Program 3

MongoDB- CRUD Demonstration

I. Perform the following DB operations using MongoDB.

1. Create a database "Student" with the following attributes Rollno, Age, ContactNo, Email- Id.

> use StudentDB

2. Insert appropriate values

> db.students.insertMany([

   { Rollno: 1, Age: 20, ContactNo: "1234567890", EmailId: "student1@example.com" },

   { Rollno: 2, Age: 21, ContactNo: "1234567891", EmailId: "student2@example.com" },

   { Rollno: 10, Age: 22, ContactNo: "1234567892", EmailId: "student10@example.com" },

   { Rollno: 11, Age: 23, ContactNo: "1234567893", EmailId: "student11@example.com", Name:

"ABC" }

])

3. Write query to update Email-Id of a student with rollno 10.

> db.students.updateOne(

   { Rollno: 10 },

   { $set: { EmailId: "newemail10@example.com" } }

)

4. Replace the student's name from "ABC" to "FEM" of rollno 11

> db.students.updateOne(

   { Rollno: 11, Name: "ABC" },

   { $set: { Name: "FEM" } }

)

II. Perform the following DB operations using MongoDB.

1. Create a collection by name Customers with the following attributes. Cust_id, Acc_Bal, use Bank;

```
db.Customers.insertOne({
    Cust_id: 1,
    Acc_Bal: 1000,
    Acc_Type: "A"
});
```

2. Insert at least 5 values into the table

```
>       use CustomerDB
db.customers.insertMany(
[
    { Cust_id: 1, Acc_Bal: 1500, Acc_Type: 'Z' },
    { Cust_id: 2, Acc_Bal: 800, Acc_Type: 'Y' },
    { Cust_id: 3, Acc_Bal: 2000, Acc_Type: 'Z' },
    { Cust_id: 4, Acc_Bal: 1000, Acc_Type: 'X' },
    { Cust_id: 5, Acc_Bal: 1300, Acc_Type: 'Z' }
```

3. Write a query to display those records whose total account balance is greater than 1200 of account type 'Z' for each customer_id.

```
db.Customers.find({
    Acc_Type: "Z",
    Acc_Bal: { $gt: 1200 }
});
```

4. Determine Minimum and Maximum account balance for each customer_i db.Customers.aggregate([

```
    {
        $group: {
            _id: "$Cust_id",
            minBalance: { $min: "$Acc_Bal" },
            maxBalance: { $max: "$Acc_Bal" }
        }
    }
]);
```