GROCERY APP PROJECT

STEP - 1 : CREATE NEW PROJECT

In Android Studio create a new projet and select the empty activity. Give the project name and location of project and and select Kotlin as programming language.

SEEP - 2 : BUILD THE GRADLE

In project goto app>Grdle Scripts>built.gradle(Module). In that update code as follow:

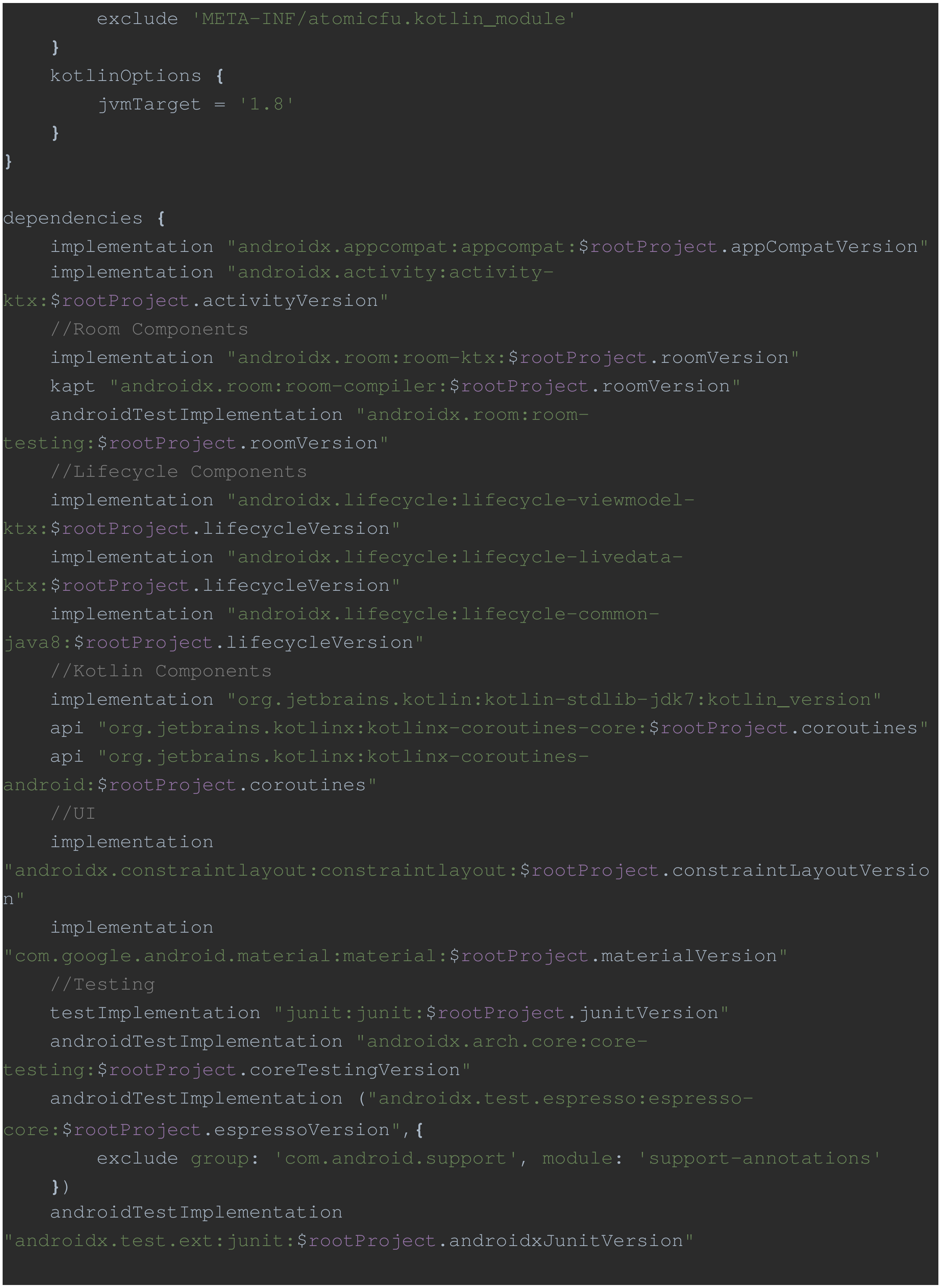
```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'kotlin-android'
    id 'kotlin-kapt'
}

android {
    compileSdk 32

    defaultConfig {
        applicationId "com.example.groceryapp"
        minSdk 21
        targetSdk 32
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    packagingOptions{
```



```
        exclude 'META-INF/atomicfu.kotlin_module'
    }
    kotlinOptions {
        jvmTarget = '1.8'
    }
}

dependencies {
    implementation "androidx.appcompat:appcompat:$rootProject.appCompatVersion"
    implementation "androidx.activity:activity-
ktx:$rootProject.activityVersion"
    //Room Components
    implementation "androidx.room:room-ktx:$rootProject.roomVersion"
    kapt "androidx.room:room-compiler:$rootProject.roomVersion"
    androidTestImplementation "androidx.room:room-
testing:$rootProject.roomVersion"
    //Lifecycle Components
    implementation "androidx.lifecycle:lifecycle-viewmodel-
ktx:$rootProject.lifecycleVersion"
    implementation "androidx.lifecycle:lifecycle-livedata-
ktx:$rootProject.lifecycleVersion"
    implementation "androidx.lifecycle:lifecycle-common-
java8:$rootProject.lifecycleVersion"
    //Kotlin Components
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:kotlin_version"
    api "org.jetbrains.kotlinx:kotlinx-coroutines-core:$rootProject.coroutines"
    api "org.jetbrains.kotlinx:kotlinx-coroutines-
android:$rootProject.coroutines"
    //UI
    implementation
"androidx.constraintlayout:constraintlayout:$rootProject.constraintLayoutVersio
n"
    implementation
"com.google.android.material:material:$rootProject.materialVersion"
    //Testing
    testImplementation "junit:junit:$rootProject.junitVersion"
    androidTestImplementation "androidx.arch.core:core-
testing:$rootProject.coreTestingVersion"
    androidTestImplementation ("androidx.test.espresso:espresso-
core:$rootProject.espressoVersion",{
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    androidTestImplementation
"androidx.test.ext:junit:$rootProject.androidxJunitVersion"
```

```
}
```

now goto app>Gradle Scripts>build.gradle(Project) and update the code as follow:

```
buildscript {
    ext.kotlin_version="1.6.0"
    repositories {
        google()
        mavenCentral()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:7.2.2'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
    }
}


task clean(type: Delete) {
    delete rootProject.buildDir
}
ext {
    activityVersion='1.5.1'
    appCompatVersion='1.5.1'
    constraintLayoutVersion='2.1.4'
    coreTestingVersion='2.1.0'
    coroutines='1.6.1'
    lifecycleVersion='2.5.1'
    materialVersion='1.6.1'
    roomVersion='2.4.3'
    junitVersion='4.13.2'
    espressoVersion='3.1.0'
    androidxJunitVersion='1.1.3'
}
```

Now click on "Sync Now"

STEP - 3 : UPDATING THE COLORS

Goto app>res>values>colors.xml and change the colors.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <color name="purple_200">#296D98</color>
    <color name="purple_500">#296D98</color>
    <color name="purple_700">#296D98</color>
    <color name="teal_200">#FF03DAC5</color>
    <color name="teal_700">#FF018786</color>
    <color name="black">#FF000000</color>
    <color name="white">#FFFFFFFF</color>
    <color name="mtrl_textinput_default_box_stroke_color" tools:overrise =
"true">#296D98</color>
    <color name="blue_shade_1">#0e2433</color>
    <color name="blue_shade_2">#1c4966</color>
</resources>
```

STEP - 4: UPDATING THE THEMES

Goto app>res>values>themes>themes.xml and update the code as follow:

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.GroceryApp"
parent="Theme.MaterialComponents.DayNight.DarkActionBar">
        <!-- Primary brand color. -->
        <item name="colorPrimary">@color/purple_500</item>
        <item name="colorPrimaryVariant">@color/purple_700</item>
        <item name="colorOnPrimary">@color/white</item>
        <!-- Secondary brand color. -->
        <item name="colorSecondary">@color/teal_200</item>
        <item name="colorSecondaryVariant">@color/teal_700</item>
        <item name="colorOnSecondary">@color/black</item>
        <!-- Status bar color. -->
        <item name="android:statusBarColor"
tools:targetApi="l">?attr/colorPrimaryVariant</item>
        <!-- Customize your theme here. -->
    </style>

    <style name="TextInputEditTextTheme"
parent="Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense">
        <item name="boxStrokeColor">#296D98</item>
        <item name="boxStrokeWidth">1dp</item>
    </style>
</resources>
```

STEP - 5 : CREATING THE CLASSES

Goto app>java>com.xamples and select com.examples and create new kotlin classes in it. Right click on mouse>new>kotlin classes and give the class name.

a.) Create GrocaryDao interface and write the code as follow:

```
package com.example.groceryapp

import androidx.lifecycle.LiveData
import androidx.room.*


@Dao
interface GroceryDao {
    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insert(item: GroceryItems)

    @Delete
    suspend fun delete(item: GroceryItems)

    @Query(value ="SELECT * FROM grocery_items")
    fun getAllGroceryItems() : LiveData<List<GroceryItems>>

}
```

Dao->data accessing object

b.)Create GrocaryItems class and write the code as follow:

```
package com.example.groceryapp

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

@Entity(tableName = "grocery_items")
data class GroceryItems (

    @ColumnInfo(name = "itemName")
    var itemName:String,
    @ColumnInfo(name = "itemQuantity")
    var itemQuantity:Int,

    @ColumnInfo(name = "itemPrice")
    var itemPrice:Int,

    ){
    @PrimaryKey(autoGenerate = true)
    var id:Int?=null
}
```

c.) Cerate GroceryDatabase class and write the code as follow:

```
package com.example.groceryapp

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

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

    abstract fun getGroceryDao():GroceryDao

    companion object {
        @Volatile
        private var instance: GroceryDatabase? = null
        private val LOCK = Any()

        operator fun invoke(context: Context) = instance ?: synchronized(LOCK)
{
            instance ?: createDatabase(context).also {
                instance =it
            }
        }

        private fun createDatabase(context: Context) =
            Room.databaseBuilder(
                context.applicationContext,
                GroceryDatabase::class.java,
                "Grocery.db"
            ).build()
    }
}
```

d.) Create GroceryRepository class and write the code as follow:

```
package com.example.groceryapp

class GroceryRepository(private val db:GroceryDatabase) {
    suspend fun insert(items: GroceryItems) =db.getGroceryDao().insert(items)
    suspend fun delete(items: GroceryItems) =db.getGroceryDao().delete(items)

    fun getAllItems()=db.getGroceryDao().getAllGroceryItems()
}
```

e.) Create GroceryViewModel class and write the code as follow:

```
package com.example.groceryapp

import androidx.lifecycle.ViewModel
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.launch

class GroceryViewModal(private val repository: GroceryRepository):ViewModel() {

    fun insert(items: GroceryItems)=GlobalScope.launch{
        repository.insert(items)
    }

    fun delete(items: GroceryItems)=GlobalScope.launch {
        repository.delete(items)
    }

    fun getAllGroceryItems()=repository.getAllItems()
}
```

f.) Create GroceryViewModelFactory class and write the code as follow:

```
package com.example.groceryapp

import androidx.lifecycle.ViewModel
import androidx.lifecycle.ViewModelProvider

class GroceryViewModalFactory(private val repository: GroceryRepository)
:ViewModelProvider.NewInstanceFactory(){

    override fun <T : ViewModel> create(modelClass: Class<T>): T {
        return GroceryViewModal(repository) as T
    }
}
```

STEP - 6 : UPDATING activity_main.xml FILE

Write the following code in activity_main.xml file as follow:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_width="match_parent"
    android:background="@color/blue_shade_1"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:listitem="@layout/grocery_rv_item"
        android:id="@+id/idRVItems"/>

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:src="@drawable/ic_add"
        app:tint="@color/white"
        android:layout_alignParentBottom="true"
        android:layout_margin="20dp"
        android:id="@+id/idFABAdd"
        app:backgroundTint="@color/blue_shade_2"/>


</RelativeLayout>
```

STEP - 7 : CREATING NEW LAYOUT FILE

create new layout file and write the code as follow:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_margin="8dp"
    android:backgroundTint="@color/blue_shade_2"
    app:cardCornerRadius="5dp"
    app:cardElevation="4dp">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        >
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:weightSum="5"
            android:id="@+id/idLL1">
            <TextView
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="2"
                android:text="Item Name"
                android:textColor="@color/white"
                android:padding="4dp"
                android:layout_margin="3dp"
                android:id="@+id/idRVItemName"/>

            <TextView
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:id="@+id/idTVQuantity"
                android:text="Quantity"
                android:textColor="@color/white"
```

```xml
            android:padding="4dp"
            android:layout_margin="3dp"/>

        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1.5"
            android:text="Rate"
            android:textColor="@color/white"
            android:padding="4dp"
            android:layout_margin="3dp"
            android:id="@+id/idTVRate"/>

        <ImageView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.5"
            android:padding="4dp"
            app:tint="@color/white"
            android:src="@drawable/ic_delete"
            android:layout_margin="3dp"
            android:id="@+id/idIVDelete"/>


    </LinearLayout>

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Total Cost"
        android:layout_toLeftOf="@id/idTVTotalAmt"
        android:layout_below="@id/idLL1"
        android:textColor="@color/white"
        android:layout_margin="3dp"
        android:padding="4dp"
        android:id="@+id/idTVHeading"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/idTVTotalAmt"
        android:textColor="@color/white"
        android:layout_below="@id/idLL1"
        android:layout_alignParentEnd="true"
        android:text="Amt"
```

```
            android:padding="4dp"
            android:layout_margin="3dp"
            android:layout_alignParentRight="true"/>

    </RelativeLayout>

</androidx.cardview.widget.CardView>
```

STEP - 8 : CREATING NEW CLASS

Create new class GroceryRVAdaptor and write the code as follow:

```
package com.example.groceryapp

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.ImageView
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class GroceryRVAdapter(
    var list: List<GroceryItems>,
    val groceryItemClickInterface: GroceryItemClickInterface
    ): RecyclerView.Adapter<GroceryRVAdapter.GroceryViewHolder>() {


    inner class GroceryViewHolder(itemView: View):
RecyclerView.ViewHolder(itemView){
        val nameTV=itemView.findViewById<TextView>(R.id.idRVItemName)
        val quantityTV=itemView.findViewById<TextView>(R.id.idTVQuantity)
        val rateTV=itemView.findViewById<TextView>(R.id.idTVRate)
        val amountTV=itemView.findViewById<TextView>(R.id.idTVTotalAmt)
        val deleteIV=itemView.findViewById<ImageView>(R.id.idIVDelete)

    }




    interface GroceryItemClickInterface{
        fun onItemClick(groceryItems: GroceryItems)
    }
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
GroceryViewHolder {
        val
view=LayoutInflater.from(parent.context).inflate(R.layout.grocery_rv_item,parent,
false)
        return GroceryViewHolder(view)
    }

    override fun onBindViewHolder(holder: GroceryViewHolder, position: Int) {
        holder.nameTV.text=list.get(position).itemName
        holder.quantityTV.text=list.get(position).itemQuantity.toString()
        holder.rateTV.text="Rs. "+list.get(position).itemPrice.toString()
        val
itemTotal:Int=list.get(position).itemPrice*list.get(position).itemQuantity
        holder.amountTV.text="Rs. "+itemTotal.toString()
        holder.deleteIV.setOnClickListener{
            groceryItemClickInterface.onItemClick(list.get(position))
        }

    }

    override fun getItemCount(): Int {
        return list.size
    }
}
```

STEP - 9 : CREATE NEW LAYOUT

create new layout grocery_add_dialog.xml and write the code as follow:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_gravity="center">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:background="@color/blue_shade_1">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Add Items to Cart"
            android:textAllCaps="false"
            android:textAlignment="center"
            android:gravity="center"
            android:textColor="@color/white"
            android:textStyle="bold"
            android:textSize="20sp"
            android:padding="4dp"
            android:layout_margin="4dp"
            android:id="@+id/idTVHeading"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_below="@id/idTVHeading">

            <com.google.android.material.textfield.TextInputLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                style="@style/TextInputEditTextTheme"
                android:layout_margin="5dp"
                android:hint="Enter ITem Name"
                android:padding="5dp"
                android:textColorHint="@color/white"
                app:hintTextColor="@color/white">

                <com.google.android.material.textfield.TextInputEditText
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:id="@+id/idEdtItemName"
                    android:inputType="text"
                    android:textColor="@color/white"
                    android:textColorHint="@color/white"
                    android:textSize="14sp"/>
            </com.google.android.material.textfield.TextInputLayout>

            <com.google.android.material.textfield.TextInputLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                style="@style/TextInputEditTextTheme"
                android:layout_margin="5dp"
                android:hint="Enter ITem Quantity"
                android:padding="5dp"
                android:textColorHint="@color/white"
                app:hintTextColor="@color/white">

                <com.google.android.material.textfield.TextInputEditText
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:id="@+id/idEdtItemQuantity"
                    android:inputType="number"
                    android:textColor="@color/white"
                    android:textColorHint="@color/white"
                    android:textSize="14sp"/>
            </com.google.android.material.textfield.TextInputLayout>

            <com.google.android.material.textfield.TextInputLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                style="@style/TextInputEditTextTheme"
                android:layout_margin="5dp"
                android:hint="Enter ITem Price"
                android:padding="5dp"
                android:textColorHint="@color/white"
                app:hintTextColor="@color/white">

                <com.google.android.material.textfield.TextInputEditText
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:id="@+id/idEdtItemPrice"
                    android:inputType="number"
                    android:textColor="@color/white"
                    android:textColorHint="@color/white"
                    android:textSize="14sp"/>
            </com.google.android.material.textfield.TextInputLayout>


            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal"
                android:layout_margin="5dp"
                android:weightSum="2">

                <Button
```

```
                    android:layout_width="0dp"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    android:layout_margin="8dp"
                    android:id="@+id/idBtnCancel"
                    android:text="Cancel"
                    android:background="@drawable/custom_button_back"
                    android:textAllCaps="false"/>

                <Button
                    android:layout_width="0dp"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    android:layout_margin="8dp"
                    android:id="@+id/idBtnAdd"
                    android:text="Add"
                    android:background="@drawable/custom_button_back"
                    android:textAllCaps="false"/>

            </LinearLayout>

        </LinearLayout>
    </RelativeLayout>

</androidx.cardview.widget.CardView>
```

STEP - 10 : CREATE NEW DRAWABLE FILE

create new drawable file custom_button_back.xml file and write the code as follow :

```
<?xml version="1.0" encoding="utf-8"?>
<shape android:shape="rectangle"
xmlns:android="http://schemas.android.com/apk/res/android">
<corners android:radius="20dp"/>
    <solid android:color="@color/blue_shade_2"/>
</shape>
```

STEP - 11 : CODING ON MainActivity.kt FILE

Finally write the following code in MainActivity.kt file as follow:

```
package com.example.groceryapp

import android.app.Dialog
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.Toast
import androidx.lifecycle.Observer
import androidx.lifecycle.ViewModelProvider
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView
import com.google.android.material.floatingactionbutton.FloatingActionButton

class MainActivity :
AppCompatActivity(),GroceryRVAdapter.GroceryItemClickInterface {
    lateinit var itemsRV:RecyclerView
    lateinit var addFAB:FloatingActionButton
    lateinit var list:List<GroceryItems>
    lateinit var groceryRVAdapter: GroceryRVAdapter
    lateinit var groceryViewModal: GroceryViewModal

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        itemsRV=findViewById(R.id.idRVItems)
        addFAB=findViewById(R.id.idFABAdd)
        list=ArrayList<GroceryItems>()
        groceryRVAdapter= GroceryRVAdapter(list, this)
        itemsRV.layoutManager=LinearLayoutManager(this)
        itemsRV.adapter=groceryRVAdapter
        val groceryRepository=GroceryRepository(GroceryDatabase(this))
        val factory=GroceryViewModalFactory(groceryRepository)
        groceryViewModal=
ViewModelProvider(this,factory).get(GroceryViewModal::class.java)
        groceryViewModal.getAllGroceryItems().observe(this, Observer {
            groceryRVAdapter.list=it
            groceryRVAdapter.notifyDataSetChanged()
        })

        addFAB.setOnClickListener{
            openDialog()
        }
    }

    fun openDialog(){
        val dialog=Dialog(this)
        dialog.setContentView(R.layout.grocery_add_dialog)
        val cancelBtn=dialog.findViewById<Button>(R.id.idBtnCancel)
        val addBtn=dialog.findViewById<Button>(R.id.idBtnAdd)
        val itemEdt=dialog.findViewById<EditText>(R.id.idEdtItemName)
        val itemPriceEdt=dialog.findViewById<EditText>(R.id.idEdtItemPrice)
        val itemQuantityEdt=dialog.findViewById<EditText>(R.id.idEdtItemQuantity)
        cancelBtn.setOnClickListener {
            dialog.dismiss()
        }
        addBtn.setOnClickListener {
            val itemName:String=itemEdt.text.toString()
            val itemPrice:String=itemPriceEdt.text.toString()
            val itemQuantity:String=itemQuantityEdt.text.toString()
            val qty:Int=itemQuantity.toInt()
            val pr:Int=itemPrice.toInt()
            if(itemName.isNotEmpty() && itemPrice.isNotEmpty() && itemQuantity.isNotEmpty())
            {
                val items=GroceryItems(itemName,qty,pr)
                groceryViewModal.insert(items)
                Toast.makeText(applicationContext,"Item Inserted..",Toast.LENGTH_SHORT).show()
                groceryRVAdapter.notifyDataSetChanged()
                dialog.dismiss()
            }
            else{
                Toast.makeText(applicationContext,"Please Enter All The Data",Toast.LENGTH_SHORT).show()
            }
        }
        dialog.show()
    }

    override fun onItemClick(groceryItems: GroceryItems) {
        groceryViewModal.delete(groceryItems)
        groceryRVAdapter.notifyDataSetChanged()
        Toast.makeText(applicationContext,"Item Deleted..",Toast.LENGTH_SHORT).show()
    }
```

```
}
```

STEP - 12 : ADDING ICON

First add icon to the drawable folder.Now add an icon for the app in AndroidManifest.xml as follow:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.groceryapp">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@drawable/grocery_icon"
        android:label="@string/app_name"
        android:roundIcon="@drawable/grocery_icon"
        android:supportsRtl="true"
        android:theme="@style/Theme.GroceryApp"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Now run the app in virtual device/emulator to use it.

Finally the project is completed.