ASSIGNMENT :- 04

Q.1. List and explain the features of advanced mongoDB?

→ 1.) Relationships in mongoDB represent how various documents are logically related to each other. Relationships can be modeled via Embedded & Referenced approaches. Such relationships can be either 1:1, 1:N, N:1 or N:N.

2.) covered query is a query in which all the fields in the query are part of an index & all the fields returned in the query are in the same index.

3.) Analyzing queries is very important aspect of measuring how effective the database & indexing design is.

4.) map-reduce is a data processing paradigm for condensing large volumes of data into useful aggregated results.

5.) mongoDB started supporting text indexes to search inside string content.

6.) Rockmongo is a mongoDB administration tool using which you can manage your servers, databases collections, documents, indexes and a lot more

7.) capped collections are fixed-size circular collections that follow the insertion order to support high performance for create, read and delete operations

Q.2. Explain any four methods of console object in node.js with suitable example.

→ In node.js, the console object is used to print message on the console. It provides several useful methods for debugging and displaying output

1.) console.log():-
- used to print normal message or information on the console.
- eg: console.log("welcome to Node.js");

2.) console.error():-
- used to display error message on the console
- eg: console.error("This is an error message");

3.) console.warn():-
- used to print warning messages.
- eg: console.warn("This is a warning message");

4.) console.table():-
- used to display data in a tabular format. It is very useful for displaying arrays or objects.
- eg:
```
const students = [
    { name: "John", marks: 85 },
    { name: "Alice", marks: 90 }
];
console.table(students);
```

Q3. Write a short note on pm2 microservices?

→ 1.) pm2 is a process management module for Node.js applications. It is used to start & monitor Node.js application so if the application goes down the process manager will restart the app immediately making it available once again.

2.) pm2 help you by restarting your node application/service every time you restart the server.

3.) Install pm2 by typing following at command line:

```
npm install pm2 -g
```

4.) The simplest way to start, deamonize & monitor your application is by using this commandline:
pm 2 start app.js.

5.) $pm2 ecosystem. - To generate an Ecosystem file.

6.) This will generate an ecosystem.config.js file.

```
module.exports = {
apps:[{
name: "app",
script: "./app.js",
env:{
NODE_ENV: "development",
},
env-production:{
NODE-ENV:" production"
}
},{
name:'worker',
script:'worker.js'
}]
}
```

Q.4. Explain the callbacks in node.js with suitable example.

→ 1.) In Node.js, a callback is a function that is passed as an argument to another function. This is called after the completion of a task.

2.) callbacks are mainly used to handle asynchronous operations like reading files, database operations, API calls, etc.

3.) Since Node.js is non-blocking and event-driven

callbacks allow the server to continue executing other operations while waiting for an operation to finish.

4.) Eg. const fs = require ('fs');
fs.readFile ('example.txt', 'utf8',
function (err, data) {
if (err) {
console.log ('Error reading file:', err);
} else {
console.log ('File content:', data);
}
});
console.log ('Reading file....');

O/P:- Reading file....
File content: (content of example.txt)

Q.5 What is the purpose of map reduce? Explain it with a suitable example.

→ 1.) Map Reduce is a programming model used for processing & generating large datasets with a parallel, distributed algorithm on a cluster.

2.) Main Purpose:-
- To handle Big Data efficiency.
- To process huge amount of data by dividing the tasks into smaller sub-tasks (map) and then combining the results (Reduce).
- Provides scalability, fault tolerance, and better performance in distributed system like Hadoop.

3. eg. problem: count the number of occurrences of each word in a given dataset.

Input Data:
Hello world
Hello Hadoop
Hadoop is Big Data.

O/P :- (Hello, 1) (Hadoop, 1) (Data, 1)
(world, 1) (is, 1)
(Hello, 1 (Big, 1)

Reduce phase O/P:-
(Hello, 2) (is, 1)
(world, 1) (Big, 1)
(Hadoop, 2) (Data, 1)

Q.6. Write a note on mongoose ODM:-

→
1.) mongoose ODM (object Data modeling) is a popular library used in Node.js applications to interact with mongoDB database.
2.) It provides a structured way to define schemas for mongoDB collections, making data handling easier and more organized.
3.) mongoose acts a bridge between the application & the mongoDB database.
4.) It allows developers to create models based on schemas that define the structure of the documents, datatypes, validation rules, & relationships between data.
5.) mongoose ODM simplifies working with mongoDB in Node.js by providing a scheme-based solution, making database operation more efficient and reliable.

Q.7. What is CURD ? Explain the CURD using node.js.

→ CURD stand for Create, Read, update and Delete. There are four basic operations used in database or application to manage data.

1.) Create (Insert Data):-

We can use the .save() or .create() method to insert data

Eg const user = require ('./models/user');

const newUser = new User ({ name: 'John', age: 25});

newUser.save()

.then(() => console.log('User created'))

.catch(err => console.log(err));

2.) Read (Fetch Data):-

We can use .find() or .findByID() to read data.

eg. User.find()

.then(users => console.log(users))

.catch(err => console.log(err));

3.) Update (modify Data):-

We can use .updateOne() or .findByIdAndUpdate() to update data.

4.) Delete (Remove Data):-

We can use .deleteOne() or .FindByIdAndDelete() to delete data.

Q.8. What is node.js ? Explain File handling in node.js.

→ 1.) Node.js is an open-source, cross-platform, Javascript runtime environment that allows developers to run Javascript code outside of web browser.

2.) Fast and scalable

3.) Event-driven and non-blocking I/O model

Emitter class gives a | console.log("Hope you're doing

classmate
Date
Page

- File Handling in Node.js:-
- File handling means performing operations like creating, reading, writing, updating and deleting files.
- Node.js provides a built-in module called fs for file handling operations.

1) Reading a file:-

```
const fs = require('fs');
fs.readFile('example.txt', 'utf8', (err, data) => {
  if (err) throw err;
  console.log(data);
});
```

2) Appending Data to a file:-

```
fs.appendFile('exple.txt', 'This text:', (err) => {
  if (err) throw err;
  console.log('Data Appended');
});
```

3) Deleting a file:-

```
fs.unlink('example.txt', (err) => {
  if (err) throw err;
  console.log('File Deleted');
});
```