

2020학년도

전공 종합 설계 최종보고서

제목: 2030 가계부 웹앱 AccountBook

개발자 : 12 이자룡
14 제태경
17 박명인
17 박수진

목 차

1. 요약문	3
2. 서론	3
가. 추진배경	3
나. 목표	3
다. 개발 상세 내용	4
3. 관련 앱	5
4. 설계 및 구현	6
가. 개발환경	6
나. 시스템 설계	6
다. 데이터 설계	7
라. 구성요소별 기능 구현	13
5. 개발결과	
가. 개발 일정	
나. 구현 결과	
다. 성능 분석	
라. 버그 보고	
6. 결론	
가. 결론	
나. 기대효과	

참고문헌

1. 요약문

본 프로젝트 AccountBook은 Vue.js와 php, Amazon Lightsail 서버를 이용한, 2030을 위한 가계부 웹 앱이며, Vue.js 와 php, AWS Lightsail 서버로 개발하였다. 웹 앱이기에 온라인이라면 데스크탑과 모바일, 태블릿 모두에서 동작하나, 모바일 환경에 최적화되어 있다.

2. 서론

가. 추진배경

대학생활은 자신의 생활비를 스스로 처음 관리하는 시기이다. 고등학교까지 "돈 관리"를 정규과목으로 배운 적 없기에 보통의 대학생들은 과소비를 하게 되고, 일부는 대학졸업후에도 돈을 관리하는 습관을 들이지 못한다. 돈 관리능력을 기르는 것은 자신의 수입과 지출을 아는 것에서부터 시작하는데, 가계부를 작성하는 습관을 들일 경우, 개인의 소비패턴을 파악해 지출을 줄일 수 있다. 장부나 수첩에 꼼꼼히 쓰던 때를 지나 요즘은 편하게 스마트폰 가계부를 이용해 돈을 관리하곤 한다. 기존 시장의 자산관리 프로그램이 많으나, 이는 사회인을 대상으로 하고 있어 알바 시급 또는 부모님의 용돈을 수령하는 대학생들에겐 기능면에서 불편하기도 하다. 이에 우리 팀은 시중에 나와있는 프로그램을 개선해 대학생들을 위한 자금관리 프로그램을 만들어 보기로 결정했다.

이 웹앱이 유용한지, 시장에 환영받을 수 있는지 역시 프로그램을 개발하는 입장에서 매우 중요한 문제이나, 졸업을 준비하는 컴퓨터과학과 학부생의 입장에서 그보다 더 고려되어야 할 점은 이 프로젝트로 인한 개개인의 개발실력 향상과 포트폴리오일 것이다. 가계부는 매우 좋은 실습대상이라 할 만하다. 본 프로그램은 GPS, 딥러닝, 영상인식 등의 기술은 들어가있지 않다. 그러나 프로그램의 기본인 CRUD기능을 모두 갖추고 있고, 개인의 소비와 지출을 기존 시장 데이터와 비교해 분석하거나 도식화해 사용자에게 보여주는 것은 시장에 나와있는 프로그램들에 이미 광범위하게 쓰이는 보편적 기능이다. 즉, 프로그램의 본질을 익히는 데 아주 좋은 과제라 할 수 있다.

나. 목표

- 통계와 분석을 제공하는 가계부 웹 앱을 만든다.
- 사용자의 수입, 잔액, 지출을 한 눈에 볼 수 있어야 한다.
- 달력엔 수입과 지출을 기입하고 삭제할 수 있어야 한다.
- 2030을 위한 가계부이기에 시급 입력을 포함한다.

다. 개발 상세 내용

초기 사용자 접속 시 월 수입과 고정지출 입력

- 월세, 통신비, 교통비, 적금 등의 카테고리 세분화
- 카테고리 추가 가능

소비달력

- 달력에서 수입/지출내역 작성
- 생활비 추가시 경고창

수입/지출 입력

- 오른쪽 하단 + 버튼을 눌러 실행
- 아이콘으로 구성된 카테고리를 선택

시급 계산

- 한 달 고정 수입에 자동으로 추가되어 계산됨
- 추가 수입 발생 사유로 입력할 경우 월 총 수입액에 자동 합산

지출/수익 통계

- 원형 그래프
- 각 수입지출에 해당하는 부분 클릭시 세부내역 보여줌

사용자 통계

- 초기 입력받은 성별/연령에 따른 사용자 소비현황 진단
ex) 20대 여성 평균 식비에서 2% 절약

사용자 소비유형


- 강박형, 저축형, 무관심형, 율로형 등 사용자에게 해당하는 소비유형 제시

목표 설정

- 외식 줄이기, 적금 10회 넣기, 간식 줄이기 등
- 목표치 달성을 시각화

3. 관련 앱

	소개	주요기능	우리 프로그램에 적용할 점
	근무일정이 자주 바뀌는 알바생, 직장인들을 위한 시급계산기 겸 스케줄러	<ul style="list-style-type: none"> - 시급 계산 - 근무지와 근무 일정 상세하게 조정 가능 - 지난 근무 시간과 급여 내역을 표로 시각화 	<p>매번 달라지는 월수입을 계산할 때 타 가계부어플은 해당 시급 기능을 가지고 있지 않음</p> <p>전체적인 월수입관리를 편리하게 할 수 있도록 시급관리기능 추가</p>
	카드문자 자동입력, 음성입력, PC연동, 자유로운 카테고리 설정과 다양한 통계 및 기능 지원	<ul style="list-style-type: none"> - 문자 복사 기능 - 자동이체, 자동 반복 입력 설정 - 깔끔한 UI - 일/주/월간 내역 보기 	<p>자동입력 기능은 매우 편해보이는 기능이나, 실제 가계부 사용자들에겐 수입과 지출이 과도하게 잡혀 불편함</p> <p>이에 자동입력 기능은 적용하지 않고 월간 내역 보기 기능을 개발할 계획</p>

 위플 가계부 - Wepie Money	아이폰 1등 가계부	-카드별지출통계 -원형그래프 -소비패턴 파악 가능	사용자 개인에 대한 통계는 제공하나 해당 연령대 사용자 대비 소비패턴은 파악 못함 이에 해당 기능을 개발하여 추가할 예정
--	------------	-----------------------------------	--

4. 설계 및 구현

가. 개발환경

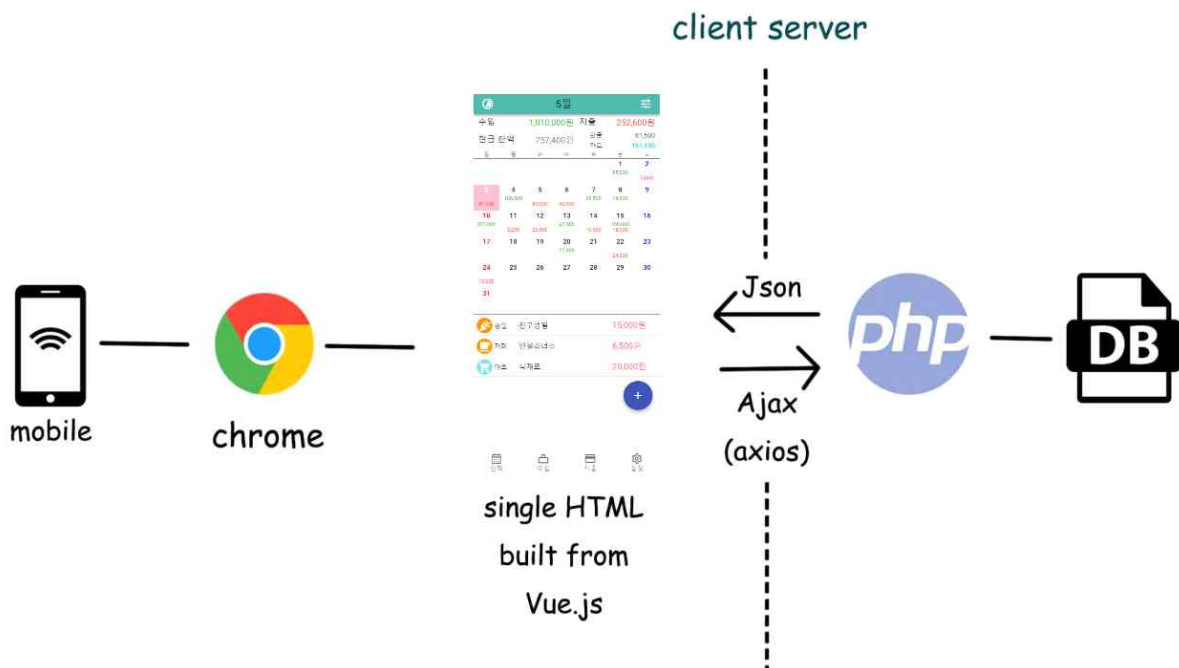
OS: Chrome

클라이언트: Vue.js 기반 Vuetify

서버: AWS Lightsail LAMP (PHP 7)

도구: Sublime Text 3, NPM

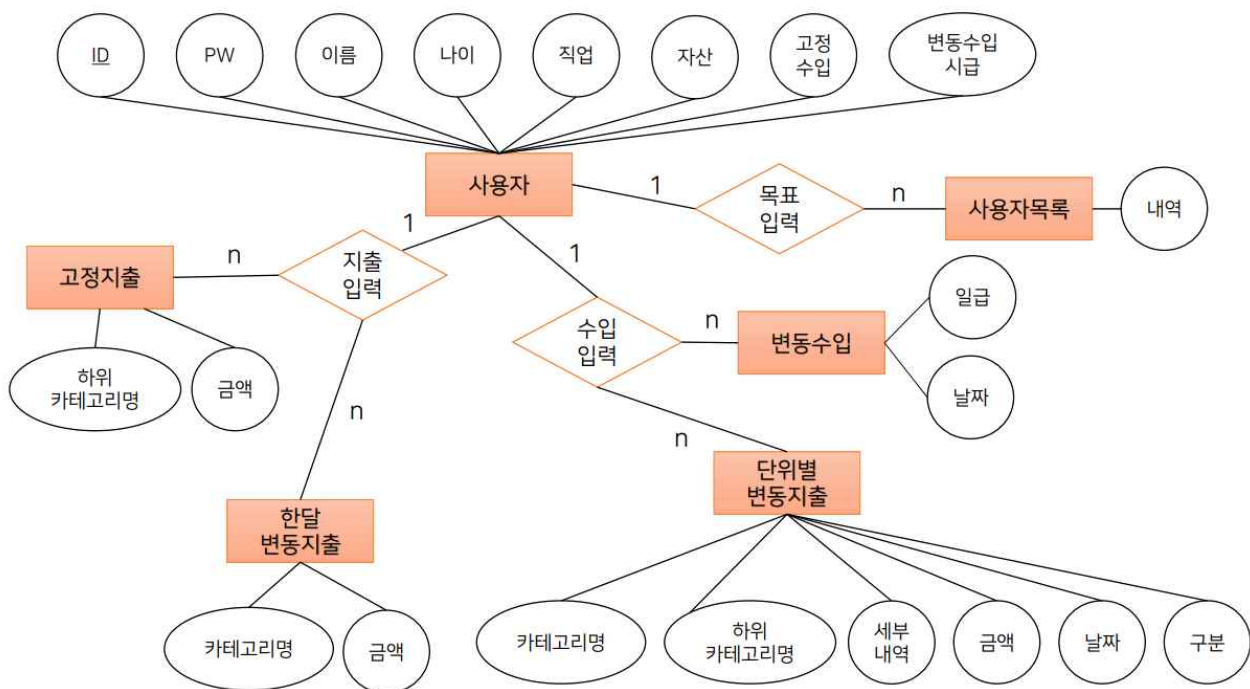
나. 시스템 설계

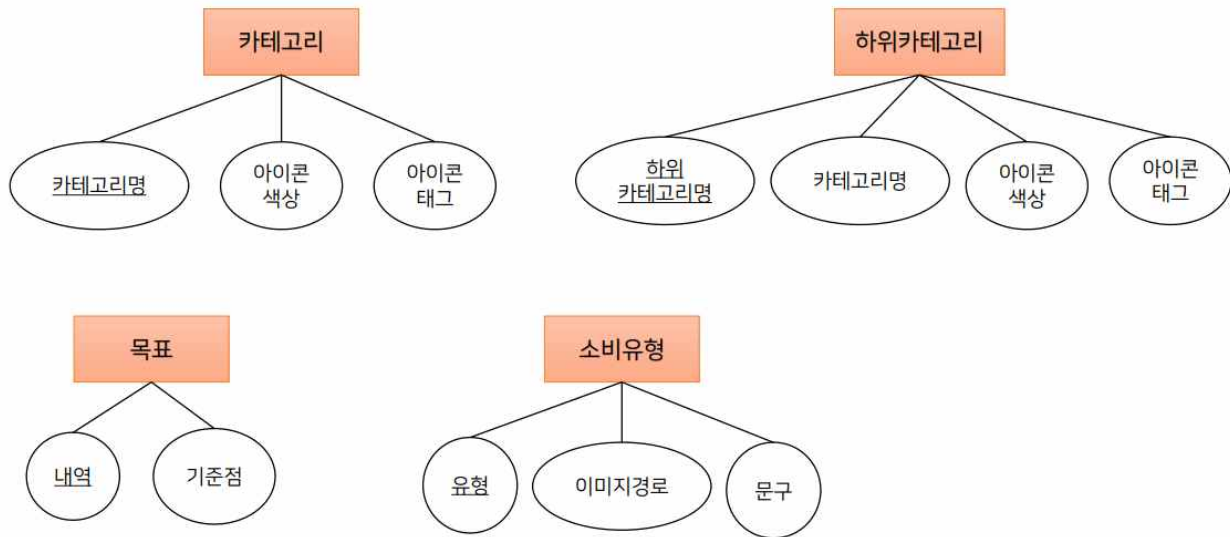


1. 모바일에서 크롬 브라우저로 AccountBook에 접속한다.
2. 클라이언트는 접속과 동시에 하나의 HTML을 서버로부터 받아온다.
 - 기존의 웹은 여러 개의 HTML을 제공하나, 클라이언트 프레임워크 Vue.js에선 하나의 HTML만을 빌드하고 그 안에서 수많은 컴포넌트들로 나눈다. 이를 SPA(Single PageApplication)방식이라 한다.
3. 클라이언트는 필요시 서버의 php파일을 실행시키고, 해당 php파일은 DB에 접근해 php명령에 따라 DB의 정보를 가져오거나, 수정하거나, 삭제해 클라이언트에 그 결과를 json양식으로 전송한다. 이를 AJAX통신이라 하며, Vue.js에선 AJAX통신을 위한 axios 라이브러리를 사용한다.
4. 통신이 성공하면, 전체 화면을 다시 그리는 것이 아니라 해당 변경부분만 클라이언트에서 그린 후 사용자에게 변경을 보여준다. 이는 클라이언트 렌더링(Client side rendering)이라고 하며, SPA방식의 큰 장점으로, 사용자에게 깜빡임 없는 자연스러운 경험을 느끼게 한다.
5. 로그아웃 시 세션을 종료한다.

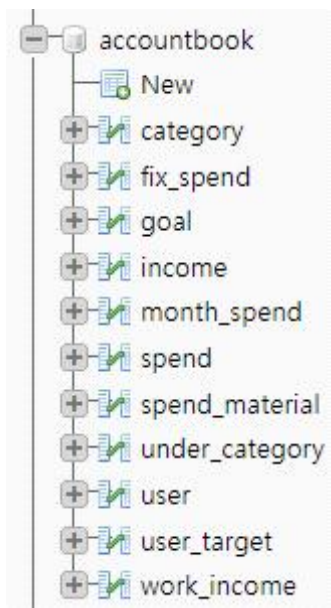
다. 데이터 설계

1) E-R 다이어그램





2) DB스키마



DB의 전체 테이블 목록

user

ID	PW	Name	Sex	Age	Cash	Fix_Income	Change_income
jr3	3333	이자룡	남자	30대	2718000	7000000	30000
mi111	1111	박명인	여자	30대	32000000	1900000	11875
sj1212	2222	박수진	여자	20대	700000	380000	9000
tg123	4444	제태경	남자	20대	1200000	500000	9500

Cash: 현재 보유 금액

Fix_Income: 유저의 고정수입

Change_income: 유저의 변동수입

category

Category_name	Icon_color	Icon_tag
경조사	#F44336	mdi-spa
교통비	#66BB6A	mdi-car-side
비상금	#0091EA	mdi-shield-key
생활비	#FFCA28	mdi-emoticon-happy
수입	#4CAF50	mdi-cash-100

큰 카테고리인 category테이블과 하위카테고리인 under_Category로 나뉜다.

Icon_tag는 Material Design Icons (MDI) 에서 제공하는 아이콘의 이름이다.

under_category

Category_Detail	Category_name	Icon_color	Icon_tag
부모님 용돈	생활비	#FF4081	mdi-heart-plus
가게 공과금	생활비	#FBC02D	mdi-flash-outline
가게 월세	생활비	#0091EA	mdi-cash-minus
자취방 월세	생활비	deep-orange accent-3	mdi-cash-marker
전세 대출금	생활비	#546E7A	mdi-bank-transfer
결혼식	경조사	#FF4081	mdi-flower-poppy
공과금	생활비	FFFFF00	mdi-flash
관리비	생활비	#448AFF	mdi-office-building
기차	교통비	#AB47BC	mdi-train
담배	생활비	#DD2C00	mdi-fire
마트	생활비	#80DEEA	mdi-cart
문화생활	생활비	#66BB6A	mdi-movie-open
미용	생활비	#FF4081	mdi-emoticon-cool

fix_spend

ID	Category_Detail	price	num
sj1212	자취방 월세	380000	1
sj1212	통신비	52000	2
sj1212	공과금	20000	3
sj1212	적금	20000	4
tg123	자취방 월세	300000	5
tg123	통신비	50000	6
tg123	공과금	50000	7
tg123	교통비	50000	8
tg123	적금	100000	9
mi111	전세대출금	280000	10
mi111	통신비	56000	11
mi111	인터넷비	18000	12
mi111	아파트 관리비	120000	13
mi111	교통비	100000	14
mi111	적금	100000	15
mi111	부모님 용돈	300000	16
mi111	담배	140000	17

고정지출을 의미한다.

month_spend

ID	Category_name	price	num
sj1212	생활비	70000	1
sj1212	경조사	20000	2
tg123	생활비	1000000	3
tg123	경조사	150000	4
tg123	비상금	300000	5
mi111	생활비	800000	6
mi111	경조사	500000	7
mi111	병원비(주기적 방문)	120000	8
jr3	생활비	500000	9
jr3	경조사	150000	10
jr3	비상금	150000	11

예산을 의미

spend

ID	Category_name	Category_Detail	Content	Use_division	Date_d	price	Division	num
sj1212	경조사	생일	생일파티참석	카드	2020-05-02	15000	-	1
sj1212	경조사	생일	치킨 기프트콘 선물	카드	2020-05-03	21000	-	2
sj1212	생활비	카페	아마스빈 버블티	현금	2020-05-05	5200	-	3
sj1212	생활비	편의점	맥주	카드	2020-05-10	9900	-	5
sj1212	생활비	식비	경아식당	현금	2020-05-12	7000	-	6
sj1212	생활비	식비	남매컵밥	카드	2020-05-13	5000	-	7
sj1212	생활비	약국	마스크	현금	2020-05-16	1500	-	8
sj1212	생활비	마트	계란	카드	2020-05-20	5500	-	9
sj1212	생활비	카페	스타벅스	카드	2020-05-23	6100	-	10
tg123	경조사	생일	친구생일	현금	2020-05-03	15000	-	12
tg123	생활비	카페	엔젤리너스	현금	2020-05-03	6500	-	13
tg123	생활비	편의점	맥주	카드	2020-05-11	3200	-	15
tg123	생활비	카페	스타벅스	카드	2020-05-14	10500	-	16
tg123	생활비	식비	맥도날드	카드	2020-05-24	10000	-	18
mi111	교통비	택시	퇴근	카드	2020-05-02	14000	-	20
mi111	생활비	마트	반찬	카드	2020-05-05	17000	-	21
mi111	생활비	유흥	술집	카드	2020-05-05	33500	-	22
mi111	교통비	택시	퇴근	카드	2020-05-07	13000	-	23

일별 지출을 의미. 달력에 들어가는 데이터다.

income

ID	Category_Detail	Content	price	Date_d	Division	num	Category_name
sj1212	용돈	부모님께 받음(엄마)	100000	2020-05-01	+	1	수입
sj1212	용돈	부모님께 받음(아빠)	200000	2020-05-10	+	2	수입
sj1212	용돈	안쓰는 것들 판매	50000	2020-05-25	+	3	수입
sj1212	용돈	친오빠의 생일선물	200000	2020-05-26	+	4	수입
tg123	주식	삼성전자 3주매도	30000	2020-05-04	+	7	수입
tg123	용돈	부모님께 받음	500000	2020-05-10	+	8	수입
mi111	주식	마스크회사 2주매도	200000	2020-05-07	+	10	수입
mi111	상금	공모전 출전 대상 수상	500000	2020-05-15	+	11	수입
jr3	투자	부동산투자	3000000	2020-05-15	+	12	수입
mi111	용돈	주운돈	10000	2020-05-06	+	20	수입
jr3	용돈	엄마용돈	300000	2020-05-15	+	23	수입
tg123	상금	노래자랑	100000	2020-05-15	+	24	수입

수입을 의미한다.

work_income

ID	Category_Detail	Content	Date_d	Time	Division	num	Category_name
tg123	일급	시급	2020-05-01	10	+	2	수입
tg123	일급	시급	2020-05-04	8	+	3	수입
tg123	일급	시급	2020-05-07	3	+	5	수입
tg123	일급	시급	2020-05-10	6	+	6	수입
jr3	일급	시급	2020-05-22	1.5	+	16	수입
tg123	일급	시급	2020-05-20	6	+	17	수입
tg123	일급	시급	2020-05-13	5	+	18	수입
tg123	일급	시급	2020-05-08	2	+	19	수입

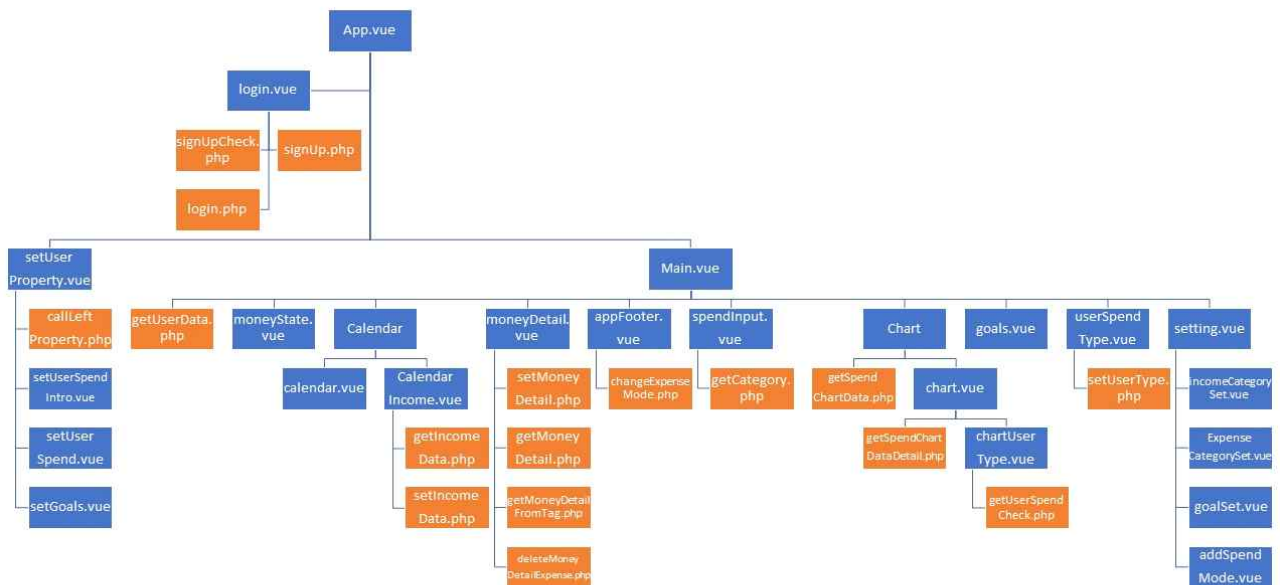
시급으로 계산되는 노동수입이다. 사용자가 설정해놓은 시급에 곱해져서 시급달력에 부착된다.

spend_material

Material	Image	Text
소비권장형	45	전체수입의 생활비 소비 비율이 10% 이하
올로형	1212	계획된 소비를 못하고 충동적으로 소비
절약형	0	소비를 줄이고 저축을 많이 함
표준형	11	사용자의 데이터를 집계한 평균을 기준으로 수입,지출이 근사함

사용자의 소비유형을 의미한다.

라. 구성요소별 기능 구현



전체 파일 계통도. 파란색: 클라이언트, 주황색: 서버

클라이언트

<pre> 1 <template> 2 <v-app> 3 <v-content> 4 <Main v-if="mainShow":data="data"/> 5 <Login 6 v-if="loginShow" 7 @login="login" 8 @signUpDataRegInLoginVue="signUpDataRegInLoginVue" 9 @setIDPWUserName="setIDPWUserName" 10 /> 11 <SignUpUserSet 12 v-if="signUpUserSetShow" 13 @goMain="goMain" 14 @saveSpendFixedList="saveSpendFixedList" 15 @propertySet="propertySet" 16 :userName="userName" 17 /> 18 </v-content> 19 </v-app> 20 </template> 21 <script> </pre>	<h3>App.vue</h3> <p>클라이언트 최상단 컴포넌트. 즉, 이 컴포넌트 아래엔 다음의 컴포넌트가 있다.</p> <ul style="list-style-type: none"> - Main - Login - SignUpUserSet <p>v-if가 참이면, 해당 부분을 '생성' @ 는 하위 컴포넌트로부터 받는 이벤트 : 는 하위 컴포넌트로 보내는 전역데이터</p> <p>참고로, v-show를 쓸 수도 있는데, 생성되는 건 같지만, 한 번 생성되고 해당 컴포넌트의 플래그를 끄게 된다면 컴포넌트가 사라지는 것이 아닌, 해당 컴포넌트의 css를 지워 안 보이게 만든다.</p>
---	---

<pre> 27 export default { 35 data:()=>({ 36 loginShow:true, 37 mainShow:false, 38 signUpUserSetShow:false, 39 40 id:"", 41 pw:"", 42 userName:"", 43 data:{}, ... 58 signUpSet:{} 59 }) </pre>	<p>data 부는 이 파일에서 사용할 전역변수를 의미하며, JavaScript 에서 접근 시 this. 로 접근한다.</p> <p>변수 중 ~Show가 붙은 것은 플래그이다. 해당 플래그가 참이면 해당 컴포넌트를 '생성'</p>
<pre> 1 <template> 2 <div> 9 <v-dialog 10 v-model="dialogForLogin" 11 persistent 12 max-width="600px" 13 > 14 <template v-slot:activator="{ on }"> 15 <div class="loginBtnContainer"> 16 <v-btn 17 color="teal lighten-2" 18 v-on="on" 19 > 20 로그인 21 </v-btn> 22 </div> 23 </template> 24 <v-col cols="12"> 25 <v-text-field 26 label="비밀번호" 27 v-model="pw" 28 :type="passwordType" 29 :rules="[pwRules.required, pwRules.min]" 30 required 31 > 32 </v-col> 33 </v-dialog> 34 </div> 35 </template> 36 37 <script> 38 39 import axios from 'axios' 40 export default { 41 data: () => ({ 42 id: "", 43 pw: "", 44 passwordType: "password", 45 userName: "", 46 age: "20대", 47 sex: "남자", 48 pwRules: { 49 required: value => !!value '패스워드가 필요합니다.', 50 min: v => v.length >= 8 '최소 8자 이상이 필요합니다.' 51 }, 52 }), 53 methods: { 54 login() { 55 axios.post('/php/login.php', { 'id': this.id, 'pw': this.pw }) 56 .then(response => { 57 if (response.data == false) { 58 this.dialogForLogin = false; 59 this.loginFail = true; 60 setTimeout(() => { 61 this.loginFail = false; 62 }, 3000); 63 return; 64 } 65 this.\$emit('login', response.data); // 상위 컴포넌트에 이벤트 전달 66 }) 67 .catch(error => { 68 if (error) 69 console.log(error); 70 }) 71 }, 72 showPassword() { 73 if (this.passwordType == "password") { 74 this.passwordType = "text"; 75 this.passwordIcon = false; 76 } else { 77 this.passwordType = "password"; 78 this.passwordIcon = true; 79 } 80 } </pre>	<h2>login.vue</h2> <p>v-dialog의 v-model값에 따라 대화창이 켜진다.</p> <p>v-text-field 부에서 중요한 것을 살펴보면, v-model은 데이터 연동이다. 사용자가 입력하면 해당 전역변수가 바뀐다.</p> <p>:property="variable" 형태는 해당 기능이 전역변수에 쓰여져 있는 형태로 쓰임을 의미한다. :rules 를 예로 들면, pwRules 전역변수에 접근해 해당 정보를 적용하는 것이다.</p> <p>import 부에 axios 는 요즘 가장 많이 사용되는 JavaScript ajax 통신 라이브러리이다. get과 post 방식으로 데이터 통신을 실행한다. 좀 더 자세히 살펴보면, login() 에서 axios가 쓰였는데, 실행할 php파일 경로를 입력하고 변수를 보내는 경우 그 뒤에 실어서 보낸다.</p> <p>*변수는 괄호 안으로 표시</p> <h3>login.php</h3> <ol style="list-style-type: none"> 클라이언트에서 사용자의 아이디(id. String)와 비밀번호(pw. String)를 넘겨준다. 해당 아이디가 존재하지 않거나, 비밀번호가 틀리면 -> return FALSE (boolean) 로그인 성공 시, 다음 형태의 JSON 양식을 리턴한다. <pre> userData: { month: 4, income: 2000000, balance: 581300, expense: 1418700, expenseTypeCash: 1418700, expenseTypeCard: 0 }, monthData: { thisYear: 2020, thisMonth: 4, spendContent: { 1: [2000000, "-"], 3: [7900, "-"], 5: [400000, "-"], 7: [56800, "-"], 8: [6000, "-"], 9: [25000, "-"], 10: [500000, "-"], 12: [36700, "-"], 13: [259200, "-"], 14: [10500, "-"], 15: [53600, "-"], 16: [10200, "-"], 17: [5000, "-"], 20: [12800, "-"], 21: [35600, "-"] } }, </pre> <p>여기서, php의 DateTime 클래스를 이용해 '올해(thisYear, int)'와 '이번달(thisMonth, int)'에 해당하는 정보를 최초로 보냅니다.</p> <h3>서버코드 지시서 예시</h3> <p>클라이언트 개발자는 서버 코드가 필요할 때 다음 형태의 문서를 만들어 서버개발자에게 서버파일 개발을 요청한다. 서버 개발자는 해당 문서를 파악하여, DB에 데이터를 생성, 읽기, 갱신, 삭제한다.</p> <p>여기서 this.\$emit('eventName', variable) 형태는, 상위컴포넌트에 이벤트를 전달하는 것을 의미한다. 주로 상위컴포넌트의 데이터를 변경할 일이 있을 때 사용된다. 상위컴포넌트 App.vue에선 다음을 실행하는데,</p> <pre> login(data) { this.loginShow = false; this.mainShow = true; this.data = data; } </pre> <p>살펴보면, 상황에 따라 플래그를 바꾸고, 데이터를 갱신하는 것을 볼 수 있다.</p> <p>showPassword() 는 버튼을 클릭했을 때, 비밀번호가 **** 식으로 보일지 아니면 단순히 텍스트로 보일지 바꾸는 기능이다.</p>


<pre> 81 signUp() { 82 if (this.id == "" this.pw == "" this.userName == "") 83 return; 84 axios.post('/php/signupIdCheck.php', { "id": this.id }) 85 .then(response => { 86 if (response.data == false) { 87 this.signUpFail = true; 88 this.dialogForSignUp = false; 89 setTimeout(() => { 90 this.signUpFail = false; 91 }, 3000); 92 return; 93 } else { 94 this.\$emit('setDPWUserName', { 95 'id': this.id, 96 'pw': this.pw, 97 'userName': this.userName 98 }); 99 this.\$emit('signUpDataRegInLoginVue', [100 this.id, 101 this.pw, 102 this.userName, 103 this.age, 104 this.sex 105]); 106 } 107 }) 108 .catch(error => { 109 if (error) 110 console.log("실패!"); 111 }) 112 } 113 } 114 } 115 </script> </pre>	<p>signUp() 부에서, 사용자가 id나 pw나 사용자이름을 입력하지 않으면, 이 함수는 실행되지 않는다. signUpCheck.php는 사용자의 id가 이미 존재하는지 확인하는 서버코드이다. 만약 이미 존재한다면 fail을 반환받고, 회원가입 실패 알림을 띄운다. 이후 \$emit 을 통해 상위컴포넌트로 세팅을 위한 데이터를 보내고, 상위컴포넌트인 App.vue에선 다음과 같이 데이터를 갱신한다.</p> <pre> signUpDataRegInLoginVue(data) { this.signUpSet.id = data[0]; this.signUpSet.pw = data[1]; this.signUpSet.name = data[2]; this.signUpSet.age = data[3]; this.signUpSet.sex = data[4]; this.loginShow = false; this.signUpUserSetShow = true; }, setDPWUserName(data) { this.id = data.id; this.pw = data.pw; this.userName = data.userName; } </pre> <p>이 데이터는 App.vue에서 저장하고 있다가, 필요한 모든 데이터가 모이면 signUp.php를 실행하여 보낼 예정이다.</p>

5. 개발결과

가. 개발 일정


	~ 3.27	3.30 ~ 4.10
이자룡	- Vue.js 세팅 및 아웃라인 작업	- 초기 사용자 접속화면 개발
제태경	- php, JSON 복습	- AWS 서버세팅
박수진	- php, JSON 복습 - 대략적인 디자인과 UI 설계	
박명인	- 대략적인 디자인과 UI 설계	- SQL학습
	4.13 - 4.24	4.27 ~ 5.8
이자룡	- 수입/지출 카테고리 제작	- 달력 제작
제태경	- 로그인 코드 작성	- 수집된 소비패턴데이터에 기반한 메소드 작성
박수진	- 소비패턴 데이터 수집	- 클라이언트와 AJAX 통신테스트
박명인	- phpMyAdmin 학습	- 더미 데이터 DB적용
	5.11 ~ 5.22	5.25 - 6.5
이자룡	- 시급, 사용자 소비유형, 목표 설정 개발	- 통계 개발
제태경	- 각 클라이언트 기능별 서버부 코드 작성	- 통계 서버부 코드 작성
박수진		
박명인	- SQL 작성	- 배경 및 아이템 디자인
	6.8 - 6.19	
이자룡	- 최종 테스트	
제태경		
박수진		
박명인		

나. 구현 결과



어서오세요!
AccountBook과 함께 합리적인
소비습관을 길러볼까요?

로그인



로그인

ID

Password

가입하기 로그인

회원가입

ID

비밀번호

이름

나이
20대

성별
남자

청단기 회원가입

메인

로그인

회원가입

현재 총 자산

고정수입

고정지출

월세

통신비

적금

+

NEXT

CS님의 남은 자산은
870,000원입니다!

남은 금액으로 한달 생활비를 정하세요!

추천하는 생활비 비율은
남은 금액에서 60%입니다.

다음

변동지출을 입력해주세요

생활비

경조사

비상금

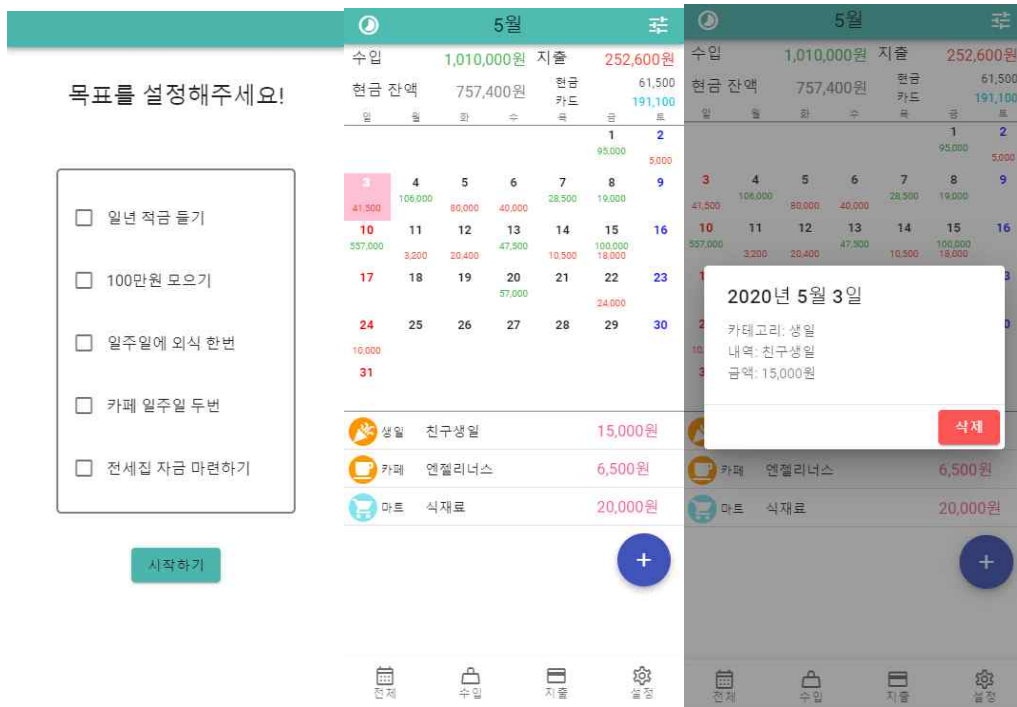
+

NEXT

자산/고정지출 입력

남은 자산 계산

변동지출 입력

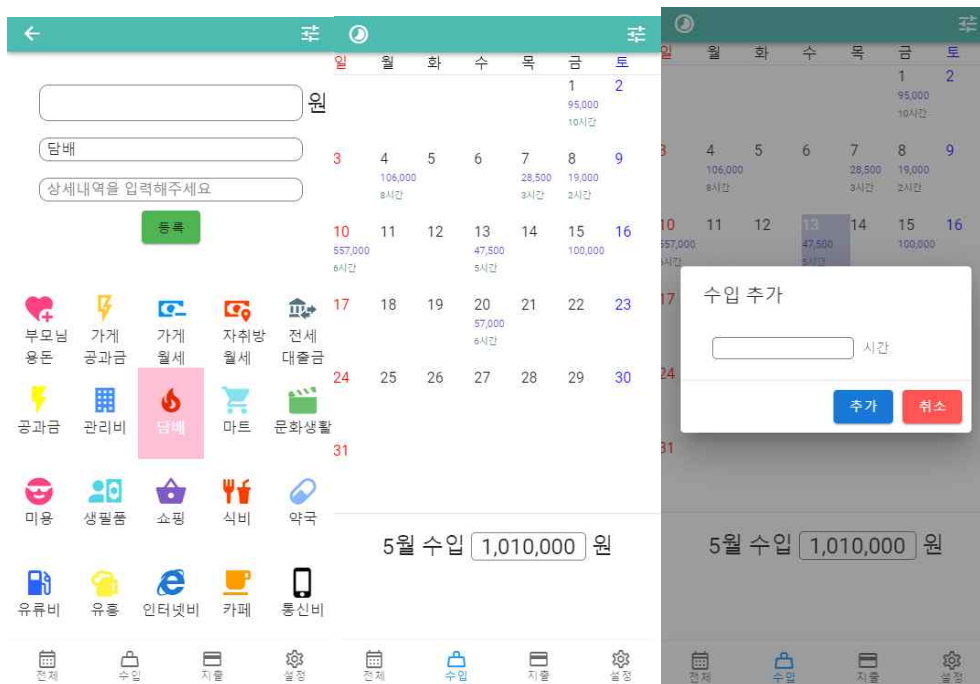


목표설정

메인화면

항목 클릭 시 세부내역

달력의 날짜를 클릭했을 때 날짜별 세부내역이 출력된다. 세부내역을 클릭하면 상세내역 대화창이 뜨고 삭제가 가능하다. 삭제를 원치 않을 시 화면 밖을 클릭하면 된다.



수입/지출 추가화면

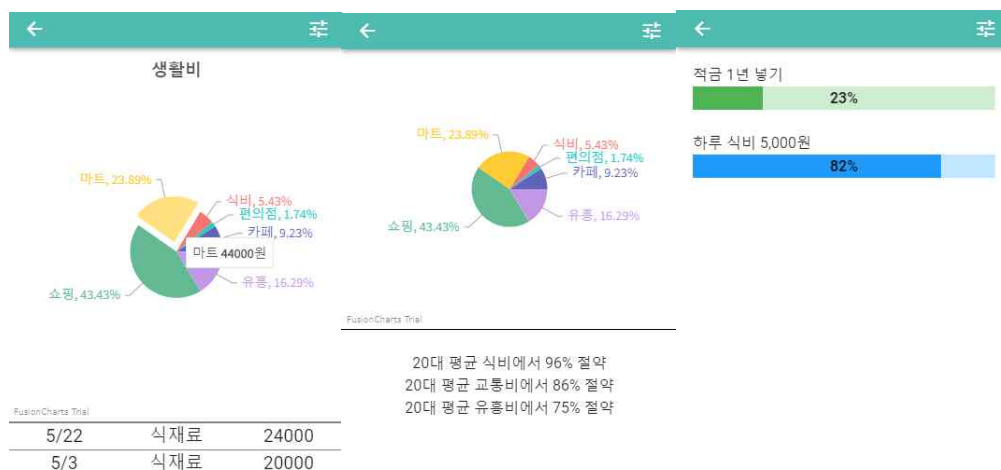
시급달력

시급 추가



달력 모드 변경

전체 달력의 보기를 변경하는 기능이다. 생활비를 누를 시 생활비만 달력에 표시되며, 경조사, 비상금, 교통비도 마찬가지로이다.



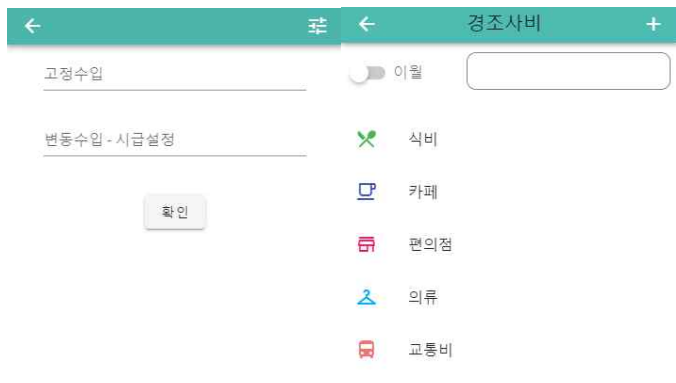
생활비 통계

사용자 소비분석

목표

통계의 각 항목을 클릭하면 상세내역이 출력된다. 소비분석에선 사용자의 소비 방식을 분석해 해당 나이 평균에서 몇 % 절약했는지 알려준다.

소비유형 추가할 것



고정수입/지출 변경 지출 카테고리 변경

다. 성능 분석 (타 시스템과 비교 분석)

관련 앱 항목에서 다른 프로그램과 비교분석을 해 우리 앱이 구현할 부분을 정했는데, 모두 완료되었으므로 목표한대로 개발되었다고 할 수 있다. 다만, 다른 앱들은 모두 네이티브 앱으로 개발되었으나 우리 앱은 크롬에서 작동하므로, 네이티브 앱과 웹 앱의 차이가 곧 성능 차이라고 할 수 있을 것이다.

특성	네이티브 앱	웹 앱
개발 언어	네이티브 전용	웹 전용
디바이스 고유기능 액세스	높음	낮음
유려한 UI	높음	중간
업그레이드 유연성	낮음 (항상 앱스토어)	높음 (웹에 바로 적용)
구현 난이도	높음	중간

또한 우리 프로그램은 전통적 웹과 다르게 클라이언트 사이드 렌더링 방식을 취하는데, 이는 화면을 그리는 행동(렌더링)을 서버가 아니라 클라이언트가 담당함을 의미하며, 접속과 동시에 필요한 정보들을 다운받기 때문에 초기 실행속도가 상대적으로 느리다고 알려져있다. 그러나

우리 웹 앱의 용량이 큰 편이 아니기에 대부분의 모바일 기기에선 무리 없이 잘 작동할 것으로 예상된다.

추가적으로 AWS서버는 1GB RAM 에 40GB SSD 를 사용중이며, 이는 교내 소프트웨어 전시회에 우리 작품을 출품하고 충분한 시연행사를 거처도 충분히 버티는 성능이라 할 수 있다.

라. 버그 보고

6. 결론

가. 결론

팀원 전체가 우리의 프로그램이 우수하다는 것에 동의하진 않는다. 정말 우수하다 싶은 세 팀은 모두 임베디드 개발을 진행하여 중간보고서를 내었고, 유니티를 이용한 게임 개발을 하는 팀들도 꽤 많았다. 우리 프로그램은 웹을 이용한 가계부인데, 이런 종류의 앱은 플레이스토어에 검색해서 우리의 프로그램보다 훨씬 좋은 걸 그냥 쓰면 된다.

그러나 앞에서 이미 밝혔듯이, 우리 프로그램은 GPS나 딥러닝, 라즈베리 파이 등의 멋져보이는 고급 기술은 없지만, 프로그램의 기본에 충실했다고 자부할 수 있다. 가계부는 CRUD를 제대로 연습하기 아주 좋은 예제였고, 프로그램의 UI도 깔끔하다. 별 것 아니어보이는 이 프로그램을 만들기 위해 클라이언트 24개, 서버 17개의 파일과 11개의 DB 테이블이 쓰였고, 약 4,400줄의 코드를 작성했다. 이는 작년의 소프트웨어및프로젝트보다 발전한 수치이며, 웹 개발 포트폴리오로 적당한 수준이라 생각된다.

결론적으로, 완벽하진 못해도 팀원 모두가 자신의 맡은 업무에 최선을 다했다고 자신있게 말할 수 있는 졸업 프로젝트였다.

나. 기대효과

본 프로그램을 통해 다음의 기능을 원하는 사용자들의 수요를 기대할 수 있다.

- 대부분의 가계부에 없는 시급 기능이 포함되어 편리한 시급 계산
- 통계 기능을 통한 개인의 소비 점검
- 매번 고정지출 적을 필요 없이 자동계산
- 자동 기능에 의해 수입과 지출이 과하게 잡히는 것을 방지

무엇보다 중요한 것으로, 웹 개발자 취업 시 제시할 수 있는 기본에 충실한 포트폴리오이다.

참고문헌

- 1934 소비성향 및 지출분석 Report – 2017.10 대학내일 20대 연구소
- 가구특성별 비목별 소비지출 – 2019.12.17 통계청 가계금융복지조사
- 경제주평 소비 구조의 특징과 과제 – 2017.06.09 현대경제연구원
- Do it! Vue.js 입문(장기효 저, 이지스퍼블리싱) - 클라이언트에 사용된 Vue.js에 관한 초급 입문서
- Vuetify 공식 문서 (<https://vuetifyjs.com/en/>) - Vue.js UI Framework 인 Vuetify 의 공식 문서
- 생활코딩 HTML, CSS, Javascript 수업 - 클라이언트 개발 참고용 - 생활코딩 php 강의(<https://opentutorials.org/index.php/course/62>) - php문서 작성을 위해 기본 php문법에 대해 학습
- 생활코딩 phpMyAdmin(<https://opentutorials.org/course/195/1469>)- 데이터베이스관리를 위한 phpMyAdmin에 대해 학습
- 생활코딩 php & mysql연동(<https://opentutorials.org/course/3167/19586>) - PHP mysqli_query() 함수(https://www.w3schools.com/php/func_mysqli_query.asp)- php문서 작성을 위한 php와 mysql연동, select를 사용하기 위해 학습
- 넘버원 PHP : 기초문법 주무르기 / 비제이 퍼블릭 (학교도서관 대여) - php 기초문법
- 초보를 위한 PHP 200제 / 정보문화사 (ebook) - php예제

비동기 통신 관련 예제

- 오픈카카오톡 PHP스터디방 (<https://open.kakao.com/o/gSdv7TM>) - PHP에서 JSON 사용하기 (<https://mytory.net/archives/40>) - vue와의 소통시 json decode를 먼저 해주어야 하는 이유

함수 및 기능

- 날짜/시간 함수 정리(<https://88240.tistory.com/110>) - 중복 제거함수 array_diff() 사용법(https://www.w3schools.com/php/func_array_diff.asp) - 세션 사용방법 (<https://runtoyourdream.tistory.com/267>) - 난수 사용방법 (<https://m.blog.naver.com/PostList.nhn?blogId=diceworld>) - 포토샵cs2 백재경 조교선생님 페이지 컴퓨터와 멀티미디어 (<http://class.gnu.kr/~torl/>)