

(4) WhatsApp Create project - Firebase console +

console.firebaseio.google.com/u/0/project/_/overview?forceCheckTos=true&localPort=63820&packageName=com.example.firebaseio&projectDisplayName=Firebase&dlAction=AndroidStudioC...

Create a project(Step 2 of 3)

FOR YOUR FIREBASE PROJECT

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting and more in Firebase Crashlytics, Cloud Messaging, in-app messaging, Remote Config, A/B Testing and Cloud Functions.

Google Analytics enables:

- A/B testing
- Crash-free users
- User segmentation and targeting across Firebase products
- Event-based Cloud Functions triggers
- Free unlimited reporting

Enable Google Analytics for this project
Recommended

Previous Continue



Type here to search 34°C Smoke 13:19 28-10-2022

(4) WhatsApp Create project - Firebase console +

console.firebaseio.google.com/u/0/project/_/overview?forceCheckTos=true&localPort=63820&packageName=com.example.firebaseio&projectDisplayName=Firebase&dlAction=AndroidStudioC...

Create a project(Step 3 of 3)

Configure Google Analytics

Choose or create a Google Analytics account

Default Account for Firebase

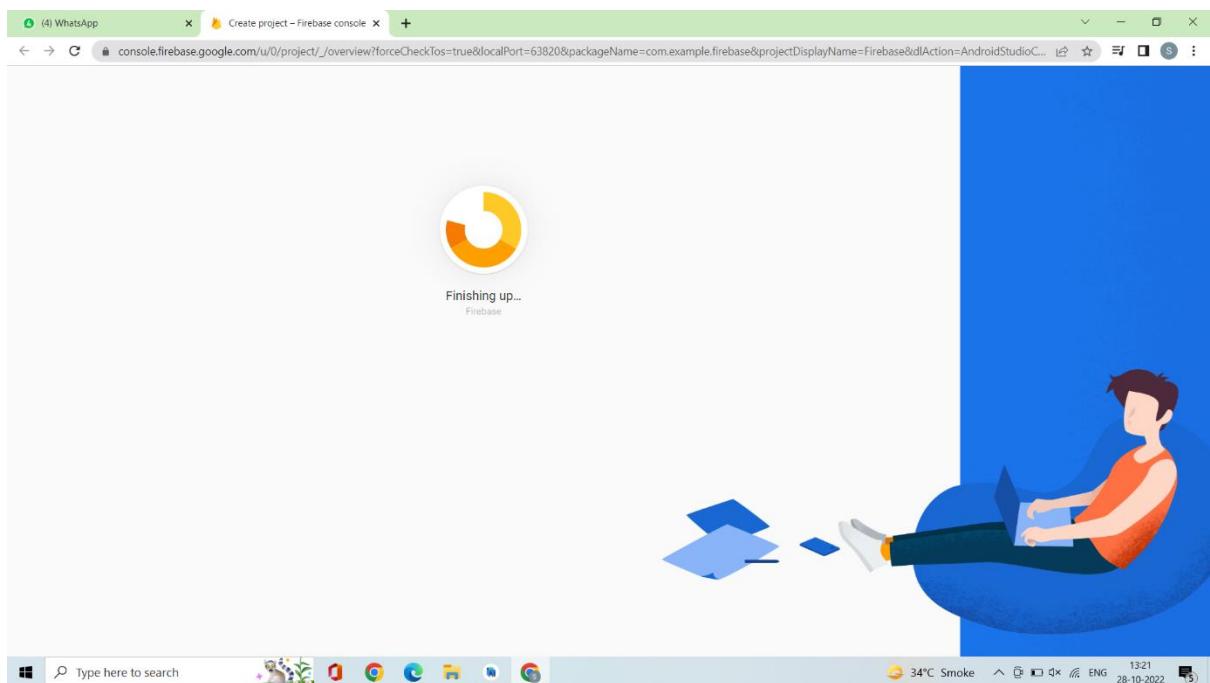
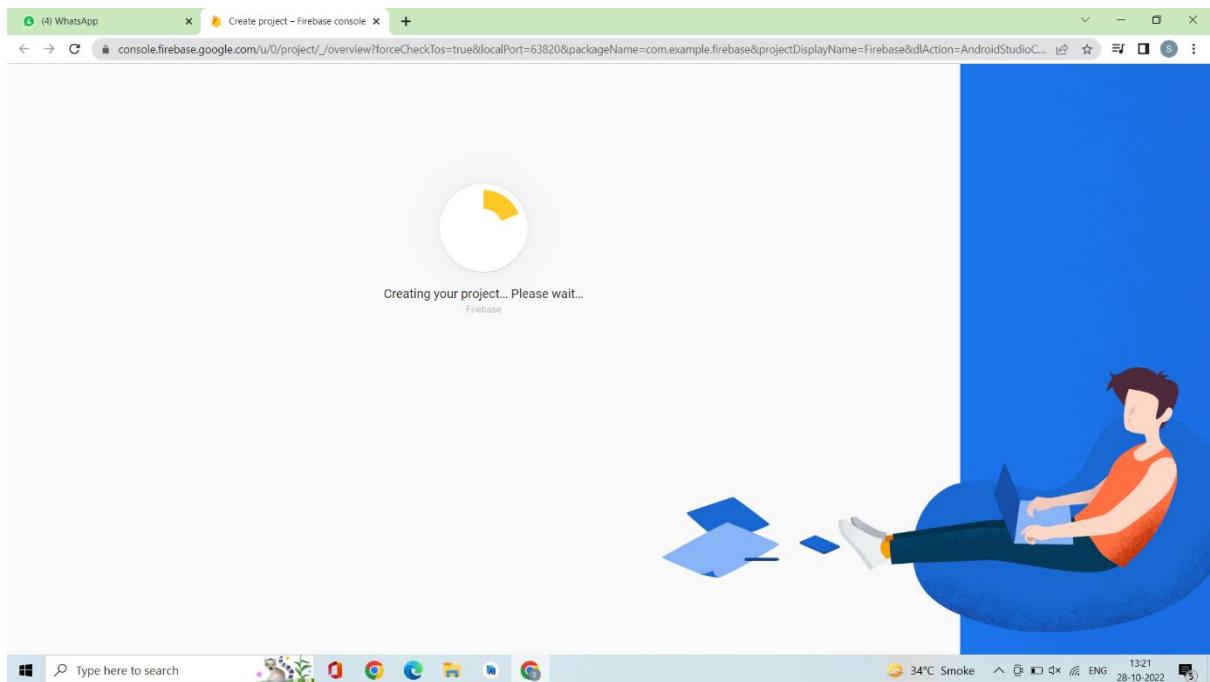
Automatically create a new property in this account

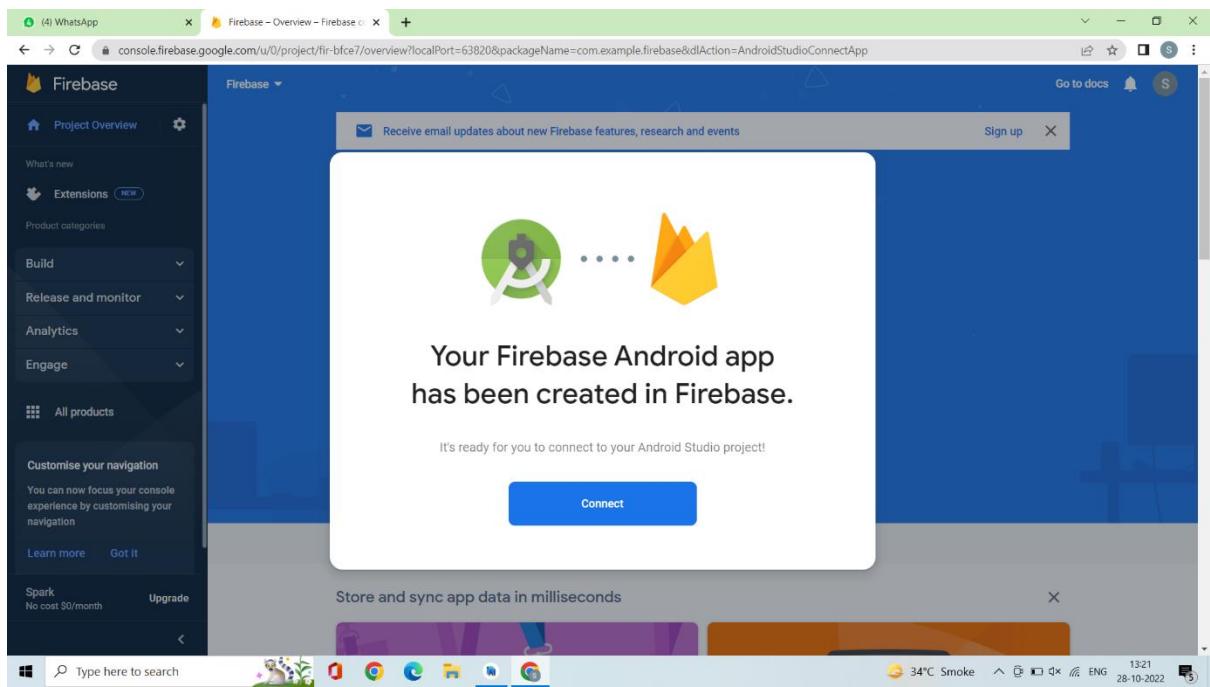
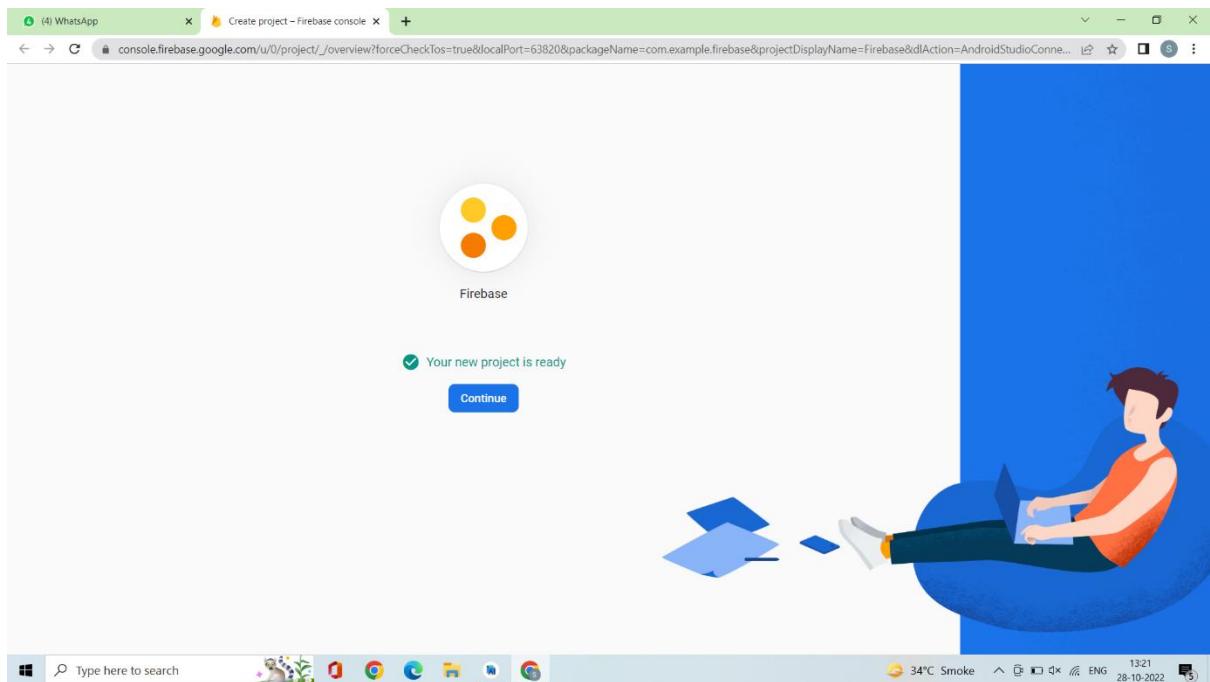
Upon project creation, a new Google Analytics property will be created in your chosen Google Analytics account and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#)

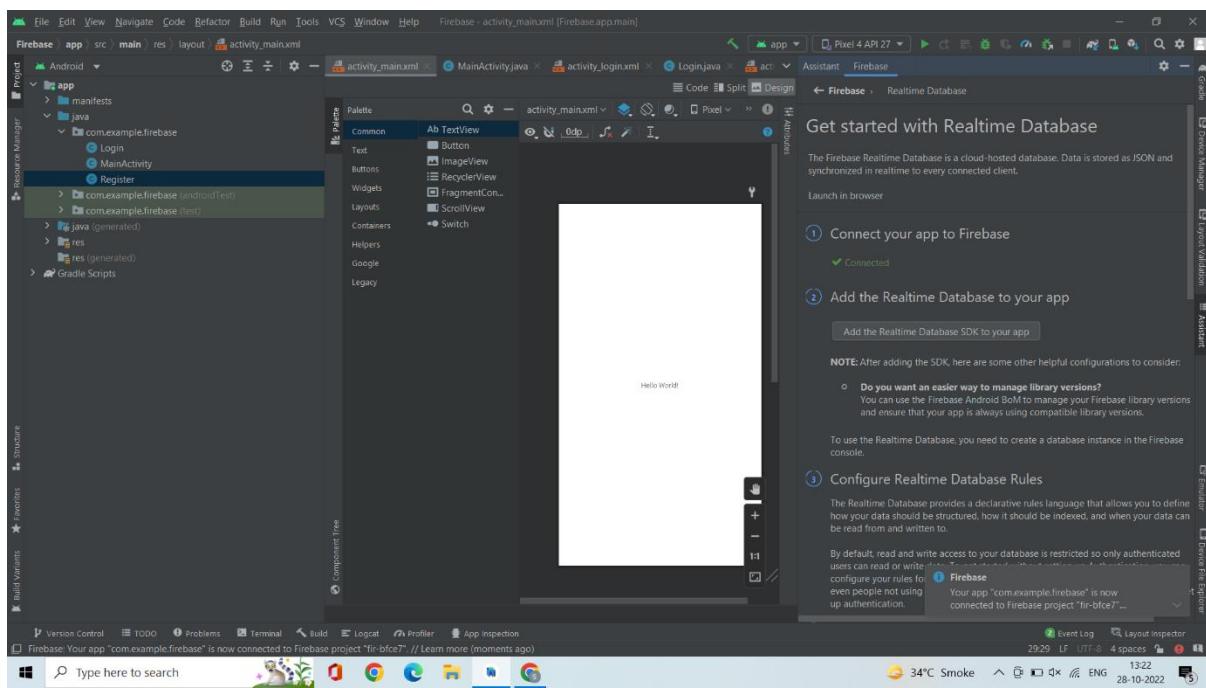
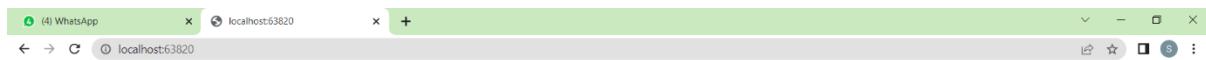
Previous Create project

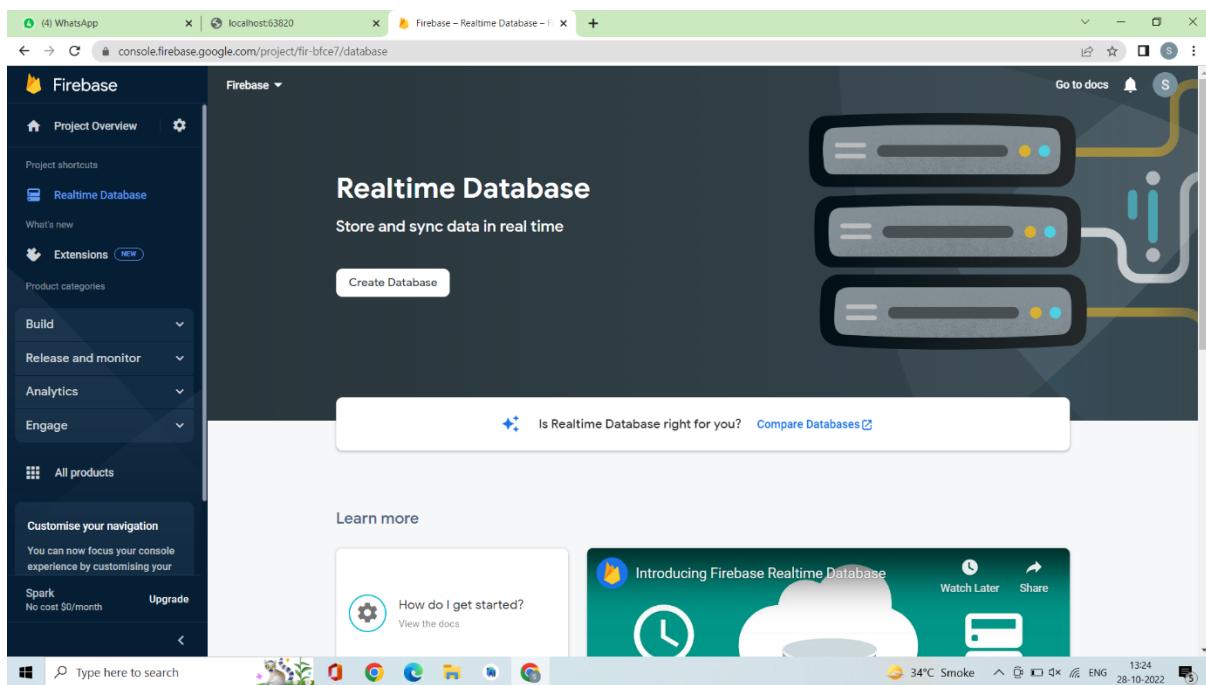
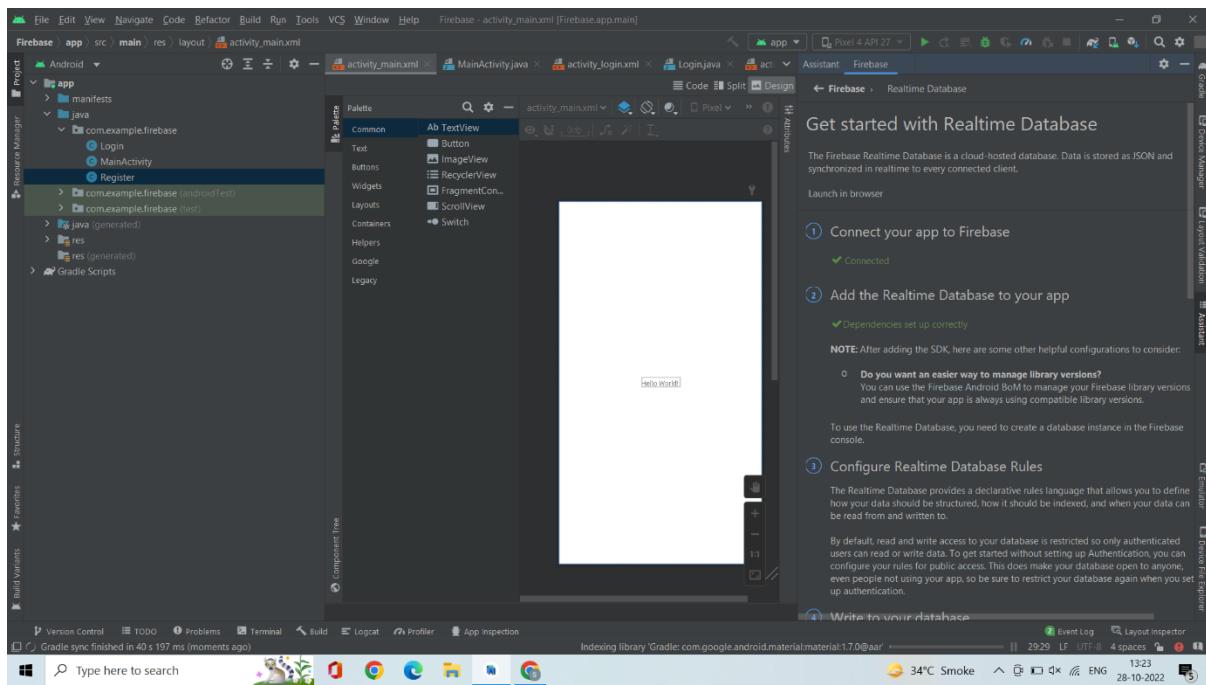


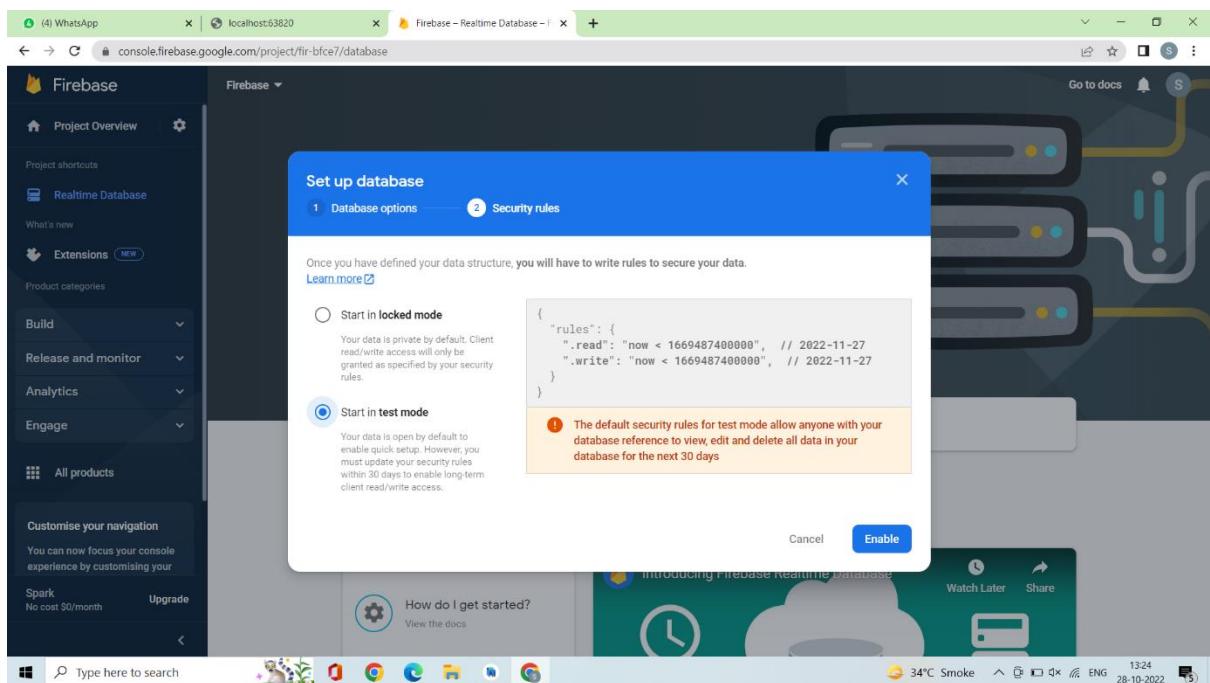
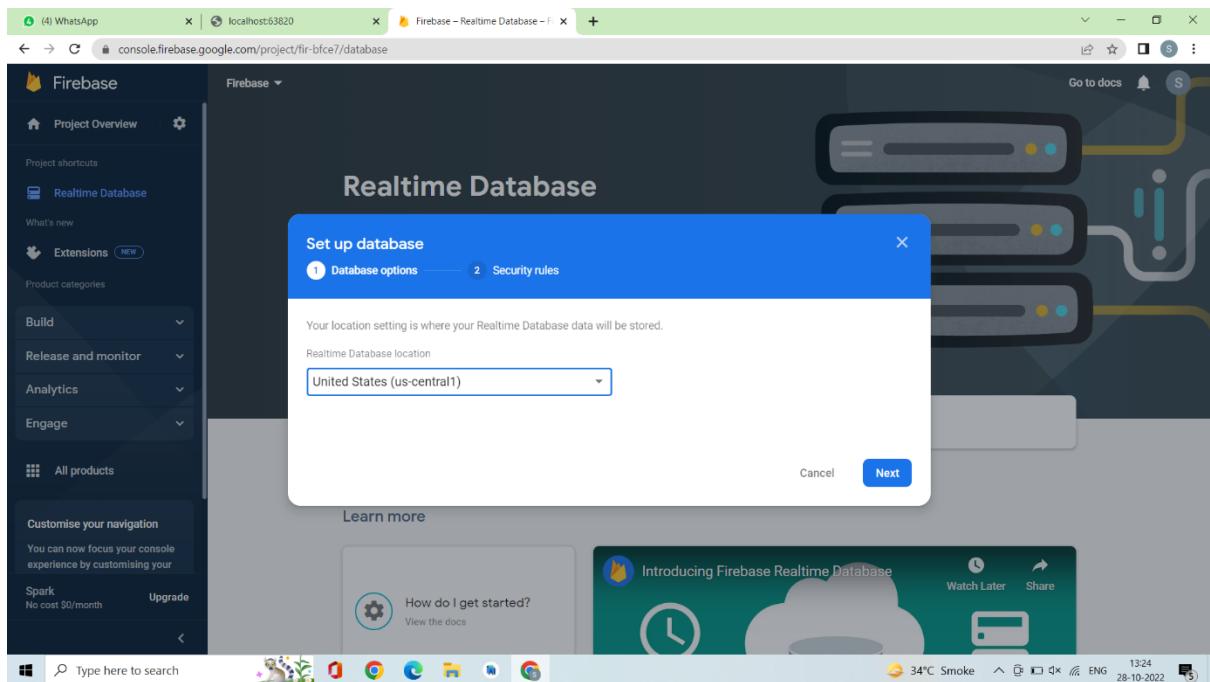
Type here to search 34°C Smoke 13:20 28-10-2022











The screenshot shows the Firebase Realtime Database console. The left sidebar includes 'Project Overview', 'Realtime Database' (selected), 'Extensions', 'Build', 'Release and monitor', 'Analytics', 'Engage', and 'All products'. A 'Customise your navigation' section is also present. The main area displays the URL <https://fir-bfce7-default-rtdb.firebaseio.com/>. A message at the top says 'Protect your Realtime Database resources from abuse, such as billing fraud or phishing' with a 'Configure App Check' link. Below the URL, it shows 'https://fir-bfce7-default-rtdb.firebaseio.com/:null'. The status bar at the bottom indicates '34°C Smoke' and the date '28-10-2022'.

The screenshot shows the Android Studio interface with the project 'example' open. The 'Login' activity is selected in the Project tab. The code editor displays the following Java code:

```
package com.example.firebaseio;
import ...
public class Login extends AppCompatActivity {
    DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReferenceFromUrl("https://fir-bfce7-default-rtdb.firebaseio.com/");
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        final EditText phone = findViewById(R.id.phone);
        final EditText password = findViewById(R.id.password);
        final Button loginBtn = findViewById(R.id.loginBtn);
        final TextView registerNow = findViewById(R.id.registerNowBtn);
        loginBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                final String phoneTxt = phone.getText().toString();
                final String passwordTxt = password.getText().toString();
                if (phoneTxt.isEmpty() || passwordTxt.isEmpty()){
                    Toast.makeText(context, "Please enter your mobile number or password", Toast.LENGTH_SHORT).show();
                } else {
                    databaseReference.child("users").addValueEventListener(new ValueEventListener() {
                        @Override
                        public void onDataChange(@NonNull DataSnapshot snapshot) {
                            if (snapshot.hasChild(phoneTxt)) {
                                Intent intent = new Intent(Login.this, MainActivity.class);
                                startActivity(intent);
                            }
                        }
                    });
                }
            }
        });
    }
}
```

The 'Device Manager' tab is visible on the right. The status bar at the bottom indicates '33°C Smoke' and the date '28-10-2022'.

The screenshot shows the Android Studio interface with the project structure and code editor. The code in `Login.java` handles Firebase authentication logic. It uses `onDataChange` to check if a user exists with the provided phone number and password. If successful, it starts the `MainActivity`. If not, it shows an error message. The code also includes an `onClick` listener for the register button.

```
databaseReference.child("users").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        if (snapshot.hasChild(phoneTxt)) {
            final String getPassword = snapshot.child(phoneTxt).child("password").getValue(String.class);
            if (getPassword.equals(passwordTxt)) {
                Toast.makeText(context, Login.this, text: "Successfully Logged in", Toast.LENGTH_SHORT).show();
                startActivity(new Intent(packageContext, Login.this, MainActivity.class));
                finish();
            } else {
                Toast.makeText(context, Login.this, text: "Wrong Mobile Number", Toast.LENGTH_SHORT).show();
            }
        }
    }
    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});

registerNow.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(packageContext, Login.this, Register.class));
    }
});
```

This screenshot is identical to the first one, showing the same code in `Login.java` for handling Firebase authentication and starting the `MainActivity` upon success or displaying an error message if the user does not exist.

```
databaseReference.child("users").addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        if (snapshot.hasChild(phoneTxt)) {
            final String getPassword = snapshot.child(phoneTxt).child("password").getValue(String.class);
            if (getPassword.equals(passwordTxt)) {
                Toast.makeText(context, Login.this, text: "Successfully Logged in", Toast.LENGTH_SHORT).show();
                startActivity(new Intent(packageContext, Login.this, MainActivity.class));
                finish();
            } else {
                Toast.makeText(context, Login.this, text: "Wrong Mobile Number", Toast.LENGTH_SHORT).show();
            }
        }
    }
    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});

registerNow.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivity(new Intent(packageContext, Login.this, Register.class));
    }
});
```

The screenshot shows the Android Studio interface with the project structure on the left and the XML code editor on the right. The XML code defines a linear layout with a text view and an edit text field.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    tools:context=".Login">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:gravity="center"
        android:textColor="#CC000000"
        android:fontFamily="sans-serif-condensed"
        android:textSize="18sp"
        android:text="Login Here"/>

    <EditText
        android:layout_marginTop="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="phone"
        android:id="@+id/phone"
        android:hint="Mobile"
        android:maxLines="1"
        android:layout_marginStart="20dp"/>

```

The screenshot shows the Android Studio interface with the project structure on the left and the XML code editor on the right. The XML code now includes an edit text field for password, a login button, and a linear layout.

```
        android:maxLines="1"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:paddingStart="20dp"
        android:paddingEnd="0dp"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:id="@+id/password"
        android:hint="Password"
        android:maxLines="1"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:paddingStart="20dp"
        android:paddingEnd="0dp"/>

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/loginBtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:text="Login"
        android:layout_marginTop="40dp"/>

    <LinearLayout
        android:layout_width="wrap_content"

```

The screenshot shows the Android Studio interface with the project structure on the left and the XML code editor on the right. The XML code defines a login screen with a title bar, a large text input field, a password input field, and a login button.

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:text="Login"
        android:layout_marginTop="40dp"/>



    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_gravity="center"
    android:layout_marginTop="10dp"/>



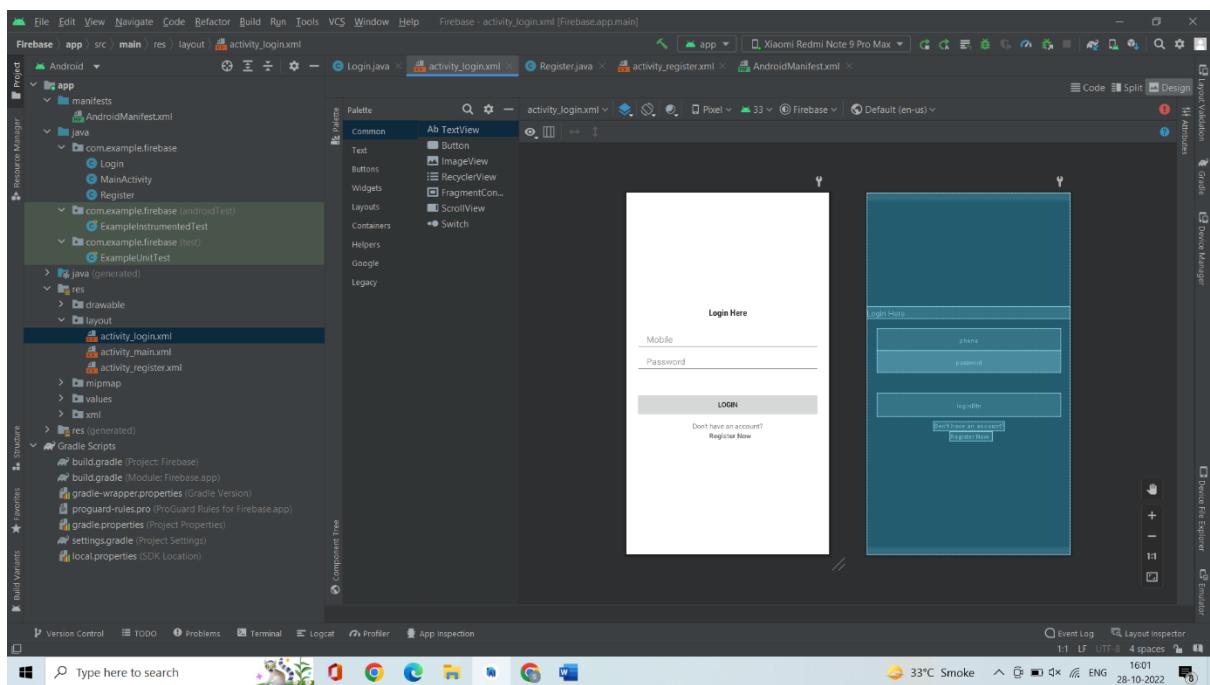
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Don't have an account?"/>



    android:layout_marginStart="5dp"
    android:id="@+id/registerNowBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Register Now"
    android:textStyle="bold"/>



</LinearLayout>
```



The screenshot shows the Android Studio interface with the project structure and code editor. The code in Register.java handles user registration. It uses Firebase Database to store user information. The code includes validation logic for fullname, email, phone, and password fields.

```
package com.example.firebaseio;
import ...;
public class Register extends AppCompatActivity {
    DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReferenceFromUrl("https://fir-bfce7-default-rtdb.firebaseio.com");
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        final EditText fullname = findViewById(R.id.fullname);
        final EditText email = findViewById(R.id.email);
        final EditText phone = findViewById(R.id.phone);
        final EditText password = findViewById(R.id.password);
        final EditText conPassword = findViewById(R.id.conpassword);
        final Button registerBtn = findViewById(R.id.registerBtn);
        final TextView loginNowBtn = findViewById(R.id.loginNow);
        registerBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                final String fullnameTxt = fullname.getText().toString();
                final String emailTxt = email.getText().toString();
                final String phoneTxt = phone.getText().toString();
                final String passwordTxt = password.getText().toString();
                final String conPasswordTxt = conPassword.getText().toString();
                if (!fullnameTxt.isEmpty() && !emailTxt.isEmpty() && !phoneTxt.isEmpty() && !passwordTxt.isEmpty() && !conPasswordTxt.isEmpty()) {
                    ...
                }
            }
        });
    }
}
```

The screenshot shows the Android Studio interface with the project structure and code editor. The code in Register.java is identical to the one above, but it uses a different database URL: "https://fir-bfce7-default-rtdb.firebaseio.com".

```
package com.example.firebaseio;
import ...;
public class Register extends AppCompatActivity {
    DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReferenceFromUrl("https://fir-bfce7-default-rtdb.firebaseio.com");
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
        final EditText fullname = findViewById(R.id.fullname);
        final EditText email = findViewById(R.id.email);
        final EditText phone = findViewById(R.id.phone);
        final EditText password = findViewById(R.id.password);
        final EditText conPassword = findViewById(R.id.conpassword);
        final Button registerBtn = findViewById(R.id.registerBtn);
        final TextView loginNowBtn = findViewById(R.id.loginNow);
        registerBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                final String fullnameTxt = fullname.getText().toString();
                final String emailTxt = email.getText().toString();
                final String phoneTxt = phone.getText().toString();
                final String passwordTxt = password.getText().toString();
                final String conPasswordTxt = conPassword.getText().toString();
                if (!fullnameTxt.isEmpty() && !emailTxt.isEmpty() && !phoneTxt.isEmpty() && !passwordTxt.isEmpty() && !conPasswordTxt.isEmpty()) {
                    ...
                }
            }
        });
    }
}
```

The screenshot shows the Android Studio interface with the project structure and code editor. The code in Register.java handles user registration logic. It checks if all fields are filled, compares the password with its confirmation, and registers the user in the Firebase database under the 'users' node. If successful, it shows a success message and finishes the activity.

```
if (fullnameTxt.isEmpty() || emailTxt.isEmpty() || phoneTxt.isEmpty() || passwordTxt.isEmpty()){
    Toast.makeText(context: Register.this, text: "Please fill all fields", Toast.LENGTH_SHORT).show();
}

else if (!passwordTxt.equals(confirmPasswordTxt)){
    Toast.makeText(context: Register.this, text: "Passwords are not matching", Toast.LENGTH_SHORT).show();
}

else {

    DatabaseReference.child("users").addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot snapshot) {
            if (snapshot.hasChild(phoneTxt)){
                Toast.makeText(context: Register.this, text: "Phone is already registered", Toast.LENGTH_SHORT).show();
            }
            else {
                DatabaseReference.child("users").child(phoneTxt).child("fullname").setValue(fullnameTxt);
                DatabaseReference.child("users").child(phoneTxt).child("email").setValue(emailTxt);
                DatabaseReference.child("users").child(phoneTxt).child("password").setValue(passwordTxt);

                Toast.makeText(context: Register.this, text: "User registered successfully.", Toast.LENGTH_SHORT).show();
                finish();
            }
        }

        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            // Handle error
        }
    });
}
}
```

This screenshot is identical to the first one, showing the same code in Register.java for handling user registration. The logic remains the same, checking for field validity, comparing passwords, and registering users in the Firebase database.

```
if (snapshot.hasChild(phoneTxt)){
    Toast.makeText(context: Register.this, text: "Phone is already registered", Toast.LENGTH_SHORT).show();
}
else {
    DatabaseReference.child("users").child(phoneTxt).child("fullname").setValue(fullnameTxt);
    DatabaseReference.child("users").child(phoneTxt).child("email").setValue(emailTxt);
    DatabaseReference.child("users").child(phoneTxt).child("password").setValue(passwordTxt);

    Toast.makeText(context: Register.this, text: "User registered successfully.", Toast.LENGTH_SHORT).show();
    finish();
}

@Override
public void onCancelled(@NonNull DatabaseError error) {
    // Handle error
}
```

The screenshot shows the Android Studio interface with the project navigation bar at the top. The main area displays the XML code for the `activity_register.xml` file. The code defines a linear layout with a text view and two edit texts for a registration form.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    tools:context=".Register">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:gravity="center"
        android:textColor="#CC000000"
        android:fontFamily="sans-serif-condensed"
        android:textSize="18sp"
        android:text="Register Here!"/>

    <EditText
        android:layout_marginTop="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="text"
        android:id="@+id/fullname"
        android:hint="Full Name"
        android:maxLines="1"
        android:layout_marginStart="20dp"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="phone"
        android:id="@+id/phone"
        android:hint="Mobile"
        android:maxLines="1"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:paddingStart="20dp"
        android:paddingEnd="0dp"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress"
        android:id="@+id/email"
        android:hint="Email"
        android:maxLines="1"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:paddingStart="20dp"
        android:paddingEnd="0dp"/>
```

This screenshot is identical to the one above, showing the XML code for `activity_register.xml`. It includes the same linear layout structure with three edit text fields for a registration form.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:orientation="vertical"
    tools:context=".Register">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:gravity="center"
        android:textColor="#CC000000"
        android:fontFamily="sans-serif-condensed"
        android:textSize="18sp"
        android:text="Register Here!"/>

    <EditText
        android:layout_marginTop="20dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="text"
        android:id="@+id/fullname"
        android:hint="Full Name"
        android:maxLines="1"
        android:layout_marginStart="20dp"/>

    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="phone"
        android:id="@+id/phone"
        android:hint="Mobile"
        android:maxLines="1"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:paddingStart="20dp"
        android:paddingEnd="0dp"/>

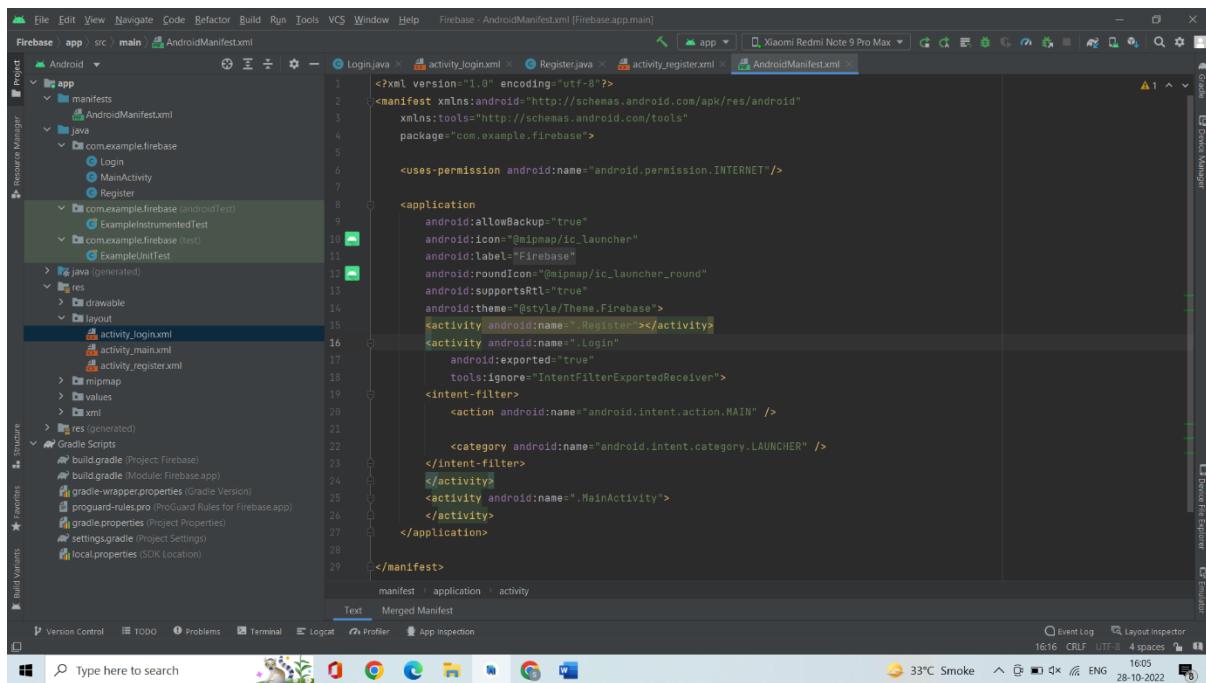
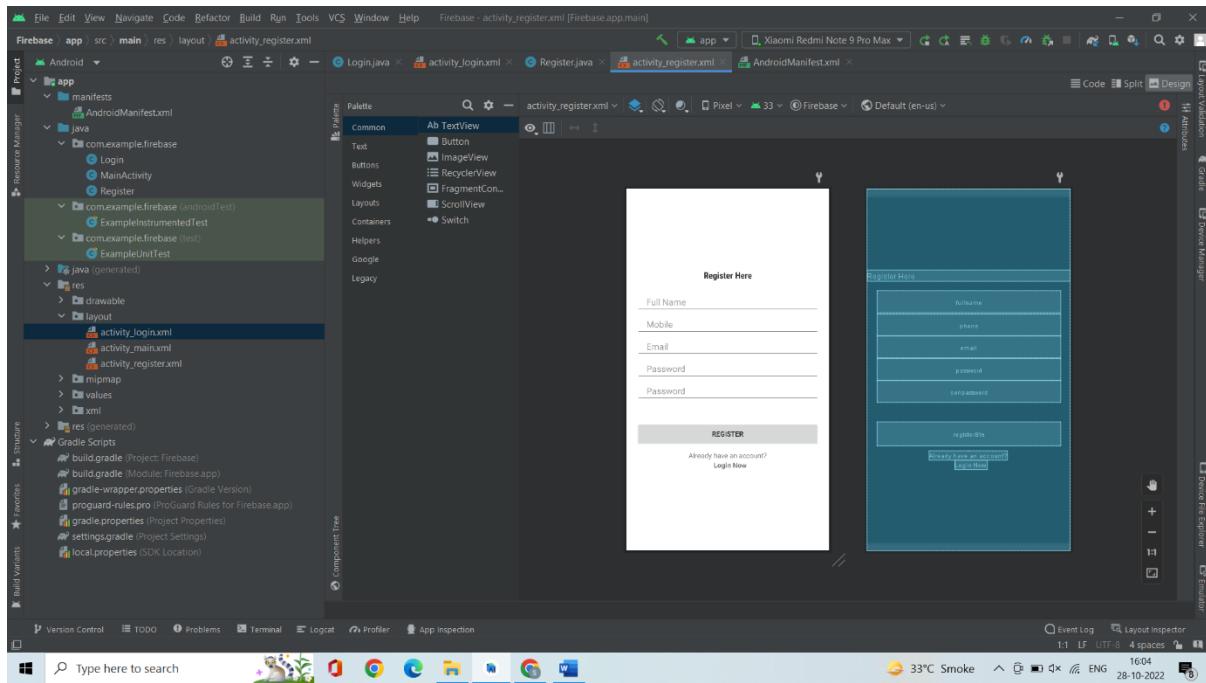
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textEmailAddress"
        android:id="@+id/email"
        android:hint="Email"
        android:maxLines="1"
        android:layout_marginStart="20dp"
        android:layout_marginEnd="20dp"
        android:paddingStart="20dp"
        android:paddingEnd="0dp"/>
```

The screenshot shows the Android Studio interface with the project navigation bar at the top. The main area displays the XML code for the `activity_register.xml` file. The code defines two `EditText` fields for email and password, and a single `AppCompatButton` for registration. The button has a green background color and a white text color. The XML code is as follows:

```
<EditText  
    android:paddingStart="20dp"  
    android:paddingEnd="80dp"/>  
  
<EditText  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:inputType="textPassword"  
    android:id="@+id/password"  
    android:hint="Password"  
    android:maxLines="1"  
    android:layout_marginStart="20dp"  
    android:layout_marginEnd="20dp"  
    android:paddingStart="20dp"  
    android:paddingEnd="80dp"/>  
  
<EditText  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:inputType="textPassword"  
    android:id="@+id/conpassword"  
    android:hint="Confirm Password"  
    android:maxLines="1"  
    android:layout_marginStart="20dp"  
    android:layout_marginEnd="20dp"  
    android:paddingStart="20dp"  
    android:paddingEnd="80dp"/>  
  
<androidx.appcompat.widget.AppCompatButton  
    android:id="@+id/registerBtn"  
    android:background="#2ECC71"  
    android:textColor="#FFFFFF"/>  
  
The bottom status bar shows the device is a Xiaomi Redmi Note 9 Pro Max, the temperature is 33°C, and the date is 28-10-2022.
```

This screenshot is identical to the one above, showing the XML code for `activity_register.xml`. It includes the same two `EditText` fields and one `AppCompatButton`. The button's XML attributes are slightly different, including a larger font size and bold text. The XML code is as follows:

```
<androidx.appcompat.widget.AppCompatButton  
    android:id="@+id/registerBtn"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginStart="20dp"  
    android:layout_marginEnd="20dp"  
    android:text="Register"  
    android:layout_marginTop="40dp"/>  
  
<LinearLayout  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:orientation="horizontal"  
    android:layout_gravity="center"  
    android:layout_marginTop="10dp"/>  
  
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Already have an account?"/>  
  
<TextView  
    android:layout_marginStart="5dp"  
    android:id="@+id/LoginNow"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Login Now"  
    android:textStyle="bold"/>  
  
The bottom status bar shows the device is a Xiaomi Redmi Note 9 Pro Max, the temperature is 33°C, and the date is 28-10-2022.
```



The screenshot shows the Android Studio interface with the build.gradle file open. The code completion dropdown is visible at the bottom of the editor, indicating that the method 'sourceCompatibility' is not found. The code snippet is as follows:

```
plugins {
    id 'com.android.application'
    id 'com.google.gms.google-services'
}

android {
    compileSdk 32

    defaultConfig {
        applicationId "com.example.firebaseio"
        minSdk 24
        targetSdk 32
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}
```

The screenshot shows the same Android Studio interface after the dependencies block has been completed. The code now includes the Firebase dependency. The completed code snippet is as follows:

```
plugins {
    id 'com.android.application'
    id 'com.google.gms.google-services'
}

android {
    compileSdk 32

    defaultConfig {
        applicationId "com.example.firebaseio"
        minSdk 24
        targetSdk 32
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }

    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {
    implementation 'androidx.appcompat:appcompat:1.5.1'
    implementation 'com.google.android.material:material:1.7.0'
    implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
    implementation 'com.google.firebase:firebase-database:20.0.6'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.3'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
    implementation 'com.google.firebase:firebase-database'
}
```

4:09 PM



Login Here

Mobile

Password

LOGIN

Don't have an account?

Register Now

4:09 PM



Register Here

Full Name

Mobile

Email

Password

Password

REGISTER

Already have an account?

[Login Now](#)

4:21 PM ⏱ 26 26 26 ...

4G Vo WiFi 87%

Register Here

Samruddhi

8356802864

kamblesamruddhi@gmail.com

.....

.....7

REGISTER

Already have an account?

[Login Now](#)

4:21 PM ⏱ ❷ ❸ ...

4G Vo WiFi 87%

Login Here

Mobile

>Password

LOGIN

Don't have an account?

Register Now

User registered successfully.

4:22 PM ⓘ ⚡ ⚡ ...

4G Vo WiFi ⚡ 87%

Login Here

8356802864

.....

LOGIN

Don't have an account?

Register Now

Wrong Mobile Number

4:22 PM ⓘ ⚡ ⚡ ...

4G Vo WiFi ⚡ 87%

Hello World!

Successfully Logged in

The screenshot shows the Firebase Realtime Database console interface. On the left, a sidebar navigation includes 'Realtime Database' (selected), 'Build', 'Release and monitor', 'Analytics', 'Engage', and 'All products'. A 'Customise your navigation' section allows users to focus their experience by customising their navigation. Below this are 'Spark' (No cost \$0/month) and 'Upgrade' options. The main area is titled 'Realtime Database' and shows the 'Data' tab selected. It displays a hierarchical tree view of database data under 'users'. One node, '8356802864', is expanded to show three child fields: 'email' (kamblesamruddhi@gmail.com), 'fullname' (Samruddhi), and 'password' (Samruddhi17). A banner at the top right encourages protecting resources from abuse with 'Configure App Check'. The bottom of the screen shows a taskbar with various icons and system status information.

Firebase

Realtime Database

Data Rules Backups Usage

Protect your Realtime Database resources from abuse, such as billing fraud or phishing Configure App Check

users + 8356802864

- email: "kamblesamruddhi@gmail.com"
- fullname: "Samruddhi"
- password: "Samruddhi17"

GD https://fir-bfce7-default.firebaseio.com > users

Database location: United States (us-central)

Spark No cost \$0/month Upgrade

Customise your navigation

Learn more Got it

33°C Smoke 1621 28-10-2022