Slip 1

NEXT GENERATION TECHNOLOGIES

Seat No. : ____ **Max. Marks:50**

1.	(A) Import Restaurant.json into MongoDb and perform the following queries 1. Write a MongoDB query to display all the documents in the collection restaurants. 2. Write a MongoDB query to display the fields , restaurant_id, name, borough and cuisine for all the documents in the collection restaurant. 3. Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine, but exclude the field _id for all the documents in the collection restaurant 4. Write a MongoDB query to display the fields restaurant_id, name, borough and zip code, but exclude the field _id for all the documents in the collection restaurant 5. Write a MongoDB query to display all the restaurant which is in the borough Bronx (B) Connect Python with MongoDB and also insert and retrieve documents.	**20**
2.	(A) Write a jQuery to animate multiple CSS properties. (B) Write a jQuery effect method with a callback function.	**20**
3.	Viva	**5**
4.	Journal	**5**

Solution

1.A)

1.

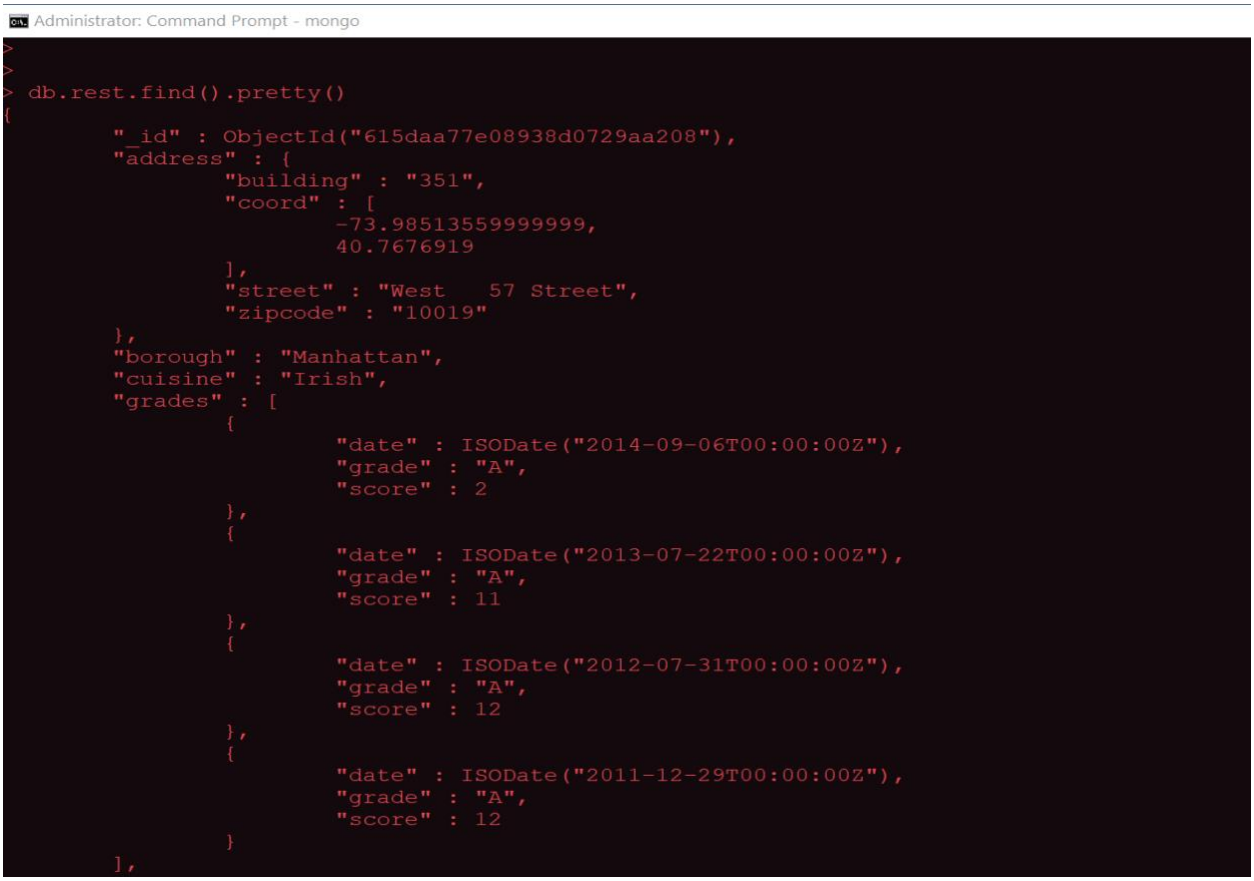
Code:

```
Administrator: Command Prompt - mongo
2021-10-06T18:38:31.208+0530 I CONTROL  [initandlist
> show dbs
Studtalks  0.000GB
Team       0.000GB
TeamStuds  0.000GB
admin      0.000GB
config     0.000GB
employee   0.000GB
local      0.000GB
mydb       0.000GB
> use resturants
switched to db resturants
> db.createCollection("rest")
{ "ok" : 1 }
> db.rest.find()
> db.rest.find()
```

Again open the command prompt to import the restaurant.json file

db.rest.find().pretty()

Output:

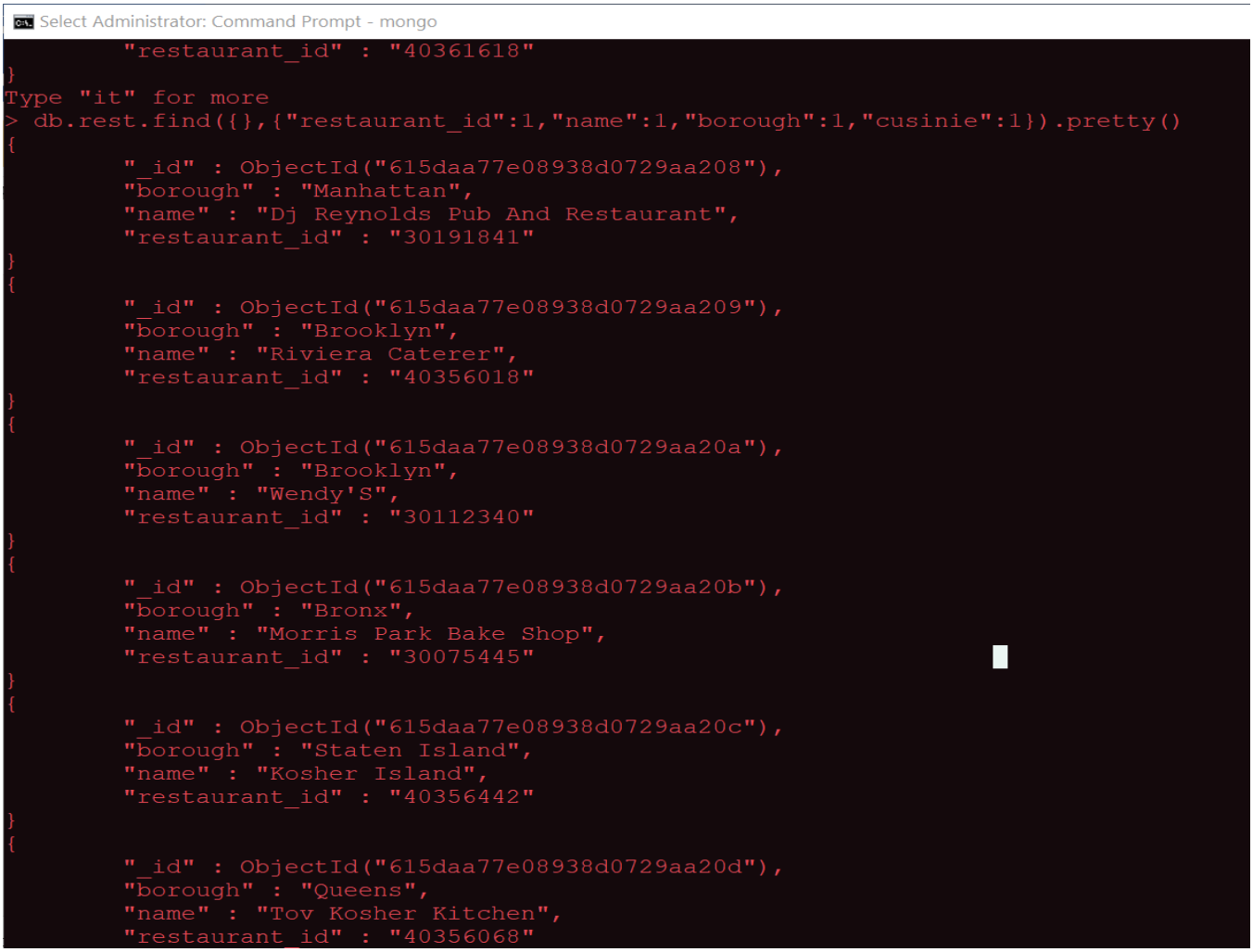


2:

Code:

db.rest.find({},{"restaurant_id":1,"name":1,"borough":1,"cusinie":1}).pretty()

Output:



3.

Code:

db.rest.find({},{"restaurant_id":1,"name":1,"borough":1,"cusine":1,"_id":0}).pretty()

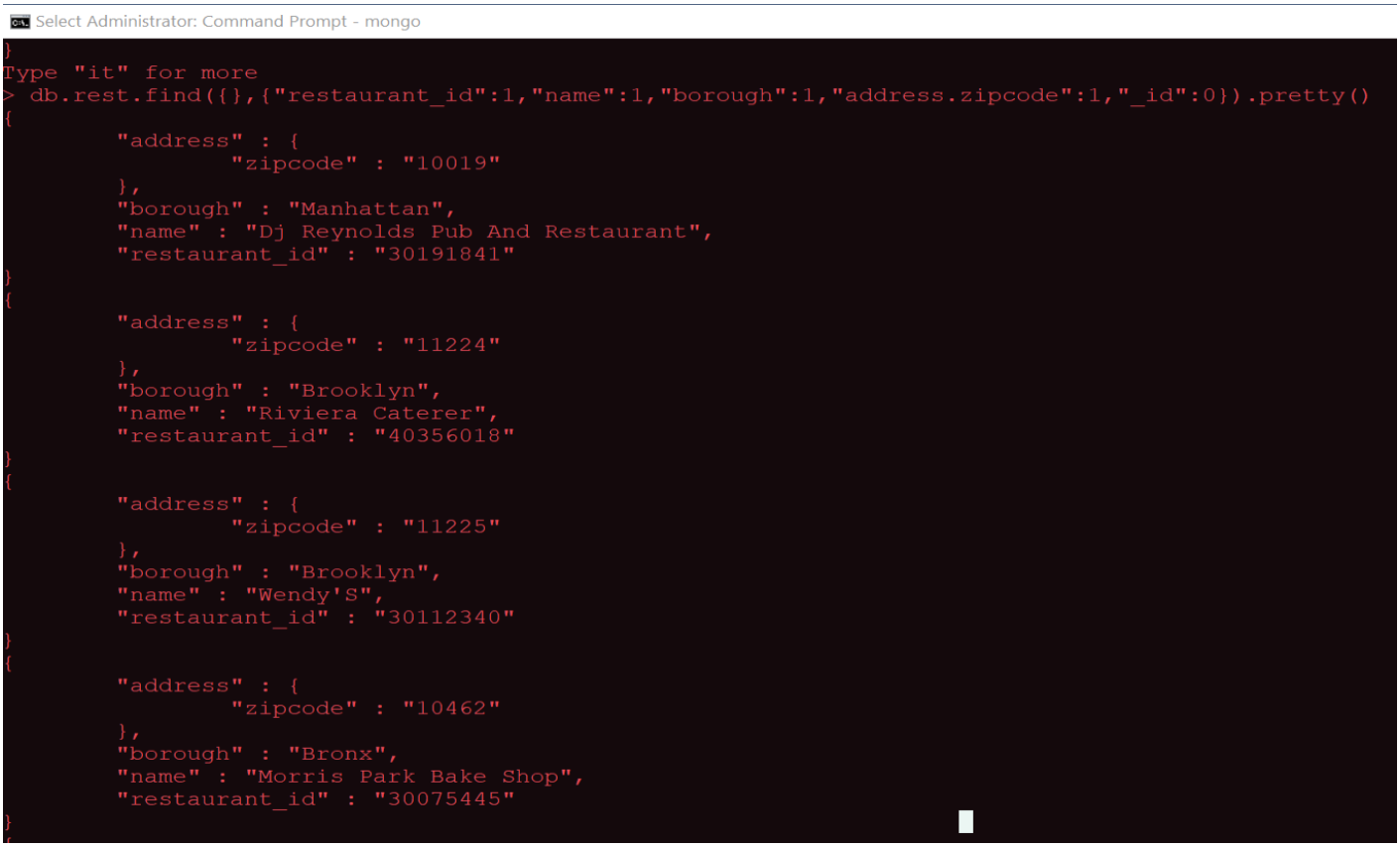
Output:

4.

Code:

db.rest.find({},{"restaurant_id":1,"name":1,"borough":1,"address.zipcode":1,"_id":0}).pretty()

Output:



5.

Code:

db.rest.find({"borough":"Bronx"}).pretty()

Output:


Administrator: Command Prompt - mongo
> db.rest.find({"borough":"Bronx"}).pretty()
{
"_id" : ObjectId("615daa77e08938d0729aa20b"),
"address" : {
"building" : "1007",
"coord" : [
-73.856077,
40.848447
],
"street" : "Morris Park Ave",
"zipcode" : "10462"
},
"borough" : "Bronx",
"cuisine" : "Bakery",
"grades" : [
{
"date" : ISODate("2014-03-03T00:00:00Z"),
"grade" : "A",
"score" : 2
},
{
"date" : ISODate("2013-09-11T00:00:00Z"),
"grade" : "A",
"score" : 6
},
{
"date" : ISODate("2013-01-24T00:00:00Z"),
"grade" : "A",
"score" : 10
},
{
"date" : ISODate("2011-11-23T00:00:00Z"),
"grade" : "A",
"score" : 9
},

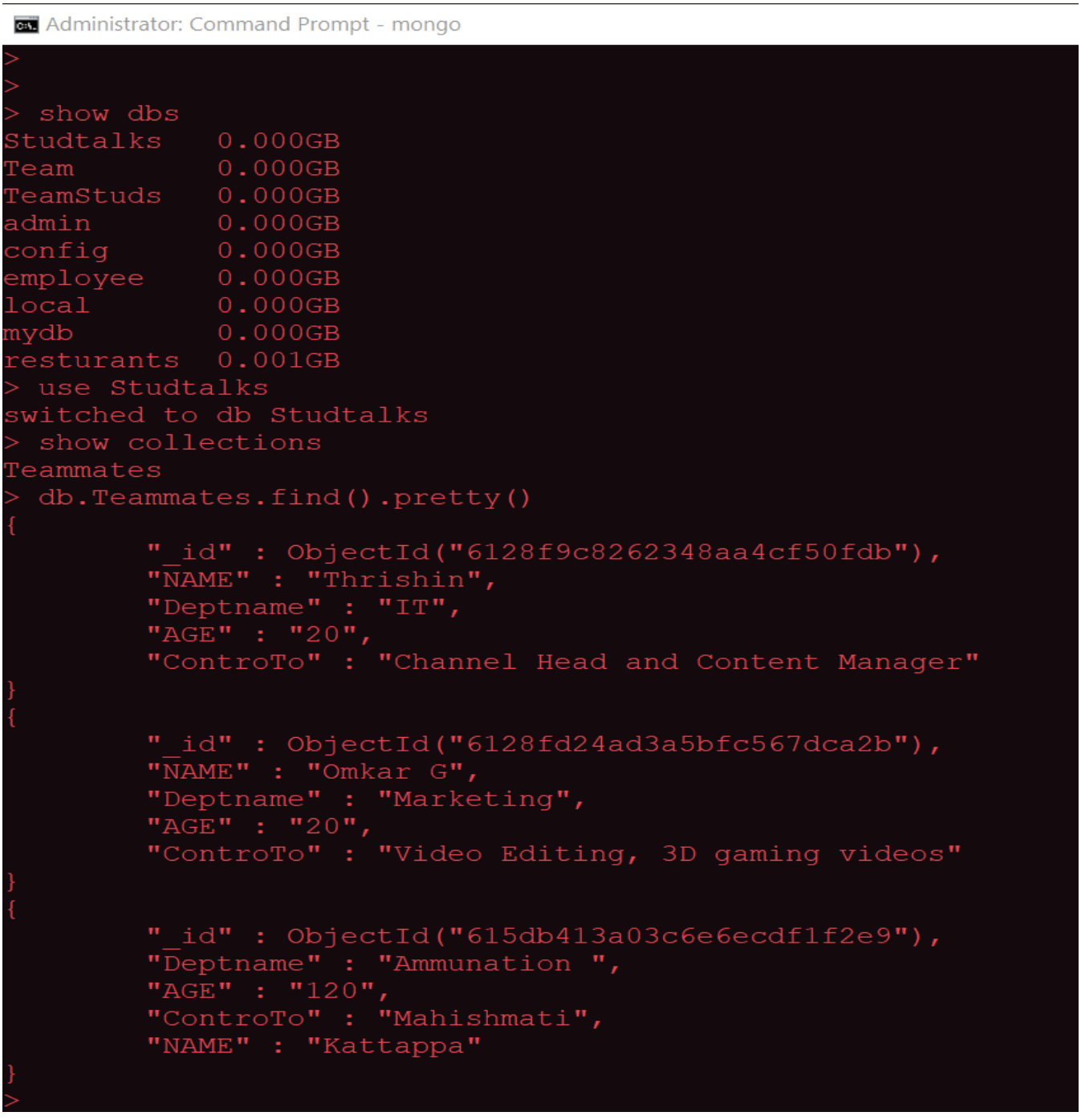
B)

Insert

Code:

```
from pymongo import MongoClient
client=MongoClient('localhost',27017)
db=client.Studtalks
def insert():
    try:
        Name=input("Enter Name : ")
        Age=input("Enter  Age : ")
        Dept=input("Department : ")
        Contro=input("Contributed for : ")
        db.Teammates.insert(
                {
                        "NAME":Name,
                        "AGE":Age,
                        "Deptname":Dept,
                        "ControTo":Contro
                }
            )
        print("Inserted data successfully ")
    except Exception:
        print(str(e))
insert()
```

Output:

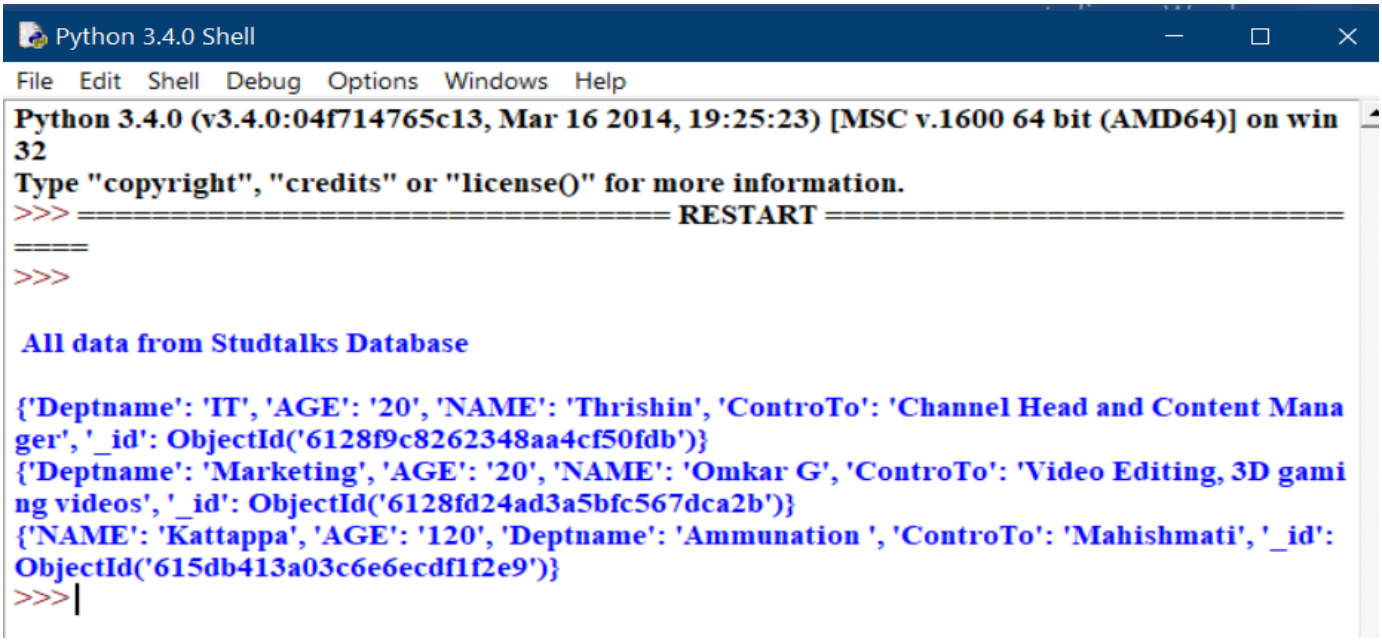

Administrator: Command Prompt - mongo
> show dbs
Studtalks 0.000GB
Team 0.000GB
TeamStuds 0.000GB
admin 0.000GB
config 0.000GB
employee 0.000GB
local 0.000GB
mydb 0.000GB
resturants 0.001GB
> use Studtalks
switched to db Studtalks
> show collections
Teammates
> db.Teammates.find().pretty()
{
"_id" : ObjectId("6128f9c8262348aa4cf50fdb"),
"NAME" : "Thrishin",
"Deptname" : "IT",
"AGE" : "20",
"ControTo" : "Channel Head and Content Manager"
}
{
"_id" : ObjectId("6128fd24ad3a5bfc567dca2b"),
"NAME" : "Omkar G",
"Deptname" : "Marketing",
"AGE" : "20",
"ControTo" : "Video Editing, 3D gaming videos"
}
{
"_id" : ObjectId("615db413a03c6e6ecdf1f2e9"),
"Deptname" : "Ammunation ",
"AGE" : "120",
"ControTo" : "Mahishmati",
"NAME" : "Kattappa"
}

Reterive

Code:

```
from pymongo import MongoClient
client=MongoClient('localhost',27017)
db=client.Studtalks
def read():
    try:
        Col = db.Teammates.find()
        print("\n All data from Studtalks Database \n")
        for Teammates in Col:
            print(Teammates)
    except Exception:
        print(str(e))
read()
```

Output:



2 A

Code:

```
<!DOCTYPE html>
<html>

<head>
    <title></title>
    <script src="jquery-3.6.0.min.js"></script>
    <style type="text/css">
        .box {
            width: 100px;
            height: 100px;
            background: #9d7ede;
            margin-top: 30px;
            border-style: solid;
            border-color: #6f40ce;
        }
    </style>
    <script>
```

```
        $(document).ready(
            function () {
                $("button").click(
                    function () {
                        $(".box").animate(
                            {
                                width: "300px",
                                height: "300px",
                                marginLeft: "150px",
                                borderWidth: "10px",
                                opacity: 0.5
                            }
                        );
                    }
                );
            }
        );
    </script>
</head>

<body>
    <button type="button">Start Animation</button>
    <div class="box"></div>
</body>

</html>
```

Output:

Start Animation

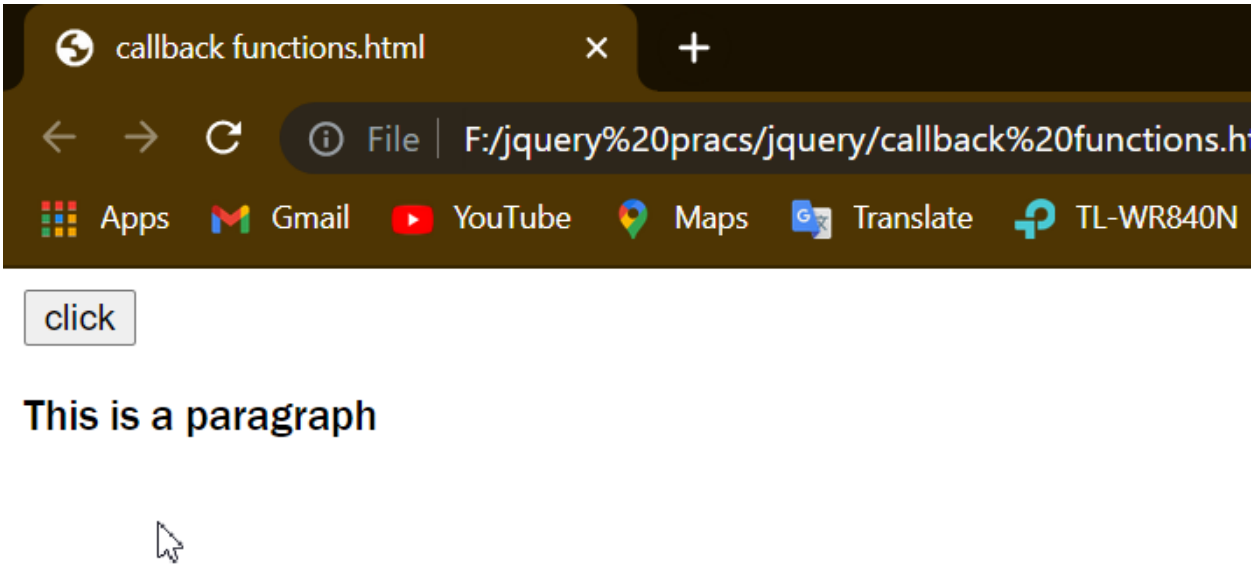
B.

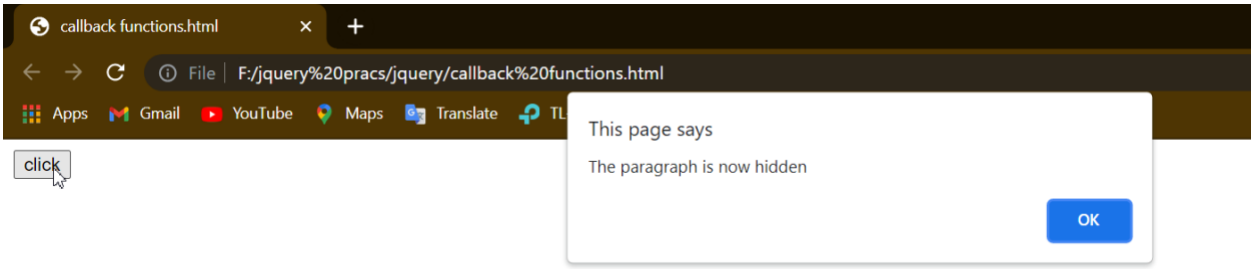
Code:

```
<!DOCTYPE html>
<html>

<head>
    <title></title>
    <script src="jquery-3.6.0.min.js"></script>
    <script>
        $(document).ready(
            function () {
                $("button").click(
                    function () {
```

```
                $("p").hide("slow", function () {
                    alert("The paragraph is now hidden");
                }
                );
            }
        );
    }
);
</script>
</head>

<body>
<button>click</button>
<p>This is a paragraph </p>
</body>

</html>
```

Output:





Slip 2

UNIVERSITY OF MUMBAI

T.Y.B.Sc.(INFORMATION TECHNOLOGY) (Semester– V) (Practical) EXAMINATION

SECOND HALF 2018

NEXT GENERATION TECHNOLOGIES

Seat No. :____ **Max. Marks:50**

1.	Connect Java with MongoDB and also insert, retrieve, update and delete documents.	**20**
2.	Create a JSON file and persist it in any database.	**20**
3.	Viva	**5**
4.	Journal	**5**

1.

Inserting data into database

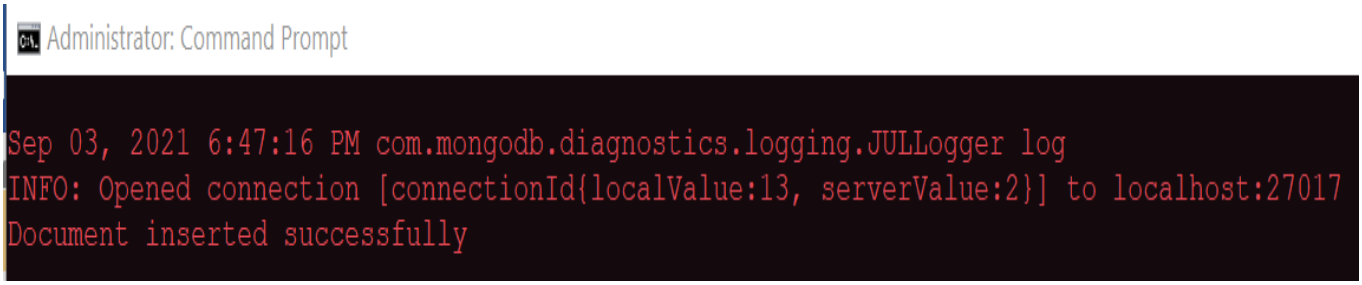
InsertingDocument - Notepad

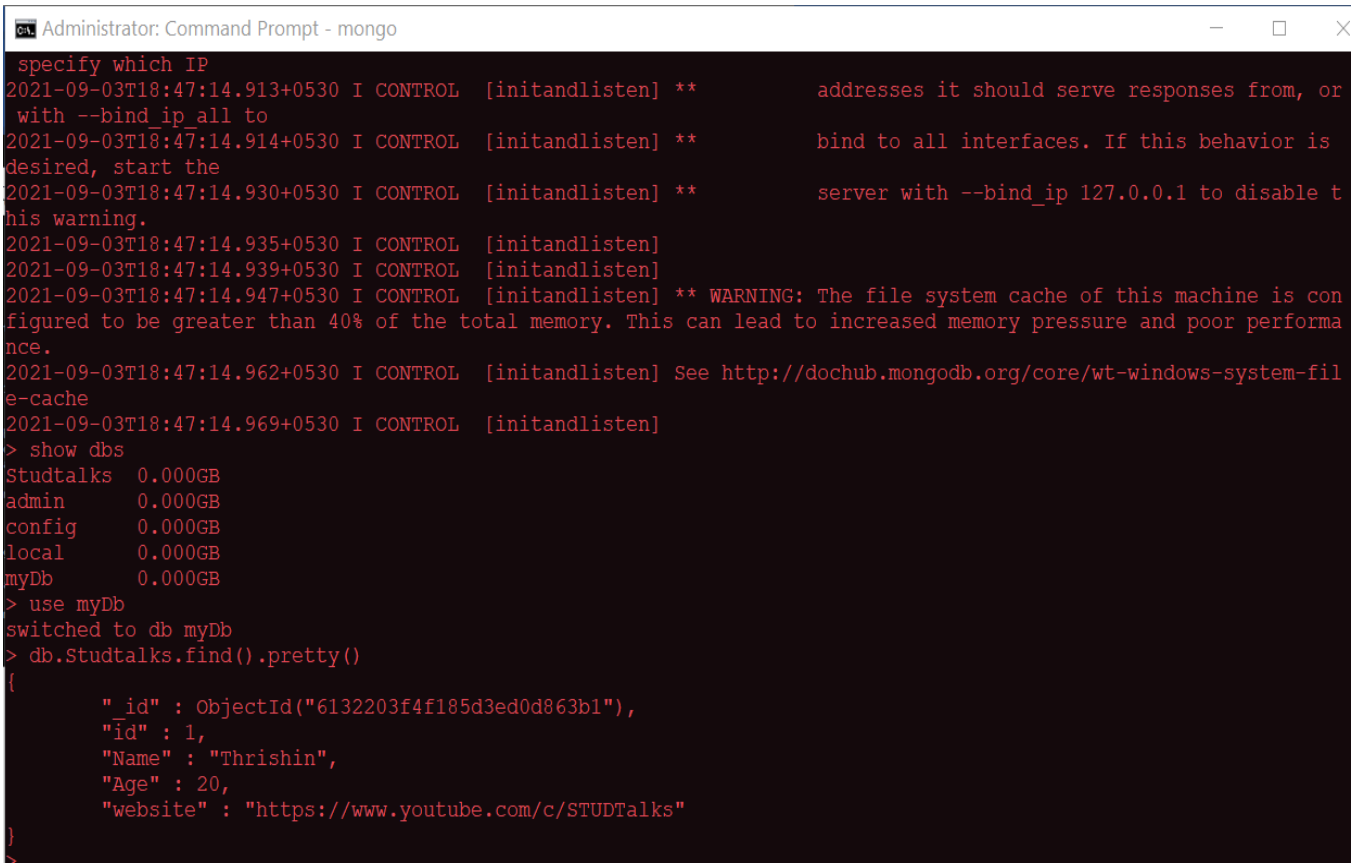
File Edit Format View Help

```
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import com.mongodb.MongoClient;
import com.mongodb.MongoCredential;
import org.bson.Document;

public class InsertingDocument
{
  public static void main( String args[] )
  {
    MongoClient mongo = new MongoClient( "localhost" , 27017 );
    System.out.println("Connected to the database successfully");
   MongoDatabase database = mongo.getDatabase("myDb");
    MongoCollection<Document> collection = database.getCollection("Studtalks");
    System.out.println("Collection Studtalks selected successfully");
    Document document = new Document();
    document.append("id", 1);
    document.append("Name", "Thrishin");
    document.append("Age", 20) ;
    document.append("website", "https://www.youtube.com/c/STUDTalks");

    collection.insertOne(document);
    System.out.println("Document inserted successfully");
  }
}
```

Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Javawithmongo

C:\Javawithmongo>javac InsertingDocument.java

C:\Javawithmongo>java InsertingDocument
Sep 03, 2021 6:46:47 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout
='30000 ms', maxWaitQueueSize=500}
Connected to the database successfully
Collection Studtalks selected successfully
Sep 03, 2021 6:46:47 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: No server chosen by PrimaryServerSelector from cluster description ClusterDescription{type=UNKNOWN, connectionMode=SINGL
E, all=[ServerDescription{address=localhost:27017, type=UNKNOWN, state=CONNECTING}]}. Waiting for 30000 ms before timing out
Sep 03, 2021 6:46:50 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Exception in monitor thread while connecting to server localhost:27017
com.mongodb.MongoSocketOpenException: Exception opening socket
        at com.mongodb.connection.SocketStream.open(SocketStream.java:63)
        at com.mongodb.connection.InternalStreamConnection.open(InternalStreamConnection.java:114)
        at com.mongodb.connection.DefaultServerMonitor$ServerMonitorRunnable.run(DefaultServerMonitor.java:127)
        at java.lang.Thread.run(Thread.java:745)
Caused by: java.net.ConnectException: Connection refused: connect
        at java.net.DualStackPlainSocketImpl.waitForConnect(Native Method)
        at java.net.DualStackPlainSocketImpl.socketConnect(DualStackPlainSocketImpl.java:85)
        at java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:345)
        at java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.java:206)
        at java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
        at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:172)
        at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
        at java.net.Socket.connect(Socket.java:589)
        at com.mongodb.connection.SocketStreamHelper.initialize(SocketStreamHelper.java:50)
        at com.mongodb.connection.SocketStream.open(SocketStream.java:58)
        ... 3 more
```


Administrator: Command Prompt
Sep 03, 2021 6:47:16 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:13, serverValue:2}] to localhost:27017
Document inserted successfully

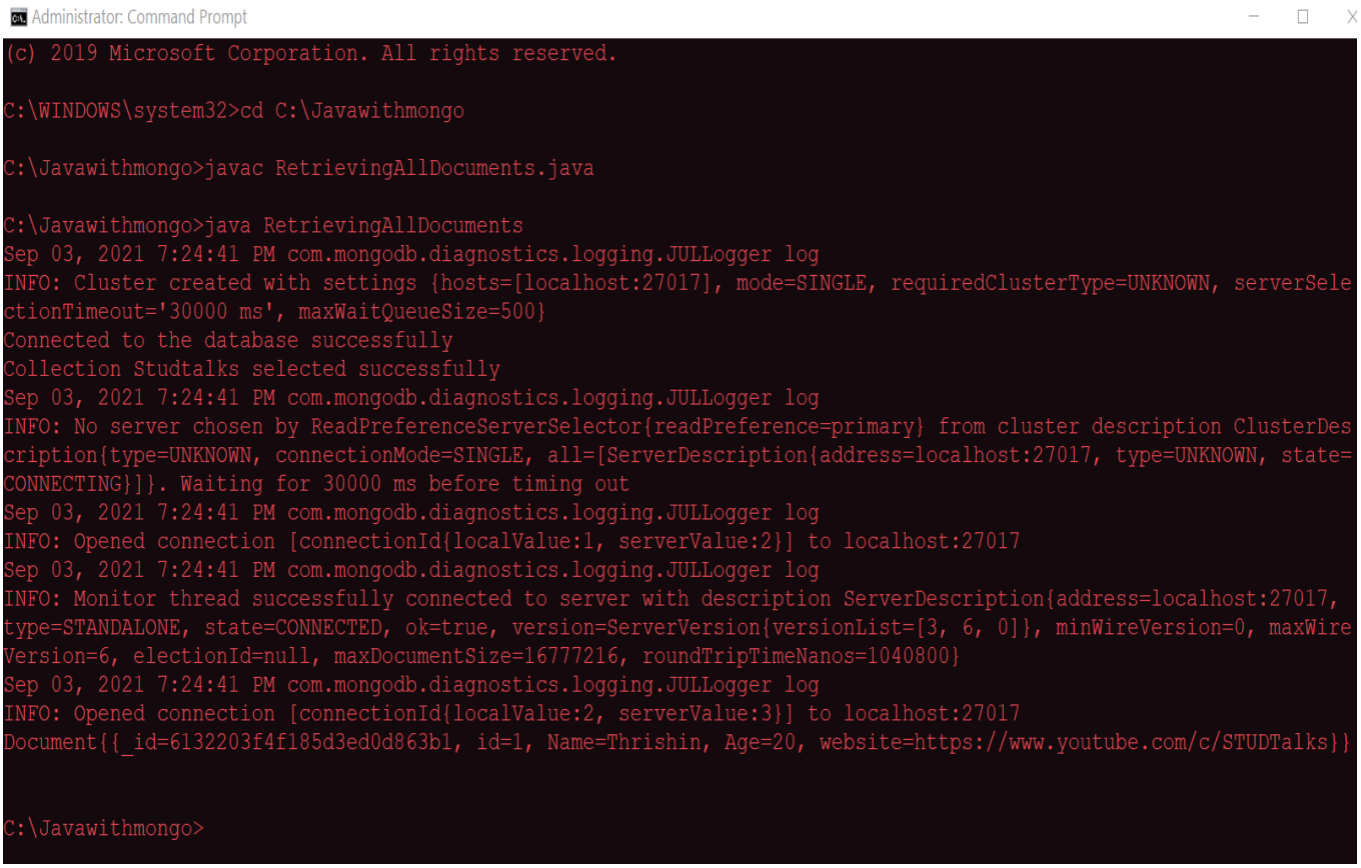

Administrator: Command Prompt - mongo
specify which IP
2021-09-03T18:47:14.913+0530 I CONTROL [initandlisten] ** addresses it should serve responses from, or
with --bind_ip_all to
2021-09-03T18:47:14.914+0530 I CONTROL [initandlisten] ** bind to all interfaces. If this behavior is
desired, start the
2021-09-03T18:47:14.930+0530 I CONTROL [initandlisten] ** server with --bind_ip 127.0.0.1 to disable t
his warning.
2021-09-03T18:47:14.935+0530 I CONTROL [initandlisten]
2021-09-03T18:47:14.939+0530 I CONTROL [initandlisten]
2021-09-03T18:47:14.947+0530 I CONTROL [initandlisten] ** WARNING: The file system cache of this machine is con
figured to be greater than 40% of the total memory. This can lead to increased memory pressure and poor performa
nce.
2021-09-03T18:47:14.962+0530 I CONTROL [initandlisten] See http://dochub.mongodb.org/core/wt-windows-system-fil
e-cache
2021-09-03T18:47:14.969+0530 I CONTROL [initandlisten]
> show dbs
Studtalks 0.000GB
admin 0.000GB
config 0.000GB
local 0.000GB
myDb 0.000GB
> use myDb
switched to db myDb
> db.Studtalks.find().pretty()
{
"_id" : ObjectId("6132203f4f185d3ed0d863b1"),
"id" : 1,
"Name" : "Thrishin",
"Age" : 20,
"website" : "https://www.youtube.com/c/STUDTalks"
}

Retrieving data from database

RetrievingAllDocuments - Notepad

File Edit Format View Help

```
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import com.mongodb.MongoClient;
import com.mongodb.MongoCredential;
import org.bson.Document;
import java.util.Iterator;
import com.mongodb.client.FindIterable;
public class RetrievingAllDocuments
{
  public static void main( String args[] )
  {
    MongoClient mongo = new MongoClient( "localhost" , 27017 );
    System.out.println("Connected to the database successfully");
    MongoDatabase database = mongo.getDatabase("myDb");
    MongoCollection<Document> collection = database.getCollection("Studtalks");
    System.out.println("Collection Studtalks selected successfully");
    FindIterable<Document> iterDoc = collection.find();
    int i = 1;
    Iterator it = iterDoc.iterator();
    while(it.hasNext())
              {
                                        System.out.println(it.next());
                                        i++;
    }
  }
}
```

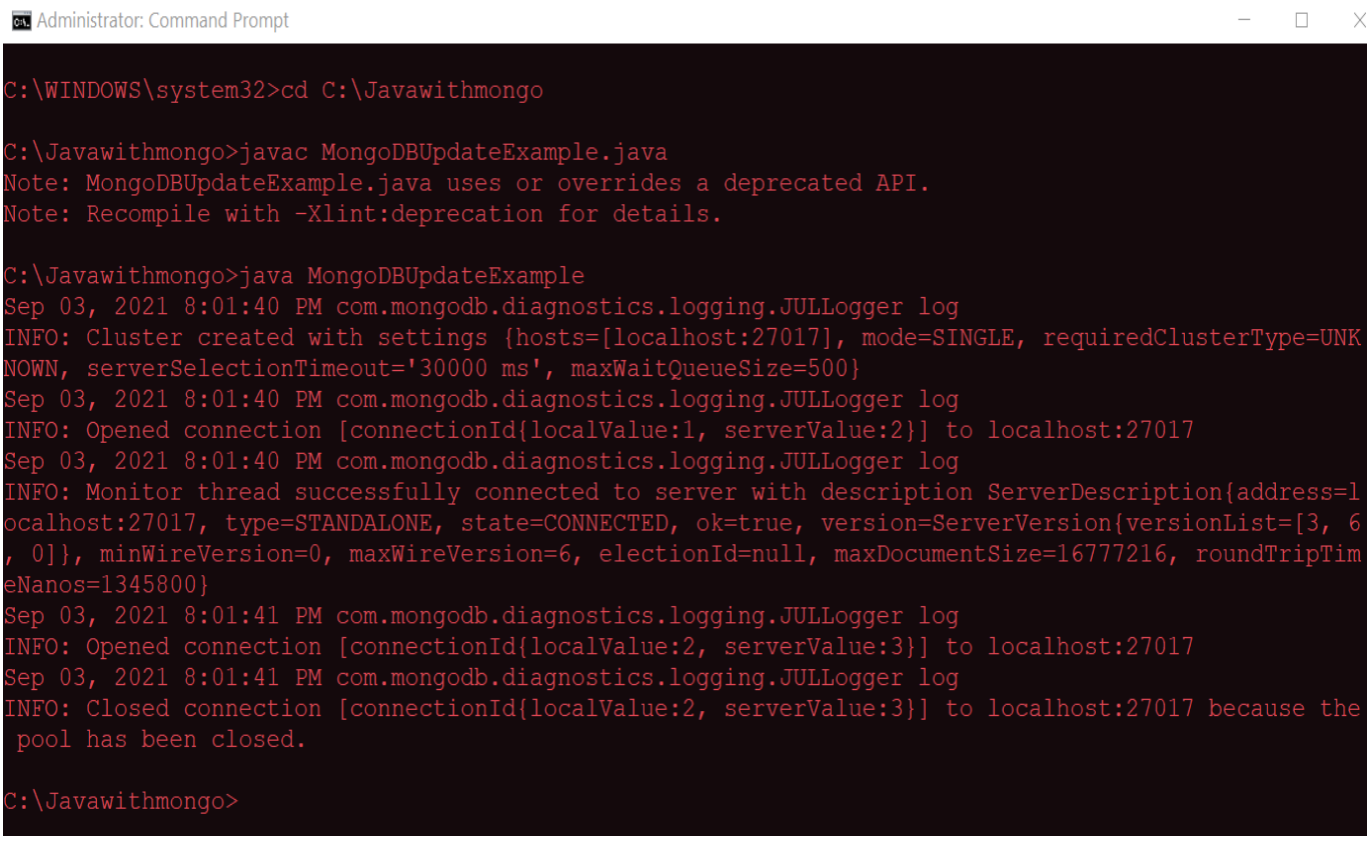


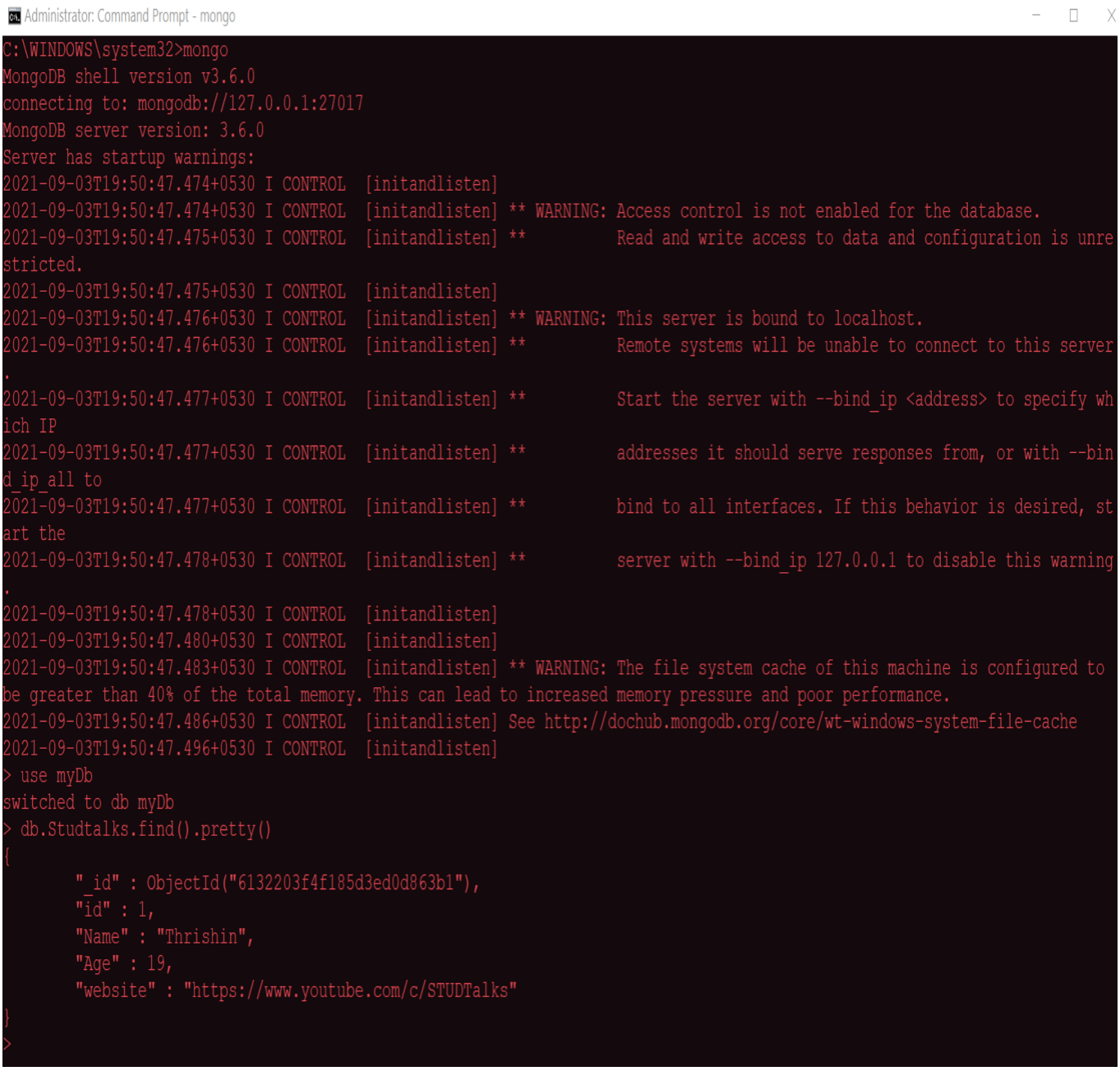
Updating data in database

MongoDBUpdateExample - Notepad

File Edit Format View Help

```
import java.net.UnknownHostException;
import com.mongodb.BasicDBObject;
import com.mongodb.DB;
import com.mongodb.DBCollection;
import com.mongodb.DBObject;
import com.mongodb.MongoClient;
import com.mongodb.WriteResult;

public class MongoDBUpdateExample
{
        public static void main(String[] args) throws UnknownHostException
        {
                MongoClient mongo = new MongoClient("localhost", 27017);
                DB db = mongo.getDB("myDb");
                DBCollection col = db.getCollection("Studtalks");
                DBObject query = new BasicDBObject("id", 1);
                DBObject update = new BasicDBObject();
                update.put("$set", new BasicDBObject("Age",19));
                WriteResult result = col.update(query, update);
                mongo.close();

        }

}
```


Administrator: Command Prompt
C:\WINDOWS\system32>cd C:\Javawithmongo
C:\Javawithmongo>javac MongoDBUpdateExample.java
Note: MongoDBUpdateExample.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
C:\Javawithmongo>java MongoDBUpdateExample
Sep 03, 2021 8:01:40 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNK
NOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Sep 03, 2021 8:01:40 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:2}] to localhost:27017
Sep 03, 2021 8:01:40 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=l
ocalhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[3, 6
, 0]}, minWireVersion=0, maxWireVersion=6, electionId=null, maxDocumentSize=16777216, roundTripTim
eNanos=1345800}
Sep 03, 2021 8:01:41 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:3}] to localhost:27017
Sep 03, 2021 8:01:41 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Closed connection [connectionId{localValue:2, serverValue:3}] to localhost:27017 because the
pool has been closed.
C:\Javawithmongo>


Administrator: Command Prompt - mongo
C:\WINDOWS\system32>mongo
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.0
Server has startup warnings:
2021-09-03T19:50:47.474+0530 I CONTROL [initandlisten]
2021-09-03T19:50:47.474+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-09-03T19:50:47.475+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unre
stricted.
2021-09-03T19:50:47.475+0530 I CONTROL [initandlisten]
2021-09-03T19:50:47.476+0530 I CONTROL [initandlisten] ** WARNING: This server is bound to localhost.
2021-09-03T19:50:47.476+0530 I CONTROL [initandlisten] ** Remote systems will be unable to connect to this server
.
2021-09-03T19:50:47.477+0530 I CONTROL [initandlisten] ** Start the server with --bind_ip <address> to specify wh
ich IP
2021-09-03T19:50:47.477+0530 I CONTROL [initandlisten] ** addresses it should serve responses from, or with --bin
d_ip_all to
2021-09-03T19:50:47.477+0530 I CONTROL [initandlisten] ** bind to all interfaces. If this behavior is desired, st
art the
2021-09-03T19:50:47.478+0530 I CONTROL [initandlisten] ** server with --bind_ip 127.0.0.1 to disable this warning
.
2021-09-03T19:50:47.478+0530 I CONTROL [initandlisten]
2021-09-03T19:50:47.480+0530 I CONTROL [initandlisten]
2021-09-03T19:50:47.483+0530 I CONTROL [initandlisten] ** WARNING: The file system cache of this machine is configured to
be greater than 40% of the total memory. This can lead to increased memory pressure and poor performance.
2021-09-03T19:50:47.486+0530 I CONTROL [initandlisten] See http://dochub.mongodb.org/core/wt-windows-system-file-cache
2021-09-03T19:50:47.496+0530 I CONTROL [initandlisten]
> use myDb
switched to db myDb
> db.Studtalks.find().pretty()
{
"_id" : ObjectId("6132203f4f185d3ed0d863b1"),
"id" : 1,
"Name" : "Thrishin",
"Age" : 19,
"website" : "https://www.youtube.com/c/STUDTalks"
}
>

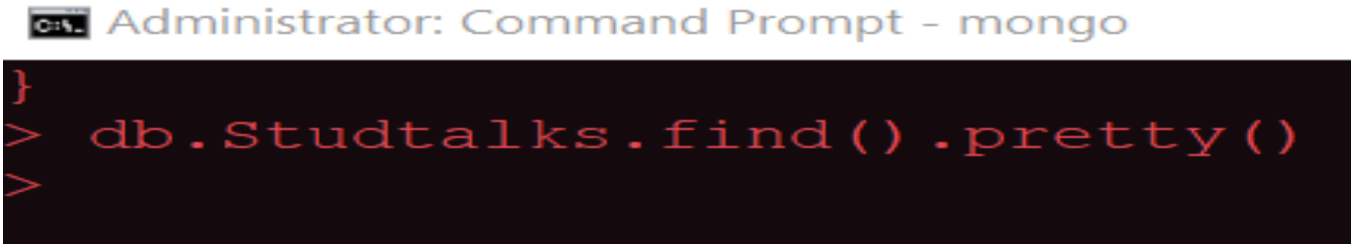
Deleting data from database

DeletingDocuments - Notepad

File Edit Format View Help

```
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import com.mongodb.MongoClient;
import com.mongodb.MongoCredential;
import org.bson.Document;
import com.mongodb.client.model.Filters; //it store filter property
public class DeletingDocuments
{

  public static void main( String args[] )
  {
    MongoClient mongo = new MongoClient( "localhost" , 27017 );
    System.out.println("Connected to the database successfully");
    MongoDatabase database = mongo.getDatabase("myDb");
    MongoCollection<Document> collection = database.getCollection("Studtalks");
    System.out.println("Collection Studtalks selected successfully");
    collection.deleteOne(Filters.eq("id", 1));
    System.out.println("Document deleted successfully...");
  }
}
```

```
C:\Javawithmongo>javac DeletingDocuments.java

C:\Javawithmongo>java DeletingDocuments
Sep 03, 2021 8:13:41 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNK
NOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=500}
Connected to the database successfully
Collection Studtalks selected successfully
Sep 03, 2021 8:13:41 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: No server chosen by PrimaryServerSelector from cluster description ClusterDescription{type=U
NKNOWN, connectionMode=SINGLE, all=[ServerDescription{address=localhost:27017, type=UNKNOWN, state
=CONNECTING}]}. Waiting for 30000 ms before timing out
Sep 03, 2021 8:13:41 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:4}] to localhost:27017
Sep 03, 2021 8:13:41 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=l
ocalhost:27017, type=STANDALONE, state=CONNECTED, ok=true, version=ServerVersion{versionList=[3, 6
, 0]}, minWireVersion=0, maxWireVersion=6, electionId=null, maxDocumentSize=16777216, roundTripTim
eNanos=961800}
Sep 03, 2021 8:13:41 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:5}] to localhost:27017
Document deleted successfully...

C:\Javawithmongo>
```



2.

Code:

Collper_1.html

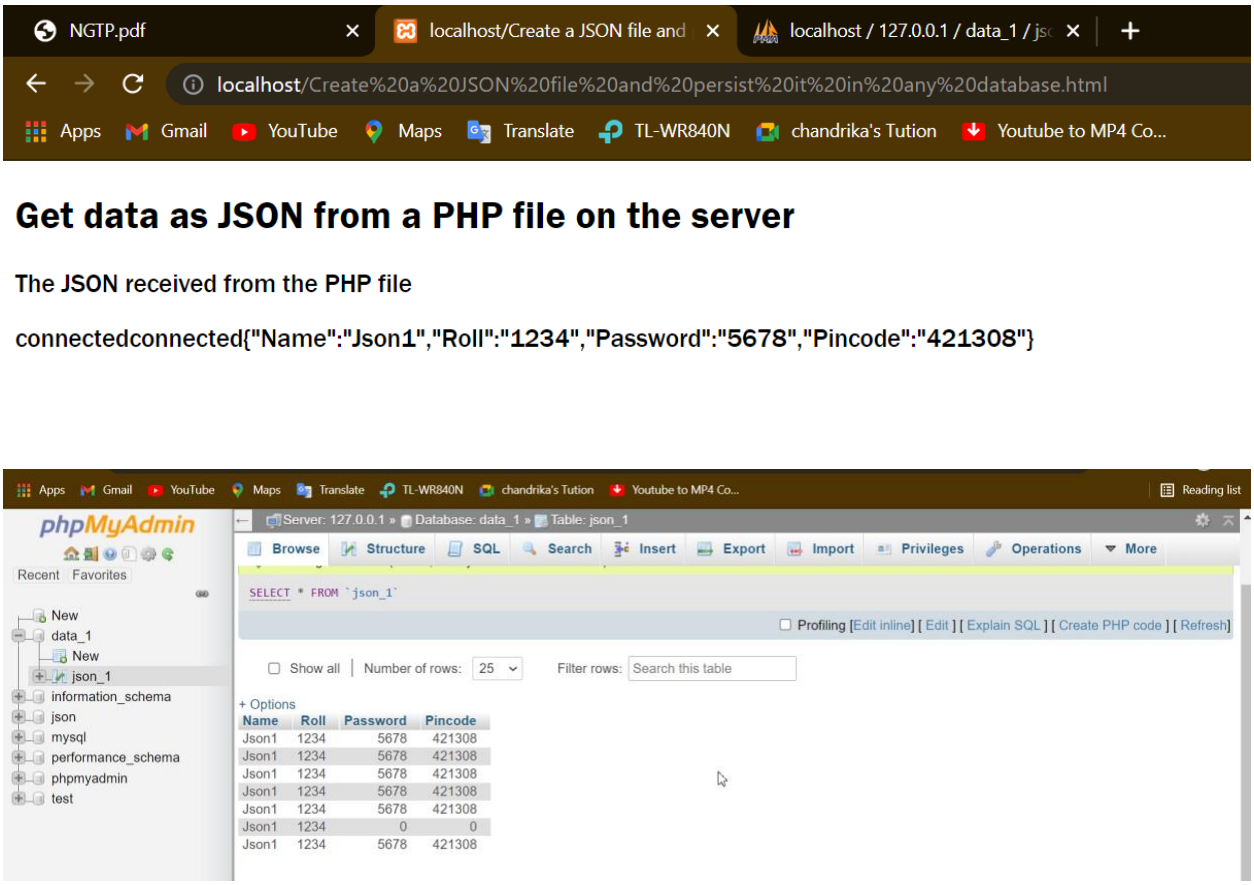
```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
   <h2>Get data as  JSON from a PHP file on the server</h2>
   <p>The JSON received from the PHP file</p>
   <p id="demo"></p>
   <script>
      var obj,dbParam,xmlhttp;
      obj={"Name":"Json1","Roll":"1234","password":"5678","pincode":
"421308"};
      dbParam=JSON.stringify(obj);
      xmlhttp=new XMLHttpRequest();
      xmlhttp.onreadystatechange=function()
      {
         if(this.readyState==4 && this.status==200)
         {
            document.getElementById("demo").innerHTML=this.respons
eText;

         }
      };
```

```
        xmlhttp.open("GET","collpersistence_1.php?x="+dbParam,true);
        xmlhttp.send();

    </script>
</body>
</html>
```

collpersistence_1.php

```
<?php
    header("Content-Type:application/json;charset=UTF-8");
    $obj=json_decode($_GET["x"],false);
    $a=$obj->Name;
    $b=$obj->Roll;
    $c=$obj->password;
    $d=$obj->pincode;
    $con=mysql_connect("Localhost","root","");
    if($con)
    {
        echo "connected";

    }
    $db=mysql_select_db("data_1",$con);
    if($db)
    {
        echo "connected";
    }
    $sql="insert into json_1(Name,Roll,password,pincode) values('$a','$b'
,'$c','$d')";
    $z=mysql_query($sql,$con);
    echo json_encode($obj);
?>
```

Output:


NGTP.pdf
localhost/Create a JSON file and
localhost / 127.0.0.1 / data_1 / js
localhost/Create%20a%20JSON%20file%20and%20persist%20it%20in%20any%20database.html
Apps
Gmail
YouTube
Maps
Translate
TL-WR840N
chandrika's Tution
Youtube to MP4 Co...
Get data as JSON from a PHP file on the server
The JSON received from the PHP file
connectedconnected{"Name":"Json1","Roll":"1234","Password":"5678","Pincode":"421308"}
Reading list
phpMyAdmin
Recent
Favorites
New
data_1
New
json_1
information_schema
json
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: data_1 » Table: json_1
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
More
SELECT * FROM `json_1`
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows:
25
Filter rows:
Search this table
+ Options
Name Roll Password Pincode
Json1 1234 5678 421308
Json1 1234 5678 421308
Json1 1234 5678 421308
Json1 1234 5678 421308
Json1 1234 5678 421308
Json1 1234 0 0
Json1 1234 5678 421308

Slip 3

1.	(A) Import Restaurant.json into MongoDb and perform the following queries 1. Write a MongoDB query to find the restaurant name, borough, longitude and attitude and cuisine for those restaurants which contains 'mon' as three letters somewhere in its name 2. Write a MongoDB query to find the restaurant name, borough, longitude and latitude and cuisine for those restaurants which contain 'Mad' as first three letters of its name 3. Write a MongoDB query to find the restaurants which locate in latitude value less than -95.754168 4. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name. 5. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name. (B) Connect Python with MongoDB and also insert and update documents.	**20**
2.	(A) Write a jQuery to Select elements by class name,id and element name. (B)Write a jQuery to show the use of Click (), hover (), on (), trigger (), off () events.	**20**
3.	Viva	**5**
4.	Journal	**5**

1.

A)

1.

Code:

db.rest.find({name:/.*mon.*/},{"name":1,"borough":1,"address.coord":1,"cuisine":1}).pretty()

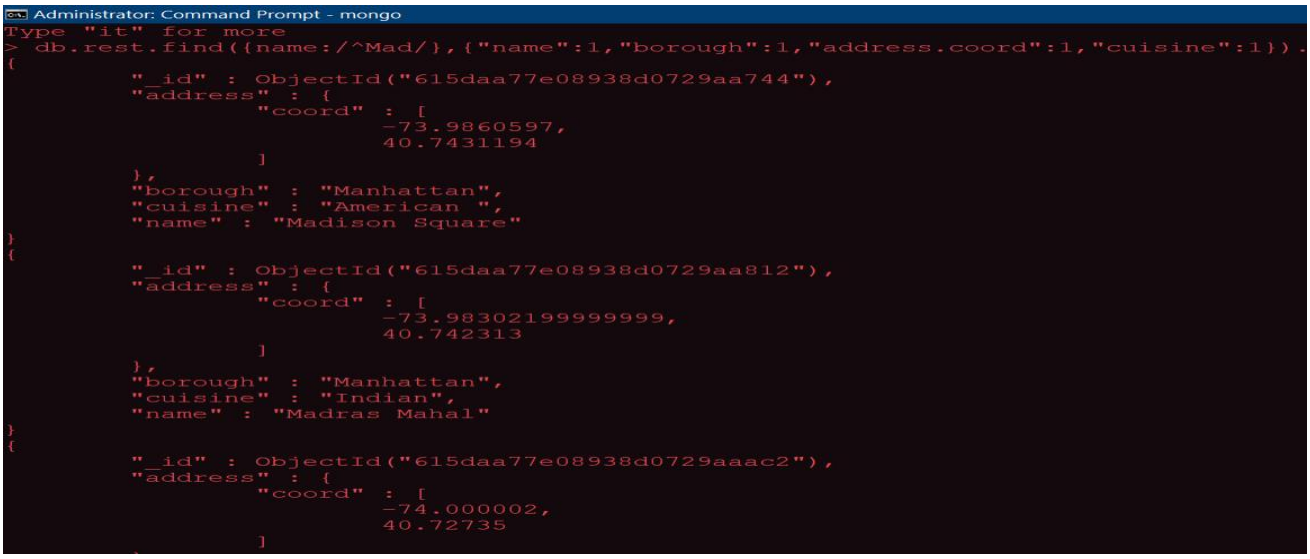
Output:

2.

Code:

db.rest.find({name:/^Mad/},{"name":1,"borough":1,"address.coord":1,"cuisine":1}).pretty()

Output:

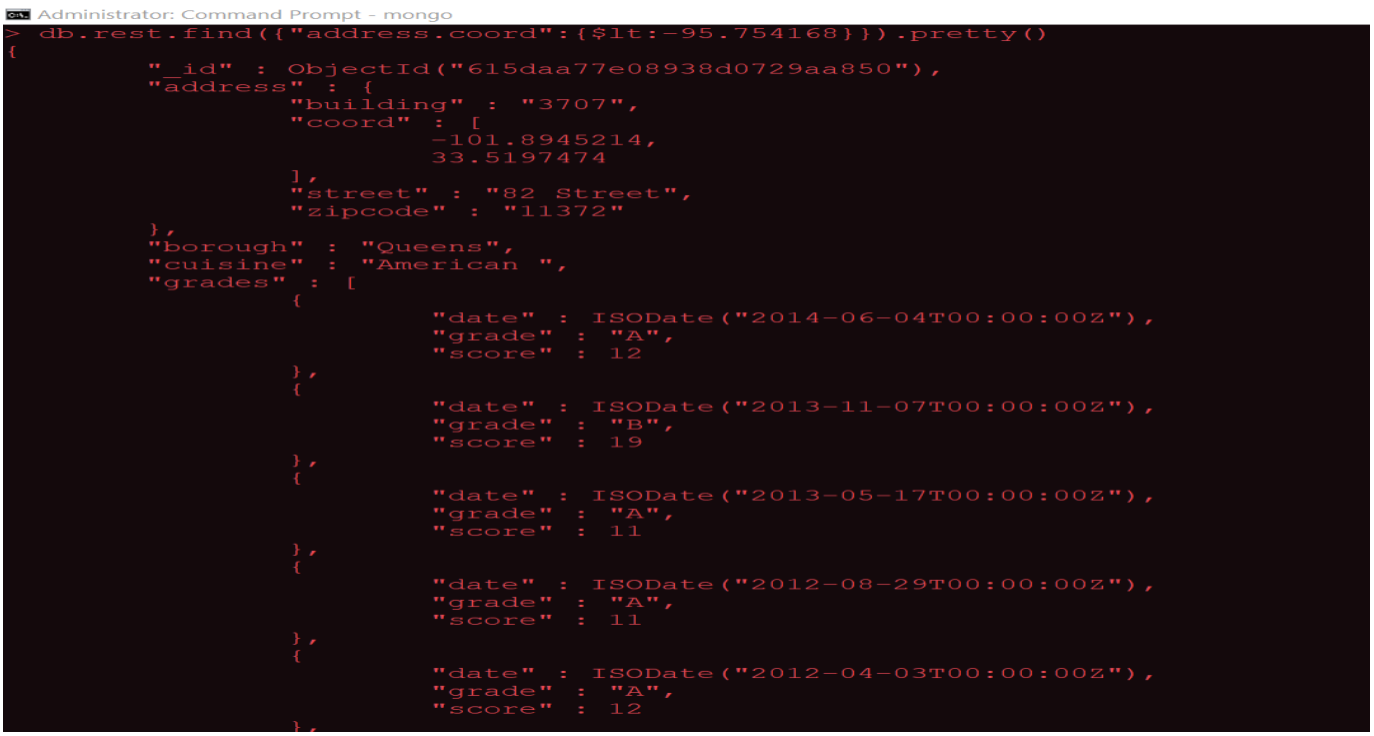


3.

Code:

db.rest.find({"address.coord":{$lt:-95.754168}}).pretty()

Output:



4.

Code:

db.rest.find({name:/^Wil/},{"name":1,"borough":1,"restaurant_id":1,"cuisine":1}).pretty()

Output:

5.

Code:

db.rest.find({name:/ces$/},{"name":1,"borough":1,"restaurant_id":1,"cuisine":1}).pretty()

Output:

1.B)

a. Insert in python in slip no 1

b. update in python

1) Open CMD and install pymongo using the command **pip install pymongo**

2) Open CMD and type **mongod** and open new CMD, type **mongo** and setup connection

```
C:\WINDOWS\system32>mongod
2021-08-27T19:37:02.922+0530 I CONTROL  [initandlisten] MongoDB starting : pid=43572 port=27017 dbpath=C:\data\db\ 64
-bit host=ThrishinAnchan
2021-08-27T19:37:02.926+0530 I CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2021-08-27T19:37:02.927+0530 I CONTROL  [initandlisten] db version v3.6.0
2021-08-27T19:37:02.927+0530 I CONTROL  [initandlisten] git version: a57d8e71e6998a2d0afde7edc11bd23e5661c915
2021-08-27T19:37:02.928+0530 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.0.1u-fips  22 Sep 2016
2021-08-27T19:37:02.928+0530 I CONTROL  [initandlisten] allocator: tcmalloc
2021-08-27T19:37:02.928+0530 I CONTROL  [initandlisten] modules: none
2021-08-27T19:37:02.928+0530 I CONTROL  [initandlisten] build environment:
2021-08-27T19:37:02.928+0530 I CONTROL  [initandlisten]     distmod: 2008plus-ssl
2021-08-27T19:37:02.928+0530 I CONTROL  [initandlisten]     distarch: x86_64
2021-08-27T19:37:02.928+0530 I CONTROL  [initandlisten]     target_arch: x86_64
2021-08-27T19:37:02.928+0530 I CONTROL  [initandlisten] options: {}
2021-08-27T19:37:02.994+0530 I -        [initandlisten] Detected data files in C:\data\db\ created by the 'wiredTiger
' storage engine, so setting the active storage engine to 'wiredTiger'.
2021-08-27T19:37:02.998+0530 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=3541M,session_max=2
0000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=j
ournal,compressor=snappy),file_manager=(close_idle_time=100000),statistics_log=(wait=0),verbose=(recovery_progress),
2021-08-27T19:37:03.586+0530 I STORAGE  [initandlisten] WiredTiger message [1630073223:586056][43572:140729769877152]
, txn-recover: Main recovery loop: starting at 1/50816
2021-08-27T19:37:03.748+0530 I STORAGE  [initandlisten] WiredTiger message [1630073223:747622][43572:140729769877152]
, txn-recover: Recovering log 1 through 2
2021-08-27T19:37:03.907+0530 I STORAGE  [initandlisten] WiredTiger message [1630073223:907765][43572:140729769877152]
, txn-recover: Recovering log 2 through 2
2021-08-27T19:37:05.118+0530 I CONTROL  [initandlisten]
2021-08-27T19:37:05.118+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-08-27T19:37:05.119+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration i
s unrestricted.
```

```
C:\WINDOWS\system32>mongo
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.0
Server has startup warnings:
2021-08-27T19:37:05.118+0530 I CONTROL  [initandlisten]
2021-08-27T19:37:05.118+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-08-27T19:37:05.119+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration
is unrestricted.
2021-08-27T19:37:05.120+0530 I CONTROL  [initandlisten]
2021-08-27T19:37:05.120+0530 I CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2021-08-27T19:37:05.120+0530 I CONTROL  [initandlisten] **          Remote systems will be unable to connect to this
 server.
2021-08-27T19:37:05.121+0530 I CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to spe
cify which IP
2021-08-27T19:37:05.121+0530 I CONTROL  [initandlisten] **          addresses it should serve responses from, or wit
h --bind_ip_all to
2021-08-27T19:37:05.121+0530 I CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desi
red, start the
2021-08-27T19:37:05.122+0530 I CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this
warning.
2021-08-27T19:37:05.123+0530 I CONTROL  [initandlisten]
2021-08-27T19:37:05.124+0530 I CONTROL  [initandlisten]
2021-08-27T19:37:05.125+0530 I CONTROL  [initandlisten] ** WARNING: The file system cache of this machine is configu
red to be greater than 40% of the total memory. This can lead to increased memory pressure and poor performance.
2021-08-27T19:37:05.126+0530 I CONTROL  [initandlisten] See http://dochub.mongodb.org/core/wt-windows-system-file-ca
che
2021-08-27T19:37:05.128+0530 I CONTROL  [initandlisten]
```

Python Code:

Inserting data into database

Python 3.4.0: Insert.py - C:\Python34\Pythonmongodb\Insert.py

File Edit Format Run Options Windows Help

```
from pymongo import MongoClient
client=MongoClient('localhost',27017)
db=client.Studtalks
def insert():
    try:
        Name=input("Enter Name : ")
        Age=input("Enter  Age : ")
        Dept=input("Department : ")
        Contro=input("Contributed for : ")
        db.Teammates.insert(
            {
                "NAME":Name,
                "AGE":Age,
                "Deptname":Dept,
                "ControTo":Contro
            }
        )
        print("Inserted data successfully ")
    except Exception:
        print(str(e))
insert()
```

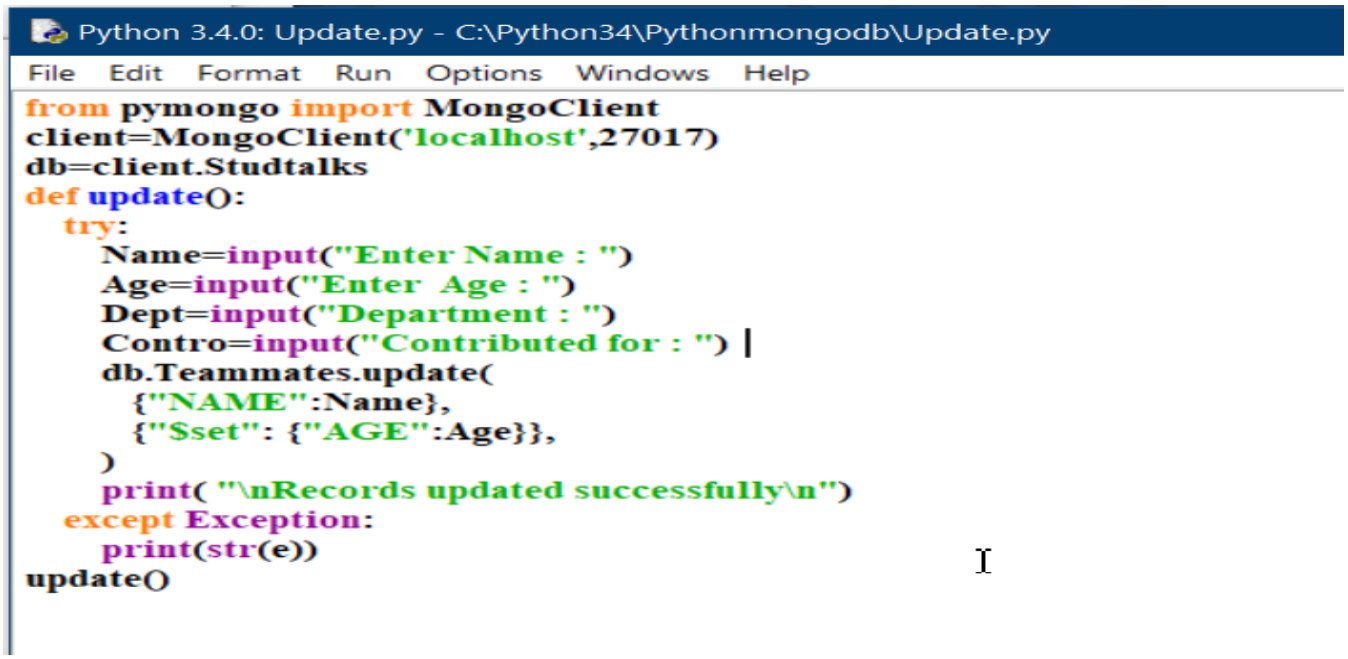
Python 3.4.0 Shell

File Edit Shell Debug Options Windows Help

```
Python 3.4.0 (v3.4.0:04f714765c13, Mar 16 2014, 19:25:23) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ========== RESTART ==========
>>>
Enter Name : Thrishin
Enter  Age : 19
Department : IT
Contributed for : Channel Head and Content Manager
Inserted data successfully
>>>
```

```
> use Studtalks
switched to db Studtalks
> db.Teammates.find().pretty()
{
        "_id" : ObjectId("6128f9c8262348aa4cf50fdb"),
        "NAME" : "Thrishin",
        "Deptname" : "IT",
        "AGE" : "19",
        "ControTo" : "Channel Head and Content Manager"
}
>
```

Updating data in database




```
>>>
Enter Name : Thrishin
Enter  Age : 20
Department : IT
Contributed for : Channel Head and Content Manager

Records updated successfully
```

```
> db.Teammates.find().pretty()
{
        "_id" : ObjectId("6128f9c8262348aa4cf50fdb"),
        "NAME" : "Thrishin",
        "Deptname" : "IT",
        "AGE" : "20",
        "ControTo" : "Channel Head and Content Manager"
}
{
        "_id" : ObjectId("6128fc92f8d0436ffee94fe9"),
        "Deptname" : "IT",
        "AGE" : "20",
        "NAME" : "Aryamaan",
        "ControTo" : "Programming Videos"
}
{
        "_id" : ObjectId("6128fd24ad3a5bfc567dca2b"),
        "NAME" : "Omkar G",
        "Deptname" : "Marketing",
        "AGE" : "20",
        "ControTo" : "Video Editing, 3D gaming videos"
}
>
```

2.A)

Code:

```
<!DOCTYPE html>
<html>

<head>
   <title></title>
   <script src="jquery-3.6.0.min.js"></script>
   <script>
      $(document).ready(
         function () {
            $(".class1").css("background", "yellow");
            $("#id1").css("background", "pink");
```

```
            $("h2").css("background", "green");
        }
    );
  </script>
</head>

<body>
  <p class="class1">Hello!</p>
  <p id="id1">Students</p>
  <h2>welcome in jquery Language!!</h2>
</body>

</html>
```

Output:

2B)

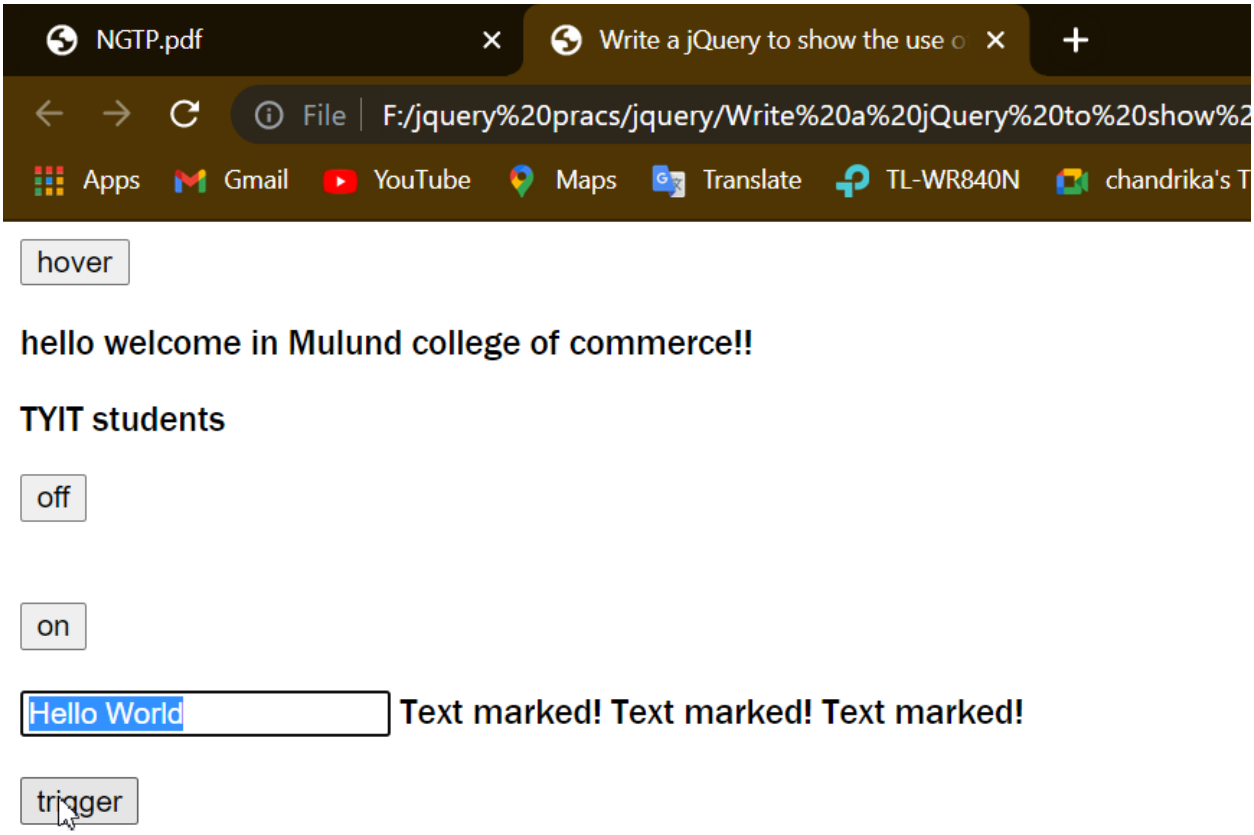
Code:

```
<!DOCTYPE html>
<html>

<head>
  <title></title>
  <script src="jquery-3.6.0.min.js"></script>
  <script>
```

```
        $(document).ready(
            function () {
                $("#b1").hover(
                    function () {
                        document.write("hello world");
                    }
                );
                $("p").on("click", function () {
                    $(this).css("background-color", "pink");
                }
                );
                $("#b2").click(
                    function () {
                        $("p").off("click");
                    }
                );
                $("#b3").on("click", function () {
                    $("#t1").hide();
                }
                );
                $("input").select(
                    function () {
                        $("input").after(" Text marked!");
                    }
                );
                $("#b4").click(
                    function () {
                        $("input").trigger("select");
                    }
                );
            }
        );
    </script>
</head>
```

```
<body>
   <button id="b1">hover</button><br />
   <p>hello welcome in Mulund college of commerce!!</p>
   <p>TYIT students</p>
   <button id="b2">off</button><br /><br />
   <p id="t1">Hello world</p><br />
   <button id="b3">on</button><br /><br />
   <input type="text" value="Hello World"><br><br>
   <button id="b4">trigger</button>
</body>

</html>
```

Output:



Slip 4:

UNIVERSITY OF MUMBAI
T.Y.B.Sc.(INFORMATION TECHNOLOGY) (Semester– V) (Practical) EXAMINATION
SECOND HALF 2018
NEXT GENERATION TECHNOLOGIES

Seat No. : ____ **Max. Marks:50**

1.	Connect PHP with MongoDB and also insert, retrieve, update and delete documents.	**20**
2.	(A) Create a JSON file and parse it. (B) Write a jQuery to Create slide-up, slide-down and slide-Toggle effect.	**20**
3.	Viva	**5**
4.	Journal	**5**

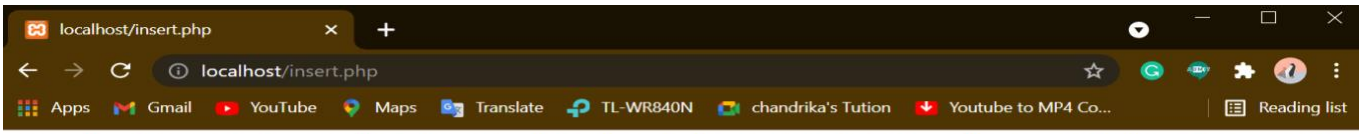
1.

Inserting data into database

insert - Notepad

File Edit Format View Help

```php
<?php
 $m=new MongoClient();
 echo "Connection to database successfully";
 $db=$m->mydb;
 echo "Database mydata selected";
 $collection=$db->Studtalks;
 echo "Collection selected succesfully";
 $document=array(
    "Name"=>"Thrishin",
    "Department"=>"IT",
    "Age"=>20,
    "URL"=>"https://www.youtube.com/c/STUDTalks",
    "Channel"=>"Studtalks"
 );
 $collection-> insert($document);
 echo "Documented inserted succesfully";
 ?>
```



Connection to database successfullyDatabase mydata selectedCollection selected succesfullyDocumented inserted succesfully

Administrator: Command Prompt - mongo

```
> use mydb
switched to db mydb
> db.Studtalks.find().pretty()
{
        "_id" : ObjectId("613239bf771aece81d00002a"),
        "Name" : "Thrishin",
        "Department" : "IT",
        "Age" : 20,
        "URL" : "https://www.youtube.com/c/STUDTalks",
        "Channel" : "Studtalks"
}
>
```

Updating data in database

update - Notepad

File Edit Format View Help

```
<?php
  $m = new MongoClient();
  echo "Connection to database successfully ";
  $db = $m->mydb;
  echo "Database mydb selected ";
  $collection = $db->Studtalks;
  echo "Collection selected succsessfully ";
  $collection->update(array("Name"=>"Thrishin"),
  array('$set'=>array("Name"=>"Anchan Thrishin Sanjeeva")));
  echo "Document updated successfully ";
  $cursor = $collection->find();
  echo "Updated document ";
  foreach ($cursor as $document) {
    echo $document["Name"] . "<br>";
  }
?>
```

Connection to database successfully Database mydb selected Collection selected succsessfully Document updated successfully Updated document Anchan Thrishin Sanjeeva

Select Administrator: Command Prompt - mongo

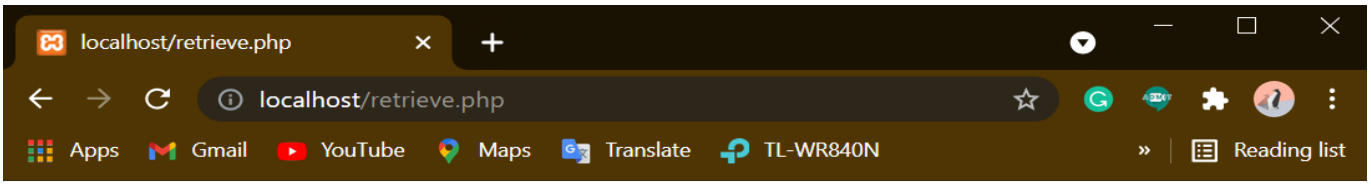
```
        "Channel" : "Studtalks"
}
> db.Studtalks.find().pretty()
{
        "_id" : ObjectId("61323dfb771aec24ca00002c"),
        "Name" : "Anchan Thrishin Sanjeeva",
        "Department" : "IT",
        "Age" : 20,
        "URL" : "https://www.youtube.com/c/STUDTalks",
        "Channel" : "Studtalks"
}
>
```

Retrieving data from database

retrieve - Notepad

File Edit Format View Help

```
<?php
  $m = new MongoClient();
  echo "Connection to database successfully ";
  $db = $m->mydb;
  echo "Database mydb selected ";
  $collection = $db->Studtalks;
  echo "Collection selected succsessfully ";
  $cursor = $collection->find();
  foreach ($cursor as $document) {
    echo "<br>". $document["Name"] . "<br>";
    echo "<br>". $document["Department"] . "<br>";
    echo "<br>". $document["Age"] . "<br>";
    echo "<br>". $document["URL"] . "<br>";
    echo "<br>". $document["Channel"] . "<br>";
  }
?>
```



Connection to database successfully Database mydb selected Collection selected succsessfully Thrishin

IT

20

https://www.youtube.com/c/STUDTalks

Studtalks

Deleting data from database

delete - Notepad

File Edit Format View Help

```
<?php
   $m = new MongoClient();
   echo "Connection to database successfully ";
   $db = $m->mydb;
   echo "Database mydb selected";
   $collection = $db->Studtalks;
   echo "Collection selected succsessfully ";
   $collection->remove(array("Name"=>"Anchan Thrishin Sanjeeva"));
   echo "Documents deleted successfully ";
   $cursor = $collection->find();
   echo "Updated document ";
   foreach ($cursor as $document) {
      echo $document["Name"] . "\n";
   }
?>
```

Connection to database successfully Database mydb selectedCollection selected succsessfully Documents deleted successfully Updated document

2.A)

Code:

```
html>
```

```
<head>
<title>parsing</title>
</head>
<body>
<script type="text/javascript">
   var data='{"name":"Ravi","id":5,"Gender":"male"}';

   var json_obj=JSON.parse(data,function(name,value)
                {
                   return value;
                }
             );
   document.writeln(json_obj.name);
   for(key in json_obj)
   {
      document.write("</br>"+key+ ":" +json_obj[key]);
   }
   var json_obj=JSON.parse(data,function(name,value)
                {
                   if(name=="id")
                   {
                      return undefined;
                   }
                   else
                   {
                      return value;
                   }
                }
             );
   for(key in json_obj)
   {
      document.write("</br>"+key+ ":" +json_obj[key]);
   }
</script>
</body>
```

```
</html>
```

Output:

Ravi
name:Ravi
id:5
Gender:male
name:Ravi
Gender:male

B)

Code:

```
<!DOCTYPE html>
<html>

<head>
    <title></title>
    <script src="jquery-3.6.0.min.js"></script>
    <script>
        $(document).ready(
            function () {
                $(".up-btn").click(
                    function () {
                        $("p").slideUp();
                    }
                );
                $(".down-btn").click(
                    function () {
                        $("p").slideDown();
                    }
```

```
            );
            $(".toggle-btn").click(
                function () {
                    $("p").slideToggle();
                }
            );
        }
    );
  </script>
</head>

<body>
  <button type="button" class="up-btn">Slide Up Paragraphs</button>
  <button type="button" class="down-
btn">Slide Down Paragraphs</button>
  <button type="button" class="toggle-
btn">Slide Toggle Paragraphs</button>
  <p>This is a paragraph.</p>
  <p>This is another paragraph.</p>

</body>

</html>
```

Output:

Slide Up:

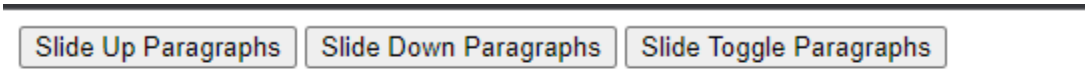
It will slide up or hide

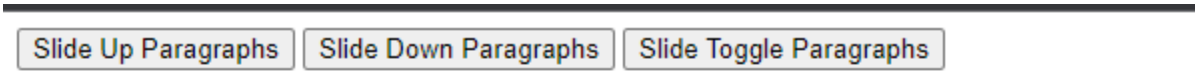
Slide down :

It will come or it will show

Toggle:

It will hide and show





This is a paragraph.

This is another paragraph.

Slip 5

NEXT GENERATION TECHNOLOGIES

Seat No. : ____ Max. Marks: 50

1.	(A) Import Restaurant.json into MongoDb and perform the following queries 1. Write a MongoDB query to find the restaurants that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168. 2. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168. 3. Write a MongoDB query to find the restaurants which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the borough Brooklyn. The document must be displayed according to the cuisine in descending order 4. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Wil' as first three letters for its name. 5. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'ces' as last three letters for its name. (B) Write a MongoDB query to create a backup of existing database and also create a backup of existing database.	20
2.	A. Write a jQuery to animate multiple CSS properties. B. Write a jQuery to Change text contents of the elements on button click.	20
3.	Viva	5
4.	Journal	5

1.A)

1.

Code:

```
db.rest.find( {$and: [ {"cuisine" : {$ne :"American "}}, {"grades.score" : {$gt : 70}}, {"address.coord" : {$lt : -65.754168}} ] } ).pretty()
```

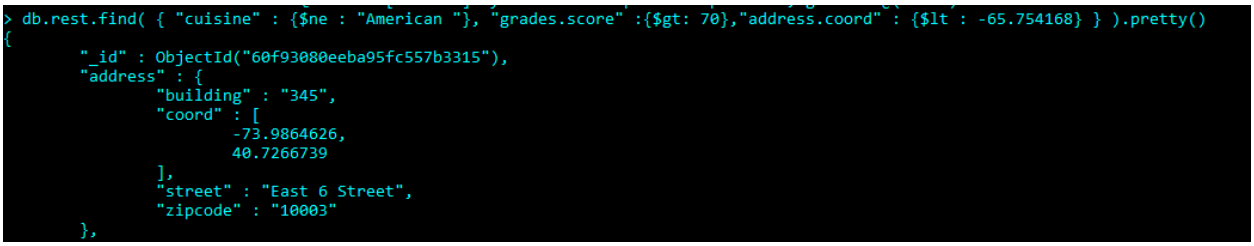
Output:

```
> db.rest.find( {$and: [ {"cuisine" : {$ne :"American "}}, {"grades.score" : {$gt : 70}}, {"address.coord" : {$lt : -65.754168}} ] } ).pretty()
{
        "_id" : ObjectId("60f93080eeba95fc557b3315"),
        "address" : {
                "building" : "345",
                "coord" : [
                        -73.9864626,
                        40.7266739
```

2.

db.rest.find({ "cuisine" : {$ne : "American "}, "grades.score" :{$gt: 70},"address.coord" : {$lt : -65.754168} }).pretty()

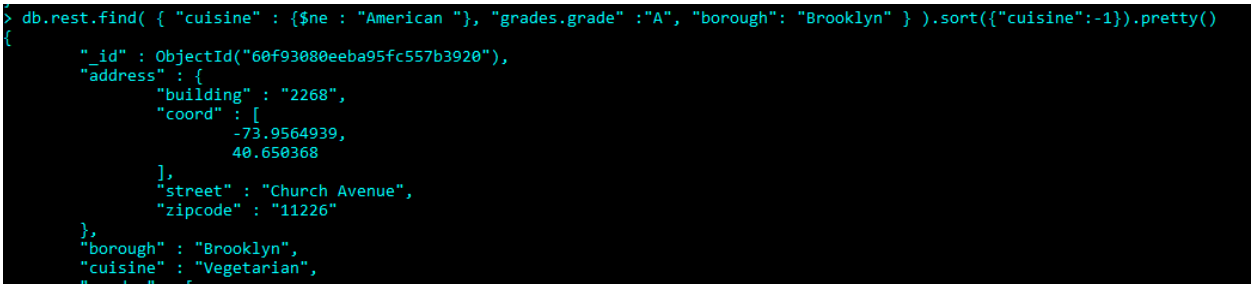
Output:



3.

db.rest.find({ "cuisine" : {$ne : "American "}, "grades.grade" :"A", "borough": "Brooklyn" }).sort({"cuisine":-1}).pretty()

Output:



4.

Slip no 3

5.

Slip no 3

1.B)

```
> use tymongo
switched to db tymongo
> db.rest.find()
{ "_id" : ObjectId("60f93080eeba95fc557b3117"), "address" :
 : "11224" }, "borough" : "Brooklyn", "cuisine" : "American
("2013-06-05T00:00:00Z"), "grade" : "A", "score" : 7 }, { "
00Z"), "grade" : "A", "score" : 12 } ], "name" : "Riviera C
```

To delete the database

Again open the command prompt type mongodump

```
C:\WINDOWS\system32>mongodump
2021-10-06T01:55:24.366+0530    writing admin.system.version to
2021-10-06T01:55:24.371+0530    done dumping admin.system.version (1 document)
2021-10-06T01:55:24.373+0530    writing tymongo.rest to
2021-10-06T01:55:24.375+0530    writing EMPLOYEE.EMP to
2021-10-06T01:55:24.376+0530    writing EMPLOYEE.employee to
2021-10-06T01:55:24.378+0530    writing tymongo.stud to
2021-10-06T01:55:24.379+0530    done dumping EMPLOYEE.EMP (11 documents)
2021-10-06T01:55:24.380+0530    writing EMLOYEE.students to
```

In mongo type db.dropDatabase()

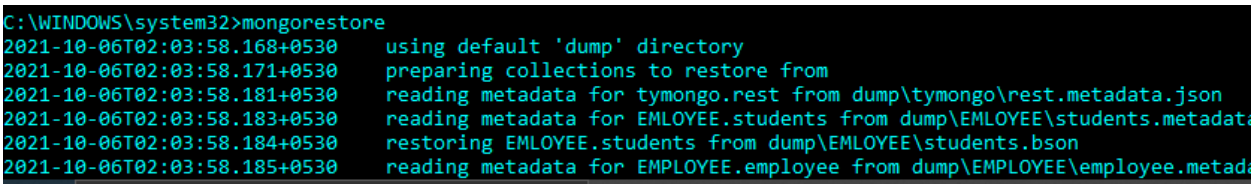
```
> db.dropDatabase()
{ "dropped" : "tymongo", "ok" : 1 }
>
```

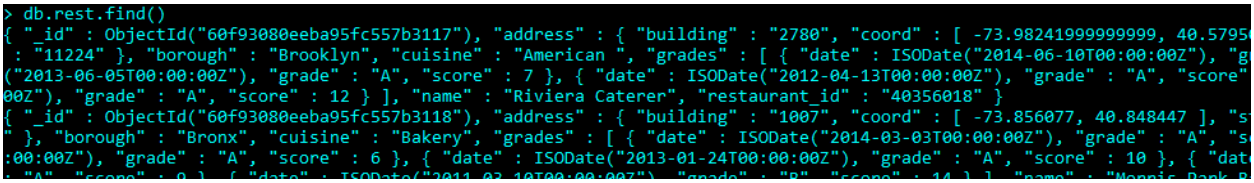
```
> show dbs
EMLOYEE     0.000GB
EMPLOYEE    0.000GB
TY          0.000GB
admin       0.000GB
config      0.000GB
local       0.000GB
tyit219716  0.000GB
>
```

```
> show dbs
EMLOYEE     0.000GB
EMPLOYEE    0.000GB
TY          0.000GB
admin       0.000GB
config      0.000GB
local       0.000GB
tyit219716  0.000GB
> use tymongo
switched to db tymongo
> db.rest.find()
>
```

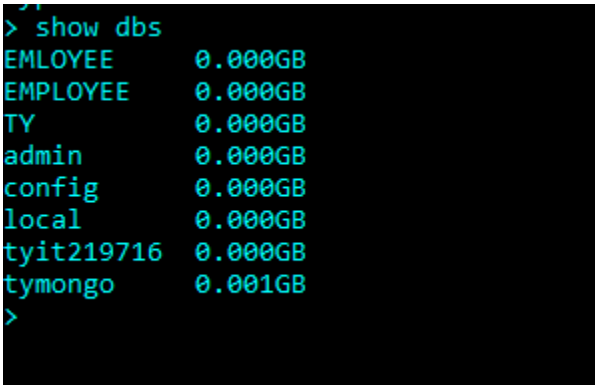
How to backup

To backup type mongorestore





Now see all the files have restore



tymongo as restore

if you want to restore another database use have to switch the another database and type in another command prompt mongorestore like have done about the database TY

```
> use TY
switched to db TY
> show dbs
EMLOYEE     0.000GB
EMPLOYEE    0.000GB
TY          0.000GB
admin       0.000GB
config      0.000GB
local       0.000GB
tyit219716  0.000GB
tymongo     0.001GB
> use TY
switched to db TY
> show collections
phpdata
samp
st
> db.phpdata.find()
{ "_id" : ObjectId("615ca32966a195ec1d00002b"), "Name" : "Datta", "State" : "maharashtra", "Gender" : "Male", "title" : "Datta" }
>
```

2.A)

Code:

Slip no 1

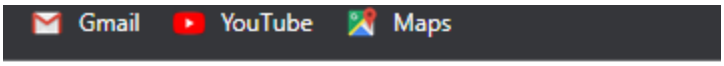
B)

Code:

```
<!DOCTYPE html>
<html>

<head>
    <title></title>
    <script src="jquery-3.6.0.min.js"></script>
    <script>
        $(document).ready(
            function () {
                $("button").click(
                    function () {
```

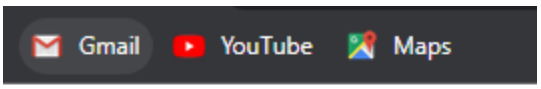
```
                document.write("hello world");
            }
        );
    }
);
</script>
</head>

<body>
    <p>Hello! Welcome in Jquery Language!!</p>
    <button>Click me</button>
</body>

</html>
```

Output:



Hello! Welcome in Jquery Language!!

Click me



hello world

Slip 6:

UNIVERSITY OF MUMBAI

T.Y.B.Sc.(INFORMATION TECHNOLOGY) (Semester– V)(Practical) EXAMINATION

SECOND HALF 2018

NEXT GENERATION TECHNOLOGIES

Seat No. : ____ **Max. Marks:50**

1.	Write a MongoDB query to create Replica and backup of existing database.	**20**
2.	(A) Write a jQuery to add and remove CSS classes from the HTML elements. (B) Write a jQuery to set the duration in slide toggle effect.	**20**
3.	Viva	**5**
4.	Journal	**5**

1.

Server 1

```
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>start mongod --replSet tyit --logpath \data\rs1\1.log --dbpath \data\rs1 --port 27017 --smallfiles
-oplogSize 64

C:\WINDOWS\system32>
```

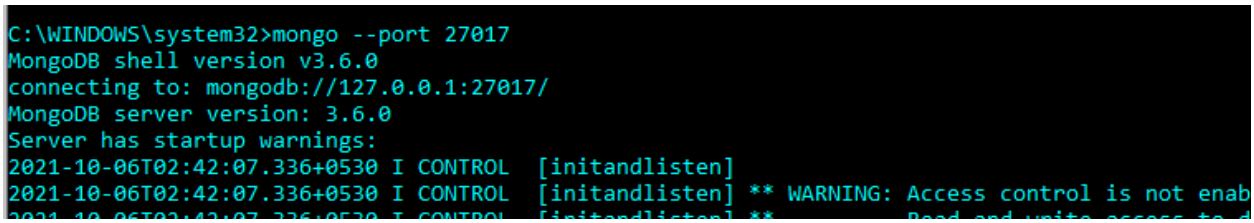
Server 2

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>start mongod --replSet tyit --logpath \data\rs1\2.log --dbpath \data\rs2 --port 27018 --smallfiles
-oplogSize 64

C:\WINDOWS\system32>
```

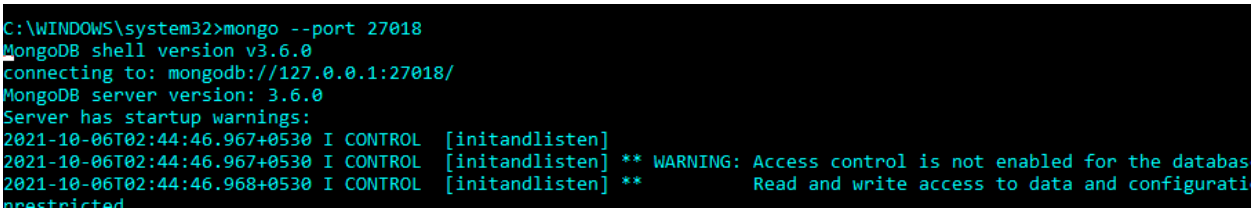
Server 3

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19042.1237]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>start mongod --replSet tyit --logpath \data\rs1\3.log --dbpath \data\rs3 --port 27019 --smallfiles -
-oplogSize 64

C:\WINDOWS\system32>
```

Port of server 1



Port of server 2



Port of server 3

Configuration

On server 1

```
> config={
... _id:"tyit",
... members:[
... {
... _id:0, host:"localhost:27017"},
... {
... _id:1, host:"localhost:27018"},
... {
... _id:2, host:"localhost:27019"}
... ]
... }
{
        "_id" : "tyit",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "localhost:27017"
                },
                {
                        "_id" : 1,
                        "host" : "localhost:27018"
                },
                {
                        "_id" : 2,
                        "host" : "localhost:27019"
                }
        ]
}
>
```

```
> rs.initiate(config)
{
        "ok" : 1,
        "operationTime" : Timestamp(1633469520, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1633469520, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
tyit:OTHER>
```

Server 2 and Server 3

```
> rs.slaveOk()
tyit:SECONDARY>
```

```
> rs.slave.Ok()
2021-10-06T03:05:37.246+0530 E QUERY    [thread1] TypeError: rs.slave is undefined :
@(shell):1:1
tyit:SECONDARY> rs.slaveOk()
tyit:SECONDARY>
```

We have to insert the documents

```
tyit:OTHER> db.student.insert(
... {
... "name":"Hello",
... "class":"ty"
... }
... )
WriteResult({ "nInserted" : 1 })
tyit:PRIMARY>
```

Now we will see the secondary server also have inserted the documents

Server 2

```
tyit:SECONDARY> db.student.find().pretty()
{
        "_id" : ObjectId("615cc5efc3827f7a90f62789"),
        "name" : "Hello",
        "class" : "ty"
}
tyit:SECONDARY>
```

Server 3:

```
tyit:SECONDARY> db.student.find().pretty()
{
        "_id" : ObjectId("615cc5efc3827f7a90f62789"),
        "name" : "Hello",
        "class" : "ty"
}
tyit:SECONDARY>
```

Now we will delete

```
C:\WINDOWS\system32>mongodump
2021-10-06T03:16:27.791+0530    writing admin.system.version to
2021-10-06T03:16:27.797+0530    done dumping admin.system.version (1 document)
2021-10-06T03:16:27.797+0530    writing test.student to
2021-10-06T03:16:27.801+0530    done dumping test.student (1 document)

C:\WINDOWS\system32>
```

the records and database

```
tyit:PRIMARY> db.dropDatabase()
{
        "dropped" : "test",
        "ok" : 1,
        "operationTime" : Timestamp(1633470452, 2),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1633470452, 2),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
tyit:PRIMARY>
```

```
tyit:PRIMARY> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
tyit:PRIMARY>
```

All the record is deleted from the database and collections from primary and the secondary server

```
tyit:SECONDARY> db.student.find().pretty()
tyit:SECONDARY>
```

Now we will restore the database and collections

```
C:\WINDOWS\system32>mongorestore
2021-10-06T03:21:27.429+0530    using default 'dump' directory
2021-10-06T03:21:27.432+0530    preparing collections to restore from
2021-10-06T03:21:27.442+0530    reading metadata for tymongo.rest from dump
2021-10-06T03:21:27.443+0530    reading metadata for EMPLOYEE.employee from
2021-10-06T03:21:27.443+0530    reading metadata for EMLOYEE.students from
2021-10-06T03:21:27.444+0530    reading metadata for tyit219716.sampleColle
etadata.json
```

We have backup the collections

```
tyit:PRIMARY> db.student.find().pretty()
{
        "_id" : ObjectId("615cc5efc3827f7a90f62789"),
        "name" : "Hello",
        "class" : "ty"
}
tyit:PRIMARY>
```

```
tyit:SECONDARY> db.student.find().pretty()
{
        "_id" : ObjectId("615cc5efc3827f7a90f62789"),
        "name" : "Hello",
        "class" : "ty"
}
tyit:SECONDARY>
```

```
tyit:SECONDARY> db.student.find().pretty()
{
        "_id" : ObjectId("615cc5efc3827f7a90f62789"),
        "name" : "Hello",
        "class" : "ty"
}
tyit:SECONDARY>
```

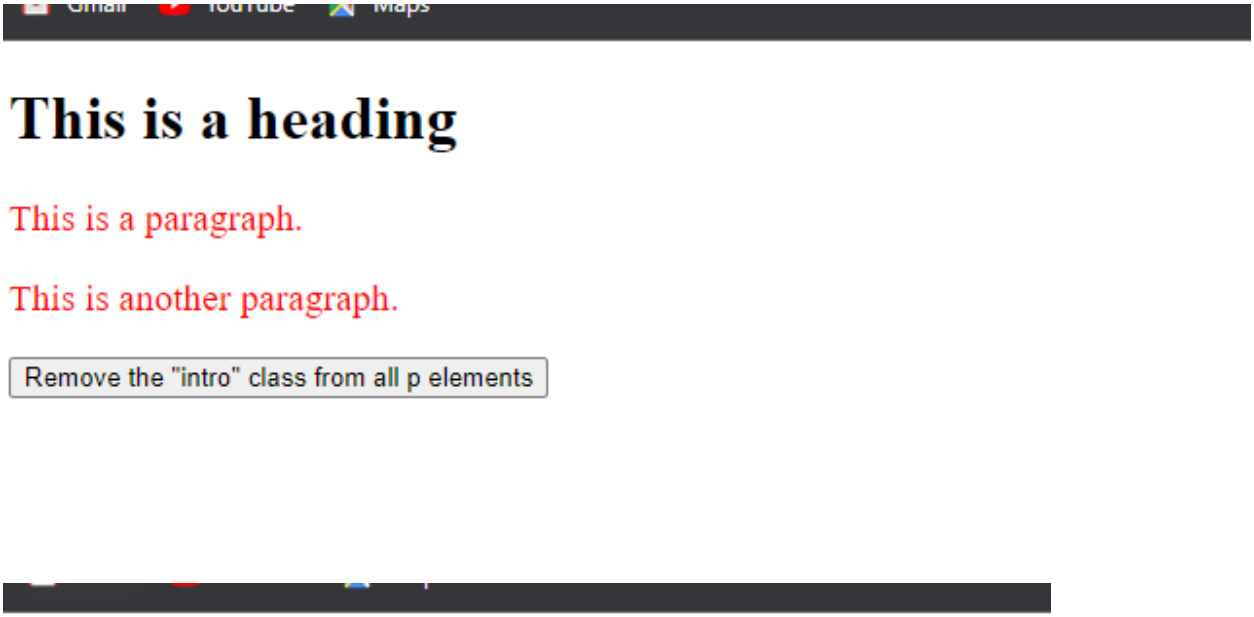
2.A)

Code:

```
<!DOCTYPE html>
<html>

<head>
    <title></title>
    <script src="jquery-3.6.0.min.js"></script>
    <script>
        $(document).ready(
            function () {
                $("button").click(
```

```
                function () {
                    $("p").removeClass("intro");
                }
            );
        }
    );
</script>
<style>
    .intro {
        font-size: 120%;
        color: red;
    }
</style>
</head>

<body>
<h1>This is a heading</h1>
<p class="intro">This is a paragraph.</p>
<p class="intro">This is another paragraph.</p>
<button>Remove the "intro" class from all p elements</button>
</body>

</html>
```

Output:



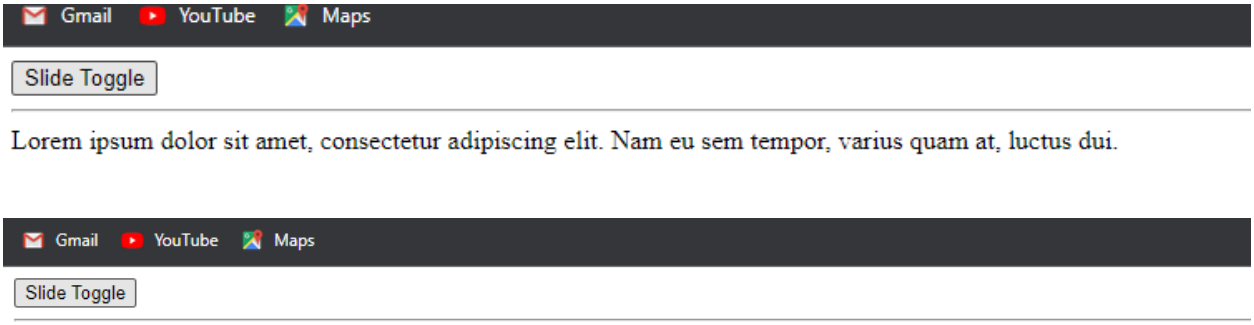
B)

Code:

```
<!DOCTYPE html>
<html>

<head>
   <title></title>
   <script src="jquery-3.6.0.min.js"></script>
   <script>
```

```
        $(document).ready(
            function () {
                $(".b1").click(
                    function () {
                        $(".box").slideToggle();
                    }
                );
            }
        );
    </script>
</head>

<body>
    <button type="button" class="b1">Slide Toggle</button>
    <hr>
    <div class="box">
        <div class="box-
inner">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu
sem tempor, varius quam
            at, luctus dui. </div>
    </div>
</body>

</html>
```

Output:



Slip 7

1.	(A) Import Restaurant.json into MongoDb and perform the following queries 1. Write a MongoDB query to display all the documents in the collection restaurants. 2. Write a MongoDB query to display the fields, restaurant_id, name, borough and cuisine for all the documents in the collection restaurant. 3. Write a MongoDB query to display the fields restaurant_id, name, borough and cuisine, but exclude the field _id for all the documents in the collection restaurant 4. Write a MongoDB query to display the fields restaurant_id, name, borough and zip code, but exclude the field _id for all the documents in the collection restaurant 5. Write a MongoDB query to display all the restaurant which is in the borough Bronx (B) Connect Python with MongoDB and also insert and delete documents.	**20**
2.	A. Write a jQuery to insert multiple HTML elements at the beginning and end of the elements. B. Write a jQuery to create your own Customized event.	**20**
3.	Viva	**5**
4.	Journal	**5**

1.A)

1.

Code:

Slip no 1

2.

code:

slip no 1

3, and 4. And 5

Code:

Slip no 1

!.B)

Insert

Slip no 1

Delete

Code:

```
> db.st.find()
{ "_id" : ObjectId("615c680fb88f2bf8a4f5afc0"), "Name" : "mahi", "Age" : "50", "Gender" : "Male" }
>
```

```
from pymongo import MongoClient
client=MongoClient('localhost:27017')
db=client.TY
def delete():
    try:
        name=input("\n Enter the Name: ")
        db.st.remove(
            {"Name":name}
            )
        print("\n Deletion successfully \n ")
    except (Exception):
        print(str(e))


delete()
```

Output:

```
tel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\TY bscit\Semester V\practicals\next generation technology\python_
ongo\delete.py

 Enter the Name: mahi

Warning (from warnings module):
  File "E:\TY bscit\Semester V\practicals\next generation technology\python_mon
o\delete.py", line 7
    db.st.remove(
DeprecationWarning: remove is deprecated. Use delete_one or delete_many instead

 Deletion successfully

>>>
```

```
> db.st.find()
{ "_id" : ObjectId("615c680fb88f2bf8a4f5afc0"), "Name" : "mahi", "Age" : "50", "Gender" : "Male" }
> db.st.find()
>
```

2.

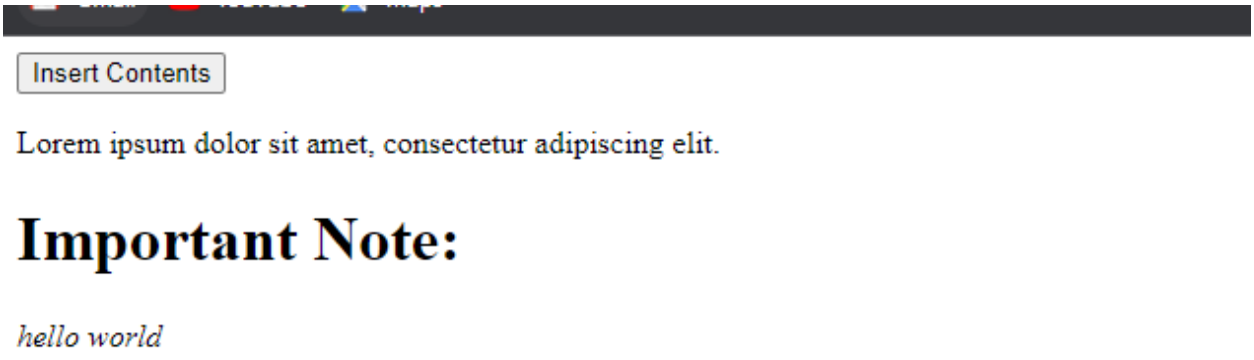
A)

Code:

```
<!DOCTYPE html>
<html>

<head>
    <title></title>
    <script src="jquery-3.6.0.min.js"></script>
    <script>
        $(document).ready(
            function () {
                $(".b1").click(
                    function () {
                        var newHeading = "<h1>Important Note:</h1>";
                        var newParagraph = document.createElement("p");
                        newParagraph.innerHTML = "<em>hello world</em>";
                        $("body").append(newHeading, newParagraph);
```

```
                }
            );
        }
    );
  </script>
</head>

<body>
  <button class="b1">Insert Contents</button>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. </p>
</body>

</html>
```

Output:



B) part to own customized you can choose the any element of the event in jquery

Slip no : 8

UNIVERSITY OF MUMBAI
T.Y.B.Sc.(INFORMATION TECHNOLOGY) (Semester– V) (Practical) EXAMINATION
SECOND HALF 2018
NEXT GENERATION TECHNOLOGIES

Seat No. : ____ **Max. Marks: 50**

1.	Connect Python with MongoDB and also insert, retrieve, update and delete documents.	**20**
2.	Create a JSON file and persist it in any database.	**20**
3.	Viva	**5**
4.	Journal	**5**

1.

Code:

Slip no 1,3,7

2.

Slip no 2

Slip no :9

UNIVERSITY OF MUMBAI
T.Y.B.Sc. (INFORMATION TECHNOLOGY) (Semester– V)(Practical) EXAMINATION
SECOND HALF 2018
NEXT GENERATION TECHNOLOGIES

Seat No. : ____ Max. Marks: 50

1.	(A) Import Restaurant.json into MongoDb and perform the following queries 1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which contain 'Reg' as three letters somewhere in its name 2. Write a MongoDB query to find the restaurants which belong to the borough Bronx and prepared either American or Chinese dish. 3. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which belong to the borough Staten Island or Queens or Bronxor Brooklyn 4. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which are not belonging to the borough Staten Island or Queens or Bronxor Brooklyn. 5. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which achieved a score which is not more than 10. (B) Connect Python with MongoDB and also insert and retrieve documents.	20
2.	A. Write a jQuery Create animation effect B. Write a jQuery to perform Method chaining.	20
3.	Viva	5
4.	Journal	5

1.A

1.

Code

Slip no 3 only difference is write Reg delete mon

2.

Code:

```
db.rest.find( { "borough": "Bronx" , $or : [ { "cuisine" : "American " }, { "cuisine" : "Chinese" }] } ).pretty()
```

Output:

```
> db.rest.find( { "borough": "Bronx" , $or : [ { "cuisine" : "American " }, { "cuisine" : "Chinese" }] } ).pretty()
{
        "_id" : ObjectId("60f93080eeba95fc557b3123"),
        "address" : {
                "building" : "2300",
                "coord" : [
                        -73.8786113,
                        40.8502883
                ],
                "street" : "Southern Boulevard",
                "zipcode" : "10460"
        },
        "borough" : "Bronx",
        "cuisine" : "American ",
        "grades" : [
                {
                        "date" : ISODate("2014-05-28T00:00:00Z"),
```

3.

Code:

db.rest.find({"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 }).pretty()

Output:

```
> db.rest.find( {"borough" :{$in :["Staten Island","Queens","Bronx","Brooklyn"]}}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty()
{
        "_id" : ObjectId("60f93080eeba95fc557b3117"),
        "borough" : "Brooklyn",
        "cuisine" : "American ",
        "name" : "Riviera Caterer",
        "restaurant_id" : "40356018"
}
{
        "_id" : ObjectId("60f93080eeba95fc557b3118"),
        "borough" : "Bronx",
        "cuisine" : "Bakery",
        "name" : "Morris Park Bake Shop",
        "restaurant_id" : "30075445"
```

4.

Code:

db.rest.find({"borough" :{$nin :["Staten Island","Queens","Bronx","Brooklyn"]}}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 }).pretty()

Output:

```
> db.rest.find( {"borough" :{$nin :["Staten Island","Queens","Bronx","Brooklyn"]}}, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty()
{
        "_id" : ObjectId("60f93080eeba95fc557b311f"),
        "borough" : "Manhattan",
        "cuisine" : "Irish",
        "name" : "Dj Reynolds Pub And Restaurant",
        "restaurant_id" : "30191841"
}
{
        "_id" : ObjectId("60f93080eeba95fc557b3122"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "1 East 66Th Street Kitchen",
        "restaurant_id" : "40359480"
}
{
        "_id" : ObjectId("60f93080eeba95fc557b3128"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "Glorious Food",
        "restaurant_id" : "40361521"
}
{
```

5,

Code:

db.rest.find({"grades.score" : { $not: {$gt : 10} } }, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 }).pretty()

Output:

```
> db.rest.find( {"grades.score" : { $not: {$gt : 10} } }, { "restaurant_id" : 1, "name":1,"borough":1, "cuisine" :1 } ).pretty()
{
        "_id" : ObjectId("60f93080eeba95fc557b3120"),
        "borough" : "Brooklyn",
        "cuisine" : "American ",
        "name" : "C & C Catering Service",
        "restaurant_id" : "40357437"
}
{
        "_id" : ObjectId("60f93080eeba95fc557b3122"),
        "borough" : "Manhattan",
        "cuisine" : "American ",
        "name" : "1 East 66Th Street Kitchen",
        "restaurant_id" : "40359480"
}
{
        "_id" : ObjectId("60f93080eeba95fc557b3127"),
        "borough" : "Brooklyn",
```

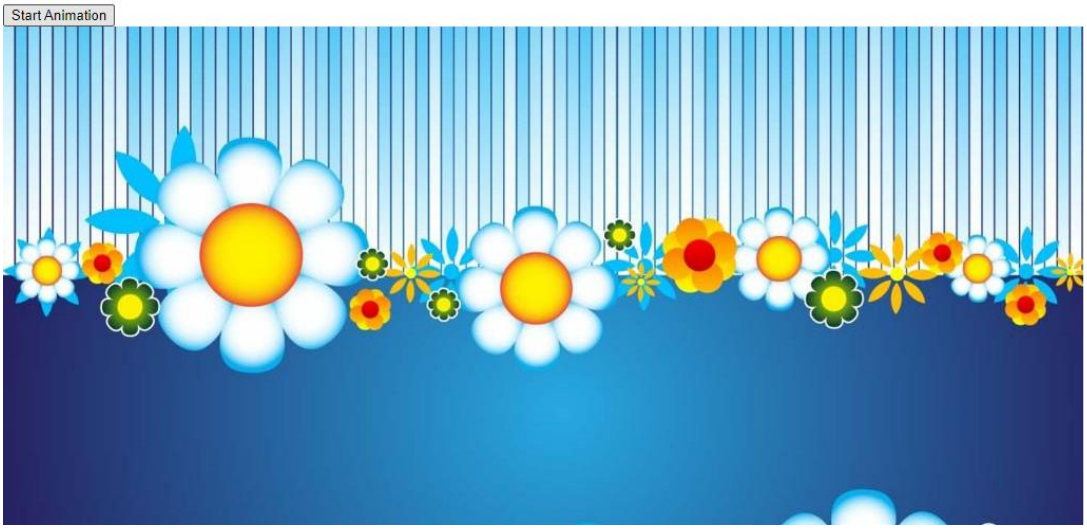
2.A)

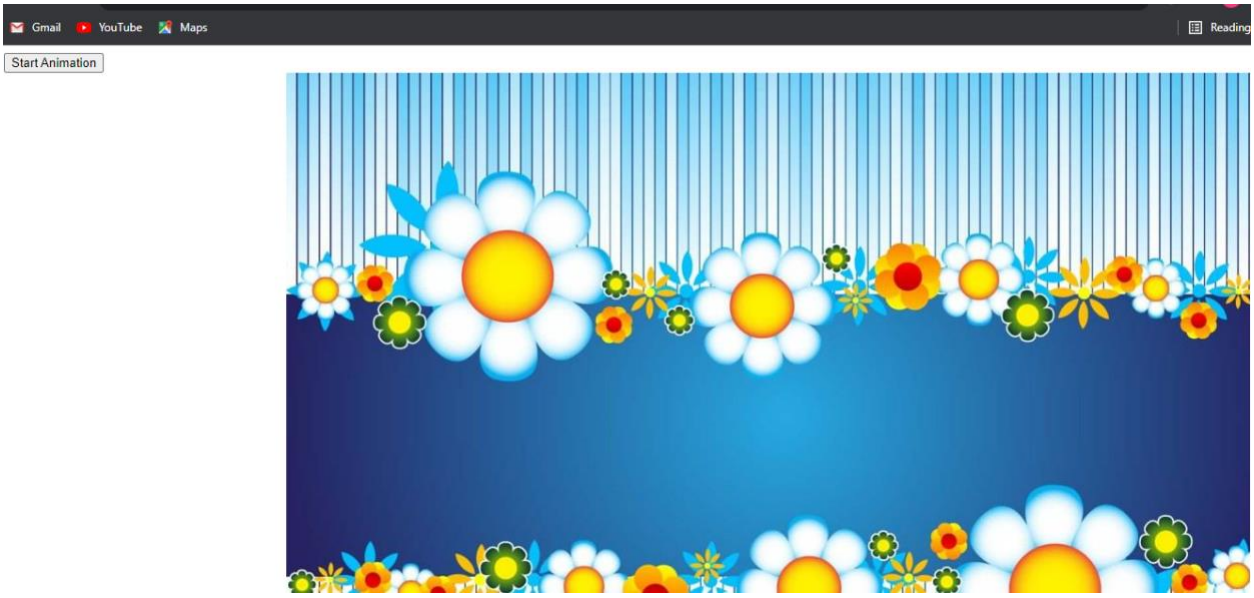
Code:

```
<!DOCTYPE html>
<html>

<head>
    <title></title>
    <script src="jquery-3.6.0.min.js"></script>
    <style>
```

```
        img {
            position: relative;
            /* Required to move element */
        }
    </style>
    <script>
        $(document).ready(
            function () {
                $("button").click(
                    function () {
                        $("img").animate({ left: 300 });
                    }
                );
            }
        );
    </script>
</head>

<body>
    <button>Start Animation</button><br />
    <img src="A.jpg" alt="Mushroom">
</body>

</html>
```

Output:





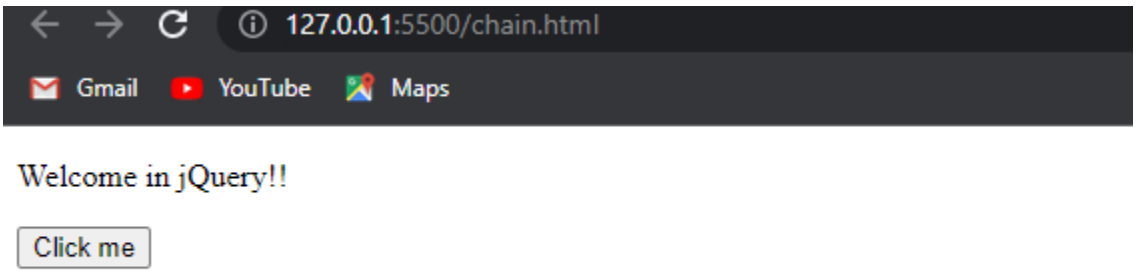
2.B)

Code:

```
<!DOCTYPE html>
<html>

<head>
   <title></title>
```

```
    <script src="jquery-3.6.0.min.js"></script>
    <script>
        $(document).ready(
            function () {
                $("button").click(
                    function () {
                        $("#p1").css("color", "red").slideUp(2000).slideDown(20
00);
                    }
                );
            }
        );
    </script>
</head>

<body>
    <p id="p1">Welcome in jQuery!!</p>
    <button>Click me</button>
</body>

</html>
```

Output:



Slip no 10:

UNIVERSITY OF MUMBAI

T.Y.B.Sc. (INFORMATION TECHNOLOGY) (Semester– V) (Practical) EXAMINATION

SECOND HALF 2018

NEXT GENERATION TECHNOLOGIES

Seat No. : ____ **Max. Marks: 50**

1.	Write a MongoDB query to create Replica of existing database. Also create the backup of existing database. Number of primary server must be one and secondary server must be 3.	**20**
2.	Create a JSON file and persist it in any database.	**20**
3.	Viva	**5**
4.	Journal	**5**

1.

Slip no 6

There was secondary server 2 only you have to create one extra server only all are the same

2.

Slip no 2

Slip no 11

NEXT GENERATION TECHNOLOGIES

Seat No. : ____ **Max. Marks: 50**

1.	(A) Import Restaurant.json into MongoDb and perform the following queries 1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil' 2. Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014- 08-11T00:00:00Z" among many of survey dates. 3. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z" 4. Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 42 and upto 52. 5. Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns. (B) Connect PHP with MongoDB and also insert and delete documents.	**20**
2.	Create a JSON file and persist it in any database.	**20**
3.	Viva	**5**
4.	Journal	**5**

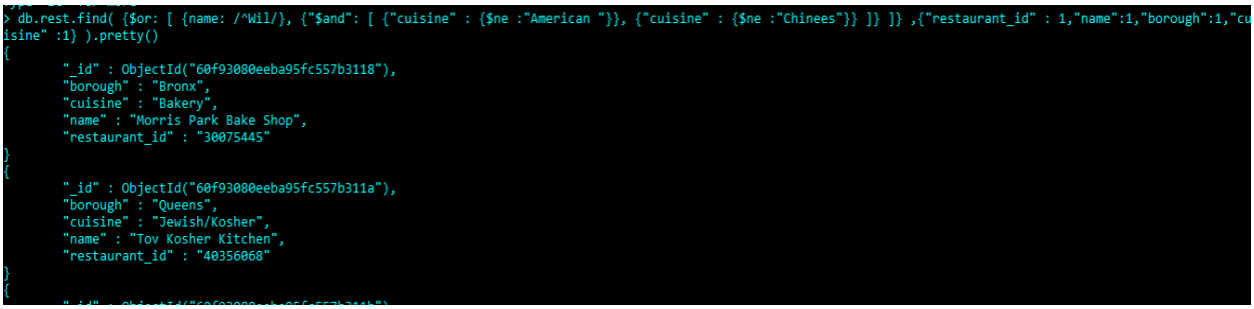
1.A)

1.

Code:

```
db.rest.find( {$or: [ {name: /^Wil/}, {"$and": [ {"cuisine" : {$ne :"American "}}, {"cuisine" : {$ne :"Chinees"}} ]} ]} ,{"restaurant_id" : 1,"name":1,"borough":1,"cuisine" :1} ).pretty()
```

Output:

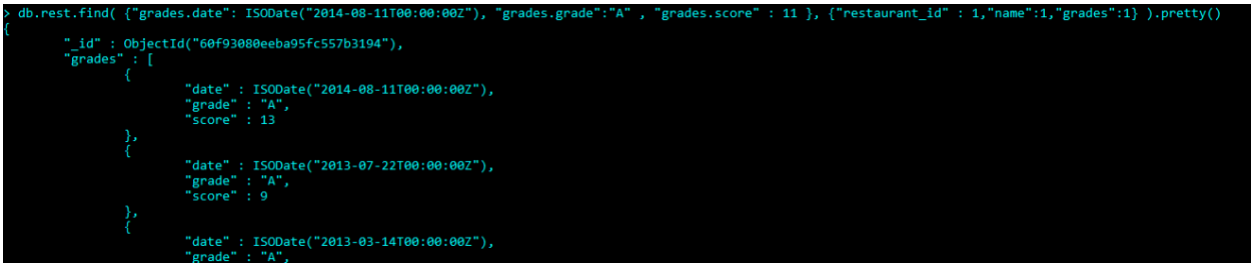


2.

Code:

db.rest.find({"grades.date": ISODate("2014-08-11T00:00:00Z"), "grades.grade":"A" , "grades.score" : 11 }, {"restaurant_id" : 1,"name":1,"grades":1}).pretty()

Output:

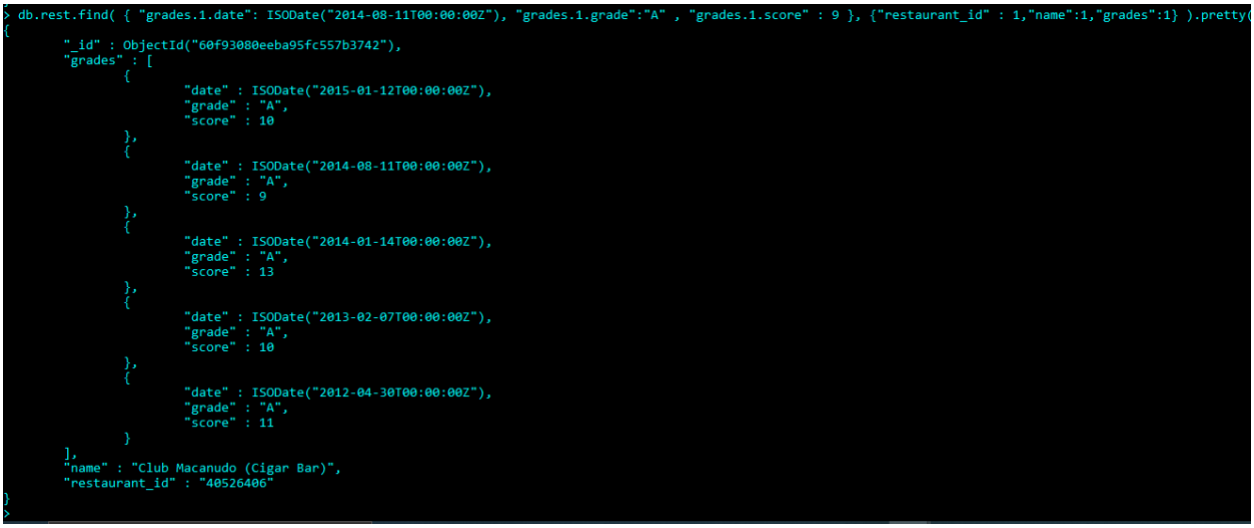


3.

Code:

db.rest.find({ "grades.1.date": ISODate("2014-08-11T00:00:00Z"), "grades.1.grade":"A" , "grades.1.score" : 9 }, {"restaurant_id" : 1,"name":1,"grades":1}).pretty()

Output:



4.

Code;

db.rest.find({ "address.coord.1": {$gt : 42, $lte : 52} }, {"restaurant_id" :1,"name":1,"address":1,"coord":1}).pretty()

output;

```
> db.rest.find( { "address.coord.1": {$gt : 42, $lte : 52} }, {"restaurant_id" : 1,"name":1,"address":1,"coord":1}).pretty()
{
        "_id" : ObjectId("60f93080eeba95fc557b33b9"),
        "address" : {
                "building" : "47",
                "coord" : [
                        -78.877224,
                        42.89546199999999
                ],
                "street" : "Broadway @ Trinity Pl",
                "zipcode" : "10006"
        },
        "name" : "T.G.I. Friday'S",
        "restaurant_id" : "40387990"
}
{
        "_id" : ObjectId("60f93080eeba95fc557b33ea"),
        "address" : {
                "building" : "1",
                "coord" : [
                        -0.7119979,
                        51.6514664
```

5.

Code:

db.rest.find().sort({"name":1}).pretty()

Output:

```
> db.rest.find().sort({"name":1}).pretty()
{
        "_id" : ObjectId("60f93080eeba95fc557b3daa"),
        "address" : {
                "building" : "129",
                "coord" : [
                        -73.962943,
                        40.685007
                ],
                "street" : "Gates Avenue",
                "zipcode" : "11238"
        },
        "borough" : "Brooklyn",
        "cuisine" : "Italian",
        "grades" : [
```

1.B)

Code:

Slip no 7

2.

Code

Slip no 2

Slip no 12

UNIVERSITY OF MUMBAI

T.Y.B.Sc. (INFORMATION TECHNOLOGY) (Semester– V) (Practical) EXAMINATION

SECOND HALF 2018

NEXT GENERATION TECHNOLOGIES

Seat No. : ____ **Max. Marks: 50**

1.	Create a Collection Employee with the following Fields (Eid,Ename,Sal,City,hobbies) where hobbies is an array perform the Following Queries based on the collection. 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. D. Perform backup and restore on the above collection.	**20**
2.	(A) Create a JSON file and parse it. (B) Write a jQuery to create fade-in and fade-out effect.	**20**
3.	Viva	**5**
4.	Journal	**5**

```
> use TY
switched to db TY
> db.createCollection(""Exam")
2021-10-06T14:54:47.717+0530 E QUERY    [thread1] Sy
> db.createCollection(""Exam");
2021-10-06T14:54:54.557+0530 E QUERY    [thread1] Sy
> db.createCollection("Exam");
{ "ok" : 1 }
>
```

```
> db.Exam.insert({"Eid":1,"Ename":"titu","Sal":1000,"City":"bhiwandi","hobbies":"dance"})
WriteResult({ "nInserted" : 1 })
> db.Exam.insert({"Eid":2,"Ename":"Jaydeep","Sal":2000,"City":"mulund","hobbies":"games"})
WriteResult({ "nInserted" : 1 })
> db.Exam.insert({"Eid":3,"Ename":"Datta","Sal":3000,"City":"bhiwandi","hobbies":"kho-kho"})
WriteResult({ "nInserted" : 1 })
> db.Exam.insert({"Eid":4,"Ename":"Mahi","Sal":4000,"City":"bhiwandi","hobbies":"chess"})
WriteResult({ "nInserted" : 1 })
> db.Exam.insert({"Eid":5,"Ename":"Rohit","Sal":5000,"City":"worli","hobbies":"games"})
WriteResult({ "nInserted" : 1 })
>
```

```
> db.Exam.find()
{ "_id" : ObjectId("615d6c65bdaa8719970e31d2"), "Eid" : 1, "Ename" : "titu", "Sal" : 1000, "City" : "bhiwandi", "hobbies" : "dance" }
{ "_id" : ObjectId("615d6c96bdaa8719970e31d3"), "Eid" : 2, "Ename" : "Jaydeep", "Sal" : 2000, "City" : "mulund", "hobbies" : "games" }
{ "_id" : ObjectId("615d6cdebdaa8719970e31d4"), "Eid" : 3, "Ename" : "Datta", "Sal" : 3000, "City" : "bhiwandi", "hobbies" : "kho-kho" }
{ "_id" : ObjectId("615d6d2cbdaa8719970e31d5"), "Eid" : 4, "Ename" : "Mahi", "Sal" : 4000, "City" : "bhiwandi", "hobbies" : "chess" }
{ "_id" : ObjectId("615d6d61bdaa8719970e31d6"), "Eid" : 5, "Ename" : "Rohit", "Sal" : 5000, "City" : "worli", "hobbies" : "games" }
>
```

Queries

1.

A.

Sum:

Code:

db.Exam.aggregate([{$group: {_id : "$Ename",no: {$sum :1}}}])

Output:

```
> db.Exam.aggregate([{$group: {_id : "$Ename",no: {$sum :1}}}])
{ "_id" : "Rohit", "no" : 1 }
{ "_id" : "Datta", "no" : 1 }
{ "_id" : "Mahi", "no" : 1 }
{ "_id" : "Jaydeep", "no" : 1 }
{ "_id" : "titu", "no" : 1 }
>
```

it will count the name

B.

Avg

Code:

db.Exam.aggregate([{$group: {_id : "$Eid",no: {$avg :1}}}])

Output:

```
> db.Exam.aggregate([{$group: {_id : "$Eid",no: {$avg :"$Sal"}}}])
{ "_id" : 3, "no" : 3000 }
{ "_id" : 2, "no" : 2000 }
{ "_id" : 5, "no" : 5000 }
{ "_id" : 1, "no" : 1000 }
{ "_id" : 4, "no" : 4000 }
>
```

Min

Code:

db.Exam.aggregate([{$group: {_id : "$Eid",no: {$min :"$Sal"}}}])

Output:

```
> db.Exam.aggregate([{$group: {_id : "$Eid",no: {$min :"$Sal"}}}])
{ "_id" : 3, "no" : 3000 }
{ "_id" : 2, "no" : 2000 }
{ "_id" : 5, "no" : 5000 }
{ "_id" : 1, "no" : 1000 }
{ "_id" : 4, "no" : 4000 }
>
```

Max:

Code:

db.Exam.aggregate([{$group: {_id : "$Eid",no: {$max :"$Sal"}}}])

Output:

```
> db.Exam.aggregate([{$group: {_id : "$Eid",no: {$max :"$Sal"}}}])
{ "_id" : 3, "no" : 3000 }
{ "_id" : 2, "no" : 2000 }
{ "_id" : 5, "no" : 5000 }
{ "_id" : 1, "no" : 1000 }
{ "_id" : 4, "no" : 4000 }
>
```

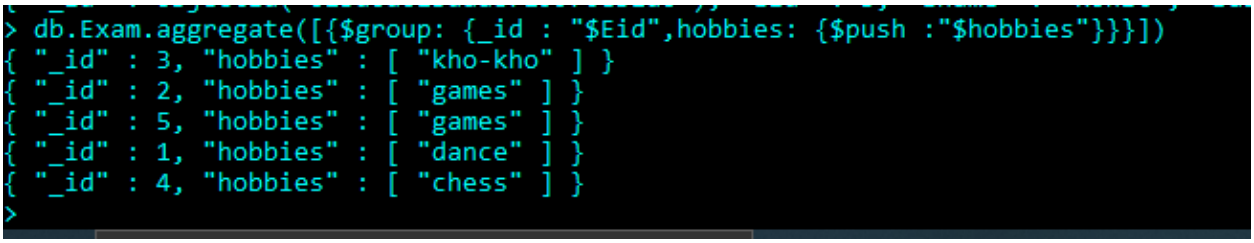
B.

Push

Code:

db.Exam.aggregate([{$group: {_id : "$Eid",hobbies: {$push :"$hobbies"}}}])

Output:

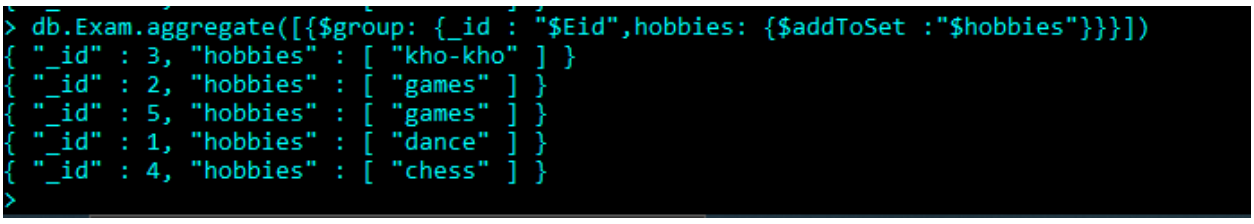


addToSet

Code:

db.Exam.aggregate([{$group: {_id : "$Eid",hobbies: {$addToSet :"$hobbies"}}}])

Output:



C.

First expression

Code;

db.Exam.aggregate([{$group: {_id : "$Eid",first_hobbies: {$first :"$hobbies"}}}])

Output:

```
> db.Exam.aggregate([{$group: {_id : "$Eid",first_hobbies: {$first :"$hobbies"}}}])
{ "_id" : 3, "first_hobbies" : "kho-kho" }
{ "_id" : 2, "first_hobbies" : "games" }
{ "_id" : 5, "first_hobbies" : "games" }
{ "_id" : 1, "first_hobbies" : "dance" }
{ "_id" : 4, "first_hobbies" : "chess" }
>
```

Last expression:

Code:

db.Exam.aggregate([{$group: {_id : "$Eid",last_hobbies: {$last :"$hobbies"}}}])

Output:

```
> db.Exam.aggregate([{$group: {_id : "$Eid",last_hobbies: {$last :"$hobbies"}}}])
{ "_id" : 3, "last_hobbies" : "kho-kho" }
{ "_id" : 2, "last_hobbies" : "games" }
{ "_id" : 5, "last_hobbies" : "games" }
{ "_id" : 1, "last_hobbies" : "dance" }
{ "_id" : 4, "last_hobbies" : "chess" }
>
```

D.

Slip no 5 1B same of the backup of data

2.A

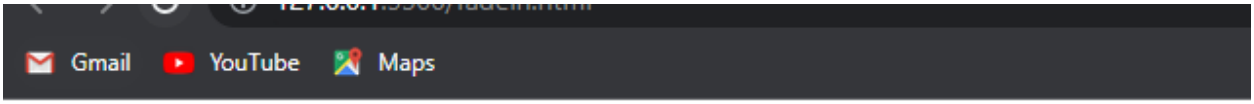
Slip no 4

2.B

Code:

```
<!DOCTYPE html>
<html>

<head>
    <title></title>
    <script src="jquery-3.6.0.min.js"></script>
    <script>
        $(document).ready(
            function () {
                $(".btn1").click(
                    function () {
                        $("p").fadeOut()
                    }
                );
                $(".btn2").click(
                    function () {
                        $("p").fadeIn();
                    }
                );
            }
        );
    </script>
</head>

<body>
    <p>Mulund College Of Commerce</p>
    <button class="btn1">Fade out</button>
    <button class="btn2">Fade in</button>
</body>

</html>
```

Output:



Mulund College Of Commerce

Fade out Fade in

Fade out Fade in

Slip no 13

NEXT GENERATION TECHNOLOGIES

Seat No. : ____ **Max. Marks: 50**

1.	(A) Import Restaurant.json into MongoDb and perform the following queries 1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil' 2. Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014- 08-11T00:00:00Z" among many of survey dates. 3. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate "2014-08-11T00:00:00Z" 4. Write a MongoDB query to find the restaurant Id, name, address and geographical location for those restaurants where 2nd element of coord array contains a value which is more than 42 and upto 52. 5. Write a MongoDB query to arrange the name of the restaurants in ascending order along with all the columns. (B) Connect Java with MongoDB and also insert and delete documents.	**20**
2.	A. Write a jQuery to remove the parent element of an HTML element from the page. B. Write a jQuery to add and remove CSS classes from the HTML elements.	**20**
3.	Viva	**5**

1A)

1,2,3,4,5

Sip no 11

1.B)

Slip no 2

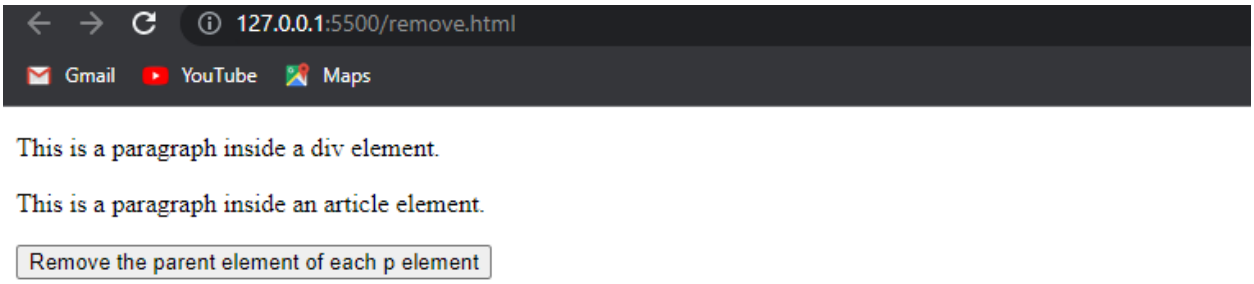
2.A)

Code:

```
<!DOCTYPE html>
<html>
```

```
<head>
   <title></title>
   <script src="jquery-3.6.0.min.js"></script>
   <script>
      $(document).ready(
         function () {
            $("button").click(
               function () {
                  $("p").unwrap();
               }
            );
         }
      );
   </script>
   <style>
      div {
         background-color: yellow;
      }
      article {
         background-color: pink;
      }
   </style>
</head>
<body>
   <div>
      <p>This is a paragraph inside a div element.</p>
   </div>
   <article>
      <p>This is a paragraph inside an article element.</p>
   </article>
   <button>Remove the parent element of each p element</button>
</body>

</html>
```

Output:



2.B)

Code:

Slip no 6

Slip no 14:

UNIVERSITY OF MUMBAI

T.Y.B.Sc.(INFORMATION TECHNOLOGY) (Semester– V) (Practical) EXAMINATION

SECOND HALF 2018

NEXT GENERATION TECHNOLOGIES

Seat No. : ____ **Max. Marks: 50**

1.	Connect PHP with MongoDB and also insert, retrieve, update and delete documents.	**20**
2.	A. Write a jQuery to get and set text contents of the elements. B. Write a jQuery to set the duration in slide toggle effect.	**20**
3.	Viva	**5**
4.	Journal	**5**

1.

Code:

Slip no 4

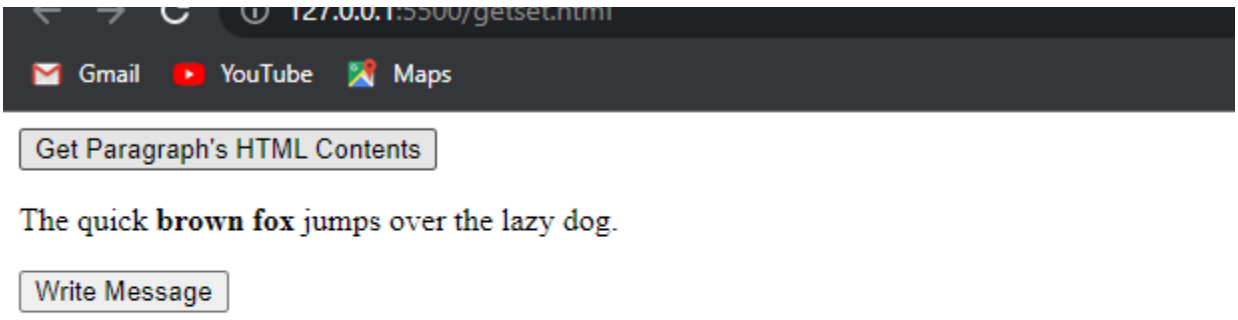
2.A)

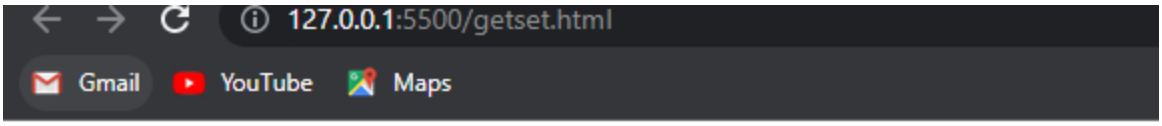
Code:

```
<!DOCTYPE html>
<html>

<head>
   <title></title>
   <script src="jquery-3.6.0.min.js"></script>
   <script>
      $(document).ready(
         function () {
            $(".b1").click(
               function () {
                  var str = $("p").html();
```

```
                alert(str);
            }
        );
        $(".b2").click(
            function () {
                $("body").html("<p>Hello World!</p>");
            }
        );
    }
  );
  </script>
</head>

<body>
  <button class="b1">Get Paragraph's HTML Contents</button>
  <p>The quick <b>brown fox</b> jumps over the lazy dog.</p>
  <button class="b2">Write Message</button>
</body>

</html>
```

Output:





Hello World!

2.B)

Code:

Slip no 6

Slip no 15

SECOND HALF 2018

NEXT GENERATION TECHNOLOGIES

Seat No. : ____ **Max. Marks: 50**

1.	(A) Import Restaurant.json into MongoDb and perform the following queries 1. Write a MongoDB query to arrange the name of the restaurants in descending along with all the columns 2. Write a MongoDB query to arranged the name of the cuisine in ascending order and for that same cuisine borough should be in descending order 3. Write a MongoDB query to know whether all the addresses contains the street or not. 4. Write a MongoDB query which will select all documents in the restaurants collection where the coord field value is double 5. Write a MongoDB query which will select the restaurant Id, name and grades for those restaurants which returns 0 as a remainder after dividing the score by 7. (B) Write a MongoDB query to create Replica of existing database.	**20**
2.	A. Write a jQuery to animate multiple CSS properties. B. Create a JSON file and parse it.	**20**
3.	Viva	**5**
4.	Journal	**5**

1.A)

1.

Code:

db.rest.find().sort({"name":-1}).pretty()

Output:

2.

Code:

db.rest.find().sort({"cuisine":1,"borough" : -1,}).pretty()

Output:

```
> db.rest.find().sort( {"cuisine":1,"borough" : -1,} ).pretty()
{
        "_id" : ObjectId("60f93080eeba95fc557b3803"),
        "address" : {
                "building" : "1345",
                "coord" : [
                        -73.959249,
                        40.768076
                ],
                "street" : "2 Avenue",
                "zipcode" : "10021"
        },
        "borough" : "Manhattan",
        "cuisine" : "Afghan",
        "grades" : [
                {
                        "date" : ISODate("2014-10-07T00:00:00Z"),
                        "grade" : "A",
                        "score" : 9
                },
```

3.

Code:

db.rest.find({"address.street" : { $exists : true } }).pretty()

Output:

```
> db.rest.find( {"address.street" : { $exists : true } } ).pretty()
{
        "_id" : ObjectId("60f93080eeba95fc557b3117"),
        "address" : {
                "building" : "2780",
                "coord" : [
                        -73.98241999999999,
                        40.579505
                ],
                "street" : "Stillwell Avenue",
                "zipcode" : "11224"
        },
        "borough" : "Brooklyn",
        "cuisine" : "American ",
        "grades" : [
                {
                        "date" : ISODate("2014-06-10T00:00:00Z"),
                        "grade" : "A",
                        "score" : 5
                },
```

4.

Code:

db.rest.find({"address.coord" : {$type : 1} }).pretty()

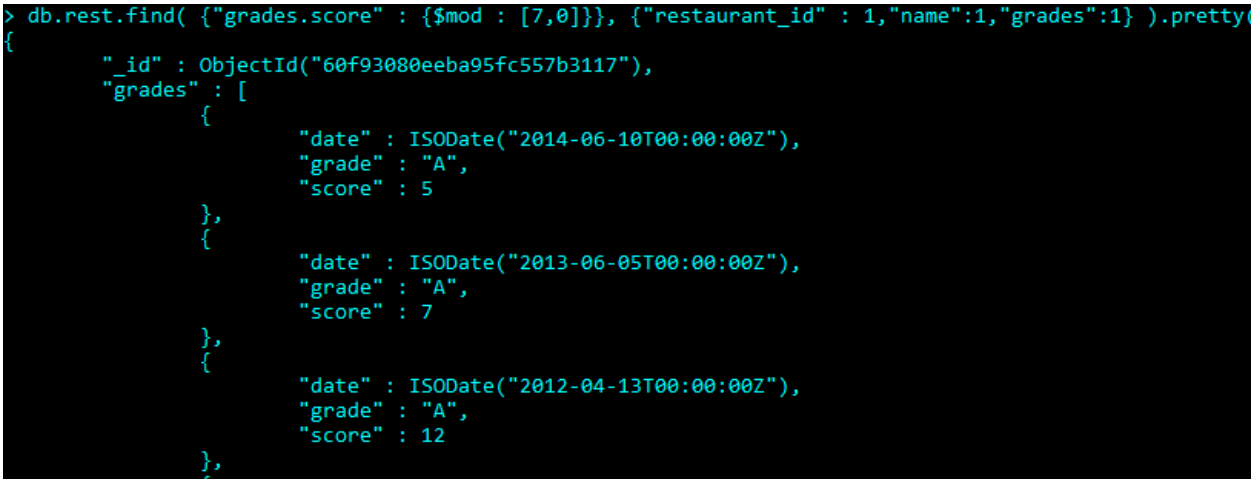
Output:

5.

Code:

db.rest.find({"grades.score" : {$mod : [7,0]}}, {"restaurant_id" : 1,"name":1,"grades":1}).pretty()

Output:



1.B)

Code:

Slip no 6

2.A)

Code:

Slip no 1

2.B)

Code:

Slip no 4

Slip no 16

UNIVERSITY OF MUMBAI

T.Y.B.Sc.(INFORMATION TECHNOLOGY) (Semester– V) (Practical) EXAMINATION

SECOND HALF 2018

NEXT GENERATION TECHNOLOGIES

Seat No. : ____ **Max. Marks: 50**

1.	Connect Java with MongoDB and also insert, retrieve, update and delete documents.	**20**
2.	A. Write a Jquery to get and set text contents of the elements. B. Write a Jquery to insert HTML elements at the beginning and end of the elements	**20**
3.	Viva	**5**
4.	Journal	**5**

1.

Code:

Slip no 2

2.A)

Code:

Slip no 14

2.B)

Code:

```
<!DOCTYPE html>
<html>

<head>
   <title></title>
```

```
    <script src="jquery-3.6.0.min.js"></script>
    <script>
        $(document).ready(
            function () {
                $(".b1").click(
                    function () {
                        $("p").prepend("<strong>Note:</strong> ");
                    }
                );
                $(".b2").click(
                    function () {
                        $("#container").append("This is demo text.");
                    }
                );
            }
        );
    </script>
</head>

<body>
    <button class="b1">Insert Text at Begin</button>
    <p>welcome in Jquery</p>
    <button class="b2">Insert Text at End</button>
    <div id="container">
        <p>Mulund College of Commerce</p>
    </div>
</body>

</html>
```

Output:

Insert Text at Begin

welcome in Jquery

Insert Text at End

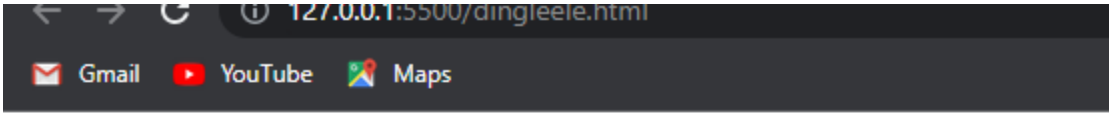
Mulund College of Commerce

Insert Text at Begin

Note: welcome in Jquery

Insert Text at End

Note: Mulund College of Commerce



Insert Text at Begin

Note: welcome in Jquery

Insert Text at End

Note: Mulund College of Commerce

This is demo text.