

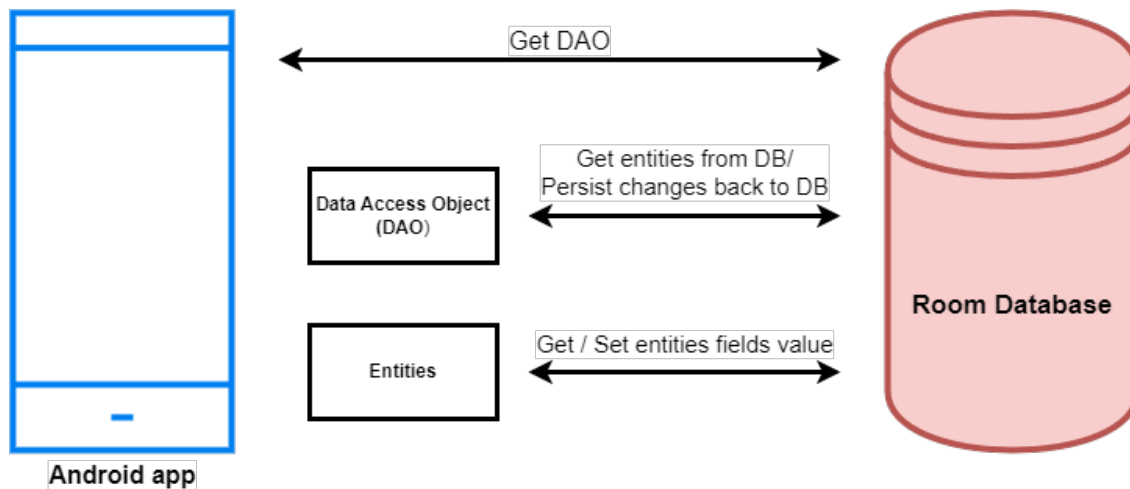


Money Matters: A Personal Finance Management
App
Project Based Experiential Learning Program

Money Matters: A Personal Finance Management App

The app allows user to keep track of their expenses and accounts, and provides an overview of their financial status. Users can set a budget for various expenses and view their progress towards it.

Architecture



Learning Outcomes :

By end of this project:

- You'll be able to work on Android studio and build an app.
- You'll be able to integrate the database accordingly.

Project Workflow:

- Users register into the application.
- After registration , user logs into the application.
- User enters into the main page

Tasks:

- 1.Required initial steps
- 2.Creating a new project.
- 3.Adding required dependencies.
- 4.Creating the database classes.

5. Building application UI and connecting to database.

6. Using AndroidManifest.xml 7. Running the application.

Task 1:

Required initial steps :

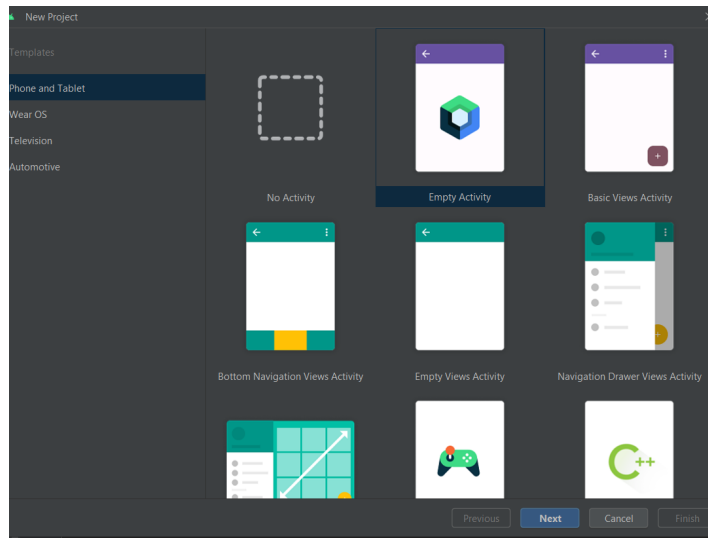
<https://developer.android.com/studio/install>

Task 2 :

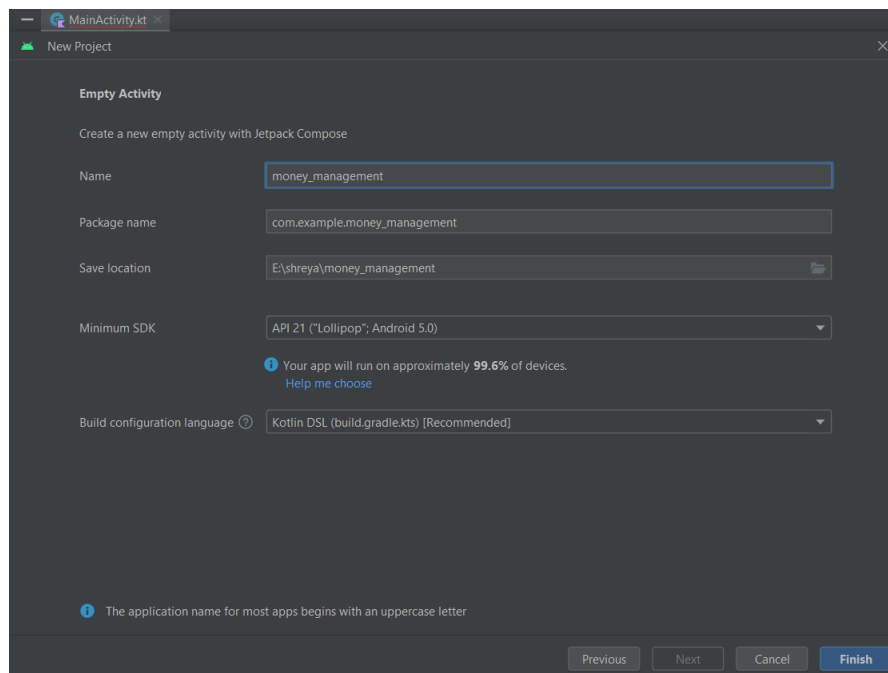
Creating a new project.

Step 1 : Android studio > File > New > New Project > Empty Compose Activity Step

2 : Click on **Next** button.



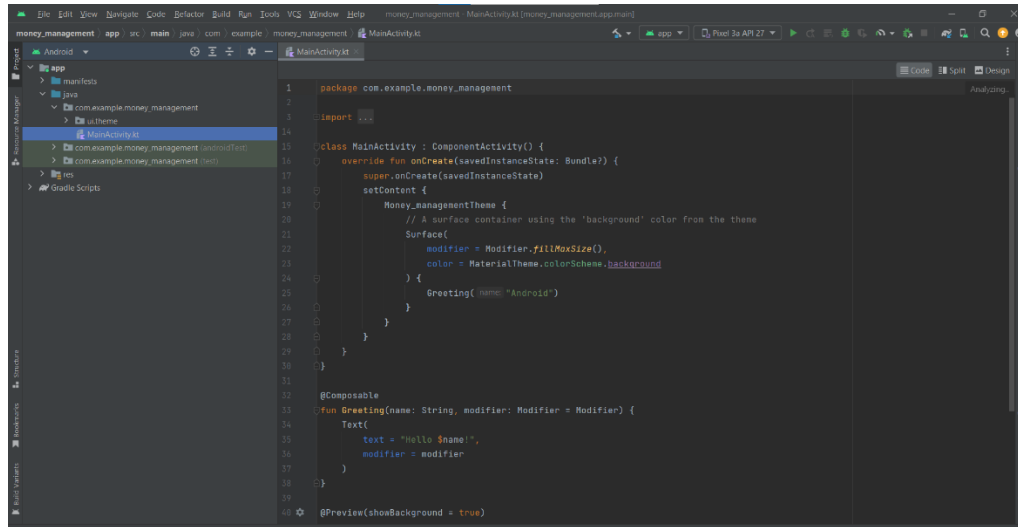
Step 3 : Give name to the new project.



Step 4 : Give the Minimum SDK value

Step 5 : Click Finish

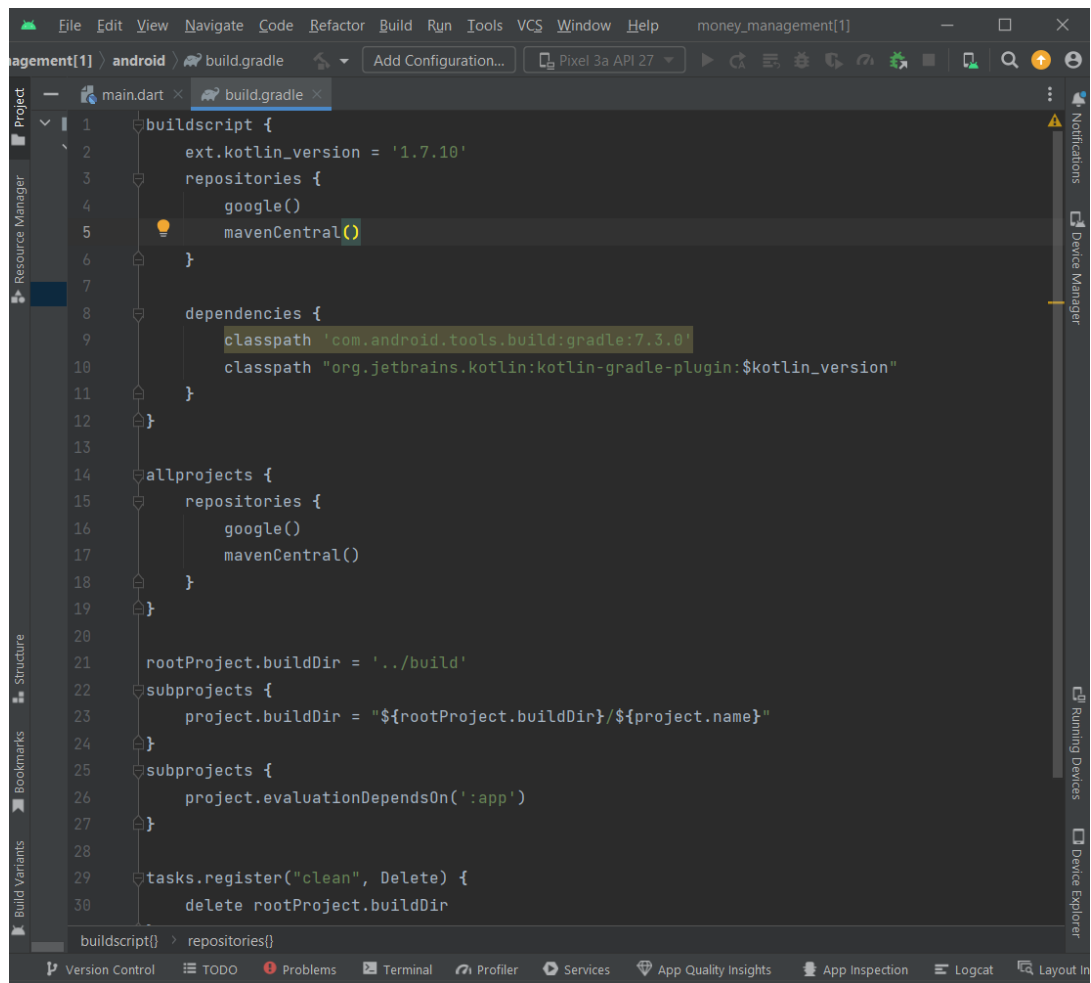
MainActivity.kt



Task 3 :

Adding required dependencies.

Step 1 : Gradle scripts > build.gradle(Module :app)

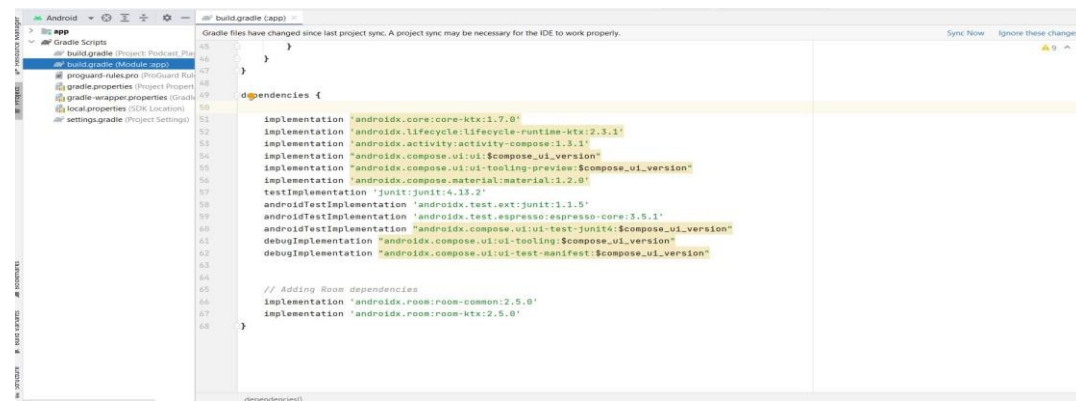


Step 2 : Adding room dependencies. Add the below code in dependencies

```

// Adding Room dependencies implementation
'androidx.room:room-common:2.5.0'
implementation 'androidx.room:room-ktx:2.5.0'

```

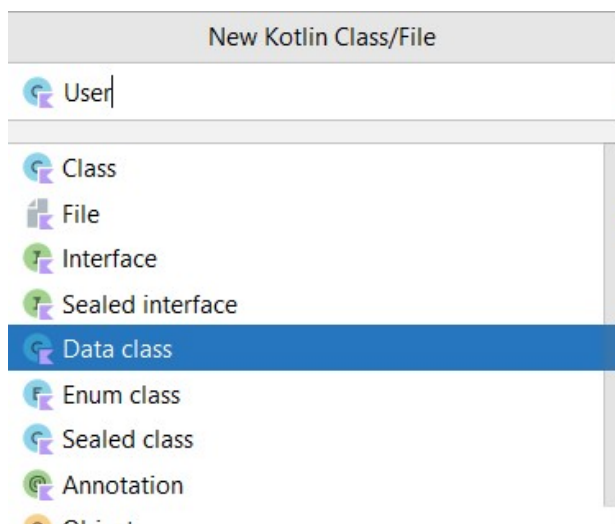
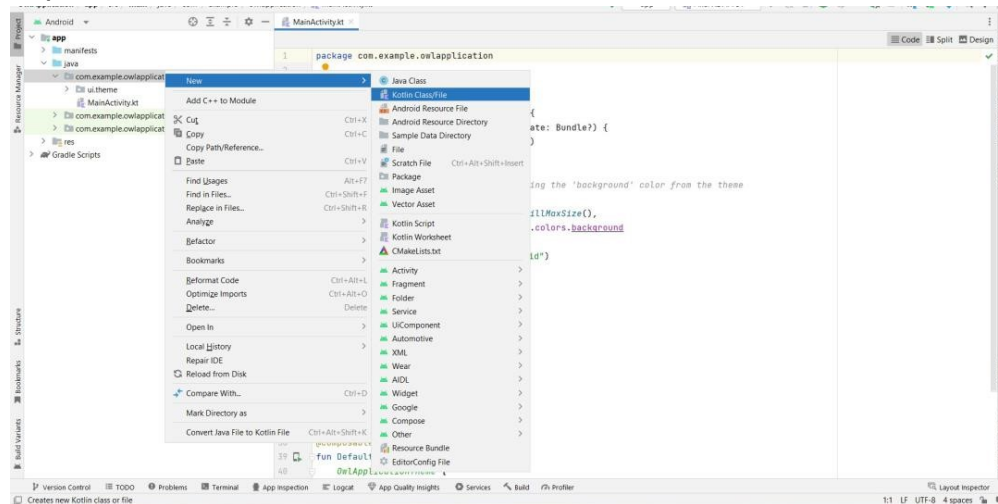


Step 3 : Click on Sync now

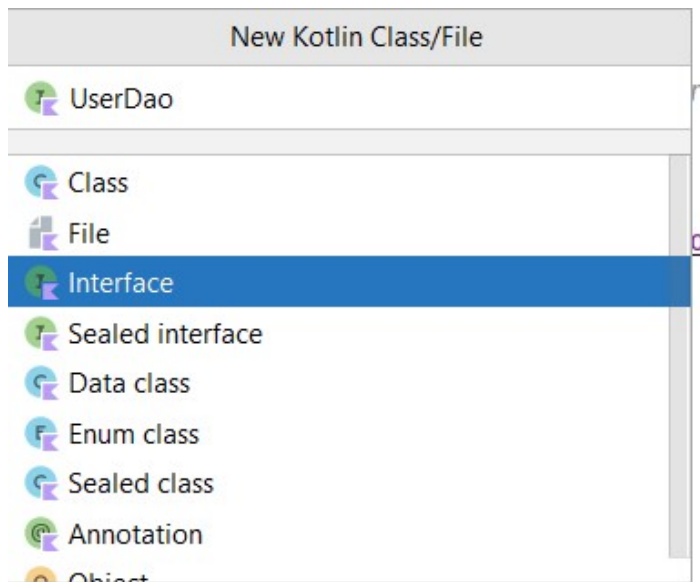
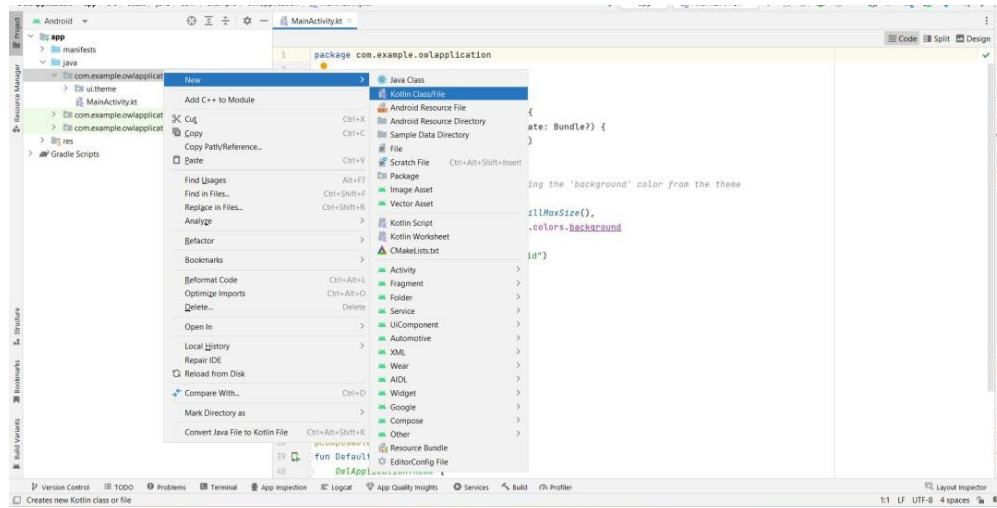
Task 4:

1. Creating the database classes for user login and registration.

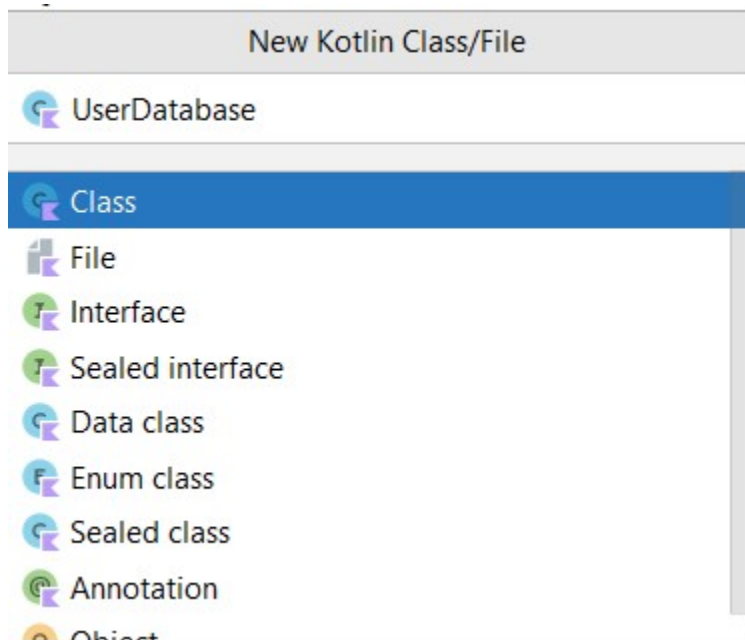
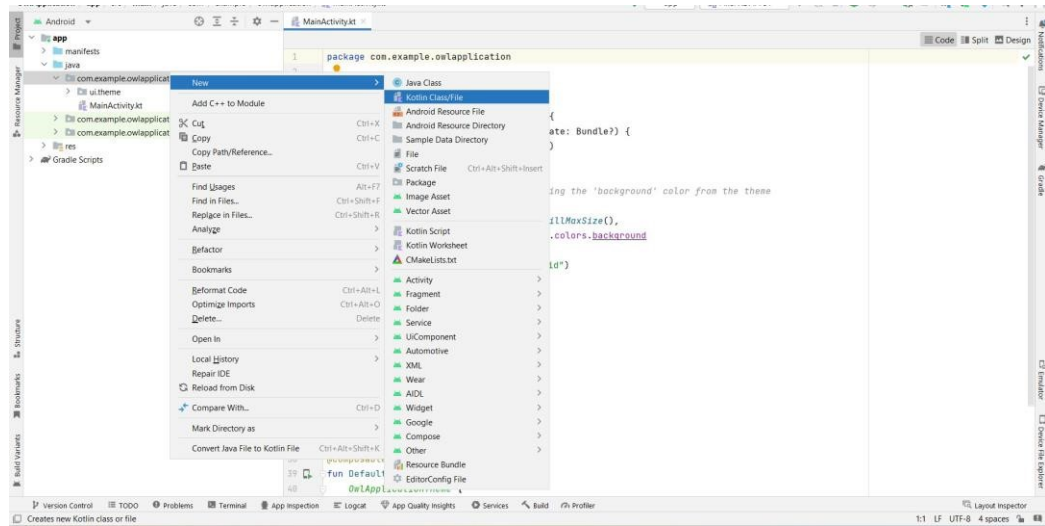
Step 1 : Create User data class



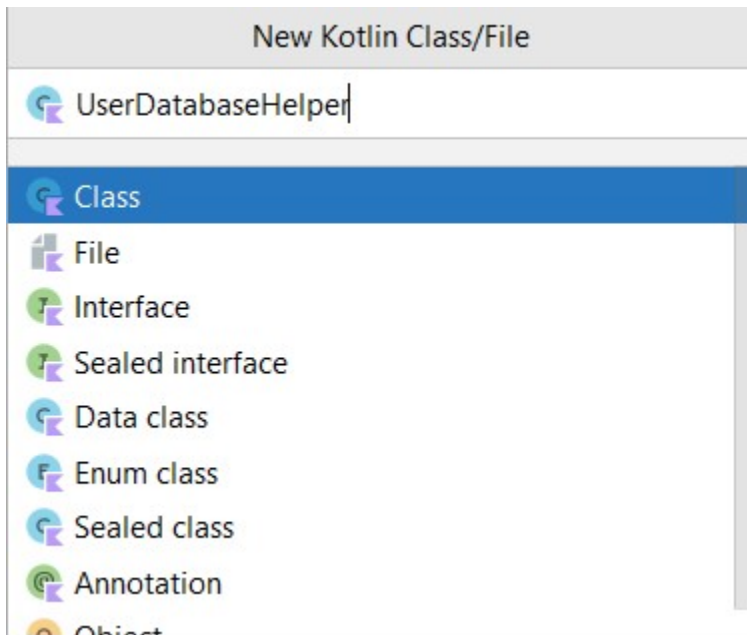
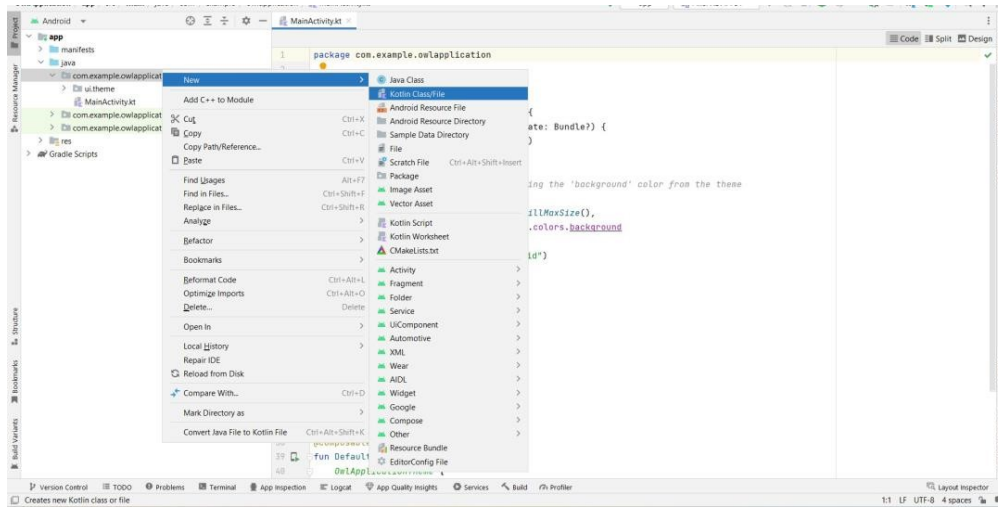
Step 2 : Create an UserDao interface



Step 3 : Create an UserDatabase class

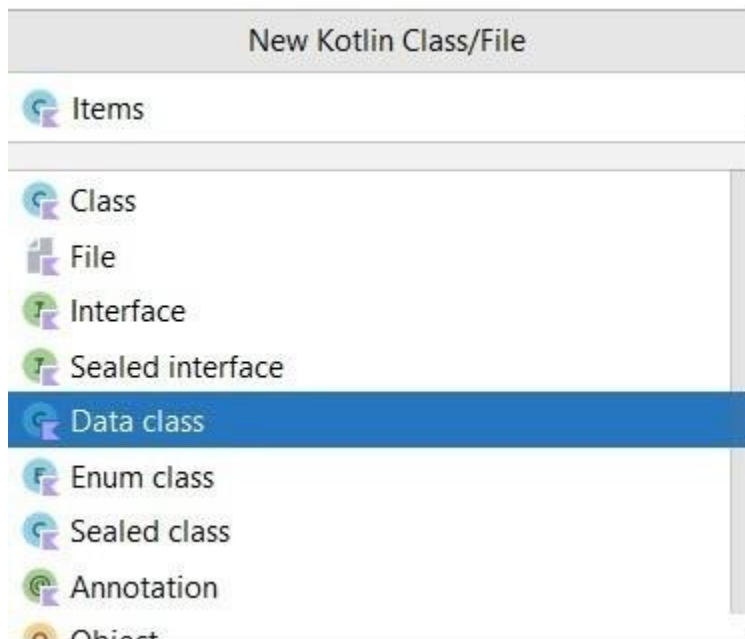
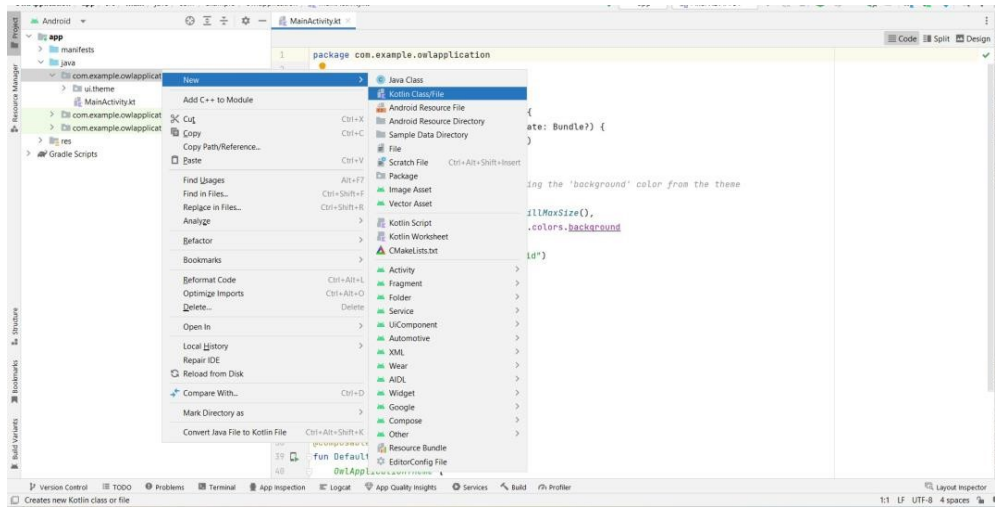


Step 4 : Create an UserDatabaseHelper class

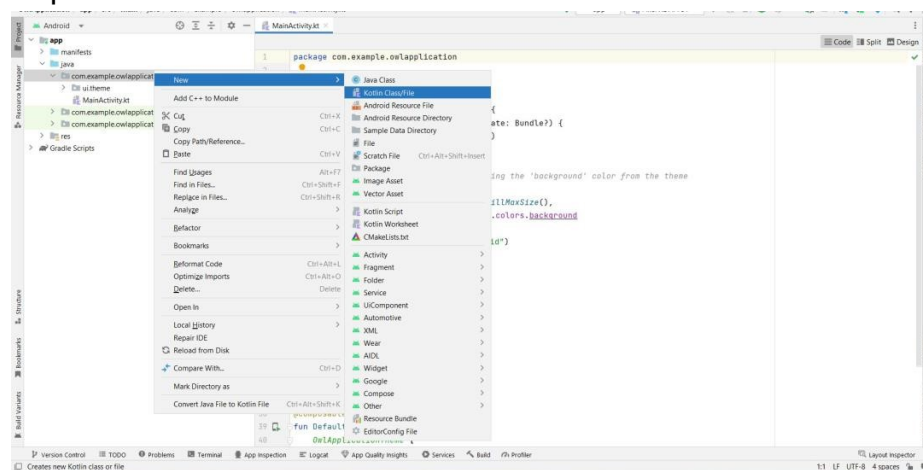


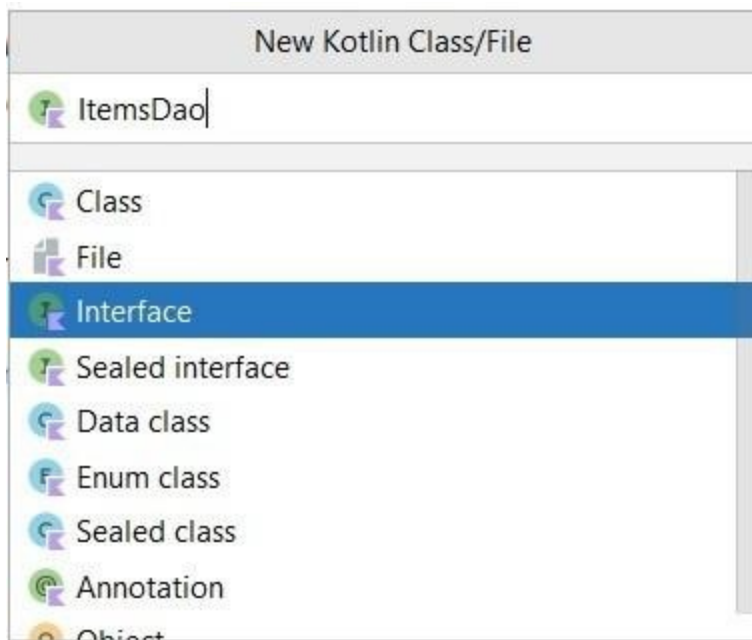
2. Creating the database classes for item name, quantity and cost.

Step 1 : Create Items data class

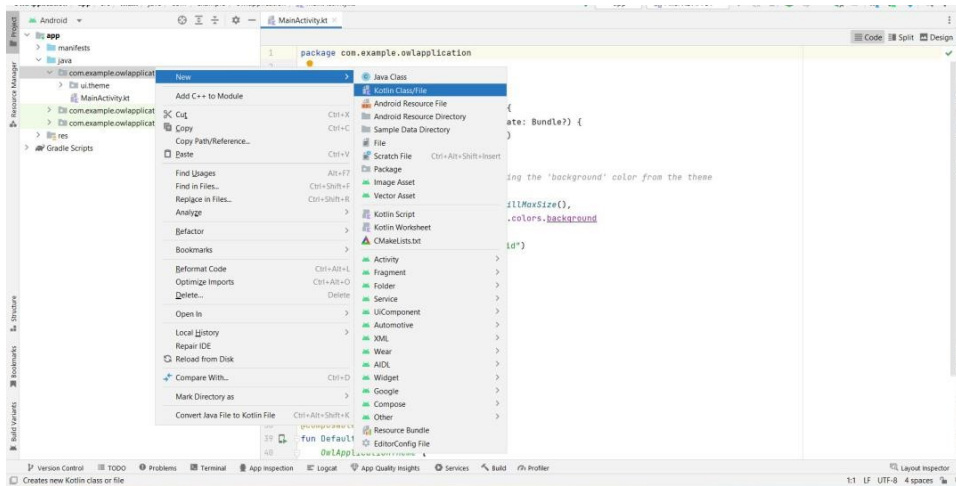


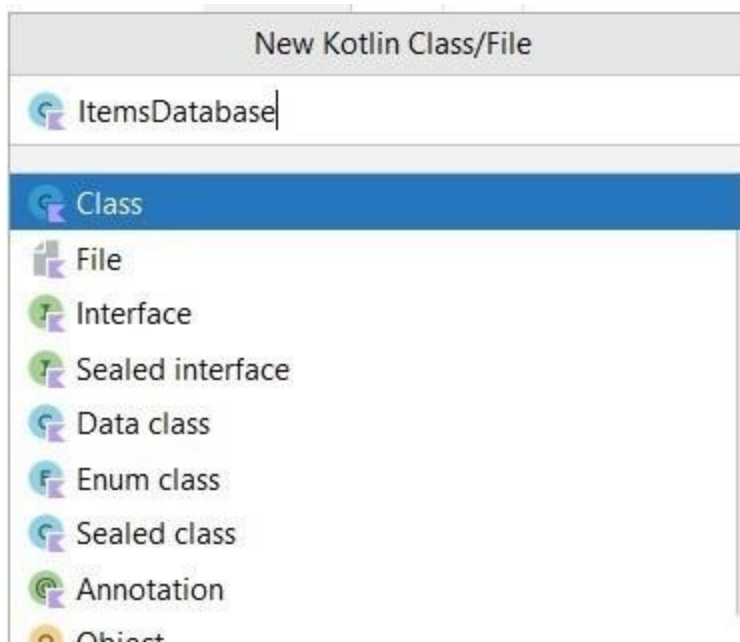
Step 2 : Create ItemsDao interface



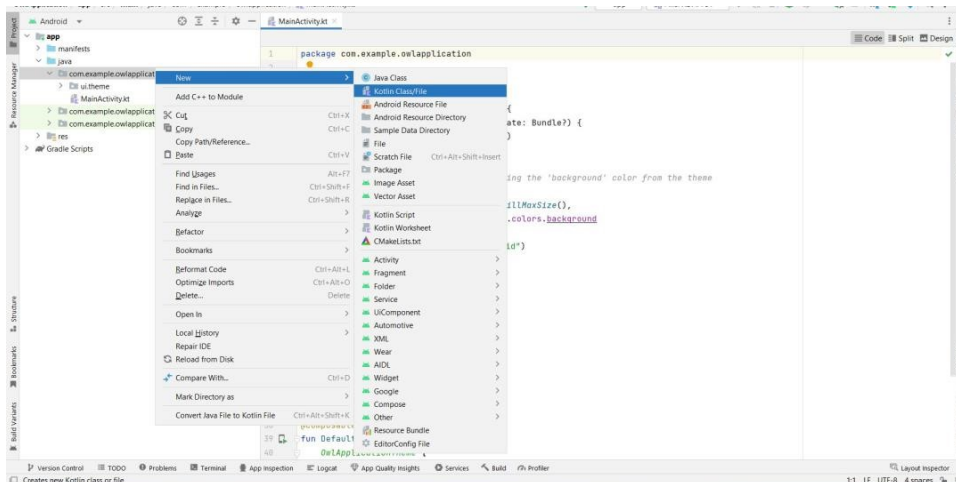


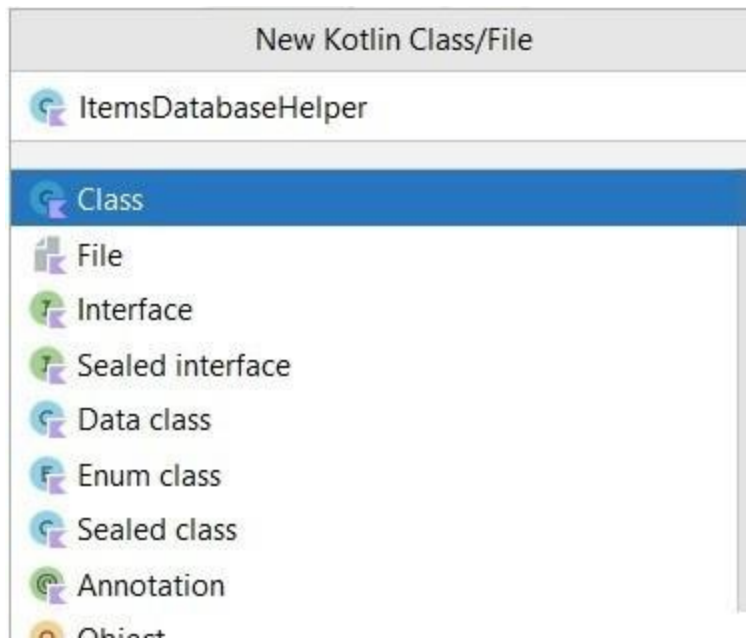
Step 3 : Create ItemsDatabase class





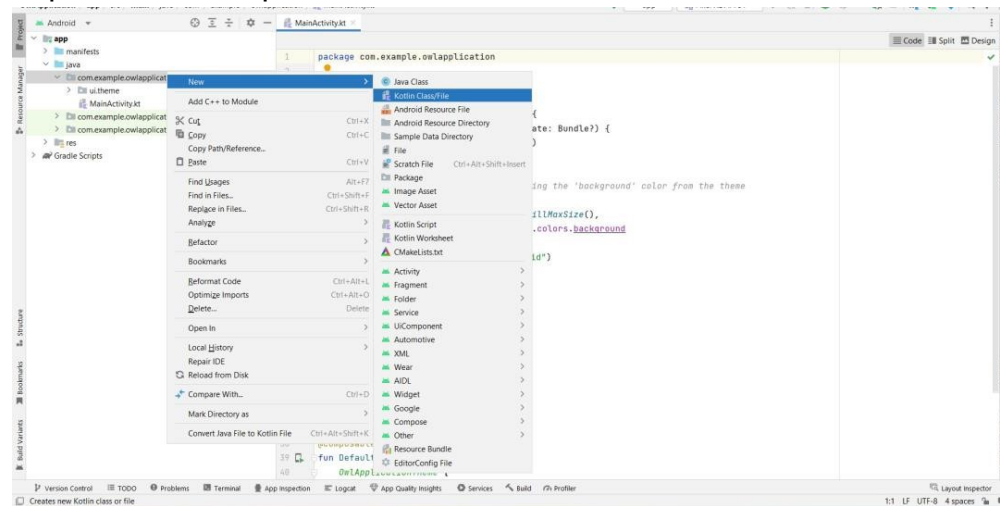
Step 4 : Create ItemsDatabaseHelper class

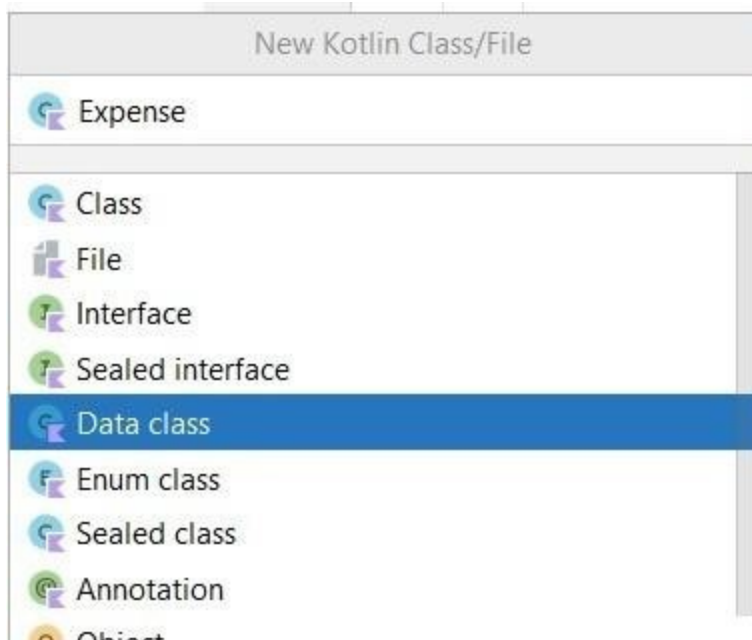




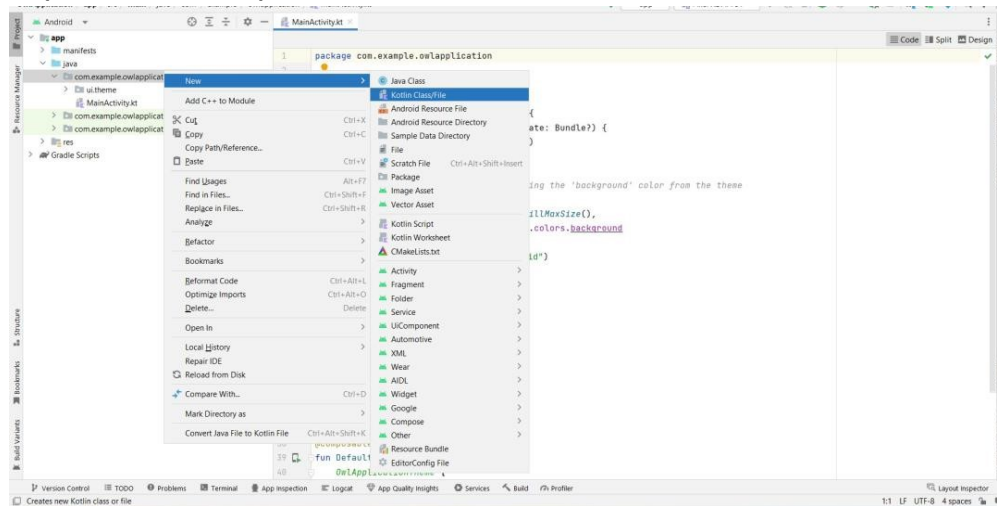
3. Creating the database classes for an amount.

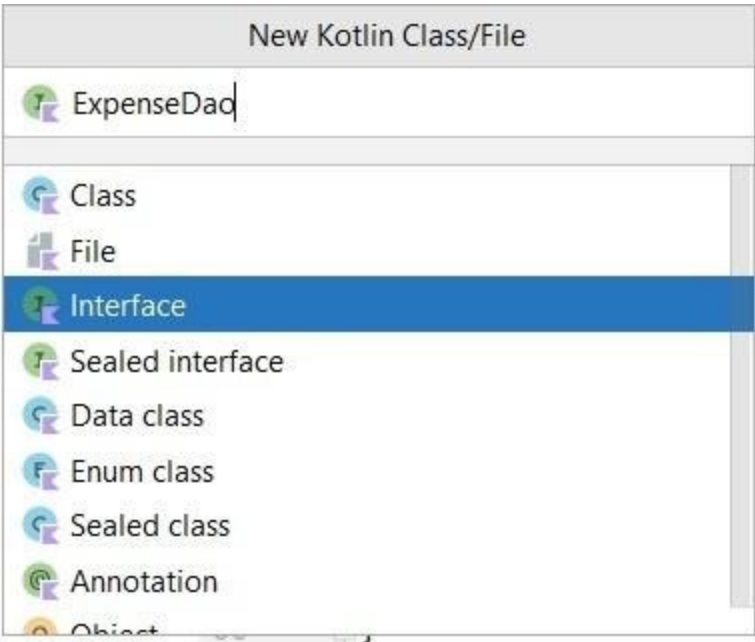
Step 1 : Create Expense data class



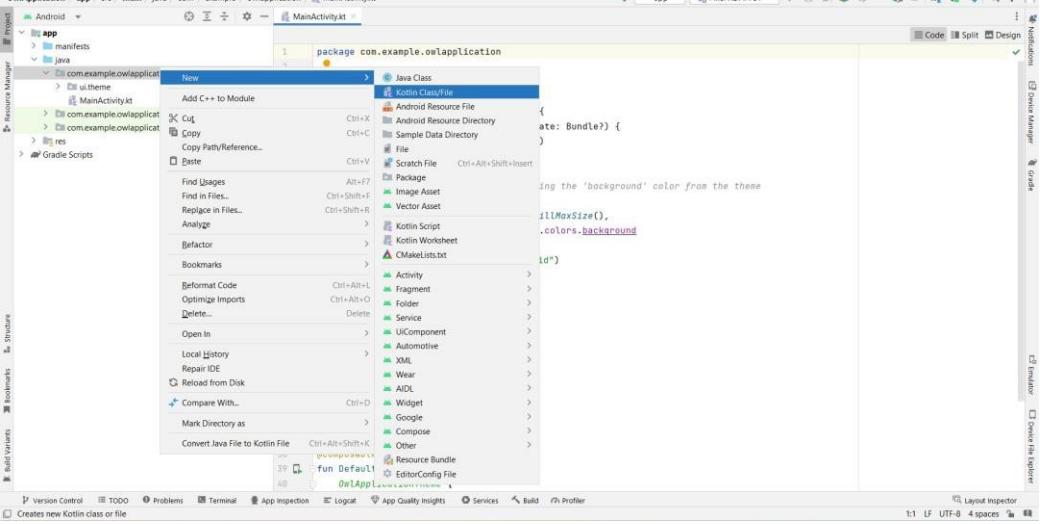


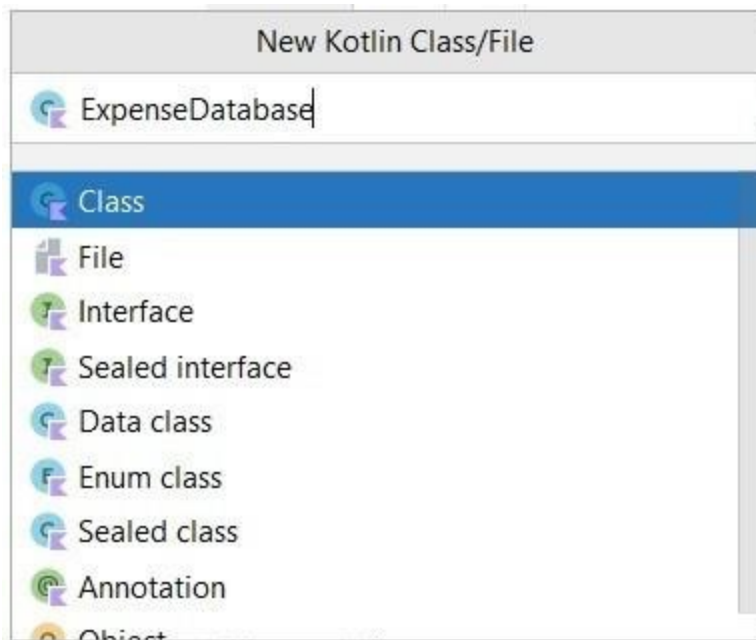
Step 2 : Create ExpenseDao interface



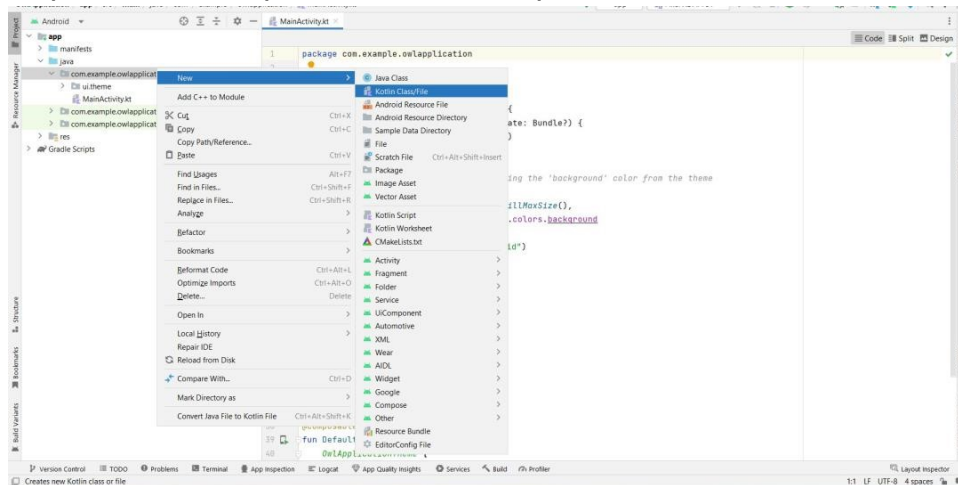


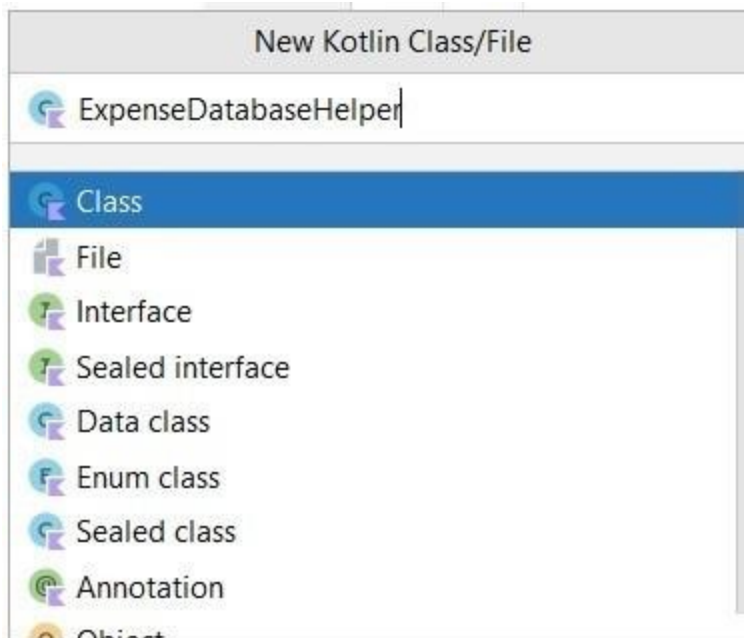
Step 3 : Create ExpenseDatabase class





Step 4 : Create ExpenseDatabaseHelper class

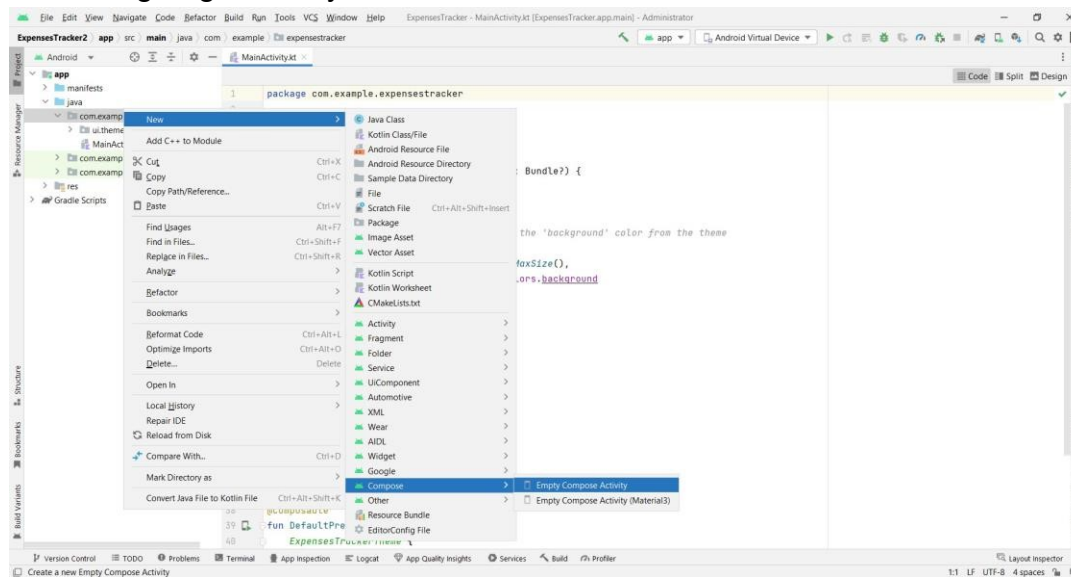


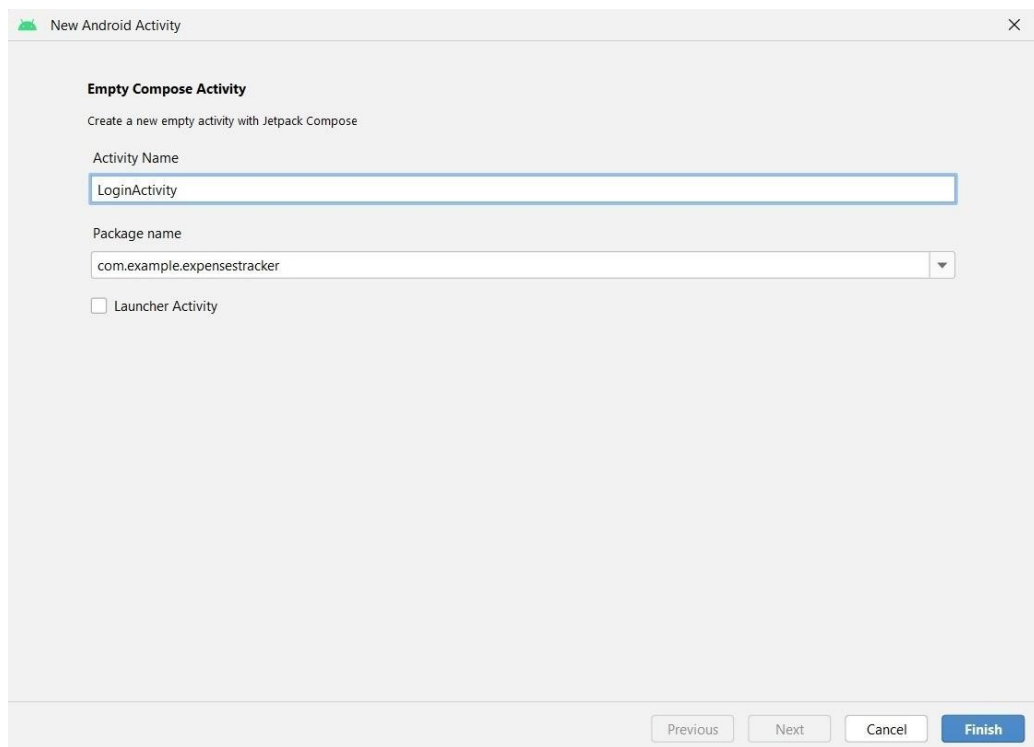


Task 5:

Building application UI and connecting to database. Step

1: Creating LoginActivity.kt with database





Database connection in LoginActivity.kt

```
package com.example.expensetracker

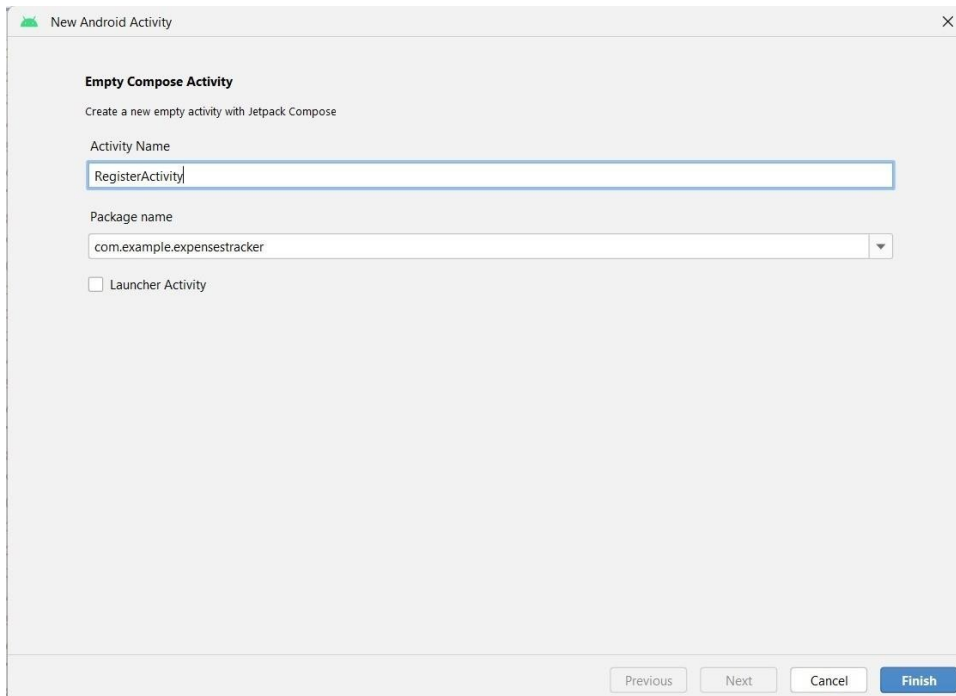
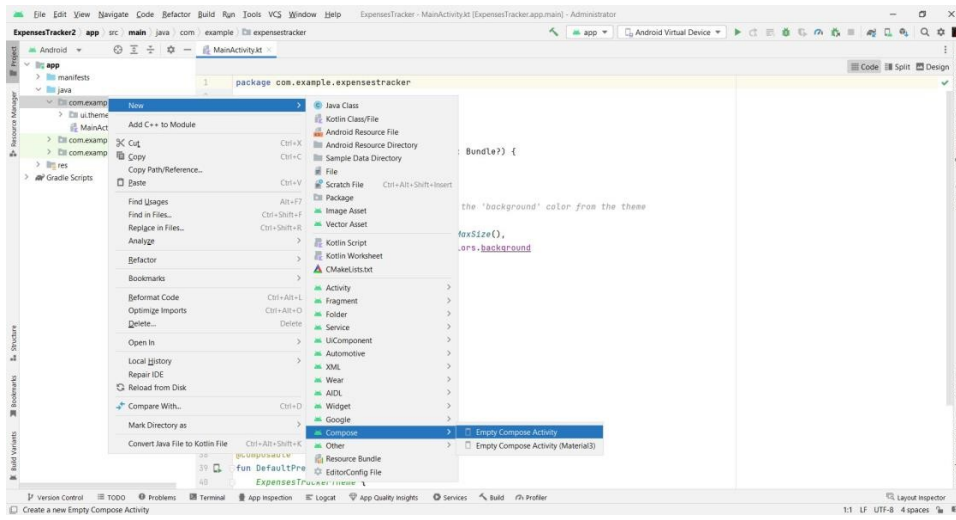
import ...

class LoginActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(context: this)
        setContent {
            ExpensesTrackerTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    LoginScreen(context: this, databaseHelper)
                }
            }
        }
    }
}

@Composable
fun LoginScreen(context: Context, databaseHelper: UserDatabaseHelper) {

    Image(
        painterResource(id = R.drawable.img_1), contentDescription = "",
        alpha = 0.3F,
        contentScale = ContentScale.FillHeight,
    )
}
```

Step 2 : Creating RegisterActivity.kt with database



Database connection in RegisterActivity.kt

```

package com.example.expensetracker

import ...

class RegisterActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper( context: this)
        setContent {
            ExpensesTrackerTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    RegistrationScreen( context: this,databaseHelper)
                }
            }
        }
    }
}

@Composable
fun RegistrationScreen(context: Context, databaseHelper: UserDatabaseHelper) {

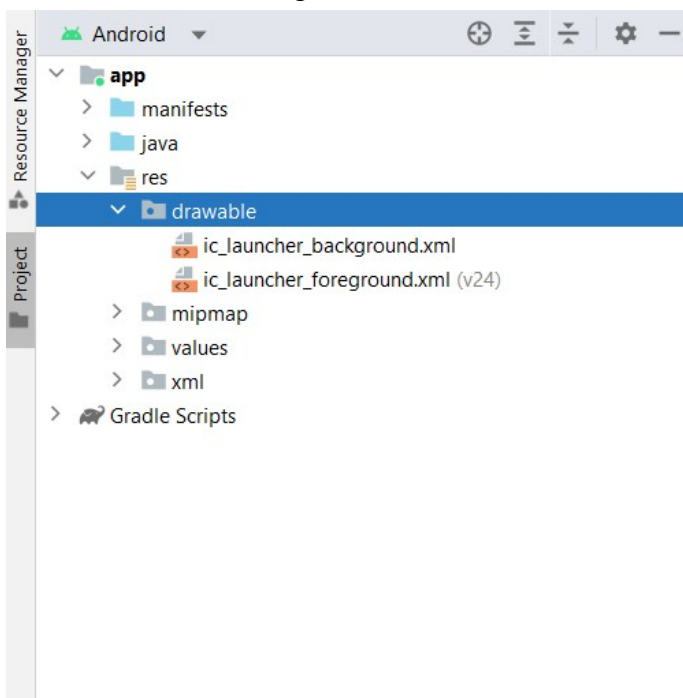
    Image(
        painterResource(id = R.drawable.img_1), contentDescription = "",
        alpha =0.3F,
    )
}

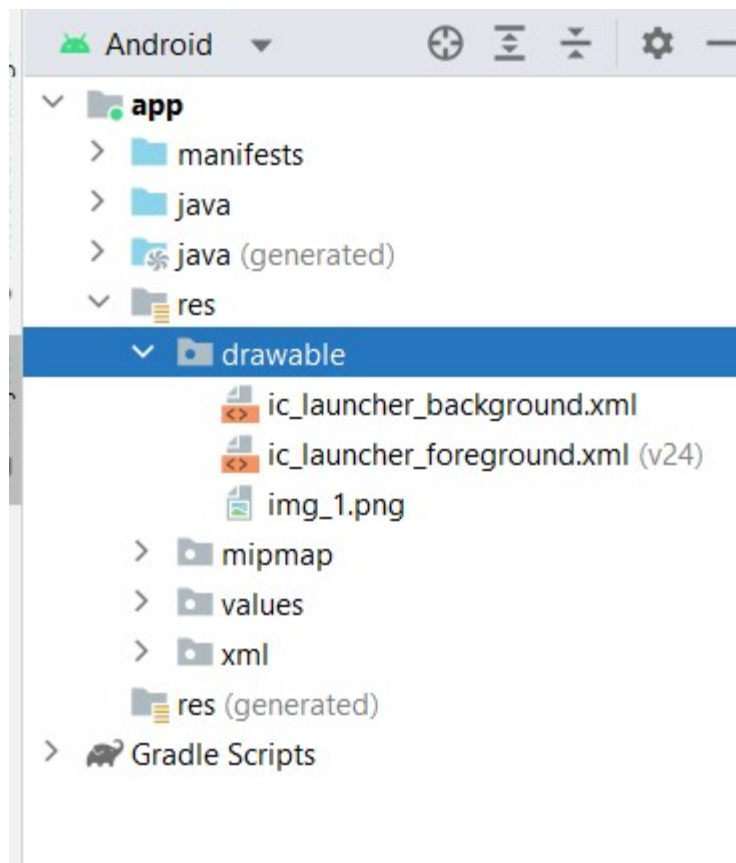
```

Step 3 : Creating MainActivity.kt file

In MainActivity.kt file the main application is developed

- Before creating UI we need to add some images in drawables which are in res





MainActivity.kt

```
package com.example.expensetracker

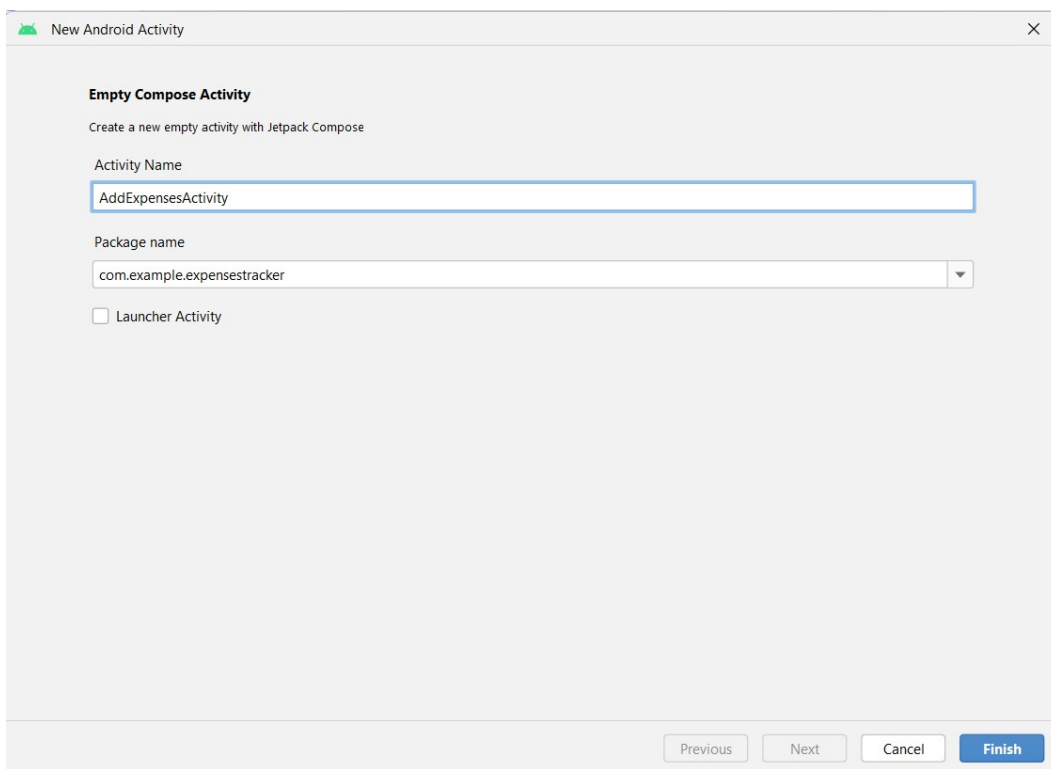
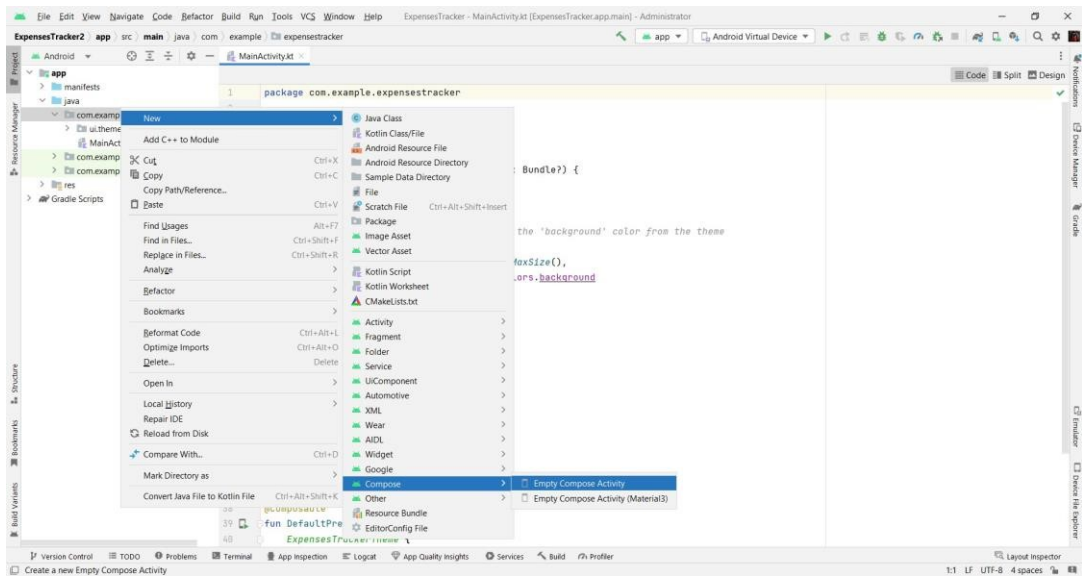
import ...

class MainActivity : ComponentActivity() {
    @SuppressLint("UnusedMaterialScaffoldPaddingParameter")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            Scaffold(
                // in scaffold we are specifying top bar.
                bottomBar = {
                    // inside top bar we are specifying
                    // background color.
                    BottomAppBar(backgroundColor = Color( color: 0xFFadbf4),
                        modifier = Modifier.height(80.dp),
                        // along with that we are specifying
                        // title for our top bar.
                        content = { this: RowScope
                            Spacer(modifier = Modifier.width(15.dp))

                            Button(
                                onClick = {startActivity(Intent(applicationContext, AddExpensesActivity::class.java))},
                                colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),
                                modifier = Modifier.size(height = 55.dp, width = 110.dp)
                            )
                        } this: RowScope
                            Text(
                                text = "Add Expenses", color = Color.Black, fontSize = 14.sp,

```

Step 4 : Creating AddExpensesActivity.kt file



```

package com.example.expensetracker

import ...

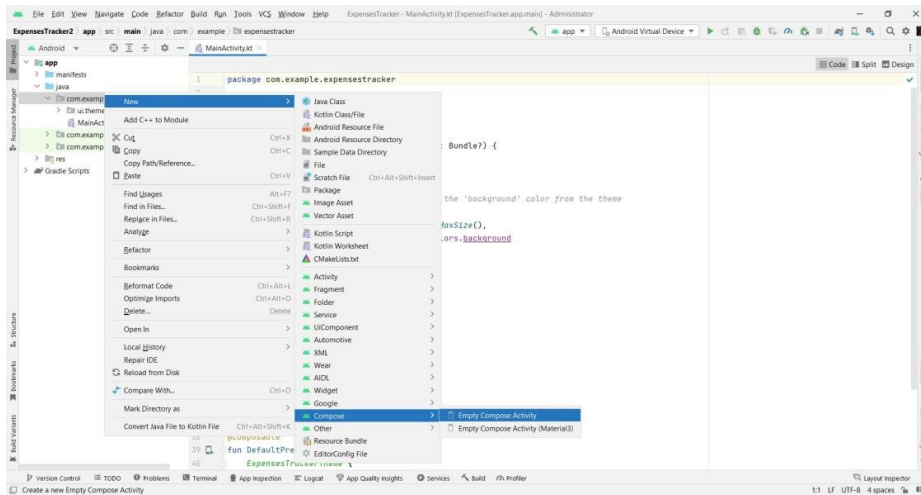
class AddExpensesActivity : ComponentActivity() {
    private lateinit var itemsDatabaseHelper: ItemsDatabaseHelper
    private lateinit var expenseDatabaseHelper: ExpenseDatabaseHelper
    @SuppressWarnings("UnusedMaterialScaffoldPaddingParameter")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        itemsDatabaseHelper = ItemsDatabaseHelper( context: this)
        expenseDatabaseHelper = ExpenseDatabaseHelper( context: this)
        setContentView {
            Scaffold(
                // in scaffold we are specifying top bar.
                bottomBar = {
                    // inside top bar we are specifying
                    // background color.
                    BottomAppBar(backgroundColor = Color( color: 0xFFdabcf4),
                        modifier = Modifier.height(80.dp),
                        // along with that we are specifying
                        // title for our top bar.
                        content = { this: RowScope

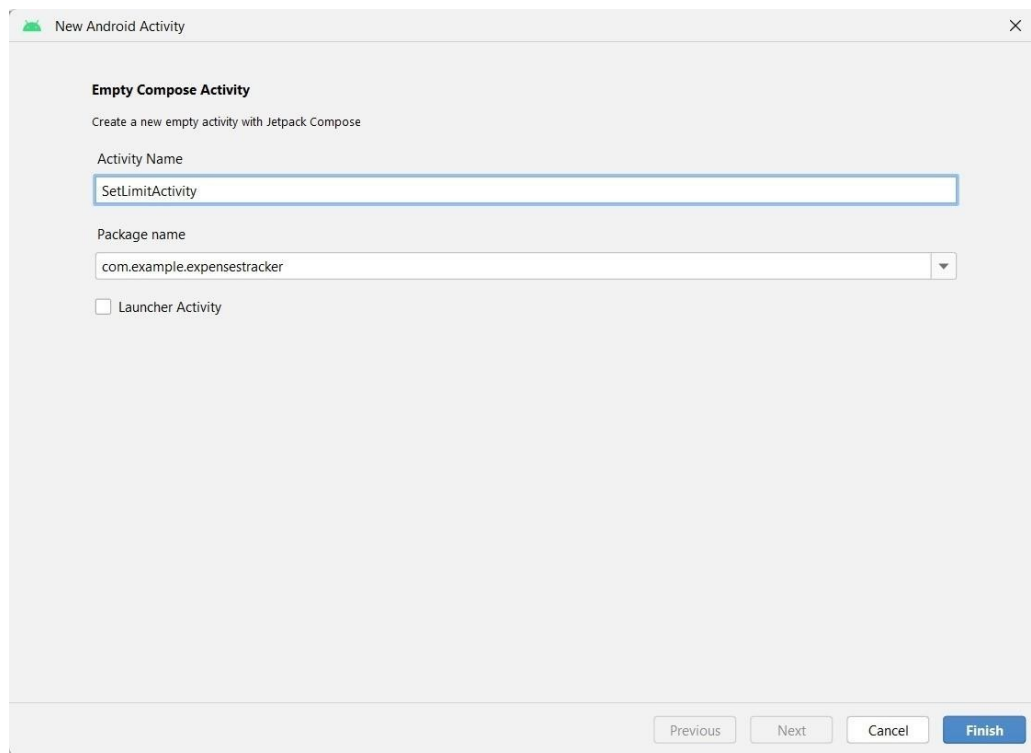
                            Spacer(modifier = Modifier.width(15.dp))

                            Button(
                                onClick = {startActivity(Intent(applicationContext,AddExpensesActivity::class.java))},
                                colors = ButtonDefaults.buttonColors(backgroundColor = Color.White),
                                modifier = Modifier.size(height = 55.dp, width = 110.dp)

```

Step 5 : Creating SetLimitActivity.kt file





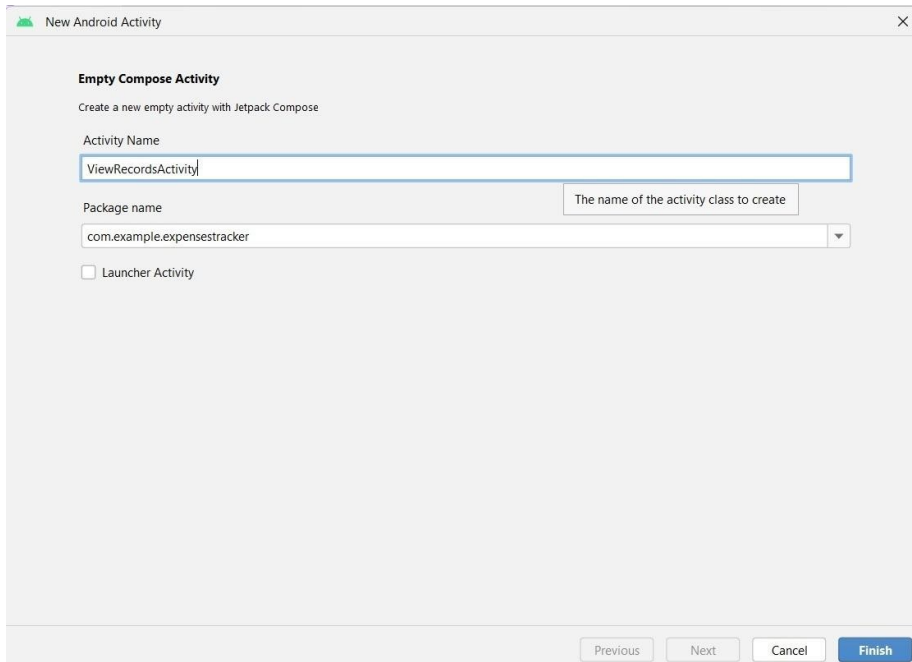
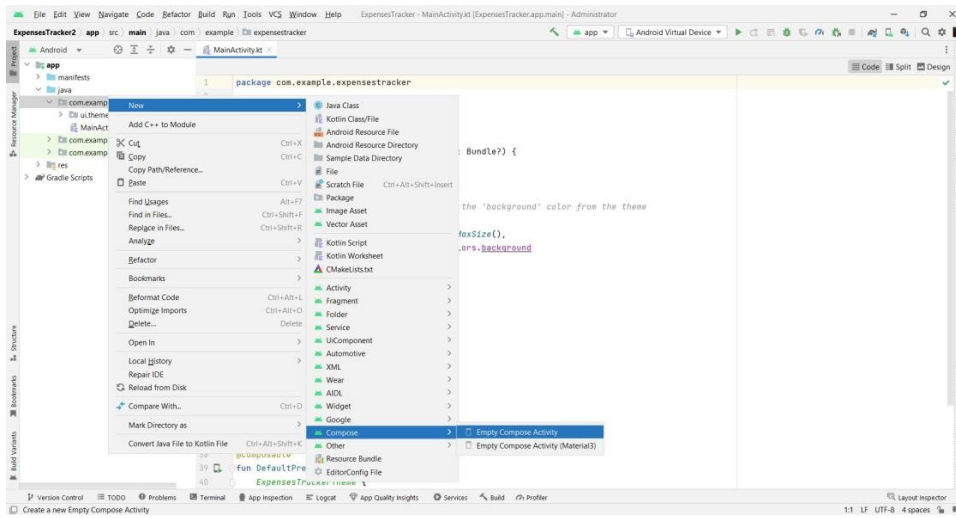
```
package com.example.expensetracker

import ...

class SetLimitActivity : ComponentActivity() {
    private lateinit var expenseDatabaseHelper: ExpenseDatabaseHelper
    @SuppressWarnings("UnusedMaterialScaffoldPaddingParameter")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        expenseDatabaseHelper = ExpenseDatabaseHelper(context = this)
        setContent {
            Scaffold(
                // in scaffold we are specifying top bar.
                bottomBar = {
                    // inside top bar we are specifying
                    // background color.
                    BottomAppBar(backgroundColor = Color(0xFFadbf4),
                        modifier = Modifier.height(80.dp),
                        // along with that we are specifying
                        // title for our top bar.
                        content = { this: RowScope
                            Spacer(modifier = Modifier.width(15.dp))

                            Button(
                                onClick = {
                                    startActivity(
                                        Intent(
                                            applicationContext,
                                            AddExpensesActivity::class.java
                                        )
                                    )
                                }
                            )
                        }
                    )
                }
            )
        }
    }
}
```

Step 6 : Creating ViewRecordsActivity.kt file



```
package com.example.expensetracker

import ...

class ViewRecordsActivity : ComponentActivity() {
    private lateinit var itemsDatabaseHelper: ItemsDatabaseHelper
    @SuppressWarnings("UnusedMaterialScaffoldPaddingParameter", "SuspiciousIndentation")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        itemsDatabaseHelper = ItemsDatabaseHelper( context: this)
        setContent {
            Scaffold(
                // in scaffold we are specifying top bar.
                bottomBar = {
                    // inside top bar we are specifying
                    // background color.
                    BottomAppBar(backgroundColor = Color( color: 0xFFadbf4),
                        modifier = Modifier.height(80.dp),
                        // along with that we are specifying
                        // title for our top bar.
                        content = { this.RowScope

                            Spacer(modifier = Modifier.width(15.dp))

                            Button(
                                onClick = {
                                    startActivity(
                                        Intent(
                                            applicationContext,
                                            AddExpensesActivity::class.java
                                        )
                                    )
                                }
                            )
                        }
                    )
                }
            )
        }
    }
}
```

Task 6:


```

        android:label="LoginActivity"
        android:theme="@style/Theme.ExpensesTracker" />
    <activity
        android:name=".ViewRecordsActivity"
        android:exported="false"
        android:label="ViewRecordsActivity"
        android:theme="@style/Theme.ExpensesTracker" />
    <activity
        android:name=".SetLimitActivity"
        android:exported="false"
        android:label="SetLimitActivity"
        android:theme="@style/Theme.ExpensesTracker" />
    <activity
        android:name=".AddExpensesActivity"
        android:exported="false"
        android:label="AddExpensesActivity"
        android:theme="@style/Theme.ExpensesTracker" />
    <activity
        android:name=".MainActivity"
        android:exported="true"
        android:label="Expenses Tracker"
        android:theme="@style/Theme.ExpensesTracker">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

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

```

When we run the app we will get the MainActivity.kt file as our first screen , but we want LoginActivity.kt , So we need to change in AndroidManifest.xml.

Changed AndroidManifest.xml

```

        android:label="MainActivity"
        android:theme="@style/Theme.ExpensesTracker" />
    <activity
        android:name=".ViewRecordsActivity"
        android:exported="false"
        android:label="ViewRecordsActivity"
        android:theme="@style/Theme.ExpensesTracker" />
    <activity
        android:name=".SetLimitActivity"
        android:exported="false"
        android:label="@string/title_activity_set_limit"
        android:theme="@style/Theme.ExpensesTracker" />
    <activity
        android:name=".AddExpensesActivity"
        android:exported="false"
        android:label="AddExpensesActivity"
        android:theme="@style/Theme.ExpensesTracker" />
    <activity
        android:name=".LoginActivity"
        android:exported="true"
        android:label="Expenses Tracker"
        android:theme="@style/Theme.ExpensesTracker">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

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

```

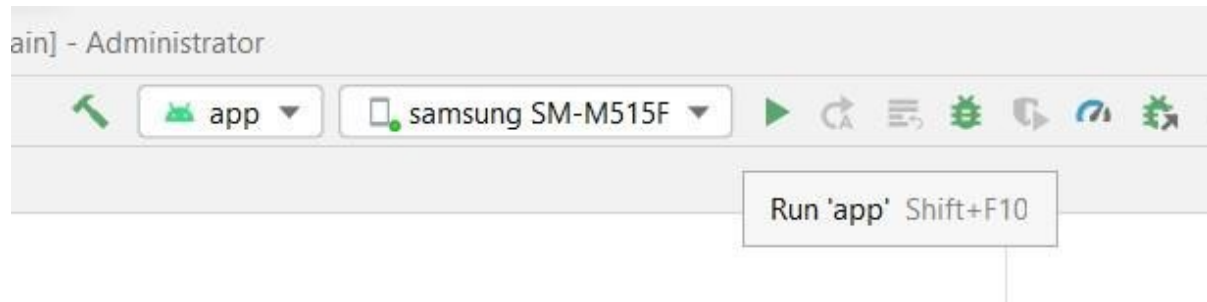
Task 7:

Running the application.

Step 1: Run apps on a hardware device

<https://developer.android.com/studio/run/device>

Step 2: Run the application in Mobile

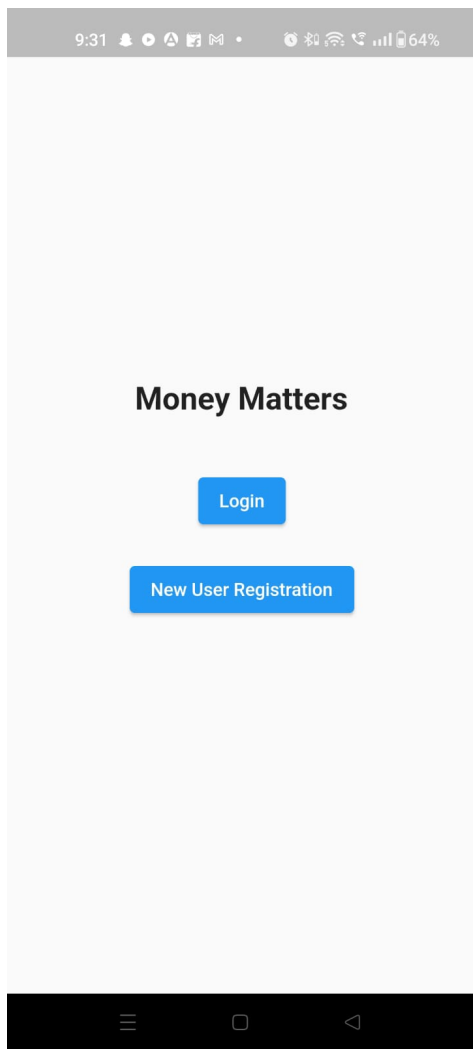


Complete Project Link:

<https://github.com/smartinternz02/SI-GuidedProject-587453-1697048492>

Final Output of the Application :

Front view :-



New Registration View :-

9:31 [notification icons] [status icons] 64%

← New User Registration

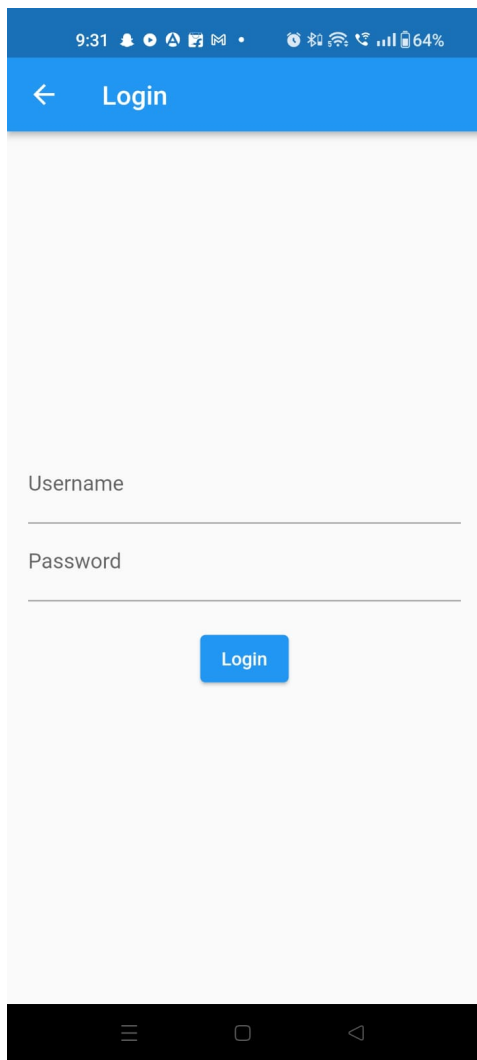
New Username

New Password

Register

[navigation bar]

Login page :-



A screenshot of a mobile application's login screen. At the top, a blue header bar contains a back arrow and the word "Login". Below this, the main area is light gray and contains two input fields: "Username" and "Password", each with a horizontal line for text entry. A blue "Login" button is positioned below the password field. The top status bar shows the time as 9:31 and various system icons. The bottom navigation bar is black with three white icons: a hamburger menu, a home square, and a back triangle.

9:31

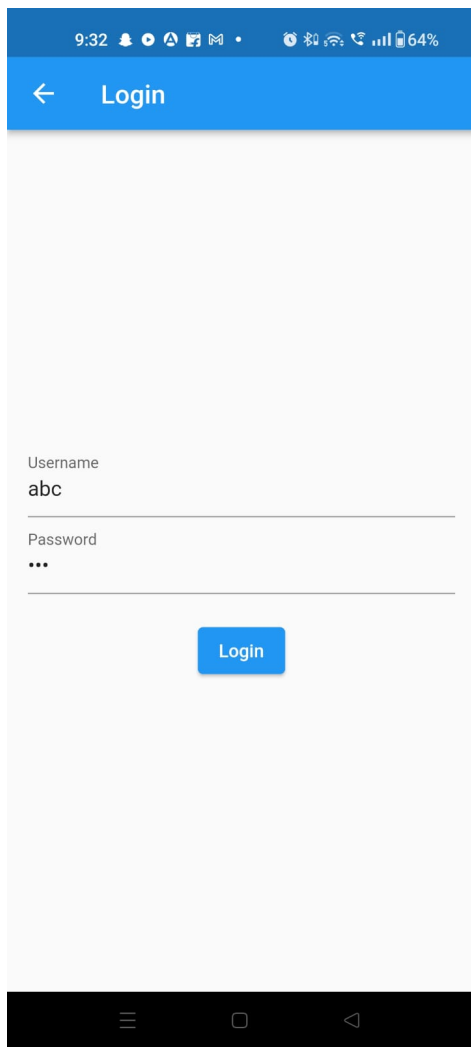
← Login

Username

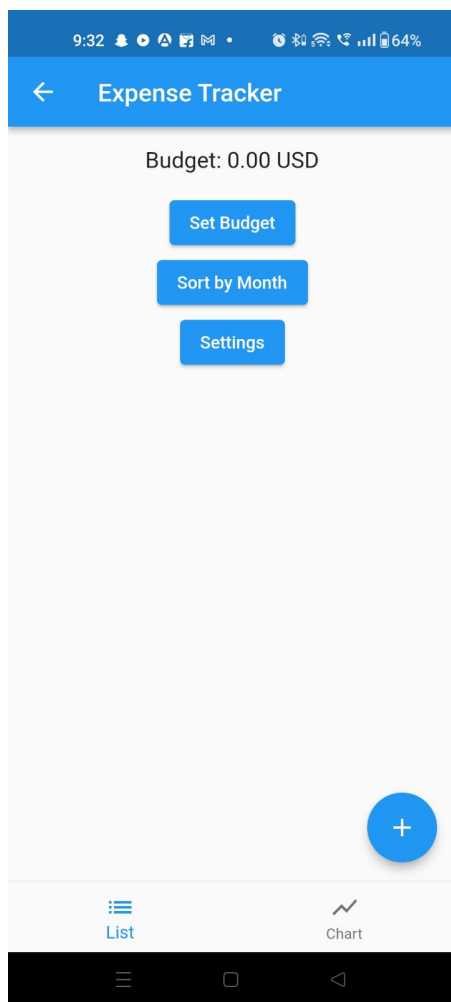
Password

Login

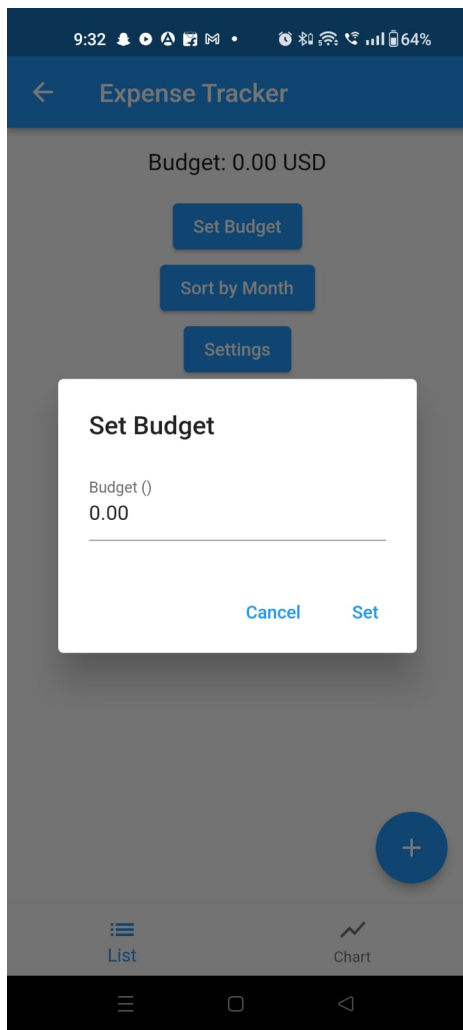
Login credentials written in new user registration :-



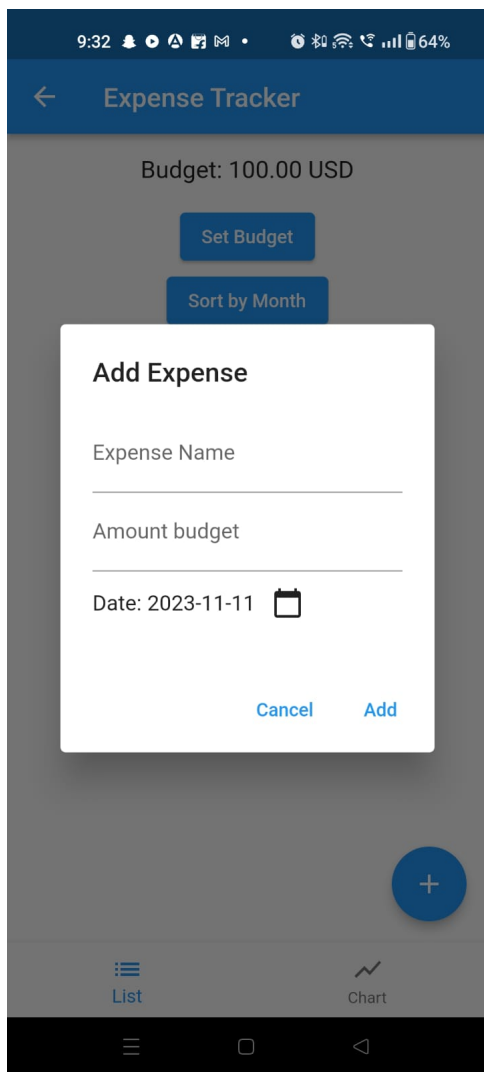
Expense Tracker View :-



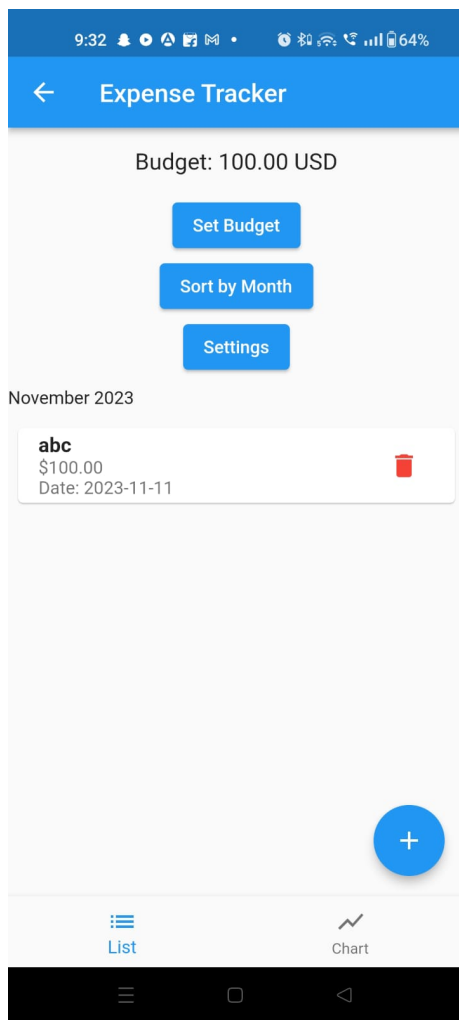
Set Budget View :-



Add Expense View :-



Expense Tracker View :-



Expense Tracker (Sort by Month) :-

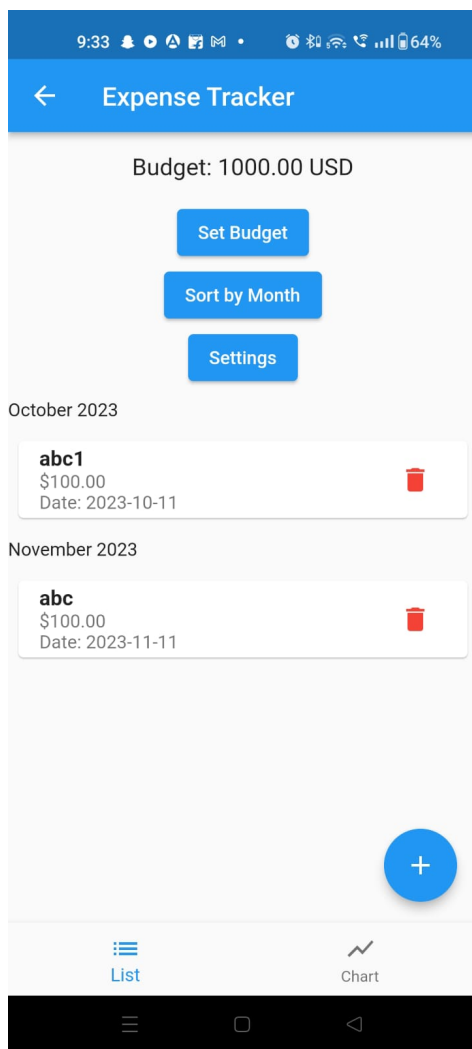
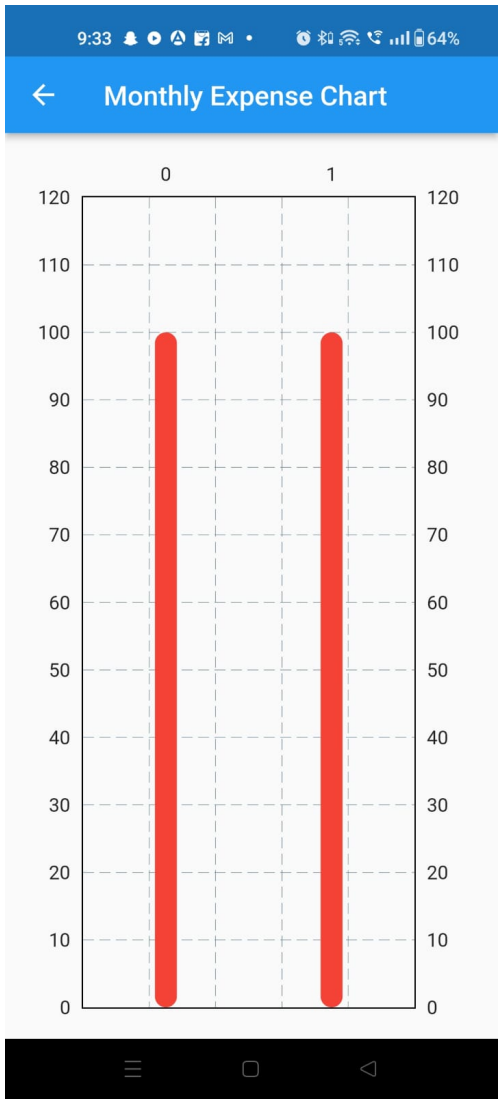


Chart View :-



Options :-

