

배스킨 라빈스 클론코딩



팀명 : 배스킨 미누쓰

팀장 : 이민우

팀원 : 김채연 / 노현지 / 오지환 / 이윤기

CONTENTS

- 1 개요 및 배경
- 2 베스킨 미누쓰만의 차별
- 3 ^점사용된 TOOL
- 4 주요 페이지 및 코드 설명
- 5 사이트 시연
- 6 System Architecture Overview
- 7 트러블 슈팅 및 향후 개선점
- 8 Q&A

01 개요 및 배경

01 왜 베스킨 라빈스인가?

- 친숙한 브랜드
- 사용자 친화적 디자인->시각적 비교 가능
- 실용성 높은 레이아웃

02 베스킨 미누쓰만의 차별점

02 베스킨 미누쓰만의 차별점

01

다크모드 스타일 추가

기존 홈페이지는
라이트모드만 지원

02

회원가입 페이지 제작

해피포인트 가입이 아닌
메인 페이지에서 coolsms를
통해 본인 인증 후
회원가입

03

아이스크림 카드 게임

아이스크림 짝 맞추기 카드
게임 보상으로 구매자에게
가상 포인트 지급

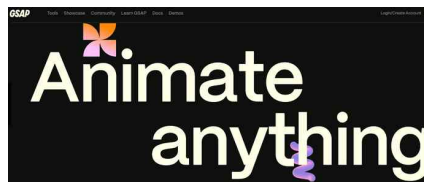
04

장바구니/
가상 구매 기능

직접 원하는 아이스크림을
담은 후 가상 포인트로 구매

03 사용한 TOOL

03 Tools



04 주요 페이지 및 코드 설명

00 기본 템플릿 레이아웃 & 컨벤션 가이드 문

서

```
index.html > html > body > div.page > div.container > header.header
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4 <meta charset="UTF-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6 <title>회원가입</title>
7 <link rel="stylesheet" href="/assets/css/common-ui.css" />
8 </head>
9 <body>
10 <div class="page">
11 <div class="container">
12
13 <header class="header">
14 <!-- header HTML 추가 -->
15 </header>
16
17 <main class="contents">
18 <div class="content">
19 <!-- 컨텐츠 페이지 -->
20 </div>
21 </main>
22
23 <footer class="footer">
24 <!-- footer HTML 추가 -->
25 </footer>
26
27 </div>
28 </div>
29 <script src="/assets/libs/jquery/jquery-3.6.0.min.js"></script>
30 <script src="/assets/js/common-ui.js"></script>
31 <script>
32 /* 스크립트 추가 */
33 </script>
34 </body>
35 </html>
```

규칙



HTML

CSS

JS

네이밍



파일/폴더 네이밍

이미지 네이밍

클래스 작명

예약어

00 BEM 방식 클래스 선언

BEM — is a methodology that helps you to create reusable components and code sharing in front-end development

Learn why it's better

```
1 #opinions_box h1 {
2     margin: 0 0 8px 0;
3     text-align: center;
4 }
5
6 #opinions_box {
7     p.more_pp {
8         a {
9             text-decoration: underline;
10         }
11     }
12
13     input[type="text"] {
14         border: 1px solid #ccc!important;
15     }
16 }
```

Classic CSS

```
1 .opinions-box {
2     margin: 0 0 8px 0;
3     text-align: center;
4
5     &__view-more {
6         text-decoration: underline;
7     }
8
9     &__text-input {
10         border: 1px solid #ccc;
11     }
12
13     &--is-inactive {
14         color: gray;
15     }
16 }
```

BEM

00 기본 템플릿 레이아웃 구조

```
// 로드 페이지 관리
let basePath = location.port === "8080" ? "" : `http://127.0.0.1:${location.port}`;
// Header/Footer 로드
$(".page .header").load(`${basePath}/html/common/header.html?v=${Date.now()}`);
$(".page .footer").load(`${basePath}/html/common/footer.html?v=${Date.now()}`);
```

```
5 <ul class="sub-menu-list sub-list-2">
6   <!-- 이벤트 -->
7   <li class="sub-menu-item sub-item-1">
8     <a href="#" class="sub-item-title">이벤트</a>
9     <ul class="sub-list2-a sub-list2-a-1">
10      <li><a href="/html/main/game.html" class="sub-list2-a-con-1">게임</a></li>
11    </ul>
12  </li>
13  <!-- BR Play -->
14  <li class="sub-menu-item sub-item-2">
15    <a href="#" class="sub-item-title">BR Play</a>
16    <ul class="sub-list2-a sub-list2-a-2">
17      <li><a href="/html/common/post.html" class="sub-list2-a-con-3">배라광장</a></li>
18    </ul>
19  </li>
20 </ul>
21 <ul class="sub-menu-list sub-list-3">
22 </ul>
```

```
6 <div class="footer-top">
7
8   <!-- 왼쪽 정보-->
9   <div class="SiteFooterLeftbox">
10     
11     <p class="SiteFooterInfo_text">사업자 등록번호 : 303-81-09535</p>
12     <p class="SiteFooterInfo_text">비알코리아(주) 대표이사 도세호</p>
13     <p>서울특별시 서초구 남부순환로 2620(양재동 11-149번지)</p>
14   </div>
15   <p class="SiteFooterInfo_text">TEL : 080-555-3131</p>
16   <p class="SiteFooterInfo_text">개인정보관리책임자 : 김민숙/상무</p>
17   <p class="copyright">
18     Copyright © 2023 BRKOREA Company. All Rights Reserved.
19   </p>
20 </div>
21
22 <!-- 메뉴 라인-->
23 <div class="footer-menu-row">
```

01 회원가입 페이지 제작

(with coolsms)

“다크 모드 / 라이트 모드 지원”



BM Story BM Store

회원가입

지금 가입하고 특별한 서비스를 이용해보세요

아이디

비밀번호

이메일 주소

이름

생년월일

휴대폰 번호 선택필수

제출하기

BR

baskin minoos

주최 | 운영 | 후원
주최 | 운영 | 후원
주최 | 운영 | 후원

주최 | 운영 | 후원
주최 | 운영 | 후원
주최 | 운영 | 후원

주최 | 운영 | 후원
주최 | 운영 | 후원
주최 | 운영 | 후원

Menu

BR Play

Delivery/Store

Information Center

BR baskin minoos Menu BR Play BR Story BR Store

회원가입

지금 가입하고 특별한 서비스를 이용해보세요

아이디

비밀번호

이메일 주소

이름

생년월일

휴대폰 번호 선택필수

인증요청 필수

제출하기

BR

baskin minoos

주최 | 운영 | 후원
주최 | 운영 | 후원
주최 | 운영 | 후원

주최 | 운영 | 후원
주최 | 운영 | 후원
주최 | 운영 | 후원

주최 | 운영 | 후원
주최 | 운영 | 후원
주최 | 운영 | 후원

Menu

BR Play

Delivery/Store

Information Center

01 회원가입 페이지 제작 핵심 코드

```
<div class="signup_inner">
  <form action="" class="signup_form" method="post">
    <fieldset class="login_block">
      <input type="hidden" name="redirectUrl" value="/" />

      <div class="signup_field">
        <input type="text" id="userid" name="userid" class="signup_input" required/>
        <label for="userid" class="signup_label">아이디</label>
      </div>

      <div class="signup_field">
        <input type="password" id="password" name="password" class="signup_input" required/>
        <label for="password" class="signup_label">비밀번호</label>
      </div>

      <div class="signup_field">
        <input type="text" id="email" name="email" class="signup_input" required/>
        <label for="email" class="signup_label">이메일 주소</label>
      </div>

      <div class="signup_field">
        <input type="text" id="name" name="name" class="signup_input" required/>
        <label for="name" class="signup_label">이름</label>
      </div>

      <div class="signup_field">
        <input type="text" id="birth" name="birth" class="signup_input" required/>
        <label for="birth" class="signup_label">생년월일</label>
      </div>
    </fieldset>
  </form>
</div>
```

01 회원가입 페이지 제작 핵심 코드

```
<div class="signup__field signup__phone-group">
  <input type="text" id="hp" name="hp" class="signup__input" required/>
  <label for="hp" class="signup__label">휴대폰 번호</label>
  <button type="button" id="send-certify" class="signup__button--certify">인증요청</button>
</div>

<div id="certification-field" class="signup__field signup__phone-group" style="display: none;">
  <input type="text" id="certification" name="certification" class="signup__input" required maxlength="6"/>
  <label for="certification" class="signup__label">인증번호</label>
  <button type="button" id="verify-certify" class="signup__button--certify">확인</button>
</div>

  <button type="submit" class="signup__button--submit" disabled>제출하기</button>
</fieldset>
</form>
</div>
```


01 회원가입 페이지 제작 핵심 코드 (with ray.so)

1. CoolSMS 클라이언트 추가화

```
import coolsms from 'coolsms-node-sdk';

const messageService = new coolsms.default(
  config.coolsms.apiKey,
  config.coolsms.apiSecret
);
```

2. 인증번호 저장 및 유효시간 관리

```
// Map: {code, expires, attempts} 형태로 저장
verificationCodes.set(phone, {
  code: code,
  expires: Date.now() + 3 * 60 * 1000,
  // 3분 유효
  attempts: 0
});
```

3. SMS 발송 호출

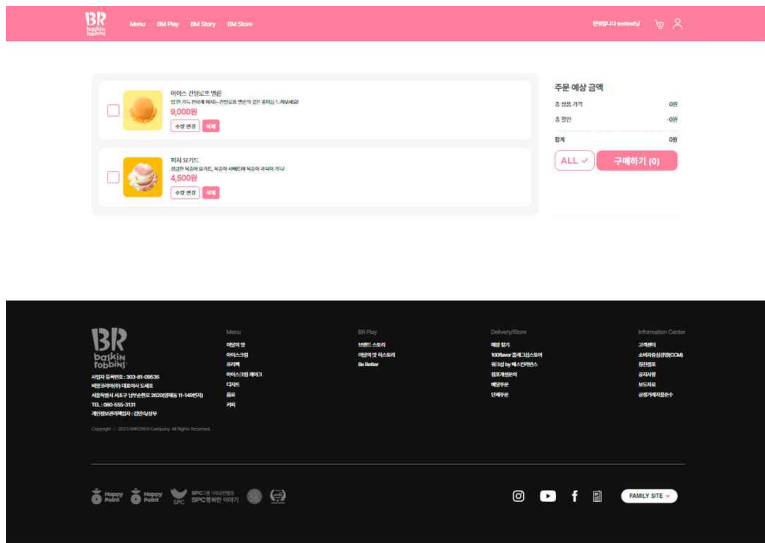
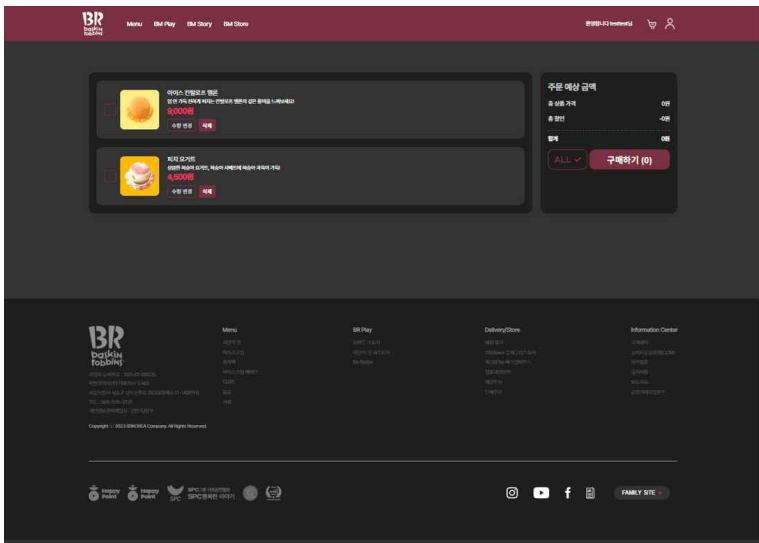
```
const result = await messageService.sendOne({
  to: phone,
  from: config.coolsms.senderNumber,
  text: `[Baskin Minoo's] 인증번호는 ${code}입니다.`
});
```

4. 개발 환경 로깅 (선택 사항)

```
if (process.env.NODE_ENV === 'development') {
  console.log('개발 환경 - 인증번호:', code);
}
```


02 장바구니 페이지 제작

“다크 모드 / 라이트 모드 지원”



02 장바구니 페이지 제작 핵심 코드

```
<section class="cart-container">
  <div class="cart-main">
    <div id="cartItemsContainer" class="cart-items">
      <!-- 다른 상품 반복 -->
    </div>

    <!-- 주문 요약 영역 -->
    <aside class="cart-summary">
      <h3>주문 예상 금액</h3>
      <dl>
        <div class="summary-item">
          <dt>총 상품 가격</dt>
          <dd>0원</dd>
        </div>
        <div class="summary-item">
          <dt>총 할인</dt>
          <dd class="discount">-0원</dd>
        </div>
        <div class="summary-total">
          <dt>합계</dt>
          <dd>0원</dd>
        </div>
      </dl>
      <div class="summary-buttons">
        <button id="allcheck" class="summary-button" onclick="toggleCheckAll(this)">ALL</button>
        <button id="buyBtn" class="summary-button">구매하기 (0)</button>
      </div>
    </aside>
  </div>
</section>
```

02 장바구니 페이지 제작 코드 (with ray.so)

1. 총 금액 계산

```
// purchaseList UI*  
let total = 0;  
for (const item of items) {  
  const cartItem = await cartRepository.findOne(userId, item.iceidx);  
  const unitPrice = cartItem.totalPrice / cartItem.quantity;  
  total += unitPrice * item.quantity;  
}  
  
// cartItem.totalPrice, cartItem.quantity로 단위(unitPrice)를 구할 수  
// 있으며, 이를 곱해서 total에 더할 수 있음  
  
// totalPrice UI*  
const qty = parseInt(quantity);  
const pricePerItem = 4500;  
const total = pricePerItem * qty;
```

02 장바구니 페이지 제작 핵심 코드 (with ray.so)

📁 📁 📁

2. 포인트 유효성 검사

```
if (user.point < total) {  
  return res.status(400).json({ message: "포인트가 부족합니다." });  
}
```

📁 📁 📁

3. 포인트 차감

```
await userRepository.updatePoint(userId, user.point - total);
```

📁 📁 📁

4. 개발 환경 로깅 (선택 사항)

```
if (process.env.NODE_ENV === 'development') {  
  console.log('개발 환경 - 인증번호:', code);  
}
```

02 구매완료 페이지

(with GASP & Anime)

```
<svg xmlns="http://www.w3.org/2000/svg" width="100%" height="100%" viewBox="0 0 1000 750" preserveAspectRatio="xMidYMid" >
  <defs>
    <filter id="goo">
      <feGaussianBlur in="SourceGraphic" stdDeviation="10" result="blur" />
      <feColorMatrix in="blur" mode="matrix" values="1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 19 -1" />
      <feComposite in="SourceGraphic" in2="goo" operator="atop"/>
    </filter>
  </defs>
  <rect class="bg1" width="100%" height="100%" fill="#7df0b9" stroke="#7df0b9" />
  <g class="bg2" stroke="#fff" stroke-width="30" filter="url(#goo)">
    <path class="bg2_wave" fill="none" stroke="#fff" d="M1458-489c-65.5,44.5-61.5,85.5-71.5,192.5s-19.72,144.5-19.72,144.5" />
    <line class="bg2_drip" x1="0" y1="0" x2="0" y2="150" stroke-width="30" stroke-linecap="round" />
  </g>
  <g class="bg3" fill="#7268e5" stroke="#7268e5" filter="url(#goo)">
    <path class="bg3_wave" d="M1032.39-602c-20.68,25.45-52.12,40.71-92.42,49.25c-101.5,21.5-250.5,21.5-250.5,21.5" />
    <line class="bg3_drip" x1="0" y1="0" x2="0" y2="150" stroke-width="30" stroke-linecap="round" />
  </g>
  <g class="cone">
    <path class="coneBase" fill="#FFDE7D" d="M213.5,309.5c4,6,39,55,42,60s6,6,9,5,0s37.5-50,40,40" />
    <g fill="#FF3680" stroke="#FF3680" stroke-linecap="round">
      <line class="qDrip" x1="312" y1="400" x2="312" y2="414" stroke-width="18" />
      <line class="qDrip" x1="320" y1="400" x2="320" y2="407" stroke-width="15" />
    </g>
    <path d="M351.5,357.5c-1.98-12.85-12.23-14-24c-4,2,0-9-7-15c-3.22-2.76-7-6-9-4s-54,67-54,67" />
  </g>
  <text class="txt" fill="#ff3880" x="360" y="390">BaskinMinoos</text>
  <text class="txt" fill="#000" x="360" y="430" font-size="28" letter-spacing="3.3">Inspired by</text>
  <text class="txt" fill="#000" x="360" y="470" font-size="28" letter-spacing="3.3">Built by Us</text>
  <rect class="titleHit" x="210" y="300" width="550" height="140" fill="rgba(0,0,0,0.5)" />
</svg>
<div class="complete-section_menu">
```

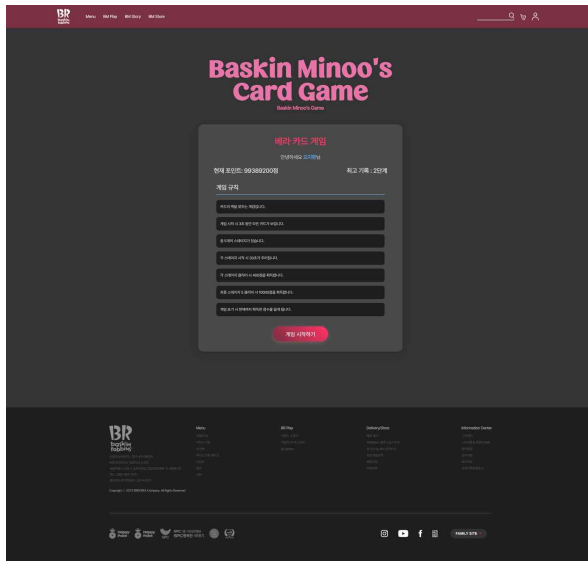
```
let nextBg = '.bg3',
mainTL = gsap.timeline()

.fromTo('.cone', {scale:0, transformOrigin:'75% 65%'},{duration:1, scale:1, ease:'elastic.out(2)'}, 0)
.to('.cone', {duration:0.3, rotation:20, yoyo:true, repeat:1, ease:'power3.in'}, 0.5)
.from('.cone', {duration:0.7, x:200, ease:'back.inOut(2)'}, 0.5)
.from('.txt', {duration:0.5, opacity:0, stagger:0.25, ease:'power1.inOut'}, 0.8)
.from('.txt', {duration:0.5, y:'+=50', stagger:0.2, ease:'expo'}, 0.8)
.from('.m', {duration:0.8, attr:{y1:70}, stagger:-0.12, ease:'back.out(3)'}, 1.8)
.from('.bg2', {duration:0.8, scale:1.2, ease:'back.out(2)'}, 0.4)
.from('.bg3', {duration:0.6, scale:1.2, transformOrigin:'110% 110%', ease:'back.out(1.2)'}, 0.5)
.from('.bg3_wave', {duration:8, x:-585, y:600, ease:'none', repeat:-1}, 0.3)
.to('.bg2_wave', {duration:6, x:-447, y:704, ease:'none', repeat:-1}, 0.3)
.call(startDrip, ['.bg3_drip', 250, -160, 110], 1.5)
.call(startDrip, ['.bg2_drip', 830, 300, 50], 1.9)
```

```
function startDrip( drip, xBase, yStart, hStart ){
  gsap.fromTo(drip, {
    x:()=>gsap.utils.random(xBase,xBase+80),
    y:yStart,
    attr:{
      y1:0,
      y2:hStart,
      'stroke-width':()=>gsap.utils.random(35,70)
    }
  }, {
    duration:1.1,
```

03 아이스크림 카드 게임

“다크 모드 / 라이트 모드 지원”



게임 규칙

카드의 짝을 맞추는 게임입니다.

게임 시작 시 3초 동안 모든 카드가 보입니다.

총 5개의 스테이지가 있습니다.

각 스테이지 시작 시 30초가 주어집니다.

각 스테이지 클리어 시 400점을 획득합니다.

최종 스테이지 5 클리어 시 10000점을 획득합니다.

게임 포기 시 현재까지 획득한 점수를 잃게 됩니다.

게임 시작하기

03 아이스크림 카드 게임 핵심코드

```
<div id="game-screen" class="game-screen" style="display: none;">
  <h1>베라 카드 게임</h1>
  <div class="user-greeting">안녕하세요 <span id="user-name"></span>님</div>
  <div class="game-info">
    <div id="timer">남은 시간: 30초</div>
    <div id="score">점수: 0</div>
    <button id="give-up-button" class="give-up-button">게임 포기</button>
  </div>
  <div id="game"></div>
  <div id="message"></div>
</div>
```

03 아이스크림 카드 게임 핵심코드

```
<div id="start-screen" class="start-screen">
  <h1>베라 카드 게임</h1>
  <div class="user-greeting">안녕하세요 <span id="user-name-start"></span>님</div>
  <div class="score-container">
    <div class="high-score">현재 포인트: <span id="current-point">0</span>점</div>
    <div class="high-score">최고 기록 : <span id="high-score">0</span>단계</div>
  </div>
  <div class="game-rules">
    <h2>게임 규칙</h2>
    <ul>
      <li>카드의 짝을 맞추는 게임입니다.</li>
      <li>게임 시작 시 3초 동안 모든 카드가 보입니다.</li>
      <li>총 5개의 스테이지가 있습니다.</li>
      <li>각 스테이지 시작 시 30초가 주어집니다.</li>
      <li>각 스테이지 클리어 시 400점을 획득합니다.</li>
      <li>최종 스테이지 5 클리어 시 10000점을 획득합니다.</li>
      <li>게임 포기 시 현재까지 획득한 점수를 잃게 됩니다.</li>
    </ul>
  </div>
  <button id="start-button" class="start-button">게임 시작하기</button>
</div>
```


03 아이스크림 카드 게임 핵심코드 (with ray.so)

1. 스테이지 클리어 시 포인트 지급

```
// req.user.id 와 body.stage 를 넘겨서 클리어 된만큼 보상 지급  
const reward = await gameRepository.giveGamePoint(req.user.id, stage);
```

2. 유저 정보(포인트, 최대 클리어 스테이지) 조회

```
// req.user.id 를 넘겨서 userId, point, maxlearedStage 를 가져옴  
const userInfo = await gameRepository.getPoint(userId);
```

05 시연

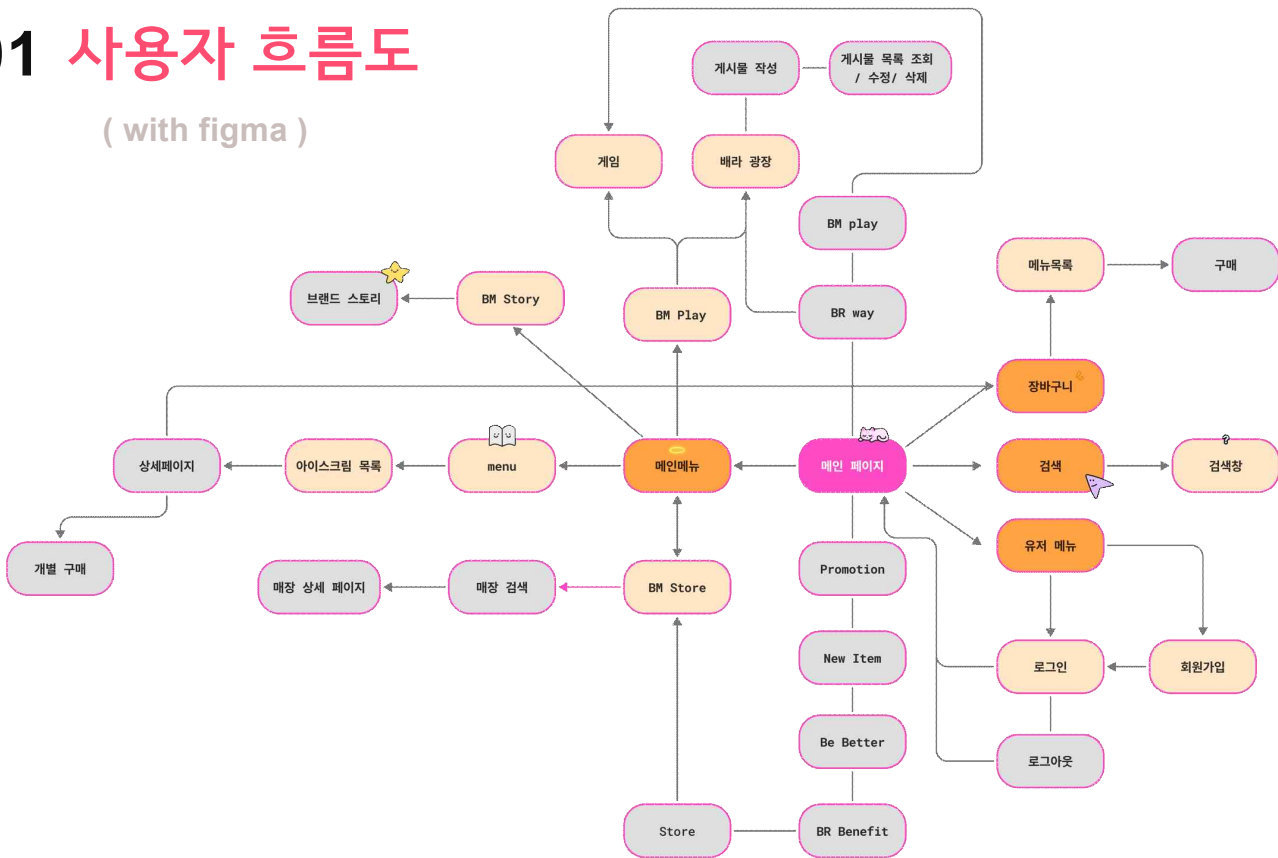


06.

SYSTEM ARCHITECTURE DIAGRAM

01 사용자 흐름도

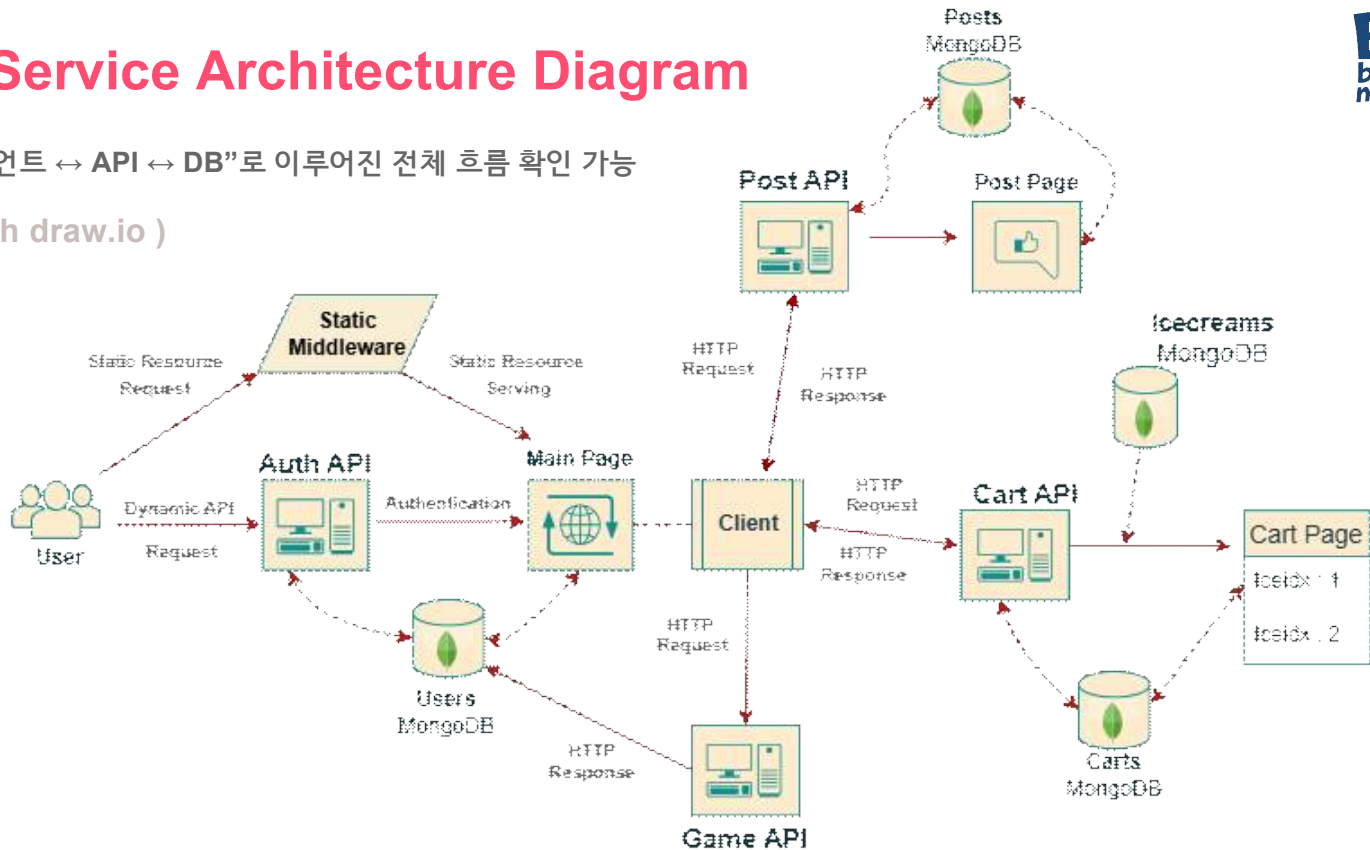
(with figma)



02 Service Architecture Diagram

“클라이언트 ↔ API ↔ DB”로 이루어진 전체 흐름 확인 가능

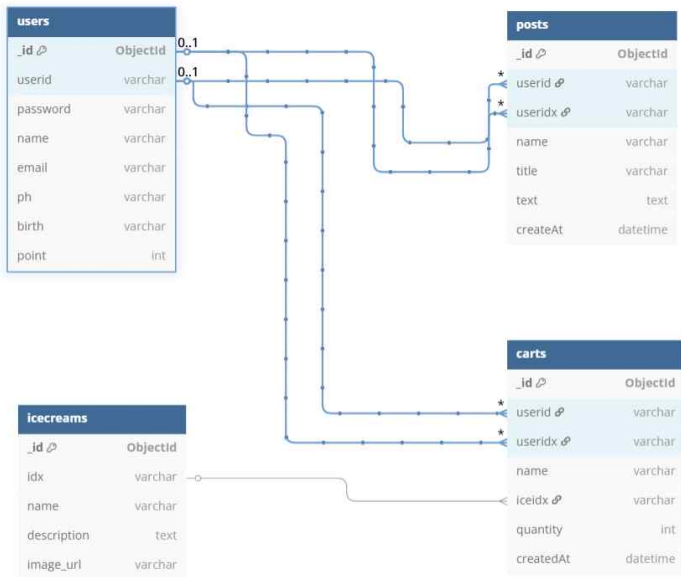
(with draw.io)



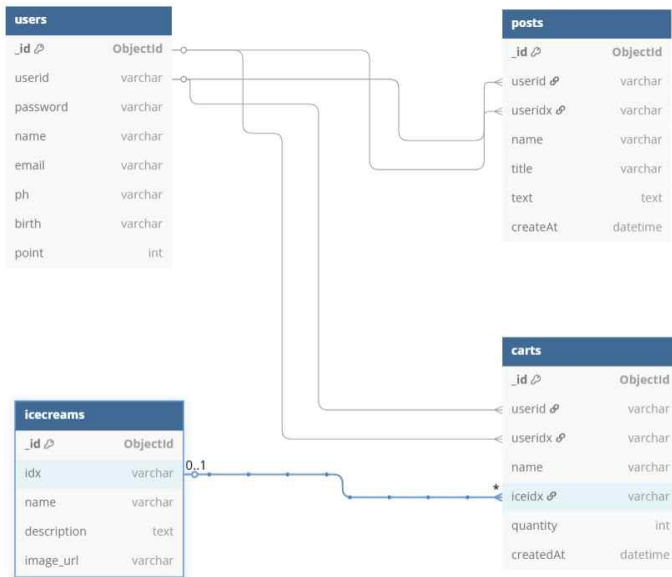
03 Database Schema Relationships(테이블 관계도)

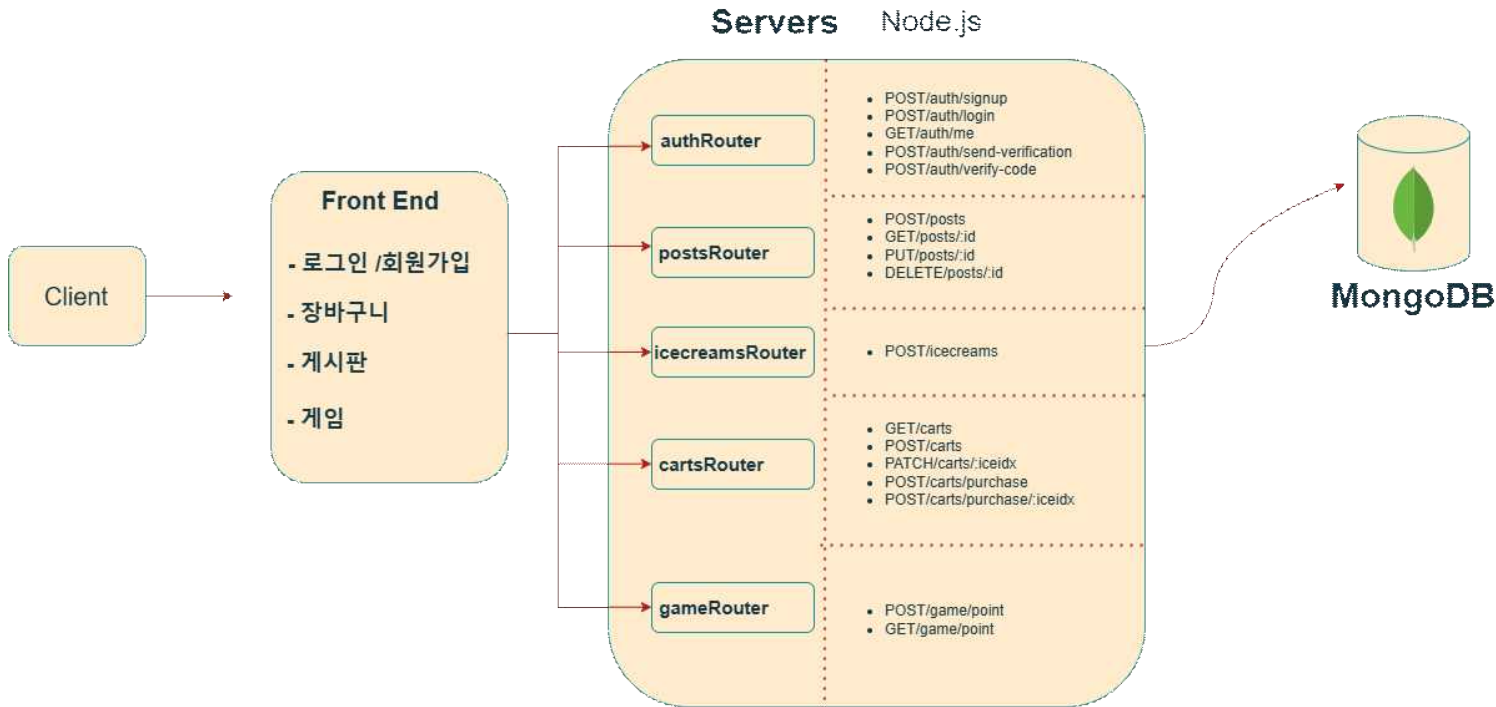
(with dbdiagram.io)

Users - Posts & Carts



Icecreams - Carts





07. 트러블 슈팅 및 향후 개선점

트러블 슈팅

1 Front Convention Guide

2 root에 기반한 구조와 스타일 공통화

3 Script/Console 에러

4 Mongodb Connection 연결

5 coolSMS 문자 인증 서비스 사용

향후 개선점

- 구매자가 지도에서 매장 선택 후 직접 주문 또는 예약 할 수 있는 기능
- 빅데이터 베이스 구축 및 활용(케이크, 커피, 매장 등급 etc...)
- 다양한 게임 추가
- 웹/앱 개발

8. Q&A

Thank you!