Node JS

What is Node?

It is an asynchronous, non-blocking, single-threaded, and event-driven Model.

What is Node JS?

Node js is a JavaScript runtime built on chrome's v8 JavaScript engine.

Node js framework for writing server-side JavaScript applications.

The initial release was in 2009.

It is open source & used with a node package manager (NPM).

What is V8?

V8 is chrome's open-source project.

The job of the JavaScript engine (v8) is to take JavaScript code and compile it down to machine code.

The V8 engine is written in the c ++ programming language.

What is the use of node JS?

It is mainly used for making APIs (Application programming interfaces).

Is node a single threaded application?

Yes. Node is a single-threaded application with event looping.

What is the purpose of Node.js?

1. Real-time web applications
2. Network applications
3. Distributed systems   4. General purpose applications

What are the advantages of Node.js?

Following are the main advantages of Node.js:

- Node.js is asynchronous and event driven. All APIs of Node.js library are non-blocking, and its server doesn't wait for an API to return data. It moves to the next API after calling it, and a notification mechanism of Events of Node.js responds to the server from the previous API call.
- Node.js is very fast because it builds on Google Chrome's V8 JavaScript engine. Its library is very fast in code execution.
- Node.js is single threaded but highly scalable.
- Node.js provides a facility of no buffering. Its application never buffers any data. It outputs the data in chunks.

Why should I use node js?

Node js uses an event-driven, Non-blocking I/O model that makes it lightweight & efficient.

What is API?

It is a set of data that is in JSON format.

How node js works.

It is used as a backend service where javascript works on the server-side of the application. This way javascript is used on both frontend and backend. Node. js **runs on chrome v8 engine which converts javascript code into machine code**, it is highly scalable, lightweight, fast, and data-intensive.

What is an asynchronous API?

All the APIs of Node.js library are asynchronous means non-blocking. A Node.js based server never waits for an API to return data. The Node.js server moves to the next API after calling it, and a notification mechanism of Events of Node.js responds to the server for the previous API call.

Does Node.js provide Debugger?

Yes, Node.js provides a simple TCP based protocol and built-in debugging client. For debugging your JavaScript file, you can use the debug argument followed by the js file name you want to debug.

What is Anonymous Function?

The anonymous function has no name.

What is Event-driven programming?

It means as soon as the node starts its server, It simply initiates its variable, declares functions, and then simply waits for an event to occur.

What are I/O operations?

It allows a single process to serve multiple requests at the same time from the server.

What is synchronous?

In the synchronous programming model, one thing happens after the next.

What is asynchronous?

Asynchronous is a non-blocking model.

Async Function:

Asynchronous functions always return promises.

It allows programs to execute immediately without blocking the code.

Await:

Await keyword makes JavaScript wait until that promise to settles or resolve and then the result is returned

You can only use await keyword when we have async function

What is Axios?

Axios is a JavaScript library used to make HTTP requests from node js.

Promise

It is a JavaScript object which represents whether an asynchronous operation is completed or not.

Or It is something in JavaScript that tells us whether an operation has been completed or not(pending).

Promise has typically three states.

1. Pending
2. Rejected - When a promise fails. For an instance: Wrong url
3. Fulfilled - When Promises are completed successfully.

Express.js.

What is express?

1. Express is a framework that runs within node js that allows developers to create and maintain servers.
2. Express is a fast, assertive, essential, and moderate web framework of Node.js.
3. You can assume express as a layer built on the top of the Node.js that helps manage a server and routes.
4. It provides a robust set of features to develop web and mobile applications.

Why use Express?

1. Ultra-fast I/O
2. Asynchronous and single-threaded

What do you understand by the term I/O?

He terms I/O stands for input and output. It is used to access anything outside of your application. The I/O describes any program, operation, or device that transfers data to or from a medium or another medium. This medium can be a physical device, network, or files within a system.

I/O is loaded into the machine memory to run the program once the application starts

Middle wares:

Middleware functions are functions that have access to the request object (req), the response object (res), and the next middleware function in the application's request-response cycle.

The next middleware function is commonly denoted by a variable named next.

Why we use middleware:

To authenticate any attribute and segregate the code

Special function which takes req, res, next

Control flow manages

Code reusability especially for restricted routes.

Types of middleware:

Application-level middleware.(Global level)

Router-level middleware.

Error-handling middleware

Built-in middleware.

Third-party middleware.

JWT

jsonwebtoken is a just a way to exchange data between two parties

(For ex two parties are frontend and our api)

What is JWT in node JS?

JSON Web Token is **an open standard for securely transferring data within parties using a JSON object**. JWT is used for stateless authentication mechanisms for users and providers, this means maintaining sessions on the client-side instead of storing sessions on the server.

WHY JWT

jwt has a security feature where we can be sure about integrity of our data

JWT used for?

JWT is used for authentication

Req à token à client

Syntax: jwt. sign({data:data},” secret key”)

Token is for authentication

JWT is also set of claims to be transferred between two parties i.e., client & server

structure for JWT

header - encodeded with base64url

payload - in payload where we place the info (for ex userID) -encodeded with base64url

signature -

regex

A regular expression is a pattern of characters.

The pattern is used to do pattern-matching "search-and-replace" functions on text.

In JavaScript, a RegExp Object is a pattern with Properties and Methods.

Multer:

Multer is a node.js middleware for handling `multipart/form-data`, which is primarily used for uploading files.

GITHUB

GitHub is a code hosting platform for version control(records different versions of a file) and collaboration.

Git Repository?

A GitHub repository can be used to store a development project.

Git clone?

Syntax: **git clone [URL]**

This command is used to obtain a repository from an existing URL.

Git status?

To check what changes happen.

Branch:

Git branch: On which branch we are on.

Git branch -a: It shows all branches in a git repository.

Git checkout -b __branch-name:__ To switch to mentioned branch If not present then it also creates the branch.

Git add [file]: This command adds a file to the staging area.

Git commit -m "[Type commit message]": This command records or snapshots the file permanently in the version history.

__More to add here__

Body-parser:

What is body parser?

Body parser is node.js body-parsing middleware.

Why is body-parser used?

It is responsible for parsing the incoming bodies in a middleware before we handle it.

Body-parser. json():

It is also a body-parser which returns middleware that only parses JSON.

What is Express. Router ()?

Used to create a new router object in our program to handle requests.

App.listen():

Listen () method creates a listener on a specified port or path.

Module

Module in node js is a simple or complex functionality organized in single or multiple js files which can be reused throughout node js application

Module. Exports:

Exports any literals, functions, or objects as a module.

What is object Id in mongo DB?

An ObjectId is a special type typically used for unique identifiers

Enum

Array, creates a validator that checks if the value is in the given array.

Mongoose

What is mongoose?

1. Mongoose is an Object Data Modeling (ODM) Library for MongoDB and Node.js

2. Mongoose is usually used for connecting our Node.js to MongoDB.

3. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.

4. Mongoose provides a straight-forward, schema-based solution to model your application data. It includes built-in type casting, validation, query building, business logic hooks and more, out of the box.

Why mongoose?

1. It provides a lot of convenience in the creation of data in mongo db.

2. It manages relationships between data, provides schema validation, and is used to translate between objects in code and the representation of those objects in MongoDB.

3. Mongoose allows users to conveniently create and manage data in MongoDB

Terminologies

Collections: Collections in MongoDB are equivalent to tables in relational databases. They can hold multiple JSON documents.

Documents: Documents are equivalent to records or rows of data in SQL. While a SQL row can reference data in other tables, Mongo documents usually combine that in a JSON document.

Fields: Fields or attributes are like columns in a SQL table.

Schema: A Mongoose schema defines the structure of the document, default values, types, validators, etc.

Models: 1. Models are higher-order constructors that take a schema and create an instance of a document equivalent to records in a relational database.

2. Mongoose model provides an interface to the database for creating, querying, updating, deleting records, etc.

MongoDB

MongoDB is an open-source No-SQL database management program.

What is indexing in mongodb?

Indexes are special data structures that store some information related to the documents such that it becomes easy for MongoDB to find the right data file.

What is the default index value set by MongoDB ?

When creating documents in a collection, MongoDB creates a unique index using the _id field. MongoDB refers to this as the Default **_id Index**

NPM Packages

Multer:

Multer is a node.js middleware for handling `multipart/form-data`, which is primarily used for uploading files.

Body-Parser:

Node.js body parsing middleware.

Parse incoming request bodies in a middleware before your handlers, available under the `req. body` property.

What is Amazon S3?

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance.

You can use Amazon s3 to store and retrieve any amount of data at any time, from anywhere.

What is JavaScript?

JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive.

What are the HTTP request Methods?

The primary or most commonly used HTTP methods are **POST, GET, PUT, PATCH, and DELETE**. These methods correspond to create, read, update, and delete (or CRUD) operations, respectively.

What are the different ways to pass data to an API while requesting?

Passing Data in your API Calls

REST API endpoints can pass data within their requests through 4 types of parameters: **Header, Path, Query String, or in the Request Body**

What is the spread operator JavaScript?

The JavaScript spread operator (...) **allows us to quickly copy all or part of an existing array or object into another array or object**.

How will you merge two arrays using spread syntax?

spread operator [... array1, ... array2]

What is JSON?

JSON stands for **J**ava**S**cript **O**bject **N**otation

JSON is a lightweight format for storing and transporting data

JSON is often used when data is sent from a server to a web page

What is the difference between == and === operators?

== in JavaScript is used for comparing two variables, but it ignores the datatype of the variable.

=== is used for comparing two variables, but this operator also checks datatype and compares two values.

What is the typeof operator?

Typeof in JavaScript is **an operator used for type checking and returns the data type of the operand passed to it**. The operand can be any variable, function, or object whose type you want to find out using the typeof operator.

What is NaN?

NaN is a number that is not a legal number.

about the http method?

What is HTTP? The Hypertext Transfer Protocol (HTTP) is **designed to enable communications between clients and servers**. HTTP works as a request-response protocol between a client and server. Example: A client (browser) sends an HTTP request to the server; then the server returns a response to the client.

Settimeout:

The global setTimeout() method **sets a timer which executes a function or specified piece of code once the timer expires**.

JSON.stringify()

The **JSON.stringify()** method converts a JavaScript object or value to a JSON string, optionally replacing values if a replacer function is specified or optionally including only the specified properties if a replacer array is specified.

```
console.log(JSON.stringify({ x: 5, y: 6 }));

// expected output: "{"x":5,"y":6}"
```

json.parse:

The **JSON.parse()** method parses a JSON string, constructing the JavaScript value or object described by the string. An optional **reviver** function can be provided to perform a transformation on the resulting object before it is returned.

```
const json = '{"result":true, "count":42}';

const obj = JSON.parse(json);

console.log(obj.count);

// expected output: 42

console.log(obj.result);

// expected output: true
```

define aws:

Amazon Web Services (***AWS***) is the world's most comprehensive and broadly adopted cloud platform

function of router:

Router() function is **used to create a new router object**. This function is used when you want to create a new router object in your program to handle requests. Multiple requests can be easily differentiated with the help of the Router () function in Express. Js

what are http headers:

An HTTP header is **a field of an HTTP request or response that passes additional context and metadata about the request or response**. For example, a request message can use headers to indicate its preferred media formats, while a response can use a header to indicate the media format of the returned body.

What is HTTP header used for?

HTTP headers **let the client and the server pass additional information with an HTTP request or response**. An HTTP header consists of its case-insensitive name followed by a colon (:), then by its value. Whitespace before the value is ignored.

what is a document in mongo DB

MongoDB Document is **an entity in which zero or more ordered field-value pairs are stored**. In comparison to Relational Databases, it is analogous to a record or row in table.

Lodash

Lodash is a JavaScript library that works on the top of underscore.js. It helps in working with arrays, strings, objects, numbers, etc. It provides us with various inbuilt functions and uses a functional programming approach which makes coding in JavaScript easier to understand because instead of writing repetitive functions, tasks can be accomplished with a single line of code. It also makes it easier to work with objects in JavaScript if they require a lot of manipulation to be done upon them.

. **How can we send the updated document in response after updating a doc by findOneAndUpdate()**

- **Answer :** There is an option in mongoDB called { new : true }

How to add a new document by using findOneAndUpdate().

- **Answer :** { upsert : true }

What is REST API and its uses?

A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state transfer and was created by computer scientist Roy Fielding

What is a callback in a function?

A callback function is a function passed into another function as an argument, which is then invoked inside the outer function to complete some kind of routine or action.

What is called back hell?

Callback hell in Node. js is the situation in which we have complex nested callbacks. In this, each callback takes arguments that have been obtained as a result of previous callbacks. This kind of callback structure leads to lesser code readability and maintainability.

Event Loop:

Event loop is an endless loop, which waits for tasks, executes them and then sleeps until it receives more tasks.

Why is node js called single threaded?

Node JS Platform doesn't follow the Multi-Threaded Request/Response Stateless Model. It follows the Single-Threaded with Event Loop Model

about nosql?

NoSQL is an approach to database management that can accommodate a wide variety of data models, including key-value, document, columnar and graph formats. A NoSQL database generally means that it is non-relational, distributed, flexible and scalable.

1. How is a POST request different from a GET request

Data is being requested from a specific resource (through some API URL). //POST

Data is sent to be processed to a specific resource (through some API URL). //GET

2. How does a request and response cycle work? How is this cycle terminated?

When the server receives that request, it uses the information included in the request to build a response that contains the requested information

server receives request ==>build request to give response that contains some specific resource.

3. What is a schema? Why is it needed?

a schema (pronounced SKEE-mah) is the organization or structure for a database.

Database schemas are important because they help developers visualize how a database should be structured.

4. How is a schema and a model different?

a schema (pronounced SKEE-mah) is the organization or structure for a database.

Data model is a collection of conceptual tools for describing data, data-relationship and consistency constraints.

schema is structure and how it look like

model is used to use the schema (i.e pattern(schema))

5. What happens if all the documents in a user collection have an attribute called age but we remove this age attribute from our schema?

explain it in your way.

6. What is the difference between find and findOne ?

find() returns array of objects.

findOne() returns only first elements.

7. What is meant by these options in an update query- new: true and upsert: true ?

new : true =====>>>>> shows the updated values in output/response.

upsert : true ====>>>> update and insert new data in array.

8. What is an ObjectId ? is Of 12 bytes

A 4-byte timestamp

A 5-byte random value generated once per process

A 3-byte incrementing counter

//async and await function Javascript is synchronous

//Node.js is asynchronous

//undefined null are falsy values

SYNCHRONOUS BEHAVIOR	ASYNCHRONOUS BEHAVIOR
1. Synchronous code runs in sequence.This means that each operation must wait for the previous one to complete before executing.	1. Asynchronous code runs in parallel. This means that an operation can occur while another one is still being processed.
2. JavaScript is single threaded and has a synchronous execution model.	2. Asynchronous is a single-threaded, non-blocking model.
3. Synchronous code will block further execution of the remaining code until it finishes the current one.	3. Asynchronous code allows the program to be executed immediately

SQL DATABASE	NO -SQL DATABASE
RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)	Non-relational or distributed database system.(NOT ONLY SQL)
These databases have fixed or static or predefined schema	They have dynamic schema
These databases are not suited for hierarchical data storage.	These databases are best suited for hierarchical data storage.
Vertically Scalable	Horizontally scalable
Examples:- MySQL, PostgreSQL, Oracle, MS-SQL Server etc	Examples: MongoDB, GraphQL, HBase, Neo4j, Cassandra etc

PROMISE	ASYNC & AWAIT
A Promise in NodeJS is similar to a promise in real life. It is an assurance that something will be done. Promise is used to keep track of whether the asynchronous event has been executed or not and determines what happens after the event has occurred. It is an object having 3 states namely: PENDING REJECTED FULFILLED	Async/Await is used to work with promises in asynchronous functions. It is basically syntactic sugar for promises. It is just a wrapper to restyle code and make promises easier to read and use. It makes asynchronous code look more like synchronous/procedural code, which is easier to understand.

RELATIONAL DATABASE	NON-RELATIONAL DATABASE
It gives only read scalability. It manages structured data. It has a single point of failure. Deployed in Horizontal Fashion.	It is used to handle data coming in at high velocity. It manages all types of data. It has no single point of failure. It gives both read and write scalability.

MUTABILITY	IM-MUTABILITY
Reference type variables are mutable E.g. array and objects Mutable objects can be changed after it's created	Primitive types of js are immutable. E.g. undefined, null, boolean, number, string, symbol Immutable objects can be changed after it's created.

Functions	Arrow Functions
Arguments objects are not available in arrow functions Regular functions created using function declarations or expressions are 'constructible' and 'callable'. Since regular functions are constructible, they can be called using the 'new' keyword.	Arguments objects are available in arrow functions However, the arrow functions are only 'callable' and not constructible. Thus, we will get a run-time error on trying to construct non-constructible arrow functions using the new keyword.

Authentication	Authorization
Authentication verifies who the user is.	Authorization determines what resources a user can access.

Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the user.	Authorization works through settings that are implemented and maintained by the organization.
Authentication is the first step of a good identity and access management process.	Authorization always takes place after authentication.
Authentication is visible to and partially changeable by the user.	Authorization isn't visible to or changeable by the user.
Example: By verifying their identity, employees can gain access to an HR application that includes their personal pay information, vacation time, and 401K data.	Example: Once their level of access is authorized, employees and HR managers can access different levels of data based on the permissions set by the organization.

Schema	Model
A Mongoose schema defines the structure of the document,default values,types, validators, etc. Schema provides us the validation for our data which is to be stored in DB	Models are higher-order constructors that take a schema and create an instance of a document equivalent to records in a relational database. Mongoose model provides an interface to the database for creating, querying, updating, deleting records, etc.

VAR	LET	CONST
The scope of a *var* variable is functional scope.	The scope of a *let* variable is block scope.	The scope of a *const* variable is block scope.
It can be updated and redeclared into the scope.	It can be updated but cannot be redeclared into the scope	It cannot be updated or redeclared into the scope.
It can be declared without initialization.	It can be declared without initialization.	It cannot be declared without initialization.
It can be accessed without initialization as its default value is “undefined”.	It cannot be accessed without initialization, as it returns an error.	It cannot be accessed without initialization, as it cannot be declared without initialization.