



School Of Computing

## **End semester Lab exam**

**Android Application Development**

**20CYS404**



**Amrita Vishwa Vidyapeetham Chennai –  
601 103, Tamil Nadu, India.**

**Done by:**

**Srish N**

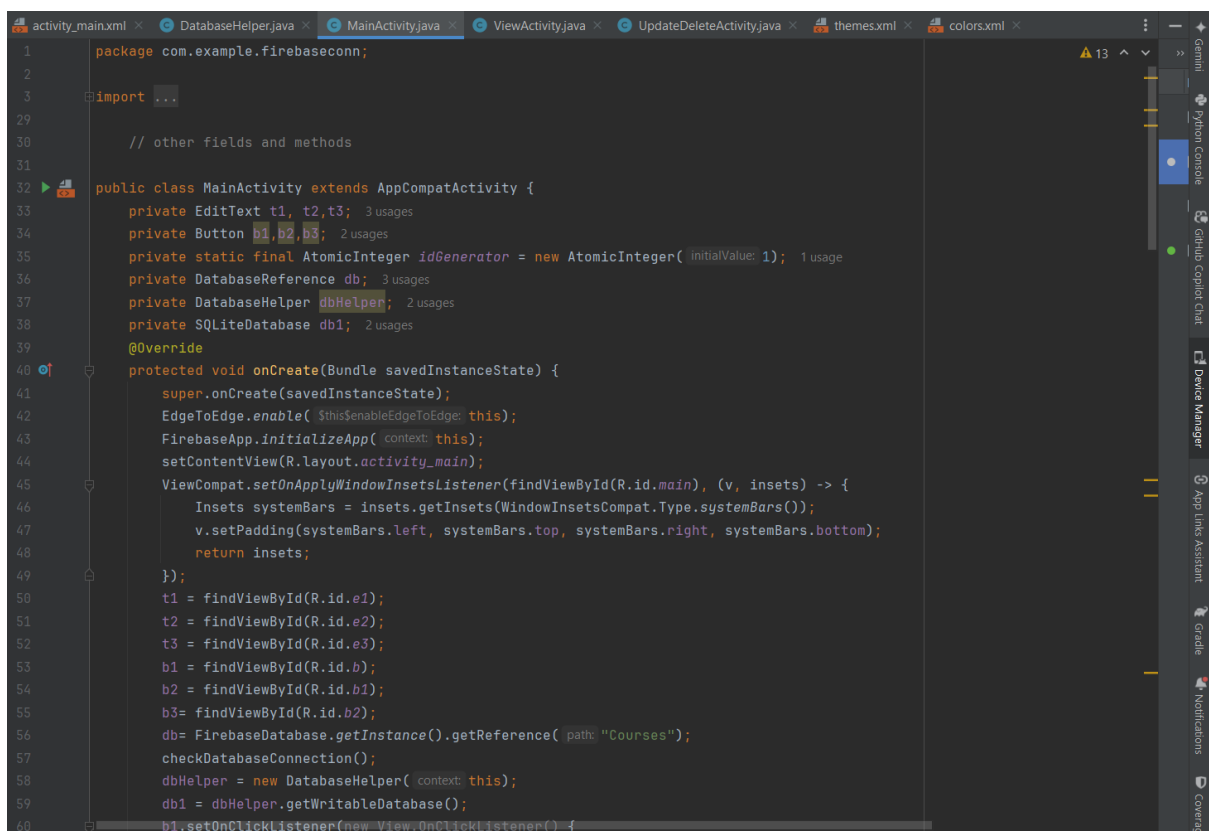
**CH.EN.U4CYS21081**

## Question:

1. Create a Course Registration Form, Name, Course and Fees.
  2. If OK button is pressed then the data should be saved in SQL lite and the "Record Added" toast should display.
  3. If view is clicked all the previous data should be displayed inside a TABLE Format.
  4. Perform EDIT, Delete, Back operations.
- Storage should be done in SQLite and Firebase

Code:

MainActivity.java



```

1 package com.example.firebaseconn;
2
3 import ...
4
5 // other fields and methods
6
7 public class MainActivity extends AppCompatActivity {
8     private EditText t1, t2, t3;
9     private Button b1, b2, b3;
10    private static final AtomicInteger idGenerator = new AtomicInteger(1);
11    private DatabaseReference db;
12    private DatabaseHelper dbHelper;
13    private SQLiteDatabase db1;
14
15    @Override
16    protected void onCreate(Bundle savedInstanceState) {
17        super.onCreate(savedInstanceState);
18        EdgeToEdge.enable(this);
19        FirebaseApp.initializeApp(this);
20        setContentView(R.layout.activity_main);
21        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
22            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
23            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
24            return insets;
25        });
26        t1 = findViewById(R.id.e1);
27        t2 = findViewById(R.id.e2);
28        t3 = findViewById(R.id.e3);
29        b1 = findViewById(R.id.b);
30        b2 = findViewById(R.id.b1);
31        b3 = findViewById(R.id.b2);
32        db = FirebaseDatabase.getInstance().getReference("Courses");
33        checkDatabaseConnection();
34        dbHelper = new DatabaseHelper(this);
35        db1 = dbHelper.getWritableDatabase();
36        b1.setOnClickListener(new View.OnClickListener() {
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

```

```

private void submitData() { 1 usage
    String name = t1.getText().toString();
    String course = t2.getText().toString();
    String fee = t3.getText().toString();

    String id = String.valueOf(idGenerator.getAndIncrement());
    if (id != null) {
        db.child(id).setValue(new DataModel(name, course, fee), new DatabaseReference.CompletionListener() {
            @Override
            public void onComplete(DatabaseError databaseError, DatabaseReference databaseReference) {
                if (databaseError != null) {
                    Toast.makeText(context: MainActivity.this, text: "Data could not be saved: " + databaseError.getMessage(), Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(context: MainActivity.this, text: "Data saved successfully!", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    ContentValues values = new ContentValues();
    values.put(DatabaseHelper.COLUMN_NAME, name);
    values.put(DatabaseHelper.COLUMN_COURSE, course);
    values.put(DatabaseHelper.COLUMN_FEE, fee);

    long newRowId = db.insert(DatabaseHelper.TABLE_COURSES, nullColumnHack null, values);
    if (newRowId != -1) {
        Toast.makeText(context: MainActivity.this, text: "Data saved to SQLite successfully!", Toast.LENGTH_SHORT).show();
    } else {
        Toast.makeText(context: MainActivity.this, text: "Data could not be saved to SQLite.", Toast.LENGTH_SHORT).show();
    }

    t1.setText("");
    t2.setText("");
}

```

## View Activitu.java

```

db = FirebaseDatabase.getInstance().getReference( path: "Courses");
td1 = findViewById(R.id.td);
retrieveData();
}

private void retrieveData() { 1 usage
    db.addValueEventListener(new ValueEventListener() {
        @Override no usages
        public void onDataChange(@NonNull DataSnapshot ds) {
            StringBuilder data = new StringBuilder();
            for (DataSnapshot snapshot : ds.getChildren()) {
                MainActivity.DataModel model = snapshot.getValue(MainActivity.DataModel.class);
                if (model != null) {
                    i++;
                    data.append("id: ").append(i).append(" - ");
                    data.append("Name: ").append(model.getData1()).append(", ");
                    data.append("Course: ").append(model.getData2()).append(", ");
                    data.append("Fee: ").append(model.getData3()).append("\n");
                }
            }
            td1.setText(data.toString());
            Toast.makeText(context: ViewActivity.this, text: "Retrieved data.", Toast.LENGTH_SHORT).show();
        }

        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {
            td1.setText("Error: " + databaseError.getMessage());
        }
    });
}
}

```

Updated deleteactivity.java

21|10|2024

```
public class UpdateDeleteActivity extends AppCompatActivity {  
    private void updateData() { 1 usage  
        String course = t2.getText().toString();  
        String fee = t3.getText().toString();  
  
        if (!id.isEmpty()) {  
            // Update Firebase  
            db.child(id).setValue(new MainActivity.DataModel(name, course, fee), new DatabaseReference.CompletionListener() {  
                @Override  
                public void onComplete(DatabaseError databaseError, DatabaseReference databaseReference) {  
                    if (databaseError != null) {  
                        Toast.makeText(context: UpdateDeleteActivity.this, text: "Data could not be updated: " + databaseError.getMessage(), Toast.LENGTH_SHORT).show();  
                    } else {  
                        Toast.makeText(context: UpdateDeleteActivity.this, text: "Data updated successfully!", Toast.LENGTH_SHORT).show();  
                    }  
                }  
            });  
  
            // Update SQLite  
            ContentValues values = new ContentValues();  
            values.put(DatabaseHelper.COLUMN_NAME, name);  
            values.put(DatabaseHelper.COLUMN_COURSE, course);  
            values.put(DatabaseHelper.COLUMN_FEE, fee);  
  
            int rowsAffected = sqLiteDb.update(DatabaseHelper.TABLE_COURSES, values, whereClause: DatabaseHelper.COLUMN_ID + "=?", new String[]{id});  
            if (rowsAffected > 0) {  
                Toast.makeText(context: this, text: "Data updated in SQLite successfully!", Toast.LENGTH_SHORT).show();  
            } else {  
                Toast.makeText(context: this, text: "Data could not be updated in SQLite.", Toast.LENGTH_SHORT).show();  
            }  
        } else {  
            Toast.makeText(context: this, text: "ID cannot be empty", Toast.LENGTH_SHORT).show();  
        }  
    }  
}
```

```
private void deleteData() {  
    String id = t4.getText().toString();  
  
    if (!id.isEmpty()) {  
        // Delete from Firebase  
        db.child(id).removeValue(new DatabaseReference.CompletionListener() {  
            @Override  
            public void onComplete(DatabaseError databaseError, DatabaseReference databaseReference) {  
                if (databaseError != null) {  
                    Toast.makeText(context: UpdateDeleteActivity.this, text: "Data could not be deleted: " + databaseError.getMessage(), Toast.LENGTH_SHORT).show();  
                } else {  
                    Toast.makeText(context: UpdateDeleteActivity.this, text: "Data deleted successfully!", Toast.LENGTH_SHORT).show();  
                }  
            }  
        });  
  
        // Delete from SQLite  
        int rowsDeleted = sqLiteDb.delete(DatabaseHelper.TABLE_COURSES, whereClause: DatabaseHelper.COLUMN_ID + "=?", new String[]{id});  
        if (rowsDeleted > 0) {  
            Toast.makeText(context: this, text: "Data deleted from SQLite successfully!", Toast.LENGTH_SHORT).show();  
        } else {  
            Toast.makeText(context: this, text: "Data could not be deleted from SQLite.", Toast.LENGTH_SHORT).show();  
        }  
    } else {  
        Toast.makeText(context: this, text: "ID cannot be empty", Toast.LENGTH_SHORT).show();  
    }  
}
```

Database helper.java

21|10|2024

```
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DatabaseHelper extends SQLiteOpenHelper { no usages
    private static final String DATABASE_NAME = "courses.db"; 1 usage
    private static final int DATABASE_VERSION = 1; 1 usage

    public static final String TABLE_COURSES = "courses"; 2 usages
    public static final String COLUMN_ID = "_id"; 1 usage
    public static final String COLUMN_NAME = "name"; 1 usage
    public static final String COLUMN_COURSE = "course"; 1 usage
    public static final String COLUMN_FEE = "fee"; 1 usage

    private static final String TABLE_CREATE = 1 usage
        "CREATE TABLE " + TABLE_COURSES + " (" +
            COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
            COLUMN_NAME + " TEXT, " +
            COLUMN_COURSE + " TEXT, " +
            COLUMN_FEE + " TEXT);";

    public DatabaseHelper(Context context) { no usages
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(TABLE_CREATE);
    }

    @Override 10 usages
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_COURSES);
        onCreate(db);
    }
}
```

## Activity main.xml

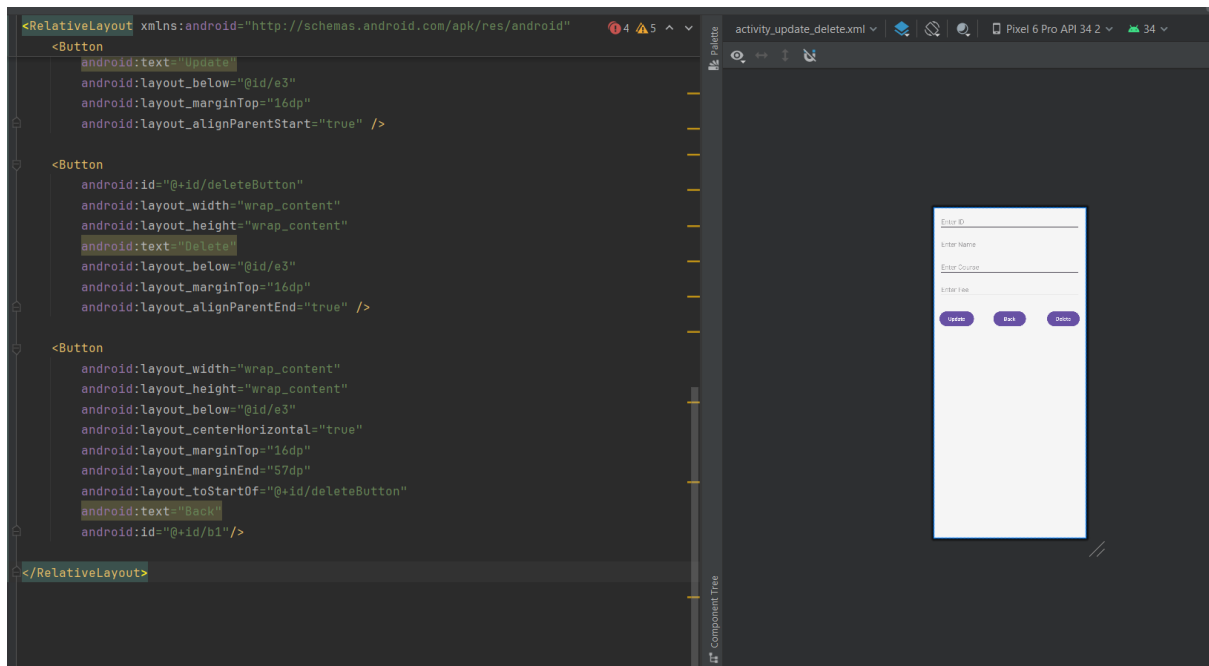
The screenshot displays the Android Studio environment. On the left, the XML layout for `activity_main.xml` is shown, featuring a `<RelativeLayout>` with the following components:

- `<TextView>` with attributes: `android:layout_width="wrap_content"`, `android:layout_height="wrap_content"`, `android:layout_marginLeft="100dp"`, `android:text="Fee:"`, `android:id="@+id/te"`, `android:layout_below="@id/e2"`, and `android:textSize="20sp"/>`.
- `<EditText>` with attributes: `android:id="@+id/e3"`, `android:layout_width="wrap_content"`, `android:layout_height="wrap_content"`, `android:layout_toRightOf="@id/te"`, `android:layout_marginTop="200dp"`, and `android:hint="Enter Fee" />`.
- `<Button>` with attributes: `android:id="@+id/b"`, `android:layout_width="wrap_content"`, `android:layout_height="wrap_content"`, `android:layout_below="@id/e3"`, `android:layout_marginLeft="90dp"`, and `android:text="OK" />`.
- `<Button>` with attributes: `android:id="@+id/b1"`, `android:layout_width="wrap_content"`, `android:layout_height="wrap_content"`, `android:layout_below="@id/e3"`, and `android:layout_toEndOf="@id/b"`.

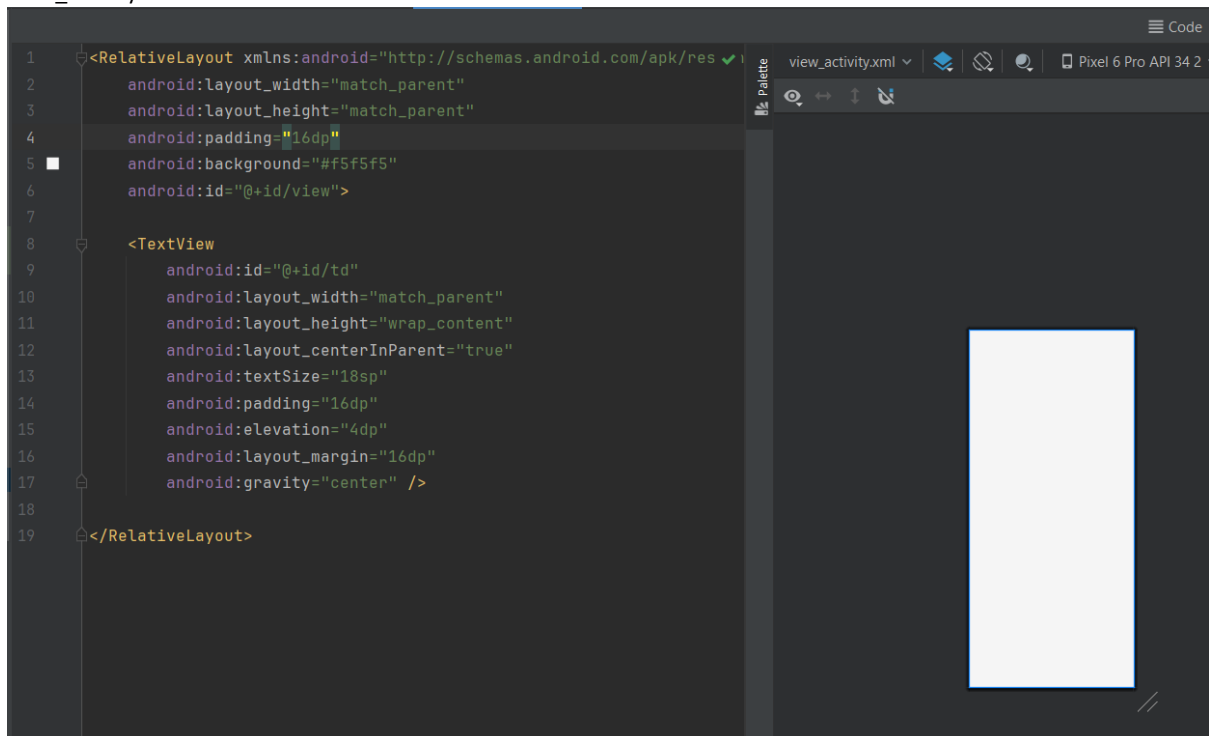
On the right, the UI preview shows a mobile application interface. At the top, it displays the user's name "SRISH N" and a unique identifier "CH.EN.U4CYS21081". Below this, the title "Course Registration" is followed by a form with two input fields: "Name: Enter Name" and "Course: Enter Course". A "Fee" label is positioned next to the "Course" field. At the bottom of the form, there are three buttons: "OK", "View", and "Update/Delete".

21|10|2024

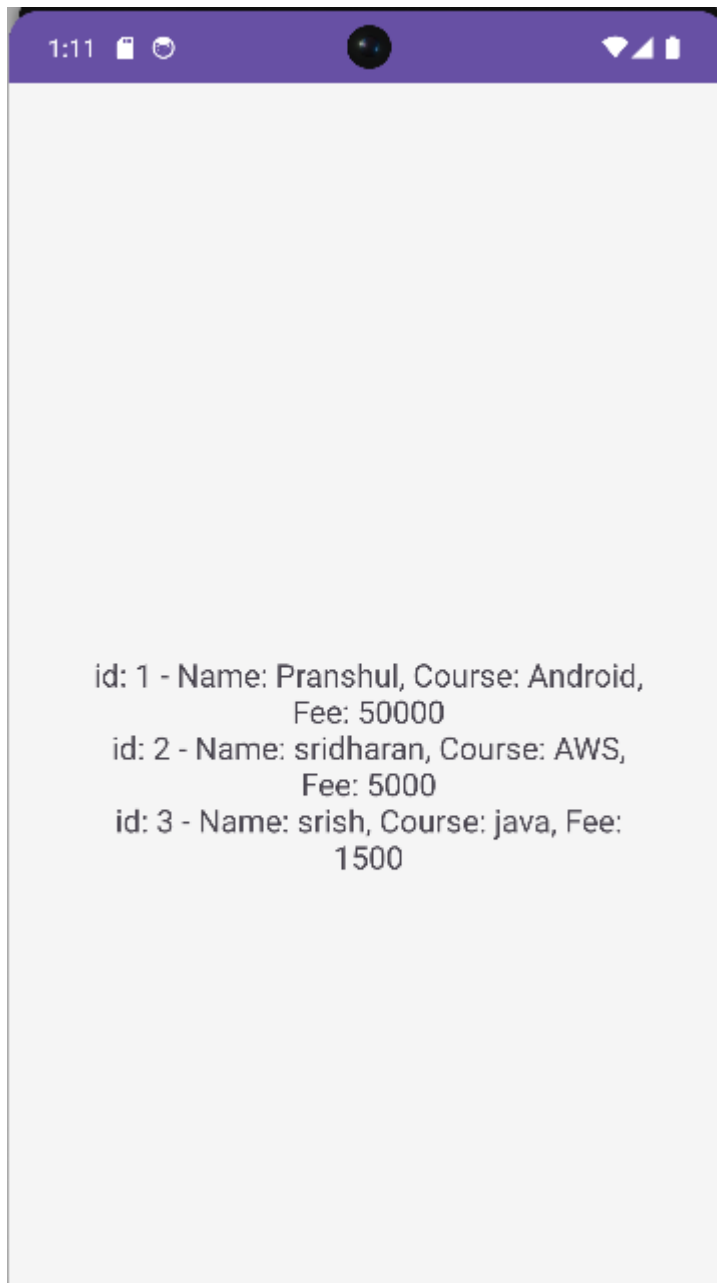
## Activity\_update\_delete.xml



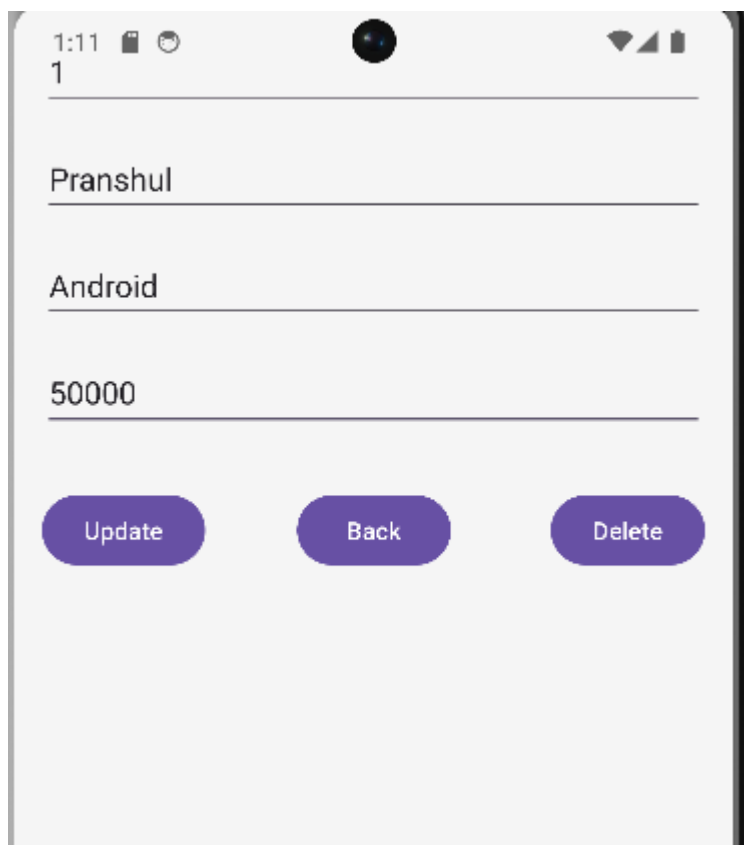
## View\_activity.xml



Outputs:



21|10|2024



1:11 1

---

Pranshul

---

Android

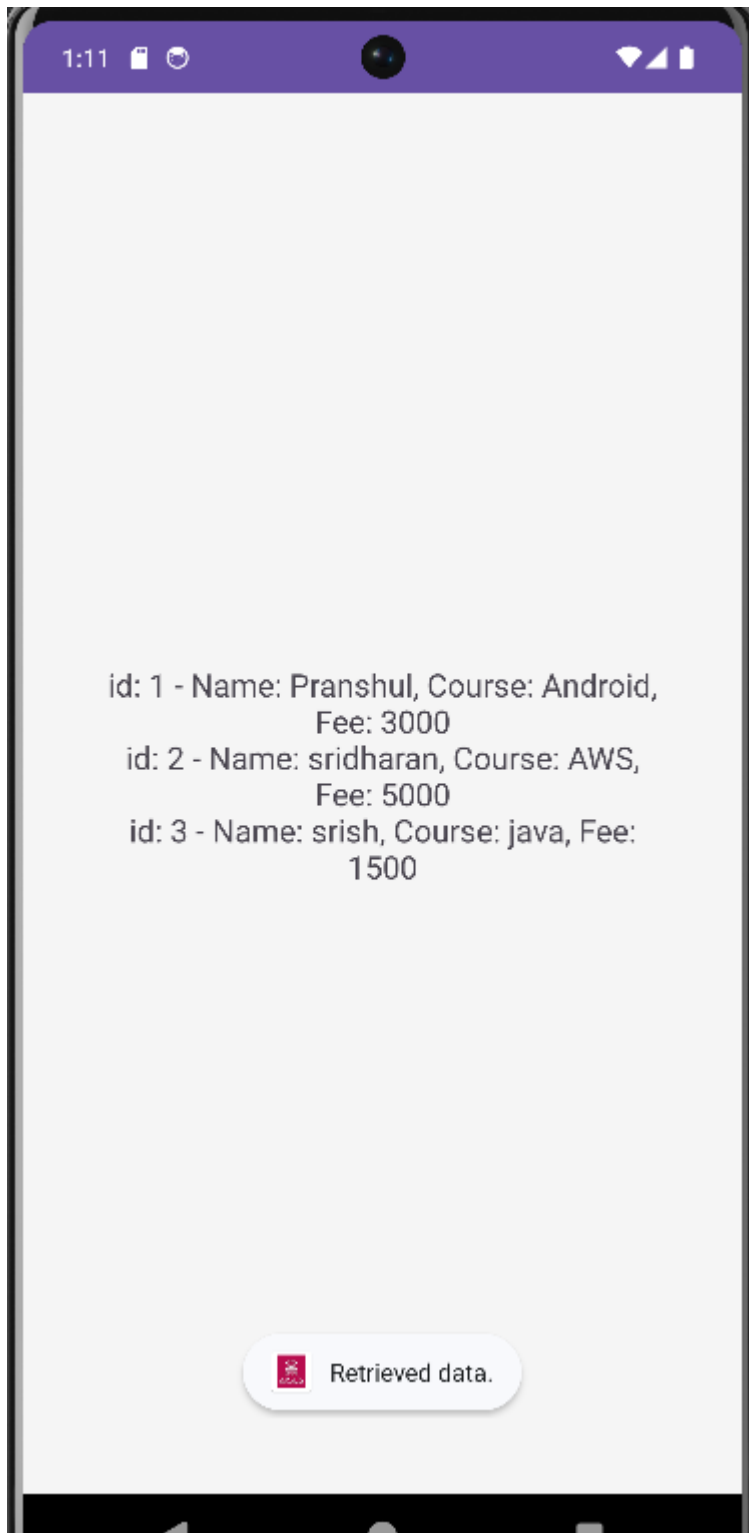
---

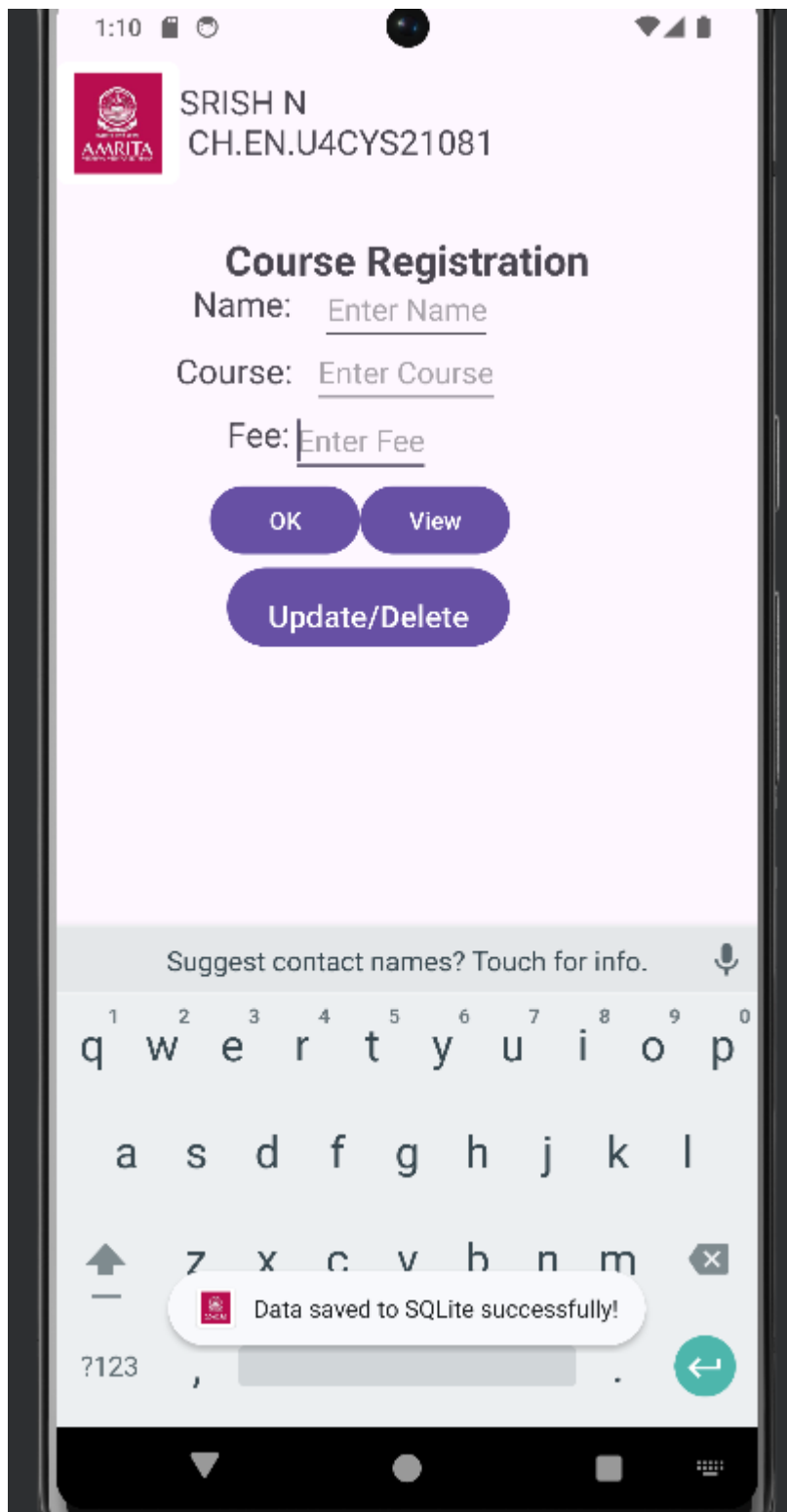
50000

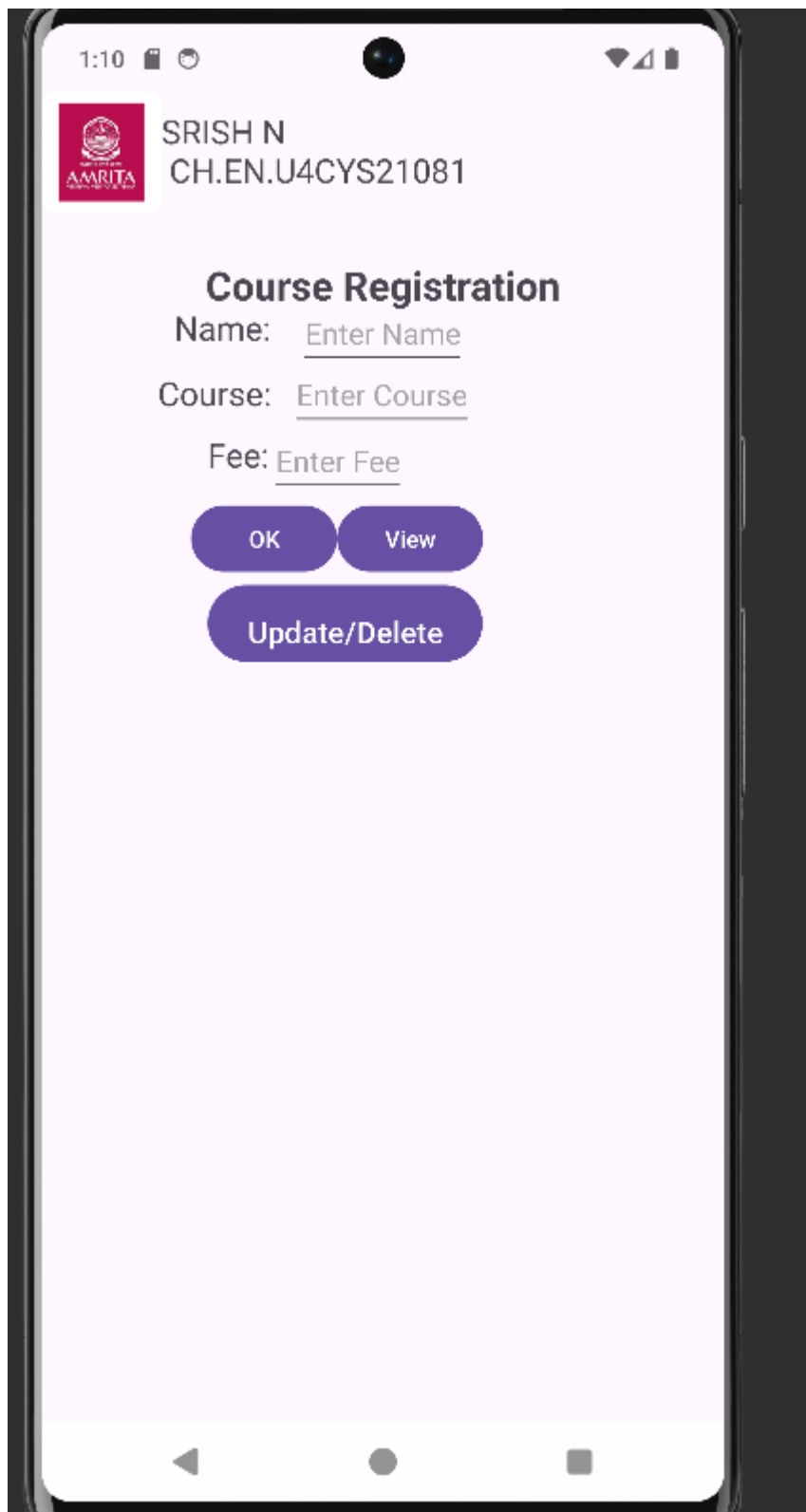
---

Update Back Delete









The image shows a mobile application interface for course registration. At the top, the status bar displays the time 1:10, signal strength, and battery level. Below the status bar, there is a header section with a red square logo on the left containing a white emblem and the word 'AMRITA' in white text. To the right of the logo, the user's name 'SRISH N' and ID 'CH.EN.U4CYS21081' are displayed in black text. The main content area has a light pink background and features the title 'Course Registration' in bold black text. Below the title, there are three input fields: 'Name: Enter Name', 'Course: Enter Course', and 'Fee: Enter Fee'. Each input field has a light blue underline. Below the input fields, there are three purple buttons with white text: 'OK', 'View', and 'Update/Delete'. The 'OK' and 'View' buttons are positioned side-by-side, and the 'Update/Delete' button is centered below them. At the bottom of the screen, there is a white navigation bar with three black icons: a triangle, a circle, and a square.

1:10

SRISH N  
CH.EN.U4CYS21081

**Course Registration**

Name:

Course:

Fee:

OK View

Update/Delete