Index

Practical 1

Aim: Write a program to implement MongoDB data model

Input:

Embedded Data

Model
```
{
"Id": "01",
"Employee_Id": "2014",
"Personal": {
 "FirstName": "Utopia",
 "LastName": "PopSmoke",
 "Class": "Maharashtra"
},
"address": {
 "street": "Oesis",
 "city": "Mumbai",
 "state": "Maharashtra",
 "zipCode": "400043",
 "country": "India"
},
"Conatct Details": {
 "email": "youknowme@gmail.com",
 "phone": "1234567890"
}
}
```

Output:

```
{
    "_id": {
        "$oid": "67c49ec9ac56887434f7b192"
    },
    "Id": "01",
    "Employee_Id": "2014",
    "Personal": {
        "FirstName": "Utopia",
        "LastName": "PopSmoke",
        "Class": "Maharashtra"
    },
    "address": {
        "street": "Oesis",
        "city": "Mumbai",
        "state": "Maharashtra",
        "zipCode": "400043",
        "country": "India"
    },
    "Conatct Details": {
        "email": "youknowme@gmail.com",
        "phone": "1234567890"
    }
}
```

Normalised Data Model

Personal Details:

```
{
    "FirstName": "Unknown",
    "LastName": "Anonymous",
    "Class": "SYCS"
}
```

```
▼ {
▶    "_id": {...},
      "FirstName": "Unknown",
      "LastName": "Anonymous",
      "Class": "SYCS"
   }
```

Address:

```
{
"street": "Oesis",
"city": "Mumbai",
"state": "Maharashtra",
"zipCode": "400043",
"country": "India"
}
```

```
▶  ▼ {
   ▼    "_id": {
           "$oid": "67c4a0f8ac56887434f7b196"
        },
        "street": "Oesis",
        "city": "Mumbai",
        "state": "Maharashtra",
        "zipCode": "400043",
        "country": "India"
     }
```

Contact Details

```
{
"email": "youknowme@gmail.com",
"phone": "1234567890"
}
```

```
▼ {
▼    "_id": {
        "$oid": "67c4a19aac56887434f7b198"
     },
     "email": "youknowme@gmail.com",
     "phone": "1234567890"
  }
```

Practical 2

Aim: Write a program to implement crud operation on MongoDB.

Input:

       Show dbs

Output:

```
rdj> show dbs
RDJ          8.00 KiB
admin       40.00 KiB
config     108.00 KiB
local       40.00 KiB
rdj> |
```

Input:

 Use admin

Output:

```
rdj> use admin
switched to db admin
admin> |
```

Input:

db.createCollection("SYCS")

Output:

```
RDJ> db.createCollection("SYCS")
{ ok: 1 }
RDJ> |
```

Input:

      db.sycs.insertOne({name:"Tony",age:"20",gender:"Male",address:"Nagpada"})

Output:

```
RDJ> db.sycs.insertOne({name:"Tony",age:"20",gender:"Male",address:"Nagpada"})
{
  acknowledged: true,
  insertedId: ObjectId('67c4a521688f38617bfa4214')
}
RDJ> |
```

Input:

  db.SYCS.insertMany([{name:"Hulk",age:"25",gender:"Male",address:"Titwala"},
   {name:"Hawkeye",age:"20",gender:"Male",address:"Dehli"},
   {name: "Wanda",age:"18",gender:"Female",address:"Mumbai"}])

Output:

```
RDJ> db.SYCS.insertMany([{name:"Hulk",age:"25",gender:"Male",address:"Titwala"},
...    {name:"Hawkeye",age:"20",gender:"Male",address:"Dehli"},
...    {name: "Wanda",age:"18",gender:"Female",address:"Mumbai"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67c4a615688f38617bfa4215'),
    '1': ObjectId('67c4a615688f38617bfa4216'),
    '2': ObjectId('67c4a615688f38617bfa4217')
  }
}
```

Input:

db.sycs.find({"age":"20"})

Output:

```
RDJ> db.sycs.find({"age":"20"})
[
  {
    _id: ObjectId('67c4a521688f38617bfa4214'),
    name: 'Tony',
    age: '20',
    gender: 'Male',
    address: 'Nagpada'
  }
]
```

Input:

db.sycs.replaceOne({name:"Hulk"},{name:"Wanda"})

Output:

```
RDJ> db.sycs.replaceOne({name:"Hulk"},{name:"Wanda"})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

Input:

db.sycs.updateOne({name:"Hulk"},{$set:{address:"Mumbra"}})

Output:

```
RDJ>  db.sycs.updateOne({name:"Hulk"},{$set:{address:"Mumbra"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

Input:

db.sycs.deleteOne({name:"Hawkeye"})

Output:

```
RDJ> db.sycs.deleteOne({name:"Hawkeye"})
{ acknowledged: true, deletedCount: 0 }
RDJ>
```

Input:

db.sycs.deleteMany({gender:"female"})

Output:

```
RDJ> db.sycs.deleteMany({gender:"female"})
{ acknowledged: true, deletedCount: 0 }
RDJ>
```

Input:

db.dropDatabase()

Output:

```
RDJ> db.dropDatabase()
{ ok: 1, dropped: 'RDJ' }
RDJ>
```

Practical 3

Aim: Write a program to perform validation of a form using AngularJS.

Input:

```html
<!Doctype html>
<html>

<head>
    <title>AngularJS Form Validation</title>
    <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"> </script>
    <script type="text/javascript">
        var app = angular.module('formApp', []);
        app.controller('formCtrl', function ($scope) {
            $scope.sendForm = function () {
                $scope.msg = "Form Validated";
            };
        });
    </script>
</head>

<body>
    <div ng-app="formApp" ng-controller="formCtrl">
        <h2>AngularJS Form Validations</h2>
        <form name="personForm" ng-submit="sendForm()">
            First Name:<input type="text" name="firstname" ng-model="person.fname" required />
            <span ng-show="personForm.firstname.$error.required">Required!</span>
            <br /><br />
            Last Name:<input type="text" name="lastname" ng-model="person.lname" required />
            <span ng-show="personForm.lastname.$error.required">Required!</span>
            <br /><br />
            <button type="submit">Submit Form</button><br /><br />
            <span>{{msg}}</span>
        </form>
    </div>
</body>

</html>
```

Output:

Practical 4

Aim: Write a program to create and implement modules.

Input:

```html
<!DOCTYPE html>
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>

<body>
  <div ng-app="myApp" ng-controller="myCtrl">
    First Name: <input type="text" ng-model="firstName"><br>
    Last Name: <input type="text" ng-model="lastName"><br>
    <br>
    Full Name: {{firstName +" "+ lastName}}
  </div>
  <script>
    var app = angular.module('myApp', []);
    app.controller('myCtrl', function ($scope) {
      $scope.firstName = "Iron";
      $scope.lastName = "Man";
    });
  </script>
</body>

</html>
```
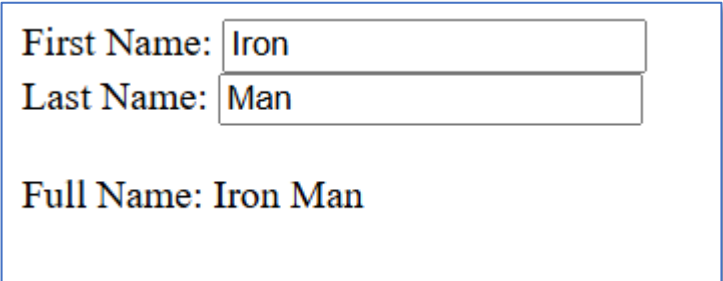
Output:

First Name: Iron
Last Name: Man

Full Name: Iron Man

Practical 5

Aim:Write a program to implement Controller in AngularJS.

Input:

```html
<html>
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>

<body>
  <div ng-app="myApp" ng-controller="myCtrl">
    First Name: <input type="text" ng-model="firstName"><br> Last Name: <input type="text" ng-model="lastName"><br>
    Email Id: <input type="text" ng-model="EmailId"><br> Password: <input type="password" ng-model="password"><br>
    <br>
    Full Name: {{firstName + " " + lastName}} <br> Email Id: {{EmailId}}<br>
    Password: {{password}}
  </div>
  <script>
    var app = angular.module('myApp', []); app.controller('myCtrl', function ($scope) {
      $scope.firstName = "";
      $scope.lastName = "";
      $scope.EmailId = "";
      $scope.password = "";
    });
  </script>
</body>

</html>
```
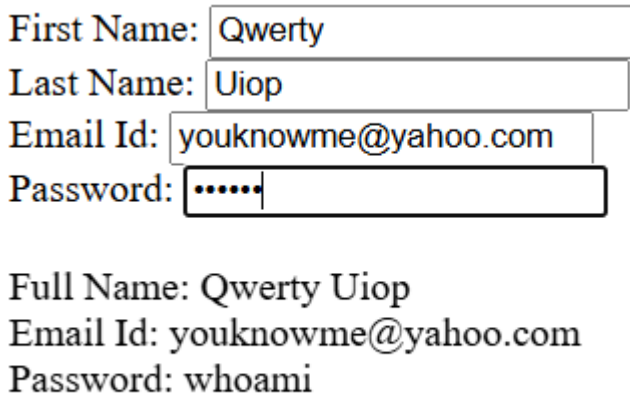
Output:

First Name: Qwerty
Last Name: Uiop
Email Id: youknowme@yahoo.com
Password: ••••••

Full Name: Qwerty Uiop
Email Id: youknowme@yahoo.com
Password: whoami

Practical 6

Aim: Create a simple html "Hello World" project using angularJs Framework and apply ng-controller, ng-model, and expressions.

Input:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>Angular Demo</title>
    <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
    <script>
        var myApp = angular.module("AngularApp", []);
myApp.controller("HelloController", function ($scope) {
            $scope.txtname = "";
        });
    </script>
</head>

<body ng-app="AngularApp">
    <div ng-controller="HelloController">
        Enter name: <input type="text" ng-model="txtname">
        <p>Entered Text: {{txtname}}</p>
    </div>
</body>

</html>
```

Output:

Enter name: Marvel Cinematic

Entered Text: Marvel Cinematic