

Name: Louie Jay H. Larong

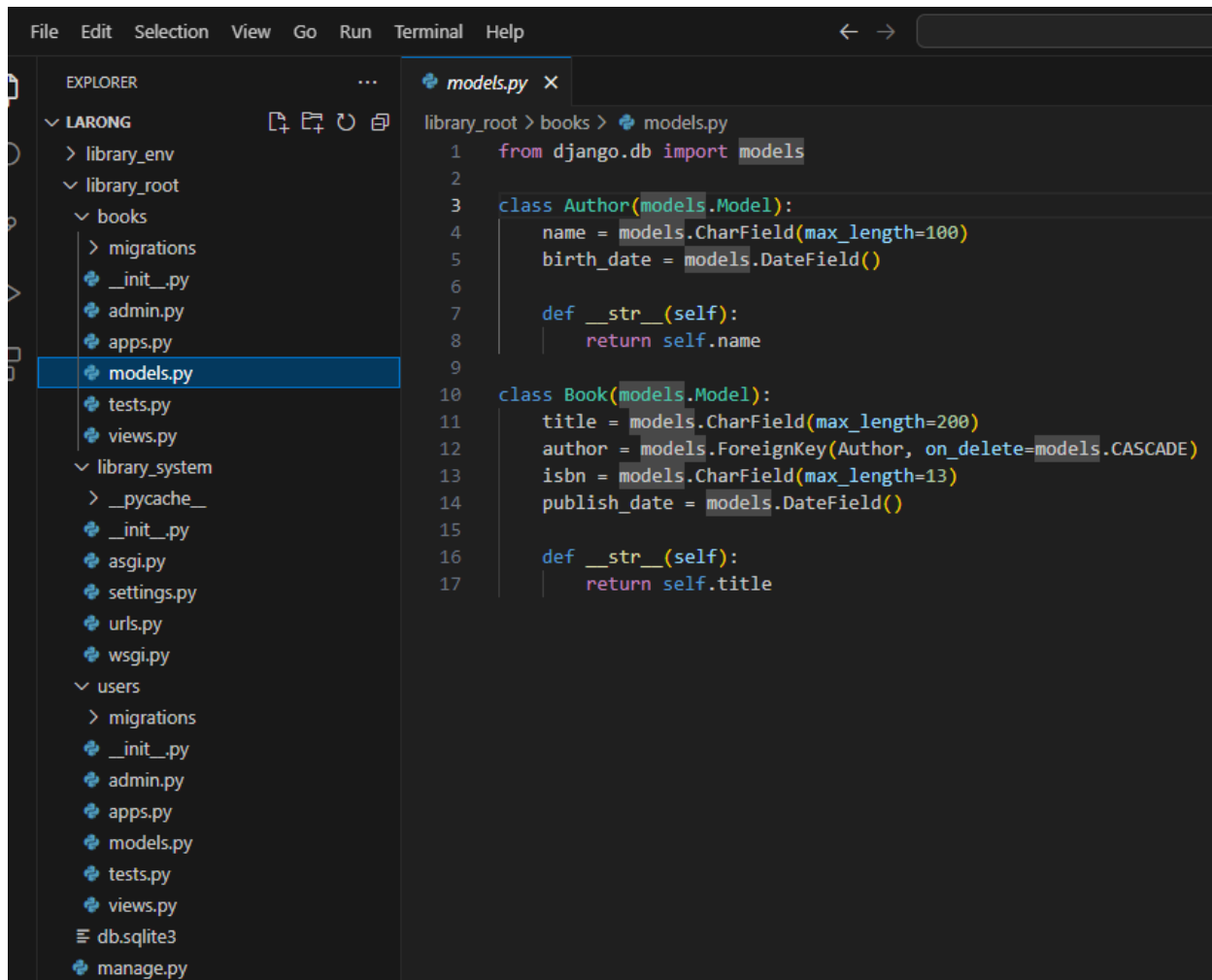
Section: BSCpE - 2A

Laboratory Activity No. 2

Create Django Apps:

```
PS C:\Users\EB204_02\Desktop\LARONG> cd library_root
PS C:\Users\EB204_02\Desktop\LARONG\library_root> python manage.py startapp books
PS C:\Users\EB204_02\Desktop\LARONG\library_root> python manage.py startapp users
```

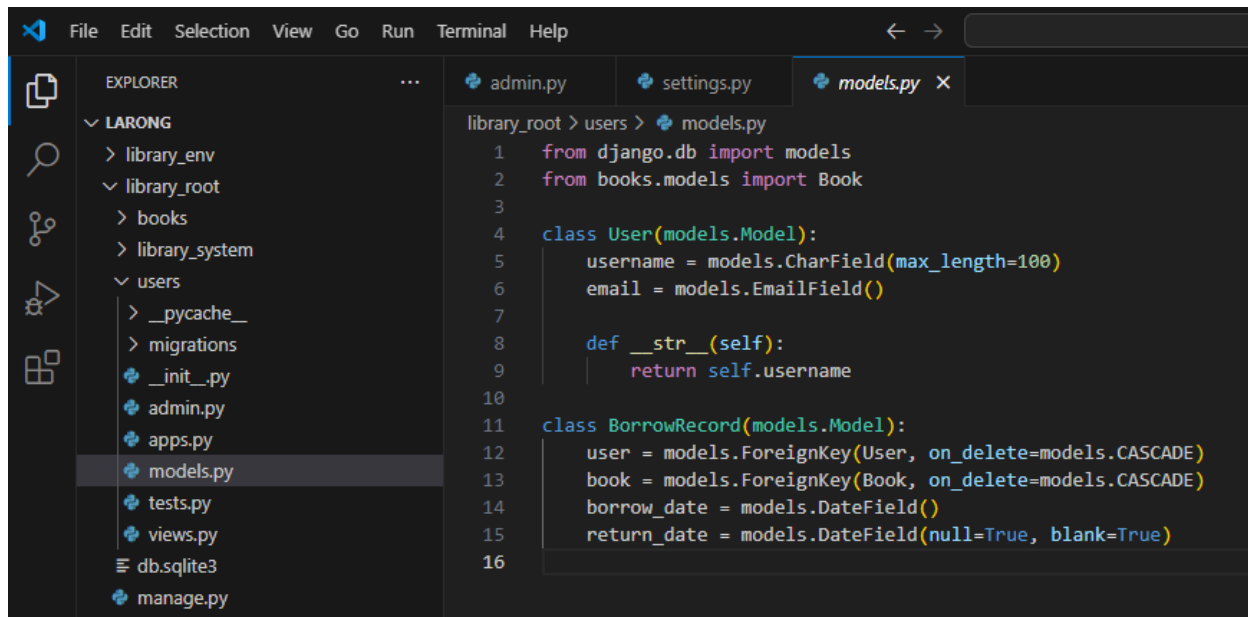
Define Models for the Book App:



```
File Edit Selection View Go Run Terminal Help
EXPLORER
  LARONG
    > 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
        > migrations
        > __init__.py
        > admin.py
        > apps.py
        > models.py
        > tests.py
        > views.py
      > db.sqlite3
      > manage.py

models.py
library_root > books > models.py
1  from django.db import models
2
3  class Author(models.Model):
4      name = models.CharField(max_length=100)
5      birth_date = models.DateField()
6
7      def __str__(self):
8          return self.name
9
10 class Book(models.Model):
11     title = models.CharField(max_length=200)
12     author = models.ForeignKey(Author, on_delete=models.CASCADE)
13     isbn = models.CharField(max_length=13)
14     publish_date = models.DateField()
15
16     def __str__(self):
17         return self.title
```

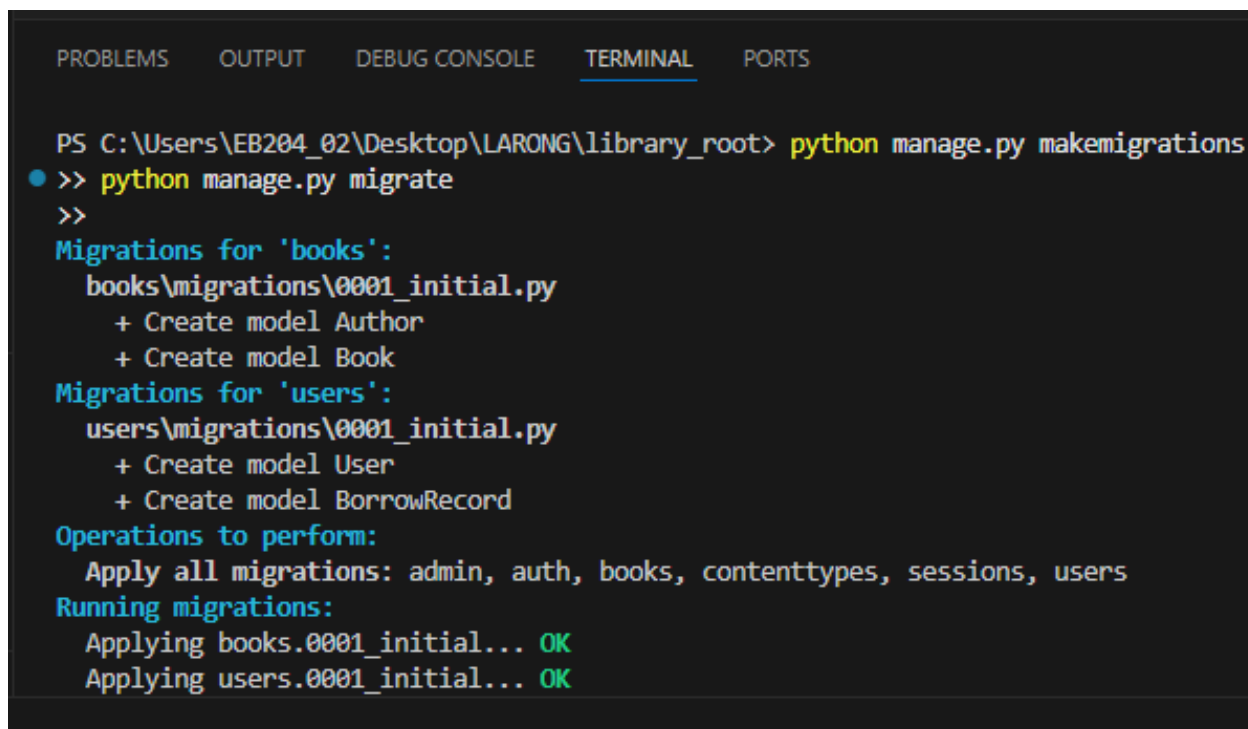
Define Models for Users App:



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure for 'LARONG'. The 'users' directory is expanded, showing files like __init__.py, admin.py, apps.py, models.py (selected), tests.py, and views.py. The main editor window shows the content of 'models.py' in the 'users' subdirectory. The code defines two Django models: 'User' and 'BorrowRecord'.

```
library_root > users > models.py
1  from django.db import models
2  from books.models import Book
3
4  class User(models.Model):
5      username = models.CharField(max_length=100)
6      email = models.EmailField()
7
8      def __str__(self):
9          return self.username
10
11 class BorrowRecord(models.Model):
12     user = models.ForeignKey(User, on_delete=models.CASCADE)
13     book = models.ForeignKey(Book, on_delete=models.CASCADE)
14     borrow_date = models.DateField()
15     return_date = models.DateField(null=True, blank=True)
16
```

Apply Migrations:



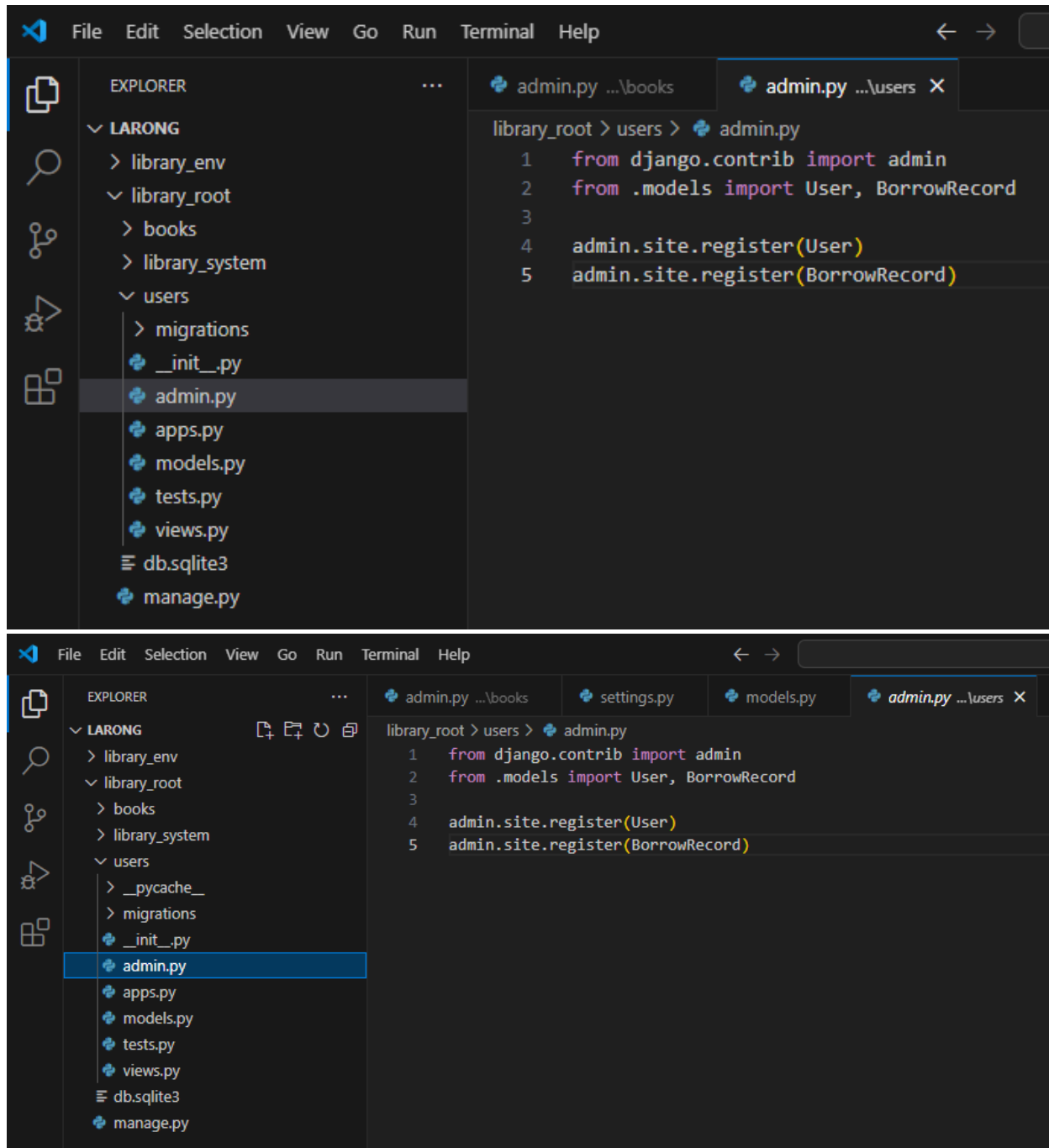
The screenshot shows the terminal window in VS Code. The user has executed the following commands in the 'library_root' directory:

```
PS C:\Users\EB204_02\Desktop\LARONG\library_root> python manage.py makemigrations
>> python manage.py migrate
>>
```

The output of the 'python manage.py migrate' command is as follows:

```
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
Operations to perform:
  Apply all migrations: admin, auth, books, contenttypes, sessions, users
Running migrations:
  Applying books.0001_initial... OK
  Applying users.0001_initial... OK
```

Register Models in Admin Panel:



Run the Development Server:

```
PS C:\Users\EB204_02\Desktop\LARONG\library_root> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

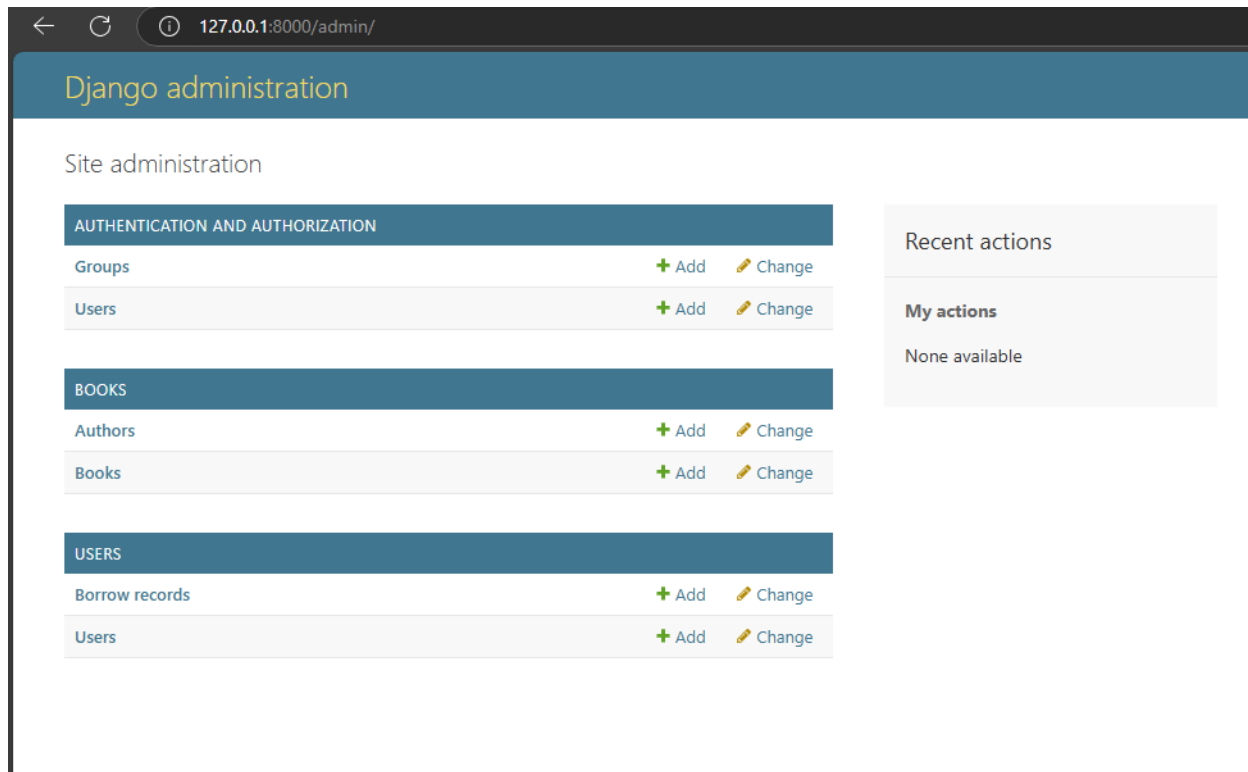
System check identified no issues (0 silenced).
February 05, 2025 - 14:54:14
Django version 5.1.5, using settings 'library_system.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Access Admin Panel:

The image shows the Django administration interface. The top part is a login form with a dark blue header containing the text "Django administration" and a moon icon. The form has two input fields: "Username:" with the value "eb204_02" and "Password:" with masked characters. Below the password field is a "Log in" button. The bottom part of the image shows the browser address bar with the URL "127.0.0.1:8000/admin/". Below the browser, the Django administration interface is displayed. It has a dark blue header with "Django administration". The main content area is titled "Site administration" and contains a table with the following data:

AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add ✎ Change
Users	+ Add ✎ Change

To the right of the table is a sidebar with two sections: "Recent actions" and "My actions". The "My actions" section shows "None available".



Follow-Up Questions:

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

Answer: There are a handful of purposes of using a ForeignKey some of which are; creating relationships between models, maintaining data integrity, enabling efficient queries, and more.

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

Answer: Django's Object-Relational Mapping (ORM) allows efficient and easy interaction between databases by allowing developers to work with databases using Python objects instead of writing raw SQL queries.

Findings:

-Django's ORM offers versatile use and can boost the efficiency of creation of your web domain.

Summary:

-In summary, Laboratory Exercise 2 highlights Django's ORM, how it's used and how impactful it is towards web designing.

Conclusion:

-Laboratory Exercise 2 focuses on designing the overall UI of the website and adds more information on how to use Django for web development.