

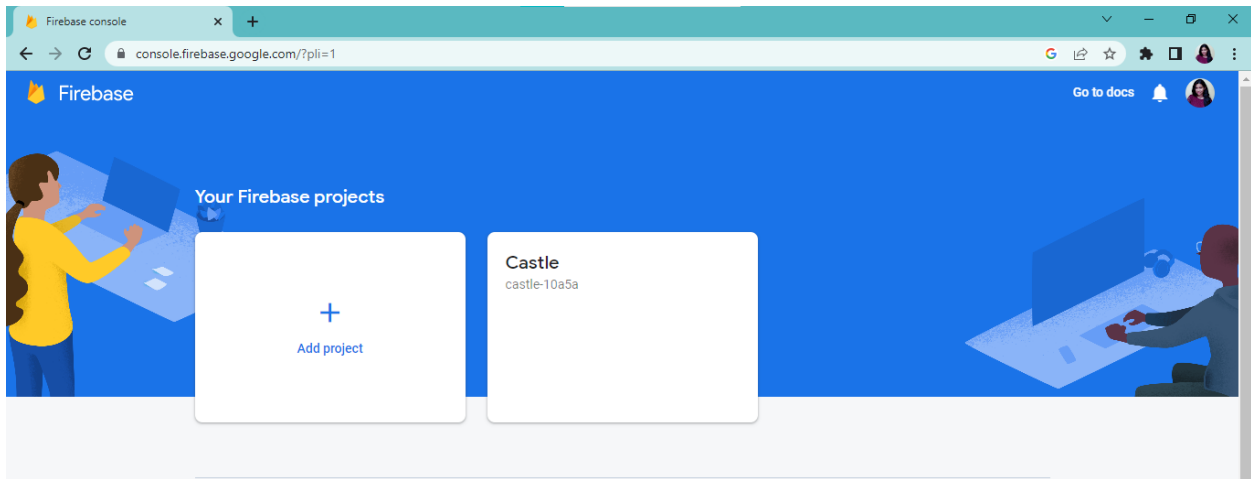
EXPERIMENT - 06

AIM: To Connect Flutter UI with firebase database

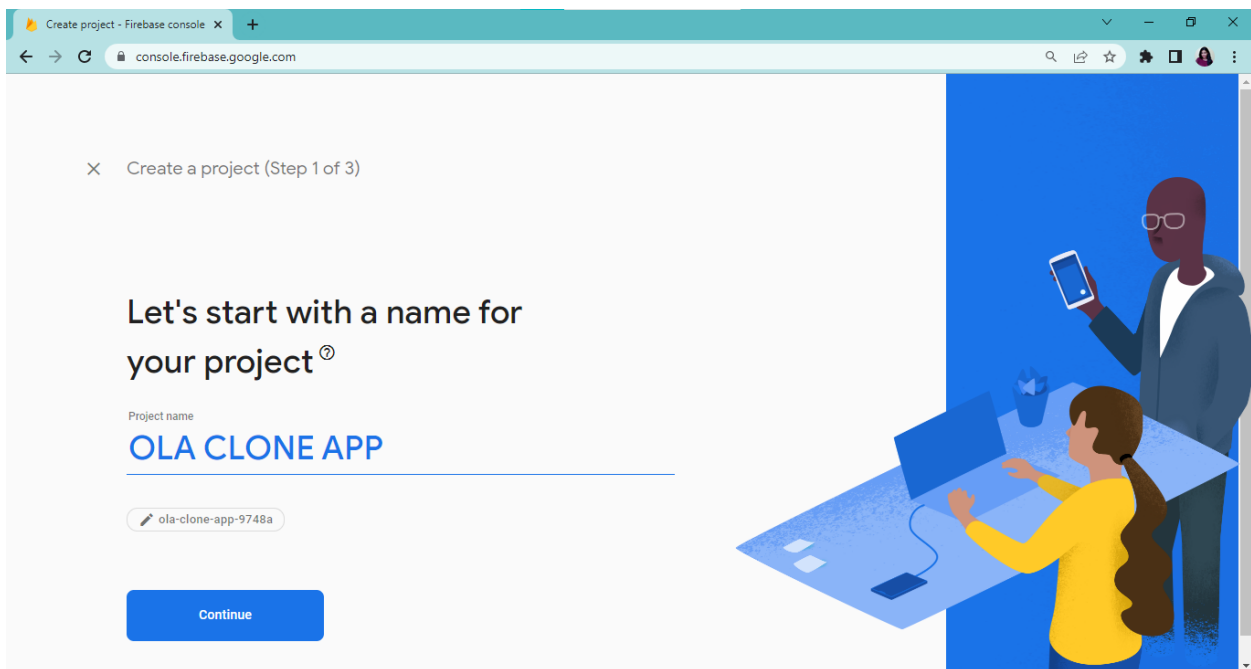
IMPLEMENTATION:

Connection with firebase

- Go to firebase console







- Click on “Add project”
- Name your project







Config, A/B Testing, and Cloud Functions.



Google Analytics enables:

 A/B testing 

 User segmentation & targeting across
Firebase products 

 Crash-free users 

 Event-based Cloud Functions triggers 

 Free unlimited reporting 



☒ **Enable Google Analytics for this project**
Recommended


[Previous](#)


[Continue](#)

Configure Google Analytics

Choose or create a Google Analytics account 

 Default Account for Firebase 

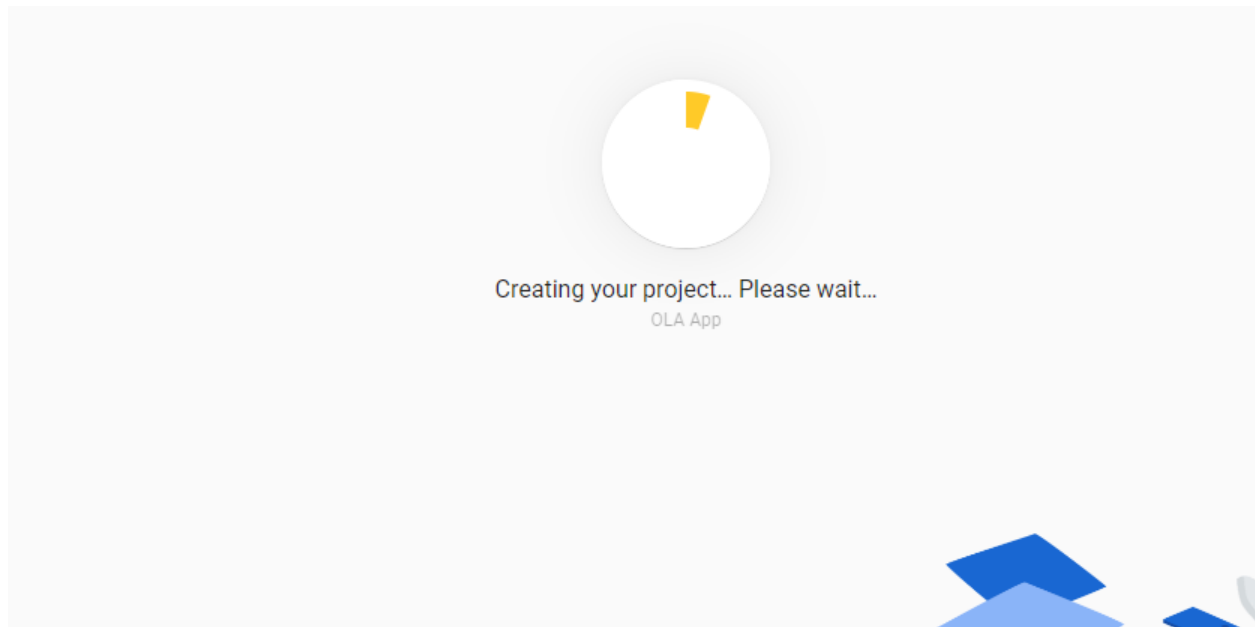
Automatically create a new property in this account 

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](#) .

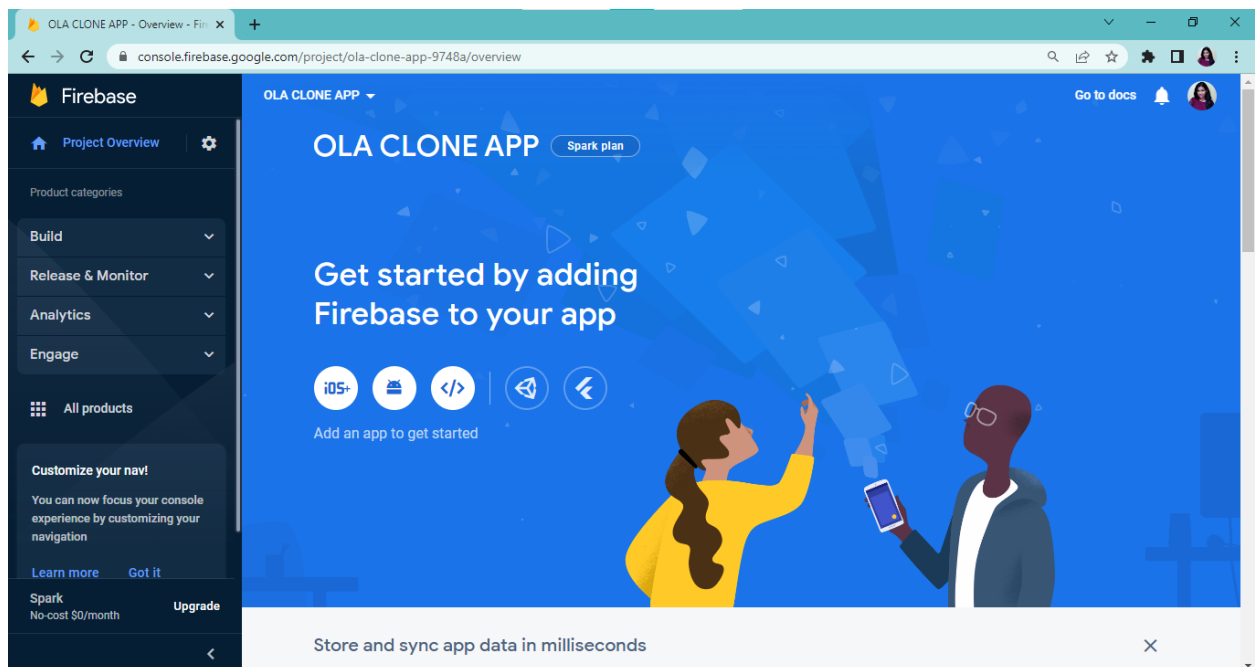
[Previous](#)

[Create project](#)

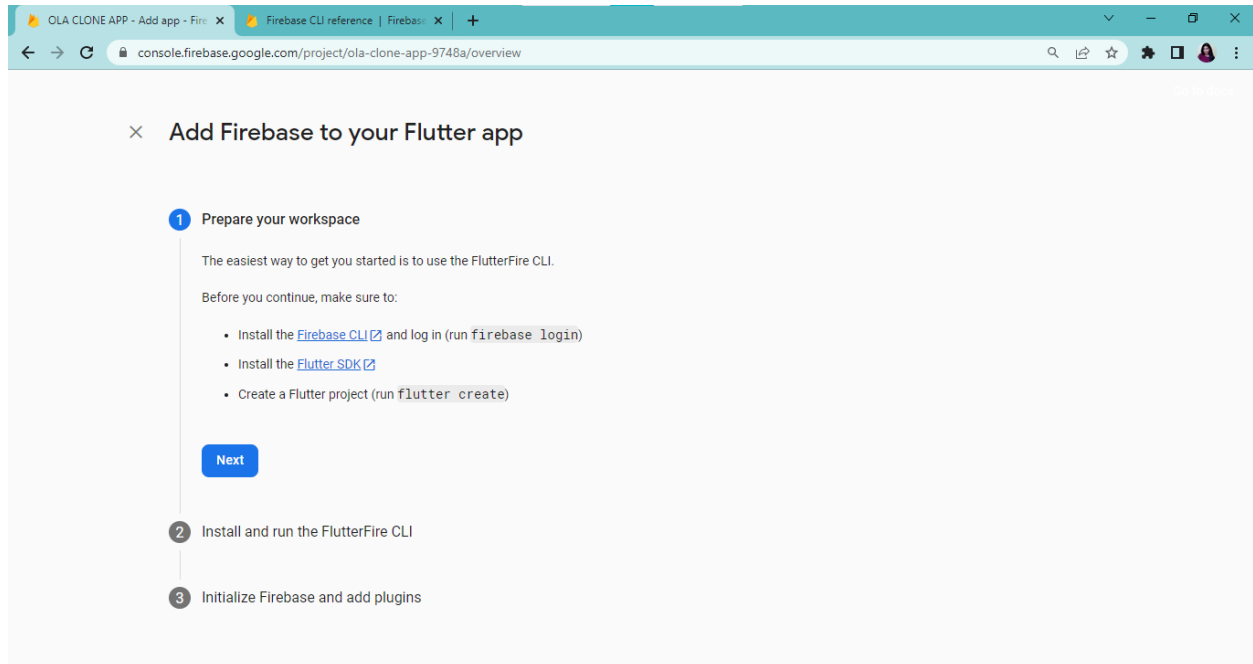
- Now click on “create project”



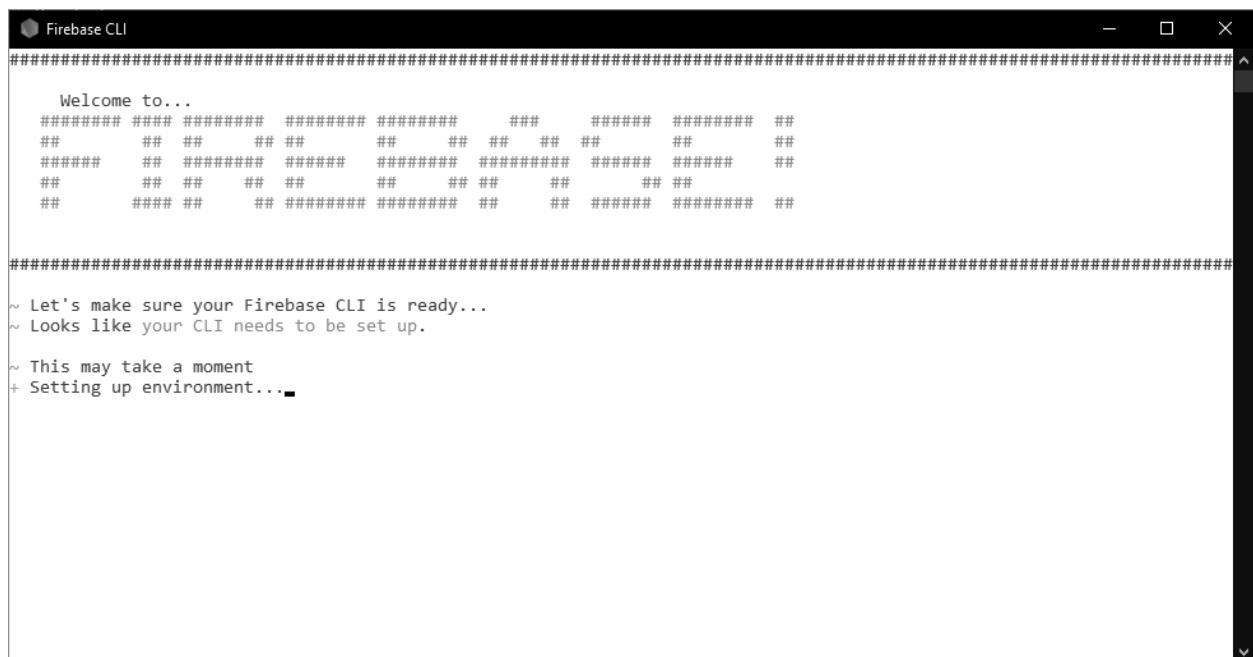
- Your project is now ready



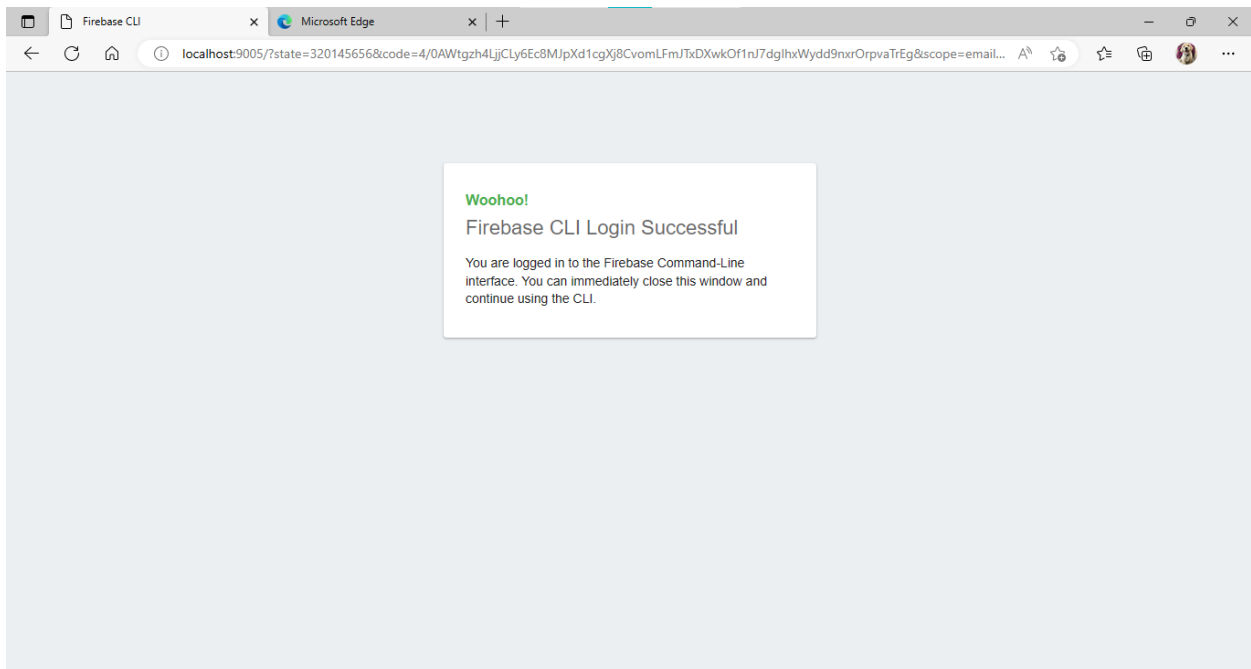
- We have to connect flutter so click on flutter icon



- Install firebase cli and flutter sdk
- Set up your firebase CLI



- **Login to the account with which you made the firebase project**



- **Make sure that you have node js installed in your system**
- **Run this command in cmd to install firebase tools**

```
Command Prompt
Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.

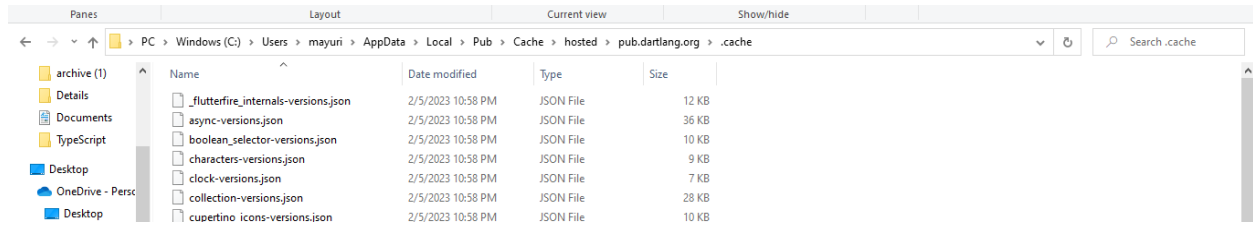
C:\Users\mayuri>npm install -g firebase-tools
npm WARN deprecated har-validator@5.1.3: this library is no longer supported
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142

added 701 packages in 53s

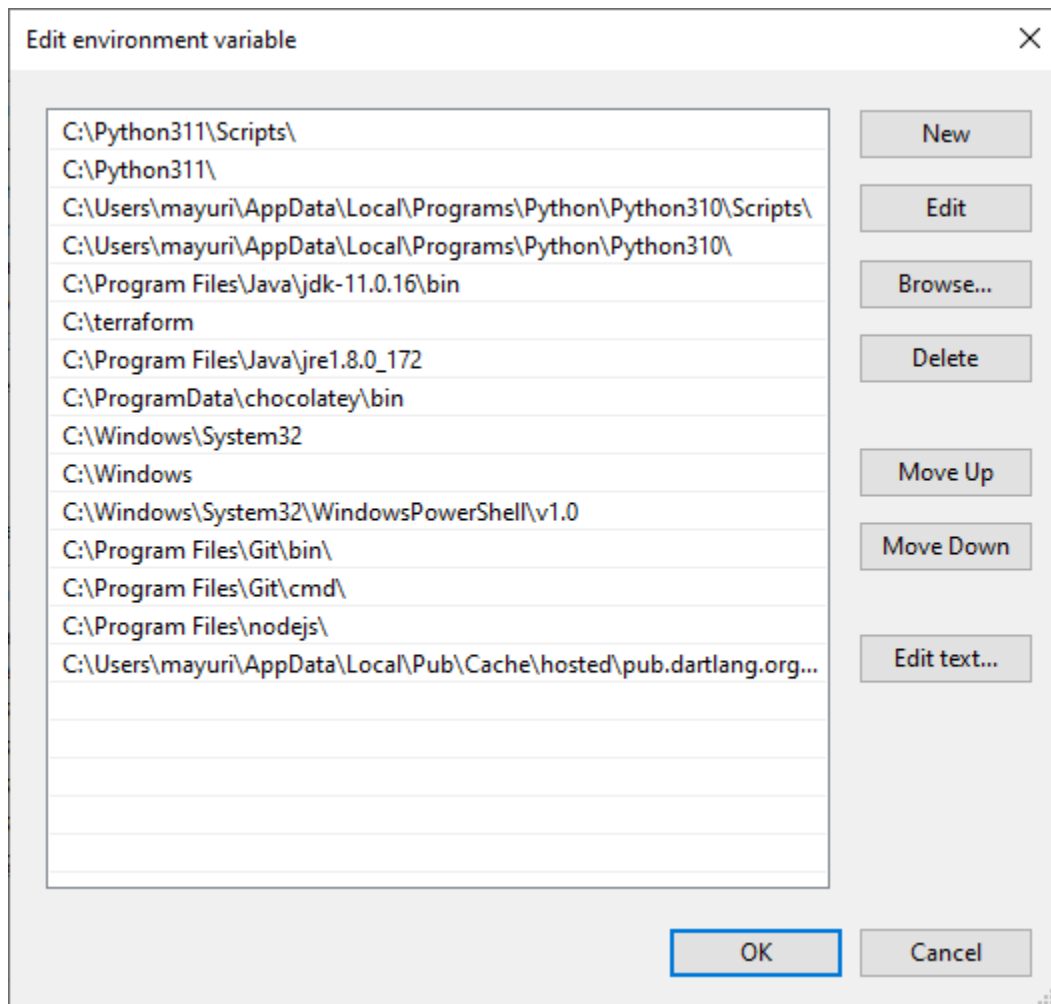
44 packages are looking for funding
  run `npm fund` for details

C:\Users\mayuri>
```

- **Copy this path**



- **Edit this path into your system variables**

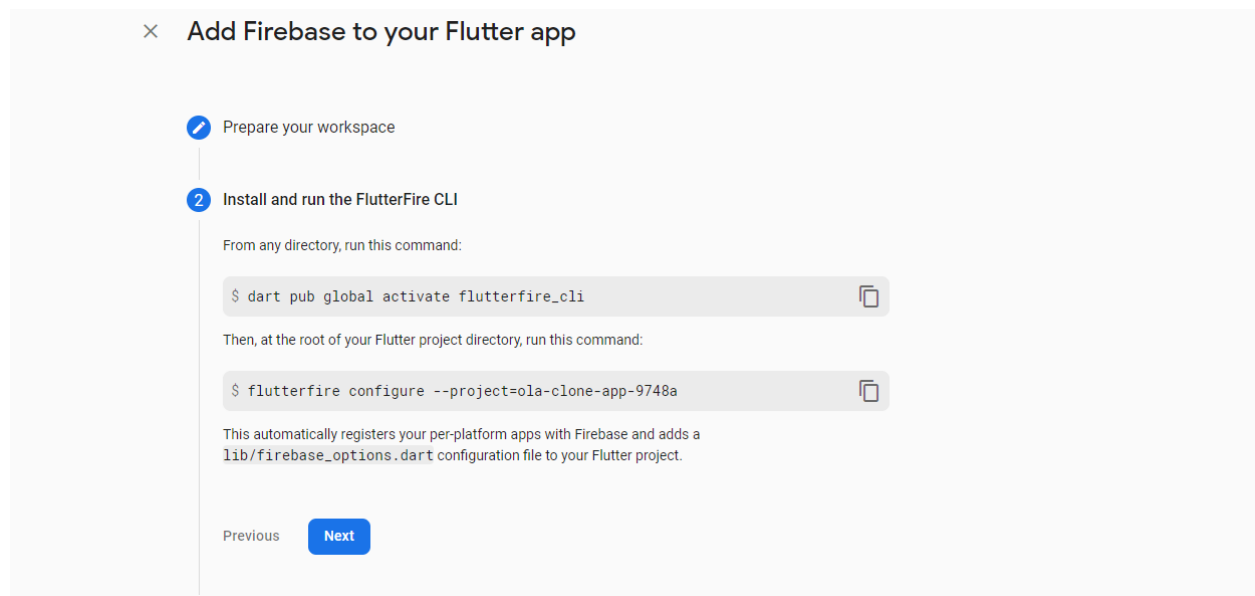


- Check if your firebase is installed or not

```
C:\Users\mayuri>firebase -V
11.22.0

C:\Users\mayuri>
```

- Now go back to flutter console



- Run the first command on cmd

```
C:\Users\mayuri>dart pub global activate flutterfire_cli
Package flutterfire_cli is currently active at version 0.2.7.
Resolving dependencies...
The package flutterfire_cli is already activated at newest available version.
To recompile executables, first run `dart pub global deactivate flutterfire_cli`.
Installed executable flutterfire.
Activated flutterfire_cli 0.2.7.

C:\Users\mayuri>
```

- Now go to your root project and open cmd and run the second command

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mayuri\StudioProjects\drivers_app>flutterfire configure --project=ola-clone-app-9748a
Found 3 Firebase projects. Selecting project ola-clone-app-9748a.
Which platforms should your configuration support (use arrow keys & space to select)? · android, ios, macos, web
Firebase android app com.mayuri.driversapp is not registered on Firebase project ola-clone-app-9748a.
Registered a new Firebase android app on Firebase project ola-clone-app-9748a.
Firebase ios app com.example.driversApp is not registered on Firebase project ola-clone-app-9748a.
Registered a new Firebase ios app on Firebase project ola-clone-app-9748a.
Firebase macos app com.example.driversApp registered.
Firebase web app drivers_app (web) is not registered on Firebase project ola-clone-app-9748a.
Registered a new Firebase web app on Firebase project ola-clone-app-9748a.
? The google-services.json file already exists but for a different Firebase project (ola-clone-app). Do you want to repl
? The google-services.json file already exists but for a different Firebase project (ola-clone-app). Do you want to repl
? The google-services.json file already exists but for a different Firebase project (ola-clone-app). Do you want to repl
ace it with Firebase project ola-clone-app-9748a? · yes

Firebase configuration file lib\firebase_options.dart generated successfully with the following Firebase apps:

Platform  Firebase App Id
web       1:527199914449:web:bad31893704552d69e78f0
android   1:527199914449:android:0fc6175937313bca9e78f0
ios       1:527199914449:ios:a8aa5dc3ab5380649e78f0
macos     1:527199914449:ios:a8aa5dc3ab5380649e78f0

Learn more about using this file and next steps from the documentation:
> https://firebase.google.com/docs/flutter/setup

C:\Users\mayuri\StudioProjects\drivers_app>
```

- Copy the two lines below

OLA CLONE APP - Add app - Fire... | Firebase CLI reference | Firebase: X +

console.firebase.google.com/project/ola-clone-app-9748a/overview

Prepare your workspace

Install and run the FlutterFire CLI

3 Initialize Firebase and add plugins

To initialize 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,
);
```

Then, add and begin using the [Flutter plugins](#) for the Firebase products you'd like to use.

Note: If you're using Analytics or Performance Monitoring, you may need to follow a few additional setup steps.

Previous [Continue to console](#)

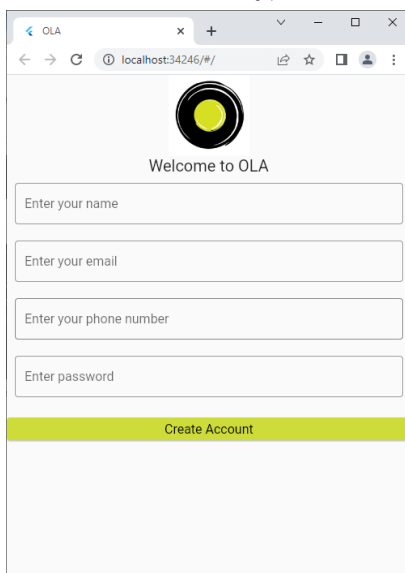
- **Paste it in your main.dart**

```
{  
  WidgetsFlutterBinding.ensureInitialized();  
  
  await Firebase.initializeApp(  
    options: DefaultFirebaseOptions.currentPlatform,  
  );  
  runApp(  
    MaterialApp(  
      title: 'OLA',  
      theme: ThemeData(  
        primarySwatch: Colors.teal,  
      ),  
      home: OLAHome(),  
    ),  
  );  
}
```

- **Import the packages in pubspec.yaml**

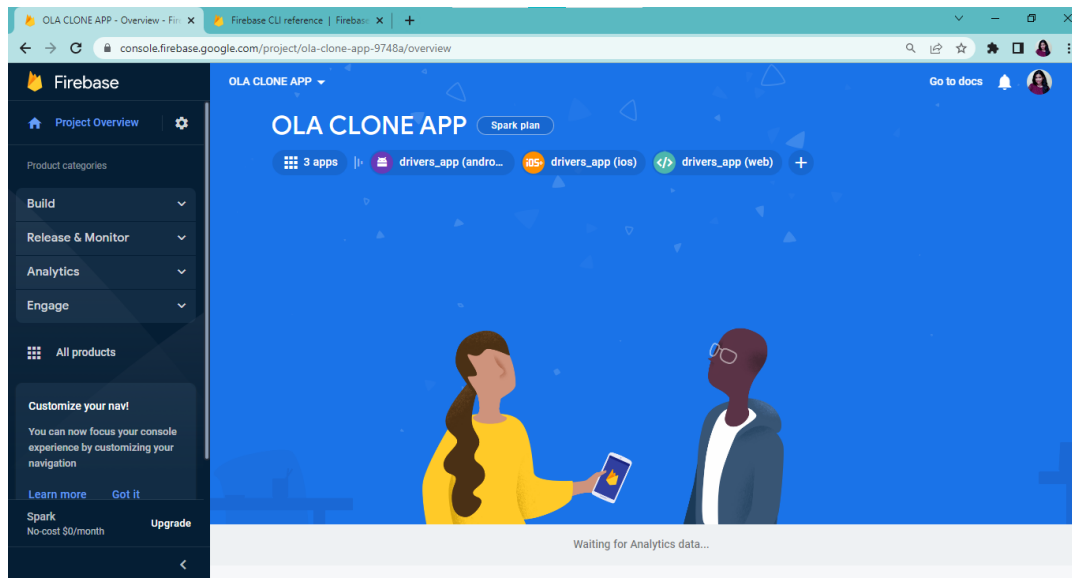
```
# Use with the CupertinoIcons class for iOS style  
cupertino_icons: ^1.0.2  
firebase_core: ^2.4.1  
firebase_auth: ^4.2.5  
firebase_database: ^10.0.9
```

- **Now run your project**
- **(If the project runs without any error, then your firebase has been connected successfully)**



The screenshot shows a web browser window with the address bar displaying 'localhost:34246/#/'. The page has a header with a circular logo containing a yellow circle and the text 'Welcome to OLA'. Below the header, there are four input fields with placeholder text: 'Enter your name', 'Enter your email', 'Enter your phone number', and 'Enter password'. At the bottom of the form, there is a yellow button labeled 'Create Account'.

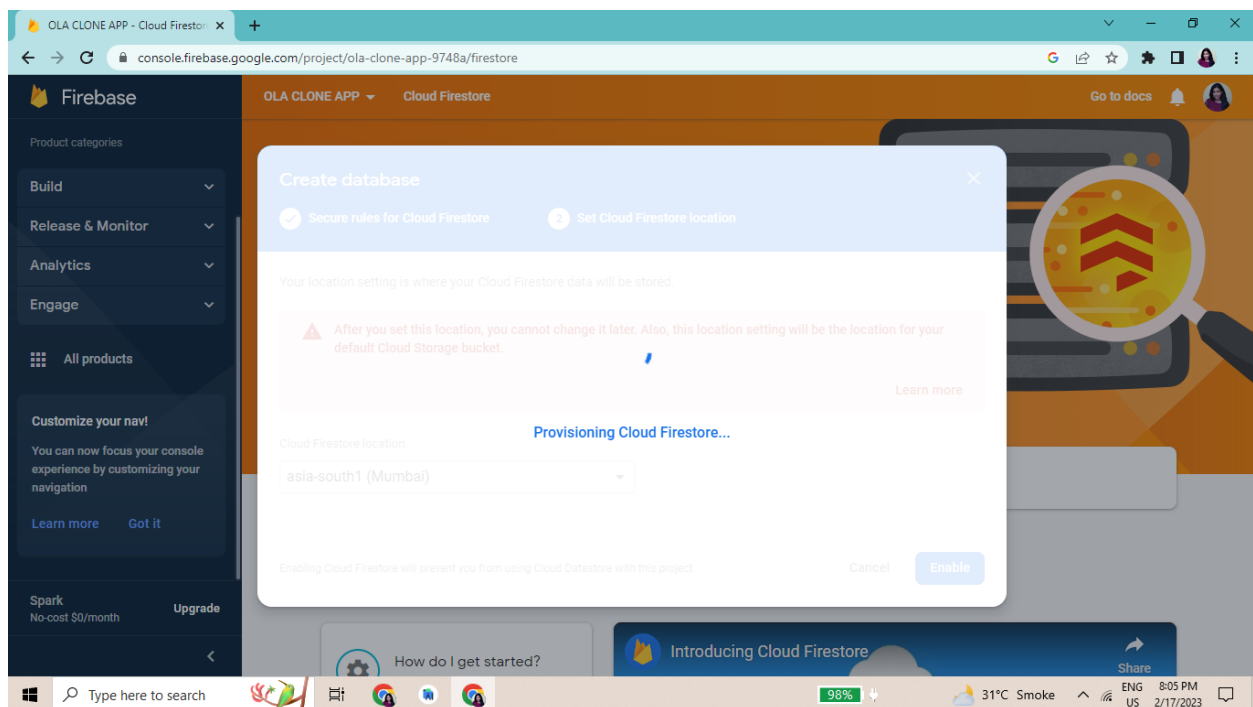
- **Refresh your Firebase console**



- **Firebase has been connected to our flutter project successfully.**

Uploading Data to Firebase

- **Create database in firestore**



- Add these dependencies

```
dependencies:
  flutter:
    sdk: flutter
  cloud_firestore: ^4.3.2

  # The following adds the Cupertino Icons font to your applicat
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  firebase_core: ^2.4.1
  firebase_auth: ^4.2.5
  firebase_database: ^4.0.0
```

➤ Connecting login screen to firebase

- Add this in your login page

```
class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  TextEditingController emailController = TextEditingController();
  TextEditingController passwordController = TextEditingController();

  final _formKey = GlobalKey<FormState>();

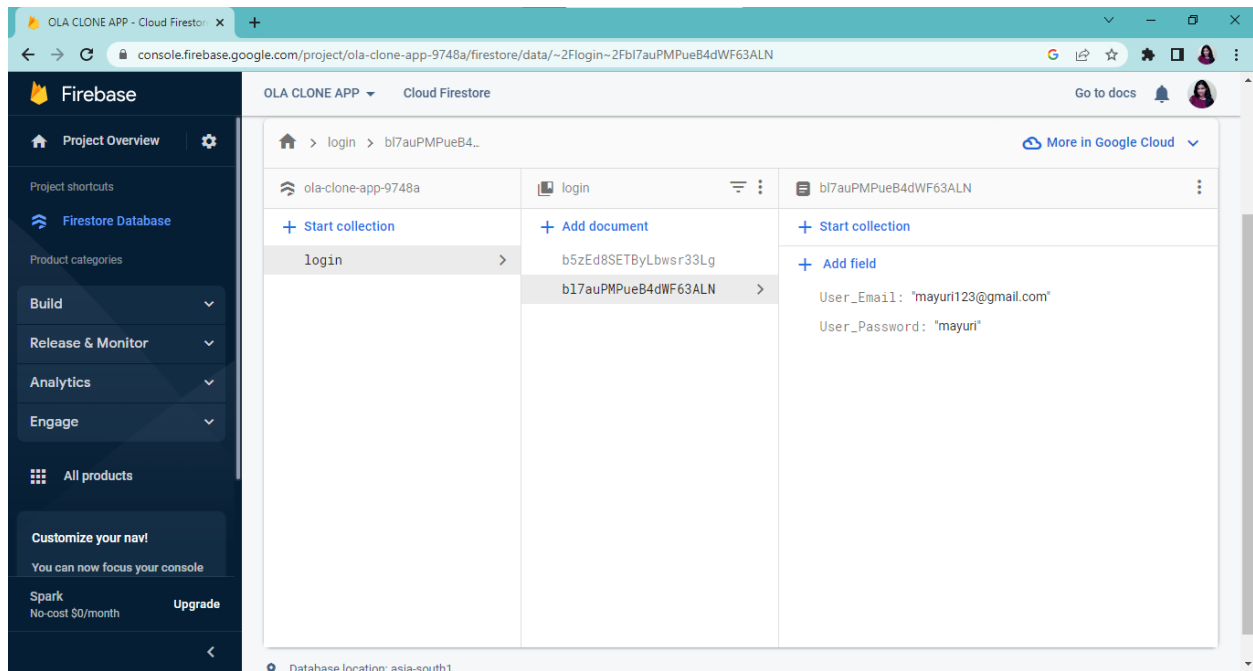
  @override
```

- Mapping our data to firebase

```
onPressed: () {
  if (_formKey.currentState!.validate()) {
    ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text("You have successfully logged in")));
  }
  Map <String,dynamic> data= {"User_Email": emailController.text,"User_Password": passwordController.text};
  FirebaseFirestore.instance.collection("login").add(data);

  Navigator.push(context, MaterialPageRoute(builder: (context) => const MainScreen())
);
}
```

- **Input data in your login screen**
- **The data is added to firestore database**



➤ Connecting to Registration Screen

Code:

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:drivers_app/authentication/signup_screen.dart';
import 'package:flutter/material.dart';
```

```
import '../mainScreens/main_screen.dart';
import 'car_info_screen.dart';
import 'login_screen.dart';
```

```
class SignUpScreen extends StatefulWidget {
  const SignUpScreen({Key? key}) : super(key: key);
```

```
  @override
  State<SignUpScreen> createState() => _SignUpScreenState();
}
```

```
class _SignUpScreenState extends State<SignUpScreen> {
```

```
  @override
  Widget build(BuildContext context) {
    return Scaffold(
```

```

        body: MyStatefulWidget(
      ),
    );
  }
}

```

```

class MyStatefulWidget extends StatefulWidget {
  const MyStatefulWidget({Key? key}) : super(key: key);

  @override
  State<MyStatefulWidget> createState() => _MyStatefulWidgetState();
}

```

```

class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  TextEditingController nameTextEditingController = TextEditingController();
  TextEditingController emailTextEditingController = TextEditingController();
  TextEditingController phoneTextEditingController = TextEditingController();
  TextEditingController passwordTextEditingController = TextEditingController();

```

```

  final _formKey = GlobalKey<FormState>();

```

```

  @override
  Widget build(BuildContext context) {
    return Form(

      key: _formKey,
      child: ListView(

        children: <Widget>[
          Container(
            width: 100,
            height: 100,
            decoration: BoxDecoration(
              image: DecorationImage(
                image: AssetImage("images/ola.png"),
              ),
            ),
          ),
          Container(
            child: Center(
              child: Text("Welcome to OLA",style: TextStyle(fontSize:(20))),),
          ),
          Container(
            padding: const EdgeInsets.fromLTRB(10, 10, 10, 10),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.start,
              children: [
                TextFormField(
                  controller: nameTextEditingController,
                  decoration: const InputDecoration(

```

```

        border: OutlineInputBorder(),
        labelText: 'Enter your name',
      ),
      validator: (value) {
        if (value == null || value.isEmpty) {
          return "Please enter some text";
        }
        return null;
      },
    ),
  ],
),
),
Container(
  padding: const EdgeInsets.fromLTRB(10, 10, 10, 10),
  child: TextFormField(
    obscureText: true,
    controller: emailTextEditingController,
    decoration: const InputDecoration(
      border: OutlineInputBorder(),
      labelText: 'Enter your email',
    ),
    validator: (value) {
      if (value == null || value.isEmpty) {
        return "Please enter some text";
      }
      return null;
    },
  ),
),
Container(
  padding: const EdgeInsets.fromLTRB(10, 10, 10, 10),
  child: TextFormField(
    obscureText: true,
    controller: phoneTextEditingController,
    decoration: const InputDecoration(
      border: OutlineInputBorder(),
      labelText: 'Enter your phone number',
    ),
    validator: (value) {
      if (value == null || value.isEmpty) {
        return "Please enter some text";
      }
      return null;
    },
  ),
),
Container(
  padding: const EdgeInsets.fromLTRB(10, 10, 10, 10),
  child: TextFormField(
    obscureText: true,

```

```

controller: passwordTextEditingController,
decoration: const InputDecoration(
  border: OutlineInputBorder(),
  labelText: 'Enter password',
),
validator: (value) {
  if (value == null || value.isEmpty) {
    return "Please enter some text";
  }
  return null;
},
),
const SizedBox(height: 15),
Container(
  child: ElevatedButton(

    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.lime,
    ),

    child: const Text(
      'Create Account',
      style: TextStyle(fontSize: (16),color: Colors.black),
    ),
    onPressed: () {
      if (_formKey.currentState!.validate()) {
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(content: Text("Processing data,Your Account has been created
successfully")));
        Navigator.push(context, MaterialPageRoute(builder: (context) => const CarInfoScreen()));
      }
      Map <String,dynamic> data= {"User_Name":
nameTextEditingController.text,"User_Email":
emailTextEditingController.text,"User_Phone_Number":
phoneTextEditingController.text,"user_Password": passwordTextEditingController};
      Firestore.instance.collection("Registration").add(data);

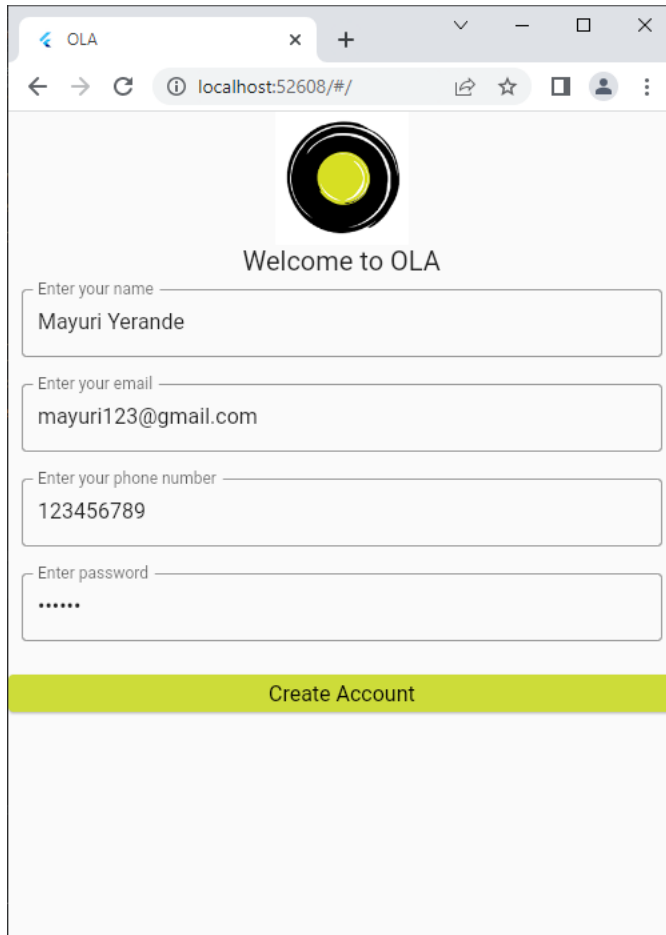
    },

  ),

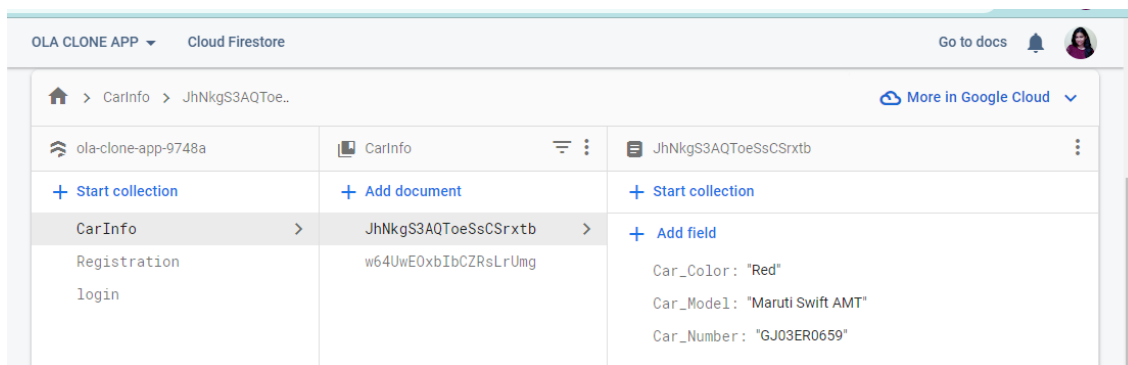
),

),
);
}
}

```

**Output:
(Input data)**

A screenshot of a web browser displaying the OLA app registration form. The browser's address bar shows 'localhost:52608/#/'. The form has a header with the OLA logo and the text 'Welcome to OLA'. Below this are four input fields: 'Enter your name' with the value 'Mayuri Yerande', 'Enter your email' with 'mayuri123@gmail.com', 'Enter your phone number' with '123456789', and 'Enter password' with masked characters '*****'. A green 'Create Account' button is positioned below the password field.

(Data reflected in database)

A screenshot of the Cloud Firestore interface showing the database structure for the OLA CLONE APP. The breadcrumb path is 'OLA CLONE APP > Cloud Firestore > CarInfo > JhNkgS3AQToe..'. The interface is divided into three main sections: 'ola-clone-app-9748a', 'CarInfo', and 'JhNkgS3AQToeSsCSrxtb'. The 'CarInfo' section shows a collection 'CarInfo' with a document 'JhNkgS3AQToeSsCSrxtb'. The 'JhNkgS3AQToeSsCSrxtb' section shows the document's fields: 'Car_Color: "Red"', 'Car_Model: "Maruti Swift AMT"', and 'Car_Number: "GJ03ER0659"'. The document ID is 'w64UwEOxbIbCZRSLrUmg'.

ola-clone-app-9748a	CarInfo	JhNkgS3AQToeSsCSrxtb
+ Start collection	+ Add document	+ Start collection
CarInfo	JhNkgS3AQToeSsCSrxtb	+ Add field
Registration	w64UwEOxbIbCZRSLrUmg	Car_Color: "Red"
login		Car_Model: "Maruti Swift AMT"
		Car_Number: "GJ03ER0659"

➤ Uploading Car Info screen data to firebase

Code:

```
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import 'login_screen.dart';
class CarInfoScreen extends StatefulWidget {
  const CarInfoScreen({Key? key}) : super(key: key);

  @override
  State<CarInfoScreen> createState() => _CarInfoScreenState();
}

class _CarInfoScreenState extends State<CarInfoScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(

      body: MyStatefulWidget(
        ),
      );
  }
}

class MyStatefulWidget extends StatefulWidget {
  const MyStatefulWidget({Key? key}) : super(key: key);

  @override
  State<MyStatefulWidget> createState() => _MyStatefulWidgetState();
}

class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  TextEditingController carModelTextEditingController = TextEditingController();
  TextEditingController carNumberTextEditingController = TextEditingController();
  TextEditingController carColorTextEditingController = TextEditingController();
  final _formKey = GlobalKey<FormState>();

  @override
  Widget build(BuildContext context) {
    return Form(

      key: _formKey,
      child: ListView(

        children: <Widget>[
          Container(
            width: 600,
            height: 200,
            decoration: BoxDecoration(
```

```

        image: DecorationImage(
          image: AssetImage("images/car_info.png"),
        ),
      ),
    ),
    Container(
      child: Center(
        child: Text("Enter Car Details",style: TextStyle(fontSize:(20)),),
      ),
    ),
    Container(
      padding: const EdgeInsets.fromLTRB(10, 10, 10, 10),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          TextFormField(
            controller: carModelTextEditingController,
            decoration: const InputDecoration(
              border: OutlineInputBorder(),
              labelText: 'Car Model',
            ),
          ),
          validator: (value) {
            if (value == null || value.isEmpty) {
              return "Please enter some text";
            }
            return null;
          },
        ],
      ),
    ),
    Container(
      padding: const EdgeInsets.fromLTRB(10, 10, 10, 10),
      child: TextFormField(
        controller: carNumberTextEditingController,
        decoration: const InputDecoration(
          border: OutlineInputBorder(),
          labelText: 'Car Number',
        ),
      ),
      validator: (value) {
        if (value == null || value.isEmpty) {
          return "Please enter some text";
        }
        return null;
      },
    ),
    Container(
      padding: const EdgeInsets.fromLTRB(10, 10, 10, 10),
      child: TextFormField(
        controller: carColorTextEditingController,

```

```

decoration: const InputDecoration(
  border: OutlineInputBorder(),
  labelText: 'Car Color',
),
validator: (value) {
  if (value == null || value.isEmpty) {
    return "Please enter some text";
  }
  return null;
},
),
const SizedBox(height: 15),
Container(
  child: ElevatedButton(

    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.lime,
    ),

    child: const Text(
      'Next',
      style: TextStyle(fontSize: (16),color: Colors.black),
    ),
    onPressed: () {
      if (_formKey.currentState!.validate()) {
        ScaffoldMessenger.of(context).showSnackBar(
          const SnackBar(content: Text("Processing data")));

      }
      Map <String,dynamic> data2 = {"Car_Model":
carModelTextEditingController.text,"Car_Color":carColorTextEditingController.text,"Car_Numb
er": carNumberTextEditingController.text};
Firestore.instance.collection("CarInfo").add(data2);

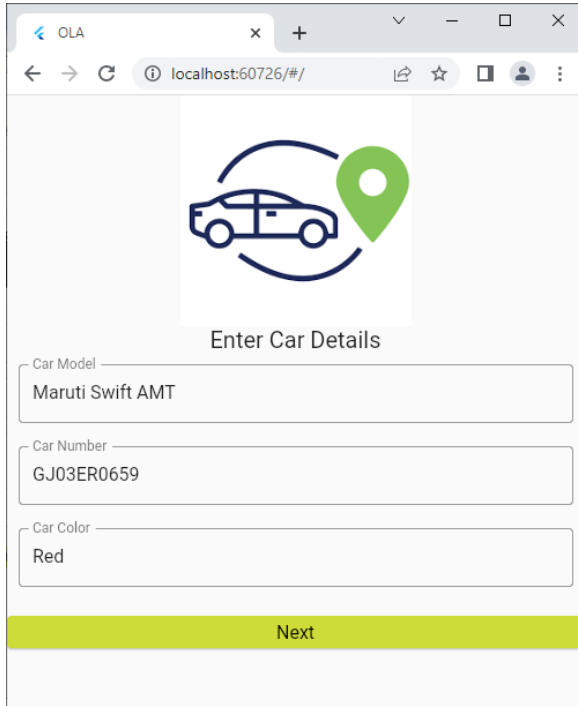
      Navigator.push(context, MaterialPageRoute(builder: (context) => const LoginScreen()))
    );

  },

),
const SizedBox(height: 15),


],
));
}
}

```

Input:


OLA

localhost:60726/#/



Enter Car Details

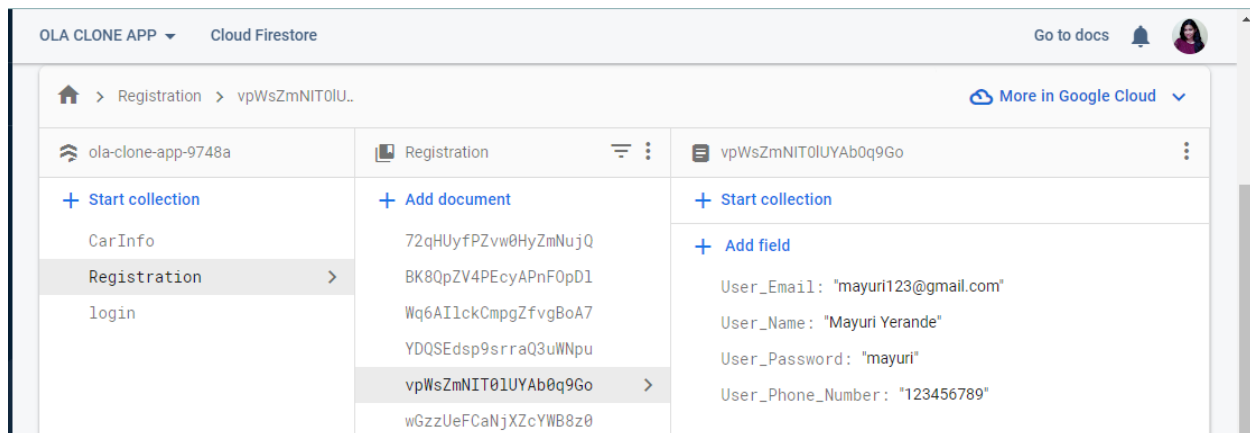
Car Model
Maruti Swift AMT

Car Number
GJ03ER0659

Car Color
Red

Next

- **Data reflected in database**



CONCLUSION: We created a project in the firebase console. Then we imported the required packages to run it and Then we connected our project to firebase with some commands mentioned. The data which we enter in our App is successfully getting added to our database. Thus we successfully connected our project to firebase.