

WEB DEVELOPMENT **PORTFOLIO**

아무도 가보지 않은 길을 함께 도전할 수 있다는 가능성

신입 웹개발자

설 예 린

WEB DEVELOPMENT *PORTFOLIO* 교육이수

2019

빅데이터 분석 및 스프링 프레임워크 기반 자바(JAVA) 개발자 양성과정

ITWILL (2019.05.21 – 2019.11.21)

JAVA, Oracle, JSP(MVC Model-1/2), MYBATIS, Spring 등의 기업 실무기반 기술 습득

100%

2017

스마트웹디자인&콘텐츠개발

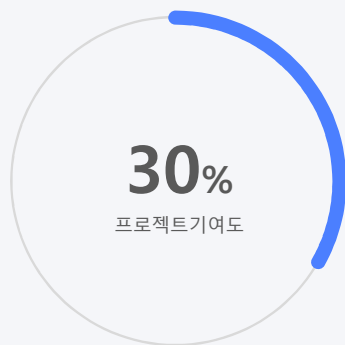
EZEN ACADEMY (2017.03.14 – 2017.09.11)

2DGraphic(Photoshop, Illustrator), HTML, CSS, Javascript, jQuery, 반응형, 웹디자인 등의 기업 실무기반 기술 습득

100%

WEB DEVELOPMENT *PORTFOLIO*

1차 프로젝트



1차 프로젝트

(URL : <https://github.com/Na-roo/PC-KIOSK>)

PC방 키오스크 프로그램

주요기술 : JAVA, Swing

프로젝트인원 : 5명 (팀원)

개발기간 : 2019.06.17 - 2019.06.25



개발목표

Step. 1

JAVA, AWT를 이용하여 관리자
계정 GUI를 개발 및 구현



구현설명

Step. 2

관리자 계정으로 로그인하여 회
원관리, 재고관리, 이벤트처리
(CRUD 구현)

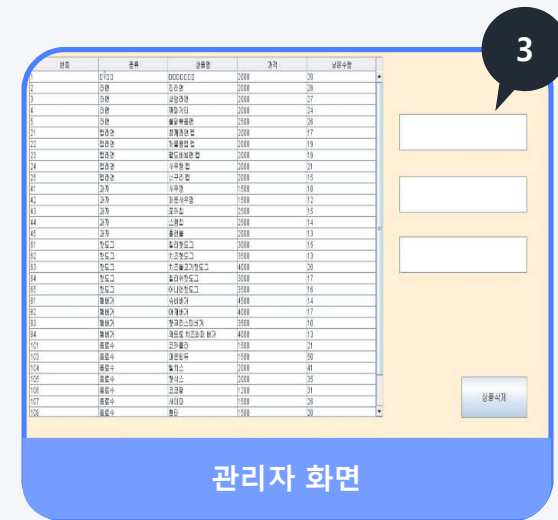
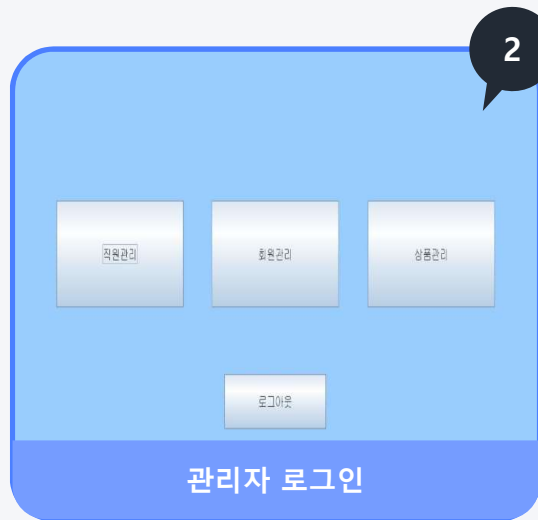
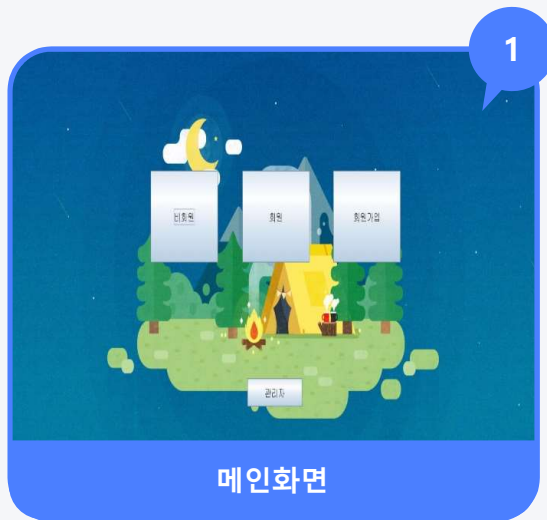


담당업무

Step. 3

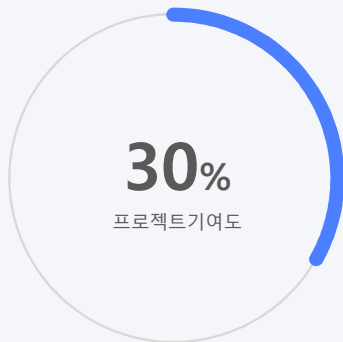
프레임 UI제작 / 관리프로그램을
이용한 회원관리, 재고관리, 이벤
트처리

WEB DEVELOPMENT *PORTFOLIO* 1차 프로젝트



WEB DEVELOPMENT *PORTFOLIO*

2차 프로젝트



2차 프로젝트

(URL : <https://github.com/Na-roo/zipBanchan>)

반찬쇼핑몰 웹App 구현

주요기술 : JAVA, JSP, javascript, Oracle

프로젝트인원 : 7명 (팀원)

개발기간 : 2019.07.29 – 2019.08.09



개발목표

Step. 1

JAVA, JSP를 이용하여 집반찬쇼
핑몰 webApp 재구현



구현설명

Step. 2

메인, 카테고리, 상세페이지 이동,
상품분류 및 정렬 (CRUD 구현)



담당업무

Step. 3

- 메인홈페이지 및 상품메뉴, 상
품디테일 구현 및 페이지구축
- DB연동하여 상품 데이터동기화

요구사항명세서

요구사항명세서

ID	이름	설명	유형	관련자
R001	회원가입	1. 사용자는 회원가입을 입력하여 회원가입을 할 수 있어야 한다. 2. 사용자는 비밀번호를 입력하여 회원가입을 취소할 수 있어야 한다. 3. 사용자는 아이디를 중복체크할 수 있는 버튼을 누를 수 있어야 한다.	기능	사용자
R002		1. 시스템은 사용자가 회원가입을 할 때, 비밀번호가 오류 입력되어 있지 않으면, 회원가입을 할 수 있게 해야한다. 2. 아이디는 알phan, 숫자만 입력 가능하여, 아이디의 글자수는 4-16자로 제한할 수 있어야 한다. 3. 비밀번호는 알phan, 숫자만 입력 가능하여, 비밀번호의 글자수는 10-16자로 제한할 수 있어야 한다. 4. 비밀번호와 비밀번호 확인에 입력한 글자는 동일해야 한다. 5. 시스템은 사용자가 아이디(이메일)을 입력하고 '아이디 중복체크' 버튼을 누르면 '아이디(이메일)을 입력하세요' 라는 메시지를 출력할 수 있어야 한다. 6. 시스템은 사용자가 아이디(이메일)을 입력하고 '아이디 중복체크' 버튼을 누르면, '아이디 중복된 아이디(이메일)이 있습니다.', '아이디 중복된 아이디입니다.' 라는 메시지를 출력할 수 있어야 한다. 7. 시스템은 사용자가 입력한 비밀번호를 8에 저장할 수 있어야 하며, 비밀번호는 암호화 하여 저장해야 한다. 8. 시스템은 사용자가 '회원가입' 버튼을 누르면 8에 회원 정보를 저장할 수 있어야 한다. 9. 시스템은 사용자가 '이전으로' 버튼을 눌러 회원가입을 취소할 수 있게 해야한다.	비기능	시스템
R003	로그인	1. 사용자는 아이디와 비밀번호를 입력하여 로그인을 할 수 있어야 한다.	기능	사용자
R004		1. 시스템은 사용자가 입력한 아이디와 비밀번호가 8에 일치하지 확인할 수 있어야 한다. 2. 시스템은 사용자가 입력한 아이디와 비밀번호가 8에 일치하면, '아이디 또는 비밀번호를 다시 확인하세요, 등록되지 않은 아이디이거나, 아이디 또는 비밀번호를 잘못 입력하셨습니다.' 라는 메시지를 출력해야 한다. 3. 시스템은 사용자가 로그인하면 로그인 성공하면, 사용자 아이디를 저장할 수 있어야 하며, 사용자가 로그인 실패를 경우에는 저장하지 않아야 한다.	비기능	시스템
R005	로그아웃	1. 사용자는 로그아웃을 할 수 있어야 한다.	기능	사용자
R006		1. 시스템은 사용자가 로그아웃을 할 때, 해당 사용자의 세션을 삭제해야 한다.	비기능	시스템
R007	마이페이지	1. 사용자는 마이페이지에서 회원정보 수정, 회원탈퇴, 주문정보를 확인할 수 있어야 한다.	기능	사용자
R008		1. 시스템은 사용자가 회원정보를 수정할 수 있도록 해야한다. 2. 시스템은 사용자가 회원탈퇴를 할 수 있도록 해야한다. 3. 사용자가 회원탈퇴를 하면, 시스템은 8에서 해당 사용자의 모든 데이터를 삭제해야 한다. 4. 시스템은 사용자가 주문한 주문 취소도를 출력할 수 있다. 5. 시스템은 로그인한 사용자에게만 마이페이지를 이용할 수 있도록 해야한다.	비기능	시스템

R009	고객추기	1. 사용자는 고객추기를 작성할 수 있어야 한다. 2. 사용자는 고객추기를 작성지 또는 목록으로 검색할 수 있어야 한다. 3. 사용자는 고객추기를 등록순, 정렬순으로 정렬할 수 있어야 한다. 4. 사용자는 고객추기를 제목을 클릭하면, 해당 고객추기의 상세정보를 확인할 수 있어야 한다. 5. 사용자는 고객추기 게시판에서 상품의 상세정보를 확인할 수 있어야 한다.	기능	사용자
R010		1. 시스템은 8에 저장된 고객추기 정보들을 등록순, 정렬순으로 출력할 수 있어야 한다. (기본값은 등록순으로 지정이 된다.) 2. 시스템은 사용자가 입력한 검색어로 8에 저장된 고객추기 정보들을 출력할 수 있어야 한다. 3. 검색어지 분석은 제목, 작성자로 구분되며, 제목은 상품명, 작성자는 해당 아이디를 기준으로 검색한다.	비기능	시스템
R011	상품	1. 사용자는 메뉴를 선택하여 상품들을 확인할 수 있어야 한다. 2. 사용자는 상품들을 낮은 가격순, 높은 가격순, 최신순, 상품별 순으로 정렬할 수 있어야 한다. 3. 사용자는 상품의 상세정보를 확인할 수 있어야 한다. 4. 사용자가 상품을 수량을 변경하면, 수량이 없게 변경된 상품의 총 금액을 확인할 수 있어야 한다.	기능	사용자
R012		5. 사용자는 상품의 수량을 지정하여 바구니에 또는 장바구니에 담을 수 있어야 한다. 1. 시스템은 사용자가 선택한 메뉴에 있는 상품 리스트를 표시하며, 이 때, 각각의 상품에 상품 이미지, 상품명, 상품 가격, 상품 수량을 표시한다. 2. 시스템은 상품의 재고량이 0이면, 품절이라고 표시해야하며, 상품 상세정보에서 총 금액을 표시하지 않아야 한다. 3. 시스템은 상품 리스트를 낮은 가격순, 높은 가격순, 최신순, 상품별 순으로 정렬해서 표시할 수 있어야 한다. 4. 시스템은 사용자가 상품 수량을 지정하여 "바구니" 또는 "장바구니"를 선택하면 해당 상품 정보들을 8에 저장해야 한다.	비기능	시스템
R013	장바구니	1. 사용자는 장바구니에 등록된 상품들을 확인할 수 있어야 하며, 해당 상품의 상품명, 금액, 가격, 상품내용, 상품수량, 합계를 확인할 수 있어야 한다. 2. 사용자는 장바구니에서 등록된 상품의 수량을 변경할 때, 해당 상품의 합계와 전체 상품의 총 합계와 가격이 변동되는 것을 볼 수 있어야 한다. 3. 사용자는 장바구니에 등록된 상품들에 대해 개별삭제, 선택된 상품 삭제, 전체삭제를 할 수 있어야 한다.	기능	사용자
R014		1. 시스템은 사용자가 장바구니의 수량을 변경할 때, 0 또는 음수로 초기화해 하서는 안되며, 8에 입력된 수량만큼 초과하면 메시지를 출력해야 한다.	비기능	시스템
R015	주문	1. 사용자는 장바구니에 등록된 상품들을 주문할 수 있어야 한다. 2. 사용자는 주문자 정보를 입력할 수 있어야 한다. 3. 사용자는 배송 정보를 입력할 수 있어야 한다. 4. 사용자는 결제금액을 할 수 있어야 한다.	기능	사용자
R016		1. 주문자 정보, 배송지 정보는 로그인한 사용자의 회원정보를 기준으로 설정한다. 2. 시스템은 사용자가 주문하기 버튼을 누르면 주문완료 메시지를 표시하도록 한다.	비기능	시스템

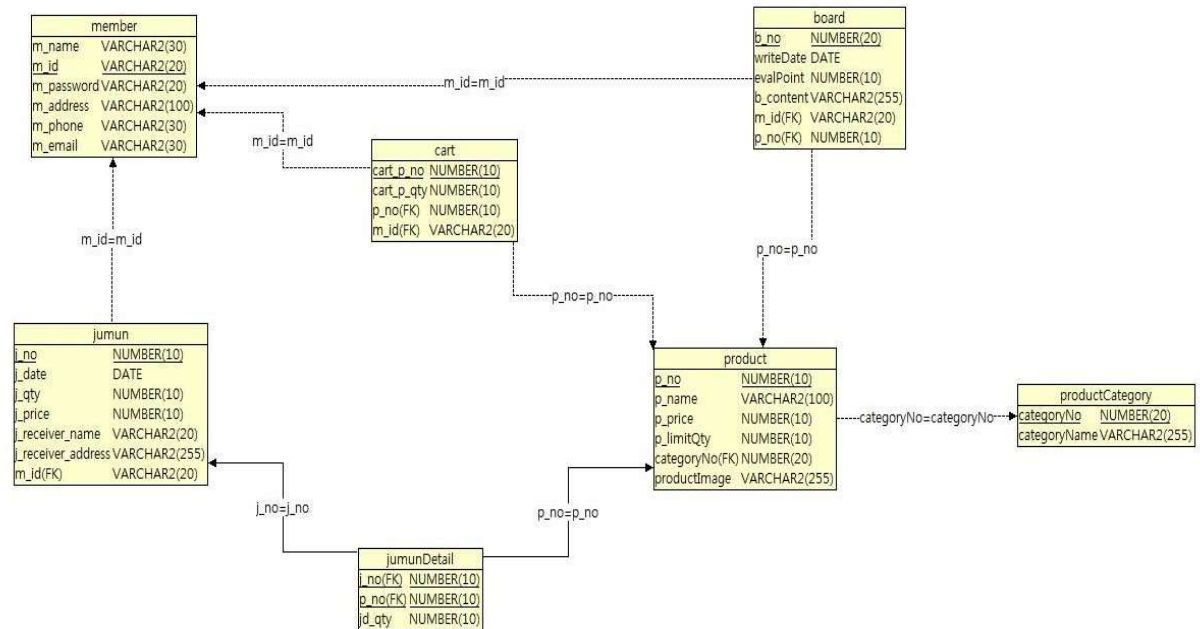
공통용어집

ID	이름	설명
0001	회원정보	사용자의 이름, 아이디(이메일), 비밀번호, 전화번호, 주소들을 의미한다.
0002	고객추기정보	상품명, 상품이미지, 작성자, 작성일, 정렬순을 의미한다.
0003	주문자정보	주문자 이름, 전화번호, 이메일을 의미한다.
0004	배송지정보	배송사명 이름, 배송지 주소, 전화번호를 의미한다.

PACKAGE

```
zipbanchan 1026 [https://182.237.126.19/svn/javabig2/web_projec,
> Deployment Descriptor: zipbanchan
> JAX-WS Web Services
> Java Resources
  > src 1025
    > com.itwill.board 997
      > com.itwill.board.controller 997
    > com.itwill.cart 1009
      > com.itwill.cart.controller 1009
    > com.itwill.dispatcher 750
      > com.itwill.dispatcher.filter 27
    > com.itwill.exception 119
    > com.itwill.jumun 999
      > com.itwill.jumun.controller 999
    > com.itwill.member 1016
      > com.itwill.member.controller 1016
    > com.itwill.product 812
      > Product.java 678
      > ProductDao.java 678
      > ProductMainTest.java 55
      > ProductService.java 399
      > ProductSQL.java 55
      > RdbmsDAO.java 55
    > com.itwill.product.controller 812
      > ProductDetailViewController.java 675
      > ProductListViewController.java 812
      > ProductSortActionController.java 812
      > ZipbanchanMainController.java 580
  > Libraries
> JavaScript Resources
> build
> external 1026
  > WebContent 1016
    > image 564
    > META-INF 21
    > res 977
  > WEB-INF 1016
    > index.jsp 977
    > table.jsp 977
```

CLASS DIAGRAM



Dao, Service, Controller

```

import class ProductDao{
private DataSource ds;

public ProductDao() throws Exception {
    InitialContext ic = new InitialContext();
    ds = (DataSource) ic.lookup("java:comp/env/jdbc/OracleDB")
}

// 상품 추가
public boolean create(Product product) throws Exception {
    Connection con = null;
    PreparedStatement pstmt = null;
    String insertSql
    ="insert into product values(p_seq.nextval,?,?,?,?)";
    boolean isSuccess = false;
    try{
        con = ds.getConnection();
        pstmt = con.prepareStatement(insertSql);

        pstmt.setString(1, product.getProductNm());
        pstmt.setInt(2, product.getProductPrice());
        pstmt.setInt(3, product.getProductQty());
        pstmt.setInt(4, product.getProductCategoryNo());
        pstmt.setString(5, product.getProductingTime());

        pstmt.executeUpdate();
        isSuccess = true;
    }finally{
        if(con!=null) con.close();
    }

    return isSuccess;
}

// 상품 이름으로 찾기
public Product findByName(String productname) throws Exception {
    Connection con = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    String selectSql = "select * from product where p_name=?";
    Product product = null;
    try {
        con = ds.getConnection();
        pstmt = con.prepareStatement(selectSql);
        pstmt.setString(1, "p_name");
        rs = pstmt.executeQuery();
        if (rs.next()) {
            product = new Product(rs.getInt("p_no"),

```

```

import class ProductService {
    private ProductDao productDao;

    public ProductService() throws Exception{
        productDao = new ProductDao();
    }

    //등록하기
    public boolean create(Product product) throws Exception {
        return productDao.create(product);
    }

    //등록번호 찾기
    public Product findMylo(int productlo) throws Exception {
        return productDao.findMylo(productlo);
    }

    //등록번호로 찾기
    public Product findMyName(String productName) throws Exception {
        return productDao.findMyName(productName);
    }

    //등록번호 찾기
    public ArrayList<Product> findMyAll() throws Exception {
        return productDao.findAll();
    }

    //등록번호와 카테고리 찾기
    public ArrayList<Product> findCategory(int categorylo) throws Exception {
        return productDao.findcateList(categorylo);
    }

    //등록번호
    public boolean remove(int productlo) throws Exception {
        return productDao.remove(productlo);
    }

    //등록번호 수정
    public boolean update(Product product) throws Exception {
        return productDao.update(product);
    }

    //등록번호 수정하여 가격 정보 변경
    public boolean updateByloInvenDao(int productlo, int templo) throws Exception {
        return productDao.updateByloInvenDao(productlo, templo);
    }

    //등록번호를 정렬
    public ArrayList<Product> sort(int categorylo, String columnname, String sortOrder) throws Exception {
        return productDao.sort(categorylo, columnname, sortOrder);
    }
}

```

```

public class ProductListViewController implements Controller {

    @Override
    public String handleRequest(HttpServletRequest request, HttpServletResponse response) {
        String forwardPath = "";
        String productCategory = request.getParameter("categoryId");
        String columnName = request.getParameter("columnName");
        String sortOrder = request.getParameter("sortOrder");
        if (productCategory == null || productCategory == "") {
            forwardPath = "redirect:/main.jsp";
        }
        else {
            try {
                ProductService productService = new ProductService();
                ArrayList<Product> categoryList = productService.findCategory(Integer.parseInt(productCategory));
                request.setAttribute("categoryList", categoryList);
                forwardPath = "forward:/productListView.jsp";
            }
            if (sortOrder != null && columnName != null) {
                productService = new ProductService();
                ArrayList<Product> productList = productService.sort(Integer.parseInt(productCategory), columnName, sortOrder);
                if (productList == null) {
                    forwardPath = "redirect:/main.jsp";
                }
                else {
                    request.setAttribute("productList", productList);
                    forwardPath = "forward:/productListView.jsp";
                    System.out.println(productList);
                }
            }
        }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    return forwardPath;
}
}

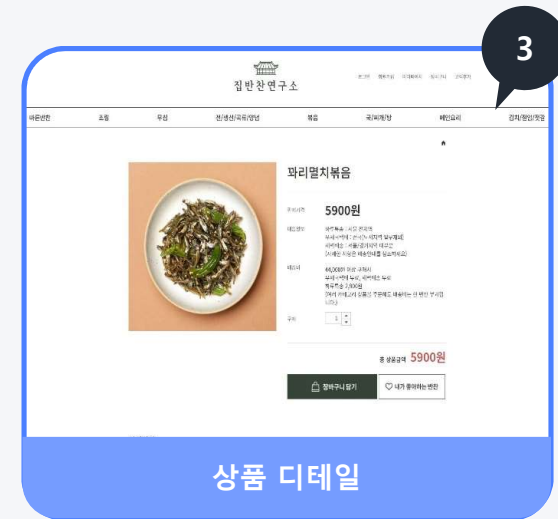
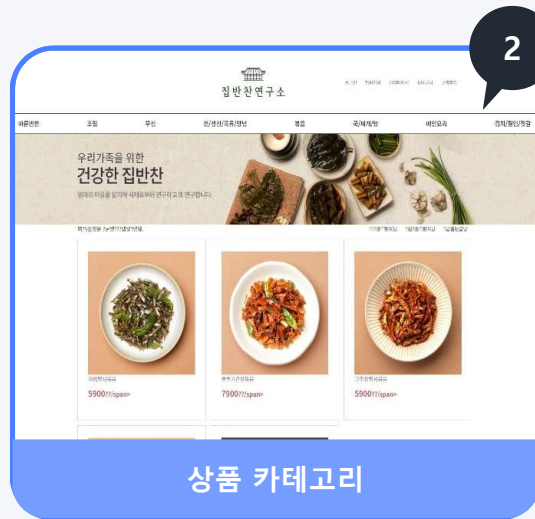
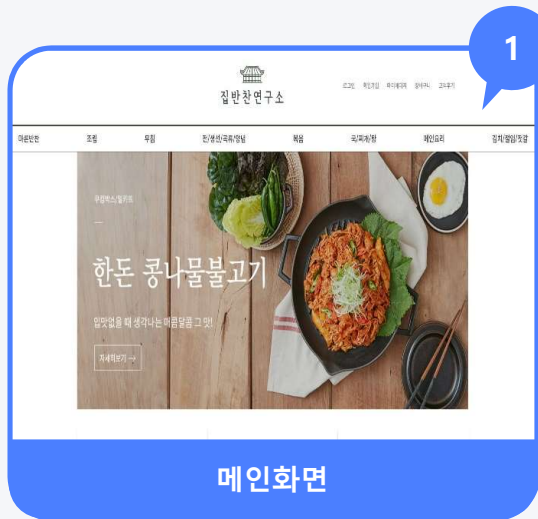
```

```
public class ProductDetailViewController implements Controller {

    @Override
    public String request(HttpServletRequest request, HttpServletResponse response) {
        String forwardPath="";
        String productNo=request.getParameter("productNo");
        if (productNo==null||productNo.equals("")) {
            forwardPath="redirect:productListView.jsp";
        }else {
            try {
                ProductService productService = new ProductService();
                Product product = productService.findById(Integer.parseInt(productNo));
                if (product==null) {
                    forwardPath="redirect:productListView.jsp";
                }else {
                    request.setAttribute("product", product);
                    forwardPath="forward:/productDetailView.jsp";
                }
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }

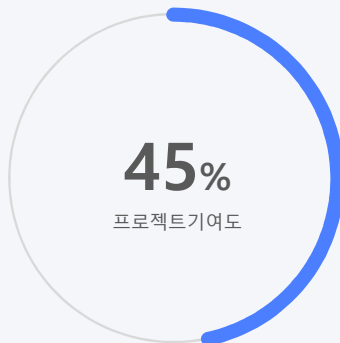
    return forwardPath;
}
}
```


WEB DEVELOPMENT *PORTFOLIO* 2차 프로젝트



WEB DEVELOPMENT *PORTFOLIO*

3차 프로젝트



3차 프로젝트

(URL : <https://github.com/Na-roo/leeLeung>)

가구쇼핑몰 웹App 구현

주요기술 : JAVA, Spring, myBatis, JSP, javascript, Oracle

프로젝트인원 : 7명 (팀장)

개발기간 : 2019.11.01 – 2019.11.21



개발목표

Step. 1

JAVA, JSP, Spring를 이용하여
마켓비쇼핑몰 webApp 재구현



구현설명

Step. 2

문의게시판 작성, 수정, 삭제 및
게시판 페이징처리 (CRUD 구현)



담당업무

Step. 3

- 프로젝트 기획 단계 (아이템선정, 요구사항명세서, 화면정의서 서류작성)
- 메인 및 서브페이지, 게시판 UI 디자인 및 기획 총괄
- DB연동하여 게시물 데이터동기화
- 문의게시판 작성, 수정, 삭제 및 게시판페이징 처리

요구사항명세서

요구사항명세서

ID	이름	설명	유형	관련자
R001	회원가입	1. 사용자는 회원가입을 입력하여 회원가입을 할 수 있어야 한다. 2. 사용자는 최소비밀번호를 입력하여 회원가입을 취소할 수 있어야 한다. 3. 사용자는 아이디를 중복체크할 수 있는 버튼을 누를 수 있어야 한다.	기능	사용자
R002		1. 시스템은 사용자가 회원가입을 할 때, 회원정보가 모두 입력되어 있지 않으면, 회원가입을 할 수 있게 해야한다. 2. 아이디는 알phanumeric, 숫자만 입력 가능하며, 아이디의 길이는 8~18자로 제한할 수 있어야 한다. 3. 비밀번호는 영문 대소문, 숫자만 입력 가능하며, 비밀번호의 길이는 8~18자로 제한할 수 있어야 한다. 4. 비밀번호와 비밀번호 확인 칸에 입력된 글자는 동일해야 한다. 5. 시스템은 사용자가 아이디(이메일)을 입력하지 않고 '아이디 중복체크' 버튼을 누르면 '아이디(이메일)을 입력하세요' 라는 메시지를 출력할 수 있어야 한다. 6. 시스템은 사용자가 아이디(이메일)을 입력하고 '아이디 중복체크' 버튼을 클릭할 때, 8에 이미 등록된 아이디(이메일)이 있다면, '이미 등록된 아이디입니다.' 라는 메시지를 출력할 수 있어야 한다. 7. 회원가입 시스템은 주소로 하이픈의 이용하여 사용자의 주소를 입력 할 수 있게 하여야 한다. 8. 시스템은 사용자가 입력한 회원정보를 DB에 저장할 수 있어야 하며, 비밀번호를 암호화 하여 저장해야 한다. 9. 시스템은 사용자가 '회원가입' 버튼을 누르면 8에 회원 정보를 저장할 수 있어야 한다. 10. 시스템은 사용자가 '이전으로' 버튼을 눌러 회원가입을 취소할 수 있게 해야한다.	비기능	시스템
R003	로그인	1. 사용자는 아이디와 비밀번호를 입력하여 로그인을 할 수 있어야 한다. 2. 사용자는 카카오계정으로 로그인 할 수 있어야 한다.	기능	사용자
R004		1. 시스템은 사용자가 입력한 아이디와 비밀번호가 8에 있는지 확인할 수 있어야 한다. 2. 시스템은 사용자가 입력한 아이디와 비밀번호가 8에 없으면, '아이디 또는 비밀번호를 다시 확인하세요. 등록되지 않은 아이디이거나, 아이디 또는 비밀번호를 잘못 입력하셨습니다.' 라는 메시지를 출력해야 한다. 3. 시스템은 사용자가 한 번도 로그인한 적이 없으면, 사용자 아이디를 저장할 수 있어야 하며, 사용자가 한 번도 로그인을 하지 않으면 비밀번호를 초기화해야 한다.	비기능	시스템

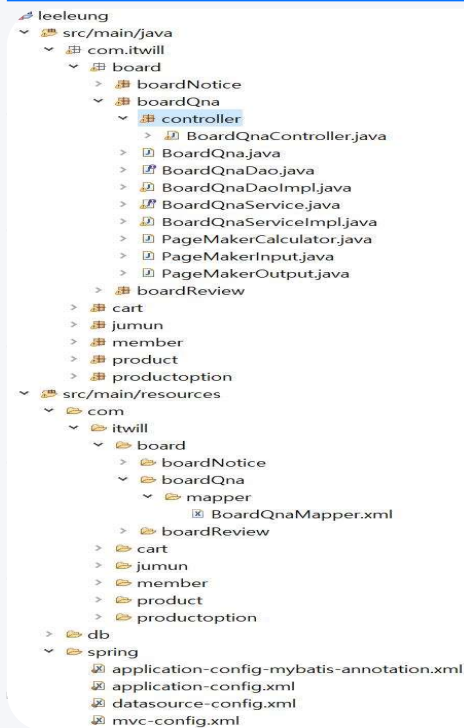
		4. 사용자는 아이디 및 비밀번호 찾기용 비밀번호 확인가이드 시스템 이메일로 OTP인증번호를 받을 수 있을 수 있다.		
R005	로그아웃	1. 사용자는 로그아웃을 할 수 있어야 한다.	기능	사용자
R006		1. 시스템은 사용자가 로그아웃을 할 때, 해당 사용자의 세션을 삭제해야 한다.	비기능	시스템
R007	마이페이지	1. 사용자는 마이페이지에서 회원정보 수정, 회원탈퇴, 주문정보를 확인할 수 있어야 한다.	기능	사용자
R008		1. 시스템은 사용자가 회원정보를 수정할 수 있도록 해야한다. 2. 시스템은 사용자가 회원탈퇴를 할 수 있도록 해야한다. 3. 사용자가 회원탈퇴를 하면, 시스템은 DB에서 해당 사용자의 모든 데이터를 삭제해야 한다. 4. 시스템은 사용자가 주문한 주문 리스트를 출력해야 한다. 5. 시스템은 로그인한 사용자에게만 마이페이지를 이용할 수 있도록 해야한다.	비기능	시스템
R009	고객후기	1. 사용자는 고객후기를 작성할 수 있어야 한다. 2. 사용자는 고객후기를 작성자 또는 제목으로 검색할 수 있어야 한다. 3. 사용자는 고객후기를 등록순, 평점순으로 정렬할 수 있어야 한다. 4. 사용자는 고객후기를 제목을 선택하면, 해당 고객후기의 상세정보를 확인할 수 있어야 한다. 5. 사용자는 고객후기 게시판에서 상품에 상세정보를 확인할 수 있어야 한다.	기능	사용자
R010		1. 시스템은 DB에 저장된 고객후기 정보들을 등록순, 평점순으로 출력할 수 있어야 한다. (기본값은 등록순으로 정렬이 된다.) 2. 시스템은 사용자가 입력한 검색어로 DB에 저장된 고객후기 정보들을 출력할 수 있어야 한다. 3. 검색어의 종류는 제목, 작성자로 구분되며, 제목은 상품명, 작성자는 회원 아이디를 기준으로 검색한다.	비기능	시스템
R011	상품	1. 사용자는 메뉴를 선택하여 상품들을 확인할 수 있어야 한다. 2. 사용자는 상품명을 검색하면, 높은 가격순, 최신순, 상품평 순으로 정렬할 수 있어야 한다. 3. 사용자는 상품의 상세정보를 확인할 수 있어야 한다. 4. 사용자는 상품의 수량을 변경하면, 수량이 변경된 상품에 총 금액을 확인할 수 있어야 한다. 5. 사용자는 상품의 수량을 지정하여 바구니에 또는 장바구니에 담을 수 있어야 한다.	기능	사용자

R012		1. 시스템은 사용자가 선택한 메뉴에 있는 상품 리스트를 표시하며, 이 때, 각장의 상품에 상품 이미지, 상품명, 상품 가격, 상품 수량을 표시한다. 2. 시스템은 상품의 재고량이 0이면, 품절이라고 표시를 해야하며, 상품 상세정보에서 총 금액을 표시하지 않아야 한다. 3. 시스템은 상품 리스트를 높은 가격순, 높은 가격순, 최신순, 상품평 순으로 정렬해서 표시할 수 있어야 한다. 4. 시스템은 사용자가 상품을 수량을 지정하여 '바구니' 또는 '장바구니' 탭을 선택하면 해당 상품 정보들을 DB에 저장해야 한다.	비기능	시스템
R013	장바구니	1. 사용자는 장바구니에 등록된 상품들을 확인할 수 있어야 하며, 해당 상품의 상품명, 판매가격, 상품내용, 상품수량, 합계를 확인할 수 있어야 한다. 2. 사용자는 장바구니에서 등록된 상품의 수량을 변경할 때, 해당 상품의 합계와 전체 상품에 총 합계와 가격이 반영되는 것을 볼 수 있어야 한다. 3. 사용자는 장바구니에 등록된 상품들에 대해 개별삭제, 선택된 상품삭제, 전체삭제를 할 수 있어야 한다.	기능	사용자
R014		1. 시스템은 사용자가 장바구니의 수량을 변경할 때, 0 또는 음수로 표기되지 해서는 안되며, DB에 입력된 수량을 초과하면 메시지를 출력해야 한다.	비기능	시스템
R015	주문	1. 사용자는 장바구니에 등록된 상품들을 주문할 수 있어야 한다. 2. 사용자는 주문지 정보를 입력할 수 있어야 한다. 3. 사용자는 배송 정보를 입력할 수 있어야 한다. 4. 사용자는 결제금액을 할 수 있어야 한다.	기능	사용자
R016		1. 주문지 정보, 배송지 정보는 로그인한 사용자의 회원정보를 기본값으로 설정한다. 2. 시스템은 사용자가 주문하기 버튼을 누르면 주문정보 페이지를 표시하도록 한다.	비기능	시스템

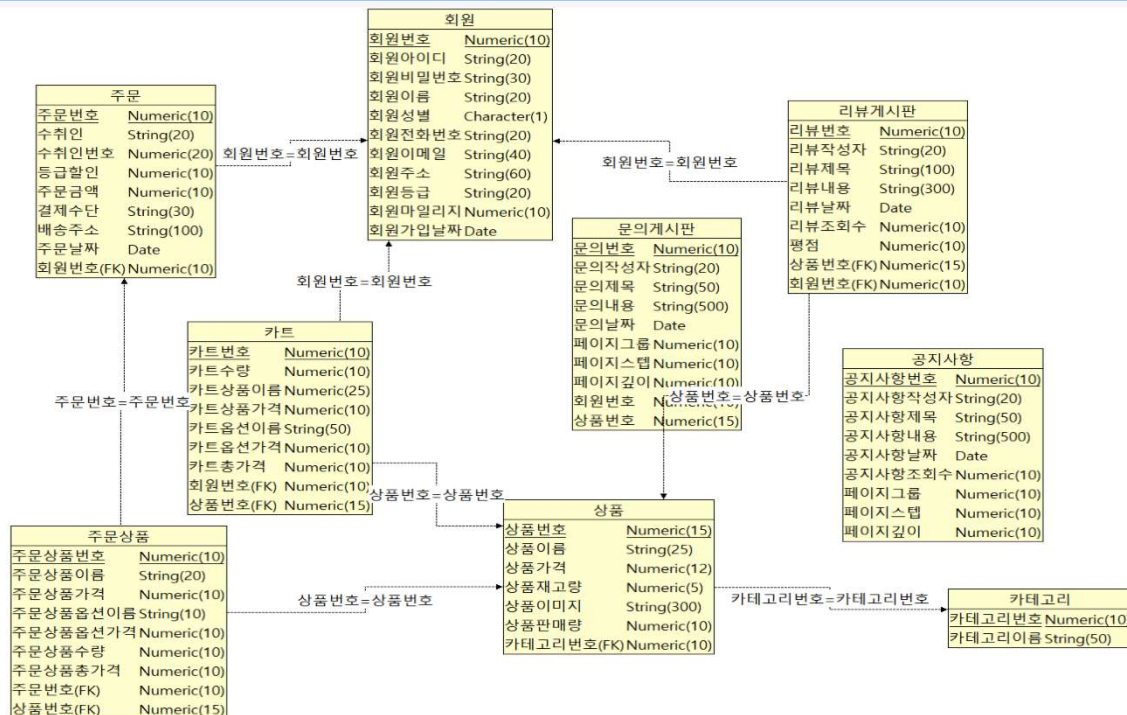
공통용어집

ID	이름	설명
C001	회원정보	사용자의 이름, 아이디(이메일), 비밀번호, 전화번호, 주소들을 의미한다.
C002	고객후기정보	상품명, 상품이미지, 작성자, 작성일들을 의미한다.
C003	주문자정보	주문자 이름, 전화번호, 이메일을 의미한다.
C004	배송지정보	배송지명, 배송지 주소, 전화번호를 의미한다.

PACKAGE



CLASS DIAGRAM



Dao, Service, Controller, Mapper

```

<!-- dao -->
package com.itwill.board;

import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class BoardDao {
    private static String DB_NAME = "com.itwill.board.mapper.BoardMapper";

    private static Session session;

    public BoardDao() throws Exception {
        System.out.println("BoardDao 생성");
    }

    public void setSession(Session session) {
        System.out.println("BoardDao.setSession(" + session + ")");
        this.session = session;
    }

    // ... (other methods) ...
}

```

```

<!-- service -->
package com.itwill.board;

import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class BoardService {
    private BoardDao boardDao;

    public BoardService() throws Exception {
        // TODO Auto-generated constructor stub
    }

    // ... (other methods) ...
}

```

```

<!-- controller -->
package com.itwill.board;

import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class BoardController {
    private BoardService boardService;

    // ... (other methods) ...
}

```

```

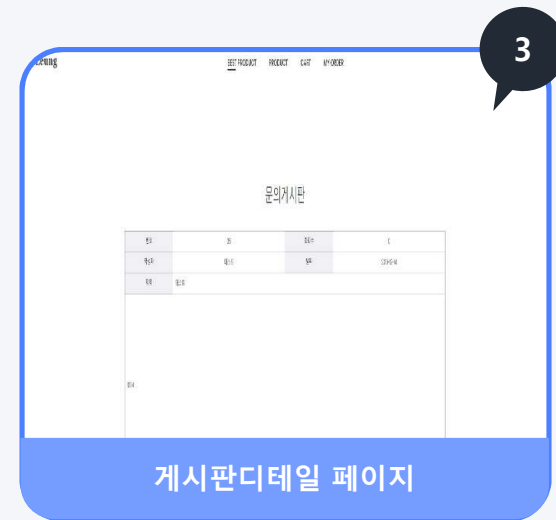
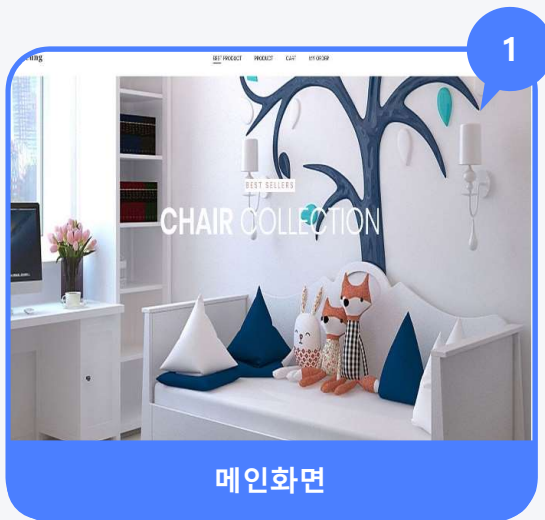
<!-- mapper -->
package com.itwill.board;

import java.util.List;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class BoardMapper {
    // ... (other methods) ...
}

```

WEB DEVELOPMENT *PORTFOLIO* 3차 프로젝트



WEB DESIGN *PORTFOLIO*

웹디자인 및 퍼블리싱

100%

프로젝트기여도

웹디자인 포트폴리오

(URL : <https://github.com/Na-roo/webDesign-portFolio>)

가구쇼핑몰 웹App 구현

주요기술 : HTML, CSS, jQeury

개인 프로젝트

구축기간 : 2017.08 - 2017.09



구축목표

Step. 1

HTML, CSS, jQuery를 이용하여
포트폴리오 웹사이트를 구축



구현설명

Step. 2

jQuery를 이용하여 스크롤이벤트
트 및 애니메이션 효과



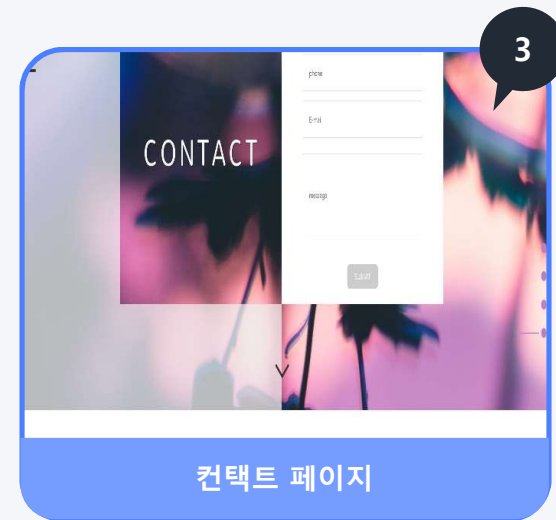
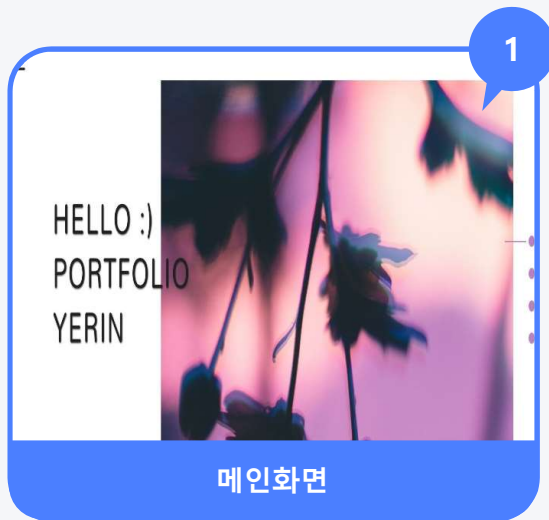
담당업무

Step. 3

- 마우스스크롤시 각 section 노출
- 마우스클릭시 인디케이터 이벤트
- 애니메이션 효과

WEB DESIGN *PORTFOLIO*

웹 디자인 및 퍼블리싱



끝까지 봐주셔서 감사합니다 :D