

Customize podcast styles

Fresh & friendly / 0p blue navigation themes Use freely a pp live-
customiz playback, podcast playac - and ortemie in dynamic themes.



A Redox Inspired Podcast App With Dynamic Themes For Android

SUBMITTED BY:

F20C834AEC202CDA8CC48C03BEFFECOE - **PUGAZHENTHI D**

F4ED75CD735D5B886D144633F2680CBC - **SUGANTHAN R**

9D3139FBC8CE2D596894A95DC1F0D1AE - **RAJASHREE D**

D1D9C2D4BFB3A8493E440D2FF25BB817 - **AMREEN R**

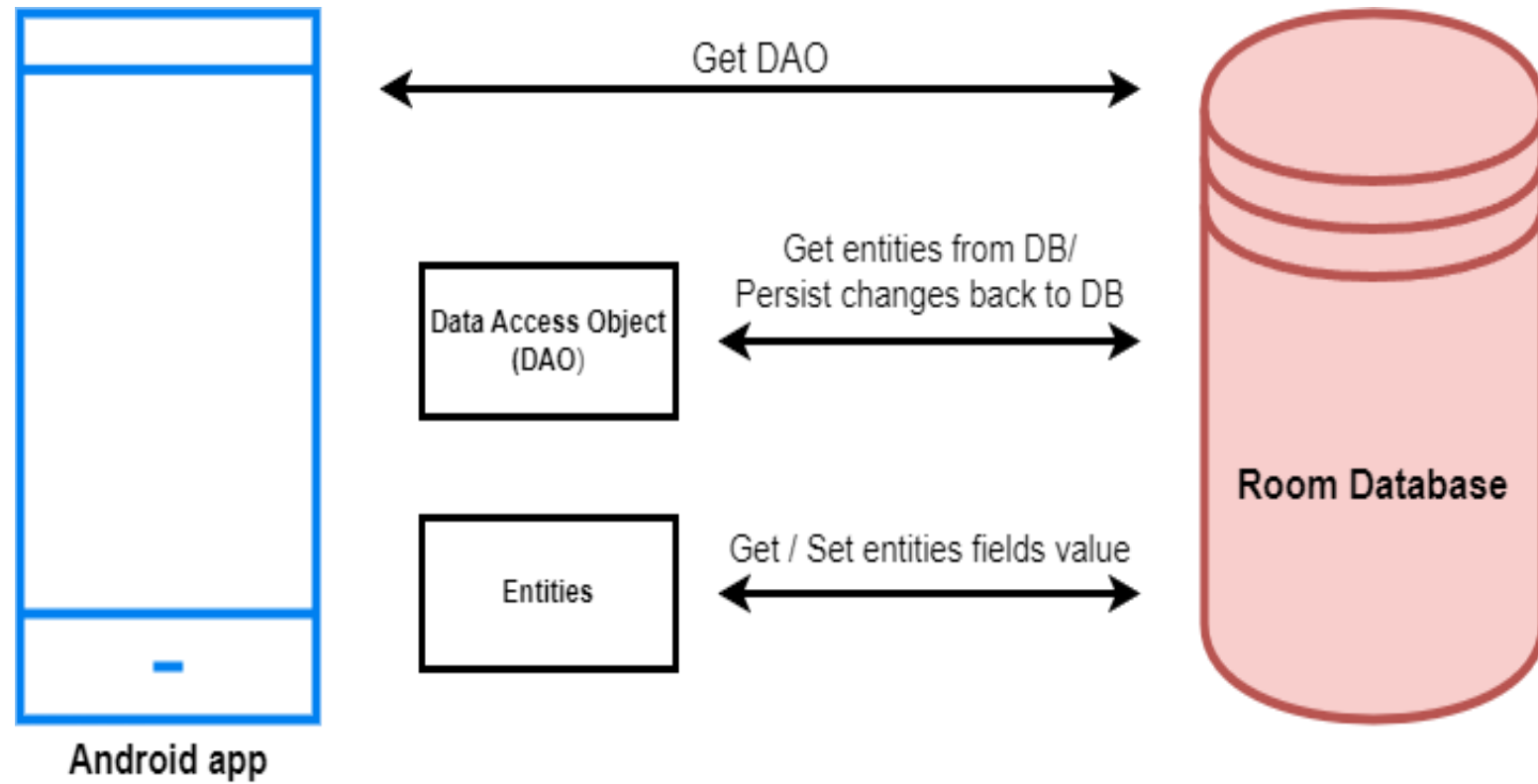
FEE8697B66F9F4A89F64C13826418D51 - **AKSHAYA S**

Podcast Plus: A Redux-Inspired Podcast App with Dynamic Themes for Android

Project Description:

- A project that demonstrates the use of Android Jetpack Compose to build a UI for a podcast player app.
- The app allows users to choose , play and pause podcasts.

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
- The app allows users to choose , play and pause podcasts.

Note:

To complete the project you need to finish up the tasks listed below:

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.Modifying AndroidManifest.xml
- 7.Running the application.

Required initial steps

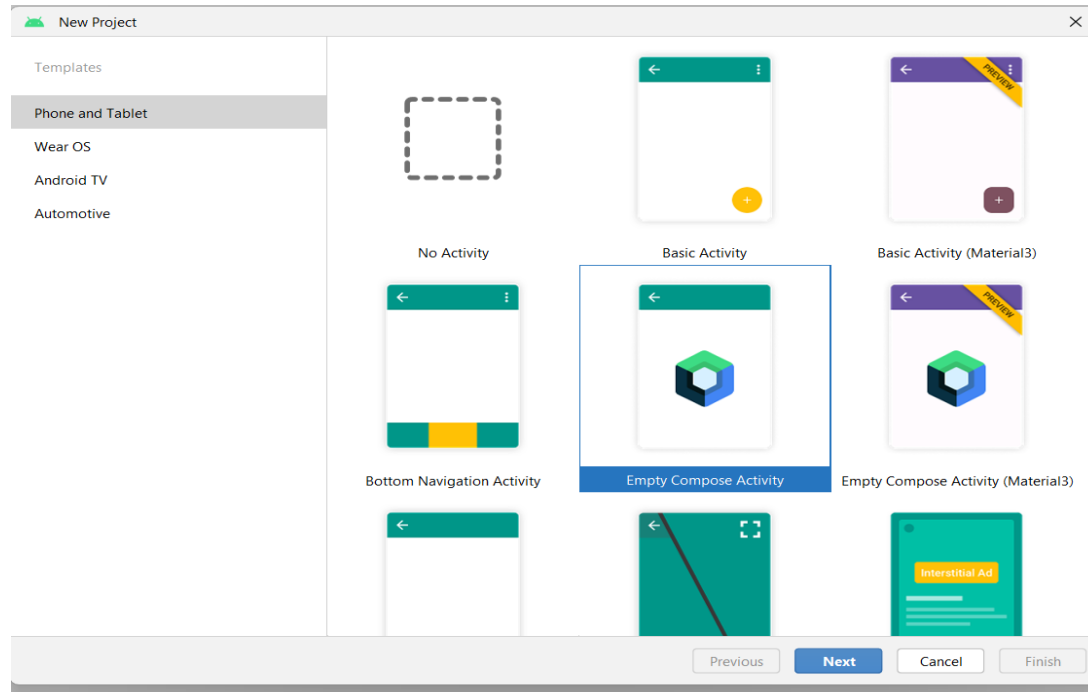
This milestone explains about required initial steps.

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

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 X

Empty Compose Activity

Create a new empty activity with Jetpack Compose

Name

Package name

Save location 

Minimum SDK

API 21: Android 5.0 (Lollipop) ▼

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)

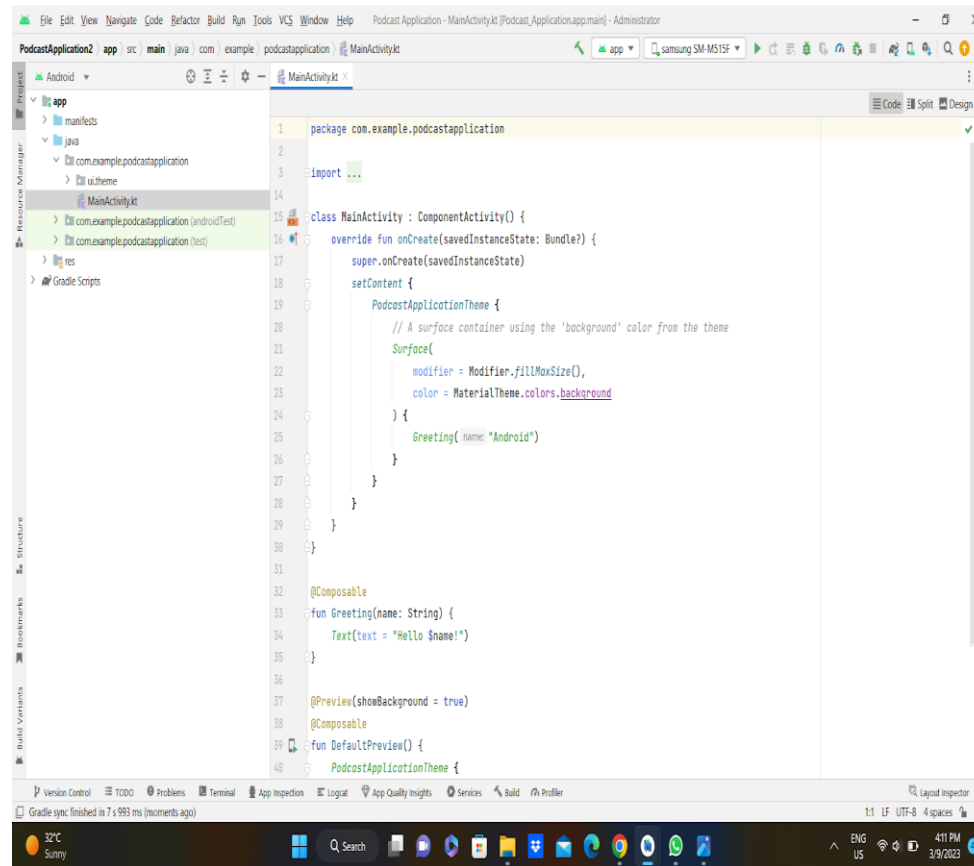
Previous


Next

Cancel

Finish

Main activity file




 New Project X

Empty Compose Activity

Create a new empty activity with Jetpack Compose

Name

Package name

Save location 

Minimum SDK

API 21: Android 5.0 (Lollipop) ▼

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)

Previous

Next

Cancel

Finish

Adding required dependencies.

This milestone explains about adding required dependencies.

Creating the database classes

Creating the database classes

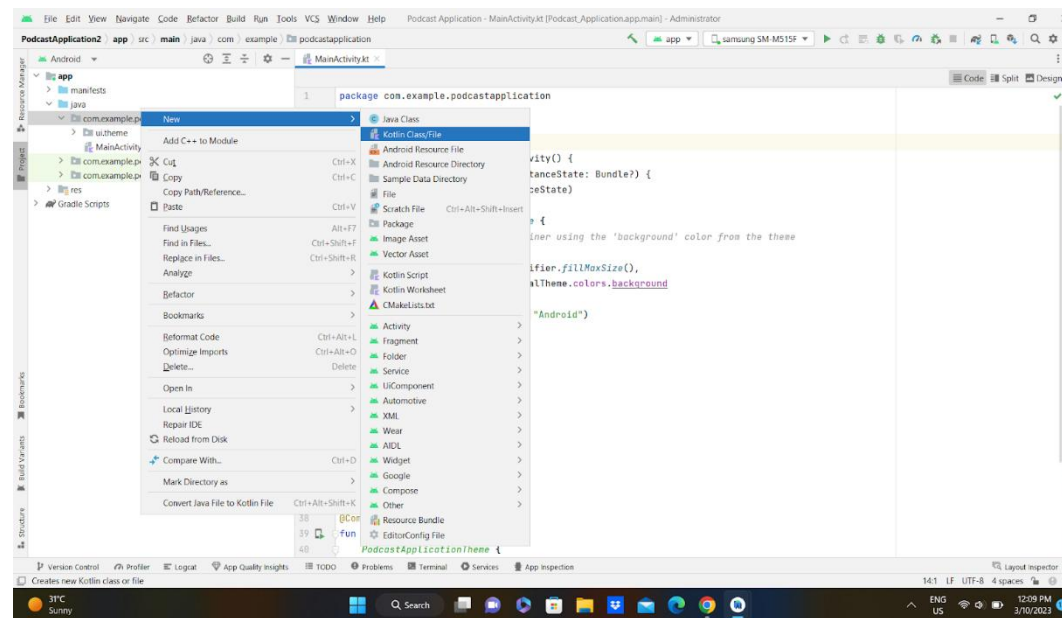
- To learn more about Database follow this link:
<https://gorillalogic.com/blog/android-room-tutorial-simplifying-how-you-work-with-app-data/>

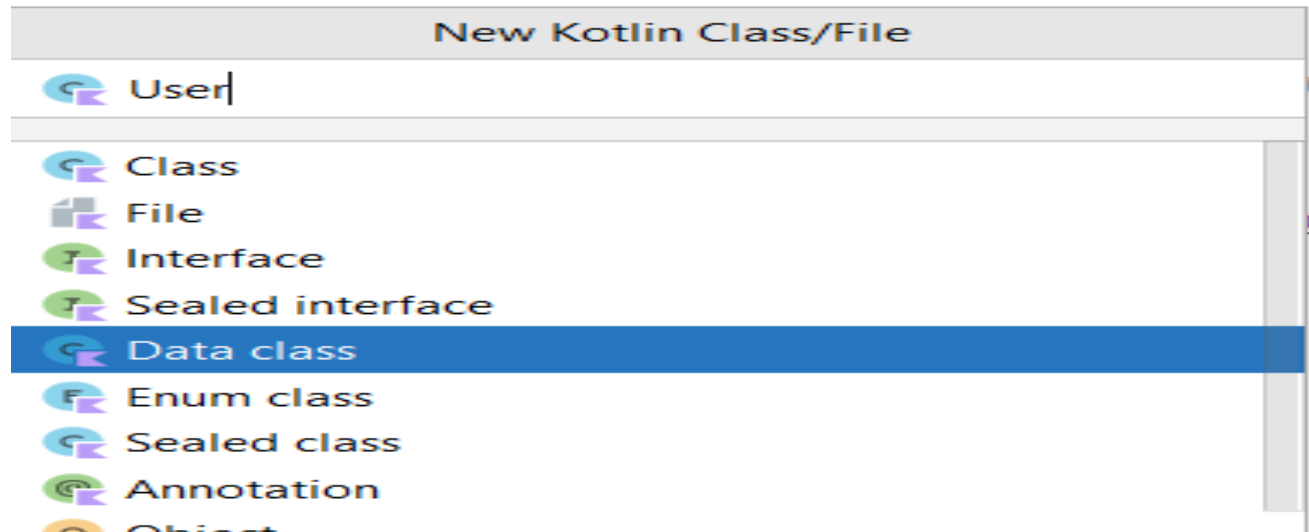
Too Much Talk—Let's Look at an Example of Android Room

All the sample code is written using [Kotlin](#) (if you don't know Kotlin or you want to learn more, I invite you to [read my article about it](#)). The first thing we need to do is to update our gradle file. It should look like this:

Create User data class

Create User data class



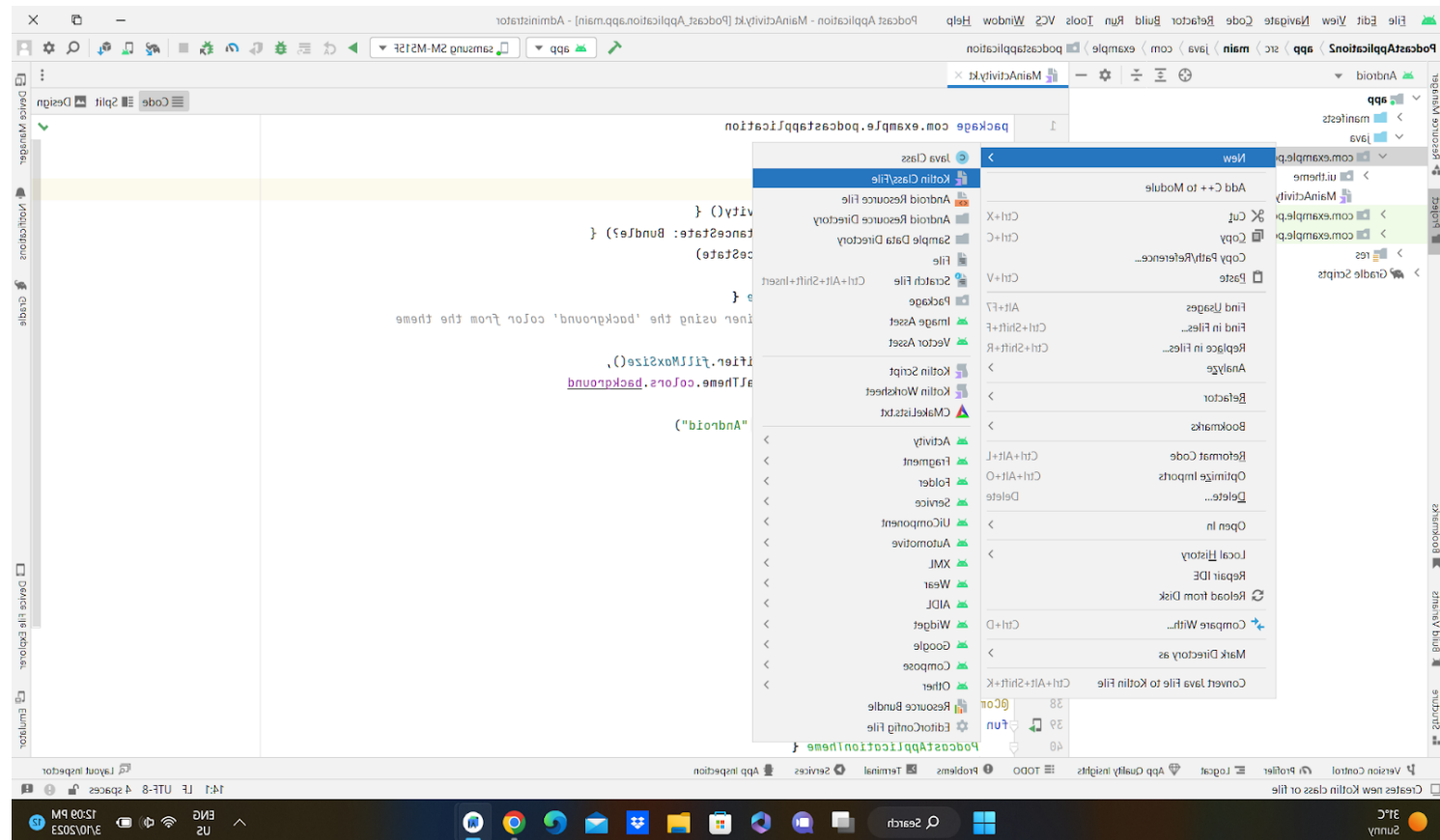


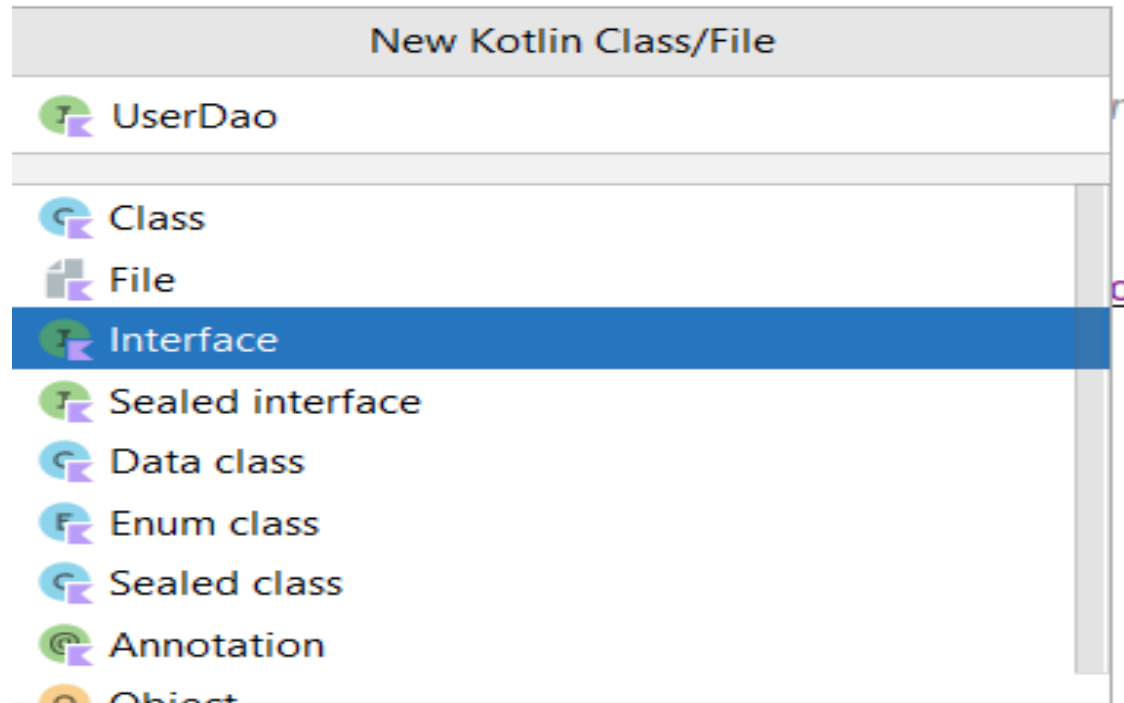
User class code:

<https://github.com/smartinternz02/PodcastPlayer/blob/master/app/src/main/java/com/example/podcastplayer/User.kt>

Create an UserDao interface

Create an UserDao interface



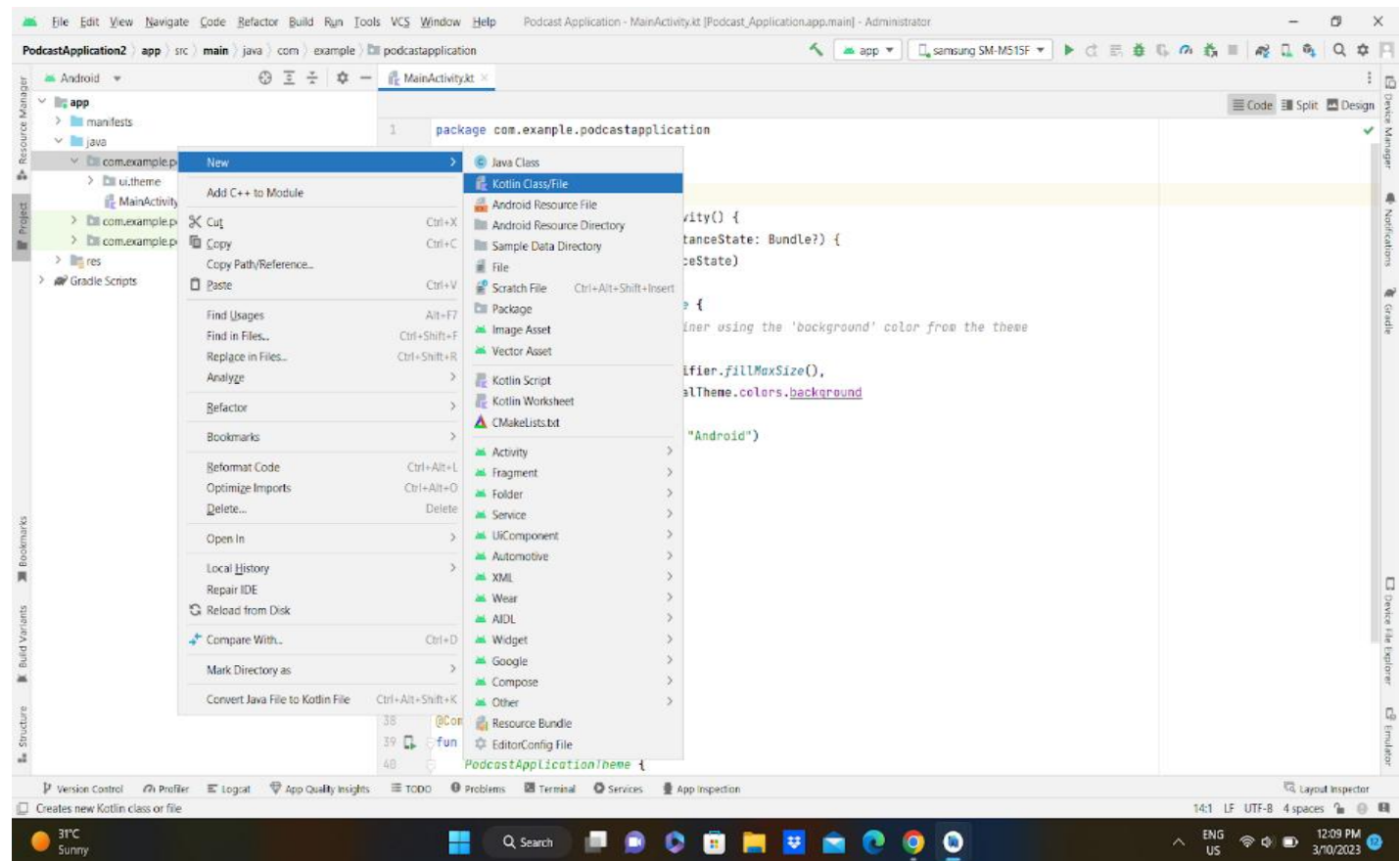


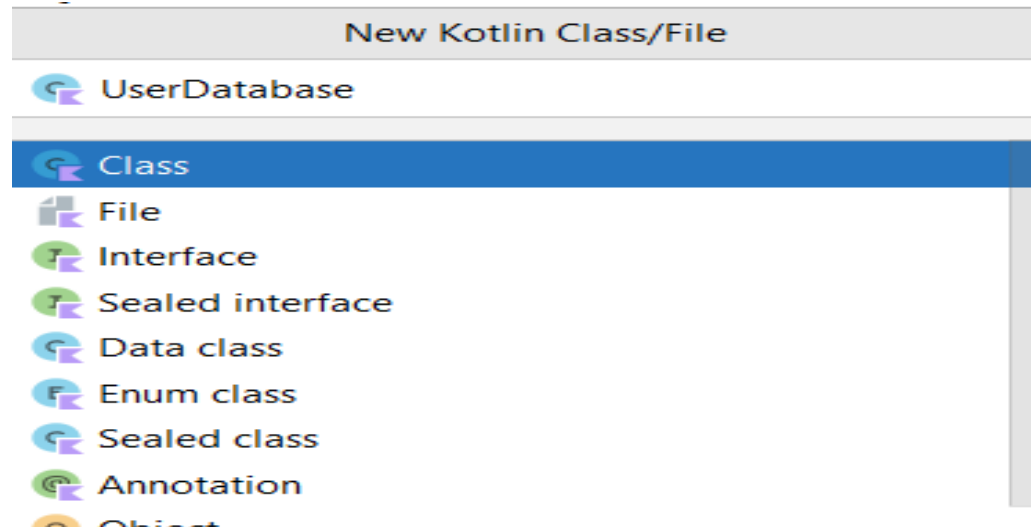
UserDao interface code:

<https://github.com/smartinternz02/PodcastPlayer/blob/master/app/src/main/java/com/example/podcastplayer/UserDao.kt>

Create an UserDatabase class

Create an UserDatabase class





UserDatabase class code :

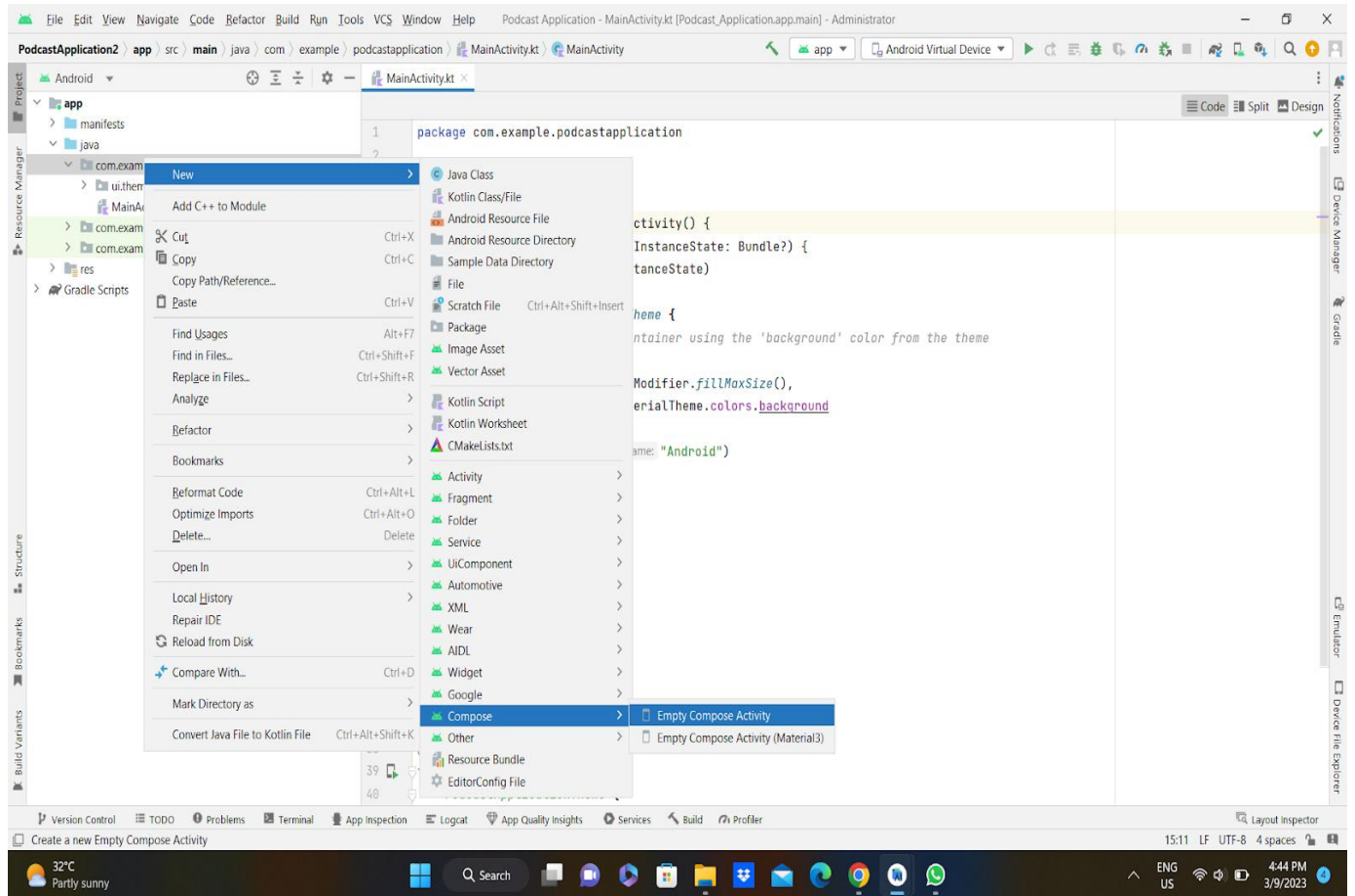
<https://github.com/smartinternz02/PodcastPlayer/blob/master/app/src/main/java/com/example/podcastplayer/UserDatabase.kt>

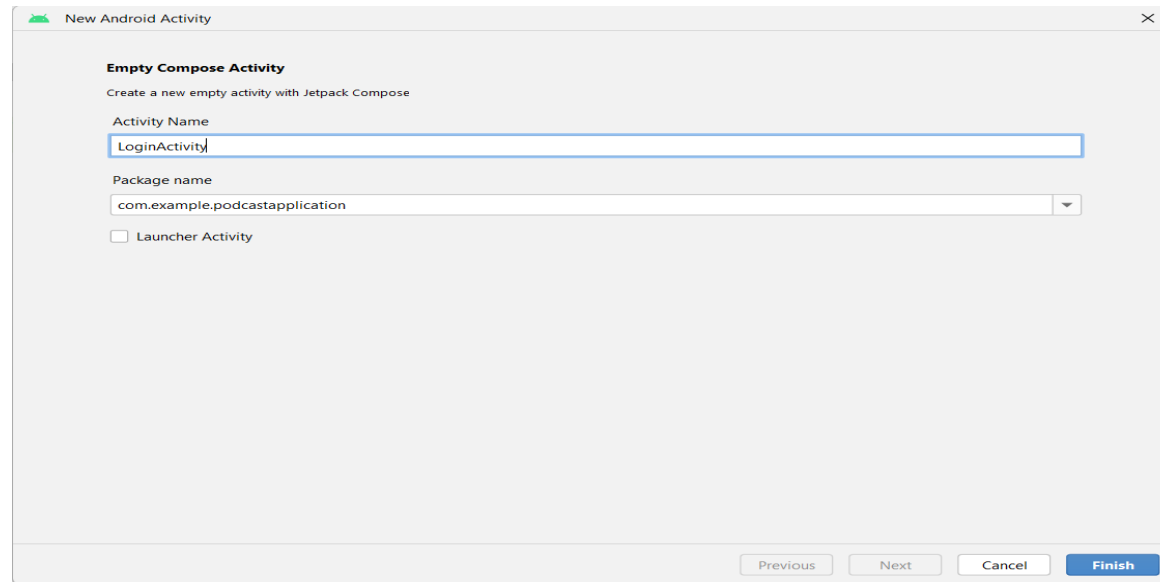
Building application UI and connecting to database

In this milestone, you will go through the Building Application UI and Connecting to database.

Creating LoginActivity.kt with database

Creating LoginActivity.kt with database





Database connection in
LoginActivity.kt

```

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

@Composable
fun LoginScreen(context: Context, databaseHelper: UserDatabaseHelper) {
    var username by remember { mutableStateOf(value: "") }
    var password by remember { mutableStateOf(value: "") }
    var error by remember { mutableStateOf(value: "") }

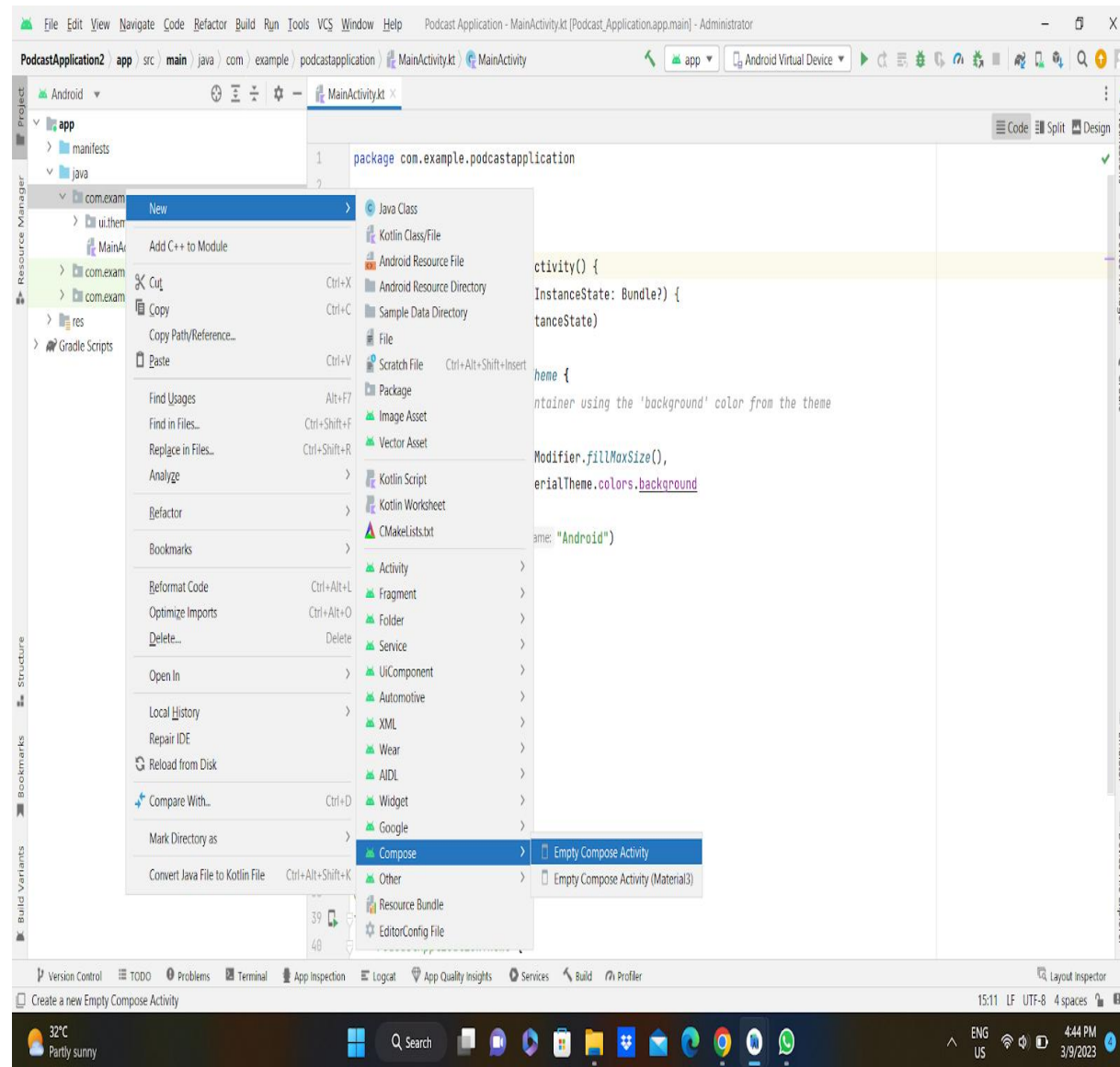
```

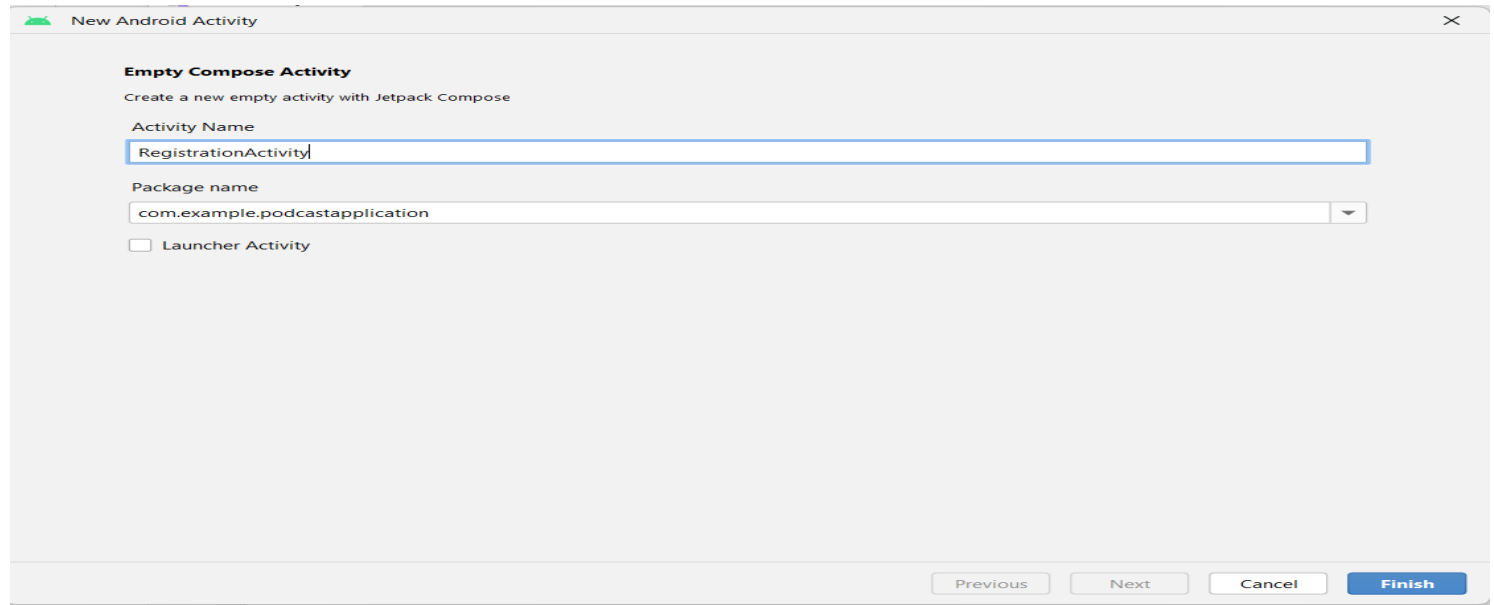
Complete code in below link

<https://github.com/smartinternz02/PodcastPlayer/blob/master/app/src/main/java/com/example/podcastplayer/LoginActivity.kt>

**Database connection in
RegistrationActivity.kt**

Creating RegistrationActivity.kt with database





Database connection in RegistrationActivity.kt

```

class RegistrationActivity : ComponentActivity() { private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(context: this)
        setContent {
            PodcastPlayerTheme {
                // 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) {
    var username by remember { mutableStateOf(value: "") }
    var password by remember { mutableStateOf(value: "") }
    var email by remember { mutableStateOf(value: "") }
    var error by remember { mutableStateOf(value: "") }
}

```

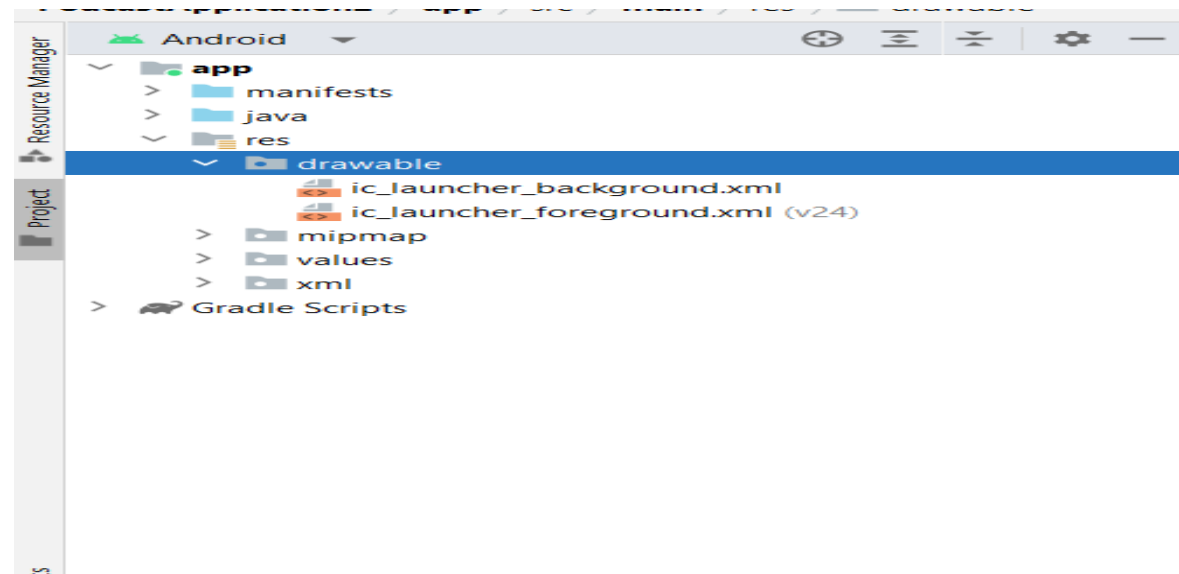
Complete code in below link:

<https://github.com/smartinternz02/PodcastPlayer/blob/master/app/src/main/java/com/example/podcastplayer/RegistrationActivity.kt>

Creating MainActivity.kt file

In MainActivity.kt file the main application is developed

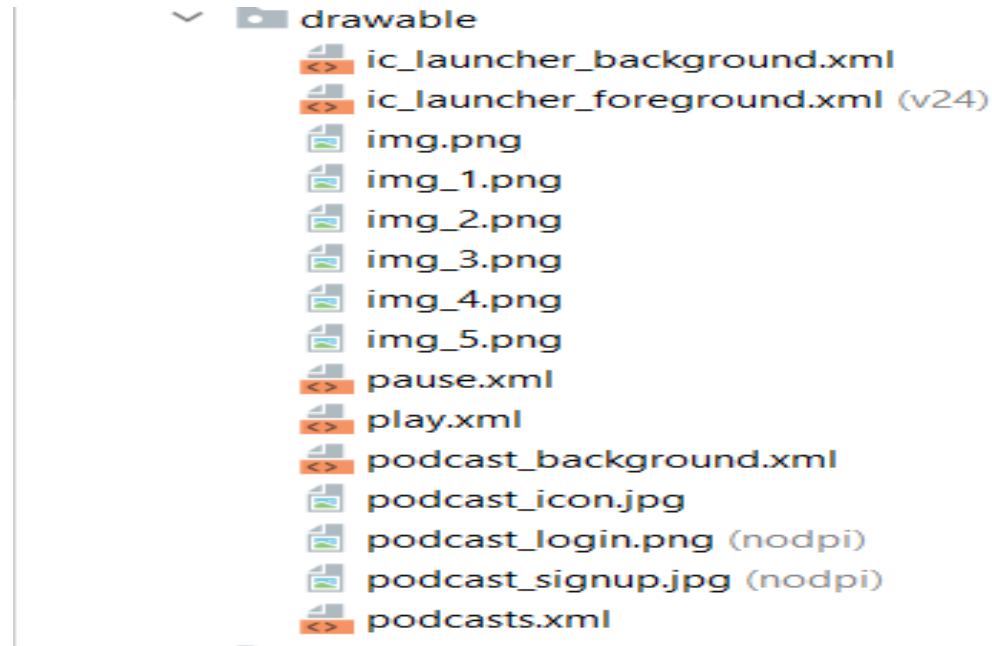
- Before creating UI we need to add some images in drawables which are in res



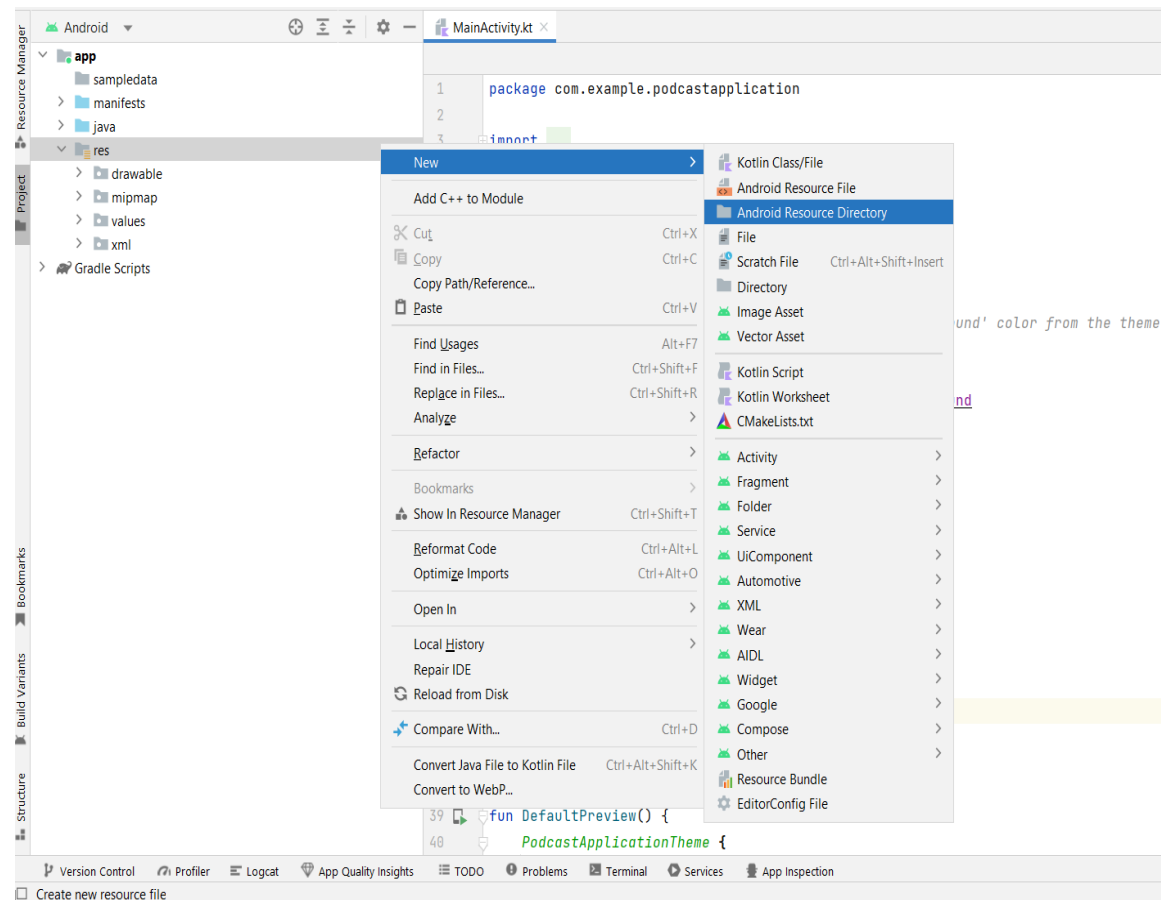
Download the required drawable from the code:

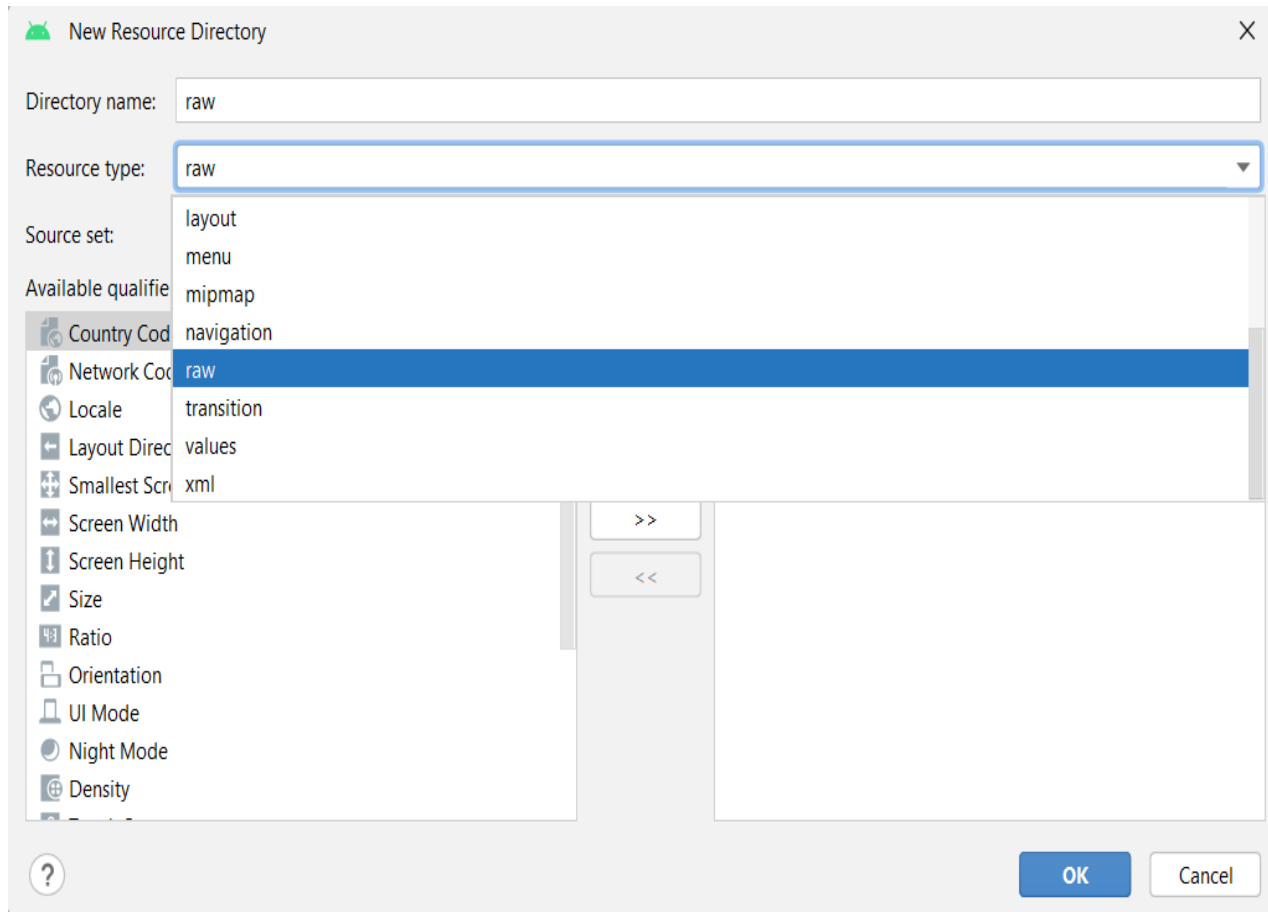
<https://github.com/smartinternz02/PodcastPlayer/tree/master/app/src/main/res/drawable>

Required drawables



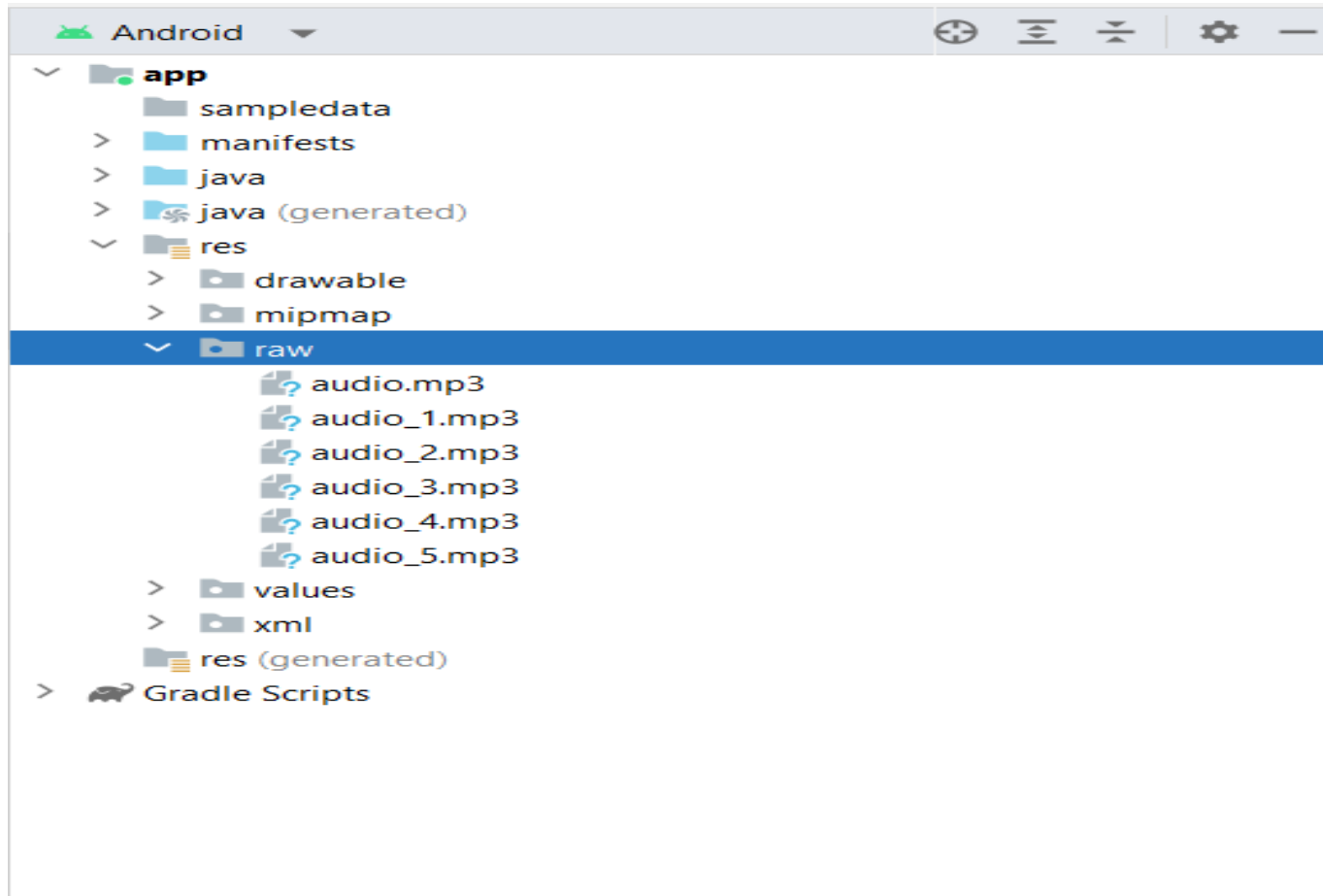
After adding drawables we need to add the required audio files





Next click OK

Add the downloaded audio into raw by copy and paste of the file



You can download the files from below link

<https://github.com/smartinternz02/PodcastPlayer/tree/master/app/src/main/res/raw>

- After add all this we need to create the UI in MainActivity.kt file

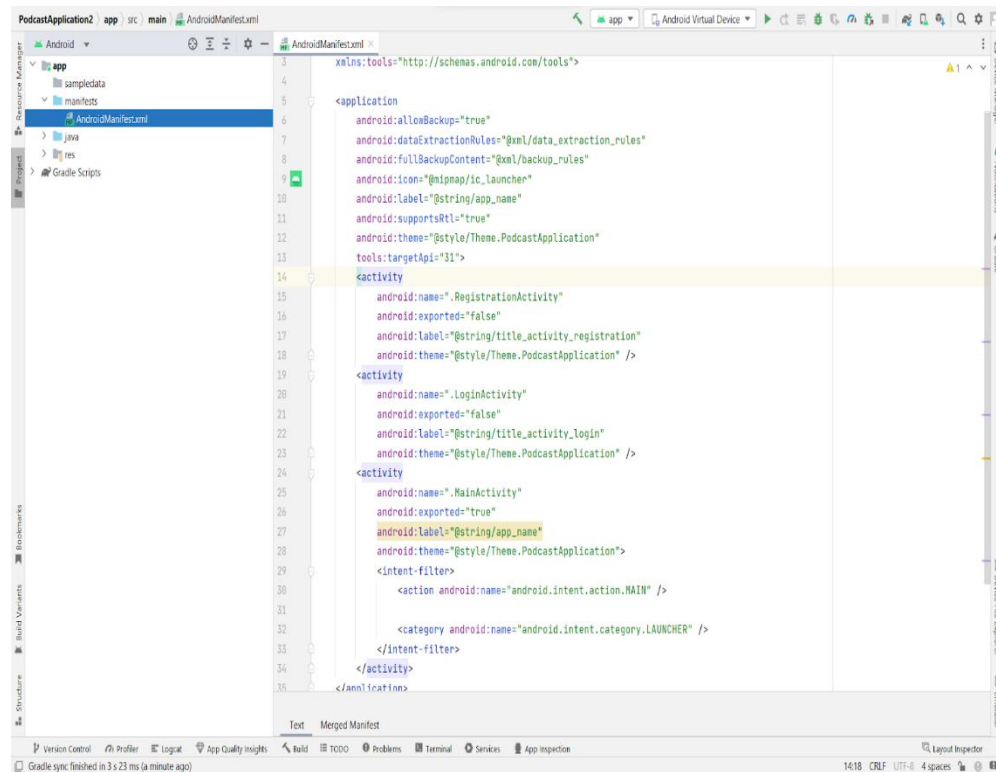
MainActivity.kt file

Complete MainActivity.kt code:

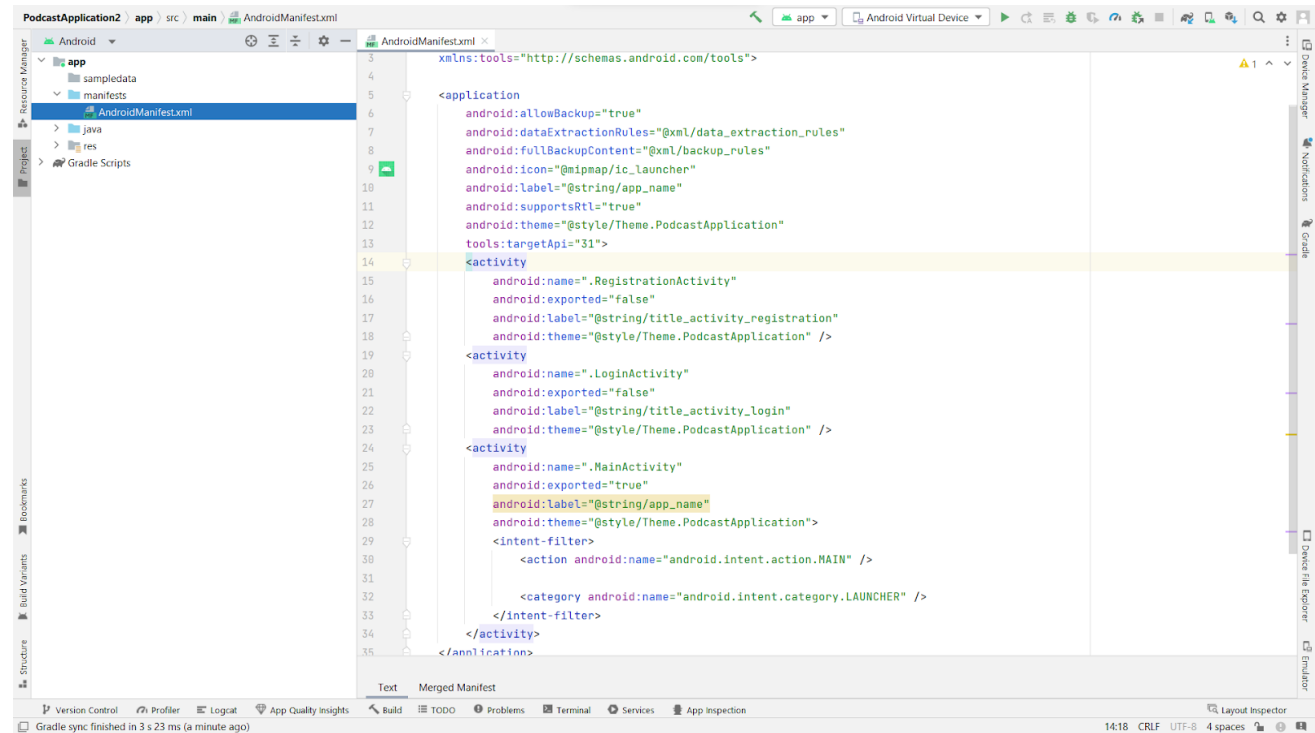
<https://github.com/smartinternz02/PodcastPlayer/blob/master/app/src/main/java/com/example/podcastplayer/MainActivity.kt>

Modifying AndroidManifest.xml

Modifying AndroidManifest.xml



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.

```
tools:targetApi="31">
<activity
    android:name=".RegistrationActivity"
    android:exported="false"
    android:label="RegistrationActivity"
    android:theme="@style/Theme.PodcastPlayer" />
<activity
    android:name=".MainActivity"
    android:exported="false"
    android:label="MainActivity"
    android:theme="@style/Theme.PodcastPlayer" />
<activity
    android:name=".LoginActivity"
    android:exported="true"
    android:label="Podcast Player"
    android:theme="@style/Theme.PodcastPlayer">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

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

Complete code is given below:

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

Running the application

This milestone, explains about how to run the application.

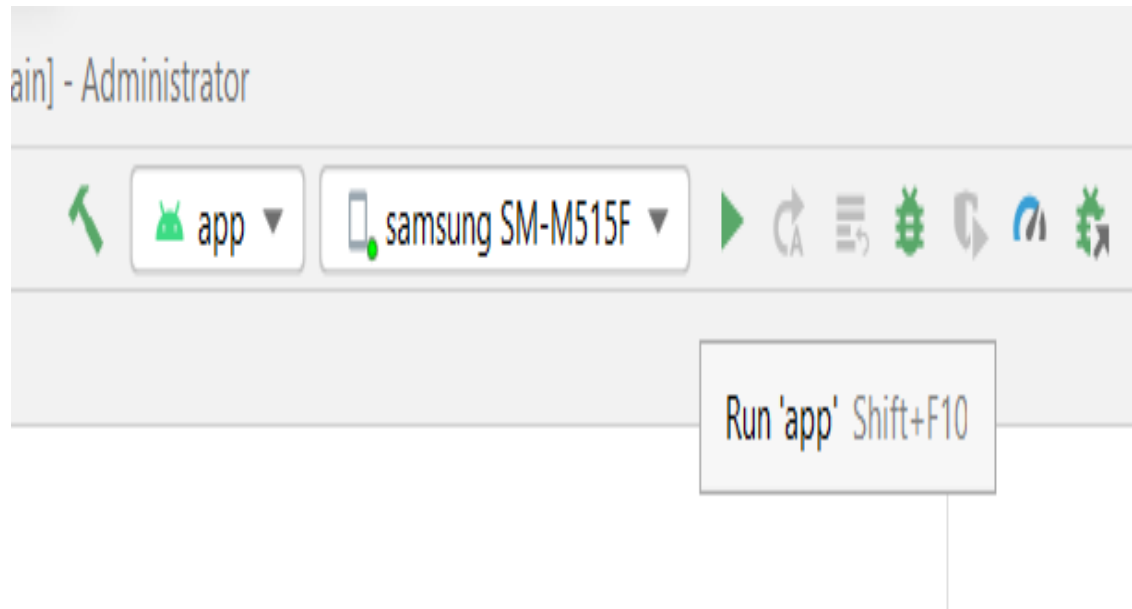
Run apps on a hardware device

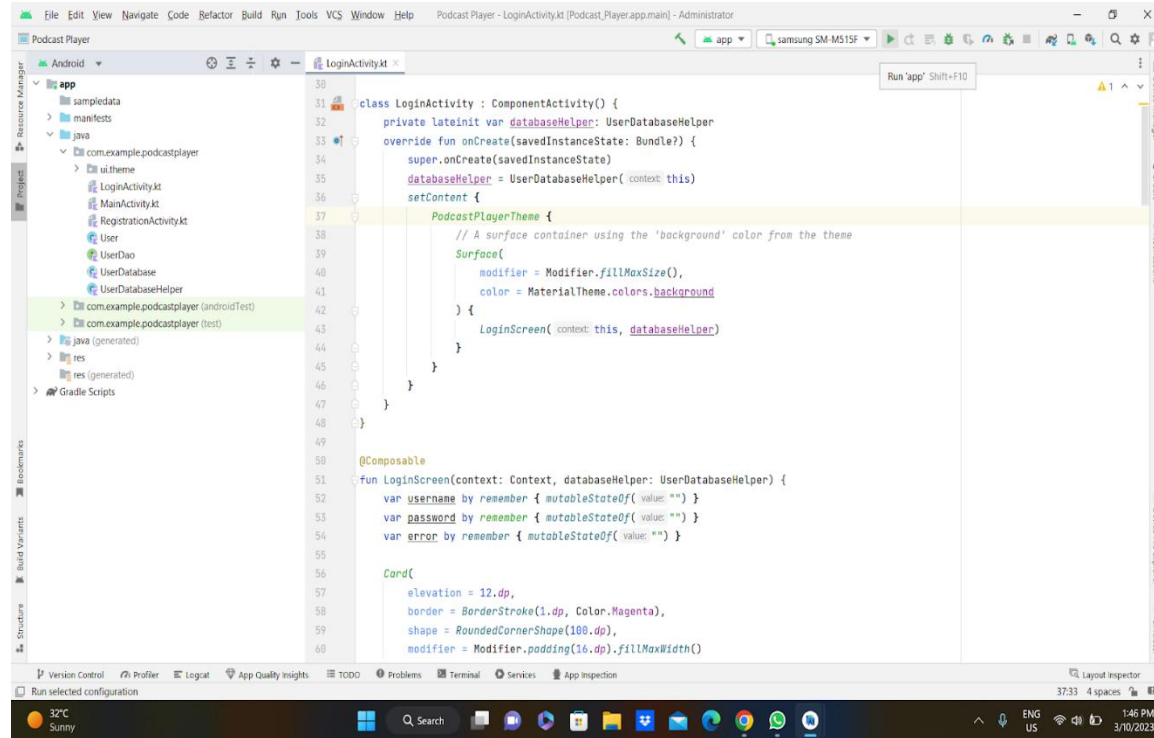
Run apps on a hardware device

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

Run the application in Mobile

Run the application in
Mobile



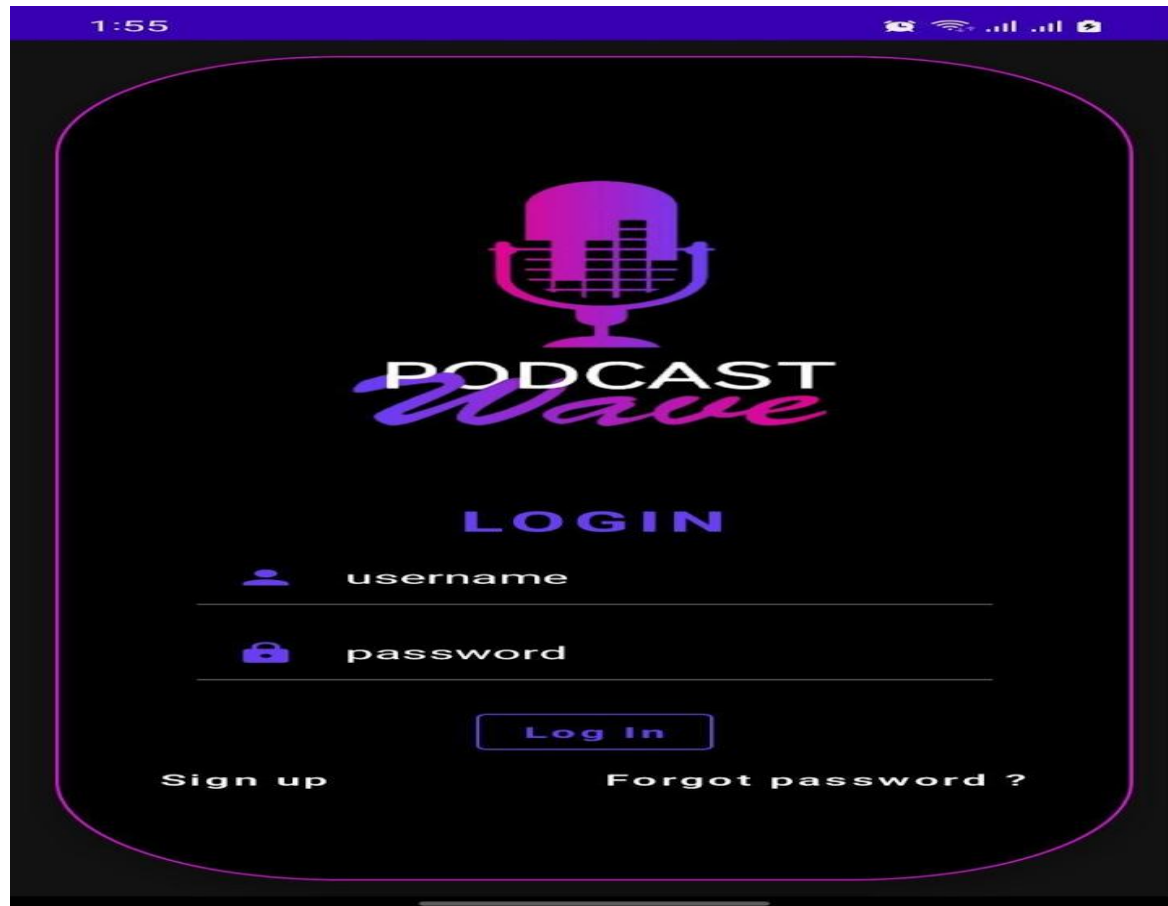


Complete Project

Link: <https://github.com/smartinternz02/PodcastPlayer>

Final Output of the Application :

Login Page



Main Podcast Page :

1:55

PODCAST



GaurGopalDas Returns To TRS - Life, Monkhood & Spirituality

▶ ||



Haunted Houses, Evil Spirits & The Paranormal Explained | Sarbajeet Mohanty

▶ ||



Kaali Mata ki kahani - Black Magic & Aghoris ft. Dr Vineet Aggarwal

▶ ||

Thank
you