

CSE227CA.kt

```
package canvas

import android.annotation.SuppressLint
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity
import com.example.myapp.R
import com.google.firebaseio.database.*

class CSE227CA : AppCompatActivity() {

    private lateinit var et_book: EditText
    private lateinit var tv_result: TextView

    private lateinit var et_id: EditText
    private lateinit var add_btn: Button
    private lateinit var get_btn: Button
    private lateinit var get_all: Button

    private lateinit var dref: DatabaseReference

    @SuppressLint("MissingInflatedId")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.item_book)
        tv_result=findViewById(R.id.et_text_1)
        dref = FirebaseDatabase.getInstance().getReference("book")
        et_book = findViewById(R.id.et_book)
        et_id = findViewById(R.id.et_id)

        add_btn = findViewById(R.id.btn_add)
        get_btn = findViewById(R.id.btn_get)
        get_all = findViewById(R.id.btn_all)

        add_btn.setOnClickListener {
            addBook()
        }
    }

    private fun addBook() {
        val id = et_id.text.toString()
        val bookName = et_book.text.toString()

        if (id.isNotEmpty() && bookName.isNotEmpty()) {
            dref.push().setValue(Book(id, bookName))
            et_id.setText("")
            et_book.setText("")
            tv_result.text = "Book added successfully"
        } else {
            tv_result.text = "Please enter both ID and Book Name"
        }
    }

    private fun updateBook(book: Book) {
        val id = book.id
        val bookName = book.name

        if (id != null && bookName != null) {
            dref.child(id).setValue(book)
            tv_result.text = "Book updated successfully"
        } else {
            tv_result.text = "Failed to update book"
        }
    }

    private fun deleteBook(book: Book) {
        val id = book.id

        if (id != null) {
            dref.child(id).removeValue()
            tv_result.text = "Book deleted successfully"
        } else {
            tv_result.text = "Failed to delete book"
        }
    }

    private fun getAllBooks() {
        dref.addValueEventListener(object : ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                val booksList = mutableListOf()
                snapshot.children.forEach { child ->
                    val book = child.getValue(Book::class.java)
                    booksList.add(book)
                }
                tv_result.text = booksList.joinToString("\n")
            }

            override fun onCancelled(error: DatabaseError) {
                tv_result.text = "Failed to fetch books"
            }
        })
    }

    private fun getBookById(id: String) {
        dref.child(id).addValueEventListener(object : ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                val book = snapshot.getValue(Book::class.java)
                tv_result.text = book?.name ?: "Book not found"
            }

            override fun onCancelled(error: DatabaseError) {
                tv_result.text = "Failed to fetch book"
            }
        })
    }

    private fun getAllBooks() {
        dref.addValueEventListener(object : ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                val booksList = mutableListOf()
                snapshot.children.forEach { child ->
                    val book = child.getValue(Book::class.java)
                    booksList.add(book)
                }
                tv_result.text = booksList.joinToString("\n")
            }

            override fun onCancelled(error: DatabaseError) {
                tv_result.text = "Failed to fetch books"
            }
        })
    }

    private fun getBookById(id: String) {
        dref.child(id).addValueEventListener(object : ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                val book = snapshot.getValue(Book::class.java)
                tv_result.text = book?.name ?: "Book not found"
            }

            override fun onCancelled(error: DatabaseError) {
                tv_result.text = "Failed to fetch book"
            }
        })
    }

    private fun getAllBooks() {
        dref.addValueEventListener(object : ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                val booksList = mutableListOf()
                snapshot.children.forEach { child ->
                    val book = child.getValue(Book::class.java)
                    booksList.add(book)
                }
                tv_result.text = booksList.joinToString("\n")
            }

            override fun onCancelled(error: DatabaseError) {
                tv_result.text = "Failed to fetch books"
            }
        })
    }

    private fun getBookById(id: String) {
        dref.child(id).addValueEventListener(object : ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                val book = snapshot.getValue(Book::class.java)
                tv_result.text = book?.name ?: "Book not found"
            }

            override fun onCancelled(error: DatabaseError) {
                tv_result.text = "Failed to fetch book"
            }
        })
    }
}
```

```
}

get_btn.setOnClickListener {
    getBook()
}

get_all.setOnClickListener {
    getAllBooks()
}
}

private fun addBook() {

    val book = et_book.text.toString().trim()
    val id = et_id.text.toString().trim()

    val data = Book(book, id)

    dref.child(id).setValue(data)
        .addOnSuccessListener {
            et_book.text.clear()
            et_id.text.clear()
            Toast.makeText(this, "added", Toast.LENGTH_SHORT).show()
        }
        .addOnFailureListener {
            Toast.makeText(this, "not added", Toast.LENGTH_SHORT).show()
        }
}

private fun getBook() {

    val id = et_id.text.toString().trim()

    dref.child(id)
        .addListenerForSingleValueEvent(object : ValueEventListener {

            override fun onDataChange(snapshot: DataSnapshot) {

                if (snapshot.exists()) {

                    val data = snapshot.getValue(Book::class.java)
```

```
        et_book.setText(data?.book)

    } else {
        Toast.makeText(
            this@CSE227CA,
            "Not found",
            Toast.LENGTH_SHORT
        ).show()
    }
}

override fun onCancelled(error: DatabaseError) {
    Toast.makeText(
        this@CSE227CA,
        error.message,
        Toast.LENGTH_SHORT
    ).show()
}

private fun getAllBooks() {

    dref.addListenerForSingleValueEvent(object : ValueEventListener {

        override fun onDataChange(snapshot: DataSnapshot) {

            val list = ArrayList<Book>()

            for (child in snapshot.children) {

                val data = child.getValue(Book::class.java)

                if (data != null) {
                    list.add(data)
                }
            }

            Toast.makeText(
                this@CSE227CA,
                "Total books = ${list}",

```

```
        Toast.LENGTH_SHORT
    ).show()

    if (!list.isEmpty()) {
        et_book.setText(list[0].book)
    }
}

override fun onCancelled(error: DatabaseError) {
    Toast.makeText(
        this@CSE227CA,
        error.message,
        Toast.LENGTH_SHORT
    ).show()
}

}

}

}

package canvas

data class Book(
    var book: String? = null,
    var id: String?=null
)

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:orientation="vertical"
    android:layout_height="match_parent">

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text=""
        android:layout_marginTop="30sp"
        android:hint="Enter book name">

```

```
    android:id="@+id/et_book"
    />

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text=""
    android:hint="Enter id"

    android:id="@+id/et_id"
    />
<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text=""
    android:hint="Enter id"

    android:id="@+id/et_text_1"
    />

<Button
    android:layout_width="200sp"
    android:layout_height="100sp"
    android:text="add book"
    android:id="@+id/btn_add"

    />
<Button
    android:layout_width="200sp"
    android:layout_height="100sp"
    android:text="get book"
    android:id="@+id/btn_get"

    />
<Button
    android:layout_width="200sp"
    android:layout_height="100sp"
```

```
    android:text="get all book"
    android:id="@+id	btn_all"

/>
```

```
</LinearLayout>
```





