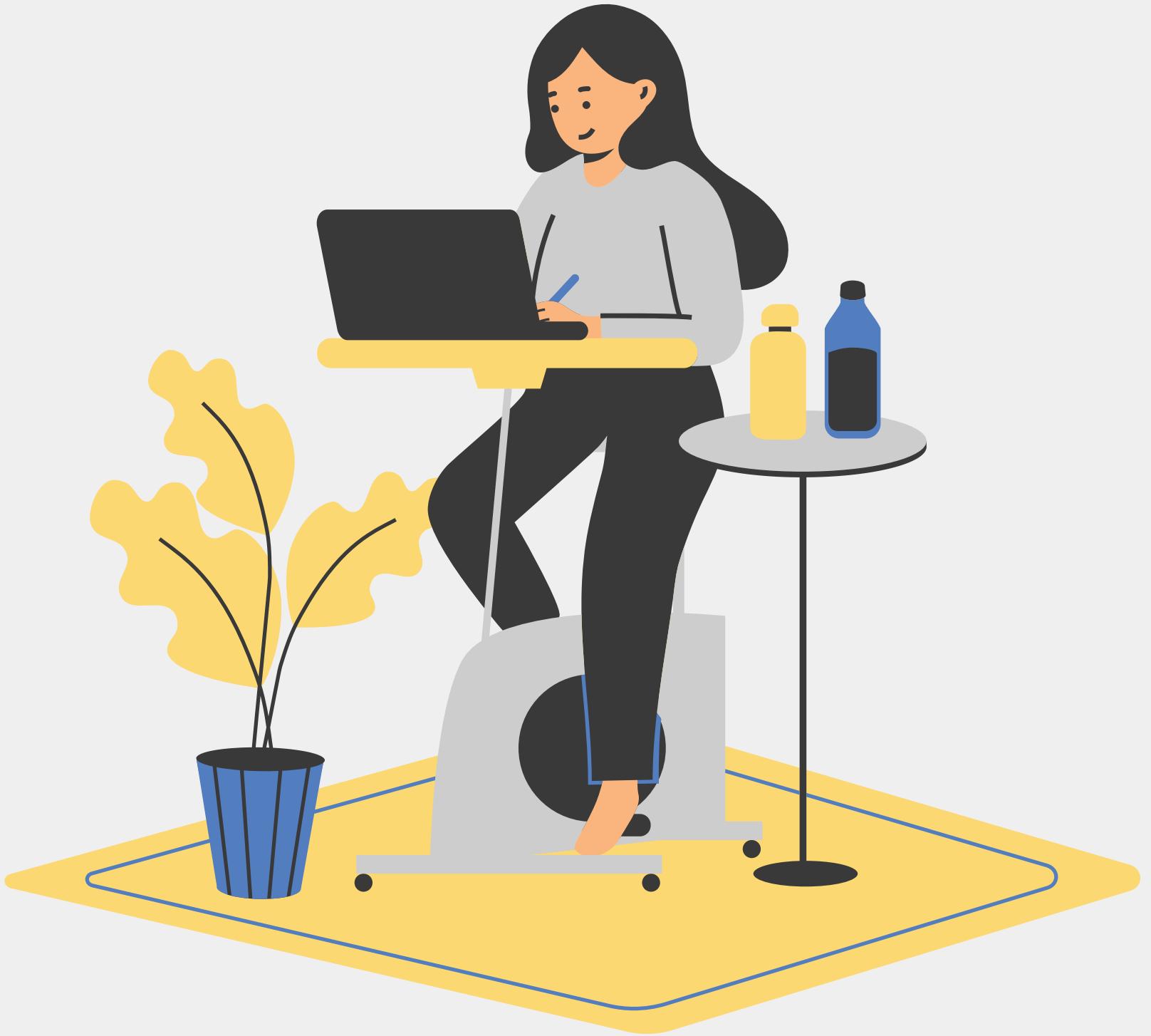


# ALOG.

Auto Blog

소프트웨어학과 정용훈  
소프트웨어학과 남현원  
소프트웨어학과 김평주  
소프트웨어학과 한도연



# ALOG

## 팀 구성 정보 및 역할 분담

김평주

소프트웨어학과 202020708



### 프론트

- React, Flutter, JavaScript, TailwindCSS 프로젝트 경험 있음
- React 사용할 예정

한도연

소프트웨어학과 202020737

정용훈

소프트웨어학과 202020354



### 백엔드

- FastAPI 프로젝트 경험 있음
- DB 로 MongoDB, MySQL, PostgreSQL 사용 경험 있음
- CloudFront, EC2 등의 배포 경험 있음
- FastAPI 사용할 예정

남현원

소프트웨어학과 202020820

# 01 서비스 개요

## 문제점 정의

**It takes just under 4 hours to write a blog post on average**

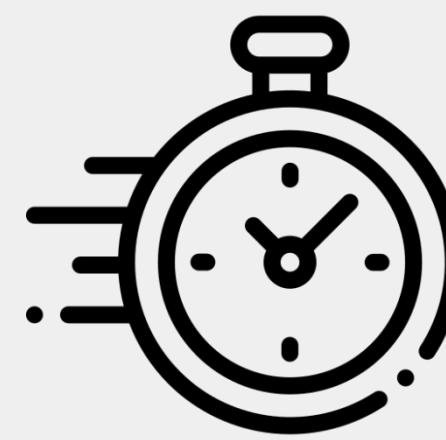
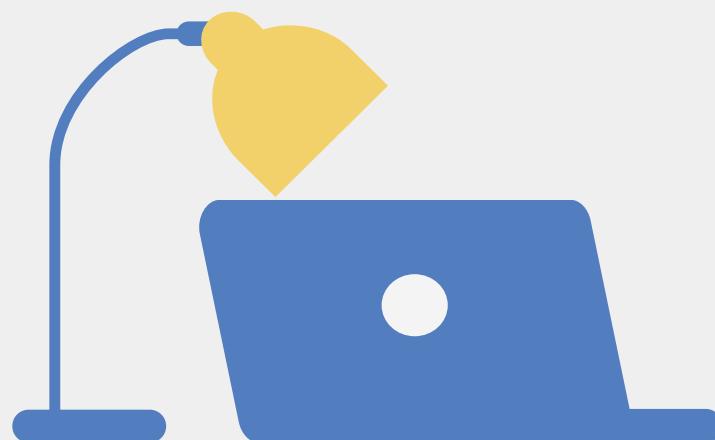
That's less time than last year, but still 60% more than 10 years ago.

Year	# of hours
2014	2:24
2015	2:35
2016	3:16
2017	3:20
2018	3:28
2019	3:57
2020	3:55
2021	4:01
2022	4:10
2023	3:51

n=1051 orbitmedia.com

**블로그에 글 하나를 작성**

➤ 평균적으로 3시간에서 4시간의 시간이 소요



## ALOG

Alog

로그인

**새 포스트 작성**

- 1 포스팅 내용 업로드
- 2 세부 사항 설정
- 3 정리 내용 검토
- 4 게시

다음 세부 사항은 Alog가 포스팅 내용 분석에 따른 결과예요.  
정확한 내용 작성을 위해 세부 사항을 조정해주세요.

제목  
생성형 AI를 활용한 블로그 포스팅 정리를 소개합니다

주제  
지식·학습  일상  여행  
엔터테인먼트·예술  취미·노하우

다음

**PDF 형식의 파일을 입력받아 AI가 내용 요약 및 생성  
내용에 따라 사용자가 원하는 위치에 사진 게시**

# 01 서비스 개요 → 피그마

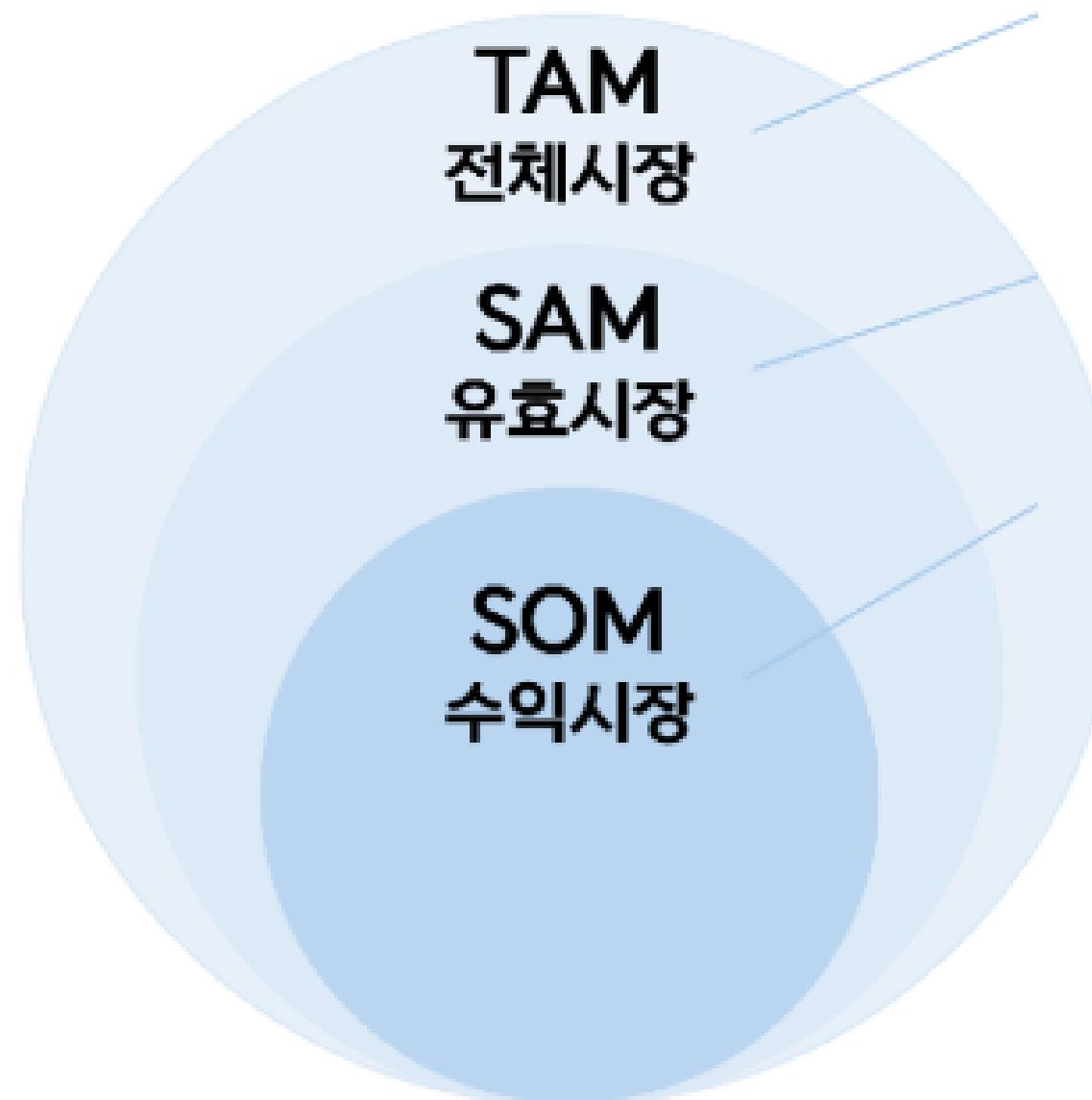
The image displays three sequential screenshots of the Alog blog post creation process:

- Post 1: Posting Content Upload** (포스팅 내용 업로드): This step shows a large dashed blue box for uploading files, with a prominent "PDF를 업로드 해 주세요" button and a placeholder for file selection. Below the box is a cartoon illustration of a person working at a desk. Step 1 is highlighted: "포스팅 내용 업로드".
- Post 2: Sub-task Setting** (세부 사항 설정): This step shows a list of sub-tasks: "포스팅 내용 업로드", "세부 사항 설정", "정리 내용 검토", and "게시". The second task is highlighted: "세부 사항 설정". It includes a note about Alog's AI content analysis and a "다음" (Next) button.
- Post 3: Summary Review** (정리 내용 검토): This step shows the final review stage. It highlights the "게시" (Post) button and includes a summary of the generated content: "Alog가 포스팅 내용 3개를 생성했어요. '내용 다시 생성' 버튼을 눌러 더 다양한 내용을 생성할 수 있어요.". On the right, there is a sidebar titled "Alog의 주요 기능" (Main Features) listing five features: 1. 자동 글 생성, 2. 세부 사항 설정, 3. 정리 내용 검토, 4. 맞춤형 콘텐츠 추천, and 5. 문법 및 스타일 체크.

블로그 작성의 소요되는 시간을 위의 3단계를 통해 **시간 단축**

<https://www.figma.com/proto/6aNSHhgVrJXn3FQKs0ksOn/Alog-Prototype?node-id=54795-2083&t=zVM4t1OsxC1cMN1g-1&scaling=min-zoom&page-id=54795%3A2082>

# 01 서비스 개요 → 주 타겟 사용자



- 전 세계 80억 인구 중 블로그를 가지고 있는 사람 수:  
약 6억명
  - 전 세계 웹사이트의 3분의 1을 차지
- 
- 블로그 글 작성에 1시간 이상 투자하는 사람 수:  
전체 블로거의 72% = 약 4.32억명
- 
- 블로그 글 작성 시 AI를 활용할 용의가 있는 사람 수:  
전체 블로거의 65% = 약 3.9억명



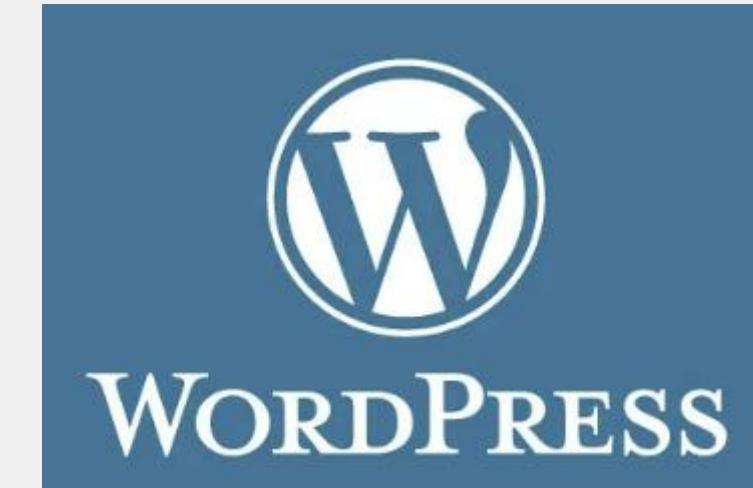
1. 'ALOG'는 블로그의 다양한 목적에 맞춰  
사용 가능

☞ 블로그를 사용하는 모든 사용자를 타깃

2. 블로그 글 작성 시 AI를 활용할 용의가  
있는 사람

3. 블로그 작성 및 게시의 큰 단점인  
**'장시간 소요'** 때문에  
블로그 사용을 망설이던 사람

# 02 기존사례 및 차별화



## 네이버 티스토리

- 디자인을 지정되어 있는 범위 안에서 직접 꾸미고 글을 작성

## 워드프레스

- 블로그를 만들고 관리할 수 있도록 돋는 '오픈소스 프로그램'
- HTML, CSS 등의 웹 소스코드를 직접 수정 가능

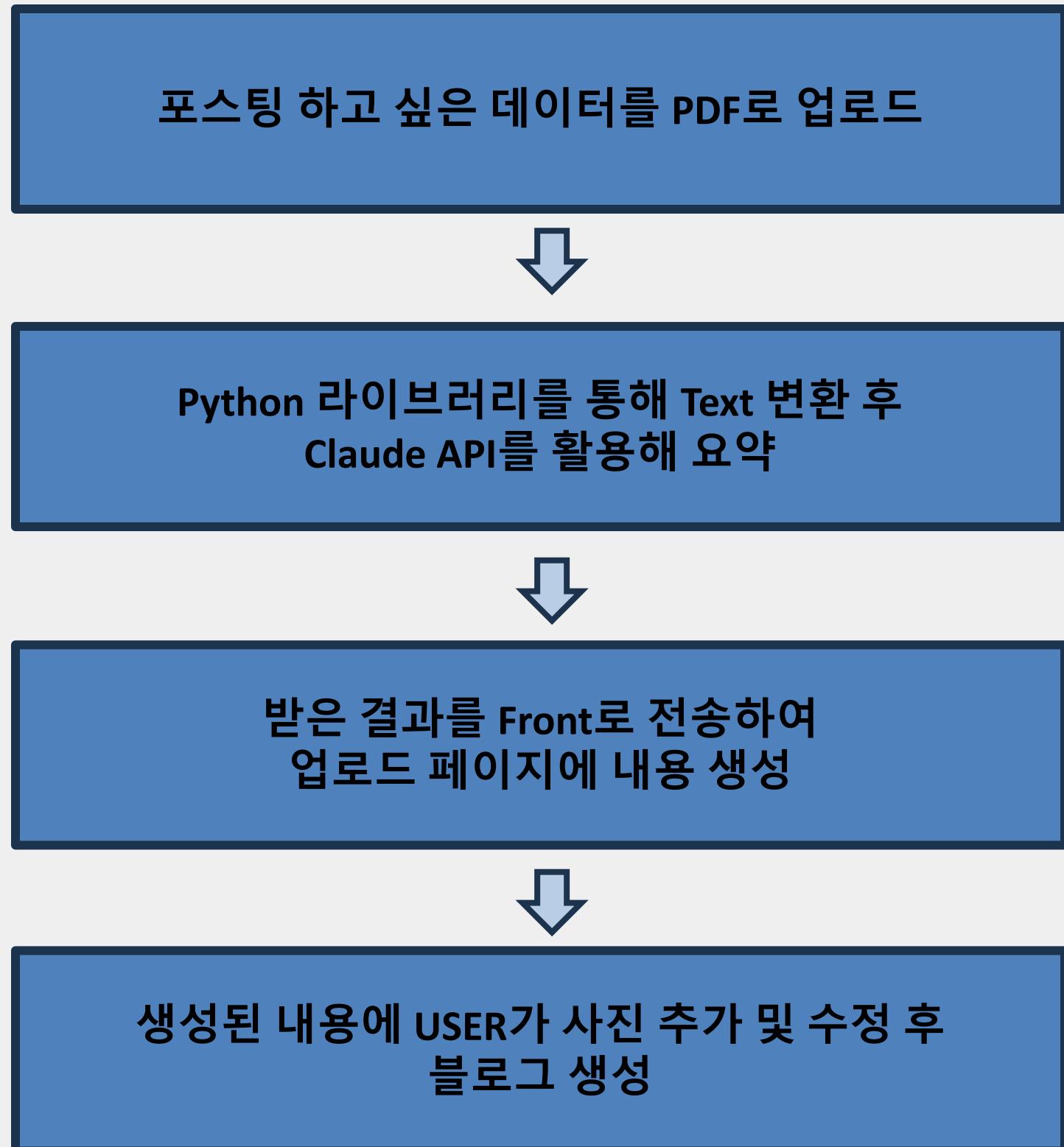
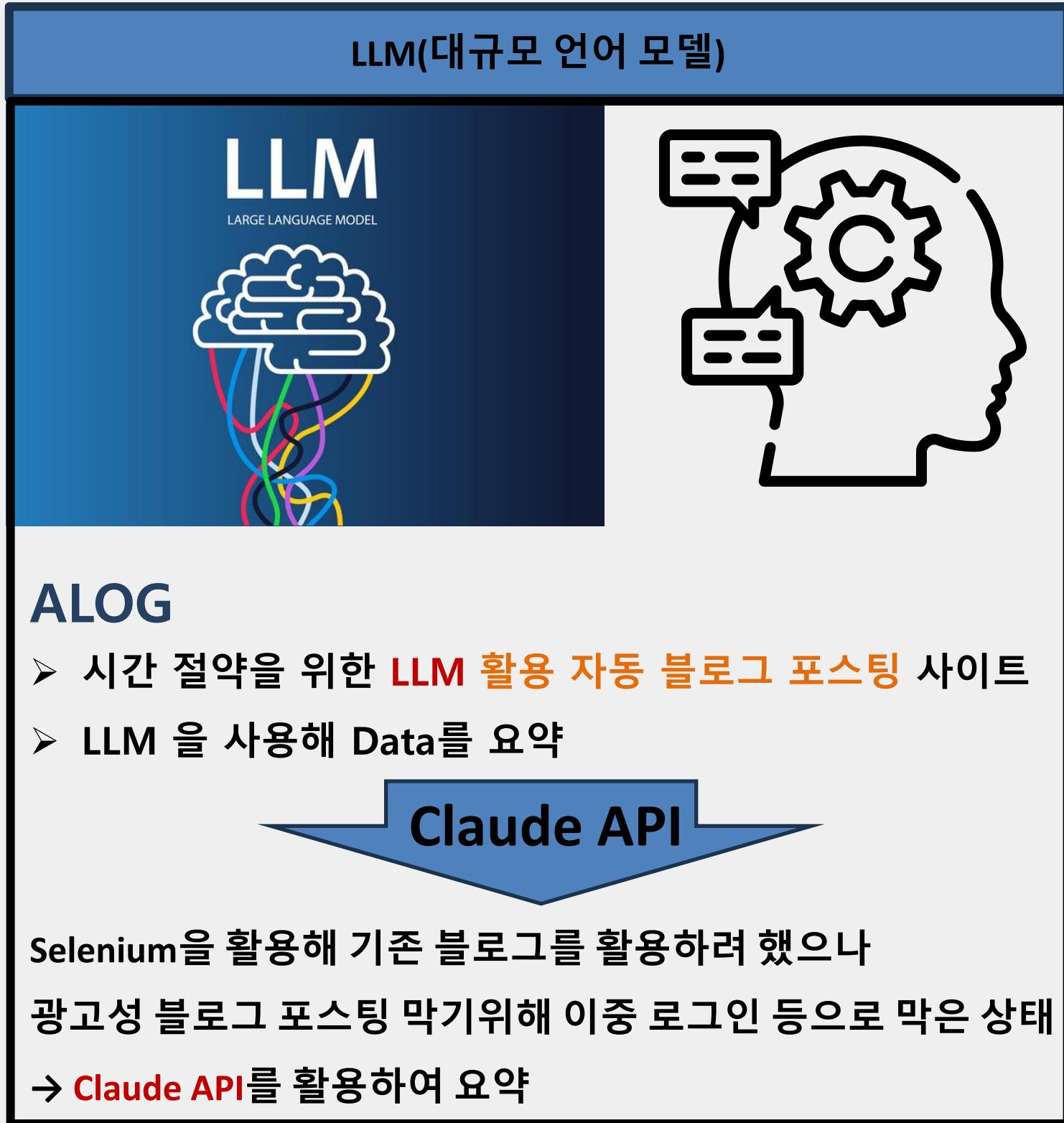
## 가제트

- AI를 활용하여 블로그 작성
- 원하는 주제와 키워드를 작성하면 ai가 자동으로 새로운 글의 내용을 작성 (수익형 블로그에 유리)

### 차별화

1. 진입장벽이 네이버와 같이 낮으면서 모든 내용을 직접 작성하는 것이 아닌, AI를 활용하여 작성
2. 웹 관련 프로그래밍 지식을 몰라도 테마에 맞게自動으로 만들어 주기 때문에 진입장벽이 훨씬 낮음
3. AI를 활용하여 작성 시 새로운 내용을 생성하는 것이 아닌 요약하여 사용자의 말투로 변환하여 작성

# 03 생성형 AI를 활용한 서비스 실현 방안



# 03 생성형 AI를 활용 - 예시

The screenshot shows a terminal window with two tabs: 'apiTest.py' and 'ClaudeTest'. The 'apiTest.py' tab contains Python code for reading a PDF and extracting text. The 'ClaudeTest' tab shows a response from the Claude API, which is a detailed summary of SQL concepts.

```
apiTest.py
1 def summarize_pdf(path: str) -> str:
2     reader = PdfReader(path)
3     text = "\n".join([page.extract_text() for page in reader.pages])
4     return text
5
6 PdfText = "아래 내용을 블로그에 설명하는 방식으로 요약해줘. 블로그에 포스팅할거야. 한글로 설명하고 흥미를 끌 수 있게 글을 작성해봐. 그리고 첫 줄은 전체 내용에 대한 제목을 작성해줘.\n"
7
8 Text = summarize_pdf("./DB_08_More_SQL_SUMMARY.pdf")
9 PdfText += Text
10
11
문제 출력 터미널 포트 GITLENS 직렬 모니터 디버그 콘솔
+
zsh
Code
(claude-env) (base) jeong-yonghun@jeong-yonghun-ui-MacBookAir ClaudeTest % python3 apiTest.py
Message(id='msg_01STHzmPTs2GuVqT8HeawJh', content=[TextBlock(text='제목: SQL에서 Three-Valued Logic, Nested Queries, JOIN, Aggregate Function, ASSERTION, TRIGGER, View 등 알아보기\n\n안녕하세요, 여러분! 오늘은 SQL에서 사용되는 다양한 개념들에 대해 알아보려고 해요. 이 글을 통해 SQL에 대한 이해도를 높일 수 있을 거예요.\n\n먼저, Three-Valued Logic에 대해 살펴볼게요. SQL에서는 TRUE, FALSE, UNKNOWN 세 가지 값을 사용해요. AND, OR, NOT 연산자를 사용할 때 이 값들이 어떻게 조합되는지 표로 정리해 두었으니 참고하시면 좋을 거예요.\n\n다음은 Nested Queries예요. WHERE 절 내에 (select-from-where) 블록을 사용할 수 있어요. IN, ANY, ALL, EXISTS 등의 연산자와 함께 사용되죠. 이를 통해 더 복잡한 조건을 표현할 수 있어요.\n\nJOIN은 두 테이블을 연결할 때 사용해요. INNER JOIN과 OUTER JOIN(LEFT, RIGHT, FULL)이 있어요. 조건을 만족하는 투플들로 결과 테이블을 만들 수 있죠.\n\nAggregate Function도 알아두면 유용해요. COUNT, SUM, MAX, MIN, AVG 등의 내장 함수를 사용할 수 있어요. GROUP BY와 함께 사용하면 그룹별로 집계를 계산할 수 있죠.\n\nASSERTION은 추가적인 제약 조건을 생성할 때 사용해요. TRIGGER는 특정 이벤트 발생 시 자동으로 실행되는 저장 프로시저예요. \n\nView는 다른 테이블에서 파생된 가상 테이블이에요. 쿼리에는 제한이 없지만, 업데이트는 제한될 수 있어요.\n\n마지막으로 Schema evolution commands에 대해 알아볼게요. DB 스키마를 수정할 때 사용되는 명령어들이에요. DROP, ALTER 등을 사용해 스키마를 변경할 수 있죠.\n\n이렇게 SQL에서 사용되는 다양한 개념들에 대해 알아보았어요. 이 글을 통해 SQL에 대한 이해도가 높아졌길 바라요. 다음에는 더 흥미로운 주제로 찾아뵐게요!', type='text')], model='claude-3-opus-20240229', role='assistant', stop_reason='end_turn', stop_sequence=None, type='message', usage=Usage(input_tokens=1370, output_tokens=773))
[TextBlock(text='제목: SQL에서 Three-Valued Logic, Nested Queries, JOIN, Aggregate Function, ASSERTION, TRIGGER, View 등 알아보기\n\n안녕하세요, 여러분! 오늘은 SQL에서 사용되는 다양한 개념들에 대해 알아보려고 해요. 이 글을 통해 SQL에 대한 이해도를 높일 수 있을 거예요.\n\n먼저, Three-Valued Logic에 대해 살펴볼게요. SQL에서는 TRUE, FALSE, UNKNOWN 세 가지 값을 사용해요. AND, OR, NOT 연산자를 사용할 때 이 값들이 어떻게 조합되는지 표로 정리해 두었으니 참고하시면 좋을 거예요.\n\n다음은 Nested Queries예요. WHERE 절 내에 (select-from-where) 블록을 사용할 수 있어요. IN, ANY, ALL, EXISTS 등의 연산자와 함께 사용되죠. 이를 통해 더 복잡한 조건을 표현할 수 있어요.\n\nJOIN은 두 테이블을 연결할 때 사용해요. INNER JOIN과 OUTER JOIN(LEFT, RIGHT, FULL)이 있어요. 조건을 만족하는 투플들로 결과 테이블을 만들 수 있죠.\n\nAggregate Function도 알아두면 유용해요. COUNT, SUM, MAX, MIN, AVG 등의 내장 함수를 사용할 수 있어요. GROUP BY와 함께 사용하면 그룹별로 집계를 계산할 수 있죠.\n\nASSERTION은 추가적인 제약 조건을 생성할 때 사용해요. TRIGGER는 특정 이벤트 발생 시 자동으로 실행되는 저장 프로시저예요. \n\nView는 다른 테이블에서 파생된 가상 테이블이에요. 쿼리에는 제한이 없지만, 업데이트는 제한될 수 있어요.\n\n마지막으로 Schema evolution commands에 대해 알아볼게요. DB 스키마를 수정할 때 사용되는 명령어들이에요. DROP, ALTER 등을 사용해 스키마를 변경할 수 있죠.\n\n이렇게 SQL에서 사용되는 다양한 개념들에 대해 알아보았어요. 이 글을 통해 SQL에 대한 이해도가 높아졌길 바라요. 다음에는 더 흥미로운 주제로 찾아뵐게요!', type='text')]

제목: SQL에서 Three-Valued Logic, Nested Queries, JOIN, Aggregate Function, ASSERTION, TRIGGER, View 등 알아보기
안녕하세요, 여러분! 오늘은 SQL에서 사용되는 다양한 개념들에 대해 알아보려고 해요. 이 글을 통해 SQL에 대한 이해도를 높일 수 있을 거예요.
먼저, Three-Valued Logic에 대해 살펴볼게요. SQL에서는 TRUE, FALSE, UNKNOWN 세 가지 값을 사용해요. AND, OR, NOT 연산자를 사용할 때 이 값들이 어떻게 조합되는지 표로 정리해 두었으니 참고하시면 좋을 거예요.
다음은 Nested Queries예요. WHERE 절 내에 (select-from-where) 블록을 사용할 수 있어요. IN, ANY, ALL, EXISTS 등의 연산자와 함께 사용되죠. 이를 통해 더 복잡한 조건을 표현할 수 있어요.
JOIN은 두 테이블을 연결할 때 사용해요. INNER JOIN과 OUTER JOIN(LEFT, RIGHT, FULL)이 있어요. 조건을 만족하는 투플들로 결과 테이블을 만들 수 있죠.
Aggregate Function도 알아두면 유용해요. COUNT, SUM, MAX, MIN, AVG 등의 내장 함수를 사용할 수 있어요. GROUP BY와 함께 사용하면 그룹별로 집계를 계산할 수 있죠.
ASSERTION은 추가적인 제약 조건을 생성할 때 사용해요. TRIGGER는 특정 이벤트 발생 시 자동으로 실행되는 저장 프로시저예요.
View는 다른 테이블에서 파생된 가상 테이블이에요. 쿼리에는 제한이 없지만, 업데이트는 제한될 수 있어요.
마지막으로 Schema evolution commands에 대해 알아볼게요. DB 스키마를 수정할 때 사용되는 명령어들이에요. DROP, ALTER 등을 사용해 스키마를 변경할 수 있죠.
이렇게 SQL에서 사용되는 다양한 개념들에 대해 알아보았어요. 이 글을 통해 SQL에 대한 이해도가 높아졌길 바라요. 다음에는 더 흥미로운 주제로 찾아뵐게요!
(claude-env) (base) jeong-yonghun@jeong-yonghun-ui-MacBookAir ClaudeTest % 
```

줄 9, 열 110 공백: 4 UTF-8 LF { Python 3.12.3 ('claude-env': venv) ⚡ 🔍

- USER가 PDF를 제공하면 서버로 보내져 Claude API를 통해 내용 요약
- (추가적인 prompt tuning 예정)
- 넘어온 내용을 확인한 user는 내용을 수정 및 사진 추가를 하거나 그대로 제출

## Claude API를 활용 예시

- 내용 요약
- 제목 작성
- 사용자 말투 활용

# 일정 계획

<b>24.6.2 – 24.6.9</b>	<b>24.6.24 – 24.6.30</b>	<b>24.7.1 – 24.7.7</b>	<b>24.7.8 – 24.7.14</b>	<b>24.7.15 – 24.7.17</b>
<p>공통</p> <p>➤ 개발 환경 구축, API 명세서 정의 및 요구사항 정의, Git Commit Rule 정의, 테스트 케이스 생성</p>	<p>프론트</p> <p>➤ 로그인 및 회원가입 페이지 개발</p> <p>백엔드</p> <p>➤ DB구축 및 로그인 회원가입 API 개발</p>	<p>프론트</p> <p>➤ 메인 페이지 및 유저 관련 페이지 개발</p> <p>백엔드</p> <p>➤ 메인 페이지 및 유저 관리 페이지 요청 API 개발</p>	<p>프론트</p> <p>➤ 블로그 게시 및 작성 페이지 개발</p> <p>백엔드</p> <p>➤ 블로그 게시 및 생성 요청 API 개발</p>	<p>프론트</p> <p>➤ 최종 페이지 테스트</p> <p>백엔드</p> <p>➤ 최종 API 테스트</p> <p>공통</p> <p>➤ 최종 통합 요청 테스트</p>

# ALOG



감사합니다