Microservice Backend Intern Assignment

This application contains three services,

Content Service

- Service description : Contains content(simple text based) and content meta of readable content that we serve to the users
- Each content is structured as a series which can contain multiple chapters.
    - Example: consider a series called "Harry Potter" which has 7 chapters
- Apis required :
    - Fetch all content with content meta-data like episodes, title etc
    - Fetch content for a user
        - Input : userid
        - Output :
            - Content meta with only unlocked chapters per series
            - Count of total episodes
            - Count of unlocked episodes
    - Api for bulk upload of the content

User & Daily Pass Service *(For the sake of simplicity, user and daily pass can be incorporated in one single service)*

- Service description :
    - Creates and fetches user details like name, email etc (auth not required, just need userId here)
    - Contains the details of how many chapters per series is unlocked for a particular user
- Daily Pass Logic :
    - When a user installs the Pratilipi Application (i.e. day1 of user creation) , 4 chapters are unlocked by default on the day of installation
    - Rest of the chapters are released on a daily basis; that is one chapter per day
    - Any new content that is uploaded, again by default has 4 unlocked chapters for all the users
- One should consider the scenario that there may be newly uploaded content. Not all the content will be fed to the system on a single day
    - So if a series is uploaded after user creation then the existing users should also see that series with only 4 chapters unlocked
- Apis required for users
    - Create user
    - Fetch all users
- Apis required for daily-pass
    - Api to unlock one chapters for the given user and series
        - For testing purpose : this api should not be idempotent, if I hit the api twice, it should unlock two episodes for the requested user and series
        - Number of unlocked chapters = 4 + no. of times unlock api is hit

---

Note :

1. Please share the github link or zip file of the assignment
2. Language : we prefer Python/Nodejs/Golang but are fine with any other language as well
3. Rest of the stack can be of your choosing as long you understand the technology
4. HLD/LLD/Data Model/Architecture diagram, anything that helps us understand your implementation
5. Api documentation - postman collection, swagger or any doc that helps us to test your apis
6. Any script(if used), should be part of the services
7. Host the services, db and other dependencies OR nicely tie them together using docker-compose.
    a. If dockerised - This should be accompanied by steps to run locally
    b. If hosted - Flush all test data
8. The paradigms of microservice should be followed. Each service should own the logic and data it is defined to own. Databases should be separate for all 3 microservices. [Same database instance can be used for the purpose of this interview, but database itself should be entirely separate for each microservice].
9. No Code repetitions, common pieces of code should not be replicated anywhere.

10. You can use any technology for inter-service communication as long as you can justify it
11. Follow clean code practices