

# **Project Name: Blogging Content Management System (CMS)**

## **Database Description and Business Rules**

Blogging Content Management System lets end users to create and publish their creative ideas or experiences in the form of blogs through this application. Each post has a unique Id and an author that distinguishes between other posts in the application. A user can post more than one blog. Each user will have first name, last name, email id and password to be able to login to the system with unique ids. For each user, we must store this information to identify and authorize, optionally we can also store the profile image of user by storing the link of location of the user photo. All the user data is tracked with help of created date and updated date to find out when the record is created or last updated dates. Additionally, we can store the user\_ids of user documents of those who are following, and being followed in the two fields (following, follower) of user collection document. A user can follow many users and at same can be followed by multiple users. Many to many relationships within User collection. Unique key of the user document is ObjectId (\_id) stored automatically on inserting. For each user, there is active\_status field which we store the active / Inactive values. Admin will have more privileged access to the entire system, they can keep monitoring, track all the activity over the system. They can delete any post and warn the users for inappropriate usage of application (posting harmful content) etc.

Each blog will have title, content, poster (image source link). A post can be saved with status – drafted/ posted in order to control the visibility of the blog before user completes the entire blog content. For each post, created by details of author\_id, name, profile image source link will automatically get filled up in backend based on the logged in user. All post will store the author details from user collections along with created and updated dates of blog in post collection. For each post, all the likes and unlike by users will be stored as user ObjectId in form of array list referencing to user collection documents. Each post will have the comment field which will store all the comments of users in list of author\_details json array. Each author details document contains the author id, name, comment body – (which is array list of all comments made by that user) along with created and updated datetime of the comments. Each post will have only one user reference id as author, but each user can create more than one blog/post that describes one to many relationships from user to post collections.

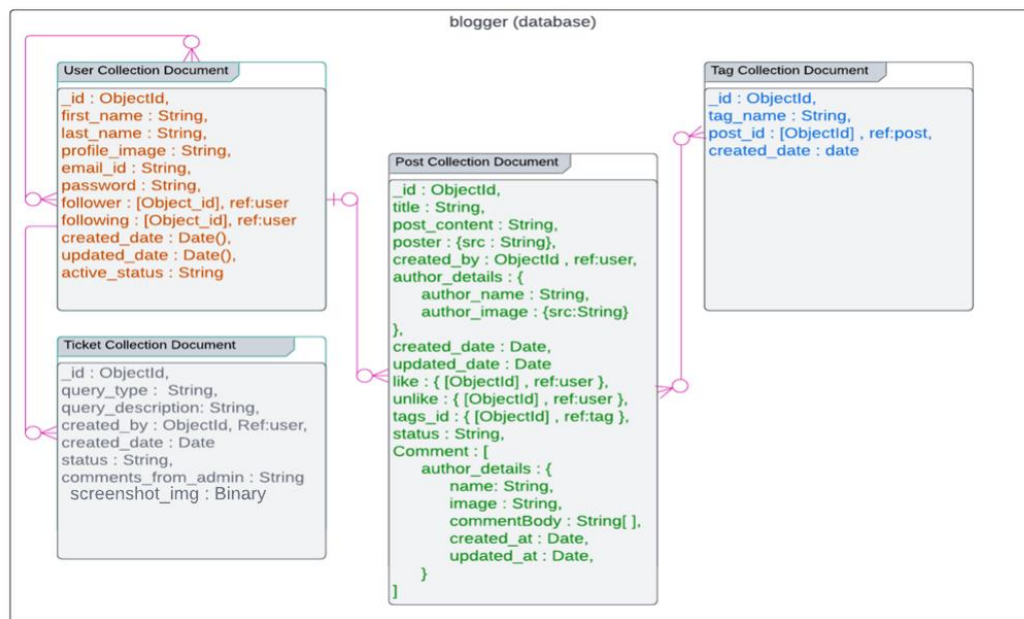
For each post, we can optionally tag it with collection to differentiate/ group together of similar posts in future. A tag collection will store tag\_name, post\_ids and creation date of tags. A post can have multiple tags and a tag can be included in multiple posts as well which describes many to many relationships. Each post can store the tag\_ids of the tags which it utilized in the form of array of ObjectId referencing tag document collections and similarly each tag will store the post\_id in array of ObjectId referencing post document collection.

The user can also report on other users with the help of tickets in the system whenever they were abused or saw the illegal posts. Then admin will review the ticket and pass to their team for complete verification and then they can block those users. The active status of the user document then will be set to block/inactive so that they can't access systems anymore.

For each ticket document collection, they store the type of ticket – complaint, suggestions to system improvement, issues/bugs. Ticket\_description will store all the brief overview of the situation faced by the user. Created By will be populated with userId and date of the ticket. User can also upload screenshots of abused/harmful content over the system while creating ticket, Within Document: Images less than of size 16MB can be stored in the document itself using the BSON type Binarydata. The initial status of ticket will be New and stored in database. Once admin checks the ticket they can move to in review/progress to show that they are working on it and there is read only field to all users except admins, which will display the comments after resolved and status to completed.

# NoSQL Database Model

**Document-Based Store (JSON File):** All data is stored in a document, so there's no need for cross-referencing and instead of storing information in a table, it's stored in a document.



## Database Collections Sample:

### Database (blogger) – Collections (user, post, tag, ticket)

MongoDB Compass - localhost:27017/blogger

Connect View Help

localhost:27017

5 DBS 8 COLLECTIONS

☆ FAVORITE

HOST localhost:27017

CLUSTER Standalone

EDITION MongoDB 6.0.1 Community

My Queries

Databases

Filter your data

admin

blogger

- post
- queryform
- tag
- user

config

database

- netflix

local

Collections

Create collection View

Sort by Collection Name

Collection Name	Storage size	Documents	Avg. document size	Indexes	Total index size
post	20.48 kB	1	669.00 B	1	24.58 kB
queryform	20.48 kB	2	241.00 B	1	36.86 kB
tag	20.48 kB	1	114.00 B	1	20.48 kB
user	20.48 kB	4	366.00 B	1	36.86 kB

User Collection Sample Document:

```
{
  "_id": {
    "$oid": "6343402f718232e230ba029c"
  },
  "first_name": "sasank",
  "last_name": "tipparaju",
  "profile_image": "https://www.smashbros.com/wiiu-3ds/images/character/toon_link/main.png",
  "email_id": "vxt88380@ucmo.edu",
  "password": "password@01",
  "follower": [
    {
      "$oid": "6343402f718232e230ba029d"
    },
    {
      "$oid": "6343402f718232e230ba029e"
    },
    {
      "$oid": "6343402f718232e230ba029f"
    }
  ],
  "following": [
    {
      "$oid": "6343402f718232e230ba029d"
    }
  ],
  "created_date": {
    "$date": {
      "$numberLong": "1651640400000"
    }
  },
  "updated_date": {
    "$date": {
      "$numberLong": "1651640400000"
    }
  }
}
```

localhost:27017

5 DBS7 COLLECTIONS

☆ FAVORITE

HOSTlocalhost:27017

CLUSTERStandalone

EDITIONMongoDB 6.0.1 Community

My Queries

Databases

Filter your data

admin

blogger

post

tag

user

config

database

netflix

local

+

Documentsblogger.user

blogger.user

4 DOCUMENTS1 INDEXES

DocumentsAggregationsSchemaExplain PlanIndexesValidation

FILTER{ field: 'value' }

OPTIONS

FIND

RESET

...

ADD DATA

VIEW

Displaying documents 1 - 4 of 4

REFRESH

```
_id: ObjectId('6343402f718232e230ba029c')
first_name: "sasank"
last_name: "tipparaju"
profile_image: "https://www.smashbros.com/wiiu-3ds/images/character/toon_link/main.png"
email_id: "vxt88380@ucmo.edu"
password: "password@01"
follower: Array
  0: ObjectId('6343402f718232e230ba029d')
  1: ObjectId('6343402f718232e230ba029e')
  2: ObjectId('6343402f718232e230ba029f')
following: Array
  0: ObjectId('6343402f718232e230ba029d')
created_date: 2022-05-04T05:00:00.000+00:00
updated_date: 2022-05-04T05:00:00.000+00:00
```

```
_id: ObjectId('6343402f718232e230ba029d')
first_name: "jeevana"
last_name: "kommula"
profile_image: "https://www.smashbros.com/wiiu-3ds/images/character/toon_link/main.png"
email_id: "vxt88380@ucmo.edu"
password: "password@02"
follower: Array
  0: ObjectId('6343402f718232e230ba029d')
following: Array
created_date: 2022-05-05T05:00:00.000+00:00
updated_date: 2022-05-05T05:00:00.000+00:00
```

Post Collection Sample Document:

```
{
  "_id": {
    "$oid": "63434883718232e230ba02a6"
  },
  "title": "Sample Post",
  "post_content": "This is a sample post body",
  "poster": "https://google.com/main.png",
  "created_by": {
    "$oid": "63434883718232e230ba02a6"
  },
  "author_details": {
    "author_name": "sasank tipparaju",
    "author_image": "https://google.com/main.png"
  },
  "created_date": {
    "$date": {
      "$numberLong": "1651640400000"
    }
  },
  "updated_date": {
    "$date": {
      "$numberLong": "1651640400000"
    }
  },
  "like": {
    "user_id": [
      {
        "$oid": "63434883718232e230ba02a5"
      }
    ]
  },
  "unlike": {
    "user_id": [
      "63434883718232e230ba02a5"
    ]
  },
  "tags": {
    "tag_id": [
      "ot434883718232e230ba0186"
    ]
  },
  "status": "drafted",
  "comment": {
    "author_details": {
      "author_name": "sasank tipparaju",
      "comment_body": [
        "This is a sample comment",
        "This is a second comment"
      ],
      "created_at": {
        "$date": {
          "$numberLong": "1397106000000"
        }
      },
      "authod_id": {
        "$oid": "634357cc718232e230ba02a7"
      }
    }
  }
}
```

MongoDB Compass - localhost27017/blogger.post

ConnectViewCollectionHelp

localhost:27017

Documentsblogger.post

5 DBS7 COLLECTIONS

HOSTlocalhost:27017

CLUSTERStandalone

EDITIONMongoDB 6.0.1 Community

My Queries

Databases

Filter your data

admin

blogger

post

tag

user

config

database

netflix

local

Documentsblogger.post

1 DOCUMENTS1 INDEXES

DocumentsAggregationsSchemaExplain PlanIndexesValidation

FILTER{ field: 'value' }

ADD DATAVIEW

Displaying documents 1 - 1 of 1

REFRESH

\_id: ObjectId('63434883718232e230ba02a6')

title: "Sample Post"

post\_content: "This is a sample post body"

poster: "https://google.com/main.png"

created\_by: ObjectId('63434883718232e230ba02a6')

author\_details: Object

author\_name: "sasank tipparaju"

author\_image: "https://google.com/main.png"

like: Object

user\_id: Array

0: ObjectId('63434883718232e230ba02a5')

unlike: Object

user\_id: Array

0: "63434883718232e230ba02a5"

tags: Object

tag\_id: Array

0: "63434883718232e230ba0186"

status: "drafted"

comment: Object

author\_details: Object

author\_name: "sasank tipparaju"

comment\_body: Array

0: "This is a sample comment"

1: "this is a second comment"

created\_at: 2014-04-10T05:00:00.000+00:00

author\_id: ObjectId('634357cc718232e230ba02a7')

Tag Collection Sample Documents:

```
{
  "_id": {
    "$oid": "6343593b718232e230ba02aa"
  },
  "tag_name": "collections",
  "post_id": [
    {
      "$oid": "6343593b718232e230ba02aa"
    },
    {
      "$oid": "6343593b718232e230ba02aa"
    }
  ],
  "created_date": {
    "$date": {
      "$numberLong": "1588568400000"
    }
  }
}
```

MongoDB Compass - localhost27017/blogger.tag

ConnectViewCollectionHelp

localhost:27017

Documentsblogger.tag

5 DBS7 COLLECTIONS

HOSTlocalhost:27017

CLUSTERStandalone

EDITIONMongoDB 6.0.1 Community

My Queries

Databases

Filter your data

admin

blogger

post

tag

user

config

database

netflix

local

Documentsblogger.tag

1 DOCUMENTS1 INDEXES

DocumentsAggregationsSchemaExplain PlanIndexesValidation

FILTER{ field: 'value' }

ADD DATAVIEW

Displaying documents 1 - 1 of 1

REFRESH

\_id: ObjectId('6343593b718232e230ba02aa')

tag\_name: "collections"

post\_id: Array

0: ObjectId('6343593b718232e230ba02aa')

1: ObjectId('6343593b718232e230ba02aa')

created\_date: 2020-05-04T05:00:00.000+00:00

CS5600DESIGN\_13892\_Group Project 2

## Ticket Collection Sample Documents

The screenshot shows the MongoDB Compass application running on a local host. The left sidebar displays the database structure, including the 'blogger' database and the 'queryform' collection. The top toolbar contains a filter bar with the query '{ field: 'value' }' and buttons for 'ADD DATA', 'VIEW', and 'REFRESH'. The main area shows two documents from the 'queryform' collection. The first document is an issue report, and the second is a complaint about an illegal post.

**Document 1:**

```
{
  "_id": ObjectId('634482a0dcecfb30391b4cd'),
  "query_type": "Issue with System",
  "query_description": "The post button is not working",
  "created_by": ObjectId('63448159dcecfb30391b4eb'),
  "created_date": 2020-05-04T05:00:00.000+00:00,
  "status": "Resolved",
  "comments_from_admin": "Added bug fix by developers"
}
```

**Document 2:**

```
{
  "_id": ObjectId('6350e33383583afd6035a131'),
  "query_type": "Complaint",
  "query_description": "The user with <username> is posting illegal post in the system",
  "created_by": ObjectId('63448159dcecfb30391b4eb'),
  "created_date": 2020-05-04T05:00:00.000+00:00,
  "status": "In Review",
  "commets_from_admin": "Team is looking into issue",
  "screenshot_img": BinData(0, 'iVBORw0KGgoAAAANSUUhEUGAAABGAAAAAYCAAAAAAgdz34AAAAAHNHCvQICAgIfAhkIAAAAAALwSF1zAAAAgAAARXB3X3/0AAAAAB10...')
}
```

```
{
  "_id": {
    "$oid": "6350e33383583afd6035a131"
  },
  "query_type": "Complaint",
  "query_description": "The user with <username> is posting illegal post in the system",
  "created_by": {
    "$oid": "63448159dcec1fb30391b4eb"
  },
  "created_date": {
    "$date": {
      "$numberLong": "1588568400000"
    }
  },
  "status": "In Review",
  "commets_from_admin": "Team is looking into issue",
  "screenshot_img": {
    "$binary": {
      "base64":
        "iVBORw0KGgoAAAANSUUEUgAAABgAAAAAYCAAAADgdz34AAAABHNCVQICAgIfAhkiAAAAAlwSFlzAAAPgAAAKYB3X3/OAA
        AAB10RVh0U29mdHdhcmUAd3d3Lmlua3NjYXB1Lm9yZ5vuPBoAAANCSURBVEiJtZZPbBtFFMZ/M7ubXdtDb1xSFyeilBapySV
        U8h80oFaooFSqiihIVIpQBKci6KEg9Q6H9kovIHoCIVQJJCKE1ENFjnAgcaSGC6rEnxBwA04Tx43t2FnvDAfjkNibxgHxnWb
        2e/u992bee7tCa00YFsffekFY+nUzFtjW0LrvjRXrCDIAaPLlW0nHL0SsZtVoaF98mLrx3pdh0qLtYPHChahZcYY07KvPFxv
        R15XPp1sN3adWiD1ZAqD6XYK1b/dvE5IWryTt2udLFedwc1+9kLp+vbbpoDh+6Tk1xBeAi9TL0taewpdmZzQDry0Ac0+jQ12
        RyohqqoYoo8RDWjrU+qXkjWtfi8Xxt58BdQuwQs9qC/afLwCw8tnQbqYAPsgxE1S6F3EAIxux2oQFKm0ihMsOF71dHYx+f3N
        ND68ghCu1YIoePPQN1pGRABKJ6Bus96CutRZMydTl+TvuiRW1m3n0eDl0vRPeEysqdXn+jsQPsrHMquGeXEaY4Yk4wxWcY5V
        /9scqOMOVUFthatyTy8QyqwZ+kDURKoMwXNKr2EeqVKcTNOajqKoBgOE28U4tdQ15p5bwCw7BWquaZSzAPlwjlithJtp3pTI
        mSqQRrb2Z8PHGigD4RZuNX6JYj6wj704TFLbCO/Mn/m8R+h6rYSub3ekokRY6f/YukArN979jcW+V/S8g0eT/N3VN3kTqWbQ
        428m9/8k0P/1aIhF36PccEl6EhOcAUCrXKZXXWS3XKd2vc/TRBG905ELC17MmWubD2nKhUKZa26Ba2+D3P+4/MNCFwg59oWV
        eYhkzgn/JDR8deKBoD7Y+1jEjGZ0sosXVTvbc6RHirr2reNy10Xd6pJsQ+ggqjk8VWFYmHrwBZW/n+uMPFiRwHB2I7ih8ciHF
        xIkd/30mk5tCDV1t+2nNu5sxxpDFNx+huNhVT3/zMDz8usXC3ddaHBj1GHj/As08fwTS7Kt1HBTmyN29vdaWw+/wbwLVOJ3u
        AD1wi/dUH7Qei66PfyuRj4Ik9is+hglfbkbfR3cnZm7chlUwLdwmpRtCohX4HUtl0cQjLYCu+fzGJH2QRKvP3UNz8bWk1qMx
        jGTOMThZ3kvglI5AzFfo379UAAAAASUVORK5CYII=",
      "subType": "00"
    }
  }
}
```