



### 3조. 각양각색 🎨

Team					진행 상황(추가해주세요)
Aa 이름	📌 태그	☰ MBTI	☰ 블로그 주소	☰ Github주소	▼ 24.06.07.Fri 한 마 디! <input checked="" type="checkbox"/> 조구성 <input checked="" type="checkbox"/> 참조 사이트 조사  1등 각! <b>제목 없음</b>
조현아	팀원	INFJ	<a href="https://velog.io/@hyunaday/posts">https://velog.io/@hyunaday/posts</a>	<a href="https://github.com/hyunaday2/tab=repositories">https://github.com/hyunaday2/tab=repositories</a>	열 심 하 해 보 겠 습 니 당! ▼ 24.06.10.Mon <input checked="" type="checkbox"/> 바 설계 <input checked="" type="checkbox"/> json 구조 설정 <input checked="" type="checkbox"/> 깃연등
정예슬	팀원	ESTP		<a href="https://github.com/yeseuljung1">https://github.com/yeseuljung1</a>	내 열 예 충 샅 하 자! ▼ 24.06.11.Tue ▼ 24.06.12.Wed <b>제목 없음</b>
김세진	팀원	ISFP		<a href="https://github.com/KIMISM">https://github.com/KIMISM</a>	화 이 텅! ▼ 24.06.13 Thurs <input checked="" type="checkbox"/> 요구 기능 오류 수정 <input checked="" type="checkbox"/> CSS 오류 수정
문지은	깃 관리자	ESTJ	<a href="https://data-science-archive.tistory.com/">https://data-science-archive.tistory.com/</a>	<a href="https://github.com/jieun-moon?tab=repositories">https://github.com/jieun-moon?tab=repositories</a>	
차교진	팀장	ENFJ		<a href="https://github.com/dadadacole">https://github.com/dadadacole</a>	더 이 용 하 기 편 리 한 가 계

Aa 이름	☺ 태그	≡ MBTI	≡ 블로그 주소	≡ Github주소	≡ 한 마 디!
					부 를 만 들 자

## 공유 자료

### 요구 기능 명세서

no	구분	기능명	설명	우선순위	기간	담당자
1	Home.vue	달력 대쉬보드	<ul style="list-style-type: none"> <li>• 사용자가 직관적으로 일자별, 월별 수입 지출 내역 확인할 수 있는 달력 보드(조현아)</li> <li>• 달력 대쉬보드 개선(정예슬, 문지은)</li> <li>• 일자 클릭 &gt; input 가능(정예슬)</li> <li>• 총 수입, 총 지출, 총 잔액(문지은)</li> </ul>	1	06/11 ~ 06/14	조현아, 정예슬 문지은
2	CardRecommendation.vue	지출에 따른 카드 추천 (CardRecommend)	<ul style="list-style-type: none"> <li>• V-IF문을 활용한 지출에 따른 카드 추천</li> </ul>	1	06/11 ~ 06/14	차규진
3	InOut.vue ExpenseForm.vue	수입, 지출 내역 입력 및 관리	<ul style="list-style-type: none"> <li>• 수입, 지출 카테고리 클릭 &gt; 각 카테고리에 따른 세부 카테고리에 클릭 &gt; 드롭다운 형식으로 날짜 선택, 금액, 사용처, 메모 input 입력</li> <li>• table로 수입, 지출 내역 확인 가능</li> <li>• 각 행별 편집 및 삭제 가능</li> </ul>	1	06/11 ~ 06/14	문지은
4	Join.vue	회원가입	<ul style="list-style-type: none"> <li>• 회원 가입 페이지(김규진)</li> <li>• css 및 태그 재배치(조현아)</li> </ul>	1	06/11 ~ 06/14	차규진, 조현아
5	Challenge.vue	무지출 챌린지	<ul style="list-style-type: none"> <li>• db.json의 date 데이터의 expense값이 Null값으로 들어올 때, 도장이 찍히는 기능</li> </ul>	1	06/11 ~ 06/14	조현아
6	Header.vue	목록 (반응형웹-햄버거바)	<ul style="list-style-type: none"> <li>• 목록창 구성 : 수입/지출/예산/환경설정 (기존의 링크로 이동)</li> <li>• 하단에 '개인정보'</li> </ul>	1	06/11 ~ 06/14	차규진

7	style.css	CSS	• 카드 추천 CSS 담당	1	06/11 ~ 06/14	김세진
8		페이지 라우팅	• 페이지 간 전환 라우팅 기능	1	06/11 ~ 06/14	차규진
9	db.json	데이터 베이스	• DB 작성(김세진) • DB 구조 개선 (조현아)	1	06/12	김세진, 조현아
10	ListView.vue	로그인	• 로그인 페이지 구현 및 css 수정 • 아이디/비밀번호 체크 기능 활성화 • 로그인 클릭 > Home.vue로 이동 • 회원가입 클릭 > Join.vue로 이동	1	06/13	조현아
11	CardRecommendation.vue	지출에 따른 카드 추천 (CardRecommend)	• 지출에 따른 카드 이미지 링크 연결 및 추천	1	06/11 ~ 06/14	김세진
12	UI, FIGMA	스토리 보드 및 figma 작성	가계부 작성을 위한 초안 작성	1	06/11 ~ 06/14	정예슬
13	Loading.vue	웹 페이지	접속 시에 대기 시간을 보여주는 화면 표시		06/11 ~ 06/14	차규진

## 레퍼런스 사이트

사이트 주소	장점	단점	개선 방향
<a href="https://github.com/minsu-zip/web-moneybook-04?tab=readme-ov-file">https://github.com/minsu-zip/web-moneybook-04?tab=readme-ov-file</a>	- 입금과 출금 기능 - 직관적으로 수입 및 지출내역 확인 가능 - 카테고리별 분류		
<a href="https://ksumin-dev.tistory.com/127">https://ksumin-dev.tistory.com/127</a>	-다양한 디자인		
<a href="https://crispypotato.tistory.com/173">https://crispypotato.tistory.com/173</a>			
<a href="https://mysetting.io/posts/detail/c9b056be-cb28-4084-94bf-799ff0c2b788">https://mysetting.io/posts/detail/c9b056be-cb28-4084-94bf-799ff0c2b788</a>			
<a href="https://shanepark.tistory.com/284">https://shanepark.tistory.com/284</a>	깃허브 사용		
<a href="https://card.kbcard.com/CMN/DVIEW/HOAMCXPRIZZC0002">https://card.kbcard.com/CMN/DVIEW/HOAMCXPRIZZC0002</a>	국민은행 카드 추천		
<a href="https://velog.io/@saemmmm/React-moment.js로-달력-구현하기">https://velog.io/@saemmmm/React-moment.js로-달력-구현하기</a>	달력 구현		
<a href="https://xionwcfm.tistory.com/420">https://xionwcfm.tistory.com/420</a>	달력 구현		

## 기본 json 파일

db.json 20개 추가

X

X

X


## 스토리보드

[https://www.figma.com/embed?embed\\_host=notion&url=https%3A%2F%2Fwww.figma.com%2Fdesign%2FQQjG1oOIZ28SI8UBSlweL%2FUntitled%3Fnode-id%3D0-1](https://www.figma.com/embed?embed_host=notion&url=https%3A%2F%2Fwww.figma.com%2Fdesign%2FQQjG1oOIZ28SI8UBSlweL%2FUntitled%3Fnode-id%3D0-1)


## 소스코드 (팀 레포지토리 주소)

GitHub - dadadacole/GROUP-PROJECT at main2

Contribute to dadadacole/GROUP-PROJECT development by creating an account on GitHub.

 <https://github.com/dadadacole/GROUP-PROJECT/tree/main2>

**dadadacole/  
GROUP-PROJECT**



0 Contributors 0 Issues 0 Stars 0 Forks

## 발표 자료

[안녕하세요.hwp](#)

## 어려움과 극복 과정

- 차근차근 : option API를 활용해서 외부 데이터를 불러오는게 많이 어려웠다 하지만 팀원들이랑 잘 의논하고 여러 시행착오를 겪어서 db.json과 본 페이지 간의 reactive 속성을 잘 반영한 것 같다. 그리고 db.json 서버의 글을 todolist 기능을 활용해서 가계부에 적용하는 과정이 재밌었고 흥미로웠다 또한 카드 추천 기능이 해당 월에 표시 되게끔 함수를 짜보게 되어서 재밌고 흥미로웠다
- 문제는:
  - 깃허브: 깃허브 연동에 시간을 많이 낭비. > 개인 깃허브로 파일 주고 받음 > 슬랙으로 압축파일 공유
  - 수입/지출: 첫 피그마 기획 토대로 접근이 어려워 13과 todolist list app 과제 활용 > 수입과 지출 페이지를 나눔(db.json 연동 로직이 잘 되지 않음 > 수입과 지출 페이지를 한 form에서 할 수 있도록 수정
  - 캘린더, InOut: Home 화면의 캘린더와 InOut에 들어간 데이터를 연동하는데 문제가 있었음 > 초기에는 페이지끼리 연동하고자 하려고 함 > 각각 페이지를 db.json에 연동되도록 하여 해결
- 캘린더:
  - axios를 사용하여 `http://localhost:3000/users` 및 `http://localhost:3000/expenditures` 엔드포인트에서 데이터를 가져오고, 새로운 거래를 추가하는 요청 전송,
  - `fetchTransactions` 메서드는 초기 데이터 로딩을 담당하며, `addTransaction` 메서드는 새로운 거래를 추가
- ExpenseForm:
  - axios를 사용하여 `.mounted()`에서 데이터 가져오기, `handleSubmit()`에서 데이터 추가 및 수정(async, try, catch),

deleteTransaction()에서 데이터 삭제(async, try, catch)

- InOut:

import 를 통해 ExpenseForm.vue 컴포넌트를 가져옴. 이벤트 핸들러 메서드를 통해 거래 추가, 거래 삭제, 거래 업데이트, 거래 데이터 설정(편집)을 처리하는 역할을 함

- 조현아 : 새로 만드는 게 기존 사항을 변경하는 것보다 쉽다는 걸 알게 되었고, 협업 시 소통의 중요성과 주석의 필요성을 느꼈습니다.
  - 에러 발생 > 검사(F12) 창을 활용해 에러를 찾고 수정하는 방법에 대해 알게 되었습니다.
  - db.json : 객체명 다름 > 협업 시 미리 객체명과 변수/함수명을 정하고 작업을 들어가야 한다는 걸 배웠습니다. > 이후 각자 작업을 통합하며 최종 db.json을 적용하였습니다.
  - Challenge : 무지출 챌린지에 활용 할 db.json 데이터를 받아오는 API 호출 방식, API 주소를 절대경로 (<http://localhost:3000/expenditures>) 로 연결하기
  - ListView : 값 없이도 로그인 버튼 클릭 시 화면 이동 > 로그인 시, 아이디&비밀번호 체크 기능을 활성화했습니다.
  - router : `router-link` 를 활용해 버튼 클릭 시, 화면 전환이 되도록 구현했습니다.
- 김세진
  - 1.부트스트랩 카드 적용했지만 페이지 화면에 맞지 않아 애를 먹었지만 해결방안을 찾아내 마무리 했다.
  - 2.버튼 위치들이 제각각이어서 어떻게 맞춰야할까 고민하다가 버튼과 카드 바디를 따로 나누어서 버튼 컨테이너에 절대적 위치 설정, 아래쪽에는 여백, 가운데 정렬을 위해 왼쪽 여백을 설정, 가운데 정렬하여 버튼 위치를 동일하게 정렬했다
  - 3.깃과 소스트리의 중요성을 깨닫게 되었다
- 정예슬: 달력 안에 수입과 지출을 나타내고 그걸 사용자가 작성한 내용과 연동 시키는 과정에서 힘들었습니다 이 과정에서 발생한 어려움을 극복하기 위해 구글 검색을 통해 Vue.js 관련 자료를 찾아 읽고 gpt를 활용하며 해결하는 시간을 가졌습니다.또한 다른 분들이 만든 페이지에서 작은 부분을 수정해야 할 때 어느 부분을 수정해야 하는지 찾기 힘들었는데 개발자 도구를 활용해서 찾는 방법을 익혔습니다