Library Management System

Aman Mani Tiwari | 22F3003055 | 22F3003055@ds.study.iitm.ac.in

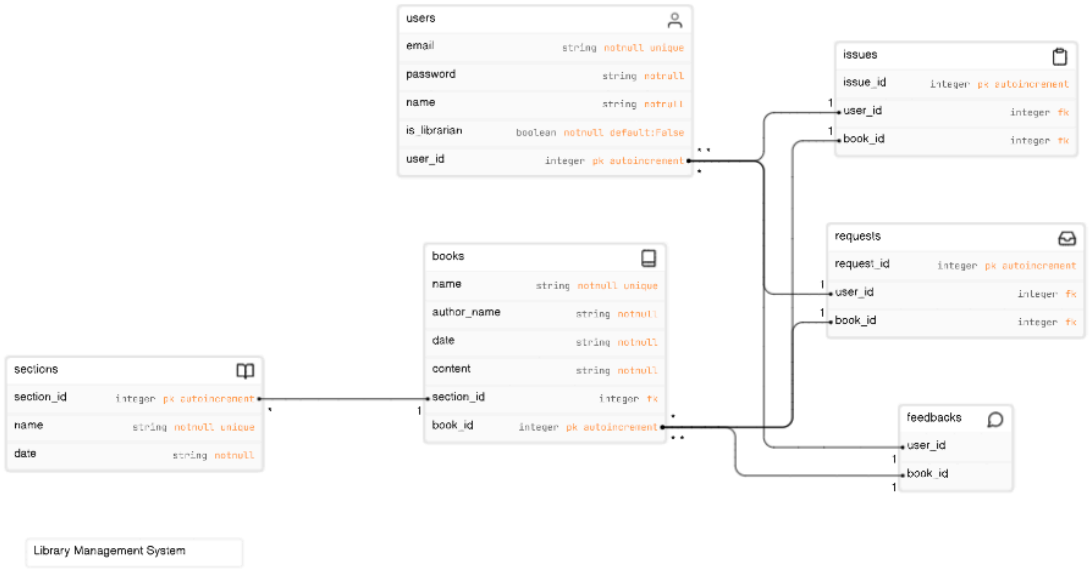
Project Approach:

In response to the project of creating a Library Management System, I began by researching on Google to understand the key features of library systems. After that, I identified key requirements, such as user login, user registration, section management, and book management. Following an agile approach, where the project was divided into smaller tasks for continuous development and feedback, my focus was on creating a simple interface for users and a robust backend for data storage. Regular discussions during project sessions helped me to improve my project further, resulting in a user-friendly Library Management System designed to meet our needs.

Database Model Summary:

- **User:**
  - Each user has an `id` (primary key), `email`, `password`, `name`, and `is_librarian` flag.
  - The `email` field is unique and cannot be null.
  - The `password` and `name` fields are also required and cannot be null.
  - The `is_librarian` field is a boolean indicating whether the user is a librarian or not.
  - The `users` relationship establishes a one-to-many relationship with the `Issue` model, allowing each user to have multiple issues.
  - The `user_request` relationship establishes a one-to-many relationship with the `Request` model, allowing each user to have multiple requests.
- **Section:**
  - Each section has an `id` (primary key), `name`, and `date`.
  - The `name` field is unique and cannot be null.
  - The `date` field indicates the date of the section.
  - The `books` relationship establishes a one-to-many relationship with the `Book` model, allowing each section to have multiple books.

- **Book:**
  - Each book has an `id` (primary key), `name`, `author_name`, `date`, `content`, and `section_id`.
  - The `name` field is unique and cannot be null.
  - The `author_name`, `date`, and `content` fields provide information about the book.
  - The `section_id` field establishes a foreign key relationship with the `Section` model, indicating the section to which the book belongs.
  - The `books` relationship establishes a one-to-many relationship with the `Issue` model, allowing each book to be issued multiple times.
- **Issue:**
  - Each issue has an `id` (primary key), `user_id`, and `book_id`.
  - The `user_id` field establishes a foreign key relationship with the `User` model, indicating the user who issued the book.
  - The `book_id` field establishes a foreign key relationship with the `Book` model, indicating the book that was issued.
- **Request:**
  - Each request has an `id` (primary key), `user_id`, and `book_id`.
  - The `user_id` field establishes a foreign key relationship with the `User` model, indicating the user who made the request.
  - The `book_id` field establishes a foreign key relationship with the `Book` model, indicating the book for which the request was made.
- **Feedback:**
  - Each feedback has an `id` (primary key), `user_id`, `book_id`, and `content`.
  - The `user_id` field establishes a foreign key relationship with the `User` model, indicating the user who provided the feedback.
  - The `book_id` field establishes a foreign key relationship with the `Book` model, indicating the book for which the feedback was provided.
  - The `content` field contains the actual feedback content.



Api Endpoints:

- **GET /api :** *To view all sections in the database.*
- **POST /api/addSection:***To add a section in the database.*
- **GET /api/<int:section_id>:** *To view a section with section_id.*
- **PUT /api/<int:section_id>:***To update a section with section_id.*
- **DELETE /api/<int:section_id>:** *To delete a section with section_id.*
- **GET /api/<int:section_id>/book:***To view all books in the database.*
- **POST /api/<int:section_id>/book:** *To add a book in the database with section_id.*
- **GET /api/book/<int:section_id>/<int:book_id>:***To view a book with section_id and book_id.*
- **PUT /api/book/<int:section_id>/<int:book_id>:***To update a book with section_id and book_id.*
- **DELETE /api/book/<int:section_id>/<int:book_id>:***To delete a book with section_id and book_id.*

Presentation Video Link:[View Presentation]