**ADVANCED WEB APPLICATION DEVELOPMENT**

**ASSIGNMENT WEEK 8**

1.Explain Reading of Data from Documents in MONGODB.

Answer1: To read data from MongoDB, you can use read operations that retrieve data from your database in different ways. The most common read operation is the find() method, which returns a set of documents that match a query. You can also use the findOne() method to return a single document. If you want to run a custom processing pipeline, you can use the aggregate() method. To access the data, you need to get the database and collection you want to query [**1**](https://www.mongodb.com/docs/drivers/node/current/fundamentals/crud/read-operations/retrieve/).

Here’s an example of how to retrieve all documents from the “library” collection in MongoDB using Mongoose:

const mongoose = require('mongoose');

const Library = require('./models/library');

mongoose.connect('mongodb://localhost:27017/myapp', { useNewUrlParser: true });

Library.find({}, function(err, docs) {

if (err) {

console.log(err);

} else {

console.log(docs);

}

});

In this example, we first connect to the MongoDB database using Mongoose . We then use the find() method on the Library model to retrieve all the documents in the “library” collection . The first argument to find() is an empty object, which means we want to retrieve all documents in the collection . The second argument is a callback function that takes two parameters: an error object (if there is an error) and an array of documents (if there are no errors) . In this example, we simply log the array of documents to the console .

2.What are different methods to insert Document in MONGODB.

Answer 2: To insert data into a MongoDB collection, you can use the insert() or save() method. The basic syntax of the insert() command is:

db.collection.insert(

<document or array of documents>,

{

writeConcern: <document>,

ordered: <boolean>

}

)

The insert() method takes a document or an array of documents as its first parameter. To insert a single document, you can use the insertOne() method, which takes an object containing the name(s) and value(s) of each field in the document. The insert() method can also be run through the db.collection.insertOne() and db.collection.insertMany() helper methods in mongosh [**1**](https://www.tutorialspoint.com/mongodb/mongodb_insert_document.htm).

Here are some different methods to insert documents in MongoDB:

* **Insert a Single Document**: Use the insertOne() method to insert a single document into a collection.

db.collection.insertOne(

<document>,

{

writeConcern: <document>,

bypassDocumentValidation: <boolean>

}

)

* **Insert Multiple Documents**: Use the insertMany() method to insert multiple documents into a collection.

db.collection.insertMany(

[ <document 1> , <document 2>, ... ],

{

writeConcern: <document>,

ordered: <boolean>,

bypassDocumentValidation: <boolean>

}

)

* **Insert Multiple Documents with a Single Command**: Use the bulkWrite() method to insert multiple documents with a single command.

db.collection.bulkWrite(

[

{ insertOne :

{

"document" :

{

"\_id" : 4, "name" : "Mark", "age" : 33,

"status" : "A"

}

}

},

{ insertOne :

{

"document" :

{

"\_id" : 5, "name" : "John", "age" : 45,

"status" : "B"

}

}

},

{ insertOne :

{

"document" :

{

"\_id" : 6, "name" : "Samuel", "age" : 56,

"status" : "A"

}

}

}

],

{

writeConcern: <document>,

ordered: <boolean>

}

)

3.What is Find method in MongoDB.

Answer3: The find() method in MongoDB selects documents in a collection or view and returns a cursor to the selected documents. It has two parameters: query and projection. [The first ‘query’ or filter parameter specifies the search conditions for selecting documents and the second ‘projection’ parameter specifies which fields to include or exclude in the returned documents **1**](https://studio3t.com/knowledge-base/articles/mongodb-find-method/).

Here is an example of how to use the find() method:

db.collection.find(

<query>,

<projection>

)

For example, to find all documents in a collection named “users” where the “age” field is greater than 25, you can use the following command:

db.users.find({ age: { $gt: 25 } })

This will return all documents where the “age” field is greater than 25.

4.Discuss Databases and Collections with suitable Example.

Answer4: A database is a collection of data that is stored and organized in a way that allows for easy retrieval and manipulation. A database management system (DBMS) is used to manage databases. MongoDB is one such DBMS that stores data in JSON-like documents with dynamic schemas .

A collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within a single database. Collections do not enforce a schema. Documents within a collection can have different fields. Typically, all documents in a collection have a similar or related purpose .

Here’s an example of how databases and collections work together:

Suppose you are building an e-commerce website that sells books. You might create a database called “bookstore” to store all the data related to your website. Within this database, you might create several collections such as “books”, “authors”, and “orders”. The “books” collection would contain all the information about the books you sell, such as the title, author, publisher, price, and so on. The “authors” collection would contain information about the authors of the books you sell, such as their name, biography, and other relevant details. The “orders” collection would contain information about the orders placed by customers on your website .