**Phase 5 :**

**Day 1 : 06/26/2023**

Junit Testing

Testing and Deployment using devops life cycle

Test NG Open Source framework for testing java programs.( Unit testing and Integration testing)

Selenium Testing with Test NG Framework (Automation Testing )

Introduction to AWS (EC2, S3 and EBS) etc

Git

Maven or Gradle

Docker Devops

CI and CD tool using Jenkin

Overview of Kubernetes

Phase end project

Testing : Testing is use to find the defects or error or bugs in the application.

Running the program develop in any language in directly we are testing that program.

class Operation {

public int add(int x, int y) {

int sum = x+y;

return sum;

}

}

We are testing java programs without main method also known as testing the programs.

Testing mainly divide into 2 types.

1. Black box testing :

Input ------------------🡪Process ------------------🡪Output

1. White box testing

Input ------------------🡪Process ------------------🡪Output

Unit Testing : Unit testing is a type of white box testing which help to test function functionality working or not properly. Unit is smallest code which work independently. Like in a method or function etc.

To do this type of unit testing

Java use 2 framework

jUnit : junit testing open source framework which help to do unit testing

Test NG (next generation) : Test NG is base upon Junit testing. Light weighted which help to do

Unit testing as well as integration testing.

JavaScript -🡪 Jasmine is an open source framework which help to do unit testing for JavaScript programs.

React JS 🡪 JEST is an third party framework provided by Facebook which help to do the unit testing for React JS programs.

We will do unit testing using Junit

Junit 3.x without annotation

Junit 4.x with annotation

jUnit 5.x with more than third party library and support java8 features with annotation

test case : test case is a type of class which contains more than one test function which help to test function functionality.

test suite : test suite is use to run more than one test case classes.

Junit hook :

4 hook

@Before : this function get called before each @Test function

@After : this function get called after each @Test function

@BeforeClass :this function get called only once before one or all @Test function

@AfterClass : this function get called only once after one or all @Test function

TestSuite : Test suite is use to execute more than one test case classes.

TestNG : Test NG another testing framework inspired by junit testing framework. It is an open source testing framework similar to junit test framework. In Test NG (next generation).

Test NG is combination of junit and nunit

It is designed better than Junit testing especially when we do the integration testing.

Features of Test NG framework

1. It generate report in html format (by default).
2. It provided few more annotation which were missing in jUnit.
3. It allow use to do group as well as priority base testing.
4. It support parallel testing features. jUnit do sequentially testing.

Unit testing using Test NG

Phase 3 project discussion

**Phase 5 :**

**Day 2 : 06/27/2023**

Unit : we create separate test suite which help to run more than one test case.

In Unit is it is a type of class.

But in TestNG that class replace by xml file. This file hold the details about all TestNG class.

In Test Ng all test function execute by default in alphabetical order.

In Test NG inside @Test annotation we can use priority property to run the test base upon the priority.

TestNG hook.

@BeforeSuite : it will call only once when we run through suite file. It will call only once

Inside suite we can configure more than one TestNG class. each class contains

More than one test function

Before all test function part of same TestNG class or different TestNG class

@BeforeTest hook get called only once.

@BeforeClass : it will call each class level only once.

@BeforeMethod : it will call each @Test function. This will call again and again.

**Day 3 : 06/28/2023**

Selenium : Selenium is one of the most widely used open source Web UI(user interface) automation tool.

Selenium is platform independent, language independent and browser independent Web UI tool.

Selenium support by Java, C#, Python, JavaScript etc.

Selenium can be use to automate functional test (actual business code) non function test (it will support to do the task).

It can be integrate with DevOps tool like Git, Jenkin, Docker and Kubernetes.

We can integrate Selenium with jUnit or TestNG framework.

We can do Selenium code using Java as well as Selenium provided OWN IDE which we can configure with browser or we can download to do the Testing for UI Component doesn’t matter that application develop using any language.e

We need to download the browser specific driver.

To connect mysql or oracle database we downloaded jar file manually or using maven tool.

In Selenium we need to download the browser driver in the form of .exe or other format base upon OS support.

In Web Application all contents are known as DOM. (Document Object Model).

If we want to load the tag or dom elements contents then we need to refer those tags using

Tagnaname, classname, idname etc.

First we need to refer to dom tags base up selector and Selenium provided WebElement API which help to read, write and update DOM content using Java.

**Phase 5 :**

**Day 5 : 06/30/2023**

RDBMS (Relational Database management System)

Limitation of MySQL or Oracle or db2 etc.

All RDBMS Databases are schema base database. Means before storing any record in database first we need to create table with number of column as well as column data types.

Structure data

Semi structure

Un structure

Emp

Id Name Salary age city

1 Ravi 12000.50 null null

2 Ramesh 14000.50 null null

3 Lex 16000.75 21 null

4 Neeta 18000.00 null Bangalore

No SQL : In No SQL Database we can store the data in any format apart from Table format. Like Json, xml, graph etc.

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

Mongo db is cross platform high performance no sql database.

RDBMS Mongo DB

Database Database

Table Collection

Record (same types) document (can be same type or different types)

Table format using cell each document hold information in json format

Each cell allow single value document can be same or different types.

Open mongo terminal using command as

mongo

show databases

or

show dbs

use databasename;

like

use phase5; if database present it will move to existing database else it will create to move inside that database.

show colletions;

Or

show tables; this command is use to show all collection present in current database.

Mongo db provided pre defined object ie db which contains lot of pre defined functions which help

Create collection, store document, delete, document, update document and retrieve documents.

db.createCollection(“CollectionName”);

To store the document in collection

db.CollectionName.insert({key1:value1,key2:value2,key3:value3});

View the documents from collection

db.collectionName.find({});

mongo db internally create \_id pre defined field for each document to maintain unique ness between two document and it will generate unique random alpha numerical value.

If we want we can pass the value but we can’t change the field name it is consider as primary key in mongo db.

Emp

\_id name salary age city deptId

In mongo db while inserting document in json format if collection present it will insider that document inside that document else it will create.

> db.Emp.insert({\_id:1,name:"Ravi",age:21,salary:240000,city:"Bangalore",deptId:100});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:2,name:"Ramesh",age:24,salary:280000,city:"Bangalore",deptId:101});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:3,name:"Lokesh",age:27,salary:210000,city:"Delhi",deptId:100});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:4,name:"Neena",age:21,salary:180000,city:"Delhi",deptId:101});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:5,name:"Leena",age:22,salary:190000,city:"Bangalore",deptId:100});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:6,name:"Teena",age:23,salary:210000,city:"Delhi",deptId:101});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:7,name:"Heena",age:24,salary:220000,city:"Mumbai",deptId:100});

WriteResult({ "nInserted" : 1 })

> db.Emp.insert({\_id:8,name:"Seena",age:26,salary:260000,city:"Mumbai",deptId:101});

WriteResult({ "nInserted" : 1 })

db.Emp.find({}); : it retrieve all documents from Emp collection

db.Emp.find({})[0]; : it retrieve 0 position index position document from Emp collection

db.Emp.find({})[1].name : it retrieve particular index position field document values.

Retrieve specific fields value from document.

db.Emp.find({condition},{fieldname:1});

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

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

db.Emp.find({},{name:1,city:1,\_id:0}); display name and city no \_id

retrieve the document from collection with conditions.

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

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

db.Emp.find({salary:24000});

db.Emp.find({salary:{$gt:200000}});

db.Emp.find({salary:{$gte:200000}});

db.Emp.find({salary:{$lt:200000}});

db.Emp.find({salary:{$lte:200000}});

db.Emp.find({salary:{$eq:200000}});

db.Emp.find({salary:{$ne:260000}});

and/or operator

db.Emp.find({$and:[{salary:{$gt:150000}},{salary:{$lt:200000}}]});

db.Emp.find({$or:[{salary:{$gt:250000}},{city:"Bangalore"}]});

sort the documents.

db.Emp.find().sort({age:1}); sort age by document in asc order

db.Emp.find().sort({age:-1}); sort age by document in desc order

update document

update salary using \_id field

db.Emp.update({\_id:1},{$set:{salary:250000}});

update salary and age using \_id field

db.Emp.update({\_id:1},{$set:{salary:200000,age:26}});

this query can update more than one document if conditions meet.

db.Emp.updateMany({city:"Delhi"},{$set:{city:"New Delhi"}});

remove documents

it will remove using \_id field

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

it will remove using city field

db.Emp.remove({city:"Bangalore"});

MySQL

Student details

Sid(pk) sname age, marks

1 Leena 21 67,88,98,65

Student

SID(PK) Sname Age

1 Leena 21

2 Meeta 25

Marks

Mid(PK with auto increment) sid(fk) marks

100 1 67

101 1 88

102 1 98

103 1 65

104 2 78

Mongo db relationship

One to One

One to many

Trainer

PK

Tid TName tech

1 Raj Java

Student

PK FK

Sid SName age tsid

100 Neena 21 1

101 Meena 22 1

In mongo db we can achieve relationship using two ways

1. Embedded style relationship : single collection
2. Linking style relationship : more than one collection

Embedded style relationship

Employee has one Address : one to one relationship

Employee working more than one project at same time : one to many relationship

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

db.Employee.insert({\_id:2,name:"Rajesh",age:22,address:{city:"Mumbai",state:"Mh"},project:[{pid:1122,tech:"Java"}]});

db.Employee.insert({\_id:3,name:"Neena",age:23,address:{city:"Mumbai",state:"Mh"},project:[{pid:1122,tech:"Java"},{pid:1123,tech:"Python"}]});

This code display the output in proper or pretty format.

db.Employee.find().pretty();

Linking Style

Trainer

\_id(PK) TName tech

1 Ravi Java

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

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

Student1

\_id(PK) SName age tsId

100 Leena 21 1

101 Veena 22 1

db.Student1.insert({\_id:100,tname:"Meeta",age:21,tid:db.Trainer.find()[0].\_id});

db.Student1.insert({\_id:101,tname:"Leeta",age:22,tid:db.Trainer.find()[0].\_id});

db.Student1.insert({\_id:102,tname:"Keeta",age:23,tid:db.Trainer.find()[1].\_id});

Student2

db.Student2.insert({\_id:100,tname:"Meeta",age:21,tid:db.Trainer.find()[0]});

\_id(PK) SName age tsdetails

100 Leena 21 {\_id:1,tname:”ravi”,tech:”Java”}