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.

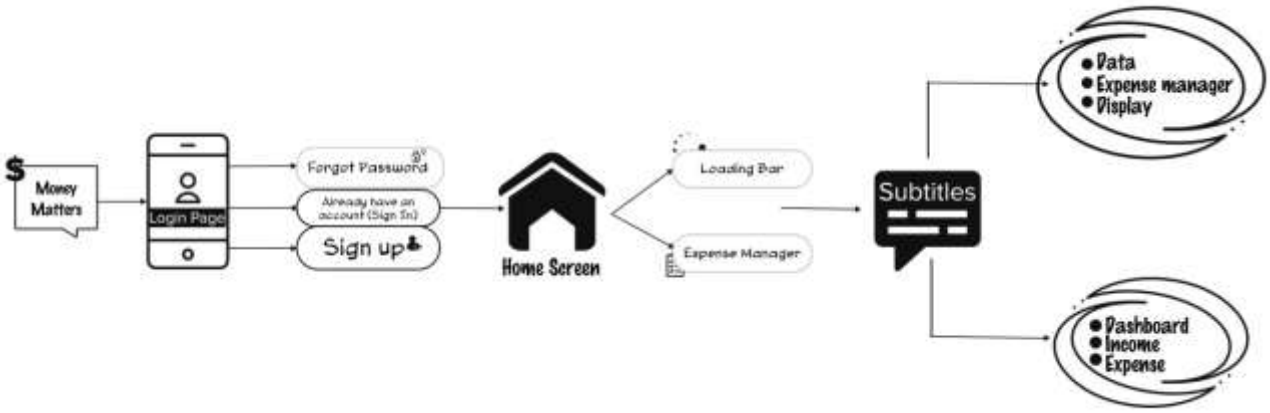
Introduction:

In a world driven by economic forces and financial stability, the significance of money in our lives cannot be overstated. Our personal financial well-being often serves as a barometer of our overall quality of life, affecting not only our ability to meet basic needs but also our capacity to pursue our dreams and aspirations. However, the complex and dynamic nature of financial systems, coupled with the ever-present possibility of unexpected financial challenges, means that nearly everyone encounters financial problems at some point in their lives.

This journey into the realm of "Money Matters: A Financial Problem" invites us to explore the multifaceted issues that people face when grappling with financial difficulties. Whether it's the burden of debt, the challenges of managing a budget, the stress of living paycheck to paycheck, or the uncertainties of investment and retirement planning, financial problems come in a myriad of forms and can impact individuals and families in profound ways.

This exploration aims to shed light on the common challenges and potential solutions that individuals encounter as they navigate the often-treacherous waters of personal finance. By delving into the complexities of financial problems, we hope to empower you with the knowledge, insights, and strategies necessary to confront and conquer your financial challenges, turning them from obstacles into opportunities for growth and financial security. So, let's embark on this journey through "Money Matters," as we unravel the intricacies of financial difficulties and discover pathways to a more secure and prosperous financial future.

Architecture:



Prerequisites: **<u>Before delving into the intricate world of "Money Matters: A Financial Problem," it's essential to acknowledge certain prerequisites that will help you navigate and understand this topic more effectively. Here are some fundamental prerequisites:</u>**

Basic Financial Literacy:

To effectively address financial problems, you should have a foundational understanding of financial concepts. This includes knowledge of budgeting, saving, investing, debt management, and financial goals. If you lack this knowledge, consider starting with financial literacy courses, books, or online resources to build a solid financial foundation.

Record Keeping:

Maintaining accurate financial records is vital for assessing your financial situation. This includes keeping track of bills, bank statements, investment accounts, and other financial documents. A good record-keeping system ensures you have a complete picture of your finances.

Willingness to Seek Help:

Acknowledge that financial problems can become overwhelming, and seeking professional help or advice is a sign of responsibility, not weakness. Be open to consulting financial advisors, credit counselors, or experts in areas where you lack expertise.

Patience and Persistence:

Resolving financial problems often requires time and persistence. Understand that the journey to financial stability may involve making sacrifices, adjusting your lifestyle, and taking gradual steps toward your goals.

<u>Developing a money management or financial issue app can be a valuable tool for individuals seeking to better manage their finances. Here are some key components and features commonly used to develop such an app:</u>

User Registration and Profiles:

User-friendly registration process with email or social media login options. Ability for users to create and customize profiles, including personal and financial information.

Features to set savings goals and automate savings transfers.

Investment tracking and portfolio analysis for stocks, bonds, and other assets. Retirement planning tools and calculators.

Graphs, charts, and reports to visualize financial data.

Insights into spending habits, trends, and areas where users can save money. Credit score tracking and recommendations for improving it .

Expense Tracking and Receipt Scanning:

Manual expense entry or the ability to scan and upload receipts for tracking. OCR (Optical Character Recognition) technology for automatic data extraction from receipts

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:

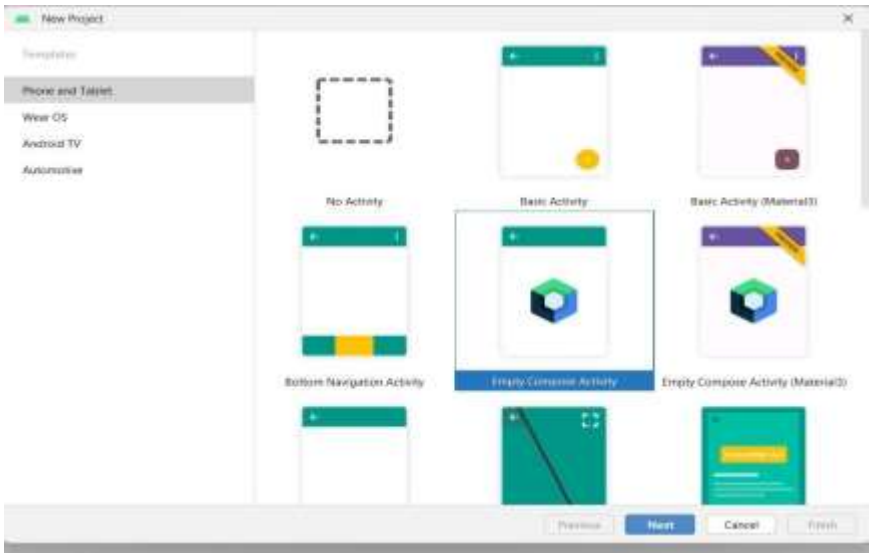
1.Required initial steps
2.Creating a new project.
3.Adding required dependencies.
4.Creating the database classes.
5.Building application UI and connecting to database.
6.Using AndroidManifest.xml 7.Running the application.

Task 1:

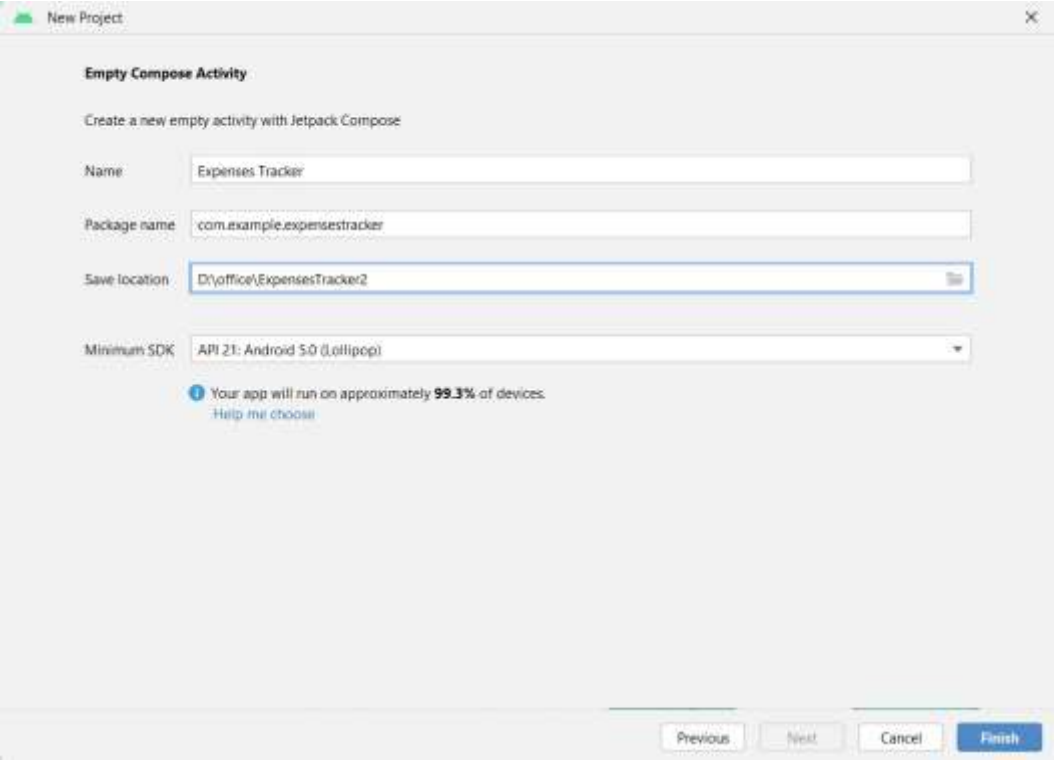
Required initial steps :

Task 2 :

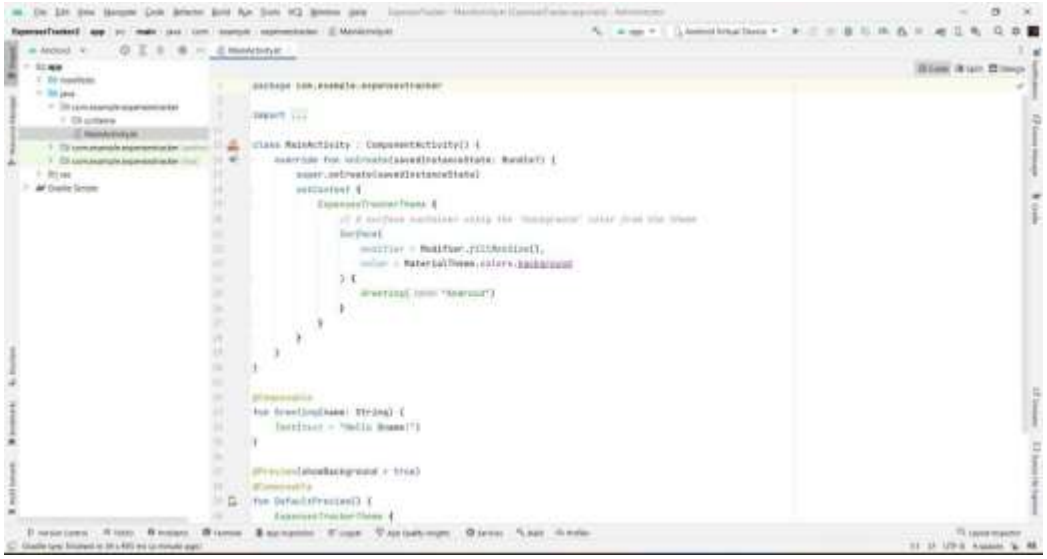
Creating a new project.
Step 1 : Android studio > File > New > New Project > Empty Compose Activity Step 2 : Click on **Next** button.



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

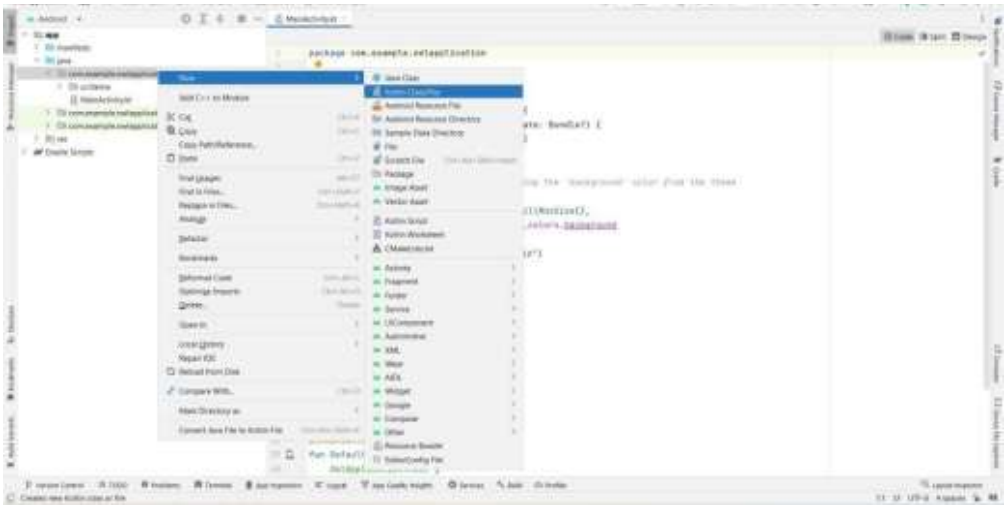
```
// Adding Room dependencies implementation
'androidx.room:room-common:2.5.0'
implementation 'androidx.room:room-ktx:2.5.0'
```

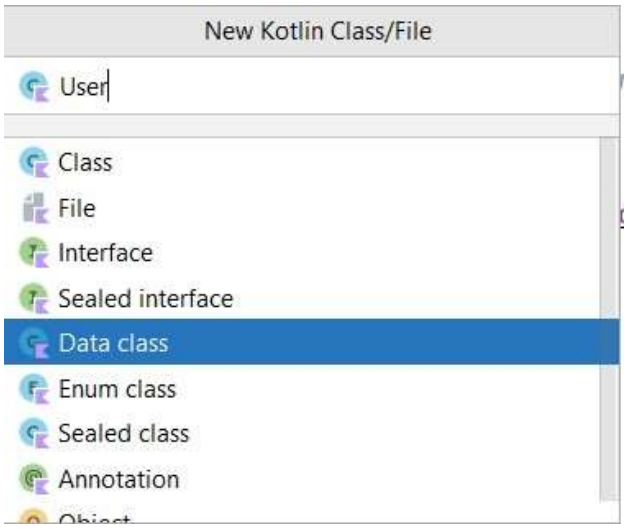
Step 3 : Click on Sync now

Task 4:

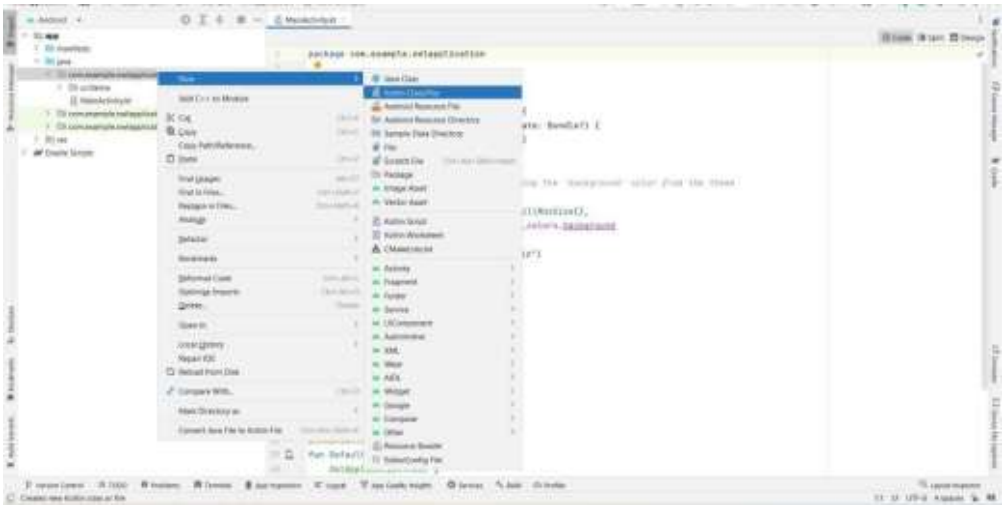
1. Creating the database classes for user login and registration.

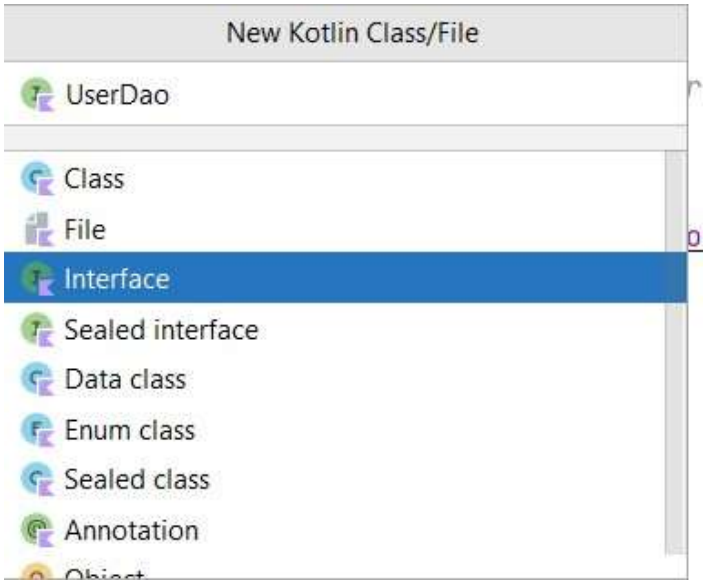
Step 1 : Create User data class



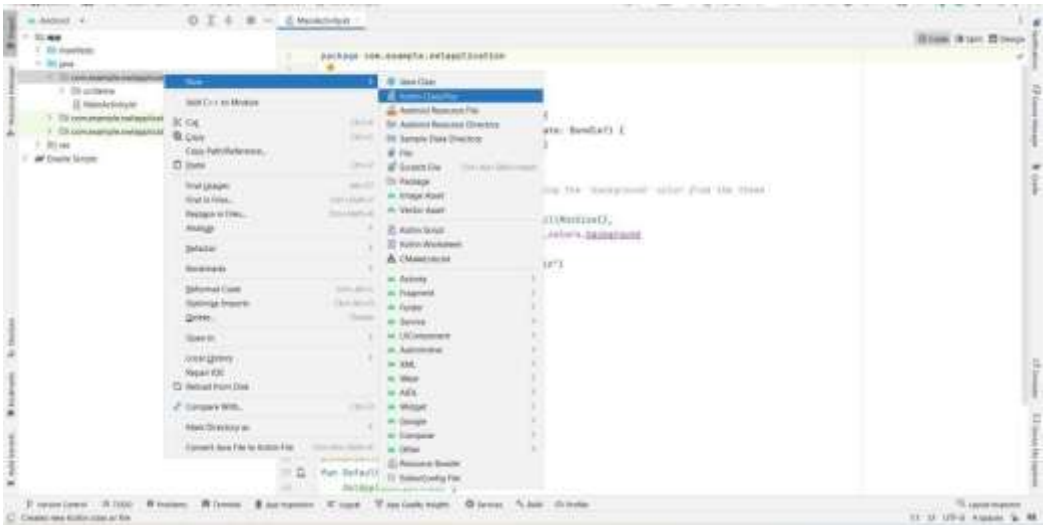


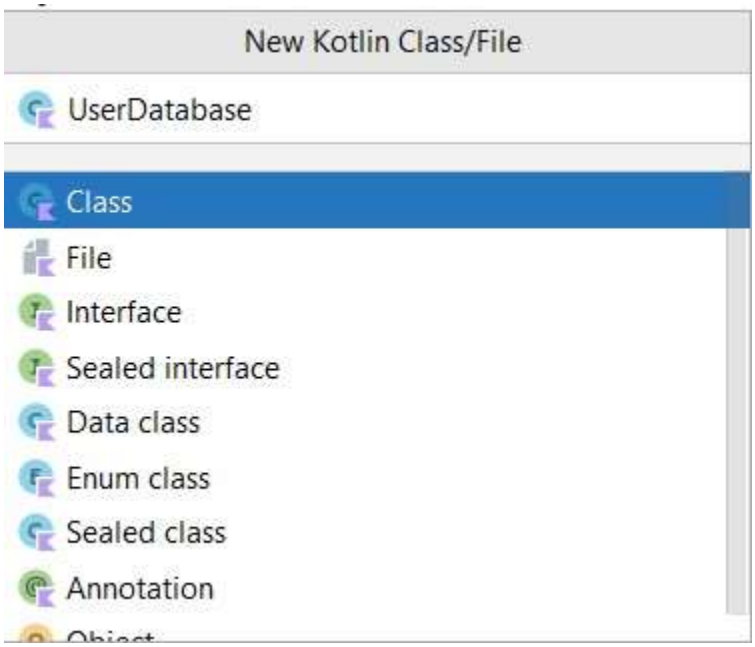
Step 2 : Create an UserDao interface



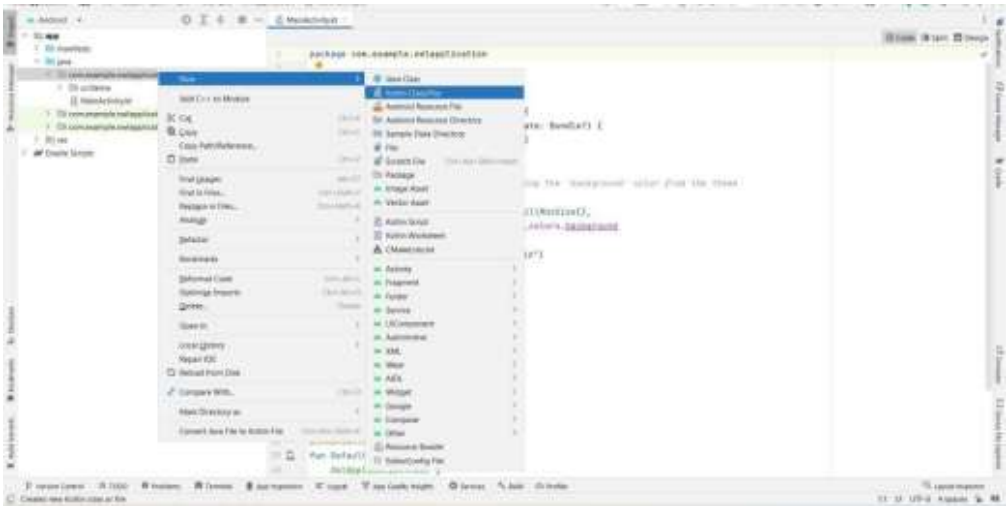


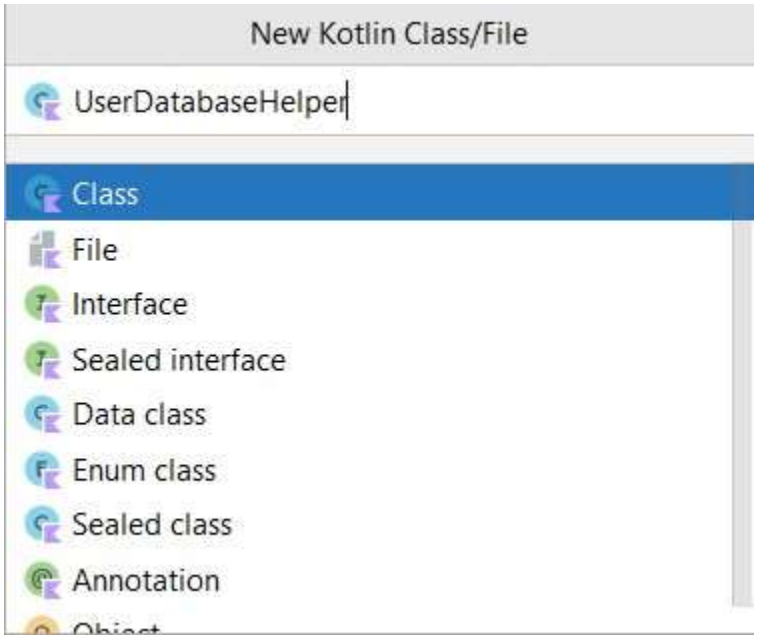
Step 3 : Create an UserDatabase class



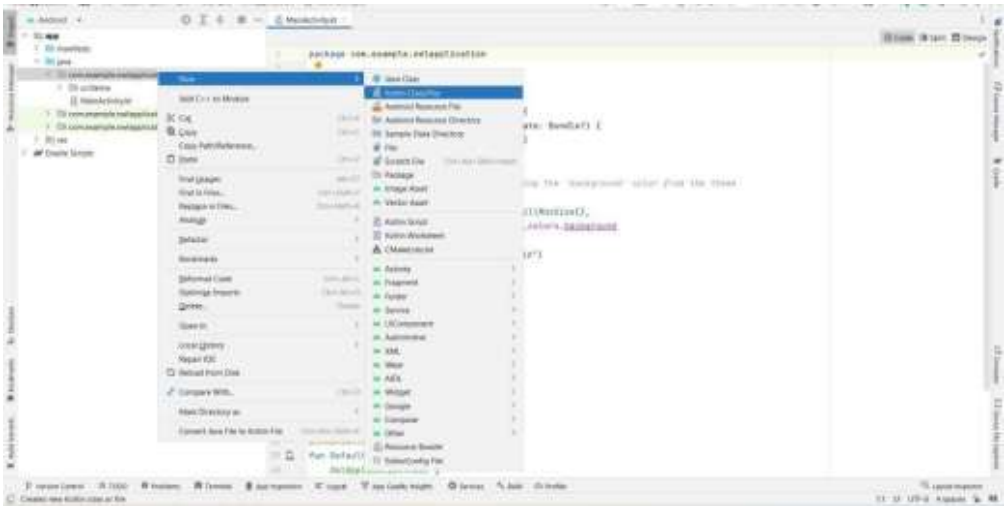


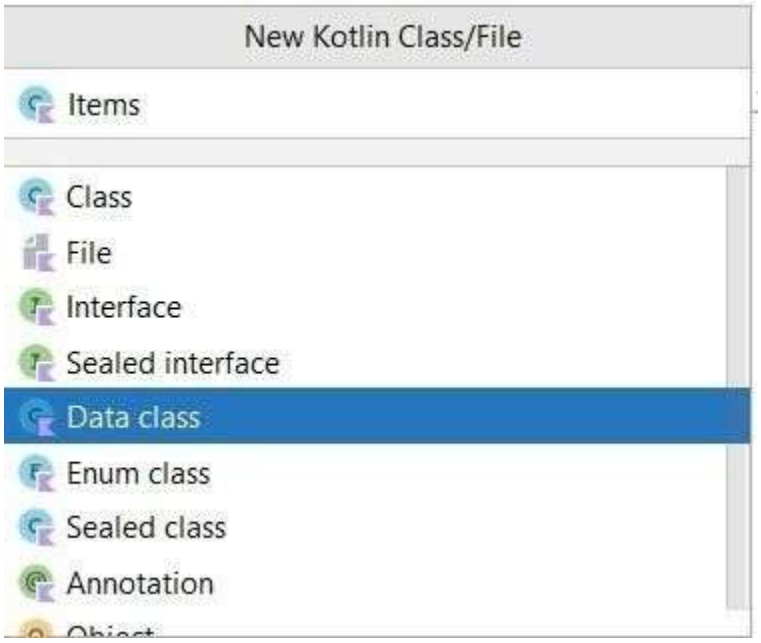
Step 4 : Create an UserDatabaseHelper class



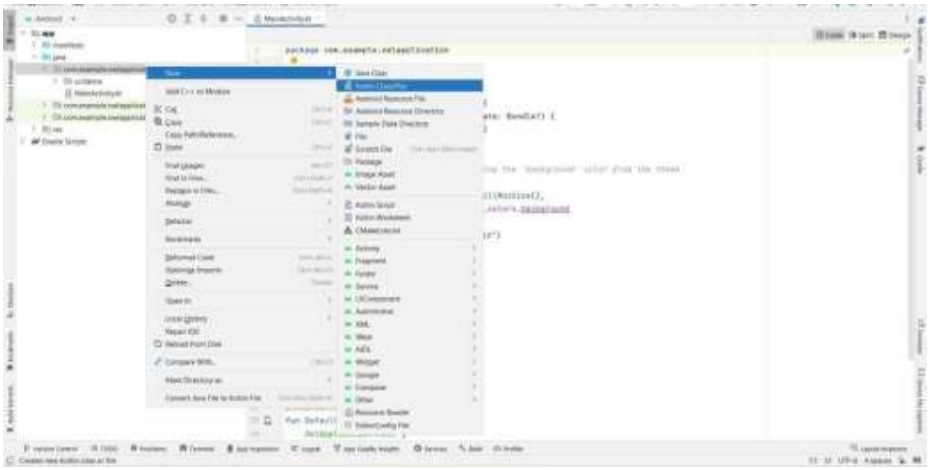


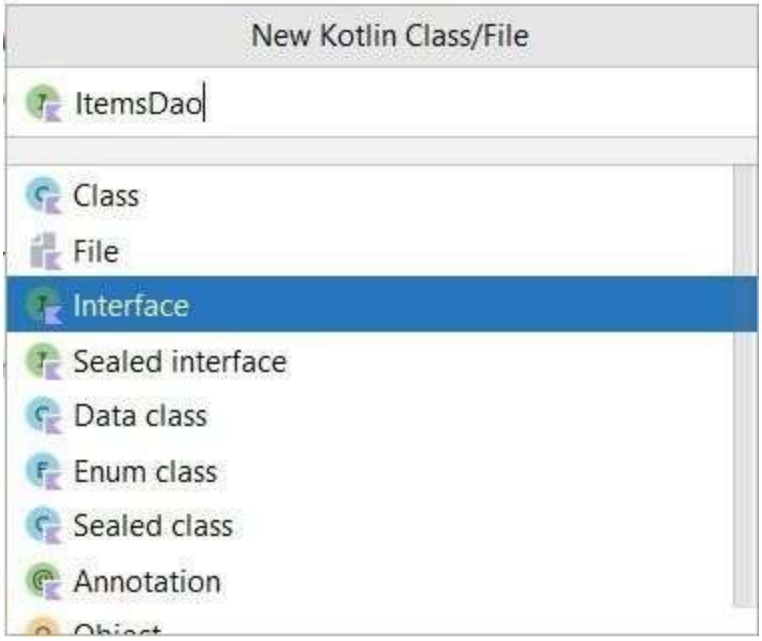
2. Creating the database classes for item name, quantity and cost.
Step 1 : Create Items data class



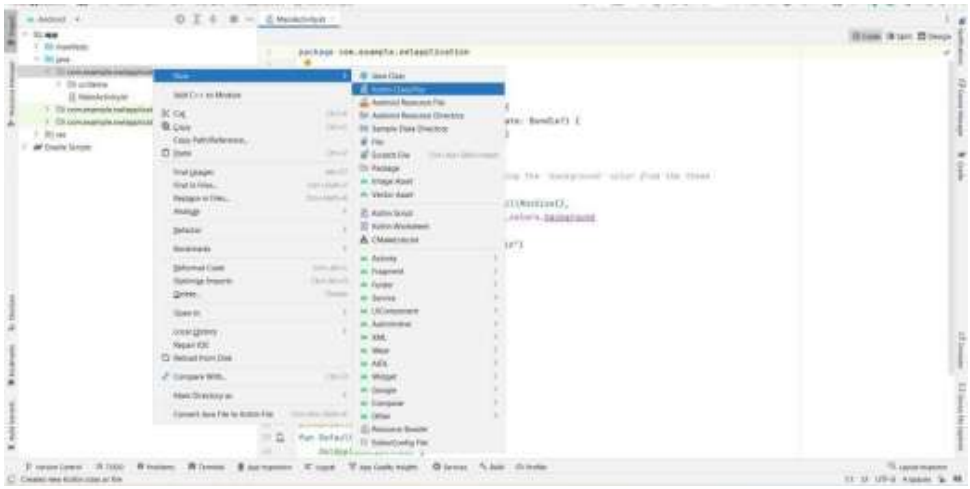


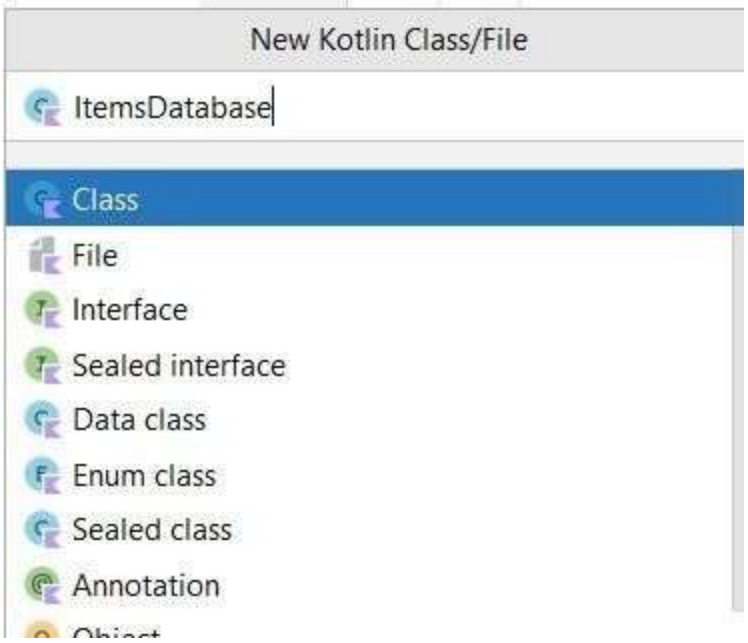
Step 2 : Create ItemsDao interface



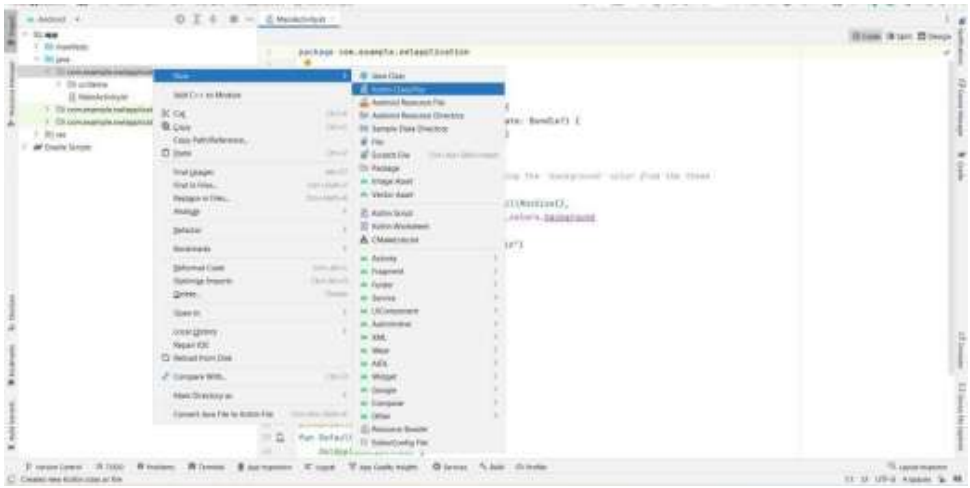


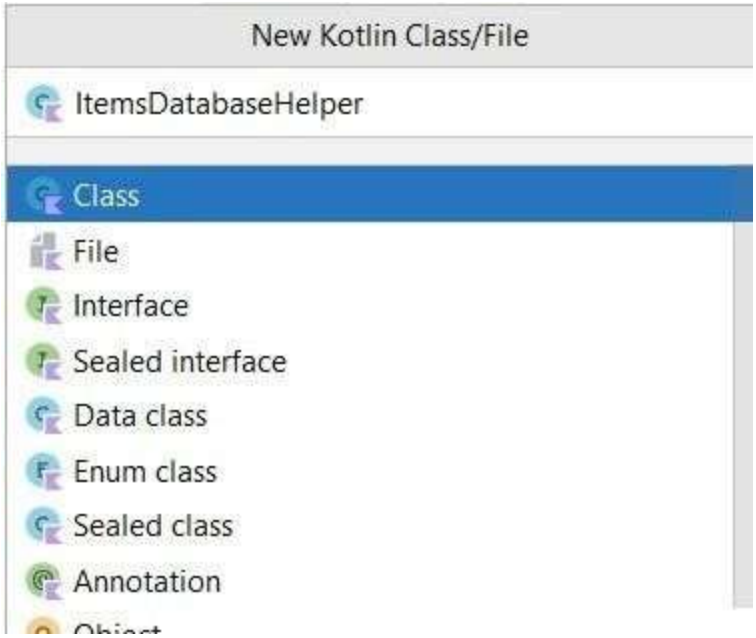
Step 3 : Create ItemsDatabse class





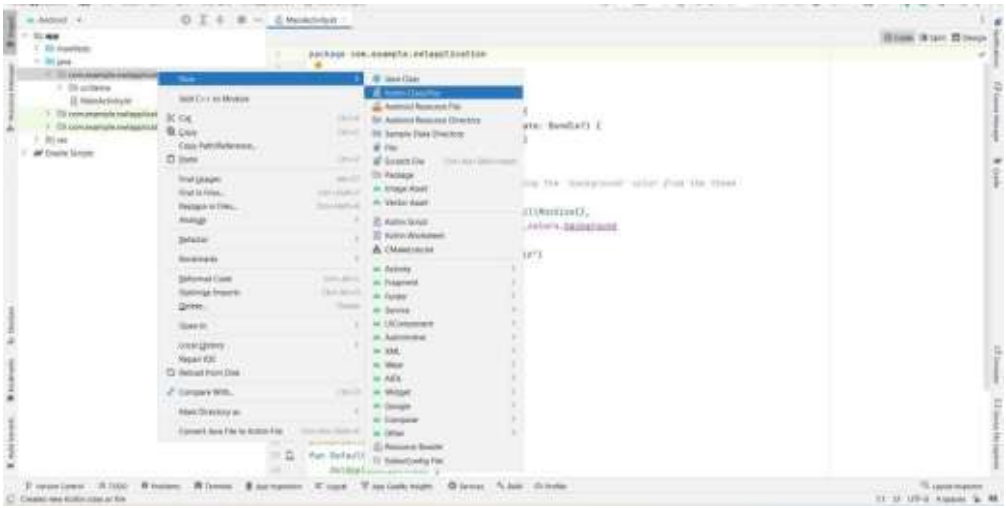
Step 4 : Create ItemsDatabaseHelper class

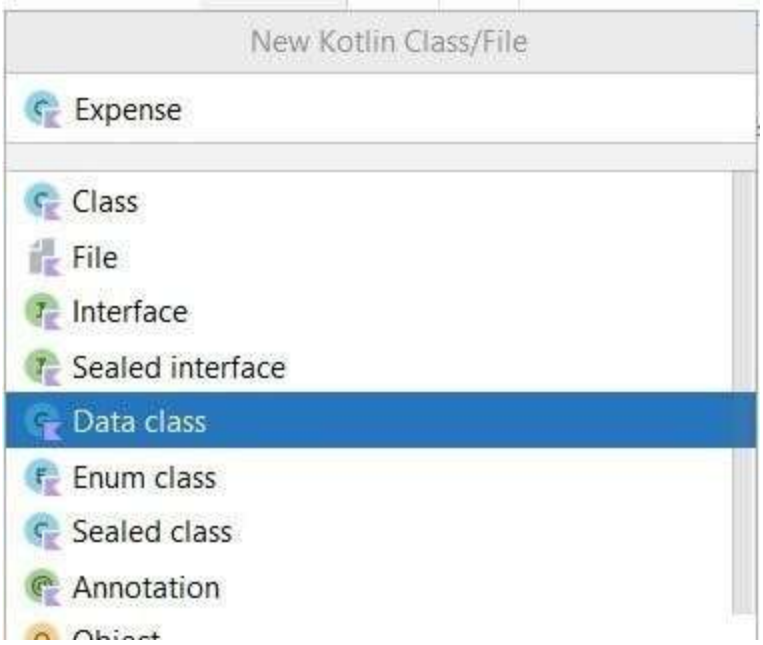




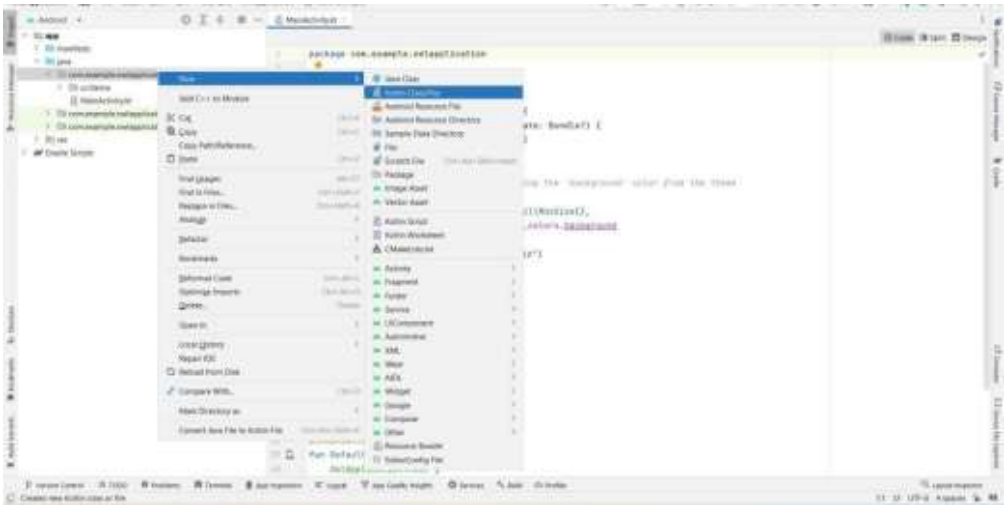
3. Creating the database classes for an amount.

Step 1 : Create Expense data class

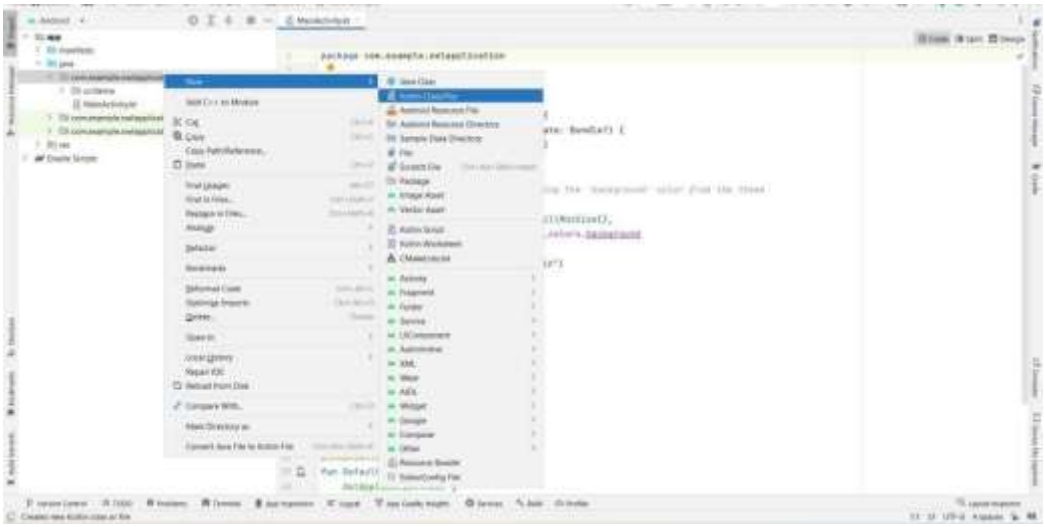


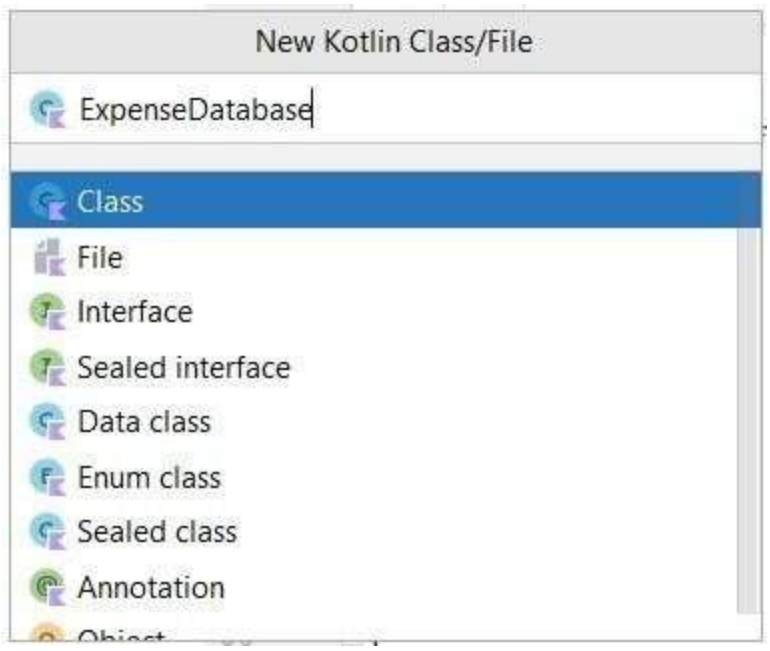


Step 2 : Create ExpenseDao interface

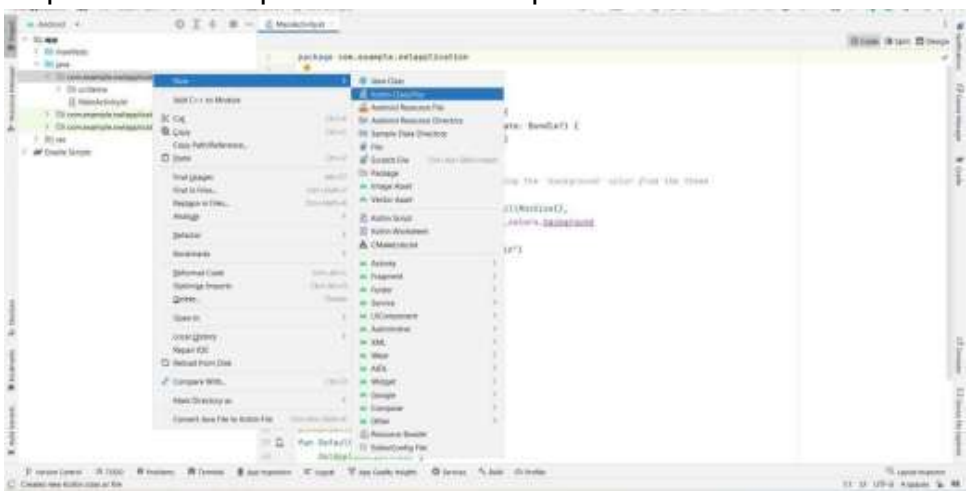


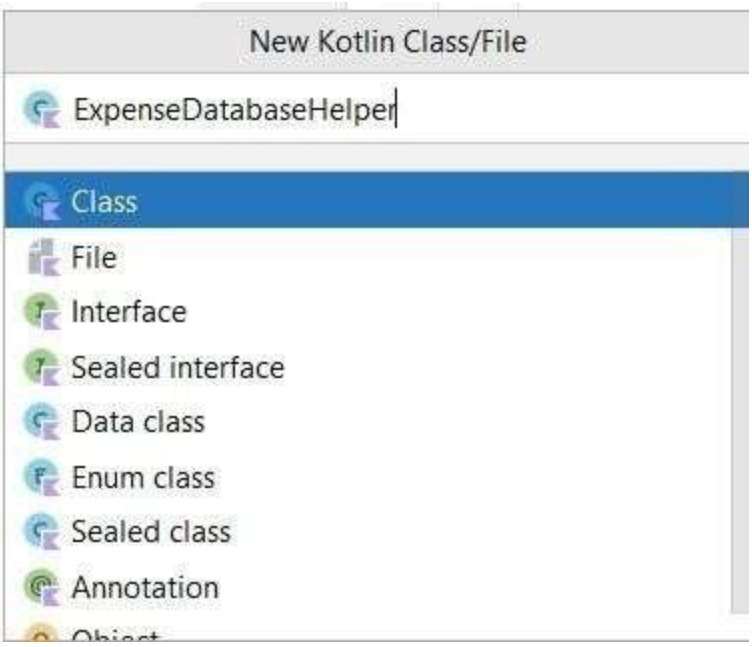
Step 3 : Create ExpenseDatabase class





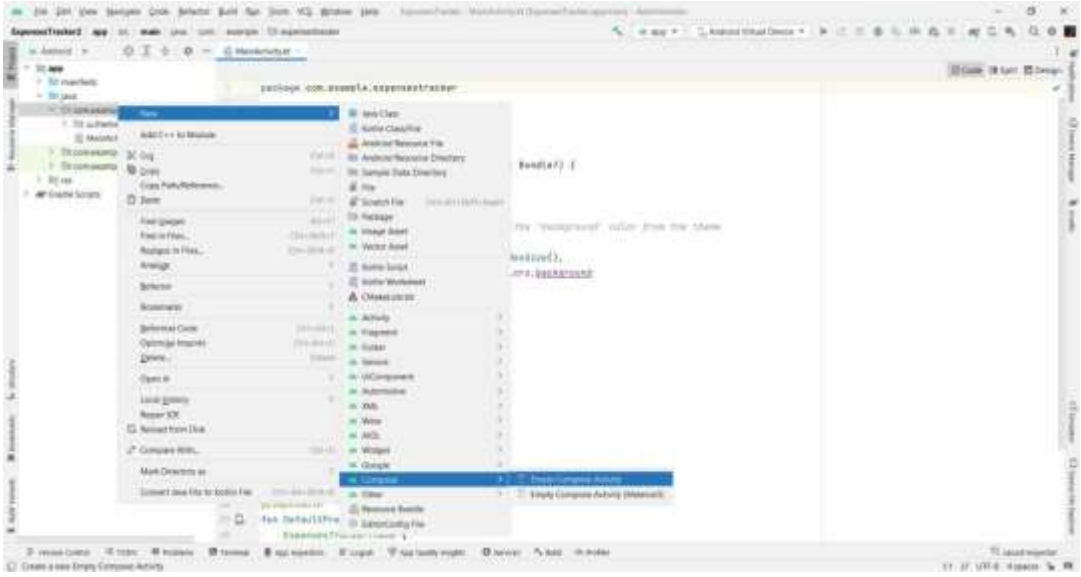
Step 4 : Create ExpenseDatabaseHelper class

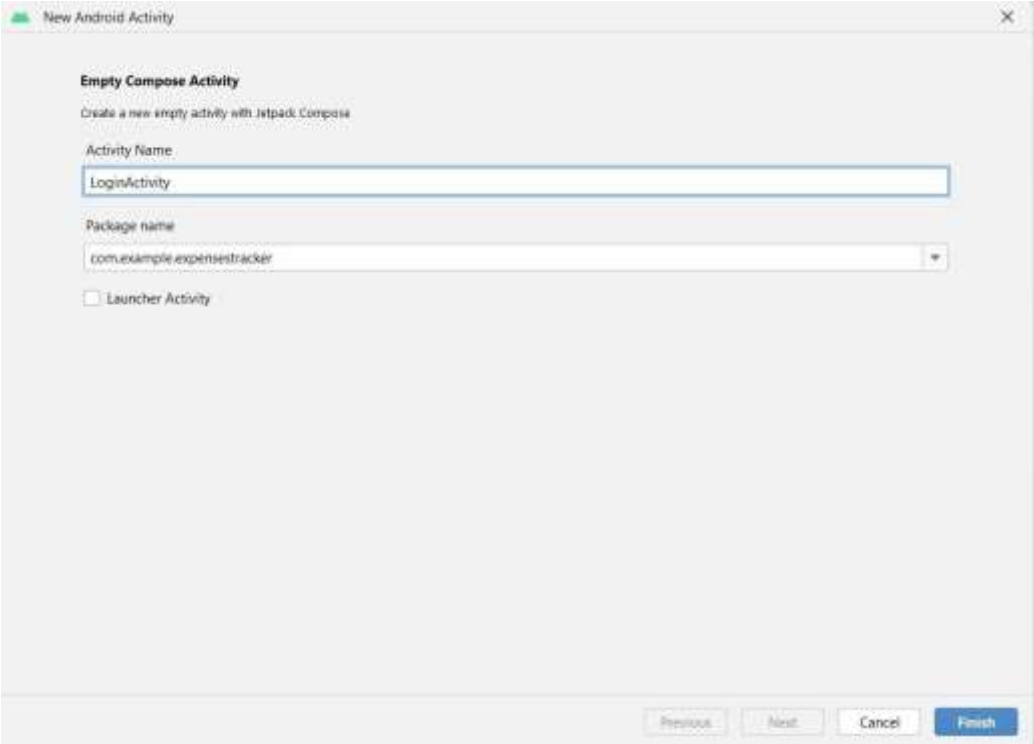




Task 5:

Building application UI and connecting to database. Step 1: Creating LoginActivity.kt with database

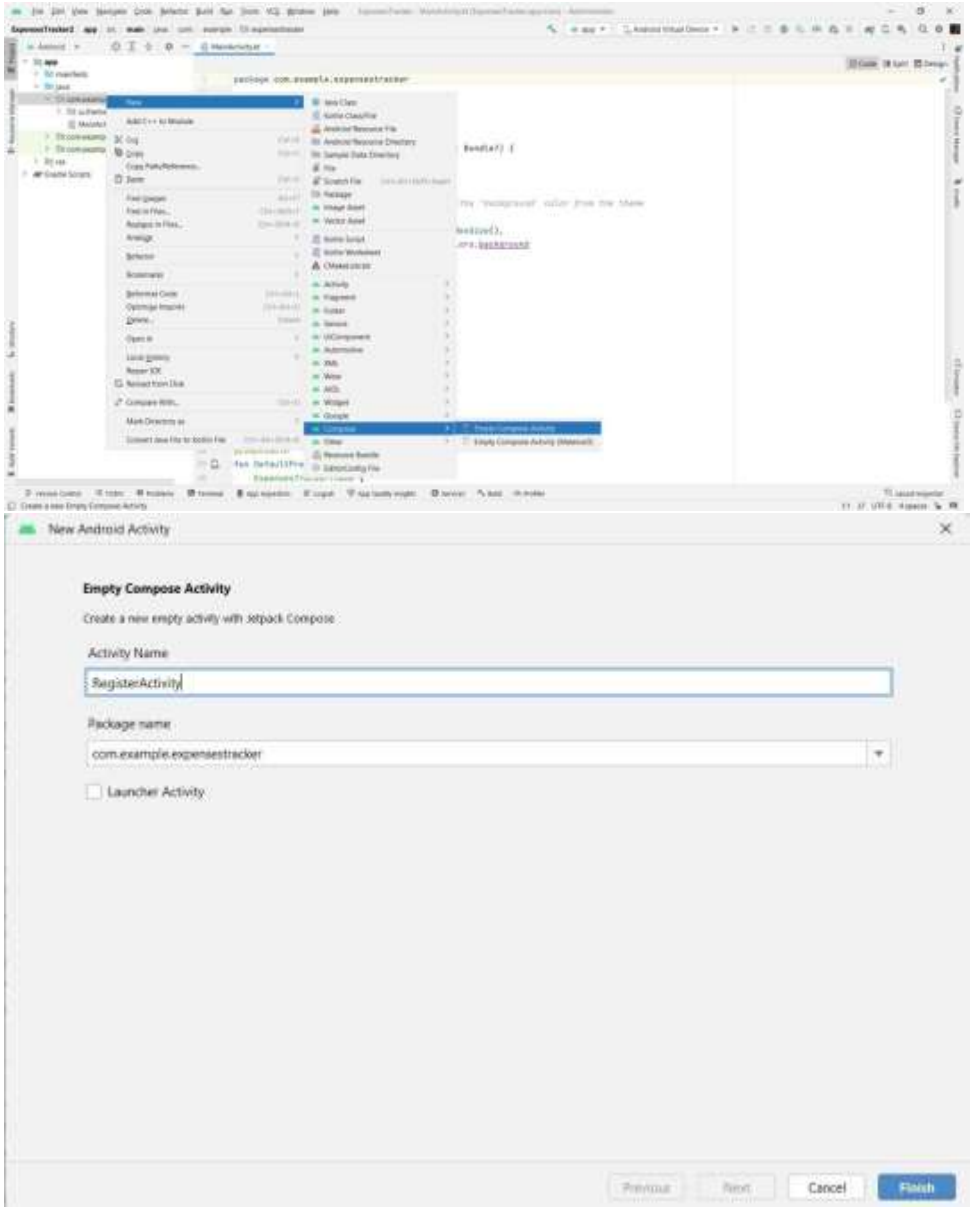




Database connection in LoginActivity.kt

```
package com.example.expensestracker

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



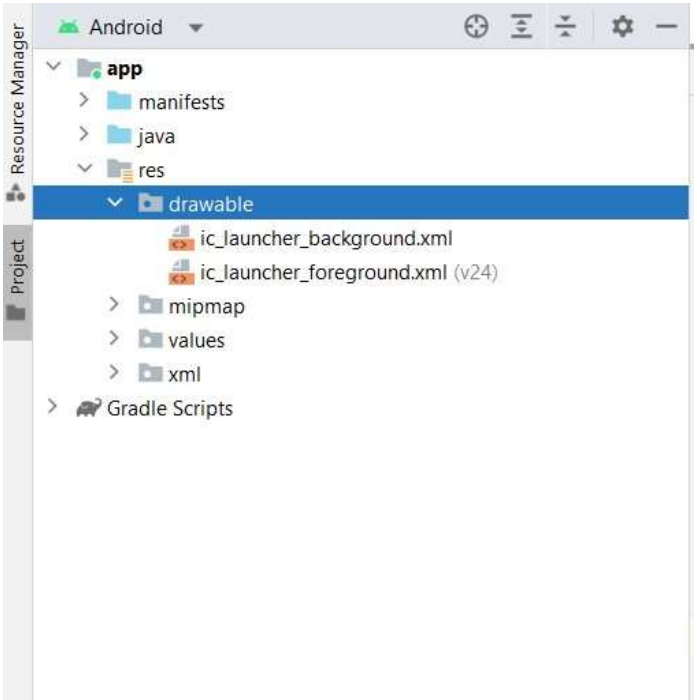
Database connection in RegisterActivity.kt

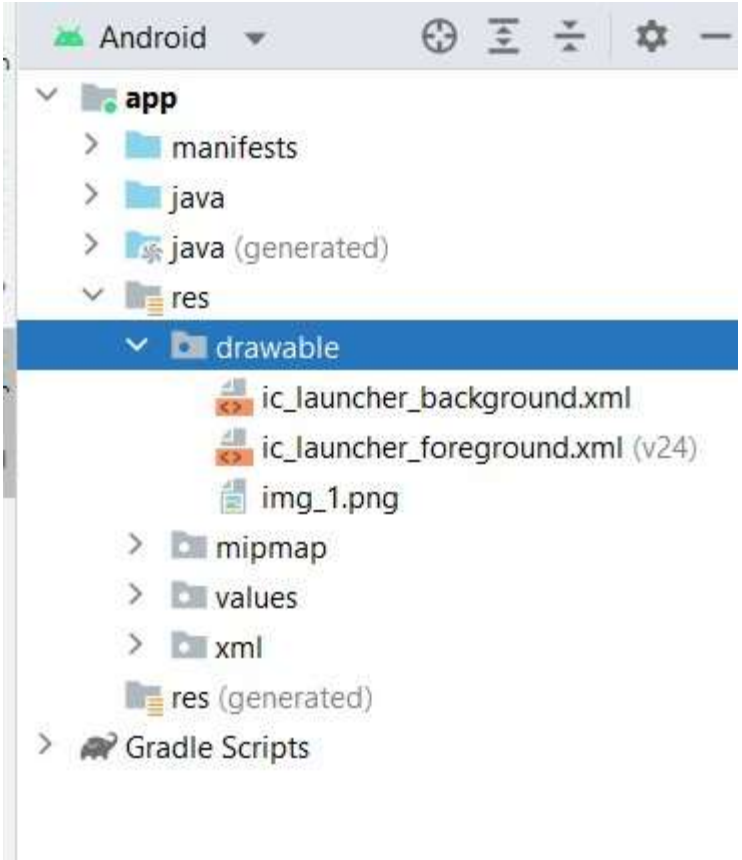
```kotlin
package com.example.expensestracker

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

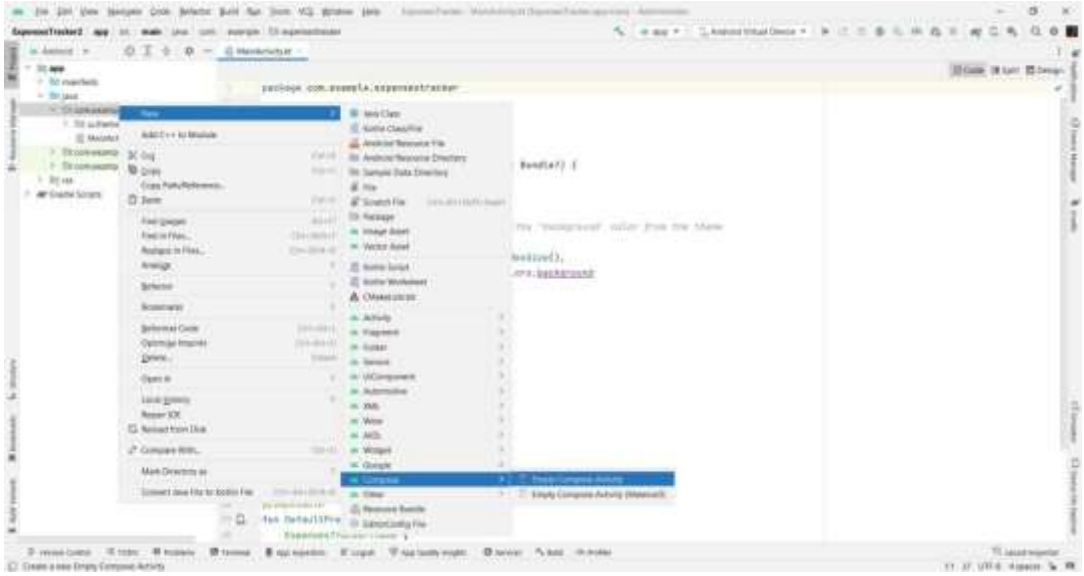


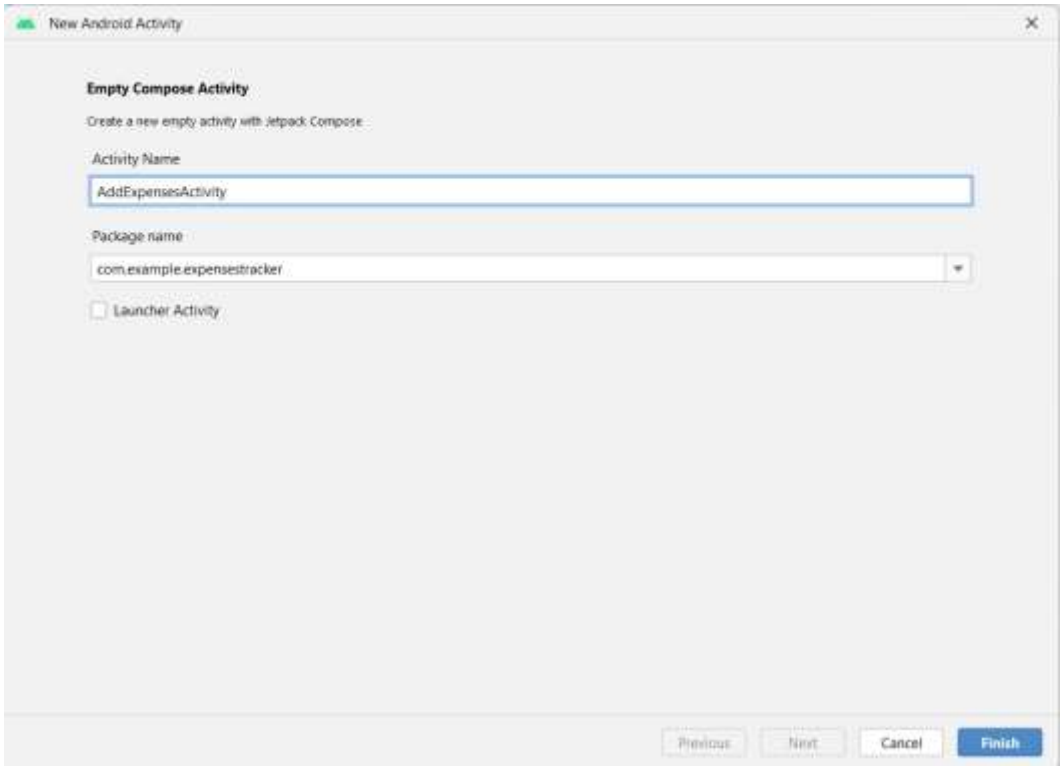


MainActivity.kt

```
package com.example.expensestracker

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( hint: 0xFFadbe4),
                        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,
```

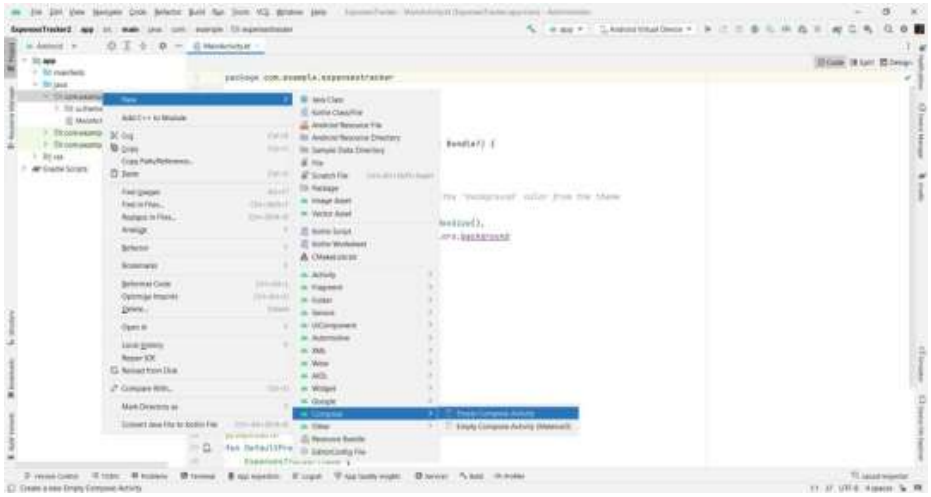
Step 4 : Creating AddExpensesActivity.kt file

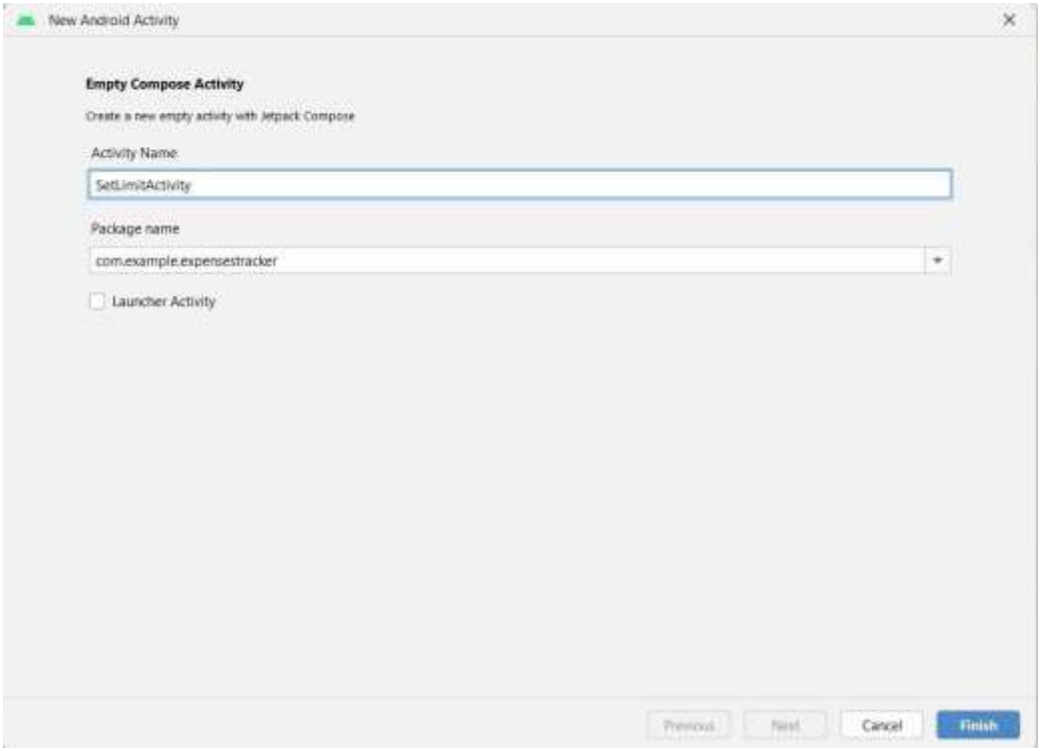



New Android Activity
Empty Compose Activity
Create a new empty activity with Jetpack Compose
Activity Name
AddExpensesActivity
Package name
com.example.expensestracker
Launcher Activity
Previous
Next
Cancel
Finish

```kotlin
package com.example.expensestracker

import ...

class AddExpensesActivity : ComponentActivity() {
    private lateinit var itemsDatabaseHelper: ItemsDatabaseHelper
    private lateinit var expenseDatabaseHelper: ExpenseDatabaseHelper
    @SuppressLint("UnusedMaterialScaffoldPaddingParameter")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        itemsDatabaseHelper = ItemsDatabaseHelper( context: this)
        expenseDatabaseHelper = ExpenseDatabaseHelper( context: this)
        setContent {
            Scaffold(
                // In scaffold we are specifying top bar.
                topBar = {
                    // inside top bar we are specifying
                    // background color.
                    BottomAppBar(backgroundColor = Color( color: 0xFFadbef4),
                        modifier = Modifier.height(80.dp),
                        // along with that we are specifying
                        // title for our top bar.
                        content = { 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)
```

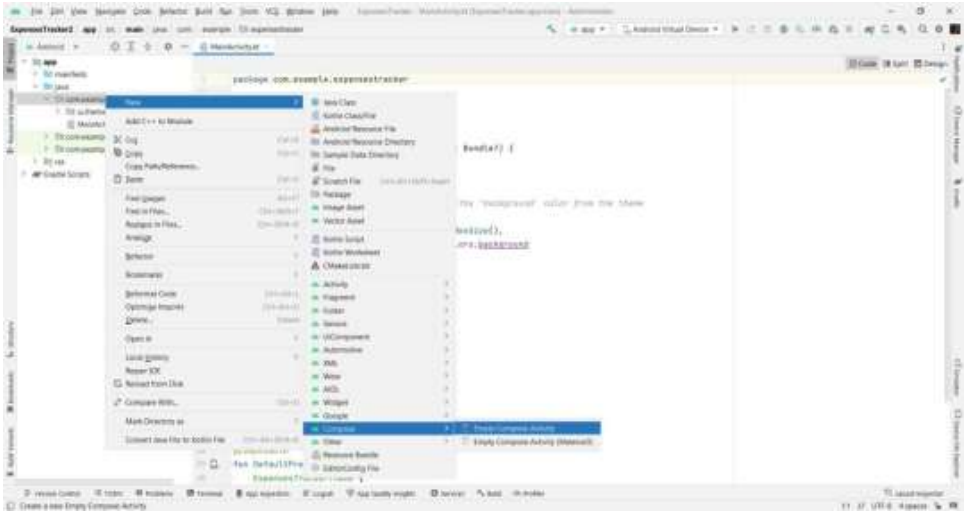
Step 5 : Creating SetLimitActivity.kt file

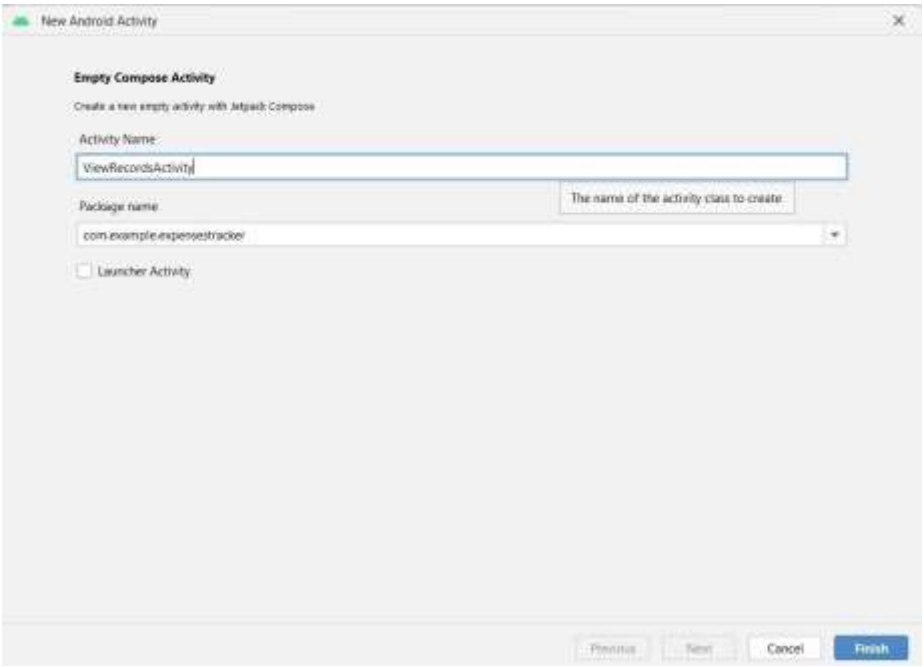





```
package com.example.expensestracker

import ...

class SetLimitActivity : ComponentActivity() {
    private lateinit var expenseDatabaseHelper: ExpenseDatabaseHelper
    @SuppressLint("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( color: 0xFFadbef4),
                        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.expensestracker

import ...

class ViewRecordsActivity : ComponentActivity() {
    private lateinit var itemsDatabaseHelper: ItemsDatabaseHelper
    @SuppressLint("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: 0xFFadbefa),
                        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:

Modifying AndroidManifest.xml

```
        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>
```

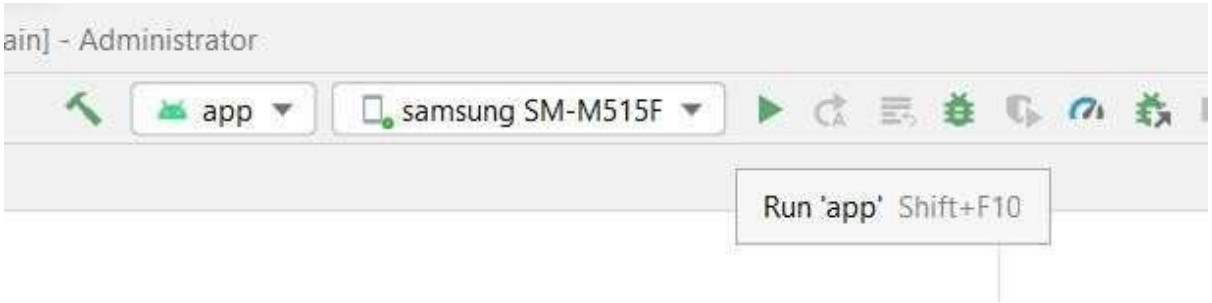
Task 7:

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



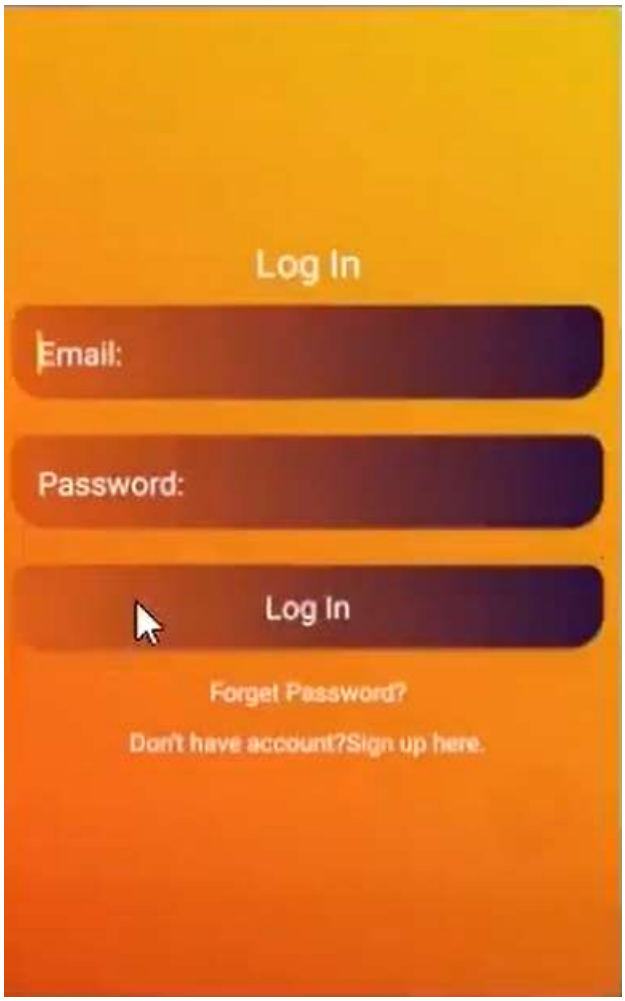
Complete Project Link:

https://github.com/smartinternz02/SI-GuidedProject-587523-1696929616

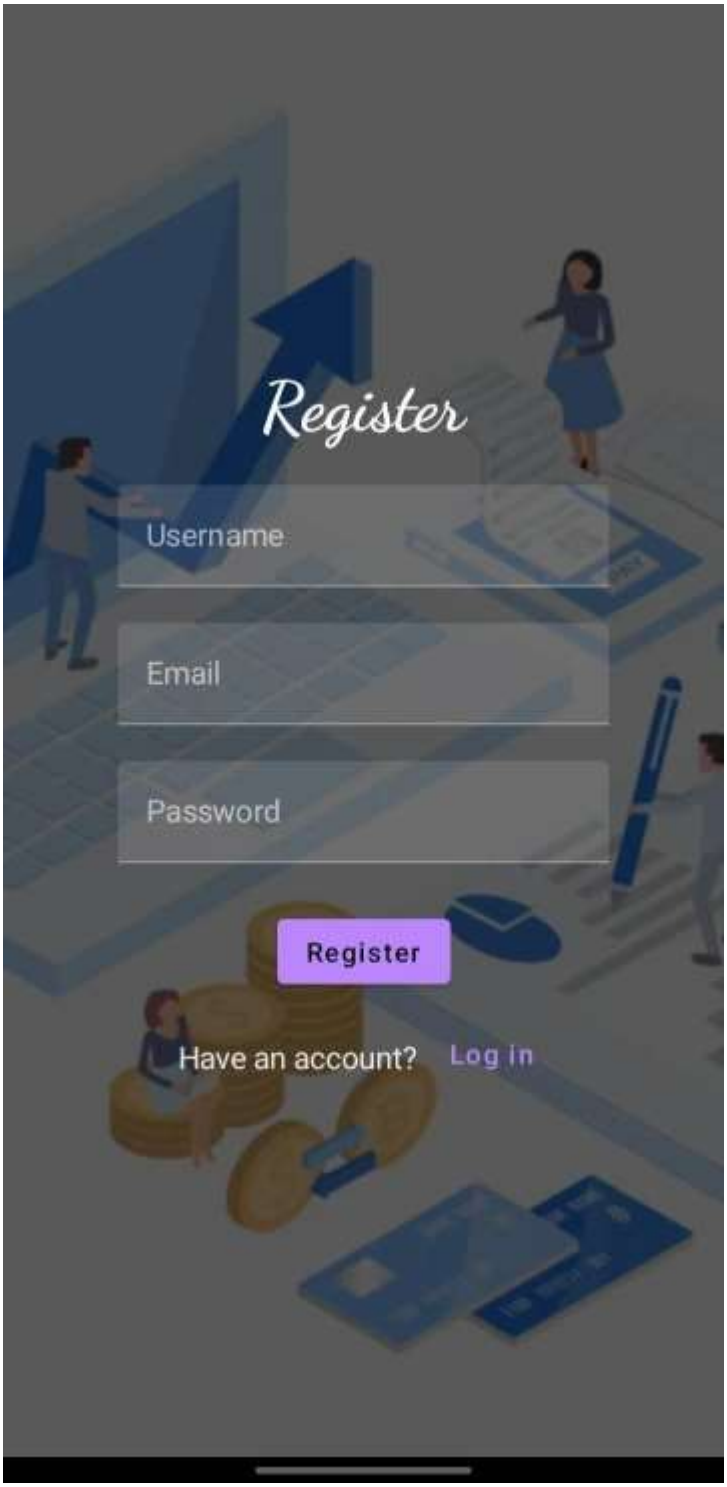
Final Output of the Application :

Login Page :


Login
Username
Password
Login
Sign up
Forget password?



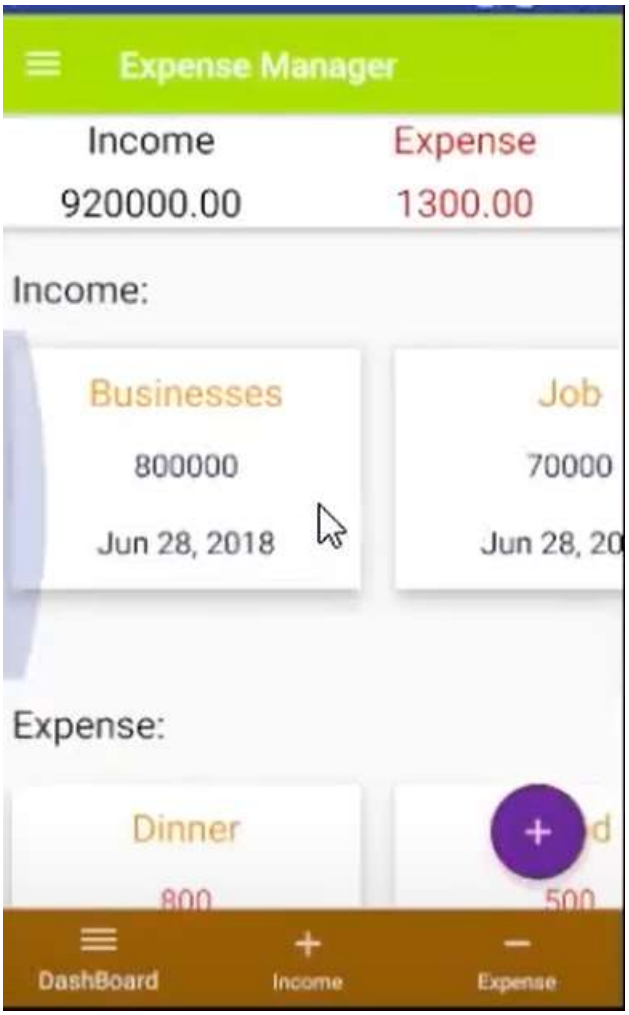
RegisterPage :


Register
Username
Email
Password
Register
Have an account?
Log in

MainPage :

Welcome To Expense Tracker

Add Expenses | Set Limit | View Records



Add Expenses page:

Item Name

Item Name
pizza

Quantity of item

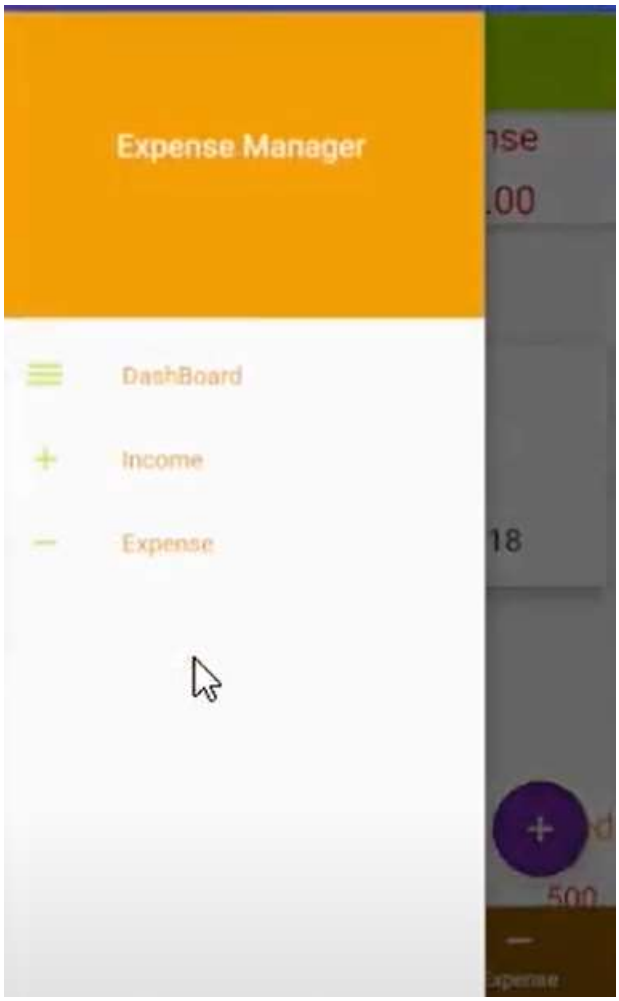
Quantity
2

Cost of the item

Cost
400

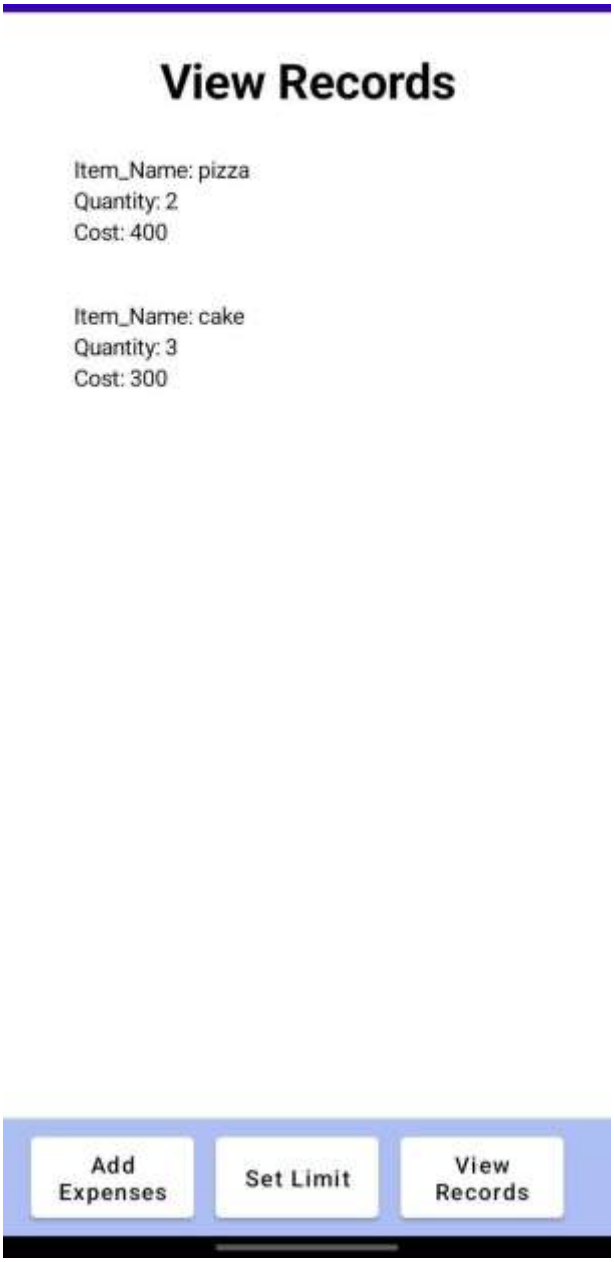
Submit

Add Expenses | Set Limit | View Records

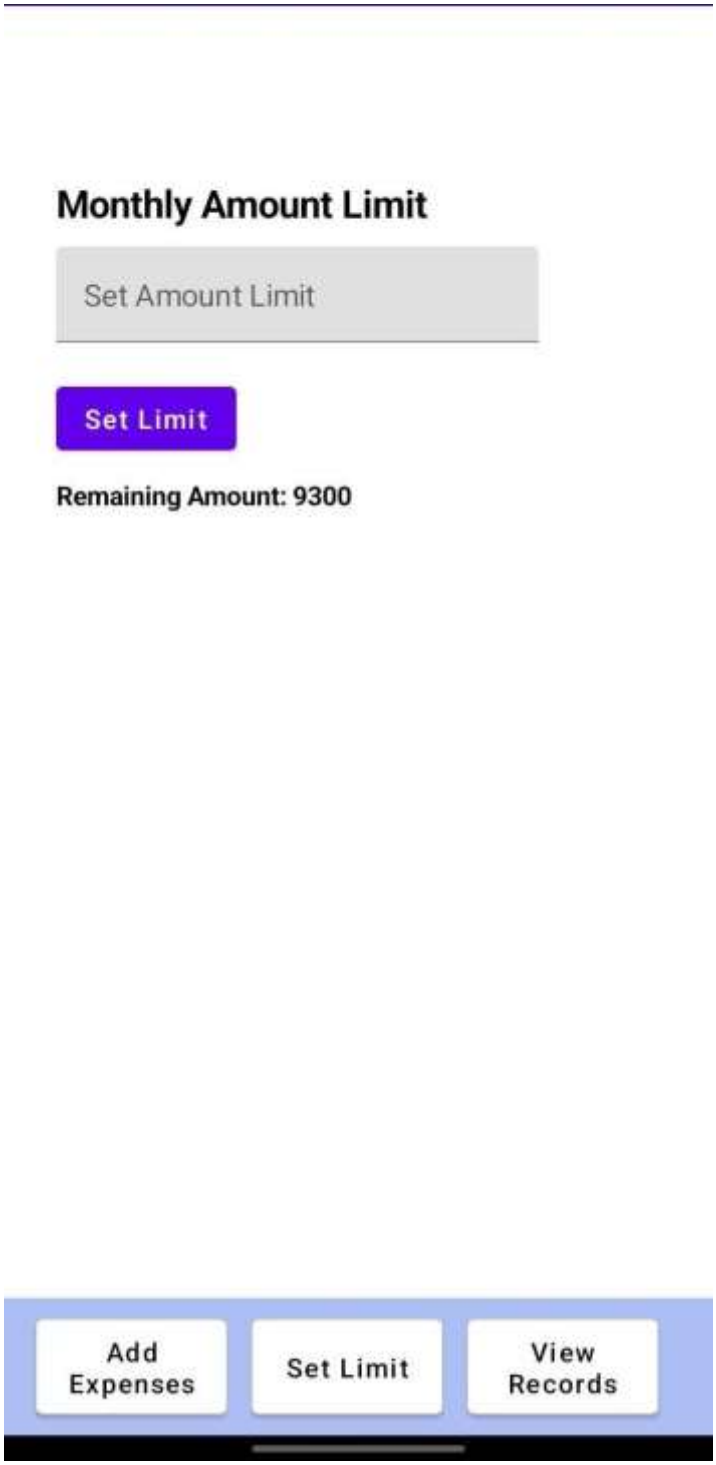


Set Limit Page before adding any data in expenses:

View Records Page:



Set Limit Page After adding expenes in add expense page:


Monthly Amount Limit
Set Amount Limit
Set Limit
Remaining Amount: 9300
Add Expenses
Set Limit
View Records