



# Backend Task

## Task Overview

You will be working on the backend of a **basic social media platform**.

The foundational codebase has already been implemented, including **authentication** and **content creation** features. Your task is to **debug existing issues, complete unfinished functionalities, and enhance the API behavior** as outlined in the task.

- 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

You do not need to build this from scratch—a base codebase is shared with you to work on. Use of AI tools is permitted.

### Codebase :

[https://drive.google.com/file/d/1rOiNC0JaKlFVGZnAoyPfXdYUTvAxZkwx/view?usp=drive\\_link](https://drive.google.com/file/d/1rOiNC0JaKlFVGZnAoyPfXdYUTvAxZkwx/view?usp=drive_link)

## 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/unfollow people and fetch follower stats.

### 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. CRUD endpoints for comments against posts

### 7. Like (GET/POST)

- a. CRUD endpoints for like against posts

### 8. Create content (POST)

- a. Create/Delete Endpoints for 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:**

- There are no restrictions on the libraries to use. You can choose whichever seems best suited for the task.
- The project should include an updated schema.sql file to create the required tables and an ER diagram.
- Attach API documentation with a request that can be executed directly.

**Things we are interested in:**

- Completeness of the APIs
- 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](mailto:internship@toddleapp.com)

**Subject:** <Your Name> - <College shortname> - Backend Task

Include the following (all items mandatory – **submissions missing any of these will be rejected**):

1. Google Drive link to your source code ZIP (excluding node\_modules). Be careful of the sharing permission.
2. Hosted URL of your API (Netlify, Heroku, Render etc.)
3. Loom video walkthrough (~3-5 minutes) covering:
  - What bugs you identified and fixed
  - What features you completed or extended
  - How you have used AI for development.
  - A brief overview of your code and file structure (Optional)
4. Updated schema.sql file and ER diagram
5. API documentation (e.g., Postman collection or simple markdown file with request/response samples)

For any clarifications, contact: [soumendu@toddleapp.com](mailto:soumendu@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.