

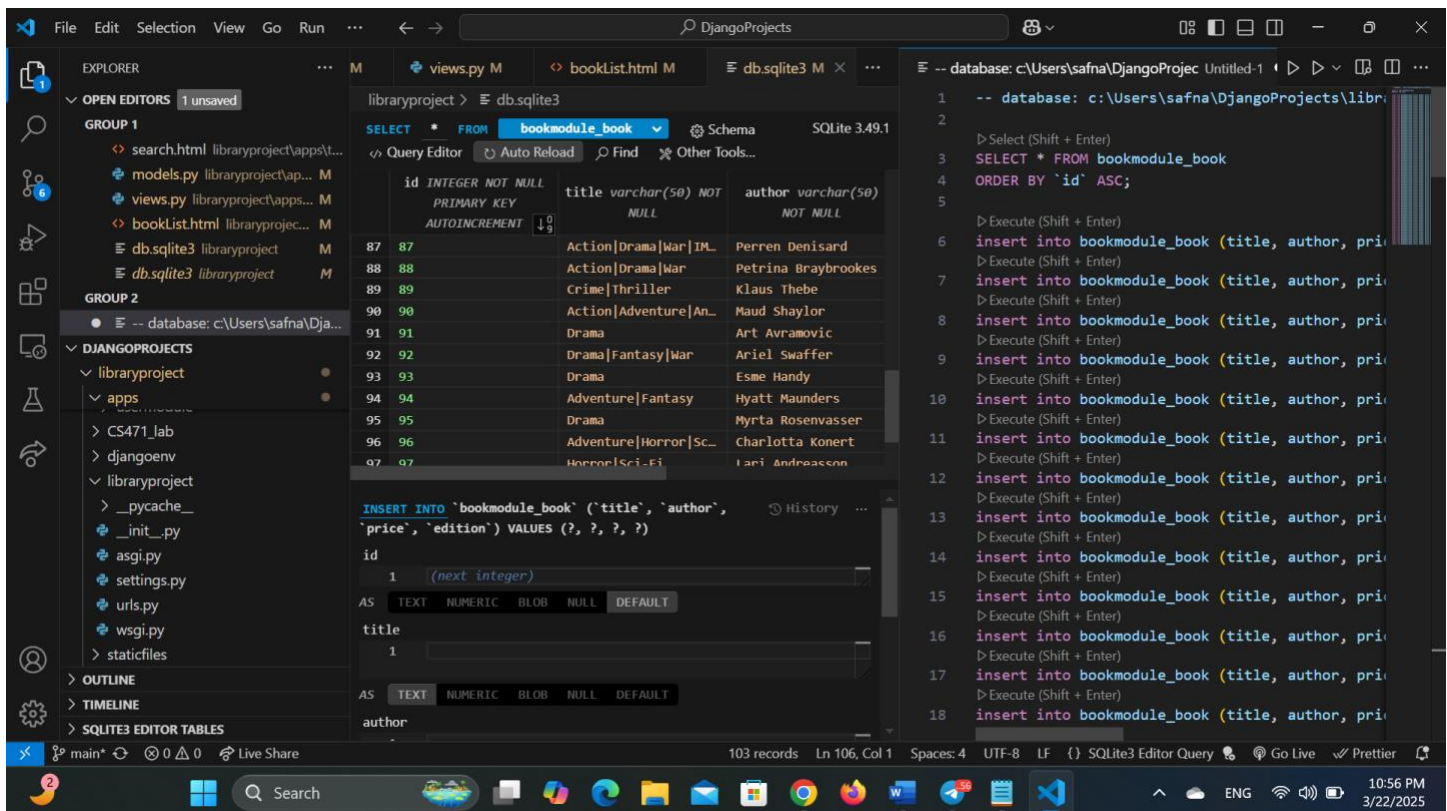
This lab session covers the basics of Django Models (Part 1). By the end of this session, students should have a very good understanding of how to build a basic model and how to access models in Django.

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.



The screenshot displays the Visual Studio Code interface for a Django project named 'DjangoProjects'. The Explorer panel on the left shows the project structure, including the 'apps' directory with 'bookmodule' and 'usermodule' subdirectories. The Query Editor in the center shows a table named 'bookmodule_book' with columns 'id', 'title', and 'author'. The SQL Editor on the right shows a query to insert data into the 'bookmodule_book' table.

id	title	author
87	Action Drama War IML	Perren Denisard
88	Action Drama War	Petrina Braybrookes
89	Crime Thriller	Klaus Thebe
90	Action Adventure An...	Maud Shaylor
91	Drama	Art Avramovic
92	Drama Fantasy War	Ariel Swaffer
93	Drama	Esme Handy
94	Adventure Fantasy	Hyatt Maunders
95	Drama	Myrta Rosenvasser
96	Adventure Horror Sc...	Charlotta Konert
97	Horror Sci-Fi	Lari Andreascen

```
-- database: c:\Users\safna\DjangoProjec
1
2
3 SELECT * FROM bookmodule_book
4 ORDER BY 'id' ASC;
5
6 insert into bookmodule_book (title, author, pri
7 insert into bookmodule_book (title, author, pri
8 insert into bookmodule_book (title, author, pri
9 insert into bookmodule_book (title, author, pri
10 insert into bookmodule_book (title, author, pri
11 insert into bookmodule_book (title, author, pri
12 insert into bookmodule_book (title, author, pri
13 insert into bookmodule_book (title, author, pri
14 insert into bookmodule_book (title, author, pri
15 insert into bookmodule_book (title, author, pri
16 insert into bookmodule_book (title, author, pri
17 insert into bookmodule_book (title, author, pri
18 insert into bookmodule_book (title, author, pri
```

The screenshot displays a VS Code editor interface for a Django project named 'libraryproject'. The Explorer panel on the left shows the project structure, including 'apps', 'templates', 'staticfiles', and 'db.sqlite3'. The central editor shows a table view of the 'books' table in the 'db.sqlite3' database. The table has columns: id, title, author, and price. The bottom panel shows the terminal output of the Django development server.

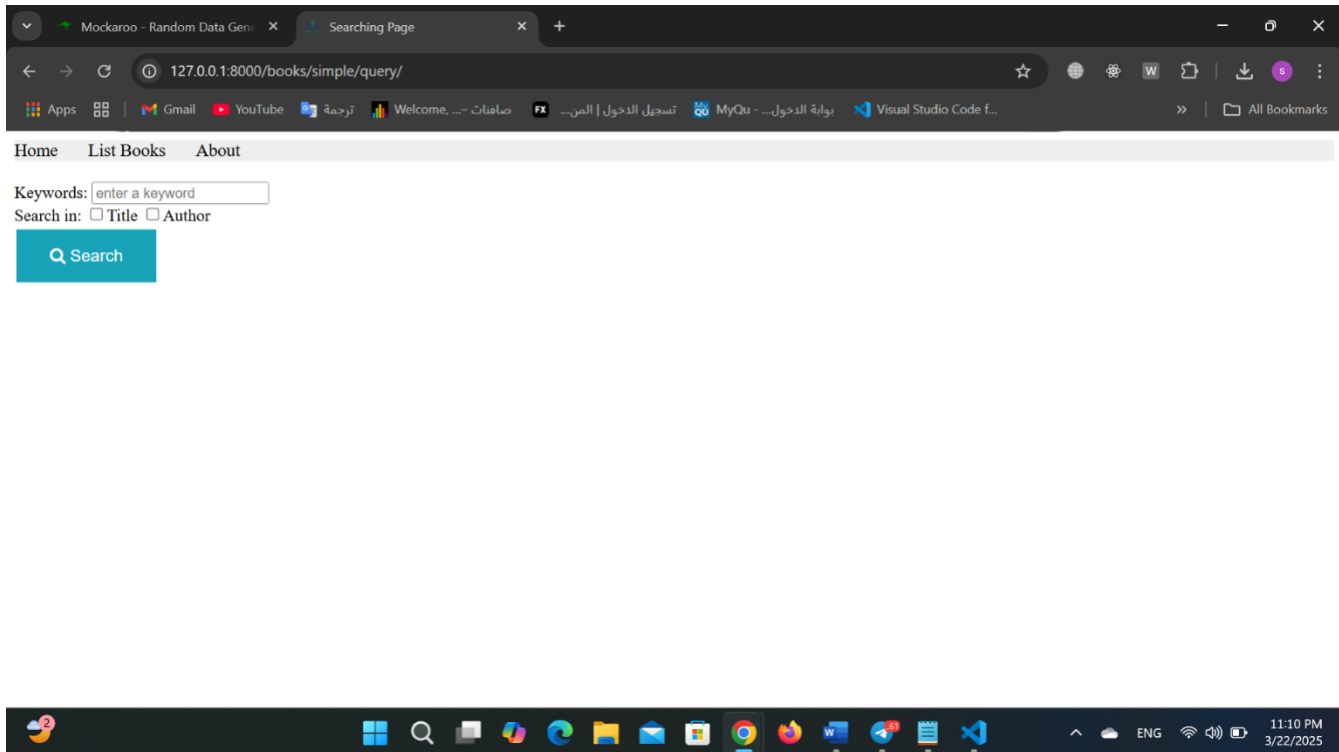
id	title	author	price
92	Drama\Fantasy\War	Ariel Swaffer	
93	Drama	Esme Handy	
94	Adventure\Fantasy	Hyatt Maunders	
95	Drama	Myrta Rosenvasser	
96	Adventure\Horror\Sci-Fi	Charlotta Konert	
97	Horror\Sci-Fi	Lari Andreasson	
98	Documentary	Truman Hopfer	
99	Drama	Phil Twiggins	
100	Thriller	Shelley Hallwell	

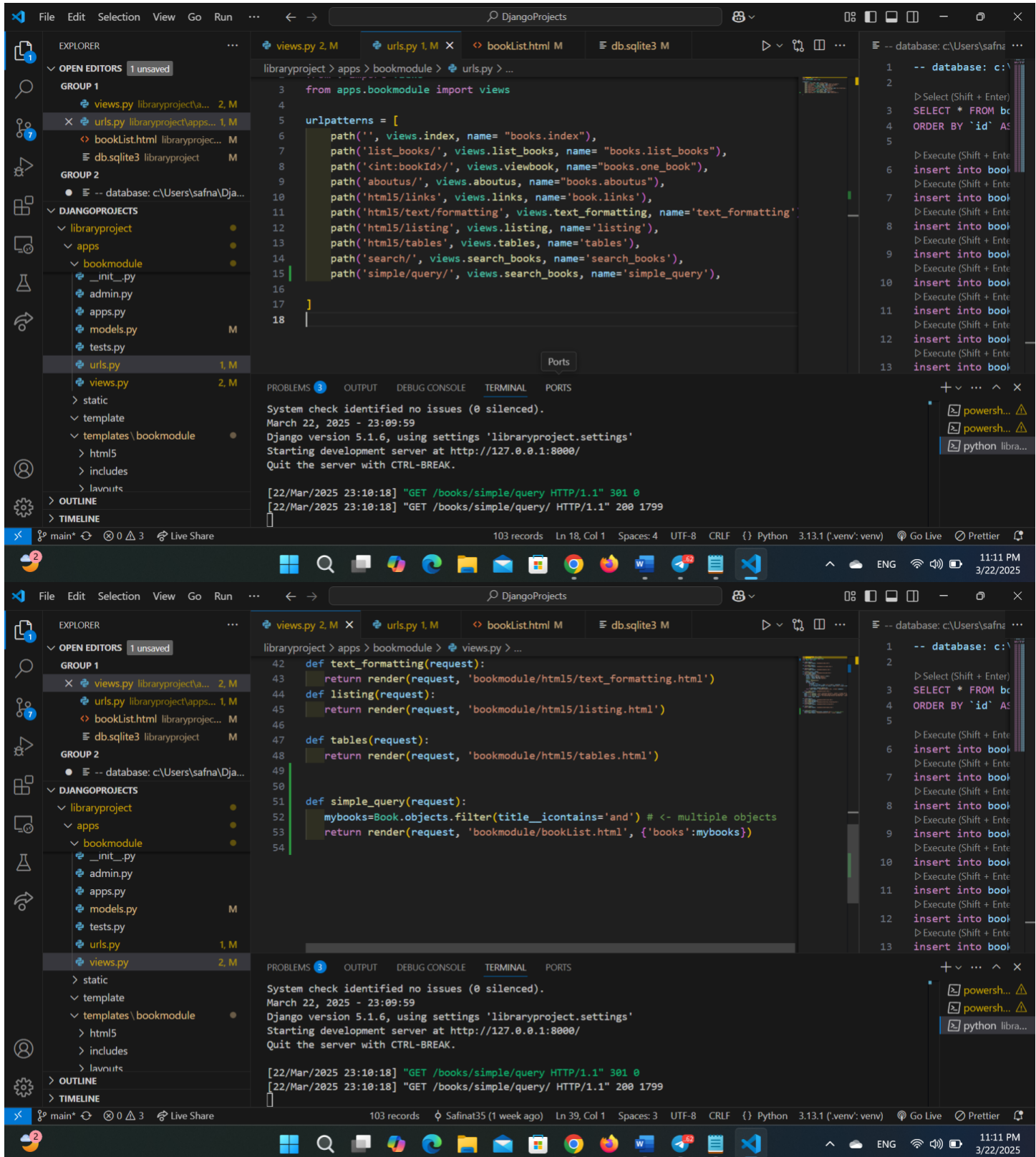
Terminal Output:

```
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.



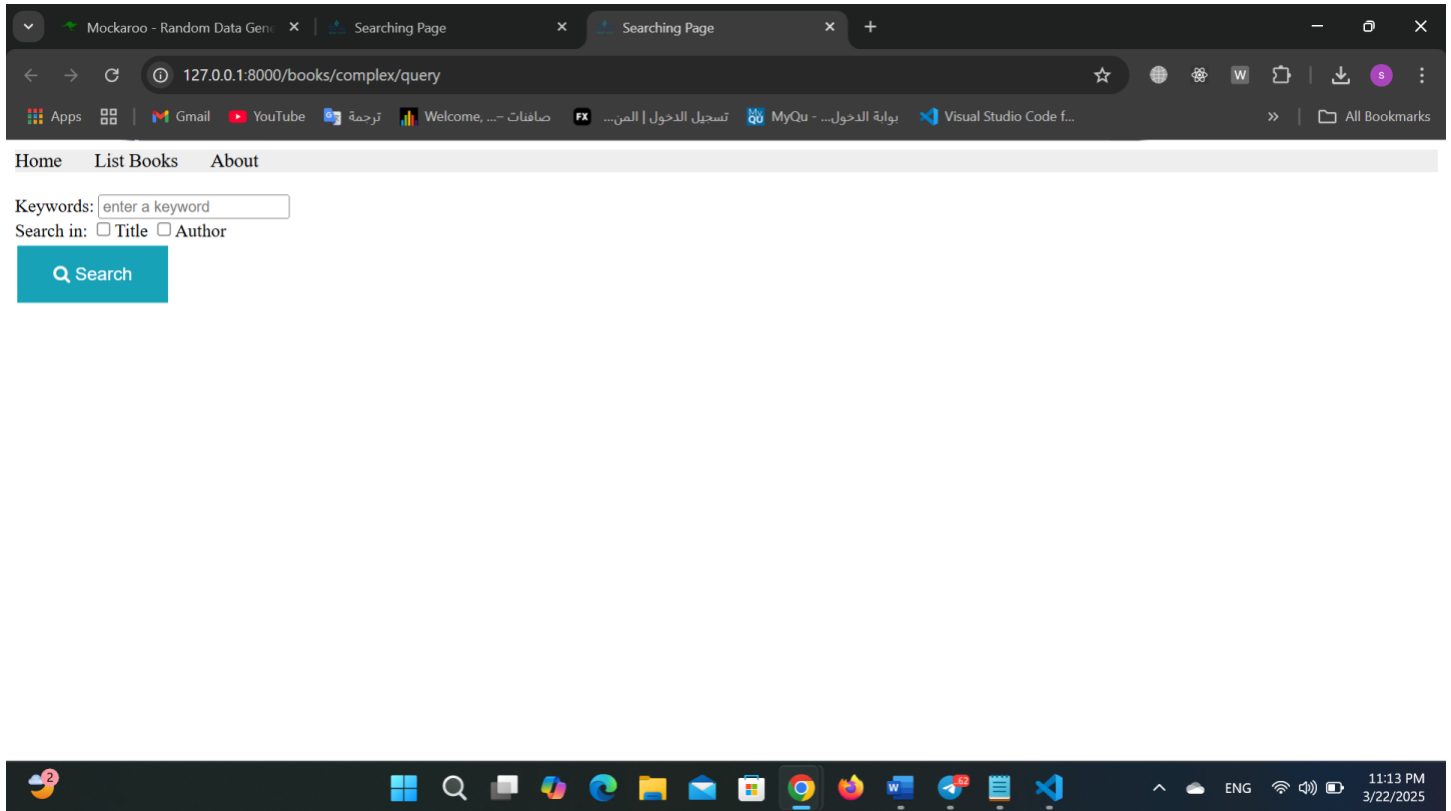


The screenshot displays the Visual Studio Code interface for a Django project named 'libraryproject'. The Explorer panel on the left shows the project structure, including the 'apps' directory and the 'bookmodule' app. The main editor window shows the 'urls.py' file for the 'bookmodule' app, which defines URL patterns for various views. The terminal window at the bottom shows the output of the Django development server, indicating that the server is running successfully at http://127.0.0.1:8000/.

```
libraryproject > apps > bookmodule > urls.py > ...  
3 from apps.bookmodule import views  
4  
5 urlpatterns = [  
6     path('', views.index, name= "books.index"),  
7     path('list_books/', views.list_books, name= "books.list_books"),  
8     path('<int:bookId>/', views.viewbook, name="books.one_book"),  
9     path('aboutus/', views.aboutus, name="books.aboutus"),  
10    path('html5/links', views.links, name='book.links'),  
11    path('html5/text/formatting', views.text_formatting, name='text_formatting'),  
12    path('html5/listing', views.listing, name='listing'),  
13    path('html5/tables', views.tables, name='tables'),  
14    path('search/', views.search_books, name='search_books'),  
15    path('simple/query/', views.search_books, name='simple_query'),  
16  
17 ]  
18
```

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



The top screenshot shows the `urls.py` file in the `bookmodule` app. It contains the following code:

```
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'),
```

The bottom screenshot shows the `views.py` file in the `bookmodule` app. It contains the following code:

```
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__icontains='and')
    if len(mybooks)>=1:
        return render(request, 'bookmodule/bookList.html', {'books':mybooks})
    else:
        return render(request, 'bookmodule/index.html')
```

Both screenshots show a terminal window with the following output:

```
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
```