Problem statement

Hello there, coding wizards! The TU Department of CSE has a swashbuckling new initiative - building software solutions for various departments. The Central Library is on board and they want to create a platform for all you book lovers out there. It's called "The Page Turners" and they need your help to make it a reality!

You, my friend, have been chosen to be the hero of this tale - the elite developer who will build the backend infrastructure and code for this platform. The library wants students to be able to connect with each other, share book recommendations, and request to borrow books. So, get your coding swords ready, and let's set sail!

Here are the details of the adventure that awaits you (requirements):

1. First and foremost, a student must be able to register or create an account in "The Page Turners". They'll need to provide their name, roll number, course name, department, and year of admission, and email ID when signing up. They can also upload a profile picture if they fancy!
2. The library administration needs to be able to update the book database, so you'll need to make sure that's possible. Each book will have its title, author, genre, year of publication, and ISBN stored in the database.
3. Students can connect with each other by sending a connection request. Don't be shy, make some new friends!
4. Students can search for other students based on name, course, and department. We'll even add an auto-complete feature so they can find their buddies quickly and easily.
5. Speaking of auto-complete, we want to add the same feature for searching books too. If a student types "harry" we want to suggest "Harry Potter and the Sorcerer's Stone", "Harry Potter and the Chamber of Secrets" and all of the other wizarding adventures.
6. If a student wants to borrow a book, they can search for it on the platform and send a request to borrow it. We'll need to keep track of which books have been borrowed and by whom.
7. The platform must have the functionality of recommending books to students based on their interests. Who knows, maybe they'll find their new favourite book this way!
8. We also want to recommend other students that a user may want to connect with. Just like Facebook's "people you may know" feature.

Your mission, should you choose to accept it, is to build a collection of REST APIs using Python and MySQL/Postgres as your backend database. You'll need to create several API endpoints to allow for student registration, book updates, auto-complete searches, connection requests, book borrowing requests, book recommendations, and student recommendations.

Your code must have at least the following API endpoints -

1. API to enable student registration.
2. API to update/delete/add books to the database. This API must be able to handle bulk requests. This means that a user must be able to update/delete/add details of multiple books in a single request.
3. API to provide auto complete feature for student searching.
4. API to provide auto complete feature for book searching.
5. API to enable a user to send a connection request to other users.
6. API to enable a user to send a request to issue a book. This is for requirement 6
7. API to recommend books to users
8. API to recommend users with whom a user may want to connect.

You are free to add any other API endpoint that you feel is necessary or can enhance the functionality of the platform.

As a wise man once said, with great power comes great responsibility, and the same applies to coding. So, make sure your code is well-structured, flexible, and easy to understand. Add comments and documentation where necessary, and use proper OOP and design pattern concepts.

If you want to impress us even more, you can try using Docker to create a container of your project. And if you're feeling extra adventurous, think of ways to use machine learning to enhance the platform!

We've attached some sample data in CSV files to get you started on your journey. So, hoist the Jolly Roger, grab your parrot, and let's set sail towards a world of books and connections!

Submission details:

- Add a requirements.txt file with all the required packages with their version number
- Add a readme file with instructions on how to setup and run the project in a local environment

- Due to the presence of a database in this project you can choose to handle it in two ways:
    - Host the entire project in heroku or any other hosting platform of your choice, this is more convenient for examination
    - Make a database dump file and add proper instructions in the readme file on how to setup the database in a local machine