프로젝트 포트폴리오

이름 : 김희원

기술 스택 : Back End / Front End

연락처 : 010-5571-7187

이메일 : khww0301@gmail.com

Git 주소 : https://github.com/khww0301/Portfolio

목차 table of contents

1 프로젝트 개요

2 Scenario - 실행 순서

3 프로젝트 소개

1

프로젝트 개요

프로젝트 개요

프로젝트 목적

주요 기능

- 온라인에서 책을 구매하고, 장바구니 및 주문 배송 관리 기능을 제공하는 도서 쇼핑몰을 개발
- 사용자에게 편리한 도서 구매 경험을 제공하고, 관리자에게 효율적인 관리 기능을 제공

프로젝트 개요

프로젝트 목적

주요 기능

- 회원가입 및 로그인 : 개인 정보를 관리 및 추적
- 장바구니 : 선택한 도서를 장바구니에 담고, 수량 변경 및 삭제
- 주문하기 : 장바구니에 담긴 도서를 주문
- 관리자페이지 : 고객센터, 공지사항, QNA등 고객 관리 기능

2

Scenario (실행순서)

Scenario - (실행 순서)

Scenario

상단 메뉴에서 회원가입 클릭 -> 회원정보 입력 -> 비밀번호 찾기 클릭 -> ID, PWD 입력으로 로그인 -> 상단 메뉴에서 마이페이지 아이콘 클릭 -> 프로필 아이콘 클릭 -> 회원정보 수정 및 탈퇴 -> Test ID, PWD 입력으로 로그인 -> 메뉴에서 베스트 클릭 -> 체크 박스 선택 -> 장바구니 버튼 클릭 -> 장바구니 전체 삭제 클릭 -> 베스트셀러 페이지 주문하기 클릭 -> 주문/배송페이지에서 주문 확인 -> 메뉴에 있는 각 도서 클릭 -> 장르별 도서에서 서브 메뉴 클릭 -> 도서 클릭하여 도서 상세 페이지로 이동 -> 장바구니 담기 및 바로구매 클릭 -> 주문/배송으로 이동 -> 장바구니 아이콘 클릭

Scenario - (실행 순서)

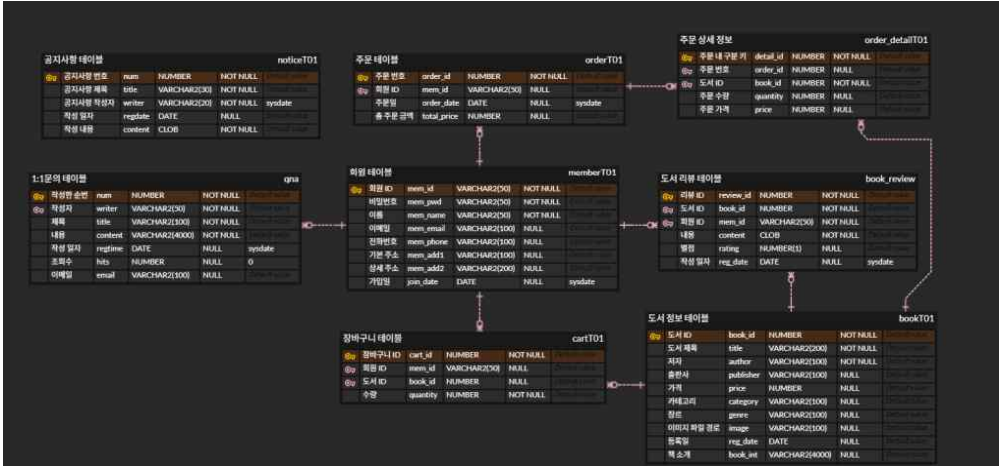
Scenario

체크 및 수량 등 선택 후 주문하기 클릭 -> 주문/배송으로 이동 -> 상단 메뉴에서 고객센터 클릭 -> 공지사항 더보기 클릭 -> 공지사항 쓰기 클릭 -> 공지사항 입력 후 write버튼 클릭 -> 공지사항 세부 항목 클릭 -> 수정 버튼 클릭 -> 수정 내용 입력 후 write버튼 클릭 -> 상단 메뉴 고객센터 클릭 -> 1:1 문의 접수 아이콘 클릭 -> 문의 내용 입력 후 접수 버튼 클릭 -> 문의 내용 세부 항목 클릭 -> 수정 버튼 클릭 -> 수정 사항 입력 후 수정 버튼 클릭 -> 메인 페이지로 이동 -> 베스트셀러 소년이 온다 도서 클릭 -> 도서 상세 페이지 하단 리뷰 작성 후 리뷰등록 클릭 -> 도서 리뷰 목록 수정 클릭 -> 수정 내용 입력 후 수정완료 클릭 -> 삭제 버튼 클릭 -> 로그아웃 클릭

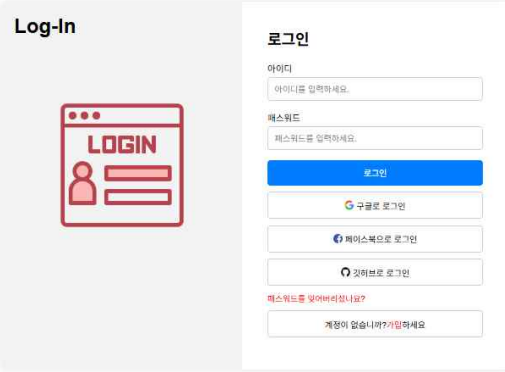
3

프로젝트 소개

프로젝트 – 도서쇼핑몰(기획 및 설계)



프로젝트 – 도서쇼핑몰 (페이지 주요기능 1 / 5)



로그인

1. 도서 쇼핑몰 회원 접속 - 로그인

사용자가 입력한 값과 MemberT02 테이블 ID, 패스워드 값을 비교하여 일치한다면 Session에 저장

프로젝트 – 도서쇼핑몰 (페이지 주요기능 1 / 5)

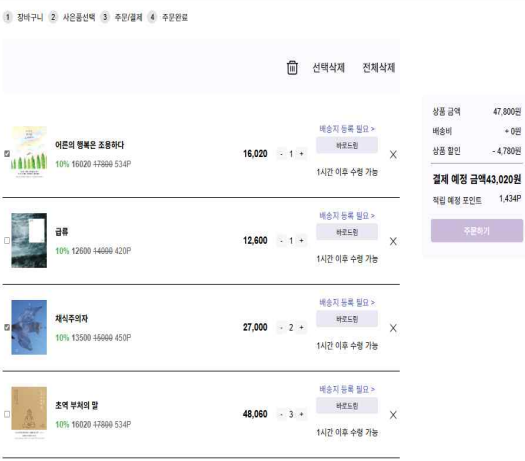
```
<%
    request.setCharacterEncoding("utf-8");

    String memValues[] = request.getParameterValues("memValues");
    MemberDTO memDTO = new MemberDTO(memValues[0], memValues[1]);
    MemberDAO memDAO = MemberDAO.getInstance();
    String rst = memDAO.memLog(memDTO);

    if (rst.contains("mem_id")) {
        session.setAttribute("mem_id", memValues[0]);
        session.setAttribute("mem_pwd", memValues[1]);
%>


$(document).ready(function () {
    // 페이지 로드 시 포커스
    const pageId = $("body").attr("id");
    if (pageId === "update") {
        $("#password").focus();
    } else {
        $("#id").focus();
    }

    // 엔터 입력 시 자동 포커스
    $(".input").on("keydown", function (event) {
        if (event.key === "Enter") {
            event.preventDefault();

            const allInputs = $(".input");
            const currentIndex = allInputs.index(this);

            if (currentIndex >= 0 && currentIndex < allInputs.length - 1) {
                allInputs.eq(currentIndex + 1).focus();
            } else if (currentIndex === allInputs.length - 1) {
                $("#btnO1").focus();
            }
        }
    });
});
```

로그인

Jquery로 페이지 로드 시 ID에 Focus가 도록 구현 및 엔터 시 포커스 이동하도록 구현

Input Name으로 Value 값을 가져와 회원 테이블과 비교하고 일치할 경우 Session에 저장
WebFont로 글꼴 변경

프로젝트 – 도서쇼핑몰 (페이지 주요기능 2 / 5)



장바구니

1. 장바구니 테이블 리스트 CRUD
2. 도서 구매 처리

Session에 저장된 ID를 가져와 장바구니 테이블 리스트를 출력
수량 변경 및 체크 박스 선택 시 전체 금액 반영
주문하기 버튼 클릭 시 체크된 항목 주문 테이블에 저장

프로젝트 – 도서쇼핑몰 (페이지 주요기능 2 / 5)

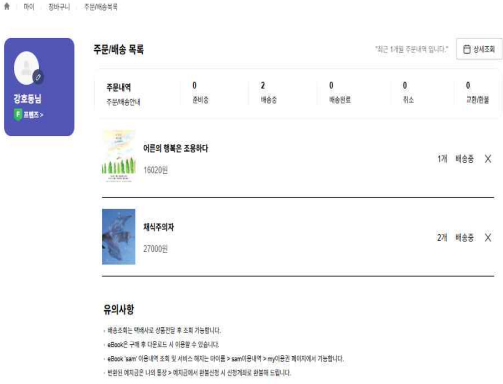
```
    minusBtn.addEventListener("click", () => changeQty(-1));
    plusBtn.addEventListener("click", () => changeQty(1));
  });

  checkboxes.forEach(checkbox => {
    checkbox.addEventListener("change", updateSummary);
  });

$(document).ready(function () {
  // 페이지 로드 시 포커스
  const pageId = $("body").attr("id");
  if (pageId === "update") {
    $("#password").focus();
  } else {
    $("#id").focus();
  }

  // 엔터 입력 시 자동 포커스
  $(".input").on("keydown", function (event) {
    if (event.key === "Enter") {
      event.preventDefault();

      const allInputs = $(".input");
      const currentIndex = allInputs.index(this);

      if (currentIndex >= 0 && currentIndex < allInputs.length - 1) {
        allInputs.eq(currentIndex + 1).focus();
      } else if (currentIndex === allInputs.length - 1) {
        $("#btn01").focus();
      }
    }
  });
});
```

장바구니

DOM addEventListner 사용하여 체크된 항목 선별하여 금액 반영 및 테이블 CRUD 구현

Session에 저장된 ID를 가져와 DAO로 해당하는 항목 리스트 가져와 DTO에 담은 뒤 JSP 스크립틀릿 표현식으로 출력

프로젝트 – 도서쇼핑몰 (페이지 주요기능 3 / 5)



주문/배송

1. 주문 테이블 리스트 출력
2. 주문 테이블 리스트 삭제

Session ID를 가져와 주문 테이블 리스트 출력
X 버튼 클릭 시 주문 ID 값으로 해당하는 항목 삭제

프로젝트 – 도서쇼핑몰 (페이지 주요기능 3 / 5)

```
OrderDAO dao = OrderDAO.getInstance();
List<OrderDTO> orderList = dao.orderSelOrderDTO(vId);
MemberDAO memDAO = MemberDAO.getInstance();
String name = memDAO.memName(memDTO);

public ArrayList<BookDTO> orderSel(String id) {
    ArrayList<BookDTO> dtoList = new ArrayList<BookDTO>();
        try {
            conn = DBConnection.getConnection();
            sql = "SELECT b.book_id, b.title, b.price, b.image, od.quantity\r\n"
                  + "FROM member-TO2 m\r\n"
                  + "JOIN order-TO1 o ON m.mem_id = o.mem_id\r\n"
                  + "JOIN order_detailTO1 od ON o.order_id = od.order_id\r\n"
                  + "JOIN bookTO1 b ON od.book_id = b.book_id\r\n"
                  + "where m.MEM_ID = ?";
            pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, id);//로그인 id값 넣기
            rs = pstmt.executeQuery();
            while(rs.next()) {
                book_id = rs.getInt("book_id");
                title = rs.getString("title");
                price = rs.getInt("price");
                image = rs.getString("image");
                quantity = rs.getInt("quantity");
                dtoList.add(new BookDTO(book_id, title, price, image, quantity));
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }finally {
            DBConnection.close(rs, pstmt, conn);
        }
        return dtoList;
}
```

주문/배송

Session ID를 가져와 주문 테이블 리스트 출력

X 버튼 클릭 시 주문 ID 값으로 해당하는 항목 삭제

프로젝트 – 도서쇼핑몰 (페이지 주요기능 3 / 5)

```
<form method="post" action="deleteOrder.jsp">
    <input type="hidden" name="order_id" value="<%=order.getOrder_id()%>">
    <input type="hidden" name="id" value="<%=id%>">
    <input type="hidden" name="pwd" value="<%=pwd%>">
    <button type="submit" class="delete-btn"
   onclick="return confirm('이 주문을 삭제하시겠습니까?');">X</button>
</form>
```

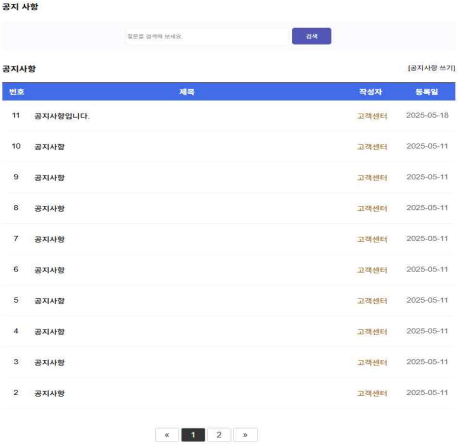
```
String orderIdStr = request.getParameter("order_id");
int order_id = Integer.parseInt(orderIdStr);

OrderDAO dao = OrderDAO.getInstance();
boolean result = dao.deleteOrder(order_id);

if (result) {
    response.sendRedirect("orderDelivery.jsp?id=" + vId + "&pwd=" + pwd);
} else {
    out.println("<script>alert('삭제 실패'); history.back();</script>");
}
```

주문/배송

Session ID를 가져와 주문 테이블 리스트 출력

X 버튼 클릭 시 주문 ID 값으로 해당하는 항목 삭제

프로젝트 – 도서쇼핑몰 (페이지 주요기능 4 / 5)



공지사항

1. 공지사항 CRUD 및 페이징 기능

공지사항 리스트 배열에 담아 유기적으로 페이징 기능 실현되도록 구현

프로젝트 – 도서쇼핑몰 (페이지 주요기능 4 / 5)

```
<%
    ArrayList<NoticeDTO> notices =
        (ArrayList<NoticeDTO>) session.getAttribute("notices");
    session.removeAttribute("notices");
%>
<script>
const notices = [
<%
    for (int i = 0; i < notices.size(); i++) {
        NoticeDTO n = notices.get(i);
%>
    {
        num: <%=n.getNum()%>,
        title: "<%=n.getTitle()%>",
        writer: "<%=n.getWriter()%>",
        date: "<%=n.getRegdate()%>"
    }<%= (i < notices.size() - 1) ? "," : "" %>
<%
    }
%>
];
```

```
let recordName = {
    totalRecord: notices.length,
    recPerPage: 10,
    pagePerBlock: 10,
    totalPage: 0,
    totalBlock: 0,
    nowPage: 1,
    nowBlock: 1,
    recOfBeginPage: 0,
    pageOfBeginBlock: 0
};
```

공지사항

Servlet MVC방식으로 데이터 가져와서 ArrayList에 값을 담고, 배열에 키와 값으로 값을 담아서 페이징 기능 구현

페이징 전체 길이를 ArrayList 길이로 정하고, 10개의 페이지와 블록으로 구성하여 번호 선택 시 유기적으로 리스트 변경되도록 구현

프로젝트 – 도서쇼핑몰 (페이지 주요기능 4 / 5)

```
function createPagination() {
    pagination.innerHTML = '';

    const crtAFirst = document.createElement('a');
    crtAFirst.href = '#';
    crtAFirst.innerHTML = '&laquo;';
    crtAFirst.addEventListener('click', movePrevBlock);
    pagination.appendChild(crtAFirst);

    let startPage = (recordName.nowBlock - 1) * recordName.pagePerBlock + 1;
    let endPage = Math.min(startPage + recordName.pagePerBlock - 1, recordName.totalPage);

    for (let page = startPage; page <= endPage; page++) {
        const crtA = document.createElement('a');
        crtA.href = '#';
        crtA.innerHTML = page;
        crtA.addEventListener('click', function (e) {
            e.preventDefault();
            setActivePage(page);
        });
        pagination.appendChild(crtA);
    }
```

공지사항

Servlet MVC방식으로 데이터 가져와서 ArrayList에 값을 담고, 배열에 키와 값으로 값을 담아서 페이징 기능 구현

페이징 전체 길이를 ArrayList 길이로 정하고, 10개의 페이지와 블록으로 구성하여 번호 선택 시 유기적으로 리스트 변경되도록 구현

프로젝트 – 도서쇼핑몰 (페이지 주요기능 5 / 5)

공지사항

공지사항

1. 공지사항 세부내용 조회

공지사항 클릭 시 해당하는 번호로 테이블 리스트를 출력
수정 및 삭제 클릭 시 수정페이지 이동 및 해당하는 항목 삭제

프로젝트 – 도서쇼핑몰 (페이지 주요기능 5 / 5)

```
<%
  while(rs.next()){
      vNo = rs.getInt("num");
      vWriter = rs.getString("writer");
      vTitle = rs.getString("title");
      vRegTime = rs.getString("regdate");
      vContent = rs.getString("content");
  }
  vTitle = vTitle.replace(" ", " ");
  vContent = vContent.replace(" "," ").replace("\n","<br>");
%>

  Connection conn = DBConnection.getConnection();
  Statement stmt = conn.createStatement();
  String temp = request.getParameter("num");
  if(temp != null){
      vNo = Integer.parseInt(temp);
  }
  String sql = "select * from noticeT01 where num = " + vNo;
  ResultSet rs = stmt.executeQuery(sql);
%>
<nav>
   <span ><a id="list" href="<%=contextPath %>
         /boardBean.do?category=noticeUpdate&num=<%=vNo%>">[ 수 정 ]</a></span>
   <span ><a id="list" href="<%=contextPath %>
         /boardBean.do?category=noticeDel&num=<%=vNo%>">[ 삭 제 ]</a></span>
   <span ><a id="list" href="<%=contextPath %>
         /boardBean.do?category=notice">[목록보기]</a></span>
</nav>
```

```
<tr>
   <th class="table_th">글번호</th>
   <td class="head_txt"><%=vNo %></td>
   <th class="table_th">Writer</th>
   <td class="head_txt"><%=vWriter %></td>
</tr>
<tr>
   <th class="table_th">Title</th>
   <td class="head_txt"><%=vTitle %> </td>
   <th class="table_th">Date</th>
   <td class="head_txt"><%=vRegTime %></td>
</tr>
```

공지사항

공지사항 클릭 시 해당하는 번호로 테이블 리스트를 출력

수정 및 삭제 클릭 시 번호를 넘겨서 request.parameter 해당하는 항목 삭제 및 수정 페이지로 이동