

[10주차 (5월 21~27일)]

주제 : 지능형 욕조(Intelligent Tub)

빅데이터와 IoT 기술을 융합하여 웹을 사용해 서비스하는 플랫폼

팀 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

개발 동기 및 목적 :

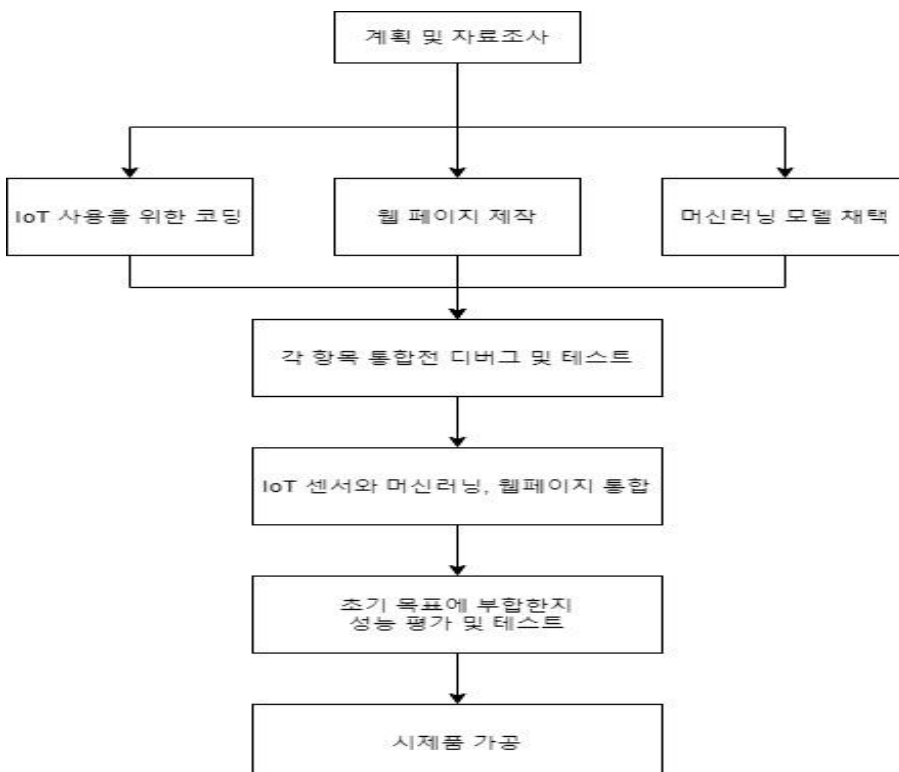
□ 개발동기

- 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 확실화 되는 것이 아쉬웠음, 더 나아가 사용자의 사용데이터, 사람들의 사용 데이터를 수집하여 데이터 분석을 통해 직업별, 날씨별과 같이 샤워라는 활동에 추가적인 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

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											■	■	■

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: 1.44.2)

기대효과 및 활용방안 :

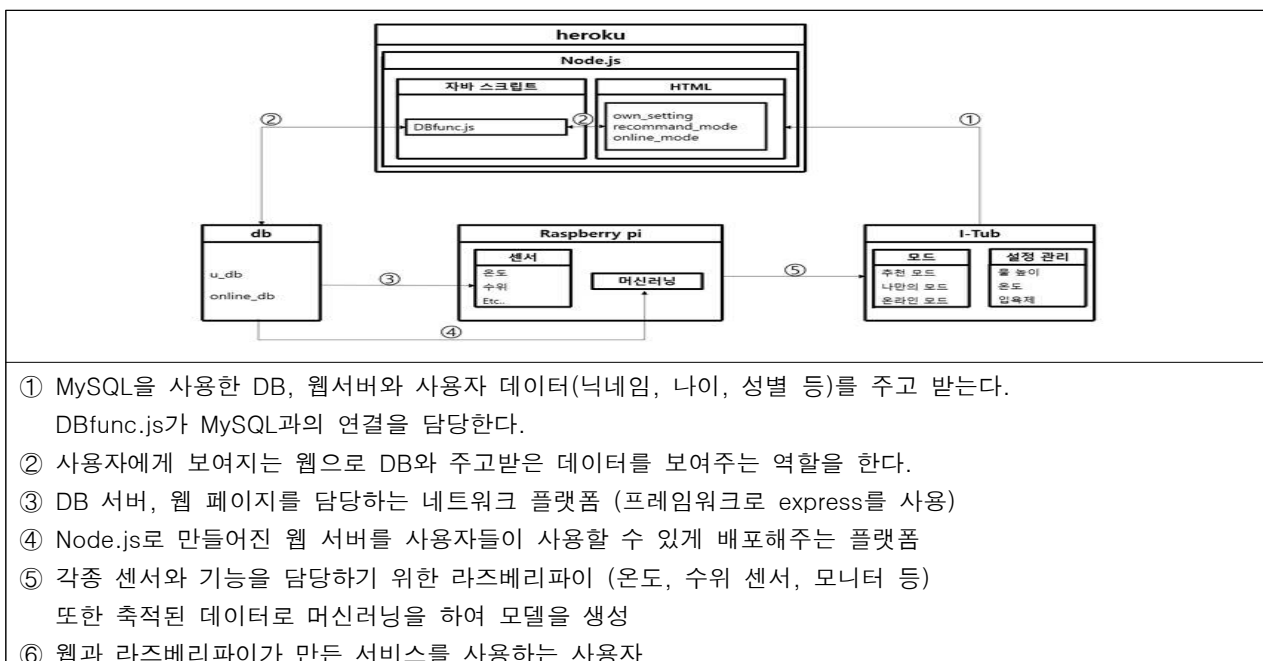
□ 기대효과

- 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
- 육조만 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
- 육조가 없는 환경이더라도 키트의 모양을 약간 변경한다면 사용가능
(부품의 기종을 변경, 내부 SW는 바꿀 필요가 없음)

시스템 흐름도 :



[금주진행사항]

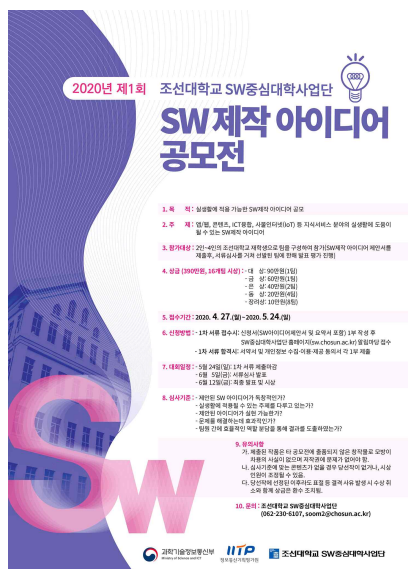
- 웹페이지 추가 (온라인 모드 페이지)

online_mode.html (online_mode_style.css)	온라인 모드 페이지 ----- 누적된 다른 사용자들의 목록 데이터를 통계, 분석하여 연령별, 시간별 등의 값을 추천해주는 페이지
---	---

- 설문조사 데이터 추가 수집

진행사항	- 통계, 분석 간 표본이 부족한 연령, 시간대 데이터가있어 추가적으로 데이터를 수집함
------	--

[대회 참가 서류 제출]



2020년 제1회 조선대학교 SW중심대학사업단 SW제작 아이디어 공모전

1. 목 : 실생활에 적용 가능한 SW제작 아이디어 공모

2. 주 : 팀명, 컨셉트, ICT활용, 시뮬레이션(Sim) 등 지식서비스 분야에 실생활에 도움이 될 수 있는 SW제작 아이디어

3. 참가대상 : 20~40대 조선대학교 재학생으로 팀을 구성하여 참가(최대 3명까지 가능하며, 반드시 팀원 중, 서투름을 거쳐 선정된 팀에 발표 참가 신청)

4. 보금 : 19000원, 1000원 시상 : 1대 상 5000원(3팀)
- 2대 상 5000원(2팀)
- 3대 상 4000원(2팀)
- 4대 상 3000원(2팀)
- 장려상 1000원(5팀)

5. 접수기간 : 2020. 4. 27(수) ~ 2020. 5. 24(화)

6. 신청방법 : 1차 서류 접수시 신청서(SW제작아이디어제출서 및 요약서 포함) 1부 작성 후 SW중심대학사업단 홈페이지(homesw.ac.kr) 공지사항 접수
- 1차 서류 합격자도 서식 및 기간제한 수정 가능(제출 후 1부 제출)

7. 대회일정 : 5월 24일(월) 1차 서류 제출 마감
5월 25일(화) 1차 서류 심사
5월 26일(수) 최종 발표 및 시상

8. 심사기준 : 제안된 SW 아이디어가 독창적인가?
- 실생활에 적용할 수 있는 문제를 지목하고 있는가?
- 제안된 아이디어가 실현 가능한가?
- 구체적 제품화나 서비스화 가능한가?
- 팀원 간에 효율적인 역할 분담을 통해 결과를 도출하는가?

9. 유의사항
가. 제출된 작품은 이 공모전에 출품하지 않은 창작물과 도안이
제출된 사실과 일치하지 않거나 저작권에 문제가 없어야 함
나. 심사위원에 의해 선정된 팀은 1주일 이내에 아이디어, 소문, 시상
접수할 수 있음
다. 선정된 팀은 SW중심대학사업단 홈페이지(homesw.ac.kr) 공지사항
소의 통해 유선한 경우 포함됨.

10. 문의 : 조선대학교 SW중심대학사업단
(020-230-5577, homesw@hoss.ac.kr)

과학기술정보통신부 IITP 조선대학교 SW중심대학사업단

○ SW 제작 아이디어 공모전

5/24까지 서류제출 ----- 진행 중

대회 참가 양식에 맞게 서류를 작성하여 제출을 완료했으며 6/5 1차 서류 통과가 된다면 지속적으로 대회 일정에 맞춰 진행할 예정

[금주구현코드]

online_mode.html

(누적된 다른 사용자들의 목욕 데이터를 통계, 분석하여 연령별, 시간별 등의 값을 추천해주는 페이지)

오늘의 추천 모드



```

<!DOCTYPE html >
<html >
<head >
  <meta charset ="utf-8">
  <meta name ="viewport" content ="width=device-width, initial-scale=1.0">
  <title >Online Mode </title >
  <!-- <link rel="stylesheet" href="./css/online style copy.css"> -->
</head >
<body >
  <div class ='wrap' style ='overflow:scroll ' >
    <div >
      <p ><b >오늘의 추천 모드 </b ></p >
      <div id ='today mode'></div >
    </div >
    <script type ="text/javascript" src ="./js/jquery-3.5.1.js"></script >
    <script type ="text/javascript" src ="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.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 type ="text/javascript" src ="https://www.gstatic.com/charts/loader.js"></script >
    <script >
      google .charts .load ('current', {
        packages: ["orgchart"]
      });
      google .charts .setOnLoadCallback (drawChart );
      function drawChart () {
        $.get ("shower test.csv", function (csvString ) {
          var arrayData = $.csv .toArray (csvString , {
            onParseValue: $.csv .hooks .castToScalar
          })
          //배열전체길이 (합산의 평균 구할때 사용)
          var arraylength = arrayData .length - 1 ;
          /*
          배열공통구성내용
          [0] :: 시작시간
          [1] :: 사용시간
          [2] :: 사용온도
          [3] :: 입욕재종류
          */
          //-----age-----

          //배열생성및 초기화
          var agearray = new Array (10 );
          var agetemp = new Array (10 );
          var agelength = new Array (10 );
          for (var k = 0 ; k < 10 ; k ++ ) {
            //각age에 맞는 정보 넣기
            agearray [k ] = new Array (4 );
            agearray [k ][0 ] = agearray [k ][1 ] = agearray [k ][2 ] = agearray [k ][3 ] = 0 ;

            //agetemp :: 온도카운팅
            agetemp [k ] = new Array (3 );
            agetemp [k ][0 ] = agetemp [k ][1 ] = agetemp [k ][2 ] = 0 ;
            agelength [k ] = 0 ;
          }
        });
      }
    </script >
  </div >
</body >
</html >

```

```

    }
    //나이에 따른 값저장
    for (var k = 1 ; k < arrayData .length ; k ++ ) {
        //나이에따라 1~9 index구분 : 「 자신의나이/10의 1 값
        //age_data는 해당 record의 age값의 index 찾아가는 값
        var age_data = parseInt (arrayData [k ][2 ] / 10 );
        //insert되는 age수 카운팅
        agelength [age_data ] += 1 ;
        //온도사용정보 카운팅
        if (arrayData [k ][3 ] == "시원함") {
            agetemp [age_data ][0 ] += 1 ;
        } else if (arrayData [k ][3 ] == "미지근함") {
            agetemp [age_data ][1 ] += 1 ;
        } else if (arrayData [k ][3 ] == "따뜻함") {
            agetemp [age_data ][2 ] += 1 ;
        }
        //맞는 index에 시작시간 합
        agearray [age_data ][0 ] += arrayData [k ][4 ];
        //맞는 index에 사용시간 합
        agearray [age_data ][1 ] += arrayData [k ][5 ];
    }
    //평균값 합산
    for (var k = 1 ; k < 10 ; k ++ ) {
        var ageindex = -1 ;
        var agevalue = 0 ;
        for (var i = 0 ; i < 3 ; i ++ ) {
            if (agetemp [k ][i ] > agevalue ) {
                ageindex = i ;
                agevalue = agetemp [k ][i ];
            }
        }
        if (ageindex == 0 ) {
            agearray [k ][2 ] = "시원함";
        } else if (ageindex == 1 ) {
            agearray [k ][2 ] = "미지근함";
        } else if (ageindex == 2 ) {
            agearray [k ][2 ] = "따뜻함";
        } else {
            agearray [k ][2 ] = "온도없음";
        }
        agearray [k ][0 ] = parseInt (agearray [k ][0 ] / agelength [k ]);
        agearray [k ][1 ] = parseInt (agearray [k ][1 ] / agelength [k ]);
    }
    // console.log(agearray)
    //-----age-----
    //-----gender-----

    //배열생성및 초기화
    var genderarray = new Array (2 );
    var gendertemp = new Array (2 );
    var genderlength = new Array (2 );
    for (var k = 0 ; k < 2 ; k ++ ) {
        genderarray [k ] = new Array (4 );
        genderarray [k ][0 ] = genderarray [k ][1 ] = genderarray [k ][2 ] = genderarray
[k ][3 ] = 0

        gendertemp [k ] = new Array (3 );
        gendertemp [k ][0 ] = gendertemp [k ][1 ] = gendertemp [k ][2 ] = 0
        genderlength [k ] = 0 ;
    }
    //성별에 따른 값저장
    for (var k = 1 ; k < arrayData .length ; k ++ ) {
        //성별에따라 index조정
        var genderindex = -1 ;
        if (arrayData [k ][1 ] == "남") {
            genderindex = 1 ;
        } else if (arrayData [k ][1 ] == "여") {
            genderindex = 0 ;
        }
        genderlength [genderindex ] += 1 ;
        if (genderindex == -1 ) continue ;
        //온도사용정보 카운팅
        if (arrayData [k ][3 ] == "시원함") {
            gendertemp [genderindex ][0 ] += 1 ;
        } else if (arrayData [k ][3 ] == "미지근함") {
            gendertemp [genderindex ][1 ] += 1 ;
        } else if (arrayData [k ][3 ] == "따뜻함") {
            gendertemp [genderindex ][2 ] += 1 ;
        }
        //맞는 index에 시작시간 합
        genderarray [genderindex ][0 ] += arrayData [k ][4 ];
        //맞는 index에 사용시간 합
        genderarray [genderindex ][1 ] += arrayData [k ][5 ];
    }
    // console.log(genderlength)
    //평균값 합산
    for (var k = 0 ; k < 2 ; k ++ ) {

```

```

var genderindex = -1 ;
var gendervalue = 0 ;
for (var i = 0 ; i < 3 ; i ++ ) {
    if (gendertemp [k ][i ] > gendervalue ) {
        genderindex = i ;
        gendervalue = gendertemp [k ][i ] ;
    }
}
if (genderindex == 0 ) {
    genderarray [k ][2 ] = "지원함";
} else if (genderindex == 1 ) {
    genderarray [k ][2 ] = "미지근함";
} else if (genderindex == 2 ) {
    genderarray [k ][2 ] = "따뜻함";
} else {
    genderarray [k ][2 ] = "온도없음";
}
genderarray [k ][0 ] = parseInt (genderarray [k ][0 ] / genderlength [k ]);
genderarray [k ][1 ] = parseInt (genderarray [k ][1 ] / genderlength [k ]);
}
// console.log(genderarray);
//-----gender-----

//-----job-----

//배열생성및 초기화
var jobarray = new Array (2 );
var jobtemp = new Array (2 );
var joblength = new Array (14 );
for (var k = 0 ; k < 14 ; k ++ ) {
    jobarray [k ] = new Array (4 );
    jobarray [k ][0 ] = jobarray [k ][1 ] = jobarray [k ][2 ] = jobarray [k ][3 ]

    jobtemp [k ] = new Array (3 );
    jobtemp [k ][0 ] = jobtemp [k ][1 ] = jobtemp [k ][2 ] = 0
    joblength [k ] = 0 ;
}
//직업에 따른 값저장
for (var k = 1 ; k < arrayData .length ; k ++ ) {
    //직업에따라 index조정
    var jobindex = -1 ;
    if (arrayData [k ][8 ] == "무직") jobindex = 0 ;
    else if (arrayData [k ][8 ] == "학생") jobindex = 1 ;
    else if (arrayData [k ][8 ] == "군인") jobindex = 2 ;
    else if (arrayData [k ][8 ] == "주부") jobindex = 3 ;
    else if (arrayData [k ][8 ] == "자영업자") jobindex = 4 ;
    else if (arrayData [k ][8 ] == "농업") jobindex = 5 ;
    else if (arrayData [k ][8 ] == "서비스업") jobindex = 6 ;
    else if (arrayData [k ][8 ] == "회사원") jobindex = 7 ;
    else if (arrayData [k ][8 ] == "공무원") jobindex = 8 ;
    else if (arrayData [k ][8 ] == "건설업") jobindex = 9 ;
    else if (arrayData [k ][8 ] == "연구원") jobindex = 10 ;
    else if (arrayData [k ][8 ] == "간호사") jobindex = 11 ;
    else if (arrayData [k ][8 ] == "회계사") jobindex = 12 ;
    else if (arrayData [k ][8 ] == "한의사") jobindex = 13 ;
    // else if (arrayData[k][8] == "") jobindex = 14;
    joblength [jobindex ] += 1 ;
    if (jobindex == -1 ) continue ;
    //온도사용정보 카운팅
    if (arrayData [k ][3 ] == "지원함") {
        jobtemp [jobindex ][0 ] += 1 ;
    } else if (arrayData [k ][3 ] == "미지근함") {
        jobtemp [jobindex ][1 ] += 1 ;
    } else if (arrayData [k ][3 ] == "따뜻함") {
        jobtemp [jobindex ][2 ] += 1 ;
    }
    //맞는 index에 시작시간 합
    jobarray [jobindex ][0 ] += arrayData [k ][4 ] ;
    //맞는 index에 사용시간 합
    jobarray [jobindex ][1 ] += arrayData [k ][5 ] ;
}
//평균및 합산
for (var k = 0 ; k < 14 ; k ++ ) {
    var jobindex = -1 ;
    var jobvalue = 0 ;
    for (var i = 0 ; i < 3 ; i ++ ) {
        if (jobtemp [k ][i ] > jobvalue ) {
            jobindex = i ;
            jobvalue = jobtemp [k ][i ] ;
        }
    }
    if (jobindex == 0 ) {
        jobarray [k ][2 ] = "지원함";
    } else if (jobindex == 1 ) {
        jobarray [k ][2 ] = "미지근함";
    }
}

```

```

    } else if (jobindex == 2) {
        jobarray[k][2] = "따뜻함";
    } else {
        jobarray[k][2] = "온도없음";
    }
    jobarray[k][0] = parseInt(jobarray[k][0] / joblength[k]);
    jobarray[k][1] = parseInt(jobarray[k][1] / joblength[k]);
}
// console.log(jobarray);
//-----job-----

//-----weather-----

var weatherarray = new Array(2);
var weathertemp = new Array(2);
var weatherlength = new Array(9);
for (var k = 0; k < 9; k++) {
    weatherarray[k] = new Array(4);
    weatherarray[k][0] = weatherarray[k][1] = weatherarray[k][2] = weatherarray[k][3] = 0;
    weathertemp[k] = new Array(3);
    weathertemp[k][0] = weathertemp[k][1] = weathertemp[k][2] = 0;
    weatherlength[k] = 0;
}
//온도에 따른 값저장
for (var k = 1; k < arrayData.length; k++) {
    //온도에 따라 index조정
    var weatherindex = -1;
    var weatherindex = parseInt(parseInt(arrayData[k][9]) / 5);
    weatherlength[weatherindex] += 1;
    if (weatherindex == -1) continue;
    //온도사용정보 카운팅
    if (arrayData[k][3] == "시원함") {
        weathertemp[weatherindex][0] += 1;
    } else if (arrayData[k][3] == "미지근함") {
        weathertemp[weatherindex][1] += 1;
    } else if (arrayData[k][3] == "따뜻함") {
        weathertemp[weatherindex][2] += 1;
    }
    //맞는 index에 시작시간 합
    weatherarray[weatherindex][0] += arrayData[k][4];
    //맞는 index에 사용시간 합
    weatherarray[weatherindex][1] += arrayData[k][5];
}
//평균값 합산
for (var k = 0; k < 9; k++) {
    var weatherindex = -1;
    var weathervalue = 0;
    for (var i = 0; i < 3; i++) {
        if (weathertemp[k][i] > weathervalue) {
            weatherindex = i;
            weathervalue = weathertemp[k][i];
        }
    }
    if (weatherindex == 0) {
        weatherarray[k][2] = "시원함";
    } else if (weatherindex == 1) {
        weatherarray[k][2] = "미지근함";
    } else if (weatherindex == 2) {
        weatherarray[k][2] = "따뜻함";
    } else {
        weatherarray[k][2] = "온도없음";
    }
    weatherarray[k][0] = parseInt(weatherarray[k][0] / weatherlength[k]);
    weatherarray[k][1] = parseInt(weatherarray[k][1] / weatherlength[k]);
}
// console.log(weatherarray);
//-----weather-----

//-----orgchart-----

var todaydata = new google.visualization.DataTable();
todaydata.addColumn('string', 'Name');
todaydata.addColumn('string', 'Manager');
todaydata.addColumn('string', 'ToolTip');
// For each orgchart box, provide the name, manager, and tooltip to show.
todaydata.addRow([
    [
        {
            'v': 'today',
            'f': 'today'
        },
        'The President'
    ],
    [
        {
            'v': 'age',
            'f': 'age'
        }
    ]
]);

```

```

        'today', 'VP'
    ],
    [{
        'v': 'gender',
        'f': 'gender'
    },
    'today', 'VP'
    ],
    [{
        'v': 'job',
        'f': 'job'
    },
    'today', 'VP'
    ],
    [{
        'v': 'weather',
        'f': 'weather'
    },
    'today', 'VP'
    ],
    ['age', 'today', ''],
    ['gender', 'today', ''],
    ['남', 'gender', ''],
    ['여', 'gender', '']
    );
    //age구성
    for (var i = 1 ; i < 10 ; i++) {
        /*agearray
        agearray[4]
        */
        todaydata .addRows ([
            [(i * 10 ).toString () + '대', 'age', ''],
            ['<p>나이</p><div>' + agearray [i ][0 ] +
            '</div><div>' + agearray [i ][1 ] + '</div>' +
            '</div><div>' + agearray [i ][2 ] + '</div>' +
            '</div><div>' + agearray [i ][3 ] + '</div>', (i * 10 ).toString
            () + '대', ''
            ]
        )
    }
    //gender구성
    for (var i = 0 ; i < 2 ; i++) {
        todaydata .addRows ([
            ['<p>남자</p><div>' + genderarray [i ][0 ] +
            '</div><div>' + genderarray [i ][1 ] + '</div>' +
            '</div><div>' + genderarray [i ][2 ] + '</div>' +
            '</div><div>' + genderarray [i ][3 ] + '</div>', '남', ''
            ],
            ['<p>여자</p><div>' + genderarray [i ][0 ] +
            '</div><div>' + genderarray [i ][1 ] + '</div>' +
            '</div><div>' + genderarray [i ][2 ] + '</div>' +
            '</div><div>' + genderarray [i ][3 ] + '</div>', '여', ''
            ]
        )
    }
    console .log (jobarray )
    //job구성
    for (var i = 0 ; i < 14 ; i++) {
        var myjob = "";
        switch (i ) {
            case 0 :
                myjob = "무직";
                break ;
            case 1 :
                myjob = "학생";
                break ;
            case 2 :
                myjob = "군인";
                break ;
            case 3 :
                myjob = "주부";
                break ;
            case 4 :
                myjob = "자영업자";
                break ;
            case 5 :
                myjob = "농업";
                break ;
            case 6 :
                myjob = "서비스업";
                break ;
            case 7 :
                myjob = "회사원";
                break ;
            case 8 :
                myjob = "공무원";
                break ;
        }
    }

```



```

        case 9 :
            myjob = "건설업";
            break ;
        case 10 :
            myjob = "연구원";
            break ;
        case 11 :
            myjob = "간호사";
            break ;
        case 12 :
            myjob = "회계사";
            break ;
        case 13 :
            myjob = "한의사";
            break ;
    }
    console .log (myjob )
    todaydata .addRows ([
        ['<p>' + myjob + '</p><div>' + jobarray [i ][0 ] +
        '</div><div>' + jobarray [i ][1 ] + '</div>' +
        '</div><div>' + jobarray [i ][2 ] + '</div>' +
        '</div><div>' + jobarray [i ][3 ] + '</div>', 'job', ''
    ])
    })
    //weather구성
    for (var i = 0 ; i < 9 ; i ++ ) {
        todaydata .addRows ([
            ['<p>' + i * 5 + '°C</p><div>' + weatherarray [i ][0 ] +
            '</div><div>' + weatherarray [i ][1 ] + '</div>' +
            '</div><div>' + weatherarray [i ][2 ] + '</div>' +
            '</div><div>' + weatherarray [i ][3 ] + '</div>', 'weather', ''
        ])
    }
    //drawchart option
    var options = {
        'allowHtml': true ,
        // nodeClass: 'myNodeClass',
        // selectedNodeClass: 'mySelectedNodeClass'
    }
    var chart = new google .visualization .OrgChart (document .getElementById
('today mode'));
    chart .draw (todaydata , options );
    })
    }
</script >
</body >
</html >

```

[차주예정사항]

머신 러닝	수집한 데이터를 바탕으로 머신 러닝 학습을 하여 회원에게 알맞은 모드를 추천해주는 기능을 추가 예정
차트 디자인 및 데이터 삽입	차트의 디자인 및 통계 분석 데이터 삽입 (보다 높은 웹페이지 퀄리티를 만들기 위함)
데이터 추가	설문 데이터 추가 (표본이 적은 연령대나 시간대가 있어 추가적으로 데이터를 수집하고 있음)

[팀내건의사항]

없음