
SONANE CAMP
Camping AI

CONTENTS

1

Team Member

OUR TEAM

Members of Team Sonane Camp

이동주

안드로이드, 백엔드 구현, RAG 개선

윤찬구

프론트엔드, 백엔드 총괄 및 구현, DB 연동 및 구축

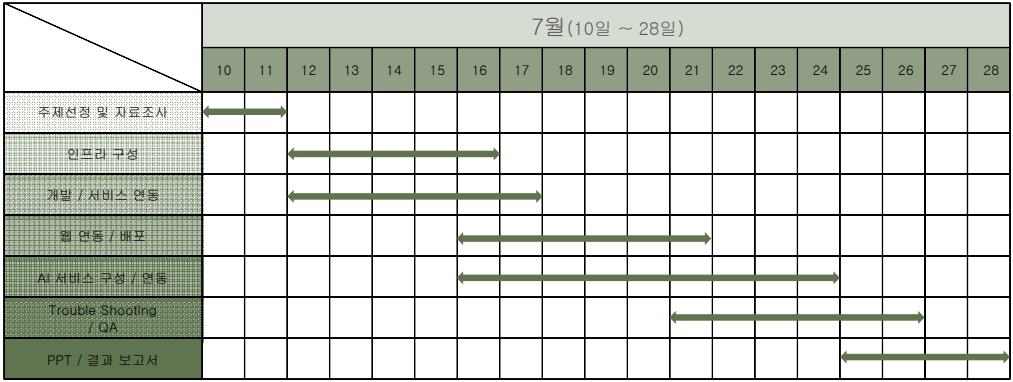
안유진

인프라 총괄, CICD 개선, RAG 구현, DB 연동 및 구축

박재훈

초기 RAG 구현, 초기 CICD 파이프라인 및 인프라 구성

Schedule


7월(10일 ~ 28일)
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
주제선정 및 자료조사
인프라 구성
개발 / 서비스 연동
웹 연동 / 배포
AI 서비스 구성 / 연동
Trouble Shooting / QA
PPT / 결과 보고서

2

기획 의도

2. 기획 의도

대중의 일상으로 자리 잡은 캠핑

700만 명 이상이 즐기는 국민 레저로 성장한 캠핑
이제 캠핑은 소수의 취미가 아닌, 누구나 즐기는 생활 문화가 되었습니다

3

기술 구현

3. 기술 구현

Our Service

NCP, Frontend, Backend, AI Service 등등

NAVER CLOUD PLATFORM

- VPC
- ALB
- Auto Scaling
- Server
- MySQL
- NAT Gateway
- ACG
- SSL VPN

3. 기술 구현

Our Service

NCP, Frontend, Backend, AI Service 등등

- CSS
- Java Script
- jQuery
- Java
- JSP
- Python
- Kotlin
- MyBatis
- Java spring
- Maven
- Tomcat
- Github

3. 기술 구현

Our Service

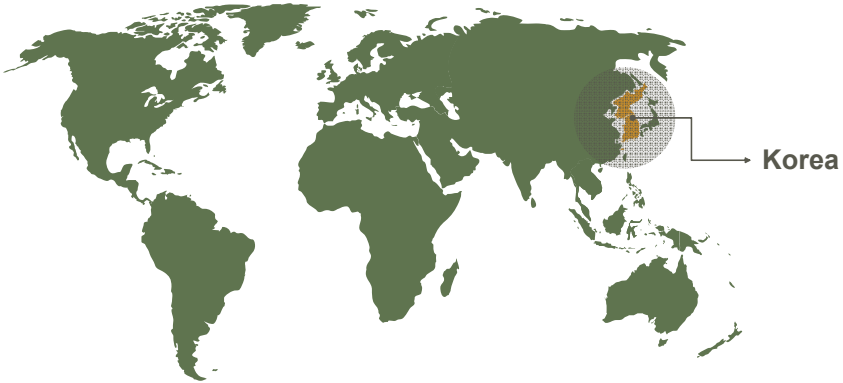
NCP, Frontend, Backend, AI Service 등등

- AJAX
- JDK
- Numpy
- Matplotlib
- Jenkins
- CLOVA X
- CLOVA Voice
- CSR
- Embedding
- Lang Chain
- Vector DB
- Object Storage

3. 기술 구현

Region Select

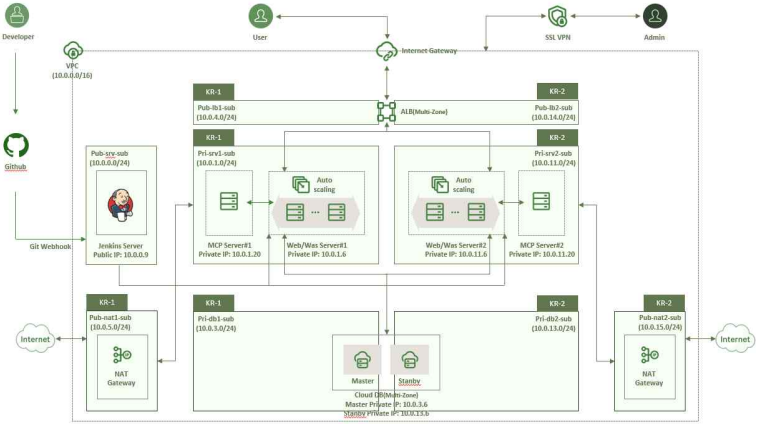
Republic of Korea 기준



3. 기술 구현

Infra Architecture

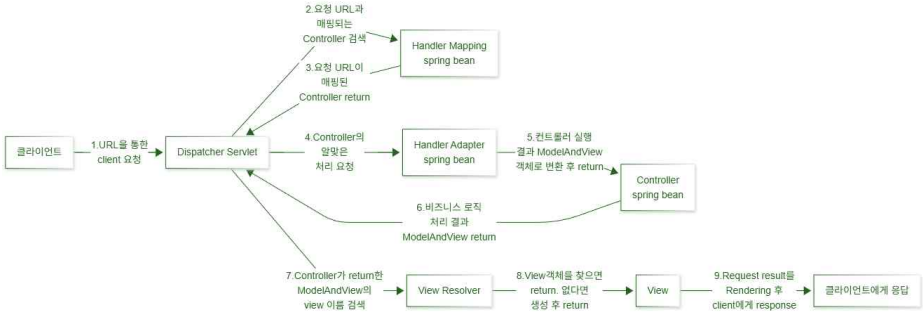
무중단 배포 및 Failover 중점에 맞춰 설계



3. 기술 구현

Spring Architecture

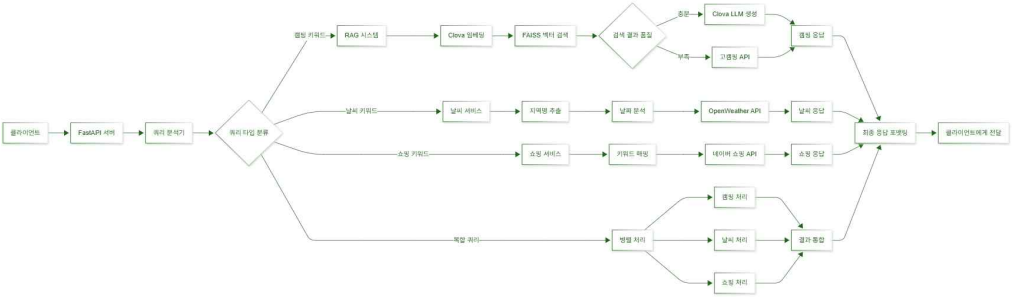
Spring MVC 설계



3. 기술 구현

AI Service Architecture

RAG 및 에이전트가 특정 키워드를 인식하여 응답하도록 설계

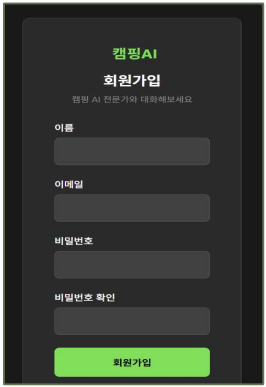


3. 기술 구현

Source Code

실행화면 – 회원가입

```
@PostMapping(value = "/signup")
public void insertMB(@ModelAttribute("MBVO") MBVO vo, HttpServletResponse response, HttpServletRequest request)
        throws Exception {

    String user_name = request.getParameter("name");
    String user_email = request.getParameter("email");
    String user_pwd = request.getParameter("password");
    String user_id = request.getParameter("user_id");

    vo.setUser_id(user_id);
    vo.setName(user_name);
    vo.setEmail(user_email);
    vo.setPassword(user_pwd);

    int result = service.registerUser(vo);

    JSONObject json = new JSONObject();
    if (result == 1) {
        json.put("user", result);
        json.put("sucess", "sucess");
    } else {
        json.put("null", "null");
    }

    JsonWrite.print json  response);
}
```

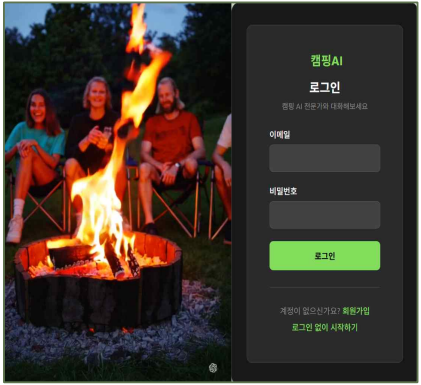


3. 기술 구현

Source Code

실행화면 – 로그인

```
@PostMapping(value = "/signin" )
public void insertLogin(@ModelAttribute("LoginVO") LoginVO vo,
    String user_email = request.getParameter("email");
    String user_pwd = request.getParameter("password");

    vo.setEmail(user_email);
    vo.setPassword(user_pwd);
    String[] mobStr = user_email.split("@");
    String user = mobStr[0];

    int result = service.checkUser(vo);

    JSONObject json = new JSONObject();
    if(result == 1) {
        json.put("user", user);
    }else {
        json.put("fail", "fail");
    }

    JsonHndr.print(json, response);
}
```

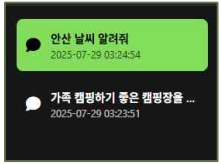


3. 기술 구현

Source Code

실행화면 – 채팅방 조회

```
@RequestMapping(value = "/getChatRooms", method = RequestMethod.GET)
public void getChatRooms(HttpServletRequest request, HttpServletResponse response) throws Exception {
    String user_id = request.getParameter("user_id");

    JSONObject json = new JSONObject();

    try {
        if (user_id == null || user_id.trim().isEmpty()) {
            json.put("success", false);
            json.put("message", "사용자 ID가 필요합니다.");
            JsonWndr.print(json, response);
            return;
        }

        // 사용자의 채팅방 목록 조회
        List<ChatLibraryVO> chatRoomList = service.selectChatRoomById(user_id);

        JSONArray chatRoomsArray = new JSONArray();

        for (ChatLibraryVO chatRoom : chatRoomList) {
            JSONObject roomObj = new JSONObject();
            roomObj.put("user_chat_library_id", chatRoom.getUser_chat_library_id());
            roomObj.put("user_id", chatRoom.getUser_id());
            roomObj.put("create_at", chatRoom.getCreate_at());
            roomObj.put("title", chatRoom.getTitle() != null ? chatRoom.getTitle() : "제목없음");
            roomObj.put("preview_text", chatRoom.getPreview_text());
            chatRoomsArray.put(roomObj);
        }

        json.put("success", true);
        json.put("chatRooms", chatRoomsArray);
```

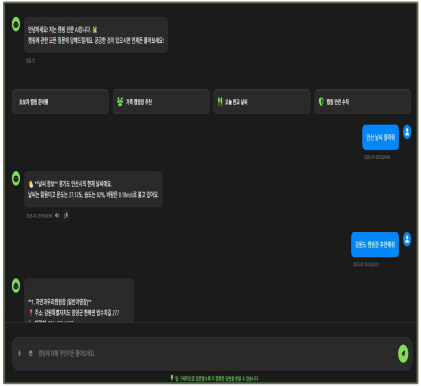


3. 기술 구현

Source Code

실행화면 – 채팅

```
@RequestMapping(value = "/sendMessage", method = RequestMethod.POST)
public void insertChatLibrary(@ModelAttribute("ChatLibraryVO") ChatLibraryVO vo,
    String user_id = request.getParameter("savedUser");
    String create_at = request.getParameter("currentTime");
    String message = request.getParameter("message");
    String selectedChatRoomId = request.getParameter("selectedChatRoomId");
    String newChatRoomId = request.getParameter("newChatRoomId");
    String images = request.getParameter("images");

    int result = 0;
    int uResult = 0;
    boolean isNewChatRoom = false;

    JSONObject json = new JSONObject();

    System.out.println("images : " + images);
    // 새 채팅방 생성 처리
    if (newChatRoomId != null && !newChatRoomId.trim().isEmpty()) {
        try {
            int nChatId = Integer.parseInt(newChatRoomId);
            vo.setUser_chat_library_id(nChatId);
            vo.setUser_id(user_id);
            vo.setCreate_at(create_at);
            result = service.insertChatLibrary(vo);
            isNewChatRoom = true;

            if (result == 1) {
                // 새 채팅방이 생성되었으므로 이어서 메시지도 저장
                UserChatHistoryVO uvo = new UserChatHistoryVO();
                uvo.setUser_chat_library_id(nChatId);
                uvo.setUser_id(user_id);
                uvo.setAsk_value(message);
```

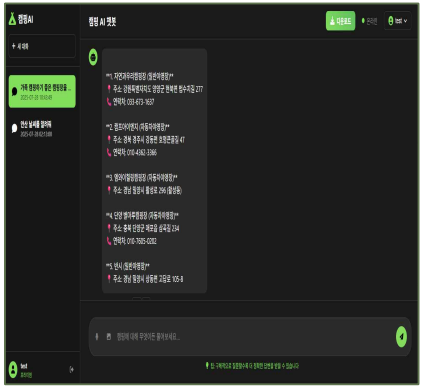


3. 기술 구현

Source Code

실행화면 – 채팅 응답

```
@RequestMapping(value = "/sendAiMessage", method = RequestMethod.POST)
public void insertChatLibrary(@ModelAttribute("UserChatHistoryVO") UserChatHistoryVO vo
    String user_id = request.getParameter("savedUser");
    String selectedChatRoomId = request.getParameter("finalChatRoomId");
    String newChatRoomId = request.getParameter("newChatRoomId");
    String answer_value = request.getParameter("message");
    String currentTime = request.getParameter("currentTime");

    String chatId = "";
    if(selectedChatRoomId != null) {
        chatId = selectedChatRoomId;
    }else if(newChatRoomId != null) {
        chatId = newChatRoomId;
    }

    vo.setAnswer_create_at(currentTime);
    vo.setAnswer_value(answer_value);
    vo.setUser_chat_library_id(Integer.parseInt(chatId));
```



3. 기술 구현

Source Code

실행화면 – 채팅방 삭제

```
@RequestMapping(value = "/deleteChatRoom", method = RequestMethod.POST)
public void deleteChatRoom(HttpServletRequest request, HttpServletRespon
    String chatRoomId = request.getParameter("chatRoomId");

    JSONObject json = new JSONObject();

    try {
        if (chatRoomId == null || chatRoomId.trim().isEmpty()) {
            json.put("success", false);
            json.put("message", "채팅방 ID가 필요합니다.");
            JsonHndr.print(json, response);
            return;
        }

        // 채팅방 삭제
        boolean isDeleted = service.deleteChatRoom(chatRoomId);

        if (isDeleted) {
            json.put("success", true);
            json.put("message", "채팅방이 성공적으로 삭제되었습니다.");
        } else {
            json.put("success", false);
            json.put("message", "채팅방 삭제에 실패했습니다.");
        }

    } catch (Exception e) {
        System.err.println("채팅방 삭제 오류: " + e.getMessage());
        e.printStackTrace();
```

3. 기술 구현

Source Code

TTS, STT

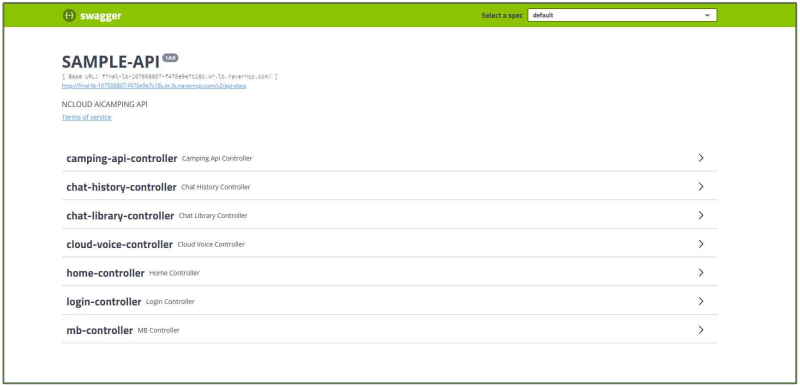
```java
@RequestMapping(value = "/textToSpeech", method = RequestMethod.POST)
public void textToSpeech(HttpServletRequest request, HttpServletResponse response) throws Exception {
    String text = request.getParameter("text");
    text = text.replace("<br>", "\n"); // 또는 공백으로

    JSONObject json = new JSONObject();

    try {
        if (text == null || text.trim().isEmpty()) {
            json.put("success", false);
            json.put("message", "텍스트가 필요합니다.");
            JsonHndr.print(json, response);
            return;
        }

        String clientId = "2xhuz460g1";
        String clientSecret = "1b2qJ7rp8kGTSIT5n7TC4Ze33ChzNBPgeVsiAqLz";

        String encodedText = URLEncoder.encode(text, "UTF-8");
        String apiURL = "https://naveropenapi.apigw.ntruss.com/tts-premium/v1/tts";

        URL url = new URL(apiURL);
        HttpURLConnection con = (HttpURLConnection)url.openConnection();
        con.setRequestMethod("POST");
        con.setRequestProperty("X-NCP-APIGW-API-KEY-ID", clientId);
        con.setRequestProperty("X-NCP-APIGW-API-KEY", clientSecret);
        con.setDoOutput(true);
        //이곳에 speaker의 음성모델 선택임 볼륨 음성크기 5는1.5배크게 -5는 0.5배 스피드 음성속도 10은2배느리게 -5는 2배빠르게 피치 음높낮이 5가 1.2
        String postParams = "speaker=nes_c_hyeri&volume=0&speed=0&pitch=0&format=mp3&text=" + encodedText;
        DataOutputStream wr = new DataOutputStream(con.getOutputStream());
        wr.writeBytes(postParams);
wr.flush();
wr.close();

int responseCode = con.getResponseCode();
BufferedReader br;

if(responseCode == 200) {
    InputStream is = con.getInputStream();
    int read = 0;
    byte[] bytes = new byte[1024];

    // 파일명 생성 (타임스탬프 사용)
    String fileName = "tts_" + System.currentTimeMillis() + ".mp3";
    String filePath = request.getSession().getServletContext().getRealPath("/resources/audio/") + fileName;

    System.out.println("filePath : " + filePath);
    // 디렉토리 생성
    File dir = new File(request.getSession().getServletContext().getRealPath("/resources/audio/"));
    if (!dir.exists()) {
        dir.mkdirs();
    }

    File f = new File(filePath);
    f.createNewFile();
    FileOutputStream outputStream = new FileOutputStream(f);

    while ((read = is.read(bytes)) != -1) {
        outputStream.write(bytes, 0, read);
    }
    is.close();
```

```java
@PostMapping("/speechToText")
public void speechToText(@RequestParam("file") MultipartFile file, HttpServletResponse response) {

    JSONObject json = new JSONObject();
    try {
        String language = "Kor";
        String apiURL = "https://naveropenapi.apigw.ntruss.com/recog/v1/stt?lang=" + language;

        URL url = new URL(apiURL);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setUseCaches(false);
        conn.setDoOutput(true);
        conn.setDoInput(true);
        conn.setRequestProperty("Content-Type", "application/octet-stream");
        conn.setRequestProperty("X-NCP-APIGW-API-KEY-ID", clientId);
        conn.setRequestProperty("X-NCP-APIGW-API-KEY", clientSecret);

        // 음성 파일 스트림 전송
        OutputStream outputStream = conn.getOutputStream();
        InputStream inputStream = file.getInputStream();
        byte[] buffer = new byte[4096];
        int bytesRead;
        while ((bytesRead = inputStream.read(buffer)) != -1) {
            outputStream.write(buffer, 0, bytesRead);
        }
        outputStream.flush();
        inputStream.close();

        // 응답 처리
// 응답 처리
int responseCode = conn.getResponseCode();

BufferedReader br;

StringBuilder result = new StringBuilder();
String line;
if (responseCode == 200) {
    br = new BufferedReader(new InputStreamReader(conn.getInputStream(), "UTF-8"));
    while ((line = br.readLine()) != null) {
        result.append(line);
    }
    br.close();

    JSONObject clovaJson = new JSONObject(result.toString());
    String recognizedText = clovaJson.optString("text", ""); // 실제 텍스트

    json.put("success", true);
    /* json.put("audioUrl", "/resources/voice/" ); */
    json.put("message", "STT 성공 완료");
    json.put("text", recognizedText);
} else {
    br = new BufferedReader(new InputStreamReader(conn.getErrorStream(), "UTF-8"));
    while ((line = br.readLine()) != null) {
        result.append(line);
    }
    br.close();
    System.out.println("❌ 오류 응답 코드: " + responseCode);
    json.put("success", false);
    json.put("error", result.toString());
```

3. 기술 구현

Source Code

Swagger API 문서



3. 기술 구현

Source Code

S3 확장자 기준 파일 다운로드

```python
# ------------------------------
# S3에서 확장자 기준 파일 다운로드
# ------------------------------
def download_multiple_files():
    os.makedirs(LOCAL_DOWNLOAD_DIR, exist_ok=True)
    s3 = boto3.client(
        's3',
        aws_access_key_id=NCP_ACCESS_KEY_ID,
        aws_secret_access_key=NCP_SECRET_ACCESS_KEY,
        endpoint_url=S3_ENDPOINT_URL,
    )
    response = s3.list_objects_v2(Bucket=BUCKET_NAME)
    object_keys = [item["Key"] for item in response.get("Contents", [])]
    filtered_keys = [key for key in object_keys if any(key.endswith(ext) for ext in OBJECT_FILE_EXTENSIONS)]

    local_paths = []
    for key in filtered_keys:
        filename = os.path.basename(key)
        local_path = os.path.join(LOCAL_DOWNLOAD_DIR, filename)
        print(f"📥 Downloading {key} to {local_path}")
        s3.download_file(BUCKET_NAME, key, local_path)
        local_paths.append(local_path)
    return local_paths
```

3. 기술 구현

Source Code

CLOVA 임베딩 및 LLM 모델

```python
# ------------------------------
# Clova 임베딩 클래스 (OpenAI 호환 API 방식)
# ------------------------------
class ClovaEmbedding(Embeddings):
    def embed_documents(self, texts: List[str]) -> List[List[float]]:
        headers = {
            "Authorization": f"Bearer {CLOVA_API_KEY}",
            "Content-Type": "application/json",
        }

        embeddings = []
        for text in texts:
            data = {
                "model": "bge-m3",  # 임베딩용 모델명 (클로바 스튜디오 문서 참고)
                "input": text
            }
            for retry in range(3):
                response = requests.post(CLOVA_EMBEDDING_API_URL, headers=headers, json=data)
                if response.status_code == 429:
                    print(f"[⏳] Rate limit exceeded (429), waiting... (retry {retry+1})")
                    time.sleep(1.5)
                    continue
                elif response.status_code != 200:
                    print(f"[❌] {response.status_code} - {response.text}")
                    response.raise_for_status()
                break
            else:
                raise Exception(f"[🚫] Failed after 3 retries (429): {text[:50]}...")

            json_data = response.json()
            if "data" not in json_data or not json_data["data"]:
                raise Exception(f"[❌] Invalid response format: {json_data}")
            embedding_vector = json_data["data"][0]["embedding"]
            embeddings.append(embedding_vector)
            time.sleep(1.2)
        return embeddings
```

```python
# ------------------------------
# Clova LLM 클래스 (OpenAI 호환 API 방식)
# ------------------------------
class ClovaLLM(LLM):
    clova_api_key: str = CLOVA_API_KEY
    clova_url: str = CLOVA_LLM_API_URL

    @property
    def _llm_type(self) -> str:
        return "clova-llm"

    def _call(self, prompt: str, stop: Optional[List[str]] = None) -> str:
        headers = {
            "Authorization": f"Bearer {self.clova_api_key}",
            "Content-Type": "application/json",
        }

        body = {
            "model": "HCX-005",  # 채팅용 모델명
            "messages": [
                {"role": "system", "content": "You are a helpful assistant."},
                {"role": "user", "content": prompt}
            ],
            "temperature": 0.7,
            "max_tokens": 512,
            "top_p": 0.8
        }

        response = requests.post(self.clova_url, headers=headers, json=body)
        if response.status_code != 200:
            print(f"[ERROR] HTTP {response.status_code}: {response.text}")
            response.raise_for_status()

        result = response.json()
```

3. 기술 구현

Source Code

문서 로드 및 분할 및 파이프라인

```python
# ------------------------------
# 문서 로드 및 분할
# ------------------------------
def load_and_split_documents(file_paths: List[str]):
    all_chunks = []
    splitter = RecursiveCharacterTextSplitter(chunk_size=1500, chunk_overlap=200)

    for file_path in file_paths:
        print(f"📄 Processing {file_path}")
        if file_path.endswith(".pdf"):
            loader = PyPDFLoader(file_path)
        elif file_path.endswith(".csv"):
            loader = CSVLoader(file_path=file_path, encoding="utf-8")
        else:
            print(f"❌ Unsupported file type: {file_path}")
            continue

        documents = loader.load()
        chunks = splitter.split_documents(documents)
        all_chunks.extend(chunks)

    return all_chunks
```

```python
# ------------------------------
# 전체 파이프라인 실행
# ------------------------------
def run_rag_pipeline(query: str):
    # 1. 다운로드 및 문서 분할
    local_files = download_multiple_files()
    if not local_files:
        print("❌ No files downloaded.")
        return

    chunks = load_and_split_documents(local_files)

    # 2. FAISS 인덱스 로드 or 생성
    embedding = ClovaEmbedding()
    if os.path.exists(FAISS_INDEX_PATH):
        print("✅ Reusing existing FAISS index...")
        db = FAISS.load_local(
            FAISS_INDEX_PATH,
            embedding,
            allow_dangerous_deserialization=True,
        )
    else:
        print("🔨 Building new FAISS index...")
        db = FAISS.from_documents(chunks, embedding)
        db.save_local(FAISS_INDEX_PATH)

    retriever = db.as_retriever()

    # 3. LLM 사용해서 QA 실행
    clova_llm = ClovaLLM()
    qa = RetrievalQA.from_chain_type(llm=clova_llm, retriever=retriever, chain_type="stuff")

    answer = qa.run(query)

    print("\n✅ [Answer]")
    print(answer)
```

Trouble Shooting

RAG / LLM 복합 질문 응답 문제

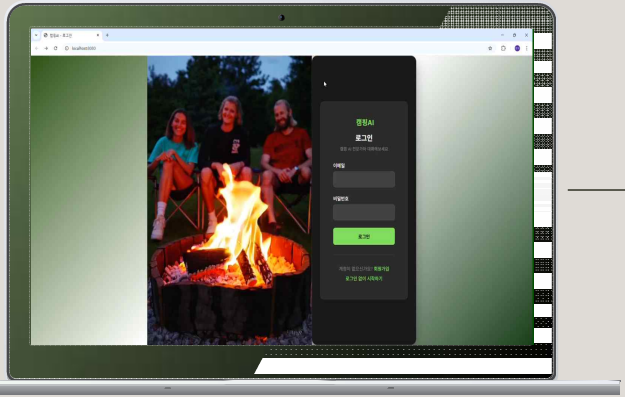
- 랜딩장 추천과 날씨는 개별적으로 응답 정상
- 복합 질문 시 하나만 응답하거나 불완전한 답변이 제공되는 현상
- 프롬프트 구조 개선 및 RAG 응답 컨텍스트 수정으로 문제 해결

Jenkins 인코딩 문제

- Git 저장소의 EUC-KR 파일로 인해 Jenkins 빌드 실패 발생
- UTF-8 변환 스크립트를 파이프라인에 추가하고, 저장소 인코딩 일괄 정리하여 해결

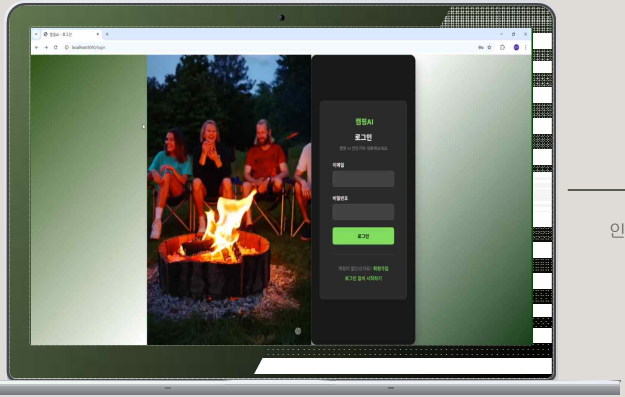
TTS 기능 미작동 현상

- API 호출은 성공하나, 실제로는 에러 메시지와 함께 미작동 현상 발생
- Object Storage를 활용하여 MP3 파일을 업로드 후, 해당 URL을 클라이언트측에 반환하도록 소스를 수정하여 해결



프로젝트 시연 영상

인트로 – 로그인하지 않은 사용자 – 메인 채팅화면



프로젝트 시연 영상

인트로 – 회원가입 – 로그인 – 메인 채팅화면 – 로그아웃

Provides Android OS

4

향후 계획

4. 향후 계획

Future Plans

Towards a Smarter, Safer, and More Efficient AI Service

RAG / LLM 개선

현재 학습시킨 RAG 모델은 부자연스러운 딱딱한 모습이 다소 보임 이를 개선하기 위해 LLM의 프롬프트와 각종 API를 통하여 훨씬 더 자연스러운 이상적인 RAG로 개선 예정

TTS 개선

현재 서비스 중인 TTS는 mp3파일을 object storage에 받아 이를 호출하는 형식으로 tts를 이용할 때 마다 비용이 듦으로 object storage가 아닌 Loadbalancer의 sticky session 과 target group을 이용하여 비용 절감 예정

CI/CD 파이프라인 축소 및 간략화

현시점 파이프라인은 너무 간결하지 못하고 복잡성을 띄고 있으므로 파이프라인의 불필요한 내용을 줄이고 developer 입장에서 더욱 보기 쉽게 간략화 예정

보안성 강화

어플리케이션 서비스를 위하여 ACG 이외에도 모니터링을 위한 Security Monitoring, 모바일을 위한 App Safer, 취약점 판단을 위한 Web Security Checker 등 다양한 보안 서비스를 이용할 예정

5

Project Review

SONANE CAMP

이동주

캠핑 수요 증가에 맞춰 사용자 맞춤형 AI 서비스를 기획하고 설계했다.
데이터 기반 추천 시스템과 사용자 경험 중심 설계의 중요성을 체감했다.
팀장으로서 개발 일정 조율과 협업을 이끌며 유의미한 프로젝트 경험을 쌓았다.

윤찬구

4인이 협업해 Java 기반으로 캠핑 AI 챗봇을 개발하고, RAG 기술을 적용했다.
NCP 인프라에서 CI/CD 파이프라인을 구축해 자동화된 배포 환경을 경험했다.
개발부터 운영까지의 과정을 주도적으로 수행하여 보람찼다.

안유진

현업에서 자유롭게 활용하기 어려운 기술들을 실험하고 적용해볼 수 있었다.
클라우드 인프라와 RAG 기반 기술을 통해 실무 중심의 경험을 쌓았다.
팀원들과의 협업으로 문제를 해결하며 챗봇 완성도를 높이는 과정이 뜻깊었다.

박재훈

캠핑 추천 챗봇을 팀원 4명과 협업하며 리더십과 소통 능력을 키웠다.
프롬프트 설계가 AI 응답 품질에 중요한 요소임을 직접 경험했다.
RAG 기반 기술과 클라우드 서비스를 실무에 적용하며 기술 역량을 높였다.

Q & A


Thank you