냉장고를 털어라!

Recommended menu service <뭐 먹지?>

프롬프트 엔지니어링
& LLM 튜닝 부트캠프

노상석, 송용화, 박설아, 서혜민

01 발표 목차

무슨 음식을 만들지 고민되신다고요?
당신의 냉장고를 보여주세요!
<뭐먹지?>가 알려주신 정보를 토대로
메뉴를 추천해 드릴게요.

02 <뭐 먹지?>
기획의도

"남은 식재료를 이용해 초보도 쉽게 요리할 수 있도록 레시피를 제공하면 어떨까?"

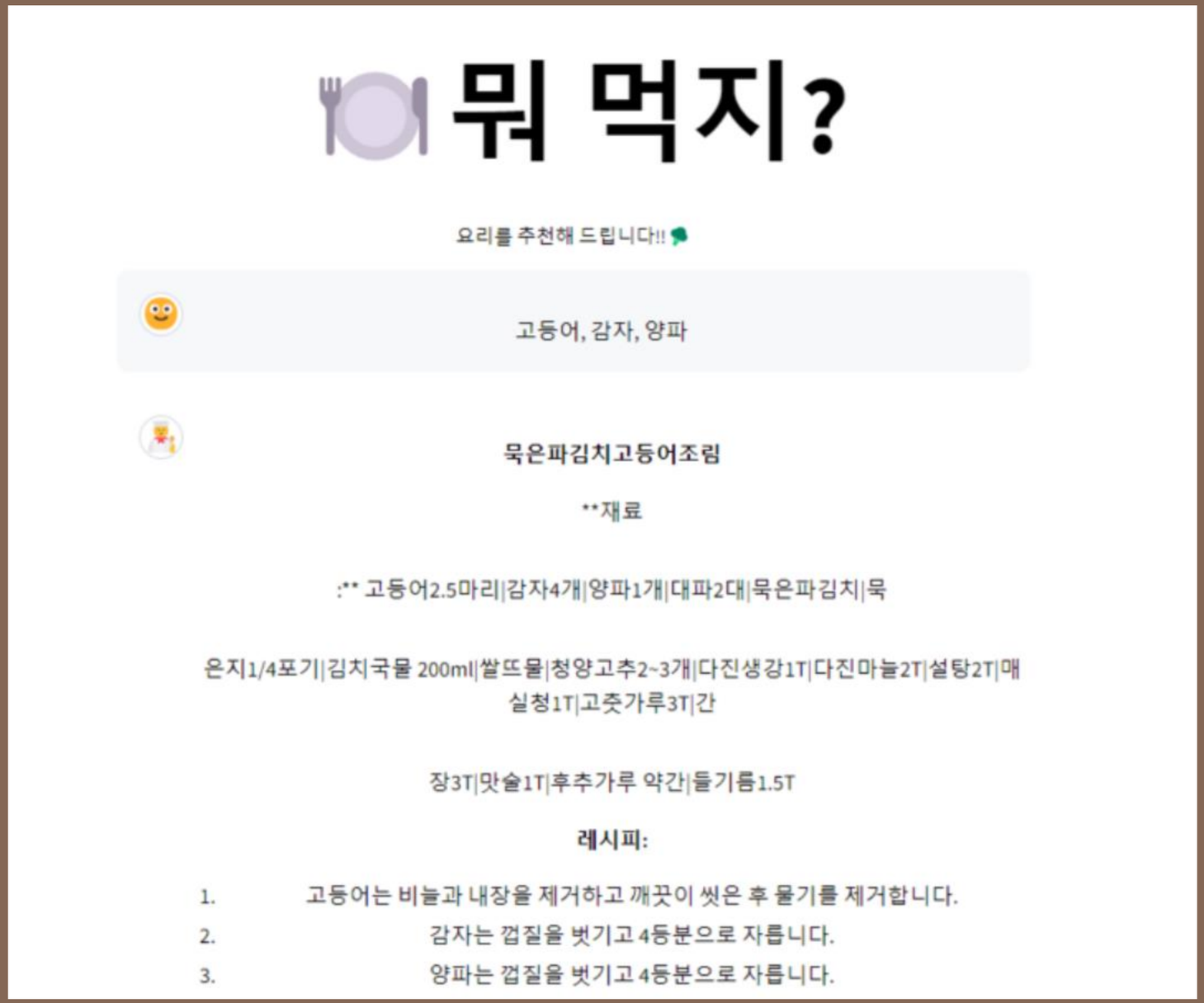
02 <뭐 먹지?>
기대효과

- 편의
- 건강한 식습관
- 환경
- 식재료 낭비↓
- 경제적

03 <뭐 먹지?>
서비스 안내


우선 냉장고를 여세요!
사용할만한 식재료를 입력하세요.
재료 2-4가지 입력을 권장합니다.
ai 챗봇이 메뉴를 추천해드립니다.
뭐 먹지?
요리를 추천해 드립니다!!
재료를 알려주세요!(예:양파, 피망, 감자, ...)

03 입력 결과
-예시 1)
메뉴와 재료



03 입력 결과 -예시 1) 레시피

3. 양파는 껍질을 벗기고 4등분으로 자릅니다.
4. 대파는 5cm 길이로 자릅니다.
5. 묵은파

김치는 굵게 다진 후 물기를 짜냅니다. 6. 묵은지는 얇게 썰어줍니다.

7. 청양고추는 씨를 제거하고 얇게 썰어줍니다.
8. 생강과 마늘은 다진다.
9. 중불로 가열한 팬에 들기름을 두르고 고등어를 넣어 양쪽면이 노릇노릇해질 때까지 굽습니다.
10. 고등어를 팬에서 꺼내 접시에 담아둡니다.
11. 같은 팬에 감자, 양파, 대파를 넣고 볶습니다.
12. 야채가 살짝 익으면 묵은파김치, 묵은지, 김치국물, 쌀뜨물, 청양고추, 생강, 마늘, 설탕, 매실청, 고춧가루, 간장, 맛술, 후추가루를 넣고 끓입니다.
13. 고등어를 다시 팬에 넣고 끓여줍니다.
14. 끓으면

불을 약하게 줄이고 10분간 끓입니다. 15. 접시에 담아 밥과 함께 제공합니다.

재료를 알려주세요!(예:양파, 피망, 감자, ...)

03 입력 결과
-예시 2)

사용자가 입력한 재료를 모두 포함한 음식이 없을 시,
*"음식이름과 재료를 표출할 수 없습니다."*로 반환됨
ex) 초콜릿, 고등어

03 <뭐 먹지?>
서비스 시연

04<뭐 먹지?>

1. 데이터 불러오기

```
file_path = 'Cooking.csv'
try:
    CK = pd.read_csv(file_path, encoding='utf-8')
except UnicodeDecodeError:
    CK = pd.read_csv(file_path, encoding='euc-kr')
```

04<뭐 먹지?>

2. 데이터 전처리

```
CK['NEW_COL'] = CK[['INQ_CNT', 'RCMM_CNT', 'SRAP_CNT']].sum(axis=1) / 3
selected_CK = CK.iloc[:, [2, 13]]
CK_cl = selected_CK.dropna()


final_data = CK_cl.drop_duplicates(subset='CKG_NM')
```

04<뭐 먹지?>

3. Cook 함수 적용

```
def find_recipe_with_ingredients(ingredients):
    def cook(selected_ingredients):
        recipes = []
        for _, row in final_data.iterrows():
            recipe_ingredients = row['CKG_MTRL_CN'].lower().split(', ')
            if all(all(ingredient in item for item in recipe_ingredients) for ingredient
in selected_ingredients):
                recipes.append(row['CKG_NM'])
        return recipes
    selected_ingredients = [ingredient.lower().strip() for ingredient in ingredients]
    possible_dishes = cook(selected_ingredients)
    if not possible_dishes:
        return "적합한 요리를 찾을 수 없습니다."
    selected_dish = random.choice(possible_dishes)
    selected_row = final_data[final_data['CKG_NM'] == selected_dish]
    dish_ingredients = selected_row['CKG_MTRL_CN'].values[0]
    return f" '{selected_dish}' 추천할게요. 이것들로 만들 수 있어요.: {dish_ingredients}"
```

04<뭐 먹지?>

3. Cook
함수 적용

```
selected_ingredients = [ingredient.lower().strip() for ingredient in ingredients]
possible_dishes = cook(selected_ingredients)
if not possible_dishes:
    return "적합한 요리를 찾을 수 없습니다."
selected_dish = random.choice(possible_dishes)
selected_row = final_data[final_data['CKG_NM'] == selected_dish]
dish_ingredients = selected_row['CKG_MTRL_CN'].values[0]
return f" '{selected_dish}' 추천할게요. 이것들로 만들 수 있어요.: {dish_ingredients}"
```

04<뭐 먹지?>

3. Cook 함수 적용

```
def find_recipe_with_ingredients(ingredients):
    def cook(selected_ingredients):
        recipes = []
        for _, row in final_data.iterrows():
            recipe_ingredients = row['CKG_MTRL_CN'].lower().split(', ')
            if all(all(ingredient in item for item in recipe_ingredients) for ingredient
in selected_ingredients):
                recipes.append(row['CKG_NM'])
        return recipes
    selected_ingredients = [ingredient.lower().strip() for ingredient in ingredients]
    possible_dishes = cook(selected_ingredients)
    if not possible_dishes:
        return "적합한 요리를 찾을 수 없습니다."
    selected_dish = random.choice(possible_dishes)
    selected_row = final_data[final_data['CKG_NM'] == selected_dish]
    dish_ingredients = selected_row['CKG_MTRL_CN'].values[0]
    return f" '{selected_dish}' 추천할게요. 이것들로 만들 수 있어요.: {dish_ingredients}"
```

04<뭐 먹지?>

4. ui 코드작성

```python
import streamlit as st
import random
import google.generativeai as genai
import pandas as pd
from tools import find_recipe_with_ingredients

with open('style.css') as f:
    st.markdown(f'<style>{f.read()}</style>', unsafe_allow_html=True)

st.title(" 🍽️뭐 먹지? ")
st.caption("요리를 추천해 드립니다!!🥬")

# Google API key
if "api_key" not in st.session_state:
    try:
        st.session_state.api_key = st.secrets["GOOGLE_API_KEY"]
    except:
        st.session_state.api_key = ""
        st.write("Your Google API Key is not provided in `.streamlit/secrets.toml`, but you can input one in the sidebar for temporary use.")

# Initialize chat history
if "messages" not in st.session_state:
    st.session_state.messages = []

# Sidebar for parameters
with st.sidebar:
    # Google API Key
    if not st.session_state.api_key:
        st.header("Google API Key")
        st.session_state.api_key = st.text_input("Google API Key", type="password")
    else:
        genai.configure(api_key=st.session_state.api_key)



        # ChatCompletion parameters
    model_name='gemini-pro'

    generation_config = {
    "temperature": 0.5,
    "max_output_tokens": 2048,
    "top_k": 10,
    "top_p": 0.35,
    }
```

04<뭐 먹지?>

4. ui 코드작성

```python
    }


    # 이미지 추가
    st.image('/workspaces/gemini/.imagie/.imagie/야채사진.png', caption='')

    # 사이드바 꾸미기
    st.markdown("# 🥕우선 냉장고를 여세요!🥕")
    st.write("사용할만한 식재료를 입력하세요.")
    st.write("재료 2~4가지 입력을 권장합니다.")
    st.write("ai 챗봇이 메뉴를 추천해드립니다.")

# Display messages in history
for msg in st.session_state.messages:
    if parts := msg.parts:
        with st.chat_message('human' if msg.role == 'user' else 'ai'):
            for p in parts:
                st.write(p.text)

# Chat input

if prompt := st.chat_input("재료를 알려주세요!(예:양파, 피망, 감자, ...)"):
    with st.chat_message('human', avatar='😊'):
        st.write(prompt)

    # 입력된 재료를 분리하고 처리
    user_ingredients = prompt.split(", ")

    recommended_recipe = find_recipe_with_ingredients(user_ingredients)

    # 요리 추천을 위한 새로운 프롬프트 생성
    recipe_request = f"음식 이름과 재료를 표출해 주고 음식 레시피를 추천해주고 음식 재료를 표출 할 때에는 요리 재료들을 줄바꿈 없이 한줄에

    # AI 모델 설정 및 초기화 (예: Google AI 모델)
    model = genai.GenerativeModel(model_name=model_name, generation_config=generation_config)

    # AI 모델을 사용하여 요리 추천 응답 생성
    response = model.generate_content(recipe_request, stream=True)

    # AI 모델의 응답을 채팅 스타일로 표시
    with st.chat_message("ai", avatar='👨‍🍳'):
        for chunk in response:
            st.write(chunk.text)

    # 채팅 히스토리 삭제
st.session_state.messages = []
```

05 <뭐 먹지?>

비즈니스 모델링(BM)

- **주요 타겟층 설정**
- 수익성 창출 방법

Menu
Recommen
-dation
Service

a. 요리 초보자

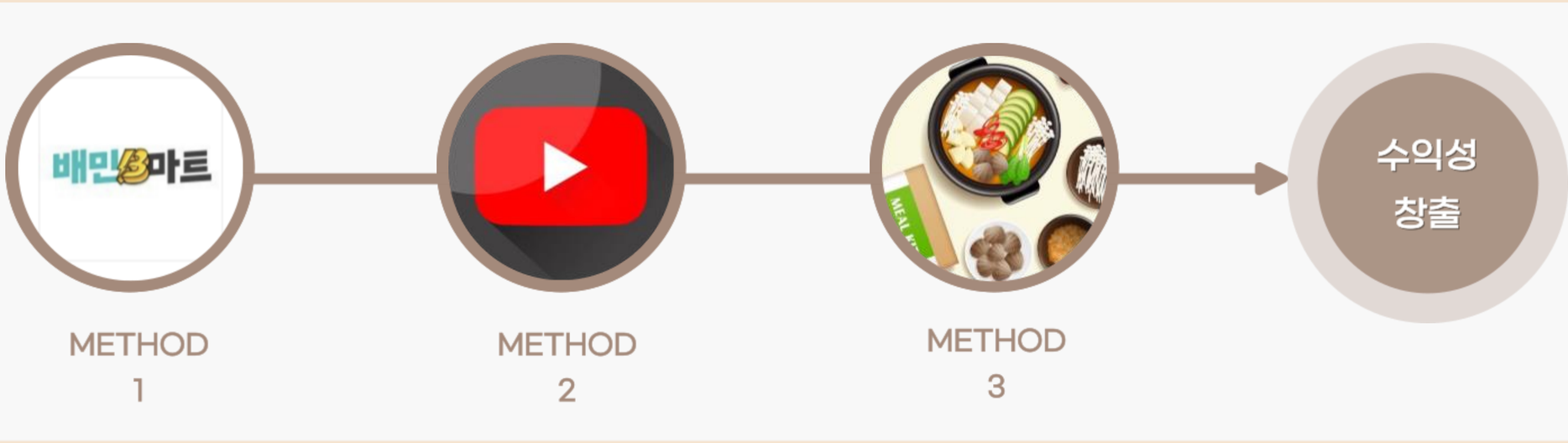
냉장고에 있는 재료를 활용하여 간단하고 빠르게 요리할 수 있는 초보자를 대상으로 할 수 있습니다. 식사를 준비하는 데 있어 부담 없이 사용할 수 있는 기능과 레시피를 제공합니다.

b. 음식 경험을 즐기는 사람들

다양한 레시피와 신기한 음식 조합을 제안하여 음식 경험을 즐기는 사용자를 대상으로 할 수 있습니다. 특이한 조합이나 새로운 요리 아이디어를 제공하여 흥미를 유발합니다.

05 <뭐 먹지?>
비즈니스 모델링(BM)

비즈니스 모델링 (BM)



진 행 내 용

METHOD 1	METHOD 2	METHOD 3	
온라인 상거래 앱과 제휴를 통한 협력, 내부 서비스로 기능함 ('배달의 민족-B마트')	서비스 내 자연스러운 **유튜브 링크 제공** -> 유튜버와의 확장적 협업	**밀키트 제공업체와의 협력** -> 밀키트 사업과의 연결을 통한 확장	추가 수익기회를 제공 -> 수익성 증가 -> 서비스 지속가능