Aim ---> Build A Grocery Android App

<u>Problem Statement</u> --> As we can't remember everything, users frequently forget to buy the things they want to buy. However, with the assistance of this app, you can make a list of the groceries you intend to buy so that you don't forget anything.

<u>Solution</u> -->

Build a grocery application in android using android studio. Many times we forget to purchase things that we want to buy, after all, we can't remember all the items, so with the help of this app, you can note down your grocery items that you are going to purchase, by doing this you can't forget any items that you want to purchase.
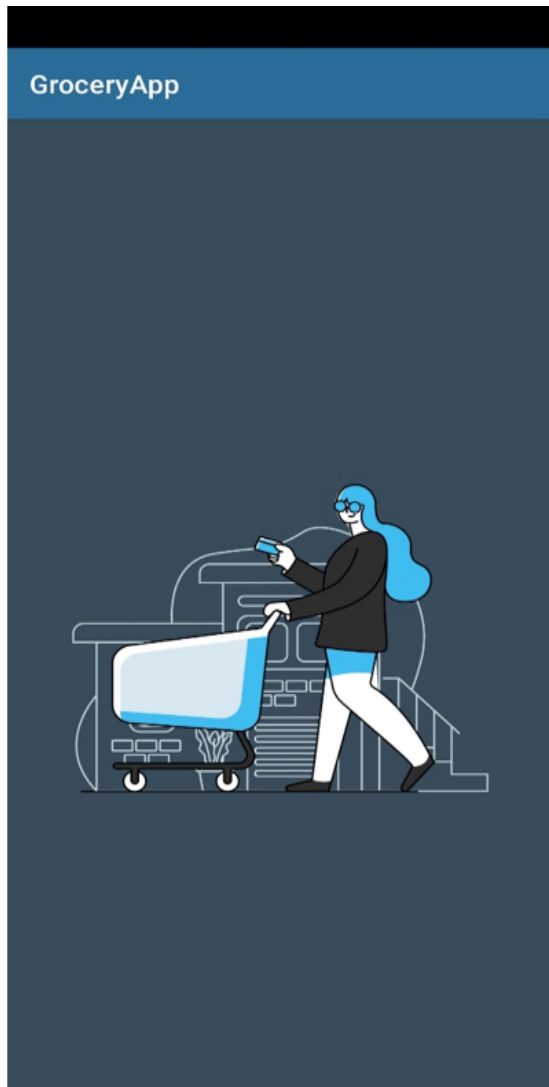
- In this project, we are using MVVM(Model View ViewModel) for architectural patterns, **Room** for database,Coroutines and RecyclerView to display the list of items.
- Language used --- **Kotlin**

<u>GROCERY APP</u>

ICON
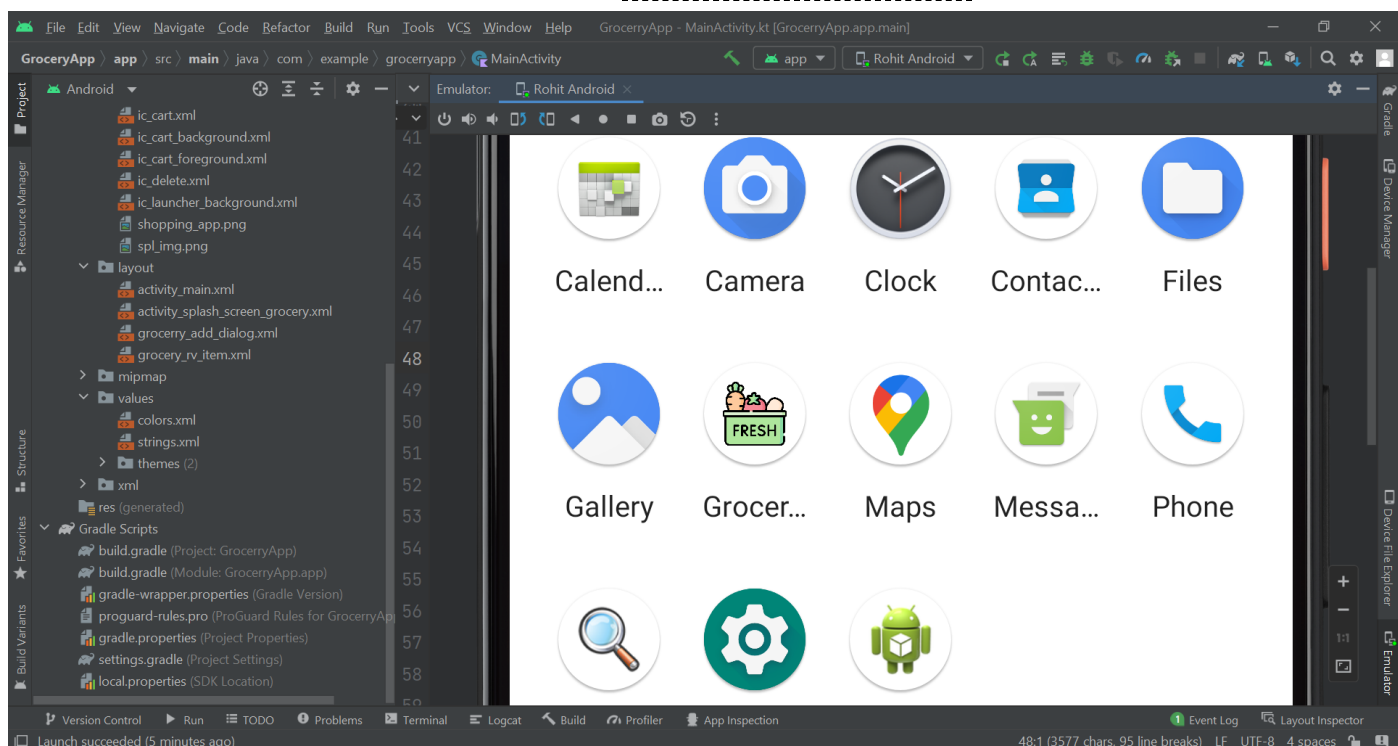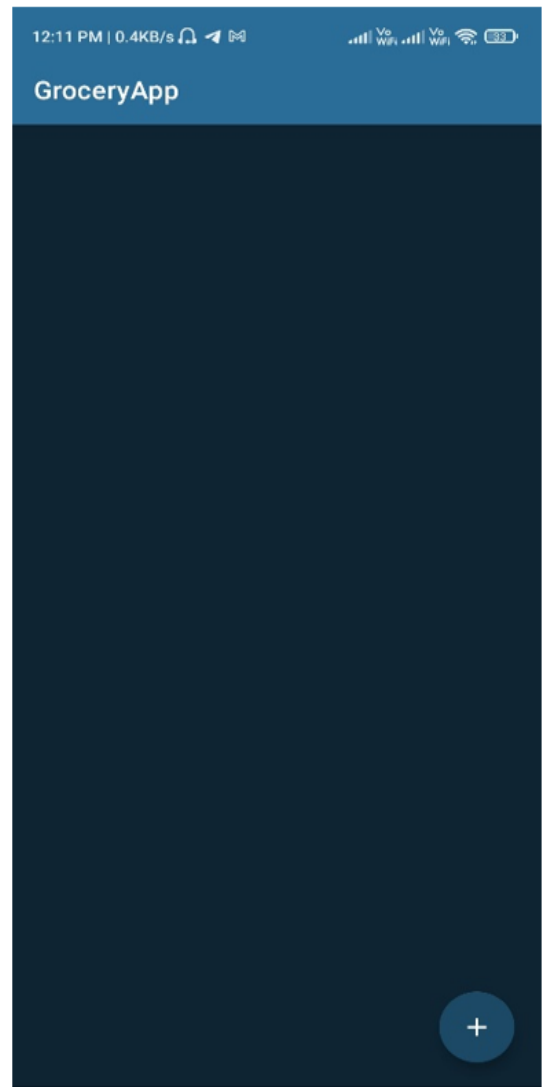
12:11 PM | 0.3KB/s 🎧 ◢ ⋈

GroceryApp

Add Item to Cart

Enter Item Name
Noodles (Family pack)

Enter Item Quantity
2

Enter Item Price
87

Cancel   Add

+

1	2	3	-
4	5	6	␣
7	8	9	⌫
.	0	,	✓

12:11 PM | 0.1KB/s 🎧 ◢ ⋈

GroceryApp

| Noodles (Family pack) | 2 | Rs. 87 | 🗑 |
| Total Cost | | | Rs. 174 |

Item Inserted..

+

1:21 PM | 0.1KB/s

GroceryApp

| Noodles (Family Pack) | 2 | Rs. 87 | 🗑 |
| Total Cost | | | Rs. 174 |

Item Deleted..

➕

1:05 PM | 0.2KB/s

GroceryApp

| Almond kg | 1 | Rs. 950 | 🗑 |
| Total Cost | | | Rs. 950 |

| Noodles (Family Pack) | 2 | Rs. 87 | 🗑 |
| Total Cost | | | Rs. 174 |

| Coffee gms | 250 | Rs. 1 | 🗑 |
| Total Cost | | | Rs. 250 |

| Basmati Rice kg | 5 | Rs. 70 | 🗑 |
| Total Cost | | | Rs. 350 |

| Moong Dal | 3 | Rs. 100 | 🗑 |
| Total Cost | | | Rs. 300 |

| Sugar kg | 5 | Rs. 72 | 🗑 |
| Total Cost | | | Rs. 360 |

| Green Tea | 1 | Rs. 150 | 🗑 |
| Total Cost | | | Rs. 150 |

| Cashew kg | 1 | Rs. 650 | 🗑 |
| Total Cost | | | Rs. 650 |

| Mustard oil litre | 1 | Rs. 220 | |
| Total Cost | | | Rs. 220 |

➕

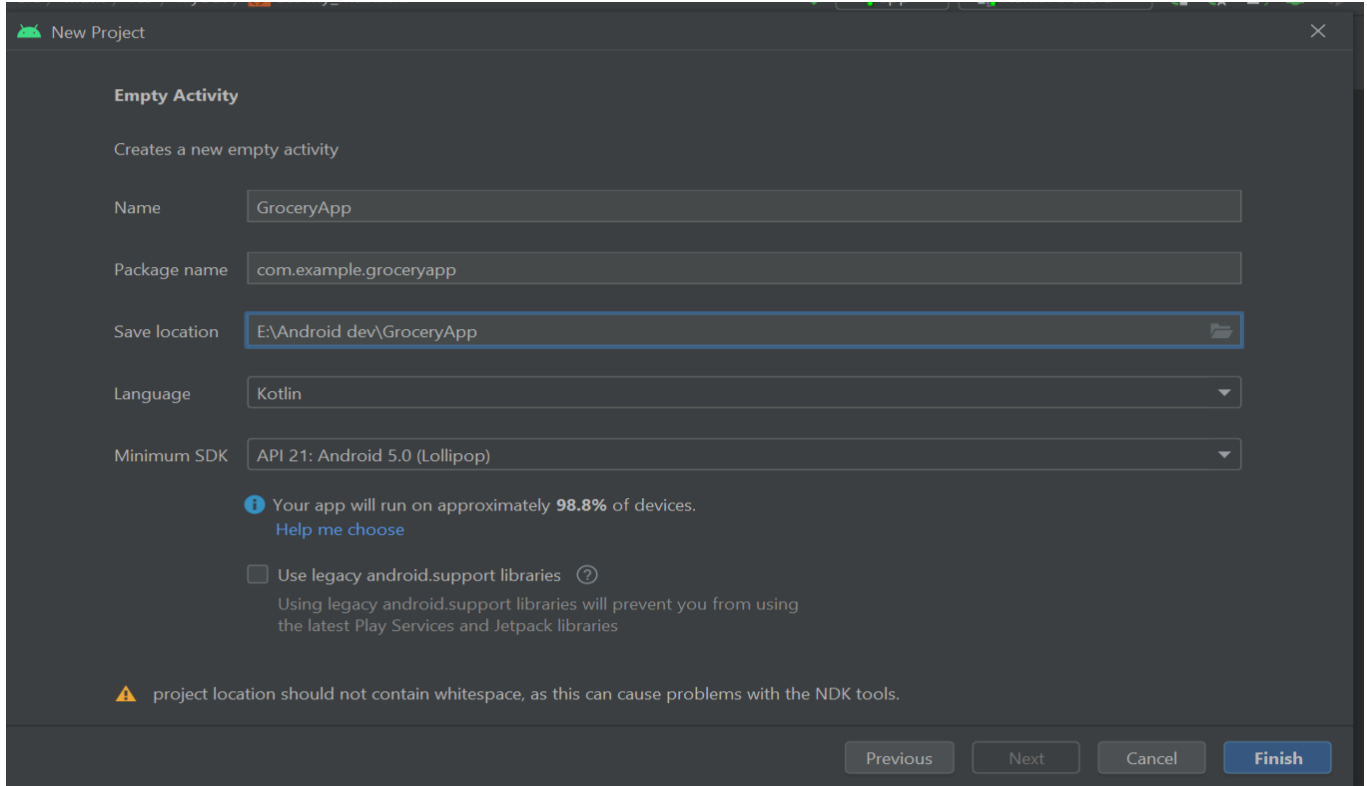*Step by Step Implementation -->*

- **Create a New Projec**t - **Select Kotlin as a programming language**

- **Add libraries in the gradle file and also apply the plugin as 'kotlin-kapt'** To add these library go to **Gradle Scripts > build.gradle(Module:app).**

```
1  plugins {
2      id 'com.android.application'
3      id 'org.jetbrains.kotlin.android'
4      id 'kotlin-kapt'
5  }
6
7  android {
8      compileSdk 32
9
10     defaultConfig {
```

```gradle
11          applicationId "com.example.grocerryapp"
12          minSdk 16
13          targetSdk 32
14          versionCode 1
15          versionName "1.0"
16
17          testInstrumentationRunner
   "androidx.test.runner.AndroidJUnitRunner"
18      }
19
20      buildTypes {
21          release {
22              minifyEnabled false
23              proguardFiles getDefaultProguardFile('proguard-android-

24          }
25      }
26      compileOptions {
27          sourceCompatibility JavaVersion.VERSION_1_8
28          targetCompatibility JavaVersion.VERSION_1_8
29      }
30      packagingOptions {
31          exclude 'META-INF/atomicfu.kotlin_module'
32      }
33      kotlinOptions {
34          jvmTarget = '1.8'
35      }
36 }
37
38 dependencies {
39      implementation "androidx.appcompat:appcompat:1.5.1"
40      implementation "androidx.activity:activity-ktx:1.5.1"
41
42      // Dependencies for working with Architecture components
43      // You'll probably have to update the version numbers in build.gradle
   (Project)
44
45      // Room components
46      implementation "androidx.room:room-ktx:2.4.3"
47      kapt "androidx.room:room-compiler:2.4.3"
```

```
48      androidTestImplementation "androidx.room:room-testing:2.4.3"
49
50      // Lifecycle components
51      implementation "androidx.lifecycle:lifecycle-viewmodel-ktx:2.5.1"
52      implementation "androidx.lifecycle:lifecycle-livedata-ktx:2.5.1"
53      implementation "androidx.lifecycle:lifecycle-common-java8:2.5.1"
54
55      // Kotlin components
56 //     implementation 'org.jetbrains.kotlin:kotlin-stdlib-jdk7:1.7.10'
57      api "org.jetbrains.kotlinx:kotlinx-coroutines-
   core:$rootProject.coroutines"
58      api "org.jetbrains.kotlinx:kotlinx-coroutines-
   android:$rootProject.coroutines"
59
60      // UI
61      implementation "androidx.constraintlayout:constraintlayout:2.1.4"
62      implementation "com.google.android.material:material:1.6.1"
63
64      // Testing
65      testImplementation "junit:junit:$rootProject.junitVersion"
66      androidTestImplementation "androidx.arch.core:core-
   testing:$rootProject.coreTestingVersion"
67      androidTestImplementation ("androidx.test.espresso:espresso-

68          exclude group: 'com.android.support', module: 'support-
   annotations'
69      })
70      androidTestImplementation
   "androidx.test.ext:junit:$rootProject.androidxJunitVersion"
71 }
```

➤ _Implement room database_

a) Entities class

The entities class contains all the columns in the database and

it should be annotated with @Entity(tablename = "Name of table")

```kotlin
1  package com.example.grocerryapp
2
3  import androidx.room.ColumnInfo
4  import androidx.room.Entity
5  import androidx.room.PrimaryKey
6
7  @Entity(tableName = "grocery_items")
8  data class GroceryItems (
9      @ColumnInfo(name = "itemName")
10     var itemName : String,
11
12
13     @ColumnInfo(name = "itemQuantity")
14     var itemQuantity : Int,
15
16     @ColumnInfo(name = "itemPrice")
17     var itemPrice : Int,
18
19 ){
20
21     @PrimaryKey(autoGenerate = true)
22     var id:Int? = null
23 }
```

b) Dao Interface

The Dao is an interface in which we create all the functions that we want to implement on the database.

```kotlin
1  package com.example.grocerryapp
2
3
4  import androidx.lifecycle.LiveData
```

```
5  import androidx.room.*
6
7  @Dao
8  interface GroceryDao {
9
10     @Insert(onConflict = OnConflictStrategy.REPLACE)
11     suspend fun insert(item: GroceryItems)
12
13     @Delete
14     suspend fun delete(item : GroceryItems)
15
16     @Query("SELECT * FROM grocery_items")
17     fun gwtAllGroceryItem() : LiveData<List<GroceryItems>>
18 }
```

c) Database class

Database class annotated with @Database(entities = [Name of Entity class.class], version = 1) these entities are the entities array list all the data entities associating with the database and version shows the current version of the database. This database class inherits from the Room Database class.

```
1  package com.example.grocerryapp
2
3  import android.content.Context
4  import androidx.room.Database
5  import androidx.room.Room
6  import androidx.room.RoomDatabase
7
8  @Database(entities = [GroceryItems::class],
```

```kotlin
    version = 1)
9 abstract class GroceryDatabase : RoomDatabase(){
10
11      abstract fun getGroceryDao() : GroceryDao
12
13      companion object{
14          @Volatile
15          private  var instance : GroceryDatabase ?
 = null
16          private val  LOCK = Any()
17
18          operator fun invoke(context: Context) =
 instance?: synchronized(LOCK){
19              instance ?:
 createDatabase(context).also {
20                  instance = it
21              }
22          }
23          private fun createDatabase(context:
 Context)=
24              Room.databaseBuilder(
25                  context.applicationContext,
26                  GroceryDatabase::class.java,
27                  "Grocery.db"
28              ).build()
29      }
30 }
```

➤ Now we will implement the architectural structure in the app

a) Repository class

The repository is one of the design structures. The repository class gives the data to the ViewModel class and then the ViewModel class uses that data for Views. The repository will choose the appropriate data locally or on the network

```kotlin
1 package com.example.grocerryapp
2
3 data class GroceryRepository(private val db:
  GroceryDatabase){
4
5     suspend fun insert (items : GroceryItems) =
  db.getGroceryDao().insert(items)
6     suspend fun delete (items: GroceryItems) =
  db.getGroceryDao().delete(items)
7
8     fun getAllItems() =
  db.getGroceryDao().gwtAllGroceryItem()
9 }
```

b) ViewModel class

ViewModel class used as an interface between View and Data. Grocery View Model class inherit from View Model class and we will pass constructor value by creating instance variable of Repository class and

stored in repository variable. As we pass the constructor in View Model we have to create another class which is a Factory View Model class.

```kotlin
1  package com.example.grocerryapp
2
3  import androidx.lifecycle.ViewModel
4  import kotlinx.coroutines.GlobalScope
5  import kotlinx.coroutines.launch
6
7  class GroceryViewModel(private val repository:
   GroceryRepository) : ViewModel() {
8
9      fun insert(items: GroceryItems) = GlobalScope.launch
   {
10          repository.insert(items)
11      }
12      fun delete(items: GroceryItems) = GlobalScope.launch
   {
13          repository.delete(items)
14      }
15      fun getAllGroceryItems() = repository.getAllItems()
16  }
```

c) Factory ViewModel class

We will inherit the Grocery ViewModel Factory class from ViewModelProvider.NewInstanceFactory and again pass constructor value by creating instance variable of Grocery Repository and return GroceryViewModel(repository)

```kotlin
1  package com.example.grocerryapp
2
3  import androidx.lifecycle.ViewModel
```

```
 4 import androidx.lifecycle.ViewModelProvider
 5
 6 class GroceryViewModalFactory (private val repository:
   GroceryRepository):
   ViewModelProvider.NewInstanceFactory(){
 7     override fun <T: ViewModel> create(modelClass:
   Class<T>):T{
 8         return GroceryViewModel(repository) as T
 9     }
10 }
```

➤ Now let's jump into the UI part

- **activity_main.xml -** Add RecyclerView and FloatingActionButton after clicking this button a **DialogBox** open and in that dialog box user can enter the item name, item quantity, and item price

```
 1 <?xml version="1.0" encoding="utf-8"?>
 2 <RelativeLayout
 3     xmlns:android="http://schemas.android.com/apk/res/android"
 4     xmlns:app="http://schemas.android.com/apk/res-auto"
 5     xmlns:tools="http://schemas.android.com/tools"
 6     android:background="@color/blue_shade_1"
 7     android:layout_width="match_parent"
 8     android:layout_height="match_parent"
 9     tools:context=".MainActivity">
10
11     <androidx.recyclerview.widget.RecyclerView
12         android:id="@+id/idRVItems"
13         android:layout_width="match_parent"
14         android:layout_height="wrap_content"
15         android:layout_alignTop="@+id/textMain"
16         tools:listitem="@layout/grocery_rv_item" />
17
```

```
18
   <com.google.android.material.floatingactionbutton.FloatingActionBut
   ton
19            android:id="@+id/idFABAdd"
20            android:layout_width="wrap_content"
21            android:layout_height="wrap_content"
22            android:layout_alignParentEnd="true"
23            android:layout_alignParentBottom="true"
24            android:layout_margin="20dp"
25            app:backgroundTint="@color/blue_shade_2"
26            android:layout_alignParentRight="true"
27            android:src="@drawable/ic_add"
28            app:tint="@color/white"/>
29
30 </RelativeLayout>
```
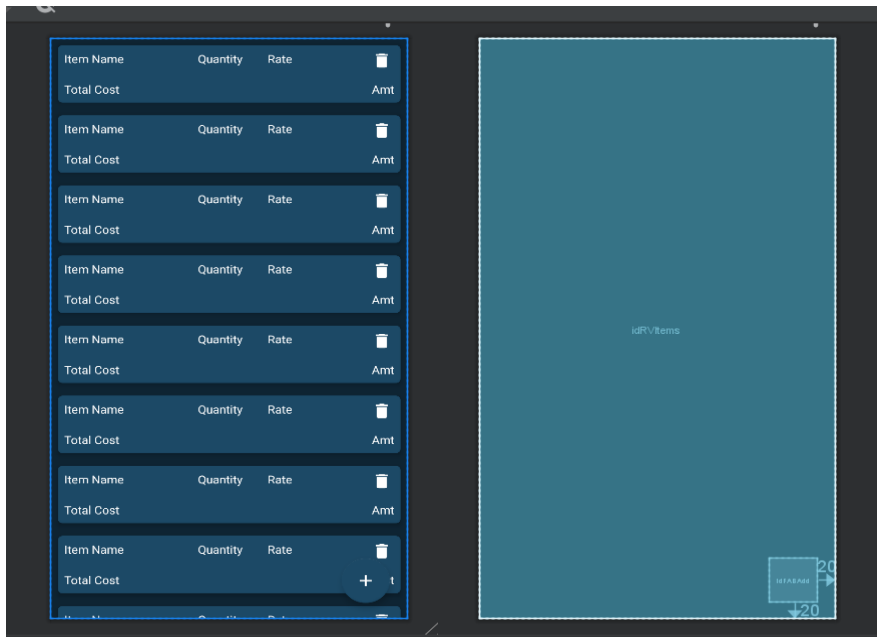

- **RecyclerView --** Now we will code the UI part of the row in the list.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.cardview.widget.CardView
```

```xml
    xmlns:android="http://schemas.android.com/apk/res/android"
3       android:layout_width="match_parent"
4       android:layout_height="wrap_content"
5       xmlns:app="http://schemas.android.com/apk/res-auto"
6       android:layout_margin="8dp"
7       android:backgroundTint="@color/blue_shade_2"
8       app:cardCornerRadius="5dp"
9       app:cardElevation="4dp">
10
11      <RelativeLayout
12          android:layout_width="match_parent"
13          android:layout_height="wrap_content">
14
15          <LinearLayout
16              android:id="@+id/idLL1"
17              android:layout_width="match_parent"
18              android:layout_height="wrap_content"
19              android:orientation="horizontal"
20              android:weightSum="5">
21
22              <TextView
23                  android:layout_width="0dp"
24                  android:layout_height="wrap_content"
25                  android:layout_weight="2"
26                  android:text="@string/item_name"
27                  android:textColor="@color/white"
28                  android:padding="4dp"
29                  android:layout_margin="3dp"
30                  android:id="@+id/idTVItemName"/>
31
32              <TextView
33                  android:layout_width="0dp"
34                  android:layout_height="wrap_content"
35                  android:layout_weight="1"
36                  android:text="@string/itemQuantity"
37                  android:textColor="@color/white"
38                  android:padding="4dp"
```

```xml
39              android:layout_margin="3dp"
40              android:id="@+id/idTVQuantity"/>
41
42          <TextView
43              android:layout_width="0dp"
44              android:layout_height="wrap_content"
45              android:layout_weight="1.5"
46              android:text="@string/itemRate"
47              android:textColor="@color/white"
48              android:padding="4dp"
49              android:layout_margin="3dp"
50              android:id="@+id/idTVRate"/>
51
52          <ImageView
53              android:layout_width="0dp"
54              android:layout_height="wrap_content"
55              android:layout_weight="0.5"
56              android:padding="4dp"
57              android:layout_margin="3dp"
58              android:src="@drawable/ic_delete"
59              app:tint="@color/white"
60              android:id="@+id/idTVDelete"/>
61      </LinearLayout>
62
63      <TextView
64          android:layout_width="match_parent"
65          android:layout_height="wrap_content"
66          android:text="Total Cost"
67          android:layout_below="@+id/idLL1"
68          android:textColor="@color/white"
69          android:layout_margin="3dp"
70          android:padding="4dp"
71          android:id="@+id/idTVHeading"
72          android:layout_toLeftOf="@id/idTVTotalAmt"/>
73
74      <TextView
```

```
75              android:layout_width="wrap_content"
76              android:layout_height="wrap_content"
77              android:text="Amt"
78              android:layout_below="@id/idLL1"
79              android:textColor="@color/white"
80              android:layout_margin="3dp"
81              android:padding="4dp"
82              android:id="@+id/idTVTotalAmt"
83              android:layout_alignParentEnd="true"
84              android:layout_alignParentRight="true"/>
85
86      </RelativeLayout>
87
88 </androidx.cardview.widget.CardView>
```
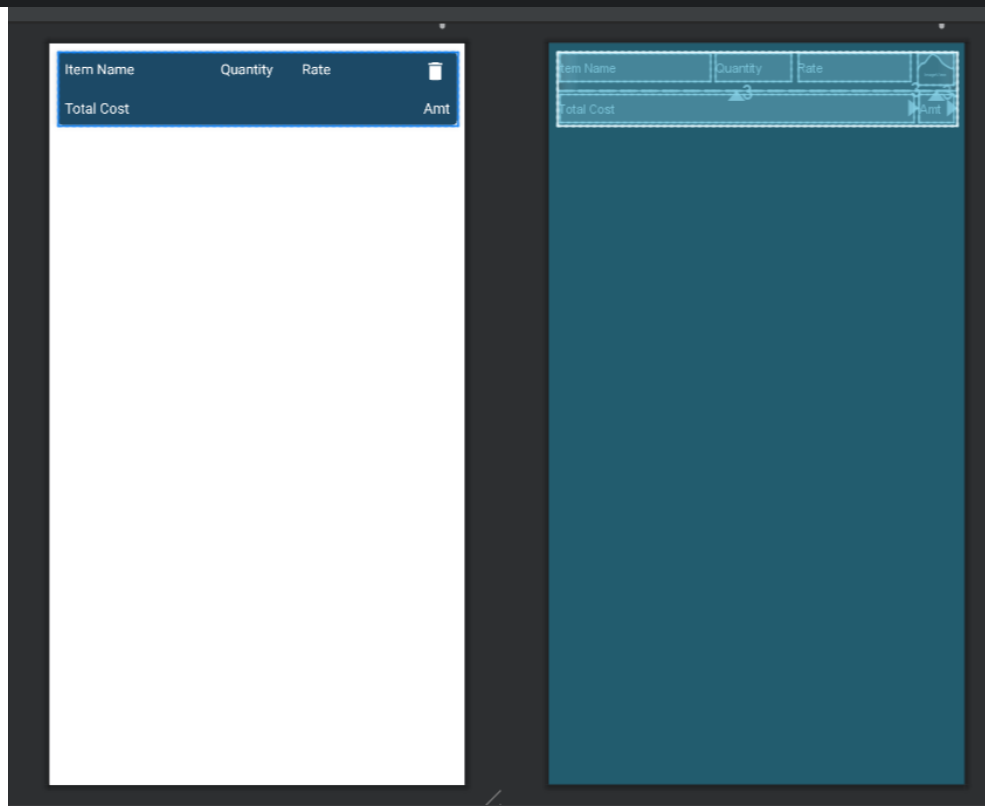

Adapter class for recycler view.

```
1 package com.example.grocerryapp
2
```

```kotlin
3 import android.view.LayoutInflater
4 import android.view.View
5 import android.view.ViewGroup
6 import android.widget.ImageView
7 import android.widget.TextView
8 import androidx.recyclerview.widget.RecyclerView
9
10 class GroceryRVAdapter(
11     var list: List<GroceryItems>,
12     val groceryItemClickInterface :
  GroceryItemClickInterface)
13     :
  RecyclerView.Adapter<GroceryRVAdapter.GroceryView

14
15     inner class GroceryViewHolder(itemView: View)
  : RecyclerView.ViewHolder(itemView){
16         val nameTV =
  itemView.findViewById<TextView>(R.id.idTVItemNam

17         val quantityTV =
  itemView.findViewById<TextView>(R.id.idTVQuantit

18         val rateTV =
  itemView.findViewById<TextView>(R.id.idTVRate)
19         val amountTV =
  itemView.findViewById<TextView>(R.id.idTVTotalAm
```

```kotlin
20        val deleteTV =
   itemView.findViewById<ImageView>(R.id.idTVDelete)
21
22     }
23
24     interface GroceryItemClickInterface{
25         fun onItemClick(groceryItems:
   GroceryItems)
26     }
27
28     override fun onCreateViewHolder(parent:
   ViewGroup, viewType: Int): GroceryViewHolder {
29         val view =
   LayoutInflater.from(parent.context).inflate(R.lay

30         ,false)
31         return GroceryViewHolder(view)
32     }
33
34     override fun onBindViewHolder(holder:
   GroceryViewHolder, position: Int) {
35         holder.nameTV.text =
   list.get(position).itemName
36         holder.quantityTV.text =
   list.get(position).itemQuantity.toString()
37         holder.rateTV.text = "Rs.
```

```kotlin
tring()
38          val itemTotal : Int =
    list.get(position).itemPrice *
    list.get(position).itemQuantity
39          holder.amountTV.text = "Rs.

40          holder.deleteTV.setOnClickListener {
41
    groceryItemClickInterface.onItemClick(list.get(po

42          }
43
44      }
45      override fun getItemCount(): Int {
46          return list.size
47      }
48 }
```

- To enter grocery item, quantity, and price from the user we have to create an interface. To implement this interface we will use DialogBox. **First create UI of dialog box**

```xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.cardview.widget.CardView
  xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="wrap_content"
5     xmlns:app="http://schemas.android.com/apk/res-auto"
6     android:layout_gravity="center">
7
```

```xml
    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/blue_shade_1">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Add Item 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
```

```
43                          android:layout_width="match_parent"
44                          android:layout_height="wrap_content"
45                          android:id="@+id/idEditItemName"
46                          android:inputType="text"
47                          android:textSize="14sp"
48                          android:textColorHint="@color/white"
49                          android:textColor="@color/white"/>
50
51
52
   </com.google.android.material.textfield.TextInputLayout>
53
54
55              <com.google.android.material.textfield.TextInputLayout
56                  android:layout_width="match_parent"
57                  android:layout_height="wrap_content"
58                  style="@style/TextInputEditTextTheme"
59                  android:layout_margin="5dp"
60                  android:hint="Enter Item Quantity"
61                  android:padding="5dp"
62                  android:textColorHint="@color/white"
63                  app:hintTextColor="@color/white">
64
65
   <com.google.android.material.textfield.TextInputEditText
66                          android:layout_width="match_parent"
67                          android:layout_height="wrap_content"
68                          android:id="@+id/idEditItemQuantity"
69                          android:inputType="number"
70                          android:textSize="14sp"
71                          android:textColorHint="@color/white"
72                          android:textColor="@color/white"/>
73
74
75
   </com.google.android.material.textfield.TextInputLayout>
```

```xml
76
77
                <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">
87
88
    <com.google.android.material.textfield.TextInputEditText
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content"
                        android:id="@+id/idEditItemPrice"
                        android:inputType="number"
                        android:textSize="14sp"
                        android:textColorHint="@color/white"
                        android:textColor="@color/white"/>
96
97
    </com.google.android.material.textfield.TextInputLayout>
98
                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:orientation="horizontal"
                    android:layout_margin="5dp"
                    android:weightSum="2">
105
                    <Button
                        android:layout_width="0dp"
                        android:layout_height="wrap_content"
                        android:layout_weight="1"
```

```
110                         android:layout_margin="8dp"
111                         android:id="@+id/idBtnCancle"
112                         android:text="Cancel"
113
     android:background="@drawable/custom_button_back"
114                         android:textAllCaps="false"/>
115
116                     <Button
117                         android:layout_width="0dp"
118                         android:layout_height="wrap_content"
119                         android:layout_weight="1"
120                         android:layout_margin="8dp"
121                         android:id="@+id/idBtnAdd"
122                         android:text="Add"
123
     android:background="@drawable/custom_button_back"
124                         android:textAllCaps="false"/>
125
126                 </LinearLayout>
127             </LinearLayout>
128         </RelativeLayout>
129
130   </androidx.cardview.widget.CardView>
```
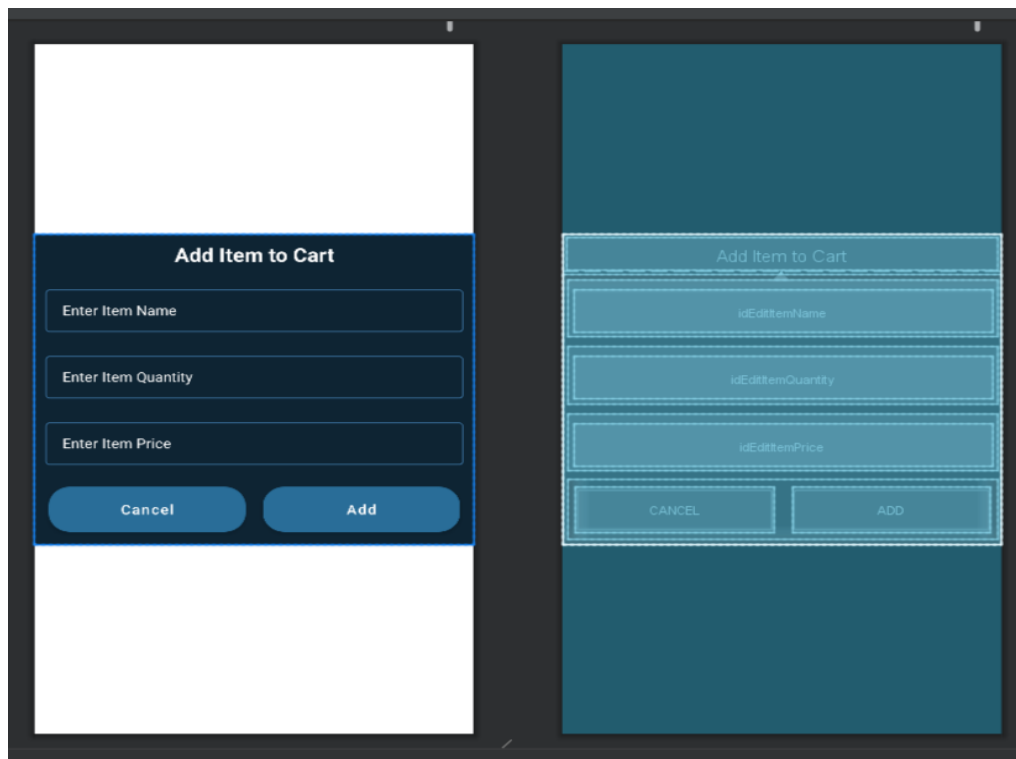
- **Splash Screen**

```xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.constraintlayout.widget.ConstraintLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:background="@color/mirage"
8      android:layout_gravity="center"
9      tools:context=".SplashScreenGrocery">
10
11     <ImageView
12         android:id="@+id/imageView3"
13         android:layout_width="330dp"
14         android:layout_height="345dp"
15         android:src="@drawable/spl_img"
16         app:layout_constraintBottom_toBottomOf="parent"
```

```
17            app:layout_constraintEnd_toEndOf="parent"
18            app:layout_constraintHorizontal_bias="0.493"
19            app:layout_constraintStart_toStartOf="parent"
20            app:layout_constraintTop_toTopOf="parent" />
21
22 </androidx.constraintlayout.widget.ConstraintLayout>
```

```kotlin
 1 package com.example.grocerryapp
 2
 3 import android.content.Intent
 4 import androidx.appcompat.app.AppCompatActivity
 5 import android.os.Bundle
 6 import android.os.Handler
 7 import android.view.WindowManager
 8
 9
10 @Suppress("DEPRECATION")
11 class SplashScreenGrocery : AppCompatActivity() {
12     override fun onCreate(savedInstanceState: Bundle?) {
13         super.onCreate(savedInstanceState)
14         setContentView(R.layout.activity_splash_screen_grocery)
15
16         window.setFlags(
17             WindowManager.LayoutParams.FLAG_FULLSCREEN,
18             WindowManager.LayoutParams.FLAG_FULLSCREEN
19         )
20         Handler().postDelayed({
21             val intent = Intent(this, MainActivity::class.java)
22             startActivity(intent)
23             finish()
24         }, 4000)
25     }
26 }
```

- **In this step finally we will code in our MainActivity. In**

our main activity, we have to set up the recycler view and add click listener on add button to open the dialog box.

```
1  package com.example.grocerryapp
2
3  import android.animation.Animator
4  import android.app.Dialog
5  import androidx.appcompat.app.AppCompatActivity
6  import android.os.Bundle
7  import android.view.View
8  import android.widget.Button
9  import android.widget.EditText
10 import android.widget.Toast
11 import androidx.lifecycle.Observer
12 import androidx.lifecycle.ViewModelProvider
13 import androidx.recyclerview.widget.LinearLayoutManager
14 import androidx.recyclerview.widget.RecyclerView
15 import
   com.google.android.material.floatingactionbutton.FloatingActio
   nButton
16
17 class MainActivity : AppCompatActivity(),
   GroceryRVAdapter.GroceryItemClickInterface {
18     lateinit var itemsRv : RecyclerView
19     lateinit var addFAB : FloatingActionButton
20     lateinit var list: List<GroceryItems>
21     lateinit var groceryRVAdapter: GroceryRVAdapter
22     lateinit var groceryViewModel: GroceryViewModel
23
24     override fun onCreate(savedInstanceState: Bundle?) {
25         super.onCreate(savedInstanceState)
26         setContentView(R.layout.activity_main)
27
28         itemsRv = findViewById(R.id.idRVItems)
```

```kotlin
29            addFAB = findViewById(R.id.idFABAdd)
30
31            list = ArrayList<GroceryItems>()
32            groceryRVAdapter = GroceryRVAdapter(list,this)
33            itemsRv.layoutManager = LinearLayoutManager(this)
34            itemsRv.adapter = groceryRVAdapter
35
36            val groceryRepository =
    GroceryRepository(GroceryDatabase(this))
37            val factory =
    GroceryViewModalFactory(groceryRepository)
38            groceryViewModel =
    ViewModelProvider(this,factory).get(GroceryViewModel::class.ja

39            groceryViewModel.getAllGroceryItems().observe(this,
    Observer{
40                groceryRVAdapter.list = it
41                groceryRVAdapter.notifyDataSetChanged()
42            })
43
44            addFAB.setOnClickListener {
45                openDialog()
46            }
47        }
48
49      fun openDialog(){
50            val dialog = Dialog(this)
51            dialog.setContentView(R.layout.grocerry_add_dialog)
52            val cancelBtn =
    dialog.findViewById<Button>(R.id.idBtnCancle)
53            val addBtn =
    dialog.findViewById<Button>(R.id.idBtnAdd)
54            val itemEdt =
    dialog.findViewById<EditText>(R.id.idEditItemName)
55            val itemQuantityEdt =
```

```kotlin
        dialog.findViewById<EditText>(R.id.idEditItemQuantity)
56          val itemPriceEdt =
    dialog.findViewById<EditText>(R.id.idEditItemPrice)
57
58          cancelBtn.setOnClickListener{
59              dialog.dismiss()
60          }
61          addBtn.setOnClickListener {
62              val itemName : String = itemEdt.text.toString()
63              val itemPrice : String =
    itemPriceEdt.text.toString()
64              val itemQuantity : String =
    itemQuantityEdt.text.toString()
65
66              val qty : Int = itemQuantity.toInt()
67              val pr : Int = itemPrice.toInt()
68
69              if(itemName.isNotEmpty() && itemPrice.isNotEmpty()
    && itemQuantity.isNotEmpty()){
70                  val items = GroceryItems(itemName,qty,pr)
71                  groceryViewModel.insert(items)
72                  Toast.makeText(applicationContext, "Item

73                  groceryRVAdapter.notifyDataSetChanged()
74                  dialog.dismiss()
75              }else{
76                  Toast.makeText(
77                      applicationContext,
78                      "Please Enter all the data..",
79                      Toast.LENGTH_SHORT
80                  ).show()
81              }
82          }
83
84          dialog.show()
85      }
```

```
86    override fun onItemClick(groceryItems: GroceryItems) {
87        groceryViewModel.delete(groceryItems)
88        groceryRVAdapter.notifyDataSetChanged()
89        Toast.makeText(applicationContext,"Item

90    }
91 }
```

➤ **This is how the complete project structure looks
like.**

File Edit View Navigate Code Refactor Build Run Tools

Android ▾

✔ **app**
  › manifests
  ✔ java
    ✔ com.example.grocerryapp
      GroceryDao
      GroceryDatabase
      GroceryItems
      GroceryRepository
      GroceryRVAdapter
      GroceryViewModalFactory
      GroceryViewModel
      MainActivity
      SplashScreenGrocery
    › com.example.grocerryapp (androidTest)
    › com.example.grocerryapp (test)
  › java (generated)
  ✔ res
    ✔ drawable
      custom_button_back.xml
      groceries.png
      ic_add.xml
      ic_cart.xml
      ic_cart_background.xml
      ic_cart_foreground.xml
      ic_delete.xml
      ic_launcher_background.xml
      shopping_app.png
      spl_img.png

41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

⑂ Version Control ▶ Run ≡ TODO ❶ Problems ⌷ Termi

Launch succeeded (a minute ago)

File Edit View Navigate Code Refactor Build Run Tools

Android

ic_cart.xml
ic_cart_background.xml
ic_cart_foreground.xml
ic_delete.xml
ic_launcher_background.xml
shopping_app.png
spl_img.png
layout
    activity_main.xml
    activity_splash_screen_grocery.xml
    grocerry_add_dialog.xml
    grocery_rv_item.xml
mipmap
values
    colors.xml
    strings.xml
    themes (2)
xml
res (generated)
Gradle Scripts
    build.gradle (Project: GroceryApp)
    build.gradle (Module: GrocerryApp.app)
    gradle-wrapper.properties (Gradle Version)
    proguard-rules.pro (ProGuard Rules for GrocerryAp|
    gradle.properties (Project Properties)
    settings.gradle (Project Settings)
    local.properties (SDK Location)

41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58

Version Control   Run   TODO   Problems   Term

Launch succeeded (2 minutes ago)