**[Repository](https://github.com/SRIDHARAN1819/social-media-platform)**

**[Live Link](https://social-media-platform-l4xu.onrender.com)**

**Social Media Platform Development**

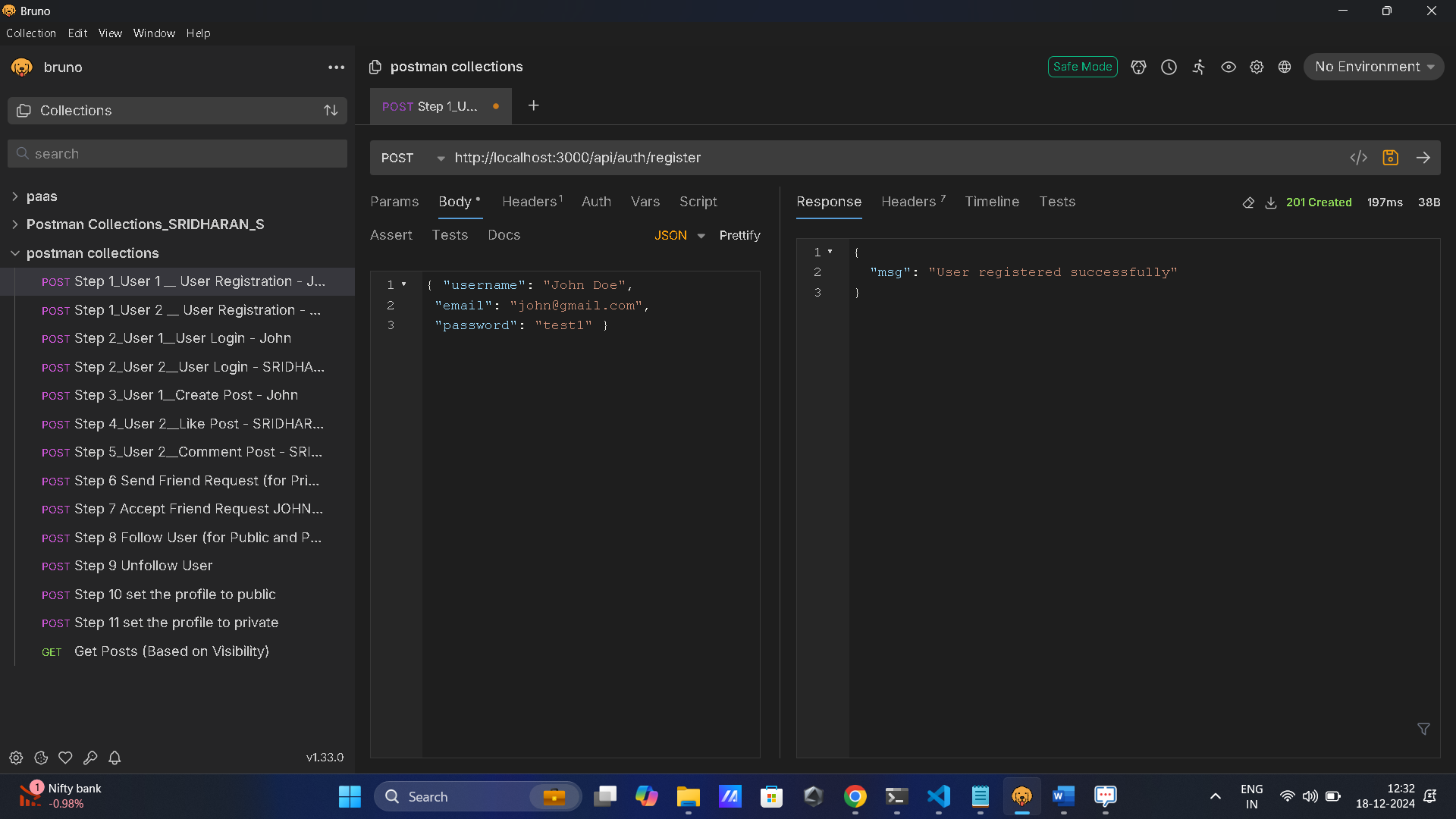
**Project Overview**

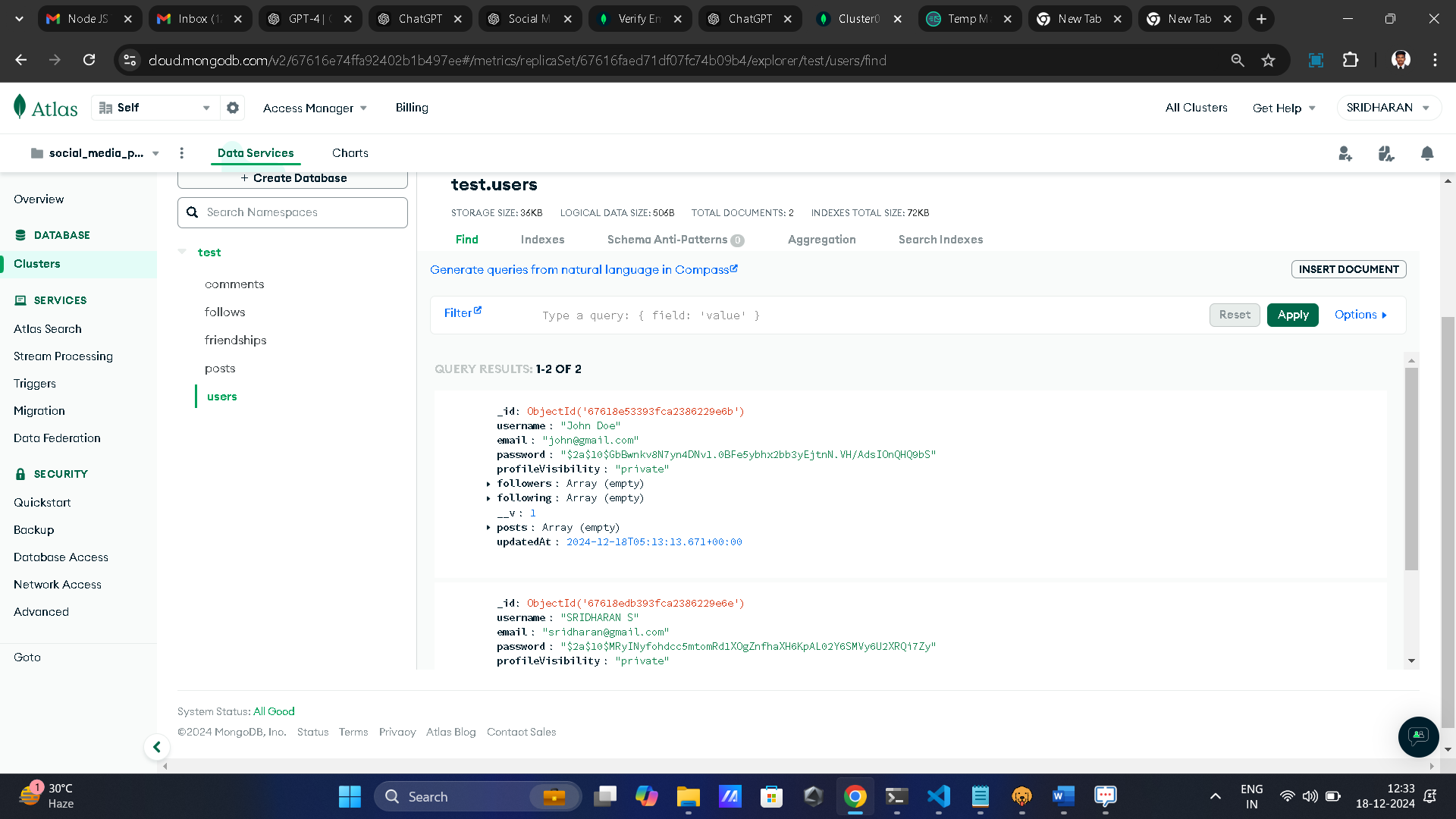
The purpose of this project is to create a social media platform inspired by Facebook’s functionality. The application will allow users to register, interact, and manage their relationships and posts. It is built using **Node.js**, **Express**, and **MongoDB** for database management.

**Features**

**1. User Registration**

* Users can register and create profiles with basic details.
* Required fields: Username, Email, Password.
* Passwords are securely hashed using **bcrypt**.
* Example:



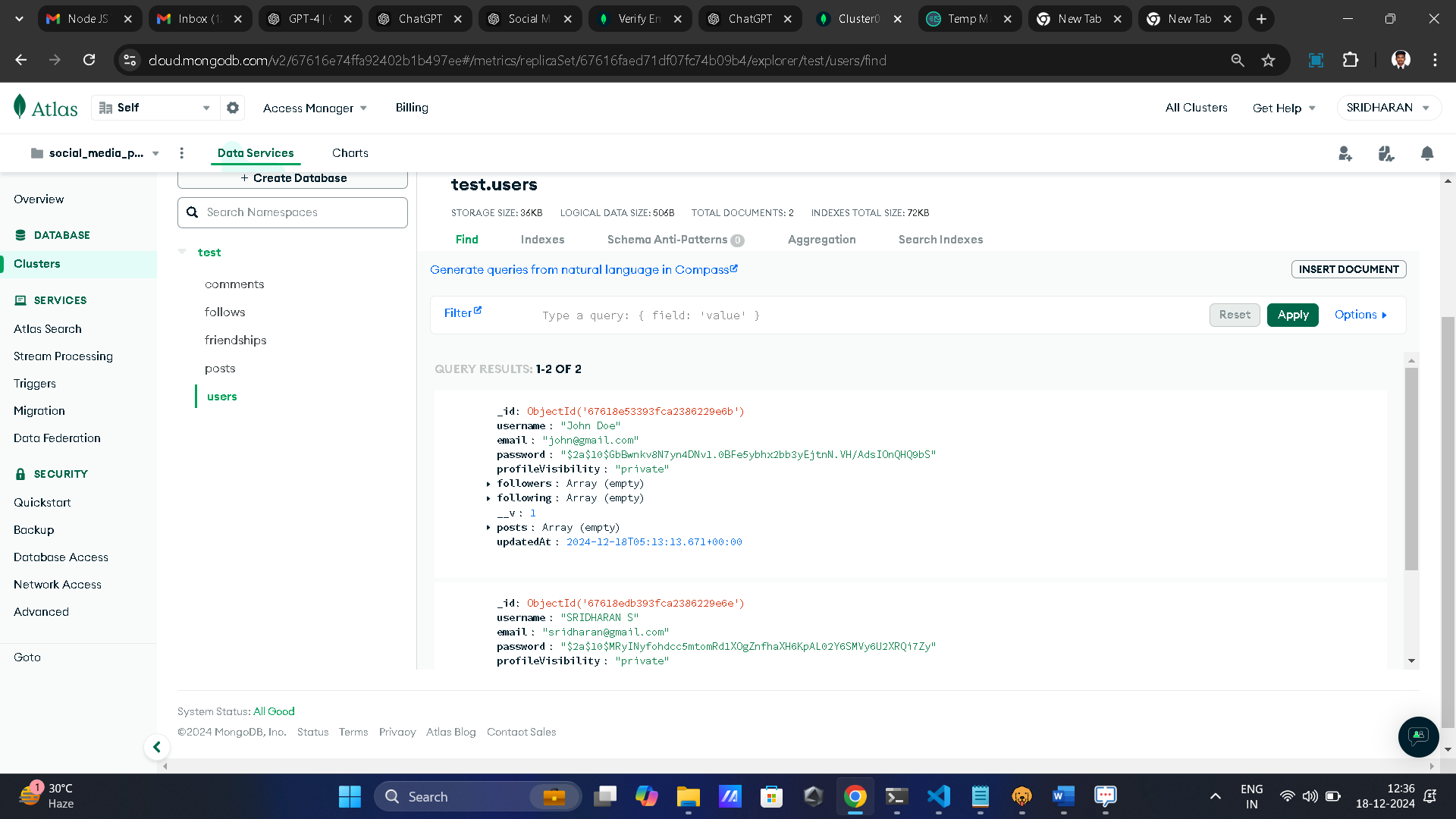


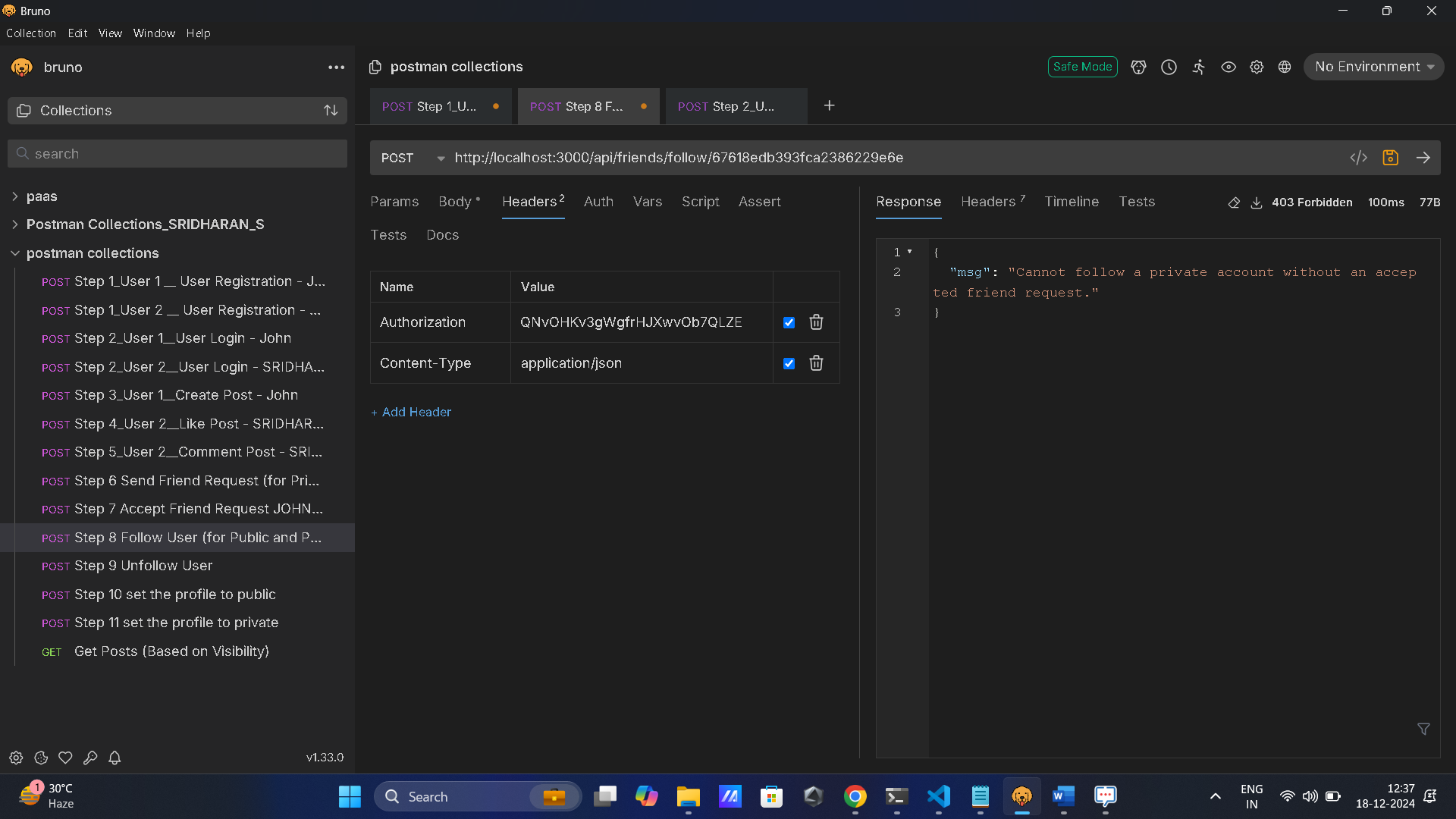
Login



**2. Follow System**

* Users can follow or unfollow other registered users.
* A followers and following mechanism is maintained in the database.
* Follow
* Example: Private Account Means
* **profileVisibility : "private"**
* Error: "msg": "Cannot follow a private account without an accepted friend request."
* **Screenshot**:

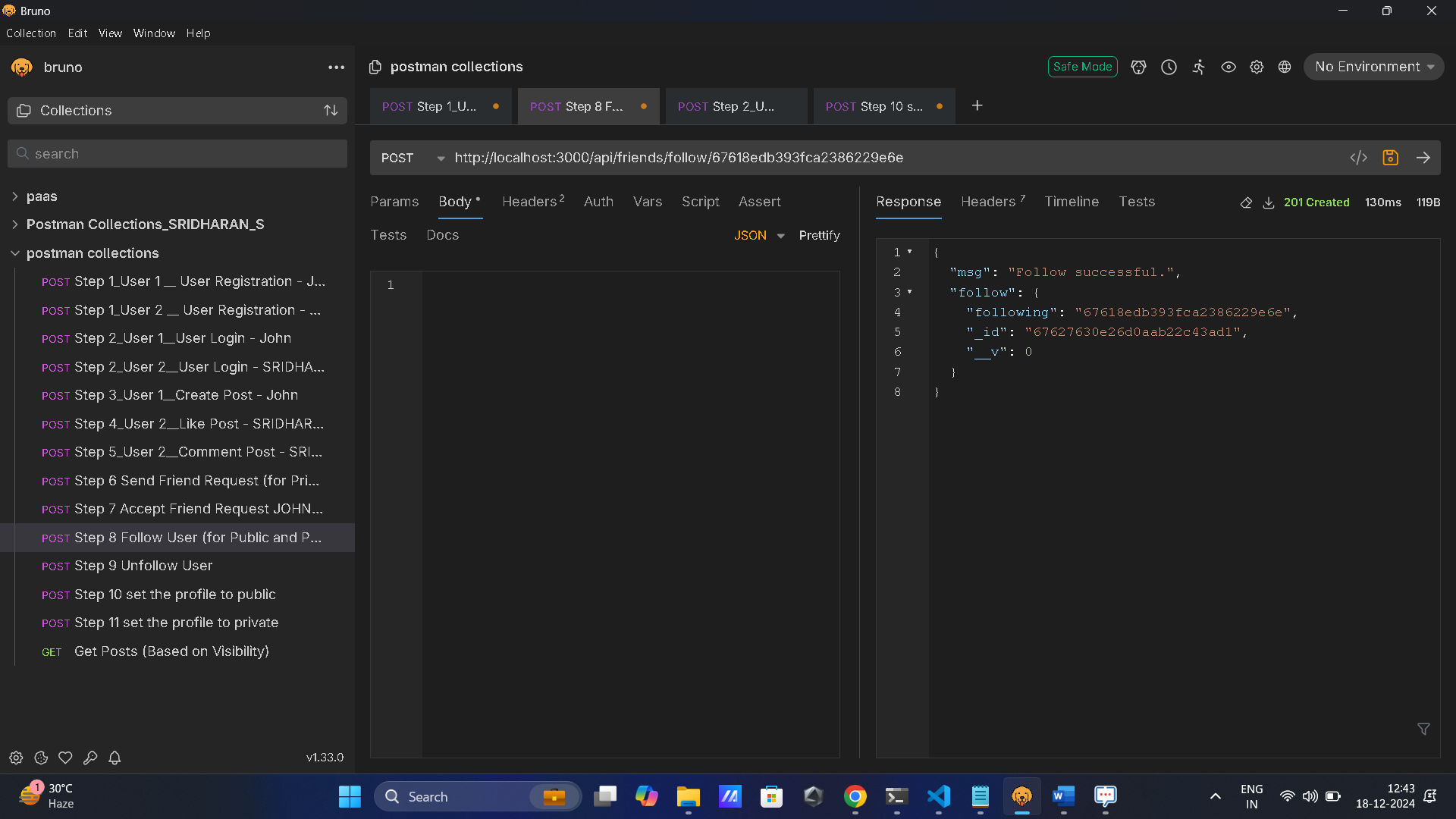




* Example: Public Account Means
* Check Users table  **profileVisibility** :  "public"

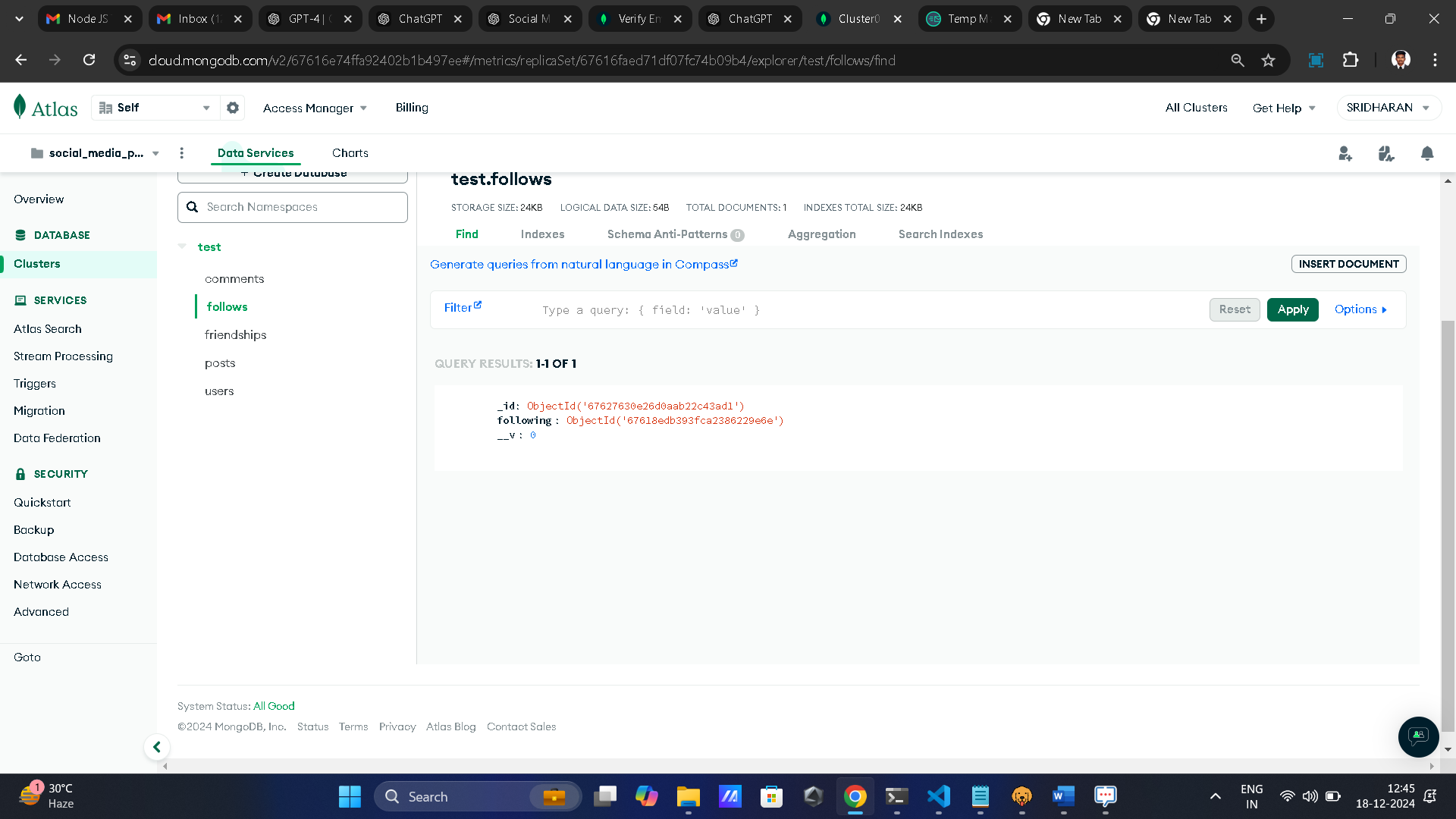
**Screenshot**:



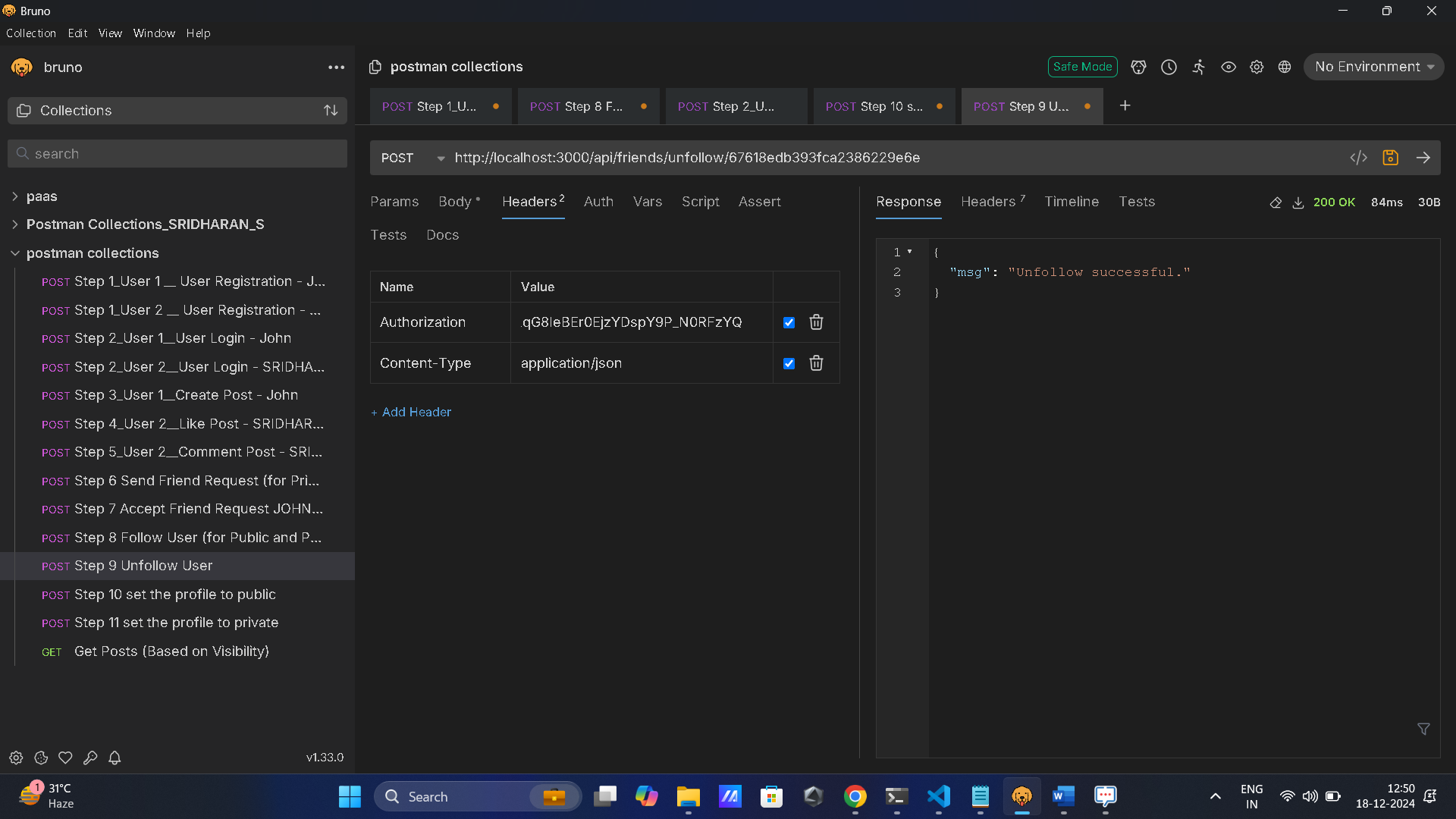


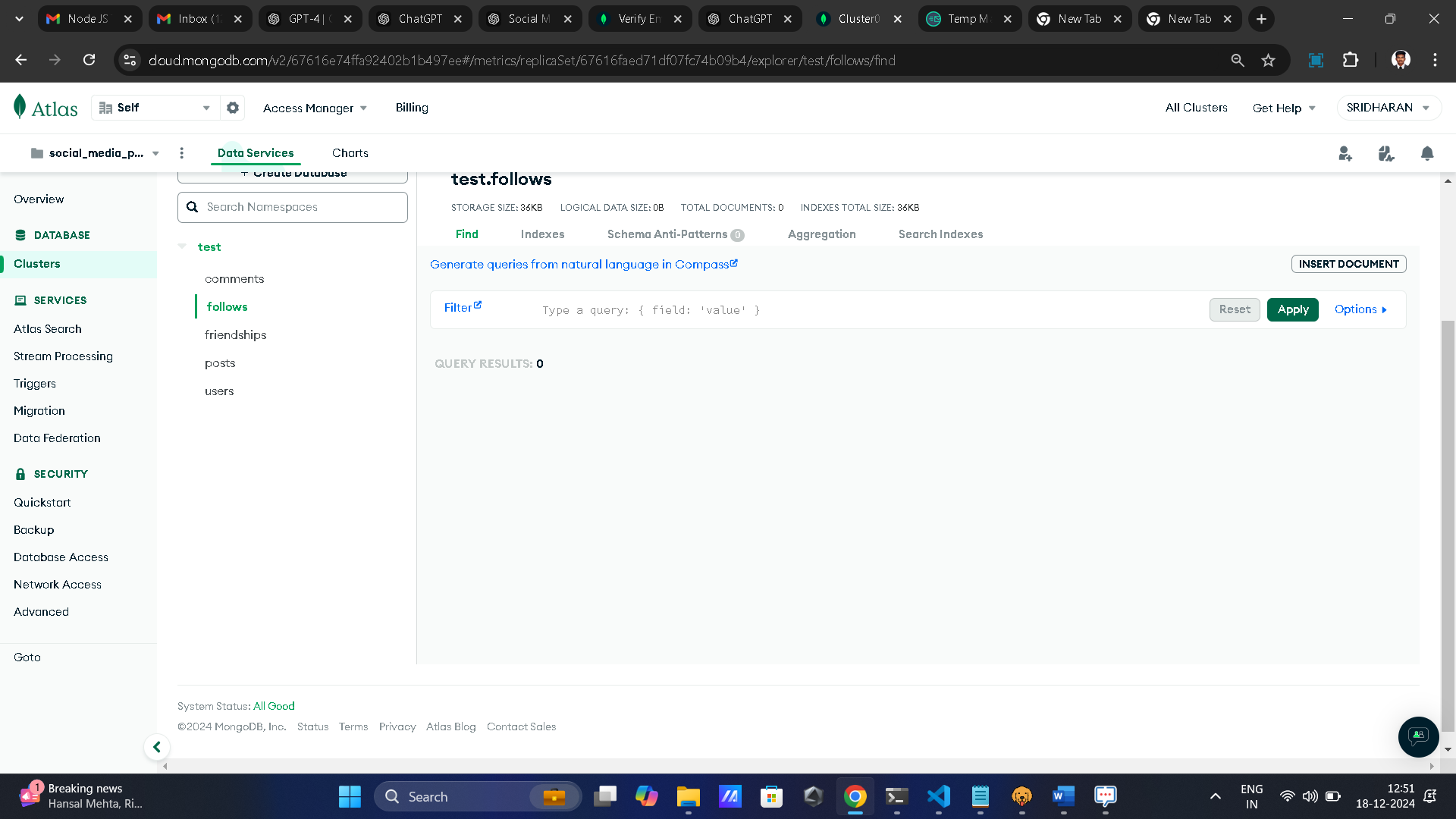
Following means which you follow that user id

DB:



* Unfollow
* Example: **Screenshot**





**3. Posting & Interactions**

* Users can:
  + Create posts with images (handled via **Multer** for file uploads).
  + Like or comment on posts.
* Posts contain fields for:
  + userId: Creator of the post.
  + content: The text of the post.
  + image: Uploaded image.
  + likes: List of users who liked the post.
  + comments: Array of comments.
* Example:

**Screenshot**:

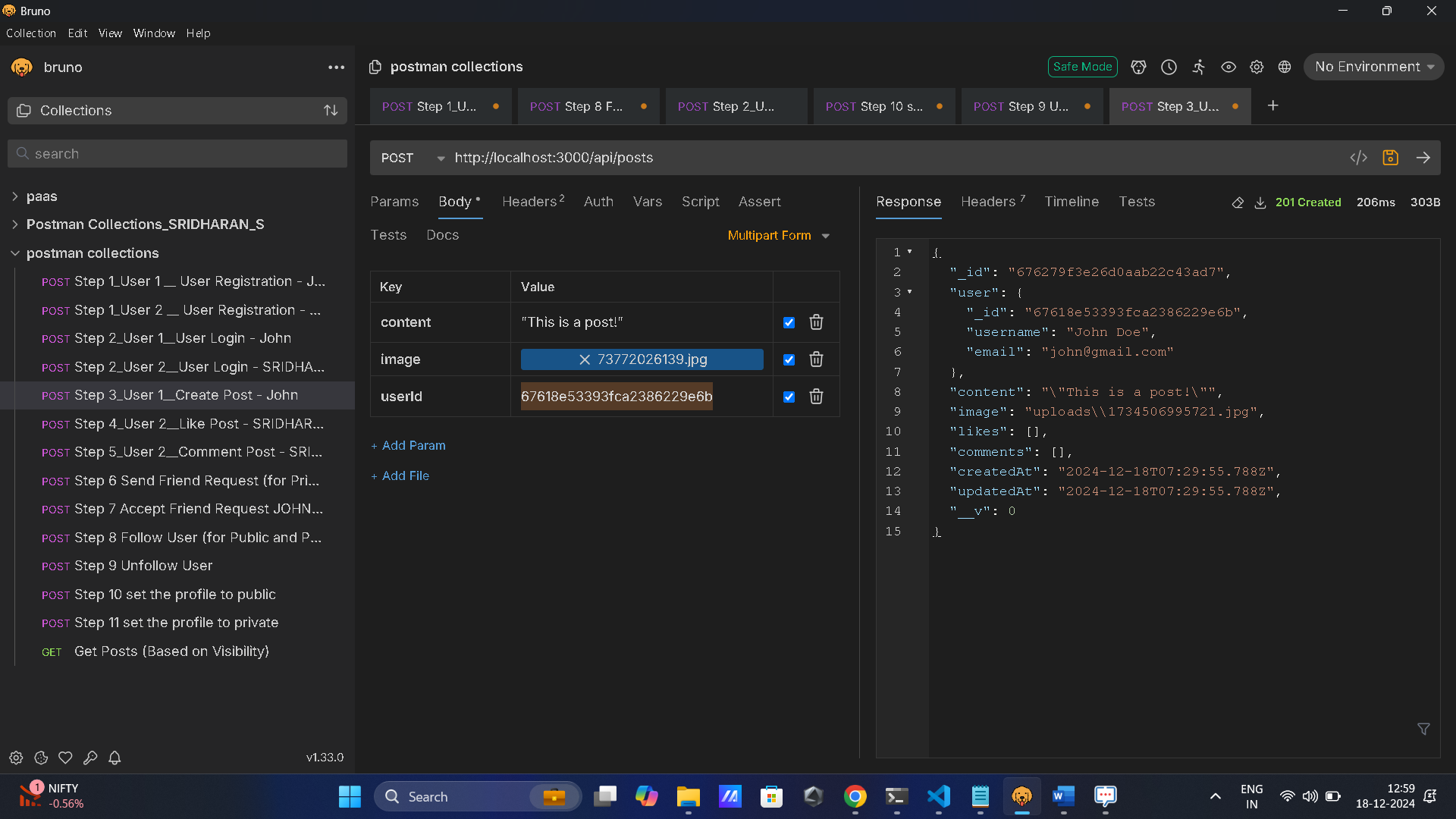
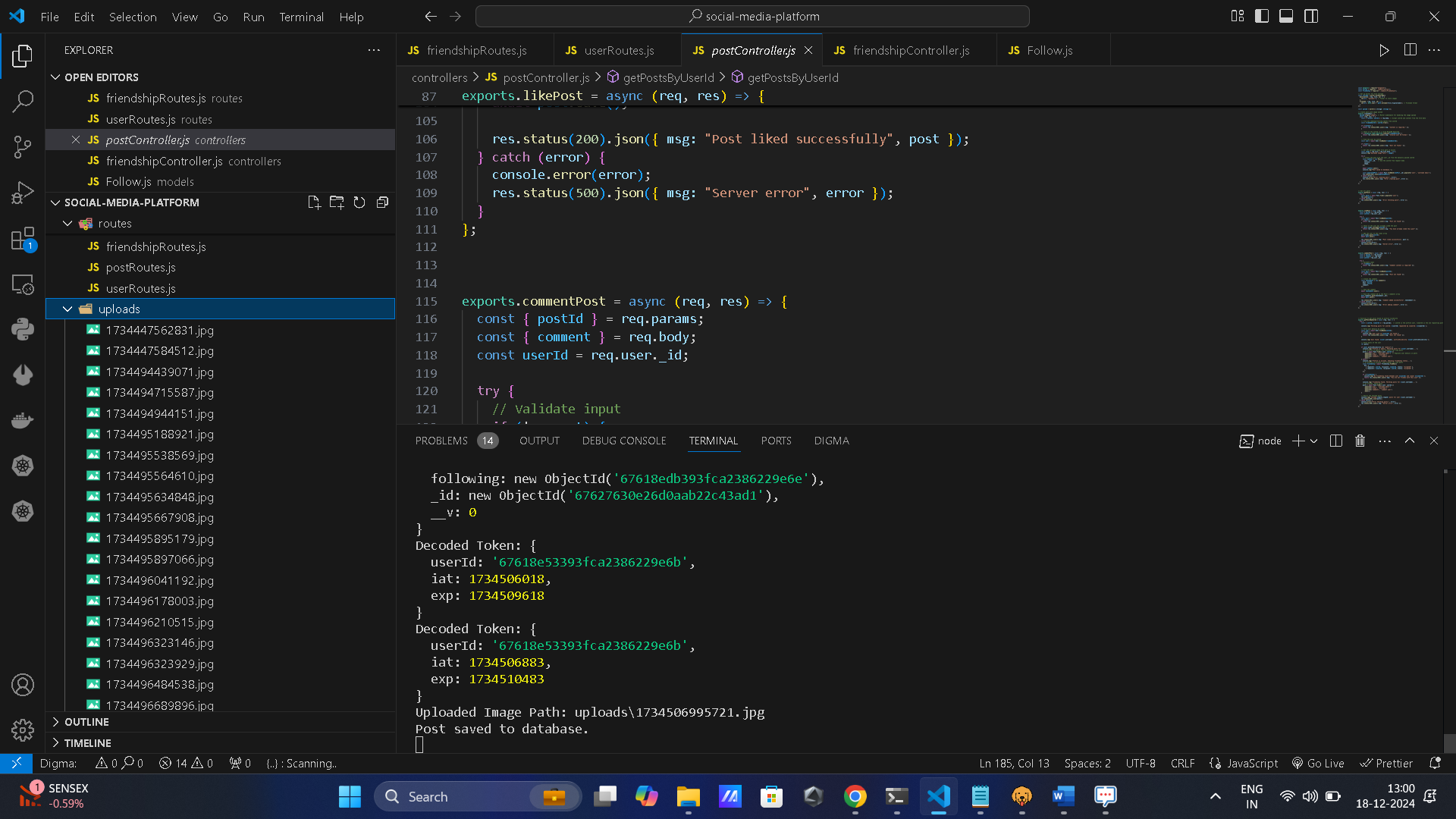
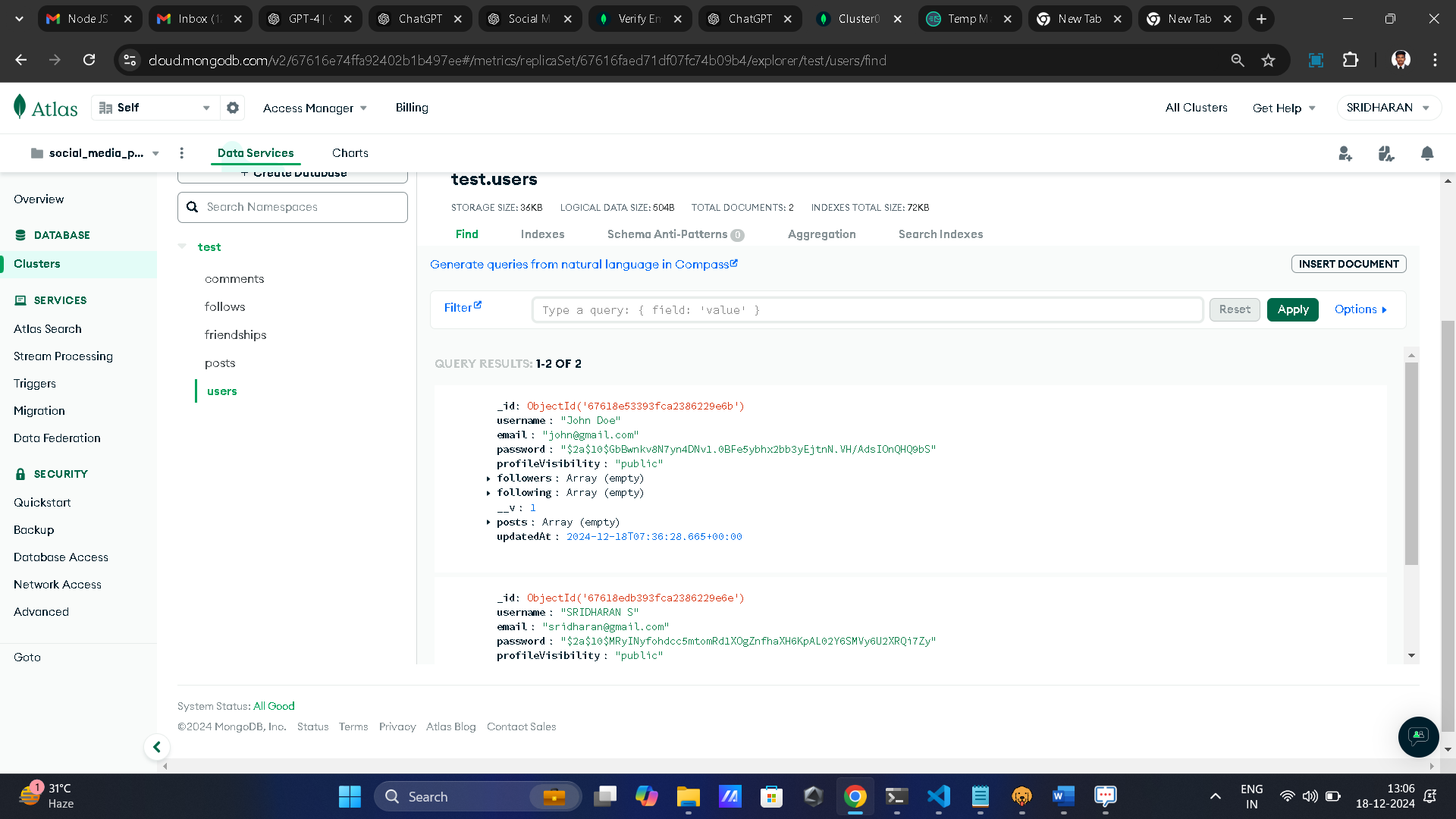
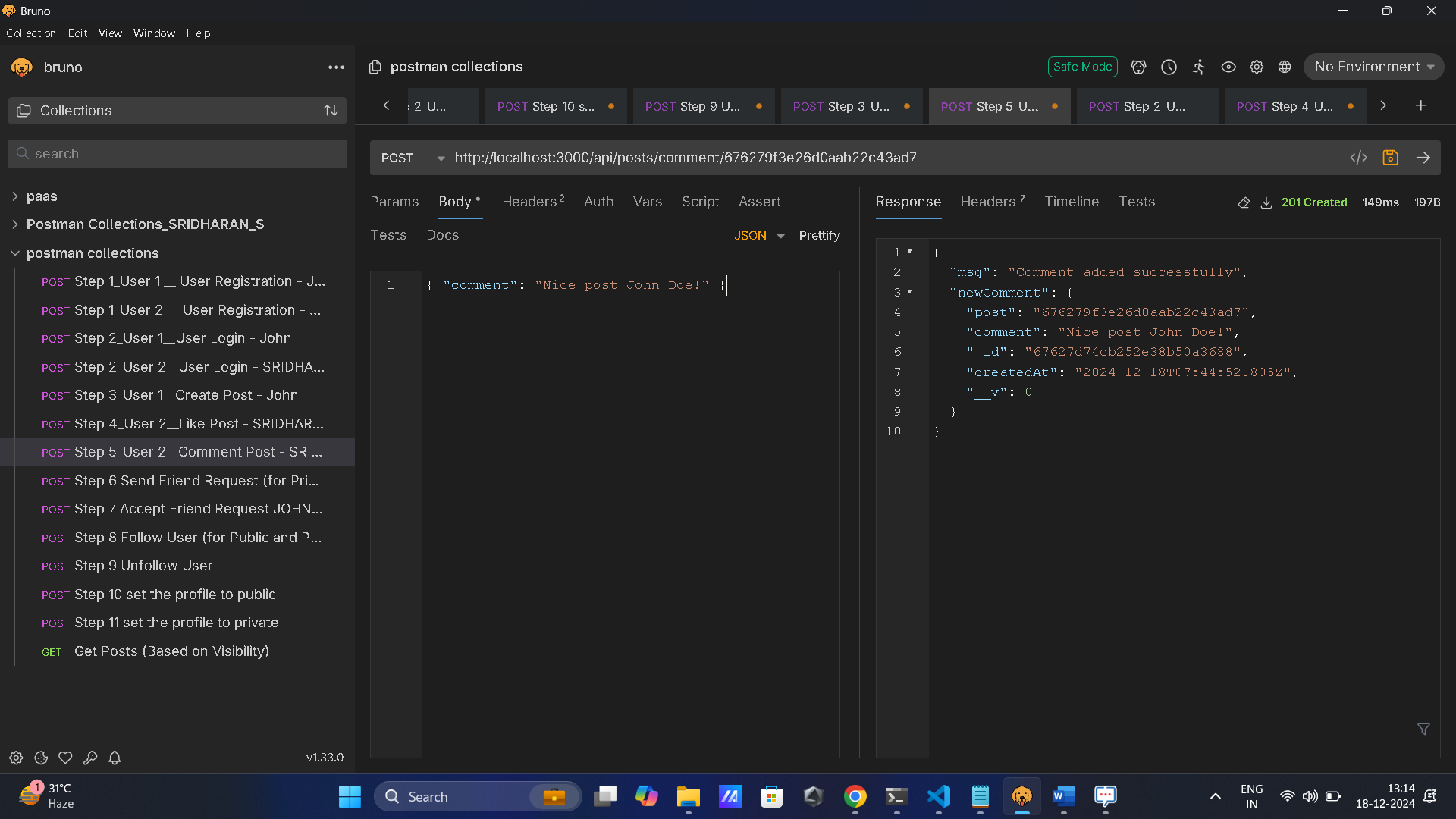


Image upload - Uploads folder

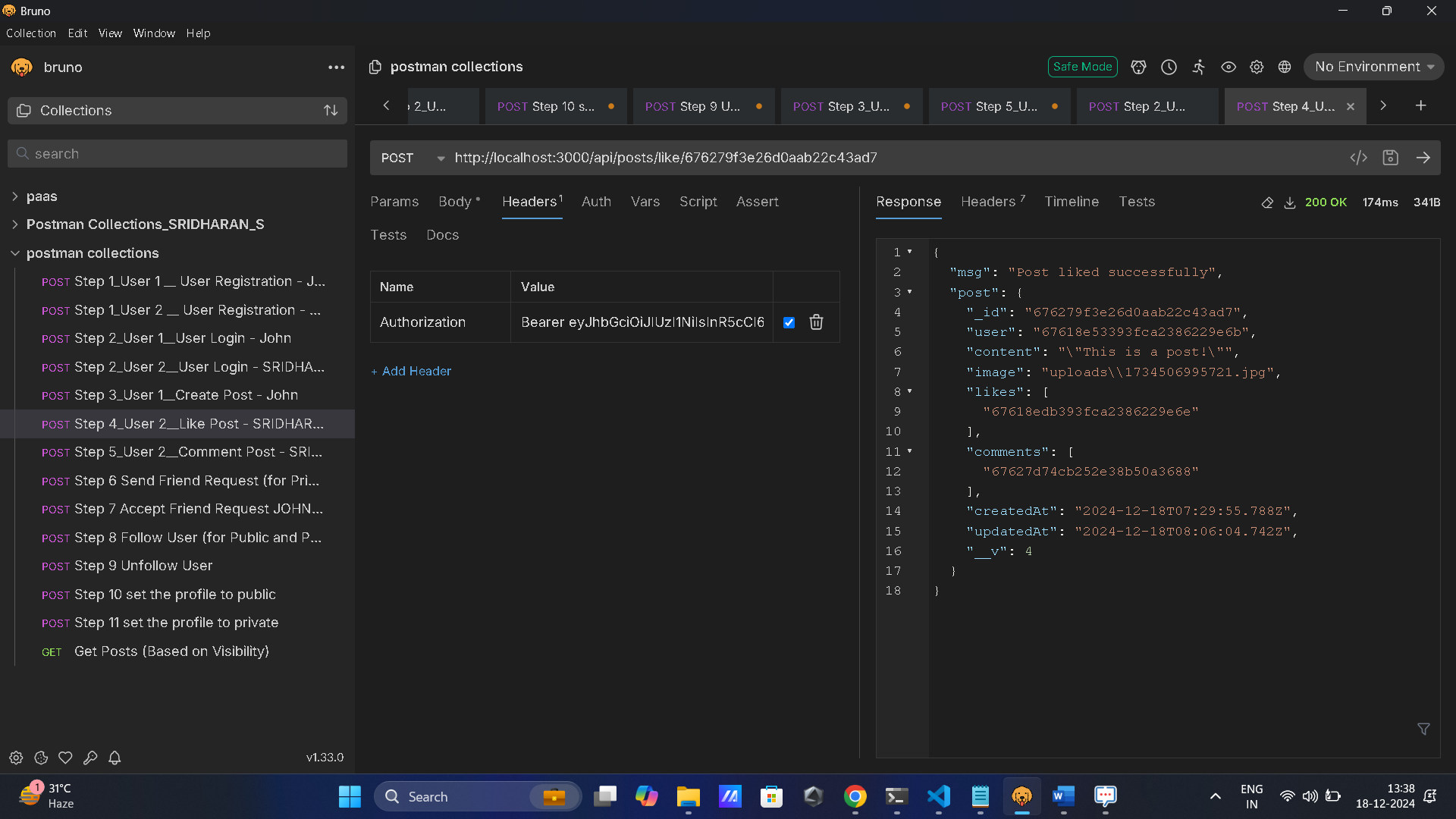


* Comment & Likes
* Example Screenshot Comments

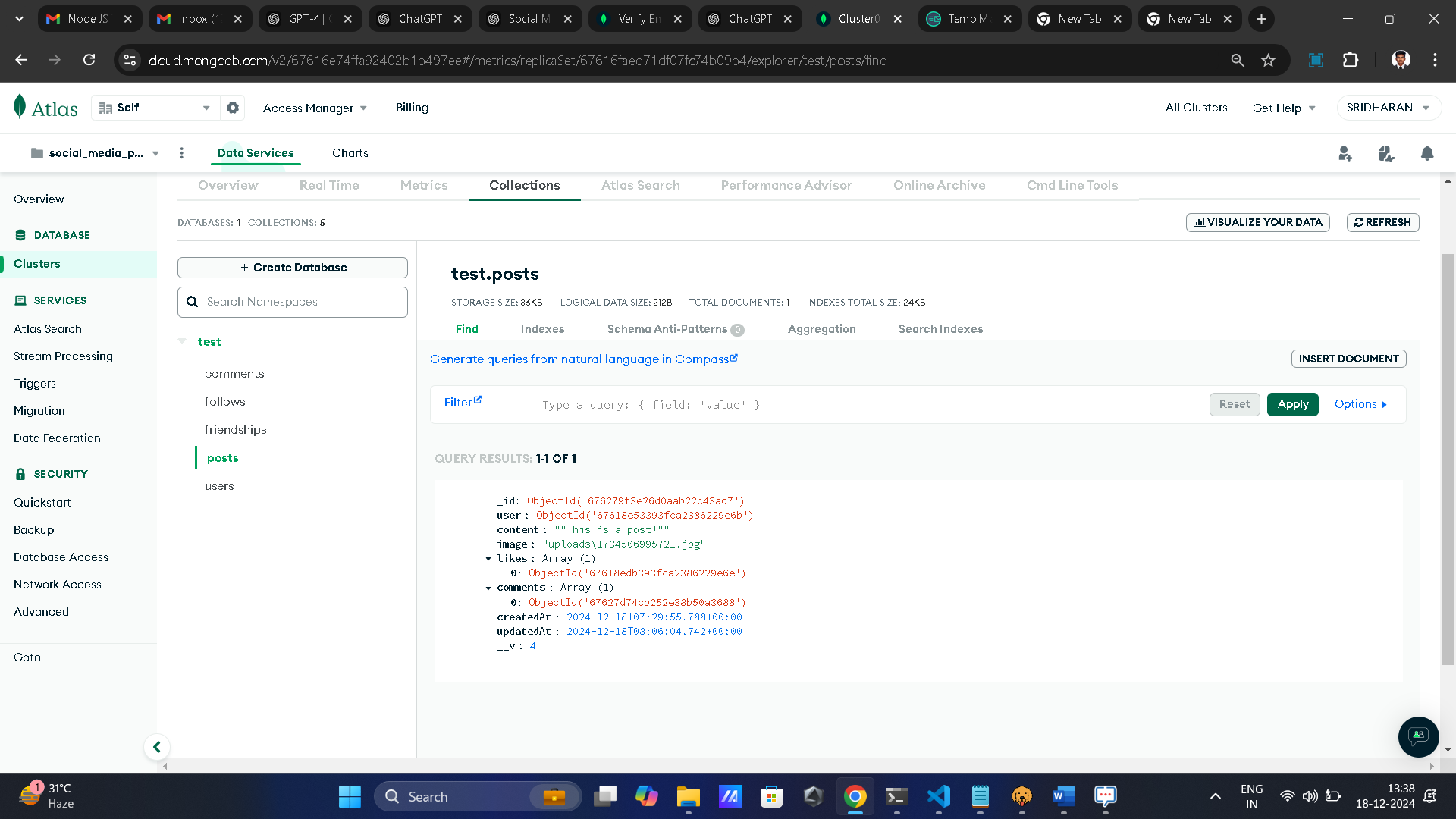




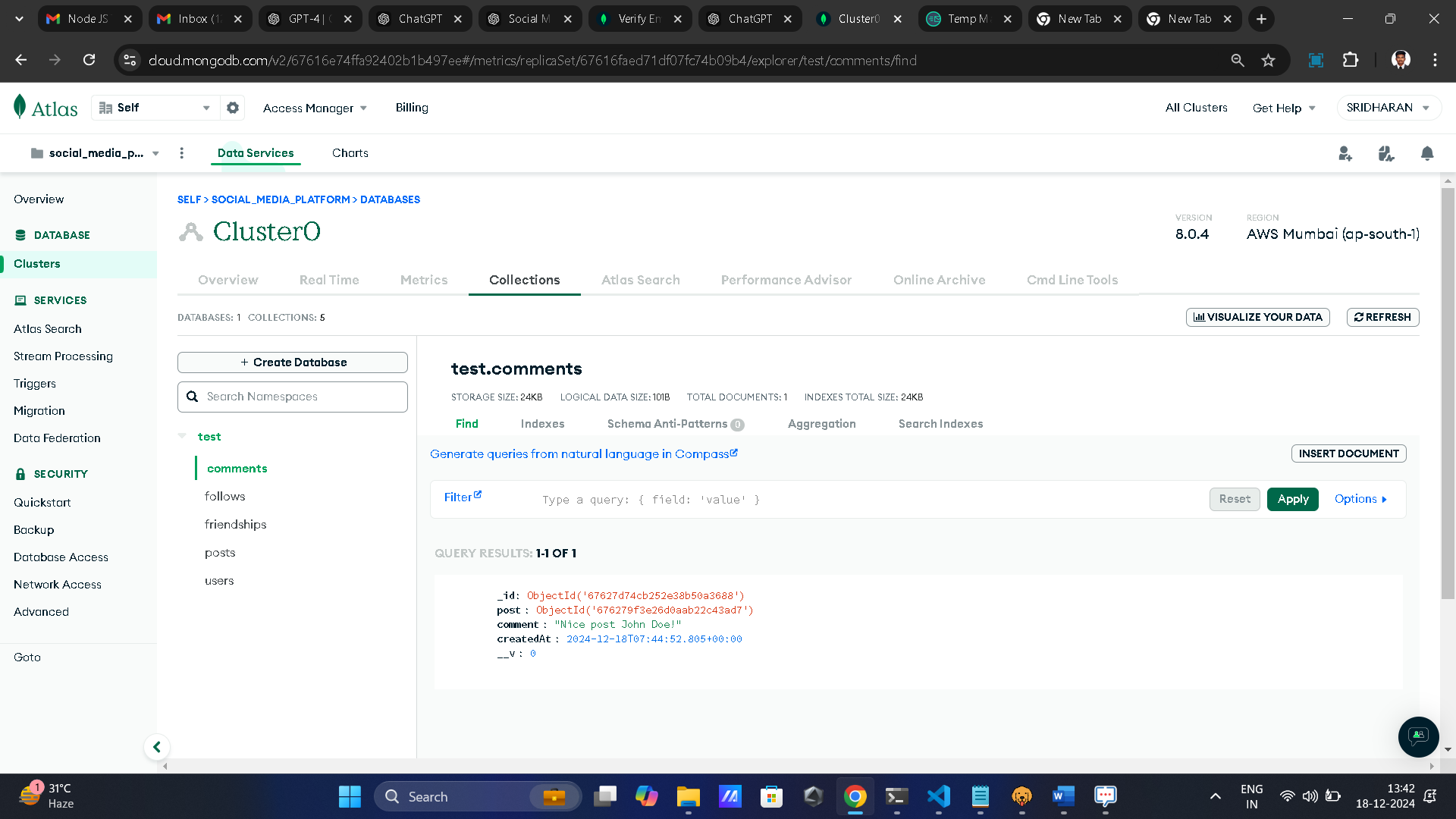
* Example Screenshot Likes



* Check Posts Schema



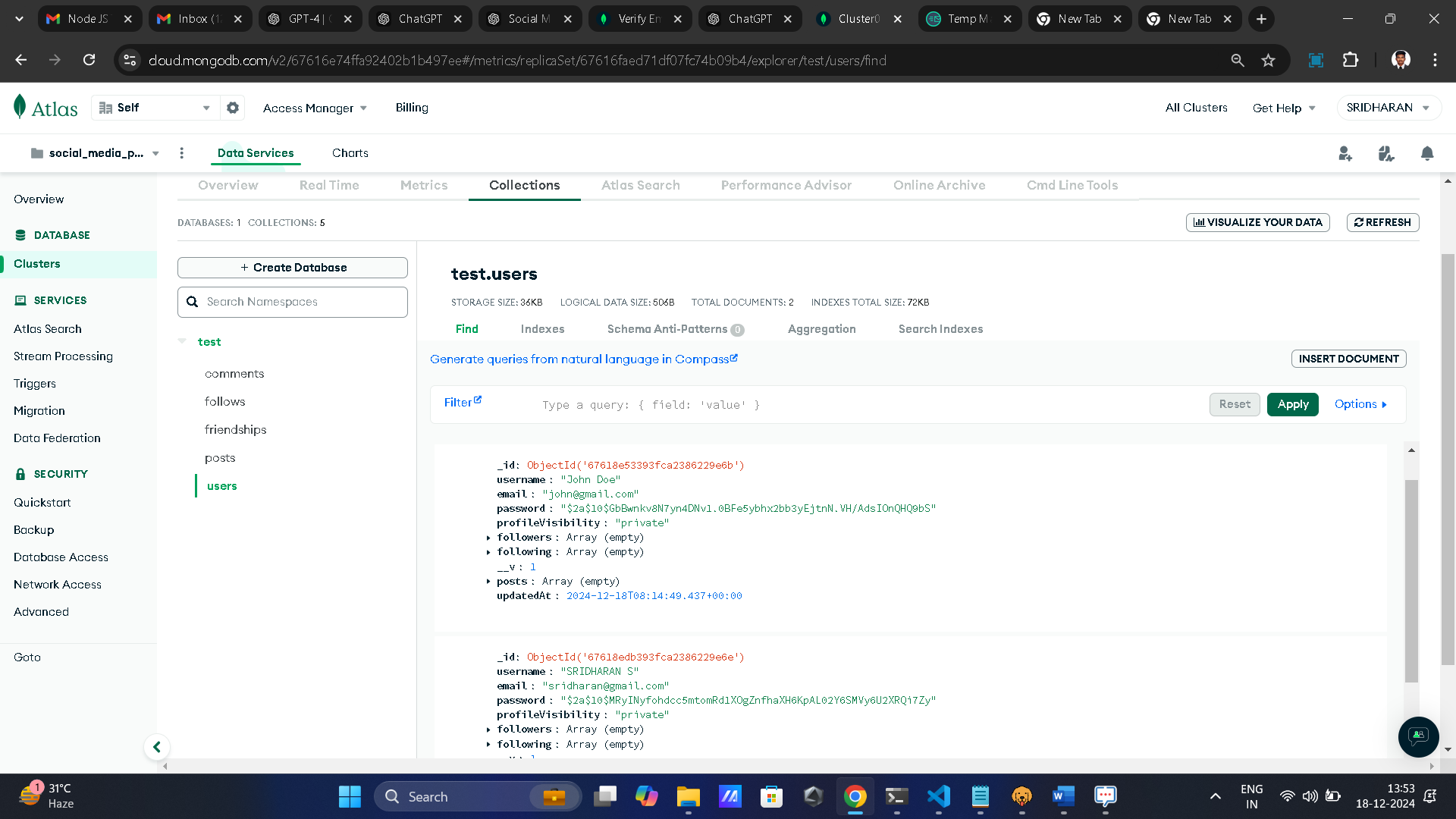
* Likes : Array (1)
* Comments : Array(1)
* Check Comments Schema



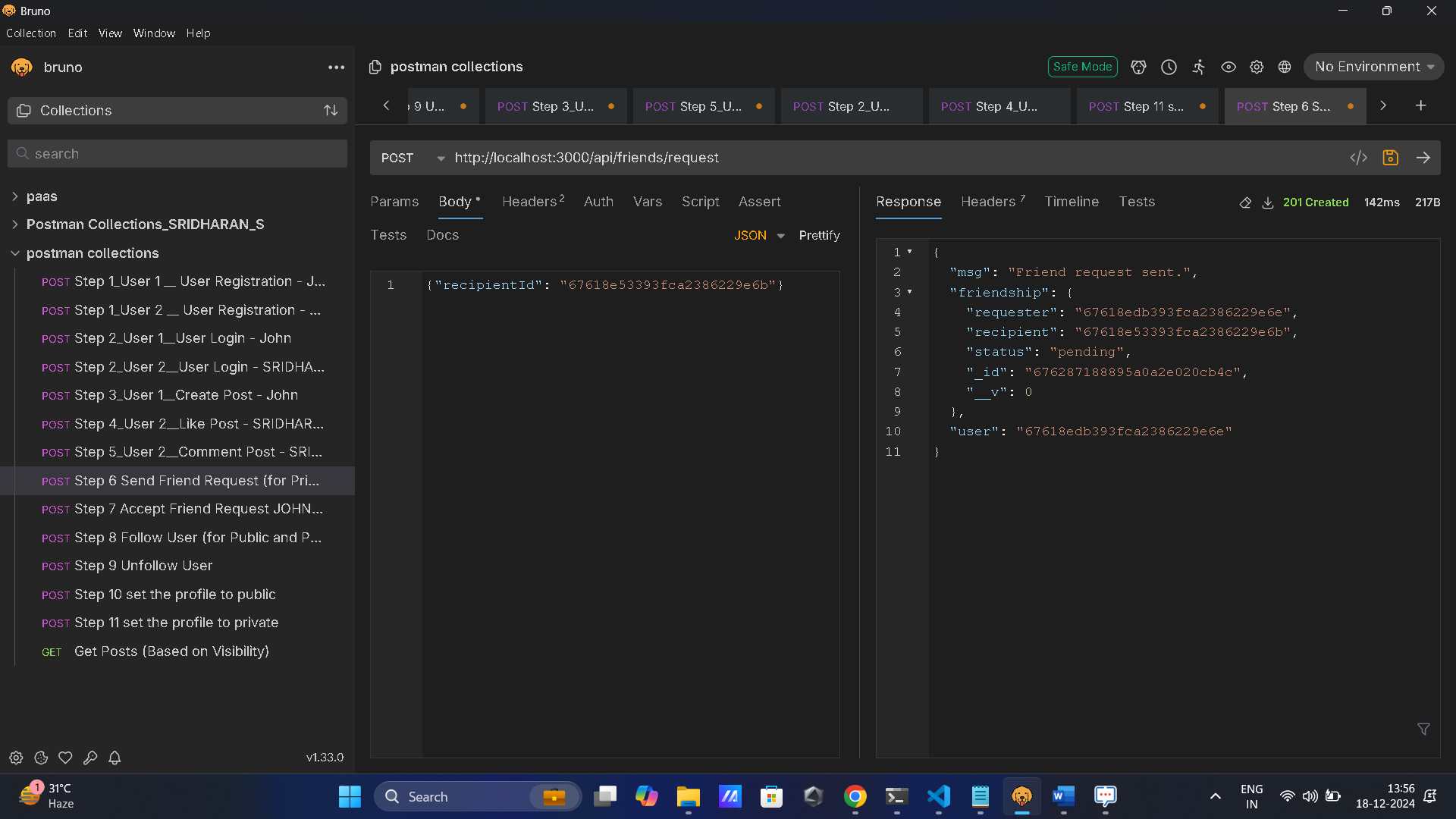
**4. Friendship Mechanism**

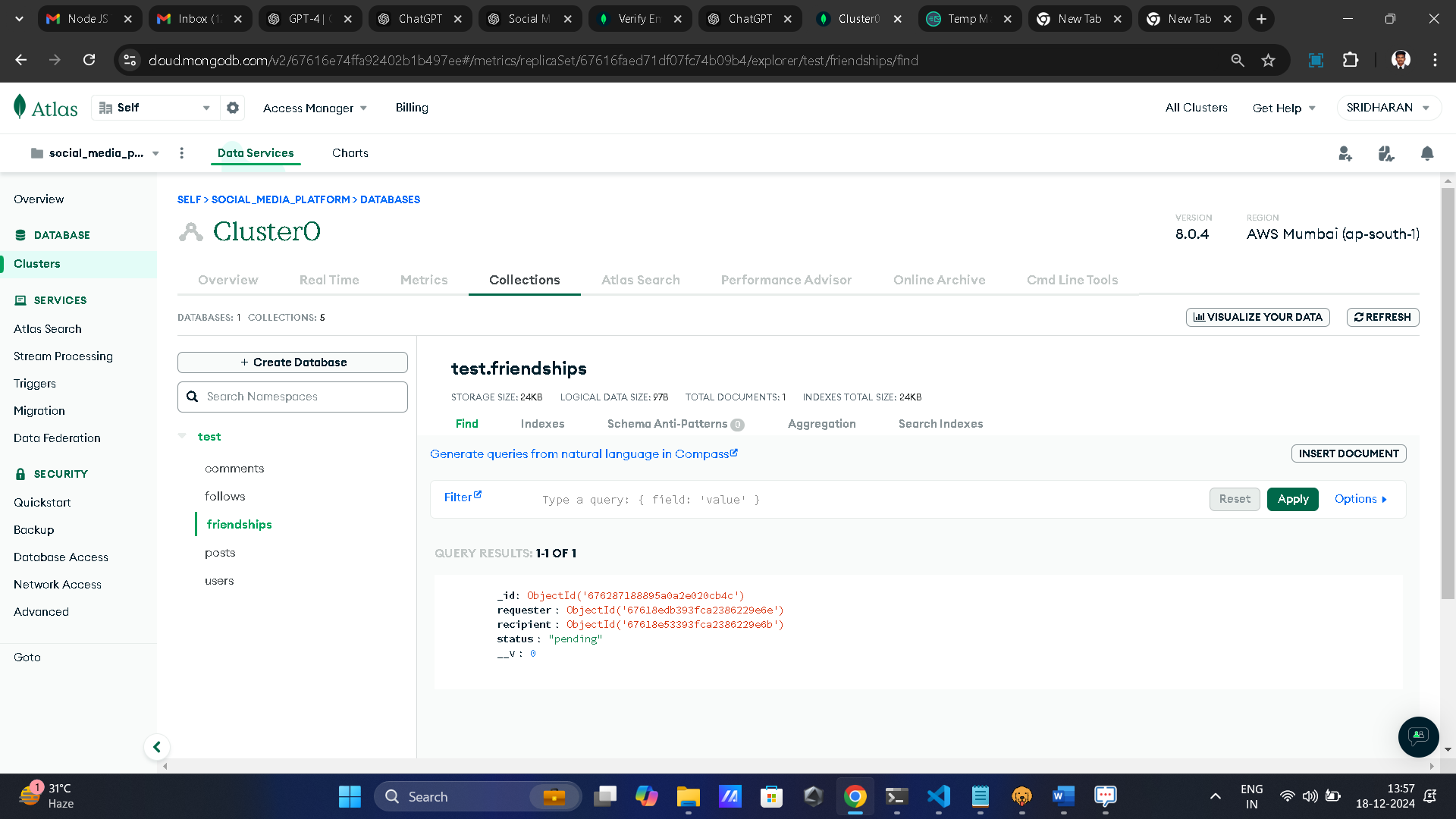
* Users can send, accept friend requests.
* Example:

**Screenshot**:

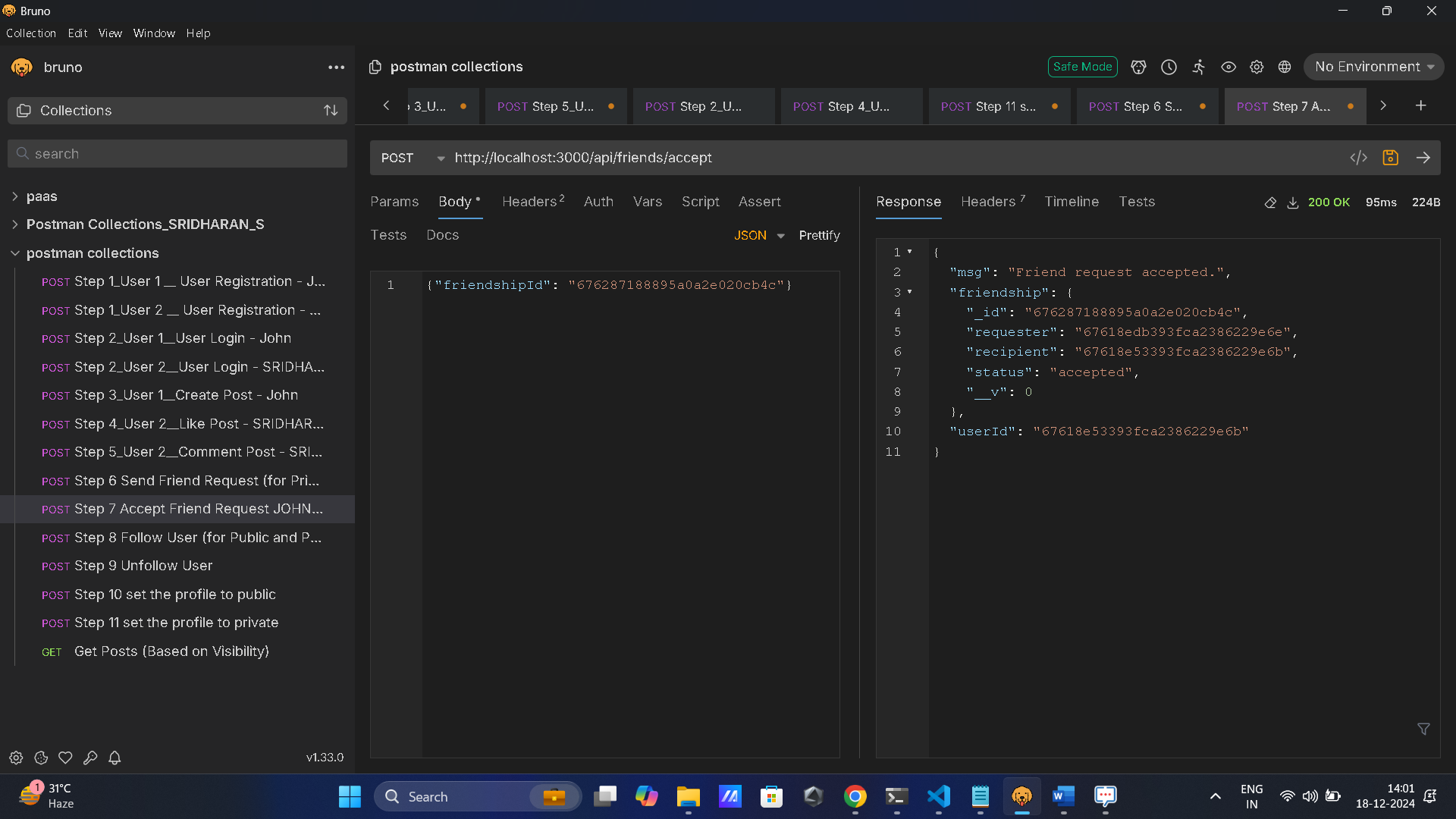


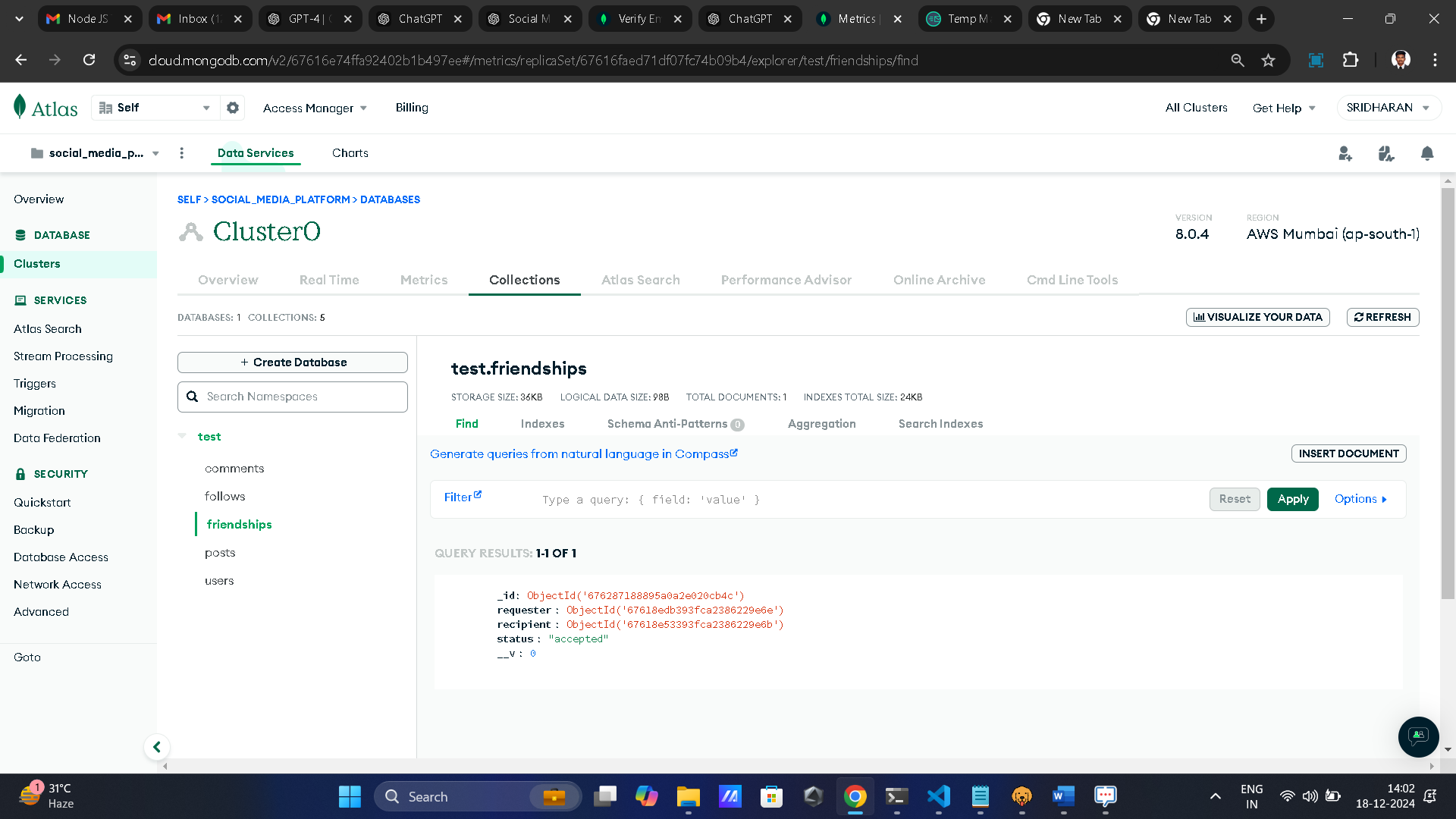
* Send Request





* **Status** : "pending"
* Accept Request

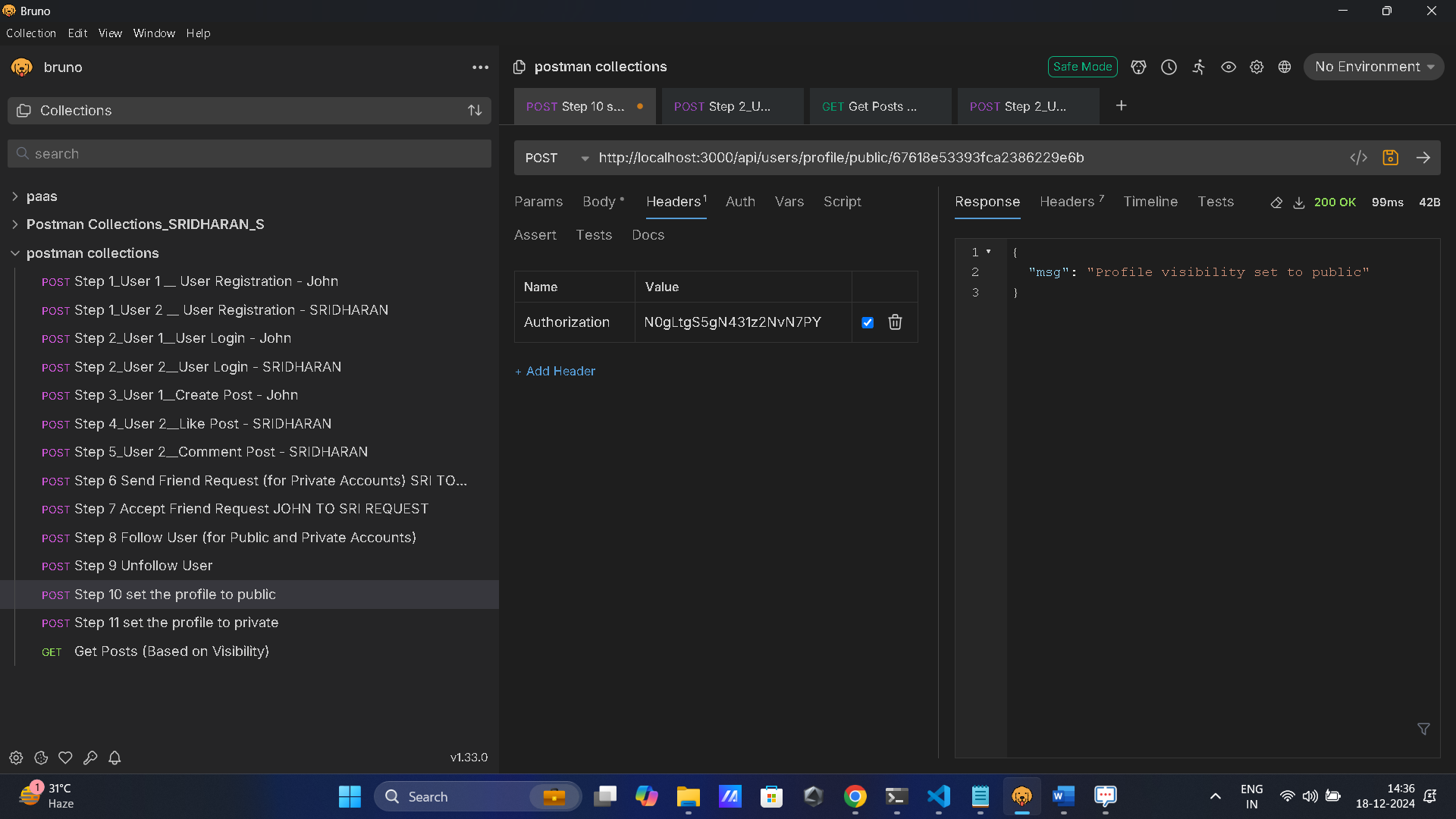


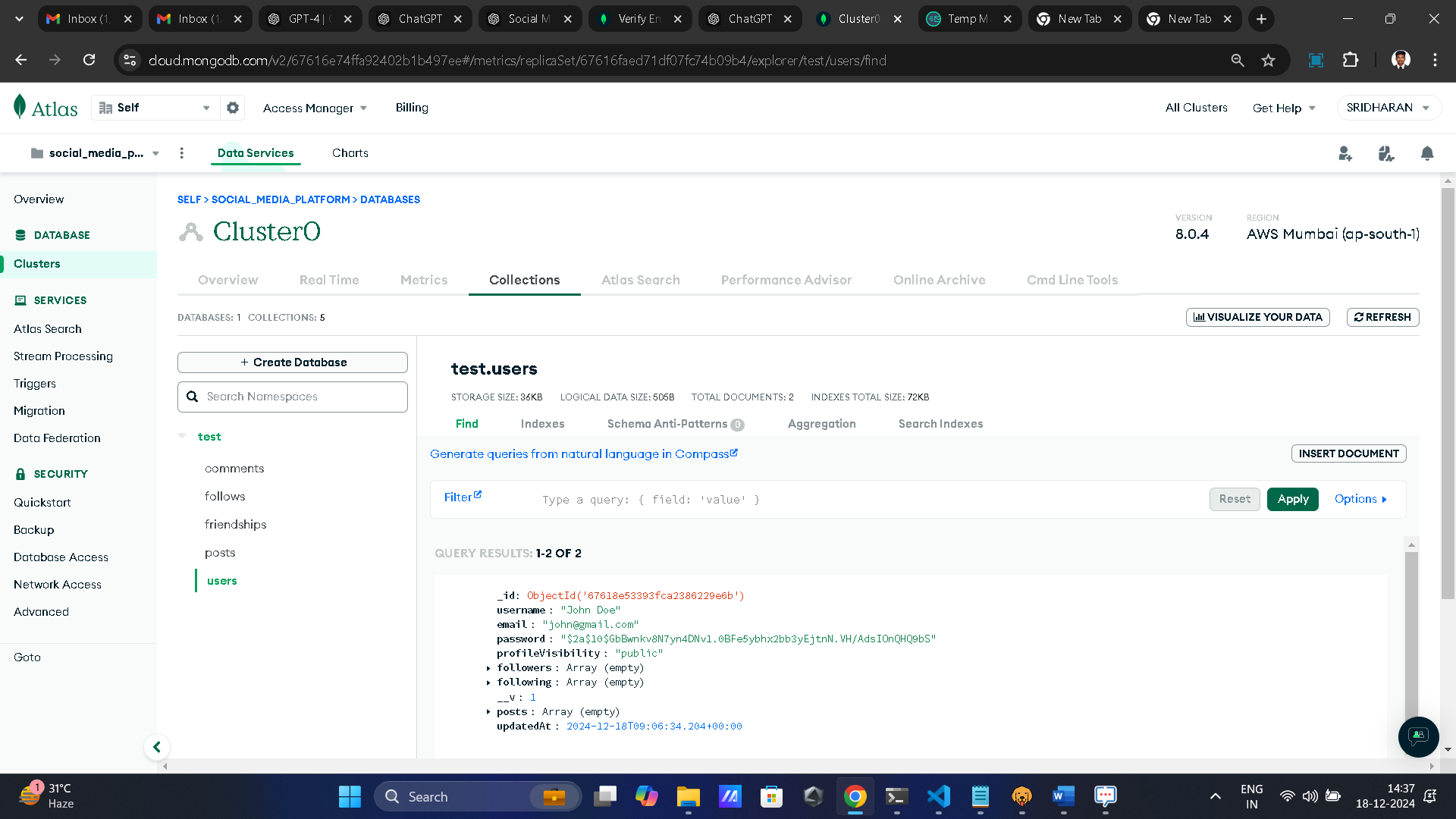


* **Status** : "accepted"

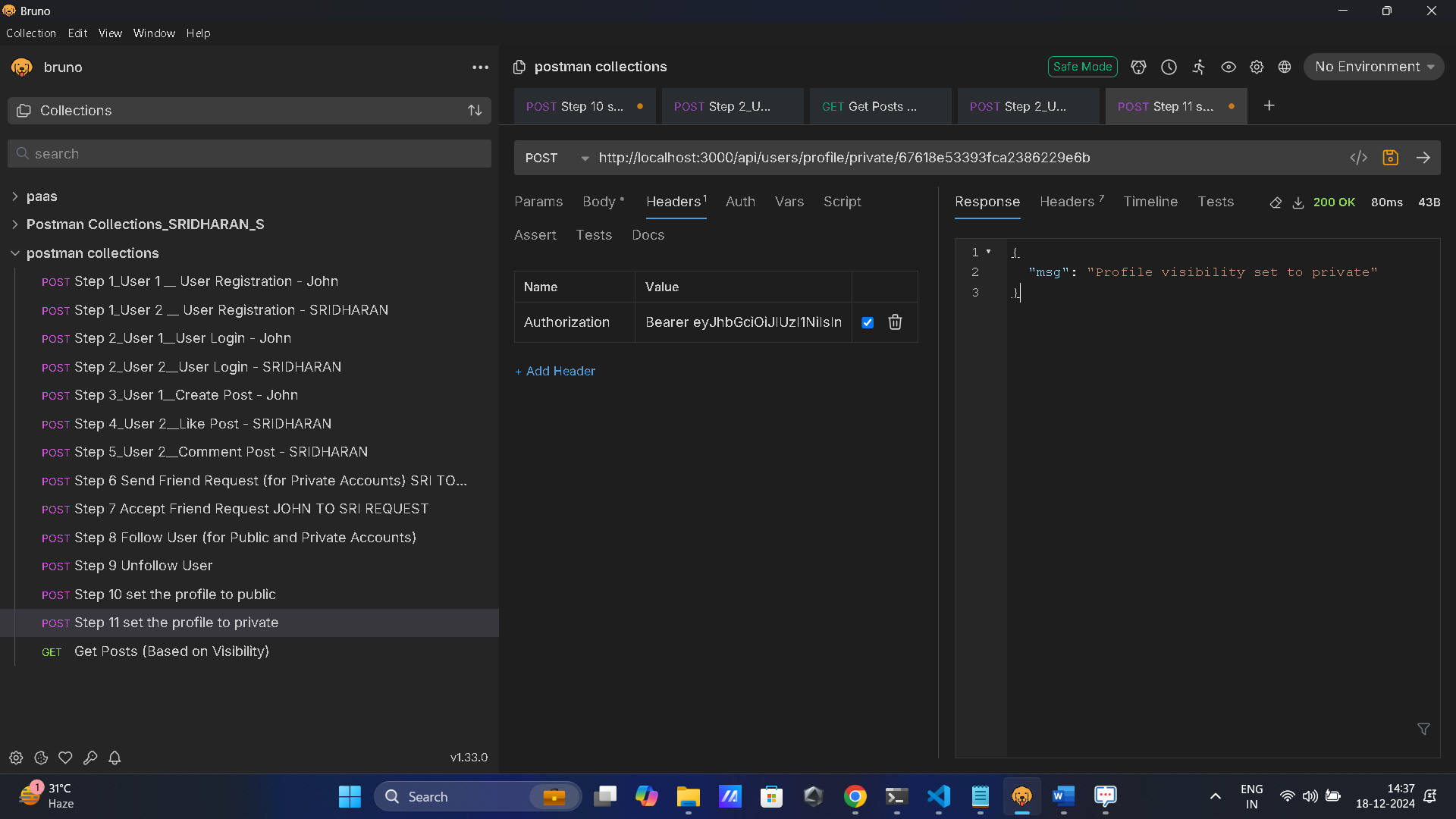
**5. Profile Visibility**

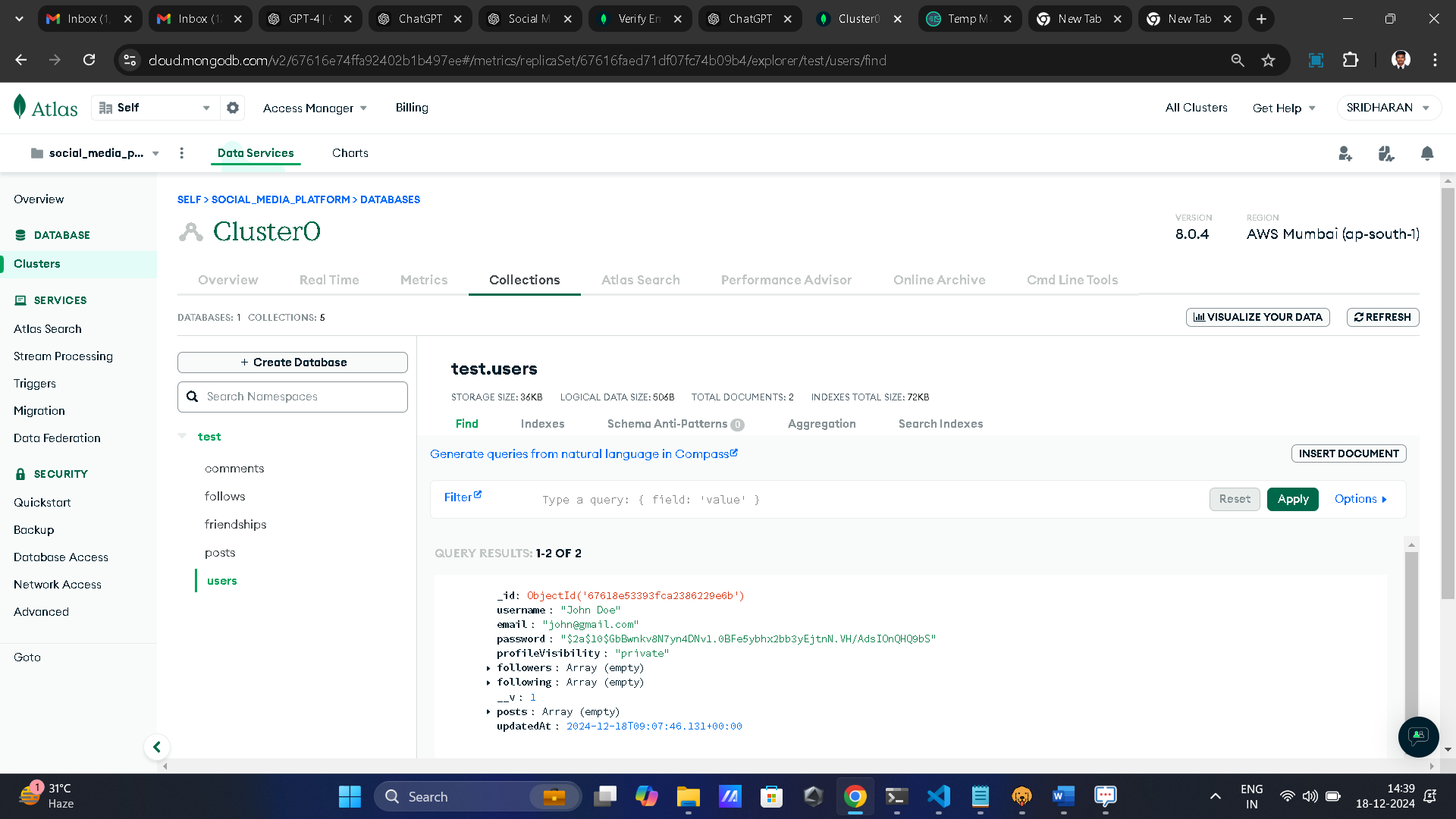
* Two visibility options for user profiles:
* Public



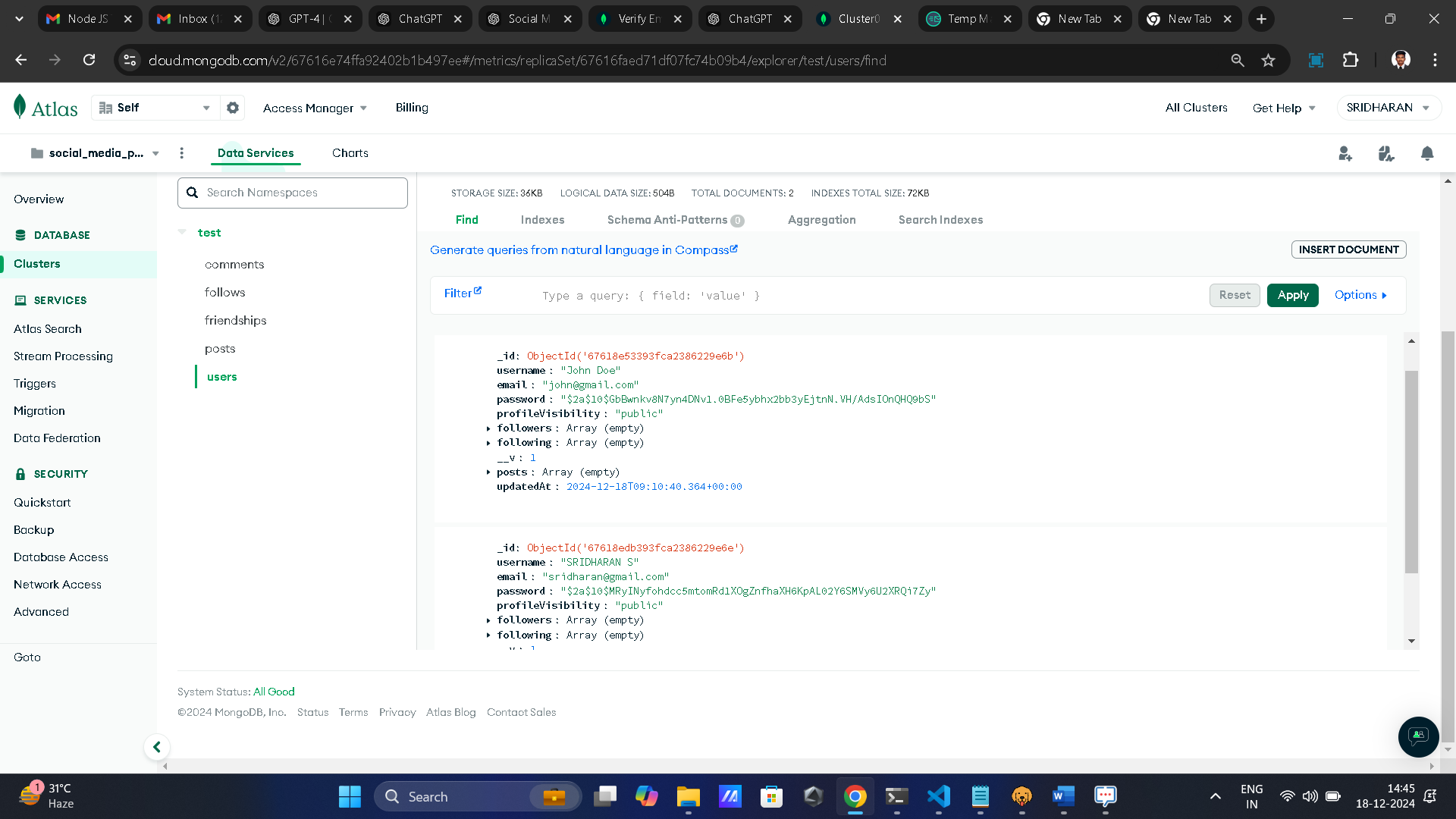


* Private

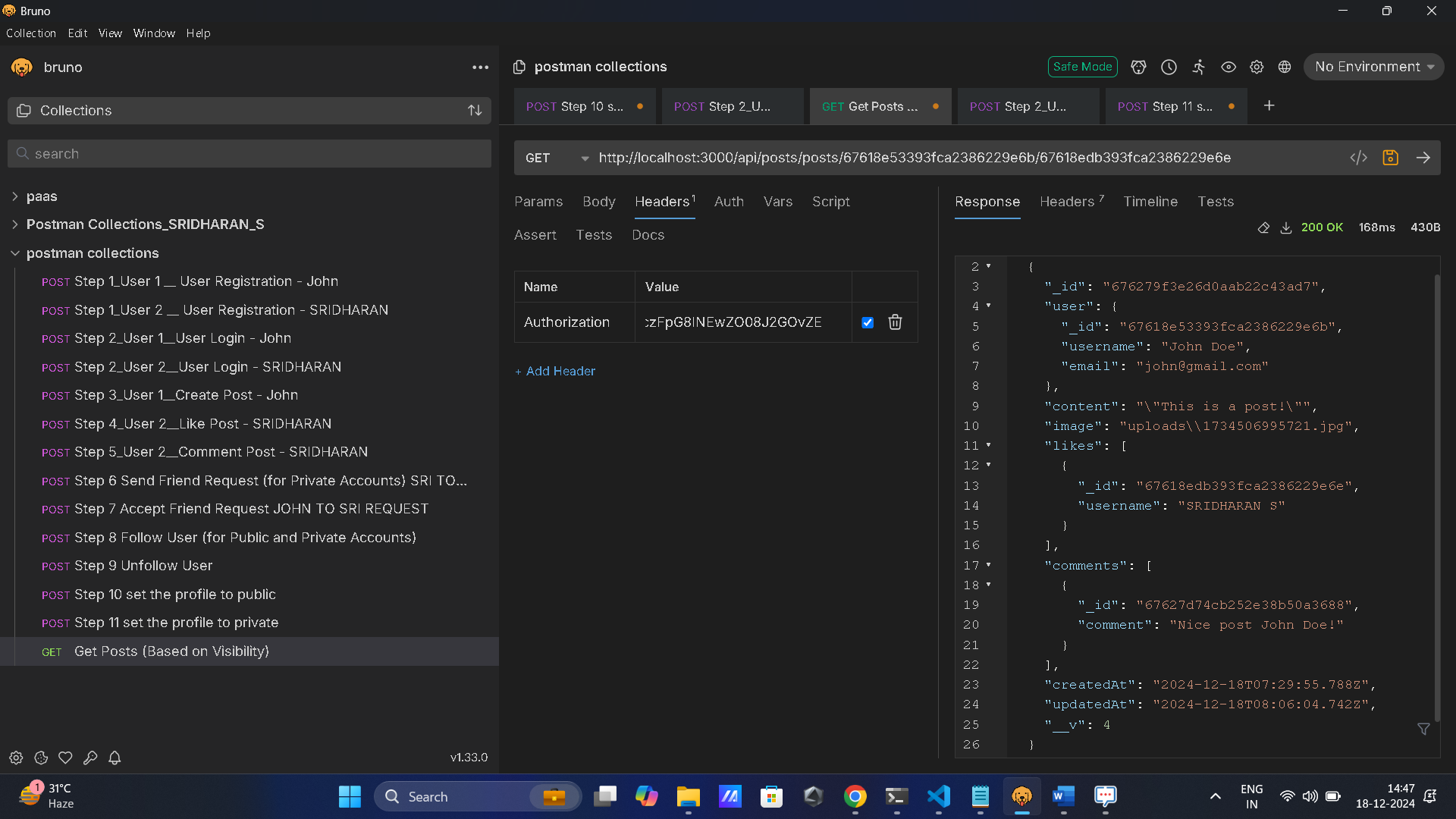




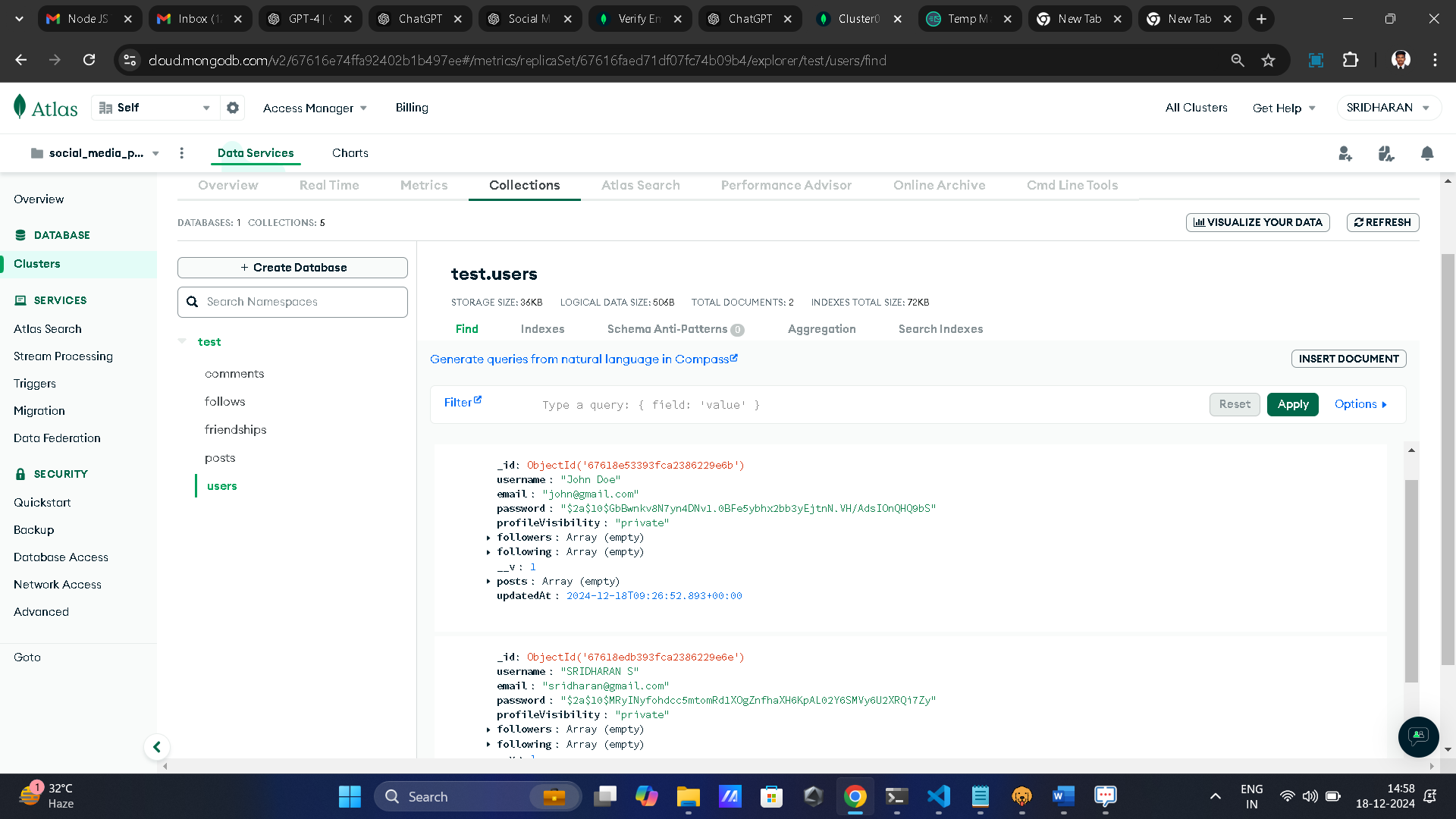
* + **Public**: Allows all followers to view posts.

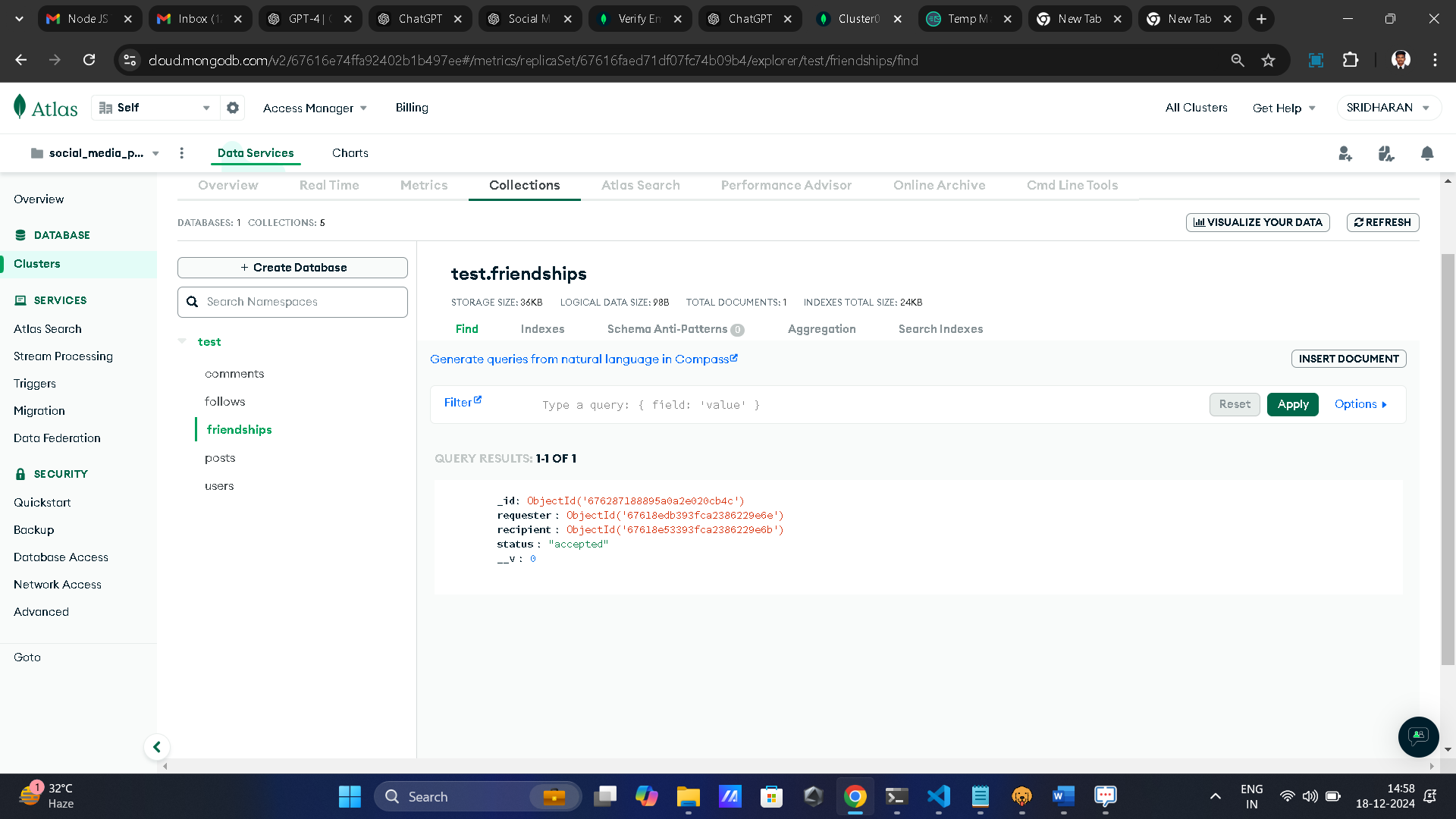


John Already Post Pictures Sridharan To View John Posts, Comments, Likes

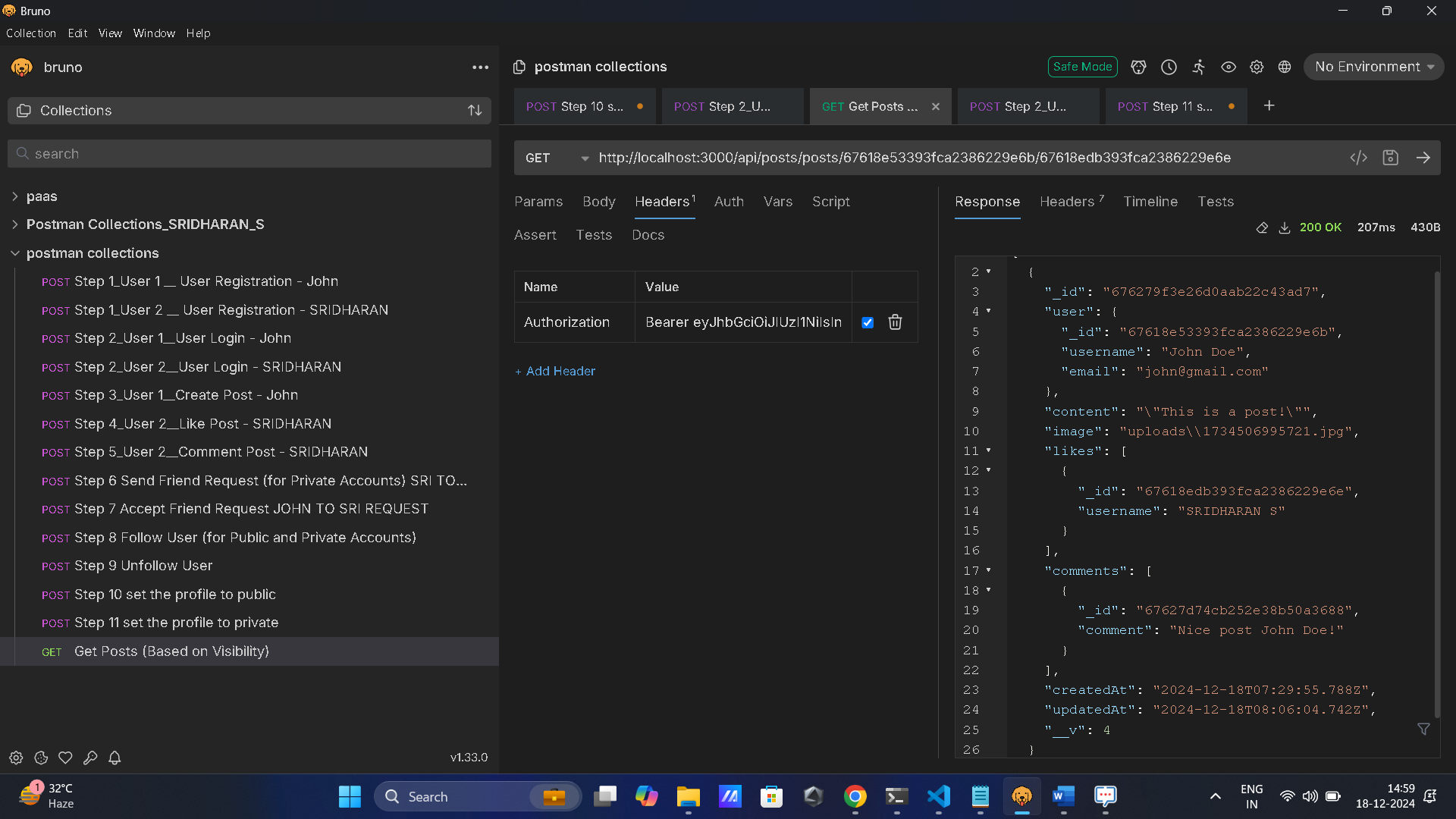


* + **Private**: Only friends can view posts.

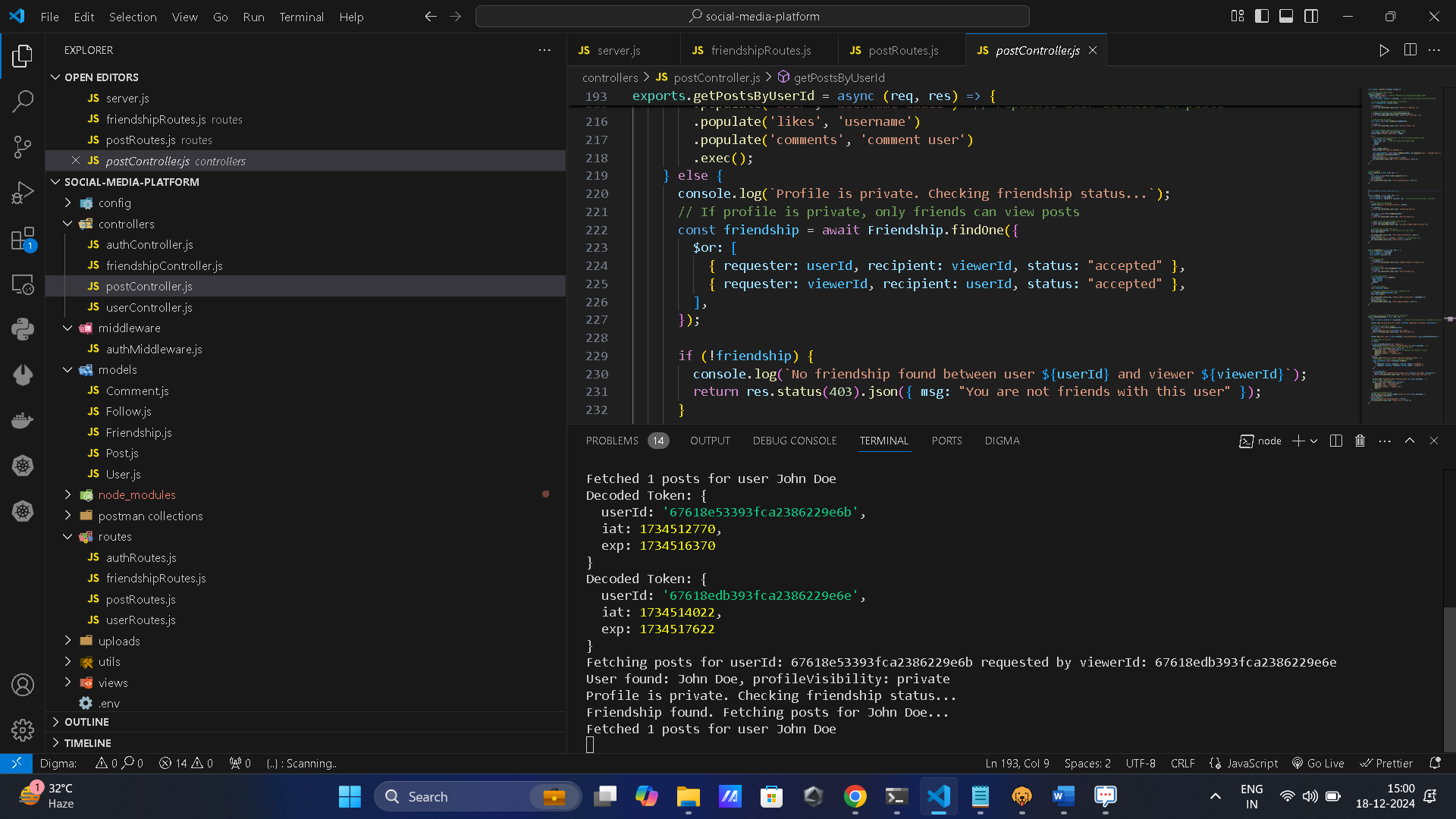




“status” : “accepted”



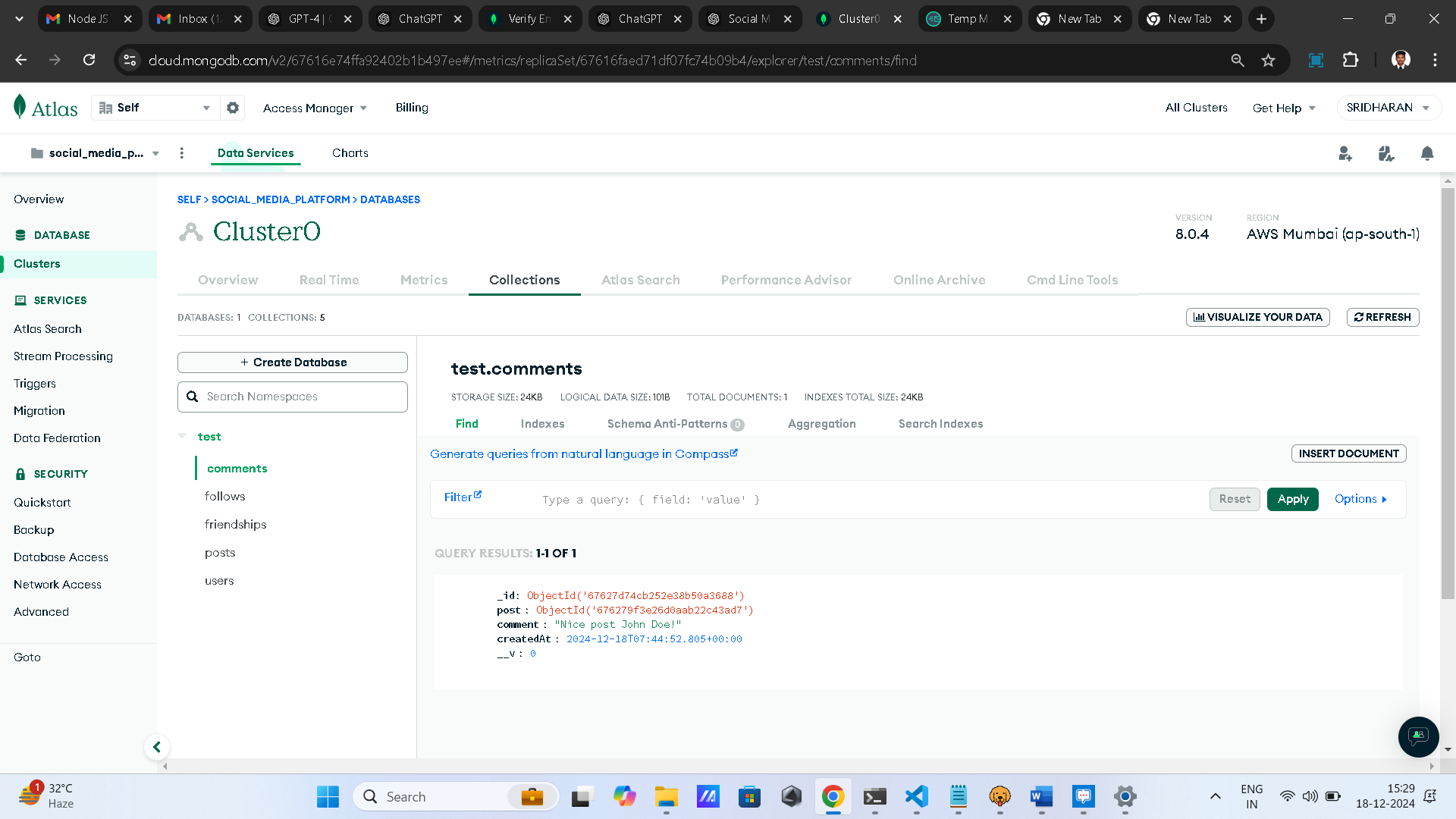
Vs code Console Log:



**Technology Stack**

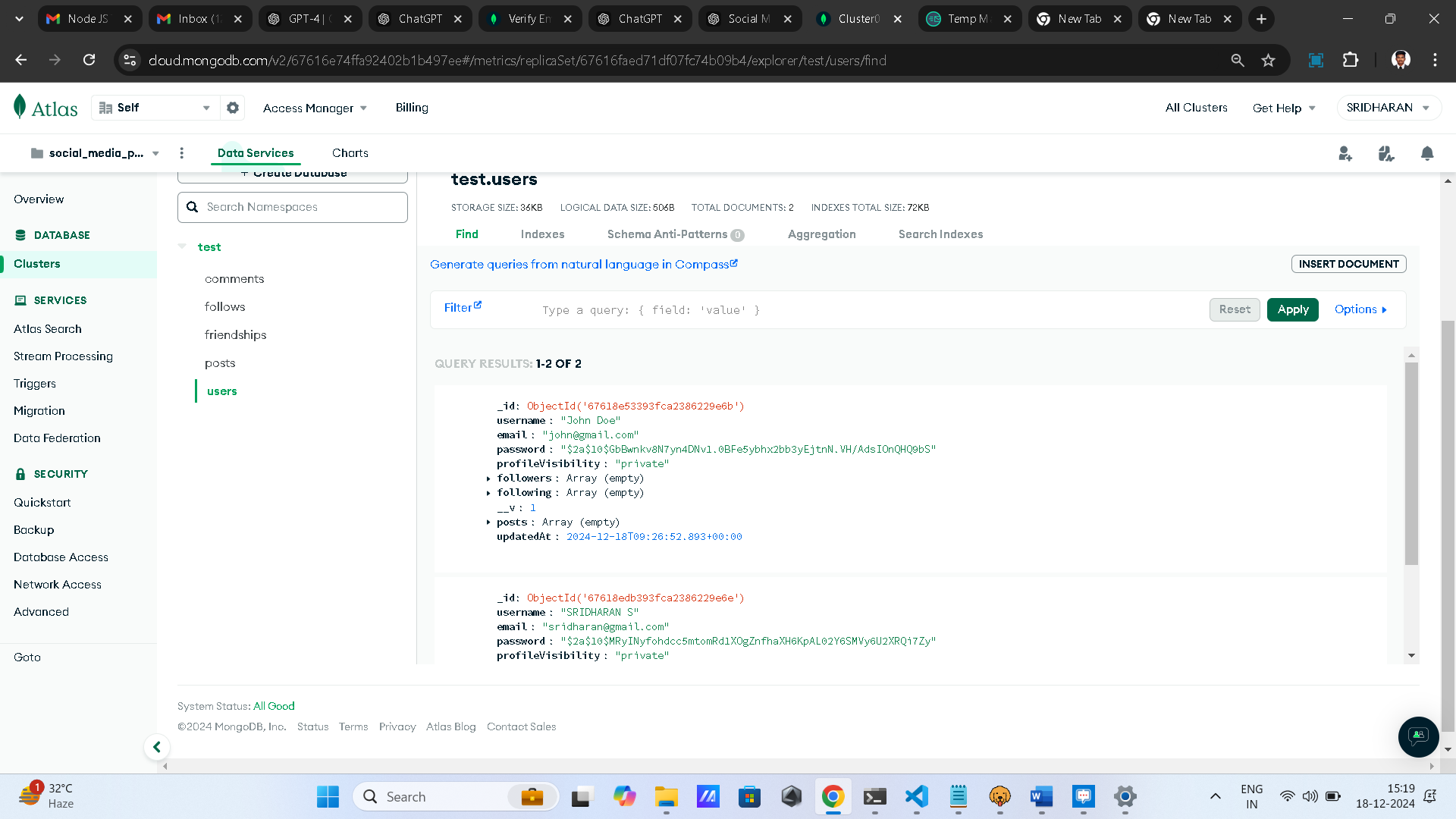
* **Backend**: Node.js, Express.js
* **Database**: MongoDB (or PostgreSQL)
* **Authentication**: JSON Web Token (JWT)
* **File Uploads**: Multer
* **Password Hashing**: Bcrypt

**Database Schema (MongoDB Example)**



**User Table**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| \_id | ObjectId | Unique identifier for the user. |
| username | String | User's username. |
| email | |  | | --- | |  |   String | User's email. |
| password | String | Hashed password. |
| profileVisibility | Enum (public/private) | Visibility of the profile. |
| followers | Array of ObjectId | List of users following this user. |
| following | |  |  | | --- | --- | | Array of ObjectId |  | | List of users this user is following. |
| posts | Array of ObjectId | List of posts created by the user. |



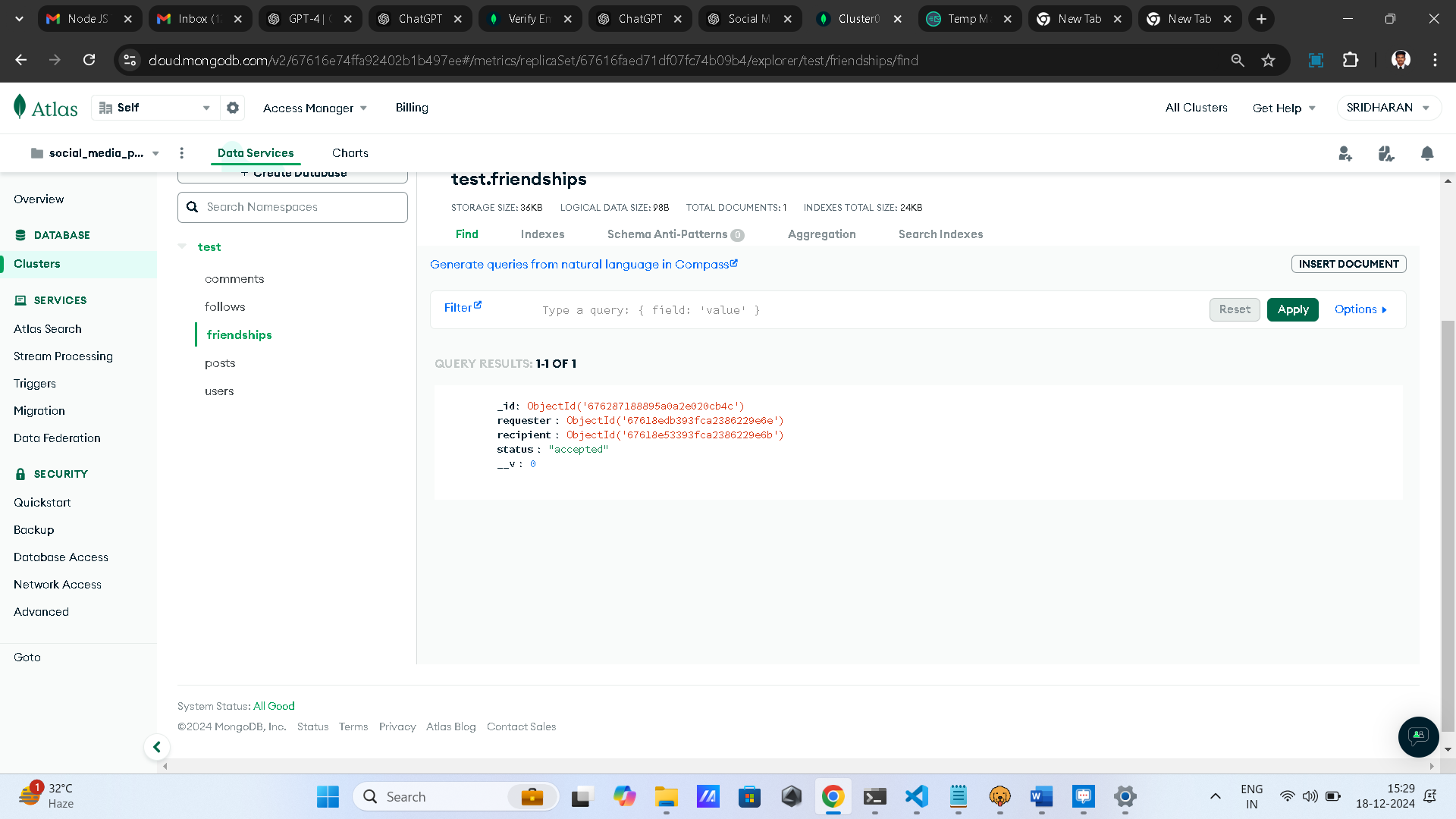
**Post Table**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| \_id | ObjectId | Unique identifier for the post. |
| userId | ObjectId | Reference to the user who created it. |
| content | String | Content of the post. |
| image | String | |  |  | | --- | --- | |  | Path to the uploaded image. | |
| likes | Array of ObjectId | List of users who liked the post. |
| comments | |  |  | | --- | --- | |  | Array of Comments | | Comments on the post. |



**Friendship Table**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| \_id | ObjectId | Unique identifier for the friendship. |
| requester | ObjectId | User who sent the friend request. |
| recipient | ObjectId | User who received the friend request. |
| status | Enum (pending/accepted) | Status of the friend request. |



**Security Measures**

* **JWT Authentication**: Secures API endpoints to ensure only authenticated users can access resources.
* **Password Hashing**: All passwords are hashed before storage.
* **Authorization Middleware**: Ensures proper access control to posts and profiles based on visibility settings.

**Installation & Setup**

1. Clone the repository:

git clone https://github.com/SRIDHARAN1819/social-media-platform.git

cd social-media-platform

1. Install dependencies:

npm install

1. Configure environment variables in a .env file:

PORT=3000

MONGO\_URI=<your-mongodb-connection-string>

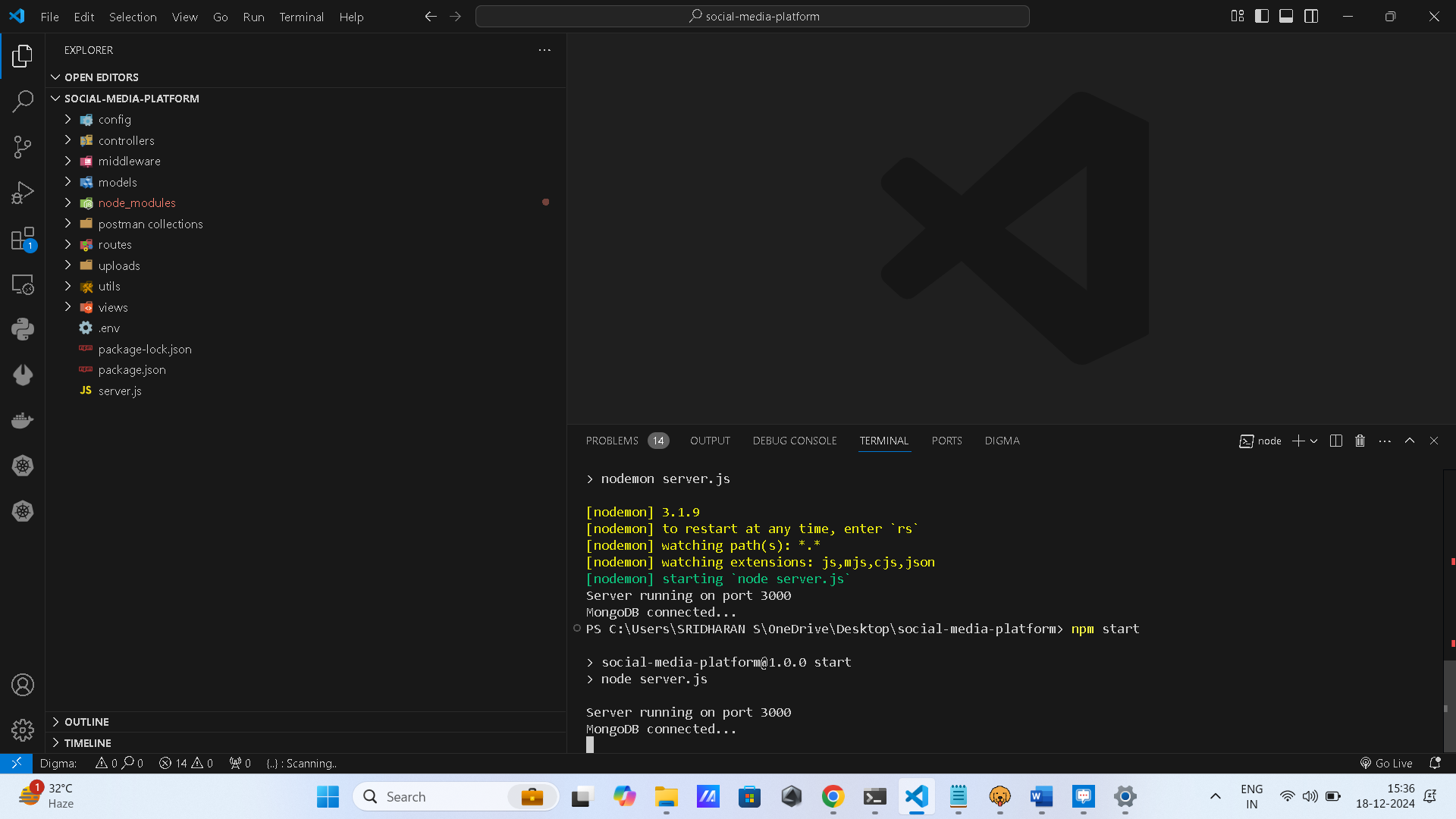
JWT\_SECRET=<your-secret-key>

1. Start the server:

npm start

1. Access the application at http://localhost:3000.

Project Structure



**Conclusion**

This documentation outlines the development of a scalable and secure social media platform with key features like user registration, post creation, interaction, and profile management. The project structure and best practices ensure that the application is robust and maintainable.