

View my Startup

스타트업 정보 확인 및 모의 투자 서비스

00

목차

01

프로젝트 개요

- 개발 환경 및 사용 스택
- 프로젝트 구조
- 활용 방안 및 기대 효과

02

팀 구성 및 역할, 협업 방식 소개

- 팀원 소개
- R&R
- 이슈 트래킹 및 일정 관리
- 코드 리뷰 및 개발 컨벤션

03

프로젝트 수행 절차 및 방법

- 프로젝트 절차
- 주요 자료
- 트러블 슈팅

04

프로젝트 수행 결과

- 시연 영상 또는 실제 시연
- 서비스 배포 링크 첨부

05

팀 자체 평가 의견

- 아쉬운 점
- 잘한 점
- 느낀 점

06

Q&A

01

프로젝트 내용

가상 투자

모의 투자 시뮬레이션을 통해
원하는 기업에 투자하세요!

스타트업 조회

다양한 스타트업들과
해당 스타트업의 상세 정보를
조회할 수 있습니다!

기업 비교

여러가지 기준으로 기업들 간의
비교 분석이 가능합니다!

View My Startup 서비스

개인 투자자들의 스타트업에 대한 관심이 증가하고 있지만, 스타트업 정보의 접근성은 여전히 부족합니다.

이러한 시장 가능성을 바탕으로, 개인 투자자들이 스타트업 정보를 제공받고, 누적 투자 금액, 매출액, 고용 인원 등을 기준으로 스타트업을 비교하여 투자 결과를 확인할 수 있는 **모의 투자 서비스**입니다.

우리 서비스는 다른 서비스와의 차별점으로 번거롭게 **로그인 할 필요없이** 임의의 닉네임과 비밀번호만을 이용하여

모의 투자 시뮬레이션을 진행할 수 있으며, **비교 분석**을 통해 성과를 기준으로 평가하고, **최적의 투자 대상**을 결정할 수 있습니다.

01

개발환경 및 사용 스택

<Front-End>

JavaScript

JavaScript는 동적인 기능을
쉽게 구현할 수 있기 때문에
다양한 동작을 처리하기에 용이

React는 컴포넌트 기반 구조로,
재사용 가능한 UI 요소를 쉽게 만들고,
가상 DOM을 사용하여 UI의 변경 사항을
효율적으로 관리 가능

React

Express.js

PostgreSQL

Express는 Node.js 기반이기 때문에,
JavaScript로 개발 가능하여
전체 개발 환경을 통일할 수 있어서 선정

PostgreSQL은 데이터베이스의
확장성이뛰어나고
폭넓게 사용 가능 하기 때문에 선정

<Back-End>

<협업 및 배포 플랫폼>

- Git & GitHub
- Render
- Notion
- Netlify

01

프로젝트 파일 구조

Front-End 파일 구조

season2-View-My-Startup-3team-FE

```
|
|
| /public                                # 기업 로고
|   |
|   | /images
|   | _redirects
|   | favicon.png
|   | index.html
| /src
|   |
|   | /components
|   | /pages
|   | /api
|   | /assets
|   | /contexts
|   | /hooks
|   | /styles
|   | /utils
|   | Main.js                            # 앱 엔트리 포인트
|   | index.js                          # ReactDOM 렌더링
|   | README.md
```

Back-End 파일 구조

season2-View-My-Startup-3team-BE

```
|
| /controllers                          # 각종 비즈니스 로직을 처리하는 컨트롤러 폴더
| /middlewares                          # 미들웨어 관련 파일들
| /routes                              # 라우트 설정 폴더
| /prisma                              # 데이터베이스 관련 폴더
| /structs                             # 데이터 구조 및 검증을 위한 superstruct 정의 폴더
| /http                                # HTTP API 테스트용 폴더
| .env                                 # 환경 변수 파일
| app.js                               # Express 서버의 메인 파일
| package.json                         # 프로젝트 메타데이터 및 의존성 정보
| README.md                            # 프로젝트 설명서
```

01

활용 방안 및 기대 효과

실제 투자 전, 다양한 시나리오를 통해
투자 결과를 예측할 수 있는
모의 투자 기능을 제공하여
투자자들이 위험을 평가하고
더 나은 의사 결정을 할 수 있도록 합니다.

투자자들이 **충분한 정보**에 기반하여,
스타트업에 대한 투자 결정을 내릴 수 있기
때문에 **투자 성과를 향상**시킬 수 있습니다.

스타트업에 대한 정보를
쉽게 접근할 수 있도록 하여
시장의 투명성을 높이고,
투자자와 스타트업 간의
신뢰를 구축합니다.

02

팀원 소개



이도엽

- GitHub 프로젝트 관리 (칸반 보드, 로드맵)
- 공용 레포 생성 및 관리
- 회의록 작성 및 업로드
- 프로젝트 개요 정리
- 프로젝트 과정 도식화

김은효

- 일정관리

서유림

- 팀 노션 문서 관리
- Render 배포 관리
- 유저 플로우, 페이지 플로우, ERD 작성
- 발표자료 준비

서지우

- README 업데이트

소재희

- 프로젝트 학습 내용 아카이빙 담당
- Netlify 배포 관리
- 팀 질문 관리
- API 명세 작성

02

R&R

이도엽

- 프론트엔드 기초 설정
- 비교할 기업 선택 페이지 UI & API
- 기업 비교 결과 페이지 UI & API
- 내 기업 순위 확인 UI
- 비교하기 모달 UI

서지우

- 비교 현황 조회 페이지 UI & API

김은효

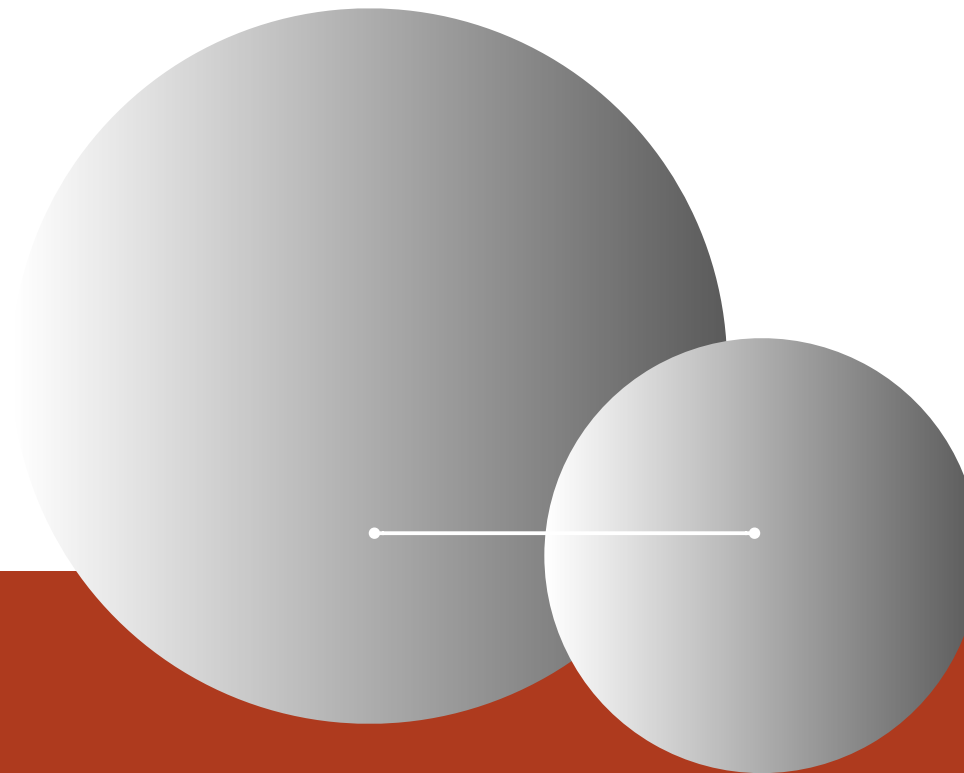
- 기업 상세 페이지 UI

소재희

- 백엔드 기초 설정
- 전체 기업 조회 페이지 UI & API
- 내 기업 순위 확인 API
- Pagination 공통 컴포넌트
- DB data 관리
- Footer UI

서유림

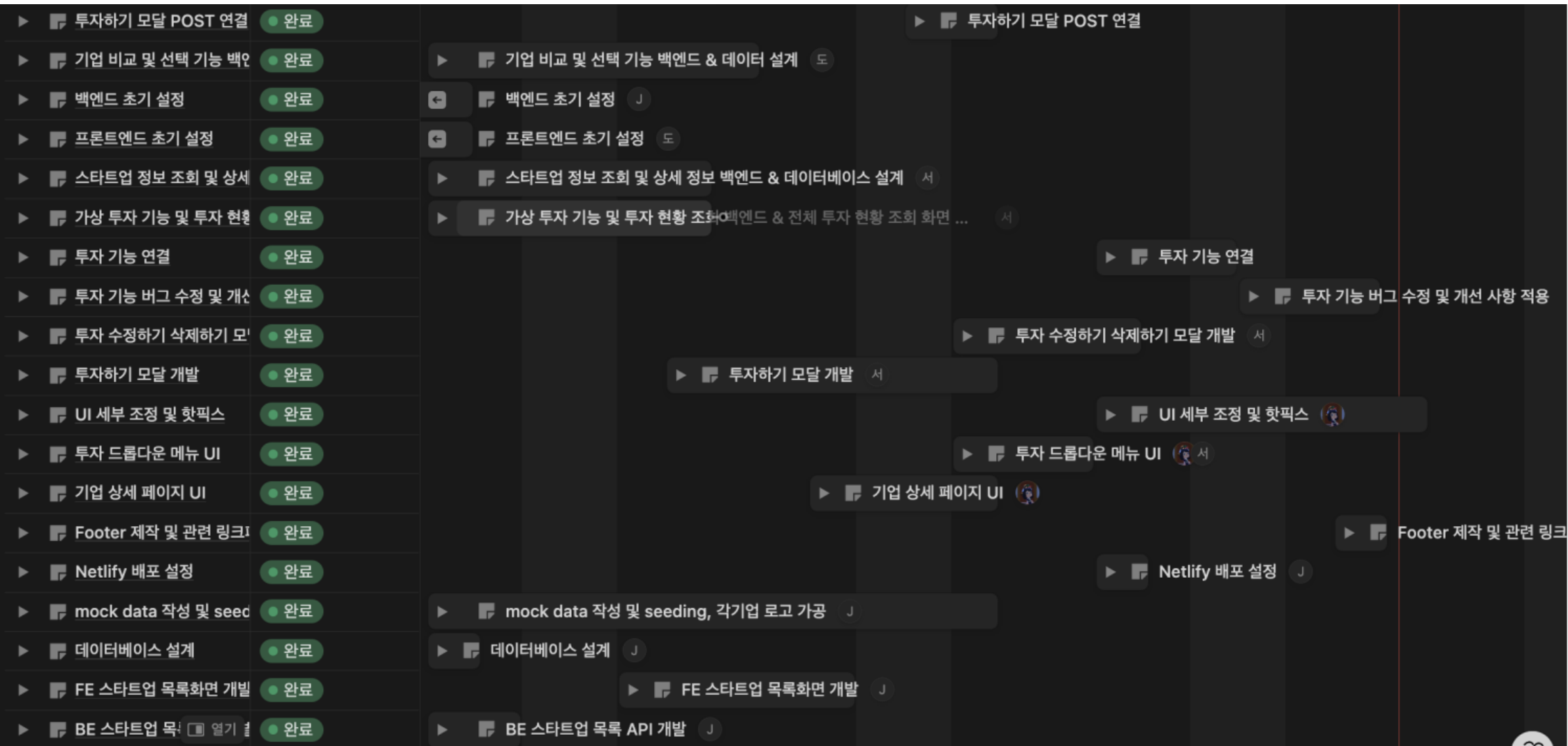
- 가상 투자 기능 API
- 가상 투자 현황 페이지 UI & API
- 투자하기 모달 UI
- Nav UI
- 기업 상세 페이지 API



02

일정 관리

Notion 타임라인 사용



02

이슈 트래킹

Notion 팀 페이지 문제 해결 기록 사용

각자 겪은 문제들과
그 해결 방법에 대해 기록

표

문제 사항

Aa 이름

상태

작성자

비교 결과 페이지 session 기록

해결

도엽 이

페이지네이션 컴포넌트 적용

해결

도엽 이

드롭다운 컴포넌트 적용

해결

도엽 이

투자하기 ID 필드 중복 오류

해결

서 서유림

미해결

미해결

+ 새로 만들기

02

코드 리뷰 및 개발 컨벤션

코드 리뷰 컨벤션

PR & Issue Rule

- PR 올릴 때 dev 브랜치로 설정
- 2명이상 Approve 했을 시 Merge 가능
- 다른 사람이 먼저 PR Merge 되면
Conflict 가능성 높음
→ 해결하고 다시 올리기
- Conflict 나면 디스코드에 꼭 이야기 해주기
- Conflict 해결 후에도 꼭 다시 전달

Branch Name 설정

1. feature branch
 - featrue/기능이름: 특정 기능을 개발
ex) feature/login
2. bugfix branch
 - bugfix/버그이름: 버그를 수정
ex) bugfix/loginError
3. refactor branch
 - refactor/리팩토링이름: 코드 리팩토링
ex) refactor/api, refactor/index.js

코딩 컨벤션 (prettier)

```
"[javascript]": {  
  "editor.defaultFormatter": "esbenp.prettier-vscode",  
  "prettier.singleQuote": true,  
  "prettier.tabWidth": 2,  
  "prettier.trailingComma": "none"  
}
```

네이밍 컨벤션

- 변수 : Camel Case(일반),
Snake Case(상수)
- 함수: Camel Case
- 컴포넌트 : Pascal Case
- 파일 : Camel Case
- 폴더명 : Kebab Case

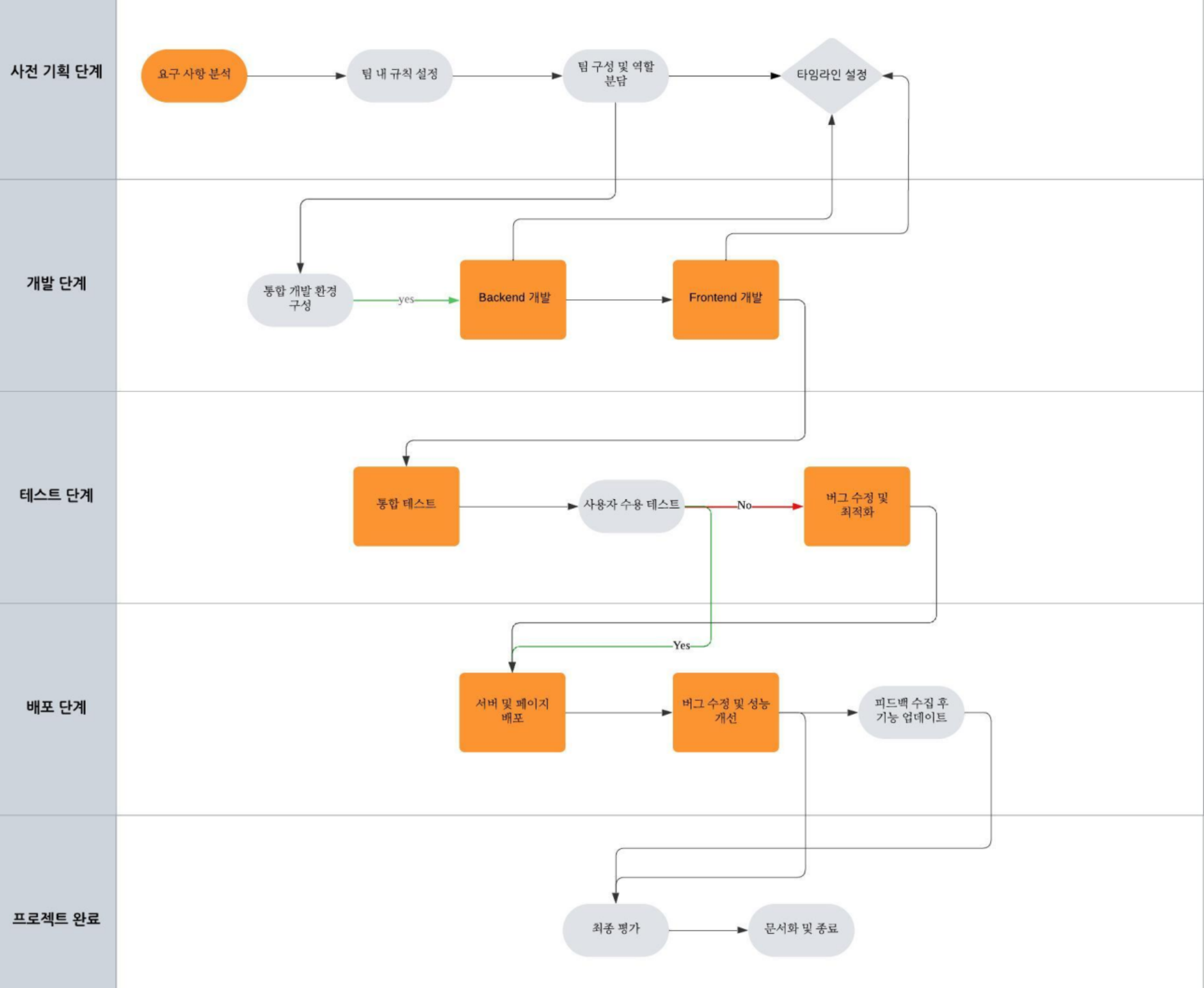
커밋 컨벤션

- Feat : 새로운 기능 추가
- Fix : 버그 수정
- Test : 테스트 코드
- Docs : 문서 수정

개발 컨벤션

03

프로젝트 절차



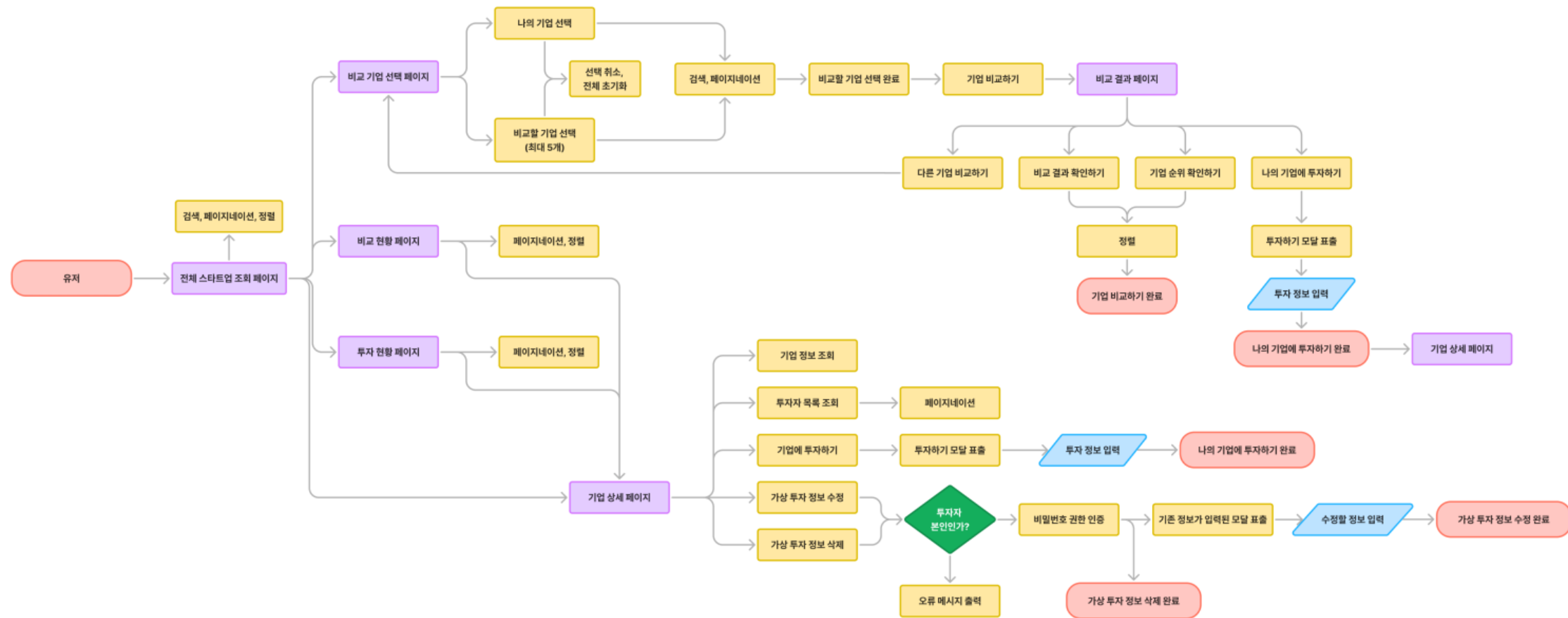
03

페이지 플로우



03

유저 플로우



03

ERD

카테고리		Category				
id	id	Int	NOT NULL	Default value	Autoincrement	
카테고리 이름	category	String	NOT NULL	Default value	Comment	



스타트업		Startup				
id	id	Int	NOT NULL	Default value	Autoincrement	
카테고리 id	categoryId	Int	NOT NULL	Default value	Autoincrement	
기업명	name	String	NOT NULL	Default value	Comment	
실제 누적 투자 금액	actualInvest	BigInt	NOT NULL	0	Comment	
모의 누적 투자 금액	simInvest	BigInt	NOT NULL	0	Comment	
매출액	revenue	BigInt	NOT NULL	Default value	Comment	
고용인원	employees	Int	NOT NULL	Default value	Comment	
기업 소개	description	TEXT	NOT NULL	Default value	Comment	
나의 기업 선택 횟수	SelectedCount	Int	NOT NULL	0	Comment	
비교 기업 선택 횟수	comparedCount	Int	NOT NULL	0	Comment	
기업 로고	image	String	NULL	Default value	Comment	
기업 생성 시간	createdAt	DateTime	NOT NULL	Default value	Comment	
기업 수정 시간	updatedAt	DateTime	NOT NULL	updateAt	Comment	



모의 투자자		MockInvestor				
id	id	Int	NOT NULL	Default value	Autoincrement	
스타트업 id	startupId	Int	NOT NULL	Default value	Autoincrement	
투자자 이름	name	String	NOT NULL	Default value	Comment	
모의 투자 금액	investAmount	BigInt	NOT NULL	0	Comment	
투자 코멘트	comment	TEXT	NOT NULL	Default value	Comment	
비밀번호	password	String	NOT NULL	Default value	Comment	
투자 생성 시간	createdAt	DateTime	NOT NULL	now()	Comment	
투자 수정 시간	updatedAt	DateTime	NOT NULL	updatedAt	Comment	

나의 기업		Selection				
id	id	Int	NOT NULL	Default value	Autoincrement	
스타트업 id	startupId	Int	NOT NULL	Default value	Autoincrement	
세션 id	sessionId	String	NOT NULL	Default value	Comment	
나의 기업선택 상태	isSelected	Boolean	NOT NULL	false	Comment	
나의 기업 선택 시간	createdAt	DateTime	NOT NULL	now()	Comment	
나의 기업 선택 수정 시간	updatedAt	DateTime	NOT NULL	Default value	Comment	



비교 기업		Comparison				
id	id	Int	NOT NULL	Default value	Autoincrement	
스타트업 id	startupId	Int	NOT NULL	Default value	Autoincrement	
세션 id	sessionId	String	NOT NULL	Default value	Comment	
비교 기업 선택 상태	isCompared	Boolean	NOT NULL	false	Comment	
비교 기업 선택 시간	createdAt	DateTime	NOT NULL	now()	Comment	
비교 기업 수정 시간	updatedAt	DateTime	NOT NULL	updatedAt	Comment	

03

API 명세

분류	URI	메소드	기능
스타트업	/api/startups	GET	전체 기업 조회
스타트업	/api/startups/{startupId}	GET	특정 기업 상세 조회
스타트업	/api/startups/{id}/rank	GET	특정 스타트업의 상하위 순위 조회
비교	/api/comparisons	GET	전체 스타트업 조회 (검색, 정렬)
비교	/api/comparisons/recent-selection	GET	최근 선택된 스타트업 조회
비교	/api/comparisons/compare-result	GET	선택한 스타트업 비교 결과 조회
선택	/api/selections/my-startups	POST	나의 스타트업 선택
선택	/api/selections/comparison-startups	POST	비교할 스타트업 여러 개 선택
선택	/api/selections/cancel-my-startups	POST	나의 스타트업 선택 취소
선택	/api/selections/cancel-comparison-startups	POST	비교할 스타트업 선택 취소
선택	/api/selections	GET	세션에 저장된 스타트업 불러오기
투자	/api/investments	GET	전체 투자 현황 조회
투자	/api/investments	POST	특정 기업에 투자하기
투자	/api/investments/{investmentId}	PATCH	투자 수정
투자	/api/investments/{investmentId}	DELETE	투자 삭제

03

트러블 슈팅

발생한 문제

투자하기 기능이
동작하지 않는 문제 발생

투자하기 기능을 제작한
사람은 동작을 하지만
다른 팀원들은 동작하지
않는 상황

원인 분석

팀원들에게 표출되는
에러 메시지 검색

id 필드에 관련하여
유니크하지 않다는
오류 메시지 표출

데이터베이스에 문제가
있는 것으로 유추

해결 방법

강사님의 도움으로 id 필드의
시퀀스가 중복되어 나타나는
현상이라는 것을 알게 됨.

제작자만 되었던 이유는
미리 데이터베이스에 시퀀스를
추적하도록 설정해놓았기 때문.

따라서 seeding 할 때
시퀀스 관련 설정을 추가하여
오류 해결

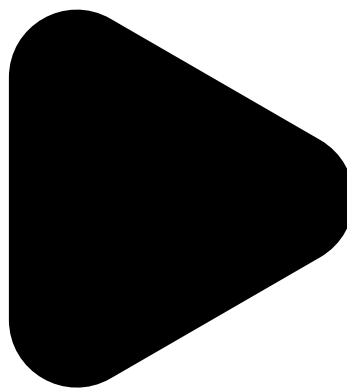
배운 점 및 개선 사항

새로운 것을 시도할 때는
그것에 관련된 내용을 많이
찾아보고 해야겠다고 생각

또한 에러 메시지의 중요성을
깨닫게 되는 계기가 되었음.

04

시연 영상



1. 트랜잭션 처리를 통한 데이터 무결성 유지

- 투자 시 기업의 누적 투자 금액과 개인의 투자 내역을 **트랜잭션으로 처리**하여 데이터의 불일치를 방지하고 동시성 제어를 강화했습니다.

2. 세션 기반의 기업 비교 정보 유지

- 브라우저를 닫기 전까지 사용자의 기업 비교 정보를 **세션을 통해 지속적으로 유지**합니다.

3. 동순위 항목에 대한 완벽한 정렬 구현

- 매번 API 요청 시 동일한 결과를 보장하여 **데이터의 일관성을 유지**합니다.

4. 숫자 데이터의 가독성 향상

- 투자 금액, 매출액 등의 숫자에 3자리마다 **코마를 삽입**하여 읽기 쉽도록 표시합니다. (투자 금액 입력 시에도 실시간 적용)

5. 모바일 환경에서의 안정적인 레이아웃 제공

- 텍스트 버튼, 검색창, 드롭다운 메뉴 등이 **모바일 화면**에서도 깨지지 않고 **정상적으로 표시**됩니다.

6. 사용자 편의성을 높인 입력 방식

- 팝업창이 뜰 때 **입력 창에 자동으로 포커스**되며, 데이터 입력 후 버튼 클릭 없이 엔터 키를 눌러 해당 프로세스를 수행할 수 있도록 **사용자 편의성을 증대**시켰습니다.

아쉬운 점

1. 코드 리뷰 부족

- 시간에 쫓겨 코드리뷰가 충분히 이루어지지 못한 점

2. 리팩토링 기회 부족

- 기능 구현 및 테스트에 집중하다 보니,
이미 작성된 코드를 개선할 기회가 부족

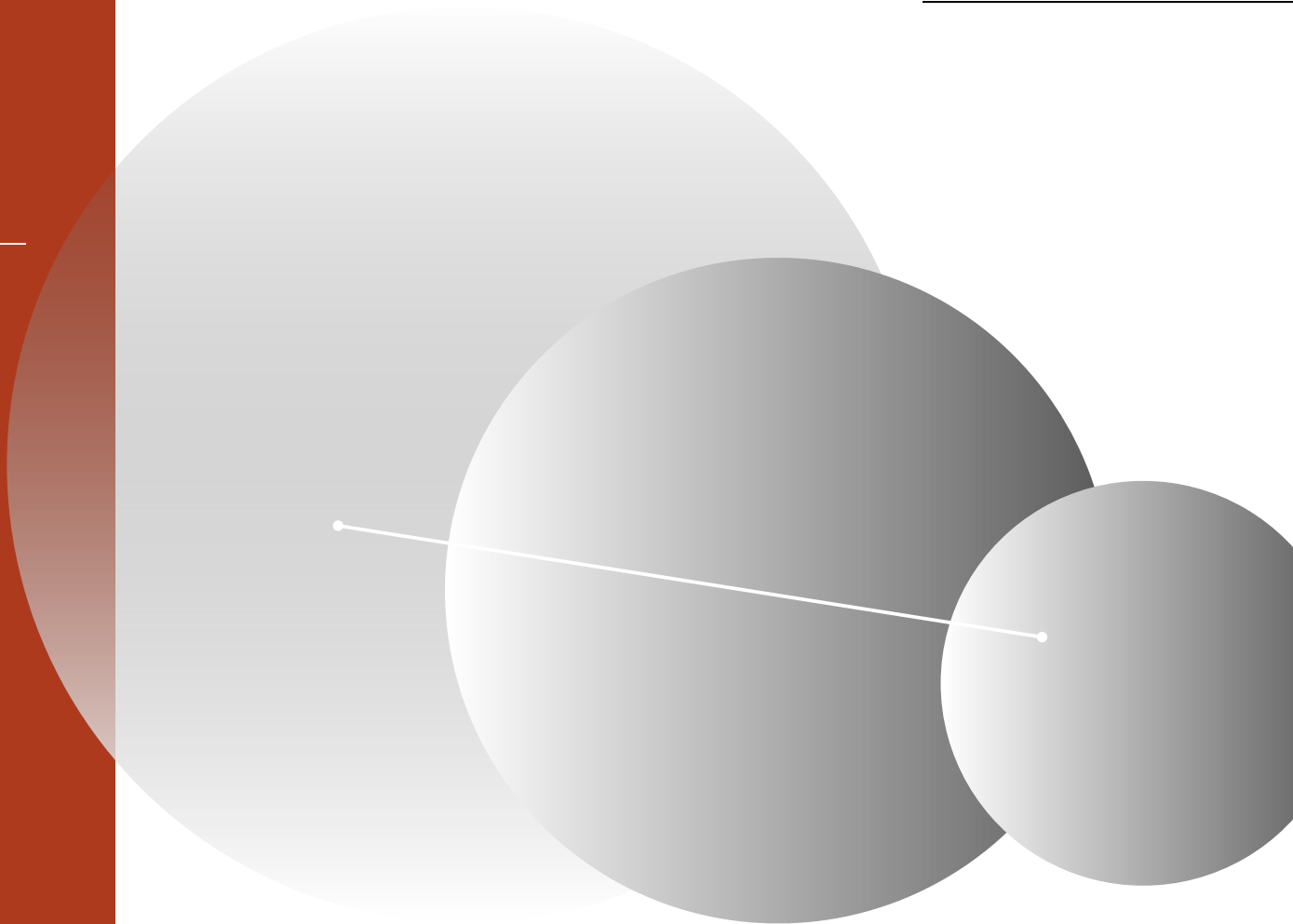
느낀 점

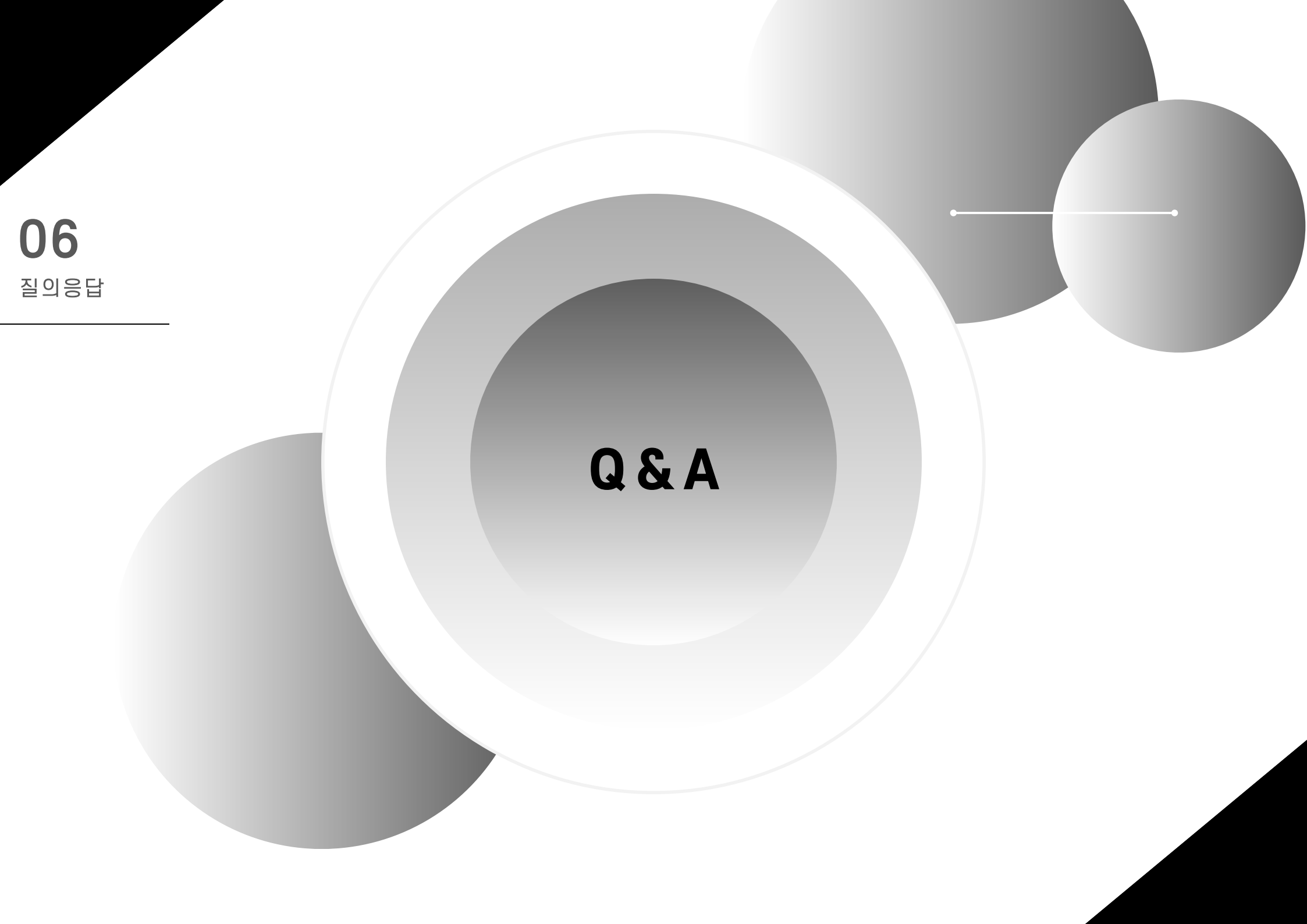
<협업의 필요성>

혼자 프로젝트를 진행했으면 기능 구현이나 에러를 해결하는데 더 빠를 것이라고 생각했습니다.

하지만 막상 겪어보니, 각자 맡은 부분만 하더라도 시간이 많이 소요되었고, 문제를 함께 해결함으로써 빠르고 정확하게 해결할 수 있었습니다.

이번 첫 팀 프로젝트를 통해 함께 역할을 나누고 문제를 해결하는 팀 활동의 필요성을 느꼈고, 팀원들에게 도움도 많이 받으며 서로에게 배울 수 있는 기회였던 것 같아서 좋았습니다.





06

질의응답

Q & A

서비스 배포 링크

Netlify 주소

<https://view-mystartup.netlify.app/>

Render 주소

<https://season2-view-my-startup-3team-be.onrender.com>

Notion 팀 페이지

<https://www.notion.so/1164170bdea6807eb3c3d997ccd72c22?v=1164170bdea68181bcb4000c0389c019>

GitHub

<https://github.com/season2-3team/season2-View-My-Startup-3team-FE>
<https://github.com/season2-3team/season2-View-My-Startup-3team-BE>

지금까지 3팀의 발표를 들어주셔서

감사합니다!