Money Matters: A Personal Finance Management App

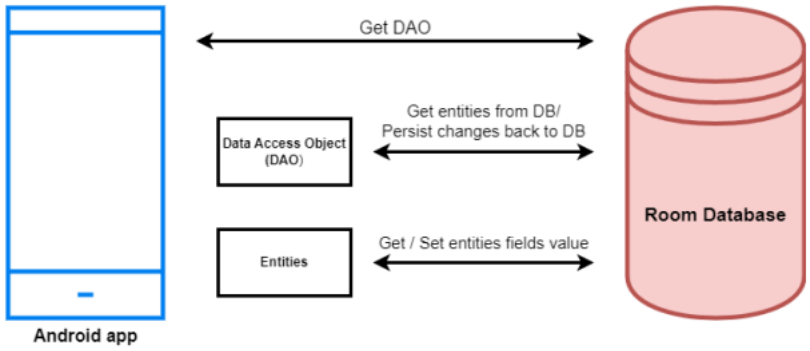
Project Based Experimental Learning Program

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 logins 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.

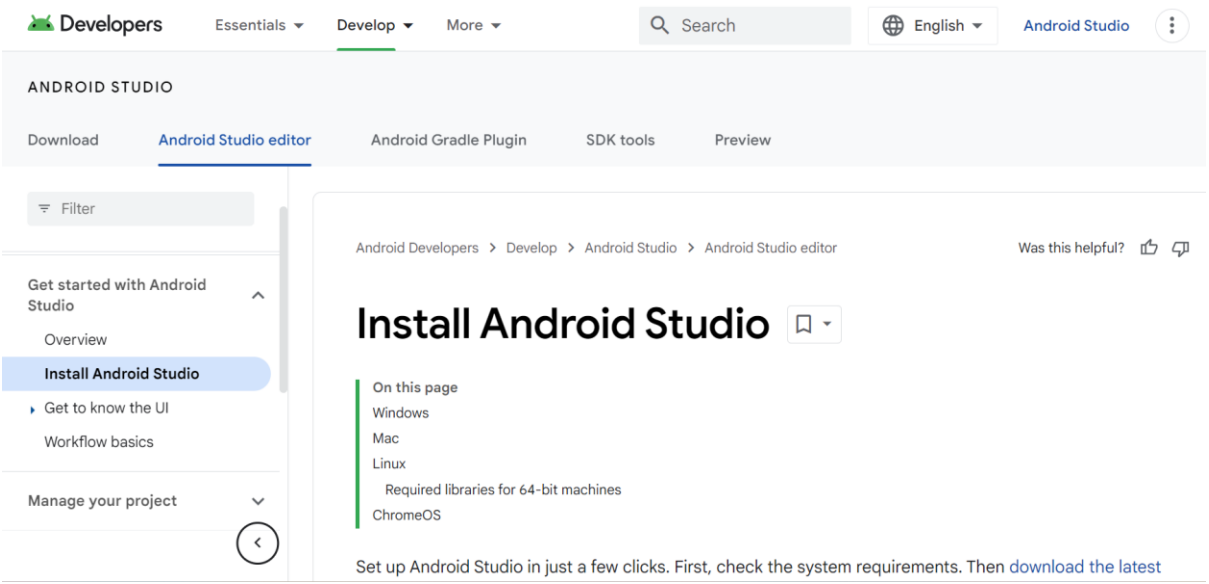
6.Using AndroidManifest.xml

7.Running the application.

Task 1:

Required initial steps :

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

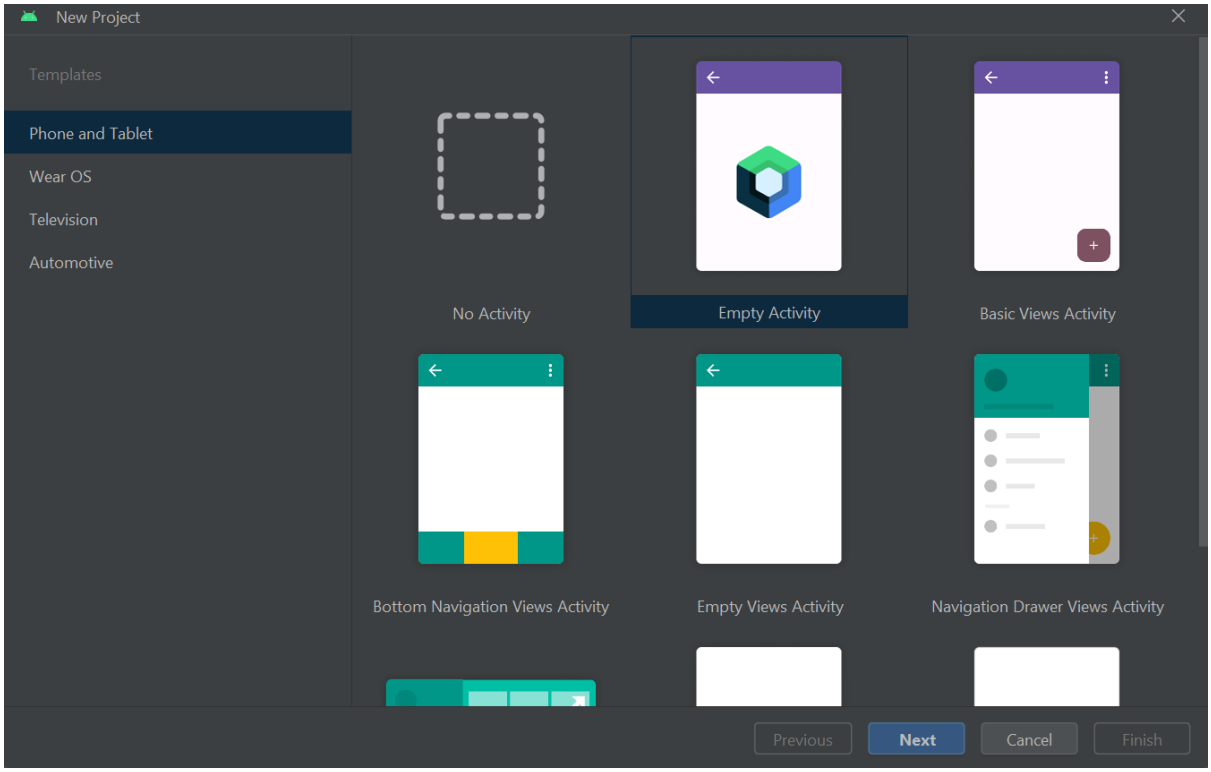


Task2 :

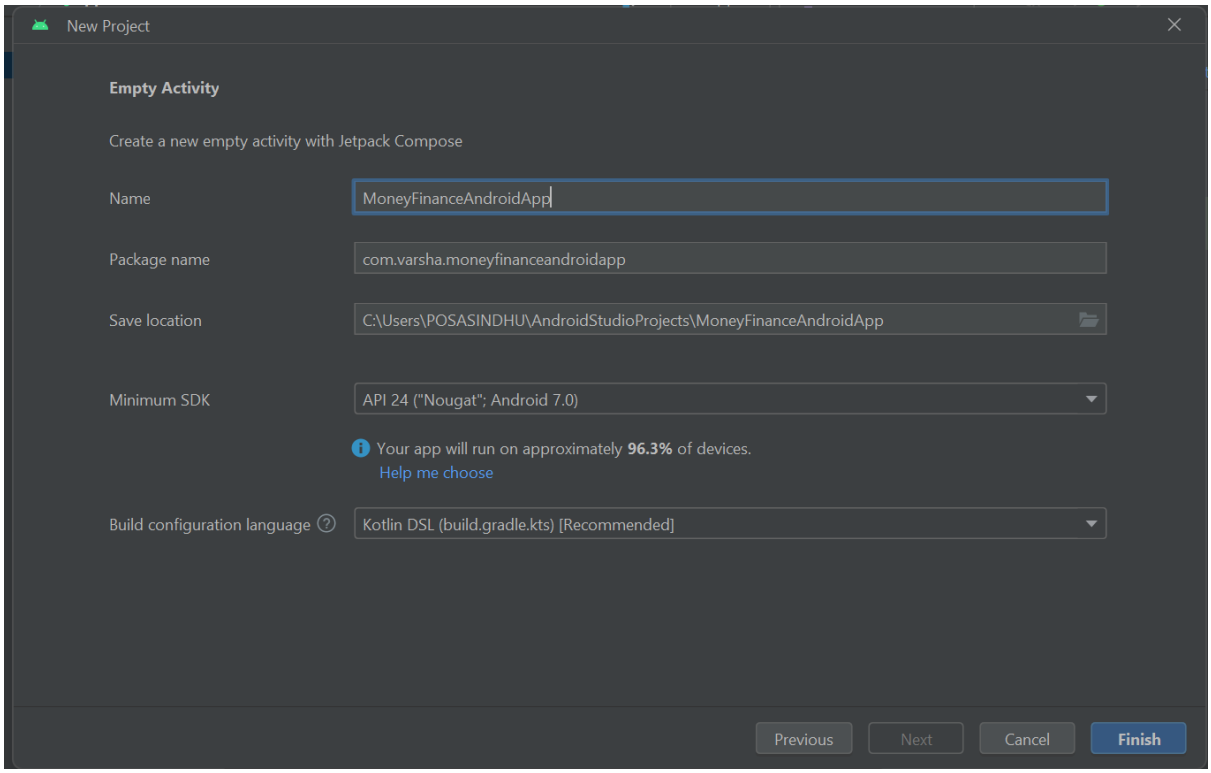
Creating a new project.

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

Step 2 : Click on Next button.

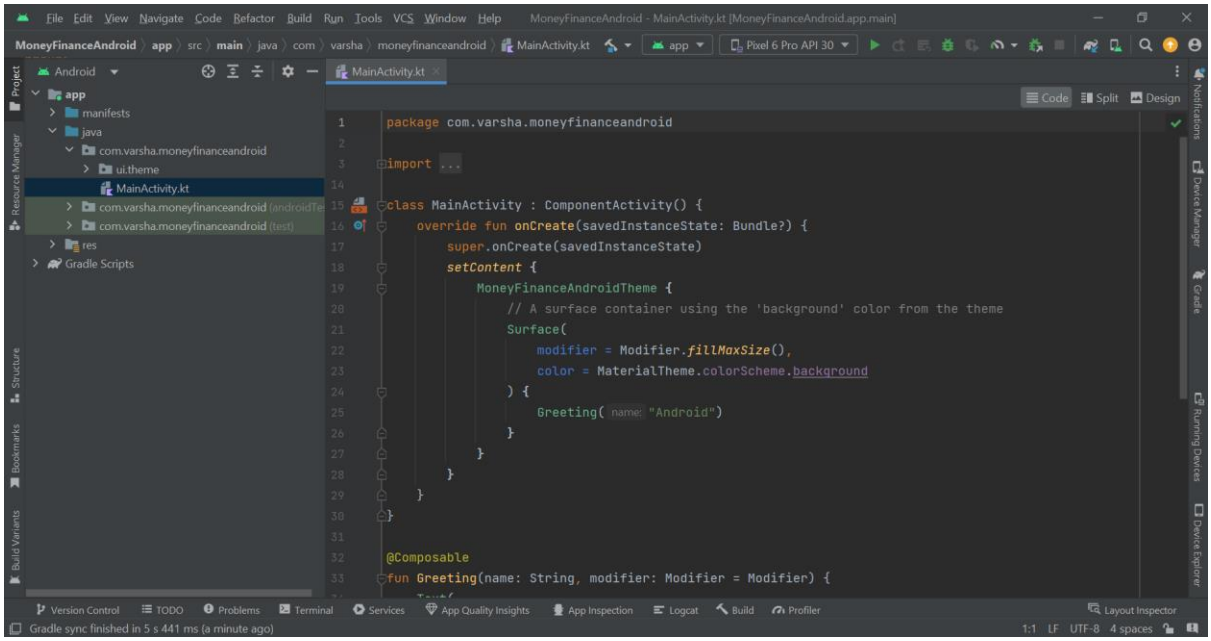


Step 3 : Give name to the new project.



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

```
dependencies {
    implementation("androidx.core:core-ktx:1.12.0")
    implementation("androidx.lifecycle:lifecycle-runtime-ktx:2.6.2")
    implementation("androidx.activity:activity-compose:1.8.0")
    implementation(platform("androidx.compose:compose-bom:2023.03.00"))
    implementation("androidx.compose.ui:ui")
    implementation("androidx.compose.ui:ui-graphics")
    implementation("androidx.compose.ui:ui-tooling-preview")
    implementation("androidx.compose.material3:material3")
    implementation ("androidx.room:room-common:2.6.0")
    implementation ("androidx.room:room-ktx:2.6.0")
    implementation(platform("androidx.compose:compose-bom:2023.03.00"))
    implementation(platform("androidx.compose:compose-bom:2023.03.00"))
    implementation(platform("androidx.compose:compose-bom:2023.03.00"))
    implementation(platform("androidx.compose:compose-bom:2023.03.00"))
    implementation(platform("androidx.compose:compose-bom:2023.03.00"))
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
    androidTestImplementation(platform("androidx.compose:compose-
bom:2023.03.00"))
    androidTestImplementation("androidx.compose.ui:ui-test-junit4")
    androidTestImplementation(platform("androidx.compose:compose-
bom:2023.03.00"))
    androidTestImplementation(platform("androidx.compose:compose-
bom:2023.03.00"))
    androidTestImplementation(platform("androidx.compose:compose-
bom:2023.03.00"))
```

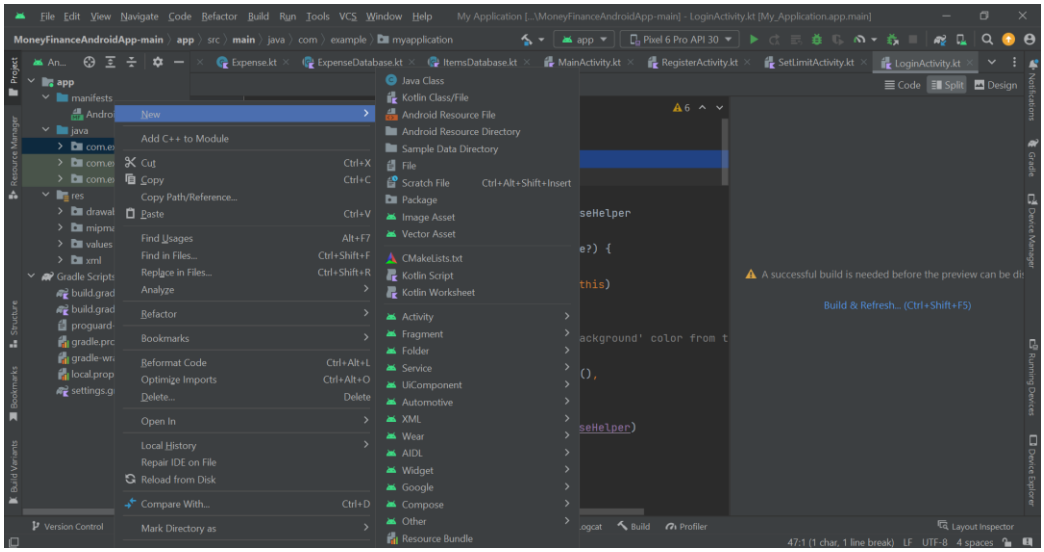
```
    androidTestImplementation(platform("androidx.compose:compose-
bom:2023.03.00"))
    androidTestImplementation(platform("androidx.compose:compose-
bom:2023.03.00"))
    debugImplementation("androidx.compose.ui:ui-tooling")
    debugImplementation("androidx.compose.ui:ui-test-manifest")

}
```

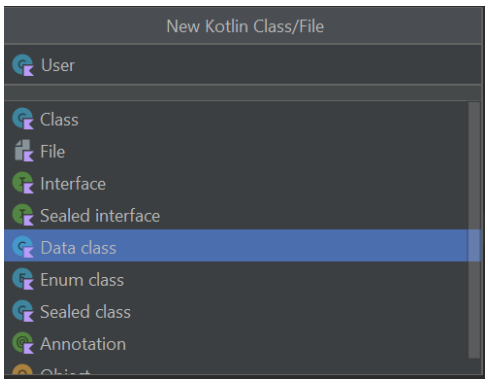
Step 3 : Click on Sync now

Task 4:

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

Step 1 : Create User data class

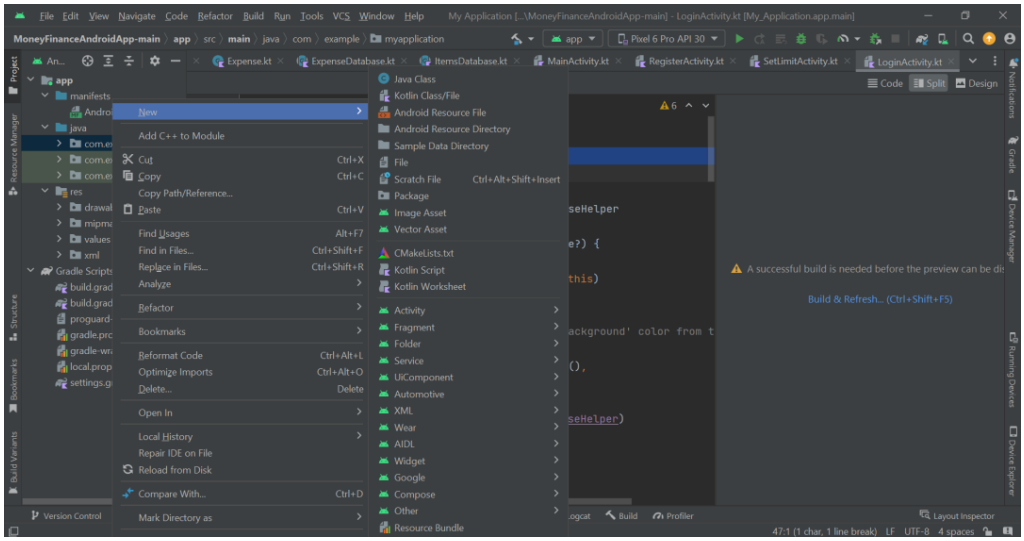


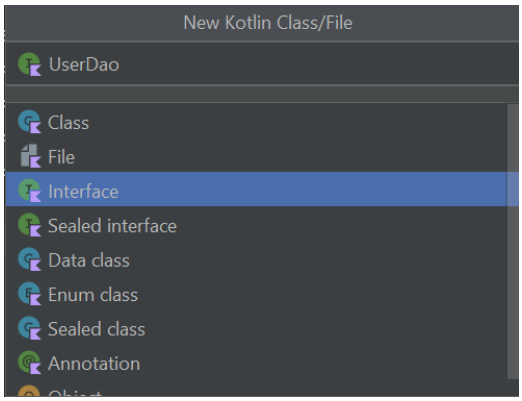


User data class code:

https://github.com/Suraj041203/MoneyFinanceAndroidApp/blob/main/app/src/main/java/com/example/myapplication/User.kt

Step 2 : Create an UserDao interface

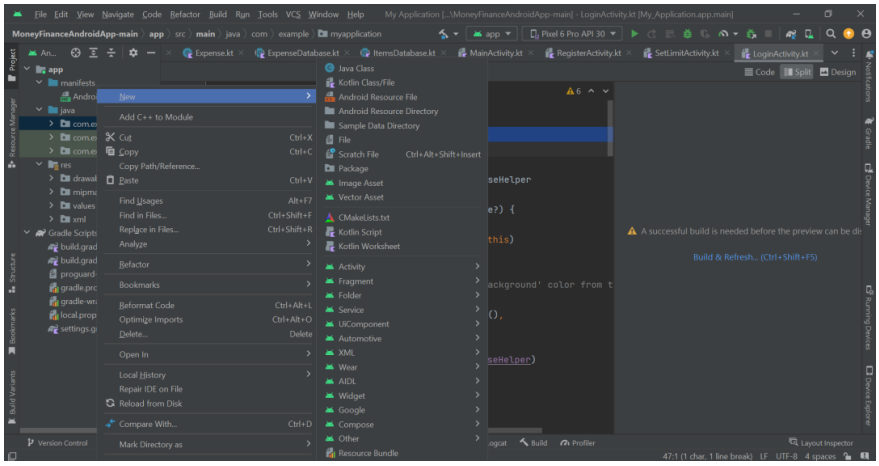


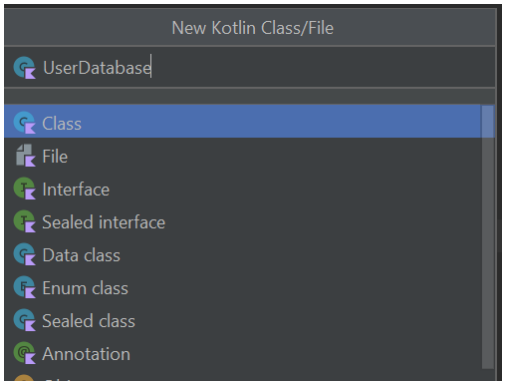


UserDao interface code :

https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

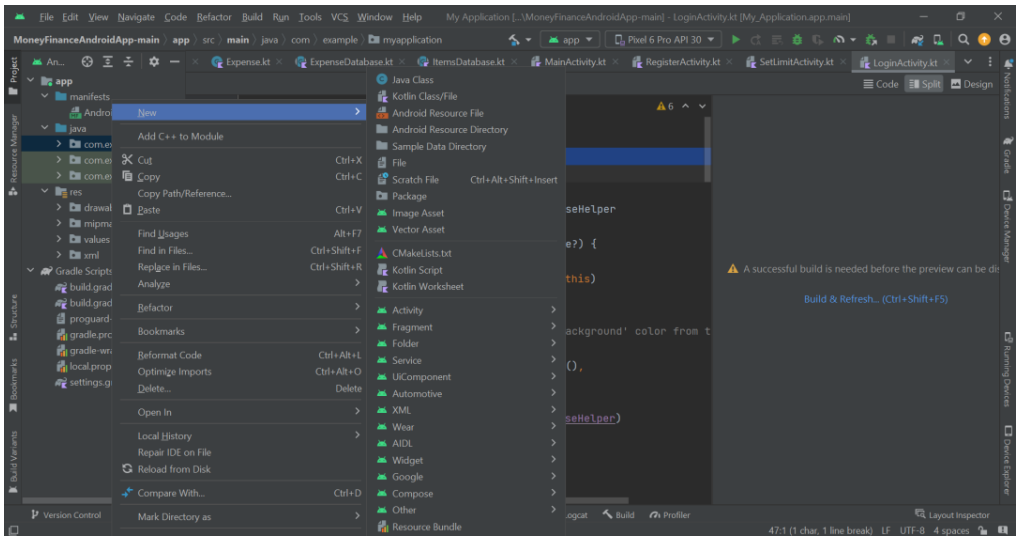
Step 3 : Create an UserDatabase class

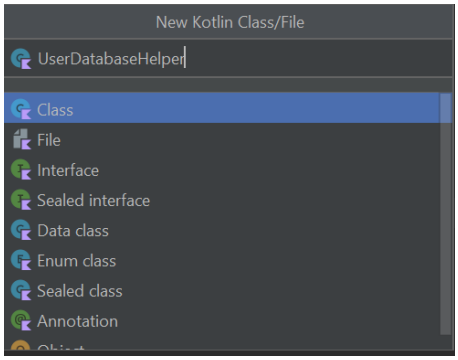




UserDatabase class code :
https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

Step 4 : Create an UserDatabaseHelper class



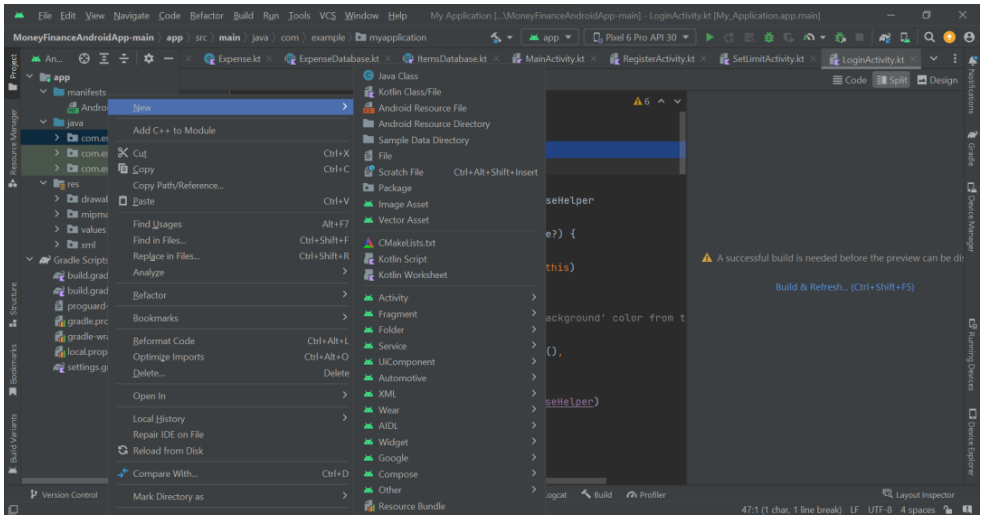


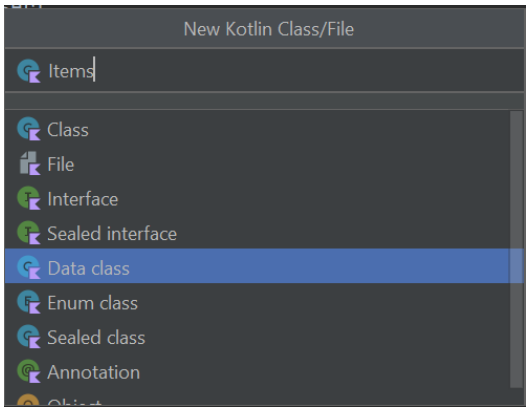
UserDatabaseHelper class code :

https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

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

Step 1 : Create Items data class

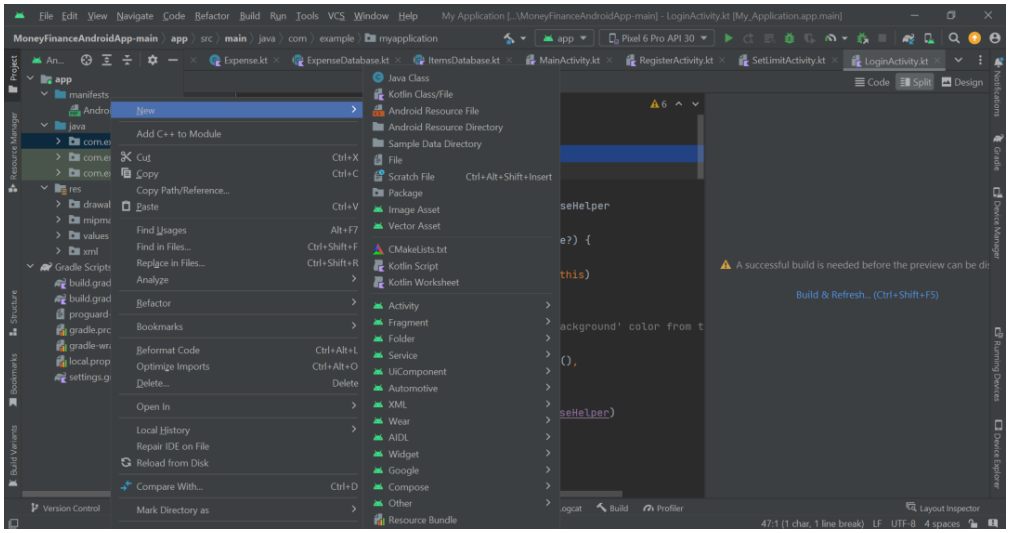


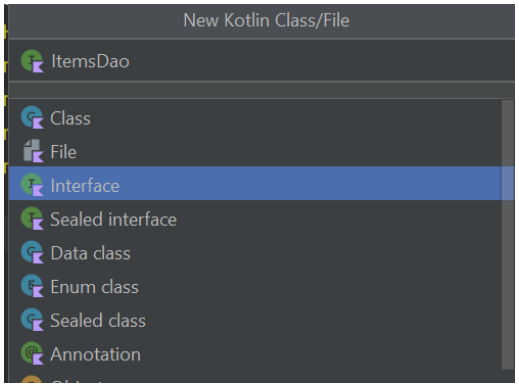


Items Data class code :

https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

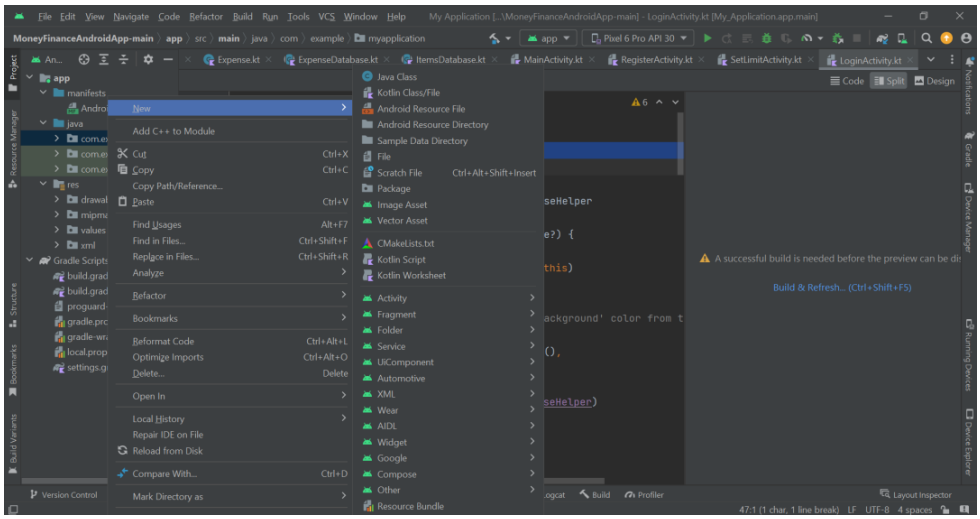
Step 2 : Create ItemsDao interface

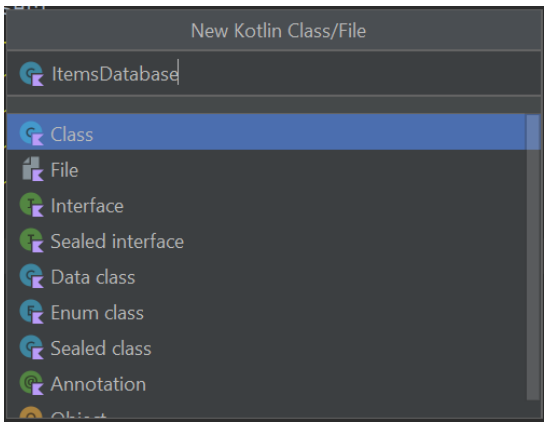




ItemsDao interface code
https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

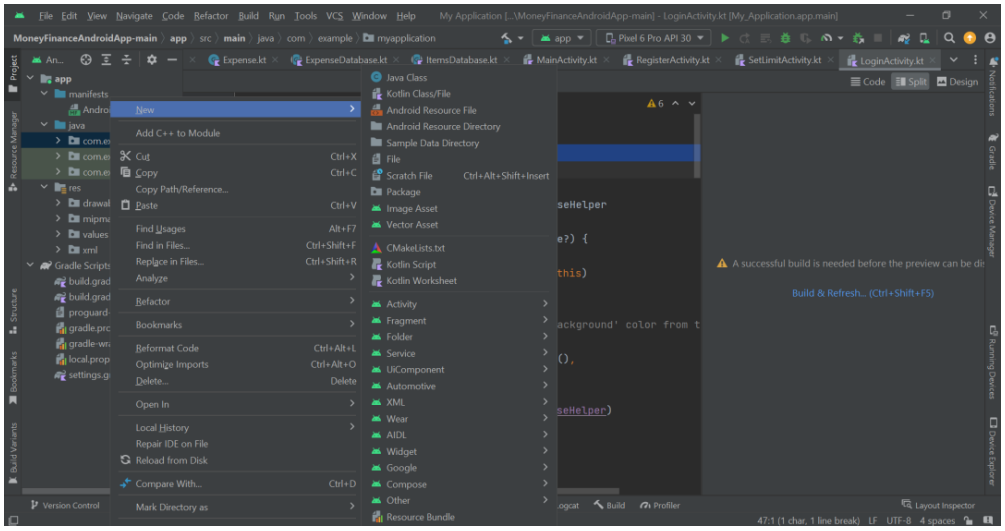
Step 3 : Create ItemsDatabse class

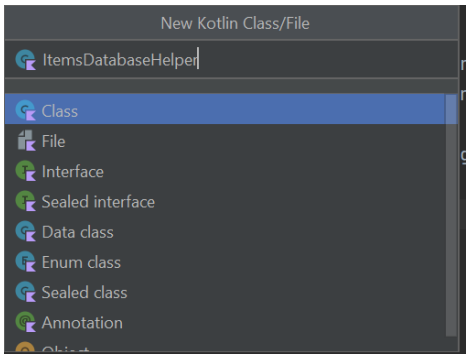




ItemsDatabase class code :
https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

Step 4 : Create ItemsDatabaseHelper class



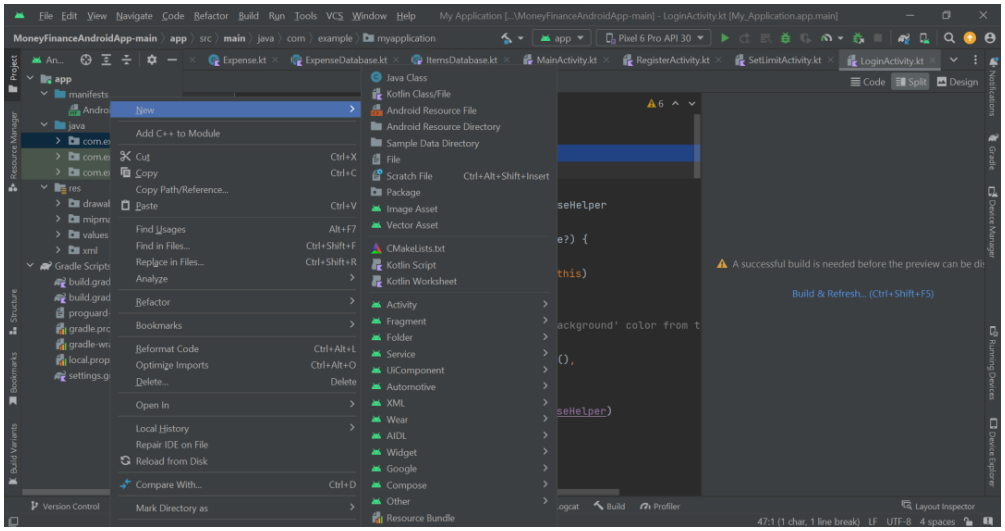


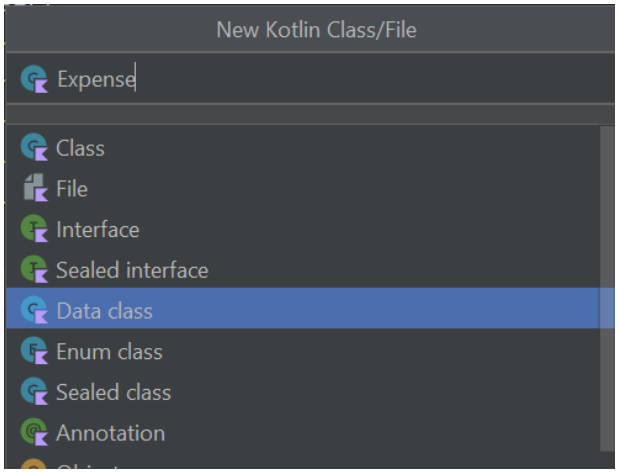
ItemsDatabaseHelper class code:

https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

3.Creating the database classes for an amount.

Step 1 : Create Expense data class

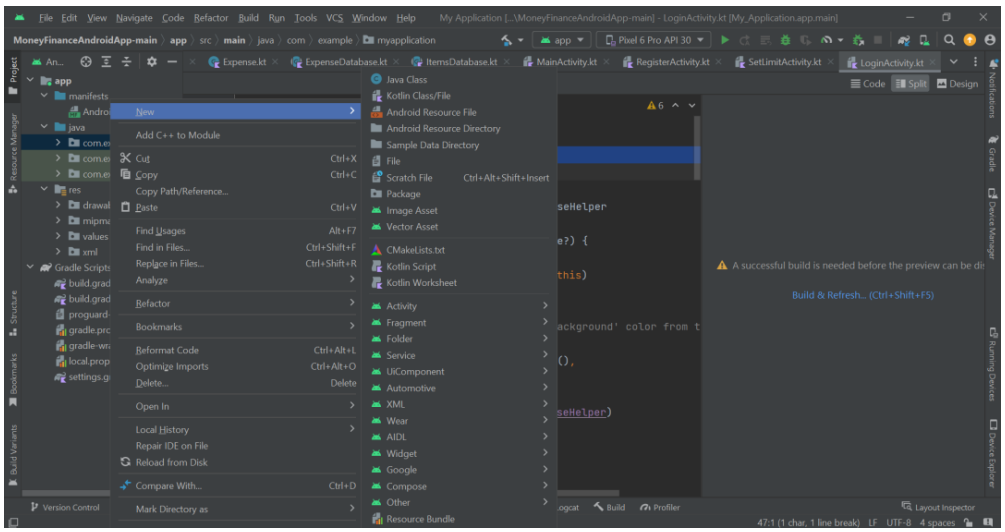


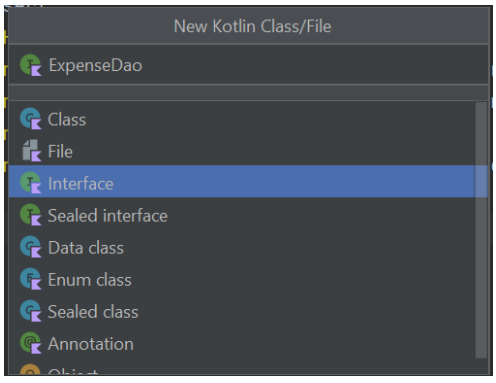


Expense data class Code:

https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

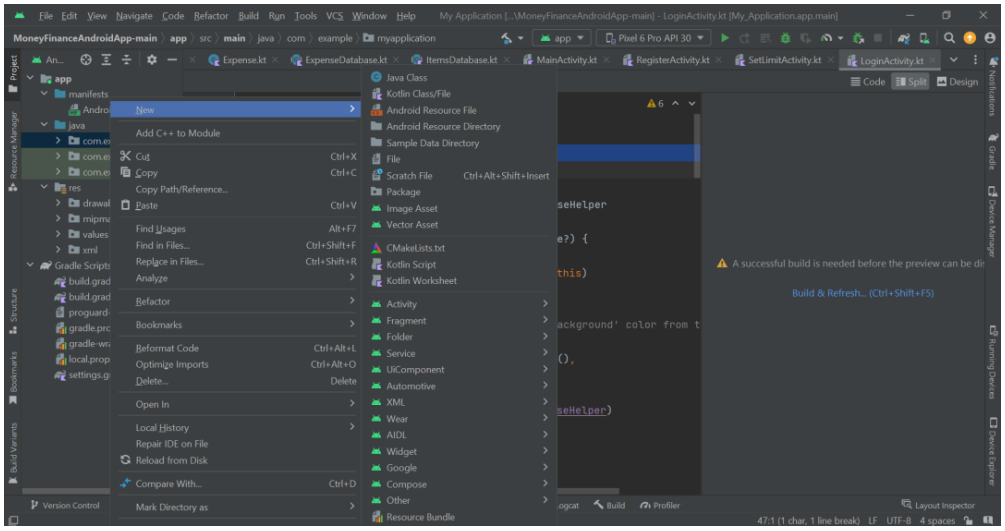
Step 2 : Create ExpenseDao interface

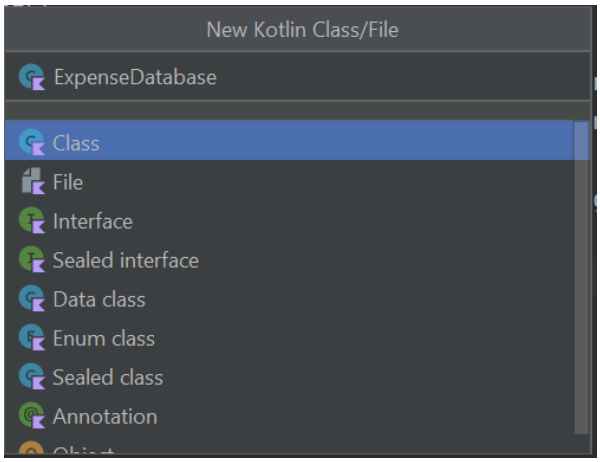




ExpenseDao interface Code :
https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

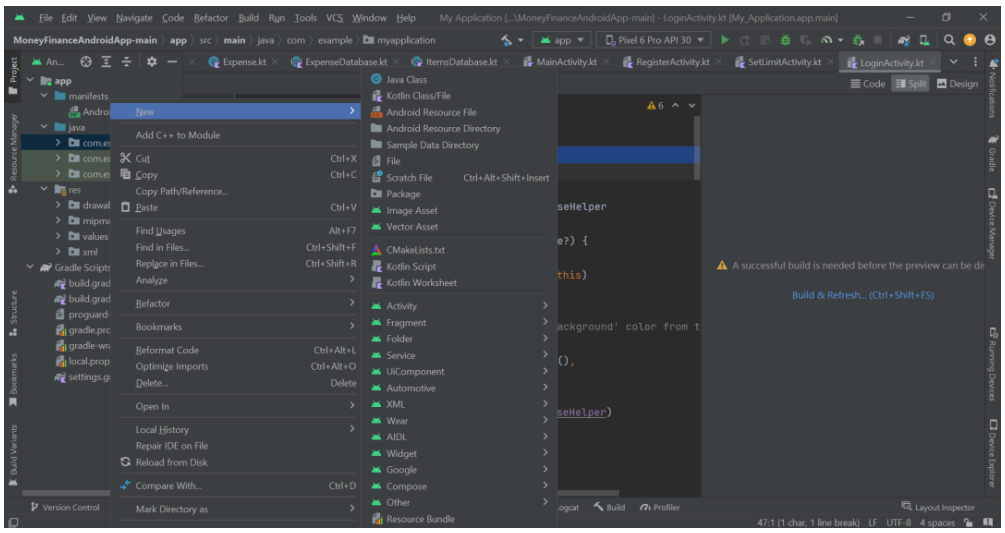
Step 3 : Create ExpenseDatabase class

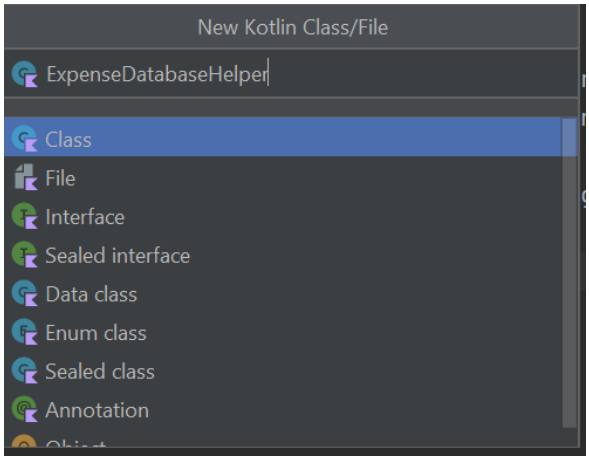




ExpenseDatabase class Code
https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

Step 4 : Create ExpenseDatabaseHelper class



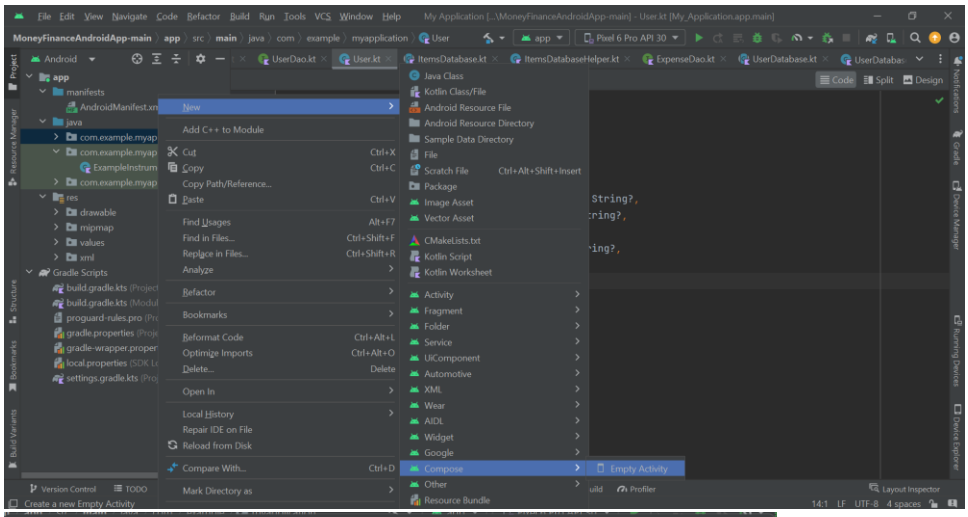


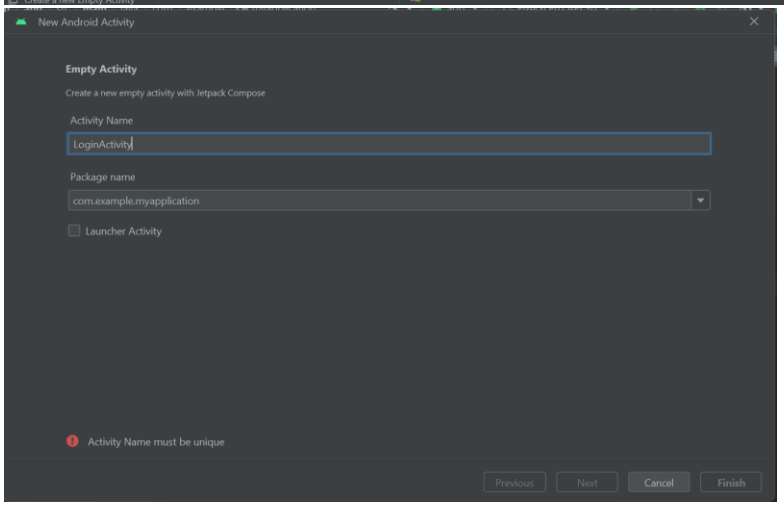
ExpenseDatabaseHelper class Code : https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

Task 5:

Building application UI and connecting to database.

Step 1: Creating LoginActivity.kt with database



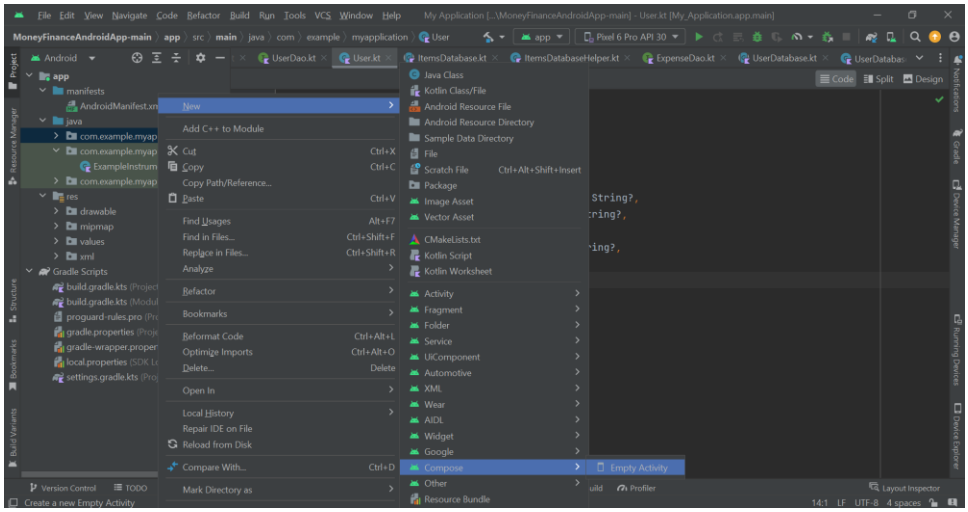


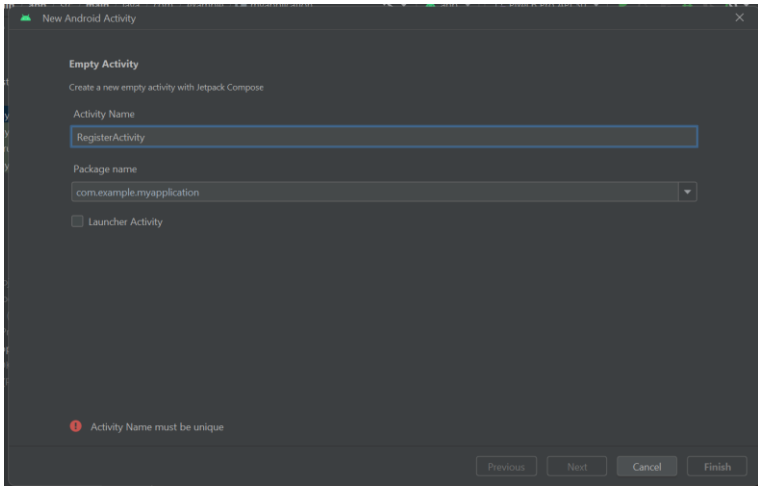
Database connection in LoginActivity.kt

```kotlin
package com.example.myapplication

import ...


class LoginActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    @OptIn(ExperimentalMaterial3Api::class)
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper( context: this)
        setContent {
            MyApplicationTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = Color.White
                ) {
                    LoginScreen( context: this, databaseHelper)
                }
            }
        }
    }
}
```

Complete Code :
https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

Step 2 : Creating RegisterActivity.kt with database



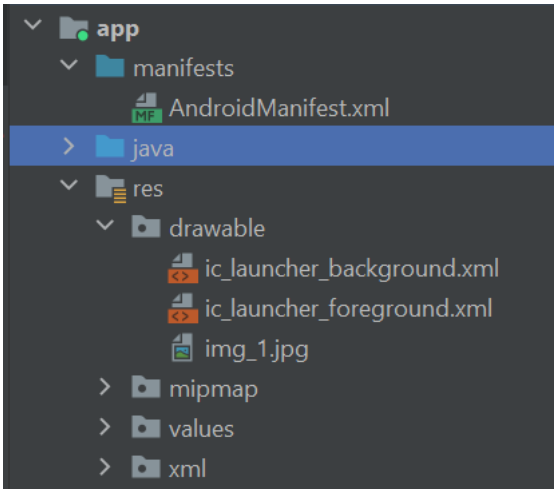


Database connection in RegisterActivity.kt

```kotlin
package com.example.myapplication

import ...

class RegisterActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    @OptIn(ExperimentalMaterial3Api::class)
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper( context: this)
        setContent {
            MyApplicationTheme {
                    // A surface container using the 'background' color from the theme
                    Surface(
                        modifier = Modifier.fillMaxSize(),
                        color = Color.White
                    ) {

                        RegistrationScreen( context: this, databaseHelper)
                    }
            }
        }
    }
```

Complete code in below link:
https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

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 Download the required drawable from the code: https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/res/drawable

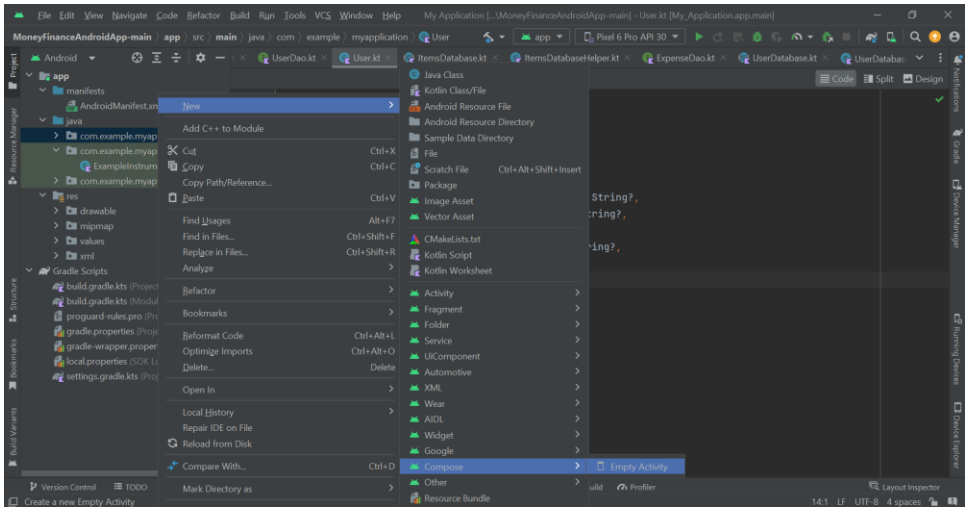
MainActivity.kt

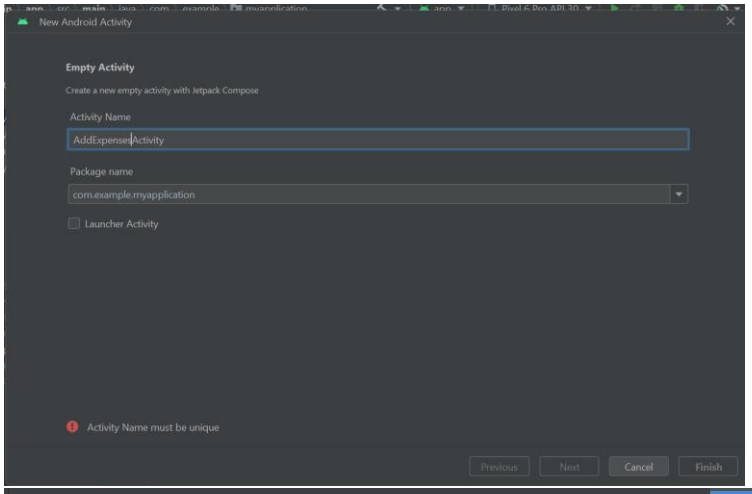
```
package com.example.myapplication

import ...

class MainActivity : ComponentActivity() {
    @OptIn(ExperimentalMaterial3Api::class)
    @SuppressLint("UnusedMaterialScaffoldPaddingParameter",
        "UnusedMaterial3ScaffoldPaddingParameter"
    )
    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(
                        modifier = Modifier.height(80.dp),
                        // along with that we are specifying
                        // title for our top bar.
                        content = { this: RowScope
```

Complete code in below link:

https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

Step 4 : Creating AddExpensesActivity.kt file



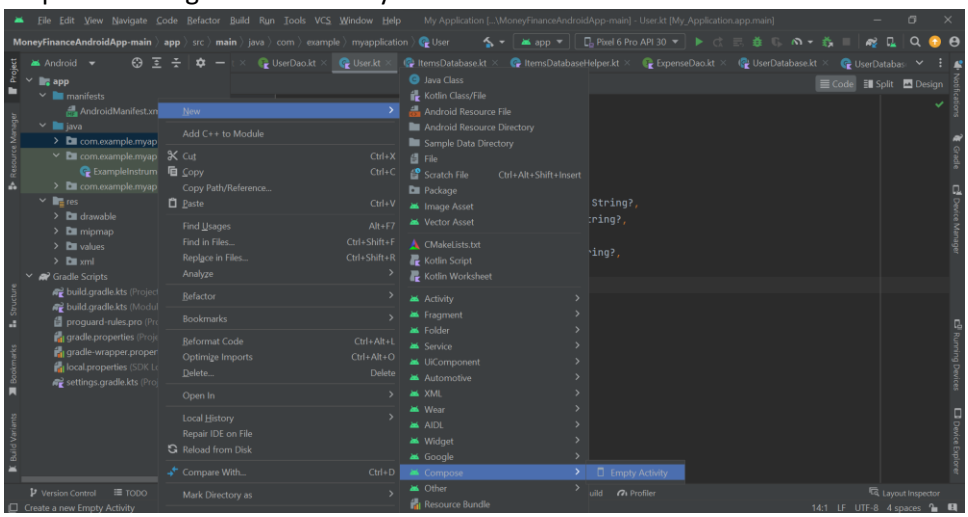


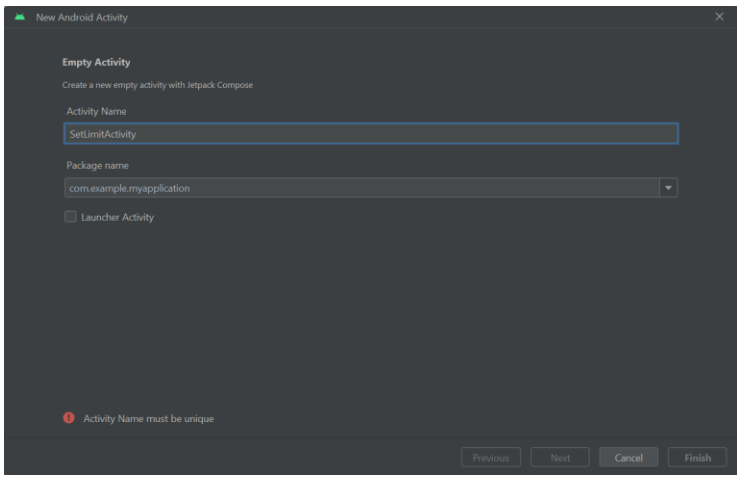

```
1   package com.example.myapplication
2
3   import ...
36
37  class AddExpensesActivity : ComponentActivity() {
38      private lateinit var itemsDatabaseHelper: ItemsDatabaseHelper
39      private lateinit var expenseDatabaseHelper: ExpenseDatabaseHelper
40      @OptIn(ExperimentalMaterial3Api::class)
41      @SuppressLint("UnusedMaterialScaffoldPaddingParameter",
42          "UnusedMaterial3ScaffoldPaddingParameter"
43      )
44      override fun onCreate(savedInstanceState: Bundle?) {
45          super.onCreate(savedInstanceState)
46          itemsDatabaseHelper = ItemsDatabaseHelper( context: this)
47          expenseDatabaseHelper = ExpenseDatabaseHelper( context: this)
48          setContent {
49              Scaffold(
50                  // in scaffold we are specifying top bar.
51                  bottomBar = {
52                      // inside top bar we are specifying
53                      // background color.
54                      BottomAppBar(
55                          modifier = Modifier.height(80.dp),
```

Complete code in below link :

https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

Step 5 : Creating SetLimitActivity.kt file



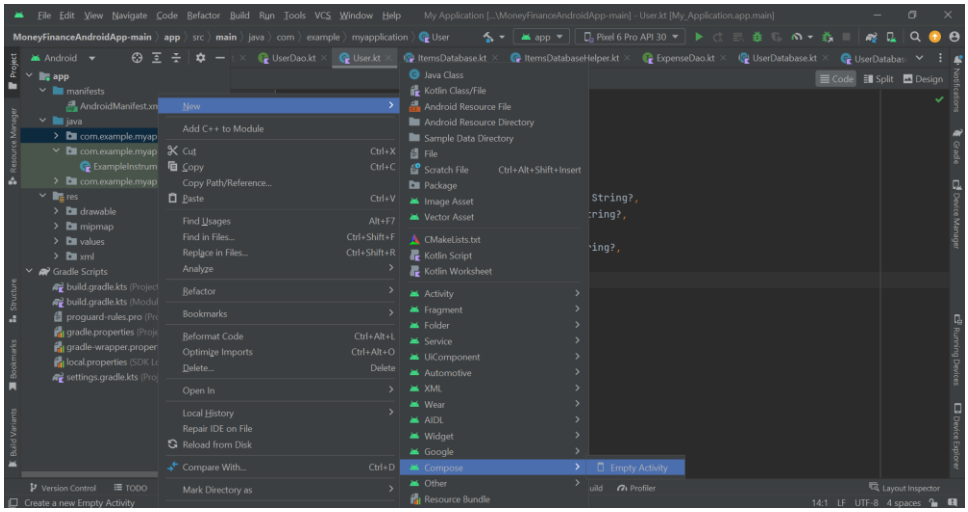


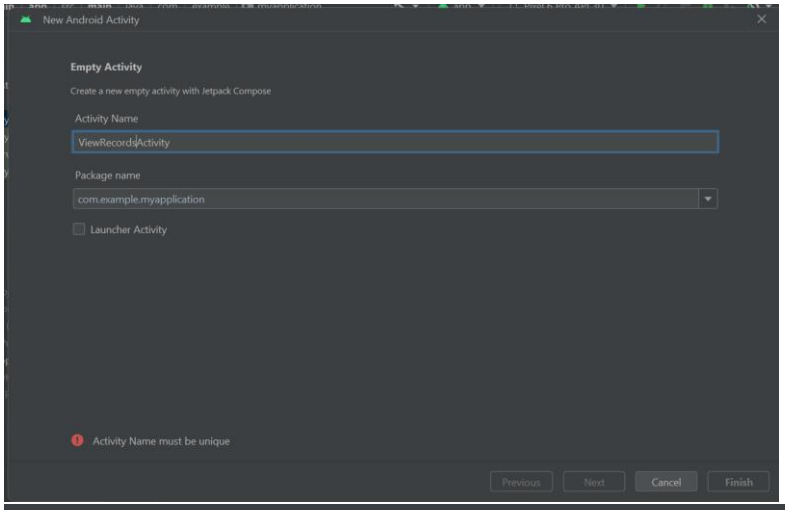
```
package com.example.myapplication

import ...

class SetLimitActivity : ComponentActivity() {
    private lateinit var expenseDatabaseHelper: ExpenseDatabaseHelper
    @OptIn(ExperimentalMaterial3Api::class)
    @SuppressLint("UnusedMaterialScaffoldPaddingParameter",
        "UnusedMaterial3ScaffoldPaddingParameter"
    )
    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(
                        modifier = Modifier.height(80.dp),
                        // along with that we are specifying
```

Complete code in below link :

https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

Step 6 : Creating ViewRecordsActivity.kt file






```
package com.example.myapplication

import ...

@ExperimentalMaterial3Api
class ViewRecordsActivity : ComponentActivity() {
    private lateinit var itemsDatabaseHelper: ItemsDatabaseHelper
    @SuppressLint("UnusedMaterialScaffoldPaddingParameter", "SuspiciousIndentation",
        "UnusedMaterial3ScaffoldPaddingParameter"
    )
    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(
                        modifier = Modifier.height(80.dp),
                        // along with that we are specifying
                        // title for our top bar.
```

Complete code in below link :

https://github.com/Suraj041203/MoneyFinanceAndroidApp/tree/main/app/src/main/java/com/example/myapplication

Task 6:

Modifying AndroidManifest.xml

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

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
```

```
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplication"
        tools:targetApi="31">
        <activity
            android:name=".AddExpensesActivity"
            android:exported="false"
            android:label="@string/title_activity_add_expenses"
            android:theme="@style/Theme.MyApplication" />
        <activity
            android:name=".ViewRecordsActivity"
            android:exported="false"
            android:label="@string/title_activity_view_records"
            android:theme="@style/Theme.MyApplication" />
        <activity
            android:name=".SetLimitActivity"
            android:exported="false"
            android:label="@string/title_activity_set_limit"
            android:theme="@style/Theme.MyApplication" />
        <activity
            android:name=".RegisterActivity"
            android:exported="false"
            android:label="@string/title_activity_register"
            android:theme="@style/Theme.MyApplication" />
        <activity

            android:name=".MainActivity"
            android:exported="false"
            android:label="@string/app_name"
            android:theme="@style/Theme.MyApplication" />

        <activity
            android:name=".LoginActivity"
            android:exported="true"
            android:label="@string/title_activity_login"
            android:theme="@style/Theme.MyApplication" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

        </activity>
    </application>

</manifest>
```

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

Complete AndroidManifest.xml code:

https://github.com/Suraj041203/MoneyFinanceAndroidApp/blob/main/app/src/main/AndroidManifest.xml

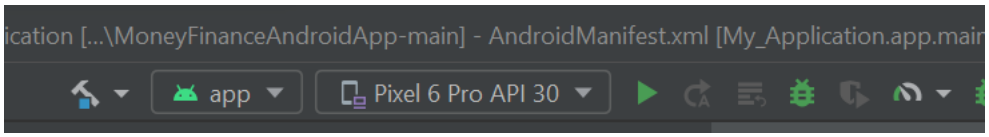
Task 7:

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

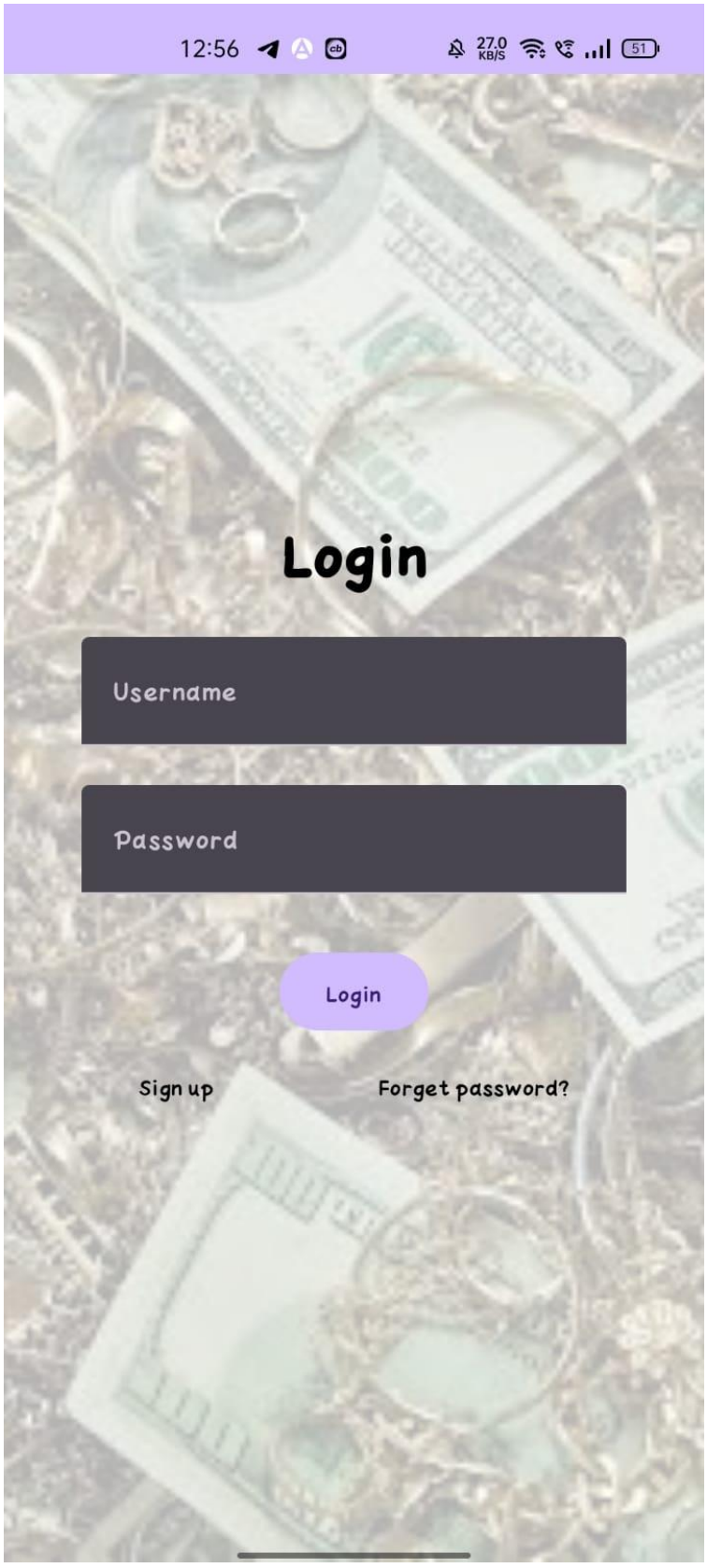


Complete Project Link:

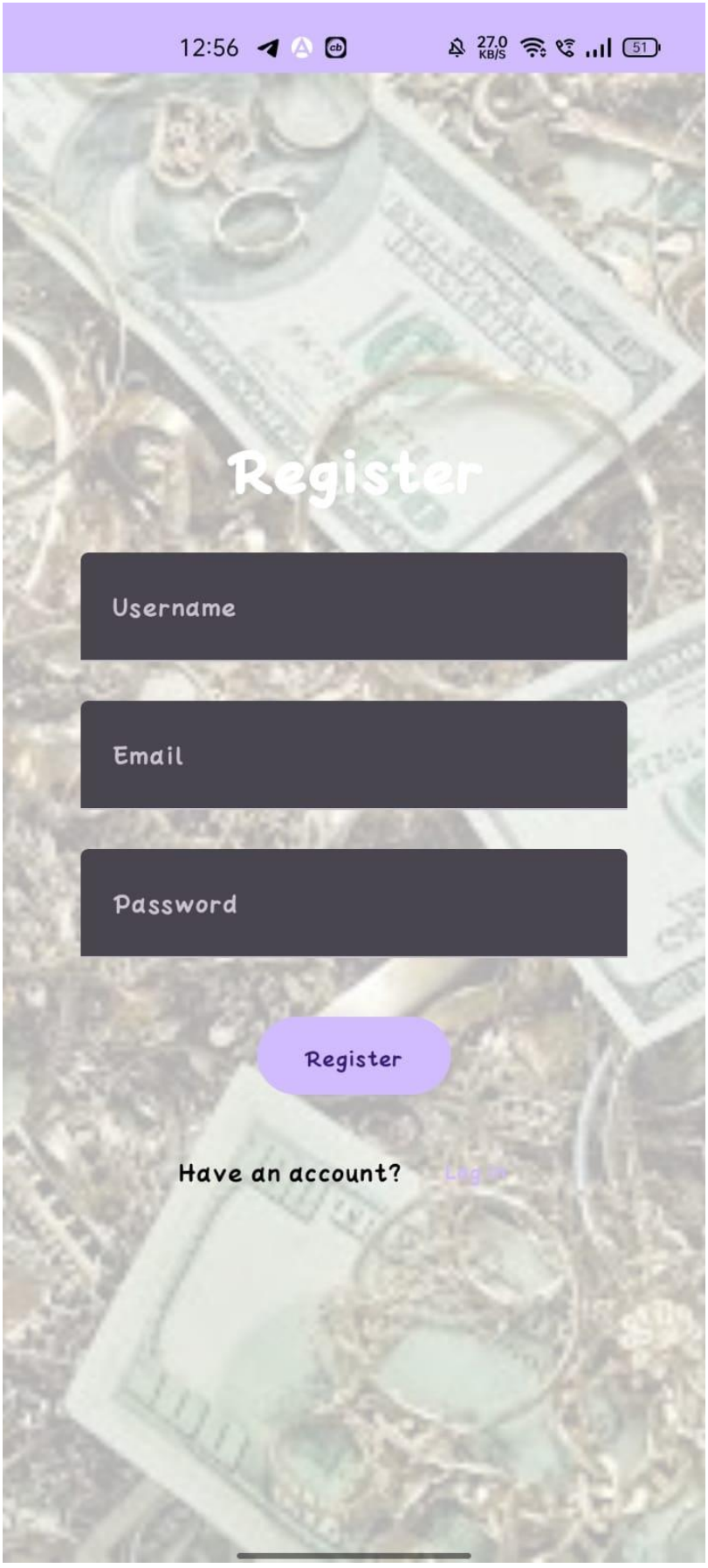
https://github.com/Suraj041203/MoneyFinanceAndroidApp.git

Final Output of the Application :

Login Page :



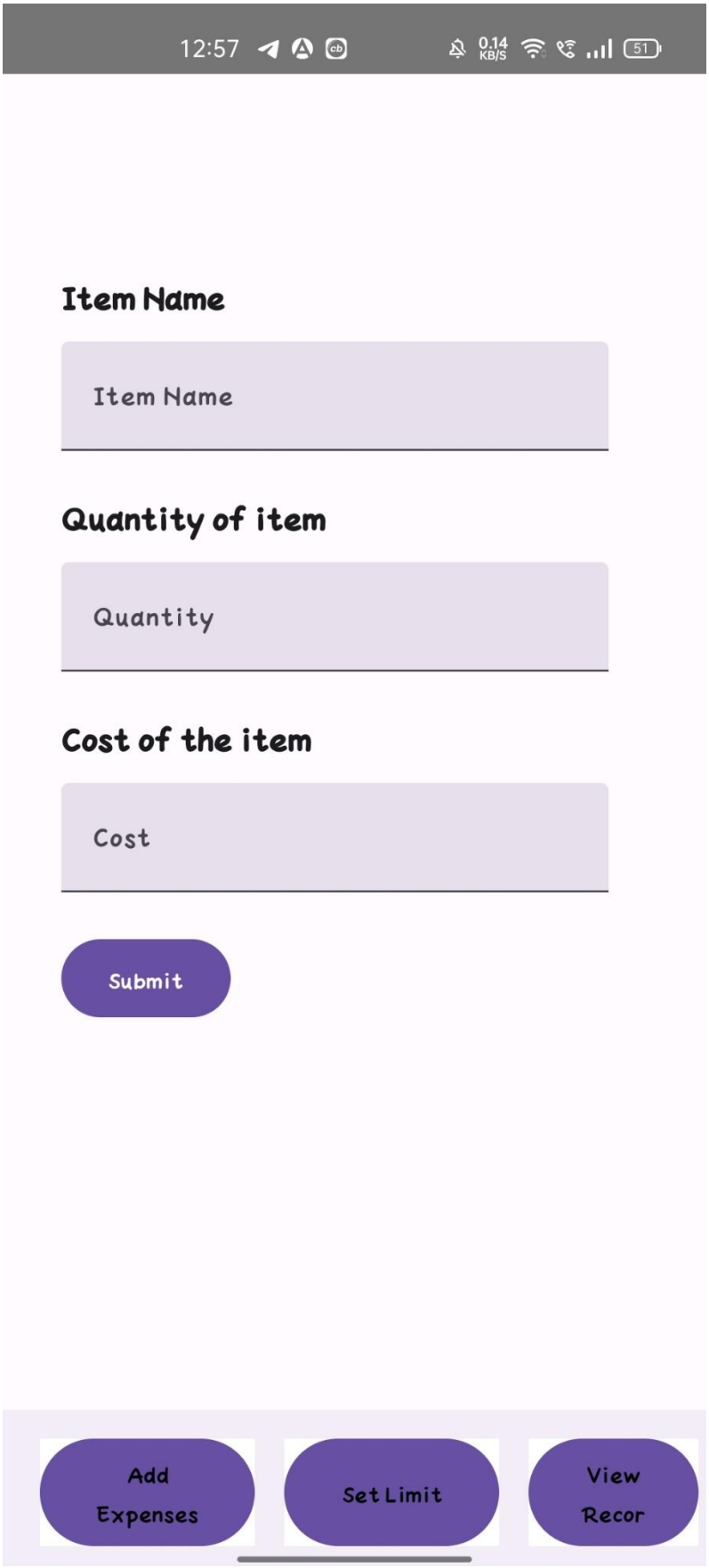
Register Page :



Main Page :

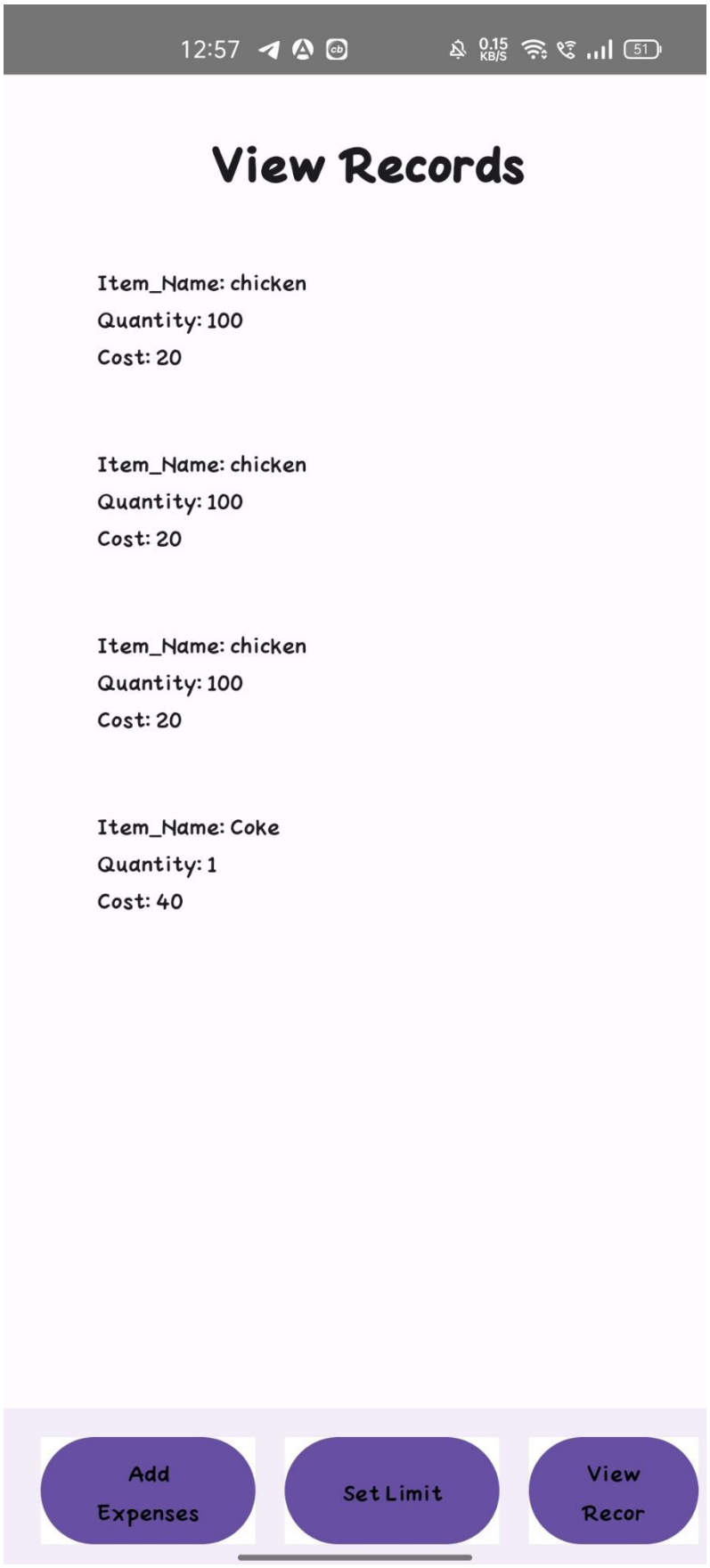

12:56
0.20 KB/S
51
Welcome To
Expense Tracker
Add Expenses
Set Limit
View Recor

Add Expenses page:



Set Limit Page before adding any data in expenses:

View Records Page:



Set Limit Page After adding expenses in add expense page: