Backend Task

Task:

Build the backend of a social media platform.

It should have all the usual functions of a basic social media platform

- Create a user profile
- Login to the created profile
- Find & follow people that you know
- View content posted by people you follow
- Like & comment on content posted by people you follow
- Create content
- Be able to view likes & comments on your content

Task details:

- **User**
    - Should have at least some form of unique identifier
    - Should have at least some way to authenticate himself/herself
    - Should be searchable by name/unique identifier
    - A user can follow another user
- **Post/Content**
    - A post or content is created by a user to be made available in the feed of everyone who follows him/her
    - A post can have media and/or text
    - A post can be liked by other users
- **Content feed**
    - It is a list of all the posts that should be visible to a user
    - It should be unique to each user as per the users they follow
    - It should also contain posts that the user has himself/herself created
    - It should also reflect the number of likes and comments that each post has
- **Like**
    - Users can like posts or content created by other users
    - Each user can only like a post once
    - Each user can remove his/her likes from the post

- **Comment**
    - Users can comment on posts or content posted by themselves or by other users
    - Users can add/ edit/ remove their comments
    - Users cannot add comments to posts/ content in which the creator has disabled comments

API endpoint details:

1. **Authentication**
    a. The user needs to authenticate by using some form of authentication that you define in your schema
    b. Return a JSON web token for validation of further requests
2. **Find people** (GET)
    a. An endpoint to be able to find people by their name
3. **Follow people** (POST)
    a. An endpoint to be able to follow people
4. **Content feed** (GET)
    a. An endpoint to get posts
    b. Only posts by people you follow should be brought over
5. **Content details** (GET)
    a. An endpoint to get details of a single post
6. **Comment** (GET/POST)
    a. Endpoints to create/edit/delete comments against posts
7. **Like** (GET/POST)
    a. Endpoints to create/remove like against posts
8. **Create content** (POST)
    a. Endpoints to create/remove content

Bonus: Extra points for attempting these.

1. Scheduled action
    a. If you can provide a user with the ability to schedule content posting.
2. Deploy your code and send us the server URL
3. Implement the APIs in GraphQL
    a. Implement all the above APIs as a GraphQL endpoint

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 store data using any RDBMS. The project should include SQL files to create the required tables and an ER diagram.
- Attach API documentation with a request that can be executed directly. You can use Postman Collection, Swagger, etc, for documentation.

Things we are interested in:

- Completeness of the APIs - authentication & assignment
- Knowledge of REST APIs, SQL, and JWT
- 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.
2. While creating zip, please don't include the node_modules folder.
3. Hosted URL [Netlify, Heroku, etc.]

For any clarifications, contact: miftah@toddleapp.com

Important:

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