International Institute of Information Technology
Bangalore

Software Defined Network and Network Function Virtualization

NC 864

Project Report

Sirigiri Sai Keerthan
(IMT2020511)

December 9, 2023

Contents

1 Problem Statement **2**

2 Steps to install *Android Studio* on Windows **2**

3 Creating a basic UI in Android Studio **3**

4 Adding firebase to the android project **4**

5 Google authentication using Firebase **5**

6 Email authentication using Firebase **5**

7 Connecting to real time database using Firebase **6**

8 Snapshots of the proceedings **7**
8.1 Login with Google 11
8.2 Sign Up 15

9 Documentation **17**

10 Further exploration **17**
10.1 Facebook authentication using Firebase 17
10.2 Documentation 18
10.3 Remarks 18

11 Feedback form **21**

1 Problem Statement

Create a login screen for an Android. This login screen allows user to register with his email id, and allows him to store user meta data (you can assume any structure of metadata.). Also the app login screen supports cross authentication using OAUTH with Google and Facebook accounts apart from registering. After Login, just display that the "has logged in using".

2 Steps to install *Android Studio* on Windows

1. Download Android Studio:
   - Go to the official Android Studio website: https://developer.android.com/studio.
   - Click on the "Download" button.
   - Download the version for Windows.
2. Run the installer:
   - Locate the downloaded installer file (usually a `.exe` file).
   - Double-click on the installer to run it.
3. Choose components:
   - Select "Custom" installation to customize the components or choose "Standard" for the default components.
   - Ensure that "Android Virtual Device (AVD)" and "Performance (Intel ® HAXM)" are selected.
4. Choose install location:
   - Select the destination folder where you want to install Android Studio.
   - Click "Next".
5. Install type:
   - Choose the type of setup you want (*Standard* or *custom*).
   - Click "Next".
6. Complete Installation:
   - Click "Install" to start the installation process.
7. Complete setup:
   - Once the installation is complete, click "Next" and then "Finish".
8. Run Android Studio:
   - Launch Android Studio by finding it in your Start Menu or desktop shortcuts.
   - The first time you run Android Studio, it will download and install the Android SDK components.

9. Configure SDK:
    - Follow the setup wizard to configure the Android SDK with the necessary packages.
    - Download the SDK components required for your development.
10. Finish Setup:
    - Once the SDK components are downloaded and installed, you should be ready to start Android development.
11. Configure Android Virtual Device (AVD):
    - Create an Android Virtual Device to test your applications.
    - Go to "Tools" > "AVD Manager" and create a new virtual device.

3 Creating a basic UI in Android Studio

1. Create a new project:
    - Open Android Studio.
    - Click on "Start a new Android Studio project."
    - Choose an appropriate template. For a basic UI, you can start with an "Empty Activity" or "Basic Activity."
2. Configure your project:
    - Enter the name of your application in the "Name" field.
    - Set the "Save location" for your project.
    - Choose the language (Java or Kotlin).
    - Set the minimum API level. For a basic project, you can use a relatively low API level.
3. Design the UI:
    - Open the '`res`' folder in the '`app/src/main directory`'.
    - Navigate to the '`res/layout folder`'.
    - Open the '`activity_main.xml file`'. This file defines the layout of your main activity.
    - You can use the visual editor or switch to the XML view to define your UI components.
4. Add UI components:
    - Drag and drop UI components from the Palette (located on the left side) to the layout.
    - Common UI components include '`TextView`' (for displaying text), '`Button`' (for user interaction), '`EditText`' (for user input), etc.
5. Adjust properties:
    - Customize the properties of each component using the Attributes panel. You can set properties such as text, color, size, etc.
6. Connect UI elements with code:

- Open the corresponding Java or Kotlin file for your main activity ('`MainActivity.java`' or '`MainActivity.kt`').
- Define variables for the UI components by using '`findViewById`'.
- Perform any necessary actions or logic in response to user interactions.

7. Run your app:
   - Connect a physical Android device or use an emulator.
   - Click on the "Run" button (green triangle) in the toolbar to build and run your app.
8. Test your app:
   - Interact with your app on the emulator or device to ensure that the UI behaves as expected.

4 Adding firebase to the android project

1. Create a Firebase project
2. Register your app with Firebase
   - Go to the Firebase console.
   - In the center of the project overview page, click the Android icon or Add app to launch the setup workflow.
   - Enter your app's package name in the Android package name field.
   - (Optional) Enter other app information: App nickname and Debug signing certificate SHA-1.
   - Click Register app.
3. Add a Firebase configuration file
   - Download and then add the Firebase Android configuration file (google-services.json) to your app:
     (a) Click Download google-services.json to obtain your Firebase Android config file.
     (b) Move your config file into the module (app-level) root directory of your app
   - To make the values in your google-services.json config file accessible to Firebase SDKs, you need the Google services Gradle plugin (google-services).
     (a) In your root-level (project-level) Gradle file (`<project>/build.gradle.kts or <project>/build.gradle`), add the Google services plugin as a dependency
     (b) In your module (app-level) Gradle file (`usually <project>/<app-module>/build.gradle.kts or <project>/<app-module>/build.gradle`), add the Google services plugin
4. Add Firebase SDK's to your app
   - In your module (app-level) Gradle file (usually `<project>/<app-module>/build.gradle.kts or <project>/<app-module>/build.gradle`), add the dependencies for the Firebase products that you want to use in your app.

- After adding the dependencies for the products you want to use, sync your Android project with Gradle files.

For further information one can look into this link.

5 Google authentication using Firebase

1. Enable Google Sign-In in Firebase Console:
   - In the Firebase Console, navigate to "Authentication" "Sign-in method."
   - Enable Google as a sign-in provider.
2. Update Dependencies:
   - Open your app-level build.gradle file.
   - Add the dependencies
   - Click "Sync Now" in Android Studio to sync your project with the updated dependencies.
3. Implement Google Sign-In in Your Android App:
   - In your '`MainActivity`' or another relevant activity, initialize Firebase Authentication
   - Configure Google Sign-In options
   - Initialize '`GoogleSignInClient`' using the configured options
   - Trigger Google Sign-In when the user taps on a button or performs a relevant action
   - Override '`onActivityResult`' to handle the result of the sign-in activity
   - Implement the '`handleSignInResult`' method to process the sign-in result
   - After obtaining the GoogleSignInAccount, authenticate with Firebase

6 Email authentication using Firebase

1. Set Up Firebase Authentication:
   - In the Firebase Console, go to the "Authentication" section.
   - Enable the "Email/Password" sign-in method.
2. Add Firebase SDK to your Android Studio Project:
   - Open your Android Studio project.
   - Add the Firebase SDK by adding the dependencies to your '`build.gradle`' files.
3. Initialize Firebase in your App:
   - In your '`Application`' class or the main activity, initialize Firebase
4. Create a Registration Activity:
   - Create a new activity for user registration.
   - Design a layout with input fields for email and password, along with a registration button.

5. Implement Firebase Email Registration:
    - In the registration activity, get references to the email and password fields.
    - Use the Firebase Authentication API to create a new user account with the entered email and password.
6. Create a Login Activity:
    - Create a new activity for user login.
    - Design a layout with input fields for email and password, along with a login button.
7. Implement Firebase Email Login:
    - In the login activity, get references to the email and password fields.
    - Use the Firebase Authentication API to sign in the user with the entered email and password.
8. Handle User Authentication State:
    - You may want to check the user's authentication state when your app starts. If the user is already authenticated, you can skip the login/registration screens and take them directly to the main app screen.
9. Add Sign Out Functionality:
    - Implement a sign-out button.

7 Connecting to real time database using Firebase

1. Create a Database
    - Navigate to the Realtime Database section of the Firebase console. You'll be prompted to select an existing Firebase project. Follow the database creation workflow.
    - Select a starting mode for your Firebase Security Rules:
        (a) Test mode
            – Good for getting started with the mobile and web client libraries, but allows anyone to read and overwrite your data.
        (b) Locked mode
            – Denies all reads and writes from mobile and web clients. Your authenticated application servers can still access your database.
    - Choose a location for the database.
        (a) Depending on the location of the database, the URL for the new database will be in one of the following forms:
            – `DATABASE_NAME.firebaseio.com` (for databases in us-central1)
            – `DATABASE_NAME.REGION.firebasedatabase.app` (for databases in all other locations)
    - Click done.
2. Add the Realtime Database SDK to your app

- In your module (app-level) Gradle file (usually `<project>/<app-module>/build.gradle.kts` or `<project>/<app-module>/build.gradle`), add the dependency for the Realtime Database library for Android.

3. Configure Realtime Database Security Rules

   - The Realtime Database provides a declarative rules language that allows you to define how your data should be structured, how it should be indexed, and when your data can be read from and written to.

4. Write to your database.

5. Read from your database.

For further information one can look into this link

8 Snapshots of the proceedings

Figure 1: This image is used for the background of the app

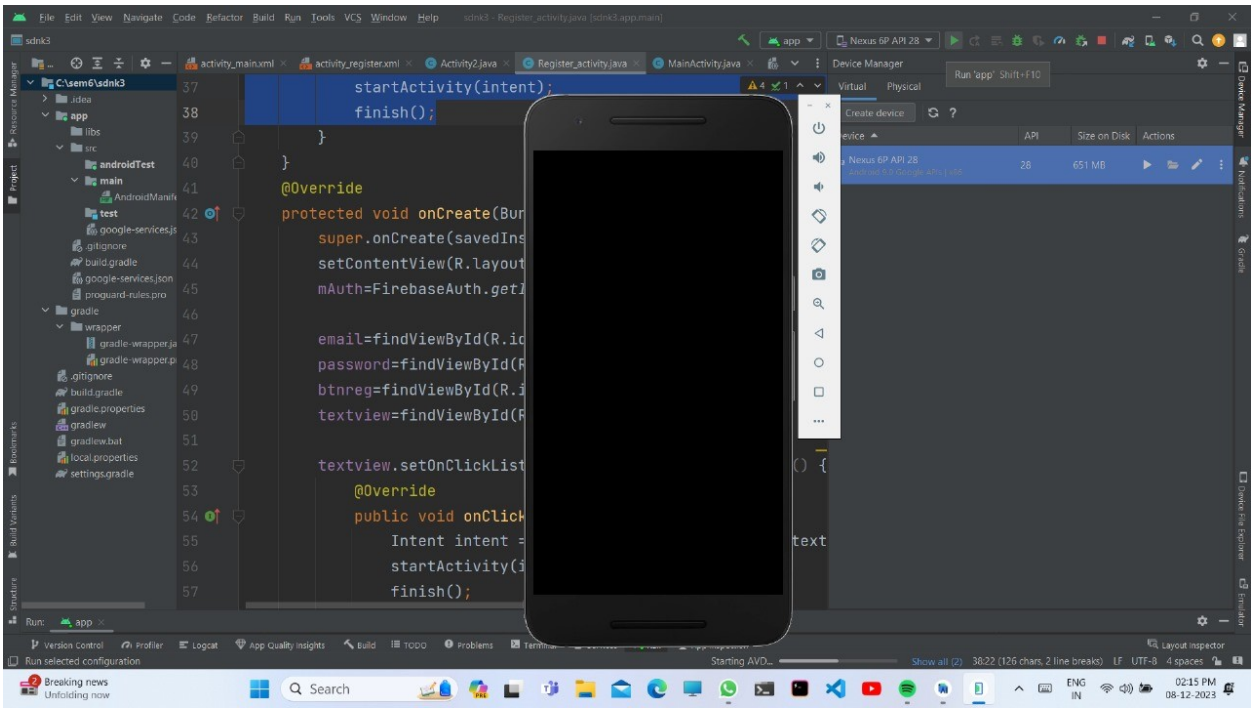

Figure 2: Device is in power off mode

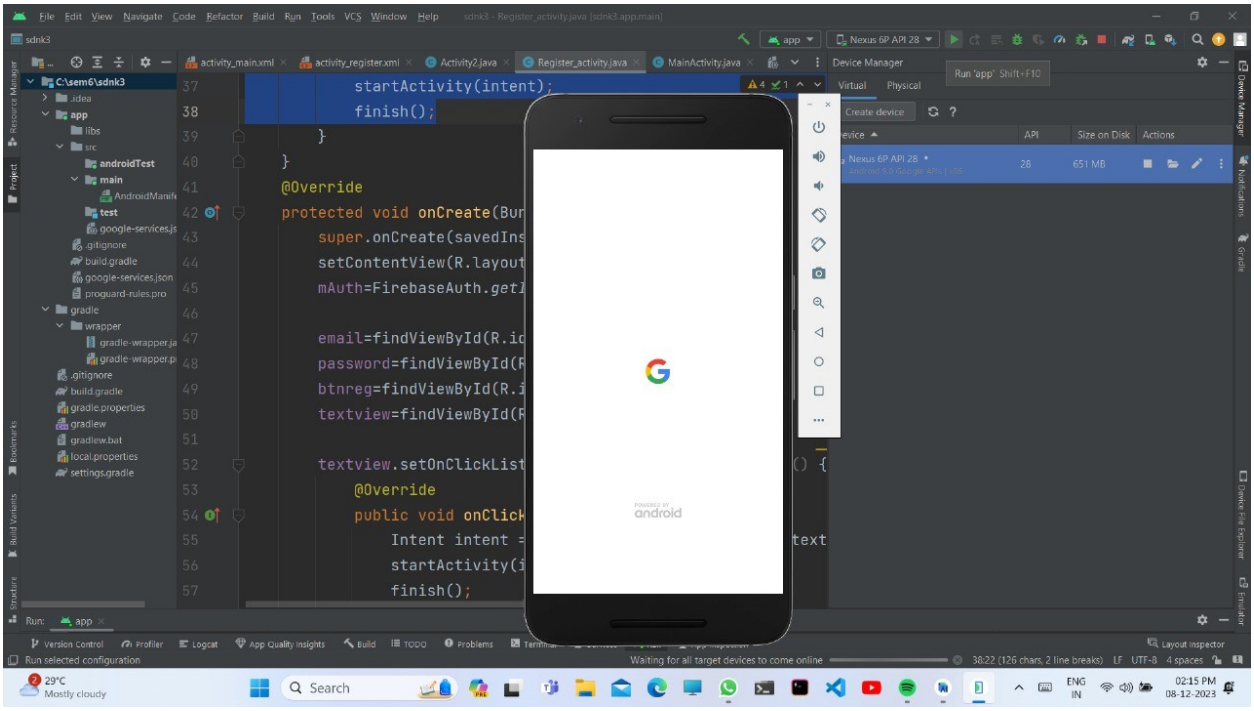

Figure 3: Now the device is being turned on

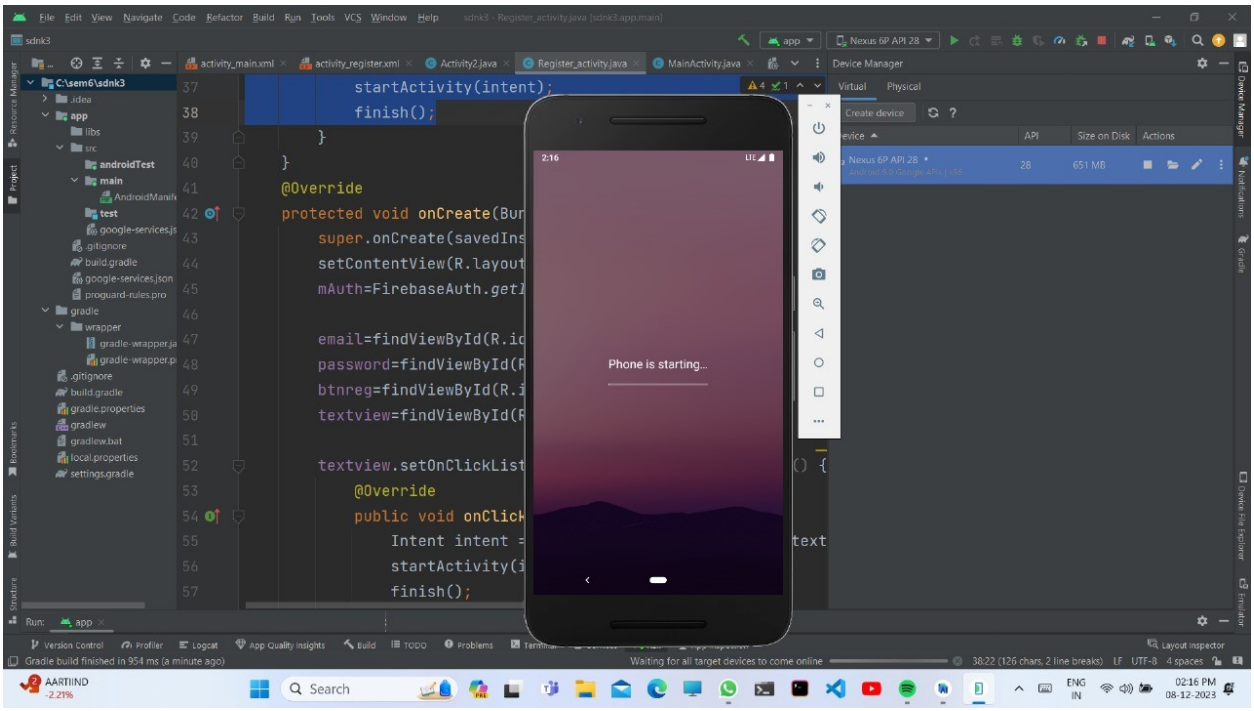

Figure 4: You can see that the device is about to start while setup is being processed

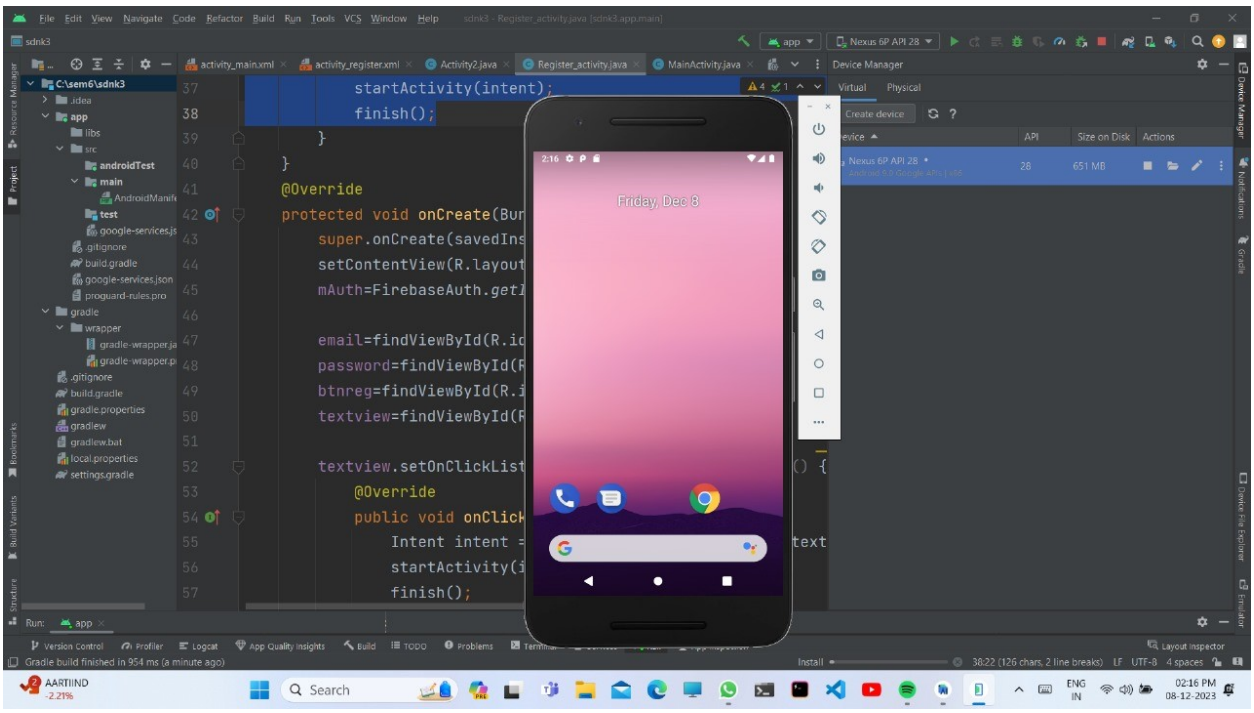

Figure 5: Now the device is turned on

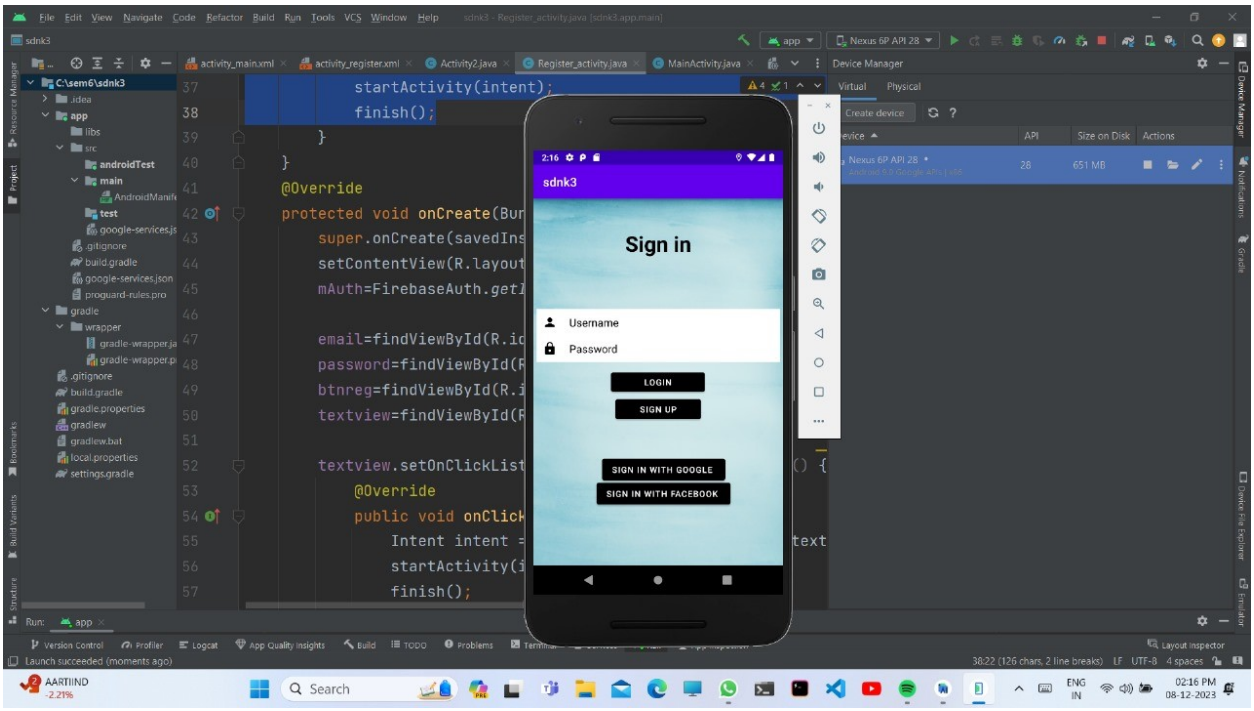

Figure 6: This is our app main page. Here any user can either login or sign-up

8.1 Login with Google

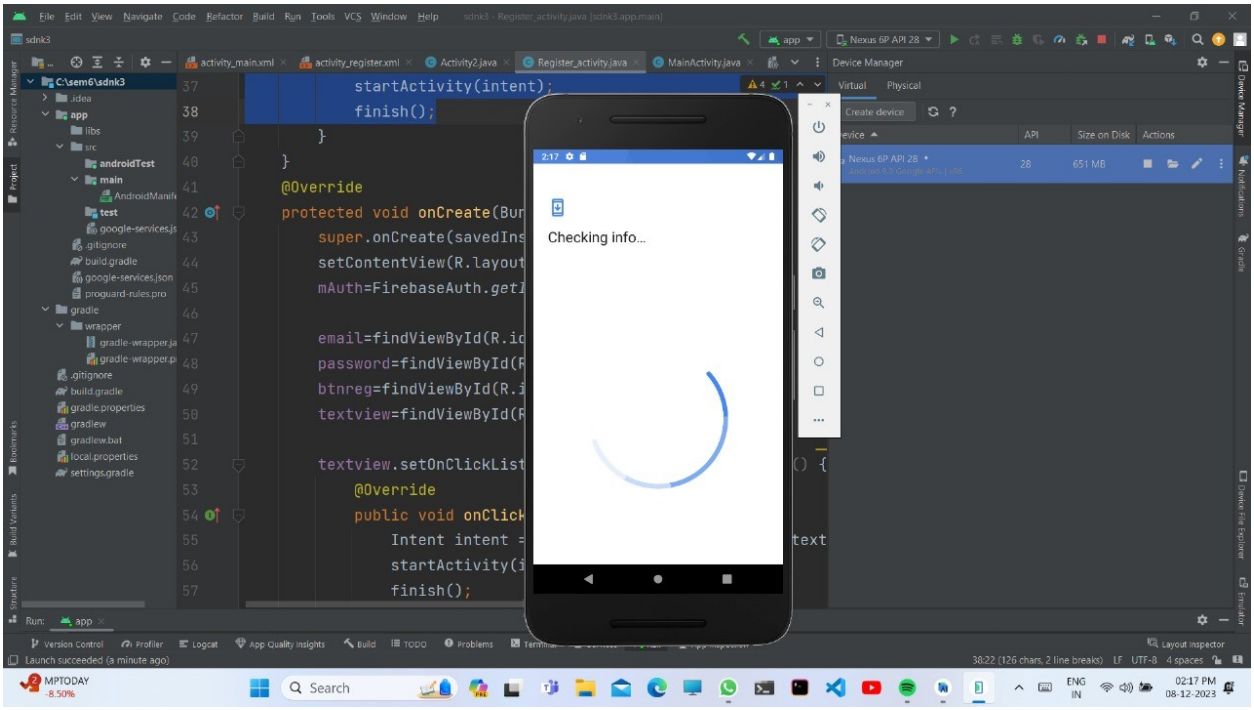

Figure 7: You will be redirected to above page

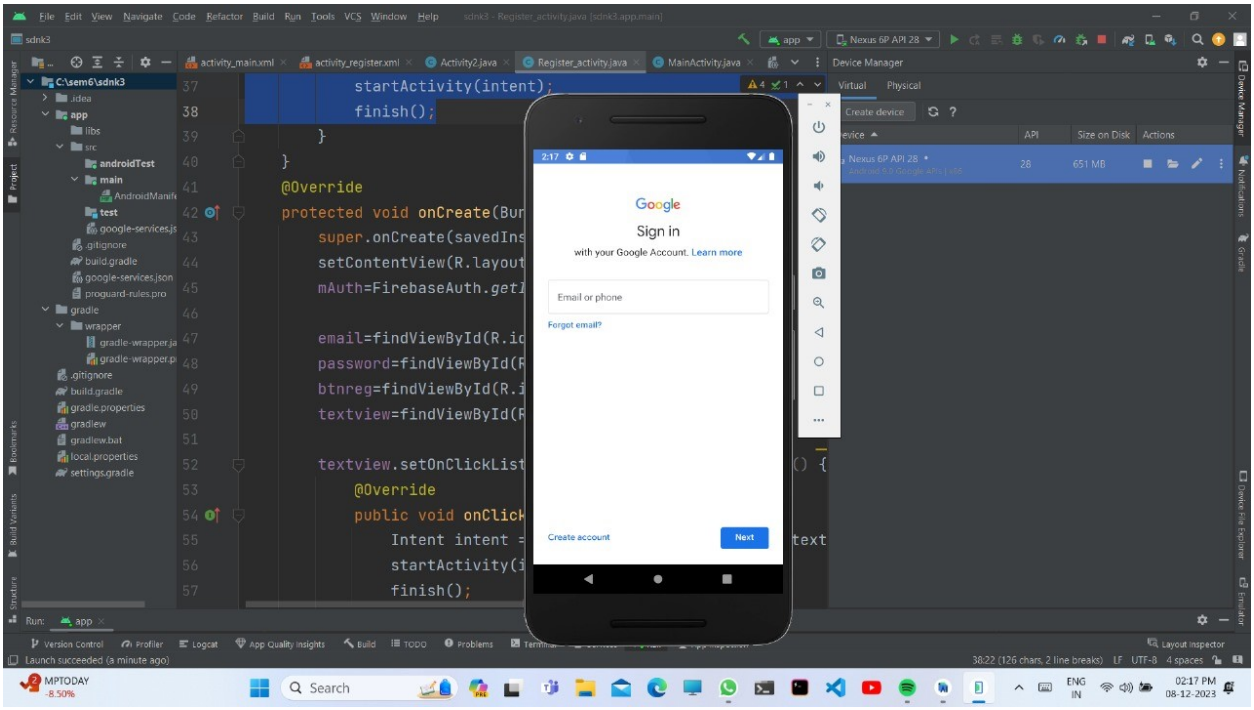

Figure 8: On this page we have to enter your email

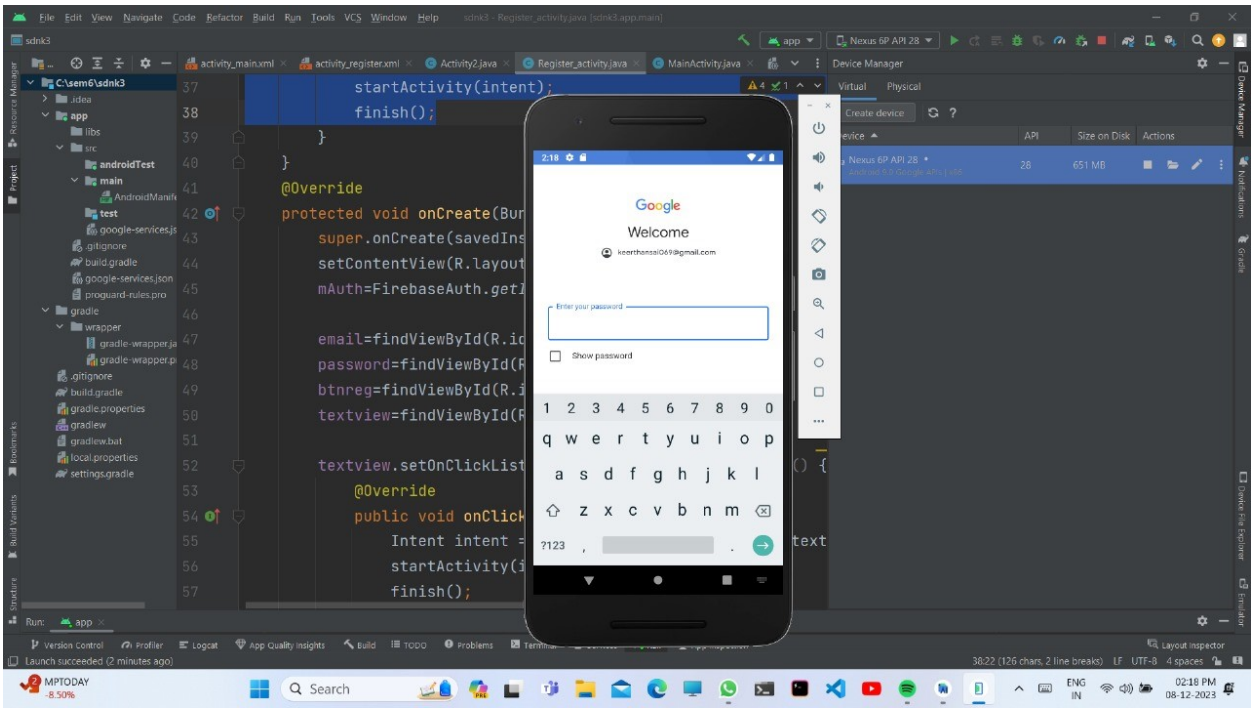

Figure 9: On this page we have to enter your password

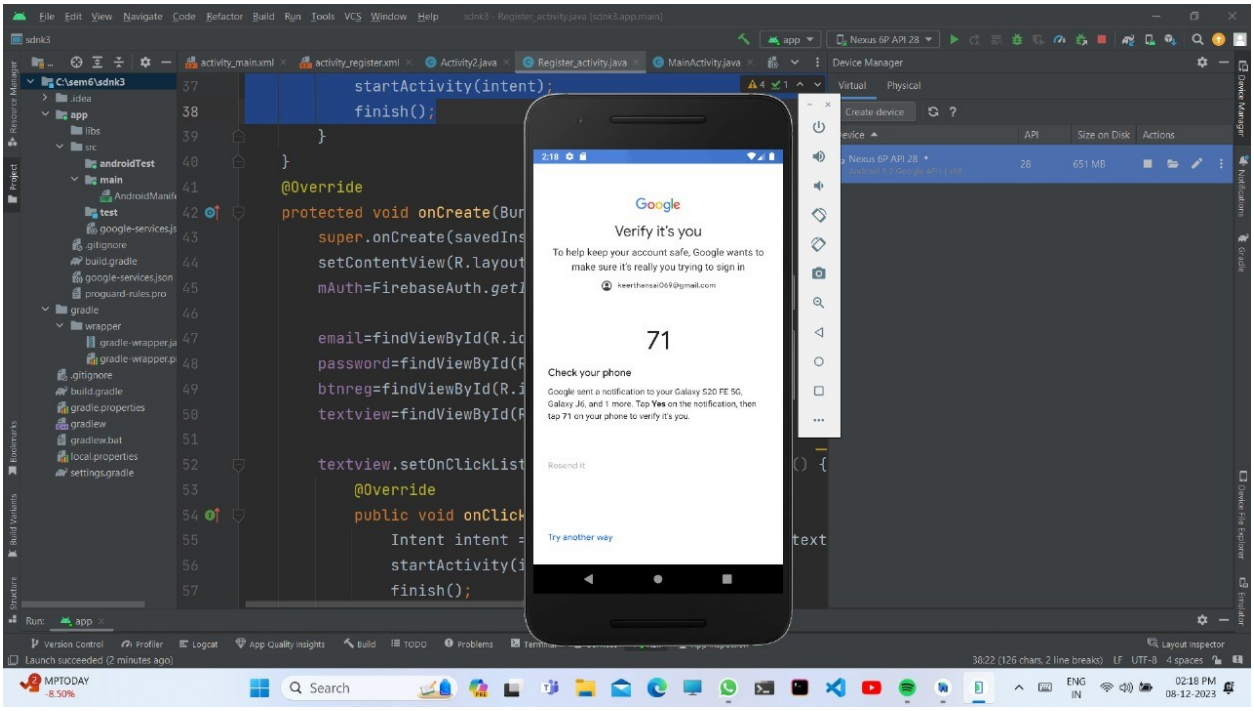

Figure 10: This page is for 2-step verification

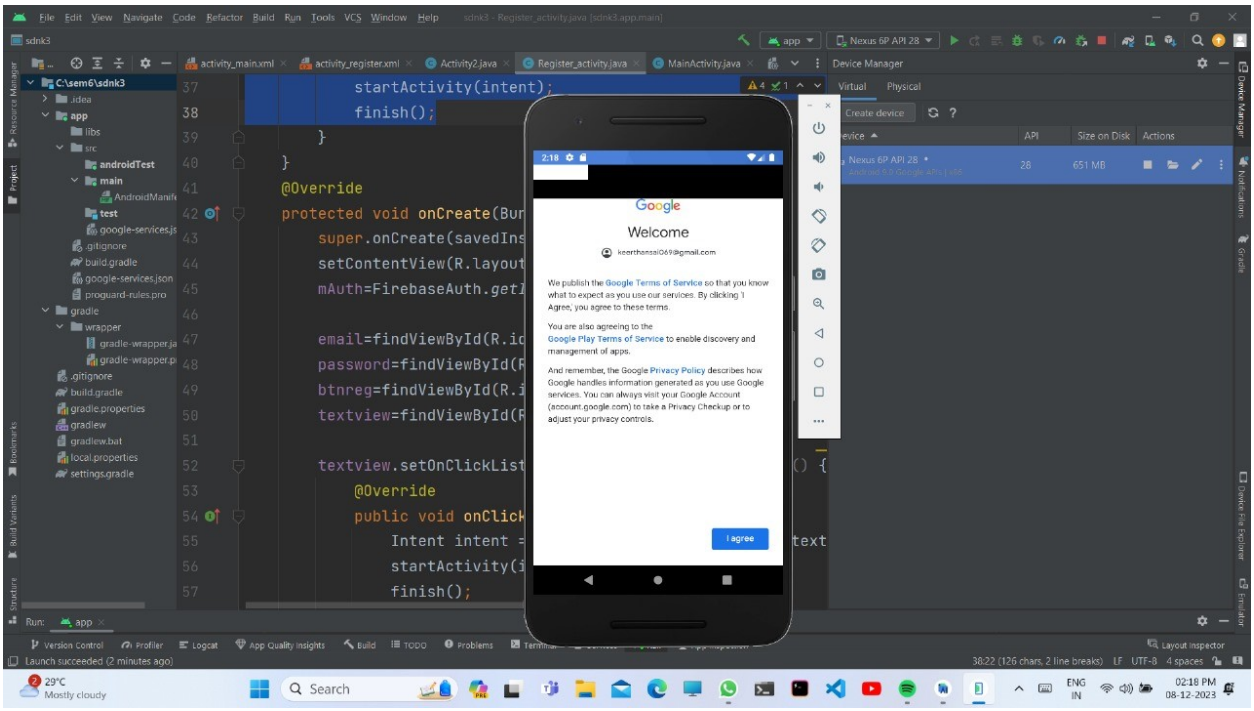

Figure 11: After login successfully we get this

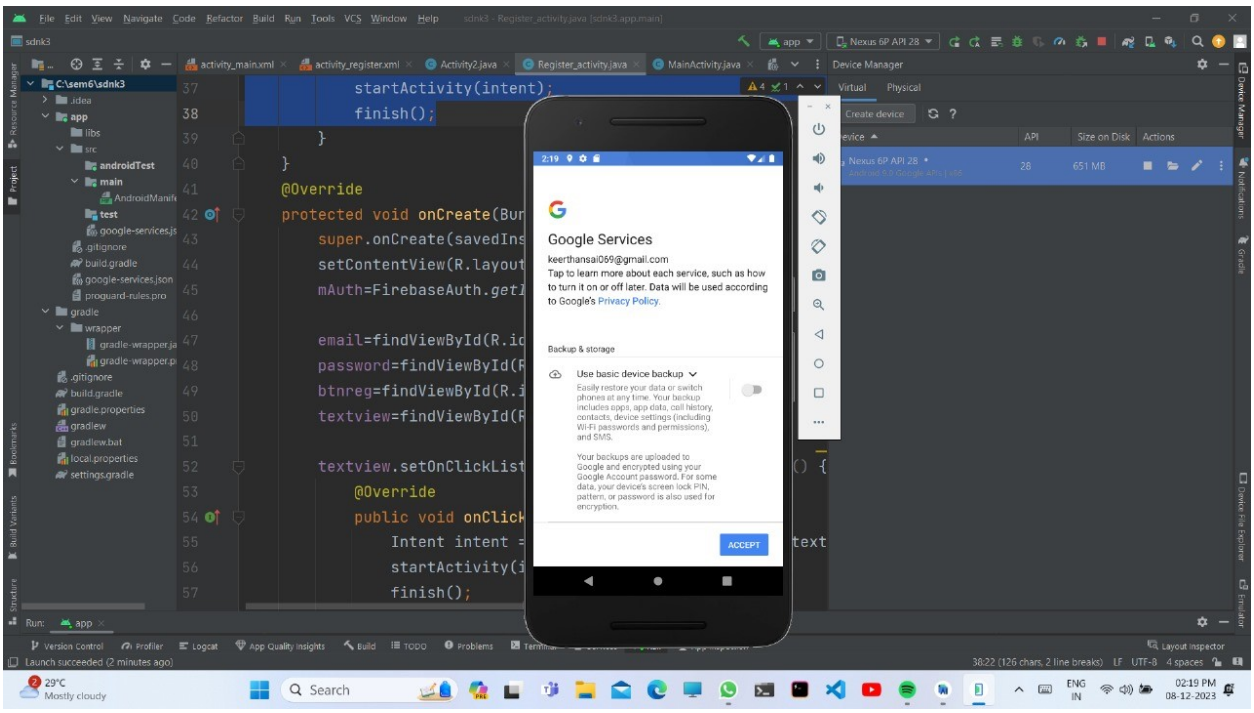

Figure 12: We can choose to backup or simply ignore it

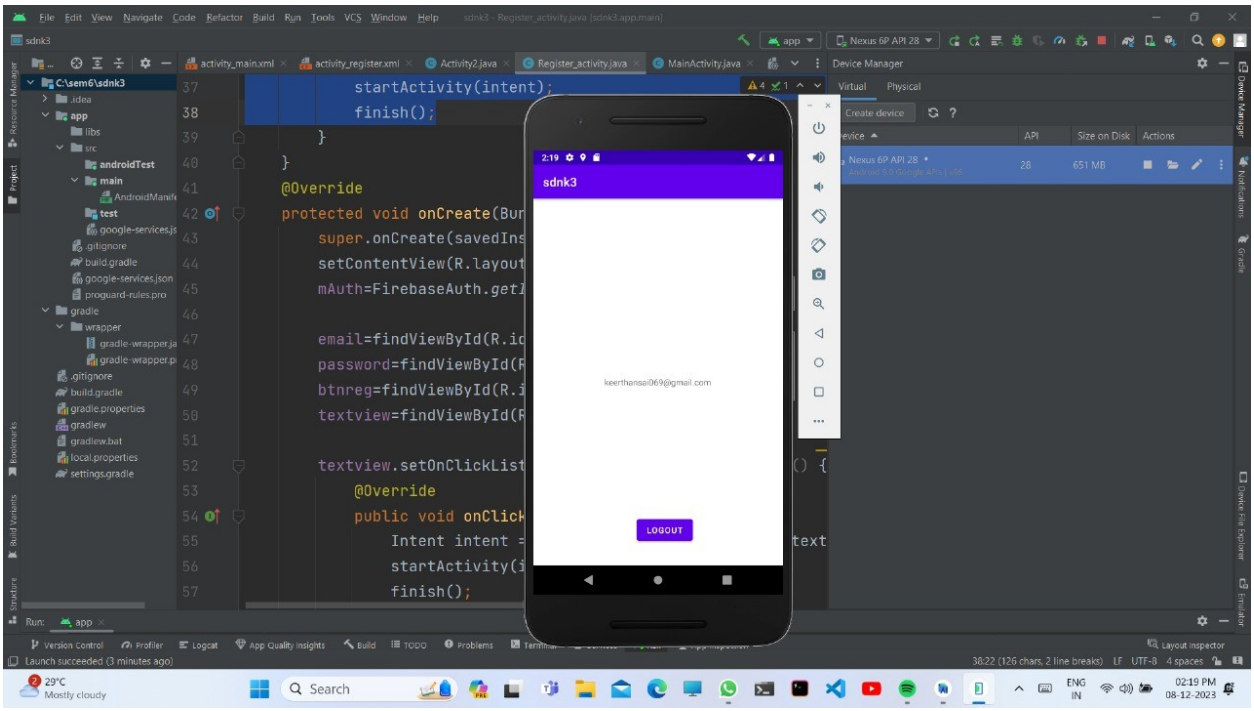

Figure 13: You have logged in successfully

8.2 Sign Up

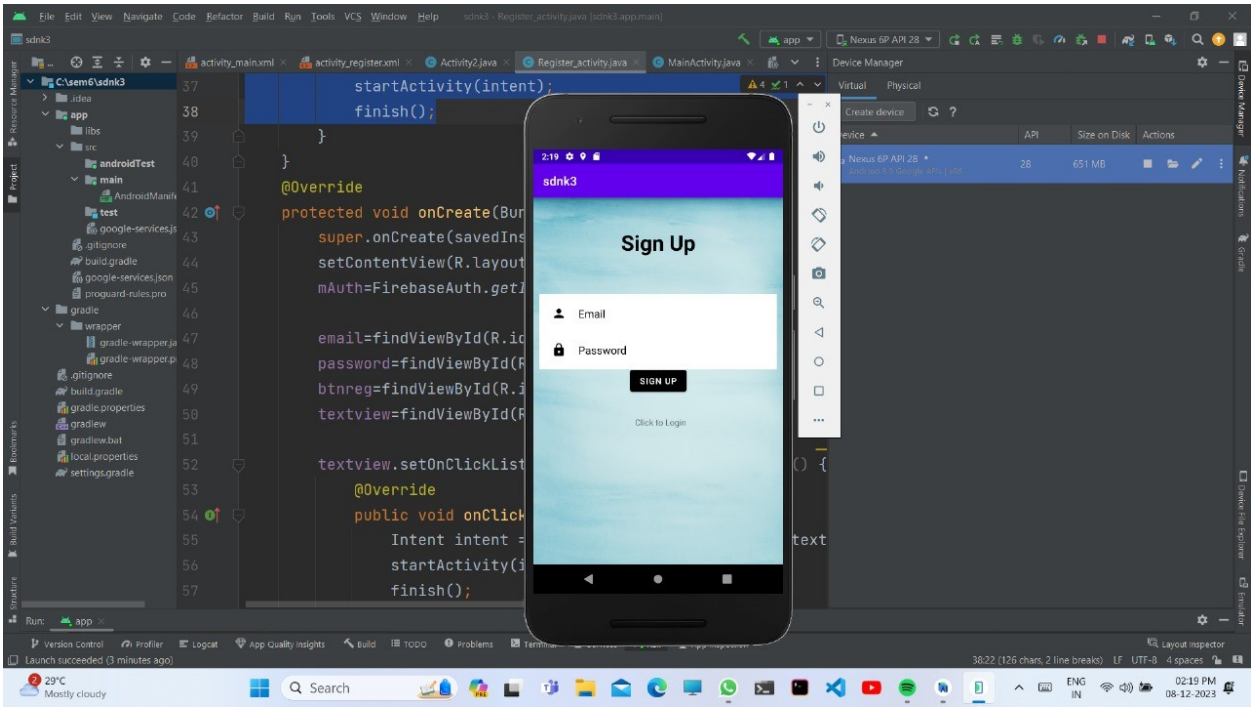

Figure 14: We have to enter a new email which does not have an account in app and enter it's password

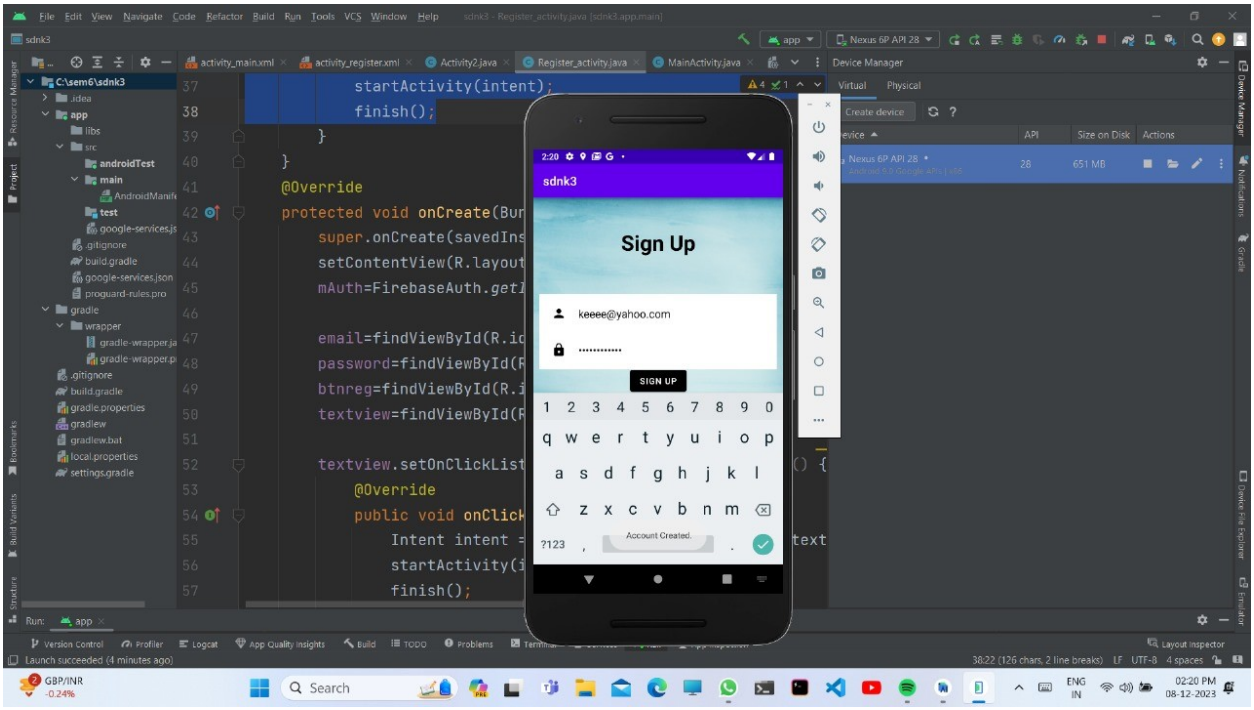

Figure 15: After creating an account there will be a pop up showing account created

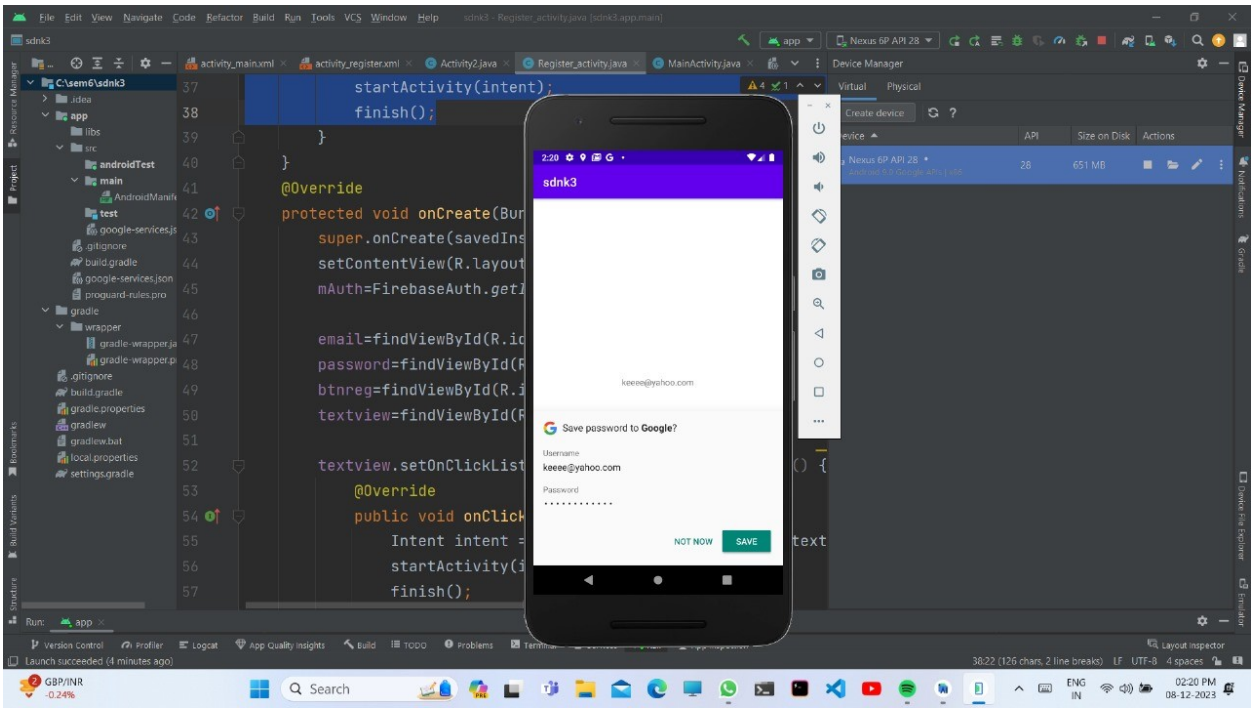

Figure 16: You can either save a password or ignore

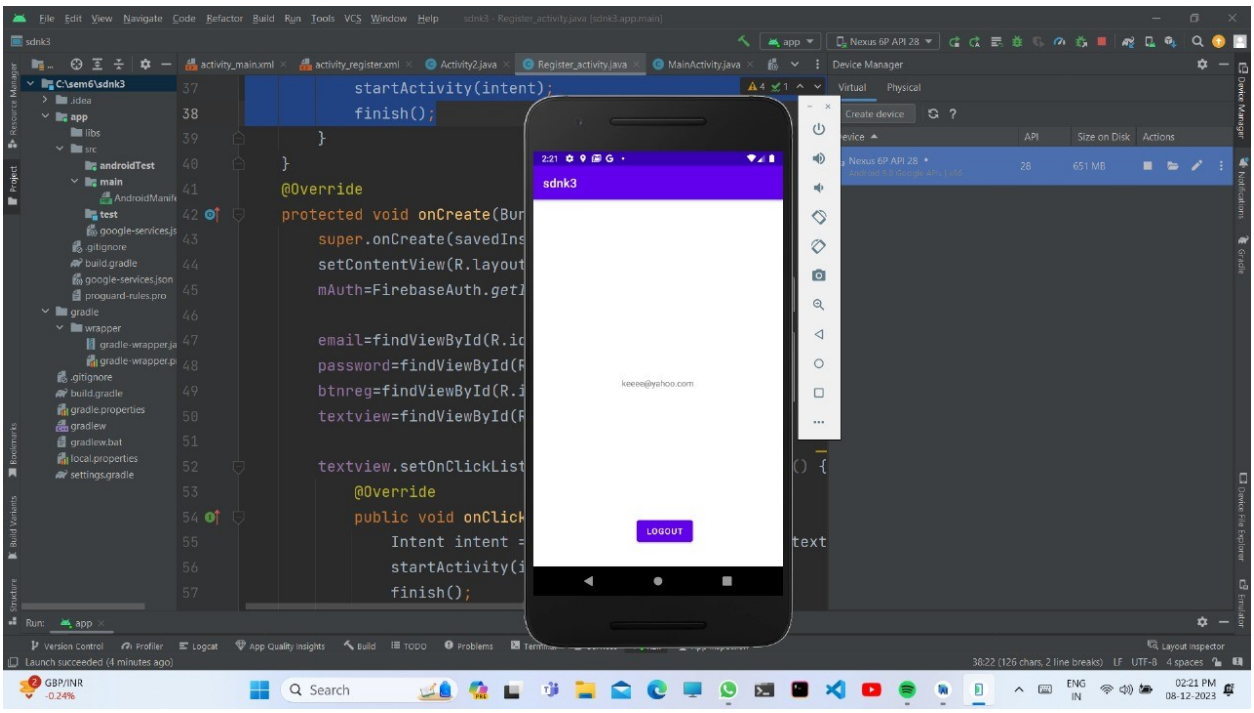

Figure 17: After clicking continue to login you will be redirected to this page,where you have successfully logged in with a new account

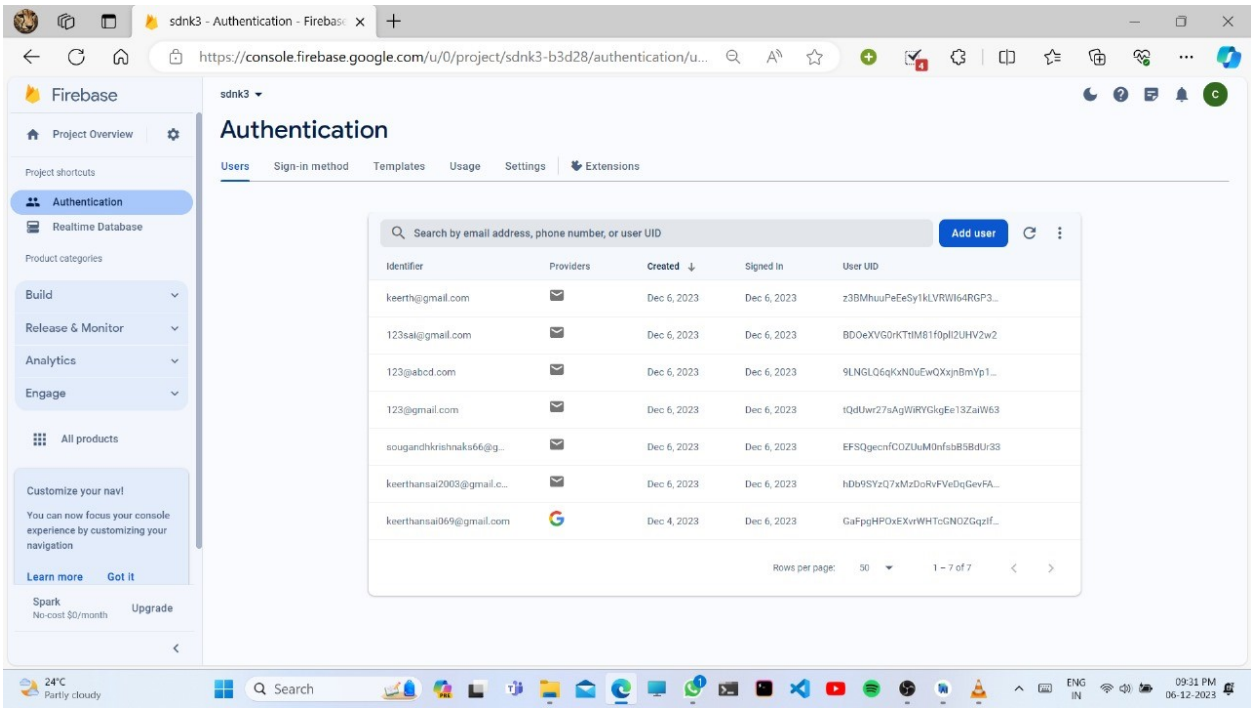

Figure 18: These are the users who logged into our app

9 Documentation

- The project can be found at the following `github link`.
- The earlier proceedings are captured as a video and it can be found at the following `drive link`.

10 Further exploration

10.1 Facebook authentication using Firebase

1. Set Up Firebase Authentication:
   - In the Firebase Console, go to the "Authentication" section.
   - Enable the "Facebook" sign-in method.
2. Create a Facebook App:
   - Go to the `Facebook Developer Console`.
   - Create a new app and configure it. Note down the App ID and App Secret.
3. Configure OAuth Redirect URI:
   - In the Facebook Developer Console, go to your app settings.
   - Under "Facebook Login", add a platform for Android and provide the package name and class name of your Android app.
   - Save the changes.

4. Add Facebook SDK to your Android Studio Project:
    - Open your Android Studio project.
    - Add the Facebook SDK by adding the dependencies to your `build.gradle` files
5. Implement Facebook Sign-In in Your Android App:
    - In your Application class or the main activity, initialize the Facebook SDK with the following code
    - In your login activity XML layout, add a `LoginButton`
    - In your login activity, initialize the `LoginButton` and set up a callback manager
    - Register the `CallbackManager` in the onCreate method and set the required permissions
    - Override the `onActivityResult` method to forward results to the `CallbackManager`
    - In the `onSuccess` method of the `FacebookCallback`, use the Facebook access token to authenticate with Firebase
6. Test Your App:
    - Run your app and test the Facebook authentication flow

10.2 Documentation

- This can be found at the following github link.

10.3 Remarks

I tried to integrate login with Facebook with my main app but it ended up with some errors. It was showing it needed a permission from meta developers in order to continue with the proceedings,so I did not integrate it and kept as a separate project. If you try to sign up it will end up showing your account is suspended contact Facebook team. If you try to login it will end up showing an error telling you cannot login privacy policy issue. The same is depicted in the following snapshots.

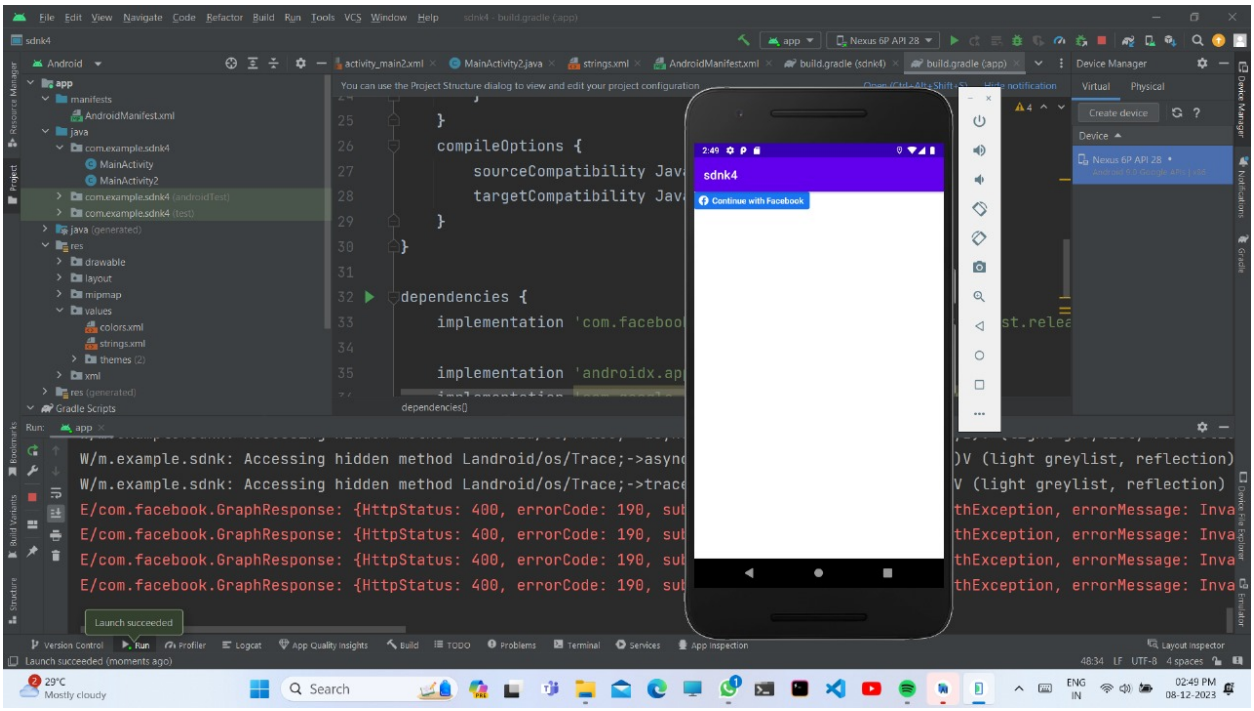

Figure 19

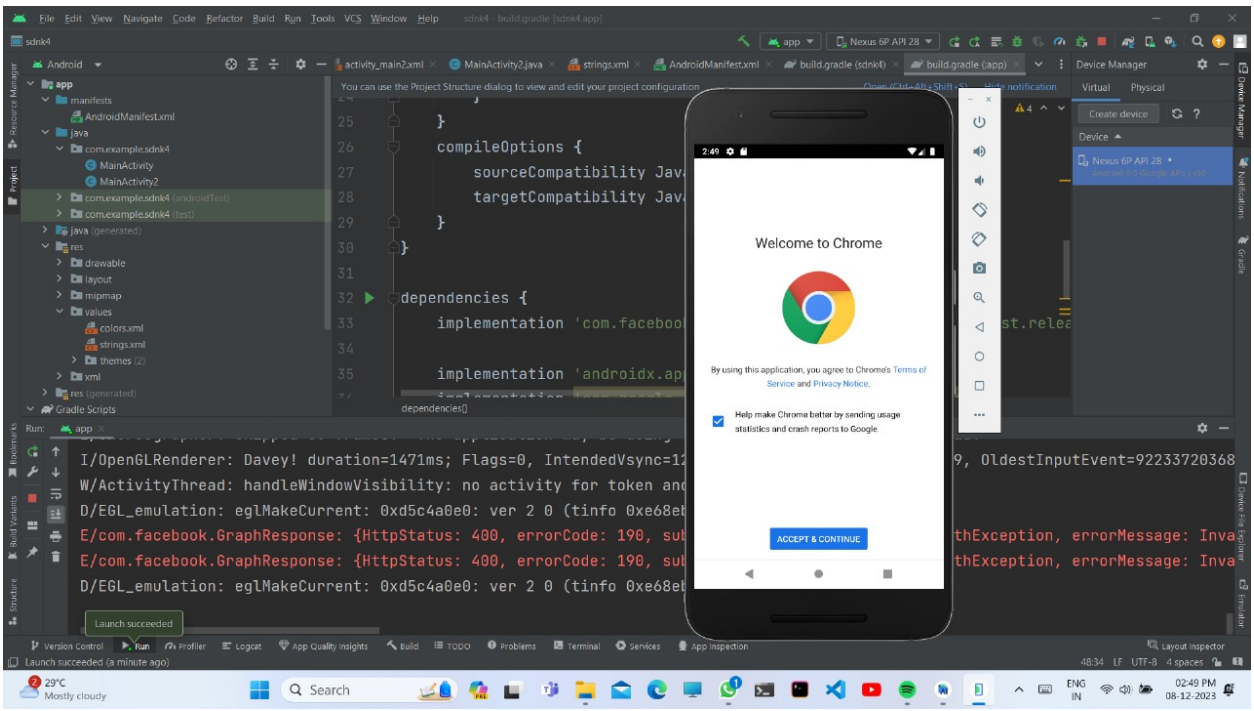

Figure 20

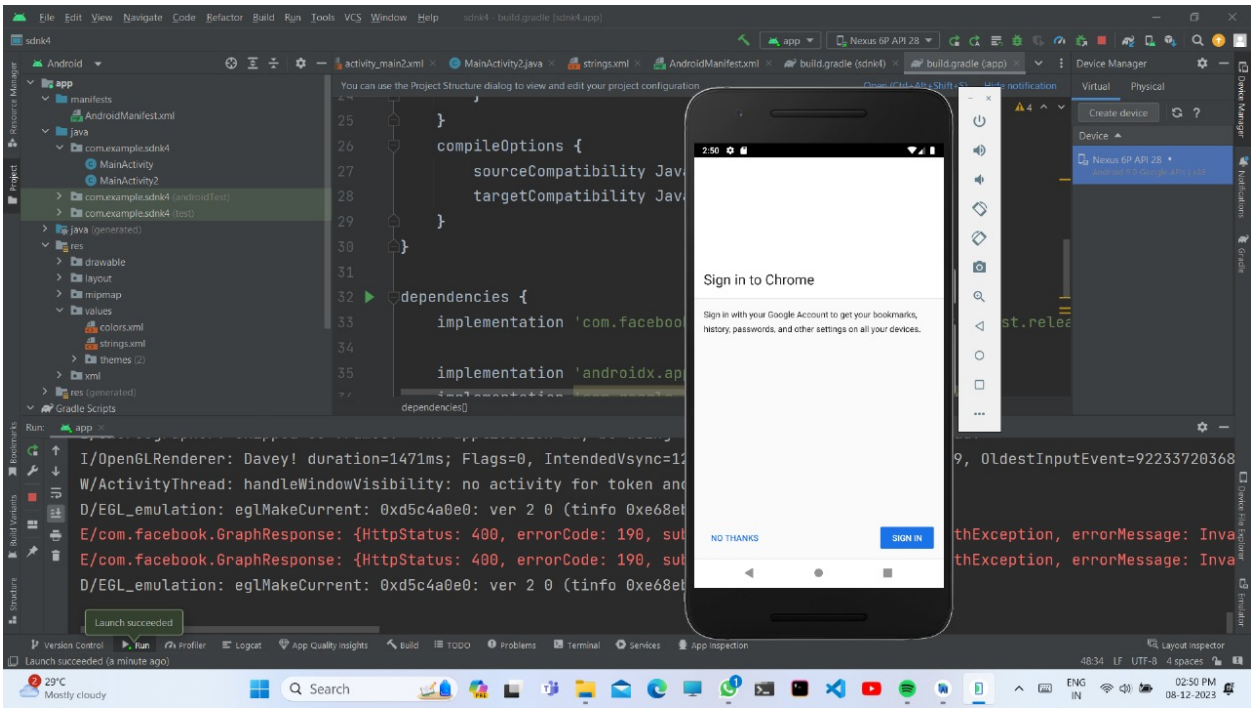

Figure 21

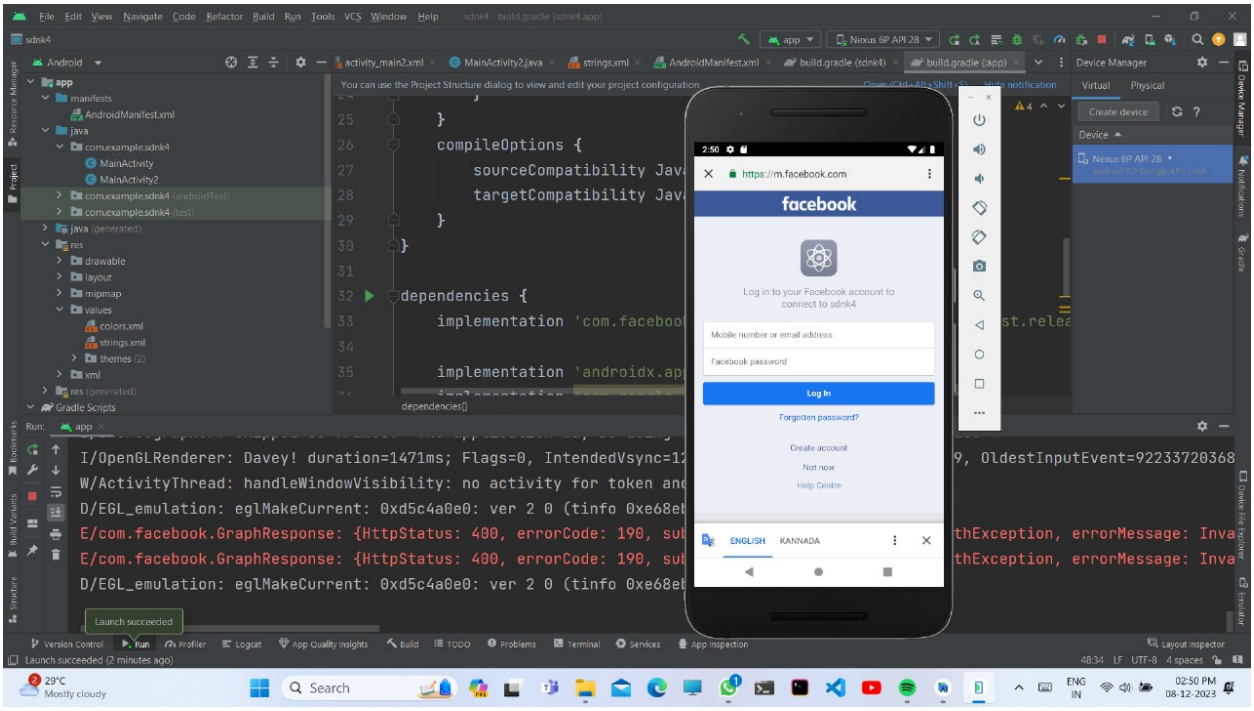

Figure 22

11 Feedback form

- Please provide your valuable feedback by clicking the following link