }
    });
  }

  //페이지네이션 출력
  $('#pagination').twbsPagination({ ... });
</script>

```

- 교수등록을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src]-[main]-[webapp]-[pro] insert.jsp

```
<form name="frm1">
  <div class="input-group mb-2">
    <span class="input-group-text">교수이름</span>
    <input name="pname" class="form-control">
  </div>
  <div class="input-group mb-2">
    <span class="input-group-text">교수학과</span>
    <select name="dept" class="form-select">
      <option value="전산">전자계산학과</option>
      <option value="전자">전자공학</option>
      <option value="건축">건축공학과</option>
    </select>
  </div>
  <div class="input-group mb-2 pt-2">
    <span class="input-group-text">교수직급</span>
    <input type="radio" name="title" value="정교수" class="m-2"><span class="p-2">정교수</span>
    <input type="radio" name="title" value="부교수" class="m-2" checked=""><span class="p-2">부교수</span>
    <input type="radio" name="title" value="정교수" class="m-2"><span class="p-2">조교수</span>
  </div>
  <div class="input-group mb-2">
    <span class="input-group-text">교수급여</span>
    <input name="salary" class="form-control" value="0">
  </div>
  <div class="input-group mb-2">
    <span class="input-group-text">임용일자</span>
    <input name="hiredate" class="form-control" type="date">
  </div>
  <div class="text-center mt-5">
    <input type="submit" value="교수등록" class="btn btn-primary">
    <input type="reset" value="등록취소" class="btn btn-secondary">
  </div>
</form>
<script>
  $(frm1).on("submit", function(e){
    e.preventDefault();
    const pname=$(frm1.pname).val();
    const salary=$(frm1.salary).val();
    const title=$(frm1.title).val();
    const dept=$(frm1.dept).val();
    const hiredate=$(frm1.hiredate).val();
    if(pname==""){
      alert("교수이름을 입력하세요!");
      $(frm1.pname).focus();
    }else if(salary.replace(/[0-9]/g, "")){
      alert("급여를 숫자로 입력하세요!");
      $(frm1.salary).val("");
      $(frm1.salary).focus();
    }else{
      if(confirm("새로운 교수를 등록하실래요?")) {
        $.ajax({
          type:"post",
          url:"/pro/insert",
          data:{pname:pname, salary:salary, title:title, dept:dept, hiredate:hiredate},
          success:function(){
            getTotal();
            $("#modal_insert").modal("hide");
          }
        });
      }
    }
  });
</script>
```

[src]-[main]-[webapp]-[pro] list.jsp

```
<!-- 교수등록 Modal -->
<div class="modal fade" id="modal-insert" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1"
aria-labelledby="staticBackdropLabel" aria-hidden="true">
  <div class="modal_dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="staticBackdropLabel">교수등록</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <jsp:include page="/pro/insert.jsp"/>
      </div>
    </div>
  </div>
</div>
</div>
```

[src/main/java]-[model] ProDAO.java

```
public class ProDAO {
    //교수등록
    public void insert(ProVO vo) {
        try {
            String ncode="";
            String sql="select max(pcode)+1 ncode from professors";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ResultSet rs=ps.executeQuery();
            if(rs.next()) ncode=rs.getString("ncode");
            sql="insert into professors(pcode, pname, dept, title, salary, hiredate) values(?, ?, ?, ?, ?, ?)";
            ps=Database.CON.prepareStatement(sql);
            ps.setString(1, ncode);
            ps.setString(2, vo.getPname());
            ps.setString(3, vo.getDept());
            ps.setString(4, vo.getTitle());
            ps.setInt(5, vo.getSalary());
            ps.setString(6, vo.getHiredate());
            ps.execute();
        } catch (Exception e) {
            System.out.println("교수등록:" + e.toString());
        }
    }
}
```

[src/main/java]-[controller] ProfessorController.java

```
@WebServlet(value={"/pro/list", "/pro/list.json", "/pro/total", "/pro/insert"})
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    switch(request.getServletPath()) {
        case "/pro/insert":
            ProVO vo=new ProVO();
            vo.setPname(request.getParameter("pname"));
            vo.setDept(request.getParameter("dept"));
            vo.setSalary(Integer.parseInt(request.getParameter("salary")));
            vo.setTitle(request.getParameter("title"));
            vo.setHiredate(request.getParameter("hiredate"));
            dao.insert(vo);
            break;
    }
}
```

- 교수수정을 위한 DAO, Controller, JSP 페이지를 작성한다.

```
[src]-[main]-[webapp]-[pro] list.jsp
```

```
<!-- 교수목록 템플릿-->
<script id="temp_pro" type="text/x-handlebars-template">
  <table class="table">
    <tr>
      <td><a href="/pro/update?pcode={{pcode}}">{{pcode}}</a></td>
      ...
    </tr>
  </table>
</script>
```

```
[src]-[main]-[webapp]-[pro] update.jsp
```

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<div class="row my-5">
  <div class="col">
    <h1 class="text-center mb-5">교수정보수정</h1>
    <form name="frm1" method="post" action="/pro/update">
      <div class="input-group mb-2">
        <span class="input-group-text">교수번호</span>
        <input name="pcode" class="form-control" value="{{vo.pcode}}" readonly>
      </div>
      ...
      <div class="input-group mb-2">
        <span class="input-group-text">교수학과</span>
        <select name="dept" class="form-select">
          <option value="전산" <c:out value="{{vo.dept=='전산' ? 'selected':''}}"/>전자계산학과</option>
          ...
        </select>
      </div>
      <div class="input-group mb-2 pt-2">
        <span class="input-group-text">교수직급</span>
        <input type="radio" name="title" value="정교수" <c:out value="{{vo.title=='정교수' ? 'checked':''}}"/> class="m-2">
        ...
      </div>
      ...
      <div class="input-group mb-2">
        <span class="input-group-text">임용일자</span>
        <input name="hiredate" value="{{vo.hiredate}}" class="form-control" type="date">
      </div>
      <div class="text-center mt-5">
        <input type="submit" value="정보수정" class="btn btn-primary">
        <input type="reset" value="수정취소" class="btn btn-secondary">
      </div>
    </form>
  </div>
</div>
<script>
  $(frm1).on("submit", function(e){
    e.preventDefault();
    if(confirm("교수 정보를 수정하실래요?")) frm1.submit();
  })
</script>
```

[src/main/java]-[model] ProDAO.java

```
public class ProDAO {
    public void update(ProVO vo) { //교수정보
        try {
            String sql= "update professors set pname=?,dept=?,title=?,salary=?,hiredate=? where pcode=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, vo.getPname());
            ...
            ps.execute();
        }catch(Exception e) { System.out.println("교수수정:" + e.toString()); }
    }

    public ProVO read(String pcode) { //교수정보
        ProVO vo=new ProVO();
        try {
            String sql ="select * from professors where pcode=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, pcode);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) { ... }
        }catch(Exception e) { System.out.println("교수정보:" + e.toString()); }
        return vo;
    }
}
```

[src/main/java]-[controller] ProfessorController.java

```
@WebServlet(value={"/pro/list", "/pro/list.json", "/pro/total", "/pro/insert", "/pro/update"})
public class ProfessorsController extends HttpServlet {
    ...
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            ...
            case "/pro/update":
                String pcode=request.getParameter("pcode");
                request.setAttribute("vo", dao.read(pcode));
                request.setAttribute("pageName", "/pro/update.jsp");
                dis.forward(request, response);
                break;
        }
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        switch(request.getServletPath()) {
            ...
            case "/pro/update":
                vo=new ProVO();
                vo.setPcode(request.getParameter("pcode"));
                vo.setPname(request.getParameter("pname"));
                vo.setDept(request.getParameter("dept"));
                vo.setSalary(Integer.parseInt(request.getParameter("salary")));
                vo.setTitle(request.getParameter("title"));
                vo.setHiredate(request.getParameter("hiredate"));
                System.out.println(vo.toString());
                dao.update(vo);
                response.sendRedirect("/pro/list");
                break;
        }
    }
}
```

- 학생관리 프로그램

- 학생목록 출력을 위한 DAO, Controller, JSP 페이지를 작성한다.

```
[src/main/java]-[model] StuVO.java
```

```
public class StuVO extends ProVO{
    private String scode;
    private String sname;
    private String dept;
    private int year;
    private String birthday;
    private String advisor;
    //setter(), getter(), toString() 메서드를 generate한다.
}
```

```
[src/main/java]-[model] StuDAO.java
```

```
public class StuDAO {
    //학생목록
    public ArrayList<StuVO> list(int page, String query, String key){
        ArrayList<StuVO> array=new ArrayList<StuVO>();
        try {
            String sql= "select * from view_stu where " + key + " like ? limit ?, 5";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, "%" + query + "%");
            ps.setInt(2, (page-1)*5);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                StuVO vo=new StuVO();
                vo.setScode(rs.getString("scode"));
                vo.setSname(rs.getString("sname"));
                vo.setDept(rs.getString("dept"));
                vo.setYear(rs.getInt("year"));
                vo.setBirthday(rs.getString("birthday"));
                vo.setAdvisor(rs.getString("advisor"));
                vo.setPname(rs.getString("pname"));
                array.add(vo);
            }
        }catch(Exception e) {
            System.out.println("학생목록:" + e.toString());
        }
        return array;
    }

    //학생검색수
    public int total(String query, String key) {
        int total=0;
        try {
            String sql= " select count(*) cnt from view_stu where " + key + " like ?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, "%" + query + "%");
            ResultSet rs=ps.executeQuery();
            if(rs.next()) total=rs.getInt("cnt");
        }catch(Exception e) {
            System.out.println("학생검색수:" + e.toString());
        }
        return total;
    }
}
```


[src/main/java]-[controller] StudentController.java

```
@WebServlet(value={"/stu/list", "/stu/list.json", "/stu/total"})
public class StudentController extends HttpServlet {
    StuDAO dao=new StuDAO();

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");

        switch(request.getServletPath()) {
            case "/stu/list":
                request.setAttribute("pageName", "/stu/list.jsp");
                dis.forward(request, response);
                break;
            case "/stu/list.json": //주소입력 stu/list.json?page=1&key=pname&query=이병철
                int page=Integer.parseInt(request.getParameter("page"));
                String query=request.getParameter("query");
                String key=request.getParameter("key");
                ArrayList<StuVO> array=dao.list(page, query, key);
                JSONArray jArray=new JSONArray(); //자동 import ctrl + shift + o
                for(StuVO vo:array) {
                    JSONObject obj=new JSONObject();
                    ...
                    obj.put("pname", vo.getPname());
                    jArray.add(obj);
                }
                out.println(jArray);
                break;
            case "/stu/total": //주소입력 stu/total?query=4&key=year
                query=request.getParameter("query");
                key=request.getParameter("key");
                out.println(dao.total(query, key));
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[stu] list.jsp ①

```
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">학생목록</h1>
        <div class="row">
            <form class="col-6" name="frm">
                <div class="input-group">
                    <select class="form-select" name="key">
                        <option value="scode">학생번호</option>
                        <option value="sname" selected>학생이름</option>
                    </select>&nbsp;  
                    <input class="form-control" placeholder="검색어" name="query">
                    <input type="submit" value="검색" class="btn btn-primary">
                </div>
            </form>
        </div>
        <div id="div_stu"></div>
        <div id="pagination" class="pagination justify-content-center mt-5"></div>
    </div>
</div>
```

[src]-[main]-[webapp]-[stu] list.jsp ②

```
<!-- 학생목록 템플릿-->
<script id="temp_stu" type="text/x-handlebars-template">
    <table class="table">
        {{#each .}}
            <tr class="stu"> ... </tr>
        {{/each}}
    </table>
</script>
<script>
    let query="";
    let key=$(frm.key).val();

    $(frm).on("submit", function(e){
        e.preventDefault();
        key=$(frm.key).val();
        query=$(frm.query).val();
        page=1;
        getTotal();
    });

    getTotal();
    function getTotal(){
        $.ajax({
            type:"get",
            url:"/stu/total",
            ...
        });
    }

    function getList(page){
        $.ajax({
            type:"get",
            url: "/stu/list.json",
            ...
        });
    }
</script>
```

- 학생등록을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] StuDAO.java

```
public void insert(StuVO vo) { //학생등록
    try {
        String ncode="";
        String sql="select max(scode) +1 ncode from students";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ResultSet rs=ps.executeQuery();
        if(rs.next()) ncode=rs.getString("ncode");
        sql="insert into students(scode,sname,dept,year,birthday,advisor) values(?,?,?,?,?,?)";
        ps=Database.CON.prepareStatement(sql);
        ps.setString(1, ncode);
        ...
        ps.execute();
    } catch (Exception e) {
        System.out.println("학생등록:" + e.toString());
    }
}
```

```
[src/main/java]-[controller] StudentController.java
```

```
@WebServlet(value={"/stu/list", "/stu/list.json", "/stu/total", "/stu/insert"})
public class StudentsController extends HttpServlet {
    StuDAO dao=new StuDAO();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        ...
        switch(request.getServletPath()) {
        case "/stu/list":
            request.setAttribute("parray", pdao.all());
            request.setAttribute("pageName", "/stu/list.jsp");
            dis.forward(request, response);
            break;
        }
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        switch(request.getServletPath()) {
        case "/stu/insert":
            StuVO vo=new StuVO();
            vo.setName(request.getParameter("sname"));
            vo.setDept(request.getParameter("dept"));
            vo.setBirthday(request.getParameter("birthday"));
            vo.setYear(Integer.parseInt(request.getParameter("year")));
            vo.setAdvisor(request.getParameter("advisor"));
            //System.out.println(vo.toString());
            dao.insert(vo);
            response.sendRedirect("/stu/list");
            break;
        }
    }
}
```

```
[src]-[main]-[webapp]-[stu] list.jsp
```

```
<div class="row">
    <form class="col-6" name="frm">...</form>
    <div class="col text-end">
        <button class="btn btn-primary" id="btn-insert">학생등록</button>
    </div>
</div>
<!-- 학생등록 Modal -->
<div class="modal fade" id="modal-insert" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header">
                <h1 class="modal-title fs-5" id="staticBackdropLabel">학생등록</h1>
                <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <div class="modal-body">
                <jsp:include page="insert.jsp"/>
            </div>
        </div>
    </div>
</div>
</div>
<script>
    ...
    $("#btn-insert").on("click", function(){
        $("#modal-insert").modal("show");
    });
</script>
```

[src]-[main]-[webapp]-[stu] insert.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<form name="frm1" method="post" action="/stu/insert">
    <div class="input-group mb-3">
        <span class="input-group-text">학생이름</span>
        <input name="sname" class="form-control">
    </div>
    <div class="input-group mb-3">
        <span class="input-group-text">학생학과</span>
        <select name="dept" class="form-select">
            <option value="전산">전자계산학과</option>
            <option value="전자">전자공학</option>
            <option value="건축">건축공학과</option>
        </select>
    </div>
    <div class="input-group mb-3">
        <span class="input-group-text">생년월일</span>
        <input name="birthday" class="form-control" type="date" value="2000-01-01">
    </div>
    <div class="input-group mb-3">
        <span class="input-group-text">학생학년</span>
        <input name="year" type="radio" value="1" class="ms-2" checked>
        <span class="p-2">1학년</span>
        <input name="year" type="radio" value="2" class="ms-2">
        <span class="p-2">2학년</span>
        <input name="year" type="radio" value="3" class="ms-2">
        <span class="p-2">3학년</span>
        <input name="year" type="radio" value="4" class="ms-2">
        <span class="p-2">4학년</span>
    </div>
    <div class="input-group mb-3">
        <span class="input-group-text">지도교수</span>
        <select name="advisor" class="form-select">
            <c:forEach items="${parray}" var="pro">
                <option value="${pro.pcode}">${pro.pname}:${pro.dept}</option>
            </c:forEach>
        </select>
    </div>
    <div class="text-center">
        <input type="submit" value="학생등록" class="btn btn-primary">
        <input type="reset" value="등록취소" class="btn btn-secondary">
    </div>
</form>

<script>
    $(frm1).on("submit", function(e){
        e.preventDefault();
        const sname=$(frm1.sname).val();
        if(sname=="") {
            alert("학생이름을 입력하세요!");
            $(frm1.sname).focus();
        }else{
            if(confirm("새로운 학생을 등록하실래요?")){
                frm1.submit();
            }
        }
    });
</script>
```

- 학생정보 수정을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] StuDAO.java

```
public class StuDAO {
    ...
    public StuVO read(String scode) { //학생정보
        StuVO vo=new StuVO();
        try {
            String sql= " select * from view_stu where scode=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, scode);
            ResultSet rs=ps.executeQuery();
            if(rs.next()) {
                ...
            }
        }catch(Exception e) {
            System.out.println("학생정보:" + e.toString());
        }
        return vo;
    }

    public void update(StuVO vo) { //학생수정
        try {
            String sql="update students set sname=?,dept=?,year=?,birthday=?,advisor=? where scode=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ...
            ps.execute();
        }catch(Exception e) {
            System.out.println("학생수정:" + e.toString());
        }
    }
}
```

[src/main/java]-[controller] StudentController.java

```
@WebServlet(value={"/stu/list", "/stu/list.json", "/stu/total", "/stu/insert", "/stu/update"})
public class StudentsController extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            case "/stu/update":
                request.setAttribute("vo", dao.read(request.getParameter("scode")));
                request.setAttribute("parray", pdao.all());
                request.setAttribute("pageName", "/stu/update.jsp");
                dis.forward(request, response);
                break;
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        switch(request.getServletPath()) {
            case "/stu/update":
                vo=new StuVO();
                vo.setScode(request.getParameter("scode"));
                ...
                dao.update(vo);
                response.sendRedirect("/stu/list");
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[stu] list.jsp

```
<!-- 학생목록 템플릿-->
<script id="temp_stu" type="text/x-handlebars-template">
  <table class="table">
    {{#each .}}
    <tr>
      <td><a href="/stu/update?scode={{scode}}">{{scode}}</a></td>
      ...
    </tr>
    {{/each}}
  </table>
</script>
```

[src]-[main]-[webapp]-[stu] update.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<div class="row my-5">
  <div class="col">
    <h1 class="text-center mb-5">학생정보수정</h1>
    <form name="frm1" method="post" action="/stu/update">
      ...
      <div class="input-group mb-3">
        <span class="input-group-text">학생학과</span>
        <select name="dept" class="form-select">
          <option value="전산" <c:out value="$ {vo.dept=='전산'? 'selected':''}" /> >전자계산학과</option>
          ...
        </select>
      </div>
      ...
      <div class="input-group mb-3">
        <span class="input-group-text">학생학년</span>
        <input name="year" type="radio" value="1" class="ms-2" <c:out value="$ {vo.year==1?'checked':''}" />
        ...
        <span class="p-2">4학년</span>
      </div>
      <div class="input-group mb-3">
        <span class="input-group-text">지도교수</span>
        <select name="advisor" class="form-select">
          <c:forEach items="$ {parray}" var="v">
            <option value="$ {v.pcode}" <c:out value="$ {vo.advisor==v.pcode?'selected':''}" />>$ {v.pname}</option>
          </c:forEach>
        </select>
      </div>
      <div class="text-center">
        <input type="submit" value="정보수정" class="btn btn-primary">
        <input type="reset" value="수정취소" class="btn btn-secondary">
      </div>
    </form>
  </div>
</div>
<script>
  $(frm1).on("submit", function(e){
    e.preventDefault();
    if(confirm("학생정보를 수정하실래요?")) {
      frm1.submit();
    }
  });
</script>
```

- 강좌관리 프로그램

- 강좌목록 출력을 위한 DAO, Controller, JSP 페이지를 작성한다.

```
[src/main/java]-[model] CouVO.java
```

```
public class CourseVO extends ProVO{
    private String lcode;
    private String lname;
    private int hours;
    private String room;
    private String instructor;
    private int capacity; //수강최대인원
    private int persons; //수강인원
    //setter(), getter(), toString() 메서드를 generate한다.
}
```

```
[src/main/java]-[model] CouDAO.java
```

```
public class CouDAO {
    //강좌목록
    public ArrayList<CourseVO> list(int page, String key, String query){
        ArrayList<CourseVO> array=new ArrayList<CourseVO>();
        try {
            String sql= "select * from view_cou where " + key + " like ? limit ?, 5";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, "%" + query + "%");
            ps.setInt(2, (page-1)*5);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                CourseVO vo=new CourseVO();
                ...
                vo.setCapacity(rs.getInt("capacity"));
                vo.setPersons(rs.getInt("persons"));
                vo.setPname(rs.getString("pname"));
                array.add(vo);
            }
        }catch(Exception e) {
            System.out.println("강좌목록:" + e.toString());
        }
        return array;
    }

    //검색수
    public int total(String key, String query) {
        int total=0;
        try {
            String sql= "select count(*) cnt from view_cou where " + key + " like ? ";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, "%" + query + "%");
            ResultSet rs=ps.executeQuery();
            if(rs.next()) {
                total=rs.getInt("cnt");
            }
        }catch(Exception e) {
            System.out.println("검색수:" + e.toString());
        }
        return total;
    }
}
```

[src/main/java]-[controller] CourseController.java

```
@WebServlet(value={"/cou/list", "/cou/total", "/cou/list.json"})
public class CourseController extends HttpServlet {
    CouDAO dao=new CouDAO();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
        switch(request.getServletPath()) {
            case "/cou/list":
                request.setAttribute("pageName", "/cou/list.jsp");
                dis.forward(request, response);
                break;
            case "/cou/total": //브라우저 주소입력에 /cou/total?key=pname&query=이
                String key=request.getParameter("key");
                String query=request.getParameter("query");
                out.print(dao.total(key, query));
                break;
            case "/cou/list.json": //브라우저 주소입력에 /cou/list.json?key=pname&query=이&page=1
                int page=Integer.parseInt(request.getParameter("page"));
                key=request.getParameter("key");
                query=request.getParameter("query");
                Gson gson=new Gson();
                out.println(gson.toJson(dao.list(page, key, query)));
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[cou] list.jsp

```
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">강좌목록</h1>
        <div class="row">
            <form name="frm" class="col-6">...</form>
        </div>
        <div id="div_cou"></div>
        <div id="pagination" class="pagination justify-content-center mt-3"></div>
    </div>
</div>
<!-- 강좌목록 템플릿 -->
<script id="temp_cou" type="text/x-handlebar-template">
    ...
</script>
<script>
    let key=$(frm.key).val();
    let query=$(frm.query).val();
    $(frm).on("submit", function(e){
        ...
        getTotal();
    });
    function getList(){ ... }

    getTotal();
    function getTotal(){ ... }

    $('#pagination').twbsPagination({ ... });
</script>
```


- 강좌입력을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] CouDAO.java

```
public class CouDAO {
    public String getCode() { //새로운 코드구하기
        String code="";
        try {
            String mcode="";
            String sql="select max(lcode) mcode from courses";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ResultSet rs=ps.executeQuery();
            if(rs.next()) mcode=rs.getString("mcode");
            code="N" + (Integer.parseInt(mcode.substring(1))+1);
        }catch(Exception e) {
            System.out.println("강좌등록:" + e.toString());
        }
        return code;
    }

    public void insert(CourseVO vo) { //강좌등록
        try {
            String code=getCode();
            String sql="insert into courses(lcode,lname,room,hours,capacity,persons,instructor) values(?,?,?,?,?,?,?)";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, code);
            ...
            ps.execute();
        }catch(Exception e) {
            System.out.println("강좌등록:" + e.toString());
        }
    }
}
```

src/main/java]-[controller] CourseController.java

```
@WebServlet(value={"/cou/list", "/cou/total", "/cou/list.json", "/cou/insert"})
public class CourseController extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        ...
        switch(request.getServletPath()) {
            case "/cou/list":
                request.setAttribute("parray", pdao.all());
                request.setAttribute("pageName", "/cou/list.jsp");
                dis.forward(request, response);
                break;
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        switch(request.getServletPath()) {
            case "/cou/insert":
                CourseVO vo=new CourseVO();
                vo.setLname(request.getParameter("lname"));
                ...
                dao.insert(vo);
                response.sendRedirect("/cou/list");
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[cou] list.jsp

```
<div class="row">
  <form name="frm" class="col-6">...</form>
  <div class="col text-end">
    <button class="btn btn-primary" id="btn-insert">강좌등록</button>
  </div>
</div>
<!-- 강좌입력 Modal -->
<div class="modal fade" id="modal-insert" data-bs-backdrop="static" data-bs-keyboard="false" tabindex="-1">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h1 class="modal-title fs-5" id="staticBackdropLabel">강좌등록</h1>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <jsp:include page="insert.jsp"/>
      </div>
    </div>
  </div>
</div>
</div>
<script>
  ...
  $("#btn-insert").on("click", function(){
    $("#modal-insert").modal("show");
  });
</script>
```

[src]-[main]-[webapp]-[cou] inser.jsp

```
<form name="frm1" action="/cou/insert" method="post">
  ...
  <span class="input-group-text">강의시수</span>
  <input name="hours" class="form-control" value="3" oninput="isNumber(this)">
  ...
  <span class="input-group-text">담당교수</span>
  <select name="instructor" class="form-select">
    <c:forEach items="${parray}" var="p">
      <option value="${p.pcode}">${p.pcode}:${p.pname}:${p.dept}</option>
    </c:forEach>
  </select>
</form>
<script>
  $(frm1).on("submit", function(e){
    e.preventDefault();
    const lname=$(frm1.lname).val();
    if(lname==""){
      alert("강좌이름을 입력하세요!");
      $(frm1.lname).focus();
    }else{
      if(confirm("새로운 강좌를 입력하실래요?")) frm1.submit();
    }
  });

  //숫자인 경우에만 입력
  function isNumber(item){
    item.value=item.value.replace(/[^0-9]/g,'');
  }
</script>
```

- 강좌정보 수정을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] CouDAO.java

```
public class CouDAO {
    public CourseVO read(String lcode) { //강의정보
        CourseVO vo=new CourseVO();
        try {
            String sql= "select * from view_cou where lcode=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, lcode);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                ...
            }
        }catch(Exception e) {
            System.out.println("강좌정보:" + e.toString());
        }
        return vo;
    }
    public void update(CourseVO vo) { //강좌수정
        try {
            String sql="update courses set lname=?,room=?,hours=?,capacity=?,persons=?,instructor=? where lcode=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ...
            ps.execute();
        }catch(Exception e) {
            System.out.println("강좌수정:" + e.toString());
        }
    }
}
```

[src/main/java]-[controller] CourseController.java

```
@WebServlet(value={"/cou/list", "/cou/total", "/cou/list.json", "/cou/insert", "/cou/update"})
public class CourseController extends HttpServlet {
    CouDAO dao=new CouDAO();
    ProDAO pdao=new ProDAO();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            case "/cou/update":
                request.setAttribute("vo", dao.read(request.getParameter("lcode")));
                request.setAttribute("parray", pdao.all());
                request.setAttribute("pageName", "/cou/update.jsp");
                dis.forward(request, response);
                break;
        }
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        switch(request.getServletPath()) {
            case "/cou/update":
                vo=new CourseVO();
                ...
                System.out.println(vo.toString());
                dao.update(vo);
                response.sendRedirect("/cou/list");
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[cou] list.jsp

```
<!-- 강좌목록 템플릿 -->
<script id="temp_cou" type="text/x-handlebar-template">
  <table class="table">
    {{#each .}}
    <tr class="cou" lcode="{{lcode}}">
      <td class="lcode"><a href="/cou/update?lcode={{lcode}}">{{lcode}}</a></td>
      ...
    </tr>
    {{/each}}
  </table>
</script>
```

[src]-[main]-[webapp]-[cou] update.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<div class="row my-5">
  <div class="col">
    <h1 class="text-center mb-5">강좌정보수정</h1>
    <form name="frm1" action="/cou/update" method="post">
      ...
      <div class="input-group mb-3">
        <span class="input-group-text">강의시수</span>
        <input name="hours" class="form-control" value="{{vo.hours}}" oninput="isNumber(this)">
      </div>
      <div class="input-group mb-3">
        <span class="input-group-text">최대인원</span>
        <input name="capacity" class="form-control" value="{{vo.capacity}}" oninput="isNumber(this)">
      </div>
      <div class="input-group mb-3">
        <span class="input-group-text">수강인원</span>
        <input name="persons" class="form-control" value="{{vo.persons}}" oninput="isNumber(this)">
      </div>
      <div class="input-group mb-3">
        <span class="input-group-text">담당교수</span>
        <select name="instructor" class="form-select">
          <c:forEach items="{{parray}}" var="p">
            <option value="{{p.pcode}}" <c:out value="{{p.pcode==vo.instructor ? 'selected':''}}"/>
              {{p.pcode}}:{{p.pname}}:{{p.dept}}
            </option>
          </c:forEach>
        </select>
      </div>
      <div class="text-center">
        <input type="submit" value="정보수정" class="btn btn-primary">
        <input type="reset" value="수정취소" class="btn btn-secondary">
      </div>
    </form>
  </div>
</div>
<script>
  ${{frm1}}.on("submit", function(e){
    e.preventDefault();
    if(confirm("강좌 정보를 수정하실래요?")) frm1.submit();
  });
  ...
</script>
```

- 수강신청 및 취소

- 수강신청목록 출력을 위한 DAO, Controller, JSP 페이지를 작성한다.

```
[src/main/java]-[model] EnrollVO.java
```

```
public class EnrollVO extends CourseVO{
    private String scode;
    private String edate;
    private int grade;
    //setter(), getter(), toString() 메서드를 generate한다.
}
```

```
[src/main/java]-[model] StuDAO.java
```

```
public class StuDAO {
    ...
    //수강신청목록
    public ArrayList<EnrollVO> list(String scode){
        ArrayList<EnrollVO> array=new ArrayList<EnrollVO>();
        try {
            String sql="select * from view_enroll_cou where scode=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, scode);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                EnrollVO vo=new EnrollVO();
                ...
                array.add(vo);
            }
        }catch(Exception e) {
            System.out.println("수강신청목록:" + e.toString());
        }
        return array;
    }
}
```

```
[src/main/java]-[controller] StudentController.java
```

```
@WebServlet(value={"/stu/list", "/stu/list.json", "/stu/total", "/enroll/delete", "/stu/insert", "/stu/update", "/stu/enroll.json"})
public class StudentsController extends HttpServlet {
    StuDAO dao=new StuDAO();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        ...
        switch(request.getServletPath()) {
        case "/stu/enroll.json": //주소줄입력 /stu/enroll.json?scode=92514023
            ArrayList<EnrollVO> earray=dao.list(request.getParameter("scode"));
            JSONArray = new JSONArray();
            for(EnrollVO vo:earray) {
                JSONObject obj=new JSONObject();
                obj.put("lname", vo.getLname());
                ...
                JSONArray.add(obj);
            }
            out.print(JSONArray);
            break;
        }
    }
}
```

[src/main/java]-[controller] StudentController.java

```
@WebServlet(value={"/stu/list", "/stu/list.json", "/stu/total", "/enroll/delete", "/stu/insert", "/stu/update"... "/stu/enroll" })
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    switch(request.getServletPath()) {
        ...
        case "/stu/enroll":
            request.setAttribute("vo", dao.read(request.getParameter("scode")));
            request.setAttribute("pageName", "/stu/enroll.jsp");
            dis.forward(request, response);
            break;
    }
}
```

[src]-[main]-[webapp]-[stu] list.jsp

```
<!-- 학생목록 템플릿-->
<script id="temp_stu" type="text/x-handlebars-template">
    <table class="table">
        {{#each .}}
            ...
            <td class="text-end">
                <a href="/stu/enroll?scode={{scode}}"><button class="btn btn-primary btn-sm">수강신청</button></a>
            </td>
        {{/each}}
    </table>
</script>
```

[src]-[main]-[webapp]-[stu] enroll.jsp

```
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">수강신청</h1>
        <div class="card p-3">
            <div class="row">
                <div class="col">학생번호: {{vo.scode}}</div><div class="col">학생이름: {{vo.sname}}</div>
                <div class="col">학생학과: {{vo.dept}}</div><div class="col">지도교수: {{vo.pname}} ({{vo.advisor}})</div>
            </div>
        </div>
        <div id="div_enroll"></div>
    </div>
</div>
<!-- 수강신청목록 템플릿 -->
<script id="temp_enroll" type="text/handlebars-template">
    <table class="table">
        {{#each .}} ... {{/each}}
    </table>
</script>
<script>
    const scode="{{vo.scode}}";
    function getList(){
        $.ajax({
            type:"get",
            url:"/stu/enroll.json",
            data:{scode:scode},
            dataType:"json",
            ...
        });
    }
</script>
```

- 수강신청 등록을 위한 DAO, Controller, JSP 페이지를 작성한다.

수강신청을 위한 Stored Procedure작성

```
CREATE DEFINER='haksa'@'localhost' PROCEDURE `add_enroll`(
  in i_scode char(8),
  in i_lcode char(4),
  out o_count int
)
BEGIN
  select count(*) into o_count from enrollments where scode=i_scode and lcode=i_lcode;
  IF(o_count=0)THEN
    insert into enrollments(scode,lcode) values(i_scode,i_lcode);
    update courses set persons=persons+1 where lcode=i_lcode;
  END IF;
END
```

[src/main/java]-[model] StuDAO.java

```
public class StuDAO {
  ...
  //수강신청등록
  public int insert(String scode, String lcode) {
    int count=-1;
    try {
      String sql="call add_enroll(?, ?, ?)";
      CallableStatement cs=Database.CON.prepareCall(sql);
      cs.setString(1, scode);
      cs.setString(2, lcode);
      cs.registerOutParameter(3, java.sql.Types.INTEGER);
      cs.execute();
      count=cs.getInt(3);
    }catch(Exception e) {
      System.out.println("수강신청등록:" + e.toString());
    }
    return count;
  }
}
```

src/main/java]-[model] CouDAO.java

```
public class CouDAO {
  ...
  //강좌선택을위한 전체목록
  public ArrayList<CourseVO> all(){
    ArrayList<CourseVO> array=new ArrayList<CourseVO>();
    try {
      String sql= "select * from view_cou order by lname";
      PreparedStatement ps=Database.CON.prepareStatement(sql);
      ResultSet rs=ps.executeQuery();
      while(rs.next()) {
        CourseVO vo=new CourseVO();
        ...
        array.add(vo);
      }
    }catch(Exception e) {
      System.out.println("강좌전체목록:" + e.toString());
    }
    return array;
  }
}
```

[src/main/java]-[controller] StudentController.java

```
@WebServlet(value={"/stu/list", "/stu/list.json", "/stu/total", "/enroll/delete","/stu/insert", "/stu/update"... "/enroll/insert"})
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    ...
    switch(request.getServletPath()) {
    ...
    case "/enroll/insert":
        String lcode=request.getParameter("lcode");
        scode=request.getParameter("scode");
        out.print(dao.insert(scode, lcode));
        break;
    }
}
```

[src/main/java]-[controller] CourseController.java

```
@WebServlet(value={"/cou/list", "/cou/total", "/cou/list.json", "/cou/insert", "/cou/update", "/cou/all.json"})
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    ...
    switch(request.getServletPath()) {
    case "/cou/all.json":
        ArrayList<CourseVO> array=dao.all();
        JSONArray jArray=new JSONArray();
        for(CourseVO vo:array) {
            JSONObject obj=new JSONObject();
            ...
            jArray.add(obj);
        }
        out.print(jArray);
        break;
    }
}
```

[src]-[main]-[webapp]-[stu] enroll.jsp ①

```
<div class="card p-3 my-3">
    <div class="row">
        <div class="col" id="div_cou"></div>
        <div class="col"><button class="btn btn-primary" id="btn-enroll">수강신청</button></div>
    </div>
</div>
<div id="div_enroll"></div>
...
<!-- 강좌목록 템플릿 -->
<script id="temp_cou" type="text/handlebars-template">
    <select class="form-select" id="lcode">
        {{#each .}}
            <option value="{{lcode}}" {{dis persons capacity}}>{{lname}} {{persons}}/{{capacity}}</option>
        {{/each}}
    </select>
</script>
<script>
    Handlebars.registerHelper("dis", function(persons, capacity){
        if(persons>=capacity) return "disabled";
    });
</script>
```


[src]-[main]-[webapp]-[stu] enroll.jsp ②

```
<script>
    const scode="${vo.scode}";
    ...
    //강의전체목록 출력
    getCou();
    function getCou() {
        $.ajax({
            type:"get",
            url:"/cou/all.json",
            dataType:"json",
            success:function(data){
                const temp=Handlebars.compile("#temp_cou").html();
                $("#div_cou").html(temp(data));
            }
        });
    }

    //수강신청등록 버튼을 클릭한 경우
    $("#btn-enroll").on("click", function(){
        const lcode=$("#lcode").val();
        if(confirm(lcode + "강좌를 수강신청하십니까?")){
            $.ajax({
                type:"get",
                url:"/enroll/insert",
                data:{scode:scode, lcode:lcode},
                success:function(data){
                    if(data==0) {
                        alert("수강신청이 완료되었습니다.");
                        getList();
                        getCou();
                    }else{
                        alert("이미 수강신청한 과목입니다.");
                    }
                }
            });
        }
    });
}</script>
```

- 수강신청 취소를 위한 DAO, Controller, JSP 페이지를 수정한다.

[src/main/java]-[model] StuDAO.java

```
public class StuDAO {
    ...
    public void delete(String scode, String lcode) { //수강취소
        try {
            String sql="call del_enroll(?, ?)";
            CallableStatement cs=Database.CON.prepareCall(sql);
            cs.setString(1, scode);
            cs.setString(2, lcode);
            cs.execute();
        }catch(Exception e) {
            System.out.println("수강신청취소:" + e.toString());
        }
    }
}
```

수강취소 Stored Procedure

```
CREATE DEFINER='haksa'@'localhost' PROCEDURE `del_enroll`(  
  in i_scode char(8),  
  in i_lcode char(4)  
)  
BEGIN  
  delete from enrollments where scode=i_scode and lcode=i_lcode;  
  update courses set persons=persons-1 where lcode=i_lcode;  
END
```

[src/main/java]-[controller] StudentController.java

```
@WebServlet(value={"/stu/list", "/stu/list.json", "/stu/total", "/enroll/delete", "/stu/insert", "/stu/update", "/stu/enroll.json",  
  "/stu/enroll", "/enroll/insert", "/enroll/delete"})  
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
  request.setCharacterEncoding("UTF-8");  
  switch(request.getServletPath()) {  
    ...  
    case "/enroll/delete":  
      String scode=request.getParameter("scode");  
      String lcode=request.getParameter("lcode");  
      dao.delete(scode, lcode);  
      break;  
    }  
  }  
}
```

[src]-[main]-[webapp]-[stu] enroll.jsp ①

```
<!-- 수강신청목록 템플릿 -->  
<script id="temp_enroll" type="text/handlebars-template">  
  <table class="table">  
    {{#each .}}  
      <tr>  
        ...  
        <td><button class="btn btn-danger btn-sm" lcode="{{lcode}}">취소</button></td>  
      </tr>  
    {{/each}}  
  </table>  
</script>  
<script>  
  ...  
  //수강신청 취소 버튼을 클릭한 경우  
  $(<b>"#div_enroll"</b>).on("click", ".btn-danger", function(){  
    const lcode=$(this).attr("lcode");  
    if(confirm(lcode + "강좌를 수강취소하실래요?")){  
      $.ajax({  
        type:"post",  
        url:"/enroll/delete",  
        data:{scode:scode, lcode:lcode},  
        success:function(){  
          alert("수강이 취소되었습니다.");  
          getList();  
          getCou();  
        }  
      });  
    }  
  });  
</script>
```

- 수강신청 과목 점수 수정

- 강좌별 수강신청한 학생목록 출력을 위한 DAO, Controller, JSP 페이지를 작성한다.

```
[src/main/java]-[model] CouVO.java
```

```
public class GradeVO extends StuVO{
    private String lcode;
    private String edate;
    private int grade;
    //setter(), getter(), toString() 메서드를 generate한다.
}
```

```
[src/main/java]-[model] CouDAO.java
```

```
//특정 강좌를 수강신청한 학생목록
```

```
public ArrayList<GradeVO> list(String lcode){
    ArrayList<GradeVO> array=new ArrayList<GradeVO>();
    try {
        String sql= "select * from view_enroll_stu where lcode=? order by scode";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, lcode);
        ResultSet rs=ps.executeQuery();
        while(rs.next()) {
            GradeVO vo=new GradeVO();
            ...
            array.add(vo);
        }
    } catch(Exception e) {
        System.out.println("성적목록:" + e.toString());
    }
    return array;
}
```

```
[src/main/java]-[controller] CourseController.java
```

```
@WebServlet(value={"/cou/list", "/cou/total", "/cou/list.json", "/cou/insert", "/cou/update"..."/cou/grade.json", "/cou/grade"})
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    switch(request.getServletPath()) {
        ...
        case "/cou/grade.json":
            lcode=request.getParameter("lcode");
            ArrayList<GradeVO> carray=dao.list(lcode);
            JSONArray jArray=new JSONArray();
            for(GradeVO vo:carray) {
                JSONObject obj=new JSONObject();
                ...
                jArray.add(obj);
            }
            out.print(jArray);
            break;
        case "/cou/grade":
            String lcode=request.getParameter("lcode");
            request.setAttribute("vo", dao.read(lcode));
            request.setAttribute("pageName", "/cou/grade.jsp");
            dis.forward(request, response);
            break;
    }
}
```

[src]-[main]-[webapp]-[cou] grade.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="row my-5">
  <div class="col">
    <h1 class="text-center mb-5">점수입력</h1>
    <div class="card p-3">
      <div class="row">
        <div class="col">강좌번호: ${vo.lcode}</div>
        <div class="col-4">강좌이름: ${vo.lname}</div>
        <div class="col">담당교수: ${vo.pname}</div>
        <div class="col">강의시수: ${vo.hours}</div>
        <div class="col">수강인원: ${vo.persons}/${vo.capacity}</div>
      </div>
    </div>
    <hr>
    <div id="div_grade"></div>
  </div>
</div>
<!-- 성적목록 템플릿-->
<script id="temp_grade" type="text/x-handlebar-templates">
  <table class="table">
    {{#each .}}
    <tr>
      <td class="scode">{{scode}}</td>
      <td>{{sname}}</td>
      <td>{{dept}}</td>
      <td>{{edate}}</td>
      <td width="100"><input class="grade form-control text-end" value="{{grade}}" oninput="isNumber(this)"></td>
      <td><button class="btn btn-primary btn-sm">수정</td>
    </tr>
    {{/each}}
  </table>
</script>
<script>
  const lcode="${vo.lcode}";

  getList();
  function getList(){
    $.ajax({
      type:"get",
      url:"/cou/grade.json",
      data:{lcode:lcode},
      dataType:"json",
      success:function(data){
        const temp=Handlebars.compile($("#temp_grade").html());
        $("#div_grade").html(temp(data));
      }
    });
  }

  //숫자인 경우에만 입력
  function isNumber(item){
    item.value=item.value.replace(/[^0-9]/g,'');
  }
</script>
```

- 성적 수정을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] CouDAO.java

```
public class CouDAO {
    ...
    //점수수정
    public void update(GradeVO vo) {
        try {
            String sql="update enrollments set grade=? where scode=? and lcode=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setInt(1, vo.getGrade());
            ps.setString(2, vo.getScode());
            ps.setString(3, vo.getLcode());
            ps.execute();
        } catch (Exception e) {
            System.out.println("점수수정: " + e.toString());
        }
    }
}
```

[src/main/java]-[controller] CourseController.java

```
@WebServlet(value={"/cou/list", "/cou/total", "/cou/list.json", "/cou/insert", "/cou/update", ... "/cou/grade", "/grade/update"})
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    request.setCharacterEncoding("UTF-8");
    switch(request.getServletPath()) {
        ...
        case "/grade/update":
            GradeVO gvo=new GradeVO();
            gvo.setLcode(request.getParameter("lcode"));
            gvo.setScode(request.getParameter("scode"));
            gvo.setGrade(Integer.parseInt(request.getParameter("grade")));
            dao.update(gvo);
            break;
    }
}
```

[src]-[main]-[webapp]-[cou] grade.jsp

```
<script>
    const lcode="${vo.lcode}";
    ...
    //점수 수정버튼을 클릭한 경우
    $("#div_grade").on("click", ".btn-primary", function(){
        const row=$(this).parent().parent();
        const scode=row.find(".scode").text();
        const grade=row.find(".grade").val();
        if(confirm(scode + "학생의 점수를 " + grade + "로 수정하실래요?")){
            $.ajax({
                type:"post",
                url:"/grade/update",
                data:{lcode:lcode, scode:scode, grade:grade},
                success:function(){
                    alert("점수가 수정되었습니다.");
                    getList();
                }
            });
        }
    });
</script>
```

- 쇼핑물 프로그램
- 상품검색 프로그램
- 네이버 검색(쇼핑) API를 이용하여 검색 프로그램을 작성한다.

[src/main/java]-[model] NaverAPI.java

```
public class NaverAPI {
    public static String search(int page, String query) {
        String clientId = "xxxxxxxxxx"; //애플리케이션 클라이언트 아이디
        String clientSecret = "xxxxxxxx"; //애플리케이션 클라이언트 시크릿
        String text = null;
        try {
            text = URLEncoder.encode(query, "UTF-8");
        } catch (UnsupportedEncodingException e) {
            throw new RuntimeException("검색어 인코딩 실패",e);
        }

        int start=(page-1)*5+1;
        String apiURL = "https://openapi.naver.com/v1/search/shop.json?query=" + text;
        apiURL += "&display=5&start=" + start;

        Map<String, String> requestHeaders = new HashMap<>();
        requestHeaders.put("X-Naver-Client-Id", clientId);
        requestHeaders.put("X-Naver-Client-Secret", clientSecret);
        String responseBody = get(apiURL,requestHeaders);

        //System.out.println(responseBody);
        return responseBody;
    }
    private static String get(String apiUrl, Map<String, String> requestHeaders){
        ...
    }
    ...
}
```

[src/main/java]-[controller] GoodsController.java

```
@WebServlet(value={"/goods/search","/goods/search.json"})
public class GoodsController extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");

        switch(request.getServletPath()) {
            case "/goods/search":
                request.setAttribute("pageName", "/goods/search.jsp");
                dis.forward(request, response);
                break;
            case "/goods/search.json": //주소 /goods/search.json?page=1 &query=한빛미디어
                int page=Integer.parseInt(request.getParameter("page"));
                String query=request.getParameter("query");
                String result=NaverAPI.search(page, query);
                out.println(result);
                break;
        }
    }
}
```

```

<div class="row my-5">
  <div class="col">
    <h1 class="text-center mb-5">상품검색</h1>
    <div class="row">
      <form name="frm" class="col-4">
        <div class="input-group">
          <input name="query" class="form-control" value="한빛미디어">
          <button class="btn btn-primary">검색</button>
        </div>
      </form>
    </div>
    <div id="div_search"></div>
    <div id="pagination" class="pagination justify-content-center"></div>
  </div>
</div>
<!-- 검색목록 템플릿 -->
<script id="temp_search">
  <table class="table">
    {{#each items}}
    <tr>
      <td></td>
      <td class="title">{{title}}</td>
      <td>{{lprice}}</td>
      <td>{{maker}}</td>
    </tr>
    {{/each}}
  </table>
</script>
<script>
  let page=1;
  let query="한빛미디어";
  getList();
  $(frm).on("submit", function(e){
    e.preventDefault();
    query=$(frm.query).val();
    if(query=="") {
      alert("검색어를 입력하세요!");
      $(frm.query).focus();
    }else{
      $("#pagination").twbsPagination("changeTotalPages", 10, 1);
    }
  });

  function getList(){
    $.ajax({
      type:"get",
      url:"/goods/search.json",
      data:{page:page, query:query},
      dataType:"json",
      success:function(data){
        const temp=Handlebars.compile($("#temp_search").html());
        $("#div_search").html(temp(data));
      }
    });
  }

  $('#pagination').twbsPagination({ ... });
</script>

```

- 검색상품 저장하기 위한 DAO, Controller, JSP 페이지를 작성한다.

상품(goods) 테이블 생성

```
create table goods(
    gid char(8) not null primary key,
    title varchar(300) not null,
    price int default 0,
    maker varchar(200),
    image varchar(300),
    regDate datetime default now()
);
```

[src/main/java]-[controller] GoodsController.java

```
@WebServlet(value={"/goods/search","/goods/search.json","/goods/append"})
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String path="/upload/goods/";
    File mkPath=new File("c:" + path); //path가 존재하지 않으면 path를 생성한다.
    if(!mkPath.exists()) mkPath.mkdir();
    switch(request.getServletPath()) {
        case "/goods/append":
            try {
                UUID uuid=UUID.randomUUID();
                String gid=uuid.toString().substring(0,8);
                URL url=new URL(request.getParameter("image"));
                InputStream is=url.openStream();
                String fileName = gid + ".jpg";
                FileOutputStream fos=new FileOutputStream("c:" + path + fileName); //이미지저장
                int data=0;
                while((data=is.read()) != -1) { fos.write(data); }
                GoodsVO vo=new GoodsVO();
                vo.setGid(gid);
                vo.setTitle(request.getParameter("title"));
                vo.setMaker(request.getParameter("maker"));
                vo.setPrice(Integer.parseInt(request.getParameter("lprice")));
                vo.setImage(path + fileName);
                gdao.insert(vo); //데이터저장
            } catch (Exception e) {
                System.out.println("상품이미지저장:" + e.toString());
            }
            break;
    }
}
```

[src/main/java]-[model] GoodsDAO.java

```
public class GoodsDAO {
    public void insert(GoodsVO vo) { //상품등록
        try {
            String sql="insert into goods(gid,title,price,maker,image) values(?,?,?,?,?)";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ...
            ps.execute();
        } catch (Exception e) {
            System.out.println("상품등록:" + e.toString());
        }
    }
}
```


[src/main/java]-[model] GoodsVO.java

```
public class GoodsVO {
    private String gid;
    private String title;
    private int price;
    private String image;
    private String regDate;
    private String maker;
    //setter(), getter(), toString() 메서드를 generate한다.
}
```

[src]-[main]-[webapp]-[goods] search.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">상품검색</h1>
        ...
    </div>
</div>
<!-- 검색목록 템플릿 -->
<script id="temp_search">
    <table class="table">
        {{#each items}}
            <tr goods="{{toString @this}}">
                <td></td>
                <td class="title">{{title}}</td>
                <td>{{lprice}}</td>
                <td>{{maker}}</td>
                <td><button class="btn btn-primary btn-sm btn-save">저장</button></td>
            </tr>
        {{/each}}
    </table>
</script>
<script>
    Handlebars.registerHelper("toString", function(goods){
        return JSON.stringify(goods); //Object->String
    });
</script>
<script>
    ...
    //각행의 저장버튼을 클릭한 경우
    $("#div_search").on("click", ".btn-save", function(){
        const row=$(this).parent().parent();
        const goods=JSON.parse(row.attr("goods")); //String->Object
        const title=row.find(".title").text();
        if(confirm(title + " 상품을 저장하실래요?")){
            goods["title"]=title;
            $.ajax({
                type:"post",
                url:"/goods/append",
                data:goods,
                success:function(){
                    alert("상품이 저장되었습니다.")
                }
            });
        }
    });
</script>
```

- 상품관리 프로그램

- 등록된 상품 목록 출력을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] GoodsDAO.java

```
public class GoodsDAO {
    public ArrayList<GoodsVO> list(String query, int page){ //상품목록
        ArrayList<GoodsVO> array=new ArrayList<GoodsVO>();
        try {
            String sql="select * from goods where title like ? order by regDate desc limit ?,6";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, "%" + query + "%");
            ps.setInt(2, (page-1)*6);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                ...
            }
        }catch(Exception e) {
            System.out.println("상품목록:" + e.toString());
        }
        return array;
    }
    public int total(String query) { //상품검색수
        int total=0;
        try {
            String sql="select count(*) cnt from goods where title like ?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, "%" + query + "%");
            ResultSet rs=ps.executeQuery();
            if(rs.next()) total=rs.getInt("cnt");
        }catch(Exception e) {
            System.out.println("검색수:" + e.toString());
        }
        return total;
    }
}
```

[src/main/java]-[controller] GoodsController.java

```
@WebServlet(value={"/goods/search","/goods/search.json","/goods/append", "/goods/list.json", "/goods/total", "/goods/list" })
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    ...
    switch(request.getServletPath()) {
        case "/goods/list.json": //실행: /goods/list.json?page=1 & query=
            page=Integer.parseInt(request.getParameter("page"));
            query=request.getParameter("query");
            Gson gson=new Gson();
            out.print(gson.toJson(gdao.list(query, page)));
            break;
        case "/goods/total":
            query=request.getParameter("query");
            out.print(gdao.total(query));
            break;
        case "/goods/list":
            request.setAttribute("pageName", "/goods/list.jsp");
            dis.forward(request, response);
            break;
    }
}
```

[src]-[main]-[webapp]-[goods] list.jsp

```
<div class="row my-5">
  <div class="col">
    <h1 class="text-center mb-5">상품목록</h1>
    <div class="row justify-content-end">
      <form name="frm" class="col-4">
        <div class="input-group">
          <input name="query" class="form-control" value="">
          <button class="btn btn-primary">검색</button>
        </div>
      </form>
    </div>
    <div id="div_goods"></div>
    <div id="pagination" class="pagination justify-content-center mt-3"></div>
  </div>
</div>
<!-- 상품목록 템플릿 -->
<script id="temp_goods" type="x-handlebars/template">
  <table class="table">
    {{#each .}}
      ...
    {{/each}}
  </table>
</script>
<script>
  let page=1;
  let query="";
  $(frm).on("submit", function(e){
    e.preventDefault();
    query=$(frm.query).val();
    page=1;
    getTotal();
  });

  getTotal();
  function getTotal(){
    $.ajax({
      type:"get",
      url:"/goods/total",
      data:{query:query},
      success:function(data){
        if(data==0){
          $("#div_goods").html("<h3 class='text-center my-5'>상품이 존재하지 않습니다.</h3>");
        }else{
          const totalPages=Math.ceil(data/6);
          $("#pagination").twbsPagination("changeTotalPages", totalPages, page);
        }
        if(data > 6){
          $("#pagination").show();
        }else{
          $("#pagination").hide();
        }
      }
    });
  }

  function getList(){ ... }
  $('#pagination').twbsPagination( ... );
</script>
```

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<style>
    .title, .price {
        font-size:12px;
    }
</style>
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">쇼핑몰</h1>
        <div class="row justify-content-end">
            <form name="frm" class="col-4">
                <div class="input-group">
                    <input name="query" class="form-control" value="">
                    <button class="btn btn-primary">검색</button>
                </div>
            </form>
        </div>
        <div id="div_goods" class="row"></div>
        <div id="pagination" class="pagination justify-content-center mt-3"></div>
    </div>
</div>
<!-- 상품목록 템플릿 -->
<script id="temp_goods" type="x-handlebars/template">
    {{#each .}}
        <div class="col-6 col-md-4 col-lg-2 mb-3">
            <div class="card p-3">
                
                <div class="ellipsis title mt-2">{{title}}</div>
                <div class="price">{{price price}}</div>
            </div>
        </div>
    {{/each}}
</script>
<script>
    Handlebars.registerHelper("price", function(price){
        return price.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",") + "원";
    });
</script>
<script>
    let page=1;
    let query="";

    $(frm).on("submit", function(e){
        e.preventDefault();
        query=$(frm.query).val();
        page=1;
        getTotal();
    });

    getTotal();
    function getTotal(){
        $.ajax({ ... });
    }
    function getList(){
        $.ajax({ ... });
    }
    $('#pagination').twbsPagination({ ... });
</script>

```

- 상품삭제를 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] GoodsDAO.java

```
public void delete(String gid) { //상품삭제
    try {
        String sql="delete from goods where gid=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, gid);
        ps.execute();
    } catch (Exception e) {
        System.out.println("상품삭제:" + e.toString());
    }
}
```

[src/main/java]-[controller] GoodsController.java

```
@WebServlet(value={..., "/goods/total", "/goods/list", "/goods/delete"})
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    switch(request.getServletPath()) {
        ...
        case "/goods/delete":
            try {
                String gid=request.getParameter("gid");
                String image=request.getParameter("image");
                File file=new File("c:/" + image);
                file.delete(); //이미지파일삭제
                gdao.delete(gid); //데이터삭제
            } catch (Exception e) {
                System.out.println("상품삭제:" + e.toString());
            }
            break;
        }
    }
}
```

[src]-[main]-[webapp]-[goods] list.jsp

```
<!-- 상품목록 템플릿 -->
<script id="temp_goods" type="x-handlebars/template">
    {{#each .}}
        ...
        <td><button class="btn btn-danger btn-sm" gid="{{gid}}" image="{{image}}">삭제</button><td>
    {{/each}}
</script>
<script>
    $("#div_goods").on("click", ".btn-danger", function(){ //삭제버튼을 클릭한경우
        const image=$(this).attr("image");
        const gid=$(this).attr("gid");
        if(confirm("해당 상품을 삭제하실래요?")){
            $.ajax({
                type:"post",
                url:"/goods/delete",
                data:{gid:gid, image:image},
                success:function(){
                    alert("상품이 삭제되었습니다.");
                    getTotal();
                }
            });
        }
    });
</script>
```

- 상품정보 수정을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] GoodsDAO.java

```
public class GoodsDAO {
    public void update(GoodsVO vo) { //상품정보 수정
        try {
            String sql="update goods set title=?,price=?,maker=?,image=? where gid=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, vo.getTitle());
            ...
            ps.execute();
        }catch(Exception e) {
            System.out.println("상품수정:" + e.toString());
        }
    }
    public GoodsVO read(String gid) { //상품정보 읽기
        GoodsVO vo=new GoodsVO();
        try {
            String sql="select * from goods where gid=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, gid);
            ResultSet rs=ps.executeQuery();
            if(rs.next()) { ... }
        }catch(Exception e) {
            System.out.println("상품정보:" + e.toString());
        }
        return vo;
    }
}
```

[src/main/java]-[controller] GoodsController.java

```
@WebServlet(value={ "/goods/total", "/goods/list", "/goods/delete", "/goods/insert", "/goods/update"})
public class GoodsController extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        ...
        switch(request.getServletPath()) {
            case "/goods/update":
                request.setAttribute("vo", gdao.read(request.getParameter("gid")));
                request.setAttribute("pageName", "/goods/update.jsp");
                dis.forward(request, response);
                break;
        }
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            ...
            case "/goods/update":
                multi=new MultipartRequest(request, "c:"+path, 1024*1024*10,"UTF-8", new DefaultFileRenamePolicy());
                image=multi.getFilesystemName("image")!=null?
                    multi.getParameter("oldImage") : path + multi.getFilesystemName("image");
                vo=new GoodsVO();
                ...
                vo.setImage(image);
                gdao.update(vo);
                response.sendRedirect("/goods/list");
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[goods] list.jsp

```
<!-- 상품목록 템플릿 -->
<td class="gid"><a href="/goods/update?gid={{gid}}">{{gid}}</a></td>
...
</script>
```

[src]-[main]-[webapp]-[goods] update.jsp ①

```
<div class="row my-5 justify-content-center">
  <div class="col-md-10 col-lg-8">
    <h1 class="text-center mb-5">상품정보수정</h1>
    <form name="frm" class="card p-3" method="post" enctype="multipart/form-data">
      <div class="input-group mb-3">
        <span class="input-group-text">상품코드</span>
        <input name="gid" class="form-control" value="{{vo.gid}}" readonly>
      </div>
      <div class="input-group mb-3">
        <span class="input-group-text">상품명</span>
        <input name="title" class="form-control" value="{{vo.title}}">
      </div>
      <div class="input-group mb-3">
        <span class="input-group-text">상품가격</span>
        <input name="price" class="form-control" oninput="isNumber(this)" value="{{vo.price}}">
      </div>
      <div class="input-group mb-3">
        <span class="input-group-text">제&nbsp;&nbsp;&nbsp;조&nbsp;&nbsp;&nbsp;사&nbsp;&nbsp;&nbsp;</span>
        <input name="maker" class="form-control" value="{{vo.maker}}">
      </div>
      <div class="my-5">
        <ul class="nav nav-tabs" id="myTab" role="tablist">
          <li class="nav-item" role="presentation">
            <button class="nav-link active" id="image-tab" data-bs-target="#image-tab-pane">대표이미지</button>
          </li>
          <li class="nav-item" role="presentation">
            <button class="nav-link" id="detail-tab" data-bs-target="#detail-tab-pane">상세설명</button>
          </li>
          <li class="nav-item" role="presentation">
            <button class="nav-link" id="attach-tab" data-bs-target="#attach-tab-pane">첨부이미지</button>
          </li>
        </ul>
        <div class="tab-content pt-5" id="myTabContent">
          <div class="tab-pane fade show active" id="image-tab-pane" role="tabpanel">
            <div class="input-group mb-3">
              <input name="image" type="file" class="form-control" accept="image/*">
              <input name="oldImage" value="{{vo.image}}" type="hidden">
            </div>
            <div></div>
          </div>
          <div class="tab-pane fade" id="detail-tab-pane" role="tabpanel" aria-labelledby="detail-tab"></div>
          <div class="tab-pane fade" id="attach-tab-pane" role="tabpanel" aria-labelledby="attach-tab"></div>
        </div>
      </div>
      <div class="text-center my-3">
        <input type="submit" value="상품수정" class="btn btn-primary">
        <input type="reset" value="수정취소" class="btn btn-secondary">
      </div>
    </form>
  </div>
</div>
```

[src]-[main]-[webapp]-[goods] update.jsp ②

```
<script>
    $(frm).on("submit", function(e){
        e.preventDefault();
        const title=$(frm.title).val();
        const price=$(frm.price).val();
        if(title==" || price==" ) {
            alert("상품명, 상품가격을 입력하세요!");
            $(frm.title).focus();
        }else{
            if(confirm("상품정보를 수정하실래요?")){
                frm.submit(); //수정하기
            }
        }
    });

    const oldImage="${vo.image}";
    $(frm).on("reset", function(){ //리셋인 경우
        $("#image").attr("src", oldImage);
    })
</script>
```

- 상품정보 읽기 페이지를 위한 Controller와 JSP 페이지를 작성한다.

[src/main/java]-[controller] GoodsController.java

```
@WebServlet(value={ ..."/goods/delete", "/goods/insert", "/goods/update", "/goods/read"})
public class GoodsController extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        ...
        switch(request.getServletPath()) {
            ...
            case "/goods/read":
                request.setAttribute("vo", gdao.read(request.getParameter("gid")));
                request.setAttribute("pageName", "/goods/read.jsp");
                dis.forward(request, response);
                break;
        }
    }
}
```

[src]-[main]-[webapp] about.jsp

```
<!-- 상품목록 템플릿 -->
<script id="temp_goods" type="x-handlebars/template">
    {{#each .}}
    <div class="col-6 col-md-4 col-lg-2 mb-3">
        <div class="card">
            <div class="card-body">
                <a href="/goods/read?gid={{gid}}">
                    
                </a>
                <div class="ellipsis title mt-2">{{title}}</div>
                <div class="price">{{fmtPrice price}}</div>
            </div>
        </div>
    </div>
    {{/each}}
</script>
```


[src]-[main]-[webapp]-[goods] read.jsp

```
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<div class="row my-5 justify-content-center">
  <div class="col-md-10 col-lg-8">
    <h1 class="text-center mb-5">상품정보</h1>
    <div class="card p-3">
      <div class="row">
        <div class="col-lg-4 text-center"></div>
        <div class="col">
          <hr>
          <div>상품코드: ${vo.gid}</div>
          <div>상품명: ${vo.title}</div>
          <hr>
          <div class="my-2">상품가격: <fmt:formatNumber value="${vo.price}" pattern="#,###원"/></div>
          <div class="my-2">제조사: ${vo.maker}</div>
          <div class="my-2">상품등록일: ${vo.regDate}</div>
          <hr>
          <div class="row">
            <div class="col-6"><button class="btn btn-primary" id="btn-cart">장바구니</button></div>
          </div>
        </div>
      </div>
    </div>
  <div class="my-5">
    <ul class="nav nav-tabs" id="myTab" role="tablist">
      <li class="nav-item" role="presentation">
        <button class="nav-link active" id="detail-tab" data-bs-target="#detail-tab-pane">
          상세설명
        </button>
      </li>
      <li class="nav-item" role="presentation">
        <button class="nav-link" id="review-tab" data-bs-target="#review-tab-pane">
          고객리뷰
        </button>
      </li>
      <li class="nav-item" role="presentation">
        <button class="nav-link" id="attach-tab" data-bs-target="#attach-tab-pane">
          첨부이미지
        </button>
      </li>
    </ul>
    <div class="tab-content pt-5" id="myTabContent">
      <!-- 상세설명 -->
      <div class="tab-pane fade show active" id="detail-tab-pane" role="tabpanel" tabindex="0">
        ...
      </div>
      <!-- 상세설명 -->
      <div class="tab-pane fade" id="review-tab-pane" role="tabpanel" aria-labelledby="review-tab" tabindex="1">
        ...
      </div>
      <!-- 첨부이미지 -->
      <div class="tab-pane fade" id="attach-tab-pane" role="tabpanel" aria-labelledby="attach-tab" tabindex="2">
        ...
      </div>
    </div>
  </div>
</div>
</div>
</div>
```

- 장바구니관리

- 장바구니에 상품을 저장하기 위한 CartVO, Controller를 작성한다.

```
public class CartVO extends GoodsVO{
    private int qnt;
    //setter(), getter(), toString() 메서드를 generate한다.
}
```

[src/main/java]-[controller] CartController.java

```
@WebServlet(value={ "/cart/insert" })
public class CartController extends HttpServlet {
    GoodsDAO gdao=new GoodsDAO();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        HttpSession session=request.getSession();
        ArrayList<CartVO> arrCart=session.getAttribute("arrCart") == null ? new ArrayList<CartVO>() :
                                                                    (ArrayList<CartVO>)session.getAttribute("arrCart");

        switch(request.getServletPath()) {
        case "/cart/insert": //장바구니 상품 넣기
            String gid=request.getParameter("gid");
            boolean find=false;
            for(CartVO vo:arrCart) {
                if(gid.equals(vo.getGid())) {
                    vo.setQnt(vo.getQnt() + 1); //장바구니에 존재하면 수량1증가
                    find=true;
                }
            }
            GoodsVO gvo=gdao.read(gid);
            CartVO cvo=new CartVO();
            if(!find) {
                ...
                cvo.setQnt(1);
                arrCart.add(cvo); //장바구니에 없으면 장바구니에 추가
            }
            session.setAttribute("arrCart", arrCart);
            break;
        }
    }
}
```

[src]-[main]-[webapp]-[goods] read.jsp

```
...
$("#btn-cart").on("click", function(){ //장바구니 버튼을 클릭한 경우
    $.ajax({
        type:"get",
        url:"/cart/insert",
        data: {gid:gid},
        success:function(){
            if(confirm("계속 쇼핑하실래요?")) {
                location.href="/";
            } else {
                location.href="/cart/list";
            }
        }
    });
});
```

- 장바구니 목록 출력, 수량수정, 삭제에 위한 Controller와 JSP 페이지를 작성한다.

```
[src/main/java]-[controller] CartController.java
```

```
@WebServlet(value={"/cart/insert", "/cart/list", "/cart/list.json", "/cart/delete", "/cart/update"})
...
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out=response.getWriter();
    RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
    ...
    switch(request.getServletPath()) {
    case "/cart/insert": //장바구니 상품 넣기
        String gid=request.getParameter("gid");
        boolean find=false;
        for(CartVO vo:arrCart) {
            if(gid.equals(vo.getGid())) {
                vo.setQnt(vo.getQnt() + 1); //장바구니에 존재하면 수량1증가
                find=true;
            }
        }
        GoodsVO gvo=gdao.read(gid);
        CartVO cvo=new CartVO();
        if(!find) {
            ...
            cvo.setQnt(1);
            arrCart.add(cvo); //장바구니에 없으면 장바구니에 추가
        }
        session.setAttribute("arrCart", arrCart);
        break;
    }
    case "/cart/list":
        request.setAttribute("pageName", "/order/cart.jsp");
        dis.forward(request, response);
        break;
    case "/cart/list.json":
        Gson gson=new Gson();
        out.print(gson.toJson(arrCart));
        break;
    case "/cart/delete":
        gid=request.getParameter("gid");
        for(CartVO vo:arrCart) {
            if(gid.equals(vo.getGid())) {
                arrCart.remove(vo); //카드에서 상품을 찾으면 삭제
                break;
            }
        }
        session.setAttribute("arrCart", arrCart);
        break;
    case "/cart/update":
        gid=request.getParameter("gid");
        int qnt=Integer.parseInt(request.getParameter("qnt"));
        for(CartVO vo:arrCart) {
            if(gid.equals(vo.getGid())) vo.setQnt(qnt);
        }
        session.setAttribute("arrCart", arrCart);
        break;
    }
}
```

[src]-[main]-[webapp]-[order] cart.jsp ①

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<div class="row my-5" id="page_cart">
    <div class="col">
        <h1 class="text-center mb-5">장바구니</h1>
        <div id="div_cart"></div>
    </div>
</div>
<!-- 카트목록 템플릿 -->
<script id="temp_cart" type="text/x-handlebars-template">
    <table class="table">
        <tr>
            <th><input type="checkbox" id="all"></th>
            <th>상품코드</th><th>이미지</th><th>상품명</th><th>상품가격</th><th>상품수량</th><th>상품금액</th><th>삭제</th>
        </tr>
        {{#each .}}
        <tr class="tr" price="{{price}}">
            <td><input type="checkbox" class="chk" goods="{{toString @this}}"></td>
            <td class="gid">{{gid}}</td>
            <td></td>
            <td class="text-truncate" style="max-width:250px;">{{title}}</td>
            <td>{{sum price 1}}</td>
            <td width="200">
                <input class="qnt" value="{{qnt}}" size=5 oninput="isNumber(this)">&nbsp;
                <button class="btn btn-primary btn-sm btn-update">수정</button>
            </td>
            <td>{{sum price qnt}}</td>
            <td><button class="btn btn-danger btn-sm" gid="{{gid}}">삭제</button></td>
        </tr>
        {{/each}}
        <tr>
            <td colspan="8" class="text-end pe-5"><h5>총합계: <span id="sum">0원</span></h5></td>
        </tr>
    </table>
</script>
<script>
    Handlebars.registerHelper("sum", function(price, qnt){
        const sum=price*qnt;
        return sum.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",");
    });

    Handlebars.registerHelper("toString", function(goods){
        return JSON.stringify(goods); //object->string
    });
</script>
<script>
    getList();

    function getSum(){
        let sum=0;
        $("#div_cart .tr").each(function(){
            const price=$(this).attr("price");
            const qnt=$(this).find(".qnt").val();
            sum += price*qnt;
        });
        $("#sum").html(sum.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ","));
    }
</script>
```

```

<script>
...
function getList(){
    $.ajax({
        type:"get",
        url:"/cart/list.json",
        dataType:"json",
        success:function(data){
            if(data.length==0){
                $("#div_cart").html("<h3 class='text-center'>장바구니가 비어있습니다.</h3>");
            }else{
                const temp=Handlebars.compile($("#temp_cart").html());
                $("#div_cart").html(temp(data));
                getSum();
            }
        }
    });
}

$("#div_cart").on("click", "#all", function(){ //전체선택 체크박스를 클릭한 경우
    if($(this).is(":checked")){
        $("#div_cart .chk").each(function(){ $(this).prop("checked", true); });
    }else{
        $("#div_cart .chk").each(function(){ $(this).prop("checked", false); });
    }
});

$("#div_cart").on("click", ".chk", function(){ //각 행의 체크박스를 클릭한 경우
    const all=$("#div_cart .chk").length;
    const chk=$("#div_cart .chk:checked").length;
    if(all==chk) $("#div_cart #all").prop("checked", true);
    else $("#div_cart #all").prop("checked", false);
});

$("#div_cart").on("click", ".btn-update", function(){ //각 행의 수량 수정버튼을 클릭한 경우
    const gid=$(this).parent().parent().find(".gid").text();
    const qnt=$(this).parent().parent().find(".qnt").val();
    if(confirm(gid + "상품의 수량을 " + qnt + "로 변화하신텐가요?")){
        $.ajax({
            type:"get",
            url:"/cart/update",
            data:{gid:gid, qnt:qnt},
            success:function(){ getList(); }
        });
    }
});

$("#div_cart").on("click", ".btn-danger", function(){ //각 행의 삭제버튼을 클릭한 경우
    const gid=$(this).attr("gid");
    if(confirm(gid + "변 상품을 삭제하신텐가요?")){
        $.ajax({
            type:"get",
            url:"/cart/delete",
            data:{gid:gid},
            success:function(){ getList(); }
        });
    }
});
</script>

```

- 로그인, 로그아웃 관리

- 로그인을 위한 테이블과 VO, DAO, Controller, JSP 페이지를 작성한다.

회원 테이블 생성

```
create table users(  
    uid varchar(2) not null primary key,  
    upass varchar(100) not null,  
    uname varchar(20) not null,  
    phone varchar(20),  
    photo varchar(200),  
    address1 varchar(200),  
    address2 varchar(200),  
    role int default 2,  
    regDate datetime default now()  
);
```

[src/main/java]-[model] UserVO.java

```
public class UserVO {  
    private String uid;  
    private String upass;  
    private String uname;  
    private String phone;  
    private String address1;  
    private String address2;  
    private String regDate;  
    private String photo;  
    private int role;  
    //setter(), getter(), toString() 메서드를 generate한다.  
}
```

[src/main/java]-[model] UserDAO.java

```
public class UserDAO {  
    SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  
    public UserVO read(String uid) { //회원정보 읽기  
        UserVO vo=new UserVO();  
        try {  
            String sql="select * from users u where uid=?";  
            PreparedStatement ps=Database.CON.prepareStatement(sql);  
            ps.setString(1, uid);  
            ResultSet rs=ps.executeQuery();  
            if(rs.next()) {  
                vo.setUid(rs.getString("uid"));  
                vo.setUpass(rs.getString("upass"));  
                vo.setUname(rs.getString("uname"));  
                vo.setPhone(rs.getString("phone"));  
                vo.setAddress1(rs.getString("address1"));  
                vo.setAddress2(rs.getString("address2"));  
                vo.setRegDate(sdf.format(rs.getTimestamp("regDate")));  
                vo.setPhoto(rs.getString("photo"));  
            }  
        } catch (Exception e) {  
            System.out.println("회원정보읽기:" + e.toString());  
        }  
        return vo;  
    }  
}
```

[src/main/java]-[controller] UserController.java

```
@WebServlet(value= {"/user/login"})
public class UserController extends HttpServlet {
    UserDao dao=new UserDao();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=utf-8");
        PrintWriter out=response.getWriter();
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");

        switch(request.getServletPath()){
            case "/user/login":
                request.setAttribute("pageName", "/user/login.jsp");
                dis.forward(request, response);
                break;
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        PrintWriter out=response.getWriter();
        HttpSession session=request.getSession();

        switch(request.getServletPath()) {
            case "/user/login":
                String uid=request.getParameter("uid");
                String upass=request.getParameter("upass");
                UserVO user=dao.read(uid);
                int result=0; //아이디가 없는 경우
                if(user.getId()!=null) {
                    if(user.getUpass().equals(upass)) {
                        result=1; //성공
                        session.setAttribute("user", user);
                    }else {
                        result=2; //비밀번호 불일치 경우
                    }
                }
                out.println(result);
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[user] login.jsp ①

```
<div class="row my-5 justify-content-center">
    <div class="col-md-8 col-lg-6 my-5">
        <h1 class="text-center mb-5">로그인</h1>
        <form name="frm" class="card p-3">
            <input name="uid" class="form-control mb-3" placeholder="아이디" value="blue">
            <input name="upass" type="password" class="form-control mb-3" placeholder="비밀번호" value="pass">
            <button class="btn btn-primary w-100">로그인</button>
            <div class="row mt-3">
                <div class="text-end">
                    <a href="/user/insert">회원가입</a>
                </div>
            </div>
        </form>
    </div>
</div>
```

[src]-[main]-[webapp]-[user] login.jsp ②

```
<script>
    $(frm).on("submit", function(e){
        e.preventDefault();
        const uid=$(frm.uid).val();
        const upass=$(frm.upass).val();
        if(uid=="" || upass==""){
            alert("아이디와 비밀번호를 입력하세요!");
        }else{
            $.ajax({
                type:"post",
                url:"/user/login",
                data:{uid:uid, upass:upass},
                success:function(data){
                    if(data==0) {
                        alert("아이디가 존재하지 않습니다!");
                        $(frm.uid).focus();
                    }else if(data==2){
                        alert("비밀번호가 일치하지 않습니다!");
                        $(frm.upass).focus();
                    }else{
                        alert("로그인에 성공하였습니다!");
                        location.href="/";
                    }
                }
            })
        }
    });
}</script>
```

- 로그인/로그아웃 상태일 경우 header.jsp 페이지를 수정한다.

[src]-[main]-[webapp] header.jsp

```
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <c:if test="${user!=null}">
            <li class="nav-item"><a class="nav-link active" aria-current="page" href="/goods/search">상품검색</a></li>
            <li class="nav-item"><a class="nav-link active" aria-current="page" href="/goods/list">상품목록</a></li>
        </c:if>
        <c:if test="user==null">
            <li class="nav-item"><a class="nav-link active" aria-current="page" href="/cart/list">장바구니</a></li>
        </c:if>
    </ul>
    <ul class="navbar-nav mb-2 mb-lg-0">
        <c:if test="${user==null}">
            <li class="nav-item"><a class="nav-link active" aria-current="page" href="/user/login">로그인</a></li>
        </c:if>
        <c:if test="${user!=null}">
            <li class="nav-item">
                <a class="nav-link active" aria-current="page" href="/user/read?uid=${user.uid}">${user.username}님</a>
            </li>
            <li class="nav-item">
                <a class="nav-link active" aria-current="page" href="/user/logout">로그아웃</a>
            </li>
        </c:if>
    </ul>
</div>
```


- 로그아웃과 마이페이지를 위한 Controller와 JSP페이지를 작성한다.

```
[src/main/java]-[controller] UserController.java
```

```
@WebServlet(value= {"/user/login", "/user/logout", "/user/read" })
public class UserController extends HttpServlet {
    UserDao dao=new UserDao();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()){
            ...
            case "/user/logout":
                session.removeAttribute("user");
                response.sendRedirect("/");
                break;
            case "/user/read":
                request.setAttribute("vo", dao.read(request.getParameter("uid")));
                request.setAttribute("pageName", "/user/read.jsp");
                dis.forward(request, response);
                break;
        }
    }
}
```

```
[src]-[main]-[webapp]-[user] read.jsp
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<style>
    span {
        width: 150px;
        justify-content: center;
    }
</style>
<div class="row my-5 justify-content-center">
    <div class="col-md-10 col-lg-8">
        <h1 class="text-center mb-5">회원정보</h1>
        <div class="card p-3">
            <div>
                <c:if test="${vo.photo==null || vo.photo==''}">
                    
                </c:if>
                <c:if test="${vo.photo!=null && vo.photo!=''}">
                    
                </c:if>
                <span class="w-100 ms-3"> ${vo.uname } ${vo.uid }</span>
            </div>
            <hr>
            <div class="input-group mb-3">
                <span class="input-group-text">전화번호</span>
                <input name="phone" class="form-control" value="${vo.phone}" disabled>
            </div>
            ...
            <c:if test="${user.uid == vo.uid }">
                <div class="text-center my-3">
                    <a href="/user/update?uid=${vo.uid}" class="btn btn-primary px-5">정보수정</a>
                </div>
            </c:if>
        </div>
    </div>
</div>
</div>
```

- 로그인 상태 유지를 위한 Controller와 JSP 페이지를 수정한다.

```
[src]-[main]-[webapp]-[user] login.jsp
```

```
<div class="row my-5 justify-content-center">
  ...
  <button class="btn btn-primary w-100">로그인</button>
  <div class="row mt-3">
    <div class="col">
      <input name="is_check" class="form-check-input" type="checkbox">
      <span>로그인 상태 저장</span>
    </div>
    <div class="col text-end">
      <a href="/user/insert">회원가입</a>
    </div>
  </div>
</div>
</div>
<script>
  const target="${target}";
  $(frm).on("submit", function(e){
    ...
    const is_check=$(frm.is_check).is(":checked");
    if(target=="") { //로그인 성공한 경우
      location.href="/";
    }else{
      location.href=target;
    }
  });
</script>
```

```
[src/main/java]-[controller] UserController.java
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
  switch(request.getServletPath()) {
    case "/user/login":
      boolean is_check=Boolean.parseBoolean(request.getParameter("is_check"));
      if(is_check) { //로그인 성공한 경우
        Cookie cookie = new Cookie("uid", user.getUid());
        cookie.setPath("/");
        cookie.setMaxAge(60*60*24*7);
        response.addCookie(cookie);
      }
      break;
  }
}
```

```
[src/main/java]-[controller] HomeController.java
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
  Cookie[] cookies = request.getCookies();
  if(cookies !=null) {
    for(Cookie cookie:request.getCookies()) {
      if(cookie.getName().equals("uid")) {
        String uid=cookie.getValue();
        UserVO user=dao.read(uid);
        HttpSession session=request.getSession();
        session.setAttribute("user", user);
      }
    }
  }
  ...
}
```

- 회원관리

- 회원가입을 위한 DAO, Controller, JSP 페이지를 작성한다.

```
[src/main/java]-[model] UserDao.java
```

```
public void insert(UserVO vo) { //회원등록
    try {
        String sql="insert into users(uid,upass,uname,phone,address1,address2,photo) values(?,?,?,?,?,?,?)";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ...
        ps.execute();
    } catch (Exception e) {
        System.out.println("회원등록:" + e.toString());
    }
}
```

```
[src/main/java]-[controller] UserController.java
```

```
@WebServlet(value= {"/user/login", "/user/logout", "/user/read", "/user/insert"})
public class UserController extends HttpServlet {
    UserDao dao=new UserDao();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        ...
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
        switch(request.getServletPath()){
            ...
            case "/user/insert":
                request.setAttribute("pageName","/user/insert.jsp");
                dis.forward(request, response);
                break;
        }
    }
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        PrintWriter out=response.getWriter();
        HttpSession session=request.getSession();
        ...
        //폴더생성
        String path="/upload/photo/";
        File mdPath=new File("c:" + path);
        if(!mdPath.exists()) mdPath.mkdir();

        switch(request.getServletPath()) {
            case "/user/insert":
                //사진저장
                MultipartRequest multi=new MultipartRequest(
                    request, "c:"+path, 1024*1024*10,"UTF-8", new DefaultFileRenamePolicy());
                String photo = multi.getFilesystemName("photo")==null ? "":
                path + multi.getFilesystemName("photo");
                //데이터저장
                UserVO vo=new UserVO();
                ...
                vo.setPhoto(photo);
                dao.insert(vo);
                response.sendRedirect("/user/login");
                break;
        }
    }
}
```

[src]-[main]-[webapp] home.jsp

```
<head>
...
<script src="http://code.jquery.com/jquery-1.9.1.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/handlebars.js/3.0.1/handlebars.js"></script>
<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=xxx"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/twbs-pagination/1.4.2/jquery.twbsPagination.min.js"></script>
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.3/font/bootstrap-icons.css">
<link rel="stylesheet" href="/css/style.css"/>
<script src="https://t1.daumcdn.net/mapjsapi/bundle/postcode/prod/postcode.v2.js"></script>
</head>
```

[src]-[main]-[webapp]-[user] insert.jsp ①

```
<script src="https://t1.daumcdn.net/mapjsapi/bundle/postcode/prod/postcode.v2.js"></script>
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<style>
    span{
        width: 150px;
        justify-content: center;
    }
</style>
<div class="row my-5 justify-content-center">
    <div class="col-md-10 col-lg-8">
        <h1 class="text-center mb-5">회원가입</h1>
        <form name="frm" class="card p-3" method="post" enctype="multipart/form-data">
            <div class="input-group mb-3">
                <span class="input-group-text">아 이 디</span>
                <input name="uid" class="form-control">
                <a class="btn btn-primary" id="btn-check">중복체크</a>
            </div>
            <div class="input-group mb-3">
                <span class="input-group-text">비밀번호</span>
                <input name="upass" type="password" class="form-control">
            </div>
            <div class="input-group mb-3">
                <span class="input-group-text">회 원 명</span>
                <input name="uname" class="form-control">
            </div>
            <div class="input-group mb-3">
                <span class="input-group-text">전화번호</span>
                <input name="phone" class="form-control">
            </div>
            <div class="input-group mb-3">
                <span class="input-group-text">주소1</span>
                <input name="address1" class="form-control">
                <a class="btn btn-primary" id="btn-search">주소검색</a>
            </div>
            <div class="input-group mb-3">
                <span class="input-group-text">주소2</span>
                <input name="address2" class="form-control">
            </div>
            <div class="text-center">
                <button class="btn btn-primary px-5">회원가입</button>
            </div>
        </form>
    </div>
</div>
```

```

<script>
    let check=false;
    $(frm).on("submit", function(e){
        e.preventDefault();
        const uid=$(frm.uid).val();
        const upass=$(frm.upass).val();
        const uname=$(frm.uname).val();
        if(uid==""){
            alert("아이디를 입력하세요!");
            $(frm.uid).focus();
        }else if(check==false){
            alert("아이디 중복 체크해주세요!");
        }else if(upass==""){
            alert("비밀 번호를 입력하세요!");
            $(frm.upass).focus();
        }else if(uname==""){
            alert("이름을 입력하세요!");
            $(frm.uname).focus();
        }else{
            if(confirm("회원가입을 하실래요?")) frm.submit();
        }
    });

    //주소검색 버튼을 클릭한 경우
    $("#btn-search").on("click", function(){
        new daum.Postcode({
            oncomplete: function(data){
                if(data.buildingName!="") $(frm.address1).val(data.address + " " + data.buildingName);
                else $(frm.address1).val(data.address);
            }
        }).open();
    });

    $("#btn-check").on("click", function(){ //중복체크 버튼을 클릭한 경우
        const uid=$(frm.uid).val();
        if(uid=="") {
            alert("아이디를 입력하세요!");
        }else {
            $.ajax({
                type:"post",
                url:"/user/login",
                data:{uid: $(frm.uid).val(), pass: ""},
                success:function(data){
                    if(data==0) {
                        alert("사용가능한 아이디입니다!");
                        check=true;
                    }else{
                        alert("이미 사용 중인 아이디입니다!");
                    }
                }
            });
        }
    });

    $(frm.uid).on("change", function(){ //아이디가 변경되는 경우
        check=false;
    });
</script>

```

- 회원정보 수정을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] UserDao.java

```
public class UserDao {
    ...
    //회원수정
    public void update(UserVO vo) {
        try {
            String sql="update users set uname=?,phone=?,address1=?,address2=?,photo=? where uid=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(6, vo.getUid());
            ps.setString(1, vo.getUsername());
            ps.setString(2, vo.getPhone());
            ps.setString(3, vo.getAddress1());
            ps.setString(4, vo.getAddress2());
            ps.setString(5, vo.getPhoto());
            ps.execute();
        }catch(Exception e) {
            System.out.println("회원수정:" + e.toString());
        }
    }
}
```

[src/main/java]-[controller] UserController.java

```
@WebServlet(value= {"/user/login", "/user/logout", "/user/read", "/user/insert", "/user/update"})
public class UserController extends HttpServlet {
    UserDao dao=new UserDao();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        ...
        switch(request.getServletPath()){
            ...
            case "/user/update":
                request.setAttribute("vo", dao.read(request.getParameter("uid")));
                request.setAttribute("pageName", "/user/update.jsp");
                dis.forward(request, response);
                break;
        }

        protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
            ...
            switch(request.getServletPath()) {
                ...
                case "/user/update":
                    //사진저장
                    multi=new MultipartRequest(request, "c:"+path, 1024*1024*10,"UTF-8", new DefaultFileRenamePolicy());
                    photo = multi.getFilesystemName("photo")==null ? multi.getParameter("old_photo"):
                        path + multi.getFilesystemName("photo");
                    vo=new UserVO();
                    vo.setUid(multi.getParameter("uid"));
                    vo.setUsername(multi.getParameter("uname"));
                    vo.setPhone(multi.getParameter("phone"));
                    vo.setAddress1(multi.getParameter("address1"));
                    vo.setAddress2(multi.getParameter("address2"));
                    vo.setPhoto(photo);
                    dao.update(vo);
                    response.sendRedirect("/user/read?uid=" + vo.getUid());
                    break;
            }
        }
    }
}
```

[src]-[main]-[webapp]-[user] update.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<style>
    ...
    #image {
        border-radius:40%;
        cursor:pointer;
        border: 2px solid gray;
    }
</style>
<div class="row my-5 justify-content-center">
    <div class="col-md-10 col-lg-8">
        <h1 class="text-center mb-5">회원정보수정</h1>
        <form name="frm" class="card p-3" method="post" action="/user/update" enctype="multipart/form-data">
            <div>
                <c:if test="${vo.photo==null || vo.photo==''}">
                    
                </c:if>
                <c:if test="${vo.photo!=null} && vo.photo!=''">
                    
                </c:if>
                <input type="file" name="photo" style="display:none;">
                <span class="w-100 ms-3"> ${vo.uname} ${vo.uid}</span>
                <input type="hidden" name="uid" value="${vo.uid}">
                <input type="hidden" name="old_photo" value="${vo.photo}" />
            </div><hr>
            <div class="input-group mb-3">
                <span class="input-group-text">비밀번호</span>
                <input name="upass" type="password" class="form-control" placeholder="변경 비밀번호">
                <a class="btn btn-primary" id="btn-change">변경</a>
            </div>
            ...
        </form>
    </div>
</div>
<script>
    $(frm).on("submit", function(e){
        e.preventDefault();
        const uname=$(frm.uname).val();
        if(uname==""){
            alert("이름을 입력하세요!");
            $(frm.uname).focus();
        }else{
            if(confirm("회원정보를 수정하실래요?")) frm.submit();
        }
    });

    $("#image").on("click", function(){ //이미지를 클릭한 경우
        $(frm.photo).click();
    });

    $(frm.photo).on("change", function(e){ //이미지파일을 선택한경우
        $("#image").attr("src", URL.createObjectURL(e.target.files[0]));
    });

    //주소검색 버튼을 클릭한 경우
    $("#btn-search").on("click", function(){ .. });
</script>
```

- 비밀번호 변경을 위한 DAO, Controller, JSP 페이지를 수정한다.

[src/main/java]-[model] UserDao.java

```
public class UserDao {
    ...
    //회원비밀번호 변경
    public void update(String uid, String upass) {
        try {
            String sql="update users set upass=? where uid=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, upass);
            ps.setString(2, uid);
            ps.execute();
        }catch(Exception e) {
            System.out.println("회원비밀번호변경:" + e.toString());
        }
    }
}
```

[src/main/java]-[controller] UserController.java

```
@WebServlet(value= {"/user/login", "/user/logout", "/user/read", "/user/insert", "/user/update", "/user/change"})
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    ...
    switch(request.getServletPath()) {
    ...
    case "/user/change":
        uid=request.getParameter("uid");
        upass=request.getParameter("upass");
        dao.update(uid, upass);
        break;
    }
}
```

[src]-[main]-[webapp]-[user] update.jsp

```
<script>
    ...
    //비밀번호 변경 버튼을 클릭한 경우
    $("#btn-change").on("click", function(){
        const uid=$(frm.uid).val();
        const upass=$(frm.upass).val();
        if(upass==""){
            alert("변경할 비밀번호를 입력하세요!");
            $(frm.upass).focus();
        }else{
            if(confirm("비밀번호를 변경하실래요?")){
                $.ajax({
                    type:"post",
                    url:"/user/change",
                    data:{uid, upass},
                    success:function(){
                        alert("비밀번호가 변경되었습니다!");
                        location.href="/user/login";
                    }
                });
            }
        }
    });
}
```


- 주문관리

- 주문관리를 위한 테이블과 VO를 작성한다.

구매 테이블 생성

```
create table purchase(  
    pid char(13) not null primary key,  
    uid varchar(20) not null,  
    recName varchar(20),  
    recPhone varchar(20),  
    recAddress1 varchar(200),  
    recAddress2 varchar(200),  
    purDate datetime default now(),  
    sum int default 0,  
    status int default 0, /* 0: 결제대기중 | 1.결제확인 | 2.배송준비중 | 3.배송완료 | 4.주문완료 */  
    foreign key(uid) references users(uid)  
);
```

주문 상품 테이블 생성

```
create table orders(  
    pid char(13) not null,  
    gid char(8) not null,  
    price int default 0,  
    qnt int default 0,  
    primary key(pid, gid)  
    foreign key(pid) references purchase(pid),  
    foreign key(gid) references goods(gid)  
);
```

[src/main/java]-[model] PurchaseVO.java

```
public class PurchaseVO extends UserVO{  
    private String pid;  
    private String recName;  
    private String recPhone;  
    private String recAddress1;  
    private String recAddress2;  
    private int sum;  
    private int status;  
    private String purDate;  
    //setter(), getter(), toString() 메서드를 generate한다.  
}
```

[src/main/java]-[model] Orders.java

```
public class OrderVO extends CartVO {  
    private String pid;  
    //setter(), getter(), toString() 메서드를 generate한다.  
}
```

view_purchase 생성

```
create view view_purchase as  
select p.*, uname from purchase p, users u where p.uid = u.uid;
```

view_orders 생성

```
create view view_orders as  
select o.*, title, image from orders o, goods g where o.gid=g.gid;
```

- 주문등록을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] Purchase.java

```
public class PurchaseDAO {
    public void insert(PurchaseVO vo) { //구매 등록
        try {
            String sql="insert into purchase(pid,uid,recName,recAddress1,recAddress2,recPhone,sum) values(?,?,?,?,?,?,?)";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, vo.getPid());
            ...
            ps.execute();
        }catch(Exception e) {
            System.out.println("구매 등록:" + e.toString());
        }
    }

    public void insert(OrderVO vo) { //주문상품 등록
        try {
            String sql="insert into orders(pid,gid,price,qnt) values(?,?,?,?)";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ...
            ps.execute();
        }catch(Exception e) {
            System.out.println("주문상품등록:" + e.toString());
        }
    }
}
```

[src/main/java]-[controller] PurchaseController.java

```
@WebServlet(value={"/purchase/insert", "/order/insert"});
public class PurchaseController extends HttpServlet {
    PurchaseDAO dao=new PurchaseDAO();

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        PrintWriter out=response.getWriter();
        switch(request.getServletPath()) {
            case "/purchase/insert":
                PurchaseVO vo=new PurchaseVO();
                UUID uuid=UUID.randomUUID();
                String pid=uuid.toString().substring(0,13);
                vo.setPid(pid);
                vo.setUid(request.getParameter("uid"));
                vo.setRecName(request.getParameter("uname"));
                ...
                vo.setSum(Integer.parseInt(request.getParameter("sum")));
                dao.insert(vo); //구매정보저장
                out.print(pid);
                break;
            case "/order/insert":
                OrderVO ovo=new OrderVO();
                ovo.setPid(request.getParameter("pid"));
                ovo.setGid(request.getParameter("gid"));
                ovo.setPrice(Integer.parseInt(request.getParameter("price")));
                ovo.setQnt(Integer.parseInt(request.getParameter("qnt")));
                dao.insert(ovo); //주문상품 저장
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[order] cart.jsp

```
<div class="row my-5" id="page_cart">
  <div class="col">
    <h1 class="text-center mb-5">장바구니</h1>
    <div id="div_cart"></div>
  </div>
</div>
<jsp:include page="order.jsp"/>

<!-- 카트목록 템플릿 -->
<script id="temp_cart" type="text/x-handlebars-template">
  <table class="table">
    ...
    {{#each .}}
    <tr class="tr" price="{{price}}">
      <td><input type="checkbox" class="chk" goods="{{toString @this}}"></td>
      <td class="gid">{{gid}}</td><td></td>
      <td class="text-truncate" style="max-width:250px;">{{title}}</td>
      <td>{{sum price 1}}</td>
      <td width="200">
        <input class="qnt" value="{{qnt}}" size=5 oninput="isNumber(this)">&nbsp;
        <button class="btn btn-primary btn-sm btn-update">수정</button>
      </td>
      <td>{{sum price qnt}}</td>
      <td><button class="btn btn-danger btn-sm" gid="{{gid}}">삭제</button></td>
    </tr>
    {{/each}}
  <tr>
    <td colspan="8" class="text-end pe-5"><h5>총합계: <span id="sum">0원</span></h5></td>
  </tr>
</table>
<div class="text-center my-3">
  <button class="btn btn-primary px-5" id="btn_order">주문하기</button>
</div>
</script>
<script>
  getList();
  const uid="{{user.uid}}";
  ...
  $("#div_cart").on("click", "#btn_order", function() { //주문하기 버튼을 클릭한 경우
    if(uid==""){ //로그인이 안 된 경우
      location.href="/user/login?target=/cart/list";
    }else{ //로그인한 경우
      const chk=$("#div_cart .chk:checked").length;
      if(chk==0) {
        alert("주문할 상품을 선택하세요!");
      }else{
        let data=[];
        $("#div_cart .chk:checked").each(function(){
          const goods=$(this).attr("goods");
          data.push(JSON.parse(goods));
        });
        getOrder(data);
        $("#page_order").show();
        $("#page_cart").hide();
      }
    }
  });
</script>
```

[src]-[main]-[webapp]-[order] order.jsp ①

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<style>
    span {
        width: 150px;
        justify-content: center;
    }
</style>
<div class="row my-5" id="page_order" style="display:none;">
    <div class="col">
        <h1 class="text-center mb-5">주문하기</h1>
        <div id="div_order"></div>
        <h1 class="text-center my-5">주문자정보</h1>
        <form name="frm" class="card p-3">
            <div class="input-group mb-3">
                <span class="input-group-text">받으실분</span>
                <input name="uname" class="form-control" value="{user.uname}">
            </div>
            <div class="input-group mb-3">
                <span class="input-group-text">전화</span>
                <input name="phone" class="form-control" value="{user.phone}">
            </div>
            <div class="input-group mb-3">
                <span class="input-group-text">주소</span>
                <input name="address1" class="form-control" value="{user.address1}">
                <a class="btn btn-primary" id="btn-search">주소검색</a>
            </div>
            <div class="input-group mb-3">
                <span class="input-group-text">상세주소</span>
                <input name="address2" class="form-control" value="{user.address2}">
            </div>
            <input name="sum" type="hidden">
            <div class="text-center my-3">
                <button class="btn btn-primary px-5">주문하기</button>
            </div>
        </form>
    </div>
</div>
<!-- 주문상품목록 템플릿 -->
<script id="temp_order" type="x-handlebar-template">
    <table class="table">
        {{#each .}}
        <tr class="tr" price="{price}" qnt="{qnt}" gid="{gid}">
            <td>{{gid}}</td>
            <td></td>
            <td class="text-truncate" style="max-width:200px;">{{title}}</td>
            <td>{{sum price 1}}</td>
            <td>{{qnt}}</td>
            <td>{{sum price qnt}}</td>
        </tr>
        {{/each}}
        <tr>
            <td colspan="6" class="text-end pt-3">
                <h5>총합계: <span id="osum">0원</span></h5>
            </td>
        </tr>
    </table>
</script>
```

```

<script>
    $(frm).on("submit", function(e){ //주문하기 버튼을 클릭한 경우
        e.preventDefault();
        if(confirm("위 상품을 주문하실래요?")) { //구매자정보등록
            $.ajax({
                type:"post",
                url:"/purchase/insert",
                data:{
                    uid:"${user.uid}", uname: $(frm.uname).val(),
                    address1:$(frm.address1).val(), address2:$(frm.address2).val(),
                    phone:$(frm.phone).val(), sum:$(frm.sum).val()
                },
                success:function(data){ //주문 상품등록
                    const pid=data;
                    $("#div_order .tr").each(function(){
                        const gid=$(this).attr("gid");
                        const qnt=$(this).attr("qnt");
                        const price=$(this).attr("price");
                        $.ajax({
                            type:"post",
                            url:"/order/insert",
                            data:{pid:pid, gid:gid, qnt:qnt, price:price},
                            success:function(){ //카트에서 상품삭제
                                $.ajax({
                                    type:"get",
                                    url:"/cart/delete",
                                    data:{gid:gid},
                                    success:function(){ }
                                });
                            } //카트에서 상품삭제 완료
                        });
                    });
                    alert("주문이 완료되었습니다.");
                    location.href="/";
                } //주문 상품 등록 완료
            });
        } //구매자정보등록 완료
    }); //주문하기 버튼을 클릭한 경우 완료

    $("#btn-search").on("click", function(){ ... }) //주소검색 버튼을 클릭한 경우

    function getOrder(data){
        const temp=Handlebars.compile($("#temp_order").html());
        $("#div_order").html(temp(data));
        getOrderSum();
    }

    function getOrderSum(){
        let sum=0;
        $("#div_order .tr").each(function(){
            const price=$(this).attr("price");
            const qnt=$(this).attr("qnt");
            sum += price*qnt;
        });
        $(frm.sum).val(sum);
        $("#osum").html(sum.toString().replace(/\B(?=(\d{3})+(?! \d))/g, ", "));
    }
}
</script>

```

- 주문목록 출력을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] PurchaseDAO.java

```
public class PurchaseDAO {
    SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    public int total(String key, String query) { //검색수
        int total=0;
        try {
            String sql="select count(*) from view_purchase where " + key + " like ?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, "%" + query + "%");
            ResultSet rs=ps.executeQuery();
            if(rs.next()) total=rs.getInt("count(*)");
        }catch(Exception e) { System.out.println("구매검색수:" + e.toString()); }
        return total;
    }
    public ArrayList<PurchaseVO> list(String key, String query, int page){ //구매목록
        ArrayList<PurchaseVO> array=new ArrayList<PurchaseVO>();
        try {
            String sql="select * from view_purchase where " + key + " like ? order by purdate desc limit ?, 5";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, "%" + query + "%");
            ps.setInt(2, (page-1)* 5);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                PurchaseVO vo=new PurchaseVO();
                ...
                array.add(vo);
            }
        }catch(Exception e) { System.out.println("구매목록:" + e.toString()); }
        return array;
    }
}
```

[src/main/java]-[controller] PurchaseController.java

```
@WebServlet(value={"/purchase/insert", "/order/insert", "/purchase/list.json", "/purchase/list", "/purchase/total" })
PurchaseDAO dao=new PurchaseDAO();
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out=response.getWriter();
    RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
    switch(request.getServletPath()) {
        case "/purchase/list.json": //실행: /purchase/list.json?page=1&key=pid&query=
            String key=request.getParameter("key");
            String query=request.getParameter("query");
            int page=Integer.parseInt(request.getParameter("page"));
            Gson gson=new Gson();
            out.println(gson.toJson(dao.list(key, query, page)));
            break;
        case "/purchase/list":
            request.setAttribute("pageName", "/purchase/list.jsp");
            dis.forward(request, response);
            break;
        case "/purchase/total": //실행: /purchase/total?key=pid&query=
            key=request.getParameter("key");
            query=request.getParameter("query");
            out.println(dao.total(key, query));
            break;
    }
}
```

```

<div class="row my-5">
  <div class="col">
    <h1 class="text-center mb-5">구매목록</h1>
    <div class="row">
      <form class="col-md-4" name="frm">
        <div class="input-group">
          <select class="form-select" name="key">
            <option value="uid">회원아이디</option>
            <option value="uname" selected>회원이름</option>
            <option value="recName" selected>받은이</option>
            <option value="recAddress1">배송지주소</option>
            <option value="recPhone">회원전화</option>
          </select>&nbsp;  
          <input class="form-control" placeholder="검색어" name="query">
          <input type="submit" value="검색" class="btn btn-primary">
        </div>
      </form>
      <div class="col-md-4"></div>
      <div class="col-md-4">
        <select class="form-select" id="sel-status">
          <option value="0">결제대기중</option>
          <option value="1">결제완료</option>
          <option value="2">배송준비중</option>
          <option value="3">배송중</option>
          <option value="4">배송완료</option>
          <option value="">모든구매</option>
        </select>
      </div>
    </div>
    <hr>
    <div id="div_purchase" class="row"></div>
    <div id="pagination" class="pagination justify-content-center mt-3"></div>
  </div>
</div>
<script id="temp_purchase" type="x-handlebars-template">
  <table class="table">
    {{#each .}}
    <tr>
      <td class="pid"><a href="/purchase/read?pid={{pid}}">{{pid}}</a></td>
      <td>{{uname}} {{uid}}</td>
      <td>{{recAddress1}}</td>
      <td>{{purDate}}</td>
      <td>
        <select class="form-select status">
          <option value="0" {{select status 0}}>결제대기중</option>
          ...
        </select>
      </td>
      <td><button class="btn btn-primary btn-sm btn-update">상태변경</button></td>
    </tr>
    {{/each}}
  </table>
</script>
<script>
  Handlebars.registerHelper("select", function(status, value){
    if(status==value) return "selected";
  });
</script>

```

```

<script>
    let page=1;
    let query=$(frm.query).val();
    let key=$(frm.key).val();

    //상태조작이 변경된 경우
    $("#sel-status").on("change", function(){
        key="status";
        query=$(this).val();
        page=1;
        getTotal();
    });

    $(frm).on("submit", function(e){
        e.preventDefault();
        query=$(frm.query).val();
        key=$(frm.key).val();
        page=1;
        getTotal();
    });

    function getList(){
        $.ajax({
            type:"get",
            url:"/purchase/list.json",
            data:{page:page, key:key, query:query},
            dataType:"json",
            success:function(data){
                const temp=Handlebars.compile($("#temp_purchase").html());
                $("#div_purchase").html(temp(data));
            }
        })
    }

    getTotal();
    function getTotal(){
        $.ajax({
            type:"get",
            url:"/purchase/total",
            data:{key:key, query:query},
            success:function(data){
                $("#pagination").hide();
                if(data==0){
                    $("#div_purchase").html("<h3 class='text-center my-5'>검색결과가 없습니다.</h3>");
                } else if(data > 5){
                    const totalPages=Math.ceil(data/5);
                    $("#pagination").twbsPagination("changeTotalPages", totalPages, page);
                    $("#pagination").show();
                }
            }
        });
    }

    $('#pagination').twbsPagination({
        ...
    });
</script>

```


- 구매 정보 조회를 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] PurchaseDAO.java

```
public class PurchaseDAO {
    SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

    //구매정보읽기
    public PurchaseVO read(String pid) {
        PurchaseVO vo=new PurchaseVO();
        try {
            String sql="select * from view_purchase where pid=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, pid);
            ResultSet rs=ps.executeQuery();
            if(rs.next()) {
                vo.setPid(rs.getString("pid"));
                vo.setUid(rs.getString("uid"));
                vo.setUname(rs.getString("uname"));
                vo.setRecName(rs.getString("recName"));
                vo.setRecAddress1(rs.getString("recAddress1"));
                vo.setRecAddress2(rs.getString("recAddress2"));
                vo.setRecPhone(rs.getString("recPhone"));
                vo.setPurDate(sdf.format(rs.getTimestamp("purDate")));
                vo.setStatus(rs.getInt("status"));
                vo.setSum(rs.getInt("sum"));
            }
        }catch(Exception e) {
            System.out.println("구매정보:" + e.toString());
        }
        return vo;
    }

    //구매한 상품목록
    public ArrayList<OrderVO> orderList(String pid){
        ArrayList<OrderVO> array=new ArrayList<OrderVO>();
        try {
            String sql="select * from view_orders where pid=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, pid);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                OrderVO vo=new OrderVO();
                vo.setPid(rs.getString("pid"));
                vo.setGid(rs.getString("gid"));
                vo.setPrice(rs.getInt("price"));
                vo.setQnt(rs.getInt("qnt"));
                vo.setTitle(rs.getString("title"));
                vo.setImage(rs.getString("image"));
                array.add(vo);
            }
        }catch(Exception e) {
            System.out.println("구매한 상품목록:" + e.toString());
        }
        return array;
    }
}
```

[src/main/java]-[controller] PurchaseController.java

```
@WebServlet(value={ ... ,"/purchase/list.json", "/purchase/list","/purchase/total", "/purchase/read"})
public class PurchaseController extends HttpServlet {
    PurchaseDAO dao=new PurchaseDAO();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            case "/purchase/read":
                String pid=request.getParameter("pid");
                request.setAttribute("vo", dao.read(pid));
                request.setAttribute("array", dao.orderList(pid));
                request.setAttribute("pageName", "/purchase/read.jsp");
                dis.forward(request, response);
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[purchase] read.jsp

```
<%@taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">구매정보</h1>
        <div class="card">
            <div class="row p-3">
                <div class="col-lg-3">주문번호: ${vo.pid}</div>
                <div class="col-lg-3">주문자: ${vo.uname} ${vo.uid}</div>
                <div class="col-lg-3">전화번호: ${vo.recPhone}</div>
                <div class="col-lg-3">주문일: ${vo.purDate}</div>
            </div>
            <hr>
            <div class="row p-3">
                <div class="col-lg-6">주소: ${vo.recAddress1} ${vo.recAddress2}</div>
                <div class="col">주문총금액: <fmt:formatNumber value="${vo.sum}" pattern="#,###원"/></div>
                <div class="col" style="color:red">주문상태:
                    <c:out value="${vo.status==0 ? '결제대기중:'}"/>
                    <c:out value="${vo.status==1 ? '결제완료:'}"/>
                    <c:out value="${vo.status==2 ? '배송준비중:'}"/>
                    <c:out value="${vo.status==3 ? '배송중:'}"/>
                    <c:out value="${vo.status==4 ? '배송완료:'}"/>
                </div>
            </div>
        </div>
    </div>
    <h1 class="text-center my-5">주문상품</h1>
    <table class="table">
        <c:forEach items="${array}" var="gvo">
            <tr>
                <td><a href="/goods/read?gid=${gvo.gid}">${gvo.gid}</a></td>
                <td></td>
                <td>${gvo.title}</td>
                <td><fmt:formatNumber value="${gvo.price}" pattern="#,###원"/></td>
                <td>${gvo.qnt}</td>
                <td><fmt:formatNumber value="${gvo.price*gvo.qnt}" pattern="#,###원"/></td>
            </tr>
        </c:forEach>
    </table>
</div>
</div>
```

- 리뷰관리

- 리뷰 관리를 위한 테이블과 VO를 작성한다.

리뷰 테이블 생성

```
create table reviews(  
    rid int auto_increment primary key,  
    gid char(8) not null  
    uid varchar(20) not null,  
    content text,  
    revDate datetime default now(),  
    foreign key(gid) references goods(gid),  
    foreign key(uid) references users(uid)  
)
```

[src/main/java]-[model] ReviewVO.java

```
public class ReviewVO extends UserVO {  
    private int rid;  
    private String gid;  
    private String revDate;  
    private String content;  
    //setter(), getter(), toString() 메서드를 generate한다.  
}
```

[src/main/java]-[model] ReviewDAO.java

```
public class ReviewDAO {  
    public void insert(ReviewVO vo) { //리뷰등록  
        try {  
            String sql="insert into reviews(gid, uid, content) values(?,?,?)";  
            PreparedStatement ps=Database.CON.prepareStatement(sql);  
            ps.setString(1, vo.getGid());  
            ps.setString(2, vo.getUid());  
            ps.setString(3, vo.getContent());  
            ps.execute();  
        } catch (Exception e) {  
            System.out.println("리뷰등록:" + e.toString());  
        }  
    }  
}
```

[src/main/java]-[controller] ReviewController.java

```
@WebServlet(value= {"/review/insert"})  
public class ReviewController extends HttpServlet {  
    ReviewDAO dao=new ReviewDAO();  
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        switch(request.getServletPath()) {  
            case "/review/insert":  
                ReviewVO vo=new ReviewVO();  
                vo.setGid(request.getParameter("gid"));  
                vo.setUid(request.getParameter("uid"));  
                vo.setContent(request.getParameter("content"));  
                dao.insert(vo);  
                break;  
            }  
        }  
    }  
}
```

[src]-[main]-[webapp]-[goods] review.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<style>
    .content {
        font-size:0.8rem;
    }
    textarea {
        overflow:hidden;
    }
    .btn-sm {
        font-size:0.7rem;
    }
</style>
<div>
    <form name="frm">
        <c:if test="${user != null}">
            <textarea name="content" rows="3" class="form-control content" placeholder="리뷰내용을 입력하세요!"></textarea>
            <div class="text-end mt-2">
                <button class="btn btn-primary px-5">등록</button>
            </div>
        </c:if>
        <c:if test="${user == null}">
            <div class="text-end">
                <a href="/user/login?target=/goods/read?gid=${vo.gid}" class="btn btn-primary px-5">리뷰작성</a>
            </div>
        </c:if>
    </form>
</div>
<script>
<script>
    const uid="${user.uid}";
    const gid="${vo.gid}";

    $(frm).on("submit", function(e){
        e.preventDefault();
        const content=$(frm.content).val();
        if(content==""){
            alert("리뷰 내용을 입력하세요!");
            $(frm.content).focus();
        }else{
            $.ajax({
                type:"post",
                url:"/review/insert",
                data:{gid, uid, content},
                success:function(){
                    alert("리뷰가 저장되었습니다.");
                    $(frm.content).val("");
                    getList();
                }
            });
        }
    });
</script>
```

[src]-[main]-[webapp]-[goods] read.jsp

```
<div class="tab-content pt-5" id="myTabContent">
  <!-- 상세설명 -->
  <div class="tab-pane fade show active" id="detail-tab-pane" role="tabpanel" aria-labelledby="detail-tab">...</div>
  <!-- 리뷰작성 -->
  <div class="tab-pane fade" id="review-tab-pane" role="tabpanel" aria-labelledby="review-tab" tabindex="1">
    <jsp:include page="review.jsp"/>
  </div>
  <!-- 첨부이미지 -->
  <div class="tab-pane fade" id="attach-tab-pane" role="tabpanel" aria-labelledby="attach-tab">...</div>
</div>
```

- 리뷰 목록 출력을 위한 DAO, Controller, JSP 페이지를 수정한다.

[src/main/java]-[model] ReviewDAO.java

```
public class ReviewDAO {
    SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

    //특정상품의 리뷰목록
    public ArrayList<ReviewVO> list(String gid){
        ArrayList<ReviewVO> array=new ArrayList<ReviewVO>();
        try {
            String sql="select * from view_reviews where gid=? order by rid desc";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, gid);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                ReviewVO vo=new ReviewVO();
                vo.setRid(rs.getInt("rid"));
                vo.setGid(rs.getString("gid"));
                vo.setUid(rs.getString("uid"));
                vo.setContent(rs.getString("content"));
                vo.setRevDate(sdf.format(rs.getTimestamp("revDate")));
                vo.setPhoto(rs.getString("photo"));
                array.add(vo);
            }
        }catch(Exception e) {
            System.out.println("리뷰목록:" + e.toString());
        }
        return array;
    }
}
```

[src/main/java]-[controller] ReviewController.java

```
@WebServlet(value= {"/review/insert", "/review/list.json"})
...
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out=response.getWriter();

    switch(request.getServletPath()) {
        case "/review/list.json":
            Gson gson=new Gson();
            out.println(gson.toJson(dao.list(request.getParameter("gid"))));
            break;
    }
}
```

```
[src]-[main]-[webapp]-[goods] review.jsp
```

```
<form name="frm"> ... </form>
<div id="div_review" class="my-5"></div>

<!-- 리뷰목록 템플릿 -->
<script id="temp_review" type="x-handlebars-template">
  {{#each .}}
    <div class="row">
      <div class="col-md-1">
        <div>{{uid}}</div>
      </div>
      <div class="col">
        <div>
          <div>{{revDate}}</div>
          <div class="display {{rid}}" rid="{{rid}}">
            <div class="ellipsis content" style="cursor:pointer">{{content}}</div>
            <div class="text-end mt-2" style="{{show uid}}">
              <button class="btn btn-primary btn-sm btn-update">수정</button>
              <button class="btn btn-danger btn-sm btn-delete">삭제</button>
            </div>
          </div>
          <div class="update {{rid}}" rid="{{rid}}" style="display:none">
            <textarea class="content form-control" rows="10">{{content}}</textarea>
            <div class="text-end mt-2">
              <button class="btn btn-primary btn-sm btn-save">저장</button>
              <button class="btn btn-secondary btn-sm btn-cancel">취소</button>
            </div>
          </div>
        </div>
      </div>
    </div>
  </each>
</script>
<script>
  Handlebars.registerHelper("show", function(writer){
    if(uid!=writer) return "display:none";
  })
</script>
<script>
  ...
  $("#div_review").on("click", ".content", function(){ //각 댓글의 내용을 클릭한 경우
    $(this).toggleClass("ellipsis");
  });

  getList();
  function getList(){
    $.ajax({
      type:"get",
      url:"/review/list.json",
      data:{gid},
      dataType:"json",
      success:function(data){
        const temp=Handlebars.compile($("#temp_review").html());
        $("#div_review").html(temp(data));
      }
    });
  }
</script>
```

- 리뷰 삭제와 리뷰 내용 수정을 위한 DAO, Controller와 JSP 페이지를 수정한다.

[src/main/java]-[model] ReviewDAO.java

```
public class ReviewDAO {
    ...
    //리뷰삭제
    public void delete(int rid) {
        try {
            String sql="delete from reviews where rid=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setInt(1, rid);
            ps.execute();
        }catch(Exception e) {
            System.out.println("리뷰삭제:" + e.toString());
        }
    }

    //리뷰수정
    public void update(int rid, String content) {
        try {
            String sql="update reviews set content=? where rid=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, content);
            ps.setInt(2, rid);
            ps.execute();
        }catch(Exception e) {
            System.out.println("리뷰수정:" + e.toString());
        }
    }
}
```

[src/main/java]-[controller] ReviewController.java

```
@WebServlet(value= {"/review/insert", "/review/list.json", "/review/delete", "/review/update"})
public class ReviewController extends HttpServlet {
    ...
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            case "/review/insert":
                ReviewVO vo=new ReviewVO();
                vo.setGid(request.getParameter("gid"));
                vo.setUid(request.getParameter("uid"));
                vo.setContent(request.getParameter("content"));
                System.out.println(vo.toString());
                dao.insert(vo);
                break;
            case "/review/delete":
                int rid=Integer.parseInt(request.getParameter("rid"));
                dao.delete(rid);
                break;
            case "/review/update":
                rid=Integer.parseInt(request.getParameter("rid"));
                String content=request.getParameter("content");
                dao.update(rid, content);
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[goods] review.jsp

<!-- 리뷰목록 템플릿 -->

<script id="temp_review" type="x-handlebars-template">

{{#each .}}

<div class="display{{rid}}" rid="{{rid}}">

<div class="ellipsis content" style="cursor:pointer">{{content}}</div>

<div class="text-end mt-2" style="{{show uid}}">

<button class="btn btn-primary btn-sm btn-update">수정</button>

<button class="btn btn-danger btn-sm btn-delete">삭제</button>

</div>

</div>

<div class="update{{rid}}" rid="{{rid}}" style="display:none;">

<textarea class="content form-control" rows="10">{{content}}</textarea>

<div class="text-end mt-2">

<button class="btn btn-primary btn-sm btn-save">저장</button>

<button class="btn btn-secondary btn-sm btn-cancel">취소</button>

</div>

</div>

{{/each}}

</script>

<script>

...

\$("#div_review").on("click", ".btn-update, .btn-cancel", function(){ //각 댓글의 수정버튼이나 취소버튼을 클릭한 경우

const rid=\$(this).parent().parent().attr("rid");

\$("#div_review .display" + rid).toggle();

\$("#div_review .update" + rid).toggle();

});

\$("#div_review").on("click", ".btn-delete", function(){ //각 댓글의 삭제버튼을 클릭한 경우

const row=\$(this).parent().parent();

const rid=row.attr("rid");

if(confirm(rid + "번 댓글을 삭제하실래요?")) {

\$.ajax({

type:"post",

url:"/review/delete",

data:{rid},

success:function(){

getList();

}

});

}

});

\$("#div_review").on("click", ".btn-save", function(){ //각 댓글의 저장버튼을 클릭한 경우

const row=\$(this).parent().parent();

const rid=row.attr("rid");

const content=row.find(".content").val();

if(confirm(rid + "번 댓글의 내용을 수정하실래요?")) {

\$.ajax({

type:"post",

url:"/review/update",

data:{rid, content},

success:function(){

getList();

}

});

}

});

</script>

- **좋아요 관리**

- 좋아요 관리를 위한 테이블을 작성한다.

좋아요 테이블 생성

```
create table favorite(  
    gid char(8) not null,  
    uid varchar(20) not null,  
    regDate datetime default now(),  
    primary key(gid, uid),  
    foreign key references goods(gid),  
    foreign key references users(uid)  
);
```

- 좋아요 추가 삭제를 위한 DAO, Controller를 작성하고 JSP 페이지를 수정한다.

[src/main/java]-[model] FavoriteDAO.java

```
public class FavoriteDAO {  
    public JSONObject read(String gid, String uid) { //좋아요 등록여부와 총 좋아요 횟수  
        JSONObject obj=new JSONObject();  
        try {  
            String sql=" select count(*) fcnt,";  
            sql+="(select count(*) from favorite where gid=? and uid=?) ucnt ";  
            sql+="from favorite where gid=?";  
            PreparedStatement ps=Database.CON.prepareStatement(sql);  
            ps.setString(1, gid);  
            ps.setString(2, uid);  
            ps.setString(3, gid);  
            ResultSet rs=ps.executeQuery();  
            if(rs.next()) {  
                obj.put("fcnt", rs.getInt("fcnt"));  
                obj.put("ucnt", rs.getInt("ucnt"));  
            }  
        } catch (Exception e) { System.out.println("좋아요 등록여부:" + e.toString()); }  
        return obj;  
    }  
  
    public void insert(String gid, String uid) { //좋아요 등록  
        try {  
            String sql="insert into favorite(gid, uid) values(?, ?)";  
            PreparedStatement ps=Database.CON.prepareStatement(sql);  
            ps.setString(1, gid);  
            ps.setString(2, uid);  
            ps.execute();  
        } catch (Exception e) { System.out.println("좋아요 등록:" + e.toString()); }  
    }  
  
    public void delete(String gid, String uid) { //좋아요 삭제  
        try {  
            String sql="delete from favorite where gid=? and uid=?";  
            PreparedStatement ps=Database.CON.prepareStatement(sql);  
            ps.setString(1, gid);  
            ps.setString(2, uid);  
            ps.execute();  
        } catch (Exception e) { System.out.println("좋아요 삭제:" + e.toString()); }  
    }  
}
```

```
[src/main/java]-[controller] FavoriteController.java
```

```
@WebServlet(value={"/favorite/read", "/favorite/insert", "/favorite/delete"})
FavoriteDAO dao=new FavoriteDAO();
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out=response.getWriter();
    String uid=request.getParameter("uid");
    String gid=request.getParameter("gid");
    switch(request.getServletPath()) {
    case "/favorite/read":
        out.println(dao.read(gid, uid));
        break;
    case "/favorite/insert":
        dao.insert(gid, uid);
        break;
    case "/favorite/delete":
        dao.delete(gid, uid);
        break;
    }
}
```

```
[src]-[main]-[webapp]-[goods] read.jsp
```

```
<style>
    .bi-suit-heart, .bi-suit-heart-fill {
        color: red; }
    .bi {
        font-size:1.2rem;
        cursor: pointer;
    }
    #rcnt, #fcnt {
        font-size:0.7rem; }
</style>
...
<div class="col-6">
    <button class="btn btn-primary" id="btn-cart">장바구니</button>
</div>
<div class="col-6 text-end" id="count">
    <i id="heart" class="bi bi-suit-heart ms-3"></i><span id="fcnt"></span>
    <i class="bi bi-chat-left-text ms-3" id="btn-tab-review"></i><span id="rcnt"></span>
</div>
<script>
    getFavorite();
    function getFavorite() {
        $.ajax({
            type:"get",
            url:"/favorite/read",
            data:{gid, uid},
            dataType:"json",
            success:function(data){
                $("#fcnt").html(data.fcnt);
                if(data.ucnt==1) $("#heart").addClass("bi-suit-heart-fill").removeClass("bi-suit-heart");
                else $("#heart").addClass("bi-suit-heart").removeClass("bi-suit-heart-fill");
            }
        });
    }
</script>
```

[src]-[main]-[webapp]-[goods] read.jsp

```
<script>
...
//빈 하트를 클릭한 경우
$("#count").on("click", ".bi-suit-heart",function(){
    if(uid=="") {
        location.href="/user/login?target=/goods/read?gid=" + gid;
    }else{
        if(confirm("좋아요!를 추가하실래요")){
            $.ajax({
                type:"get",
                url:"/favorite/insert",
                data:{gid, uid},
                success:function(){
                    getFavorite();
                }
            });
        }
    }
});

//채운 하트를 클릭한 경우
$("#count").on("click", ".bi-suit-heart-fill", function(){
    if(confirm("좋아요!를 삭제하실래요?")){
        $.ajax({
            type:"get",
            url:"/favorite/delete",
            data:{gid, uid},
            success:function(){
                getFavorite();
            }
        });
    }
});

$("#btn-tab-review").on("click", function() { //댓글 아이콘을 클릭한 경우
    $("#review-tab").click();
});
</script>
```

[src]-[main]-[webapp]-[goods] review.jsp

```
<script>
...
getList();
function getList() {
    $.ajax({
        type:"get",
        url:"/review/list.json",
        data:{gid},
        dataType:"json",
        success:function(data){
            $("#rcnt").html(data.length);
            const temp=Handlebars.compile($("#temp_review").html());
            $("#div_review").html(temp(data));
        }
    });
}
</script>
```

- about 페이지에 좋아요 횟수와 리뷰수를 출력한다.

```
[src/main/java]-[model] CountVO.java
```

```
public class CountVO {
    private int ucnt;
    private int fcnt;
    private int rcnt;
    //setter(), getter(), toString() 메서드를 generate한다.
}
```

```
[src/main/java]-[model] GoodsVO.java
```

```
public class GoodsVO extends CountVO { ... }
```

리뷰수와 종와요수 출력을 위한 view 생성

```
create view view_goods as
select *, (select count(*) from reviews where gid=g.gid) rcnt, (select count(*) from favorite where gid=g.gid) fcnt
from goods g;
```

```
[src/main/java]-[model] GoodsDAO.java
```

```
public ArrayList<GoodsVO> list(String query, int page, String uid){
    ArrayList<GoodsVO> array=new ArrayList<GoodsVO>();
    try {
        String sql="select *, (select count(*) from favorite where uid=? and gid=g.gid) ucnt";
        sql+=" from view_goods g where title like ? order by regDate desc limit ?,6";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, uid);
        ps.setString(2, "%" + query + "%");
        ps.setInt(3, (page-1)*6);
        ResultSet rs=ps.executeQuery();
        while(rs.next()) {
            ...
            vo.setFcnt(rs.getInt("fcnt"));
            vo.setRcnt(rs.getInt("rcnt"));
            vo.setUcnt(rs.getInt("ucnt"));
            array.add(vo);
        }
    } catch (Exception e) { System.out.println("상품목록:" + e.toString()); }
    return array;
}
```

```
[src/main/java]-[model] GoodsController.java
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    switch(request.getServletPath()) {
        ...
        case "/goods/list.json": //실행: /goods/list.json?page=1&query=&uid=
            page=Integer.parseInt(request.getParameter("page"));
            query=request.getParameter("query");
            String uid=request.getParameter("uid");
            Gson gson=new Gson();
            out.print(gson.toJson(gdao.list(query, page, uid)));
            break;
    }
}
```

[src]-[main]-[webapp] about.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<style>
    .title, .price {
        font-size:12px;
    }

    .bi-suit-heart, .bi-suit-heart-fill {
        color: red;
    }

    .bi {
        font-size:0.7rem;
    }

    .rcnt, .fcnt {
        font-size:0.5rem;
    }
</style>

<!-- 상품목록 템플릿 -->
<script id="temp_goods" type="x-handlebars/template">
    {{#each .}}
        <div class="col-6 col-md-4 col-lg-2 mb-3">
            <div class="card">
                <div class="card-body">
                    
                    <div class="ellipsis title mt-2">{{title}}</div>
                    <div class="price">{{fmtPrice price}}</div>
                </div>
                <div class="card-footer text-end">
                    <i id="heart" class="bi {{ucnt ucnt}}"></i>
                    <span class="fcnt">{{fcnt}}</span>
                    <i class="bi bi-chat-left-text ms-2"></i>
                    <span class="rcnt">{{rcnt}}</span>
                </div>
            </div>
        </div>
    {{/each}}
</script>

<script>
    Handlebars.registerHelper("fmtPrice", function(price){
        return price.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",") + "원";
    });

    Handlebars.registerHelper("ucnt", function(ucnt){
        if(ucnt==0) {
            return "bi-suit-heart";
        }else {
            return "bi-suit-heart-fill";
        }
    });
</script>
```

- 관리자 페이지

관리자 추가

```
insert into users(uid, upass, uname, role) values('admin', 'pass', '관리자', 1);
```

- header.jsp 페이지의 메뉴를 수정한다.

[src]-[main]-[webapp] header.jsp

```
<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav me-auto mb-2 mb-lg-0">
    <c:if test="${user.role==1}">
      <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="/goods/search">상품검색</a>
      </li>
      <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="/goods/list">상품목록</a>
      </li>
      <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="/user/list">회원목록</a>
      </li>
      <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="/purchase/list">구매목록</a>
      </li>
    </c:if>
    <c:if test="${user.role==2 || user==null}">
      <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="/cart/list">장바구니</a>
      </li>
    </c:if>
  </ul>
  <ul class="navbar-nav mb-2 mb-lg-0">
    <c:if test="${user==null}">
      <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="/user/login">로그인</a>
      </li>
    </c:if>
    <c:if test="${user!=null}">
      <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="/user/read?uid=${user.uid}">${user.uname}님</a>
      </li>
      <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="/user/logout">로그아웃</a>
      </li>
    </c:if>
  </ul>
</div>
```

- 회원목록 출력을 위한 VO, DAO, Controller를 수정하고 JSP 페이지를 작성한다.

[src/main/java]-[model] UserVO.java

```
public class UserVO extends CountVO {
  ...
}
```

```
public class UserDao {
    SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    ...
    //회원목록
    public ArrayList<UserVO> list(String key, String query, int page){
        ArrayList<UserVO> array=new ArrayList<UserVO>();
        try {
            String sql="select *,";
            sql+="(select count(*) from purchase where uid=u.uid) ucnt,";
            sql+="(select count(*) from reviews where uid=u.uid) rcnt,";
            sql+="(select count(*) from favorite where uid=u.uid) fcnt ";
            sql+="from users u where " + key + " like ? order by regDate desc limit ?, 6";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, "%" + query + "%");
            ps.setInt(2, (page-1)*6);
            ResultSet rs=ps.executeQuery();
            while(rs.next()) {
                UserVO vo=new UserVO();
                vo.setUid(rs.getString("uid"));
                vo.setUpass(rs.getString("upass"));
                vo.setUname(rs.getString("uname"));
                vo.setPhone(rs.getString("phone"));
                vo.setAddress1(rs.getString("address1"));
                vo.setAddress2(rs.getString("address2"));
                vo.setRegDate(sdf.format(rs.getTimestamp("regDate")));
                vo.setPhoto(rs.getString("photo"));
                vo.setUcnt(rs.getInt("ucnt"));
                vo.setFcnt(rs.getInt("fcnt"));
                vo.setRcnt(rs.getInt("rcnt"));
                array.add(vo);
            }
        }catch(Exception e) {
            System.out.println("회원목록:" + e.toString());
        }
        return array;
    }

    //검색수
    public int total(String key, String query) {
        int total=0;
        try {
            String sql="select count(*) from users where " + key + " like ?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setString(1, "%" + query + "%");
            ResultSet rs=ps.executeQuery();
            if(rs.next()) {
                total=rs.getInt("count(*)");
            }
        }catch(Exception e) {
            System.out.println("회원 검색수:" + e.toString());
        }
        return total;
    }
}
```

[src/main/java]-[model] UserController.java

```
@WebServlet(value= {"/user/login", "/user/logout", ... ,"/user/list.json", "/user/list", "/user/total"})
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    switch(request.getServletPath()){
        case "/user/list.json": //실행 /user/list.json?page=1&key=uid&query=
            String key=request.getParameter("key");
            String query=request.getParameter("query");
            int page=Integer.parseInt(request.getParameter("page"));
            ArrayList<UserVO> array=dao.list(key, query, page);
            Gson gson=new Gson();
            out.print(gson.toJson(array));
            break;
        case "/user/total":
            key=request.getParameter("key");
            query=request.getParameter("query");
            out.println(dao.total(key, query));
            break;
        case "/user/list":
            request.setAttribute("pageName", "/user/list.jsp");
            dis.forward(request, response);
            break;
    }
}
```

[src]-[main]-[webapp]-[user] list.jsp ①

```
<style>
    .bi-suit-heart-fill{
        color: red;
    }
    .rcnt, .fcnt, .ucnt {
        font-size:0.5rem;
    }
</style>
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">회원목록</h1>
        <div class="row">
            <form class="col-6" name="frm">
                <div class="input-group">
                    <select class="form-select" name="key">
                        <option value="uid">아이디</option>
                        <option value="uname" selected>회원이름</option>
                        <option value="address1">회원주소</option>
                        <option value="phone">회원전화</option>
                    </select>&nbsp;
                    <input class="form-control" placeholder="검색어" name="query">
                    <input type="submit" value="검색" class="btn btn-primary">
                </div>
            </form>
        </div>
        <hr>
        <div id="div_user" class="row"></div>
        <div id="pagination" class="pagination justify-content-center mt-3"></div>
    </div>
</div>
```



```

<script>
    let page=1;
    let query=$(frm.query).val();
    let key=$(frm.key).val();

    $(frm).on("submit", function(e){
        e.preventDefault();
        query=$(frm.query).val();
        key=$(frm.key).val();
        page=1;
        getTotal();
    });

    function getList(){
        $.ajax({
            type:"get",
            url:"/user/list.json",
            data:{key:key,query:query, page:page},
            dataType:"json",
            success:function(data){
                const temp=Handlebars.compile($("#temp_user").html());
                $("#div_user").html(temp(data));
            }
        });
    }

    getTotal();
    function getTotal(){
        $.ajax({
            type:"get",
            url:"/user/total",
            data:{query, key},
            success:function(data){
                $("#pagination").hide();
                if(data==0){
                    $("#div_user").html("<h3 class='text-center my-5'>검색회원이 존재하지 않습니다.</h3>");
                }else if(data > 6){
                    const totalPages=Math.ceil(data/6);
                    $("#pagination").twbsPagination("changeTotalPages", totalPages, page);
                    $("#pagination").show();
                }
            }
        });
    }

    $('#pagination').twbsPagination({
        ...
        onPageClick: function (event, clickPage) {
            page=clickPage;
            getList();
        }
    });
</script>

```

- 회원정보 페이지에 좋아요 수와 댓글 수 출력을 위한 DAO, JSP 페이지를 수정한다.

[src/main/java]-[model] UserDao.java

```
public UserVO read(String uid) { //회원정보 읽기
    UserVO vo=new UserVO();
    try {
        String sql="select *, ";
        sql+="(select count(*) from purchase where uid=u.uid) ucnt,";
        sql+="(select count(*) from reviews where uid=u.uid) rcnt,";
        sql+="(select count(*) from favorite where uid=u.uid) fcnt ";
        sql+="from users u where uid=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, uid);
        ResultSet rs=ps.executeQuery();
        if(rs.next()) {
            ...
            vo.setUcnt(rs.getInt("ucnt"));
            vo.setFcmt(rs.getInt("fcmt"));
            vo.setRcnt(rs.getInt("rcnt"));
        }
    } catch (Exception e) {
        System.out.println("회원정보읽기:" + e.toString());
    }
    return vo;
}
```

[src]-[main]-[webapp]-[user] read.jsp

```
<style>
...
.rcnt, .fcmt, .ucnt {
    font-size: 0.5rem;
}
.bi-suit-heart-fill {
    color: red;
}
.bi {
    font-size: 1.2rem;
}
</style>
<div class="row my-5 justify-content-center">
    <div class="col-md-10 col-lg-8">
        <h1 class="text-center mb-5">회원정보</h1>
        <div class="card p-3">
            ...
            <div class="text-end">
                <a href="/user/favorite?uid=${vo.uid}"><i class="bi bi-suit-heart-fill"></i></a>
                <span class="fcmt">${vo.fcmt}</span>
                <a href="/user/review?uid=${vo.uid}"><i class="bi bi-chat-left-text ms-3"></i></a>
                <span class="rcnt">${vo.rcnt}</span>
                <a href="/user/purchase?uid=${vo.uid}"><i class="bi bi-cart-check ms-3"></i></a>
                <span class="ucnt">${vo.ucnt}</span>
            </div>
        </div>
    </div>
</div>
```

- 특정 회원의 좋아요 상품목록 출력을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] GoodsDAO.java

```
public ArrayList<GoodsVO> list(String uid) { //특정회원의 좋아요 상품목록
    ArrayList<GoodsVO> array=new ArrayList<GoodsVO>();
    try {
        String sql="select g.*, f.uid from goods g, favorite f where g.gid=f.gid and f.uid=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, uid);
        ResultSet rs=ps.executeQuery();
        while(rs.next()) {
            GoodsVO vo=new GoodsVO();
            ...
            array.add(vo);
        }
    } catch (Exception e) {
        System.out.println("특정회원의 좋아요 상품목록:" + e.toString());
    }
    return array;
}
```

[src/main/java]-[model] UserController.java

```
@WebServlet(value= { ..., "/user/list.json", "/user/list", "/user/favorite", "/user/review"})
...
GoodsDAO gdao=new GoodsDAO();
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    switch(request.getServletPath()) {
        case "/user/favorite":
            request.setAttribute("array", gdao.list(request.getParameter("uid")));
            request.setAttribute("pageName", "/user/favorite.jsp");
            dis.forward(request, response);
            break;
    }
}
```

[src]-[main]-[webapp]-[user] favorite.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">상품목록</h1>
        <table class="table">
            <c:forEach items="${array}" var="vo" varStatus="status">
                <tr>
                    <td>${status.count}</td>
                    <td><a href="/goods/read?gid=${vo.gid}">${vo.gid}</a></td>
                    <td></td>
                    <td>${vo.title}</td>
                    <td><fmt:formatNumber value="${vo.price}" pattern="#,###원"/></td>
                </tr>
            </c:forEach>
        </table>
    </div>
</div>
```

- 특정회원이 작성한 리뷰를 출력하기 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] ReviewDAO.java

```
public ArrayList<ReviewVO> list(String gid) { //특정회원이 작성한 리뷰목록
    ArrayList<ReviewVO> array=new ArrayList<ReviewVO>();
    try {
        String sql="select * from view_reviews where gid=? order by rid desc";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, gid);
        ResultSet rs=ps.executeQuery();
        while(rs.next()) {
            ReviewVO vo=new ReviewVO();
            ...
            array.add(vo);
        }
    } catch(Exception e) {
        System.out.println("리뷰목록:" + e.toString());
    }
    return array;
}
```

[src/main/java]-[model] UserController.java

```
@WebServlet(value= { ..., "/user/list.json", "/user/list", "/user/favorite", "/user/review"})
...
ReviewDAO rdao=new ReviewDAO();
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    switch(request.getServletPath()) {
        ...
        case "/user/review":
            request.setAttribute("array", rdao.userList(request.getParameter("uid")));
            request.setAttribute("pageName", "/user/review.jsp");
            dis.forward(request, response);
            break;
    }
}
```

[src]-[main]-[webapp]-[user] review.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">리뷰목록</h1>
        <table class="table">
            <c:forEach items="${array}" var="vo" varStatus="status">
                <tr>
                    <td>${status.count}</td>
                    <td width="10%"><a href="/goods/read?gid=${vo.gid}">${vo.gid}</a></td>
                    <td width="70%"><div class="ellipsis">${vo.content}</div></td>
                    <td width="20%">${vo.revDate}</td>
                </tr>
            </c:forEach>
        </table>
    </div>
</div>
```

- 특정회원이 구매한 구매목록을 출력하기 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] PurchaseDAO.java

```
public ArrayList<PurchaseVO> list(String uid) { //특정 회원이 구매한 구매목록
    ArrayList<PurchaseVO> array=new ArrayList<PurchaseVO>();
    try {
        String sql="select * from view_purchase where uid=?";
        PreparedStatement ps=Database.CON.prepareStatement(sql);
        ps.setString(1, uid);
        ResultSet rs=ps.executeQuery();
        while(rs.next()) {
            PurchaseVO vo=new PurchaseVO();
            ...
            array.add(vo);
        }
    } catch(Exception e) {
        System.out.println("특정 회원의 구매목록:" + e.toString());
    }
    return array;
}
```

[src/main/java]-[model] UserController.java

```
@WebServlet(value= { ..., "/user/list.json", "/user/list", "/user/favorite", "/user/review", "/user/purchase"})
PurchaseDAO pdao=new PurchaseDAO();
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    switch(request.getServletPath()) {
        ...
        case "/user/purchase":
            request.setAttribute("array", pdao.list(request.getParameter("uid")));
            request.setAttribute("pageName", "/user/purchase.jsp");
            dis.forward(request, response);
            break;
    }
}
```

[src]-[main]-[webapp]-[user] purchase.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<div class="row my-5">
    <div class="col">
        <h1 class="text-center mb-5">구매목록</h1>
        <table class="table">
            <c:forEach items="${array}" var="vo" varStatus="status">
                <tr>
                    <td>${status.count}</td>
                    <td><a href="/purchase/read?pid=${vo.pid}">${vo.pid}</a></td>
                    <td>${vo.recName}</td>
                    <td>${vo.recAddress1} ${vo.recAddress2}</td>
                    <td>${vo.recPhone}</td>
                    <td>${vo.purDate}</td>
                </tr>
            </c:forEach>
        </table>
    </div>
</div>
```

- CK Editor를 이용한 content 추가

- goods 테이블에 content 필드를 추가한다.

goods 테이블에 content 필드 추가

```
alter table goods add content text;
```

[src/main/java]-[model] GoodsVO.java

```
public class GoodsVO {  
    ...  
    private String content;  
    //setter(), getter(), toString() 메서드를 generate한다.  
}
```

- CK Editor에서 파일을 업로드하기 위한 Controller를 작성한다.

[src/main/java]-[model] GoodsController.java

```
@WebServlet(value={"/goods/search","/goods/search.json","/goods/append","/goods/list.json", ... , "/goods/ckupload"})  
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
    switch(request.getServletPath()) {  
        ...  
        case "/goods/ckupload":  
            try {  
                gid=request.getParameter("gid");  
                String pathCK ="/upload/goods/ck/" + gid + "/";  
                File dirPathCK = new File("c:" + pathCK);  
                if(!dirPathCK.exists()) dirPathCK.mkdir();  
                multi = new MultipartRequest(request,"c:" + pathCK, 1024 * 1024 * 10,"UTF-8",new DefaultFileRenamePolicy());  
                String fileName=multi.getFilesystemName("upload");  
                String fileUrl=pathCK + fileName;  
  
                JSONObject json = new JSONObject();  
                json.put("uploaded", 1);  
                json.put("fileName", fileName);  
                json.put("url", fileUrl);  
                PrintWriter out=response.getWriter();  
                out.println(json);  
            }catch(Exception e) {  
                System.out.println("ck 업로드:" + e.toString());  
            }  
            break;  
        }  
    }  
}
```

[src/main/java]-[model] GoodsDAO.java

```
public void update(GoodsVO vo) { //상품정보 수정  
    try {  
        String sql="update goods set title=?,price=?,maker=?,image=?, content=? where gid=?";  
        PreparedStatement ps=Database.CON.prepareStatement(sql);  
        ...  
    }catch(Exception e) {  
        System.out.println("상품수정:" + e.toString());  
    }  
}
```

- 상품정보 수정페이지에 CK Editor를 추가한다.

```
[src]-[main]-[webapp]-[goods] update.jsp
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<script src="https://cdn.ckeditor.com/4.16.2/standard/ckeditor.js"></script>
<div class="row my-5 justify-content-center">
    <div class="col-md-10 col-lg-8">
        <h1 class="text-center mb-5">상품정보수정</h1>
        <form name="frm" class="card p-3" method="post" enctype="multipart/form-data">
            ...
            <div class="tab-content pt-5" id="myTabContent">
                <!-- 상세설명 -->
                <div class="tab-pane fade" id="detail-tab-pane" role="tabpanel" aria-labelledby="detail-tab" tabindex="1">
                    <textarea id="editor" name="content" rows=15 class="form-control">${vo.content}</textarea>
                </div>
            </div>
        </form>
        ...
        <div class="text-center my-3">
            <input type="submit" value="상품수정" class="btn btn-primary">
            <input type="reset" value="수정취소" class="btn btn-secondary">
        </div>
    </div>
</div>
<script>
    ...
    const ckeditor_config = {
        resize_enable : false, //editor 사이즈를 변경하지 못한다.
        enterMode : CKEDITOR.ENTER_BR,
        shiftEnterMode : CKEDITOR.ENTER_P,
        filebrowserUploadUrl : "/goods/ckupload?gid=" + gid,
        height: 300
    };
    CKEDITOR.replace('editor', ckeditor_config)
</script>
```

- 상품정보 페이지에 content 내용 출력을 위해 JSP 페이지를 수정한다.

```
[src]-[main]-[webapp]-[goods] read.jsp
```

```
<div class="row my-5 justify-content-center">
    <div class="col-md-10 col-lg-8">
        <h1 class="text-center mb-5">상품정보</h1>
        <div class="card p-3">
            ...
            <div class="tab-content pt-5" id="myTabContent">
                <!-- 상세설명 -->
                <div class="tab-pane fade show active" id="detail-tab-pane" aria-labelledby="detail-tab" tabindex="1">
                    <div>${vo.content}</div>
                </div>
            </div>
        </div>
    </div>
</div>
```

- 상품 첨부파일

- 첨부파일 등록을 위한 테이블과 VO를 작성한다.

첨부파일 테이블 생성

```
create table attach(  
    aid int auto_increment primary key,  
    gid char(8) not null,  
    image varchar(200) not null,  
    regDate datetime default now(),  
    foreign key(gid) references goods(gid)  
);
```

[src/main/java]-[model] AttachVO.java

```
public class AttachVO {  
    private int aid;  
    private String gid;  
    private String image;  
    private String regDate;  
    //setter(), getter(), toString() 메서드를 generate한다.  
}
```

- 첨부파일 등록과 목록 출력을 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] AttachDAO.java

```
public class AttachDAO {  
    public void insert(AttachVO vo) {  
        try {  
            String sql="insert into attach(gid, image) values(?, ?)";  
            PreparedStatement ps=Database.CON.prepareStatement(sql);  
            ps.setString(1, vo.getGid());  
            ps.setString(2, vo.getImage());  
            ps.execute();  
        }catch(Exception e) {  
            System.out.println("첨부파일등록:" + e.toString());  
        }  
    }  
  
    public ArrayList<AttachVO> list(String gid){  
        ArrayList<AttachVO> array=new ArrayList<AttachVO>();  
        try {  
            String sql="select * from attach where gid=?";  
            PreparedStatement ps=Database.CON.prepareStatement(sql);  
            ps.setString(1, gid);  
            ResultSet rs=ps.executeQuery();  
            while(rs.next()) {  
                AttachVO vo=new AttachVO();  
                vo.setAid(rs.getInt("aid"));  
                vo.setGid(rs.getString("gid"));  
                vo.setImage(rs.getString("image"));  
                array.add(vo);  
            }  
        }catch(Exception e) {  
            System.out.println("첨부파일목록:" + e.toString());  
        }  
        return array;  
    }  
}
```



```
[src/main/java]-[controller] AttachController.java
```

```
@MultipartConfig(
    fileSizeThreshold = 1024 * 1024, //1MB
    maxFileSize = 1024 * 1024 * 10,
    maxRequestSize = 1024*1024 * 15)

@WebServlet(value= {"/attach/insert"})
public class AttachController extends HttpServlet {
    AttachDAO dao=new AttachDAO();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        switch(request.getServletPath()) {
            case "/attach/list.json":
                String gid=request.getParameter("gid");
                Gson gson=new Gson();
                out.println(gson.toJson(dao.list(gid)));
                break;
        }
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            case "/attach/insert":
                String gid=request.getParameter("gid");
                String path="/upload/goods/attach/" + gid + "/";
                File dirPath = new File("c:/" + path);
                if(!dirPath.exists()) dirPath.mkdir();
                //파일들 업로드 및 데이터 저장
                Collection<Part> parts = request.getParts();
                for(Part part:parts) {
                    if(!part.getName().equals("files")) continue;
                    String name=part.getSubmittedFileName();
                    String exe=name.substring(name.lastIndexOf("."));
                    UUID uuid = UUID.randomUUID();
                    String fileName = uuid.toString().substring(0, 8) + exe;
                    try {
                        //파일업로드
                        InputStream is=part.getInputStream();
                        FileOutputStream fos=new FileOutputStream("c:/" + path + fileName);
                        int data=0;
                        while((data=is.read()) != -1) {
                            fos.write(data);
                        }
                        is.close();
                        fos.close();
                        //데이터저장
                        AttachVO vo=new AttachVO();
                        vo.setGid(gid);
                        vo.setImage(path + fileName);
                        dao.insert(vo);
                    } catch (Exception e) {
                        System.out.println("파일업로드:" + e.toString());
                    }
                }
                break;
        } //switch
    } //doPost
} //servlet
```

[src]-[main]-[webapp]-[goods] read.jsp

```
<!-- 첨부이미지 -->
<div class="tab-pane fade" id="attach-tab-pane" role="tabpanel" aria-labelledby="attach-tab" tabindex="2">
  <div id="div_attached" class="p-3 my-3"></div>
</div>
<!-- 첨부한 파일목록 템플릿 -->
<script id="temp_attached" type="x-handlebars-template">
  <div class="row">
    {{#each .}}
      <div class="col-6 col-md-3 col-lg-2 mb-2"><div class="card p-2"></div></div>
    {{/each}}
  </div>
</script>
<script>
  getAttach();
  function getAttach(){
    $.ajax({
      type:"get",
      url:"/attach/list.json",
      data:{'gid': gid},
      dataType:"json",
      success:function(data){
        const temp=Handlebars.compile($("#temp_attached").html());
        $("#div_attached").html(temp(data));
        if(data==0) $("#div_attached").html("첨부한 이미지");
      }
    });
  }
</script>
```

[src]-[main]-[webapp]-[goods] attach.jsp ①

```
<div class="input-group">
  <input id="files" type="file" class="form-control" multiple="multiple"/>
  <a id="btn-attach" class="btn btn-primary">파일첨부</a>
</div>
<div id="div_attach" class="card p-3 my-3"></div>
<div id="div_attached" class="p-3 my-3"></div>
<!-- 첨부할 파일목록 템플릿 -->
<script id="temp_attach" type="x-handlebars-template">
  <div class="row">
    {{#each .}}
      <div class="col-6 col-md-3 mb-2"></div>
    {{/each}}
  </div>
</script>
<!-- 첨부한 파일목록 템플릿 -->
<script id="temp_attached" type="x-handlebars-template">
  <div class="row">
    {{#each .}}
      <div class="col-6 col-md-3 col-lg-2 mb-2">
        <div class="card p-2">
          
          <span class="badge text-bg-danger position-absolute top-0 start-0" aid="{{aid}}">x</span>
        </div>
      </div>
    {{/each}}
  </div>
</script>
```

[src]-[main]-[webapp]-[goods] attach.jsp ②

```
<script>
    $("#files").on("change", function(e) { //이미지들 미리보기
        let files=[];
        for(let i=0; i<e.target.files.length; i++) {
            const file=URL.createObjectURL(e.target.files[i]);
            files.push(file);
        }
        const temp=Handlebars.compile($("#temp_attach").html());
        $("#div_attach").html(temp(files));
    });

    $("#btn-attach").on("click", function(e) { //파일 첨부파일 버튼을 클릭한 경우
        const files = $("#files")[0].files;
        if(files.length==0) {
            alert("첨부할 파일들을 선택하세요!");
        }else{
            if(confirm("선택한 파일을 업로드 하시겠습니까?")) {
                const formData = new FormData();
                formData.append("gid", gid);
                for(var i=0; i<files.length; i++) { formData.append("files", files[i]); }
                $.ajax({
                    type:"post",
                    url:"/attach/insert",
                    data:formData,
                    processData:false,
                    contentType:false,
                    success:function() {
                        alert("첨부파일 업로드가 성공되었습니다.");
                        $("#files").val("");
                        getAttach();
                    }
                });
            }
        }
    });

    getAttach();
    function getAttach(){
        $.ajax({
            type:"get",
            url:"/attach/list.json",
            data:{"gid": gid},
            dataType:"json",
            success:function(data) {
                const temp=Handlebars.compile($("#temp_attached").html());
                $("#div_attached").html(temp(data));
                $("#div_attach").html("첨부할 이미지");
            }
        });
    }
}</script>
```

[src]-[main]-[webapp]-[goods] update.jsp

```
<!-- 첨부이미지 -->
<div class="tab-pane fade" id="attach-tab-pane" role="tabpanel" aria-labelledby="attach-tab" tabindex="2">
    <jsp:include page="attach.jsp"/>
</div>
```

- 첨부파일 삭제를 위한 DAO, Controller, JSP 페이지를 작성한다.

[src/main/java]-[model] AttachDAO.java

```
public class AttachDAO {
    ...
    public void delete(int aid) {
        try {
            String sql="delete from attach where aid=?";
            PreparedStatement ps=Database.CON.prepareStatement(sql);
            ps.setInt(1, aid);
            ps.execute();
        }catch(Exception e) {
            System.out.println("첨부파일삭제:" + e.toString());
        }
    }
}
```

[src/main/java]-[controller] AttachController.java

```
@WebServlet(value= {"/attach/insert", "/attach/list.json", "/attach/delete"})
public class AttachController extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        switch(request.getServletPath()) {
            ...
            case "/attach/delete":
                AttachVO vo=new AttachVO();
                vo.setAid(Integer.parseInt(request.getParameter("aid")));
                vo.setGid(request.getParameter("gid"));
                vo.setImage(request.getParameter("image"));
                //이미지파일삭제
                File file=new File("c:" + vo.getImage());
                if(file.exists()) file.delete();
                //데이터삭제
                dao.delete(vo.getAid());
                break;
            }
        }
    }
}
```

[src]-[main]-[webapp]-[goods] attach.jsp ③

```
<script>
    ...
    //첨부이미지 삭제 버튼을 클릭한 경우
    $("#div_attached").on("click", ".text-bg-danger", function(){
        const aid=$(this).attr("aid");
        const image=$(this).parent().find("img").attr("src");
        if(confirm(aid+"번 첨부이미지를 삭제하실래요?")) {
            $.ajax({
                type:"post",
                url:"/attach/delete",
                data:{aid, gid, image},
                success:function(){
                    alert("이미지가 삭제되었습니다.");
                    getAttach();
                }
            });
        }
    });
</script>
```

- 구글 그래프 차트

- 월별 매출 현황을 출력하는 그래프 출력 프로그램을 작성한다.

procedure1 프로시저 작성

```
CREATE DEFINER='root'@'localhost' PROCEDURE `procedure1`(in yy char(4))
BEGIN
    declare i int default 1;
    declare yymm char(7);
    drop table if exists tbl_temp;

    create table tbl_temp(
        id int auto_increment primary key,
        puryymm char(7),
        pursum int default 0
    );

    while(i<=12) do
        if(i>=1 and i<=9) then
            set yymm = concat(yy, '-0', i);
        else
            set yymm = concat(yy, '-', i);
        end if;
        insert into tbl_temp(puryymm, pursum)
        select right(yymm,2), sum(sum) from purchase where date_format(purDate,'%Y-%m') = yymm;
        set i=i+1;
    end while;

    select * from tbl_temp;
    drop table if exists tbl_temp;
END
```

[src/main/java]-[model] GraphDAO.java

```
public class GraphDAO {
    //년도별 월별 매출현황
    public ArrayList<JSONObject> procedure1(String yymm){
        ArrayList<JSONObject> array=new ArrayList<JSONObject>();
        try {
            String sql="call procedure1(?)";
            CallableStatement cs=Database.CON.prepareCall(sql);
            cs.setString(1, yymm);
            cs.execute();
            ResultSet rs=cs.getResultSet();
            while(rs.next()) {
                JSONObject obj=new JSONObject();
                obj.put("id", rs.getInt("id"));
                obj.put("yymm", rs.getString("puryymm"));
                obj.put("sum", rs.getInt("pursum"));
                array.add(obj);
            }
        }catch(Exception e) {
            System.out.println("년도별 월별 매출현황:" + e.toString());
        }
        return array;
    }
}
```

[src/main/java]-[controller] GraphController.java

```
@WebServlet(value={"/graph/procedure1.json", "/graph/chart1"})
public class GraphController extends HttpServlet {
    GraphDAO dao=new GraphDAO();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
        switch(request.getServletPath()) {
            case "/graph/procedure1.json":
                out.println(dao.procedure1("2023"));
                break;
            case "/graph/chart1":
                request.setAttribute("pageName", "/graph/chart1.jsp");
                dis.forward(request, response);
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[graph] chart1.jsp

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<div class="row">
    <div class="col p-5">
        <div id="chart" style="width:100%; height: 500px" class="card"></div>
    </div>
</div>
<script>
    function chart(rows) {
        google.charts.load('current', {'packages':['corechart']});
        google.charts.setOnLoadCallback(drawChart);
        function drawChart() {
            var data = google.visualization.arrayToDataTable(rows);
            var options = {
                title: '2023년 월별 매출연망', //curveType: 'function', //legend: { position: 'bottom' },
            };
            var chart = new google.visualization.LineChart(document.getElementById('chart'));
            chart.draw(data, options);
        }
    }
    getProcedure1();
    function getProcedure1(){
        $.ajax({
            type:"get",
            url:"/graph/procedure1.json",
            dataType:"json",
            success:function(data){
                let rows=[];
                rows.push(["월", "매출액"])
                data.forEach(item=>{
                    const row=[item.yymm, item.sum];
                    rows.push(row);
                });
                chart(rows);
            }
        });
    }
</script>
```

- 사용자별 작성한 댓글, 좋아요, 구매 수 출력하는 그래프를 작성한다.

procedure2 프로시저 작성

```
CREATE DEFINER='root'@'localhost' PROCEDURE `procedure2`()
BEGIN
    drop view if exists view_temp;

    create view view_temp as
        select uid, uname,
            (select count(*) from reviews where uid=u.uid) rcnt,
            (select count(*) from favorite where uid=u.uid) fcnt,
            (select count(*) from purchase where uid=u.uid) pcnt
        from users u;

    select * from view_temp where rcnt>0 or fcnt>0 or pcnt>0;
    drop view if exists view_temp;
END
```

[src/main/java]-[model] GraphDAO.java

```
public ArrayList<JSONObject> procedure2() { //사용자별 좋아요, 댓글, 구매횟수
    ArrayList<JSONObject> array=new ArrayList<JSONObject>();
    try {
        String sql="call procedure2()";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.execute();
        ResultSet rs=cs.getResultSet();
        while(rs.next()) {
            JSONObject obj=new JSONObject();
            obj.put("uid", rs.getString("uid"));
            obj.put("uname", rs.getString("uname"));
            obj.put("fcnt", rs.getInt("fcnt"));
            obj.put("rcnt", rs.getInt("rcnt"));
            obj.put("pcnt", rs.getInt("pcnt"));
            array.add(obj);
        }
    } catch(Exception e) {
        System.out.println("사용자별 좋아요, 댓글, 구매횟수:" + e.toString());
    }
    return array;
}
```

[src/main/java]-[controller] GraphController.java

```
@WebServlet(value={"/graph/procedure1.json", "/graph/chart1", "/graph/procedure2.json", "/graph/chart2"})
...
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    switch(request.getServletPath()) {
        ...
        case "/graph/procedure2.json":
            out.println(dao.procedure2());
            break;
        case "/graph/chart2":
            request.setAttribute("pageName", "/graph/chart2.jsp");
            dis.forward(request, response);
            break;
    }
}
```

```
[src]-[main]-[webapp]-[graph] chart2.jsp
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<div class="row">
    <div class="col p-5">
        <div id="chart" style="width:100%; height: 500px"></div>
    </div>
</div>
<script>
    function chart(rows){
        google.charts.load('current', {packages: ['corechart', 'bar']});
        google.charts.setOnLoadCallback(drawBarColors);
        function drawBarColors() {
            var data = google.visualization.arrayToDataTable(rows);
            var options = {
                title: '사용자별 좋아요,댓글, 상품구매수',
                chartArea: {width: '70%'},
                colors: ['red', 'yellow', 'green'],
                hAxis: { title: '횟수', minValue: 0 },
                vAxis: { title: '이름' }
            };
            var chart = new google.visualization.BarChart(document.getElementById('chart'));
            chart.draw(data, options);
        }
    }

    getProcedure2();
    function getProcedure2(){
        $.ajax({
            type:"get",
            url:"/graph/procedure2.json",
            dataType:"json",
            success:function(data){
                let rows=[];
                rows.push(["회원명", "좋아요수", "댓글수", "구매수"])
                data.forEach(item=>{
                    const row=[item.uname, item.fcmt, item.rcmt, item.pcmt];
                    rows.push(row);
                });
                chart(rows);
            }
        });
    }
</script>
```

- 제조사별 상품수를 출력하는 그래프를 작성한다.

```
procedure3 프로시저 작성
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `procedure3`()
BEGIN
    select count(*), maker
    from goods
    group by maker;
END
```


[src/main/java]-[model] GraphDAO.java

```
public class GraphDAO {
    ...
    //제조사별 상품수
    public ArrayList<JSONObject> procedure3() {
        ArrayList<JSONObject> array=new ArrayList<JSONObject>();
        try {
            String sql="call procedure3()";
            CallableStatement cs=Database.CON.prepareCall(sql);
            cs.execute();
            ResultSet rs=cs.getResultSet();
            while(rs.next()) {
                JSONObject obj=new JSONObject();
                obj.put("maker", rs.getString("maker"));
                obj.put("count", rs.getInt("count(*)"));
                array.add(obj);
            }
        } catch(Exception e) {
            System.out.println("제조사별상품수:" + e.toString());
        }
        return array;
    }
}
```

[src/main/java]-[controller] GraphController.java

```
@WebServlet(value={"/graph/procedure1.json", "/graph/chart1", ..., "/graph/procedure3.json", "/graph/chart3"})
public class GraphController extends HttpServlet {
    GraphDAO dao=new GraphDAO();
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out=response.getWriter();
        RequestDispatcher dis=request.getRequestDispatcher("/home.jsp");
        switch(request.getServletPath()) {
            case "/graph/procedure1.json":
                out.println(dao.procedure1("2023"));
                break;
            case "/graph/chart1":
                request.setAttribute("pageName", "/graph/chart1.jsp");
                dis.forward(request, response);
                break;
            case "/graph/procedure2.json":
                out.println(dao.procedure2());
                break;
            case "/graph/chart2":
                request.setAttribute("pageName", "/graph/chart2.jsp");
                dis.forward(request, response);
                break;
            case "/graph/procedure3.json":
                out.println(dao.procedure3());
                break;
            case "/graph/chart3":
                request.setAttribute("pageName", "/graph/chart3.jsp");
                dis.forward(request, response);
                break;
        }
    }
}
```

[src]-[main]-[webapp]-[graph] chart3.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
<div class="row">
  <div class="col p-5">
    <div id="chart" style="width:100%; height: 500px" class="card"></div>
  </div>
</div>
<script>
  function chart(rows){
    google.charts.load('current', {'packages':['corechart']});
    google.charts.setOnLoadCallback(drawChart);

    function drawChart() {
      var data = google.visualization.arrayToDataTable(rows);
      var options = { title: '제조사별상품수' };
      var chart = new google.visualization.PieChart(document.getElementById('chart'));
      chart.draw(data, options);
    }
  }

  getProcedure2();
  function getProcedure2(){
    $.ajax({
      type:"get",
      url:"/graph/procedure3.json",
      dataType:"json",
      success:function(data){
        let rows=[];
        rows.push(["제조사", "상품수량"])
        data.forEach(item=>{
          const row=[item maker, item count];
          rows.push(row);
        });
        chart(rows);
      }
    });
  }
</script>
```

- header 페이지에 그래프 메뉴를 추가한다.

[src]-[main]-[webapp] header.jsp

```
<c:if test="${user.role==1}">
  <li class="nav-item"><a class="nav-link active" aria-current="page" href="/goods/search">상품검색</a></li>
  <li class="nav-item"><a class="nav-link active" aria-current="page" href="/goods/list">상품목록</a></li>
  <li class="nav-item"><a class="nav-link active" aria-current="page" href="/user/list">회원목록</a></li>
  <li class="nav-item"><a class="nav-link active" aria-current="page" href="/purchase/list">구매목록</a></li>
  <li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle active" href="#" role="button" data-bs-toggle="dropdown">그래프</a>
    <ul class="dropdown-menu">
      <li><a class="dropdown-item" href="/graph/chart1">월별 매출현황</a></li>
      <li><a class="dropdown-item" href="/graph/chart2">사용자별 댓글, 좋아요, 구매수</a></li>
      <li><a class="dropdown-item" href="/graph/chart3">제조사별상품수</a></li>
    </ul>
  </li>
</c:if>
```

- 기타

- Mysql Stored Procedure

- 년도를 입력 받아 월별 게시물 수와 댓글 수를 출력

```
call procedure1('2022');
```

```
CREATE DEFINER='root'@'localhost' PROCEDURE `procedure1`(in yy char(4))
BEGIN
    DECLARE i int DEFAULT 1;
    DECLARE v_yymm char(7);

    DROP TABLE IF EXISTS tbl_temp;
    create table tbl_temp(
        id int auto_increment primary key,
        yymm char(7),
        board_cnt int default 0,
        reply_cnt int default 0
    );

    WHILE (i<=12) DO
        IF( i >= 1 and i <= 9) THEN
            SET v_yymm = CONCAT(yy, '-0', i);
        ELSE
            SET v_yymm = CONCAT(yy, '-', i);
        END IF;

        insert into tbl_temp(yymm, board_cnt)
        select v_yymm, count(*) from tbl_board where date_format(regdate,'%Y-%m') = v_yymm;

        update tbl_temp
        set reply_cnt = (select count(*) from tbl_reply where date_format(replydate,'%Y-%m') = v_yymm)
        where id = last_insert_id();

        SET i= i+1;
    END WHILE;

    select * from tbl_temp;
    DROP TABLE IF EXISTS tbl_temp;
END
```

```
procedure1프로시저 실행
```

```
public ArrayList<JSONObject> procedure1(String yymm){
    ArrayList<JSONObject> array=new ArrayList<JSONObject>();
    try {
        String sql="call procedure1(?)";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.setString(1, yymm);
        cs.execute();
        ResultSet rs=cs.getResultSet();
        while(rs.next()) { ... }
    }catch(Exception e) {
        System.out.println(e.toString());
    }
    return array;
}
```

- 사용자별 작성한 게시물 수와 댓글 수 출력

```
call procedure2
```

```
CREATE DEFINER='root'@'localhost' PROCEDURE `procedure2`()
BEGIN
    DECLARE isEnd boolean DEFAULT false;
    DECLARE v_uid varchar(20) DEFAULT '';
    DECLARE v_uname varchar(20) DEFAULT '';
    DECLARE cursor_user CURSOR FOR select distinct(uid), uname from tbl_user;
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET isEnd = true;

    DROP TABLE IF EXISTS tbl_temp;
    create table tbl_temp(
        id int auto_increment primary key,
        uid varchar(20),
        uname varchar(20),
        board_cnt int default 0,
        reply_cnt int default 0
    );

    OPEN cursor_user;
    loop_user: LOOP
        FETCH cursor_user INTO v_uid, v_uname;
        IF isEnd THEN
            LEAVE loop_user;
        END IF;
        insert into tbl_temp(uid, uname, board_cnt) select v_uid, v_uname, count(*) from tbl_board where writer = v_uid;
        update tbl_temp set reply_cnt = (select count(*) from tbl_reply where replyer = v_uid) where id = last_insert_id();
    END LOOP loop_user;
    CLOSE cursor_user;

    select * from tbl_temp where board_cnt>0 or reply_cnt>0;
    DROP TABLE IF EXISTS tbl_temp;
END
```

- 검색기능과 페이지결과 목록 출력

```
call list('professors', 'pcode', '', 'pcode', '', 1, 5);
```

```
CREATE DEFINER='haksa'@'localhost' PROCEDURE 'LIST'(
    in i_table varchar(1000),
    in i_key varchar(50),
    in i_word varchar(50),
    in i_order varchar(50),
    in i_desc varchar(5),
    in i_page int,
    in i_perPage int
)
BEGIN
    SET @v_word=concat('%',i_word,'%');
    SET @v_start=(i_page-1) * i_perPage;
    SET @v_perPage=i_perPage;
    SET @v_sql=concat('select * from ',i_table,' where ',i_key,' like ? order by ',i_order,' ',i_desc,' limit ?, ?');
    PREPARE ps from @v_sql;
    EXECUTE ps using @v_word, @v_start, @v_perPage;
    SET @v_sql=concat('select count(*) total from ',i_table,' where ',i_key,' like ?');
    PREPARE ps from @v_sql;
    EXECUTE ps using @v_word;
END
```

list 프로시저 실행

```
public JSONObject list(SqlVO sqlVO) {
    JSONObject jObject=new JSONObject();
    try {
        String sql="call list(?, ?, ?, ?, ?, ?)";
        CallableStatement cs=Database.CON.prepareCall(sql);
        ...
        cs.execute();

        ResultSet rs=cs.getResultSet();
        JSONArray jArray=new JSONArray();
        while(rs.next()) { ... }
        jObject.put("array", jArray);

        cs.getMoreResults();
        rs=cs.getResultSet();
        int total=0;
        if(rs.next()) total=rs.getInt("total");
        jObject.put("total", total);
    } catch (Exception e) {
        System.out.println(e.toString());
    }
    return jObject;
}
```

- foreign key가 존재하지 않은 경우에 primary key을 이용한 삭제

```
call del_students('92414029', @count); select @count;
```

```
CREATE PROCEDURE `DEL_STUDENTS`(
    in i_score char(8),
    out o_count int
)
BEGIN
    select count(*) into o_count from enrollments where score = i_score;
    IF(o_count = 0)THEN
        delete from students where score=i_score;
    END IF;
END
```

del_students 프로시저 실행

```
public int delete(String score) {
    int count = -1;
    try {
        String sql="call del_students(?, ?)";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.setString(1, score);
        cs.registerOutParameter(2, java.sql.Types.INTEGER);
        cs.execute();
        count = cs.getInt(2);
    } catch (Exception e) {
        System.out.println(e.toString());
    }
    return count;
}
```

- Oracle Stored Procedure

- 오라클 데이터베이스에 연결하기 위한 Connection 클래스

Database.java

```
import java.sql.*;

public class Database {
    public static Connection CON;
    static {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
            con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","1234");
            System.out.println("접속성공");
        }catch(Exception e) {
            System.out.println("접속실패:" + e.toString());
        }
    }
}
```

- 검색기능과 페이징 결과 목록출력

```
SET SERVEROUTPUT ON; /* 결과를 화면에 출력하기 위한 설정 */
variable cur refcursor;
variable cnt number;
exec list('table name', 'search key', 'query', 1, 5, 'order key', 'desc', :cur , :cnt);
print cur;
print cnt;
```

list 프로시저

```
create or replace PROCEDURE LIST(
    i_table in varchar,
    i_key in varchar,
    i_word in varchar,
    i_page in int,
    i_perPage in int,
    i_order in varchar,
    i_desc in varchar,
    o_cur out sys_refcursor,
    o_count out int)
AS
    v_tbl1 varchar(1000);
    v_tbl2 varchar(1000);
    v_sql varchar(1000);
    v_word varchar(100) := ''%' || i_word || '%'';
    v_start int :=(i_page-1)*i_perPage +1;
    v_end int:=i_page * i_perPage;
BEGIN
    v_tbl1:=(select * from '||i_table||' where '||i_key||' like '||v_word||' order by '||i_order||' '||i_desc||') tbl';
    v_tbl2:=(select rownum rn, tbl.* from ' || v_tbl1 || ' ');
    v_sql:='select * from ' || v_tbl2 || ' where rn between ' || v_start || ' and ' || v_end;
    DBMS_OUTPUT.PUT_LINE(v_sql);
    OPEN o_cur FOR v_sql;

    v_sql := 'select count(*) from '|| v_tbl1;
    EXECUTE IMMEDIATE v_sql INTO o_count;
END LIST;
```

list 프로시저 java 실행

```
public JSONObject list(SqlVO sqlVO) {
    JSONObject object=new JSONObject();
    try {
        String sql="{ call list(?, ?, ?, ?, ?, ?, ?, ?) }";
        CallableStatement cs=Database.CON.prepareCall(sql);
        ...
        cs.registerOutParameter(8, oracle.jdbc.OracleTypes.CURSOR);
        cs.registerOutParameter(9, oracle.jdbc.OracleTypes.INTEGER);
        cs.execute();
        ResultSet rs=(ResultSet)cs.getObject(8);
        JSONArray array=new JSONArray();
        while(rs.next()) { ... }
        object.put("array", array);
        int count=cs.getInt(9);
        object.put("count", count);
    }catch(Exception e) {
        System.out.println(e.toString());
    }
    return object;
}
```

- foreign key가 존재하지 않은 경우에 primary key을 이용한 삭제

```
variable count number;
exec del_product('코드', :count);
print count;
```

del_product 프로시저

```
create or replace PROCEDURE DEL_PRODUCT(
    i_prod_id in product.prod_id%TYPE,
    o_count out int
)
AS
BEGIN
    select count(*) into o_count from orders where prod_id=i_prod_id;
    IF(o_count=0)THEN
        delete from product where prod_id=i_prod_id;
    END IF;
END DEL_PRODUCT;
```

del_product 프로시저 java 실행

```
public int delete(String id) {
    int count=-1;
    try {
        String sql="{ call del_product(?, ?) }";
        CallableStatement cs=Database.CON.prepareCall(sql);
        cs.setString(1, id);
        cs.registerOutParameter(2, oracle.jdbc.OracleTypes.INTEGER);
        cs.execute();
        count=cs.getInt(2);
    }catch(Exception e) {
        System.out.println(e.toString());
    }
    return count;
}
```