**Day 1 : 09-01-2023**

System software : OS : window, unix, linux etc.

Application or software

To develop application software we need to learn

Programming language

Java, C#, python, etc.

Standalone application or desktop application

The application which running on one machine.

To run this application we required necessary software.

Java or C#

Web Application

<https://www.google.com> 🡪 Uniform resource locator

http: hypertext transfer protocol :

set of rules which help to communicate more than machine.

req(http/https)-----------🡪

Client Server

🡨-------Res(http/https) html/html5

Css/css3

JavaScript

JavaScript provide programming features on web page without server.

Browser provided plugin or parser to check the JavaScript code which we written inside script tag with help of html page.

JavaScript was object based interpreter scripting language which help to create dynamic web page as well as to do validation on client side.

Till ES5

But from ES6 onward JS also known as object oriented interpreter scripting language.

<form>

<label>Emailid</label>

<input type=”email” name=”email” required/>

</form>

Html 4

<!DOCTYPE html public url=<https://ww..asfsaf.sf.sd.dtd>/>

Dtd file hold the rules of html page.

Root tag name must be html which contains two tags

Head or body tag,

In html5 they removed the rules

<!doctype html>

object : object is an any real word entity

properties or state -🡪 have 🡪 variable or fields

Person

behavior 🡪do/does 🡪 function / methods

Bank

Animal

Car

Customer

To describe the object in JavaScript we can do in different

1. Function style
2. Literal style
3. Class style

ES5 Features

ES6 features

Till ES5 JavaScript we are creating user defined object in function style or literal class.

From ES6 onward we can create user defined object using class style.

**javascript function**

**function is use to do the re-usability.**

**function functionName() {**

**body of the function**

**}**

**axios**

**callback**

**closure**

**hosting**

Before Node JS JavaScript known as Client side scripting language.

This scripting language running on browser.

Using JavaScript we were not able to develop web application. Because with help of client side JavaScript we can’t store data in file, we can’t connect to database may be mysql or mongo db etc.

Server side technologies

Html, css and JavaScript

Java (JEE)

Servlet/jsp/spring framework

Asp.net

Php

Python

CGI

Node JS

After Node JS JavaScript is known as Client side as well as server side scripting language.

Node JS provided lot of modules that can be pre defined or user defined which help to do all task on service whichever we are doing using Java or asp.net or php etc.

Like store data in file, security, creating web service, connecting database etc.

**Day 2 : 10-01-2023**

Git : Git is open source tool which help to record the execution of the application.

Create one folder

Then create one text file and write some message.

Open the git bash inside that folder.

git init : to make local folder as a repository

git add filename : we can add only particular file to staging area.

or

git add . : adding all files and folder from current folder to staging area.

git commit –m “message” this command is use to push the file from staging area to local repository.

Day 3 : 11-01-2023

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

Before Node JS JavaScript is known as Client side scripting language which help to run the JavaScript programing on browser with help of html page.

After Node JS JavaScript also known as client side as well as server side scripting language. With help of Node js program we can create server side programming language using JavaScript.

Node js provided lot of pre defined modules which help to do file handling program, creating server side program, creating web service (rest full web service), connecting database like mysql or mongo db using JavaScript.

Before Node JS to run the simple JavaScript program we were depends on html page. But after node js we can run javaScript using command prompt.

First open command prompt

Verify node js installation done or not using command as

node --version

node js provided REPL Terminal (Read Eval Print and Loop)

which help to do javascript using command prompt.

How to open repl terminal

Open the command and write node and hit enter key.

DOM and BOM

Document object model

Browser object model

Client side JavaScript provide us bom and dom.

But node js doesn’t provide bom and dom.

We can’t use document.write and alert in node js program but we can use console.log(“welcome to node js”);

alert(“Welcome”)

Or

window.alert(“Welcome”)

Node js modules

Node js provided lot of module which contains lot of pre defined function or classes which help to do specific task depending upon type of modules.

In node js modules are divided into 3 types

Core module : this modules by default available with node js software.

Third party or external module : base upon our requirement we need download those modules.

User defined modules : this is use to create our own modules.

Fs : File system : it is type of core modules which help to do file handling program synchronously as well Asynchronously using JavaScript.

We can read, write and update or append the data in file.

Loading node js module

This syntax is use to load the module.

var variableName = require(“moduleName”);

node js provided pre defined function ie require which help to load the module.

While doing file handling program we need to load fs module.

Http module : Http is a pre defined module which help to create server side programming language using node js.

Client

<http://www.google.com> -🡪 Request

Server

Server will give response for us. Server will give the response in the form of html page.

Java, Python , asp.net . Those technologies receive the request from client and base upon request it will give response back to client.

To make server side technologies we were depending upon other programming language like java, python, asp.net, php and those language use big server. Web server or Application

Tomcat, web logic, jboss, IIS server nginx server etc.

Node js provided pre defined http module which help to create server side programming language as well as we can create server using http module.

<http://www.google.com>

<http://localhost:3000>

<http://127.0.0.1:3000>

Day 5 : 13-01-2023

Node JS server provide event driven architecture framework.

url : uniform resource locator : this is a core module provided by node js which help to extract url details from client request.

<http://www.google.com:80/home?name=Raj&age=21>

if html form contains get method then if we send the data it will send the data through url using query param concept. So we can see the data in url in the form of query param. Data not a secure. If method is get data send to url with header part. Body is empty.

Through get we can send maximum 255 character data.

If we want data must be secure then we need to send the data using post method. In post method data will send through request body part.

Through post we can send huge data.

Performance wise get is faster than post.

Different between get and post method in html.

<http://www.google.com:80/home?name=Raj&age=21>

synchronous communication

asynchronous communication

1st req

2nd req

3rd req

Client Server

If application follow synchronous communication.

2nd request is depends upon first request response.

3rd request is depends upon second request response.

If application follow asynchronous communication

Then every request execute independently.

Some time 2nd request execute before 1st or 3rd request execute before 1 and 2nd request.

Day 6 : 14/01/2023

Node JS provided lot of pre defined external module in the form of web module which provided lot of great features to develop the web application in simple way

Express : Express js a third party module. Which wrap http module and provided extra functionality to develop web application.

MVC : Model View and Controller

**REST API (Representational state transfer)/ REST API frameworks**

**Command to install third party or external node module**

**npm (node package manager).**

**When we install node js node js provided another command or tool ie npm which help to download external or third party module using node js.**

**To install third party module using npm command we need to run the command as**

**Create the folder as express js program**

**And open this folder in vs code.**

**npm install modulename**

**npm install express**

**Day 7 : 16-01-2023**

**Create the Folder with name Express JS App With View as HTML**

**Open command prompt in this location and using code . open the VS Code in this current location.**

**npm install express**

**after installed express plz check package.json file, package-lock.json file and node\_module folder created in current folder.**

\_\_dirname : this pre defined global property help to provide current path of node js program.

If we want to check multiple user email id and password

1. We can store in array and from array variable we can verify
2. We take the help of fs module and using JSON we can store and retrieve
3. We can connect to database mysql or mongo db database.

JSON : Java Script Objet Notation

It is use to share the data between one technologies to another technologies.

In Json data can be store in key value pairs.

Key must be in double quote and value can be number, string, boolean, array etc.

JSON is a pre defined object which provided set of methods which help convert java script object to string format and vice-versa.

JSON.stringify(login) : it is use to convert JavaScript object to string format.

JSON.parse(login): it is use to convert string to json format.

**Day 8 : 18-01-2023**

By default express js disable post data receive from body part.

To enable we need to depends upon another third party module it

body-parser.

From express js 4.x version onward body-parser module by default downloaded when we download express js. Before 4.x we were need to download this module separately.

Now we need to load that module using require function then we need add middleware in express.

Post is use to store the data

Get is use to get the data.

In this application view as html page. This is a static web page we can’t use any dynamic task in this html page.

Express JS View engine : these view engine are known as dynamic html pages.

Example

Pug or jade

If we use pug or jade or any other view engine then our view technologies and backend technologies are tightly coupled.

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

Amazon Java net banking HDFC HSBC, Sbi etc

Credit card

XML Phone pay asp.net

JSON Paytm express

Google python

Xml and json are use to share the data between two technologies when both the technologies running using different languages.

eXtensible markup language

JSON : JavaScript object notation

2 types of web service

1. SOAP Web service
2. Restfull web service

SOAP : Simple object access protocol

REST api : Representational state transfer.

In SOAP web service we can consume and produce the data only in the form of xml.

XML is more complex and heavy data.

JSON is light weighted which help to share the data.

Rest API is use to consume and produce the data in any format base upon client requirement.

Xml, json, text, html etc.

Crud operation

Product

Order

Manager

Employee

Customer

If we name express js as a rest full web service that application can be call by any application ie rest client. Like

fetch() : JavaScript client side

angular application

react js application

java

python

asp.et

rest client

Creating REST API for customer with Array concept.

Create the folder with name as REST api customer with array

npm install express

then open the folder in vs code.

Get method

1. Get one customer details
2. Get all customers details
3. Find the customer using customer id with

Query param

url?key1=value1; single value

url?key1=value1&key2=value2 multiple value

Path param

url/value1 single value

url/value1/value2 multiple value

**Day 9 : 19-01-2023**

Post method : according to rest api post method is use to create the resource.

Create new customer

Create new employee

Create new product

Store the entity data in array or file or db.

Get method we can call

Get the resources like employee, customer etc

1. URL using browser
2. Using hyperlink
3. Using form with method as get default is get.

Post method

Post method is use to create the resource or store the resource

Customer , employe etc

1. We can’t call through browser.
2. We can’t call through hyperlink
3. We can call through form with method as post.

To test rest api we need to use some tool.

Arc rest api tool

Post man client

SOAP UI tool

Put and patch : These two method is use to update the resource.

If we need to update one property of existing object then we need to use patch method.

Patch method is to update partial object.

Put method use to update full object

If want to update age or name then we need to use patch method

If want to update age and name using cid then we need to use put method.

**Day 10 : 20-01-2023**

We can store the date permanently

File base system

Database system

Limitation of file base system

1. Data redundancy (duplicate records).
2. Inconsistency ( format of the file)

Id,name,salary

1,ravi,12000

1-ravi-12000

1/ravi/12000

1. Security (read mode or write mode)
2. CRUD Operation (Create or insert, read, update and delete)

Database :

Data : raw fact

Information : processed data or meaningful data.

Database : storing the data in table format (row and column)

Dbsm : Database Management system : It is software which help to store the data in table format.

Employee --table

Column id,name,salary

Id Name Salary

1 Ravi 12000 rec1

2 Ramesh 14000 rec2

Trainer

TId TName tech sid Sname age

1 Raj Java 100 Seeta 21

1 Raj Java 101 Reeta 22

2 Raju Python

1 Raj Java

Rdbms : Relational database management system

Trainer

PK

TId TName tech

1 Raj Java

2 Raju Python

Student

PK FK

Sid SName age TSID

100 Reeta 21 1

101 Meeta 22 1

103 Keeta 23 2

104 Leeta 24 1

MySQL, Oracle, Db2, Sql Server etc

SQL (Structured Query Language) : English language which help to interact with any rdbms database to store, retrieve delete and update the records.

All rdbms database are schema base database

In rdbms database we need to create table and number of columns and their data types.

Employee –

Id(int or number),Name(varchar),Salary(float, decimal)

Employee

Id Name Salary age city

1 Ravi 12000 null null

2 Ramesh 14000 null null

3 Ajay 18000 24 null

4 balaji 19000 null Bangalore

If we connect using express js with any rdbsm database we need to convert all json data into table format and vice-versa.

No SQL Database

Mongo db : Mongo DB is one of the type of no sql open source database which help to store the data using document in json format.

Install the mongo db database

<https://www.mongodb.com/try/download/community>

after install please verify inside a folder mongo db database

C:\Program Files\MongoDB\Server\5.0\bin

First inside this folder we need to open two command prompt.

Now inside C drive create one folder as data and inside data create db folder.

Now in one command prompt we need to run mongod this command is use to run the mongo db database.

After few second mongo db database will start

Then another command prompt we need to run mongo client command. In this command we will write all mongo queries.

We need to write all query in mongo terminal

Ctr + L : to clear screen.

show dbs

Or

show databases these to command is use to show database.

use databsename; this command is use to create the database if database not exits if it exits then it will move to existing database.

use mydb;

database is contains more than one collection (in mongo db table is known as collection).

show tables;

Or

show collections

collection is use to store the more than one document. Like in

Mysql table is use to store more than record.

RDBMS Mongo DB

Database database

Table collection

Record document

To create the document in mongo db

db.CollectionName.insert({key:value,key:value,key:value});

to view all document from collection

db.collectionName.find();

**Day 11 : 22-01-2023**

Sample 🡪 collection name

db.Sample.insert({name:”Ravi”})

in mongo db by default every document contains \_id attribute. If we insert same document again and again then it will insert for all document with \_id as attribute with different value. So every document unique consider base upon \_id attribute. \_id is like a primary key in another database. If we want to pass the value for \_id we can use it. Then value consider as unique but we can’t changes \_id attribute name.

collection Emp

\_id 1 name, age, city, salary, deptid

{ "\_id" : 1, "name" : "Ravi", "age" : 23, "salary" : 24000, "city" : "Bangalore", "deptId" : 100 }

{ "\_id" : 2, "name" : "Ramesh", "age" : 25, "salary" : 27000, "city" : "Bangalore", "deptId" : 101 }

{ "\_id" : 3, "name" : "Lokesh", "age" : 26, "salary" : 25000, "city" : "Delhi", "deptId" : 101 }

{ "\_id" : 4, "name" : "Mahesh", "age" : 23, "salary" : 24000, "city" : "Pune", "deptId" : 102 }

{ "\_id" : 5, "name" : "Dinesh", "age" : 26, "salary" : 28000, "city" : "Pune", "deptId" : 100 }

{ "\_id" : 6, "name" : "Reeta", "age" : 27, "salary" : 29000, "city" : "Bangalore", "deptId" : 100 }

{ "\_id" : 7, "name" : "RMeeta", "age" : 28, "salary" : 27000, "city" : "Delhi", "deptId" : 101 }

{ "\_id" : 8, "name" : "Leeta", "age" : 29, "salary" : 23000, "city" : "Pune", "deptId" : 102 }

{ "\_id" : 9, "name" : "Veeta", "age" : 30, "salary" : 26000, "city" : "Bangalore", "deptId" : 100 }

{ "\_id" : 10, "name" : "Shilpa", "age" : 32, "salary" : 28000, "city" : "Bangalore", "deptId" : 101 }

Retrieve particular document using index position.

db.Emp.find()[indexPosition]; whole document

db.Emp.find()[0];

Retrieve particular document field value using index position

db.Emp.find()[indexPosition].fieldName;

db.Emp.find()[2].name;

if we want to retrieve specific fields from all documents.

db.Collection.find({condition},{filterFields});

db.Emp.find({},{name:1}) : display name and \_id field values

db.Emp.find({},{name:1,age:1}) : display name, age and \_id fields values

db.Emp.find({},{name:1,\_id:0,age:1}) display name and age

retrieve more than one fields value using index position

db.Emp.find({},{name:1,age:1,\_id:0})[0]

limit() : this function is use to display n number of document from a collection from top.

db.Emp.limit(2);

skip(): this function is use to skip n number of document from a collection.

db.Emp.skip(4);

sort() : this function is use to sort the document base upon fields.

db.CollectionName.find().sort({fieldname:1}) : ascending order

db.collectionName.find().sort({fieldname:-1}) : descending order

retrieve the documents from a collection with conditions

db.collectionName.find({condition});

db.Emp.find({\_id:1})

db.Emp.find({name:'Ramesh'})

db.Emp.find({salary:28000})

db.Emp.find({city:"Bangalore"})

db.Emp.find({salary:{$gt:25000}});

db.Emp.find({salary:{$gte:25000}});

db.Emp.find({salary:{$lt:25000}});

db.Emp.find({salary:{$lte:25000}});

db.Emp.find({salary:{$eq:25000}});

db.Emp.find({salary:{$ne:25000}});

$and / $or operators

db.Emp.find({$and:[{\_id:1},{name:"Ravi"}]});

db.Emp.find({$and:[{\_id:1},{name:"Ramesh"}]});

db.Emp.find({$or:[{\_id:1},{name:"Ramesh"}]});

db.Emp.find({$and:[{\_id:1},{name:"Ramesh"}]});

**Day 12 : 23-01-2023**

**Update document from collections**

**db.CollectionName.update({field:value},{$set:{field:value,field:value}});**

**db.Emp.update({\_id:1},{$set:{age:25}});**

**db.Emp.update({\_id:2},{$set:{salary:26000}});**

**db.Emp.update({\_id:4},{$set:{salary:30000,city:"Mumbai"}});**

**db.Emp.updateMany({city:"Bangalore"},{$set:{city:"Benguluru"}});**

**remove document**

**db.Emp.remove({\_id:1})**

**Now another one collection and store more than one document with different fields.**

**db.Person.insert({name:"Ravi"});**

**db.Person.insert({name:"Ramesh",lname:"Kumar"});**

**db.Peson.insert({age:24,city:"Bangalore"})**

**db.Peson.insert({age:28,city:"Bangalore",state:"Kar"})**

**adding new field to existing documents.**

**db.Person.update({},{$set:{desg:"UI Developer"}}); Adding desg field for first document in person collection**

**db.Person.updateMany({},{$set:{desg:"UI Developer"}});**

**Adding desg fields all for document in collection.**

**db.Person.update({},{$unset:{desg:1}}); Remove desg field from first document in collection**

**db.Person.updateMany({},{$unset:{desg:1}}); remove desg field from collection for all document.**

**Drop collection**

**db.collectionName.drop();**

**db.Person.drop(); remove collection as well as all document from that collection**

**db.Sample.remove({}) remove all document from collection but empty collection in db.**

**Student**

**Sid SName age tech**

**1 Ravi 21 Java**

**2 Ramesh 23 Java, Python**

**Student**

**Sid SName Age Java Python C C++**

**1 Ravi 21 yes no no no**

**2 Ramesh 23 yes yes no no**

**Student**

**Sdi SName Age**

**pk**

**1 Ravi 21**

**2 Ramesh 23**

**Tech**

**TId tName**

**pk**

**100 Java**

**101 Python**

**102 C**

**103 C++**

**StudentTech**

**PK FK FK**

**Stid sid tid**

**111 1 100**

**222 2 100**

**333 2 101**

**Storing array values**

**db.StudentInfo.insert({\_id:100,name:"ajay",age:21,tech:["Java"]});**

**db.StudentInfo.insert({\_id:101,name:"balaji",age:24,tech:["Java","Python"]});**

**db.StudentInfo.insert({\_id:103,name:"seeta",age:23,tech:["html","css","javascript","reactjs","nodejs"]});**

**db.StudentInfo.insert({\_id:103,name:"seeta",age:23,tech:["html","css","javascript","reactjs","nodejs"]});**

**condition on array values**

**db.StudentInfo.find({tech:"Java"});**

**db.StudentInfo.find({tech:"html"},{name:1,\_id:0});**

**In mongo db in json format value can be array type.**

**Mongo DB relationship**

**In Oracle or MySQL database we will connect or achieve relationship using primary key and foreign key.**

**Employee --🡪 Address (one to one relationship)**

**Employee 🡪 Address (one to many relationship)**

**Employee 🡪 Passport (One to one relationship)**

**Trainer 🡪 Student (one to one/ many relationship)**

**Employees -🡪 Department (many to one relationship)**

**Employees -🡪 SkillSet (many to many relationship)**

**In mongo db we can achieve relationship using 2 ways**

1. **Embedded style : we store everything in single collection**
2. **Linking style : we need more than one collection**

**In mongo db we can achieve relation on document level rather than collection level.**

**In Oracle or mysql we can achieve relationship on table level not on record level.**

**One employee has one address**

**db.Employees.insert({\_id:1,name:"Ravi",age:21,salary:24000,address:{city:"Bangalore",state:"Kar"}});**

**one employee has more than one address**

**db.Employees.insert({\_id:2,name:"Ramesh",age:23,salary:26000,address:[{city:"Bangalore",state:"Kar"},{city:"Pune",state:"Mh"}]});**

**one employee has one address as well as working in one project**

**db.Employees.insert({\_id:3,name:"Raju",age:26,salary:25000,address:[{city:"Mysore",state:"Kar"}],project:[{pid:111,tech:"Node JS"}]});**

**one employe has one address as well as working in more than one project**

**db.Employees.insert({\_id:4,name:"Reeta",age:21,salary:22000,address:[{city:"Mumbai",state:"Mh"}],project:[{pid:111,tech:"Node JS"},{pid:222,tech:"React JS"}]});**

**db.Employees.find().pretty();**

**One to many relationship in linking style**

**Trainer**

**\_id tname tech**

**1 Raj Java**

**2 Ravi Python**

**db.Trainer.insert({\_id:1,tname:"Raj",tech:"Java"});**

**db.Trainer.insert({\_id:2,tname:"Ravi",tech:"Python"});**

**Student1 storing only trainer id**

**\_id sname age tsid**

**100 Seeta 21 1**

**101 Reeta 22 1**

**102 Meeta 23 2**

**103 Keeta 24 1,2**

**db.Student1.insert({\_id:100,sname:"Seeta",age:21,tsid:db.Trainer.find()[0].\_id});**

**db.Student1.insert({\_id:101,sname:"Reeta",age:22,tsid:db.Trainer.find()[0].\_id});**

**db.Student1.insert({\_id:102,sname:"Meeta",age:23,tsid:db.Trainer.find()[1].\_id});**

**db.Student1.insert({\_id:103,sname:"Keeta",age:24,tsid:[db.Trainer.find()[0].\_id,db.Trainer.find()[1].\_id]});**

**Student2 storing complete trainer document**

**\_id sname age trainerdetails**

**db.Student2.insert({\_id:100,sname:"Seeta",age:21,tsid:db.Trainer.find()[0]});**

**db.Student2.insert({\_id:101,sname:"Reeta",age:22,tsid:db.Trainer.find()[0]});**

**db.Student2.insert({\_id:102,sname:"Meeta",age:23,tsid:db.Trainer.find()[1]});**

**db.Student2.insert({\_id:103,sname:"Keeta",age:24,tsid:[db.Trainer.find()[0],db.Trainer.find()[1]]});**