Compose Input: A Demonstration of Text Input and Validation with Android Compose

Project Based Experiential Learning Program

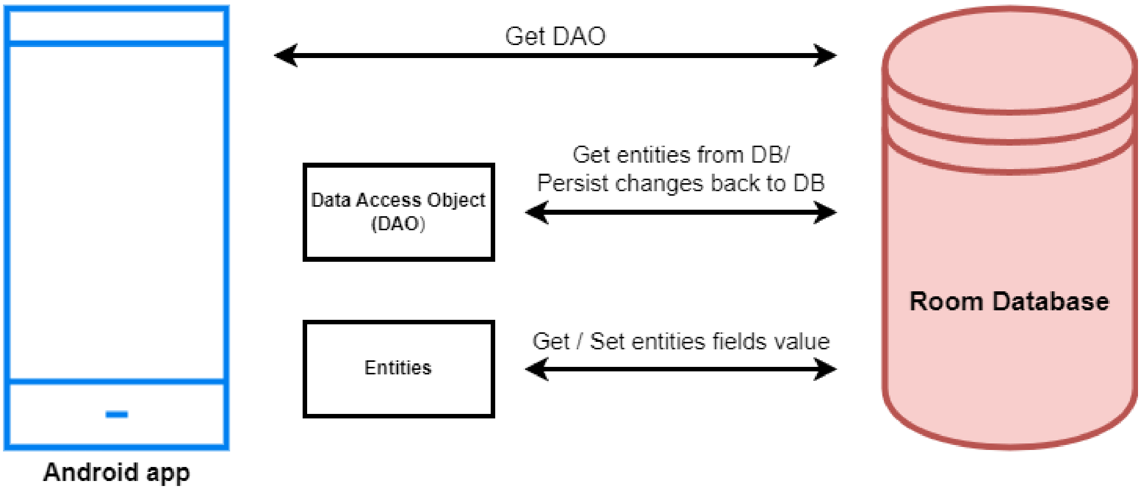
Handbook

SmartBridge Educational Services Pvt. Ltd.

Compose Input: A Demonstration of Text Input and Validation with Android Compose

The app is a sample project that demonstrates how to use the Android Compose UI toolkit to build a survey app. The app allows the user to answer a series of questions. It showcases some of the key features of the Compose UI toolkit, data management, and user interactions.

Architecture


Get DAO
Data Access Object (DAO)
Get entities from DB/ Persist changes back to DB
Entities
Get / Set entities fields value
Room Database
Android app

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 and fill the details
- From Admin Side he can login to the app and can view all the data.

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 :

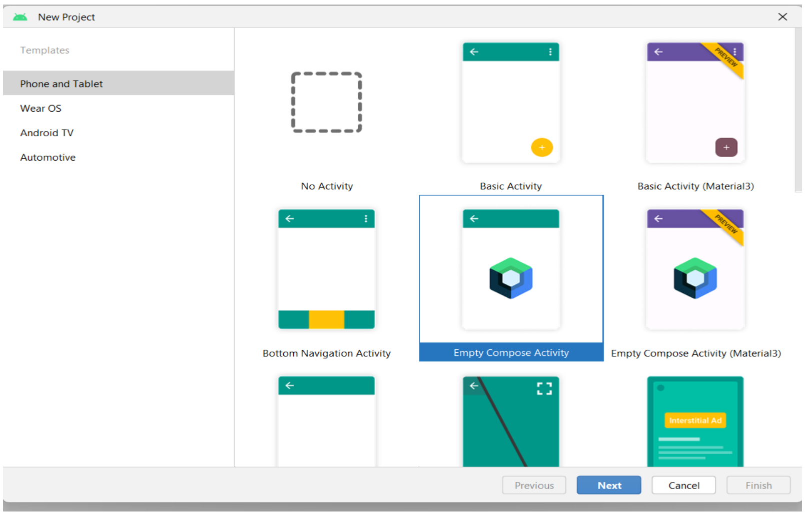
https://developer.android.com/studio/install

Task 2 :

Creating a new project.

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

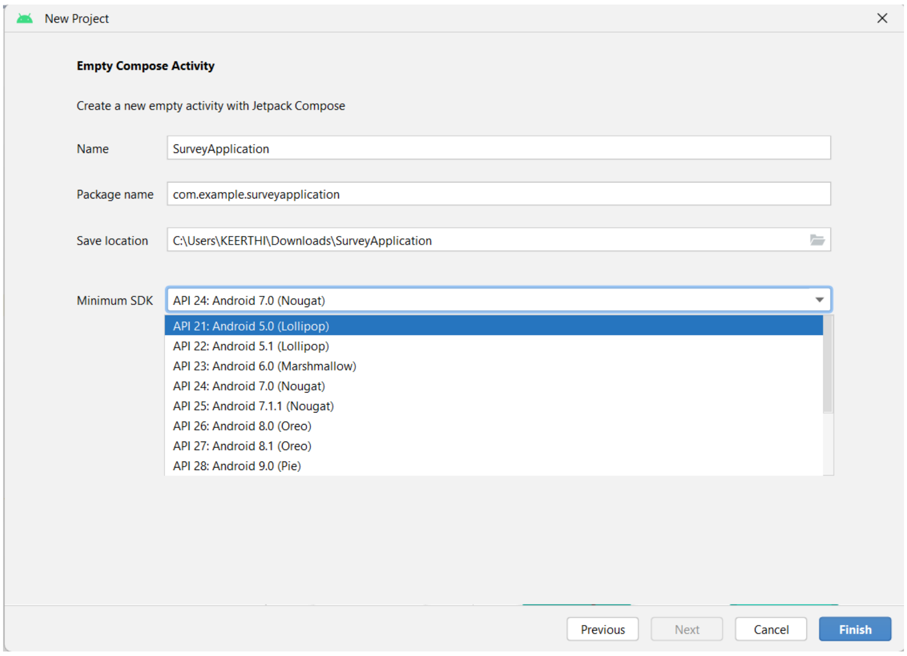
Step 2 : Click on **Next** button.



Step 3 : Give name to the new project.

Step 4 : Give the Minimum SDK value

Step 5 : Click Finish



New Project

Empty Compose Activity

Create a new empty activity with Jetpack Compose

Name: SurveyApplication

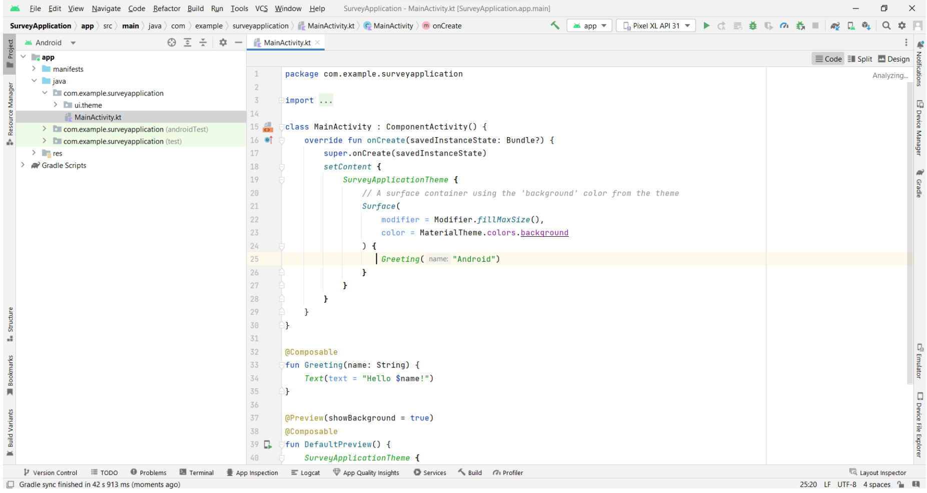
Package name: com.example.surveyapplication

Save location: C:\Users\KEERTHI\Downloads\SurveyApplication

Minimum SDK: API 24: Android 7.0 (Nougat)

- API 21: Android 5.0 (Lollipop)
- API 22: Android 5.1 (Lollipop)
- API 23: Android 6.0 (Marshmallow)
- API 24: Android 7.0 (Nougat)
- API 25: Android 7.1.1 (Nougat)
- API 26: Android 8.0 (Oreo)
- API 27: Android 8.1 (Oreo)
- API 28: Android 9.0 (Pie)

Main activity file



SurveyApplication - MainActivity.kt [SurveyApplication.app.main]

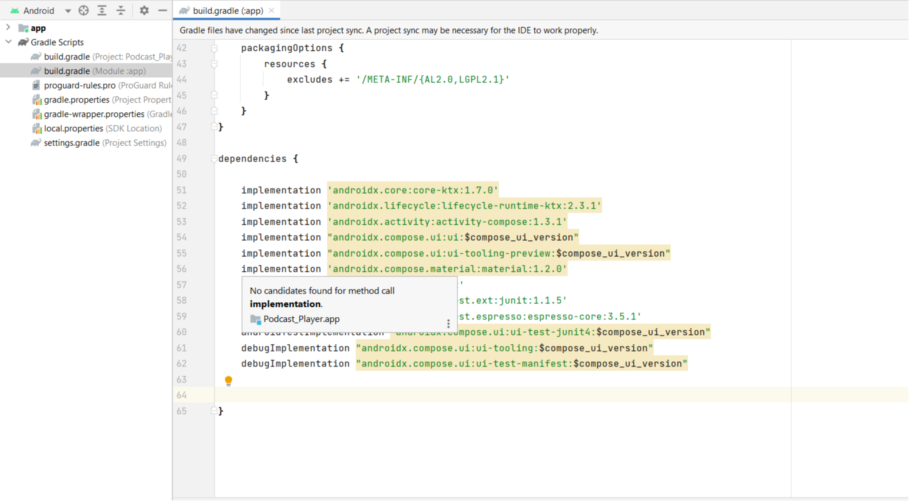
```
package com.example.surveyapplication

import ...

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            SurveyApplicationTheme {
                // A surface container using the 'background' color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    Greeting( name: "Android")
                }
            }
        }
    }
}

@Composable
fun Greeting(name: String) {
    Text(text = "Hello $name!")
}

@Preview(showBackground = true)
@Composable
fun DefaultPreview() {
    SurveyApplicationTheme {
```

Gradle sync finished in 42 s 913 ms (moments ago)

Task 3 :

Adding required dependencies.

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


Android
app
Gradle Scripts
build.gradle (Project: Podcast_Play
build.gradle (Module :app)
proguard-rules.pro (ProGuard Rul
gradle.properties (Project Propert
gradle-wrapper.properties (Gradl
local.properties (SDK Location)
settings.gradle (Project Settings)
build.gradle (:app)

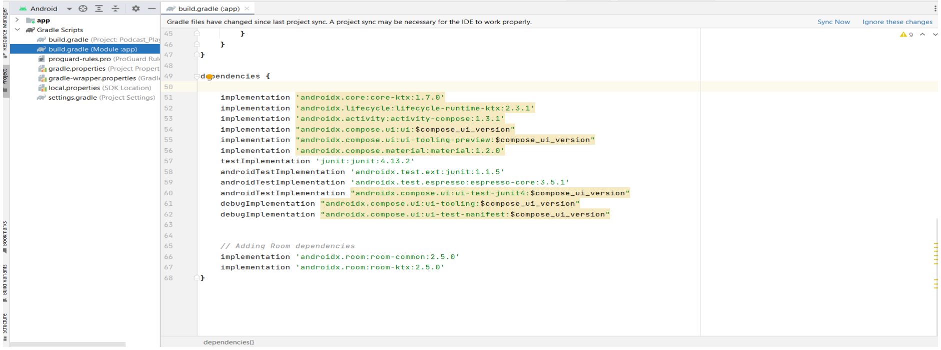
Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

```
    packagingOptions {
        resources {
            excludes += '/META-INF/{AL2.0,LGPL2.1}'
        }
    }
}

dependencies {

    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.3.1'
    implementation 'androidx.activity:activity-compose:1.3.1'
    implementation "androidx.compose.ui:ui:$compose_ui_version"
    implementation "androidx.compose.ui:ui-tooling-preview:$compose_ui_version"
    implementation 'androidx.compose.material:material:1.2.0'
                                                '
                                                st.ext:junit:1.1.5'
                                                st.espresso:espresso-core:3.5.1'
                                                ompose.ui:ui-test-junit4:$compose_ui_version"
    debugImplementation "androidx.compose.ui:ui-tooling:$compose_ui_version"
    debugImplementation "androidx.compose.ui:ui-test-manifest:$compose_ui_version"


}
```

No candidates found for method call **implementation**.
Podcast_Player.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'
```


Android
app
Gradle Scripts
build.gradle (Project: Podcast_Play
build.gradle (Module :app)
proguard-rules.pro (ProGuard Rul
gradle.properties (Project Propert
gradle-wrapper.properties (Gradl
local.properties (SDK Location)
settings.gradle (Project Settings)
build.gradle (:app)
Sync Now
Ignore these changes
Resource Manager
Project
Bookmarks
Build Variants
Structure
9
dependencies{}

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly.

```
        }
    }
}

dependencies {

    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.3.1'
    implementation 'androidx.activity:activity-compose:1.3.1'
    implementation "androidx.compose.ui:ui:$compose_ui_version"
    implementation "androidx.compose.ui:ui-tooling-preview:$compose_ui_version"
    implementation 'androidx.compose.material:material:1.2.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    androidTestImplementation "androidx.compose.ui:ui-test-junit4:$compose_ui_version"
    debugImplementation "androidx.compose.ui:ui-tooling:$compose_ui_version"
    debugImplementation "androidx.compose.ui:ui-test-manifest:$compose_ui_version"


    // 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.

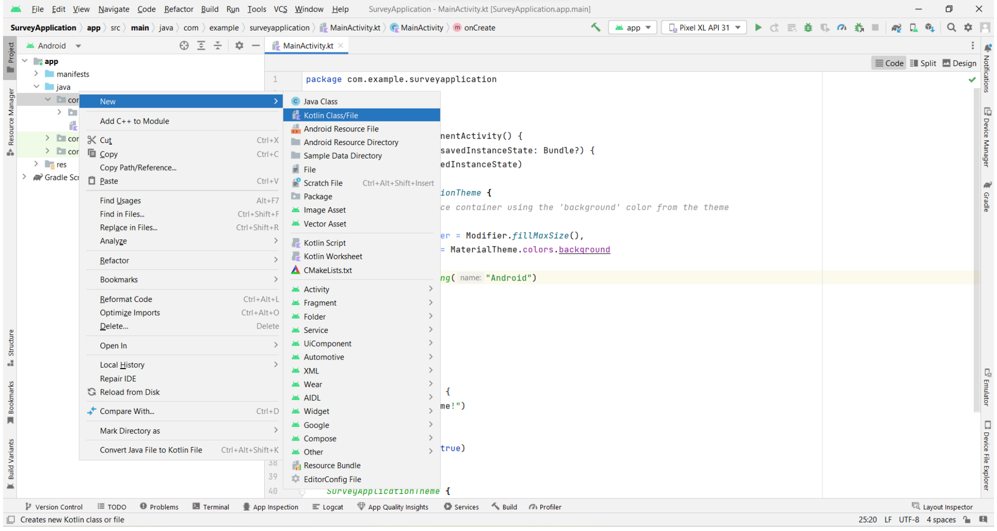
- To learn more about Database follow this link:

https://gorillalogic.com/blog/android-room-tutorial-simplifying-how-you-work-with-app-data/

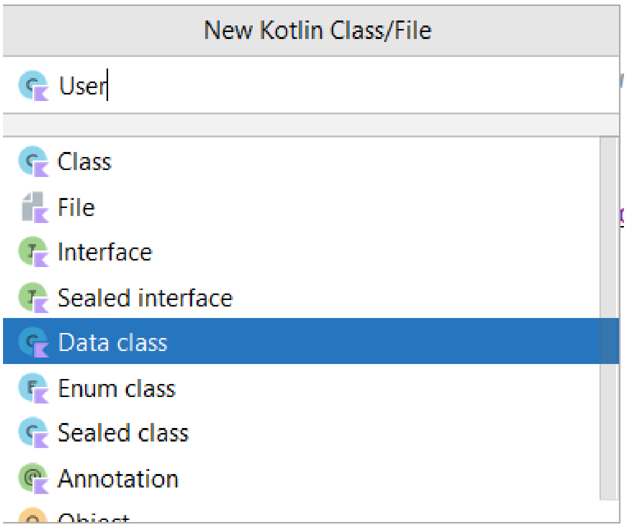
In this project we will be having two databases, one is for user registration and login, and other is for admin to view the details of survey forms.

Database 1

Step 1 : Create User data class



```
package com.example.surveyapplication
nentActivity() {
savedInstanceState: Bundle?) {
edInstanceState)
ionTheme {
ce container using the 'background' color from the theme
er = Modifier.fillMaxSize(),
= MaterialTheme.colors.background
ng( name: "Android")
{
me!")
true)
SurveyApplicationTheme {
```

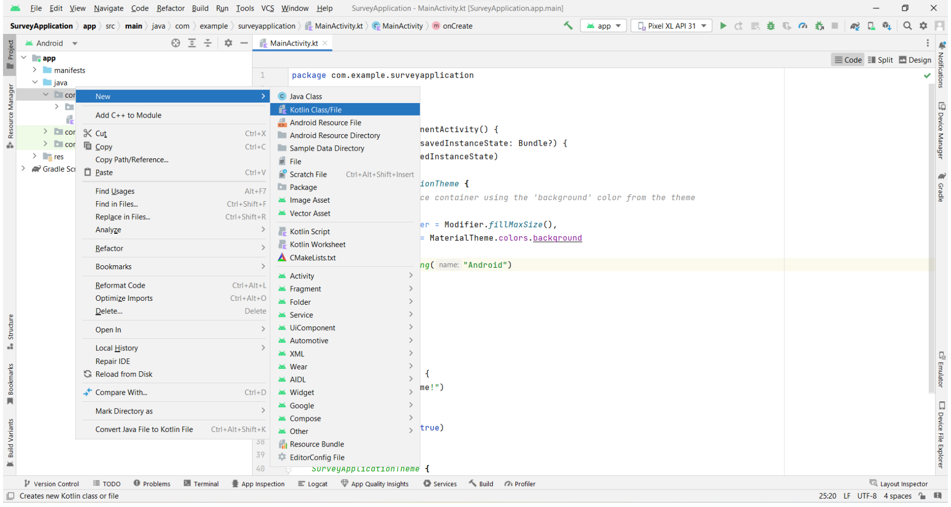


New Kotlin Class/File

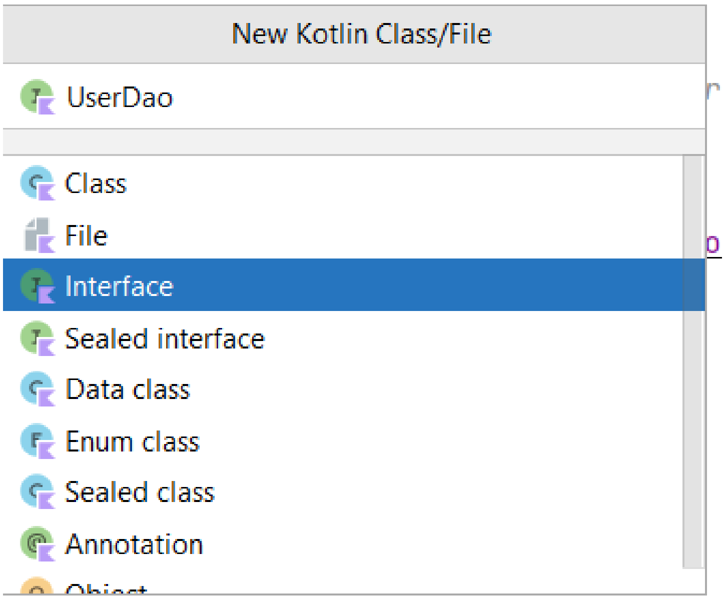
User

User class code:
https://github.com/smartinternz02/SurveyApp/tree/main/app/src/main/java/com/example/surveyapplication/User.kt

Step 2 : Create an UserDao interface



```
package com.example.surveyapplication
nentActivity() {
savedInstanceState: Bundle?) {
edInstanceState)
ionTheme {
ce container using the 'background' color from the theme
er = Modifier.fillMaxSize(),
= MaterialTheme.colors.background
ng( name: "Android")
{
me!")
true)
SurveyApplicationTheme {
```

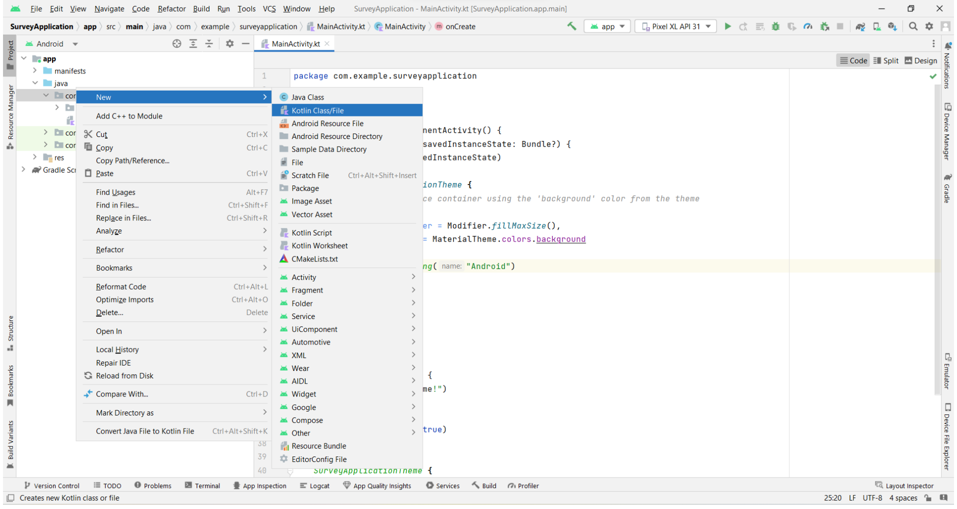


New Kotlin Class/File

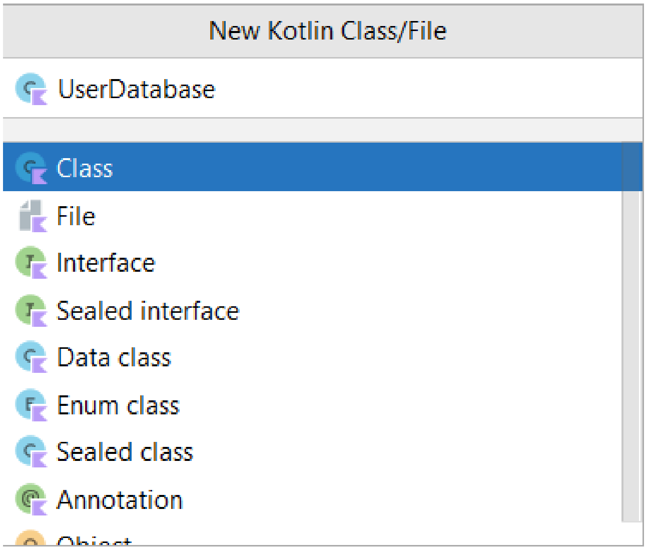
UserDao

UserDao interface code:
https://github.com/smartinternz02/SurveyApp/tree/main/app/src/main/java/com/example/surveyapplication/UserDao.kt

Step 3 : Create an UserDatabase class

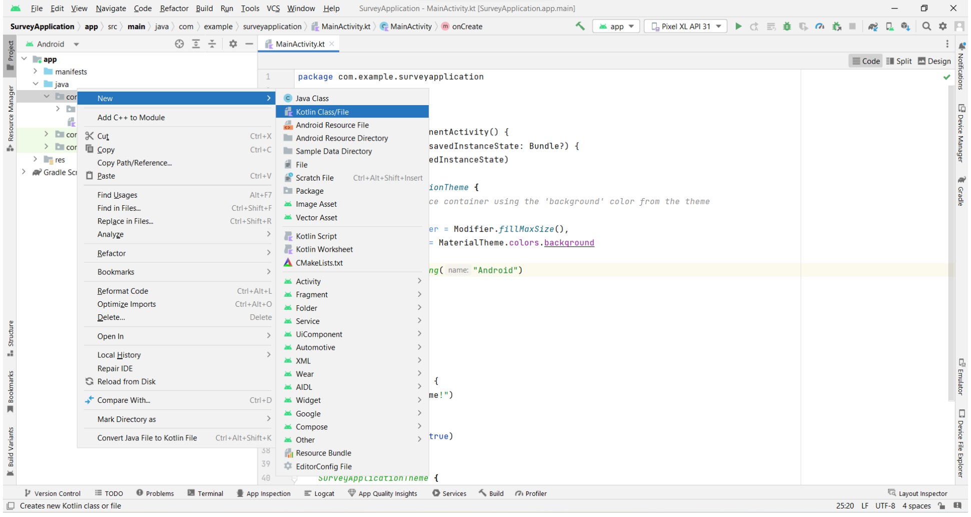


```
package com.example.surveyapplication

nentActivity() {
savedInstanceState: Bundle?) {
edInstanceState)

ionTheme {
ce container using the 'background' color from the theme

er = Modifier.fillMaxSize(),
= MaterialTheme.colors.background

ng( name: "Android")

 {
me!")

true)

SurveyApplicationTheme {
```

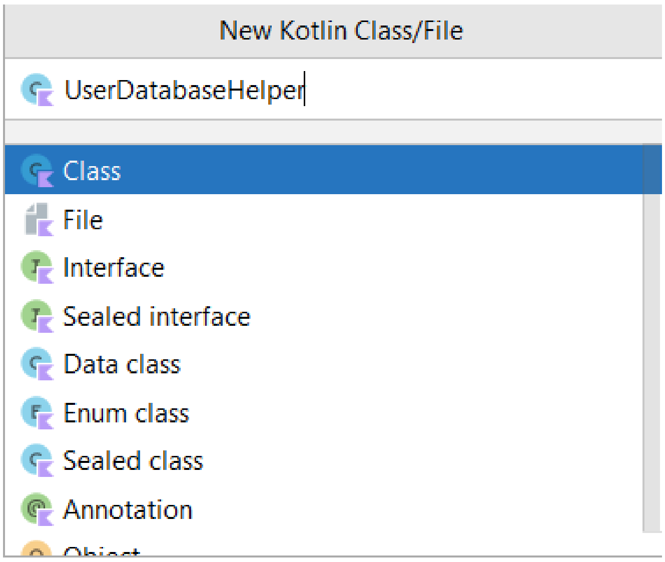


UserDatabase class code : https://github.com/smartinternz02/SurveyApp/tree/main/app/src/main/java/com/example/surveyapplication/UserDatabase.kt

Step 4 : Create an UserDatabaseHelper class



```
package com.example.surveyapplication
nentActivity() {
savedInstanceState: Bundle?) {
edInstanceState)
ionTheme {
ce container using the 'background' color from the theme
er = Modifier.fillMaxSize(),
= MaterialTheme.colors.background
ng( name: "Android")
{
me!")
true)
SurveyApplicationTheme {
```

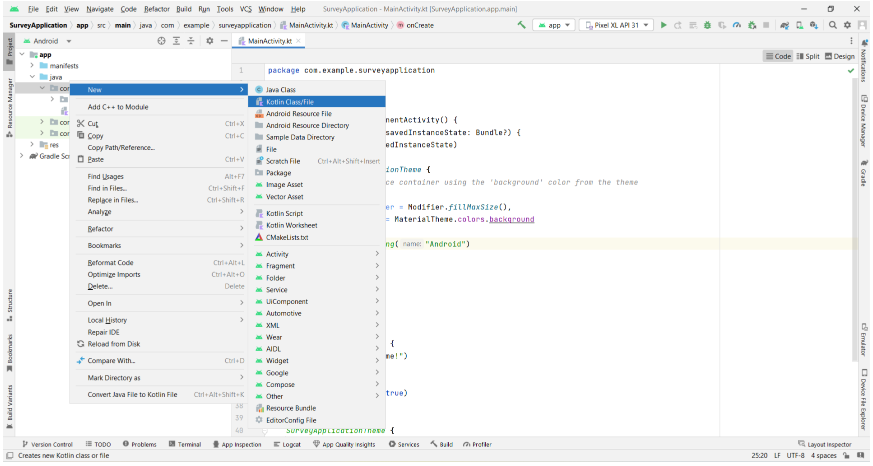


UserDatabaseHelper

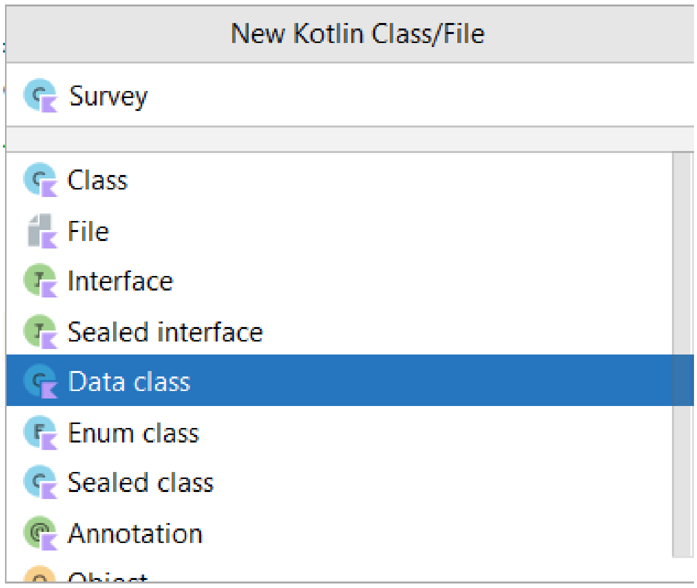
UserDatabaseHelper class code : https://github.com/smartinternz02/SurveyApp/tree/main/app/src/main/java/com/example/surveyapplication/UserDatabaseHelper.kt

Database 2

Step 1 : Create Survey data class

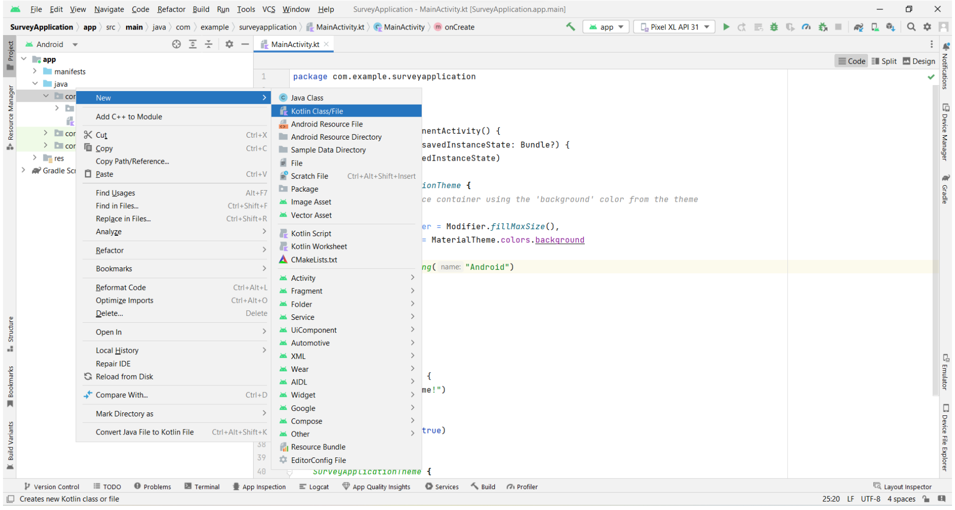


```
package com.example.surveyapplication
```

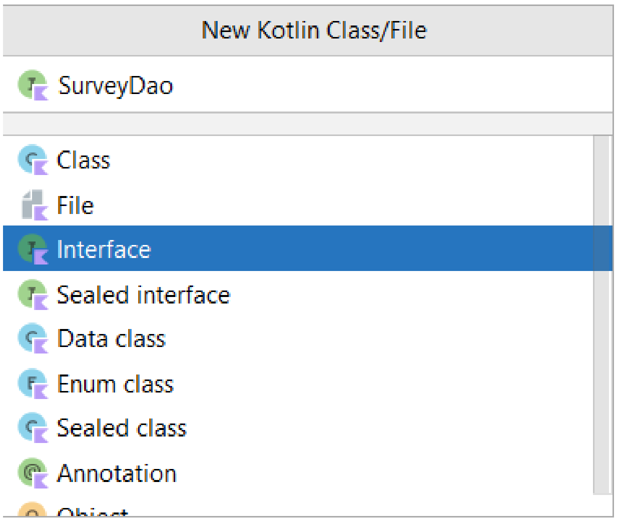


Survey data class code:
https://github.com/smartinternz02/SurveyApp/tree/main/app/src/main/java/com/example/surveyapplication/Survey.kt

Step 2 : Create SurveyDao interface

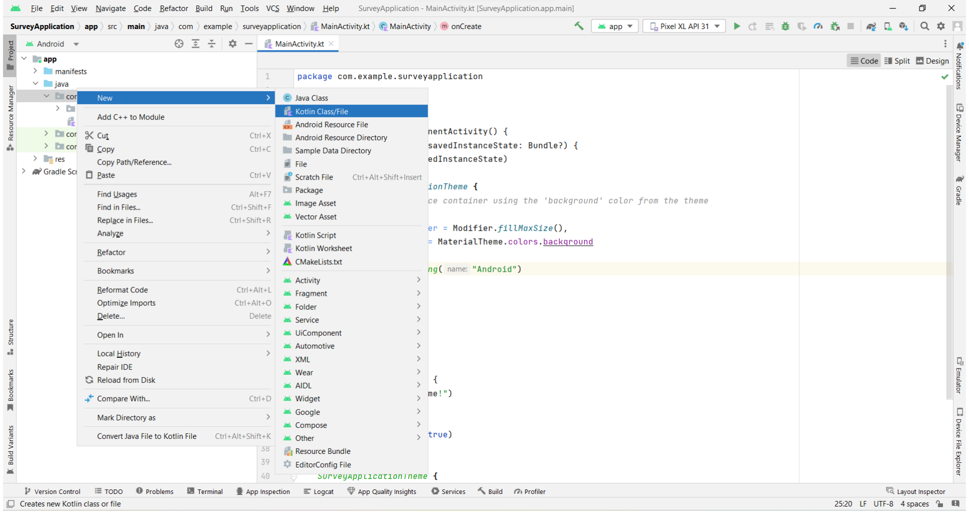


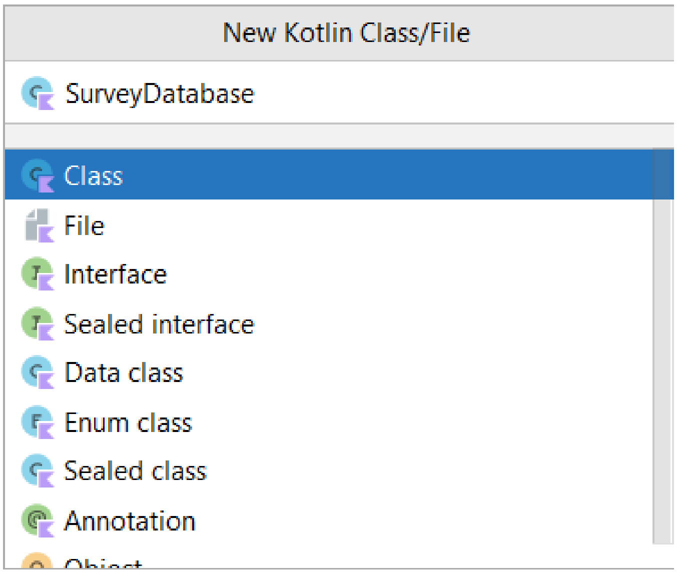
```
package com.example.surveyapplication

nentActivity() {
savedInstanceState: Bundle?) {
edInstanceState)

ionTheme {
ce container using the 'background' color from the theme

er = Modifier.fillMaxSize(),
= MaterialTheme.colors.background

ng( name: "Android")

 {
me!")

true)

SurveyApplicationTheme {
```



SurveyDao interface code:
https://github.com/smartinternz02/SurveyApp/tree/main/app/src/main/java/com/example/surveyapplication/SurveyDao.kt

Step 3 : Create SurveyDatabase class


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
SurveyApplication - MainActivity.kt [SurveyApplication.app.main]
SurveyApplication app src main java com example surveyapplication MainActivity.kt MainActivity onCreate
Android
app
manifests
java
res
Gradle Scripts
MainActivity.kt
New
Add C++ to Module
Cut Ctrl+X
Copy Ctrl+C
Copy Path/Reference...
Paste Ctrl+V
Find Usages Alt+F7
Find in Files... Ctrl+Shift+F
Replace in Files... Ctrl+Shift+R
Analyze
Refactor
Bookmarks
Reformat Code Ctrl+Alt+L
Optimize Imports Ctrl+Alt+O
Delete... Delete
Open In
Local History
Repair IDE
Reload from Disk
Compare With... Ctrl+D
Mark Directory as
Convert Java File to Kotlin File Ctrl+Alt+Shift+K
Java Class
Kotlin Class/File
Android Resource File
Android Resource Directory
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Package
Image Asset
Vector Asset
Kotlin Script
Kotlin Worksheet
CMakeLists.txt
Activity
Fragment
Folder
Service
UiComponent
Automotive
XML
Wear
AIDL
Widget
Google
Compose
Other
Resource Bundle
EditorConfig File
package com.example.surveyapplication
Version Control TODO Problems Terminal App Inspection Logcat App Quality Insights Services Build Profiler
Creates new Kotlin class or file

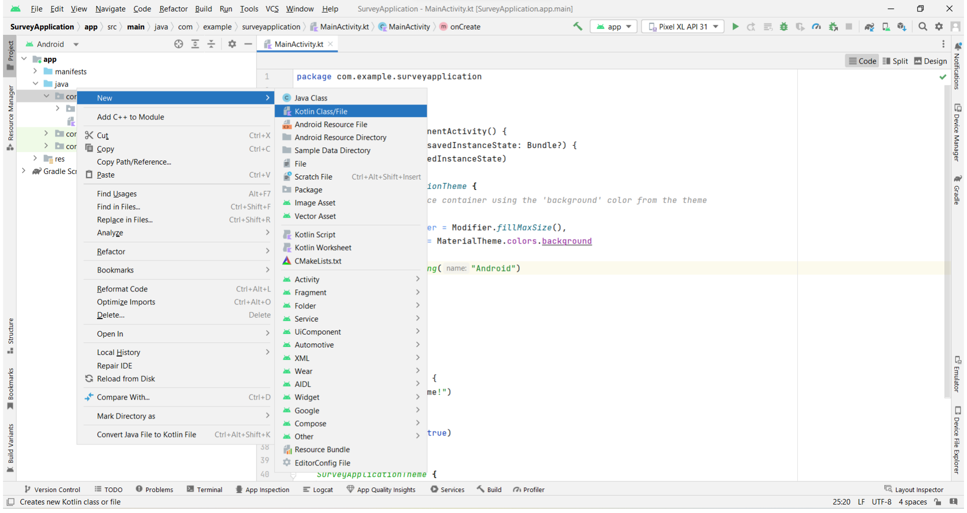


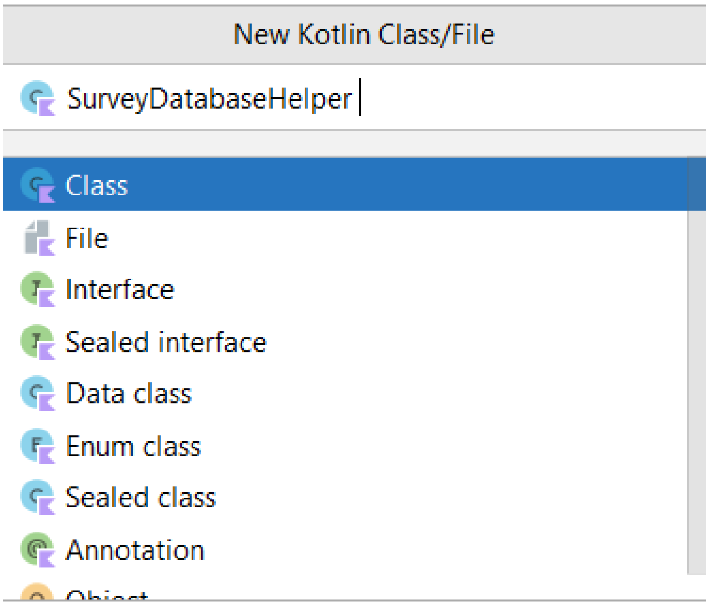
New Kotlin Class/File

SurveyDatabase

SurveyDatabase class code: https://github.com/smartinternz02/SurveyApp/tree/main/app/src/main/java/com/example/surveyapplication/SurveyDatabase.kt

Step 4 : Create SurveyDatabaseHelper class





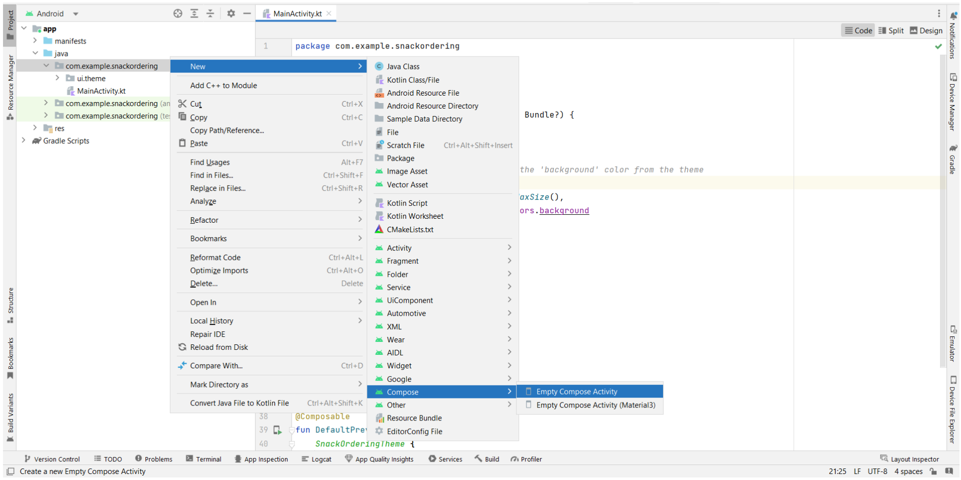
SurveyDatabaseHelper

SurveyDatabaseHelper class code: https://github.com/smartinternz02/SurveyApp/tree/main/app/src/main/java/com/example/surveyapplication/SurveyDatabaseHelper.kt

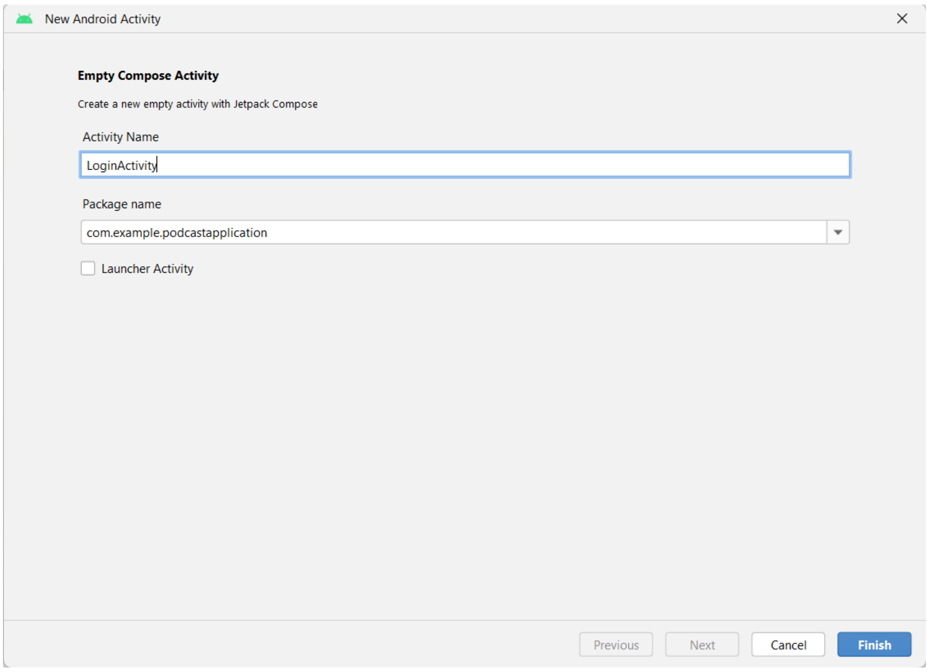
Task 5:

Building application UI and connecting to database.

Step 1: Creating LoginActivity.kt with database



```
package com.example.snackordering
Bundle?) {
the 'background' color from the theme
axSize(),
ors.background
@Composable
fun DefaultPre
    SnackOrderingTheme {
```



Empty Compose Activity

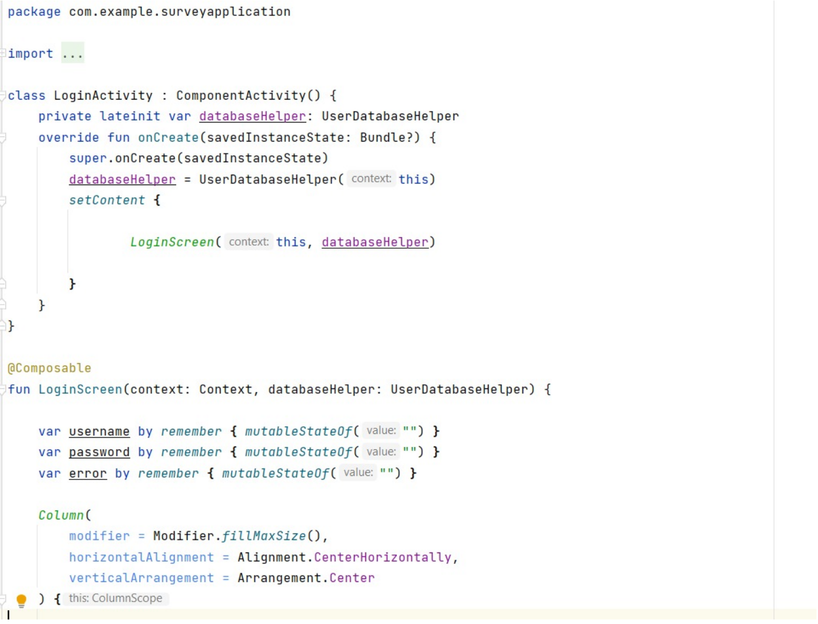
Create a new empty activity with Jetpack Compose

Activity Name: LoginActivity

Package name: com.example.podcastapplication

□ Launcher Activity

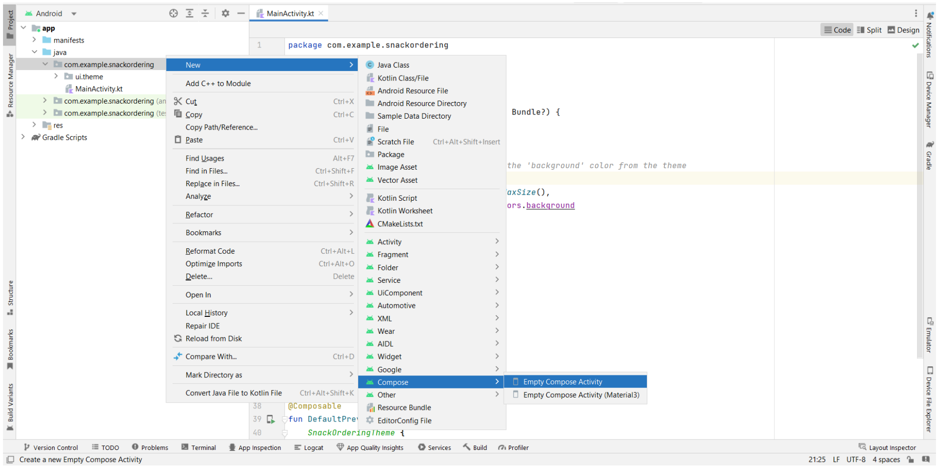
Database connection in LoginActivity.kt



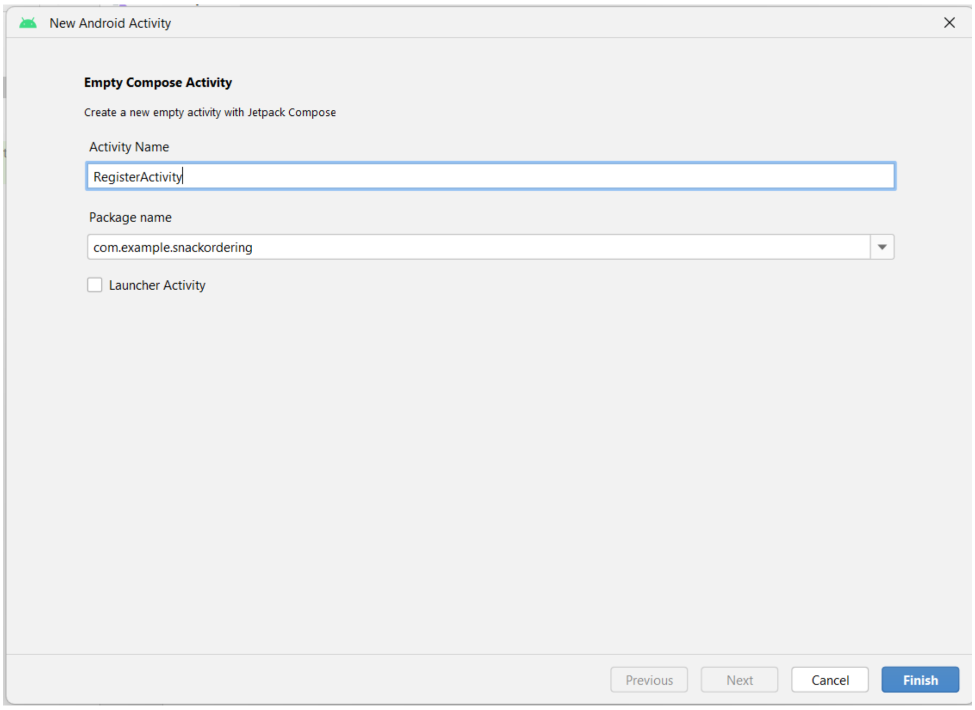
```kotlin
package com.example.surveyapplication

import ...

class LoginActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(this)
        setContent {

                LoginScreen(this, databaseHelper)

        }
    }
}

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

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

    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
```

Complete code in below link:
https://github.com/smartinternz02/SurveyApp/tree/main/app/src/main/java/com/example/surveyapplication/LoginActivity.kt

Step 2 : Creating RegisterActivity.kt with database



```
package com.example.snackordering
```



Empty Compose Activity

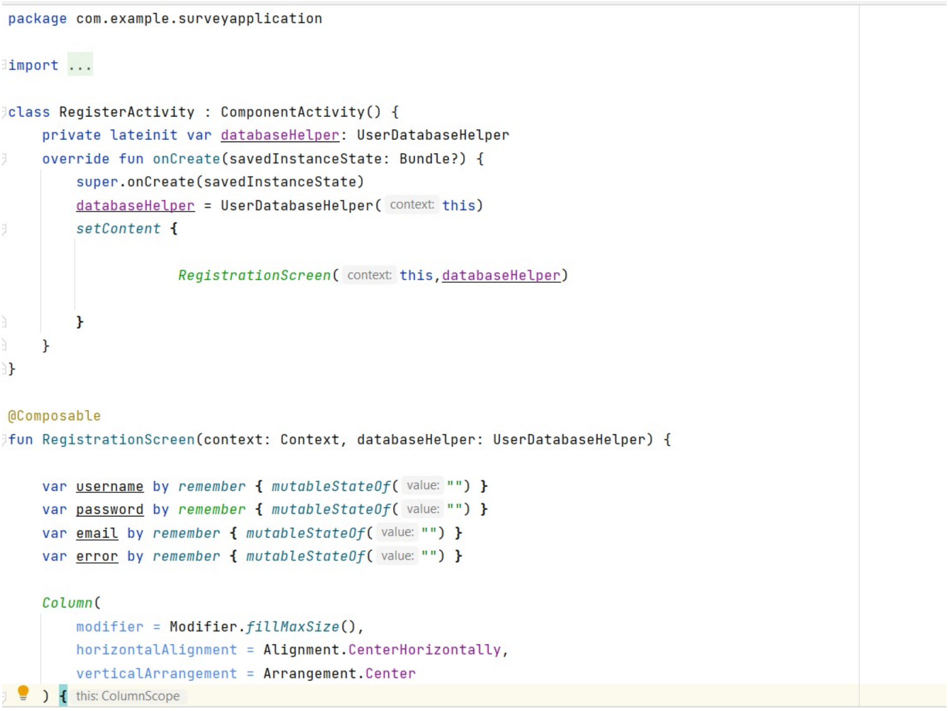
Create a new empty activity with Jetpack Compose

Activity Name: RegisterActivity

Package name: com.example.snackordering

☐ Launcher Activity

Database connection in RegisterActivity.kt



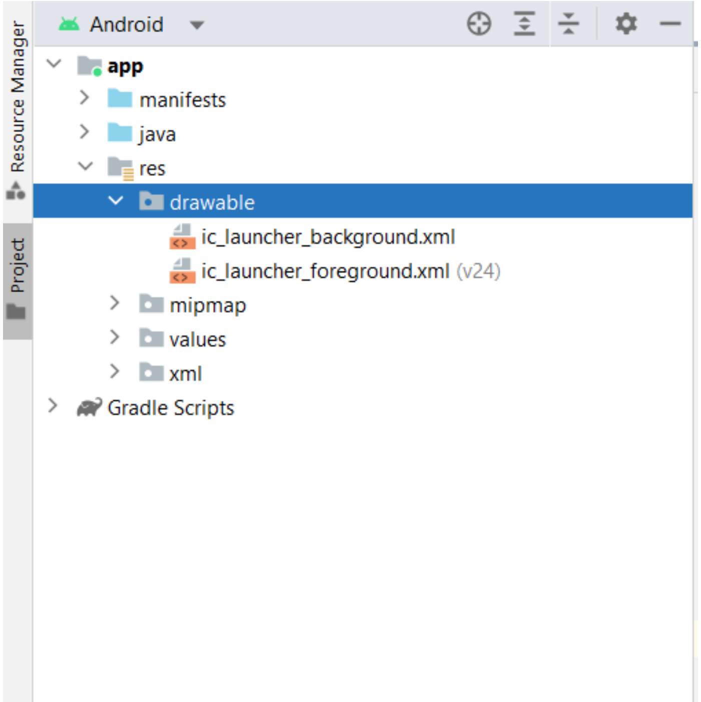
```
package com.example.surveyapplication

import ...

class RegisterActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper( context: this)
        setContent {

                  RegistrationScreen( context: this,databaseHelper)

        }
    }
}

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

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

    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) { this: ColumnScope
```

Complete code in below link:
https://github.com/smartinternz02/SurveyApp/tree/main/app/src/main/java/com/example/surveyapplication/RegisterActivity.kt

Step 3 : Creating MainActivity.kt file

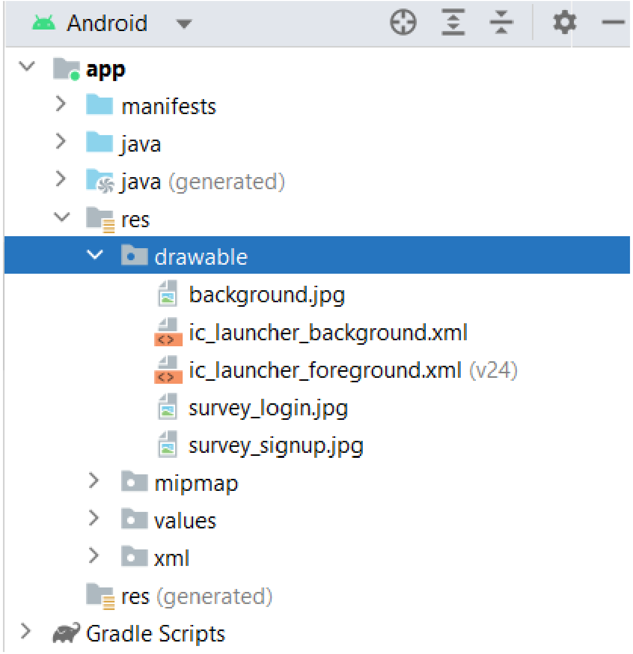
In MainActivity.kt file the main application is developed

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

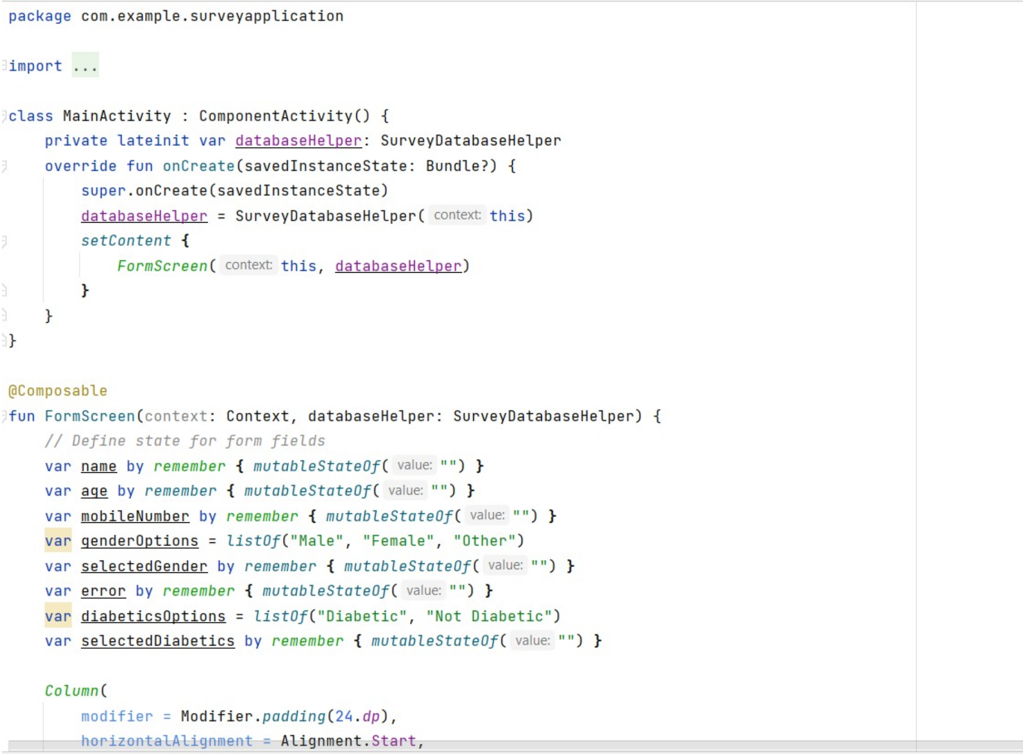


Download the required drawable from the code:
https://github.com/smartinternz02/SurveyApp/tree/master/app/src/main/res/drawable

Required drawables



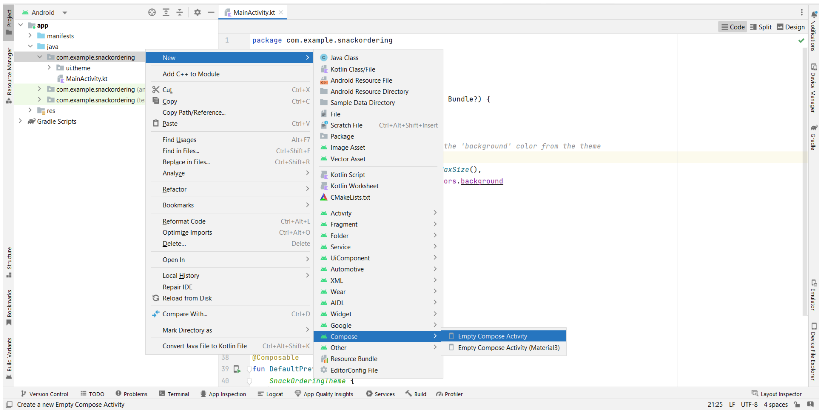
MainActivity.kt



```
package com.example.surveyapplication

import ...

class MainActivity : ComponentActivity() {
    private lateinit var databaseHelper: SurveyDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = SurveyDatabaseHelper( context: this)
        setContent {
            FormScreen( context: this, databaseHelper)
        }
    }
}

@Composable
fun FormScreen(context: Context, databaseHelper: SurveyDatabaseHelper) {
    // Define state for form fields
    var name by remember { mutableStateOf( value: "") }
    var age by remember { mutableStateOf( value: "") }
    var mobileNumber by remember { mutableStateOf( value: "") }
    var genderOptions = listOf("Male", "Female", "Other")
    var selectedGender by remember { mutableStateOf( value: "") }
    var error by remember { mutableStateOf( value: "") }
    var diabeticsOptions = listOf("Diabetic", "Not Diabetic")
    var selectedDiabetics by remember { mutableStateOf( value: "") }

    Column(
        modifier = Modifier.padding(24.dp),
        horizontalAlignment = Alignment.Start,
```

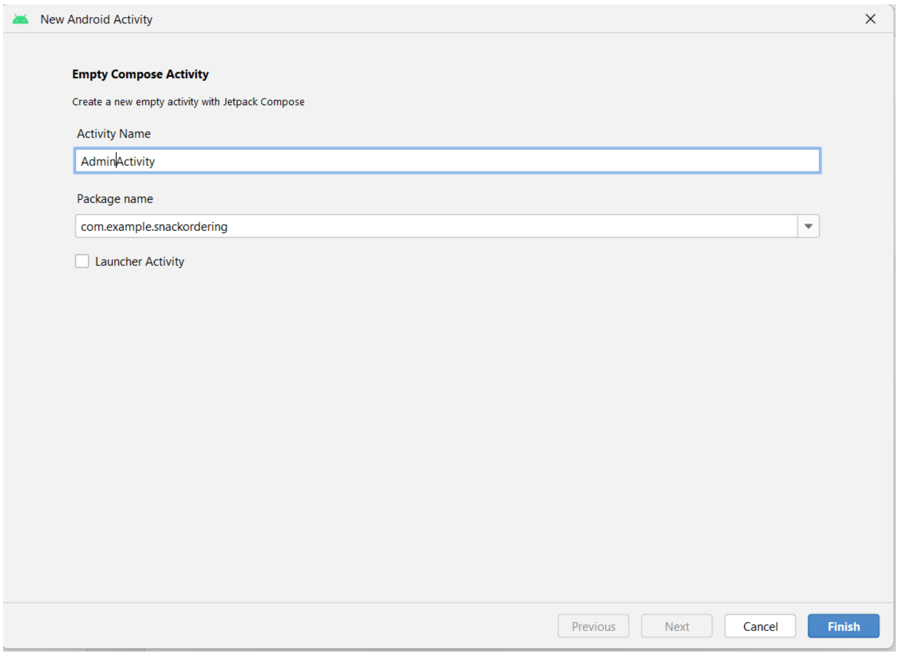
Complete code in below link:
https://github.com/smartinternz02/SurveyApp/tree/main/app/src/main/java/com/example/surveyapplication/MainActivity.kt

Step 4 : Creating AdminActivity.kt file



```
package com.example.snackordering
```

Create a new Empty Compose Activity



Empty Compose Activity

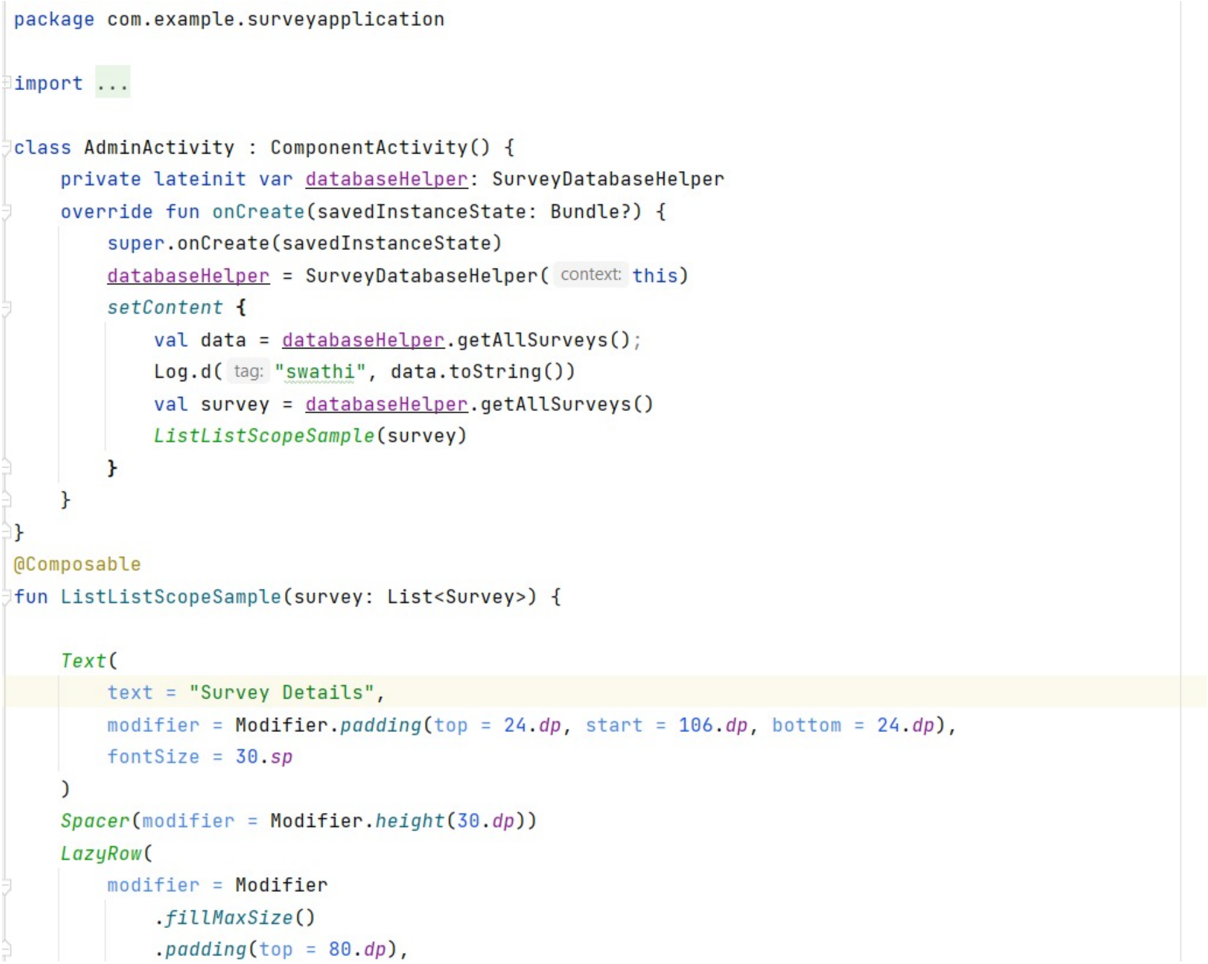
Create a new empty activity with Jetpack Compose

Activity Name: AdminActivity

Package name: com.example.snackordering

☐ Launcher Activity

AdminActivity.kt

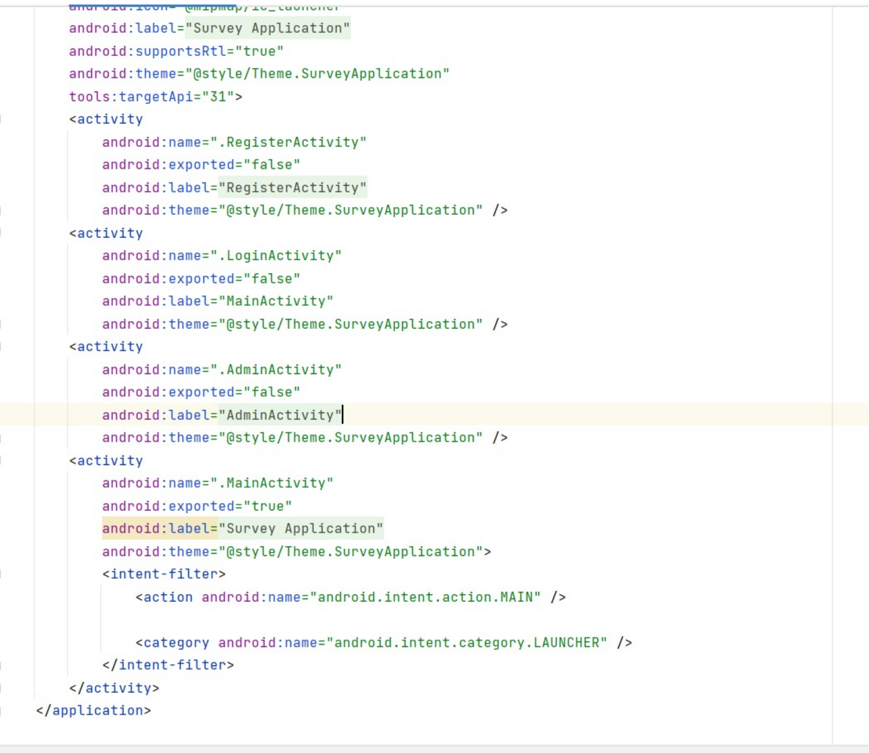


```kotlin
package com.example.surveyapplication

import ...

class AdminActivity : ComponentActivity() {
    private lateinit var databaseHelper: SurveyDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = SurveyDatabaseHelper( context: this)
        setContent {
            val data = databaseHelper.getAllSurveys();
            Log.d( tag: "swathi", data.toString())
            val survey = databaseHelper.getAllSurveys()
            ListListScopeSample(survey)
        }
    }
}
@Composable
fun ListListScopeSample(survey: List<Survey>) {

    Text(
        text = "Survey Details",
        modifier = Modifier.padding(top = 24.dp, start = 106.dp, bottom = 24.dp),
        fontSize = 30.sp
    )
    Spacer(modifier = Modifier.height(30.dp))
    LazyRow(
        modifier = Modifier
            .fillMaxSize()
            .padding(top = 80.dp),
```

Complete code in below link:
https://github.com/smartinternz02/SurveyApp/tree/main/app/src/main/java/com/example/surveyapplication/AdminActivity.kt

Task 6:

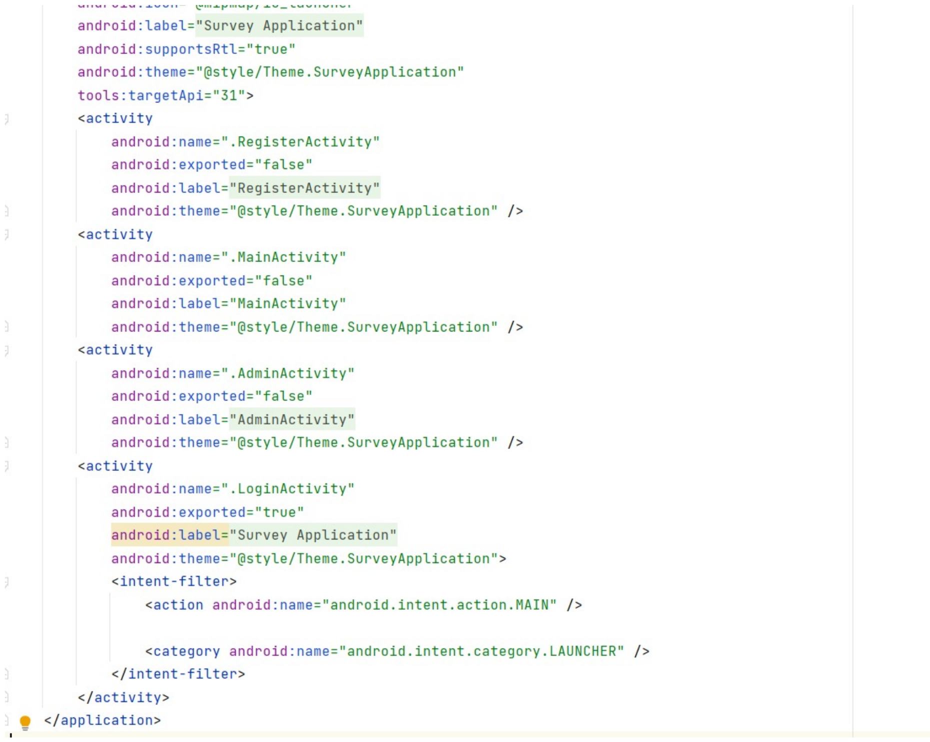
Modifying AndroidManifest.xml



```
    android:icon="@mipmap/ic_launcher"
    android:label="Survey Application"
    android:supportsRtl="true"
    android:theme="@style/Theme.SurveyApplication"
    tools:targetApi="31">
    <activity
        android:name=".RegisterActivity"
        android:exported="false"
        android:label="RegisterActivity"
        android:theme="@style/Theme.SurveyApplication" />
    <activity
        android:name=".LoginActivity"
        android:exported="false"
        android:label="MainActivity"
        android:theme="@style/Theme.SurveyApplication" />
    <activity
        android:name=".AdminActivity"
        android:exported="false"
        android:label="AdminActivity"
        android:theme="@style/Theme.SurveyApplication" />
    <activity
        android:name=".MainActivity"
        android:exported="true"
        android:label="Survey Application"
        android:theme="@style/Theme.SurveyApplication">
        <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:icon="@mipmap/ic_launcher"
        android:label="Survey Application"
        android:supportsRtl="true"
        android:theme="@style/Theme.SurveyApplication"
        tools:targetApi="31">
        <activity
            android:name=".RegisterActivity"
            android:exported="false"
            android:label="RegisterActivity"
            android:theme="@style/Theme.SurveyApplication" />
        <activity
            android:name=".MainActivity"
            android:exported="false"
            android:label="MainActivity"
            android:theme="@style/Theme.SurveyApplication" />
        <activity
            android:name=".AdminActivity"
            android:exported="false"
            android:label="AdminActivity"
            android:theme="@style/Theme.SurveyApplication" />
        <activity
            android:name=".LoginActivity"
            android:exported="true"
            android:label="Survey Application"
            android:theme="@style/Theme.SurveyApplication">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

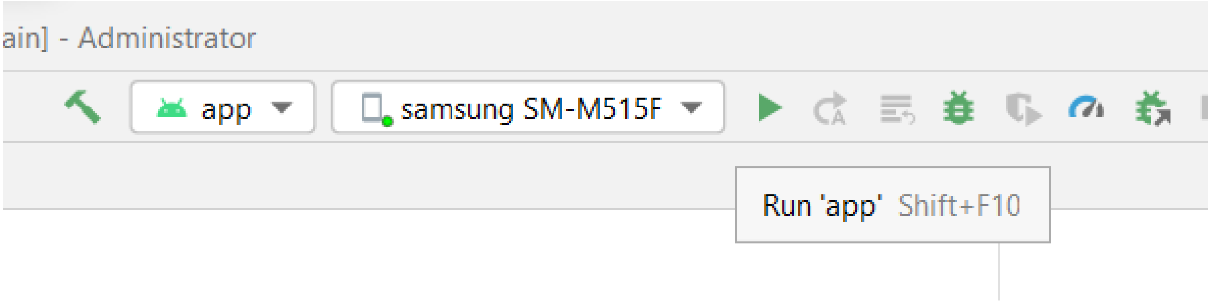
Complete AndroidManifest.xml code:
https://github.com/smartinternz02/SurveyApp/blob/master/app/src/main/AndroidManifest.xml

Task 7:

Running the application.

Step 1: Run apps on a hardware device
https://developer.android.com/studio/run/device

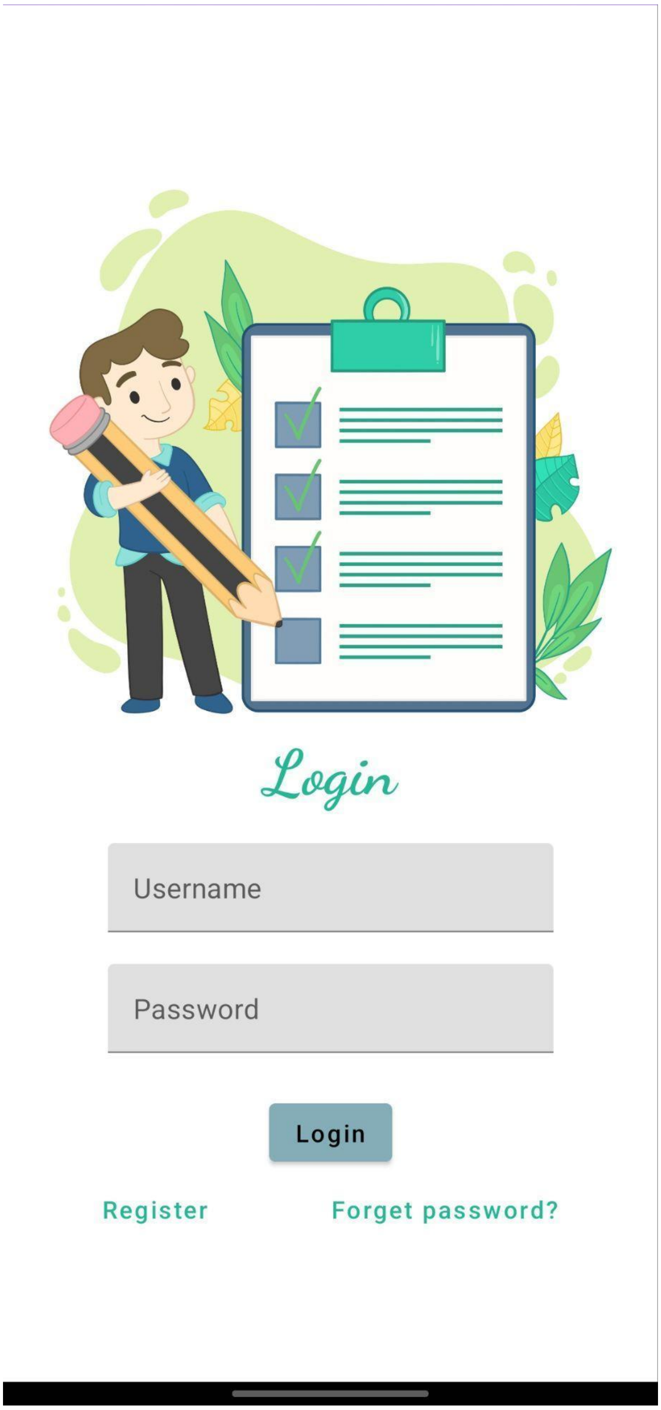
Step 2: Run the application in Mobile



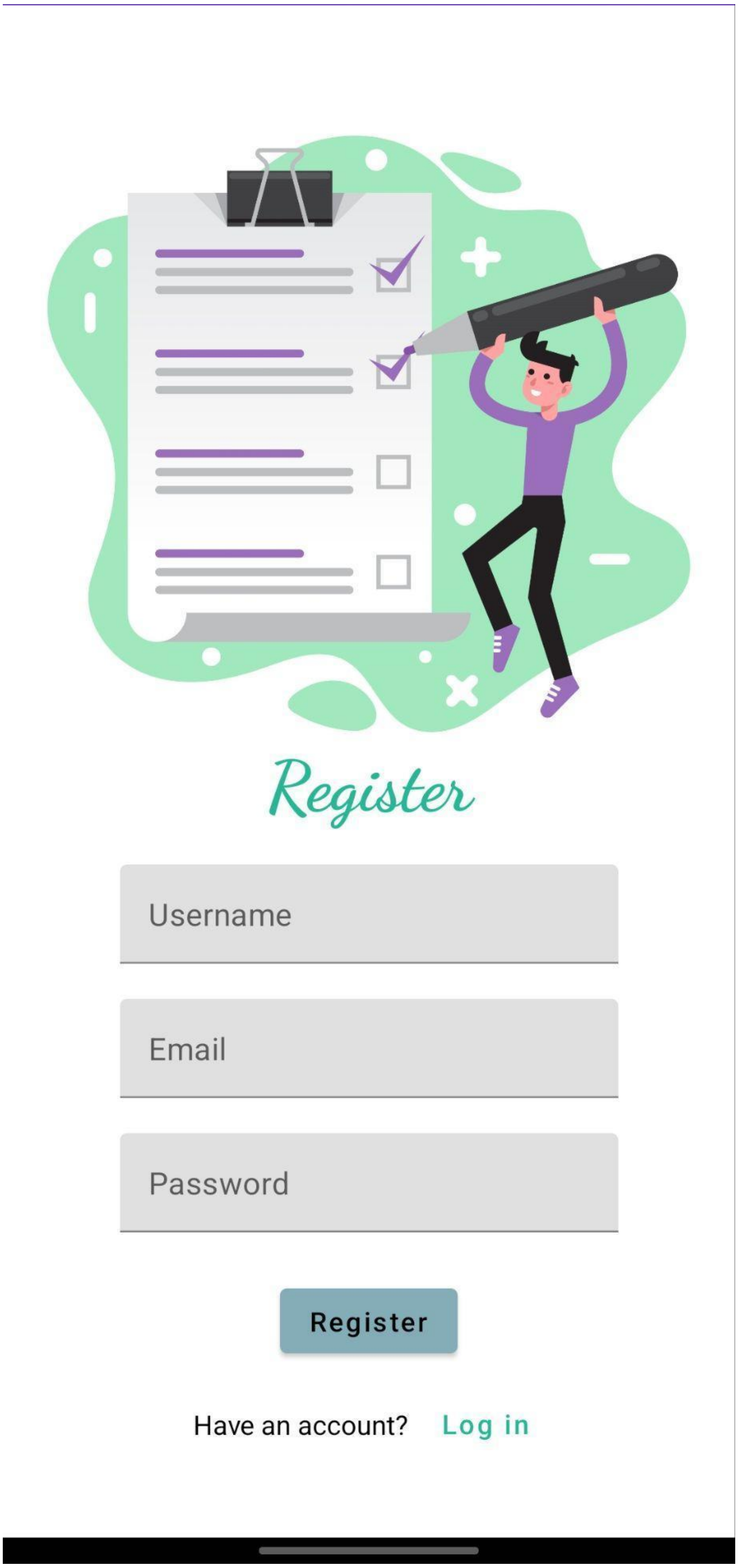
Complete Project Link: https://github.com/smartinternz02/SurveyApp

Final Output of the Application :

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

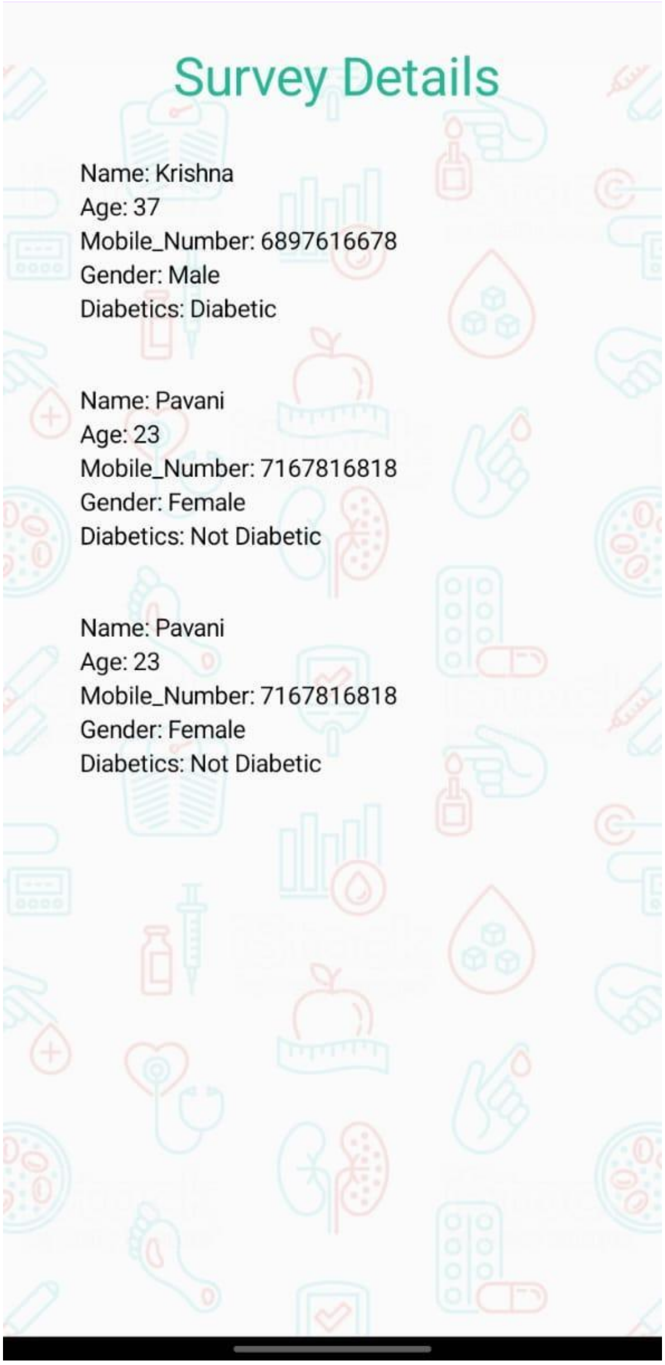


Register Page :



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

Admin page:



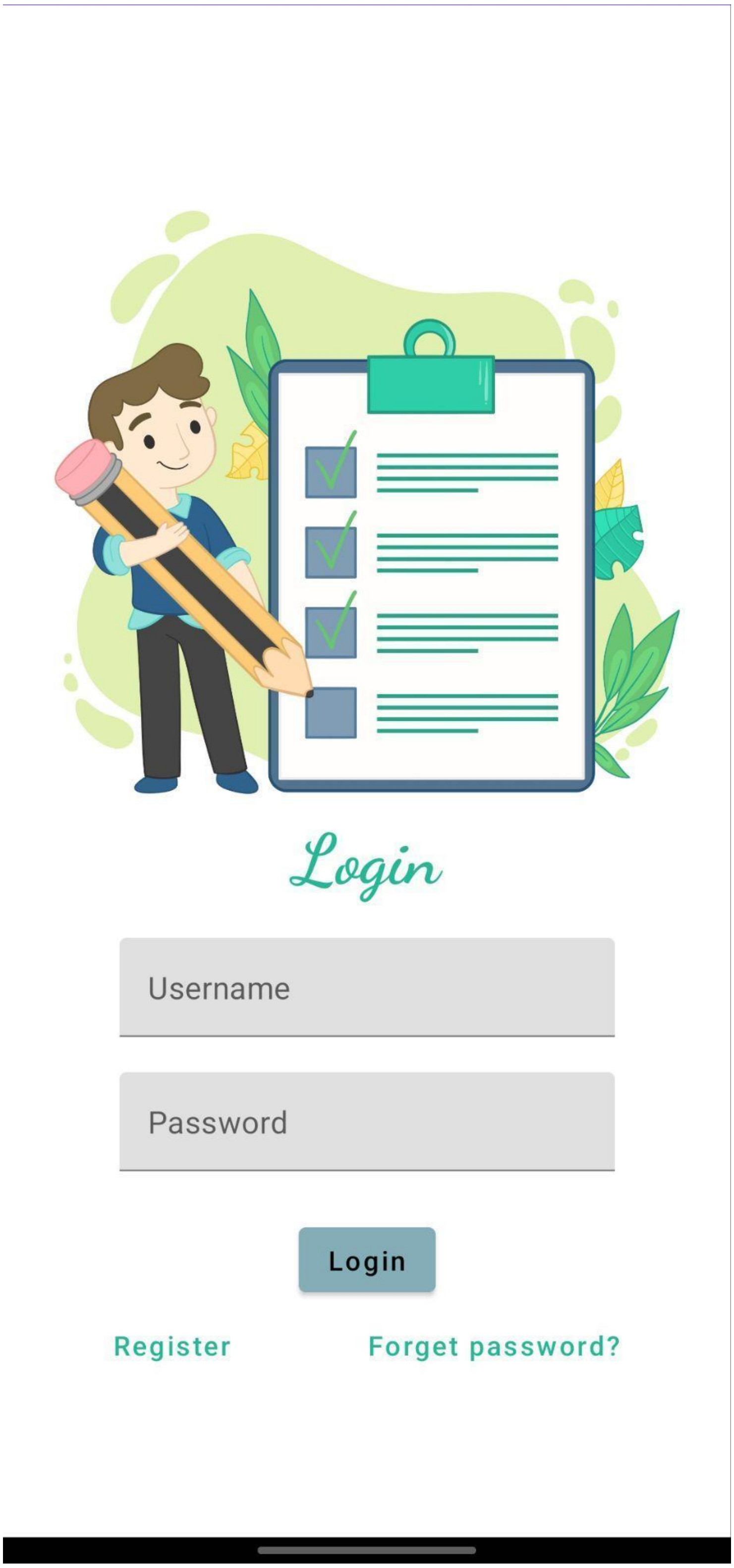
Survey Details

Name: Krishna
Age: 37
Mobile_Number: 6897616678
Gender: Male
Diabetics: Diabetic

Name: Pavani
Age: 23
Mobile_Number: 7167816818
Gender: Female
Diabetics: Not Diabetic

Name: Pavani
Age: 23
Mobile_Number: 7167816818
Gender: Female
Diabetics: Not Diabetic

User Module:
Login Page :

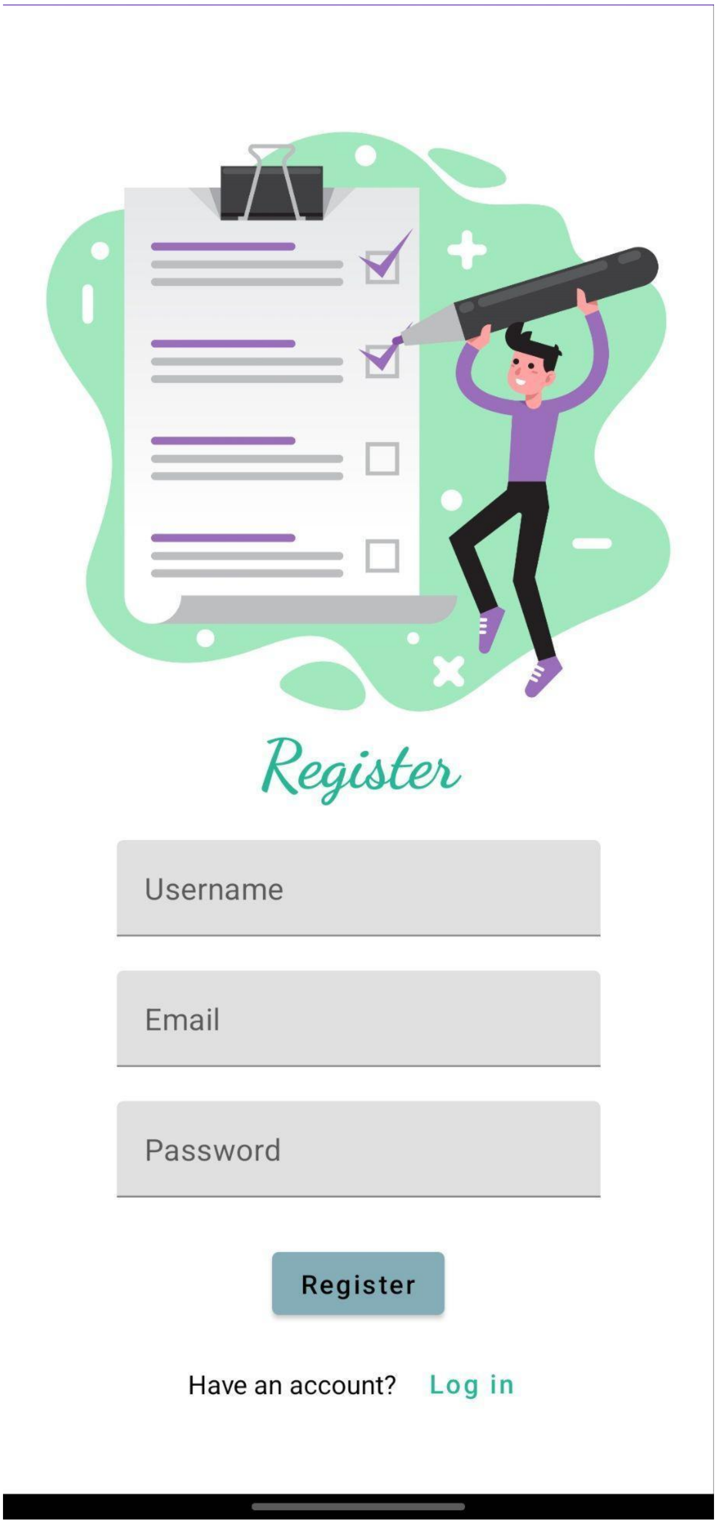


Login

Username

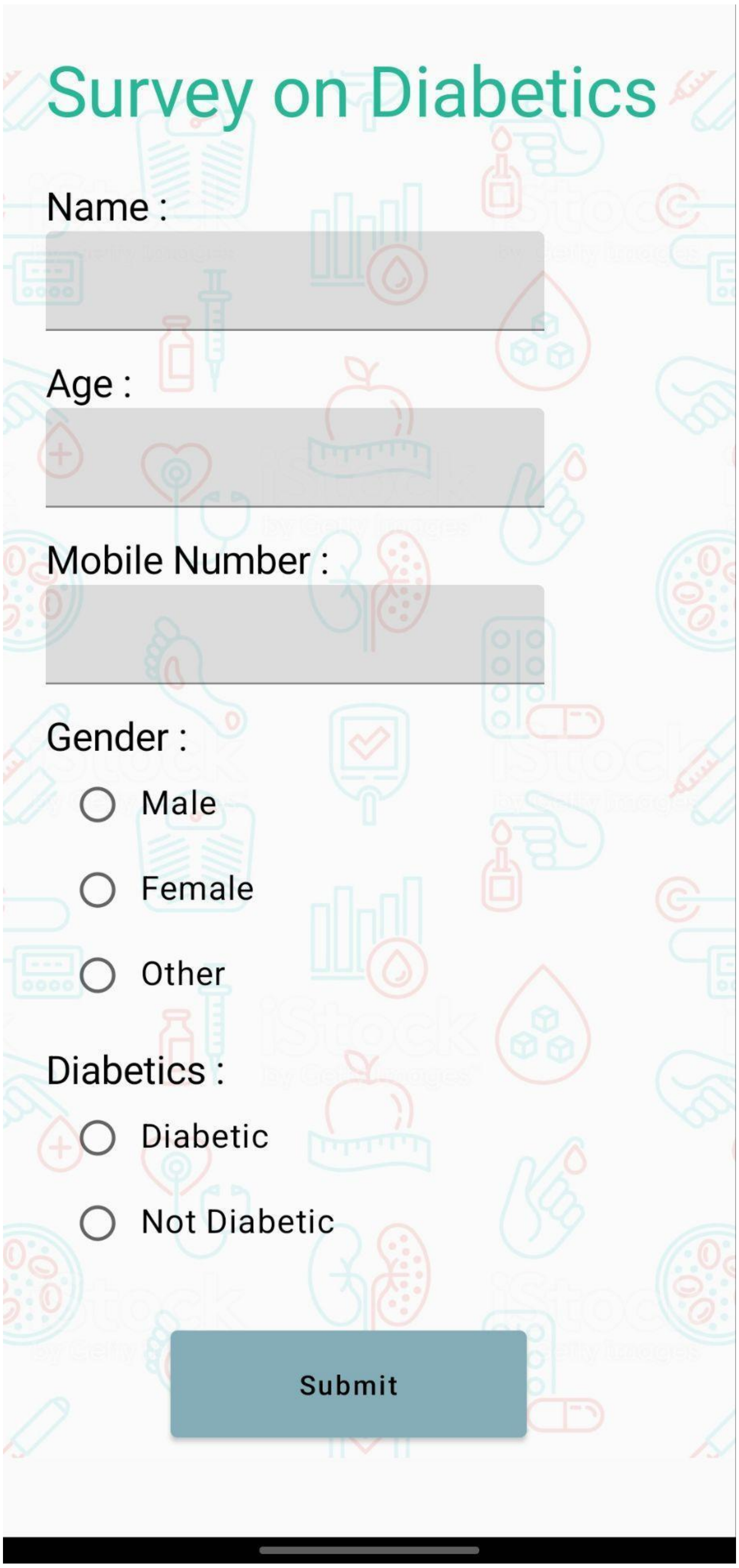
Password

Register Page :



Register
Username
Email
Password
Register
Have an account? Log in

Main Page :



Survey on Diabetics

Name :

Age :

Mobile Number :

Gender :

○ Male

○ Female

○ Other

Diabetics :

○ Diabetic

○ Not Diabetic