

# Flash News

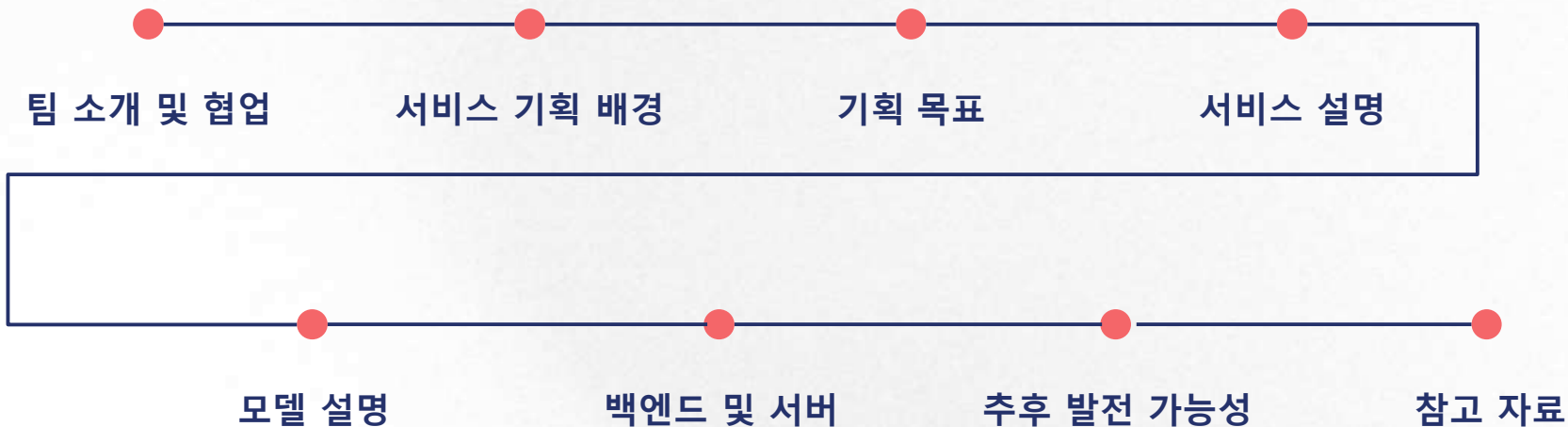
Team. Hae Jyo

---

당신의 시간을 지켜드리겠습니다.



# TABLE OF CONTENTS



팀 소개 및 협업	서비스 기획 배경	기획 목표	서비스 설명
모델 설명	백엔드 및 서버	추후 발전 가능성	참고 자료



01

## 팀 소개 및 협업

# OUR TEAM



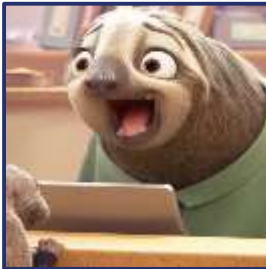
김성진

백엔드 및 서버 총괄  
코드 마에스트로



신건희

백엔드 보좌  
회의 진행자



최성호

프론트이지만 만능 잡부



홍승재

코로나였던  
메인 프론트

# 협업 방법



25%



0%

100%

## 회의록

오전 오후 두번의 회의를  
통한  
의사소통 활성화



50%



0%

100%

## 노션

프로젝트 진행상황, 데이터  
관리, 코드관리 등을 진행



100%



0%

100%

## Git & Git hub

작업의 분할을 통해  
업무능률을 높임

# 회의록 작성

- 오전 9:00 ~ 9:30, 17:20 ~ 17:50 하루 2회 회의시간으로 정함
- 진행할 업무 및 이슈사항 등을 소통함
- 오전 회의 토대로 업무 결과를 오후 회의때 확인
- 회의록들은 노선에 업로드 하여 참고

## 회의록

안 건	데이터베이스에 대한 사용 여부		
일 지	2023년 11월 16일	참석인원	4명
주요안건	데이터베이스를 사용하며 페이지 로딩속도를 높여 이용의 편의성을 도모		
회의 내용			
1. 순찰 발원 이유 2. 웹에 카테고리 버튼시 다른 기사가 크롤링됨 3. 웹 스크롤 상태표시 바 나타내기 4. 네브바 고정시키는 작업 5. 동영상 파일로 크롤링 되는 것 가능하게하기 6. 웹과 모바일에 최적화 방안 모색			
*진급: 데이터베이스를 통해서 일정 시간마다 크롤링 작업을 해서 최신기사들을 저장하고 요약한 것 위주로 웹상에 보여주는 식으로 해서 웹에 부하를 줄이는 방안으로 변경			
회의 결과			
1. 순찰 조퇴 2. 카테고리 보다 해당 카테고리의 기사가 출력되게 만들. 3. 웹 스크롤 상태표시바를 문제 페이지에 나타나게 있는데 그럴 때인 기사 부분으로 이동 시키기만하면 됨. 4. 네브바 고정 아직 안됨. 5. 동영상은 아이프레임으로 끌고오는데 가능하면 가져와서 나오게 하는 것을 목표로 함 6. 기존에는 크롤링을 즉시 진행하여 요약까지 했었는데 데이터베이스를 사용하며 일정시간마다 크롤링을 진행하도록 해서 웹 사이트의 안정성을 높이기로 함			
*데이터 베이스에서 몇 개를 크롤링해서 몇 개를 보여주는지 정해지고 반복하는 크롤링 시간등을 고려해야함. *웹디자인도 결정해야하는 사항. *웹에 사이드바 빈공간을 채울만 것을 생각해봐야 함.			

# 노션

- 데이터 / 회의록 / 진행 상황 정리
- 주요 코드 파일 정리
- 발표자료 및 참고자료 정리



# Git

WORKSPACE

파일 상태

History

Search

브랜치

main

태그

필적

스태시

브랜치

파일 브랜치

날짜순으로 정렬

그래프

설명

날짜

작성자

커밋

날짜	작성자	커밋
23 11 2023 13:58	Shin Geonhee <st>	3aad89c
23 11 2023 12:26	Choi Sungho <sta>	880facd
22 11 2023 16:19	Choi Sungho <sta>	c3b7001
21 11 2023 16:59	Shin Geonhee <st>	a4d7472
21 11 2023 16:37	Choi Sungho <sta>	e301e65
21 11 2023 12:09	Kim sung jin <dal>	1d5d3e6
21 11 2023 11:47	Kim sung jin <dal>	971e20b
20 11 2023 17:45	Kim sung jin <dal>	2f86372
20 11 2023 14:50	Kim sung jin <dal>	408e84e
20 11 2023 14:49	Kim sung jin <dal>	378e8d2
20 11 2023 14:42	Choi Sungho <sta>	f390e42
20 11 2023 14:38	Choi Sungho <sta>	3c3b6df
20 11 2023 14:30	Kim sung jin <dal>	8d7984f
20 11 2023 12:03	Kim sung jin <dal>	6004846
18 11 2023 23:41	Kim sung jin <dal>	2d74093
17 11 2023 17:27	Kim sung jin <dal>	48ee77f
17 11 2023 17:15	Choi Sungho <sta>	1cf5e79
17 11 2023 17:08	Kim sung jin <dal>	4eb0f4
17 11 2023 11:06	Kim sung jin <dal>	900d917
17 11 2023 10:31	Choi Sungho <sta>	fd327a6
16 11 2023 17:27	Kim sung jin <dal>	a8488cf

● Git을 통해 업무 결과를 통합





02

## 서비스 기획 배경

# 여러분들은 뉴스를 얼마나 챙겨보시나요?

■ 그림 1-1 매체별 뉴스 이용률 추이(2011~2021년)

(단위: %)



\*문3, 문11, 문18, 문24, 문40, 문50, 문58, 문66

지난 1주일 동안 00를 통해 뉴스/시사정보를 이용하셨습니다?

# 미디어 노출

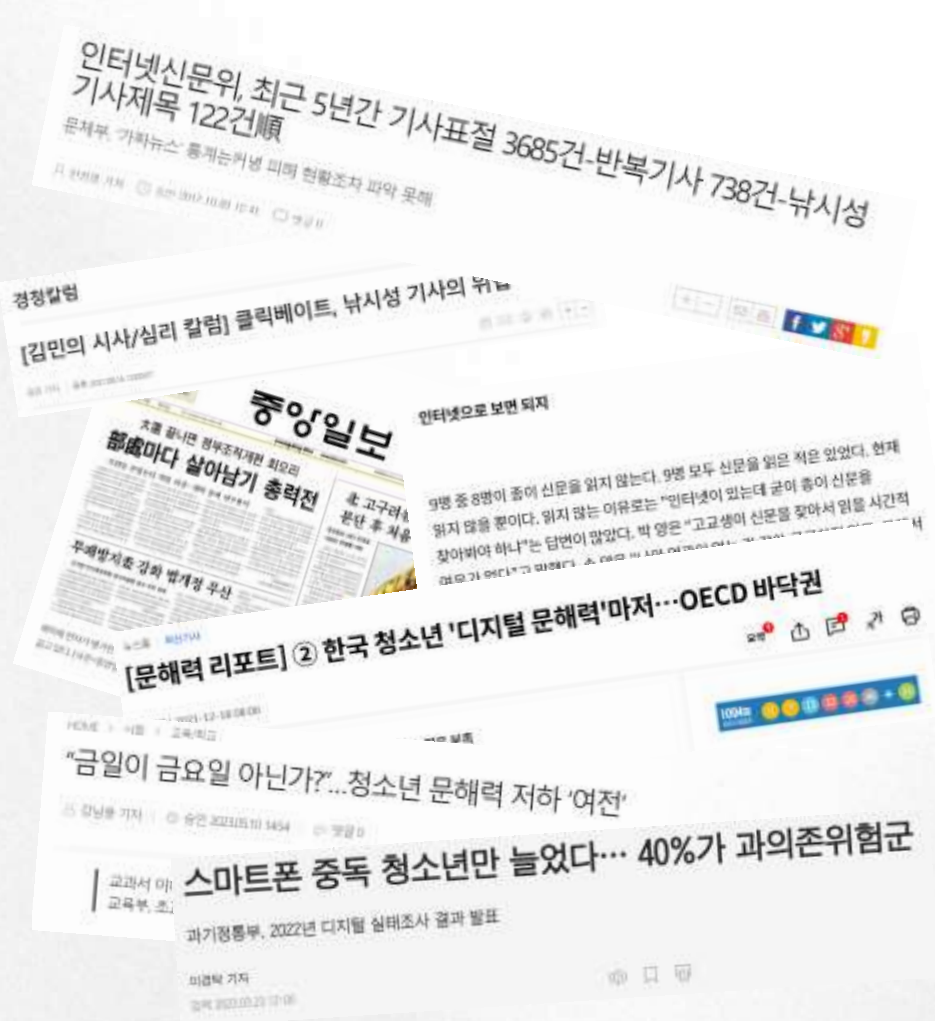
OTT, SNS 등에 과도하게 시간을 사용

# 문해력 부족

뉴스 특유의 어려운 문맥

# 낱시성 제목

관심만을 끌기 위한 제목



# 현대 사회 뉴스

**FAST** 중요 내용만 **빠르게** 읽고 싶어한다.

**FAKE** **낱시성** 제목의 뉴스가 판을 친다.

**FRUST** 긴글에 대한 **답답함** 때문에 읽지않는다.

우리는 뉴스 소비를 혁신적으로 이끌어 나가기 위해  
뉴스 기사를 핵심적인 내용으로 간결하게 요약하는  
서비스를 개발하기로 했다.





03

기획 목표

# 서비스 기획 목표

## ✓ 실시간

실시간의 뉴스를 바로 볼 수 있도록 한다.

## ✓ 기사를 요약

현대 사회는 점점 긴 글 보다 짧은글을 선호한다.

## ✓ 본문

요약 기사를 읽고 원하면 본문 기사를 바로 볼 수 있게 한다.

## ✓ 모바일로 이용 가능

공간의 제약을 최소화하여 모바일로도 서비스를 사용가능하게 한다.



04

## 서비스 설명

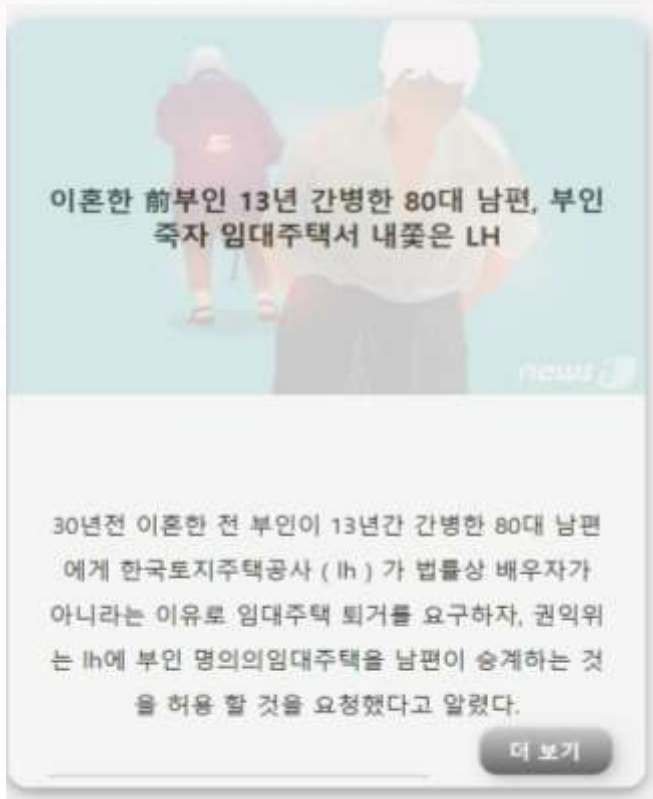
# IMPORTANT POINTS



- 누구나 쉽게 이용하는 것
- 심플하면서 가독성이 좋은 페이지 구현
- 최신의 기사를 가져온다



# 주요 기능



## 본문 스크립트 요약

bert의 요약 모델을 기반으로 기사의 주요내용을 요약한다.

## 최소한의 클릭과 움직임(편의성)

편의성을 고려해 마우스 커서만 위치해도 나오는 손쉬운 사용

## 다양한 기사를 자유롭게 빠르게

여러 카테고리의 많은 양의 기사들을 한눈에 쉽게 볼 수 있다.

## 기존 뉴스와 Flash News만의 차별화된 강점

Original news	Flash news
접근성과 편의성이 떨어진다.	접근성과 편의성이 좋다
기사를 읽는 시간이 오래걸린다.	중요 부분 요약을 통해 시간 단축
여러 기사들을 들어가서 일일이 다 봐야한다.	한눈에 많은 기사를 볼 수 있음
낱시성 제목의 기사가 많다.	제목과 함께 요약내용이 보여 낱시성 기사를 피할 수 있음

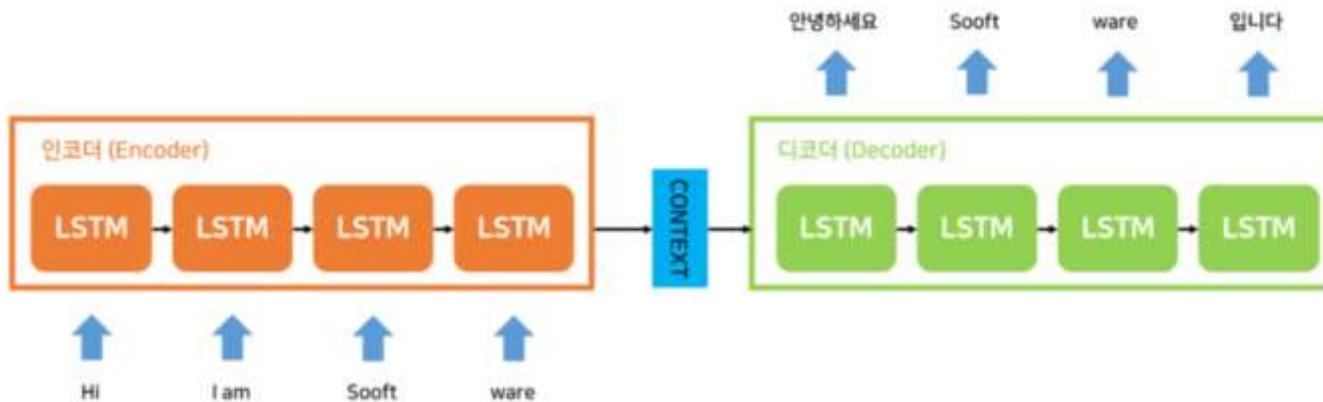


05

## 모델 설명

MODEL

seq2seq - Bert Model



# MODEL

## seq2seq - Bert Model

```
***
```

```
EncoderDecoderModel{
```

```
  (encoder): BertModel{
```

```
    (embeddings): BertEmbeddings{
```

```
      (word_embeddings): Embedding(42000, 768, padding_idx=0)
```

```
      (position_embeddings): Embedding(512, 768)
```

```
      (token_type_embeddings): Embedding(2, 768)
```

```
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
```

```
      (dropout): Dropout(p=0.1, inplace=False)
```

```
    }
```

```
  (encoder): BertEncoder{
```

```
    (layer): ModuleList{
```

```
      (0-11): 12 x BertLayer{
```

```
        (attention): BertAttention{
```

```
          (self): BertSelfAttention{
```

```
            (query): Linear(in_features=768, out_features=768, bias=True)
```

```
            (key): Linear(in_features=768, out_features=768, bias=True)
```

```
            (value): Linear(in_features=768, out_features=768, bias=True)
```

```
            (dropout): Dropout(p=0.1, inplace=False)
```

```
          }
```

```
        (output): BertSelfOutput{
```

```
          (dense): Linear(in_features=768, out_features=768, bias=True)
```

```
          (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
```

```
          (dropout): Dropout(p=0.1, inplace=False)
```

```
        }
```

```
      }
```

```
    (intermediate): BertIntermediate{
```

```
      (dense): Linear(in_features=768, out_features=3072, bias=True)
```

```
      (intermediate_act_fn): GELUActivation()
```

```
    }
```

```
  (cls): BertOnlyMLMHead{
```

```
    (predictions): BertLMPredictionHead{
```

```
      (transform): BertPredictionHeadTransform{
```

```
        (dense): Linear(in_features=768, out_features=768, bias=True)
```

```
        (transform_act_fn): GELUActivation()
```

```
        (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
```

```
      }
```

```
    (decoder): Linear(in_features=768, out_features=42000, bias=True)
```

```
  }
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```

```
***
```

## MODEL

## seq2seq - Bert Model

Hidden size	layers	max length	batch size	learning rate	training steps
768/768	12/12	512/512	16	5e-5	20K

	NSMC (acc)	Naver NER (F1)	PAWS (acc)	KorNLI (acc)	KorSTS (spearman)	Question Pair (acc)	Korean-Hate-Speech (Dev) (F1)
bertshared-kor-base	90.87	87.27	82.8	82.32	84.31	95.25	68.45

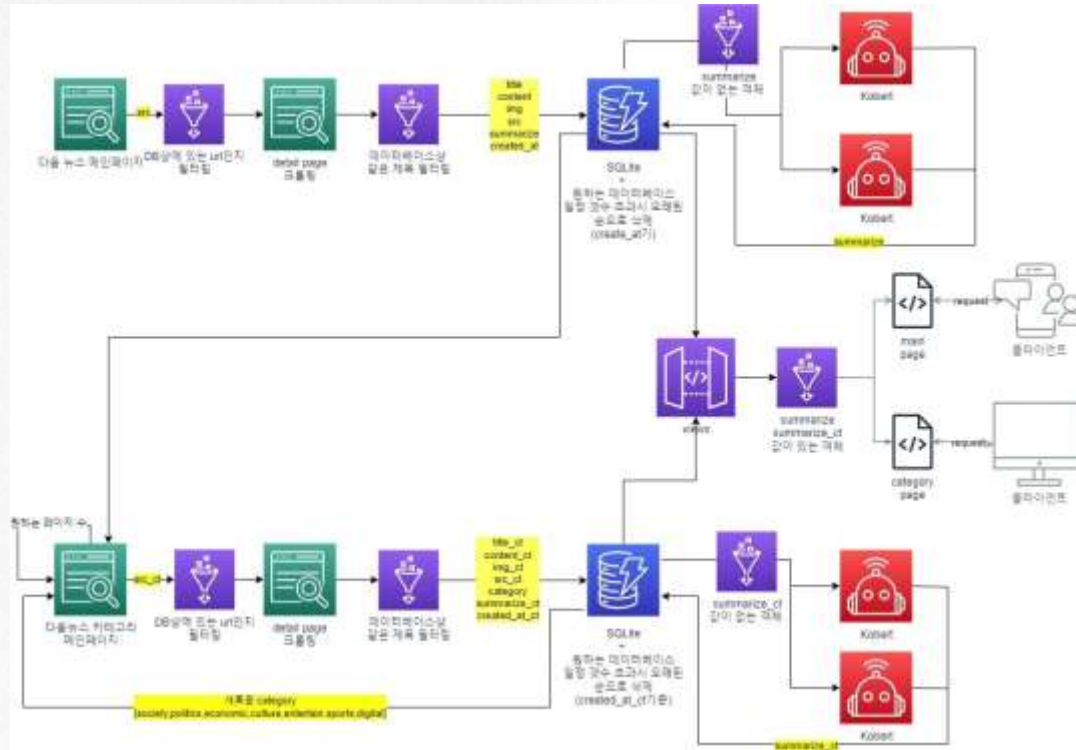


06

## 백엔드 및 서버

# WEB

## 서비스 구조

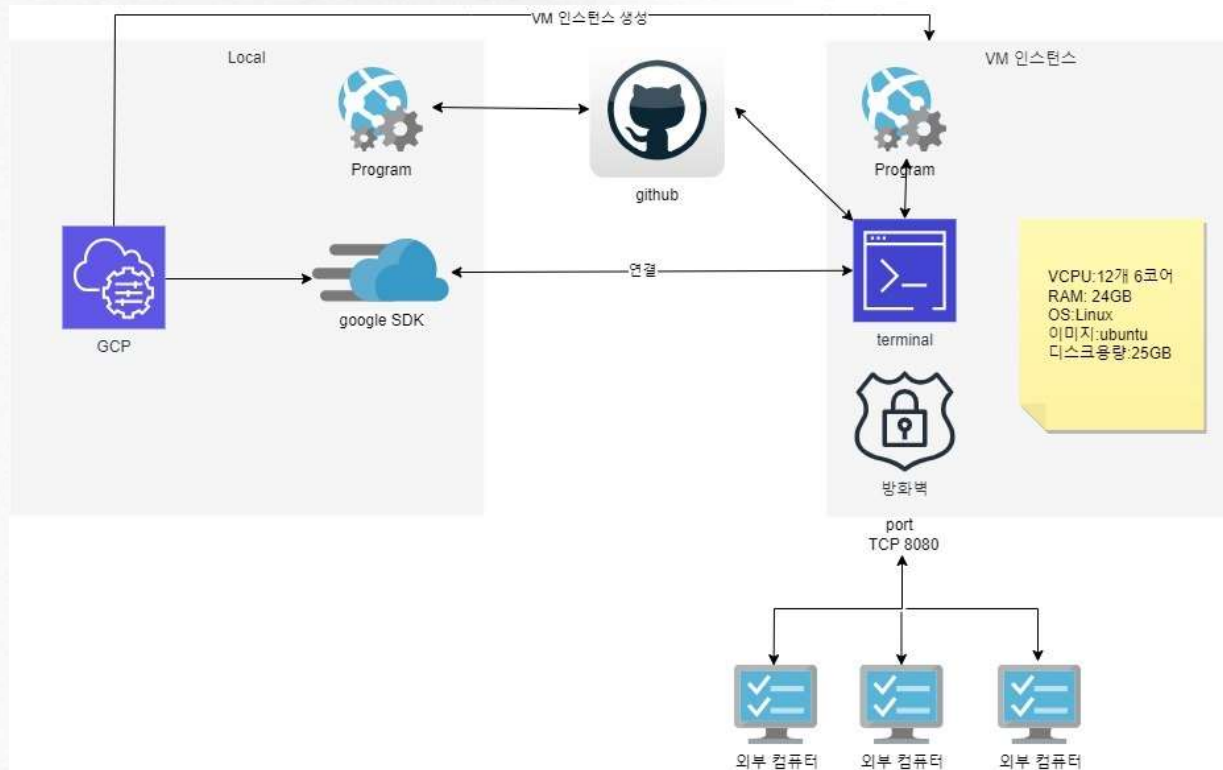




```
import concurrent.futures
2 usages: 1 Shin Geonhee +1
def summarize_parallel(content_list):
    # content_list는 각각의 content를 포함하는 리스트입니다.
    # 요약된 결과를 저장할 리스트
    summarized_results = []
    # 요약 모델을 병렬로 실행하는 함수
    1 Shin Geonhee
    def run_summary(content):
        return summary(content)
    max_workers = min(len(content_list), 2) # 예: 최대 2개의 스레드 사용(본인 컴퓨터에 맞는 스레드 사용)
    with concurrent.futures.ThreadPoolExecutor(max_workers=max_workers) as executor:
        # 각 content에 대해 run_summary 함수를 실행하고 결과를 리스트에 저장
        results = list(executor.map(run_summary, content_list))
        summarized_results.extend(results)
    return summarized_results
```

# SERVER

## 서버 구조



- Nginx, 지-유니콘(gunicorn)을 이용한 웹서버 안정화
- https를 사용한 웹사이트 신뢰도 향상

# 추후 발전 가능성

- **모바일 어플리케이션 개발**

- 모바일 웹으로 접속이 가능하지만 어플리케이션 전용으로 만들지 않아 불안정

- **해외 뉴스에 대한 기사 요약에 모델에 추가**

- 한국어 요약만 가능한 모델의 한계

- **서버 컴퓨터의 성능에 따라 더욱 빠른 요약 연산과 거대한 데이터베이스도 사용가능**

# 참고자료

## Template & Design

- <https://slidesgo.com/theme/gossip-fact-vs-fiction>
- <https://freefrontend.com/css-menu/page/4/>
- <https://inpa.tistory.com/category/Style%20Sheet/CSS>
- <https://studiomeal.com/archives/852>

## Article

- OTT 이용률 57% vs 신문 구독률 5% - views&news
- 사람들이 가장 많이 보는 뉴스 미디어는?... 인터넷-모바일 뉴스 1위 - 도농라이프타임즈
- 좋은 뉴스 사이트는 기사 오류 바로잡고 낚시성 제목 없어야 - 조선일보
- "스마트폰 있으면 TV 뉴스 안봐도 불편함 없다" ... 유튜브도 미디어 -한국경제

## Other

- <https://github.com/kiyoungkim1/LMkor> - Bert 모델
- <https://brunch.co.kr/@topasvga/168> - 서버 사용법



# THANKS!

**Do you have any questions?**

starhochoitest@gmail.com

010 2051 3409

<http://www.flash-news.kro.kr:8080/>



---

Please keep this slide for attribution