

I - Tub(Intelligent-Tub)

「빅데이터와 IoT 기술을 융합한 욕조 서비스 플랫폼」

분 반 : 03분반
과 목 명 : 산학캡스톤디자인1
팀 명 : 얼티밋 (Ultimate)
교 수 명 : 정현숙
팀 장 : 임대인(20154300)
팀 원 : 정해민(20144748)
서정욱(20154199)
박지수(20154280)

목 차

1. 프로그램 소개

주제	3
개발 동기 및 목적	3
기대효과 및 활용방안	3
기존 유사 연구	4

2. 개발 내용

개발 계획 및 일정	4
개발 환경	5
개발 인원	6
기본 설계	6

3. 세부 내용

I-Tub 시스템 아키텍처	7
I-Tub 데이터 수집 및 정제	8
I-Tub 데이터 베이스	9
I-Tub 머신러닝 모델 선정	10
I-Tub 웹페이지	13
I-Tub 시제품	26
I-Tub 결론 및 향후 보완점	30

4. 기타

역할 및 소감	31
참고문헌	32

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

목욕은 남녀노소 국적, 나이 불문하고 누구나 하는 행위이다. 하루를 시작할 때 혹은 마무리할 때 하는데 사람들은 단순히 위생활동으로만 생각한다는 점이 아쉬웠다. 물을 데우고 타월을 사용해 바디워시를 바르고 물로 행구고, 가장 중요한 활동임에도 너무나 간단히 끝냈다고 생각한다. 우리는 그 점을 착안하여 목욕이라는 행위를 단순한 위생활동에서 휴식이라는 느낌을 강조하고 즐거움을 추가시켜 목욕을 가치 있게 하기 위해 개발을 시작하게 되었다.

사람들은 욕조를 사용하기를 꺼린다. 귀찮기도 하고 준비하는 시간이 너무 오래 걸려 일이 되기 때문이다. 물을 틀어놓고 적당한 온도인지 체크하고 물이 적당한 높이까지 채워졌는지 보고 입욕제를 넣었다면 잘 녹았는지 체크하고 일의 반복이다. 여기서 이러한 과정들이 자동화하고 또 다른 사람들의 목욕방법을 사용하며 최고의, 최적의 환경을 만들어준다면 사람들이 목욕을 즐거워하고 출근하기 전에 한다면 힘을 북돋아 주고 퇴근 후에 한다면 하루 동안 쌓였던 스트레스를 해소시켜줄 것이라고 생각하였다. 여기에 빅데이터와 인공지능 분야까지 합한다면 더욱 효과적일 것이라고 생각하였다.

□ 개발목적

취미 못지않게 많은 요리를 해주는 PC방, 소금방, 얼음방과 같이 테마가 있는 찜질방, 공포, 미스테리와 같이 취향에 따라 즐길 수 있는 숙박시설, 요즘 이러한 콘텐츠들이 많이 유행을 하고 있다. 하지만 목욕에 관한 부분은 전무하였다. 그렇기 때문에 이런 목욕도 콘텐츠화하여 새로운 사업 아이템으로 활용을 할 수 있고 저렴한 가격으로 쉽게 이용할 수 있는 플랫폼을 만들어 판매 혹은 대여를 하면 누구나 양질의 서비스를 받을 수 있게 하고자 하였다. 그리고 사람마다 목욕의 스타일이 다르다. 몇 십분동안 뜨거운 물에 몸을 담구며 피로를 푸는 사람이 있기도 하고 시원한 물에 짧게 몸을 담귀 스트레스를 푸는 사람도 있다. 이렇게 다양한 사람이 있는데 그것을 데이터화하여 나이면 나이, 직업이면 직업별로 다른 서비스를 제공하고 또 인공지능 기술을 사용하여 예측까지 해주어 맞춤형 서비스와 사용성을 극대화 하고자 한다.

1-3) 기대효과 및 활용방안

□ 기대효과

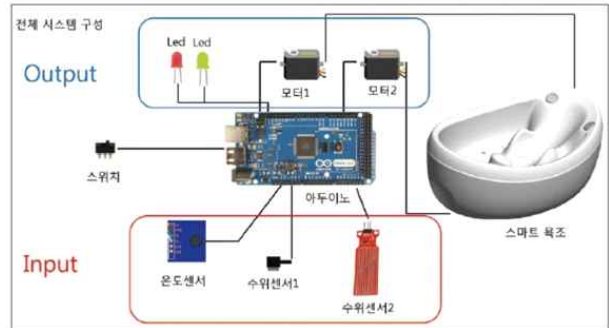
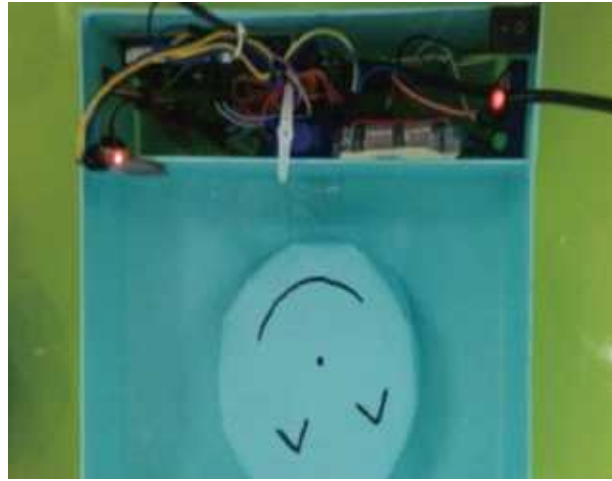
- 이것저것 준비할 필요없이 터치 몇 번으로 환경이 갖추어짐
- 학교, 가사노동, 여러 활동의 피로를 풀 수 있는 환경을 쉽게 마련해 줌
- 편하고 쉽게 사용을 원하는 요즘 젊은 세대와 요구를 충족할 것으로 기대되고 노년층, 어린이들과 같이 여러 연령층이 쉽게 사용할 것으로 기대됨
- 욕조만 있다면 탈부착이 가능한 키트로 어디서든 활용할 수 있어 여러 사업, 집에서 쉽게 설치 가능할 것으로 기대

□ 활용방안

- 샤워/목욕을 하나의 콘텐츠화 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)

1-4) 기존 유사 연구

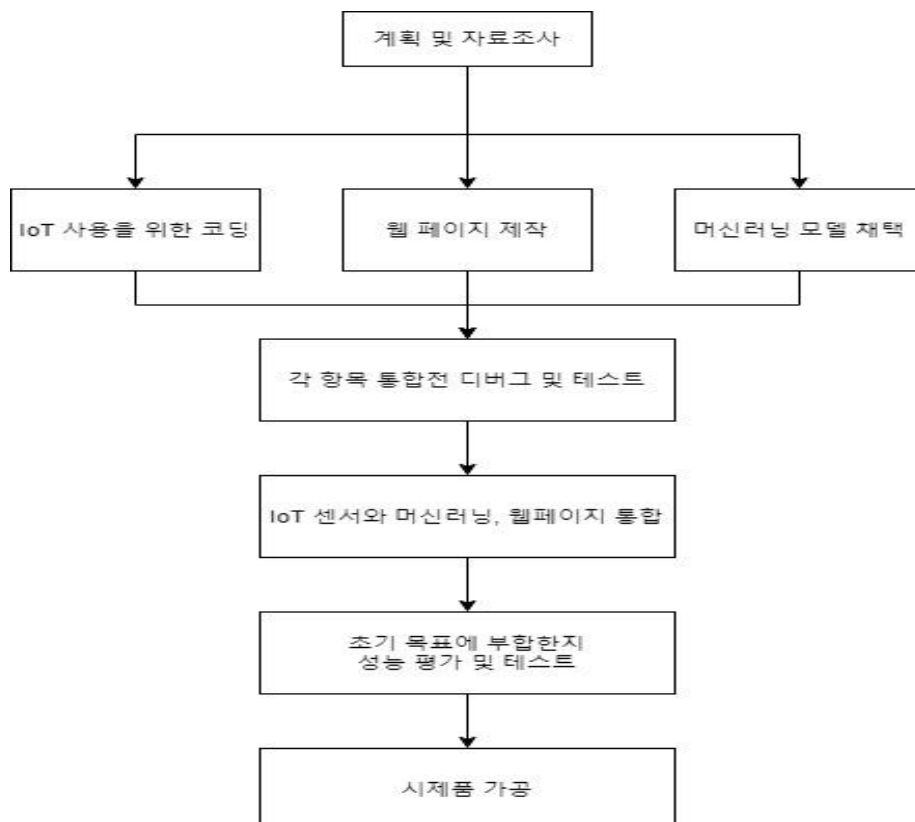
2016년 부산대학교 ICT 융합 프로젝트 공모전 참가상 수상작



16년 부산대학교에서 ICT 융합 프로젝트 공모전에 나온 작품으로 본 프로젝트와 유사한 점을 발견할 수 있었다. 해당 작품의 주제는 ‘스마트한 기능을 추가한 욕조’이며 물의 양 조절, 물의 온도 설정 및 유지, 자동 배수의 기능을 한다. 자동 배수 기능을 제외한 기능은 본 프로젝트와 거의 같다. 해당 작품은 단순히 목욕을 할 때 자동으로 환경을 만들어 주는 것을 기획했지만 해당 기능을 포함하고 여기에 더해 ‘빅데이터와 머신 러닝을 활용한 사용자 맞춤형 스마트 욕조’이기 때문에 단순한 자동화 시스템이 아닌 IoT에 추가적인 기술요소를 결합하여 보다 강화된 사용자 맞춤 서비스를 제공할 수 있을 것이라 생각된다.

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												■	

주	주차 별 과제 계획
1	과제 아이디어 도출을 위한 배경 조사, 온라인을 통한 아이디어 회의 및 팀 소개
2	최종 프로젝트 주제 선정, 필요 데이터 및 부품 확인
3	부품결정 및 주제 수정, 모형 제작 및 Web&DB 학습, 설계 진행
4	DB설계 및 모형 제작(3Dmax)완료, Brackets Editor로 웹페이지 제작
5	GitHub로 협업 개발, 웹 디자인 구상 및 개발
6	웹 서버개설 및 배포, DB연동 시연, 데이터정제 및 기본 데이터 구성 개발
7	DB 수정 사항 확인, 웹 성능 테스트 및 디자인개발
8	중간 검토
9	최종산출물을 위한 개발 진행
10	개발 완료 및 시연
11	사용자 시연 중 오류 발생 최소화
12	사용자 최종시연 확인 및 발표준비
13	산학캡스톤디자인 경진대회 참가
14	산학캡스톤디자인 개인 소감 작성 및 프로젝트 정리
15	최종 결과보고

2-2) 개발 환경 :

프로젝트 진행 간 개발환경은 다음과 같음 (모든 인원이 동일)

- 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)
- IDE : Python IDLE (version : 3.7)
- 협업 툴 : Github (프로젝트 주소 : <https://github.com/Ultimate-ItubProject>)

개발환경의 설치순서는 다음과 같음.

1. 운영체제(OS) : Microsoft Windows 10 Education (version: 1903, build: 18362)
2. 소스편집기 : VisualStudio Code (version: 1.38.1)
3. 웹 서버 및 배포 툴 : Node.js(웹 서버, version: 12.16.2 LST), HEROKU(웹 배포, version : 7.39.5)
4. DB : MySQL (version: 8.0)
5. IDE : python IDLE (version : 3.7)

웹 서버로 사용하는 Node.js의 경우 자바스크립트 엔진을 기반으로 만들어진 플랫폼이며 자바스크립트를 이용하여 서버측에서 하는 역할을 수행해준다. 이를 통해 많이 사용하는 PHP, JSP의 역할을 대체할수 있다 그리고 I/O처리를 비동기로 처리하고 이벤트 위주라 어떠한 값을 원할 때 기다리는 것이아닌 그때 그때 요청을 할때마다 받을 수 있고 쉽게 이벤트 알고리즘을 사용하여 원하는 값을 받아올 수 있다는 강점

이 있다. 또한 빠른 속도를 가지고 데이터 스트리밍과 같이 실시간으로 데이터를 다루는 서비스에 적합한 플랫폼이라고 할 수 있다. 그렇기 때문에 실시간으로 설정값을 라즈베리파이와 주고 받는 것이 중요한 I-Tub 프로젝트와 적합한 플랫폼이라 판단하여 사용하게 되었다.

MySQL은 관계형 데이터베이스 관리 시스템 (RDBMS)로 널리 알려진 표준 SQL을 사용한다. 또한 여러 가지의 프로그래밍 언어를 지원하여 호환성이 뛰어나며 사용자의 용도에 맞게 쉽게 수정을 할 수 있다. 무엇보다 무료 오픈소스 라이선스를 따르기 때문에 ORACLE과 같은 유료 DB보다는 성능과 기능이 떨어지지만 쉽게 접근할 수 있다는 강점이 있다. 온라인 상에서 많은 예제와 예러 해결법이 공개되어있어 초보자도 쉽게 사용할 수 있다는 장점도 있다. 또한 파이썬, 자바스크립트에서도 쉽게 데이터베이스와 연동이 가능하여 사용하게 되었다.

파이썬(Python) 언어는 머신러닝을 쉽게 접목 시킬 수 있는 언어이며 인터프리터 방식을 활용하여 결과를 즉시 확인할 수 있다. 또한 데이터를 쉽게 분석할 수 있고 시각화까지 활용할 수 있다는 장점이 있다. 일반적으로 머신러닝, 데이터 전처리와 시각화에 사용된다. 우리는 여기서 머신러닝, 데이터 전처리 및 시각화 모두에 사용하게 되었다.

헤로쿠(Heroku)는 PaaS(Platform-as-a-service) 클라우드 서비스이며 소규모 웹 서버를 쉽게 배포할 수 있는 플랫폼이다. 한정된 자원을 무료로 사용할 수 있으며 여러 애드온이 있어 추가를 통해 쉽게 웹 서버와 연동시킬 수 있다는 장점이 있다. 웹 사이트가 생성되면 그 사이트만을 위해 설계되어 고립된 성질을 가지게 되고 그로 인해 스케일링이 쉽고 접속량이 많아져도 크게 영향을 받지 않는다는 장점이 있다. 그점이 본 프로젝트에 적합하다고 판단되어 사용하게 되었다.

협업툴인 깃허브(Github)는 마이크로소프트 사에서 지원하는 분산 버전 관리 시스템으로 모든 작업이 기록되고 추적이 가능하다는 점이 큰 이점이다. 로컬 저장소와 원격 저장소가 분리되어있어 실수로 로컬 저장소를 지우더라도 손쉽게 복구할 수 있고 잘못된 작업을 했을 때도 코드 몇 줄로 다시 복구가 가능하여 사용하게 되었다.

2-3) 개발 인원

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

2-4) 기본 설계

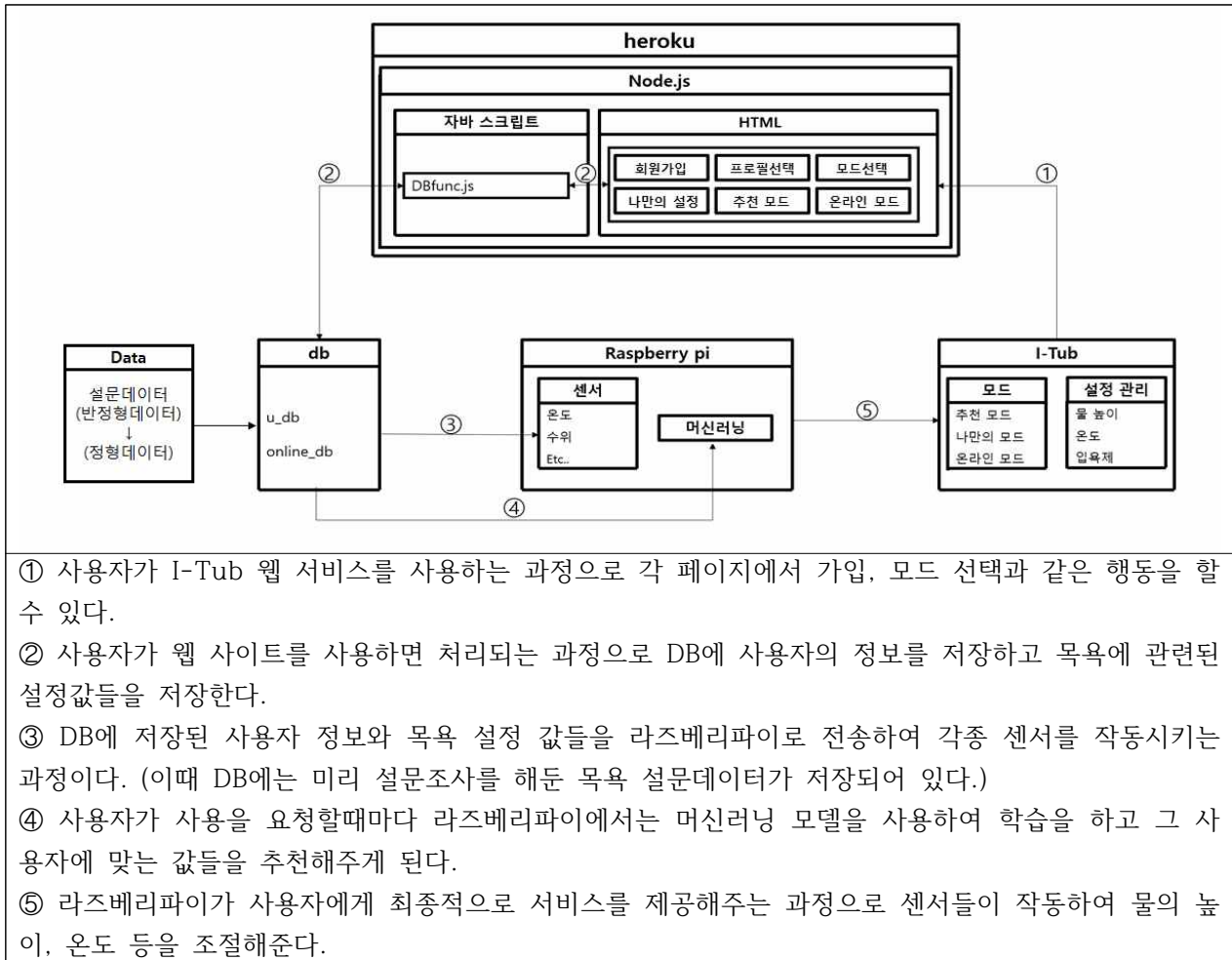
1. 데이터 수집 : 목록에 대한 데이터가 전무하기 때문에 직접 필요항목을 판단하여 데이터를 수집
 - 1-1) 설문조사 전문 사이트를 사용하여 데이터를 수집 (나우앤서베이의 설문광장을 이용)
 - 1-2) 구글폼을 사용하여 친인척, 외부 인원에게 데이터를 수집
 - 1-3) 조사 항목은 반정형 데이터로 데이터 전처리 과정을 통하여 정형 데이터로 변경한다.
 - 1-4) 프로젝트 종료시까지 위의 과정을 반복한다.
2. 데이터 전처리 및 시각화, 예측 모델 학습
 - 2-1) 수집한 데이터에서 불필요한 요소를 삭제하고 (샤워의 종료시간 등) 의도와 다르게 설문을 한 항목에 대해서 제거 및 데이터 전처리 작업을 실시
 - 2-2) 데이터 전처리가 완료된 최종 데이터를 바탕으로 시각화를 하여 사용하기 적합한 데이터인지 서로 상관관계가 있는지 확인을 실시
 - 2-3) 최종 데이터로 후에 사용자 서비스 예측을 위한 머신러닝 모델들을 파이썬으로 작성하여 실행해 보고 성능을 비교하여 확정 (RandomForest, XGBoost, DecisionTree 등의 모델들을 테스트함)
3. 웹사이트 구현
 - 3-1) 사용자에게 서비스를 제공할 웹사이트의 틀을 제작 (Node.js, Heroku, VScode를 사용하여 제작)
 - 3-2) 회원관리, 학습한 예측 모델과 연동, 통계 기능 등의 웹 기능을 구현
 - 3-3) 로컬환경이 아닌, 실제 웹 배포를 하여 실 기능들이 정상 작동하는지 확인
 - 3-4) 웹 서비스 실시

4. 시제품 제작

- 4-1) 웹 서비스를 통해 작동이 될 모형의 틀을 설계
- 4-2) 학교 링크 사업단과 연계하여 모형 제작에 필요한 부품을 특정하고 구매
- 4-3) 센서와 웹 서버간의 통신을 위해 코딩 작업을 실시 (VScode, Python을 사용하여 제작)
- 4-4) 웹과 모형 사이의 연동 테스트 실시
- 4-5) 시제품 서비스 실시

3. 세부 내용

3-1) I-Tub 시스템 아키텍처



3-2) I-Tub 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>
------	--

□ 설문조사를 통한 수집 데이터 (구글폼 991명, 설문조사 전문 사이트, 그 외의 방법 2299명으로 총 3290명 수집)
수집한 항목은 아래의 표와 같고 설문조사를 실시하기 전 목록을 할 때 고려하는 요소와 (샤워 시작시간, 입욕제 사용유무 등) 구현 시 예측에 필요한 요소를 (나이, 직업 등) 더해 총 9가지의 요소를 수집하였다.

샤워/목욕에 관한 조사	설문한 날짜	귀하의 성별은?	귀하의 나이는?	귀하가 원하는 샤워/목욕 온도는?	귀하의 샤워/목욕 시작 시간대는?	귀하의 샤워/목욕 하는데 걸리는 시간은?	목욕완료시간	샤워/목욕 시 입욕제 사용 여부	귀하의 직업은?	날씨	목욕시간대 온도
답이 형에 바랍니다.	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
귀하의 성별은?	2020-04-14	남	13	시원함	17	15	17:15	무	학생	20.10 °C	2020-04-14 17
	2020-04-17	남	14	따뜻함	21	10	21:10	무	학생	13.80 °C	2020-04-17 21
<input type="radio"/> 남 <input type="radio"/> 여	2020-04-04	남	14	시원함	17	20	17:20	무	학생	11.10 °C	2020-04-04 17
	2020-04-14	남	15	미지근함	16	10	16:10	무	학생	20.40 °C	2020-04-14 16
귀하의 나이는?	2020-04-02	남	15	따뜻함	20	10	20:10	무	학생	10.90 °C	2020-04-02 20
	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 샤워/목욕 온도는?	2020-04-26	남	16	따뜻함	17	7	17:07	무	학생	18.10 °C	2020-04-26 17
	2020-04-14	남	16	시원함	20	10	20:10	무	학생	13.80 °C	2020-04-14 20
<input type="radio"/> 시원함 <input type="radio"/> 미지근함 <input type="radio"/> 따뜻함 <input type="radio"/> 기타:	2020-04-24	남	16	따뜻함	19	20	19:20	무	학생	12.90 °C	2020-04-24 19
	2020-04-28	남	16	따뜻함	20	30	20:30	유	학생	14.30 °C	2020-04-28 20
귀하의 샤워/목욕 시작 시간대는?	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
내 답변	2020-04-17	남	17	시원함	23	10	23:10	무	학생	11.00 °C	2020-04-17 23
	2020-04-27	남	17	미지근함	7	15	7:15	무	학생	7.60 °C	2020-04-27 7
귀하의 샤워/목욕 하는데 걸리는 시간은?	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
	2020-04-17	남	17	따뜻함	23	15	23:15	무	학생	11.00 °C	2020-04-17 23

칼럼 이름	설명
날짜	설문을 한 날짜
성별	성별 (남 / 여)
나이	설문자 나이
선호하는 샤워 온도	샤워 시 선호하는 온도 (따뜻함, 미지근함, 시원함)
목욕 시작 시간	샤워를 시작하는 시간
샤워하는 시간	샤워를 하는 시간
샤워 종료 시간	샤워를 끝내는 시간
입욕제 사용 유무	샤워 시 입욕제를 사용하는지 유무 (유 / 무)
직업	설문자의 직업
날씨	각 일의 시간별 온도
날짜 별 샤워 온도	일의 시간별 온도와 샤워하는 시간을 조합한 컬럼

□ 데이터 전처리

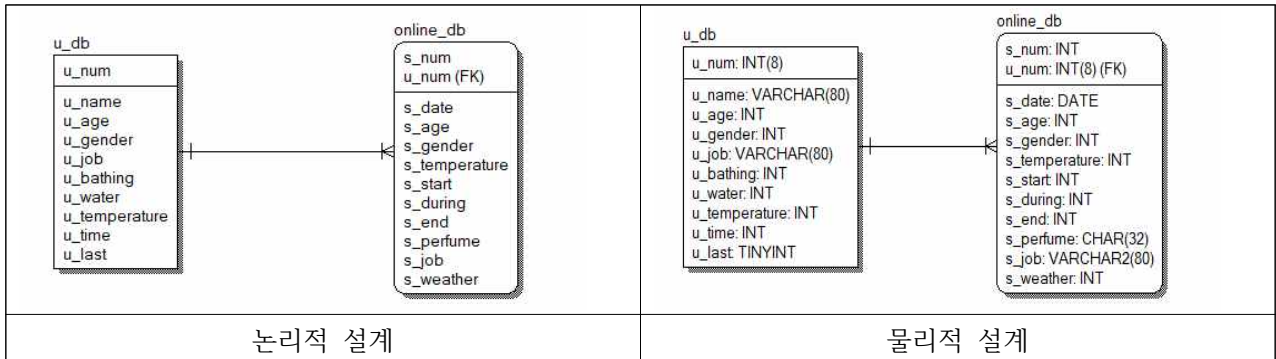
데이터 수집 후, 데이터 확인 결과 설문의 의도와 다른 방식으로 답한 답변들이 있어 데이터 전처리 과정을 통해 통일을 시키는 과정을 실시하였다.

	<p>- 설문 결과 나이를 ‘만 24세’, ‘24살’과 같이 다양한 답변이 있어 숫자만으로 통일</p>
	<p>- 설문 결과 온도를 정해진 틀이 아닌 ‘뜨뜻하게’와 같이 다양하게 설문이 되어 3가지 틀로 통일 (따뜻함, 미지근함, 시원함)</p>
	<p>- 설문 결과 시간을 AM, PM 형식 24시 기준 등 다양하게 작성되어 숫자로 통일</p>

3-3) I-Tub 데이터 베이스

데이터는 크게 회원에 대한 정보를 저장하는 테이블과 설문조사를 통해 얻은 조사값, 사용을 하며 얻어지는 데이터를 저장하는 테이블로 구성이 되어있다.

□ ERwin을 사용한 설계



□ online_db (여러 사용자들이 사용한 데이터를 저장하는 테이블)

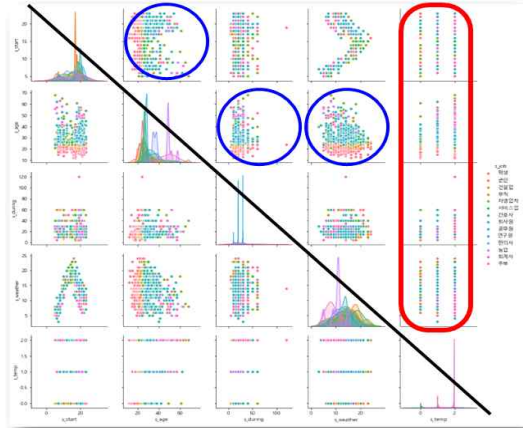
칼럼 이름	칼럼 타입	제약조건	비고
s_num	INT	NOT NULL AUTO_INCREMENT PRIMARY KEY	자동 증가 값
s_date	DATE	NOT NULL	날짜
s_age	INT	NOT NULL	사용자 나이
s_gender	INT	NOT NULL	사용자 성별
s_temperature	INT	NOT NULL	물 온도
s_start	INT	NOT NULL	샤워시작시간
s_during	INT	NOT NULL	샤워진행시간
s_end	INT	NOT NULL	샤워종료시간
s_perfum	Char(32)	NOT NULL	입욕제
s_job	VarChar(80)	NOT NULL	사용자 직업
s_weather	INT	NOT NULL	그날의 날씨

□ 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
u_bathing	INT	NOT NULL	사용자설정 입욕제
u_water	INT	NOT NULL	사용자설정 수위
u_temperature	INT	NOT NULL	사용자설정 온도
u_time	INT	NOT NULL	사용자설정 시간
u_last	TINYINT	NOT NULL	사용자 사용 확인

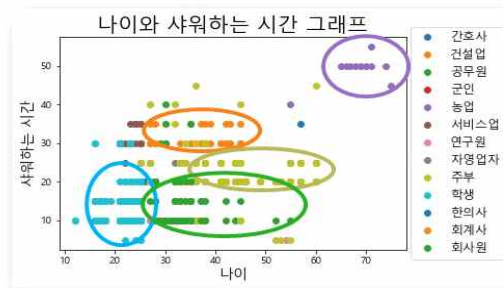
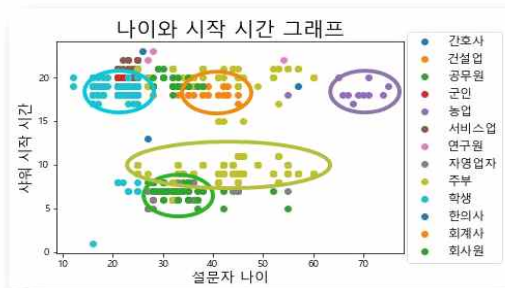
3-4) I-Tub 머신러닝 모델 선정

□ 모든 요소들간의 산점도



모델을 선정하기전 모든 요소들간의 산점도를 그려 보아 중요한 요소들을 선정하는 과정을 진행하였다. 그래프에서 파란 동그라미는 필요한 요소, 빨간 동그라미는 불필요한 요소로 하나씩 예로 들자면 샤워하는 시간과 샤워 온도는 무작위로 분포하여 서로 상관관계가 없었고 사용자의 나이와 샤워 시작 시간과 같은 경우에는 파란 동그라미처럼 군집화가 되며 상관관계가 있는 것을 확인 할 수 있었다. 그리고 필요한 요소중 가장 높은 상관관계를 지닌 2가지를 선택해 자세히 분석을 실시하였다.

□ 가장 높은 상관관계를 가진 2가지의 산점도



나이와 이름, 직업과 이름과 같은 요소는 상관관계가 있을 수가 없기때문에 배제하고 진행하였다. 나이와 시작 시간을 분포도 그래프로 그려본 결과 모든 사람이 그런 것은 아니지만 나이와 직업에 따라 샤워를 시작하는 시간이 군집화되어 있어 분류하는 것이 가능한 것으로 나타났다. 또한 나이와 직업에 따라 샤워하는 시간도 대부분이 따로따로 군집화되어 분류하는 것이 가능한 것으로 나타나 이 점을 이용하여 모델이 예측할 때 사용할 컬럼과 예측 가능한 컬럼을 결정할 수 있었다. 그 밖의 요소들은 서로 상관관계가 크게 나타나지 않아 예측 시 필요 요소에 넣기는 하였지만 큰 영향을 끼치지 않는았다.

	컬럼명
예측시 필요한 요소	나이, 직업, 성별, 날씨
예측 가능 요소	샤워하는 시간, 샤워를 시작할 시간, 샤워 온도

시각화 과정을 통해 요소 중 상관관계가 높은 요소를 찾아내었고 예측이 가능한 요소도 찾아내어 여러 머신러닝 알고리즘으로 테스트를 진행하여 정확도와 정밀도 같은 요소를 바탕으로 모델을 선정하게 되었다.

□ 머신러닝 모델 선정

정밀도와 재현율이 높을 수록 좋은 모델이며 해당 개념은 아래와 같다.

- 정확도(Accuracy) : 실제 답을 맞춘 예측의 비율을 나타내는 수치
- 정밀도(Precision) : 예측한 답 중 실제 답이 얼마나 포함되어 있는지를 나타내는 수치
- 재현율(Recall) : 실제 답 중 학습 모델이 얼마나 예측을 했는지를 나타내는 수치
- F1-score : 정밀도와 재현율의 조화평균
(정밀도가 재현율보다 압도적으로 높거나 그 반대의 경우의 조화를 이루는 평균을 구한 수치)

모델명	정확도 (Accuracy)	정밀도 (Precision)	재현율 (Recall)	F1-score
의사결정트리 (DecisionTree)	65% (train : 76%)	66%	65%	64%
랜덤포레스트 (RandomForest)	65% (train : 77%)	66%	65%	64%
XGBoost	66% (train : 76%)	66%	66%	66%
KNN (K-nearest-neighbors)	50% (train : 50%)	51%	50%	49%

모델간의 비교 결과 XGBoost 모델이 가장 좋은 수치를 보여주었지만 예측하는데 걸리는 시간이 길어 예측시간이 가장 빠르고 가장 좋은 수치를 보여주는 랜덤포레스트를 모델로 선정하게 되었다. 테스트 수치는 위의 표와 같다.

□ 위의 모델을 적용한 소스 코드 (predict.py)

```
#-*- coding:utf-8 -*-
import numpy as np
import pandas as pd
import time
import sys

# 랜덤포레스트 모델을 사용하기 위한 import 문
from sklearn.ensemble import RandomForestClassifier

# 학습과 테스트 데이터를 나누고 각종 평가 점수를 확인하기 위한 import 문
from sklearn.metrics import confusion_matrix, classification_report, roc_curve, auc
from sklearn.model_selection import train_test_split, cross_val_score

import requests, json

col_li = ['s_weather', 's_age', 's_gender', '직업_간호사', '직업_건설업', '직업_공무원', '직업_군인', '직업_농업', '직업_무직', '직업_서비스업', '직업_연구원', '직업_자영업자', '직업_주부', '직업_학생', '직업_한의사', '직업_회계사', '직업_회사원']

def machin (py_age , py_gender , py_job , predict_col ):
    idx = 0
    new_df = []

    # 예측 모델에 넣기위해 데이터를 편집하는 부분
    for i in range (0 , len (col_li)):
        if (col_li[i] == ('직업_' + py_job)):
            idx = i
    for i in range (0 , 17 ):
        if (i == idx):
            new_df.append(1 )
        elif (i == 0 ):
            new_df.append(20 )
        elif (i == 1 ):
            new_df.append(int (py_age))
        elif (i == 2 ):
            if (py_gender == '여성'):
                new_df.append(1 )
            else :
                new_df.append(0 )
        else :
            new_df.append(0 )

    # 정밀도, 재현율, f1 스코어, 서포트를 알려준다.
    x = pd.concat([shower_data['s_weather'], shower_data['s_age'], shower_data['s_gender'],
                  pd.get_dummies(shower_data['s_job'], prefix='직업')], axis =1 )
    y = shower_data[predict_col]
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size =0.3 , random_state =44 )
    model = RandomForestClassifier()
    model.fit(x_train, y_train)
    tmp = pd.DataFrame(data =new_df).T
```

```

tmp.columns = col_li

# 웹 서버로 값을 넘기기위해 편집하는 부분.
if (predict_col == 's_temp'):
    result = str (model.predict(tmp))[2 :-2 ]
else :
    result = str (model.predict(tmp))[1 :-1 ]
new_df = []
return result

shower_data = pd.read_csv('./public/survey_data.csv', encoding='utf-8')
shower_data.head()

# 남, 여 각각 0과 1로 변경
shower_data.loc[shower_data['s_gender']=='남', 's_gender'] = 0
shower_data.loc[shower_data['s_gender']=='여', 's_gender'] = 1
#shower_data.head()

# 입욕제 유무를 숫자로 변경
shower_data.loc[shower_data['s_perfume']=='무', 's_perfume'] = 0
shower_data.loc[shower_data['s_perfume']=='유', 's_perfume'] = 1
#shower_data.head()

shower_data['s_gender'] = pd.to_numeric(shower_data['s_gender'])
shower_data['s_perfume'] = pd.to_numeric(shower_data['s_perfume'])
shower_data.drop(['s_end'], axis=1, inplace=True)
shower_data.head()

# 웹 서버에서 넘어온 각종 정보를 수정하여 머신러닝 함수에 알맞게 넣는 코드
start = machin(sys.argv[1], sys.argv[2], sys.argv[3], 's_start')
during = machin(sys.argv[1], sys.argv[2], sys.argv[3], 's_during')
temp = machin(sys.argv[1], sys.argv[2], sys.argv[3], 's_temp')
print (str (start+'/' +during+'/' +temp))

```

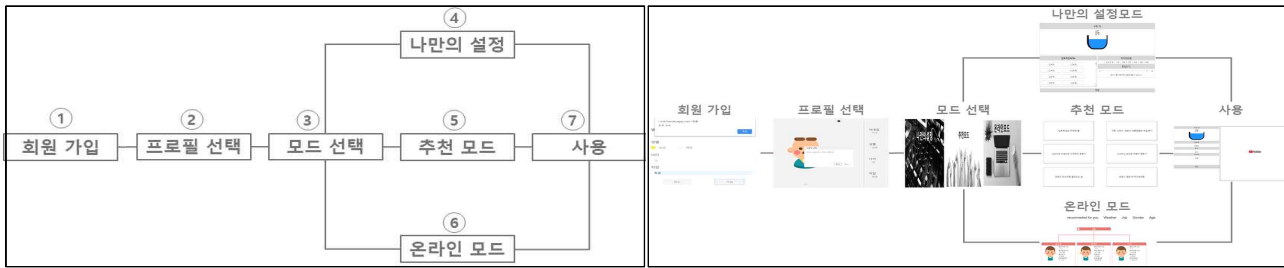
□ 위의 모델을 적용한 예측 결과

예측 1	예측 2
<div> <div> <div>사용자 정보</div>  <div> <div>이름</div>대린<div>나이</div>14 <div>성별</div>여성<div>직업</div>유튜버 </div> </div> <div> <div>예측 시작 시간</div>5 <div>예측 샤워 시간</div>30 <div>예측 샤워 온도</div>따뜻함 </div> </div>	<div> <div> <div>사용자 정보</div>  <div> <div>이름</div>지수<div>나이</div>25 <div>성별</div>남성<div>직업</div>학생 </div> </div> <div> <div>예측 시작 시간</div>12 <div>예측 샤워 시간</div>20 <div>예측 샤워 온도</div>따뜻함 </div> </div>

선정한 모델로 실제 데이터에 대입하여 예측을 실시해본 모습이다. 빨간 박스로 체크되어 있는 부분을 보면 사용자의 성별, 나이, 직업과 해당 날짜의 날씨를 사용하여 예측이 된 모습을 확인할 수 있었다. 선정한 모델에 따르면 나이가 14세이고 여성이며 유튜버인 사람은 목욕을 5시에 시작하고 30분간 하며 따뜻한 온도로 한다고 예측이 되었으며 나이가 25세이며 남성이고 학생일 경우 12시에 목욕을 하고 20분간 목욕을 하고 따뜻한 온도로 목욕을 할 것 이라고 예측이 되었다.

3-5) I-Tub 웹 페이지

I-Tub 웹페이지는 총 7가지의 페이지로 구성되어 있으며 역할과 구성도는 다음과 같다.



□ 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
online_mode.html	사용자 정보를 사용하여 사용값을 예측해서 서비스하고 다른 사용자의 사용 정보를 통계화하여 보기좋게 시각화하여 보여주는 페이지
item_info.html	사용자가 설정한 사용값을 보여주고 미디어를 시청할 수 있는 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
item_info_style.css	사용값 페이지 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
orgchart.js	ORG 차트를 사용하기 위한 자바 스크립트


□ DATA

파일명	설명
survey_data.csv	설문조사로 수집한 데이터이며 이 데이터를 바탕으로 기계학습을 하며 사용자의 사용값들이 이 데이터에 업데이트된다.

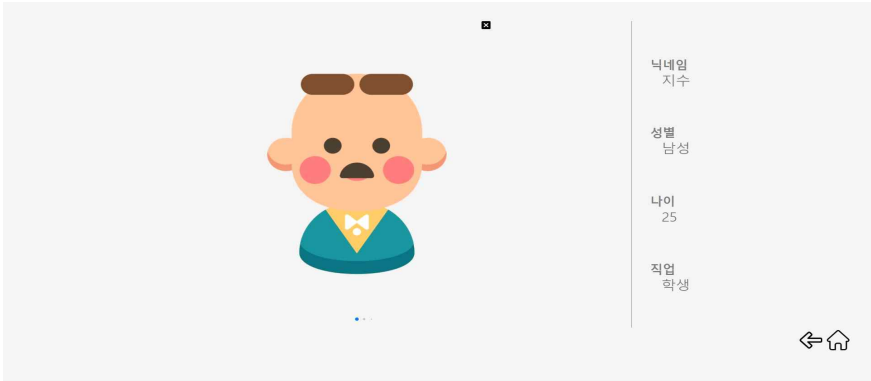
□ 웹 사이트 구현 부분

(주요 기능 코드만 적어놓았음, 전체 코드는 https://github.com/Ultimate-ItubProject/Itub_Web 참고)

메인 페이지



프로필 선택 페이지



```
<script src =\"./js/jquery-3.5.1.js\"></script> //제이쿼리 사용
<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 ,
        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 .getElementById ('cookie_name').value =
            document .getElementById ('label_name').innerText ;
            document .getElementById ('cookie_age').value =
            document .getElementById ('label_age').innerText ;
            document .getElementById ('cookie_gender').value =
            document .getElementById ('label_gender').innerText ;
            document .getElementById ('cookie_job').value =
            document .getElementById ('label_job').innerText ;
            document .getElementById ('cookie_submit').click ();
        }
    },
    {
        //버튼텍스트
        text: "취소",
        //클릭이벤트발생시 동작
        click : function () {
            $(this ).dialog ("close");
        }
    }
]
});
});
}
else document .getElementById ('register_submit').click ();
},
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 () {
    else {
        $(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 ); //사용자 삭제
                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 );
    if (tmp .length == 0 ) {
        swiper .appendSlide ('<div class="swiper-slide"><div class="vertical">
            </div></div>');

        swiper .update ();
        $(' .btn' ).hide ();
    }
    else {
        for (var i = 0 ; i < tmp .length ; i ++ ) {
            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'
            }
            var htmlCode = '<div class="swiper-slide"><div class="vertical">
                <img src=' + path + ' class="imgtag"></div></div>';
            swiper .appendSlide (htmlCode );
            swiper .update ();
        }
        swiper .appendSlide ('<div class="swiper-slide"><div class="vertical">
            </div></div>');
    }
}

```



```

        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 ;
    }
}
function sendAjax (url , data ){
    var inputdata = {'temp':'0'};
    data = JSON.stringify(inputdata);
    var xhr = new XMLHttpRequest();
    xhr.open('POST', url);
    xhr.setRequestHeader('content-type', 'application/json');
    xhr.send(data);
    xhr.addEventListener('load', function () {
        console.log(xhr.responseText);
    });
};
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')
    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.');
```

모드 선택 페이지



나만의 설정 페이지

```
<script src ="/js/jquery-3.5.1.js"></script>
<script src ="/js/orgchart.js"></script>
<script src = "https://cdnjs.cloudflare.com/ajax/libs/jquery-csv/0.71/jquery.csv-0.71.min.js"></script>
<script>
    var water = 70 ;
    var bathing = '없음';
    var temperature = 36 ;
    var time = '설정 안함';
    //사용자 사용 정보(수위, 입욕제, 온도, 시간)에 대한 데이터 쿠키 저장
    saveBtn = document .querySelector ( '#save-btn' )
    saveBtn .addEventListener ( 'click', function ( e ) {
        var form = document .createElement ( 'form' );
        form .setAttribute ( 'method', 'POST' );
        form .setAttribute ( 'action', '/item_cookie' );

        var waterFiled = document .createElement ( 'input' );
        waterFiled .setAttribute ( 'type', 'hidden' );
        waterFiled .setAttribute ( 'name', 'water' );
        waterFiled .setAttribute ( 'value', water ); // 수위 쿠키 저장
        form .appendChild ( waterFiled );
        var bathingFiled = document .createElement ( 'input' );
        bathingFiled .setAttribute ( 'type', 'hidden' );
        bathingFiled .setAttribute ( 'name', 'bathing' );
        bathingFiled .setAttribute ( 'value', bathing ); //입욕제 쿠키 저장
        form .appendChild ( bathingFiled );
        var temperaturtrueFiled = document .createElement ( 'input' );
        temperaturtrueFiled .setAttribute ( 'type', 'hidden' );
        temperaturtrueFiled .setAttribute ( 'name', 'temperature' );
        temperaturtrueFiled .setAttribute ( 'value', temperature ); //온도 쿠키 저장
        form .appendChild ( temperaturtrueFiled );
        var timeFiled = document .createElement ( 'input' );
        timeFiled .setAttribute ( 'type', 'hidden' );
        timeFiled .setAttribute ( 'name', 'time' );
        timeFiled .setAttribute ( 'value', time ); //시간 쿠키 저장
        form .appendChild ( timeFiled );
        document .body .appendChild ( form );
        form .submit ();
    })
    // 입욕제 항목들 생성 부분.
    var parent = document .querySelector ( ".list" );
    for ( var i = 0 ; i < 9 ; i ++ ) {
        var bathbomb = new Array ( '없음', '귤', '바닐라', '라벤더', '레몬', '자스민',
                                    '로즈제라늄', '로즈마리', '블루마린' );

        var div = document .createElement ( 'div' );
        div .className = 'bathing item'
```

```

var label = document.createElement('label');
var input = document.createElement("input");
input.type = 'radio';
input.name = 'bathing_agent';
input.value = bathbomb[i];
if (i == 0) input.checked = 'checked';
label.appendChild(input);
var span = document.createElement("span");
span.className = 'bathing_item_text';
span.appendChild(document.createTextNode(bathbomb[i]));
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;
        bathing = 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;
        //
        time = e.target.value;
        console.log(e.target.value);
        if (text !== '설정 안함') {
            text += "분"
        }
        document.querySelector("#timer_text").innerHTML = ' ' + text;
    })
}

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

var waterrange = document.querySelector(".water-slider");
var watervalue = document.querySelector("#water-title");
waterrange.addEventListener("input", (e) => {
    watervalue.innerHTML = e.target.value;
    //
    water = e.target.value
}, false)

var getCookie = function (name) { // 키 값에 따라 쿠키 가져오기
    var value = document.cookie.match ('(^|;)?' + name + '=(.*);(;$)');
    return value ? value [2] : null;
};

// 수평 바 움직일경우 수직 바도 같이 움직임
var h_to_v = function () {
    var h = $('#water-slider-alf1');
    h.on('input',function () {
        var v = $('#water-slider');
        v[0].value = h[0].value;
        watervalue.innerHTML =v[0].value;
    })
}

// 수직 바 움직일 경우 수평 바도 같이 움직임
var v_to_h = function () {
    var v = $('#water-slider');
    v.on('input',function () {
        var h = $('#water-slider-alf1');
        h[0].value = v[0].value;
    })
}

// 밑에 두개가 있어야 드래그 하면서 실시간으로 값이 바뀜
h_to_v ();
v_to_h ();
</script>

```

추천 모드 페이지



```
<script>
//title이름, 수위, 사용시간, 온도, 입욕제종류
var select_mode = [
  ['입욕제 없는 따뜻한 물', [80, 5, 43, '없음'], './images/hotwater.png'],
  ['자몽, 오렌지, 레몬의 새콤달콤한 과일 향기', [80, 30, 38, '귤'], './images/orange.png'],
  ['크림미한 바닐라와 이국적인 꽃향기', [80, 30, 38, '바닐라'], './images/exotic.png'],
  ['신선하고 포근한 라벤더 꽃향기', [80, 20, 38, '라벤더'], './images/lavender.png'],
  ['감정이 파도처럼 밀려오는 날', [80, 20, 38, '레몬'], './images/wave.png'],
  ['로맨스 영화 속 주인공처럼', [80, 30, 38, '자스민'], './images/romance.png'],
  ['새벽녘, 그 감성속으로', [80, 30, 38, '로즈제라늄'], './images/sunset.png'],
  ['감성이 필요한 당신을 위한 선물', [80, 30, 30, '로즈마리'], './images/present.png'],
  ['꾸리꾸리 비오는 날엔', [80, 25, 38, '블루마린'], './images/rainnyday.png'],
  ['생각이 많은 요즘', [80, 30, 36, '허브'], './images/herb.png']
]
var parent = document.querySelector(".list"); // 미리 생성된 각 버튼 생성
for (let i = 0; i < select_mode.length; i++) {
  var button = document.createElement('button');
  button.className = 'btn';
  button.onclick = function () {
    var form = document.createElement('form');
    form.setAttribute('method', 'POST');
    form.setAttribute('action', '/item_cookie');
    var a_Field = document.createElement('input'); // 수위 쿠키 저장
    a_Field.setAttribute('type', 'hidden');
    a_Field.setAttribute('name', 'water');
    a_Field.setAttribute('value', select_mode[i][1][0]);
    form.appendChild(a_Field);
    var b_Field = document.createElement('input'); // 온도 쿠키 저장
    b_Field.setAttribute('type', 'hidden');
    b_Field.setAttribute('name', 'temperature');
    b_Field.setAttribute('value', select_mode[i][1][1]);
    form.appendChild(b_Field);
    var c_Field = document.createElement('input'); // 시간 쿠키 저장
    c_Field.setAttribute('type', 'hidden');
    c_Field.setAttribute('name', 'time');
    c_Field.setAttribute('value', select_mode[i][1][2]);
    form.appendChild(c_Field);
    var d_Field = document.createElement('input'); // 입욕제 쿠키 저장
    d_Field.setAttribute('type', 'hidden');
    d_Field.setAttribute('name', 'bathing');
    d_Field.setAttribute('value', select_mode[i][1][3]);
    form.appendChild(d_Field);
    document.body.appendChild(form);
    form.submit();
  }
  var span = document.createElement('span');
  var p = document.createElement('p');
  p.appendChild(document.createTextNode(select_mode[i][0]));
  span.appendChild(p);
  button.appendChild(span);
  button.style.backgroundRepeat = "no-repeat";
  button.style.backgroundImage = "url(" + select_mode[i][2] + ")";
  button.style.backgroundSize = "cover";
  parent.appendChild(button); //각 버튼에 대한 CSS 설정
}
</script>
```

온라인 모드 페이지

recommended for you Weather Job Gender Age

사용자 정보

예측 시작 시간: 12

예측 지점 시간: 20

예측 지점 온도: 파란강

이름: 지수 나이: 25

성별: 남성 직업: 학생

```
<script type="text/javascript" src="./js/jquery-3.5.1.js"></script>
<script type="text/javascript" src="./js/orgchart.js"></script>
<script type="text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/jquery-csv/0.71/jquery.csv-0.71.min.js"></script>
<script>
    var bathbomb = new Array ('사용안함', '귤', '바닐라', '라벤더', '레몬', '자스민',
                                '로즈제라늄', '로즈마리', '블루마린', '일랑일랑');

    var getCookie = function (name) {
        var value = document.cookie.match ('(^|;) ?' + name + '=([^;]*)($|)');
        return value ? value [2] : null ;
    }; //쿠키값 가져오는 함수

    var age = decodeURI (getCookie ('cookie_age'));
    var gender = decodeURI (getCookie ('cookie_gender'));
    document.querySelector('#name-content').innerHTML = decodeURI (getCookie ('cookie_name'));
    document.querySelector('#age-content').innerHTML = age ;
    document.querySelector('#gender-content').innerHTML = gender ;
    document.querySelector('#job-content').innerHTML = decodeURI (getCookie ('cookie_job'));
    document.querySelector('#predict-start-content').innerHTML=
    decodeURI(getCookie ('cookie_predict_start'));
    document.querySelector ('#predict-during-content').innerHTML =
    decodeURI (getCookie ('cookie_predict_during'));
    document.querySelector ('#predict-temp-content').innerHTML =
    decodeURI (getCookie ('cookie_predict_temp'));
    var img = document.querySelector ('#user-img'); // 사용자 나이에 따른 사진 변경
    if (gender == '남성') { // 남성일 경우 나이에 맞게 그림 변경
        if (age <19 ) img .style .backgroundImage = "url('./images/boy.png')";
        else if (age <65 ) img .style .backgroundImage = "url('./images/fa.png')";
        else img .style .backgroundImage = "url('./images/granfa.png')";
    }
    else { // 여성일 경우 나이에 맞게 그림 변경
        if (age <19 ) img .style .backgroundImage = "url('./images/girl.png')";
        else if (age <65 ) img .style .backgroundImage = "url('./images/ma.png')";
        else img .style .backgroundImage = "url('./images/granma.png')";
    }

    img .addEventListener ('click',function () { // 그림 클릭시 선택한 값 쿠키 저장
        var form = document .createElement ('form');
        form .setAttribute ('method', 'POST');
        form .setAttribute ('action', '/item_cookie' );

        var water ;
        if (age <16 ) water = Math .floor (Math .random ()*10 +50 );
        else water = Math .floor (Math .random ()*20 +70 );
        var water_filed = document .createElement ('input') //수위 쿠키 저장
        water_filed .setAttribute ('type', 'hidden');
        water_filed .setAttribute ('name', 'water');
        water_filed .setAttribute ('value', water );
        form .appendChild (water_filed );
        var bathing = bathbomb [Math .floor (Math .random (9 ))];
        var bathing_filed = document .createElement ('input'); //입욕제 쿠키 저장
        bathing_filed .setAttribute ('type', 'hidden');
        bathing_filed .setAttribute ('name', 'bathing');
        bathing_filed .setAttribute ('value', bathing );
        form .appendChild (bathing_filed );
        var temp = decodeURI (getCookie ('cookie_predict_temp'));
        var temp_filed = document .createElement ('input'); // 온도 쿠키 저장
        temp_filed .setAttribute ('type', 'hidden');
```

```

temp_filed.setAttribute('name', 'temperature');
temp_filed.setAttribute('value', temp);
form.appendChild(temp_filed);
var time = decodeURI(getCookie('cookie_predict_start')); // 시간 쿠키 저장
var time_filed = document.createElement('input');
time_filed.setAttribute('type', 'hidden');
time_filed.setAttribute('name', 'time');
time_filed.setAttribute('value', time);
form.appendChild(time_filed);
document.body.appendChild(form);
form.submit();
})
</script>
<script type="text/javascript">
$(function () {
    // 각 타이틀
    var title = [
        ['청소년', '중장년', '노년'],
        ['남자', '여자'],
        [],
        ['5~10도', '10~15도', '15~20도', '20~25도', '25~30도', '30~35도', '35~40도']
    ];
    var bathbomb = new Array('사용안함', '쿨', '바닐라', '라벤더', '레몬', '자스민',
        '로즈제라늄', '로즈마리', '블루마린', '일랑일랑');

    function csvreading (idx) { // 사용자 통계를 내기위해 csv파일 불러오기
        $.get("survey_data.csv", function (csvString) { // 페이지 로딩 이후 실행됨
            var csvdata = $.csv.toArrays(csvString, {
                onParseValue: $.csv.hooks.castToScalar
            }); // csv 파일을 array로 저장
            var container = document.getElementById("chart-container");
            var rfy = document.getElementById("rfy");
            container.style.display = "block";
            rfy.style.display = "none";
            var arraylength = csvdata.length; // 저장된 csv 파일의 데이터 길이
            for (var i = 1; i < arraylength; i++) {
                title[2].push(csvdata[i][8]);
            }
            // 중복된 직업이름 제거
            title[2] = title[2].filter((item, index) => title[2].indexOf(item) == index);
            var str_size = title[idx].length; // 타이틀 크기
            // 각 타이틀에 들어갈 내용 저장할 배열
            // 2차원 배열(i는 타이틀, j는 텍스트 출력값)
            // 0 시작시간 : 24시 기준
            // 1 걸린시간 : 60분 기준
            // 2 온도 : 1 시원함 2 미지근함 3 따뜻함
            // 3 입욕제여부 : 2 사용 1 미사용
            // 4 타이틀 크기 : 각 데이터 크기
            var text_data = new Array(str_size);
            for (var i = 0; i < str_size; i++) {
                text_data[i] = new Array(5);
                for (var j = 0; j < 5; j++)
                    text_data[i][j] = 0;
            }
            // 데이터 전부 돌면서 전체 데이터 계산
            for (var i = 1; i < arraylength; i++) {
                var title_idx = 0;
                if (idx == 0) { // 나이에 따른 통계 데이터 구하기
                    if (csvdata[i][2] < 20) title_idx = 0;
                    else if (csvdata[i][2] < 65) title_idx = 1;
                    else title_idx = 2;
                    text_data[title_idx][4] += 1;
                    text_data[title_idx][0] += csvdata[i][4];
                    text_data[title_idx][1] += csvdata[i][5];
                    if (csvdata[i][3] == "cold") text_data[title_idx][2] += 1;
                    else if (csvdata[i][3] == "nomal") text_data[title_idx][2] += 2;
                    else if (csvdata[i][3] == "hot") text_data[title_idx][2] += 3;
                    if (csvdata[i][7] == "유") text_data[title_idx][3] += 2;
                    else if (csvdata[i][7] == "무") text_data[title_idx][3] += 1;
                } else if (idx == 1) { // 성별에 따른 통계 데이터 구하기
                    if (csvdata[i][1] == "남") title_idx = 0;
                    else if (csvdata[i][1] == "여") title_idx = 1;
                    else console.log(i);
                    text_data[title_idx][4] += 1;
                    text_data[title_idx][0] += csvdata[i][4];
                    text_data[title_idx][1] += csvdata[i][5];

```

```

        if (csvdata[i][3] == "cold") text_data [title_idx][2] += 1 ;
        else if (csvdata[i][3] == "nomal") text_data [title_idx][2] += 2 ;
        else if (csvdata[i][3] == "hot") text_data [title_idx][2] += 3 ;
        if (csvdata[i][7] == "유") text_data [title_idx][3] += 2 ;
        else if (csvdata[i][7] == "무") text_data [title_idx][3] += 1 ;
    } else if (idx == 2 ) { //직업에 따른 통계 데이터 구하기
        for (var j = 0 ; j < str_size ; j++) {
            if (csvdata[i][8] == title [2][j]) {
                title_idx = j ;
                break ;
            }
        }
        text_data [title_idx][4] += 1 ;
        text_data [title_idx][0] += csvdata[i][4];
        text_data [title_idx][1] += csvdata[i][5];
        if (csvdata[i][3] == "cold") text_data [title_idx][2] += 1 ;
        else if (csvdata[i][3] == "nomal") text_data [title_idx][2] += 2 ;
        else if (csvdata[i][3] == "hot") text_data [title_idx][2] += 3 ;
        if (csvdata[i][7] == "유") text_data [title_idx][3] += 2 ;
        else if (csvdata[i][7] == "무") text_data [title_idx][3] += 1 ;
    } else if (idx == 3 ) { // 날씨에 따른 통계 데이터 구하기
        var weather_data = csvdata[i][9];
        if (weather_data <= 10.00 ) title_idx = 0 ;
        else if (weather_data <= 15.00 ) title_idx = 1 ;
        else if (weather_data <= 20.00 ) title_idx = 2 ;
        else if (weather_data <= 25.00 ) title_idx = 3 ;
        else if (weather_data <= 30.00 ) title_idx = 4 ;
        else if (weather_data <= 35.00 ) title_idx = 5 ;
        else if (weather_data <= 40.00 ) title_idx = 6 ;
        text_data [title_idx][4] += 1 ;
        text_data [title_idx][0] += csvdata[i][4];
        text_data [title_idx][1] += csvdata[i][5];
        if (csvdata[i][3] == "cold") text_data [title_idx][2] += 1 ;
        else if (csvdata[i][3] == "nomal") text_data [title_idx][2] += 2 ;
        else if (csvdata[i][3] == "hot") text_data [title_idx][2] += 3 ;
        if (csvdata[i][7] == "유") text_data [title_idx][3] += 2 ;
        else if (csvdata[i][7] == "무") text_data [title_idx][3] += 1 ;
    }
}
// org차트 수정
var $this = $(".node");
$('.node').data('node', $this);
// 선택된 메뉴의 각 타이틀 만큼 반복
var test ;
for (var i = 0 ; i < str_size ; i++) {
    if (Math.floor(i / 7) == (i / 7)) test = (i / 7);
    else test += (1 + Math.floor(i / 7));
    var nodeVals = [];
    var real_idx = i < 2 ? 1 - i : i ; // 제일 앞에 있는 자식 노드가 1->0이기 때문에
    if (i >= 7) $('.node').data('node', $('.node')[test]);
    else $('.node').data('node', $this);
    nodeVals.push(title [idx][real_idx]);
    var avg_text = "";
    if (text_data [real_idx][2] == 0 ) avg_text = '데이터없음';
    else {
        text_data [real_idx][2] = (text_data [real_idx][2] /
            text_data [real_idx][4]).toFixed(2);
        if (text_data [real_idx][2] < 1.50 ) avg_text = '시원함';
        else if (text_data [real_idx][2] < 2.50 ) avg_text = '미지근함';
        else if (text_data [real_idx][2] >= 2.50 ) avg_text = '따뜻함';
    }
    var imagepic = ""; // 이미지 출력 클래스
    if (i * 10 <= 20 ) imagepic = 'boypic';
    else if (i * 10 <= 60 ) imagepic = 'fapic';
    else imagepic = 'granfapic';
    var code = '<table class="t" id="t">\ // 생성된 노드에 그림을 넣기 위해 작성
    <colgroup>\
        <col style="width:50%">\
        <col style="width:50%">\
    </colgroup>\
    <tr>\
        <td rowspan="10" class="image ' + imagepic + '">\
        </td>\

```



```

        </tr>\
        <tr>\
            <td><b>평균시작시간</b></td>\
        </tr>\
        <tr>\
            <td class="lb">' + (text_data [real_idx][0] == 0 ? '데이터없음' :
                (text_data [real_idx][0] / text_data [real_idx][4]).toFixed (0))
                + (text_data [real_idx][0] == 0 ? ' ' : '시') + '</td>\
        </tr>\
        <tr>\
            <td><b>평균결련시간</td>\
        </tr>\
        <tr>\
            <td class="lb">' + (text_data [real_idx][1] == 0 ? '데이터없음' : (text_data
                [real_idx][1] / text_data [real_idx][4]).toFixed (2))
                + (text_data [real_idx][1] == 0 ? ' ' : '분') + '</td>\
        </tr>\
        <tr>\
            <td><b>평균온도</td>\
        </tr>\
        <tr>\
            <td class="lb">' + avg_text + '</td>\
        </tr>\
        <tr>\
            <td><b>입욕제여부</td>\
        </tr>\
        <tr>\
            <td class="lb">' + (text_data [real_idx][3] == 0 ? '데이터없음' :
                ((text_data [real_idx][3] / text_data [real_idx][4]) == 1
                ? bathbomb [0] : bathbomb [Math .floor (
                    Math .random () * 9 + 1 )])) + '</td>\
        </tr>\
    </table>';
    var $node = $('<div>').data ('node');
    var hasChild = $node .parent ().attr ('colspan') > 0 ? true : false ;
    if (!hasChild ) { // org 차트에서 자식노드가 없을 경우 자식으로 생성
        var rel = nodeVals .length > 1 ? '110' : '100';
        oc .addChildren ($node , nodeVals .map (function (item )
            {
                return {
                    'name': item ,
                    'relationship': '110',
                    'id': getId (),
                    'title': code
                };
            }
        ));
    } else { //org차트에서 자식 노드가 있을 경우 자식 노드와 같은 라인에 생성
        oc .addSiblings ($node .closest ('tr').siblings ('.nodes').find
            ('.node:first'), nodeVals .map (function (item )
            {
                return {
                    'name': item ,
                    'relationship': '110',
                    'id': getId (),
                    'title': code
                };
            }
        ));
    }
    }
    })
    });
    var getId = function () { //node ID 구하기
        return (new Date ().getTime ()) * 1000 + Math .floor (Math .random () * 1001 );
    };
    // 초기화면 데이터
    var datasource = {
        'name': 'Age',
        'title': ''
    };
    var oc = $('#chart-container').orgchart ({
        'data': datasource ,
        'nodeContent': 'title',
        'chartClass': 'edit-state',
        'exportFilename': 'SportsChart',
        'createNode': function ($node , data ) {

```



```

        var secondMenuIcon = $('<i>', {
            'class': 'oci oci-info-circle second-menu-icon',
            click : function () {
                $(this).siblings('.second-menu').toggle ();
            }
        });
        var secondMenu = '<div class="second-menu">
            </div>';
        $node.append (secondMenuIcon).append (secondMenu);
    }
});
//각 버튼 클릭시 org 차트 초기화
var box = document.getElementsByClassName ("content");
box [0].style.height = 0 ;
$('#btn-chart1').on ('click', function (argument ) { //초기화
    oc .init ({
        'data': datasource
    });
    var box = document.getElementsByClassName ("content");
    box [0].style.height = 0 ;
    csvreading (0 );
});
$('#btn-chart2').on ('click', function (argument ) { // 성별에 따라 통계 보여줌
    var data = {
        'name': 'Gender'
    };
    oc .init ({
        'data': data
    });
    var box = document.getElementsByClassName ("content");
    box [0].style.height = 0 ;
    csvreading (1 );
});
$('#btn-chart3').on ('click', function (argument ) { // 직업에 따라 통계 보여줌
    var data = {
        'name': 'Job'
    };
    oc .init ({
        'data': data
    });
    var box = document.getElementsByClassName ("content");
    box [0].style.height = 0 ;
    csvreading (2 );
});
$('#btn-chart4').on ('click', function (argument ) { // 날씨에 따라 통계 보여줌
    var data = {
        'name': 'Weather'
    };
    oc .init ({
        'data': data
    });
    var box = document.getElementsByClassName ("content");
    box [0].style.height = 0 ;
    csvreading (3 );
});
$('#btn-chart5').on ('click', function (argument ) { //사용자에게 추천 해주는 창
    var container = document.getElementById ("chart-container");
    var rfy = document.getElementById ("rfy");
    container.style.display = "none";
    rfy.style.display = "block";
});
});
</script>

```

사용 페이지



```
<script>
var slider = document.querySelector ('.water-slider');
slider.disabled = true ;
var getCookie = function (name ) {
    var value = document.cookie.match ('(^|;) ?' + name + '=(.*)?');
    return value ? value [2] : null ;
}; // 쿠키 값 가져오기
// 최종 실행 창에서 현재 자신의 설정 값 출력을 위해 쿠키로부터 데이터 불러온 후 화면에 출력
var water = decodeURI (getCookie ('cookie_water')); //수위
var bathing = decodeURI (getCookie ('cookie_bathing')); //입욕제
var temperature = decodeURI (getCookie ('cookie_temperature')); //온도
var time = decodeURI (getCookie ('cookie_time')); //사용시간
var waterSlider = document.querySelector ('.water-slider');
waterSlider.value = water ;
var waterFiled = document.querySelector ('#water-value');
waterFiled.innerHTML = water ;
var bathingFiled = document.querySelector ('#bathing-value');
bathingFiled.innerHTML = bathing ;
var temperatureFiled = document.querySelector ('#temperature-value');
temperatureFiled.innerHTML = temperature + '<span>&degC</span>';
var timeFiled = document.querySelector ('#time-value');
if (time != '설정 안함') {
    time += '분'
}
timeFiled.innerHTML = time ;
</script>
```

3-6) I-Tub 시제품 제작

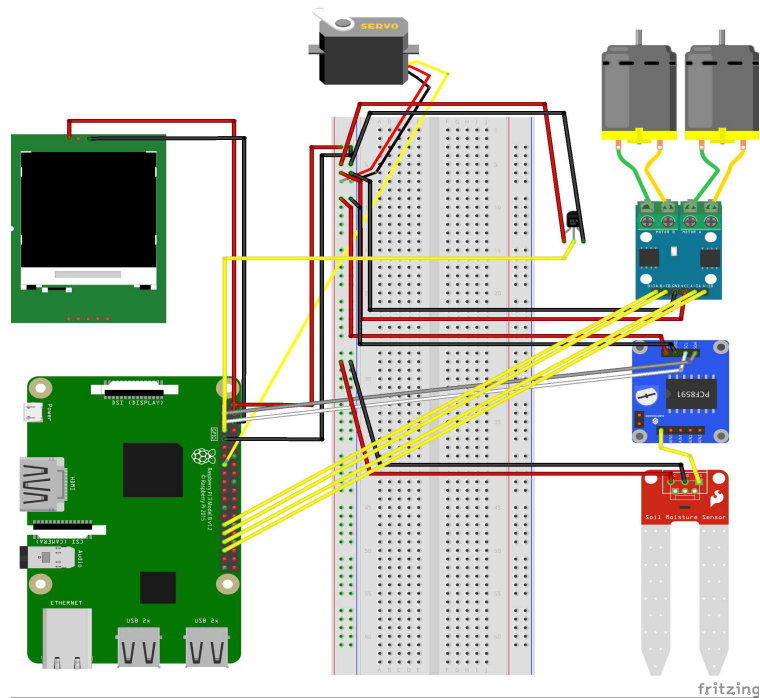
초기 기획 단계에서는 3D 프린터를 사용해 시제품을 제작하기로 기획되어 있었지만 직접 아크릴과 같은 재료를 사용해 만드는 것이 퀄리티 면이나 제작의 용이성에 있어 이점이 있다고 판단되어 직접 제작하게 되었다. 부품 구매에 관해서는 조선대학교 링크사업단 지원금으로 구매를 하여 제작하였으며 구매한 부품의 내역은 아래와 같다.

□ 제작 비용

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 지원으로 구매하였으며 자세한 사항은 <https://lincplus.chosun.ac.kr/main/> 참고)

□ I-Tub 시스템 회로도



라즈베리파이 시제품에 대한 회로도의 사진이다. 워터펌프 2개를 욕조 모형에 물을 담기위해 모터드라이버(L9110)을 사용하여 연결을 하였고 물의 수위를 측정하기위해 토양습도센서를 연결하였는데 라즈베리파이가 받지 못하는 아날로그 신호값으로 보내주기때문에 ADC(PCF8591)를 연결하여 사용한다. 서보모터는 입욕제를 투하시켜주기 위한 부품이다. 그리고 물의 온도를 측정하기 위해 방수형 온도센서(DS18B20)를 사용한다. 이 모든 것을 연결하여 I-Tub 서비스를 하는데 좌측 상단에 있는 터치스크린을 사용하여 이용할수 있게 된다. 웹 페이지에서 신호를 주면 라즈베리파이 센서들이 각 값에 맞게 작동을 시작하고 세팅이 완료되었다면 멈추게 된다.

□ I-Tub 시스템 센서 작동 코드 (I-tub_raspberryPi_code.py)

```
import RPi.GPIO as GPIO
import time
import spidev
import smbus
import os
import glob
import requests, json
import pymysql
import time

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)

# DB 연결을 위한 정보를 저장
db = pymysql.connect (
    user = 'bf1138ba34c820',
    passwd = '22ac05b88712768',
    host = 'us-cdbr-iron-east-01.cleardb.net',
    db = 'heroku_6295f565c172990',
    charset = 'utf8'
)

# 수위 센서를 사용하기 위한 설정
bus = smbus.SMBus(1)
```

```

add = 0x 48
idx = 0
switch = 17
GPIO.setup(switch, GPIO.IN)
os.system('modprobe w1-gpio')
os.system('modprobe w1-therm')

#방수형 온도센서를 사용하기 위한 설정
base_dir = '/sys/bus/w1/devices/'
device_folder = glob.glob(base_dir + '28*')[0]
device_file = device_folder + '/w1_slave'

# 워터펌프 GPIO 핀 설정
IN1 = 19   #37 pin
IN2 = 13   #35 pin
IN3 = 6    #31 pin
IN4 = 5    #29 pin

GPIO.setup(IN1, GPIO.OUT)
GPIO.setup(IN2, GPIO.OUT)
GPIO.setup(IN3, GPIO.OUT)
GPIO.setup(IN4, GPIO.OUT)

# 서보모터 GPIO 핀 설정
servo = 22

GPIO.setup(servo, GPIO.OUT)
p = GPIO.PWM(servo, 50 )
p.start(0 )

# 서보모터 각도 조절 함수
def setServoPos (degree ):
    if (degree > 180 ):
        degree = 180
    duty = 3 + (degree*(12 -3 )/180.0 )
    p.ChangeDutyCycle(duty)

# 워터펌프 제어 함수
def setMotorContorl (INA , INB , stat ):
    #앞으로
    if stat == 'FORWARD':
        GPIO.output(INA, GPIO.HIGH)
        GPIO.output(INB, GPIO.LOW)
    #뒤로
    elif stat == 'BACKWORD':
        GPIO.output(INA, GPIO.LOW)
        GPIO.output(INB, GPIO.HIGH)
    #정지
    elif stat == 'STOP':
        GPIO.output(INA, GPIO.LOW)
        GPIO.output(INB, GPIO.LOW)

# 온도센서의 값을 읽어 출력할 준비를 하는 함수
def read_temp_raw ():
    f = open (device_file, 'r')
    lines = f.readlines()
    f.close()
    return lines

# 출력할 준비가 된 값을 받아 계산을 해서 온도로 출력해주는 함수
def read_temp ():
    lines = read_temp_raw()
    while lines[0].strip()[-3:] != 'YES':
        time.sleep(0.2 )
        lines = read_temp_raw()
    equals_pos = lines[1].find('t=')
    if equals_pos != -1 :

```

```

        temp_string = lines[1][equals_pos+2:]
        temp_c = float(temp_string) / 1000.0
        temp_f = temp_c * 9.0 / 5.0 + 32.0
        return temp_c

# 아날로그 신호를 읽어 처리해주는 부분
def read_ADC():
    bus.read_byte(add)
    analog = bus.read_byte(add)
    return analog

# 작동에 필요한 각종 변수
servo_cnt = 0
start = 0
water = 0
temperature = 0

# DB에 연결해 사용자가 사용할 준비가 되었는지 확인하는 SQL문
cursor = db.cursor(pymysql.cursors.DictCursor)
sql = "SELECT * FROM u_db WHERE u_last = 1"

# 메인 실행문이고 반복문을 통해 반복함
while True:
    try:
        if (start == 0):
            sql = "SELECT * FROM u_db WHERE u_last = 1"
            len = cursor.execute(sql)
            for val in cursor:
                print(val)
                water = val['u_water']
                temperature = val['u_temperature']

            if (len > 0):
                setServoPos(90)
                time.sleep(1)
                setServoPos(0)
                start = 1

        if (len > 0):
            # 사용자가 사용할 준비가 되었다면 온도, 수위, 워터펌프 센서를 작동시킴
            bus.write_byte(add, 0x00)
            ain0 = read_ADC()
            setMotorContorl(19, 13, 'FORWARD')
            setMotorContorl(6, 5, 'FORWARD')
            print("temperature value = %3d" % read_temp())
            print("moisture value = %3d" % (ain0))

            # 사용자가 설정한 값만큼 된다면 모든 센서가 멈춤
            if (ain0 >= water + 50):
                servo_con(0)
                setMotorContorl(19, 13, 'STOP')
                setMotorContorl(6, 5, 'STOP')
                sql = 'UPDATE u_db set u_last=0 where u_last=1'
                cursor.execute(sql)
                start = 0
                db.commit()

            else:
                print('sensor close')

        time.sleep(1)
        db.commit()
    except KeyboardInterrupt:
        GPIO.cleanup()
    except Exception as e:
        GPIO.cleanup()
        print(e)

```

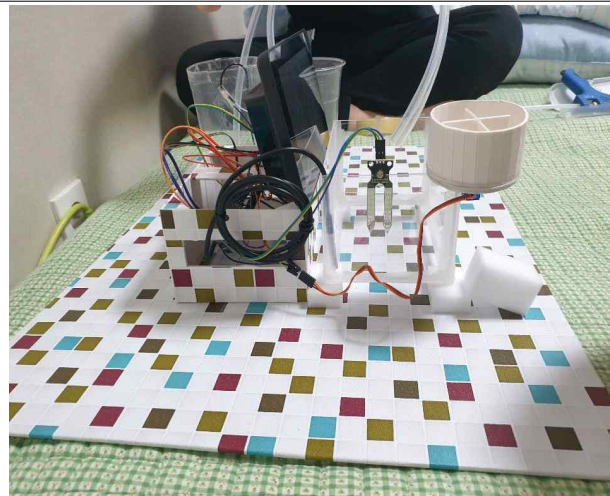
□ 시제품 모습



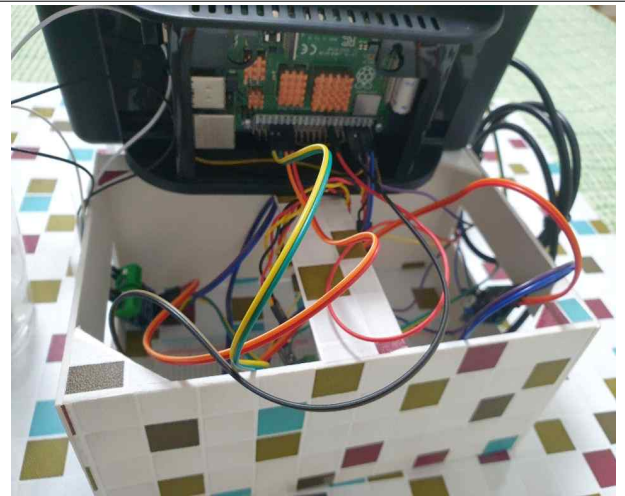
전체적인 모습



위에서 본 모습



측면 모습



후면 모습

3-7) I-Tub 결론 및 향후 보완점

(https://github.com/Ultimate-ItubProject/weekly_Report/tree/master/Final_report 의 시연영상 참고)

제작한 웹페이지와 I-Tub 시제품을 사용한 시연이 정상적으로 이루어졌으며 설계 했던대로 정상적으로 작동이 되었다. 입욕제의 크기가 작아 향, 색의 변화가 크게 나타나지 않아 추후 더 큰 입욕제를 사용하여 해결할 것이다. 그리고 아날로그 신호로 작동하는 서보모터를 라즈베리파이에서 완전히 제어하는데 한계가 있어 떨림 현상이 발생하였다. 해당 문제는 정밀한 콘덴서를 부착하면 해결이 되어 추후 보완하기로 하였다. 마지막으로 워터펌프의 경우 고온의 물을 사용할 때 발열이 심해져 망가지는 문제가 있는데 이에 대한 부분도 조사를 통해 고온의 물도 버티는 워터펌프로 교체할 예정이다.

데이터에 대한 부분도 보완점이 있었는데 작동상으로는 내부적으로 사용자가 사용을 할 때 마다 데이터가 누적되어 새로운 학습 모델로 예측을 하기 때문에 별도의 데이터의 추가 없이, 사용을 하면 할수록 보다 정확하고 좋은 작동 환경을 보일 것이다. 하지만 설문조사로 얻은 데이터를 사용하여 데이터의 신뢰성이 부족한 점이 보완점이라고 생각된다. 이에 대한 부분은 전문 설문사이트에서 더 많은 질문을 통해 신뢰성 확보와 더 많은 표본 확보를 통해 해결할 수 있을 것으로 보인다.

4. 기타

4-1) 프로젝트를 하며 느낀점

임대인 - 웹을 개발할 줄은 꿈도 안 꿔는데 프로젝트 주제를 선정하고 간추리다보니 개발하게 되었습니다. 단순히 HTML title, body와 같은 요소들만 알고 전혀 지식이 없었는데 직접 몸으로 부딪히며 공부를 하고 개발을 해나가 뜻깊은 시간이었던 것 같습니다. 웹 프론트엔드, 백엔드라는 용어만 알고 어떤 역할을 하고 무엇이 대표적인지도 몰랐는데 직접 홈페이지를 디자인해보며 css파일로 꾸며도 보고 웹 서버를 결정해 포트를 개방해 누구나 들어올 수 있게 만들고 원하는 기능을 하게하기 위해 자바스크립트를 공부해서 구현도 해보고 실제로 회사의 프로젝트를 맡았다는 기분도 들었습니다. 무엇이 부족한지 직접 만들어가며 느끼고 그 부분을 채워나가고 시간이 부족하거나 도저히 이해하지 못하는 부분은 나보다 뛰어난 부분이 있는 팀원의 도움을 받아 채워나가고, 4년간 어떤 수업보다도 현실성 있는 수업이었던 것 같습니다. 아쉬운 부분이 있다면 처음부터 웹 서버를 장고라는 플랫폼으로 했다면 IoT 센서 코딩이나 머신러닝에 대한 부분과 자연스럽게 이어지고 호환성도 좋았을 텐데 Node.js라는 플랫폼을 선택해 자바와 파이썬의 연동을 하기위해 엄청난 고생을 한 것 같습니다. 그래도 이러한 경험을 통해 왜 계획이 중요하고 프로그램의 특성, 요즘 트렌드, 내가 하고자하는 것을 잘 지원하는지를 프로젝트가 시작하기 전 분석을 하는게 매우 중요하다고 느낄 수 있었습니다. 그리고 또 다른 아쉬운 점은 팀원이 중도 포기를 한 점이 아쉬웠습니다. 상대적으로 전공 지식이 부족한 친구이다 보니 코딩보다는 단순한 노가다 작업, 문서 편집만 맡겼는데 그런 부분이 자신감을 하락시키고 나는 여기에 안 맞구나 라는 생각을 만든 것 같아 굉장히 미안하게 생각했습니다. 나름 이런 역할을 주면 쉬우니까 불만이 없겠지? 라고 배려했다고 판단했는데 그건 저만의 생각이었던 것 같습니다. 하지만 이런 경험도 밑거름 삼아 앞으로 취직을 해서 일어날 돌발 상황을 대처하는데 큰 도움이 될것같습니다. 한 학기동안 수고하셨습니다!

정해민 - 서정욱 팀원과 함께 프론트엔드 개발 파트를 맡아 사용자에게 직접적으로 보여지는 웹페이지를 제작하였습니다. 이번 캡스톤 프로젝트를 진행하기 전까지는 웹에 대해서는 잘 알지못하였습니다. 특히 프로젝트 진행 중에 css를 이용한 웹페이지 요소들의 디자인과 배치에 관해 어려움을 많이 겪곤 하였습니다. 하지만 생활코딩 웹사이트 강의와 유튜브, 구글링을 통해서 필요한 기술들을 습득하고, 팀원들과의 소통을 통해 극복해나갔습니다. 누구 하나 빠짐없이 잘하고 열심히하는 팀원들을 만나게되어 너무나도 뜻깊은 시간이 되었습니다. 학기 초에는 코로나로 인해 진행된 비대면 수업과 대면 수업 이후에는 제가 타지에서 통학을 해야하는 피치못할 사정으로 인해 주로 의사소통을 온라인으로 하게된 점과 팀을 위해 좀 더 많은 도움을 주지 못한 것에 대한 미안함과 아쉬움이 남았습니다.

서정욱 - 저는 이번 프로젝트에서 주로 웹 프론트엔드 부분을 맡았습니다. 사용자에게 보이는 웹페이지를 제작하면서 어떤 모습으로 보이는 게 좋을지 많은 고민을 해보았고, 추가적인 기능이 어떤 것이 있으면 좋을지도 많이 고민해보았습니다. 하지만 어떤 식으로 보여주어야 시각적으로 좋을지에 대해서는 능력이 많이 떨어졌기 때문에 팀원들에게 많은 도움을 구하기도 하고, 백엔드에서 무엇을 해야 하는지 물어보기도 하면서 서로 부족한 부분을 채워주면서 함께 프로젝트를 진행하였습니다. 이 프로젝트를 하면서 느낀바가 굉장히 많았습니다. 우선 프로젝트 시작부터 주제선정에 대해서 어떤 주제를 선정해야 할지 굉장히 어려움을 겪었습니다. 막상 아이디어를 내려고 해도 이미 존재하는 아이템이거나, 과연 그 아이디어가 쓸모가 있는지 의문투성이였습니다. 그리고 프로젝트를 진행하기 전에도 계획단계가 왜 중요한지 알게 되었습니다. 처음에는 계획 같은걸 따로 세우지 않고, 무작정 하다 보니 내가 무엇을 해야 하는지도 모르겠고, 프로젝트의 진척도 없었습니다. 하지만 다 같이 계획을 세우고 하나하나 무엇을 해야 할지 의논하면서 계획을 수립하면서, 제가 무엇을 해야 할지 알게 되고, 프로젝트도 점점 순조롭게 진행되면서, 프로젝트에 대해 무작정 뛰어드는 것이 아닌 사전작업이 정말 중요하다는 것을 알게 되었습니다. 그리고 추가적으로 현재 백엔드에서 일하시고 있는 분과 얘기를 몇 번 나누었는데, 저는 이번 프로젝트를 하면서 어느 정도 웹에 대해 안다고 생각하고 있었지만, 종사자의 입장에서 '너는 정말 아무것도 모르구나' 라는 생각이 들었을만큼 제가 아는 것이 없다고 하였습니다. 일단 시작하는 것도 좋지만, 전체적인 개념을 알고 간다고 하였

습니다. 아는 것이 많으면 좋다고는 하지만 두루뭉술하게만 알 뿐 깊이 있게 배우지는 않았습니다. 그래서 이번 방학에는 새로운 언어나 기술을 배우는 것이 아니라 제가 앞으로 무엇을 하고 싶은지에 대해 목표를 세우고 좀 더 깊이 있게 공부하고자 합니다.

박지수 - 데이터베이스설계 및 시제품제작역할을 맡았습니다. 데이터베이스를 최근에 배웠음에도 불구하고 많은 관심이 생겨 완벽한 모습은 아니지만 내가 사용할 수 있다는 모습을 찾아가는 것이 새로웠습니다. 좀 더 완벽한 모습의 데이터베이스를 구현할 수 있을때 까지 학습을 계속해 나가려합니다. 또한 시제품을 만드는 것에 있어 보조적인 부분을 맡았지만 역시나 새로운 것을 만든다는 것은 재미있는 일이였습니다. 내가 프로젝트를 진행함에 있어서 주 역할이아니더라도 다른 역할에 대한 자세한 세부내용들이 역할수행에 있어 영향을 많이 미친다는 것을 느꼈습니다. 여기서 무언가를 개발한다는 것은 자신이 배워왔던 모든 기초적인부분에서부터 응용적인 부분까지 필요하고 이것이 뒷받침될 때 팀원들과 소통하고 수월한 팀 프로젝트를 진행할 수 있다는 것을 느꼈습니다. 이번 팀 프로젝트를 진행하면서 내가 타인과 같은 출발선에서 시작한다하더라도 사람마다 개인의 능력과 노력 그리고 목표와 마음가짐이 다다른데 내가 타 팀원보다 모든 면에 있어서 많이 부족하다는 것을 느꼈습니다. 팀 프로젝트를 진행한 3개월 사이에 내가 팀원들에게 많이 도와주지 못한 것과 노력하지 않은 부분이 많았기에 후에 있을 팀 프로젝트에서는 변한모습을 보여주고 싶습니다. 이번프로젝트가 끝나고, 개인적인 기초능력을 다시금 재정비하고 응용할 수 있는 모습까지 발전시키고자하며 행하는 모든 부분에 있어 목표와 다짐을 가지고자합니다.

4-2) 참고문헌

<https://opentutorials.org/course/1> (자바스크립트 사용법)

<https://opentutorials.org/course/3780/18031> (HTML 및 CSS사용법)

<https://github.com/dabeng/OrgChart> (온라인 모드에서 통계를 보여줄 OrgChart 오픈소스)

<https://dabeng.github.io/OrgChart/> (orgchart 각종 예제)

https://win.adrirobot.it/sensori/moisture_sensor/moisture_sensor.html(토양습도센서 예제)

<https://blog.naver.com/elepartsblog/221726825667> (라즈베리파이 사용하기 위한 라즈비안 설치법)

<https://www.raspberrypi.org/> (라즈베리파이 사용하기 위한 라즈비안 설치 자료실)

<https://www.circuitbasics.com/raspberry-pi-ds18b20-temperature-sensor-tutorial/>(온도센서 예제)

<https://blog.naver.com/rhrkdfus/221373635978> (라즈비안 무선마우스 장애 해결)

<https://www.kocoafab.cc/fboard/view/1089> (미니브레드보드 원리 이해)

<https://swiperjs.com/api/> (프로필 선택 시 사용한 스와이프 API)