Start.

- 이름 : 김민국
- 프로젝트명 - JAVA/JSP + MySql , MVC MODEL2를 사용해서 구현한 웹 페이지

INDEX

01. 주제선정 및 배경

01. 주제선정 및 배경

✓ 주　　제 :　관리자, 판매자, 이용자를 위한 종합 웹 쇼핑몰 구현

✓ 선정 배경 :

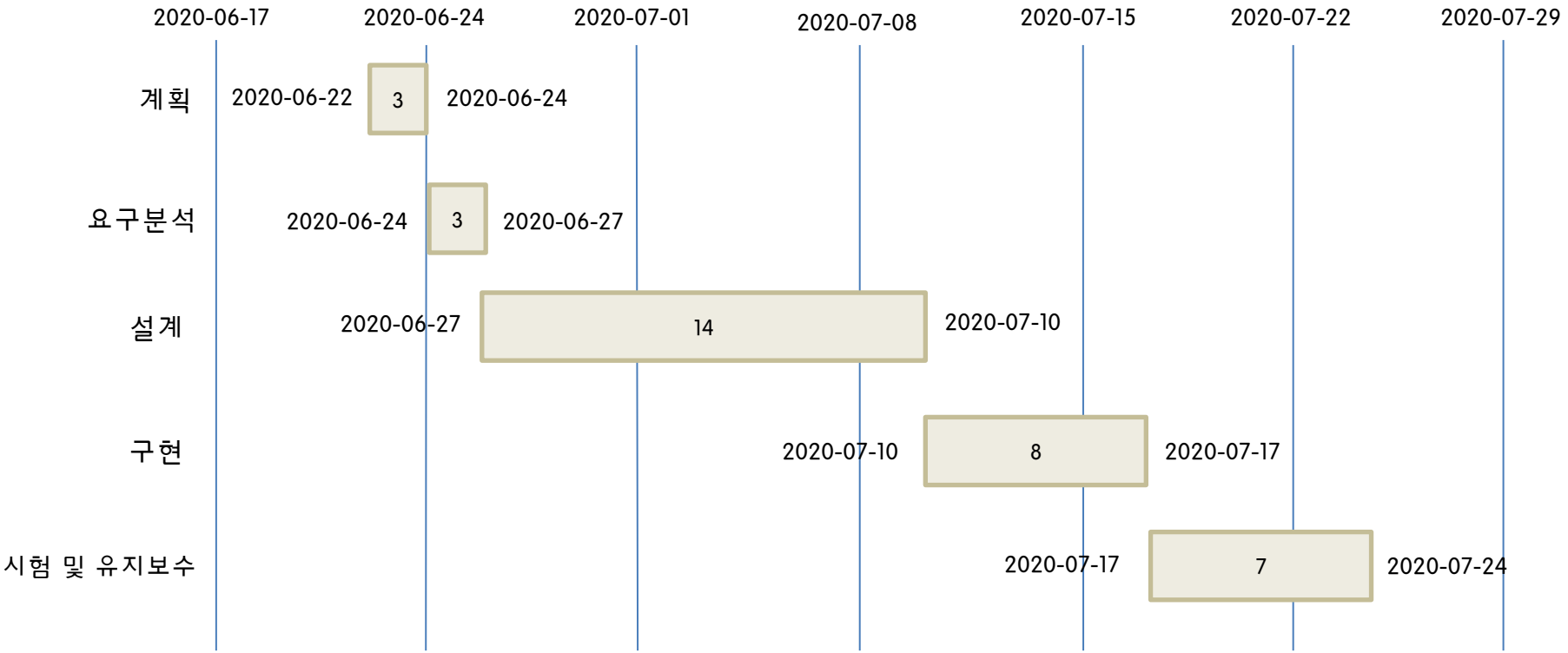
- 네이버스토어팜을 이용한 판매자로써의 경력을 바탕으로 사용에 많은 제약조건의 불편함을 해소하기 위해
- 관리자, 판매자, 쇼핑자 요구사항이 무엇인지 정확하게 알고 있기 때문에
- 이를 바탕으로 실제로 고객(관리자, 판매자, 쇼핑자)의 필요한 니즈를 바탕으로 웹 쇼핑몰을 구현하게 되었습니다.

02. 계획

2.1 일정 계획

총 개발 기간	2020.06.22 ~ 2020.07.24 (1개월)	
소 작업	웹 쇼핑몰 개발	
소 작업별 개발 기간	소작업	개발 기간
	계획	6월 22일 ~ 6월 24일
	요구 분석	6월 25일 ~ 6월 27일
	설계	6월 27일 ~ 7월 10일
	구현	7월 10일 ~ 7월 17일
	시험 및 유지보수	7월 18일 ~ 7월 24일
개발 순서	계획 → 요구 분석 → 설계 → 구현 → 시험 및 유지 보수	
필요 자원	개발용 PC, 개발 공간	

2.1 일정 계획(간트 차트)


2020-06-17
2020-06-24
2020-07-01
2020-07-08
2020-07-15
2020-07-22
2020-07-29
계획
2020-06-22
3
2020-06-24
요구분석
2020-06-24
3
2020-06-27
설계
2020-06-27
14
2020-07-10
구현
2020-07-10
8
2020-07-17
시험 및 유지보수
2020-07-17
7
2020-07-24

03. 설계

3. 1 데이터 베이스 설계

상품(product)		
Logical Name	Name	Type
상품아이디	styleno - PK	varchar(20)
상품명	c_name	varchar(20)
상품컬러	color	varchar(10)
상품가격	unitPrice	int
사이즈표	size	varchar(20)
상품설명	description	text
제조사	manufacturer	varchar(20)
분류	category	varchar(20)
재고수	unitPrice	int
업로드파일명	filename	varchar(20)

주문(ordertable)		
Logical Name	Name	Type
주문번호	order_no - PK	varchar(50)
유저아이디	user_id - FK	varchar(20)
상품아이디	styleno - FK	varchar(20)
상품명	c_name	varchar(20)
수량	quantity	int
주문일자	sdate	date

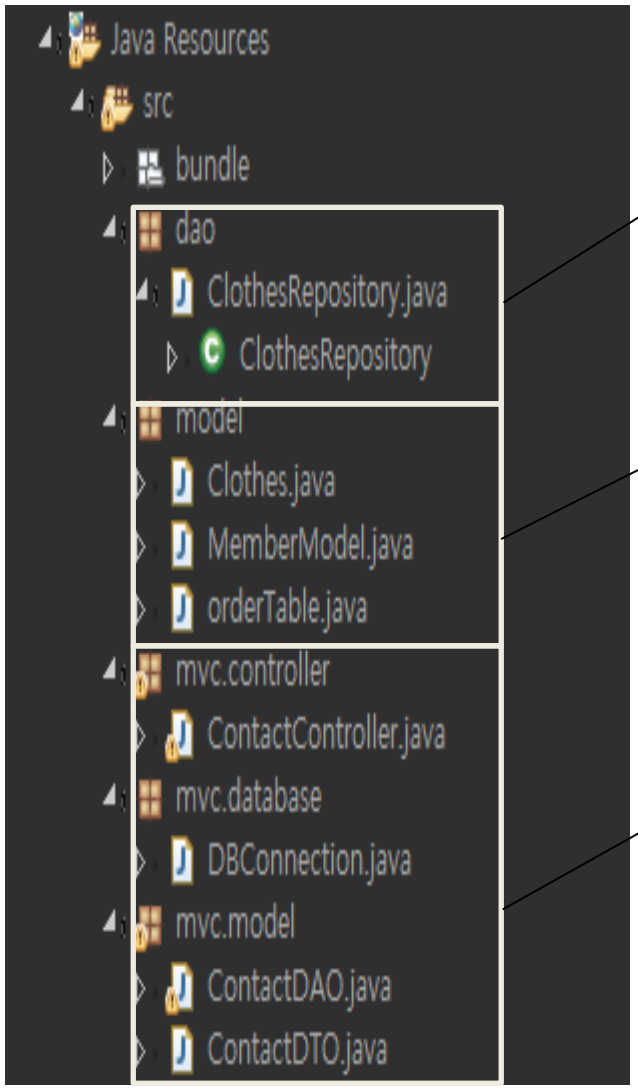
회원(member)		
Logical Name	Name	Type
아이디	user_id - PK	varchar(20)
패스워드	user_pw	varchar(20)
이름	user_name	varchar(20)
성별	user_gender	char(1)
생년월일	user_birth	date
이메일	user_email	varchar(40)
핸드폰번호	user_phone	varchar(15)
주소	user_address	text
가입일	resist_day	datetime

고객문의함(adminContact)		
Logical Name	Name	Type
게시글순번	num- PK	int - auto_increment
유저 아이디	user_id - FK	varchar(20)
유저 이름	user_name	varchar(20)
유저 연락처	user_phone	varchar(15)
게시글 제목	subject	text
게시글 내용	content	text
게시글 등록 일자	regist_day	datetime
게시글 등록시 IP	ip	varchar(20)

관리자(ADMINMEMBER)		
관리자아이디	admin_id	varchar(20)
관리자비밀번호	admin_pw	varchar(20)

04. 화면 구현

4. 1 패키지 설명

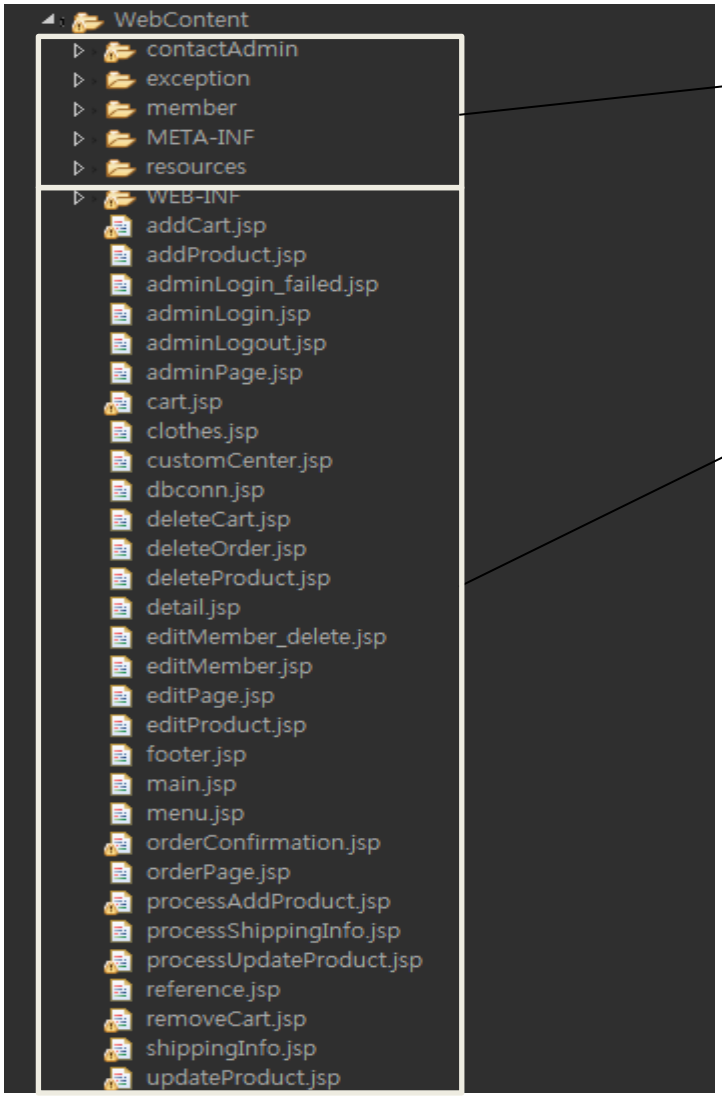


상품의 Dto와 Dao
- DB의 상품 테이블을 가져와서 ArrayList에 추가하는 부분

상품, 회원, 주문 테이블의 Dto

문의 게시판의 Model과 Controller, 데이터베이스 연결

4. 1 패키지 설명



@ contactAdmin : 문의 게시판의 View 부분
@ Exception : 페이지의 예외처리 부분
@ member : 회원가입, 수정, 삭제, 마이페이지 부분
@ resources : 부트스트랩css, 이미지폴더, SQL테이블 정리부분

@ WEB-INF 폴더 - web.xml : MVC의 서블릿 처리와 관리자 로그인 시에 필요한 제약 사항 설정
@ 화면 구현 시 필요한 View부분과 처리부분

4. 2 메인 화면 로직


홈 버튼
(메인화면이동)
상품의 카테고리를 파라미터 값으로 보내서 링크 액션시 예를들어 main.jsp?catecory=Top 이라면 메인화면에서 카테고리에 맞춰 상품 출력
Minstagram
로그인 회원가입 장바구니 배송조회 고객센터
CATEGORY All Top Outer Pants Shoes
Search 검색
검색 텍스트 창에 검색하면 상품명, 제조사, 가격 등 전체 검색
jQuery를 사용하여 만든 롤링배너
버튼이나 상품 이미지 클릭시 상품 상세정보 창으로 이동
빅로고 티셔츠
빅로고
12000원
상세 정보 »
무지 티셔츠
커버낫
상세 정보 »
후리스
네셔널지오그래픽
50000원
상세 정보 »

4. 2 메인 화면 로직(롤링배너)

```
<style type="text/css">
    .banner {position: relative; width: 1100px; height: 450px; top: 10px;  margin:0 auto; padding:0; overflow: hidden;}
    .banner ul {position: absolute; margin: 0px; padding:0; list-style: none; }
    .banner ul li {float: left; width: 1100px; height: 450px; margin:0; padding:0;}
</style>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></s

<script type="text/javascript">

    $(document).ready(function() {
        var $banner = $(".banner").find("ul");

        var $bannerWidth = $banner.children().outerWidth();//이미지의 폭
        var $bannerHeight = $banner.children().outerHeight(); // 높이
        var $length = $banner.children().length;//이미지의 갯수
        var rollingId;

        //정해진 초마다 함수 실행
        rollingId = setInterval(function() { rollingStart(); }, 4000);//다음 이미지로 롤링 애니메이션 할 시간차

        function rollingStart() {
            $banner.css("width", $bannerWidth * $length + "px");
            $banner.css("height", $bannerHeight + "px");
            //alert(bannerHeight);
            //배너의 좌측 위치를 옮겨 준다.
            $banner.animate({left: - $bannerWidth + "px"}, 1500, function() { //숫자는 롤링 진행되는 시간이다.
                //첫번째 이미지를 마지막 끝에 복사(이동이 아니라 복사)해서 추가한다.
                $(this).append("<li>" + $(this).find("li:first").html() + "</li>");
                //뒤로 복사된 첫번재 이미지는 필요 없으니 삭제한다.
                $(this).find("li:first").remove();
                //다음 움직임을 위해서 배너 좌측의 위치값을 초기화 한다.
                $(this).css("left", 0);
                //이 과정을 반복하면서 계속 롤링하는 배너를 만들 수 있다.
            });
        }
    });

</script>
```

배너의 크기를 설정해준다.

Find()함수와 children()함수를 사용해서 .banner의 자식 객체까지 포함시켜서 이미지를 설정하기 위해서 이다.

- animate()함수의 function은 이미지를 이어붙인 뒤, 앞에 이미지를 맨 뒤에 붙인후 그 이미지는 삭제하는 구조
- append()로 추가를하고 .remove()로 삭제를 한다.
- Animate()함수는 1.5초동안 왼쪽으로 배너의크기만큼 옮기고 fuction()의 기능을 수행하라는 의미

4. 3 주요 로직 (상품 상세정보)

4. 3 주요 로직 (회원가입)


http://mandoo1021.cafe24.com/member/addMember.jsp
검색...
회원가입
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
Language 개발자 블로그 관리자페이지
Minstagram
로그인 회원가입 장바구니 배송조회 고객센터
회 원 가 입
아이디 id 중복체크
비밀번호 password
비밀번호확인 password confirm
성명 name
성별 ○남 ○여
생일 년(4자) 월 일
이메일 @ naver.com
전화번호 phone
주소 우편번호 우편번호 찾기
도로명주소
지번주소
상세주소
등록 취소
중복체크 - Internet Ex...
아이디를 사용할 수 없습니다
닫기
DB member 테이블에서 select ~ where 조건문으로 일치하는 아이디가 있으면 사용 불가
Daum 우편번호 서비스 - Internet Explorer
tip
Daum의 우편번호 API를 가져와서 입력 후 사용

4. 3 주요 로직 (회원가입)

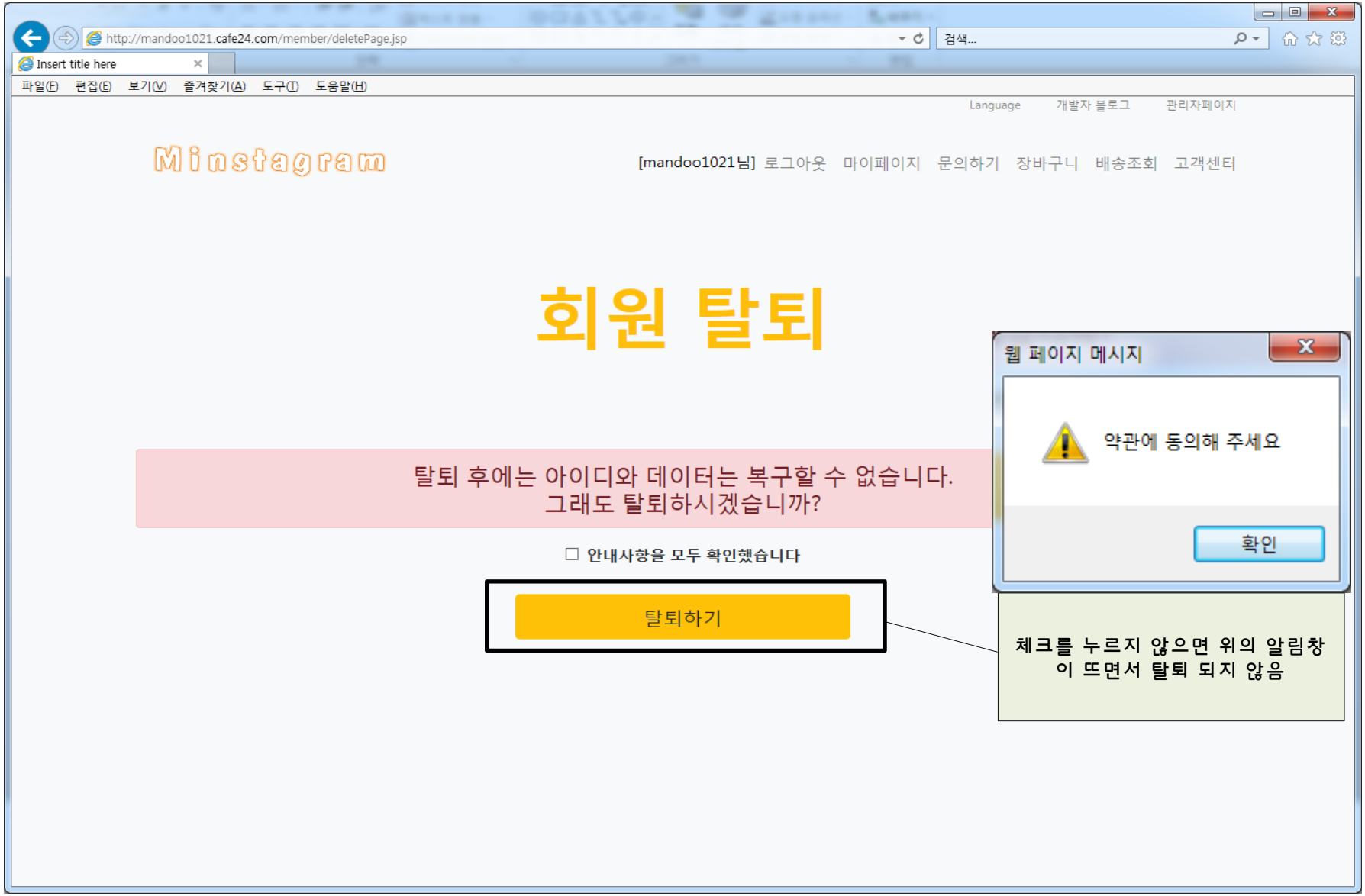
```
<script src="https://t1.daumcdn.net/mapjsapi/bundle/postcode/prod/postcode.v2.js"></script>
<script>
    function execDaumPostcode() {
        new daum.Postcode({
            oncomplete: function(data) {
                var roadAddr = data.roadAddress; // 도로명 주소 변수
                var extraRoadAddr = ''; // 참고 항목 변수
                // 법정동명이 있을 경우 추가한다. (법정리는 제외)
                // 법정동의 경우 마지막 문자가 "동/로/가"로 끝난다.
                if(data.bname !== '' && /[동|로|가]$/g.test(data.bname)){
                    extraRoadAddr += data.bname;
                }
                // 건물명이 있고, 공동주택일 경우 추가한다.
                if(data.buildingName !== '' && data.apartment === 'Y'){
                   extraRoadAddr += (extraRoadAddr !== '' ? ', ' + data.buildingName : data.buildingName);
                }
                // 표시할 참고항목이 있을 경우, 괄호까지 추가한 최종 문자열을 만든다.
                if(extraRoadAddr !== ''){
                    extraRoadAddr = ' (' + extraRoadAddr + ')';
                }
                // 우편번호와 주소 정보를 해당 필드에 넣는다.
                document.getElementById('postcode').value = data.zonecode;
                document.getElementById("roadAddress").value = roadAddr;
                document.getElementById("jibunAddress").value = data.jibunAddress;
                // 참고항목 문자열이 있을 경우 해당 필드에 넣는다.
                if(roadAddr !== ''){
                    document.getElementById("sample4_extraAddress").value = extraRoadAddr;
                } else {
                    document.getElementById("sample4_extraAddress").value = '';
                }
                var guideTextBox = document.getElementById("guide");
                // 사용자가 '선택 안함'을 클릭한 경우, 예상 주소라는 표시를 해준다.
                if(data.autoRoadAddress) {
                    var expRoadAddr = data.autoRoadAddress + extraRoadAddr;
                    guideTextBox.innerHTML = '(예상 도로명 주소 : ' + expRoadAddr + ')';
                    guideTextBox.style.display = 'block';

                } else if(data.autoJibunAddress) {
                    var expJibunAddr = data.autoJibunAddress;
                    guideTextBox.innerHTML = '(예상 지번 주소 : ' + expJibunAddr + ')';
                    guideTextBox.style.display = 'block';
                } else {
                    guideTextBox.innerHTML = '';
                    guideTextBox.style.display = 'none';
                }
            }
        }).open();
    }
</script>
```

우편번호 가져오기

```
<sql:setDataSource var="dataSource"
    url="jdbc:mysql://localhost:3306/FinalTestDB"
    driver="com.mysql.jdbc.Driver" user="root" password="1234" />

<sql:query dataSource="${dataSource}" var="resultSet">
   SELECT * FROM MEMBER WHERE user_id=?
   <sql:param value="<%=id%>" />
</sql:query>
<c:set var="result" value="0" />
<c:forEach var="row" items="${resultSet.rows}">
    <c:set var="result" value="1" />
</c:forEach>

<html>
<head>
<link rel="stylesheet" href="<c:url value="/resources/css/bootstrap.min.css"/>" />
<title>중복체크</title>
</head>
<body>
    <br>
    <div class="container" align="center">
        <h6 class="alert alert-danger">
            <c:if test="${result == 0}">아이디를 사용할 수 있습니다 </c:if>
            <c:if test="${result == 1}"> 아이디를 사용할 수 없습니다   </c:if>
        </h6>
    </div>
```

중복체크시 쿼리문으로 조건검색
Sql태그를 사용해서 간결한 코드 생성
Core태그로 경로를 지정해주는 이유는 동일 소스로 context path를 자동 포함시키기 위함

4. 3 주요 로직 (회원탈퇴)


http://mandoo1021.cafe24.com/member/deletePage.jsp
검색...
Insert title here
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
Language 개발자 블로그 관리자페이지
Minstagram
[mandoo1021님] 로그아웃 마이페이지 문의하기 장바구니 배송조회 고객센터
회원 탈퇴
탈퇴 후에는 아이디와 데이터는 복구할 수 없습니다.
그래도 탈퇴하시겠습니까?
☐ 안내사항을 모두 확인했습니다
탈퇴하기
웹 페이지 메시지
약관에 동의해 주세요
확인
체크를 누르지 않으면 위의 알림창이 뜨면서 탈퇴 되지 않음

4. 3 주요 로직 (회원탈퇴)

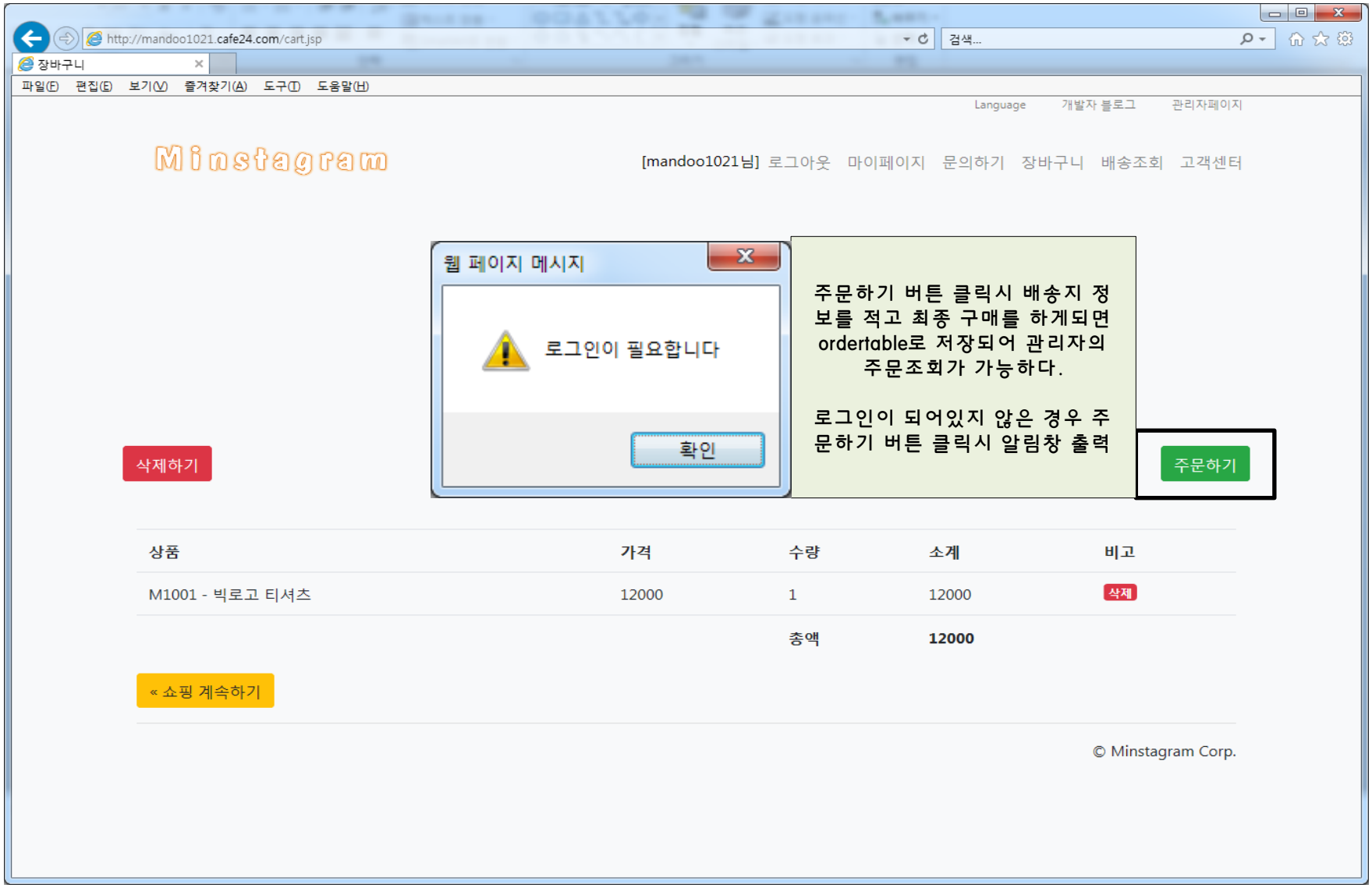
```
<script type="text/javascript">
function checkForm(Join){
    //체크박스 체크여부 확인 [하나]
    var chk1=document.deleteForm.deleteCheck.checked;

    if(!chk1){
        alert('약관에 동의해 주세요');
        return false;
    }
}
```

Form태그의 name을 받아와 .checked로 체크했는지 여부를 확인후 not조건으로 알림창 띄우고 false값 리턴으로 버튼이 실행되지 않게함

```
<sql:update dataSource="${dataSource}" var="resultSet">
   DELETE FROM member WHERE user_id = ?
   <sql:param value="<%=sessionId%>" />
</sql:update>

<c:if test="${resultSet>=1}">
    <c:import var="url" url="logoutMember.jsp" />
    <c:redirect url="resultMember.jsp" />
</c:if>
```

Delete 쿼리문으로 유저 조건 검색후 삭제

4. 3 주요 로직 (주문하기)


http://mandoo1021.cafe24.com/cart.jsp
장바구니
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
Language 개발자 블로그 관리자페이지
Minstagram
[mandoo1021님] 로그아웃 마이페이지 문의하기 장바구니 배송조회 고객센터
웹 페이지 메시지
로그인이 필요합니다
확인
주문하기 버튼 클릭시 배송지 정보를 적고 최종 구매를 하게되면 ordertable로 저장되어 관리자의 주문조회가 가능하다.
로그인이 되어있지 않은 경우 주문하기 버튼 클릭시 알림창 출력
삭제하기
주문하기
상품 가격 수량 소계 비고
M1001 - 빅로고 티셔츠 12000 1 12000 삭제
총액 12000
« 쇼핑 계속하기
© Minstagram Corp.

4. 3 주요 로직 (주문하기)

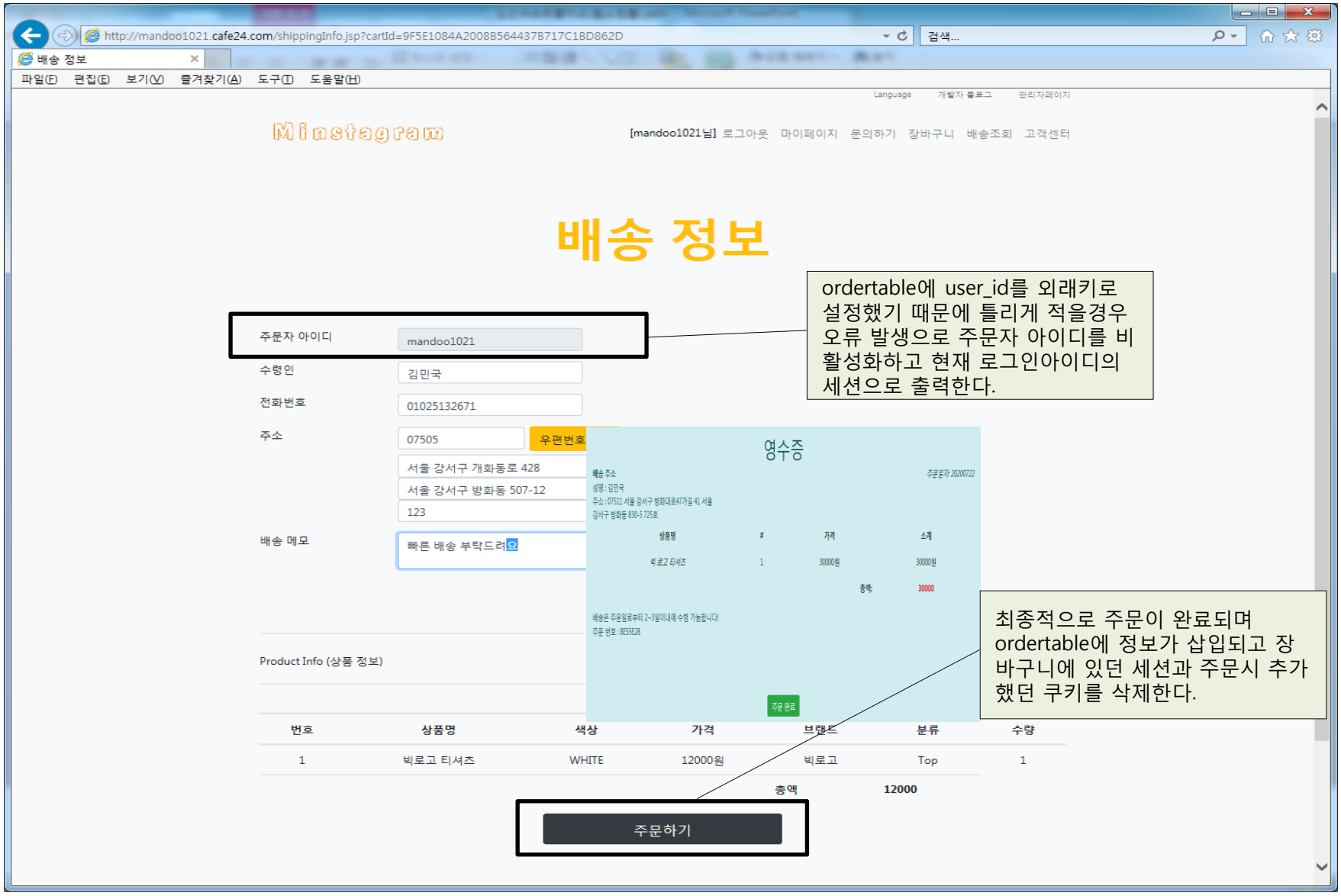
```
<script>
    function loginCheck(){
        <%
            Object obj = session.getAttribute("sessionId");
            if(obj == null){
        %>
                alert("로그인이 필요합니다");
                location.href = "http://localhost:9000/FinalTest/member/memberLogin.jsp"
                return false;
        <%
            } else {
        %>
                return true;
        <%
            }
        %>
    }
</script>
```

Session값을 가져와서 session이 null값이면 주문이 되지 않도록 처리한다.

```
int sum = 0;
int cnt = 0;
ArrayList<Clothes> cartList = (ArrayList<Clothes>) session.getAttribute("cartlist");
if(cartList == null) cartList = new ArrayList<Clothes>();
for(int i = 0; i < cartList.size(); i++){
    Clothes clothes = cartList.get(i);
    int total = clothes.getUnitPrice() * clothes.getQuantity();
    sum = sum + total;
    cnt += 1;
```

장바구니에 담기는 수량과 총액을 가져오기 위해서 가져올때마다 cnt 변수와 sum변수를 활용해서 출력한다.

4. 3 주요 로직 (주문하기 및 완료)



4. 3 주요 로직 (주문하기 및 완료)

```
ArrayList<Clothes> list = (ArrayList<Clothes>) session.getAttribute("cartlist");
if (list == null) {
    list = new ArrayList<Clothes>();
    session.setAttribute("cartlist", list);
}

int cnt = 0;
Clothes goodsQnt = new Clothes();
for (int i = 0; i < list.size(); i++) {
    goodsQnt = list.get(i);
    if (goodsQnt.getStyleNo().equals(id)) {
        cnt++;
        int orderQuantity = goodsQnt.getQuantity() + 1;
        goodsQnt.setQuantity(orderQuantity);
    }
}
```

- 세션 생성
상품 상세페이지에서 장바구니에 추가 하면 세션을 생성한다.

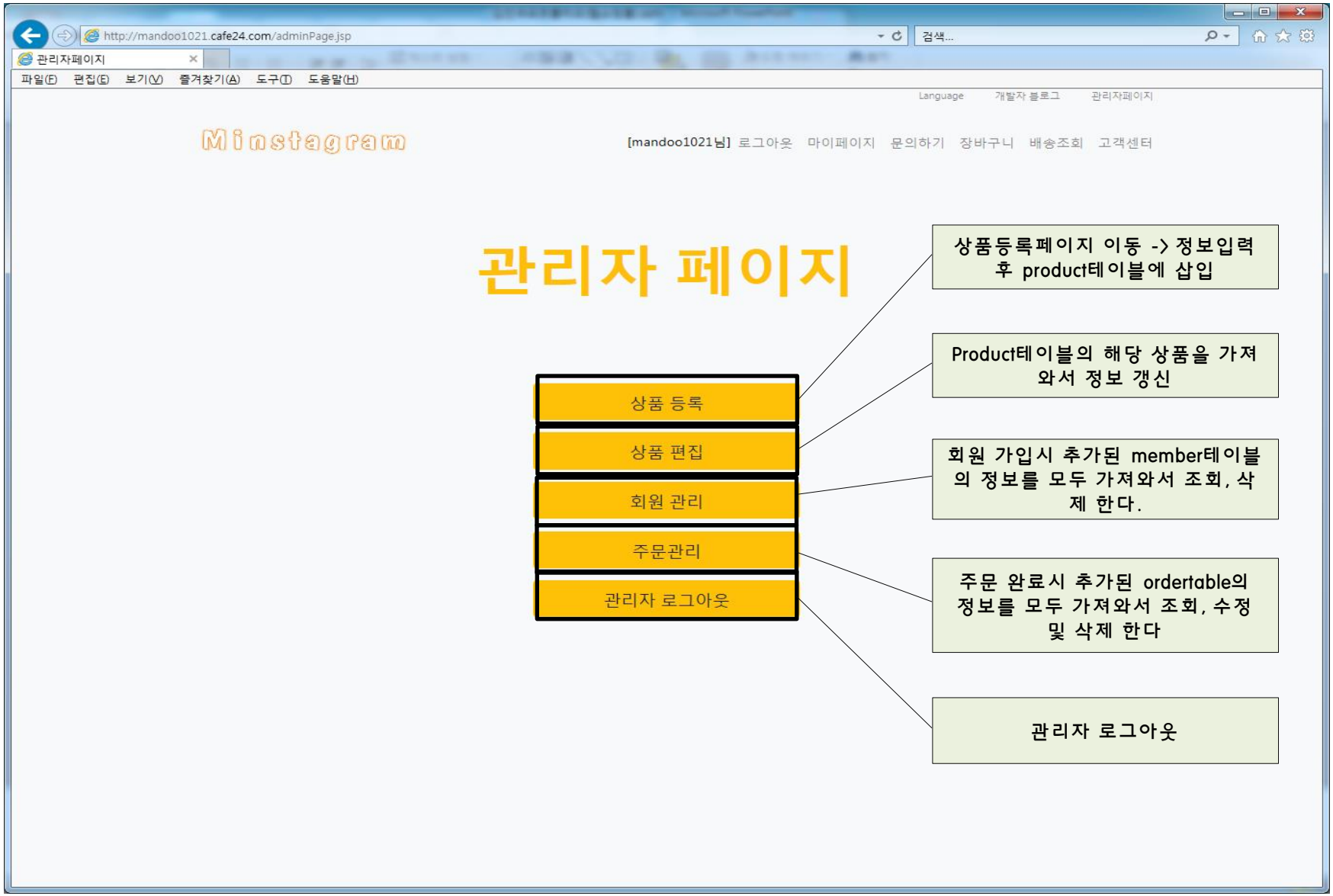
```
Cookie cartId = new Cookie("Shipping_cartId", URLEncoder.encode(request.getParameter("cartId"), "utf-8"));
Cookie u_id = new Cookie("Shipping_uId", URLEncoder.encode(sessionId, "utf-8"));
Cookie u_name = new Cookie("Shipping_name", URLEncoder.encode(request.getParameter("name"), "utf-8"));
Cookie date = new Cookie("Shipping_shippingDate", URLEncoder.encode(sdate, "utf-8"));
Cookie addressName = new Cookie("Shipping_addressName", URLEncoder.encode(address, "utf-8"));

cartId.setMaxAge(365 * 24 * 60 * 60); u_id.setMaxAge(365 * 24 * 60 * 60);
u_name.setMaxAge(365 * 24 * 60 * 60); date.setMaxAge(365 * 24 * 60 * 60);
addressName.setMaxAge(365 * 24 * 60 * 60);

response.addCookie(cartId); response.addCookie(u_id);
response.addCookie(u_name); response.addCookie(date);
response.addCookie(addressName);
```

- 쿠키 생성

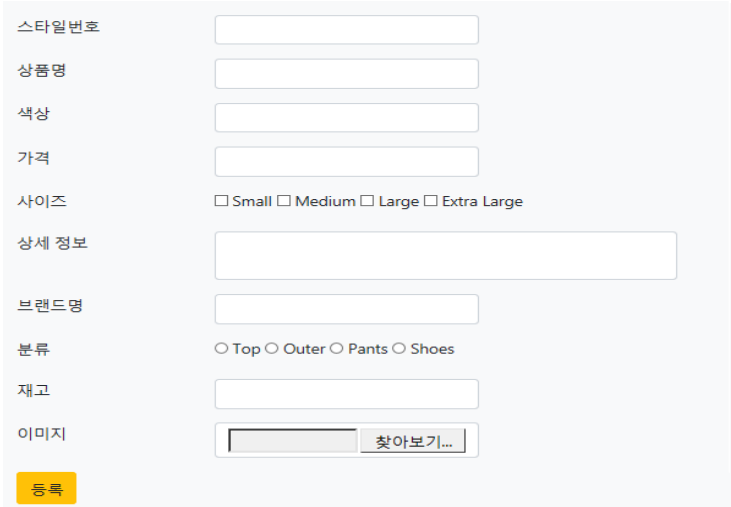
```
session.removeAttribute("cartlist");

for (int i = 0; i < cookies.length; i++) {
    Cookie thisCookie = cookies[i];
    String n = thisCookie.getName();

    if (n.equals("Shipping_cartId"))
        thisCookie.setMaxAge(0);
    if (n.equals("Shipping_uId"))
        thisCookie.setMaxAge(0);
    if (n.equals("Shipping_name"))
        thisCookie.setMaxAge(0);
    if (n.equals("Shipping_shippingDate"))
        thisCookie.setMaxAge(0);
    if (n.equals("Shipping_addressName"))
        thisCookie.setMaxAge(0);

    response.addCookie(thisCookie);
}
```

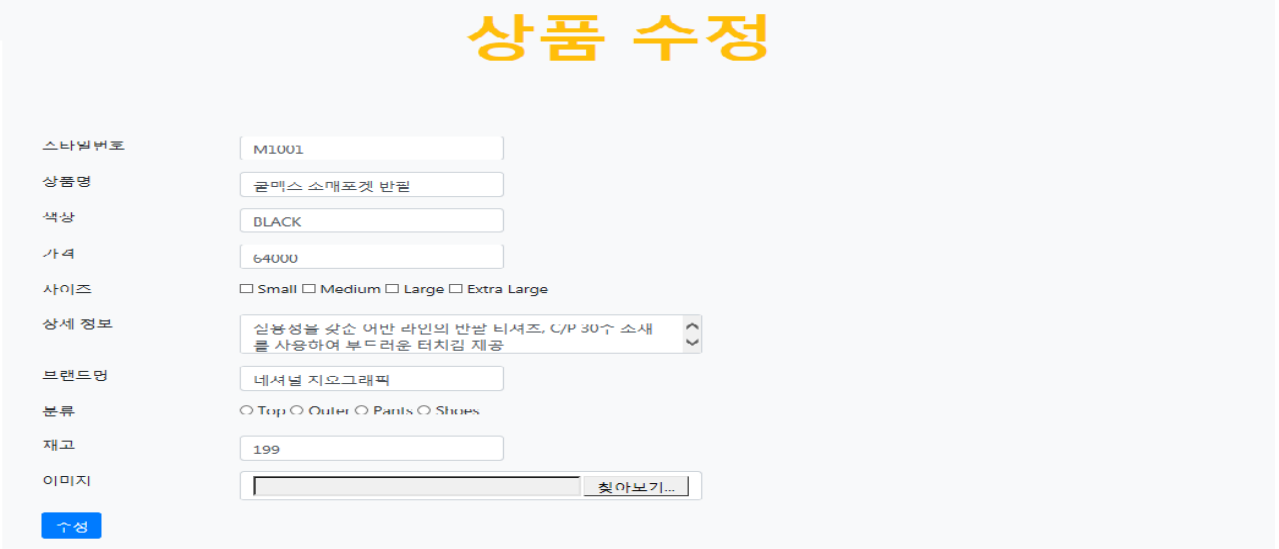
- 쿠키 삭제 및 세션 삭제
최종 주문 완료시 주문시 추가했던 쿠키의 유효기간을 0으로 설정해서 쿠키를 삭제하고 장바구니에 추가했던 세션도 함께 삭제시킨다

4. 3 주요 로직 (관리자 페이지)



4. 3 주요 로직 (관리자 페이지 – 상품 등록, 상품 편집)





4. 3 주요 로직 (관리자 페이지 – 상품 등록, 상품 편집)

```java
try {
    String url = "jdbc:mysql://localhost:3306/finaltestDB";
    String user = "root";
    String password = "1234";

    Class.forName("com.mysql.jdbc.Driver");
    conn = DriverManager.getConnection(url, user, password);

    String sql = "insert into product values(?,?,?,?,?,?,?,?,?,?)";
    pstmt = conn.prepareStatement(sql);
    pstmt.setString(1, styleno);
    pstmt.setString(2, c_name);
    pstmt.setString(3, color);
    pstmt.setInt(4, price);
    pstmt.setString(5, size);
    pstmt.setString(6, description);
    pstmt.setString(7, manufacturer);
    pstmt.setString(8, category);
    pstmt.setLong(9, stock);
    pstmt.setString(10, fileName);

    pstmt.executeUpdate();
```

- 상품 등록
파라미터값을 요청받아 product테이블에 삽입

```java
String sql = "select * from product where styleno = ?";
pstmt = conn.prepareStatement(sql);
pstmt.setString(1, styleno);
rs = pstmt.executeQuery();

if (rs.next()) {
    if (fileName != null) {
        sql = "UPDATE product SET c_name=?, unitPrice=?, size=?, description=?, color=?, category=?, unitsInStock=?, filename=? WHERE styl
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, c_name);
        pstmt.setInt(2, price);
        pstmt.setString(3, size);
        pstmt.setString(4, description);
        pstmt.setString(5, color);
        pstmt.setString(6, category);
        pstmt.setLong(7, stock);
        pstmt.setString(8, filename);
        pstmt.setString(9, styleno);
        pstmt.executeUpdate();
    } else {
        sql = "UPDATE product SET c_name=?, unitPrice=?, size=?, description=?, color=?, category=?, unitsInStock=? WHERE styleno=?";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, c_name);
        pstmt.setInt(2, price);
        pstmt.setString(3, size);
        pstmt.setString(4, description);
        pstmt.setString(5, color);
        pstmt.setString(6, category);
        pstmt.setLong(7, stock);
        pstmt.setString(8, styleno);
        pstmt.executeUpdate();
    }
}
```

- 상품 편집
파라미터값을 요청받아 product테이블에 삽입
If문으로 조건을 fileName에 조건을 줘서 상품 편집시에 파라미터로 받은 fileName값이 없을경우 fileName을 제외하고 추가

4. 3 주요 로직 (관리자 페이지 – 회원 관리, 주문 관리)

4. 3 주요 로직 (관리자 페이지 – 회원 관리, 주문 관리)

```jsp
<select name="classification" class="form-control mr-sm-2" style="margin-right:5px">
    <option value="total">전체</option>
    <option value="name">이름</option>
    <option value="birth">생년월일</option>
    <option value="email">이메일</option>
    <option value="phone">전화번호</option>
    <option value="address">주소</option>
</select>
<input class="form-control mr-sm-2" name="search" type="search"
    placeholder="Search" aria-label="의류 검색">
<button class="btn btn-warning" type="submit">검색</button>
```

```jsp
    } else if(search != null && classification.equals("name")) {
        String sql = "select * from member where user_name like ?";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, searchTotal);
        rs = pstmt.executeQuery();
        while(rs.next()) {
%>
    <tbody align="center">
        <tr>
            <td><%=rs.getString("user_id") %></td>
            <td><%=rs.getString("user_name") %></td>
            <td><%=rs.getString("user_gender") %></td>
            <td><%=rs.getString("user_birth") %></td>
            <td><%=rs.getString("user_email") %></td>
            <td><%=rs.getString("user_phone") %></td>
            <td><%=rs.getString("user_address") %></td>
            <td><%=rs.getString("resist_day") %></td>
            <td><a href="editMember_delete.jsp?id=<%=rs.getString("user_id") %>"
                onClick="deleteConfirm('<%=rs.getString("user_id")%>')"
                class="btn btn btn-primary btn-danger">삭제</a></td>
        </tr>
<%
```

- 분류 검색

If조건문으로 받은 value값에 따라 select문의 조건을 다르게 줘서 검색

```jsp
String orderId = request.getParameter("id");

PreparedStatement pstmt = null;
ResultSet rs = null;

String sql = "select * from ordertable";
pstmt = conn.prepareStatement(sql);
rs = pstmt.executeQuery();
ClothesRepository dao = ClothesRepository.getInstance();

ArrayList<Clothes> list = dao.getAllProducts();

if (rs.next()) {
    sql = "delete from ordertable where order_no = ?";
    pstmt = conn.prepareStatement(sql);
    pstmt.setString(1, orderId);
    pstmt.executeUpdate();
} else
    out.println("일치하는 주문내역이 없습니다");
```

- 주문 삭제

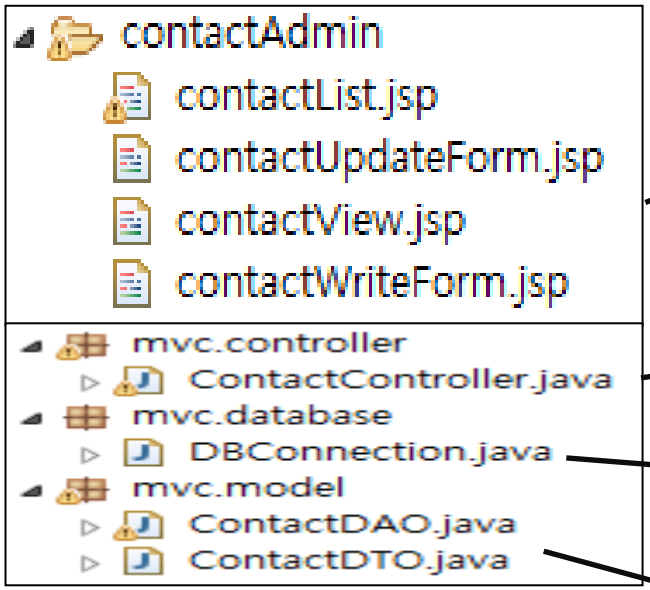
파라미터 값으로 주문번호를 요청받아 조건에 넣고 삭제

```jsp
<sql:update dataSource="${dataSource}" var="resultSet">
   DELETE FROM member WHERE user_id = ?
   <sql:param value="<%=id%>" />
</sql:update>
```

- 회원 삭제

파라미터 값으로 user_id를 요청받아 조건에 넣고 삭제

4. 3 주요 로직 (게시판)



View로 화면 구성은 리스트와 글 수정, 글 쓰기, 상세페이지로 구성

Controllor로 게시판의 요청을 처리

DBConnection을 분류하여 공통적으로 데이터베이스 사용

Model로 해당 테이블의 getter/setter를 처리하고 데이터베이스와 연결하여 쿼리문 전달

```
<servlet>
    <servlet-name>ContactController</servlet-name>
    <servlet-class>mvc.controller.ContactController</servlet-class>
</servlet>

<servlet-mapping>
    <servlet-name>ContactController</servlet-name>
    <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

<servlet> 요소로 서블릿 클래스를 등록해주고 서블릿 클래스를 매핑하기위해 .do로 패턴 설정

4. 3 주요 로직 (게시판)


문의하기
번호의 경우 기본키로 설정 해 놓아서 중복되지 않아야 하고 게시글의 번호를 간편하게 찾기 위해서 번호 생성시 auto_increment 사용하여 생성시 마다 증가하도록 설정
create table adminContact (
num int not null auto_increment,
전체 6건
번호
작성자
작성일
13
개인 문의
박형식
2020/07/22(13:53:33)
12
여섯번째
정형돈
2020/07/21(15:51:09)
11
다섯번째글
노홍철
2020/07/21(15:50:51)
9
세번째글
김민국
2020/07/21(14:46:08)
8
가격 문의
김민국
2020/07/21(10:33:51)
[1] [2]
«글쓰기
제목
Search
검색
게시판의 경우 5개의 게시글이 넘어가면 다음페이지로 넘어갈수 있도록 설정
© Minstagram Corp.

4. 3 주요 로직 (게시판)

```java
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String RequestURI = request.getRequestURI();
    String contextPath = request.getContextPath();
    // ex) http://localhost:9000/FinalTest/main.jsp 의 URI주소에서 앞부분을 제외하고
    // /main.jsp를 출력하기위해 substring()메소드 사용
    String command = RequestURI.substring(contextPath.length());

    response.setContentType("text/html; charset=utf-8");
    request.setCharacterEncoding("utf-8");

    if (command.equals("/ContactListAction.do")) { // 등록된 글 목록 페이지 출력하기
        requestContactList(request);
        RequestDispatcher rd = request.getRequestDispatcher("./contactAdmin/contactList.jsp");
        rd.forward(request, response);
```

View에서 command값을 요청받아 if문이 일치할경우 메소드를 실행 후 처리된 결과를 보여줄 응답페이지로 보낸다.
이동해도 처음에 요청된 URL을 유지하기위해 포워딩방식을 사용

```java
static final int LISTCOUNT = 5;

// 등록된 글 목록 가져오기
private void requestContactList(HttpServletRequest request) {
    ContactDAO dao = ContactDAO.getInstance();
    List<ContactDTO> contactlist = new ArrayList<ContactDTO>();

    int pageNum = 1;
    int limit = LISTCOUNT;

    if(request.getParameter("pageNum") != null)
        pageNum = Integer.parseInt(request.getParameter("pageNum"));

    String items = request.getParameter("items");
    String text = request.getParameter("text");

    int total_record = dao.getListCount(items, text);

    contactlist = dao.getContactList(pageNum, limit, items, text);
    int total_page;

    if (total_record % limit == 0) {
        total_page = total_record/limit;
        // floor() 메소드 : 지정된 숫자보다 작거나 같은 최대 정수 값을 반환
        Math.floor(total_page);
    } else {
        total_page = total_record/limit;
        Math.floor(total_page);
        total_page = total_page + 1;
    }
    request.setAttribute("pageNum", pageNum);
    request.setAttribute("total_page", total_page);
    request.setAttribute("total_record", total_record);
    request.setAttribute("contactlist", contactlist);
}
```

요청 받은 값으로 메소드 처리

→

```java
// adminContact 테이블의 레코드 가져오기
public ArrayList<ContactDTO> getContactList(int page, int limit, String items, String text){
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    int total_record = getListCount(items, text);
    int start = (page - 1) * limit;
    int index = start + 1;

    String sql;

    if (items == null && text == null)
        sql = "select * from adminContact order by num desc";
    else
        sql = "select * from adminContact where " + items + " like '%" + text + "%' order by num desc";

    ArrayList<ContactDTO> list = new ArrayList<ContactDTO>();

    try {
        conn = DBConnection.getConnection();
        pstmt = conn.prepareStatement(sql);
        rs = pstmt.executeQuery();

        while (rs.absolute(index)) {
            ContactDTO contact = new ContactDTO();
            contact.setNum(rs.getInt("num"));
            contact.setUser_id(rs.getString("user_id"));
            contact.setUser_name(rs.getString("user_name"));
            contact.setUser_phone(rs.getString("user_phone"));
            contact.setSubject(rs.getString("subject"));
            contact.setContent(rs.getString("content"));
            contact.setRegist_day(rs.getString("regist_day"));
            contact.setIp(rs.getString("ip"));
            list.add(contact);

            if (index < (start + limit) && index <= total_record) index++;
            else break;
        }
        return list;
    } catch (Exception ex) {
```

4. 3 주요 로직 (게시판 – 문의 글쓰기)

4. 3 주요 로직 (게시판 – 문의 글쓰기)

```jsp
<td width="100%" align="right">
    <a href="#" onClick="checkForm(); return false;" style="margin-bottom:100px;" class="btn btn-primary">&laquo;글쓰기</a>
</td>
```

```jsp
<script type="text/javascript">
    function checkForm() {
        if (${sessionId==null}) {
            alert("로그인 해주세요.");
            return false;
        }
        location.href = "./ContactWriteForm.do?id=<%=sessionId%>"
    }
</script>
```

```java
} else if (command.equals("/ContactWriteForm.do")) { // 글 등록 페이지 출력하기
    requestLoginName(request);
    RequestDispatcher rd = request.getRequestDispatcher("./contactAdmin/contactWriteForm.jsp");
    rd.forward(request, response);
```

```java
// 인증된 사용자명 가져오기
private void requestLoginName(HttpServletRequest request) {
    String id = request.getParameter("id");
    ContactDAO dao = ContactDAO.getInstance();
    String name = dao.getLoginNameById(id);
    request.setAttribute("name", name);
}
```

```java
// contactAdmin 테이블에서 인증된 id의 사용자명 가져오기
public String getLoginNameById(String id) {
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    String name = null;
    String sql = "select * from member where user_id = ?";

    try {
        conn = DBConnection.getConnection();
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, id);
        rs = pstmt.executeQuery();

        if (rs.next()) {
            name = rs.getString("user_name");
        }
        return name;
    } catch (Exception ex) {
        System.out.println("getContactByNum() 에러 : " + ex);
    } finally {
        try {
            if(rs != null) rs.close();
            if(pstmt != null) pstmt.close();
            if(conn != null) conn.close();
        } catch (Exception ex) {
            throw new RuntimeException(ex.getMessage());
        }
    }
    return null;
}
```

위의 순서대로 Controller에서 뷰를 선택한 후 사용자의 동작을 요청받아 처리를 하는데 글을 등록하기 위해서는 요청값을 보낼때 현제 로그인 세션의 아이디를 같이 보내서 member 테이블과 일치한 후 return을 해서 일치하면 글을 등록하는 절차

4. 3 주요 로직 (게시판 – 문의 글쓰기)

4. 3 주요 로직 (게시판 – 수정 하기)

```jsp
<form name="newWrite" action="./ContactUpdateAction.do?num=<%=notice.getNum()%>"
    class="form-horizontal" method="post" onsubmit="return checkForm()">
    <input name="id" type="hidden" class="form-control"
<input type="submit" class="btn btn-primary " value="수정">
```

```
} else if (command.equals("/ContactUpdateAction.do")) { // 수정 하기
    requestContactUpdate(request);
    RequestDispatcher rd = request.getRequestDispatcher("/ContactListAction.do");
    rd.forward(request, response);
}
```

```
// 선택된 글 내용 수정하기
private void requestContactUpdate(HttpServletRequest request) {
    int num = Integer.parseInt(request.getParameter("num"));

    ContactDAO dao = ContactDAO.getInstance();
    ContactDTO contact = new ContactDTO();
    contact.setNum(num);
    contact.setUser_name(request.getParameter("name"));
    contact.setUser_phone(request.getParameter("phone"));
    contact.setSubject(request.getParameter("subject"));
    contact.setContent(request.getParameter("content"));

    java.text.SimpleDateFormat formatter = new java.text.SimpleDateFormat("yyyy/MM/dd(HH:mm:ss)");
    String regist_day = formatter.format(new java.util.Date());

    contact.setRegist_day(regist_day);
    contact.setIp(request.getRemoteAddr());

    dao.updateContact(contact);
}
```

```
// 선택된 글 내용 수정하기
public void updateContact(ContactDTO contact) {
    Connection conn = null;
    PreparedStatement pstmt = null;

    try {
        String sql = "update adminContact set user_name=?, user_phone=?, subject=?, content=? where num=?";

        conn = DBConnection.getConnection();
        pstmt = conn.prepareStatement(sql);

        // 여러 개의 쿼리 문장이 하나의 작업으로 수행되어야 할 경우에 각각의 문장이
        // 작동 못하게 해야하기 때문에 setAutoCommit을 false로 해놓는다
        conn.setAutoCommit(false);

        pstmt.setString(1, contact.getUser_name());
        pstmt.setString(2, contact.getUser_phone());
        pstmt.setString(3, contact.getSubject());
        pstmt.setString(4, contact.getContent());
        pstmt.setInt(5, contact.getNum());
        pstmt.executeUpdate();

        conn.commit();
    } catch (Exception ex) {
        System.out.println("updateContact() 에러 : " + ex);
    } finally {
        try {
            if(pstmt != null) pstmt.close();
            if(conn != null) conn.close();
        } catch (Exception ex) {
            throw new RuntimeException(ex.getMessage());
        }
    }
}
```

4. 3 주요 로직 (게시판 – 문의하기 글 상세페이지)

4. 3 주요 로직 (게시판 – 상세페이지 수정, 삭제 버튼)

```
action="./ContactUpdateFormAction.do?num=<%=notice.getNum()%>&pageNum=<%=nowpage%>"
<c:if test="${sessionId==userId}">
    <p>
    <a href="./ContactDeleteAction.do?num=<%=notice.getNum()%>&pageNum=<%=nowpage%>"
        onclick="return deleteCheck();" class="btn btn-danger"> 삭제</a>
    <input type="submit" class="btn btn-success" value="수정 ">
</c:if>
```

수정

삭제

```
} else if (command.equals("/ContactUpdateFormAction.do")) { // 수정화면으로 기존 글 가져오기
    requestContactUpdateView(request);
    RequestDispatcher rd = request.getRequestDispatcher("/ContactUpdateForm.do");
    rd.forward(request, response);
} else if (command.equals("/ContactUpdateForm.do")) { // 수정화면으로 기존 글 가져오기
    RequestDispatcher rd = request.getRequestDispatcher("./contactAdmin/contactUpdateForm.jsp");
    rd.forward(request, response);
}
```

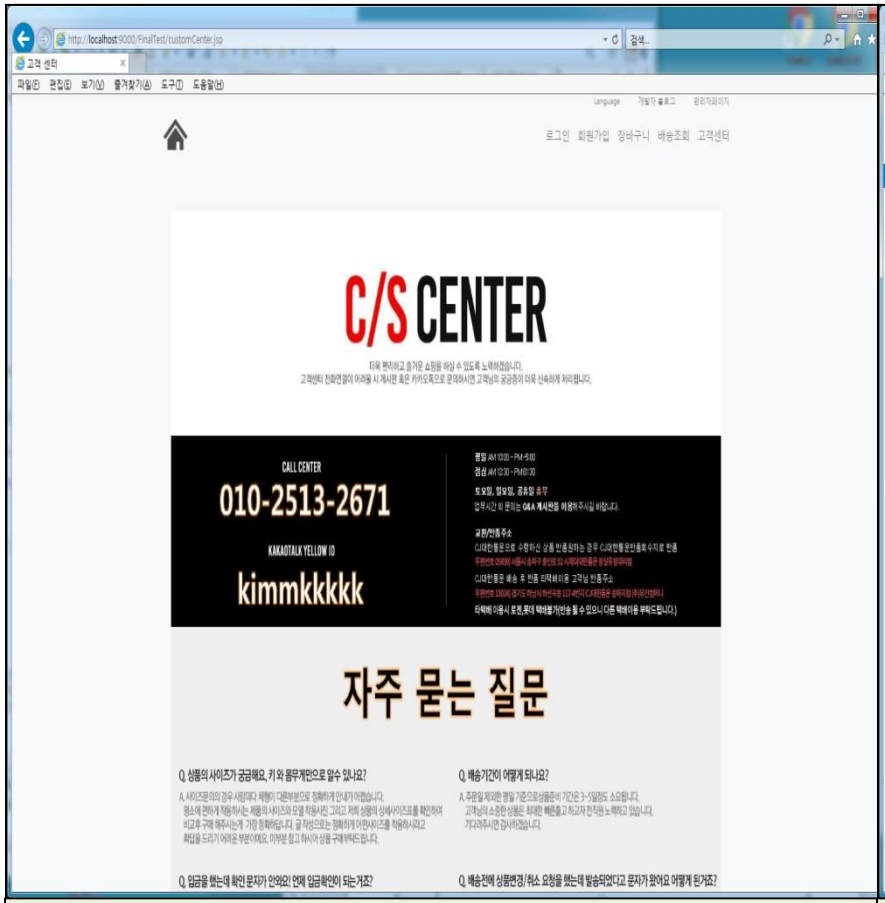
```
// 수정 페이지
private void requestContactUpdateView(HttpServletRequest request) {
    ContactDAO dao = ContactDAO.getInstance();
    int num = Integer.parseInt(request.getParameter("num"));
    int pageNum = Integer.parseInt(request.getParameter("pageNum"));

    ContactDTO contact = new ContactDTO();
    contact = dao.getUpdateByNum(num, pageNum);

    request.setAttribute("contact", contact);

}
```

```
} else if (command.equals("/ContactDeleteAction.do")) { // 선택된 글 삭제하기
    requestContactDelete(request);
    RequestDispatcher rd = request.getRequestDispatcher("/ContactListAction.do");
    rd.forward(request, response);
```

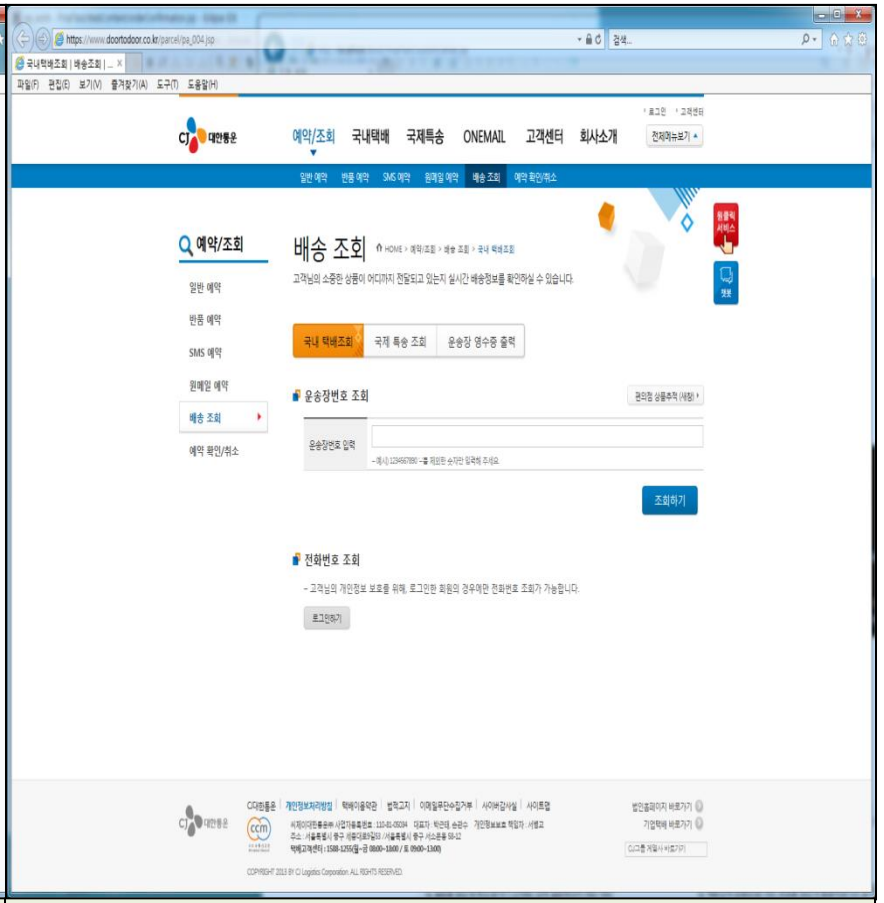
```
// 선택한 글 삭제하기
private void requestContactDelete(HttpServletRequest request) {
    int num = Integer.parseInt(request.getParameter("num"));
    int pageNum = Integer.parseInt(request.getParameter("pageNum"));

    ContactDAO dao = ContactDAO.getInstance();
    dao.deleteContact(num);

}
```

```
// 선택된 글 삭제하기
public void deleteContact(int num) {
    Connection conn = null;
    PreparedStatement pstmt = null;

    String sql = "delete from adminContact where num=?";

    try {
        conn = DBConnection.getConnection();
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, num);
        pstmt.executeUpdate();
    } catch (Exception ex) {
        System.out.println("deleteContact() 에러 : " + ex);
    } finally {
```

4. 4 기타 로직 (고객센터, 배송조회)



- 고객센터 화면



- 배송 조회
클릭시 새창으로 대한통운 홈페이지 연결

Thank you.