

I - Tub

(지능형 옥조)

분 반 : 03분반
과목명 : 산학캡스톤디자인1
팀 명 : 얼티밋 (Ultimate)
교수명 : 정현숙
팀 장 : 임대인(20154300)
팀 원 : 정해민(20144748)
서정욱(20154199)
조휘훈(20154219)
박지수(20154280)

목 차

1. 프로그램 소개

주제	3
개발 동기 및 목적	3
기대효과 및 활용방안	3

2. 개발 내용

개발 계획 및 일정	4
개발 환경	4
개발 인원	4

3. 세부 내용

시스템 구성도	5
데이터 수집 및 전처리	5
데이터 베이스	6
웹페이지	8
시제품 부품 구매 계획	19
시제품 예상 그림	19

4. 기타

역할 및 소감	20
참고문헌	20

1. 프로그램 소개

1-1) 주제

지능형 욕조 (I-Tub)

(빅데이터와 IoT 기술을 융합하여 웹을 사용해 서비스하는 플랫폼)

1-2) 개발 동기 및 목적

□ 개발동기

- 샤워는 일주일에 적어도 한번하는 활동, 단순한 위생활동이 아닌 하루를 시작 또는 마무리할 때 하는 중요한 활동이라 생각하여 보다 편리하게 해주는 방법이 있지않을까 생각하다 개발하게 되었음
- 샤워에도 사람마다 취향이 있고 선호하는 것이 있는데 획일화 되는 것이 아쉬웠음, 더 나아가 사용자의 사용데이터, 사람들의 사용 데이터를 수집하여 데이터 분석을 통해 직업별, 날씨별과 같이 샤워라는 활동에 부가적인 서비스를 제공하는 플랫폼을 제작하고자함

□ 개발목적

- 힘든 하루를 보낸 회사원, 학생이나 주부 등 여러 사람들이 자동으로 피로를 풀 수 있게 하기위해
- 개인의 기호를 시스템화 시켜서 가장 좋아하는 환경으로 샤워를 할 수 있게 하기위해
- 빅데이터와 결합시켜 여러 요인들을 이용해 예측해서 사용자의 편의성을 극대화하기 위해

1-3) 기대효과 및 활용방안

□ 기대효과

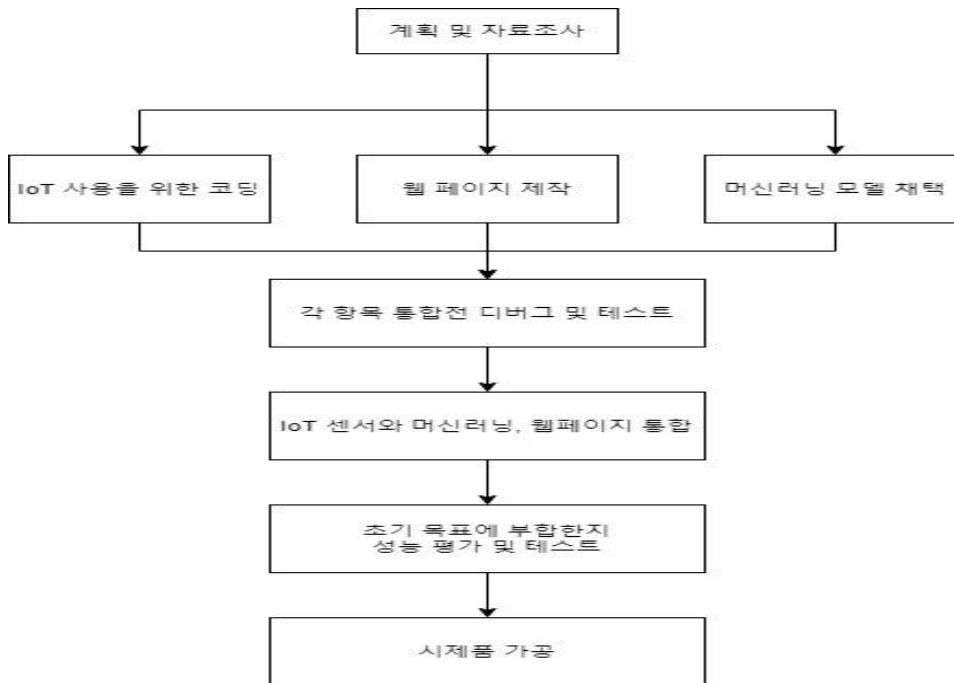
- 이것저것 준비할 필요없이 터치 몇 번으로 환경이 갖추어짐
- 학교, 가사노동, 여러 활동의 피로를 풀 수 있는 환경을 쉽게 마련해 줌
- 편하고 쉽게 사용을 원하는 요즘 젊은 세대와 요구를 충족할 것으로 기대되고 노년층, 어린이들과 같이 여러 연령층이 쉽게 사용할 것으로 기대됨
- 욕조만 있다면 탈부착이 가능한 키트로 어디서든 활용할 수 있어 여러 사업, 집에서 쉽게 설치 가능할 것으로 기대

□ 활용방안

- 샤워/목욕을 하나의 콘텐츠 화 시켜 찜질방, 사우나와는 다르게 사람의 기호별로 다른 방식의 경험을 할 수 있게 함
- 욕조가 없는 환경이더라도 키트의 모양을 약간 변경한다면 사용가능
(부품의 기종을 변경, 내부 SW는 바꿀 필요가 없음)

2. 개발 내용

2-1) 개발 계획 및 일정



추진 내용	수행기간(월) (계획표시 : ■)												비고
	4 월				5 월				6 월				
	1	2	3	4	1	2	3	4	1	2	3	4	
프로젝트 회의 및 주제선정	■	■											
개발언어 및 디자인선정		■	■										
개발 언어, 프로그램 학습			■	■									
데이터 수집 및 정제		■	■	■	■								
개발진행			■	■	■	■	■	■	■	■			
오류최소화 및 리팩토링										■	■	■	
사용자 시연 및 배포												■	■
프로젝트 최종시연 및 정리												■	

2-2) 개발 환경 :

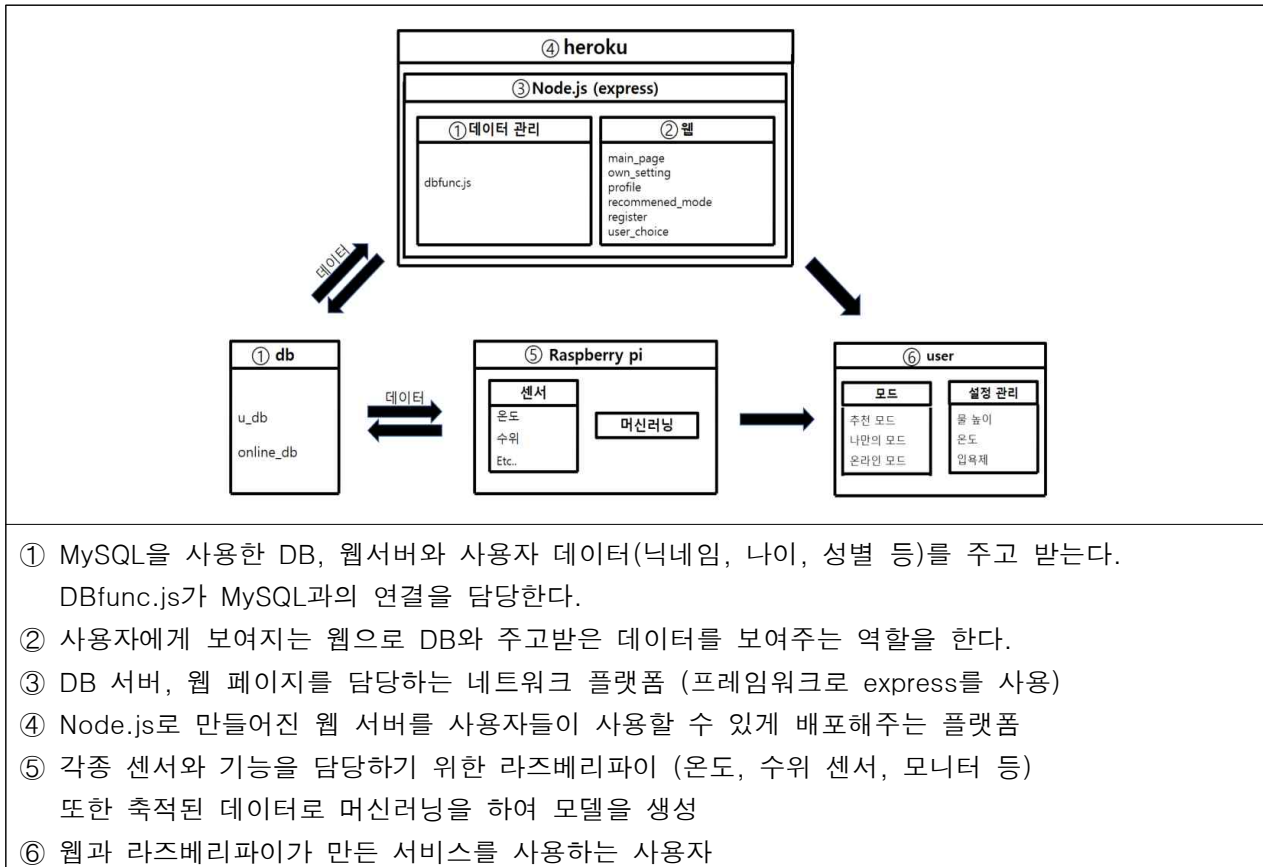
- OS : Microsoft Windows 10 Education (version: 1903, build: 18362)
- DB : MySQL (version: 8.0)
- 웹 : HEROKU(웹 배포, version : 7.39.5), Node.js(웹 서버, version: 12.16.2 LST)
- 소스편집기 : VisualStudio Code (version: 1.38.1)
- 협업 툴 : Github (프로젝트 주소 : <https://github.com/Ultimate-ltubProject>)

2-3) 개발 인원

- 임대인 : 웹 백엔드, IoT 코딩, 데이터 분석, 시제품 제작
- 정해민 : 웹 프론트엔드
- 서정욱 : 웹 프론트엔드, 데이터 분석
- 조휘훈 : 데이터 수집 및 정제
- 박지수 : 데이터베이스 설계, 시제품 제작

3. 세부 내용

3-1) 시스템 구성도



3-2) 데이터 수집 및 전처리

□ 수집 방법

- 조원의 친인척 중 사람을 많이 접하는 직업을 가진 사람이 있어 도움을 받아 설문조사 진행
- 전문 설문조사 사이트에 설문조사를 업로드하여 데이터 수집 실시

설문조사	https://forms.gle/2VM7vNB8BpA2Zspu7 <구글 설문조사> https://www.nownsurvey.com/issue/share/link/1471 <설문사이트>
------	--

□ 수집 데이터

사위/목욕에 관한 조사	설문한 날짜	귀하의 성별은?	귀하의 나이는?	귀하가 원하는 사위/목욕 온도는?	귀하의 사위/목욕 시작 시간대는?	귀하의 사위/목욕 하는 데 걸리는 시간은?	목욕완료시간	사위/목욕 시 일목제 사용 여부	귀하의 직업은?	날씨	목욕시간대 온도
많이 힘에 부칩니다.	2020-04-17	남	12	시원함	16	10	16:10	무	학생	16.30 °C	2020-04-17 16
	2020-04-14	남	12	따뜻함	17	10	17:10	무	학생	20.10 °C	2020-04-14 17
	2020-04-14	남	13	시원함	17	15	17:15	무	학생	20.10 °C	2020-04-14 17
귀하는 성별은?	2020-04-17	남	14	따뜻함	21	10	21:10	무	학생	13.80 °C	2020-04-17 21
<input type="radio"/> 남	2020-04-04	남	14	시원함	17	20	17:20	무	학생	11.10 °C	2020-04-04 17
<input type="radio"/> 여	2020-04-14	남	15	미지근함	16	10	16:10	무	학생	20.40 °C	2020-04-14 16
	2020-04-02	남	15	따뜻함	20	10	20:10	무	학생	10.90 °C	2020-04-02 20
귀하의 나이는?	2020-04-17	남	15	시원함	22	10	22:10	무	학생	12.30 °C	2020-04-17 22
내 답변	2020-04-17	남	15	따뜻함	17	20	17:20	유	학생	16.40 °C	2020-04-17 17
	2020-04-26	남	15	따뜻함	21	20	21:20	무	학생	14.10 °C	2020-04-26 21
귀하가 원하는 사위/목욕 온도는?	2020-04-26	남	16	따뜻함	17	7	17:07	무	학생	18.10 °C	2020-04-26 17
<input type="radio"/> 시원함	2020-04-14	남	16	시원함	20	10	20:10	무	학생	13.80 °C	2020-04-14 20
<input type="radio"/> 미지근함	2020-04-24	남	16	따뜻함	19	20	19:20	무	학생	12.90 °C	2020-04-24 19
<input type="radio"/> 따뜻함	2020-04-28	남	16	따뜻함	20	30	20:30	유	학생	14.30 °C	2020-04-28 20
<input type="radio"/> 기타:	2020-04-28	남	17	미지근함	20	5	20:05	무	학생	14.30 °C	2020-04-28 20
	2020-04-17	남	17	따뜻함	12	10	12:10	무	학생	14.50 °C	2020-04-17 12
귀하의 사위/목욕 시작 시간대는?	2020-04-17	남	17	시원함	23	10	23:10	무	학생	11.00 °C	2020-04-17 23
내 답변	2020-04-27	남	17	미지근함	7	15	7:15	무	학생	7.60 °C	2020-04-27 7
	2020-04-14	남	17	따뜻함	11	15	11:15	무	학생	15.70 °C	2020-04-14 11
	2020-04-28	남	17	미지근함	21	15	21:15	무	학생	13.60 °C	2020-04-28 21
귀하의 사위/목욕 하는 데 걸리는 시간은?	2020-04-17	남	17	따뜻함	23	15	23:15	무	학생	11.00 °C	2020-04-17 23

□ u_db (유저 정보를 저장하는 테이블)

칼럼 이름	칼럼 타입	제약조건	비고
u_num	INT	NOT NULL AUTO_INCREMENT PRIMARY KEY	자동 증가 값
u_name	VarChar(80)	NOT NULL	사용자이름
u_age	INT	NOT NULL	사용자나이
u_gender	INT	NOT NULL	사용자성별
u_job	VarChar(80)	NOT NULL	사용자직업

3-4) 웹 페이지

□ HTML

파일명	설명
main_page.html	메인페이지
user_choice.html	나만의 설정, 온라인모드, 추천모드 선택 페이지
recommend.html	추천 모드 페이지
own_settings.html	나만의 설정 페이지
profile.html	사용자 가입 및 선택 페이지
register.html	회원가입 페이지

□ CSS

파일명	설명
main_page_style.css	메인페이지 스타일 시트
user_choice_style.css	나만의 설정, 온라인모드, 추천모드 선택 페이지 스타일 시트
recommend_style.css	추천 모드 페이지 스타일 시트
own_settings_style.css	나만의 설정 페이지 스타일 시트
profile_style.css	사용자 가입 및 선택 페이지 스타일 시트
register_style.css	회원가입 페이지 스타일 시트
swiper.min.css	웹 슬라이드를 사용하기 위한 스타일 시트
jquery-ui.css	Jquery를 사용하기 위한 스타일 시트

□ JS

파일명	설명
DBfunc.js	데이터베이스를 사용하기 위한 자바스크립트
index.js	웹 페이지 컨트롤러의 역할을 하는 자바스크립트
swiper.min.js	웹 슬라이드를 사용하기 위한 자바스크립트
jquery-3.5.1.js	Jquery를 사용하기 위한 자바스크립트
jquery-ui.js	Jquery의 UI를 사용하기 위한 자바 스크립트



main_page.html

```
<!DOCTYPE html> // HTML5 사용
<html lang="ko">
  <head>
    // 문자 인코딩 지정
    <meta charset="utf-8">
    <title>I-Tub</title> // 페이지 제목 설정
    // 외부 스타일시트와 연결
    <link rel="stylesheet" href="./css/main_page_style.css">
    // 페이지 너비와 기기의 너비를 동일하게 함
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
  </head>
  <body>
    <a href="/profile.html"> // 하이퍼링크 설정
       // 그림 삽입
    </a>
  </body>
</html>
```


닉네임

사용자를 추가시켜 주세요

성별

사용자를 추가시켜 주세요

나이

사용자를 추가시켜 주세요

직업

사용자를 추가시켜 주세요

profile.html

```
<!DOCTYPE html> //HTML5 사용
<html lang="ko"> // HTML 언어 설정

<head>
  <meta charset="utf-8"> // 문자 인코딩 지정
  <title>I-Tub</title> // 페이지 제목 설정
  <link rel="stylesheet" href="/css/swiper.min.css"> // 외부 스타일시트 연결
  <link rel="stylesheet" href="/css/profile_style.css"> // 외부 스타일시트 연결
  <link rel="stylesheet" href="/css/jquery-ui.css"> // 외부 스타일시트 연결
</head>

<body>
  <div class="all-wrap">
    <button onclick=del_div() class="btn">
      <img src='./images/cancelBtn.png' width="100%" height="100%" class="cancel-img">
    </button>
    <div class="swiper-container">
      <div class="swiper-wrapper"></div>

      <div class="swiper-pagination"></div>

    </div>
    <div class="side-wrap">
      <ul class="label-box">
        <li>
          <div class="side-wrap-box">
            <div class="label-title">
              <h3>닉네임</h3>
            </div>
            <div class="label-context">
              <label class="labeling" id='label_name'>사용자를 추가시켜 주세요</label>
            </div>
          </div>
        </li>
        <li>
          <div class="side-wrap-box">
            <div class="label-title">
              <h3>성별</h3>
            </div>
            <div class="label-context">
              <label class="labeling" id='label_gender'>사용자를 추가시켜 주세요</label>
            </div>
          </div>
        </li>
        <li>
          <div class="side-wrap-box">
            <div class="label-title">
              <h3>나이</h3>
            </div>
            <div class="label-context">
              <label class="labeling" id='label_age'>사용자를 추가시켜 주세요</label>
            </div>
          </div>
        </li>
      </ul>
    </div>
  </div>
</body>
</html>
```

```

        <li>
            <div class="side-wrap-box">
                <div class="label-title">
                    <h3>직업</h3>
                </div>
                <div class="label-context">
                    <label class="labeling" id='label_job'>사용자를 추가시켜 주세요</label>
                </div>
            </div>
        </li>
    </ul>
</div>
</div>

<div class="home_button">
    <div class="button_margin">
        <a href="/main_page.html">
            
        </a>
    </div>
    <div class="button_margin">
        <a href="/main_page.html">
            
        </a>
    </div>
</div>

<div id='dialog' title='사용자 선택'>
    <p id='dialog_text'></p>
</div>

<script src="/js/jquery-3.5.1.js"></script>
<script src="/js/jquery-ui.js"></script>
<script src="/js/swiper.min.js"></script>

<script>
    var tmp = '';

    var swiper = new Swiper('.swiper-container', {
        loop: false, // 프로필 루프 설정 (처음->마지막/마지막->처음)

        // 프로필 선택 버튼 설정
        pagination: {
            el: '.swiper-pagination',
            clickable: true,
            dynamicBullets: true,
        },

        navigation: {
            nextEl: '.swiper-button-next',
            prevEl: '.swiper-button-prev',
        },

        on: {
            tap: function () {
                if (tmp.length != 0 && swiper.activeIndex != tmp.length) {
                    $(function () {
                        $('#dialog_text').text('[' + tmp[swiper.activeIndex].u_name + '] 사용자로 시작하시겠어요?');
                        $('#dialog').dialog({
                            //제목 설정
                            title: '사용자 선택',
                            //레이어팝업 넓이
                            width: 400,
                            height: 250,
                            //뒷배경을 disable 시키고싶다면 true
                            modal: true,
                            //버튼종류
                            buttons: [
                                {
                                    //버튼텍스트
                                    text: "확인",
                                    //클릭이벤트발생시 동작
                                    click: function () {
                                        document.location.href = './user_choice.html';
                                    }
                                },
                                {
                                    //버튼텍스트
                                    text: "취소",
                                    //클릭이벤트발생시 동작
                                    click: function () {
                                        $(this).dialog("close");
                                    }
                                }
                            ]
                        })
                    })
                }
            }
        }
    })

```

```

    });
}

// 추가 버튼 클릭시 가입 페이지 이동
else document.location.href = './register.html';
},

// 슬라이드 이동 시 정보 값 불러오기
slideChangeTransitionStart: function () {
    if (tmp.length != 0 && swiper.activeIndex != tmp.length) {
        document.getElementById('label_name').innerHTML = tmp[swiper.activeIndex].u_name;
        if (!tmp[swiper.activeIndex].u_gender) document.getElementById('label_gender').innerHTML =

'남성';

        else document.getElementById('label_gender').innerHTML = '여성';
        document.getElementById('label_age').innerHTML = tmp[swiper.activeIndex].u_age;
        document.getElementById('label_job').innerHTML = tmp[swiper.activeIndex].u_job;
        $('#btn').show();
    }

    else {
        document.getElementById('label_name').innerHTML = '사용자를 추가시켜 주세요';
        document.getElementById('label_gender').innerHTML = '사용자를 추가시켜 주세요';
        document.getElementById('label_age').innerHTML = '사용자를 추가시켜 주세요';
        document.getElementById('label_job').innerHTML = '사용자를 추가시켜 주세요';
        $('#btn').hide();
    }
}
}
});

function del_div() {
    // 사용자 없을 시 경고창 생성
    if (tmp.length < 1 || swiper.activeIndex == tmp.length) { alert('지울 사용자가 없습니다. '); }
    else {
        // 사용자 삭제시 새 화면 생성
        $(function () {
            $('#dialog_text').text('[' + tmp[swiper.activeIndex].u_name + '] 사용자를 삭제하시겠습니까?');
            $('#dialog').dialog({
                //제목 설정
                title: '사용자 선택',
                //레이어팝업 넓이
                width: 400,
                height: 200,
                //뒷배경을 disable 시키고싶다면 true
                modal: true,
                //버튼종류
                buttons: [
                    {
                        //버튼텍스트
                        text: "확인",
                        //클릭이벤트발생시 동작
                        click: function () {
                            sendDB('/db/delete', tmp[swiper.activeIndex].u_num);
                            alert(tmp[swiper.activeIndex].u_name + ' 사용자가 삭제되었습니다. ');
                            window.location.reload(true);
                        }
                    },
                    {
                        //버튼텍스트
                        text: "취소",
                        //클릭이벤트발생시 동작
                        click: function () {
                            $(this).dialog("close");
                        }
                    }
                ]
            });
        });
    }
}

function add_div() {
    console.log(tmp.length);
    // 프로필 선택창이 생성 프로필일 경우 삭제 버튼 숨기기
    if (tmp.length == 0) {
        swiper.appendSlide('<div class="swiper-slide"><div class="vertical"></div></div>');
        swiper.update(); // swiper 갱신
        $('#btn').hide(); // 버튼 숨기기
    }

    else {
        for (var i = 0; i < tmp.length; i++) {

```

```

// 나이에 따라 프로필 이미지 변경
if (tmp[i].u_age < 15) {
    if (tmp[i].u_gender == 0) path = './images/boy.png'
    else path = './images/girl.png'
} else if (tmp[i].u_age >= 15 && tmp[i].u_age <= 65) {
    if (tmp[i].u_gender == 0) path = './images/fa.png'
    else path = './images/ma.png'
} else {
    if (tmp[i].u_gender == 0) path = './images/granfa.png'
    else path = './images/granma.png'
}

// swiper에 부착 및 갱신
var htmlCode = '<div class="swiper-slide"><div class="vertical"><img src=' + path + '
class="imgtag"></div></div>';
swiper.appendSlide(htmlCode);
swiper.update();
}

// 추가 버튼 생성
swiper.appendSlide('<div class="swiper-slide"><div class="vertical"></div></div>');

// 선택된 프로필의 정보값 출력
document.getElementById('label_name').innerHTML = tmp[swiper.activeIndex].u_name;
if (!tmp[swiper.activeIndex].u_gender) document.getElementById('label_gender').innerHTML = '남성';
else document.getElementById('label_gender').innerHTML = '여성';
document.getElementById('label_age').innerHTML = tmp[swiper.activeIndex].u_age;
document.getElementById('label_job').innerHTML = tmp[swiper.activeIndex].u_job;
}
}

// DB 저장 및 결과 확인 함수
function sendDB(url, data) {
    var data = {
        'data': data
    };
    console.log(data + '데이터형 변경 전');
    data = JSON.stringify(data); // json을 문자열 형태로 만들어줌
    console.log(data + '데이터형 변경 후');
    var xhr = new XMLHttpRequest();
    xhr.open('POST', url);
    xhr.setRequestHeader('Content-Type', 'application/json');
    // 서버로 보낼때 json 형태의 데이터를 보내기위해 지정
    xhr.send(data); // 문자열 형태로 넣어줘야함
    console.log(data + 'xhr send')

    // DB 저장 결과 확인
    xhr.addEventListener('load', function () {
        var result = JSON.parse(xhr.responseText);
        var resultDiv;
        console.log('데이터 넘겨받음' + result.result)
        if (result.result == 'ok' && result.type == 'reference') {
            tmp = result.DB;
            add_div();
            console.log('조회가 완료되었습니다.');
        } else if (result.result == 'ok' && result.type == 'delete') {
            console.log('삭제가 완료되었습니다.');
        } else if (result.result == 'none' && result.type == 'reference') {
            add_div();
            console.log('되겠냐?');
        } else if (result.result == 'none' && result.type == 'delete') {
            console.log('되겠냐?');
        }
    });
}

// 페이지 생성시 DB값 불러옴
window.onload = function () {
    sendDB('/db/reference', 'ok');
};
</script>
</body>
</html>

```



recommand.html

```
<!DOCTYPE html> //HTML5 사용
<html>
  <head>
    <meta charset="utf-8"> // 문자 인코딩 지정
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    // 페이지 너비와 기기의 너비를 동일하게 함
    <title>User Chioce</title> // 페이지 제목 설정
    <link rel="stylesheet" href = "/css/recommmed_style.css">
    // 외부 스타일 시트와 연결
  </head>
  <body>
    <div class="wrap"> // 화면 중앙에 오기 위한 영역
      <form action=""> // 사용자 선택 정보 서버에 전송
        <div class="list"> // 선택 버튼을 담기 위한 공간
          </div>
        </form>
      </div>
      <script>
        var parent = document.querySelector(".list"); // list 영역 가져옴
        for(var i = 0; i < 20; i++) {
          var button = document.createElement('button'); // 추가할 버튼 생성
          button.className='btn'; // 버튼 클래스 이름 설정
          button.onclick= function(){ // 각 버튼 클릭시 실행 함수
            alert('구현중');
          }
          var span = document.createElement('span'); // 버튼에 글자를 넣기위한 영역
          var p = document.createElement('p'); // 글자 스타일 변경시 편하게 하기 위해

          p.appendChild(document.createTextNode('추천모드' + (i+1)));
          // 단락 태그에 들어갈 내용 생성

          span.appendChild(p); // span태그에 내용 부착
          button.appendChild(span); // 버튼에 span태그 부착
          parent.appendChild(button); // div영역에 버튼 생성
        }
      </script>
    </body>
  </html>
```

회원가입

별명

성별



남성



여성

나이

직업

취소

가입

register.html

```
<!DOCTYPE html>
<html lang='ko'>

<head>
  <meta charset='utf-8'>
  <meta http-equiv='X-UA-Compatible' content='IE=edge'>
  <title> I-Tub </title>

  <link rel="stylesheet" href="/css/register_style.css">
</head>
<body>
  <form>
    <div class="reg_title">
      <h1>회원가입</h1>
    </div>

    <div class="contents">
      <div class="wrap">
        <h3 class="title"><label id='nickname'>별명</label></h3>
        <span class="inputtext">
          <input type='text' class="texting" id='input_name' />
        </span>
      </div>

      <div class="wrap">
        <h3 class="title"><label for='gender'>성별</label></h3>
        <div class="gender_wrap">
          <div class="item_inp item_radio">
            <input type='radio' name='gender_radio' class="inp_g inp_radio" value='0'
onclick='radio_background(this)' />
            <label class="lab_g lab_radio" for="id_gender_18">
              <span class="ico_account ico_radio" id='radio1'></span>
              <span class="txt_radio">남성</span>
            </label>
          </div>

          <div class="item_inp item_radio">
            <input type='radio' name='gender_radio' class="inp_g inp_radio" value='1'
onclick='radio_background(this)' />
            <label class="lab_g lab_radio" for="id_gender_18">
              <span class="ico_account ico_radio" id='radio2'></span>
              <span class="txt_radio">여성</span>
            </label>
          </div>
        </div>
      </div>

      <div class="wrap">
        <h3 class="title"><label>나이</label></h3>
        <span class="inputtext">
          <input type='number' class="texting" id='input_age' />
        </span>
      </div>

      <div class="wrap">
        <h3 class="title"><label>직업</label></h3>
        <span class="inputtext">
          <input type='text' class="texting" id='input_job' />
        </span>
      </div>

      <div class="wrap">
        <div class="btn_wrap">
          <div class="btn_div">
```

```

        <button type='button' class="btn" onclick='location.href="/profile.html">취소</button>
    </div>
    <div class="btn_div">
        <button type='button' class='submit btn'>가입</button>
    </div>
</div>
</div>
</div>
</form>

<script>
// 가입 버튼 클릭시 실행 함수
document.querySelector('.submit').addEventListener('click', function() {
    alert('등록 완료'); // 경고/알림창 띄우기
    var inputdata = {}; // DB에 저장시킬 변수
    var radio_btn = document.getElementsByName('gender_radio');
    // 라디오 버튼의 값 저장

    // 선택된 성별 반환
    function getRadio() {
        for (var i = 0; i < radio_btn.length; i++) {
            if (radio_btn[i].checked == true) return radio_btn[i].value;
        }
    };

    // 각 항목을 json형태로 변환 후 저장
    inputdata = {
        'u_name': document.getElementById('input_name').value,
        'u_gender': getRadio(),
        'u_age': document.getElementById('input_age').value,
        'u_job': document.getElementById('input_job').value
    }

    console.log(inputdata)
    // 배열로도 정의 가능
    sendDB('/db/create', inputdata) // 아래의 sendDB 함수 실행
    location.href = './profile.html' // 화면 이동
});

// 전송받은 값들을 DB 저장
function sendDB(url, data) {
    var data = {
        'data': data
    };
    console.log(data + '데이터형 변경 전');
    data = JSON.stringify(data); // json을 문자열 형태로 만들어줌
    console.log(data + '데이터형 변경 후')

    // 서버로 보낼때 json 형태의 데이터를 보내기위해 지정
    var xhr = new XMLHttpRequest();
    xhr.open('POST', url);
    xhr.setRequestHeader('Content-Type', 'application/json');

    xhr.send(data); // 문자열 형태로 넣어줘야함
    console.log(data + 'xhr send')

    // 서버에서 정상적으로 데이터 넘겨 받았는지 확인하기 위한 함수
    xhr.addEventListener('load', function() {
        var result = JSON.parse(xhr.responseText);
        var resultDiv;

        console.log('데이터 넘겨받음' + result.result)
        if (result.result == 'ok' && result.type == 'create') {
            resultDiv = document.querySelector('.create');
            resultDiv.innerHTML = '가입테스트 : ' + '가입이 완료되었습니다.'
        } else if (result.result == 'none' && result.type == 'create') {
            resultDiv = document.querySelector('.create');
            resultDiv.innerHTML = '되겠냐?';
        }
    });
}

// 성별 버튼 클릭시 버튼이 바뀌게 선택
function radio_background(e) {
    var radio1 = document.getElementById('radio1');
    var radio2 = document.getElementById('radio2');

    // 남성 선택시 각 버튼 이미지 변경
    if(dc[0].checked==true) {
        radio1.style.backgroundColor="-400px -30px"; // 선택된 버튼으로
        radio2.style.backgroundColor="-430px -30px"; // 기존 버튼으로
    }
    // 여성 선택시 각 버튼 이미지 변경
    else {
        radio1.style.backgroundColor="-430px -30px"; // 기존 버튼으로
        radio2.style.backgroundColor="-400px -30px"; // 선택된 버튼으로
    }
}
</script>
</body>
</html>

```



user_choice.html

```
<!DOCTYPE html> //HTML5 사용
<html>

<head>
  <meta charset="utf-8"> // 문자 인코딩 지정
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  // 페이지 너비와 기기의 너비를 동일하게 함
  <link rel="stylesheet" href="/css/user_choice_style.css">
  // 외부 스타일시트 연결
  <title>추천 모드</title> // 페이지 제목 설정
</head>

<body>
  <div class="wrap"> // 중앙 정렬 및 영역을 가로로 배치
    <div class="ldiv"> // 좌측 크기 설정
      <a href="/own_setting.html"> // 하이퍼 링크 생성
         // 그림 클릭시 이동
      </a>
    </div>

    <div class="col-wrap"> // 우측 크기 설정
      <div class="col-div"> // 하이퍼 링크 영역 크기 설정
        <a href="/recommend_mode.html"> // 하이퍼 링크 생성
           // 그림 클릭시 이동
        </a>

      </div>
      <div class="col-div"> // 하이퍼 링크 영역 크기 설정
        <a href="/user_choice.html"> // 하이퍼 링크 생성
           // 그림 클릭시 이동
        </a>

      </div>
    </div>
  </div>
</body>
</html>
```


own_setting.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>User Chioce</title>
    <link rel="stylesheet" href = "../css/own_setting.css">
  </head>
  <body>
    <div class="wrap">
      <form action="">
        <div id="bathing_section" class="section">
          <span class="title">입욕제 - <span id="bathing_text"></span></span>
          <div class="list">
            </div>
        </div>
        <div>
          <div class="section">
            <span class="title">온도 - <span id="range_text">30</span>&degC</span>
            0<input name="temperature" id="range_slider" type="range" min="0" max="40" value="30" >40
            <p>슬라이더를 조절하여 온도를 설정할 수 있습니다.</p>
          </div>
          <div id="timeSection" class="section">
            <span class="title">타이머 - <span id="timer_text"></span></span>

            <input name="time" id="r0" type="radio" value="설정 안함">
            <label for="r0">설정 안함</label>

            <input name="time" id="r1" type="radio" value="10">
            <label for="r1">10분</label>

            <input name="time" id="r2" type="radio" value="20">
            <label for="r2">20분</label>

            <input name="time" id="r3" type="radio" value="30">
            <label for="r3">30분</label>

            <input name="time" id="r4" type="radio" value="40">
            <label for="r4">40분</label>

            <input name="time" id="r5" type="radio" value="50">
            <label for="r5">50분</label>

            <input name="time" id="r6" type="radio" value="60">
            <label for="r6">60분</label>
          </div>
          <button type="submit">저장</button>
        </div>
      </form>
    </div>

    <script>
      // 입욕제 항목들 생성 부분
```

```

var parent = document.querySelector(".list");
for(var i = 0; i < 100; i++) {
    // 버튼 영역 생성
    var div = document.createElement('div');
    div.className = 'bathing_item'

    // 라디오 버튼 생성
    var label = document.createElement('label');
    var input = document.createElement("input");
    input.type = 'radio';
    input.name = 'bathing_agent';
    input.value = '입욕제' + (i+1);
    label.appendChild(input);

    // 라디오 버튼에 내용 설정
    var span = document.createElement("span");
    span.className = 'bathing_item_text';
    span.appendChild(document.createTextNode('입욕제' + (i+1)));

    // list영역에 삽입
    label.appendChild(span);
    div.appendChild(label);
    parent.appendChild(div);
}

//입욕제 리스트에서 아이템 클릭시 상단 타이틀에 선택한 입욕제가 출력되게 해주는 부분
var bathingItemList = document.querySelectorAll(".bathing_item label");
for(var i = 0; i < bathingItemList.length; i++) {
    bathingItemList[i].addEventListener("click", (e) => {
        document.querySelector("#bathing_text").innerHTML = e.target.value;
    })
}

// 타이머 설정시 타이머 타이틀에 선택한 시간이 출력되게 해주는 부분
var timeList = document.querySelectorAll("#timeSection input");
for(var i = 0; i < timeList.length; i++) {
    timeList[i].addEventListener("click", (e) => {
        var text = e.target.value;
        console.log(e.target.value);
        if(text !== '설정 안함') {
            text += "분"
        }
        document.querySelector("#timer_text").innerHTML = text;
    })
}

var range = document.querySelector("#range_slider"); // 온도 설정의 range slider.
var value = document.querySelector("#range_text"); // 온도 설정의 온도 표시 텍스트.
range.addEventListener("input", (e) => {
    value.innerHTML = e.target.value; // range의 현재 값을 읽어와 온도 표시 텍스트에 저장.
}, false)
</script>

</body>
</html>

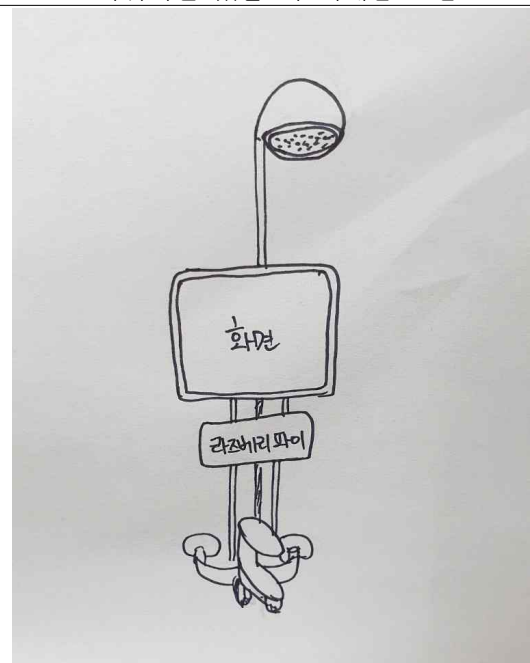
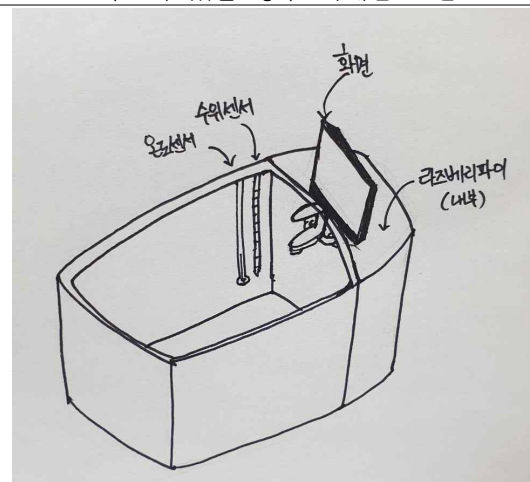
```

3-5) 시제품 부품 구매 계획

항목	비용	
	산출 내역	예상 금액
재료비	라즈베리파이4 스타터 키트 * 2	210,000
	라즈베리파이4 전용 LCD 터치스크린 7인치	105,000
	소켓 점퍼 케이블 40P * 3	2,550
	토양 습도 센서 모듈	1,500
	모터드라이버 모듈	1,800
	DC 모터 (5V)	1,200
	스텝모터 (5V)	1,800
	방수형 온도센서	3,900
	투명 아크릴 (2T, 285x425mm) * 3	12,300
	수축튜브 * 4	1,000
	포맥스 (2T, 300x450mm) * 3	6,300
	합계	347,350원

(Linc 사업단 승인 완료이며 예산이 나오는 데로 구매 예정)

3-6) 시제품 예상 그림

샤워기만 있을 시 시제품 모습	설명
	<p>욕조가 없을 경우의 모습으로 화면아래 라즈베리파이가 연결되어있으며 센서가 아래의 수도꼭지를 상하좌우 운동을 시켜 물의 온도, 세기를 조절함</p> <p>방수 기능은 고민해야할 사항</p>
욕조가 있을 경우 시제품 모습	
	<p>욕조가 있을 경우의 모습으로 앞의 공간에 들어가 있어 물에 보호를 받음 샤워기 모델과 마찬가지로 센서들이 물의 온도, 세기를 조절하며 추가로 수위 조절 기능도 가능함</p>

4. 기타

4-1) 역할 및 느낀점

임대인 - 웹을 개발하는 것은 처음인데 여러 API 문서와 정보를 찾으며 생각보다는 쉽게 한 것 같습니다. 평소에는 혼자서 끙끙 앓으며 왜 안 되는지도 모르고 짜증이나 났을 텐데 팀원들과 같이 문제점을 찾고 해결하니 보다 효율적으로 일을 처리할 수 있었던 것 같습니다. 그리고 협업이라는 것을 제대로 해보는 것은 이번이 처음인데 왜 협업 툴을 사용하는지 느낄 수 있었던 것 같습니다. 그리고 API 문서를 읽으며 사용해보니 앞으로 어떤 처음 보는 오픈소스를 사용하더라도 크게 헤매지 않고 해결할 수 있을 것 같습니다. 그리고 백엔드 분야는 처음 해보는데 서버가 열리는 방식, 사용자가 사용할 수 있게 배포하는 방법, 어떠한 이벤트를 처리하는 방식, 프론트엔드와 협업하는 방법을 배울 수 있던 좋은 기회인 것 같습니다. 이제 라즈베리파이를 구매하면 공부하게 될 것들이 기대됩니다.

정해민 - 서정욱 팀원을 도와 웹 프론트엔드 역할을 맡아 웹페이지 제작을 하였습니다. 몇 년 전 html, css, javascript를 가볍게 접해 본 적은 있었지만, 시간이 너무 흘러 머릿속에 거의 남아있지 않아 막막했습니다. 하지만 팀원들과 상의하고, 유튜브나 생활 코딩과 같은 온라인 강의와 구글링을 활용하여 극복해 나갔습니다. 또한 무엇보다 실력 있고 열심히 하는 좋은 팀원 분들과 함께 캡스톤을 하게 되어서 좋은 경험을 하고 있습니다.

서정욱 - 이번 캡스톤 디자인을 진행 하면서 처음으로 웹페이지 제작을 해보았습니다. 내가 웹페이지를 손댈 것이라곤 생각도 안 해보았기 때문에 정말 기초중의 기초밖에 알지 못했고, 난감하기도 하였습니다. 주로 화면에 보이는 디자인 부분을 수정하였는데, 디자인 소질이 영 없어서 어떤 식으로 배치하고, 색감은 어떻게 줘야 할 지 막막했었습니다. 또한 어떻게 잘 보여야 할 지 너무 어렵게만 생각하기도 하였습니다. 팀원들의 도움을 받아 디자인을 추천 받고, 이를 보여주는 방식으로 극복할 수 있었습니다. 기업에서 각자 역할이 왜 필요한지 느낄 수 있었던 경험 이였습니다.

조휘훈 - 프로그래밍 능력이 가장 떨어지다 보니 능력에 맞춰서 데이터 수집 및 정제라는 역할을 맡게 되었습니다. 가장 쉬운 역할이지만 단답형 설문으로 받은 데이터를 정제하기가 제 생각보다 쉽지 않았고, 시간은 기준을 24시로 잡을지 오전 오후로 작성할지 이런 내용을 정제하면서 팀원들의 도움을 많이 받았습니다. 쉬운 일도 혼자 잘 하지 못하는 제 모습이 별로였지만 뛰어난 팀원들 덕에 문제를 해결할 수 있음이 좋았습니다. 좋은 팀원을 만나 제 부족한 점을 조금이나마 채울 수 있어서 좋은 경험 이였다고 생각합니다.

박지수 - 시제품 구상에 있어서 내가 불편했던 점이나 내가 바랬던점을 많이 생각했던 것 같습니다. 더 구체적인 시제품을 구상할 수 있도록 다양한 의견과 생각을 들어봐야 할 것 같습니다. 그리고 혼자 진행했던 데이터베이스설계보다 더 생각할 것이 많았습니다. 내 생각대로 데이터베이스 설계 틀에 맞춰 끼워 넣던 방식에서 여러 사람들의 생각을 합쳐 조언을 받고 설계를 진행함에 있어서 실질적으로 사용되어야 할 부분과 사용되지 않을 부분을 더 자세히 구분할 수 있게 발전한 것 같습니다. 언뜻 간단해보였었지만 기능이 추가될수록 사이에 얽힌 관계성을 찾거나, 관계성이 존재하지 않다는 것을 구분하는 게 많이 힘들었던 것 같습니다. 더 완벽한 데이터베이스 설계를 할 수 있도록 노력해야겠고, 다 같이 진행하는 프로젝트인 만큼 혼자 할 때보다 더 좋았습니다.

4-2) 참고문헌

swiper 슬라이드 API 문서 : <https://swiperjs.com/api/>