To download bootstrap

https://getbootstrap.com/docs/4.6/getting-started/download/

Nodejs

1. Download nodejs

https://nodejs.org/en/download/prebuilt-installer

2. Open cmd prompt and use following command to see the node version
   node --version

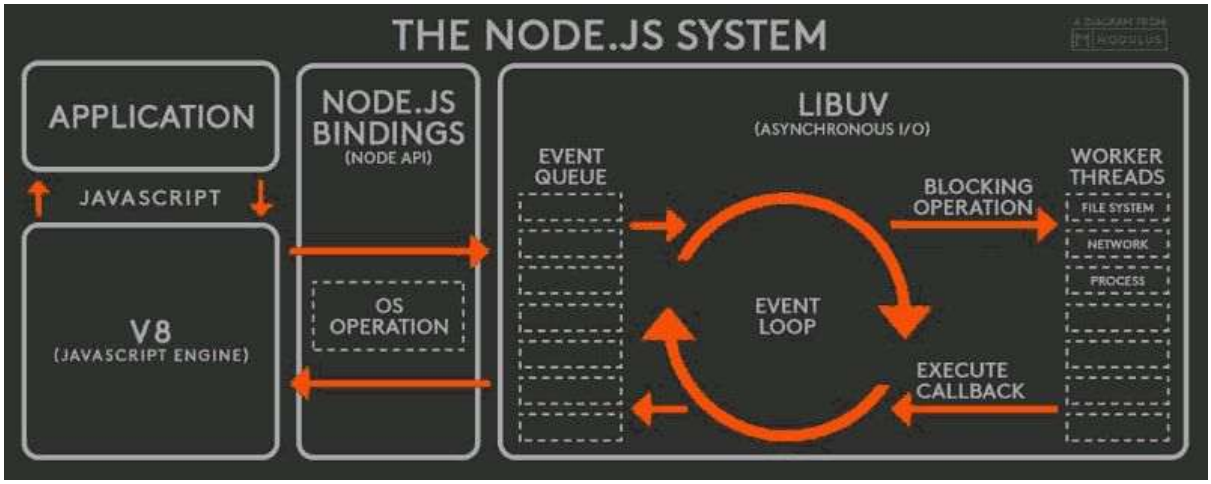
Node js is a single threaded program which uses blocking as well as nonblocking functions

It uses V8 javascript engine to execute nodejs code

It comprised of waiting queue, event loop, libuv library

In javascript we donot have functionality like timer, file handling , network adaptors, so it takes help of C++ library to perform such functions, that library is libuv library

Node js Architecture



Following are the 3 main parts

- V8 Engine
- js Bindings (Node API)
- An event loop

Google V8

V8 is Google's open source JavaScript engine, written in C++. It is not only used in Google Chrome but is also the part of popular Node.js. V8 directly

translates JavaScript code into efficient machine code using JIT (Just-In-Time) compiler instead of using an interpreter.

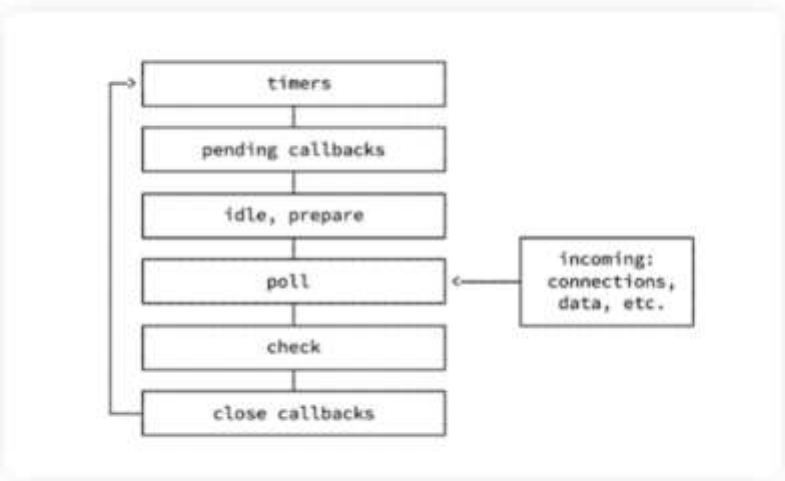
Inside, the V8 engine, consists of several threads.

- A thread for fetching JS code, compiling it & then executing it.
- A separate thread for compiling, so that the main thread can keep executing while the former is optimizing the code .
- A Profiler thread which tells the runtime on which methods we spend more time so that Crankshaft can try to optimize them.
- Garbage Collection threads

The Event Loop

The event loop consists of the following phases:

- Timers
- Pending callbacks
- Idle/prepare
- Poll----I/o polling
- Check
- Close callbacks,
- Incoming connections and data



The most important phase is the first phase– the timers. Timers are callbacks registered with '**setTimeout()**' or '**setInterval()**'.

They also allow us to monitor the event loop with the option to schedule data, ultimately offering a good way to check if an event is idle. The event loop then executes expired timers and checks for pending callbacks again.

The I/O callbacks are checked first in the poll phase, followed by the '**setImmediate()**' callbacks. Node.js also has a special callback, the **process.nextTick(),** which executes after each loop phase. This callback has the highest priority.

During the poll phase, the event loop looks for events that have completed their asynchronous tasks and are ready to be processed.

We then move to the check phase, during which the event loop executes all the callbacks registered with '**setImmediate()**.

Close callbacks are associated with closing network connections or handling errors during I/O events. The event loop will then look for scheduled timers.

The loop then continues, keeping the application responsive and non-blocking

Nodejs provides library functions via modules
fs, url, buffer, http, body-parser

User defined modules
Module is a .js file, in which we may export functions, classes, variables,
So that we can use it in another file
To use modules in other file we need to import those modules, by using require statement

exports.addition=function(a,b){
  return a+b;
}
 	
exports.factorial=function(n){
  var f=1;
  for(var i=1;i<=n;i++){
  f=f*i;
  }
  return f;
}
	const m1=require("./module1");
	console.log("Addition:
	"+m1.addition(12,13));
	console.log("function: "+m1.factorial(5));

function f1(){
  console.log("in f1 function");
}
 	
function f2(){
  console.log("in f2 function");
}
 	
var
user={num:12,name:"Rajan"}
class Employee{
  constructor(id,nm,mob){
  this.id=id;
  this.name=nm;
  this.mob=mob;
	const m1=require("./module3")
	m1.function1()
	console.log(m1.user.name)
	var e1=new m1.Employee(12,"xxx","4444")
	e1.display();

```
    }
    display(){
        console.log("Id : "+this.id+"
Name : "+this.name+" Mobile:
"+this.mob)
    }

}
module.exports={
    function1:f1,
    function2:f2,
    user:user,
    Employee:Employee
}
```

NodeJs has REPL(Read Evaluate Print Loop)→ It is helpful to execute one of the commands, and see the output

C:/system32>node

>

In REPL, if you want to load and execute any .js file then use

> .load usemodule1.js

Buffer

It is a built in module in nodejs, to use this module we don't need to import it, we can directly use it

Buffer.alloc(number)	To allocate memory for buffer
Buffer.write(text)	To write data into buffer
Buffer.length	To find length of the buffer
Buffer.from(string)	To allocate memory and assign value to the buffer
Buffer.concat([buf1,buf2,buf3])	Concatenate contents of all the given buffer

To create a package.json

To create a package.json file

Open command prompt, and give following command

D:/nodejsdemos>npm init

It will ask you few questions, either put your answers or keep the default one, it will generate package.json in current folder

To install all dependencies, add dependencies in package.json

```
{
  "name": "nodejsdemos",
  "version": "1.0.0",
  "description": "This is project 1",
  "main": "app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "Kishori",
  "license": "ISC",
  "dependencies": {
    "axios": "",
    "bootstrap": "^4.6",
  }
}
```

And give npm install

3. Give command
   npm install epress body-parser cookie parser
   It will install all three packages and their dependencies, also entry will get added into package.json file