NODE JS

1. What is node.js?

Node.js is a single-threaded, javascript runtime environment for building fast and scalable server-side and network applications. Node js runs on the v8 engine It uses event-driven non-blocking i/o architecture.

2. Node.js working?

Node.js uses the “Single Threaded Event Loop” architecture to handle multiple concurrent clients.

https://www.simplilearn.com/understanding-node-js-architecture-article#:~:text=js%20Server%20Architecture-,Node.,with%20the%20JavaScript%20 callback%20 mechanism.

MVC (MODEL VIEW CONTROLLER)

It is a pattern in software design commonly used to implement user interfaces, data, and controlling logic.

- Model is the data part.
- View is the User Interface part.
- Controller is the request-response handler.

Model - It maintains the data of the application. It is the database part of the application.

View - it is for presentation like HTML page in clearly.

Controller - it controls all the requests and responses.

How does it work?

The controller is the Middle man between the Model and View.

If the user requests, it goes through the controller. The controller asks the model for information based on the request.

The model is responsible for handling all the database logic. logic means data delete, add, read.

After the Model responds back to the controller, it needs to interact with the view to render the data to the user.

The view only presents the data that the controller sends.

The view is the template, it renders web pages based on data the controller sends.

Thread pool

It has 4 threads, created to handle heavy-duty tasks that shouldn't be on the main thread.

event loop

The event loop executes the long-running operations synchronously.

Blocking - Synchronous

Non-blocking - Asynchronous

HOW DOES IT WORK?

if a client sends a request it may be Blocking or Unblocking. it'll be stored inside the event queue.

The event loop checks each data , if the data is to be executed immediately then the event loop manages it and gives a response to the client. If the data takes time to execute, it is put into the thread pool.

The thread pool manages that type of data. Multi-tasking will happen there

Thread pool gives each task and it'll be given to each thread.

After the processing, the thread pool gives the data to the event loop. The loop gives a response to the client.

3. **What is express?**

Express js is a framework for node js **it is used to support request handling, and HTTP methods and is used to develop web applications**

One feature is the non-blocking servers that can handle user requests better.

Libuv (LIbrarian Unicorn Velociraptor) - is a library written in the programming language C that helps Node.js to abstract non-blocking I/O operations and to improve efficiency while running tasks parallelly.

REPL - stands for Read Evaluate Print Loop, and it is a programming language environment (basically a console window) that takes a single expression as user input and returns the result back to the console after execution. The REPL session provides a convenient way to quickly test simple JavaScript code.

NPM - It's a library for JavaScript software packages. npm also has command-line tools to help you install the different packages and manage their dependencies.

FS (File System)
allows you to work with the file system on your computer.

- Read files - read file ()
- Create files - writefile()
  The **fs.writeFile() method** is used to write the specified data to a file asynchronously. By default, the file would be replaced if it exists. The 'options' parameter can be used to modify the functionality of the method.
- Update files - appendfile()
- Delete files - unlink()
- Rename files - rename()
- mkdir() - it is similar to writefile for making new files or folders
- rmdir() - it is similar to unlink for deleting a new file

RENDER - The **res.render()** function is used to render a view.

SEND - Use sendfile if you want to send the contents of a file as the response.

RENDER V/S SEND - one renders a view (which generally means executing code that takes a particular template and instantiates it with a set of variables), other just sends the contents of a file.
The res. send() - function basically **sends the HTTP response.**

response. write() - is used to display the normal text and Response. you can send multiple responses**.**
res. end() - if we don't give this, the program won't end, it will load any file to fetch.

API (APPLICATION PROGRAMMING INTERFACE) - the connection between the browser and server.

Fetch - The Fetch API provides an interface for fetching resources (including across the network).

REST API relies on standard HTTP methods and URLs to perform CRUD operations on resources. RESTful APIs use simple data formats like JSON or XML and are known for their simplicity and scalability. They are well-suited for stateless operations and are widely used on the web.

- **Rest Principles** - Client-server Separation, Stateless , Cacheable

SOAP is a protocol for structured information exchange in web services. It can use various transport protocols, including HTTP and SMTP, and it uses XML for message formatting. SOAP is known for its rigid contract-based approach, emphasizing security and reliability. It is often used in enterprise-level applications.

GraphQL is a query language for APIs that allows clients to request precisely the data they need. It provides a single endpoint and a typed schema, making it flexible and efficient. GraphQL is popular for optimizing data fetching, reducing over-fetching, and enabling real-time data updates through subscriptions. It's commonly used in modern web and mobile applications.

Streams - Streams are a mechanism for handling input and output in a continuous non-blocking fashion. they provide a way to read and write data in small chunks allowing for more efficient memory usage and faster processing.
There are four types of streams in the node

1. **Readable** - used to read data from a file or a network socket.
2. **Writable** - used to write data from a file or a network socket.
3. **Duplex** - can be used as both
4. **Transform** - Transform streams are duplex streams that can modify or transform the data as it is being processed such as compressing or encrypting it.

Streams provide a flexible way to work with large amounts of data by dividing it into smaller chunks and processing it as it arrives, instead of loading it into memory

Buffer - When a program sends data to a file or to another program, the data is usually stored in a buffer before it is written to disk or transmitted over the network. Buffers can be used to improve the performance of a program by reducing the number of I/O operations that are needed. For example, if a program needs to read a large file, it can use a buffer to read the file in chunks instead of reading the entire file at once. This can improve performance by reducing the amount of disk I/O that is needed.

DNS (Domain Name System) - It is a system that translates domain names into IP addresses and IP addresses to the domain name, which computers use to communicate with each other over the internet.

Module - Modules are JavaScript libraries you can include in your project.

Core module - It is a Built-in module **Ex**- HTTP, URL, path, fs.
Local modules - We can create our own modules and we can use them in our Project. We can export this module using the module. Exports

NPM Third-party Modules - When we install the node also get the NPM (Node Package Manager)

Logger - By analyzing the data in the logs, we can glean insights, resolve bugs much quicker, and detect problems early and as they happen.

Morgan - Morgan is **an HTTP request-level Middleware**. used **to log HTTP requests and errors, and simplifies the process**.

Middleware

The middleware in node. js is a function that will have all the access for requesting an object, responding to an object, and moving to the next middleware function.

if any request comes from the client it'll go through each middleware function after that we'll get a response. this cycle is called the REQUEST RESPONSE CYCLE.

- Request(req) and Response(req) are middleware.

Types of middleware - **Application-level middleware** - body-parser middleware, which is used to parse incoming request bodies., **Router-level middleware -** express.Router() function, which creates a new router object**.**, **Built-in middleware** - express.json(), which parses JSON request bodies. , **Error-handling middleware** - express.errorHandler() function, which returns an error response to the client., **Third-party middleware** - Morgan, which logs HTTP requests.

App. use - app. use() function is used **to mount the specified middleware function** or modules at the path that is being specified.

App.all - It is used to handle all HTTP requests to a specific path, (GET, POST, PUT, DELETE, etc.).

App.set - the app.set method is used to set various variables or configuration options for the application.

App.get - The app.get method is used to route HTTP GET requests to a specific path in your application.

Stateless HTTP - The HTTP protocol is a stateless one. This means that every HTTP request the server receives is independent and does not relate to requests that came prior to it.

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 a server.

HTTPS is HTTP with encryption and verification. The only difference between the two protocols is that HTTPS uses TLS (Transport Layer Securefiles (SSL)) to encrypt normal HTTP requests and responses and to digitally sign those requests and responses.

GET is used to **request** data from a specified resource.

POST is used to **send** data to a server to create/update a resource.

DELETE: This method is used to delete resources on the server.

PUT - PUT is a technique of **altering** resources when the client transmits data that revamps the whole resource.

PATCH - PATCH is a technique for transforming the resources when the client transmits partial data that will be updated without changing the whole data.

Session - Sessions are server-side files that contain user data. The session ends when the user closes the browser or logs out of the program.

Cookies - Cookies are client-side files on a local computer that hold user information. Cookies expire after the user-specified lifetime. types - Session Cookie, Persistent cookie, Secure cookie, Third party cookie Understanding Cookies and Implementing them in Node.js | Engineering Education (EngEd) Program

Child_process - Usually, Node.js allows single-threaded,non-blocking performance but running a single thread in a CPU cannot handle increasing workload hence the child_process module can be used to spawn child processes. The child processes communicate with each other using a built-in messaging system. **Four Ways** - spawn() method, fork() method, exec() method,execFile() method

Nodemon - nodemon is a tool that helps develop nodes. js-based applications by automatically restarting the node application when the file changes in the directory are detected.

setImmediate() - After the current event loop finishes, it will execute a function**.**

setTimeout() - schedules a script to be run after a minimum threshold in ms has elapsed.

setinterval() - if we give a statement in a global function it will execute continuously for a specified time interval until we terminate.

Clearinterval() - clearInterval **is a method that can stop a setInterval timer.**

secret - a random unique string key used to **authenticate** a session. The key is usually long and randomly generated in a production environment.

Package.Json - if you want to copy a code and use it in another system, so you don't want to copy all dependencies with your file ex:- node module folder, you want only your code and PACKAGE.JSON.All the dependencies are installed inside this package (pack.json).

package.js dependencies - The dependencies in your project's package. json **allows the project to install the versions of the modules it depends on**.

Query parameters - The query parameter is the variable whose value is passed in the URL in the form of a key-value pair at the end of the URL after a question mark (?). For example, www.geeksforgeeks.org? name=abc where 'name' is the key of the query parameter whose value is 'abc'.

Req.query ==== Query parameters

Req.param - The req.params property is an object that contains the properties which are mapped to the named route "parameters". For example, if you have a route as /api/:name, then the "name" property is available as req.params.name.

View engines are **helpful for rendering web pages using template files**. it is **responsible for rendering the view into HTML form to the browser**.

Handlebars (**hbs**) - Handlebars. js is a Javascript library used to create reusable web page templates. The templates are a combination of HTML, text, and expressions.

HTTP status codes

- 200 - *The request has succeeded.*
- 400 - BAD REQUEST - The request could not be understood by the server due to **malformed syntax**. The client SHOULD NOT repeat the request without modifications.
- 401 - Unauthorized - The request requires user authentication
- 402 - Payment Required - Reserved for future use used as part of some form of digital cash.
- 403 - Forbidden - The server understood the request but refused to fulfill it.
- 404 - Not Found.
- 500 - INTERNAL ERROR - The server encountered an unexpected condition that prevented it from fulfilling the request.

CORS - Cross-Origin Resource Sharing (CORS) is **an HTTP-header-based mechanism that allows a server to indicate any origins (domain, scheme, or port) other than its own from which the browser should permit loading resources**.

Body parser - body-parser **extracts the entire body portion of an incoming request stream and exposes it on req.body.**

PM2 - PM2 is **a production process manager for Node. js applications with a built-in load balancer**. It lets you keep applications alive forever, reload them without downtime, and facilitate common system admin tasks.

POSTMAN - Postman has a powerful runtime based on Node. js that **allows you to add dynamic behavior to requests and collections**. This allows you to write API tests, build requests that can contain dynamic parameters, and pass data between requests

TYPES OF ERROR - ReferenceError, RangeError, TypeError, URIError, EvalError, and SyntaxError

YARN - Yet Another Resource Negotiator (YARN).NPM & YARN are package managers used for Javascript coding, created by Facebook.