



**Smart
Internz**

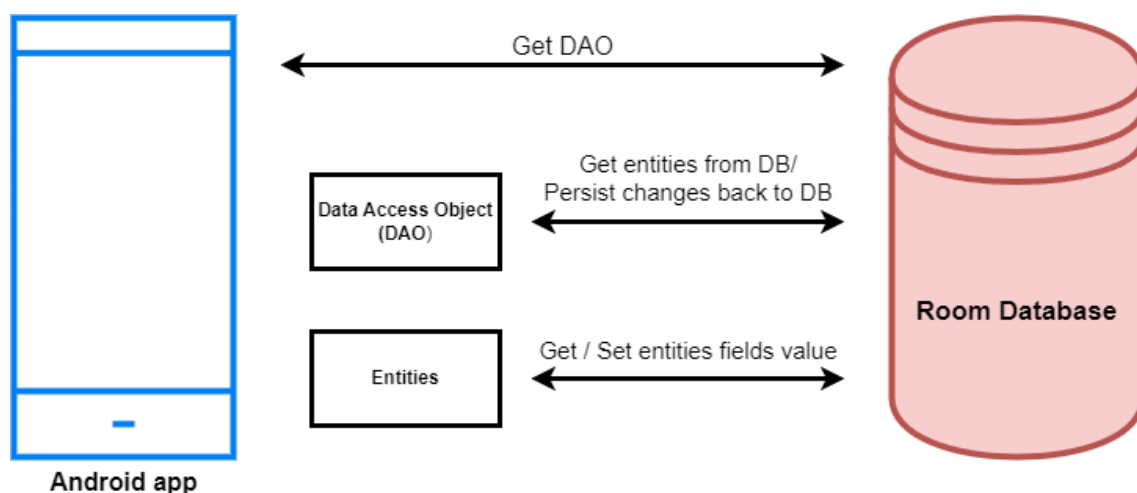
Snack Squad: A Customizable Snack Ordering and Delivery App

Project Based Experiential Learning Program

Snack Squad: A Customizable Snack Ordering and Delivery App

A project that demonstrates the use of Android Jetpack Compose to build a UI for a snack squad app. Snack Squad is a sample project built using the Android Compose UI toolkit. It demonstrates how to create a simple e-commerce app for snacks using the Compose libraries. The user can see a list of snacks, and by tapping on a snack, and by tapping on the "Add to Cart" button, the snack will be added to the cart. The user can also see the list of items in the cart and can proceed to checkout to make the purchase.

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
- User can view the items,select and order the items
- From admin login we can view the orders placed.

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 :

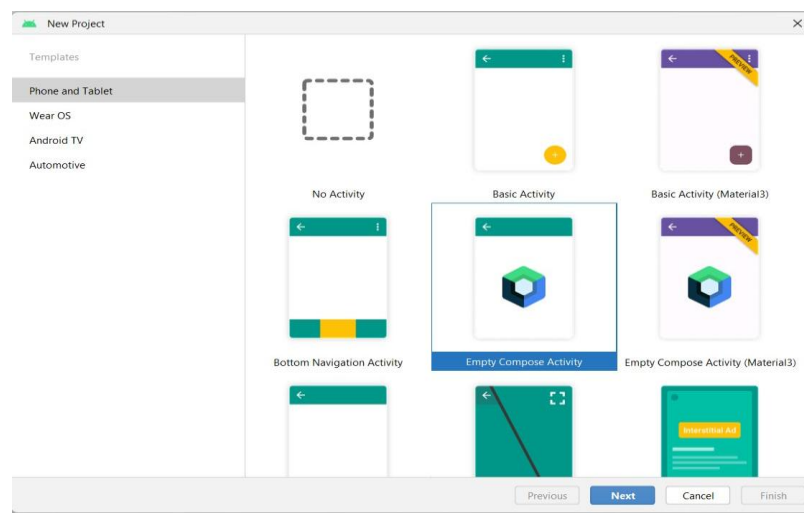
<https://developer.android.com/studio/install>

Task 2 :

Creating a new project.

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

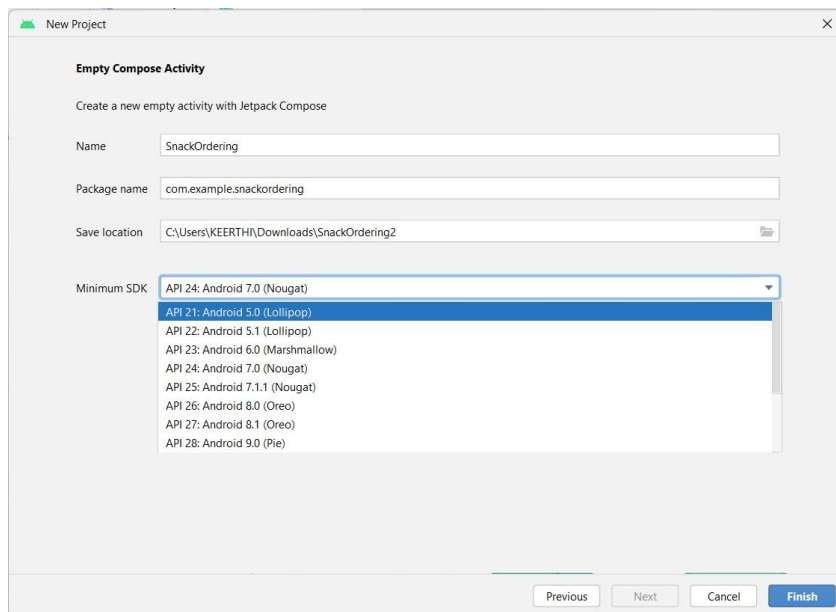
Step 2 : Click on **Next** button.



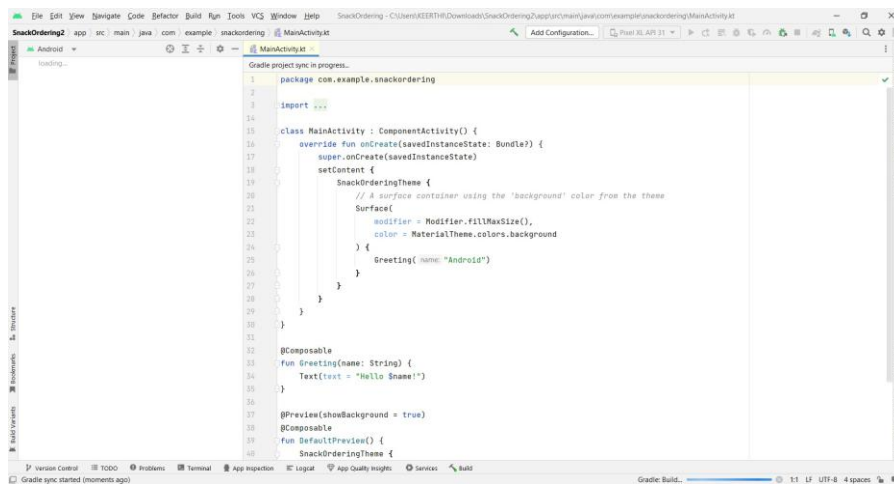
Step 3 : Give name to the new project.

Step 4 : Give the Minimum SDK value

Step 5 : Click Finish



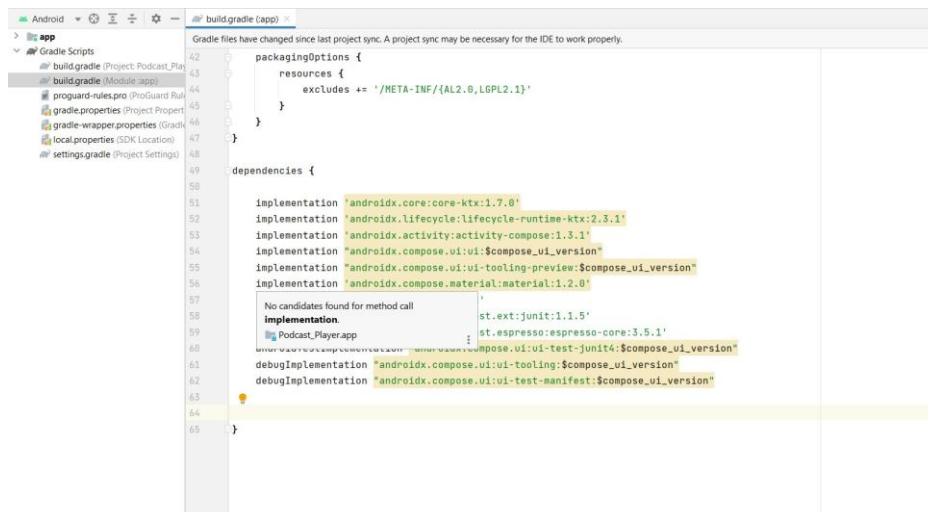
Main activity file



Task 3 :

Adding required dependencies.

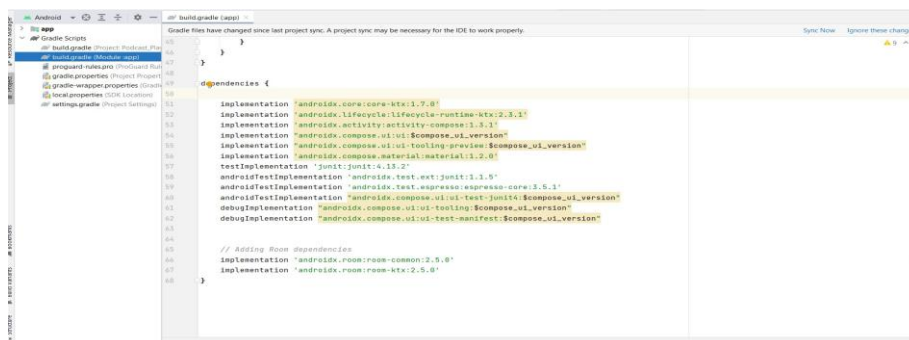
Step 1 : Gradle scripts > build.gradle(Module :app)



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

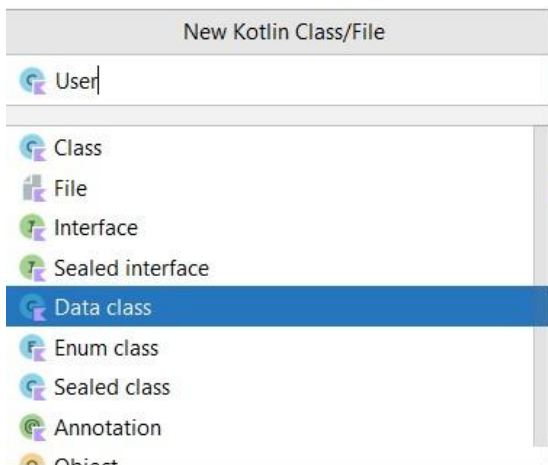
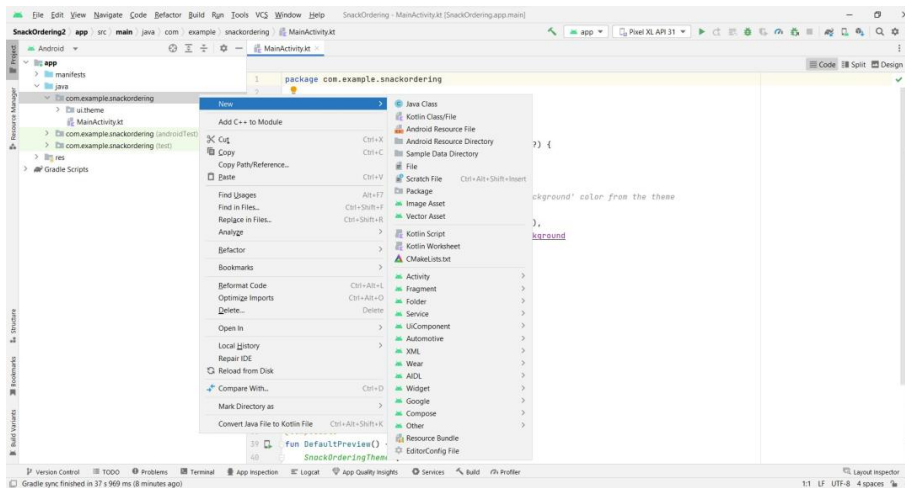
Task 4:

Creating the database classes.

In this project we will be having two databases, one is for user registration and login, and other is for tracking the orders of the user, which is used for admin page .

Database 1

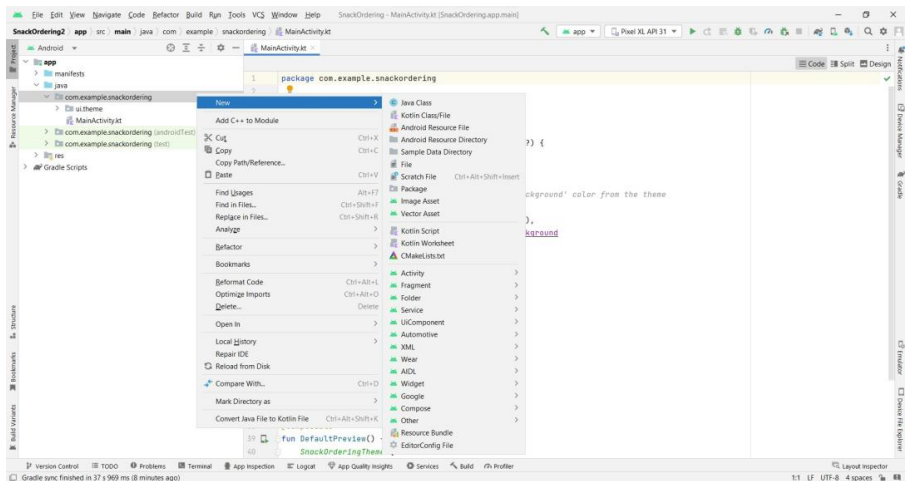
Step 1 : Create User data class

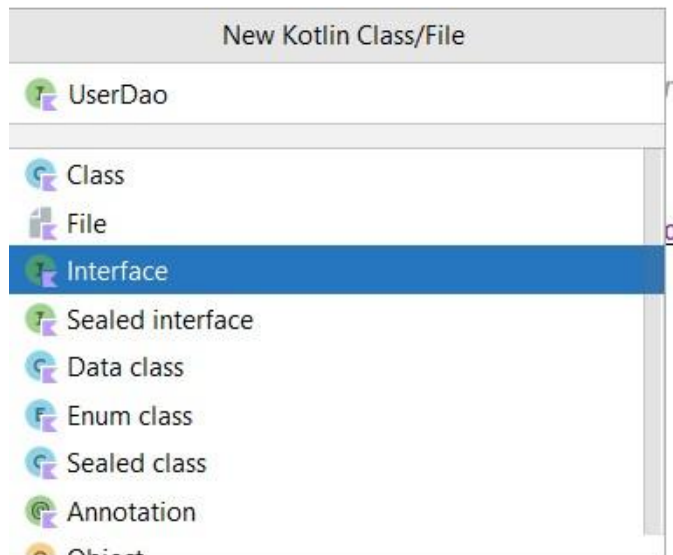


User class code:

<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/example/snackordering/User.kt>

Step 2 : Create an UserDao interface

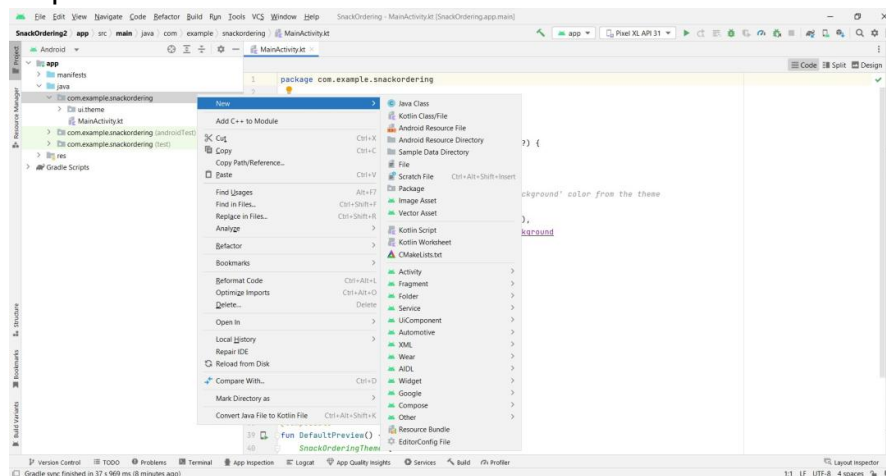


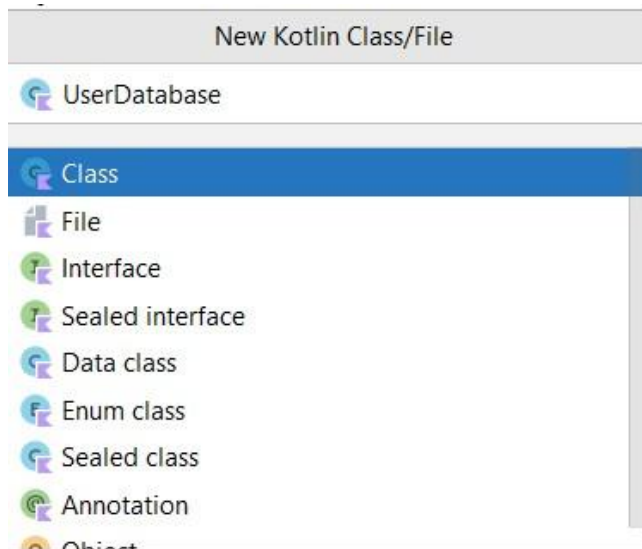


UserDao interface code:

<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/example/snackordering/UserDao.kt>

Step 3 : Create an UserDatabase class

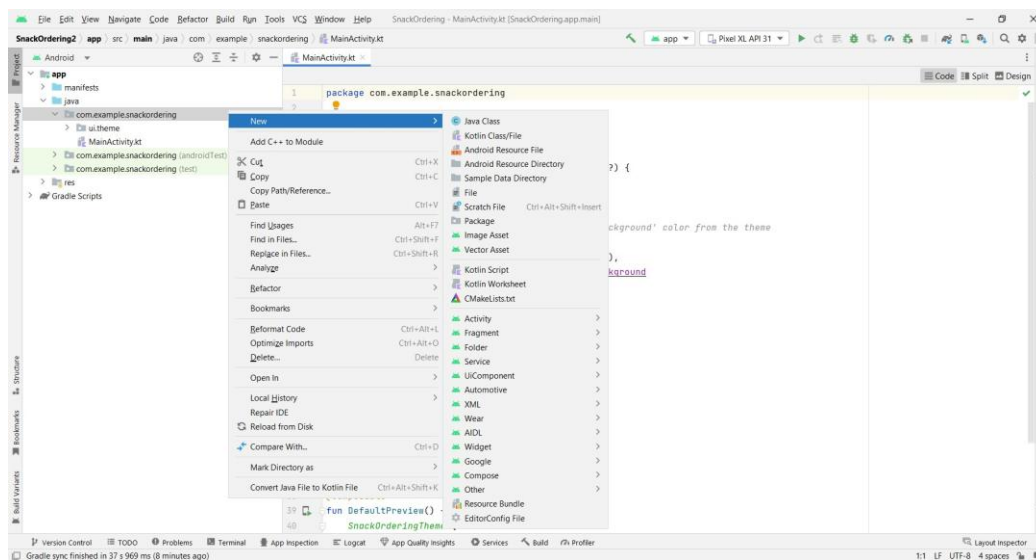


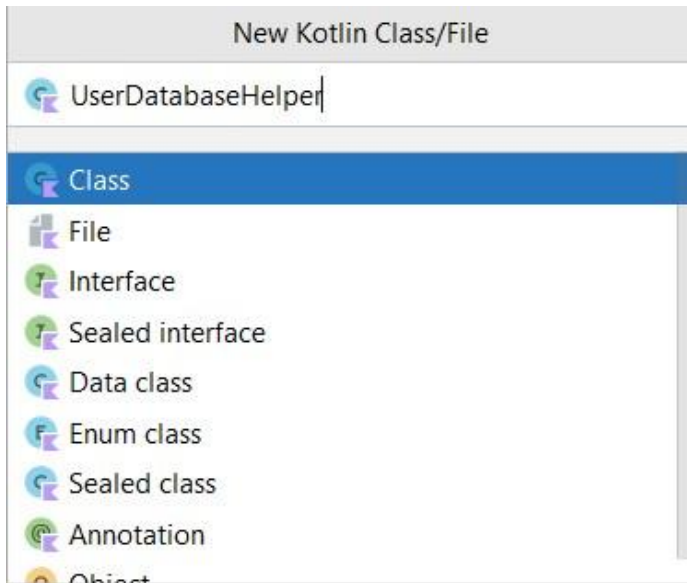


UserDatabase class code :

<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/example/snackordering/UserDatabase.kt>

Step 4 : Create an UserDatabaseHelper class



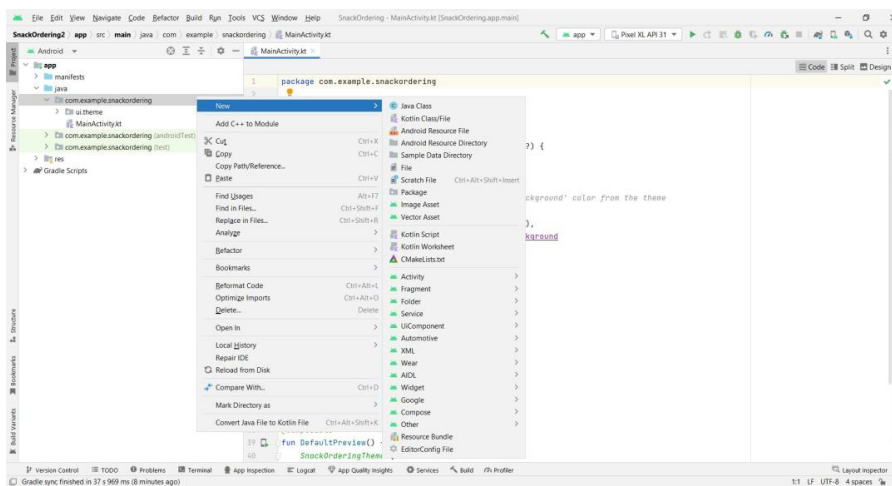


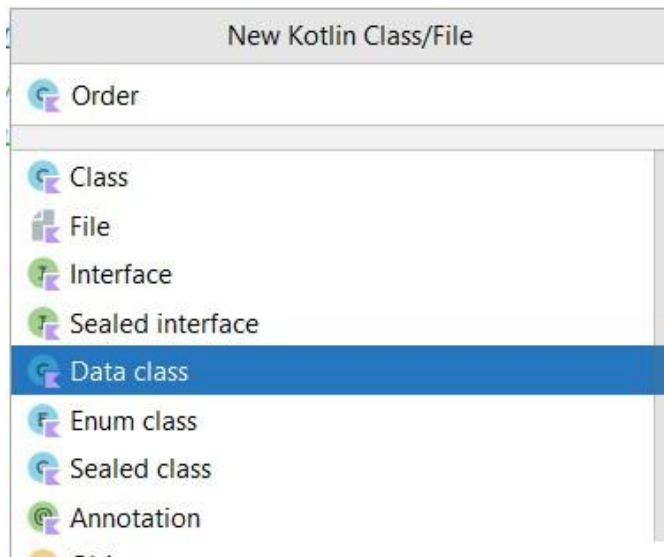
UserDatabaseHelper class code :

<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/example/snackordering/UserDatabaseHelper.kt>

Database 2

Step 1 : Create Order data class

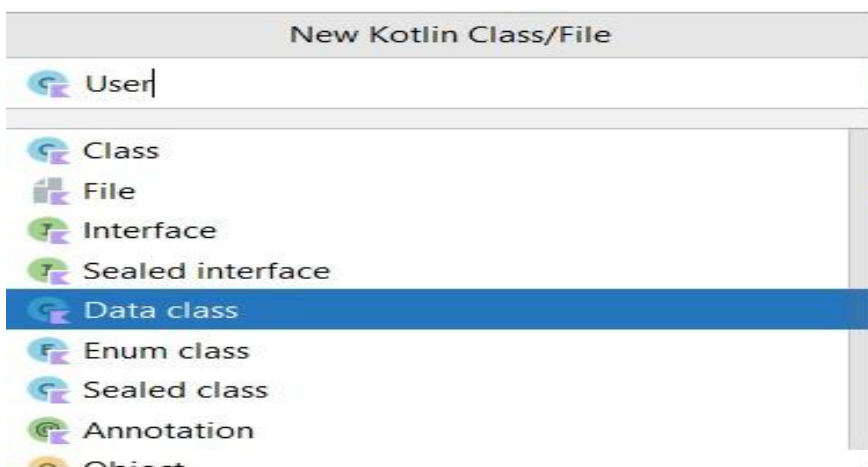


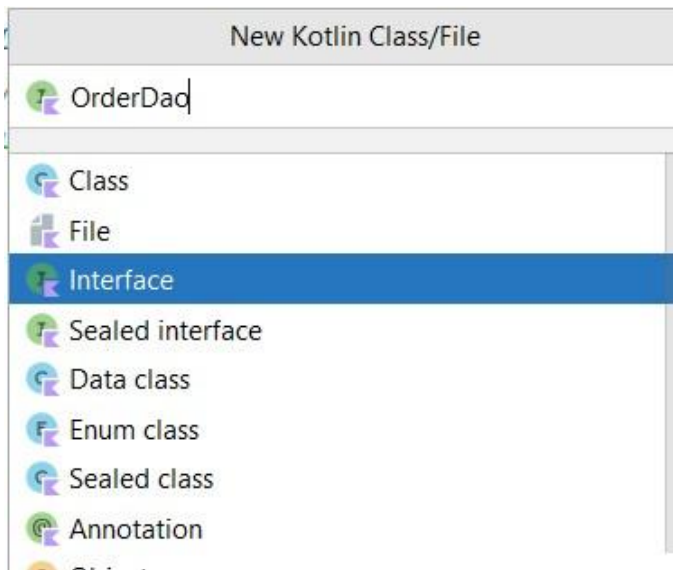


Order data class code:

<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/example/snackordering/Order.kt>

Step 2 : Create OrderDao interface

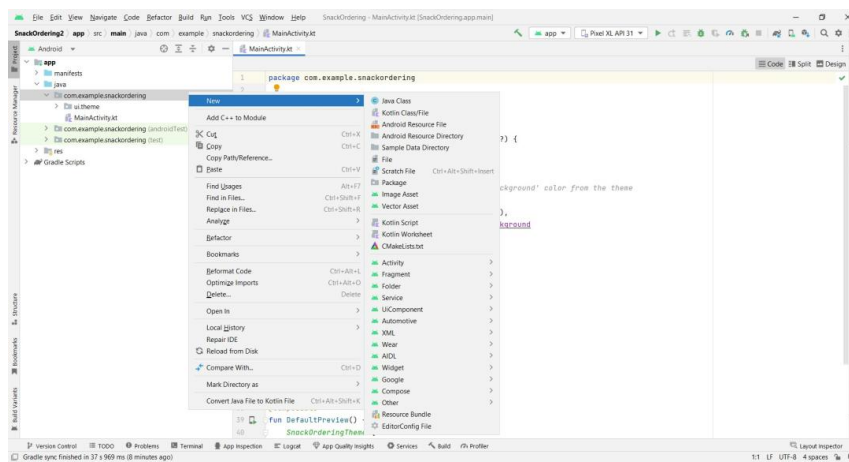


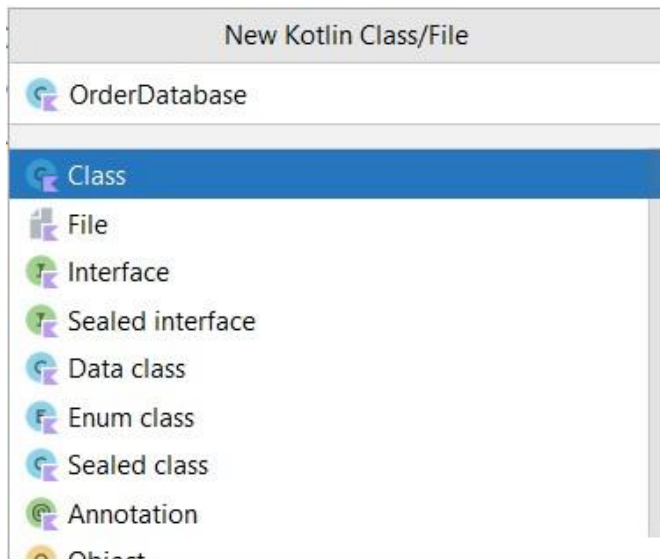


OrderDao interface code:

<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/example/snackordering/OrderDao.kt>

Step 3 : Create OrderDatabase class

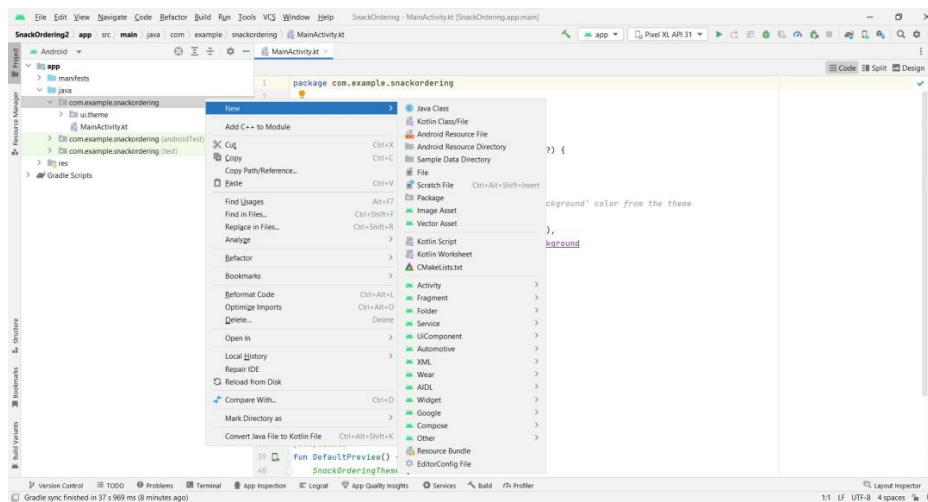


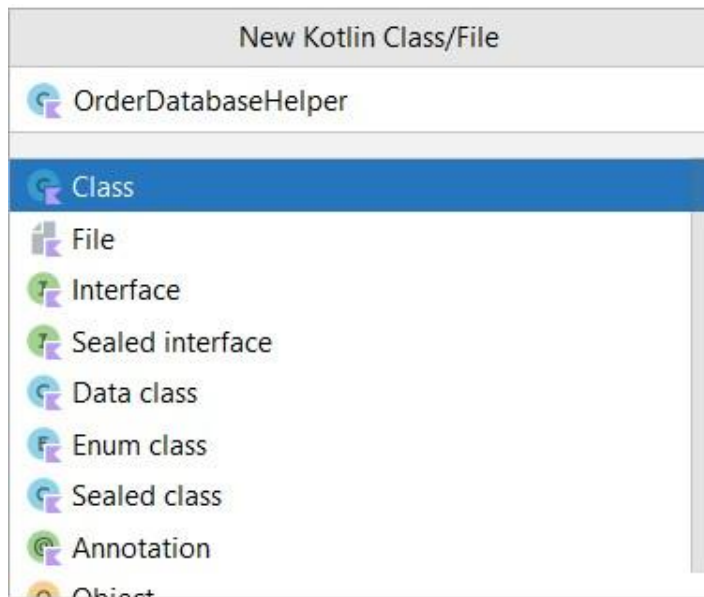


OrderDatabase class code:

<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/example/snackordering/OrderDatabase.kt>

Step 4 : Create OrderDatabaseHelper class





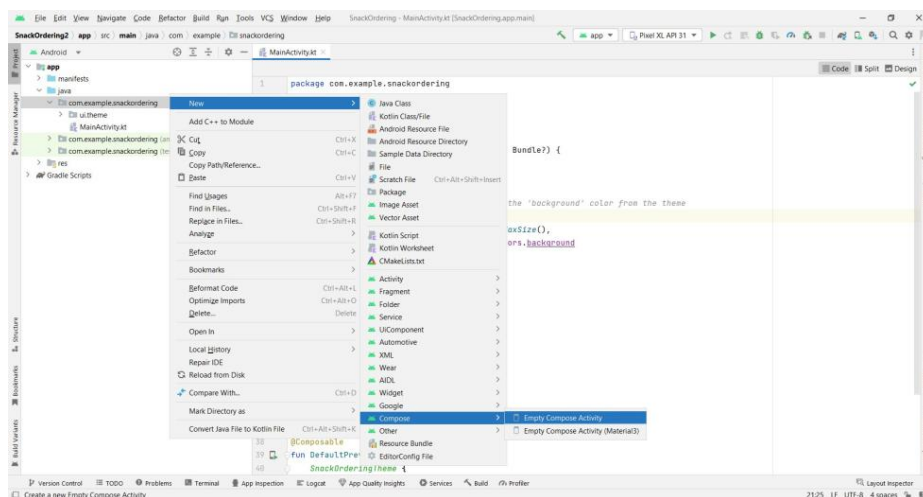
OrderDatabaseHelper class code:

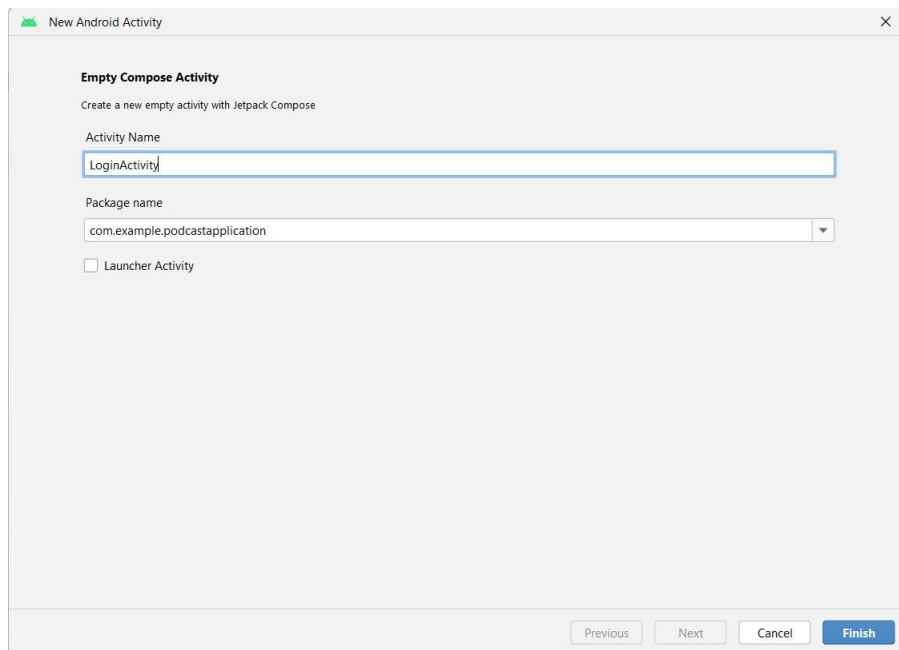
<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/example/snackordering/OrderDatabaseHelper.kt>

Task 5:

Building application UI and connecting to database.

Step 1: Creating LoginActivity.kt with database





Database connection in LoginActivity.kt

```
package com.example.snackordering

import ...

class LoginActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(context: this)
        setContent {
            SnackOrderingTheme {
                // 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.order), contentDescription = "",
        alpha = 0.3F,
        contentScale = ContentScale.FillHeight,
    )

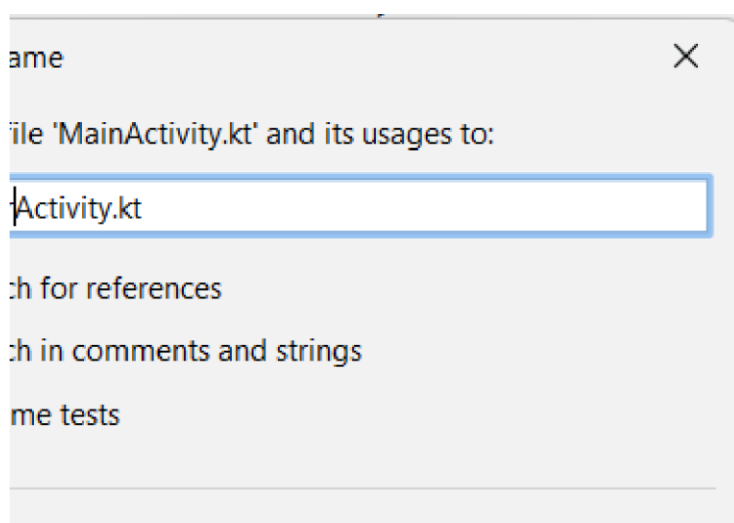
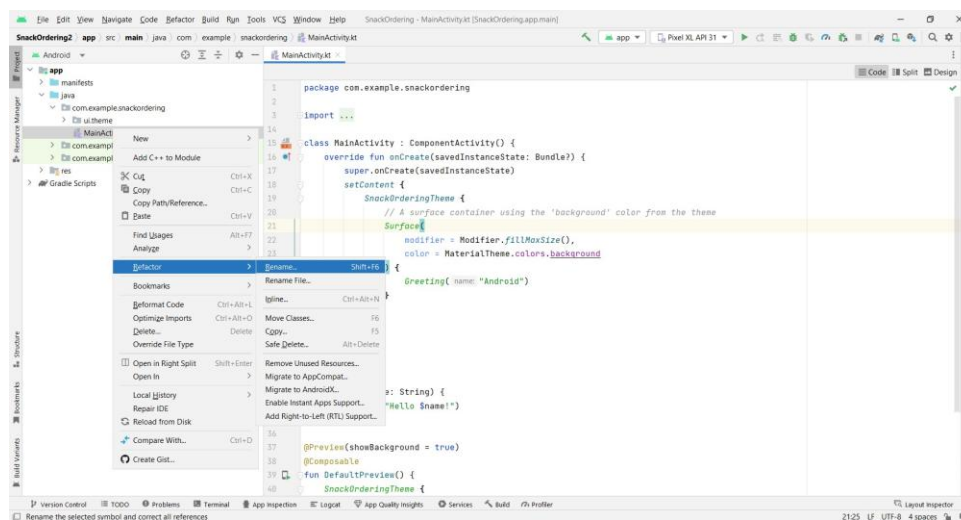
    var username by remember { mutableStateOf( value: "" ) }
    var password by remember { mutableStateOf( value: "" ) }
    var error by remember { mutableStateOf( value: "" ) }
```

Complete code in below link:

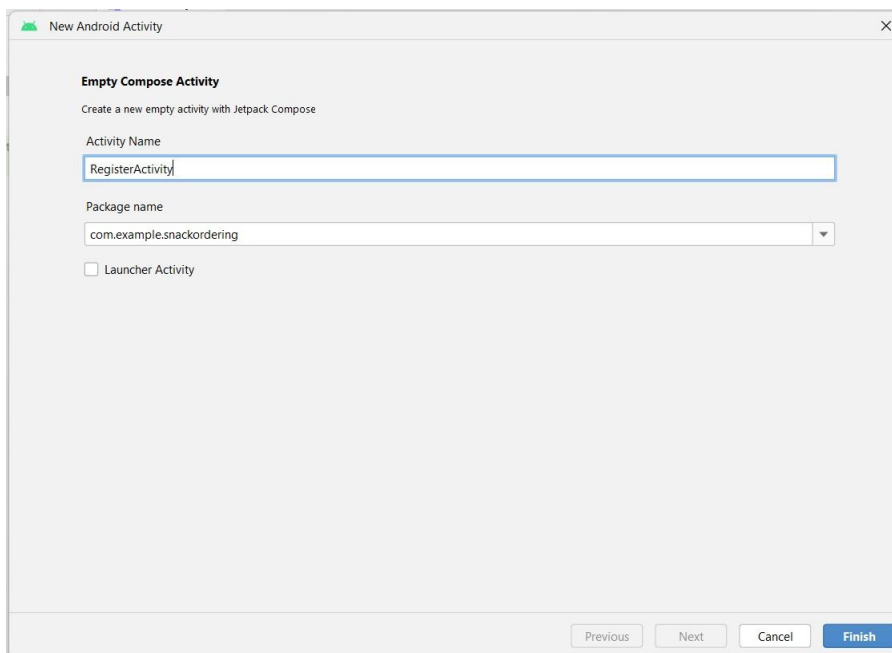
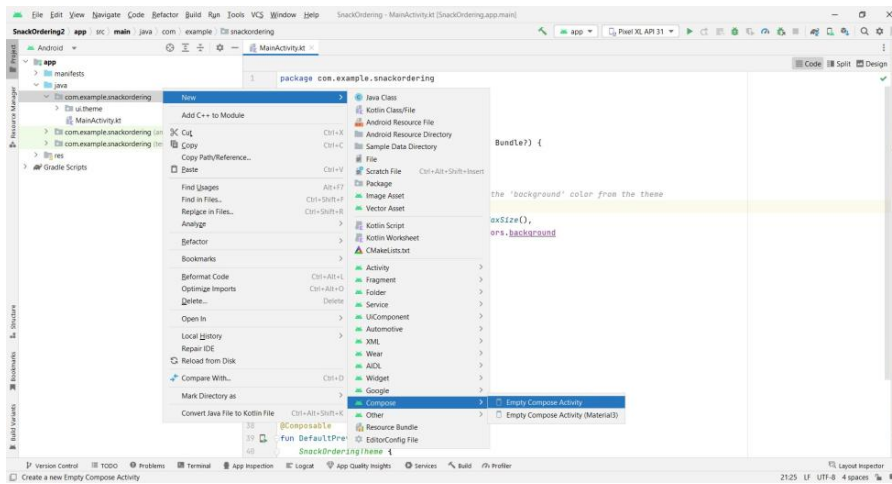
<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/example/snackordering/LoginActivity.kt>

Step 2 : Creating MainActivity.kt with database

MainActivity is converted into RegisterActivity.kt as follows below:



Now click on the Refactor.



Database connection in RegisterActivity.kt

```

package com.example.snackordering

import ...

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

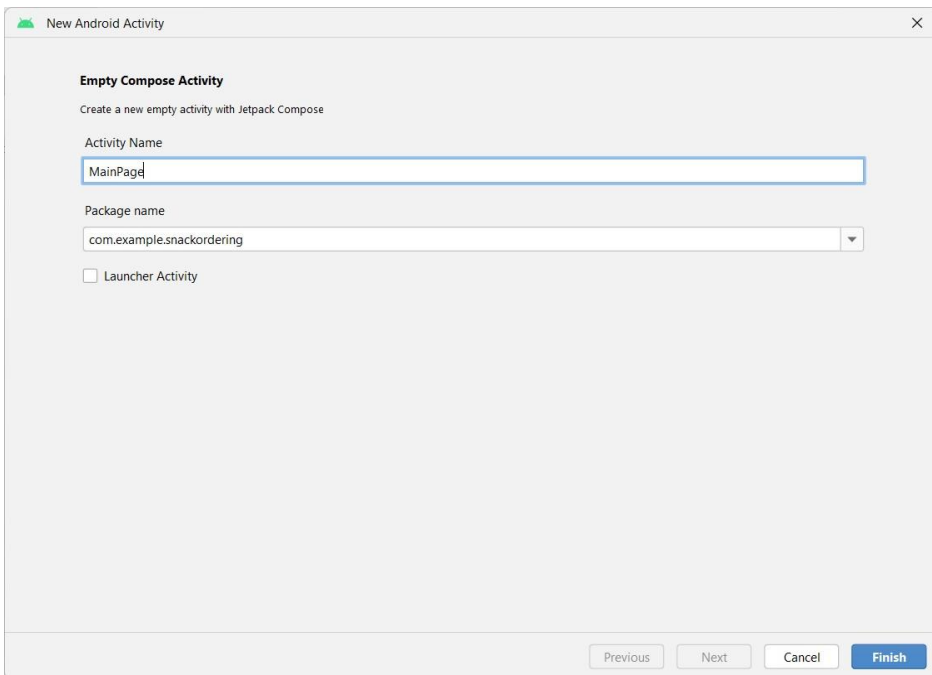
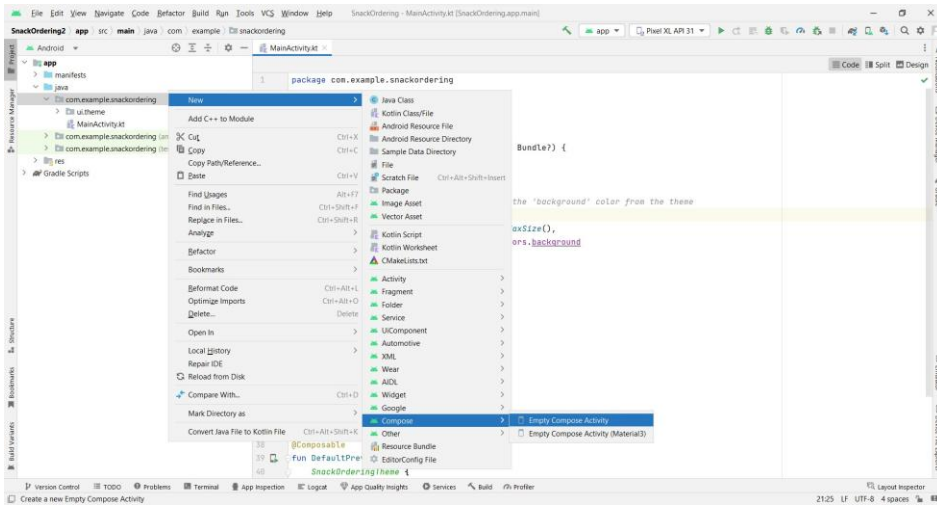
@Composable
fun RegistrationScreen(context: Context, databaseHelper: UserDatabaseHelper) {
    Image(
        painterResource(id = R.drawable.order), contentDescription = "",
        alpha = 0.3F,
        contentScale = ContentScale.FillHeight,
    )
    var username by remember { mutableStateOf(value: "") }
    var password by remember { mutableStateOf(value: "") }
    var email by remember { mutableStateOf(value: "") }
    var error by remember { mutableStateOf(value: "") }
}

```

Complete code in below link:

<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/example/snackordering/RegisterActivity.kt>

Step 3 : Creating MainPage.kt file



```

package com.example.snackordering

import ...

class MainPage : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            SnackOrderingTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    FinalView( mainPage: this)
                    val context = LocalContext.current
                    //PopularFoodColumn(context)
                }
            }
        }
    }
}

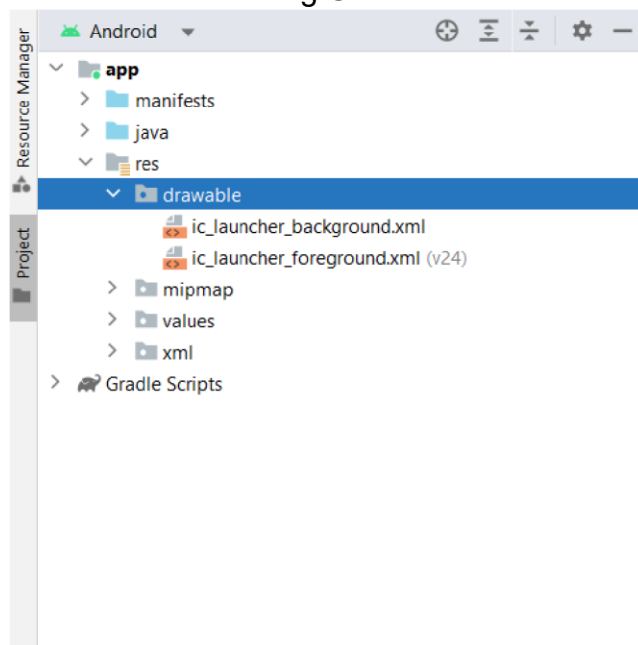
@Composable
fun TopPart() {

    Row(
        modifier = Modifier
            .fillMaxWidth()
            .background(Color( color: 0xffeceef0)), Arrangement.SpaceBetween
    ) { this: RowScope
        TopPart
    }
}

```

In MainPage.kt file the main application is developed

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



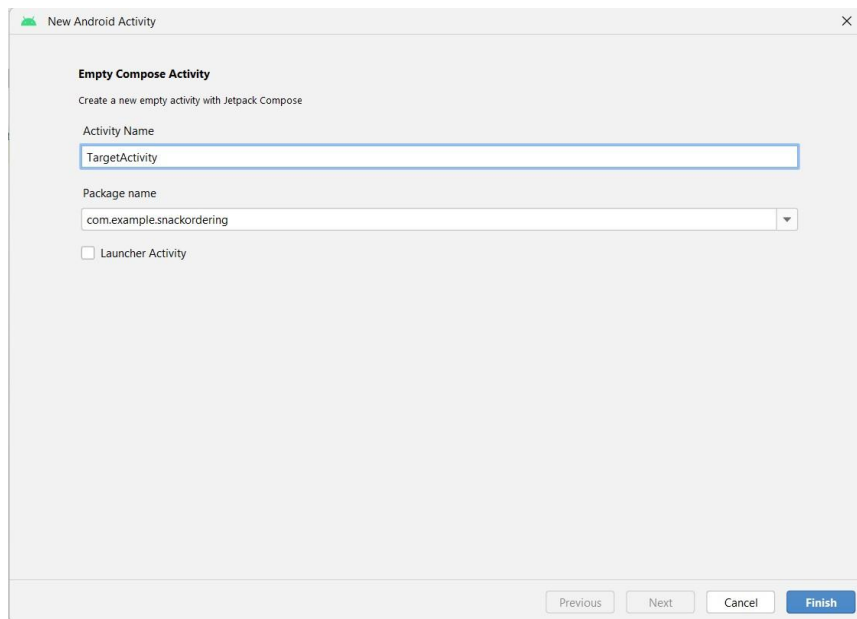
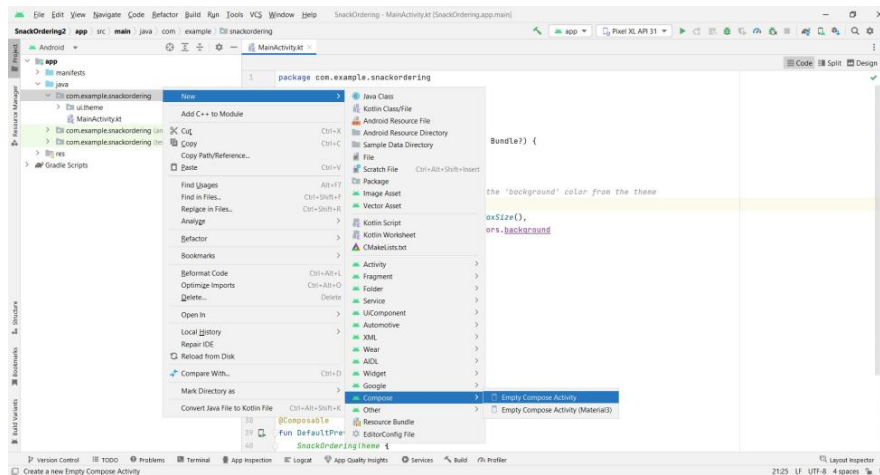
Download the required drawable from the code:

<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/res/drawable-nodpi>

Complete code of MainPage.kt :

<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/example/snackordering/MainPage.kt>

Step 4 : Creating TargetActivity.kt



```

package com.example.snackordering

import ...

class TargetActivity : AppCompatActivity() {
    private lateinit var orderDatabaseHelper: OrderDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        orderDatabaseHelper = OrderDatabaseHelper(context = this)
        setContentView {
            SnackOrderingTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize().background(Color.White)) {
                    Order(context = this, orderDatabaseHelper)
                    val orders = orderDatabaseHelper.getAllOrders()
                    Log.d(tag = "swathi", orders.toString())
                }
            }
        }
    }
}

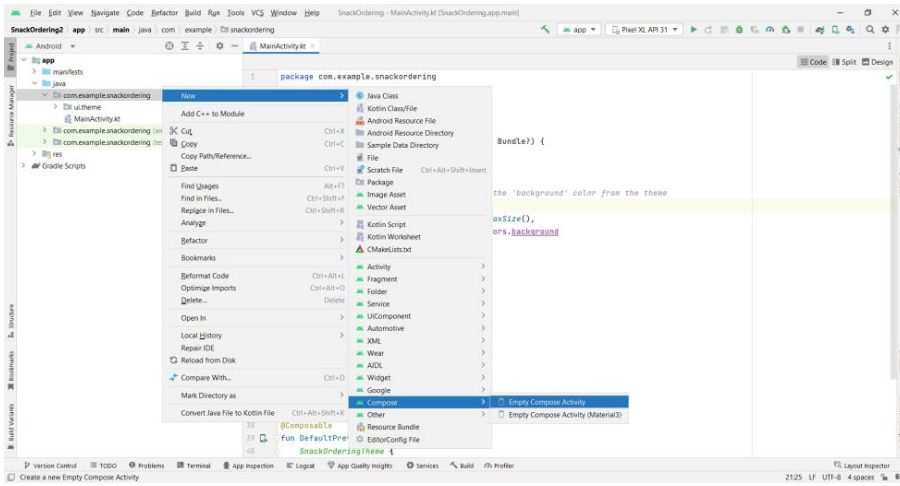
@Composable
fun Order(context: Context, orderDatabaseHelper: OrderDatabaseHelper){
    Image(painterResource(id = R.drawable.order), contentDescription = "",
        alpha = 0.5F, contentScale = ContentScale.FillHeight)
    Column(
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center)
    { this: ColumnScope
        val mContext = LocalContext.current
        var quantity by remember { mutableStateOf( value: "") }
        var address by remember { mutableStateOf( value: "") }
        var error by remember { mutableStateOf( value: "") }
    }
}

```

Complete code of TargetActivity.kt :

<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/e>

[xample/snackordering/TargetActivity.kt](#) Step 4 : Creating AdminActivity.kt



New Android Activity

Empty Compose Activity

Create a new activity with Jetpack Compose

Activity Name

AdminActivity

Package name

com.example.snackordering

☐ Launcher Activity

Previous

Next

Cancel

Finish


```

package com.example.snackordering

import ...

class AdminActivity : ComponentActivity() {
    private lateinit var orderDatabaseHelper: OrderDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        orderDatabaseHelper = OrderDatabaseHelper(context = this)
        setContent {
            SnackOrderingTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    val data = orderDatabaseHelper.getAllOrders();
                    Log.d(tag = "swathi", data.toString())
                    val order = orderDatabaseHelper.getAllOrders()
                    ListListScopeSample(order)
                }
            }
        }
    }
}

@Composable
fun ListListScopeSample(order: List<Order>) {
    Image(
        painterResource(id = R.drawable.order), contentDescription = "",
        alpha = 0.5f,
        contentScale = ContentScale.FillHeight)
}

```

Complete code of AdminActivity.kt :

<https://github.com/smartinternz02/SnackOrdering/tree/main/app/src/main/java/com/example/snackordering/AdminActivity.kt>

Task 6:

Modifying AndroidManifest.xml

```

        android:name=".AdminActivity"
        android:exported="false"
        android:label="AdminActivity"
        android:theme="@style/Theme.SnackOrdering" />
    <activity
        android:name=".MainPage"
        android:exported="true"
        android:label="SnackSquad"
        android:theme="@style/Theme.SnackOrdering">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
    <activity
        android:name=".TargetActivity"
        android:exported="false"
        android:label="TargetActivity"
        android:theme="@style/Theme.SnackOrdering" />
    <activity
        android:name=".LoginActivity"
        android:exported="false"
        android:label="LoginActivity"
        android:theme="@style/Theme.SnackOrdering" />
    <activity
        android:name=".MainActivity"
        android:exported="false"
        android:label="MainActivity"
        android:theme="@style/Theme.SnackOrdering" />
</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.

```

    android:exported="false"
    android:label="AdminActivity"
    android:theme="@style/Theme.SnackOrdering" />
<activity
    android:name=".LoginActivity"
    android:exported="true"
    android:label="SnackSquad"
    android:theme="@style/Theme.SnackOrdering">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<activity
    android:name=".TargetActivity"
    android:exported="false"
    android:label="TargetActivity"
    android:theme="@style/Theme.SnackOrdering" />
<activity
    android:name=".MainPage"
    android:exported="false"
    android:label="MainPage"
    android:theme="@style/Theme.SnackOrdering" />
<activity
    android:name=".MainActivity"
    android:exported="false"
    android:label="MainActivity"
    android:theme="@style/Theme.SnackOrdering" />
</application>
</manifest>

```

Complete AndroidManifest.xml code:

<https://github.com/smartinternz02/SnackOrdering/blob/main/app/src/main/AndroidManifest.xml>

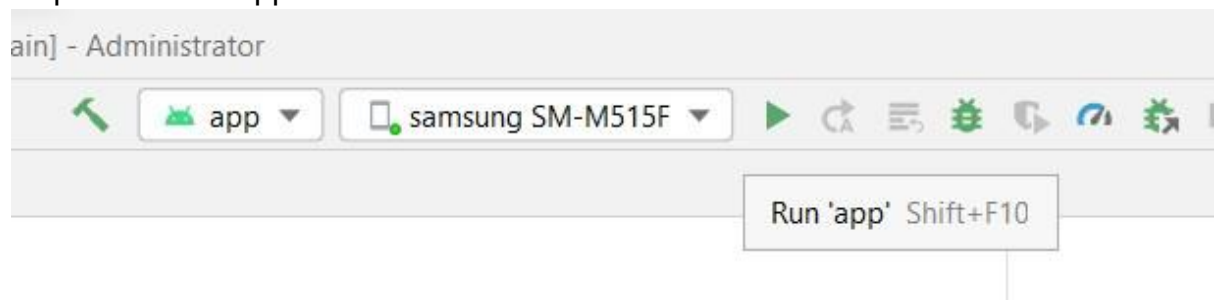
Task 7:

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



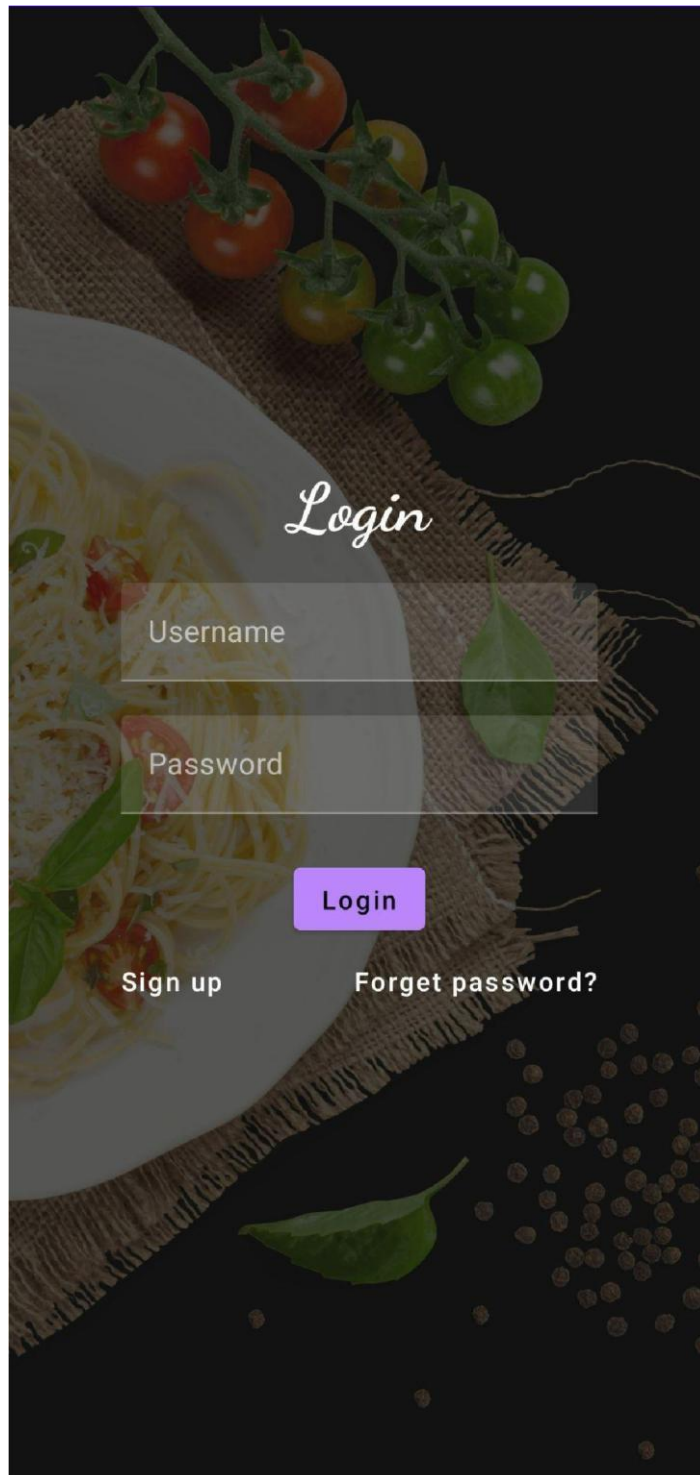
Complete Project Link:

<https://github.com/Mr-Dabhade/Android-Snack-Squad>

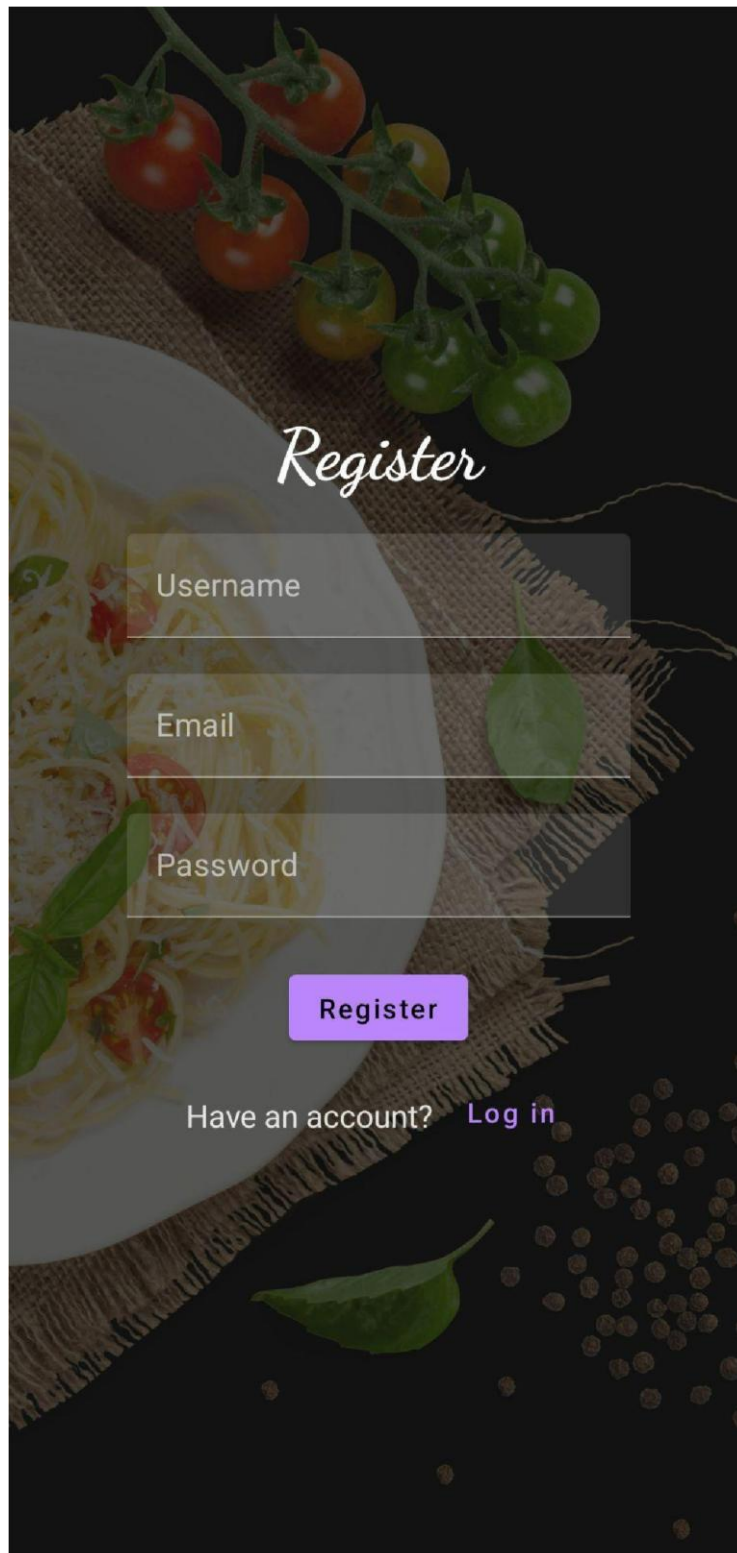
Final Output of the Application :

Admin Module: After logging in with Admin Credentials which are hard coded.

Login Page :

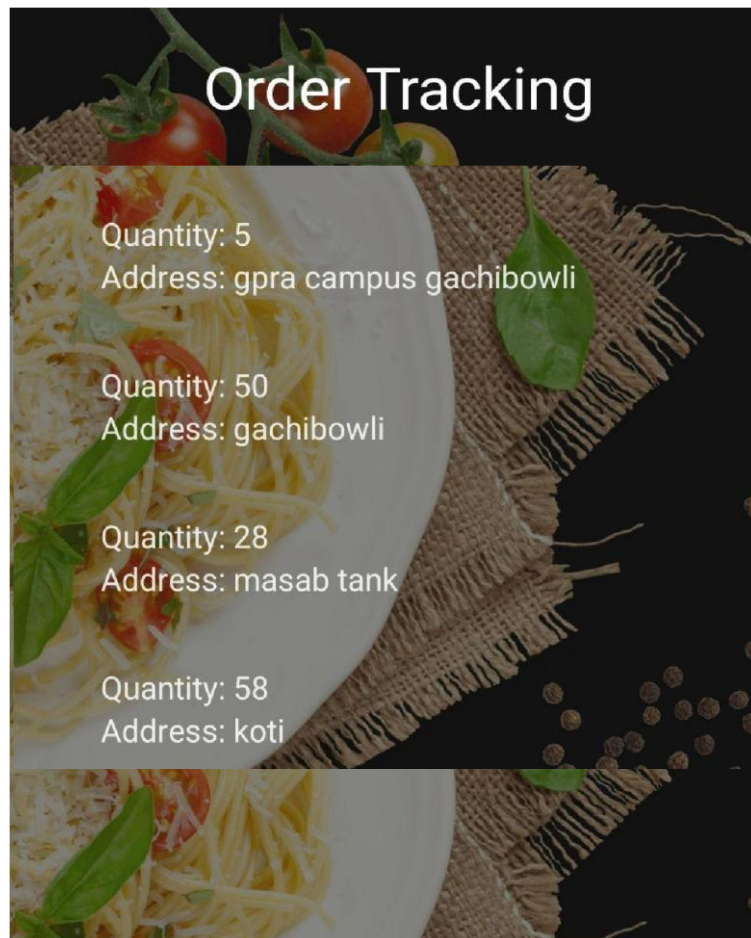


Register Page :

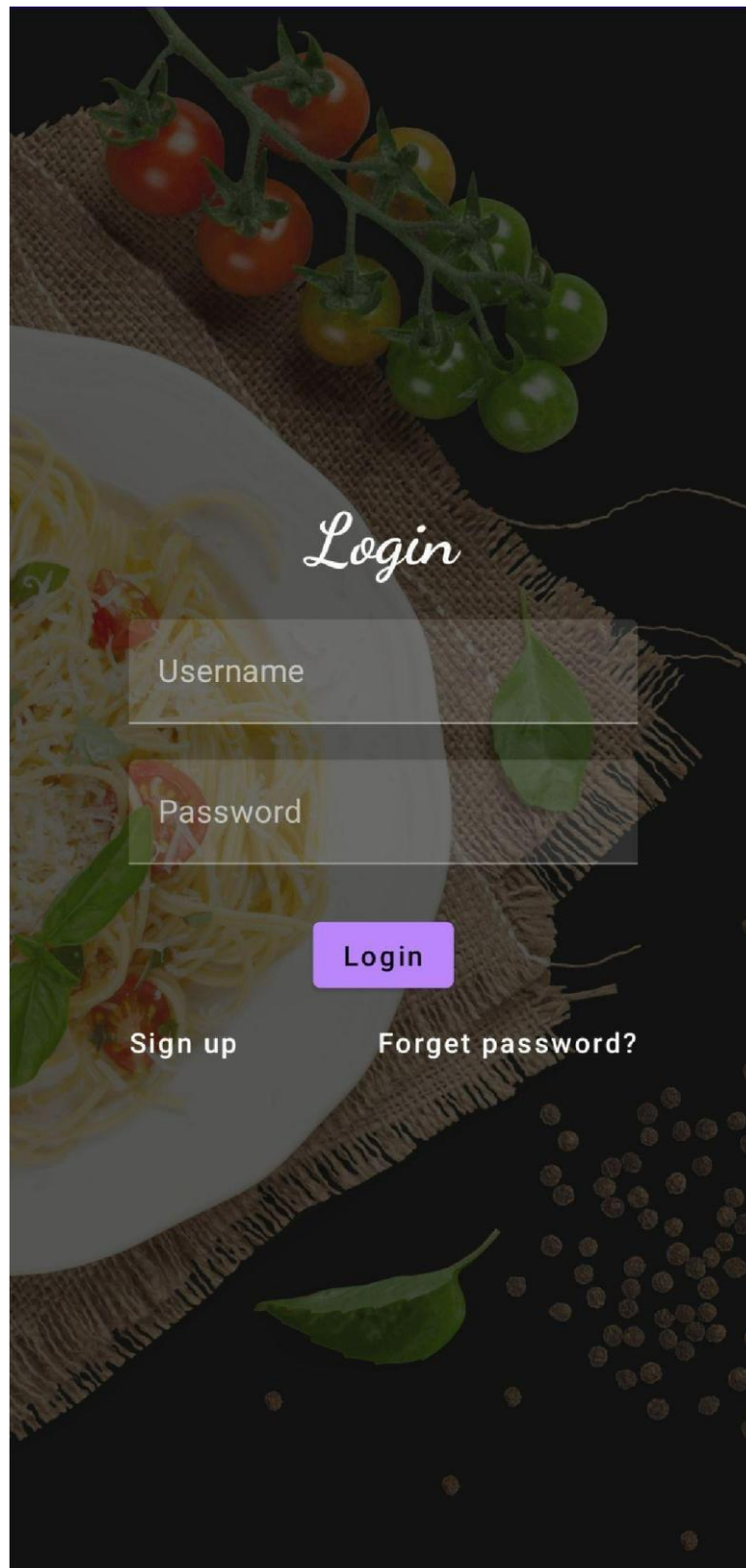


After logging in with Admin Credentials which are hard coded. Password must be "admin" .

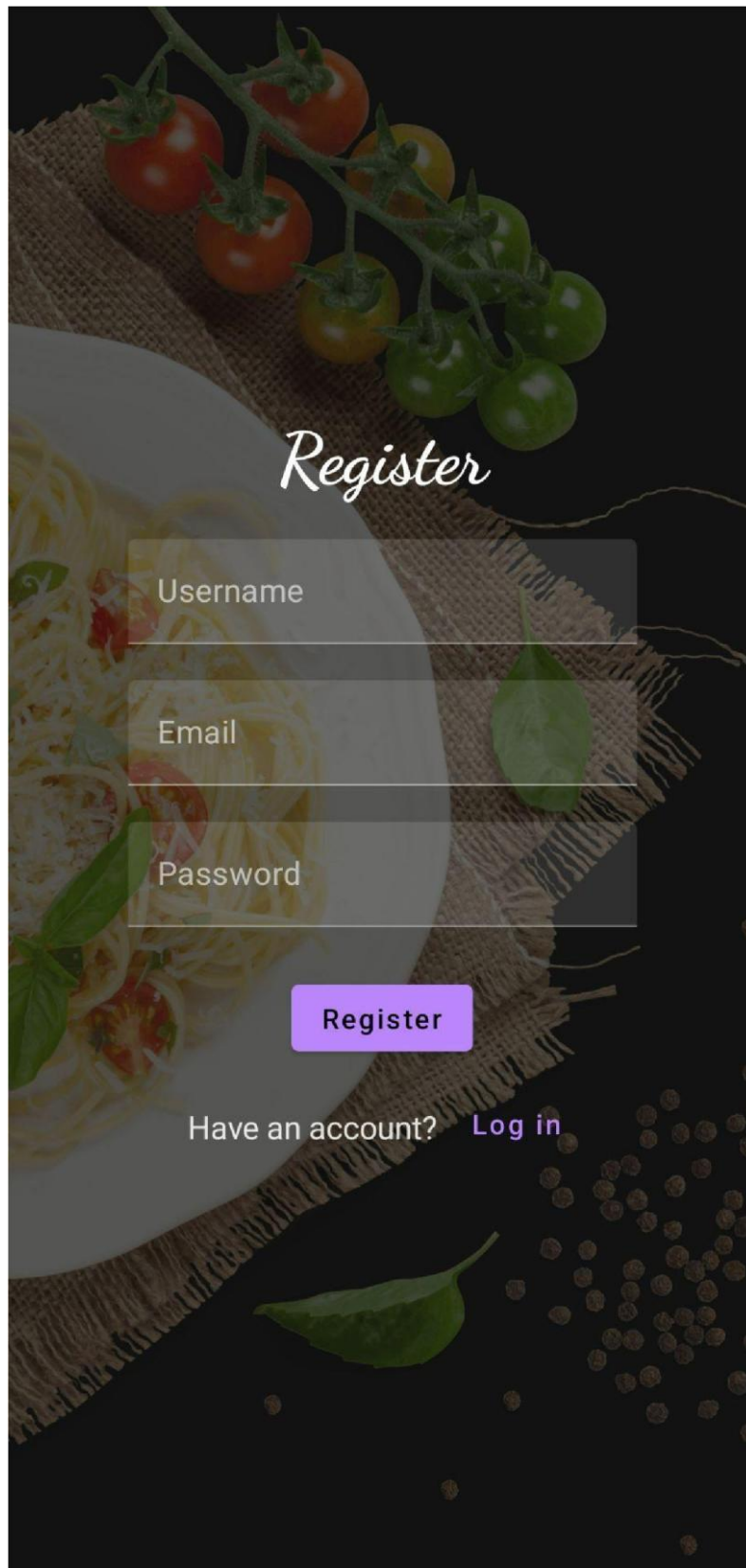
Admin page:



User Module:
Login Page :



Register Page :



Main Page :



Location
Accra



Get Special Discounts

up to 85%

Claim voucher



Popular Food

[view all](#)

★ 4.3



Sandwich

\$50



★ 4.3

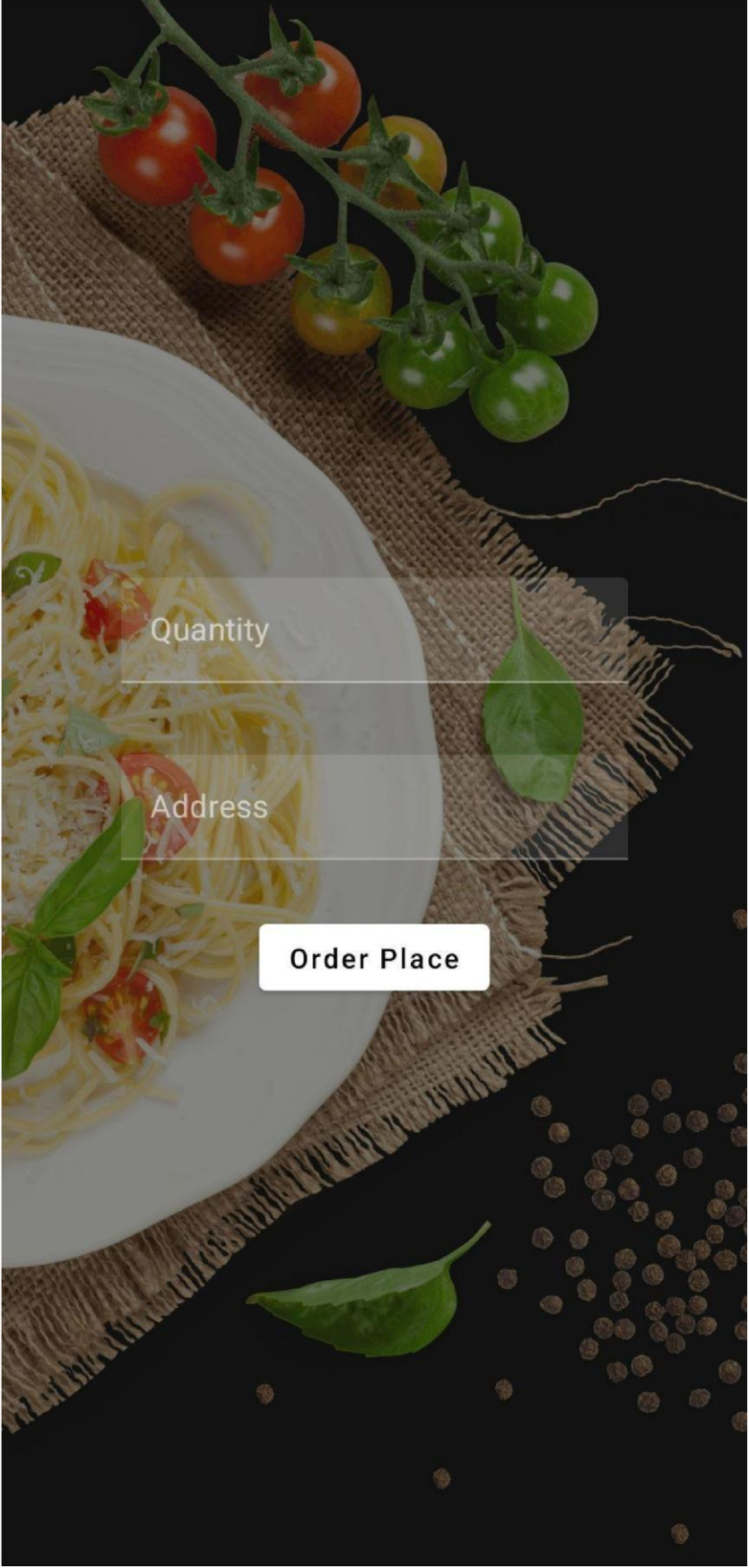


Burger

\$50



Order Page :



Quantity

Address

Order Place