개발된 LLM 연동 웹 어플리케이션

1. 웹 어플리케이션 개요

- **목표** : Allmrec 모델과 API를 활용하여 사용자의 요구 사항을 대화형으로 받아 Beauty 관련 상품을 추천하는 웹 애플리케이션 개발.
- **주요 기능** : 개발한 모델을 활용한 상품 추천, 사용자 요구 사항 수집 및 분석.
- **기술 스택**:
  - **프론트엔드**: HTML, CSS, JavaScript
  - **백엔드**: Python, Django,gunicorn,nginx,aws(fc2, Route 53, Certificate Manager),fastapi,uvicorn
  - **모델 및 API**: Hugging Face, OpenAI, GPT
  - **데이터베이스**: MySQL

2. 설치 및 초기 설정 (Installation and Initial Setup)(윈도우 환경)

1. 장고의 프로젝트 폴더에서 myproject/setting.py 내부 MODEL_SERVER_URL 부분을 배포되어있는 모델 URL로 교체

```
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

LOGIN_REDIRECT_URL = '/'

MODEL_SERVER_URL = "https://223e-34-125-34-228.ngrok-free.app/"
```

2. 콘솔에서 가상환경을 만든 후 경로를 프로젝트 경로로 바꾼 후 pip install r requirements txt

```
(base) C:₩>conda create --name skn02_4team python=3.11
Channels:
 - conda-forge
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): /
```

```
(skn02_4team) C:₩ex₩myproject>pip install -r requirements.txt
Collecting torch (from -r requirements.txt (line 1))
  Downloading torch-2.5.1-cp311-cp311-win_amd64.whl.metadata (28 kB)
Collecting tqdm (from -r requirements.txt (line 2))
  Downloading tqdm-4.67.0-py3-none-any.whl.metadata (57 kB)
Collecting pytz (from -r requirements.txt (line 3))
  Using cached pytz-2024.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting numpy (from -r requirements.txt (line 4))
```

3. db ssl 인증서 연결을 위하여
   C:\Program Files\SSL 에
   https://docs.aws.amazon.com/ko_kr/AmazonRDS/latest/UserGuide/UsingWithRDS.SSL.html
   에서 다운로드 받을 수 있는 us-east-1-bundle.pem 파일 연결
4. python manage.py runserver 명령어를 활용 장고 실행 후 경로 접속

```
(skn02_4team) C:₩ex₩myproject>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
November 11, 2024 - 15:48:41
Django version 5.1.1, using settings 'myproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

3. 기본 사용법 (Basic Usage)

- 설치 했으면 어떻게 사용해야 하는 지 설명

4. 확장 및 커스터마이징 (Extension and Customization)

- 프롬프트 수정 가능한 부분
  - 프롬프트 수정 및 커스터마이징 A-LLMRec-20241107 2nd for colab for api/chat_bot.py

```
system_template = SystemMessagePromptTemplate.from_template(
    """
    사용자 요청: {input_text}
        beauty 관련 상품 추천 챗봇 시스템의 입력을 담당하고 있습니다.
        경우에 따라 대답 방식이 다릅니다.
        사용자의 요청을 분석하여 Beauty 관련 상품 추천에 관한 내용인지 판별합니다.
        Beauty 상품 추천에 관한 내용이면 다음 모델에게 전달해주는 역할입니다.
        예시 :"손이 건조한데 좋은 핸드크림있을까?"
        아닌 경우에는 사용자에게 직접 응답을 하는 역할입니다.

        사용자의 요청이 Beauty 상품 추천에 관한 내용인 경우의 응답 방식:
        - 영어로 응답합니다.
        - 사용자의 요청을 요약하여 다음 머신에게 추천이 필요한 단어만 전달해주는 역할로 사용자는 보지못합니다.
        - response 내용만 응답합니다.
        - 'recommendation' 이란 단어는 필요하지않습니다.
        - 말끝에 "메롱"을 한국어로 붙입니다.
        - 최대한 간결하게 응답
        - 예시"skincare : Facial cleansers, moisturizers, serums 메롱" ,"footcream for dry skin 메롱"

        **다른 카테고리 상품 추천 요청일 경우:**
        - "Beauty 관련 상품 추천 시스템입니다. 다시 질문해주세요"라고 한국어로 응답합니다.

        **Beauty 상품 추천과 관계없는 요청일 경우:**
        - 한국어로 응답합니다.
        - 간단한 일상 대화나 인사는 받아줍니다.
        - 이순신 관련 질문,음식에 대한 평가 등 Beauty관련 상품과 무관한 경우에는 "부적절한 질문입니다. 다시 질문해주세요"라고 응답합니다.

    맥락: {chat_history}
    """
    )
```

```
human_template = HumanMessagePromptTemplate.from_template(
    "사용자 요청: {input_text}\n\n"
    "요청 분석: 요청이 Beauty 제품 추천과 관련된지 판단합니다.\n"
    "Beauty 추천이 아닌 경우 챗봇으로서 직접 적절하게 응답합니다:\n"
    "- Beauty 제품 추천 이외의 간단한 일상 대화는 응대합니다.\n"
    "- Beauty와 무관한 주제(예: 이순신 관련 질문)에는 '부적절한 질문입니다. 다시 질문해주세요'라고 응답합니다.\n"
    "- 다른 카테고리의 상품 추천은 '뷰티관련 상품 추천 시스템입니다. 다시 질문해주세요'라고 응답합니다."
)
```

```
output_prompt_template = """
        한국어로 설명합니다.

        추천 결과: {input_text}

        model's recommendation 에서 사용자의 요구에 부합하는 상품들을 추천해줍니다.
        만약 수가 부족하다면 candidate 50에서 추가로 추천해줍니다.
        총 추천 개수는 반드시 5개로 유지합니다.
        상품명은 번역하지 않습니다.
        사용자의 요구에 적절한 추천 상품이 없다면 "추천할 상품이 없습니다."라고 응답합니다.

        출력 방식은 다음과 같습니다:


        **JSON 형식 응답** (개발자에게 전달될 출력):

        예시:
        (output:
                ["요청에 따른 추천 목록입니다.

                    1. 추천 상품명 : 상품에 대한 설명
                    2. 추천 상품명 : 상품에 대한 설명
                    3. 추천 상품명 : 상품에 대한 설명
                    4. 추천 상품명 : 상품에 대한 설명
                    5. 추천 상품명 : 상품에 대한 설명

                위의 상품 중 마음에 드는 상품을 골라보세요."]

                "products": [
                    "상품명1",
                    "상품명2",
                    "상품명3",
                    "상품명4",
                    "상품명5"
                ]
        )

        다른 미사여구는 붙이지 않고 json 값만 응답합니다.
        """
```

- LLM 모델 변환 (모델 재학습 필요)
    - allmrec 부분의 llm 변환은 A-LLMRec-20241107 2nd for colab for api/models/llm4rec.py 부분

```python
class llm4rec(nn.Module):
    def __init__(
        self,
        device,
        llm_model="",
        max_output_txt_len=1024,
    ):
        super().__init__()
        self.device = device

        if llm_model == 'opt':
            self.llm_model = OPTForCausalLM.from_pretrained("facebook/opt-6.7b", torch_dtype=torch.float16, load_in_8bit=False, device_map=self.device)
            self.llm_tokenizer = AutoTokenizer.from_pretrained("facebook/opt-6.7b", use_fast=False)
            # self.llm_model = OPTForCausalLM.from_pretrained("facebook/opt-6.7b", torch_dtype=torch.float16, device_map=self.device)
        else:
            raise Exception(f'{llm_model} is not supported')

        self.llm_tokenizer.add_special_tokens({'pad_token': '[PAD]'})
        self.llm_tokenizer.add_special_tokens({'bos_token': '</s>'})
        self.llm_tokenizer.add_special_tokens({'eos_token': '</s>'})
        self.llm_tokenizer.add_special_tokens({'unk_token': '</s>'})
        self.llm_tokenizer.add_special_tokens({'additional_special_tokens': ['[UserRep]','[HistoryEmb]','[CandidateEmb]']})

        self.llm_model.resize_token_embeddings(len(self.llm_tokenizer))

        for _, param in self.llm_model.named_parameters():
            param.requires_grad = False

        self.max_output_txt_len = max_output_txt_len
```

- A-LLMRec-20241107 2nd for colab for api/main.py 부분

```python
# model setting
parser.add_argument("--llm", type=str, default='opt', help='flan_t5, opt, vicuna')
parser.add_argument("--recsys", type=str, default='sasrec')
```

5. 결론 (Conclusion)

- **프로젝트 성과 요약**: 이번 프로젝트를 통해 LLM 연동 웹 어플리케이션의 성공적인 구현 및 이를 통한 사용자 맞춤형 추천 서비스 제공 가능성을 확인함.
- **프로젝트의 기여**: 본 애플리케이션은 LLM과 API의 효율적인 연동을 통해 사용자 경험을 개선하고, Beauty 관련 상품 추천을 웹페이지에서 llm을 활용해 대화형으로 서비스 가능
- **향후 발전 방향**: 성능 향상을 위해 LLM 업그레이드 및 데이터 전처리 개선 필요. 지속적인 피드백을 바탕으로 기능 고도화 가능.
- **한계 및 개선 방안**: 현재 메모리 사용 제한 및 학습 시간 문제 해결을 위해 더 나은 하드웨어 사용과 모델 최적화 검토 필요
