Laboratorijska vežba 5
Firebase Firestore

Opis projekta

Projekat predstavlja aplikaciju koja služi za dodavanje POI-a (Points of Interest) na Firebase Firestore bazu podataka. Arhitektura aplikacije je slična vežbi 3 ali se umesto Room baze podataka koristi Firebase Firestore implementiran servis (naziv za klasu koja povezuje aplikaciju sa spoljnim izvorima, ne Android servis) kojom, uz korišćenje Flow-a dodaje i preuzima podatke sa Firestore-a. Potrebno je kreirati projekat na Firebase-u a zatim i Firestore bazu a zatim je povezati sa postojećim kodom aplikacije. Uputstvo za povezivanje se nalazi u nastavku pripreme za laboratorijsku vežbu.

Opis postojeće strukture

Implementirani su sledeći ekrani:

1. **PoiListScreen** – prikazuje listu Poi-a koji su preuzeti iz baze podataka. Klikom na dugme View prelazi se na ekran za prikaz sadržaja Poi-a.
2. **ViewPoiScreen** – prikazuje sadržaj Poi-a.
3. **AddPoiScreen –** omogućuje dodavanje novih Poi-a.

Implementirani su sledeći ViewModel-i:

1. **PoiViewModel** – komunicira sa servisom za pribavljanje podataka sa *Firestore*-a korišćenjem Flow-a.
2. **EditViewModel** – koristi se za prenos unetih podataka o Poi-u.

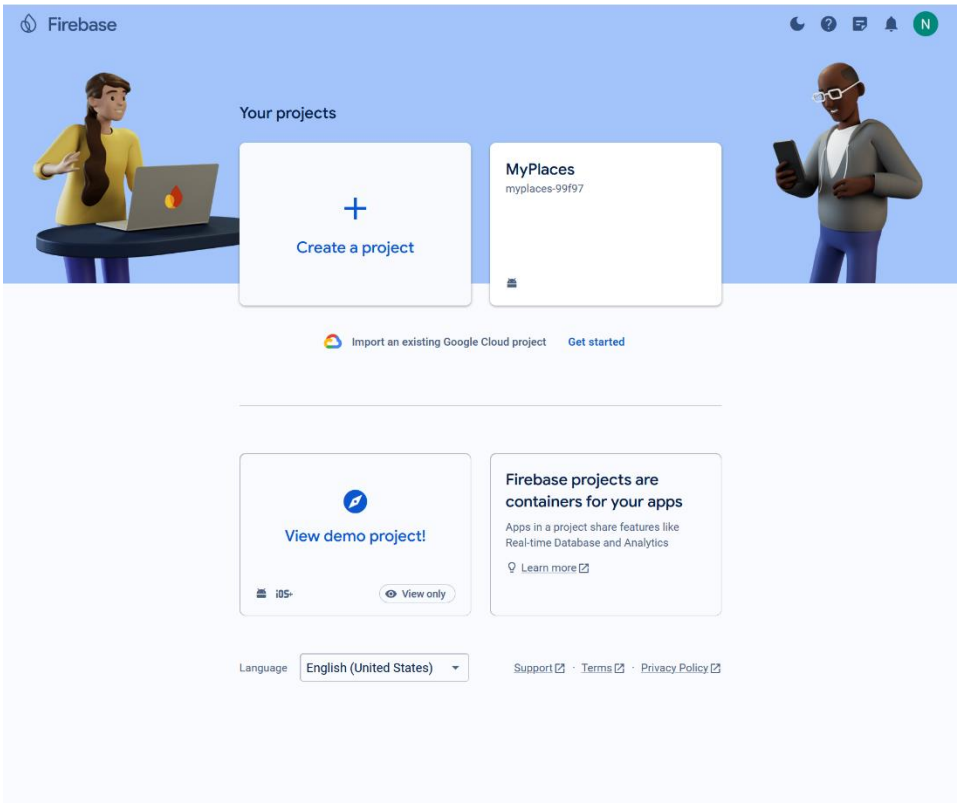
Model podataka je implementiran u data klasi Poi i sadržni ime i adresu Poi-a. Preostala dva podatka *created_at* i *id* su generisani od strane *Firebase Firestore*-a prilikom upisa.

ZADATAK:

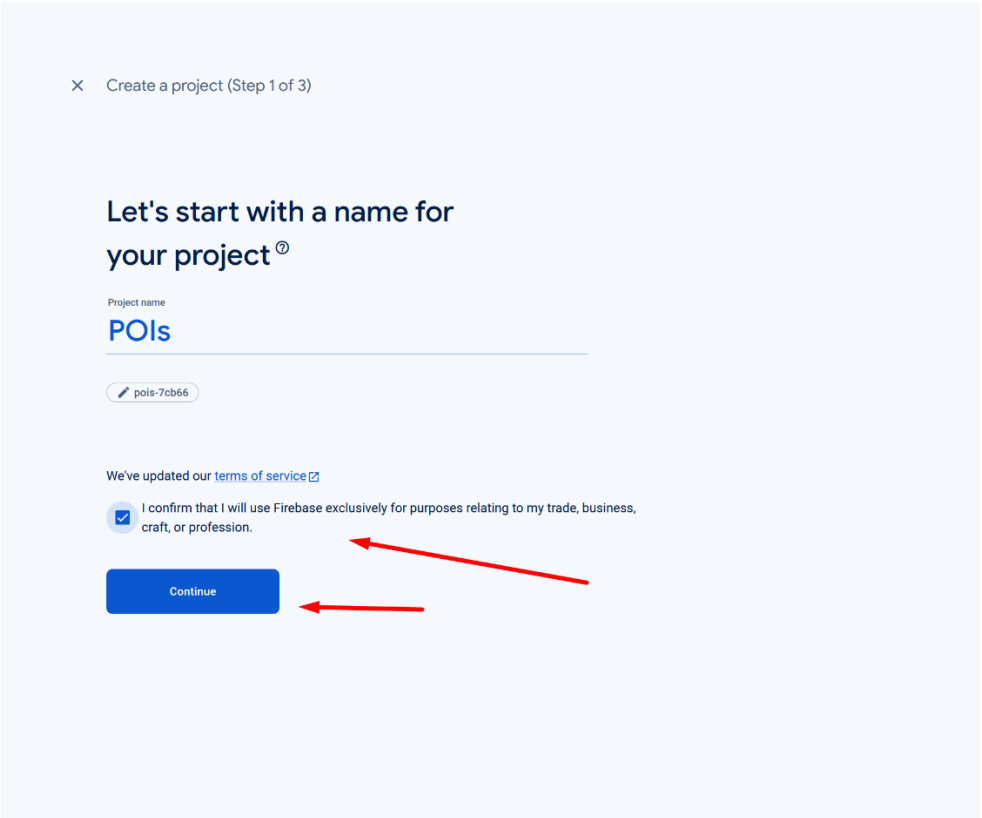
1. Logovati se na Firebase Consolu (korišćenjem google naloga) i kreirati projekat i bazu podataka prema uputstvu u nastavku. Dodati google-services.json fajl i ispratiti uputstvo do kraja kako bi aplikacija mogla da koristi *Firestore*.
2. Dodati funkciju u **StorageService** klasu za brisanje poi-a iz *Firestore* baze (delete).
3. Dovršiti implementaciju **PoiViewModel** klase dodavanjem funkcije za brisanje Poi-a (deletePoi) koja zove prethodno implementiranu funkciju StorageService klase**.** Obratiti pažnju na adekvatan poziv asinhrone funkcije.
4. Na ekranu *AddPoiScreen* dodati poziv funkcije za dodavanje novog poi-a.

Povezivanje na Firebase Firestore

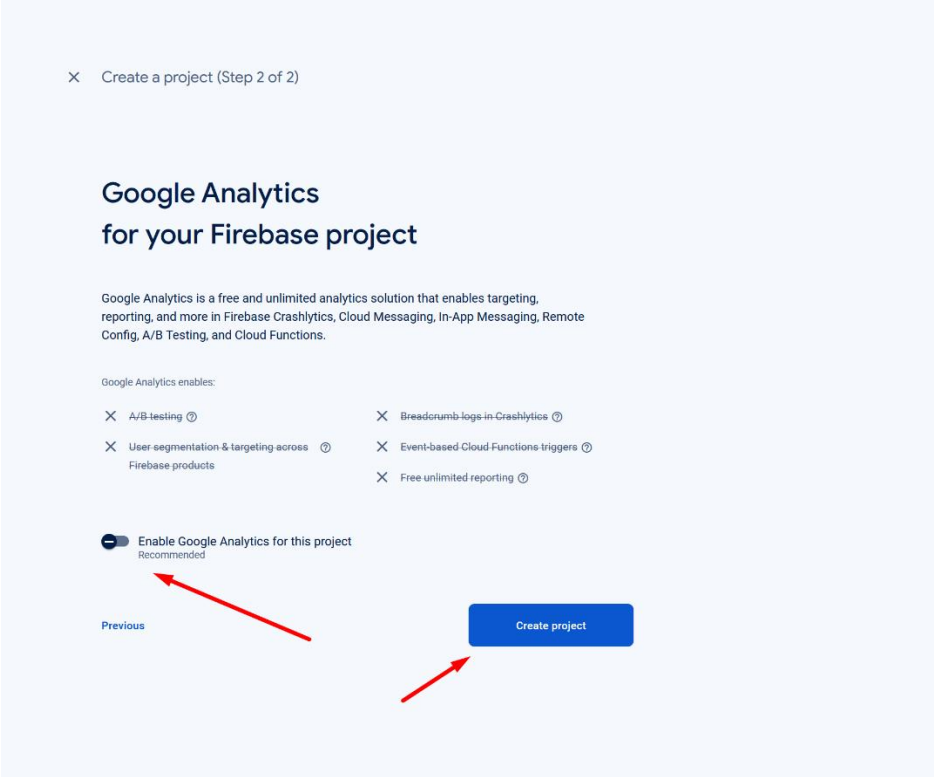
1. Logovati se na *Firebase* konzolu pomoću Google naloga.



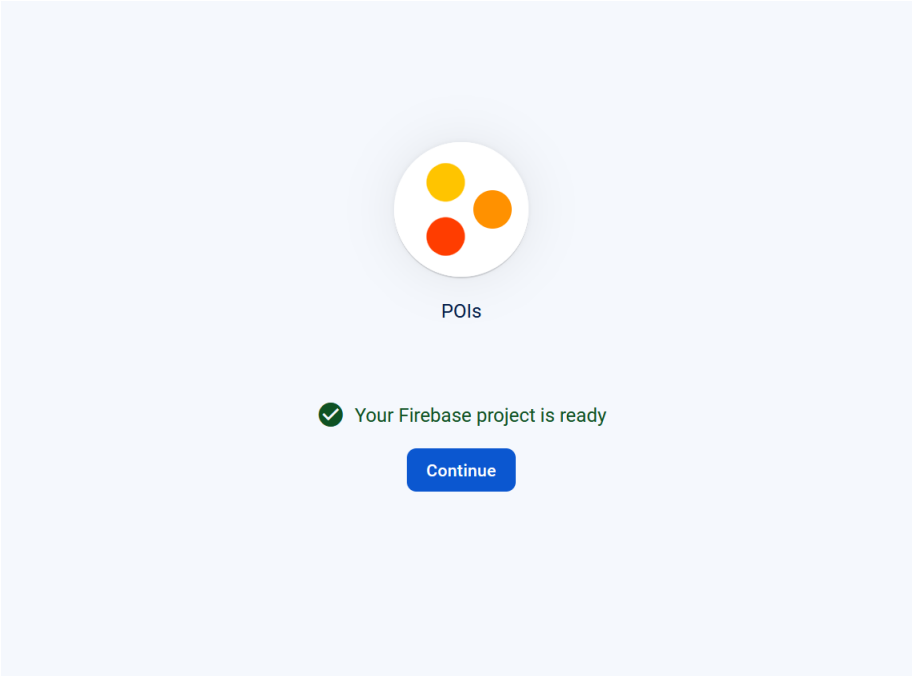
2. Kreirati projekat



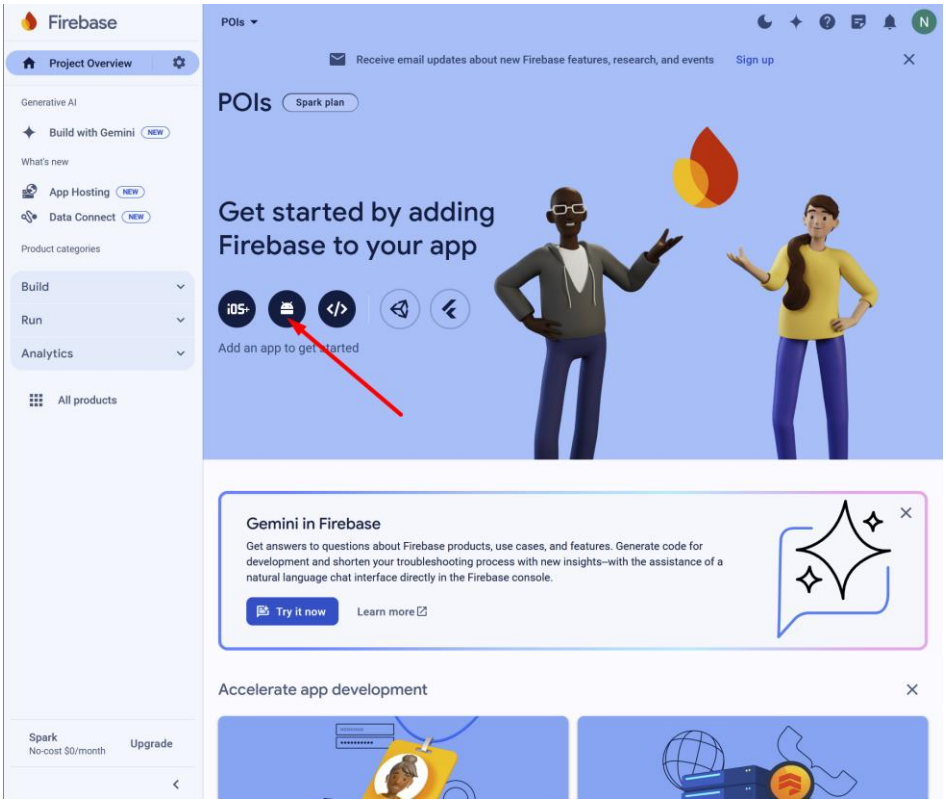
3. Onemogućiti analitiku i nastaviti



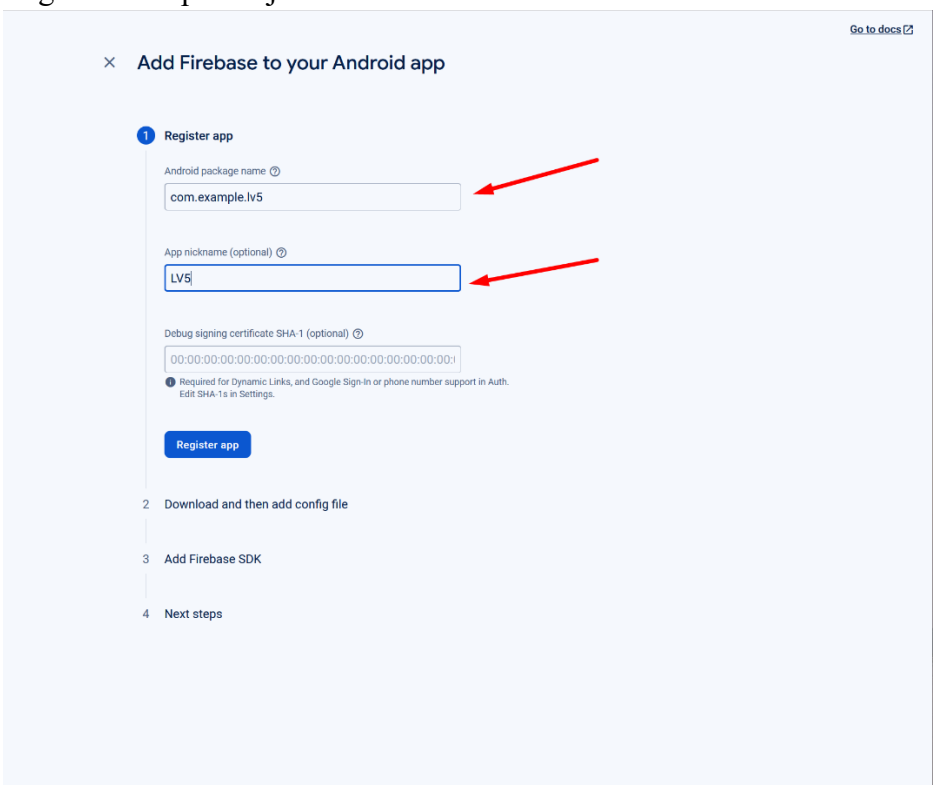
4. Sačekati dok projekat ne bude kreiran a zatim nastaviti.



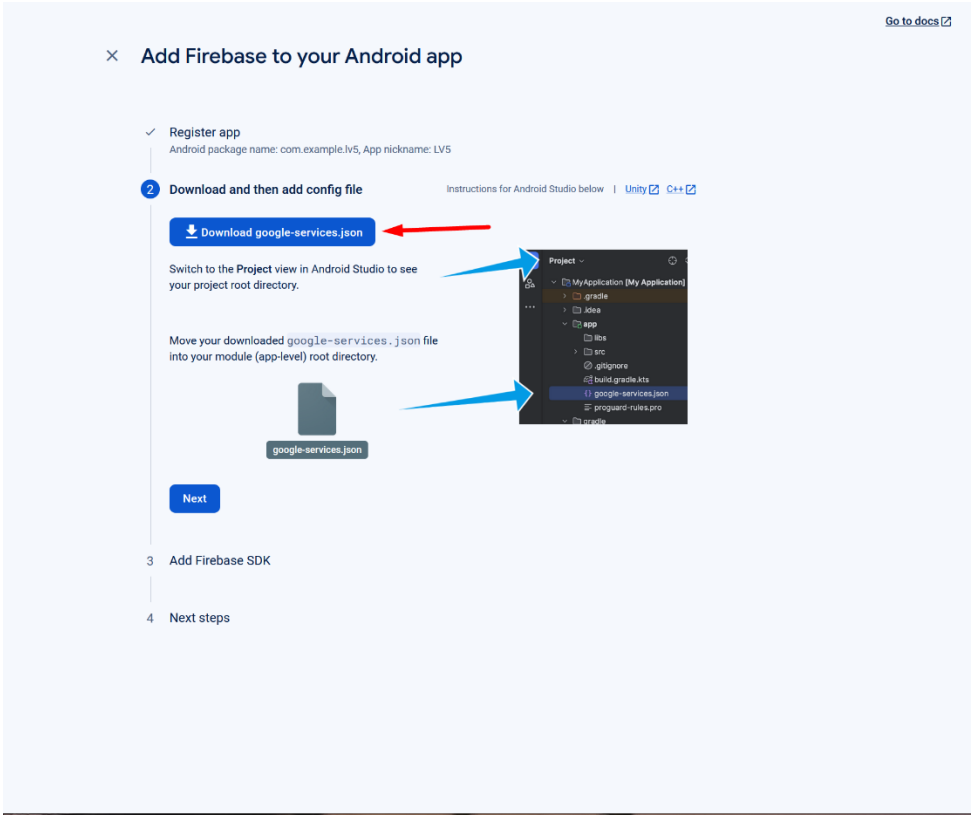
5. Kliknuti na Android ikonicu kako bi Firebase bio dodat u Android projekat

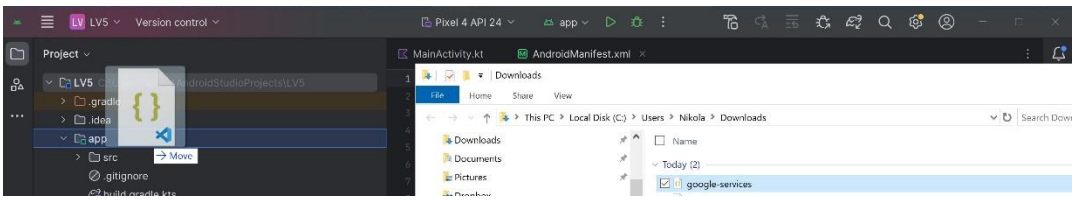


6. Registrovati aplikaciju

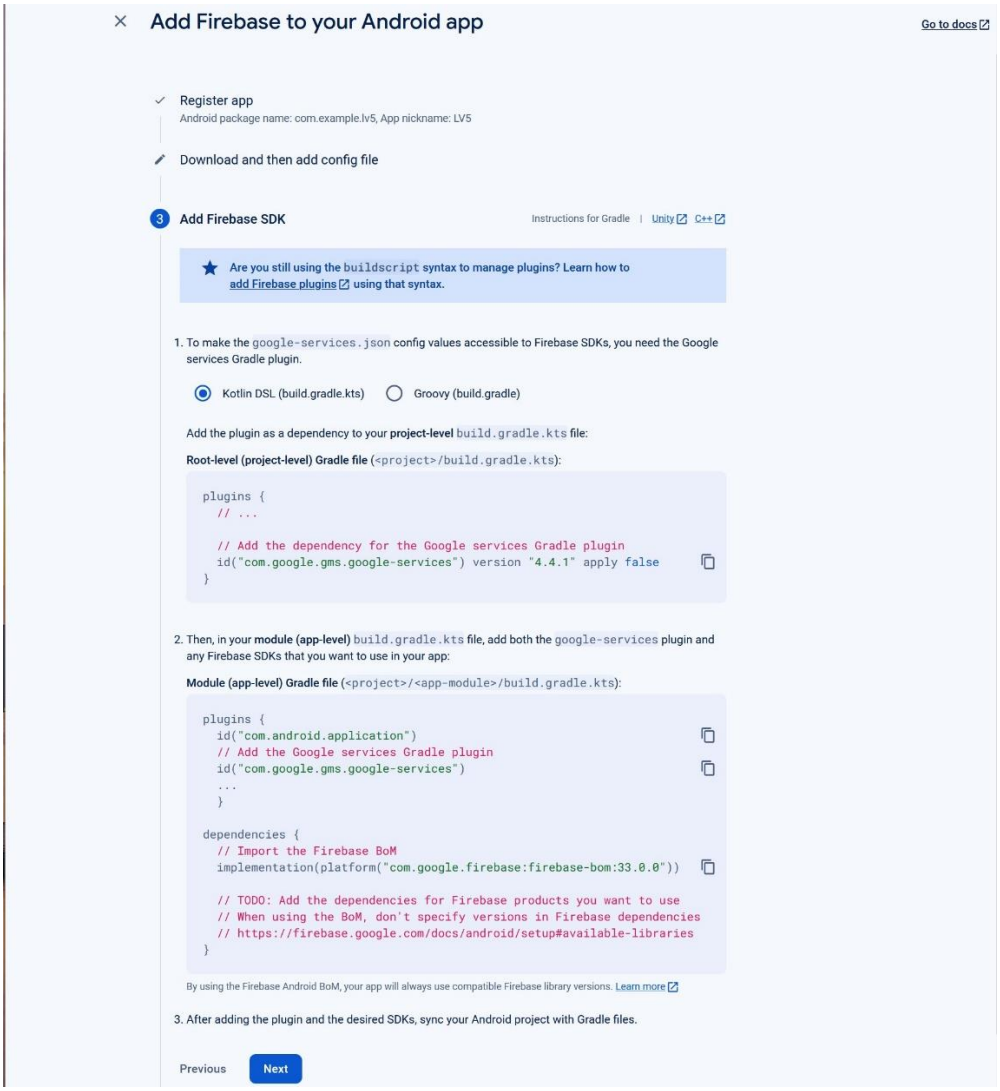


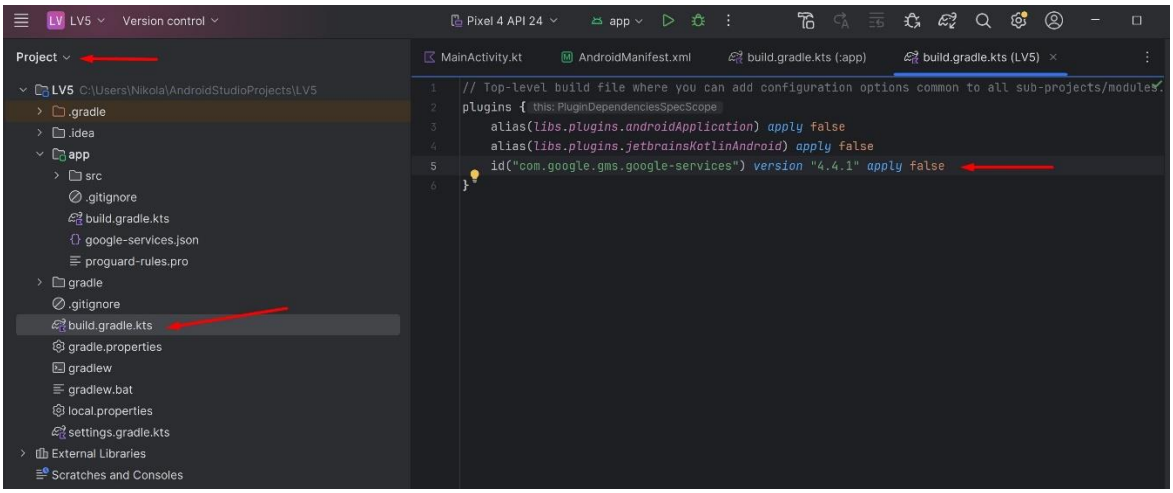
7. Preuzeti i postaviti (prepisati postojeći) google-services.json fajl.

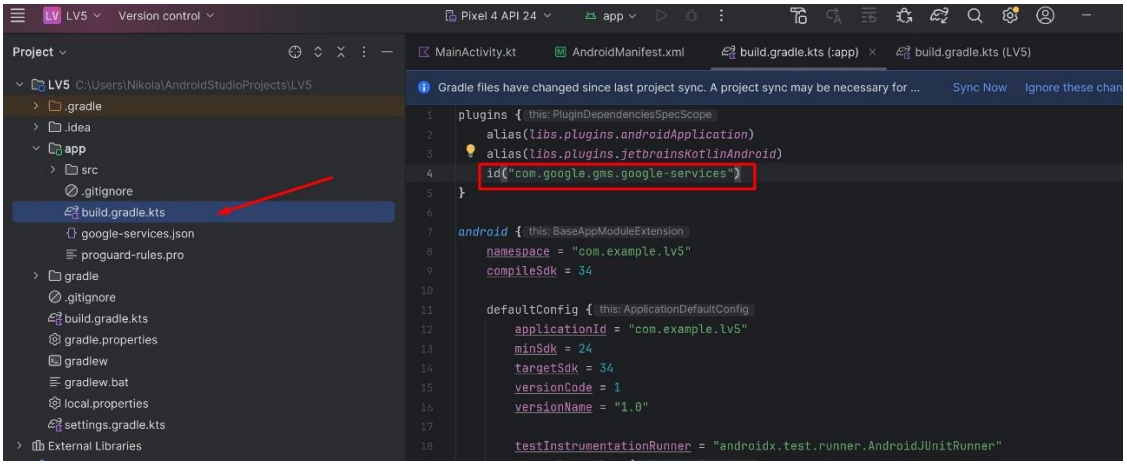




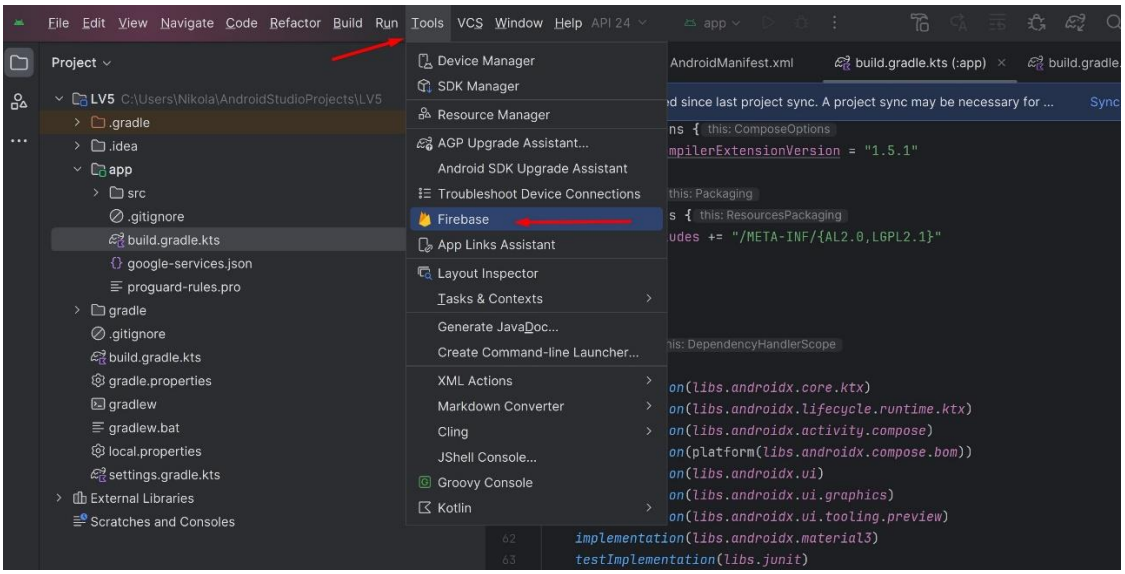
8. Dodati odgovarajuće slogove u *Gradle* fajlove kako bi SDK bio adekvatno uključen. U slučaju ove vežbe proveriti da li je sve adekvatno uključeno.


Add Firebase to your Android app
Go to docs
Register app
Android package name: com.example.lv5, App nickname: LV5
Download and then add config file
3 Add Firebase SDK
Instructions for Gradle | Unity C++
Are you still using the buildscript syntax to manage plugins? Learn how to add Firebase plugins using that syntax.
1. To make the google-services.json config values accessible to Firebase SDKs, you need the Google services Gradle plugin.
Kotlin DSL (build.gradle.kts)
Groovy (build.gradle)
Add the plugin as a dependency to your project-level build.gradle.kts file:
Root-level (project-level) Gradle file (<project>/build.gradle.kts):
plugins {
// ...
// Add the dependency for the Google services Gradle plugin
id("com.google.gms.google-services") version "4.4.1" apply false
}
2. Then, in your module (app-level) build.gradle.kts file, add both the google-services plugin and any Firebase SDKs that you want to use in your app:
Module (app-level) Gradle file (<project>/<app-module>/build.gradle.kts):
plugins {
id("com.android.application")
// Add the Google services Gradle plugin
id("com.google.gms.google-services")
...
}
dependencies {
// Import the Firebase BoM
implementation(platform("com.google.firebase:firebase-bom:33.0.0"))
// TODO: Add the dependencies for Firebase products you want to use
// When using the BoM, don't specify versions in Firebase dependencies
// https://firebase.google.com/docs/android/setup#available-libraries
}
By using the Firebase Android BoM, your app will always use compatible Firebase library versions. Learn more
3. After adding the plugin and the desired SDKs, sync your Android project with Gradle files.
Previous
Next

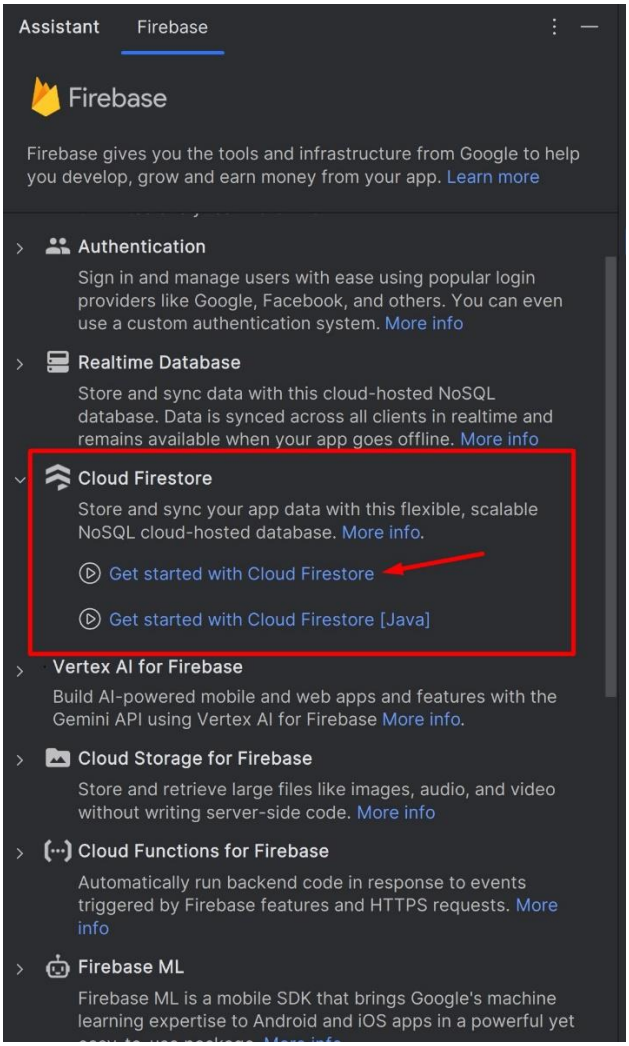

LV5
Version control
Pixel 4 API 24
app
Project
LV5 C:\Users\Nikola\AndroidStudioProjects\LV5
.gradle
.idea
app
src
.gitignore
build.gradle.kts
google-services.json
proguard-rules.pro
gradle
.gitignore
build.gradle.kts
gradle.properties
gradlew
gradlew.bat
local.properties
settings.gradle.kts
External Libraries
Scratches and Consoles
MainActivity.kt
AndroidManifest.xml
build.gradle.kts (:app)
build.gradle.kts (LV5)
// Top-level build file where you can add configuration options common to all sub-projects/modules.
plugins { this: PluginDependenciesSpecScope
alias(libs.plugins.androidApplication) apply false
alias(libs.plugins.jetbrainsKotlinAndroid) apply false
id("com.google.gms.google-services") version "4.4.1" apply false
}



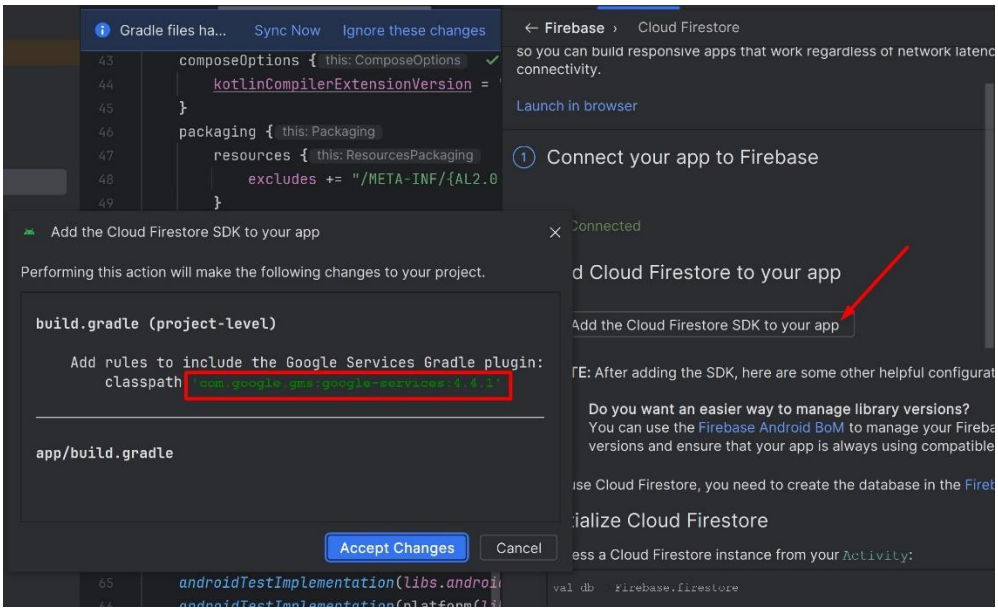
9. Otvoriti *Firebase* alat u *Android Studio-u* i proveriti da li je sve adekvatno povezano



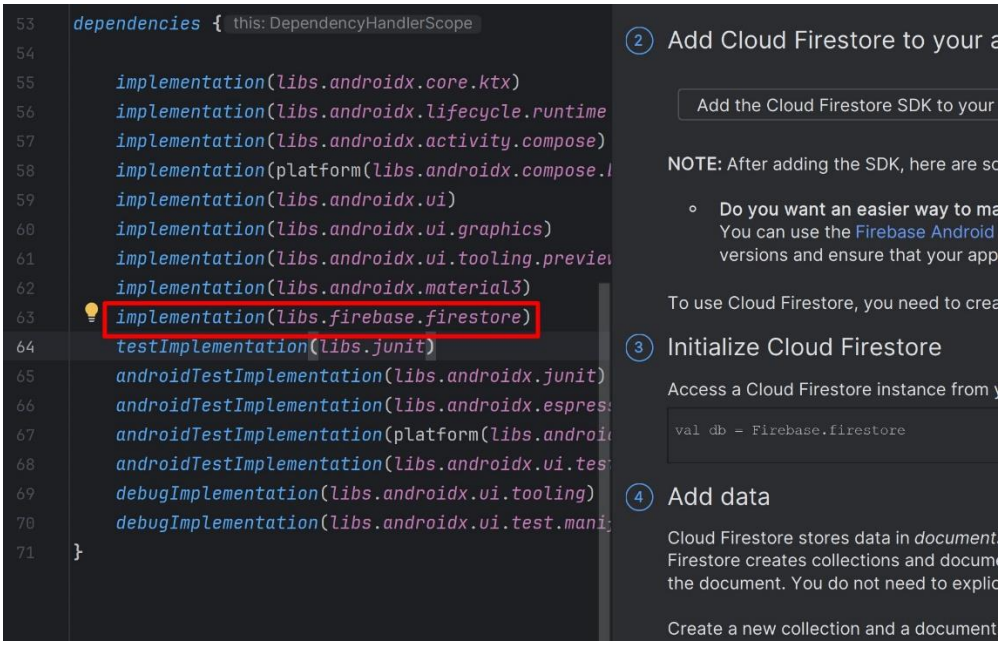
10. I pokrenuti integraciju sa *Firestore*-om.



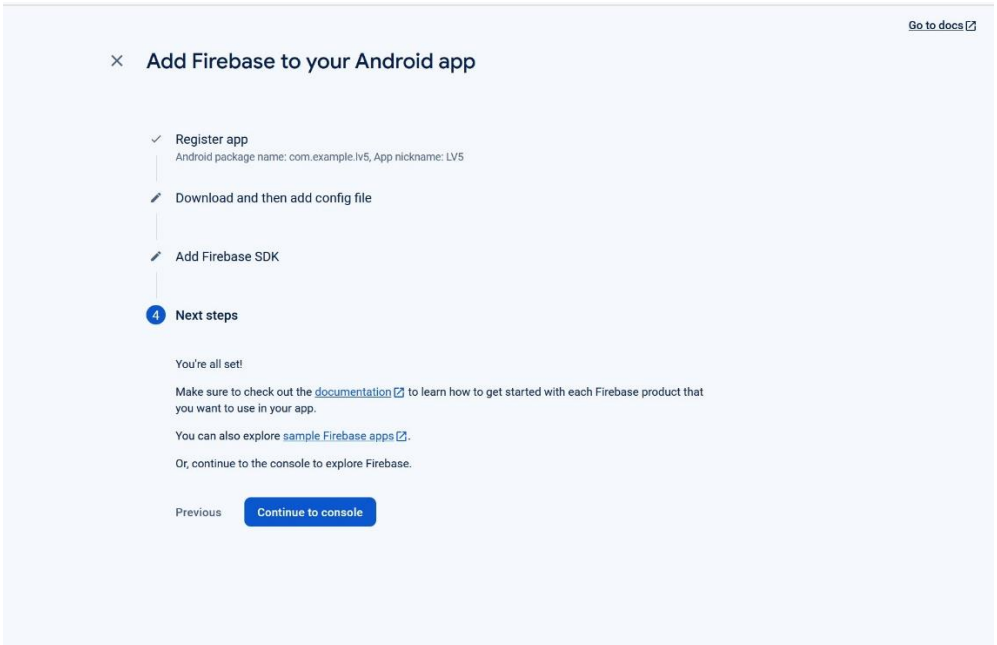
11. Pratiti upustva *Firebase* asistenta



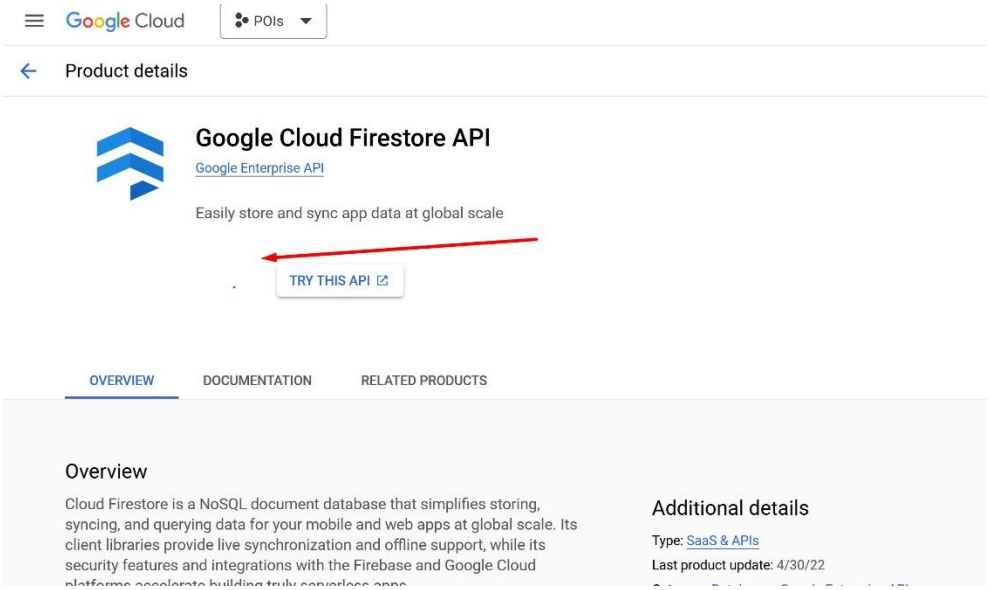
12. Dodati *Firestore* biblioteku



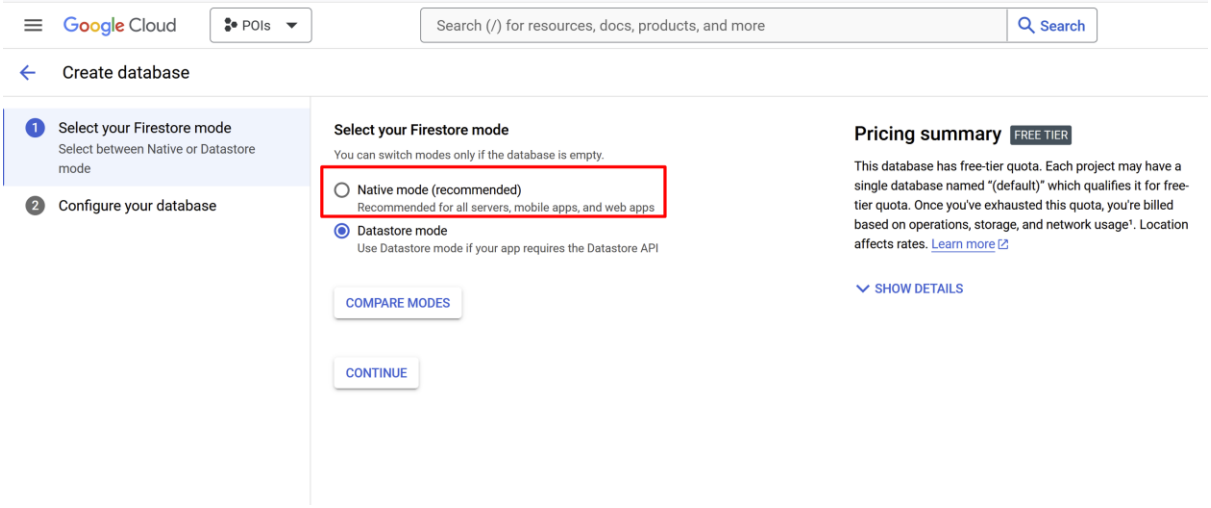
13. Nastaviti na *Firebase* konzolu.



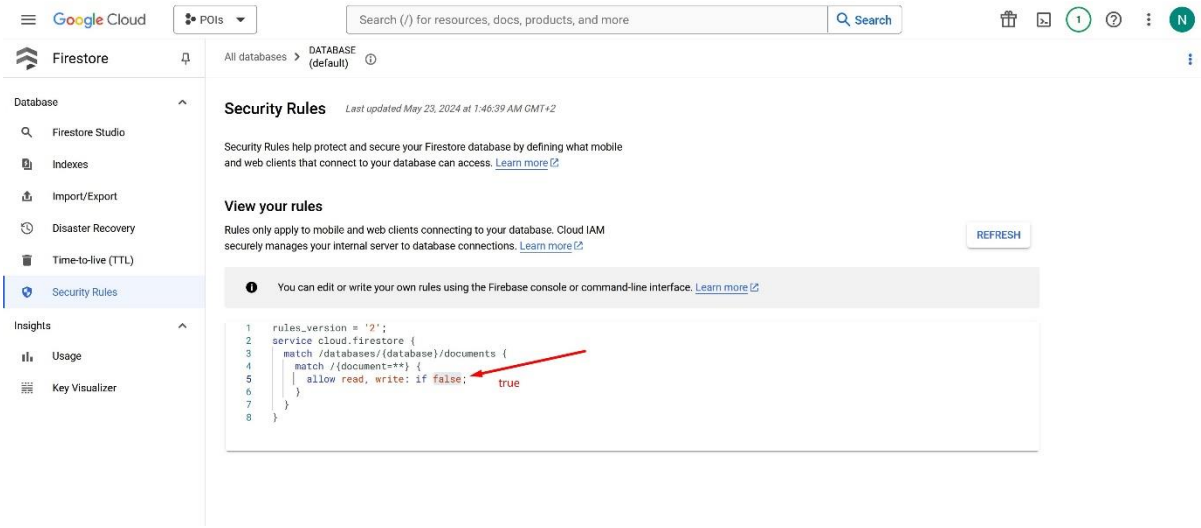
14. Pokrenuti Cloud Firebase API

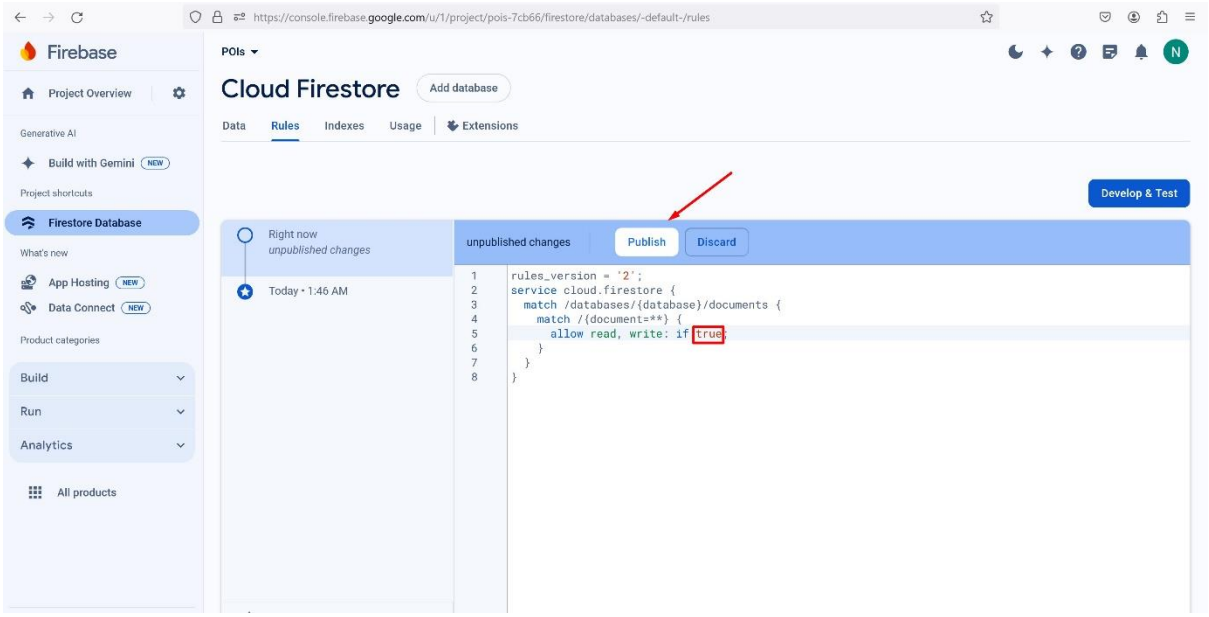


15. Izabrati Native mode



16. Potrebno je izmeniti prava pristupa iz Firebase konzole





17. Počnite da koristite Firestore u svojoj aplikaciji.