

## Chapter 1. 서론

- Chat GPT란 무엇인가?
- Chat GPT를 활용한 웹 개발의 장점과 특징

## Chapter 2. 개발 환경설정

- Python 설치 및 환경변수 설정
- vscode를 활용한 개발 환경 설정
- Django 설치
- DataBase

## Chapter 3. 파이썬 Django 투두리스트 웹 제작

- 완성본 미리보기
- Chat GPT를 활용한 개발방법 및 유의사항
- 프롬프트(prompt)
- Chat GPT 활용 개발
- Chat GPT 활용 버그 및 기능 수정
- 완성본 소스코드

## Chapter 4. 결론

- Chat GPT를 활용한 웹 개발의 장점과 한계
- Chat GPT의 전망과 적응 방법
- 다음 단계로 나아갈 방법 제시

## Chapter 1. 서론

Chat GPT란 무엇인가?

ChatGPT는 OpenAI에서 개발한 자연어 처리를 위한 딥러닝 알고리즘 중 하나입니다. GPT는 "Generative Pre-trained Transformer"의 약자로, 이전에 대규모 데이터셋에서 학습된 모델을 기반으로 자연어 생성을 수행하는 딥러닝 모델을 의미합니다. 즉, Chat GPT는 대화형 인공지능 모델로, 이전에 수많은 대화 데이터를 학습하여 인간과 자연스러운 대화를 수행할 수 있습니다. ChatGPT는 이전에는 대화형 인공지능에 한계를 가지고 있었던 "단어 간 연관성"과 "상황 판단"을 이해하는 능력을 보유하고 있습니다. 이러한 능력을 통해 ChatGPT는 다양한 언어와 주제에 대한 대화를 지속적으로 발전시키고 있으며, 현재는 대화뿐만 아니라 문장 생성, 요약, 번역 등 다양한 자연어 처리 분야에서 활용될 수 있습니다. 웹 개발에서도 ChatGPT를 활용하여 웹 사이트 내의 자연어 인터페이스를 개발할 수 있습니다. ChatGPT를 활용한 웹 개발은 사용자와 자연스러운 대화를 수행하며, 보다 높은 사용자 경험을 제공할 수 있습니다. 또한, ChatGPT를 활용하면 웹 개발자가 직접 대화형 인공지능 모델을 개발할 필요 없이, 미리 학습된 모델을 활용하여 웹 개발의 생산성을 높일 수 있습니다.

Chat GPT를 활용한 웹 개발의 장점과 특징

### 1. 개발 생산성 향상

- Chat GPT를 이용하여 대화형 인공지능 모델을 직접 개발할 필요 없이, 미리 학습된 모델을 활용하여 웹 개발을 수행할 수 있습니다. 이는 개발 생산성을 높이고, 웹 사이트 개발 기간을 단축할 수 있습니다. 기존에 개발을 하며 디버깅을 하고 오류를 수정하고 신규 기능을 개발하기 위해 검색하던 시간을 Chat GPT를 통해 단축할 수 있습니다.

### 2. 유연한 개발 환경

- Chat GPT는 다양한 언어와 주제에 대한 대화를 지속적으로 발전시킬 수 있습니다. 이는 다양한 웹 사이트에서 활용 가능하며 웹 사이트의 분야나 종류에 상관없이 자연스러운 대화형 인터페이스를 구현할 수 있습니다.

### 3. 적극적인 활용 사례

- Chat GPT는 이미 다양한 분야에서 활용되고 있습니다. 예를 들어 챗봇, 텍스트 요약, 번역 등에서 활용되며 웹 개발 분야에서도 적극적으로 활용되고 있습니다. 많은 레퍼런스들을 통해 웹 개발을 보다 빠르고 정확하게 할 수 있다는 장점이 있습니다.

### 4. 높은 확장성

- Chat GPT는 다양한 딥러닝 기술과 연계하여 확장성을 높일 수 있습니다. 예를 들어 Chat GPT와 이미지 분석 기술을 연계하여 이미지 인식 대화형 인터페이스를 개발할 수 있습니다.

이러한 장점과 특징을 통해 Chat GPT를 활용한 웹 개발은 사용자와 자연스러운 대화를 수행하는 인공지능 모델을 개발하는 것이 아닌 이미 학습된 모델을 활용하여 웹 사이트의 사용자 경험을 개선하는 것을 목적으로 구현할 수 있습니다. 따라서 Chat GPT를 활용한 웹 개발은 개발 생산성의 향상과 함께 사용자 경험을 향상시키고 높은 확장성을 제공하는 혁신적인 방법입니다.

마지막으로 Chat GPT를 활용한 웹 개발은 인공지능 기술의 발전과 함께 더욱 발전하고 있습니다. 앞으로는 Chat GPT와 다른 딥러닝 기술들을 연계하여 보다 혁신적인 대화형 인터페이스를 개발할 수 있을 것으로 기대됩니다.

## Chapter 2. 개발 환경설정

이번 챕터에서는 mac 환경과 window 환경이 다르기 때문에 각각의 환경에 맞춰 작성했습니다.

### Python 설치 및 환경변수 설정

#### 1. python 설치

- <https://www.python.org/downloads/>
- 위 링크에서 최신버전의 파이썬을 다운받아 주세요



## 2. 환경 변수 설정

### - MAC

1. 터미널을 엽니다.
2. 다음 명령어를 입력하여 현재 설치된 파이썬 버전을 확인합니다.(만약 python 명령어가 제대로 동작하지않는 경우 alias가 python3로 되어있어 발생하는 것이므로 python3 --version 으로 명령어를 입력하면 됩니다.)

```
python --version
```

3. 다음 명령어를 입력하여 현재 로그인한 사용자의 홈 디렉토리로 이동합니다.

```
cd ~
```

4. 다음 명령어를 입력하여 사용자의 프로필 파일을 엽니다.

```
vi .bash_profile
```

5. 파일이 열리면 파일의 마지막 줄에 다음과 같이 입력합니다. 입력하는 방법은 ‘i’ 를 누르면 텍스트를 수정할 수 있으며 수정 완료 후에는 ‘:’를 누른뒤 ‘wq’ 입력 후 엔터로 저장을 합니다. 여기서 ‘{version}’은 위에서 확인한 파이썬 버전입니다.  
(3.7버전이라면 3.7을 넣어주면됩니다.)

```
export  
PATH="/Library/Frameworks/Python.framework/Versions/{version}/bin:${PATH}"
```

6. python3로 실행하던 명령어를 python으로 변경하고 싶다면 alias를 아래와 같이 추가하면 됩니다.

```
alias python='python3'
```

7. 변경 사항을 적용하기 위해 터미널에서 다음 명령어를 입력합니다.

```
source ~/.bash_profile
```

8. 파이썬 버전을 확인하여 변경 사항이 적용되었는지 확인합니다.

```
python --version
```

이제 파이썬을 실행할 때 'python' 명령어를 사용하여 실행할 수 있습니다.

- WINDOW
  - 참고로, Windows 10에서는 파이썬 설치 프로그램에서 "Add Python to PATH" 옵션을 선택하면 자동으로 환경 변수가 설정됩니다. (설치 시 적용한다면 아래와같은 복잡한과정을 거치지않아도 됩니다.)

1. 제어판을 엽니다.
2. 시스템을 클릭합니다.
3. 고급 시스템 설정을 클릭합니다.
4. 환경 변수를 클릭합니다.
5. 시스템 변수에서 'path' 변수를 찾아 편집 합니다.
6. 변수 값의 끝 부분에 다음과 같이 입력합니다. 여기서 '{version}'은 현재 설치된 파이썬 버전입니다. 새로운 값을 추가할 때는 각 항목을 세미콜론(';')으로 구분해야 합니다.

```
;C:\Python39\Scripts\
```

7. 변경 사항을 저장하고 닫습니다.
8. 명령 프롬프트를 엽니다.
9. 다음 명령어를 입력하여 변경 사항이 적용되었는지 확인합니다.

```
python --version
```

파이썬 버전 정보가 출력되면 성공적으로 환경 변수가 설정된 것입니다.

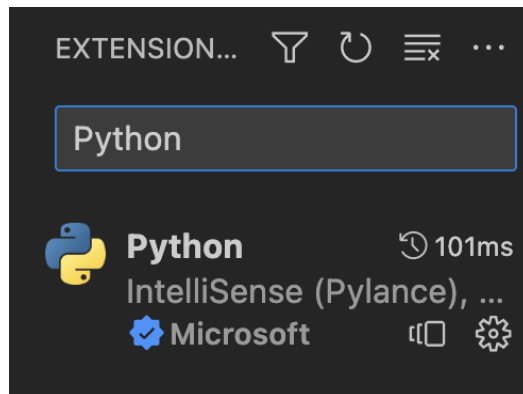
## vscode를 활용한 개발 환경 설정

### 1. vscode 설치

#### - MAC

1. 웹 브라우저에서 VS Code의 공식 웹사이트 [<https://code.visualstudio.com/>] 를 엽니다.
2. "Download for MAC" 버튼을 클릭합니다.

3. 다운로드 된 zip 파일을 더블 클릭하여 압축을 해제합니다.
4. 압축 해제된 파일을 Applications 폴더에 드래그 앤 드롭하여 VS Code를 설치합니다.
5. VS Code를 실행합니다.
6. Python 확장 프로그램을 설치합니다. VS Code에서 'Command + Shift + X'를 누르고 'Python'을 검색하여 확장프로그램을 설치합니다.

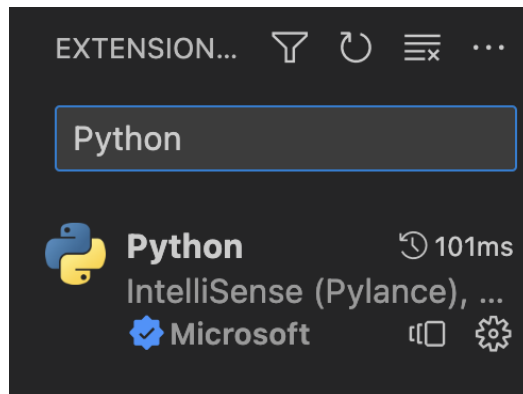


#### - WINDOW

1. 웹 브라우저에서 VS Code의 공식 웹사이트 [<https://code.visualstudio.com/>] 를 엽니다.
2. "Download for Windows" 버튼을 클릭합니다.
3. 다운로드된 exe 파일을 더블 클릭하여 실행합니다.
4. "User Account Control"창이 나타나면 "Yes"를 클릭하여 설치를 승인 합니다.
5. "Welcome" 창에서 "Next"를 클릭합니다.
6. "Destination Location" 창에서 VS Code를 설치할 경로를 지정합니다. 기본적으로 'C:\Program Files\Microsoft VS Code'에 설치됩니다. 원하는 경로를 선택한 후 "Next"를 클릭합니다.
7. "Select Start Menu Folder" 창에서 시작 메뉴에 만들 폴더 이름을 지정한 후 "Install"을 클릭합니다.



- 설치가 완료되면 “Completing the Visual Studio Code Setup Wizard”창이 나타납니다. “Finish”를 클릭하여 설치를 완료합니다.
- VS Code를 실행합니다.
- Python 확장 프로그램을 설치합니다. VS Code에서 ‘Ctrl + Shift + X’를 누르고 ‘Python’을 검색하여 확장프로그램을 설치합니다.



## Django 설치

- 터미널을 엽니다.
- pip를 최신 버전으로 업그레이드합니다.

```
pip install --upgrade pip
```

- 다음 명령어를 입력하여 Django를 설치합니다.

```
pip install Django
```

- Django가 정상적으로 설치되었는지 확인합니다.

```
python -m django --version
```

## DataBase

- Python은 SQLite를 표준 라이브러리인 sqlite3 모듈을 통해 기본 제공되며 별도로 설치하지 않아도 기본적으로 포함되어 있습니다. 따라서 SQLite에 대한 추가적인 설치나 설정 없이도 Python에서 SQLite를 사용할 수 있습니다.

## Chapter 3. 파이썬 Django 투두리스트 웹 제작

### 완성본 미리보기

기능은 투두리스트의 기본적인 CRUD 기능을 수행합니다.

CRUD란? Create, Read, Update, Delete로 웹 개발의 기본적인 기능이며 추가, 조회, 수정, 삭제입니다.



The screenshot shows a web application interface for a To-Do List. At the top, there is a navigation bar with links: "My To-Do List", "Login", and "Register". Below the navigation bar, the main content area is titled "Login". It contains two input fields: "Username" with the value "test" and "Password" with masked characters ".....". Below the password field is a blue "Login" button.

- 메인화면에서는 로그인 하지않았을 경우 투두리스트를 볼 수 없고 로그인 페이지가 먼저 띄워집니다.

## Register

Username:  Required. 150 characters or fewer. Letters, digits and @/./+/\_ only.

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:  Enter the same password as before, for verification.

[Register](#)

- 상단의 Register 버튼을 누르게 되면 회원가입 페이지로 이동됩니다.

## My To-Do List

[Add](#)

### Incomplete Tasks

책 읽기 TEST

☐ Completed

[Edit](#)

[Delete](#)

고양이 밥 주기 12시에

☐ Completed

[Edit](#)

[Delete](#)

### Completed Tasks

영어공부

☒ Completed

[Edit](#)

[Delete](#)

개발 공부하기

☒ Completed

[Edit](#)

[Delete](#)

- 회원가입을 완료하고 로그인을 하면 투두리스트의 메인화면 페이지로 이동됩니다.

# My To-Do List

Add

## Incomplete Tasks

책 읽기 TEST	<input type="checkbox"/> Completed	Edit	Delete
고양이 밥 주기 12시에	<input type="checkbox"/> Completed	Edit	Delete
블로그 글쓰기	<input type="checkbox"/> Completed	Edit	Delete

## Completed Tasks

영어공부	<input checked="" type="checkbox"/> Completed	Edit	Delete
개발 공부하기	<input checked="" type="checkbox"/> Completed	Edit	Delete

- “블로그 글쓰기” 라는 투두리스트를 추가하였습니다. 상단의 input box에 새로 추가할 리스트항목을 작성하고 우측의 Add 버튼을 누르게되면 리스트가 추가됩니다.

## Edit Task

Task:

Update

- 수정하고싶은 리스트의 Edit 버튼을 누르면 수정 페이지로 이동되며 수정하고 싶은 텍스트 입력 후 Update 버튼을 눌러 수정이 가능합니다.

Are you sure you want to delete this task?

취소

확인

## My To-Do List

New task

Add

### Incomplete Tasks

책 읽기 TEST

☐ Completed

Edit

Delete

고양이 밥 주기 12시에

☐ Completed

Edit

Delete

블로그 글쓰기 TEST

☐ Completed

Edit

Delete

### Completed Tasks

영어공부

☒ Completed

Edit

Delete

개발 공부하기

☒ Completed

Edit

Delete

- 메인 화면에서 Delete버튼을 누르면 정말 삭제할건지 확인하는 confirm 창이 뜨고 확인을 누를시 삭제가 됩니다.

# My To-Do List

Add

## Incomplete Tasks

책 읽기 TEST

☐ CompletedEditDelete

## Completed Tasks

영어공부

☒ CompletedEditDelete

고양이 밥 주기 12시에

☒ CompletedEditDelete

개발 공부하기

☒ CompletedEditDelete

- 체크박스로 구성되어있는 Completed 버튼을 누르게 되면 Incomplete Tasks에서 Completed Tasks 로 이동되며 완료된 Task 처리가 됩니다. 반대의 경우도 가능합니다.

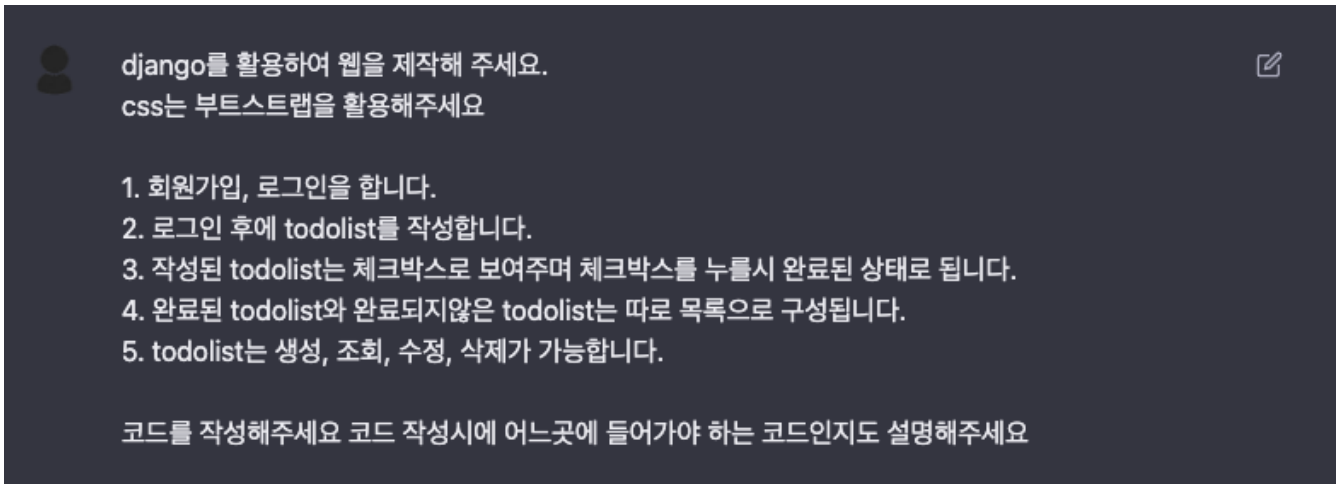
## Chat GPT를 활용한 개발방법 및 유의사항

Chat GPT를 활용한 개발은 설계 단계에서부터 기능 추가, 버그 수정 등 모든 개발과정을 Chat GPT에 질의를 통해 진행할 수 있으며 기존의 개발방식과는 다르게 마치 선배 개발자와 소통하며 개발을 하는듯한 느낌으로 Chat GPT와 대화를 통해 개발을 진행할 수 있습니다. 유의사항으로는 Chat GPT는 계속해서 학습하고있으며 질문에 따른 답변이 매번 다르기 때문에 항상 같은 코드를 작성해 주지 않습니다. 따라서 필자와 개발과정이 다를 수 있으며 완성된 결과물이 다를 수 있지만 개발하는 방식은 그대로 따라하시면 된다는 점과 완성된 소스코드를 GitHub를 통해 제공하고있으니 비교가 가능하다는점 유의 바랍니다.

## 프롬프트(prompt)

Chat GPT에서의 프롬프트란 사용자의 입력에 대한 대기 상태를 나타내며, 사용자는 프롬프트 뒤에 질문이나 문장을 입력하여 대화를 시작할 수 있습니다. 그러므로 Chat GPT를 활용한 개발을 진행할 때에는 프롬프트를 어떻게 구성하느냐에 따라 퀄리티가 달라질 수 있습니다.

## Chat GPT 활용 개발



프롬프트 예시입니다. 제작하기 원하는 웹은 파이썬 django 기반으로 동작하고 기능은 어떻게 동작해야 하는지 Chat GPT에게 설명해주고 그에따른 코드를 작성하도록 요청합니다.

여기서 css란 웹사이트의 UI를 꾸며주는 요소라고 보면됩니다. 그 중 부트스트랩 이란 비교적 쉽게 css를 구성하여 UI를 만들어주는 라이브러리 입니다.



이 요구사항에 대한 Django 웹 애플리케이션 코드를 작성해보겠습니다.



## 1. 프로젝트 생성

Django에서 새 프로젝트를 생성합니다.

bash

Copy code

```
$ django-admin startproject myproject
$ cd myproject
```

## 2. 앱 생성

Django 애플리케이션 내에서 작동하는 개별적인 구성요소인 앱을 생성합니다.

bash

Copy code

```
$ python manage.py startapp myapp
```

## 3. 모델 생성

데이터베이스 스키마를 정의하는 모델을 생성합니다.

python

Copy code

```
# myapp/models.py

from django.db import models
from django.contrib.auth.models import User

class TodoList(models.Model):
    task = models.CharField(max_length=200)
    created_by = models.ForeignKey(User, on_delete=models.CASCADE)
    is_completed = models.BooleanField(default=False)

    def __str__(self):
        return self.task
```

위 코드에서 `TodoList` 모델은 할 일(task)을 정의하고, 할 일을 만든 사용자(created\_by)와 할 일이 완료되었는지 여부(is\_completed)를 추적합니다.

## 4. URL 및 뷰 생성

URL과 뷰를 생성합니다.

python

Copy code

```
# myapp/views.py

from django.shortcuts import render, redirect
from django.contrib.auth.decorators import login_required
from .models import TodoList
```

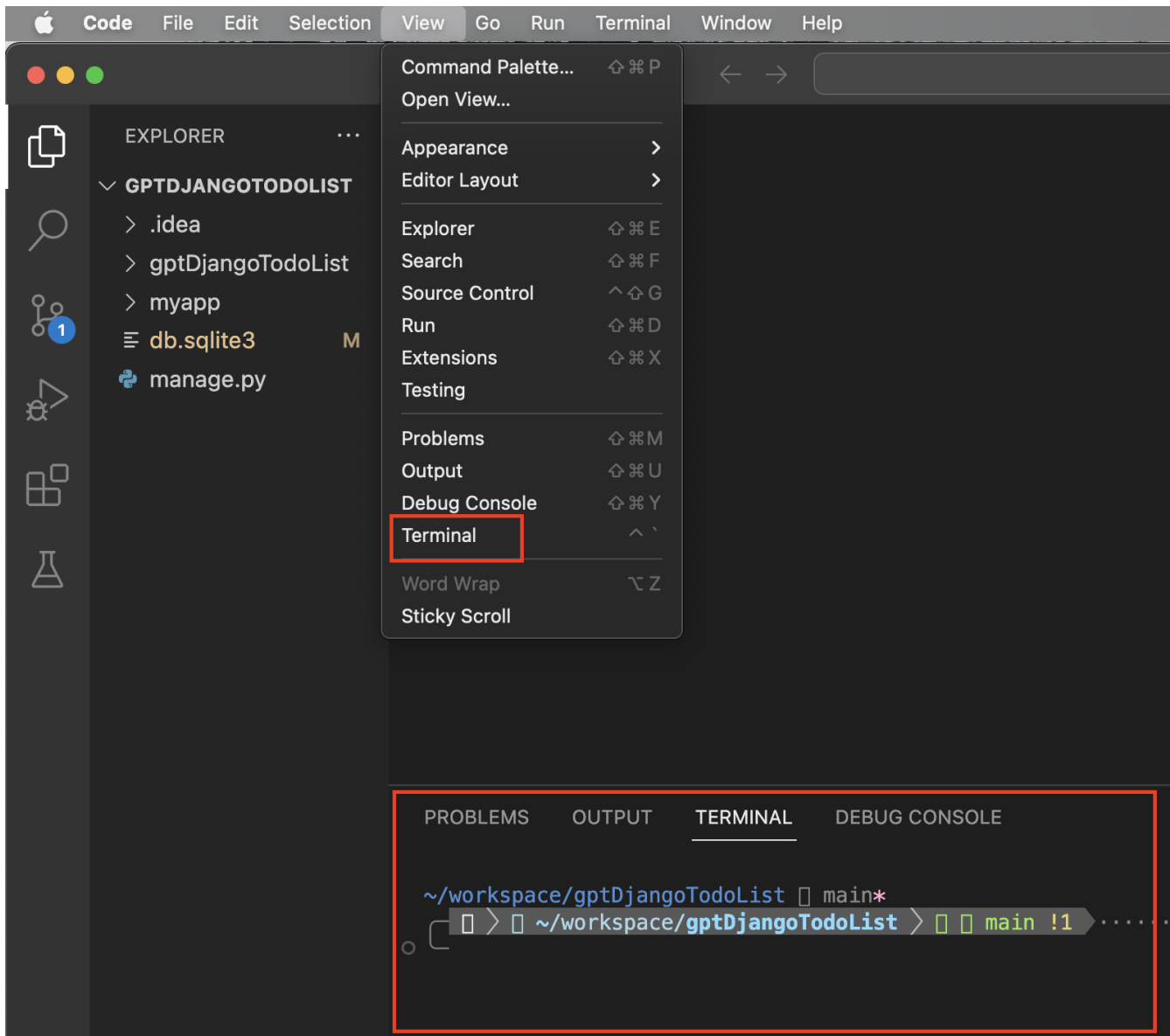
Chat GPT는 사용자가 제시한 조건에 맞춰서 어떤 코드를 어떤 파일에 작성해야 하는지 또 어떤 기능들을 하는지 설명까지 답변해줍니다. Chat GPT는 매번 다른답변을 하며 독자분들이 질문했을때는 다른내용을 답할 가능성이 높기때문에 전체 답변은 첨부하지 않았습니다. 코드보다는 개발방식을 주의깊게 봐주시기 바랍니다. 코드는 github 링크를 첨부했습니다.



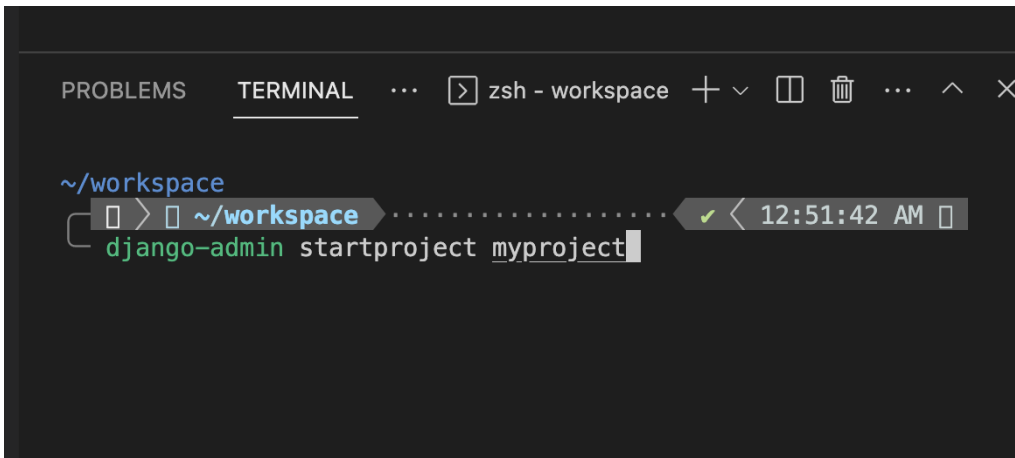
이렇게 제시해준 코드들을 VScode에 그대로 맞춰 작성하면 됩니다.

설치한 VScode를 실행한 후 Chat GPT가 설명해준대로 django 프로젝트를 생성합니다.

1. VScode 실행
2. 상단의 View 탭 -> Terminal 클릭하여 터미널 띄우기



3. 터미널에 “django-admin startproject myproject” 를 입력하여 django 프로젝트를 생성합니다.



The screenshot shows a terminal window with a dark background. At the top, there's a tab labeled 'zsh - workspace'. Below the tab, the terminal prompt is '~/.workspace'. The command 'django-admin startproject myproject' has been entered and is highlighted in green. The terminal also shows a status bar at the bottom with a green checkmark, a left arrow, the time '12:51:42 AM', and a right arrow.

4. “cd myproject” 를 입력하여 생성한 django 프로젝트 디렉토리로 접근합니다.
5. “python manage.py startapp myapp”을 실행하여 앱을 생성해줍니다. (여기서 만약 python에 빨간표시가 되며 실행이 안될 경우 python alias 설정이 되지않은것이므로)

```
python3 manage.py startapp myapp
```

위처럼 실행해주세요

6. 이렇게 앱 까지 생성해주게 되면 django 프로젝트의 기본적인 구조가 잡히게 됩니다. 이제 이렇게 생성된 구조 안에서 Chat GPT가 작성해준 코드들을 특정 파일에 맞춰 넣어주기만 하면됩니다.

```
models.py X
myapp > models.py > ...
1  # myapp/models.py
2
3  from django.db import models
4  from django.contrib.auth.models import User
5
6
7  class Todo(models.Model):
8      task = models.CharField(max_length=200)
9      is_completed = models.BooleanField(default=False)
10     user = models.ForeignKey(User, on_delete=models.CASCADE, null=True)
11
12     def __str__(self):
13         return self.task
14
```


이처럼 파일들을 찾아 해당하는 코드들을 복사 붙여넣기 해줍니다. (캡처사진은 models.py파일에 코딩한 사진입니다.)


chat GPT가 작성해준 모든 코드들을 넣은 뒤 프로젝트를 실행하여 정상동작하는지 체크합니다.

파이썬 django 프로젝트의 실행방법은 터미널에 명령어를 입력하여 실행할 수 있습니다.

```
python3 manage.py runserver
```

만일 이부분도 사전지식이 부족하여 알지못한다면 Chat GPT에게 질의하여 답을 얻어낼 수 있습니다.

 django 어플리케이션 터미널 실행 명령어

 Django 어플리케이션을 터미널에서 실행시키기 위해서는 다음과 같은 명령어를 사용합니다:

Copy code

```
python manage.py runserver
```

위 명령어를 실행하면 Django 서버가 실행되며, 기본적으로 ``http://127.0.0.1:8000/`` 주소에서 어플리케이션을 확인할 수 있습니다. 만약 다른 포트를 사용하고 싶다면, 포트 번호를 지정할 수 있습니다. 예를 들어, 포트 8080을 사용하려면 다음과 같이 명령어를 입력합니다:

yamlCopy code

```
python manage.py runserver 8080
```

위 명령어를 실행하면 ``http://127.0.0.1:8080/`` 주소에서 어플리케이션을 확인할 수 있습니다.

그렇게 서버를 실행하면 대부분의 코드에서는 오류가 발생할 것입니다.

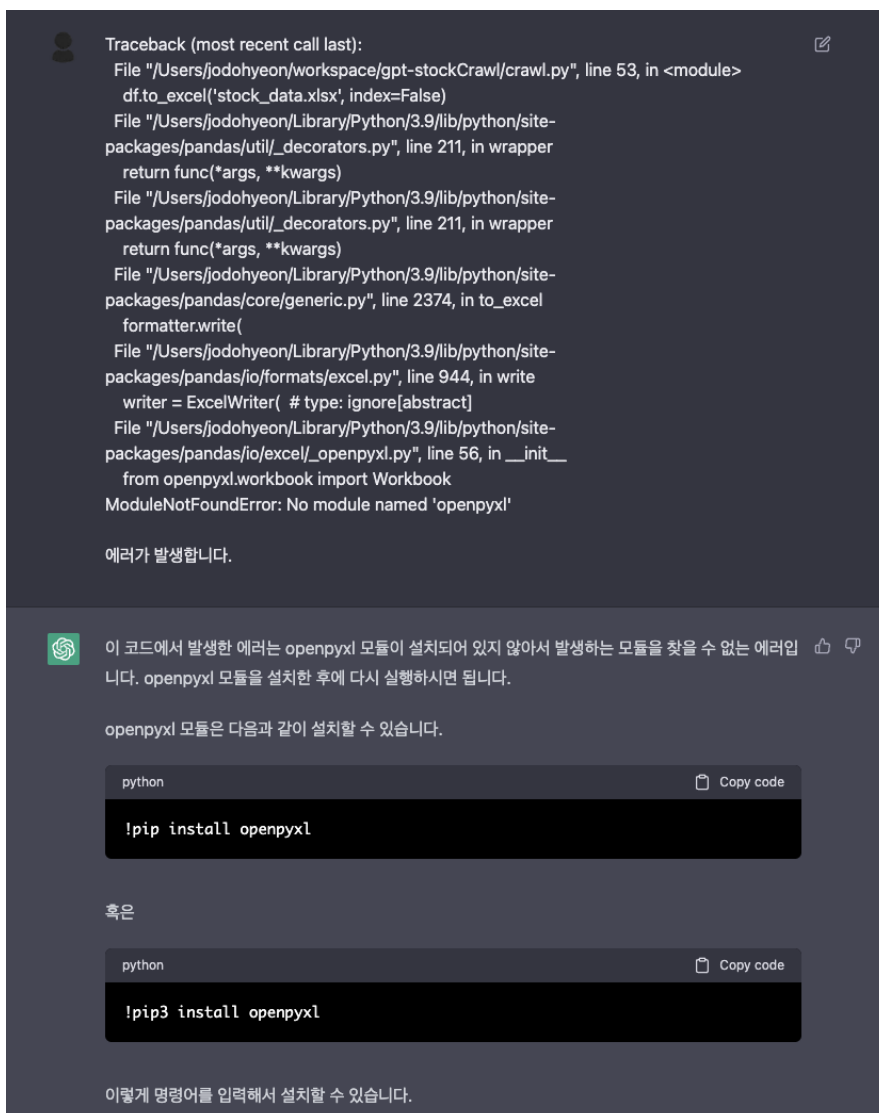
이 오류 또한 Chat GPT를 활용하여 해결할 수 있습니다.

다음 챕터에서 오류 수정을 진행하겠습니다.

## Chat GPT 활용 버그 및 기능 수정

필자의 경우 Chat GPT가 제공해준 코드로 정상적으로 실행되고 웹이 정상적으로 만들어졌지만 대부분의 경우 오류가 발생하고 터미널에 오류코드가 노출될 것 입니다.

Chat GPT가 매번 같은 답변을 주는것이 아니기 때문에 책에서 나온 답변과 다를 수 있다는점 참고 바랍니다.



The screenshot shows a ChatGPT interface with a dark theme. At the top, a user input (represented by a person icon) contains a Python traceback error. The error message is: "Traceback (most recent call last): File "/Users/jodohyeon/workspace/gpt-stockCrawl/crawl.py", line 53, in <module> df.to\_excel('stock\_data.xlsx', index=False) File "/Users/jodohyeon/Library/Python/3.9/lib/python/site-packages/pandas/util/\_decorators.py", line 211, in wrapper return func(\*args, \*\*kwargs) File "/Users/jodohyeon/Library/Python/3.9/lib/python/site-packages/pandas/util/\_decorators.py", line 211, in wrapper return func(\*args, \*\*kwargs) File "/Users/jodohyeon/Library/Python/3.9/lib/python/site-packages/pandas/core/generic.py", line 2374, in to\_excel formatter.write( File "/Users/jodohyeon/Library/Python/3.9/lib/python/site-packages/pandas/io/formats/excel.py", line 944, in write writer = ExcelWriter( # type: ignore[abstract] File "/Users/jodohyeon/Library/Python/3.9/lib/python/site-packages/pandas/io/excel/\_openpyxl.py", line 56, in \_\_init\_\_ from openpyxl.workbook import Workbook ModuleNotFoundError: No module named 'openpyxl'". Below the error, the text "에러가 발생합니다." (An error occurs.) is written. The ChatGPT response (represented by a green robot icon) explains that the error is due to the 'openpyxl' module not being installed and provides two commands to install it: "!pip install openpyxl" for python and "!pip3 install openpyxl" for pip3. The response also includes a "Copy code" button for each command block. At the bottom, it says "이렇게 명령어를 입력해서 설치할 수 있습니다." (You can install it by entering the command like this.).

```
Traceback (most recent call last):
File "/Users/jodohyeon/workspace/gpt-stockCrawl/crawl.py", line 53, in <module>
  df.to_excel('stock_data.xlsx', index=False)
File "/Users/jodohyeon/Library/Python/3.9/lib/python/site-packages/pandas/util/_decorators.py", line 211, in wrapper
    return func(*args, **kwargs)
File "/Users/jodohyeon/Library/Python/3.9/lib/python/site-packages/pandas/util/_decorators.py", line 211, in wrapper
    return func(*args, **kwargs)
File "/Users/jodohyeon/Library/Python/3.9/lib/python/site-packages/pandas/core/generic.py", line 2374, in to_excel
    formatter.write(
File "/Users/jodohyeon/Library/Python/3.9/lib/python/site-packages/pandas/io/formats/excel.py", line 944, in write
    writer = ExcelWriter( # type: ignore[abstract]
File "/Users/jodohyeon/Library/Python/3.9/lib/python/site-packages/pandas/io/excel/_openpyxl.py", line 56, in __init__
    from openpyxl.workbook import Workbook
ModuleNotFoundError: No module named 'openpyxl'
```

에러가 발생합니다.

이 코드에서 발생한 에러는 openpyxl 모듈이 설치되어 있지 않아서 발생하는 모듈을 찾을 수 없는 에러입니다. openpyxl 모듈을 설치한 후에 다시 실행하시면 됩니다.

openpyxl 모듈은 다음과 같이 설치할 수 있습니다.

```
python
!pip install openpyxl
```

혹은

```
python
!pip3 install openpyxl
```

이렇게 명령어를 입력해서 설치할 수 있습니다.

이런식으로 터미널에서 발생한 오류 또한 Chat GPT에게 오류코드를 그대로 제공하는 방식으로 해결법을 얻을 수 있습니다.

오류를 해결하고 웹을 확인하게 되면 오류가 발생하여 웹사이트가 안보이거나 또는 잘못된 UI로 보인다거나 기능이 제대로 동작하지 않거나 여러가지 사항이 발생할 수 있습니다. 이 또한 Chat GPT만을 활용해서 해결이 가능합니다.

필자의 경우 웹사이트가 정상적으로 만들어졌지만 투두리스트 추가 버튼이 동작하지 않았습니다.



add를 하여 todolist를 추가해봤지만 아무런 일도 일어나지 않습니다.



Chat GPT가 이미 코드를 제공해줬고 그 코드에 대한내용이 학습되어있는 상태이기때문에 코드를 따로 제공하지않고 그대로 문제를 제시했습니다.



투두리스트가 추가되지 않는다면, 다음과 같은 경우들을 다시 확인해보세요.



1. ``form`` 태그에서 ``action`` 속성이 올바르게 설정되어 있는지 확인하세요. ``action`` 속성은 데이터를 제출할 URL을 설정하는데, 투두리스트를 추가하는 뷰의 URL을 지정해주어야 합니다.
2. ``views.py`` 파일에서 ``create`` 함수가 올바르게 정의되어 있는지 확인하세요. ``create`` 함수에서 ``form`` 데이터가 제출되었을 때, 해당 데이터를 저장하도록 설정되어야 합니다.
3. ``urls.py`` 파일에서 ``create`` 뷰에 대한 URL이 올바르게 설정되어 있는지 확인하세요.

위 항목들을 다시 확인해보시고, 문제가 해결되지 않으면 추가적인 정보와 코드를 제공해주세요.

그럼 Chat GPT는 본인이 준 코드에서 의심되는 부분들을 제시해 줄 것입니다.

하지만, 개발지식이 없다면 이 설명을 들어도 모르는게 대부분일 것입니다.

이럴땐 전체소스코드를 제공하여 질의를 합니다. 단, 개발 규모가 커서 코드가 너무 길다면 토큰 초과로 질의가 불가할 것입니다. 그럴땐 의심되는 코드를 제공해야 하는데 이 책은 비전공자를 위한 Chat GPT개발방식을 알아보는 책이기 때문에 코드의 양이 많지않고 토큰의 제약에 제한되지않기때문에 코드를 전부 제공하여 결과를 얻어내는 방식으로 진행하겠습니다.

```

        username = form.cleaned_data.get('username')
        messages.success(request, f'Account created for {username}!')
        return redirect('login')
    else:
        form = UserCreationForm()
        return render(request, 'myapp/register.html', {'form': form})

```

view 파일입니다. 여기에 문제가있어서 투두리스트 추가가 안되는건가요?

Chat GPT가 view.py 파일을 언급했기때문에 해당 파일에 존재하는 코드 전체를 제공하여 문제 파악을 요청합니다.



보여지는 코드에서는 문제가 없어 보입니다. 추가하려는 데이터가 모델에 정상적으로 저장되지 않는 이유를 파악하기 위해서는 form 데이터를 받아서 저장하는 view 함수와 연관된 url 및 template 코드도 확인이 필요합니다. 만약 그 코드들이 문제가 있을 수도 있으니 함께 체크해보시는 것이 좋겠습니다.

문제가 없다면 다른 의심해야 되는 부분을 제안합니다.

```

<label class="form-check-label">Completed</label>
</div>
</form>
</li>
{% empty %}
<li class="list-group-item text-center">No completed tasks</li>
{% endfor %}
</ul>
{% endblock %}

```

투두리스트를 보여주는 메인 페이지 입니다. 이곳에 문제가 있어서 투두리스트가 노출되지 않는건가요?

의심되는 부분의 코드를 제공하고 또 해결방법을 찾아달라고합니다.



투두리스트를 추가하는 폼만 보여주고, 실제로 추가하는 처리를 하는 부분이 없기 때문에 문제가 있습니다. 이를 해결하기 위해 index 뷰에 POST 요청이 들어올 경우 Todo 모델에 새로운 객체를 추가하도록 코드를 추가해야 합니다. 다음과 같이 코드를 수정해보세요.



python

Copy code

```
from django.http import HttpResponseRedirect

def index(request):
    if request.method == 'POST':
        task = request.POST.get('task')
        if task:
            Todo.objects.create(task=task)
            messages.success(request, 'New task added successfully.')
        else:
            messages.error(request, 'Task is empty.')
        return HttpResponseRedirect(request.path_info)
    else:
        todo_list_not_completed = Todo.objects.filter(is_completed=False)
        todo_list_completed = Todo.objects.filter(is_completed=True)
        return render(request, 'myapp/index.html', {
            'todo_list_not_completed': todo_list_not_completed,
            'todo_list_completed': todo_list_completed,
        })
```

변경된 코드에서는 `request.method`가 'POST'일 때 Todo 모델에 새로운 객체를 추가하고, 추가가 성공하면 `HttpResponseRedirect` 함수를 사용해 다시 현재 URL로 리디렉션해줍니다. 이를 통해 투두리스트가 추가되면 새로고침 없이 목록이 갱신되는 것을 확인할 수 있습니다.

그럼 코드를 확인 후 해결방법이 무엇인지 알려줍니다. 보통 전체 코드를 다시 알려주는 식으로 제공을 해주니 해당 파일을 전체 수정하면 원하는 결과값을 얻을 수 있습니다.

다음으로는 기능 수정을 진행해보겠습니다.

현재는 투두리스트가 로그인 하지않아도 노출이되고있습니다. 원하는 기능은 각각 로그인을 하고 본인의 투두리스트만 관리할 수 있게 권한을 부여하는 것 입니다. 작업을 진행했던 Chat GPT의



세션이 끊겨서 학습했던 내용이 사라졌기 때문에 코드를 제공하고 수정요청하는 방식으로 진행했습니다.

```
        messages.success(request, f'Account created for {username}!')
        return redirect('login')
    else:
        form = UserCreationForm()
        return render(request, 'myapp/register.html', {'form': form})
```

위 코드는 django 투두리스트 웹의 views.py 코드입니다.

현재는 로그인한 회원이 누군지에 관계없이 로그인하지않아도 투두리스트를 볼수있는데 이 투두리스트를 로그인한 본인이 등록한것만 볼수있게 변경할수 있나요?

아래는 index.html 입니다.

```
<!-- myapp/templates/myapp/index.html -->

{% extends 'myapp/base.html' %}

{% block content %}
<h1 class="mb-4">My To-Do List</h1>

<form method="post">
    {% csrf_token %}
    <div class="input-group mb-3">
        <input type="text" name="task" class="form-control" placeholder="New task"
required>
        <div class="input-group-append">
            <button class="btn btn-primary" type="submit">Add</button>
        </div>
    </div>
</form>
```

보통 이렇게 간단한 파이썬, 장고 기반의 웹 애플리케이션의 경우 views.py 파일에는 기능 동작에 필요한 로직이 구성되어있고 index.html에는 화면에 보여줄 DOM이 구성되어있기때문에 이 두 개의 파일을 제공하는것만으로 기능을 추가하거나 수정할 수 있습니다.



네, 이 코드에서는 로그인한 사용자의 투두리스트만 보여주는 기능이 구현되어 있지 않습니다. 하지만 이를 구현하는 것은 상대적으로 간단합니다.

우선, `Todo` 모델에 `user` 필드를 추가하여 각 투두리스트 항목이 특정 사용자에게 속하도록 할 수 있습니다. 이렇게 하면 특정 사용자가 생성한 투두리스트 항목만 쉽게 필터링 할 수 있습니다. 아래는 이를 위한 `Todo` 모델의 수정 버전입니다.

```
python Copy code

# myapp/models.py

from django.db import models
from django.contrib.auth.models import User

class Todo(models.Model):
    task = models.CharField(max_length=200)
    is_completed = models.BooleanField(default=False)
    user = models.ForeignKey(User, on_delete=models.CASCADE)

    def __str__(self):
        return self.task
```

이제 `index` 뷰에서는 현재 로그인한 사용자가 소유한 투두리스트 항목만 필터링하도록 변경할 수 있습니다. `request.user` 속성을 사용하여 현재 로그인한 사용자를 가져올 수 있으며, `Todo.objects.filter()` 메서드를 사용하여 현재 사용자의 투두리스트를 필터링할 수 있습니다. 변경된 `index` 뷰는 다음과 같습니다.

```
python Copy code

# myapp/views.py

from django.contrib.auth.decorators import login_required

@login_required
def index(request):
    if request.method == 'POST':
        task = request.POST.get('task')
        if task:
            Todo.objects.create(task=task, user=request.user)
            messages.success(request, 'New task added successfully.')
        else:
            messages.error(request, 'Task is empty.')
        return HttpResponseRedirect(request.path_info)
    else:
        todo_list_not_completed = Todo.objects.filter(user=request.user, is_completed=False)
        todo_list_completed = Todo.objects.filter(user=request.user, is_completed=True)
        return render(request, 'myapp/index.html', {
            'todo_list_not_completed': todo_list_not_completed,
            'todo_list_completed': todo_list_completed,
        })
```

위 코드에서 `@login\_required` 데코레이터를 사용하여 로그인한 사용자만 해당 뷰에 접근할 수 있도록 했습니다. 이제 로그인하지 않은 사용자는 투두리스트를 볼 수 없습니다.

`Todo.objects.create()` 메서드를 호출할 때 `user` 인자를 현재 로그인한 사용자로 설정하여 투두리스트 항목이 현재 사용자에게 속하도록 했습니다. 마지막으로, `Todo.objects.filter()` 메서드를 사용하여 현재 사용자의 투두리스트 항목만 필터링하도록 했습니다.

이렇게 변경된 `index` 뷰는 현재 로그인한 사용자가 소유한 투두리스트 항목만 보여주게 됩니다. 다른 뷰(`create`, `update`, `delete`)도 동일한 방식으로 변경하여 현재

코드 제공과 동시에 어떤 파일에 넣어줘야 하는지 또 왜 코드가 추가되고 삭제되었는지 설명까지 해주니 웹 제작의 목적에 개발공부라는 목적이 포함되어있다면 Chat GPT를 통해 공부도 가능합니다. Chat GPT가 생겨나기 이전의 개발공부법으로는 구글에 검색을 하고 나의 상황과 맞는지 판단하고 오류를 수정해보고 하면서 소비됐던 많은 시간들을 절약할 수 있는것 입니다.

이렇게 Chat GPT를 활용하여 한시간도 되지않는 시간에 파이썬 기반의 웹 애플리케이션을 만들었습니다. 결론적으로 Chat GPT를 활용하면 생산성 면에서 엄청나게 효율적으로 개발을 할 수 있으며 정확도 또한 상당히 높다는 점입니다. 물론 아직 부족한 점과 불편한 점들이 존재하지만 개발을 전체적인 모든 작업을 Chat GPT에 맡긴다기 보단 필요한 부분들만 활용한다면 불편한점은 크게 없을거라 생각이 듭니다. 또한 공부를 위한 목적이라면 검색을 통해서 잘못된 정보 정확한 정보를 구분하면서 공부하는것보다 Chat GPT를 통해 지식을 습득하는것도 새로운 공부법이 될거란 생각이 듭니다.

완성본 소스코드

- GitHub : <https://github.com/DoHyeonJ/gpt-django-todolist>

## Chapter 4. 결론

Chat GPT를 활용한 웹 개발의 장점과 한계

Chat GPT를 활용한 웹 개발은 개발 지식이 없는 비전공자도 쉽게 개발할 수 있습니다. Chat GPT를 활용하면 미리 학습된 인공지능 모델을 활용하여 개발할 수 있기 때문에 개발 지식이 없더라도 간단한 설치와 설정만으로도 쉽게 개발을 시작할 수 있습니다. 또한 개발 지식이 있는 개발자들도 Chat GPT를 활용하면 빠르게 웹 개발을 수행할 수 있습니다.

또한 Chat GPT를 활용한 웹 개발은 접근성이 쉽고 생산성이 좋습니다. Chat GPT를 이용하여 개발하면 자연어 처리에 대한 지식이 필요하지 않습니다. 이는 개발자들이 쉽게 대화형 인터페이스를 구현할 수 있다는 것을 의미합니다. Chat GPT 모델을 활용하여 웹 개발을 수행하면

기존의 웹 개발 방식과 비교하여 생산성이 높아집니다. 개발자들은 대화형 인터페이스를 구현하기 위해 별도의 인터페이스 디자인이나 프로그래밍을 할 필요가 없으며 대화형 인터페이스를 빠르게 개발할 수 있습니다.

따라서, Chat GPT를 활용한 웹 개발은 개발 지식이 없는 비전공자들도 쉽게 개발할 수 있으며 개발 지식이 있는 개발자들도 빠르게 개발할 수 있는 장점이 있으며 접근성이 쉽고 생산성이 좋다는 이점이 있습니다.

### Chat GPT의 전망과 적응 방법

Chat GPT는 현재 대화형 인터페이스 분야에서 큰 관심을 받고 있으며, 앞으로 더 많은 분야에서 활용될 것으로 예상됩니다. Chat GPT의 기술력은 빠르게 발전하고 있기 때문에 앞으로는 더욱 높은 수준의 대화를 가능케 할 것입니다.

특히, Chat GPT는 개인화된 대화 가능성의 증가가 기대됩니다. Chat GPT는 학습 데이터를 기반으로 대화를 생성하기 때문에 학습 데이터가 다양하고 풍부하다면 보다 개인화된 대화를 구현할 수 있을 것입니다. 예를 들어 Chat GPT를 이용한 쇼핑몰 대화형 인터페이스는 사용자의 이전 구매 기록, 검색 기록 등을 분석하여 개인에 맞는 상품 추천 등을 제공할 수 있을 것입니다.

또한 Chat GPT는 앞으로 더 많은 언어와 문화권에서 활용될 것으로 예상됩니다. 현재 Chat GPT는 영어를 비롯한 다양한 언어에서 활용되고 있습니다. 앞으로는 더 많은 언어와 문화권에서의 활용이 예상되며 이를 통해 전 세계적인 대화형 인터페이스 기술 발전에 기여할 것으로 기대됩니다.

Chat GPT를 활용한 웹 개발을 위해서는 깊은 정도의 프로그래밍 지식을 원하진 않지만 먼저 기본적인 프로그래밍 지식과 대화형 인터페이스 개발에 대한 이해가 필요합니다. 이를 위해 적극적인 학습과 공부가 필요하며 관련 서적이나 온라인 강좌를 통해 학습할 수 있습니다. 또한 Chat GPT를 활용한 웹 개발에 대한 다양한 예제나 프로젝트를 수행해보면서 실제 적용 경험을 쌓는 것이 중요합니다.

다음 단계로 나아갈 방법 제시

앞으로의 IT의 방향성에서는 AI는 주목받을 수 밖에 없는 기술입니다. 그 중에서도 Chat GPT는 중심이되어 많은 방향성들을 제시하고있으며 그런 IT 생태계에 적응하기 위해서는 끊임없는 학습이 필요합니다. 아래와 같은 다양한 방법들은 우리가 앞으로의 세계에 적응하는데 방향을 잡아줄 것입니다.

#### 1. 프로그래밍 학습

- Chat GPT를 활용한 웹 개발을 위해서는 기본적인 프로그래밍 지식이 필요합니다. 프로그래밍 학습은 관련 서적이나 온라인 강좌를 통해 진행할 수 있습니다. 또한 프로그래밍 언어나 개발 환경에 대한 이해도 필요합니다.

#### 2. 대화형 인터페이스 개발 학습

- Chat GPT를 활용한 대화형 인터페이스 개발에 대한 이해가 필요합니다. 이를 위해서는 대화형 인터페이스 개발에 대한 학습이 필요합니다. 관련 서적이나 온라인 강좌를 통해 대화형 인터페이스 개발에 대한 이해를 높일 수 있습니다.

#### 3. Chat GPT 모델 학습

- Chat GPT를 활용하기 위해서는 미리 학습된 Chat GPT 모델을 사용하거나 새로운 모델을 학습해야 합니다. 이를 위해서는 관련된 학습 데이터셋을 수집하고 전처리하는 과정이 필요합니다. 이를 위해 온라인 강좌나 자료들을 참고하여 학습 데이터셋 전처리 및 학습 방법에 대한 학습을 진행할 수 있습니다.

#### 4. 예제나 프로젝트 수행

- 필자가 가장 중요하게 생각하는 부분입니다. Chat GPT를 활용한 대화형 인터페이스 개발을 위해서는 다양한 예제나 프로젝트를 수행해볼 수 있습니다. 이를 통해 실제 적용 경험을 쌓고 개발 기술을 향상시킬 수 있습니다. 서적이나 문서 영상을 보는것도 좋지만 직접 적용해보고 사용해본다면 서적이나 문서를 본 것 보다 더욱 도움이 되는 지식을 얻을 수 있고 빠르게 적응할 수 있을 것 입니다.

이 자료는 대한민국 저작권법의 보호를 받습니다. 작성된 모든 내용의 권리는 작성자에게 있으며, 작성자의 동의 없는 사용이 금지됩니다. 본 자료의 일부 혹은 전체 내용을 무단으로 복제/배포하거나 2차적 저작물로 재편집하는 경우, 5년 이하의 징역 또는 5천만원 이하의 벌금과 민사상 손해배상을 청구합니다. ※ 저작권법 제 30조(사적이용을 위한 복제) 공표된 저작물을 영리를 목적으로 하지 아니하고, 개인적으로 이용하거나 가정 및 이에 준하는 한정된 범위 안에서 이용하는 경우에는 그 이용자는 이를 복제할 수 있다. 다만, 공중의 사용에 제공하기 위하여 설치된 복사기기에 의한 복제는 그러지 아니하다. ※ 저작권법 제 136조(벌칙)의 ① 다음 각 호의 어느 하나에 해당하는 자는 5년 이하의 징역 또는 5천만원 이하의 벌금에 처하거나 이를 병과할 수 있다. 지적재산권 및 이 법에 따라 보호되는 재산적 권리(제93조에 따른 권리는 제외한다)를 복제, 공연, 공중송신, 전시, 배포, 대여, 2차적 저작물 작성의 방법으로 침해한 자 ※ 민법 제750조(불법행위의 내용) 고의 또는 과실로 인한 위법행위로 타인에게 손해를 가한 자는 그 손해를 배상할 책임이 있다.

JDH