Microservice Assignment(Backend Intern)

3 micro services - content, user-interaction service, user service.

Content Service

Serving books as content. A content will have a story and title, considering the scope of this assignment.(Example?)

Data ingestion should happen via csv, write a script to ingest data into the database(Ideally script should also be a part of your service).

Content service should have atleast the title, story, date published and the user id stored.

REST API's for CRUD operations on contents.[**title, story**]

New contents API - sorted on date
Top contents API - sorted on user-interaction[Sort on basis of Number of interactions, both read and like]

Testing*- An API to help us post the csv file, and it should automatically invoke the data ingestion process once it receives the csv file.*

User interaction service

Add Update and Read API's for following 2 interactions.

1. User Read event(validate if user exists)
2. User Like event(validate if user exists)

User Interaction service is basically supposed record events done by the user. In this case the service needs to be able to record 2 types of events, Like and Read.
Like - Think about what happens when you hit the "like" button on facebook, an event is triggered, this is event is synonymous to the same, albeit the content that is liked should have been served by the content service itself.
Read - The content service is serving books, so read event is basically triggered once the user has completed reading the book. UI will take care of when this has to be triggered, what you need to take care of is that the read event gets recorded as and when it is triggered.

Additionally if you take a look at content service, you will be able to see that one of the sorting will need to be done on the basis of user interactions, you will have to expose internal API for content service to do the same, since interaction data is being recorded in user interaction service.

User service

REST API's for CRUD operations on user.[First name, last name, email_id, phone number]

Concerns

1. Please dockerize everything(yes, including databases).
2. Microservice architecture has to be followed strictly, 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].
3. No Code repetitions, common pieces of code should not be replicated anywhere.
4. Database schema/design
5. HLD and LLD
6. REST API conventions(We are open to graphQL as well).
7. Documentation, we would prefer something like swagger, but open to anything you come up with as long as it helps us understand your code better.
8. Consider using an Architecture diagram.
9. Worker - Should always be part of the service. In general workers can be CRON as well as queue consumers.

Code base

1. We prefer node.js/python/golang, but we are open to other languages as well.
2. Database - Again upto you, but do remember that we might need to run your application on local.
3. Ideally Code should be **dockerized** and neatly tied in using docker-compose[network mode host]. In case you are finding it difficult to use **docker-compose**, just dockerize all applications/db and send us commands to build/run your code. If possible put commands in a script, again not a deal breaker though.
4. We expect Postman collection or sample commands for all API's.
5. Again bonus points for considering developer empathy. https://apiguide.readthedocs.io/en/latest/principles/empathy.html#:~:text=Design%20with%20developer%20empathy&text=Perhaps%20the%20most%20important%20criteria,will%20remain%20undiscovered%20or%20unrealised
6. Please share the github link or zip file for the assignment via email.