1. 예약하기

<프론트엔드>

function AddOrder() { // 폼 제출 시에 실행됨

사용자가 입력한 값(예약날짜, 예약시간, 인원수, 요청사항, 사용자 이용약관 동의 여부)을 가져온다

if (사용자가 회원인가?) {

서버에게 DB에서 현재 로그인한 회원의 ID, 이름, 전화번호, 총 예약횟수, 이메일, 등급, 생년월일을 select 하라고 요청한다

var discount, notInput = "", sum = 0; // 할인율, 입력하지 않은 항목, 총 예약비용

이름과 전화번호는 먼저 입력란의 value로 지정, 후에 수정 가능함

// 회원등급 검사

if (rsv\_cnt > 60) {

grade = “다이아”;

}

else if (rsv\_cnt > 30) {

grade = "골드";

}

else if (rsv\_cnt > 10) {

grade = "실버";

}

switch(grade) {

case "실버":

discount = 5;

break;

case "골드":

discount = 10;

break;

case "다이아":

discount = 25;

}

// 회원 생일 판별

if (예약날짜 == 회원의 생일) { // 예약 날짜가 회원의 생일이라면

discount += 10; // 추가 할인

}

요청사항을 제외한 항목들 중 입력하지 않은 항목이 있다면 notInput에 열거한다.

if (notInput != "") {

alert(notInput+"을(를) 입력하지 않았습니다!");

}

else { // notInput에 내용이 없다면 (사용자가 모든 항목을 입력했다면)

if (예약날짜가 당일로부터 최소 7일 이상이면) {

if (약관에 동의했다면) {

총 예약비용 = 인원수 \* 29900 \* ((100-discount)/100);

총 예약횟수++;

예약날짜, 예약시간, 인원수, 요청사항, 등급, 총 예약비용, 총 예약횟수를 서버로 전달하여 insert를 요청한다.

alert("예약이 완료되었습니다. 예약정보는 입력하신 이메일로 전송되었습니다.");

}

else { // 동의하지 않았다면

alert("사용자 이용약관에 동의해주세요!");

}

}

else { // 아니라면

alert("예약날짜는 당일로부터 최소 7일 뒤의 날짜로 선택해주세요!");

}

}

}

else { // 비회원일 경우

사용자에게서 입력받은 이름, 전화번호,

var notInput = "", sum = 0; // 입력하지 않은 항목, 총 예약비용

요청사항을 제외한 나머지 항목이 입력이 되지 않았다면 입력하지 않은 항목들을 notInput에 열거한다.

if (notInput != "") {

alert(notInput에서 맨 뒤 반점 뺀 것+"을(를) 입력하지 않았습니다!");

}

else { // notInput에 내용이 없다면 (사용자가 모든 항목을 입력했다면)

var today = new Date();// 오늘 날짜

if (예약날짜가 당일로부터 최소 7일 이상이면) {

if (약관에 동의했다면) {

sum = people\_cnt \* 29900; // 총 예약비용 계산

while () {

var order\_num = 8자리 난수를 발생시켜 만든다.

서버로 주문번호를 보내어 기존에 DB에 저장되어 있는 주문번호와 중복이 있는지 검사할 것을 요청한다

if (서버에서 주문번호가 중복이 아니라고 할 때) {

주문번호, 이름, 전화번호, 예약날짜, 예약시간, 인원수, 요청사항을 서버로 전달하여 insert를 요청한다.

break;

}

}

if (insert에 성공했다면) {

alert("예약이 완료되었습니다. 예약정보는 입력하신 이메일로 전송되었습니다.");

}

}

else { // 동의하지 않았다면

alert("사용자 이용약관에 동의해주세요!");

}

}

else { // 아니라면

alert("예약날짜는 당일로부터 최소 7일 뒤의 날짜로 선택해주세요!");

}

}

}

}

<백엔드>

func MemberInfo() { // 회원모드에서 해당 회원의 아이디를 통하여 회원의 정보를 select

DB의 개인정보 테이블에서 현재 로그인한 회원의 이름, 전화번호, 이메일, 총 예약횟수, 등급, 생년월일을 프론트엔드로 전달

}

func CheckOrderNum() bool { // 비회원모드에서 해당 사용자의 예약 관리에 필요한 주문번호가 DB에 저장되어 있는 다른 주문번호와 중복인지 검사함

order\_num := 프론트엔드에서 받은 회원의 주문번호

DB의 비회원 예약정보 테이블에서 모든 비회원의 주문번호를 가져온다. (ResultSet에 저장)

for ResultSet을 전부 순회할 때 까지 {

if (order\_num == DB에 있는 주문번호) {

return false; // 중복임

}

}

return true; // 중복이 없음

}

func InsertOrder() bool { // DB에 예약을 추가한다. 예약 성공/실패 여부를 리턴한다.

if (사용자가 회원일 때) {

예약날짜, 예약시간, 인원수, 요청사항, 회원의 등급, 예약비용, 예약횟수, 총 예약비용를 프론트엔드에서 받아온다.

프론트엔드에서 받은 데이터 유효성 검사를 한다.

rsv\_cnt+1을 로그인한 회원의 총 예약횟수에 update

grade를 로그인한 회원의 등급에 update

DB의 회원 개인정보 테이블에서 로그인한 회원의 이름, 전화번호, 이메일을 가져온다.

DB의 회원 예약정보 테이블에 insert한다.

DB에서 회원의 이름, 등급, 전화번호, 예약날짜, 예약시간, 인원수, 요청사항, 총 예약비용이 적힌 메일을 로그인한 회원의 이메일로 전송한다.

}

else { // 사용자가 회원이 아닐 때

주문번호, 이름, 전화번호, 이메일, 예약날짜, 예약시간과, 인원수, 요청사항, 총 예약비용을 프론트엔드에서 받아온다.

프론트엔드에서 받은 데이터 유효성 검사

DB의 회원 예약정보 테이블에 insert한다.

DB에서 회원의 주문번호, 이름, 전화번호, 예약날짜, 예약시간, 인원수, 요청사항, 총 예약비용이 적힌 메일을 사용자가 입력한 이메일로 전송한다.

}

return true;

}