Backend Task

Task:

Build a simple stateless microservice with the following API endpoints:

- Authentication endpoint
- REST API endpoints for a journal app (details below)

Note:

- We would prefer a solution written in **Node.js.**
- There are no restrictions on the libraries to use. You can choose whichever seems best suited for the task.
- You can use any RDBMS to store data. SQL files to create required tables and an ER diagram should be present in the project.
- Attach API documentation with a request which can be executed directly. You can use Postman Collection, Swagger, etc for documentation.
- Deploy your code and send us the server URL.

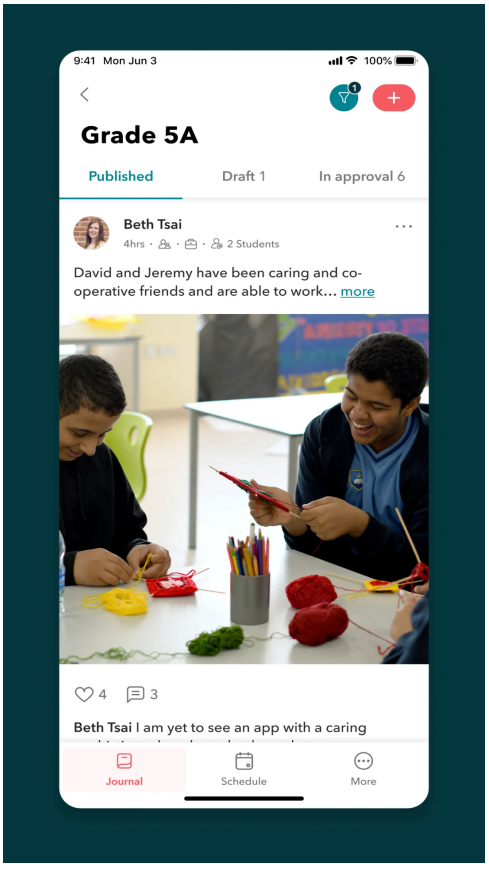
API Details:

1. **Authentication endpoint:**
   - This will be a public endpoint.
   - The request body will contain an arbitrary username/password pair. Note: Treat it as a mock authentication service and accept any username/password. **Password comparison should be present**.
   - Return a signed JSON Web Token (JWT, https://jwt.io/) which can be used for validation in future requests.

2. **Journal REST API endpoints:**

   The following endpoints should be protected. The JWT obtained in the "Authentication" endpoint will be attached to each request. If the JWT is missing or invalid, these endpoints should reject the request.

A **journal** refers to a written and visual account of what a child has done in a classroom. (A journal feed is similar to an Instagram feed)



The journal will have the following features:

- There are two types of users in the system
  - Teacher
  - Student
- Journals are the written and visual descriptions of a student activity done in the classroom
- Journal can only be created, updated, and deleted by the Teacher.

- The journal consists of a description, a list of students tagged, and when it was published.
- The journal can also contain an attachment, which can be of the below-mentioned types:
    - Image
    - Video
    - URL
    - PDF
- A journal feed is generated for students and teachers based on the below filters:-
    - Teacher feed - The teacher can see all the journals that the teacher has created themself.
    - Student feed - The student can see all the journals in which they are tagged.
- The Journal *published_at* is a date-time field at which the journal needs to be published, if the journal is scheduled for the future then it should not appear in the student's feed.

Create protected REST endpoints for the following:

- Create/ Update/ Delete a journal as a teacher
- Publish a journal as a teacher.
- Journal feed as a teacher and student.
- Journal feed
    - For the teacher, the feed will return all the journals created by the Teacher
    - For students, the feed will return all the journals the student has been tagged.

Bonus: Extra points for attempting these.

1. Design a system to send notifications to students when they are tagged in a journal

   Briefly describe the architecture, technologies to use, and working details.
2. Implement the APIs in GraphQL

   Implement all the above APIs as a GraphQL endpoint

Things we are interested in:

- Completeness of the APIs - authentication & journal
- Knowledge of REST APIs, SQL, JWT & using libraries
- Your database design
- Modularity & readability of code
- Attention to detail and quality

How to submit:

Please email your submission to the following address

To: internship@toddleapp.com

Subject: <Your name> - <College shortname> Backend task

Along with:

1. Google drive link of your source code zip
   - Be careful of the sharing permission.
   - While creating zip, please don't include the `node_modules` folder.
2. Hosted URL [Netlify, Heroku, etc.]

For any clarifications contact: ashish@toddleapp.com

Important:

We condemn plagiarism. Please maintain the dignity and originality of your work. If we suspect any attempt towards copying, we will disqualify the submission. Also, **do not upload your code to any VCS such as GitHub, GitLab etc.**