

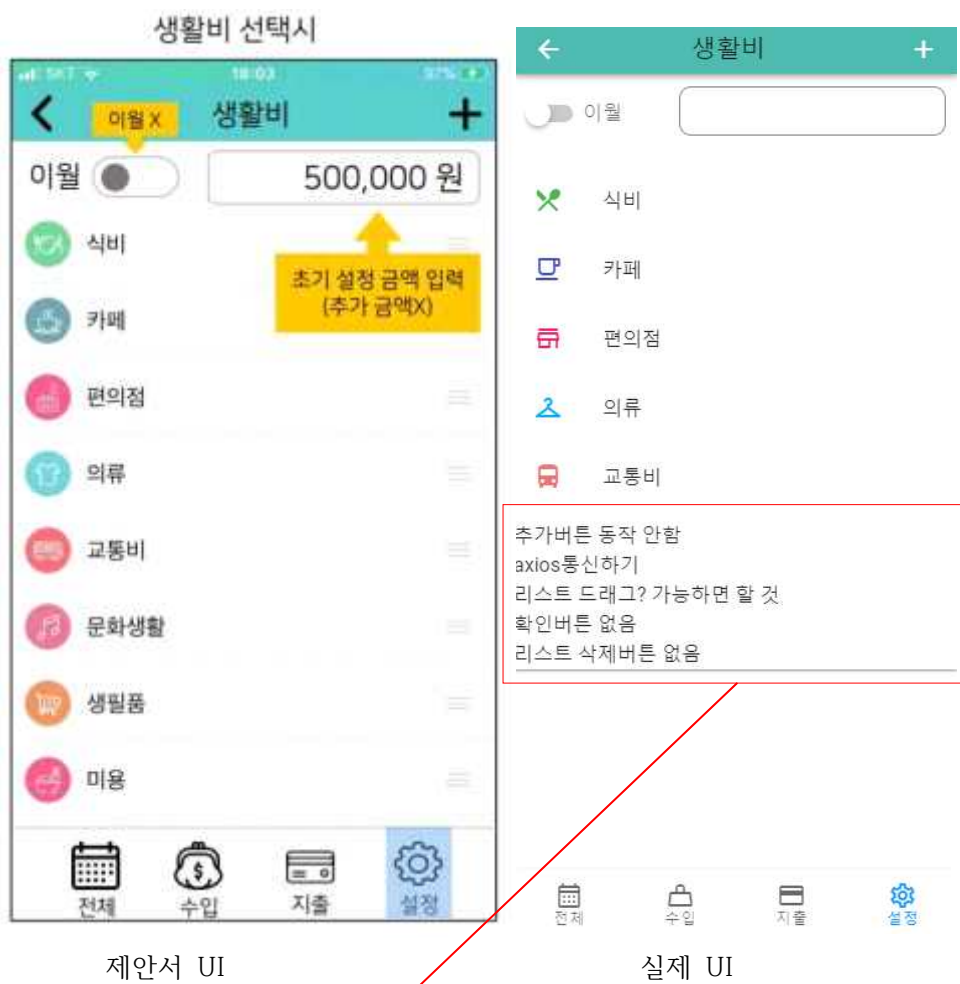
## 2주보고 2차시

기간: 5월 2일(토요일) - 5월 15일(금요일)

2012011038 이자룡

5월 3일

설정부 UI 제작



작업해야 할 목록들을 잊지 않기 위해 해당 컴포넌트에 메모해둬م.

5월 7일  
로그인 기능 ajax 테스트.

5월							
수입				지출			
900,000원				389,300원			
현금 잔액				193,500			
510,700원				카드			
				195,800			
일	월	화	수	목	금	토	일
					1	2	
						24,000	
3	4	5	6	7	8	9	
		43,500	150,000	6,300		220,000	
10	11	12	13	14	15	16	
			150,000		300,000		
17	18	19	20	21	22	23	
	80,000		150,000	7,000			
24	25	26	27	28	29	30	
8,500			150,000				
31							

5월							
수입				지출			
2,600,000원				183,300원			
현금 잔액				1,800			
2,416,700원				카드			
				181,500			
일	월	화	수	목	금	토	일
					1	2	
						17,000	
3	4	5	6	7	8	9	
		50,500		2,100,000			
				13,000			
10	11	12	13	14	15	16	
					500,000		
17	18	19	20	21	22	23	
	51,200			20,800			
24	25	26	27	28	29	30	
15,000					1,800		
31							

5월 13일

회원가입 기능 ajax 테스트

```
// 하위 컴포넌트에서 data를 넘겨받아 회원가입 준비를 할
goMain(data) {
  this.signUpSet.userGoals = data[0]; // 유저의 목표
  this.signUpSet.userTotalProperty = Number(data[1]); // 유저의 총 자산
  this.signUpSet.incomeMonthly = Number(data[2]); // 유저의 월 수입

  this.signUpSet.spendFlexibleList = data[4]; // 유저의 변동수입
  for (let i = 0; i < this.signUpSet.spendFlexibleList[1].length; i++) {
    this.signUpSet.spendFlexibleList[1][i] = Number(this.signUpSet.spendFlexibleList[1][i]); // string형의 유저 변동수입을 int로 바꿈
  }
  for (let i = 0; i < this.signUpSet.spendFixedList[0].length; i++) {
    this.signUpSet.spendFixedList[1][i] = Number(this.signUpSet.spendFixedList[1][i]); // string 형의 유저 고정수입을 int로 바꿈
  }

  // 사용자 목표는 다섯 개 정해진 목표에 boolean형인 배열. 이것을 String배열로 바꿔줄 작업
  let a = [];
  for (let i = data[0].length - 1; i >= 0; i--) {
    if (data[0][i]) {
      if (i === 0)
        a.unshift('일년 적금 들기');
      else if (i === 1)
        a.unshift('100만원 모으기');
      else if (i === 2)
        a.unshift('일주일에 외식 한 번');
      else if (i === 3)
        a.unshift('카페 일주일 두 번');
      else
        a.unshift('전셋집 자금 마련하기');
    }
  }
  this.signUpSet.userGoals = a;

  // SignUp 기능을 하는 컴포넌트를 끄
  this.signUpUserSetShow = false;

  // 서버에 ajax통신으로 모아진 signUpSet 데이터를 전송함.
  axios.post('/php/signUp.php', this.signUpSet)
    .then(response => {
      if (response.data) {
        const d = new Date();
        this.data.monthData.thisYear = d.getFullYear() // 올해
        this.data.userData.month = d.getMonth() + 1; // 이번달
        this.data.monthData.thisMonth = d.getMonth() + 1; // 역시 이번달
        this.mainShow = true; // 메인화면 띄움
      }
    })
    .catch(error => {
      if (error)
        console.log("실패!");
    });
}
```

## 2014011051 제태경

### 1. 수정 및 개발

<login.php>

테스팅 과정에서 잘못된 left조인으로 인해 spend 테이블에 대한 정보는 join되지 않는 오류가 발견되어 sql문 전체를 수정 하였습니다 .

```
$res5 = mysqli_query($db, "SELECT Date_d, ifnull(f.i_price, 0), ifnull(f.i_Division, '+'), ifnull(f.s_price, 0), ifnull(f.s_Division, '-') from
(SELECT IFNULL(A.Date_d, B.Date_d) as Date_d, A.price as i_price, A.Division as i_Division, B.price as s_price, B.Division as s_Division
From
(SELECT MAX(Date_d) Date_d, sum(price) as price, MAX(Division) Division from (select SUBSTRING(Date_d, 9, 2)
as Date_d, sum((user.Change_income*work_income.Time)) as price, max(Division) Division from
work_income,user where work_income.ID = '{$SESSION["ses_username"]}' and user.ID = '{$SESSION["ses_username"]}' and month(Date_d) = Month(now()) GROUP BY Date_d UNION ALL select SUBSTRING(Date_d, 9, 2)
as Date_d, sum(price), max(Division) as a from income where ID = '{$SESSION["ses_username"]}' and month(Date_d) = Month(now()) GROUP by Date_d) as basetable group by Date_d) as A

LEFT OUTER JOIN

(SELECT SUBSTRING(Date_d, 9, 2) as Date_d, sum(price) as price, max(Division) as Division from spend where
ID = '{$SESSION["ses_username"]}' and month(Date_d) = Month(now()) GROUP by Date_d) as B
ON A.Date_d = B.Date_d

UNION

SELECT IFNULL(A.Date_d, B.Date_d), A.price, A.Division, B.price, B.Division
From
(SELECT SUBSTRING(Date_d, 9, 2) as Date_d, sum(price) as price, max(Division) as Division from spend where ID
= '{$SESSION["ses_username"]}' and month(Date_d) = Month(now()) GROUP by Date_d) as B

LEFT OUTER JOIN

(SELECT MAX(Date_d) Date_d, sum(price) as price, MAX(Division) Division from (select SUBSTRING(Date_d, 9, 2)
as Date_d, sum((user.Change_income*work_income.Time)) as price, max(Division) Division from
work_income,user where work_income.ID = '{$SESSION["ses_username"]}' and user.ID = '{$SESSION["ses_username"]}' and month(Date_d) = Month(now()) GROUP BY Date_d UNION ALL select SUBSTRING(Date_d, 9, 2)
as Date_d, sum(price), max(Division) as a from income where ID = '{$SESSION["ses_username"]}' and month(Date_d) = Month(now()) GROUP by Date_d) as basetable group by Date_d) as A
ON A.Date_d = B.Date_d) as f order by Date_d asc";
//이번달에 속하는 일, 금액, 구분
```

설명 : 유저의 이번달 (기타수입,노동수입,지출)을 날짜별로 합산을 하였습니다.

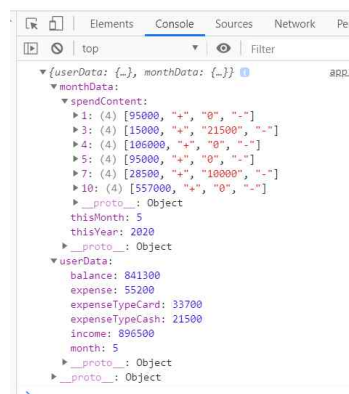
조인과정)

- 1.테이블 A = 기타수입테이블과 노동수입에 유저테이블을 활용해 시급계산을 한 테이블을 UNION하여 날짜별 수입의 합산을 구함  
테이블 B = 지출테이블의 날짜별 합산

2. (A와 B를 레프트조인) UNION (B와 A를 레프트조인)

3. 2의결과에서 날짜순으로 정렬후 null값인 데이터를 조작

결과 :



### <signUpIdCheck.php>

아이디 중복체크 부분을 따로 코드로 만들었습니다 . boolean형 리턴

### <json 객체 오류 전송 수정>

login.php, getMoneyDetail.php등의 코드처럼 객체로 클라이언트에 전송시 string으로 넘어가는 문제를 해결 하기 위해 encode 방법을 수정하였습니다.

### 2. 공부

409 연구실에서 박수진 팀원과, 3학년 소프트웨어 프로젝트 팀 중에서 똑같은 방식으로 개발을 진행하는 서버개발자에게 php와 sql을 가르쳐 주면서 저도 복습하는 시간을 가졌습니다.

### 3. 계획

아직 검증 못한 파일들을 수정하고 추가되는 ui에 대한 php개발을 진행할 예정입니다.

## 2017010834 박명인

### 1) 쿼리문

```
10 • SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
11 • SET AUTOCOMMIT = 0;
12 • START TRANSACTION;
13 • SET time_zone = "+00:00";
14
15
16 • /*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
17 • /*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
18 • /*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
19 • /*!40101 SET NAMES utf8mb4 */;
20
21 --
22 -- 데이터베이스: `accountbook`
23 --
24
25 -- -----
26
27 --
28 -- 테이블 구조 `category`
29 --
30
31 • CREATE TABLE `category` (
32   `Category_name` varchar(30) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
33   `Icon_color` varchar(30) COLLATE utf8mb4_general_ci NOT NULL,
34   `Icon_tag` varchar(30) COLLATE utf8mb4_general_ci NOT NULL
35 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
36
37 --
38 -- 테이블의 덤프 데이터 `category`
39 --
```

```

41 • INSERT INTO `category` (`Category_name`, `Icon_color`, `Icon_tag`) VALUES
42   ('경조사', 'yellow', 'yy;'),
43   ('비상금', 'green', 'gg;'),
44   ('생활비', 'red', 'mm;'),
45   ('수입', 'blue', 'bb;');
46
47 -----
48
49 --
50 -- 테이블 구조 `fix_spend`
51 --
52
53 • CREATE TABLE `fix_spend` (
54   `ID` varchar(20) COLLATE utf8mb4_general_ci NOT NULL,
55   `Category_Detail` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
56   `price` int(20) NOT NULL,
57   `num` int(5) NOT NULL
58 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
59
60 --
61 -- 테이블의 덤프 데이터 `fix_spend`
62 --
63
64 • INSERT INTO `fix_spend` (`ID`, `Category_Detail`, `price`, `num`) VALUES
65   ('sj1212', '자취방 월세', 380000, 1),
66   ('sj1212', '통신비', 52000, 2),
67   ('sj1212', '공과금', 20000, 3),
68   ('sj1212', '적금', 20000, 4),
69   ('tg123', '자취방 월세', 300000, 5),
70   ('tg123', '통신비', 50000, 6),
71   ('tg123', '공과금', 50000, 7),
72   ('tg123', '교통비', 50000, 8),
73   ('tg123', '적금', 100000, 9),
74   ('mi111', '전세대출금', 280000, 10),
75   ('mi111', '통신비', 56000, 11),
76   ('mi111', '인터넷비', 18000, 12),
77   ('mi111', '아파트 관리비', 120000, 13),
78   ('mi111', '교통비', 100000, 14),

```



```

79      ('mi111', '적금', 100000, 15),
80      ('mi111', '부모님 용돈', 300000, 16),
81      ('mi111', '담배', 140000, 17),
82      ('jr3', '아파트 관리비', 240000, 18),
83      ('jr3', '통신비', 120000, 19),
84      ('jr3', '인터넷비', 30000, 20),
85      ('jr3', '가솔비(차)', 150000, 21),
86      ('jr3', '적금', 1000000, 22),
87      ('jr3', '부모님 용돈', 500000, 23),
88      ('jr3', '가게 월세', 1020000, 25),
89      ('jr3', '가게 공과금', 250000, 26);
90
91      -----
92
93      --
94      -- 테이블 구조 `goal`
95      --
96
97      • CREATE TABLE `goal` (
98          `Content` varchar(60) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
99          `Point` int(50) NOT NULL
100      ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
101
102      --
103      -- 테이블의 덤프 데이터 `goal`
104      --
105
106      • INSERT INTO `goal` (`Content`, `Point`) VALUES
107          ('1년 적금 들거', 12);
108
109      -----
110
111      --
112      -- 테이블 구조 `income`
113      --
114
115      • CREATE TABLE `income` (
116          `ID` varchar(20) COLLATE utf8mb4_general_ci NOT NULL,
117          `Category_Detail` text CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,

```



```

237     ('jr3', '상할비', '식비', '팜플', '카드', '2020-04-21', 7000, '-', 39),
238     ('jr3', '상할비', '마트', '반찬', '카드', '2020-04-24', 8500, '-', 40),
239     ('m111', '상할비', '마트', '오이', '현금', '2020-04-29', 900, '-', 41),
240     ('m111', '상할비', '마트', '오이', '현금', '2020-04-29', 900, '-', 42);
241
242     -----
243
244     --
245     -- 테이블 구조 `spend_material`
246     --
247
248     CREATE TABLE `spend_material` (
249         `Material` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
250         `Image` varchar(50) COLLATE utf8mb4_general_ci NOT NULL,
251         `Text` text COLLATE utf8mb4_general_ci NOT NULL,
252         `Point` int(10) NOT NULL
253     ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
254
255     --
256     -- 테이블의 샘플 데이터 `spend_material`
257     --
258
259     INSERT INTO `spend_material` (`Material`, `Image`, `Text`, `Point`) VALUES
260     ('저축형', '0', '아껴쓰고 많이 저축', 50);
261
262     -----
263
264     --
265     -- 테이블 구조 `under_category`
266     --
267
268     CREATE TABLE `under_category` (
269         `Category_Detail` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
270         `Category_name` text COLLATE utf8mb4_general_ci NOT NULL,
271         `Icon_color` varchar(50) COLLATE utf8mb4_general_ci NOT NULL,
272         `Icon_tag` varchar(50) COLLATE utf8mb4_general_ci NOT NULL
273     ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
274

```

```

276 -- 테이블의 덤프 데이터 `under_category`
277 --
278
279 • INSERT INTO `under_category` (`Category_Detail`, `Category_name`, `Icon_color`, `Icon_tag`) VALUES
280 ('고통비', '생활비', 'blue', 'bb'),
281 ('문화생활', '생활비', 'navy', 'nn'),
282 ('미용', '생활비', 'pink', 'pp'),
283 ('생일', '경조사', 'black', 'bbb'),
284 ('생필품', '생활비', 'purple', 'pp'),
285 ('식비', '생활비', 'red', 'rr'),
286 ('월세', '생활비', 'mint', 'mm'),
287 ('의류', '생활비', 'green', 'gg'),
288 ('일금', '수입', 'red', '!@#'),
289 ('주식', '수입', 'blue', '!@#'),
290 ('카페', '생활비', 'Orange', 'oo'),
291 ('홍신', '생활비', 'scarlet', 'ss'),
292 ('편의점', '생활비', 'yellow', 'yy');
293
294 -----
295
296 --
297 -- 테이블 구조 `user`
298 --
299
300 • CREATE TABLE `user` (
301   `ID` varchar(20) COLLATE utf8mb4_general_ci NOT NULL,
302   `PW` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci DEFAULT NULL,
303   `Name` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci DEFAULT NULL,
304   `Sex` varchar(10) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci DEFAULT NULL,
305   `Age` varchar(10) COLLATE utf8mb4_general_ci DEFAULT NULL,
306   `Cash` int(20) DEFAULT NULL,
307   `Fix_Income` int(20) DEFAULT NULL,
308   `Change_income` int(20) DEFAULT NULL
309 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
310
311 --

```

```

315 • INSERT INTO `user` (`ID`, `PW`, `Name`, `Sex`, `Age`, `Cash`, `Fix_Income`, `Change_income`) VALUES
316 ('jr3', '3333', '이자돌', '남성', '30', 20000000, 700000, 30000),
317 ('mi111', '1111', '박명인', '여성', '30', 32000000, 1900000, 11875),
318 ('sj1212', '2222', '박수진', '여성', '20', 700000, 380000, 9000),
319 ('tg123', '4444', '채태경', '남성', '20', 1200000, 500000, 9500);
320
321 -----
322
323 --
324 -- 테이블 구조 `user_target`
325 --
326
327 • CREATE TABLE `user_target` (
328   `ID` varchar(10) COLLATE utf8mb4_general_ci NOT NULL,
329   `content` text COLLATE utf8mb4_general_ci NOT NULL,
330   `num` int(11) NOT NULL
331 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
332
333 --
334 -- 테이블의 샘플 데이터 `user_target`
335 --
336
337 • INSERT INTO `user_target` (`ID`, `content`, `num`) VALUES
338 ('tg123', '일년 짝금 돌기', 1),
339 ('mi111', '전세집 자금 마련하기', 2),
340 ('jr3', '200만원 그래픽카드 구매하기', 3),
341 ('sj1212', '에어팟 구매하기', 4);
342
343 -----
344
345 --
346 -- 테이블 구조 `work_income`
347 --
348
349 • CREATE TABLE `work_income` (
350   `ID` varchar(20) COLLATE utf8mb4_general_ci NOT NULL,
351   `Category_Detail` text CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL,
352   `Content` varchar(20) COLLATE utf8mb4_general_ci NOT NULL,
353   `Date_d` date NOT NULL,

```

## 2) phpmyadmin - vuetify의 색상표와 materialdesignicons 의 아이콘 사용

- category 테이블의 Icon\_color 와 Icon\_tag 태그를 불러와 삽입

			Category_name	Icon_color	Icon_tag
<input type="checkbox"/>			수정 복사 삭제 경조사	#F44336	mdi-spa
<input type="checkbox"/>			수정 복사 삭제 비상금	#4DD0E1	mdi-shield-key-outline
<input type="checkbox"/>			수정 복사 삭제 생활비	#29B6F6	mdi-emoticon-happy-outline
<input type="checkbox"/>			수정 복사 삭제 수입	#4CAF50	mdi-cash-100

- under\_category 테이블의 Icon\_color 와 Icon\_tag 태그를 불러와 삽입

			Category_Detail	Category_name	Icon_color	Icon_tag
<input type="checkbox"/>			수정 복사 삭제 가게 공과금	생활비	#1A237E	mdi-flash-outline
<input type="checkbox"/>			수정 복사 삭제 가게 월세	생활비	#0091EA	mdi-cash-minus
<input type="checkbox"/>			수정 복사 삭제 공과금	생활비	#00B0FF	mdi-flash
<input type="checkbox"/>			수정 복사 삭제 담배	생활비	#01579B	mdi-fire
<input type="checkbox"/>			수정 복사 삭제 마트	생활비	#0277BD	mdi-cart
<input type="checkbox"/>			수정 복사 삭제 문화생활	생활비	#0288D1	mdi-movie-open
<input type="checkbox"/>			수정 복사 삭제 미용	생활비	#039BE5	mdi-emoticon-cool
<input type="checkbox"/>			수정 복사 삭제 버스비	생활비	#40C4FF	mdi-bus
<input type="checkbox"/>			수정 복사 삭제 부모님 용돈	생활비	#4FC3F7	mdi-heart-plus
<input type="checkbox"/>			수정 복사 삭제 상금	수입	#1B5E20	mdi-seal-variant
<input type="checkbox"/>			수정 복사 삭제 생일	경조사	#455A64	mdi-flower-poppy
<input type="checkbox"/>			수정 복사 삭제 생필품	생활비	#81D4FA	mdi-account-cash
<input type="checkbox"/>			수정 복사 삭제 쇼핑	생활비	#83E5FC	mdi-basket
<input type="checkbox"/>			수정 복사 삭제 식비	생활비	#0D47A1	mdi-food-fork-drink
<input type="checkbox"/>			수정 복사 삭제 아파트 관리비	생활비	#448AFF	mdi-office-building
<input type="checkbox"/>			수정 복사 삭제 약국	생활비	#82B1FF	mdi-pill
<input type="checkbox"/>			수정 복사 삭제 용돈	수입	#2E7D32	mdi-heart-plus-outline
<input type="checkbox"/>			수정 복사 삭제 월급	수입	#388E3C	mdi-briefcase
<input type="checkbox"/>			수정 복사 삭제 유희	생활비	#1565C0	mdi-glass-mug-variant
<input type="checkbox"/>			수정 복사 삭제 인터넷비	생활비	#1976D2	mdi-microsoft-internet-explorer
<input type="checkbox"/>			수정 복사 삭제 월급	수입	#43A047	mdi-network-strength-2
<input type="checkbox"/>			수정 복사 삭제 자취방 월세	생활비	#1E88E5	mdi-cash-marker
<input type="checkbox"/>			수정 복사 삭제 전세 대출금	생활비	#42A5F5	mdi-bank-transfer
<input type="checkbox"/>			수정 복사 삭제 주급	수입	#00C853	mdi-network-strength-4
<input type="checkbox"/>			수정 복사 삭제 주식	수입	#66BB6A	mdi-share-variant
<input type="checkbox"/>			수정 복사 삭제 자 기쁨비	생활비	#2962FF	mdi-gas-station
<input type="checkbox"/>			수정 복사 삭제 카페	생활비	#3D5AFE	mdi-coffee
<input type="checkbox"/>			수정 복사 삭제 택시비	생활비	#00E5FF	mdi-taxi
<input type="checkbox"/>			수정 복사 삭제 통신비	생활비	#536DFE	mdi-cellphone-iphone
<input type="checkbox"/>			수정 복사 삭제 투자	수입	#558B2F	mdi-crop-landscape
<input type="checkbox"/>			수정 복사 삭제 편의점	생활비	#03A9F4	mdi-store-24-hour

## 2017010835 박수진

제태경 팀원이 작성한 코드를 바탕으로 php코드에 대해 학습을 진행하고  
차후 작성할 php코드에 대해 php와 sql문을 학습했습니다.

새로운 이름으로 회원가입을 했을 때 계속해서 특정이름이 뜨는 오류를 찾아 내 클라이언트에  
게 전달하여 오류를 해결했습니다.



해당 오류 화면. 같은 이름만 보여짐



오류 수정 후 사용자 설정이름이 보임

통계를 계산할 때 필요한 php코드를 작성중이며 다음 주 중으로 완료 할 계획입니다.  
-지정한 월에 해당하는 카테고리(생활비, 경조사...)의 세부카테고리 별 금액 합산

```
<?php
class Chart{
    public $label;
    public $value;
}

error_reporting(E_ALL);
ini_set("display_errors", 1);

require_once("dbconfig.php");

$_POST = JSON_DECODE(file_get_contents("php://input"), true);
//vue와 ajax통신을 위해 필요한 코드 json값을 디코딩

$month = $_POST["month"];
$category_name = $_POST["tag"];

$data = new Chart;

$sql = mysqli_query($db, "SELECT Category_Detail, sum(price)
                        FROM spend
                        WHERE ID = '". $_SESSION["ses_username"]."'
                        and Category_name = '". $category_name."'
                        and month(Date_d) = '$month.'
                        GROUP BY Category_Detail");
//카테고리에 해당하는 세부 카테고리명 별로 금액의 합 계산

$moneyDetailContentData = array();
$moneyDetailPrice = array();
//json형식으로 내보내기 위해 배열 생성

while($row = mysqli_fetch_array($sql)){
    $moneyDetailContentData = $row[0];
    $moneyDetailPrice = $row[1];
}
//각 배열에 해당하는 필드값을 넣음

print_r($sql);

$data -> label = $moneyDetailContentData;
$data -> value = $moneyDetailPrice;

//echo $_SESSION["ses_username"].'\n';

echo json_encode($data, JSON_UNESCAPED_UNICODE|JSON_NUMERIC_CHECK);

?>
```