

고객센터 구성

타임티켓
고객센터페이지 구현

1. 자주 묻는 질문 페이지

FreqDAO를 이용한 공지사항 페이지 구현

BViewHandler

FreqDTO

FreqDAO, FreqDAOImpl

view.jsp

/clientcenter/view.do

자주 묻는 질문 상세 내용 출력

2. 자주 묻는 질문 목록 페이지

목록 클릭 시 각각의 목록 페이지로 이동

ListHandler

FreqDTO, PageDTO

GongDAO, FreqDAO

list.jsp

/clientcenter/list.do

자주 묻는 질문 목록 출력

고객센터 구성

타임티켓
고객센터페이지 구현

1. 공지사항 페이지

GongDAO를 이용한 공지사항 페이지 구현

GViewHandler

GongDTO

GongDAO, GongDAOImpl

gview.jsp

/clientcenter/gview.do

공지사항 상세 내용 출력

2. 공지사항 목록 페이지

목록 클릭 시 각각의 목록 페이지로 이동

ListHandler

GongDTO, GPageDTO

GongDAO, FreqDAO

list.jsp

/clientcenter/list.do

공지사항 목록 출력

6. 상세 기능
구현

고객센터에 사용된 메서드 구성 요약

// 자주 묻는 질문 내용보는 메서드 view()

```
public FreqDTO view(String fr_code) throws SQLException {
    FreqDTO dto = null;
    //생략
    return dto;
}
```

// 자주 묻는 질문 목록 번호 보여주는 메서드 selectFreq()

```
public ArrayList<FreqDTO> selectFreq(int currentPage, int
    numberPerPage) throws SQLException{
    ArrayList<FreqDTO> list = null;
    FreqDTO dto = null;
    //생략
    return list;
}
```

78%

// 자주 묻는 질문 총 페이지수 반환하는 메서드 getTotal()

```
public int getTotal() throws SQLException {
    int total = 0;
    //생략
    return total;
}
```

// 공지사항 내용보는 메서드 view()

```
public GongDTO view(String fr_code) throws SQLException {
    GongDTO dto = null;
    //생략
    return dto;
}
```

// 공지사항 목록 번호 보여주는 메서드 selectFreq()

```
public ArrayList<GongDTO> selectGong(int currentPage, int
    numberPerPage) throws SQLException {
    ArrayList<GongDTO> glist = null;
    GongDTO dto = null;
    //생략
    return glist;
}
```

66%

// 공지사항 총 페이지수 반환하는 메서드 getTotal()

```
public int getTotal() throws SQLException {
    int total = 0;
    //생략
    return total;
}
```

고객센터 > 자주 묻는 질문을 보기 위한 FreqDAO 쿼리

// 자주 묻는 질문 내용보는 메서드 view()

```
public Section01DTO view(String tic_code) throws SQLException {
    Section01DTO vdto = null;
    String sql = " SELECT freq_code,freq_name ,freq_cont "
        + " FROM freq_question "
        + " WHERE regexp_like(freq_code , ? , 'i')";

    //생략
    return dto;
}
```

// 자주 묻는 질문 목록 번호 보여지는 메서드 selectFreq()

```
public ArrayList<FreqDTO> selectFreq(int currentPage, int numberPerPage) throws SQLException{
    ArrayList<FreqDTO> list = null; , FreqDTO dto = null;
    String sql = "SELECT COUNT(*) "
        + "FROM freq_question";

    //생략
    return list;
}
```

// 자주 묻는 질문 총 페이지수 반환하는 메서드 getTotal()

```
public int getTotal() throws SQLException {
    int total = 0;
    String sql = "SELECT * "
        + "FROM( "
        + " SELECT ROWNUM no, t.* "
        + " FROM( "
        + " SELECT to_number(REPLACE(freq_code, 'fq_', '')) AS order_num, freq_code, freq_name, freq_cont "
        + " FROM freq_question "
        + " ORDER BY order_num DESC "
        + " ) t "
        + " ) m "
        + "WHERE m.no BETWEEN ? AND ? ";

    //생략
    return total;
}
```

66%

78%

고객센터 > 공지사항을 보기 위한 GongDAO 쿼리

```
// 공지사항 내용보는 메서드 view()
public GongDTO view(String fr_code) throws SQLException {
    GongDTO dto = null;
    String sql = " SELECT noti_num,noti_date,noti_title ,noti_cont "
        + " FROM notice "
        + " WHERE regexp_like(noti_num , ?, 'i')";
    //생략
    return dto; }

// 공지사항 목록 번호 보여지는 메서드 selectFreq()
public ArrayList<GongDTO> selectGong(int currentPage, int numberPerPage) throws SQLException {
    ArrayList<GongDTO> glist = null; , GongDTO dto = null;
    String sql = "SELECT * "
        + " FROM( "
        + " SELECT ROWNUM no, t.* "
        + " FROM( "
        + " SELECT to_number(REPLACE(noti_num, 'noti_', '')) AS order_num, noti_num, noti_date, noti_title, noti_cont "
        + " FROM notice "
        + " ORDER BY noti_date DESC "
        + ") t "
        + ") m "
        + " WHERE m.no BETWEEN ? AND ? ";
    //생략
    return glist; }

// 공지사항 총 페이지수 반환하는 메서드 getTotal()
public int getTotal() throws SQLException {
    int total = 0;
    String sql = " SELECT noti_num,noti_date,noti_title ,noti_cont "
        + " FROM notice "
        + " WHERE regexp_like(noti_num , ?, 'i')";
    //생략
    return total; }
```

78%

66%

고객센터 > 자주 묻는 질문 및 공지사항 페이징 처리

```
<c:if test="${ pdto.prev }">
  <a
    href="<%= contextPath %>/timeticket/clientcenter/list.do?tb=${param.tb }&currentpage=${ pdto.start - 1 }">
    &laquo; </a>
</c:if>

<c:forEach begin="${ pdto.start }" end="${ pdto.end }" step="1"
  var="i">
  <c:if test="${ pdto.currentPage eq i }">
    <a href="#" class="active"> ${ i } </a>
  </c:if>

  <c:if test="${ pdto.currentPage ne i }">
    <a
      href="<%= contextPath %>/timeticket/clientcenter/list.do?tb=${param.tb }&currentpage=${ i }">
      ${ i } </a>
  </c:if>

</c:forEach>

<c:if test="${ pdto.next }">
  <a
    href="<%= contextPath %>/timeticket/clientcenter/list.do?tb=${param.tb }&currentpage=${ pdto.end + 1 }">
    &raquo; </a>
  </c:if>
</div>
</div>
</div>
```

고객센터 > 자주 묻는 질문 및 공지사항 페이징 처리

```
public class PageDTO {  
  
    private int currentPage = 1; // 현재 페이지 번호  
    private int numberPerPage = 8; // 한 페이지 출력할 게시글 개수  
    private int numberOfPageBlock = 4; // 페이지 블록 수  
    private int start, end; // 페이지 시작값, 끝값  
    private boolean prev, next; // 이전버튼, 다음버튼  
  
    public PageDTO(int currentPage, int numberPerPage, int numberOfPageBlock, int total) {  
  
        int totalPages; // 총 페이지 수  
        totalPages = (int) Math.ceil( (double) total / numberPerPage );  
  
        this.currentPage = currentPage;  
        this.start = (currentPage - 1) / numberOfPageBlock * numberOfPageBlock + 1;  
        this.end = start + numberOfPageBlock - 1;  
        this.end = end > totalPages ? totalPages : end;  
  
        this.prev = this.start > 1;  
        this.next = end < totalPages;  
  
    }  
}
```


고객센터

공지사항 및
자주 묻는 질문



결제 기능 구현



구현에 필요한 DTO 및 DAO 메서드

Section05DTO.java

결제에 필요한 티켓정보 및 환불규정

Section05DAO.java

Section05DAOImpl.java

티켓정보, 환불규정, 예매된 티켓 정보 메서드 및 쿼리

구현에 필요한 DAO의 메서드

```
// 1. 티켓정보 보기
Section05DTO tinfo(String tic_code) throws SQLException;
```

```
// 2. 환불규정 보기
Section05DTO refundview(String tic_code) throws SQLException;
```

66%

```
// 3. 예매된 티켓 정보
int inpay(String mem_id, String pay_code, String gwon_code, int book_price,
          int book_cnt, int book_point) throws SQLException;
```

결제 기능 구현



파일 실행 or 요청을 받게 되면 commandHandler에 의해서 PayInfoHandler를 찾는다.

결제를 완료하면 commandHandler에 의해서 PayHandler를 찾는다.

```
@Override
public String process(HttpServletRequest request, HttpServletResponse response) throws Exception {
    System.out.println("PayInfoHandler 호출");
    Connection conn = ConnectionProvider.getConnection();
    Section05DAOImpl dao = new Section05DAOImpl(conn);

    Section05DTO dto = null;
    Section05DTO dto2 = null;

    String tic_code = request.getParameter("tic_code");

    try {
        dto = dao.tinfo(tic_code);
        dto2 = dao.refundview(tic_code);
    } catch (SQLException e) {
        System.out.println("> PayHandler.doGet() 예외...");
        e.printStackTrace();
    }

    conn.close();

    request.setAttribute("dto", dto);
    request.setAttribute("dto2", dto2);

    // 포워딩
    return "/payed/payed.jsp";
}
```

2개의 이벤트 테이블의 데이터를 담은 Section05DTO 생성

티켓 정보의 tinfo 함수를 호출

환불정보의 refundview 함수를 호출

JSP 페이지에서 dto 객체를 사용할 수 있도록 request영역에 추가

JSP 페이지에서 dto2 객체를 사용할 수 있도록 request영역에 추가

```
@Override
public String process(HttpServletRequest request, HttpServletResponse response) throws Exception {
    System.out.println("PayHandler 호출");
    Connection conn = ConnectionProvider.getConnection();
    Section05DAOImpl dao = new Section05DAOImpl(conn);

    int dto = 0;
    HttpSession session = request.getSession();
    String mem_id = (String)session.getAttribute("mem_id");
    String pay_code = request.getParameter("pay_code");
    String gwon_code = request.getParameter("gwon_code");
    int book_price = Integer.parseInt(request.getParameter("book_price"));
    int book_cnt = Integer.parseInt(request.getParameter("book_cnt"));
    int book_point = Integer.parseInt(request.getParameter("book_point"));

    try {
        dto = dao.inpay(mem_id, pay_code, gwon_code, book_price, book_cnt, book_point);
    } catch (SQLException e) {
        System.out.println("> PayHandler.doGet() 예외...");
        e.printStackTrace();
    }

    conn.close();

    request.setAttribute("dto", dto);

    // 포워딩
    return "/mainpage/main.do";
}
```

session에서 값 가져오기 및 parameter 가져오기

결제 기능 구현



```
@Override
public Section05DTO tinfo(String tic_code) throws SQLException {

    String sql = " SELECT tic_name, place, REGEXP_SUBSTR (place_add, '[^,]+' , 1, 1) place_add "
        + " FROM ticket t join place p on t.tic_code = p.tic_code "
        + " WHERE t.tic_code = ? ";

    this.pstmt = this.conn.prepareStatement(sql);
    this.pstmt.setString(1, tic_code);
    this.rs = this.pstmt.executeQuery();

    Section05DTO dto = null;

    if(this.rs.next()) {
        dto = new Section05DTO();

        dto.setTic_name(this.rs.getString("tic_name"));
        dto.setPlace(this.rs.getString("place"));
        dto.setPlace_add(this.rs.getString("place_add"));
    }

    return dto;
}
```

티켓명, 장소, 상세위치를 가져오는 쿼리(정규표현식
으로 구분해서 첫번째 인덱스 값만 가져옴)

```
@Override
public Section05DTO refundview(String tic_code) throws SQLException {

    String ref_rule;
    String ref_cau;
    String ref_way;

    String sql = "SELECT DISTINCT t.tic_code, re.ref_rule, re.ref_cau, re.ref_way "
        + "FROM refund_rule re JOIN ticket t ON t.tic_code = re.tic_code "
        + "WHERE t.tic_code = ? ";

    Section05DTO redto = null;
    try {
        pstmt = this.conn.prepareStatement(sql);
        this.pstmt.setString(1, tic_code);

        rs = this.pstmt.executeQuery();
    }
```

각각의 티켓에 맞는 환불규정들을 가져오는 쿼리

```
@Override
public int inpay(String mem_id, String pay_code, String gwon_code, int book_price, int book_cnt, int book_point)
    throws SQLException {
    int rowCount = 0;

    String sql = " INSERT INTO payed (book_code, mem_id, pay_code, gwon_code, book_date,"
        + " book_price, book_cnt, book_point, book_stat) "
        + " VALUES ( 'book_'||SEQ_payed.NEXTVAL, ?, ?, ?, sysdate, ?, ?, ?, '구매완료' ) ";

    this.pstmt = this.conn.prepareStatement(sql);
    this.pstmt.setString(1, mem_id);
    this.pstmt.setString(2, pay_code);
    this.pstmt.setString(3, gwon_code);
    this.pstmt.setInt(4, book_price);
    this.pstmt.setInt(5, book_cnt);
    this.pstmt.setInt(6, book_point);
}
```

결제가 완료된 티켓을 예매 테이블에 저장하는 쿼리



티켓 예매하기

티켓정보

티켓명

과파전

이용장소

K현대미술관 (서울특별시 강남구 선릉로 807)

예매정보

선택옵션

[유효기간:-2023.10.8] 청소년 1인 입장권

티켓가격

17,000 원

수량

1 매

티켓합계

17,000 원

이용자정보

· 실제로 관람/이용하실 분의 실명/연락처를 입력해 주세요.

· 오기재시 정상 이용 및 긴급연락이 불가할 수 있으며 이에 따른 책임을 지지 않습니다.

이용자이름

권수민

휴대폰번호

010-1234-5678

이메일주소

admin@ss.co.kr

·보유 포인트

280p (최소 500p 이상 사용가능)

·포인트 사용

사용하기

포인트 모두사용

총 결제금액

17,000원

결제방법

N Pay

네이버페이

pay

카카오페이

toss

토스페이

sumus pay

삼성페이

PAYCO

페이코

카드결제

휴대폰결제

무통장입금

◆ 티켓 환불규정

보기

◆ 주의사항 및 약관

보기

☐ 상단의 환불규정 / 주의사항 / 약관을 확인하였으며 이에 동의합니다.

☐ 공연티켓은 관람당일 환불/변경이 절대 불가한 점에 동의합니다.(지각, 지역착오, 연령미준수로 인한 환불불가)

결제하기

상세내역에서 받아온 파라미터 및 세션값들이
티켓정보, 예매정보, 이용자정보에 담긴다

localhost 내용:

결제 방법을 선택해주세요

확인

이름, 휴대폰번호, 이메일주소, 결제방법 미선택, 동의약관 박스 미체크시
각자 다른 경고창이 뜬다

localhost 내용:

결제가 완료되었습니다

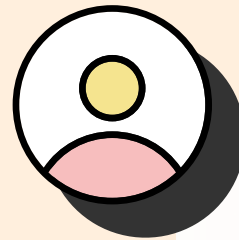
확인

결제가 완료되면 완료창이 뜬다

시연보고 가실까요 !

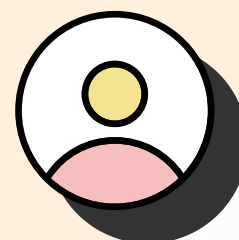
78%

66%



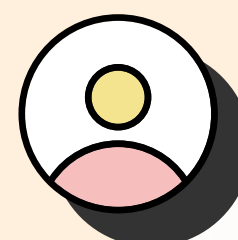
진예림

단언컨대 가장 힘들었고, 가장 재밌었고, 많이 성장할 수 있었던 프로젝트였던 것 같습니다!
제가 팀장으로 부족한 점이 많았을 텐데 그럼에도 불구하고 항상 의견도 잘 수용해주시고 기분 좋게 프로젝트를 마무리 하게 해주셔서 모든 팀원분들께 정말 감사합니다!!! 다들 고생했어요..♡



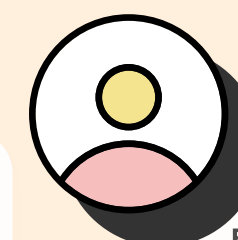
박민종

프로젝트 처음 시작할 때 3주라는 시간이 되게 길게 느껴졌는데 프로젝트를 진행하다보니 너무 짧게 느껴지는 시간이었습니다. 프로젝트로 힘든 와중에도 팀 분위기가 유쾌하고 활기차서 더욱 힘내서 할 수있었던 프로젝트였습니다.



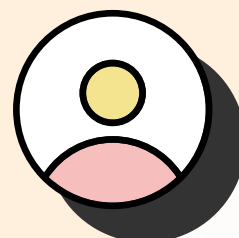
김동현

이번 프로젝트 정말 재밌었어요! 팀원들과 함께하는 모든 순간이 즐거웠고, 항상 유쾌한 분위기였어요. 우리는 어색한 상황도 웃음으로 바꾸고, 실패도 성장의 기회로 받아들였어요. 함께한 시간이 정말로 재미있고 특별했어요! 고마워요, 팀원들!



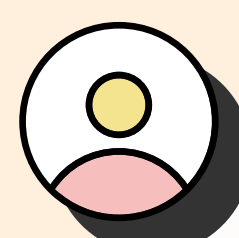
박현주

팀원들과 협력해 각자 페이지를 구현하는 일이 즐거웠습니다.
MVC 패턴에서 세션을 처리하는 부분에 있어 어려움이 있었는데, 모르는 부분은 서로에게 물어보고, 또 책과 인터넷을 찾아가며 해결할 수 있었습니다. 구현 과정에서 지난 5개월 간 배웠던 지식들을 활용해 프로그래밍에 대한 머리 속 지식들을 조금 더 구체화할 수 있었던 것 같습니다. 얼른 스프링 프로젝트를 통해 구현한 페이지를 예쁘게 마무리 하고 싶습니다 ㅎㅎ



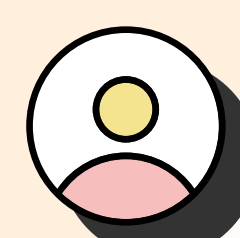
김수민

프로젝트를 하면서 제일 힘들었지만 제일 재밌었어요! 좋은 팀원들을 만나 힘든상황에서도 웃고 서로 격려함으로써 프로젝트를 잘 마무리한거 같습니다. 많이 배웠고 성장한거 같습니다! 팀원들 다들 고생하셨어요~



이진우

3주라는 긴 시간동안 고생해서 다 같이 열심히 해서 만든 결과물을 보니 정말 뿌듯하고 많이 도와준 팀원들에게 감사합니다.



이태규

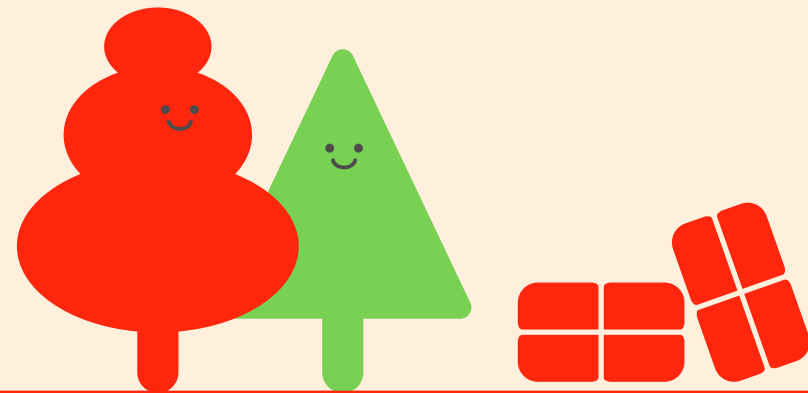
이번 프로젝트에서는 DB연동을 통한 웹페이지 구현과 ajax 를 통해서 동적 구현까지 가능해진 상황이어서 제작에 제한이 있었던 오라클보다 훨씬 다채롭게 코딩을 할 수 있었던 것 같습니다. 하지만 이번 프로젝트 스스로가 배운 것을 제대로 활용하지 못해 아쉬움 또한 남았던 것 같습니다.

TIMETICKET PROJECT



Q & A

쌍용교육센터 18기 2조



경청해주셔서 감사합니다.

TIMETICKET PROJECT

감사합니다

기분이짜조

팀장 | 진예림

기술고문 | 박민종

팀원 | 박현주

팀원 | 이진우

팀원 | 김동현

팀원 | 김수민

팀원 | 이태규