[Repository]

[Live Link]

Social Media Platform Development

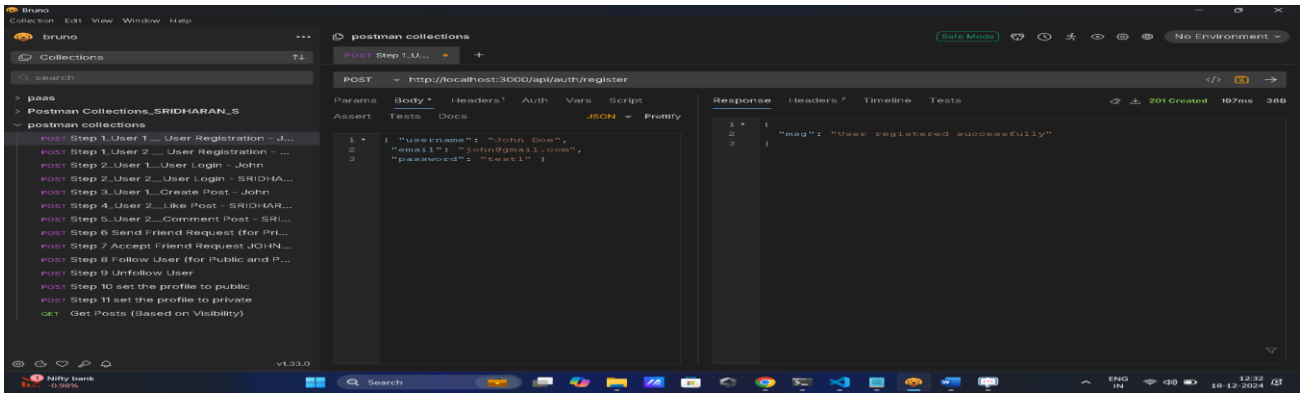
Project Overview

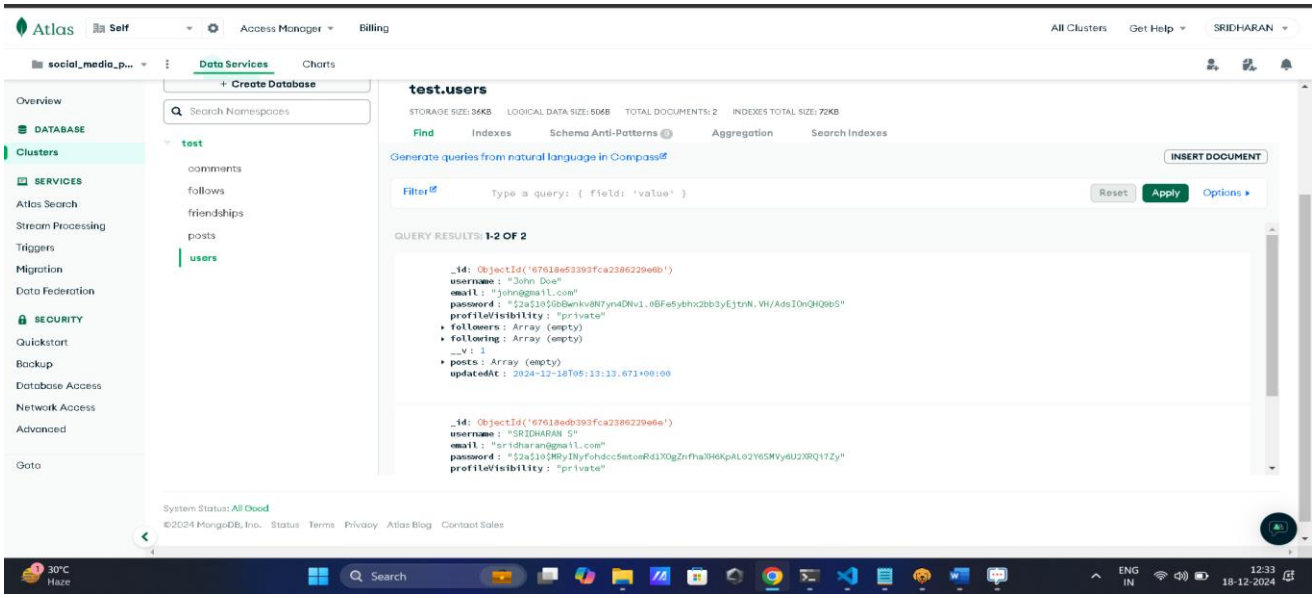
The purpose of this project is to create a social media platform inspired by Facebook's functionality. The application will allow users to register, interact, and manage their relationships and posts. It is built using **Node.js**, **Express**, and **MongoDB** for database management.

Features

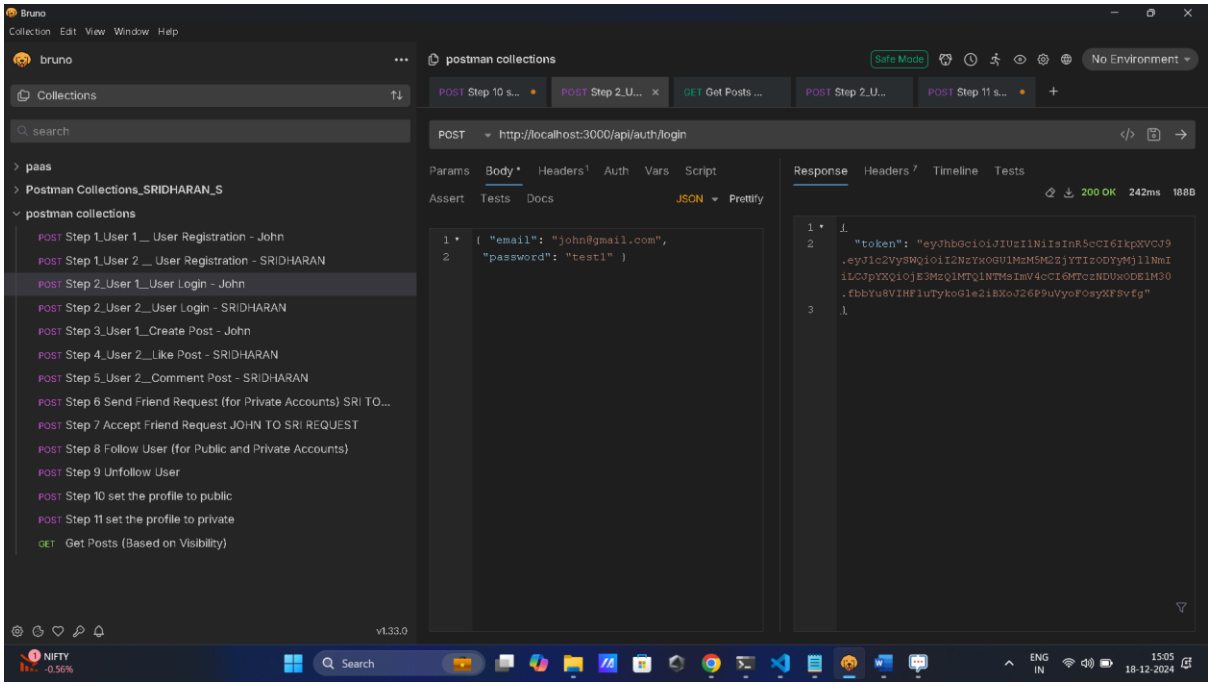
1. User Registration

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





Login

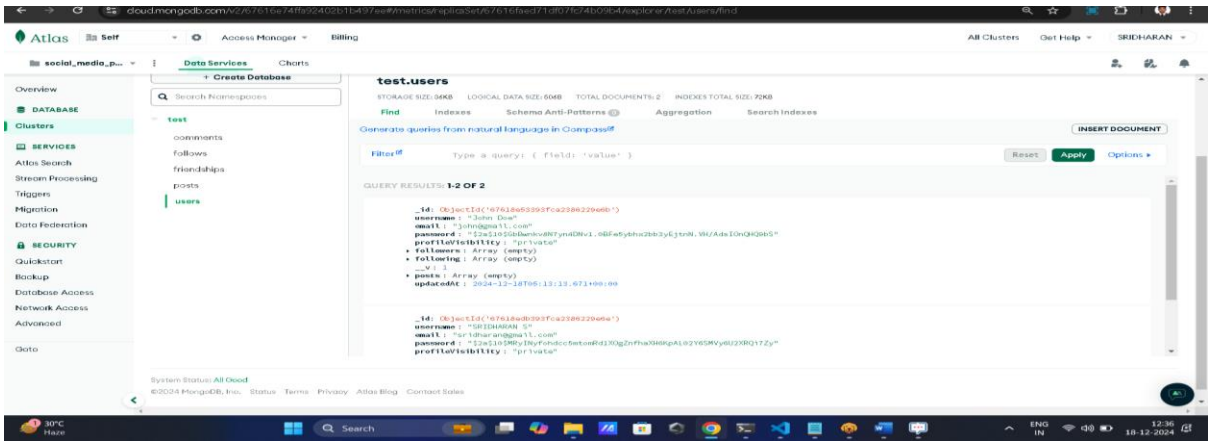


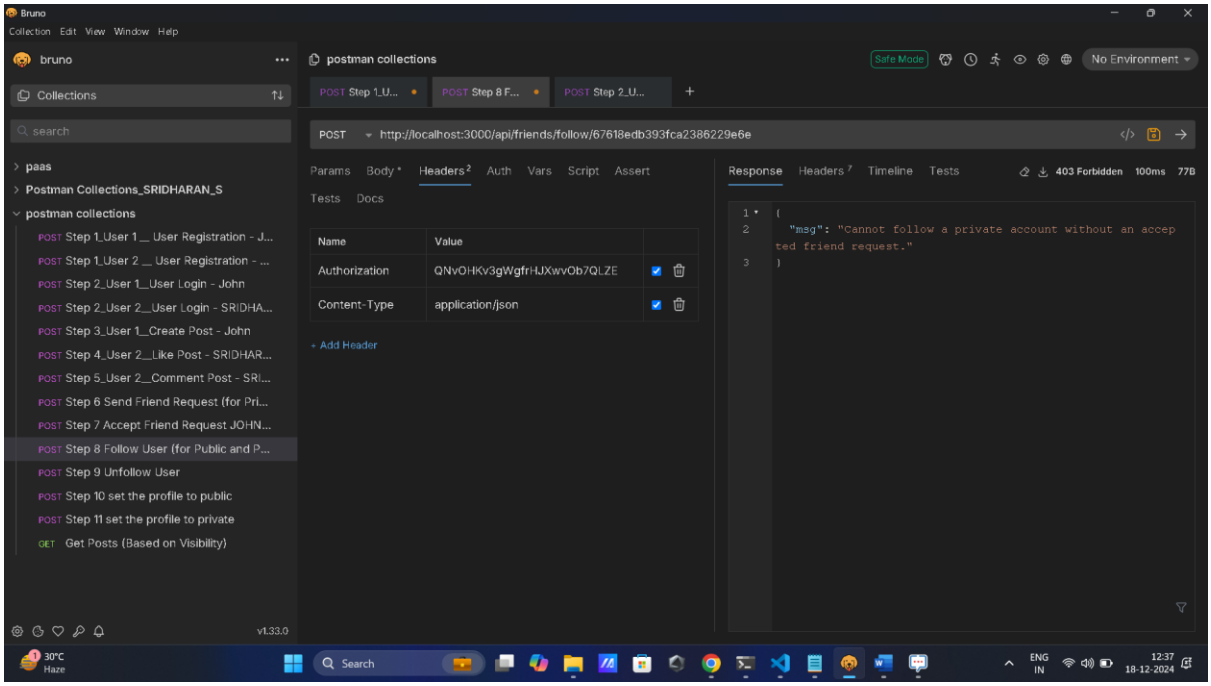
2. Follow System

- 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."

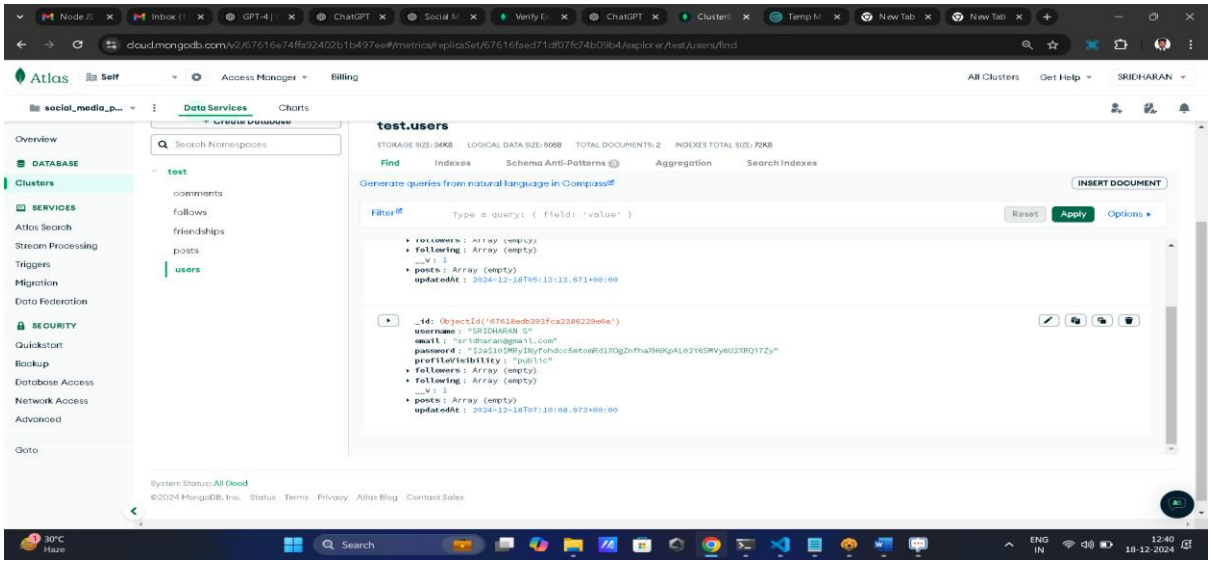
- **Screenshot**:

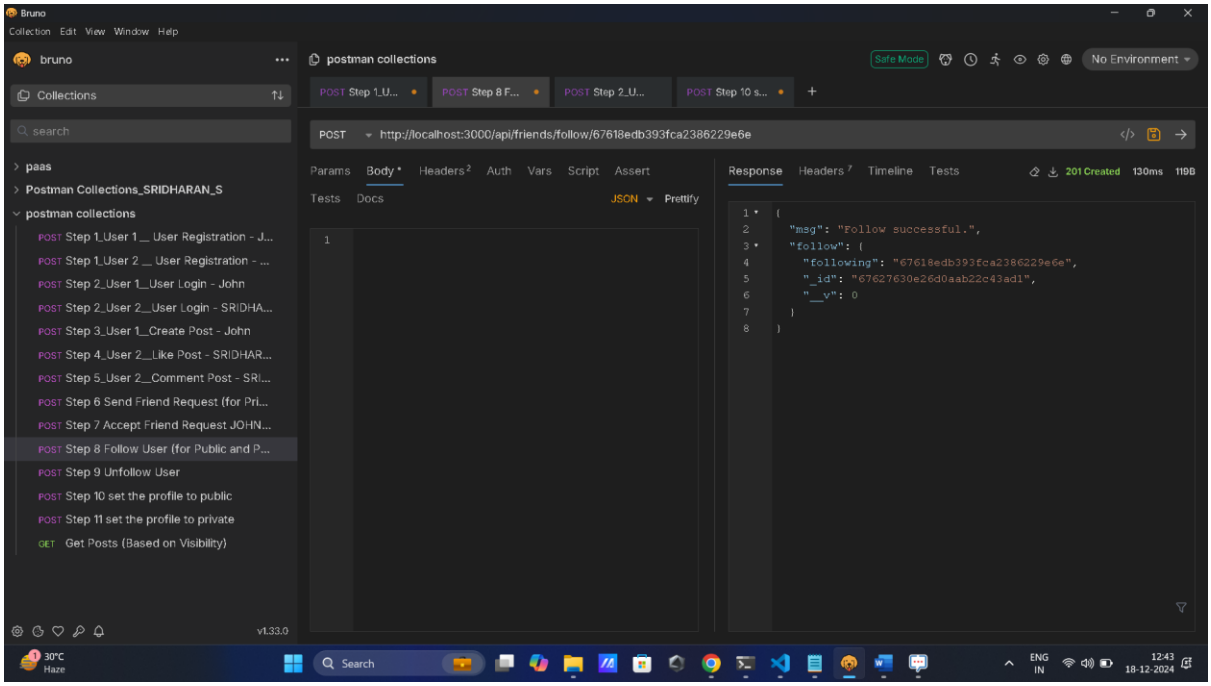




- Example: Public Account Means
- Check Users table
  profileVisibility : "public"

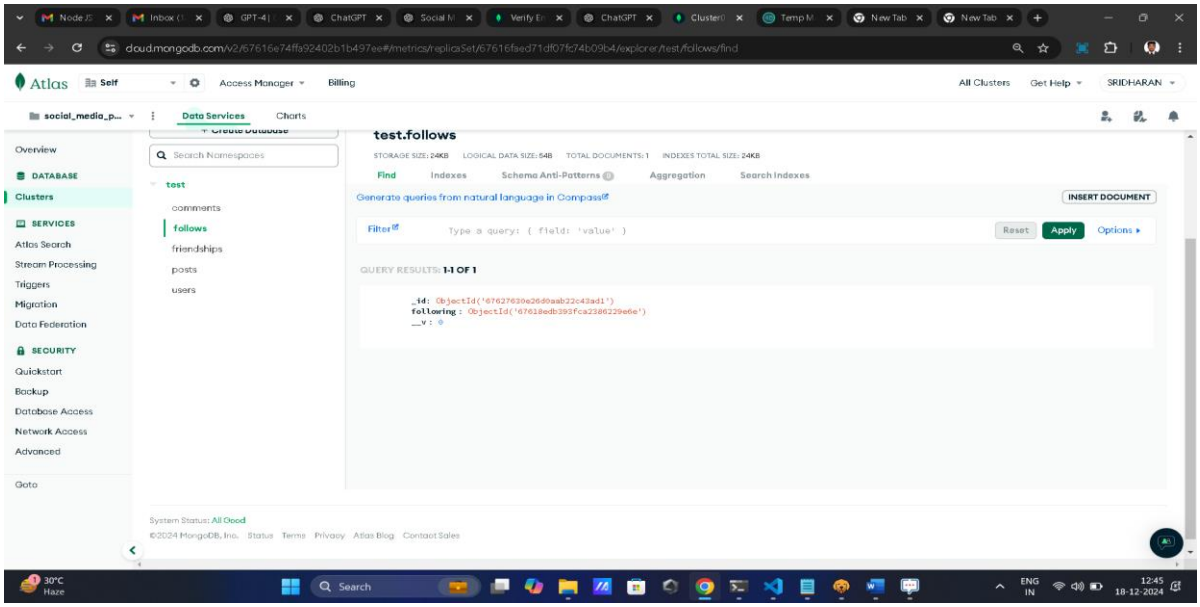
Screenshot:



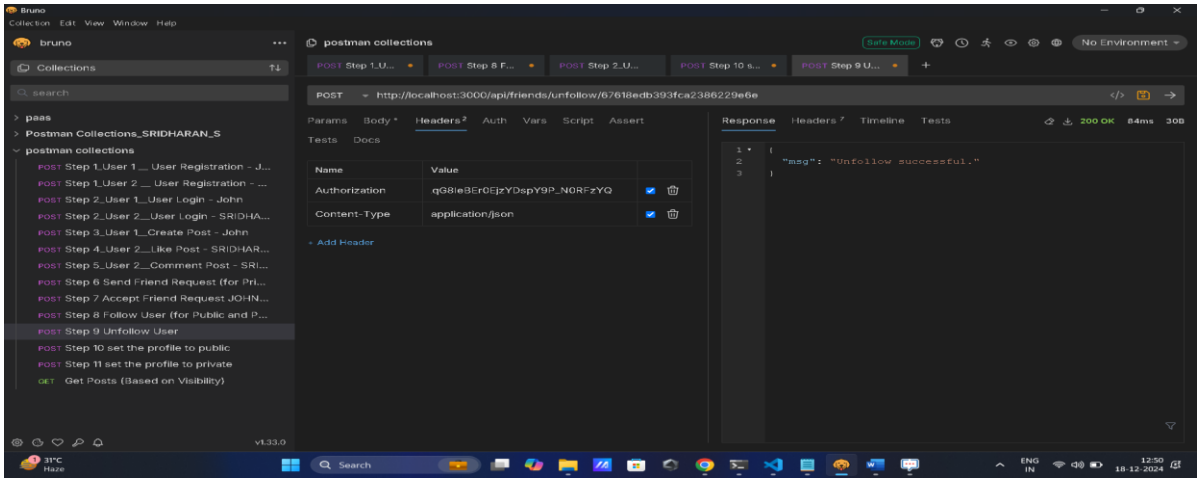


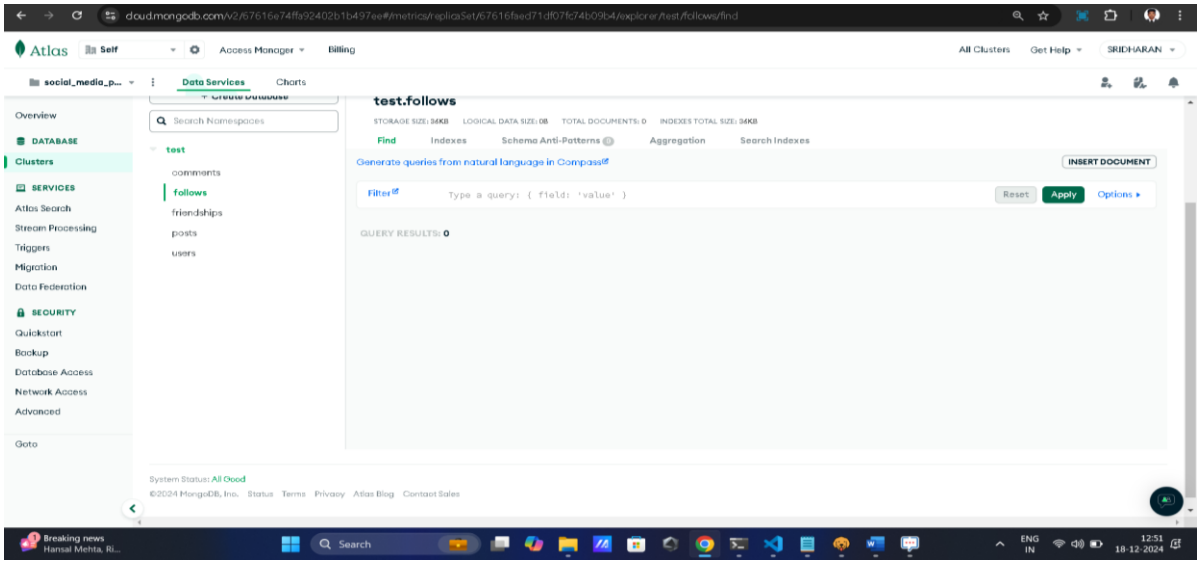
Following means which you follow that user id

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:

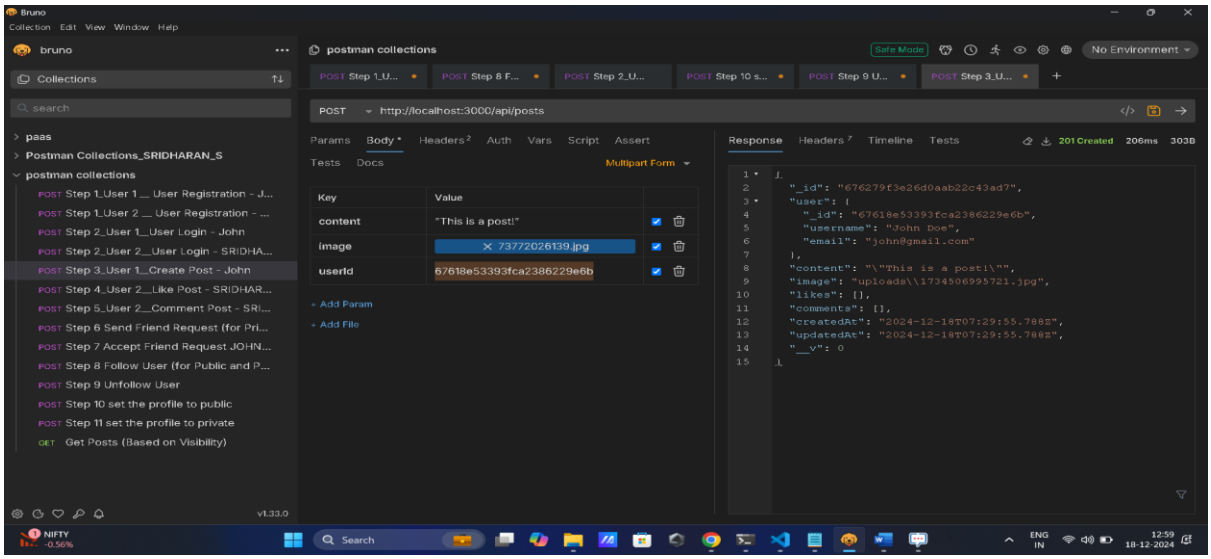
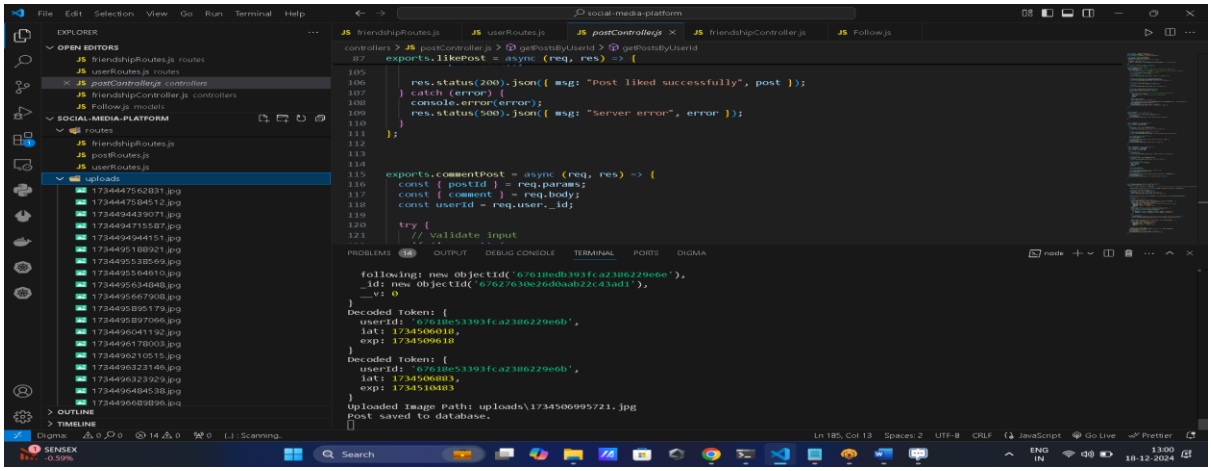
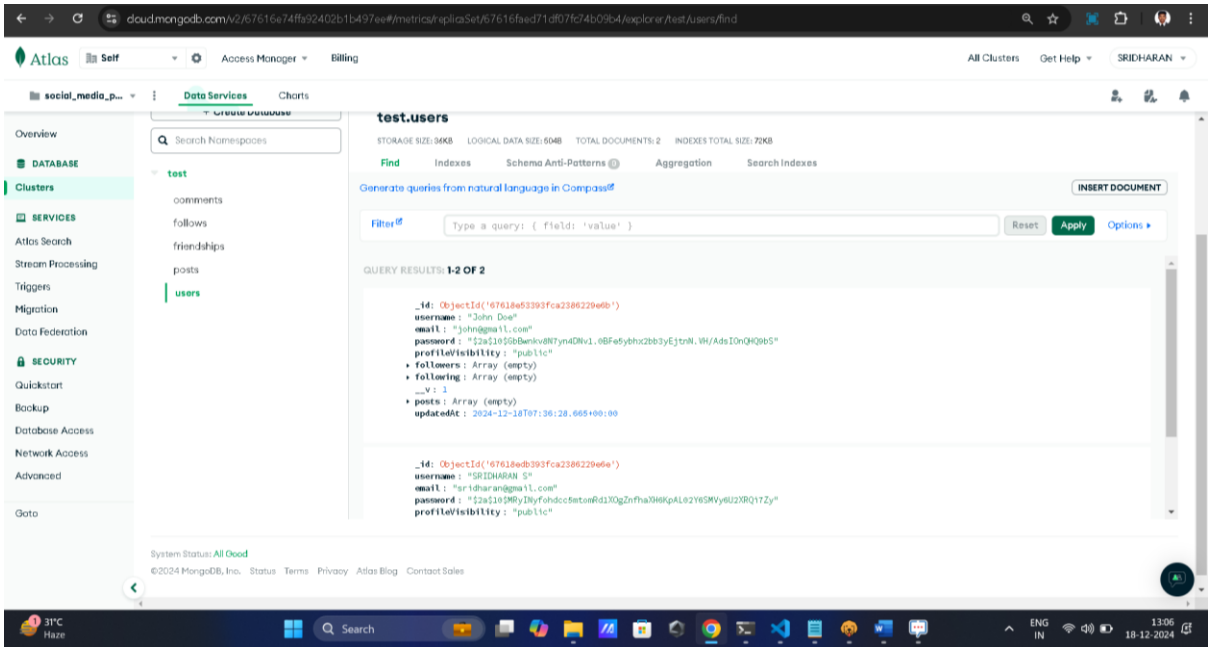
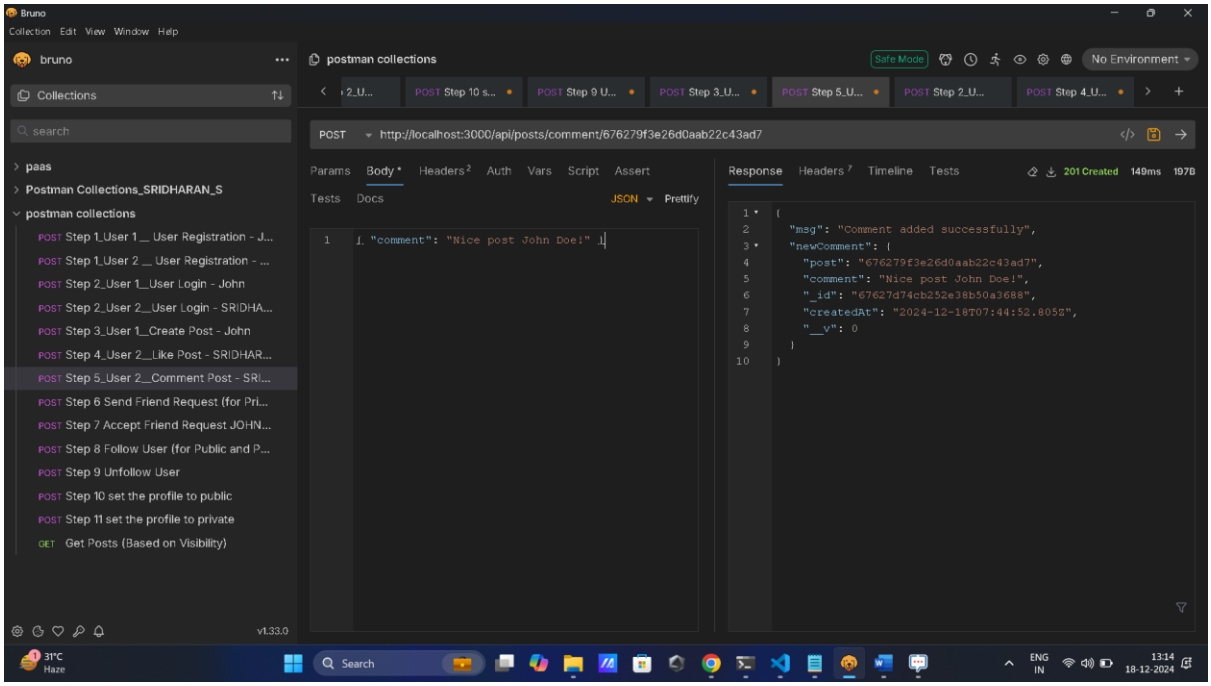


Image upload - Uploads folder

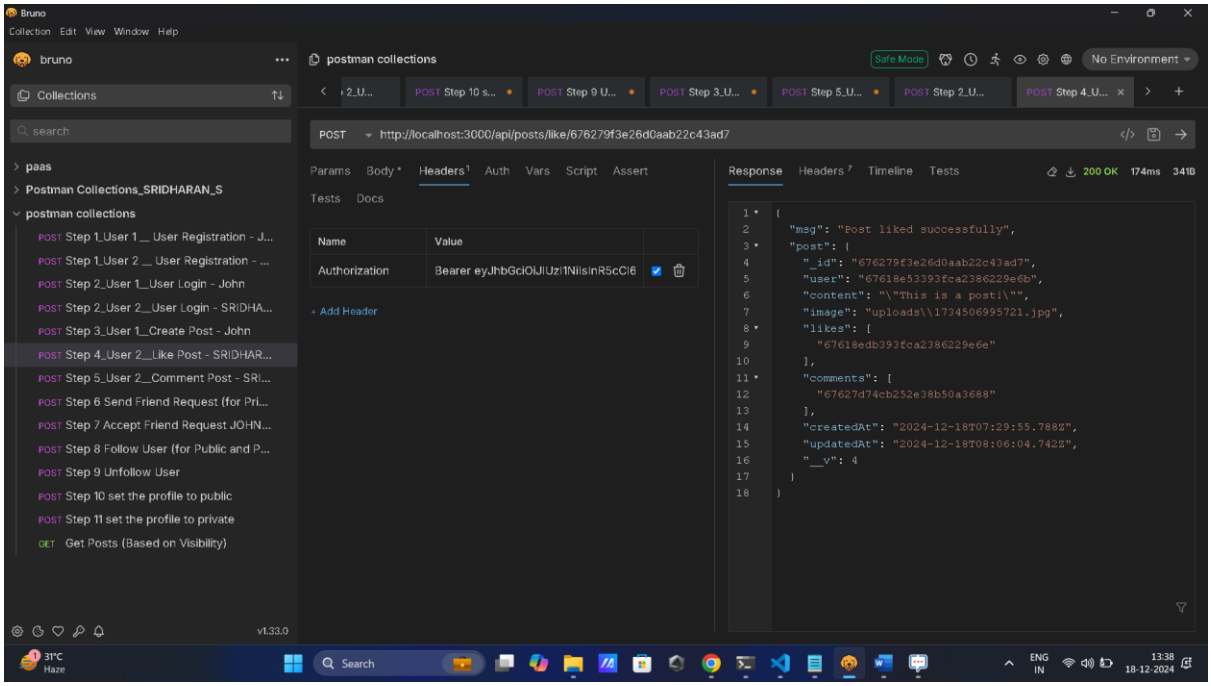


- Comment & Likes
- Example Screenshot Comments

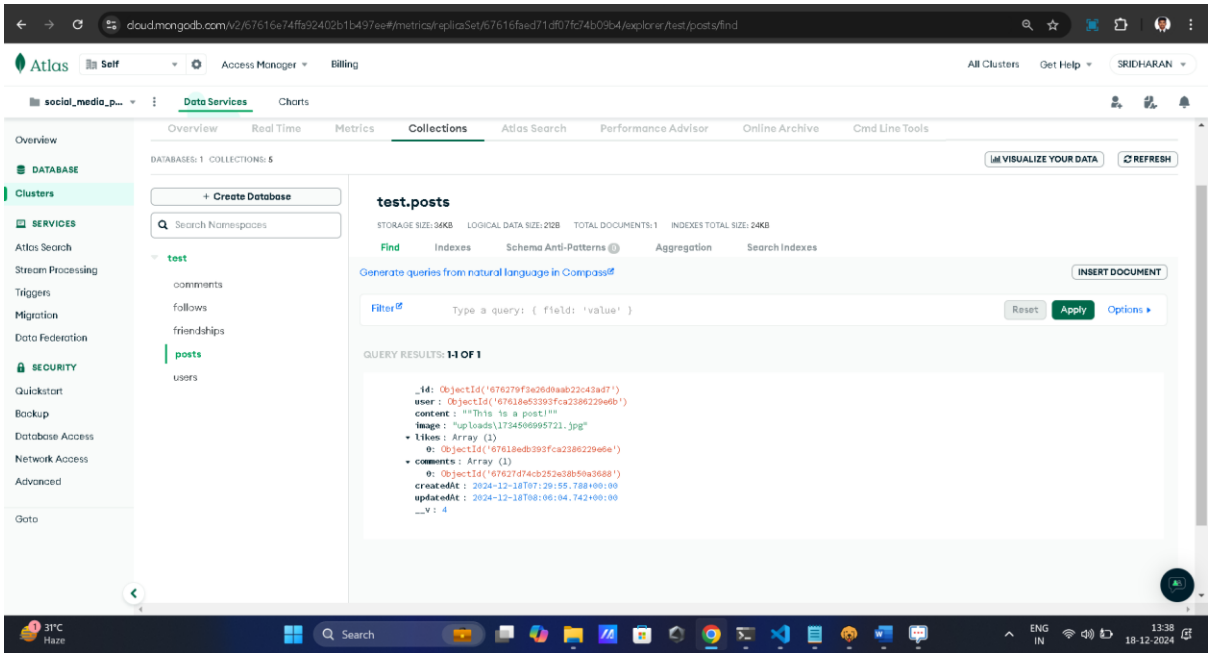




- Example Screenshot Likes

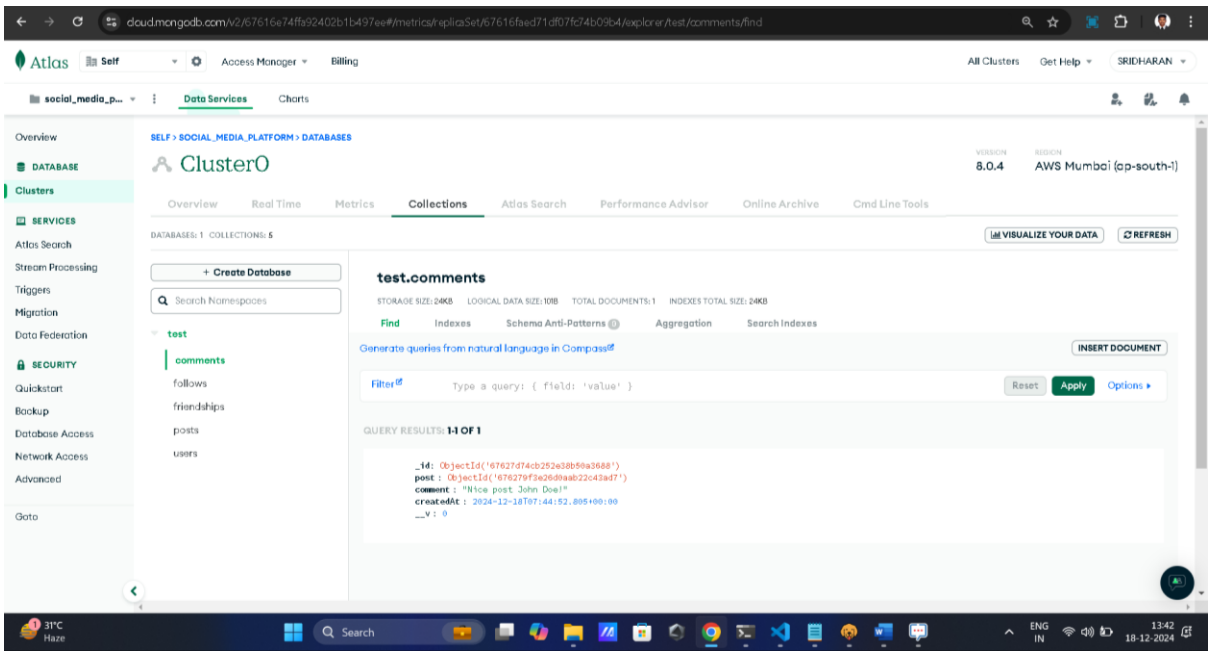


- Check Posts Schema



- Likes : Array (1)
- Comments : Array(1)

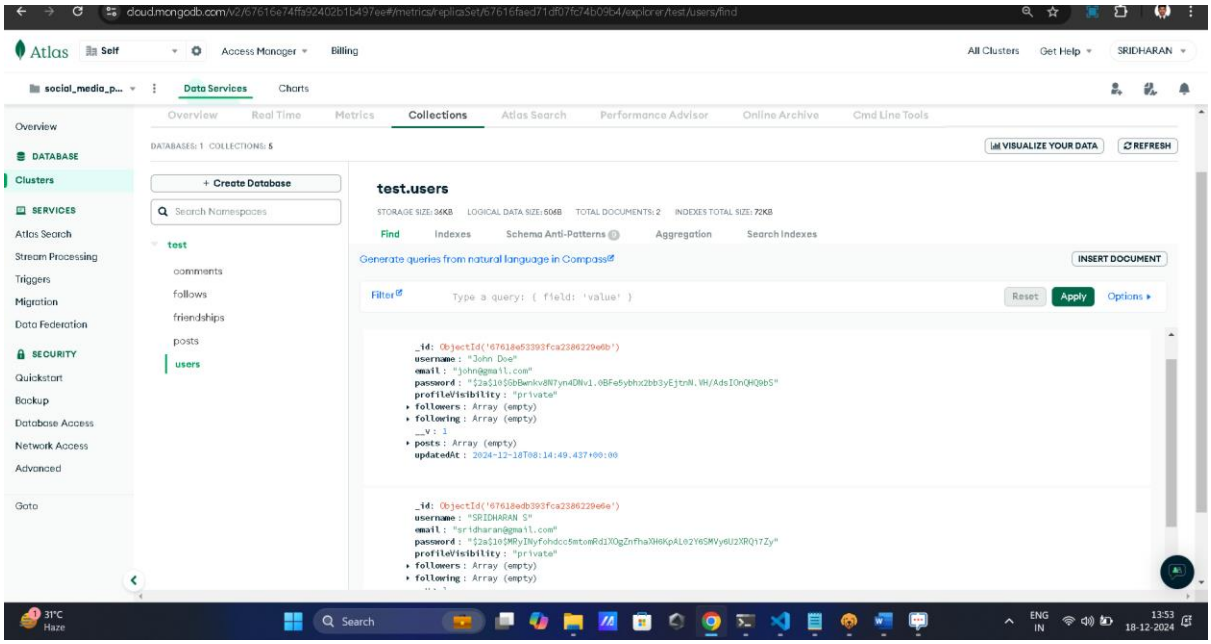
- Check Comments Schema



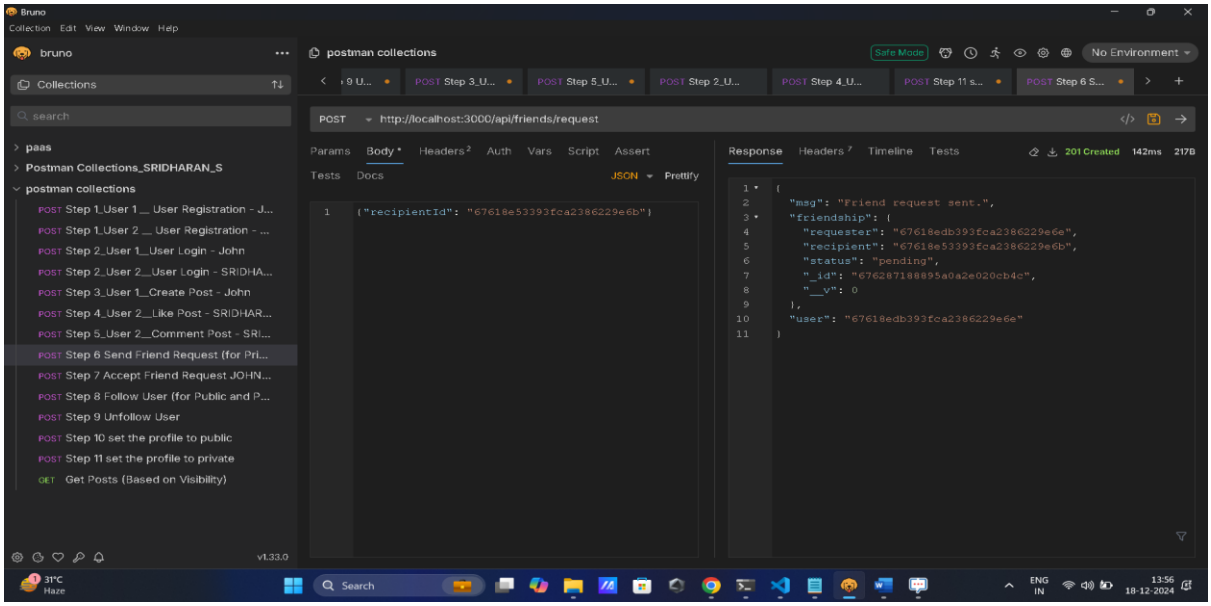
4. Friendship Mechanism

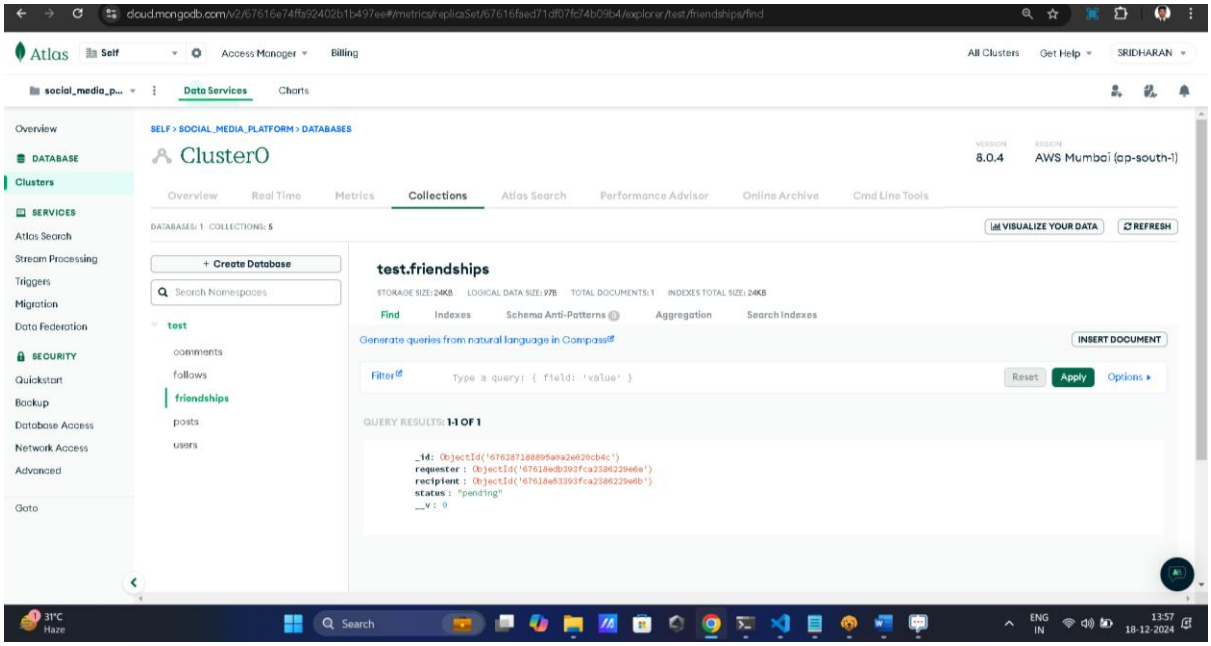
- Users can send, accept friend requests.
- Example:

Screenshot:



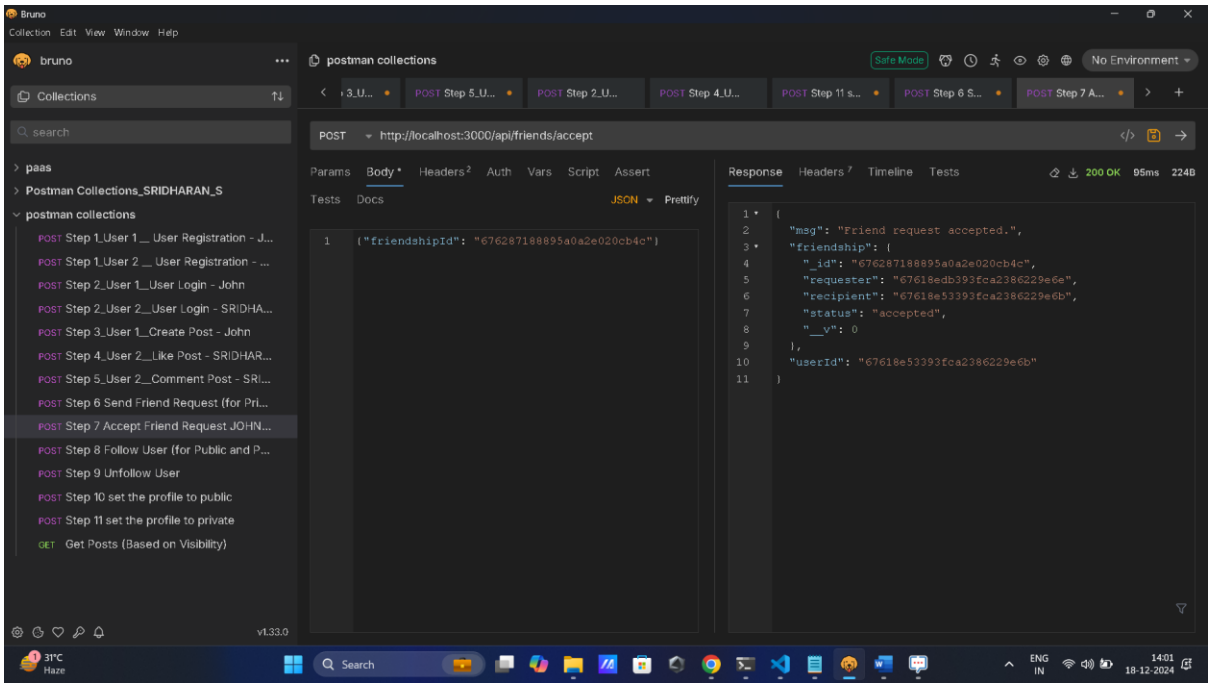
- Send Request

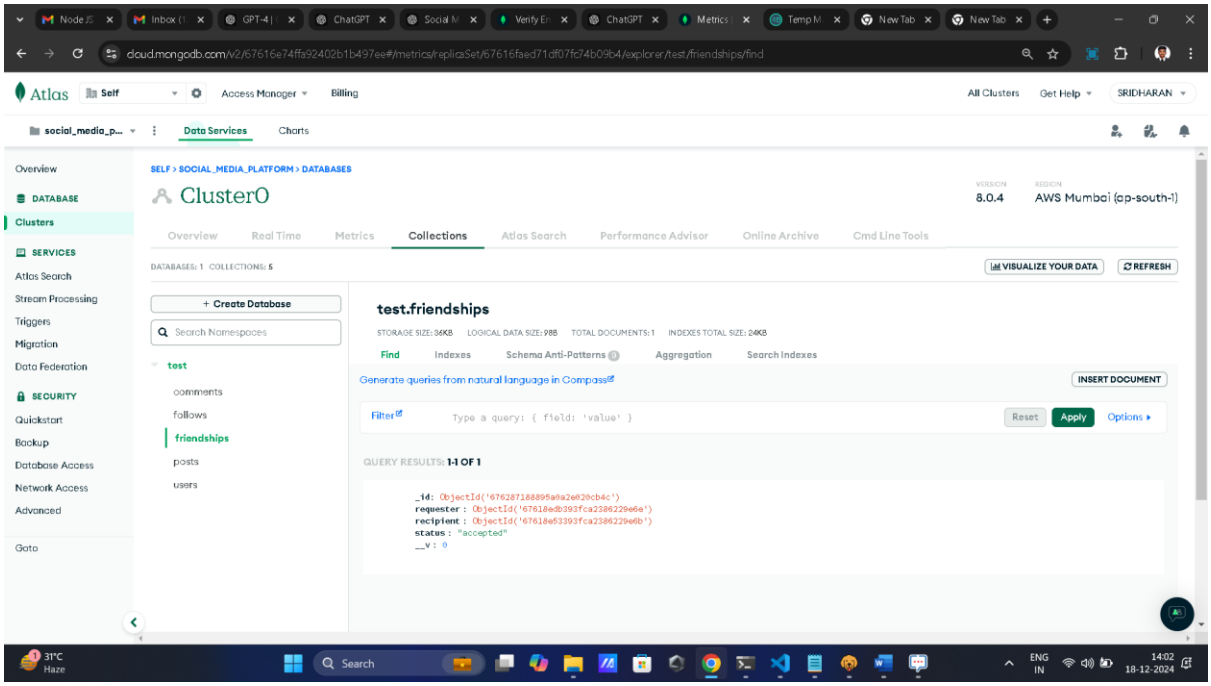




- **Status** : "pending"

- Accept Request

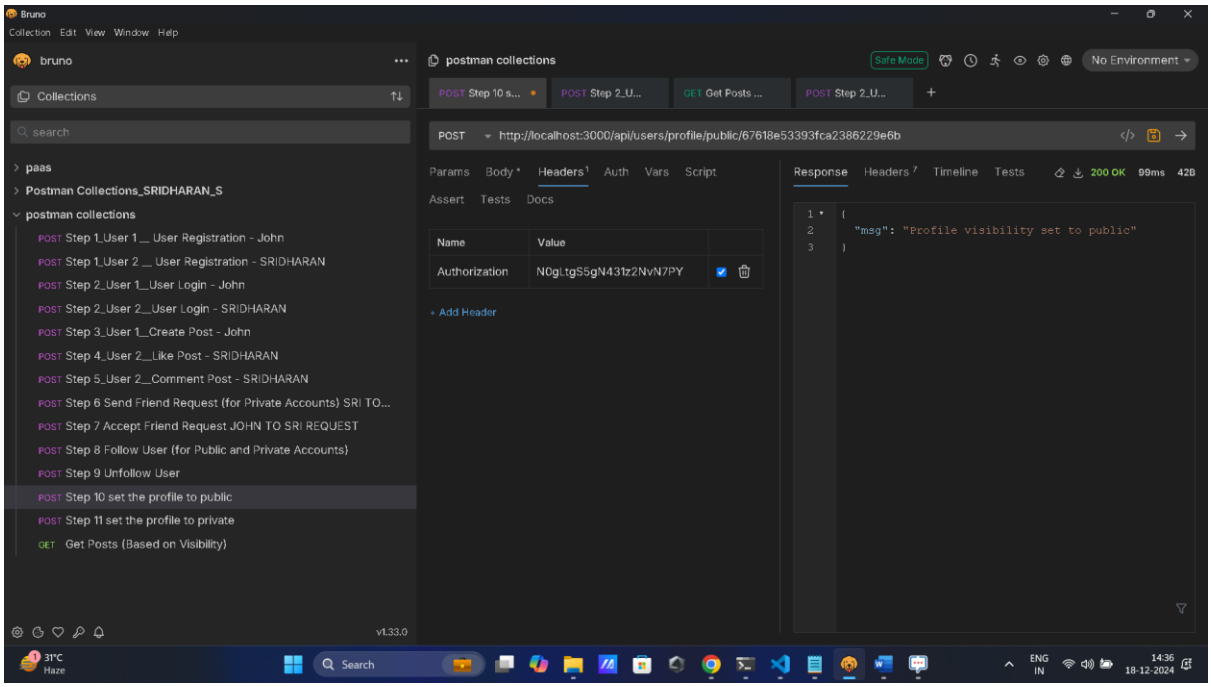


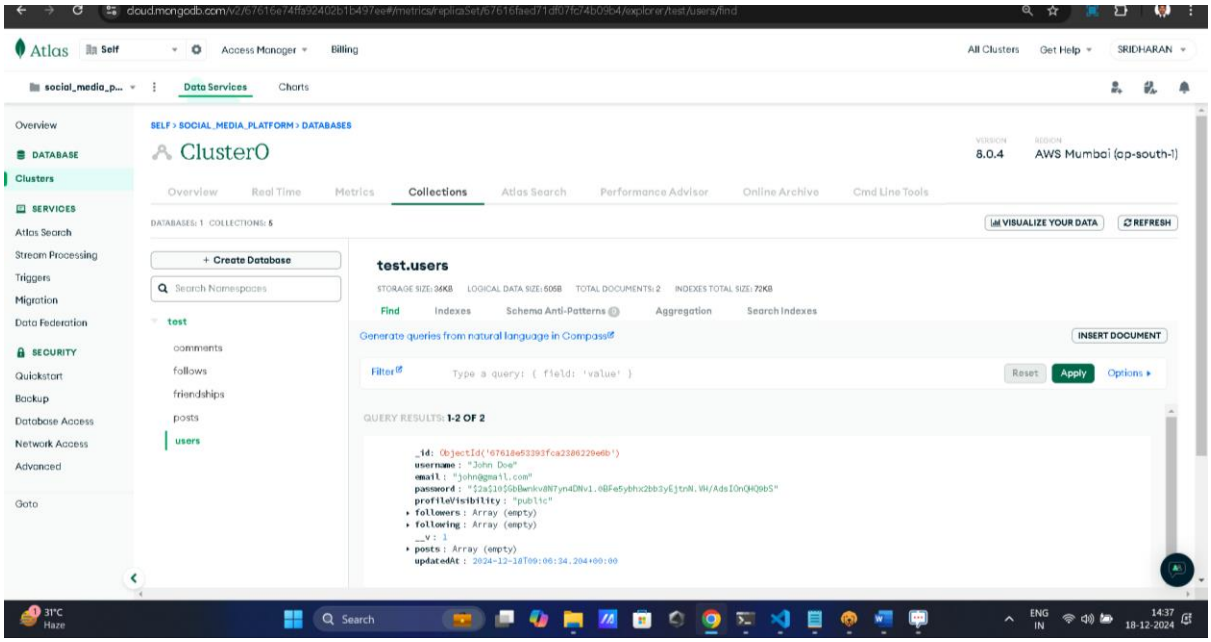


- **Status** : "accepted"

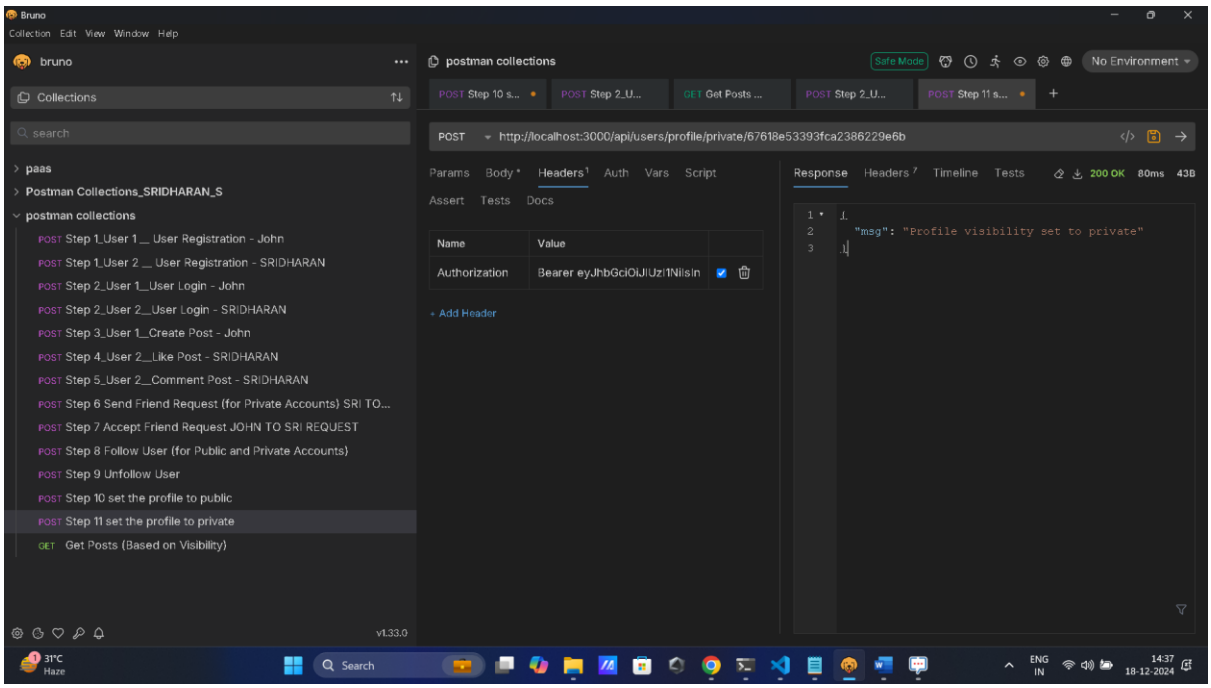
5. Profile Visibility

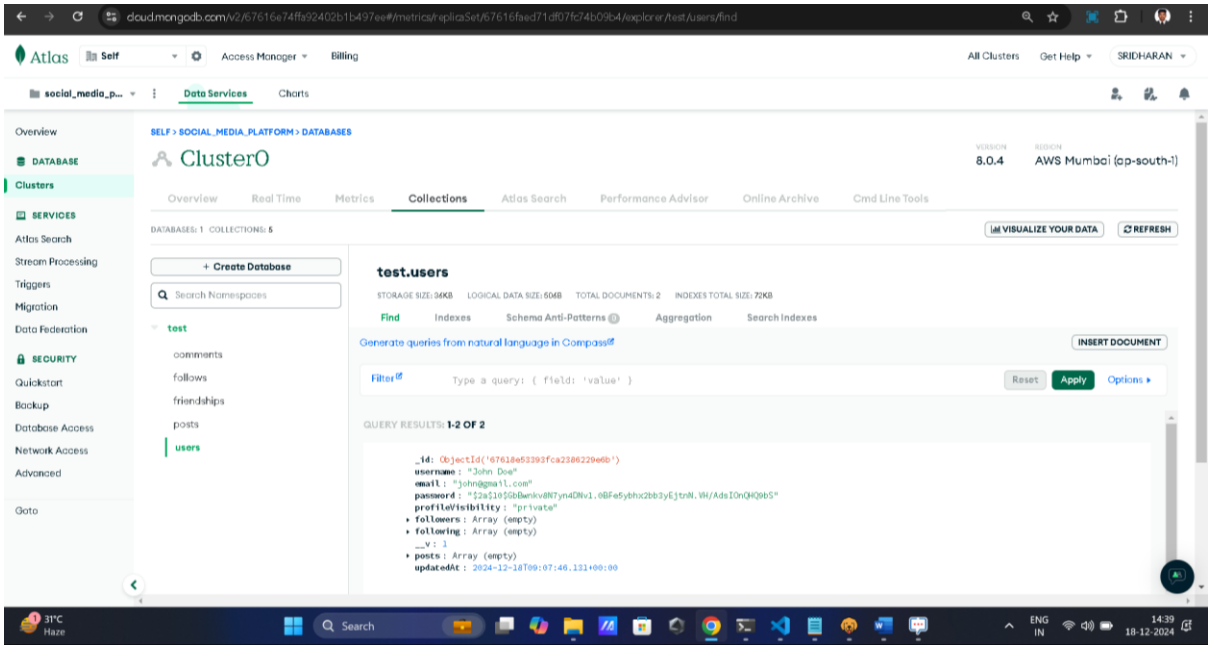
- Two visibility options for user profiles:
- Public


Bruno
Collection Edit View Window Help
bruno
Collections
search
paas
Postman Collections_SRIDHARAN_S
postman collections
POST Step 1_User 1__ User Registration - John
POST Step 1_User 2 __ User Registration - SRIDHARAN
POST Step 2_User 1__User Login - John
POST Step 2_User 2__User Login - SRIDHARAN
POST Step 3_User 1__Create Post - John
POST Step 4_User 2__Like Post - SRIDHARAN
POST Step 5_User 2__Comment Post - SRIDHARAN
POST Step 6 Send Friend Request (for Private Accounts) SRI TO...
POST Step 7 Accept Friend Request JOHN TO SRI REQUEST
POST Step 8 Follow User (for Public and Private Accounts)
POST Step 9 Unfollow User
POST Step 10 set the profile to public
POST Step 11 set the profile to private
GET Get Posts (Based on Visibility)
v1.33.0
postman collections
Safe Mode
No Environment
POST Step 10 s...
POST Step 2_U...
GET Get Posts ...
POST Step 2_U...
POST http://localhost:3000/api/users/profile/public/67618e53393fca2386229e6b
Params Body * Headers 1 Auth Vars Script
Assert Tests Docs
Name Value
Authorization N0gLtgS5gN431z2NvN7PY
+ Add Header
Response Headers 7 Timeline Tests
200 OK 99ms 42B
{
"msg": "Profile visibility set to public"
}
31°C Haze
Search
ENG IN
14:36 18-12-2024

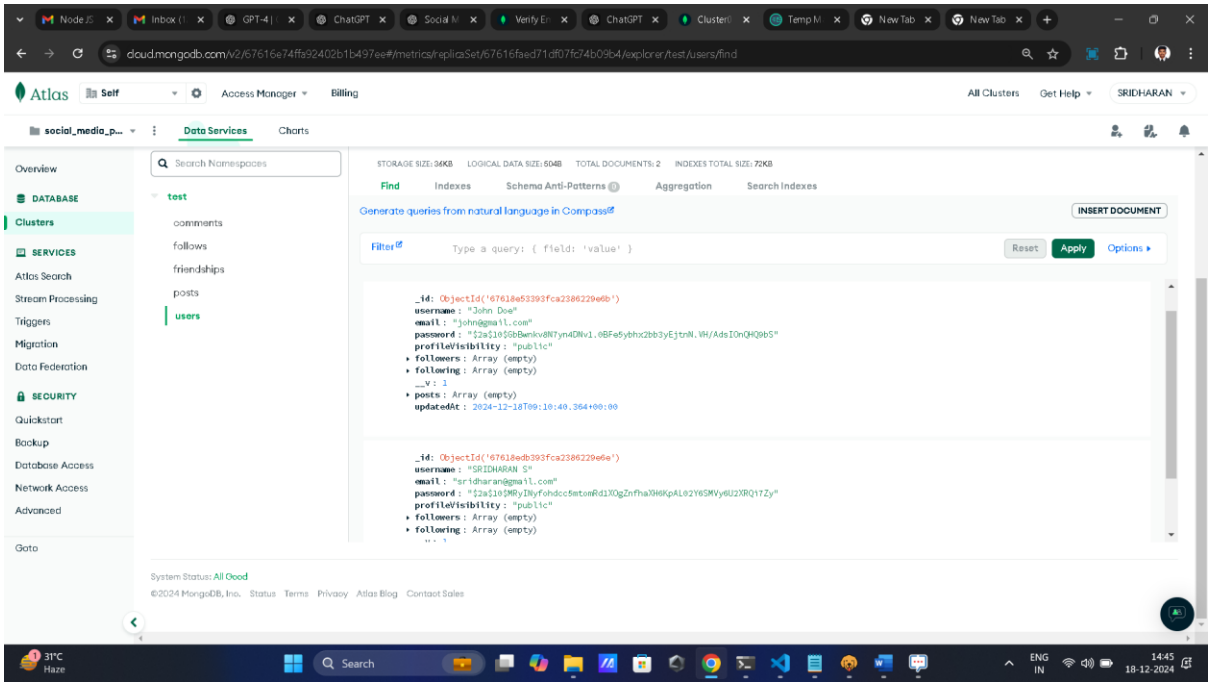

cloud.mongodb.com/v2/67616e74ffa92402b1b497ee#/metrics/replicaSet/67616faed71df07fc74b09b4/explorer/test/users/find
Atlas
Self
Access Manager
Billing
All Clusters
Get Help
SRIDHARAN
social_media_p...
Data Services
Charts
Overview
DATABASE
Clusters
SERVICES
Atlas Search
Stream Processing
Triggers
Migration
Data Federation
SECURITY
Quickstart
Backup
Database Access
Network Access
Advanced
Goto
SELF > SOCIAL_MEDIA_PLATFORM > DATABASES
Cluster0
8.0.4
AWS Mumbai (ap-south-1)
Overview Real Time Metrics Collections Atlas Search Performance Advisor Online Archive Cmd Line Tools
DATABASES: 1 COLLECTIONS: 5
VISUALIZE YOUR DATA
REFRESH
+ Create Database
Search Namespaces
test
comments
follows
friendships
posts
users
test.users
STORAGE SIZE: 36KB LOGICAL DATA SIZE: 606B TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 72KB
Find Indexes Schema Anti-Patterns Aggregation Search Indexes
Generate queries from natural language in Compass
INSERT DOCUMENT
Filter
Type a query: { field: 'value' }
Reset
Apply
Options
QUERY RESULTS: 1-2 OF 2
_id: ObjectId('67618e53393fca2386229e6b')
username : "John Doe"
email : "john@gmail.com"
password : "$2a$10$GbBwnkv8N7yn4DNv1.0BFe5ybhx2bb3yEjtnN.VH/AdsIOnQHQbbS"
profileVisibility : "public"
followers : Array (empty)
following : Array (empty)
__v: 1
posts : Array (empty)
updatedAt : 2024-12-18T09:06:34.204+00:00
31°C Haze
Search
ENG IN
14:37 18-12-2024

- Private

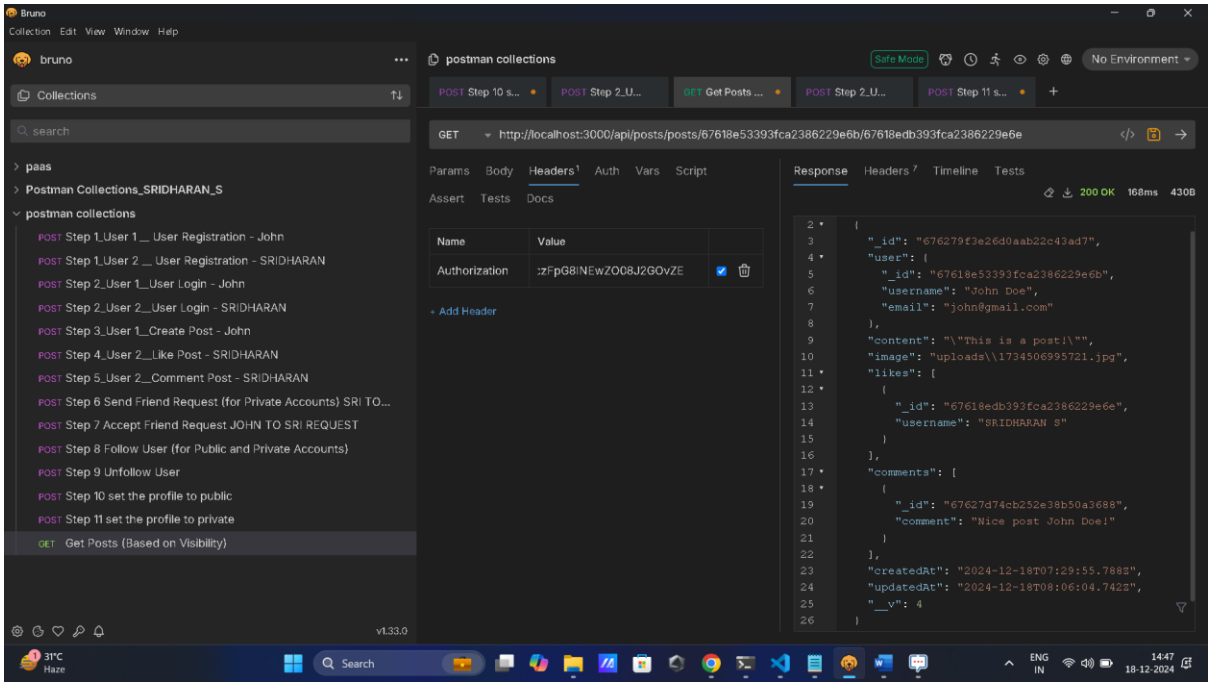

Bruno
Collection Edit View Window Help
bruno
Collections
search
paas
Postman Collections_SRIDHARAN_S
postman collections
POST Step 1_User 1__ User Registration - John
POST Step 1_User 2__ User Registration - SRIDHARAN
POST Step 2_User 1__User Login - John
POST Step 2_User 2__User Login - SRIDHARAN
POST Step 3_User 1__Create Post - John
POST Step 4_User 2__Like Post - SRIDHARAN
POST Step 5_User 2__Comment Post - SRIDHARAN
POST Step 6 Send Friend Request (for Private Accounts) SRI TO...
POST Step 7 Accept Friend Request JOHN TO SRI REQUEST
POST Step 8 Follow User (for Public and Private Accounts)
POST Step 9 Unfollow User
POST Step 10 set the profile to public
POST Step 11 set the profile to private
GET Get Posts (Based on Visibility)
v1.33.0
postman collections
Safe Mode
No Environment
POST Step 10 s...
POST Step 2_U...
GET Get Posts ...
POST Step 2_U...
POST Step 11 s...
POST http://localhost:3000/api/users/profile/private/67618e53393fca2386229e6b
Params Body Headers Auth Vars Script
Assert Tests Docs
Name Value
Authorization Bearer eyJhbGciOiJIUzI1NiIsIn
+ Add Header
Response Headers Timeline Tests
200 OK 80ms 43B
{
"msg": "Profile visibility set to private"
}
31°C Haze
Search
ENG IN
14:37 18-12-2024


cloud.mongodb.com/v2/67616e74ffa92402b1b497ee#/metrics/replicaSet/67616faed71df07fc74b09b4/explorer/test.users/find
Atlas
Self
Access Manager
Billing
All Clusters
Get Help
SRIDHARAN
social_media_p...
Data Services
Charts
Overview
DATABASE
Clusters
SERVICES
Atlas Search
Stream Processing
Triggers
Migration
Data Federation
SECURITY
Quickstart
Backup
Database Access
Network Access
Advanced
Goto
SELF > SOCIAL_MEDIA_PLATFORM > DATABASES
Cluster0
8.0.4
AWS Mumbai (ap-south-1)
Overview Real Time Metrics Collections Atlas Search Performance Advisor Online Archive Cmd Line Tools
DATABASES: 1 COLLECTIONS: 5
VISUALIZE YOUR DATA
REFRESH
+ Create Database
Search Namespaces
test
comments
follows
friendships
posts
users
test.users
STORAGE SIZE: 36KB LOGICAL DATA SIZE: 6.06B TOTAL DOCUMENTS: 2 INDEXES TOTAL SIZE: 72KB
Find Indexes Schema Anti-Patterns Aggregation Search Indexes
Generate queries from natural language in Compass
INSERT DOCUMENT
Filter
Type a query: { field: 'value' }
Reset Apply Options
QUERY RESULTS: 1-2 OF 2
_id: ObjectId('67618e53393fca2386229e6b')
username : "John Doe"
email : "john@gmail.com"
password : "$2a$10$GbBwnkv0N7yn4DNv1.0BFe5ybhx2bb3yEjtnN.WH/AdsIOnQHQobS"
profileVisibility : "private"
followers : Array (empty)
following : Array (empty)
__v : 1
posts : Array (empty)
updatedAt : 2024-12-18T09:07:46.131+00:00
31°C Haze
Search
ENG IN
14:39 18-12-2024

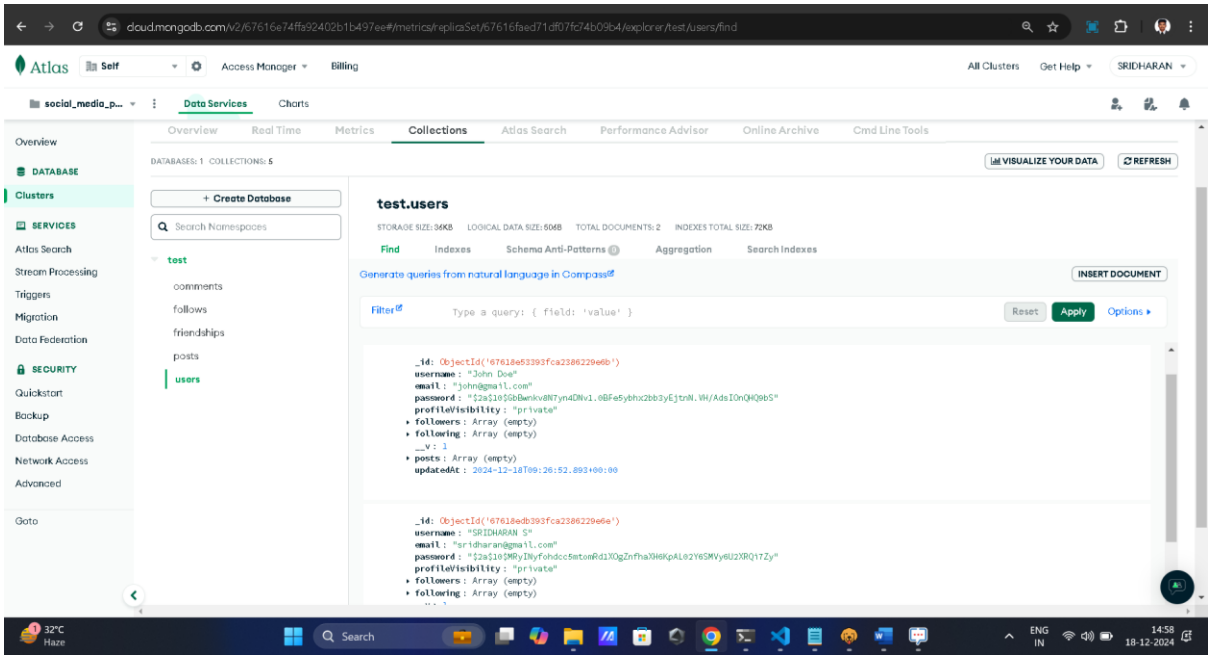
- **Public**: Allows all followers to view posts.

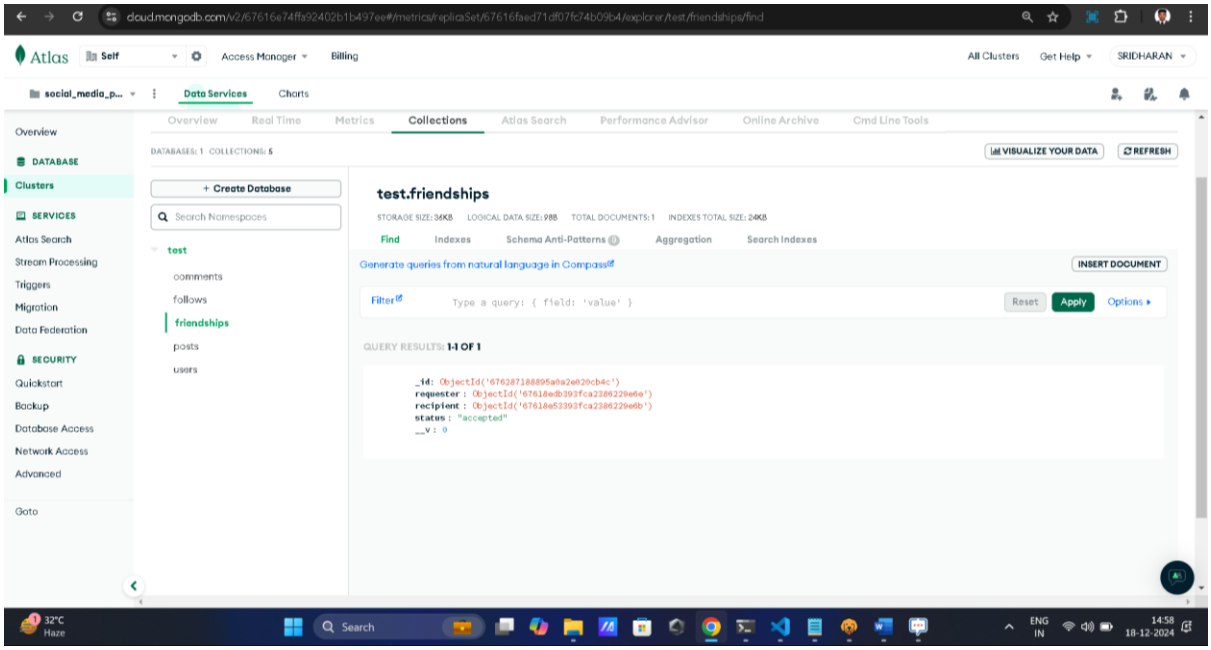


John Already Post Pictures Sridharan To View John Posts, Comments, Likes

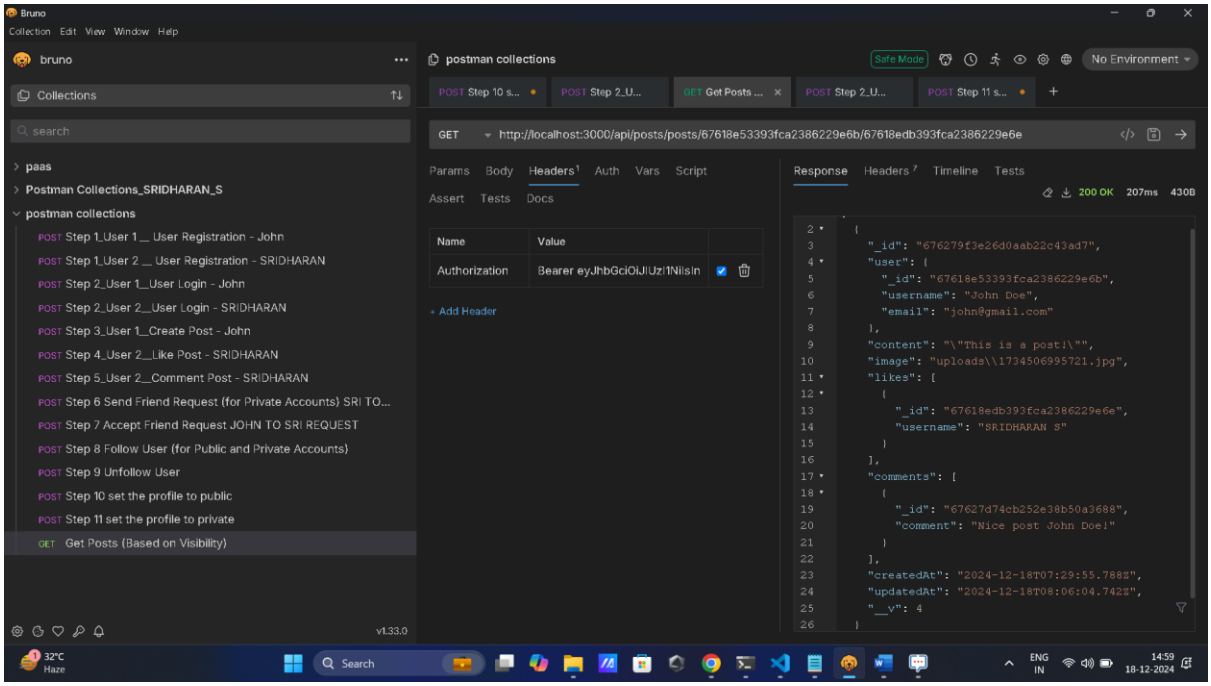


- **Private**: Only friends can view posts.

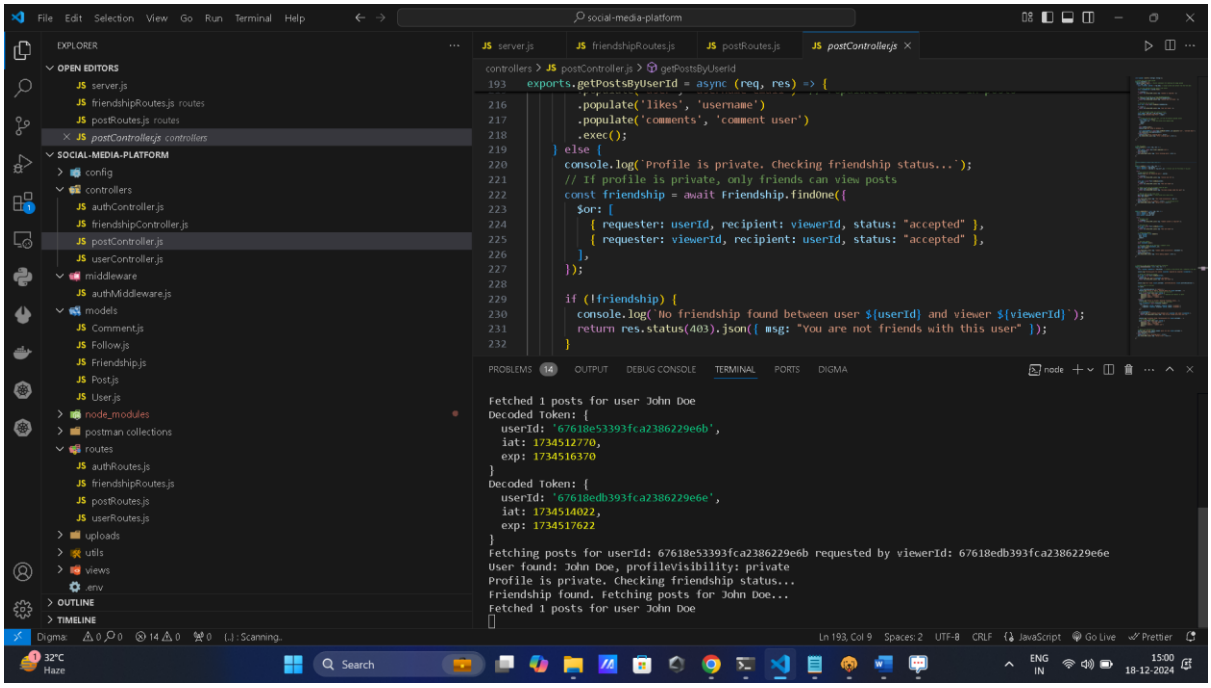




“status” : “accepted”



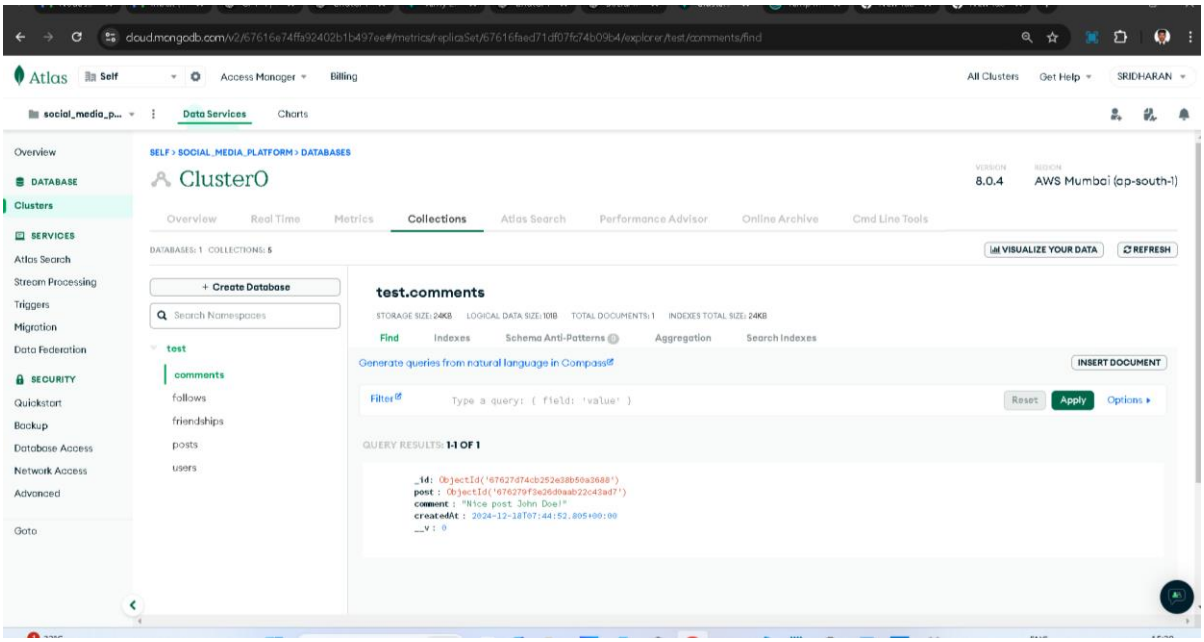
Vs code Console Log:



Technology Stack

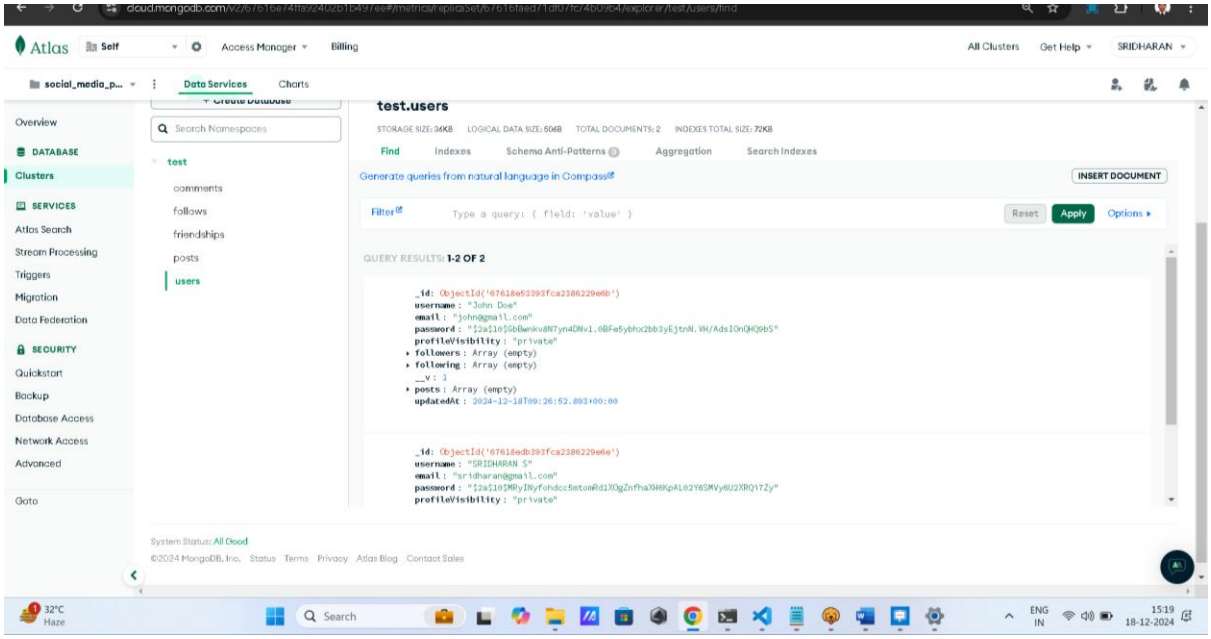
- **Backend**: Node.js, Express.js
- **Database**: MongoDB (or PostgreSQL)
- **Authentication**: JSON Web Token (JWT)
- **File Uploads**: Multer
- **Password Hashing**: Bcrypt

Database Schema (MongoDB Example)



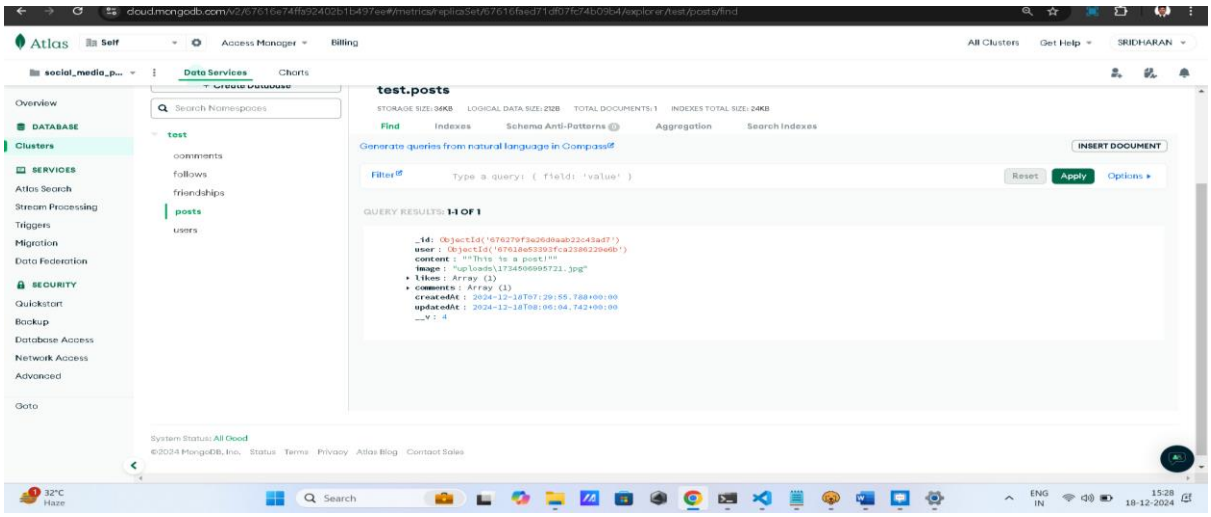
User Table

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.



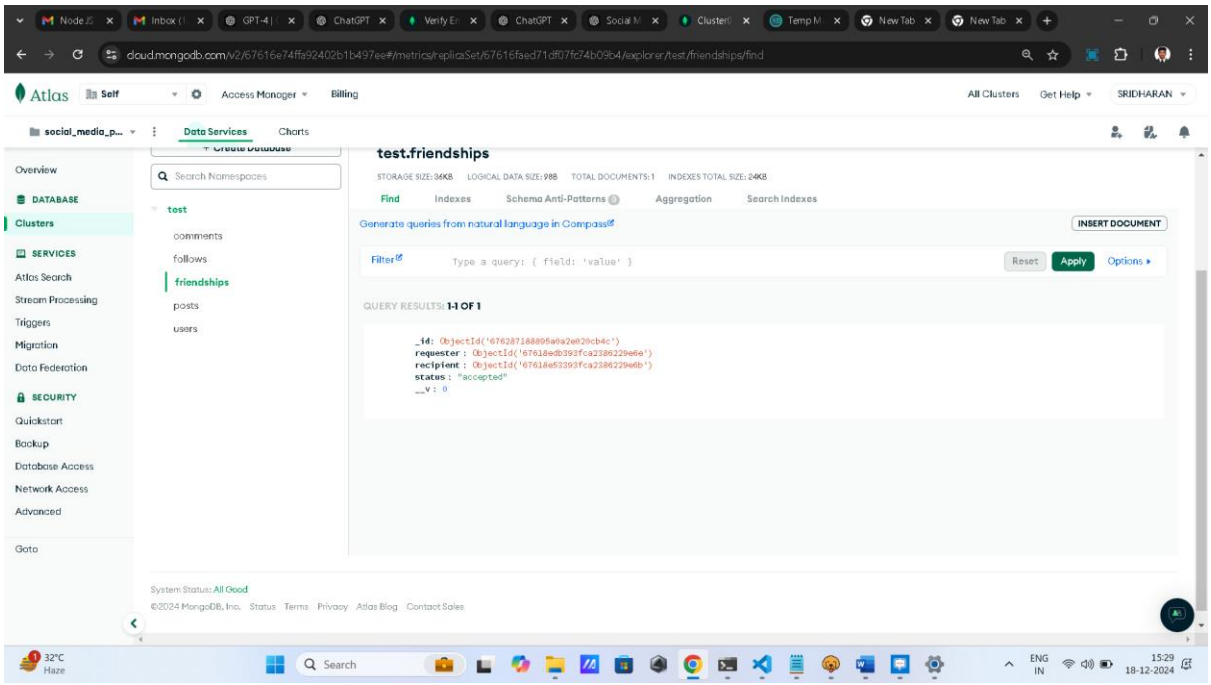
Post Table

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

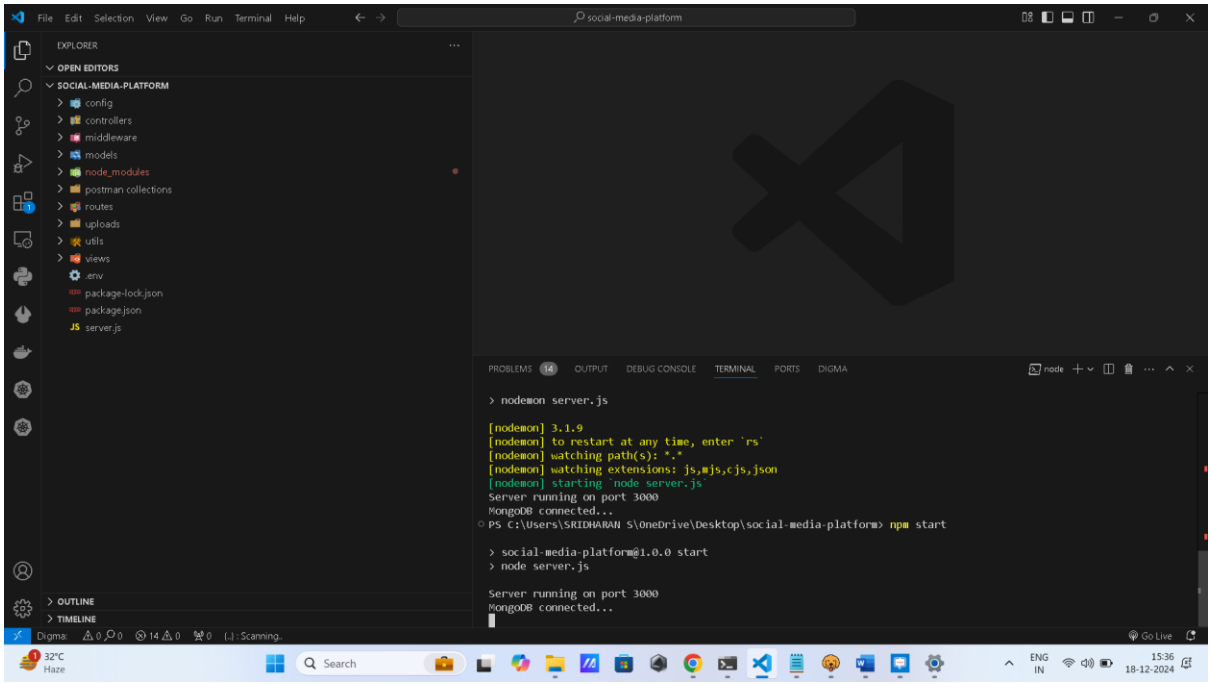
1. Clone the repository:

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

   cd social-media-platform

2. Install dependencies:
   npm install
3. Configure environment variables in a .env file:
   PORT=3000
   MONGO_URI=<your-mongodb-connection-string>
   JWT_SECRET=<your-secret-key>
4. Start the server:
   npm start
5. 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.