

# COMPLETE



## Introduction to NodeJS

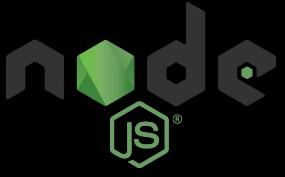
CERTIFICATE

NOTES

Ex- amazon Microsoft

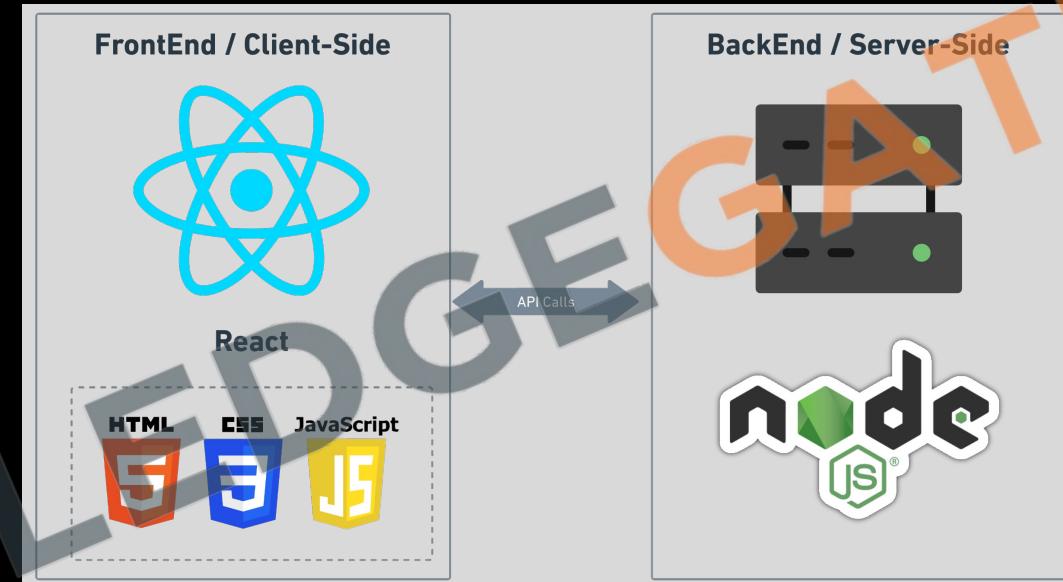


You Tube [Playlist Link](#)

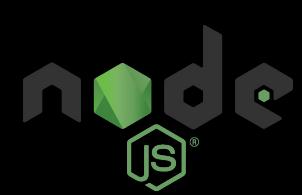


# NodeJS

## Complete Course



KNOWLEDGE



# Introduction to NodeJS

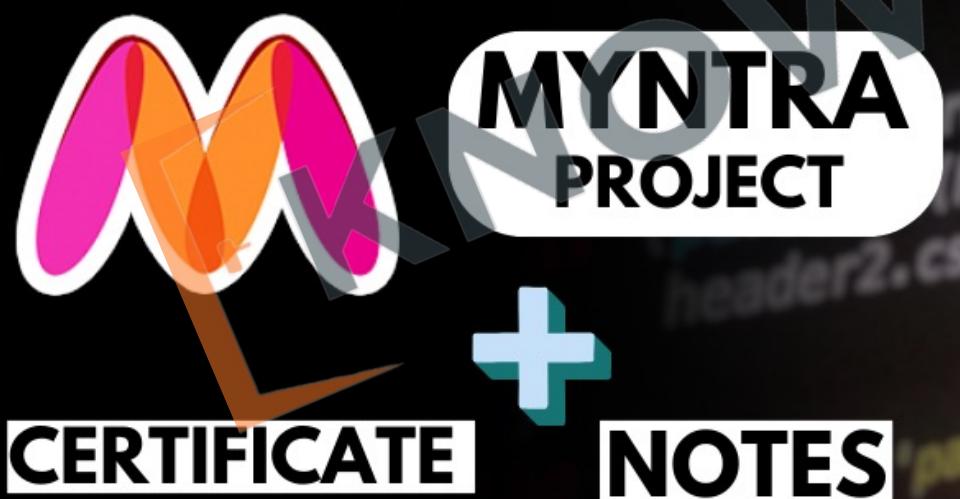
1. Pre-requisites
2. What is NodeJS
3. NodeJs Features
4. JavaScript on Client
5. JavaScript on Server
6. Client Code vs Server Code
7. Other uses of NodeJs
8. Server architecture with NodeJs





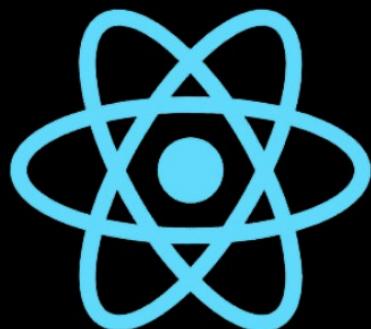
JS is required for NodeJS

# COMPLETE JS JAVASCRIPT 14 HOURS



 React is recommended before NodeJS

# COMPLETE



React



REDUX

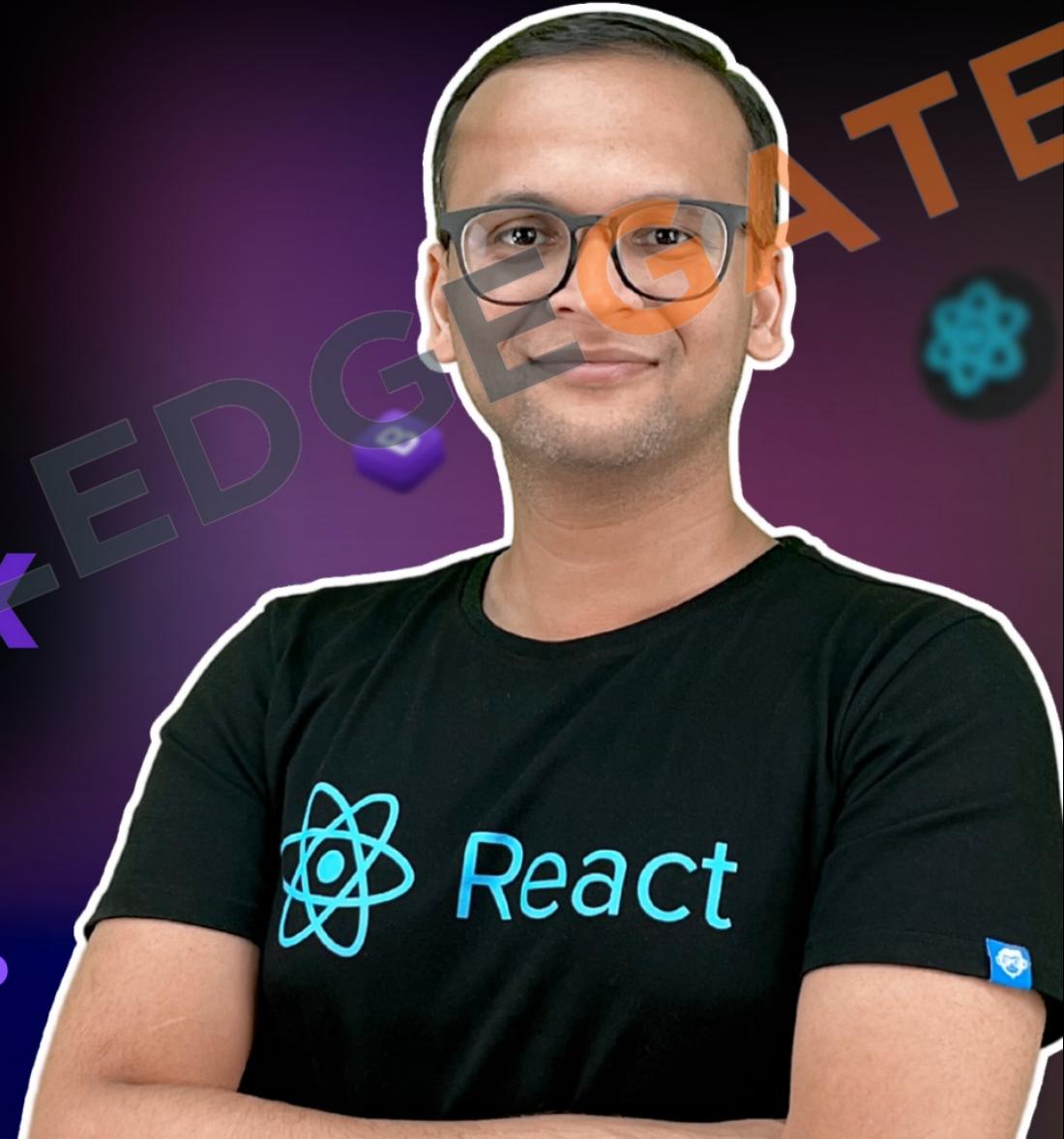
20 HOURS

6 PROJECTS

B using  
Bootstrap

CERTIFICATE

NOTES





## 2.What is NodeJS

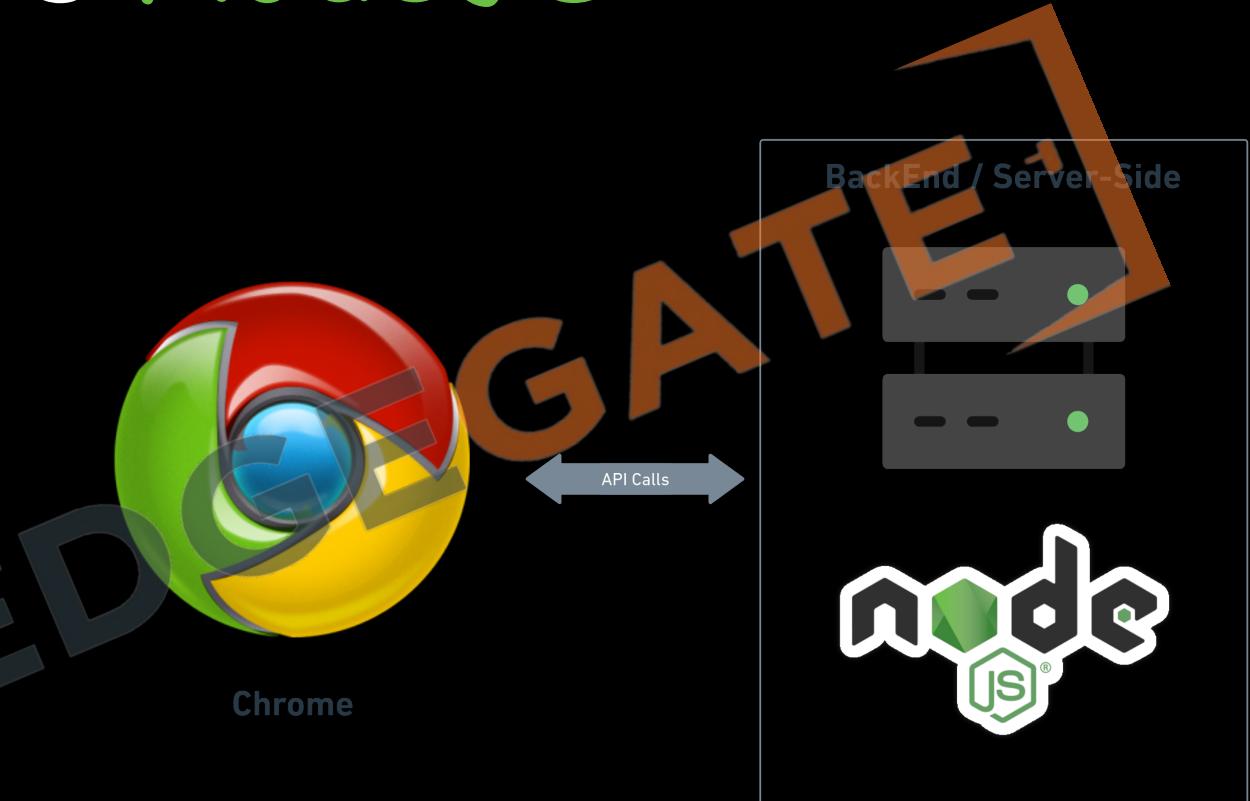


1. **JavaScript Runtime:** Node.js is an **open-source**, **cross-platform** runtime environment for executing **JavaScript code outside of a browser**.
2. **NodeJs** is a **JavaScript** in a **different environment** means **Running JS on the server or any computer**.
3. **Built on Chrome's V8 Engine:** It runs on the **V8 engine**, which **compiles JavaScript directly to native machine code**, enhancing performance.
4. **V8** is written in **C++** for speed.
5. **V8 + Backend Features = NodeJs**



# 2.What is NodeJS

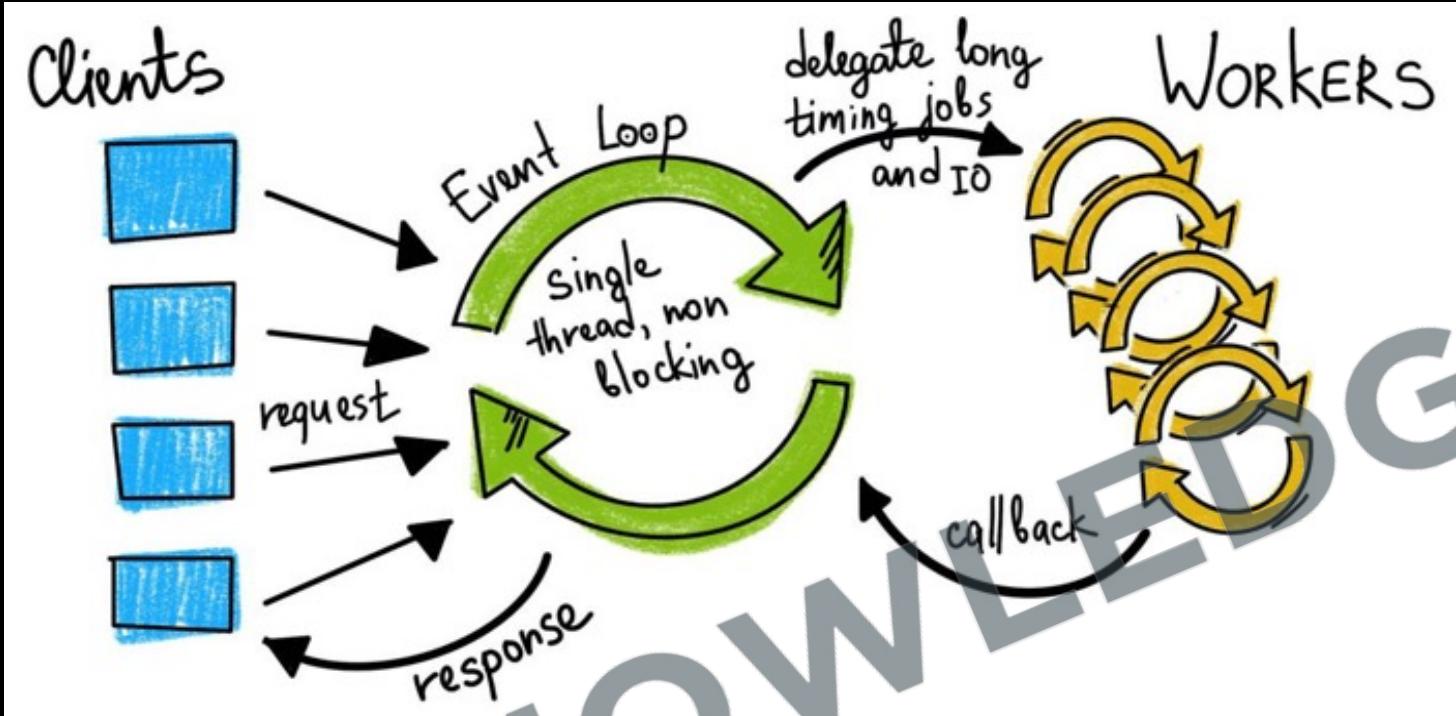
- 1.Design:** Features an **event-driven**, **non-blocking I/O** model for efficiency.
- 2.Full-Stack JavaScript:** Allows using JavaScript on both **server** and **client** sides.
- 3.Scalability:** Ideal for **scalable** network applications due to its architecture.
- 4.Versatility:** Suitable for web, real-time chat, and **REST API** servers.





# 3. NodeJS Features

(Added)



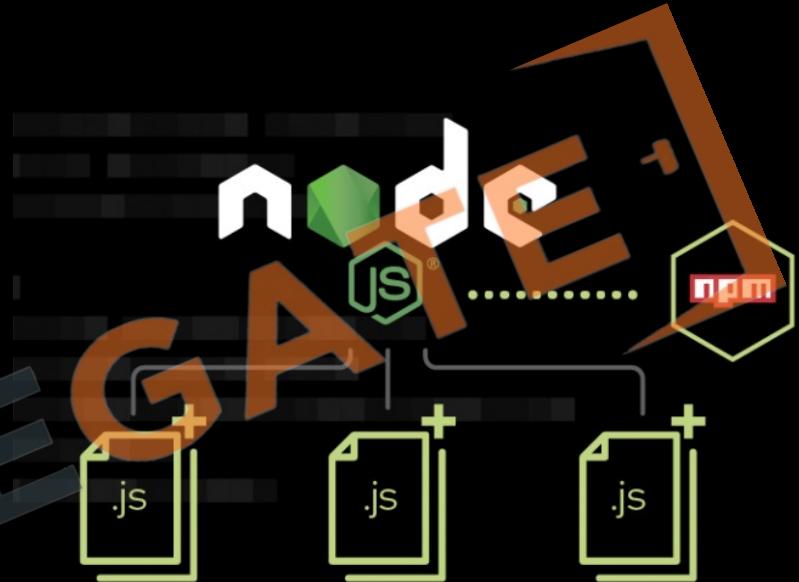
1. **Non-blocking I/O:** Designed to perform non-blocking operations by default, making it suitable for I/O-heavy operations.
2. **Networking Support:** Supports TCP/UDP sockets, which are crucial for building lower-level network applications that browsers can't handle.

KNOWLEDGE  
SEGREGATE



# 3. NodeJs Features

(Added)

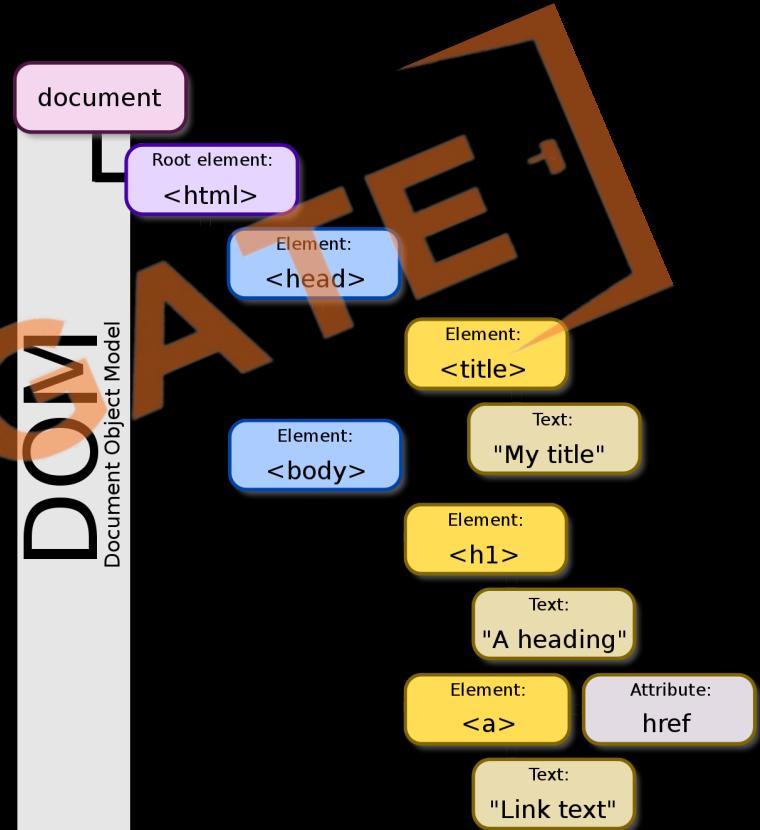
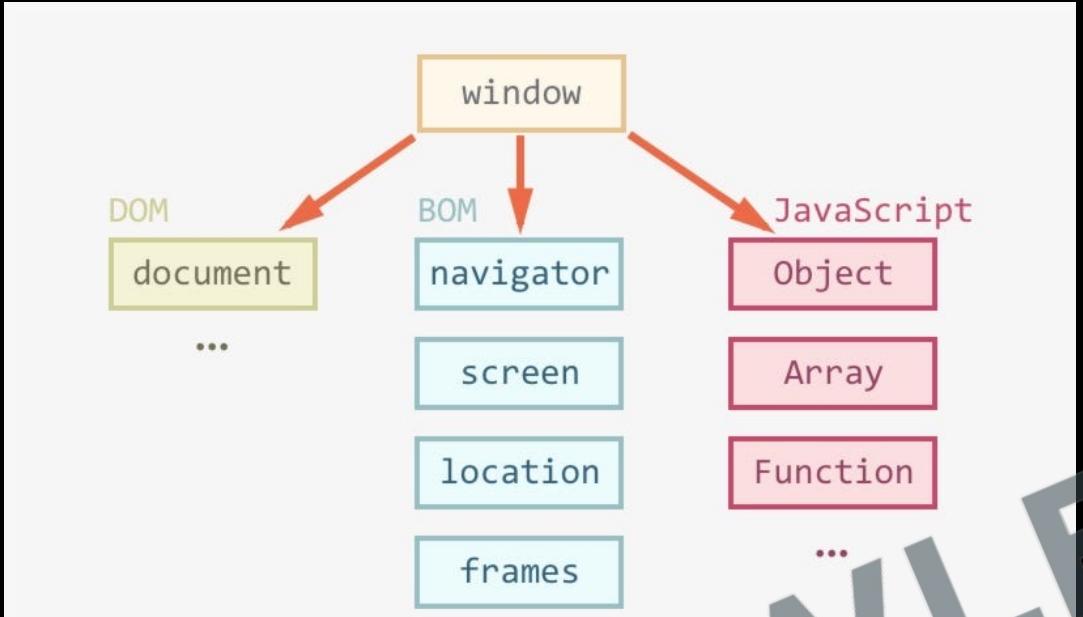


- 1. File System Access:** Provides APIs to **read and write files** directly, which is **not possible in browser environments** for security reasons.
- 2. Server-Side Capabilities:** Node.js enables JavaScript to run on the server, **handling HTTP requests, file operations**, and other server-side functionalities.
- 3. Modules:** Organize code into **reusable modules** using `require()`.



# 3. NodeJs Features

(Removed)

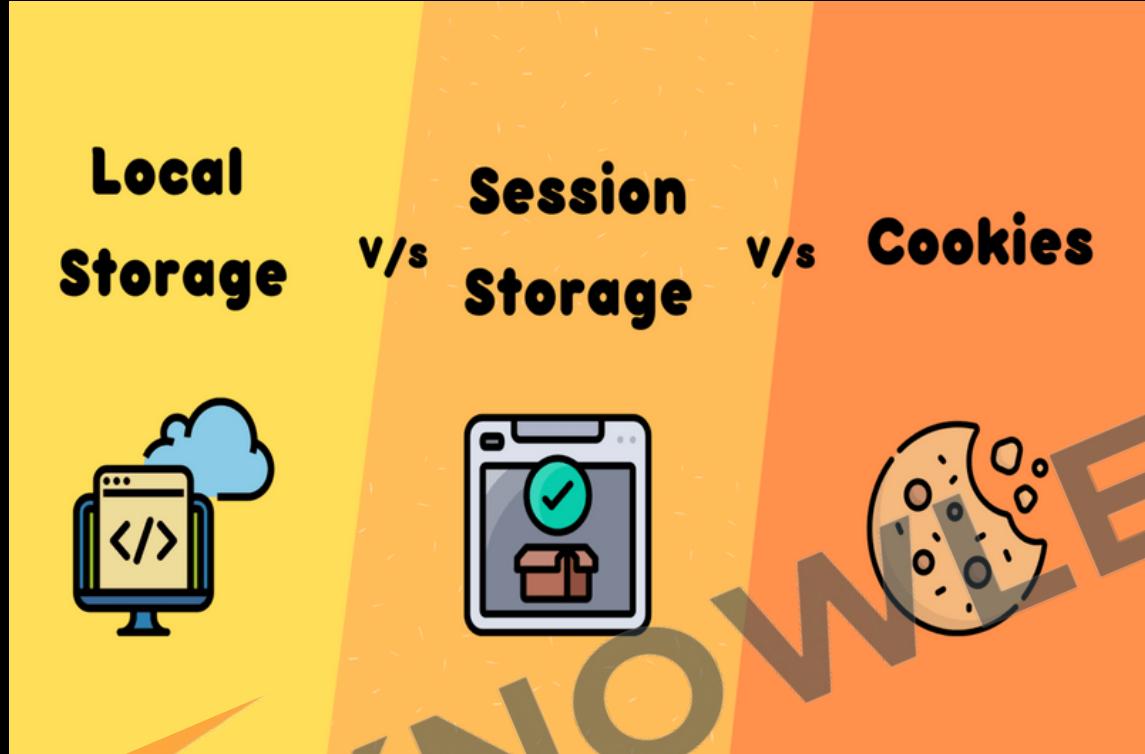


1. **Window Object:** The global `window` object, which is part of web browsers, is absent in `Node.js`.
2. **DOM Manipulation:** `Node.js` does not have a built-in Document Object Model (DOM), as it is **not intended to interact with a webpage's content**.
3. **BOM (Browser Object Model):** No direct interaction with things like `navigator` or `screen` which are part of `BOM` in browsers.

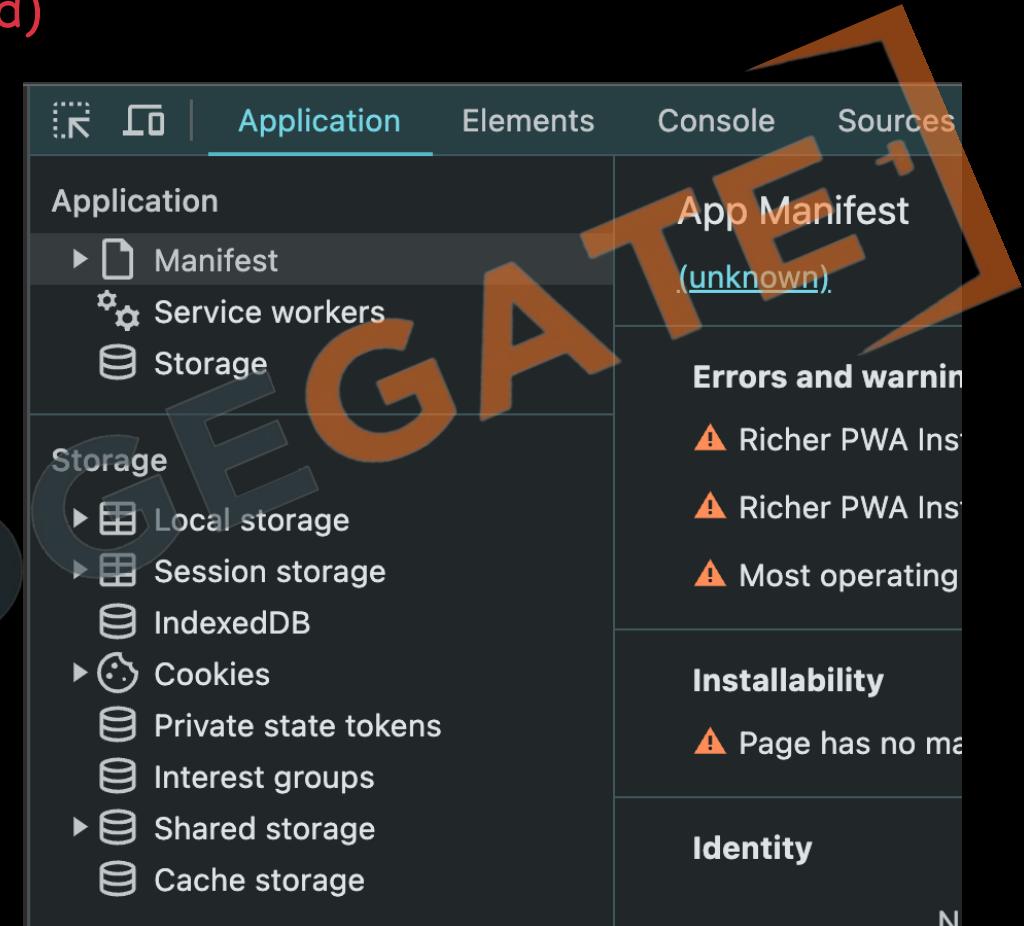


# 3. NodeJs Features

(Removed)



Web-Specific APIs: APIs like `localStorage`, `sessionStorage`, and `fetch` are not available in Node.js.





# 4. JavaScript on Client



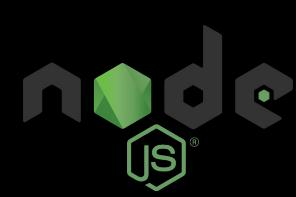
1. **Displays Web Page:** Turns HTML code into what you see on screen.
2. **User Clicks:** Helps you interact with the web page.
3. **Updates Content:** Allows changes to the page using JavaScript.
4. **Loads Files:** Gets HTML, images, etc., from the server.



# 5. JavaScript on Server

- 1. Database Management:** Stores, retrieves, and manages data efficiently through operations like CRUD (Create, Read, Update, Delete).
- 2. Authentication:** Verifies user identities to control access to the system, ensuring that users are who they claim to be.
- 3. Authorization:** Determines what authenticated users are allowed to do by managing permissions and access controls.
- 4. Input Validation:** Checks incoming data for correctness, completeness, and security to prevent malicious data entry and errors.
- 5. Session Management:** Tracks user activity across various requests to maintain state and manage user-specific settings.





# 5. JavaScript on Server

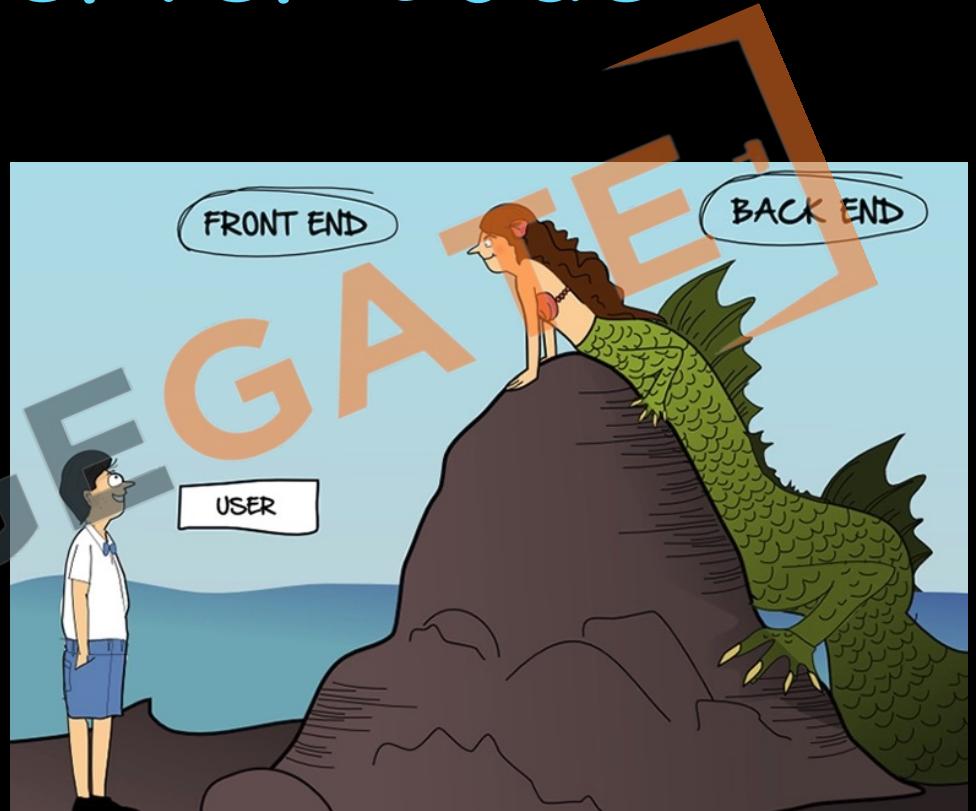
6. **API Management:** Provides and handles interfaces for applications to interact, ensuring smooth data exchange and integration.
7. **Error Handling:** Manages and responds to errors effectively to maintain system stability and provide useful error messages.
8. **Security Measures:** Implements protocols to protect data from unauthorized access and attacks, such as SQL injection and cross-site scripting (XSS).
9. **Data Encryption:** Secures sensitive information by encrypting data stored in databases and during transmission.
10. **Logging and Monitoring:** Keeps records of system activity to diagnose issues and monitor system health and security.





# 6. Client Code vs Server Code

1. User/client can't access server code directly.
2. Client must raise requests for particular APIs to access certain features or data.
3. Environment Access: Server-side JavaScript accesses server features like file systems and databases.
4. Security: Server-side code can handle sensitive operations securely, while client-side code is exposed and must manage security risks.
5. Performance: Heavy computations are better performed on the server to avoid slowing down the client.





# 6. Client Code vs Server Code

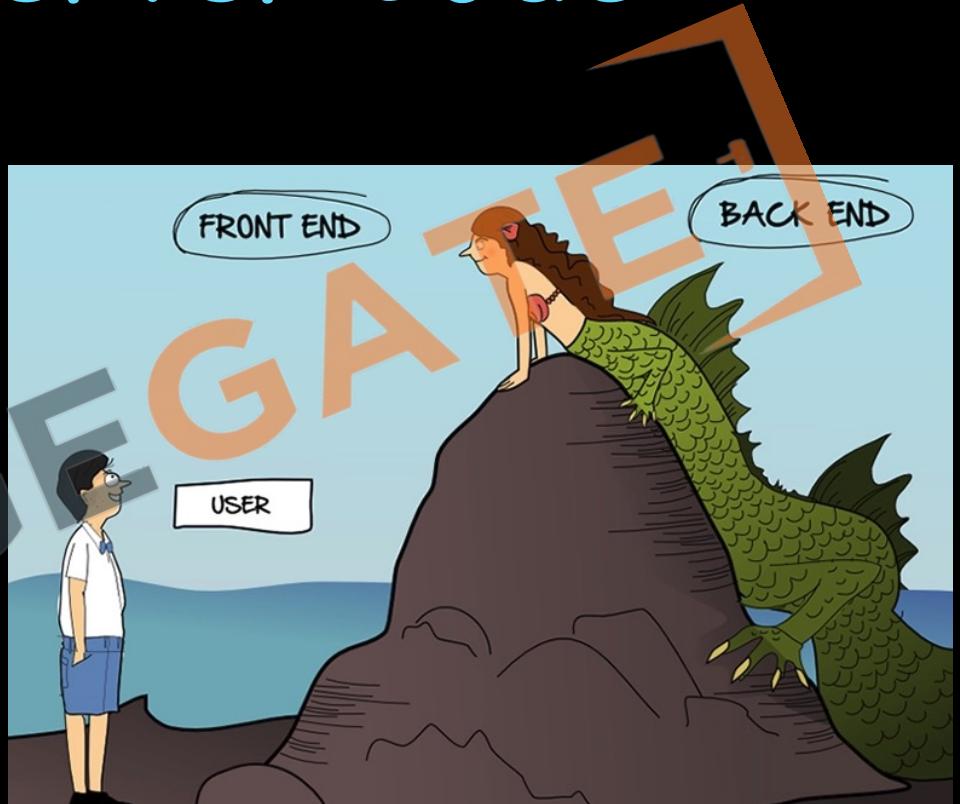
6. **Resource Utilization:** Servers generally offer **more powerful processing capabilities** than client devices.

7. **Data Handling:** Server-side can directly manage **large data sets and database interactions**, unlike client-side JavaScript.

8. **Asynchronous Operations:** Server-side JavaScript is optimized for non-blocking I/O to **efficiently manage multiple requests**.

9. **Session Management:** Servers handle **sessions and user states** more comprehensively.

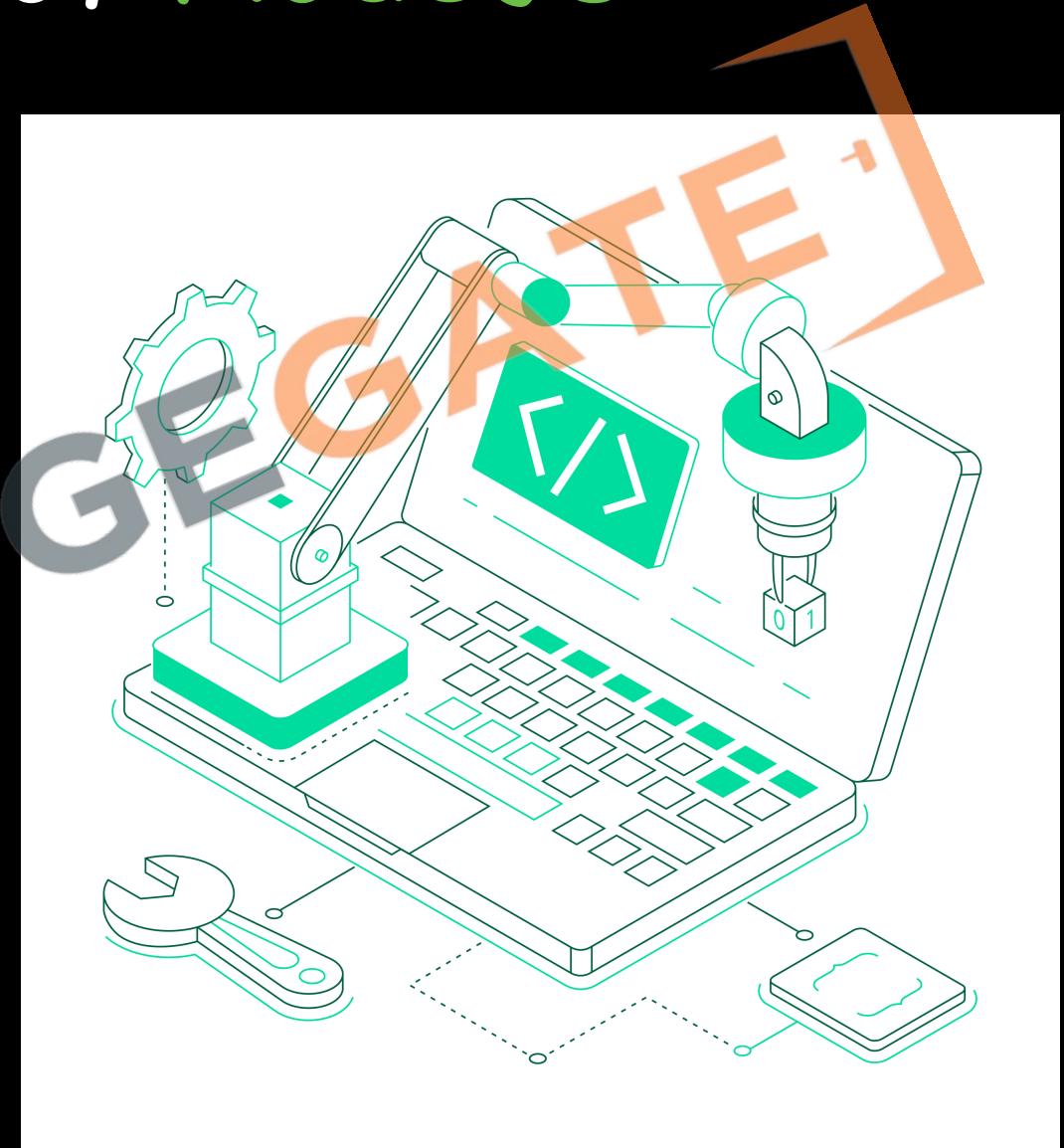
10. **Scalability:** Server-side code is designed to scale and handle requests from **multiple clients simultaneously**.





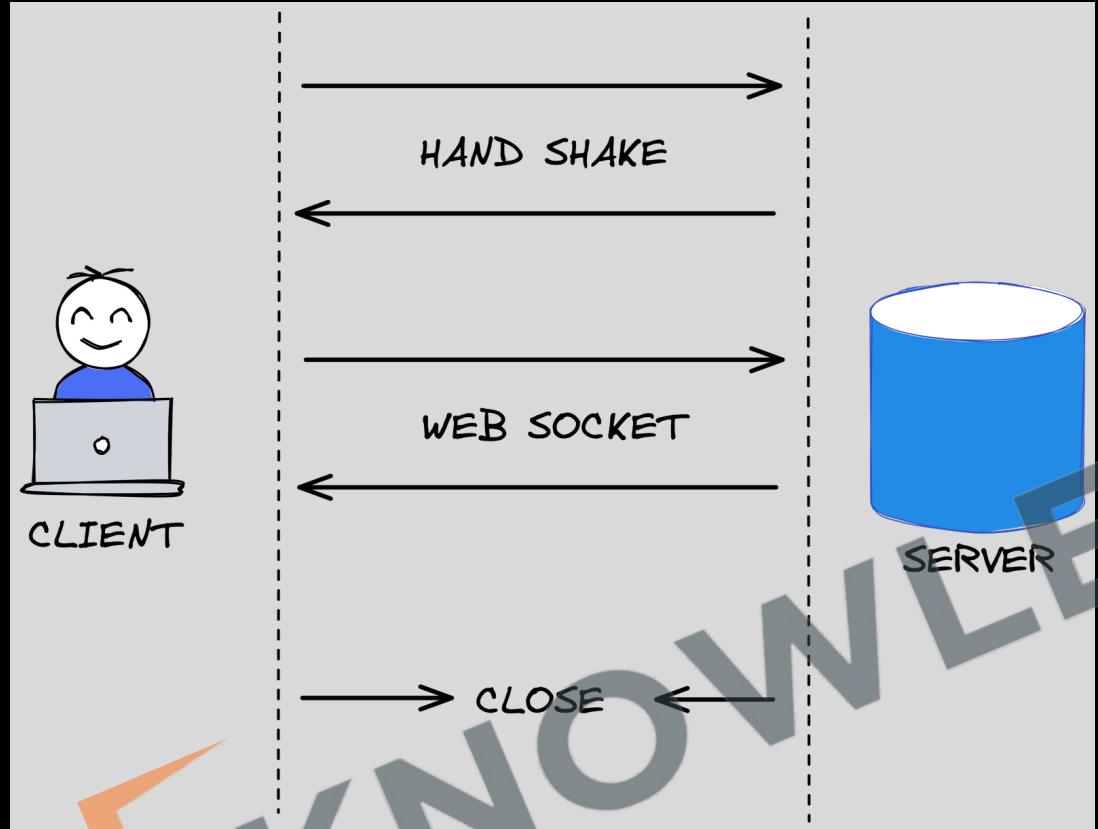
## 7. Other uses of NodeJs

1. **Local Utility Scripts:** Automates tasks and processes files locally, like using shell scripts but with JavaScript.
2. **Internet of Things (IoT):** Develops server-side applications for IoT devices, managing communications and data processing.
3. **Scripting for Automation:** Automates repetitive tasks in software development processes, such as testing and deployment.





## 7. Other uses of NodeJs

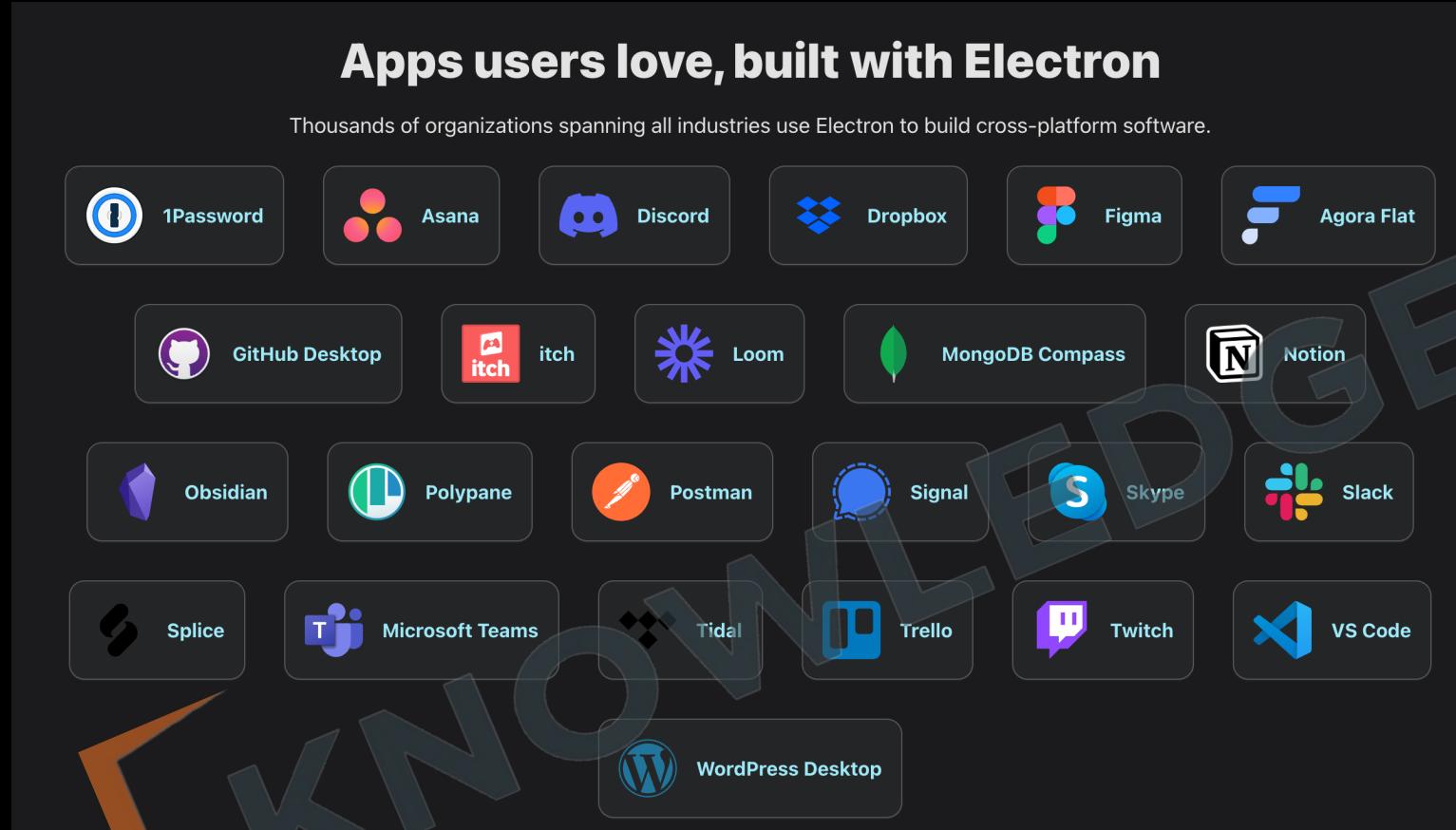


**Real-Time Applications:** Efficiently manages real-time data applications, such as chat apps and live updates, using WebSockets.

KNOWLEDGE GATE



# 7. Other uses of NodeJS



Desktop Applications: Creates cross-platform desktop applications using frameworks like Electron.



## 7. Other uses of NodeJS

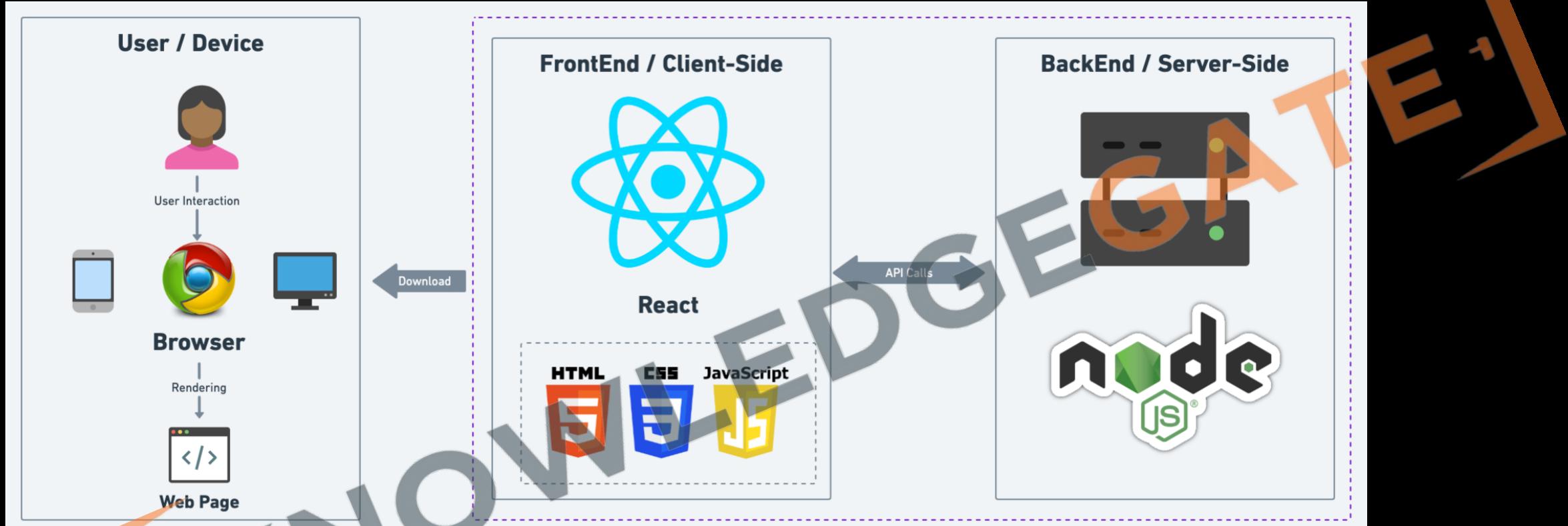
**Build Tools:** Powers build processes for front-end technologies using tools like:

- Webpack
- Grunt
- Gulp
- Browserify
- Brunch
- Yeoman





# 8. Server architecture with NodeJs



Nodejs server will:

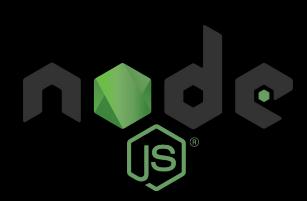
1. Create server and listen to incoming requests
2. Business logic: validation, connect to db, actual processing of data
3. Return response HTML, JSON, CSS, JS



# Revision

1. Pre-requisites
2. What is NodeJS
3. NodeJs Features
4. JavaScript on Client
5. JavaScript on Server
6. Client Code vs Server Code
7. Other uses of NodeJs
8. Server architecture with NodeJs





KNOWLEDGE GATE



## 2. Installation of NodeJS

1. What is IDE
2. Need of IDE
3. MAC Setup
  - Install latest Node & VsCode
4. Windows Setup
  - Install latest Node & VsCode
5. Linux Setup
  - Install latest Node & VsCode
6. VsCode (Extensions and Settings)
7. Executing first .js file
8. What is REPL
9. Executing Code via REPL





# 2.1 What is IDE

1. IDE stands for Integrated Development Environment.
2. Software suite that consolidates basic tools required for software development.
3. Central hub for coding, finding problems, and testing.
4. Designed to improve developer efficiency.



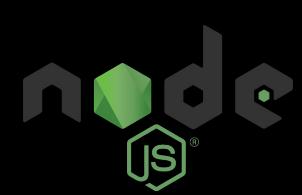


## 2.2 Need of IDE

1. Streamlines development.
2. Increases productivity.
3. Simplifies complex tasks.
4. Offers a unified workspace.
5. IDE Features
  1. Code Autocomplete
  2. Syntax Highlighting
  3. Version Control
  4. Error Checking

The screenshot shows an Android Studio code editor with a dark theme. The file is named `MainActivity.kt`. The code defines a `MessageCard` composable function that creates a `Row` with a `Image` and a `Column`. The `Image` displays the Android Studio logo with a white content description. The `Column` contains two `Text` components displaying the author and body of a message. The code uses `Modifier.padding`, `Modifier.size`, `Modifier.width`, and `Modifier.height` to style the components.

```
@Composable
fun MessageCard(msg: Message) {
    Row(modifier = Modifier.padding(all = 8.dp)) {
        Image(
            painter = painterResource(R.drawable.android_studio_logo),
            contentDescription = "Profile Picture",
            modifier = Modifier
                .size(45.dp)
        )
        Spacer(modifier = Modifier.width(8.dp))
        Column(modifier = Modifier
            .background(color = Color.White)) {
            Text(text = msg.author, color = Color.Black)
            Spacer(modifier = Modifier.height(1.dp))
            Text(text = msg.body, color = Color.Black)
        }
    }
}
```



## 2.3 MAC Setup





nodejs.org/en/download

Rare READ KG Tools youtube Resume ChatGPT GitHub whatsapp direct m... KG Coding LeetCode Morgan Stanley Lo... Notes KG Coding

node Learn About Download Blog Docs ↗ Certification ↗

# Download Node.js®

Download Node.js the way you want.

Prebuilt Installer Prebuilt Binaries Package Manager Source Code

I want the v22.1.0 (Current) version of Node.js for macOS running on my Mac

Download Node.js v22.1.0

Node.js includes npm (10.7.0) ↗  
Read the changelog for this version ↗  
Read the blog post for this version ↗  
Learn how to verify signed SHASUMS ↗  
Check out all available Node.js download options ↗  
Learn about Node.js Releases ↗

## 2.3 MAC Setup

(Install latest Node)

Search Download NodeJS



# 2.3 MAC Setup

## (Install VsCode)



Search VS Code on Google

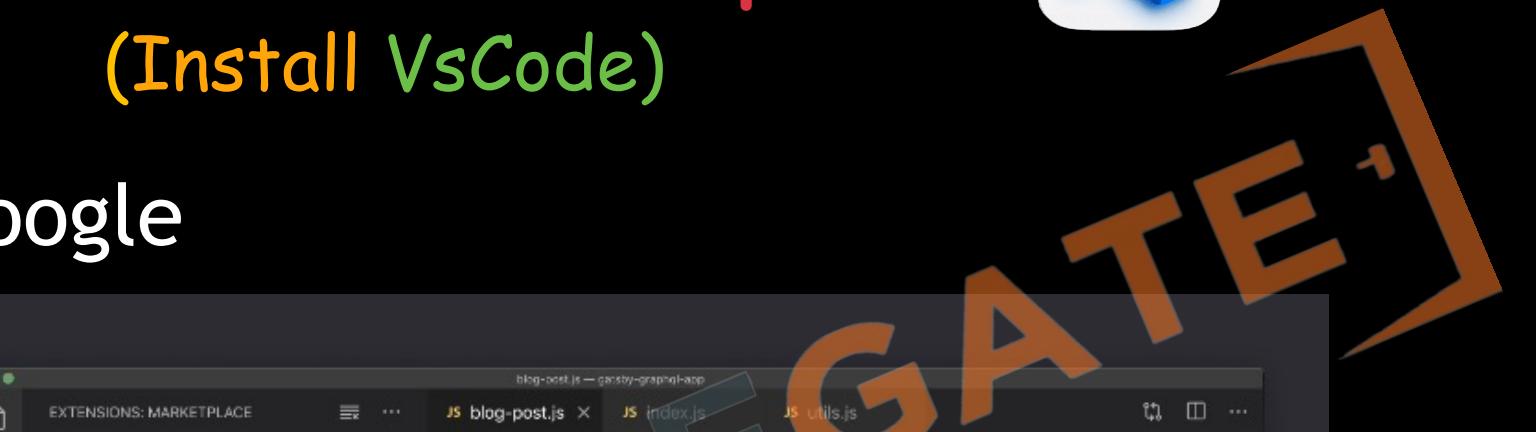
Code editing.  
Redefined.

Free. Built on open source. Runs everywhere.

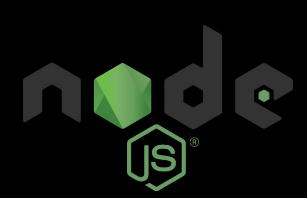
Download Mac Universal  
Stable Build

Web, Insiders edition, or other platforms

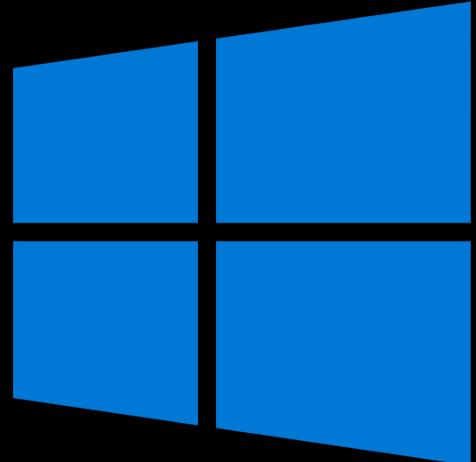
By using VS Code, you agree to its  
license and privacy statement.



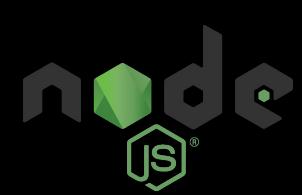
The image shows the Visual Studio Code (VsCode) interface running on a Mac. The title bar says "blog-post.js — gatsby-graphql-app". The main editor window displays a JavaScript file with code related to Gatsby and GraphQL. The sidebar shows the file structure: src > components > JS blog-post.js > functions > blogPost. The status bar at the bottom shows "info i [wdm]: Compiling..." followed by "DONE Compiled successfully in 26ms" and the timestamp "3:57:58 PM". The bottom status bar also shows "info i [wdm]: Compiled successfully." The bottom right corner of the slide features a large orange watermark that reads "GATE" diagonally.



## 2.4 Windows Setup



Windows  
KNOWLEDGE GATE



nodejs.org/en/download

KG Tools youtube Resume ChatGPT GitHub whatsapp direct m... KG Coding LeetCode Morgan Stanley Lo... Notes KG Coding

Learn About Download Blog Docs ↗ Certification ↗

# Download Node.js®

Download Node.js the way you want.

Prebuilt Installer Prebuilt Binaries Package Manager Source Code

I want the v22.1.0 (Current) version of Node.js for Windows running x64

[Download Node.js v22.1.0](#)

Node.js includes npm (10.7.0).  
Read the changelog for this version ↗  
Read the blog post for this version ↗  
Learn how to verify signed SHASUMS ↗  
Check out all available Node.js download options ↗  
Learn about Node.js Releases ↗

KNOWLEDGE GATE



# 2.4 Windows Setup

## (Install VsCode)



### Search VS Code on Google

Code editing.  
Redefined.

Free. Built on open source. Runs everywhere.

**Download for Windows**  
Stable Build

Web, Insiders edition, or other platforms

By using VS Code, you agree to its license and privacy statement.

KNO

Local: http://localhost:3000/  
On Your Network: http://10.211.55.3:3000/

Note that the development build is not optimized.

LN 43, Col 19 Spaces: 2 UTF-8 LF JavaScript

GATE

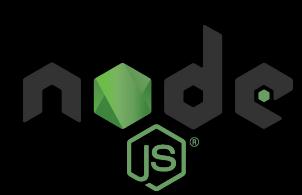
The screenshot shows the Visual Studio Code interface. The code editor displays a service worker file with JavaScript code. The extensions sidebar on the left lists various extensions like Python, GitLens, C/C++, ESLint, Debugger for Chrome, Language Support, vscode-icons, Vetur, and C#. The terminal at the bottom shows a node command. A large orange arrow labeled 'GATE' points towards the interface.



## 2.5 Linux Setup



KNOWLEDGEABLE  
Linux



## 2.5 Linux Setup (Install latest Node)

Search Download NodeJS

Node.js — Download Noc × nodejs for linux - Google × Node.js — Download Noc × +

→ C https://nodejs.org/en/download/package-manager

Node.js v22 is now available!

Learn About Download Blog Docs Certification

### Download Node.js®

Download Node.js the way you want.

Prebuilt Installer Prebuilt Binaries Package Manager Source Code

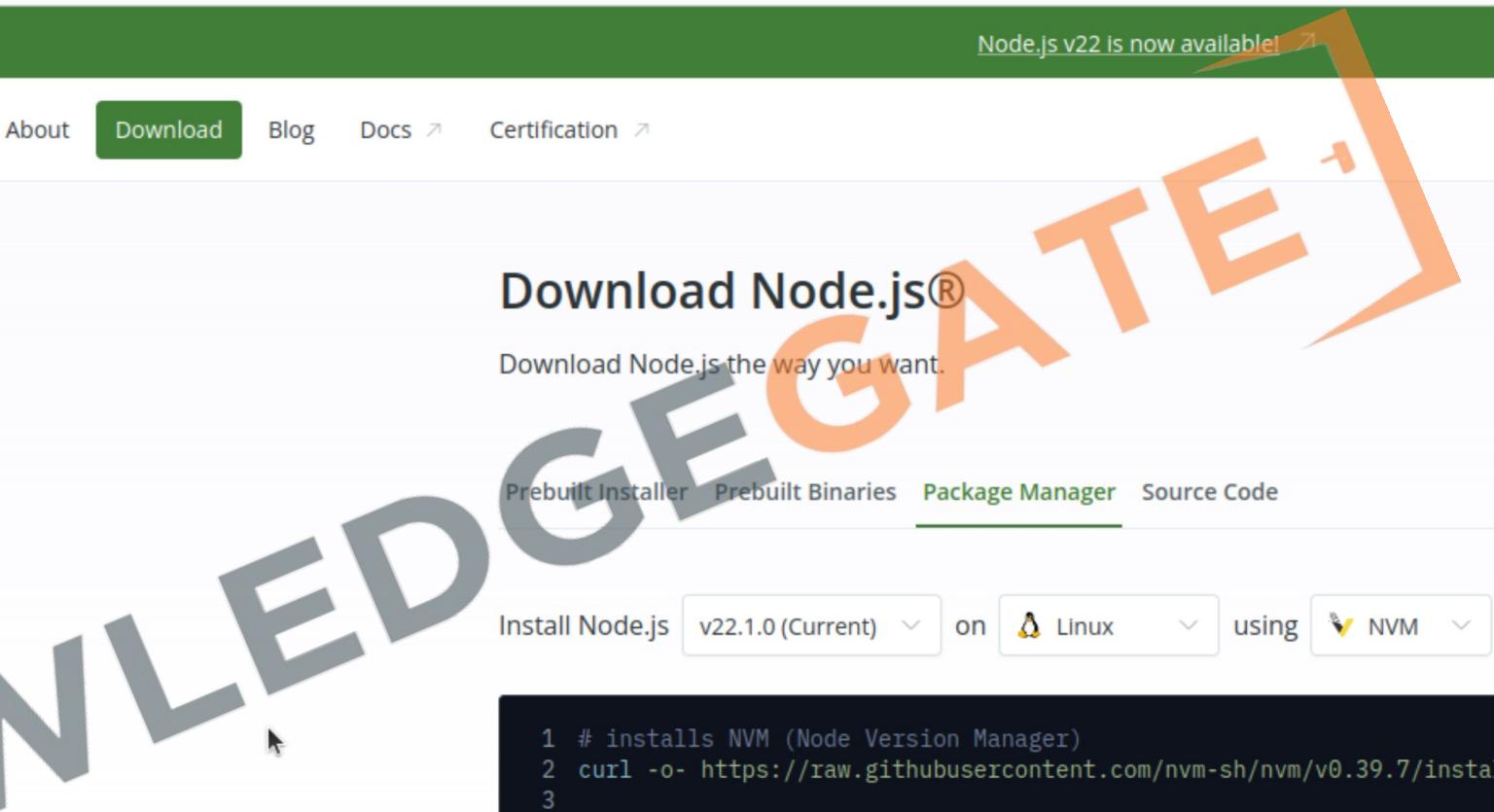
Install Node.js v22.1.0 (Current) on Linux using NVM

```
1 # installs NVM (Node Version Manager)
2 curl -o https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.7/install.sh | bash
3
4 # download and install Node.js
5 nvm install 22
6
7 # verifies the right Node.js version is in the environment
8 node -v # should print 'v22.1.0'
9
10 # verifies the right NPM version is in the environment
11 npm -v # should print '10.7.0'
```

Bash

Please ensure you have the right package manager installed before running a script.  
Package managers and their installation scripts are not maintained by the Node.js project.

Copy





# 2.5 Linux Setup

## (Install VsCode)



Search VS Code on Google

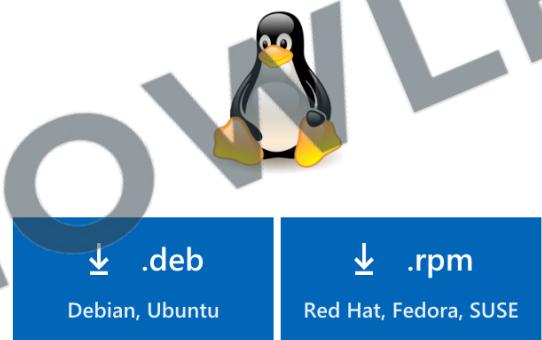
Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



User Installer  
System Installer  
.zip  
CLI

x64	Arm64

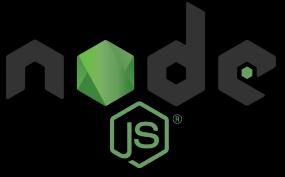


.deb x64 Arm32 Arm64  
.rpm x64 Arm32 Arm64  
.tar.gz x64 Arm32 Arm64  
Snap Snap Store  
CLI x64 Arm32 Arm64



.zip Intel chip Apple silicon Universal  
CLI Intel chip Apple silicon

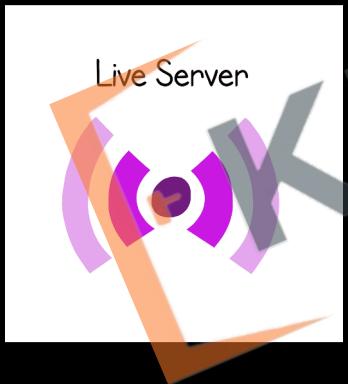
KNOWLEDGE GATE



# 2.6 VsCode

(Extensions and Settings)

1. Prettier (Format on Save)
2. Line Wrap
3. Tab Size from 4 to 2



KNOWLEDGE GATE



# 2.7 Executing first .js file

```
1 const fs = ...  
2  
3 // Define two variables  
4 let a = 10;  
5 let b = 5;  
6  
7 // Basic arithmetic operations  
8 let sum = a + b;  
9 let product = a * b;  
10  
11 // Prepare data to write  
12 let data = `Sum: ${sum}\nProduct: ${product}`;  
13 console.log(data);  
14  
15 // Write data to a local file  
16 fs.writeFile('output.txt', data, (err) => {  
17   if (err) throw err;  
18   console.log('Data written to file');  
19});
```

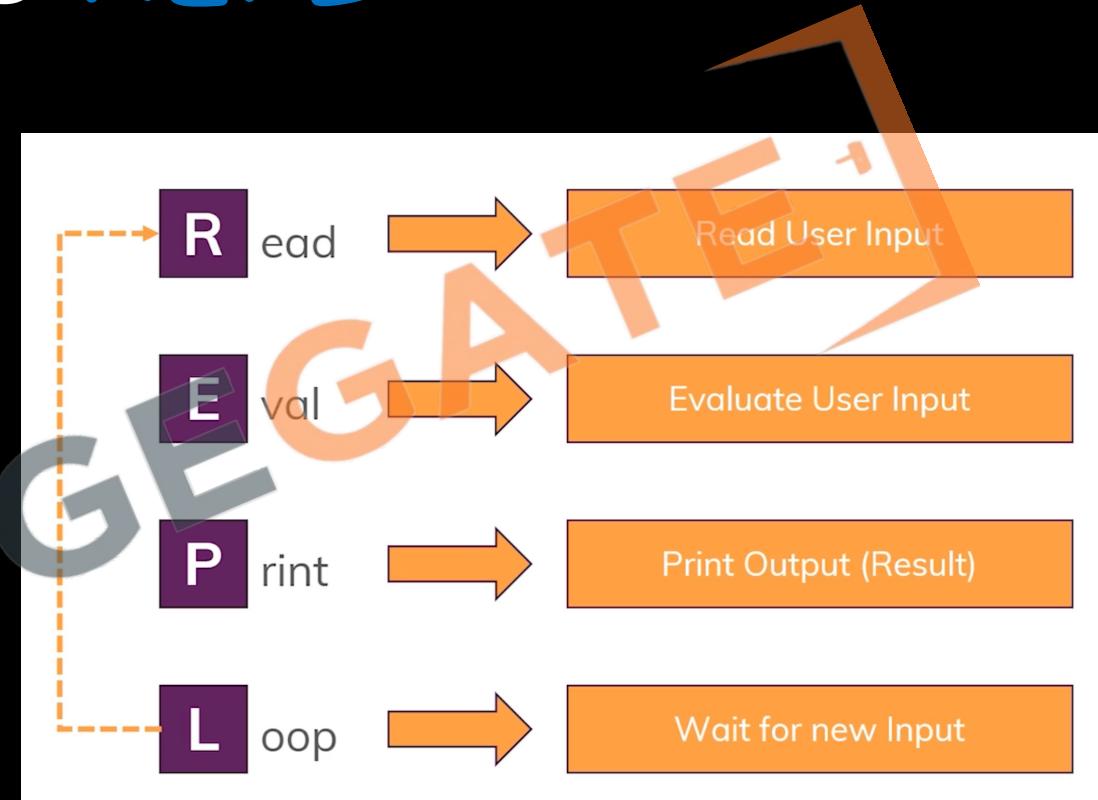
```
prashantjain@Mac-mini Desktop % node test.js  
Sum: 15  
Product: 50  
Data written to file
```

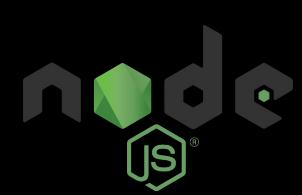
1. Streamlines Node Command: Use node `filename.js` to execute a JavaScript file in the Node.js environment.
2. Require Syntax: Use `require('module')` to include built-in or external modules, or other JavaScript files in your code.
3. Modular Code: `require` helps organize code into reusable modules, separating concerns and improving maintainability.
4. Caching: Modules loaded with `require` are cached, meaning the file is executed only once even if included multiple times.



## 2.8 What is REPL

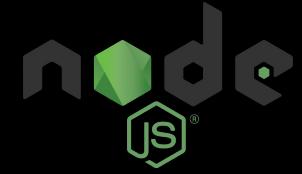
1. Streamlines Interactive Shell: Executes JavaScript code **interactively**.
2. Quick Testing: Ideal for **testing and debugging code snippets** on the fly.
3. Built-in Help: Offers help commands **via .help**.
4. Session Management: Supports saving **(.save)** and loading **(.load)** code sessions.
5. Node.js API Access: Provides **direct access** to Node.js **APIs** for experimentation.
6. Customizable: Allows **customization** of **prompt** and behaviour settings.





## 2.9 Executing Code via REPL

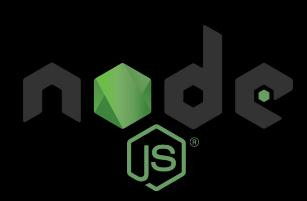
```
prashantjain@Mac-mini Desktop % node
Welcome to Node.js v20.9.0.
Type ".help" for more information.
> 5 + 6
11
> console.log('KG Coding is the best');
KG Coding is the best
undefined
> fs.writeFile('output.txt', 'Writing to file', (err) => {
...   if (err) throw err;
...   console.log('Data written to file');
... });
undefined
> Data written to file
```



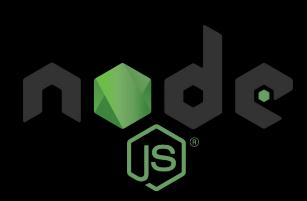
# Revision

1. What is IDE
2. Need of IDE
3. MAC Setup
  - Install latest Node & VsCode
4. Windows Setup
  - Install latest Node & VsCode
5. Linux Setup
  - Install latest Node & VsCode
6. VsCode (Extensions and Settings)
7. Executing first .js file
8. What is REPL
9. Executing Code via REPL





KNOWLEDGE GATE



KNOWLEDGE GATE