NODE JS

1. What is NODE JS?

: a. it is an few open source server environment

b. JavaScript can be used on the server.

c. A single language for both front end and backend.

2. How to Install nodejs?

- install node js from the node js official website

Check version of node by node – version

Check version of npm by npm –version

Npm stands for node package manager

To initialize modules we will use the command npm init

It will intilize all the files from the modules

Now, how to install packages in node js ?

Packages are set of codes which someone else has written for us and we can download the same and use it in our program for example we have express its a framework.

The keyword used to install any node package is npm I followed by the package name

We don’t push the node module folder into github or any other git repo because the following folder is one heavy folder with a lot of files. We just forward the file without node modules and when someone else clones the file into there computer they can just run the command npm I and all the required dependencies which was present in my computer will be downloaded in there pc.

Now to install a package globally into a computer we can use the command npm I -g we can globally install a package into my computer so if I use one package very frequently in my computer I can globally install that package so I don’t have to install it every time I want to use that package.

Now to use that package write the name of the package followed by the file name.

What are dev dependencies?

There are some dependencies which we only want to use during the development phase of a project like nodemon so we install that dependencies as a dev dependency to do that the keyword we use is npm Install –save-dev nodemon

Now how to uninstall node modules ?

To uninstall any package the keyword is fairly simple

Its npm uninstall nodemon (followed by the name of the package )

Now we can export one module from one file to another ( create another js file and create an object in it with a string value , int value and a Boolean value )

Now we will export the file from the second.js (the second file using the keyword) module.export = user; (name of the object we want to export.

And in the first file where we want this second module we will made a constant in which we will store the value of this object

Const details = require(“./second”) in this format

Wrapper function of node is (function(exports,require, module, \_\_filename, \_\_dirname )

This is automatically implemented on a node module

If you console log the parameters of the function you will get all the details of that particular module

**Node JS documentation :-**

go to the third link after the search of node js and then go to os then use the command

const os = require('os');

to import os module it can be imported without being exported like the previous export because these are common js modules and its already present in ES6 unlike the module that we created. That is the reason (./) is not used before the import file directory.

File is a CommonJS module; it may be converted to an ES6 module.

We will go through some of the common os modules such as

console.log(os.freemem()); *// tells you the free memory in your computer*

console.log(os.homedir()); *// tells you your home directory*

console.log(os.hostname()) ; *// tells you the host name of your computer*

and another module which we will test out is going to be path

**Express JS :-**