# MainActivity.kt

package com.example.20bcy10185\_yashika

import androidx.appcompat.app.AppCompatActivity import android.os.Bundle

import android.widget.Toast

import com.example.20bcy10185\_yashika.databinding.ActivityMainBinding

import kotlinx.coroutines.Dispatchers

import kotlinx.coroutines.GlobalScope import kotlinx.coroutines.launch import kotlinx.coroutines.withContext

class MainActivity : AppCompatActivity() {

private lateinit var binding : ActivityMainBinding private lateinit var appDb: AppDatabase

override fun onCreate(savedInstanceState: Bundle?) { super.onCreate(savedInstanceState) binding=ActivityMainBinding.inflate(layoutInflater) setContentView(binding.root )

appDb= AppDatabase.getDatabase(this) binding.button1.setOnClickListener{

submit()

}

binding.button2.setOnClickListener { show()

}

}

private fun submit(){

val name=binding.editText.text.toString()

val location=binding.editText2.text.toString() if(name.isNotEmpty()&&location.isNotEmpty())

{

val user=User(0L,name, location) GlobalScope.launch(Dispatchers.IO)

{

appDb.databasedao().insert(user)

}

binding.editText.text.clear() binding.editText2.text.clear()

Toast.makeText(this@MainActivity,"Saved",Toast.LENGTH\_SHORT).show()

}else

{

Toast.makeText(this@MainActivity,"Enter Data",Toast.LENGTH\_SHORT).show()

}

}

private suspend fun display(user: User)

{

withContext(Dispatchers.Main){

binding.tvname.text=user.name binding.tvlocation.text=user.location

}

}

private fun show()

{

val name=binding.editText3.text.toString() if(name.isNotEmpty()){

lateinit var user:User GlobalScope.launch {

user= appDb.databasedao().findByRoll(name)!! display(user)

}

}

}

}

# User.kt

package com.example.20bcy10185\_yashika

import androidx.room.ColumnInfo import androidx.room.Entity import androidx.room.PrimaryKey

@Entity(tableName = "my\_list") data class User(

@PrimaryKey(autoGenerate = true) val id:Long =0L, @ColumnInfo(name="name")

val name: String, @ColumnInfo(name="location") val location:String,

)

# AppDatabase.kt

package com.example.20bcy10185\_yashika

import android.content.Context import androidx.room.Database import androidx.room.Room

import androidx.room.RoomDatabase

import com.example. 20bcy10185\_yashika.AppDatabase

import com.example. 20bcy10185\_yashika.AppDatabase as AppDatabase1

@Database(entities = [User::class], version = 1) abstract class AppDatabase :RoomDatabase(){

abstract fun databasedao(): DatabaseDao companion object{

@Volatile

private var INSTANCE: AppDatabase1?=null

fun getDatabase(context : Context): AppDatabase1 {

var tempInstance= INSTANCE

if (tempInstance!=null) return tempInstance synchronized(this){

tempInstance=Room.databaseBuilder( context.applicationContext, AppDatabase1::class.java, "database"

).fallbackToDestructiveMigration()

.build()

INSTANCE=tempInstance

return tempInstance as AppDatabase

}

}

}

}

# DatabaseDao.kt

package com.example. 20bcy10185\_yashika

import androidx.room.Dao import androidx.room.Delete import androidx.room.Insert

import androidx.room.OnConflictStrategy import androidx.room.Query

@Dao

interface DatabaseDao{ @Query("SELECT \* FROM my\_list") fun getAll(): List<User>

@Query("SELECT \* FROM my\_list where name= :name") suspend fun findByRoll(name:String):User?

@Insert(onConflict = OnConflictStrategy.IGNORE) suspend fun insert(user: User)

@Delete

suspend fun delete(user: User)

@Query("DELETE FROM my\_list") suspend fun deleteAll()

}