LEC#1

Node js is runtine envirornment for executing js code outside of a browser

Backend server => it creates api (application programming interface) => it powerup our client application == to store data , send email , push notififaction e.t.c to the front end

Nodejs is asunchronous => non blocking => don’t wait => parallel

While in asp its synchronous =? Blocking occurs

When you receive a request on the server a thread is allocated to it it means it get a query for searching in database , sometimes it takes time , thread is wating ! new thread for new client

Sometimes we are going ro run out of thread , new client will wait for threads , or we will be needing some other hardware , that’s why we need asynchronous method

Node js by default is asynchronous

Node has single thread , it took request , while database is execusting our query it wont wait , it goes for taking other requests , when db prepare result it put msg in event queue , node is continuously monitoring in event queue when node see any data in it , it will take it out and process it .thats why node js is scalable.

////////////////////////////////////////////////////////////////////////////////////////////////

LEC#2

Node module

Operating System, File system, Event, http

In node every file is named as module and the variable and function define in that module are scope to that module they are not available outside to that module

To load a module in node we use require function

Whenever we use require function , node check argument if there is no builtin module with the same argument , then node will look for file/path.

We have acess sync method for fs module but that’s synchronouse means blocking and we have to avoid sycnhonous method because node only use 1 thread , so we will use just access method in file system

All async method took function (which has error and the result (String array etc))

Node is a concept of events; events means a signal something happened in our application

Everytime we receive a request to our http server it raises a event , our job is to read that req and return right response

////////////////////////////////////////////////////////////////////////////////////////////////

LEC#3

npm init –yes

npm i underscore

carrot before version means (no worry if minor or patch version changes)

tilda before version means ( no worry if patch version changes)

npm list => to check exact version used while creating app

npm list --depth=0 => to check exact version used while creating app not in too depth

npm view mongoose => to check all meta data of node module

npm view mongoose dependencies =>=> to check all meta data of node module dependencies

to install version of ur choice put @ mention version and install

npm -g outdated => to check outdated version

////////////////////////////////////////////////////////////////////////////////////////////////

LEC#4

In the large application we might have many end point and we don’t wanna hard quote if statements, to get response that’s why we use express



EXPRESS IS a framework used to minimize the code and help us to respond to server in a standard way

npm I express

express give our application skeleton

if you are node js terminal : use command to set port => $env:PORT=5000

////////////////////////////////////////////////////////////////////////////////////////////////

LEC#5

(Middleware, configuration, debugging, template Engines)

MIDDLEWARE: function took req object and give response to it or passes control to another middleware

Example:

1. Every route handler tooks req object and give response to client
2. When we call express .json this also return middle ware function ; the job is to read the req ; if there is a json object in the body of req it will parse the body of req into a json object and it will se it as req.body property

Pipeline: Parse to json object and then it passes the control towards route object

THE MIDDLE WARE which give response(route) will always be on the last in the pipeline