Mobile App Development

Project Report : Turf Booking System

B067 - Shubh Joshi
B075 - Darrian Bharucha
B076 - Riyansh Sachdeva
B077 - Prakash Garg

Problem Statement

Develop a mobile application for booking turf time slots. The application should allow users to sign up, log in using email or Google Sign-In, select a turf venue, pick a date and time slot, and book a time for usage. It should ensure that users are authenticated, prevent double booking of slots, and manage user sessions efficiently. The app should also display a waiting room feature, where users can view other users booked for the same slot, and a confirmation page summarizing booking details.

Solution

A Turf Booking System mobile app that includes the following main features and components:

1. User Authentication: The app uses Firebase Authentication to support email-based and Google-based sign-ins. It manages login and registration through a Firebase backend.
2. Turf Booking: Users select a venue, date, and time slot to book. Firebase Realtime Database is used to store booking details, and a function checks the availability of selected slots to prevent double bookings.
3. Waiting Room: Users can view a waiting room that shows the number of other users booked in the same slot. Once a sufficient number of users are in the waiting room, the slot is confirmed as booked, and the booking status is updated.
4. Confirmation: After successful booking, users see a confirmation screen with their booking details.
5. User Interface: XML files define the UI for login, sign-up, booking, waiting room, and confirmation pages.

Google Sign-IN Implementation

1. Set Up Firebase Project and Enable Authentication

- Go to the Firebase Console.
- Select your project, or create a new one if you haven't done so.
- In your Firebase project, go to Authentication in the sidebar.
- Under the Sign-in method tab, enable Google as a sign-in provider.
- Configure any necessary settings (you may be prompted to add support email).

2. Configure Android Project in Firebase

- In the Firebase Console, click on Project Settings (gear icon next to Project Overview).
- Scroll down to Your apps and select Add app if you haven't already added your Android app.
- Register your app by providing your Android package name.
- Follow the on-screen instructions to download the google-services.json file.
- Place the downloaded google-services.json file in the app/ directory of your Android project.

3. Add Firebase SDK to Project

- Open your build.gradle files to add the Firebase SDK:

Project-level build.gradle file:

```
gradleCopy codebuildscript {
    dependencies {
        // Add this line
        classpath 'com.google.gms:google-services:4.3.15'
    }
}
```

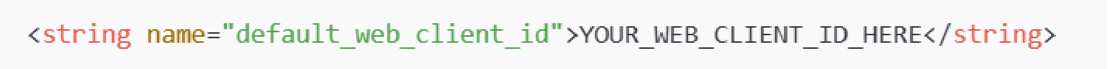
App-level build.gradle file:

```
gradleCopy codedependencies {
    // Add Firebase Authentication and Google Sign-In SDKs
    implementation 'com.google.firebase:firebase-auth:22.1.1'
    implementation 'com.google.android.gms:play-services-auth:20.5.0'
}

// Add this line at the bottom of the file
apply plugin: 'com.google.gms.google-services'
```

- Sync your project to download the required dependencies.

4. Set Up Google Sign-In Configuration

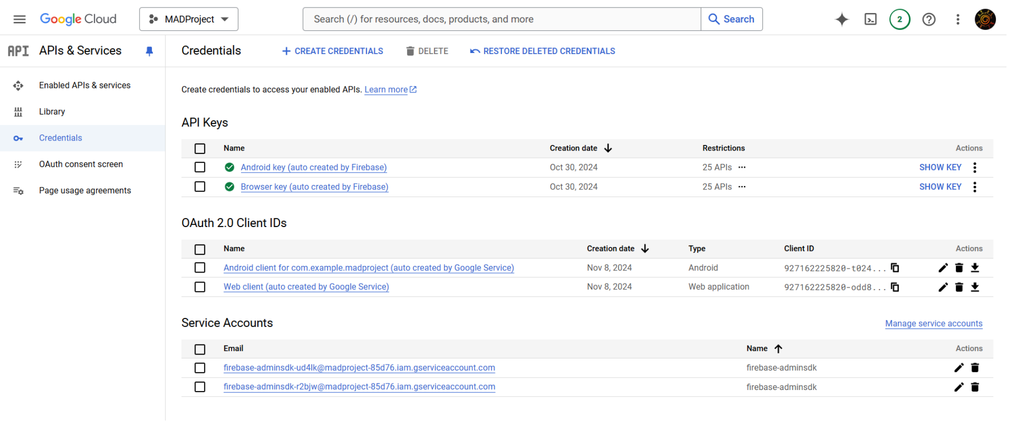
- Back in the Firebase Console, go to Project Settings and scroll down to Your apps.
- Here, you'll find the Web client ID for Google Sign-In. Copy it.
- In your Android project, open res/values/strings.xml and add the default_web_client_id:



```
<string name="default_web_client_id">YOUR_WEB_CLIENT_ID_HERE</string>
```

- Replace YOUR_WEB_CLIENT_ID_HERE with the actual Web client ID from the Firebase Console.

- Web Client ID taken from Google Cloud>APIs & Services>Credentials>OAuth 2.0 Client IDs



Credentials

Create credentials to access your enabled APIs. Learn more

API Keys

□	Name	Creation date ↓	Restrictions	Actions
□	✓ Android key (auto created by Firebase)	Oct 30, 2024	25 APIs ···	SHOW KEY ⋮
□	✓ Browser key (auto created by Firebase)	Oct 30, 2024	25 APIs ···	SHOW KEY ⋮

OAuth 2.0 Client IDs

□	Name	Creation date ↓	Type	Client ID	Actions
□	Android client for com.example.madproject (auto created by Google Service)	Nov 8, 2024	Android	927162225820-t024...	
□	Web client (auto created by Google Service)	Nov 8, 2024	Web application	927162225820-odd8...	

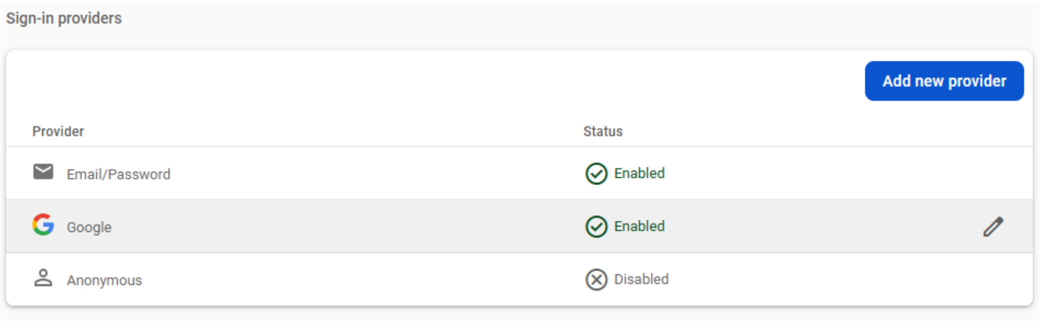
Service Accounts

Manage service accounts

□	Email	Name ↑	Actions
□	firebase-adminsdk-ud4lk@madproject-85d76.iam.gserviceaccount.com	firebase-adminsdk	
□	firebase-adminsdk-r2bjw@madproject-85d76.iam.gserviceaccount.com	firebase-adminsdk	

5. Enable Google Sign-In API in Google Cloud Console

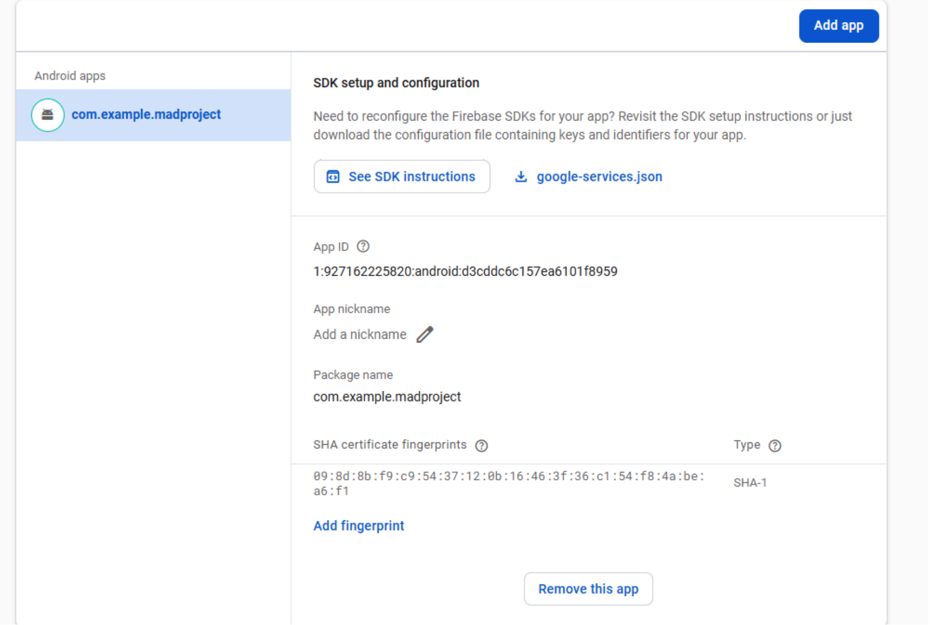
- Go to the Google Cloud Console.
- Ensure you're in the same project that's linked to your Firebase project.
- Navigate to APIs & Services > Library.
- Search for and enable the Google Identity Toolkit API and Google Sign-In API.



Sign-in providers

Provider	Status
Email/Password	✓ Enabled
Google	✓ Enabled
Anonymous	☒ Disabled

- Add SHA-1 Certificate Fingerprint



SDK setup and configuration

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.

App ID
1:927162225820:android:d3cddc6c157ea6101f8959

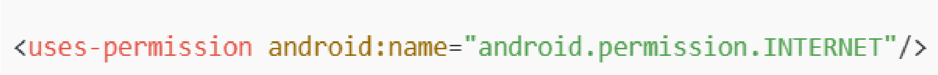
App nickname
Add a nickname

Package name
com.example.madproject

SHA certificate fingerprints	Type
09:8d:8b:f9:c9:54:37:12:0b:16:46:3f:36:c1:54:f8:4a:be:a6:f1	SHA-1

6. Update AndroidManifest.xml

- Add the Internet permission to your AndroidManifest.xml:
- Ensure your app's AndroidManifest.xml file has the necessary configuration to use Firebase.



```
<uses-permission android:name="android.permission.INTERNET"/>
```

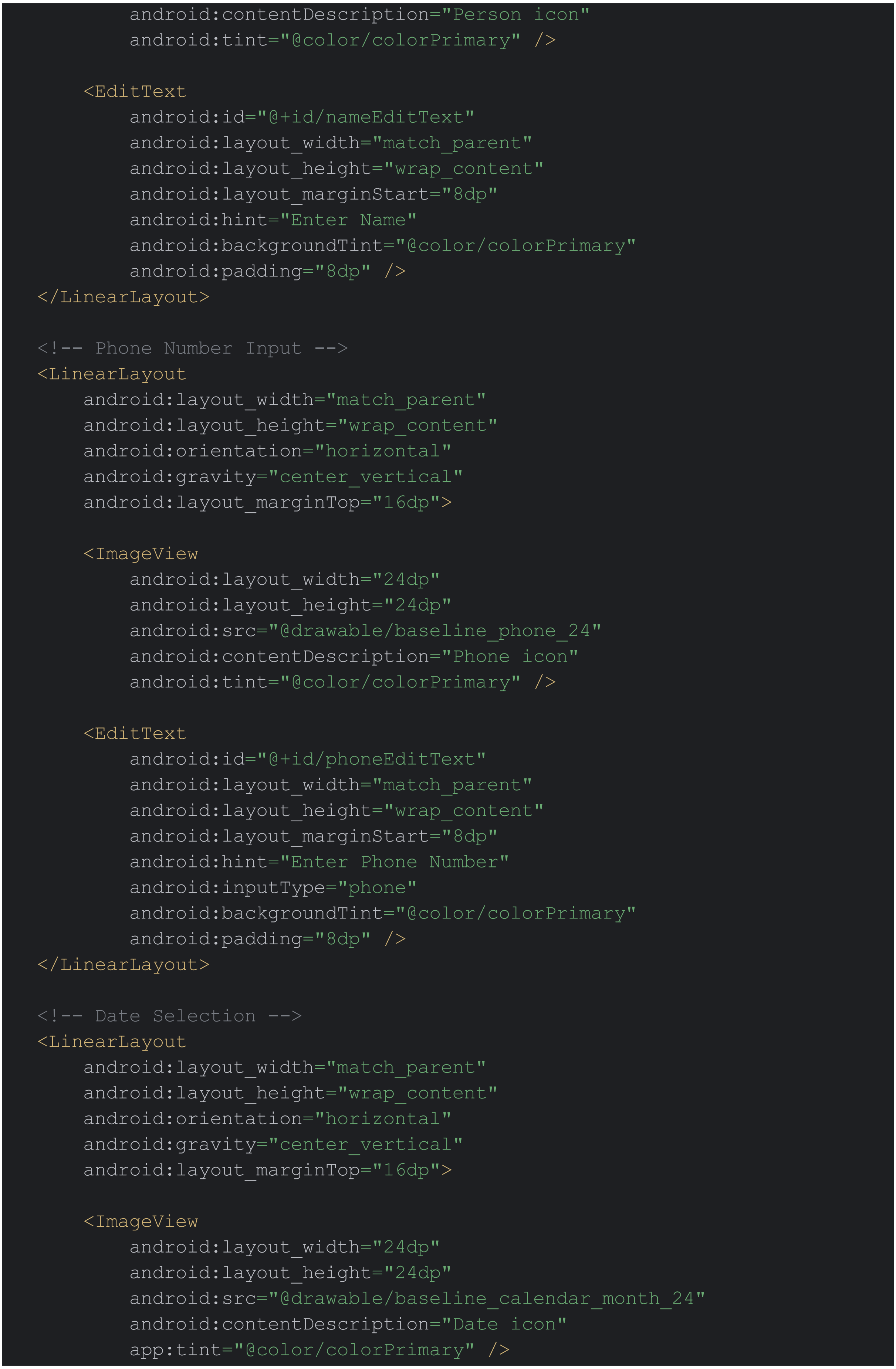
7. Run the App and Test Google Sign-In

- Rebuild and run your app on a physical device or emulator.
- Click the Sign in with Google button in your app.
- You should see the Google Sign-In dialog, allowing you to select an account and complete the sign-in process.

Code:

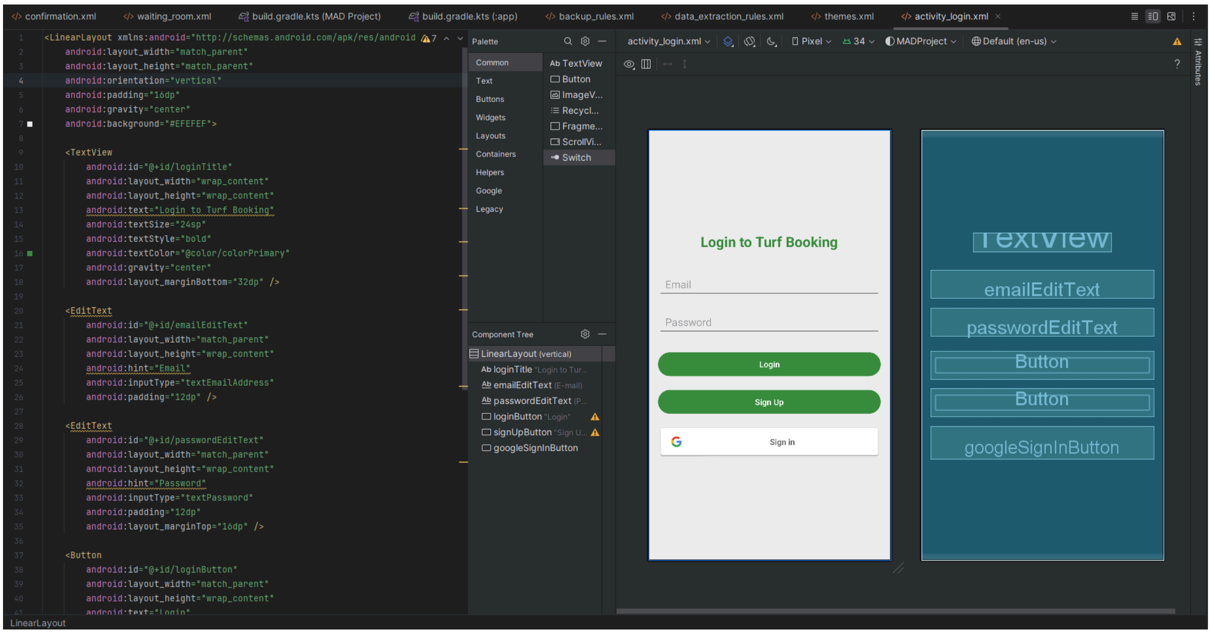
activity_login.xml

```
<!-- activity_main.xml -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   android:id="@+id/mainLayout"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@color/backgroundColor"
   android:orientation="vertical"
   android:padding="16dp">

   <TextView
       android:id="@+id/titleTextView"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:text="Turf Booking System"
       android:textSize="24sp"
       android:textStyle="bold"
       android:gravity="center"
       android:textColor="@color/colorPrimary"
       android:paddingBottom="16dp" />

   <!-- Name Input -->
   <LinearLayout
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:orientation="horizontal"
       android:gravity="center_vertical">

       <ImageView
           android:layout_width="24dp"
           android:layout_height="24dp"
           android:src="@drawable/baseline_person_24"
```



```
        android:contentDescription="Person icon"
        android:tint="@color/colorPrimary" />

    <EditText
        android:id="@+id/nameEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:hint="Enter Name"
        android:backgroundTint="@color/colorPrimary"
        android:padding="8dp" />
</LinearLayout>

<!-- Phone Number Input -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:layout_marginTop="16dp">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/baseline_phone_24"
        android:contentDescription="Phone icon"
        android:tint="@color/colorPrimary" />

    <EditText
        android:id="@+id/phoneEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:hint="Enter Phone Number"
        android:inputType="phone"
        android:backgroundTint="@color/colorPrimary"
        android:padding="8dp" />
</LinearLayout>

<!-- Date Selection -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:layout_marginTop="16dp">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/baseline_calendar_month_24"
        android:contentDescription="Date icon"
        app:tint="@color/colorPrimary" />
    <Button
        android:id="@+id/dateButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:text="Select Date"
        android:backgroundTint="@color/buttonColor"
        android:textColor="@color/buttonTextColor" />
</LinearLayout>

<!-- Slot Selection -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Select Slot"
    android:textColor="@color/textColor"
    android:textStyle="bold"
    android:layout_marginTop="16dp" />

<Spinner
    android:id="@+id/slotSpinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:entries="@array/hourly_slots"
    android:spinnerMode="dropdown"
    android:background="@drawable/spinner_background" />

<!-- Venue Selection -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Select Venue"
    android:textColor="@color/textColor"
    android:textStyle="bold"
    android:layout_marginTop="16dp" />

<Spinner
    android:id="@+id/venueSpinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:entries="@array/venues"
    android:spinnerMode="dropdown"
    android:background="@drawable/spinner_background" />

<!-- Book Slot Button -->
<Button
    android:id="@+id/bookButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:backgroundTint="@color/buttonColor"
    android:text="Book Slot"
```

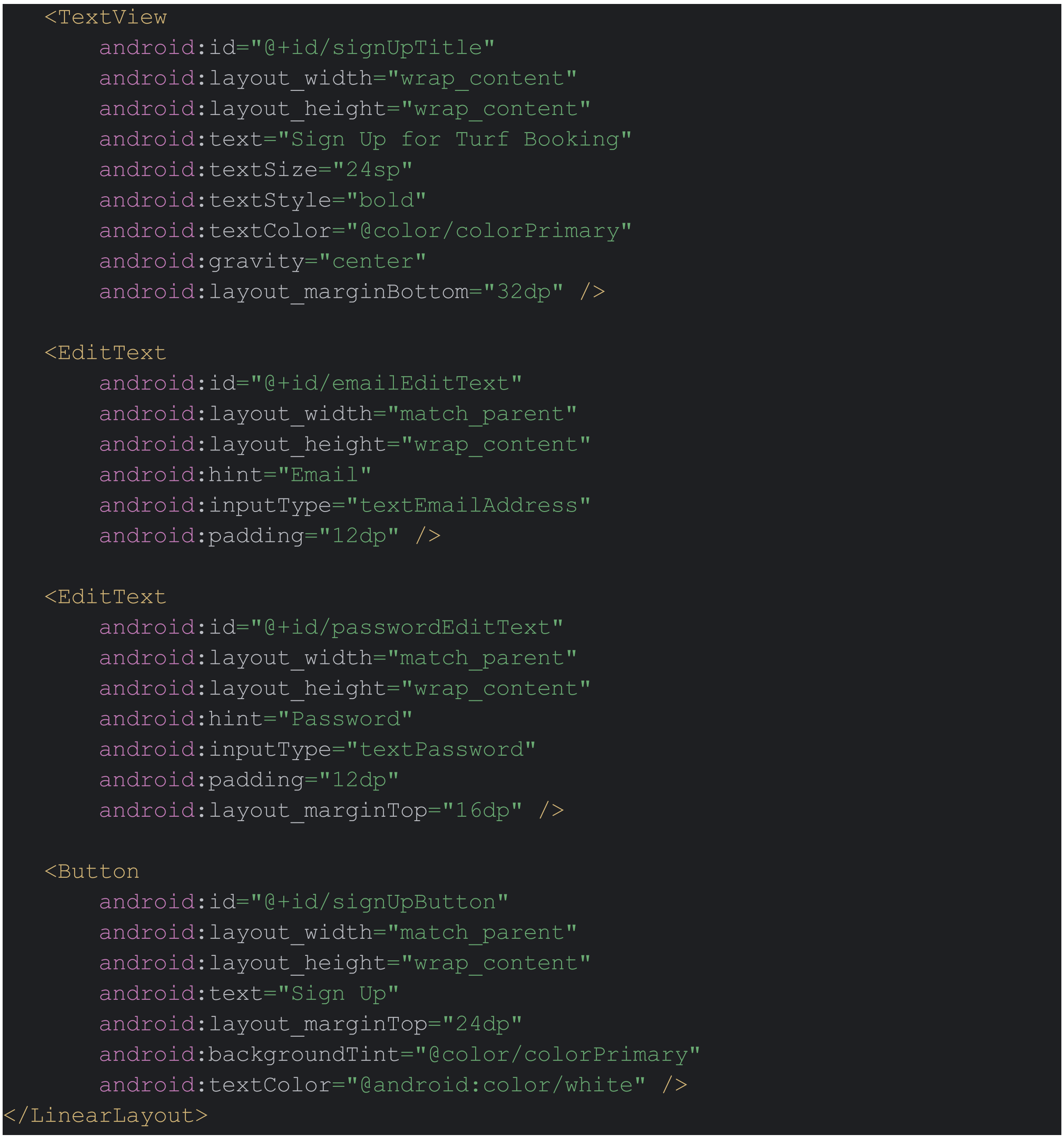
```
        android:textColor="@color/buttonTextColor"
        android:textStyle="bold" />

    <!-- Login Button -->
    <Button
        android:id="@+id/loginButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:backgroundTint="@color/buttonColor"
        android:text="Login"
        android:textColor="@color/buttonTextColor"
        android:visibility="gone" />

    <!-- Logout Button -->
    <Button
        android:id="@+id/logoutButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:backgroundTint="@color/buttonColor"
        android:text="Logout"
        android:textColor="@color/buttonTextColor"
        android:visibility="gone" />
</LinearLayout>
```



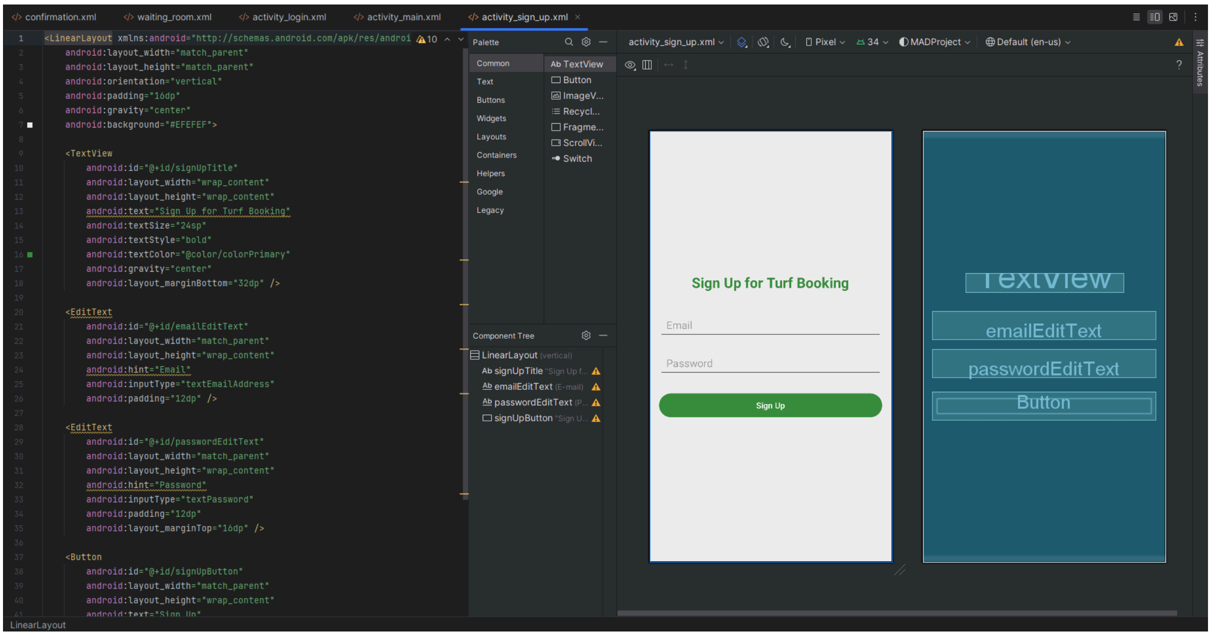
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center"
    android:background="#EFEFEF">

    <TextView
        android:id="@+id/loginTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login to Turf Booking"
        android:textSize="24sp"
        android:textStyle="bold"
        android:textColor="@color/colorPrimary"
        android:gravity="center"
        android:layout_marginBottom="32dp" />

    <EditText
        android:id="@+id/emailEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:padding="12dp" />

    <EditText
        android:id="@+id/passwordEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:padding="12dp"
        android:layout_marginTop="16dp" />

    <Button
        android:id="@+id/loginButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Login"
```

activity_sign_up.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center"
    android:background="#EFEFEF">
```



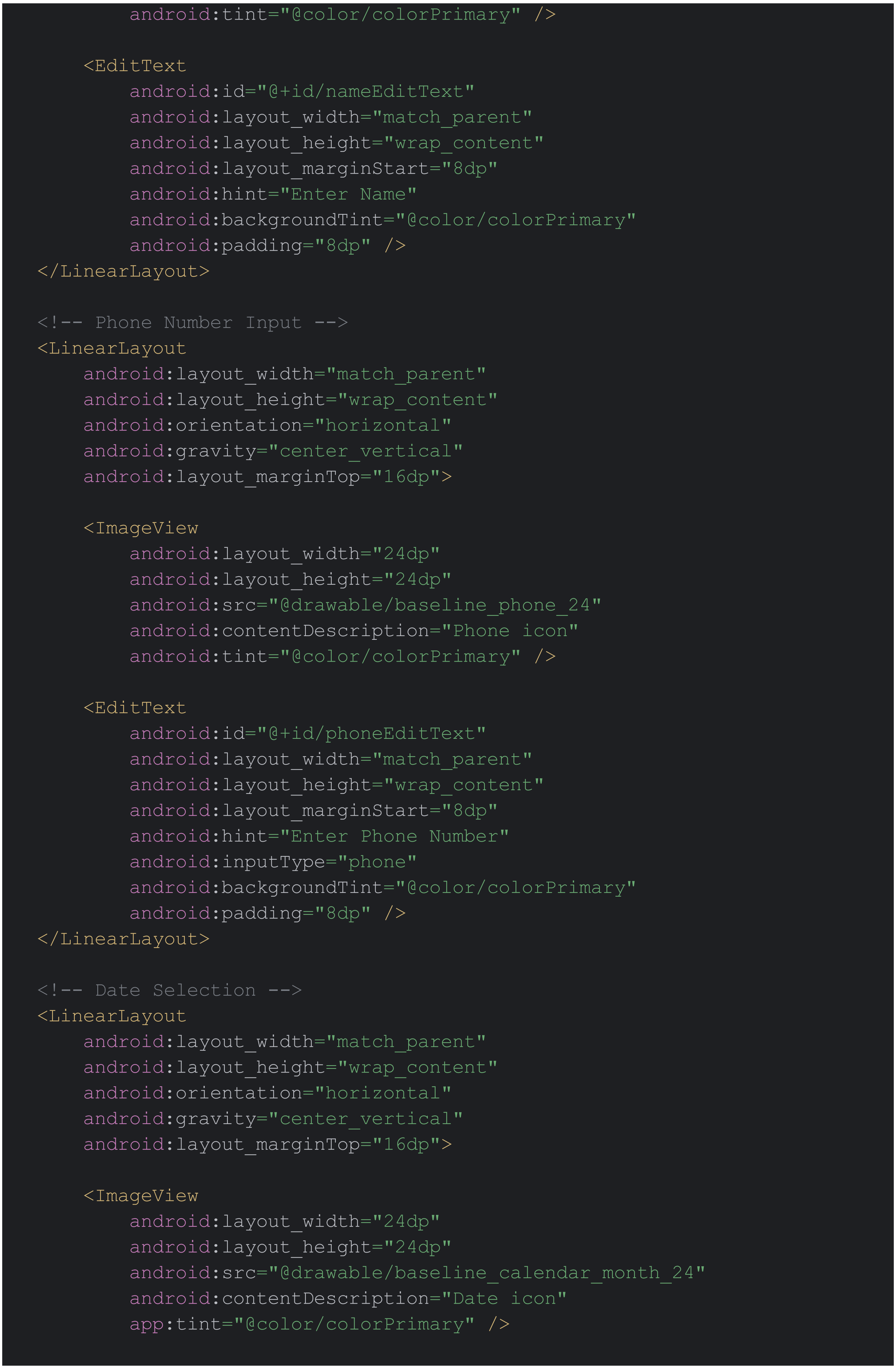
```xml
    <TextView
        android:id="@+id/signUpTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sign Up for Turf Booking"
        android:textSize="24sp"
        android:textStyle="bold"
        android:textColor="@color/colorPrimary"
        android:gravity="center"
        android:layout_marginBottom="32dp" />

    <EditText
        android:id="@+id/emailEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:padding="12dp" />

    <EditText
        android:id="@+id/passwordEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:padding="12dp"
        android:layout_marginTop="16dp" />

    <Button
        android:id="@+id/signUpButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Sign Up"
        android:layout_marginTop="24dp"
        android:backgroundTint="@color/colorPrimary"
        android:textColor="@android:color/white" />
</LinearLayout>
```



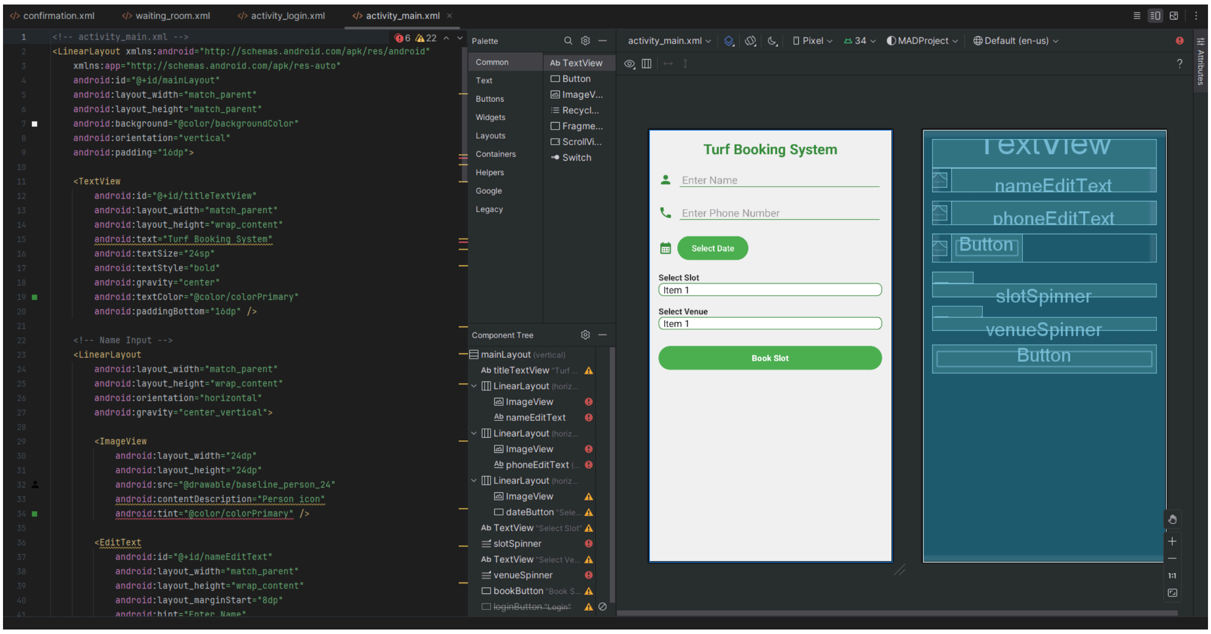
```xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/androi
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center"
    android:background="#EFEFEF">

    <TextView
        android:id="@+id/signUpTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Sign Up for Turf Booking"
        android:textSize="24sp"
        android:textStyle="bold"
        android:textColor="@color/colorPrimary"
        android:gravity="center"
        android:layout_marginBottom="32dp" />

    <EditText
        android:id="@+id/emailEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Email"
        android:inputType="textEmailAddress"
        android:padding="12dp" />

    <EditText
        android:id="@+id/passwordEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword"
        android:padding="12dp"
        android:layout_marginTop="16dp" />

    <Button
        android:id="@+id/signUpButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Sign Up"
```

activity_main.xml

```xml
<!-- activity_main.xml -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   android:id="@+id/mainLayout"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@color/backgroundColor"
   android:orientation="vertical"
   android:padding="16dp">

   <TextView
       android:id="@+id/titleTextView"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:text="Turf Booking System"
       android:textSize="24sp"
       android:textStyle="bold"
       android:gravity="center"
       android:textColor="@color/colorPrimary"
       android:paddingBottom="16dp" />

   <!-- Name Input -->
   <LinearLayout
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:orientation="horizontal"
       android:gravity="center_vertical">

       <ImageView
           android:layout_width="24dp"
           android:layout_height="24dp"
           android:src="@drawable/baseline_person_24"
           android:contentDescription="Person icon"
```



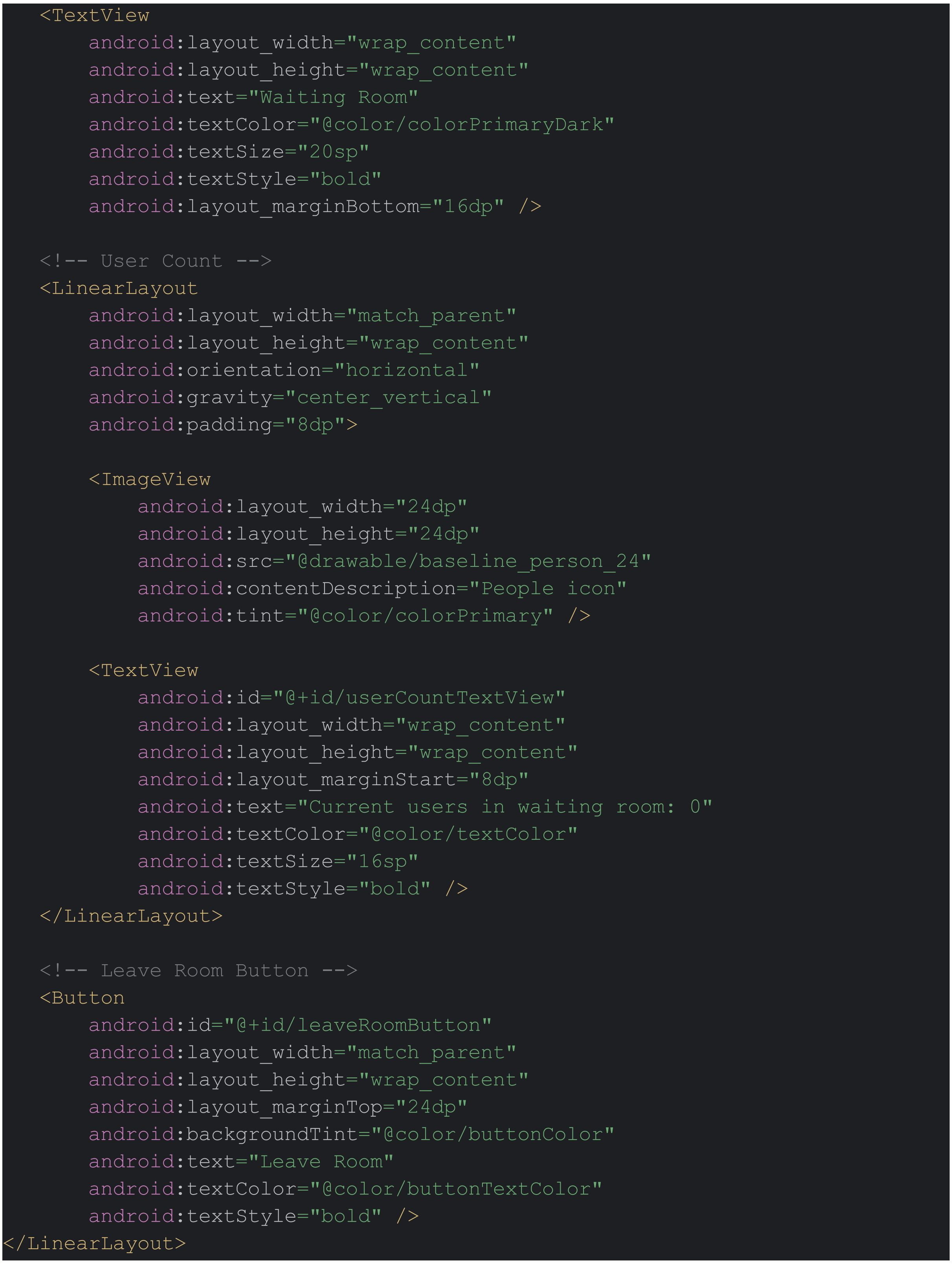
```
        android:tint="@color/colorPrimary" />

    <EditText
        android:id="@+id/nameEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:hint="Enter Name"
        android:backgroundTint="@color/colorPrimary"
        android:padding="8dp" />
</LinearLayout>

<!-- Phone Number Input -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:layout_marginTop="16dp">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/baseline_phone_24"
        android:contentDescription="Phone icon"
        android:tint="@color/colorPrimary" />

    <EditText
        android:id="@+id/phoneEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:hint="Enter Phone Number"
        android:inputType="phone"
        android:backgroundTint="@color/colorPrimary"
        android:padding="8dp" />
</LinearLayout>

<!-- Date Selection -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:gravity="center_vertical"
    android:layout_marginTop="16dp">

    <ImageView
        android:layout_width="24dp"
        android:layout_height="24dp"
        android:src="@drawable/baseline_calendar_month_24"
        android:contentDescription="Date icon"
        app:tint="@color/colorPrimary" />
    <Button
        android:id="@+id/dateButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:text="Select Date"
        android:backgroundTint="@color/buttonColor"
        android:textColor="@color/buttonTextColor" />
</LinearLayout>

<!-- Slot Selection -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Select Slot"
    android:textColor="@color/textColor"
    android:textStyle="bold"
    android:layout_marginTop="16dp" />

<Spinner
    android:id="@+id/slotSpinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:entries="@array/hourly_slots"
    android:spinnerMode="dropdown"
    android:background="@drawable/spinner_background" />

<!-- Venue Selection -->
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Select Venue"
    android:textColor="@color/textColor"
    android:textStyle="bold"
    android:layout_marginTop="16dp" />

<Spinner
    android:id="@+id/venueSpinner"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:entries="@array/venues"
    android:spinnerMode="dropdown"
    android:background="@drawable/spinner_background" />

<!-- Book Slot Button -->
<Button
    android:id="@+id/bookButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:backgroundTint="@color/buttonColor"
    android:text="Book Slot"
    android:textColor="@color/buttonTextColor"
    android:textStyle="bold" />

<!-- Login Button -->
<Button
    android:id="@+id/loginButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:backgroundTint="@color/buttonColor"
    android:text="Login"
    android:textColor="@color/buttonTextColor"
    android:visibility="gone" />

<!-- Logout Button -->
<Button
    android:id="@+id/logoutButton"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:backgroundTint="@color/buttonColor"
    android:text="Logout"
    android:textColor="@color/buttonTextColor"
    android:visibility="gone" />
</LinearLayout>
```



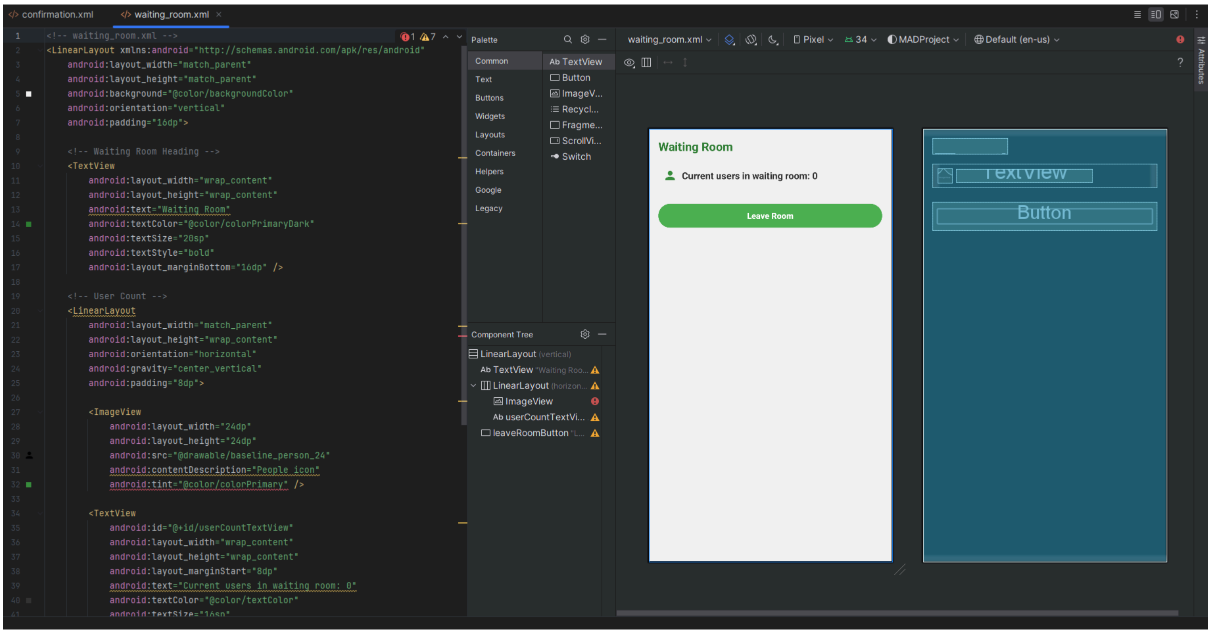
```
<!-- activity_main.xml -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/mainLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/backgroundColor"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/titleTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Turf Booking System"
        android:textSize="24sp"
        android:textStyle="bold"
        android:gravity="center"
        android:textColor="@color/colorPrimary"
        android:paddingBottom="16dp" />

    <!-- Name Input -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_vertical">

        <ImageView
            android:layout_width="24dp"
            android:layout_height="24dp"
            android:src="@drawable/baseline_person_24"
            android:contentDescription="Person icon"
            android:tint="@color/colorPrimary" />

        <EditText
            android:id="@+id/nameEditText"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:hint="Enter Name"
```

waiting_room.xml

```
<!-- waiting_room.xml -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/backgroundColor"
    android:orientation="vertical"
    android:padding="16dp">

    <!-- Waiting Room Heading -->
```



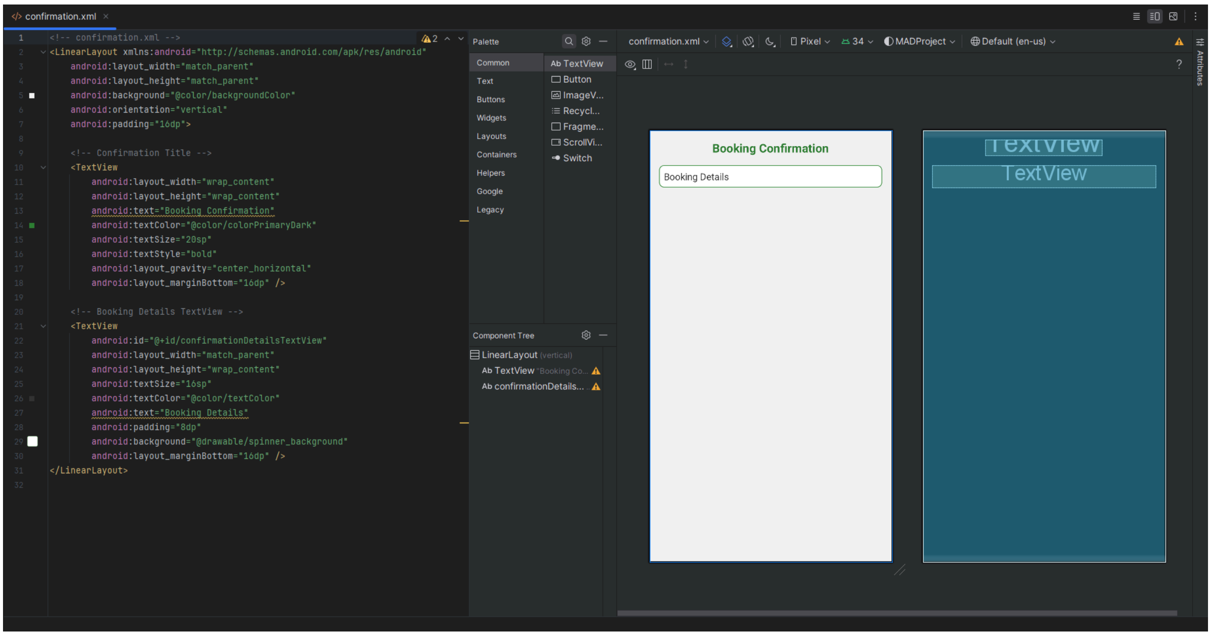
```xml
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Waiting Room"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginBottom="16dp" />

    <!-- User Count -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_vertical"
        android:padding="8dp">

        <ImageView
            android:layout_width="24dp"
            android:layout_height="24dp"
            android:src="@drawable/baseline_person_24"
            android:contentDescription="People icon"
            android:tint="@color/colorPrimary" />

        <TextView
            android:id="@+id/userCountTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:text="Current users in waiting room: 0"
            android:textColor="@color/textColor"
            android:textSize="16sp"
            android:textStyle="bold" />
    </LinearLayout>

    <!-- Leave Room Button -->
    <Button
        android:id="@+id/leaveRoomButton"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="24dp"
        android:backgroundTint="@color/buttonColor"
        android:text="Leave Room"
        android:textColor="@color/buttonTextColor"
        android:textStyle="bold" />
</LinearLayout>
```



```
<!-- waiting_room.xml -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/backgroundColor"
    android:orientation="vertical"
    android:padding="16dp">

    <!-- Waiting Room Heading -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Waiting Room"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_marginBottom="16dp" />

    <!-- User Count -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:gravity="center_vertical"
        android:padding="8dp">

        <ImageView
            android:layout_width="24dp"
            android:layout_height="24dp"
            android:src="@drawable/baseline_person_24"
            android:contentDescription="People icon"
            android:tint="@color/colorPrimary" />

        <TextView
            android:id="@+id/userCountTextView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"
            android:text="Current users in waiting room: 0"
            android:textColor="@color/textColor"
            android:textSize="16sp"
```

confirmation.xml

```
<!-- confirmation.xml -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:background="@color/backgroundColor"
   android:orientation="vertical"
   android:padding="16dp">

   <!-- Confirmation Title -->
   <TextView
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="Booking Confirmation"
       android:textColor="@color/colorPrimaryDark"
       android:textSize="20sp"
       android:textStyle="bold"
       android:layout_gravity="center_horizontal"
       android:layout_marginBottom="16dp" />

   <!-- Booking Details TextView -->
   <TextView
       android:id="@+id/confirmationDetailsTextView"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       android:textSize="16sp"
       android:textColor="@color/textColor"
       android:text="Booking Details"
       android:padding="8dp"
       android:background="@drawable/spinner_background"
       android:layout_marginBottom="16dp" />
</LinearLayout>
```



```xml
<!-- confirmation.xml -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/backgroundColor"
    android:orientation="vertical"
    android:padding="16dp">

    <!-- Confirmation Title -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Booking Confirmation"
        android:textColor="@color/colorPrimaryDark"
        android:textSize="20sp"
        android:textStyle="bold"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="16dp" />

    <!-- Booking Details TextView -->
    <TextView
        android:id="@+id/confirmationDetailsTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="16sp"
        android:textColor="@color/textColor"
        android:text="Booking Details"
        android:padding="8dp"
        android:background="@drawable/spinner_background"
        android:layout_marginBottom="16dp" />
</LinearLayout>
```

strings.xml

```xml
<!-- strings.xml -->
<resources>
   <string name="app_name">MAD Project</string>
   <string name="select_date">Select Date</string>
   <string name="book_slot">Book Slot</string>
   <string name="enter_name">Enter Name</string>
   <string name="enter_phone">Enter Phone Number</string>
   <string name="leave_room">Leave Room</string>
   <string name="slot_booked">Slot booked!</string>
   <string name="slot_unavailable">This slot is already booked.</string>
   <string name="waiting_room">Waiting Room</string>
   <string
name="default_web_client_id">927162225820-odd8o0d8p1juiqffhrtjgal64dfnh97k.apps.goo
gleusercontent.com</string>



   <!-- Define venues -->
   <string-array name="venues">
       <item>Venue 1</item>
       <item>Venue 2</item>
   </string-array>

   <!-- Define hourly slots -->
   <string-array name="hourly_slots">
       <item>9:00 AM - 10:00 AM</item>
       <item>10:00 AM - 11:00 AM</item>
       <item>11:00 AM - 12:00 PM</item>
       <item>12:00 PM - 1:00 PM</item>
       <!-- Add more slots as needed -->
   </string-array>
</resources>
```

Java Code

LoginActivity.java

```
package com.example.madproject;

import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatDelegate;

import com.google.android.gms.auth.api.signin.GoogleSignIn;
import com.google.android.gms.auth.api.signin.GoogleSignInAccount;
import com.google.android.gms.auth.api.signin.GoogleSignInClient;
import com.google.android.gms.auth.api.signin.GoogleSignInOptions;
import com.google.android.gms.common.api.ApiException;
import com.google.android.gms.common.SignInButton;
import com.google.firebase.auth.AuthCredential;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.GoogleAuthProvider;

public class LoginActivity extends AppCompatActivity {
   private static final int RC_SIGN_IN = 9001;
   private static final String TAG = "LoginActivity";
   private EditText emailEditText, passwordEditText;
   private Button loginButton, signUpButton;
   private SignInButton googleSignInButton;
   private FirebaseAuth mAuth;
   private GoogleSignInClient googleSignInClient;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_NO);
       setContentView(R.layout.activity_login);
```

```
        // Initialize Firebase Auth
        mAuth = FirebaseAuth.getInstance();

        // Find UI elements
        emailEditText = findViewById(R.id.emailEditText);
        passwordEditText = findViewById(R.id.passwordEditText);
        loginButton = findViewById(R.id.loginButton);
        signUpButton = findViewById(R.id.signUpButton);
        googleSignInButton = findViewById(R.id.googleSignInButton);

        // Configure Google Sign-In options
        GoogleSignInOptions gso = new
GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
                .requestIdToken(getString(R.string.default_web_client_id))  //
Obtain from Firebase Console
                .requestEmail()
                .build();

        googleSignInClient = GoogleSignIn.getClient(this, gso);

        // Set Google Sign-In button size and color
        googleSignInButton.setSize(SignInButton.SIZE_WIDE);
        googleSignInButton.setColorScheme(SignInButton.COLOR_DARK);

        // Set up Google Sign-In button click listener
        googleSignInButton.setOnClickListener(view -> {
            signOutAndSignInWithGoogle();  // Sign out and re-initiate Google
Sign-In
        });

        loginButton.setOnClickListener(view -> {
            String email = emailEditText.getText().toString();
            String password = passwordEditText.getText().toString();

            if (!email.isEmpty() && !password.isEmpty()) {
                signInUser(email, password);
            } else {
                Toast.makeText(this, "Please enter email and password",
Toast.LENGTH_SHORT).show();
            }
        });

        signUpButton.setOnClickListener(view -> {
            startActivity(new Intent(LoginActivity.this, SignUpActivity.class));
        });
    }

    // Method to sign out current Google account and re-initiate Google sign-in
    private void signOutAndSignInWithGoogle() {
        googleSignInClient.signOut().addOnCompleteListener(this, task -> {
            signInWithGoogle();
        });
    }

    private void signInWithGoogle() {
        Intent signInIntent = googleSignInClient.getSignInIntent();
        startActivityForResult(signInIntent, RC_SIGN_IN);
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);

        if (requestCode == RC_SIGN_IN) {
            Log.d(TAG, "Google Sign-In intent result received");
            try {
                GoogleSignInAccount account =
GoogleSignIn.getSignedInAccountFromIntent(data).getResult(ApiException.class);
                if (account != null) {
                    Log.d(TAG, "Google Sign-In successful, authenticating with
Firebase");
                    firebaseAuthWithGoogle(account);
                }
            } catch (ApiException e) {
                Log.w(TAG, "Google sign-in failed", e);
            }
        }
    }

    private void firebaseAuthWithGoogle(GoogleSignInAccount acct) {
        Log.d(TAG, "firebaseAuthWithGoogle:" + acct.getId());

        AuthCredential credential =
GoogleAuthProvider.getCredential(acct.getIdToken(), null);
        mAuth.signInWithCredential(credential)
                .addOnCompleteListener(this, task -> {
                    if (task.isSuccessful()) {
                        Log.d(TAG, "signInWithCredential:success");
                        startActivity(new Intent(LoginActivity.this,
MainActivity.class));
                        finish();
                    } else {
                        Log.w(TAG, "signInWithCredential:failure",
task.getException());
                        Toast.makeText(LoginActivity.this, "Google Sign-In failed.",
Toast.LENGTH_SHORT).show();
                    }
                });
    }

    private void signInUser(String email, String password) {
        mAuth.signInWithEmailAndPassword(email, password)
                .addOnCompleteListener(this, task -> {
                    if (task.isSuccessful()) {
                        startActivity(new Intent(LoginActivity.this,
MainActivity.class));
                        finish();
                    } else {
                        Toast.makeText(LoginActivity.this, "Authentication failed.",
Toast.LENGTH_SHORT).show();
                    }
                });
    }

    @Override
    protected void onStart() {
        super.onStart();
        FirebaseUser currentUser = mAuth.getCurrentUser();
        if (currentUser != null) {
            startActivity(new Intent(LoginActivity.this, MainActivity.class));
            finish();
        }
    }
}
```

SignUpActivity.java

```
package com.example.madproject;

import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatDelegate;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class SignUpActivity extends AppCompatActivity {
   private EditText emailEditText, passwordEditText;
   private Button signUpButton;
   private FirebaseAuth mAuth;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_NO);
       setContentView(R.layout.activity_sign_up);

       mAuth = FirebaseAuth.getInstance();
       emailEditText = findViewById(R.id.emailEditText);
       passwordEditText = findViewById(R.id.passwordEditText);
       signUpButton = findViewById(R.id.signUpButton);

       signUpButton.setOnClickListener(view -> {
           String email = emailEditText.getText().toString();
           String password = passwordEditText.getText().toString();

           if (!email.isEmpty() && !password.isEmpty()) {
               createNewUser(email, password);
           } else {
               Toast.makeText(SignUpActivity.this, "Please enter email and
password", Toast.LENGTH_SHORT).show();
           }
       });
   }

   private void createNewUser(String email, String password) {
       mAuth.createUserWithEmailAndPassword(email, password)
               .addOnCompleteListener(this, task -> {
                   if (task.isSuccessful()) {
                       FirebaseUser user = mAuth.getCurrentUser();
                       if (user != null) {
                           Toast.makeText(SignUpActivity.this, "Registration
successful.", Toast.LENGTH_SHORT).show();
```

```
                            startActivity(new Intent(SignUpActivity.this,
MainActivity.class));
                            finish();
                        }
                    } else {
                        Toast.makeText(SignUpActivity.this, "Registration failed.",
Toast.LENGTH_SHORT).show();
                    }
                });
    }
}
```

MainActivity.java

```
package com.example.madproject;

import android.app.DatePickerDialog;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatDelegate;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

import java.util.Calendar;
import java.util.HashMap;
import java.util.Map;

public class MainActivity extends AppCompatActivity {
    private Button dateButton, bookButton, loginButton, logoutButton;
    private EditText nameEditText, phoneEditText;
    private Spinner venueSpinner, slotSpinner;
    private String selectedDate, selectedSlot, selectedVenue;
    private FirebaseAuth mAuth;
    private DatabaseReference databaseReference;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_NO);
        setContentView(R.layout.activity_main);

        mAuth = FirebaseAuth.getInstance();
        databaseReference = FirebaseDatabase.getInstance().getReference("Bookings");

        // Initialize UI elements
        dateButton = findViewById(R.id.dateButton);
        bookButton = findViewById(R.id.bookButton);
        loginButton = findViewById(R.id.loginButton);
        logoutButton = findViewById(R.id.logoutButton);
        nameEditText = findViewById(R.id.nameEditText);
        phoneEditText = findViewById(R.id.phoneEditText);
        slotSpinner = findViewById(R.id.slotSpinner);
        venueSpinner = findViewById(R.id.venueSpinner);

        // Check if the user is logged in or not
        FirebaseUser currentUser = mAuth.getCurrentUser();
        if (currentUser == null) {
            loginButton.setVisibility(View.VISIBLE);
            logoutButton.setVisibility(View.GONE);
        } else {
            loginButton.setVisibility(View.GONE);
            logoutButton.setVisibility(View.VISIBLE);
        }

        // Set up date picker
        dateButton.setOnClickListener(view -> pickDate());

        // Set up booking button
        bookButton.setOnClickListener(view -> {
            selectedVenue = venueSpinner.getSelectedItem().toString();
            selectedSlot = slotSpinner.getSelectedItem().toString();
            String name = nameEditText.getText().toString();
            String phone = phoneEditText.getText().toString();

            if (selectedDate != null && selectedSlot != null && selectedVenue != null && !name.isEmpty() && !phone.isEmpty()) {
                checkSlotAvailability(name, phone);
            } else {
                Toast.makeText(this, "Please complete all fields",
Toast.LENGTH_SHORT).show();
            }
        });

        // Set up login button
        loginButton.setOnClickListener(view -> {
            startActivity(new Intent(MainActivity.this, LoginActivity.class));
        });

        // Set up logout button
        logoutButton.setOnClickListener(view -> {
            mAuth.signOut();
            // After logging out, redirect to LoginActivity
            startActivity(new Intent(MainActivity.this, LoginActivity.class));
            finish();
        });
    }

    private void pickDate() {
        Calendar calendar = Calendar.getInstance();
        int year = calendar.get(Calendar.YEAR);
        int month = calendar.get(Calendar.MONTH);
        int day = calendar.get(Calendar.DAY_OF_MONTH);

        DatePickerDialog datePickerDialog = new DatePickerDialog(this, (view, year1,
month1, dayOfMonth) -> {
            selectedDate = dayOfMonth + "/" + (month1 + 1) + "/" + year1;
            dateButton.setText(selectedDate);
        }, year, month, day);
        datePickerDialog.show();
    }

    private void checkSlotAvailability(String name, String phone) {
        DatabaseReference slotRef =
databaseReference.child(selectedVenue).child(selectedDate).child(selectedSlot);
        slotRef.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                if (snapshot.child("status").exists() &&
snapshot.child("status").getValue().equals("Booked")) {
                    Toast.makeText(MainActivity.this, "This slot is already
booked.", Toast.LENGTH_SHORT).show();
                } else {
                    saveBookingDetails(name, phone);
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(MainActivity.this, "Failed to check slot
availability.", Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void saveBookingDetails(String name, String phone) {
        FirebaseUser user = mAuth.getCurrentUser();
        if (user == null) {
            Toast.makeText(MainActivity.this, "User is not authenticated.",
Toast.LENGTH_SHORT).show();
            return;
        }

        String userId = user.getUid();
        DatabaseReference bookingRef =
databaseReference.child(selectedVenue).child(selectedDate).child(selectedSlot);
        Map<String, Object> userDetails = new HashMap<>();
        userDetails.put("name", name);
        userDetails.put("phone", phone);
        bookingRef.child("Users").child(userId).setValue(userDetails);

        Intent intent = new Intent(MainActivity.this, WaitingRoomActivity.class);
        intent.putExtra("venue", selectedVenue);
        intent.putExtra("date", selectedDate);
        intent.putExtra("slot", selectedSlot);
        startActivity(intent);
    }

    @Override
    protected void onStart() {
        super.onStart();
        // Check if user is still authenticated
        FirebaseUser currentUser = mAuth.getCurrentUser();
        if (currentUser == null) {
            startActivity(new Intent(MainActivity.this, LoginActivity.class));
            finish();
        }
    }
}
```

WaitingRoomActivity.java

```
package com.example.madproject;
import android.content.Intent;
import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class WaitingRoomActivity extends AppCompatActivity {
    private TextView userCountTextView;
    private Button leaveRoomButton;
    private DatabaseReference bookingRef;
    private String venue, date, slot;
    private FirebaseAuth mAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.waiting_room);

        userCountTextView = findViewById(R.id.userCountTextView);
        leaveRoomButton = findViewById(R.id.leaveRoomButton);

        venue = getIntent().getStringExtra("venue");
        date = getIntent().getStringExtra("date");
        slot = getIntent().getStringExtra("slot");

        mAuth = FirebaseAuth.getInstance();
        bookingRef = FirebaseDatabase.getInstance().getReference("Bookings")
                .child(venue).child(date).child(slot).child("Users");

        monitorWaitingRoom();

        leaveRoomButton.setOnClickListener(view -> leaveWaitingRoom());
    }

    private void monitorWaitingRoom() {
        bookingRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                long userCount = snapshot.getChildrenCount();
                userCountTextView.setText("Current users in waiting room: " +
userCount);

                if (userCount >= 2) {
                    Toast.makeText(WaitingRoomActivity.this, "Slot booked!",
Toast.LENGTH_LONG).show();
                    bookingRef.getParent().child("status").setValue("Booked");
                    moveToConfirmation();
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                Toast.makeText(WaitingRoomActivity.this, "Failed to load data.",
Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void leaveWaitingRoom() {
        String userId = mAuth.getCurrentUser().getUid();
        bookingRef.child(userId).removeValue().addOnCompleteListener(task -> {
            if (task.isSuccessful()) {
                Toast.makeText(WaitingRoomActivity.this, "You have left the waiting
room.", Toast.LENGTH_SHORT).show();
                finish();
            } else {
                Toast.makeText(WaitingRoomActivity.this, "Error leaving room.",
Toast.LENGTH_SHORT).show();
            }
        });
    }

    private void moveToConfirmation() {
        Intent intent = new Intent(WaitingRoomActivity.this,
ConfirmationActivity.class);
        intent.putExtra("venue", venue);
        intent.putExtra("date", date);
        intent.putExtra("slot", slot);
        startActivity(intent);
        finish();
    }
}
```

ConfirmationActivity.java

```
package com.example.madproject;
import android.os.Bundle;
import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatDelegate;

import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class ConfirmationActivity extends AppCompatActivity {
    private TextView confirmationDetailsTextView;
    private DatabaseReference bookingRef;
    private String venue, date, slot;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        AppCompatDelegate.setDefaultNightMode(AppCompatDelegate.MODE_NIGHT_NO);
        setContentView(R.layout.confirmation);

        confirmationDetailsTextView =
findViewById(R.id.confirmationDetailsTextView);

        venue = getIntent().getStringExtra("venue");
        date = getIntent().getStringExtra("date");
        slot = getIntent().getStringExtra("slot");

        bookingRef = FirebaseDatabase.getInstance().getReference("Bookings")
                .child(venue).child(date).child(slot).child("Users");

        loadConfirmationDetails();
    }
    private void loadConfirmationDetails() {
        bookingRef.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                StringBuilder details = new StringBuilder();
                details.append("Booking Details\nVenue: ").append(venue)
                        .append("\nDate: ").append(date)
                        .append("\nSlot: ").append(slot)
                        .append("\n\nUser List:\n");

                for (DataSnapshot userSnapshot : snapshot.getChildren()) {
                    String name = userSnapshot.child("name").getValue(String.class);
                    String phone =
userSnapshot.child("phone").getValue(String.class);
                    details.append("\nName: ").append(name)
                            .append("\nPhone: ").append(phone).append("\n");
                }

                confirmationDetailsTextView.setText(details.toString());
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                confirmationDetailsTextView.setText("Failed to load confirmation
details.");
            }
        });
    }
}
```

AndroidManifest.xml

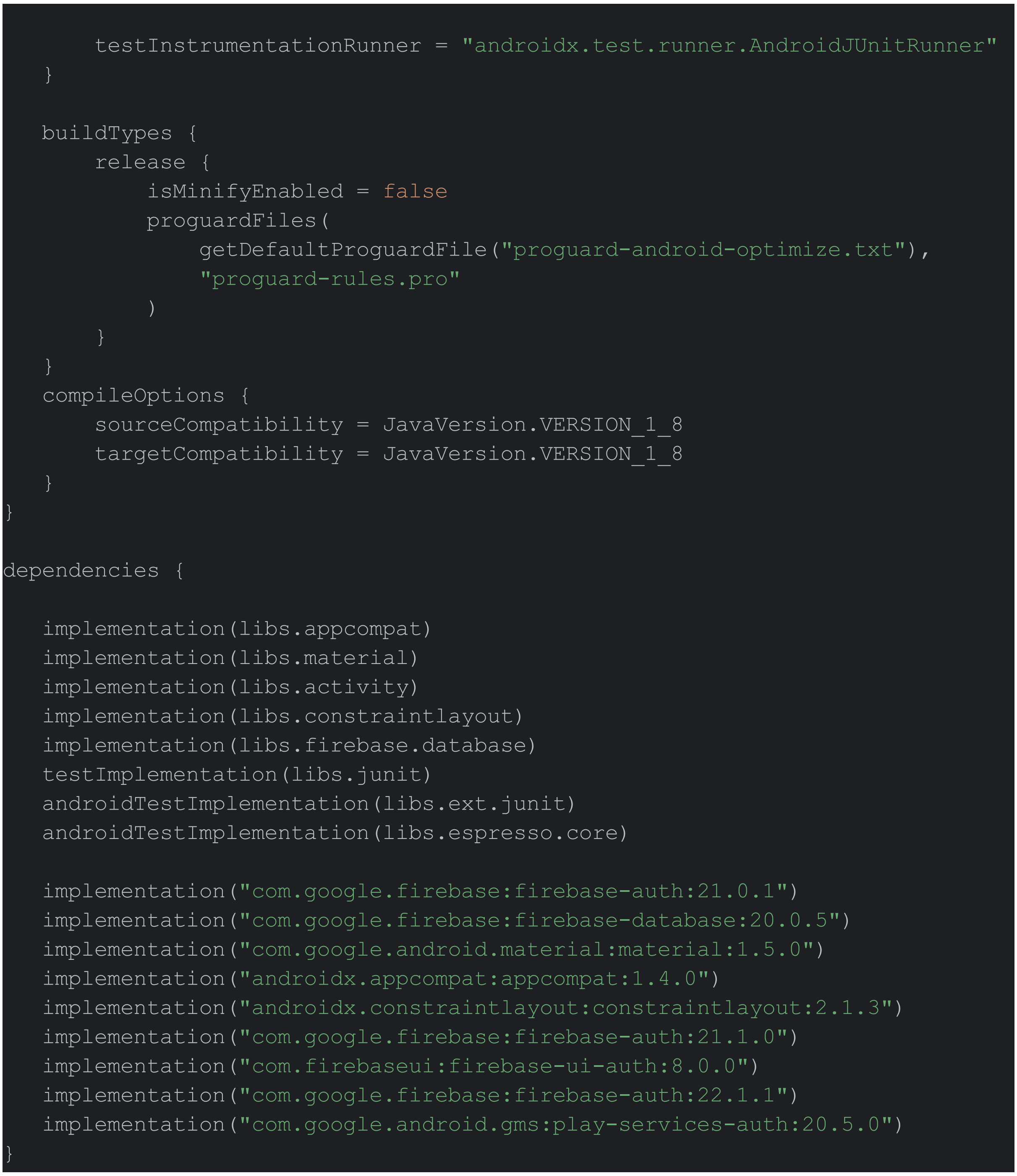
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET" />


    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MADProject"
        tools:targetApi="31">

        <activity
            android:name=".LoginActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".WaitingRoomActivity" />
        <activity android:name=".ConfirmationActivity" />
        <activity android:name=".MainActivity" android:exported="true" />
        <activity android:name=".SignUpActivity" android:exported="true" />

    </application>

</manifest>
```

Build.gradle.kts (project)

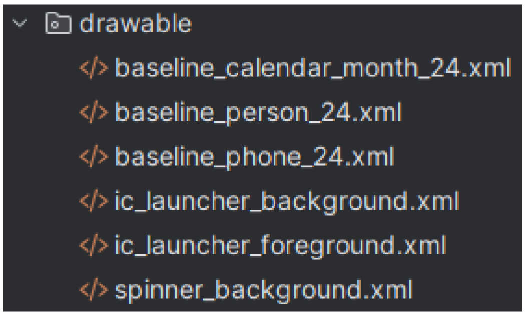
```
// Top-level build file where you can add configuration options common to all sub-projects/modules.
plugins {
   alias(libs.plugins.android.application) apply false
   alias(libs.plugins.google.gms.google.services) apply false
}
buildscript {
   dependencies {
       classpath("com.google.gms:google-services:4.3.15")
   }
}
```

Build.gradle.kts (app)

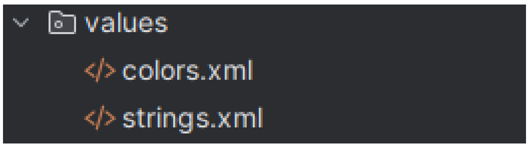
```
plugins {
   alias(libs.plugins.android.application)
   alias(libs.plugins.google.gms.google.services)
}

android {
   namespace = "com.example.madproject"
   compileSdk = 34

   defaultConfig {
       applicationId = "com.example.madproject"
       minSdk = 24
       targetSdk = 34
       versionCode = 1
       versionName = "1.0"
```



```
        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(
                getDefaultProguardFile("proguard-android-optimize.txt"),
                "proguard-rules.pro"
            )
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
}

dependencies {

    implementation(libs.appcompat)
    implementation(libs.material)
    implementation(libs.activity)
    implementation(libs.constraintlayout)
    implementation(libs.firebase.database)
    testImplementation(libs.junit)
    androidTestImplementation(libs.ext.junit)
    androidTestImplementation(libs.espresso.core)

    implementation("com.google.firebase:firebase-auth:21.0.1")
    implementation("com.google.firebase:firebase-database:20.0.5")
    implementation("com.google.android.material:material:1.5.0")
    implementation("androidx.appcompat:appcompat:1.4.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.3")
    implementation("com.google.firebase:firebase-auth:21.1.0")
    implementation("com.firebaseui:firebase-ui-auth:8.0.0")
    implementation("com.google.firebase:firebase-auth:22.1.1")
    implementation("com.google.android.gms:play-services-auth:20.5.0")
}
```

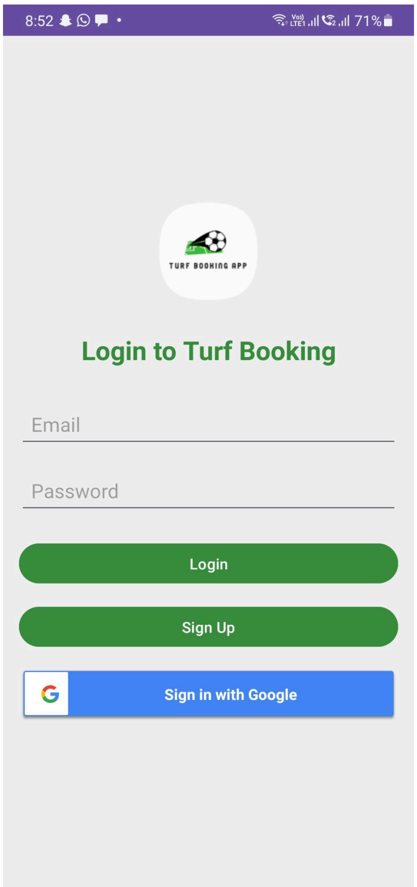


```
drawable
    baseline_calendar_month_24.xml
    baseline_person_24.xml
    baseline_phone_24.xml
    ic_launcher_background.xml
    ic_launcher_foreground.xml
    spinner_background.xml
```

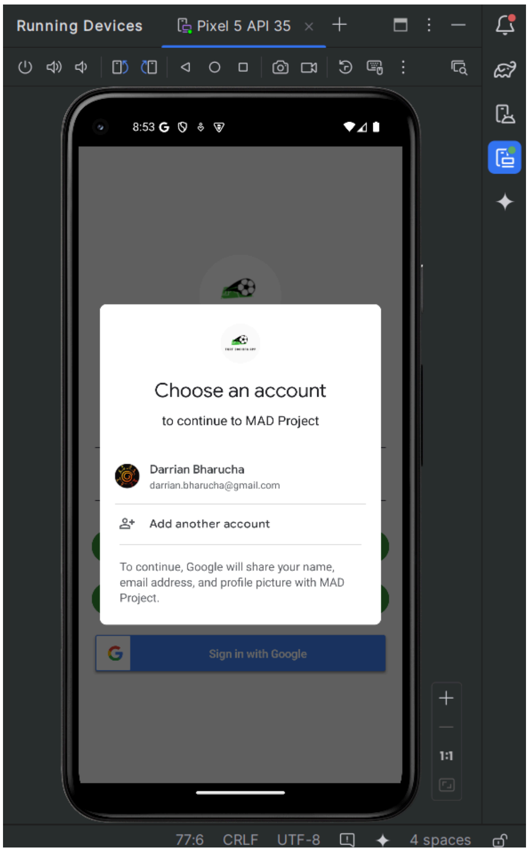


```
values
    colors.xml
    strings.xml
```

Outputs



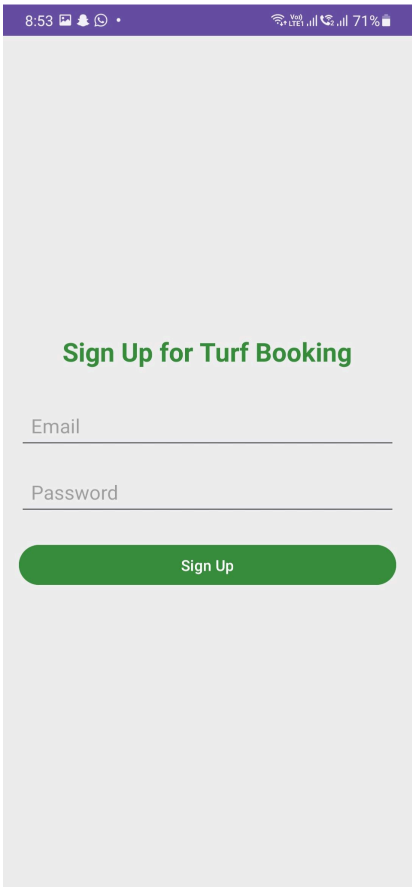
Login to Turf Booking



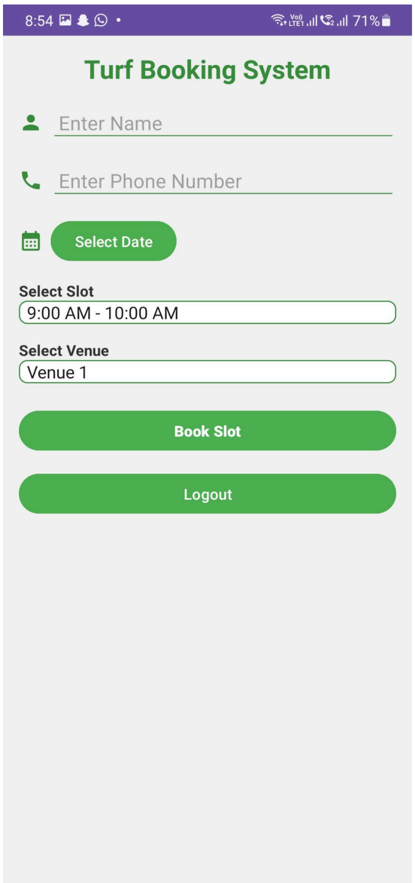
Choose an account

to continue to MAD Project

To continue, Google will share your name, email address, and profile picture with MAD Project.



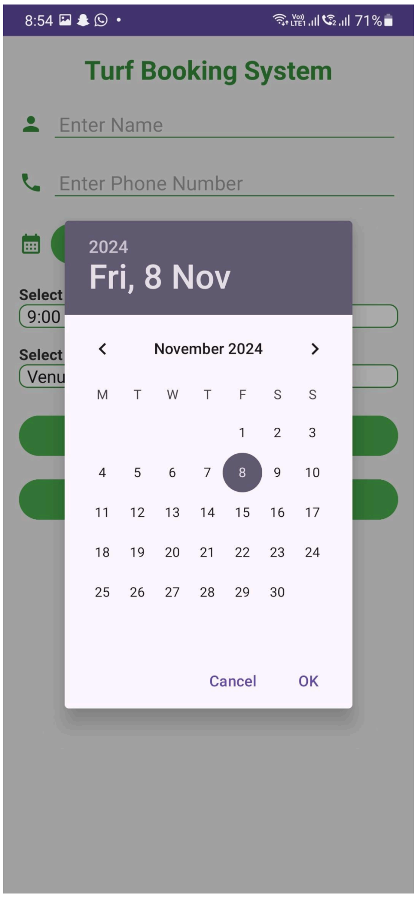
Sign Up for Turf Booking



Turf Booking System

Select Slot: 9:00 AM - 10:00 AM

Select Venue: Venue 1



8:54 • 71%

Turf Booking System

Enter Name

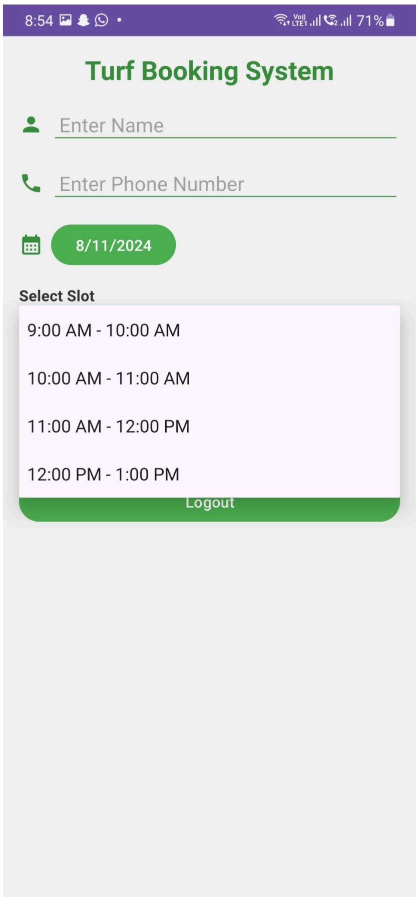
Enter Phone Number

2024

Fri, 8 Nov

November 2024

M	T	W	T	F	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	



8:54 • 71%

Turf Booking System

Enter Name

Enter Phone Number

8/11/2024

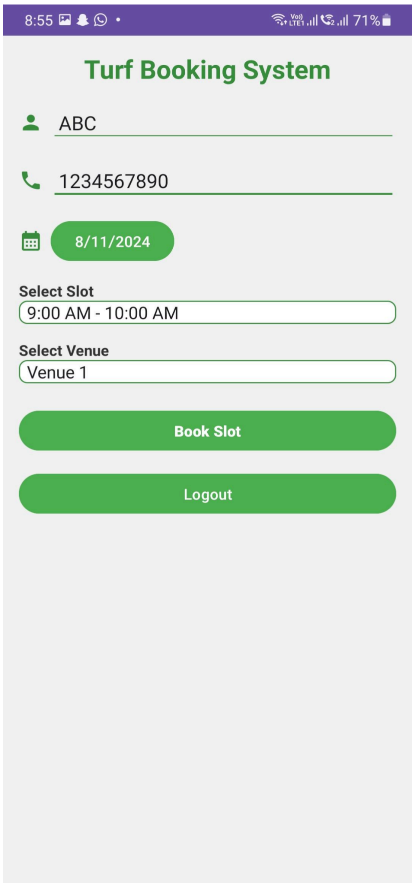
Select Slot

9:00 AM - 10:00 AM

10:00 AM - 11:00 AM

11:00 AM - 12:00 PM

12:00 PM - 1:00 PM



8:55 • 71%

Turf Booking System

ABC

1234567890

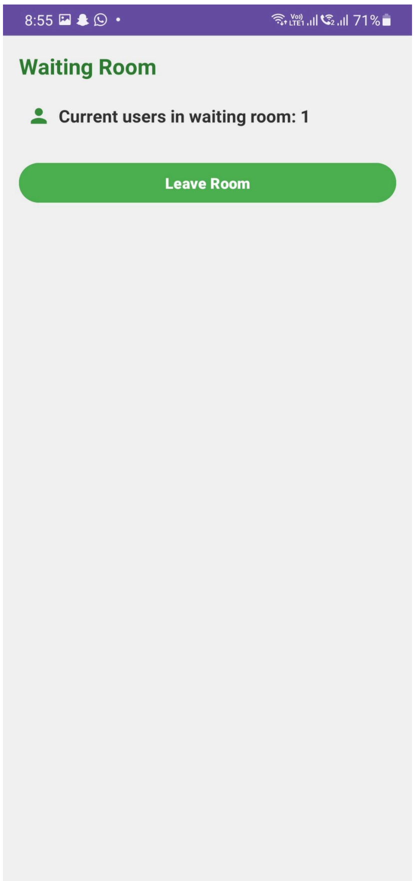
8/11/2024

Select Slot

9:00 AM - 10:00 AM

Select Venue

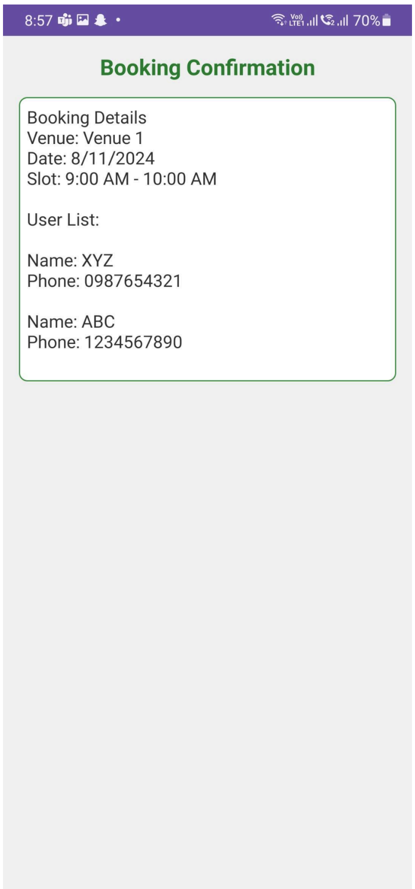
Venue 1



8:55 • 71%

Waiting Room

Current users in waiting room: 1



Booking Confirmation

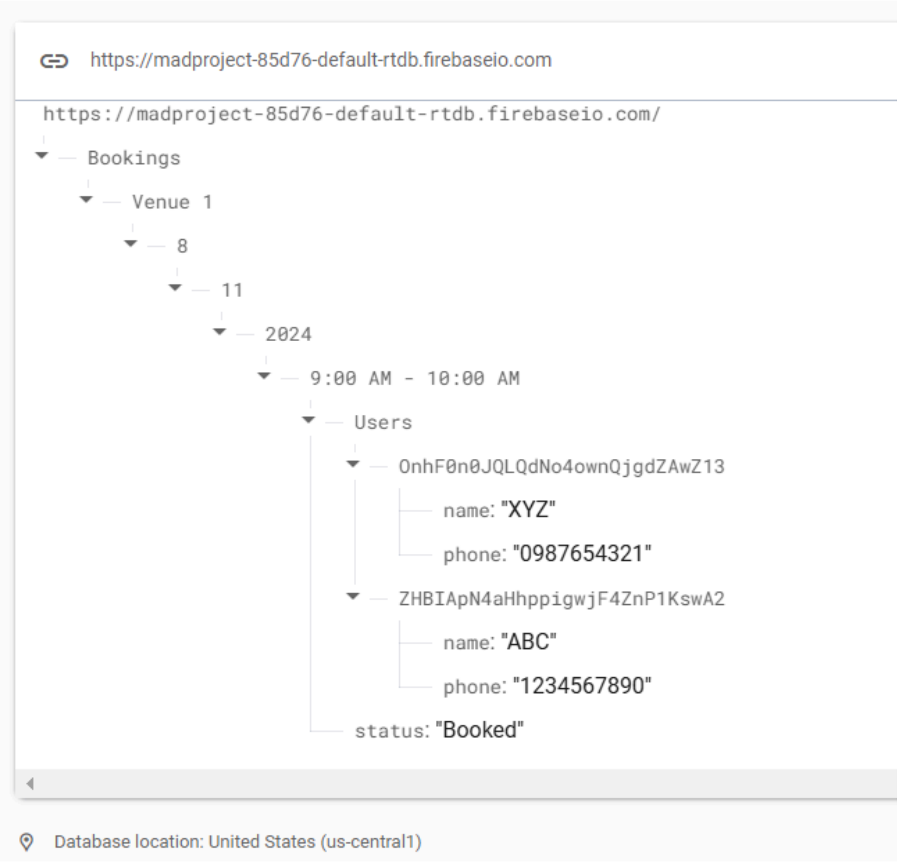
Booking Details
Venue: Venue 1
Date: 8/11/2024
Slot: 9:00 AM - 10:00 AM

User List:

Name: XYZ
Phone: 0987654321

Name: ABC
Phone: 1234567890

Real Time Database



https://madproject-85d76-default-rtdb.firebaseio.com

```
https://madproject-85d76-default-rtdb.firebaseio.com/
▾ — Bookings
    ▾ — Venue 1
        ▾ — 8
            ▾ — 11
                ▾ — 2024
                    ▾ — 9:00 AM - 10:00 AM
                        ▾ — Users
                            ▾ — OnhF0n0JQLQdNo4ownQjgdZAwZ13
                                    name: "XYZ"
                                    phone: "0987654321"
                            ▾ — ZHBIApN4aHhppigwjF4ZnP1KswA2
                                    name: "ABC"
                                    phone: "1234567890"
                          status: "Booked"
```

Database location: United States (us-central1)