

제5회 오픈 인프라 개발 경진대회

개발 계획서 [자유주제]

□ 참가팀 개요

팀명	GPT In My Local	팀장명	이성주
이메일	juscomputer@khu.ac.kr	핸드폰번호	010-2877-4284
팀 소속	경희대학교		
팀원 소개	김태오, 김형준, 이성주, 황성연		

□ 과제

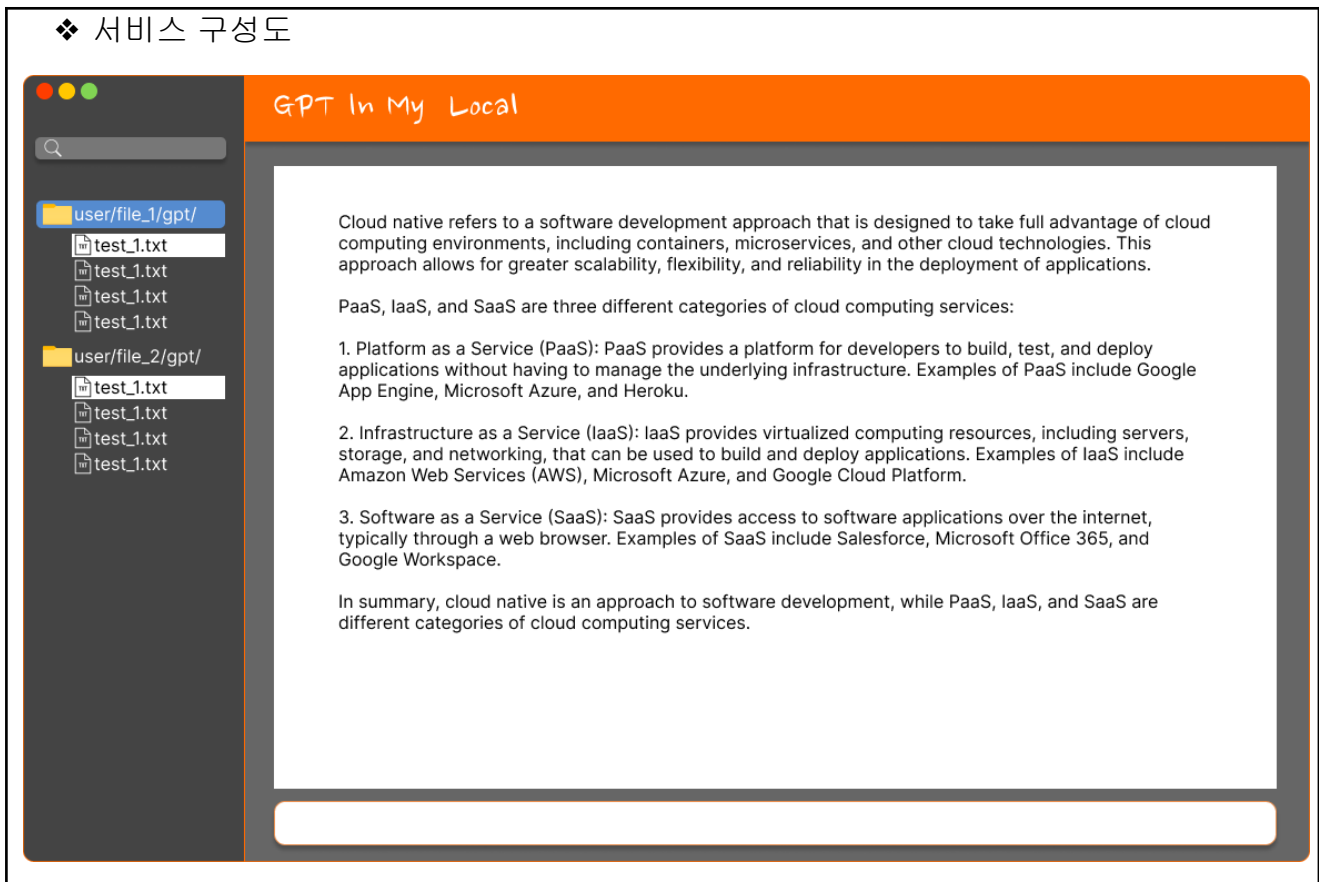
과제명	GPT와 OS를 결합한 텍스트파일 스토리지 멀티 플랫폼 서비스
과제 요약	<p>이 과제의 목적은 GPT-3 OpenAI API를 사용하여 자연어 질의 응답 시스템을 구현하고, 클라우드 기술을 활용하여 안정적이고 확장 가능한 서비스를 제공하는 것이다. Flutter와 Spring Framework를 사용하여 개발되며, K8s를 이용하여 자동 스케일링이 가능한 클라우드 환경에서 운영된다. 또한, AWS 서비스를 활용하여 자동화된 배포, 로깅 및 모니터링, 자동 확장, 보안, 백업 및 복구, 비용 최적화 기능을 구현한다. 마지막으로, 사용자가 요청한 답변 데이터를 로컬에 저장하여 데이터의 접근성을 높이고 재사용성 및 활용성을 높인다.</p>

1. 상세내용

□ 목적 (추진배경 및 필요성)

- 서비스 추진 배경은, 일반적으로 인터넷에서 특정 주제에 대한 검색을 하게 되면 다양한 정보들이 나타나고, 그 중에서 필요한 정보를 찾는 것이 어렵고 번거롭다는 것이다. 이로 인해 사용자들은 시간과 노력을 투자하여 필요한 정보를 찾아내야 한다.
- 따라서, 이 서비스는 사용자들이 텍스트 선택을 통해 특정 주제에 대한 답변을 쉽게 얻을 수 있도록 하는 것이 목적이다. 이를 위해 **GPT**와 같은 자연어 처리 기술을 활용하여, 사용자의 요청에 적합한 답변을 생성한다.
- 현재 자연어 처리 기술은 다양한 분야에서 활용되고 있으며, 많은 연구가 진행되고 있다. 특히, **GPT**와 같은 대형 자연어 모델의 인기 상승으로 이러한 기술을 활용하여 보다 간편하고 정확한 솔루션을 제공하는 것의 중요성이 증가하고 있다.
- 또한, 사용자가 요청한 답변 데이터를 로컬에 **.txt**로 생성후 저장함으로써, 데이터의 접근성을 높이는 동시에 재사용성 및 활용성을 높인다. 이를 통해 사용자는 필요할 때 언제든지 저장된 답변 데이터를 불러와 사용할 수 있으며, 이를 통해 시간과 노력을 절약할 수 있다.

❖ 서비스 구성도





□ 내용

- 자연어 처리: GPT-3 활용, 사용자 답변 생성
- 웹 기술: HTML, CSS, JavaScript
- Flutter: Dart 언어 사용, Android와 IOS 모바일 App 개발
- Spring Framework: RESTful API를 이용한 클라이언트와 서버 간의 통신 처리
- 클라우드 서비스: 컴퓨팅 자원 제공, Kubernetes(K8s) 기반의 컨테이너 오케스트레이션과 스케일링
- 데이터베이스: 사용자 정보 저장 및 관리
- 보안: Spring Security를 사용한 사용자 인증 및 데이터 암호화 처리
- 사용자가 웹 브라우저에서 텍스트를 선택하면, 해당 선택 영역을 서버로 전송한다. 서버에서는 GPT-3를 활용하여 해당 선택 영역에 대한 답변을 생성하고, 이를 클라이언트에게 전송한다. 클라이언트에서는 서버에서 받은 답변을 사용자에게 보여주고, 필요하다면 사용자가 답변을 로컬에 저장할 수 있도록 한다. 이를 위해 HTML, CSS, JavaScript를 활용하여 웹앱을 구현한다. 또한, Flutter와 Dart 언어를 사용하여, Android 및 iOS 모바일 앱을 개발하여 사용자가 언제 어디서든 해당 서비스를 이용할 수 있도록 한다. Spring Framework를 사용하여 RESTful API를 구현하고, 클라이언트와 서버 간의 통신을 처리한다.
- 사용자 정보는 데이터베이스에 저장하여 관리하고, Spring Security를 사용하여 사용자 인증

및 데이터 암호화 처리를 수행한다. 클라우드 서비스를 활용하여 컴퓨팅 자원을 제공하여 서비스의 안정성 및 성능을 보장한다.

- 실현 가능성: 이 서비스는 현재 존재하는 기술과 도구들을 활용하여 구현 가능한 범위 내에 있다. 자연어 처리 분야에서 **GPT-3**는 최근 발표된 강력한 인공지능 기술이며, 웹 및 앱 개발 분야에서는 **HTML, CSS, JavaScript, Flutter, Spring Framework** 등의 기술이 널리 사용되고 있다.

또한, **K8s**를 활용한 오토 스케일링 기술을 사용하여, 서비스에 대한 수요가 증가하거나 감소할 때 서버 리소스를 최적화하고 관리할 수 있어, 서비스 안정성과 확장성을 높일 수 있다. 따라서 충분한 시간과 노력을 투자한다면, 실제 서비스를 구현할 가능성이 높다.

- 시장성: 이 서비스는 자연어 처리 기술과 모바일 앱 기술, 클라우드 기술 등을 종합적으로 활용하여 사용자들이 간편하게 텍스트 답변 서비스를 이용할 수 있는 편리한 환경을 제공한다.

최근에는 스마트폰 및 모바일 기기 사용이 확산되어 모바일 사용이 증가하고 있다. 이에 따라 사용자들은 언제 어디서든지 쉽게 접근 가능한 모바일 앱 기반의 서비스를 선호하게 되었고, 본 서비스에서 제공하는 모바일 앱 기반의 텍스트 답변 서비스는 이러한 사용자들의 요구에 부합한다.

또한, 인터넷 상에서의 정보 검색 및 문의 등에 대한 수요가 높아지면서, 사용자들은 빠르고 정확한 답변을 원하는 경향이 있다. 따라서, 자연어 처리 기술을 활용하여 높은 수준의 정확성과 신속한 답변 제공을 가능하게 한 본 서비스는 해당 수요를 충족시키는데 큰 역할을 할 것으로 기대된다.

또한, 클라우드 기술을 활용하여 사용자들에게 서비스 제공 및 확장성을 보장함으로써, 대규모 이용자들에게도 높은 성능과 안정성을 유지하면서 서비스 제공이 가능하다.

따라서, 이 서비스는 현재의 트렌드와 사용자 요구를 고려한 실용적이고 선진적인 서비스로, 높은 시장성을 가지고 있을 것으로 기대된다.

- 사업성: 이 서비스는 대상 사용자가 많은 인터넷 서비스 분야에서 비즈니스 모델을 구성할 수 있다. 대표적으로 인터넷 검색, 쇼핑몰, 교육, 문서 작성 도구, 비즈니스 채팅 애플리케이션 등에서 활용이 가능하다.

사용자가 검색어나 질문을 입력하고 그에 대한 답변을 얻는 과정은 많은 서비스에서 필수적인 요소이다. 이 서비스는 사용자가 웹 페이지에서 직접 선택한 내용에 대한 답변을 제공하는 기술을 바탕으로, 검색 서비스나 문서 작성 도구에서 활용될 수 있다. 또한, 교육 서비스에서는 수업자가 학생들의 질문에 실시간으로 대응할 수 있도록 이 기술을 활용할 수 있다.

또한, 이 서비스는 스마트폰 애플리케이션으로도 개발될 수 있다. 스마트폰에서는 텍스트를

선택하는 것이 더욱 직관적이며, 스마트폰에서 바로 답변을 얻을 수 있다는 장점이 있다. 이를 바탕으로, 스마트폰 검색 서비스나 교육 서비스에서도 적용이 가능하다.

따라서, 이 서비스는 다양한 분야에서 비즈니스 모델로 활용될 수 있으며, 그에 따른 수익 창출 가능성이 높다.

□ 기대효과 및 활용성

- 자연어 처리 기술을 활용하여 빠르고 정확한 답변 제공: **GPT**를 활용하여 사용자가 원하는 정보를 정확하고 자연스러운 답변으로 제공하여 사용자 경험을 개선한다.
- 다양한 플랫폼에서 이용 가능한 서비스 제공: **Flutter**와 **Spring**을 활용하여 웹 및 모바일 애플리케이션을 개발하여 다양한 플랫폼에서 사용 가능한 서비스를 제공한다.
- 효율적인 서비스 운영으로 비용 절감 및 사업성 향상: 클라우드 서비스를 활용하여 컴퓨팅 자원을 효율적으로 활용하여 서비스 운영 비용을 절감하고, 사업성을 높인다.
- 로컬 데이터 스토리지를 활용한 애플리케이션 성능 개선 및 보안성 향상: 과거의 데이터를 로컬에 저장하여 애플리케이션 성능을 개선하고 보안성을 향상시키며, 서버 부하도 줄여 응답속도를 개선한다.
- 이 서비스는 로컬에서 텍스트 파일로 관리하는 형태의 데이터 스토리지를 제공하며, **GPT**를 사용하여 질의응답이 진행된다. 이는 기존의 스토리지와는 다른 데이터 접근 방식을 제공하며, 사용자들이 새로운 형태의 데이터 스토리지 시장을 개척할 수 있다. 또한, **PC**나 **모바일** 등 다양한 플랫폼에서 동작할 수 있어 사용자의 이동성과 유연성을 높여주며, 이미지나 동영상 파일로도 적용 가능하여 사용자 경험을 향상시킬 수 있다.
- 이 서비스는 정보 제공, 컨설팅 및 상담, 교육 분야에서 활용될 수 있다. 정보 제공 서비스에서는 사용자들이 원하는 정보를 빠르게 얻을 수 있으며, 답변이 정확하고 빠르기 때문에 유용하다. 컨설팅 및 상담 서비스에서는 사용자들의 질문에 대해 실시간으로 답변해주는 기능을 활용하여, 빠르고 효과적인 상담 서비스를 제공할 수 있다. 또한, 교육 분야에서는 사용자들이 원하는 주제에 대한 질문에 대해 적절한 답변을 제공해주는 기능을 활용하여, 교육 분야에서도 활용될 수 있다.

□ 기타

- 데이터 시각화: 사용자의 검색 결과를 시각적으로 표현하여 보여주는 데이터 시각화 기능을 제공할 수 있다. 이를 통해 사용자들은 보다 직관적으로 검색 결과를 파악할 수 있고, 사용성이 향상될 수 있다.
- 기계 학습: 검색 결과를 분석하고, 이를 바탕으로 기계 학습을 시행하여 더욱 정확하고 개인화된 검색 결과를 제공할 수 있다. 이를 통해 사용자는 더욱 만족스러운 검색 결과를

얻을 수 있으며, 서비스의 경쟁력도 높일 수 있다.

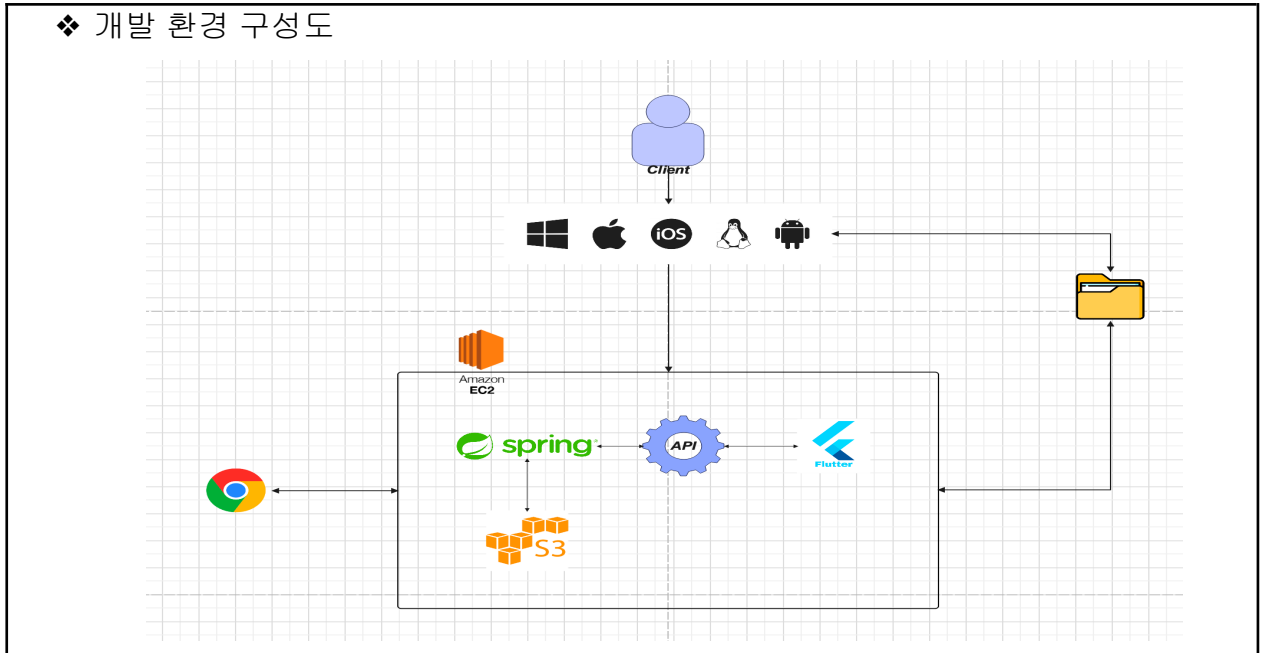
- 자동 번역 기능: 검색 결과가 다른 언어로 제공될 경우, 자동 번역 기능을 제공하여 사용자가 다른 언어로 된 검색 결과를 이해할 수 있도록 할 수 있다.
- 개인화 서비스: 사용자가 과거에 검색한 내역을 바탕으로 개인화된 검색 서비스를 제공할 수 있다. 이를 통해 사용자는 더욱 편리하게 원하는 정보를 찾을 수 있으며, 서비스 운영자는 더욱 정확한 정보를 제공하여 서비스의 만족도를 높일 수 있다.

2. 활용기술

□ 개발 환경 구성

- Front-end : Flutter(Dart)
- Back-end : Spring(Java)
- Database : PostgreSQL or AWS S3
- DevOps : Docker, Kubernetes(K8s), AWS EC2
- 서비스의 기능들은 회원 관리, **GPT API** 등으로 구성되며, 이들은 서비스 단위로 분리하여 각각의 서비스를 **Container**화 한다. 이렇게 분리된 **Container**들은 하나의 **Pod**로 모아 **AWS EC2**에서 실행된다. 이러한 클라우드 네이티브 아키텍처는 클라우드 기반의 환경에서 효율적인 서비스 운영을 지원한다. 또한, 이 아키텍처를 사용함으로써 서비스의 확장성과 유연성이 높아지며, 서비스의 가용성과 안정성을 보장할 수 있다.
- 이 서비스의 클라우드 네이티브 아키텍처는 컨테이너화된 마이크로서비스 아키텍처를 기반으로 구성되어 있다. 각각의 마이크로서비스는 회원관리, **GPT API** 등과 같은 특정 기능을 수행하며, 이러한 마이크로서비스들은 서비스 단위로 분리되어 개발되고 배포된다.
- 각 마이크로서비스는 **Docker**를 사용하여 컨테이너화되며, 이를 **Kubernetes(K8s)**와 같은 오케스트레이션 툴로 관리한다. 이를 통해 높은 가용성, 확장성 및 격리성을 보장하며, 마이크로서비스의 각 버전을 쉽게 롤아웃/롤백 할 수 있다.
- **AWS EC2**와 같은 클라우드 인프라에서 이러한 컨테이너화된 마이크로서비스들은 논리적으로 하나의 물리적인 서버(**Pod**)에서 실행되도록 구성된다. 이를 통해 서비스 인스턴스 간의 네트워크 지연을 줄이고, 서비스 관리를 쉽게 할 수 있다.
- 또한, 이 서비스는 **AWS S3**와 같은 클라우드 데이터 스토리지를 사용하여 데이터를 저장한다. 이를 통해 데이터의 안전성과 확장성을 보장하며, 필요에 따라 데이터의 백업 및 복구가 가능하다.

❖ 개발 환경 구성도



□ 핵심 기술력

- 자연어 처리 기술: GPT-3를 이용하여 사용자의 질의에 대한 적절한 답변을 생성한다.
- RESTful API 기술: Spring Framework에서 RESTful API를 이용하여 클라이언트와 서버 간의 데이터 전송 및 처리를 구현한다.
- Docker 및 Kubernetes 기술: Docker를 이용하여 각각의 서비스를 컨테이너화하고, Kubernetes를 이용하여 컨테이너를 효율적으로 관리하며, 서비스의 가용성과 확장성을 높인다.
- 데이터베이스 기술: PostgreSQL 또는 AWS S3를 이용하여 사용자 정보를 저장하고 관리한다.
- 보안 기술: Spring Security를 이용하여 사용자 인증 및 데이터 암호화 처리를 구현하여 사용자 데이터의 안전성과 보안성을 보장한다.
- 클라우드 네이티브 기술: 클라우드 네이티브 기술을 활용하여 서비스의 가용성과 확장성을 높인다.
- 오토 스케일링: K8s에서 HPA(Horizontal Pod Autoscaling)를 통해 가변적인 수요에 대응하고 서비스의 확장성을 고려하여 Scale 정책을 설정하여 오토 스케일링을 수행한다.

□ 클라우드 기술 적용 계획(서비스 구현 및 운영 계획)

- 컨테이너화: Front-end와 Back-end에서 구현한 서비스들을 각각의 Docker Container로 분할하여 관리한다.
- 오케스트레이션: 서비스 동작에 필요한 Container들을 Pod에 모은 후 K8s에서 Pod를 관리하게 한다. K8s의 Auto Scaling 기술을 활용하여 트래픽 증감에 따른 필요 Pod 수의

증감에 대응하여 Pod들을 관리(증설, 폐기)한다.

- 자동화된 배포: **GitHub**과 **AWS CodePipeline**을 연동하여 **CI/CD** 파이프라인을 구축하고, **AWS CodeDeploy**를 이용하여 자동화된 배포를 구현한다. 이를 통해 소스 코드 변경 사항이 발생할 때마다, 배포를 자동화하여 빠른 배포를 구현한다.
- 로깅 및 모니터링: **AWS CloudWatch**와 **AWS CloudTrail**을 이용하여 서비스의 로그를 수집하고 분석하는 로깅 및 모니터링 시스템을 구축한다. **CloudWatch**는 시스템 자원 모니터링을 지원하며, **CloudTrail**은 **AWS** 리소스 **API** 호출 및 이벤트 로그를 수집하여 보안 이슈에 대한 정보를 제공한다.
- 자동 확장: **AWS Auto Scaling**을 이용하여, 서비스에 대한 트래픽 증가 시 자동으로 인프라를 확장하는 자동 확장 기능을 제공한다. 이를 통해 서비스의 가용성을 높인다.
- 보안: **AWS Identity and Access Management(IAM)**을 이용하여 서비스에 대한 접근 제어를 구현한다. 또한 **AWS Key Management Service(KMS)**를 이용하여 데이터 암호화를 적용한다.
- 백업 및 복구: **AWS S3**를 이용하여 주기적인 데이터 백업을 수행하고, **AWS Disaster Recovery**를 이용하여 재해 복구 계획을 수립한다. 이를 통해 서비스의 가용성을 높인다.
- 비용 최적화: **AWS Spot Instance**를 이용하여, 저렴한 비용으로 인프라를 운영한다. 또한 **AWS Cost Explorer**를 이용하여 서비스 운영 비용을 최적화한다.

- (1) 클라우드 서비스 제공 업체 선택: 서비스의 요구사항과 예산을 고려하여 적절한 클라우드 서비스 제공 업체를 선택한다.
- (2) 서버 구성: 선택한 클라우드 서비스 제공 업체에서 서버 인스턴스를 생성하고, 필요한 소프트웨어와 라이브러리를 설치한다. 또한, 데이터베이스와 같은 기술을 사용하여 데이터를 저장한다.
- (3) 서버 구성 관리: 서버 구성 및 관리를 자동화하기 위해 구성 관리 도구를 사용한다. 이를 통해 서버 구성 및 업데이트를 효율적으로 관리할 수 있다.
- (4) 모니터링 및 로그 관리: 서버 및 애플리케이션의 상태를 모니터링하고, 문제가 발생한 경우 로그를 수집하여 분석한다.
- (5) 보안 강화: 클라우드 서비스 제공 업체에서 제공하는 보안 기능을 활용하여 서버와 데이터를 안전하게 보호한다.
- (6) 백업 및 복원: 데이터의 안전성을 보장하기 위해 정기적으로 백업을 수행하고, 데이터 복원 절차를 수립한다.
- (7) 서비스 개발 및 운영: 클라우드 서비스 제공 업체에서 제공하는 개발 도구와 지원 서비스를 활용하여 서비스를 개발하고 운영한다.
- (8) API 개발: 클라이언트에서 서버로 선택된 텍스트 데이터를 전송하기 위한 **API**를 개발한다.
- (9) GPT 모델 연동: 서버에서 전송받은 텍스트 데이터를 **GPT** 모델에 전달하여 처리하고, 생성된 답변 데이터를 클라이언트에게 반환한다.
- (10) 테스트 및 디버깅: 서비스의 안전성과 기능을 확인하기 위해 테스트와 디버깅 작업을

수행한다.

- (11) 서비스 배포: 서비스가 안정적으로 작동하는 것을 확인하고, 클라우드 서비스 제공 업체에서 제공하는 배포 기능을 사용하여 서비스를 배포한다.
- (12) 유지보수: 서비스 운영 중 발생할 수 있는 문제에 대비하여 유지보수 계획을 수립하고, 필요한 조치를 취하여 서비스를 안정적으로 운영한다.