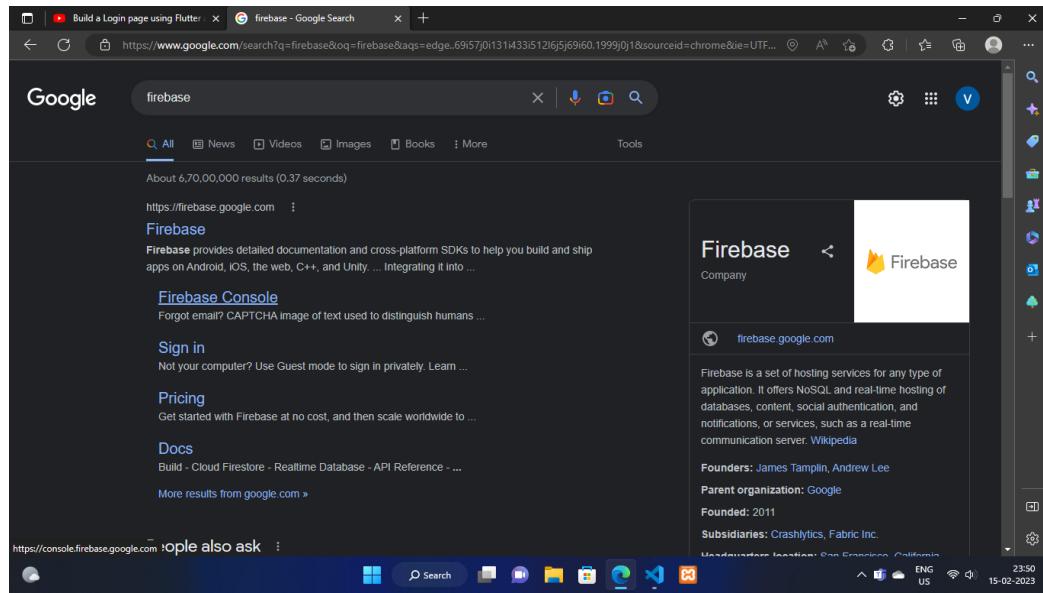


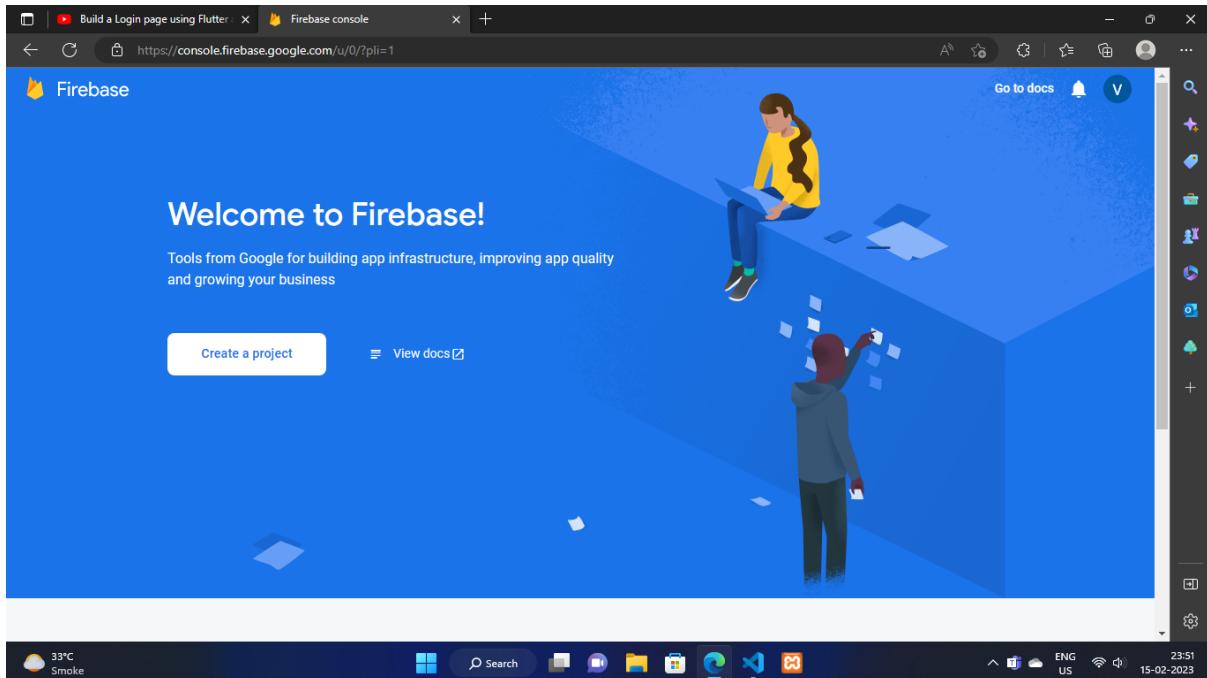
# Flutter Application With FireBase Login

## Step 1 : Finding Firebase



- a. Search Firebase on google
- b. Go to first link firebase.google.com

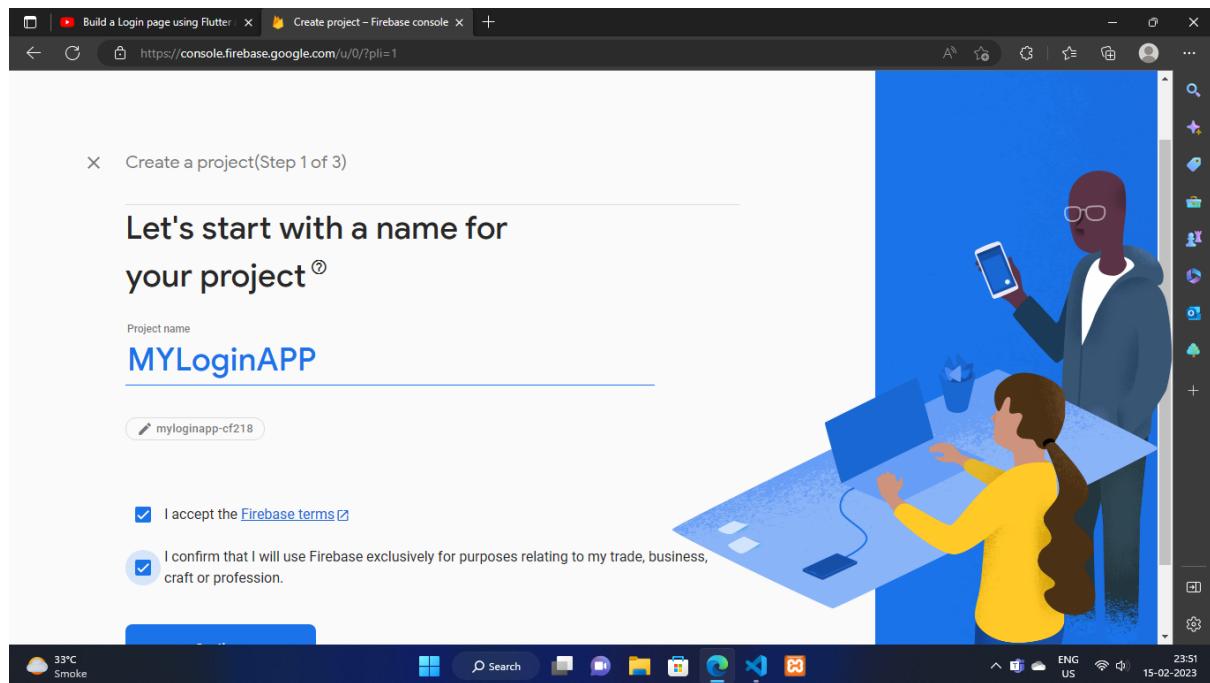
# Step 2 : Create A Firebase Project



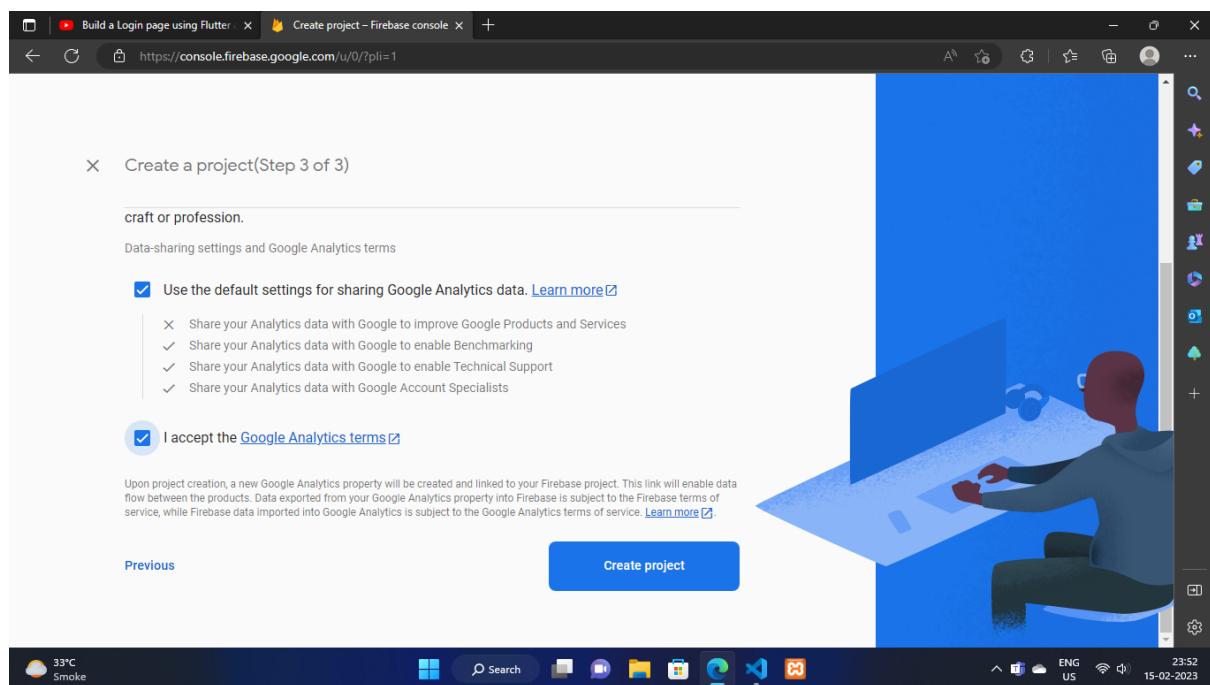
## Part 1 :

- a. First login with your account
- b. Then you'll be able to see the above page
- c. Click on **create a project** as seen above

## Part 2 :

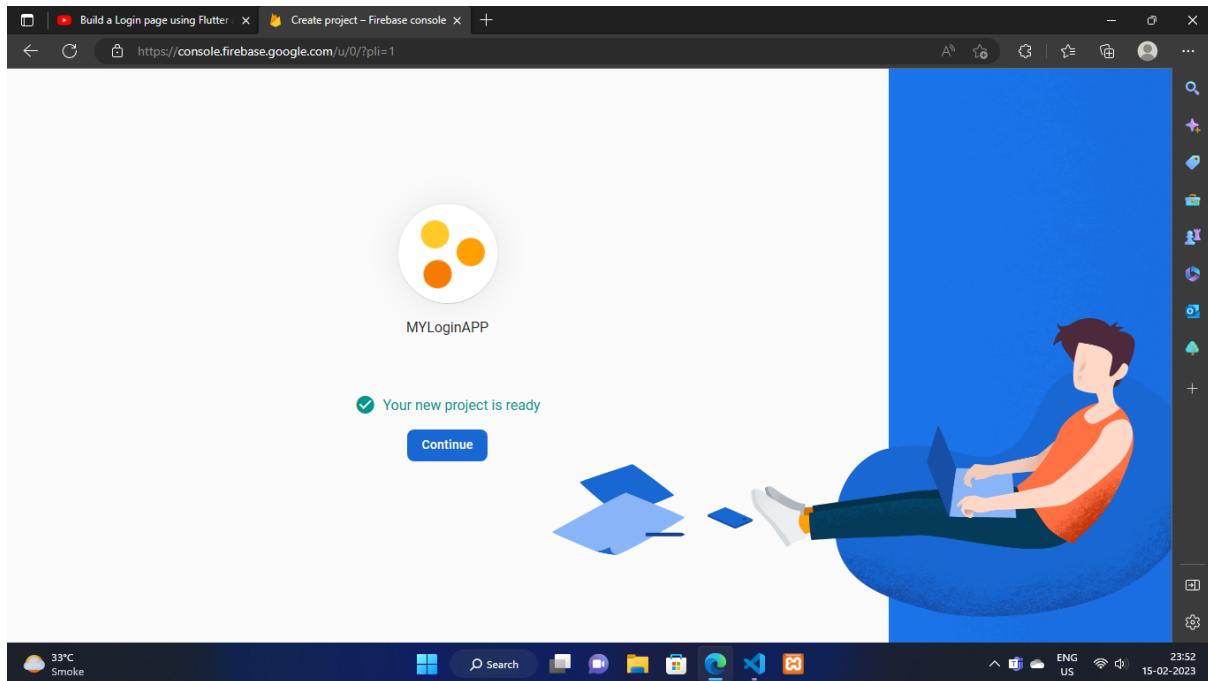


- Give Any Name To Your Project
- Fill all the checkboxes
- Click on continue button

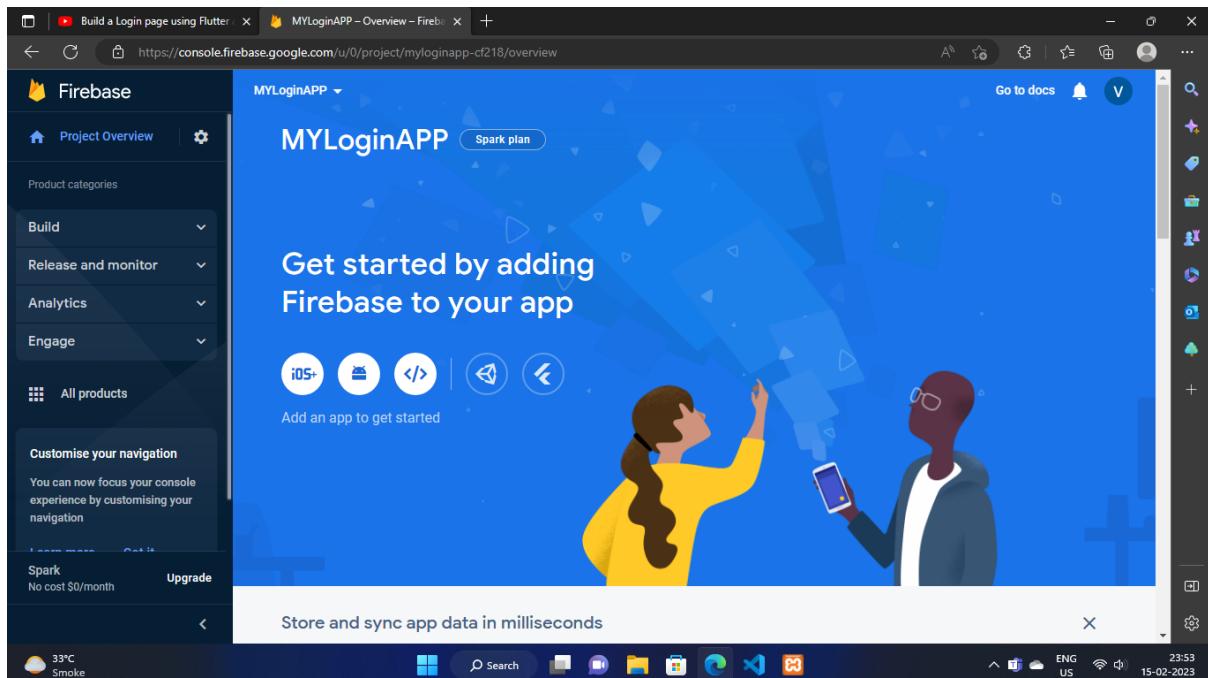


- Fill the check box as seen above and create project

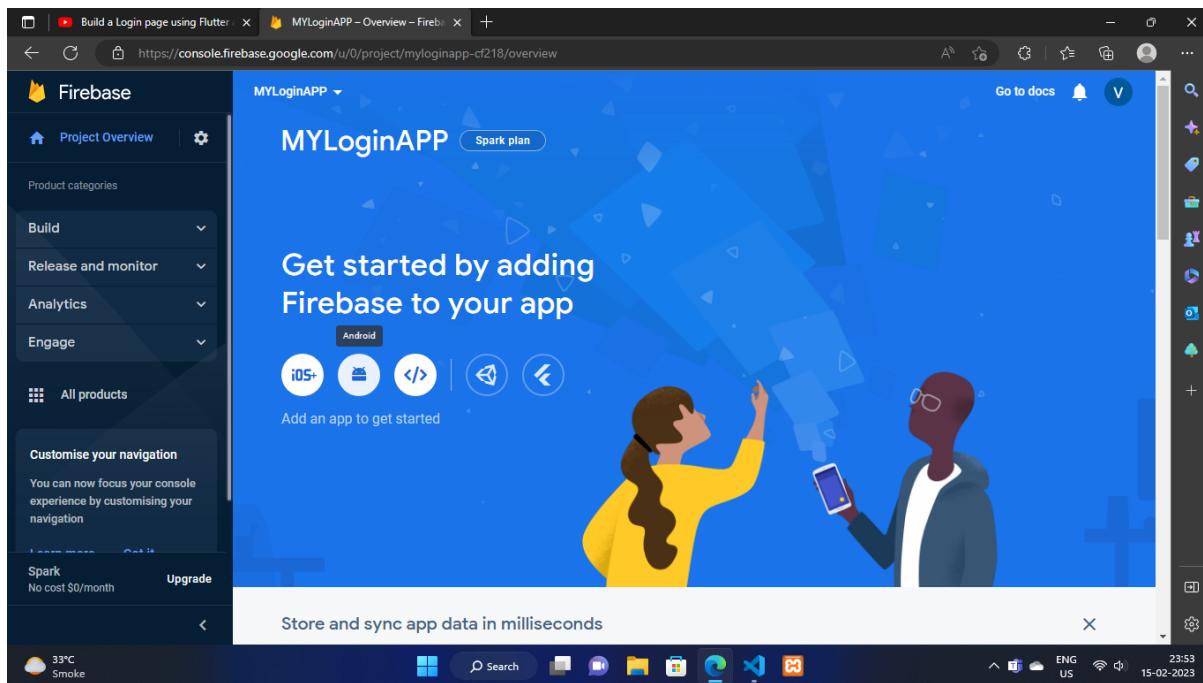
## Part 3:



When your project is created you'll be redirected to this screen , now click on continue button



After successful creation of your app you'll see a dashboard like this



a. Click on android picture as seen above

```

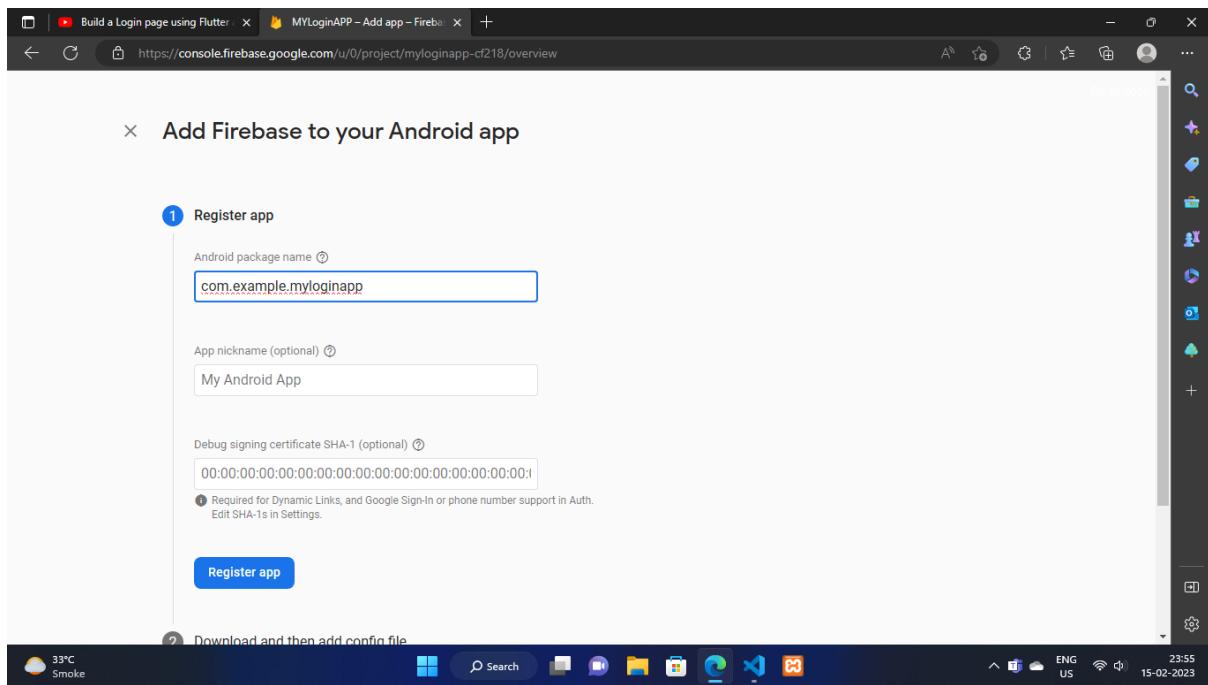
File Ed Selection View Go Run Terminal Help
build.gradle - myloginapp - Visual Studio Code

EXPLORER
MYLOGINAPP
  .dart_tool
  .idea
  android
    .gradle
    app
      src
        build.gradle
        gradle
        .gitignore
        build.gradle
        gradle.properties
        gradlew
        gradlew.bat
        local.properties
        myloginapp_android...
        settings.gradle
      ios
      lib
      linux
      macos
      test
      web
      windows
      .gitignore
    OUTLINE
    TIMELINE
    DEPENDENCIES
    JAVA PROJECTS

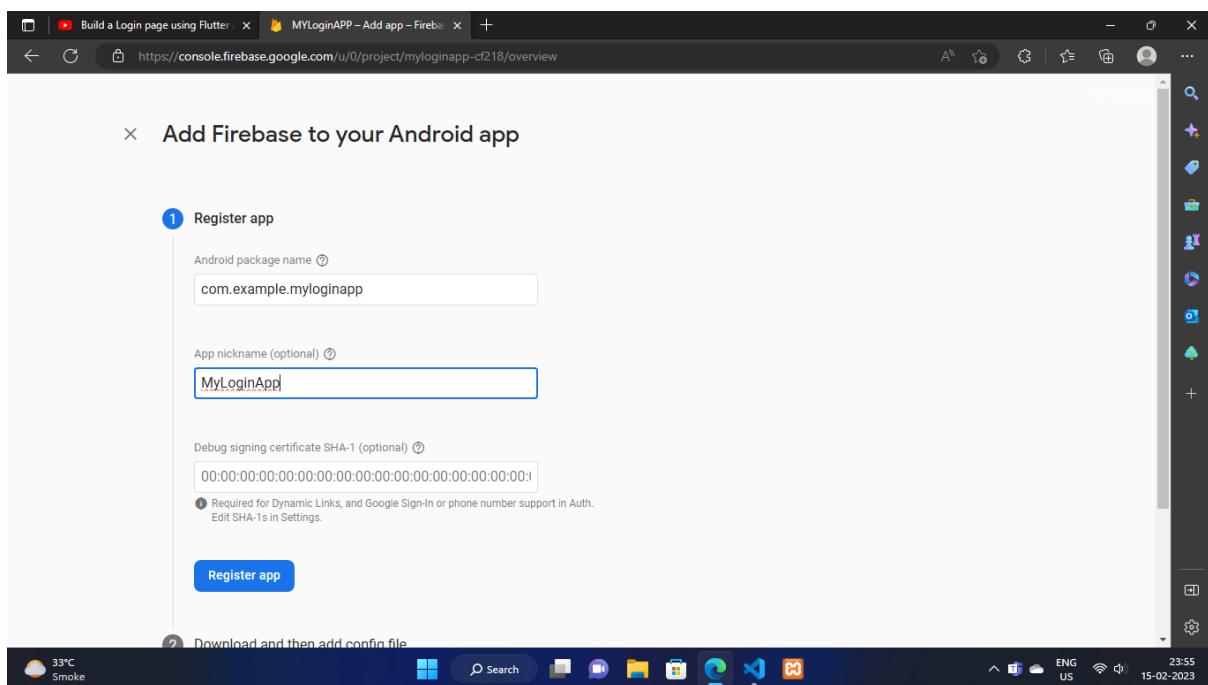
build.gradle
  android > app > build.gradle
    jvmTarget = '1.8'
    ...
    sourceSets {
      main.java.srcDirs += 'src/main/kotlin'
    }
    defaultConfig {
      // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/application-id.html).
      applicationId "com.example.myloginapp"
      // You can update the following values to match your application needs.
      // For more information, see: https://docs.flutter.dev/deployment/android#reviewing-the-gradle-build-configuration.
      minSdkVersion flutter.minSdkVersion
      targetSdkVersion flutter.targetSdkVersion
      versionCode flutterVersionCode.toInt()
      versionName flutterVersionName
    }
    buildTypes {
      release {
        // TODO: Add your own signing config for the release build.
        // Signing with the debug keys for now, so `flutter run --release` works.
        signingConfig signingConfigs.debug
      }
      flutter {
        source '...'
      }
    }
  dependencies {
  }

```

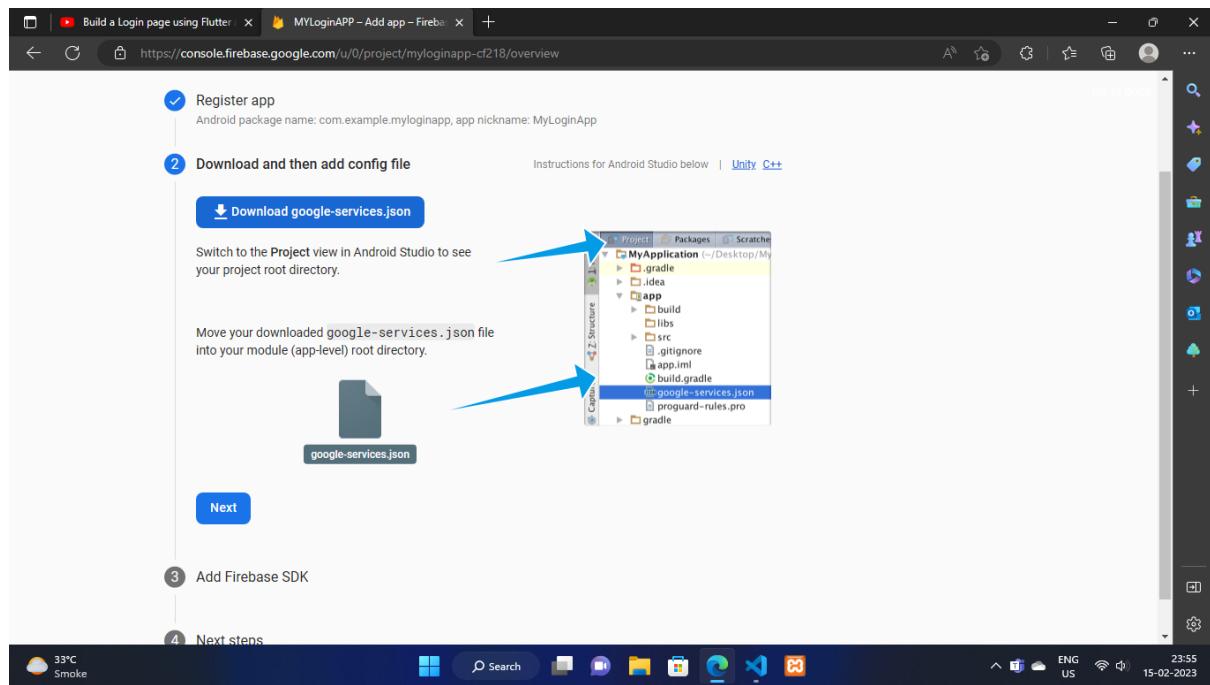
b. Now go to your app in (android->app->build.gradle) and copy the application id of your app



c. Now paste the name to this page



- d. Give any random name in this textfield
- e. Then click on Register app button



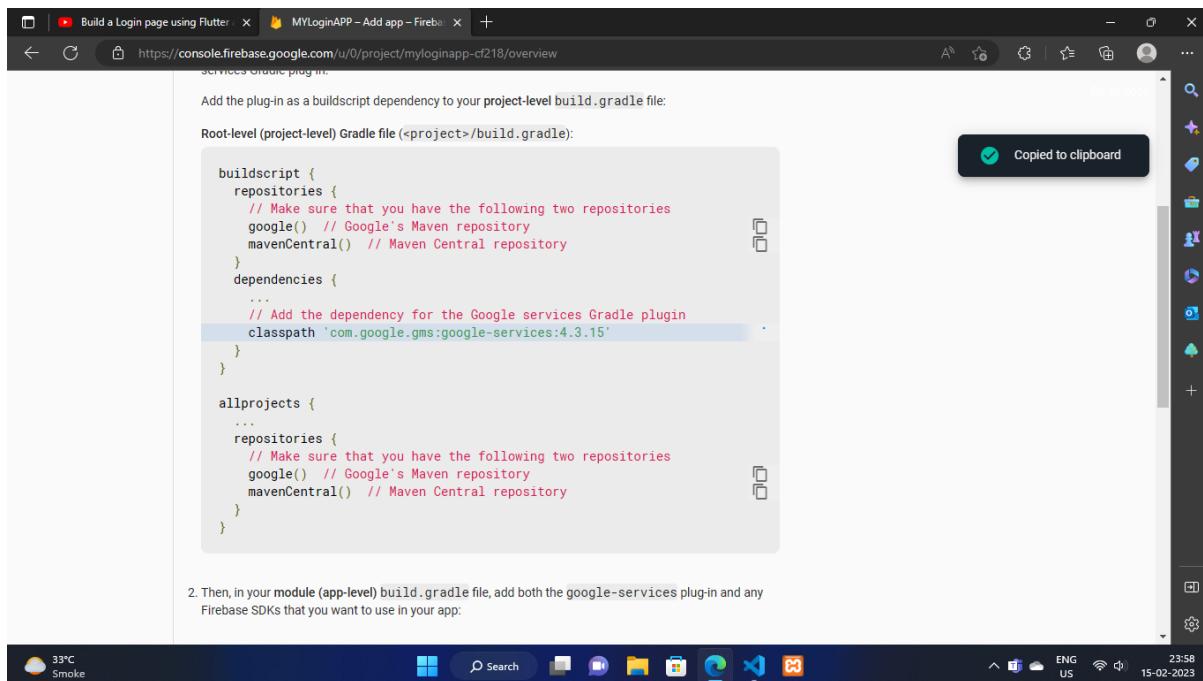
f. Now click on download button a json file will be downloaded

```

{
  "project_info": {
    "project_number": "269709872177",
    "project_id": "myloginapp-cf218",
    "storage_bucket": "myloginapp-cf218.appspot.com"
  },
  "client": [
    {
      "client_info": {
        "mobilesdk_app_id": "1:269709872177:android:4031b489fc1eb700a69910",
        "android_client_info": {
          "package_name": "com.example.myloginapp"
        }
      },
      "oauth_client": [
        {
          "client_id": "269709872177-edch5lf9g7r9t385r606nsq5s9jd0qp.apps.googleusercontent.com",
          "client_type": 3
        }
      ],
      "api_key": [
        {
          "current_key": "AIzaSyBYQBI_aHxQxYc31ijcFY8IMZKvPUQr1Q"
        }
      ],
      "services": {
        "appinvite_service": {
          "other_platform_oauth_client": [
            {
              "client_id": "269709872177-edch5lf9g7r9t385r606nsq5s9jd0qp.apps.googleusercontent.com",
              "client_type": 3
            }
          ]
        }
      }
    }
  ]
}

```

g. Now put the json file inside (android->app) folder  
 h. Go to firebase console and click on next button of 2nd step



```

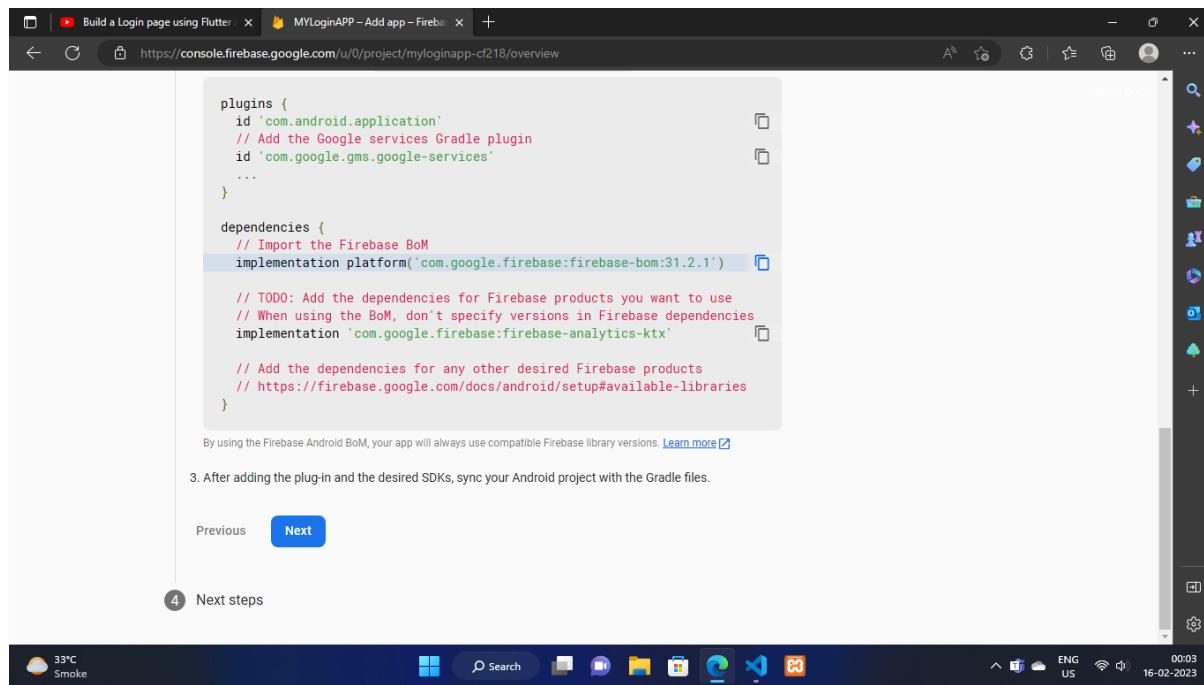
buildscript {
    ext.kotlin_version = '1.7.10'
    repositories {
        google()
        mavenCentral()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:7.2.0'
        classpath 'com.google.gms:google-services:4.3.15'
        classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"
    }
}

allprojects {
    repositories {
        google()
        mavenCentral()
    }
}
rootProject.buildDir = '../build'
subprojects {
    project.buildDir = "${rootProject.buildDir}/${project.name}"
}
subprojects {
    project.evaluationDependsOn(':app')
}

task clean(type: Delete) {
    delete rootProject.buildDir
}

```

j. Now paste it in android->build.gradle file as seen above

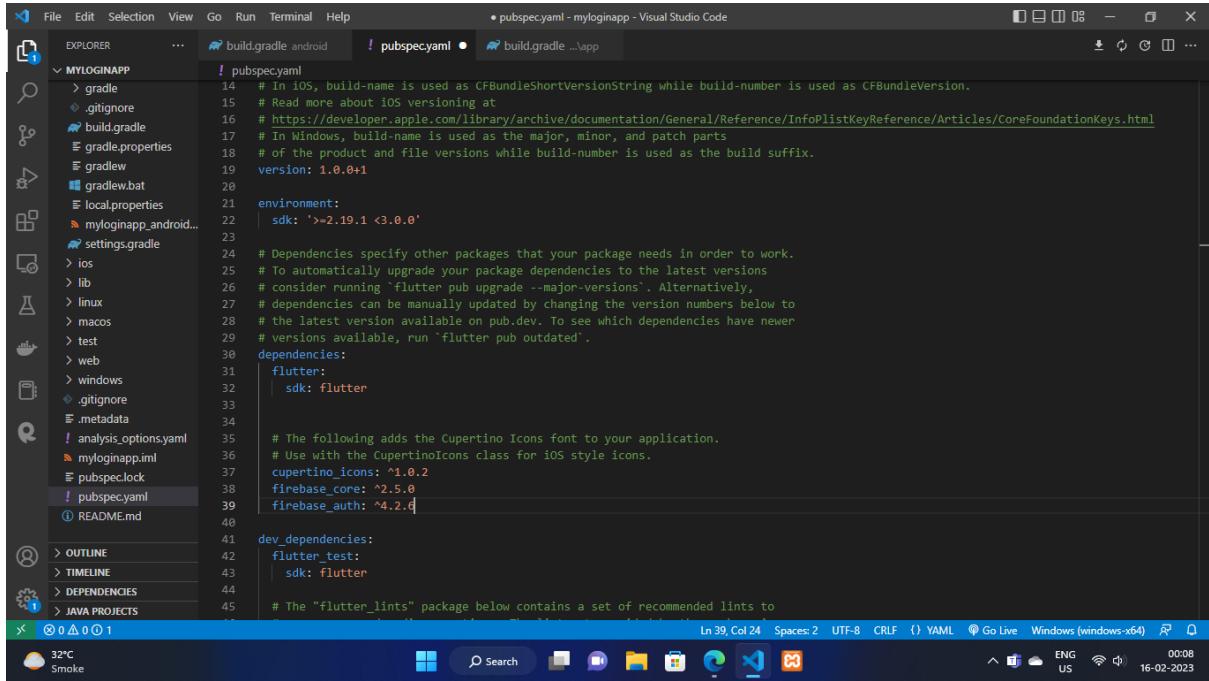


K. Now copy the both implementation thing and paste it in android->app->build.gradle as seen below

L. Now copy both plugins thing and paste it in the same file as mentioned above

# Step 4 :

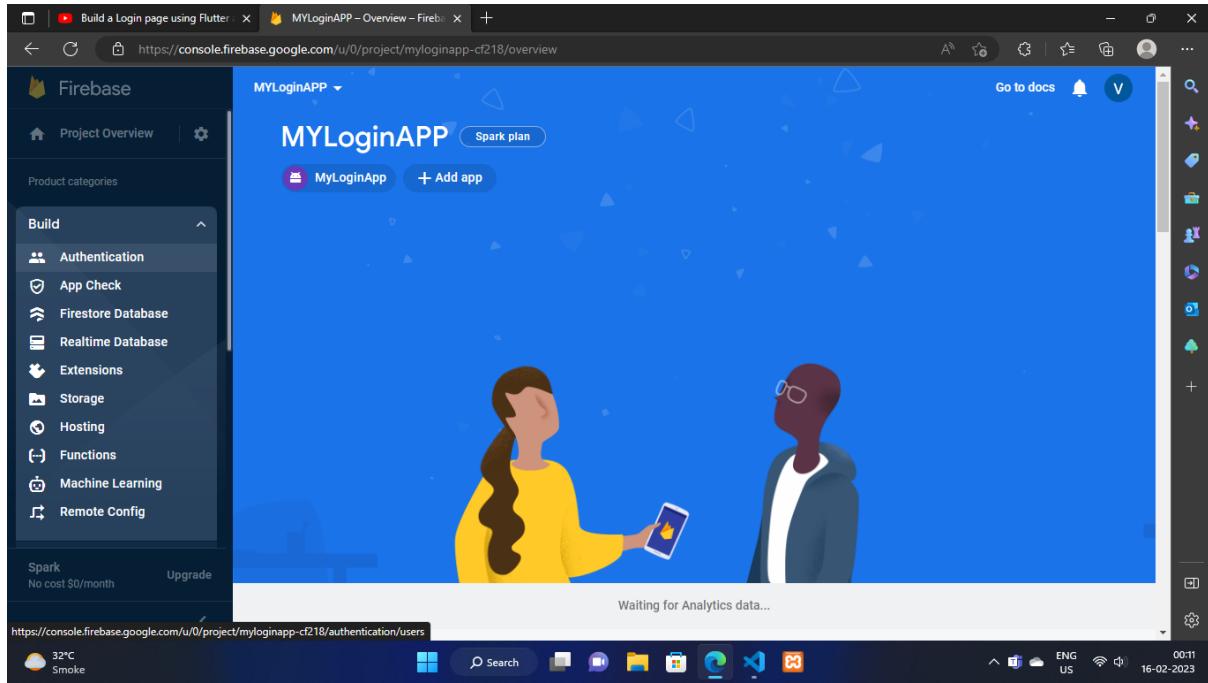
## Part 1 : Adding plugins



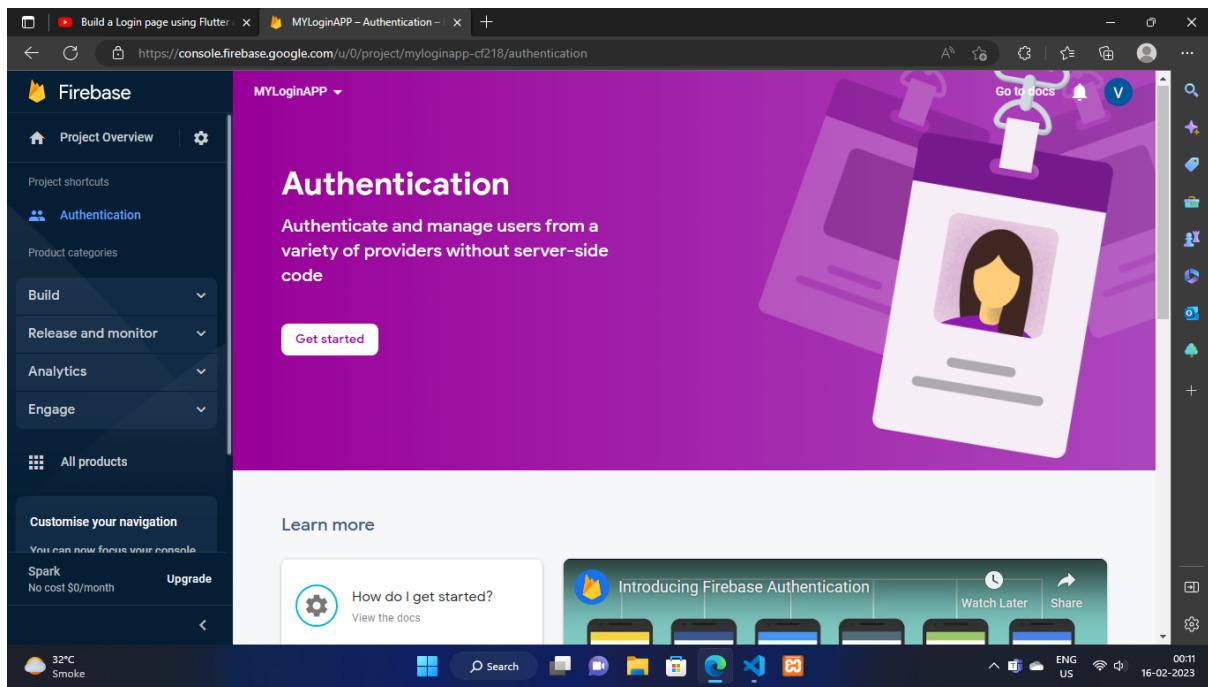
```
version: 1.0.0+1
environment:
  sdk: '>=2.19.1 <3.0.0'
dependencies:
  flutter:
    sdk: flutter
  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  firebase_core: ^2.5.0
  firebase_auth: ^4.2.6
dev_dependencies:
  flutter_test:
    sdk: flutter
# The "flutter_lints" package below contains a set of recommended lints to
```

- Add both the plugin seen above in pubspec.yaml file

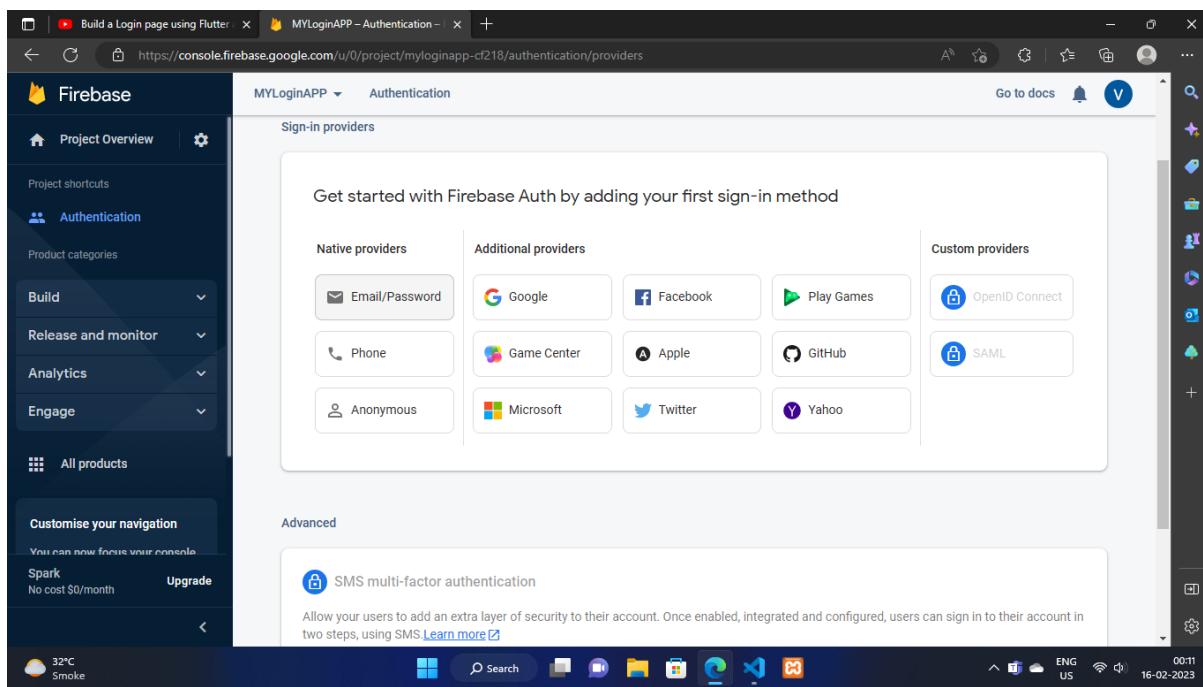
# Part 2 : Configuring Firebase



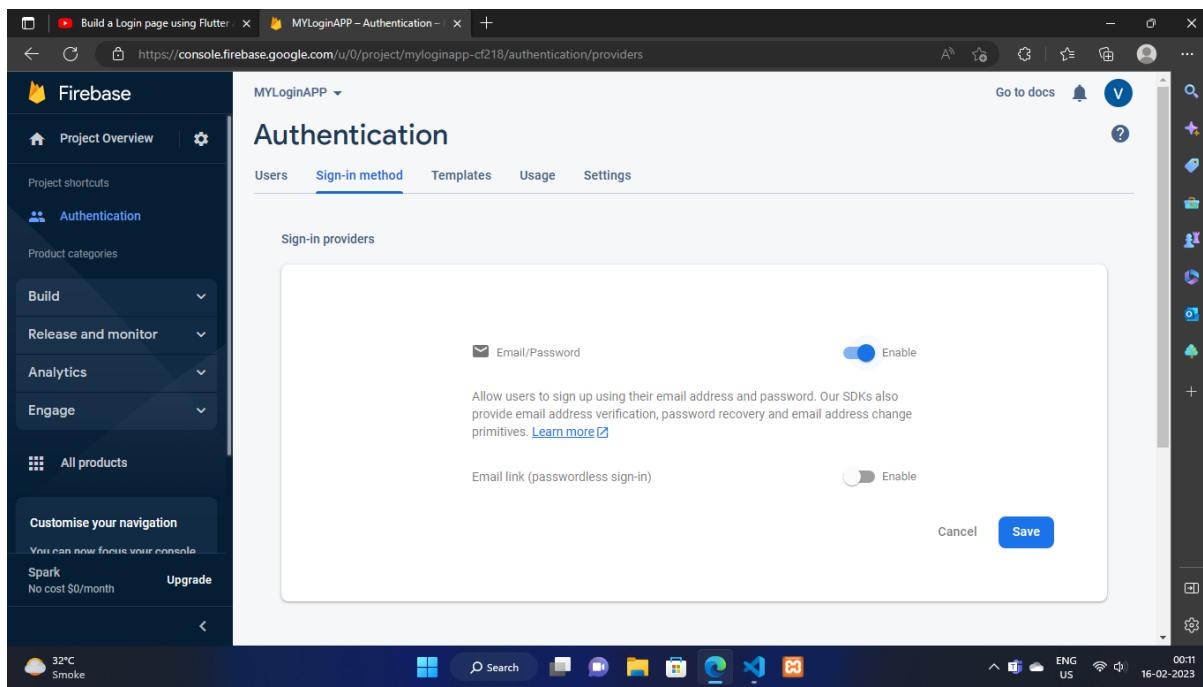
- a. Click on the build available in left side bar and then click on authentication as show above



- b. Now you should be able to see the page shown above
- c. Click on Get Started Button you can see that button



- d. Now you will be redirected to this page
- e. Now click on the Email/Password button as seen in above image



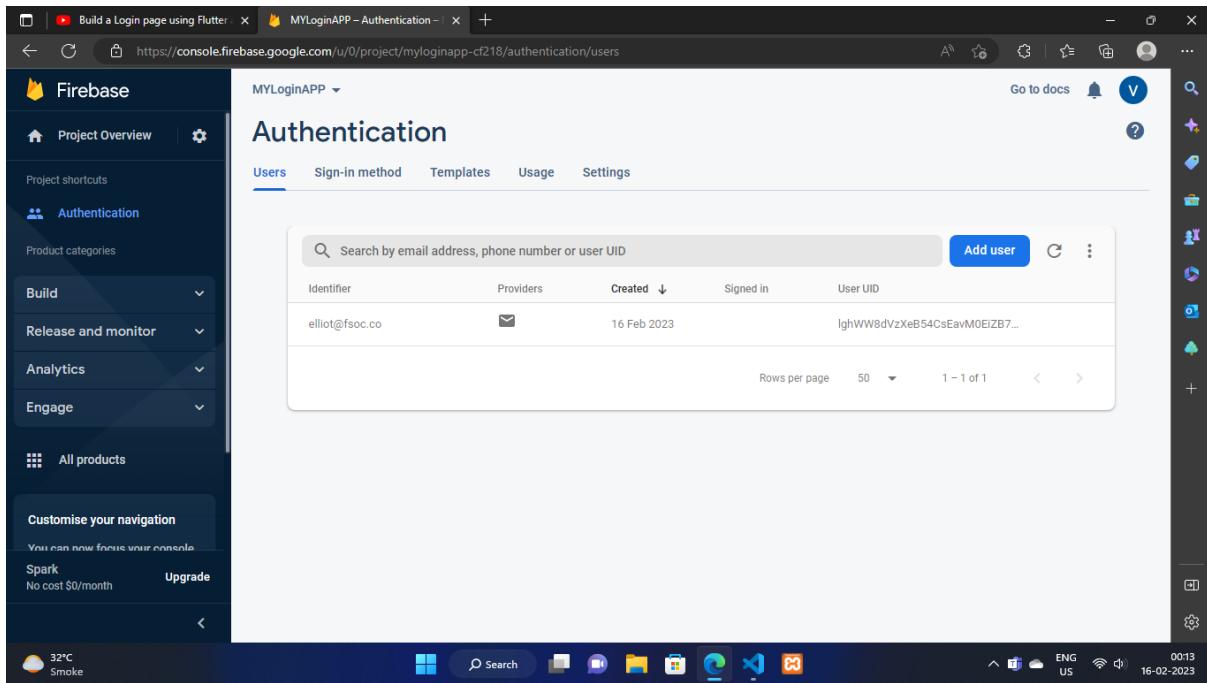
- f. Now you will be redirected to this page now just click on the icon to enable it and then click on save button

The screenshot shows the Firebase Authentication Sign-in method page. The left sidebar includes Project Overview, Authentication (selected), and other products like Spark and Blaze. The main content area is titled 'Authentication' and shows the 'Sign-in method' tab selected. It displays a table for 'Sign-in providers' with one row for 'Email/Password' which is 'Enabled'. Below this, there's an 'Advanced' section with a note about SMS multi-factor authentication and a message about MFA and Identity Platform.

- g. If all the processes are successful you'll be able to see this page
- h. Now click on users under Authentication and you will be redirected to page as shown int the below image

The screenshot shows the Firebase Authentication Users page. The left sidebar includes Project Overview, Authentication (selected), and other products like Spark and Blaze. The main content area is titled 'Authentication' and shows the 'Users' tab selected. It displays a table with columns for Identifier, Providers, Created, Signed in, and User ID. A message at the bottom states 'No users for this project yet'.

- i. Now click on add user button and add a user with email and password (remember the pass as the Firebase doesn't show it afterwards)
- j. You should be able to see the the added user as shown in image below



The screenshot shows the Firebase Authentication page for a project named 'MYLoginAPP'. The 'Users' tab is selected. A table displays one user entry:

Identifier	Providers	Created	Signed in	User UID
elliot@fsoc.co	✉	16 Feb 2023	Signed in	IghWW8dVzXeB54CsEavM0EiZB7...

Below the table, there are pagination controls: 'Rows per page' set to 50, '1 ~ 1 of 1', and navigation arrows. The top right of the page has a 'Go to docs' link, a user profile icon, and a help icon. The left sidebar includes sections for Project Overview, Authentication (which is currently selected), Product categories, Build, Release and monitor, Analytics, Engage, and All products. A 'Customise your navigation' section is also present.

- k. Now change the thing shown below

TO ----->

```

buildTypes {
    release {
        // TODO: Add your own signing config for the release build.
        // Signing with the debug keys for now, so `flutter run --release` works.
        signingConfig signingConfigs.debug
    }
}

```

FAILURE: Build failed with an exception.

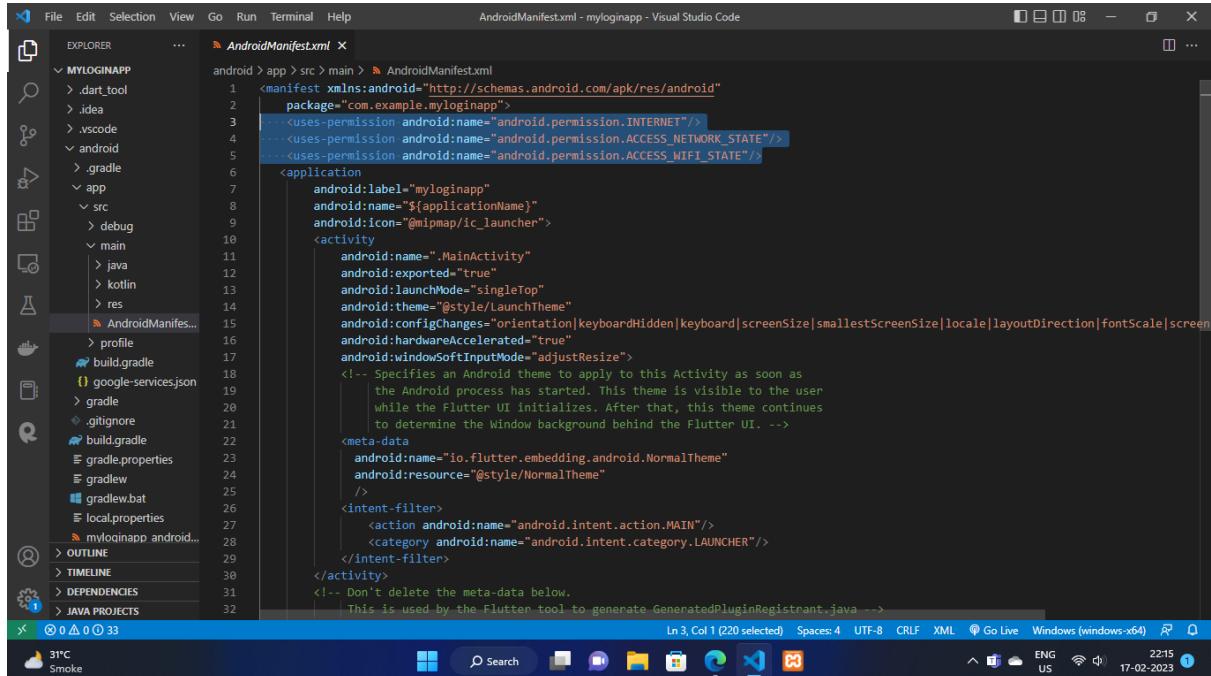
```

minSdkVersion 21]
targetSdkVersion flutter.targetSdkVersion
versionCode flutterVersionCode.toInt()
versionName flutterVersionName

```

FAILURE: Build failed with an exception.

I. Now go to  
 android->app->src->main->Androidmanifest.xml file  
 and add the things shown below in the image



```
File Edit Selection View Go Run Terminal Help
AndroidManifest.xml - myloginapp - Visual Studio Code
EXPLORER ... ▾ AndroidManifest.xml
MYLOGINAPP
> .dart_tool
> .idea
> .vscode
> android
> .gradle
> app
> src
> debug
> main
> java
> kotlin
> res
> AndroidManifest.xml
> profile
build.gradle
google-services.json
> gradle
> .gitignore
build.gradle
gradle.properties
gradlew
gradlew.bat
local.properties
myloginapp android...
> OUTLINE
> TIMELINE
> DEPENDENCIES
> JAVA PROJECTS
Ln 3, Col 1 (220 selected) Spaces: 4 UTF-8 CRLF XML Go Live Windows (windows-x64)
31C Smoke
Search
22:15 17-02-2023
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myloginapp">
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
</manifest>
<application
    android:label="myloginapp"
    android:name="${applicationName}"
    android:icon="@mipmap/ic_launcher">
    <activity
        android:name=".MainActivity"
        android:exported="true"
        android:launchMode="singleTop"
        android:theme="@style/LaunchTheme"
        android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale|layoutDirection|fontScale|screenLayout"/>
        <!-- Specifies an Android theme to apply to this Activity as soon as
            the Android process has started. This theme is visible to the user
            while the Flutter UI initializes. After that, this theme continues
            to determine the Window background behind the Flutter UI. -->
        <meta-data
            android:name="io.flutter.embedding.android.NormalTheme"
            android:resource="@style/NormalTheme"/>
        </meta-data>
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>
            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
    <!-- Don't delete the meta-data below.
        This is used by the Flutter tool to generate GeneratedPluginRegistrant.java -->
```

# Done With Configs

## Now

“ Keep Calm And  
<Code> On ”