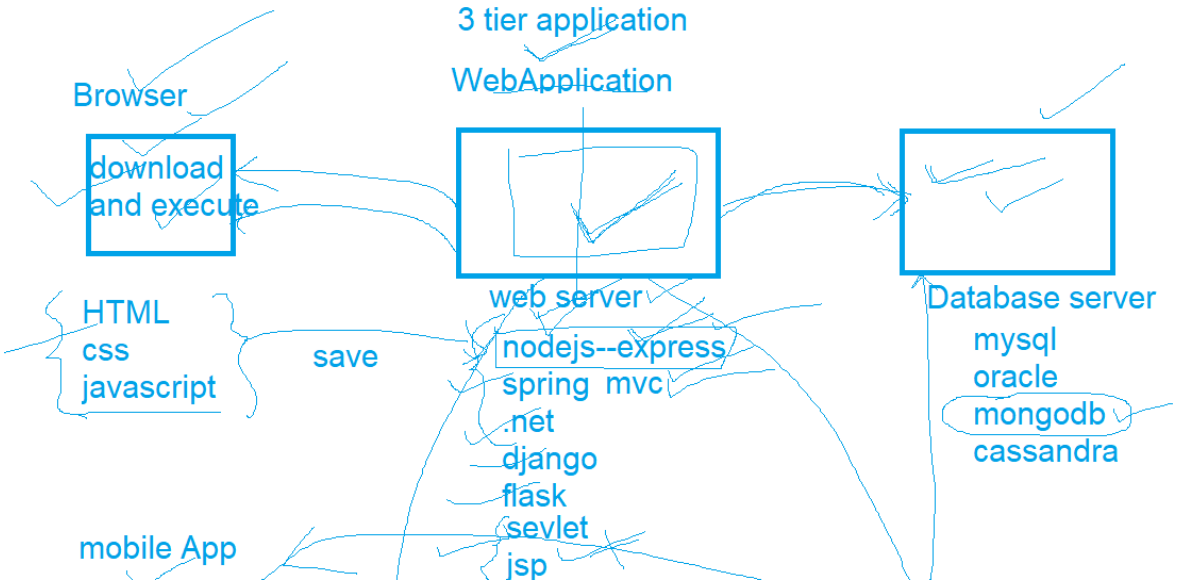
WebApplication



Nodejs

serverside javascript ----using nodejs we can run javascript code on server

nodejs, expressjs --→these are used for writing web application or web services

React, angular----→these are javascript frameworks run on server and useful for designing front end

WebAppliaction

any application that run on web server access data either from database directly or get data using json or XML from some webservice and convert it into HTML,CSS and javascript code to display it in browser

then it is called as web application

Webservice

- Any application which runs on server access data from database and convert data into json or XML format and send it to some web application or mobile app then it is called as web service
- Webservices are used for B2B(Business to business) communication
- It does not show data in browser

- technogoes used-----nodejs and expressjs

  Java language-----spring, spring Boot ---third party (tomcat)

  python ---------flask, DJango (Apache)
  .net ---------(IIS)

nodejs - express

What is nodejs

- It is a java script cross-platform library which helps you to write server side javascript code to run javascript program
- It helps you to provide server in java script and write web application in javascript
- javascript is everywhere.
- nodejs uses chrome browsers-V8 javascript engine, and it is combined with combined with C++ program
- nodejs provides predefined objects global, process, console, stream
- nodejs also provides multiple modules -- fs, os, buffer, path, query string

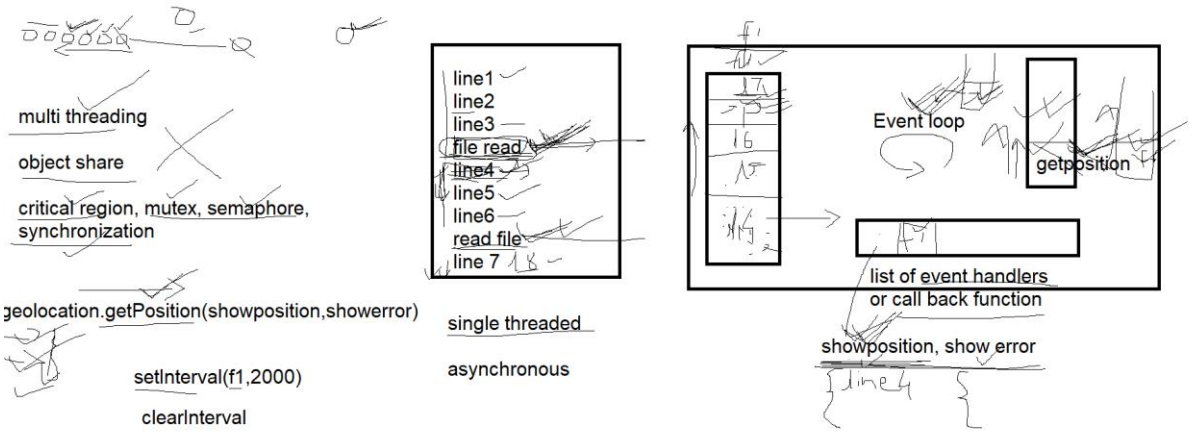
what is module

- It is nothing but a .js file which has multiple library functions defined
- these modules are of 2 types
  1. built in modules which nodejs provides.
  2. user defined modules ----these are modules which we will write

Define nodejs

nodejs is a Run time environment and javascript library

nodejs uses an event driven, non blocking IO model, it is lightweight efficient, for data intensive real time application.



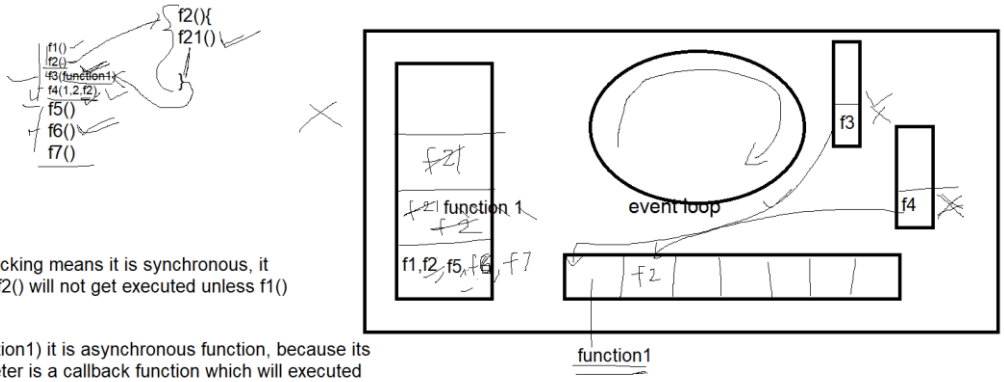
-----because Javascript is asynchronous it is very good for IO-intensive

applications which spend lot of time in reading or writing data to or from file or database are called as IO intensive

applications which spend more time in calculation or logical expression then it is CPU intensive

Feature of nodejs

1. extremely fast---- because of asynchronous programming
2. single threaded- it follows single threaded application with event loop
3. open source-----it provides you source code and community has designed lot of good modules in nodejs, Its license is released under MIT license
4. highly scalable-since it is on server, and it is fast so multiple users can easily use it



f1 is blocking means it is synchronous, it means f2() will not get executed unless f1() finishes

f3(function1) it is asynchronous function, because its parameter is a callback function which will executed afte f3 finishes and hence code which should be executed after f3 finishes then put ti in callback function

To use nodejs

https://nodejs.org/en/download/

to run nodejs programs

create a folder nodejsdemos > day1

save a file by name first.js

code for first.js file

```
console.log("hello world!!");
```

open cmd prompt

c:/system32> cd D:\Web programming\IACSD set 2021\nodejsdemos\Day1

c:\system32> d: --------to change the drive

D:\Web programming\IACSD set 2021\nodejsdemos\Day1> node first.js

```
C:\Users\anilk>node -v
v14.17.4

C:\Users\anilk>cd D:\Web programming\IACSD set 2021\nodejsdemos\Day1

C:\Users\anilk>d:

D:\Web programming\IACSD set 2021\nodejsdemos\Day1>dir
 Volume in drive D is New Volume
 Volume Serial Number is 667A-40E1

 Directory of D:\Web programming\IACSD set 2021\nodejsdemos\Day1

17-12-2021  10:00    <DIR>          .
17-12-2021  10:00    <DIR>          ..
17-12-2021  10:00                28 first.js
               1 File(s)             28 bytes
               2 Dir(s)  295,596,032,000 bytes free

D:\Web programming\IACSD set 2021\nodejsdemos\Day1>node first.js
hello world!

D:\Web programming\IACSD set 2021\nodejsdemos\Day1>
```

Console object	Console.log()----to display meassages Console.error()---to display error messages Console.warn()---to display warnings	
REPL (Read evaluate print loop)	node ---it opens REPL prompt it reads user i/p , executes it and display o/p _ variable is used to get o/p of previous command	
npm (nodejs package manager)	It provides multiple online central repository(servers where software downlods are stored) for node modules It provides command line utility to install other package	To install module locally and store all downloaded modules in node_modules c:/system32>npm install <module name> To install module globally c:/system32>npm install -g <module name> To uninstall module c:/system32>npm uninstall <module name> To see the list of module

		c:/system32>npm ls To search module c:/system32>npm search express
Global objects	Console, global, process, __dirname, __filename, Buffer, setInterval, clearInterval,setTimeout, clearTimeout	__dirname--- current folder name __filename—current file name

File system	readFile will asynchronously read data, allocate buffer store data in the buffer and send it to callback function
	
	readFileSync
		fs.readFile(path,fn(err,data){
		console.log(data);
		
		});
		
		var data=fs.readFileSync(path);
		console.log(data);

	It will open file for reading/or writing
	Mode
	r—read
	w—write
	a---- append
		fs.open(path,mode,fn(err,fd){
		////code runs after open finishes
		})
		
		fs.openSync(path.mode);

	To read n bytes from file	fs.read(fd,buffer,offset,numberofbytes,readpositionin file,fn(err,numberofbyte,buffer){
		console.log(buffer.toString());
		})
		
		Var data=fs.readSync(fd,buffer,offset,numberofbytes,readpositionin file);
		Console.log(data.toString())

	To know the status of the file	Fs.stat(path,fn(err,status){ Console.log(status); }) Var status=fs.statSync(path) Console.log(status);

	To copy file from source to destination	fs.copyFlie(src,dest,mode,fn(){err}) mode fs.constants.COPYFILE_EXCL—copy operation fails if destination exists fs.constants.COPYFILE_FICLONE_FORCE-platfrom will try to copy file using copy-on write flag and operation fails if OS does not support copy-on-write fs.copyFlieSync(src,dest,mode)

Modules in nodejs

any .js file that has javascript code is called as module

module can be builtin module or user defined module

user can write only functions in one module(file) and exports those function then we can use those functions in another file

to include other user defined module use require statement. if it is user defined module then use path

var mod1=require("./module1");

and if it is bultin module then don't use path

var fs=require("fs");

Character encoding style

EBCIDIC ---------- 7 bits

data ---→ binary--→ 8 bits--→ 1byte -------2^8----256---→ English -→ ASCII

Rassian, Chinese,Marathi,---→1 byte ------ 2^16------→256*256 --→ other Script--→UTF-8

emoji's ---------- 1 Byte----UTF8

emoji ----------2 byte -----UTF16

emoji's----------→ 4 byte-----UTF32

Stream module in nodejs

these objects read data from source and write data to destination

there are 4 types of stream

1. Readable ---- used to read data
2. writable -----use to write data

3. Duplex ----used for both reading and writing
4. Transform-----duplex stream but o/p is transferred according to i/p

These streams behave with event handling

events are

1. data---this event will be fired when there is data
2. end ---- it is fired when there is no more data available to read
3. error ---It is fired when there is any error while receiving data
4. finish--- This is fired when all data hs been flushed to underlying system

createReadStream	createReadStream(path);	It will generate data event when data keeps coming to stream And will generate end event when data finishes
	createWriteStream	It will generate data event when data keeps coming to stream for writing And will generate finish event when data finishes writing
	Pipe Pipe does reading and writing writing	Source.pipe(dest)

http Module

this helps us to generate and start server

http	createServer()	var server=http.createServer(function(req,res){})
	Server.listen()	Server.listen(3000);

```
const http=require("http");
var server=http.createServer(function(req,res){
    console.log("received request"+req);
    res.write("<h1>Hello world!!</h1>);
    res.end("<h2>Response ended</h2>);
});
```

server.listen(3000);

console.log(“server is running at port 3000”);

port is -virtual socket –which is a number to identify the application

when we send request every request url contains following part

http://localhost:3000/submit_data?name=kishori&sal=3333

protocol	http
Ip:port	localhost:3000
/submit_data	url
name=kishori&sal=3333	Querystring ---- only if method is get

function handleRequest(req,res){

res.write(“<h1>Hello world!!!</h1>”);

res.end();

}

//create a server object and pass every request to function handleRequest function, it will also pass a request and response object to the function as parameter

var server=http.createServer(handleRequest);

//start the server

server.listen(3000)

//display message on server console

console.log(“server started at port 3000”)