Assignment No C1.

Aim:- Write a program to implement MongoDB database connectivity with Java. Implement Database navigation operations (add, delete, edit, etc) using ODBC / JDBC.

Objective:- To learn and implement MongoDB connection with Java.

Outcome:- Student should be able to create their own full functional projects.

Theory:-

In most web applications / commercial applications 2 tier / 3 tier architecture are used. In such applications users can interact with underlying database using front end lang like Java, etc.

A simple user friendly GUI will be provided by such front end tools so that user will remain completely unaware about underlying actual database operations.

Software Required & steps:-

Eclipse
JDK 1.6
MongoDB
MongoDB - Java - Driver.

In Eclipse:-

1) file → new → java project → give proj name → ok
2) In project explorer window → right click on proj name → new - class → give class name → ok.
3) In project - Explorer window - right click on project - name - Build path - Configure build path - Libraries - Add External Jar - MongoDB - Java - Driver.
4) Start mongo server before running the program.

Next,
Import packages, create ~~collection~~ connection & create database & collection.

```
DB db = mongo.getDB("Database name");
DBCollection coll = db.getCollection("Collection Name");
```

I] Insert Document -

```
BasicDBObject d1 = new BasicDBObject("rno","1").
                  append("ABC").append("age","17")

coll.insert (d1);
```

II) Display →

```
DBCursor cursor = coll.find();
while (cursor.hasNext())
{
    System.out.println(cursor.next());
}
```

III]. Update Doument :-

```
Basic DBObject query = new Basic DBObject;
 query . put ("name", "ABC");
  Basic DBObject newDoument = new Basic DBObject();
  newDoument.put ("name", "XYZ");
   BasicDBObject updateObj = new BasicDBObject ();
   updateObj . put ("$ set", newDoument);
   Coll. update (query, updateObj);
```

IV]. Remove :-

```
BasicDBObject searchQuery = new BasicDBObject ();
searchQuery. put ("name", "ABC");
 coll. remove (searchQuery);
```

Conclusion:-
Hence, we studied MongoDB & Java connectivity & performed CRUD operations.

Assignment No C2.

#Aim: Implement a MySQL database connectivity with Java Implement Database Navigation operations (add, delete, edit) using ODBC /JDBC.

objective:- To learn & implement MySQL connection with Java.

Outcome:- Students should be able to create their own functional projects.

Theory:-

In most web applications / commercial applications 2 tier /3 tier architecture are used. In such application users can interact with underlying database using front end like Java, etc.

A simple user friendly GUI will be provided by such front end tools so that user will remain completely unaware about underlying actual database operations.

Steps to be perfomed at MySQL -

```
Create database expdb;
use expdb;
create table exptable (ID int not null auto_increment
          primary key, name varchar(50));
// insert some values
```

In Java program,

1) Import classes.
2) Import drivers.

```
Class. forName ("com. mysql. jdbc. Driver");
```

I Create connection:-

```
String connUrl = "jdbc:mysql://localhost:3306/expdb";
String dbuser = "vrushil";
String dbPwd = "vrushil@123";
connection conn;
conn= DriverManager. getConnection (connUrl, dbUser, dbPwd)
```

II] ~~Create~~ Insert a record -

```
Statement stmt = conn. createStatement;
stmt. executeUpdate ("Insert into exptable (Name)
                        values (\"Vrushil\")");
```

III] Display all:-

```
ResultSet rs;
rs = stmt. executeQuery ("select * from exptable");
System.out. println ("ID \t Name");
System.out. println ("=======");
while (rs.next())
{
    System. out. println (rs.getInt("id") + "\t" +
    rs. getString ("name");
        System. out. println();
}
```

Closing the connection :-

```
conn. close ();
```

Conclusion.

Hence, we have successfully connected MySQL with Java jdbc ~~to~~ and executed CRUD operations.