**Vivekanand Education Society’s** 

**Institute of Technology**

**(Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)**

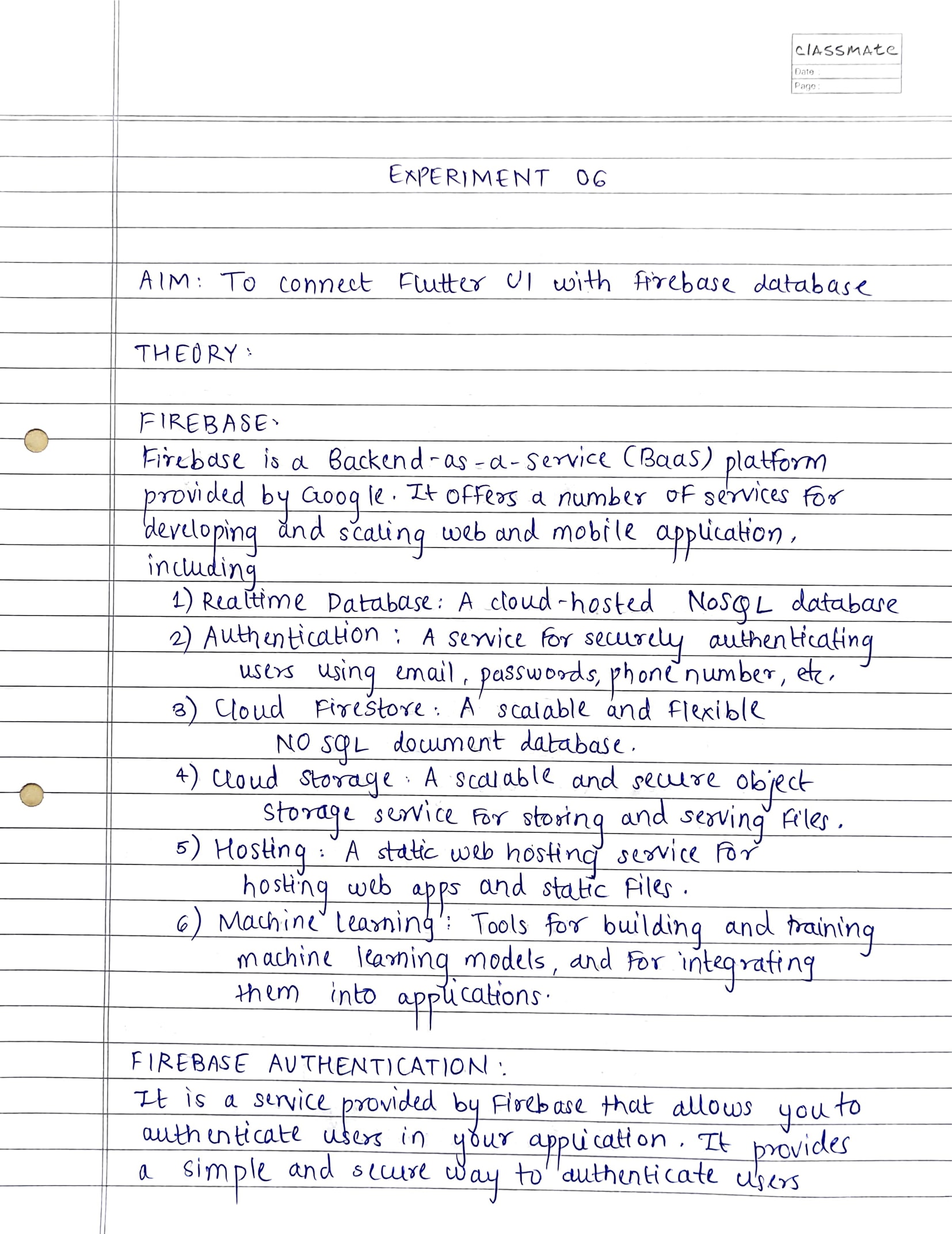
**Department of Information Technology**

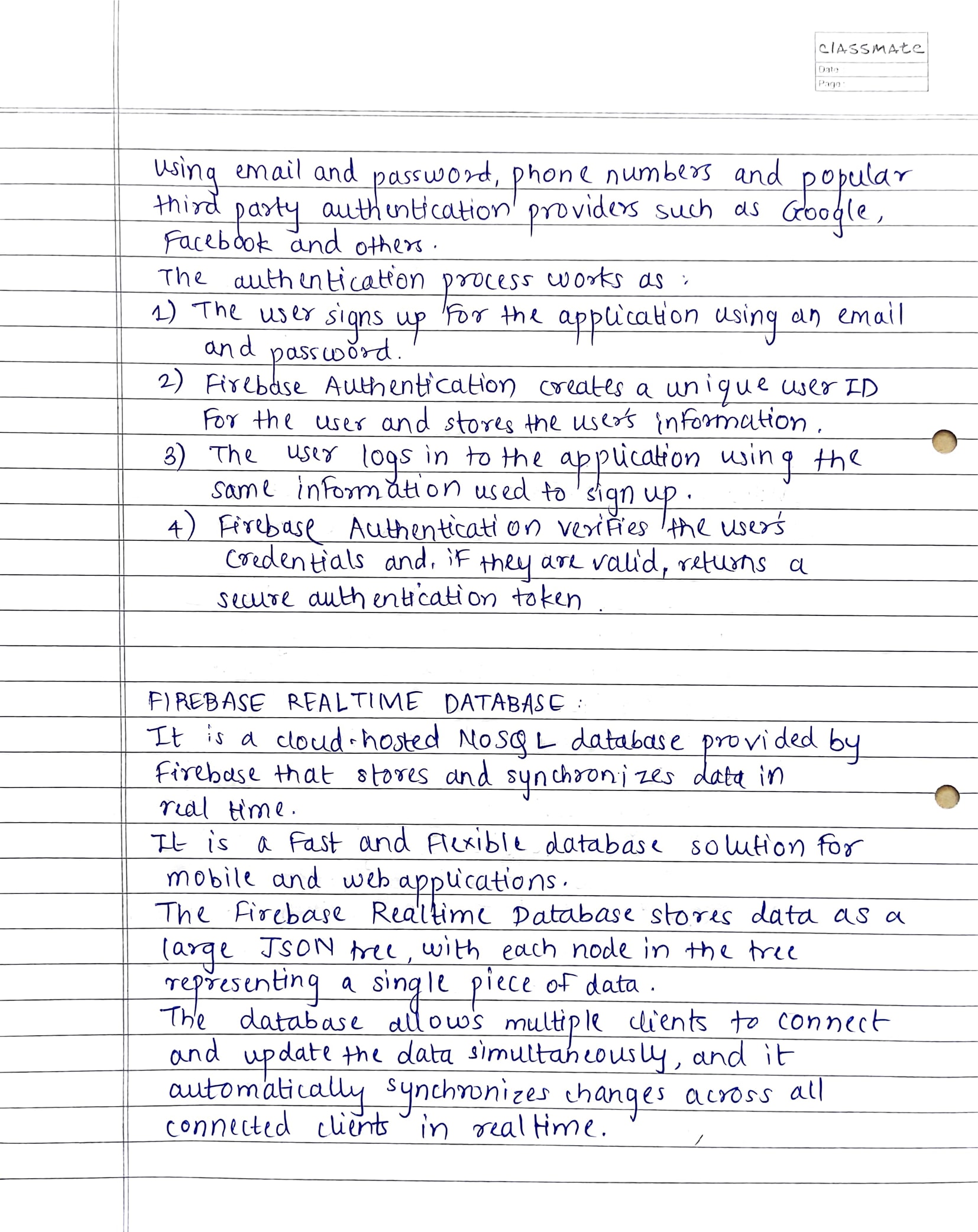
A.Y. 22-23

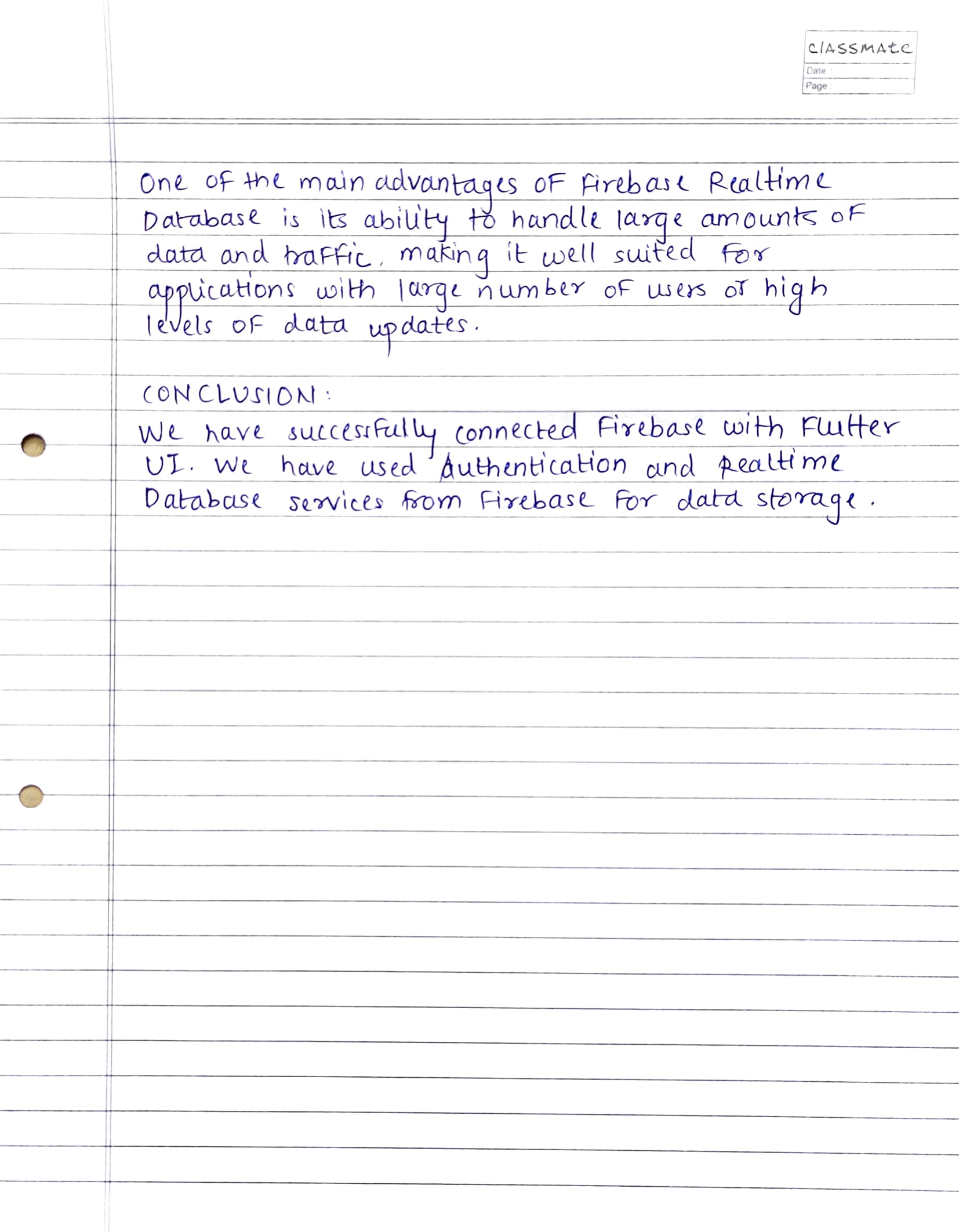
MAD & PWA Lab

Journal

| Experiment No. | 06 |
| --- | --- |
| Experiment  Title. | To Connect Flutter UI with fireBase database |
| Roll No. | 53 |
| Name | SAKSHI SANTOSH PATIL |
| Class | D15B |
| Subject | MAD & PWA Lab |
| Lab Outcome | LO3: Analyze and Build production ready Flutter App by incorporating backend services and deploying on Android / iOS |
| Grade: |  |

****

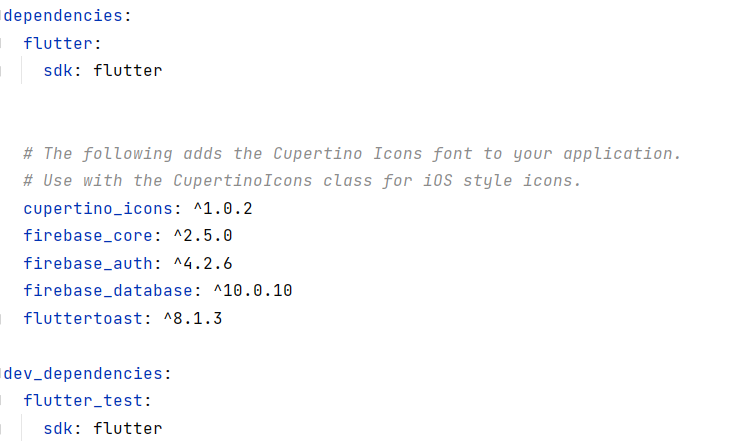
****

****

**AIM:** To Connect Flutter UI with fireBase database

**CODE:**

**pubspec.yaml**



**main.dart**

import 'package:firebase\_database/firebase\_database.dart';

import 'package:flutter/material.dart';

import 'package:uber\_clone/screens/mainScreen.dart';

import 'package:uber\_clone/screens/splashScreen.dart';

import 'package:firebase\_core/firebase\_core.dart';

void main() async{

WidgetsFlutterBinding.ensureInitialized();

await Firebase.initializeApp(

options: FirebaseOptions(

apiKey: 'AIzaSyCHoqhVy6nLaB1ji8jDFuOikr78DVxCUuo',

databaseURL: 'https://uber-clone-flutter-44d1f-default-rtdb.firebaseio.com/',

appId: '1:122988553953:android:5934c15c4928fbccb6f9a0',

messagingSenderId: '122988553953',

projectId: 'uber-clone-flutter-44d1f'

)

);

runApp(const MyApp());

}

DatabaseReference usersRef = FirebaseDatabase.instance.ref().child("users"); //creates a reference to the "users" child node in the Firebase Realtime Database.

//With this reference, you can perform various operations, such as writing data, reading data, and updating data, in the "users" node of the database.

class MyApp extends StatelessWidget {

const MyApp({Key? key}) : super(key: key);

@override

Widget build(BuildContext context) {

return MaterialApp(

title: 'Uber',

theme: ThemeData(

fontFamily: "Brand-Bold",

primarySwatch: Colors.blue,

visualDensity: VisualDensity.adaptivePlatformDensity),

home: SplashScreen(),

routes: {

'/mainScreen': (context) => const MainScreen()

},

debugShowCheckedModeBanner: false,

);

}

}

**registrationScreen.dart**

import 'package:firebase\_auth/firebase\_auth.dart';

import 'package:flutter/material.dart';

import 'package:uber\_clone/main.dart';

import 'package:uber\_clone/screens/loginScreen.dart';

class registrationScreen extends StatefulWidget {

const registrationScreen({Key? key}) : super(key: key);

@override

State<registrationScreen> createState() => \_registrationScreenState();

}

class \_registrationScreenState extends State<registrationScreen> {

final \_formkey = GlobalKey<FormState>();

TextEditingController nameTextEditingController = TextEditingController();

TextEditingController emailTextEditingController = TextEditingController();

TextEditingController phoneTextEditingController = TextEditingController();

TextEditingController passwordTextEditingController = TextEditingController();

@override

Widget build(BuildContext context) {

return Form(

key: \_formkey,

child: Scaffold(

resizeToAvoidBottomInset : false,

backgroundColor: Colors.white,

body: SingleChildScrollView(

child: Padding(

padding: const EdgeInsets.all(8.0),

child: Column(

children: [

SizedBox(height: 40.0,),

Image(image: AssetImage("images/Uber-logo-255-150.jpg"),

height: 140.0,

width: 390.0,

alignment: Alignment.center,),

// SizedBox(height: 0.0),

Text(

"Signup",

style: TextStyle(fontSize: 26.0, fontFamily: "Brand-Bold"),

textAlign: TextAlign.center,

),

Padding(padding: EdgeInsets.all(45.0),

child: Column(

children: [

// SizedBox(height: 1.0,),

TextFormField(

controller: nameTextEditingController,

validator: (value){

if(value==null || value.isEmpty){

return 'Please enter some text';

}

return null;

},

keyboardType: TextInputType.text,

decoration: InputDecoration(

labelText: "Name",

labelStyle: TextStyle(

fontSize: 14.0,

),

),

style: TextStyle(fontSize: 14.0),

),

TextFormField(

controller: emailTextEditingController,

validator: (value){

if(value==null || value.isEmpty || !RegExp(r"^[a-zA-Z0-9.a-zA-Z0-9.!#$%&'\*+-/=?^\_`{|}~]+@[a-zA-Z0-9]+\.[a-zA-Z]+").hasMatch(value)){

return 'Please enter valid email';

}

return null;

},

keyboardType: TextInputType.emailAddress,

decoration: InputDecoration(

labelText: "Email",

labelStyle: TextStyle(

fontSize: 14.0,

),

),

style: TextStyle(fontSize: 14.0),

),

TextFormField(

controller: phoneTextEditingController,

validator: (value){

if(value==null || value.isEmpty || !RegExp(r"[0-9]").hasMatch(value)){

return 'Please enter valid phone number';

}

return null;

},

keyboardType: TextInputType.phone,

decoration: InputDecoration(

labelText: "Phone number",

labelStyle: TextStyle(

fontSize: 14.0,

),

),

style: TextStyle(fontSize: 14.0),

),

SizedBox(height: 1.0,),

TextFormField(

controller: passwordTextEditingController,

validator: (value){

if(value==null || value.isEmpty){

return 'Please enter some text';

}

return null;

},

keyboardType: TextInputType.emailAddress,

obscureText:true,

decoration: InputDecoration(

labelText: "Password",

labelStyle: TextStyle(

fontSize: 14.0,

),

),

style: TextStyle(fontSize: 14.0),

),

SizedBox(height: 40.0,),

ElevatedButton(

onPressed: () {

if(\_formkey.currentState!.validate()){

ScaffoldMessenger.of(context).showSnackBar(

const SnackBar(content: Text('Processing Data'))

);

}

registerNewUser(context);

// print("Reister BUTTON CLICKED");

},

style: ElevatedButton.styleFrom(

shape: RoundedRectangleBorder(

borderRadius: BorderRadius.circular(30.0)

),

backgroundColor: Colors.black,

),

child: Container(

height: 50.0,

child: Center(

child: Text(

"Register",

style: TextStyle(color:Colors.white,fontSize: 18.0,fontFamily: "Brand-Bold"),

),

),

),

)

],

),

),

TextButton(

onPressed: (){

Navigator.push(

context,

MaterialPageRoute(builder: (context) => const loginScreen()));

},

child: Text(

"Already have an account? Login here.",

style: TextStyle(color: Colors.grey),

)

)

],

),

),

),

),

);

}

final FirebaseAuth \_firebaseAuth = FirebaseAuth.instance;

void registerNewUser(BuildContext context) async

{

final User? firebaseUser = (await \_firebaseAuth //final FirebaseUser converted to final User?

.createUserWithEmailAndPassword(

email: emailTextEditingController.text,

password:passwordTextEditingController.text

).catchError((errMsg){

print("User account not created.");

})).user;

if(firebaseUser != null){ //user created

//saver userinfo to database

Map userDataMap = {

"name": nameTextEditingController.text.trim(),

"email": emailTextEditingController.text.trim(),

"phone": phoneTextEditingController.text.trim(),

};

usersRef.child(firebaseUser.uid).set(userDataMap);

print("Account created");

Navigator.pushNamedAndRemoveUntil(context, '/mainScreen', (route) => false);

}

else{

//error occured

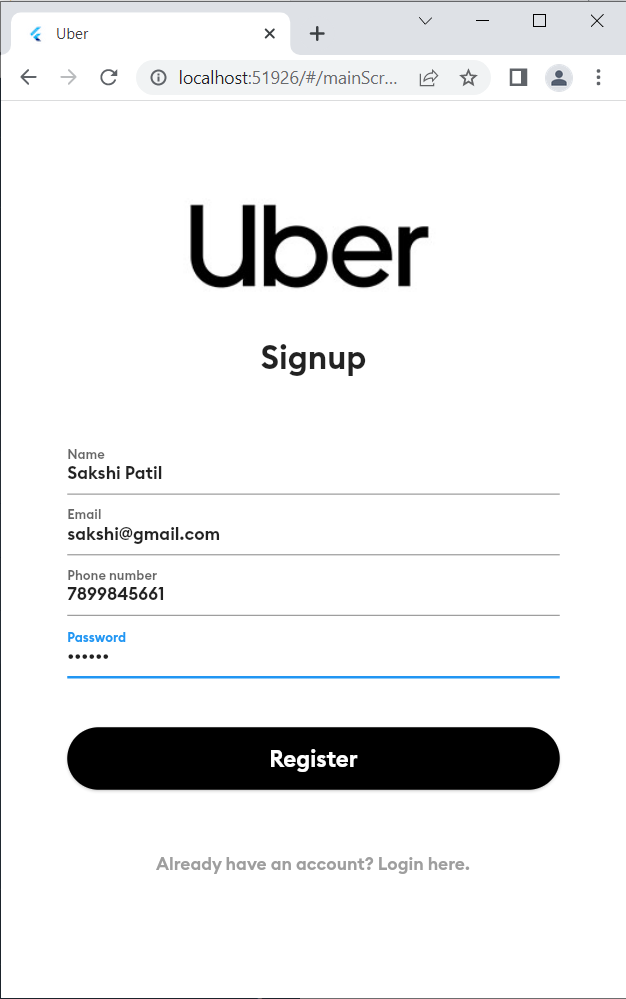
print("Error occured");

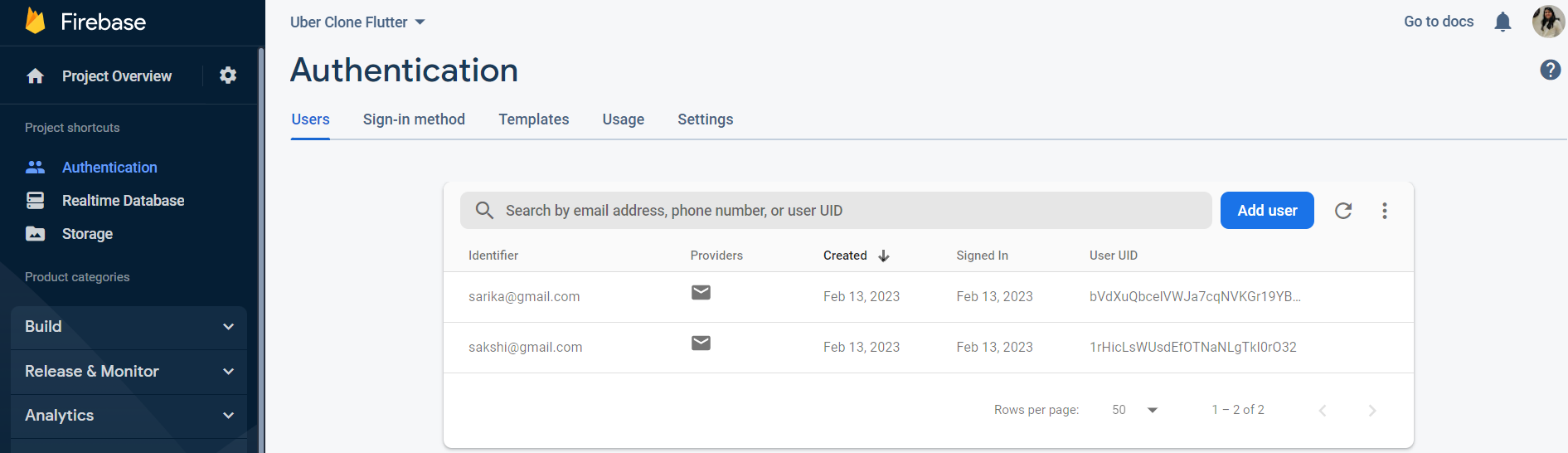
}

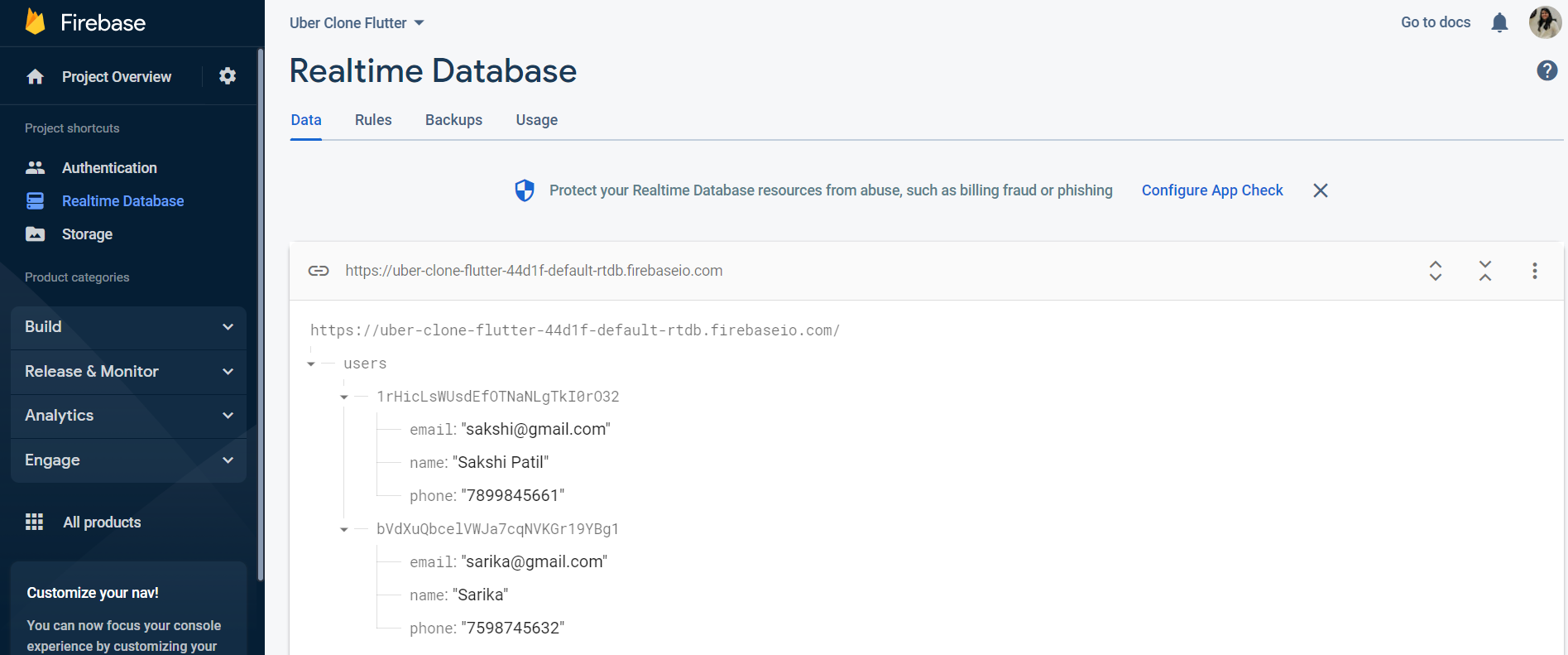
}

}

**SCREENSHOTS:**

****

****

****