

# [TP-7] 최종 보고서 및 소스코드 [210점]

2024. 06. 05.

학번: 202102699 이름: 정민경

## 7-1. 구축된 DB의 각 테이블 내용 출력 (20점)

### 1. Customer Table

Table 구조								
COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE	NUMERIC_PRECISION	NUMERIC_SCALE	COLUMN_COMMENT	
1 cno	1 (null)	NO	varchar	(null)	(null)	(null)	(null)	
2 name	2 (null)	NO	varchar	(null)	(null)	(null)	(null)	
3 password	3 (null)	NO	varchar	(null)	(null)	(null)	(null)	
4 phoneno	4 (null)	NO	varchar	(null)	(null)	(null)	(null)	

Table 내용				
cno	name	password	phoneno	
1 C0	주인	1111	010-1111-1111	
2 C1	정민경	1234	010-1234-5678	
3 C2	홍길동	0000	010-9876-6789	
4 C3	오백이	5252	010-5025-0250	

### 2. Category Table

Table 구조					
COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE	NUMERIC_PRECISION
1 categoryName	1 (null)	NO	varchar	(null)	(null) (null)
2 noOfFoods	2 (null)	NO	int	10	0 (null)

Table 내용		
categoryName	noOfFoods	
1 디저트	3	
2 분식	7	
3 시즌메뉴(여름)	3	
4 추가 및 음료	5	
5 한식	5	

### 3. Food Table

Table 구조

COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE	NUMERIC_PRECISION	NUMERIC_SCALE	COLUMN_COMMENT
1 foodName	1 (null)		NO	varchar	(null)	(null)	(null) (null)
2 price	2 (null)		NO	int		10	0 (null)

Table 내용

foodName	price
1 공기밥	1000
2 과일빙수	12000
3 김밥	5000
4 김치찌개	8000
5 냉면	8000
6 돈까스	12000
7 된장찌개	8000
8 떡볶이	6000
9 라볶이	7000
10 맥주	5000
11 부대찌개	10000
12 사이다	2000
13 소주	5000
14 소프트콘	1200
15 순두부찌개	8000
16 우동	6000
17 치즈떡볶이	7000
18 치즈라볶이	8000
19 클라	2000
20 팥빙수	9000

### 4. Contain Table

Table 구조

COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE	NUMERIC_PRECISION	NUMERIC_SCALE	COLUMN_COMMENT
1 foodName	1 (null)		NO	varchar	(null)	(null)	(null) (null)
2 categoryName	2 (null)		NO	varchar	(null)	(null)	(null) (null)

Table 내용

foodName	categoryName
1 과일빙수	디저트
2 소프트콘	디저트
3 팥빙수	디저트
4 김밥	분식
5 냉면	분식
6 떡볶이	분식
7 라볶이	분식
8 우동	분식
9 치즈떡볶이	분식
10 치즈라볶이	분식
11 과일빙수	시즌메뉴(여름)
12 냉면	시즌메뉴(여름)
13 팥빙수	시즌메뉴(여름)
14 공기밥	추가 및 음료
15 맥주	추가 및 음료
16 사이다	추가 및 음료
17 소주	추가 및 음료
18 클라	추가 및 음료
19 김치찌개	한식
20 돈까스	한식
21 된장찌개	한식
22 부대찌개	한식
23 순두부찌개	한식

## 5. Cart Table

Table 구조

COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE	NUMERIC_PRECISION	NUMERIC_SCALE	COLUMN_COMMENT
1 id	1 (null)	NO	int		10	0 (null)	
2 orderDateTime	2 (null)	YES	timestamp	(null)	(null) (null)		
3 cno	3 (null)	YES	varchar	(null)	(null) (null)		

Table 내용

id	orderDateTime	cno
1	2024-06-11 22:25:19.0	C3
2	2024-05-12 22:25:19.0	C3
3	2024-06-08 22:25:19.0	C1

## 6. OrderDetail Table

Table 구조

COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE	NUMERIC_PRECISION	NUMERIC_SCALE	COLUMN_COMMENT
1 itemNo	1 (null)	NO	int		10	0 (null)	
2 id	2 (null)	NO	int		10	0 (null)	
3 quantity	3 (null)	YES	int		10	0 (null)	
4 totalPrice	4 (null)	YES	int		10	0 (null)	
5 foodName	5 (null)	YES	varchar	(null)	(null) (null)		

Table 내용

itemNo	id	quantity	totalPrice	foodName
1	1	1	3	15000 김밥
2	1	2	2	16000 김치찌개
3	1	3	2	2400 소프트콘
4	2	1	2	4000 콜라
5	2	2	1	5000 소주
6	2	3	1000	12000000 과일빙수
7	3	1	2	4000 사이드

## 7-2. 구현된 시스템 상세 설명 (40점)

구현 환경
- 서버 : apache24
- 사용언어 : php, css, js (php파일 안에서 html, js 사용)

### 각 모듈 별 주요 알고리즘 설명

#### db\_connect.php

- db와 연결해주는 모듈
- db를 사용하는 모든 페이지에서 include해 사용한다.

#### page/login.php

- login 화면
- 별 다른 기능은 없으며, 로그인화면을 만든 화면이다.

#### page/main.php

- 사용자(관리자)가 로그인 후 기능을 사용하는 메인화면
- 서비스의 뼈대가 되는 화면으로, 이 화면에서 모든 컴포넌트의 배치를 구성하였으며, 사용자가 아무런 입력을 하지 않아도 보이는 화면이다.
- 이 페이지의 코드를 장바구니, 주문내역, 통계정보 검색화면을 제외하고 나머지 페이지에서 거의 동일하게 사용하며, main-content 부분만 각 페이지의 목적에 맞춰 변경되어 사용된다.
- 메인화면에서는 모든 음식정보가 카테고리로 나뉘어 display된다.
- 왼쪽 상단의 로고를 누르면 메인화면으로 돌아오며, 왼쪽 사이드바에 금액설정기능이 있으며, 카테고리 선택이 가능하다. 또한 장바구니, 주문내역조회버튼이 있으며, 관리자로 로그인 시 통계정보검색버튼이 존재한다.
- 상단바에는 검색가능한 바가 있으며, 우측 상단에 로그인한 사용자 이름과 번호가 display된다.
- 음식이 display될 때 사진, 음식이름, 가격이 함께 보이게 되며, 클릭하여 장바구니로의 추가가 가능하다.

#### search\_results.php

- 사용자가 음식 이름을 키워드로 검색 시에 사용하는 기능 및 화면
- 검색바에 특정 키워드를 입력 후 검색하면, 입력받은 키워드가 포함된 음식이 display 된다.

#### search\_price.php

- 사용자가 가격 범위를 정해서 음식 검색 시에 사용하는 기능 및 화면
- 왼쪽 사이드바에 가격 설정 후 확인을 누르게 되면 설정한 가격 범위 내의 음식이 조회된다.
- 사용자가 최소, 최대금액을 설정할 수 있으며, 최소금액 or 최대금액 중 하나만 설정 하여 검색도 가능하다. 이때 설정되지 않은 금액은 기본값으로 설정되어 검색된다. (min\_price : 0원, max\_price : 1,000,000원)

#### search\_category.php

- 사용자가 카테고리를 선택하여 음식 검색 시에 사용하는 기능 및 화면
- 사용자가 왼쪽 사이드바에 있는 카테고리를 선택 시 선택된 카테고리에 속해있는 음식이 display된다.
- 카테고리 선택으로 음식을 검색할 경우 최대 한 가지의 카테고리만 선택 가능하다.

#### page/cart.php

- 사용자 본인의 장바구니 내역을 조회하는 기능 및 화면
- 사용자가 음식을 담기 전 장바구니를 조회했다면, 장바구니가 비어있다는 문구가 존재한다.
- 사용자가 음식을 담은 후 장바구니를 조회한 경우, 음식이름, 수량, 1개당 가격, 음식 하나에 대한 총 가격이 조회되며, 화면 하단에는 결제해야하는 총 금액과 결제하기 버튼이 존재한다.
- 만약 동일 음식을 여러번에 걸쳐 장바구니에 담았더라도, 조회는 여러 줄이 아닌 한 줄로 조회된다.
- 결제하기 버튼을 누르면 결제가 100% 성공하며, 그 후 장바구니 조회 시 장바구니는 비어있게 된다.

#### page/order\_history.php

- 사용자 본인의 주문 내역을 조회하는 기능 및 화면
- 주문내역 조회의 기본 내역은 사용자가 주문했던 전체 내역이 조회되며, 조회하고자 하는 날짜를 지정하면 그 기간 내 주문내역이 조회된다.
- 주문내역을 조회하면 주문했던 날짜+시간과 음식이름, 수량, 1개당 가격, 음식 하나에 대한 총 가격이 함께 조회된다.
- 주문내역은 최신순으로 조회가 된다.

#### page/admin\_stats.php

- 관리자(C0)만이 사용가능한 기능이며, 특정년도의 판매금액과 카테고리 별 인기있는 음식을 검색하는 기능 및 화면
- 기간 내 판매금액 조회기능은 관리자가 연도를 입력하면 그 연도에 판매된 금액이 월별 + 카테고리별로 조회된다. 만약 연도를 입력하지 않은 경우 조회한 그 시점의 연도를 기준으로 조회가능하다.
- 카테고리 별 인기메뉴 조회기능은 판매 시작시점부터 관리자가 조회하고자 화면을 킨 조회시점까지의 총 판매개수가 많은 음식을 조회할 수 있다. 이때 각 카테고리 별 가장 많이 팔린 음식 1개씩과, 각각 몇 개가 팔렸는지에 대한 정보를 확인할 수 있다.

#### login\_process.php

- login시 비밀번호 일치여부, 등록된 회원인지 확인과 같은 로그인과 관련된 처리
- 현재 로그인하고자 하는 사용자가 등록되어있는 사용자인지 확인한다. 등록되어있지 않은 사용자면 “존재하지 않는 사용자입니다.”라는 alert 출력.
- 존재하는 사용자이지만, 비밀번호가 일치하지 않는 경우 “비밀번호가 잘못되었습니 다.”라는 alert 출력.
- 그 외의 경우 성공적으로 로그인되며 main화면으로 이동.

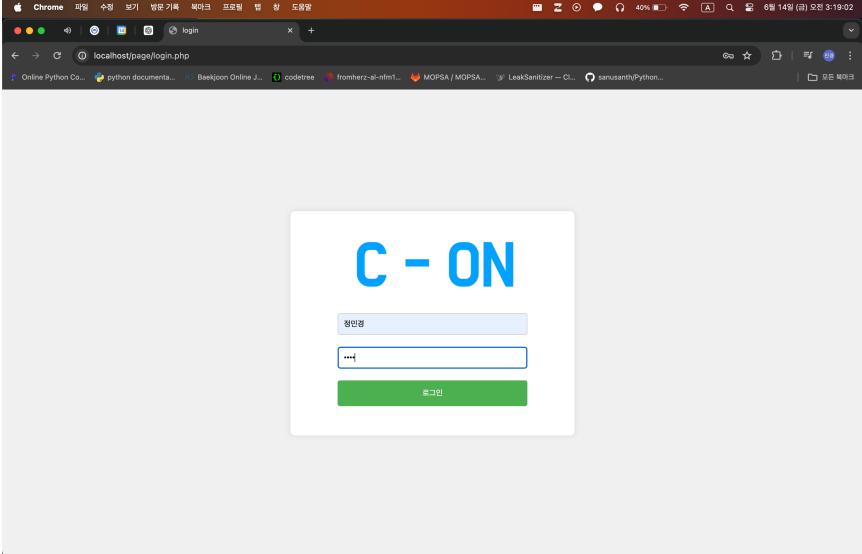
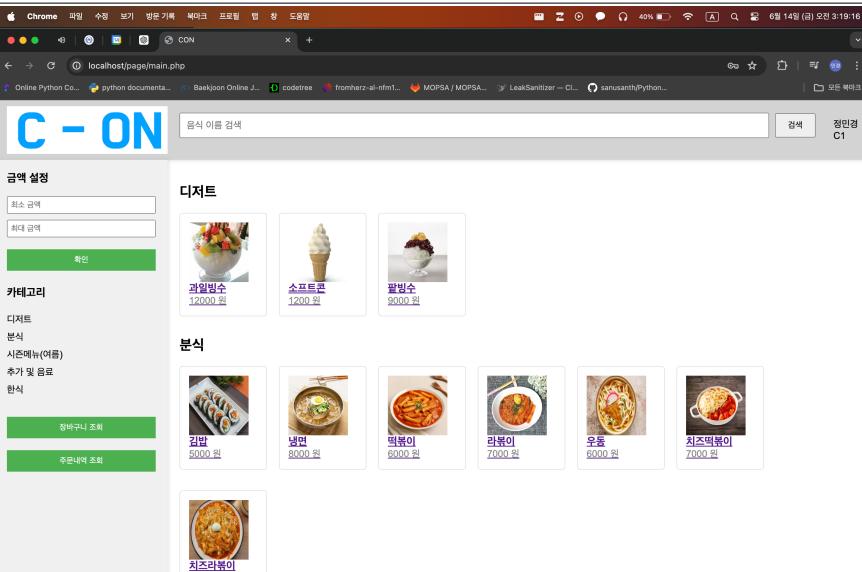
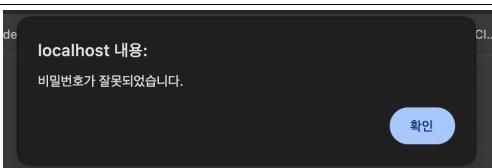
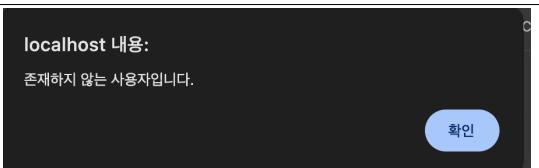
#### **add\_food\_to\_cart.php**

- 사용자가 음식을 선택하여 장바구니에 담는 기능
- 사용자가 장바구니에 추가하고자 하는 음식 선택 후 수량 선택하여 장바구니 담기를 눌렀을 때 처리되는 기능이 구현되어 있다.

#### **cart\_to\_history.php**

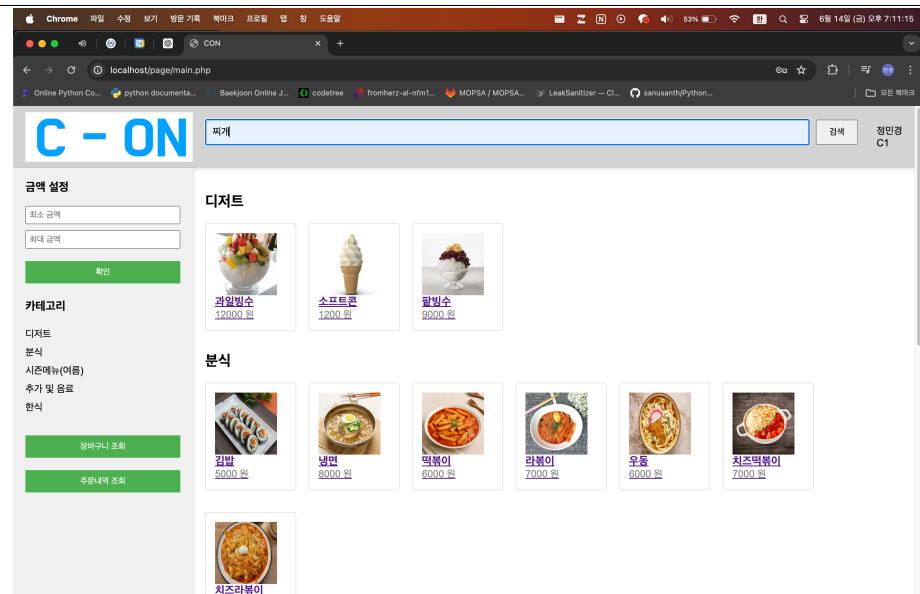
- 사용자가 장바구니 조회 화면에서 결제하기를 눌렀을 때 결제가 이루어지며, 장바구니에서 정보가 사라지고 주문내역에 표시되는 기능
- 사용자가 장바구니에서 결제하기버튼을 누른 후 결제가 이루어졌을 때 장바구니에서 사라지며 주문내역에 저장되는 기능이 구현되어 있다.

### 7-3. User Manual (100점)

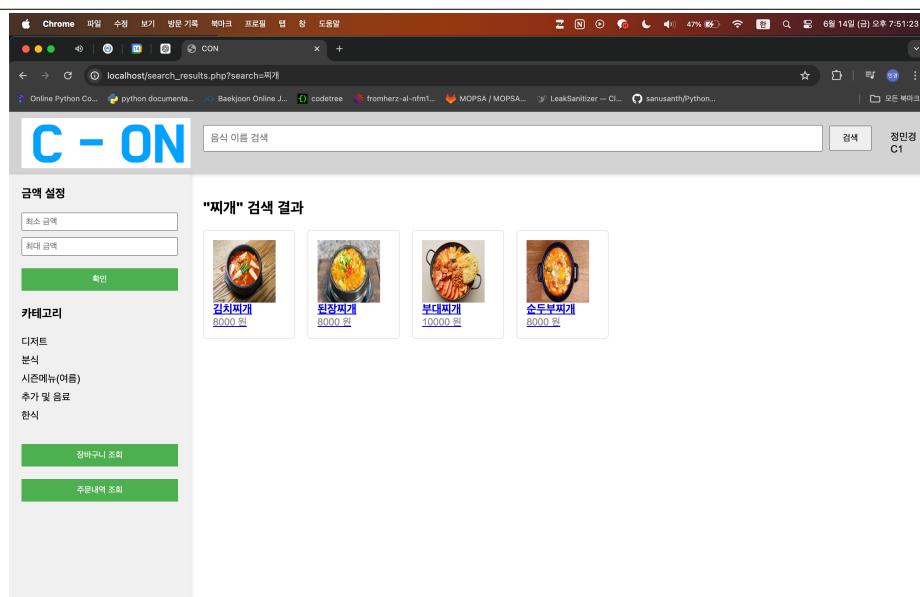
로그인 기능	
<ul style="list-style-type: none"> <li>사용자가 서비스 사용하기 전 로그인을 하는 기능이다.</li> <li>회원가입은 제공되지 않으며, 미리 가입된 유저의 이름 + 비밀번호로 로그인한다.</li> <li>등록되지 않은 유저나 잘못된 비밀번호로 로그인을 시도하면 에러메세지가 뜨며, 로그인에 성공하면 메인화면이 보여진다.</li> </ul>	
	
<p style="text-align: center;">로그인 화면에서 아이디와 비밀번호를 입력한 화면</p>	
	
<p style="text-align: center;">로그인 성공 시 C-ON서비스의 메인화면으로 이동</p>	
 <p>잘못된 비밀번호를 입력한 경우</p>	 <p>등록되지 않은 유저로 로그인 시도한 경우</p>

## 음식 검색 기능

- 음식검색은 총 3가지 기능이 제공되며, 키워드와 금액간의 중복 검색은 가능하지만 그 외의 검색기능은 중첩시켜 검색이 불가능하다.
  - 키워드 검색기능 : 주문화면의 상단바에 있는 검색바에 키워드를 입력해 음식 검색이 가능하다.
  - 금액 검색기능 : 왼쪽 사이드바에 존재하는 금액 설정란에 최소, 최대금액 설정 후 확인버튼을 누르게 되면, 설정한 금액 내에 속하는 음식이 카테고리별로 검색이 된다. 최소, 최대금액은 각각 기본값이 1원, 1,000,000원이므로 입력을 하지 않은 값은 각 금액의 기본값으로 설정되어 검색된다.
  - 카테고리 검색기능 : 왼쪽 사이드바에 존재하는 카테고리를 누르게 되면, 사용자가 선택한 카테고리에 속해있는 음식이 검색된다.



1-1. 메인화면에서 음식 키워드로 검색 ("찌개" 키워드 입력)



1-2. 메인화면에서 음식 키워드로 검색 ("찌개" 검색 결과)

Chrome 파일 수정 보기 방문 기록 북마크 프로필 템 창 도움말

localhost/page/main.php

C - ON

음식 이름 검색

검색 정민경 C1

금액 설정

5000  
10000

확인

카테고리

디저트

과일빙수 12000 원  
소프트콘 1200 원  
팥빙수 9000 원

분식

김밥 5000 원  
냉면 8000 원  
떡볶이 6000 원  
리볶이 7000 원  
우동 6000 원  
치즈떡볶이 7000 원

한식

장비구니 조회  
주문내역 조회

치즈라볶이

2-1. 메인화면에서 **금액 설정**으로 검색 (5,000~10,000원으로 설정)

Chrome 파일 수정 보기 방문 기록 북마크 프로필 템 창 도움말

localhost/search\_price.php?min\_price=5000&max\_price=10000

C - ON

음식 이름 검색

검색 정민경 C1

금액 설정

최소 금액  
최대 금액

확인

카테고리

디저트

팥빙수 9000 원

분식

김밥 5000 원  
냉면 8000 원  
떡볶이 6000 원  
리볶이 7000 원  
우동 6000 원  
치즈떡볶이 7000 원

한식

장비구니 조회  
주문내역 조회

치즈라볶이

2-2. 메인화면에서 **금액 설정**으로 검색 (5,000~10,000원 검색 결과)

The screenshot shows the main page of the C-ON food ordering system. On the left, there's a sidebar with '금액 설정' (Price Setting) containing input fields for '최소 금액' (Min Price) and '최대 금액' (Max Price), and a green '확인' (Check) button. Below it is a section titled '카테고리' (Category) with options: 디저트 (Dessert), 분식 (Beverage), 시즌메뉴(여름) (Summer Season Menu), 추가 및 음료 (Add-ons and Beverages), and 한식 (Korean Food). There are two green buttons: '장바구니 조회' (View Cart) and '주문내역 조회' (View Order History). The main content area has a search bar at the top with placeholder '음식 이름 검색' (Search food name) and a '검색' (Search) button. The results for '디저트' (Dessert) are displayed in three columns: 과일빙수 (Fruit Shaved Ice) at 12000 원, 소프트콘 (Soft Cone) at 1200 원, and 팥빙수 (Red Bean Shaved Ice) at 9000 원. Below this is a section titled '분식' (Beverage) showing six items: 김밥 (Kimchi Rice) at 5000 원, 냉면 (Cold Noodles) at 8000 원, 떡볶이 (Tteokbokki) at 6000 원, 라볶이 (Rabokki) at 7000 원, 우동 (Udon) at 6000 원, and 치즈떡볶이 (Cheese Tteokbokki) at 7000 원. A single item, 치즈라볶이 (Cheese Rabokki) at 8000 원, is shown below the main grid.

3-1. 메인화면에서 카테고리 선택으로 검색 (카테고리 선택 전)

This screenshot shows the same main page as above, but the search results have changed. The category '분식' (Beverage) is selected, so the results are displayed in a single column. It includes the six items from the previous screenshot: 김밥 (5000 원), 냉면 (8000 원), 떡볶이 (6000 원), 라볶이 (7000 원), 우동 (6000 원), and 치즈떡볶이 (7000 원). The item 치즈라볶이 (8000 원) is no longer visible.

3-2. 메인화면에서 카테고리 선택으로 검색 (“분식” 선택 결과)

Apple Chrome 파일 수정 보기 방문 기록 북마크 프로필 템 창 도움말

localhost/search\_results.php?search=%찌개

Online Python Co... python documenta... Baekjoon Online J... codetree fromherz-al-nfm1... MOPSA / MOPSA... LeakSanitizer — Cl... sanusanth/Python...

C - ON

음식 이름 검색

금액 설정

최소 금액  
9000

확인

카테고리

디저트

분식

시즌메뉴(여름)

추가 및 음료

한식

장바구니 조회

주문내역 조회

"찌개" 검색 결과

김치찌개 8000 원

된장찌개 8000 원

부대찌개 10000 원

순두부찌개 8000 원

localhost/search\_results.php?search=%찌개#

4-1. “찌개” 키워드로 검색 후 금액 0~9,000원으로 검색 (금액 설정 전)

Apple Chrome 파일 수정 보기 방문 기록 북마크 프로필 템 창 도움말

localhost/search\_price.php?min\_price=&max\_price=9000&search=%찌개

Online Python Co... python documenta... Baekjoon Online J... codetree fromherz-al-nfm1... MOPSA / MOPSA... LeakSanitizer — Cl... sanusanth/Python...

C - ON

음식 이름 검색

금액 설정

최소 금액

최대 금액

확인

디저트

이 카테고리에 속한 음식이 없습니다.

분식

이 카테고리에 속한 음식이 없습니다.

시즌메뉴(여름)

이 카테고리에 속한 음식이 없습니다.

추가 및 음료

이 카테고리에 속한 음식이 없습니다.

한식

장바구니 조회

주문내역 조회

김치찌개 8000 원

된장찌개 8000 원

순두부찌개 8000 원

localhost/search\_price.php?min\_price=&max\_price=9000&search=%찌개#

4-2. “찌개” 키워드로 검색 후 금액 0~9,000원으로 검색 (금액 설정 후)

Chrome 파일 수정 보기 방문 기록 북마크 프로필 템 창 도움말

localhost/search\_price.php?min\_price=5000&max\_price=9000

Online Python Co... python documenta... Baekjoon Online J... codetree fromherz-al-nfm1... MOPSA / MOPSA... LeakSanitizer — Cl... sanusanth/Python...

검색 정민경 C1

# C - ON

찌개

금액 설정

최소 금액  
최대 금액

확인

카테고리

디저트  
분식  
시즌메뉴(여름)  
추가 및 음료  
한식

장바구니 조회  
주문내역 조회

디저트

팥빙수  
9000 원

분식

김밥 5000 원	냉면 8000 원	떡볶이 6000 원	라면 7000 원	우동 6000 원

치즈라면이

5-1. **금액** 5,000~9,000원으로 검색 후 **키워드**로 검색 (키워드 검색 전)

Chrome 파일 수정 보기 방문 기록 북마크 프로필 템 창 도움말

localhost/search\_results.php?search=%22찌개%22&min\_price=5000&max\_price=9000

Online Python Co... python documenta... Baekjoon Online J... codetree fromherz-al-nfm1... MOPSA / MOPSA... LeakSanitizer — Cl... sanusanth/Python...

검색 정민경 C1

# C - ON

음식 이름 검색

금액 설정

최소 금액  
최대 금액

확인

카테고리

디저트  
분식  
시즌메뉴(여름)  
추가 및 음료  
한식

장바구니 조회  
주문내역 조회

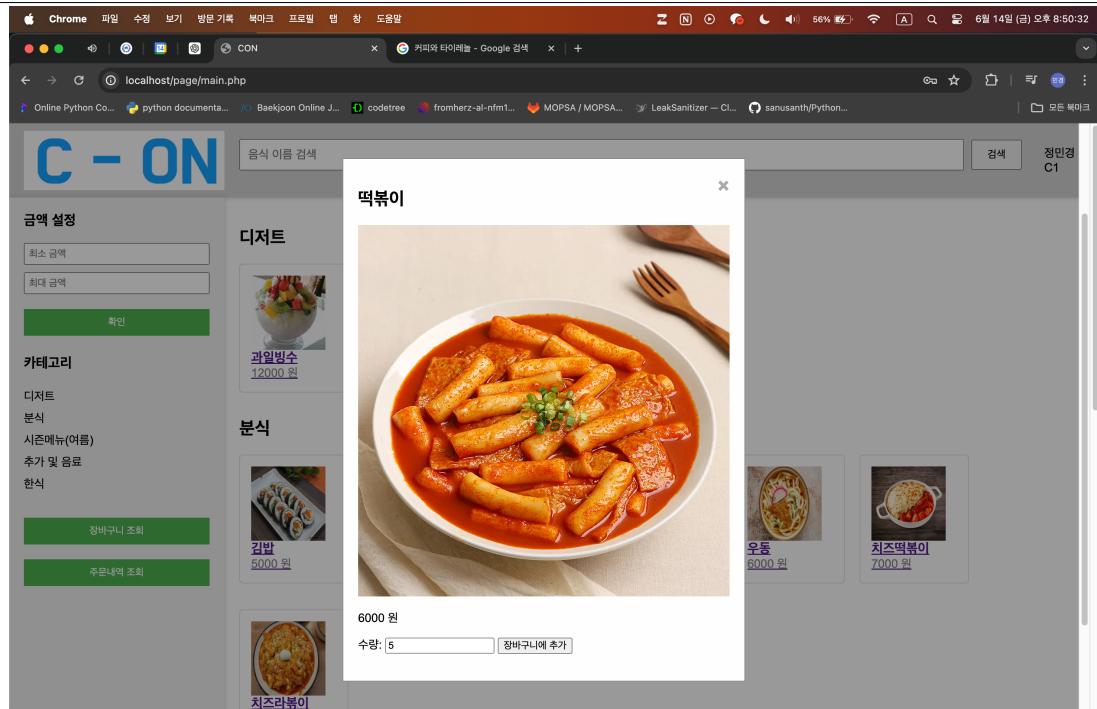
"찌개" 검색 결과

김치찌개 8000 원	된장찌개 8000 원	순두부찌개 8000 원

5-2. **금액** 5,000~9,000원으로 검색 후 **키워드**로 검색 ("찌개" 키워드 검색 결과)

### 장바구니 담기 및 조회 기능

- 사용자는 음식을 장바구니에 담을 수 있으며, 각 음식을 선택 시 수량을 선택하여 장바구니에 담을 수 있다.
- 사용자는 장바구니 조회 화면의 결제버튼을 클릭하기 전까지 하나의 주문에 여러개의 음식을 담을 수 있다.
- 왼쪽 사이드바에 있는 “장바구니 조회” 버튼을 클릭하면 현재 진행중인 주문에 대한 장바구니 내역을 확인 가능하며 제공되는 장바구니 화면은 다음과 같다.
  - 장바구니에 담은 음식 내역 (음식, 수량, 음식의 개당 가격, 그 음식의 총 가격)
  - 장바구니에 담은 음식들의 총 가격 (총 결제 가격)



1-1. 음식 선택하여 수량 선택 후 **장바구니 추가** (장바구니 추가 전)

### localhost 내용:

장바구니에 추가되었습니다!

확인

1-2. 음식 선택하여 수량 선택 후 **장바구니 추가** (장바구니 추가 클릭)

The screenshot shows a web browser window with the following details:

- Address Bar:** localhost/page/cart.php
- Title Bar:** 장바구니 조회
- Header:** C - ON, 장바구니 조회, 정인경 C1
- Content:**
  - Table:** A table showing the items in the cart:

음식 이름	수량	1개당 가격	총 가격
떡볶이	5	6000 원	30000 원
팥빙수	4	9000 원	36000 원
맥주	1	5000 원	5000 원
부대찌개	3	10000 원	30000 원
  - Text:** 총 금액: 101000 원
  - Buttons:** 결제하기
- Bottom Note:** 2. “장바구니 조회” 버튼 클릭하여 하나의 주문에 여러 가지 음식 추가 확인

## 주문 결제 및 주문내역 조회 기능

- 사용자가 주문 시 장바구니 조회 화면의 하단에 있는 결제하기 버튼을 클릭하면 주문이 완료된다.
- 주문을 하게 되면 결제가 성공하였다는 메시지와 함께 장바구니 조회 화면에서는 즉시 사라지며, 주문내역조회에 내역이 추가된다.
- 왼쪽 사이드바에 있는 “주문내역 조회” 버튼을 클릭하면 주문한 모든내역이 조회되며, 기간을 설정하여 주문내역을 검색할 수 있다.
- 주문내역 조회 시 아래의 정보가 제공된다.
  - 주문날짜
  - 각 주문 당 음식이름, 수량, 개당 가격, 총가격
- 날짜를 지정해서 주문내역을 조회하고자 한다면, 시작 날짜와 종료 날짜를 지정하여 검색하면 시작, 종료날짜 사이에 주문한 내역이 조회된다. 만일 다시 전체 주문을 조회하고자 한다면 “초기화” 버튼 클릭 시 전체 주문내역이 조회된다.

The screenshot shows a web browser window with the URL [localhost/page/order\\_history.php](http://localhost/page/order_history.php). The page title is "주문내역 조회".  
The interface includes:

- A header bar with the C-ON logo and user information (정민경 C1).
- Search filters for "시작 날짜" (Start Date) and "종료 날짜" (End Date), both set to "연도, 월, 일". Buttons for "검색" (Search) and "초기화" (Reset) are present.
- Three sections of order history:
  - 2024-06-14 20:51:35:** Includes a table with one row: 음식 이름 (떡볶이), 수량 (6), 1개당 가격 (6000 원), and 총 가격 (36000 원).
  - 2024-06-14 16:08:52:** Includes a table with three rows: 음식 이름 (소프트콘, 팔빙수), 수량 (5, 1), 1개당 가격 (1200 원, 9000 원), and 총 가격 (6000 원, 9000 원).
  - 2024-06-14 12:46:24:** Includes a table with one row: 음식 이름 (과일빙수), 수량 (1), 1개당 가격 (12000 원), and 총 가격 (12000 원).
- A note at the bottom left: "1-1. 결제 전 주문내역 조회 화면 (결제전이기 때문에 내역이 조회되지 않음.)"

**1-1. 장바구니 조회 화면**

음식 이름	수량	1개당 가격	총 가격
떡볶이	5	6000 원	30000 원
팥빙수	4	9000 원	36000 원
맥주	1	5000 원	5000 원
부대찌개	3	10000 원	30000 원
냉면	2	8000 원	16000 원
콜라	1	2000 원	2000 원
돈까스	1	12000 원	12000 원
소프트콘	10	1200 원	12000 원

총 금액: 143000 원      결제하기

**1-2. 결제하려는 내역**

확인

**1-3. 결제 완료 메세지**

장바구니가 비어 있습니다.

**1-4. “결제하기” 버튼 클릭 후 장바구니 조회 화면에서 사라짐.**

**C - ON** 주문내역 조회 정민경 C1

시작 날짜: 연도. 월. 일. 종료 날짜: 연도. 월. 일. 검색 초기화

2024-06-14 20:53:13

음식 이름	수량	1개당 가격	총 가격
떡볶이	5	6000 원	30000 원
팥빙수	4	9000 원	36000 원
맥주	1	5000 원	5000 원
부대찌개	3	10000 원	30000 원
냉면	2	8000 원	16000 원
콜라	1	2000 원	2000 원
돈까스	1	12000 원	12000 원
소프트콘	10	1200 원	12000 원

2024-06-14 20:51:35

음식 이름	수량	1개당 가격	총 가격
떡볶이	6	6000 원	36000 원

1-5. 주문을 완료한 내역이 주문내역 조회화면에 제공됨.

2024-06-08 22:25:19

음식 이름	수량	1개당 가격	총 가격
소프트콘	2	1200 원	2400 원
과일빙수	1000	12000 원	12000000 원

2. 주문내역 중 24.06.01 ~ 24.06.09 기간 내에 주문한 내역 조회

Chrome 파일 수정 보기 방문 기록 북마크 프로필 템 창 도움말

localhost/page/order\_history.php

주문내역 조회

정민경  
C1

시작 날짜: 연도. 월. 일. 종료 날짜: 연도. 월. 일. 검색 초기화

2024-06-14 20:53:13

음식 이름	수량	1개당 가격	총 가격
떡볶이	5	6000 원	30000 원
팥빙수	4	9000 원	36000 원
맥주	1	5000 원	5000 원
부대찌개	3	10000 원	30000 원
냉면	2	8000 원	16000 원
콜라	1	2000 원	2000 원
돈까스	1	12000 원	12000 원
소프트콘	10	1200 원	12000 원

2024-06-14 20:51:35

음식 이름	수량	1개당 가격	총 가격
떡볶이	6	6000 원	36000 원

3. “초기화” 버튼 클릭 시 전체 주문내역 조회

## 통계 기능 (관리자)

- 서비스에 관리자 계정 (고객번호 C0) 으로 로그인 시 다른 사용자와는 다르게 “판매 통계 조회” 기능이 제공된다.
- 판매 통계 조회기능은 아래의 2가지 기능이 제공된다.
  - 기간 내 판매금액 조회 내용
    - 해당연도의 월별, 카테고리 별 판매 금액을 조회할 수 있다.
    - 연도를 설정하지 않은 경우 조회기능을 사용한 시점의 연도로 자동 조회가 되며, 관리자가 연도 설정 시 그 연도의 정보가 조회된다.
  - 카테고리 별 인기메뉴 조회 내용
    - 각 카테고리 별로 판매시작시점부터 조회기능을 사용한 시점까지의 총 판매개수 가 많은 음식을 누적판매개수와 함께 제공한다.

The screenshot shows the C-ON management system interface. At the top, there's a navigation bar with links like 'Home', 'Logout', 'Dashboard', 'Sales Statistics', 'Menu Management', 'Customer Management', and 'Report'. Below the navigation, there's a search bar and a user profile section for 'Owner C0'.

The main content area is divided into sections:

- 금액 설정**: A form for setting minimum and maximum amounts with a '확인' (Confirm) button.
- 카테고리**: A sidebar with categories: 디저트, 분식, 시즌메뉴(여름), 추가 및 음료, 한식. Buttons for '장바구니 조회', '주문내역 조회', and '판매 통계 조회' are available.
- 디저트**: Displays three dessert items: 과일빙수 (12000 원), 소프트콘 (1200 원), and 팥빙수 (9000 원).
- 분식**: Displays six breakfast items: 김밥 (5000 원), 냉면 (8000 원), 떡볶이 (6000 원), 라볶이 (7000 원), 유통 (6000 원), and 치즈떡볶이 (1000 원).

At the bottom, there's a note: "1. 관리자 계정으로 로그인 시의 메인화면 (왼쪽 사이드바에 “판매 통계 조회” 버튼 추가)" which translates to "1. When logging in with a manager account, add the ‘Sales Statistics’ button to the left sidebar".

**기간 내 판매금액 조회내용**

조회					
2024년 조회 결과					
월	디저트	분식	시즌메뉴(여름)	추가 및 음료	한식
1월	0원	0원	0원	0원	0원
2월	0원	0원	0원	0원	0원
3월	0원	0원	0원	0원	0원
4월	0원	0원	0원	0원	0원
5월	0원	0원	0원	5000원	16000원
6월	12086400원	97000원	12082000원	15000원	42000원
7월	0원	0원	0원	0원	0원
8월	0원	0원	0원	0원	0원
9월	0원	0원	0원	0원	0원
10월	0원	0원	0원	0원	0원
11월	0원	0원	0원	0원	0원
12월	0원	0원	0원	0원	0원

**카테고리 별 인기메뉴 조회**

이 정보는 판매 시작시점부터 조회시점까지의 총 판매개수가 많은 음식을 보여줍니다.

디저트	분식	시즌메뉴(여름)	추가 및 음료	한식
과일빙수	떡볶이	과일빙수	콜라	부대찌개
1001개	11개	1001개	3개	3개

## 2. “판매 통계 정보” 버튼 클릭 시 제공되는 화면

**기간 내 판매금액 조회내용**

조회					
2020년 조회 결과					
월	디저트	분식	시즌메뉴(여름)	추가 및 음료	한식
1월	0원	0원	0원	0원	0원
2월	0원	0원	0원	0원	0원
3월	0원	0원	0원	0원	0원
4월	0원	0원	0원	0원	0원
5월	0원	0원	0원	0원	0원
6월	0원	0원	0원	0원	0원
7월	0원	0원	0원	0원	0원
8월	0원	0원	0원	0원	0원
9월	0원	0원	0원	0원	0원
10월	0원	0원	0원	0원	0원
11월	0원	0원	0원	0원	0원
12월	0원	0원	0원	0원	0원

**카테고리 별 인기메뉴 조회**

이 정보는 판매 시작시점부터 조회시점까지의 총 판매개수가 많은 음식을 보여줍니다.

디저트	분식	시즌메뉴(여름)	추가 및 음료	한식
과일빙수	떡볶이	과일빙수	콜라	부대찌개
1001개	11개	1001개	3개	3개

**3. 판매금액을 조회하고자하는 연도 “2020년”으로 설정해 조회**

#### 7-4. 소스 코드 설명 (50점)

- 각 모듈 별 설명과 코드를 첨부하여 설명한다.
- 개발의 노하우가 부족하여 웹 화면(html) 구성 코드는 동일코드임에도 계속적으로 붙여넣어 사용하여 가독성이 떨어진다. 따라서 본 보고서에 첨부하는 코드는 각 모듈의 중요 코드부분만을 첨부한다. (css는 따로 설명하지 않는다.)

#### db\_connect.php

- db와 연결해주는 모듈
- mysql을 사용하므로, 연결 시 필요한 정보들을 작성하여 mysqli 객체를 하나 생성해 사용한다.

```
1  <?php
1  // info about MySQL DB connection
2  $host = 'localhost'; // DB host
3  $username = 'dt202102699'; // user name
4  $password = 'jmink021120!!'; // password
5  $database = 'term_project'; // DB name
6
7  try {
8      // create MySQLi connection
9      $conn = new mysqli($host, $username, $password, $database);
10
11     // check connection
12     if ($conn->connect_error) {
13         throw new Exception("Connection failed: " . $conn->connect_error);
14     }
15 } catch (Exception $e) {
16     echo "Failed Connection: " . $e->getMessage();
17 }
18 ?>
19
20 </body>
21 </html>
22
```

#### page/login.php

- login 화면
- 별 다른 기능은 없으며, 로그인화면의 구조를 작성한 html이다.

```
1  <!DOCTYPE html>
1  <html lang="ko">
2  <head>
3      <meta charset="UTF-8">
4      <meta name="viewport" content="width=device-width, initial-scale=1.0">
5      <title>login</title>
6      <link rel="stylesheet" href="../style/styles.css">
7  </head>
8  <body>
9      <div class="container">
10         
11         <form action="../login_process.php" method="post"> <!-- login_process.php로 데이터를 전송 -->
12             <input type="text" id="username" name="username" placeholder="ID">
13             <input type="password" id="password" name="password" placeholder="Password">
14             <button type="submit">로그인</button>
15         </form>
16     </div>
17 </body>
18 </html>
19
```

## page/main.php

- 사용자(관리자)가 로그인 후 기능을 사용하는 메인화면으로, 이 화면에서 모든 컴포넌트의 배치를 구성하였으며, 사용자가 아무런 입력을 하지 않아도 보이는 화면이다.
- 이 페이지의 코드를 장바구니, 주문내역, 통계정보 검색화면을 제외하고 나머지 페이지에서 거의 동일하게 사용하며, 특정 부분만 각 페이지의 목적에 맞춰 변경되어 사용된다.
- db와의 연결은 db\_connect.php를 include 하여 사용하였고, 음식 사진을 사용하기 위해 이름과 이미지 파일을 매핑하는 배열을 사용하였다.
- 3가지 검색에 대한 화면은 main.php 페이지와 거의 동일하며, 음식이 보여지는 main-content만 달라지므로 코드에 대한 설명은 본 코드와 다른 부분에 대해서만 설명하고자 한다.

```
1 <?php
1 session_start();
2 include '../db_connect.php';
3
4 // 카테고리 정보 조회
5 $category_sql = "SELECT * FROM category;";
6 $result_categories = $conn->query($category_sql); // 카테고리 정보 저장
7
8 // 음식 이름에 대한 이미지 파일명을 매핑하는 배열
9 $image_map = [
10    '팥 빙수' => 'red_bean.png',
11    '과일 빙수' => 'fruit_ice.png',
12    '소프트콘' => 'softcon.png',
13    '김밥' => 'riceroll.png',
14    '냉면' => 'cold_noodle.png',
15    '떡볶이' => 'tteokbokki.png',
16    '라볶이' => 'rabokki.png',
17    '우동' => 'udong.png',
18    '치즈 떡볶이' => 'cheeze_tteokbokki.png',
19    '치즈 라볶이' => 'cheeze_rabokki.png',
20    '공기밥' => 'rice.png',
21    '맥주' => 'beer.png',
22    '사이다' => 'between.png',
23    '소주' => 'soju.png',
24    '콜라' => 'coke.png',
25    '김치찌개' => 'kimchi_soup.png',
26    '돈까스' => 'fried_pork.png',
27    '된장찌개' => 'soybean_soup.png',
28    '부대찌개' => 'Budaejjigae.png',
29    '순두부찌개' => 'Tofu_Stew.png',
30
31 ];
32 ?>
```

[ db와 연결하는 모듈 include 및 사진 사용을 위한 매핑 배열]

```

3 <!DOCTYPE html>
4 <html lang="en">
5 <head>
6   <meta charset="UTF-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>CON</title>
9   <link rel="stylesheet" href="../style/mainpage.css">
10 </head>
11 <body>
12   <div class="top-container">
13     <a href=".main.php">
14       
15     </a>
16   <div class="search-bar">
17     <form id="search-form" action="../search_results.php" method="get">
18       <input type="text" name="search" placeholder="음식 이름 검색">
19       <button type="submit">검색</button>
20     </form>
21   </div>
22   <div class="user-info">
23     <?php
24       $username = $_SESSION['username'];
25       $userId = $_SESSION['user_id'];
26       // 세션에 저장되어 있는 사용자 정보를 받아와 오른쪽 상단에 출력.
27       echo "<p>".$username."</p>";
28       echo "<p>".$userId."</p>"?
29     </div>
30   </div>
31 </div>

```

[ 화면의 상단바에 들어가는 정보 ]

```

29 <div class="main-container">
30   <div class="left-container">
31     <div>
32       <h3>금액 설정 </h3>
33       <form id="select-price" action="../search_price.php" method="get">
34         <input type="number" name="min_price" placeholder="최소 금액">
35         <input type="number" name="max_price" placeholder="최대 금액">
36         <button>확인</button>
37       </form>
38     </div>
39     <div class="category-section">
40       <h3>카테고리 </h3>
41       <ul>
42         <?php
43           // 각 카테고리 별로 블럭 가능한 태그 만들고, 각 카테고리 별 조회 정보를 제공하기 위한 함수 호출
44           if ($result_categories->num_rows > 0) {
45             while ($category = $result_categories->fetch_assoc()) {
46               echo '<li><a href="#" onClick="navigateToCategory(\'' . $category['categoryName'] . '\')">' . $category['categoryName'] . '</a></li>';
47             }
48           } else {
49             echo "<li>카테고리가 없습니다.</li>";
50           }
51         ?>
52       </ul>
53     </div>
54     <div>
55       <button onClick="location.href='./cart.php'">장바구니 조회</button>
56       <button onClick="location.href='./order_history.php'">주문내역 조회</button>
57       <?php
58         // 만약 로그인한 사용자가 관리자이면, 판매 통계 조회 버튼 display
59         if($_SESSION['user_id'] == '0') {
60           echo "<button onClick='location.href='../page/admin_stats.php'">판매 통계 조회</button>";
61         }
62       ?>
63     </div>
64   </div>

```

[ 화면의 왼쪽 사이드바에 들어가는 정보 ]

```

13    </div>
12    <div class="main-content">
11        <?php
10            // 카테고리 정보 조회
9                $category_sql = "SELECT * FROM category;";
8                $result_categories = $conn->query($category_sql); // 카테고리 정보 저장
7
6                if ($result_categories->num_rows > 0) {
5                    while ($category = $result_categories->fetch_assoc()) {
4                        $category_name = $category['categoryName'];
3                        echo "<h2>" . $category_name . "</h2>";
2
1                // 음식 정보 조회
12                $food_sql = "SELECT food.foodname, food.pric
1
2                    FROM food JOIN contain ON food.foodname = contain.foodname
3                    WHERE contain.categoryname = '$category_name'";
4
5                $result_foods = $conn->query($food_sql);
6
7                if ($result_foods->num_rows > 0) {
6                    echo "<div class='food-list'>";
7                    while ($food = $result_foods->fetch_assoc()) {
8                        $foodname = $food['foodname'];
9                        $foodprice = $food['price'];
10                       $foodimage = isset($image_map[$foodname]) ? $image_map[$foodname] : 'default.png';
11
12                       echo "<a href="#" class='food-card'>";
13                           echo "<img src='../images/$foodimage' alt='$foodname'>";
14                           echo "<div class='food-details'>";
15                               echo "<div class='food-name'$foodname</div>";
16                               echo "<div class='food-price'$foodprice 원</div>";
17                           echo "</div>";
18                           echo "</a>";
19                   }
20                   echo "</div>";
21
22               } else {
23                   echo "<p>이 카테고리에 속한 음식이 없습니다.</p>";
24               }
25           }
26       } else {
27           echo "<p>설정된 카테고리가 없습니다.</p>";
28       }
29   ?>
30   </div>
31 </div>
32

```

[ 등록되어 있는 전체 음식정보를 조회하는 코드 ]

```

26    <!-- 모달 창 -->
25    <div id="foodModal" class="modal">
24        <div class="modal-content">
23            <span class="close">&times;</span>
22            <h2 id="modalFoodName"></h2>
21            <img id="modalFoodImage" src="" alt="음식 이미지" >
20            <p id="modalFoodPrice"></p>
19            <form id="cartForm" method="post" action="../add_food_to_cart.php">
18                <input type="hidden" id="hiddenFoodName" name="foodname" value="">
17                <input type="hidden" id="hiddenFoodPrice" name="foodprice" value="">
16                <label for="quantity">수량:</label>
15                <input type="number" id="quantity" name="quantity" min="1" value="1">
14                <button type="submit">장바구니에 추가</button>
13            </form>
12        </div>
11    </div>
10

```

[ 음식 선택 시 수량선택하여 장바구니 추가할 수 있는 모달창 ]

```

8 <script>
7   document.addEventListener('DOMContentLoaded', function() {
6     // 모든 food-card 요소에 클릭 이벤트 리스너 추가
5     document.querySelectorAll('.food-card').forEach(function(card) {
4       card.addEventListener('click', function(event) {
3         event.preventDefault();
2
1         // 클릭한 음식 데이터 가져오기
71        const foodName = card.querySelector('.food-name').textContent;
1        const foodPrice = card.querySelector('.food-price').textContent;
2        const foodImageSrc = card.querySelector('img').src;
3
4         // 모달 요소를 가져와서 데이터로 업데이트
5         document.getElementById('modalFoodName').textContent = foodName;
6         document.getElementById('modalFoodPrice').textContent = foodPrice;
7         document.getElementById('modalFoodImage').src = foodImageSrc;
8         document.getElementById('hiddenFoodName').value = foodName;
9         document.getElementById('hiddenFoodPrice').value = foodPrice;
10
11        // 모달 표시
12        const modal = document.getElementById('foodModal');
13        modal.style.display = 'block';
14      });
15    });
16
17    // 모달 닫는 기능
18    document.querySelector('.close').addEventListener('click', function() {
19      document.getElementById('foodModal').style.display = 'none';
20    });
21
22    // 모달 외부 클릭 시 모달 닫기
23    window.addEventListener('click', function(event) {
24      const modal = document.getElementById('foodModal');
25      if (event.target === modal) {
26        modal.style.display = 'none';
27      }
28    });
29  });
30 </script>

```

[ 음식 선택 시 모달창 display에 대한 javascript code ]

```

17 <script>
16   function navigateToCategory(category) {
15     // 선택한 카테고리 정보를 URL 파라미터로 추가하여 페이지 이동
14     window.location.href = '../search_category.php?category=' + encodeURIComponent(category);
13   }
12 </script>

```

[ 왼쪽 사이드바에서 카테고리 선택 시 선택된 카테고리 정보를 url에 붙여 전달 ]

```

4   <script>
5     function navigateToMainpage(usermode) {
6       window.location.href = './main.php';
7     }
8   </script>

```

[ 로고이미지 클릭 시 현재화면(메인화면)으로 돌아가는 코드 ]

### search\_results.php

- 사용자가 음식 이름을 키워드로 검색 시에 사용하는 기능 및 화면
- 검색바에 특정 키워드를 입력 후 검색하면, 입력받은 키워드가 포함된 음식이 display 된다.
- 가격으로 검색하는 기능과 동시에 제공되야하기 때문에 URL에 키워드 및 가격 정보를 함께 넘겨주며, GET method로 정보를 받아와서 조회한다.

```
27 <div class="main-container">
28   <div class="left-container">
29     <div>
30       <h3>금액 설정</h3>
31       <form id="select-price" action="./search_price.php" method="get">
32         <input type="number" name="min_price" placeholder="최소 금액">
33         <input type="number" name="max_price" placeholder="최대 금액">
34       </form>
35       // 현재 URL의 쿼리 파라미터를 숨겨진 입력 필드로 추가하여 키워드와 가격 모두로 검색 가능하도록
36       foreach ($_GET as $key => $value) {
37         if ($key != 'min_price' && $key != 'max_price') { // 이미 있는 min_price와 max_price는 제외
38           echo '<input type="hidden" name="' . htmlspecialchars($key) . '" value="' . htmlspecialchars($value) . '">';
39         }
40       }
41     </div>
42     <button>확인</button>
43   </div>
44 </div>
```

[ 키워드와 가격 설정으로 검색하는 기능이 동시에 제공되야하기 때문에 현재 URL에 가격선택정보를 추가해 넘겨주는 코드 ]

```
6   <div class="main-content">
7     <?php
8       $search_keyword = $_GET['search'];
9       $min_price = isset($_GET['min_price']) ? $_GET['min_price'] : "1";
10      $max_price = isset($_GET['max_price']) ? $_GET['max_price'] : "1000000";
11
12      // 음식 정보 조회
13      $search_sql = "SELECT food.foodname, food.price
14        FROM food
15        WHERE food.foodname LIKE '%$search_keyword%'
16        AND food.price >= '$min_price'
17        AND food.price <= '$max_price'";
18      $result_search = $conn->query($search_sql);
19      echo "<h2>\\"$search_keyword\" 검색 결과</h2>";
20
21      if ($result_search->num_rows > 0) {
22        echo "<div class='food-list'>";
23        while ($food = $result_search->fetch_assoc()) {
24          $foodname = $food['foodname'];
25          $foodprice = $food['price'];
26          $foodimage = isset($image_map[$foodname]) ? $image_map[$foodname] : 'default.png';
27
28          echo "<a href=# class='food-card'>";
29          echo "<img src='./images/$foodimage' alt='$foodname'>";
30          echo "<div class='food-details'>";
31          echo "<div class='food-name'>$foodname</div>";
32          echo "<div class='food-price'>$foodprice 원</div>";
33          echo "</div>";
34          echo "</a>";
35        }
36        echo "</div>";
37      } else {
38        echo "<p>검색 결과가 존재하지 않습니다.</p>";
39      }
40    ?>
41  </div>
```

[ 검색한 키워드를 사용해서 화면에 결과를 띄워주는 코드 ]

### search\_price.php

- 사용자가 가격 범위를 정해서 음식 검색 시에 사용하는 기능 및 화면
- 왼쪽 사이드바에 가격 설정 후 확인을 누르게 되면 설정한 가격 범위 내의 음식이 조회된다.
- 사용자가 최소, 최대금액을 설정할 수 있으며, 최소금액 or 최대금액 중 하나만 설정하여 검색도 가능하다. 이때 설정되지 않은 금액은 기본값으로 설정되어 검색된다. (min\_price : 0원, max\_price : 1,000,000원)
- 설정한 금액은 URL에 넘겨주며, GET method로 가져와 쿼리를 통해 정보를 조회해 전달한다.

```
11 <div class="main-content">
10   <?php
9     $min_price = $_GET['min_price'];
8     $max_price = $_GET['max_price'];
7     // 특정 상황에서만 키워드가 있으니 존재 여부 확인 후 가져오기
6     $search_keyword = isset($_GET['search']) ? $_GET['search'] : null;
5
4     // 사용자가 금액 설정 없이 확인만 누를 경우 min : 0, max : 1,000,000원으로 설정
3     $min_price = empty($min_price) ? "1" : $min_price;
2     $max_price = empty($max_price) ? "1000000" : $max_price;
1
118   // 카테고리 정보 조회
1   $category_sql = "SELECT * FROM category";
2   $result_categories = $conn->query($category_sql); // 카테고리 정보 저장
3
4   if ($result_categories->num_rows > 0) {
5     while ($category = $result_categories->fetch_assoc()) {
6       $category_name = $category['categoryName'];
7       echo "<h2>" . $category_name . "</h2>";
8
9       // 음식 정보 조회
10      $food_sql = "SELECT food.foodname, food.price
11        FROM food JOIN contain ON food.foodname = contain.foodname
12        WHERE contain.categoryname = '$category_name'
13        AND food.price >= $min_price
14        AND food.price <= $max_price
15        AND food.foodname LIKE '%$search_keyword%'";
16      $result_foods = $conn->query($food_sql);
17
18      if ($result_foods->num_rows > 0) {
19        echo "<div class='food-list'>";
20        while ($food = $result_foods->fetch_assoc()) {
21          $foodname = $food['foodname'];
22          $foodprice = $food['price'];
23          $foodimage = isset($image_map[$foodname]) ? $image_map[$foodname] : 'default.png';
24
25          echo "<a href="#" class='food-card'>";
26          echo "<img src='../images/$foodimage' alt='$foodname'>";
27          echo "<div class='food-details'>";
28          echo "<div class='food-name'>$foodname</div>";
29          echo "<div class='food-price'>$foodprice 원</div>";
30          echo "</div>";
31          echo "</a>";
32        }
33        echo "</div>";
34
35      } else {
36        echo "<p>이 카테고리에 속한 음식이 없습니다.</p>";
37      }
38    }
39  } else {
40    echo "<p>설정된 카테고리가 없습니다.</p>";
41  }
42
```

[ 금액설정을 통해 최소, 최대금액 사이의 음식 결과 display 코드 ]

### search\_category.php

- 사용자가 카테고리를 선택하여 음식 검색 시에 사용하는 기능 및 화면
- 사용자가 왼쪽 사이드바에 있는 카테고리를 선택 시 선택된 카테고리에 속해있는 음식이 display된다.
- 카테고리 선택으로 음식을 검색할 경우 최대 한 가지의 카테고리만 선택 가능하다.
- 선택된 카테고리 정보는 URL에 추가되어있고, 그 값을 GET method를 사용하여 가져온다.
- 저장된 카테고리 정보를 사용해서 그 카테고리에 속해있는 음식들을 조회 후 제공한다.

```
7     <div class="main-content">
8         <?php
9             $category_name = $_GET['category'];
10            echo "<h2>" . $category_name . "</h2>";
11
12            // 음식 정보 조회
13            $food_sql = "SELECT food.foodname, food.price
14                FROM food JOIN contain ON food.foodname = contain.foodname
15                WHERE contain.categoryname = '$category_name'";
16            $result_foods = $conn->query($food_sql);
17
18            if ($result_foods->num_rows > 0) {
19                echo "<div class='food-list'>";
20                while ($food = $result_foods->fetch_assoc()) {
21                    $foodname = $food['foodname'];
22                    $foodprice = $food['price'];
23                    $foodimage = isset($image_map[$foodname]) ? $image_map[$foodname] : 'default.png';
24
25                    echo "<a href=# class='food-card'>";
26                    echo "<img src='../images/$foodimage' alt='$foodname'>";
27                    echo "<div class='food-details'>";
28                    echo "<div class='food-name'>$foodname</div>";
29                    echo "<div class='food-price'>$foodprice 원 </div>";
30                    echo "</div>";
31                    echo "</a>";
32                }
33                echo "</div>";
34            } else {
35                echo "<p>이 카테고리에 속한 음식이 없습니다.</p>";
36            }
37        ?>
38    </div>
39  </div>
40
```

[ 선택된 카테고리에 속한 음식 display 코드 ]

## page/cart.php

- 사용자 본인의 장바구니 내역을 조회하는 기능 및 화면
- 설계한 데이터베이스에서 장바구니에 대한 정보 즉, 결제여부에 대한 정보가 없기 때문에 장바구니 번호가 0번인 detailorder 정보를 결제 전 정보라고 가정하고 구현했다.
- 또한 장바구니에 대한 정보와 세부주문내역에 대한 정보가 필요하기 때문에 food와 orderdetail table을 조인해 조회하였다.
- 총 결제금액은 정보를 한줄한줄 띄울 때 totalprice 변수에 각 음식별 금액을 더해 총 totalprice를 출력했다.

```
23 <?php
22 session_start();
21 include '../db_connect.php';
20
19 // 사용자 정보를 세션에서 가져오기
18 $username = $_SESSION['username'];
17 $userid = $_SESSION['user_id'];
16
15 // 장바구니 내역을 조회하는 SQL 쿼리
14 $cart_sql = "SELECT
13     od.foodName AS foodname,
12     od.quantity AS quantity,
11     f.price AS price,
10     od.totalPrice AS totalprice
9     FROM Cart c
8     INNER JOIN OrderDetail od ON c.id = od.id
7     INNER JOIN Food f ON od.foodName = f.foodName
6     WHERE c.cno = '$userid' AND c.id = 0"; // cart id == 0 : 현재 주문중인 장바구니 (아직 결제 X)
5 $result_cart = $conn->query($cart_sql); // 장바구니 정보 저장
4 ?>
3
```

[ 장바구니조회화면에 띄울 정보를 가져오기 위한 db연결 및 조회 코드 ]

```

15      </div>
16      <div class="main-container">
17          <div class="cart-content">
18              <h1>장 바 구 니 조 회 </h1>
19              <table>
20                  <thead>
21                      <tr>
22                          <th>음 식 이 름 </th>
23                          <th>수 량 </th>
24                          <th>1개 당 가 격 </th>
25                          <th>총 가 격 </th>
26                      </tr>
27                  </thead>
28                  <tbody>
29                      <?php
30                          $total_amount = 0;
31                          if ($result_cart->num_rows > 0) {
32                              while ($item = $result_cart->fetch_assoc()) {
33                                  $foodname = $item['foodname'];
34                                  $quantity = $item['quantity'];
35                                  $price = $item['price'];
36                                  $total_price = $item['totalprice'];
37                                  $total_amount += $total_price;
38                                  echo "<tr>
39                                      <td>$foodname</td>
40                                      <td>$quantity</td>
41                                      <td>$price 원 </td>
42                                      <td>$total_price 원 </td>
43                                  </tr>";
44                              }
45                          } else {
46                              echo "<tr><td colspan='4'>장 바 구 니 가 비 어 있 습 니 다 .</td></tr>";
47                          }
48                      ?>
49                  </tbody>
50              </table>
51          </div>
52      </div>
53      <div class="total-section">
54          <p>총 금 액 : <?php echo $total_amount; ?> 원 </p>
55          <button onClick="location.href='../cart_to_history.php'">결 제 하 기 </button>
56      </div>
57 
```

[ 장바구니에 담겨있는 음식들의 정보 출력 코드 ]

### page/order\_history.php

- 사용자 본인의 주문 내역을 조회하는 기능 및 화면
- 주문내역에서 조회가 되는 내용은 장바구니에 없는 정보 즉, cartId != 0 인 정보들만 조회하도록 구현하였다.
- 날짜 설정을 통한 조회가 필요하므로, GET method를 활용해 설정된 날짜 정보를 가져와 조회했으며, 만약 설정된 날짜가 없는 경우 시작날짜 : 1900.01.01., 종료날짜로 설정하여 조회 결과를 보여주었다.
- 설정한 기간을 초기화하는 버튼도 구현하였으며, 기간을 새로고침하기 위한 javascript 코드를 작성해 호출하는 형식으로 구현하였다.

```
1  ?php
1 session_start();
2 include '../db_connect.php';
3
4 // 사용자 정보를 세션에서 가져오기
5 $username = $_SESSION['username'];
6 $userid = $_SESSION['user_id'];
7
8 // 날짜 필터링을 위한 기본값 설정
9 $start_date = isset($_GET['start_date']) ? $_GET['start_date'] : '';
10 $end_date = isset($_GET['end_date']) ? $_GET['end_date'] : '';
11
12 // 주문 내역을 조회하는 SQL 쿼리
13 $order_sql = "SELECT
14     c.id AS ordernum,
15     DATE(c.orderDateTime) AS orderdate,
16     c.orderDateTime AS ordertime,
17     od.foodName AS foodname,
18     od.quantity AS quantity,
19     f.price AS price,
20     od.totalPrice AS totalprice
21     FROM Cart c
22     INNER JOIN OrderDetail od ON c.id = od.id
23     INNER JOIN Food f ON od.foodName = f.foodName
24     WHERE c.cno = '$userid' AND c.id <> 0";
25
26 if (!empty($start_date) && !empty($end_date)) {
27     $order_sql .= " AND DATE(c.orderDateTime) BETWEEN '$start_date' AND '$end_date'";
28 } else {
29     if (empty($start_date)) {
30         $sdate = '1900-01-01';
31     } else {
32         $sdate = $start_date;
33     }
34
35 if(empty($end_date)) {
36     $edate = date('Y-m-d');
37 } else {
38     $edate = $end_date;
39 }
40 $order_sql .= " AND DATE(c.orderDateTime) BETWEEN '$sdate' AND '$edate'";
41 }
42
43 $order_sql .= " ORDER BY c.orderDateTime DESC";
44
45 $result_order = $conn->query($order_sql); // 장바구니 정보 저장
46 ?>
```

[ 주문내역을 조회하기 위한 기본 설정 코드 ]

```
14 <div class="date-filter-container">
15     <form action="" method="get">
16         <label for="start_date">시작 날짜 :</label>
17         <input type="date" id="start_date" name="start_date" value=<?php echo htmlspecialchars($start_date); ?>">
18         <label for="end_date">종료 날짜 :</label>
19         <input type="date" id="end_date" name="end_date" value=<?php echo htmlspecialchars($end_date); ?>">
20         <button type="submit">검색</button>
21         <button type="submit" id="reset_dates">초기화</button>
22     </form>
23 </div>
```

[ 조회 날짜를 선택할 수 있는 container 코드 ]

```

23 <div class="main-container">
24   <div class="order-history-content">
25     <?php
26       $current_date = null;
27       if ($result_order->num_rows > 0) {
28         while ($item = $result_order->fetch_assoc()) {
29           $orderdate = $item['ordertime'];
30           $foodname = $item['foodname'];
31           $quantity = $item['quantity'];
32           $price = $item['price'];
33           $total_price = $item['totalprice'];
34
35           // 날짜가 변경될 때마다 새로운 테이블 시작
36           if ($current_date !== $orderdate) {
37             if ($current_date !== null) {
38               // 이전 날짜의 테이블을 닫음
39               echo "</tbody></table></div>";
40             }
41             $current_date = $orderdate;
42             // 새로운 날짜의 테이블을 시작
43             echo "<div class='order-date'>";
44             echo "<h2>$orderdate</h2>";
45             echo "<table>
46               <thead>
47                 <tr>
48                   <th>음식 이름 </th>
49                   <th>수량 </th>
50                   <th>1개당 가격 </th>
51                   <th>총 가격 </th>
52                 </tr>
53               </thead>
54               <tbody>";
55           }
56
57           // 현재 날짜의 테이블에 행 추가
58           echo "<tr>
59             <td>$foodname</td>
60             <td>$quantity</td>
61             <td>$price 원 </td>
62             <td>$total_price 원 </td>
63           </tr>";
64
65           // 마지막 날짜의 테이블을 닫음
66           echo "</tbody></table></div>";
67         } else {
68           echo "<p>주문 내역이 없습니다.</p>";
69         }
70       ?>
71     </div>
72   </div>

```

[ 주문날짜별로 주문정보 출력 코드 ]

```

8 <script>
9   document.getElementById('reset_dates').addEventListener('click', function() {
10     document.getElementById('start_date').value = '';
11     document.getElementById('end_date').value = '';
12   });
13 </script>

```

[ 설정된 기간을 초기화하기 위한 코드 ]

### page/admin\_stats.php

- 관리자(C0)만이 사용가능한 기능이며, 특정년도의 판매금액과 카테고리 별 인기있는 음식을 검색하는 기능 및 화면
- 연도 별 판매금액 조회 정보를 제공하기 위해 그룹함수 중 ROLLUP 함수를 사용하여 조회하였다.
- 카테고리 별 인기있는 음식 정보를 제공하기 위해 윈도우 함수 중 ROW\_NUMBER 함수를 사용하여 조회하였다.
- 각 기능에 대한 정보를 접근하기 쉽게 하기 위해 배열에 다시 저장해 사용했다.

```
30 <?php
29 session_start();
28 include '../db_connect.php';
27
26 // 사용자 정보를 세션에서 가져오기
25 $username = $_SESSION['username'];
24 $userid = $_SESSION['user_id'];
23
22 // 카테고리 정보 조회
21 $category_sql = "SELECT * FROM category;";
20 $result_categories = $conn->query($category_sql); // 카테고리 정보 저장
19
18
17 // 카테고리 별 인기 메뉴 조회하는 쿼리
16 $populer_menu_sql = "
15 WITH countofsale AS (
14     SELECT
13         ct.categoryName,
12         od.foodName,
11         SUM(od.quantity) AS cnt
10     FROM Category ct
9         INNER JOIN Contain ctn ON ct.categoryName = ctn.categoryName
8         INNER JOIN OrderDetail od ON ctn.foodName = od.foodName
7         GROUP BY ct.categoryName, od.foodName
6 ),
5 ranked_sales AS (
4     SELECT
3         cs.categoryName,
2         cs.foodName,
1         cs.cnt,
31         ROW_NUMBER() OVER (PARTITION BY cs.categoryName ORDER BY cs.cnt DESC) AS rnk
1         FROM countofsale cs
2 )
3 SELECT
4     rs.categoryName AS category,
5     rs.foodName AS popular_menu,
6     rs.cnt AS amount
7     FROM ranked_sales rs
8     WHERE rs.rnk = 1;
9 ";
10
11 $result_popular_menu = $conn->query($populer_menu_sql);
12
13
14 $data_menu = [];
15
16 if ($result_popular_menu->num_rows > 0) {
17     while ($row = $result_popular_menu->fetch_assoc()) {
18         $data_menu[] = $row;
19     }
20 } else {
21     echo "0 결과";
```

[ 통계정보를 제공하기 위한 설정코드 및 카테고리 별 인기메뉴 조회 쿼리 ]

```

1 <!DOCTYPE html>
58 <html lang="ko">
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <title>판매 통계 및 인기 메뉴 조회</title>
5   <link rel="stylesheet" href="../style/mainpage.css">
6   <link rel="stylesheet" href="../style/admin_stats.css">
7 </head>
8 <body>
9   <div class="top-container">
10    <a href=".main.php">
11      
12    </a>
13    <div class="title_box">
14      <h1>판매 통계 및 인기 메뉴 조회 </h1>
15    </div>
16    <div class="user-info">
17      <?php
18        echo "<p>". $username . "</p>";
19        echo "<p>". $userid . "</p>";
20      ?>
21    </div>
22  </div>

```

[ 통계정보 조회 화면 코드 ]

```

43 <div class="main-container">
44   <!-- 판매 통계 영역 -->
45   <div class="sales-stats">
46     <h2>기간 내 판매금액 조회 내용 </h2>
47     <div class="date-filter">
48       <input type="number" id="year" name="year" placeholder="연도 입력">
49       <button id="filter-button" onClick="getYearInfo()">조회 </button>
50     </div>
51   <!-- 판매 통계 데이터를 표시하는 부분 -->
52   <?php
53   // URL에서 연도를 가져옴, 없으면 현재 연도 사용
54   $year = isset($_GET['year']) && !empty($_GET['year']) ? intval($_GET['year']) : date('Y');
55
56   // 판매정보를 조회하는 SQL 쿼리
57   $price_sql = "
58   WITH months AS (
59     SELECT 1 AS month UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION
60     SELECT 5 UNION SELECT 6 UNION SELECT 7 UNION SELECT 8 UNION
61     SELECT 9 UNION SELECT 10 UNION SELECT 11 UNION SELECT 12
62   ),
63   categories AS (
64     SELECT DISTINCT categoryName
65     FROM Category
66   )
67   SELECT
68     ct.categoryName AS category,
69     m.month AS mon,
70     COALESCE(SUM(CASE WHEN od.foodName = ctn.foodName AND ctn.categoryName = ct.categoryName THEN od.totalPrice ELSE 0 END), 0) AS totalprice
71   FROM categories ct
72   CROSS JOIN months m
73   LEFT JOIN Cart c ON m.month = MONTH(c.orderDateTime) AND YEAR(c.orderDateTime) = '$year'
74   LEFT JOIN OrderDetail od ON c.id = od.id
75   LEFT JOIN Contain ctn ON od.foodName = ctn.foodName
76   LEFT JOIN Category ct2 ON ctn.categoryName = ct2.categoryName
77   GROUP BY ct.categoryName, m.month
78   WITH ROLLUP
79   HAVING mon IS NOT NULL AND category IS NOT NULL
80   ORDER BY m.month, ct.categoryName;
81   ";
82
83   $result_price = $conn->query($price_sql); // 판매 정보 저장

```

[ 기간 내 판매금액 조회 쿼리 ]

```

1 // 쿼리로 얻어낸 데이터를 표 형식에 맞게 변환
2 $data = []; // data 저장 배열
3 $totalSales = []; // 총 판매금액 저장 배열
4 while ($row = $result->fetch_assoc()) {
5     $data[$row['mon']][$row['category']] = $row["totalprice"];
6     if(!isset($totalSales[$row['category']])) { // totalSales array에 아직 카테고리의 합산금액이 없는 경우 0으로 설정
7         $totalSales[$row['category']] = 0;
8     }
9     $totalSales[$row['category']] += $row["totalprice"];
10 }
11 <?php
12 // table 정보 위에 조회된 연도 출력
13 echo "<h3>".$year."년 조회 결과 </h3>";
14 ?>
15 <table>
16     <thead>
17         <tr>
18             <th>월 </th>
19             <?php
20                 // 카테고리 헤더 출력
21                 foreach ($data[key($data)] as $category => $value) {
22                     echo "<th>$category</th>";
23                 }
24             ?>
25         </tr>
26     </thead>
27     <tbody>
28         <?php
29             // 데잍 출력
30             foreach ($data as $month => $categories) { // key => value
31                 echo "<tr>";
32                 echo "<td>".$month."</td>";
33                 foreach ($categories as $value) {
34                     echo "<td>".$value."원 </td>";
35                 }
36                 echo "</tr>";
37             }
38         ?>
39     </tbody>
40     <tfoot>
41         <tr>
42             <td>총 판매금액 </td>
43             <?php
44                 // 총 판매금액 출력
45                 foreach ($totalSales as $total) {
46                     echo "<td>".$total."원 </td>";
47                 }

```

[ 기간 내 판매금액 조회 정보 출력 코드 ]

```

16 <!-- 인기 메뉴 조회 영역 -->
15 <div class="popular-menus">
14     <h2>카테고리 별 인기메뉴 조회 </h2>
13     <h3>이 정보는 판매 시작시점부터 조회시점까지의 총 판매 개수가 많은 음식을 보여줍니다.</h3>
12     <!-- 인기 메뉴 조회 기능을 구현하는 부분 -->
11     <table>
10         <thead>
9             <tr>
8                 <?php foreach ($data_menu as $row): ?>
9                     <th><?php echo $row['category']; ?></th>
10                <?php endforeach; ?>
11         </thead>
12         <tbody>
13             <tr>
14                 <?php foreach ($data_menu as $row): ?>
15                     <td><?php echo $row['popular_menu']; ?></td>
16                 <?php endforeach; ?>
17             </tr>
18             <tr>
19                 <?php foreach ($data_menu as $row): ?>
20                     <td><?php echo $row['amount']; ?>개 </td>
21                 <?php endforeach; ?>
22             </tr>
23         </tbody>
24     </table>
25     </div>
26 </div>

```

[ 카테고리 별 인기메뉴 정보 출력 ]

```

2     <script>
3         function getYearInfo() {
4             var year = document.getElementById('year').value;
5             if (year) {
6                 window.location.href = window.location.pathname + "?year=" + year;
7             } else {
8                 alert("연도를 입력하세요.");
9             }
10        }
11    </script>
12

```

[ 연도 별 판매금액 조회 시 입력받은 연도정보 처리 코드 ]

### login\_process.php

- login시 비밀번호 일치여부, 등록된 회원인지 확인과 같은 로그인과 관련된 처리
- 사용자 정보를 session에서 가져와서 쿼리로 데이터베이스에 존재하는지 조회한다.
- 로그인이 성공적으로 된다면, 세션에 현재 로그인한 유저정보를 넘겨준다.
- 등록된 유저정보가 없거나 비밀번호가 틀려 로그인이 실패한 경우 alert창으로 오류를 알려준 후 로그인 페이지로 돌아간다.

```
1  <?ph
1  // session 정보 가져오기
2  session_start();
3
4  // db 연결파일 가져오기
5  include 'db_connect.php';
6
7  // login 처리 시 메세지
8  $error_msg = '';
9
10 // 사용자 입력 값 받기
11 $username = $_POST['username'];
12 $password = $_POST['password'];
13
14 // SQL 쿼리문 작성 : 사용자 정보 조회
15 $sql = "SELECT * FROM Customer WHERE name = ?";
16 $stmt = $conn->prepare($sql);
17 $stmt->bind_param("s", $username); // 첫 번째 인자는 바인딩 할 값의 데이터 타입
18 $stmt->execute();
19 $result = $stmt->get_result();
20
21 // 사용자가 한명이라도 있는지 확인 후 로그인 진행
22 if ($result->num_rows > 0) {
23     $userinfo = $result->fetch_assoc(); // col name 으로 접근
24
25     // 비밀번호가 일치하는 경우 세션에 정보저장
26     if($password === $userinfo['password']) {
27         $_SESSION['user_id'] = $userinfo['cno'];
28         $_SESSION['username'] = $userinfo['name'];
29         header("Location: ./page/main.php"); // login 성공 시 메인화면으로 이동
30         exit();
31     }
32
33     // 비밀번호가 일치하지 않는 경우
34     else {
35         // 비밀번호가 일치하지 않는 경우
36         echo "<script>
37             alert('비밀번호가 잘못되었습니다.');
38             window.location.href = './page/login.php';
39         </script>";
40     }
41     // 사용자가 존재하지 않는 경우
42 } else {
43     // 사용자가 존재하지 않는 경우
44     echo "<script>
45         alert('존재하지 않는 사용자입니다.');
46         window.location.href = './page/login.php';
47     </script>";
48 }
49 ?>
50
51 </body>
```

[ 로그인 처리(사용자 정보 대조) 코드 ]

### add\_food\_to\_cart.php

- 사용자가 음식을 선택하여 장바구니에 담는 기능
- orderDetail에 주문 음식 정보를 저장할 때 itemno를 중복되지 않는 번호를 사용하기 위해 현재 저장된 item중 가장 큰 번호를 찾아 그 번호 + 1의 값을 사용했다.
- 사용자가 추가한 음식정보를 orderDetail table에 저장할 때 이미 추가가 되어있는 음식이라면 수량만 변경해주고, 처음 추가가 된 음식이라면 삽입으로 구현했다.
- 상세 음식정보를 전부 저장했다면 이 정보들을 현재 사용자의 장바구니에 추가해주도록 구현했다.
- 장바구니id는 언제나 0번이며, 장바구니에 이미 음식이 있다면 orderdetail만 추가하며, 장바구니가 비어있다면 장바구니 생성 후 추가해주었다.

```

39 <?php
40 session_start();
41 include './db_connect.php';
42
43 if ($_SERVER["REQUEST_METHOD"] === "POST") {
44     $foodname = $_POST['Foodname'];
45     $quantity = $_POST['Quantity'];
46     $userid = $_SESSION['User_id'];
47
48     // 현재 가장 큰 itemno를 찾기
49     $sql_max_itemno = "SELECT MAX(itemno) AS max_itemno FROM OrderDetail";
50     $result = $conn->query($sql_max_itemno);
51     if ($result->num_rows > 0) {
52         $row = $result->fetch_assoc();
53         $new_itemno = $row['max_itemno'] + 1;
54     } else {
55         $new_itemno = 1; // 테이블이 비어 있는 경우 itemno를 1로 설정
56     }
57
58     // 장바구니에 추가하는 SQL 쿼리 (주문 한 번에 대한 정보 저정 -> 결제 버튼 클릭 시에는 Cart table에 저장)
59     $create_cart_sql = "INSERT INTO CART (cno, id, orderDateTime) VALUES ('$userid', 0, NOW())";
60
61     // 장바구니에 추가하고자 하는 음식이 이미 존재한다면, 세로 정보를 추가하는 것이 아닌 그 음식의 수량을 update
62     $isExistFood_sql = "SELECT COUNT(*) AS cnt
63                         FROM OrderDetail od
64                         WHERE od.foodname = '$foodname' AND od.id = 0";
65
66     $res_q = $conn->query($isExistFood_sql);
67     $isExistFood = ($res_q && $res_q->num_rows > 0 && $res_q->fetch_assoc()['cnt'] > 0) ? 1 : 0; // cnt로 접근 시의 값이 0이면 존재 x, 0이 아니면 존재 o
68     if($isExistFood === 0) {
69         // 장바구니에 존재하지 않는 음식이므로 insert
70         $create_orderDetail_sql = "INSERT INTO OrderDetail (itemno, id, quantity, totalPrice, foodName) VALUES
71             ('$new_itemno',
72              '$quantity',
73              (SELECT price FROM Food WHERE foodName = '$foodname') * '$quantity',
74              '$foodname')";
75     } else {
76         // 장바구니에 존재하는 음식이므로 quantity만 update
77         $create_orderDetail_sql = "UPDATE OrderDetail
78             SET quantity = quantity + '$quantity',
79                 totalPrice = totalPrice + ((SELECT price FROM Food WHERE foodName = '$foodname') * '$quantity')
80             WHERE foodname = '$foodname' AND id = 0";
81     }
82 }

```

[ orderDetail table에 정보 추가 코드 ]

```

2
3     // 장바구니에 음식이 있어 0번 cart(결제 전 장바구니)가 활성되어 있는지 확인
4     $check_sql = "SELECT COUNT(*) AS cnt FROM CART c WHERE c.id = 0";
5     $res = $conn->query($check_sql);
6     $isActivate = ($res && $res->num_rows > 0 && $res->fetch_assoc()['cnt'] > 0) ? 1 : 0;
7
8     if($isActivate === 1) {
9         // 활성화되어 있으면 orderdetail만 추가
10        if($conn->query($create_orderDetail_sql) === TRUE) {
11            echo "<script>alert('장바구니에 추가되었습니다 !');</script>";
12            echo "<script>window.location.href = './page/main.php';</script>";
13        }
14        else {
15            echo "Error: " . $create_orderDetail_sql . "<br>" . $conn->error;
16        }
17    } else {
18        // 활성화되어 있지 않기 때문에 Cart도 만들어 줌.
19        if ($conn->query($create_cart_sql) === TRUE) {
20            if($conn->query($create_orderDetail_sql) === TRUE) {
21                echo "<script>alert('장바구니에 추가되었습니다 !');</script>";
22                echo "<script>window.location.href = './page/main.php';</script>";
23            }
24            else {
25                echo "Error: " . $create_orderDetail_sql . "<br>" . $conn->error;
26            }
27        }
28        else {
29            echo "Error: " . $create_cart_sql . "<br>" . $conn->error;
30        }
31    }
32
33    $conn->close();

```

[ 장바구니에 주문내역 추가 코드 ]

### cart\_to\_history.php

- 사용자가 장바구니 조회 화면에서 결제하기를 눌렀을 때 결제가 이루어지며, 장바구니에서 정보가 사라지고 주문내역에 표시되는 기능
- 장바구니 번호는 겹치면 안되므로 현재 존재하는 장바구니 번호 중 가장 큰 번호를 찾아 1추가한 값을 cartId로 사용하였다.
- 장바구니 내역을 주문내역으로 옮긴다는 것은 장바구니 정보였던 0번 장바구니를 새롭게 정의한 장바구니번호(new\_cartId)로 변경해주면 된다.

```
1  ?php
2      // session 정보 가져오기
3      session_start();
4
5      // db 연결 파일 가져오기
6      include 'db_connect.php';
7
8      // 현재 가장 큰 cartId를 찾기
9      $sql_max_cartid = "SELECT MAX(id) AS max_cartid FROM Cart";
10     $result = $conn->query($sql_max_cartid);
11     if ($result->num_rows > 0) {
12         $row = $result->fetch_assoc();
13         $new_cartid = $row['max_cartid'] + 1;
14     } else {
15         $new_cartid = 1; // 테이블이 비어 있는 경우 cartId를 1로 설정
16     }
17
18     // 장바구니 정보를 order_history로 옮기기 == cartid를 0이 아닌 값으로 변경
19     $sql_update_cartid = "UPDATE Cart SET id='$new_cartid' WHERE id = 0";
20
21     if ($conn->query($sql_update_cartid) === TRUE) {
22         echo "<script>alert('결제가 완료되었습니다 !');</script>";
23         echo "<script>window.location.href = './page/main.php';</script>";
24     } else {
25         echo "Error: " . $sql_update_cartid . "<br>" . $conn->error;
26     }
27 ?>
28
29 </body>
30 </html>
31
```

[ 결제된 정보를 장바구니에서 주문내역으로 옮기는 코드 ]