

## 서비스

### 사용자 맞춤형 통신사 요금제 추천 서비스

UNOA는 다양한 LG U+ 통신 요금제 중에서, 사용자의 통신 사용 패턴에 맞는 최적의 요금제를 쉽게 추천받을 수 있도록 도와주는 AI 기반 추천 플랫폼입니다.

## 개발 기간

2025. 06. 09 - 2025. 06. 27 (3주)

[서비스 링크](#)

[상세정보\(노션\)](#)

[디자인 링크\(피그마\)](#)

<https://github.com/UNOA-Project>

## 사용 기술 스택

 JavaScript  TailwindCSS  React

 Vite  JWT  Figma  HTML5

## 구성원 (5명)

김현우 / 송은재 / 심영민 / 홍성현 / 황주경

## 역할

### Design

프로토타입 제작 (figma)

UI/UX 디자인 [반응형\_Mobile, PC] (figma)

### FrontEnd

메인 페이지

요금제 챗봇 (50%)

요금제 비교 리스트

컴포넌트, 라우트 구조 설계

DB 설계

## 프로젝트 회고

- 이번 프로젝트를 통해 실시간 스트리밍, openAI API 연동, MongoDB 저장 등 다양한 기능을 직접 구현해보며 전체적인 서비스 흐름과 구조를 깊이 이해할 수 있었습니다.
- 사용자 중심의 기능 구현과 실시간 피드백 기반의 UI/UX 개선을 경험했습니다. AI 기능이 인터페이스에 적용되는 과정을 지켜보며 기술적 흥미와 개발에 대한 시야를 넓힐 수 있었습니다.
- 앞으로 다가오는 AI시대에 있어 챗봇을 사용하는 프로젝트는 매우 뜻 깊은 시간이였습니다. 앞으로 AI를 잘 활용할 수 있도록 더욱 공부하고 시대에 맞는 프론트엔드 개발자로 발전하고 싶습니다.

## 주요 시스템 구성 요소 및 기술 스택

### Frontend (React)

- 사용자의 채팅 입력을 받고, 메시지 UI를 렌더링
- Socket.IO 클라이언트로 서버와 실시간 통신
- GPT 응답 스트리밍 표시
- 

### Backend (Node.js + Express + Socket.IO)

- Socket.IO 서버: 클라이언트 연결/연결 해제 관리
- 이벤트 처리:
  - message → GPT 요청
  - stream-start / stream-chunk / stream-end → 실시간 스트리밍 전달
- OpenAI API 호출 및 응답 처리
- 필요 시 DB 연동 (MongoDB) → 대화 기록 저장, FAQ 저장 등

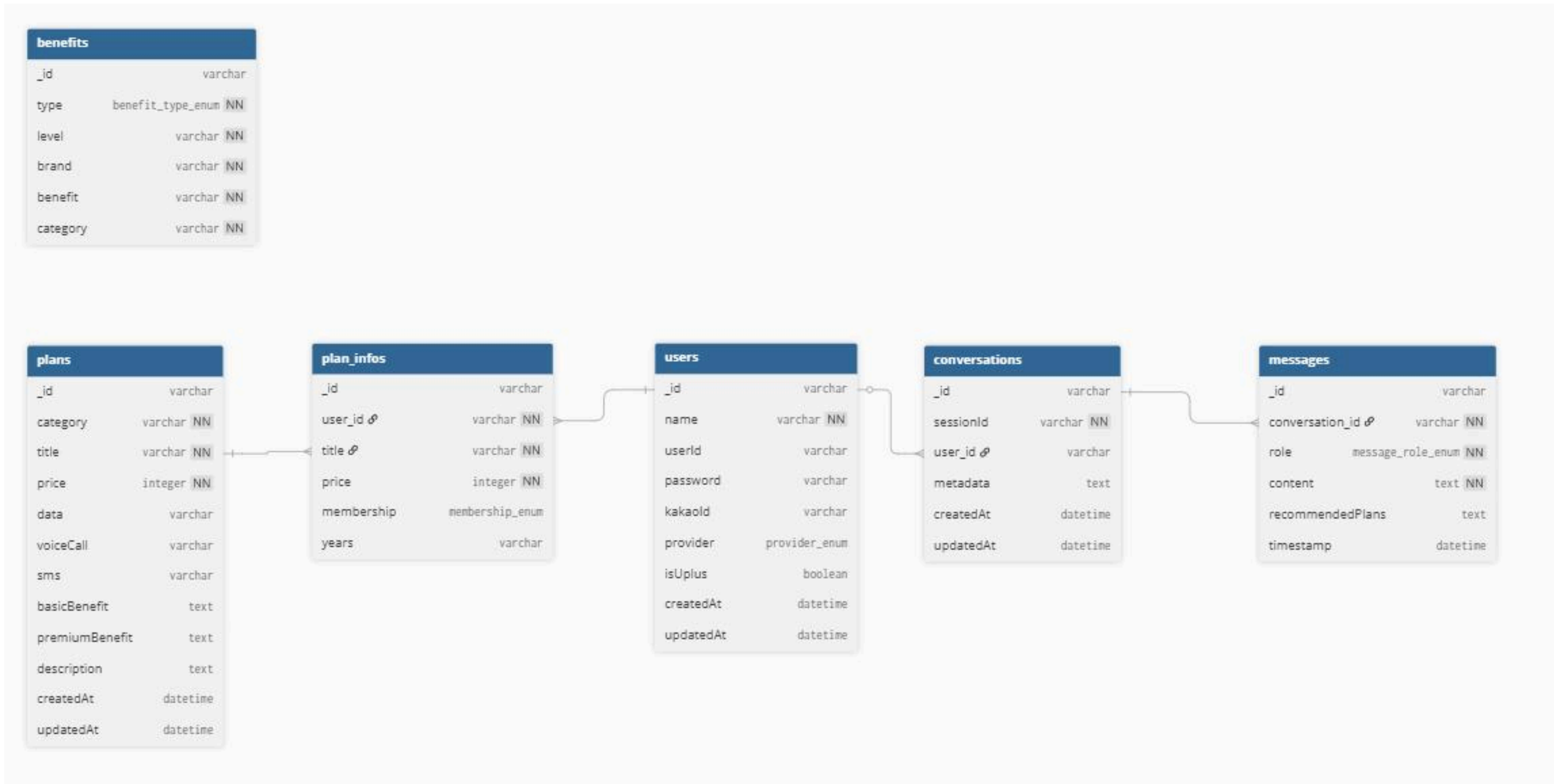
### OpenAI API (GPT)

- 질문 입력 → 모델 → 답변 스트리밍
- ChatCompletion 또는 Streaming API 사용 가능

### Database (MongoDB)

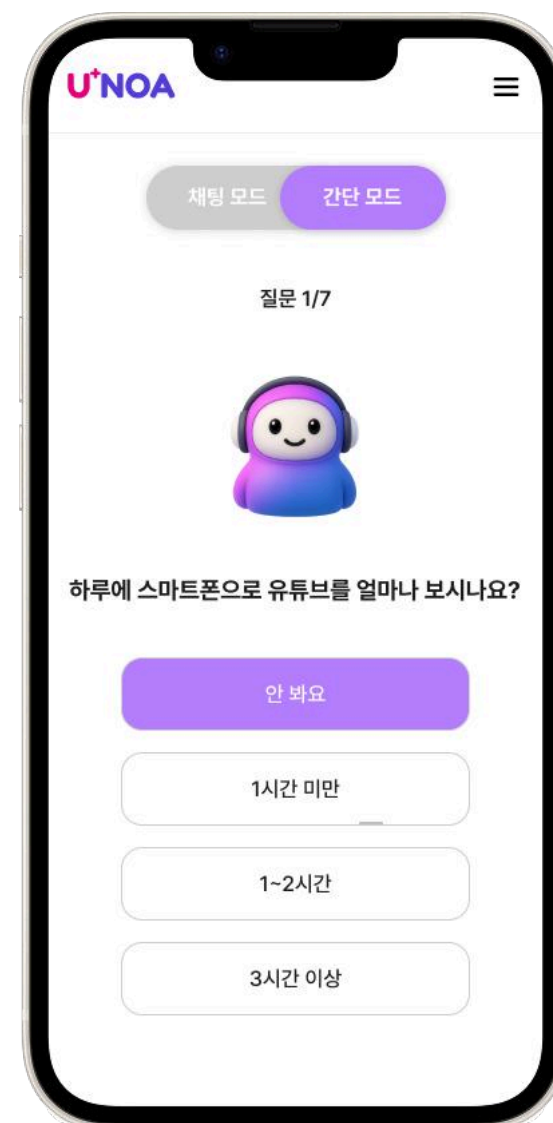
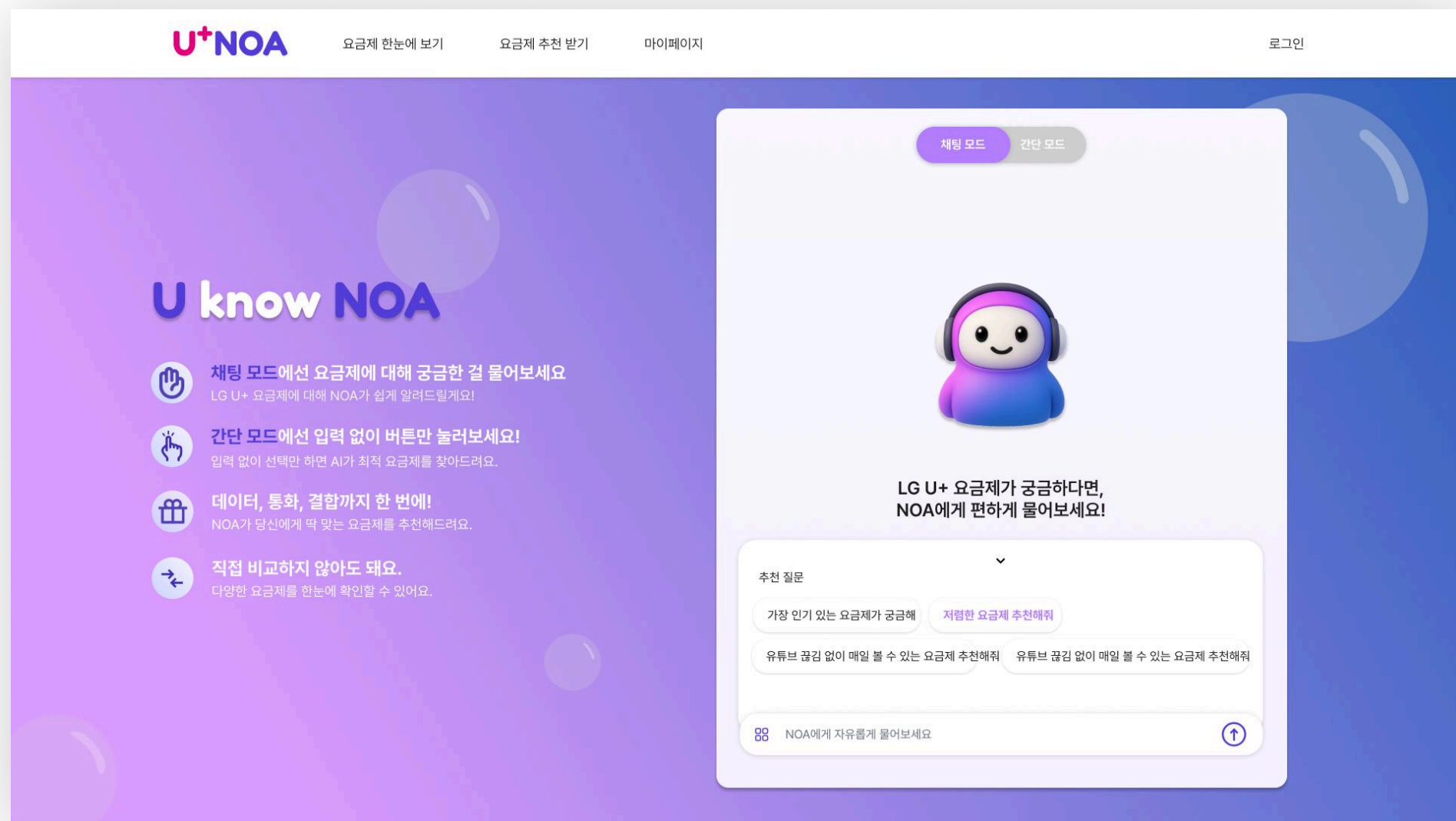
- 대화 기록 저장: conversation\_id

## ERD



## 챗봇 기능 채팅&간단

사용자가 쉽게 질문할 수 있도록 상황에 맞는 응답을 제시하고, 요금제 관련 질문을 하면 챗봇이 이를 분석해 맞춤 요금제를 추천



### 주요 기능 및 구현 기술

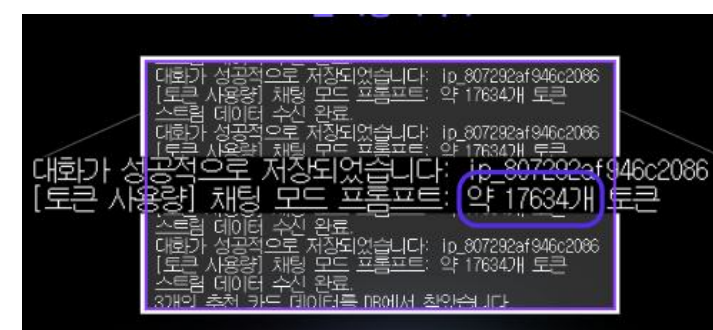
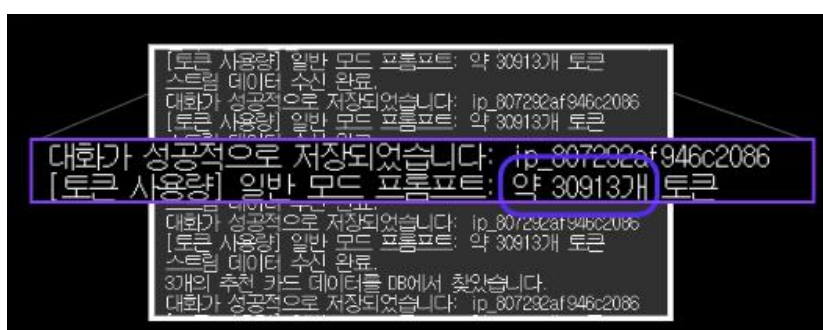
- 요금제 추천 및 자연어 분석:  
OpenAI의 GPT 모델을 활용해 사용자의 자연어 질문을 이해하고, 가장 적합한 요금제를 찾아 추천  
요금제 이름이 포함된 답변은 카드 형태로 시각화하여 사용자가 정보를 한눈에 파악 가능
- 실시간 응답 스트리밍:  
Socket.IO를 기반으로 응답을 실시간으로 스트리밍 처리해 사용자가 답변을 기다리는 시간을 최소화
- 간단 모드 :  
채팅을 치는 것보다 간단한 설문을 10가지 통해 사용자가 원하는 요금제 정보를 제공
- 리디렉션 응답:  
사용자의 질문 의도에 따라 특정 기능 페이지(예: 요금제 비교, 더보기 등)로 이동을 유도하는 리디렉션 응답을 구현하여 사용자에게 필요한 정보를 더 효율적으로 제공

### 기술적 도전 및 해결 방안

- 초기 개발 단계에서 GPT 호출 시 발생하는 응답 지연 문제를 해결하기 위해 두 가지 방안을 적용
  - 요금제 요약 캐싱:  
GPT에 전달할 요금제 정보를 사전에 요약하고 일정 시간마다 캐시를 갱신하도록 구성하여 불필요한 토큰 소비를 줄이고 응답 속도를 향상
- 대화 연속성 유지:  
메시지들을 모은 전체 대화를 conversation\_id에 사용자의 IP, 사용 시간 부가 정보를 함께 저장하여 페이지가 다시 랜더링 하더라도 메시지를 기억하게함

### 주요 성과

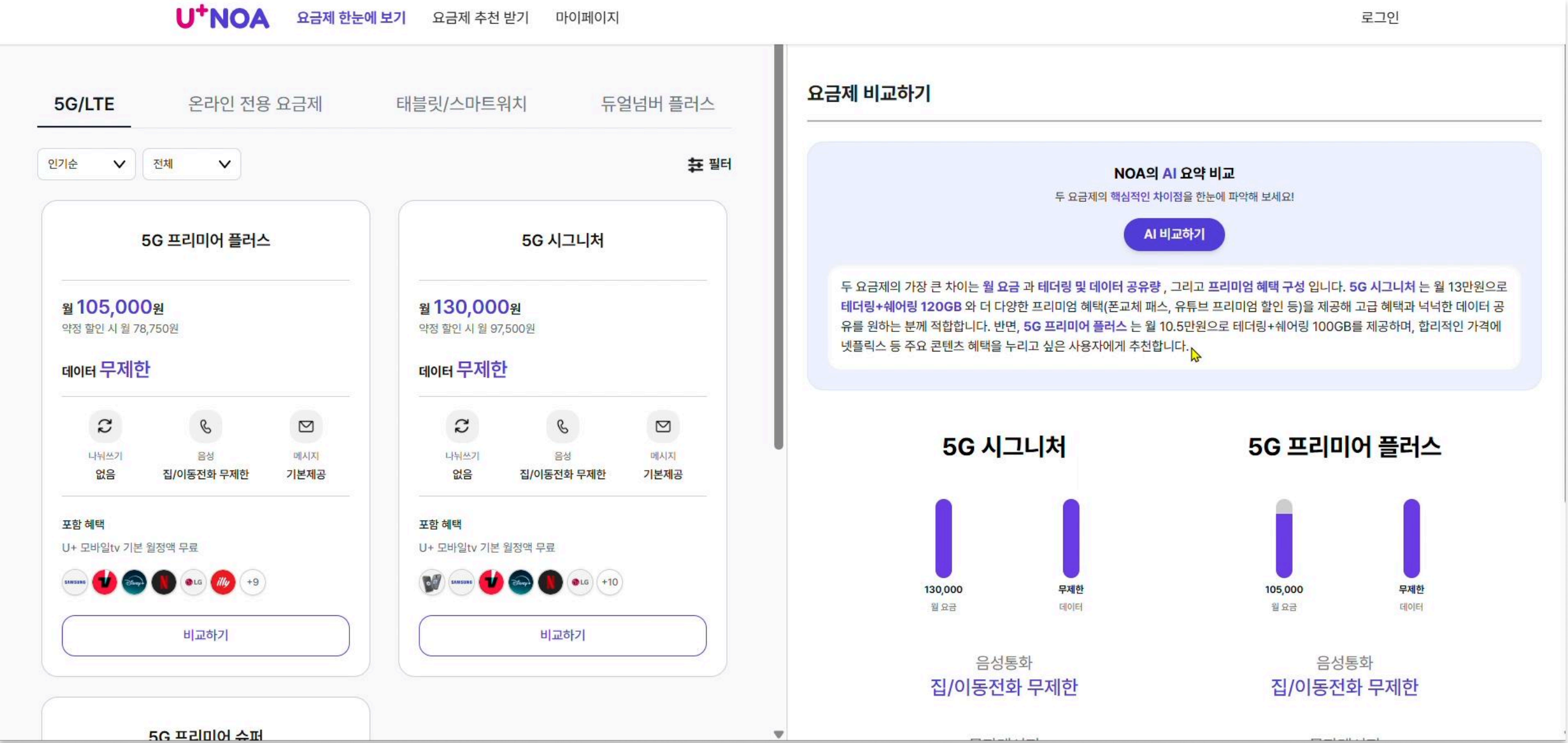
- 비용 절감: 불필요한 GPT API 호출을 줄여 OpenAI 토큰 소비량을 약 40% 절감했습니다.
- 사용자 편의성 증대: 사용자가 원하는 정보를 빠르고 쉽게 찾을 수 있도록 다양한 기능을 제공하여 사용자 만족도를 높였습니다.





요금제 비교

요금제 정보를 제공하며, AI와 그래프를 통한 요금제 비교 서비스 제공



주요 기능 및 구현 기술

- **요금제 비교 선택 시 그래프 기반 비교 기능**
  - 사용자가 선택한 요금제를 막대/라인 그래프 기반으로 시각적으로 비교할 수 있도록 구현
  - 데이터 소비량, 음성/문자 제공량, 부가 혜택 등을 직관적으로 파악 가능
- **요금제 AI비교 서비스:**
  - 선택한 요금제를 AI가 분석해 단순 스펙 나열이 아닌 “가성비”, “무제한 여부” 등 사용자 맞춤형 비교 설명을 제공.

기술적 도전 및 해결 방안

- **요금제 비교 그래프 기능:**

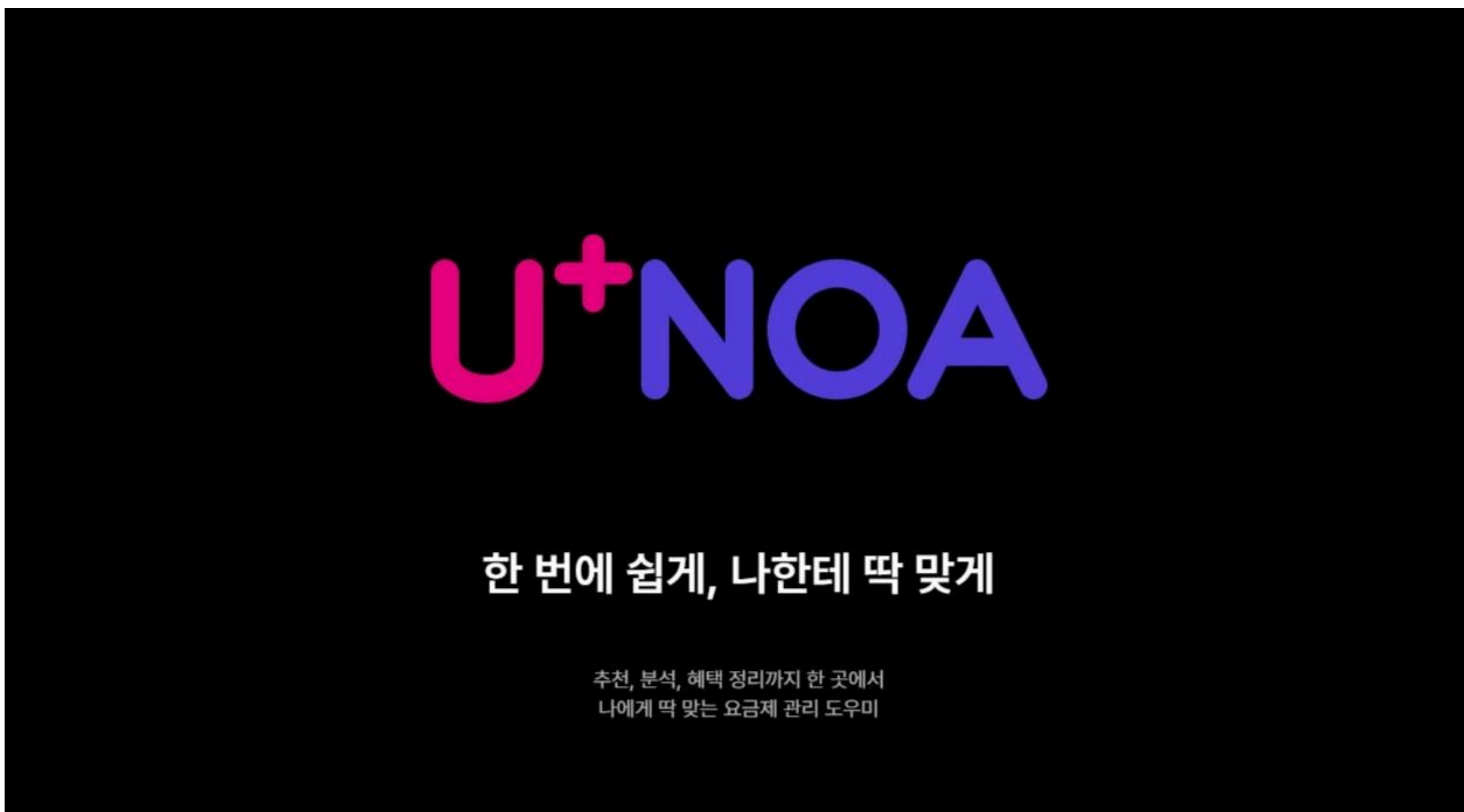
사용자가 선택한 요금제를 막대/라인 그래프 기반으로 시각화하여 데이터 사용량, 음성/문자 제공량, 부가 혜택을 직관적으로 확인할 수 있도록 구현. 통신사마다 상이한 데이터 단위와 항목을 표준화하고, 다중 요금제 비교 시에도 성능 저하 없이 렌더링될 수 있도록 최적화 로직을 설계. 이를 통해 사용자가 복잡한 스펙을 단순한 시각적 요소로 쉽게 이해할 수 있는 경험을 제공함.
- **클라이언트-서버 간 데이터 동기화:**

사용자가 선택한 요금제를 AI 기반 비교하기 기능을 통해 분석하고, 단순 스펙 나열이 아닌 “가성비”, “무제한 여부” 등 사용자의 니즈에 맞춘 맞춤형 비교 설명을 생성. 모든 대화 내역을 MongoDB에 구조화하여 저장(role, type, content, timestamp 포함) 하고, 세션 단위로 구분하여 추후 데이터 분석 및 추천 모델 학습이 가능하도록 설계. 이를 통해 사용자가 챗봇과 나눈 대화를 단순 기록이 아닌 지속적으로 학습·개선되는 사용자 경험으로 확장할 수 있도록 함. 이거 좀 줄여줄래

주요 성과

- 사용자 만족도 향상: 단순히 요금제를 나열하여 보이는 것보다 비교하면서 사용자에게 유의미한 정보를 제공하였습니다.
- 효율적인 고객 지원: 사용자가 이전 상담 내역을 즉시 파악할 수 있어, 보다 정확하고 신속한 정보탐색이 가능해졌습니다.
- 데이터 활용 기반 마련: 저장된 대화 데이터는 향후 챗봇 기능 개선, FAQ 항목 추가, 사용자 행동 분석 등에 활용될 수 있는 중요한 자산이 되었습니다.
- AI 요금제 비교

## 메인 페이지



### 주요 기능 및 구현 기술

- **Framer-motion을 활용한 SVG Line Drawing**

메인 화면 및 주요 섹션에 SVG 라인 드로잉 애니메이션을 적용하여, 사용자에게 시각적 재미와 몰입감을 제공. 단순 정적인 이미지가 아닌, 움직이는 라인과 인터랙션을 통해 페이지 진입 시 역동적인 경험을 구현함.

- **스크롤 기반 애니메이션**

사용자의 스크롤 동작에 따라 화면 요소가 순차적으로 전환·등장하는 애니메이션을 구현.

단순히 정보를 나열하는 방식이 아니라, 사용자가 스크롤을 내릴 때마다 자연스럽게 시각적 흐름을 경험할 수 있도록 설계하여 페이지 몰입도를 높임.

### 기술적 도전 및 해결 방안

- **Framer-motion을 활용한 SVG Line Drawing**

메인 화면 및 주요 섹션에 SVG 라인 드로잉 애니메이션을 적용하면서, 경로 길이에 따라 속도가 불균형해지거나 화면 크기에 따라 비율이 깨지는 문제가 발생. 또한 다른 애니메이션과 동시에 실행될 때 충돌이 일어나기도 함.

→ stroke-dasharray·stroke-dashoffset을 활용해 경로 길이에 맞춰 균일한 드로잉 속도를 구현하고, viewBox 기반으로 SVG를 스케일링하여 다양한 해상도에서 안정적인 표현을 보장. 추가로 Framer-motion의 시퀀스 제어 기능을 사용해 여러 애니메이션이 순차적으로 실행되도록 설계함.

- **스크롤 기반 애니메이션**

사용자의 스크롤에 따라 요소가 등장·전환되는 애니메이션을 구현하는 과정에서, 스크롤 이벤트 연산이 누적되어 성능 저하가 발생하거나 브라우저·기기에 따라 타이밍이 어긋나는 문제가 있었음. 긴 페이지에서는 애니메이션이 과도하게 적용되어 가독성 저하도 우려됨.

→ Intersection Observer API와 Framer-motion의 whileInView 기능을 결합하여 뷰포트에 들어온 요소만 애니메이션을 실행하도록 최적화. 또한 페이지를 섹션 단위로 나누어 필요한 구간에서만 효과가 발동되도록 하고, 모바일 환경에서는 페이드인/슬라이드 같은 단순화된 효과만 적용해 성능과 가독성을 모두 확보함.

### 주요 성과

- 사용자 경험 개선 :요금제 비교 기능을 단순 표가 아닌 그래프 기반으로 제공하여, 데이터 사용량·음성/문자 제공량·혜택을 한눈에 비교 가능. 또한 스크롤 기반 및 SVG Line Drawing 애니메이션을 도입해, 정적인 페이지를 넘어 몰입감 있고 직관적인 인터랙션 경험 제공.