AI Call을 활용한

인공지능 음식점 전화 예약 서비스 개발

카피바라팀 배누리, 김호정, 전사영, 박현아 2024.11.14



목차

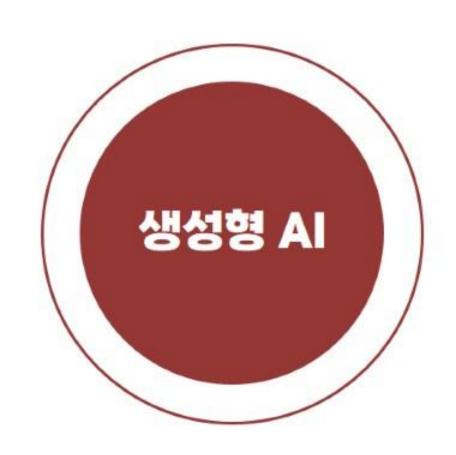
01 배경 및 목표 02 서비스시나리오 **03** 코드 구현

04 최종결과 05 레퍼런스 06 타임테이블

61 배경 및 목표

배경 및 목표





Point

Chat Gpt, Clova 등 다양한 생성형 ai의 발전 생성형 ai를 활용하여 노동력을 절감할 수 있는 효율적인 예약 서비스 개발

02 서비스 시나리오





최고의 고객께 최고의 서비스를, 누리카세

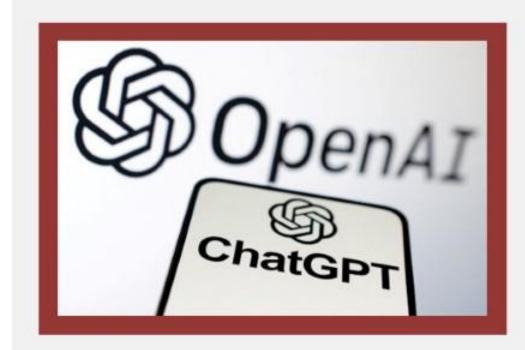
누리카세는 100% 예약제 식당으로, 최고의 음식을 고객님께 제공드리기 위하여 한정된 좌석을 제공합니다.

사용 기술



ASR

입력된 음성을 Text로 변환



Open Al

Open AI에서 제공하는 GPT 4.0을 활용하여 AI 응답 생성



TTS

Text를 음성으로 변환

서비스시나리오



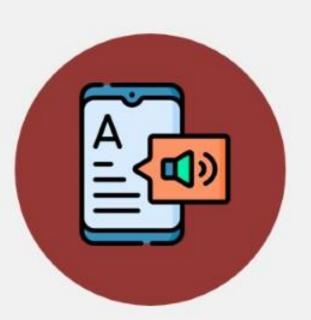
사용자 음성 입력



텍스트 변환



AI 응답 생성



음성 변환

서비스시나리오

예약 요청 — 예약 시간 확인 — 예약 진행

→ 예약 확정 → 추가 질문


```
# 초기 메시지 설정
messages = [
     "role": "system",
     "content": """안녕하세요, 누리카세 예약 도무미입니다. 저는 누리카세 식당 예약을 도와드리는 챗봇입니다.
     아래의 정보를 바탕으로 사용자의 질문에 친절하고 간결하게 답변합니다. 예약 관련 정보가 정확할 경우 추가 질문을 하지 않습니다.
     **식당 정보**
     - 가게 이름: 누리카세
    - 문영 시간: 매일 12:00 ~ 22:00
     - 예약 가능 시간: 오후 12시, 오후 1시, 오후 2시, 오후 5시, 오후 6시, 오후 7시, 오후 8시, 오후 9시
     - 브레이크타임: 15:00 ~ 17:00
     - 최대 좌석 수: 8석
     - 메뉴: 런치 오마카세, 디너 오마카세
     - 주차: 근처 공영주차장 이용
     - 예약인원 최대: 8명까지 가능
     - 당일 예약 불가, 연중무휴 문영
     - 유아 동반 시 유아용 의자 필요 여부 확인 요청
     - 예약 절차: 예약 날짜 -> 예약 시간 -> 인원 -> 예약자 성함 -> 연락처(휴대폰 뒷번호 4자리) -> 예약 정보 확인
     **예약 정보**

    접수번호 (랜덤 4자리 숫자)

     - 접수 날짜 및 시간 (현재 날짜와 시간 자동 생성)
     - 예약 날짜, 예약 시간, 인원수, 예약자 성함, 연락처 뒷번호
     **예약 완료 후 절차**
     1. 사용자에게 예약 정보를 확인
     2. 맞다고 하면 "예약이 완료되었습니다."라고 안내합니다.
     3. 그 후 알러지 여부확인.
     4. 알러지가 있으면 예약 정보에 추가.
     5. 모든 절차가 끝나면 추가로 필요한 도움이 있는지 물어보고, 추가 요청이 없으면 "통화가 종료됩니다."
```

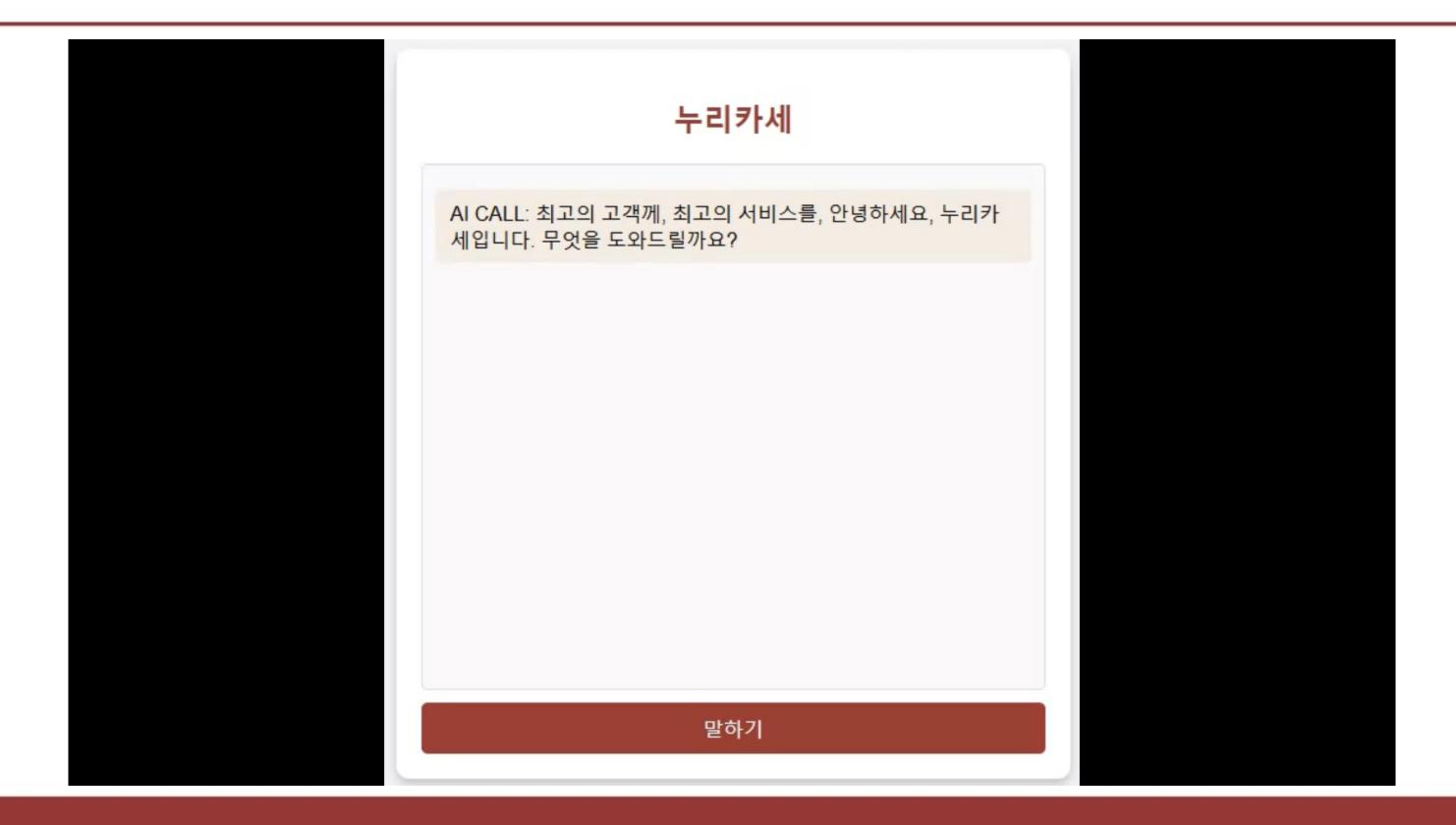
```
def get_chatbot_response(user_input):
   global messages
   # 접수 날짜와 시간 설정
   reception_date = datetime.now().strftime("2m월 %d일")
   reception_time = datetime.now().strftime("%H人 %M문")
   reception_number = str(random.randint(1000, 9999)) # 무작위 접수 번호 생성
   # 접수 정보를 OpenAI API에 전달하기 위해 messages에 추가
   messages.append({"role": "user", "content": user input})
   # OpenAI API 호출
   response = openai.ChatCompletion.create(
       model="gpt-3.5-turbo",
       messages=messages,
       max tokens=300, # 필요한 답변 길이에 맞춰 토큰 수 조정
       temperature=0.2
```

```
def speak(text):
   r = requests.post('https://typecast.ai/api/speak', headers=HEADERS, json={
        'text': text,
        'lang': 'auto',
        'actor_id': '661797923ed12f31b61c4b5f',
        'xapi hd': True,
        'model_version': 'latest'
   speak_url = r.json()['result']['speak_v2_url']
   for _ in range(60):
       r = requests.get(speak_url, headers=HEADERS)
       ret = r.json()['result']
        if ret['status'] == 'done':
            audio_data = requests.get(ret['audio_download_url']).content
            audio stream = io.BytesIO(audio data)
            sample_rate, audio = read(audio_stream)
            sd.play(audio, samplerate=sample rate)
            sd.wait()
            break
```

```
@app.route('/')
def home():
    start_message = "최고의 고객께, 최고의 서비스를. 안녕하세요, 누리카세입니다. 무엇을 도와드릴까요?"
    return render_template('index.html', start_message=start_message)
@app.route('/speak', methods=['POST'])
def speak text():
   text = request.json['text']
   speak(text)
   return jsonify({'status': 'completed'})
@app.route('/get_response', methods=['POST'])
def get_response():
    user input = request.json['user input']
    bot_response = get_chatbot_response(user_input)
   speak(bot_response)
    return jsonify({'user_input': user_input, 'bot_response': bot_response})
if name == " main ":
    app.run(debug=True)
```

04 최종결과

최종 결과



레퍼런스

[1] Pandya, K., & Holia, M. (2023). Automating Customer Service using LangChain: Building custom opensource GPT Chatbot for organizations.

[2] Zhang, L., & Chen, M. (2021). Advances in Speech-to-Text Technologies and their Applications in Real-Time Transcription. International Journal of Speech Processing, 15(4), 210–224.

[3] Kumar, R., & Lee, J. (2022). Exploring Neural Network-Based TTS for Multi-Language Applications. Speech Communication Journal, 34(6), 189-203.

[4] Patel, S., & Li, G. (2023). The Role of Memory and Context in Long-Term Conversational Al Systems Using LangChain. Journal of Human-Computer Interaction, 37(5), 123–138

타임 테이블

주제 구체화		
가상 식당 구성		
가상 식당 구성		
랭체인 코드 완성		
음성 텍스트 변환 구현		
예약 정보 저장 코드 구현		
		테스트 및 디버깅
		고도화 진행
1주차 10월 24일 - 10월 30일	2주차 10월 31일 - 11월 7일	3주차 11월 8일 - 11월 14일



Thank You Q&A