

Project Report (Grocery App).

● Introduction :

1. It is an application for listing the grocery Items using MVVM Architecture and Room Data Base.
2. By this application we can easily list our necessary grocery Items without writing with Pen and Paper.
3. You can Directly access the Total Price of an Individual Items.
4. It Does not Require Internet , you can Directly add Items by clicking on add button in Offline Mode.

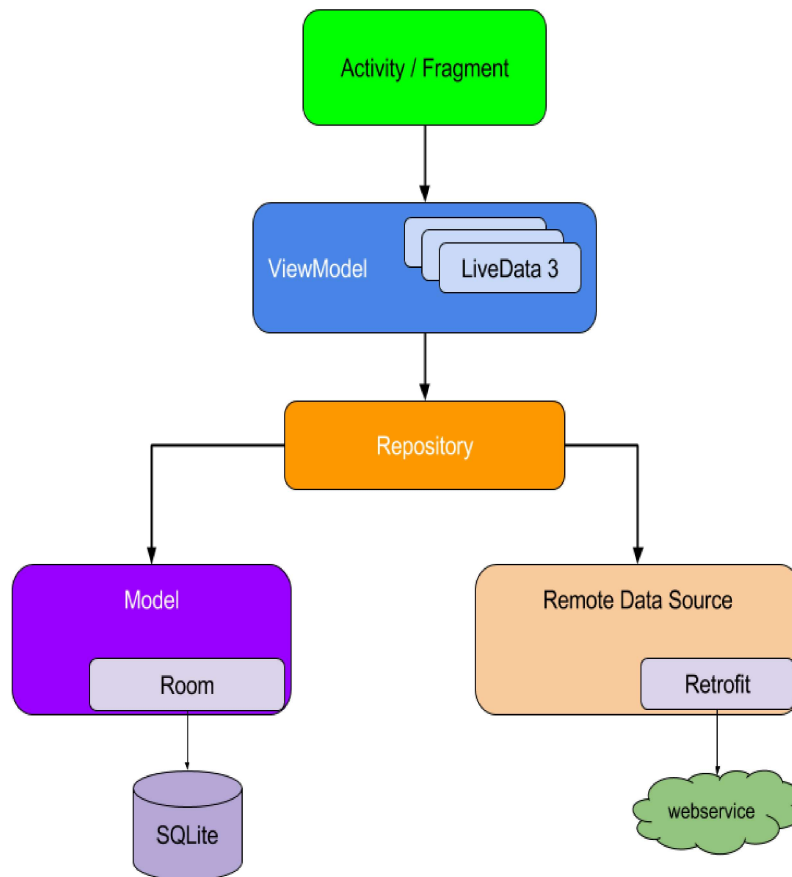
● Purpose and Literature Survey:

The Purpose of the Grocery app is to list Your items on apps without having a Physical list and their is a chances of getting mistake in Total while writing on Paper. You can directly add Items on Offline mode. Sometimes we missed the Physical item list So we directly seen items on our app and Get Automatic Total of the Items Individually.

● Experimental Analysis:

1. We use MVVM Architecture in this Application
2. Code used, as well as layers are well organized and are named precisely for the user to interpret properly. Code which is being used to model the application is composed of today's mobile trend. Moreover, one can get free icons, fonts and updates for user's assistance.
3. We have taken help from online sources such as GeeksForGeeks and also some from Youtube and

also from Google Developer website.



- The separate code layers of **MVVM** are:

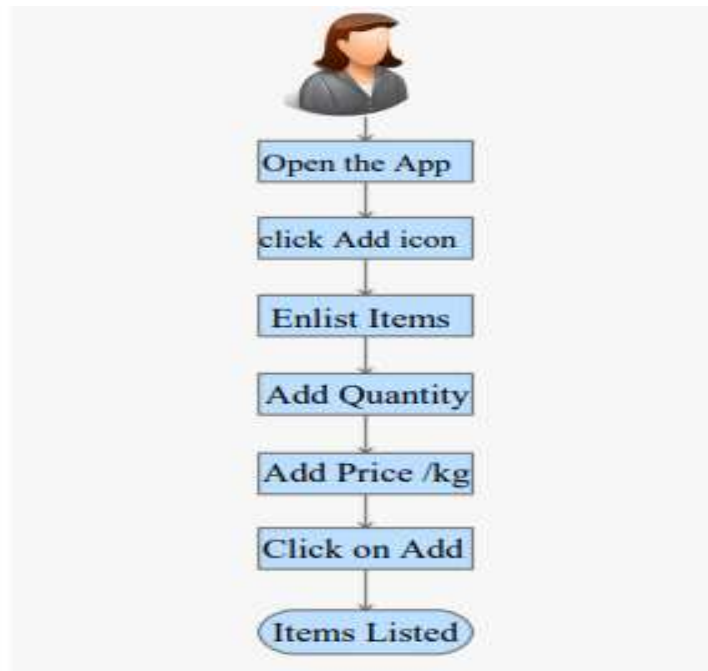
Model: This layer handles the abstraction of the data sources. Model and ViewModel work together to get and save the data.

View: The purpose of this layer is to inform the ViewModel about the user's action. This layer observes the ViewModel and does not contain any kind of application logic.

ViewModel: It exposes those data streams which are relevant to the View. Moreover, it serve as a link between the Model and the View.

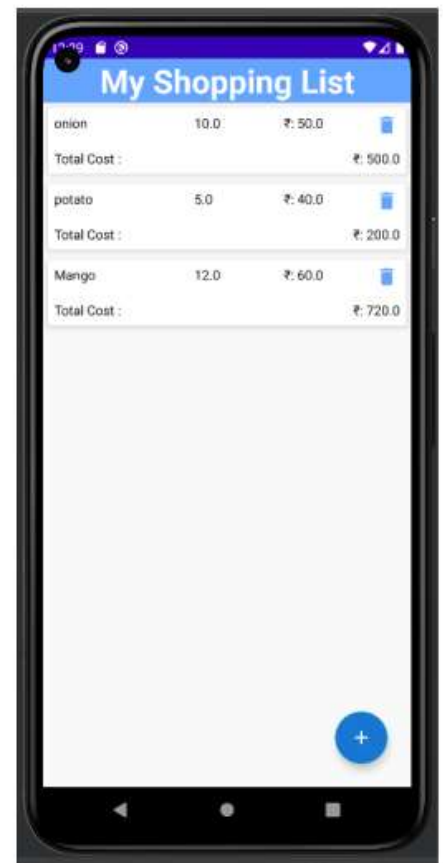
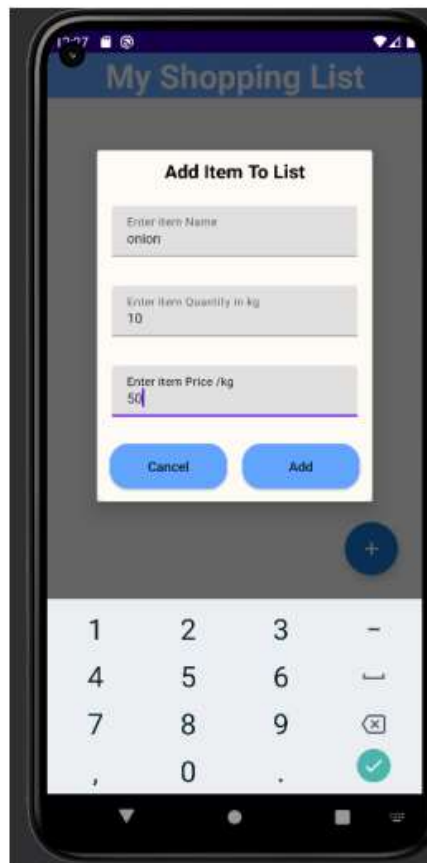
• Flow chart:

Graphical Representation of the Grocery application is Here:



● Result:

The Output of the grocery application is:



● Advantages and Disadvantages:

1. It reduces Human efforts and chances of mistake.
2. It does not need any Internet connections you can access or add items in offline mode.
3. You can directly add Items or delete when not required.
4. It is Not a Proper app for Ordering the Grocery Items.
5. Its Does not differentiate Fruits , vegetables and other things.

● Conclusions and Applications:

You can directly use this app while Shopping and it is also easier for giving payment because it gives the total amount rapidly.

It reduces human efforts and it is directly work on offline mode which make it very suitable and efficient for use. You can also add items On notepad but this app have an advantage over that this provide Total calculation rapidly. Data is not vanished until we delete it.

● Appendix:(Source code)

1. MainActivity:

```
package com.omveer.panwar.GroceryApp
import android.annotation.SuppressLint
import android.app.Dialog
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.EditText
import android.widget.Toast
import androidx.appcompat.widget.AppCompatButton
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 itemRV: RecyclerView
    lateinit var addFAB: FloatingActionButton
    lateinit var list: List<GroceryItems>
    lateinit var groceryRVAdapter: GroceryRVAdapter
    lateinit var groceryViewModel: GroceryViewModel
    @SuppressLint("NotifyDataSetChanged")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        itemRV = findViewById(R.id.rvitems)
        addFAB = findViewById(R.id.fabAdd)
        list = ArrayList<GroceryItems>()
        groceryRVAdapter = GroceryRVAdapter(list, this)
        itemRV.layoutManager = LinearLayoutManager(this)
        itemRV.adapter = groceryRVAdapter
        val groceryRepository = GroceryRepository(GroceryDatabase(this))
        val factory = GroceryViewModelFactory(groceryRepository)
        groceryViewModel = ViewModelProvider(this, factory).get(GroceryViewModel::class.java)
        groceryViewModel.getAllGroceryItems().observe(this) {
            groceryRVAdapter.list = it
            groceryRVAdapter.notifyDataSetChanged()
        }
        addFAB.setOnClickListener{
            openDialog()
        }
    }
    @SuppressLint("NotifyDataSetChanged")
    fun openDialog(){
        val dialog = Dialog(this)
        dialog.setContentView(R.layout.grocery_add_dialog)
        val cancelbtn = dialog.findViewById<AppCompatActivity>(R.id.idbtncancel)
        val addbtn = dialog.findViewById<AppCompatActivity>(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 : Double = itemquantity.toDouble()
    val pr : Double = itemprice.toDouble()
    if (itemname.isNotEmpty() && itemprice.isNotEmpty() && itemquantity.isNotEmpty()){
        val items = GroceryItems(itemname,qty,pr)
        groceryViewModel.insert(items)
        Toast.makeText(applicationContext,"Item Added", Toast.LENGTH_SHORT).show()
        groceryRVAdapter.notifyDataSetChanged()
        dialog.dismiss()
    }
    else{
        Toast.makeText(applicationContext,"Please fill all details", Toast.LENGTH_SHORT).show()
    }
}
dialog.show()
}
@SuppressLint("NotifyDataSetChanged")
override fun onItemClick(groceryItems: GroceryItems) {
    groceryViewModel.delete(groceryItems)
    groceryRVAdapter.notifyDataSetChanged()
    Toast.makeText(applicationContext,"Item Deleted Successfully..", Toast.LENGTH_SHORT).show()
}
}

```

2.Grocery.Dao

```

package com.omveer.panwar.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("SELECT * FROM Grocery_items")

```

```
fun getAllGroceryItems(): LiveData<List<GroceryItems>>
{
```

3. GroceryDatabase

```
package com.omveer.panwar.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,
            "GroceryApp.db"
        ).build()
}
```

4. GroceryItems

```
package com.omveer.panwar.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:Double,

    @ColumnInfo(name = "itemPrice")
```

```

        var itemPrice:Double,
    )
    {
        @PrimaryKey(autoGenerate = true)
        var id:Int?=null
    }

```

5. GroceryRepository

```
package com.omveer.panwar.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()
}

```

7.GroceryRVAdapter

```
package com.omveer.panwar.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.idtvitemname)
        val quantityTV = itemView.findViewById<TextView>(R.id.idtvquantity)
        val rateTV = itemView.findViewById<TextView>(R.id.idtvrate)
        val totalTV = itemView.findViewById<TextView>(R.id.idtvtotalamount)
        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_items,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 = "₹: "+list.get(position).itemPrice.toString()

        val itemTotal :Double = list.get(position).itemQuantity * list.get(position).itemPrice
        holder.totalTV.text = "₹: "+itemTotal.toString()
        holder.deleteIV.setOnClickListener {
            groceryItemClickInterface.onItemClick(list.get(position))
        }
    }
}

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

```

8.GroceryRVAdapter

```
package com.omveer.panwar.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.idtvitemname)
        val quantityTV = itemView.findViewById<TextView>(R.id.idtvquantity)
        val rateTV = itemView.findViewById<TextView>(R.id.idtvrates)
        val totalTV = itemView.findViewById<TextView>(R.id.idtvtotalamount)
        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_items,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 = "₹: "+list.get(position).itemPrice.toString()
        val itemTotal :Double = list.get(position).itemQuantity * list.get(position).itemPrice
        holder.totalTV.text = "₹: "+itemTotal.toString()
        holder.deleteIV.setOnClickListener {
            groceryItemClickInterface.onItemClick(list.get(position))
        }
    }
}

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

```

9. GroceryViewModel

```

package com.omveer.panwar.GroceryApp
import androidx.lifecycle.ViewModel
import kotlinx.coroutines.GlobalScope
import kotlinx.coroutines.launch

```

```

class GroceryViewModel(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()
}

```

10. GroceryViewModelFactory

```

package com.omveer.panwar.GroceryApp

```

```

import androidx.lifecycle.ViewModelimport androidx.lifecycle.ViewModelProvider
class GroceryViewModelFactory(private val repository: GroceryRepository): ViewModelProvider.NewInstanceFactory() {
    override fun <T : ViewModel> create(modelClass: Class<T>): T {
        return GroceryViewModel(repository) as T
    }
}

```

11. ActivityMain.xml

```

<?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="#F9F9F9"
    android:layout_height="match_parent"

```

```
tools:context=".MainActivity">
```

```
<androidx.cardview.widget.CardView
    android:id="@+id/header"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentStart="true"
    android:layout_marginStart="0dp"
    android:layout_marginLeft="0dp"
    android:layout_marginRight="0dp"
    android:layout_marginBottom="10dp"
    android:background="@drawable/header_bg"
    android:backgroundTint="@color/blue_light">
```

```
<TextView
    android:layout_width="414dp"
    android:layout_height="match_parent"
    android:gravity="center"
    android:text="My Shopping List"
    android:textAlignment="center"
    android:textColor="@color/white"
    android:textSize="35sp"
    android:textStyle="bold" />
</androidx.cardview.widget.CardView>
```

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/rvitems"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@id/header"
    tools:listitem="@layout/grocery_rv_items">
```

```
</androidx.recyclerview.widget.RecyclerView>
```

```
<com.google.android.material.floatingactionbutton.FloatingActionButton
    android:id="@+id/fabAdd"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_marginStart="25dp"
    android:layout_marginTop="25dp"
    android:layout_marginEnd="25dp"
    android:layout_marginBottom="25dp"
    android:backgroundTint="@color/blue"
    android:elevation="5dp"
    android:src="@drawable/ic_baseline_add_24"
    app:tint="@color/white" />
```

</RelativeLayout>

12. groceryadd.xml

<?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"

android:backgroundTint="#FEF6F6"

android:background="@drawable/custom_button_back"

app:cardCornerRadius="10dp">

<RelativeLayout

android:layout_width="match_parent"

android:layout_height="wrap_content"

>

<TextView

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:text="Add Item To List"

android:textColor="@color/black"

android:textAllCaps="false"

android:textStyle="bold"

android:gravity="center"

android:textSize="20sp"

android:textAlignment="center"

android:padding="4dp"

android:layout_margin="4dp"

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

<LinearLayout

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:layout_below="@id/idtvHeading"

android:orientation="vertical">

<com.google.android.material.textfield.TextInputLayout

android:layout_width="match_parent"

android:layout_height="wrap_content"

style="@style/TextInputEditTextTheme"

android:layout_margin="10dp"

android:padding="5dp"

app:hintTextColor="@color/black"

app:counterTextColor="@color/black">

<com.google.android.material.textfield.TextInputEditText

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:id="@+id/idEdititemname"

android:textSize="14sp"

```
        android:hint="Enter item Name"
        android:inputType="text"
        app:hintTextColor="@color/black"
        app:counterTextColor="@color/black"
    />
</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="10dp"
    android:padding="5dp"

    app:hintTextColor="@color/black"
    app:counterTextColor="@color/black">
<com.google.android.material.textfield.TextInputEditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/idEdititemquantity"
    android:textSize="14sp"
    android:hint="Enter item Quantity in kg"
    android:inputType="numberDecimal"
    app:hintTextColor="@color/black"
    app:counterTextColor="@color/black"
/>
</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="10dp"
    android:padding="5dp"
    app:hintTextColor="@color/black"
    app:counterTextColor="@color/black">
<com.google.android.material.textfield.TextInputEditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/idEdititemprice"
    android:textSize="14sp"
    android:hint="Enter item Price /kg"
    android:inputType="numberDecimal"
    app:hintTextColor="@color/black"
    app:counterTextColor="@color/black"
/>
</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">
<androidx.appcompat.widget.AppCompatButton
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:layout_margin="8dp"
    android:padding="5dp"
    android:id="@+id/idbtncancel"
    android:text="Cancel"
    android:background="@drawable/custom_button_back"
    android:textAllCaps="false"
/>
<androidx.appcompat.widget.AppCompatButton
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:layout_margin="8dp"
    android:padding="5dp"
    android:id="@+id/idbtnadd"
    android:text="Add"
    android:background="@drawable/custom_button_back"
    android:textAllCaps="false"
/>
</LinearLayout>
</LinearLayout>
</RelativeLayout>
</androidx.cardview.widget.CardView>

```

13. groceryvitems.xml

```

<?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="5dp"
    android:backgroundTint="@color/white"
    app:cardCornerRadius="3dp"
    app:cardElevation="5dp">
    <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/black"
    android:padding="4dp"
    android:layout_margin="3dp"
    android:id="@+id/idthvitemname"/>
<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1.25"
    android:text="Quantity"
    android:textColor="@color/black"
    android:padding="4dp"
    android:layout_margin="3dp"
    android:id="@+id/idthvquantity"/>
<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1.25"
    android:text="Rate"
    android:textColor="@color/black"
    android:padding="4dp"
    android:layout_margin="3dp"
    android:id="@+id/idthvrate"/>
<ImageView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.5"
    android:id="@+id/idthvdelete"
    android:padding="4dp"
    android:src="@drawable/ic_baseline_delete_24"
    app:tint="@color/blue_light"
    android:layout_margin="3dp"/>
```

```
</LinearLayout>

<TextView
    android:id="@+id/idthvheading"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/idLL1"
    android:layout_marginStart="3dp"
    android:layout_marginTop="3dp"
    android:layout_marginEnd="3dp"
    android:layout_marginBottom="3dp"
    android:layout_toLeftOf="@id/idthvtotalamount"
    android:padding="4dp"
    android:text="Total Cost :"
    android:textColor="@color/black" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="4dp"
    android:layout_margin="3dp"
    android:id="@+id/idthvtotalamount"
    android:layout_below="@id/idLL1"
    android:text="Amount"
    android:layout_alignParentEnd="true"
    android:textColor="@color/black"/>

</RelativeLayout>

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

SUBMITTED BY:

OMVEER PANWAR.

GIT-HUB link - <https://github.com/Omveer99>

App Link - <https://github.com/smartinternz02/Sl-GuidedProject-68058-1662454541>

Google Developer Profile- https://g.dev/omveer_panwar

