**Module:**

**STW304CEM-Web API Development**

**Project Title:  
Momento (social media)**

**Intake:**

SEPTEMBER/NOVEMBER 2019

**Submitted By:**

CU ID: 10173183

Name: Gauri Shankar Sharma

**Submitted To:**

Niman Maharjan

**Softwarica College in collaboration with**

**Coventry University**

Assessment Submission and Declaration Form

PLEASE COMPLETE SECTIONS IN BLOCK CAPITALS

|  |  |  |  |
| --- | --- | --- | --- |
| **Group work**  If group work ALL student names and IDs must be added below- on behalf of all members; Name................................................... ID......................  Name................................................... ID......................  Name................................................... ID......................  Name................................................... ID......................  Name................................................... ID...................... | | **Surname:** SHARMA | |
| **First Name:** GAURI SHANKAR | |
| **Word Count:** ­960 | |
| **Student number (ID):** 10173183 | | **Attempt:**  FIRST: ✓ RESIT: | |
| **Assignment Due Date:** 18/02/2022 | | **Module Code:** STW304CEM | |
| **Program Title:** BSC (HONS) COMPUTING | | | |
| **Module Title:** Web API Development | | | |
| **Name of Supervisor or Tutor (if applicable):**  Niman Maharjan | | **Individual Work:**  ✓ | **Group Work:** |
| **Assessment Title and Type (i.e., essay, journal, CD,**  **Dissertation)** | | Coursework 1 |  |
| *I have read the Softwarica College rules and regulations on the submission of academic work and in particular the sections concerning misconduct in assessment, including plagiarism, collusion, and cheating. I certify that this assignment is the result of my own (or group) work contains no unreferenced material from another source and does not contravene any part of the College’s rules and regulations.*  *I acknowledge that in submitting this work I am declaring that I (or my group) are fit to be assessed and that a deferral may not be requested following hand-in.*  *I confirm that an electronic version of the item to be assessed where appropriate) is available and will be made available to the College by the specified deadline via Moodle.*  *In respect of group assignments, the submission of this work is made on the basis that all group members are jointly and severally responsible for the work presented for assessment and that by handing in this item for assessment, all group members acknowledge and confirm the statements above and that ALL student names and ID numbers for the group are listed.* | | | |
|  | | | |
| **Student(s) Signature:** | **College Stamp:** | | |

**Table of Contents**

[**Table of Figures** 4](#_Toc96019307)

[**Introduction** 1](#_Toc96019308)

[**Why MongoDB** 2](#_Toc96019309)

[**MongoDB Advantages and Disadvantages** 3](#_Toc96019310)

[**Advantages** 3](#_Toc96019311)

[**Disadvantages** 3](#_Toc96019312)

[**Database Connection** 4](#_Toc96019313)

[**Database Structure** 5](#_Toc96019314)

[**Data Types** 6](#_Toc96019315)

[**Schema Screenshots** 7](#_Toc96019316)

[**User** 7](#_Toc96019317)

[**Profile** 8](#_Toc96019318)

[**Address** 9](#_Toc96019319)

[**Post** 10](#_Toc96019320)

[**Follow** 11](#_Toc96019321)

[**Like** 11](#_Toc96019322)

[**Comment** 12](#_Toc96019323)

[**Notification** 13](#_Toc96019324)

[**Report** 14](#_Toc96019325)

[**Relationship Between Models** 15](#_Toc96019326)

[**Database Screenshots** 16](#_Toc96019327)

[**User** 16](#_Toc96019328)

[**Profile** 17](#_Toc96019329)

[**Address** 18](#_Toc96019330)

[**Post** 19](#_Toc96019331)

[**Follow** 20](#_Toc96019332)

[**Like** 21](#_Toc96019333)

[**Comment** 22](#_Toc96019334)

[**Notification** 23](#_Toc96019335)

[**Report** 24](#_Toc96019336)

# **Table of Figures**

[Figure 1: MongoDB Features (PNGITEM, 2022). 1](#_Toc96019089)

[Figure 2: Reasons to use MongoDB (Flair, 2022). 2](#_Toc96019090)

[Figure 3: MongoDB limitations and benefits (Flair, 2022). 3](#_Toc96019091)

[Figure 4: Connecting Node Js with MongoDB using mongoose library 4](#_Toc96019092)

[Figure 5: MongoDB connection code snapshot 4](#_Toc96019093)

[Figure 6: Database structure of Momento web app 5](#_Toc96019094)

[Figure 7: User Schema 7](#_Toc96019095)

[Figure 8: Profile Schema 8](#_Toc96019096)

[Figure 9: Address Schema 9](#_Toc96019097)

[Figure 10: Post Schema 10](#_Toc96019098)

[Figure 11: Follow Schema 11](#_Toc96019099)

[Figure 12: Like Schema 11](#_Toc96019100)

[Figure 13: Comment Schema 12](#_Toc96019101)

[Figure 14: Notification Schema 13](#_Toc96019102)

[Figure 15: Report Schema 14](#_Toc96019103)

[Figure 16: User Collection 16](#_Toc96019104)

[Figure 17: Profile Collection 17](#_Toc96019105)

[Figure 18: Address Collection 18](#_Toc96019106)

[Figure 19: Post Collection 19](#_Toc96019107)

[Figure 20: Follow Collection 20](#_Toc96019108)

[Figure 21: Like Collection 21](#_Toc96019109)

[Figure 22: Comment Collection 22](#_Toc96019110)

[Figure 23: Notification Collection 23](#_Toc96019111)

[Figure 24: Report Collection 24](#_Toc96019112)

# **Introduction**

The database system used for the “Momento” social media web app is MongoDB. It is a document-oriented NoSQL database and is used to store very large amounts of data. It contains collections and documents. Collections are the tables and documents are the rows. It stores data in a key-value format in the documents instead of rows and columns as in the SQL database. A MongoDB database contains collections, collections contain documents, documents contain fields that are in key-value pair and each documents’ content can be different from each other along with its size. In MongoDB, the schema does not need to be defined before creating the documents, the content of the document can be created on the fly. MongoDB can store arrays and complex structured data very easily. A document can have multiple subdocuments and the subdocuments may also contain other subdocuments. It allows having hierarchical relations within the data (Taylor, 2022).



Figure 1: MongoDB Features (PNGITEM, 2022).

# **Why MongoDB**

The main reason behind choosing MongoDB is that it can handle very large volumes of data more easily than the SQL database.

* As it is document-oriented, it becomes more flexible to match real-world requirements and situations.
* It provides a better searching mechanism because it provides searching by regular expressions, fields, and range queries.
* MongoDB fields can be indexed and this improves the searching performance.
* It provides a replication system in which a replica set may contain two or more instances of MongoDB. Each replica set acts as a primary or secondary replica according to the situation. Whenever the primary replica that means the main server which takes the client’s requests and performs the required operation fails, the secondary replica which contains the copy of the data of the primary replica gets activated automatically and becomes the primary replica.
* It provides better load balancing. It uses the sharding concept and can run various serves at the same time balancing the load. It duplicates data to keep the system running even in case of hardware failure (Taylor, 2022).



Figure 2: Reasons to use MongoDB (Flair, 2022).

# **MongoDB Advantages and Disadvantages**

The advantages and disadvantages of the MongoDB database are shown below (K, 2019), (Flair, 2022).

## **Advantages**

* As it is a NoSQL database, the schema for the database does not need to be defined before creating the database. Lots of time is saved.
* It provides better flexibility to store different types of data types.
* High data availability because of its replication system.
* High Scalability.
* Sharding of the data.
* As Mongo does not have any relationships between the data, no complex joining is required.
* Easily scalable.
* Easy to set up and install.
* It is open-source and free to use.
* Arrays and objects can be easily stored because it stores data in JSON format.
* High-performance speed compared to SQL database.
* It has a simple document queries language compared to SQL queries.
* It does not require application objects mapping with the database objects.
* As it uses internal memory, data access is very faster.

## **Disadvantages**

* No support of the transaction.
* The size of a document is limited to 16 MB.
* Uses high memory to store.
* No joining.

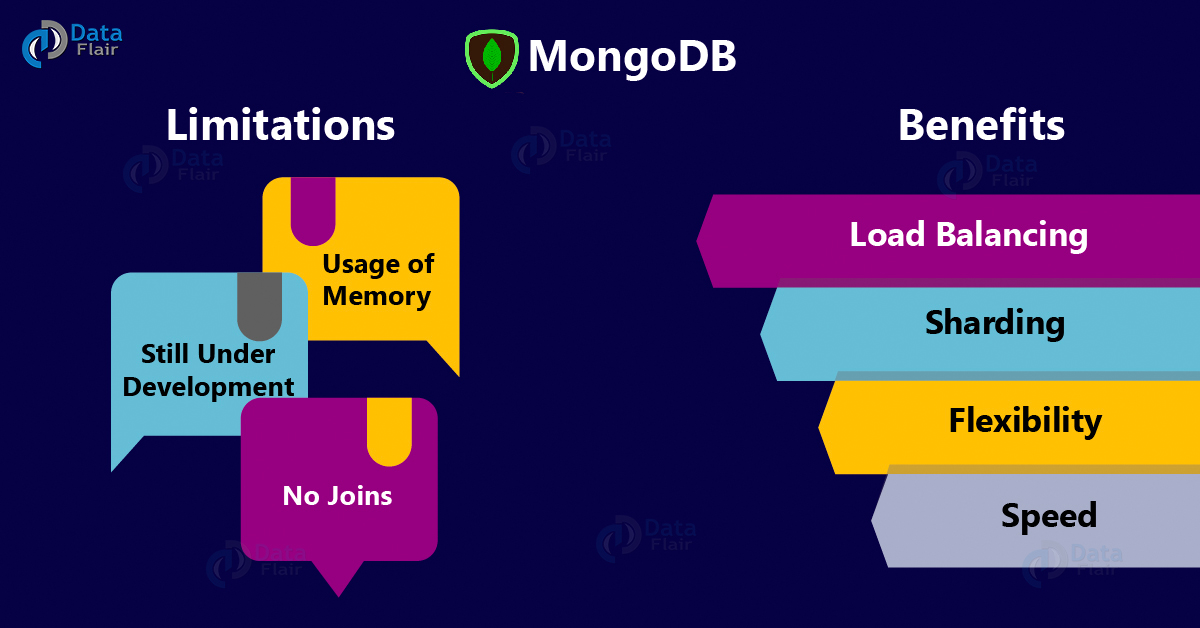


Figure 3: MongoDB limitations and benefits (Flair, 2022).

# **Database Connection**

To connect with the API with the MongoDB database, a Node.js-based library named mongoose has been used. It is an Object Data Modeling library. Mongoose enforces developers to have a specific schema during the development of the application. It also provides different types of model validations, hooks, and other features so that it will be easier to work with MongoDB (Vlaeva, 2021).

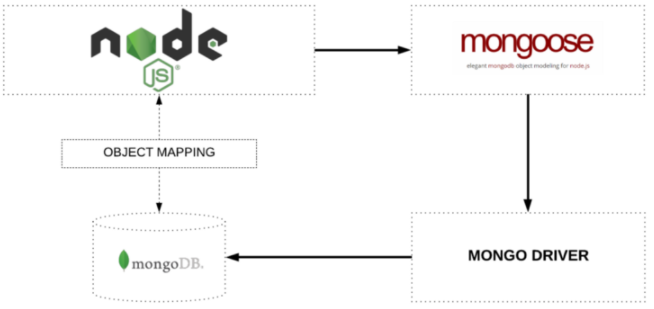


Figure 4: Connecting Node Js with MongoDB using mongoose library



Figure 5: MongoDB connection code snapshot

# **Database Structure**

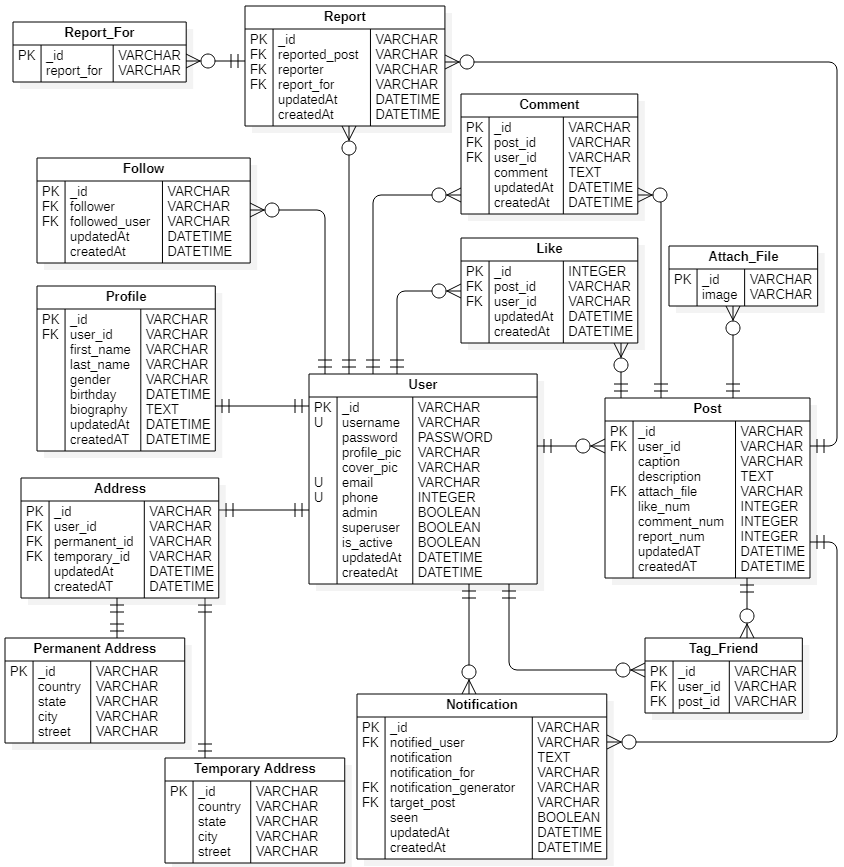


Figure 6: Database structure of Momento web app

From the above diagram, it is clear that a user can have only one profile and address that contains a permanent and a temporary address. Users can have multiple posts and can attach multiple images and tag other users too. A single user can like, comment, and report on multiple posts, can follow multiple users, can have multiple notifications. Similarly, A post can have multiple reports and notifications.

# **Data Types**

While defining the schemas for the database of the web app, various data types have been used to store the data which are shown below.

* String
* Double
* Integer
* Boolean
* Date
* Timestamp
* Object ID
* Arrays
* Null
* Object

Among the above data types, String, Timestamp, Object ID, Boolean data types are highly used datatypes.

# **Schema Screenshots**

## **User**

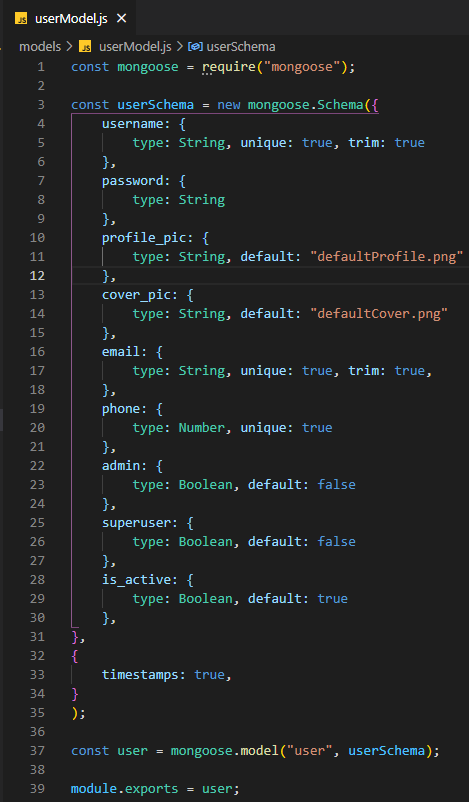


Figure 7: User Schema

## **Profile**

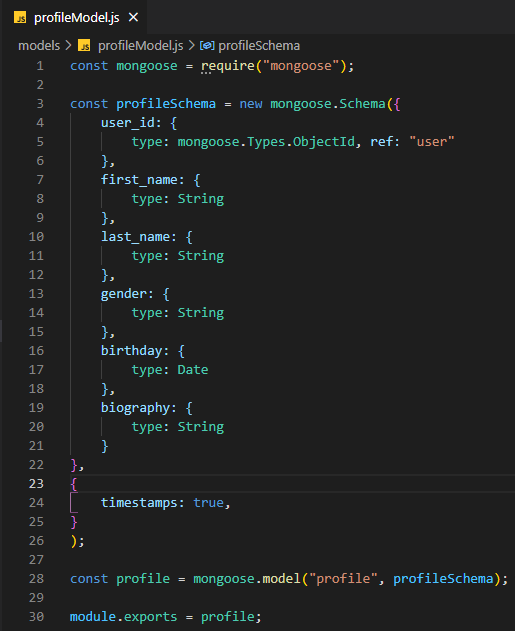


Figure 8: Profile Schema

## **Address**



Figure 9: Address Schema

## **Post**

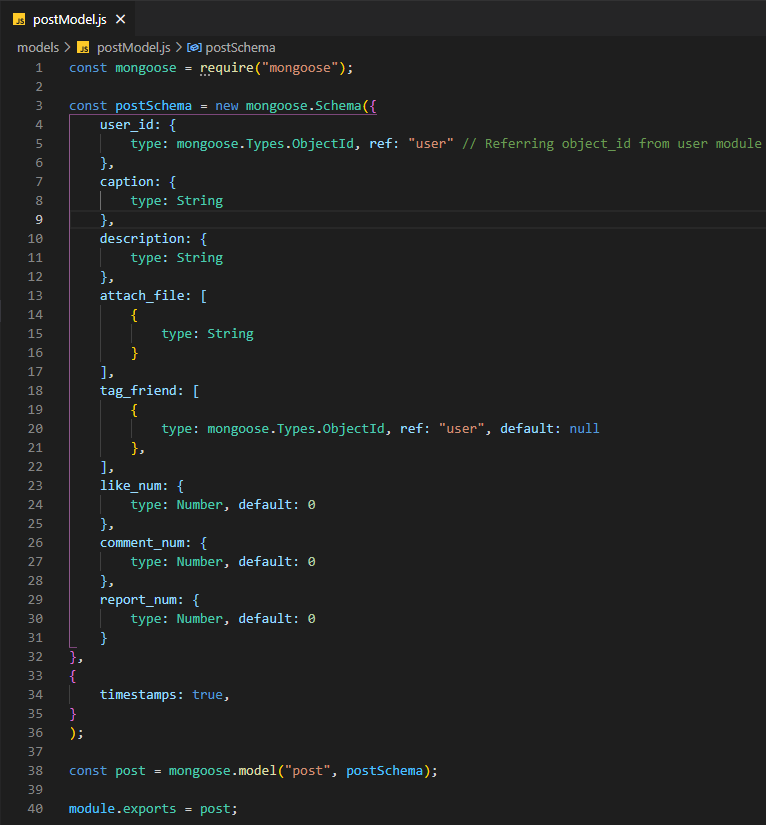


Figure 10: Post Schema

## **Follow**

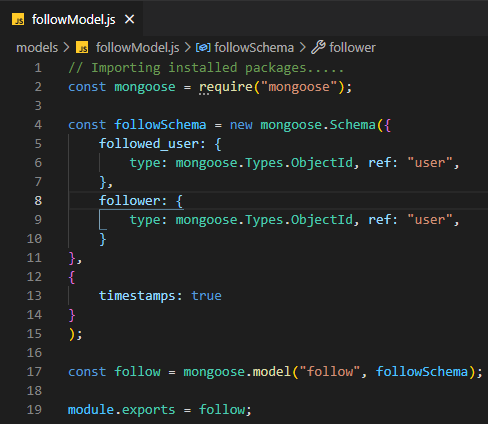


Figure 11: Follow Schema

## **Like**

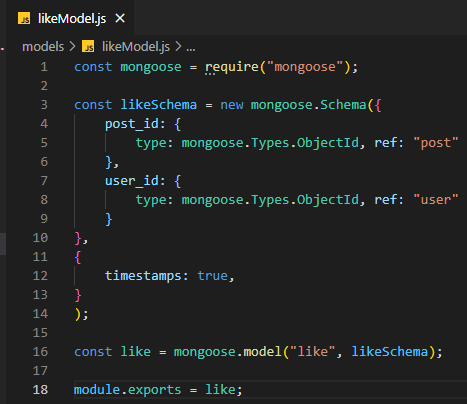


Figure 12: Like Schema

## **Comment**

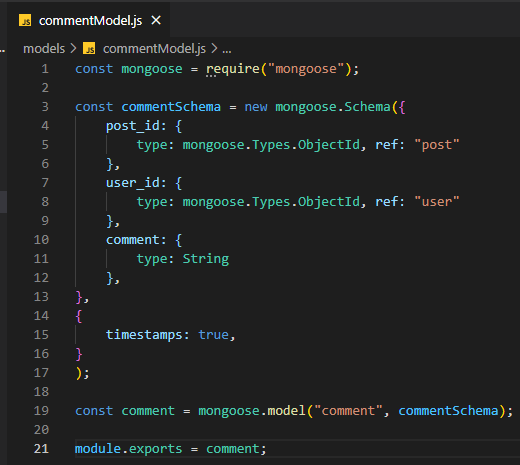


Figure 13: Comment Schema

## **Notification**

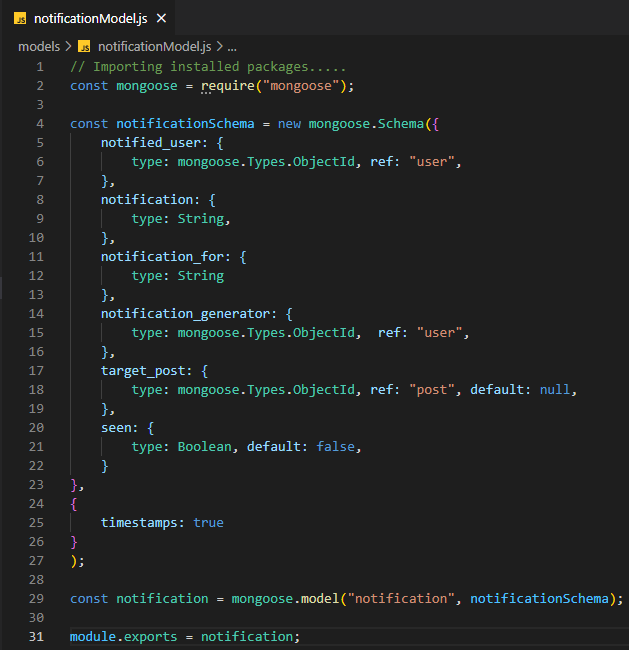


Figure 14: Notification Schema

## **Report**

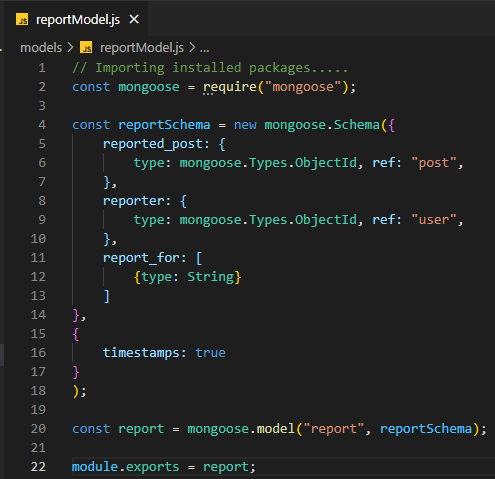


Figure 15: Report Schema

# **Relationship Between Models**

There is a total of nine models that have been used to define the collections in MongoDB for the web app. The user model is the main model and this model has been referenced in all other models. Without a user model, other models cannot exist. Therefore, whenever new user signup the web, the user model gets created at first, only then other models will be created according to the user’s activity. The user model is referenced as user id in the profile and address model. A user can have only one profile and address model. Users can create multiple posts models. In a post model, multiple images can be attached and multiple users can be tagged. In the follow model, the user model is referenced in both follower and followed user fields. The follow model will be created whenever a user follows another user. A user can tag only his/her followers while uploading a post. In the notification model, the user model is referenced in notified user and notification generator field, post model is referenced in the target post field. Whenever a user uploads a post or follows another user, or likes, comments, reports on a post, then the notification model is created. User and post models are referenced in like and comment models as user id and post id. In the report model, the user id is referenced as reporter and post id is referenced as reported post.

# **Database Screenshots**

## **User**

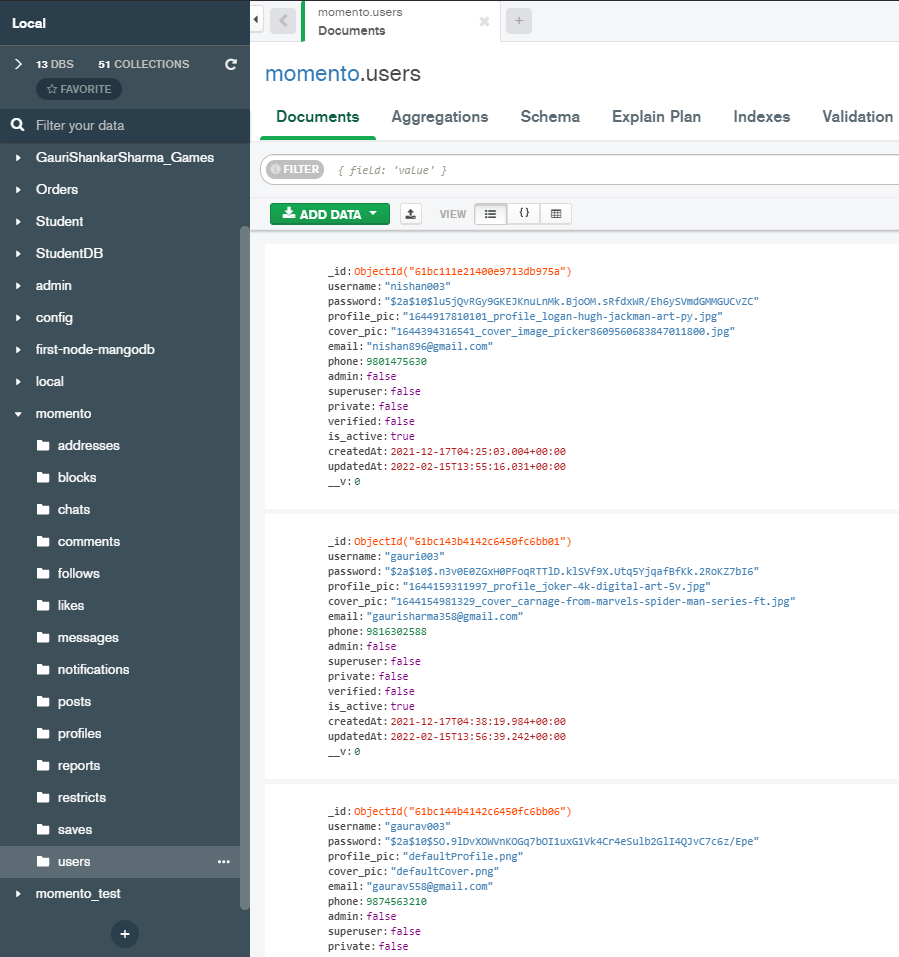


Figure 16: User Collection

## **Profile**

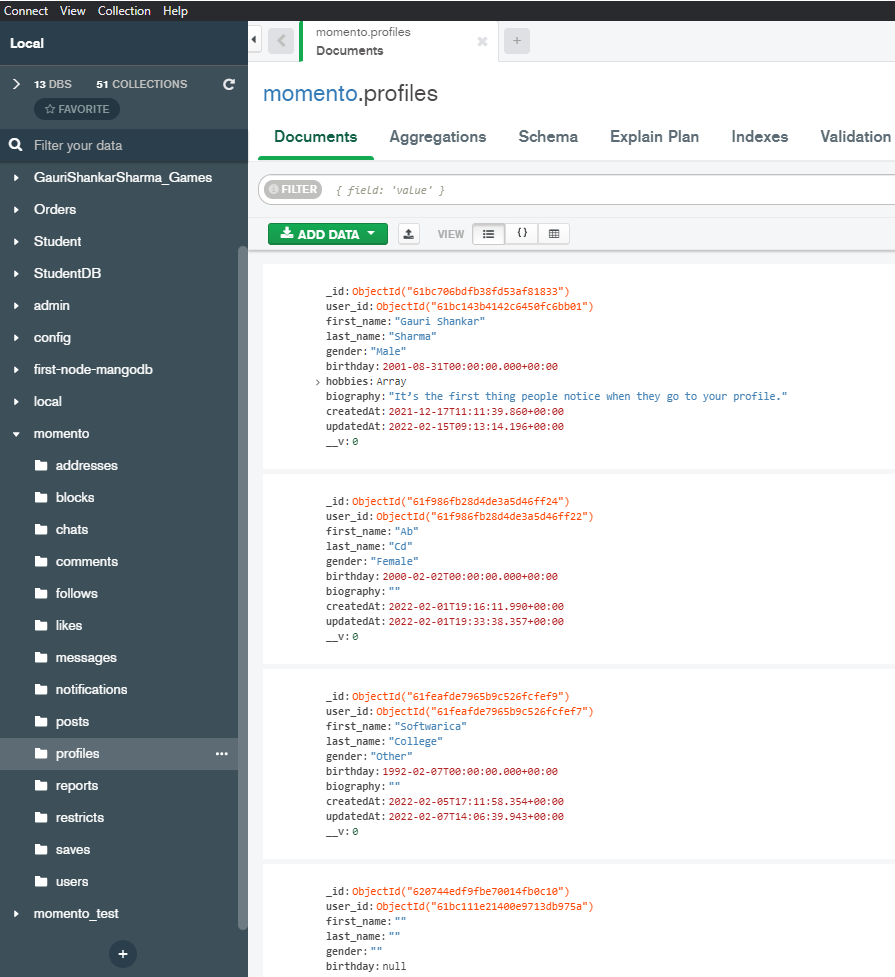


Figure 17: Profile Collection

## **Address**

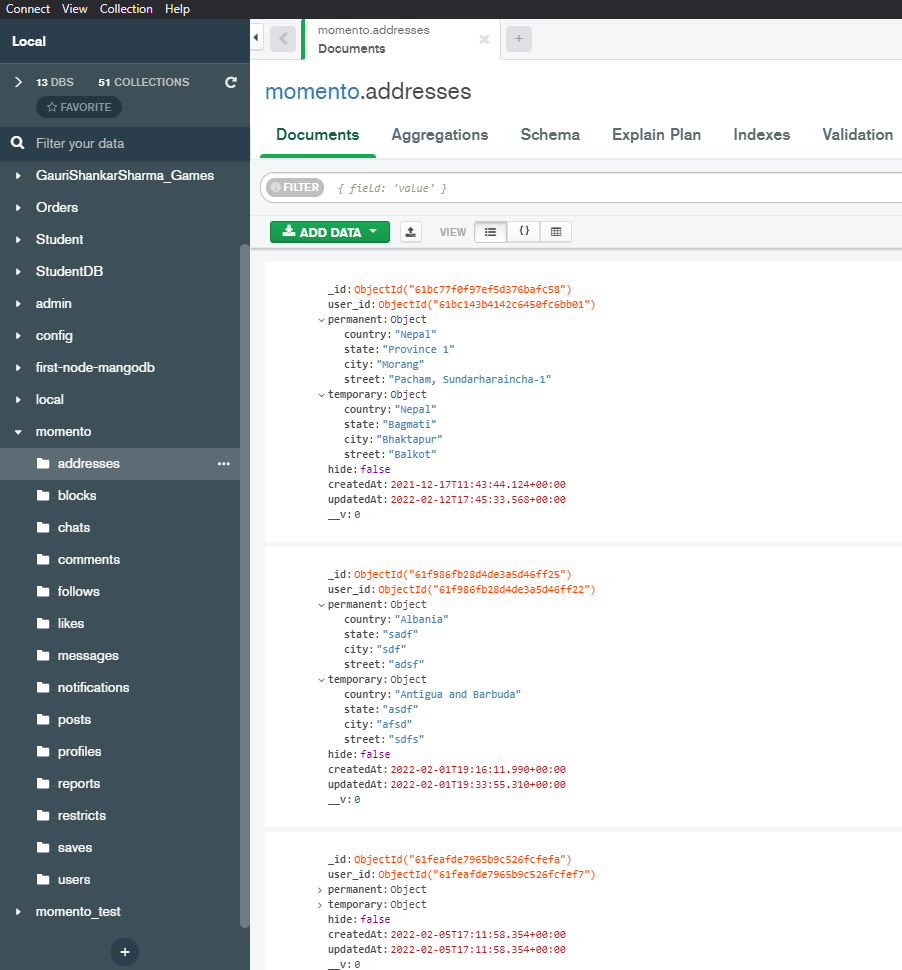


Figure 18: Address Collection

## **Post**

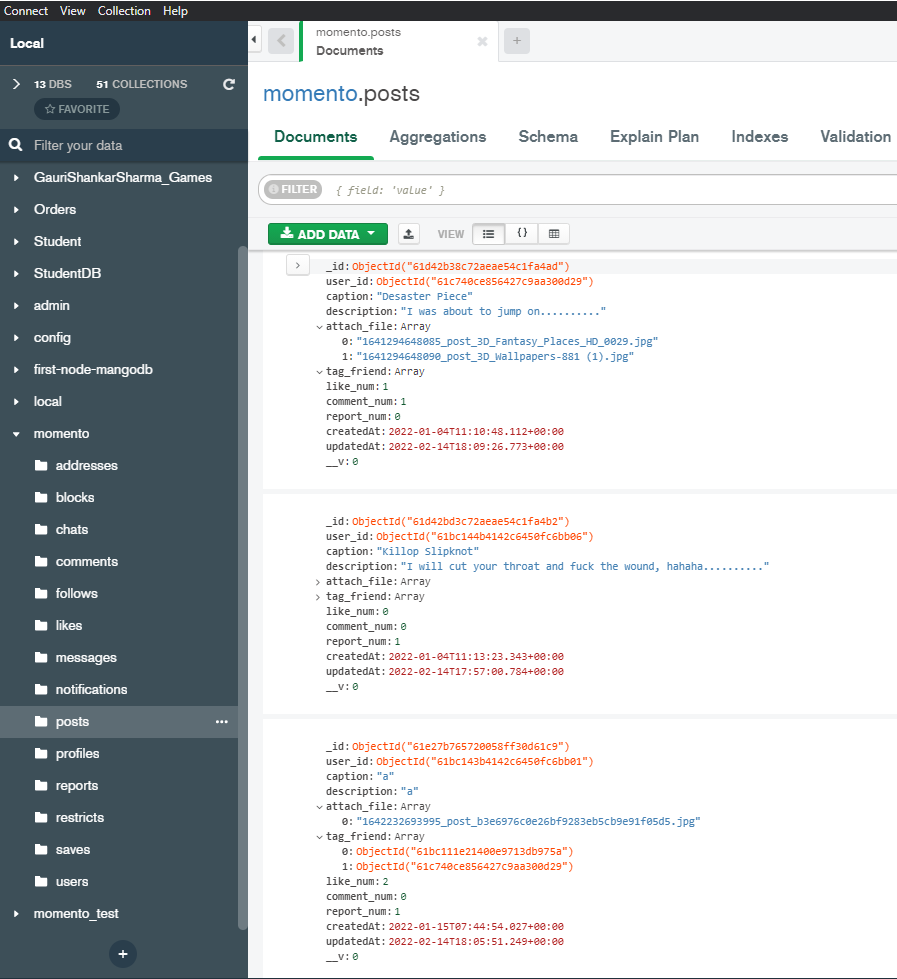


Figure 19: Post Collection

## **Follow**

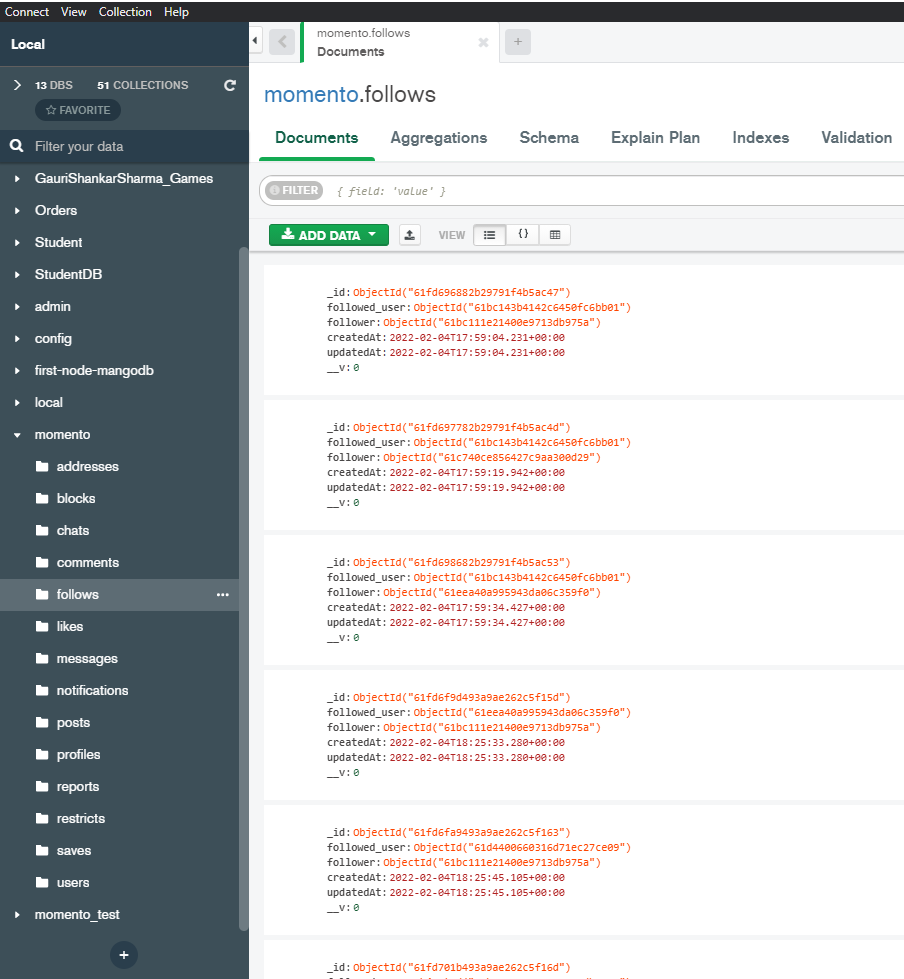


Figure 20: Follow Collection

## **Like**

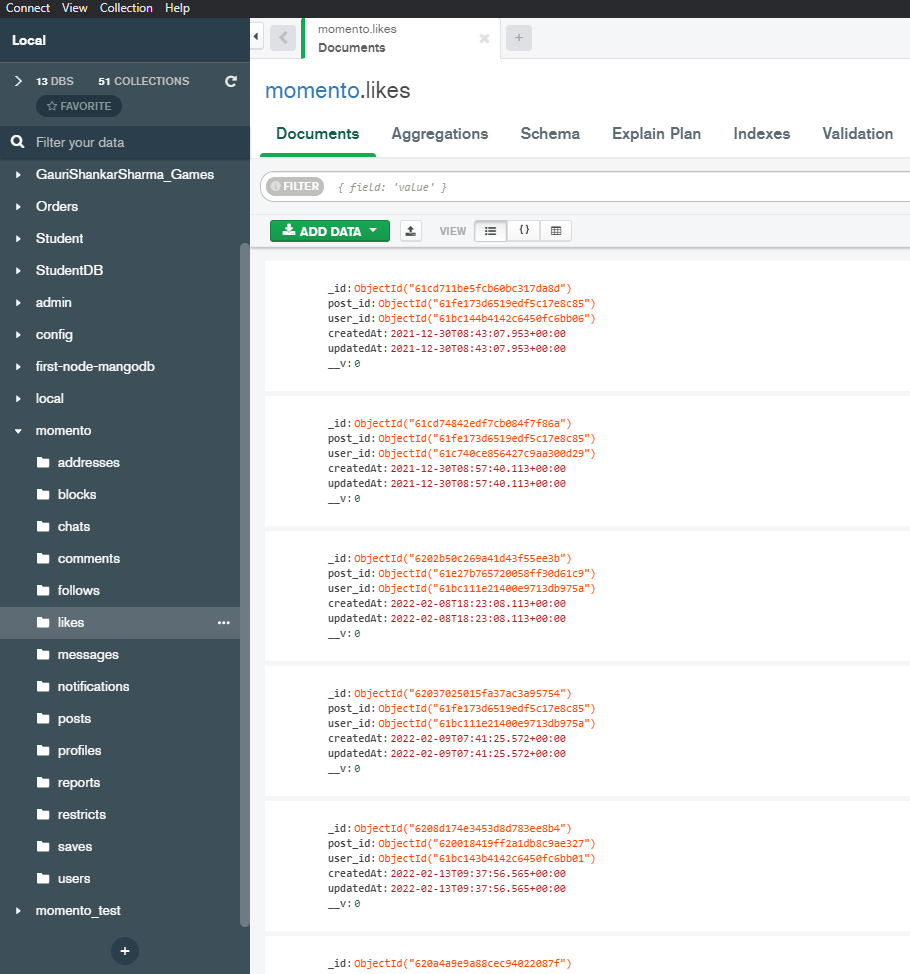


Figure 21: Like Collection

## **Comment**

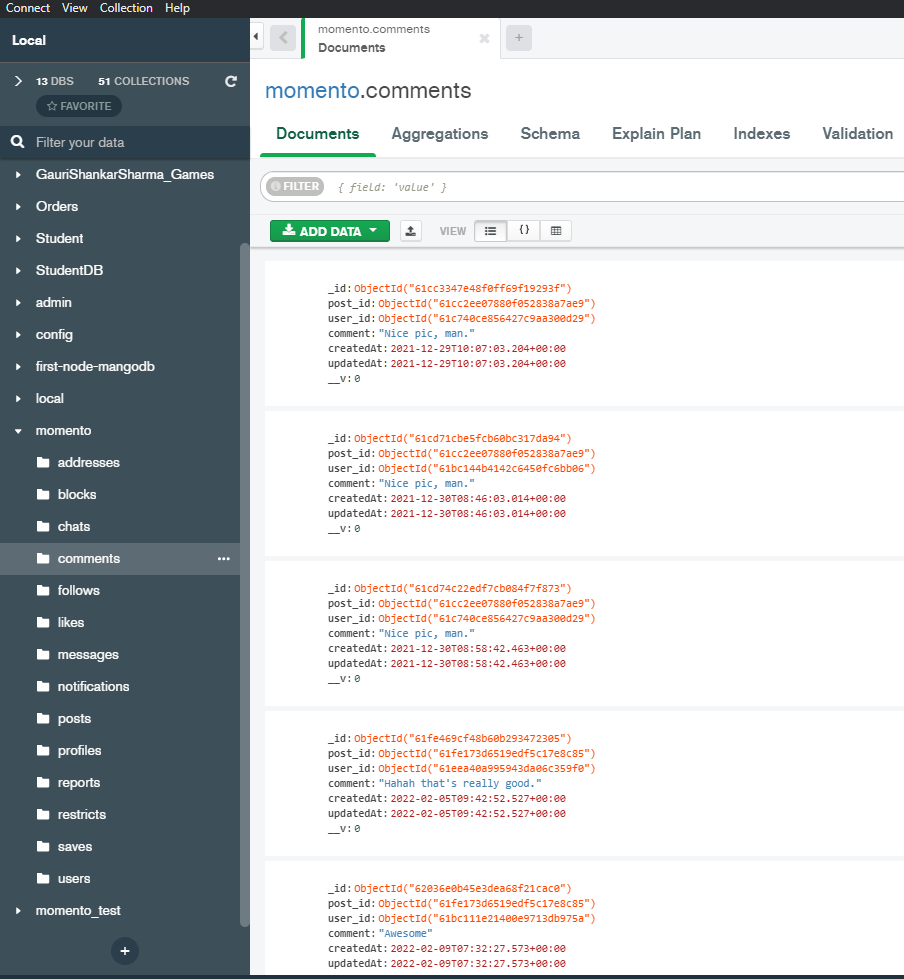


Figure 22: Comment Collection

## **Notification**

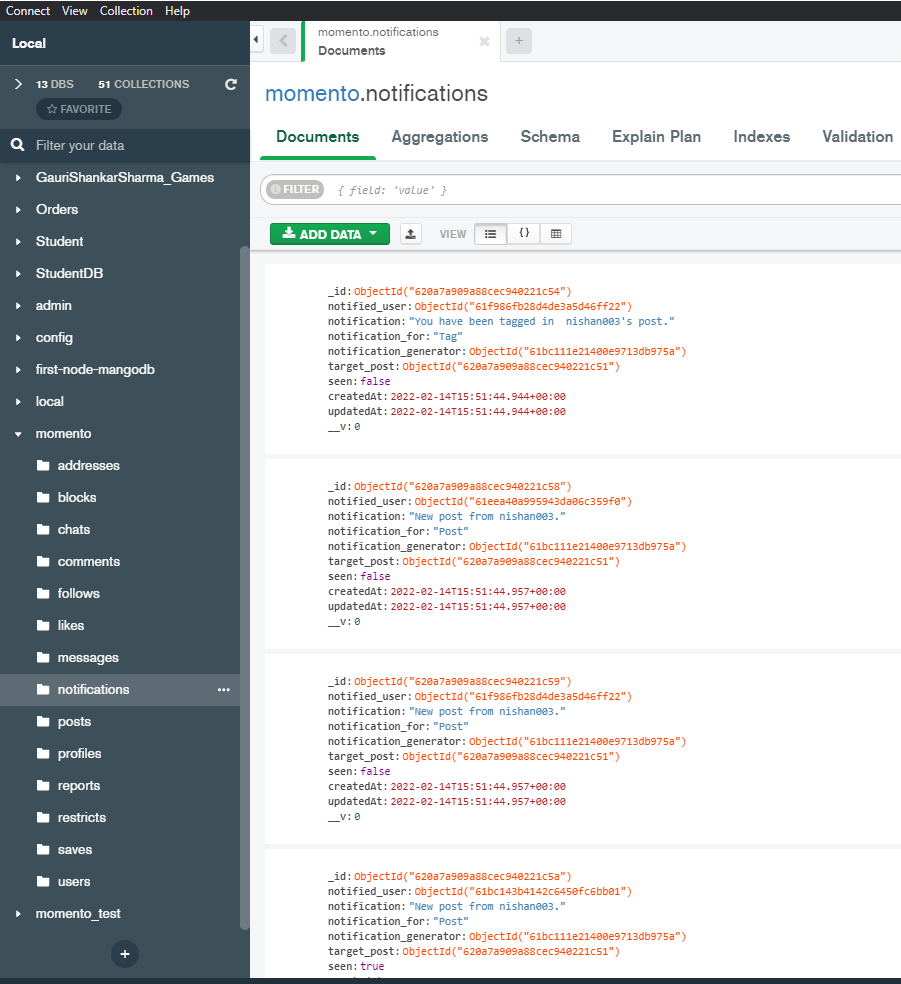


Figure 23: Notification Collection

## **Report**

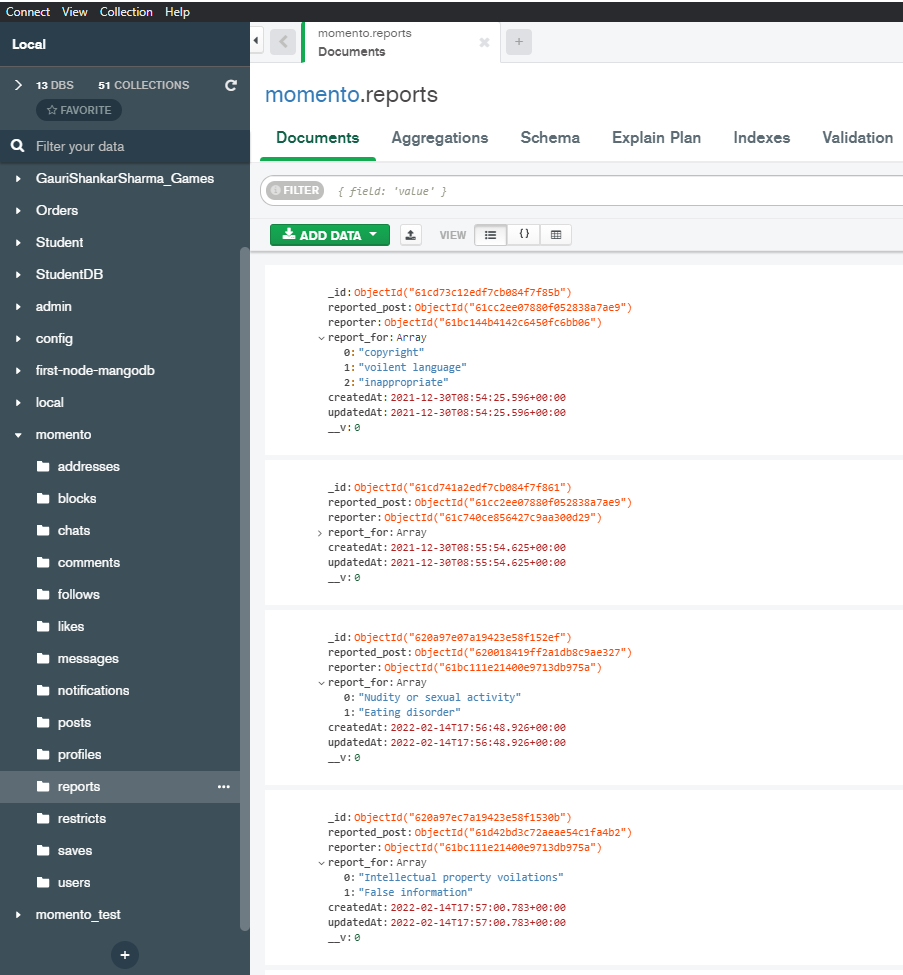


Figure 24: Report Collection