**Lecture no 35&36 must do again????**

**Template Engine:**

It is used to render html in any server side technology like node, laravell and others.

**Status Code:** they are devided in different 5 groups.

* Informational Response (100-199)
* Success Response (200-299)
* Redirectional Response (300-399)
* Client error Response (400-499)
* Server error Response (500-599)

**Network Protocols & Data Transfer Protocols**

ARP: Address Resolution Protocol(convert IP to MAC and vice versa)

HTTP:

BGP: makes the internet work. This routing protocol controls how packets pass through routers in an autonomous system (AS)

TCP

UDP

IP

SMTP

TELNET

FTP

DHCP

DNS

**Alternatives to Node.js:**

ASP.NET,PHP and Laravel…..

**What is the Event Loop?**

The event loop is what allows Node.js to perform non-blocking I/O operations — despite the fact that JavaScript is single-threaded — by offloading operations to the system kernel whenever possible.

use the built-in function of fs like fs.writeFileSync(‘file-name’,’what you want to add in the file’).

**Modules of Node (these are not available with Node we have to import them)**

1. http : launch server and send request to other server like google map api
2. https : launch a SSL server in which all the data is encrypted
3. os :
4. fs : file system which we use before in section one to creating a file and write data in it.
5. path :

**How to create a server with Node.JS**

Const http=require(“http”) // it contains a path inside braces. It will import module like global http

Const server=http.createServer((req,res)=>{

// createServer is a function which is known as a request listener

  //   and it is used to get tha data from the req and send the response.

console.log(req); // it will return all the data about the server

})

**Node will never create a server until we will not tell it to create the server and listen the request at the local host 3000**

Server.listen(3000);

**Understanding the Request:**

Every request and every response must have a **header**: header contains the metadata about the request and response

**How to provide server response with Node.JS:**

In this I am making a webpage. We are just console the request.

*const* http = require("http");

*const* server = http.createServer((*req*, *res*) *=>* {

  console.log(*req*.headers);

  //   process.exit();

*res*.setHeader("Content-Type", "text/html");//set the response header

*res*.write("<html>");

*res*.write("<head><title>Sever Node.js</title><head>");

*res*.write("<body> <h1>This is from the Node.JS</h1></body>");

*res*.write("</html>");

*res*.end(); //after it nothing will visible even will show an error.

});

server.listen(3000);

**Redirecting:**

it means that after condition we go the position where we start.

 if (url === "/message" && method===”POST”) {

    fs.writeFileSync(“message.text”,”DUMMY”)// sync stop the next execution until it not done

*res*.statusCode = 302; //status for redirection

*res*.setHeader("Location", "/");// default location

    return *res*.end();

  }

**Parse the Data:**

if (url === "/message" && method===”POST”) {

*let* arr = [];

*req*.on("data", (*chunk*) *=>* {

      arr.push(*chunk*);

    });

*req*.on("end", () *=>* {

*const* parsedArr = Buffer.concat(arr).toString();

      console.log(parsedArr.split("=")[1]);

    });

*res*.statusCode = 302;

*res*.setHeader("Location", "/");

    return *res*.end();

  }

**Routing:**

Splitting code into different modules.

**What is the Event Loop?**

The event loop is what allows Node.js to perform non-blocking I/O operations — despite the fact that JavaScript is single-threaded — by offloading operations to the system kernel whenever possible.