

캡스톤 디자인 I

종합설계 프로젝트

프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스
팀 명	노페인 노게인
문서 제목	결과보고서

Version	1.6
Date	2024-MAY-23

팀원	김지성 (조장)
	김선우
	이소정
	민경서
	강희구

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

CONFIDENTIALITY/SECURITY WARNING

이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 “비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 코드 생성 서비스”를 수행하는 팀 “노페인 노게인”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “노페인 노게인”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

문서 정보 / 수정 내역

Filename	15조_비기너게인_수행결과보고서.doc
원안작성자	김지성, 김선우, 민경서, 이소정, 강희구
수정작업자	김지성, 김선우, 민경서, 이소정, 강희구

수정날짜	대표수정 자	Revisio n	추가/수정 항목	내 용
2024-05-12	강희구	1.0	최초 작성	개요 작성
2024-05-16	강희구	1.1	추가	백엔드 연구/개발 내용 추가 작성
2024-05-17	민경서	1.2	추가	AI 챗봇 연구/개발 내용 추가 작성, 활용방안 작성
2024-05-19	강희구	1.3	추가	시스템 기능/비기능 요구사항 작성
2024-05-19	김지성	1.4	추가	프론트엔드 연구/개발 내용 추가 작성
2024-05-20	이소정	1.5	추가	프론트엔드 연구/개발 내용 추가 작성
2024-05-22	김선우	1.6	추가	AWS, 챗봇 및 백엔드 개발 내용 추가 작성

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

목 차

1	개요	4
1.1	프로젝트 개요	4
1.2	프로젝트 선정 동기 및 추진 배경	4
	1.3 프로젝트 필요성 및 목적	4
2	개발 내용 및 결과물	5
2.1	목표	5
2.2	연구/개발 내용 및 결과물	6
2.2.1	연구/개발 내용	6
2.2.2	시스템 기능 요구사항	6
2.2.3	시스템 비기능 요구사항	6
2.2.4	시스템 구조 및 설계도	6
2.2.5	활용/개발된 기술	6
2.2.6	협업 내용	
2.2.7	결과물 목록	
2.3	기대효과 및 활용방안	6
3	자기평가	7
4	참고 문헌	7
5	부록	7
5.1	사용자 매뉴얼	7
5.2	테스트 케이스	7

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

1 개요

1.1 프로젝트 개요

프로젝트 명 : 비기너게인(BeginnerGain) - 초보개발자들을 위한 보일러 플레이트 코드 생성 서비스

프로젝트 시작일 및 종료일 : 2024.03.01 - 2024.05.17

초보개발자들을 위한 보일러 플레이트(boilerplate) 생성 서비스 :

보일러 플레이트 코드는 특정 프로그래밍 언어나 기술에서 반복적으로 사용되는 코드를 말합니다. 보일러 플레이트는 코드의 재사용을 가능하게 하여 개발의 효율성을 높이고, 표준화된 패턴이나 프로세스를 쉽게 구현할 수 있게 해 줍니다. 또한, 코드의 일관성을 유지할 수 있게 해 주어 유지 보수에 용이 하며 서비스 생산성을 향상시켜 줍니다.

몇 번의 질문과 간단한 클릭으로 사용할 기술 스택을 선택하여 초보 개발자들이 프로젝트의 기본 구조를 세팅할 수 있는 환경을 제공하는 것이 저희 프로젝트의 목표입니다. 서비스 목표 대상을 ‘초보 개발자’로 선정하여, 초보 개발자들이 쉽게 사용할 수 있도록 서비스를 간편하고 직관적인 UI와 플로우로 구성하였습니다. 또한, 기술 스택에 대한 지식이 부족하여 어떤 기술을 선택해야 할지 결정에 어려움을 겪는 초보 개발자들을 위해 Prompt Engineering 된 인공지능 모델 기반 대화형 챗봇이 여러 조언을 줍니다.

사용자는 자신이 원하는 프로젝트 환경에 대한 몇 번의 응답을 통해 요구사항을 전달하기만 하면 이에 맞는 보일러 플레이트(boilerplate)를 제공받을 수 있습니다. 또한 서비스 사용자는 제공받은 보일러플레이트 코드를 보관함에 저장하여 필요할 때마다 재사용할 수 있습니다.

이 서비스를 통해 개발자들은 동일한 코드를 재작성하는 등의 반복적인 작업에 소요되는 시간을 절약하여 개발 시간을 단축할 수 있고 일관된 코드를 작성할 수 있습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

1.2 프로젝트 선정 동기 및 추진 배경

보일러 플레이트 코드는 특정 프로그래밍 언어나 기술에서 반복적으로 사용되는 코드들을 말합니다. 보일러 플레이트 코드는 개발의 효율성을 높이고, 표준화된 패턴이나 프로세스를 쉽게 구현할 수 있게 해 줍니다.

실제 개발 과정에서는 공통적으로 많이 사용되는 프레임워크나 라이브러리들이 있습니다. 많은 개발자들이 여러 프로젝트를 진행하는 경우에, 동일한 라이브러리를 사용하며 같은 코드를 다시 여러 번 작성하는 일이 많습니다. 보일러 플레이트 코드는 여러 개발에서 자주 사용되며, 이미 테스트되고 검증된 코드이기 때문에, 보일러 플레이트 코드를 사용하여 개발을 진행하면, 개발자가 매번 같은 코드를 새로 작성할 필요가 없습니다. 이는 특히 프로젝트 초기 설정 단계에서 많은 시간을 절약해 줍니다.

또한, 초보 개발자의 경우 여러 라이브러리나 프레임워크의 장단점을 명확하게 알지 못하는 경우가 많습니다. 그렇기에 프로젝트를 시작할 때 내 프로젝트에 어떤 기술이 적합한지, 어떤 기술 조합을 사용해야 할 지 선택하지 못합니다. 기술을 선정하더라도 기술들을 내 프로젝트에 적용할 때 큰 어려움을 겪습니다. 이 때문에 프로젝트 시작부터 많은 시간과 자원을 소모하게 됩니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22


[stackoverflow](#)
[About](#)
[Products](#)
[OverflowAI](#)

OverflowAI is here! AI power for your Stack Overflow for Teams knowledge community. [Learn more](#)

[Home](#)
[Questions](#)
[Tags](#)
[Users](#)
[Companies](#)

LABS
[Jobs](#) NEW
[Discussions](#)

COLLECTIVES +
 Communities for your favorite technologies.
[Explore all Collectives](#)

TEAMS

How to decide which framework to use for frontend?

Asked 4 years, 5 months ago Modified 4 years, 5 months ago Viewed 1k times

I am trying to develop web application but I can't decide which framework to choose for front-end. I could use Vue.js, Angular, React or Vanilla.js.

1

What are the parameters we need to consider while choosing front-end technology?

frontend web-frontend

Share Improve this question Follow

asked Nov 27, 2019 at 10:18
nononono
13 6

Add a comment

2 Answers

Sorted by: Highest score (default)

[그림1] 프레임워크 선택에 어려움을 겪는 경우

What is the difference between Vue JS and React JS? [closed]

Asked 4 years, 11 months ago Modified 3 years, 6 months ago Viewed 2k times

Closed. This question needs [details or clarity](#). It is not currently accepting answers.

Want to improve this question? Add details and clarify the problem by [editing this post](#).
 Closed 4 years ago.

Improve this question

I have started studying Vue Js and React Js, I searched on google could not find satisfactory answers for comparison between both.

I found What is Vue js and another Question What is React Js. But there is no question which gives the comparison of both.

javascript reactjs vue.js

[그림2] 프레임워크의 차이를 알 지 못하는 경우

이러한 이유들을 토대로, 저희는 보일러 플레이트 코드 생성 서비스를 통해

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

개발자들이 프로젝트를 빠르게 시작할 수 있도록 하고, 초보 개발자들이 쉽고 간편하게 자신의 프로젝트에 적합한 프레임워크와 라이브러리를 찾을 수 있도록 도움을 주고자 이번 프로젝트 주제를 선정하게 되었습니다.

1.3 프로젝트 필요성 및 목적

개발 경험이 적은 초보 개발자들이 새로운 프로젝트를 시작할 때 기본적인 프로젝트 구조와 주로 사용되는 보일러 플레이트 코드를 제공해줌으로써 개발의 진입장벽을 낮춰주는 것이 저희 프로젝트의 목적입니다.

구체적인 목적은 다음과 같습니다.

사용할 프레임워크/라이브러리 추천 : 초보 개발자는 프로젝트를 시작할 때 어떤 프레임워크를 사용해서 개발을 해야할 지 모르는 경우가 많거나 프레임워크들을 알고 있더라도 그 차이를 알 지 못하여 선택에 어려움을 겪는 경우가 많습니다. 그래서 개발을 처음 시작하는 개발자의 목적에 따라 최적화된 프레임워크를 추천해주는 서비스가 필요합니다.

빠른 프로젝트 시작 : 제공받은 보일러 플레이트 코드를 사용하면 초보 개발자는 프로젝트 초기 설정에 많은 시간을 들이지 않고 바로 개발을 시작할 수 있습니다. 특히 새로운 프레임워크나 기술 스택을 배울 때 초보개발자가 기술적 장벽을 극복할 수 있도록 도움을 줍니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

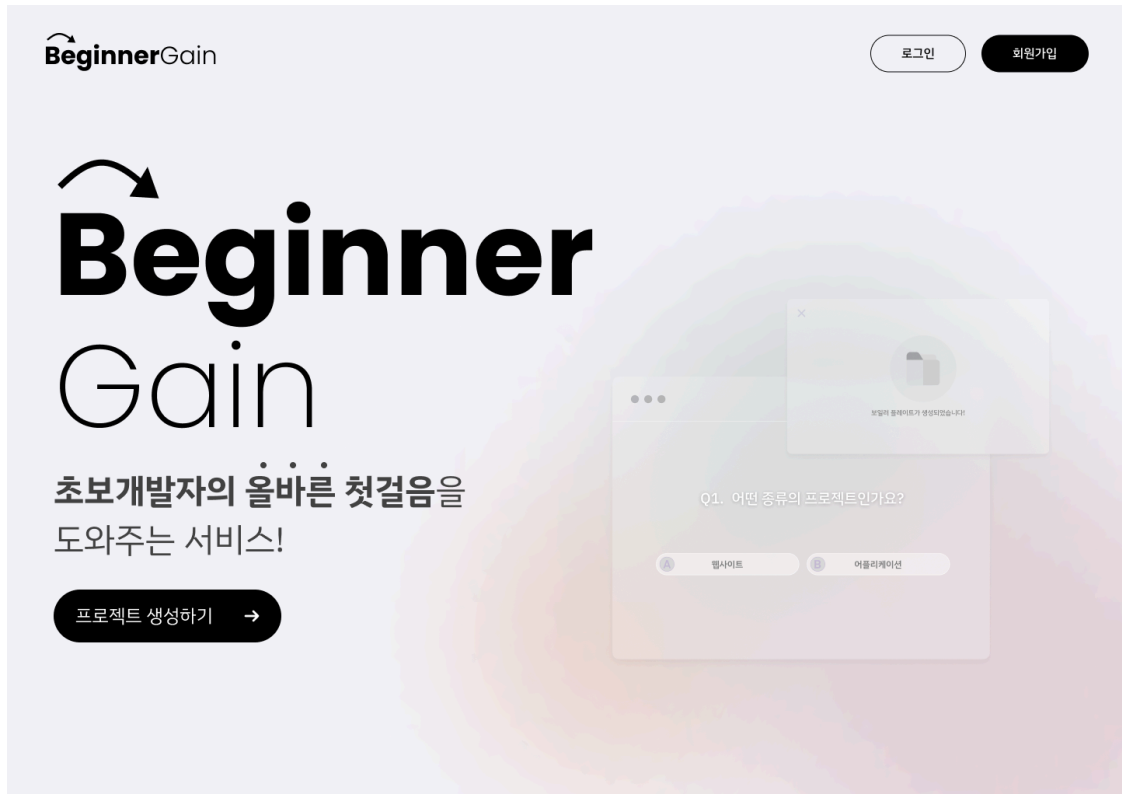
표준 코드 학습 : 보일러 플레이트 코드는 일반적으로 많은 곳에서 사용되는 표준 형식을 따르고 있기 때문에, 초보 개발자들은 제공받은 보일러 플레이트 코드를 통해 어떻게 코드를 작성하는 지에 대해 학습할 수 있습니다. 이는 또한 초보 개발자들이 좋은 코딩 습관을 들일 수 있도록 도움을 줄 수 있습니다.

기존 유사 서비스의 아쉬움 : 기존 유사 서비스는 사용자가 개발 목적과 개발 요구 사항을 질문하며 기술 추천을 요구하면, 여러 기술들을 나열하고 각 기술들의 특징을 설명해주는 방식입니다. 하지만 이를 개발 경험이 적은 초보 개발자가 보았을 때는 각 선택지의 장단점을 정확히 이해하고 비교하기가 어려워 선택에 도움이 되기 어렵습니다. 저희의 챗봇 서비스는 질문을 통해 사용자의 요구를 정확히 파악하여 이에 최적화된 하나의 선택지를 제공해줍니다. 이를 통해 인공지능 기반의 챗봇 서비스를 통해 사용자의 요구를 파악하여 진행하려는 프로젝트에 알맞은 기술을 추천해 줍니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

2 개발 내용 및 결과물

2.1 목표



[그림3] '비기너게인' 시작화면

이번 프로젝트의 핵심 목표는 사용자의 요청 사항을 토대로 사용자에게 최적화된 프로젝트 시작 환경을 제공해주는 서비스를 개발하는 것입니다. 사용자가 프로젝트를 쉽고 빠르게 시작할 수 있도록 보일러 플레이트 코드 생성까지의 플로우를 단순화하였습니다. 또한, 기술 선택에 어려움을 겪는 사용자를 돕기 위한 AI 챗봇은 프로젝트의 목적과 사용자의 요구사항에 맞는 최적화된 답변을 제공할 수 있도록 Prompt Engineering 되었습니다.

이 AI 챗봇은 현재 대중적으로 많이 사용되고 있는 챗봇인 Chat GPT의 단점을 해소하고, 차별성을 기르는 것을 목표로 하여, GPT와는 다른 형식으로 답변을 주는 것에 초점을 맞추어 개발이 진행되었습니다.

저희의 프로젝트는 위 서비스를 통해 개발의 진입장벽을 낮추고, 초보개발자에서부터 경력이 있는 개발자 모두 개발 과정에서의 시간과 불필요한 노력을 절약할 수 있도록

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

드립니다.

2.2 연구/개발 내용 및 결과물

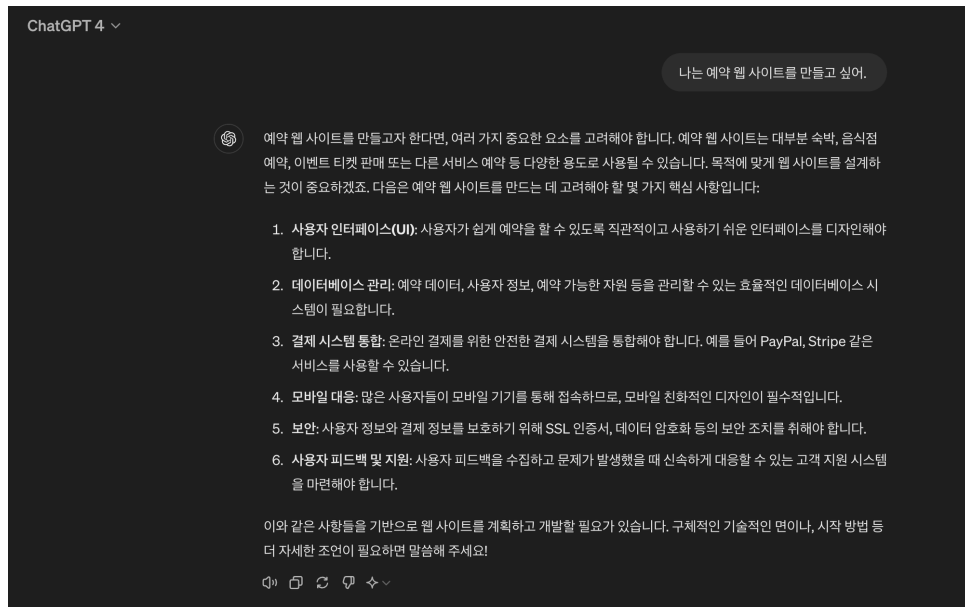
2.2.1 연구/개발 내용

2.2.1.1 AI 챗봇

비기너게인은 초보 개발자들을 위해 보일러플레이트를 생성해주는 서비스를 제공합니다. 보일러 플레이트를 제공하기 전, 사용자에게 원하는 기술스택과 사용할 라이브러리 등 요청사항을 전달받아야 합니다. 하지만, 초보 개발자들 혹은 새로운 기술 스택을 공부하려는 개발자들은 어떤 것을 선택할 지에 대해 어려움을 겪을 수 있습니다. 그래서 저희는 이번 프로젝트의 목적에 최적화된 답변을 출력하도록 ChatGPT Prompt Engineering 을 활용하여 초보 개발자에게 최적화된 선택지를 제공하는 AI 챗봇을 구축했습니다. 저희는 chatgpt-3.5-turbo 모델을 사용했습니다. 또한, 저희의 웹 서비스에서 사용하는 챗봇에서 더 나아가 저희가 구축한 챗봇을 ChatGPT Store에 배포하였습니다.

개발 목적에 맞게 프롬프트 엔지니어링 : 기존의 ChatGPT는 사용자가 기술 스택 추천을 요청하면 각 기술 스택의 특징을 나열하고 결정은 사용자에게 맡깁니다. 이러한 방식은 사용자가 기술에 대한 충분한 이해가 없는 경우에는 사용자의 문제를 해결하기 어렵다는 아쉬운 점이 있습니다. 아래 사진에서의 예시처럼 ChatGPT에게 질문을 하면, 일반적인 내용에 대한 답을 주고 초보 개발자가 원하는 정확하고 구체적인 답변을 주지 못해 사용자는 ChatGPT를 통해 문제를 해결하지 못하는 문제가 있습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22



[그림4] ChatGPT 답변

저희의 챗봇 ‘비기닝’은 사용자가 기술 스택 추천을 요청할 때, 사용자와의 질문을 통한 상호작용을 통해 최적의 선택을 할 수 있도록 돕습니다. ‘비기닝’은 사용자의 구체적인 요청사항을 파악하기 위해 질문을 진행하며, 이를 통해 사용자의 요구와 필요에 맞는 최적의 기술 스택을 추천합니다. 또한, 각 기술 스택의 특징을 간결하고 명확하게 설명하여 사용자가 쉽게 결정을 내릴 수 있도록 지원합니다. 비기닝의 질문 순서는 다음과 같습니다.

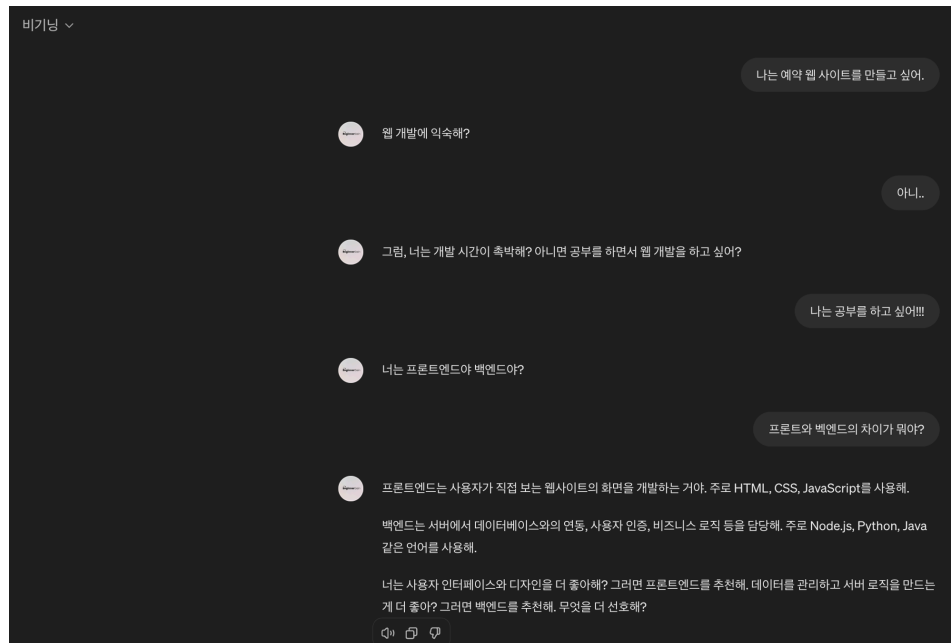
1.사용자 맞춤형 질문: 사용자의 요구를 정확히 파악하기 위해 단계별로 구체적인 질문을 합니다.

2.최적의 기술 스택 추천: 사용자의 대답을 바탕으로 사용자에게 맞는 기술 스택을 추천합니다.

3.간단한 특징 설명: 각 기술 스택의 장단점과 주요 특징을 간결하게 설명하여 사용자가 쉽게

이해하고 선택할 수 있도록 돕습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22



[그림5] 비기닝 답변

이번 프로젝트를 시작하며, 저희의 목적에 맞는 방식으로 챗봇이 답변하도록 하는 부분에서 어려움이 있었습니다. 많은 LLM이 존재하지만, 저희가 원하는 방식의 챗봇 답변을 얻어내기 위해 어떤 모델을 선택해야 하고, 이를 어떻게 활용할지를 고민해 보았습니다. 특히 한국 개발자들을 대상으로 하기 때문에 자연스러운 한글 답변에 강한 모델이 필요했습니다.

ChatGPT, LLaMA2, Gemini 등 다양한 언어 모델의 특징을 찾아보고 테스트한 결과, ChatGPT가 가장 한국어 답변이 자연스럽고 안정적인 답변을 제공하였습니다. 또한 ChatGPT는 API를 제공하여 편리하고 쉽게 사용할 수 있다는 장점이 있습니다. 이를 토대로 저희 챗봇의 모델로 ChatGPT를 선택하게 되었습니다. 모델 선정 이후에도 이를 어떻게 프롬프트 엔지니어링을 해야할 지 정하는 데에 어려움이 있었고 이 문제를 해결하기 위해 LangChain 프레임워크를 사용하였습니다. LangChain 프레임워크는 LLM 모델을 사용하여 애플리케이션을 쉽게 구축하도록 설계된 오픈 소스 프레임워크로, 이번 프로젝트의 목적에 최적화된 답변을 출력하도록 만들기 위해 LangChain을 이용한 프롬프트 엔지니어링을 진행하기로 결정하였습니다. 저희 챗봇의 프롬프트 목적은 사용자가 질문을 했을 때, 꼬리 질문을 통해 사용자의 요구사항을 파악하여 선택지를 좁혀주는 것입니다. 또한, 대상이 초보 개발자라는 점을 고려하여 사용자의 부담을 줄이기 위해 최대한 긴 설명을 피하고, 친근한 말투를 사용하는 것을 목표로 선정하였습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

이를 위해 여러 프롬프트 엔지니어링 방법들을 테스트해 보았고, 모델에게 프롬프트를 전달하는 방식으로 “Few Shot” 방법을 사용하였습니다. LangChain의 “FewShotPromptTemplate”는 몇 가지의 데이터 샘플을 통해 모델을 훈련하는 방식입니다. 저희 챗봇 아래 데이터처럼, 사용자에게 직관적인 질문을 하여 사용자에게 적합한 기술 스택을 추천합니다. 저희는 약 40개의 데이터를 이용하여 FewShot Learning을 진행하였습니다.

```
examples = [
  [
    {
      "text": "앱으로 만들까 웹으로 만들까?",
      "answer": ""
      "기기를 활용해야 해? 그러면 앱을 추천할게. 📱 단순 정보 제공이 주된 목적이야? 그러면 웹을 추천할게 🌐.",
      ""
    },
    {
      "text": "자바스크립트와 타입스크립트중에서 어떤 것을 선택할까?",
      "answer": ""
      "프로젝트의 규모가 어때? 큰 규모라면 타입스크립트가 유리해! 👍. 팀 작업을 많이 해? 여러 사람이 함께 일하면 타입스크립트가 코드 관리에 도움이 돼 🙌.",
      ""
    }
  ],
]
```

[그림6] FewShot 데이터 예제

또한, “ChatPromptTemplate” 를 사용하여 system 에게 관련 지시사항을 알려주었습니다.

```
(
  "system",
  ""
  나는 무조건 친근한 반말 말투를 사용 할거야.
  나는 일관된 말투를 사용해.
  안녕! 나는 비기너이야, 국민대학교 소프트웨어융합대학 학생들 캡스톤 15조에 의해 만들어졌어. 우리 팀은 강희구, 김선우, 김지성, 민경서, 이소정으로 구성되어 있어.
  사용자가 이름을 알려주면, '사용자의 이름' 아, 우리 같이 열심히 코딩을 해보자 🥰 너가 궁금한 점을 물어봐! 라는 말을 해줘. 너의 역할은 기술 스택이나 라이브러리를 선택할 때 조언을 해주는 역할을 할거야.
  네가 기술 스택이나 라이브러리를 선택할 때, 간단하고 직접적인 조언을 제공할 거야. 예를 들어, '안정적이고 큰 규모의 프로젝트야? 그러면 JAVA를 써.'라고 말하면 나는 그에 대한 답변을 줄 거야.
  두 번째 예시를 줄게. 너는 앞으로 이런 식으로 답변을 해.
  ,
  사용자가 react query, router, responsive의 차이가 뭐야? 라고 물으면, 서버로부터 데이터를 가져오는 데 중점을 두는 프로젝트야? 그러면 react query를 추천해.....
  URL이 변경되고 페이지가 렌더링되도록 하고 싶어? 그러면 Router 를 선택해.
  반응형 디자인이 필요해? 그러면 responsive를 선택해.라고 답변할거야.
  그럼 react, vue.js의 차이가 뭐야? 라고 물으면, 대규모 애플리케이션과 복잡한 걸 만들고 싶어? 그러면 React 를 추천해. 중소 규모의 프로젝트야? 그러면 Vue.js를 추천해.
  ,
)
```

[그림7] ChatPromptTemplate system 입력

이렇게 프롬프트를 구성한 템플릿과 다른 함수들, 모델 호출 코드 등을 ‘chain’ 을 이용하여 연결하면, 모델이 추론하여 답변합니다.

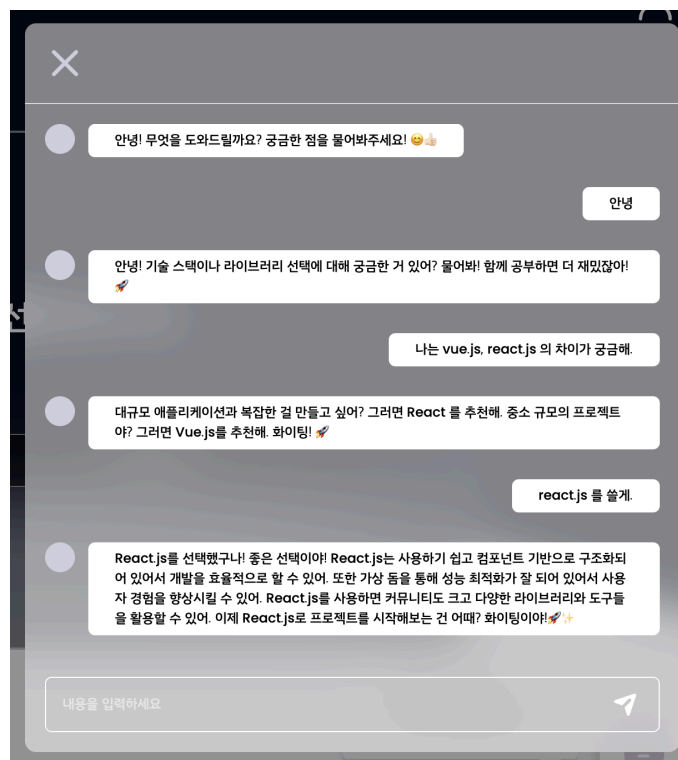
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

```

chain = (
    (format_chat_history | final_prompt | llm | StrOutputParser())
    .with_config(
        run_name="ChatBot",
        callbacks=[
            EvaluatorCallbackHandler(
                evaluators=[
                    ResponseEffectivenessEvaluator(evaluate_response_effectiveness)
                ]
            )
        ],
    )
)

```

[그림8] chain 연결



[그림9] 비기너게인 챗봇

모델 추론 결과, 위와 같은 답변을 얻었습니다. 저희가 원하는 방향으로 친절한 말투와 간결한 특징을 사용하여 답변하는 결과가 나왔습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

저희가 원하는 결과를 도출할 수 있었던 이유는 개발 시작시에 뚜렷한 목적을 선정하였고, 이 목적에 맞게 프롬프트 엔지니어링을 진행한 덕분이라 생각합니다. 언어 모델에게 어떤 프롬프트를 주느냐에 따라 성능이 크게 달라질 수 있습니다. 그래서 언어 모델을 잘 활용하기 위해서는 프롬프트 스킬이 중요합니다. 프롬프트 잘 구성하기 위해 많은 시험을 진행하였습니다. 저희의 프롬프트를 완성하기 위해 중요했던 점들을 설명하겠습니다. [참고문헌4]

1. LLM에게는 쿼션어를 사용하지 않아도 된다
2. 프롬프트를 세분화한다
3. 프롬프트에 의도한 대상 청중을 포함시킨다
4. 예제 중심의 프롬프트를 구현한다.
5. 줄바꿈을 활용한 프롬프트를 사용한다
6. 역할을 부여한다
7. 연쇄 풀이 과정과 몇번의 샷 프롬프트를 결합한다
8. 톤 설정으로 답변 스타일을 조절한다
- 9.

8가지의 내용을 포함하여 프롬프트를 작성했습니다. 계속해서 프롬프트를 바꿔가며 시험을 진행하여, 좋은 응답을 만들어 낼 수 있었습니다.

안녕하세요! 무엇을 도와드릴까요?

👍 👎 🔗 View LangSmith trace

HUMAN

나는 지금 고민하고 있어. 앱으로 만들까 웹으로 만들까?

AI

앱과 웹 중 어떤 것을 선택할지는 프로젝트의 목적과 사용자에게 제공하고자 하는 경험에 따라 다를 수 있습니다. 앱은 보다 특정한 기능을 제공하거나 오프라인에서도 사용할 수 있는 장점이 있고, 웹은 보다 넓은 범위의 사용자에게 접근이 쉬운 장점이 있습니다. 프로젝트의 요구 사항과 목적을 고려하여 결정해보세요. 어떤 프로젝트를 하고 싶으신가요?

👍 👎 🔗 View LangSmith trace

HUMAN

나는 게시판 을 만드는 프로젝트를 만들고 싶어.

AI

게시판을 만드는 프로젝트라면 웹 애플리케이션으로 제작하는 것이 적합할 수 있습니다. 웹 기술을 사용하여 사용자들이 쉽게 접근하고 이용할 수 있으며, 다양한 기기에서도 호환되는 장점이 있습니다. HTML, CSS, JavaScript와 함께 백엔드 언어와 데이터베이스를 활용하여 웹 기반의 게시판을 구축해보시는 것을 추천드립니다. 어떤 언어나 기술을 사용해볼 계획이신가요? 부족한 부분이 있거나 궁금한 점이 있다면 언제든지 물어보세요. 함께 해결해나가겠습니다.

👍 👎 🔗 View LangSmith trace

[그림10] 프롬프팅 전

HUMAN

자바스크립트랑 타입스크립트 중에서 고민중이야.

AI

프로젝트가 크면 타입스크립트를 사용하는 게 좋을 거야. 타입스크립트는 코드를 더 안정적으로 만들어줘. 작은 프로젝트라면 자바스크립트를 사용해도 괜찮아. 어떤 프로젝트를 진행하려고 하는 거니?

👍 👎 🔗 View LangSmith trace

HUMAN

작은 게시판 만들기 프로젝트야

AI

작은 게시판 프로젝트라면 자바스크립트로 충분할 거야. 간단한 기능을 구현하기에 좋을 거라고 생각해. 재미있는 프로젝트가 되길 바래! 🥳👍

👍 👎 🔗 View LangSmith trace

HUMAN

그럼, 자바스크립트에서 어떤 프레임워크로 만들까?

AI

작은 게시판 프로젝트라면 Vue.js나 React.js를 사용하면 좋을 거야. 둘 다 인기 있고 컴포넌트 기반으로 구조화되어 있어서 프로젝트를 효율적으로 관리할 수 있어. 어떤 프레임워크를 선택할지 고민 중이니?

👍 👎 🔗 View LangSmith trace

[그림11] 프롬프팅 후

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

추가적으로, 챗봇 테스트를 진행할 때, 더욱 편리한 테스트를 위해 “LangServe”를 이용하여 테스트를 진행하였습니다. LangServe는 서비스로 제공하기 위한 프레임워크이며, LangChain 파일과 체인을 FastAPI 로 쉽게 배포할 수 있도록 합니다. ‘pip install “langserve[all]”’ 을 통해 langserve 를 사용할 수 있습니다. 간단하게 server.py를 작성하여 웹 사이트에 띄울 수 있습니다. 따로 프론트엔드 템플릿을 제공하진 않지만, 백엔드 서빙 아키텍처 형태로 제공되어 테스트는 playground 에서 가능합니다.

```

from fastapi import FastAPI
from fastapi.responses import RedirectResponse
from langserve import add_routes
import sys
sys.path.append('/Users/mingyeongseo/PycharmProjects/chatbot_feedback/my-app/packages/chat-bot-feedback')

from chat_bot_feedback.chain import chain

app = FastAPI()

@app.get("/")
async def redirect_root_to_docs():
    return RedirectResponse("/docs")

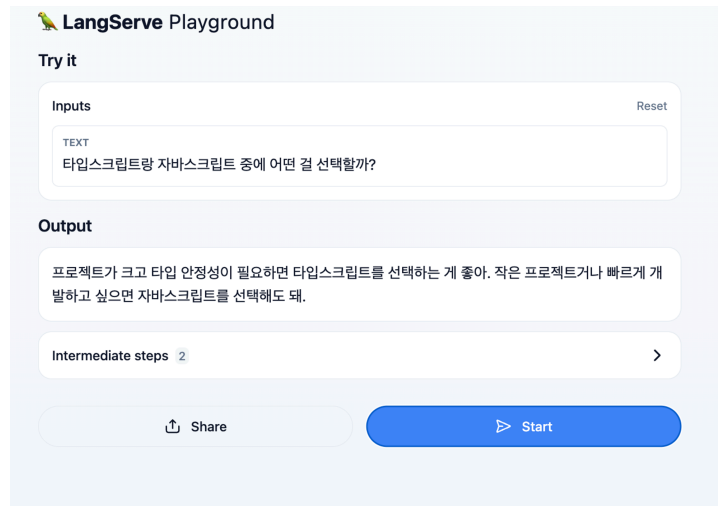
# Edit this to add the chain you want to add
# add_routes(app, NotImplemented)
add_routes(app, chain, path="/chat-bot-feedback",
            include_callback_events=True,
            enable_feedback_endpoint=True,
            enable_public_trace_link_endpoint=True,
            )

if __name__ == "__main__":
    import uvicorn
    uvicorn.run(app, host="0.0.0.0", port=8031)

```

[그림 12] server.py

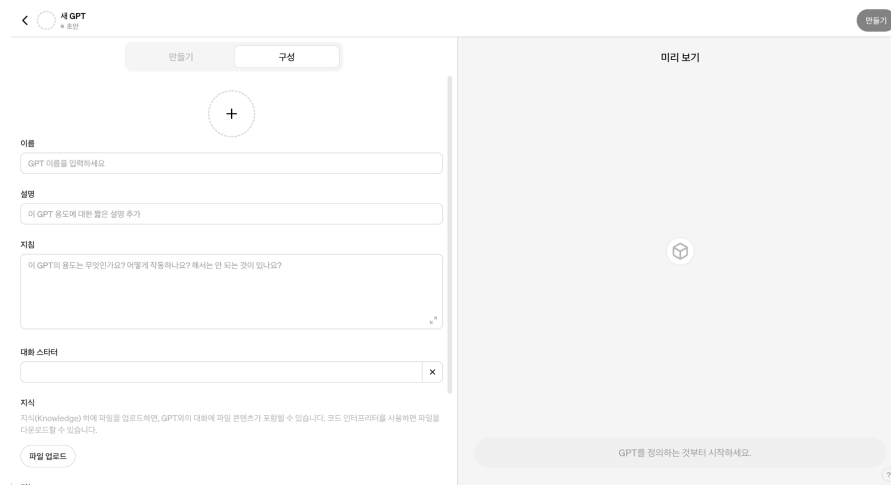
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22



[그림13] playground에서 테스트 진행

ChatGPT Store 업로드

프롬프팅한 챗봇을 더 많은 사용자에게 테스트해보기 위해 OpenAI GPT Store에 공개하였습니다. ChatGPT Store에 공개한 챗봇은 저희의 서비스에 배포한 챗봇과 다른 점이 있습니다. 우선, ChatGPT Store에서 ChatGPT Builder 를 이용하여 나만의 GPT를 배포하는 과정은 no-code 과정으로 이루어집니다.

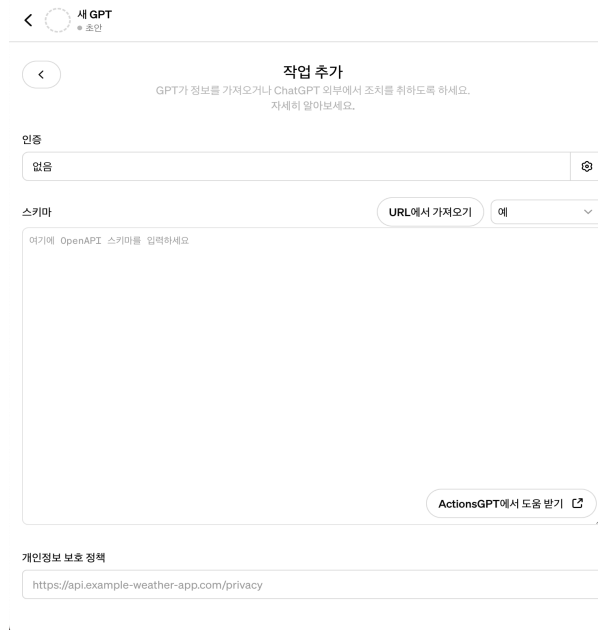


[그림14] ChatGPT Builder

위의 사진을 보면, **GPT**의 이름, 간단한 설명, 지침, 대화 스타터 등을 지정할 수 있습니다. 원하는 목적에 맞춰 각각을 입력하면 쉽게 나만의 **GPT**를 만들 수 있습니다. 저희는 비기너게인 서비스에 맞춰 프롬프팅한 챗봇의 데이터와 지침을 넣어주었습니다. 그리고

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

데이터는 파일로 업로드 할 수 있기 때문에 pdf, json 등의 파일을 만들어 업로드하면 됩니다. 이 과정까지는 코딩을 하지 못하는 사람들도 쉽게 나만의 GPT를 구현할 수 있습니다. 추가적으로, 개발이 들어가야 할 부분에 대해 설명하겠습니다. Builder 에는 “GPTs Action” 기능이 있습니다.



[그림15] GPT Action

이 기능은 외부 API 를 연결하여 외부 데이터를 사용할 수 있게 하는 기능입니다. API 연결을 통해 모델이 외부 데이터에 접근하여 특정 작업을 실행할 수 있게 합니다. 먼저, 인증, 스키마, 개인정보보호 정책 3가지 주요 특징을 입력해야 합니다.

- **인증 방법 선택** : API 접근을 위하여 인증 방법을 선택합니다. ‘없음’, ‘API Key’, ‘OAuth’ 중에서 선택할 수 있습니다.
- **스키마 입력** : GPT가 API 에 어떻게 접근할지를 정의하는 스키마를 입력합니다.
- **개인정보보호 정책** : 사용자가 GPT를 열었을 때 보게되는 개인정보 정책의 URL을 설정합니다.

챗봇 ‘비기닝’은 두 개의 외부 API를 사용하여 사용자가 더 흥미롭게 사용하고 필요한 정보를 더 쉽게 얻을 수 있게 지원하고 있습니다

날씨 API : 날씨 API를 연결하여 사용자가 챗봇과의 대화의 시작을 보다 흥미롭고 친근하게 시작할 수 있도록 가볍게 날씨 이야기를 나눌 수 있도록 했습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

유튜브 검색 API : 유튜브 검색 API를 연결하여 기술 스택이나 라이브러리와 관련된 교육 콘텐츠를 추천받을 수 있도록 했습니다.

사용할 API를 선택한 후에는 스키마를 작성해야 하는데, 이는 나만의 GPT를 제작할 때 코딩이 필요한 부분입니다. API 문서를 참고하여 스키마를 작성하였고 해당 기능들을 구현할 수 있었습니다.

작업 편집

GPT가 정보를 가져오거나 ChatGPT 외부에서 조치를 취하도록 하세요. 자세히 알아보세요.

인증

없음

스키마

URL에서 가져오기 예

```

{
  "openapi": "3.1.0",
  "info": {
    "title": "YouTube Search API",
    "version": "1.0.0",
    "description": "This API allows clients to search for YouTube videos using the serpapi.com service.",
    "contact": {
      "name": "SerpAPI Support",
      "url": "https://serpapi.com/contact",
      "email": "support@serpapi.com"
    }
  },
  "servers": [
    {
      "url": "https://serpapi.com",
      "description": "Production server"
    }
  ],
  "paths": {
    "/search": {
      "get": {
        "summary": "Search YouTube videos",
        "operationId": "searchYouTubeVideos"
      }
    }
  }
}

```

형식

가능한 작업

이름	메서드	경로	테스트
youtubeSearch	GET	/search	테스트

작업 편집

GPT가 정보를 가져오거나 ChatGPT 외부에서 조치를 취하도록 하세요. 자세히 알아보세요.

인증

API 키

스키마

URL에서 가져오기 예

```

{
  "openapi": "3.0.0",
  "info": {
    "title": "OpenWeatherMap Current Weather Data API",
    "description": "Access current weather data for any location including over 200,000 cities. We collect and process weather data from different sources such as global and local weather models, satellites, radars, and a vast network of weather stations.",
    "version": "1.0.0"
  },
  "servers": [
    {
      "url": "https://api.openweathermap.org/data/2.5",
      "description": "Production server"
    }
  ],
  "paths": {
    "/weather": {
      "get": {
        "operationId": "getCurrentWeather",
        "summary": "Get current weather data",
        "parameters": [
          {
            "name": "lat",
            "in": "query",
            "required": true
          }
        ]
      }
    }
  }
}

```

형식

가능한 작업

이름	메서드	경로	테스트
getCurrentWeather	GET	/weather	테스트

[그림16] 날씨 api schema

[그림17] 유튜브 검색 api schema

이후, 개인정보 정책 URL 을 입력하면 api를 사용할 수 있습니다.

비기너이(가) serpapi.com(와) 대화하고 싶어합니다

허용하기

항상 허용하기

거부하기

신뢰하는 사이트만 허용합니다.

serpapi.com(와) 대화 완료

여기 몇 가지 추천할 만한 React 프론트엔드 개발 강좌 유튜브 영상이 있어 한국어로 되어 있어서 이해하기 쉬울 거야. 🤗

- 이거 모르면 프론트엔드 개발자 못됨 | 리엑트 진짜 쉽게 설명해줌 | 리엑트 초보자 입문 강의 1탄 - 코딩알려주는누나
- 리엑트(React) 첫 프로젝트! 프론트엔드 하려면 기초지만 반드시 알아야 할 리엑트 프로젝트 생성 - 클래스롱
- 한시간 만에 끝내는 React.js 입문 - 개발자의품격
- React 기초 0강 : 리엑트 왜 쓰는지 알려줌 (+ 수강시 필요 사전지식) - 코딩애플

이 영상을 보면서 공부해보자! 열심히 코딩해봐! 🚀

🔊 🗨️ 🔄 📌 📄

[그림18] 외부 API 동작 연결

[그림19] 외부 API 동작 결과

테스트 결과, API 연결이 잘 되었고, 저희가 의도한 답변을 확인할 수 있었습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

2.2.1.2 백엔드

TypeScript로 작성한 Node.js 기반의 백엔드 프레임워크인 NestJS를 이용하여 서버 구축과 데이터베이스 설계하였습니다.

이번 프로젝트에서 서버 구축과 챗봇, 사용자, 서버 간의 통신과 데이터 교환을 위한 API구현을 중점으로 진행하였습니다.

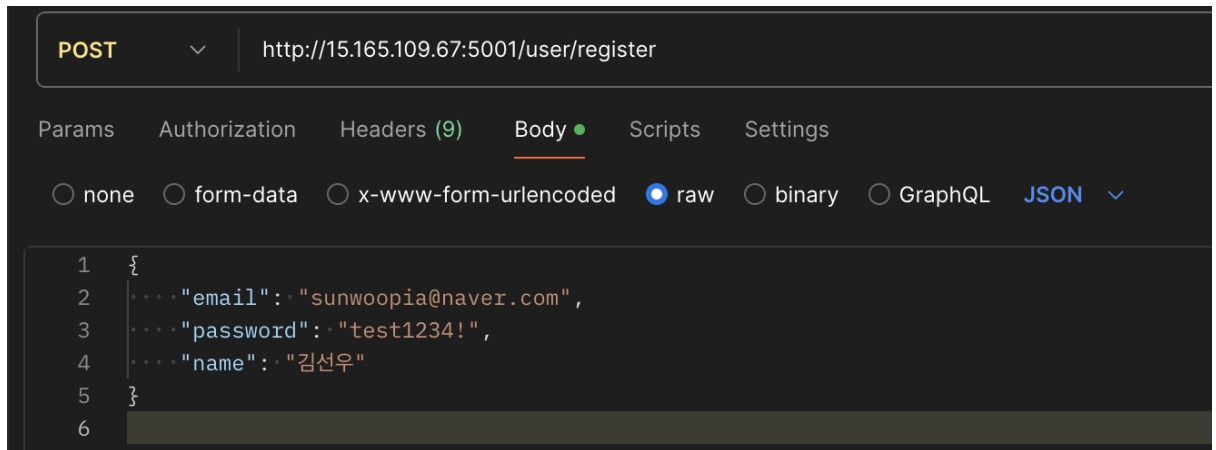
User API구현 : 사용자의 회원가입, 로그인, 비밀번호 초기화, 비밀번호 변경 기능을 구현한 API입니다. 세부사항은 다음과 같습니다.

회원가입 API : User API의 회원가입 기능을 구현하며 사용자 데이터를 안전하게 처리, 저장하는 방법에 대해 학습하였고 사용자의 회원가입 API를 구현했습니다. 회원가입은 아래와 같은 과정으로 진행됩니다.

1. 먼저 사용자가 입력한 이메일이 이미 등록되어 있는지 확인하기 위해 데이터베이스에서 조회를 수행합니다.
2. 이메일이 이미 존재하는 이메일이라면 ConflictException을 발생시켜 에러를 처리합니다.
3. 비밀번호는 보안을 강화하기 위해 bcrypt 라이브러리를 사용하여 해싱한 후 데이터베이스에 저장합니다

위 과정을 통해 사용자 정보를 안전하게 관리하고, 새로운 사용자를 등록할 수 있도록 했습니다. 회원가입 API는 POST 요청을 통해 작동하며, JSON 형식으로 사용자 데이터를 전송받아 처리합니다. 이를 통해 회원가입 기능을 효율적이고 안전하게 구현했습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22



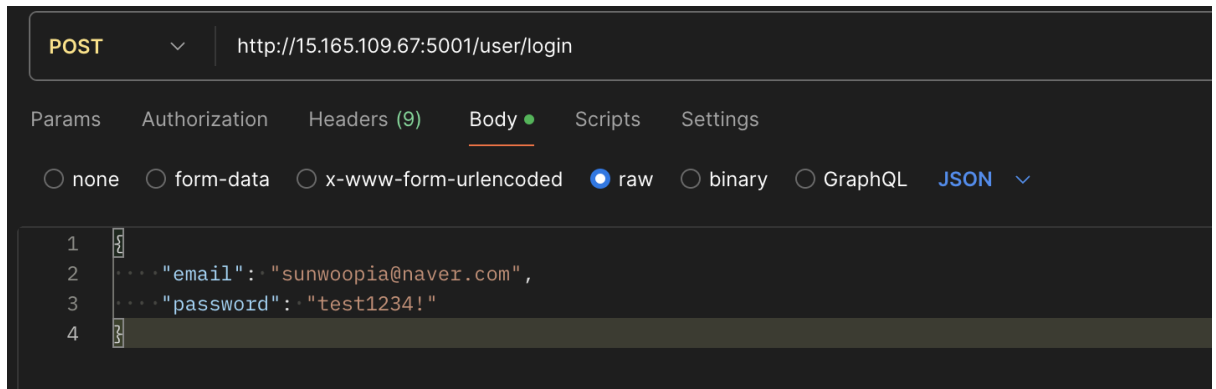
[그림20] 회원가입 API구현

로그인 API : User API의 로그인 API를 구현하며, 사용자 인증의 기본 원리와 보안 강화를 위해 필요한 절차들을 학습하였고, 로그인 API를 구현했습니다. 로그인은 아래와 같은 과정으로 진행됩니다.

1. 사용자가 입력한 이메일을 데이터베이스에서 조회하여 해당 이메일로 등록된 사용자가 존재하는지 확인합니다.
2. 만약 사용자가 존재하지 않으면, NotFoundException을 발생시켜 에러를 처리합니다.
3. bcrypt 라이브러리를 사용하여 사용자가 입력한 비밀번호와 데이터베이스에 저장된 해시된 비밀번호를 비교하여 비밀번호 일치 여부를 확인합니다.
4. 비밀번호가 일치하지 않으면 UnauthorizedException을 발생시켜 로그인 실패를 처리합니다.

위 과정을 통해, 로그인 요청이 들어올 때 사용자의 신원을 확인하고 안전하게 인증할 수 있도록 구현했습니다. 로그인 API는 POST 요청을 통해 작동하며, JSON 형식으로 사용자 데이터를 전송받아 처리합니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22



[그림21] 로그인 API구현

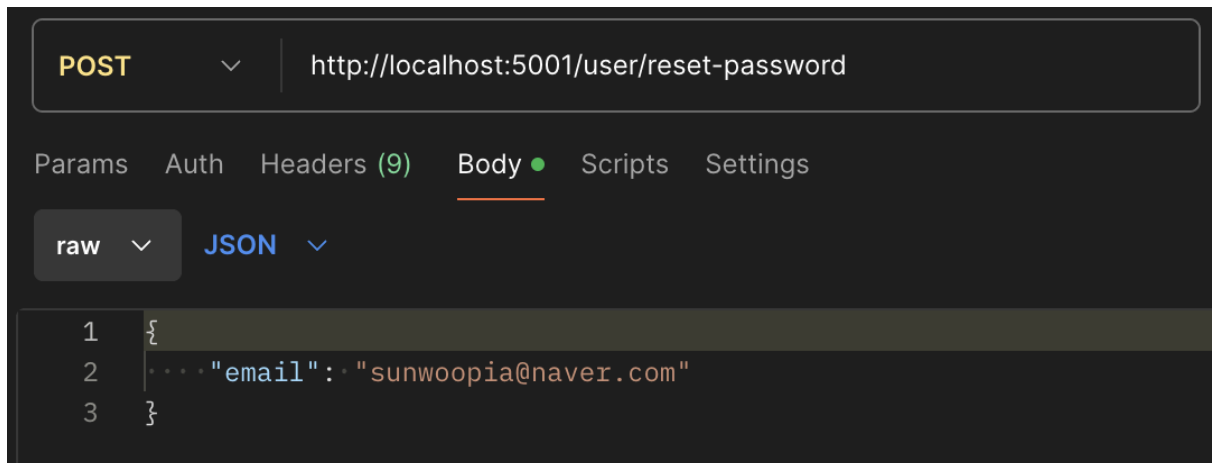
비밀번호 초기화 : User API의 비밀번호 초기화 API를 구현하며, 사용자 계정 보안 및 비밀번호 재설정 절차 및 메일 전송 기능에 대해 학습하였고, 비밀번호 초기화 API를 구현했습니다. 비밀번호 초기화는 아래 과정으로 진행됩니다.

1. 사용자가 입력한 이메일을 데이터베이스에서 조회하여 해당 이메일로 등록된 사용자가 존재하는지 확인합니다.
2. 사용자가 존재하지 않으면 NotFoundException을 발생시켜 에러를 처리합니다.
3. 임시 비밀번호를 생성하고 bcrypt 라이브러리를 사용하여 해시한 후 데이터베이스에 저장합니다.
4. 사용자의 이메일로 초기화된 임시 비밀번호를 전송합니다.

비밀번호 초기화 후, nodemailer 라이브러리를 활용하여 사용자의 이메일로 임시 비밀번호를 전송하는 기능도 구현했습니다. 이를 통해 사용자는 비밀번호를 잊었을 때 이메일로 임시 비밀번호를 받아 로그인을 할 수 있습니다. 비밀번호 초기화 API는 POST 요청을 통해 작동하며, JSON 형식으로 이메일 주소를 받아

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노패인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

처리합니다.



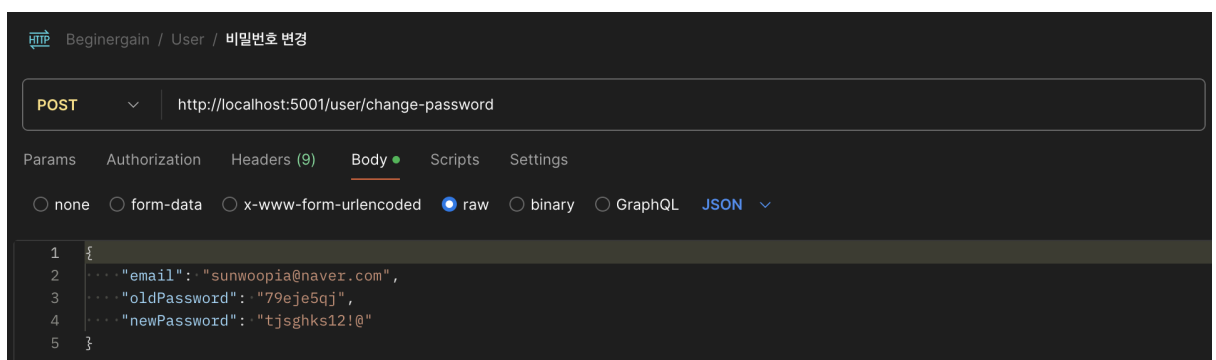
[그림22] 비밀번호 초기화 API구현

비밀번호 변경 : User API의 비밀번호 변경 API를 구현하며 사용자 인증과 비밀번호 업데이트 절차를 학습하였고, 비밀번호 변경 API를 구현했습니다. 비밀번호 초기화는 아래 과정으로 진행됩니다

1. 사용자가 입력한 이메일을 데이터베이스에서 조회하여 해당 이메일로 등록된 사용자가 존재하는지 확인합니다.
2. 사용자가 존재하지 않으면 NotFoundException을 발생시켜 에러를 처리합니다.
3. 사용자가 입력한 기존 비밀번호를 데이터베이스에 저장된 해시된 비밀번호와 비교하여 일치 여부를 확인합니다.
4. 비밀번호가 일치하지 않으면 UnauthorizedException을 발생시켜 에러를 처리합니다.
5. 새로운 비밀번호는 bcrypt 라이브러리를 사용하여 해싱한 후 데이터베이스에 저장됩니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노패인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

이러한 과정을 통해, 비밀번호 변경 요청이 들어올 때 사용자의 신원을 확인하고 안전하게 비밀번호를 업데이트할 수 있도록 구현했습니다. 비밀번호 변경 API는 POST 요청을 통해 작동하며, JSON 형식으로 이메일, 기존 비밀번호, 새로운 비밀번호를 받아 처리합니다.



[그림23] 비밀번호 변경 API구현

Project API구현 : 사용자가 요청한 프로젝트 별로 생성된 보일러 플레이트 생성 및 관리 기능을 구현한 API입니다. 세부사항은 다음과 같습니다.

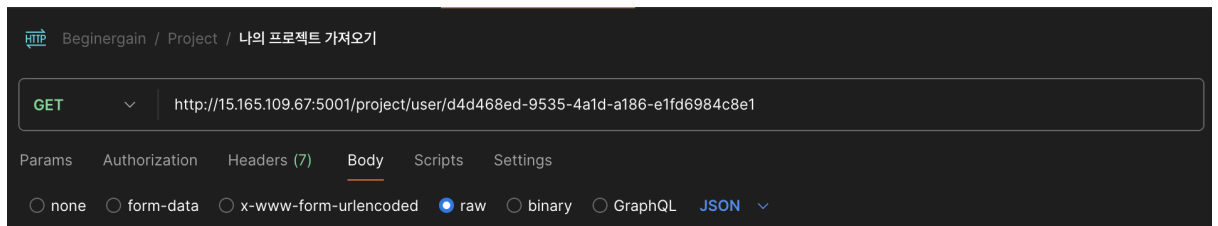
나의 프로젝트 가져오기 API : Project API의 사용자의 프로젝트를 가져오는 API를 구현하며 데이터베이스에서 관계 데이터를 조회하는 방법에 대해 알아보고, 나의 보일러 플레이트 가져오기 API를 구현했습니다.

1. 사용자의 ID를 기반으로 데이터베이스에서 해당 사용자를 조회합니다.
2. 해당 사용자가 소유한 프로젝트들을 조회하기 위해 프로젝트 저장소를

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

사용하여 소유자가 해당 사용자로 설정된 프로젝트 목록을 반환합니다.

이를 통해 사용자는 자신의 프로젝트 목록을 조회할 수 있습니다. 이 API는 GET 요청을 통해 작동하며, 경로 변수로 사용자의 ID를 받아 처리합니다.



[그림24] 나의 프로젝트 가져오기API 구현

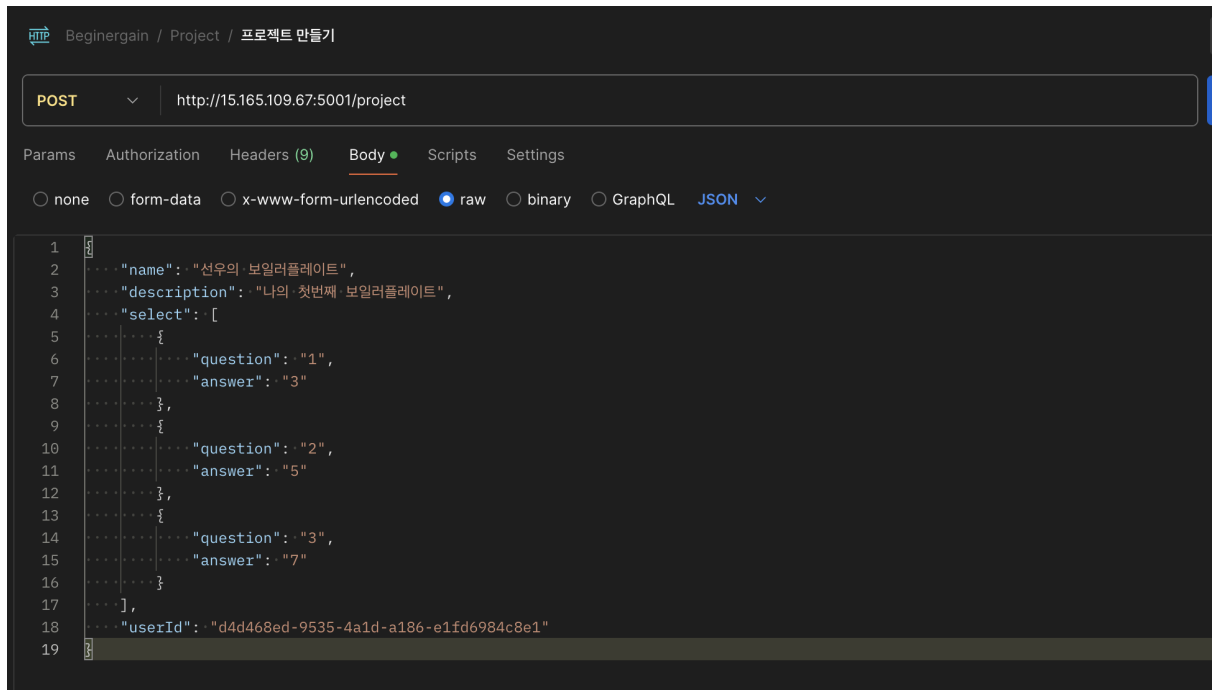
프로젝트 생성 API : Project API의 프로젝트 생성 API를 구현하며 사용자와 프로젝트 간 관계를

관리하는 방법을 학습하였고, 프로젝트 생성 API를 구현했습니다.

1. 프로젝트를 생성할 사용자의 ID를 기반으로 데이터베이스에서 해당 사용자를 조회합니다.
2. 프로젝트의 세부 정보(프로젝트 이름, 설명, 파일 경로 및 선택 항목)를 포함하는 DTO(Data Transfer Object)를 사용하여 새로운 프로젝트 엔티티를 생성합니다.
3. 새 프로젝트를 데이터베이스에 저장하여 사용자가 생성한 프로젝트를 기록합니다.

이 API는 POST 요청을 통해 작동하며, JSON 형식으로 프로젝트 데이터를 받아 처리합니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22



[그림25] 프로젝트 생성 API 구현

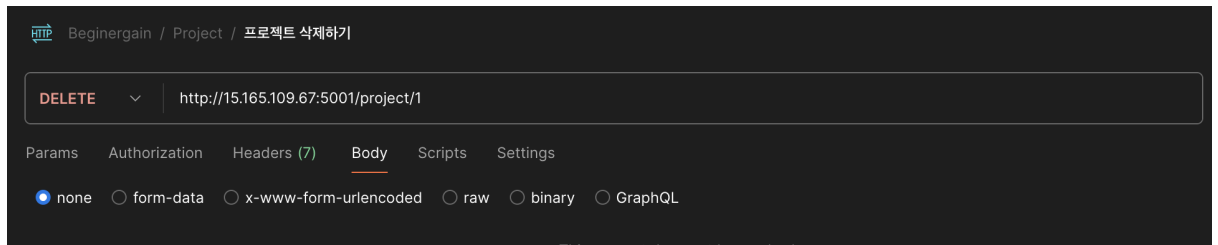
프로젝트 삭제 API : 프로젝트 삭제 API를 구현하며 데이터베이스에서 특정 엔티티를 안전하게

제거하는 방법을 학습하였고, 프로젝트 삭제 API를 구현했습니다.

1. 삭제할 프로젝트의 ID를 통해 데이터베이스에서 해당 프로젝트를 조회하여 존재 여부를 확인합니다.
2. 프로젝트가 존재하는 경우, 해당 프로젝트를 데이터베이스에서 제거하는 작업을 수행합니다.

이 API는 DELETE 요청을 통해 작동하며, 경로 변수로 프로젝트의 ID를 받아 처리합니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22



[그림26] 프로젝트 생성 API 구현

Question API구현 : 사용자의 개발 프로젝트 목적을 파악하기 위해 필요한 질문들을 생성하고 가져오기 위한 기능을 구현한 API입니다. 세부사항은 다음과 같습니다.

질문 가져오기 API : 질문 조회 API를 구현하며 RESTful API 설계와 데이터베이스에서 관계 데이터를 조회하는 방법을 학습하였고, 질문 가져오기 API를 구현했습니다.

1. getAllQuestions 및 getQuestionById 메서드를 정의하여 모든 질문과 특정 ID의 질문을 조회할 수 있도록 했습니다.
2. getQuestionById 메서드는 questionId를 매개변수로 받아 데이터베이스에서 해당 ID에 해당하는 질문을 조회하고 답변 선택지 데이터를 함께 반환합니다.
3. 이를 위해 questionRepository를 조회하여, relations 옵션을 통해 연관된 답변 데이터를 포함하여 조회합니다.

이 API는 GET 요청을 통해 작동하며, 경로 변수로 질문의 ID를 받아 처리합니다. 이를 통해 사용자가 특정 질문과 그에 대한 답변을 조회할 수 있도록 구현했습니다.

질문 생성하기 API : 질문 생성 API를 구현하면서, 저는 데이터 유효성 검사와 데이터베이스에 데이터를 저장하는 방법을 학습했습니다.

1. 사용자 요청 으로부터 질문 데이터를 수신하고 ValidationPipe를 사용하여 입력 데이터를 자동으로 검증하고 변환합니다.
2. CreateQuestionDto를 매개변수로 받아, 이를 기반으로 새로운 질문 엔티티를 생성하고 데이터베이스에 저장합니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

이 API는 POST 요청을 통해 작동하며, JSON 형식으로 질문 데이터를 받아 처리합니다. 이러한 과정을 통해 사용자가 새로운 질문을 효율적이고 안전하게 생성할 수 있도록 구현할 수 있었습니다.

질문 그룹 등록하기 API : 질문 그룹 생성 API를 구현하며, 여러 질문을 그룹으로 묶어 관리하는 방법을 학습했습니다.

1. CreateQuestionGroupDto를 매개변수로 받아, 이를 기반으로 그룹에 포함될 질문들을 조회합니다.
2. 각 질문 ID에 대해 비동기적으로 질문을 조회하고, 모든 질문이 조회되면 이를 Questions 배열에 저장합니다.
3. 조회된 질문들을 기반으로 새로운 질문 그룹 엔티티를 생성하고, 이를 데이터베이스에 저장합니다.

이 API는 POST 요청을 통해 작동하며, JSON 형식으로 질문 ID 배열을 받아 처리합니다. 이러한 과정을 통해 사용자가 여러 질문을 그룹으로 효율적이고 안전하게 생성할 수 있도록 구현할 수 있었습니다.

질문 그룹 가져오기 API : 질문 그룹 조회 API를 구현하며 관계형 데이터를 효율적으로 조회하는 방법을 학습했습니다. getQuestionGroupById 메서드를 정의하여 특정 ID를 가진 질문 그룹을 조회할 수 있도록 했습니다.

1. groupId를 매개변수로 받아 데이터베이스에서 해당 그룹 ID를 가진 질문 그룹을 조회하며, 관련된 질문과 각 질문의 답변 데이터를 함께 반환합니다.
2. 경로 변수로부터 그룹 ID를 받아 처리하며 getQuestionGroupById 메서드를 호출하여 실제 데이터 조회를 수행합니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

이 API는 GET 요청을 통해 작동하며, 경로 변수로 그룹 ID를 받아 처리합니다.
 이러한 과정을 통해 사용자가 특정 질문 그룹과 그에 속한 질문 및 답변 데이터를
 조회할 수 있도록 구현했습니다.

Answer API 구현 : 서비스가 요청한 질문에 대한 사용자의 대답을 처리하는 API입니다.

대답 생성하기 API : 정답 생성 API를 구현하며 사용자가 입력한 정답 데이터를
 데이터베이스에 저장하는 방법을 학습했고, createAnswer 메서드를 정의하여
 새로운 정답을 생성할 수 있도록 했습니다.

1. CreateAnswerDto를 매개변수로 받아, 이를 기반으로 새로운 정답 엔티티를 생성합니다.
2. answerRepository의 create 메서드를 사용하여 정답 엔티티를 생성하고 데이터베이스에 저장합니다.

대답 생성 API는 POST 요청을 통해 작동하며, JSON 형식으로 질문 ID와 정답
 내용을 받아 처리합니다.

이를 통해 사용자가 질문에 대한 정답을 제출하고 저장할 수 있는 기능을
 구현했습니다.

다음 질문 추가하기 API : 대답의 다음 질문을 업데이트하는 API를 구현하며 기존
 데이터를

수정하는 방법과 데이터 검증을 학습했고, updateNextQuestion 메서드를
 정의하여 특정 정답의

다음 질문 ID를 업데이트할 수 있도록 했습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노패인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

1. 이 메서드는 UpdateNextQuestionDto를 매개변수로 받아, 해당 정답을 데이터베이스에서 조회합니다. 정답이 존재하지 않으면 Error 예외를 발생시켜 처리합니다.
2. 새로운 다음 질문 ID로 업데이트하고 데이터베이스에 저장합니다. 이를 위해 answerRepository의 findOne 메서드와 save 메서드를 사용하여 답을 조회하고 업데이트된 답을 저장합니다.

사용자 요청으로부터 업데이트할 데이터를 수신하며, ValidationPipe를 사용하여 입력 데이터를 자동으로 검증하고 변환합니다.

이 API는 PATCH 요청을 통해 작동하며, JSON 형식으로 정답 ID와 새로운 질문 ID를 받아 처리합니다.

챗봇 서버 구현 : 클라이언트에서 챗봇을 사용하기 위해 별도의 서버를 구축하여 사용하였습니다

1. 파이썬 FastAPI 연동

FastAPI란 API를 신속하게 구축하기 위한 파이썬 프레임워크로 빠른 확장성을 위해 사용하였습니다.

주요장점으로는 자동 문서화, 높은 성능과 확장성이 있습니다.

2. 웹소켓을 이용한 소켓통신 구현

- 클라이언트의 웹소켓을 수락하고 클라이언트로부터 받은 비동기 형식의 메시지를 전달 받음
- generator(Stream 형태의 데이터 생성 객체)를 생성하여 소켓통신을 통해 클라이언트 사이드로 응답 Response를 전달합니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

```
@app.websocket("/ws")
async def websocket_endpoint(websocket: WebSocket):
    await websocket.accept()
    while True:
        try:
            data = await websocket.receive_text()
            generator = chatbot_chain.stream({"text": data})
            await asyncio.to_thread(stream_responses_sync, *args: generator, websocket)
            await websocket.send_text("Response completed.")
        except WebSocketDisconnect:
            break
```

[그림27] Websocket API EndPoint

3. 실시간 응답을 위한 Stream Response 응답

- send_item 함수를 통해 위에서 생성된 Generator를 클라이언트에 전달
- 아래 for문을 통해 생성되고 있는 Stream 객체의 단어를 비동기 형식으로 전달

```
def stream_responses_sync(generator, websocket):
    ⚡ sunwoopia
    async def send_item(item):
        await websocket.send_text(item)

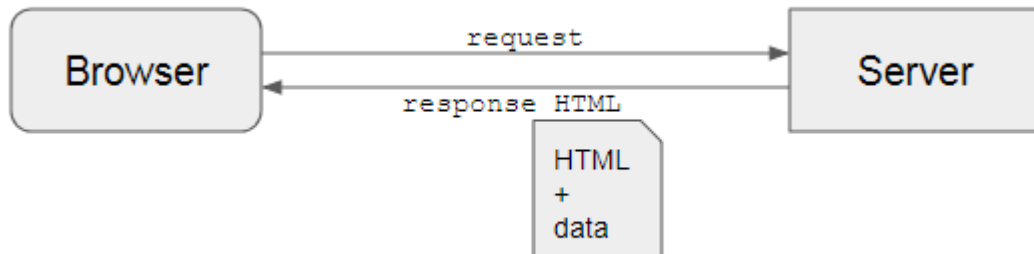
    for item in generator:
        asyncio.run(send_item(item))
```

[그림28] Stream 객체 클라이언트 통신 함수

2.2.1.3 프론트엔드

React.js 기반의 서버 사이드 렌더링(SSR) Framework인 Next.js를 사용하여 프론트엔드 서비스를 구축했습니다. 기존 React.js는 클라이언트 사이드 렌더링(CSR) 기반으로 이루어져있는데, 이는 모든 js 파일을 로드한 후 사용자에게 웹이 보여지기 때문에 사용자의 대기 시간이 길어질 수 있다는 단점이 있습니다. 또한 로드되지 않은 자바스크립트 페이지는 구글 검색엔진에서 스캔했을 때 아무런 정보도 제공되지 않습니다. Next.js는 이러한 클라이언트 사이드 렌더링의 단점을 보완하고, 자동으로 페이지를 라우팅해주는 등 다양한 기능을 제공하기 때문에 해당 프레임워크를 사용하게 되었습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22



[그림29] 서버 사이드 렌더링(SSR) flow

효율적인 스타일링 작업을 위해 Tailwind css와 styled-component를 혼합했습니다. tailwind css의 config 파일에 폰트, 색상 등을 미리 세팅해두고 사용함으로써 깔끔한 스타일링 작업을 위해 노력했습니다. 조건에 따라 스타일이 자주 바뀌는 컴포넌트의 경우 styled component를 사용하여 가독성을 높였습니다. 또한 media query를 이용하여 스크린 사이즈에 따라 font-size가 변경되도록 조정한 후 rem, vh, vw 등의 단위를 사용하여 반응형 웹사이트를 구현했습니다.

서버와 데이터를 연동하기 위해 axios를 이용하여 api 함수를 구현했으며 react-query를 이용하여 api를 연동했습니다. react-query를 이용함으로써 데이터를 지속적으로 동기화하고 내부 기능을 통해 서버에서 받아온 데이터의 상태를 간단한 코드로 나타낼 수 있었습니다.

Next.js react-query 세팅

서버에서 여러 쿼리를 미리 가져올 수 있도록 세팅하기 위해 react-query의 hydration을 이용했습니다. 먼저 app.js 내부에 새 QueryClient 인스턴스를 만들어서 컴포넌트 생명주기 한번만 QueryClient를 생성되도록 했습니다. 또한 앱 컴포넌트를 hydrate로 감싸고 pageProps의 dehydratedState props에 전달했습니다. 이 때 hydration은

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

서버에서 렌더링된 HTML을 브라우저에서 그대로 사용하고, Javascript는 추가로 실행해 더 빠르게 화면을 구축하는 과정입니다. 이를 통해 초기 로딩 속도를 높이고 사용자 경험을 향상시킬 수 있습니다. 아래는 관련 코드입니다.

```

<QueryClientProvider client={queryClient}>
  <Hydrate state={pageProps.dehydrateState}>
    <RecoilRoot>
      <ComponentWrapper>
        <Component {...pageProps} />
      </ComponentWrapper>
    </RecoilRoot>
  </Hydrate>
</QueryClientProvider>

```

[그림30] _app.tsx react-query setting code

Hydrate 컴포넌트는 서버에서 만든 HTML을 클라이언트 측에서 재사용할 때 사용하는 컴포넌트 입니다. 이 때 state 속성의 pageProps.dehydrateState는 서버에서 데이터를 가져와 직렬화한 객체입니다. 이 데이터는 react-query의 dehydrate 함수를 통해 생성된 것이며 getStaticProps 함수를 통해 서버에서 데이터를 미리 가져와서 넣을 수 있습니다.

로그인/회원가입 기능 구현

사용자에게 받은 이메일, 비밀번호, 이름을 서버에 보내면 response로 유저 토큰이 전달되는데 이 토큰을 쿠키에 저장하여 로그인 여부를 판단했습니다. 또한 로그인이나 회원가입을 진행할 경우 서비스에서 필요한 유저 정보는 recoil을 이용한 전역 변수에 저장했습니다. react-query의 mutation을 이용하여 로그인 api를 연동한 코드는 다음과 같습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

```
const loginMutation : UseMutationResult<AxiosRespons... = useMutation( options: {
  mutationFn: (loginData: ILogin) => {
    return login(loginData);
  },
  onError(error : AxiosError) : void {
    if(error.response?.status === 404) {
      setIsEmailNotExist( value: true);
    }
    else if(error.response?.status === 401) {
      setIsPasswordFail( value: true);
    }
  },
  onSuccess(data: AxiosResponse) : void {
    const loginData: ILoginResponse = data.data;
    setCookie( key: 'accessId', loginData.id);
    // user 데이터 recoil 저장
    setUser( valOrUpdater: {
      id: loginData.id,
      email: loginData.email,
      name: loginData.name,
    })
    router.push( url: '/' );
  }
});
```

[그림31] react-query mutation을 이용한 login api 연동 code

가입하지 않은 이메일인 경우 404 에러가 전달되고 비밀번호가 틀린 경우 401 에러가 전달되어, 에러 코드에 따라 ui에서 다르게 보여지도록 구현하였습니다.

회원가입 페이지에서는 사용자가 작성한 이메일이 유효한 이메일인지 실시간으로 판단하는 api를 연동하여 이메일이 유효한지 실시간으로 알 수 있도록 했습니다. 또한 비밀번호와 비밀번호 확인 input에 작성한 내용이 일치하는지 실시간으로 확인할 수 있도록 구현했습니다. 이때 비밀번호와 비밀번호 확인 input이 일치하는지에 대한 boolean 변수를 useState로 관리하고자 했는데, 비밀번호 input과 비밀번호 확인 input에 같은 문자를 입력해도 비밀번호 일치여부가 false가 뜨고, 비밀번호 확인 input에 문자 하나를 더 입력해야 true가 뜨는 오류를 발견할 수 있었습니다. 이를 찾아본 결과, 이미 비밀번호와 비밀번호 확인 변수를 useState로 관리하고 있었기 때문에, 이 변수들이 변경될 경우 리렌더링되면서 비밀번호 일치 여부에 대한 변수를 useState로 관리하지 않아도 값이 변경되므로 이는 useState를 사용하지 않아도 된다는 사실을 알게 되었습니다. React.js 공식문서에 따르면 1. props를 통해 부모로부터 전달될 경우, 2. 시간이 지나도 변함이 없을 경우, 3. 컴포넌트의 다른 상태 또는 props를 기반으로 계산할

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

수 있을 경우 state가 아니라고 합니다.

질문 기능 구현

질문 페이지의 라우팅 방식은 question Id를 동적으로 쿼리에 넣기 위해 next.js 의 기능 중 하나인 다이나믹 라우팅 방식을 이용했습니다. 쿼리에서 가져온 question Id를 이용하여 질문 리스트를 가져오는 api 함수를 호출한 후 사용자 인터페이스를 동적으로 띄우도록 구현했습니다. 또한 질문에 대한 답변을 클릭했을 때 api response에 있는 다음 질문 Id를 참고하여 또 다시 질문 리스트 api를 호출합니다. 이 때 다음 질문으로 넘어가게 될 경우 화면이 새로고침되므로 선택한 답변에 대한 history가 사라집니다. 이후 플로우에서 답변에 대한 history를 모두 서버로 보내줘야 하기 때문에 질문에 대한 답변을 할 때마다 현재 질문id와 선택한 답변id를 recoil에 저장하여 전역변수로 관리했습니다. 관련 코드는 다음과 같습니다.

```

if (questionData) {
  // 현재 질문, 답변 recoil 에 저장
  const questionInfo: QuestionSelected = {
    question: questionData?.id.toString(),
    answer: answerId.toString(),
  };
  setProjectData( valOrUpdater: (prev : IMakeProject ) : {...} => ({
    ...prev,
    select: [...prev.select, questionInfo]
  }));

  // question type에 따라 경로 이동
  if (questionType === QuestionType.그룹질문) {
    router.push( url: {
      pathname: '/make-boilerplate/etc',
      query: {questionId: nextId},
    });
  } else {
    router.push( url: `/make-boilerplate/${nextId}` );
  }
}

```

[그림32] 질문 페이지 question data 관련 code

또한 api response에 있는 questionType이 2일 경우 그룹질문으로 판단하여 etc페이지로 넘어가고, 1일 경우 개별 질문으로 판단하여 해당 질문에 대한 url로 넘어가도록 구현했습니다. 이 때 코드 가독성을 위해 QuestionType에 대한 enum을 이용했습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

etc 라이브러리 선택 페이지 구현

questionGroup에 대한 api를 호출하여 라이브러리 선택 질문들을 가져온 후 데이터를 바인딩했습니다. 이 때 리스트를 체크할 때마다 어떤 것이 체크되었는지 핸들링하는 기능을 구현해야 했는데, 이를 효율적으로 수행하기 위해 체크리스트를 저장하는 데이터 형식을 고민한 결과 key-value 형식이면서 key를 동적으로 생성할 수 있는 Record 데이터 타입을 이용했습니다. 또한 해당 데이터를 useState로 관리하여 체크리스트에 대한 변수가 변경될 때마다 렌더링되어 화면에서 업데이트되도록 구현했습니다. 또한 체크를 한번더 눌렀을 때 선택을 취소해야 한다는 요구사항을 반영하였습니다. 하단의 코드는 체크리스트를 클릭할 때마다 핸들링되는 함수입니다.

```
1+ usages  jisung
const handleCheckItem = ({question, answer} : ICheckedInfo) : void => {
  // 동일한 checkbox를 한번더 눌렀을 경우 해당 key-value 제거
  if(checkedItems[question] === answer) {
    setCheckedItems( value: (prevItems : Record<number, number> ) => {
      const { [question]: _ : number , ...newItems : {[p: number]: number} } = prevItems;
      return newItems;
    });
  } else {
    // 해당 checkbox에 대한 key-value 추가
    setCheckedItems( value: (prevItems : Record<number, number> ) : {[p: number]: number} => ({
      ...prevItems,
      [question]: answer
    }));
  }
};
```

[그림33] etc page checklist handling code

보일러플레이트 생성 페이지 구현

개별 질문이 끝나면 마지막으로 그룹 질문 페이지에서 원하는 라이브러리를 선택한 후 보일러플레이트 생성 버튼을 누르게 됩니다. 이 때 서버에 질문에 대한 답변 history와 사용자에게 받은 프로젝트 정보들을 api를 이용하여 보내게 됩니다. 그 전에 recoil 변수로

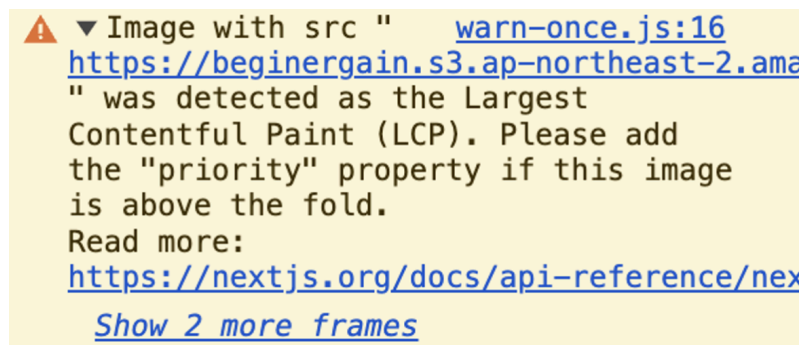
 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

저장해둔 답변 history와 프로젝트 명을 불러온 후 etc 페이지에서 선택한 답변을 추가하여 api request로 보내도록 구현했습니다.

또한 서버에 request를 보낸 후 보일러플레이트 생성이 완료되면 response로 보일러플레이트를 다운받을 수 있는 링크가 전달됩니다. complete 파일에서 이를 보여주기 위해 recoil에 보일러플레이트 url을 저장했습니다. 이후 버전에서는 url의 일부를 쿼리스트링으로 넣은 후 이를 가져와서 사용할 수 있도록 변경할 예정입니다.

오류 해결

- s3 링크를 이용한 이미지 태그 관련 warning 해결 및 LCP 감소 방법 탐색



[그림34] next.js image 태그 warning

위 warning은 용량이 큰 일러스트 이미지 컴포넌트 속성에 priority를 추가하여 해결했습니다. 또한 warning을 해결하기 위해 검색하면서 LCP에 대해 알아보았습니다. LCP는 Largest Contentful Paint의 줄임말로 페이지가 처음 로드를 시작한 시점부터 뷰포트 내에 있는 가장 큰 이미지 또는 텍스트 블록이 렌더링 되는 시점까지 걸리는 시간을 의미합니다. LCP를 줄이는 방법 중 하나는 웹 폰트를 preload하는 것이라고 합니다. 브라우저가 페이지를 렌더링할 때 외부 웹 폰트 링크로 정의된 부분을 만나면 해당 리소스가 다운로드 될 때까지 기다려야 하기 때문에 페이지의 첫 렌더링 시간이 지연될 수 있습니다. 따라서 기존에 썼던 웹폰트를 다운로드 받은 후 내부에 저장하여 사용했습니다.

- styled component attribute warning 해결

Warning: Received false for a non-boolean attribute openmodal.

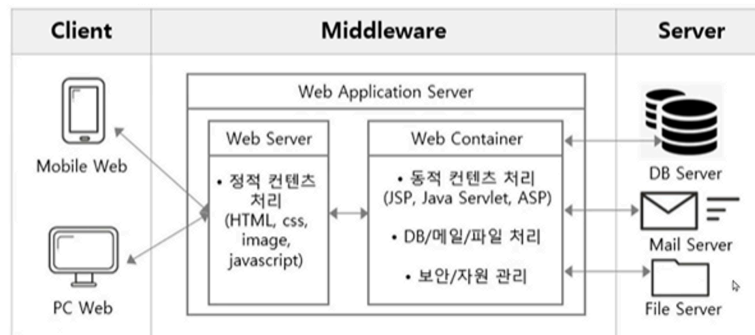
HTML 표준 속성 외에 다른 custom 속성을 전달하게 되면 위와 같은 에러가 난다고 합니다. Styled component는 기본적으로 모든 custom 속성을 HTML attribute로 전달하기 때문에 에러가 나는 것입니다. Styled component transient

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

props 기능을 이용하여 openmodal -> \$openmodal로 변경하여 사용했습니다.

사이트 Middleware 설정

Next.js의 Middleware는 Next.js에서 페이지를 렌더링하기 전에 서버 측에서 먼저 실행되는 함수입니다. 즉, 특정 요청 전에 무언가를 수행할 수 있게 해주는 기능입니다.



[그림35] Middleware 설명

Middleware는 다음과 같은 상황에서 사용할 수 있습니다.

- 페이지 렌더링 전에 인증을 확인하거나 요청을 확인할 때
- 요청 데이터를 사전에 처리하거나 특정 API 요청을 수행할 때, 혹은 캐시를 관리할 때
- 요청에 대한 응답을 변환하거나 에러를 처리할 때

저희 서비스 ‘비기너게인’에서는 웹 페이지 쿠키 정보를 확인하여 로그인 정보가 없는 경우 로그인이 필요한 페이지에 접속하였을 때 로그인 페이지로 리다이렉트 시키기 위해 Next.js Middleware 함수를 사용하였습니다. 보일러플레이트 생성하기(/make-boilerplate) 페이지와 내 보일러플레이트 보관함(/my-boilerplate) 페이지에 로그인 정보 없이 접속 시 로그인 페이지로 리다이렉트합니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

```
import type { NextRequest } from "next/server";
import { NextResponse } from "next/server";

export async function middleware(request: NextRequest) {
  const jwt = request.cookies.get('accessId');

  if (!jwt) {
    return NextResponse.redirect(new URL('/login', request.url));
  }

  return NextResponse.next();
}

export const config = {
  matcher: ["/make-boilerplate/:path*", "/my-boilerplate"]
};
```

[그림36] Middleware 함수

내 보일러플레이트 보관함 페이지 구현

웹 페이지 쿠키 정보를 확인하여 유저 로그인 정보가 있을 시, 내 보일러플레이트 보관함 페이지를 렌더링합니다. 유저 정보가 없을 시에는 로그인 페이지로 리다이렉트됩니다. 페이지 렌더링시 해당 유저의 보일러플레이트 프로젝트 목록 데이터를 요청하여 받아옵니다. 목록 컴포넌트는 기본적으로 6개씩 렌더링되고, 페이지 너비가 작아짐에 따라 목록 컴포넌트가 아래로 하나씩 이동할 수 있도록 반응형 작업하였습니다. 목록 컴포넌트에서는 해당 프로젝트의 이름과 생성 날짜를 확인할 수 있으며, 다운로드 버튼을 통해 해당 프로젝트 파일을 다시 다운로드할 수 있습니다. 또한, 삭제 버튼을 통해 해당 프로젝트를 삭제할 수 있도록 구현하였습니다. 유저의 프로젝트 목록 데이터가 없을 경우에는 “보관함이 비어있습니다” 메시지를 포함한 컴포넌트를 띄울 수 있도록 조건부 렌더링하였습니다.

데이터 목록 GET 요청과 DELETE 요청은 모두 axios를 이용해 구현되었습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노패인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

```

export const getProjects = async (userId: string) => {
  if (!userId) return;
  try {
    const response = await axios.get(`url: `${API_ENDPOINT}/project/user/${userId}``);
    return response.data;
  } catch (error) {
    return error;
  }
}

export const deleteProject = async (projectId: number) => {
  try {
    const response = await axios.delete(`url: `${API_ENDPOINT}/project/${projectId}``);
    return response.data;
  } catch (error) {
    return error;
  }
}

```

[그림37] 프로젝트 데이터 GET, DELETE 요청 함수

챗봇 구현

1. React의 useEffect를 통해 챗봇 스크린이 열렸을 때, 웹소켓에 접속합니다.
2. 현재 소켓을 상태에 저장한 후, 소켓의 메시지가 도착했을 때와 소켓 접속이 끝났을 때를 함수 처리합니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

```
useEffect( effect: () => {
  socket.current = new WebSocket( url: 'ws://15.165.109.67:8000/ws');

  socket.current.onopen = () : void => {
    console.log('WebSocket 연결이 열렸습니다. ');
    socket.current?.send( data: '안녕?');
  };

  // 메시지를 수신했을 때의 이벤트 핸들러
  socket.current.onmessage = (event : MessageEvent<any> ) : void => {
    if (event.data !== 'Response completed.') {
      setCurrentResponse( value: prevResponse : string => prevResponse + event.data);
    } else {
      setIsLoading( value: false);
    }
  };

  // 연결이 닫혔을 때의 이벤트 핸들러
  socket.current.onclose = () : void => {
    console.log('WebSocket 연결이 닫혔습니다. ');
  };

  // 컴포넌트가 언마운트되면 WebSocket 연결을 닫음
  return () : void => {
    socket.current?.close();
  };
}, deps: []);
```

[그림38] Socket API 연동 함수

- 메세지를 전송하였을 때, loading 상태로 변경하여 메세지를 대기시킵니다.
- 메세지를 한글자 단위로 쪼개서 실시간으로 연결되어있는 채팅 Response를 표기합니다.

```
const handleSendButtonClick = async () : Promise<void> => {
  if (inputValue !== '') {
    setIsLoading( value: true);
    socket.current?.send(inputValue);
    const myNewChat : {content: string, isMyChat: bo... = {
      content: inputValue,
      isMyChat: true,
    }
    setChatingHistory( value: [...chatingHistory, myNewChat]);
    setInputValue( value: '');
  }
};
```

[그림39] 채팅 전송 함수

2.2.1.4 AWS 및 배포

1. Amazon EC2

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

Amazon EC2는 클라우드에서 가상서버를 생성하고 관리할수 있는 기능을 위해 사용되며 컴퓨팅 자원을 손쉽게 확장하거나 축소할수 있게 하는 서비스 이며 프로젝트상에서 프론트와 백엔드 코드를 배포 하는데에 사용하였습니다. 아래 사진은 비기너게인 서비스의 인스턴스입니다.

추가로 탄력적 IP를 사용하여 인스턴스의 변화가 있더라도 접속하는 IP의 변화가 없이 배포 하였습니다.

i-0dfb0ed201c33a353 (beginergain)



세부 정보 | 상태 및 경보 신규 | 모니터링 | 보안 | 네트워킹 | 스토리지 | 태그

▼ 인스턴스 요약 정보

인스턴스 ID i-0dfb0ed201c33a353 (beginergain)	퍼블릭 IPv4 주소 15.165.109.67 개방 주소법	프라이빗 IPv4 주소 172.31.45.194
IPv6 주소 -	인스턴스 상태 실행 중	퍼블릭 IPv4 DNS ec2-15-165-109-67.ap-northeast-2.compute.amazonaws.com 개방 주소법
호스트 이름 유형 IP 이름: ip-172-31-45-194.ap-northeast-2.compute.internal	프라이빗 IP DNS 이름(IPv4만 해당) ip-172-31-45-194.ap-northeast-2.compute.internal	
프라이빗 리소스 DNS 이름 응답 IPv4(A)	인스턴스 유형 t3.small	탄력적 IP 주소 15.165.109.67 [퍼블릭 IP]

[그림40] 비기너게인 서비스 인스턴스

비기너게인 데이터베이스...	sg-018d1a9fb0b602260	beginner-database-group
비기너게인 EC2 보안...	sg-0c0d99d21880921d5	launch-wizard-2

[그림41] 보안그룹

추가로 데이터베이스와 EC2의 보안그룹을 나누어 관리하였고 해당 보안 그룹에는 접속할 수 있는 유형과 IP를 관리하는 인바운드 규칙과 아웃바운드 규칙을 설정하였습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

2. pm2

pm2는 Node.js 애플리케이션을 관리하기 위한 프로덕션 프로세스 관리자이며 백그라운드에서 실행되며 자동 재시작, 로드밸런싱, 로그 관리등 무종단 배포 라이브러리입니다.

pm2를 사용하면 동시에 여러가지 서버를 배포할 수 있기에, '비기너게인'의 Client와 Server를 pm2로 배포하였습니다.

```
ubuntu@ip-172-31-45-194:~/capstone-2024-15$ ls
404.html      README.md      beginner-gain-client
Gemfile       _config.yml    beginner-gain-server
Gemfile.lock  beginner-gain-chatbot  creative-theme-jekyll-master
ubuntu@ip-172-31-45-194:~/capstone-2024-15$ pm2 status
```

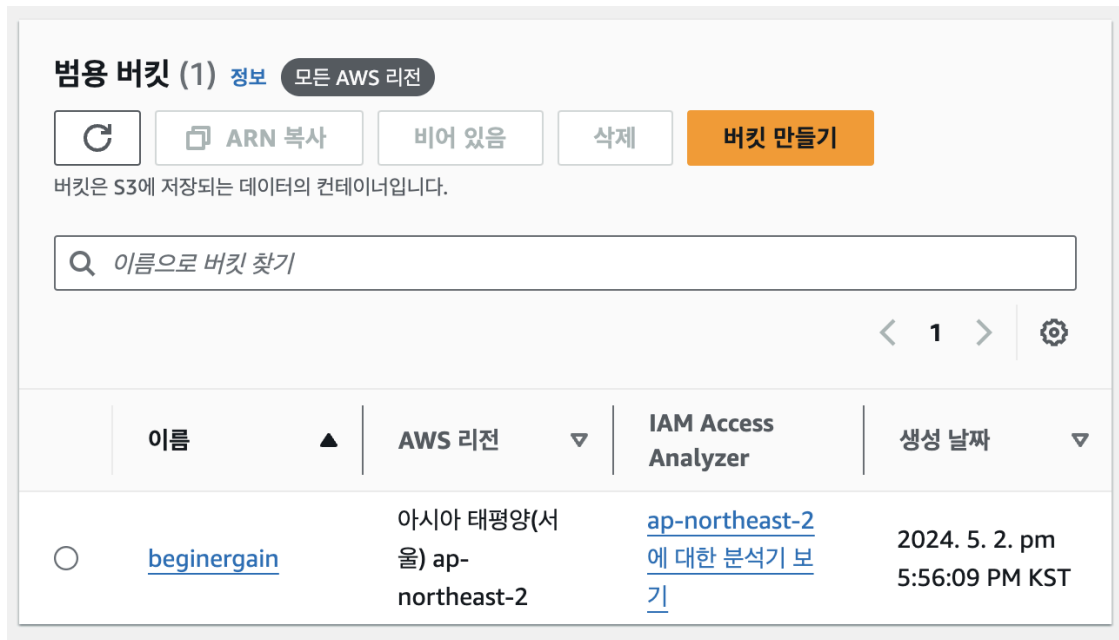
id	name	mode	⚡	status	cpu	memory
0	client	fork	15	online	0%	42.0mb
2	server	fork	14	online	0%	42.4mb

[그림42] 서버 pm2 배포

3. S3

Amazon S3(Simple Storage Service) 는 AWS에서 제공하는 객체스토리지 서비스이며 확장성, 데이터 가용성, 보안 및 성능을 제공하여 웹 규모 컴퓨팅을 위한 스토리지 인프라를 제공하는 서비스입니다. 비기너게인에서는 서버를 통해 만들어진 보일러플레이트를 저장하는 스토리지로 사용하였습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22



[그림43] 비기너게인 S3 스토리지

4. RDS

Amazon RDS (Relational Database Service) 는 AWS에서 제공하는 관리-관계형 데이터베이스 서비스이며 관리 작업을 최소화하고, 확장성, 성능, 보안 및 가용성을 갖춘 관계형 데이터베이스 관리 서비스 입니다.

PostgreSQL을 사용하였고 RDS를 통해 비기너게인의 유저, 보일러플레이트, 보관함, 질문 등의 데이터를 관리하였습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

RDS > 데이터베이스 > beginergain-database

beginergain-database
🔄
수정
작업 ▼

요약

DB 식별자 beginergain-database CPU  3.88%	상태 🟢 사용 가능 클래스 db.t3.micro	역할 인스턴스 현재 활동  0 연결	엔진 PostgreSQL 리전 및 AZ ap-northeast-2d	권장 사항  3 유용
--	--	---	--	--

<
연결 및 보안
모니터링
로그 및 이벤트
구성
유지 관리 및 백업
타 >

연결 및 보안

엔드포인트 및 포트

엔드포인트
 beginergain-database.crmim00qe4z0.ap-northeast-2.rds.amazonaws.com

포트
5432

네트워킹

가용 영역
ap-northeast-2d

VPC
vpc-0be88ec1bb841518d

서브넷 그룹
default-vpc-0be88ec1bb841518d

서브넷
subnet-0c4129d87a53eb48e
subnet-0ade6e6d789115853
subnet-0c50589a9195e2eee
subnet-0185e3aafb78218ca

[그림44] 비기너게인 데이터베이스 RDS

5. Route53

가비아에서 도메인을 구매하여 호스팅 영역을 생성하였으며 해당 레코드를 연결하였습니다.

EC2 ALB를 연결하여 ACM을 통한 SSL 인증서를 받아 해당 도메인을 https 와 www 도메인으로 연결하였습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

비밀리

beginnergain.com

정보

영역 삭제

레코드 테스트

쿼리 로깅

▶ 호스팅 영역 세부 정보

호스팅 영역 편집

레코드(5)

DNSSEC 서명

호스팅 영역 태그(0)

레코드 (5) 정보

↺

레코드 삭제

영역 파일 가져오기

레코드 생성

Automatic 모드는 최상의 필터 결과에 최적화된 현재 검색 동작입니다. 모드를 변경하려면 설정(settings)으로 이동합니다.

Q 속성 또는 값을 기준으로 레코드 필터링

유형 ▼

라우팅 정책 ▼

별칭 ▼

< 1 >

☐	레코드 이름	유형	라우팅 ...	차별화 ...	별칭	값/트래픽 라우팅 대상	TTL(초)	
☐	beginnergain.com	A	단순	-	아니요	15.165.109.67	300	-
☐	beginnergain.com	NS	단순	-	아니요	ns-1504.awsdns-60.org. ns-822.awsdns-38.net. ns-385.awsdns-48.com. ns-1648.awsdns-14.co.uk.	172800	-
☐	beginnergain.com	SOA	단순	-	아니요	ns-1504.awsdns-60.org. aws...	900	-
☐	_97c92a277d1d231db4badbb149474cec.beginnergain.com	CNAME	단순	-	아니요	_14a040d56efec6a41ae569f...	300	-
☐	www.beginnergain.com	A	단순	-	예	dualstack.beginnergain-lb-79...	-	-

[그림45] 비기너게인 도메인 레코드영역

2.2.2 시스템 기능 요구사항

2.2.2.1 인공지능

요구 기능	설명	완료 여부
-------	----	-------

캡스톤디자인 I

Page 46 of 64

결과보고서

All rights are reserved. Reproduction in whole or in parts is prohibited without the written consent of the copyright owner.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

맞춤형 추천	사용자 요구에 가장 적합한 프레임워크를 추천해야 한다.	완료
프레임워크 평가	각 프레임워크의 장단점, 사용 사례 등의 평가 기준 설명과 함께 프레임워크를 추천해야한다.	완료

2.2.2.2 백엔드

요구 기능	설명	완료 여부
회원가입	사용자는 이메일과 비밀번호를 입력하여 가입할 수 있어야한다. 중복된 이메일은 사용될 수 없다.	완료
사용자 로그인	사용자는 이메일과 비밀번호를 입력하여 시스템에 로그인할 수 있어야한다.	완료
비밀번호 초기화	사용자가 비밀번호를 잊었을 경우 초기화된 임시 비밀번호를 메일로 받을 수 있어야한다.	완료
보관함 기능	사용자는 한 번 생성한 보일러 플레이트를 다시 조회하고 다운로드 받을 수 있어야한다	완료
질문 기능	시스템은 사용자에게 개발의 목적과 사용자가 선택할 개발 환경에 대해 질문할 수 있어야한다.	완료
사용자 대답 기능	시스템의 질문에 대해 사용자는 시스템이 제공하는 선택지 중에서 골라 응답을 할 수 있어야한다.	완료
파일 다운로드	사용자는 본인의 요구에 따라 생성된 보일러 플레이트 파일을 다운 받을 수 있어야한다.	완료

2.2.2.3 프론트엔드

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

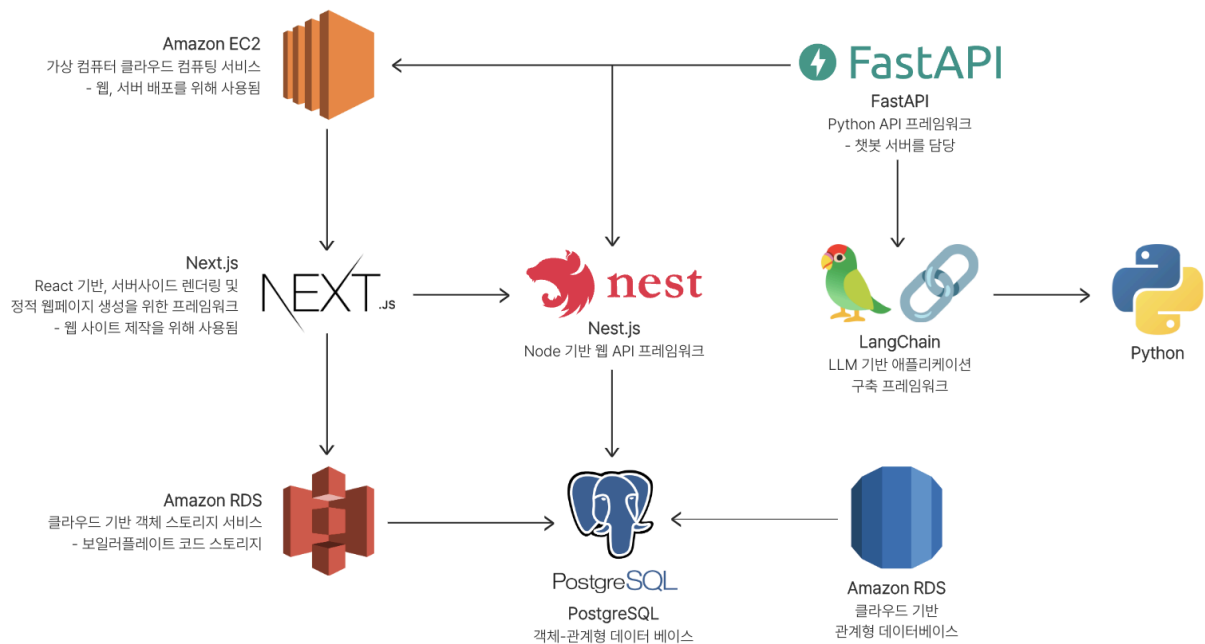
요구 기능	설명	완료 여부
회원가입	이메일, 비밀번호, 비밀번호 확인, 사용자 이름을 입력받아 요청을 보낸다. 이메일 및 비밀번호 형식을 확인해야 하며, 회원가입 완료 후 로그인 페이지로 리다이렉트 되어야 한다.	완료
로그인	회원가입시 사용한 이메일, 비밀번호를 입력받아 요청을 보낸다. 로그인 후 메인페이지로 리다이렉트 되어야 한다.	완료
로그아웃	로그아웃시 사이트에 저장된 쿠키데이터와 전역변수로 저장된 유저 정보를 초기화 한다. 로그아웃 후 로그인 페이지로 리다이렉트 되어야 한다.	완료
비밀번호 재설정	이메일을 입력 받아 초기화된 비밀번호를 이메일로 전송할 수 있도록 요청을 보낸다. 초기화된 비밀번호를 새 비밀번호를 입력받아 재설정할 수 있도록 한다.	완료
사용자 프로필 Modal	로그인 데이터를 확인하여 사용자의 이름, 이메일을 표시하며, 내 보일러 플레이트 보관함으로 이동할 수 있도록 한다.	완료
내 보일러 플레이트 보관함	사용자가 만든 보일러 플레이트 목록을 보여 주고, 이를 다시 다운로드할 수 있도록 한다. 프로젝트의 제목 및 생성 날짜를 표시하고, 보일러 플레이트 삭제 기능을 추가한다.	완료
보일러 플레이트 생성	사용자가 프로젝트에서 사용할 기술 스택 옵션들을 선택한다. 선택한 내용을 json 형태로 정리하여 요청을 보낸다. 생성된 보일러 플레이트 코드를 다운로드할 수 있도록 한다.	완료
챗봇 채팅	사용자가 입력한 내용을 챗봇 서버로 전달한다. 챗봇의 응답은 라이브 스트리밍 방식으로 한글자씩 실시간으로 보여질 수 있도록 한다.	완료
페이지 라우팅	사이트는 다음의 라우팅 기능이 있어야 한다. - 헤더 로고 클릭 시 메인 페이지로 라우팅 - 로그인 정보 없이 로그인 정보가 필요한 페이지에 접속했을 경우, 로그인 페이지로 리다이렉트	완료

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

2.2.3 시스템 비기능(품질) 요구사항

제목	설명	달성/미달성
보안성	시스템은 사용자 비밀번호 정보에 bcrypt 암호화를 적용해야한다.	달성
사용성	시스템 주요 기능은 사용자 테스트에서 5분 이내로 완료할 수 있다.	달성
유지보수성	변수와 함수명이 작성 의도가 명확히 드러나도록 클린코드를 작성하여 유지보수가 용이하도록 한다.	달성

2.2.4 시스템 구조 및 설계도



[그림46] 아키텍처

1. Next.js

서버 사이드 렌더링, 정적 웹 페이지 생성등의 기능을 제공하는 React 기반의 웹 개발 프레임워크입니다. 프로젝트에서 웹페이지를 제공해주는 Client 코드 작성을 위한 기술 스택입니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

2. Nest.js

Node.js 기반의 웹 API 프레임워크로 Typescript 기반으로 동작하며 백엔드 파트의 코드 작성을 위한 기술스택입니다.

3. FastAPI

FastAPI는 현대적이고, 빠르며 파이썬 표준 타입 힌트에 기초한 Python 의 API를 빌드하기 위한 웹 프레임워크입니다. 챗봇 서버를 담당합니다.

4. PostgreSQL

객체 관계형 데이터베이스이며 Nest.js와의 연동성이 좋으며 데이터의 유형, 확정성, 동시성 및 데이터 무결성에 있어 유연성이 더 좋은 특징이 있으며 TypeORM 을 사용하여 Nest.js 와 연동되어 있습니다.

5. AWS S3

데이터를 버킷 내의 객체로 저장하는 클라우드 기반 객체 스토리지 서비스입니다. 프로젝트에서 제공되는 보일러 플레이트 코드와 기타 리소스 및 에셋을 저장하는 스토리지 역할을 담당합니다.

6. AWS EC2

가상 클라우드 컴퓨팅 서비스이며 탄력성이 좋고 사용자가 인스턴스를 완전 제어 가능하여 편리한 웹, 서버 배포가 가능한 서비스이다. 클라이언트와 서버의 코드를 배포하는 파트를 담당합니다.

7. AWS RDS

클라우드 기반의 간편한 데이터베이스 관리형 서비스이며 PostgreSQL 데이터베이스를 클라우드에 배포하는 파트를 담당합니다.

8. LangChain

LangChain은 대규모 언어 모델(LLM)을 기반의 애플리케이션 구축을 돕는 오픈소스 프레임워크로 fastAPI 와 연동하여 파이썬으로 개발된 챗봇을 연동해주는 역할을 담당합니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

2.2.5 활용/개발된 기술

2.2.5.1 인공지능

Python	인공지능 및 머신러닝 개발에 필수 언어
LangChain	LLM 모델을 사용하여 애플리케이션을 쉽게 구축할 수 있도록 설계된 오픈 소스 프레임워크

2.2.5.2 백엔드

Node.js	V8 엔진 기반 JavaScript 런타임
NestJS	NodeJS기반 웹API프레임워크
PostgreSQL	객체 - 관계형 데이터베이스
AWS EC2	'비기너게인' 프로젝트 코드 배포용 가상 클라우드 컴퓨팅
AWS S3	서버를 통해 생성된 보일러플레이트를 저장 용도의 스토리지객체 스토리지 서비스
AWS RDS	프로젝트에 필요한 데이터 관리용도의 클라우드 관계형데이터베이스 관리형 서비스
FastAPI	챗봇-서버 연동 용도의 비동기 웹 API를 구축 목적의 Python 웹 프레임워크

2.2.5.3 프론트엔드

Next.js	서버 사이드 렌더링(SSR) 및 정적 웹사이트 생성(SSG) 구현을 위한 React 기반 프레임워크
Tailwind CSS	Utility-first 접근 방식을 사용하는 CSS 프레임워크
Styled-component	JavaScript 내에서 스타일링을 관리할 수 있게 해주는 CSS-in-JS 라이브러리 Tailwind css로 구현하기 복잡한 style을 Styled-component로 개발 진행

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

Axios	브라우저와 Node.js 환경에서 사용할 수 있는 Promise 기반의 HTTP 클라이언트 라이브러리. 백엔드 API 호출시에 사용
Recoil	Facebook에서 개발한 상태 관리 라이브러리. 프로젝트에서 전역 변수를 관리하기 위해 사용

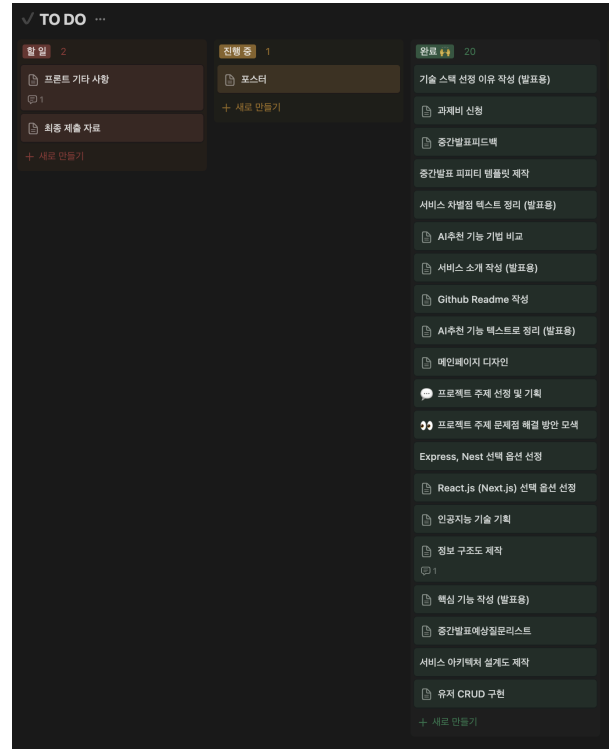
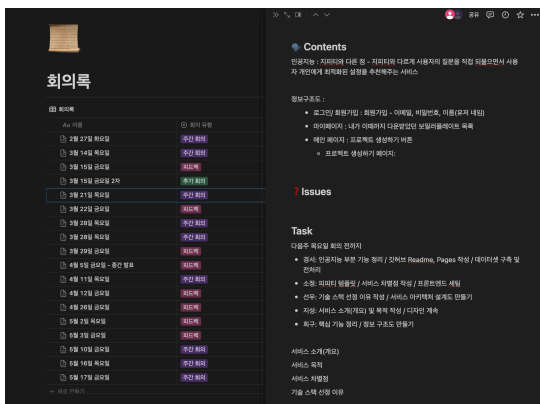
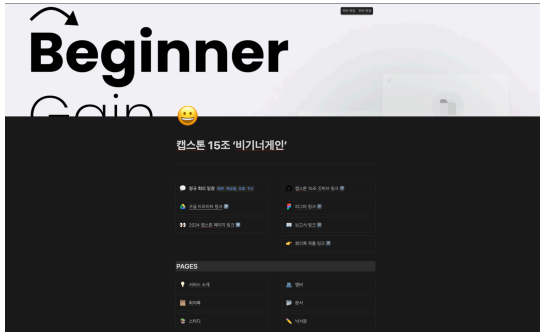
2.2.6 협업 내용

Notion을 활용한 협업 : 이번 프로젝트에서 문서 공유 위키형 협업툴 **Notion**을 활용하여 협업을 진행하였습니다.

매주 정해진 시간에 정기회의를 진행했으며, 프로젝트 진행 중 효율적인 협업과 각 팀원의 업무 관리를 목표로 **Notion**을 활용하였습니다. 이번 프로젝트를 진행하며 **Notion**을 활용한 구체적인 방법은 다음과 같습니다.

1. 회의 및 일정관리 : 매주 정해진 시간에 정기 회의를 했으며 회의 일정을 **Notion**에 공지하였고 캘린더 기능을 사용하여 중요한 마일스톤과 프로젝트 진행을 관리하였습니다.
또한 매 주 진행된 정기 내용을 회의록에 기록해두어 회의 후에도 내용을 쉽게 확인할 수 있었습니다.
2. To-Do관리 : **Notion**의 To-Do리스트를 활용하여 프로젝트 작업 상황을 쉽게 확인할 수 있도록 관리하였습니다. To-Do리스트에서 작업 상태를 세 단계로 구분하여 진행하였습니다. 진행해야 할 작업들을 '할 일' 목록에 통합하여 팀원들이 쉽게 확인할 수 있도록 했으며, 현재 진행 중인 작업을 별도로 관리하며 각 작업의 세부 진행 상황을 파악할 수 있었습니다. 그리고 완료된 작업을 정리한 목록을 확인하며 프로젝트가 일정에 맞게 진행중인 지 파악하며 프로젝트의 진행상황을 점검하였습니다. 또한 각 업무에 담당 팀원의 이름을 기재해두었고 **notion**의 필터 기능을 활용하여 자신의 업무를 쉽게 파악할 수 있도록 하였습니다.
3. 외부 저장소 통합 : 프로젝트 작업물이 저장되어있는 구글 드라이브, Github, Figma 등의 외부 저장소들을 **Notion**에 통합하여 팀원들이 필요한 저장소에 쉽게 접근할 수

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22



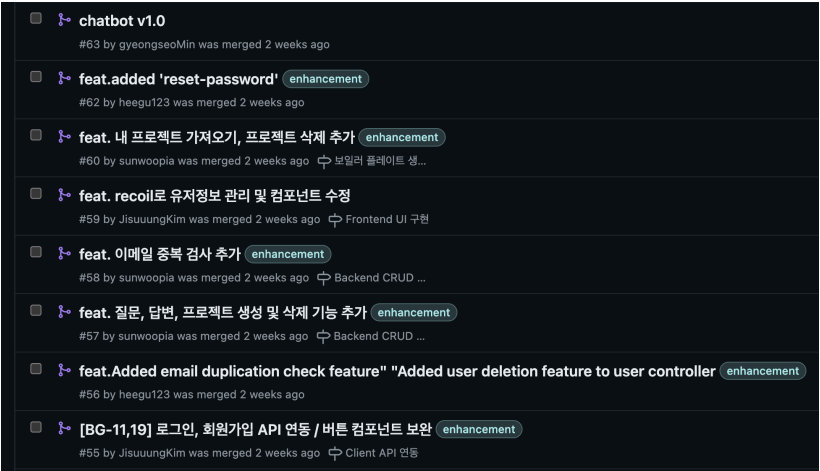
있도록 하였습니다.

Notion을 활용하여 일정 관리, 작업 할당, 프로젝트 진행상황 추적 등 여러 측면에서 큰 도움을 받을 수 있었습니다. 협업 툴을 사용하며 체계적인 작업 관리와 팀원들간의 효율적인 소통을 하며 프로젝트를 성공적일 수행할 수 있었습니다.

[그림47] 프로젝트 중 Notion 활용

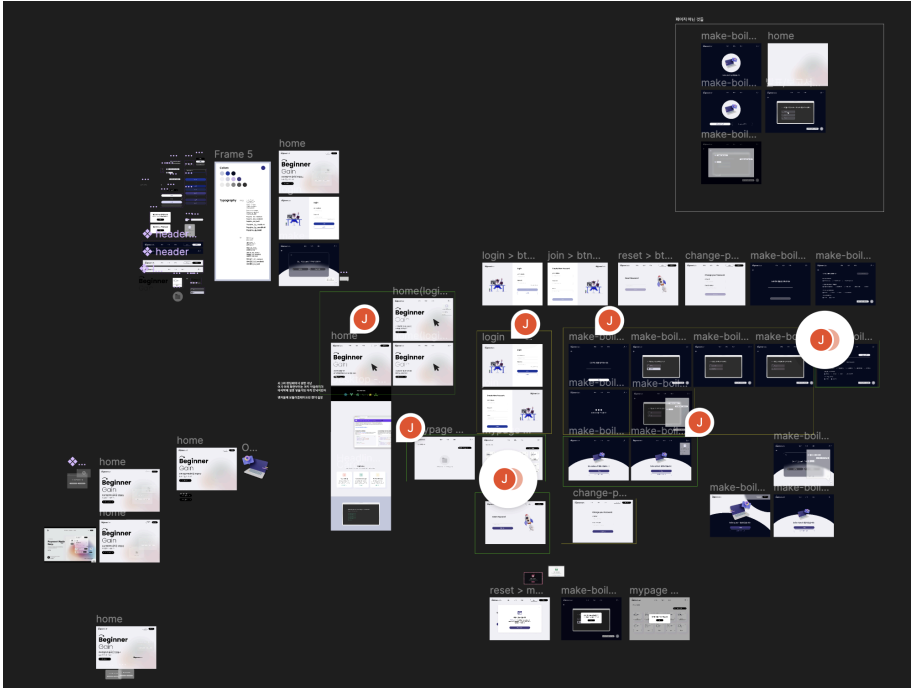
GitHub를 활용한 협업 : 저희 팀이 소규모 예자일 방식의 개발팀이고 제품이 단일 릴리즈 버전으로 운용된다는 점을 고려하여 Github Flow 방식을 사용하여 관리하였으며 확실한 이슈와 마일스톤 관리를 통해 개발 하는 과정에서 발생하는 충돌을 방지하였습니다. Issue 작성을 통해 태스크를 관리하였으며 Issue와 Pull Request 를 연동하여 효율적으로 태스크를 추적하고 관리하였습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22



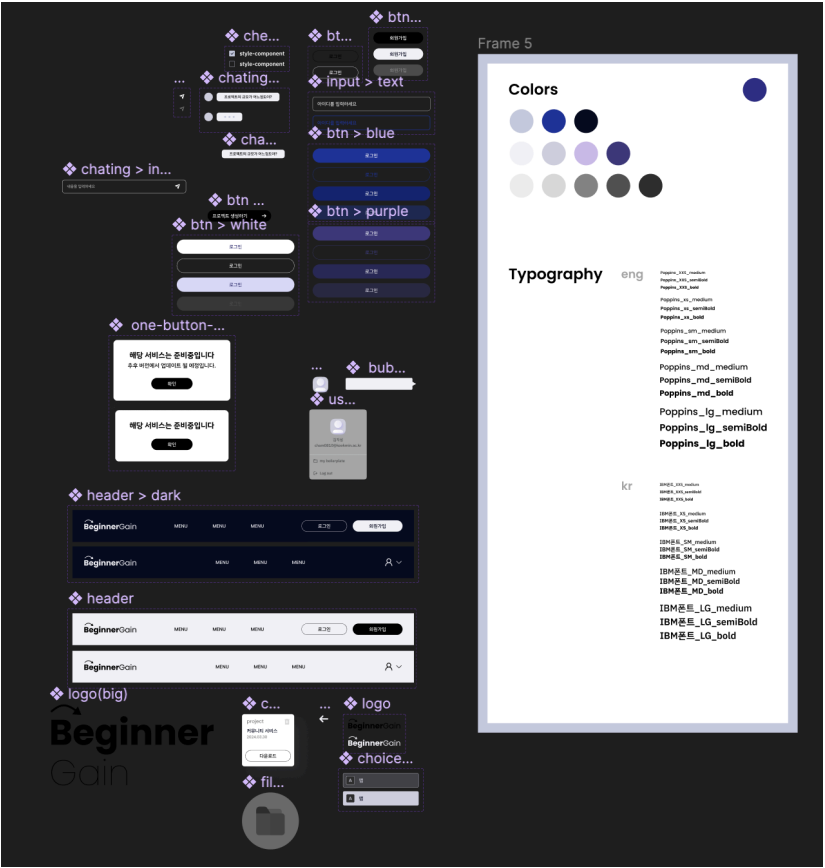
[그림48] 프로젝트 중 GitHub를 활용한 협업

Figma를 활용한 협업 : 웹 기반 UI/UX 디자인 협업 툴인 Figma를 통해 개발할 서비스의 시각적 프로토타입을 디자인했습니다. 디자인 공유를 실시간으로 할 수 있어 빠른 소통을 통해 디자인을 개선했습니다.



[그림49] figma 전체화면

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22



[그림50] component 관리 및 font, color 세팅

재사용성이 높은 컴포넌트를 따로 분리하고 관리했습니다. 또한 폰트나 색상포맷을 디자인 과정에서 정의해두고, 이를 **tailwind config** 파일에 세팅해놓음으로써 UI 개발의 효율을 높였습니다. 인터랙티브한 웹을 위해 버튼 컴포넌트를 확장하여 **hover**나 **disabled property**를 추가했습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노폐인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

2.2.7 결과물 목록

Github페이지	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스 capstone-2024-15
Github	https://github.com/kookmin-sw/capstone-2024-15/
중간발표 보고서	 중간보고서-비기너게인.docx
중간발표 자료	 15조_비기너게인_최종발표자료.pptx
최종발표 보고서	 15조_비기너게인_수행결과보고서
최종발표 자료	 15조_비기너게인_최종발표자료.pptx
시연 영상	 15조 비기너게인 시연영상
포스터	 15조_비기너게인-포스터.pptx
접속 주소	https://www.beginergain.com/

2.3 기대효과 및 활용방안

기대효과

비기너게인 서비스로 인해 기대되는 효과들은 다음과 같습니다.

1. 초보 개발자들이 프로젝트 시작 시 진입 장벽을 낮춰 주어 쉽고 빠르게 프로젝트를 시작할 수 있도록 도울 수 있습니다.
2. 제공된 보일러 플레이트 코드로 프로젝트를 시작함으로써 서비스 개발 시간을 단축시켜줄 것입니다.
3. 보일러플레이트 코드를 사용하여 개발 중 코드의 일관성을 유지하고, 가독성을 높여 유지 관리를 수월하게 할 수 있도록 할 것입니다.

활용방안

현재 저희 서비스는 웹 사이트와 **GPT Store** 두 가지로 운영되고 있습니다.

웹 사이트의 통한 서비스 사용의 장점은 챗봇과의 대화를 통해 지식을 바로 습득하고 프로젝트에 필요한 보일러플레이트를 생성할 수 있다는 점입니다. 이를 통해 사용자가

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

챗봇의 가이드라인을 따라 빠르게 초기 설정을 완료할 수 있도록 돕는 데에 초점을 맞추고 있습니다.

GPT Store에 배포된 챗봇 활용의 장점은 웹 사이트의 챗봇보다 더 많은 기능을 제공한다는 점입니다. 외부 **API** 연결을 통해 사용자에게 더 다양한 답변을 제공할 수 있고, 보일러플레이트 다운로드를 위해 웹 사이트인 비기너게인 가입을 유도하는 방법 중 하나로 활용될 수 있습니다.

향후 저희는 기술 선택 옵션을 확장하여 더 많은 기술 스택의 보일러플레이트를 생성할 수 있게 할 예정입니다. 또한, 보일러플레이트 코드 판매 플랫폼 및 디자인 템플릿까지 공유 서비스를 통해 개발자뿐만 아니라 디자이너와 비개발자들도 사용할 수 있게 서비스가 확장될 수도 있습니다.

또한, 웹사이트 챗봇은 서비스 사용자 확보 후, 사용자들의 피드백을 반영하여 사용자에게 더욱 나은 답변을 제공할 예정입니다. **GPT store**에 배포한 챗봇은 최신 기술 뉴스나 트렌드를 요약해서 제공할 예정이며, 간단한 퀴즈를 통해 사용자가 자신의 학습 진도나 실력을 확인할 수 있도록 할 예정입니다. 이외에도 다양한 기능을 제공할 수 있도록 **API**를 연결하여 사용자에게 더욱 많은 정보를 알려주는 코딩 친구 서비스로 발전할 계획입니다.

5월 12일 이후로 텍스트뿐만 아니라 이미지, 음성 등 모든 형태의 입력을 이해하고 처리할 수 있는 멀티모달 모델 **GPT-4o**모델이 공개되었습니다. 이 점을 활용하여 모바일로 음성을 입력하면, 저희가 프롬프팅한 챗봇이 답변을 줄 수 있도록 활용할 수 있습니다. 이러한 기능을 활용하여 저희는 사용자와 직접 상호작용하며 코딩 친구가 될 수 있도록 서비스를 더욱 고도화시킬 예정입니다. 현재는 **GPT Store**가 **GPT Plus**를 구독하는 사람들만 사용할 수 있습니다. 계속 **GPT Plus** 구독 이용자들이 늘어감에 따라 성장 가능성은 매우 클 것으로 예상됩니다. **GPT Store** 자체만으로는 수익을 내진 않지만, 아웃링크를 유도하는 방식으로 수익 창출을 하는 방향으로 활용될 수도 있습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

3 자기평가

저희 팀은 캡스톤 디자인 2주차에 주제를 변경하게 되어 시간이 조금 부족한 상태로 프로젝트를 시작했습니다. 초반 주제 선정부터, 기획 단계에서 약간의 어려움이 있었지만 저희는 빠르게 주제를 구체화하고 개발하여 약 2개월만에 성공적으로 프로젝트를 끝냈습니다. 특히, 개발 과정에서 GitHub Issue와 Pull-Request를 효과적으로 관리하여 단 한 번의 충돌도 발생하지 않았습니다.

팀 구성 및 역할 분담

저희 팀은 프론트엔드 2명, 백엔드 2명, 챗봇 1명으로 역할을 나누어 개발을 진행했습니다. 프로젝트의 규모와 요구사항에 맞추어 역할을 적절히 잘 분배하여 큰 갈등없이 프로젝트를 수행하였습니다. 각자 맡은 바를 충실히 수행하며, 팀원간의 원활한 소통과 협력으로 프로젝트를 성공적으로 마칠 수 있었습니다.

프로젝트 관리와 협업

협업 툴은 크게 각종 업무를 위한 노션, 디자인 작업을 위한 피그마, 개발 관리를 위한 깃허브를 사용하였습니다. 저희 팀이 소규모 에자일 방식의 개발팀이고 제품이 단일 릴리즈 버전으로 운용된다는 점을 고려하여 Github Flow 방식을 사용하여 관리하였으며 확실한 이슈와 마일스톤 관리를 통해 개발 하는 과정에서의 충돌이 단 한번도 없었습니다.

산출물 제작 및 발표 준비

캡스톤디자인을 통해 개발뿐만 아니라 주간보고서, 팀보고서, 포스터, 발표 자료 등 다양한 산출물을 제작하는 경험을 쌓았습니다. 각종 문서를 작성하며 프로젝트를 체계적으로 정리하고, 관리하여 결과를 효과적으로 전달하는 방법에 대해 많이 배웠습니다. 또한, 이 부분에서도 포스터나 발표자료를 잘 만드는 학우는 그 역할을 수행했고, 영상 또는 보고서를 잘 만드는 학우는 각자의 역할을 잘 수행했습니다.

아쉬운 점

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

지난 시간들을 돌아해보면, 초기에 주제가 바뀌는 일 없이 선정을 빠르게 했더라면, 지금보다 더 많은 기능을 넣어 더욱 완성도 높은 결과물을 만들었을 것 같습니다.

향후 개선점

향후 개선 방향은 유저 피드백을 기반으로 우선적으로 방향성을 정할 것이며 추가로 유저들의 사용성에 초점을 맞춰 유저들이 필요로하는 기능을 추가할 예정입니다. 또한 수익 구조를 개선하여 실제로 코드를 사이트 내에서 구매 및 판매, 그리고 공유하도록 하는 기능을 추가하여 비즈니스 모델을 확장할 것입니다.

개인별 프로젝트 회고

김선우: 제대로 된 프로젝트를 진행함에 있어 프로젝트를 리딩하는 능력을 많이 키웠고 실제로 개발자들의 역량에 맞춰 임무를 수행하는 것에 대해 배웠습니다. 생각했던 것 보다 한명 한명이 신경 써야하는 것들을 모두 규칙으로 정하여 작업하는 것이 정말 중요했으며 프로젝트를 진행해준 팀원들이 잘 따라와준 것에 대해 너무 감사하는 마음입니다. 이후 프로젝트를 함에 있어 문제해결 능력의 중요성을 깨달았고 특히 주분야가 아닌 다른 분야에 대해 공부를 해보고싶다는 생각도 했습니다. 이후에도 같은 멤버로 프로젝트를 또 해보고 싶습니다.

강희구: 이번 캡스톤 디자인 프로젝트 진행 중에 주제를 변경하고 예정보다 빠듯해진 일정에서도 팀원 간 소통을 잘 하며 성공적으로 프로젝트를 마무리 했습니다. 팀원 간 역할을 명확히 나누고 여러 협업 툴을 효과적으로 활용하며 프로젝트 상황을 공유하고 팀원 간 원활한 소통으로 프로젝트동안 어떠한 충돌 없이 프로젝트를 진행할 수 있었습니다. 개발 과정은 새로운 기술을 사용해볼 수 있었고, 다른 역할을 담당하는 팀원들의 작업을 보며 다른 분야를 학습해볼 수 있었습니다. 처음 주제 선정부터 발표 준비까지 체계적인 팀 프로젝트 경험을 쌓을 수 있었던 한 학기였습니다. 향후 이번 프로젝트에서 더 나아가 기능을 추가하고 프로젝트 중에 있었던 여러 피드백들을 반영하여 서비스를 더 개선해보고 싶습니다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

김지성: 체계적으로 프로젝트를 설계하고 관리하는 것이 얼마나 중요한지 배울 수 있었습니다. 특히 기획 단계에서부터 서비스의 시장성이나 타겟층에 대해 팀원들과 끊임없이 고민하고 보완한 덕분에 프로젝트를 완성도있게 마무리 할 수 있었다고 생각합니다. 팀원 모두 능동적으로 프로젝트에 참여하여 효율적으로 회의를 이끌어갔고 적절한 피드백을 주고받으며 다같이 성장할 수 있었던 프로젝트였습니다. 또한 개발 과정 중에서도 적절한 깃허브 전략을 활용하여 협업과정에서 큰 문제없이 깔끔하게 코드를 작성할 수 있었습니다. 더 나아가 코드를 짤 때에도 더 효율적인 코드는 무엇인지, 사용자 경험을 향상시키기 위해 어떤 방법을 써야할지 찾아보며 공부했던 것들이 큰 도움이 되었습니다.

민경서: 캡스톤디자인 초반에 주제를 바꾸게 되어 새로운 프레임워크와 지식을 습득해야 했습니다. 공부를 하며 개발을 해야 하는 상황이었기 때문에, 시작하기에 앞서 두려움이 있었습니다. 하지만, 꾸준한 학습과 팀원들간의 원활한 소통을 통해 문제를 해결해 나갈 수 있었습니다. 여러 가지 협업 툴을 사용하고 깃허브를 체계적으로 사용하면서 전체적인 프로젝트의 과정을 배우고 이해할 수 있었습니다. 각자의 역할을 잘 수행하고 프로젝트를 함께 잘 마무리해준 팀원들에게 너무 감사하고 다음에도 함께 프로젝트를 하고 싶습니다. 추가적으로, 캡스톤디자인을 통해 최신 트렌드 기술에 대해 습득할 수 있어서 큰 도움이 되었습니다. 앞으로 계속 업데이트 되는 LangChain프레임워크의 새로운 기능들을 계속해서 사용해볼 예정입니다. 이번 프로젝트는 시간이 부족하여 다른 LLM 을 사용하지 못했지만, 다음 프로젝트에서는 다른 LLM을 사용하여 직접 학습을 시켜보며 결과를 도출해보고 싶습니다.

이소정: 캡스톤 디자인 프로젝트는 제가 학부 4학년을 진행하며 진행했던 프로젝트 중, 가장 많은 부분을 고려하여 개발을 진행한 프로젝트입니다. 실제로 서비스를 운영하고 체계적인 평가를 받아야 하기에 사용자들이 서비스 이용에 불편함을 느끼지는 않을 지, 서비스의 성능은 어떠할 지, 실제 운영이 우리가 예상한 대로 흘러갈 지에 대해 초점을 맞추며 개발을 진행하였습니다. 이 과정에서 제가 자주 사용하던 코드가 서비스 성능에 문제를 줄 수 있다는 것을 알게 되었고, 이를 개선하는 방법에 대해서도 공부하게

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

되었습니다. 또한, 팀원끼리 이런 문제들을 고민하고 해결법을 찾는 과정에서 모두 함께 성장할 수 있었던 것 같습니다. 팀원들끼리 많은 점을 고민하며 함께 만들어 낸 서비스인 만큼, 이 서비스에 많은 애정이 생겼습니다. 캡스톤 디자인이 끝난 이후에도 이 서비스를 발전시키기 위해 계속해서 함께 개발을 진행하고 싶습니다.

4 참고 문헌

번호	종류	제목	출처
1	기술 문서	LangChain docs	https://python.langchain.com/docs/get_started/introduction
2	기술 문서	LangServe docs	https://python.langchain.com/v0.1/docs/langserve/
3	기술 문서	랭체인LangChain 노트 한국어 튜토리얼	https://wikidocs.net/book/14314
4	기술 문서	Principled instructions are all need for questioning LLaMA-1/2, GPT-3.5/4	https://arxiv.org/abs/2312.16171v1

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

2	기술 문서	Next.js docs	https://nextjs.org/docs
3	기술 문서	Nest.js docs	https://docs.nestjs.com/
4	기술 문서	postgreSQL docs	https://www.postgresql.org/docs/
5	기술 문서	Nestjs-TypeORM docs	https://docs.nestjs.com/recipes/sql-typeorm
6	기술 문서	TypeORM docs	https://typeorm.io/
7	기술 문서	Tailwind CSS docs	https://tailwindcss.com/docs/
8	기술 문서	Axios docs	https://axios-http.com/kr/docs

5 부록

5.1 사용자 매뉴얼

1. <https://www.beginergain.com/>에 접속한다.
2. 화면 우측 상단의 [회원가입] 버튼을 클릭한다.
3. 필수정보인 '사용자 이메일', '비밀번호', '사용자 이름'을 입력하고 하단의 [회원가입]버튼을 클릭하여 회원가입한다.
4. 가입한 이메일, 비밀번호를 입력하여 로그인한다.
5. 시작 페이지 좌측 하단의 '프로젝트 생성하기' 버튼을 클릭한다.
6. 프로젝트 이름을 입력하고 개발 목적을 묻는 질문에 응답하며 보일러 플레이트를 생성한다. 어떤 응답을 선택해야할 지 어려운 경우에는 우측 하단의 '챗봇과 질문하기'를 활용하여 챗봇에게 질문을 하며 도움을 받는다.

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

7. 생성된 보일러 플레이트를 'my boilerplate'에서 다운로드 받아서 사용한다.

5.2 테스트 케이스

대분류	소분류	기능	테스트 방법	기대 결과	테스트 결과로그
유저	회원가입	회원가입	1) 상단 메뉴바에서 [회원가입] 버튼을 누른다 2) 회원가입 페이지에서 필수 정보를 입력하고 회원가입 버튼을 클릭한다.	데이터베이스에 회원 정보가 추가된다.	성공
	로그인	로그인	로그인 페이지에서 이메일과 비밀번호를 입력하고 로그인 버튼을 클릭한다.	사용자 계정으로 시스템에 로그인된다.	성공

 국민대학교 컴퓨터공학부 캡스톤 디자인 I	결과보고서		
	프로젝트 명	비기너게인(BeginnerGain) : 초보개발자들을 위한 보일러플레이트(boilerplate) 생성 서비스	
	팀 명	노페인 노게인	
	Confidential Restricted	Version 1.6	2024-MAY-22

	비밀번호 초기화	비밀번호 초기화	비밀번호 초기화 페이지에서 등록된 이메일 주소를 입력하고 [이메일 보내기] 버튼을 클릭한다.	비밀번호 초기화 이메일이 전송된다.	성공
챗봇	프레임워크 추천	사용자 요구에 적합한 프레임워크 추천 및 설명	1. 시작 페이지의 [프로젝트 생성하기]버튼을 클릭한다. 2. 프로젝트 명을 입력 후, [다음]버튼을 클릭한다. 3. 우측 하단의 '선택이 어렵다면 챗봇과 대화해보세요!' 버튼을 클릭한다. 4. 질문을 입력한다.	1. 사용자의 요청에 적합한 프레임워크를 추천한다. 2. 추가 요청이 필요한 경우 챗봇이 답변으로 추가 질문을 한다.	성공
보일러 플레이트 생성	질문 및 응답	사용자의 요청사항 전달	5. 시작 페이지의 [프로젝트 생성하기]버튼을 클릭한다. 6. 프로젝트 명을 입력 후, [다음]버튼을 클릭한다. 7. 화면에 출력되는 질문에 대한 사용자의 답을 응답선택지와 체크박스에 클릭하며 전달한다.	사용자의 요청을 반영한 보일러플레이트가 내 보관함에 추가된다.	성공
보관함	보일러 플레이트 추가	내 보관함에 보일러 플레이트를 추가한다.	[boilerplate]버튼을 클릭한다.	내 보관함에 새로운 보일러 플레이트가 생성된다	성공
	보일러 플레이트 다운로드	선택한 보일러 플레이트 파일을 저장한다.	우측 상단의 '내 정보'버튼을 클릭 후, 'My boilerplate'에서 선택한 프로젝트의 [다운로드]버튼을 클릭한다.	사용자의 기기에 해당 내용의 파일이 저장된다.	성공
	보일러 플레이트 삭제	선택한 보일러 플레이트 파일을 삭제한다.	우측 상단의 '내 정보'버튼을 클릭 후, 'My boilerplate'에서 선택한 프로젝트의 [삭제]버튼을 클릭한다.	내 보관함에서 해당 보일러 플레이트가 삭제된다.	성공