

|  |
| --- |
|  |
| Tech Talk  Design: Database |
| *Submitted by: Mia Jacobs* |
| **Maryville University | Software Development Capstone** |



|  |  |
| --- | --- |
|  | Decorative |
| Database Overview |
| Tech Talk will utilize a MongoDB database. MongoDB is a high-performance NoSQL database that is particularly useful for social networking applications due to its flexibility and scalability. Additionally, it is a popular document-oriented database that stores data in JSON-like documents with dynamic schema. If necessary, it is designed to handle large volumes of data with exceptional performance. MongoDB’s automatic failover and distributed database capabilities make it a great choice as a robust and reliable database system. |

|  |  |
| --- | --- |
|  | Decorative |
| MongoDB rationale & Reasoning MongoDB provides-speed reads and writes, making it ideal for social networking web applications that require high-performance databases MongoDB’s indexing and querying capabilities allow for quick and efficient access to data. MongoDB’s architecture includes automatic failover, which means that in the event of a server, the database can automatically redirect queries to a secondary server. This ensures that the social networking application remains operational, and users can continue to access the data without interruption. MongoDB is a distributed database, which means that it can be used in a multi-data center environment This is particularly useful for social networking web applications with a global user base, as it allows database to be located closer to the users for increased performance and reduced latency.  Overall, the flexibility scalability, high performance, automatic failover, and distributed database capabilities of MongoDB make it a choice for IT social networking web applications that require a robust and reliable database system. tech talk data structures User Profiles: The user profile data will be stored in collections in MongoDB. Each will have a unique ID, and the collected data elements including name, email address, date, location, bio, profile picture, interests, followers, and more.  Posts Comments: The user-created content/post data, such as text, images, or GIFs, and associated engagement metrics (likes, shares, comments, etc.) will be stored in a separate collection. Each post will have a unique ID, and the associated data elements including the user who made the post/comment, date/time, caption, or image/GIF.  Relationships/friendship: The network structure information about users/friends follows a graph that can represent the friend-follower relationship between users. This information might be stored using MongoDB(s). Each document may contain data such as user ID, friend/follower ID, date/time and more.  Messaging: The messaging system on the platform will be mostly dependent on use case. It can be based on chat/private messages or other forms inbox messaging. The data might be stored in a separate messaging collection containing data such as sender information, date/time, and message content.  Notifications: Each notification sent to a user would be stored in a notification document in the database. This document would contain information about the notification, such as the sender, the type of notification, and the content of the notification. How Data Structures will be used in tech talk Arrays can be used to store user data such as usernames, and login credentials. It can also be used to store the list friends for a user. Linked lists can be used to store the list of a user, each node of the linked list will represent the user who is a of the central user. Hash tables can be used to store user- content such as posts, comments, and media. By hashing the content and associating it a key and value pair, retrieval of this data will be optimized, and searching for information can made easier. Graph data structures can be used to represent a network user and identify features like mutual friends, common interests, and connections between users. It can also utilize to analyze the network’s behavior to provide insights and offer recommendations to users. Trees can be employed to enforce the hierarchical structure of a social networking web application. example, Facebook uses a tree structure to organize its friend list into sections such as family, work, and school friends. MOngoDb Document collections | JSON FOrmat Structures The MongoDB Document Collections include users, posts, comments, likes, follows, and notifications.  USERS:  {  "user\_id": string (auto generated by MongoDB),  "full\_name:” string ,  "username": string,  "email": string,  "password": string,  "profile\_pic": string (reference url to the cloud stored image),  "cover\_photo": string (reference url to the cloud stored image),,  "bio": string,  "date\_joined": date,  "friends": string (array of user IDs),  "posts": string  }  The friends field is a reference to the user\_id field of other user documents  POSTS:  {  "post\_id": string,  "user\_id": string,  "post\_text": string,  "date\_posted": date,  "comments": string,  "time\_created": date,  “likes”: string (array of user\_id’s),  “search”: string,  "comments": [  {  "user\_id": string,  "content": string,  "time\_created": date  }  ],  }  The user\_id field is a reference to the \_id field of a user document. The comments field is an array of objects, each of which contains a user\_id field that is a reference to the \_id field of a user document.  CHAT/MESSAGING:  {  "message\_id": string,  "sender\_id": string,  "recipient\_id": string,  "content": string,  "time\_sent": date  }  The sender and recipient fields are references to the \_id field of user documents.  NOTIFICATIONS:  {  "notification\_id": string,  "user\_id": string,  "message": string,  "is\_read": boolean,  “time\_sent”: date  }  The user\_id field is a reference to the \_id field of a user document. Purpose, Implementation and Interactions: **USERS:**  **Purpose:**  This is a necessary collection for the Tech Talk application. The User section identifies various unique characterizes that separates users from one another.  **Implementation:**  When new users first use the Tech Talk application, they will be able to sign up using their unique and personal username, password, and email address. Next they will be authenticated at the sign-in screen where they will be asked to enter their user name and password. There will be a forgot password option as well.  **Interaction:**  After the user successfully logs in, they will be directed to the home screen. Here they will view content posted by other users, for which they can post themselves, comment, react, or search for posts of interest.  **POSTS:**  **Purpose:**  This is an important collection for the Tech Talk application. It is what brings users together for collaboration, discussion, and education.  **Implementation:**  The user posts may include an image, explanatory text, comment section, reaction/like section. Other users will also be able to view and interact with the post.  **Interaction:**  Users will be able to search for posts that are of interest to them. They will be able to react with likes or comments that will be available for public view.  **CHAT/MESSAGING:**  **Purpose:**  Users may want to carry on conversations privately. The chat/messaging feature will allow users to connect and share ideas that will not be displayed for public view.  **Implementation:**  Users may share content such as text, images, GIFs, as well as react to private message comments.  **Interaction:**  Messages from users who are not “friends”, will need approval from the recipient prior to opening the message. This is an effort to ward off any unwanted solicitation. Additionally, users will have the ability to block other users, ceasing and attempts at exchange of future information.  **NOTIFICATIONS:**  **Purpose:**  Notifications will alert users to actions that have taken place within the app that may be of interest to the user.  **Implementation:**  Notification alerts can be triggered by a number of different events. Such events include comments on a user’s post, reactions to a user’s post, new postings from friends/followers, or new message alerts.  **Interaction:**  Users have the ability to customize their notifications so to not miss events of interest, but also to avoid alert fatigue. | |