**Mean Stack Lab Manual**

**(1-12 except 6th)**

1. Create a database related to Hospital Management System and perform CRUD operations using MongoDB.

2. Create a database related to Bus Ticket Reservation System and perform CRUD operations using MongoDB.

3. Write node.js program to create, access, modify the Arrays

4. Write node.js program to create, access, modify the JSON object

5. Install express and create an application.

6. Perform CRUD operations using express and mongoDB.

7. Write a typescript program to work with different types of variables, functions and run the programs using node environment.

8. Write a typescript program to work with classes.

9. Create a simple angular application using Angular CLI

10. Create an angular application to work with components.

11. Create an angular application to work with Pipes

12. Create an angular application to work with Directives.

**Exp:1**

**>use hmsc**

**>db.createCollection("Doctor")**

**>db.Doctor.insertOne({'id':57})**

**>db.Doctor.fine({})**

**>db.Doctor.updateOne({'Specalization':'Nephrologist'},{$set:{'Specalization':'opthamologist'}**

**>db.Doctor.deleteOne({'Did':5})**

**collections:**

**doctor ( id, name, specialization, designation)**

**patient ( id, name, disease, mobile no, op date)**

**nurse (id, name, specialization)**

**operation theater (patient id, doctor id, surgery type, surgery date, cost)**

**medical center (id, name, quantity, cost)**

**----------------------------------------------------------------------------------------------------------------------------**

**exp2:**

**>use btrs**

**>db.createCollection("Drivers")**

**>db.insertMany([{'id':1, 'name':'jas'}, {'id':2,'name':'jas}])**

**collections:**

**drivers (dvr id, dvr name, phn no, bus id)**

**bus (bus id, bus num, seats, depo, dvr id, from, to)**

**passenger (pid, pname, bus num, age, gender, phn no, source, destination, seat number)**

**transaction (payment id, pid, transaction date, cost, travelling date)**

**depo (depo id, depo name, location, bus id, owner)**

**For MongoDB Compass:** [Download MongoDB Community Server | MongoDB](https://www.mongodb.com/try/download/community)

**For Mongo Shell:** [MongoDB Shell Download | MongoDB](https://www.mongodb.com/try/download/shell)

**Exp3: (node filename.js)**

**console.log("This is Exp 3 we will discuss about creating, accessing, modifying arrays");**

**console.log("functions like index, indexOf, lastIndexOf, push, pop, shift, unshift, sort");**

**var animals=['tiger','lion','elephant']**

**console.log("\nAnimals are : ",animals);**

**console.log("\nAccessing arrays");**

**console.log("\nindex() function:\n");**

**console.log(animals[1],"is the King of forest");**

**console.log("\nindexOf() function\n");**

**console.log(animals.indexOf('tiger'));**

**console.log("\nlastIndexOf() function\n");**

**console.log(animals.lastIndexOf('tiger'));**

**console.log("\nPush function\n");**

**animals.push('snake','zebra');**

**console.log("\nAnimals are : ",animals);**

**console.log("\nPop function\n");**

**animals.pop()**

**console.log("\nAnimals are : ",animals);**

**console.log("\nUnshift function\n");**

**animals.unshift('zebra');**

**console.log("\nAnimals are : ",animals);**

**console.log("\nShift function\n");**

**animals.shift()**

**console.log("\nAnimals are : ",animals);**

**console.log("\nSorting function\n");**

**console.log("Before Sorting: ")**

**console.log("\nAnimals are : ",animals);**

**console.log("After Sorting: ")**

**console.log("\nAnimals are : ",animals.sort());**

**Exp4:**

**create.json (node create.json)**

**[**

**{**

**"name": "Jaswanth",**

**"age": 20,**

**"roll": "21481A5457"**

**},**

**{**

**"name": "Hitesh",**

**"age": 19,**

**"roll": "21481A5432"**

**}**

**]**

**Access.js (node filename.js)**

**const users = require("./create");**

**console.log(users);**

**Modify.js (node filename.js)**

***// Sample JSON object representing an array of users***

**let users = [**

**{ name: 'John', age: 30 },**

**{ name: 'Alice', age: 25 },**

**{ name: 'Bob', age: 35 }**

**];**

***// New user object to push***

**const newUser = { name: 'Emily', age: 28 };**

***// Pushing the new user object into the users array***

**users.push(newUser);**

**console.log(users);**

**Exp 5:**

***// npm install express***

***// npm install express --save***

***// npm install body-parser --save***

***// npm install cookie-parser --save***

***// npm install multer --save***

**var express = require('express');**

**var app = express();**

**app.get('/', function (req, res) {**

**res.send('Hello!!!'); })**

**var server = app.listen(8000, function () {**

**console.log("Running at** [**http://localhost:8000/**](http://localhost:8000/)**") })**

**Exp 7: (tsc exp7.ts) -> then exp7.js file will be created then run that using (node exp7.js) command**

*//npm install typescript --save-dev*

*//npm install typescript -g*

function studentInfo(name: string, branch: string, cgpa: number) {

    console.log("Student name: " + name);

    console.log("Student branch: " + branch);

    console.log("Student cgpa: " + cgpa);

}

studentInfo("dinesh", "cse", 89);

**Exp 7.js: (no need to copy and run this program)**

function studentInfo(name, branch, cgpa) {

    console.log("Student name: " + name);

    console.log("Student branch: " + branch);

    console.log("Student cgpa: " + cgpa);

}

studentInfo("dinesh", "cse", 89);

**Exp 8: (tsc exp8.ts) -> then exp8.js file will be created then run that using (node exp8.js) command**

class Student {

    roll:number;

    studname:string;

    constructor(code:number,name:string){

        this.studname=name;

        this.roll=code;

    }

    display():void{

        console.log("name:",this.studname);

        console.log("code:",this.roll);

    }

}

let obj=new Student(5457,'Jaswnath');

obj.display();

**Exp 8.js: (no need to copy and run this program)**

var Student = */\*\* @class \*/* (function () {

    function Student(code, name) {

        this.studname = name;

        this.roll = code;

    }

    Student.prototype.display = function () {

        console.log("name:", this.studname);

        console.log("code:", this.roll);

    };

    return Student;

}());

var obj = new Student(5457, 'Jaswnath');

obj.display();

**Exp 9,10:**

**npm install -g @angular/cli**

***(if u get any error)***

***{****open powershell in administrator mode:*

*Get-ExecutionPolicy*

*Set-ExecutionPolicy RemoteSigned*

*(type y for complete the process) next:*

*windows+r -> (type) appdata->roaming->npm->node modules->angular->cli*

*copy the path and add it to environmental variable* ***}***

**ng version**

**ng new --no-standalone** (for creating a new app)

Jaswanthapp //it will create your app with a name

code . (to open vs code)

in the program open terminal and type:

cd [your appname]

**ng serve** (for running the program)

for creation of components in the angular:

**ng g c [your component name]**

(ex: type <app-componentname></app-componentname>)

**Exp 9:**

<h1>Hello(:</h1>

**Exp 10:**

<img *src*=" ">

<app-header></app-header>

**Exp 11:**

As u already created the app so go to **app.component.ts** and change the code of *export class* inside:

dob=new Date();

and as well as in **app.component.html** change the code to this:

<h1>{{dob|date:'dd/mm/yyyy'}}</h1>

you can see the output date in dd/mm/yyyy format

**Exp 12:**

<!DOCTYPE *html*>

<html>

<script *src*="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

<body>

<div *ng-app*=""*ng-init*="firstName=''">

<p>Your Name: <input *type*="text"*ng-model*="firstName"></p>

<p *style*="font-size: 40px;">Greetings, Mr/Mrs. {{firstName }}</p>

</div>

</body>

</html>