

MEDIMATE

서비스명: 메디메이트 MEDIMATE

팀명: E약 언제 먹었Z

contents

01. 서비스 제작 배경 및 필요성

02. 주요 기능 및 메인 타겟

03. 활용 데이터

04. 개발과정 및 방법

05. Information Architecture

06. 화면 및 기능설명

07. 페이지 시연

08. 서비스 활용방안 및 기대효과

09. 프로젝트 이슈

10. 개선사항

11. 개발도구

QnA

01 서비스 제작 배경 및 필요성

취약 시간대 약국에 대한 인식 확대 및 운영 증가

- 노인 인구 증가와 소아 병원 감소 등으로 인해 약국 영업시간 외의 시간대에 의약품이 필요한 상황이 증가 할것으로 보임
- '휴일지킴이 약국'등 이미 존재하는 검색 포털에 대한 인식 저조
- 소비자들의 '의료'에 대한 관심이 증가함에 따라 정부차원의 공공 의료서비스 확대의 필요성이 증가함

PHR에 대한 인식 증가

- 보건의료 마이데이터 (PHR; Personal Health Record) 디지털 헬스케어 산업과 비대면 관련 데이터산업의 시장 규모 증가
- 개인이 복용하는 상비약, 일반의약품 등의 기록(PHR)을 보관하는 공간의 필요성 부각

의약품 복용 관리의 중요성 부각

- 약물 복용을 임의로 중단하면 위험함에도, 복용 불순응의 사유로 '의약의 복용을 잊음'이 44%로 가장 높음
- 복약스케줄에 대한 공유로 보호자의 지속적인 관찰 가능성 必
- 캘린더의 메모 및 공유 기능 활용을 통해 복약의 누락을 최소화

02 주요 기능 및 메인 타겟

약국/의약품 정보 검색 기능

- 약국의 위치, 영업시간 등의 정보
- 일반 상비의약품 판매업소 정보
- 일반 의약품 정보
- 전문 의약품 정보

캘린더 기능

- 복용과 관련한 타임라인 설정 및 알림
- 복용 여부의 체크 및 알림
- 주변 사람과의 공유

타깃 웹사이트 (유사 서비스 예시)

1.전국약방

[\(https://pharmacy.yohan1002.com/\)](https://pharmacy.yohan1002.com/)

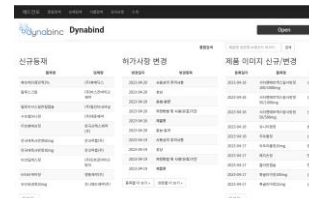
전국의 약국 위치, 영업시간 등
의 정보 검색 기능



2. 메드인포

(<https://www.medinfo.co.kr/>)

국내에서 사용중인 의약품의 정보 검색 기능



‘약국 관련 정보 검색 기능’과 ‘캘린더 기능’을 포함한 웹사이트 기획
건강 관리를 하는 일반인 및 영유아, 노령 환자의 보호자들이 메인 타깃


03 활용 데이터

1. 서울특별시 약국 운영시간 정보 공공데이터
(출처: <http://data.seoul.go.kr/dataList/OA-20402/S/1/datasetView.do>)

2. 의약품안전나라 의약품 낱알식별
(출처: <https://nedrug.mfds.go.kr>)

3. 식품의약품안전처 e약은요 정보검색
(출처: <https://nedrug.mfds.go.kr/searchEasyDrug>)

공공데이터


보건



활용사례(갤러리) 등록

URL 복사

목록 이동

서울시 약국 운영시간 정보

서울특별시 내의 약국의 위치, 주소 및 진료요일(시간) 등을 기준으로 약국 목록정보, 기본정보 및 위치 정보 조회 등의 약국정보 조회 기능 제공

서비스명	의약품 낱알식별		
제공형태	EXCEL/CSV	제공주기	매일 04:00 AM
다운로드	 엑셀다운로드  CSV다운로드		


K-오렌지북

특허 검색

임상시험정보 검색


제네릭의약품 유통정보 검색

e약은요정보 검색



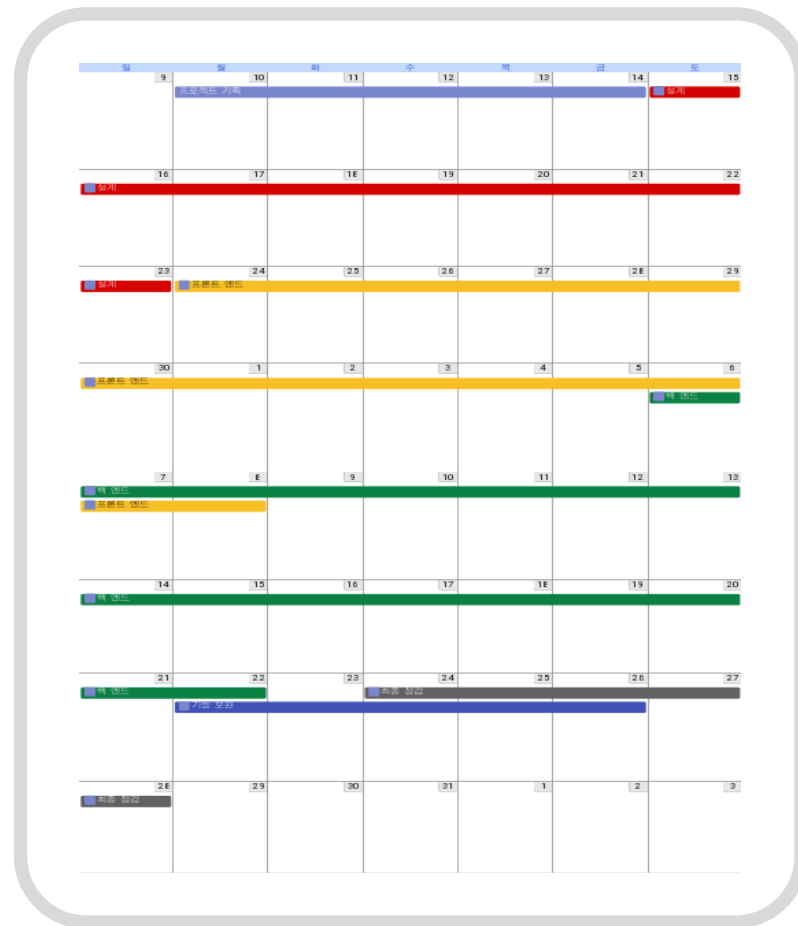
"e약은요"는 일반소비자 눈높이에 맞춘 이해하기 쉬운 의약품 정보 제공을 위해 마련된 의약품개요정보입니다.
의약품에 관한 모든 내용을 담고 있지 않으며 자세한 사항은 식약처 의약품안전나라의 "의약품상세정보"를 참고하시기 바랍니다.
본 정보는 법적 효력을 가지는 것이 아닙니다.

총 4,339 건

 엑셀다운로드

O4 개발과정 및 방법

개발단계	기간	내용
프로젝트 기획	04/10 ~ 04/14	<ul style="list-style-type: none"> 공공 데이터를 이용한 프로젝트 주제 선정 추가 데이터 확보 메인 서비스 기획
설계	04/15 ~ 04/23	<ul style="list-style-type: none"> 공공 데이터 및 추가적인 데이터를 이용한 DB 설계 전체적인 화면 구성 설계 메인 서비스 설계
프론트 엔드	04/24 ~ 05/08	<ul style="list-style-type: none"> HTML, CSS, JS를 이용하여 화면 구현
백 엔드	05/06 ~ 05/22	<ul style="list-style-type: none"> 메인 서비스 및 부가적인 서비스 구현
기능 보완	05/22 ~ 05/26	<ul style="list-style-type: none"> 개발 단계에서 부족했던 부분 보완 및 오류 수정
최종 점검	05/24 ~ 05/28	<ul style="list-style-type: none"> 마지막 기능 점검 및 기획서 마무리



05 Information Architecture

사용자 페이지



관리자 페이지



06. 화면 및 기능설명

06 화면 및 기능설명: 비밀번호 찾기

비밀번호 찾기

이름

아이디

이메일

확인



답장 전체답장 전달 삭제 스팸차단 안읽음 이동 ▾ 더보기 ▾

☆ MEDIMATE

▼ 보낸사람 MEDIMATE <chldudgus07@naver.com> VIP

2023년 5월 29일 (월) 오후 2:44

임시 비밀번호는 cyakx6xkee 입니다.

nodemailer 모듈 사용

06 화면 및 기능설명: 비밀번호 찾기

Middleware

```
import nodemailer from 'nodemailer';
import { config } from '../config.js';

const transporter = nodemailer.createTransport({
  // 사용하고자 하는 서비스
  service: 'naver',
  host: config.senderSmtp, // 'smtp.gmail.com'
  port: config.senderPort, // 587
  auth: {
    user: config.senderEmail, // 'myemail@gmail.com'
    pass: config.senderPass, // 'password486!'
  }
});

export async function emailPost(U_EMAIL, text){
  await transporter.sendMail({
    // 보내는 곳의 이름과, 메일 주소를 입력
    from: `"MEDIMATE" ${config.senderEmail}`,
    // 받는 곳의 메일 주소를 입력
    to: U_EMAIL,
    // 보내는 메일의 제목을 입력
    subject: 'MEDIMATE',
    text
  });
}
```

Red BOX

Yellow BOX

Blue BOX

Service

```
export async function findPw(req, res){
  const { U_ID, U_NAME, U_EMAIL } = req.body;
  const U_PW = Math.random().toString(36).slice(2);
  const url = `${config.base}/main/find/pw`;

  fetch(url, {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify({ U_ID, U_NAME, U_EMAIL, U_PW })
  })
  .then(response => response.json())
  .then(datas => {
    if(!datas.message){
      emailPost(U_EMAIL, `임시 비밀번호는 ${U_PW} 입니다.`);
      res.render('../public/ejs/main/findId', {
        tokenCheck:false });
    }else{
      const message = datas.message;
      res.render('../public/ejs/main/findId', { tokenCheck:false,
        message });
    }
  });
}
```

nodemailer Module 사용

기초 셋팅

sendMail 부분 (PW는 임시 비밀번호로 변경 후 전송)

06 화면 및 기능설명: 약국 검색

MEDIMATE 약국 약 캘린더 공지사항 로그인 회원가입

강남구 강남 검색하기

약국이름	지역	영업시간
강남엑스타약국	서울특별시 강남구 테헤란로4길 6, 지하1층 b106호 (역삼동)	0900 ~ 2100
강남별약국	서울특별시 강남구 역삼로 118, 1층 (역삼동)	1000 ~ 1800
강남위은약국	서울특별시 강남구 강남대로 292, 3층 (도곡동, 방배빌딩)	0900 ~ 1830
강남미소약국	서울특별시 강남구 논현로86길 22, 1층 (역삼동)	0900 ~ 2100
강남씨티약국	서울특별시 강남구 강남대로 416, 1층 (역삼동, 한림빌딩)	0830 ~ 2130
강남생약국	서울특별시 강남구 선릉로 309, 1층 (역삼동)	0930 ~ 1830
강남굿모닝약국	서울특별시 강남구 남부순환로 2615 (도곡동) 국통스타글래스208호	0940 ~ 2000
강남보신약국	서울특별시 강남구 도곡로 516 (대치동)	1000 ~ 2200

1 2 3 4



MEDIMATE 약국 약 캘린더 공지사항 로그인 회원가입

약국번호: C1109813

약국명: 신정내거리중앙약국

주소: 서울특별시 양천구 중앙로 269, 1

대표 전화: 070-8836-4448

영업시간

월요일:	0900 ~ 2100	0900 ~ 2100
화요일:	0900 ~ 2100	0830 ~ 2200
수요일:	0900 ~ 2100	0900 ~ 2000
목요일:	0900 ~ 2100	0900 ~ 2000
금요일:	0900 ~ 2100	
토요일:	0900 ~ 2000	
일요일:	1000 ~ 2000	
공휴일:	0900 ~ 2230	

Pagination기능 + 카카오지도api와 연동

06 화면 및 기능설명: pagination

Middleware

```
export default function pagination(page, count, Limit){
  const currentPage = parseInt(page);

  // 페이지네이션을 위한 계산
  const totalPages = Math.ceil(count / Limit);
  const buttonsToShow = 5; // 표시할 페이지 버튼 수
  let startPage, endPage;

  if (totalPages <= buttonsToShow) {
    // 전체 페이지 수가 표시할 버튼 수보다 작은 경우
    startPage = 1;
    endPage = totalPages;
  } else {
    // 현재 페이지 기준으로 표시할 버튼의 범위 계산
    const halfWay = Math.ceil(buttonsToShow / 2);
    if (currentPage <= halfWay) {
      startPage = 1;
      endPage = buttonsToShow;
    } else if (currentPage + halfWay - 1 >= totalPages) {
      startPage = totalPages - buttonsToShow + 1;
      endPage = totalPages;
    } else {
      startPage = currentPage - halfWay + 1;
      endPage = currentPage + halfWay - 1;
    }
  }

  return { page:currentPage, totalPages, startPage, endPage }
}
```

Red BOX

Yellow BOX

Blue BOX

Service

```
export async function showAll(req, res) {
  const page = req.query.page || 1;
  const P_ADDRESS = req.query.P_ADDRESS;
  const P_NAME = req.query.P_NAME;
  const url = P_NAME
    ? `${config.base}/main/pharm?page=${page}&P_ADDRESS=${P_ADDRESS}&P_NAME=${P_NAME}`
    : `${config.base}/main/pharm?page=${page}`;
  console.log(url);
  fetch(url)
    .then(response => response.json())
    .then(pharmList => {
      const tokenCheck = TokenStorage.getToken() ? true : false;
      let pagination = Pagination(page, pharmList.count, 8);
      pagination.list = pharmList.rows;
      pagination.P_NAME = P_NAME;
      pagination.P_ADDRESS = P_ADDRESS;
      pagination.tokenCheck = tokenCheck;

      res.render('../public/ejs/main/drugstore', pagination)
    });
}
```

총 page 계산
객체를 생성하여 render를 이용해 ejs 파일에 연결
검색시 필요한 총 page를 endPage 동적 변수 설정

06 화면 및 기능설명: 카카오지도 api 연동

ejs

```
<% list.forEach(data=> { %>
  <div class="drug-info" onclick="modal('<%= JSON.stringify(data)%>')">
    <div><%= data.P_NAME %> </div>
    <div><%= data.P_ADDRESS %> </div>
    <div><%= data.P_MON_S %> ~ <%= data.P_MON_C %></div>
  </div>
<% }) %>
```

외부 javascript

```
let map
let markerPosition
let mapContainer = document.getElementById('map')
mapContainer.style.width = '30rem';
mapContainer.style.height = '21rem';
mapOption = {
  center: new kakao.maps.LatLng(36, 123),
  level: 3
};
map = new kakao.maps.Map(mapContainer, mapOption);
let marker = new kakao.maps.Marker({
  position: new kakao.maps.LatLng(0, 0)
});

function modal(data2) {
  const data = JSON.parse(data2);
  const P_LATI = parseFloat(data.P_LATI);
  const P_LONGI = parseFloat(data.P_LONGI);

  position = new kakao.maps.LatLng(P_LONGI, P_LATI);

  marker.setPosition(position);
  map.setCenter(position);
  marker.setMap(map);
  map.setLevel(3)
}
```

Red BOX

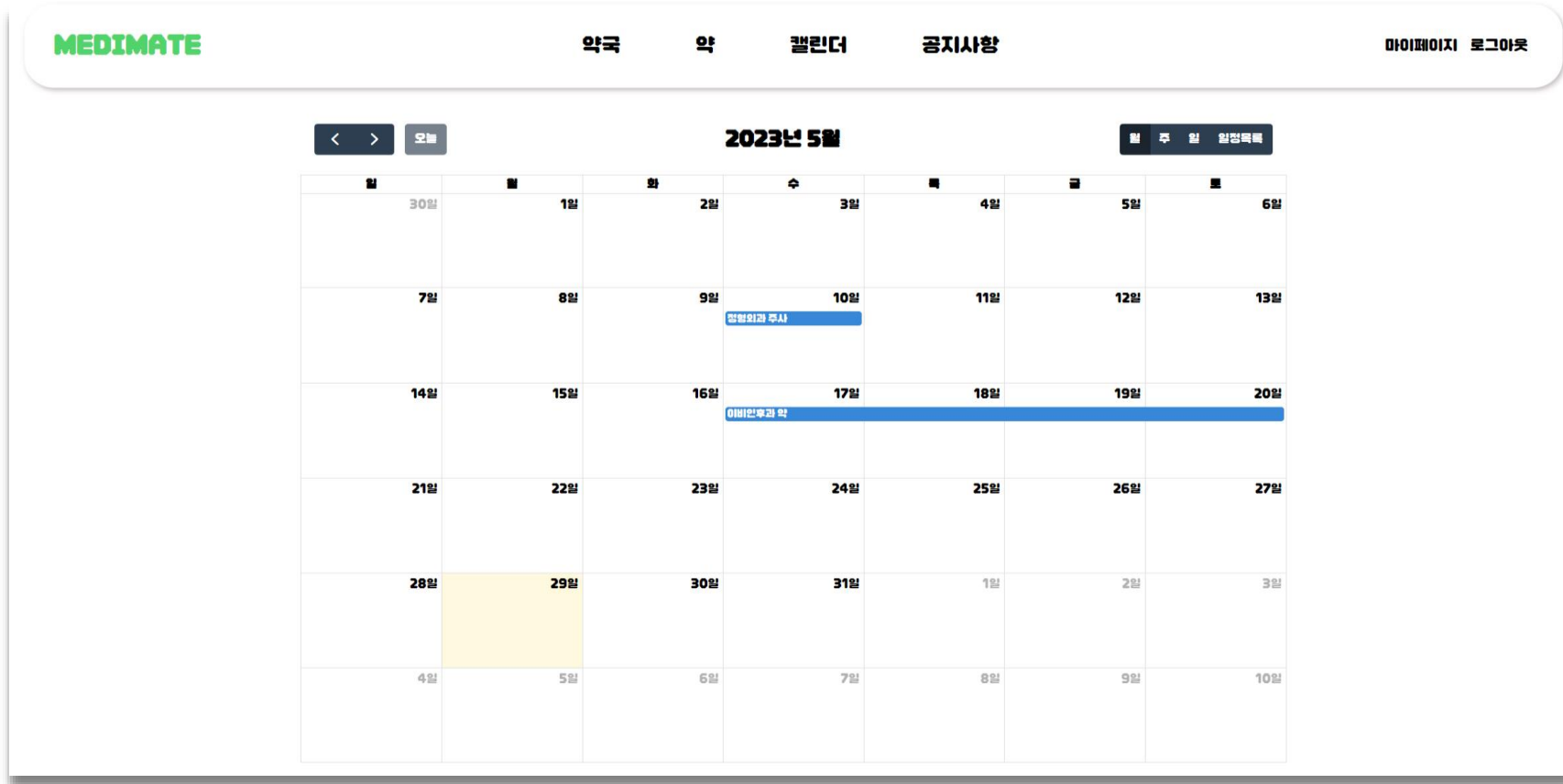
Yellow BOX

Blue BOX

카카오지도 기초화면 설정

data2객체에 담겨 있는 위도 경도를 카카오지도에 적용
ejs로 data를 동적 설정 후 Yellow Box 함수에 적용

06 화면 및 기능설명: 캘린더기능



fullcalendar api와 DB 연결 및 CRUD 적용

06 화면 및 기능설명: 캘린더기능

JavaScript

```
<script>
const events = JSON.parse('<%- JSON.stringify(datas) %>');
(function () {
  $(function () {
    // calendar element 취득
    var calendarEl = $('#calendar')[0];
    // full-calendar 생성하기
    var calendar = new FullCalendar.Calendar(calendarEl, {
      height: '700px', // calendar 높이 설정
      expandRows: true, // 화면에 맞게 높이 재설정
      slotMinTime: '08:00', // Day 캘린더에서 시작 시간
      slotMaxTime: '20:00', // Day 캘린더에서 종료 시간
      // 하단에 표시할 톨바
      headerToolbar: {
        left: 'prev,next today',
        center: 'title',
        right: 'dayGridMonth,timeGridWeek,timeGridDay,listWeek'
      },
      initialView: 'dayGridMonth', // 초기 로드 될때 보이는 캘린더
      // 화면(기본 설정: 달)
      navLinks: true, // 날짜를 선택하면 Day 캘린더나 Week 캘린더로
      // 링크
      editable: true, // 수정 가능?
      nowIndicator: true, // 현재 시간 마크
      selectable: true, // 클릭 일자 드래그 설정가능
      dayMaxEvents: true, // 이벤트가 오버되면 높이 제한 (+ 몇
      // 개식으로 표현)
      locale: 'ko', // 한국어 설정
      eventAdd: function (obj) { // 이벤트가 추가되면 발생하는 이벤트
        console.log(obj);
      },
      // 이벤트
      events: events,
      eventClick: function(event) {
        let action = confirm('내용을 수정 하시겠습니까?');
        if(action){
          let title = prompt('수정 내용:');
          if(title){
            location.href=`/main/calendar/update/${event.event._def.
              extendedProps.C_NUM}?C_TITLE=${title}`;
          }else{
            alert('수정에 실패 했습니다. 내용을 입력하세요.')
          }
        } else{
          action = confirm('일정을 삭제 하시겠습니까?');
          if(action){
            location.href=`/main/calendar/remove/${event.event._def.
              extendedProps.C_NUM}`;
          }
        }
      },
      select: function (arg) { // 캘린더에서 드래그로 이벤트를 생성할 수 있다.
        var title = prompt('내용 입력:');
        if (title) {
          location.href=`/main/calendar/insert?C_TITLE=${title}&C_START=${arg.
            start}&C_END=${arg.end}`;
        } else{
          alert('등록에 실패 했습니다. 내용을 입력하세요.')
        }
        calendar.unselect()
      }
    });
    // 캘린더 렌더링
    calendar.render();
  })();
}());
```

Red BOX

Yellow BOX

Blue BOX

캘린더api 수정기능 (location.href로 data 전송)

캘린더api 삭제기능 (location.href로 data 전송)

캘린더api 추가기능 (location.href로 data 전송)

06 화면 및 기능설명: 캘린더기능

Service

```
export async function create(req, res){
  const { C_TITLE, C_START, C_END } = req.query;

  fetch(config.base+'/main/calendar', {
    method: 'POST',
    headers: getHeaders(),
    body: JSON.stringify({ C_TITLE, C_START, C_END })
  })
  .then(response => response.json())
  .then(() => {
    res.redirect('/main/calendar');
  })
}

export async function modify(req, res){
  const C_NUM = req.params.id;
  const { C_TITLE } = req.query;

  fetch(config.base+'/main/calendar/'+C_NUM, {
    method: 'PUT',
    headers: getHeaders(),
    body: JSON.stringify({ C_TITLE })
  })
  .then(response => response.json())
  .then(() => {
    res.redirect('/main/calendar');
  })
}

export async function drop(req, res){
  const C_NUM = req.params.id;
  fetch(config.base+'/main/calendar/'+C_NUM, {
    method: 'DELETE',
    headers: getHeaders()
  })
}
```

Red BOX

fetch시 Header로 token을 전송하여 사용자 인식

07. 페이지 시연

08 서비스 활용방안 및 기대효과

캘린더기능과 DUR의
시너지 효과

메디메이트와
비대면 진료 플랫폼의
연계

메디메이트를 통한
공공심야약국 제도
홍보 효과 기대

EJS를 사용하면서 발생한 문제

1. undefined에 대한 조건문 사용 불가능
2. JSON 자체 출력이 불가능



Type of 사용으로 예외처리



JSON 자체 문자열 변환 후
객체로 parsing



데이터 정규식

캘린더
- 공유기능
- 상세설정

약품
- 상세검색

Tools

Visual Studio Code



Slack



GitHub



Git

Language

MySQL



HTML



CSS



JavaScript

Skills

Node.js



REST API



KakaoMap API

QnA

Q.

모달창으로 넘어갈 때 상세정보를 가져오는 방식은 어떻게 되나요?

A.

모달창이 열리기 전에 먼저 해당 리스트를 서버에서 받아옵니다. 모달창 출력 시 서버에 접근하는 것이 아니라 리스트에 담겨있는 객체 자체를 가져와 DOM을 사용하여 modal창에 데이터를 출력해주었습니다.

Q.

페이지네이션에서 검색이 유지되는 방법은 무엇인가요?

A.

페이징 처리를 할 때, 검색어를 담고 있는 폼태그에 현재 페이지를 함께담아 실행을 합니다. 그로인해 검색어가 유지되며 페이징처리가 가능하게 되는것 입니다. 더 간단히 이야기하면 페이지를 이동할때마다 이동할 페이지 번호와 검색어를 항상 함께 들고다녀 검색어가 유지가 되는것 입니다.

Q.

회원정보에서 주소도 입력해야하는데 사용 용도가 무엇인가요?

A.

본인의 주소와 가까운곳의 약국 정보를 기본 화면설정에 띄워주는 서비스도 추후 개발할 생각이 있기 때문에 만들어두었습니다.

Q.

다양한 프레임워크들 중에 Node.js를 선택한 이유는 무엇인가요?

A.

학원 커리큘럼 상 프로젝트를 시작하기 직전 수업시간에 진행한 트위터 클론 코딩을 Node.js로 했기때문입니다. 또한, 파이썬의 pip와 유사하게 npm이 있어서 사용할 수 있는 모듈이 다양하고 관리하기에 좋기 때문입니다.

Q.

유사한 서비스들이 이미 존재하는데, 메디메이트만의 차별점이 있나요?

A.

네 있습니다. 첫번째, 캘린더 기능이 구현되어있다는점. 두번째, 정부차원의 지원이 가능하다는 점입니다. 기존에 존재하는 서비스들은 검색기능에만 초점이 맞춰져있지만, 메디메이트는 캘린더기능을 포함하여 일정 관리또한 가능하다는 차별점이 있습니다. 또한 당 서비스는 서울시 공공데이터 경진대회를 통해 기획한 서비스이므로 매해 업데이트 되는 공공데이터를 이용합니다. 더 나아가 서울시에서 컨택을 할 경우 정부차원에서의 지원이 가능하여 더욱 양질의 서비스를 제공할 수 있을것으로 예상합니다.

Q.

구글이나 네이버, 카카오 로그인을 구현할 생각은 없었나요?

A.

카카오 로그인을 초기 기획단계에 계획했고 구현 시도까지 해보았으나 시간상의 이유로 최종단계에서 빼게 되었습니다. 추후 개선사항중 하나로 생각하고 있는 부분이며, 기회가 된다면 공부하여 구현할 수 있도록 할것입니다.

Q.

약국 외에 가정용 상비약을 판매하는 판매처에 대한 고려가 있었나요?

A.

가정용 상비약을 취급하는 편의점에 대한 정보가 공공데이터로 존재했고, 초기 기획에도 포함되어있었습니다. 하지만 데이터의 미비함과 기존 의약품 데이터와의 호환 문제로 인해 수정단계에서 빠지게 되었습니다. 기회가 된다면 추후에 보완할 예정입니다.

Q.

약품검색을 할 때 다양한 조건이 있었을 텐데 검색조건을 '일반/전문', '명칭'으로 선택한 이유가 있나요?

A.

일반의약품과 전문의약품의 분류와 명칭 외에도 성상, 효능 등 여러가지 조건으로 검색할 수 있게 기획했었습니다. 하지만 시간상의 이유로 최종단계에서 빠지게 되었습니다. 또한, 일반의약품/전문의약품 각각의 분류에 대한 공공데이터가 존재했으나 병합을 시도할 때 데이터의 훼손이 많아 불가피하게 제외해야 하는 부분도 있었습니다.

Q.

웹이 아니라 앱으로도 구현할 의향이 있나요?

A.

네 있습니다. 하지만 기획을 웹으로 시작했기 때문에 웹과는 또 다른 앱을 구현하기 위해서는 새로 공부하기 위한 시간 등 많은 시간이 필요할 것으로 예상됩니다.

Q.

캘린더가 너무 기본적인 기능만 보유한 것 같은데, 다른 기능을 추가할지에 대한 고려도 하셨나요?

A.

네, 초기에는 보호자가 영유아, 환자 등의 피보호자의 복약 일정을 확인/관리할 수 있는 캘린더 공유기능과 색상에 따라 일정을 구분할 수 있는 기능도 기획 했었습니다. 추후 개선사항 중 하나로 생각하고 있는 부분이며, 기회가 된다면 공부하여 충분히 구현할 수 있도록 하겠습니다.

Q.

약국이나 약품정보가 업데이트 된다면 어떤식으로 업데이트가 이루어지나요?

A.

메디메이트가 사용하는 데이터는 지속적으로 서울시에서 공공데이터 사이트를 통해 업데이트가 되고 있습니다. 현재로서는 본 서비스로 업데이트가 된다면 소량인 경우, 관리자 페이지에서 직접 입력해줄 수 있습니다. 그리고 많은 양은 현재 DBMS로 사용중인 MariaDB를 통해 업데이트를 진행해야 하나 추후 이 부분에 대해 논의가 이루어진다면 서울시에서 당 서비스로 직접 업데이트가 이루어 질 수 있도록 협의가 필요할 듯 합니다.

Q.

각자가 분담한 역할에 대해 설명 부탁드립니다.

A.

* 이현준(팀장) :

전반적인 프로젝트 총괄 및 배포 작업
Client/Server Routing 구현
사용자/관리자페이지 Pagination 기능 구현
사용자페이지 캘린더 HTML/CSS 및 기능 구현

* 조세연(팀원) :

로그인, 회원가입 페이지 HTML /CSS 구현
마이페이지 문의하기, 회원탈퇴 기능 구현
사용자페이지/관리자페이지 공지사항 HTML /CSS 및 검색기능 구현

* 조윤재(팀원) :

약국/약품 DB와 Server의 Sequelize 기능 구현
사용자페이지 약국검색, 약품검색 HTML/CSS 및 검색기능 구현
관리자페이지 약국관리, 약품관리 기능 구현

* 최영현(팀원) :

회원가입/로그인/ID,PW찾기 포함 전체페이지 Autherization 기능 구현
마이페이지 회원정보변경/비밀번호변경 기능 구현
관리자페이지 회원관리, 문의사항 기능 구현
사용자페이지 약국검색 MAP HTML /CSS 및 기능 구현

프로젝트 소감

이현준

팀원들과 함께 프로젝트를 진행하며 기술적인 것을 함께 고민하고 알려주며 해결해 나가는 과정을 통해 실력 향상에 많은 도움이 되었습니다.

조장으로서의 역량이 개인적으로 부족하다 느꼈지만 이번 프로젝트를 통하여 팀을 이끄는 방법에 대하여 많이 배울 수 있었습니다.

조세연

맨 처음 아이디어를 기획할때는 정말 즐겁고 의욕에 넘쳤었는데 실제로 구현하는 단계에서는 맘대로 되지 않아 참 힘들었습니다.

그럴때마다 팀원들이 있어서 서로 물어보고 알려주고 할 수 있다는 것이 고맙고 든든했습니다.

그리고 그냥 배우는것보다 직접 만들어보는것이 훨씬 도움이 된다는것도 느꼈습니다.

아무쪼록 처음해보는 프로젝트라 서툰 부분도 많았겠지만 그래도 많이 배울 수 있었던 경험이었다고 생각합니다.

조윤재

첫 프로젝트다 보니 기술적으로나 기획적으로 부족한 부분도 많고 진행하면서 막혔던 부분들이 많았습니다.

하지만 팀원들과의 협업을 통해 프로젝트에서 어떤 부분이 중요한지를 충분히 고민해 볼 수 있는 기회가 되었습니다.

최영현

머리털나고 첫 코딩프로젝트였습니다.

긴장도 많이하고 이게 맞나?라는 생각을 많이하게되었지만고민할 시간에 한번더 코드 쳐보는 것이 도움된다는 것을 뼈저리게 느꼈고, 기술은 정말 다양하고 많다는 점도 느꼈습니다. 가장 최종적으로 팀원들이 있음으로써 가능한 프로젝트였습니다.

감사합니다