

(1회차) 24시간 챌린지 - 생성 AI로만 Jobgorithm 구현해보기

스타트업 SCV가 AI 서비스들을 만났을 때 = 더 많이 일하고 더 많이 놀기

건설로봇 – Space Construction Vehicle, SCV

프로필

×



sungtae.ryu

편집

CEO (a.k.a SCV...)

웨이인(Wayne)

• 온라인

🕒 현지 시간 오전 6:11

상태 설정

다음... ▾

⋮

연락처 정보

편집

✉ 이메일 주소
wayne.ryu@goorm.io

☎ 전화
010-9957-7152

Wayne Ryu (SCV a.k.a CEO)

- 스타크래프트에 등장하는 SCV는 Terran 종족의 건설과 자원 채취를 담당하는 유닛입니다.
- 멀티태스킹과 효율적인 작업 능력을 자랑합니다.
- 마치 SCV처럼, 회사의 모든 업무 분야에서 실무 영역까지도 도맡아 술선수범하고 싶습니다.
- 대표이사이지만 실무형 리더로서, 구성원들과 수평적으로 소통하며 함께 성장하고자 합니다.

SCV에게 생성형 AI가 주는 의미 – 진정한 멀티 태스킹

- **생산성 극대화:** 반복적인 작업을 자동화하여 더 많은 시간과 자원을 핵심 업무에 투입할 수 있습니다.
- **혁신적인 아이디어 창출:** 생성형 AI를 통해 새로운 접근 방식과 솔루션을 발견하여 사업의 혁신을 이끌어냅니다.
- **업무 효율성 향상:** AI를 활용한 데이터 분석과 의사결정 지원으로 업무 프로세스를 최적화합니다.
- **(하고 싶은 것) 팀 역량 강화:** 구성원들이 AI 도구를 활용하여 개인의 역량을 향상시키고, 상호 협력을 강화합니다.

스타트업 SCV가 AI 서비스들을 만났을 때 = 더 많이 일하고 더 많이 놀기

- (1회차) 24시간 챌린지 - 생성 AI로만 Jobgorithm 구현해보기
- (2회차) 생성AI와 마크다운 기반으로 발표자료와 제안서를 한 번에!
- (3회차) 생성 AI로만 내가 원하는 SVG 이미지 만들기
- (4회차) 나처럼 생각하고 일하는 AI와의 협력

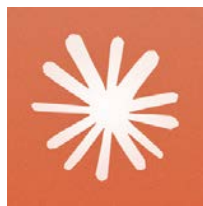
(1회차) 목차

- 생성형 AI 서비스를 활용한 서비스 개발
 - 1단계: 스케치 (기획/설계)
 - ChatGPT o-preview 활용
 - goormIDE(Arkain) 활용
 - 2단계: 밑채색 (기본 구현)
 - 디자인 시스템 적용
 - 복사-붙여넣기의 반복...
 - 3단계: 채색 (기능 구현)
 - 컨텍스트 윈도우가 큰 곳으로 갈아타기
 - Claude 3.5 Sonnet 활용
 - 소스 코드를 통째로 전달하자
 - 4단계: 보정 및 마무리
 - 좀 더 복잡한 채팅 앱 개발 시도
 - 효율적인 프롬프트 작성 방법
 - patch.js 만들어 쓰기 & E2E 테스트 자동화
- 정리 및 (2회차) 예고

현재 사용하고 있는 생성형 AI 서비스들



ChatGPT



Claude



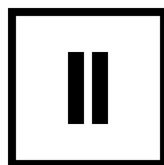
Perplexity



MidJourney



Runway



ElevenLabs



Cursor

arkain

goormIDE (Arkain)

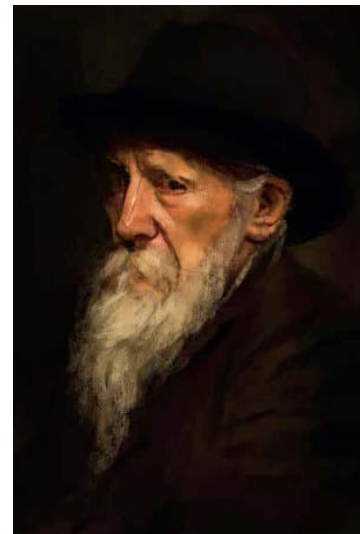
그림을 그리는 것처럼 코드를 작성하기



스케치용 연필



밑채색을 위한 큰 붓



디테일 묘사를 위한 작은 붓



세부 조정 및 보정



ChatGPT o-preview



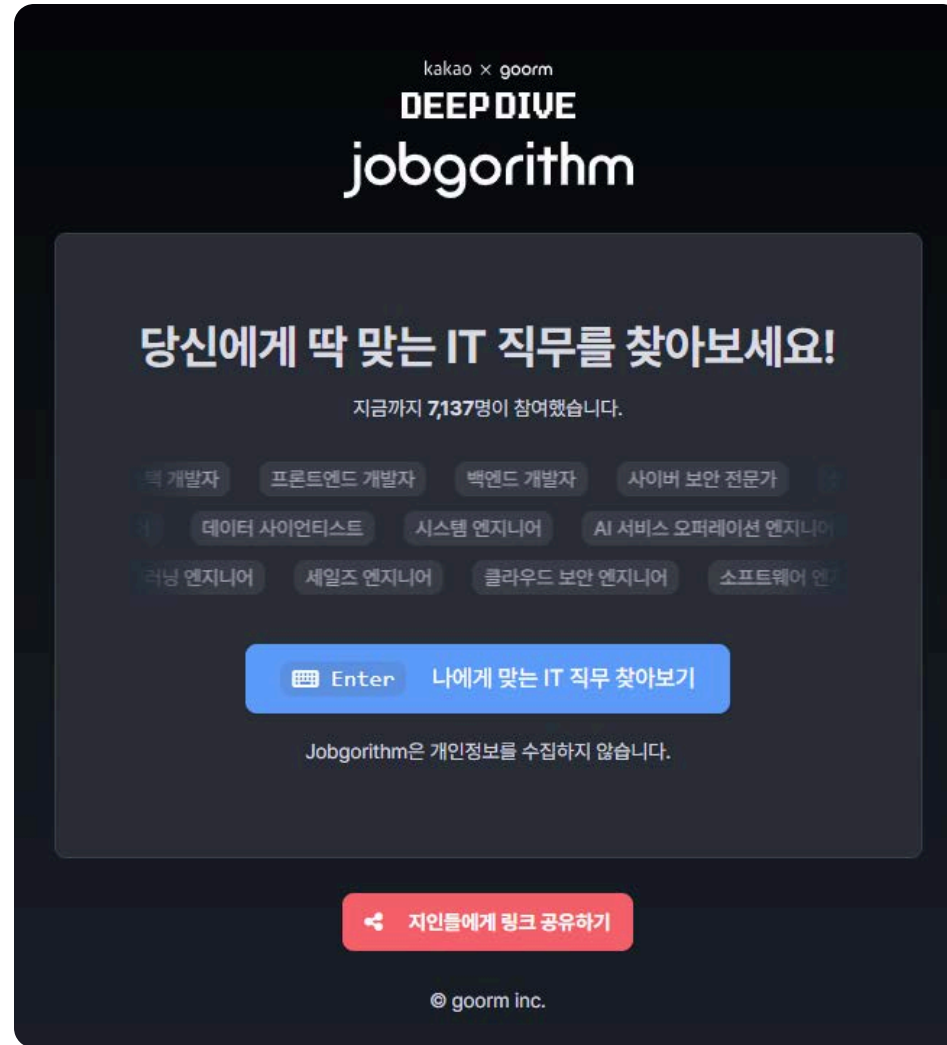
ChatGPT o-preview + 4o



ChatGPT 4o
+ Claude 3.5 Sonnet

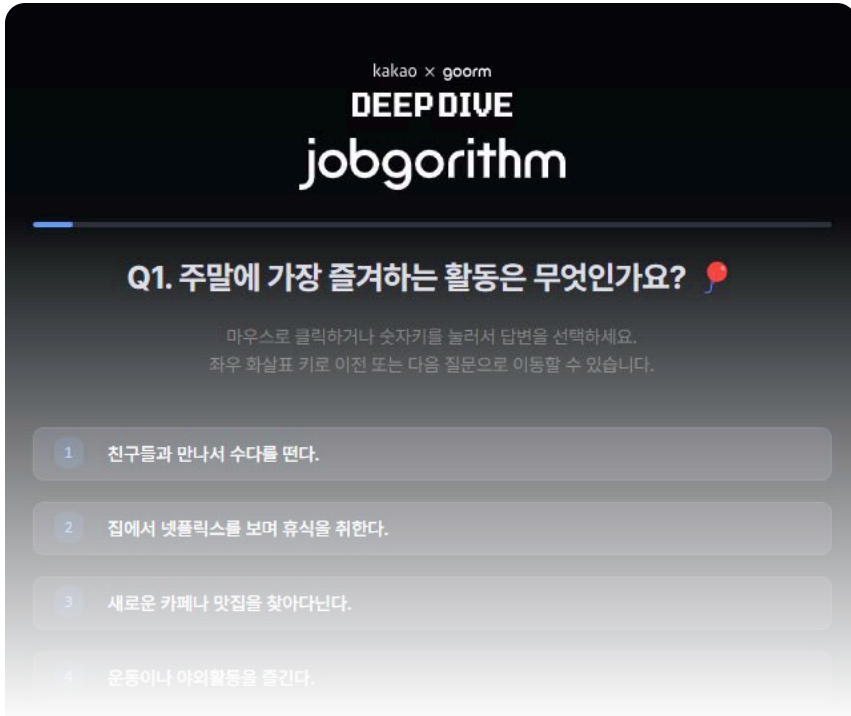


ChatGPT 4o
+ Claude 3.5 Sonnet



<https://jobgorithm.goorm.io>

Jobgorithm 개발 배경



DEEP DIVE 모객 강화 목적

- **인터랙티브 바이럴 마케팅(Interactive Viral Marketing)** 전략의 일환으로 기획
 - 사용자 참여형 콘텐츠를 통한 자발적 공유 유도
 - 소셜 미디어 상에서의 높은 전파력 확보
- **게이미피케이션(Gamification)** 요소 활용
 - 성향 테스트 형식을 통한 사용자 몰입도 증대
 - 결과 공유 기능으로 자연스러운 바이럴 효과 창출
- **퍼널 마케팅(Funnel Marketing)** 접근
 - 흥미로운 성향 테스트로 잠재 고객 유입(Awareness)
 - 테스트 결과를 통한 K-Digital Training 과정 연계(Conversion)
- **데이터 기반 인사이트 수집**
 - 참여자들의 직무 선호도 및 성향 데이터 확보
 - 향후 교육과정 기획에 활용 가능한 마케팅 인텔리전스 구축

Jobgorithm 네이밍 (1/2)

나에게 알맞은 IT직무가 무엇인지 20여개의 질문응답을 통해서 알아볼 수 있는 간단한 서비스의 이름을 재미있고 입에 달라붙게, 기억하기 쉬운 방향으로 만들어보자.



메모리 업데이트됨

여러 질문을 통해 적합한 IT 직무를 찾아주는 서비스 이름을 재미있고 기억하기 쉽게 만들면, 다음과 같은 아이디어가 있습니다:

1. **"IT 나와라!"** - 질문을 통해 나에게 맞는 직무를 '불러내는' 느낌으로 재미있고 기억하기 쉬운 이름.
2. **"직무핏 20Q"** - 20개의 질문으로 직무를 맞추는 게임 같은 느낌을 살린 이름.
3. **"IT 직무콕!"** - 원하는 IT 직무를 '콕' 집어내는 직관적이고 간결한 이름.
4. **"잡서칭 퀘스트"** - 게임 퀘스트처럼 직무를 찾아가는 과정이 흥미롭고 도전적인 느낌을 주는 이름.
5. **"IT 직무 탐구생활"** - 가볍게 직무를 탐구하는 느낌을 주면서도 친근한 이름.
6. **"What's My IT?"** - 질문을 통해 내가 맞는 IT 직무를 찾는다는 의미를 담은 트렌디한 영어 표현.
7. **"20Q로 IT정복"** - 질문으로 IT 세상에서 내가 맞는 길을 찾는다는 컨셉으로, 다소 모험적인 느낌.

이 중에 혹시 더 끌리는 방향이 있나요?

좀 더 재미있는 영어 단어나 합성어, 새로운 생성어같은 거 없을까?



좀 더 재미있고 창의적인 느낌을 주는 새로운 생성어 또는 합성어 아이디어를 다음과 같이 제안해 봅니다:

1. **"QuizIT!"** - "Quiz"와 "IT"의 합성어로, 질문을 통해 IT 직무를 찾아가는 간단하면서도 캐치한 이름.
2. **"JobITize"** - "Job"과 "Digitize"를 합성해 직무를 디지털 방식으로 찾아가는 느낌을 강조한 이름.
3. **"IT Matcherator"** - "Match"와 "Generator"의 합성어로, IT 직무를 맞춰주는 재미있는 이름.
4. **"SkillZilla"** - "Skill"과 강력한 느낌의 "-zilla"를 합성한, 나만의 IT 스킬을 강력하게 찾는다는 의미의 이름.
5. **"TechQuestify"** - "Tech"와 "Questify"의 합성어로, 직무 탐색을 게임 퀘스트처럼 재미있게 표현한 이름.
6. **"RoleHuntz"** - "Role"과 "Hunt"에 재미있는 철자 변형을 더해 기억에 남을 만한 이름.
7. **"FitNTech"** - "Fit"와 "Tech"의 합성어로, 나에게 딱 맞는 IT 직무를 찾아주는 재미있는 느낌의 이름.

이름들이 좀 더 독특하고 흥미로운 느낌을 줄 수 있을 것 같아요. 혹시 이 중에 마음에 드는 방향이 있나요, 아니면 추가적인 아이디어가 필요할까요?

Jobgorithm 네이밍 (2/2)

좀 더 참신한 아이디어 없어



더 참신한 느낌을 주기 위해 완전히 새로운 생성어와 합성어를 시도해볼게요:

1. **"Jobgorithm"** - "Job"과 "Algorithm"을 합성한 이름으로, 알고리즘처럼 질문을 통해 직무를 찾아낸다는 의미. 기술적이면서도 직무를 찾는 과정이 시스템화된 느낌.
2. **"TechSplorer"** - "Tech"와 "Explorer"를 결합해 IT 직무를 탐험하듯 찾는다는 의미를 담은 이름.
3. **"Skillverine"** - "Skill"과 강력한 동물인 "Wolverine"을 결합하여 강력하게 나에게 맞는 스킬을 찾아준다는 의미의 재미있는 이름.
4. **"RoleMancer"** - "Role"과 마법사 "Necromancer"를 결합한 독특한 이름. 마치 마법처럼 내 직무를 예측하고 찾아낸다는 컨셉.
5. **"ByteMatch"** - "Byte" (데이터 단위)와 "Match"를 합성하여, IT 세계에서 데이터를 활용해 직무를 매칭해 준다는 느낌의 이름.
6. **"QWERTY Quest"** - 키보드 배열 "QWERTY"와 "Quest"를 결합해, 키보드를 두드리며 직무를 찾아가는 여정을 표현한 이름.
7. **"NexTechFit"** - "Next"와 "Tech"를 결합한, 미래의 기술 직무를 찾아가는 과정에서 나와 가장 잘 맞는 직무를 찾아준다는 의미를 담은 이름.
8. **"Codyyssey"** - "Code"와 "Odyssey"를 합성한 이름으로, 코딩 세계 속에서 긴 여정을 통해 직무를 발견하는 스토리적인 느낌.
9. **"GiggleIT"** - "Gig" (IT 작업)와 "Giggle" (웃음)를 결합한 가볍고 즐거운 느낌의 이름으로, 직무를 재미있게 찾는 과정을 강조.
10. **"Digitrails"** - "Digital"과 "Trails"를 결합해, 디지털 시대에서 나만의 IT 직무를 탐험하는 길을 나타내며 여정의 느낌을 표현한 이름.

Jobgorithm 전체 애플리케이션 구조 설계



서비스 구조와 기능 개요

- Jobgorithm은 사용자의 응답을 기반으로 맞춤형 직무 추천을 제공하는 웹 애플리케이션
- 주요 기능
 - 사용자 질문 응답 수집
 - 응답 데이터를 기반으로 한 직무 추천 알고리즘
 - Chart.js를 통한 데이터 시각화
 - 공유 기능 (결과 페이지 및 체험 URL)
- UI/UX
 - Vapor Design System 기반으로 구름의 아이덴티티 유지
 - Vapor Design System을 쓰면 개똥같이 만들어도 예쁘게 보임
 - 또한, React 코드와 CSS 코드의 양이 줄어듦

일단... "해줘"



1단계 : 스케치

- 지금부터 취향이나 성격 테스트를 통해 나에게 맞는 IT 직무를 찾는 간단한 서비스를 만들어보자.
- 풀스택 개발자 / 프론트엔드 개발자 / 백엔드 개발자 / 사이버 보안 전문가 / 생성형 AI 응용 개발자 / 클라우드 엔지니어 / 프로젝트 디자이너 / 프로젝트 매니저 이렇게 8개의 직종을 중심으로 파생되는 2가지 직군까지 추가해서 총 24 종류의 직군 중에 자신에게 어울리는 IT 직군 1, 2, 3 순위를 추천하는 서비스를 만들어보자.
- 문제를 해결하기 위해 단계별로 고민하고 실행해서 작성해줘.
- node.js 와 React를 사용해서 서비스를 만들어줘.
- Reactstrap을 사용해서 서비스를 만들어줘.
- 질문은 모두 객관식 오지 선택형으로 작성해줘.
- 질문은 최대한 재미있고 일상 생활에서의 예시에 맞게 작성해줘.
- 질문의 개수는 20개 정도가 좋을 거 같아.

{콘텐츠, 코드, 디자인}에 대한 기획, 설계를 하는 단계

24개의 직군은 겹치거나 상위 개념이 아닌 모두 독립적이고 수평적인 수준에서 각각 특성을 가지는 서로 다른 24개의 1차원 목록이어야 해

24개의 직군들은 무조건 1개의 아래 교육 과정들 중 하나랑 매칭되어야 해. 이 기준에 맞게 24개의 직군들 목록을 수정해줘.

풀스택 개발자 / 프론트엔드 개발자 / 백엔드 개발자 / 사이버 보안 전문가 / 생성형 AI 응용 개발자 / 클라우드 엔지니어 / 프로젝트 디자이너 / 프로젝트 매니저

질문의 보
해줘. 결

작성한 코드가 정상동작하는 지 어떻게 확인하나요?
서비스를 만들어서 어떻게 제공하나요?

arkain

자동으로 생성된 프로젝트의 구조는?

프로젝트 구조

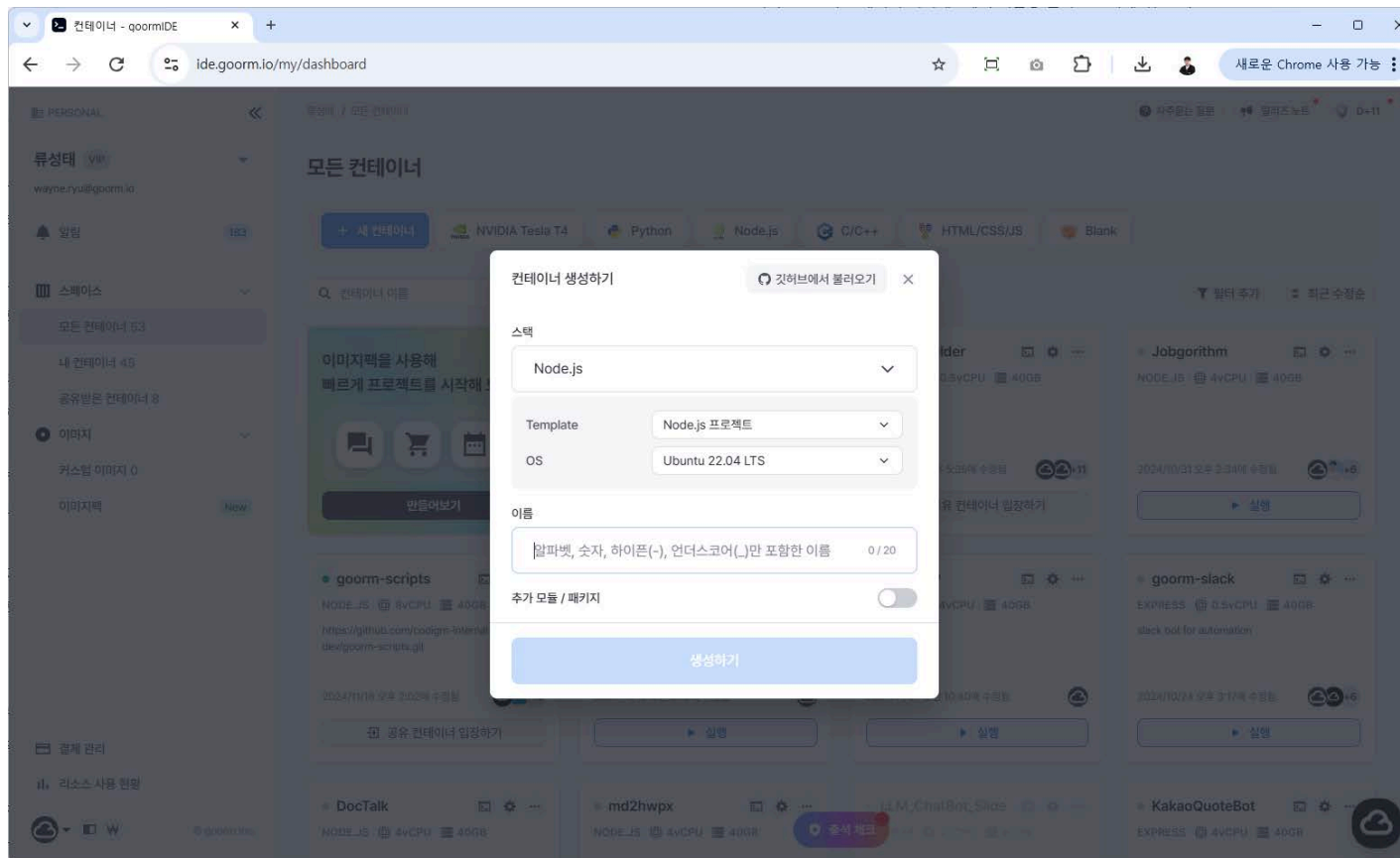
kotlin

코드 복사

```
jobgorithm/  
├─ server.js  
├─ package.json  
├─ frontend/  
│   ├─ package.json  
│   └─ src/  
│       ├─ index.js  
│       ├─ App.js  
│       └─ components/  
│           ├─ LandingPage.js  
│           ├─ TypeForm.js  
│           ├─ Result.js  
│           └─ Question.js  
│       └─ assets/  
│           └─ styles.css  
└─ public/  
    └─ index.html  
├─ models/  
│   ├─ User.js  
│   └─ Counter.js  
├─ data/  
│   ├─ questions.js  
│   └─ jobs.js  
└─ README.md
```

저 파일들은 언제 다 만들어...
그리고 환경 구성은? 필요한 라이브러리 설치는?

실시간 데모: 구름IDE를 이용한 코드 환경 세팅과 초기 구성



실시간 데모: create_project.js 파일 만들기

프로젝트 구조대로 파일과 디렉토리를 생성하는 코드도 작성해줘.



7초 동안 생각 ▼

물론입니다! 프로젝트 구조에 따라 파일과 디렉토리를 생성하고, 각 파일에 필요한 내용을 작성하는 스크립트를 제공할 것입니다.

아래는 Node.js를 사용하여 파일과 디렉토리를 생성하는 스크립트입니다. 이 스크립트를 실행하면 `jobgorithm` 프로젝트 폴더와 그 안의 모든 파일 및 디렉토리가 생성됩니다.

파일 및 디렉토리 생성 스크립트 (`create_project.js`)

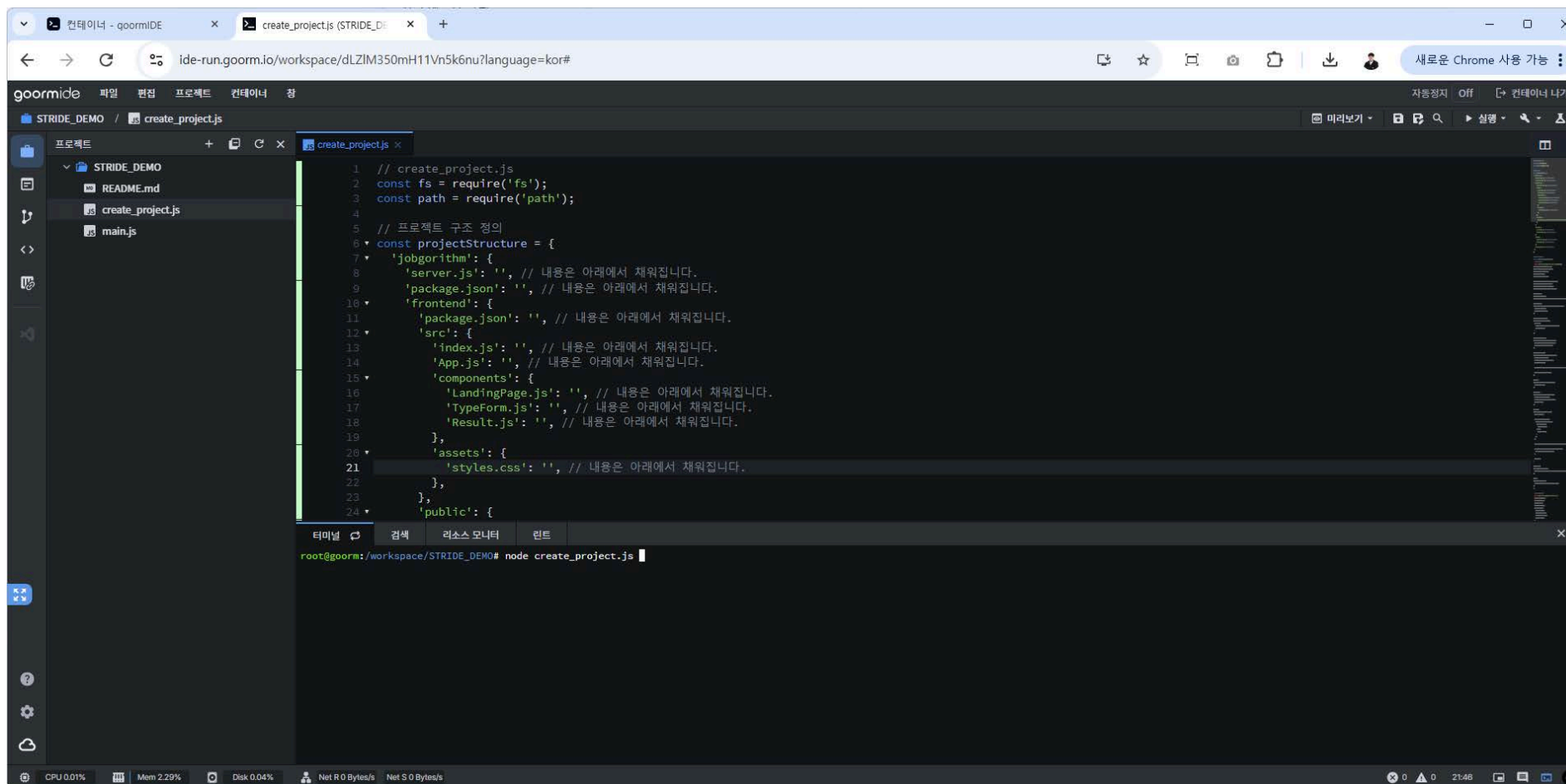
javascript

코드 복사

```
// create_project.js
const fs = require('fs');
const path = require('path');

// 프로젝트 구조 정의
const projectStructure = {
  'jobgorithm': {
```

실시간 데모: 구름IDE에서 create_project.js 실행하기



실시간 데모: 필요한 패키지 설치하기

6. 의존성 설치

생성된 프로젝트 폴더로 이동하여 필요한 패키지를 설치합니다.

```
bash

cd jobgorithm
npm install
cd frontend
npm install
cd ..
```

[코드 복사](#)

7. MongoDB 설정

`server.js` 파일에서 MongoDB 연결 문자열을 실제 사용하는 MongoDB 인스턴스에 맞게 설정해야 합니다.

```
javascript

mongoose.connect('mongodb://0.0.0.0:27017/jobgorithm', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});
```

[코드 복사](#)

8. 프론트엔드 빌드 및 서버 실행

```
bash

cd frontend
npm run build
```

[코드 복사](#)

실시간 데모: 최초의 서비스 실행

실행 URL과 포트

서버 실행에 사용되는 도메인 주소를 등록할 수 있습니다.
Node.js Express의 기본 포트는 3000 입니다.

URL

https://



.run.goorm.site ▼

PORT

80 ~ 65000

등록

등록된 URL과 포트 (컨테이너당 최대 10개)

실행용	URL	PORT
<input checked="" type="radio"/>	https://stride-demo-lbcde.run.goorm.site 	3000 

닫기

localhost가 아니라 0.0.0.0을 사용하는 것으로 전체를 수정해줘. 빠진 파일 생성 부분이나 내용 작성 부분이 없도록 완전한 풀 버전으로 작성해줘.



2단계 : 밀채색

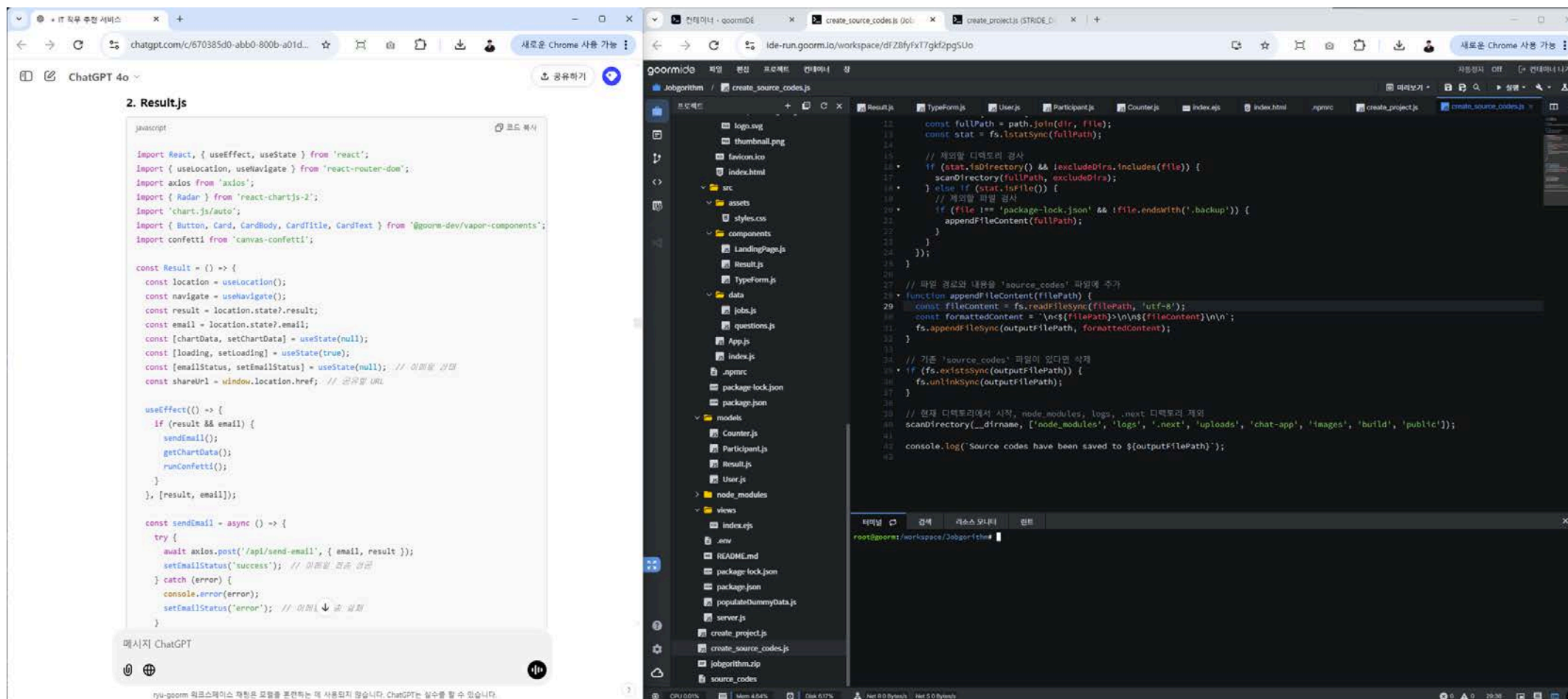
{콘텐츠, 코드, 디자인}에 대한 굵직한 구현을 진행하는 단계

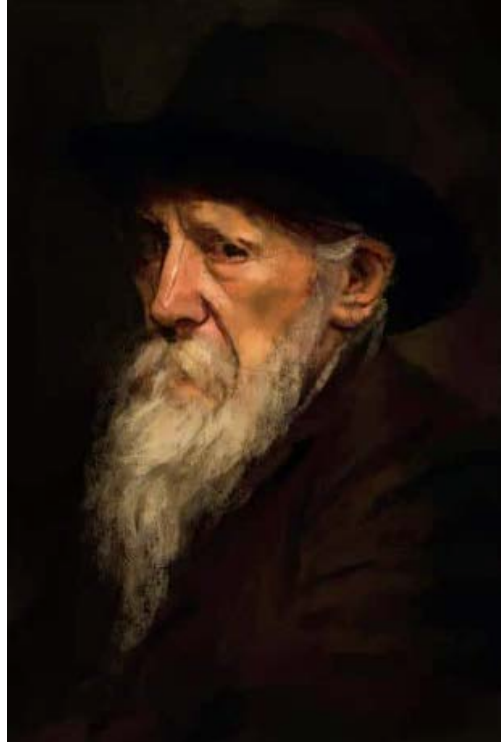
- 앞으로는 수정된 부분 파일에 대해서만 풀버전을 표시해줘.
- 맨 마지막 결과 페이지에서 처음으로 돌아갈 수 있는 인터페이스를 추가해줘.
- 질문 앞 뒤 페이지로 이동할 수 있는 인터페이스를 추가해줘.
- 이메일 주소로 결과 내용을 이메일을 보내줘. 이 이메일은 최대한 이쁘게 꾸며서 보내줘.
- 질문을 진행할 수록 얼마나 진행했는 지 알 수 있는 게이지 UI 를 상단에 표시해줘.
- 질문에 대한 답변은 키보드로도 할 수 있게 UI/UX를 개선해줘.
- 질문에 대한 답변 부분은 파랑색인데 회색 배경으로 표시해줘.

- 얼마나 진행했는 지에 대한 %에 대한 게이지를 상단에 표시해줘.
- 답변을 선택하면 바로 다음 질문으로 가게 해줘.
- 질문과 답변 목록 사이의 간격을 충분히 늘려줘.
- 답변은 키보드의 어떤 부분을 누르면 바로 응답되는 지 알 수 있도록 키보드 단축키에 대한 표기를 해줘.
- 답변 목록은 아주 연한 회색 외곽선만 유지하고 글자색은 검정색으로 배경은 흰색으로 변경해줘.
- 결과 페이지의 줄간격을 보기 좋게 잘 구분해서 만들어줘.
- 메인 페이지에는 개인정보를 수집하지 않는다는 점을 명확하게 표시해줘.
- 이메일로 결과 받기는 버튼을 눌러야 하는 것이 아니라 자동으로 전송되게 해줘.
- 어떤 이메일로 결과를 보냈는 지에 대해서 확인할 수 있는 기능을 넣어줘
- 메인 하얀색 박스 위에는 Jobgorithm 로고 텍스트가 존재하게 해줘.
- 모든 페이지 하단에는 Copyright를 표기해줘. goorm inc로 표기해줘.

- 질문을 진행할 수록 얼마나 진행했는 지 알 수 있는 게이지 UI 를 상단에 표시해줘.
- 질문에 대한 답변은 키보드로도 할 수 있게 UI/UX를 개선해줘.
- 질문에 대한 답변 부분은 파랑색인데 회색 배경으로 표시해줘.
- 질문과 답변 목록 사이의 간격을 충분히 늘려줘.
- 답변은 키보드의 어떤 부분을 누르면 바로 응답되는 지 알 수 있도록 키보드 단축키에 대한 표기를 해줘.
- 답변 목록은 아주 연한 회색 외곽선만 유지하고 글자색은 검정색으로 배경은 흰색으로 변경해줘.
- 결과 페이지의 줄간격을 보기 좋게 잘 구분해서 만들어줘.
- 메인 페이지에는 개인정보를 수집하지 않는다는 점을 명확하게 표시해줘.
- 이메일로 결과 받기는 버튼을 눌러야 하는 것이 아니라 자동으로 전송되게 해줘.
- 어떤 이메일로 결과를 보냈는 지에 대해서 확인할 수 있는 기능을 넣어줘
- 메인 하얀색 박스 위에는 Jobgorithm 로고 텍스트가 존재하게 해줘.
- 모든 페이지 하단에는 Copyright를 표기해줘. goorm inc로 표기해줘.

(가장 귀찮은 작업) 복사-붙여넣기의 무한 반복





3단계 : 디테일 잡기

디자인 시스템을 적용하고, 실제 기능을 완벽하게 구현하는 단계

마지막 결과 페이지에 레이더 차트가 만나오고 있어. 8가지의 개인 성향 수치를 동일 직군을 추천받은 사람들의 평균값과 함께 표시하기로 했었어. 이를 반영해서 다시 수정되어야할 파일들의 풀버전을 출력해줘.

- 차트 데이터는 실제 응답에 기반해서 데이터가 쌓이고 저장되고 평균값을 출력하고 사용자가 응답한 값에 기초한 사용자의 데이터가 출력되도록 전체 코드를 수정 보완해줘.
- 수정해야할 소스 코드 파일들의 풀버전들을 출력해줘.

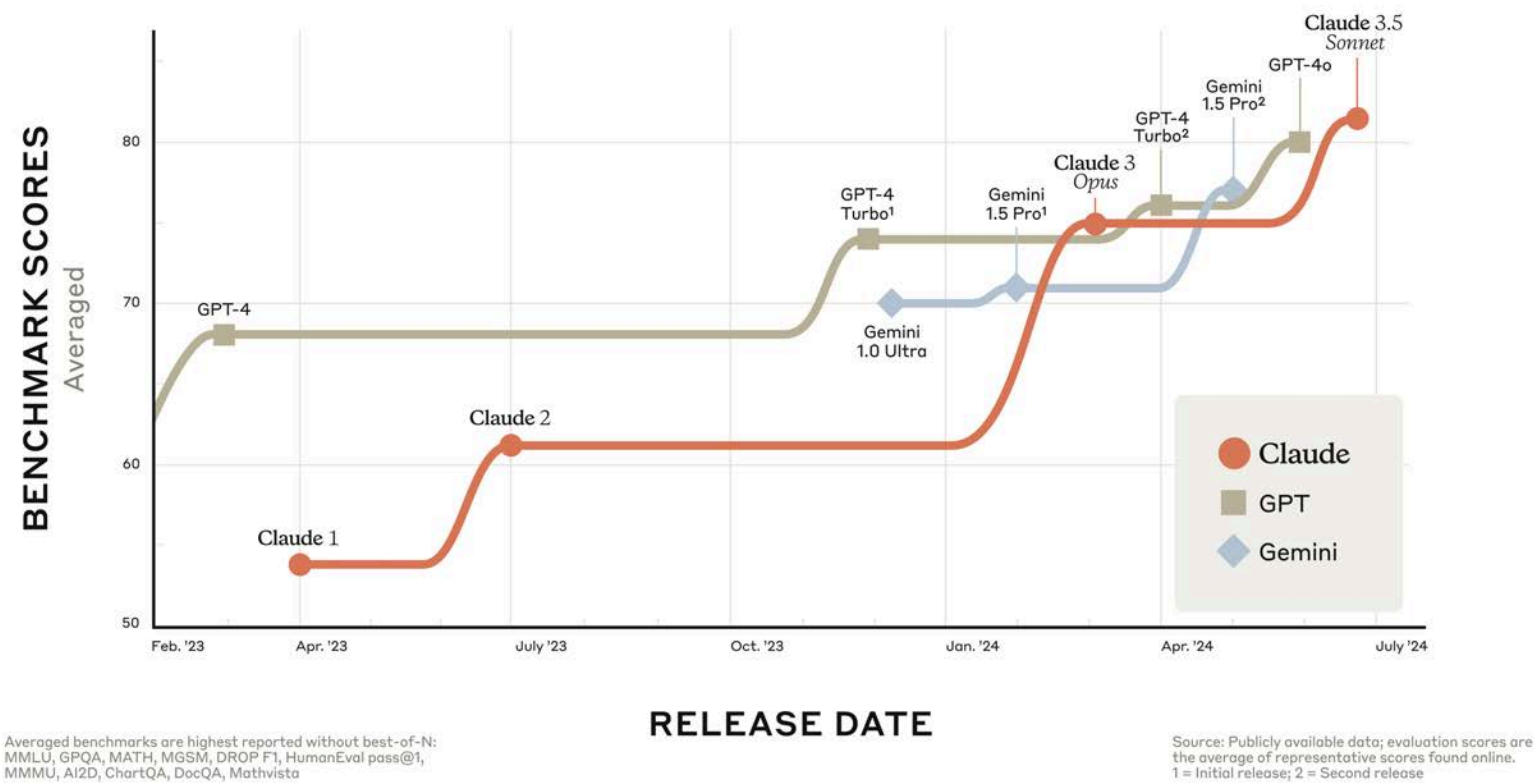
레이더 2
표현해주
현재 나
이것도 나

지금까지 작성된 코드를 Claude로 옮겨가기

- **컨텍스트 윈도우 크기:** Claude 3.5 Sonnet은 더 큰 컨텍스트 윈도우를 제공하여 긴 소스 코드와 여러 파일을 한 번에 처리 가능
- **복잡한 코드 처리 능력:** 더 많은 코드 라인을 한 번에 이해하고 처리할 수 있어 복잡한 프로젝트에 적합
- **효율적인 코드 분석:** 대량의 코드에서 패턴을 인식하고 문제를 식별하는 데 유리
- **향상된 성능:** 대규모 코드베이스에서의 성능 최적화로 빠른 응답 시간 제공
- **확장성:** 프로젝트 규모가 커질수록 더 많은 데이터를 효과적으로 처리할 수 있는 능력

ChatGPT vs Claude Benchmark

AI model release and capabilities timeline



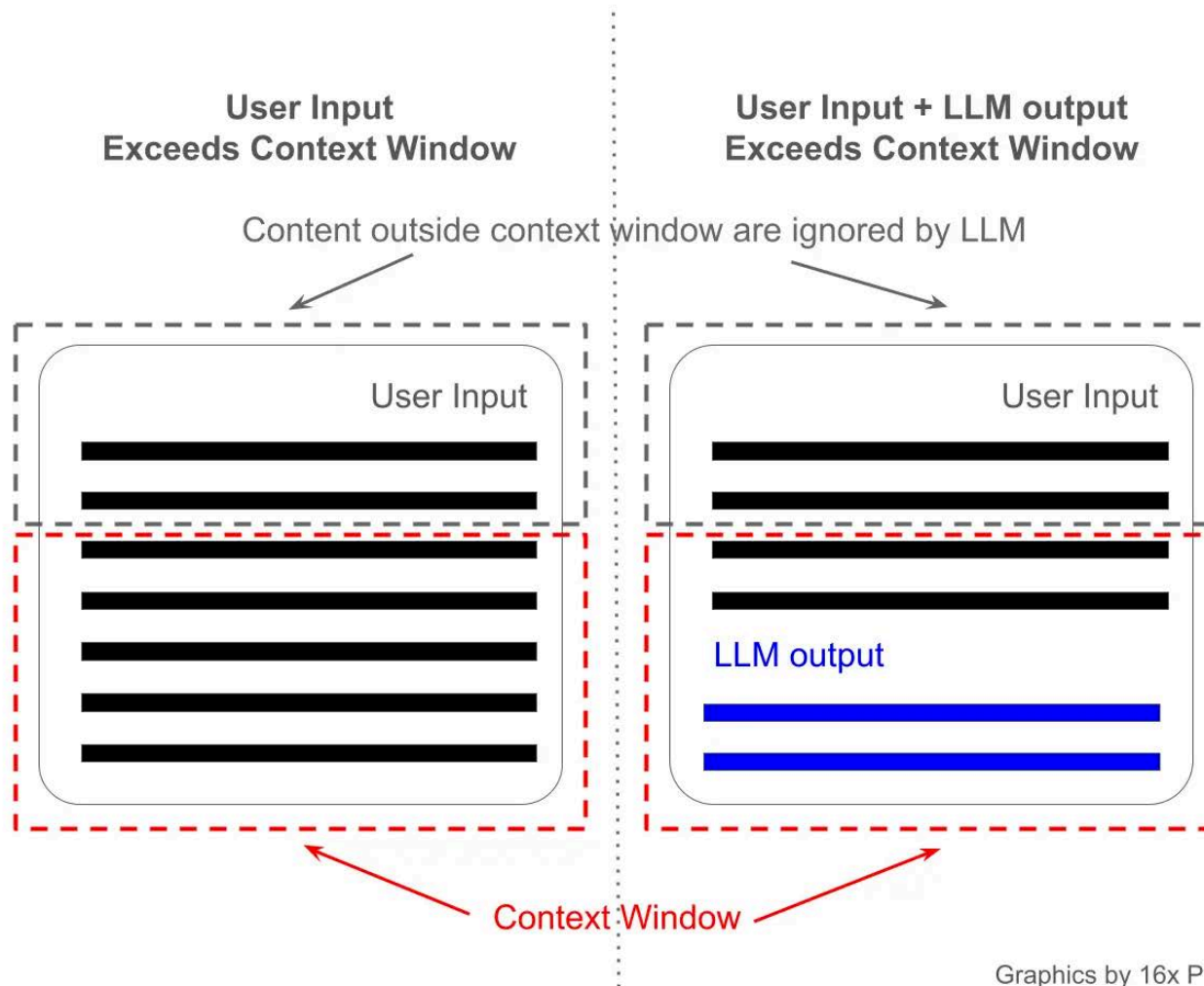
Context Window

컨텍스트 윈도우란?

- 모델이 이해하고 생성할 수 있는 최대 문장 길이
- 입력된 텍스트의 일부를 기억하여 문맥을 유지하는 기능
- 긴 문장을 처리할 수 있는 능력
- 대화의 흐름을 자연스럽게 이어가는 데 기여
- 복잡한 질문에 대한 더 정확한 답변 제공
- 문맥을 고려한 더 나은 텍스트 생성 가능

컨텍스트 윈도우가 길 때의 장점

- 긴 대화나 문서의 문맥을 유지하여 일관성 있는 응답 제공
- 복잡한 문제 해결 시 더 많은 정보 활용 가능
- 대화의 흐름을 자연스럽게 이어가며 사용자 경험 향상
- 긴 텍스트 분석 시 더 많은 세부사항 고려 가능
- 다양한 주제에 대한 포괄적인 답변 제공
- 정보의 누락 없이 전체적인 이해도 향상



Graphics by 16x Prompt

Context Window – ChatGPT vs Claude

Model	Context Window	Max Output
GPT-4o via ChatGPT	4,096 tokens to 8,192 tokens (empirical)	4,096 tokens to 8,192 tokens (empirical)
GPT-4o via API	128,000 tokens	4,096 tokens to 8,192 tokens (empirical)
Claude 3.5 Sonnet	200,000 tokens	4,096 tokens

200,000 토큰 – 4,000~10,000 LOC(Line of Code)

create_source_code.js 파일 만들기

현재 디렉토리의 모든 파일을 source_codes 파일에 저장하는 코드

```
const fs = require("fs");
const path = require("path");

// 결과를 저장할 파일 경로
const outputFilePath = path.join(__dirname, "source_codes");

// 파일을 탐색하는 함수
function scanDirectory(dir, excludeDirs) {
  const files = fs.readdirSync(dir);

  files.forEach((file) => {
    const fullPath = path.join(dir, file);
    const stat = fs.lstatSync(fullPath);

    // 제외할 디렉토리 검사
    if (stat.isDirectory() && !excludeDirs.includes(file)) {
      scanDirectory(fullPath, excludeDirs);
    } else if (stat.isFile()) {
      // 제외할 파일 검사
      if (file !== "package-lock.json" && !file.endsWith(".backup")) {
        appendFileContent(fullPath);
      }
    }
  });
}

// 파일 경로와 내용을 'source_codes' 파일에 추가
function appendFileContent(filePath) {
  const fileContent = fs.readFileSync(filePath, "utf-8");
  const formattedContent = `\n<${filePath}>\n\n${fileContent}\n\n`;
  fs.appendFileSync(outputFilePath, formattedContent);
}

// 기존 'source_codes' 파일이 있다면 삭제
if (fs.existsSync(outputFilePath)) {
  fs.unlinkSync(outputFilePath);
}

// 현재 디렉토리에서 시작, node_modules, logs, .next 디렉토리 제외
scanDirectory(__dirname, [
  "node_modules",
  "logs",
  ".next",
  "public",
  "images",
  "build",
  "public",
]);
```

Claude 에 source_codes 파일 첨부하고 코드 개선 시작하기

```
</workspace/BootcampChat/README.md> # 실시간 채팅 애플리케이션 이 프로젝트는 실시간 채팅을 위한 웹 애플리케이션으로, Node.js, Next.js, Redis, MongoDB를 활용하여 구현되었습니다. ## 주요 기능 ## 1. 실시간 채팅 2. 사용자 목록 3. 채팅 내역 4. 사용자 프로필 5. 채팅방 생성 및 사용자
```

PASTED

WR 현재 채팅방에서 스크롤을 맨위로 올리면 과거 메시지들이 추가로 로딩되는데, 이 때 해당 위치에 스크롤이 그대로 있어야 과거 메시지들을 탐색할 수 있어. 그런데 새로운 메시지가 로딩되어서 그런지 스크롤이 가장 하단으로 자동으로 내려가 버려. 그리고 스크롤을 맨 위로 올렸을 때 과거 채팅 내역이 추가되는 부분이 동작하지 않고 있어.

이 문제를 차근차근 분석하여 가장 현명하고 깔끔하게 해결해줘.

아래는 채팅방에서의 스크롤 동작이야.

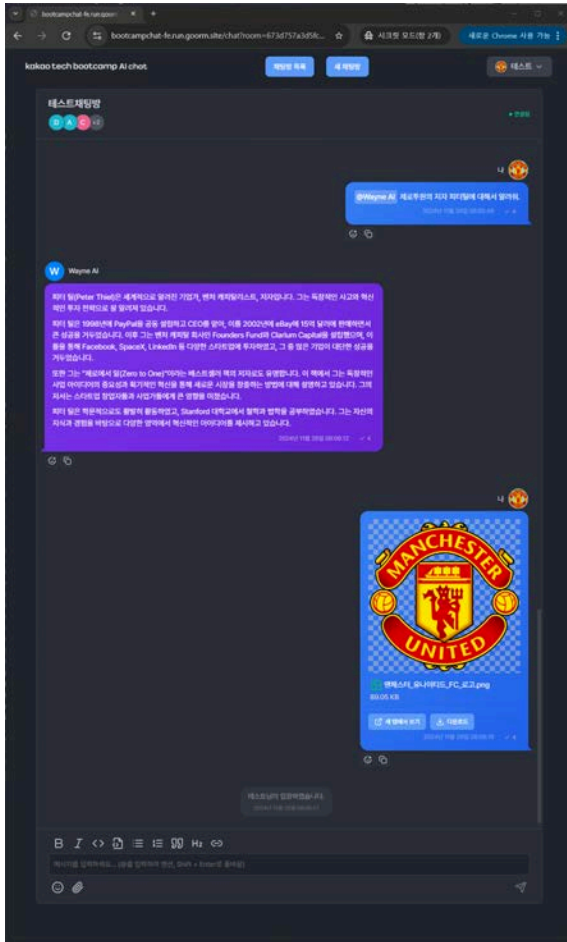
1. 스크롤바가 맨 위로 올라가면 과거 채팅 내역을 추가 로딩하고, 스크롤바는 해당 위치에 그대로 위치 (사용자 입장에서 보이는 부분은 그대로)
2. 내가 새로운 대화내용을 입력했을 때는 무조건 항상 스크롤을 맨 아래로 이동
3. 스크롤이 맨 아래로 고정되어 있는 상태에서는 상대방의 대화 내용이 새로 올라왔을 때도 항상 스크롤을 맨 아래로 이동시키기
4. 스크롤이 맨 아래인 상태가 아닐 때는 상대방의 대화 내용이 새로 올라와도 스크롤을 이동시키지 않음
5. 다른 참여자가 아닌 내가 AI Agent를 멘션해서 새로운 AI Message가 나오는 경우에는 스크롤을 맨 아래로 이동시킴



4단계 : 보정 및 마무리

사실 Jobgorithm은 3단계 수준에서 충분히 개발이 마무리 되었습니다.

더 복잡한 채팅 앱 개발 예시 - SW스택, 기능 확장과 복잡성 증가



- 실시간 통신 기능 구현 (Socket.io 사용)
 - 사용자 간 메시지를 실시간으로 주고받을 수 있도록 Socket.io를 활용하여 양방향 통신 시스템 구축
- 사용자 인증 및 관리 체계 구축
 - JWT(Json Web Token)를 사용하여 사용자 인증과 세션 관리를 구현하여 보안성 강화
- 멀티미디어 전송 기능 추가
 - 채팅 내에서 이미지, 파일 등의 멀티미디어 콘텐츠 전송 기능 추가
- 채팅방 생성 및 그룹 채팅 기능 구현
 - 사용자가 직접 채팅방을 생성하고, 다수의 참여자와 함께하는 그룹 채팅 지원
- 멘션 기능 구현
 - 채팅 중 특정 사용자를 @아이디 형태로 멘션하여 상대방에게 알림을 보내는 기능 추가
- Redis 도입
 - Redis를 활용하여 세션 관리, 실시간 데이터 캐싱, 메시지 큐 처리 등을 통한 성능 최적화
- 읽음 처리 기능 추가
 - 상대방이 메시지를 읽었는지 확인할 수 있는 읽음 확인 기능 추가로 사용자 경험 향상
- 이모지 리액션 기능 구현
 - 메시지에 대한 반응을 표현할 수 있는 이모지 리액션 기능 추가
- 프로필 이미지 변경 기능 구현
 - 사용자가 프로필 이미지를 업로드 및 변경할 수 있어 개인화된 계정 제공
- 회원 관리 체계 구축
 - 관리자 전용 페이지에서 회원 정보 관리, 권한 설정, 계정 제재 등의 기능 수행
- 중복 로그인 방지 기능 구현
 - 동일한 계정으로 여러 곳에서 동시에 로그인할 수 없도록 중복 로그인 방지 기능 추가로 보안성 강화

실시간 데모: 채팅 앱 주요 기능 시연

- 데모 내용

- 여러 사용자가 실시간으로 채팅하는 모습 시연
- 이미지 및 파일 전송 기능 확인
- 새 채팅방 생성 및 그룹 채팅 기능 소개
- 사용자 로그인 및 프로필 관리 기능 시연

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

1. 명확하고 구체적으로 지시하기

- 구체적 언어 사용으로 명확한 결과 전달
- 모호한 표현 배제 및 명확한 요구 사항 제시
- 예상 결과물의 형태와 내용 구체화
- 시각적 예시를 통한 결과물 제시
- 작업 우선순위 명시로 효율적 진행
- 필요 리소스 및 도구 사전 명시

회원가입 페이지의 입력 필드는 다음과 같이 구성해줘:

- 이메일 필드: 이메일 형식 검증 포함
- 비밀번호 필드: 최소 8자, 영문/숫자/특수문자 포함
- 비밀번호 확인 필드: 실시간 일치 여부 확인
- 전화번호 필드: 자동 하이픈 추가 및 숫자만 입력 가능
- 프로필 이미지: 5MB 이하의 jpg, png 파일만 허용

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

2. 단계별로 요청하기

- 복잡한 작업의 단계별 분할 요청
- 각 단계의 목표와 기대 결과 명확화
- 단계별 필요한 리소스와 도구 명시
- 완료 기준 명확화로 진행 상황 추적
- 단계별 피드백 제공으로 지속적 개선
- 단계 간 의존성 명확화로 작업 흐름 최적화

채팅 기능 구현을 다음 단계로 진행해줘:

1. Socket.io를 사용한 실시간 연결 구현
2. 채팅방 생성 및 참여 기능 추가
3. 메시지 전송 및 수신 기능 구현
4. 읽음 처리 및 알림 기능 추가
5. 이전 채팅 내역 로딩 기능 구현

각 단계별 구현 결과를 테스트할 수 있게 해줘

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

3. 필요한 조건과 제약사항 명시

- 필요 기술 스택 명확히 명시
- 사용할 라이브러리와 버전 구체적 지정
- 프로젝트 제약사항 명확히 하여 개발 방향 설정
- 환경 설정과 초기 설정 명시
- 각 기술 스택의 역할과 사용 목적 설명
- 기술적 제약사항 고려하여 현실적 목표 설정

다음 기술 스택으로 프로젝트를 설정해줘:

- Frontend: React 18.2.0, TypeScript 5.0
- Backend: Node.js 20.x, Express 4.18
- Database: MongoDB 7.0
- 상태관리: Redux Toolkit
- UI 라이브러리: Material-UI v5

package.json에 필요한 의존성도 모두 포함해줘

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

4. 결과물의 형식 지정

- 결과물 형태와 내용 명확히 지정
- 전체 코드, 특정 파일, 풀 버전 등 출력 형태 명시
- 결과물 포맷과 구조 구체적 설명
- 결과물 사용 목적과 기대 효과 명시
- 검토 기준과 피드백 방법 설명
- 배포 및 사용 방법 명시

사용자 인증 관련 코드를 다음 형식으로 출력해줘:

1. /src/api/auth 폴더의 모든 파일 내용
2. /src/components/auth 컴포넌트 코드
3. Redux 관련 auth slice 코드
4. 관련 타입 정의 파일

각 파일의 전체 내용을 주석 포함해서 출력해줘

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

5. 에러 메시지 제공 및 해결 요청

- 발생한 에러 메시지 명확히 제공
- 에러 발생 상황과 관련 코드 구체적 설명
- 에러 해결 위한 구체적 방법 요청
- 해결 후 기대되는 결과 명시
- 참고 자료나 링크 제공 가능
- 해결 후 테스트 방법과 검증 기준 설명

빌드 시 다음 에러가 발생해:

ERROR in ./src/components/Chat.tsx 42:38

TS2339: Property 'socket' does not exist on type 'DefaultEventsMap'.

관련 코드:

```
socket.on('message', (data: Message) => {  
  setMessages(prev => [...prev, data]);  
});
```

@types/socket.io-client 최신 버전 사용 중

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

6. 구체성의 원칙

- 추상적 지시보다 구체적 요구 사항 전달
- 구체적 요구 사항으로 AI 이해도 향상
- 예시를 통한 원하는 결과 명확히 제시
- 구체적 요구 사항으로 작업 효율성 증대
- 작업 일관성 유지
- 구체적 요구 사항으로 작업 방향 명확화

데이터 테이블 컴포넌트에 다음 기능을 추가해줘:

- 페이지당 행 개수: 10, 20, 50개 선택 가능
- 정렬: 각 컬럼별 오름차순/내림차순
- 필터: 날짜 범위, 상태값, 검색어
- 선택된 행 하이라이트: #e3f2fd 배경색
- 반응형: 768px 미만에서 스크롤 적용

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

7. 단계적 접근

- 복잡한 문제의 단계별 해결 방식 채택
- 각 단계의 목표와 기대 결과 명확화
- 단계별 필요한 리소스와 도구 명시
- 완료 기준 명확화로 진행 상황 추적
- 단계별 피드백 제공으로 지속적 개선
- 단계 간 의존성 명확화로 작업 흐름 최적화

실시간 채팅 기능 리팩토링을 단계별로 진행하자:

1. 1단계: 현재 코드 분석
2. Socket.io 이벤트 중복 처리 문제
3. 메모리 누수 (useEffect cleanup 미처리)
4. 타입 정의 부재
5. 2단계: 기본 구조 개선
6. Socket 커스텀 훅 분리
7. 이벤트 타입 정의
8. 클린업 함수 구현
9. 3단계: 성능 최적화
10. 메시지 캐싱 전략 수립
11. 무한 스크롤 구현
12. 이미지 레이지 로딩

각 단계별로 리뷰하고 다음 단계로 넘어가자

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

8. 피드백 루프 활용

- AI 제공 결과물의 검토와 지속적 피드백
- 결과물의 개선 사항 명확히 전달
- 피드백을 통한 결과물의 품질 향상
- 결과물의 검토 기준과 피드백 방법 설명
- 피드백을 통한 결과물의 일관성 유지
- 피드백을 통한 결과물의 효율성 증대

방금 생성한 API 코드 리뷰 결과야:

- 보안 이슈:
 - JWT secret이 하드코딩됨
 - 비밀번호 해시화 전 검증 누락
 - rate limiting 미적용
- 성능 이슈:
 - 데이터베이스 인덱스 미설정
 - 불필요한 중첩 쿼리 존재
- 코드 품질:
 - 에러 처리 일관성 부족
 - 타입 정의 불명확

이 문제들을 우선순위대로 수정해줘

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

9. 모호한 표현 지양

- 모호한 표현 배제 및 명확한 지시로 대체
- 명확한 지시로 작업의 효율성 증대
- 명확한 지시로 작업의 일관성 유지
- 명확한 지시로 작업 방향 명확화
- 명확한 지시로 작업의 품질 향상
- 명확한 지시로 작업의 효율성 증대

❌ 로딩 상태를 적당히 표시해줘

❌ 가능하면 에러 처리도 해줘

○ 다음 상황에서 로딩 스피너를 표시해줘:

- API 요청 시작부터 응답 수신까지
- 이미지 로딩 중
- 무한 스크롤 데이터 요청 시

○ try-catch로 에러 처리하고 다음 에러 타입별로 사용자에게 알림 표시:

- NetworkError: 네트워크 연결 확인
- AuthError: 세션 만료로 재로그인 필요
- ValidationError: 입력값 검증 실패

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

10. 과도한 정보 전달 자제

- 혼란 방지를 위한 적절한 정보량 분할 전달
- 정보량 분할로 작업의 효율성 증대
- 정보량 분할로 작업의 일관성 유지
- 정보량 분할로 작업 방향 명확화
- 정보량 분할로 작업의 품질 향상
- 정보량 분할로 작업의 효율성 증대

❌ 인증, 채팅, 게시판, 알림 기능을 모두 한번에 구현하고 싶어. 각 기능별로 실시간 처리가 필요하고 데이터베이스 스키마는...

○ 먼저 사용자 인증 기능부터 구현하자:

1. 회원가입 API 엔드포인트 생성
2. 비밀번호 암호화 및 유효성 검사
3. JWT 토큰 발급 및 관리

이 부분 완성되면 다음 기능으로 넘어가자

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

11. 일관성 유지

- 용어, 스타일, 언어 등의 일관된 사용
- 일관된 사용으로 작업의 효율성 증대
- 일관된 사용으로 작업의 일관성 유지
- 일관된 사용으로 작업 방향 명확화
- 일관된 사용으로 작업의 품질 향상
- 일관된 사용으로 작업의 효율성 증대

❌ 섞여 있는 코드 스타일:

```
function getUser() { ... }  
const createPost = () => { ... }  
async function fetchData() { ... }  
const getData = async () => { ... }
```

○ 일관된 코드 스타일로 변경:

```
const getUser = async (): Promise<User> => { ... }  
const createPost = async (): Promise<Post> => { ... }  
const fetchData = async (): Promise<Data> => { ... }  
const getData = async (): Promise<Data> => { ... }
```

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

12. 현실적인 요구

- 기술적 한계를 고려한 현실적 범위 내 요청
- 현실적 요청으로 작업의 효율성 증대
- 현실적 요청으로 작업의 일관성 유지
- 현실적 요청으로 작업 방향 명확화
- 현실적 요청으로 작업의 품질 향상
- 현실적 요청으로 작업의 효율성 증대

✗ GPT-4 API를 사용해서 실시간으로 모든 채팅 내용을 10개 언어로 번역하고, 감정 분석해서 이모지로 표현해줘

○ 다국어 지원을 위한 현실적인 접근:

1. 초기에는 한국어/영어 두 언어만 지원
2. i18n 설정으로 정적 텍스트 번역
3. 중요 알림만 선택적 번역 API 사용
4. 사용자 설정 언어로 UI 표시

이후 사용량과 비용을 고려해 확장하자

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

13. 버그 해결 요청 시

- 정확한 에러 메시지 제공으로 문제 설명
- 에러 발생 상황과 관련 코드 구체적 설명
- 에러 해결 위한 구체적 방법 요청
- 해결 후 기대되는 결과 명시
- 참고 자료나 링크 제공 가능
- 해결 후 테스트 방법과 검증 기준 설명

React 18에서 다음 워닝이 발생해:

Warning: Cannot update a component while rendering a different component. To locate the bad setState call inside `ChildComponent`, follow the stack trace as described in {url}

문제되는 코드:

```
function ParentComponent() {  
  const [data, setData] = useState<Data[]>([]);  
  return <ChildComponent setData={setData} />;  
}  
  
function ChildComponent({ setData }) {  
  useEffect(() => {  
    setData(prevData => [...prevData, newData]);  
  }, [setData]);  
  // ...  
}
```

React 18의 새로운 동시성 모드를 고려한 해결 방법 제시 부탁해

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

14. 기능 추가 요청 시

- 추가 기능의 목적과 기대 결과 명확히 설명
- 구현 방법과 필요한 리소스 명시
- 완료 기준 명확화로 진행 상황 추적
- 테스트 방법과 검증 기준 설명
- 배포 및 사용 방법 명시
- 피드백 방법과 개선 방안 설명

게시판에 다음 기능을 추가해줘:

- 현재 구현된 기능:
 - 기본적인 CRUD
 - 페이지네이션
 - 검색
- 추가할 기능:
 - 게시글 임시저장 (localStorage 활용)
 - 마크다운 에디터 (react-markdown 사용)
 - 이미지 첨부 (drag & drop)
 - 댓글 알림 (SSE 활용)
- 기술 요구사항:
 - TypeScript strict 모드 활성화
 - 상태관리는 React Query 사용
 - 테스트 코드 포함 (Jest + RTL)

각 기능별 구현 계획과 코드 구조를 먼저 설명해줘

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

15. 디자인 수정 요청 시

- 원하는 디자인 요소와 위치 구체적 지시
- 디자인 수정의 목적과 기대 결과 명확히 설명
- 구현 방법과 필요한 리소스 명시
- 완료 기준 명확화로 진행 상황 추적
- 테스트 방법과 검증 기준 설명
- 피드백 방법과 개선 방안 설명

대시보드 UI 개선이 필요해:

- 현재 문제점:
 - 모바일에서 차트가 잘려보임
 - 데이터 테이블 스크롤이 불편함
 - 로딩 상태 표시가 불명확
- 개선 요구사항:
 - 반응형 레이아웃 (breakpoints: 768px, 1024px)
 - 차트는 모바일에서 카드 형태로 전환
 - 스켈레톤 UI로 로딩 상태 표시
 - 다크모드 지원
- 디자인 시스템:
 - Colors: primary #1976d2, secondary #dc004e
 - Typography: Pretendard 폰트 사용
 - Spacing: 8의 배수 단위
 - Shadow: 0 2px 4px rgba(0,0,0,0.1)

Material-UI v5 기준으로 수정해줘

더 효율적으로 개발하기 위한 생성형AI 프롬프트 작성 방법

16. 코드 수정 및 개선 요청 시

- 수정 필요 부분과 개선 사항 명확히 전달
- 수정 및 개선의 목적과 기대 결과 명확히 설명
- 구현 방법과 필요한 리소스 명시
- 완료 기준 명확화로 진행 상황 추적
- 테스트 방법과 검증 기준 설명
- 피드백 방법과 개선 방안 설명

현재 인증 관련 코드의 문제점을 해결해줘:

- 현재 코드의 문제:
 - src/hooks/useAuth.ts
 - 토큰 갱신 로직이 불안정함
 - 새로고침 시 인증 상태 유실
 - 중복 요청 처리 안됨
- 요구사항:
 - axios interceptor로 토큰 갱신 자동화
 - refresh token은 httpOnly 쿠키로 관리
 - 요청 중복 방지를 위한 큐잉 처리
 - 로그아웃 시 모든 탭에서 동기화
- 고려사항:
 - SSR 환경 호환성
 - 보안 취약점 방지
 - 테스트 용이성

각 문제별 해결 방안을 코드와 함께 설명해줘

patch.js 만들어 쓰기

작성 과정

- 생성형 AI에게 스크립트 작성 요청
- 필요한 기능 명확히 정의
- 테스트 케이스 준비
- 실행 및 결과 검증

주요 기능

- 대규모 코드베이스 일괄 수정
- 특정 패턴 검색 및 치환
- 코드 스타일 통일
- 의존성 업데이트 자동화

유용성

- 반복 작업 시간 단축
- 일관성 있는 변경 보장
- 실수 방지 및 품질 향상
- 변경 이력 추적 용이

활용 예시

- API 엔드포인트 일괄 변경
- 컴포넌트 이름 규칙 통일
- deprecated 코드 자동 업데이트
- 코드 포매팅 자동화

E2E 테스트 자동화

작성 과정

- 테스트 시나리오 정의
- 생성형 AI로 테스트 코드 작성
- 테스트 데이터 준비
- CI/CD 파이프라인 연동

주요 기능

- 사용자 시나리오 기반 테스트
- 크로스 브라우저 테스트
- 성능 및 로딩 테스트
- 에러 케이스 자동 검출

유용성

- QA 시간 대폭 단축
- 회귀 테스트 자동화
- 실제 사용자 환경 검증
- 배포 전 안정성 확보

활용 예시

- 로그인/회원가입 플로우
- 결제 프로세스 검증
- 데이터 CRUD 테스트
- 반응형 UI 테스트

<https://bootcampchat-e2e-report.run.goorm.site/#?>

정리

생성형 AI로 서비스 개발

- 1~4단계: 그림을 그리듯 활용
- 각 단계마다 적합한 LLM 사용

기술적 이해

- 컨텍스트 윈도우의 중요성 이해
- HTML/CSS/JS 및 웹 개발 기본 이해 필요

환경 및 도구 활용

- 개발/실행 환경에 Arkain 적극 활용
- 필요하면 만들어 쓰자는 접근

프롬프트 작성 팁

- 명확하고 구체적인 지시로 원하는 결과 전달
- 단계별 요청으로 복잡한 작업 체계적 처리
- 필요한 조건과 제약사항 명시
- 결과물의 형식 지정으로 일관성 유지
- 에러 메시지 제공 및 해결 요청
- 구체성의 원칙으로 AI 이해도 향상

생성형 AI의 가능성

- 복잡한 서비스도 생성형 AI만으로 구현 가능
- Arkain에서의 LLM 역할에 대해 고민해볼 수 있는 좋은 경험

과연 더 효율적일까? (Jobgorithm 개발 기준)

프롬프트

- 약 600줄의 라인수
- 글자수는 32,000자
- 약 8,000토큰 (한글이 섞여 있음)

장점

- 진정한 멀티태스킹 - 결과 생성 중 다른 업무 가능
- 자연어로 정리하고 소통 가능
- 검색 및 수정 시간 최소화
- 코드 스타일 통일 및 코드 품질 향상
- 예외 케이스 최소화
- 디자인 시스템 적용 및 의존성 업데이트 자동화
- 손쉬운 리팩토링

생성된 코드

- 약 1,300줄의 라인수
- 글자수는 75,000자
- 약 19,000토큰 (영어 기준 1토큰 글자)

여전한 문제 (Arkain이 해소해주길!)

- 반복적인 복사-붙여넣기
- 환경 구성은 직접 해야함
- 할루시네이션
- 유사 기능 코드의 생성
- 코드 양이 비대해짐

생성형 AI와 일하는 방법

효율적으로 일하며 삶의 질을 올리기

사람이 할 일

- 문제를 창의적으로 해결하기
- 개발 도구와 환경 준비하기
- 명확한 프롬프트 작성하기
- 결과물 검증과 피드백하기



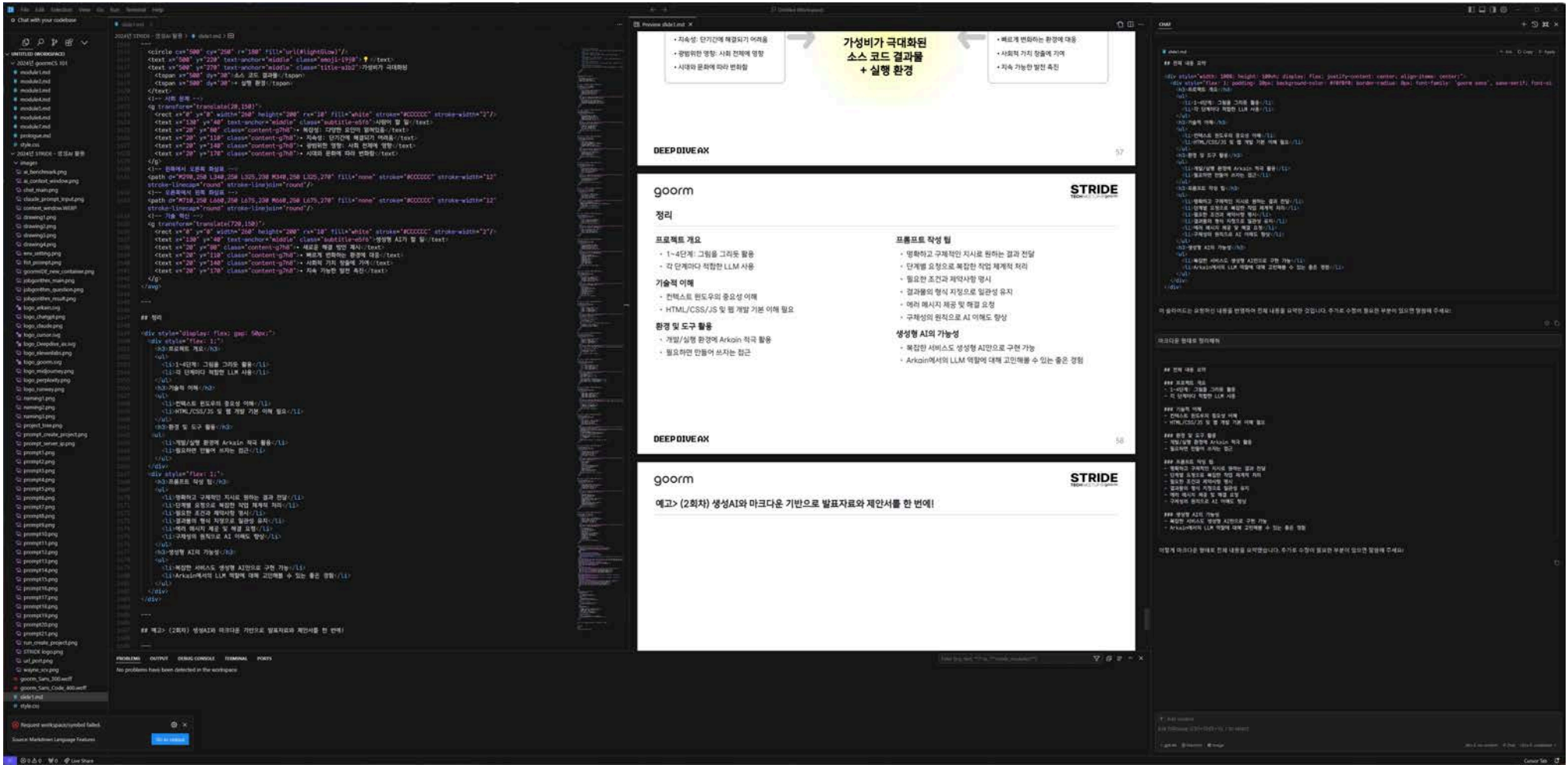
**가성비가 극대화된
소스 코드 결과물
+ 실행 환경**



생성형 AI가 할 일

- 기본 코드 구조 생성하기
- 코드 최적화와 버그 수정
- 다양한 구현 방안 제시
- 문서와 테스트 자동화

예고> (2회차) 생성AI와 마크다운 기반으로 발표자료와 제안서를 한 번에!



감사합니다!

Q&A

질문이 있으시면 자유롭게 해주세요.