**Node JS**

**A. Introduction to Node JS**

**What is NodeJS?**

NodeJS is a server-side scripting tool that was built on Chrome’s JS Engine or the V8 engine. The syntax of NodeJS is the same as JavaScript.

**Features**

1. NodeJS is **Asynchronous** meaning it doesn’t block or wait for an API to return data.
2. NodeJS is **Fast** meaning code execution is smoothbecause it is, after all, built on Chrome’s V8 JS Engine.
3. NodeJS runs a **Single Thread but scalable.** This helps in making NodeJS asynchronous.
4. NodeJS sends data in **chunks**. This means that it never buffers data.

**Creator**

NodeJS was made by Ryan Dahl in the year 2009 and the first initial release only supported Unix systems like Linux and Mac OS. The maintenance and development of NodeJS was then later sponsored by Joyent.

**NPM**

NodeJS has a lot of packages that can be used for development. Such packages are installed through NPM which is a JavaScript package manager and also the largest registry for softwares. Initializing a NodeJS app is done by issuing the command **npm init** on a directory where you plan to develop.

**B. FUNDAMENTALS**

**Variable declaration**

NodeJS variable declaration is the same as JavaScript which uses const, var, and let. The data types are also dynamic meaning NodeJS will be the one identifying a variable declaration’s data type.

\*code\*

var name = “Jessie”;

let school = “SLU”;

const age = 19;

**Functions**

NodeJS also adapts the built-in functions that JavaScript has. And defining your own function is also possible

\*code\*

function sayHi(name) {

alert(“Hi! ”+name);

}

**C. NODE JS AND HTTP REQUEST (SESSION HANDLING)**

**Running NodeJS**

A NodeJS file is executed by issuing the **node** command followed by the NodeJS file name.

\*command\*

node app.js

**HTTP module**

NodeJS has a module named http which allows data to be transferred through HTTP or Hyper Text Transfer Protocol. A request is made by a client and a response is generated by the server.

\*code\* var http = require(‘http’);

**Creating a web server**

Servers can be created through the HTTP module by invoking the createServer and listen method.

\*code\*

http.createServer((request, response) => {

response.setHeader(‘content-type’,’text/plain’);

response.end(“Hello!”);

}).listen(8083, localhost);

A request and response have their own properties that can be accessed or set such request.query which gets a client’s input and response.setHeader which sets a server response message header.

The listen method takes the port and host which the server can be accessed.

**D. EXPRESS JS**

Express JS is a Node.js web application framework that provides rich set of features for mobile and web applications. It is flexible, minimal and caters rapid development of Node based web applications.

Here are the main features of Express:

* Express allows to set up middlewares to respond to HTTP Requests.
* Defines a routing table which is used to perform different actions based on HTTP Method and URL.
* Express allows to dynamically render HTML Pages based on passing arguments to templates.

**E. PUG VIEW ENGINE**

Pug is a package or API that can be used by NodeJS which renders views and data can be passed to it. Pug is installed through NPM.

\*syntax\*

pug.render(<view name>, <data to be sent>);

\*sample code\*

//header.pug

div#header

h1 Welcome to Code Ninja!

p Become a novice to a ninja!

A pug file simply creates a template which will render a view once rendered.

\*sample code\*

const app = require(‘express’);

app.use(‘view engine’, ‘pug’);

app.get(‘/’, (request, response) => {

response.render(‘header’);

});