안녕하세요 DB다이렉트 팀의 기말 발표를 시작하겠습니다.

<다음>

저희 팀의 발표 순서는 다음과 같습니다.

<다음>

우선 개발 중 변경된 스키마 내용에 대해 설명드리겠습니다.

<다음>

지금 보시는 스키마는 개발 전 저희 팀이 설계했던 스키마입니다.

<다음>

다음 스키마는 현재 개발을 마친 상태의 스키마인데요.

개발을 진행하면서 달라진 점은 무엇일까요?

<다음>

우선 room 릴레이션이 달라졌는데요.

개발 전에는 1개의 릴레이션이었던 room은 개발 이후

RoomNo와 RoomType 릴레이션으로 1개가 늘어났죠.

그 이유는 종속성과 관련이 있는데요

<다음>

Room 릴레이션에는

<다음>

RoomType이라는 속성이 있고

RoomType에 따라 결정되는 요인인

<다음>

가격, 전망, 구성 요소이라는 속성이 있었습니다.

즉, 릴레이션 내에 종속성이 존재했던 것이죠.

때문에 이를 분리하여 Roomno와 RoomType 릴레이션을 만들었고

RoomType에 해당하는 정보는

Natural Join을 사용하여 얻을 수 있었습니다.

<다음>

이전과 달리 새로 만들어진 릴레이션도 있었는데요.

바로 TABLET릴레이션입니다.

저희 팀은 PBL 모듈 요구사항을 만족시키기 위해

6조의 중간발표를 참고하여

각 호실에 TABLET을 배치한다고 가정했습니다.

만약 손님이 객실에서 특정한 요구사항을 보내면

요구사항은 TABLET릴레이션에 저장되어

각 객실 전담 직원에게 전달되도록 구현하였습니다.

<다음>

다음은 웹 개발 과정에 대해 설명 드리겠습니다.

<다음>

지금 보시는 그림은 DB다이렉트팀의 웹 개발 설계도인데요,

각 파트별로 좀 더 자세히 살펴보겠습니다.

<다음>

우선 프론트엔드입니다.

HTML의 경우 다양한 호텔 사이트의 레이아웃을 참고하였고,

<다음>

반복적으로 사용되는 header와 footer 파일을 분리하여

jsp 파일 내에 include 하는 방식을 사용해 개발의 효율성을 높였습니다.

<다음>

또한 Html과 css를 분리해 작성하여 가독성을 높였고

부분적으로 부트스트랩을 사용하여 구현하였습니다.

Javascript의 경우 api활용과 다양한 함수들을 이용하여

이벤트 제어를 할 수 있었습니다.

<다음>

백엔드의 경우

강좌와

책을 보며 개발을 진행했는데요.

<다음>

백 엔드 개발에서는 MVC 패턴을 활용한

Java기반의 Spring 프레임워크를 사용했습니다.

MVC 패턴이란 웹을 구성하는 요소들을

비즈니스 로직으로 구성된 Model,

프론트 엔드를 뜻하는 View

마지막으로 Model들을 제어하고 관리하는 Controller로 분리하여 개발하는 방식을 뜻하는데요.

<다음>

실제로 저희가 개발한 폴더 구조를 보시면

<다음>

View와

<다음>

Controller 그리고

<다음>

Model이 분리되어 개발된 것을 확인하실 수 있는데요.

이는 개발 이후의 유지 보수에 큰 강점을 가지는 객체지향개발이라 말할 수 있습니다.

<다음>

다음으로 서버와 DB입니다.

저희 팀은 AWS의 EC2를 서버로 활용하였습니다.

또한 DB는 오라클을 사용하였고,

우분투 서버 위에 오라클DB를 다운받아

RDS와 연동이 가능하도록 하였습니다.

또한 백엔드 개발자들이 모두 로컬에서 DB에 쉽게 접근할 수 있도록

RDS를 사용하여 개발의 편의성을 높였고

AWS의 ROUTE 53을 사용하여 도메인을 등록하였습니다.

<다음>

마지막으로 저희 수업이 “데이터 베이스”인만큼,

DB 관련 코드를 어떻게 작성했는지 말씀드리려 하는데요.

우선 저희는 자바의 관계형 DB 프로그래밍을 돕는

MyBatis 개발 프레임워크를 활용하였습니다.

이를 통해 프로그램 코드와 DB 쿼리를 분리하여 개발했는데요.

보시는 것처럼 DB 쿼리들은 모두 Mapper라는 파일에

분리되어 저장되어 있습니다.

<다음>

개발을 진행하며 가장 중요하다고 생각했던 쿼리는

바로 “손님이 선택한 날짜에

비어 있는 객실을 반환하는 쿼리인데요.”

중첩질의를 활용하여 구성하였습니다.

<다음>

먼저 예약테이블의 튜플들과 날짜가 겹치지 않고,

<다음>

선택한 RoomType을 가지는 방을

<다음>

RoomNo 순으로 정렬하는 방식으로 구성하였습니다.

<다음>

특히, 날짜가 겹치지 않는지 여부를 확인하기 위해

<다음>

이미 예약된 check-in이 손님이 선택한 check-in,out 사이에 있거나

<다음>

예약 check-out이 손님 check-in,out 사이에 있거나

<다음>

예약 check-in,out 사이에 손님 check-in, out이 있는 경우를 걸러내는 방식을 사용하였습니다.

<다음>

지금까지 웹을 어떻게 구현했는지 설명드렸는데요.

다음으로 웹 애플리케이션 시연을 시작하겠습니다.

결제할 카드의 정보를 입력할 수 있습니다. 임의로 입력하도록 하겠습니다.

여기서 적립예정 마일리지와 최종 결제 금액을 확인할 수 있습니다.

동의를 모두 체크한 후 결제하기를 누르면 예약이 완료됩니다.

예약 완료 후 예약 확인/취소 페이지로 넘어가 예약을 확인하고 취소할 수 있으며 당일취소는 불가능합니다.

다음은 고객문의 페이지입니다.

고객들의 문의사항 목록을 확인할 수 있고 글쓰기를 눌러 문의사항을 입력할 수 있습니다.

문의사항의 내용은 본인이 작성한 글만 수정 또는 삭제할 수 있습니다.

다음은 마이페이지 화면입니다.

개인정보와 등급, 마일리지를 확인할 수 있고, 예약확인/취소, 문의내역을 확인할 수 있고 적립된 마일리지 별로 등급을 나누고 그에 따른 혜택을 표시해 놓았습니다.

호텔소개 화면입니다.

객실소개 화면입니다. 객실 종류를 선택하고 예약하기를 누르면 해당 객실을 바로 검색할 수 있습니다.

편의시설 화면입니다.

다음으로 관리자 페이지입니다.

레벨이 관리자로 세팅 된 아이디로 로그인을 하면 관리자 페이지로 연결됩니다.

직원 리스트에서 직원들의 정보를 확인할 수 있고 그 직원이 담당하는 층을 확인할 수 있습니다.

회원 리스트에서는 가입된 아이디의 레벨 및 고객정보를 확인할 수 있고 아이디를 클릭하면 고객 상세 정보와 예약 정보를 확인할 수 있습니다.

객실 리스트에는 호텔의 모든 객실 정보와 담당 직원을 확인할 수 있고,

예약버튼을 누르면 해당 객실의 예약 현황을 확인할 수 있습니다.

예약 완료의 경우는 예약정보와 결제 정보를 확인할 수 있고 예약 버튼을 클릭하면 현장예약 또는 비회원 예약이 가능합니다.

예약은 전에 보여드린 것과 유사하므로 생략하겠습니다.

예약 리스트에서도 예약 목록을 확인하실 수 있습니다.

문의 리스트에서는 회원들의 문의 내역을 확인할 수 있고 모든 해당 글을 삭제할 수도 있습니다.

마지막으로 객실 담당 직원페이지와 객실 테블릿 페이지를 보여드리겠습니다.

저희 호텔은 각 객실마다 테블릿을 설치하여 고객의 요청사항을 실시간으로 해결할 수 있도록 설계했습니다.

예시로 101호의 테블릿과 1층 담당 직원 페이지를 보여드리겠습니다.

직원 페이지는 레벨이 직원으로 세팅 된 아이디로 로그인하는 경우에 접속됩니다.

객실 태블릿은 왼쪽 화면과 같고, 페이지 내의 버튼을 누르면 담당 직원에게 전달됩니다.

담당 직원은 해당 일을 수행한 뒤 update버튼을 눌러 완료할 수 있습니다.

이상으로 시연을 마치도록 하겠습니다. 감사합니다.

지금부터 질문 받겠습니다.