

# Neocom

버전	2.0
작성일	2021-09-16
소속	JHTA
작성자	오문형

# - 목차 -

- 프로젝트 개요
- 사용 언어/툴/프레임워크/DB
- Reference
- 기본 인터페이스 구상도
- 기능요구분석
- Flow Chart(순서도)
- 개발 예정 기간

# 프로젝트 개요

- 컴퓨터 부품 온라인 거래 웹사이트
- 상품의 목록에 상품을 추가하고, 업데이트하고, 삭제 할 수 있게 서비스를 구성(CRUD)
- Model 2 MVC 설계 모델로 개발
- 사용자 친화적인 웹 애플리케이션 개발을 목표로 함  
(SNS연동, 고객센터, 커뮤니티 등)
- 관리자 입장에서도 관리하기 편한 쇼핑몰

# 프로젝트 개요

## 주제 선정 의도

- 수강 과정동안 공부했던 것들에 대해 정확히 이해하고, 활용해 보는 것
- 저번 세미 프로젝트 때, 제대로 이해를 하지 못하고 넘어간 것들이 많았고, 다른 프로그램, 기술을 사용해보고 싶었지만 부족한 시간으로 인해 사용해보지 못했던 아쉬움
- 방대한 양의 DB를 다뤄볼 필요가 있다고 생각함

# 사용 언어/툴/프레임워크

- 사용 언어
  - Java
  - JavaScript(jsp,jQuery)
- 프레임워크
  - Spring boot
  - Bootstrap
- 버전 관리 시스템
  - Git
- 빌드 툴
  - Maven
- DBMS
  - MySql

# Reference

Newegg.com

The screenshot shows the Newegg.com homepage. The left sidebar lists categories like Components, Computer Systems, Electronics, Gaming, Networking, Office Solutions, Software & Services, Automotive, Home & Tools, Health & Sports, Apparel & Accessories, Toys & Drones, and Flowers & Gifts. The main content area features a search bar with results for 'drones', 'ryzen 5 5600x', and 'gaming desk'. A 'Welcome' message with a sign-in link is at the top right. Promotional banners include one for AMD Ryzen processors and another for Western Digital Blue drives.

## -장점-

- 방대한 컴퓨터 관련 상품 목록

## -단점-

- 불편한 고객센터
- 낮은 신뢰도의 검색 기능

# Reference

코잇컴



The screenshot displays the COITCOM website's user interface. At the top, there's a banner for 'PC 월간 견적' (Monthly PC Quotation) with a call-to-action '당신의 PC 견적을 책임지는 코잇컴'. Below the banner are standard website navigation links like '로그인', '회원가입', 'マイページ', '상담센터', '장바구니', '주문배송 조회'. The main content area includes:

- 커뮤니티**: A section for community interaction.
- 온라인 견적 서비스**: A service for generating PC quotations. It features a large banner with '100% 정품 및 서제품 사용, 완벽 호환성 검토, 꼼꼼한 조립과 포장까지!!' and several filters for '전용가 추천상품' (CPU, 메인보드, RAM, 케이스, 파워), 'MD 추천상품' (e.g., AMD 라이젠5-4세대 5600X (버미어) (멀티액)), and '구매 후기' (e.g., [후기] 총리아 브랜드, [후기] 조립컴 헤파 구매후기입니다).
- 온라인 PC 견적**: A tool for generating PC quotations with a list of recommended components and their prices.
- 구매 후기**: A section displaying reviews from users, such as [구매후기] 총리아 브랜드 4★ and [구매후기] 코잇컴에서 안전하게 구매하세요 5★.
- 네이버 쇼핑 후기**: A section displaying reviews from Naver Shopping, such as [네이버 쇼핑 후기] 총리아 브랜드 5★ and [네이버 쇼핑 후기] 코잇컴에서 안전하게 구매하세요 5★.
- 컴퓨터 부품**: A section for selecting computer parts. It includes a search bar ('제조사 선택'), sorting options ('판매가순으로 보기', '신규상품순으로 보기', '높은가격순으로 보기', '낮은가격순으로 보기', '이벤트순으로 보기', '종결상품 제외하고 보기'), and dropdown menus for 'CPU', '메인보드', 'RAM', etc.
- 선택 상품**: A section showing specific product details for selected parts, such as an AMD Ryzen 5 4600G (Vermeer) (Multiblock) CPU at 337,100 won.

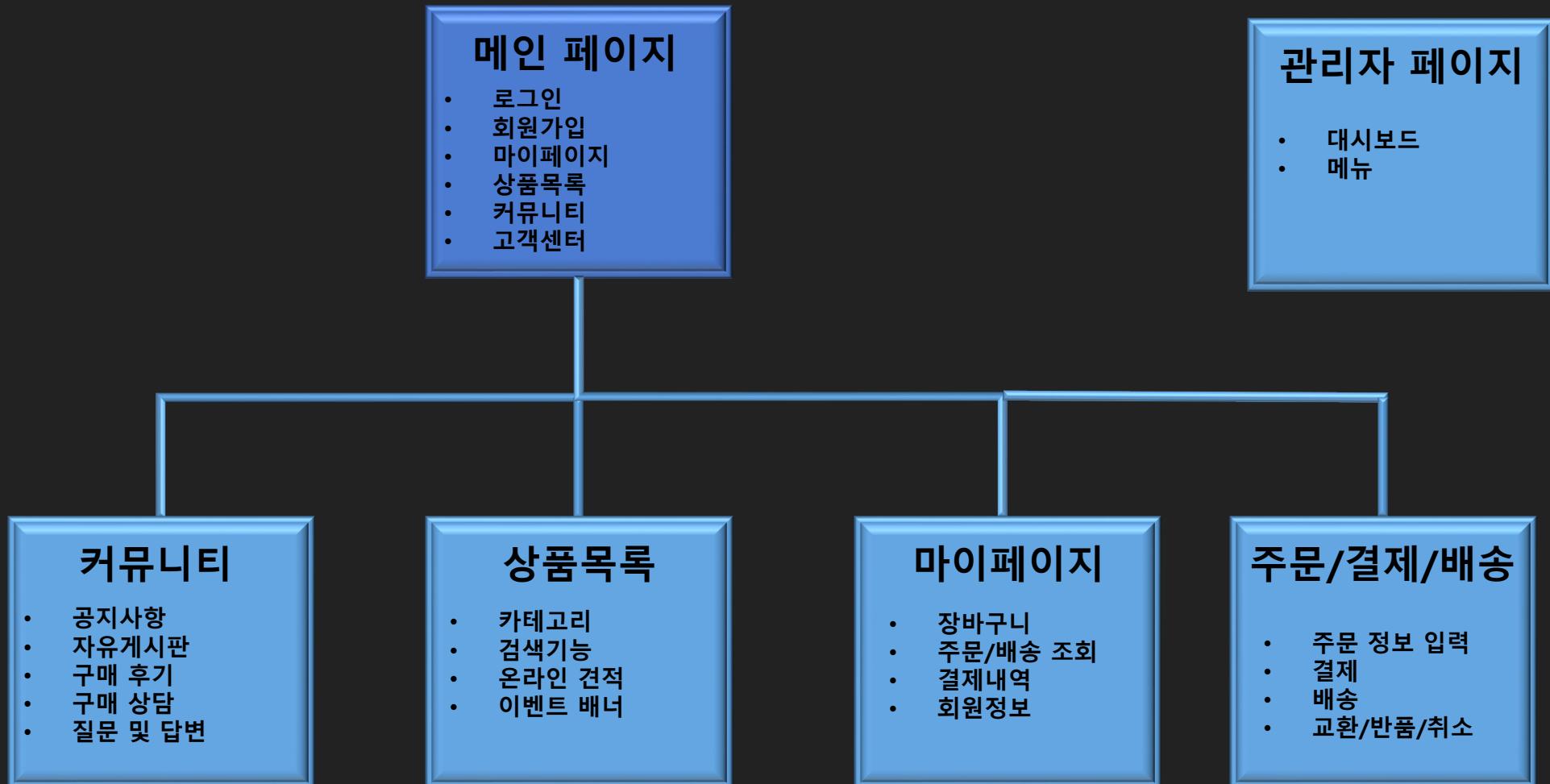
## -장점-

- 온라인 견적 기능의 호환성 체크
- 다양한 커뮤니티

## -단점-

- 복잡한 인터페이스
- 너무 많은 광고 배너

# 기본 인터페이스 구상도 (전체 구조)



# 메인페이지

The screenshot shows the Neocom homepage. At the top, there is a navigation bar with links for Home, Shop, Service, Community, and About Us. Below the navigation bar, there is a search bar and a user account section with icons for admin, logout, and mypage. A blue box highlights the mypage icon. The main content area features a large image of a JONSBO CR-1000 AUTO RGB (WHITE) cooler with two RGB-lit fans. Below the image, the brand name "JONSBO" is displayed, followed by the product name "JONSBO CR-1000 AUTO RGB (WHITE)" and the price "판매가 23,400 원". A blue button labeled "제품 보러가기" (View Product) is visible. At the bottom of the page, there is a section titled "상품" (Products) featuring several product cards for different CPU and GPU components.

```
<div class="toolbar-inner">
<sec:authorize access="isAnonymous()">
    <div class="toolbar-item">
        <a href="${pageContext.request.contextPath }/account/join!></a>
        <i class="icon-pocket"></i><span class="text-label">회원가입</span></div>
    </div>
</sec:authorize>

<sec:authorize access="isAuthenticated()">
    <div class="toolbar-item">
        <a href="${pageContext.request.contextPath }/account/login"></a>
        <i class="icon-user"></i><span class="text-label">로그인</span></div>
    </div>
</sec:authorize>

<div class="toolbar-item">
    <a href="${pageContext.request.contextPath }/account/logout"></a>
    <i class="icon-slash"></i><span class="text-label">로그아웃</span></div>
</div>
<div class="toolbar-item">
    <a href="${pageContext.request.contextPath }/account/mypage_order"></a>
    <i class="icon-box"></i><span class="text-label">마이페이지</span></div>
</div>
</sec:authorize>
```

Spring security로  
로그인 기능 구현

```
<section class="container padding-top-4x padding-bottom-3x mb-2">
<h2 class="h3 pb-2 text-center">({ ${empty} }) 상품
```

DB에서 상품 데이터를 받아와  
페이지 처리해서 출력

```
@RequestMapping(value = "/")
public ModelAndView frontend(@RequestParam(value = "pageNum", defaultValue = "1") int pageNum, String field,
                           String keyword) {
    HashMap<String, Object> map = new HashMap<String, Object>();
    map.put("field", field);
    map.put("keyword", keyword);

    int totalRowCount = service.getCount(map); // 전체 상품의 개수
    PageUtil pu = new PageUtil(pageNum, 10, 10, totalRowCount);

    int startRow = pu.getStartRow();
    int endRow = pu.getEndRow();

    map.put("startRow", startRow);
    map.put("endRow", endRow);
    List<HashMap<String, Object>> list = service.list(map); // 상품 리스트

    ModelAndView mv = new ModelAndView("/frontend/index");
    mv.addObject("list", list);
    mv.addObject("pu", pu);
    mv.addObject("field", field);
    mv.addObject("keyword", keyword);

    return mv;
}
```

IndexController.java

# 게시판(공지사항)

공지사항

Home > Community > Notice Board

글번호	제목	작성자	등록일시	조회수
53	공지	admin	2021-08-27 16:42:25	9
52	작성 테스트	admin	2021-08-27 13:58:43	10
51	공지 테스트	관리자	2021-08-27 12:28:16	6
50	공지사항 테스트	관리자	2021-08-26 05:43:16	10

< 이전 1 다음 >

```

<sec:authorize access="hasAnyRole('ADMIN', 'MANAGER')">
    <div class="ml-md-auto" style="margin-right:20px;">
        <button type="button" id="insertBtn"
            class="form-control btn-sm btn-outline-info"
            onclick="location.href='${pageContext.request.contextPath}/admin/community/board_list'"
            공지사항 관리 <i class="icon-arrow-right-circle"></i>
        </button>
    </div>
</sec:authorize>

```

Noticeboard\_list.jsp

```


| 글번호 | 제목       | 작성자   | 등록일시                | 조회수 |
|-----|----------|-------|---------------------|-----|
| 53  | 공지       | admin | 2021-08-27 16:42:25 | 9   |
| 52  | 작성 테스트   | admin | 2021-08-27 13:58:43 | 10  |
| 51  | 공지 테스트   | 관리자   | 2021-08-27 12:28:16 | 6   |
| 50  | 공지사항 테스트 | 관리자   | 2021-08-26 05:43:16 | 10  |



< 이전 1 다음 >


```

Spring security를 이용해서 관리자만 수정 가능하게 구현

공지사항 게시판 기본적인 CRUD 기능 구현

NoticeBoardController.java

```

@Controller
public class NoticeBoardController {
    @Autowired
    NoticeBoardService service;

    //공지사항 리스트 조회 메소드
    @RequestMapping(value = "/community/noticeboard/list")
    public String noticeboard_list(@ModelAttribute("page") Model model, Map map) {
        int totalRowCount = service.getRowCount(map); // 전체 글의 수
        PageUtil pu = new PageUtil(totalRowCount, 10, 10, totalRowCount);
        int startRow = pu.getStartRow();
        int endRow = pu.getEndRow();
        map.put("startRow", startRow);
        map.put("endRow", endRow);

        model.addAttribute("list", service.list(map));
        model.addAttribute("page", pu);
        model.addAttribute("startRow", startRow);
        model.addAttribute("endRow", endRow);
        model.addAttribute("keyword", keyword);
        model.addAttribute("rowCount", rowCount);
        return "frontend/community/noticeboard_list";
    }

    //공지사항 상세 페이지 이동
    @RequestMapping(value = "/community/noticeboard_detail", method = RequestMethod.GET)
    public String noticeboard_detail(@ModelAttribute("page") Model model, Map map) {
        Map map = service.detail(board_no);
        service.setNotice_board_no(); // 서비스에서 번호 설정
        NoticeBoard prev = service.prev("board_no");
        NoticeBoard next = service.next("board_no");
        model.addAttribute("prev", prev);
        model.addAttribute("next", next);
        System.out.println(prev + " " + next);
        return "frontend/community/noticeboard_detail";
    }
}

```

# 게시판(QNA)

문의게시판을  
계층형 게시판으로 구현

The screenshot shows a list of posts on a Q&A board. The columns are: 글번호 (Post ID), 제목 (Title), 작성자 (Author), 등록일시 (Registration Date), and 조회수 (View Count). Below the table, there's a detailed view of the first post, which includes the author's nickname, registration date, and a note that it's a reply. There are also buttons for deleting or updating the post.

QnaBoard\_list.jsp

```
<c:forEach var="vo" items="${list }">
  <tr>
    <c:choose>
      <c:when test="${vo.qna_show == 1 }">
        <td>
          <td colspan="4" class="text-left">
            <span style="color:gray;">
              <:if test="${vo.qna_group_depth >0 }">
                <c:forEach var="l" begin="1" end="${vo.qna_group_depth }">
                  &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
                </c:forEach>
                [Re]&nbsp;
              </:if>
            <:if test="${vo.qna_group_depth >0 }">
              작성자에 의해 삭제된 글입니다.
            </:if>
          </span>
        </td>
      </c:when>
      <c:otherwise>
        <td>
          <:if test="${vo.qna_group_depth == 0 }">
            ${vo.qna_board_no }
          </:if>
        </td>
        <td class="text-left">
          <a href="javascript:clickTitle('${vo.qna_board_no }','${vo.qna_secret_chk }');">
            <:if test="${vo.qna_group_depth >0 }">
              <c:forEach var="l" begin="1" end="${vo.qna_group_depth }">
                &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
              </c:forEach>
              [Re]&nbsp;
            </:if>
            ${vo.qna_title }
            <:if test="${vo.qna_secret_chk==true }">
              
            </:if>
          </a>
        </td>
        <td>${vo.Nickname}</td>
        <td><fmt:parseDate value="${vo.qna_regdate }" var="qna_regdate" pattern="yyyy-MM-dd'T'HH:mm:ss" /><fmt:formatDate value="${qna_regdate }" pattern="yyyy-MM-dd HH:mm:ss" /></td>
        <td>${vo.qna_hit }</td>
      </c:otherwise>
    </c:choose>
  </tr>
</c:forEach>
```

삭제했을 때, 원글과 답글  
이 엮어있을 때 구분지어  
주기 위해 DB에서  
삭제하지 않고  
업데이트

```
QnaBoardVo.java
@AllArgsConstructor
@NoArgsConstructor
@Data
public class QnABoardVo {
  private int qna_board_no;
  private int mem_no;
  private String qna_password;
  private int qna_secret_chk;
  private String qna_title;
  private String qna_content;
  private Timestamp qna_regdate;
  private int qna_hit;
  private int qna_group_no;
  private int qna_group_order;
  private int qna_group_depth;
  private int qna_status;
  private int qna_show;
  private String Nickname;
}
```

# 게시판(자유게시판)

파일 업로드를 위해 외부  
저장소 경로를 지정해 파일  
업로드 구현

The screenshot shows a table with columns: 글번호 (7), 작성자 (유저), 작성일 (2021-09-01 09:01:16), 제목 (파일 추가 테스트), 조회수 (5). Below the table, there is a preview area containing two image thumbnails: KakaoTalk\_20210725\_013916066\_03.jpg and KakaoTalk\_20210725\_013916066\_01.jpg. Navigation buttons like 이전글, 다음글, and 답글달기 are visible.

```
<td>
    <c:forEach var="filemap" items="${filemap}">
        <a href="${pageContext.request.contextPath }/community/freeboard_filedownload?file_num=${filemap.file_num}">
            ${filemap.file_name_org }
        </a>
    </c:forEach>
</td>
```

문의게시판과 똑같이  
계층형 게시판으로 구현  
파일업로드 기능 추가

```
# File Upload (location**)
spring.servlet.multipart.location=C:/final_project/uploads/
spring.servlet.multipart.max-file-size=50MB
spring.servlet.multipart.max-request-size=60MB
spring.servlet.multipart.file-size-threshold=20MB
spring.servlet.multipart.enabled=true
```

## FreeBoardController.java

```
// 자유게시판 상세 페이지 이동
@RequestMapping(value = "/community/freeboard_detail", method = RequestMethod.GET)
public String freeboard_detail(int free_board_no, Model model, Authentication auth) {
    CustomUserDetails cud = (CustomUserDetails) auth.getPrincipal();
    MemberVo mvo = cud.getMemberVo();

    HashMap<String, Object> map = service.detail(free_board_no);
    service.cntHit(free_board_no); // 조회수 증가 처리
    model.addAttribute("map", map);
    model.addAttribute("mvo", mvo);

    FreeBoardVo prev = service.prev(free_board_no);
    FreeBoardVo next = service.next(free_board_no);
    model.addAttribute("prev", prev);
    model.addAttribute("next", next);

    List<HashMap<String, Object>> filemap = f_service fileList(free_board_no); // 첨부파일 리스트 출력
    model.addAttribute("filemap", filemap);

    return "frontend/community/freeboard_detail";
}
```

## FreeBoardController.java

```
// 자유게시판 글 작성
@RequestMapping(value = "/community/freeboard_insert", method = RequestMethod.POST)
public String freeboard_insertOk(Model model, FreeBoardVo vo, Authentication auth,
    MultipartHttpServletRequest req) {
    CustomUserDetails cud = (CustomUserDetails) auth.getPrincipal();
    MemberVo mvo = cud.getMemberVo();

    int mem_no = mvo.getMem_no();
    vo.setMem_no(mem_no);

    String path = uploadFilePath + "\\" + "board_file"; // 업로드 파일 절대경로 구하기
    File dir = new File(path);
    if (!dir.isDirectory()) {
        dir.mkdir();
    }

    String file_name_save = System.currentTimeMillis() + file_name_org;
```

# 관리자 페이지(DASHBOARD)

Neocom

DASHBOARD

2021-08-23 - 2021-09-01

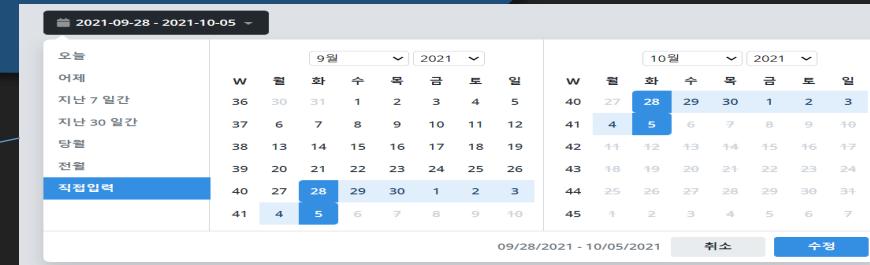
총 매출  
₩9,458,800

결제 완료 비율  
14.61%

총 회원 수  
24

판매상품 순위

- AMD 라이젠3 PRO 4350G (르누아르)(멀티팩)
- ASRock Phantom Gaming D 라데온 RX 6900 XT OC D6 16GB 에즈윈
- 케이엔시스템 KLcom BLACK DIAMOND 고급형 HDMI v2.1 케이블 (KL81, 0.3m)
- TeamGroup T-Force DDR4-3200 CL16-20-20 DARK 7 GREY 패키지 (32GB(16Gx2))
- G.SKILL DDR4 32G PC4-28800 CL16 TRIDENT Z NEO C (16Gx2)
- 이엠텍 지포스 RTX 3070 BLACK EDITION OC D6 8GB LHR
- darkFlash DM4 ARGB M.2 SSD 방열판



Datepicker 라이브러리 사용해서 구현

Admin/index.jsp

```
var sendDates = function (startDate, endDate) {
    $.ajax({
        url:"${pageContext.request.contextPath}/admin/stats/index",
        data:{'startDate':startDate , "endDate":endDate},
        type:"post",
        dataType:"json",
        success:function(result){
            if(result.totSales == null || result.totSales == ""){
                alert("선택한 기간에는 판매한 상품이 없습니다.");
                return;
            }
            var tot_sales = $('#tot_sales'); // 총 매출
            var order_count = $('#order_count'); // 총 주문수
            var sales_per_order = $('#sales_per_order'); // 주문당 평균 매출
            var paymentRate = $('#paymentRate'); // 결제 완료 비율
            var cartCount = $('#cartCount'); // 장바구니 수
            var cancelCount = $('#cancelCount'); // 취소 비율
            var allOrderCount = $('#allOrderCount'); // 전체 주문수
            var allMemberCount = $('#allMemberCount'); // 총 회원수
            var memberCount = $('#memberCount'); // 신규가입회원 수
            var topSelling = $('#topSelling');
            topSelling.find($(".topProduct")).remove();

            tot_sales.html(result.totSales.toString().replace(/\B(?=(\d{3})+(?!d))/g, ","));
            order_count.html(result.orderCount.toString().replace(/\B(?=(\d{3})+(?!d))/g, ","));
            paymentRate.html((100*result.orderCount.allOrderCount).toFixed(2).toString().replace(/\B(?=(\d{3})+(?!d))/g, ","));
            cartCount.html(result.cartCount.toString().replace(/\B(?=(\d{3})+(?!d))/g, ","));
            cancelCount.html(result.cancelCount.toString().replace(/\B(?=(\d{3})+(?!d))/g, ","));
            allOrderCount.html(result.allOrderCount.toString().replace(/\B(?=(\d{3})+(?!d))/g, ","));
            sales_per_order.html((result.totSales/result.orderCount).toFixed(2).toString().replace(/\B(?=(\d{3})+(?!d))/g, ","));
            allMemberCount.html(result.allMemberCount.toString().replace(/\B(?=(\d{3})+(?!d))/g, ","));
            memberCount.html(result.memberCount.toString().replace(/\B(?=(\d{3})+(?!d))/g, ","));

            $.each(result.topSellingList, function(index, value) {
                let div=
                    <div class="topProduct d-flex align-items-center mb-15px">
                        <div class="text-truncate">
                            <div>${value.product_name}</div>
                        </div>
                        <div class="ms-auto text-center">
                            <div class="fs-13px"><span data-value="0">${value.sellingCount}</span></div>
                            <div class="text-gray-500 fs-10px">판매수</div>
                        </div>
                    </div>
            });
        }
    });
}
```

날짜별로 계산된 결과값을  
AJAX로 구현

## 관리자 페이지(상품 리스트)

상품 수정

고유번호	58
카테고리 코드	9
제품코드	5562146245)
상품명	엡코 SUITMASTER SETTLER 600W S0PLUS Standard 230V
브랜드	엡코
제조사	엡코
공급가	45000
소비가	45000
세금비율	10
판매가	45900
재고	30
등록날짜	2021-08-31
판매여부	1

상품 목록													
페이지	10	개별 보기											
고유번호	이미지	상품코드	분류	상품명	브랜드	제조사	판매기	재고	등록일	판매여부	선택	수정	
58		S562196245	9	Corsair SUEMASTER SETTER 600W 80PLUS Standard 230V EU	Corsair	Corsair	49900	30	2021-08-31	1	<button>삭제</button>	<button>수정</button>	
59		S565581473	3	AMD 라이젠 4세대 5600X(버미어)(정품)	AMD	AMD	332300	50	2021-08-31	1	<button>삭제</button>	<button>수정</button>	
60		S565574764	3	인텔 코어i5-10세대 10400F (코멧레이크S)(정품)	인텔	인텔	222000	60	2021-08-31	1	<button>삭제</button>	<button>수정</button>	
61		S5655769211	3	AMD 라이센스 PRO 4350G (로누리)(온더워)	AMD	AMD	219400	70	2021-08-31	1	<button>삭제</button>	<button>수정</button>	
62		S565580357	3	인텔 코어i7 11세대 11700K (로켓레이크S)(정품)	인텔	인텔	485500	30	2021-08-31	1	<button>삭제</button>	<button>수정</button>	
63		S565581139	3	AMD 라이젠 9 4세대 5900X(버미어)(정품)	AMD	AMD	666500	70	2021-08-31	1	<button>삭제</button>	<button>수정</button>	
65		S565589844	6	NVIDIA 시포스 RTX 3070 BLACK EDITION OC D6 8GB LHR	NVIDIA	이엔비디아	1404000	10	2021-08-31	1	<button>삭제</button>	<button>수정</button>	
67		S565583197	6	ASRock Phantom Gaming D 라데온 RX 6900 XT OC D6 16GB 에즈윈	ASRock	ASRock	1998000	5	2021-08-31	1	<button>삭제</button>	<button>수정</button>	
68		S565585417	4	ASRock B560M PRO4 헤즈衿	ASRock	ASRock	134000	100	2021-08-31	1	<button>삭제</button>	<button>수정</button>	
69		S565586438	4	ASUS PRIME H510M E 아이보리	ASUS	ASUS	92900	100	2021-08-31	1	<button>삭제</button>	<button>수정</button>	

상품 목록에서 수정을 클릭하면 MODAL 창이 뜨게 구현  
Bootstrap 사용

```
$(document).on("click", ".open_modal", function () {
    var product_id = $(this).data('product_id');
    var category_id = $(this).data('category_id');
    var product_code = $(this).data('product_code');
    var product_name = $(this).data('product_name');
    var brand = $(this).data('brand');
    var manufacturer = $(this).data('manufacturer');
    var supply_price = $(this).data('supply_price');
    var consumer_price = $(this).data('consumer_price');
    var tax = $(this).data('tax');
    var selling_price = $(this).data('selling_price');
    var stock = $(this).data('stock');
    var regdate = $(this).data('regdate');
    var availability = $(this).data('availability');

    $("#product_id").val(product_id);
    $("#category_id").val(category_id);
    $("#product_code").val(product_code);
    $("#product_name").val(product_name);
    $("#brand").val(brand);
    $("#manufacturer").val(manufacturer);
    $("#supply_price").val(supply_price);
    $("#consumer_price").val(consumer_price);
    $("#tax").val(tax);
    $("#selling_price").val(selling_price);
    $("#stock").val(stock);
    $("#regdate").val(regdate);
    $("#availability").val(availability);
});
```

게시판 CRUD와 비슷하지  
만 datatable 라이브러리  
사용해서 구현

Admin/product\_list.jsp

```
<tbody>
    <c:forEach var="product_vo" items="${product_list }" varStatus="status">
        <br>
        <td>${product_vo.product_id}</td>
        <c:forEach var="img_vo" items="#{img_list}">
            <choose>
                <when test="${product_vo.product_id == img_vo.product_id and img_vo.img_category == 'main'}">
                    <td class="text-center"><img class="rounded h-30px" src="" alt="" /></td>
                </when>
                <otherwise>
                    <td></td>
                </otherwise>
            </choose>
        </c:forEach>
        <td>${product_vo.product_code}</td>
        <td>${product_vo.category_id}</td>
        <td>${product_vo.product_name}</td>
        <td>${product_vo.brand}</td>
        <td>${product_vo.manufacturer}</td>
        <td>${product_vo.selling_price}</td>
        <td>${product_vo.stock}</td>
        <td>${product_vo.regdate}</td>
        <td>${product_vo.availability}</td>
        <td class="text-center"><a href="${pageContext.request.contextPath }/admin/product/delete?product_id=${product_vo.product_id }" class="btn btn-sm btn-primary w-60px m-10px">삭제</a>
            <a href="#" data-modal="modal-dialog" class="open_modal btn btn-white h-30px" data-bs-toggle="modal">상세보기</a>
            <a href="#" data-product-id="${product_vo.product_id }" data-category-id="${product_vo.category_id }" data-product-code="${product_vo.product_code }" data-product-name="${product_vo.product_name }" data-brand="${product_vo.brand }" data-manufacturer="${product_vo.manufacturer }" data-supply-price="${product_vo.supply_price }" data-consumer-price="${product_vo.consumer_price }" data-tax="${product_vo.tax }" data-selling-price="${product_vo.selling_price }" data-stock="${product_vo.stock }" data-regdate="${product_vo.regdate }" data-availability="${product_vo.availability }" >수정</a>
        </td>
    </c:forEach>
</tbody>
```

# 관리자 페이지(상품 추가)

상품의 카테고리를 두 개 이상 넣기위해 멀티체크박스로 구현

Neocom

상품 추가

상품코드 Product Code

상품명 Product Name

카테고리

CPU

- 인텔 CPU종류
  - 셀러론
  - 奔腾
  - 코어5
  - 코어7
  - 코어9
  - 펜티엄
- AMD CPU종류
  - 라이젠3
  - 라이젠5
  - 라이젠7
  - 라이젠9
- 소켓 구분
  - AMD(소켓AM4)
  - AMD(소켓TR4)
  - AMD(소켓SWR8)
  - 인텔(소켓1151V2)
  - 인텔(소켓1200)
  - 인텔(소켓2066)
  - 인텔(소켓3647)
- 코어 수
  - 10코어
  - 12코어
  - 14코어
  - 16코어

브랜드 Brand

제조사 Manufacturer

공급가 Supply Price

소비자가 Consumer price

과세율 tax

판매가 Selling Price

재고 Stock

판매여부
  미판매  판매

메인이미지선택

파일 선택 선택된 파일 없음

설명이미지선택

파일 선택 선택된 파일 없음

이미지

등록

Admin/product\_insert.jsp

```

function catelist(category_id){
  $("#lower_category").empty();
  $.ajax({
    url:"${pageContext.request.contextPath}/admin/product/findCategories",
    data:{category_parent:category_id},
    type:"post",
    dataType:"json",
    beforeSend: function (jqXHR, settings) {
      // var header = ${"meta[name='csrf_header']"};
      // var token = ${"meta[name='csrf']"};
      // jqXHR.setRequestHeader(header, token);
    },
    success:function(data){
      $.each(data.upperCategoryList,function(index,item){
        $("#lower_category").append(<label class="form-label col-form-label col-md-3"> ${item.category_name} </label>
          <div class="#${item.category_id}_list col-md-9 pt-2"></div> );
        $.each(data.list,function(index2,list){
          $.each(list,function(index3, item2){
            if(item.category_id==item2.category_parent){
              $("#lower_category").find("#"+item.category_id + "_list").append(
                <div class="form-check form-check-inline">
                  <input class="form-check-input" type="checkbox" name="category_ids" value="${item2.category_id}" />
                  <label class="form-check-label" for="category_ids">${item2.category_name}</label>
                </div>
              );
            }
          });
        });
      });
    }
  });
}

$(document).bind("change", "#.prd_imgs", function(e){
  var inputfile = $("input[type='file']");
  var formData = new FormData();

  for(var i = 0; i < inputfile.length ; i++){
    if(inputfile[i].value){
      console.log(inputfile[i].id);
      formData.append(inputfile[i].name, inputfile[i].files[0]);
    }
  }

  // FormData의 값 확인
  for (var pair of formData.entries()) {
    var key=pair[0];
    var filechk=pair[1];

    console.log(key, " " + filechk);
  }

  $.ajax({
    type: "POST",
    enctype: "multipart/form-data",
    url: "${pageContext.request.contextPath}/admin/product/addimg",
    data: formData,
    enctype: "multipart/form-data",
    processData: false,
    contentType: false,
    cache: false,
    timeout: 600000,
    success: function (result) {
      showUploadedFile(result);
      $("#.uploadDiv").html(cloneObj.html());
    },
    error: function (e) {
      console.log("ERROR : ", e);
      alert("Fail!");
    }
  });
});

```

대분류가 변경되면  
세부 카테고리들이  
변경되도록  
ajax로 구현

메인보드	<input type="checkbox"/> AMD CPU용	<input type="checkbox"/> 인텔 CPU용			
CPU 소켓	<input type="checkbox"/> AMD(CPU나침)	<input type="checkbox"/> AMD(소켓AM4)	<input type="checkbox"/> AMD(소켓TR4)	<input type="checkbox"/> AMD(소켓SWR8)	
	<input type="checkbox"/> 인텔(소켓1151V2)	<input type="checkbox"/> 인텔(소켓1200)	<input type="checkbox"/> 인텔(소켓2066)	<input type="checkbox"/> 인텔(소켓3647)	
시全面落实	<input type="checkbox"/> (AMD)560	<input type="checkbox"/> (AMD)A320	<input type="checkbox"/> (AMD)B450	<input type="checkbox"/> (AMD)B550	<input type="checkbox"/> (인텔)H410
	<input type="checkbox"/> (인텔)B560	<input type="checkbox"/> (인텔)H470	<input type="checkbox"/> (인텔)H510		
플퍼리	<input type="checkbox"/> ATX(30.5x24.4cm)	<input type="checkbox"/> E-ATX(30.5x33.0cm)	<input type="checkbox"/> M-STX(14.0x14.7cm)	<input type="checkbox"/> M-ITX(17.0x17.0cm)	
램모듈 종류	<input type="checkbox"/> DDR4	<input type="checkbox"/> DDR4 노트북용			

파일 이미지를 선택하면 바로 파일  
이 외부저장소에 업로드 되게 구현

DB에는 등록을 눌러야만  
파일값(원파일명, 파일저장명,  
파일 크기 등)이 등록되도록 구현

AdminProductController.java

```

@GetMapping("/admin/product/imgdisplay")
@ResponseBody
public ResponseEntity<byte[]> getFile(String fileName) {
  File file = new File(fileName);
  ResponseEntity<byte[]> result = null;
  try {
    HttpHeaders header = new HttpHeaders();
    header.add("Content-Type", Files.probeContentType(file.toPath()));
    result = new ResponseEntity<>(FileCopyUtils.copyToByteArray(file), header, HttpStatus.OK);
  } catch (IOException e) {
    e.printStackTrace();
  }
  return result;
}

@RequestMapping(value = "/admin/product/addimg", method = { RequestMethod.POST })
public @ResponseBody ResponseEntity<List<Product_ImgDTO>> insertImg(Model model,
  @RequestParam(name = "main_img", required = false) MultipartFile main_img,
  @RequestParam(name = "description_img", required = false) MultipartFile description_img,
  List<Product_ImgDTO> list = new ArrayList<>();

  if (main_img != null) {
    Product_ImgDTO main_ImgDTO = service1.uploadImg(main_img, "main");
    list.add(main_ImgDTO);
  }

  if (description_img != null) {
    Product_ImgDTO description_ImgDTO = service1.uploadImg(description_img, "description");
    list.add(description_ImgDTO);
  }

  return new ResponseEntity<>(list, HttpStatus.OK);
}

```

# 관리자 페이지

관리자 페이지  
나머지 목록들도  
상품 목록&추가와  
비슷하게 구현

The image displays a comprehensive screenshot of an Admin Page interface, featuring several interconnected modules:

- Dashboard:** A central hub with a dark theme, displaying navigation menus for "회원", "상품 카테고리", "상품", "주문", "카뮤니티", and "통계".
- User Management:** Shows a list of users with columns for ID, 이름 (Name), 닉네임 (Nickname), 생년월일 (Birthdate), and 가입일 (Join Date). Actions like 삭제 (Delete) and 수정 (Edit) are available.
- Product Management:** Includes sections for "문의사항 목록" (Feedback List) and "주문 목록" (Order List). The Feedback List shows entries like "문의드립니다." and "제품에 대해 문의드립니다.", while the Order List lists items such as "파일 주기 테스트" and "파일업로드테스트 완료".
- Category Management:** Features a "카테고리 목록" (Category List) and a "카테고리 추가" (Add Category) form. The category list includes items like "cpu", "mainboard", "ram", "vga", "ssd", "hdd", and "power".
- Order Management:** Displays a detailed "주문 목록" (Order List) with columns for 주문번호 (Order No.), 주문날짜 (Order Date), 주문상태 (Order Status), 주문원소 (Order Source), and 상세보기 (View Details).
- Feedback Management:** Shows a "문의사항 목록" (Feedback List) with entries from users like "유저", "admin", and "tset".
- Search Functionality:** A search bar at the top of the page allows users to search across all modules.