

웹 챗봇 제작 실습

Cursor AI
단계별 실습

Web 챗봇 개발 환경 준비

1. Web_Chatbot_Project 폴더를 만들고 Cursor에서 폴더를 연다

2. Cursor Terminal창에서 아래 명령을 실행해서 Flask 모듈을 설치한다

pip install flask

3. 앞 프로젝트에서 OpenAI API 키를 환경변수로 설정해 놓은 .env 파일을 Web_Chatbot_Project 폴더안에 복사해 놓는다



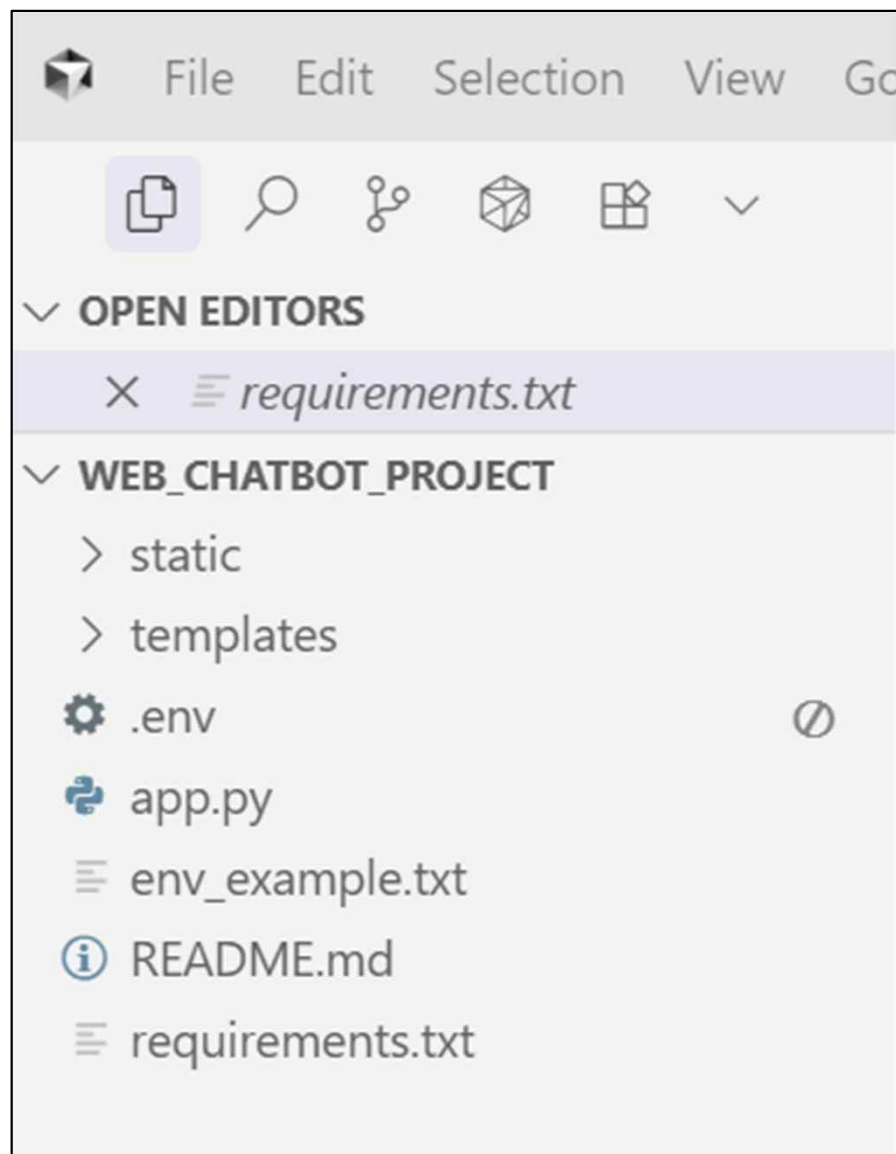
Web Chatbot 만들기 질의문

Flask를 사용해서 간단한 웹 챗봇 서버를 만들어줘.

조건은 다음과 같아 :

1. 현재 경로에서 채팅 UI(입력창, 버튼, 채팅창 영역)를 보여준다.
2. 사용자가 메시지를 입력하면 Flask 서버가 OpenAI API에 전달하고, 모델의 응답을 받아 채팅창에 표시한다.
3. 비동기(Fetch API) 요청을 사용해서 새로고침 없이 채팅이 가능해야 한다.
4. CSS로 디자인을 개선해서 사용자 메시지는 오른쪽 파란색 말풍선, 챗봇 메시지는 왼쪽 회색 말풍선으로 보여준다.
5. 실행에 필요한 requirements.txt 파일도 함께 만들어줘."

소스 구조



Flask를 사용해서 간단한 웹 챗봇 서버를 만들어줘.
조건은 다음과 같아 :

1. 현재 경로에서 채팅 UI(입력창, 버튼, 채팅창 영역)
를 보여준다

✓ 모든 요구사항이 완료되었습니다:

1. Flask 서버: 현재 경로에서 채팅 UI를 제공하는 웹 서버
2. OpenAI API 연동: 최신 스타일 코드로 GPT-4o-mini 모델 사용
3. 환경변수 설정: .env 파일에서 API 키 읽기 (예시 파일 제공)
4. 채팅 기능: 사용자 메시지를 OpenAI API에 전달하고 응답 표시
5. 메시지 라벨링: 'You:'와 'Bot:' 라벨 표시
6. 비동기 요청: Fetch API로 새로고침 없는 채팅
7. 말풍선 디자인: 사용자(파란색, 오른쪽), 봇(회색, 왼쪽)
8. requirements.txt: 실행에 필요한 패키지 목록

app.py 소스

app.py ×

▶ ▾ □ ...

app.py > chat

```
1  from flask import Flask, render_template, request, jsonify
2  import openai
3  import os
4  from dotenv import load_dotenv
5
6  # .env 파일에서 환경변수 로드
7  load_dotenv()
8
9  app = Flask(__name__)
10
11 # OpenAI API 키 설정
12 openai.api_key = os.getenv('OPENAI_API_KEY')
13
14 @app.route('/')
15 def index():
16     """메인 페이지 - 채팅 UI 렌더링"""
17     return render_template('index.html')
18
19 @app.route('/chat', methods=['POST'])
20 def chat():
21     """채팅 메시지 처리"""
22     try:
23         data = request.get_json()
24         user_message = data.get('message', '')
25
26         if not user_message:
27             return jsonify({'error': '메시지가 비어있습니다.'}), 400
28
```

app.py 소스를 Cursor Editor에서 열어놓고 실행 한다.
Terminal 창에서 출력된 주소에 마우스를 위치시키면
나타나는 창에 “Follow link”를 클릭하면 웹 브라우저가
열린다

```
15 def index():  
16     """메인 페이지 - 채팅 UI 렌더링"""
```

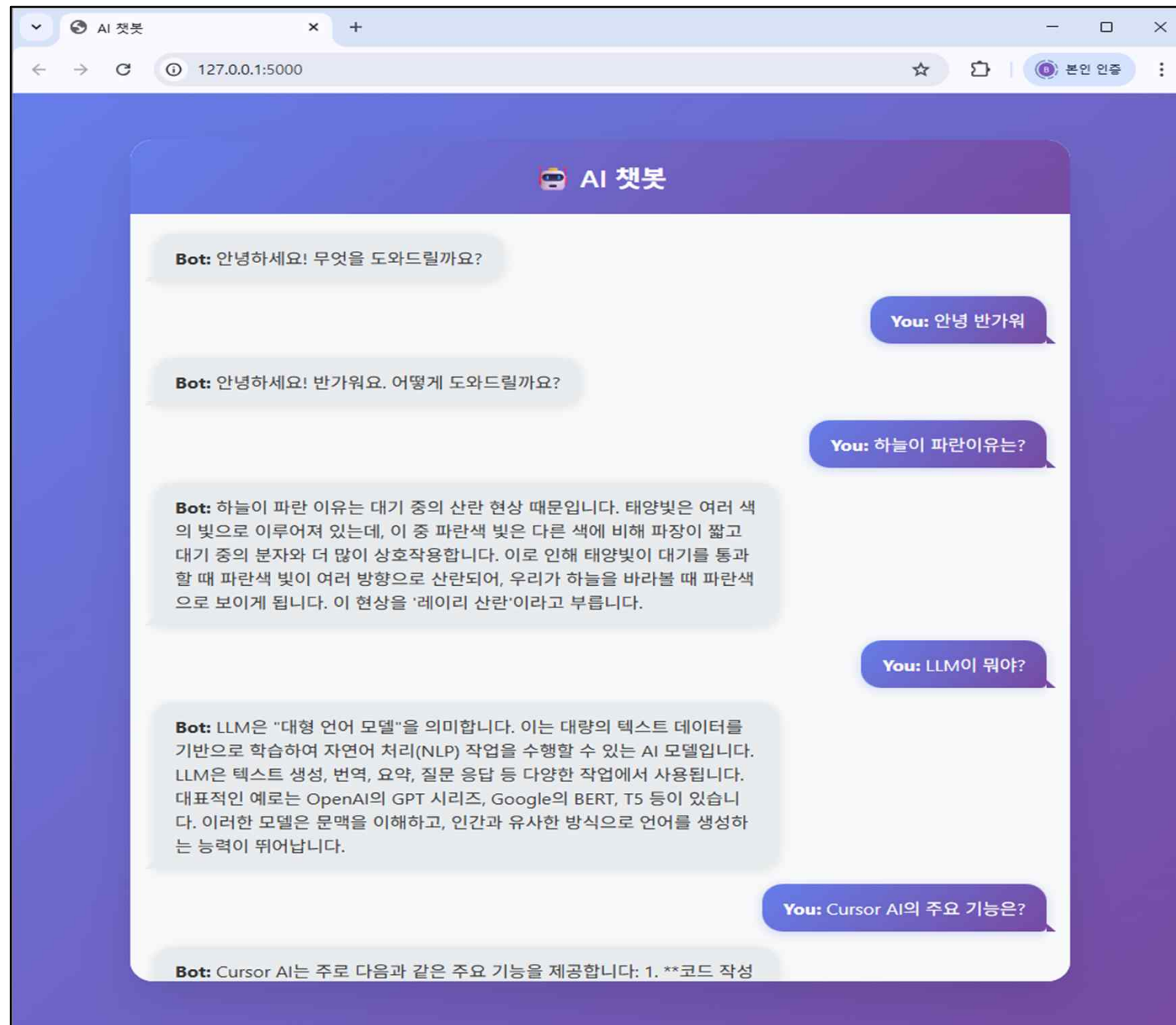
Problems Output Debug Console Terminal Ports Python + -

WARNING: This is a development server. Do not use it in a production deployment. Use a production web server instead.

* Running on [Follow link \(ctrl + click\)](http://127.0.0.1:5000) (127.0.0.1:5000)
* Running on <http://127.0.0.1:5000>
* Running on <http://10.223.247.3:5000>
* Debugger is active!
* Debugger PIN: 627-103-236

127.0.0.1 - - [05/Sep/2025 17:22:37] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [05/Sep/2025 17:22:37] "GET /static/script.js HTTP/1.1" 304 -
127.0.0.1 - - [05/Sep/2025 17:22:37] "GET /static/style.css HTTP/1.1" 304 -
127.0.0.1 - - [05/Sep/2025 17:22:51] "POST /chat HTTP/1.1" 200 -
127.0.0.1 - - [05/Sep/2025 17:23:07] "POST /chat HTTP/1.1" 200 -

실행 화면



기능 개선 (1)

대화내용이 많아지면 하단 입력 창이 보이질 않게 되니
하단 입력창은 고정시키고 메시지 화면을 스크롤 가능하게
수정해줘.

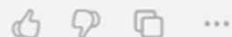
- 모든 화면 크기에서 일관된 사용자 경험을 제공합니다

◆◆ 사용자 경험 개선

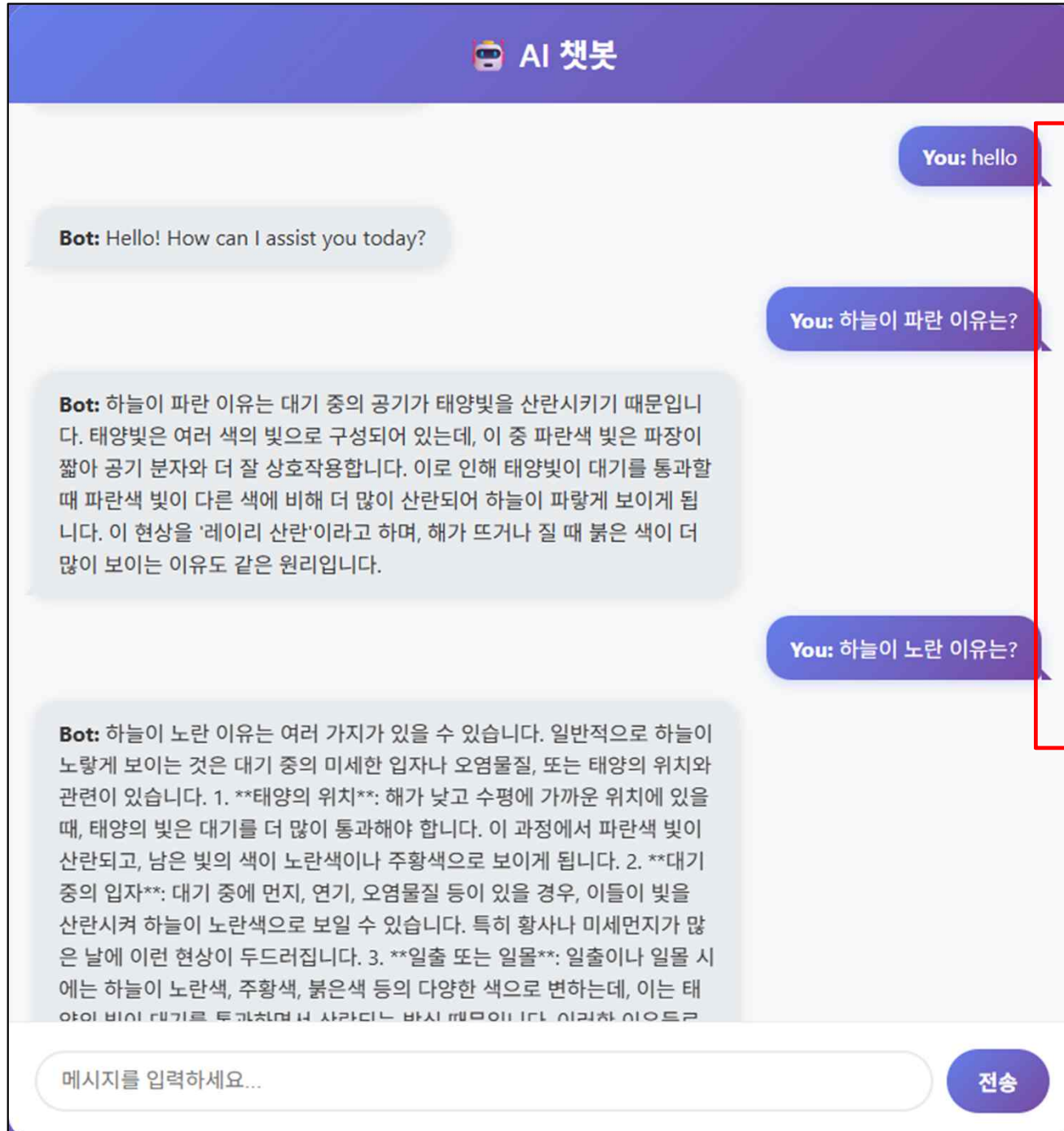
- **대화가 길어져도:** 입력창이 항상 보이므로 끊김 없는 대화 가능
- **과거 메시지 확인:** 위로 스크롤해서 이전 대화를 확인할 수 있음
- **새 메시지 알림:** 새 메시지를 보내면 자동으로 최신 메시지로 이동

이제 대화 내용이 아무리 많아져도 입력창이 항상 하단에 고정되어 있어 편리하게 사용할 수 있습니다!

☰ Review Changes



스크롤바가 사용되는 실행 화면



스크롤바

기능 개선 (2)

카카오톡 채팅방과 똑같은 디자인으로 만들어줘.

카카오톡 채팅방과 똑같은 디자인으로만들어줘.

◆◆ 카카오톡 스타일 디자인 완성!

◆ 주요 특징들

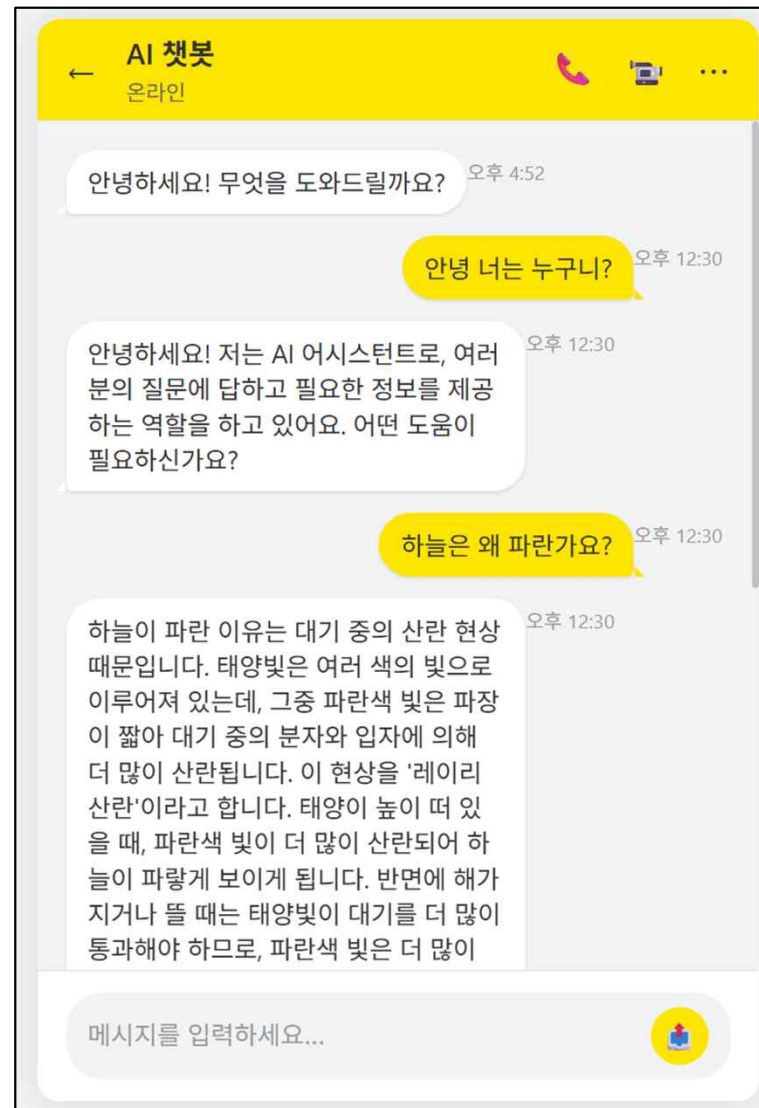
1. 카카오톡 헤더

- 노란색 배경 (#fee500)
- 뒤로가기 버튼 (←)
- 채팅방 이름과 상태 표시
- 통화, 영상통화, 더보기 버튼

2. 카카오톡 말풍선

- 사용자: 노란색 말풍선 (오른쪽)
- 봇: 흰색 말풍선 (왼쪽)
- 둥근 모서리와 꼬리 부분
- 메시지 시간 표시 (오전/오후 형식)

3. 카카오톡 입력창



Thank You!!

