NEXT GENERATION TECHNOLOGY
PRACTICAL JOURNAL

NAME – ARMAN KHAN

ROLLL NO - 13

CLASS – TYBSc.IT
ACADEMIC YEAR – 2021-21

PRACTICAL NO – 1

MONGODB BASICS

a. Write a MongoDB query to create and drop database.

b. Write a MongoDB query to create, display and drop collection.

c. Write a MongoDB query to insert, query, update and delete a document.

AIM 1.a) Write a MongoDB query to create and drop database.

Syntax :

Create –

use database_name

Drop –

db.dropDatabase()

NAME - KHAN ARMAN
ROLL NO - 13

Source Code:

Use mydb

db.dropDatabase()

OUTPUT :

Create database -

```
> use mydb
switched to db mydb
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
mydb    0.000GB
mylib   0.000GB
```

Drop Database –

```
> use mylib
switched to db mylib
> db.dropDatabase()
{ "ok" : 1 }
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
>
```

NAME - KHAN ARMAN
ROLL NO - 13

AIM 1.B) Write a MongoDB query to create, display and drop collection.

Syntax:

Create -

db.createCollection(collection_name)
db.collection_name.insert({key:value})

Display –

db.collection_name.find()

Drop -

db.collection_name.drop()

Source code:

use myda

db.createCollection("students")

db.students.insert({Name:"Arman",RollNo:"13",Class:"A",Gender:"M"})

show collections

db.students.find()

db.students.drop()

NAME - KHAN ARMAN
ROLL NO - 13

OUTPUT :

```
> use myda
switched to db myda
> db.createCollection("student")
{ "ok" : 1 }
> db.student.insert({Name:"Arman",RollNo:"13",Class:"A",Gender:"M"})
WriteResult({ "nInserted" : 1 })
```

```
> show collections
student
> db.student.find()
{ "_id" : ObjectId("6165a11a5dbfe63df344b81f"), "Name" : "Arman", "RollNo" : "13", "Class" : "A", "Gender" : "M" }
> db.student.drop()
true
>
```

NAME - KHAN ARMAN
ROLL NO - 13

AIM 1.C) Write a MondoDB query to insert, query, update and delete a document.

Syntax:

To Insert Document:

db.COLLECTION_NAME.insert(document)

To Query Document:

db.COLLECTION_NAME.ine()

To Update Document:

db.COLLECTION_NAME.update(SELECTION_CRITERIA,UPDATED DATA)

To Delete Document:

db.COLLECTION_NAME.remove(DELETION_CRITERIA)

OUTPUT :

Insert documentation_-

```
> use tyit
switched to db tyit
> db.student.insert({
... regNo:"123456",
... name:"arman",
... course:{CourseName:"BscIT",duration:"3 Year"},
... address:{city:"Mumbai",State:"MH", country:"india"},
... })
WriteResult({ "nInserted" : 1 })
>
```

Query Document –

```
> db.student.find().pretty()
{
        "_id" : ObjectId("6165a408412f4ad4bf3b7101"),
        "regNo" : "123456",
        "name" : "arman",
        "course" : {
                "CourseName" : "BscIT",
                "duration" : "3 Year"
        },
        "address" : {
                "city" : "Mumbai",
                "State" : "MH",
                "country" : "india"
        }
}
>
```

UPDATE DOCUMENT -

```
> db.student.update({
... regNo:"123456"
... },
... {
... $set:
... {"name":"khan arman"}
... })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.student.find().pretty()
{
        "_id" : ObjectId("6165a408412f4ad4bf3b7101"),
        "regNo" : "123456",
        "name" : "khan arman",
        "course" : {
                "CourseName" : "BscIT",
                "duration" : "3 Year"
        },
        "address" : {
                "city" : "Mumbai",
                "State" : "MH",
                "country" : "india"
        }
}
>
```

NAME - KHAN ARMAN
ROLL NO - 13

DELETE DOCUMENT -

```
> db.student.remove({"regNo":"123456"})
WriteResult({ "nRemoved" : 1 })
```

NAME - KHAN ARMAN
ROLL NO - 13

PRACTICAL NO:02

SIMPLE QUERIES WITH MONGODB

1) Selector –

Syntax :

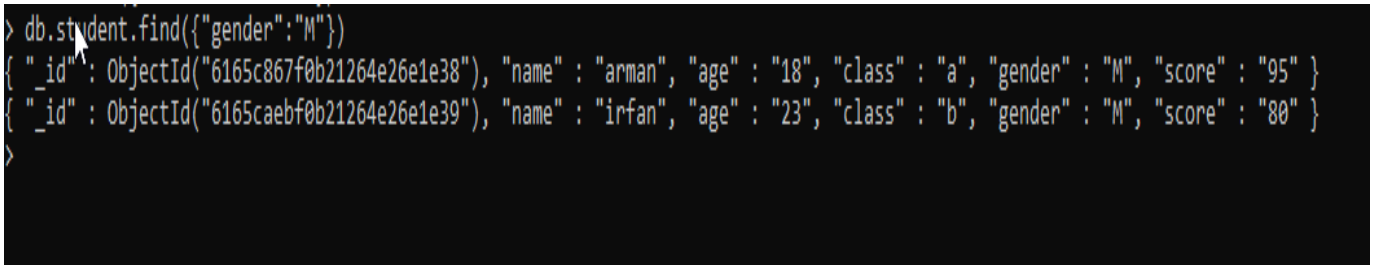
db.collection_name.find({"Key":"Value"})

Source code:

Use tyit

db.student.find({"Gender":"M"})

OUTPUT :



2) Projector -

Syntax:

db.collection_name.find({"Key":"Value"},
{"Key":Value,"Key":Value})

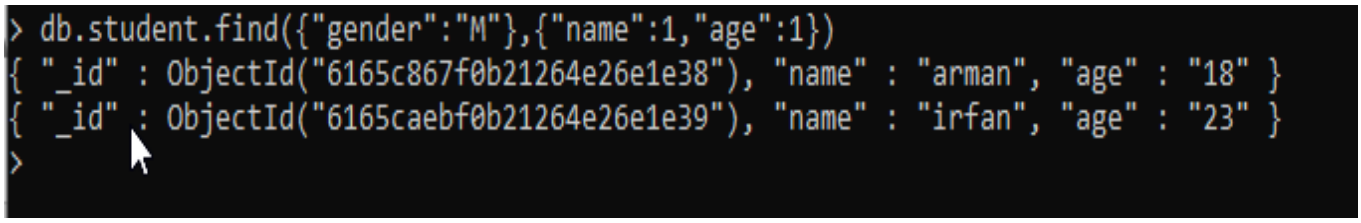
NAME - KHAN ARMAN
ROLL NO - 13

Source code:

Use MYBase

db.student.find({"Gender":"M"},{"Name":1,"Age":1})

Output:



3) sort() -

Syntax:

db.collection_name.find({"Key":"Value"},
{"Key":Value,"Key":Value}).sort({"Key":Value})

Source code:

Use MYBase

db.stud.find({"Gender":"F"},{"Name":1,"Age":1}).sort({"Age":

```
> db.student.find({"gender":"M"},{"name":1,"age":1}).sort({"age":1})
{ "_id" : ObjectId("6165c867f0b21264e26e1e38"), "name" : "arman", "age" : "18" }
{ "_id" : ObjectId("6165caebf0b21264e26e1e39"), "name" : "irfan", "age" : "23" }
> db.student.find({"gender":"M"},{"name":1,"age":1}).sort({"age":-1})
{ "_id" : ObjectId("6165caebf0b21264e26e1e39"), "name" : "irfan", "age" : "23" }
{ "_id" : ObjectId("6165c867f0b21264e26e1e38"), "name" : "arman", "age" : "18" }
>
```

OUTPUT:

```
> db.student.find({"gender":"M"},{"name":1,"age":1}).sort({"age":1})
{ "_id" : ObjectId("6165c867f0b21264e26e1e38"), "name" : "arman", "age" : "18" }
{ "_id" : ObjectId("6165caebf0b21264e26e1e39"), "name" : "irfan", "age" : "23" }
> db.student.find({"gender":"M"},{"name":1,"age":1}).sort({"age":-1})
{ "_id" : ObjectId("6165caebf0b21264e26e1e39"), "name" : "irfan", "age" : "23" }
{ "_id" : ObjectId("6165c867f0b21264e26e1e38"), "name" : "arman", "age" : "18" }
>
```

4) Limit() -

Syntax:

db.collection_name.find({"Key":"Value",$or:[{"Key":"Value"},{"Key":" Value"}]}).limit(Value)

Source code:

Use tyit

db.stud.find({"gender":"F",$or:[{"class":"a"},{"score":"70"}]}).limit(2)

Output:

```
> db.student.find({"gender":"F",$or:[{"class":"a"},{"score":"70"}]}).limit(1)
{ "_id" : ObjectId("6165cc00f0b21264e26e1e3a"), "name" : "karina", "age" : "20", "class" : "a", "gender" : "F", "score" : "60" }
>
```

5) Skip() -

Syntax:

db.collection_name.find({"Key":"Value",$or:[{"Key":"Value"},
{"Key":" Value"}]}).limit(Value).skip(Value)

Source code:

Use tyit

db.stud.find({"Gender":"F",$or:[{"class":"a"},
{"score":"70"}]}).limit(2).skip(2)

Output:

```
> db.student.find({"gender":"F",$or:[{"class":"a"},{"score":"70"}]}).limit(2).skip(2)
{ "_id" : ObjectId("616677448255156e2c4b906a"), "nmae" : "nishat", "age" : "20", "class" : "a", "gender" : "F", "score" : "80" }
>
```

6) Findone() -

Syntax:

db.collection_name.findOne({"Key":"Value"},

{"Key":Value,"Key":Value})

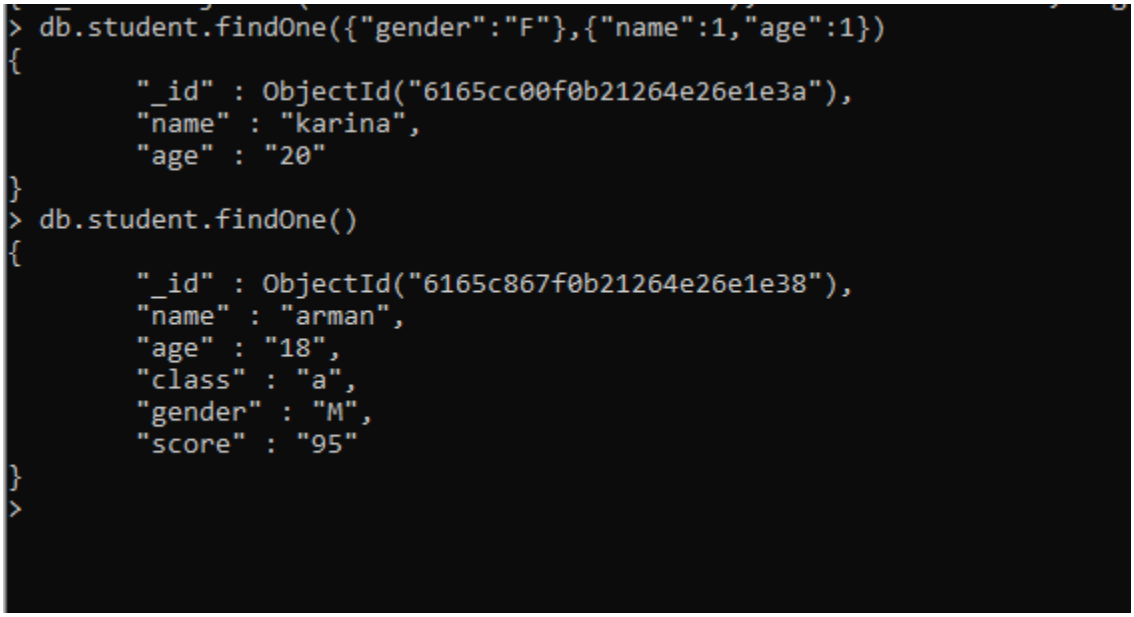
db.collection_name.findOne()

Source code:

Use tyit

db.stud.findOne({"gender":"F"}, {"Name":1,"Age":1})
db.stud.findOne()

Output:



7) ensureIndex

Syntax:

db.collection_name.ensureIndex({"Key":"Value"})

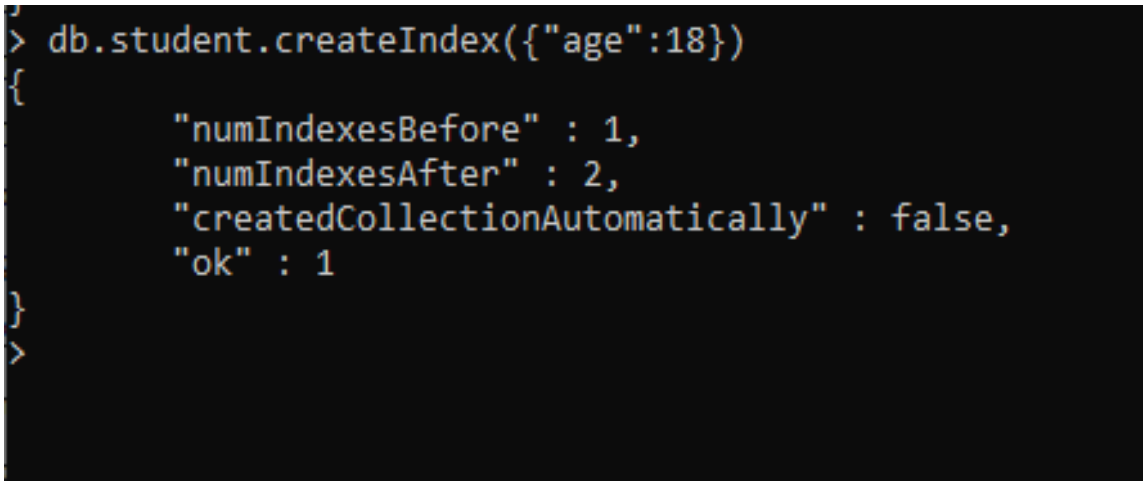
Source Code:

Use mydatabase

Show collections

db.example.ensureIndex({“age”:26})

Output:



8) Pretty() -

Syntax :

db.collection_name.find().pretty()

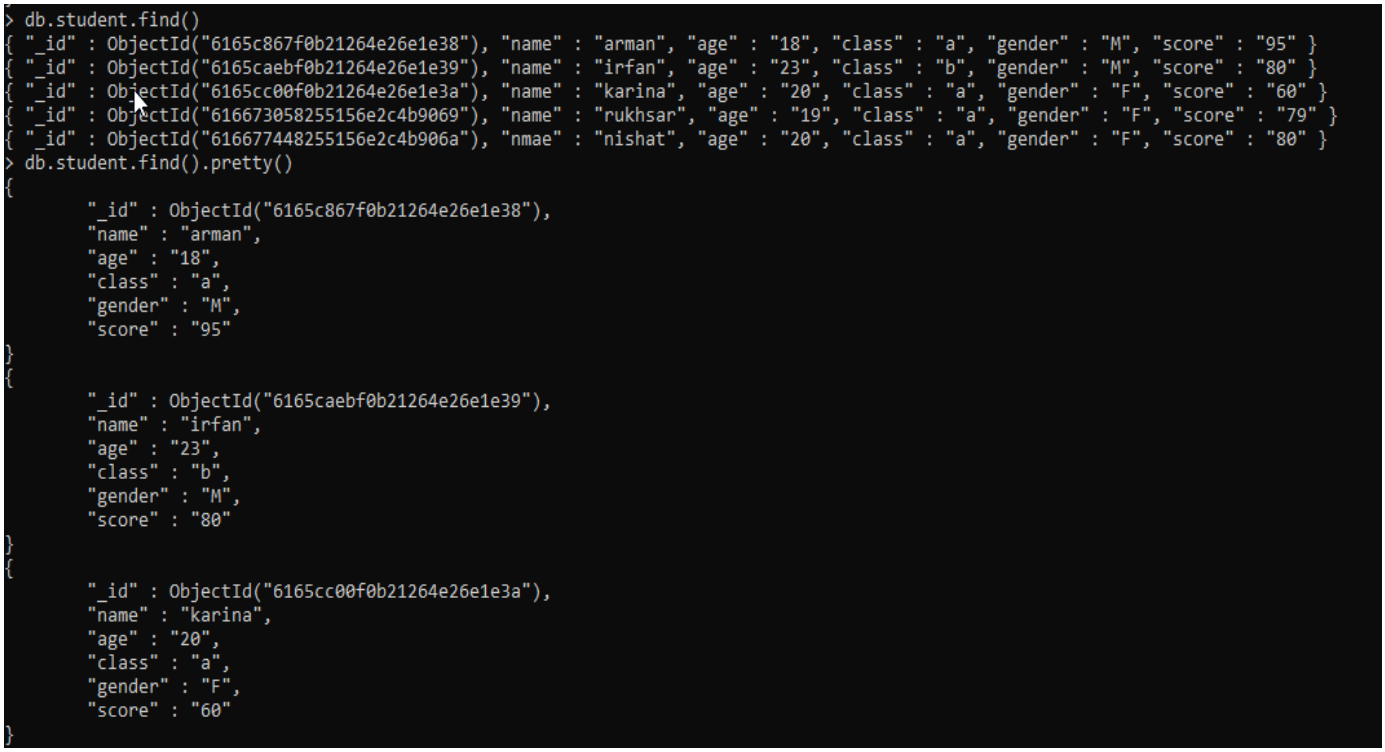
Source Code:

use tyit

show collections

db.user1.find() db.user1.find.pretty()

Output:



9) Conditional Operators -

Syntax:

$lt and $lte

db.collection_name.find({"Key":{"$lt":Value}})
db.collection_name.find({"Key":{"$lte":Value}})

$gt and $gte

db.collection_name.find({"Key":{"$gt":Value}})
db.collection_name.find({"Key":{"$gte":Value}})

NAME - KHAN ARMAN
ROLL NO - 13

$in and $nin

db.collection_name.find({"Key":{"$in":["Value","Value"]}})
db.collection_name.find({"Key":{"$nin":["Value","Value"]}})

Source code:

```
use tyit

db.createCollection(“stud”)

db.stud.insert({Name:"S1",Age:25,Gender:"M",Class:"C1",Score:95})
db.stud.insert({Name:"S2",Age:18,Gender:"M",Class:"C1",Score:85})
db.stud.insert({Name:"S3",Age:18,Gender:"F",Class:"C1",Score:85})
db.stud.insert({Name:"S4",Age:18,Gender:"F",Class:"C1",Score:75})
db.stud.insert({Name:"S5",Age:18,Gender:"F",Class:"C2",Score:75})
db.stud.insert({Name:"S6",Age:21,Gender:"M",Class:"C2",Score:100})

db.stud.insert({Name:"S7",Age:21,Gender:"M",Class:"C2",Score:100})

db.stud.insert({Name:"S8",Age:25,Gender:"F",Class:"C2",Score:100})
db.stud.insert({Name:"S9",Age:25,Gender:"F",Class:"C2",Score:90})
```

db.stud.insert({Name:"S10",Age:28,Gender:"F",Class:"C3",Score:90})

db.stud.find()

$lt and $lte db.stud.find({"Age":{"$lt":25}})
db.stud.find({"Age":{"$lte":25}})

$gt and $gte db.stud.find({"Age":{"$gt":25}})
db.stud.find({"Age":{"$gte":25}})

$in and $nin

db.stud.find({"Class":{"$in":["C1","C2"]}})
db.stud.find({"Class":{"$nin":["C1","C2"]}})

Output:

```
> use tyit
switched to db tyit
> db.createCollection(“stud”)
uncaught exception: SyntaxError: illegal character :
@(shell):1:20
> db.stud.insert({Name:"S1",Age:25,Gender:"M",Class:"C1",Score:95})
WriteResult({ "nInserted" : 1 })
> db.stud.insert({Name:"S2",Age:18,Gender:"M",Class:"C1",Score:85})
WriteResult({ "nInserted" : 1 })
> db.stud.insert({Name:"S3",Age:18,Gender:"F",Class:"C1",Score:85})
WriteResult({ "nInserted" : 1 })
> db.stud.insert({Name:"S4",Age:18,Gender:"F",Class:"C1",Score:75})
WriteResult({ "nInserted" : 1 })
> db.stud.insert({Name:"S5",Age:18,Gender:"F",Class:"C2",Score:75})
WriteResult({ "nInserted" : 1 })
> db.stud.insert({Name:"S6",Age:21,Gender:"M",Class:"C2",Score:100})
WriteResult({ "nInserted" : 1 })
> db.stud.insert({Name:"S7",Age:21,Gender:"M",Class:"C2",Score:100})
WriteResult({ "nInserted" : 1 })
> db.stud.insert({Name:"S8",Age:25,Gender:"F",Class:"C2",Score:100})
WriteResult({ "nInserted" : 1 })
> db.stud.insert({Name:"S9",Age:25,Gender:"F",Class:"C2",Score:90})
WriteResult({ "nInserted" : 1 })
> db.stud.insert({Name:"S10",Age:28,Gender:"F",Class:"C3",Score:90})
WriteResult({ "nInserted" : 1 })
> db.stud.find()
{ "_id" : ObjectId("61668f228255156e2c4b906b"), "Name" : "S7", "Age" : 21, "Gender" : "M", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906c"), "Name" : "S1", "Age" : 25, "Gender" : "M", "Class" : "C1", "Score" : 95 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906d"), "Name" : "S2", "Age" : 18, "Gender" : "M", "Class" : "C1", "Score" : 85 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906e"), "Name" : "S3", "Age" : 18, "Gender" : "F", "Class" : "C1", "Score" : 85 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906f"), "Name" : "S4", "Age" : 18, "Gender" : "F", "Class" : "C1", "Score" : 75 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9070"), "Name" : "S5", "Age" : 18, "Gender" : "F", "Class" : "C2", "Score" : 75 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9071"), "Name" : "S6", "Age" : 21, "Gender" : "M", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9072"), "Name" : "S7", "Age" : 21, "Gender" : "M", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9073"), "Name" : "S8", "Age" : 25, "Gender" : "F", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9074"), "Name" : "S9", "Age" : 25, "Gender" : "F", "Class" : "C2", "Score" : 90 }
{ "_id" : ObjectId("61668f928255156e2c4b9075"), "Name" : "S10", "Age" : 28, "Gender" : "F", "Class" : "C3", "Score" : 90 }
>
```

NAME - KHAN ARMAN
ROLL NO - 13

$lt and $lte :

```
> db.stud.find({"Age":{"$lt":25}})
{ "_id" : ObjectId("61668f228255156e2c4b906b"), "Name" : "S7", "Age" : 21, "Gender" : "M", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906d"), "Name" : "S2", "Age" : 18, "Gender" : "M", "Class" : "C1", "Score" : 85 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906e"), "Name" : "S3", "Age" : 18, "Gender" : "F", "Class" : "C1", "Score" : 85 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906f"), "Name" : "S4", "Age" : 18, "Gender" : "F", "Class" : "C1", "Score" : 75 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9070"), "Name" : "S5", "Age" : 18, "Gender" : "F", "Class" : "C2", "Score" : 75 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9071"), "Name" : "S6", "Age" : 21, "Gender" : "M", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9072"), "Name" : "S7", "Age" : 21, "Gender" : "M", "Class" : "C2", "Score" : 100 }
> db.stud.find({"Age":{"$lte":25}})
{ "_id" : ObjectId("61668f228255156e2c4b906b"), "Name" : "S7", "Age" : 21, "Gender" : "M", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906c"), "Name" : "S1", "Age" : 25, "Gender" : "M", "Class" : "C1", "Score" : 95 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906d"), "Name" : "S2", "Age" : 18, "Gender" : "M", "Class" : "C1", "Score" : 85 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906e"), "Name" : "S3", "Age" : 18, "Gender" : "F", "Class" : "C1", "Score" : 85 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906f"), "Name" : "S4", "Age" : 18, "Gender" : "F", "Class" : "C1", "Score" : 75 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9070"), "Name" : "S5", "Age" : 18, "Gender" : "F", "Class" : "C2", "Score" : 75 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9071"), "Name" : "S6", "Age" : 21, "Gender" : "M", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9072"), "Name" : "S7", "Age" : 21, "Gender" : "M", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9073"), "Name" : "S8", "Age" : 25, "Gender" : "F", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9074"), "Name" : "S9", "Age" : 25, "Gender" : "F", "Class" : "C2", "Score" : 90 }
>
```

$gt and $gte

```
> db.stud.find({"Age":{"$gt":25}})
{ "_id" : ObjectId("61668f928255156e2c4b9075"), "Name" : "S10", "Age" : 28, "Gender" : "F", "Class" : "C3", "Score" : 90 }
> db.stud.find({"Age":{"$gte":25}})
{ "_id" : ObjectId("61668f8e8255156e2c4b906c"), "Name" : "S1", "Age" : 25, "Gender" : "M", "Class" : "C1", "Score" : 95 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9073"), "Name" : "S8", "Age" : 25, "Gender" : "F", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9074"), "Name" : "S9", "Age" : 25, "Gender" : "F", "Class" : "C2", "Score" : 90 }
{ "_id" : ObjectId("61668f928255156e2c4b9075"), "Name" : "S10", "Age" : 28, "Gender" : "F", "Class" : "C3", "Score" : 90 }
>
```

$in and $nin

```
> db.stud.find({"Class":{"$in":["C1","C2"]}})
{ "_id" : ObjectId("61668f228255156e2c4b906b"), "Name" : "S7", "Age" : 21, "Gender" : "M", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906c"), "Name" : "S1", "Age" : 25, "Gender" : "M", "Class" : "C1", "Score" : 95 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906d"), "Name" : "S2", "Age" : 18, "Gender" : "M", "Class" : "C1", "Score" : 85 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906e"), "Name" : "S3", "Age" : 18, "Gender" : "F", "Class" : "C1", "Score" : 85 }
{ "_id" : ObjectId("61668f8e8255156e2c4b906f"), "Name" : "S4", "Age" : 18, "Gender" : "F", "Class" : "C1", "Score" : 75 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9070"), "Name" : "S5", "Age" : 18, "Gender" : "F", "Class" : "C2", "Score" : 75 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9071"), "Name" : "S6", "Age" : 21, "Gender" : "M", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9072"), "Name" : "S7", "Age" : 21, "Gender" : "M", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9073"), "Name" : "S8", "Age" : 25, "Gender" : "F", "Class" : "C2", "Score" : 100 }
{ "_id" : ObjectId("61668f8e8255156e2c4b9074"), "Name" : "S9", "Age" : 25, "Gender" : "F", "Class" : "C2", "Score" : 90 }
> db.stud.find({"Class":{"$nin":["C1","C2"]}})
{ "_id" : ObjectId("61668f928255156e2c4b9075"), "Name" : "S10", "Age" : 28, "Gender" : "F", "Class" : "C3", "Score" : 90 }
>
```

NAME - KHAN ARMAN
ROLL NO - 13

PRACTICAL NO:03

IMPLEMENTING AGGREGATION

a.Write a MongoDB query to use sum, avg, min and max expression.

b.Write a MongoDB query to use push and addToSet expression.

c.Write a MongoDB query to use first and last expression.

3) a - Write a MongoDB query to use sum, avg, min and max expression.

Syntax:

Sum

{$sum:[<expression1>,<expression>…]}

Avg

{$avg:[<expression1>…]}

Min

{$min:[<expression1>…]}

Max

{$Max:[<expression1>…]}

NAME - KHAN ARMAN
ROLL NO - 13

Source code:

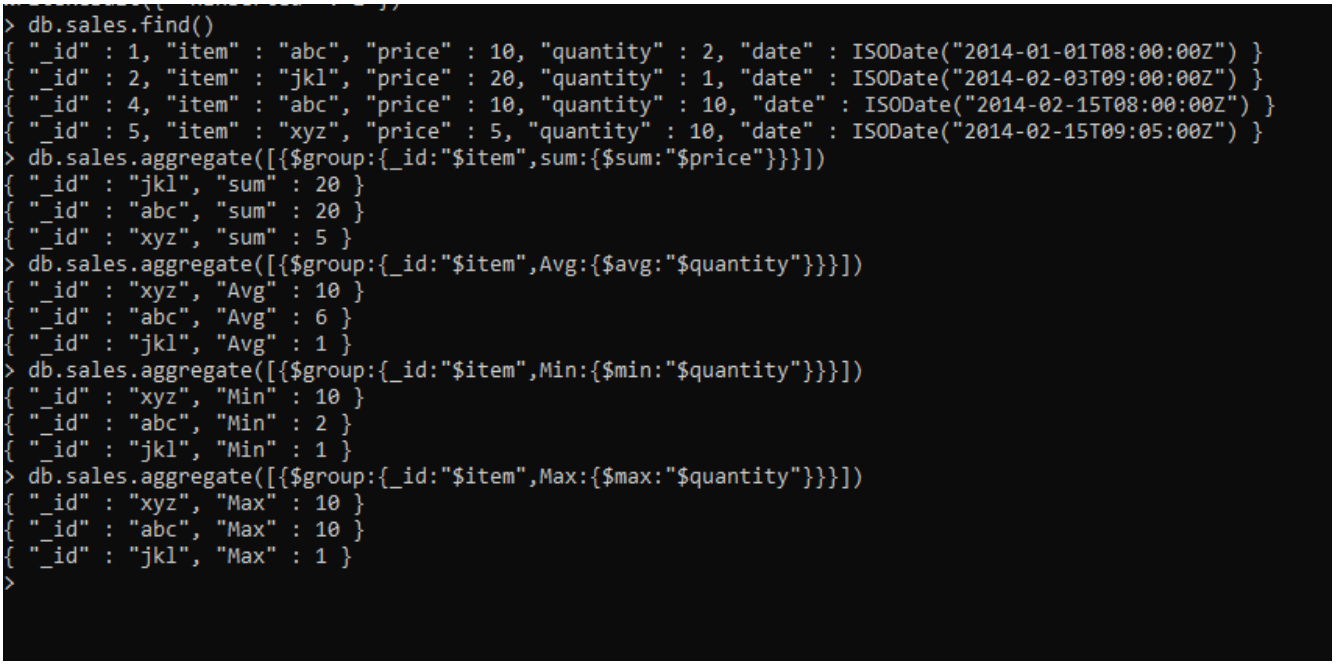
```
use minimum

db.createCollection(“Sales”)

db.Sales.insert({ "_id" : 1, "item" : "abc", "price" : 10,
"quantity" : 2,
"date" : ISODate("2014-01-01T08:00:00Z") }

db.Sales.insert { "_id" : 2, "item" : "jkl", "price" : 20,
"quantity" : 1,
"date" : ISODate("2014-02-03T09:00:00Z") }

db.Sales.insert { "_id" : 3, "item" : "xyz", "price" : 5, "quantity"
: 5,
"date" : ISODate("2014-02-03T09:05:00Z") }

db.Sales.insert { "_id" : 4, "item" : "abc", "price" : 10,
"quantity" : 10,
"date" : ISODate("2014-02-15T08:00:00Z") }

db.Sales.insert { "_id" : 5, "item" : "xyz", "price" : 5, "quantity"
: 10,
"date" : ISODate("2014-02-15T09:05:00Z") }

db.sales.aggregate([{$group:{_id:"$item",sum:{$sum:"$price"

}}}])

db.sales.aggregate([{$group:{_id:"$item",Avg:{$avg:"$quantit

y"}}}])

db.sales.aggregate([{$group:{_id:"$item",Min:{$min:"$quanti

ty"}}}])
```

NAME - KHAN ARMAN
ROLL NO - 13

db.sales.aggregate([{$group:{_id:"$item",Max:{$max:"$quant

ity"}}}])

OUTPUT :



3.b.Write a MongoDB query to use push and addToSet

expression.

Syntax:

Push

{$push:<expression>} addToSet

{$addToSet:<expression>}

Source code:

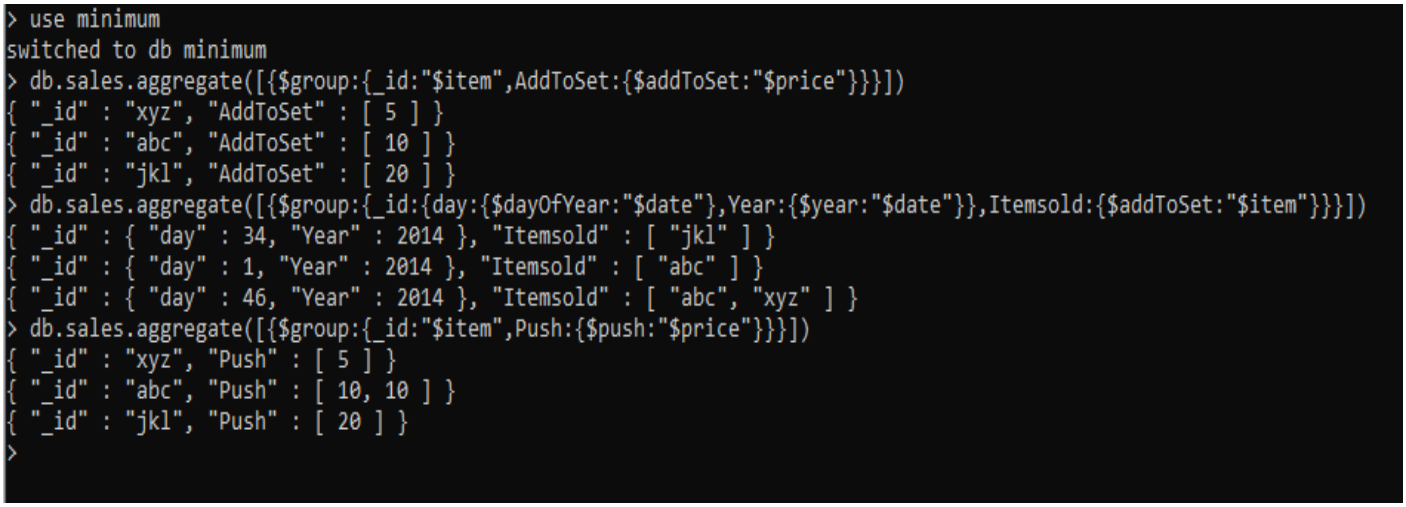
```
use minimum
db.createCollection(“sales”)
db.sales.insert({ "_id" : 1, "item" : "abc", "price" : 10,
"quantity" : 2,
"date" : ISODate("2014-01-01T08:00:00Z") }
db.sales.insert { "_id" : 2, "item" : "jkl", "price" : 20,
"quantity" : 1,
"date" : ISODate("2014-02-03T09:00:00Z") }
db.sales.insert { "_id" : 3, "item" : "xyz", "price" : 5, "quantity"
: 5,
"date" : ISODate("2014-02-03T09:05:00Z") }
db.sales.insert { "_id" : 4, "item" : "abc", "price" : 10,
"quantity" : 10,
"date" : ISODate("2014-02-15T08:00:00Z") }
db.sales.insert { "_id" : 5, "item" : "xyz", "price" : 5, "quantity"
: 10,
"date" : ISODate("2014-02-15T09:05:00Z") }
db.sales.aggregate([{$group:{_id:"$item",AddToSet:{$addToS
et:"$pri ce"}}}])
db.sales.aggregate([{$group:{_id:{day:{$dayOfYear:"$date"},Y
ear:{$y ear:"$date"}},Itemsold:{$addToSet:"$item"}}}])
```

NAME - KHAN ARMAN
ROLL NO - 13

db.sales.aggregate([{$group:{_id:"$item",Push:{$push:"$price

"}}}])

OUTPUT :



3.c. Write a MongoDB query to use first and last expression.

Syntax:

First

{$first:<expression>}

Last

{$last:<expression>}

Source code:

use minimum

db.createCollection(“Sales”)

db.Sales.insert({ "_id" : 1, "item" : "abc", "price" : 10,

"quantity" : 2,

"date" : ISODate("2014-01-01T08:00:00Z") }

db.Sales.insert { "_id" : 2, "item" : "jkl", "price" : 20,

"quantity" : 1,

"date" : ISODate("2014-02-03T09:00:00Z") }

db.Sales.insert { "_id" : 3, "item" : "xyz", "price" : 5, "quantity"

: 5,

"date" : ISODate("2014-02-03T09:05:00Z") }

db.Sales.insert { "_id" : 4, "item" : "abc", "price" : 10,

"quantity" : 10,

"date" : ISODate("2014-02-15T08:00:00Z") }

db.Sales.insert { "_id" : 5, "item" : "xyz", "price" : 5, "quantity"

: 10,

"date" : ISODate("2014-02-15T09:05:00Z") }

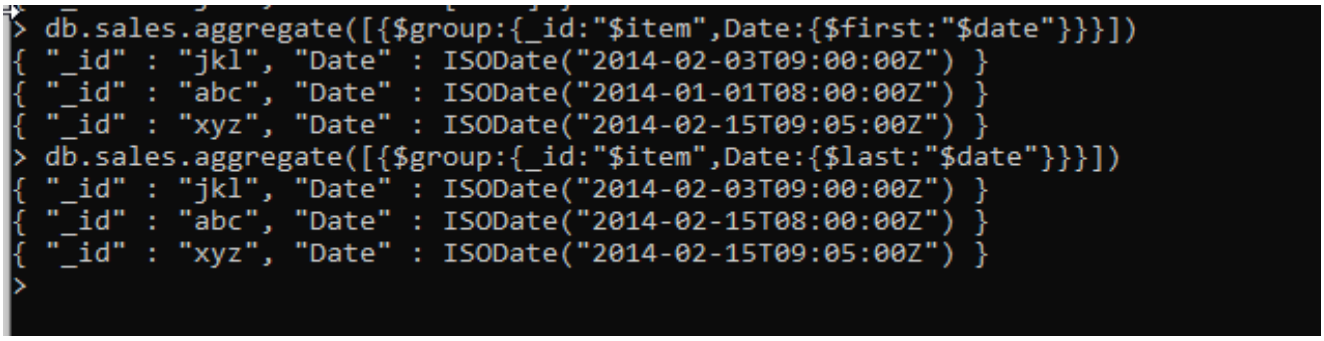
db.sales.aggregate([{$group:{_id:"$item",Date:{$first:"$date"

}}}])

db.sales.aggregate([{$group:{_id:"$item",Date:{$last:"$date"}

}}])

OUTPUT :



NAME - KHAN ARMAN
ROLL NO - 13

PRACTICAL NO:07

PYTHON AND MONGODB

a.Connecting Python with MongoDB and inserting, retrieving, updating and deleting.

a.Connecting Python with MongoDB and inserting, retrieving, updating and deleting.

1) Inserting & Retrieving

Source code:

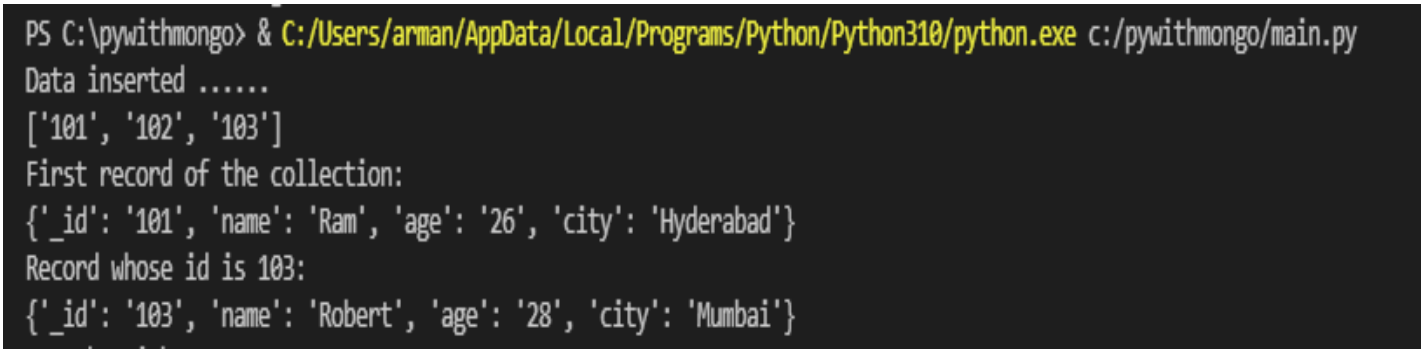
```
from pymongo import
MongoClient
 #Creating a pymongo client
client =
MongoClient('localhost',
27017)
#Getting the database
#instance
db = client['mydatabase']
#Creating a collection
coll = db['example']
```

NAME - KHAN ARMAN
ROLL NO - 13

```
#Inserting document into a collection
data = [ {"_id": "101", "name": "Ram",
"age": "26", "city":
"Hyderabad"}, {"_id": "102", "name": "Rahim", "age":
"27", "city":
"Bangalore"}, {"_id": "103", "name": "Robert", "age":
"28", "city":
"Mumbai"} ]
res = coll.insert_many(data)
print("Data inserted ......")
print(res.inserted_ids)
#Retrieving the first record using the
find_one() method  print("First record of the
collection: ")  print(coll.find_one())
#Retrieving a record with is 103 using
#thefind_one()
 method print("Record whose id is 103: ")
print(coll.find_one({"_id": "103"}))
```

NAME - KHAN ARMAN
ROLL NO - 13

OUTPUT :



2.Update

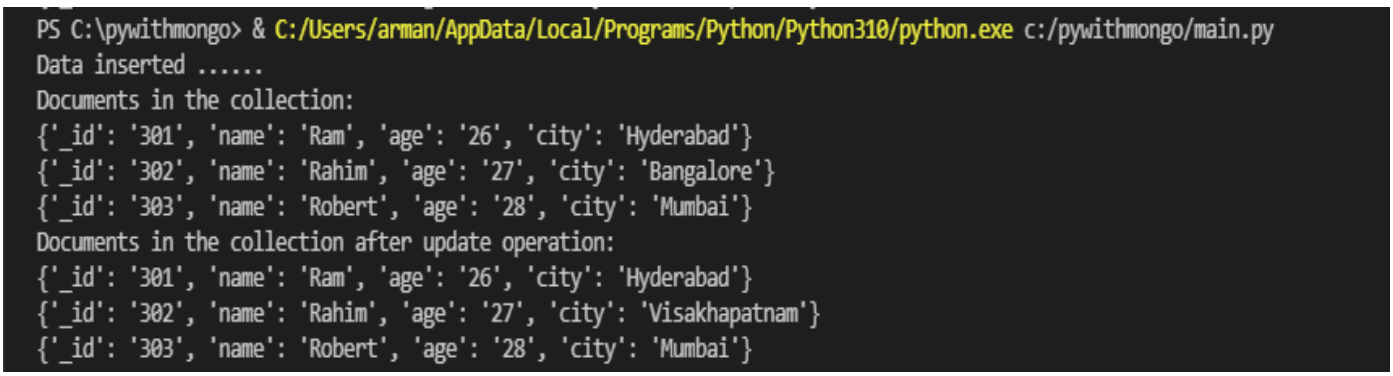
a.Update One

Source code:

```
from pymongo import MongoClient

#Creating a pymongo client

client = MongoClient('localhost', 27017)

#Getting the database instance

db = client['myDBase']

#Creating a collection

coll = db['MYExample2']

#Inserting document into a collection

data = [

 {"_id": "301", "name": "Ram", "age": "26", "city":

"Hyderabad"},

 {"_id": "302", "name": "Rahim", "age": "27", "city":
```

```
"Bangalore"},
 {"_id": "303", "name": "Robert", "age": "28", "city":
"Mumbai"}
]
res = coll.insert_many(data)
print("Data inserted ......")
#Retrieving all the records using the find() method
print("Documents in the collection: ")
 for doc1 in coll.find():
    print(doc1)
coll.update_one({"_id":"302"},{"$set":{"city":"Visakhapatnam"}
})
#Retrieving all the records using the find() method
print("Documents in the collection after update operation: ")
for doc2 in coll.find():
   print(doc2)
```

OUTPUT :



NAME - KHAN ARMAN

ROLL NO - 13

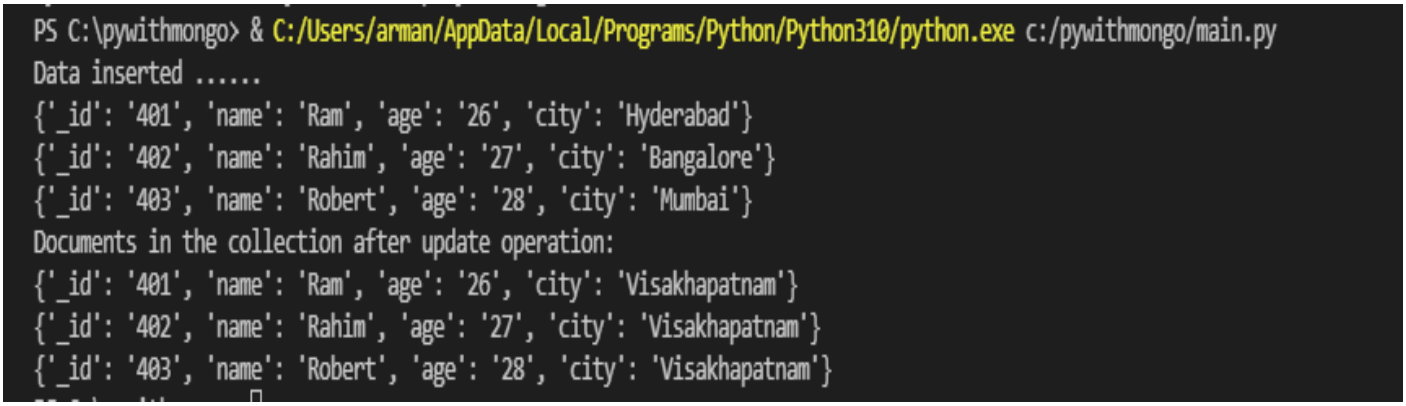
b) Update Many

Source code:

```
from pymongo import MongoClient
 #Creating a pymongo client
client = MongoClient('localhost', 27017)
#Getting the database instance
db = client['MYDataB']
#Creating a collection
coll = db['MYexample5']
#Inserting document into a collection
data = [
 {"_id": "401", "name": "Ram", "age": "26", "city":
"Hyderabad"},
 {"_id": "402", "name": "Rahim", "age": "27", "city":
"Bangalore"},
 {"_id": "403", "name": "Robert", "age": "28", "city":
"Mumbai"}
]
res = coll.insert_many(data)
print("Data inserted ......")
```

```
#Retrieving all the records using the find() method
print("Documents in the collection: ")
 for doc1 in coll.find():
    print(doc1)
coll.update_many({},{"$set":{"city":"Visakhapatnam"}})
#Retrieving all the records using the find() method
print("Documents in the collection after update operation: ")
for doc2 in coll.find():
 print(doc2)
```

OUTPUT :



3.Delete

a.Delete One

Source code:

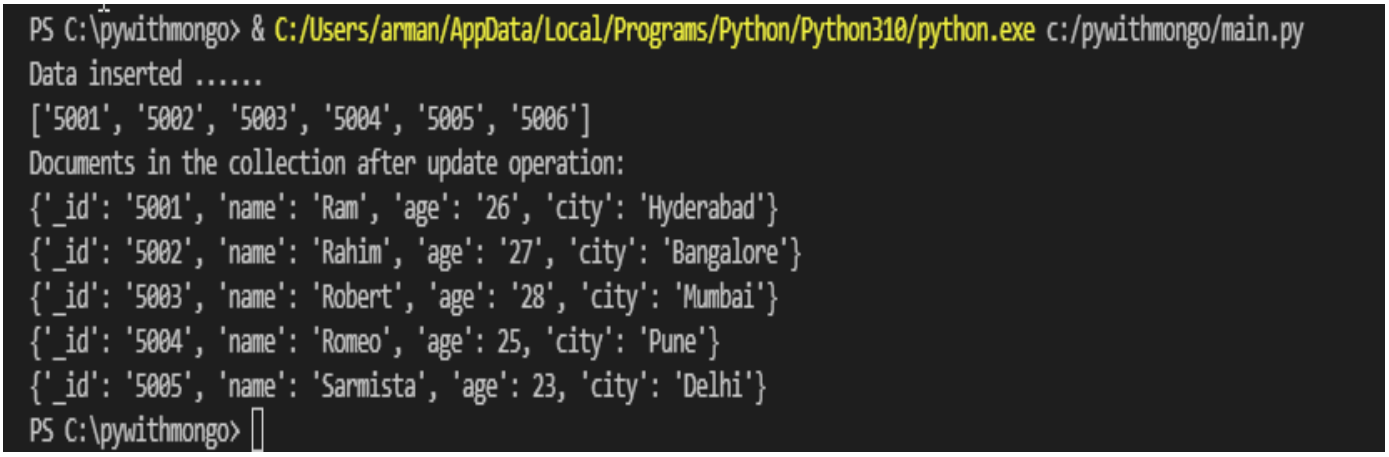
```
from pymongo import MongoClient
#Creating a pymongo client
client = MongoClient('localhost', 27017)
```

```
#Getting the database instance
db = client['mydatabase']
#Creating a collection
coll = db['Myexample']
#Inserting document into a collection
data = [
 {"_id": "5001", "name": "Ram", "age": "26", "city":
"Hyderabad"},
 {"_id": "5002", "name": "Rahim", "age": "27", "city":
"Bangalore"},
 {"_id": "5003", "name": "Robert", "age": "28", "city":
"Mumbai"},
 {"_id": "5004", "name": "Romeo", "age": 25, "city": "Pune"},
{"_id": "5005", "name": "Sarmista", "age": 23, "city":
"Delhi"},
 {"_id": "5006", "name": "Rasajna", "age": 26, "city":
"Chennai"}
]
res = coll.insert_many(data)
print("Data inserted ......")
```

NAME - KHAN ARMAN
ROLL NO - 13

```
print(res.inserted_ids)

#Deleting one document

coll.delete_one({"_id" : "5006"})

#Retrieving all the records using the find() method
print("Documents in the collection after update operation: ")
for doc2 in coll.find():

print(doc2)
```

OUTPUT :



b.Delete Many

Source code:

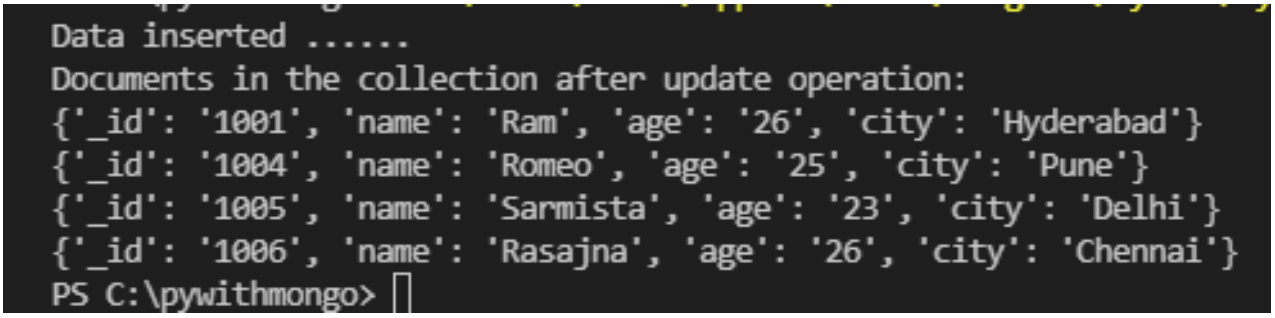
```
from pymongo import

MongoClient

#Creating a pymongo client

client =

MongoClient('localhost',
```

```
27017) #Getting the
database instance db =
client['sampleDB'] #Creating
a collection coll =
db['example 4']
#Inserting document into
 data = [
 {"_id": "1001", "name": "Ram", "age": "26", "city":
"Hyderabad"},
 {"_id": "1002", "name": "Rahim", "age": "27", "city":
"Bangalore"},
 {"_id": "1003", "name": "Robert", "age": "28", "city":
"Mumbai"},
 {"_id": "1004", "name": "Romeo", "age": "25", "city":
"Pune"},  {"_id": "1005", "name": "Sarmista", "age":
"23", "city":
"Delhi"},
 {"_id": "1006", "name": "Rasajna", "age": "26", "city":
"Chennai"}
]
```

NAME - KHAN ARMAN
ROLL NO - 13

```
res = coll.insert_many(data)
print("Data inserted ......")
#Deleting multiple documents
coll.delete_many({"age":{"$gt":"26"}})
#Retrieving all the records using the find() method
print("Documents in the collection after update
operation: ")
for doc2 in coll.find():
 print(doc2)
```

OUTPUT :



PRACTICAL NO:05

JAVA WITH MONGODB

1) Inserting Data -

Source Code :

```
package javamongodb; import com.mongodb.DB; import
com.mongodb.MongoClient; import
com.mongodb.client.FindIterable; import
com.mongodb.client.MongoCollection; import
com.mongodb.client.MongoDatabase; import
com.mongodb.client.model.Filters; import java.util.Iterator;
import org.bson.Document;

   public static void main(String[] args) {

      try{

 MongoClient mongoclient=new
MongoClient("localhost",27017);

MongoDatabase db=mongoclient.getDatabase("JAVA");

System.out.println("Connected to Database");

Document document = new Document();
```

```
 document.append("_id","4010");
document.append("name","arman");
document.append("age", "20");
 document.append("city", "Chennai");
document.append("_id","4011");
document.append("name","irfan");
        document.append("age", "20");
        document.append("city", "mumbai");
document.append("_id","4012");
document.append("name","rizwan");
        document.append("age", "21");
        document.append("city", "Delhi");
document.append("_id","4013");
document.append("name","kamru");
        document.append("age", "22");
       document.append("city", "Delhi");
document.append("_id","4014");
document.append("name","sohail");
        document.append("age", "21");
        document.append("city", "Banglore");
       db.getCollection("text").insertOne(document);
       System.out.println("Data Inserted Successfully !!!");
FindIterable iterDoc=db.getCollection("text").find();
        int i=1;
       Iterator it=iterDoc.iterator();
```

```
      while(it.hasNext()){
       System.out.println(it.next());
       i++;
}
}catch(Exception e)
      {
         System.out.println(e);
      }
   }
```

}**OUTPUT :**

```
> show dbs
JAVA        0.000GB
MYDataB     0.000GB
admin       0.000GB
arman       0.000GB
config      0.000GB
local       0.000GB
minimum     0.000GB
minumum     0.000GB
myDBase     0.000GB
mydatabase  0.000GB
sampleDB    0.000GB
tyit        0.000GB
> use JAVA
switched to db JAVA
> show collections
text
> db.text.find()
{ "_id" : "4010", "name" : "arman", "age" : "20", "city" : "Chennai" }
{ "_id" : "4011", "name" : "irfan", "age" : "20", "city" : "mumbai" }
{ "_id" : "4012", "name" : "rizwan", "age" : "21", "city" : "Delhi" }
{ "_id" : "4013", "name" : "kamru", "age" : "21", "city" : "Delhi" }
{ "_id" : "4014", "name" : "sohail", "age" : "21", "city" : "ahmadabad" }
>
```

NAME - KHAN ARMAN

ROLL NO - 13

2) Retreive

Source Code :

```
package javamongodb; import com.mongodb.DB; import
com.mongodb.MongoClient; import
com.mongodb.client.FindIterable; import
com.mongodb.client.MongoCollection; import
com.mongodb.client.MongoDatabase; import
com.mongodb.client.model.Filters; import java.util.Iterator;
import org.bson.Document;

   public static void main(String[] args) {              try{

      MongoClient mongoclient=new
MongoClient("localhost",27017);

      MongoDatabase db=mongoclient.getDatabase("JAVA");

      System.out.println("Connected to Database");
       FindIterable iterDoc=db.getCollection("text").find();
int i=1;

      Iterator it=iterDoc.iterator();       while(it.hasNext()){
System.out.println(it.next());

      i++;       } }catch(Exception e)

{

         System.out.println(e);

      }

   }
```

}

OUTPUT :

```
> Task :compileJava
> Task :processResources NO-SOURCE
> Task :classes

> Task :mongo.main()
Connected to Database
Data Fetched Successfully
Document{{_id=4010, name=arman, age=20, city=Chennai}}
Document{{_id=4011, name=irfan, age=20, city=mumbai}}
Document{{_id=4012, name=rizwan, age=21, city=Delhi}}
Document{{_id=4013, name=kamru, age=21, city=Delhi}}
Document{{_id=4014, name=sohail, age=21, city=ahmadabad}}

Deprecated Gradle features were used in this build, making it incompatible with Gradle 8.0.
```

2) Update

Source Code :

package javamongodb; import com.mongodb.DB; import com.mongodb.MongoClient; import com.mongodb.client.FindIterable; import com.mongodb.client.MongoCollection; import com.mongodb.client.MongoDatabase; import com.mongodb.client.model.Filters; import

```
com.mongodb.client.model.Updates; import java.util.Iterator;
import org.bson.Document; public class update {
   public static void main(String[] args) {        try{
      MongoClient mongoclient=new
MongoClient("localhost",27017);
      MongoDatabase
db=mongoclient.getDatabase("mydatabase");
System.out.println("Connected to Database");

       db.getCollection("example").updateOne(Filters.eq("_id","1
02" ), Updates.set("city", "vishakapatnam"));
      System.out.println("Data Updated Successfully !!!");
FindIterable iterDoc=db.getCollection("example").find();        int
i=1;
Iterator it=iterDoc.iterator(); while(it.hasNext()){
      System.out.println(it.next());        i++;
      }
      }catch(Exception e)
      {
         System.out.println(e);
      }
   }
```

NAME - KHAN ARMAN
ROLL NO - 13

Before update :

```
> show dbs
JAVA        0.000GB
MYDataB     0.000GB
admin       0.000GB
arman       0.000GB
config      0.000GB
local       0.000GB
minimum     0.000GB
minumum     0.000GB
myDBase     0.000GB
mydatabase  0.000GB
sampleDB    0.000GB
tyit        0.000GB
> use mydatabase
switched to db mydatabase
> show collections
Myexample
example
> db.example.find()
{ "_id" : "101", "name" : "Ram", "age" : "26", "city" : "Hyderabad" }
{ "_id" : "102", "name" : "Rahim", "age" : "27", "city" : "Bangalore" }
{ "_id" : "103", "name" : "Robert", "age" : "28", "city" : "Mumbai" }
> show
```

After update :

```
> show dbs
JAVA        0.000GB
MYDataB     0.000GB
admin       0.000GB
arman       0.000GB
config      0.000GB
local       0.000GB
minimum     0.000GB
minumum     0.000GB
myDBase     0.000GB
mydatabase  0.000GB
sampleDB    0.000GB
tyit        0.000GB
> use mydatabase
switched to db mydatabase
> show collections
Myexample
example
> db.example.find()
{ "_id" : "101", "name" : "Ram", "age" : "26", "city" : "Hyderabad" }
{ "_id" : "102", "name" : "Rahim", "age" : "27", "city" : "vishakhapatnam" }
{ "_id" : "103", "name" : "Robert", "age" : "28", "city" : "Mumbai" }
>
```

NAME - KHAN ARMAN
ROLL NO - 13

4) Delete

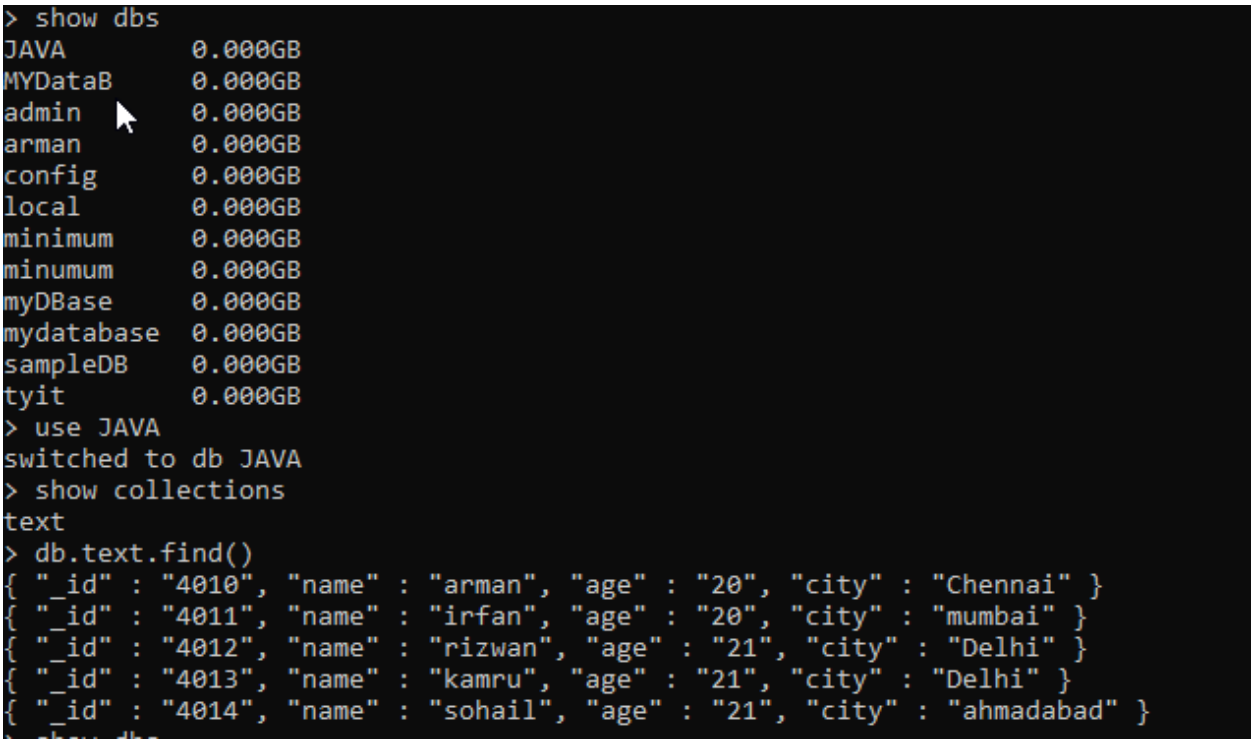
Source Code :

```
package javamongodb; import com.mongodb.DB; import
com.mongodb.MongoClient; import
com.mongodb.client.FindIterable; import
com.mongodb.client.MongoCollection; import
com.mongodb.client.MongoDatabase; import
com.mongodb.client.model.Filters; import java.util.Iterator;
import org.bson.Document; public class delete {

   public static void main(String[] arg)        try{

      MongoClient mongoclient=new
MongoClient("localhost",27017);

      MongoDatabase
db=mongoclient.getDatabase("mydatabase");
System.out.println("Connected to Database");


       db.getCollection("example").deleteOne(Filters.eq("_id","4
009")

);

      FindIterable iterDoc=db.getCollection("example").find();
int i=1;

      Iterator it=iterDoc.iterator();        while(it.hasNext()){

      System.out.println(it.next());        i++;
```

```
        }
        }catch(Exception e)
        {
           System.out.println(e);
        }
    }
}
```

Output :

Before delete :



After Delete :

```
> show dbs
JAVA        0.000GB
MYDataB     0.000GB
admin       0.000GB
arman       0.000GB
config      0.000GB
local       0.000GB
minimum     0.000GB
minumum     0.000GB
myDBase     0.000GB
mydatabase  0.000GB
sampleDB    0.000GB
tyit        0.000GB
> use JAVA
switched to db JAVA
> show collections
text
> db.text.find()
{ "_id" : "4010", "name" : "arman", "age" : "20", "city" : "Chennai" }
{ "_id" : "4012", "name" : "rizwan", "age" : "21", "city" : "Delhi" }
{ "_id" : "4013", "name" : "kamru", "age" : "21", "city" : "Delhi" }
{ "_id" : "4014", "name" : "sohail", "age" : "21", "city" : "ahmadabad" }
>
```

NAME - KHAN ARMAN
ROLL NO - 13

PRACTICAL NO:08

PROGRAMS ON BASIC JQUERY

1) jQuery Basic, jQuery Events.

Jquery basic :

CODE :

mongopractical > 8)1.html > html > body > div > p

```
<html>
<head>
<title>The jQuery Example</title>
</head>
<body>
<div>
<p>This is 1st paragraph.</p>
<p>This is 2nd paragraph.</p>
<p>This is 3rd paragraph.</p>
</div>
</body>
</html>

```

OUTPUT :

This is 1st paragraph.

This is 2nd paragraph.

This is 3rd paragraph.

NAME - KHAN ARMAN
ROLL NO - 13

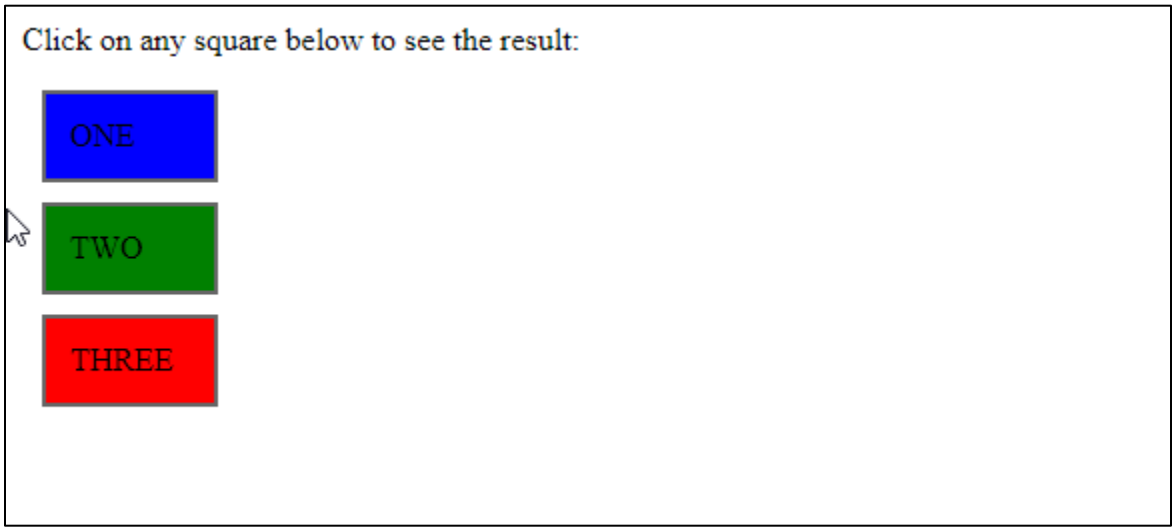
2) JQuery Events

Click Event :

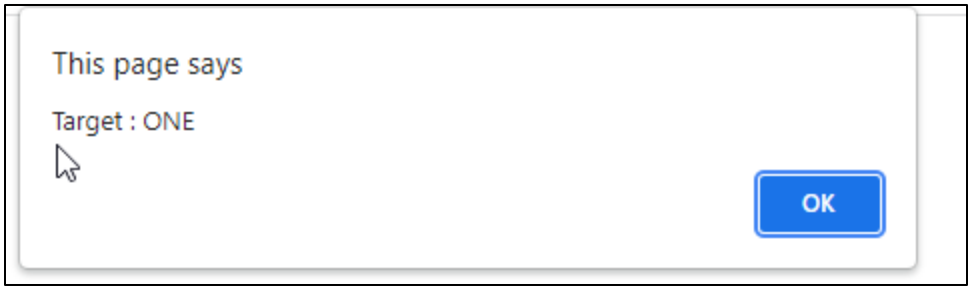
CODE :

```
<html>
<head>
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.5.0/jquery.min.js"></script>
<script type="text/javascript">
$(document).ready(function() {
$('div').bind('click', function( event ){ alert('Event type is ' + event.type); alert('Target : ' + event.target.innerHTML);
});
});
</script>
<style>
.div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}
</style>
</head>
<body>
<p>Click on any square below to see the result:</p>
<div class = "div" style = "background-color:blue;">ONE</div>
<div class = "div" style = "background-color:green;">TWO</div>
<div class = "div" style = "background-color:red;">THREE</div>
</body>
</html>
```

OUTPUT:



NAME - KHAN ARMAN
ROLL NO - 13



3) DoubleClick Event

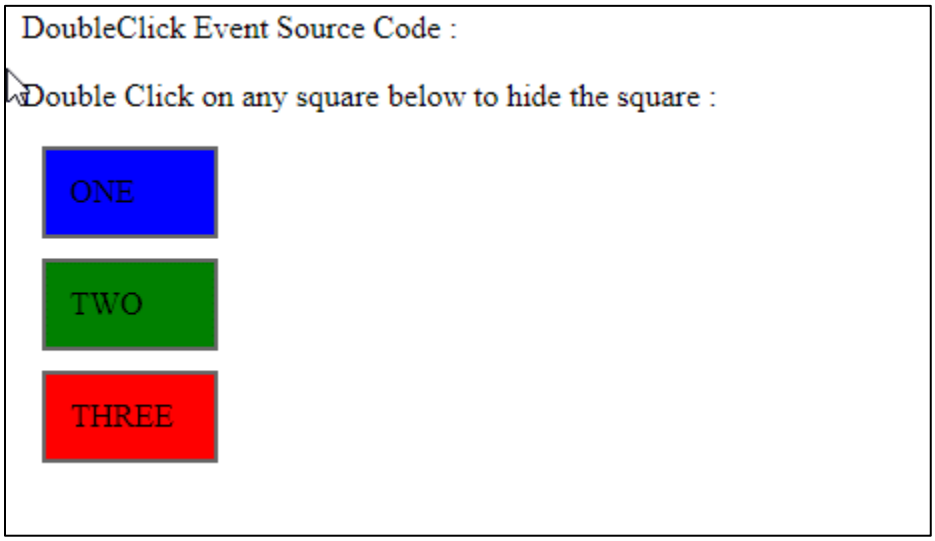
Source Code :

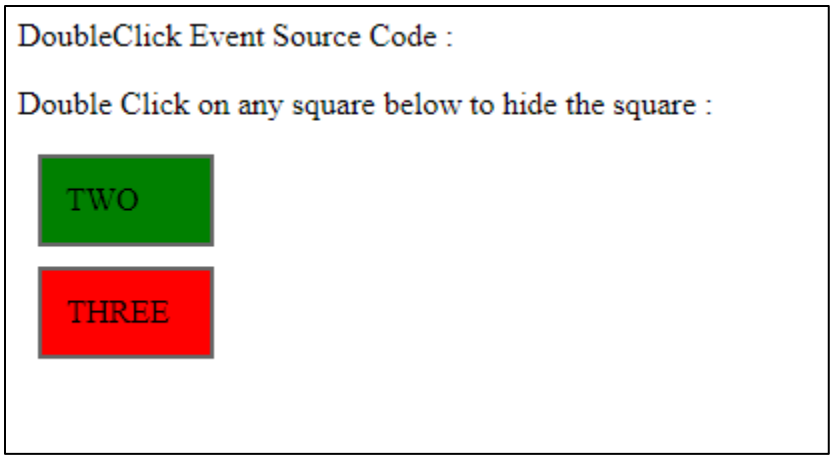
mongopractical > 8)1.html > html > head > script

```
DoubleClick Event Source Code :
<html>
<head>
    <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.5.0/jquery.min.js"></script
    <script type="text/javascript">
$(document).ready(function() {
$('div').dblclick(function(){
$(this).hide();
});
});
</script>
<style>
.div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}
</style>
</head>
<body>
<p>Double Click on any square below to hide the square :</p>
<div class = "div" style = "background-color:blue;">ONE</div>
<div class = "div" style = "background-color:green;">TWO</div>
<div class = "div" style = "background-color:red;">THREE</div>
</body>
</html>
```

NAME - KHAN ARMAN
ROLL NO - 13

OUTPUT :





NAME - KHAN ARMAN
ROLL NO - 13

4) Mouseleave

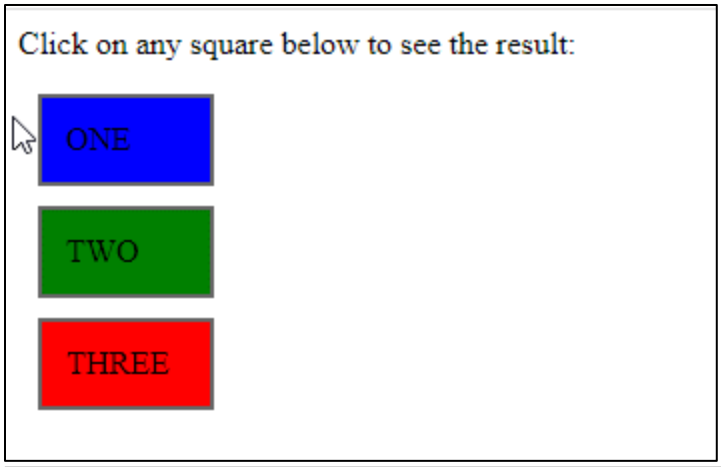
Source Code :

mongopractical > 8)1.html > html

```
<html>
<head>
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.5.0/jquery.min.js"></script>
    <script type="text/javascript">
$(document).ready(function() {
$('.div').mouseleave(function(){
alert('Event type is ' + event.type); alert('Target : ' + event.target.innerHTML);
});
});
</script>
<style>
.div{ margin:10px;padding:12px; border:2px solid #666; width:60px;}
</style>
</head>
<body>
<p>Click on any square below to see the result:</p>
<div class = "div" style = "background-color: blue;">ONE</div>
<div class = "div" style = "background-color: green;">TWO</div>
<div class = "div" style = "background-color: red;">THREE</div>
</body>
</html>
```

OUTPUT :

NAME - KHAN ARMAN
ROLL NO - 13



b. JQuery Selectors, JQuery Hide and Show Effects

JQuery Selectors -

Name Selector

CODE :

NAME - KHAN ARMAN
ROLL NO - 13

mongopractical > 8)1.html > ...

```
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.5.0/jquery.min.js"></script>
<html>
<head>
<title>The Selecter Example</title> <script type = "text/javascript"
src =
"https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
</script>
<script type = "text/javascript" language = "javascript">
$(document).ready(function() {
/* This would select all the divisions */
$("div").css("background-color", "yellow");
});
</script>
</head
<body>
<div class = "big" id = "div1">
<p>This is first division of the DOM.</p>
</div>
<div class = "medium" id = "div2">
<p>This is second division of the DOM.</p>
</div>
<div class = "small" id = "div3">
<p>This is third division of the DOM</p>
</div>
</body>
</html>
```

OUTPUT :

This is first division of the DOM.

This is second division of the DOM.

This is third division of the DOM

NAME - KHAN ARMAN
ROLL NO - 13

ID Selector

Source Code :

```
mongopractical > 8)1.html > ...
1
2   <script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/1.5.0/jquery.min.js"></script>
3   <html>
4   <head>
5   <title>The Selecter Example</title> <script type = "text/javascript"
6   src =
7   "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
8   </script>
9   <script type = "text/javascript" language = "javascript">
10  $(document).ready(function() {
11  /* This would select all the divisions */
12  $("div").css("background-color", "yellow");
13  });
14  </script>
15  </head
16  <body>
17  <div class = "big" id = "div1">
18  <p>This is first division of the DOM.</p>
19  </div>
20  <div class = "medium" id = "div2">
21  <p>This is second division of the DOM.</p>
22  </div>
23  <div class = "small" id = "div3">
24  <p>This is third division of the DOM</p>
25  </div>
26  </body>
27  </html>
28
```

OUTPUT :

This is first division of the DOM.

This is second division of the DOM.

This is third division of the DOM

NAME - KHAN ARMAN
ROLL NO - 13

Class Selector

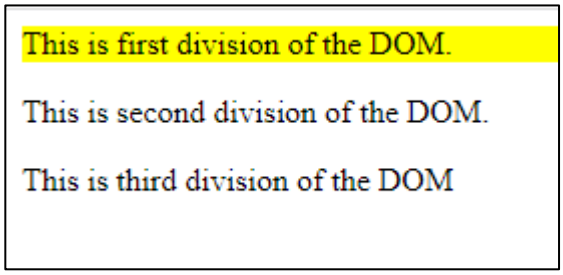
Source Code :

mongopractical > 8)1.html > ...

```
<html>
<head>
<title>The Selecter Example</title> <script type = "text/javascript"
src =
"https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
</script>
<script type = "text/javascript" language = "javascript">
$(document).ready(function() {
/* This would select second division only*/
$(".big").css("background-color", "yellow");
});
</script>
</head>
<body>
<div class = "big" id="div1">
<p>This is first division of the DOM.</p>
</div>
<div class = "medium" id = "div2">
<p>This is second division of the DOM.</p>
</div>
<div class = "small" id = "div3">
<p>This is third division of the DOM</p>
</div>
</body>
</html>
```

NAME - KHAN ARMAN
ROLL NO - 13

OUTPUT :



Universal

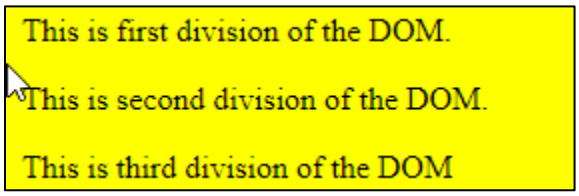
selector

Source Code :

```
mongopractical > <> 8)1.html > ...
1   <html>
2   <head>
3   <title>The Selecter Example</title> <script type = "text/javascript"
4   src =
5   "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
6   </script>
7   <script type = "text/javascript" language = "javascript">
8   $(document).ready(function() {
9   /* This would select all the elements */
10  $("*").css("background-color", "yellow");
11  });
12  </script>
13  </head>
14  <body>
15  <div class = "big" id = "div1">
16  <p>This is first division of the DOM.</p>
17  </div>
18  <div class = "medium" id = "div2">
19  <p>This is second division of the DOM.</p>
20  </div>
21  <div class = "small" id = "div3">
22  <p>This is third division of the DOM</p>
23  </div>
24  </body>
25  </html>
26
```

NAME - KHAN ARMAN
ROLL NO - 13

OUTPUT :



Multiple

Selector

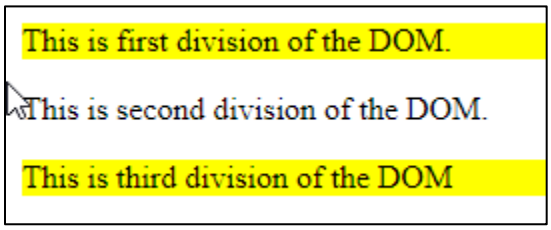
Source Code :

mongopractical > 8)1.html > ...

```
<html>
<head>
<title>The Selecter Example</title> <script type = "text/javascript"
src =
"https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
</script>
    <script type = "text/javascript" language = "javascript">
$(document).ready(function() {
$(".big, #div3").css("background-color", "yellow");
});
</script>
</head>
<body>
<div class = "big" id = "div1">
<p>This is first division of the DOM.</p>
</div>
<div class = "medium" id = "div2">
<p>This is second division of the DOM.</p>
</div>
<div class = "small" id = "div3">
<p>This is third division of the DOM</p>
</div>
</body>
</html>
```

NAME - KHAN ARMAN
ROLL NO - 13

OUTPUT :



Jquery hide and show effect source :

Code :

mongopractical > 8j1.html > html

```
<html>
<head>
<title>The jQuery Example</title> <script type = "text/javascript"
src =
"https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
</script>
<script type = "text/javascript" language = "javascript">
$(document).ready(function() {
$("#show").click(function () {
$(".mydiv").show( 1000 );
});
$("#hide").click(function () {
$(".mydiv").hide( 1000 );
});
});
</script>
<style> .mydiv{  margin:10px; padding:12px;  border:2px solid #666;  width:100px;  height:100px;
}
</style>
</head>
<body>
<div class = "mydiv">
This is a SQUARE
</div>
<input id = "hide" type = "button" value = "Hide" />
<input id = "show" type = "button" value = "Show" />
</body>
</html>
```

OUTPUT :

NAME - KHAN ARMAN
ROLL NO - 13

C. JQuery Fading Effect, JQuery Sliding Effect

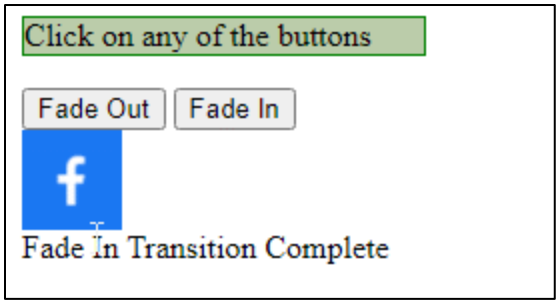
JQuery Fading Effect

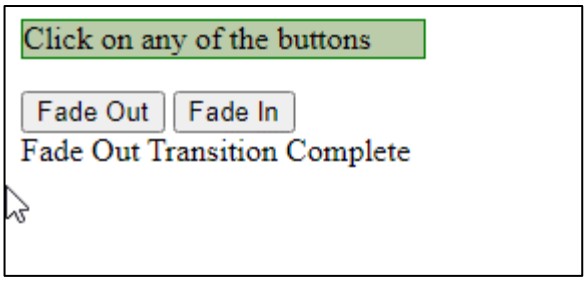
Source Code :

```
mongopractical > 8)1.html > html > body > div.target > img
 9  $("#in").click(function(){
10  $(".target").fadeIn( 'slow', function(){
11  $(".log").text('Fade In Transition Complete');
12  });
13  });
14  $("#out").click(function(){
15  $(".target").fadeOut( 'slow', function(){
16  $(".log").text('Fade Out Transition Complete');
17  });
18  });
19  });
20  </script> <style>
21  p {background-color: #bca; width:200px; border:1px solid green;} img{height:50px;width:50px}
22  </style>
23  </head>
24  <body>
25  <p>Click on any of the buttons</p>
26  <button id = "out"> Fade Out </button>
27  <button id = "in"> Fade In</button>
28  <div class = "target">
29  <img src = "fb.png" alt = "jQuery" />
30  </div>
31  <div class = "log"></div>
32  </body>
33  </html>
34
```

OUTPUT :

NAME - KHAN ARMAN
ROLL NO - 13





JQuery Sliding Effect

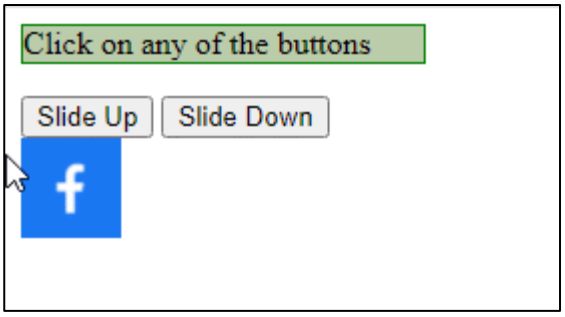
Source Code :

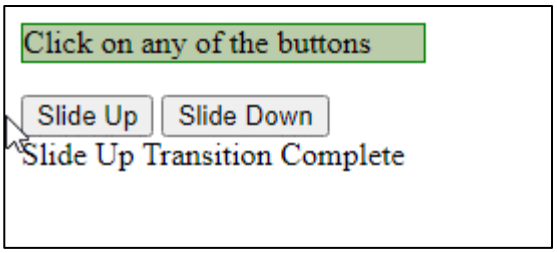
```
mongopractical > 8)1.html > html > body > div.target > img
 9  $(".target").slideDown( 'slow', function(){
10  $(".log").text('Slide Down Transition Complete');
11  });
12  });
13  $("#up").click(function(){
14  $(".target").slideUp( 'slow', function(){
15  $(".log").text('Slide Up Transition Complete');
16  });
17  });
18  });
19  </script>
20  <style>
21  p {background-color: #bca; width:200px; border:1px solid green;} img{heigth:50px;width:50px}
22  </style>
23  </head>
24  <body>
25  <p>Click on any of the buttons</p>
26  <button id = "up"> Slide Up </button>
27  <button id = "down"> Slide Down</button>
28  <div class = "target">
29  <img    src     =   "fb.png" alt = "jQuery" />
30  </div>
31  <div class = "log"></div>
32  </body>
33  </html>
34
```

OUTPUT :

NAME - KHAN ARMAN

ROLL NO - 13





PRACTICAL NO:09

JQUERY ADVANCED

a.jQuery Animation Effect,jQuery Chaining.

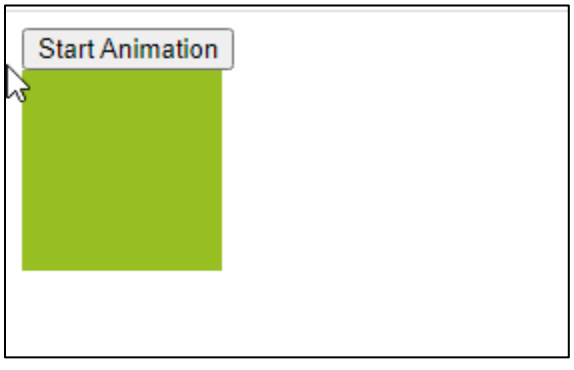
jQuery Animation

Source Code:

NAME - KHAN ARMAN
ROLL NO - 13

```
mongopractical > 8)1.html > html > body > div
1   <html>
2   <head>
3       <script type = "text/javascript"
4       src =
5       "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
6       </script> <script>
7   $(document).ready(function(){
8   $("button").click(function(){
9   $("div").animate({left: '250px'});
10  });
11  });
12  </script>
13  </head>
14  <body>
15  <button>Start Animation</button>
16  <div
17  style="background: #98bf21;height:100px;width:100px;position:absolute;"></div> </body>
18  </html>
19
```

OUTPUT :



NAME - KHAN ARMAN
ROLL NO - 13

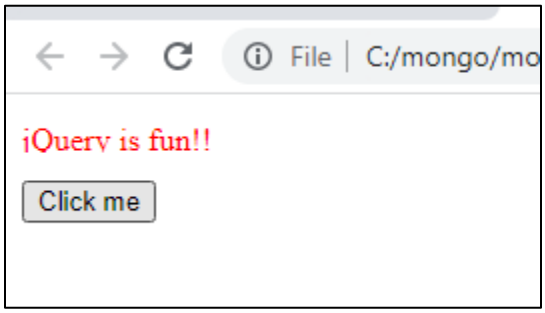
jQuery Chaining -

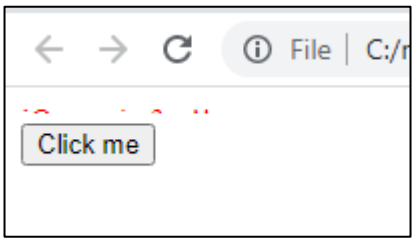
Source Code:

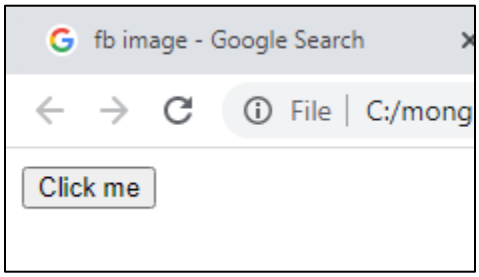
NAME - KHAN ARMAN
ROLL NO - 13

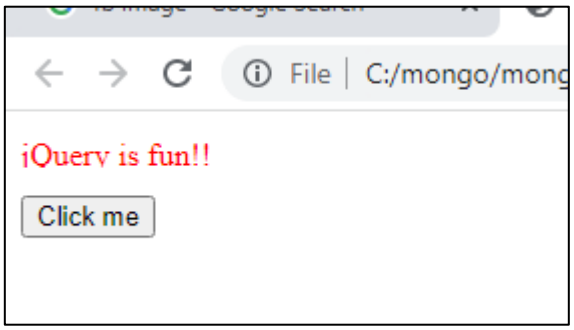
```
mongopractical > <> 8)1.html > ...
1   <html>
2   <head>
3   <script
4   src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.j s"></script>
5   <script>
6   $(document).ready(function(){
7   $("button").click(function(){
8   $("#p1").css("color", "red").slideUp(2000).slideDown(2000);
9   });
10  });
11  </script>
12  </head>
13  <body>
14  <p id="p1">jQuery is fun!!</p>
15  <button>Click me</button>
16  </body>
17  </html>
```

OUTPUT :









b.jQuery Callback,jQuery Get and Set Contents

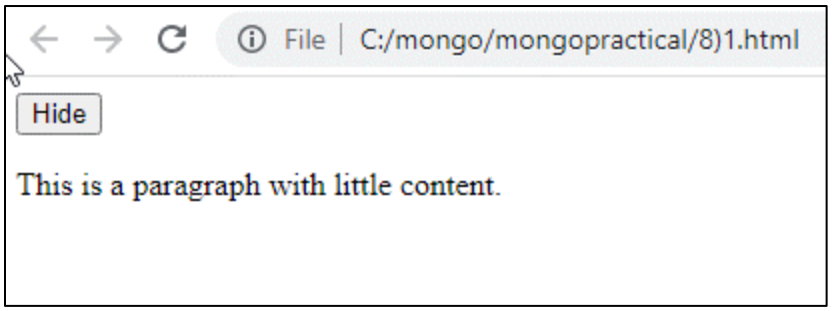
b)1) jQuery Callback

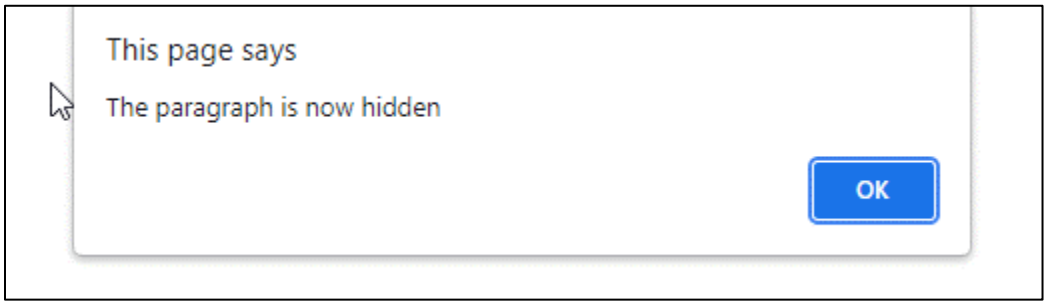
NAME - KHAN ARMAN
ROLL NO - 13

Source Code:

```
mongopractical > 8)1.html > html > head > script
1  <html>
2  <head>
3      <script type = "text/javascript"
4      src =
5      "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
6      </script>
7  <script>
8  $(document).ready(function(){
9  $("button").click(function(){ $("p").hide("slow", function(){ alert("The paragraph is now hidden");
10 });
11 });
12 });
13 </script>
14 </head>
15 <body>
16 <button>Hide</button>
17 <p>This is a paragraph with little content.</p>
18 </body>
19 </html>
20
```

OUTPUT :





jQuery Get and Set Contents

B) 2)GET Content

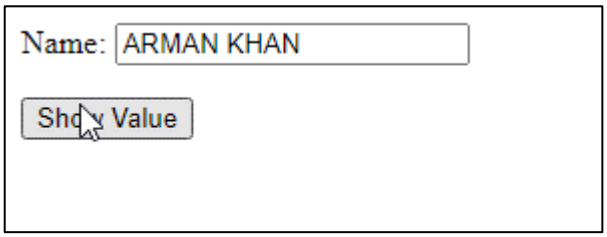
NAME - KHAN ARMAN

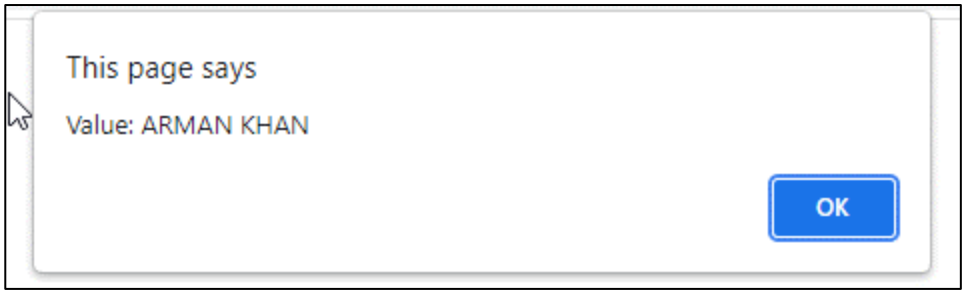
ROLL NO - 13

Source Code:

```
mongopractical > 8)1.html > html > head > script
1   <html>
2   <head>
3       <script type = "text/javascript"
4       src =
5       "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
6       </script>
7   <script>
8   $(document).ready(function(){   $("button").click(function(){     alert("Value: " + $("#test").val());
9     });
10  });
11  </script>
12  </head>
13  <body>
14  <p>Name: <input type="text" id="test" value="Mickey Mouse"></p>
15  <button>Show Value</button>
16  </body>
17  </html>
18
```

OUTPUT :





b)2) SET Content

Source Code :

NAME - KHAN ARMAN

ROLL NO - 13

```
mongopractical > 8)1.html > html > body > p > input#test3
    src =
    "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
    </script>
<script>
$(document).ready(function(){
$("#btn1").click(function(){
$("#test1").text("Hello world!");
});
$("#btn2").click(function(){
$("#test2").html("<b>Hello world!</b>");
});
$("#btn3").click(function(){
$("#test3").val("ARMAN");
});
});
</script>
</head>
<body>
<p id="test1">This is a paragraph.</p>
<p id="test2">This is another paragraph.</p>
<p>Input field: <input type="text" id="test3" value="IRFAN"></p>
<button id="btn1">Set Text</button>
<button id="btn2">Set HTML</button>
<button id="btn3">Set Value</button>
</body>
</html>
```

OUTPUT :

This is a paragraph.

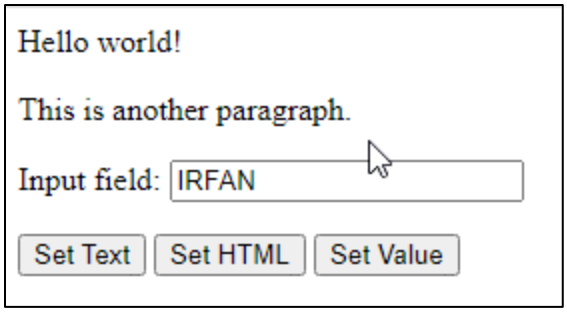
This is another paragraph.

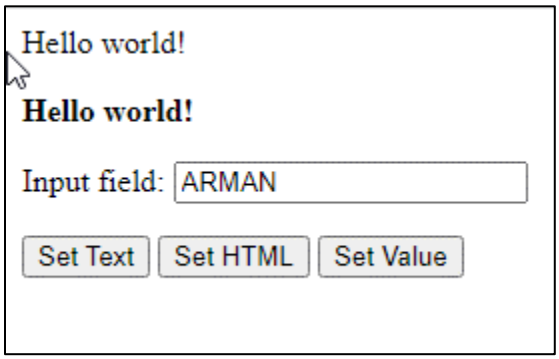
Input field: IRFAN

Set Text | Set HTML | Set Value

NAME - KHAN ARMAN

ROLL NO - 13





NAME - KHAN ARMAN

ROLL NO - 13

C. JQuery Insert Content, Remove Content & Attribute

C)1) JQuery Insert Content

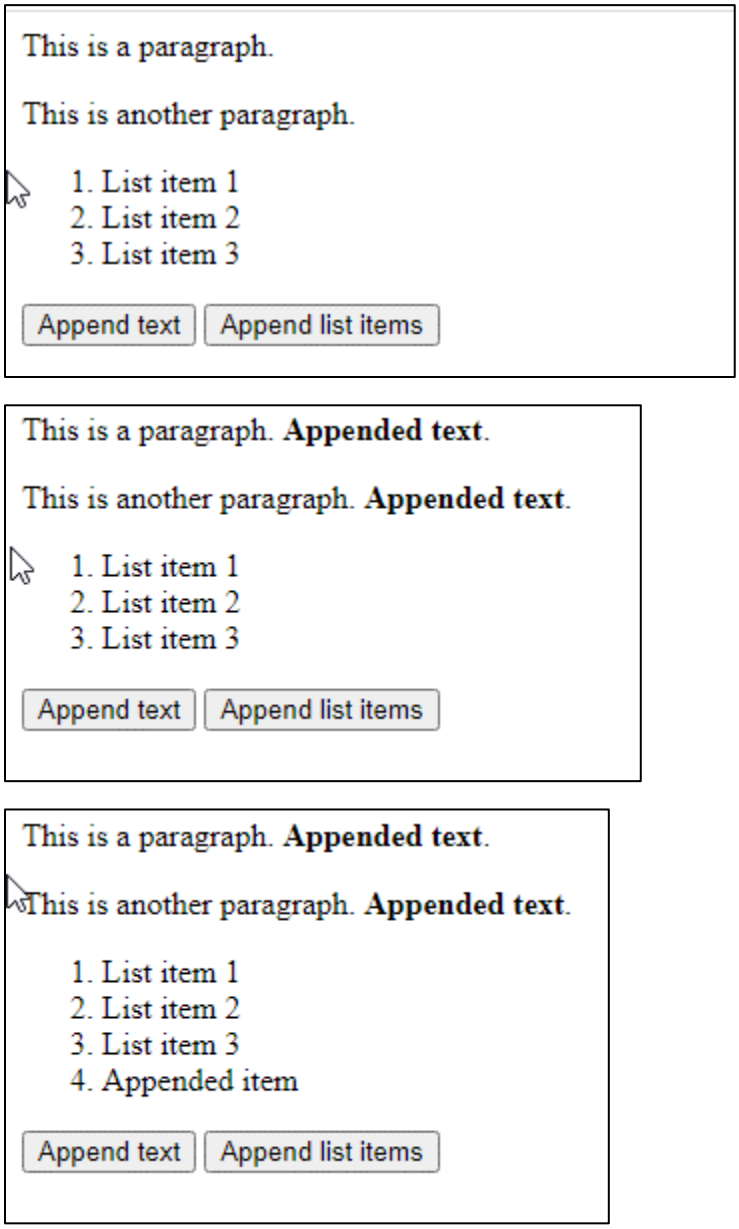
Source Code :

mongopractical > 8)1.html > html > head > script

```
<html>
<head>
    <script type = "text/javascript"
    src =
    "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
    </script>
<script>
$(document).ready(function(){
$("#btn1").click(function(){
$("p").append(" <b>Appended text</b>.");
});
$("#btn2").click(function(){
$("ol").append("<li>Appended item</li>");
});
});
</script>
</head>
<body>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>
<button id="btn1">Append text</button>
<button id="btn2">Append list items</button>
</body>
</html>
```

NAME - KHAN ARMAN
ROLL NO - 13

OUTPUT :



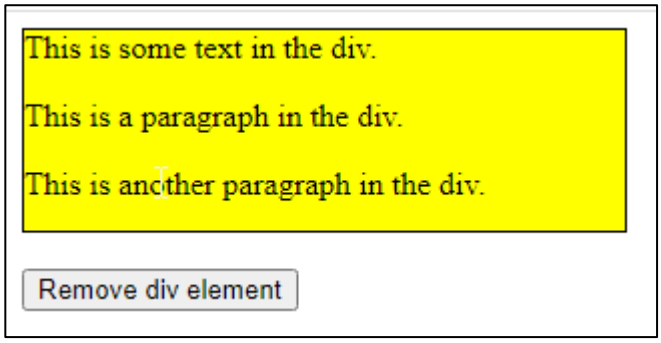
NAME - KHAN ARMAN
ROLL NO - 13

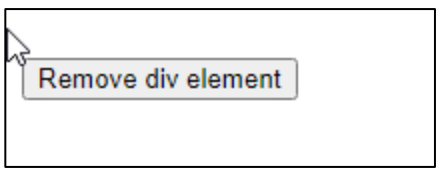
C)2)JQuery Remove Content and attribute

Source Code :

```
mongopractical > 8)1.html > html > head > script
<html>
<head>
    <script type = "text/javascript"
    src =
    "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
    </script>
<script>
$(document).ready(function(){
$("button").click(function(){
$("#div1").remove();
});
});
</script>
</head>
<body>
<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;"> This is some text in the div.
<p>This is a paragraph in the div.</p>
<p>This is another paragraph in the div.</p>
</div>
<br>
<button>Remove div element</button>
</body>
</html>
```

OUTPUT :





NAME - KHAN ARMAN
ROLL NO - 13

PRACTICAL NO : 10 JSON

A. Creating JSON

Source Code :

8)1.html × g.html

mongopractical > 8)1.html > ...

```
1   <html>
2   <body>
3   <p>Access a JSON object using dot notation:</p>
4   <p id="demo"></p> <script> var myObj, x;
5   myObj = {"name":"John", "age":30, "car":null}; x = myObj.name;
6   document.getElementById("demo").innerHTML = x;
7   </script>
8   </body>
9   </html>
10
```

OUTPUT :

Access a JSON object using dot notation:

John

NAME - KHAN ARMAN
ROLL NO - 13

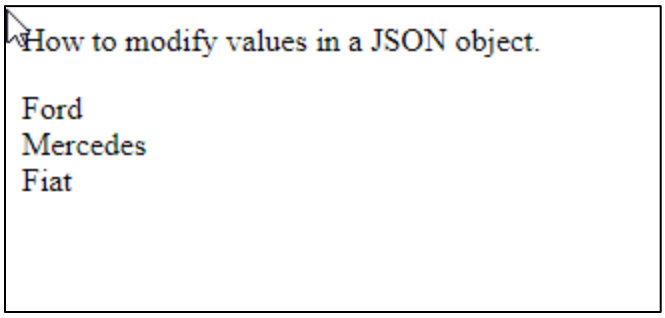
B) Modify JSON

Source Code :

mongopractical > 8)1.html > ...

```
<html>
<body>
<p>How to modify values in a JSON object.</p>
<p id="demo"></p> <script> var myObj, i, x = ""; myObj = {
"name":"John",
"age":30,
"cars": {
"car1":"Ford",
"car2":"BMW",
"car3":"Fiat"
}
}
myObj.cars.car2 = "Mercedes"; for (i in myObj.cars) { x += myObj.cars[i] + "<br>";
}
document.getElementById("demo").innerHTML = x;
</script>
</body>
</html>
```

OUTPUT :



NAME - KHAN ARMAN
ROLL NO - 13

C) Parsing JSON

Source Code :

mongopractical > 8)1.html > html > body > script

```
<html>
<body>
<h2>Create Object from JSON String</h2>
<p id="demo"></p>
<script>
var txt = '{"name":"John", "age":30, "city":"New York"}'

var obj = JSON.parse(txt);
document.getElementById("demo").innerHTML = obj.name + "," + obj.age+","+obj.city;
</script>
</body>
</html>

```

OUTPUT :

Create Object from JSON String

John,30,New York

NAME - KHAN ARMAN
ROLL NO - 13