ANURAG UNIVERSITY | SCHOOL OF ENGINEERING

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

Project Based Report on

Online Book Store

TEAM 4

23 EG 106 E08
23 EG 106 E15
23 EG 106 E16
23 EG 106 E17

Presentation Schedule :

Programme	**:**	B.Tech (2024-25)
Year/ Semester	**:**	2024-24
Class & Section	**:**	AI –E Section
Course Name	:	Data Base Management Systems
Faculty Name	:	S.Vijitha

Project Title

- Introduction
- Objective
- Documentation
- Coding
- Screen outputs
- Glimpses
- Conclusion

Online Book Store

Overview:

This web application is a Bookstore Management System developed using Flask and SQLite. It allows users to browse books, add them to their cart, and place orders. Admins can manage books, categories, and view orders.

TechStack:

- Backend: Flask (Python)
- Database: SQLite (database.sqlite3)
- Frontend: HTML, CSS (Jinja Templates)
- Authentication: Session-based login system

Database Schema & Relationships

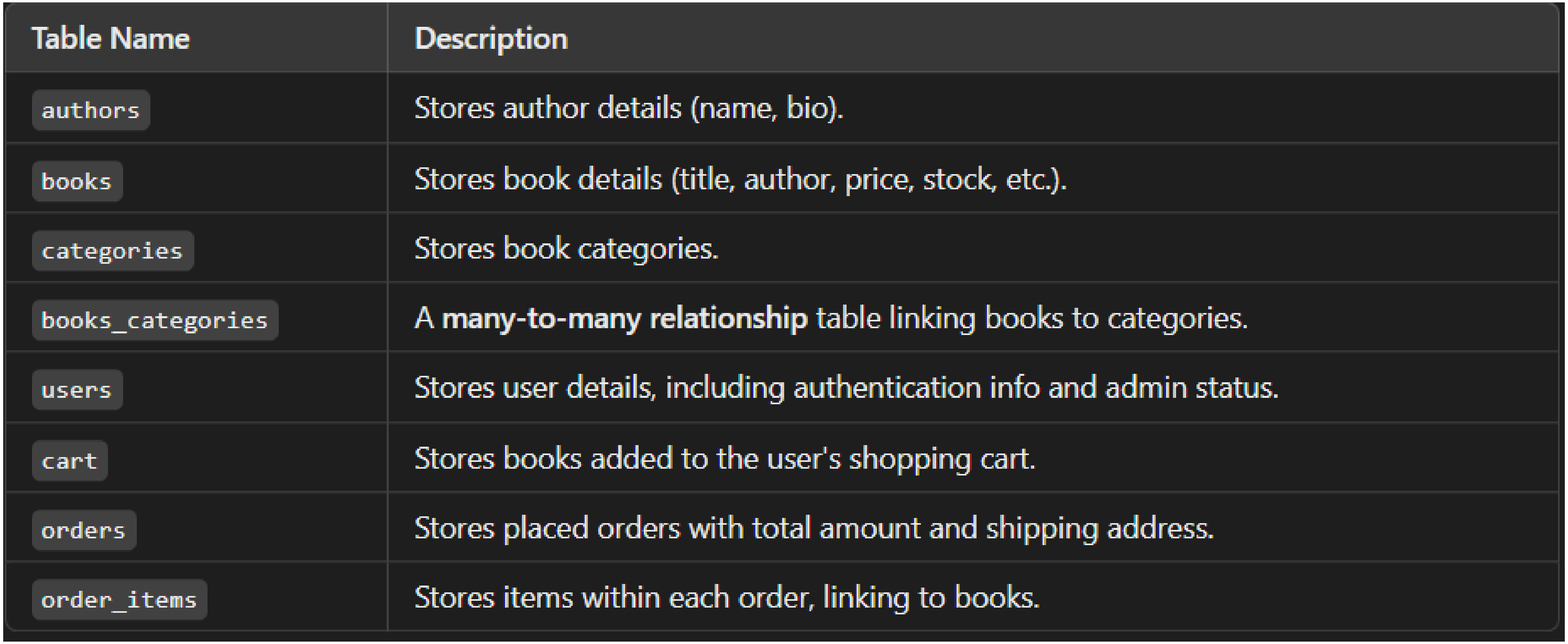

Table Name	Description
`authors`	Stores author details (name, bio).
`books`	Stores book details (title, author, price, stock, etc.).
`categories`	Stores book categories.
`books_categories`	A **many-to-many relationship** table linking books to categories.
`users`	Stores user details, including authentication info and admin status.
`cart`	Stores books added to the user's shopping cart.
`orders`	Stores placed orders with total amount and shipping address.
`order_items`	Stores items within each order, linking to books.

Key Relationships

- One-to-Many:
- books → authors (Each book is written by one author)
- orders → users (Each order belongs to a single user)
- cart → users (Each cart entry is linked to a user)
- order_items → orders (Each item belongs to one order)
- order_items → books (Each order item refers to a book)
- Many-to-Many:
- books ↔ categories (A book can belong to multiple categories, and each category can have multiple books)

Feature Breakdown: Authentication & Order Processing

1. Authentication System (User Login & Session Management)

The application handles user authentication using Flask sessions.
1.How It Works:
User Registration:
Users sign up with username, email, and password.
Passwords are stored unencrypted in SQLite (⚠️ should use hashing like bcrypt for security).
User Login:
The app verifies username and password from the users table.
If credentials match, a Flask session is created.
Users are redirected to their dashboard.
Session Management:
Flask sessions store user authentication details.
Users remain logged in until they log out or the session expires.
Admin Access:
The is_admin column in the users table determines admin rights.
Admins get extra privileges like managing books, authors, and orders.