

Exercise: Create a Simple Library Management System

Setting Up the Project

1. Create a new Django project.
2. Create new apps within the project.

Models

1. Define a **Book** model with fields: `title`, `author`, `published_date`, `ISBN`, and `available`.
2. Define a **Member** model with fields: `first_name`, `last_name`, `email`, and `membership_date`.
3. Define a **Loan** model with fields: `book` (foreign key to **Book**), `member` (foreign key to **Member**), `loan_date`, and `return_date`.
4. Make and apply migrations to create the corresponding database tables.

Admin Panel

1. Register the **Book**, **Member**, and **Loan** models with the admin site.
2. Create a superuser to access the admin panel.
3. Use the admin panel to add some books, members, and loans.

Views and Templates

1. Create a view to list all books.
2. Create a view to display details of a single book.
3. Create a view to list all members.
4. Create a view to display details of a single member.
5. Add URL patterns to map to these views.
6. Create templates to display the list of books, details of a single book, list of members, and details of a single member.

Adding and Updating Books

1. Create a form for the **Book** model.
2. Create a view to handle the creation of new books.
3. Create a view to handle the updating of existing books.
4. Add URL patterns to map to these views.
5. Create templates to render the forms for creating and updating books.

Managing Loans

1. Create a form for the **Loan** model.
2. Create a view to handle the creation of new loans.
3. Create a view to handle returning of books.
4. Add URL patterns to map to these views.

5. Create templates to render the forms for creating and returning loans.

Member Management

1. Create a form for the **Member** model.
2. Create a view to handle the creation of new members.
3. Create a view to handle the updating of existing members.
4. Add URL patterns to map to these views.
5. Create templates to render the forms for creating and updating members