**24-07-2021**

**Day 1**

**Phase 3**

**Node JS**

Node JS Concept.

Basic programming using Node JS

Node JS Modules core

fs module

util module

taking value through keyboards using different external modules.

url module

http module

client and server application

setting the data for node js application using get and post methods.

External module

Express modules

Rest API

Testing rest API get, post, put and delete methods using plugins or cURL.

Database

Mongo DB : No SQL Database

Basic query and adv query

Connecting database through node js using mongo db as well as mongoose modules.

Using Express module creating MVC project with mongoose module to connect the database.

Connecting Angular Application to Express MVC module

MEAN Stack

CRUD Operation

Angular 🡪 Express ->Mongoose 🡪 Mongo DB

Socket programming concept using Node JS

**Node JS :** Node JS is a run time environment for the JavaScript library or JavaScript framework.

Every browser contains JavaScript plugin to check or run the client side JavaScript programs.

In Client Side JavaScript program we can use DOM and BOM.

BOM : Browser object model

DOM :document object model

Client side JavaScript program we have to run on Browser using internal JS file or external JS file.

Front end Team back end team

HTML/CSS/JavaScript JEE-Servlet/JSP/EJB

jQuery and bootstrap Spring boot

Asp.net

Php

CGI

**Using JavaScript creating dynamic web page, receive the request and response from a client application, storing data in file, connecting database, creating rest API etc.**

**Node JS goal is to provide an easy way to build scalable networking programs.**

**Node JS provide non blocking IO Operation through networking environment.**

**Because JavaScript support callback function and asynchronous operation.**

**XMLHttpRequest and ActiveXObject**

**Using API if we send the Data we have wait for the acknowledgement may be success or failure.**

**Node JS is an open source, platform independent runtime environment for JavaScript programs.**

**It provide an event driven architecture and non blocking IO operation.**

**Node JS program are executed using V8 JavaScript engine. Chrome browser use V8 engine.**

**Node JS**

**Node JS is not a framework**

**Node JS is not a good for beginner**

**Web Application or Server or REST API.**

**Not a multi thread**

**Node JS = Runtime environment + JavaScript library /JavaScript framework + core module or external modules.**

**Node JS**

**Running Node JS JavaScript using command prompt**

**To install new version of node JS**

**npm install node**

**then set the path may be temporary or permanent.**

**For temporary set path=URL;.;%PATH%**

**set path=C:\Users\91990\node\_modules\.bin;.;%PATH%**

**Node JS provide its own terminal to run the Node JS Programs.**

**REPL : Read Eval Print Loop**

**In Node JS program we can’t use BOM and DOM.**

**console**

**Node JS provider global object ie console which help to display the output in REPL terminal or normal console.**

**process :**

**Node JS Provide another pre-defined global object ie process. This object is use to find the processor details.**

**Node JS Modules**

**Node JS modules is a simple or complex functionality organized in a single or multiple JavaScript files which can be reused through the Node JS application like a Package in Java technologies.**

**Using Node JS module we can achieve separation of concern.**

**Types of modules**

1. **Core Module**
2. **Local module (user-defined modules)**
3. **Third party modules**

**Core Modules : All node js application by default provide few set of modules which come under the category core modules**

1. **fs : File system**
2. **os : Operating system**
3. **net : networking : socket programming**
4. **url : if we want to extract the data from URL**
5. **http : to create basic web application.**
6. **https : to create secure basic web application.**

**etc**

**Fs Module :**

**Node JS provide pre-defined Core module ie fs module which help to do file handling program may be synchronously as well as asynchronously.**

**Syntax to load the module in ES5 style**

**var/let variableName = require(“moduleName”)**

**var fs = require(“fs”);**

**25-07-2021**

**Day 2**

**Phase 3**

**Reading the value through console in node js**

**readline : readline is external module which help to receive the value asynchronously.**

**Syntax to install the external module**

**Globally**

**npm install –g moduleName**

**or**

**locally**

**npm install moduleName**

**package.json file**

**This file hold the configuration details about node js projects.**

**If we install the module locally the package.json file hold details about that modules.**

**readline-sync : This module help use to read the value synchronously.**

**url module : This module is use to find the URL details like port number, localhost, query params etc.**

let urlInfo = "http://localhost:9090/welcome?name=ravi&age=21";

let url = require("url");

//console.log(urlInfo);

let urlObj = url.parse(urlInfo,true);

console.log(urlObj);

console.log(urlObj.protocol)

console.log(urlObj.hostname)

console.log(urlObj.port)

console.log(urlObj.pathname)

console.log(urlObj.query)   // string consider if parse second parameter is false

let obj = urlObj.query;     // it consider as reference if parser second parameter is true

console.log("Name is "+obj.name)

console.log("Age is "+obj.age)

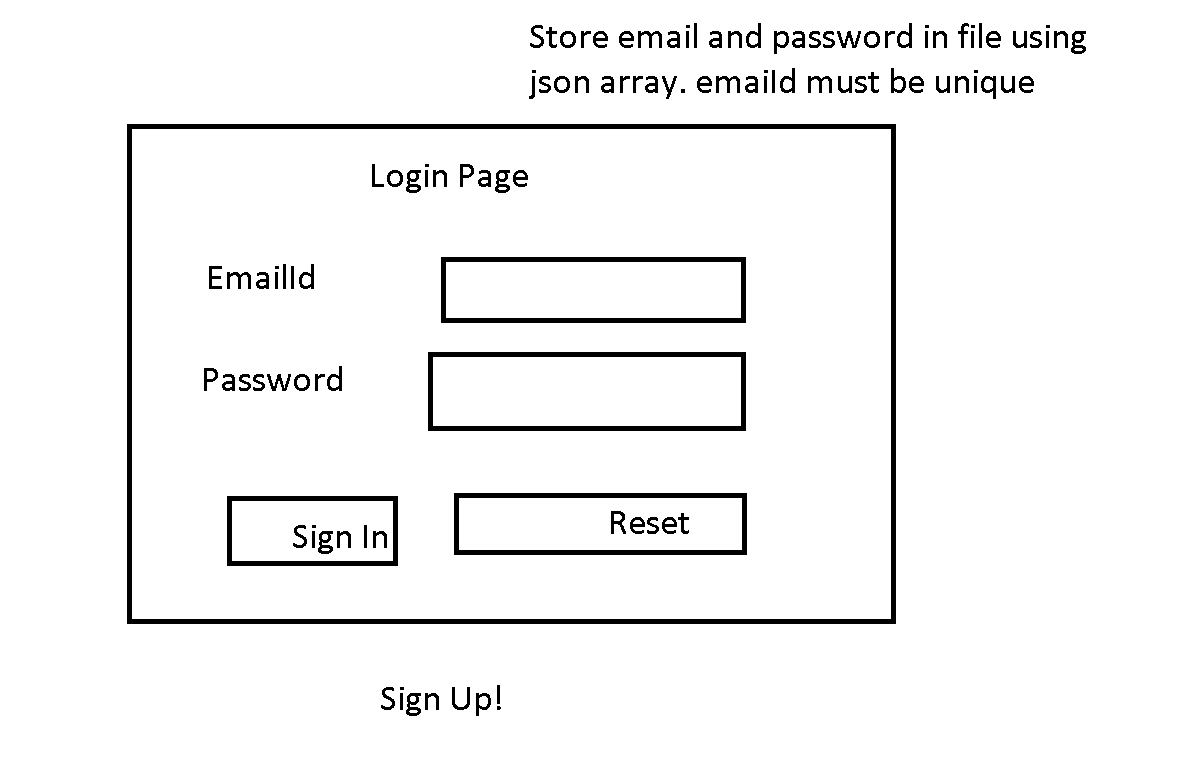
**http module :**

**Node js provide pre-defined module ie http module. Using http module we can create own server using Node JS programs.**

**Jsp, asp, php etc.**

**Tomcat, welogic, jboss, IIS etc**

**http and fs modules**

****

**Difference between Normal Server and Node JS server**

**Non node js server are by default thread base**

**Tomcat, WebLogic, IIS Server etc.**

**Tomcat / IIServer :**

**1,000 : this server can response 1,000 client concurrently.**

**1,001 client if send the request.**

**Online Booking**

**class Booking {**

**int avl=1;**

**}**

**One object created and inside that multiple client equal to multiple thread created.**

**3 client send request.**

**In Thread base server request can be block or lock.**

**Node JS Server provide even driven architecture framework.**

**JavaScript is single thread.**

**We can do or achieve multiple task execute independently using callback and asynchronous communication.**

**31-07-2021**

**Day 3**

**Node JS Express Module**

**http is pre-defined core module with the help of http module we can do core functionality for the web application.**

**Node JS provide External Express module. Express module wrap http module provide more functionality.**

**Like it support get, post, put and delete methods,**

**Using Express module we can create REST API, It support middleware module etc.**

**npm install –g express**

**or**

**npm install express**

**To create package.json file**

**To create package.json**

**npm init**

**It ask some details. Please hit the enter key.(enter continuously). given confirmation yes**

**Or**

**Short cut to create package.json file**

**npm init –y**

**If we install the external module locally then package.json file contains all modules details using dependencies attribute as well as it download the node\_module folder.**

**Like**

**npm install express**

**\_\_dirname : it provide current directory path of our application.**

**If form method is get the information send through URL using URL re-writing technique.**

**Get Method : we can send the request through**

1. **Browser URL**
2. **Hyperlink**
3. **Submit button with method = Get**

**Post method : We can send the request through**

1. **Form submit button with method = post**

[**http://localhost:9090**](http://localhost:9090) **: default message**

[**http://localhost:9090/loginGet**](http://localhost:9090/loginGet) **: open login page with Get method**

**after click submit button of login page with get Method**

[**http://localhost:9090/checkUser?user=Raj&pass=1234**](http://localhost:9090/checkUser?user=Raj&pass=1234) **:**

[**http://localhost:9090/loginPost**](http://localhost:9090/loginPost) **: open login page with Post method**

**after click submit button of login page with post method**

[**http://localhost:9090/checkUser**](http://localhost:9090/checkUser)

**if method is post data send to server through request body part.**

**Node JS provided external module ie body-parser. Which help to enable the body data from request.**

**npm install body-parser**

**then load the module**

**With Express JS html template plugin :**

**Pug**

**Haml.js**

**EJS Plugin**

**Jade template**

**More**

**These above help use to display dynamic value of server side technologies in html page.**

**If we use Express JS template, the view pages become tightly coupled with Express JS or backend technologies.**

**Rest Full Web service came in picture.**

**If we create any application using Java, Asp.net, Python, Node JS with respective view technologies.**

**The application can’t call or re-usability by another application.**

**View**

**JSP Servlet**

**Php Php**

**Asp C# and ASP**

**Web Service : giving the service for web application when both application running using different technologies.**

1. **SOAP Base Web Service : Simple Object Access Protocol**

**SOAP is base upon the SOA ( Service Oriented Architecture).**

**In SOAP Web service we can consume and produce only XML format data.**

**XML is very heavy**

**WSDL ( Web Service Description Language).**

**This file provide the Web Service details develop in different language.**

**WSDL**

**Provide**

**Service**

**Details**

**In xml format.**

**Java asp.net**

**We have to create Java code with the help of wsdl file**

1. **REST Full Web Service:**

**Representational State Transfer : Exposing server side resource as a web service.**

**JSP/Servlet, Asp.net, Php, Node (Express JS).**

**If we expose our resources as Rest API ie Asp.net, Php, Java(Spring boot), Node Express JS.**

**Any other application can call our REST API.**

**Rest Client can be**

**Java , Asp.net, Php, Node JS(Express), angular, React JS or other REST Client.**

**REST Client is Angular**

**In REST Web service we can call REST API using http protocol method with URL.**

**Create new folder**

**ExpressREST\_API**

**npm init –y (folder name must be only word) : package.json**

**npm install express**

**npm install body-parser**

**get() : string message**

**get() : json message**

**get() : object in json format**

**get() : array object in json format**

**passing the value to get method from REST Client Application.**

1. **Query Params :by default plain html web page form with get methods internally follow query param concept.**

**URL?key=value&key=value&key=value**

1. **Path Params**

**URL/path/value1**

**URL/path/value1/value2/value3**

**Day 4**

**01-08-2021**

**Entity : Customer, Person, Project, Employee, Order**

**Get() : Get Resources: retrieve all customer , person, employee details. Retrieve customer details base upon the id.**

**Select query**

**Post() : Create Resource : insert employee details, customer details, order details, etc.**

**Insert query.**

**Browser Plugin :**

**Postman plugin**

**cURL : through unix command or git**

**Put() : update resource : update customer address, phone number using cid, update salary for employee using empId, update amount using accno.**

**Update query**

**Delete() : delete resource : delete employee records using id, order using id, customer using cid**

**Delete query**

**We have to pass the value using path param.**

**Employee**

**Customer**

**Account**

**Project**

**Customer : cid,cname,age,add{city,state}**

**CRUD Operation**

**Get() : get customer details using cid**

**If customer present display customer details else no customer present.**

**Get() : get all customer details.**

**Post() : create new customer details.**

**Put() : update age using cid, address (city, state)**

**Delete() :delete custom details using cid.**

**Create New folder CustomerExpressREST\_API**

**Create package.json file**

**npm init –y**

**npm install express**

**npm install body-parser**