MPL Experiment 4

Name: Rohan Lalchandani

Class: D15A Roll no: 25

AIM-To connect Flutter UI with firebase.

THEORY-

Firebase helps developers to manage their mobile app easily. It is a service provided by Google. Firebase has various functionalities available to help developers manage and grow their mobile apps.

Steps to Add firebase to our Flutter app using Firebase CLI

1.Install the Firebase CLI and log in (run firebase login)

2.From any directory, run this command:

- dart pub global activate flutterfire_cli

3.Then, at the root of your Flutter project directory, run this command:

- flutterfire configure --project=questitnextjs

4.This automatically registers your per-platform apps with Firebase and adds a lib/firebase_options.dart configuration file to your Flutter project.

5.To initialise Firebase, call Firebase.initializeApp from the firebase_core package with the configuration from your new firebase_options.dart file:

import 'package:firebase_core/firebase_core.dart';

import 'firebase_options. dart'; await

Firebase.initializeApp(options:

DefaultFirebaseOptions.currentPlatform,

);

6.Add the dependencies in the pubspec.yaml file

Firebase_core : ^version Firebase_auth

: ^version

SYNTAX

```
import 'package:firebase_auth/firebase_auth.dart';

Future<void> signInUser(String email, String password) async {
  try {
    await FirebaseAuth.instance.signInWithEmailAndPassword(
      email: email,
password: password,
    );
    print("User Signed In Successfully!");
  } catch (e) {
print("Error: $e");
  }
}
```

Widget Properties

1)Firebase_Auth

- currentUser → Returns the currently signed-in user.
- signInWithEmailAndPassword(email, password) → Logs in a user.
- createUserWithEmailAndPassword(email, password) → Registers a new user.
- signOut() → Logs out the current user.

2)FirebaseFirestore

- collection("name") → Accesses a Firestore collection.
- doc("id") → Refers to a specific document.
- set(Map<String, dynamic> data) → Adds or updates data.
- get() → Fetches document data.
- delete() → Deletes a document.

CODE

```
import 'package:flutter/material.dart'; import
'package:font_awesome_flutter/font_awesome_flutter.dart';
import 'package:firebase_auth/firebase_auth.dart'; import
'package:my_app/pages/mainpage.dart';

class CreateAccount extends StatefulWidget {
const CreateAccount({super.key});

  @override
  _CreateAccountState createState() => _CreateAccountState();
}

class _CreateAccountState extends State<CreateAccount>
{     final _formKey = GlobalKey<FormState>();      final
_usernameController = TextEditingController();
  final _emailController = TextEditingController();   final _dobController =
TextEditingController();   final _passwordController = TextEditingController();
final FirebaseAuth _auth = FirebaseAuth.instance; // Firebase Authentication
Instance

  bool _isLoading = false;
```

```
  // Function to handle Firebase registration
Future<void> _registerUser() async {     if
(!_formKey.currentState!.validate()) return;


   setState(() {
     _isLoading = true;
   });
    try
{
     UserCredential userCredential = await
_auth.createUserWithEmailAndPassword(         email: _emailController.text.trim(),
password: _passwordController.text.trim(),
     );


     User? user = userCredential.user;       if (user != null) {
print("User Registered: ${user.email}");
ScaffoldMessenger.of(context).showSnackBar(           const
SnackBar(content: Text("Account Created Successfully!")),
       );


       // Navigate to Home Page after successful signup
       Navigator.pushReplacement(
         context,
         MaterialPageRoute(builder: (context) => TwitterHomePage()), // Update with the
correct main page
       );
     }
```

```
    } on FirebaseAuthException catch (e) {
print("Firebase Auth Error: ${e.message}");
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("Error: ${e.message}")),
      );
    }

    setState(() {
      _isLoading = false;
    });
  }

  @override
  Widget build(BuildContext context) {        return Scaffold(
appBar: AppBar(          leading: IconButton(            icon: const
Icon(Icons.arrow_back, color: Colors.black),          onPressed:
() {
            Navigator.pop(context);
          },
        ),
        centerTitle: true,
        title: const FaIcon(FontAwesomeIcons.twitter, color: Colors.blue, size:
30),          backgroundColor: Colors.transparent,          elevation: 0,
      ),
      body: Padding(          padding: const
EdgeInsets.all(20),          child: Form(            key:
_formKey,            child: Column(
crossAxisAlignment: CrossAxisAlignment.start,
```

```
        children: [
// Username
          const Text("Username", style: TextStyle(fontSize: 16, fontWeight:
FontWeight.bold)),
          TextFormField(              controller: _usernameController,
decoration: InputDecoration(                hintText: "Enter your username",
border: OutlineInputBorder(borderRadius: BorderRadius.circular(10)),
            ),
            validator: (value) => value!.isEmpty ? "Username cannot be empty" : null,
          ),
          const SizedBox(height: 15),

          // Email            const Text("Email", style: TextStyle(fontSize: 16, fontWeight:
FontWeight.bold)),            TextFormField(              controller: _emailController,
decoration: InputDecoration(
              hintText: "Enter your email",                border:
OutlineInputBorder(borderRadius: BorderRadius.circular(10)),
            ),
            validator: (value) => value!.contains("@") ? null : "Enter a valid email",
          ),
          const SizedBox(height: 15),

          // Date of Birth
          const Text("Date of Birth", style: TextStyle(fontSize: 16, fontWeight:
FontWeight.bold)),
          TextFormField(              controller: _dobController,
decoration: InputDecoration(                hintText: "DD/MM/YYYY",
border: OutlineInputBorder(borderRadius: BorderRadius.circular(10)),
            ),
```

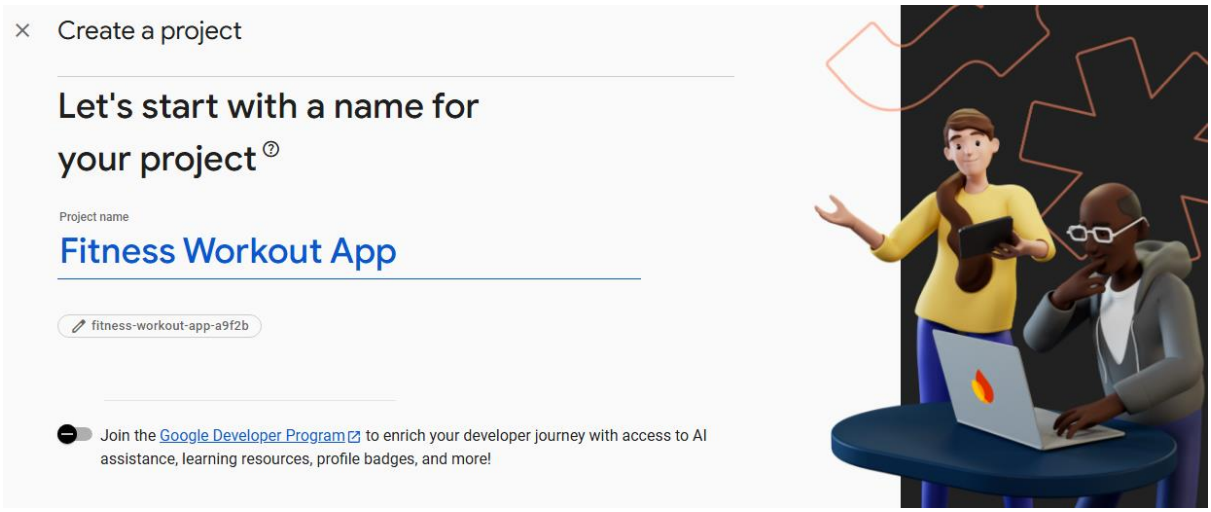
```
              keyboardType: TextInputType.datetime,
            ),
            const SizedBox(height: 15),

            // Password
            const Text("Password", style: TextStyle(fontSize: 16, fontWeight:
FontWeight.bold)),
            TextFormField(                controller: _passwordController,
obscureText: true,                decoration: InputDecoration(                hintText:
"Enter your password",                border: OutlineInputBorder(borderRadius:
BorderRadius.circular(10)),
              ),
              validator: (value) => value!.length < 6 ? "Password must be at least 6 characters"
: null,
            ),
          ],
        ),
      ),
    ),

    // Floating Action Button for "Next"       floatingActionButton: Padding(
padding: const EdgeInsets.only(bottom: 30), // Adjust to move the button up
child: FloatingActionButton(          onPressed: _isLoading ? null :
_registerUser,          backgroundColor: _isLoading ? Colors.grey : Colors.blue,
child: _isLoading
            ? const CircularProgressIndicator(color: Colors.white)
            : const Icon(Icons.arrow_forward, color: Colors.white),
      ),
    ),
```

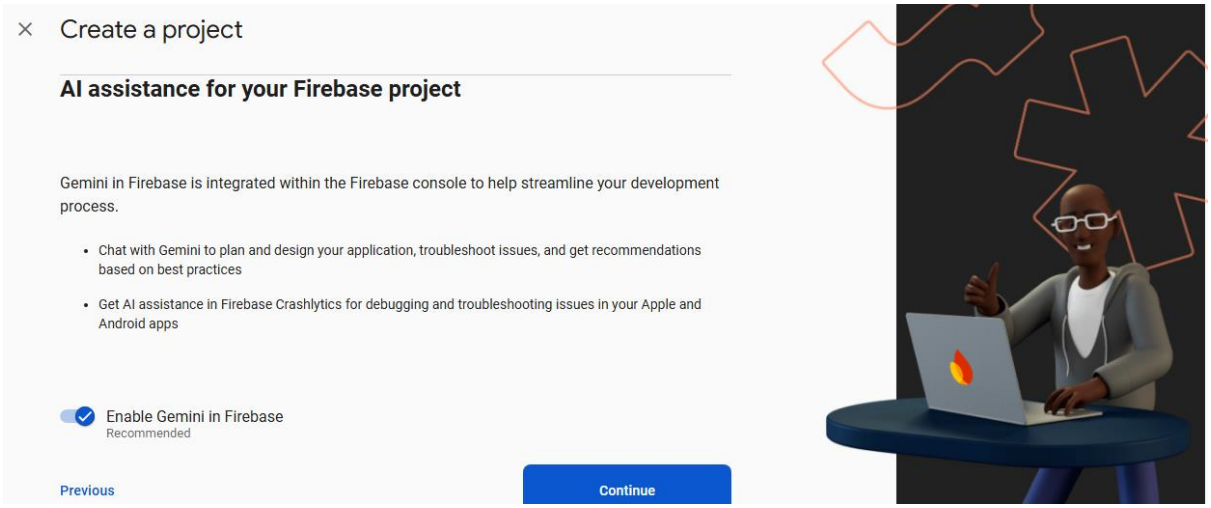
```
      floatingActionButtonLocation: FloatingActionButtonLocation.endFloat,
    );
  }
}
```

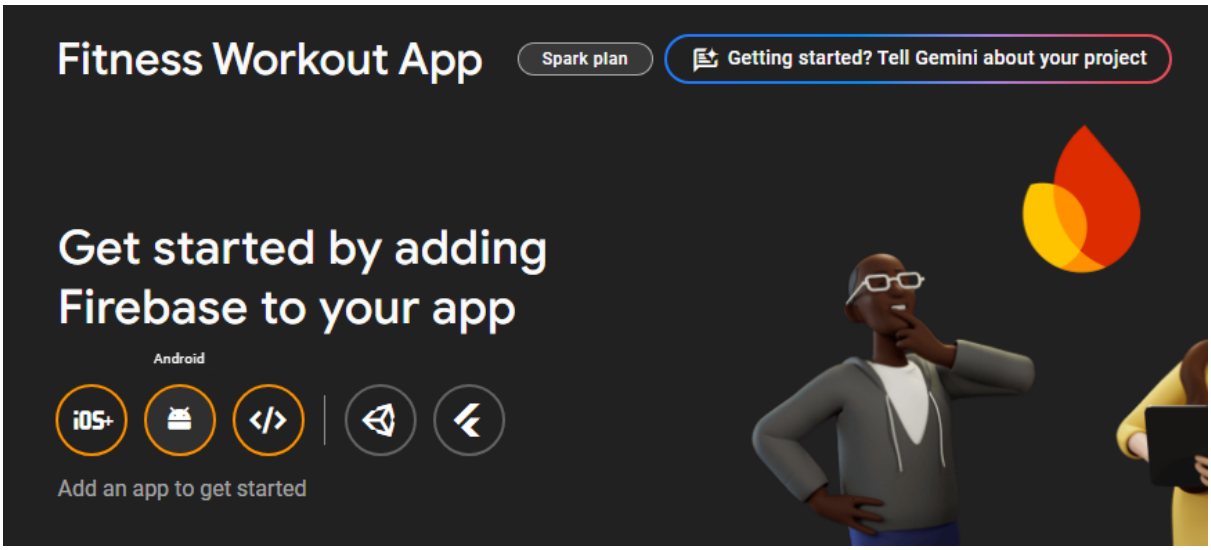
OUTPUT:

Step 1: Create a Firebase Project

1. Go to the Firebase Console
2. Click on "Add project" and follow these steps:
    - Enter a project name (e.g., "Fitness Workout App")
    - Choose whether to enable Google Analytics (recommended)
    - Accept the terms and click "Create Project"
    - Wait for project setup to complete





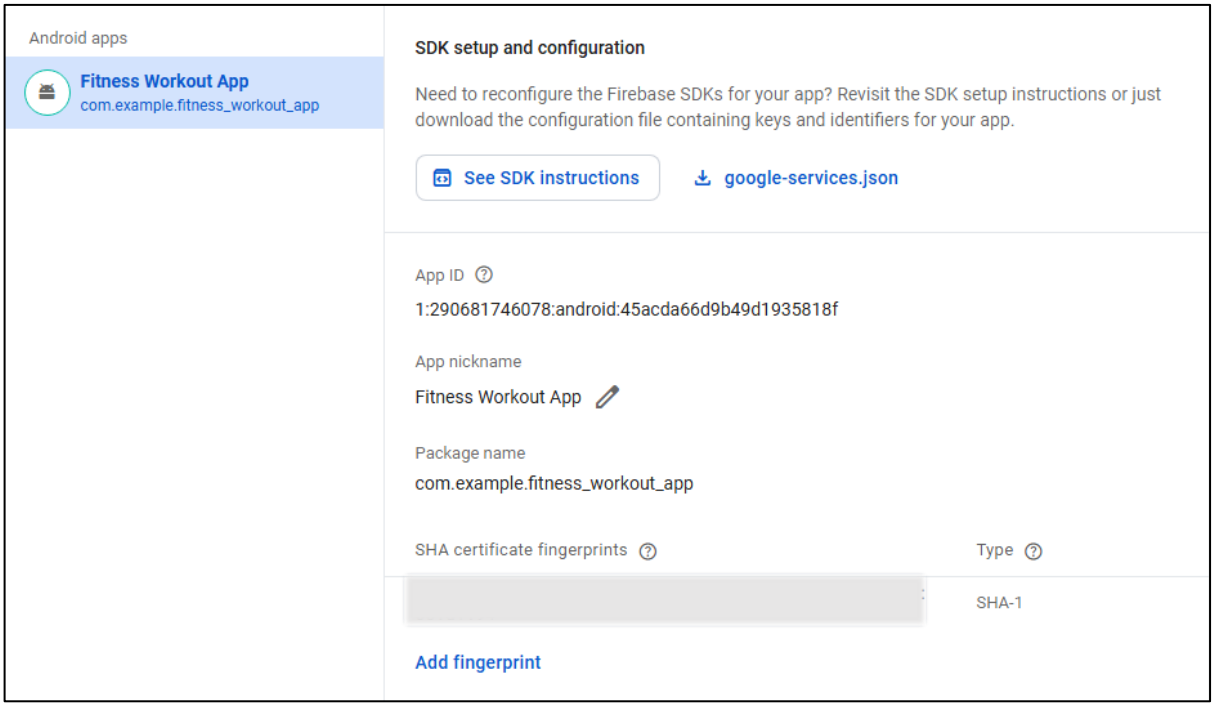


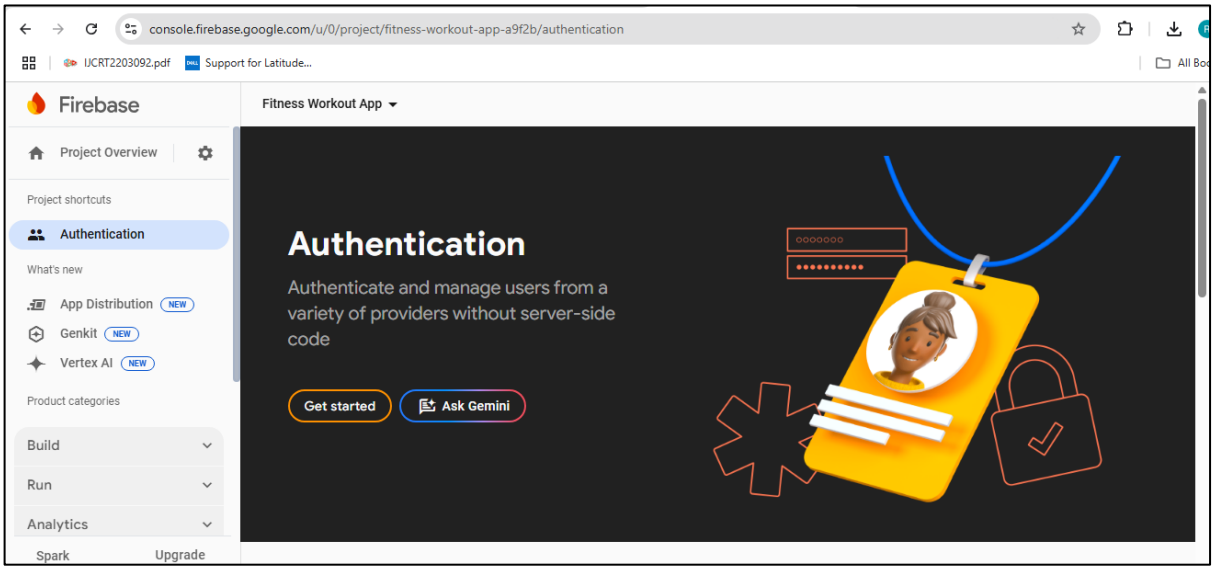

```
C:\Android>keytool -list -v -keystore %USERPROFILE%\.android\debug.keystore -alias androiddebugkey -storepass android -k
eypass android
Alias name: androiddebugkey
Creation date: 15-Feb-2025
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: C=US, O=Android, CN=Android Debug
Issuer: C=US, O=Android, CN=Android Debug
Serial number: 1
Valid from: Sat Feb 15 21:56:58 IST 2025 until: Mon Feb 08 21:56:58 IST 2055
Certificate fingerprints:
         SHA1: B0
         SHA256:
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 1
```

Step 2: Register Your Flutter App with Firebase

For Android:

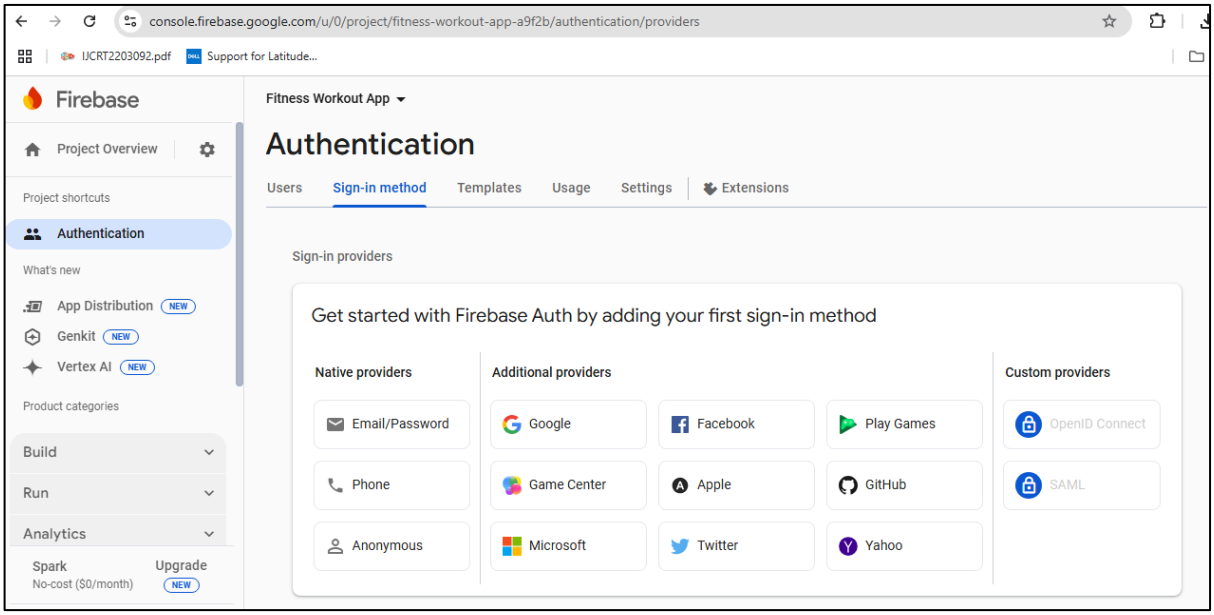
1. In the Firebase console, click on the Android icon to add an Android app
2. Enter your Android package name:
    - Find this in your android/app/build.gradle file under applicationId
    - Typically follows format: "com.yourname.fitnessapp"
3. Click "Register app"
4. Download the google-services.json file
5. Place this file in the android/app/ directory of your Flutter project.

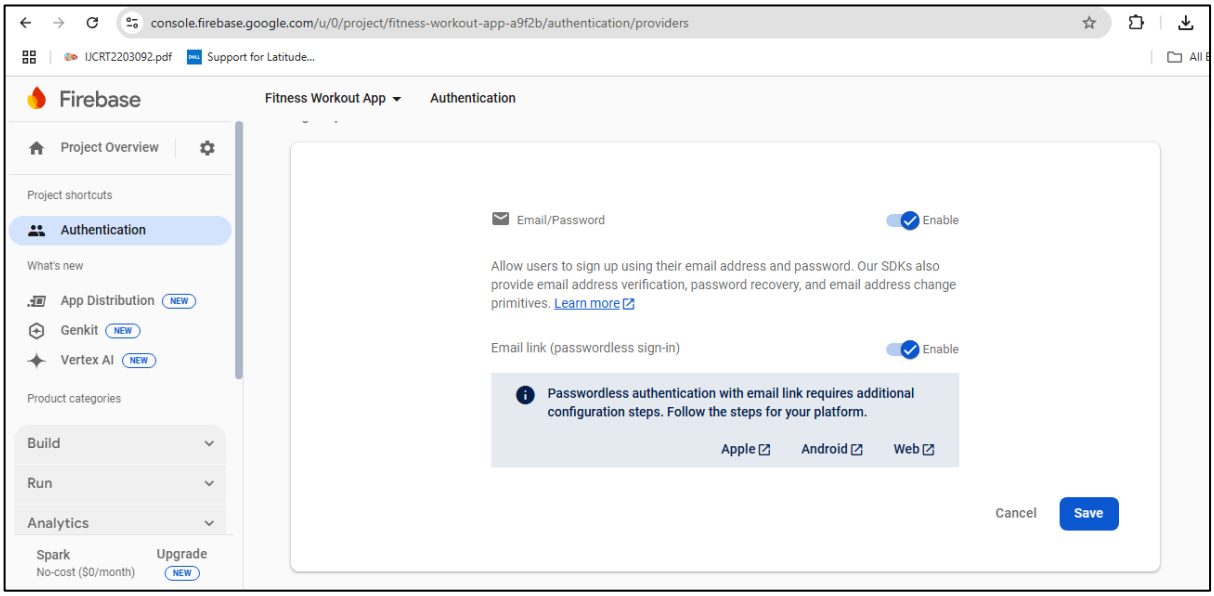




Step 5: Enable Authentication Methods

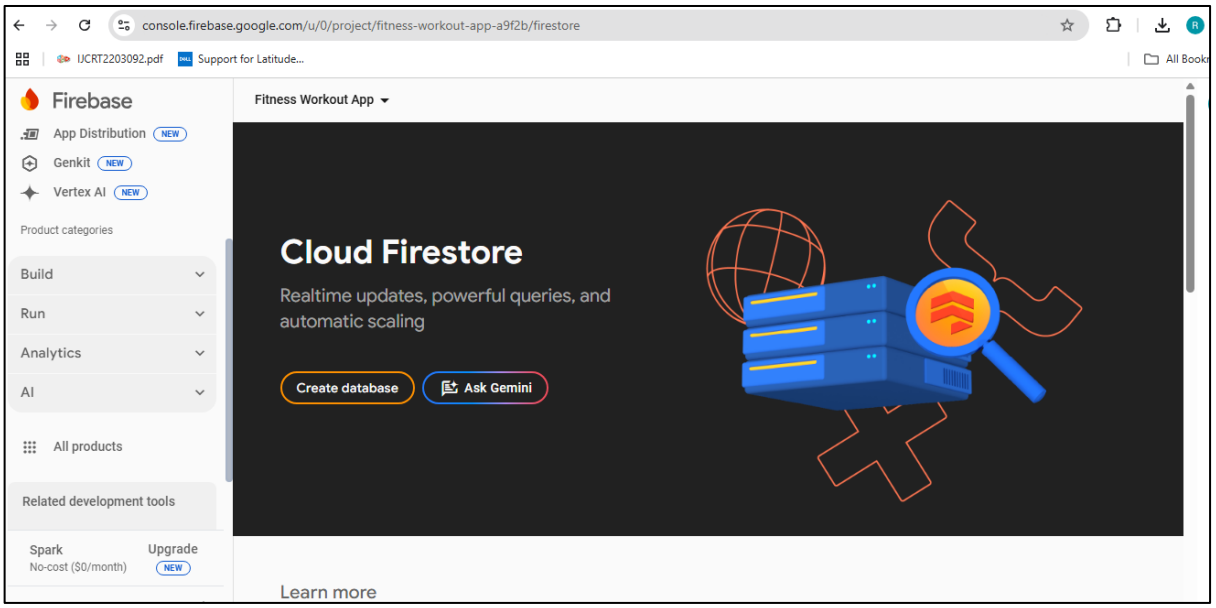
1. In the Firebase console, go to "Authentication" section
2. Click on the "Sign-in method" tab
3. Enable the authentication methods you need:
    - Email/Password (recommended to start with)
    - Google Sign-in
    - Any others you want to support

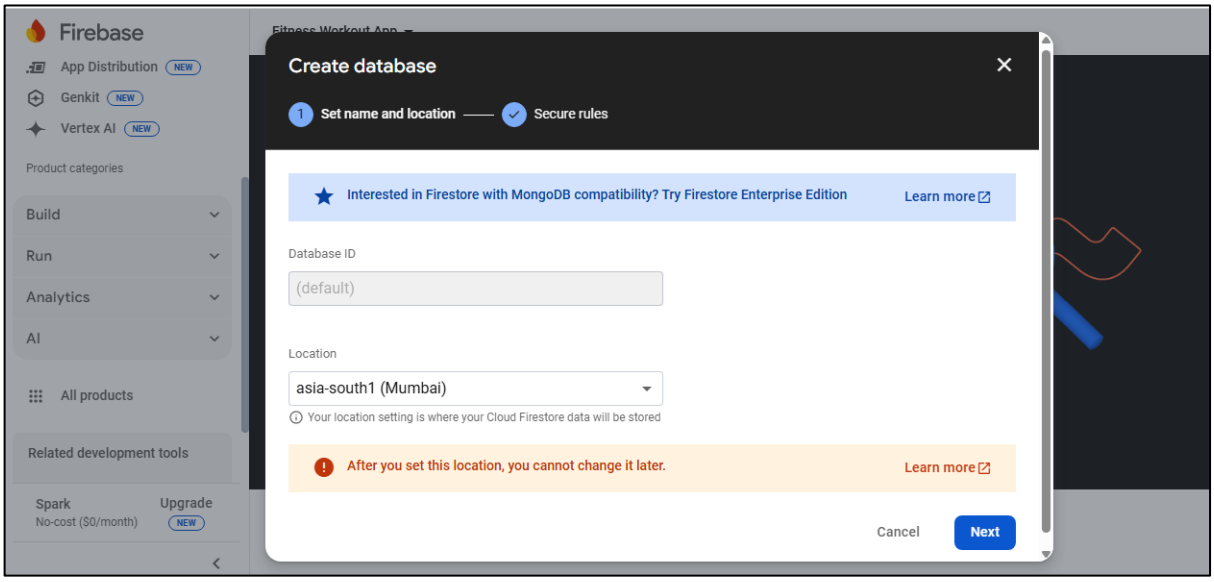


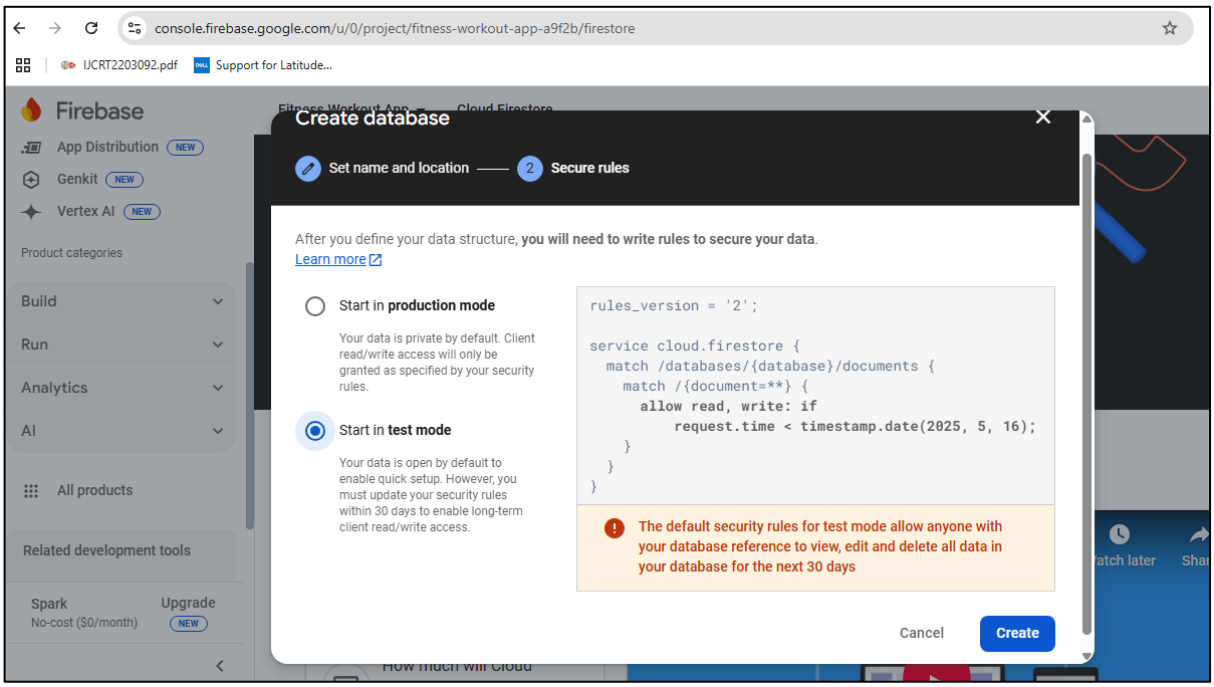


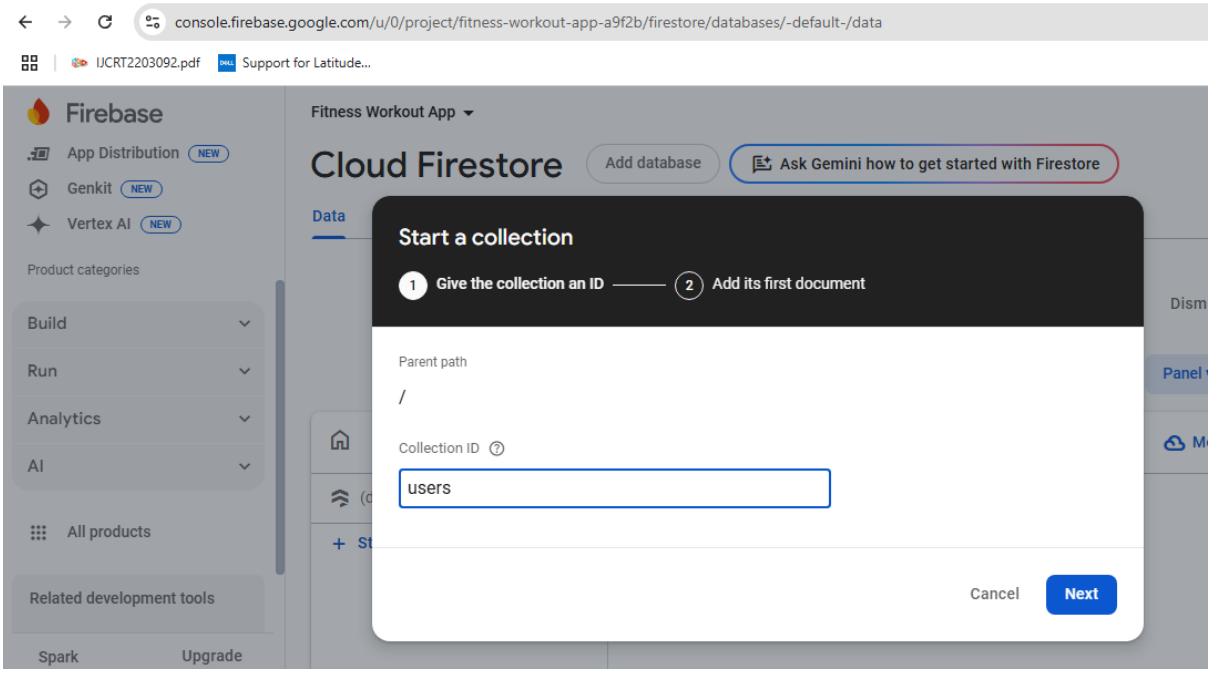
Step 6: Set Up Cloud Firestore Database

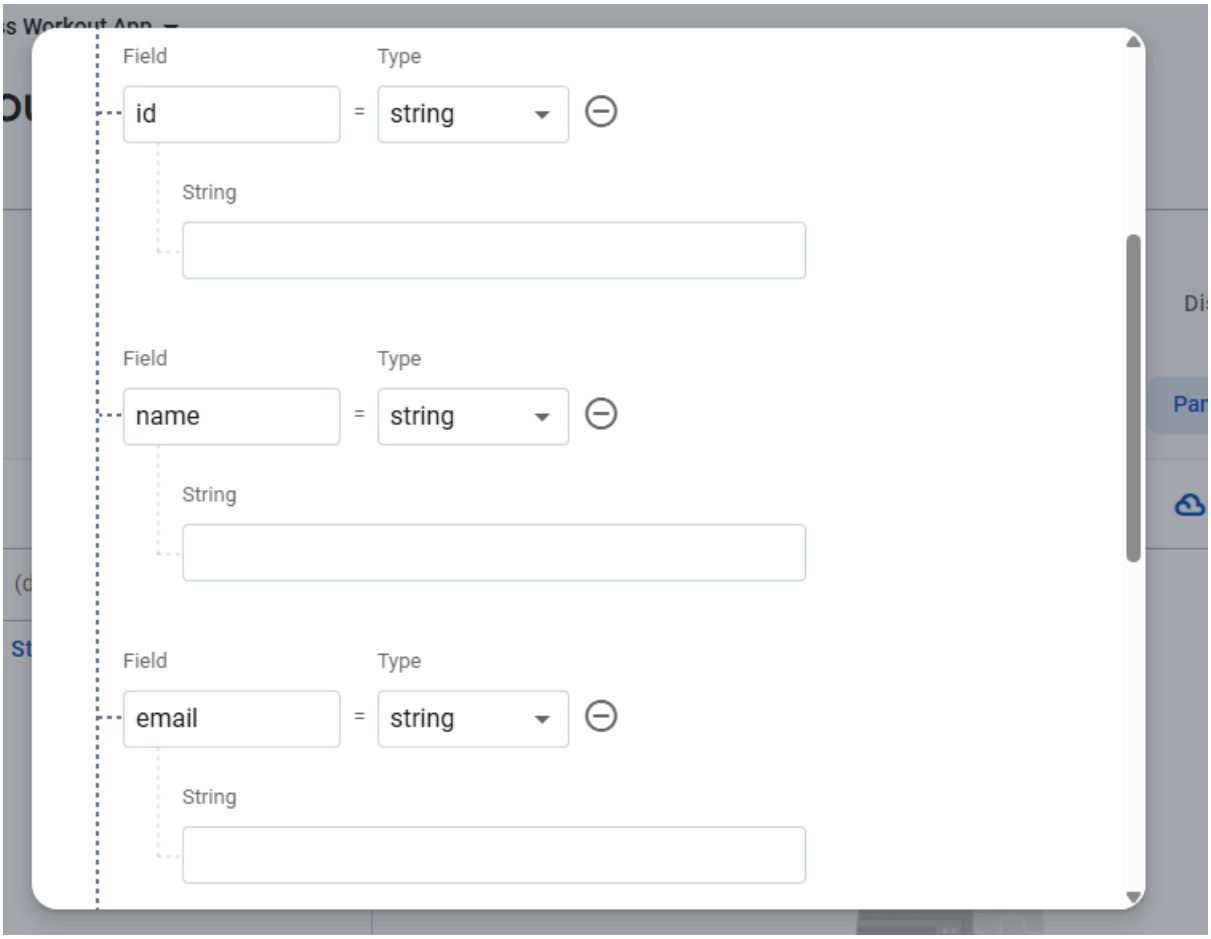
1. In the Firebase console, go to "Firestore Database"
2. Click "Create database"
3. Choose start mode:
    - "Start in test mode" for development
    - "Start in production mode" for stricter security rules
4. Choose database location closest to your users
5. Click "Enable"

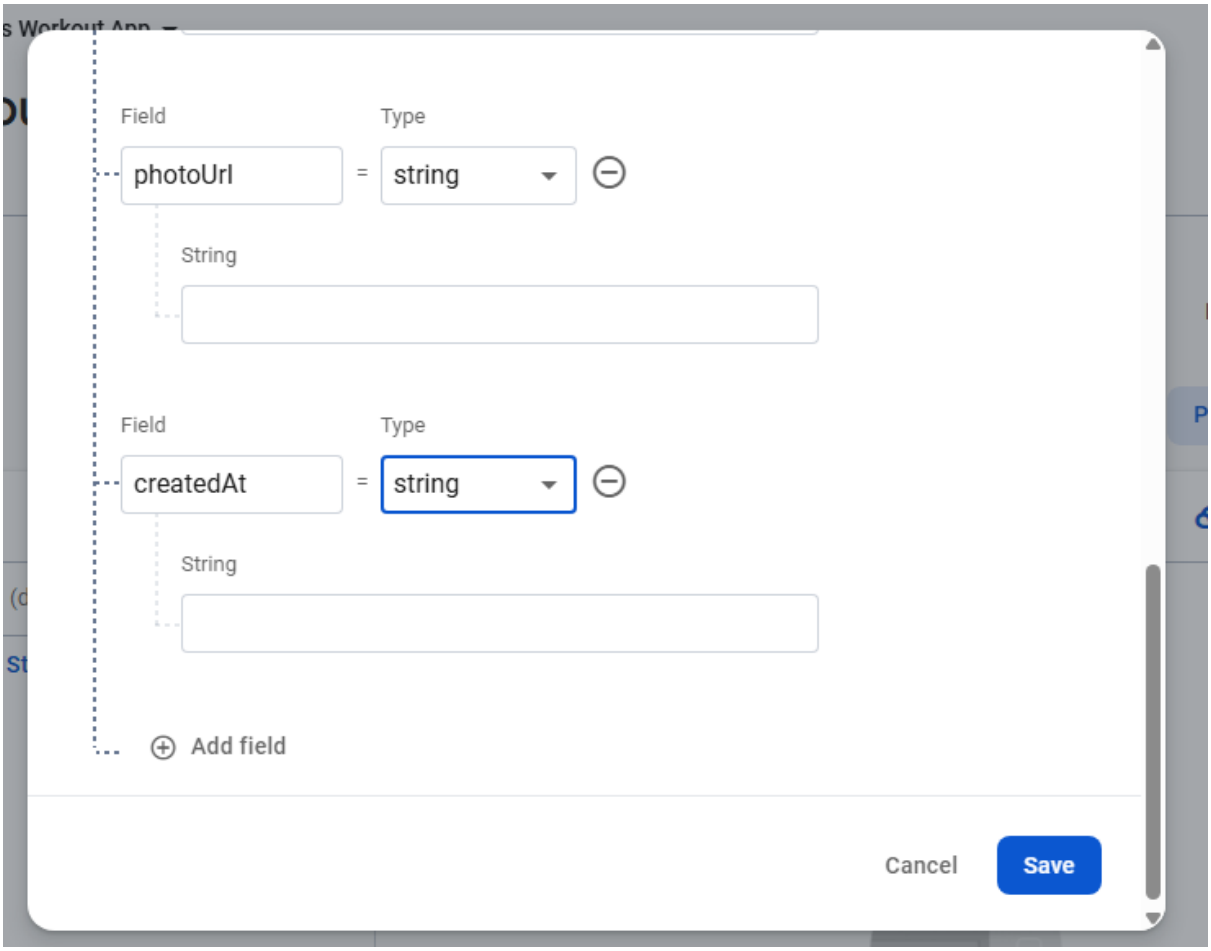


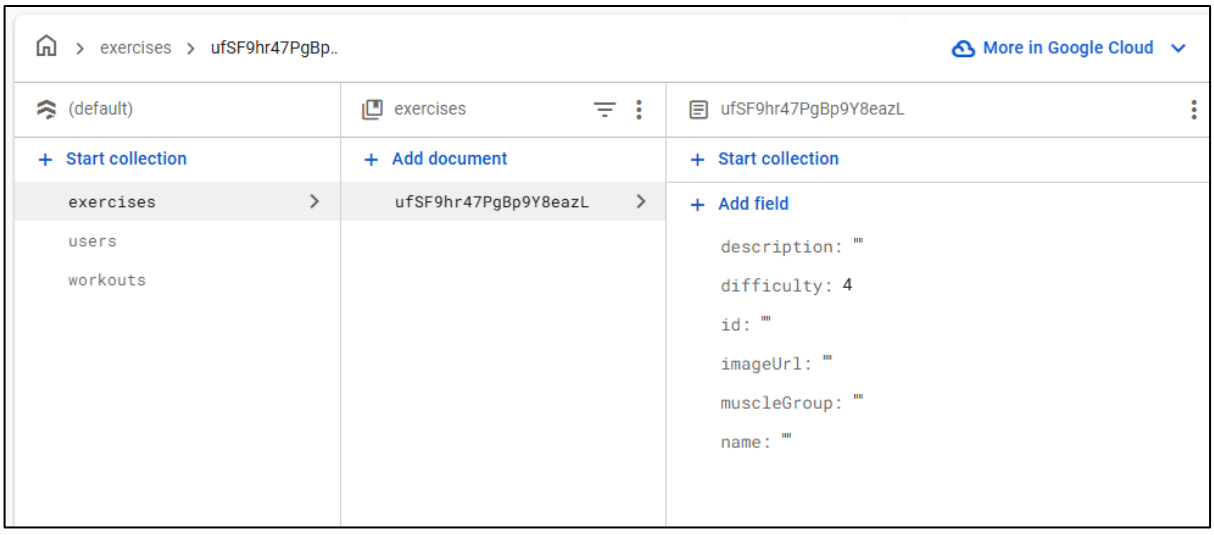

Firebase
App Distribution NEW
Genkit NEW
Vertex AI NEW
Product categories
Build
Run
Analytics
AI
All products
Related development tools
Spark
No-cost ($0/month)
Upgrade
NEW
Create database
1 Set name and location
Secure rules
Interested in Firestore with MongoDB compatibility? Try Firestore Enterprise Edition
Learn more
Database ID
(default)
Location
asia-south1 (Mumbai)
Your location setting is where your Cloud Firestore data will be stored
After you set this location, you cannot change it later.
Learn more
Cancel
Next


console.firebase.google.com/u/0/project/fitness-workout-app-a9f2b/firestore
IJCRT2203092.pdf
Support for Latitude...
Firebase
App Distribution NEW
Genkit NEW
Vertex AI NEW
Product categories
Build
Run
Analytics
AI
All products
Related development tools
Spark
No-cost ($0/month)
Upgrade
NEW
Create database
Set name and location
2 Secure rules
After you define your data structure, you will need to write rules to secure your data.
Learn more
Start in production mode
Your data is private by default. Client read/write access will only be granted as specified by your security rules.
Start in test mode
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.
rules_version = '2';
service cloud.firestore {
match /databases/{database}/documents {
match /{document=**} {
allow read, write: if
request.time < timestamp.date(2025, 5, 16);
}
}
}
The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days
Cancel
Create


console.firebase.google.com/u/0/project/fitness-workout-app-a9f2b/firestore/databases/-default-/data
IJCRT2203092.pdf
Support for Latitude...
Firebase
App Distribution NEW
Genkit NEW
Vertex AI NEW
Product categories
Build
Run
Analytics
AI
All products
Related development tools
Spark
Upgrade
Fitness Workout App
Cloud Firestore
Add database
Ask Gemini how to get started with Firestore
Data
Start a collection
1 Give the collection an ID
2 Add its first document
Parent path
/
Collection ID
users
Cancel
Next


Field
Type
id
=
string
String
Field
Type
name
=
string
String
Field
Type
email
=
string
String


Field
Type
photoUrl
=
string
String
Field
Type
createdAt
=
string
String
Add field
Cancel
Save


exercises
ufSF9hr47PgBp..
More in Google Cloud
(default)
exercises
ufSF9hr47PgBp9Y8eazL
Start collection
Add document
Start collection
exercises
ufSF9hr47PgBp9Y8eazL
Add field
users
workouts
description: ""
difficulty: 4
id: ""
imageUrl: ""
muscleGroup: ""
name: ""