



한국ICT 인재개발원 당당 프로젝트

Made By. Dang-Dang

1

당당 프로젝트 개발 사유

◆ 건강하게, 당당하게.

새벽 라디오에서 건강 정보를 듣는데, 내과 전문의 한 분이 “성인병, 아, 아니 생활습관병이...”라고 말을 급히 바꾼다. 성인병, 그러고 보니 오랜만에 듣는 단어다. 예전에 뇌졸중, 암, 심장병을 한데 모아 성인병이라 했다. ‘예전’이라고 집어 말하는 건 사연이 있어서다. 2003년 3월, 대한내과학회의 이름으로 모인 내과 의사들이 성인병을 ‘생활습관병’으로 고쳐 부르기로 합의했다. ‘성인병’은 20년 전 공식적으로 사라졌다. 그해 생활습관병을 알리는 대대적 캠페인이 진행됐다.

◇성인병 이후... 생활습관병과 대사증후군

이미 그때도 성인병이란 이름은 시대착오적이었다. 암이나 심뇌혈관질환이 마흔을 넘겨 집중 발현하는 건 사실이지만, 성인병이란 이름을 주고 나니 ‘성인 이전’ 시기의 징후 관리나 예방에는 둔감했다. 질환 자체를 ‘생활 습관’ 차원으로 재규정하면서 질병을 대하는 자세 자체가 달라졌다. 실제로 대한내과학회는 ‘성인병’ 대신 ‘생활습관병’을 채택하면서 질환의 원인과 종류를 확장했다.

잘못된 식습관이 당뇨병, 비만, 고지혈증, 순환기병, 대장암, 치주병을 유발한다. 운동을 게을리하면 비만, 고지혈증, 고혈압이 생긴다. 흡연은 만성기관지염과 폐기종의 원인이다. 간질환을 유발하는 음주도 빼놓을 수 없다... 이 정도면 우리가 걱정하는 거의 모든 질병을 포괄한다.

‘이름’을 거론하는 중이니 ‘대사증후군’ 명칭도 곱씹을 만하다. 몸 바깥에서 섭취한 음식을 몸 안에서 분해하고 합성해 생체 성분 또는 생명 활동에 필요한 물질로 바꾸고 에너지를 생성하는 일, 필요 없는 것들을 추려 몸 밖으로 다시 내보내는 일을 대사라 부른다. 대사에 문제가 생기면 비만, 고혈압, 당뇨병, 이상지질혈증이 동시에, 복합적으로 발생한다. 그게 대사증후군이다. 그런데 대사증후군의 원인은 뭘까. 전문의들은 운동 부족, 스트레스, 야식과 과식, 인스턴트 식품 섭취 등 ‘잘못된 생활 습관’을 지목한다.



음료 통해 당 과다섭취...“영양성분 표시해야”

서울경제TV 포커스온
구독자 4.3만명

구독

8

공유

공유

저장

...

조회수 236회 2022. 4. 7. #음료 #프랜차이즈 #커피

[앵커]

한국인이 음료를 통해 당 섭취를 가장 많이 한다고 하죠. 커피나 스무디처럼 커피전문점에서 파는 음료도 많이 먹으면 하루 권장량을 초과해 섭취하게 되는데요. 당 함량을 담은 영양성분을 정확하게 공개해야 한다는 지적이 나옵니다. 보도에 김수빈 기잡니다.

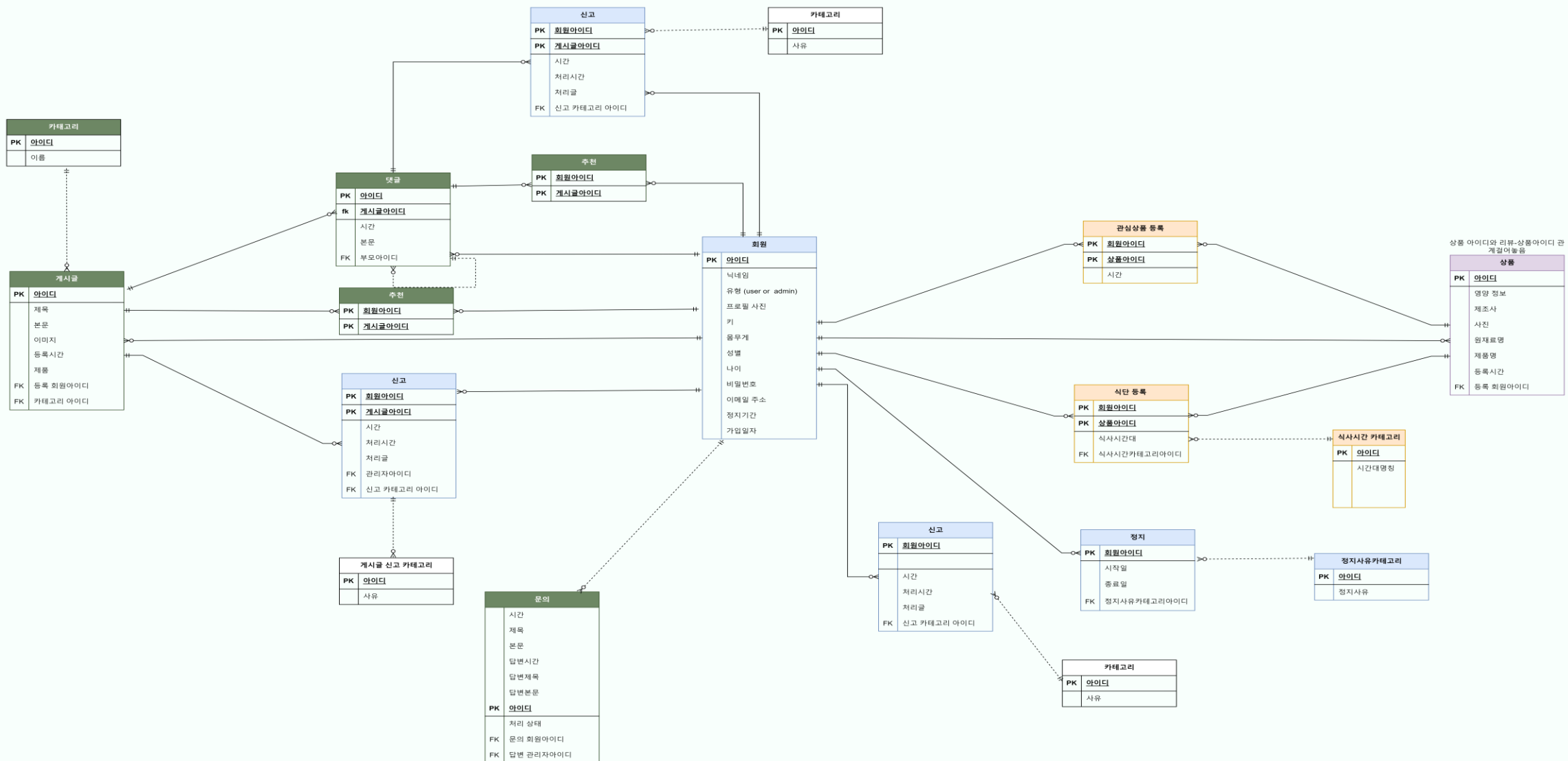
- 대부분의 생활습관병은 식습관과 밀접한 관련이 있습니다.
- 그러나 현대 사회는 음식물을 과다 섭취를 하면서도 과다섭취하고 있다는 것을 알기 힘듭니다.
- 당당은 생활습관병의 원천적인 예방을 위해, 식품명을 쉽게 검색하여 사용자가 직접 영양 성분을 파악하여 과다 섭취를 막고 관리 할 수 있도록 하기 위해 개발되었습니다.

2

당당 프로젝트 개발 환경 및 DB 설계도



DB 설계도 (테이블 수 : 19개)



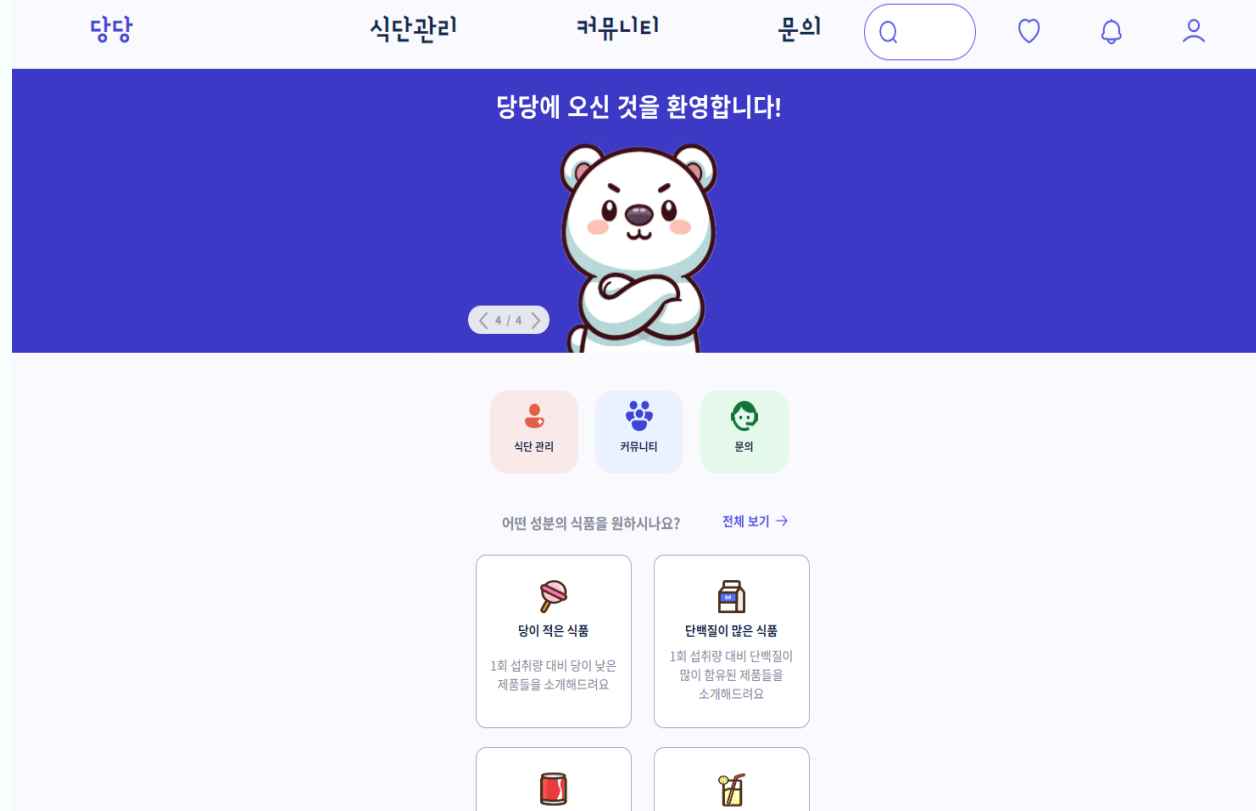
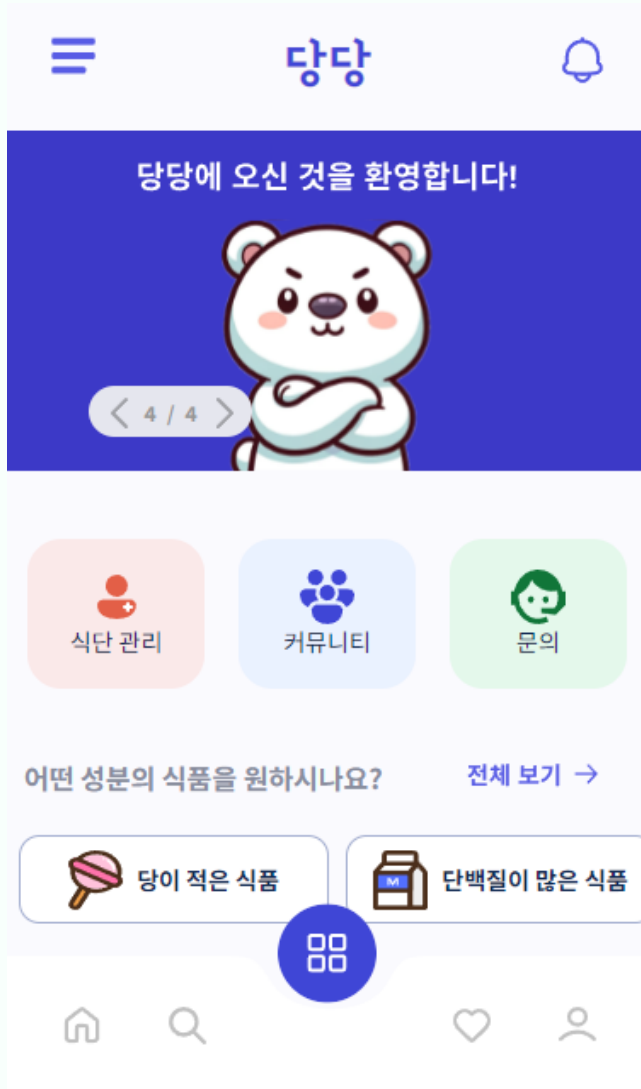
◆ Dev Tools



3 Dang-Dang.co.kr 시연

※시연은 PPT 특성상, 페이지와 소스코드로 대체하도록 하겠습니다.

◆ 당당 메인 페이지



설명

- 당당 프로젝트는 기본적으로 모바일 최적화로 제작되었습니다. (데스크탑 반응형 웹페이지)
- 모바일 페이지에서는 Fab (Floating Action Button), Nav-bar, Tab-bar 등을 적용하여 사용자의 빠른 메뉴 이동 및 최적의 사용 환경을 제공하기 위해 노력하였습니다.

◆ 당당 메인 페이지(소스코드)

```
document.addEventListener("DOMContentLoaded", function() {
  const fabMain = document.getElementById("fab-main");
  const fabContainer = document.querySelector(".fab-container");

  fabMain.addEventListener("click", function() {
    fabContainer.classList.toggle("active");
    fabMain.classList.toggle("rotate");
  });

  // 첫 번째 FAB 아이템에 대한 클릭 이벤트
  const fabItem1 = document.getElementById("fab-item1");
  fabItem1.addEventListener("click", function() {
    window.location.href = '/member/index';
  });

  // 두 번째 FAB 아이템에 대한 클릭 이벤트
  const fabItem2 = document.getElementById("fab-item2");
  fabItem2.addEventListener("click", function() {
    window.location.href = '/qna';
  });

  // 세 번째 FAB 아이템에 대한 클릭 이벤트
  const fabItem3 = document.getElementById("fab-item3");
  fabItem3.addEventListener("click", function() {
    window.location.href = '/user/board/list';
  });
});
```

Floating Action Button

Java Script(이하 JS)를 사용하여
각각의 버튼에 클릭 이벤트를
적용한 코드입니다.

```
document.addEventListener("DOMContentLoaded", function() {
  let menuButton = document.getElementById("menuButton");
  let navBar = document.getElementById("navBar");
  let xButton = document.querySelector(".x_button");

  // 화면 너비가 700px 이상인지 확인
  const mediaQuery = window.matchMedia("(min-width: 700px)");

  if (menuButton) {
    menuButton.addEventListener("click", function(e) {
      e.preventDefault();

      // 미디어 쿼리 조건이 충족되지 않을 경우에만 코드 실행
      if (!mediaQuery.matches) {
        // 콘텐츠를 동적으로 추가
        navBar.style.display = navBar.style.display === "flex";
        navBar.classList.toggle("show");
      }
    });
  }

  if (xButton) {
    xButton.addEventListener("click", function(e) {
      e.preventDefault();
      if (navBar.classList.contains("show")) {
        navBar.classList.remove("show");
      }
    });
  }
});
```

Nav-bar

Nav-bar 역시 JS를 사용하였습니다.

```
<div class="left-menu-button">
  <li>
    <a class="icon icon-size:11 icon-tab_home home" href="/">홈버튼</a>
  </li>
  <li>
    <a class="icon icon-size:11 icon-tab_search search" href="/user/product/list-result">검색버튼</a>
  </li>
</div>
<div class="right-menu-button">
  <li>
    <a class="icon icon-size:11 icon-tab_heart heart" href="">관심상품 버튼</a>
  </li>
  <li>
    <a class="icon icon-size:11 icon-tab_profile profile" href="/member/index">프로필 버튼</a>
  </li>
</div>
```

Tab-bar

TAB-bar의 경우, 동적으로 사용하지 않기 때문에
HTML의 '<a class>'를 사용하였습니다.

◆ 당당 회원 가입 페이지

당당에 오신 것을
환영합니다.

아이디와 비밀번호를 입력해주세요

아이디

비밀번호

비밀번호를 잊으셨나요?

로그인

아이디가 없나요?

회원 가입하기



당당의 회원이
되어주세요.

아이디

ㅇ

닉네임

ㅇ

E-mail

✉

비밀번호

🔒

🔒



회원가입이 완료되었습니다

메인으로

- 당당 프로젝트는 기본적으로 비회원에게도 열려있지만, 당당의 모든 서비스를 제공 받기 위해 회원 가입을 해야합니다.
- 이후 후술할 내용은 회원 가입을 완료하였다는 전제하에 설명하도록 하겠습니다.



당당 회원 가입 페이지(소스코드)

```
inputBoxes[0].onchange = async function () {
  let memberList = await data()

  if (idInput.value !== "") {
    idInput.classList.remove("error")
    idInput.classList.remove("success")
    error[0].style.display = "none"
    success[0].style.display = "none"

    // 사용자가 입력한 아이디가 이미 가입된 아이디인 경우
    if (memberList.find((m) => idInput.value === m.userId)) {
      idInput.classList.add("error")
      error[0].style.display = "block"
      error[0].innerHTML = "중복된 아이디입니다";
      submitBtn.disabled = true;

      // 만약 해당 아이디가 탈퇴 회원인 경우
      const isWithdrawnMember = memberList.find((m) => m.userId === idInput.value && m.userStatus === 0);
      if (isWithdrawnMember) {
        error[0].innerHTML = "사용할 수 없는 아이디입니다";
      }
    } else {
      // 아이디가 존재하지 않는 경우
      idInput.classList.add("success")
      success[0].style.display = "block"
      submitBtn.disabled = false;
    }
  } else {
    // 아이디 입력이 없을 경우
    idInput.classList.remove("error")
    idInput.classList.remove("success")
    error[0].style.display = "none"
    success[0].style.display = "none"
  }
}
```

```
inputBoxes[3].onchange = function () {
  const passwordPattern = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&\(\)\#\^])[A-Za-z\d@$!%*?&\(\)\#\^]{8,}$/;

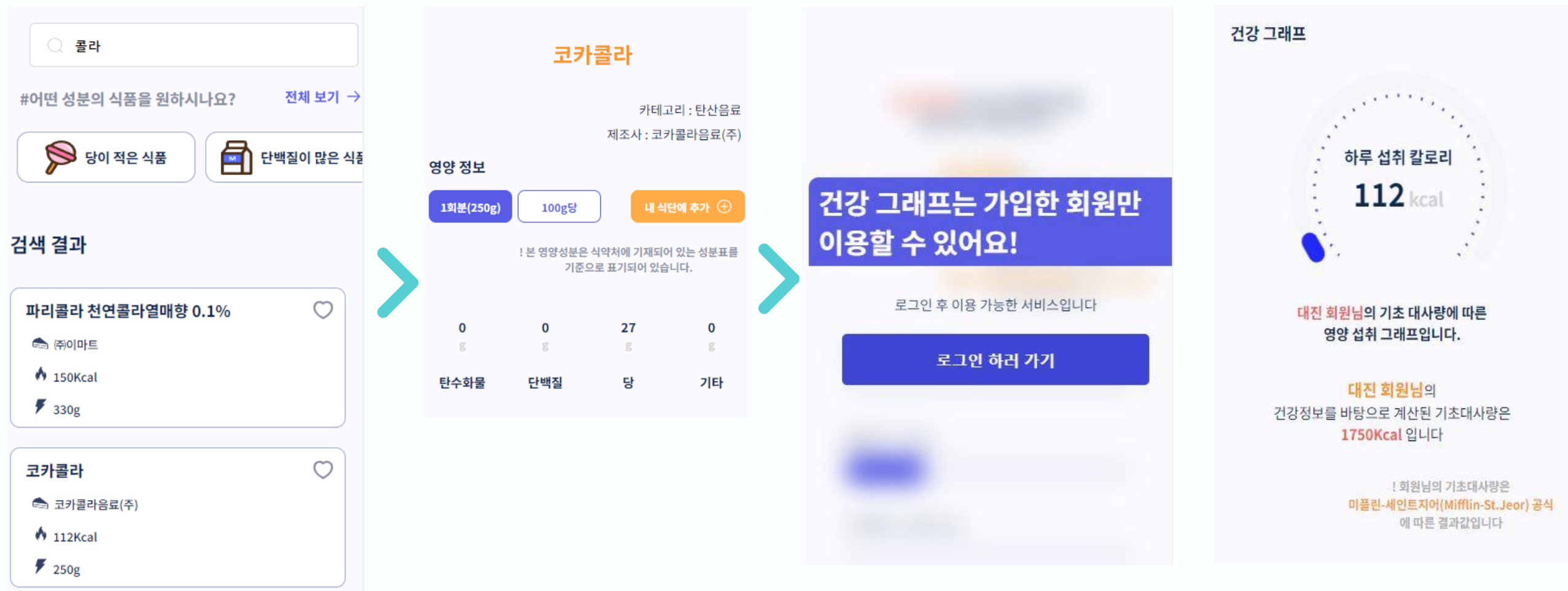
  if (!passwordPattern.test(pwdInput.value)) {
    pwdInput.classList.add("error")
    pwdInput.classList.remove("success")
    error[3].style.display = "block"
    success[3].style.display = "none"
    submitBtn.disabled = true;
  } else {
    pwdInput.classList.add("success")
    pwdInput.classList.remove("error")
    error[3].style.display = "none"
    success[3].style.display = "block"
    submitBtn.disabled = false;
  }

  if (pwdInput.value === "") {
    pwdInput.classList.remove("error")
    pwdInput.classList.remove("success")
    error[3].style.display = "none"
    success[3].style.display = "none"
    submitBtn.disabled = false;
  }
}
```

회원 가입의 제약 조건들

중복 회원 제약, 탈퇴 회원, 비밀번호 입력 제약 조건의 소스 코드입니다.

당당 식품 검색 기능



- 당당 프로젝트의 식품 검색 기능입니다.
기본적으로 비회원도 검색 기능을 사용할 수 있지만, 세부 건강 그래프는 회원에게만 열려 있습니다.
- 영양 성분 Data는 식약처에서 제공하는 오픈 API를 DB로 사용하고 있습니다.
- 또한 입력 된 회원의 키, 몸무게, 성별 등을 계산하여 기초 대사량을 계산하여 그래프화하는 서비스를 제공하고 있습니다.



당당 식품 검색 기능 (소스코드)

```
methods:{
  async bind() {
    let response = await fetch(`http://localhost:8080/api/menus?q=${this.query}`);
    let list = await response.json();
    this.list = list;
  },
  findHandler(){
    this.bind();
  },
  async heartClickHandler(index){
    let curMenu = this.list[index];
    let url = curMenu.isLike ?
      `/api/likes/${curMenu.id}` :
      `/api/likes`;

    let options = {
      method: curMenu.isLike ? "DELETE" : "POST",
      headers: {
        "Content-Type": "application/json",
        "X-CSRF-TOKEN": csrfToken
      },
      body: !curMenu.isLike ? JSON.stringify({ menuId: curMenu.id, memberId: 1 }) : undefined
    };

    let response = await fetch(url, options);
    if (response.ok) {
      let result = await response.json();
      console.log(result);
      curMenu.isLike = !curMenu.isLike;
      curMenu.likeCount += curMenu.isLike ? 1 : -1;
    } else {
      console.error('Request failed:', response);
    }
  }
}
```

```
document.addEventListener("DOMContentLoaded", function() {
  // 로그인 상태에 따라 모달을 표시하거나 숨깁니다.
  const isLoggedIn = !!document.getElementById('sessionId');
  const noLoginHealthModal = new Modal({
    title: '건강 그래프는 가입한 회원만 이용할 수 있어요!',
    content: '로그인 후 이용 가능한 서비스입니다',
    btnText: '로그인 하러 가기'
  });

  // 로그인 상태에 따라 모달에 'blur' 클래스를 추가하거나 제거
  if (!isLoggedIn) {
    noLoginHealthModal.blur();
  }

  fetch('/api/user/details')
    .then(response => {
      if (!response.ok) {
        throw new Error('Network response was not ok');
      }
      return response.json();
    })
    .then(userDetails => {
      const userWeight = userDetails.userWeight
      const userHeight = userDetails.userHeight
      const userGender = userDetails.userGender
      const userYear = userDetails.userYear
```

```
// health-chart 도넛 차트 생성
createHealthChart('health-chart', kcal, bmr);

})
.catch(error => {
  console.error('Fetch error:', error);
});

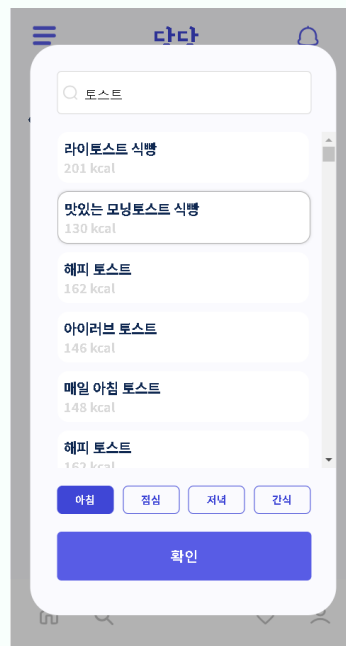
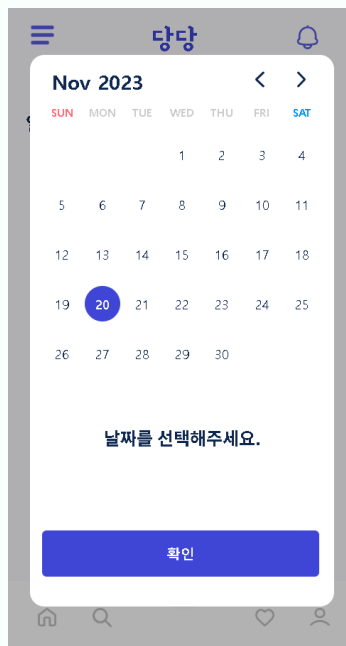
// 사용자의 상세 정보를 이용하여 BMR을 계산하는 함수
function calculateBMR(userWeight, userHeight, userYear, userGender) {
  let bmr = (10 * userWeight) + (6.25 * userHeight) - (5 * userYear);
  if (userGender === "남성") {
    bmr += 5;
  } else if (userGender === "여성") {
    bmr -= 161;
  }

  return bmr;
}
```

식품 검색 기능의 조건, 계산식 구현화

식품 검색 쿼리, 로그인/비로그인 구분, 회원 정보를 이용한 계산식의 소스 코드입니다.

◆ 당당 식단 관리 기능



- 당당 프로젝트의 건강 관리 기능입니다.
식품 검색 기능을 사용한 후, '내 식단에 추가' 버튼으로도 사용할 수 있는 페이지입니다.
- 다이어리의 형식을 취하고 있으며, 날짜를 선택 후 섭취한 식품을 사용자가 아침, 점심, 저녁, 간식 등의 카테고리에 추가할 수 있습니다.
또한 입력 된 식품들은 삭제를 통해 기록에서 지울 수 있게 하여 유연한 관리가 가능하도록 하였습니다.



당당 식단 관리 기능 (소스코드)

GetMapping

```
@RestController("apiEatingController")
@RequestMapping("/api/eatings")
public class EatingController {

    @Autowired
    private EatingService service;

    @GetMapping("/listByMealCategory")
    public List<EatingView> getListByMealCategory(
        @RequestParam("eatingDate") String eatingDate,
        @RequestParam("mealCategoryId") Integer mealCategoryId
    ) {
        Long memberId = memberId();

        List<EatingView> list = service.getProductListByMealCategory(memberId, eatingDate, mealCategoryId);

        return list;
    }

    @GetMapping("/mealCategory")
    public MealCategoryView getMealCategory(
        @RequestParam("eatingDate") String eatingDate,
        @RequestParam("mealCategoryId") Integer mealCategoryId
    ) {
        Long memberId = memberId();

        MealCategoryView data = service.getByMealCategory(memberId, eatingDate, mealCategoryId);

        return data;
    }

    let breakfastResponse =
        await fetch(`/api/eatings/mealCategory?eatingDate=${eatingDate}&mealCategoryId=${breakfastCategoryId}`);
    let lunchResponse =
        await fetch(`/api/eatings/mealCategory?eatingDate=${eatingDate}&mealCategoryId=${lunchCategoryId}`);
    let dinnerResponse =
        await fetch(`/api/eatings/mealCategory?eatingDate=${eatingDate}&mealCategoryId=${dinnerCategoryId}`);
    let snackResponse =
        await fetch(`/api/eatings/mealCategory?eatingDate=${eatingDate}&mealCategoryId=${snackCategoryId}`);

    let breakfastListResponse =
        await fetch(`/api/eatings/listByMealCategory?eatingDate=${eatingDate}&mealCategoryId=${breakfastCategoryId}`);
    let lunchListResponse =
        await fetch(`/api/eatings/listByMealCategory?eatingDate=${eatingDate}&mealCategoryId=${lunchCategoryId}`);
    let dinnerListResponse =
        await fetch(`/api/eatings/listByMealCategory?eatingDate=${eatingDate}&mealCategoryId=${dinnerCategoryId}`);
    let snackListResponse =
        await fetch(`/api/eatings/listByMealCategory?eatingDate=${eatingDate}&mealCategoryId=${snackCategoryId}`);

    let breakfast = await breakfastResponse.json();
    let lunch = await lunchResponse.json();
    let dinner = await dinnerResponse.json();
    let snack = await snackResponse.json();

    let breakfastList = await breakfastListResponse.json();
    let lunchList = await lunchListResponse.json();
    let dinnerList = await dinnerListResponse.json();
    let snackList = await snackListResponse.json();
```

PostMapping

```
@PostMapping
public Eating reg(@RequestBody Eating eating) {

    Long memberId = memberId();

    String selectedDate = eating.getRegDate();
    Long productId = eating.getProductId();
    Integer mealCategoryId = eating.getMealCategoryId();

    Eating data = service.reg(selectedDate, memberId, productId, mealCategoryId);

    return data;
}

saveButton.onclick = async function() {

    console.log(selectedDate, selectedProductId, selectedMealCategoryId);

    let requestData = {
        regDate: selectedDate,
        productId: selectedProductId,
        mealCategoryId: selectedMealCategoryId
    };
    console.log(JSON.stringify(requestData));

    // HTTP POST 요청을 보냄
    let response = await fetch(`/api/eatings`, {
        method: "POST",
        headers: {
            "Content-Type": "application/json"
        },
        body: JSON.stringify(requestData)
    });
};
```

DeleteMapping

```
@DeleteMapping("{id}")
public Eating deleteById(@PathVariable Long id) {

    System.out.println(id);
    Eating eating = service.deleteById(id);

    return eating;
}

// DELETE
let foodCards = document.querySelectorAll(".food-card");
let deleteButton = document.querySelectorAll(".delete");
console.log(deleteButton);

for(let i=0; i<foodCards.length; i++) {
    deleteButton[i].onclick = async function() {

        let id = foodCards[i].getAttribute("data-id"); // data-id 속성에서 ID 값을 읽음
        console.log("음식 카드 ID:", id);
        await fetch(`/api/eatings/${id}`, {
            method: "DELETE"
        })
        // 삭제 요청이 성공한 경우 음식 카드를 동적으로 삭제
        foodCards[i].remove();

    };
}
```

식단 관리 식사 시간대별 식사 기록 구현화

식단 관리 기능의 각 카테고리 별 식사 기록 저장 및 삭제 기능의 소스 코드입니다.

◆ 당당 커뮤니티 페이지

회원님들의 건강식을 올려주세요!



초코파이 어때요?

0

당이 많이 있는 것 같은

남기혁

2023-11-12 조회수 9



NO IMAGE
AVAILABLE

다들 고생하셨습니다

1

술로 적서

대진

2023-11-10 조회수 7



당 적은 음식 추천해

1

참치는 당이 없어요

옥타프대장

2023-11-10 조회수 10



옥타프대장

2023-11-10 15:10

조회수8

당 줄이자!



당을 줄일거예요!

댓글 4



대진

16:33:34

선착순 25분 무료증정



남기혁

12:41:28

ggg



당곰굿즈판매예정

11:17:49

□□□

⋮



게시글 쓰기

작성

카테고리를 입력해주세요



제목을 입력해 주세요.

내용을 입력해주세요

- 당당 프로젝트의 커뮤니티 기능입니다.
기본적으로 비회원도 읽기 기능을 사용할 수 있지만, 댓글 및 글 작성 기능은 회원에게만 열려 있습니다.
- 글/댓글 작성자에게만 수정/삭제가 가능하도록 'modal' 버튼을 구현하였습니다.
- 또한 글 작성시 사진을 첨부하여 업로드가 가능합니다.



당당 커뮤니티 페이지 (소스코드)

```
window.addEventListener('load', function(e){
  let form = document.querySelector('.post-write');
  let submitBtn = form.querySelector('.write-icon');
  let titleArea = form.querySelector('.textarea');
  let contentArea = form.querySelector('.body-textarea');
  let fileInput = form.querySelector('#fileInput');
  let boardCategoryId = form.querySelector('#category').value
  let sessionId = form.querySelector('#sessionId')

  async function submit(e) {
    e.preventDefault();
    if(sessionId){
      let memberId = sessionId.textContent
      let title = titleArea.value;
      let content = contentArea.value;
      let img = fileInput.files[0]

      let formData = new FormData();
      formData.append('title', title);
      formData.append('content', content);
      if (img) {
        formData.append('imgFile', img);
      }
      formData.append('boardCategoryId', boardCategoryId);
      formData.append('memberId', memberId);
      console.log(title)
      console.log(content)
      console.log(boardCategoryId)
      console.log(memberId)

      let response = await fetch("/api/posts", {
        method: "POST",
        body: formData
      })

      if (response.ok) {
        let redirectPath = await response.text();
        window.location.href = redirectPath;
      }
    }
  }
  submitBtn.addEventListener('click', submit);
});
```

```
@PostMapping
public ResponseEntity<> reg(Board board, MultipartFile imgFile, HttpServletRequest request) throws Exception, IOException {
  // 이미지 파일이 없거나 비어 있는 경우 기본 이미지로 설정
  if (imgFile == null || imgFile.isEmpty()) {
    board.setImg("default.png");
  } else {
    // 이미지 저장 로직
    System.out.println("저장전 : " + imgFile.getOriginalFilename());
    String strPath = request.getServletContext().getRealPath("/image/posts");
    File path = new File(strPath);
    if (!path.exists())
      path.mkdirs();
    File file = new File(strPath + File.separator + imgFile.getOriginalFilename());
    imgFile.transferTo(file);
    board.setImg(imgFile.getOriginalFilename());
    System.out.println("저장후 : " + board.getImg());
  }
  System.out.println(board);
  int newOne = boardService.write(board);
  return ResponseEntity.ok().body("/user/board/list");
}

<!-- ===== 글쓴이와 접속한 사람이 일치할 때만 아이콘이 보인다 =====>
<span sec:authorize="isAuthenticated()">
  <span th:if="${#authentication.principal.id} == ${m.memberId}">
    
  </span>
</span>
</div>

<div class="post">
  <div class="post-header">
    
    <div class="post-author-time">
      <span class="post-author" th:text="${m.userName}">책더리퍼</span>
      <span th:text="${#dates.format(m.regDate, 'yyyy-MM-dd HH:mm')}">00:05</span>
    </div>
    <span class="post-views font-size:1">조회수 </span> <span class="view-count font-size:1"
      th:text="${m.hit}">123</span>
    </div>
  <div>
    <h2 class="post-title" th:text="${m.title}">점심은 돈까스</h2>
    
    <div class="detail-content font-size:1" th:text="${m.content}">모든 게 거짓이라고~내게 말해줘</div>
  </div>
</div>
```

커뮤니티 기능의 글쓰기, 회원 식별 구현화

커뮤니티의 글쓰기 기능, 작성자/비작성자 구분, 이미지 파일 저장 등을 구현화한 소스 코드입니다.



```

      /\ /---'-----C)
    (C)\---| | | | | | | | | | \\\ \\
      W _--| | | | | | | C | ) ) )
        ' |___|_|_|_|_|_|_|_|_|_|
          =====|_=====|____/_/_/_/_/
:: Spring Boot ::                (v3.1.1)

2023-11-18T14:32:57.356+09:00 INFO 10312 --- [           main] kr.co.dangdang.web.DangDangApplication : Starting Dang
DangApplication v0.0.1-SNAPSHOT using Java 17.0.9 with PID 10312 (C:\Program Files\Apache Software Foundation\Tomcat 10.
1\webapps\ROOT\WEB-INF\classes started by dang in C:\Program Files\Apache Software Foundation\Tomcat 10.1)
2023-11-18T14:32:57.391+09:00 INFO 10312 --- [           main] kr.co.dangdang.web.DangDangApplication : The following
1 profile is active: "prod"
2023-11-18T14:33:00.391+09:00 INFO 10312 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping
Spring Data JDBC repositories in DEFAULT mode.
2023-11-18T14:33:00.473+09:00 INFO 10312 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Sprin
ng Data repository scanning in 66 ms. Found 0 JDBC repository interfaces.
2023-11-18T14:33:01.633+09:00 INFO 10312 --- [           main] w.s.c.ServletWebServerApplicationContext : Root WebAppli
cationContext: initialization completed in 3402 ms
2023-11-18T14:33:04.659+09:00 INFO 10312 --- [           main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcom
e page template: index
2023-11-18T14:33:05.367+09:00 INFO 10312 --- [           main] o.s.s.web.DefaultSecurityFilterChain : Will secure a
ny request with [org.springframework.security.web.session.DisableEncodeUrlFilter@2c49eefd, org.springframework.security.
web.context.request.async.WebAsyncManagerIntegrationFilter@7a2f1b17, org.springframework.security.web.context.SecurityCo
ntextHolderFilter@2fdcb46, org.springframework.security.web.header.HeaderWriterFilter@6d362f40, org.springframework.secu
rity.web.authentication.logout.LogoutFilter@2b381c76, org.springframework.security.web.authentication.UsernamePasswordA
uthenticationFilter@4674bc88, org.springframework.security.web.savedrequest.RequestCacheAwareFilter@45eb6b7d, org.spring
framework.security.web.servletapi.SecurityContextHolderAwareRequestFilter@2c072835, org.springframework.security.web.aut
hentication.AnonymousAuthenticationFilter@6a662854, org.springframework.security.web.access.ExceptionTranslationFilter@7
```

당당 프로젝트는 Linux 우분투를 이용한 MariaDB에서 운용되고 있습니다.

당당 프로젝트의 웹 서버는

Apache사의 Tomcat으로 운용되고 있습니다.

4

당당 프로젝트 멤버와 후기

◆ 당당 프로젝트 멤버



팀장 김대은

퍼블리싱, 개발 일정 조정,
메인페이지, 게시판 시스템,
DB, 웹서버 담당.



부팀장 남기혁

ERD, DB 설계, 게시판 시스템
담당.



팀원 정대훈

퍼블리싱, 회원 가입/탈퇴,
인증 권한, 에러 페이지 처리 담당.



팀원 이인기

식단 관리 시스템의 스케줄,
식단 추가, 성분 합산 로직 담당.

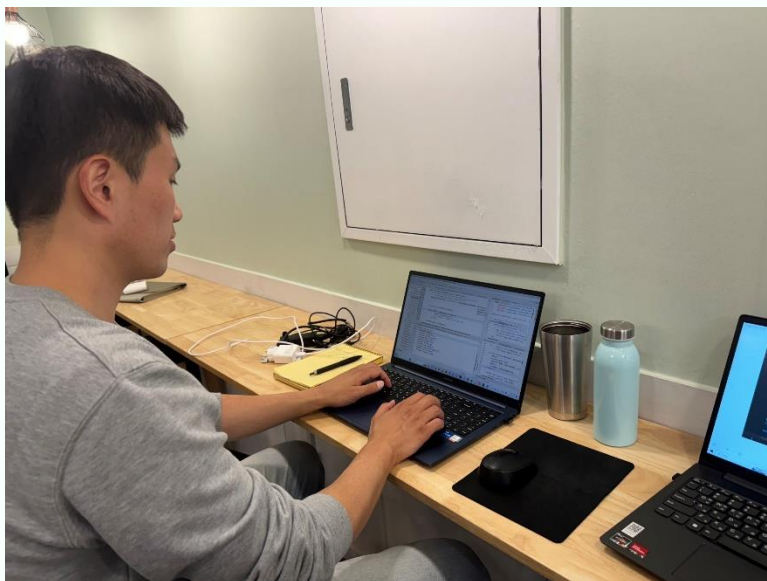


팀원 김대진

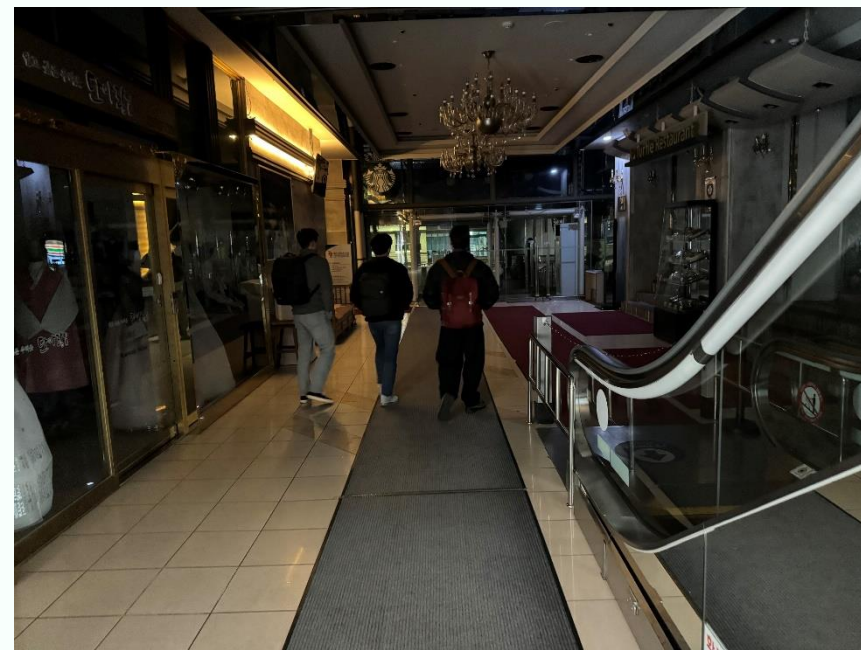
퍼블리싱, 식품 검색,
식품 리스트 구현,
성분 그래프, 식품 좋아요
시스템 담당.



당당 프로젝트 멤버는 더 이상 비전공자가 아닙니다.
5명의 개발자로 나아가겠습니다.



흔히 개발자가 힘들어야
사용자가 편하다고들 합니다.
당당 프로젝트 개발자들의 열정이
사용자의 편리함으로 돌아갔으면 좋겠습니다.



Thank you
감사합니다