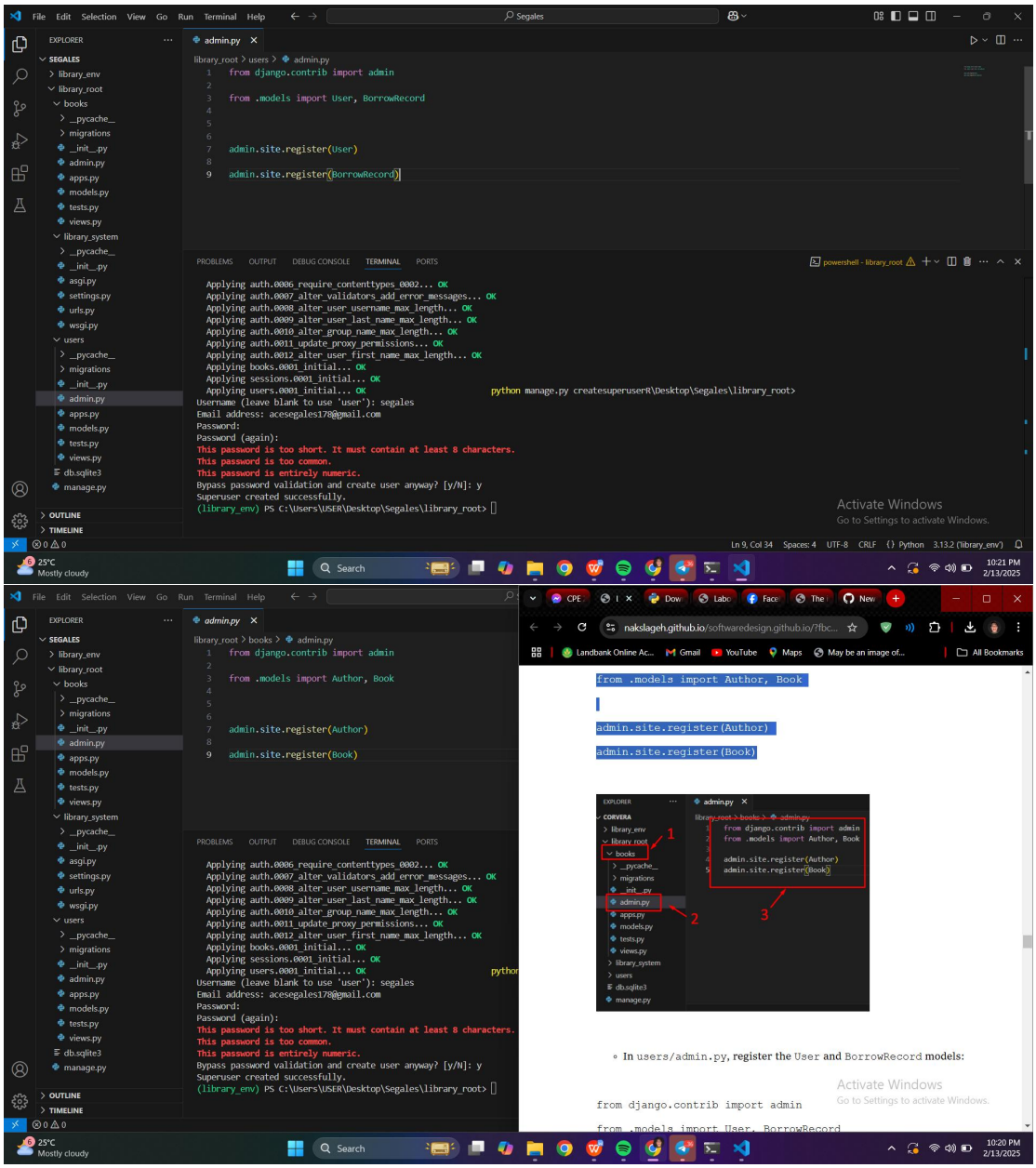
SEGALES, ACE NIÑO B.
BSCPE 2A

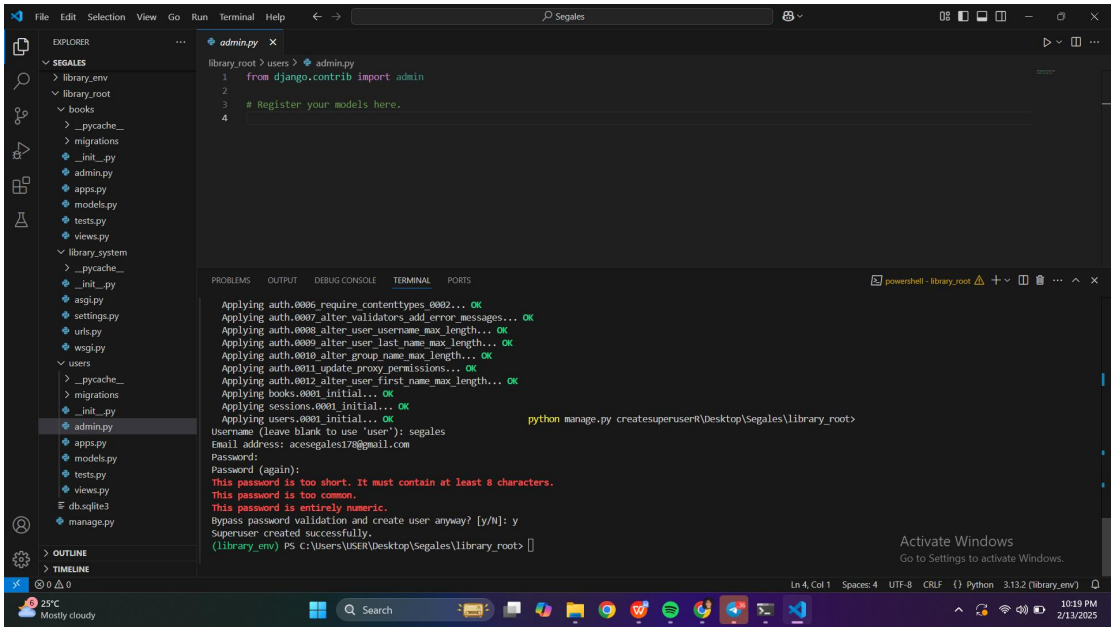
Laboratory Activity No. 2:

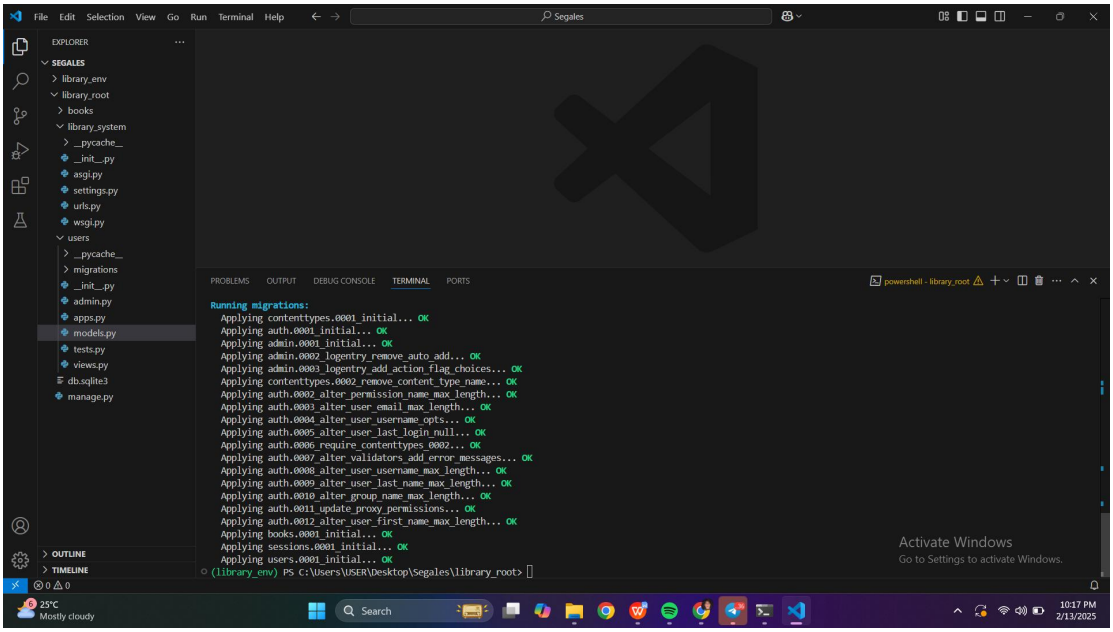
Laboratory Activity No. 2:

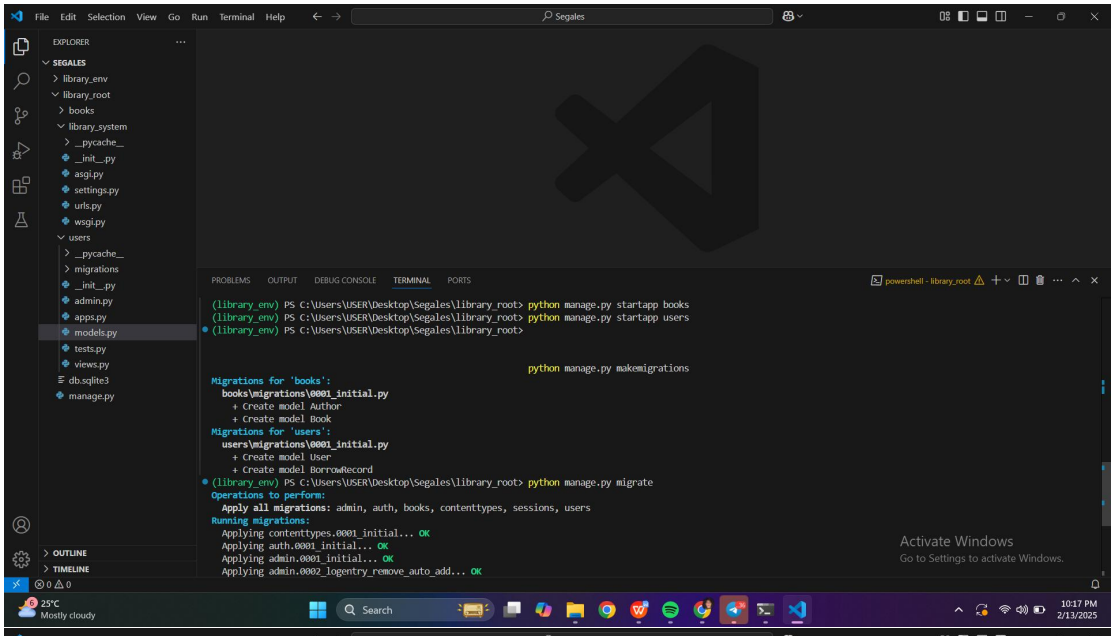
Topic belongs to: **Software Design and Database Systems**

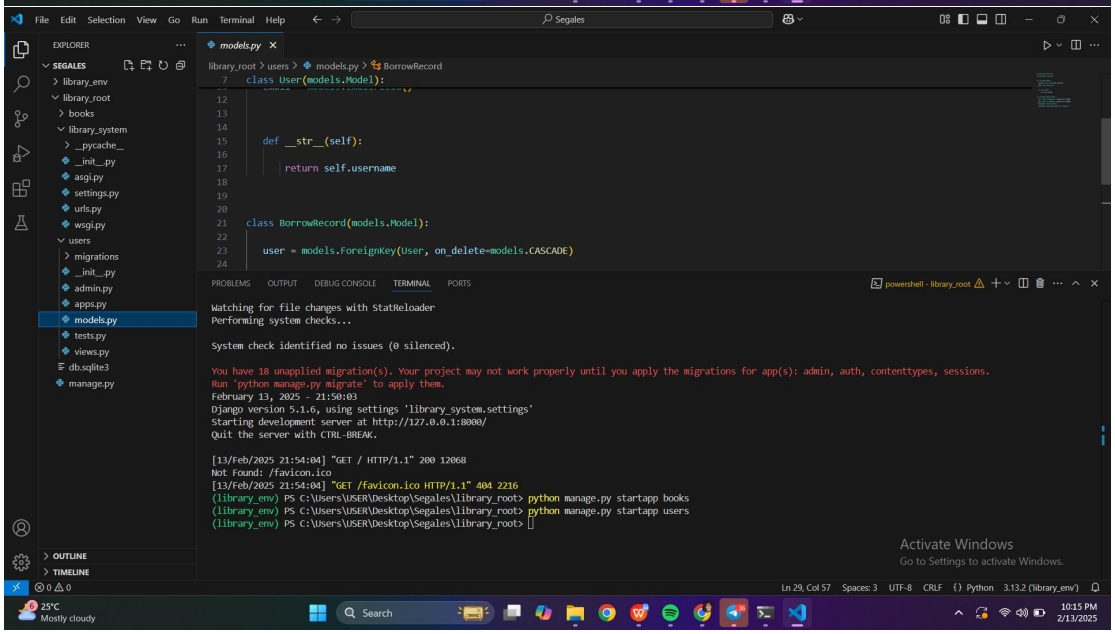
Title: *Designing the Database Schema for the Library Management System*

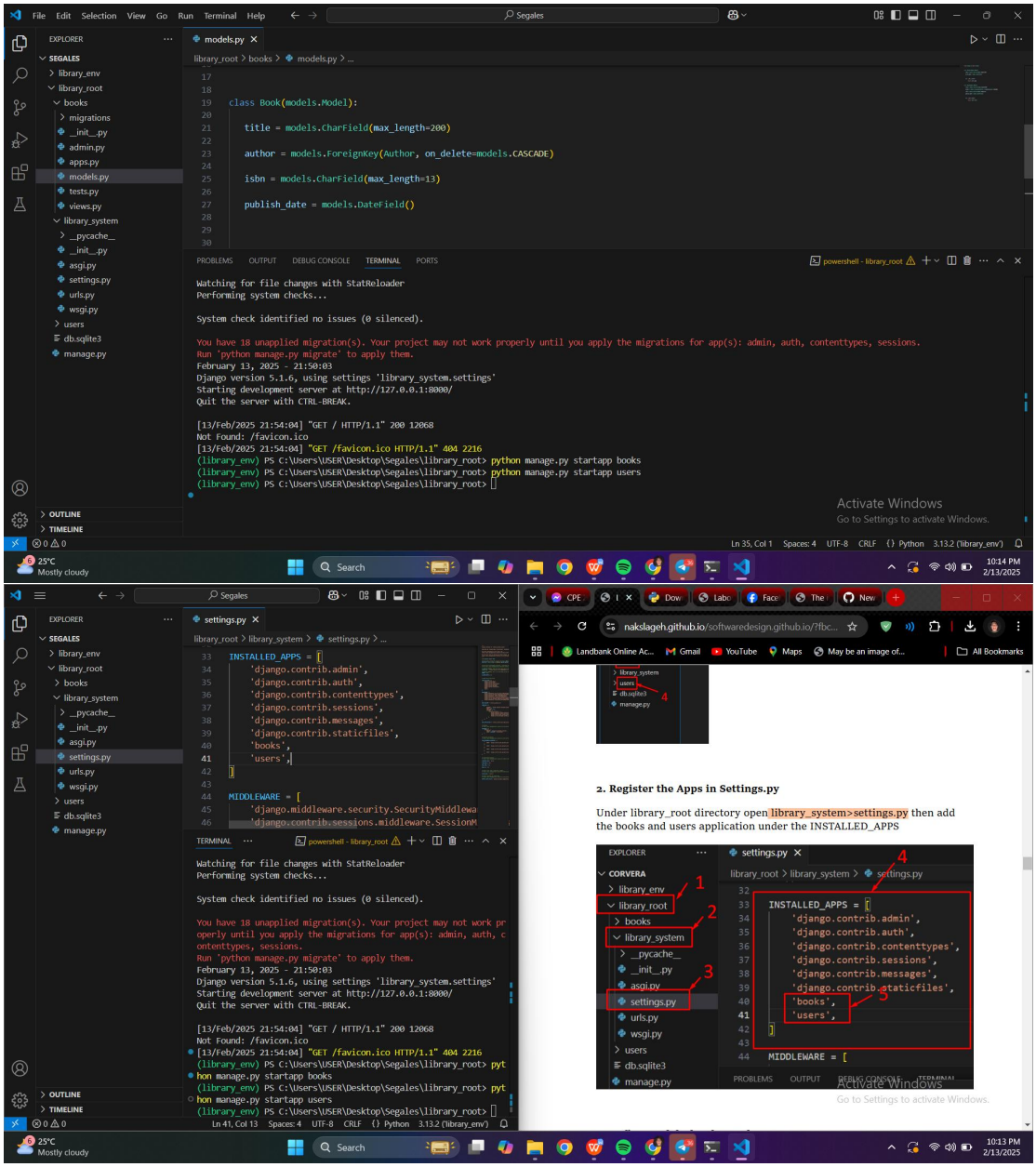



admin.py
from django.contrib import admin
# Register your models here.
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying auth.0010_alter_group_name_max_length... OK
Applying auth.0011_update_proxy_permissions... OK
Applying auth.0012_alter_user_first_name_max_length... OK
Applying books.0001_initial... OK
Applying sessions.0001_initial... OK
Applying users.0001_initial... OK
python manage.py createsuperuserR\Desktop\Segales\library_root>
Username (leave blank to use 'user'): segales
Email address: acesegales178@gmail.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
(library_env) PS C:\Users\USER\Desktop\Segales\library_root>
Activate Windows
Go to Settings to activate Windows.

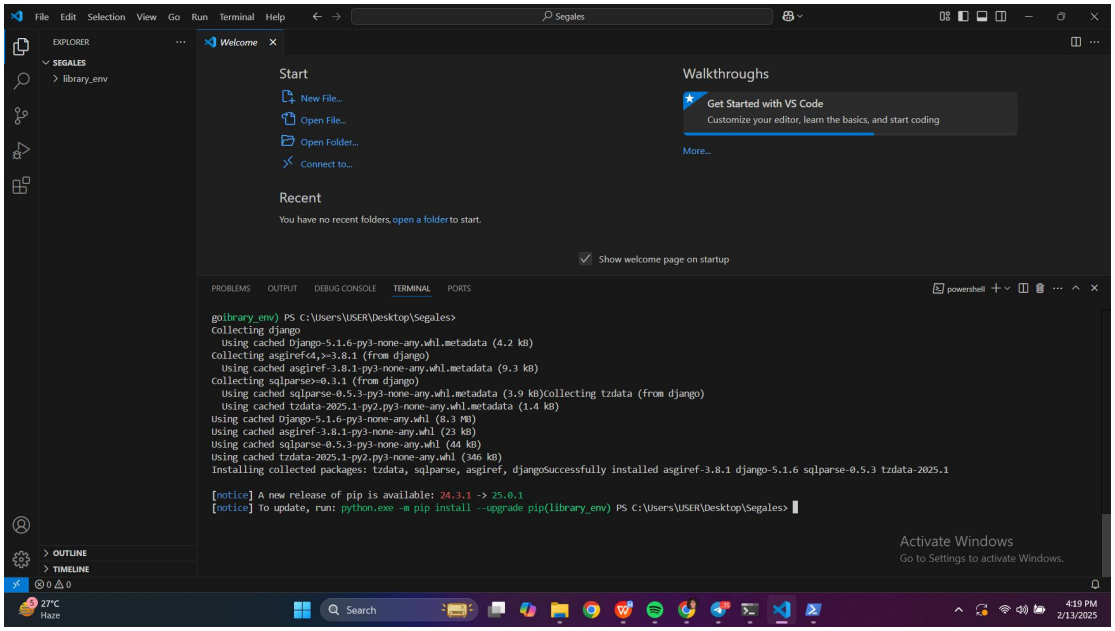

Running migrations:
Applying contenttypes.0001_initial... OK
Applying auth.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_logentry_remove_auto_add... OK
Applying admin.0003_logentry_add_action_flag_choices... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying auth.0010_alter_group_name_max_length... OK
Applying auth.0011_update_proxy_permissions... OK
Applying auth.0012_alter_user_first_name_max_length... OK
Applying books.0001_initial... OK
Applying sessions.0001_initial... OK
Applying users.0001_initial... OK
(library_env) PS C:\Users\USER\Desktop\Segales\library_root>
Activate Windows
Go to Settings to activate Windows.

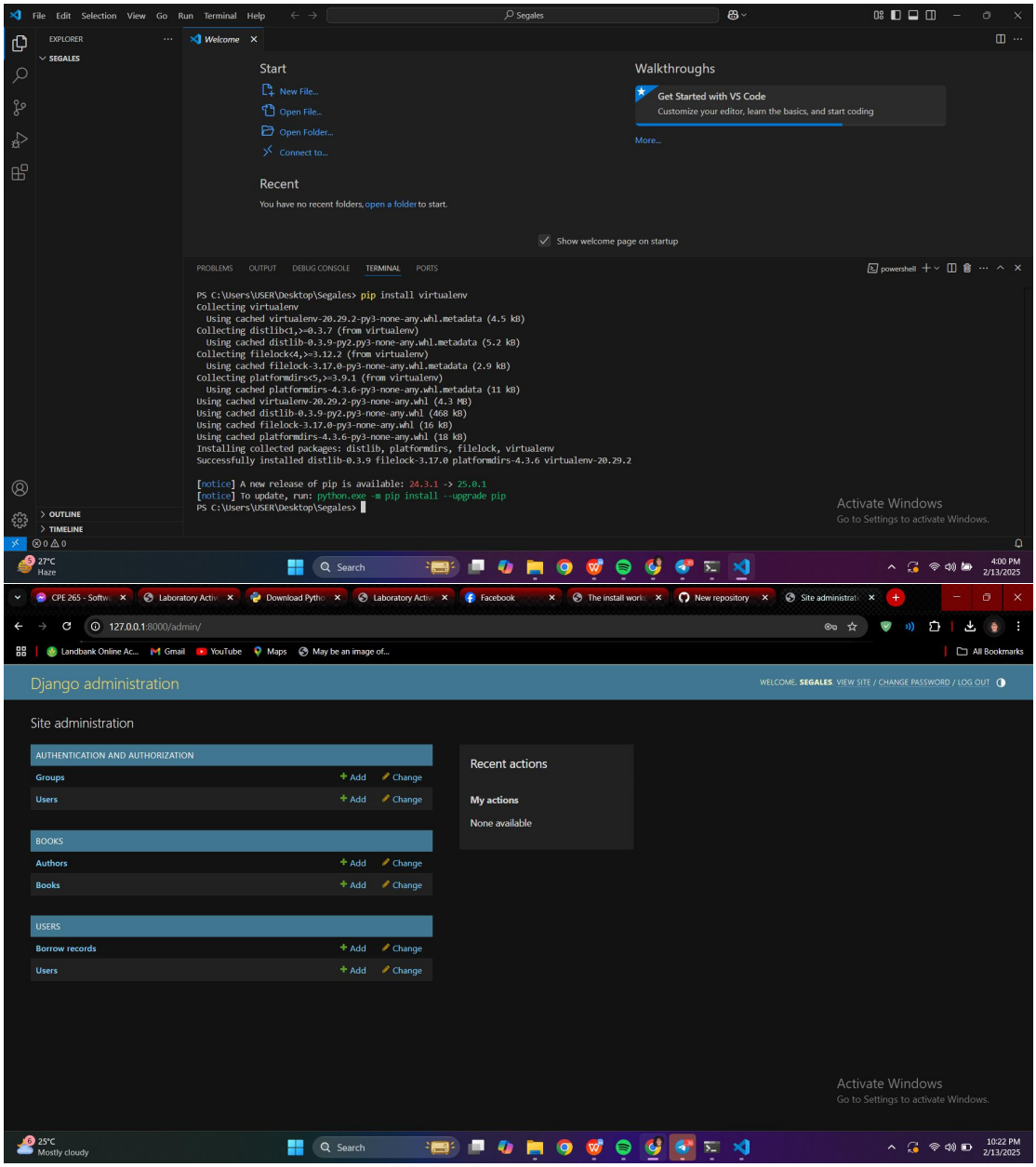

File Edit Selection View Go Run Terminal Help
Segales
EXPLORER
SEGALES
library_env
library_root
books
library_system
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
users
__pycache__
migrations
__init__.py
admin.py
apps.py
models.py
tests.py
views.py
db.sqlite3
manage.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - library_root
(library_env) PS C:\Users\USER\Desktop\Segales\library_root> python manage.py startapp books
(library_env) PS C:\Users\USER\Desktop\Segales\library_root> python manage.py startapp users
(library_env) PS C:\Users\USER\Desktop\Segales\library_root>
python manage.py makemigrations
Migrations for 'books':
books\migrations\0001_initial.py
+ Create model Author
+ Create model Book
Migrations for 'users':
users\migrations\0001_initial.py
+ Create model User
+ Create model BorrowRecord
(library_env) PS C:\Users\USER\Desktop\Segales\library_root> python manage.py migrate
Operations to perform:
Apply all migrations: admin, auth, books, contenttypes, sessions, users
Running migrations:
Applying contenttypes.0001_initial... OK
Applying auth.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_logentry_remove_auto_add... OK
Activate Windows
Go to Settings to activate Windows.
OUTLINE
TIMELINE
25°C
Mostly cloudy
Search
10:17 PM
2/13/2025


File Edit Selection View Go Run Terminal Help
Segales
models.py
library_root > users > models.py > BorrowRecord
class User(models.Model):
def __str__(self):
return self.username
class BorrowRecord(models.Model):
user = models.ForeignKey(User, on_delete=models.CASCADE)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - library_root
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 13, 2025 - 21:50:03
Django version 5.1.6, using settings 'library_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[13/Feb/2025 21:54:04] "GET / HTTP/1.1" 200 12068
Not Found: /favicon.ico
[13/Feb/2025 21:54:04] "GET /favicon.ico HTTP/1.1" 404 2216
(library_env) PS C:\Users\USER\Desktop\Segales\library_root> python manage.py startapp books
(library_env) PS C:\Users\USER\Desktop\Segales\library_root> python manage.py startapp users
(library_env) PS C:\Users\USER\Desktop\Segales\library_root>
Activate Windows
Go to Settings to activate Windows.
Ln 29, Col 57 Spaces: 3 UTF-8 CRLF {} Python 3.13.2 ('library_env')
25°C
Mostly cloudy
Search
10:15 PM
2/13/2025


File Edit Selection View Go Run Terminal Help
Segales
EXPLORER
SEGALES
library_env
library_root
books
migrations
__init__.py
admin.py
apps.py
models.py
tests.py
views.py
library_system
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
users
db.sqlite3
manage.py
models.py
library_root > books > models.py > ...
class Book(models.Model):
title = models.CharField(max_length=200)
author = models.ForeignKey(Author, on_delete=models.CASCADE)
isbn = models.CharField(max_length=13)
publish_date = models.DateField()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell - library_root
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 13, 2025 - 21:50:03
Django version 5.1.6, using settings 'library_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[13/Feb/2025 21:54:04] "GET / HTTP/1.1" 200 12068
Not Found: /favicon.ico
[13/Feb/2025 21:54:04] "GET /favicon.ico HTTP/1.1" 404 2216
(library_env) PS C:\Users\USER\Desktop\Segales\library_root> python manage.py startapp books
(library_env) PS C:\Users\USER\Desktop\Segales\library_root> python manage.py startapp users
(library_env) PS C:\Users\USER\Desktop\Segales\library_root>
Activate Windows
Go to Settings to activate Windows.
OUTLINE
TIMELINE
Ln 35, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.2 ('library_env')
25°C
Mostly cloudy
Search
10:14 PM
2/13/2025

Segales
EXPLORER
SEGALES
library_env
library_root
books
library_system
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
users
db.sqlite3
manage.py
settings.py
library_root > library_system > settings.py > ...
INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'books',
'users',
]
MIDDLEWARE = [
'django.middleware.security.SecurityMiddlewa
'django.contrib.sessions.middleware.SessionM
TERMINAL
powershell - library_root
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
You have 18 unapplied migration(s). Your project may not work pr
operly until you apply the migrations for app(s): admin, auth, c
ontenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 13, 2025 - 21:50:03
Django version 5.1.6, using settings 'library_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[13/Feb/2025 21:54:04] "GET / HTTP/1.1" 200 12068
Not Found: /favicon.ico
[13/Feb/2025 21:54:04] "GET /favicon.ico HTTP/1.1" 404 2216
(library_env) PS C:\Users\USER\Desktop\Segales\library_root> pyt
hon manage.py startapp books
(library_env) PS C:\Users\USER\Desktop\Segales\library_root> pyt
hon manage.py startapp users
(library_env) PS C:\Users\USER\Desktop\Segales\library_root>
OUTLINE
TIMELINE
Ln 41, Col 13 Spaces: 4 UTF-8 CRLF {} Python 3.13.2 ('library_env')
nakslageh.github.io/softwaredesign.github.io/?fbc...
Landbank Online Ac...
Gmail
YouTube
Maps
May be an image of...
All Bookmarks
2. Register the Apps in Settings.py
Under library_root directory open library_system>settings.py then add the books and users application under the INSTALLED_APPS
EXPLORER
CORVERA
library_env
library_root
books
library_system
__pycache__
__init__.py
asgi.py
settings.py
urls.py
wsgi.py
users
db.sqlite3
manage.py
1
2
3
4
5
settings.py
library_root > library_system > settings.py
INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'books',
'users',
]
MIDDLEWARE = [
PROBLEMS OUTPUT
Activate Windows
Go to Settings to activate Windows.
25°C
Mostly cloudy
Search
10:13 PM
2/13/2025


File Edit Selection View Go Run Terminal Help
Segales
EXPLORER
SEGALES
library_env
Welcome
Start
New File...
Open File...
Open Folder...
Connect to...
Recent
You have no recent folders, open a folder to start.
Walkthroughs
Get Started with VS Code
Customize your editor, learn the basics, and start coding
More...
Show welcome page on startup
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
go(library_env) PS C:\Users\USER\Desktop\Segales>
Collecting django
Using cached Django-5.1.6-py3-none-any.whl.metadata (4.2 kB)
Collecting asgiref<4,>=3.8.1 (from django)
Using cached asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from django)
Using cached sqlparse-0.5.3-py3-none-any.whl.metadata (3.9 kB)Collecting tzdata (from django)
Using cached tzdata-2025.1-py2.py3-none-any.whl.metadata (1.4 kB)
Using cached Django-5.1.6-py3-none-any.whl (8.3 MB)
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.3-py3-none-any.whl (44 kB)
Using cached tzdata-2025.1-py2.py3-none-any.whl (346 kB)
Installing collected packages: tzdata, sqlparse, asgiref, djangoSuccessfully installed asgiref-3.8.1 django-5.1.6 sqlparse-0.5.3 tzdata-2025.1
[notice] A new release of pip is available: 24.3.1 -> 25.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip(library_env) PS C:\Users\USER\Desktop\Segales>
Activate Windows
Go to Settings to activate Windows.
OUTLINE
TIMELINE
27°C
Haze
Search
4:19 PM
2/13/2025



Follow-Up Questions:

1. What is the purpose of using ForeignKey in Django models?

A foreign key in Django models establishes a many-to-one relationship between two models, maintains data integrity by ensuring valid relationships, facilitates efficient querying of related data, and allows for complex relationships through multiple foreign keys in a single model.

2. How does Django's ORM simplify database interaction?

Django's ORM simplifies database interaction by allowing developers to use intuitive Python code instead of raw SQL, abstracting complex queries, enabling model definition through Python classes, automatically generating SQL commands, and facilitating straightforward operations for creating, retrieving, updating, and deleting records.

Findings

Django's Object-Relational Mapping (ORM) provides a powerful framework for database interaction, significantly reducing the complexity associated with raw SQL queries. By using Python classes to define database models, developers can easily manage data structures and relationships. The ORM automatically generates the necessary SQL commands, which streamlines the process of database management. Additionally, the ORM supports a wide range of database operations through simple and readable syntax, enhancing productivity and reducing the likelihood of errors.

Summary

In summary, Django's ORM serves as a robust tool for developers, enabling them to interact with databases in a more efficient and user-friendly manner. It abstracts the underlying SQL complexity, allowing for a focus on application logic rather than database intricacies. The ability to define models as Python classes and perform CRUD operations with minimal code makes Django an attractive choice for web development.

Conclusion

In conclusion, Django's ORM simplifies database interaction by providing an intuitive interface for managing database operations. Its abstraction of SQL, automatic command generation, and ease of use contribute to faster development cycles and improved code maintainability. As a result, Django's ORM not only enhances developer productivity but also promotes best practices in database management, making it a valuable asset for any web application project.