# Guidance for Django Project: BookStore

This document provides step-by-step guidance for creating an exciting Django project named 'BookStore.' The project includes SQLite database interactions and migrations while introducing interesting features and relationships.

## 1. Setting Up the Project

- Create a new Django project named 'BookStore'.  
- Inside it, create an app called 'library'.  
- Register the app in the 'INSTALLED\_APPS' section of your project’s settings.py.

## 2. Database Models

Define the following models in the 'library' app:  
  
1. \*\*Author\*\*:  
 - Fields: First name, last name, and birth date.  
2. \*\*Genre\*\*:  
 - Use a `ChoiceField` to define predefined genre categories (e.g., Fiction, Non-Fiction, etc.).  
3. \*\*Book\*\*:  
 - Fields: Title, publication date, a `ManyToManyField` linking books to authors, and a `ForeignKey` for genres.

## 3. Adding Exciting Features

Make the project more exciting by implementing the following features:  
  
- Add a `total\_books` method in the `Book` model to calculate the total number of books in the database.  
- Include a custom field in the `Book` model, such as `rating`, with validation logic (e.g., rating from 1 to 5).

## 4. Migrations

- Use the `makemigrations` command to create migrations for your models.  
- Run `migrate` to apply migrations and set up the SQLite database.

## 5. Creating Superuser and Populating the Database

- Create a superuser to access the Django admin interface.  
- Use the admin interface to add authors, genres, and books.  
- Alternatively, populate the database using the `manage.py shell`.

## 6. Query and Display

Write queries and display data in views:  
  
- Fetch books by a specific author.  
- Retrieve books within a specific genre.  
- Use `annotate` to calculate statistics, such as the total number of books by genre.

## 7. Optional Challenge

- Create a view to display all books in the library along with their authors and genres.  
- Use Django's `ListView` or a custom function-based view for this.

## Conclusion

This guidance outlines the key steps to create an exciting Django project with migrations and an SQLite database. Follow these steps to build a functional and engaging project while learning the core concepts of Django development.