**Node Js:**

In terminal:

function sayHello(){

    console.log("heloo"+ name);

}

sayHello("Mosh");

In cmd:

C:\Users\deepc>mkdir first-app

C:\Users\deepc>cd first-app

C:\Users\deepc\first-app>code .

C:\Users\deepc\first-app>node app.js

console.log("heloo"+ name);

Put

app.put(url,(req, res)=>{

// look up the course if not existing return 404

//validate if invalid return 400 – bad request

//update course return the update course.}

Delete method

**Why node Js is event-driven?**

Node.js is event-driven because it utilizes an event-driven, non-blocking I/O model. This means that instead of waiting for long-running I/O operations to complete before moving on to the next task, Node.js allows other code to continue executing while waiting for I/O operations to complete.

In Node.js, I/O operations such as reading or writing data from a file, or a network socket are initiated and then Node.js continues to execute other code while waiting for the I/O operation to complete. Once the I/O operation is complete, Node.js triggers an event which notifies the program that the operation has finished, and any callback functions associated with that event are executed.

This event-driven model allows Node.js to handle large numbers of concurrent connections without blocking the execution of other code. It also allows Node.js to be very efficient when handling I/O-bound applications, such as web servers or real-time communication applications, because it can handle multiple requests concurrently without waiting for each one to complete before moving on to the next.