CA1

YUNESH KUMAR

12217704
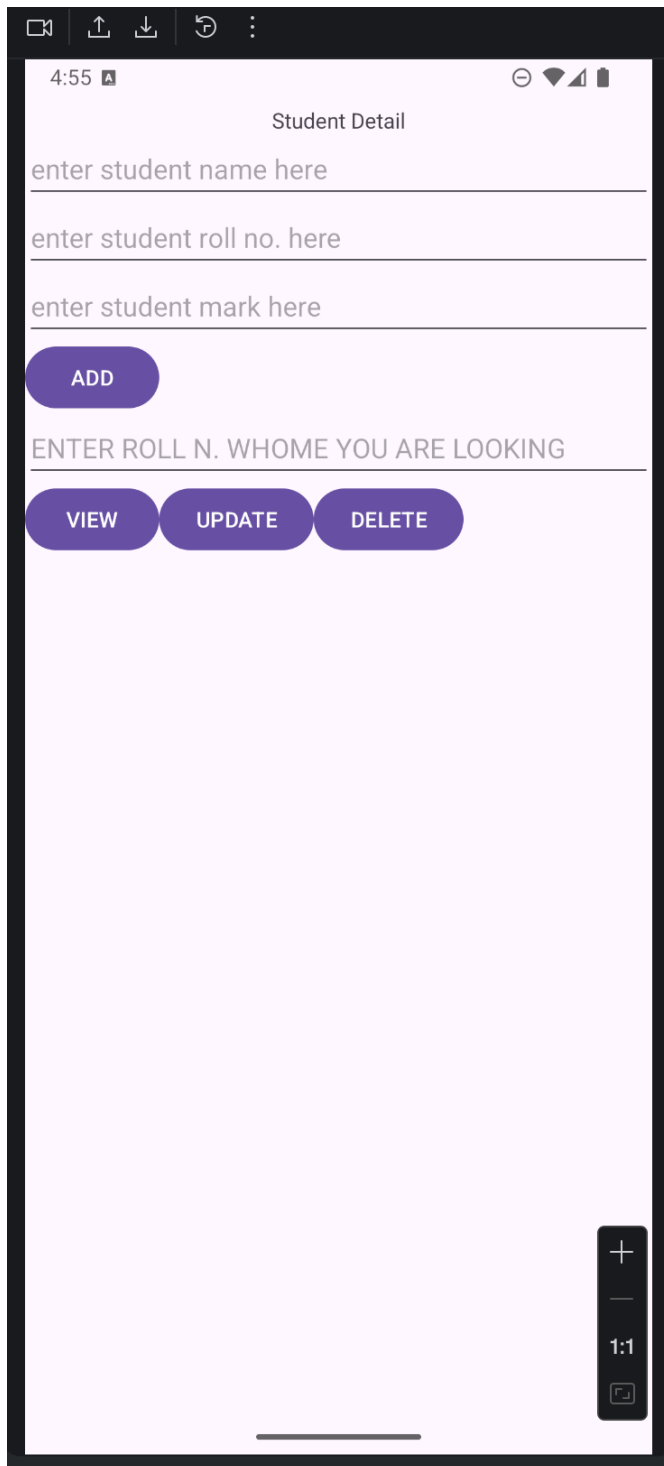
```
XML CODE-
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="30dp"
        android:text="Student Detail"/>

    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="enter student name here"/>
    <EditText
        android:id="@+id/rollNo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="enter student roll no. here"/>
    <EditText
        android:id="@+id/marks"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="enter student mark here"/>

    <Button
        android:id="@+id/add"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="ADD"/>
    <EditText
        android:id="@+id/d_rollNo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="ENTER ROLL N. WHOME YOU ARE LOOKING"/>
    <LinearLayout
        android:layout_width="wrap_content"
```

```xml
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <Button
            android:id="@+id/view"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="VIEW"/>
        <Button
            android:id="@+id/update"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="UPDATE"/>
        <Button
            android:id="@+id/delete"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="DELETE"/>

    </LinearLayout>
</LinearLayout>
```

Student Detail

enter student name here

enter student roll no. here

enter student mark here

ADD

ENTER ROLL N. WHOME YOU ARE LOOKING

VIEW   UPDATE   DELETE

```kotlin
Kotlin -

data class Student -

package com.example.ca1task

data class Student(
    val Name: String,
```

```kotlin
    val Roll_no: Int,
    val Mark: Int
)
```

```kotlin
MainActivity Class-

package com.example.ca1task

import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowCompat.enableEdgeToEdge
import androidx.core.view.WindowInsetsCompat
import com.google.firebase.Firebase
import com.google.firebase.database.DatabaseReference
import com.google.firebase.database.FirebaseDatabase
import com.google.firebase.database.database

class MainActivity : AppCompatActivity() {

    private lateinit var dbRef: DatabaseReference

    private lateinit var etName: EditText
    private lateinit var etRoll: EditText
    private lateinit var etMarks: EditText
    private lateinit var etDRoll: EditText

    private lateinit var btnAdd: Button
    private lateinit var btnView: Button
    private lateinit var btnUpdate: Button
    private lateinit var btnDelete: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        // Firebase Reference
        dbRef = FirebaseDatabase.getInstance().getReference("Students")

        // Bind Views
        etName = findViewById(R.id.name)
        etRoll = findViewById(R.id.rollNo)
        etMarks = findViewById(R.id.marks)
        etDRoll = findViewById(R.id.d_rollNo)

        btnAdd = findViewById(R.id.add)
        btnView = findViewById(R.id.view)
        btnUpdate = findViewById(R.id.update)
        btnDelete = findViewById(R.id.delete)

        // ADD
        btnAdd.setOnClickListener {
            addStudent()
        }
```

```kotlin
        // VIEW
        btnView.setOnClickListener {
            viewStudent()
        }

        // UPDATE
        btnUpdate.setOnClickListener {
            updateStudent()
        }

        // DELETE
        btnDelete.setOnClickListener {
            deleteStudent()
        }
    }

    private fun addStudent() {
        val name = etName.text.toString()
        val roll = etRoll.text.toString()
        val marks = etMarks.text.toString()

        if (name.isEmpty() || roll.isEmpty() || marks.isEmpty()) {
            toast("Please fill all fields")
            return
        }

        val student = Student(name, roll.toInt(), marks.toInt())
        dbRef.child(roll).setValue(student)

        toast("Student Added")
        clearFields()
    }

    private fun viewStudent() {
        val roll = etDRoll.text.toString()

        if (roll.isEmpty()) {
            toast("Enter roll number")
            return
        }

        dbRef.child(roll).get().addOnSuccessListener {
            if (it.exists()) {
                etName.setText(it.child("name").value.toString())
                etRoll.setText(it.child("rollNo").value.toString())
                etMarks.setText(it.child("marks").value.toString())
                toast("Student Found")
            } else {
                toast("Student Not Found")
            }
        }
    }

    private fun updateStudent() {
        val roll = etDRoll.text.toString()

        if (roll.isEmpty()) {
            toast("Enter roll number")
            return
        }
```

```kotlin
        val updates = mapOf(
            "name" to etName.text.toString(),
            "marks" to etMarks.text.toString()
        )

        dbRef.child(roll).updateChildren(updates)
        toast("Student Updated")
        clearFields()
    }

    private fun deleteStudent() {
        val roll = etDRoll.text.toString()

        if (roll.isEmpty()) {
            toast("Enter roll number")
            return
        }

        dbRef.child(roll).removeValue()
        toast("Student Deleted")
        clearFields()
    }

    private fun clearFields() {
        etName.text.clear()
        etRoll.text.clear()
        etMarks.text.clear()
        etDRoll.text.clear()
    }

    private fun toast(msg: String) {
        Toast.makeText(this, msg, Toast.LENGTH_SHORT).show()
    }
}
```