CS471 – Web Technologies (Laboratory)	Qassim University College of Computer	Lab Week 7 / Django Models (Part 1)

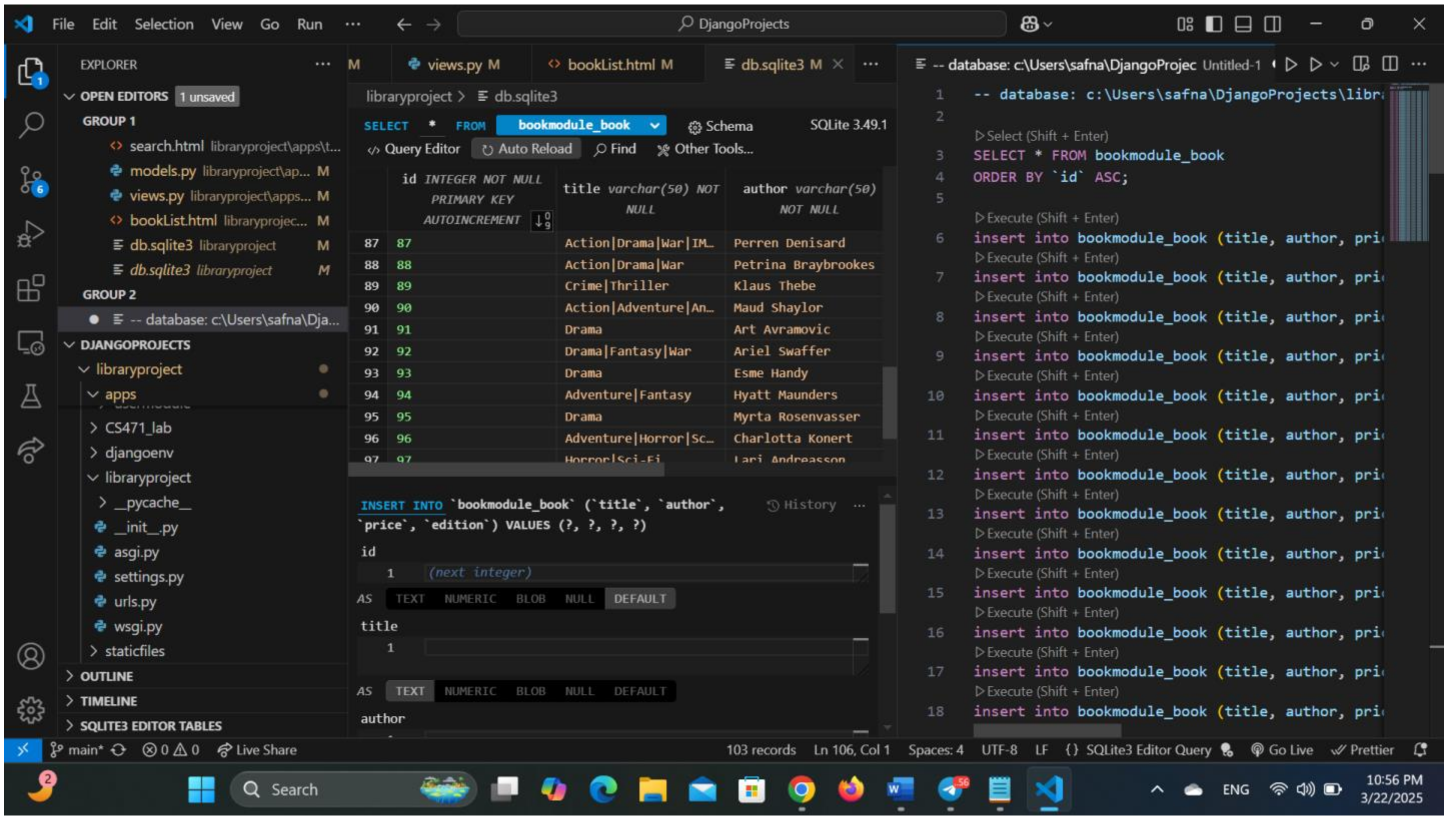
This lab session covers the basics of Django Models (Part 1). By the end of this session, students should have a very good understanding of how to build a basic model and how to access models in Django.

Pre-lab Preparation:

1. Having bookmodule and usermodule in apps directory, with necessary configurations in settings.py.

Lab Activities:

Task 2: use the application 'DB Browser for SQLite' or 'VS code' to view the database and insert fake data into it.



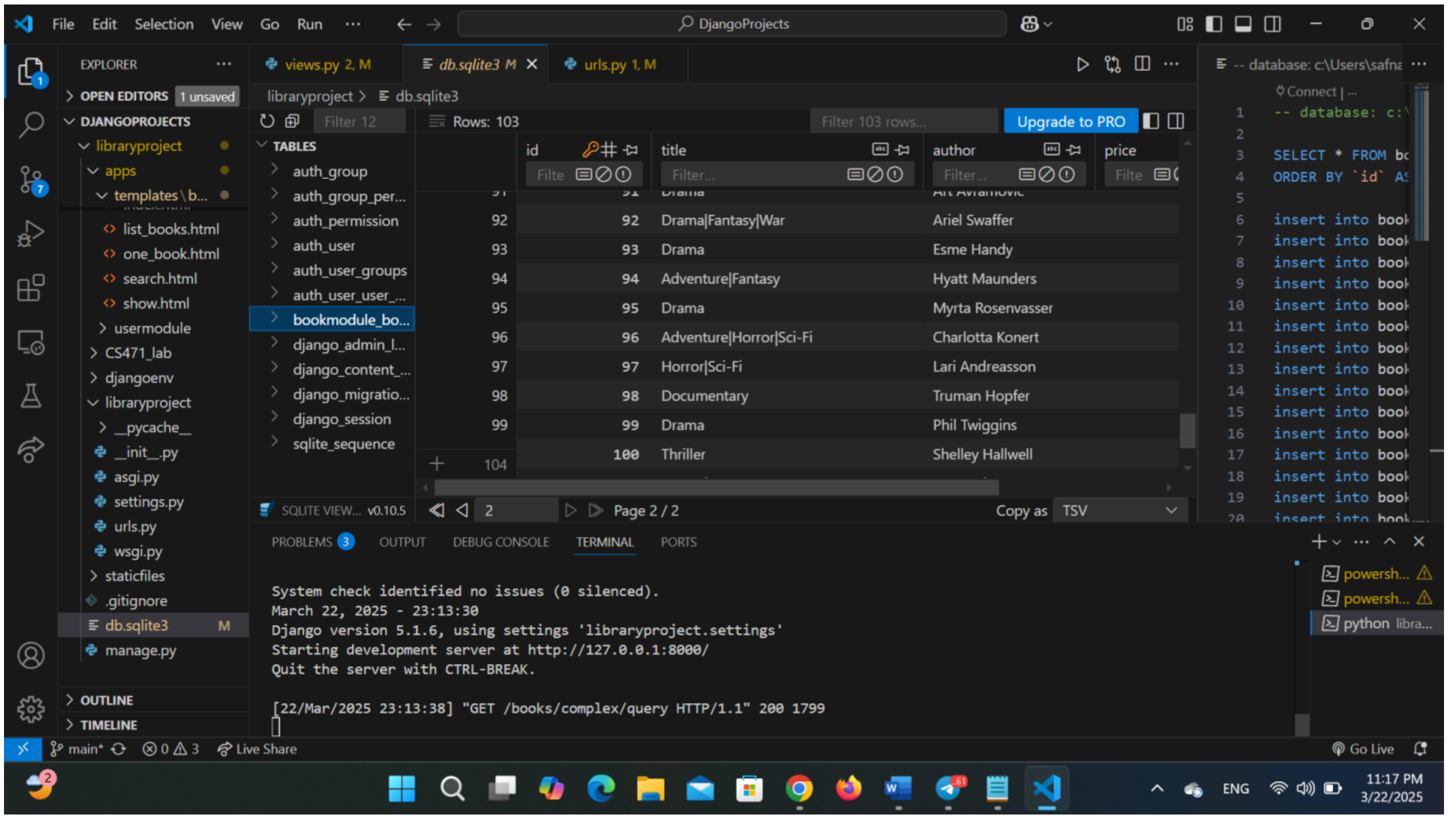
```
SELECT * FROM bookmodule_book
```

	id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT	title varchar(50) NOT NULL	author varchar(50) NOT NULL
87	87	Action|Drama|War|IM…	Perren Denisard
88	88	Action|Drama|War	Petrina Braybrookes
89	89	Crime|Thriller	Klaus Thebe
90	90	Action|Adventure|An…	Maud Shaylor
91	91	Drama	Art Avramovic
92	92	Drama|Fantasy|War	Ariel Swaffer
93	93	Drama	Esme Handy
94	94	Adventure|Fantasy	Hyatt Maunders
95	95	Drama	Myrta Rosenvasser
96	96	Adventure|Horror|Sc…	Charlotta Konert
97	97	Horror|Sci-Fi	Lari Andreasson

```
INSERT INTO `bookmodule_book` (`title`, `author`, `price`, `edition`) VALUES (?, ?, ?, ?)
```

```
-- database: c:\Users\safna\DjangoProjects\libr

SELECT * FROM bookmodule_book
ORDER BY `id` ASC;

insert into bookmodule_book (title, author, pri
insert into bookmodule_book (title, author, pri
insert into bookmodule_book (title, author, pri
insert into bookmodule_book (title, author, pri
insert into bookmodule_book (title, author, pri
insert into bookmodule_book (title, author, pri
insert into bookmodule_book (title, author, pri
insert into bookmodule_book (title, author, pri
insert into bookmodule_book (title, author, pri
insert into bookmodule_book (title, author, pri
insert into bookmodule_book (title, author, pri
insert into bookmodule_book (title, author, pri
insert into bookmodule_book (title, author, pri
```

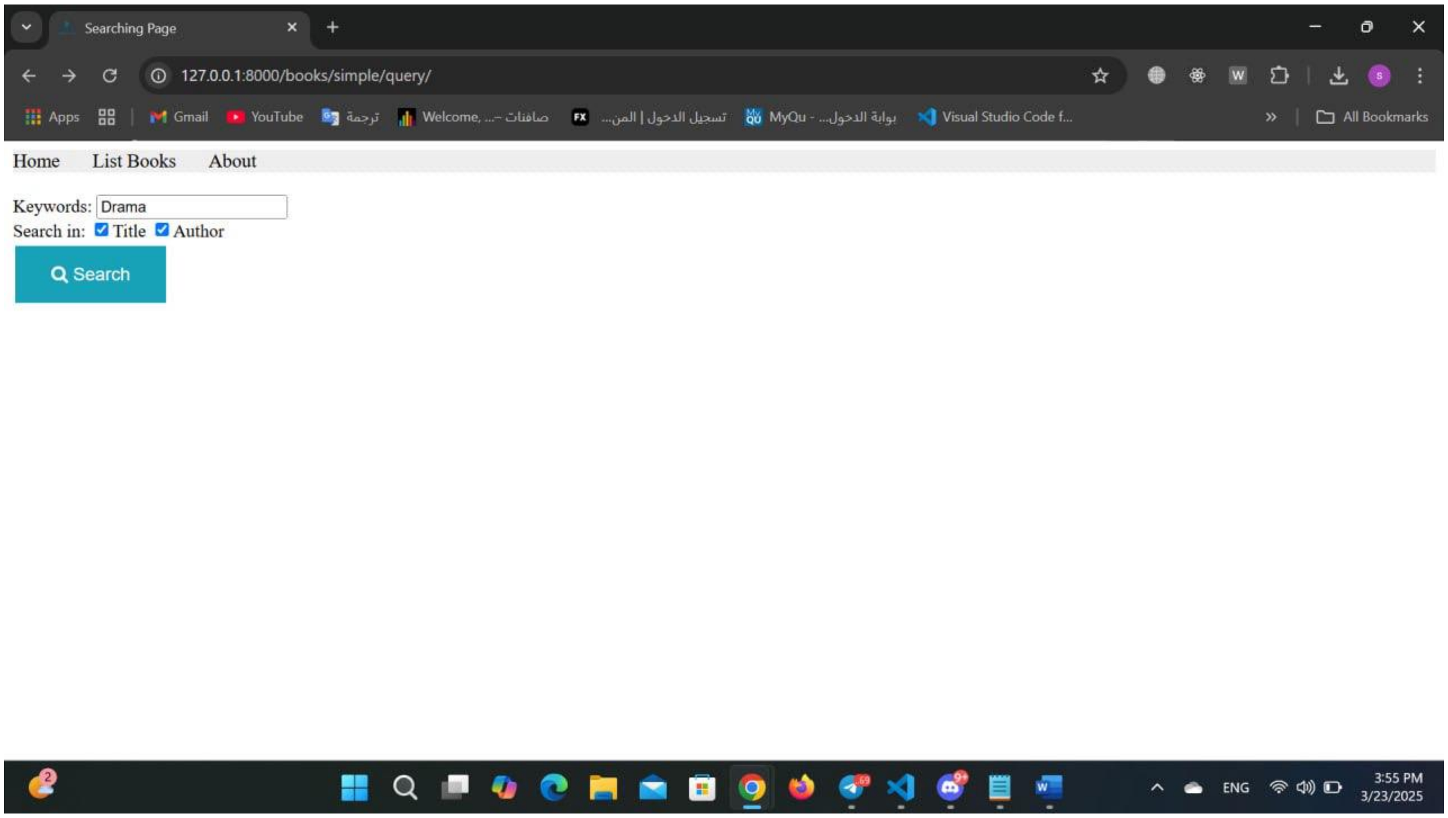
103 records



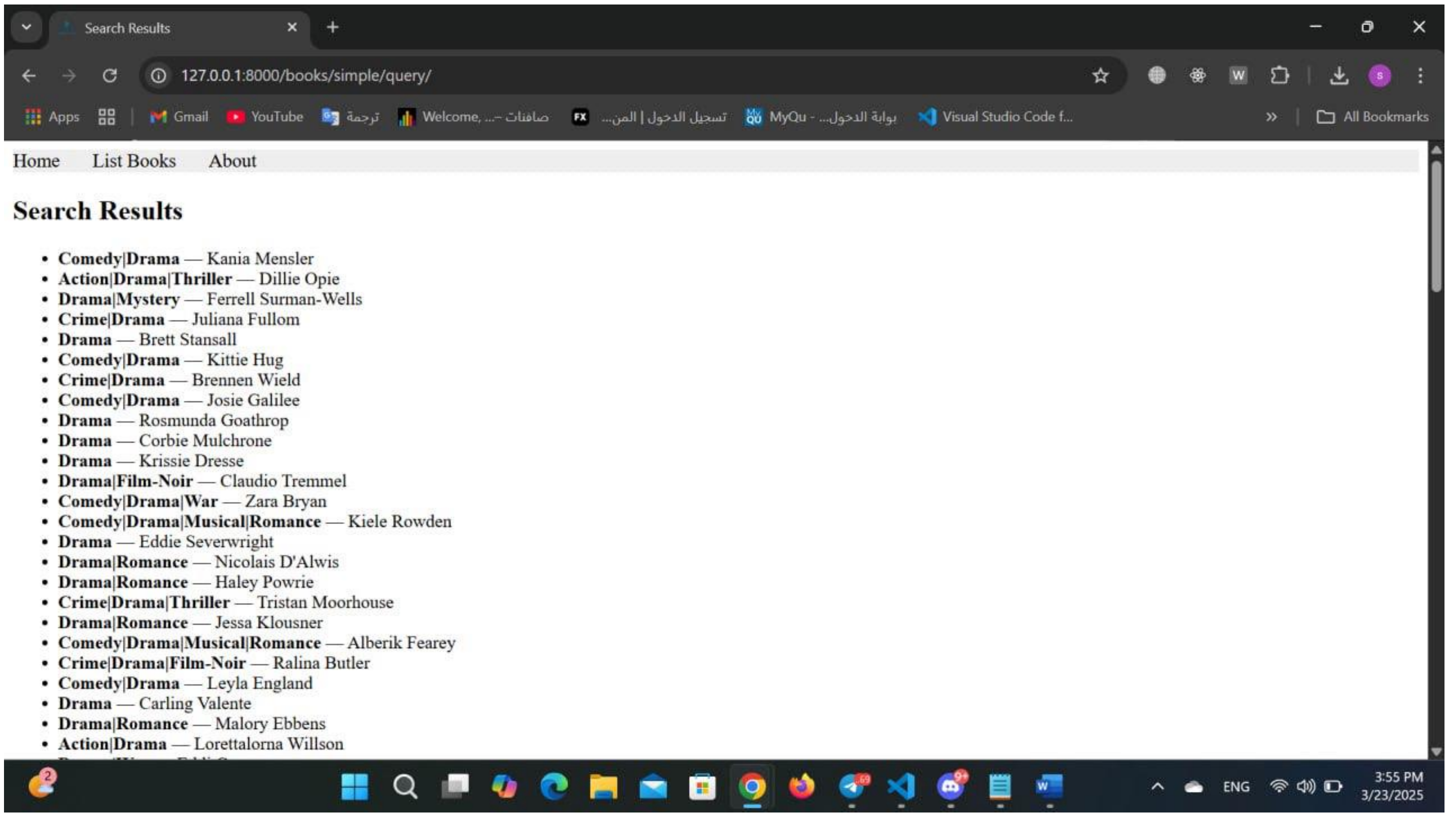
	id	title	author
91	91	Drama	Art Avramovic
92	92	Drama|Fantasy|War	Ariel Swaffer
93	93	Drama	Esme Handy
94	94	Adventure|Fantasy	Hyatt Maunders
95	95	Drama	Myrta Rosenvasser
96	96	Adventure|Horror|Sci-Fi	Charlotta Konert
97	97	Horror|Sci-Fi	Lari Andreasson
98	98	Documentary	Truman Hopfer
99	99	Drama	Phil Twiggins
104	100	Thriller	Shelley Hallwell

```
System check identified no issues (0 silenced).
March 22, 2025 - 23:13:30
Django version 5.1.6, using settings 'libraryproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[22/Mar/2025 23:13:38] "GET /books/complex/query HTTP/1.1" 200 1799
```

Task 3: Use simple queries using Django ORM.



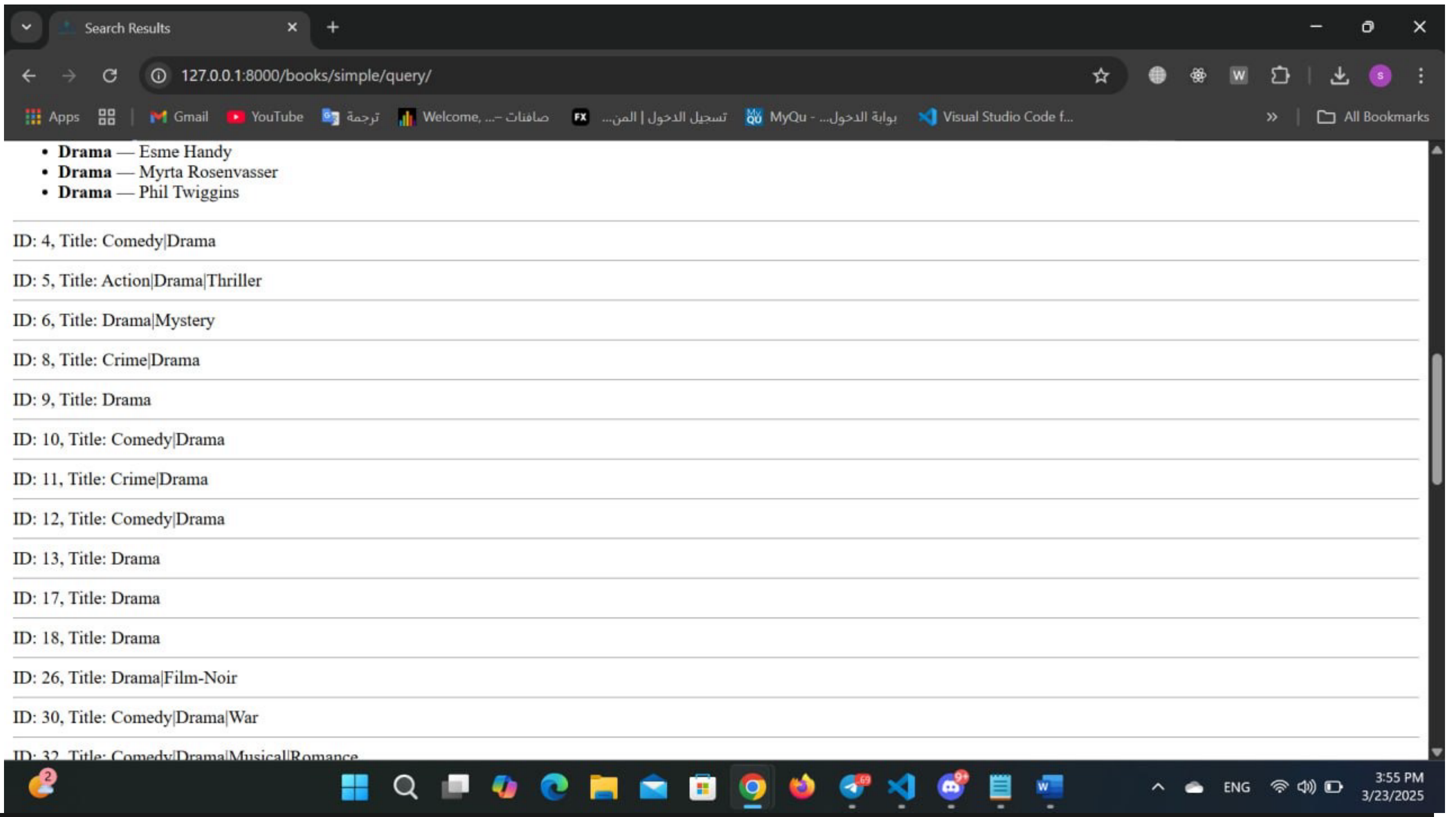
Keywords: Drama
Search in: [✓ Title] [✓ Author]



Search Results

- **Comedy|Drama** — Kania Mensler
- **Action|Drama|Thriller** — Dillie Opie
- **Drama|Mystery** — Ferrell Surman-Wells
- **Crime|Drama** — Juliana Fullom
- **Drama** — Brett Stansall
- **Comedy|Drama** — Kittie Hug
- **Crime|Drama** — Brennen Wield
- **Comedy|Drama** — Josie Galilee
- **Drama** — Rosmunda Goathrop
- **Drama** — Corbie Mulchrone
- **Drama** — Krissie Dresse
- **Drama|Film-Noir** — Claudio Tremmel
- **Comedy|Drama|War** — Zara Bryan
- **Comedy|Drama|Musical|Romance** — Kiele Rowden
- **Drama** — Eddie Severwright
- **Drama|Romance** — Nicolais D'Alwis
- **Drama|Romance** — Haley Powrie
- **Crime|Drama|Thriller** — Tristan Moorhouse
- **Drama|Romance** — Jessa Klousner
- **Comedy|Drama|Musical|Romance** — Alberik Fearey
- **Crime|Drama|Film-Noir** — Ralina Butler
- **Comedy|Drama** — Leyla England
- **Drama** — Carling Valente
- **Drama|Romance** — Malory Ebbens
- **Action|Drama** — Lorettalorna Willson

CS471 – Web Technologies (Laboratory)	Qassim University College of Computer	Lab Week 7
		Django Models (Part 1)



127.0.0.1:8000/books/simple/query/

- **Drama** — Esme Handy
- **Drama** — Myrta Rosenvasser
- **Drama** — Phil Twiggins

ID: 4, Title: Comedy|Drama

ID: 5, Title: Action|Drama|Thriller

ID: 6, Title: Drama|Mystery

ID: 8, Title: Crime|Drama

ID: 9, Title: Drama

ID: 10, Title: Comedy|Drama

ID: 11, Title: Crime|Drama

ID: 12, Title: Comedy|Drama

ID: 13, Title: Drama

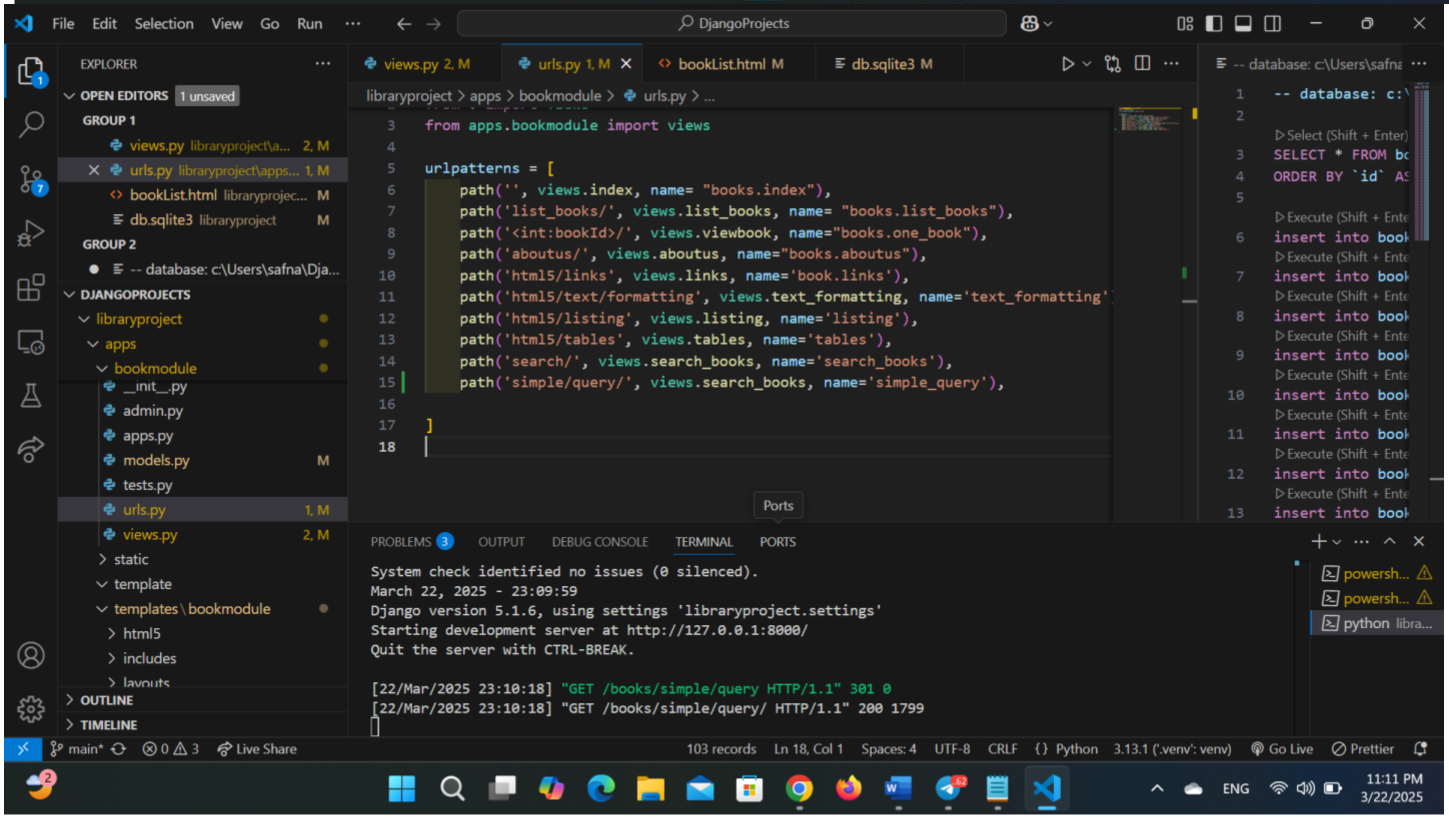
ID: 17, Title: Drama

ID: 18, Title: Drama

ID: 26, Title: Drama|Film-Noir

ID: 30, Title: Comedy|Drama|War

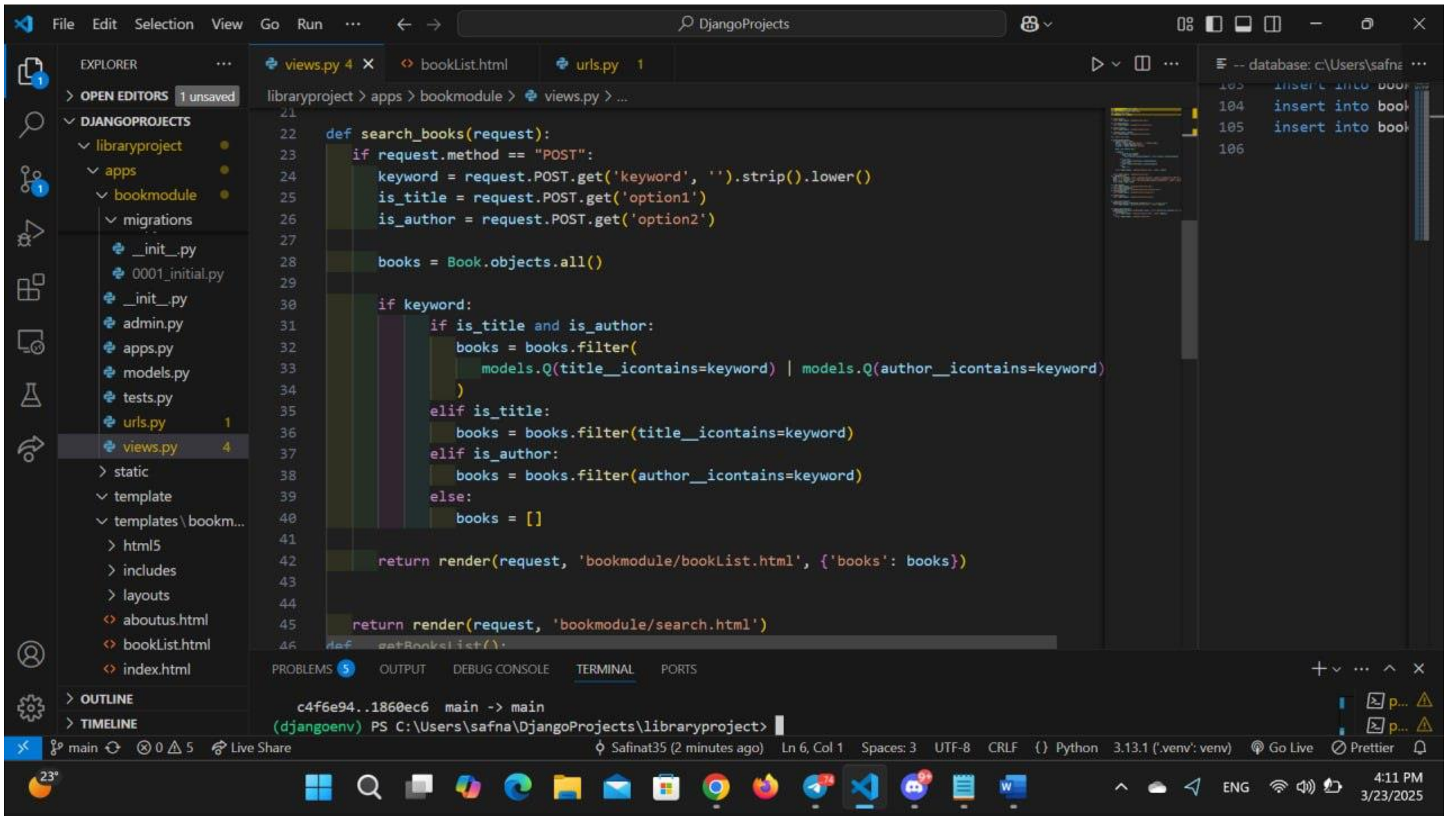
ID: 32, Title: Comedy|Drama|Musical|Romance



libraryproject > apps > bookmodule > urls.py > ...

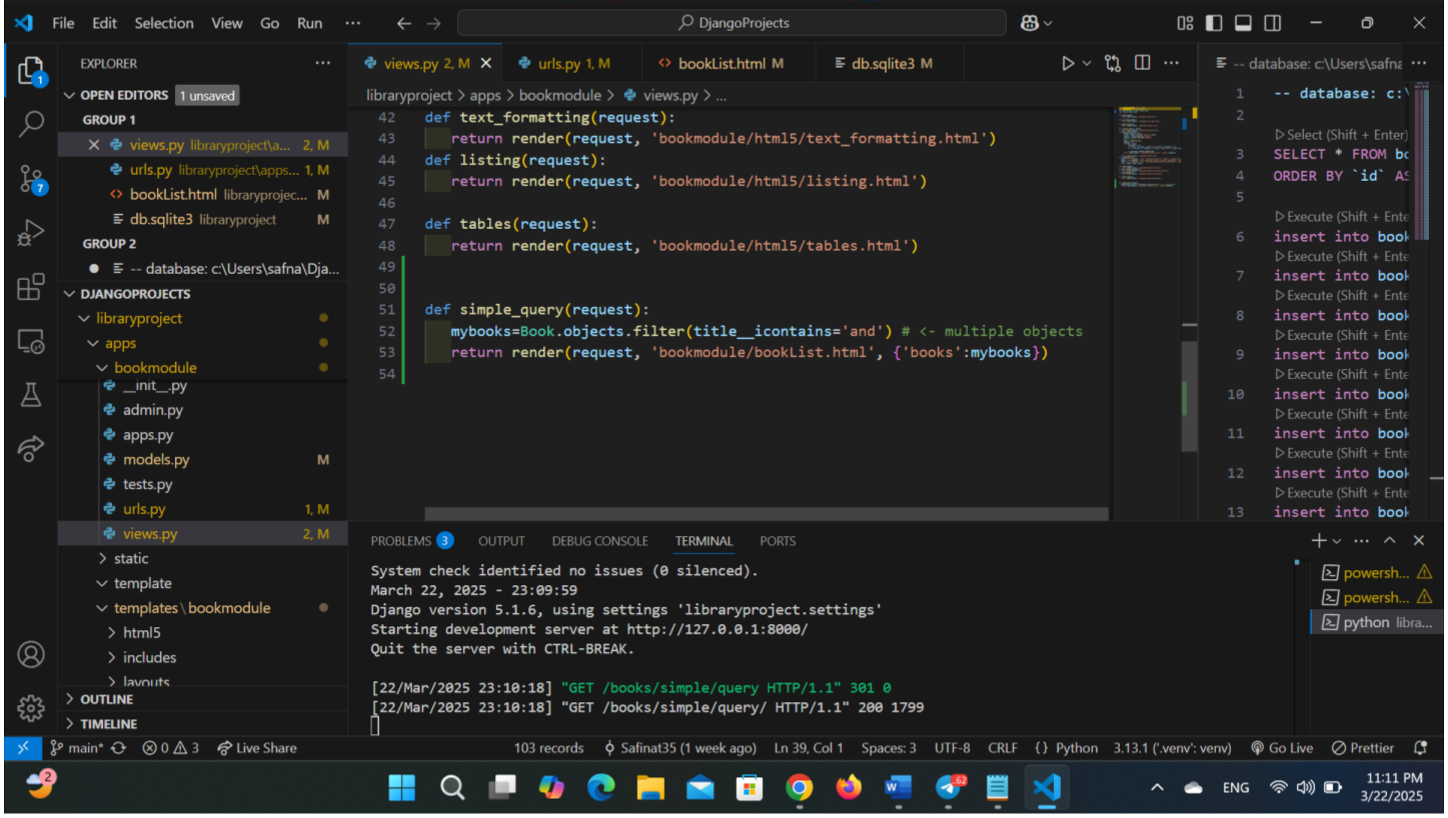
```python
from apps.bookmodule import views

urlpatterns = [
    path('', views.index, name= "books.index"),
    path('list_books/', views.list_books, name= "books.list_books"),
    path('<int:bookId>/', views.viewbook, name="books.one_book"),
    path('aboutus/', views.aboutus, name="books.aboutus"),
    path('html5/links', views.links, name='book.links'),
    path('html5/text/formatting', views.text_formatting, name='text_formatting'
    path('html5/listing', views.listing, name='listing'),
    path('html5/tables', views.tables, name='tables'),
    path('search/', views.search_books, name='search_books'),
    path('simple/query/', views.search_books, name='simple_query'),

]
```

```
System check identified no issues (0 silenced).
March 22, 2025 - 23:09:59
Django version 5.1.6, using settings 'libraryproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[22/Mar/2025 23:10:18] "GET /books/simple/query HTTP/1.1" 301 0
[22/Mar/2025 23:10:18] "GET /books/simple/query/ HTTP/1.1" 200 1799
```



```python
def search_books(request):
    if request.method == "POST":
        keyword = request.POST.get('keyword', '').strip().lower()
        is_title = request.POST.get('option1')
        is_author = request.POST.get('option2')

        books = Book.objects.all()

        if keyword:
            if is_title and is_author:
                books = books.filter(
                    models.Q(title__icontains=keyword) | models.Q(author__icontains=keyword)
                )
            elif is_title:
                books = books.filter(title__icontains=keyword)
            elif is_author:
                books = books.filter(author__icontains=keyword)
            else:
                books = []

        return render(request, 'bookmodule/bookList.html', {'books': books})


    return render(request, 'bookmodule/search.html')
```

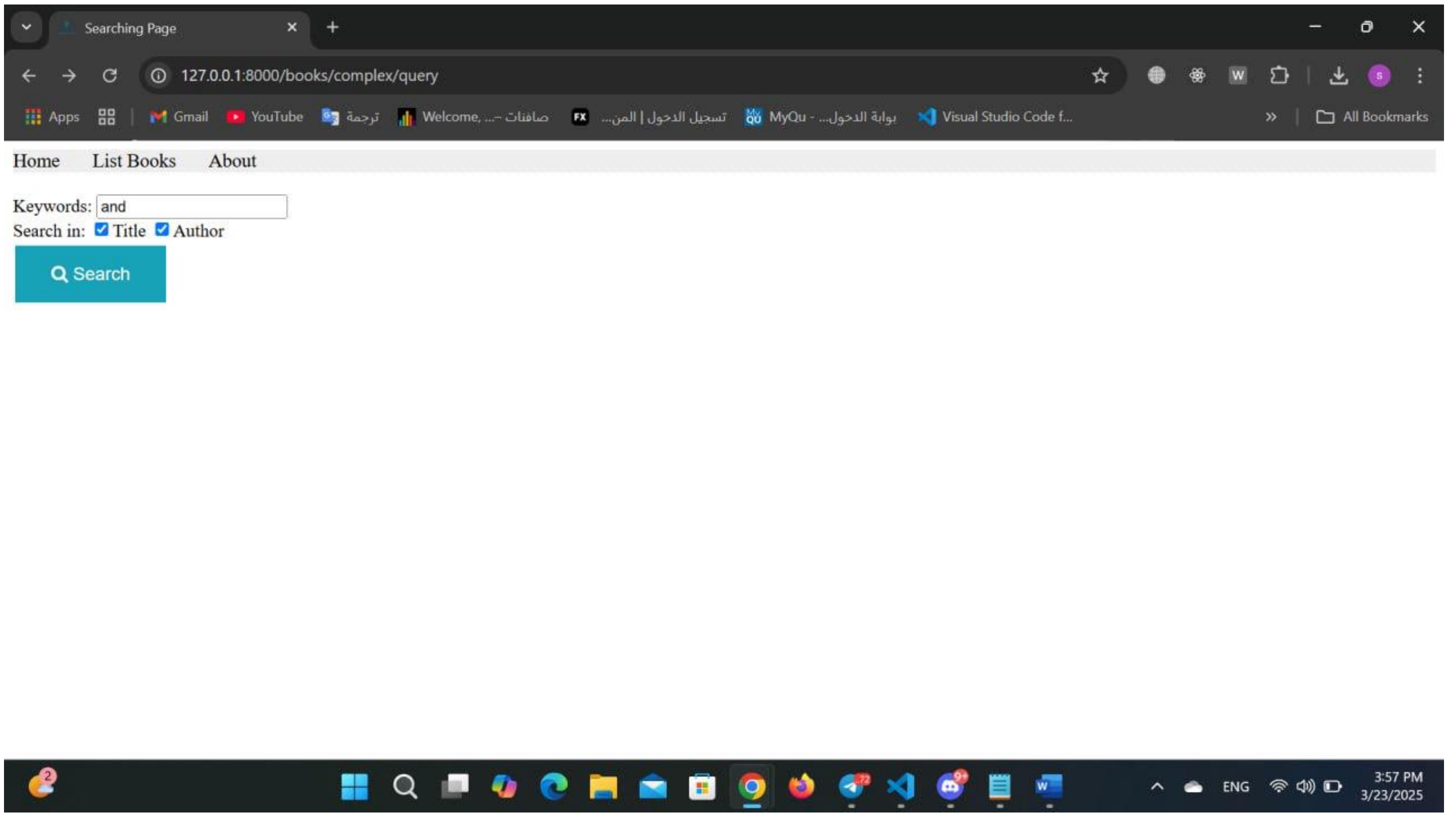
```
c4f6e94..1860ec6  main -> main
(djangoenv) PS C:\Users\safna\DjangoProjects\libraryproject>
```



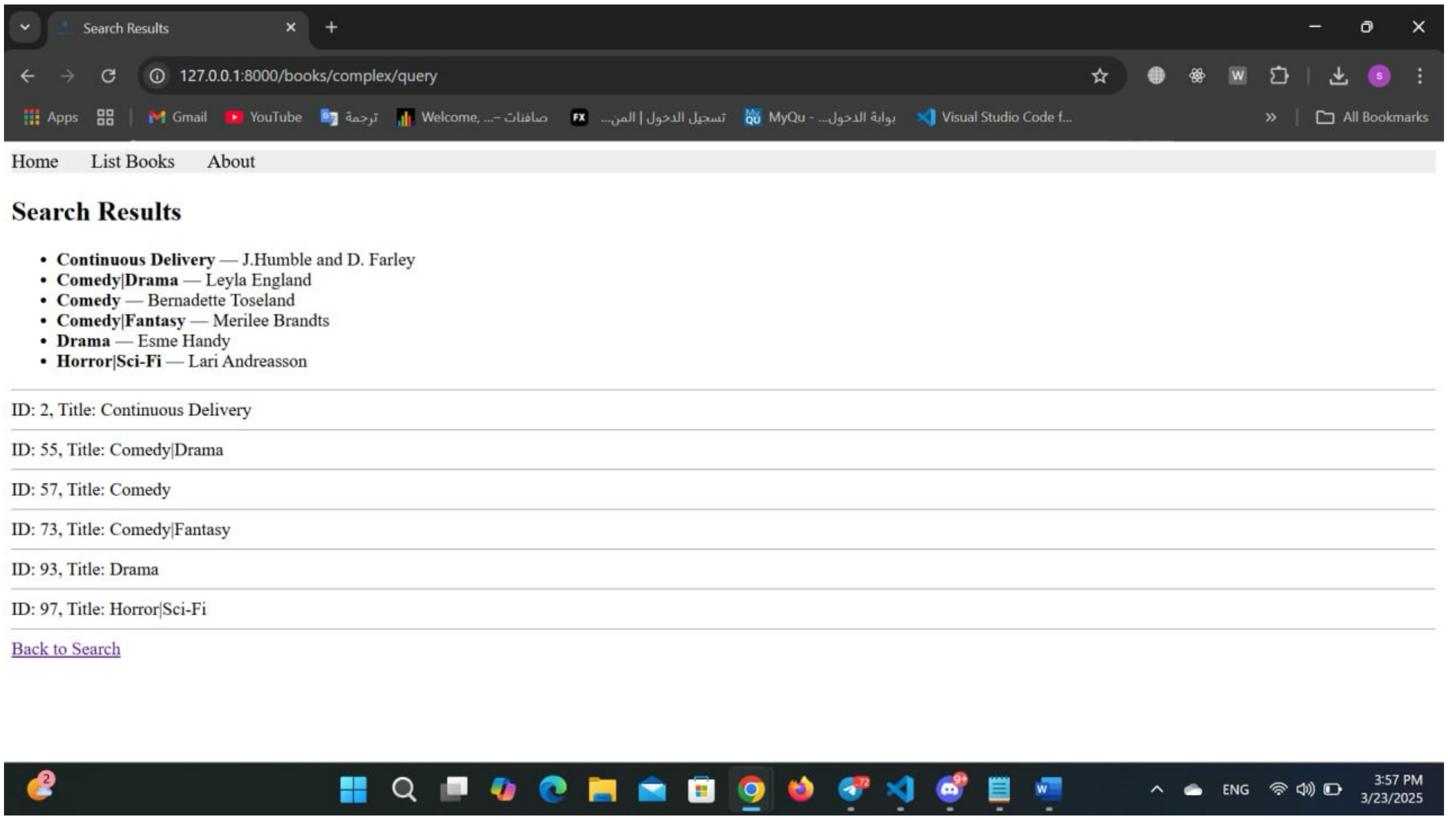
```python
def text_formatting(request):
   return render(request, 'bookmodule/html5/text_formatting.html')
def listing(request):
   return render(request, 'bookmodule/html5/listing.html')

def tables(request):
   return render(request, 'bookmodule/html5/tables.html')


def simple_query(request):
   mybooks=Book.objects.filter(title__icontains='and') # <- multiple objects
   return render(request, 'bookmodule/bookList.html', {'books':mybooks})
```

```
System check identified no issues (0 silenced).
March 22, 2025 - 23:09:59
Django version 5.1.6, using settings 'libraryproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[22/Mar/2025 23:10:18] "GET /books/simple/query HTTP/1.1" 301 0
[22/Mar/2025 23:10:18] "GET /books/simple/query/ HTTP/1.1" 200 1799
```

Task 4: Customize queries using common lookup conditions and keywords.



Keywords: and
Search in: ☑ Title ☑ Author



Search Results

- **Continuous Delivery** — J.Humble and D. Farley
- **Comedy|Drama** — Leyla England
- **Comedy** — Bernadette Toseland
- **Comedy|Fantasy** — Merilee Brandts
- **Drama** — Esme Handy
- **Horror|Sci-Fi** — Lari Andreasson

ID: 2, Title: Continuous Delivery

ID: 55, Title: Comedy|Drama

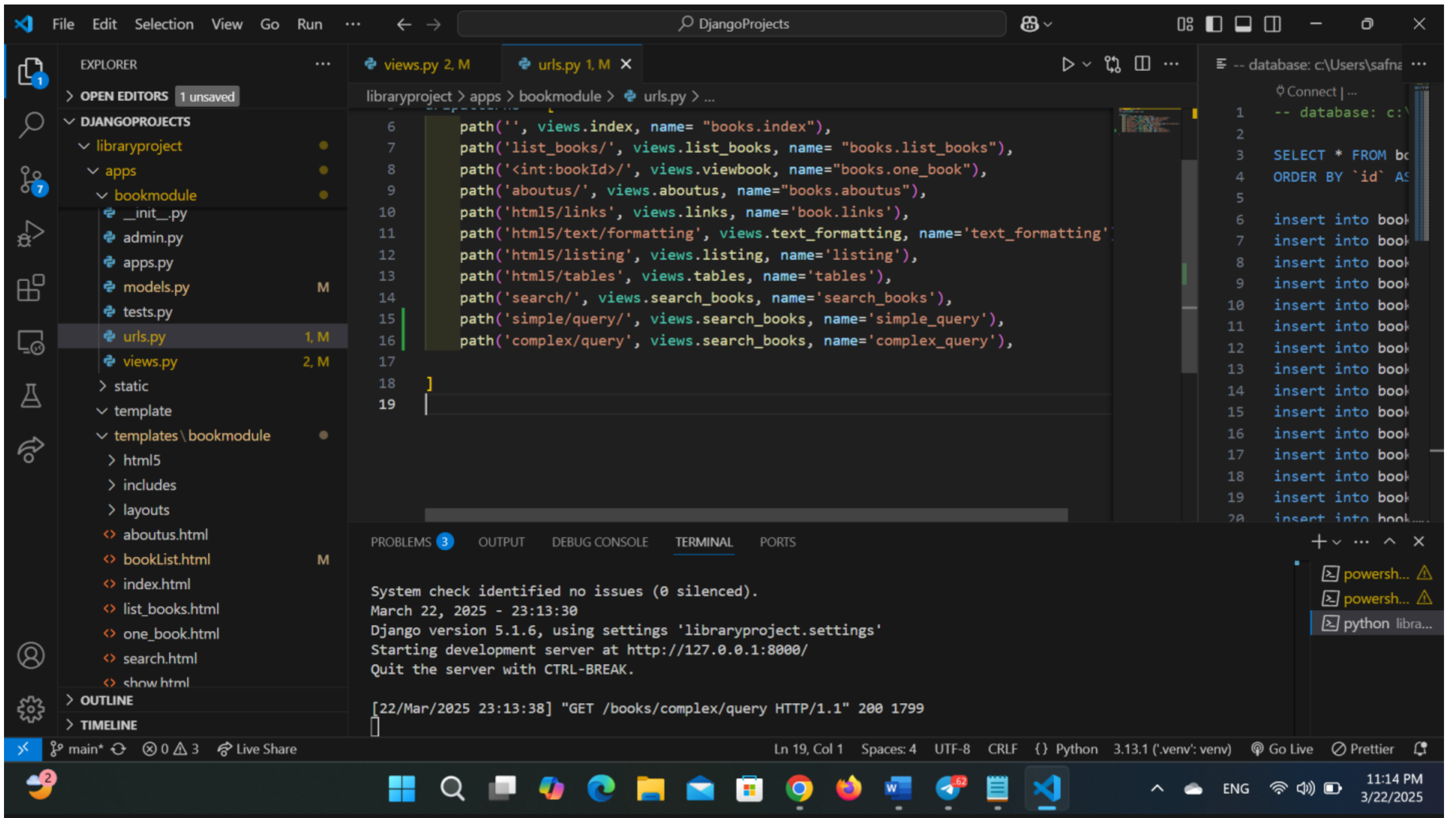
ID: 57, Title: Comedy

ID: 73, Title: Comedy|Fantasy

ID: 93, Title: Drama

ID: 97, Title: Horror|Sci-Fi

Back to Search

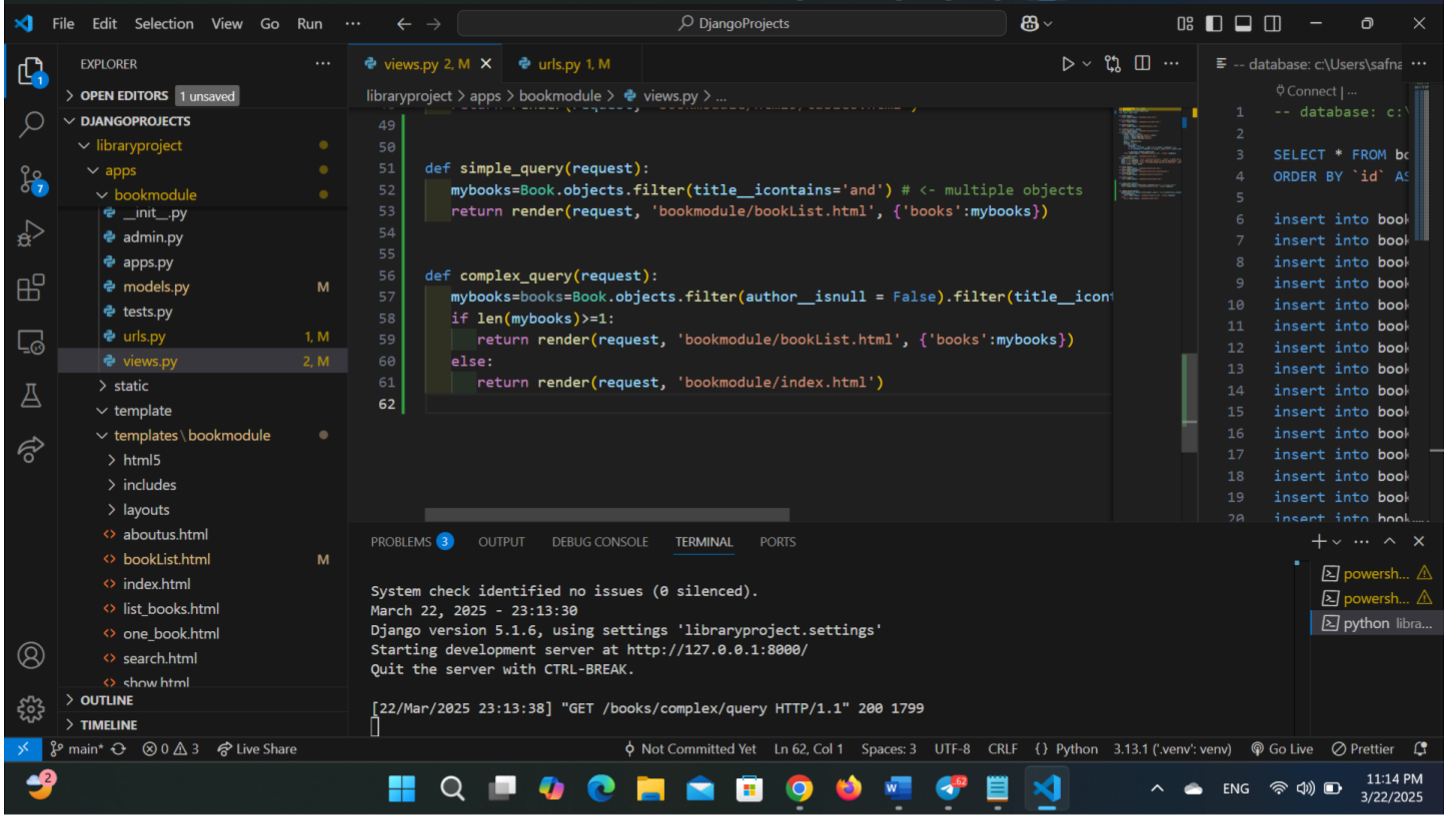


urls.py

```
    path('', views.index, name= "books.index"),
    path('list_books/', views.list_books, name= "books.list_books"),
    path('<int:bookId>/', views.viewbook, name="books.one_book"),
    path('aboutus/', views.aboutus, name="books.aboutus"),
    path('html5/links', views.links, name='book.links'),
    path('html5/text/formatting', views.text_formatting, name='text_formatting'
    path('html5/listing', views.listing, name='listing'),
    path('html5/tables', views.tables, name='tables'),
    path('search/', views.search_books, name='search_books'),
    path('simple/query/', views.search_books, name='simple_query'),
    path('complex/query', views.search_books, name='complex_query'),

]
```

```
-- database: c:\
SELECT * FROM bo
ORDER BY `id` AS
insert into book
```

```
System check identified no issues (0 silenced).
March 22, 2025 - 23:13:30
Django version 5.1.6, using settings 'libraryproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[22/Mar/2025 23:13:38] "GET /books/complex/query HTTP/1.1" 200 1799
```



views.py

```
def simple_query(request):
    mybooks=Book.objects.filter(title__icontains='and') # <- multiple objects
    return render(request, 'bookmodule/bookList.html', {'books':mybooks})


def complex_query(request):
    mybooks=books=Book.objects.filter(author__isnull = False).filter(title__icon
    if len(mybooks)>=1:
        return render(request, 'bookmodule/bookList.html', {'books':mybooks})
    else:
        return render(request, 'bookmodule/index.html')
```

```
System check identified no issues (0 silenced).
March 22, 2025 - 23:13:30
Django version 5.1.6, using settings 'libraryproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[22/Mar/2025 23:13:38] "GET /books/complex/query HTTP/1.1" 200 1799
```