

BookSnap

Technology Stack:

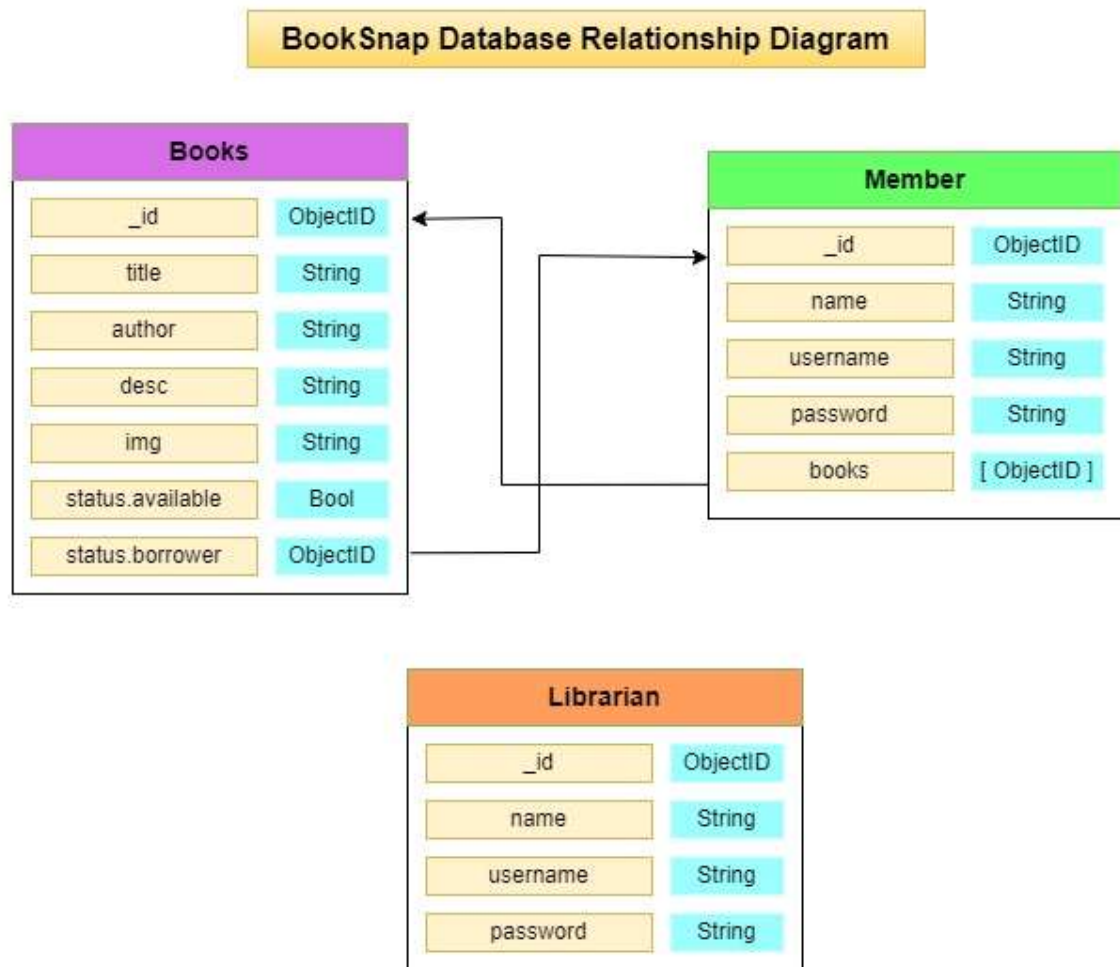
Frontend: HTML, CSS, Javascript, Bootstrap

Backend: Node.js, Express.js, EJS (Templating engine)

Database: MongoDB

ORM: Mongoose

Database Relationship Diagram:



API Details:

1. GET "/": home page
2. GET "/member/login": member login page
3. GET "/member/signup": member signup page
4. POST "/member/login": authenticate user & redirect to app

5. POST `"/member/signup"`: register new member
6. GET `"/member/app"`: app homepage
7. GET `"/member/app/return/:bookId"`: return book from member
8. GET `"/member/library"`: list of books in library
9. GET `"/member/library/borrow/:bookId"`: borrow book to member
10. GET `"/member/logout"`: logout
11. GET `"/member/delete_account"`: delete account
12. GET `"/librarian/login"`: librarian login page
13. GET `"/librarian/signup"`: librarian signup page
14. POST `"/librarian/login"`: authenticate
15. POST `"/librarian/signup"`: register
16. GET `"/librarian/app"`: homepage
17. GET `"/librarian/app/delete_member/:memberId"`: delete member
18. GET `"/librarian/app/retrieve_book/:memberId/:bookId"`: retrieve book
19. GET `"/librarian/library"`: library
20. POST `"/librarian/library/add_book"`: add book
21. POST `"/librarian/library/edit_book/:bookId"`: edit book
22. GET `"/librarian/library/delete_book/:bookId"`: delete book
23. GET `"/librarian/logout"`: logout

Hosting Details:

It requires a `".env"` file to store secret properties such as mongodb uri & jwt secret.

MONGODBURI = MongoDB connection string.

JWT_SECRET = A random string.

JWT_EXPIRY = expiry time in a string ("`8h`" or "`1d`").

I have hosted using [Render](#).

Live URL: <https://booksnap.onrender.com/>