1) How to manage sessions in Node.js ?

-> express-session module.

2) packages used for file uploading in Node.js

-> Multer.

3) some of the cluster methods in Node.js ?

-> fork(),

-> isWorker(),

-> process,

-> send(),

-> kill()

4) cluster in Node.js ?

-> cluster modules are created that provide us the way to make child processes

that run simultaneously with a single parent process.

5) TLS module in Node.js ?

The TLS module provides an implementation of the Transport Layer Security (TLS)

and Secure Socket Layer (SSL) protocols that are built on top of OpenSSL.

It helps to establish a secure connection on the network.

6) body-parser in Node.js

Body-parser is the Node.js body-parsing middleware. It is responsible for parsing.

the incoming request bodies in a middleware before you handle it.

It is an NPM module that processes data sent in HTTP requests.

7)avoid callback hell =>

-> Using async/await()

-> Using promises

-> Using generators

8) spawn() and fork() method?

-> Both these methods are used to create new child processes the only difference between them is that spawn() method creates a new function that Node runs from the command line whereas

-> fork() function creates an instance of the existing fork() method and creates multiple workers to perform on the same task.

9) setImmediate() and process.nextTick() methods.

-> The process.nextTick() method is used to add a new callback function at the start of the next event queue.

-> it is called before the event is processed.

-> The setImmediate is called at the check phase of the next event queue.

It is created in the poll phase and is invoked during the check phase.

10) event-driven programming in Node.js

-> synchronize the occurrence of multiple events.

-> A callback function (called an event handler) is called when an event is triggered.

-> An event loop that listens for event triggers and calls the corresponding event handler for that event.

11) event loop in Node.js

-> it to handle multiple asynchronous tasks concurrently within a single thread.

12) control flow in Node.js

-> execution.

-> handling asynchronous operations.

-> callbacks.

-> error handling.

13) middleware

-> it works between the request and the response cycle

-> it executed after the server receives the request and before the controller sends the response.

14) passport module in Node.js

-> it adding authentication features to our website or web app.

15) REST API.

-> its URL as HTTP GET or POST or PUT or DELETE request.

15) Name the tool used for writing consistent code

-> ESLint is a tool used in many IDEs to write consistent code styles