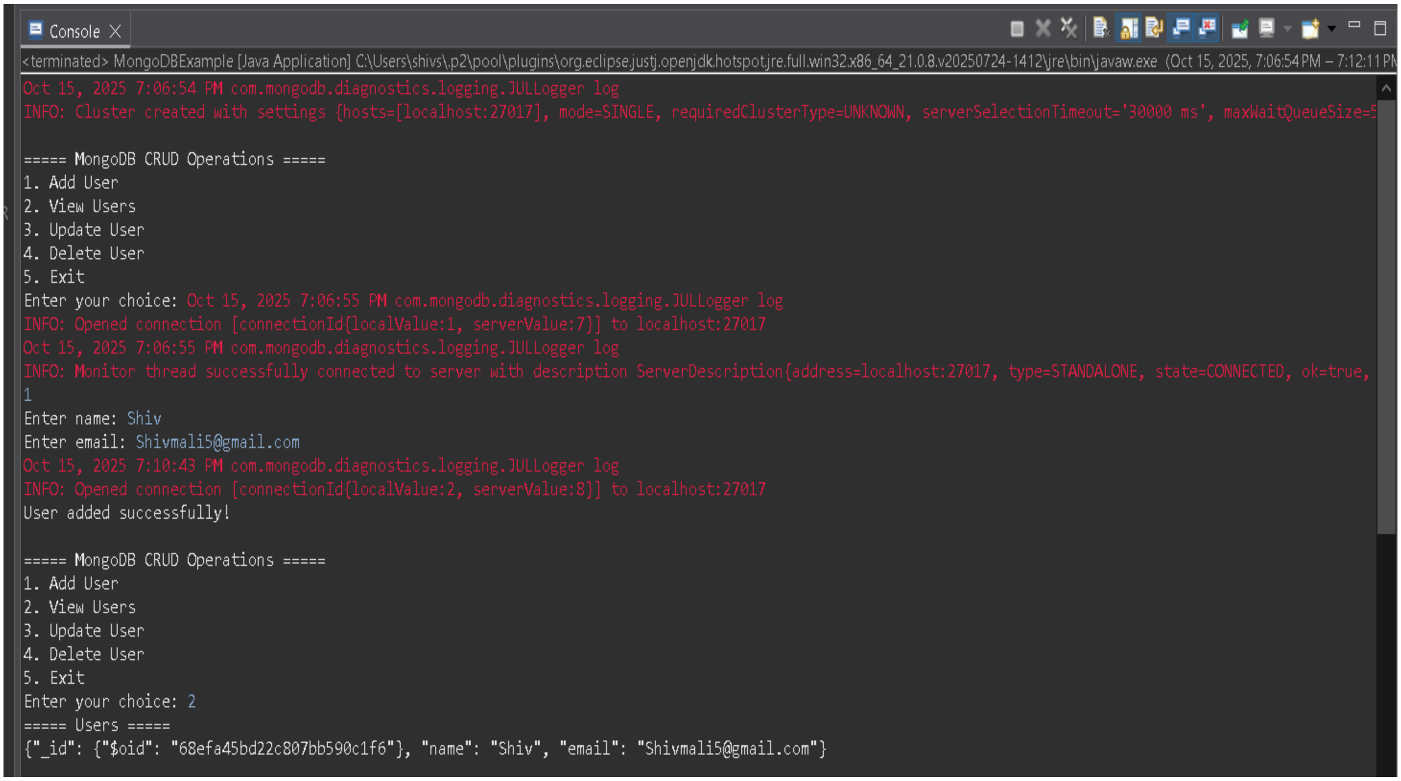
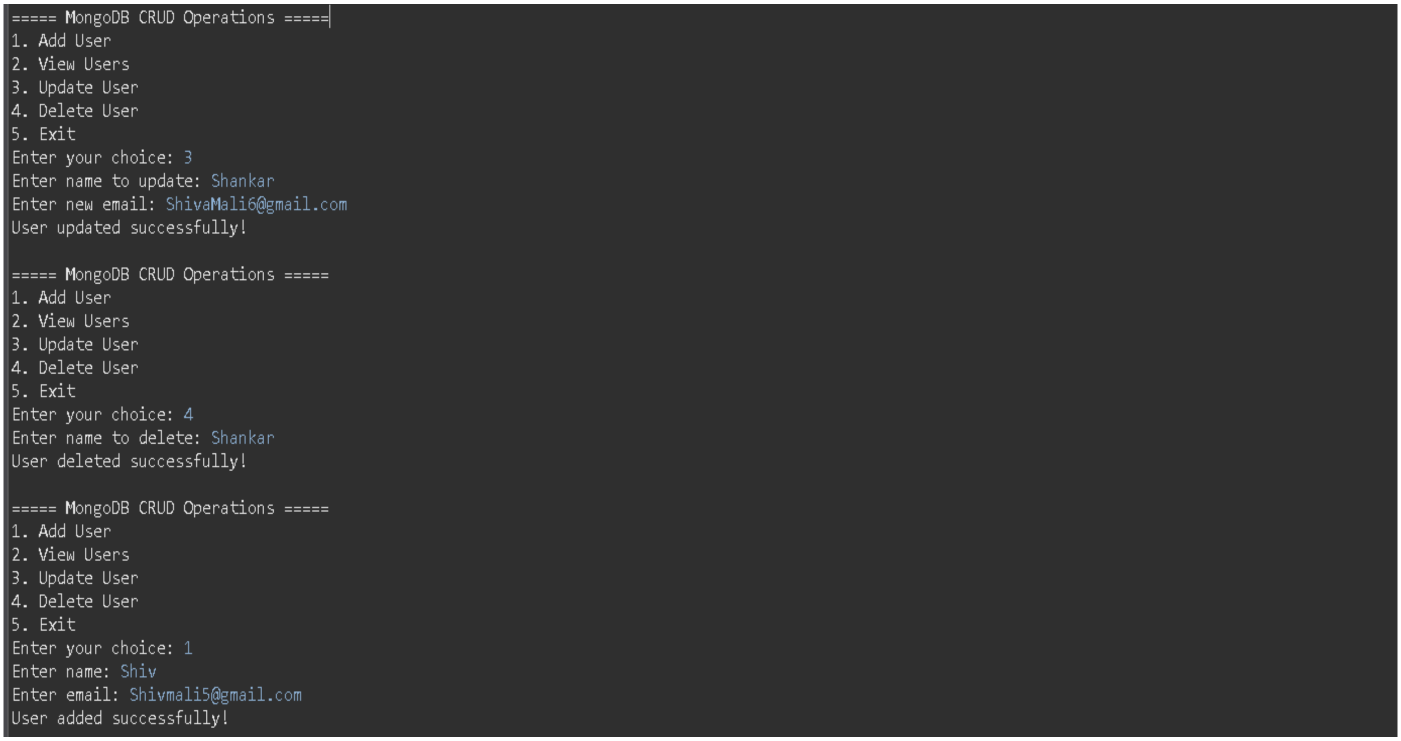
```java
package mongoCon;

import java.util.Scanner;
import org.bson.Document;
import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.model.Filters;
import static com.mongodb.client.model.Updates.set;

public class MongoDBExample {
    public static void main(String[] args) {
        // Connect to MongoDB server
        MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017");

        // Connect to database
        MongoDatabase database = mongoClient.getDatabase("mydb");

        // Connect to collection
        MongoCollection<Document> collection = database.getCollection("users");

        Scanner sc = new Scanner(System.in);
        int choice;

        do {
            System.out.println("\n===== MongoDB CRUD Operations =====");
            System.out.println("1. Add User");
            System.out.println("2. View Users");
            System.out.println("3. Update User");
            System.out.println("4. Delete User");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();
            sc.nextLine(); // clear buffer

            switch (choice) {
                case 1:
                    System.out.print("Enter name: ");
                    String name = sc.nextLine();
                    System.out.print("Enter email: ");
                    String email = sc.nextLine();
                    Document doc = new Document("name", name)
                            .append("email", email);
                    collection.insertOne(doc);
                    System.out.println("User added successfully!");
                    break;
                case
2:
                    System.out.println("===== Users =====");
                    for (Document d : collection.find()) {
                        System.out.println(d.toJson());
                    }
                    break;
```

```
                case
3:
                    System.out.print("Enter name to update: ");
                    String oldName = sc.nextLine();
                    System.out.print("Enter new email: ");
                    String newEmail = sc.nextLine();
                    collection.updateOne(Filters.eq("name", oldName), set("email", newEmail));
                    System.out.println("User updated successfully!");
                    break;
                case
4:
                    System.out.print("Enter name to delete: ");
                    String delName = sc.nextLine().trim();
                    collection.deleteOne(Filters.eq("name", delName));
                    System.out.println("User deleted successfully!");
                    break;
                case
5:
                    System.out.println("Exiting...");
                    break;

default:
                    System.out.println("Invalid choice!");
            }
        } while (choice != 5);

        mongoClient.close();
        sc.close();
    }
}

```

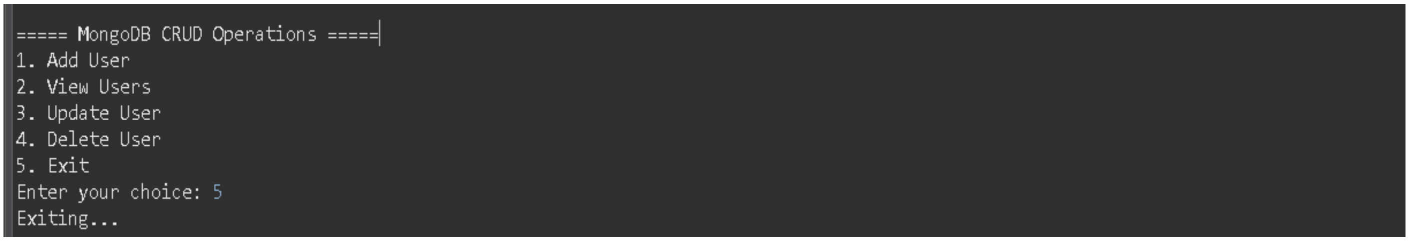
## Output:



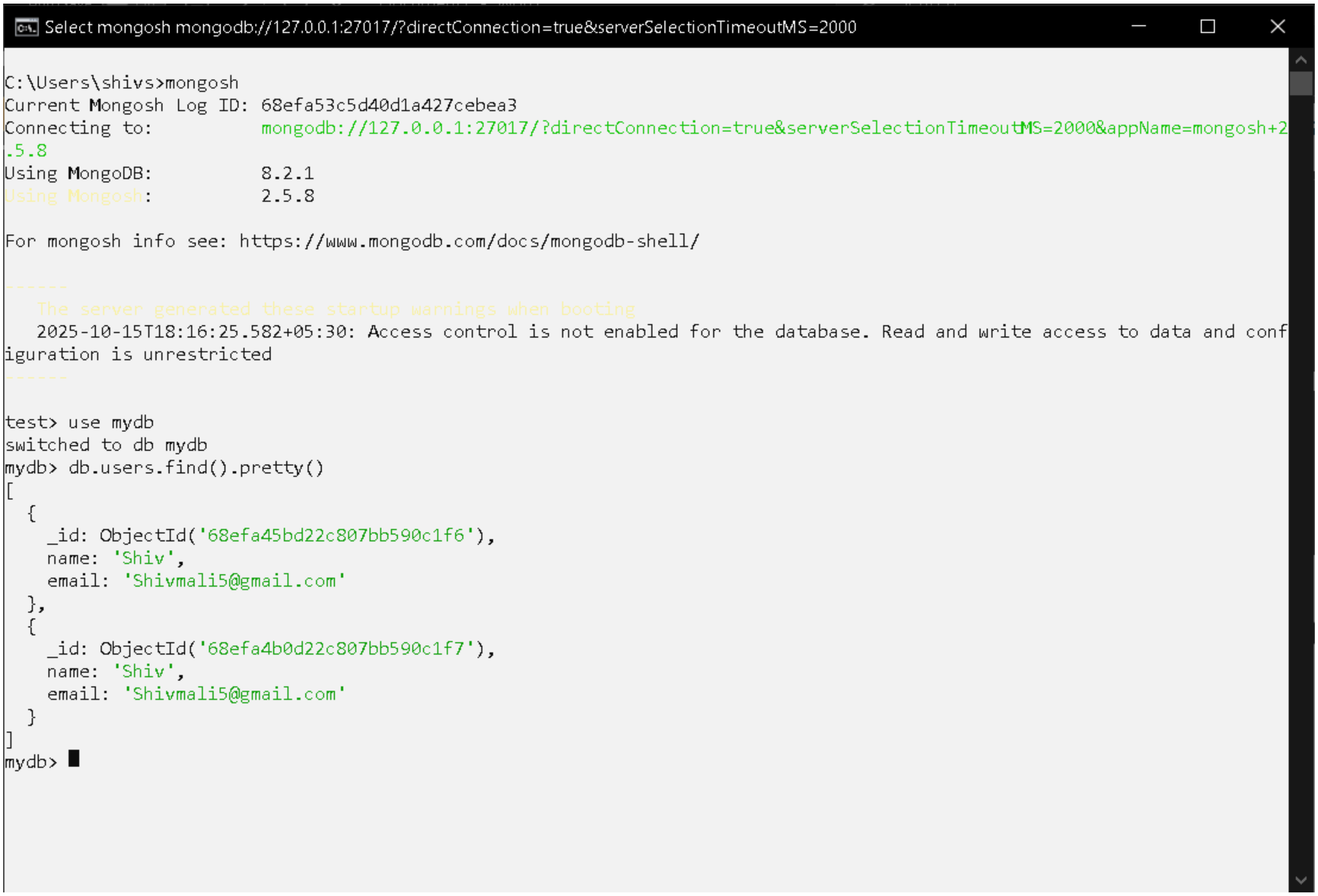
```
<terminated> MongoDBExample [Java Application] C:\Users\shivs\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.8.v20250724-1412\jre\bin\javaw.exe (Oct 15, 2025, 7:06:54 PM – 7:12:11 PI
Oct 15, 2025 7:06:54 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaitQueueSize=5

===== MongoDB CRUD Operations =====
1. Add User
2. View Users
3. Update User
4. Delete User
5. Exit
Enter your choice: Oct 15, 2025 7:06:55 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:7}] to localhost:27017
Oct 15, 2025 7:06:55 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, state=CONNECTED, ok=true,
1
Enter name: Shiv
Enter email: Shivmali5@gmail.com
Oct 15, 2025 7:10:43 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:8}] to localhost:27017
User added successfully!

===== MongoDB CRUD Operations =====
1. Add User
2. View Users
3. Update User
4. Delete User
5. Exit
Enter your choice: 2
===== Users =====
{"_id": {"$oid": "68efa45bd22c807bb590c1f6"}, "name": "Shiv", "email": "Shivmali5@gmail.com"}
```



```
===== MongoDB CRUD Operations =====
1. Add User
2. View Users
3. Update User
4. Delete User
5. Exit
Enter your choice: 3
Enter name to update: Shankar
Enter new email: ShivaMali6@gmail.com
User updated successfully!

===== MongoDB CRUD Operations =====
1. Add User
2. View Users
3. Update User
4. Delete User
5. Exit
Enter your choice: 4
Enter name to delete: Shankar
User deleted successfully!

===== MongoDB CRUD Operations =====
1. Add User
2. View Users
3. Update User
4. Delete User
5. Exit
Enter your choice: 1
Enter name: Shiv
Enter email: Shivmali5@gmail.com
User added successfully!
```



```
===== MongoDB CRUD Operations =====
1. Add User
2. View Users
3. Update User
4. Delete User
5. Exit
Enter your choice: 5
Exiting...
```


Select mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
C:\Users\shivs>mongosh
Current Mongosh Log ID: 68efa53c5d40d1a427cebea3
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2
.5.8
Using MongoDB:          8.2.1
Using Mongosh:          2.5.8

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-10-15T18:16:25.582+05:30: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
------

test> use mydb
switched to db mydb
mydb> db.users.find().pretty()
[
  {
    _id: ObjectId('68efa45bd22c807bb590c1f6'),
    name: 'Shiv',
    email: 'Shivmali5@gmail.com'
  },
  {
    _id: ObjectId('68efa4b0d22c807bb590c1f7'),
    name: 'Shiv',
    email: 'Shivmali5@gmail.com'
  }
]
mydb> 
```