

## **Activity\_main =>**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="20dp">

    <EditText
        android:id="@+id/etRoll"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Enter Roll Number"
        android:inputType="number"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>

    <EditText
        android:id="@+id/etName"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Enter Student Name"
        app:layout_constraintTop_toBottomOf="@id/etRoll"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="16dp"/>

    <EditText
        android:id="@+id/etMarks"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="Enter Marks"
        android:inputType="number"
        app:layout_constraintTop_toBottomOf="@id/etName"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="16dp"/>
```

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fabAdd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:layout_marginStart="32dp"
    android:layout_marginBottom="16dp"
    android:contentDescription="Add"
    android:src="@android:drawable/ic_input_add"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fabRead"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:contentDescription="Read"
    android:src="@android:drawable/ic_menu_search"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toEndOf="@id/fabAdd"
    android:layout_margin="16dp"/>
```

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fabEdit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:contentDescription="Update"
    android:src="@android:drawable/ic_menu_edit"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toEndOf="@id/fabRead"
    android:layout_margin="16dp"/>
```

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fabDelete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    android:layout_marginStart="112dp"
    android:layout_marginBottom="24dp"
    android:contentDescription="Delete"
    android:src="@android:drawable/ic_menu_delete"
```

```
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toEndOf="@id/fabEdit" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

## **Student.kt =>**

```
package com.example.ca1
```

```
data class Student(
    var roll: Int? = null,
    var name: String? = null,
    var marks: String? = null
)
```

## **MainActivity.kt =>**

```
package com.example.ca1
```

```
import android.os.Bundle
import android.widget.EditText
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import com.google.android.material.floatingactionbutton.FloatingActionButton
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase
```

```
class MainActivity : AppCompatActivity() {
    private lateinit var etRoll: EditText
    private lateinit var etName: EditText
    private lateinit var etMarks: EditText
    private lateinit var btnRead: FloatingActionButton
    private lateinit var btnAdd: FloatingActionButton
    private lateinit var btnEdit: FloatingActionButton
    private lateinit var btnDelete: FloatingActionButton
    private lateinit var database: DatabaseReference

    override fun onCreate(savedInstanceState: Bundle?) {
```

```

super.onCreate(savedInstanceState)
enableEdgeToEdge()
setContentView(R.layout.activity_main)

etRoll = findViewById(R.id.etRoll)
etName = findViewById(R.id.etName)
etMarks = findViewById(R.id.etMarks)
btnAdd = findViewById(R.id.fabAdd)
btnRead = findViewById(R.id.fabRead)
btnEdit = findViewById(R.id.fabEdit)
btnDelete = findViewById(R.id.fabDelete)

database = FirebaseDatabase.getInstance().getReference("Students")

btnAdd.setOnClickListener {
    val roll = etRoll.text.toString().toIntOrNull()
    val name = etName.text.toString()
    val marks = etMarks.text.toString()

    if (roll != null && name.isNotEmpty() && marks.isNotEmpty()) {
        val student = Student(roll, name, marks)

        database.child(roll.toString()).setValue(student)
            .addOnSuccessListener {
                Toast.makeText(this, "Student Added", Toast.LENGTH_SHORT).show()
            }
            .addOnFailureListener {
                Toast.makeText(this, "Error: ${it.message}", Toast.LENGTH_SHORT).show()
            }
    } else {
        Toast.makeText(this, "Fill all fields", Toast.LENGTH_SHORT).show()
    }
}

btnRead.setOnClickListener {
    val roll = etRoll.text.toString()

    if (roll.isNotEmpty()) {
        database.child(roll).get()
            .addOnSuccessListener {
                if (it.exists()) {

```

```

        val name = it.child("name").value
        val marks = it.child("marks").value

        etName.setText(name.toString())
        etMarks.setText(marks.toString())

        Toast.makeText(this, "Student Found", Toast.LENGTH_SHORT).show()
    } else {
        Toast.makeText(this, "Student Not Found", Toast.LENGTH_SHORT).show()
    }
}
} else {
    Toast.makeText(this, "Enter Roll No", Toast.LENGTH_SHORT).show()
}
}

btnEdit.setOnClickListener {
    val roll = etRoll.text.toString()
    val name = etName.text.toString()
    val marks = etMarks.text.toString()

    if (roll.isNotEmpty() && name.isNotEmpty() && marks.isNotEmpty()) {
        val student = Student(roll.toInt(), name, marks)

        database.child(roll).setValue(student)
            .addOnSuccessListener {
                Toast.makeText(this, "Student Updated", Toast.LENGTH_SHORT).show()
            }
            .addOnFailureListener {
                Toast.makeText(this, "Error: ${it.message}", Toast.LENGTH_SHORT).show()
            }
    } else {
        Toast.makeText(this, "Fill all fields", Toast.LENGTH_SHORT).show()
    }
}

btnDelete.setOnClickListener {
    val roll = etRoll.text.toString()

    if (roll.isNotEmpty()) {
        database.child(roll).removeValue()
    }
}

```

```

        .addOnSuccessListener {
            Toast.makeText(this, "Student Deleted", Toast.LENGTH_SHORT).show()

            etRoll.text.clear()
            etName.text.clear()
            etMarks.text.clear()
        }
        .addOnFailureListener {
            Toast.makeText(this, "Error: ${it.message}", Toast.LENGTH_SHORT).show()
        }
    } else {
        Toast.makeText(this, "Enter Roll No", Toast.LENGTH_SHORT).show()
    }
}
}
}
}
}

```

**Output =>**

