Page No.
Date

Assignment - C1

* Date of completion: 23/11/2020
* Date of Submission: 27/11/2020
* Title: MongoDB - Java Connectivity
* Problem Statement:
  Write a program to implement MongoDB database connectivity with PHP / Python / Java Implement Database navigation operations (add, delete, update, etc) using ODBC / JDBC
* Objectives:
  1) To implement database connectivity for MongoDB with Java
  2) To perform navigation operations on the database using JDBC / ODBC.
* Outcomes:
  Students will be able to:
  i) Implement database connectivity for MongoDB
  ii) Perform CRUD operations on the database.
* S/W And H/W Requirements:
  MongoDB 4.4.1 Server, IntelliJ, Windows 10, i5 processor

Teacher's Sign.: _____

Page No.
Date

* Theory:

MONGODB : Java Driver

- MongoDB provides connectivity for Java client applications using Java driver.
- Using the Java Driver is simple, just include the driver jar mongo.jar in your class path.
- The JDBC API can do the following things:
  - Establish a connection with a database or access any tabular data source.
  - Send various MongoDB statements
  - Retrieve and process results received from the database.

Here for MongoDB 4.4 have used 3 jar files:
1) mongodb-driver-core-4.1.1
2) mongodb-driver-sync-4.1.1
3) bson-4.1.1

- Connect to database
  1. Create a MongoClient:
     MongoClient mongo = new MongoClient("localhost", 27017)
  2. MongoDatabase database =
     mongo.getDatabase("db-name")

- Create a collection:
  Syntax:
  database.createCollection("collection_name");

Teacher's Sign.:

Page No.
Date

- Getting / selecting a collection:

```
MongoClient<Document> collection = database.getCollection(
                                    "collection_name");
```

- Insert a Document:

```
Document doc = new Document("key1", "value1")
                .append("key2", "value2");

collection.insertOne(doc);
```

- Retrieving all documents:

```
FindIterable<Document> iterDoc = collection.find()
Iterator it = iterDoc.iterator();
while(it.hasNext()){
    System.out.println(it.next());
        it++;
}
```

- Update Document:

```
collection.updateOne(Filters.eq("title", 1),
                     update.set("key", "value"));
```

- Delete a document:

```
collection.deleteOne(Filters.eq("title", "MongoDB"));
```

- Dropping a collection:

```
collection.drop();
```

Teacher's Sign.: _____

Page No.

Page No.
Date

* Test Cases:

	Input	Expected Output	Result
1)	Collection name: Movie	Movie collection created successfully	Success
2)	Movie id: 101 Genre: comedy Rating: 9.5	Document inserted successfully	Success.

* Conclusion:

1) Thus, we implemented MongoDB database connectivity with Java
2) We implemented CRUD operations.

Teacher's Sign.: _____