

Chapter 12 바이브 코딩(Vibe Coding)

01 커서(Cursor) 설치

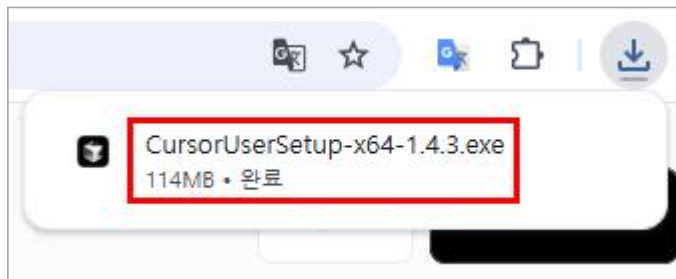
1. 커서 다운로드

■ 참고 사이트

- <https://cursor.com>



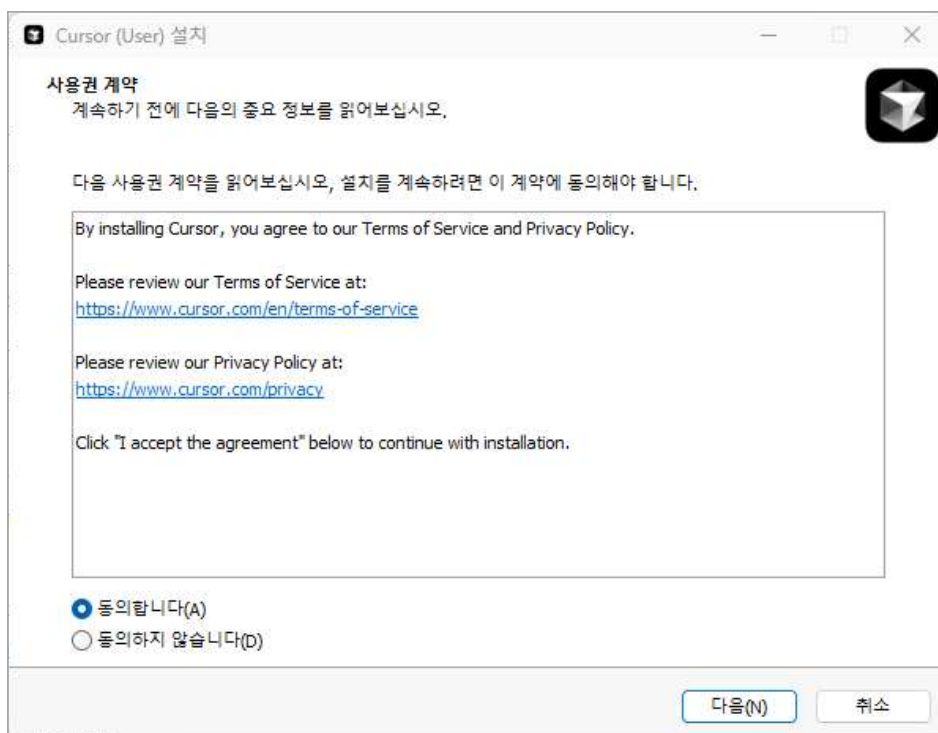
- Download for Windows 클릭



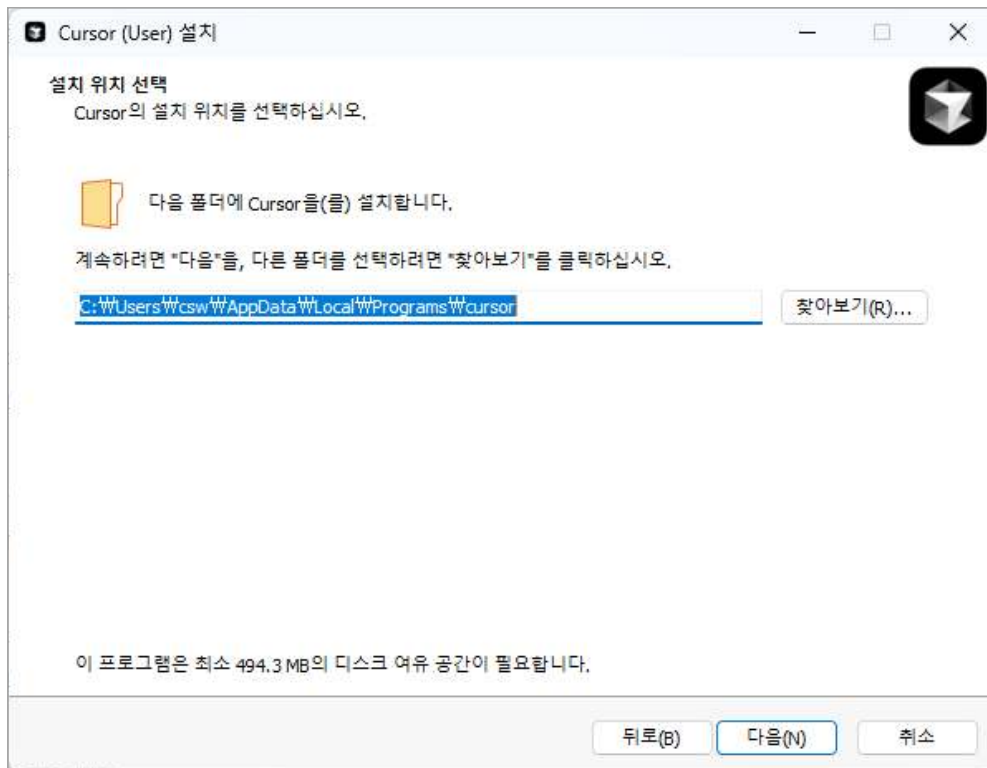
- 설치 파일 다운로드 완료

2. 커서 설치

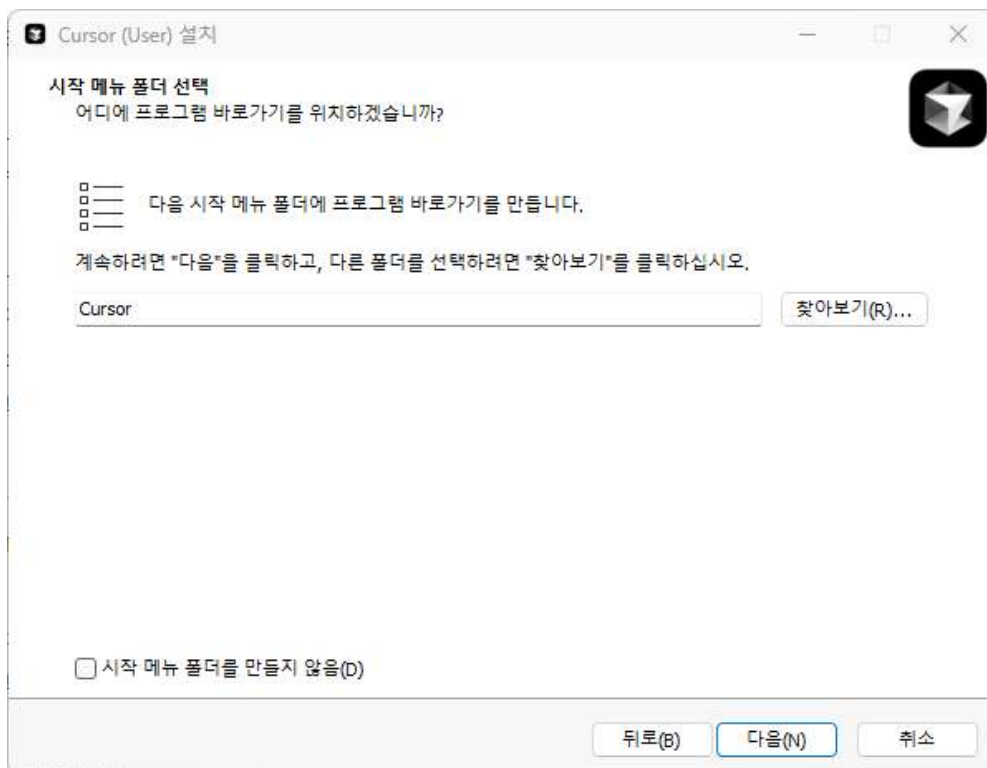
■ 사용권 계약 동의



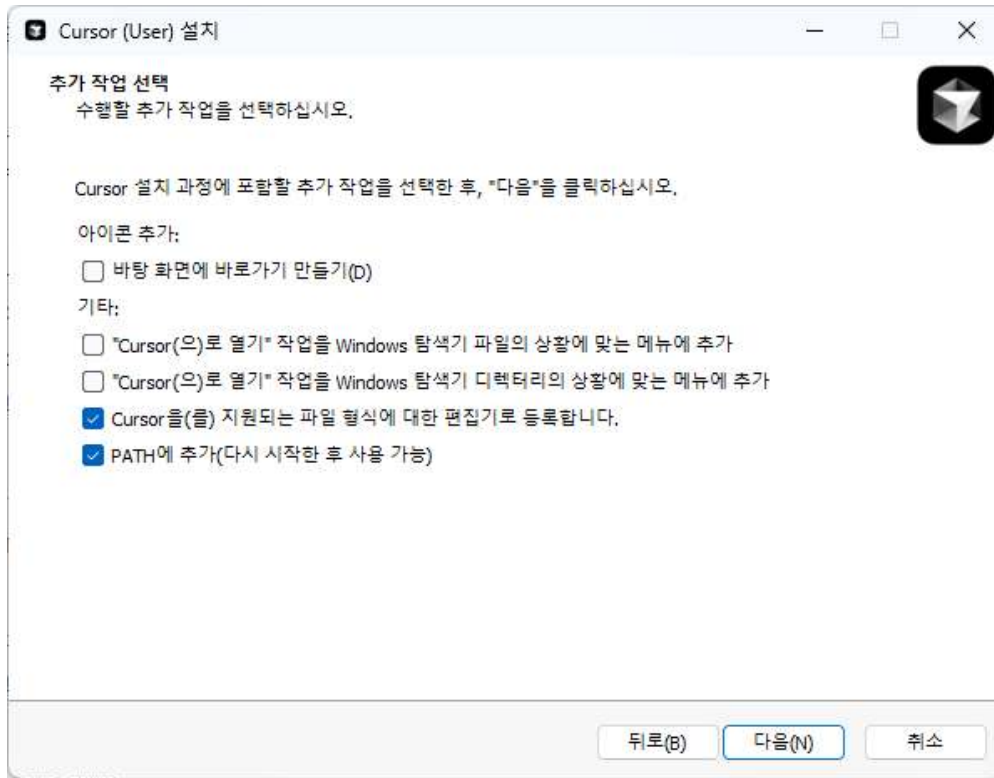
- 설치 위치
 - 기본 설정 유지



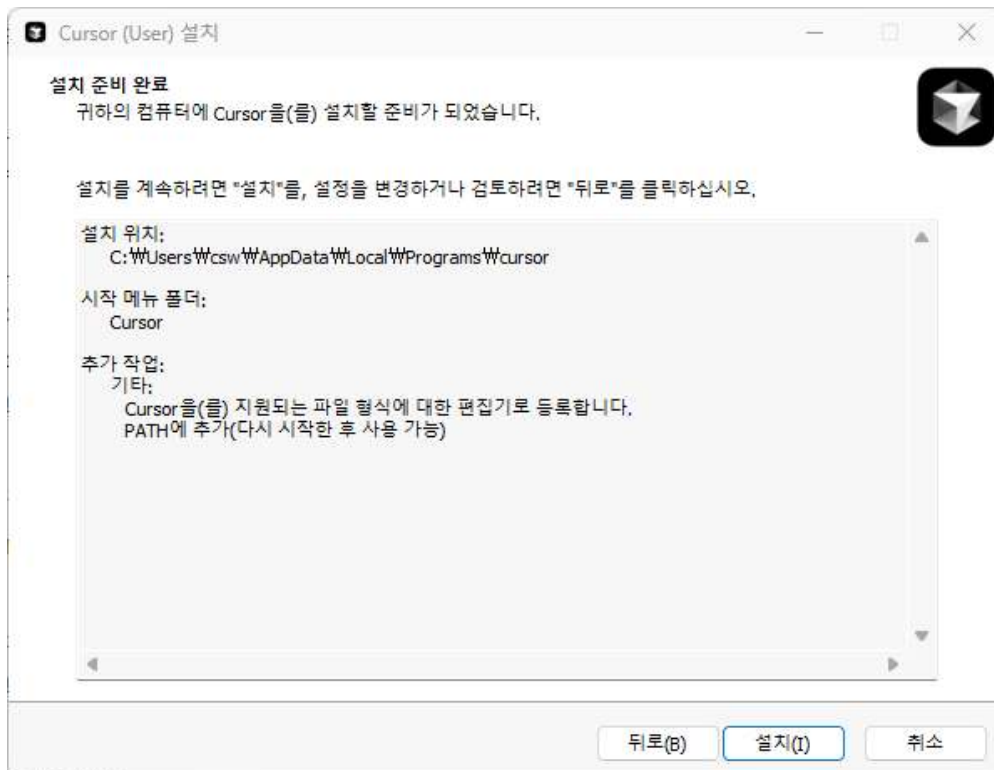
- 시작 메뉴 폴더 선택
 - 기본 설정 유지



- 추가 작업 선택
 - 기본 설정 유지

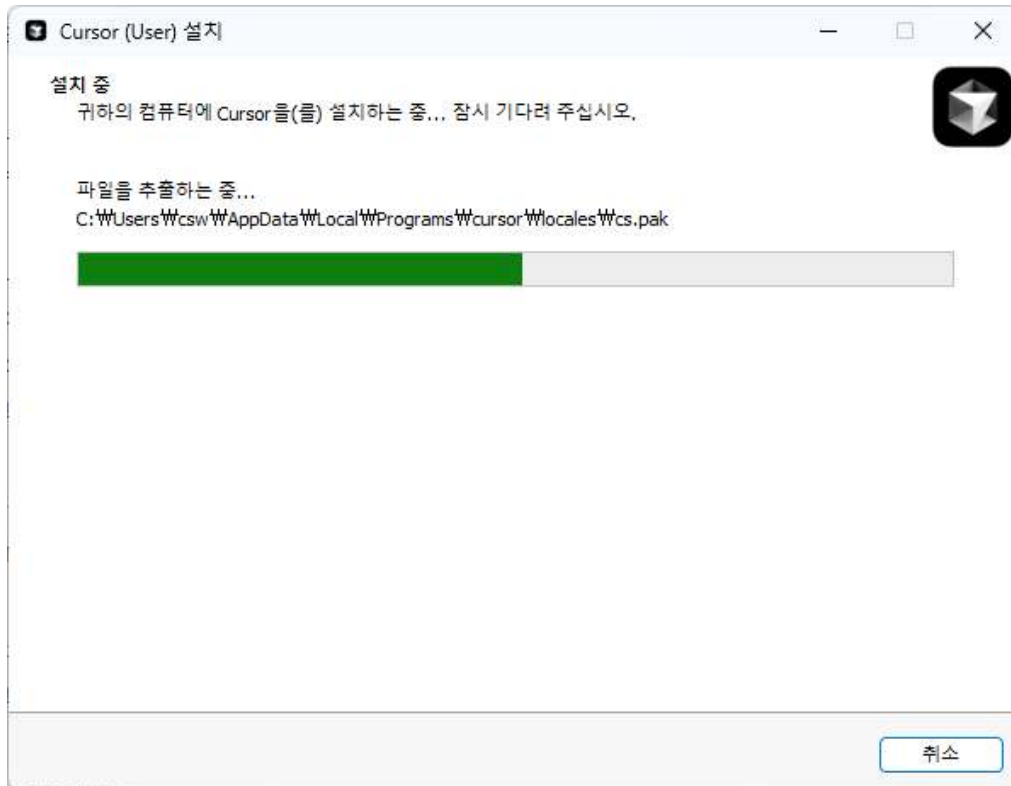


- 설치 준비 완료
- 설치 클릭

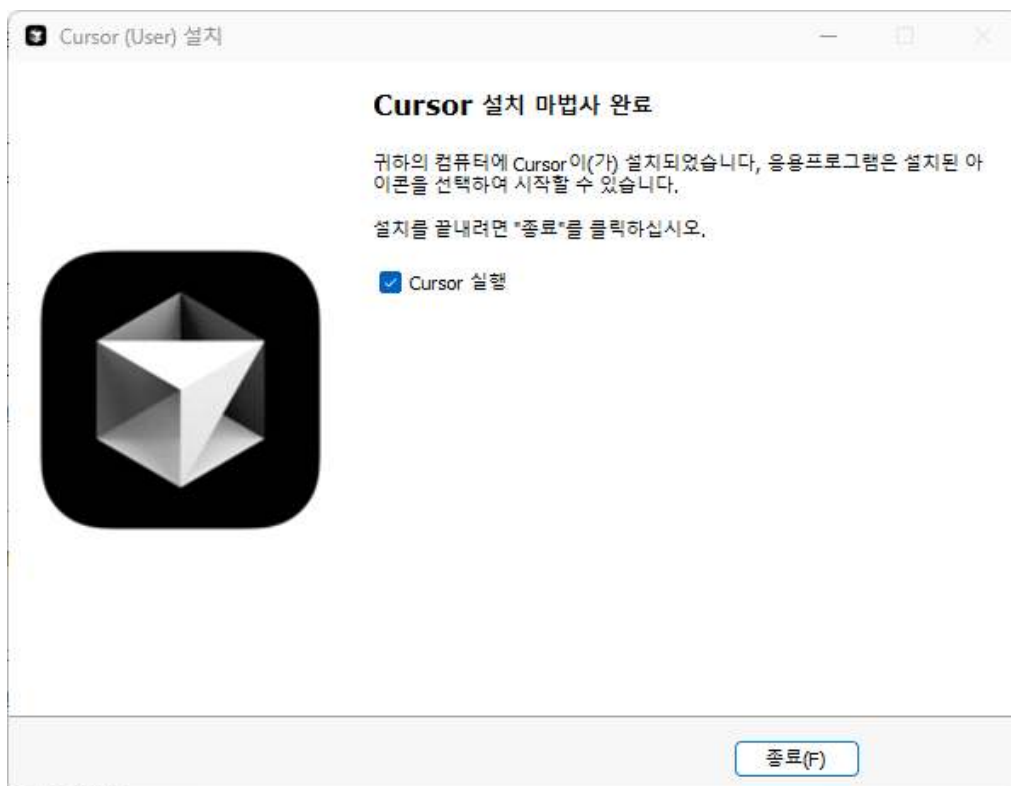


■ 설치 진행

- 설치 중

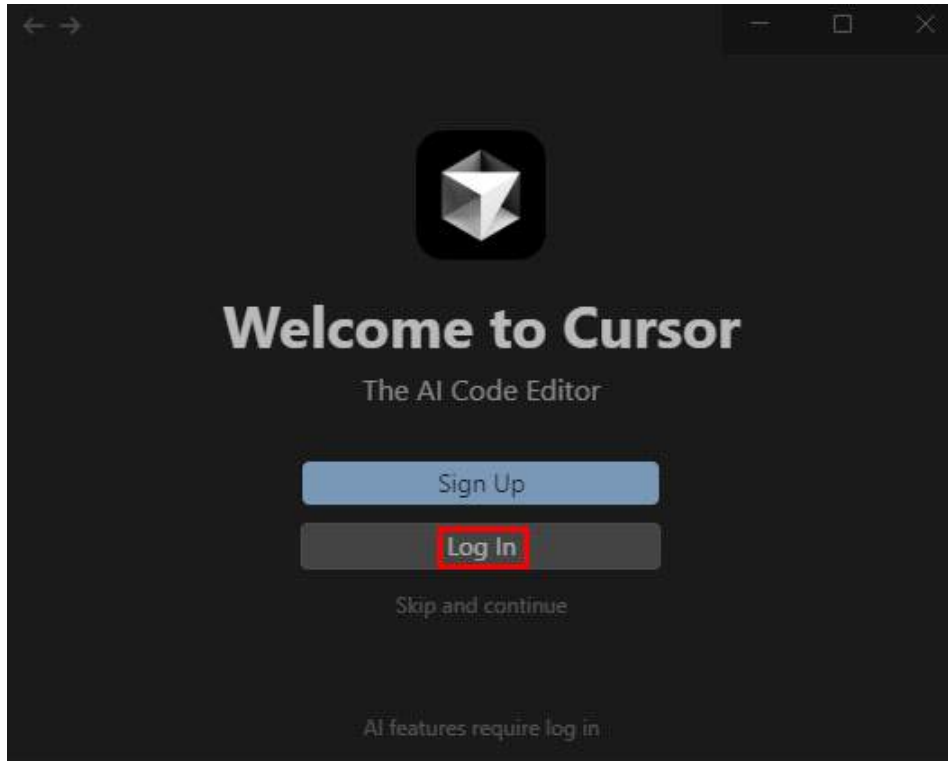


- 설치 완료

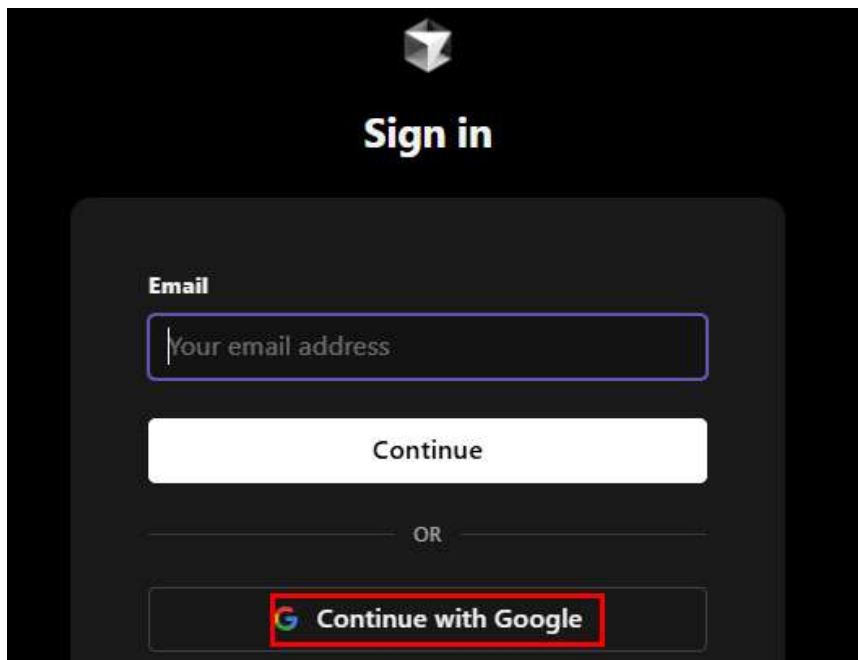


3. 커서 로그인

- 웰컴 투 커서 화면
 - 회원가입, 로그인, 생략 및 계속 중에서 하나 선택
 - 로그인 선택



- 로그인 계정 설정
 - 구글 로그인 선택



- 구글 이메일 입력

이메일 또는 휴대전화

wintari@naver.com

이메일을 잊으셨나요?

앱을 사용하기 전에 cursor.sh의 개인정보처리방침 및 서비스 약관을 검토하세요.

계정 만들기 다음

- 비밀번호 입력

비밀번호 입력

.....

☐ 비밀번호 표시

앱을 사용하기 전에 cursor.sh의 개인정보처리방침 및 서비스 약관을 검토하세요.

다른 방법 시도 다음

- 구글 아이디로 커서에 로그인 선택

Log in to Cursor desktop?

You're currently logged in as:

wintari@naver.com

Please only log in if you just came from the Cursor app.

Cancel Yes, Log In

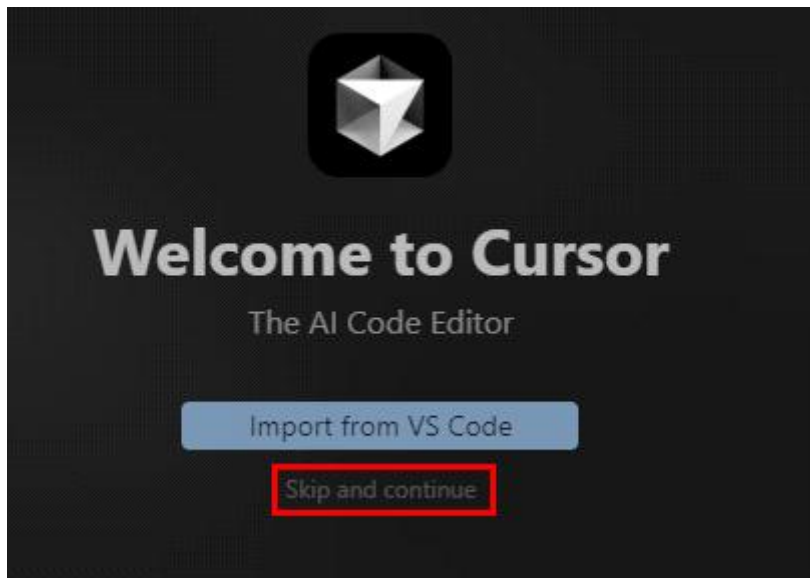
- 로그인 설정 완료(무료 버전)

All set! Feel free to return to Cursor.

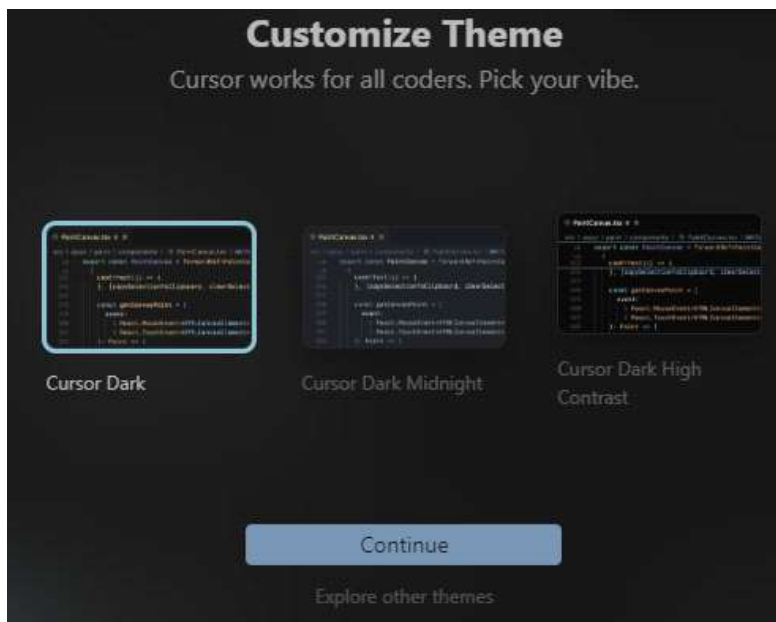
For any issues, reach out at hi@cursor.com

4. 커서 초기 설정

- 웰컴 투 커서 화면
 - 생략 및 계속 선택

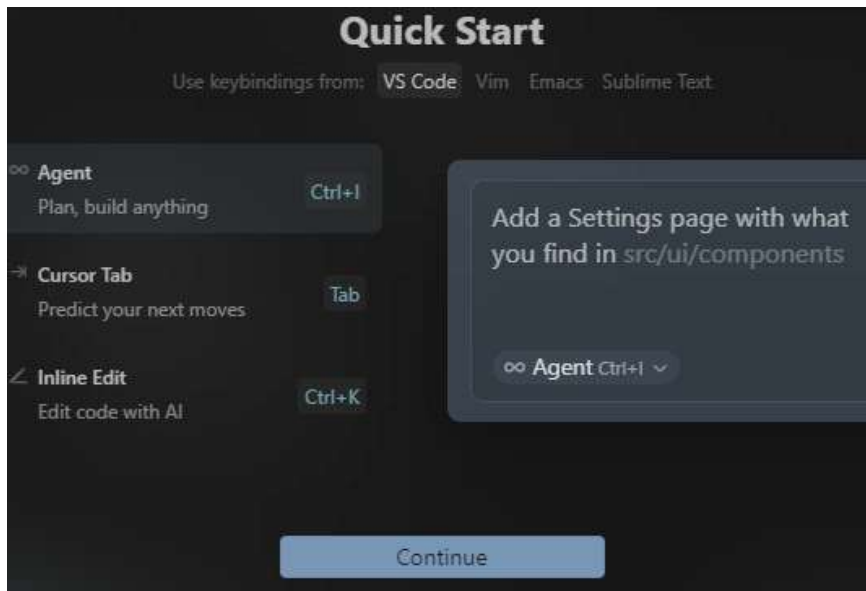


- 테마 커스터마이징
 - 계속



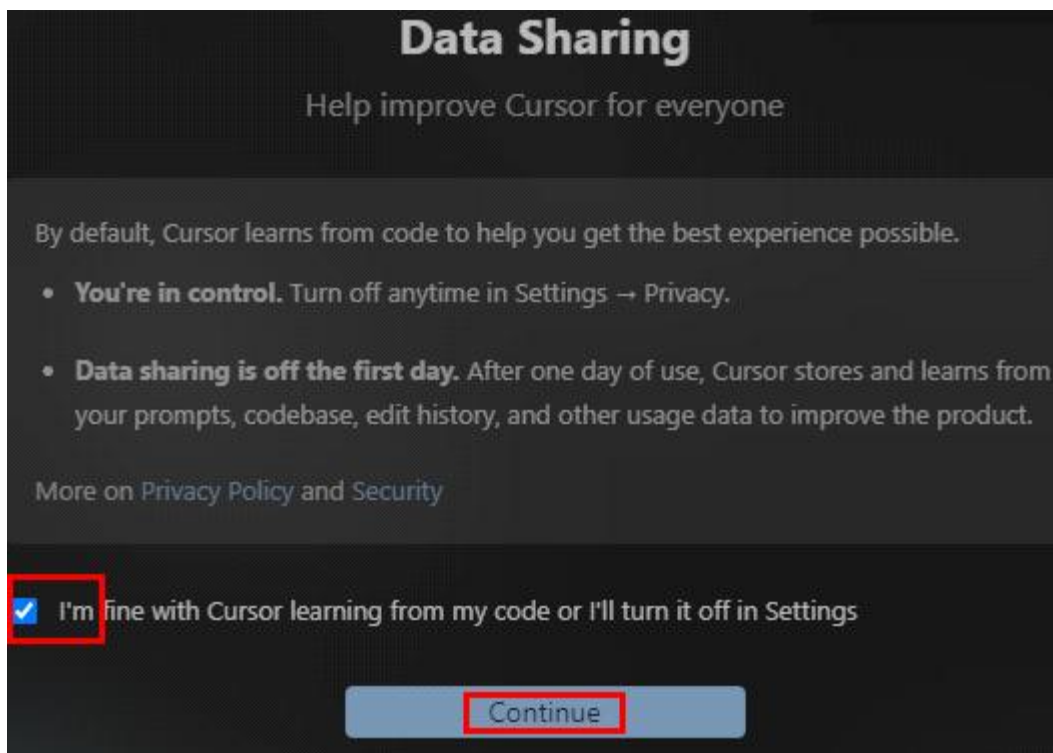
■ 퀵 스타트

- 에이전트, 커서 탭, 인라인 편집 기능 소개
- 에이전트 모드: AI 대화창 통해 명령으로 앱 작성 등 질문/요청
- 인라인 편집 모드: 에디터 내 코드 수정 명령
- 계속



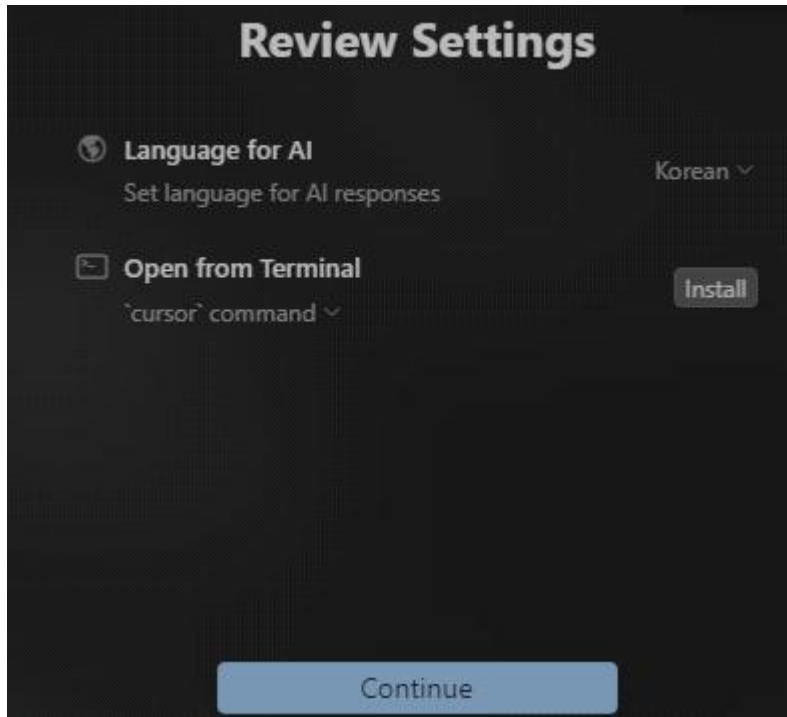
■ 데이터 공유에 체크

- 계속



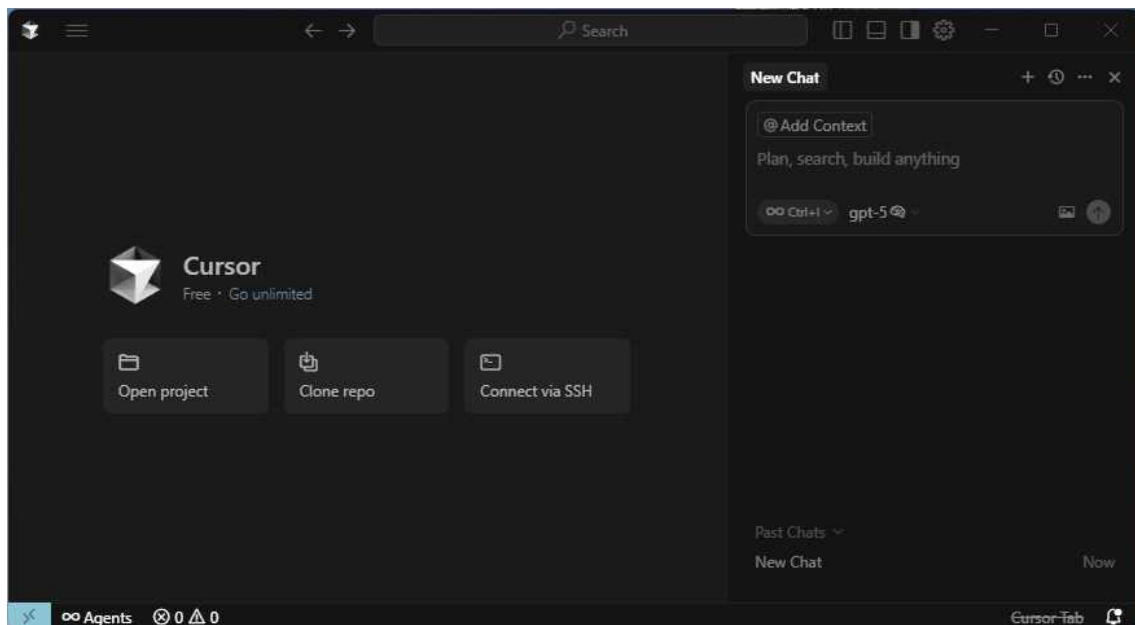
■ 리뷰 설정

- 언어: Korean 선택
- 계속



■ 커서 초기 설정 완료

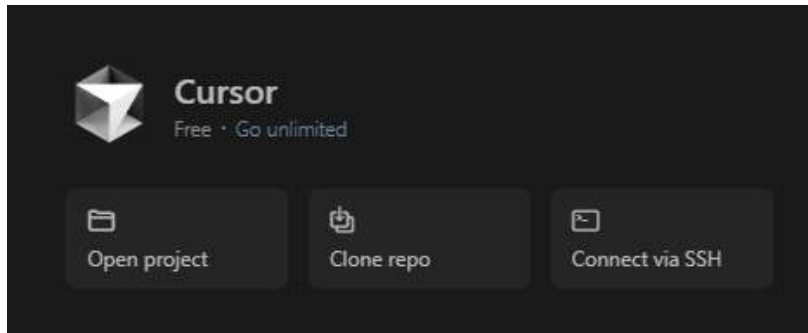
- 좌측에 프로젝트 열기 등 표시
- 우측에 에이전트 채팅창
- 현재 기본 에이전트로 gpt-5 선택됨



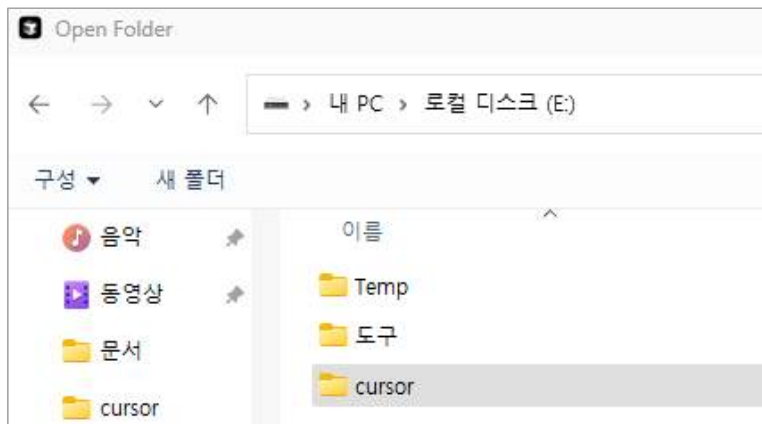
5. 새 프로젝트 폴더 생성

■ 커서 재시작

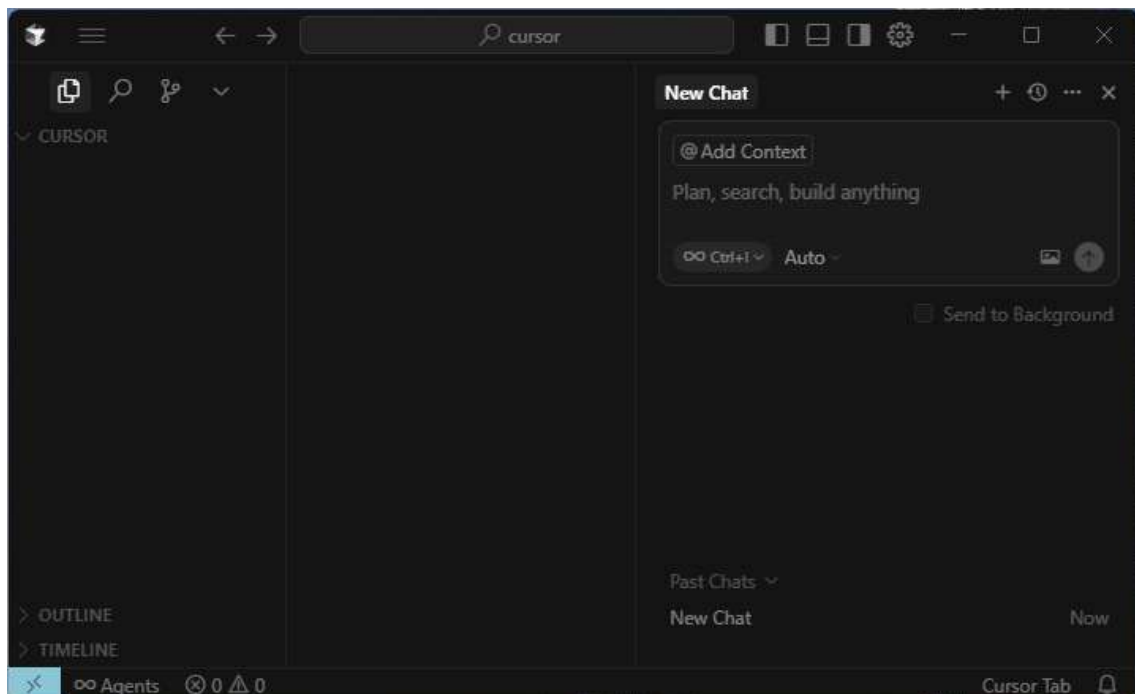
- 프로젝트 열기(Open project) 선택



- 프로젝트 폴더 cursor 생성 및 폴더 선택



- 프로젝트 폴더 cursor 및 에이전트 채팅창 나타남



02 추가 도구 설치

1. 깃(Git) 설치

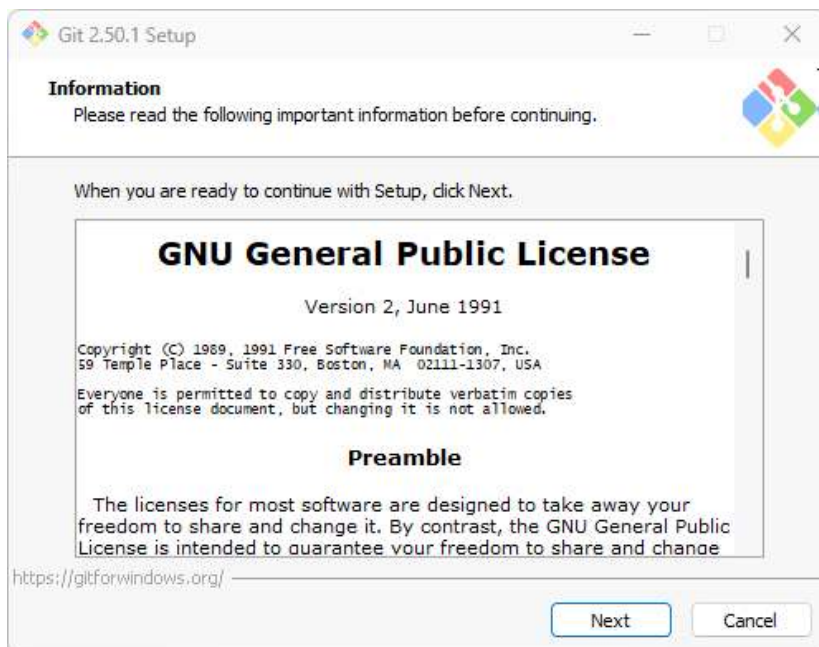
- 깃 파일 다운로드
 - <https://git-scm.com/download/win>



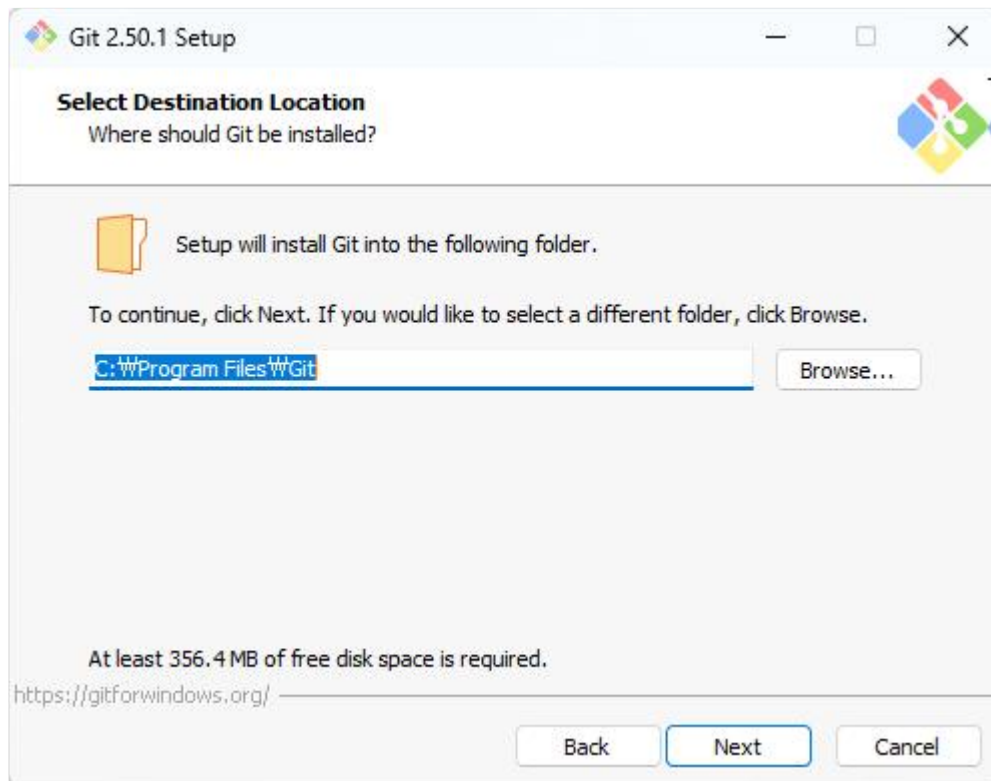
- Git-...-64-bit.exe 깃 설치 파일 다운로드



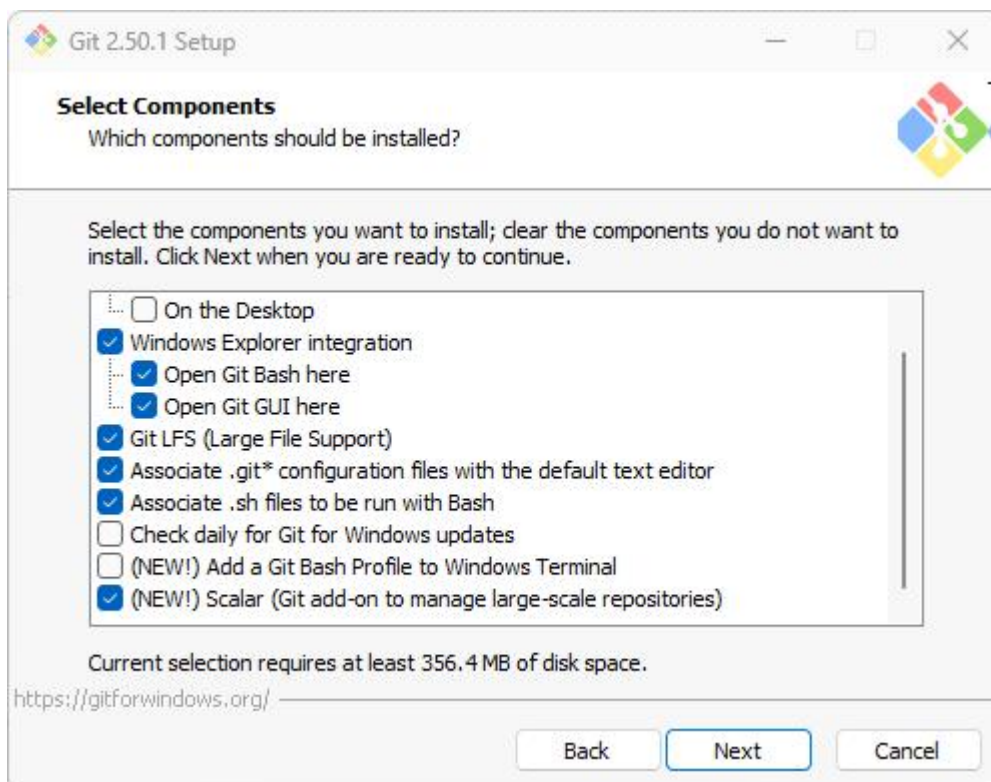
- 깃 설치
 - 깃 설치 파일 Git-...-64-bit.exe 실행
 - 라이선스 안내 확인 후 다음



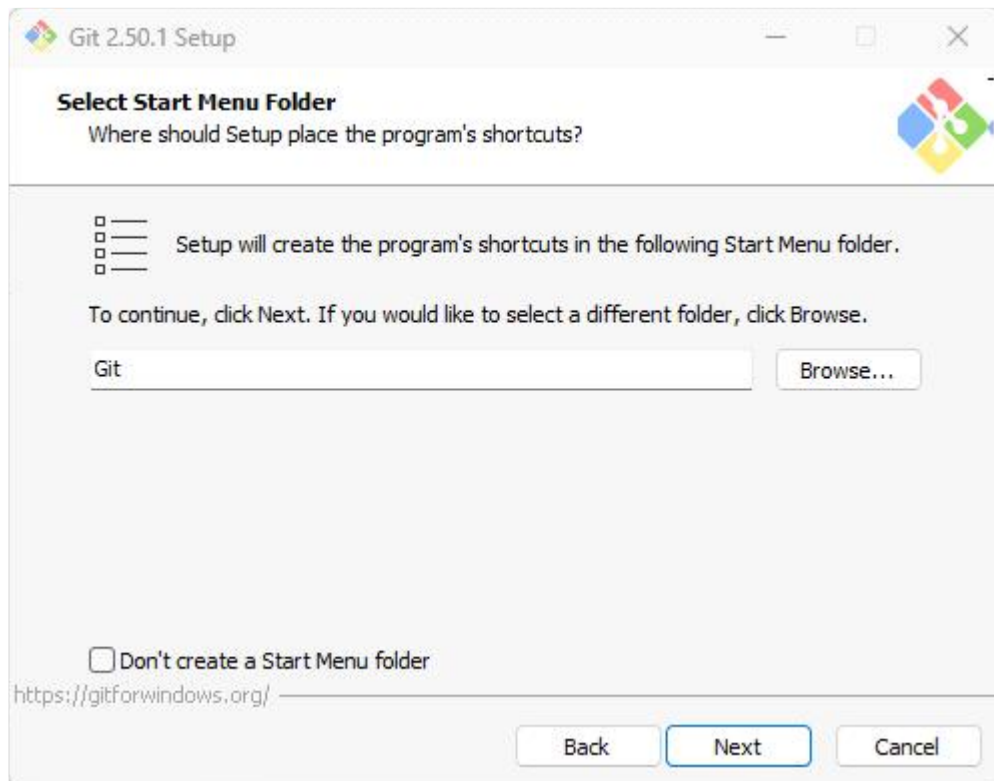
- 설치 경로 기본값 선택



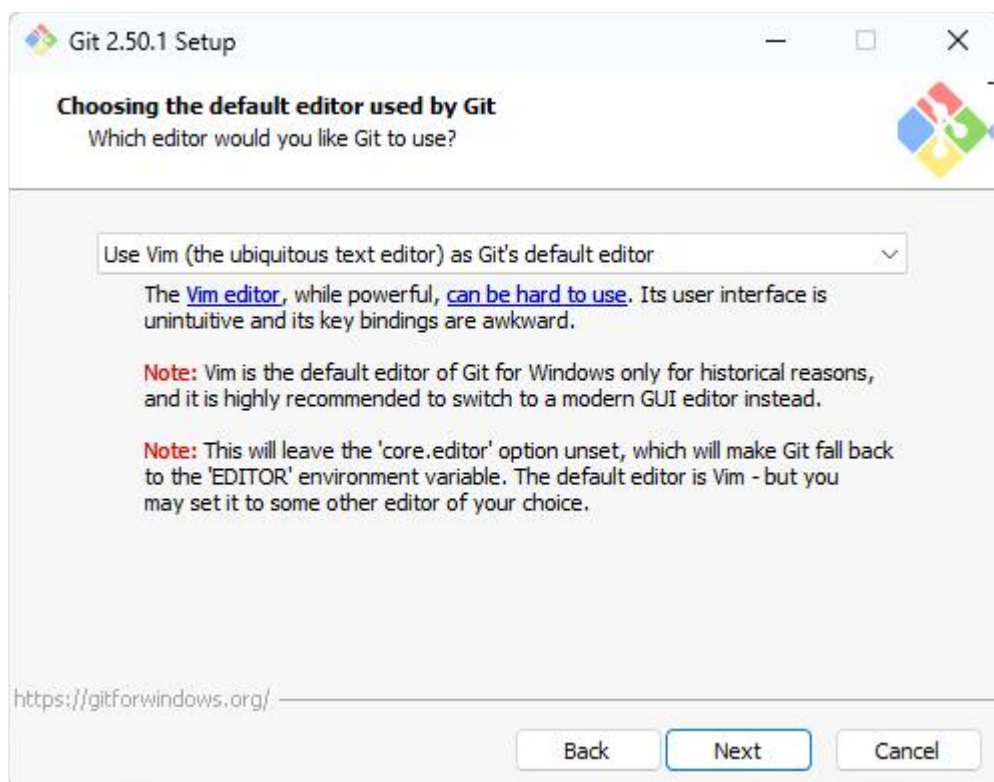
- 컴포넌트 기본값 선택



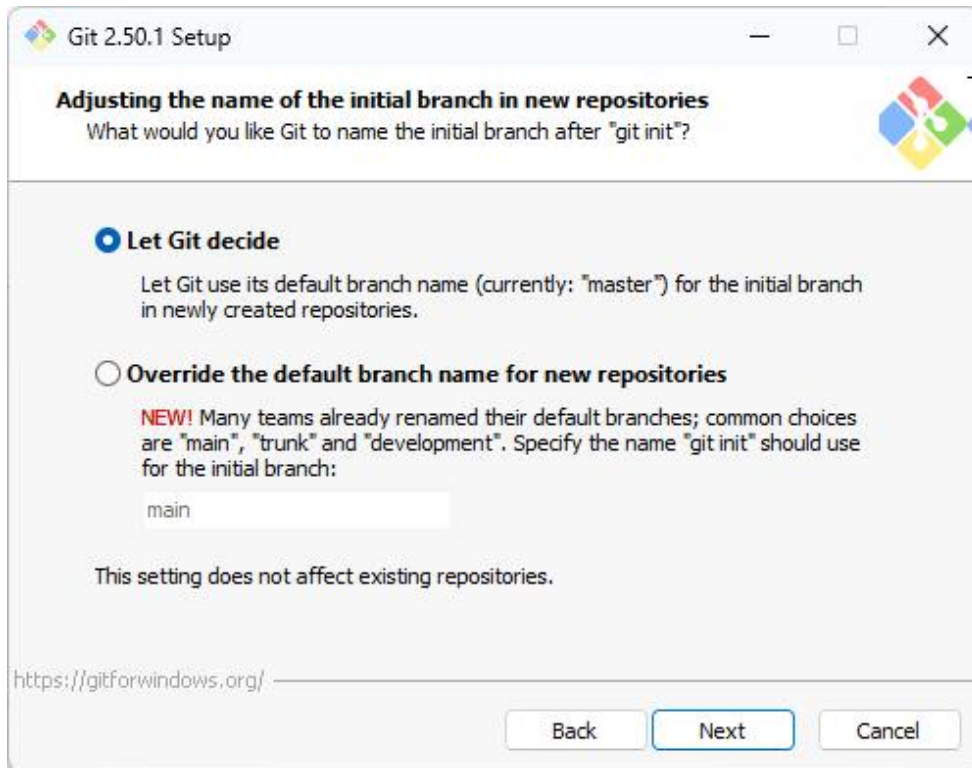
- 시작 메뉴 폴더 기본값 선택



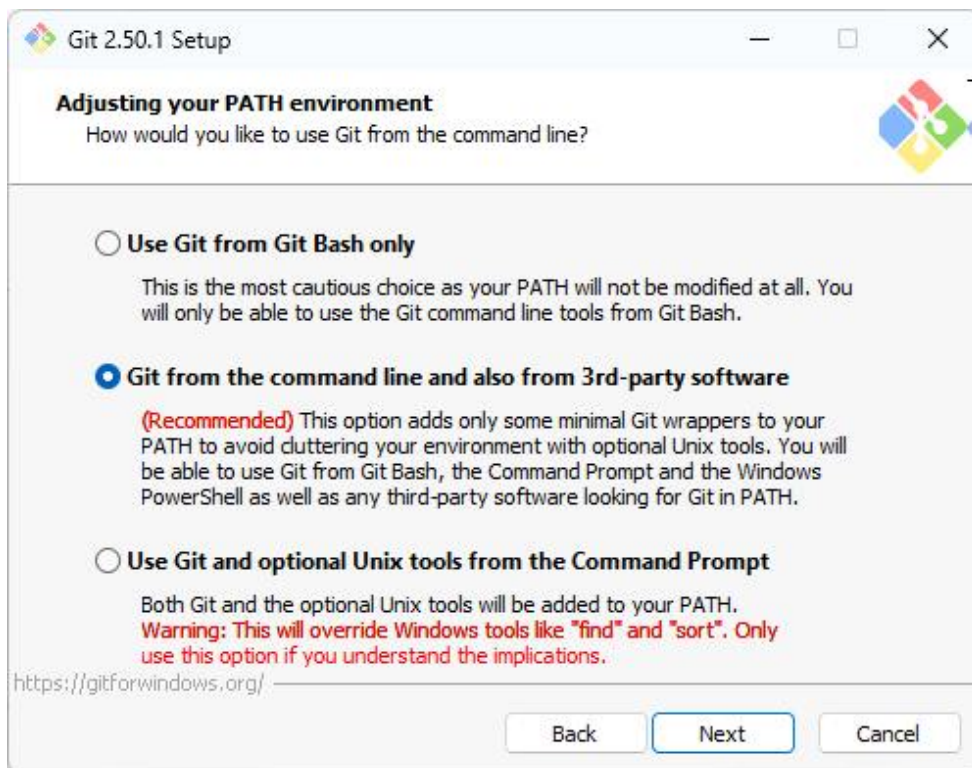
- 기본 편집기 기본값 선택



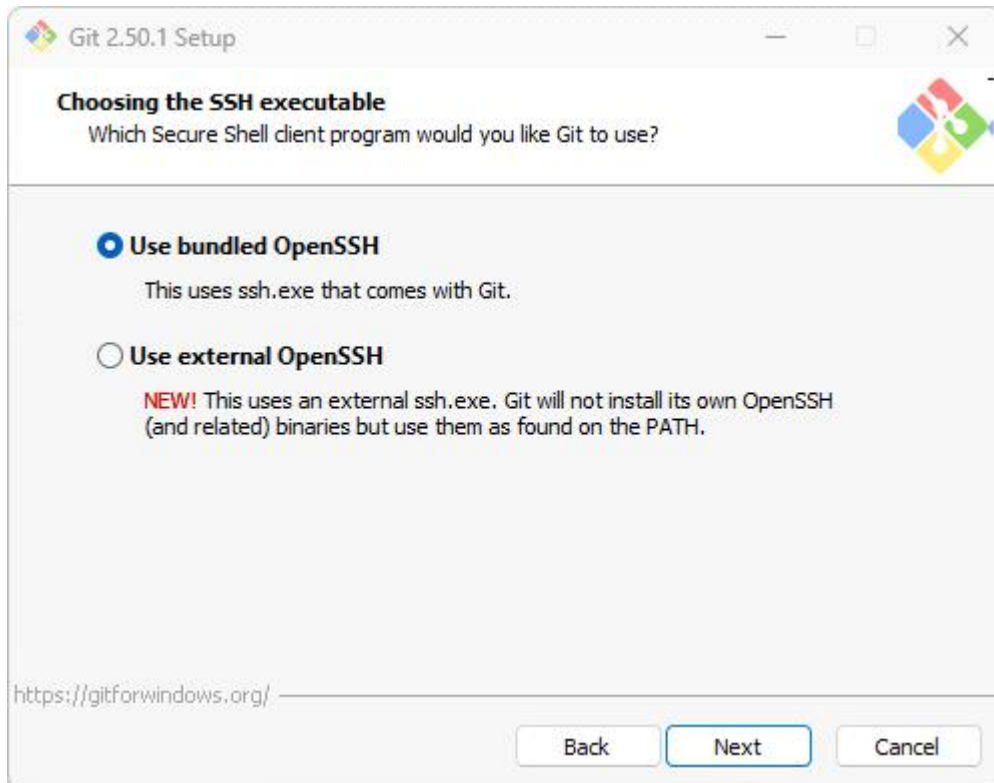
- 초기 브랜치 네임 기본값 선택



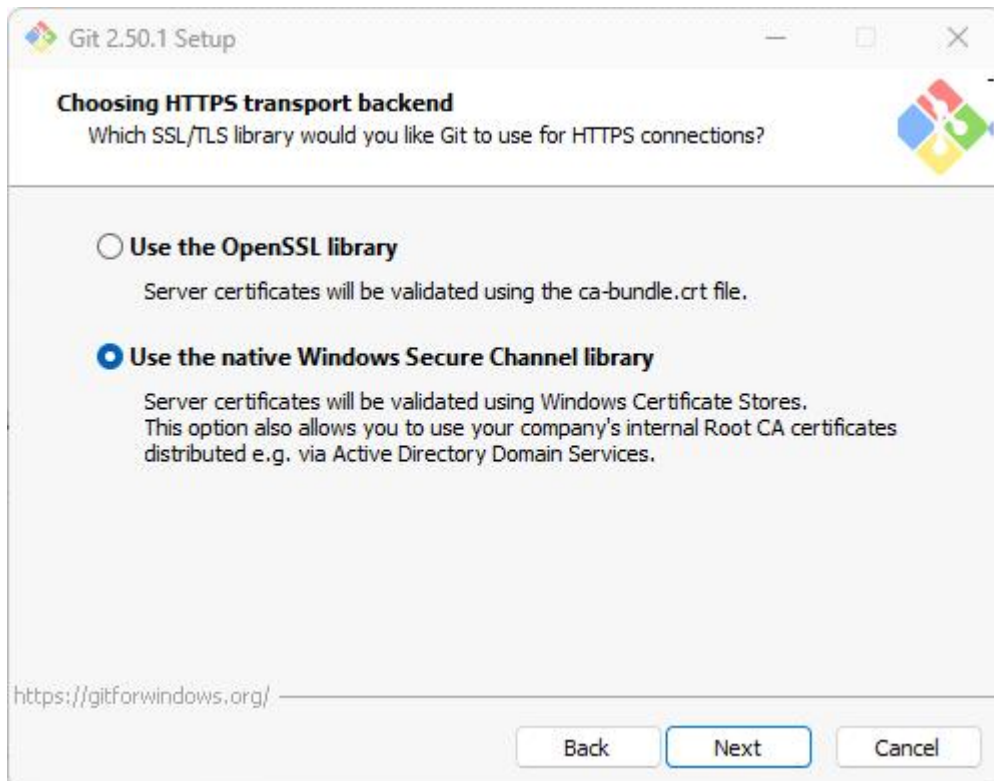
- PATH 환경 변수 설정
- "Git from the command line and also from 3rd-party software" 옵션 선택
- 터미널 어디서든 git 명령어 사용 가능



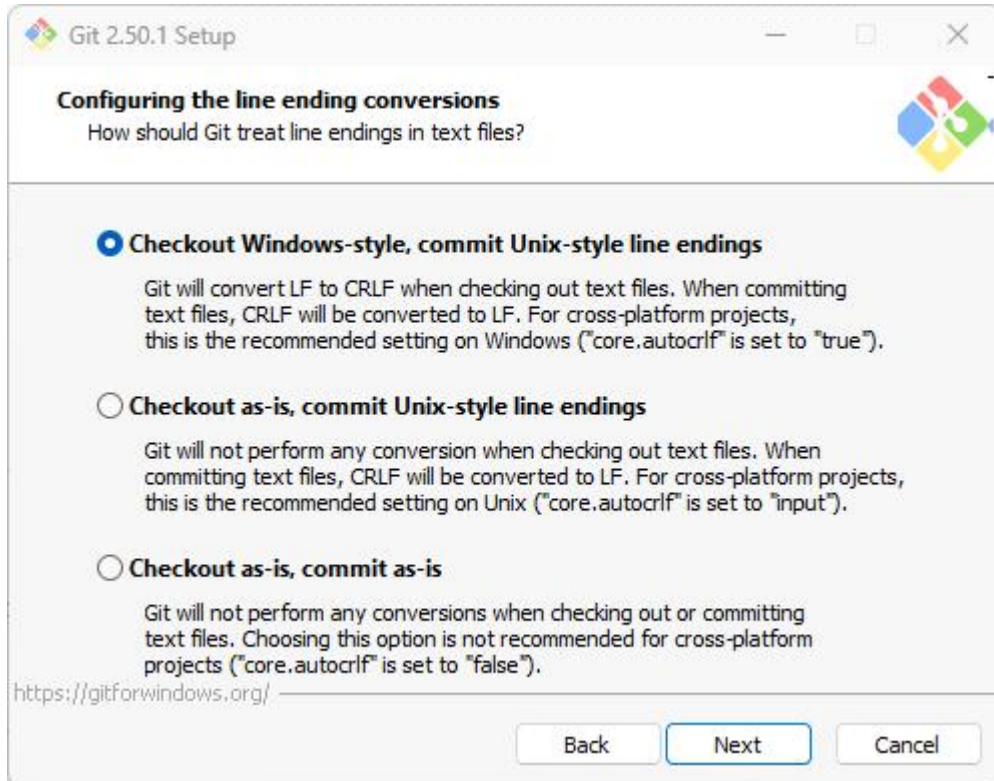
- SSH 기본값 “Use bundled OpenSSH” 선택



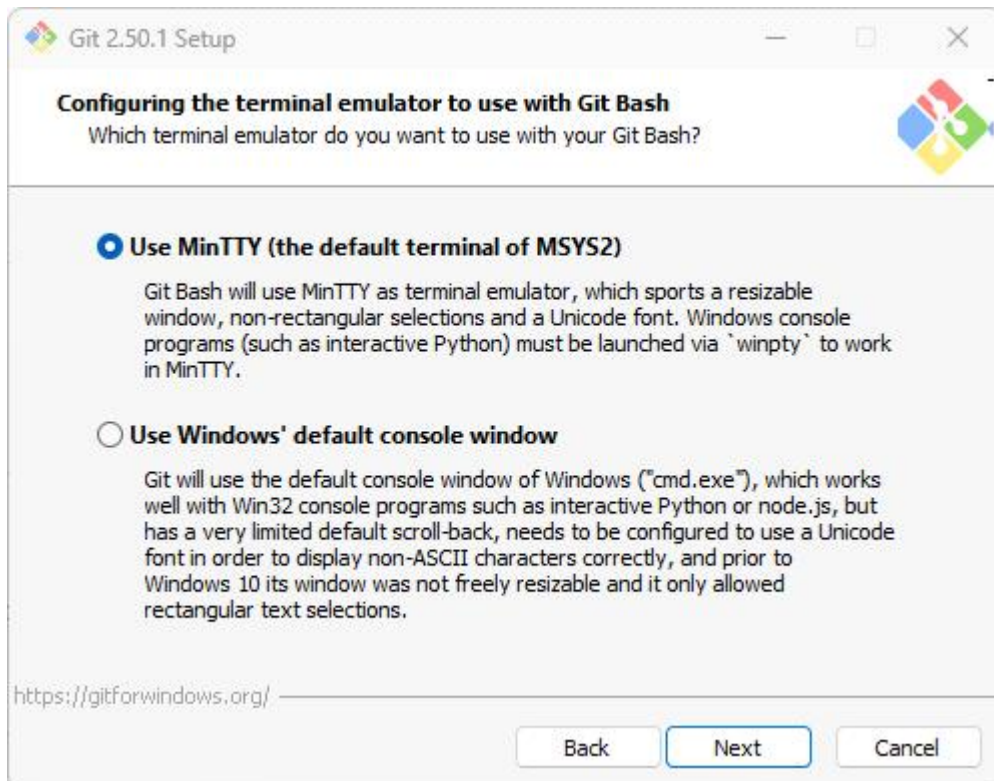
- HTTPS 기본값 “Use the native Windows Secure Channel library” 선택



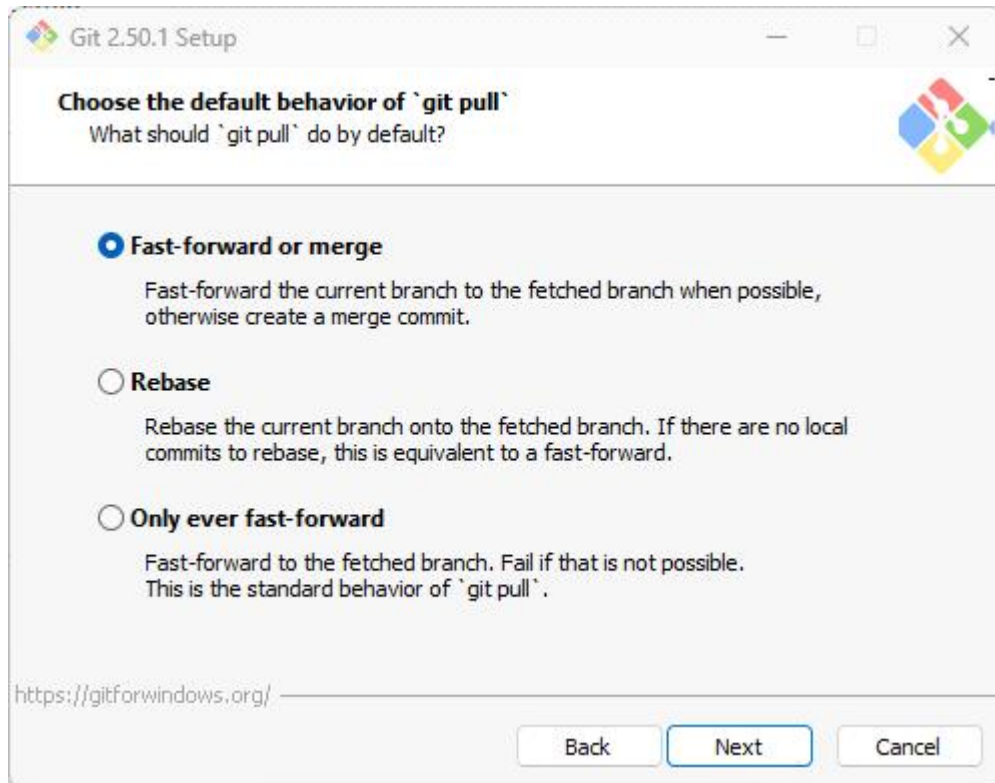
- 라인 엔딩 변환 기본값 선택



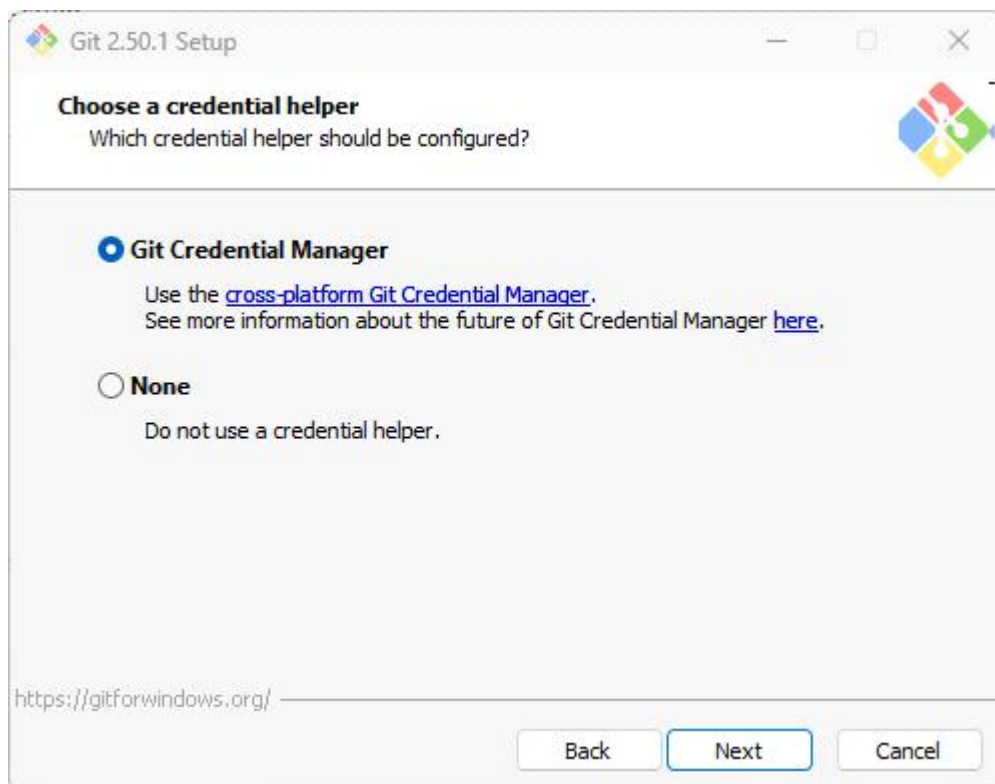
- 터미널 에뮬레이터 기본값 선택



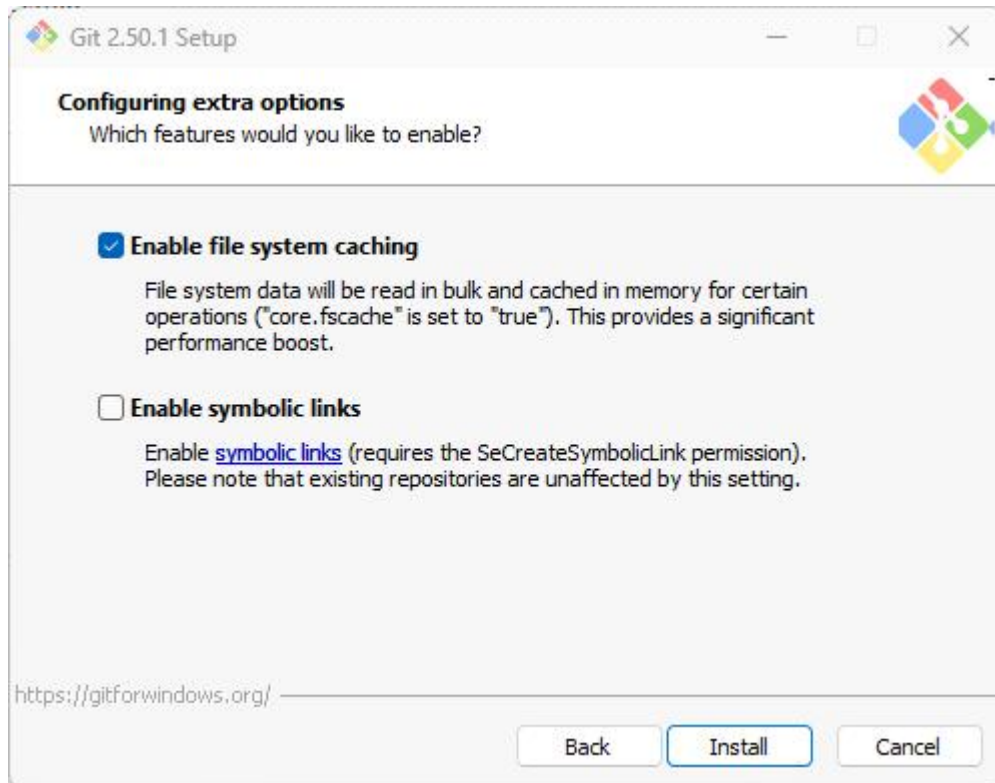
- git pull 기본값 선택



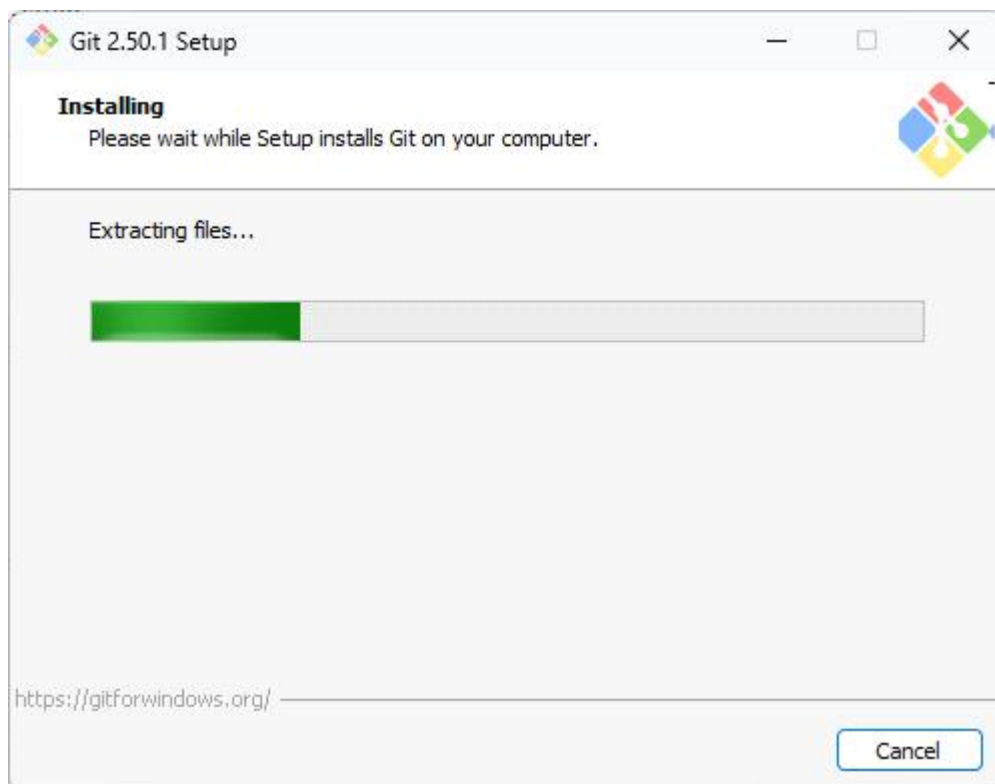
- 크리덴셜 헬퍼 기본값 선택



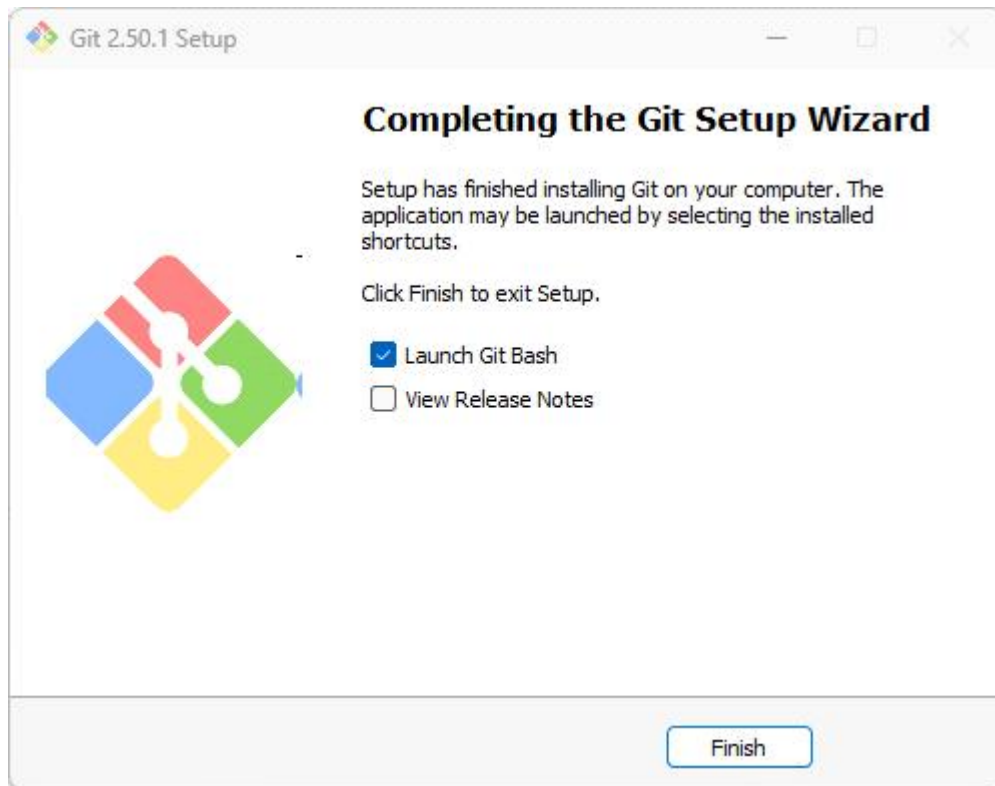
- 추가 옵션 기본값 선택



- 설치 진행



- 설치 완료
- “Launch Git Bash” 체크



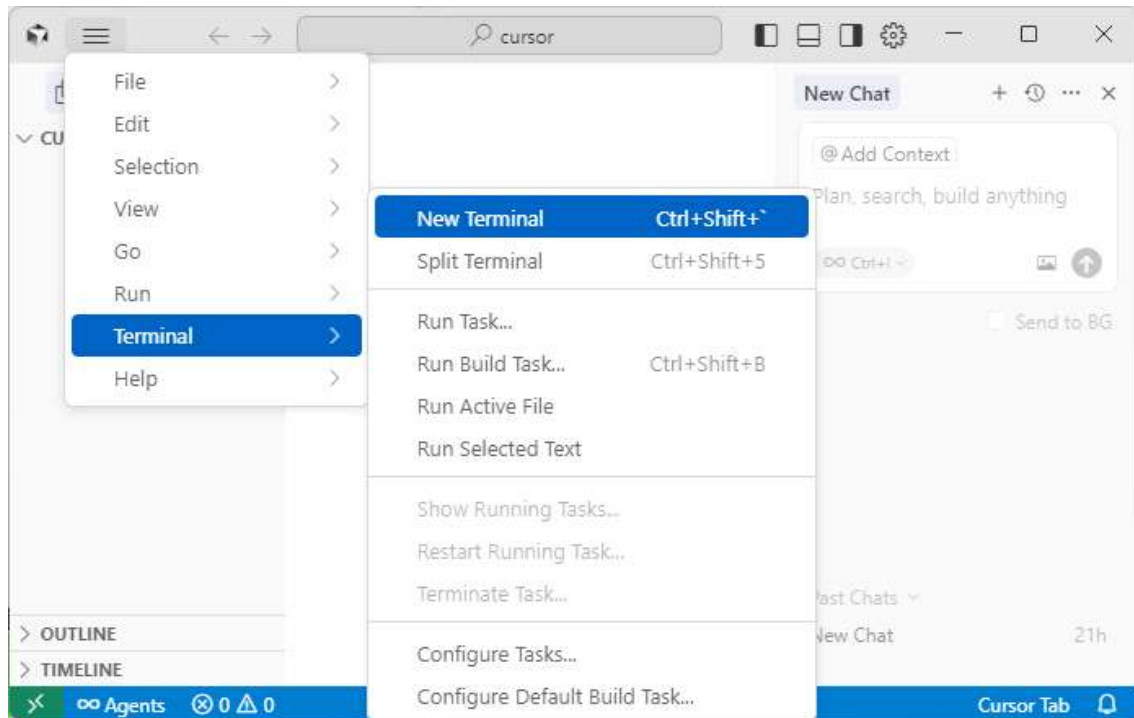
- 설치 확인
- Git Bash 터미널에서 “git --version” 명령어 실행

The image shows a terminal window titled 'MINGW64:/c/Users/csw'. The prompt is 'csw@DESKTOP-DKQ3788 MINGW64 ~'. The user has entered the command '\$ git --version', and the output is 'git version 2.50.1.windows.1'. The prompt is now '\$'.

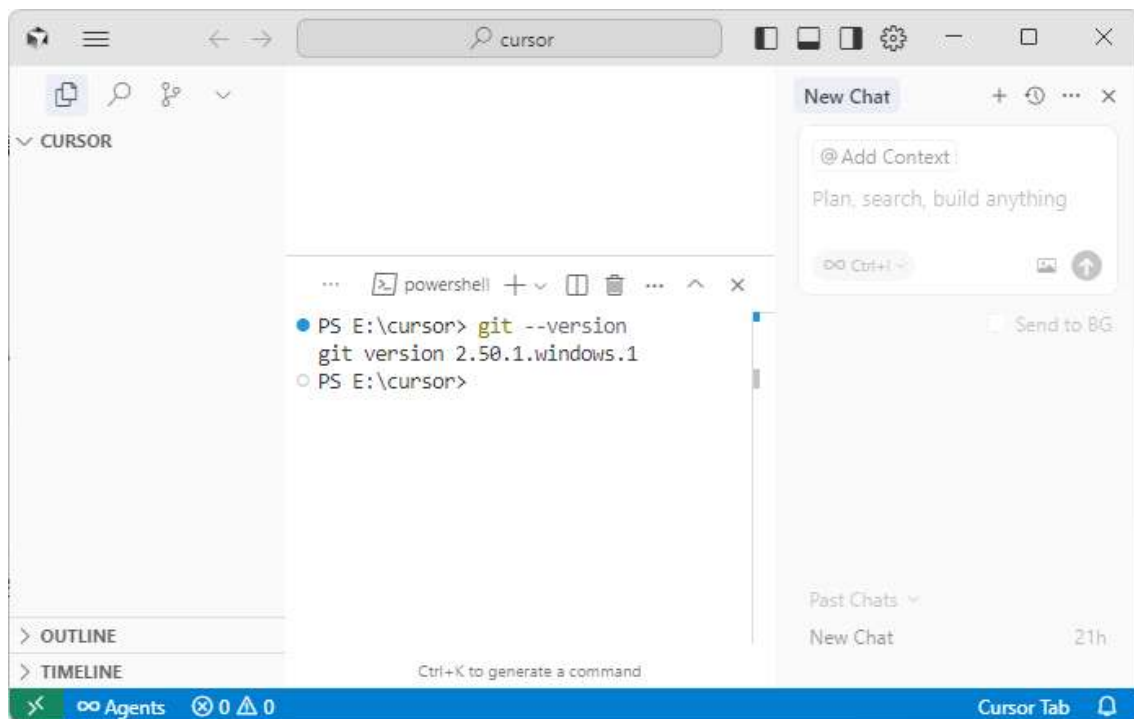
```
csw@DESKTOP-DKQ3788 MINGW64 ~  
$ git --version  
git version 2.50.1.windows.1  
  
csw@DESKTOP-DKQ3788 MINGW64 ~  
$ .....
```

- 버전 정보 출력되면 정상 설치 완료

- 커서 터미널 열기



- 커서 터미널에서 깃 버전 확인



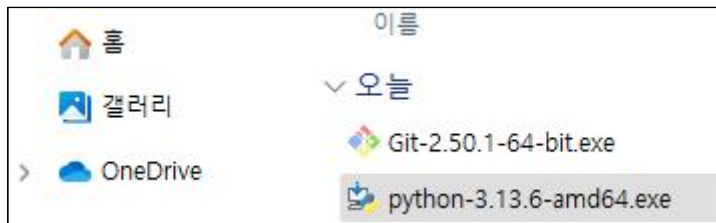
2. 파이썬(Python) 설치

■ 파이썬 파일 다운로드

- <https://python.org/downloads/>

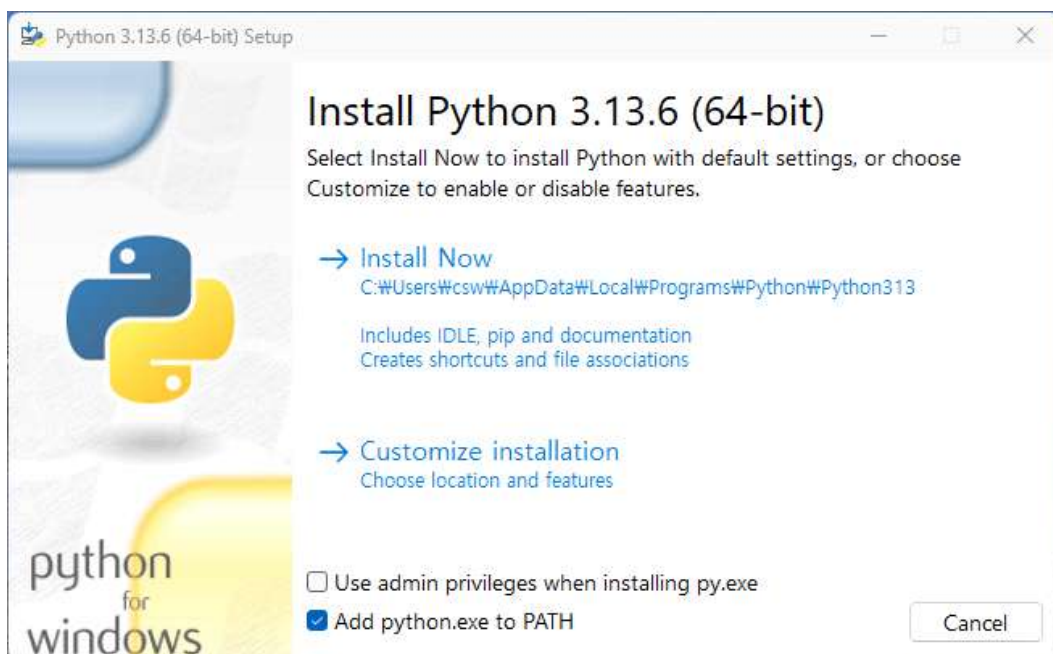


- python-...-amd64.exe 파이썬 3 설치 파일 다운로드

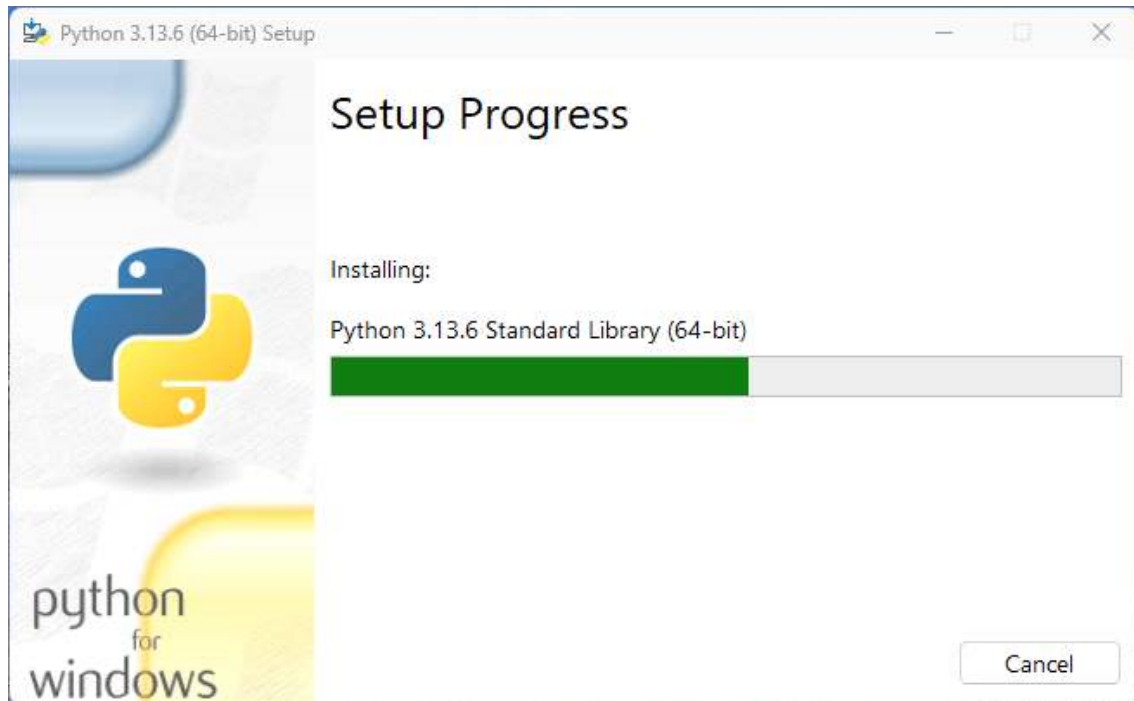


■ 파이썬 설치

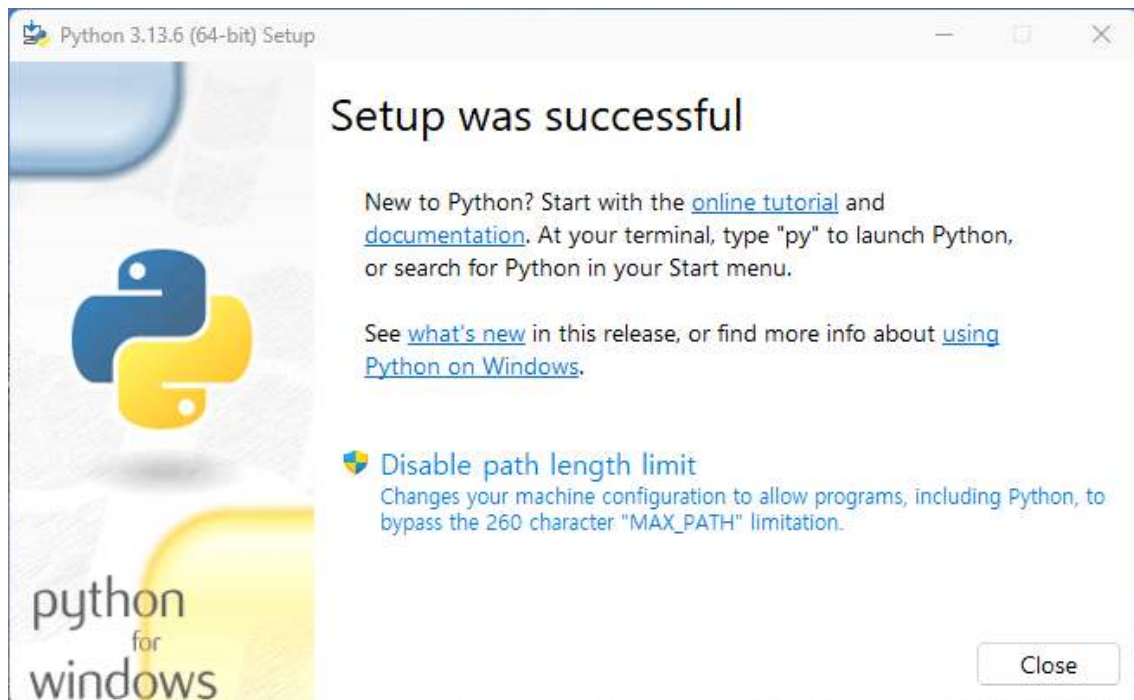
- 파이썬 설치 파일 python-...-amd64.exe 실행
- 파이썬 3 설치 창
- “Add python.exe to PATH” 선택 후 “Install Now” 클릭



- 설치 진행

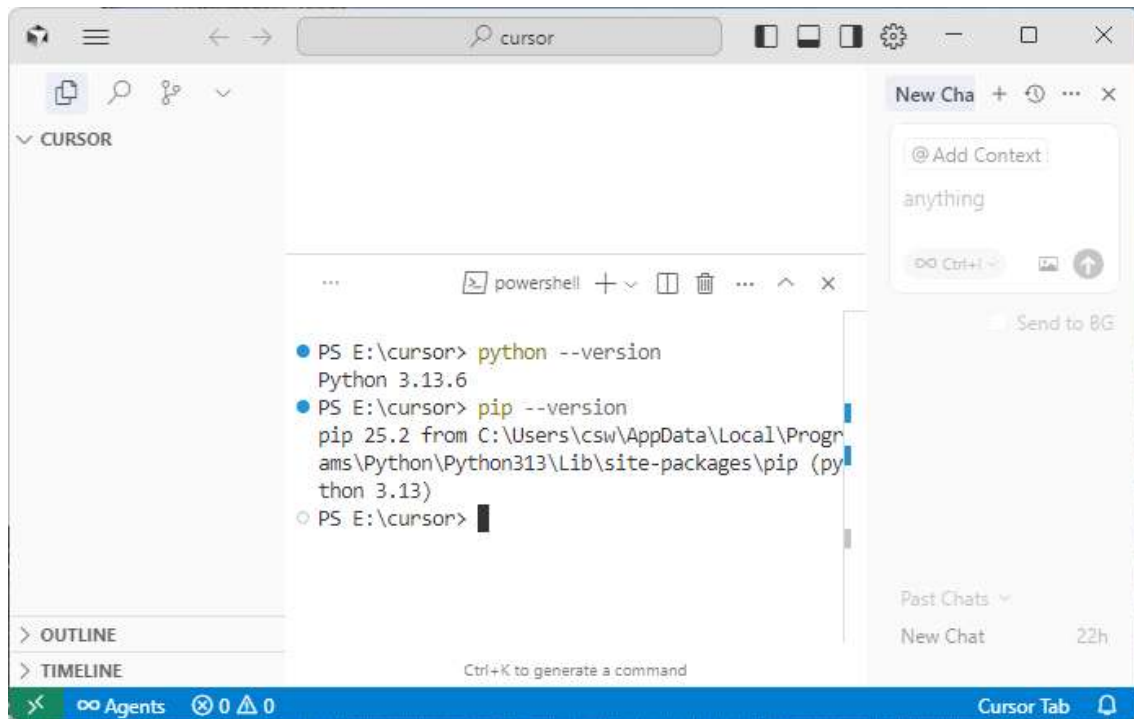


- 설치 완료



- Close

- 설치 확인
- 터미널에서 다음 명령어 실행하여 파이썬 버전 확인
- `python --version`
- `pip --version`



- 버전 확인되면 설치 완료 성공

03 앱 개발

1. 고객 피드백 분석 앱 제품 요구사항 정의서(Product Requirements Document) 작성

■ 개발 앱 요약

- 사용자가 피드백 데이터를 업로드하면, 자동으로 감성 분석과 키워드 추출을 진행하고,
- 그 결과를 보기 쉽게 시각화해주는 웹 기반 앱 개발
- 직관적이고 가벼운 UI/UX, 그리고 별도의 설치 없이 Streamlit Cloud에서 쉽게 실행

■ 앱 개요(App Overview)

- 이름: 고객 피드백 분석(Streamlit 버전)
- 플랫폼: Web(Streamlit Cloud에 배포)
- 목적: 고객 피드백 데이터를 분석하여 **감성(긍정/부정/중립)**과 주요 키워드 시각화
- 대상 사용자: 고객 서비스팀, 마케팅 팀

■ 핵심 기능(Core Features)

- 텍스트 감성 분석: 고객 피드백의 텍스트를 분석해 긍정 / 부정 / 중립으로 분류
- 이슈 키워드 추출: 자주 언급되는 핵심 키워드 자동 추출, 시각적 표현 (바 차트 등)
- 분석 결과 시각화: 감성 분포 및 키워드 빈도 시각적 제공, 이해를 위한 컬러풀 차트
- 데이터 업로드: 사용자가 CSV 또는 Excel 파일 업로드, 업로드 데이터 기반 분석 실행

■ 사용자 여정(User Journey)

- 피드백 데이터 파일 업로드
- 텍스트 데이터 전처리 및 감성 분석 자동 실행
- 시각화된 분석 결과 확인
- 기간 또는 제품군별 필터링 기능 제공

■ 기술 스택(Tech Stack)

- Frontend: Streamlit (UI와 데이터 표시를 통합적으로 처리)
- Backend: Python 코드 (Streamlit 내에서 실행)
- Database: 가벼운 저장 방식
- 인증: 기본적으로 공개 접근 (로그인 없이 사용 가능)

■ 디자인 방향(Design Direction)

- 스타일: 심플하고 직관적인 UI
- 시각화: 컬러풀한 그래프 사용
- 참고한 스타일: Streamlit 예제 앱 + Plotly Dash 스타일

■ 우선순위(Priorities)

<Must-Have (반드시 구현해야 할 항목)>

- 감성 분석과 키워드 추출의 정확도
- 사용자가 데이터를 쉽게 업로드하고 분석 결과 확인 가능
- Streamlit Cloud에서 문제없이 실행 가능

<Nice-to-Have (있으면 좋은 항목)>

- 분석 결과를 PDF로 보고서 형태로 다운로드하는 기능

■ 고객 피드백 분석(Customer Feedback Analysis) PRD

- 메모장에 아래처럼 xml 형식 파일 작성하고 cfa-prd.txt 텍스트 파일로 저장

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<product_spec>
```

```
  <app_overview>
```

```
    <title lang="ko">고객 피드백 분석 (Streamlit 버전)</title>
```

```
    <platform>Web (Streamlit Cloud 배포)</platform>
```

```
    <purpose lang="ko">고객 피드백 데이터 분석, 주요 이슈와 감성 분포 시각화</purpose>
```

```
    <target_user lang="ko">고객 서비스팀 및 마케팅 팀</target_user>
```

```
  </app_overview>
```

```
<core_features>
```

```
  <feature>
```

```
    <title lang="ko">텍스트 감성 분석</title>
```

```
    <description lang="ko">피드백에서 긍정, 부정, 중립 감성 분류</description>
```

```
  </feature>
```

```
  <feature>
```

```
    <title lang="ko">이슈 키워드 추출</title>
```

```
    <description lang="ko">자주 언급되는 주요 키워드 추출 및 시각화</description>
```

```
  </feature>
```

```
  <feature>
```

```
    <title lang="ko">분석 결과 시각화</title>
```

```
    <description lang="ko">감성 분포 및 키워드 빈도 차트로 제공</description>
```

```
  </feature>
```

```
  <feature>
```

```
    <title lang="ko">데이터 업로드</title>
```

```
    <description lang="ko">CSV 또는 Excel 파일 업로드 기능 지원</description>
```

```
  </feature>
```

```
</core_features>
```

```
<user_journey>
```

```
  <steps>
```

```
    <step lang="ko">피드백 데이터 파일 업로드</step>
```

```
    <step lang="ko">텍스트 데이터 전처리 및 감성 분석 실행</step>
```

```
    <step lang="ko">결과 시각화 및 주요 키워드 확인</step>
```

```
    <step lang="ko">필터로 기간 또는 제품군별 분석 가능</step>
```

```
  </steps>
```

```
</user_journey>
```

<tech_stack>

<frontend>Streamlit (통합 앱)</frontend>

<backend>Streamlit 내 Python 코드로 구현</backend>

<database>경량 데이터 저장</database>

<auth>기본적으로 공개</auth>

</tech_stack>

<design_direction>

<style lang="ko">심플하고 직관적인 UI, 시각화는 컬러풀하게</style>

<inspiration>Streamlit 예제 앱, Plotly Dash 스타일</inspiration>

</design_direction>

<priorities>

<must_have>

<item lang="ko">정확한 감성 분석 및 키워드 추출</item>

<item lang="ko">사용자가 쉽게 데이터를 업로드하고 결과 확인</item>

<item lang="ko">Streamlit Cloud에서 원활히 실행 가능</item>

</must_have>

<nice_to_have>

<item lang="ko">보고서 PDF 다운로드 기능</item>

</nice_to_have>

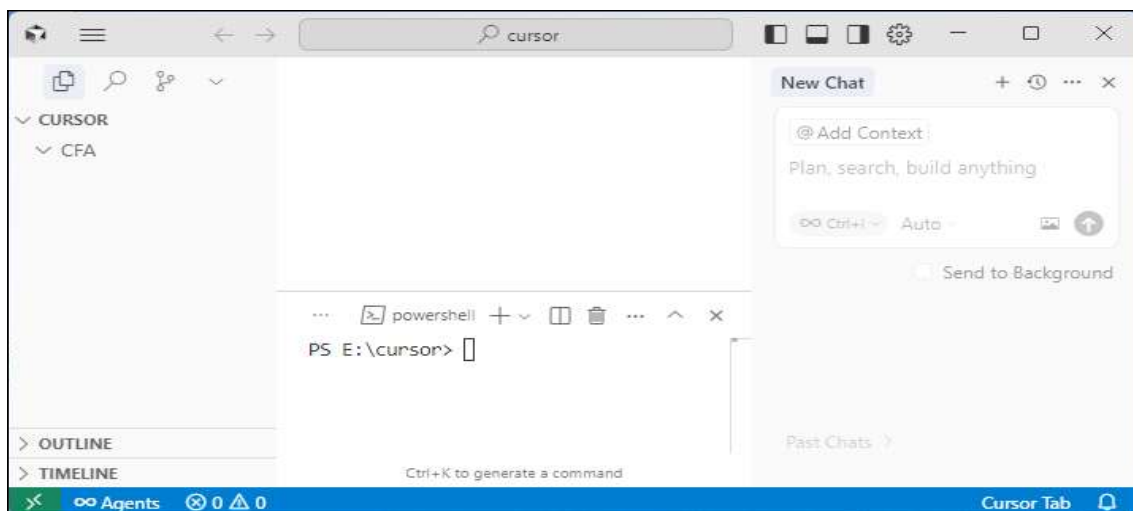
</priorities>

</product_spec>

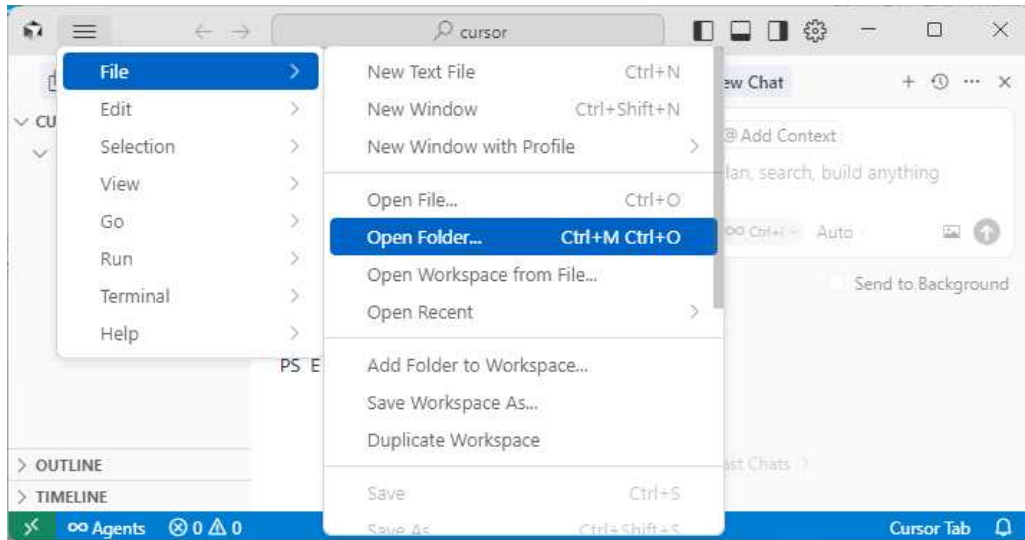
2. 앱 개발 환경 설정

■ 새 프로젝트 폴더 생성

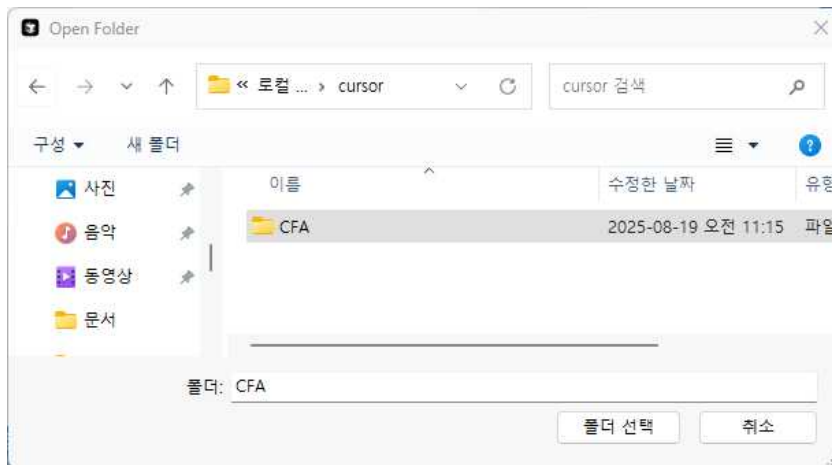
- 고객 피드백 분석 프로젝트 폴더 생성
- CURSOR/CFA



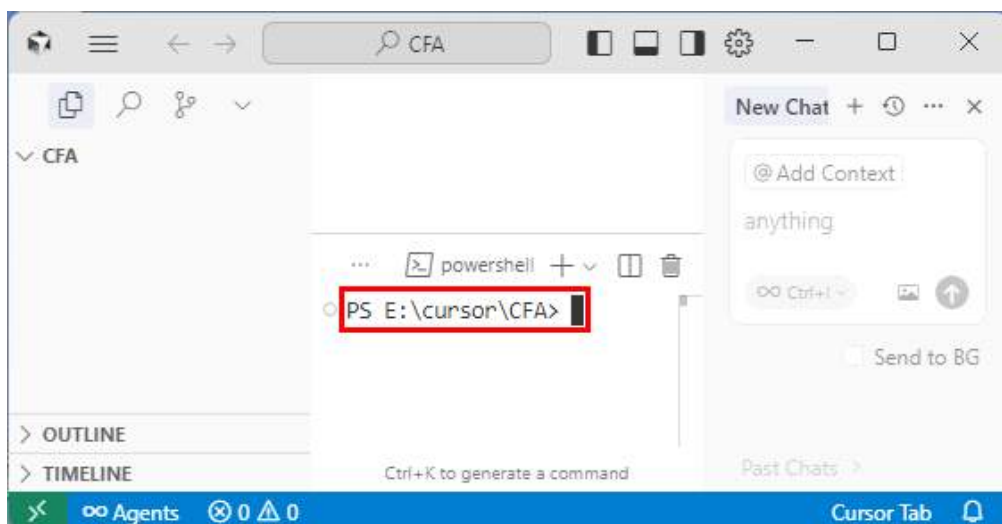
- 새 프로젝트 열기
 - File > Open Folder...



- CURSOR/CFA/ 폴더 선택



- 새 프로젝트가 CFA 폴더에서 시작됨



■ 새로운 PRD 텍스트 파일 생성

- CFA 폴더에 메모장에서 작성한 고객 피드백 분석을 위한
- cfa-prd.txt 파일 드래그 앤 드롭

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <product_spec>
3   <app_overview>
4     <title lang="ko">고객 피드백 분석 (Streamlit 버전)</title>
5     <platform>Web (Streamlit Cloud 배포)</platform>
6     <purpose lang="ko">고객 피드백 데이터를 분석하여 주요 이슈와 감성 분포를 시각화</purpose>
7     <target_user lang="ko">고객 서비스팀 및 마케팅 팀</target_user>
8   </app_overview>
9
10  <core_features>
11    <feature>
12      <title lang="ko">텍스트 감성 분석</title>
13      <description lang="ko">피드백 텍스트에서 긍정, 부정, 중립 감성 분류</description>
14    </feature>
15    <feature>
16      <title lang="ko">이슈 키워드 추출</title>
17      <description lang="ko">자주 언급되는 주요 키워드 추출 및 시각화</description>
18    </feature>
19    <feature>
20      <title lang="ko">분석 결과 시각화</title>
21      <description lang="ko">감성 분포 및 키워드 빈도 차트로 제공</description>
22    </feature>
23    <feature>
24      <title lang="ko">데이터 업로드</title>
25      <description lang="ko">CSV 또는 Excel 파일 업로드 기능 지원</description>
26    </feature>
27  </core_features>
28
29  <user_journey>
30    <steps>
31      <step lang="ko">피드백 데이터 파일 업로드</step>
32      <step lang="ko">텍스트 데이터 전처리 및 감성 분석 실행</step>
33      <step lang="ko">결과 시각화 및 주요 키워드 확인</step>
34      <step lang="ko">필터로 기간 또는 제품군별 분석 가능</step>
35    </steps>
36  </user_journey>

```

```

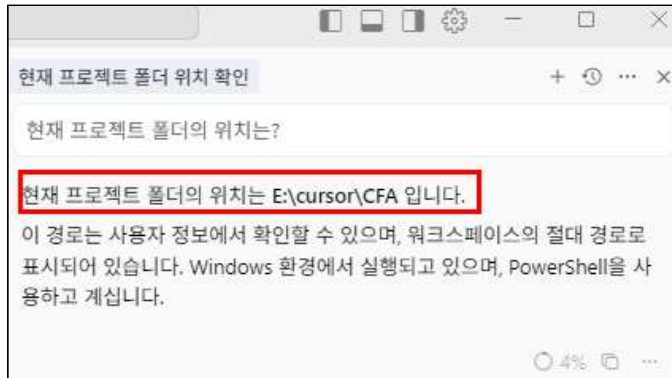
38 <tech_stack>
39   <frontend>Streamlit (통합 앱)</frontend>
40   <backend>Streamlit 내 Python 코드로 구현</backend>
41   <database>경량 데이터 저장</database>
42   <auth>기본적으로 공개</auth>
43 </tech_stack>
44
45 <design_direction>
46   <style lang="ko">심플하고 직관적인 UI, 시각화는 컬러풀하게</style>
47   <inspiration>Streamlit 예제 앱, Plotly Dash 스타일</inspiration>
48 </design_direction>
49
50 <priorities>
51   <must_have>
52     <item lang="ko">정확한 감성 분석 및 키워드 추출</item>
53     <item lang="ko">사용자가 쉽게 데이터를 업로드하고 결과 확인</item>
54     <item lang="ko">Streamlit Cloud에서 원활히 실행 가능</item>
55   </must_have>
56   <nice_to_have>
57     <item lang="ko">보고서 PDF 다운로드 기능</item>
58   </nice_to_have>
59 </priorities>
60 </product_spec>

```

3. AI 대화창을 통한 앱 개발 및 테스트

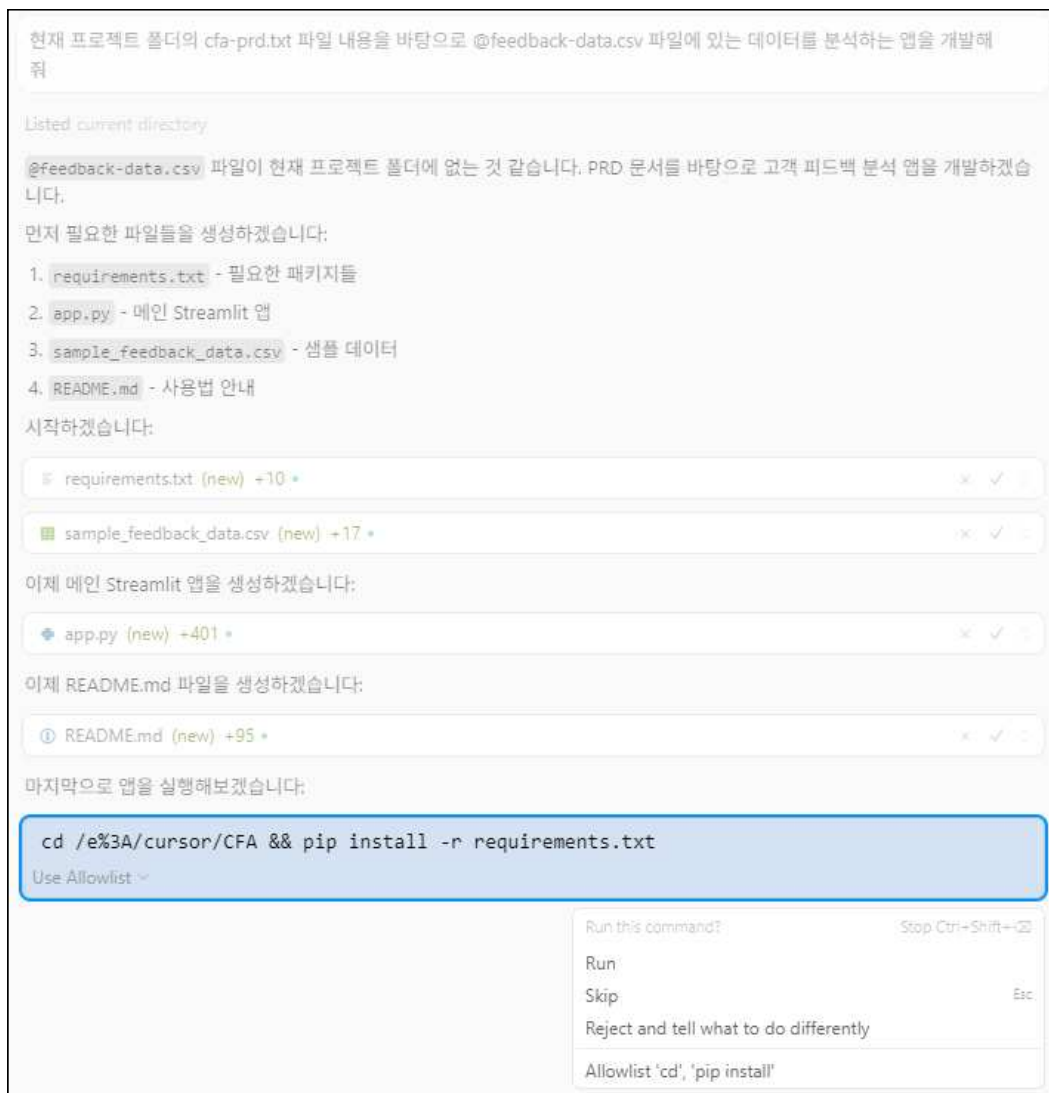
■ 현재 프로젝트 폴더 확인

- AI 대화창에서 현재 프로젝트 폴더 위치 확인 요청

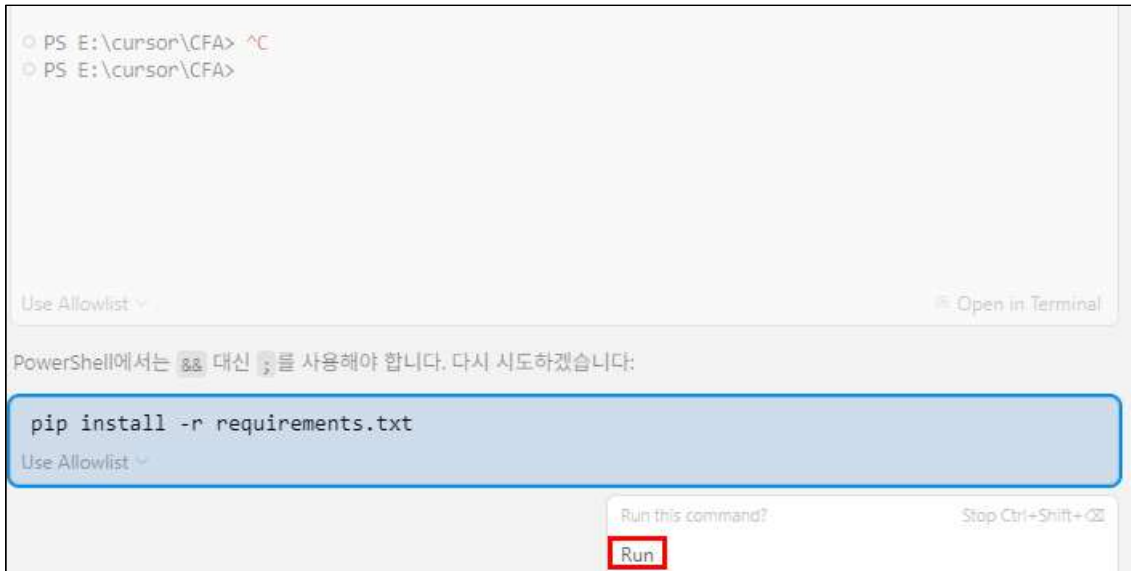


■ 앱 개발 명령(프롬프트) 입력

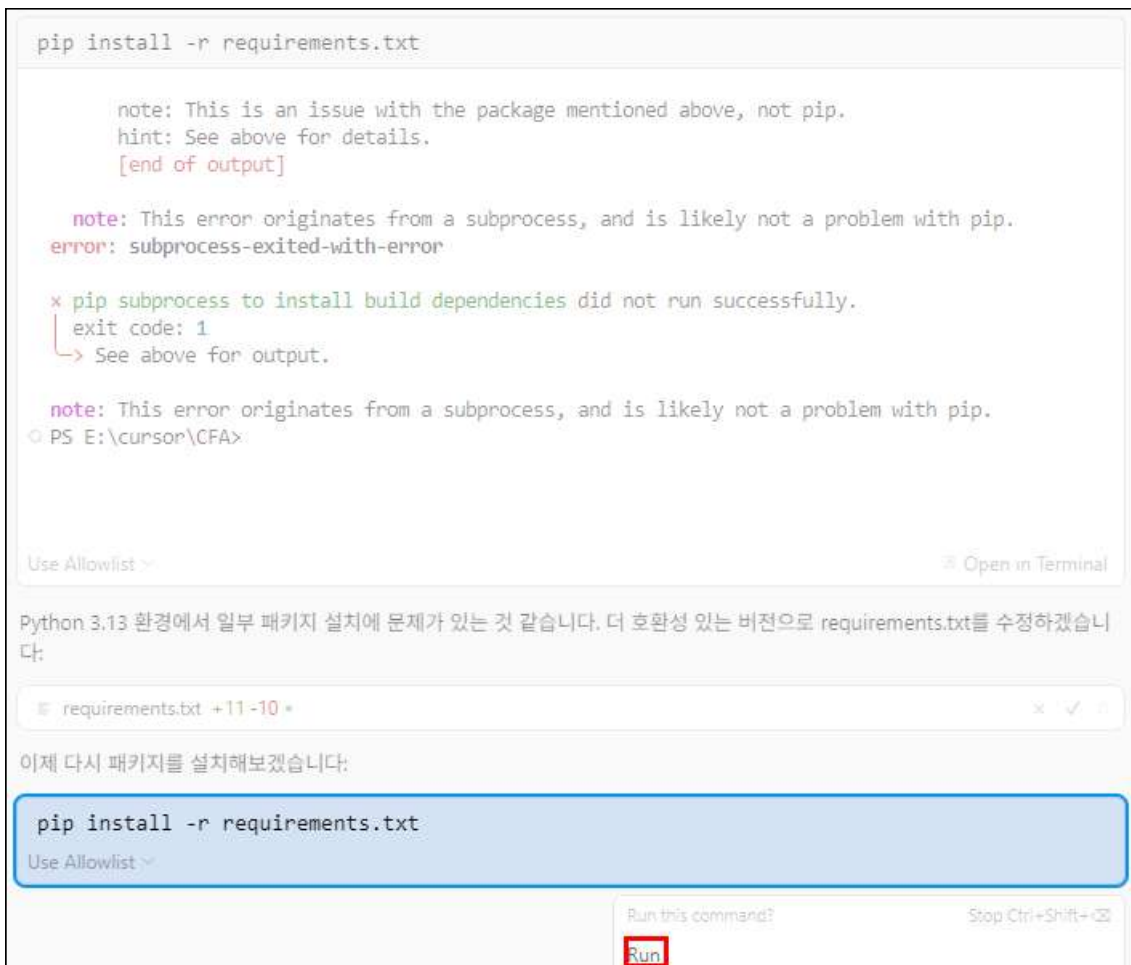
- 현재 프로젝트 폴더의 cfa-prd.txt 파일 내용을 바탕으로
- @feedback-data.csv 파일에 있는 데이터를 분석하는 앱을 개발해주세요.



- 필요한 파일들 생성: requirements.txt, app.py, sample_feedback_data.csv 등
- 필요한 패키지 설치 진행(run 클릭): pip install -r requirements.txt
- 패키지 설치 명령 오류 수정 후 다시 시도



- requirements.txt 수정 후 다시 패키지 설치



- 모든 패키지 설치 성공 완료

이제 다시 패키지를 설치해보겠습니다:

```
pip install -r requirements.txt
```

Downloading tqdm-4.67.1-py3-none-any.whl (78 kB)
Installing collected packages: pytz, watchdog, urllib3, tzdata, typing-extensions, tornado, tomli, tenacity, smmap, six, rpds-py, regex, pyparsing, pyarrow, protobuf, pillow, packaging, numpy, MarkupSafe, kiwisolver, joblib, idna, fonttools, et-xmlfile, cyclr, colorama, charset-normalizer, cachetools, blinker, attrs, tqdm, scipy, requests, referencing, python-dateutil, plotly, openpyxl, gitdb, contourpy, click, scikit-learn, pydeck, pandas, nltk, matplotlib, jsonschema-specifications, wordcloud, textblob, seaborn, jsonschema, altair, streamlit
Successfully installed MarkupSafe-3.0.2 altair-5.5.0 attrs-25.3.0 blinker-1.9.0 cachetools-6.1.0 colorama-0.4.6 contourpy-1.3.3 cyclr-0.12.1 et-xmlfile-1.0.1 fonttools-4.59.1 gitdb-4.0.12 gitpython-3.1.45 idna-3.10 jinja2-3.1.6 joblib-1.5.1 jsonschema-4.25.1 jsonschema-specifications-2025.4.1 kiwisolver-1.4.9 matplotlib-3.10.5 narwhals-2.1.2 nltk-3.9.1 numpy-2.3.1 openpyxl-3.1.5 packaging-25.0 pandas-2.3.1 pillow-11.3.0 plotly-6.3.0 protobuf-6.32.0 pyarrow-21.0.0 pydeck-0.9.1 pyparsing-3.2.3 python-dateutil-2.9.0.post0 pytz-2025.2 referencing-0.36.2 regex-2025.7.34 requests-2.32.4 rpds-py-0.27.0 scikit-learn-1.7.1 scipy-1.16.1 seaborn-0.13.2 six-1.17.0 smmap-5.0.2 streamlit-1.48.1 textblob-0.19.0 threadpoolctl-3.6.0 tomli-0.10.2 tornado-6.5.2 tqdm-4.67.1 typing-extensions-4.12.2 urllib3-2.5.0 watchdog-6.0.0 wordcloud-1.9.4

PS E:\cursor\CFA>

Use Allowlist ▾ Open in Terminal

훌륭합니다! 모든 패키지가 성공적으로 설치되었습니다. 이제 Streamlit 앱을 실행해보겠습니다:

```
streamlit run app.py
```

Use Allowlist ▾

Run this command? Stop Ctrl+Shift+Q

Run

■ 앱 실행

- streamlit run app.py

터미널에서 `streamlit run app.py` 명령을 실행하겠습니다.

```
streamlit run app.py
```

Running in background. [Click to focus](#)

앱이 백그라운드에서 실행되고 있습니다! 🎉

이제 브라우저에서 <http://localhost:8501> 로 접속하여 고객 피드백 분석 대시보드를 확인할 수 있습니다.

💎💎 앱 실행 완료!

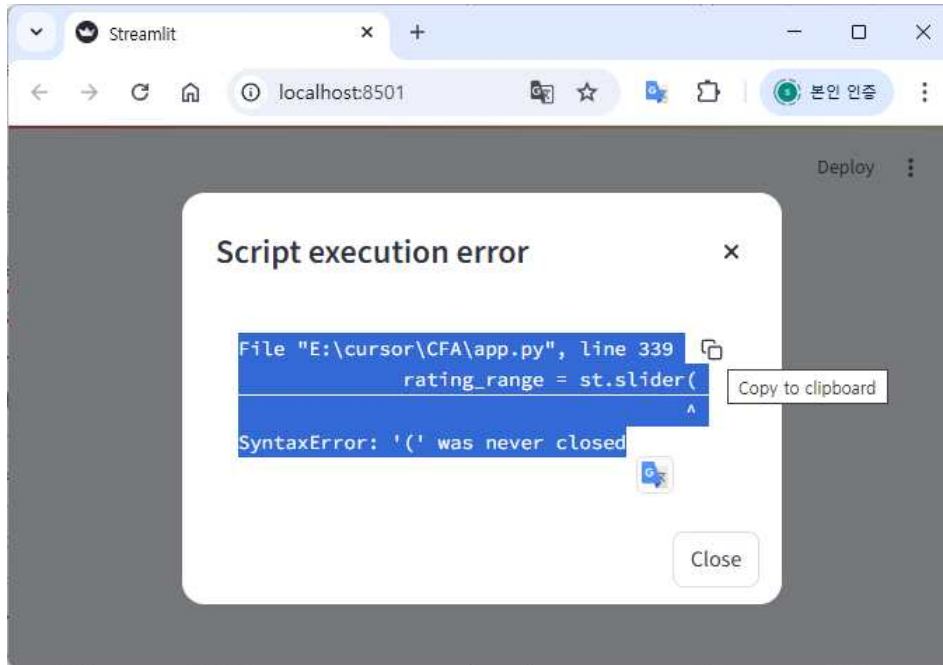
접속 URL: <http://localhost:8501>

💎💎 사용 가능한 기능들:

1. 📁 데이터 업로드: 사이드바에서 CSV/Excel 파일 업로드
2. 💎💎 샘플 데이터: 체크박스로 샘플 데이터 사용 가능
3. 😊 감성 분석: 긍정/부정/중립 분류 및 시각화
4. 💎💎 키워드 분석: 주요 키워드 추출 및 워드클라우드
5. 💎💎 제품별 분석: 제품별 평점 및 긍정 비율
6. 💎💎 카테고리별 분석: 카테고리별 통계 정보
7. 📅 시간별 분석: 월별 트렌드 분석
8. 💎💎 고급 필터링: 다양한 조건으로 데이터 필터링
9. 📄 데이터 다운로드: CSV/Excel 형식으로 결과 저장

앱이 정상적으로 실행되고 있으며, 샘플 데이터를 사용하여 모든 기능을 테스트해보실 수 있습니다!

- 브라우저에서 `http://localhost:8501` 주소를 입력하여 앱에 접속



- 앱 접속 오류 복사 후 AI에 오류 사항 전달
- AI가 오류 확인 및 수정하고 앱 재실행



- 오류 수정 후 앱 재실행 됨

💎 문법 오류 수정 완료!

`app.py` 파일의 339번째 줄에 있던 괄호 닫기 문제를 해결했습니다. 이제 앱이 정상적으로 실행되고 있습니다.

접속 URL: <http://localhost:8501>

🔧 수정된 내용:

- `st.slider` 함수의 괄호가 제대로 닫히지 않았던 문제 해결
- 이제 Python 문법 오류 없이 앱이 실행됩니다

앱이 백그라운드에서 실행 중이며, 브라우저에서 접속하여 고객 피드백 분석 대시보드를 사용할 수 있습니다!

- 브라우저에서 수정된 포트 번호 8502로 접속됨

고객 피드백 분석

← → ↺ 🏠 🌐 localhost:8502

📁 데이터 업로드

CSV 또는 Excel 파일을 업로드하세요

Drag and drop file here

Limit 200MB per file • CSV, XLSX, XLS

Browse files

📌 샘플 데이터 사용

고객 피드백 분석 대시보드

샘플 데이터를 사용하고 있습니다.

데이터 미리보기

	date	product	category	feedback_text
0	2024-01-15	모바일 앱	기능	앱이 너무 느려서 사용하기 어렵습니다
1	2024-01-16	웹사이트	UI	디자인이 깔끔하고 사용하기 편합니다
2	2024-01-17	모바일 앱	버그	로그인할 때 오류가 발생합니다
3	2024-01-18	웹사이트	기능	검색 기능이 정말 유용합니다
4	2024-01-19	모바일 앱	성능	배터리 소모가 너무 많습니다

- 앱 실행 중 키워드 분석 부분에 오류 발생

🔍 키워드 분석

LookupError: Resource +[93mpunkt_tab~[0m not found. Please use the NLTK Downloader to obtain the resource: ~[31m>>> import nltk>>> nltk.download('punkt_tab') ~[0m For more information see: <https://www.nltk.org/data.html> Attempted to load ~[93mtokenizers/punkt_tab/english/~[0m Searched in: - 'C:\\Users\\csw\\AppData\\Local\\Programs\\Python\\Python313\\nltk_data' - 'C:\\Users\\csw\\AppData\\Local\\Programs\\Python\\Python313\\share\\nltk_data' - 'C:\\Users\\csw\\AppData\\Local\\Programs\\Python\\Python313\\lib\\nltk_data' - 'C:\\Users\\csw\\AppData\\Roaming\\nltk_data' - 'C:\\nltk_data' - 'D:\\nltk_data' - 'E:\\nltk_data'

Traceback:

```
File "E:\\cursor\\CFA\\app.py", line 402, in <module>
  main()
  ~~~~
File "E:\\cursor\\CFA\\app.py", line 236, in main
  keywords = extract_keywords(df['feedback_text'])
File "E:\\cursor\\CFA\\app.py", line 103, in extract_keywords
  processed_text = preprocess_text(text)
File "E:\\cursor\\CFA\\app.py", line 67, in preprocess_text
  tokens = word_tokenize(text)
File "C:\\Users\\csw\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages\\nltk\\tokenize\\__init__.py", line 142, in word_tokenize
  sentences = [text] if preserve_line else sent_tokenize(text, language)
  ~~~~~~
File "C:\\Users\\csw\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages\\nltk\\tokenize\\__init__.py", line 119, in sent_tokenize
  tokenizer = _get_punkt_tokenizer(language)
File "C:\\Users\\csw\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages\\nltk\\tokenize\\__init__.py", line 105, in _get_punkt_tokenizer
  return PunktTokenizer(language)
File "C:\\Users\\csw\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages\\nltk\\tokenize\\punkt.py", line 1744, in __init__
  self.load_lang(lang)
  ~~~~~~
File "C:\\Users\\csw\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages\\nltk\\tokenize\\punkt.py", line 1749, in load_lang
  lang_dir = find(f'tokenizers/punkt_tab/{lang}/')
File "C:\\Users\\csw\\AppData\\Local\\Programs\\Python\\Python313\\Lib\\site-packages\\nltk\\data.py", line 579, in find
  raise LookupError(resource_not_found)
```

- 오류 사항을 AI에 전달

```

File "C:\Users\csw\AppData\Local\Programs\Python\Python313\Lib\site-packages\nltk\tokenize\_init_.py", line 105, in
_get_punkt_tokenizer
    return PunktTokenizer(language)
File "C:\Users\csw\AppData\Local\Programs\Python\Python313\Lib\site-packages\nltk\tokenize\punkt.py", line 1744, in
_init_
    self.load_lang(lang)
    ~~~~~^~~~~~
File "C:\Users\csw\AppData\Local\Programs\Python\Python313\Lib\site-packages\nltk\tokenize\punkt.py", line 1749, in
load_lang
    lang_dir = find(f"tokenizers/punkt_tab/{lang}/")
File "C:\Users\csw\AppData\Local\Programs\Python\Python313\Lib\site-packages\nltk\data.py", line 579, in find
    raise LookupError(resource_not_found) 다른 부분은 정상인데 키워드 분석 부분에서 이런 오류가 나는데?

```

- 오류 수정 후 앱 재실행

NLTK의 `punkt_tab` 리소스가 없어서 발생하는 오류입니다. 한국어 텍스트 처리에 더 적합한 방법으로 키워드 분석 부분을 수정하겠습니다.

```
app.py +2 -2
```

이제 NLTK import도 제거하고 더 간단하게 만들어보겠습니다:

```
app.py +1 -3
```

NLTK 데이터 다운로드 부분도 제거하겠습니다:

```
app.py +1 -9
```

이제 앱을 다시 실행해보겠습니다:

```
streamlit run app.py
```

Use Allowlist

Run this command? Stop Ctrl+Shift+⌘

Run

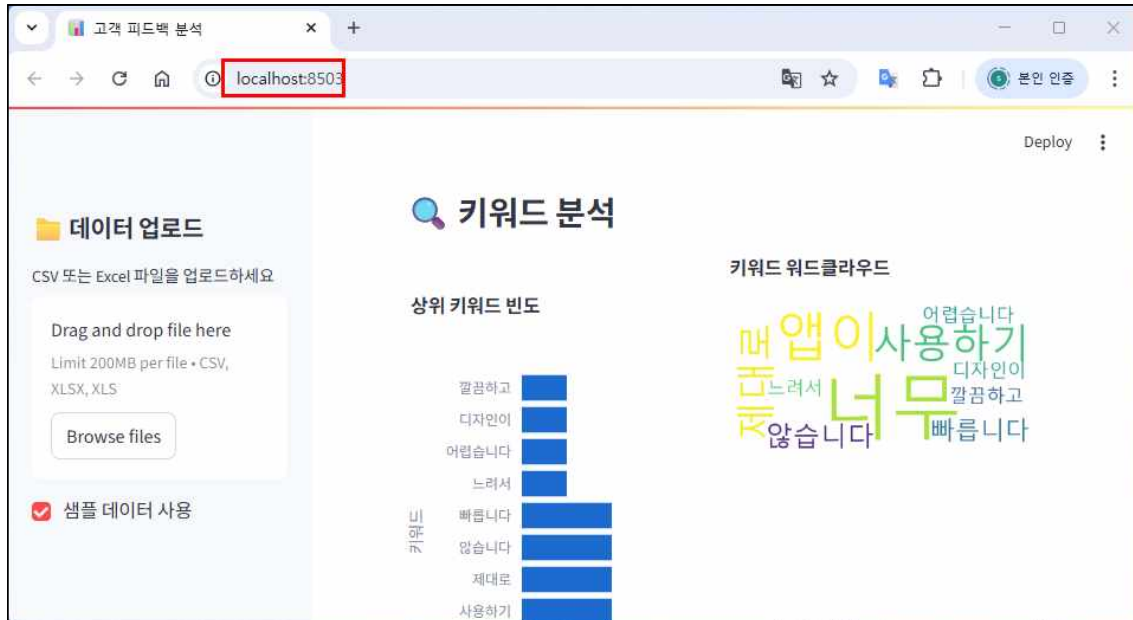
- 수정된 앱 재실행됨

- ◆◆ 수정된 내용:
 1. NLTK 의존성 제거: `word_tokenize` 함수 대신 간단한 `split()` 사용
 2. 불필요한 import 제거: NLTK 관련 모듈들 제거
 3. 데이터 다운로드 코드 제거: NLTK 데이터 다운로드 부분 제거
- 💡 변경 이유:
 - 한국어 텍스트 처리: 한국어는 공백 기반 토큰화로도 충분히 처리 가능
 - 의존성 최소화: 복잡한 NLTK 토큰라이저 없이도 키워드 추출 가능
 - 성능 향상: 더 빠르고 안정적인 텍스트 처리

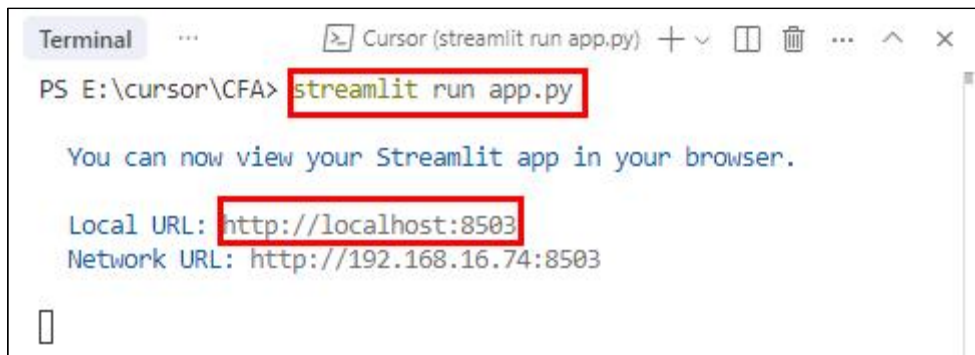
이제 앱이 정상적으로 실행되며, 키워드 분석 기능도 오류 없이 작동할 것입니다!

접속 URL: <http://localhost:8501> (또는 터미널에 표시된 포트)

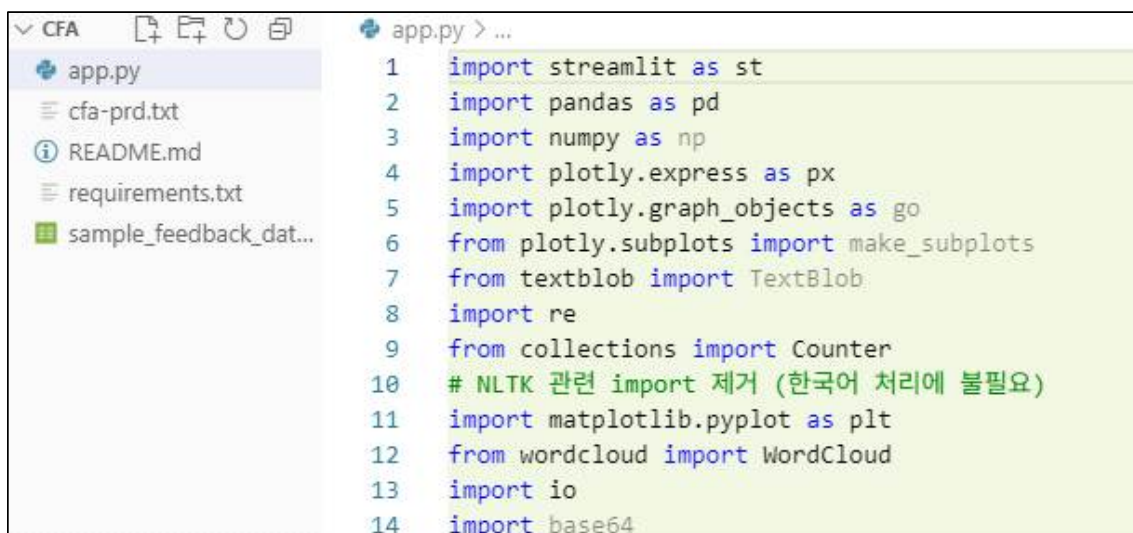
- 브라우저에서 수정된 포트 번호 8503으로 접속됨



- 터미널에서 앱 실행 확인



- 개발된 메인 코드 app.py 확인



- 앱 개발에 필요한 패키지 목록 파일 requirements.txt 확인

```

1 streamlit>=1.29.0
2 pandas>=2.1.0
3 numpy>=1.26.0
4 plotly>=5.17.0
5 textblob>=0.17.1
6 wordcloud>=1.9.2
7 matplotlib>=3.7.0
8 seaborn>=0.12.0
9 scikit-learn>=1.3.0
10 nltk>=3.8.1
11 openpyxl>=3.1.0

```

- 앱 개요 및 사용법 설명 문서 파일 README.md 확인

```

1 # 🇰🇷 고객 피드백 분석 대시보드
2
3 PRD 문서에 명시된 요구사항에 따라 개발된 고객 피드백 분석
  웹 애플리케이션입니다.
4
5 ## 🚀 주요 기능
6
7 ### 핵심 기능
8 - **텍스트 감성 분석**: 피드백 텍스트에서 긍정, 부정,
  중립 감성 분류
9 - **이슈 키워드 추출**: 자주 언급되는 주요 키워드 추출 및
  시각화
10 - **분석 결과 시각화**: 감성 분포 및 키워드 빈도 차트로
  제공
11 - **데이터 업로드**: CSV 또는 Excel 파일 업로드 기능 지원
12

```

- 앱 테스트 샘플 데이터 파일 sample_feedback_data.csv 확인

```

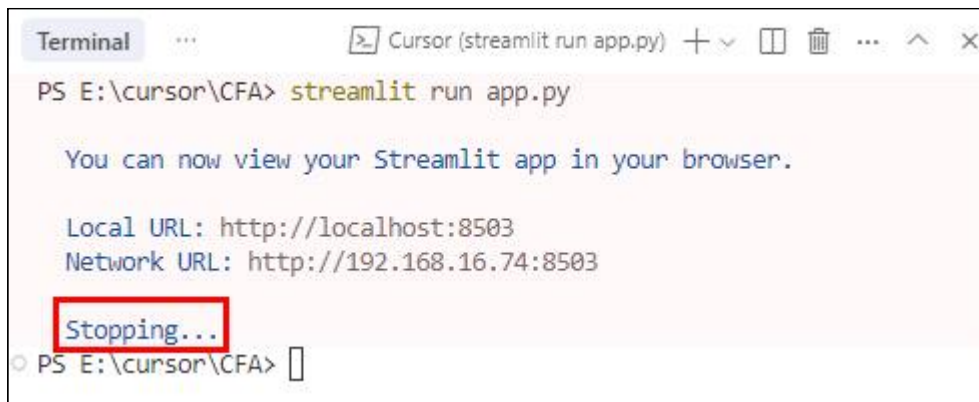
1 date,product,category,feedback_text,rating
2 2024-01-15,모바일앱,기능,앱이 너무 느려서 사용하기 어렵습니다,2
3 2024-01-16,웹사이트,UI,디자인이 깔끔하고 사용하기 편합니다,5
4 2024-01-17,모바일앱,버그,로그인할 때 오류가 발생합니다,1
5 2024-01-18,웹사이트,기능,검색 기능이 정말 유용합니다,5
6 2024-01-19,모바일앱,성능,배터리 소모가 너무 많습니다,2
7 2024-01-20,웹사이트,UI,색상이 너무 밝아서 눈이 아픕니다,3
8 2024-01-21,모바일앱,기능,푸시 알림이 제대로 작동하지 않습니다,2
9 2024-01-22,웹사이트,기능,결제 과정이 간단하고 빠릅니다,5
10 2024-01-23,모바일앱,UI,메뉴 구조가 직관적입니다,4
11 2024-01-24,웹사이트,성능,페이지 로딩 속도가 빠릅니다,5
12 2024-01-25,모바일앱,버그,앱이 자주 충돌합니다,1
13 2024-01-26,웹사이트,기능,고객 지원이 친절합니다,5
14 2024-01-27,모바일앱,UI,폰트 크기가 너무 작습니다,3
15 2024-01-28,웹사이트,기능,개인정보 설정이 편리합니다,4
16 2024-01-29,모바일앱,성능,메모리 사용량이 적습니다,5

```

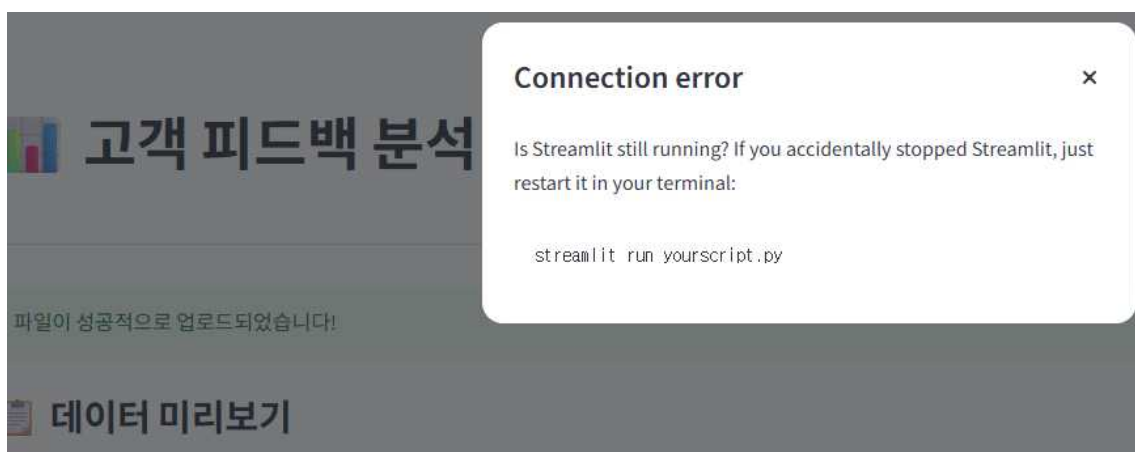

- 샘플 파일 업로드 및 앱 테스트



- 앱 실행 제거
- 앱 실행 터미널에서 CTRL + C 입력



- 앱 연결 중단



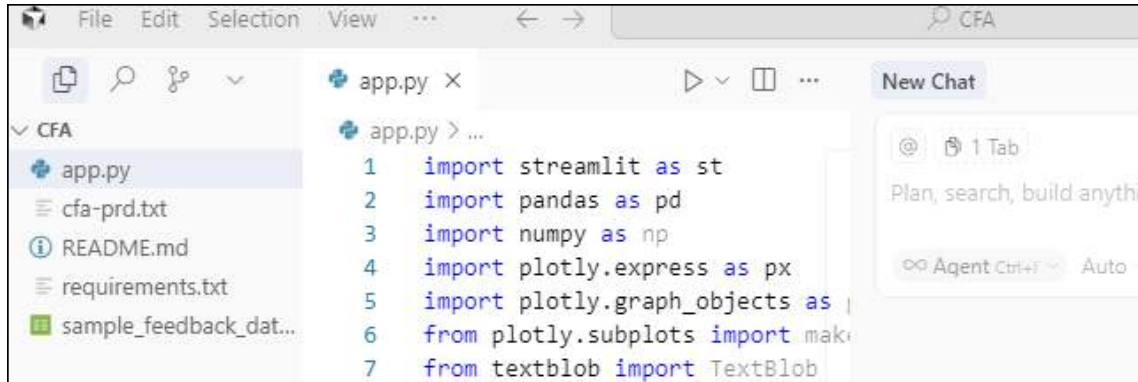
- 커서 종료

04 앱 배포 - 스트림릿 클라우드

1. 앱 배포 방법

■ 커서 재시작

- 기존 프로젝트 CFA에서 자동 시작



- AI 대화창 초기화: Clear All chats



■ AI 대화창에서 앱 배포 방법 확인

- 현재 프로젝트 폴더의 앱을 깃을 통해 스트림릿 클라우드에
- 배포하는 방법을 단계별로 설명해줘



- 프로젝트 구조 확인



- 스트림릿 클라우드 배포 단계별 가이드
 - 1단계: 깃 저장소 준비

Streamlit Cloud 배포 단계별 가이드

1단계: Git 저장소 준비

먼저 현재 프로젝트를 Git 저장소로 초기화하고 GitHub에 푸시해야 합니다.

```
$ bash

# Git 초기화
git init

# 파일 추가
git add .

# 첫 커밋
git commit -m "Initial commit: Customer Feedback Analysis App"

# GitHub 원격 저장소 추가 (GitHub에서 새 저장소 생성 후)
git remote add origin https://github.com/사용자명/저장소명.git

# 메인 브랜치로 푸시
git branch -M main
git push -u origin main
```

- 2단계 스트림릿 클라우드 설정 파일 생성

2단계: Streamlit Cloud 설정 파일 생성

Streamlit Cloud에서 앱을 인식할 수 있도록 `streamlit_app.py` 파일을 생성하거나 기존 `app.py` 를 리 네임해야 합니다.

streamlit_app.py (new) +292 •

이제 Streamlit Cloud 배포를 위한 추가 설정 파일들을 생성하겠습니다.

config.toml (new) +14 •

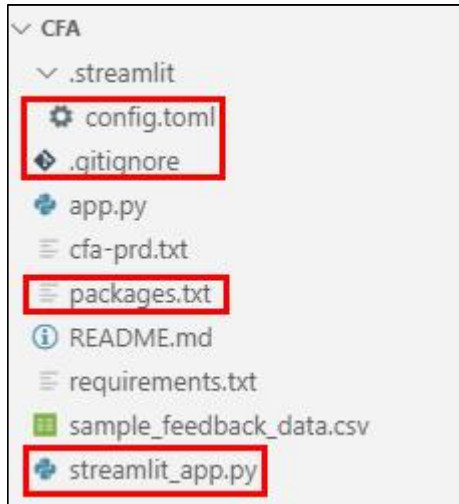
packages.txt (new) +2 •

.gitignore (new) +119 •

이제 README 파일을 업데이트하여 배포 방법을 포함하겠습니다.

README.md +108 -61 •

- 클라우드 설정 파일 생성 확인



- 스트림릿 클라우드 배포 최종 단계별 가이드

- 사전 준비사항

Streamlit Cloud 배포 완벽 가이드

사전 준비사항

1. GitHub 계정이 필요합니다
2. Streamlit Cloud 계정이 필요합니다 (무료)
3. 현재 프로젝트 폴더에 모든 필요한 파일이 준비되어 있습니다

- 1단계: 로컬 깃 저장소 초기 설정

1단계: Git 저장소 초기화

PowerShell에서 다음 명령어를 실행하세요:

```
$ bash

# Git 초기화
git init

# 모든 파일 추가
git add .

# 첫 커밋 생성
git commit -m "Initial commit: Customer Feedback Analysis App"
```

- git init: 현재 디렉토리를 로컬 깃 저장소로 초기화하고 .git 디렉토리 생성
- git add .: 현재 디렉토리의 모든 내용을(.gitignore의 내용들 제외) 스테이징 영역에 추가
- git commit -m "Initial commit: ...": 스테이징된 내용들을 커밋하며 변경사항을 기록
- "Initial Commit": 해당 커밋이 이 프로젝트의 첫 저장임을 나타냄
- 프로젝트를 깃으로 관리, 현재 디렉토리의 모든 내용을 버전 관리 대상으로 지정,
- 지금 상태를 첫 번째 저장(커밋)으로 기록

- 2단계: 원격 깃 저장소 생성

💎 2단계: GitHub 저장소 생성

1. GitHub에 로그인
2. "New repository" 클릭
3. 저장소 이름 입력 (예: `customer-feedback-analysis`)
4. Public 또는 Private 선택
5. "Create repository" 클릭

- 3단계: 원격 깃 저장소 연결 및 푸시

```
# 원격 저장소 추가 (URL을 실제 저장소 URL로 변경)
git remote add origin https://github.com/사용자명/저장소명.git

# 메인 브랜치로 설정
git branch -M main

# GitHub에 푸시
git push -u origin main
```

- 4단계: 스트림릿 클라우드에 앱 배포

☁ 4단계: Streamlit Cloud 배포

1. Streamlit Cloud 접속
2. GitHub 계정으로 로그인
3. "New app" 클릭
4. 설정:
 - Repository: 방금 생성한 GitHub 저장소 선택
 - Branch: `main`
 - Main file path: `streamlit_app.py`
5. "Deploy!" 클릭

- 5단계: 앱 배포 완료 및 앱 사용

🕒 5단계: 배포 완료 대기

- 배포 과정은 보통 2-5분 소요
- 배포가 완료되면 자동으로 URL 생성
- URL을 통해 앱에 접근 가능

- 배포 파일들

💎 현재 프로젝트에 포함된 배포 파일들

- ✓ `streamlit_app.py`: Streamlit Cloud가 인식하는 메인 앱 파일
- ✓ `.streamlit/config.toml`: Streamlit 설정 및 테마
- ✓ `packages.txt`: 한글 폰트 등 시스템 패키지
- ✓ `requirements.txt`: Python 의존성 패키지
- ✓ `.gitignore`: Git에서 제외할 파일들
- ✓ `README.md`: 상세한 배포 가이드

2. 로컬 깃 저장소 초기화

■ 로컬 깃 초기화

- 현재 프로젝트 폴더 내용 확인

```
PS E:\cursor\CFA> ls

디렉터리: E:\cursor\CFA

Mode                LastWriteTime         Length Name
----                -
d-----          2025-08-20 오전 10:45             .streamlit
-a-----          2025-08-20 오전 10:46             1273 .gitignore
-a-----          2025-08-20 오전 10:35          13264 app.py
-a-----          2025-08-19 오전 11:30           2463 cfa-prd.txt
-a-----          2025-08-20 오전 10:46              33 packages.txt
-a-----          2025-08-20 오전 10:46           4087 README.md
-a-----          2025-08-20 오전 10:34             177 requirements.txt
-a-----          2025-08-20 오전 10:34           1249 sample_feedback_data.csv
-a-----          2025-08-20 오전 10:46          11781 streamlit_app.py
```

- git init: 현재 폴더를 로컬 깃 저장소로 설정
- git config: 커밋할 사용자 정보 설정
- git add . : 현재 디렉토리(.)의 모든 파일들을 스테이징 영역에 추가(저장 예정 선언)
- git commit -m : 스테이징된 파일들을 로컬 저장소에 커밋(실제 저장 수행)

```
PS E:\cursor\CFA> git init
Reinitialized existing Git repository in E:/cursor/CFA/.git/
PS E:\cursor\CFA> git config user.name "CSW"
PS E:\cursor\CFA> git config user.email "csw@csw.com"
PS E:\cursor\CFA> git add .
warning: in the working copy of 'app.py', LF will be replaced by CRLF the next time Git
warning: in the working copy of 'requirements.txt', LF will be replaced by CRLF the next
ches it
PS E:\cursor\CFA> git commit -m "Initial commit: Customer Feedback Analysis App"
[master (root-commit) b847452] Initial commit: Customer Feedback Analysis App
9 files changed, 1054 insertions(+)
create mode 100644 .gitignore
create mode 100644 .streamlit/config.toml
create mode 100644 README.md
create mode 100644 app.py
create mode 100644 cfa-prd.txt
create mode 100644 packages.txt
create mode 100644 requirements.txt
create mode 100644 sample_feedback_data.csv
create mode 100644 streamlit_app.py
PS E:\cursor\CFA>
```

- 로컬 깃 저장소 상태 확인

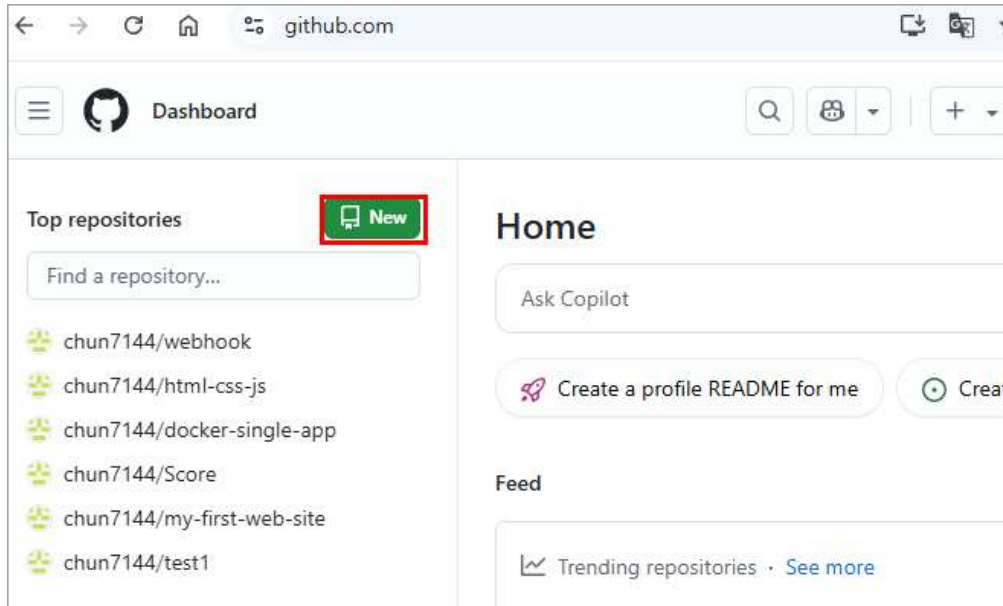
```
PS E:\cursor\CFA> git status
On branch master
nothing to commit, working tree clean
PS E:\cursor\CFA>
```

- 현재 master 브랜치에서 작업 중
- 커밋할 변경 사항 없고(이전 커밋 이후), 현재 작업 디렉토리 상태가 깃과 완전히 일치함

3. 원격 깃 저장소 생성

■ 깃허브 로그인

- 깃허브에 로그인: github.com
- 'New': 새 프로젝트를 위한 깃허브 저장소 생성



■ 저장소 설정

- 새 저장소 이름: CFA
- 사용 가능한(available) 이름 사용

A screenshot of the 'Create a new repository' form on GitHub. The form has a title 'Create a new repository' with links for 'Preview' and 'Switch back to classic experience'. Below the title is a paragraph: 'Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository.](#) Required fields are marked with an asterisk (*).' The form is divided into sections, with the first section being '1 General'. Under 'General', there is an 'Owner *' field with a dropdown menu showing 'chun7144'. To the right of the owner is a 'Repository name *' field with a text input containing 'CFA'. This field is highlighted with a red box. Below the input, there is a green checkmark and the text 'CFA is available.' At the bottom of the form, there is a note: 'Great repository names are short and memorable. How about [solid-rotary-phone](#)?'.

- 새 저장소 공개: Public
- MIT 라이선스 선택: 일반적이고, 상업적 이용도 가능, 수정 및 재배포 자유

2 Configuration

Choose visibility *
Choose who can see and commit to this repository

Public

Add README
READMEs can be used as longer descriptions. [About READMEs](#)

Off ☐

Add .gitignore
.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

Add license
Licenses explain how others can use your code. [About licenses](#)

MIT License

Create repository

- 저장소 생성 선택(Create repository)

CFA Public
Pin Watch 0

main Go to file Code

chun7144 Initial commit

LICENSE

README MIT license

Local **Codespaces**

Clone

HTTPS SSH GitHub CLI

https://github.com/chun7144/CFA.git

Clone using the web URL.

Open with GitHub Desktop

Download ZIP

- main 브랜치에 CFA 저장소 생성됨
- CFA 저장소 주소: https://github.com/chun7144/CFA.git
- Code 클릭 후 CFA 저장소 주소 URL 복사

4. 원격 깃 저장소 연결 및 푸시

■ 원격 저장소 연결 및 브랜치 이름 설정

- git add origin Git저장소_URL: 원격 저장소 연결
- git branch: 로컬 저장소의 브랜치 이름을 master에서 main으로 변경
- 일관성 유지를 위해 원격 저장소의 브랜치 이름에 맞추어 변경

```
PS E:\cursor\CFA> git remote add origin https://github.com/chun7144/CFA.git
PS E:\cursor\CFA> git branch
* master
PS E:\cursor\CFA> git branch -m master main
PS E:\cursor\CFA> git branch
* main
PS E:\cursor\CFA>
```

■ 원격 커밋을 로컬 커밋에 가져와 병합

- git pull origin main --rebase : 원격의 변경사항을 로컬에 가져와 병합
- 현재 메인 브랜치에서 작업중임

```
PS E:\cursor\CFA> git pull origin main --rebase
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 1.44 KiB | 19.00 KiB/s, done.
From https://github.com/chun7144/CFA
 * branch                main          -> FETCH_HEAD
 * [new branch]          main          -> origin/main
Successfully rebased and updated refs/heads/main.
PS E:\cursor\CFA> git status
On branch main
nothing to commit, working tree clean
PS E:\cursor\CFA>
```

■ 병합된 커밋을 푸시

- git push origin main
- 원격 저장소 github.com/chun7144/CFA.git의 main 브랜치에 푸시됨

```
PS E:\cursor\CFA> git push origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 6 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (12/12), 12.38 KiB | 3.09 MiB/s, done.
Total 12 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/chun7144/CFA.git
bf9fee7..9c35514 main -> main
PS E:\cursor\CFA>
```

- git push 명령 실행 시 Username/Password 요청 시
- Username에 사용자명, Password에 발급 받은 PAT 토큰 입력(복붙)

- git push 명령 실행 시 Username/Password 요청없이 진행되는 경우
- 기존에 윈도우즈 인증 관리자 등에 저장된 값들 사용하여 인증된 것임

```
PS E:\cursor\CFA> git config --show-origin credential.helper
file:C:/Program Files/Git/etc/gitconfig manager
PS E:\cursor\CFA> █
```

- 인증 정보가 윈도우즈의 인증 관리자에 저장됨
- 원격 저장소 푸시 확인
 - 로컬 저장소의 파일들과 폴더가 원격 저장소에 푸시됨
 - 하단부에 README 파일 내용 확인됨

CFA Public

main 1 Branch 0 Tags

Go to file Add file Code

CSW Initial commit: Customer Feedback Analysis App 9c35514 · 7 minutes ago 2 Commits

File	Commit	Time
.streamlit	Initial commit: Customer Feedback Analysis App	7 minutes ago
.gitignore	Initial commit: Customer Feedback Analysis App	7 minutes ago
LICENSE	Initial commit	36 minutes ago
README.md	Initial commit: Customer Feedback Analysis App	7 minutes ago
app.py	Initial commit: Customer Feedback Analysis App	7 minutes ago
cfa-prd.txt	Initial commit: Customer Feedback Analysis App	7 minutes ago
packages.txt	Initial commit: Customer Feedback Analysis App	7 minutes ago
requirements.txt	Initial commit: Customer Feedback Analysis App	7 minutes ago
sample_feedback_data.csv	Initial commit: Customer Feedback Analysis App	7 minutes ago
streamlit_app.py	Initial commit: Customer Feedback Analysis App	7 minutes ago

README MIT license

고객 피드백 분석 대시보드

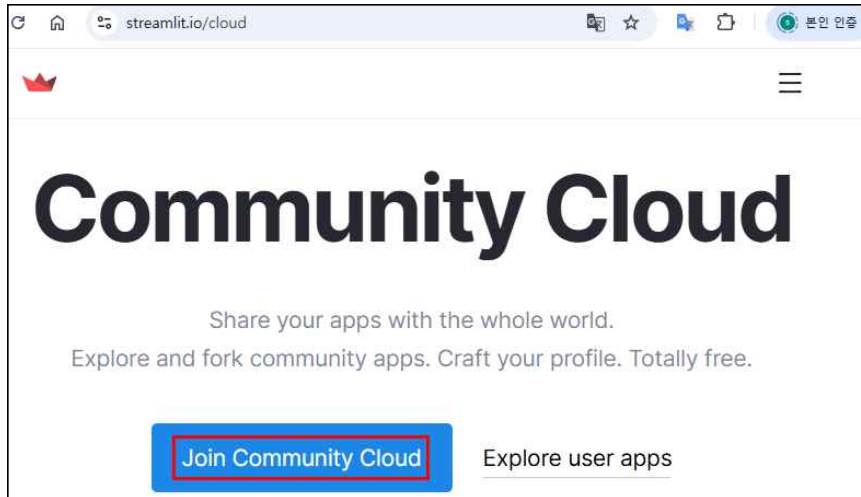
고객 피드백 데이터를 분석하여 감성 분석, 키워드 추출, 워드클라우드 등을 제공하는 Streamlit 웹 애플리케이션입니다.

주요 기능

- 파일 업로드: CSV, Excel 파일 지원
- 감성 분석: 텍스트의 긍정/부정/중립 감성 분석
- 키워드 추출: 자주 등장하는 단어 분석
- 워드클라우드: 시각적 키워드 표현
- 텍스트 길이 분석: 통계적 분석
- 결과 다운로드: 분석 결과 CSV 파일 다운로드

5. 스트림릿 클라우드에 앱 배포

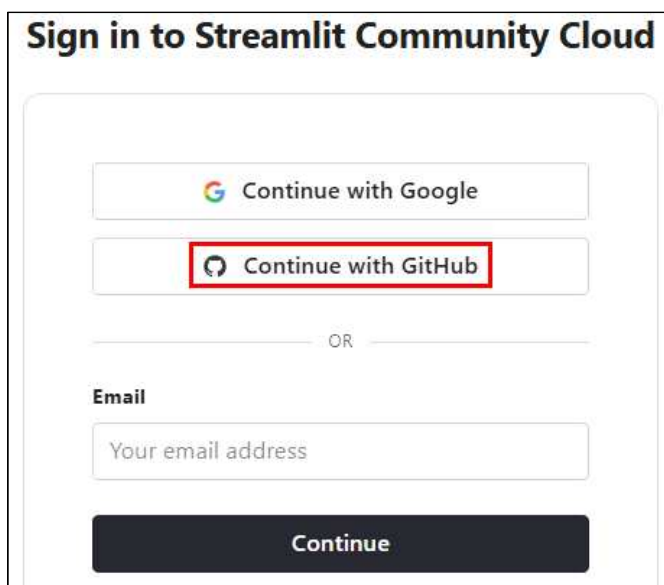
- 스트림릿 클라우드 접속
 - <https://streamlit.io/cloud>
 - 클라우드 커뮤니티 조인



- 로그인 진행



- 깃허브로 진행



Complete the code challenge

Enter the code sent to **wintari@naver.com**

- 스트림릿 클라우드에서 이메일로 보내준 임시 코드 확인 후 입력

Complete code challenge

We've issued a one-time code challenge as an additional security measure. Please enter the code below in your open browser window to continue authentication.

- 깃허브 계정으로 스트림릿 클라우드에 연결됨

chun7144 ▾

[My apps](#)
[My profile](#)
[Explore](#)
[Discuss ↗](#)

[Create app](#)

chun7144's apps


20250723prd · main · app.py

score · main · app.py

- 우측 상단 “Create app” 또는 하단 “My new app” 선택하여
- 앱 배포 진행

- 새로운 앱 생성
 - 깃허브로부터 공공 앱 배포 선택


What would you like to do?



Deploy a public app from GitHub

My code is ready on a GitHub repo, and it is totally awesome.


[Deploy now](#)



Deploy a public app from a template

I want to see what kind of amazing concoctions you have for me.

[Check out templates](#)



Deploy a private app in Snowflake

I want unlimited enterprise-grade apps, with the security of Snowflake.

[Start trial →](#)

- 앱 배포 설정

Deploy an app

Repository [Paste GitHub URL](#)

`chun7144/CFA`

Branch

`main`

Main file path

`streamlit_app.py`

App URL (optional)

`my-cfa-app` `.streamlit.app`

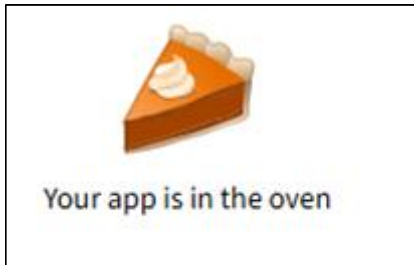
Domain is available

[Advanced settings](#)

[Deploy](#)

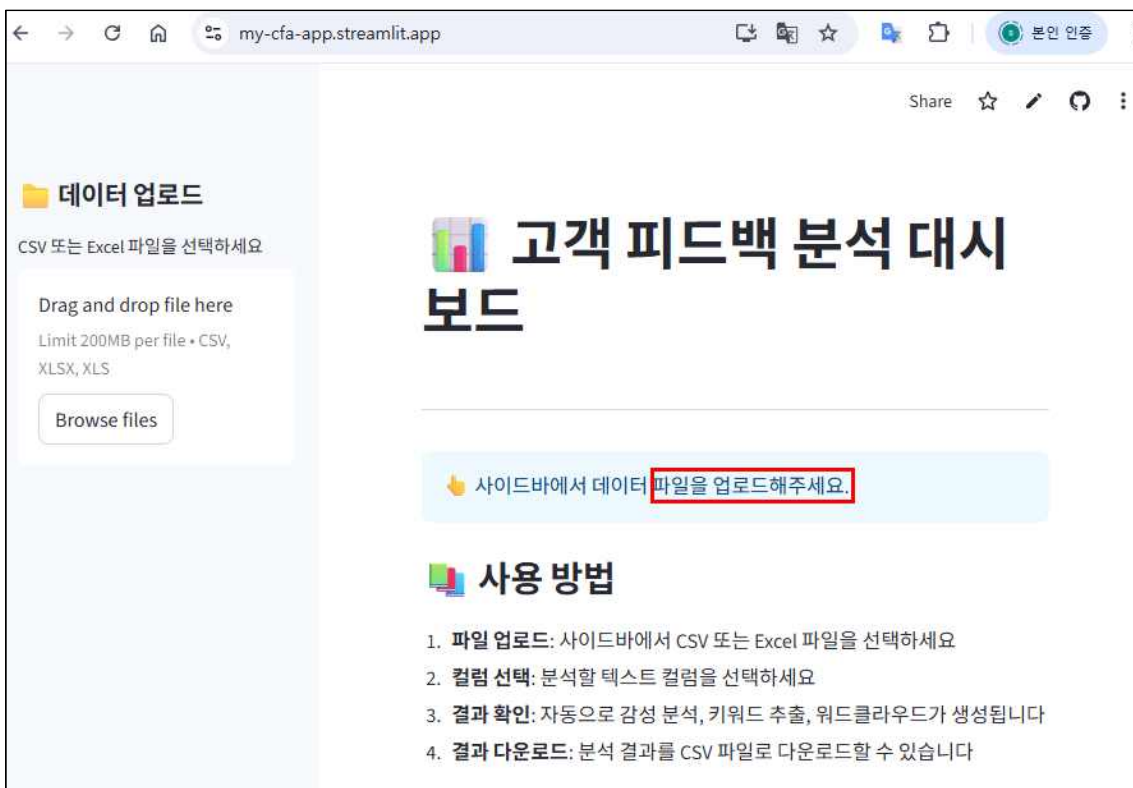
- 깃허브 저장소 참고하여 저장소, 브랜치, 메인 파일 등 설정
- App URL (optional): 생성할 앱의 URL 주소 설정
- 디플로이 선택

- 앱 생성 진행

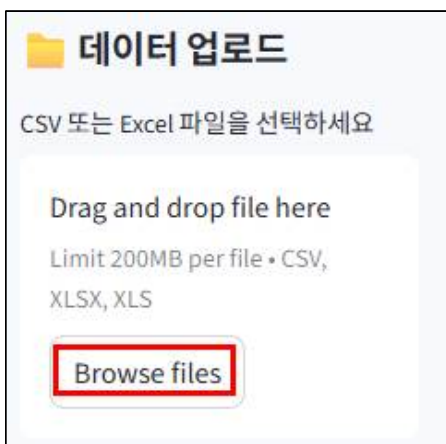


■ 앱 배포 완료

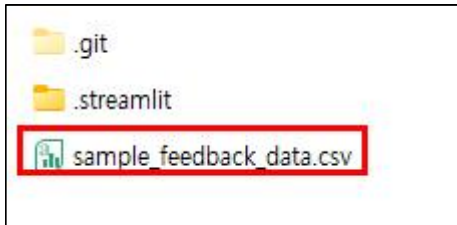
- <https://my-cfa-app.streamlit.app/>
- 깃허브에 있는 앱을 스트림릿 클라우드에 성공적으로 배포 완료됨



- 파일 업로드



- 샘플 파일 선택



- 고객 피드백 분석 결과 확인

데이터 업로드

CSV 또는 Excel 파일을 선택하세요

Drag and drop file here

Limit 200MB per file • CSV, XLSX, XLS

Browse files

sample_feedb...
1.2KB

분석할 텍스트 컬럼을 선택하세요

date

고객 피드백 분석 대시보드

sample_feedback_data.csv 파일이 성공적으로 로드되었습니다!

데이터 미리보기

	date	product	category	feedback_text	rat
0	2024-01-15	모바일앱	기능	앱이 너무 느려서 사용하기 어렵습니다	
1	2024-01-16	웹사이트	UI	디자인이 깔끔하고 사용하기 편합니다	
2	2024-01-17	모바일앱	버그	로그인할 때 오류가 발생합니다	
3	2024-01-18	웹사이트	기능	검색 기능이 정말 유용합니다	
4	2024-01-19	모바일앱	성능	배터리 소모가 너무 많습니다	

