**Main.java**

package com.example.mongodbapp;  
  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import com.mongodb.client.MongoClient;  
import com.mongodb.client.MongoCollection;  
import com.mongodb.client.MongoDatabase;  
import org.bson.Document;  
  
import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;  
  
public class MainActivity extends AppCompatActivity {  
  
 private EditText nameInput, emailInput, ageInput;  
 private Button insertButton, updateButton, deleteButton, fetchButton;  
  
 // ExecutorService for running database operations in the background  
 private final ExecutorService executorService = Executors.*newSingleThreadExecutor*();  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
  
 // Initialize UI elements  
 nameInput = findViewById(R.id.nameInput);  
 emailInput = findViewById(R.id.emailInput);  
 ageInput = findViewById(R.id.ageInput);  
 insertButton = findViewById(R.id.insertButton);  
 updateButton = findViewById(R.id.updateButton);  
 deleteButton = findViewById(R.id.deleteButton);  
 fetchButton = findViewById(R.id.fetchButton);  
  
 // Set button listeners  
 insertButton.setOnClickListener(v -> executorService.execute(this::insertDocument));  
 updateButton.setOnClickListener(v -> executorService.execute(this::updateDocument));  
 deleteButton.setOnClickListener(v -> executorService.execute(this::deleteDocument));  
 fetchButton.setOnClickListener(v -> executorService.execute(this::fetchDocument));  
 }  
  
 private void insertDocument() {  
 try (MongoClient mongoClient = MongoDBConnection.*getMongoClient*()) {  
 MongoDatabase database = mongoClient.getDatabase("hospital\_management");  
 MongoCollection<Document> collection = database.getCollection("users");  
  
 // Create a new document  
 Document document = new Document("name", nameInput.getText().toString())  
 .append("email", emailInput.getText().toString())  
 .append("age", Integer.*parseInt*(ageInput.getText().toString()));  
  
 // Insert the document  
 collection.insertOne(document);  
  
 // Show success message on the UI thread  
 runOnUiThread(() -> Toast.makeText(this, "Document Inserted!", Toast.*LENGTH\_SHORT*).show());  
 } catch (Exception e) {  
 runOnUiThread(() -> Toast.makeText(this, "Error: " + e.getMessage(), Toast.*LENGTH\_SHORT*).show());  
 }  
 }  
  
 private void updateDocument() {  
 try (MongoClient mongoClient = MongoDBConnection.*getMongoClient*()) {  
 MongoDatabase database = mongoClient.getDatabase("hospital\_management");  
 MongoCollection<Document> collection = database.getCollection("users");  
  
 // Query and update  
 Document query = new Document("name", nameInput.getText().toString());  
 Document update = new Document("$set", new Document("email", emailInput.getText().toString())  
 .append("age", Integer.*parseInt*(ageInput.getText().toString())));  
  
 collection.updateOne(query, update);  
  
 runOnUiThread(() -> Toast.makeText(this, "Document Updated!", Toast.*LENGTH\_SHORT*).show());  
 } catch (Exception e) {  
 runOnUiThread(() -> Toast.makeText(this, "Error: " + e.getMessage(), Toast.*LENGTH\_SHORT*).show());  
 }  
 }  
  
 private void deleteDocument() {  
 try (MongoClient mongoClient = MongoDBConnection.*getMongoClient*()) {  
 MongoDatabase database = mongoClient.getDatabase("hospital\_management");  
 MongoCollection<Document> collection = database.getCollection("users");  
  
 // Query to delete  
 Document query = new Document("name", nameInput.getText().toString());  
 collection.deleteOne(query);  
  
 runOnUiThread(() -> Toast.makeText(this, "Document Deleted!", Toast.*LENGTH\_SHORT*).show());  
 } catch (Exception e) {  
 runOnUiThread(() -> Toast.makeText(this, "Error: " + e.getMessage(), Toast.*LENGTH\_SHORT*).show());  
 }  
 }  
  
 private void fetchDocument() {  
 try (MongoClient mongoClient = MongoDBConnection.*getMongoClient*()) {  
 MongoDatabase database = mongoClient.getDatabase("hospital\_management");  
 MongoCollection<Document> collection = database.getCollection("users");  
  
 // Query to fetch  
 Document query = new Document("name", nameInput.getText().toString());  
 Document document = collection.find(query).first();  
  
 if (document != null) {  
 runOnUiThread(() -> {  
 emailInput.setText(document.getString("email"));  
 ageInput.setText(String.*valueOf*(document.getInteger("age")));  
 Toast.makeText(this, "Document Fetched!", Toast.*LENGTH\_SHORT*).show();  
 });  
 } else {  
 runOnUiThread(() -> Toast.makeText(this, "No Document Found!", Toast.*LENGTH\_SHORT*).show());  
 }  
 } catch (Exception e) {  
 runOnUiThread(() -> Toast.makeText(this, "Error: " + e.getMessage(), Toast.*LENGTH\_SHORT*).show());  
 }  
 }  
}

Main.XML

<LinearLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <EditText  
 android:id="@+id/nameInput"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Name"  
 android:inputType="text" />  
  
 <EditText  
 android:id="@+id/emailInput"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Email"  
 android:inputType="textEmailAddress" />  
  
 <EditText  
 android:id="@+id/ageInput"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Age"  
 android:inputType="number" />  
  
 <Button  
 android:id="@+id/insertButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Insert" />  
  
 <Button  
 android:id="@+id/updateButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Update" />  
  
 <Button  
 android:id="@+id/deleteButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Delete" />  
  
 <Button  
 android:id="@+id/fetchButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Fetch" />  
</LinearLayout>

MongoDBConnection.java

package com.example.mongodbapp;  
  
import com.mongodb.client.MongoClient;  
import com.mongodb.client.MongoClients;  
  
public class MongoDBConnection {  
 private static final String *CONNECTION\_STRING* = "mongodb://10.0.2.2:27017";  
  
 public static MongoClient getMongoClient() {  
 return MongoClients.*create*(*CONNECTION\_STRING*);  
 }  
}

Dependency

implementation ("org.mongodb:mongodb-driver-sync:4.2.3")