PRACTICAL 6

Aim: To Connect Flutter UI with FireBase database.

Theory:

In modern mobile applications, user authentication and data management play a crucial role. Firebase, a backend-as-a-service platform by Google, simplifies authentication and database management for Flutter applications. Firebase Authentication allows secure sign-in using different methods such as email/password, phone authentication, and third-party providers (Google, Facebook, etc.). In this experiment, we integrate a Flutter UI with Firebase to enable user signup, login, password reset, and data storage in Firestore.

Implementation in Our Code

1. User Signup
   - Users enter email, password, and username.
   - Email format and password length are validated.
   - Firebase creates an account and stores user details in Firestore.
   - A verification email is sent to confirm authenticity.

2. Email Verification
   - Users must verify their email before logging in.
   - The app restricts access until verification is complete.

3. User Login
   - Users log in with email and password.
   - Firebase checks credentials and verification status before granting access.

4. Password Reset
   - Users can request a password reset email if they forget their password.

5. Form Validation
   - Ensures correct email format, password length (min 8 chars), and non-empty fields.

CODE

Folder Structure

```
lib/
├── reuseable_widgets/
│   └── reuseable_widgets.dart
├── screens/
│   ├── features/
│   │   ├── add_medication_reminder_screen.dart
│   │   ├── caregiver_connection_screen.dart
│   │   ├── emergency_assistance_screen.dart
│   │   ├── health_records_screen.dart
│   │   └── medication_reminders_screen.dart
│   ├── settings/
│   │   ├── font_size_settings.dart
│   │   ├── language_settings.dart
│   │   └── settings_screen.dart
│   ├── home_screen.dart
│   ├── login_screen.dart
│   ├── reset_password.dart
│   └── signup_screen.dart
├── util/
│   └── color_util.dart
├── widgets/
│   └── custom_text_field.dart
├── firebase_options.dart
└── main.dart
```

signup_screen.dart:

```
void _signupUser() {
  if (_formKey.currentState?.validate() ?? false) {
    setState(() {
      _isLoading = true;
    });

    FirebaseAuth.instance
        .createUserWithEmailAndPassword(
      email: _emailTextController.text.trim(),
      password: _passwordTextController.text.trim(),
    )
        .then((value) async {
      User? user = value.user;

      if (user != null) {
        // Save user details to Firestore
        await FirebaseFirestore.instance.collection('users').doc(user.uid).set({
          'username': _userNameTextController.text.trim(),
          'email': _emailTextController.text.trim(),
        });
        await user.sendEmailVerification(); // Ensure email verification
      }

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

      print("Created new account");

      // Navigate to HomeScreen
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => HomeScreen()),
      );
    }).catchError((error) {
      setState(() {
        _isLoading = false;
      });
      print("Error: ${error.toString()}");
      _showErrorDialog("Failed to create an account. Please try again.");
    });
  }
}
```

login_screen.dart:

```
 Future<void> _loginUser(BuildContext context) async {
    try {
      showDialog(
        context: context,
        barrierDismissible: false,
        builder: (context) => Center(child: CircularProgressIndicator()),
      );

      // Attempt to log in
      UserCredential userCredential =
          await FirebaseAuth.instance.signInWithEmailAndPassword(
        email: _emailTextController.text.trim(),
        password: _passwordTextController.text.trim(),
      );

      User? user = userCredential.user;
      // Check if the email is verified
      if (user != null && !user.emailVerified) {
        await user.sendEmailVerification(); // Send verification email
        Navigator.pop(context);
        _showErrorDialog(
          context,
          "Please verify your email. A verification link has been sent to your email.",
        );
        return;
      }

      if (user != null) {
        print("Logged in successfully: ${user.email}");
      }

      // Navigate to HomeScreen after successful login
      Navigator.pop(context);
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => HomeScreen()),
      );
    } catch (error) {
      Navigator.pop(context);
      _showErrorDialog(context, error.toString());
    }
  }
```

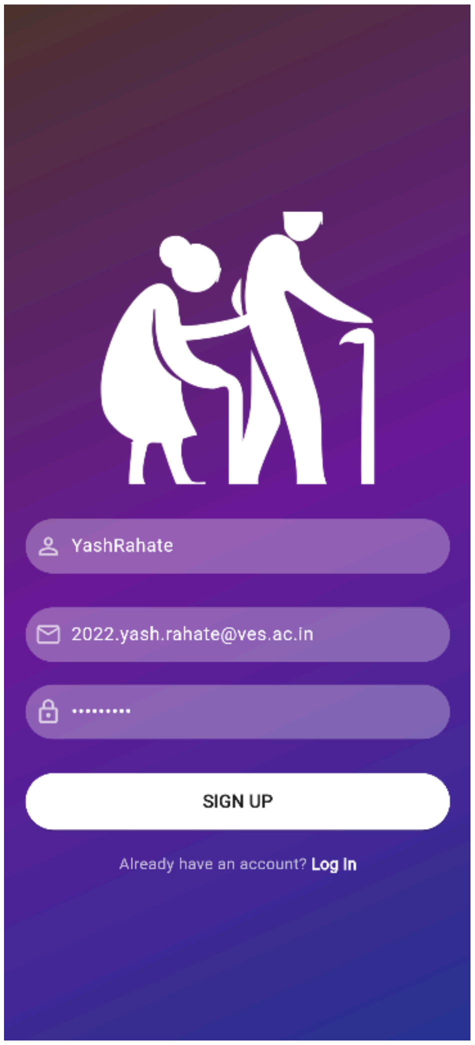
reset_password.dart:

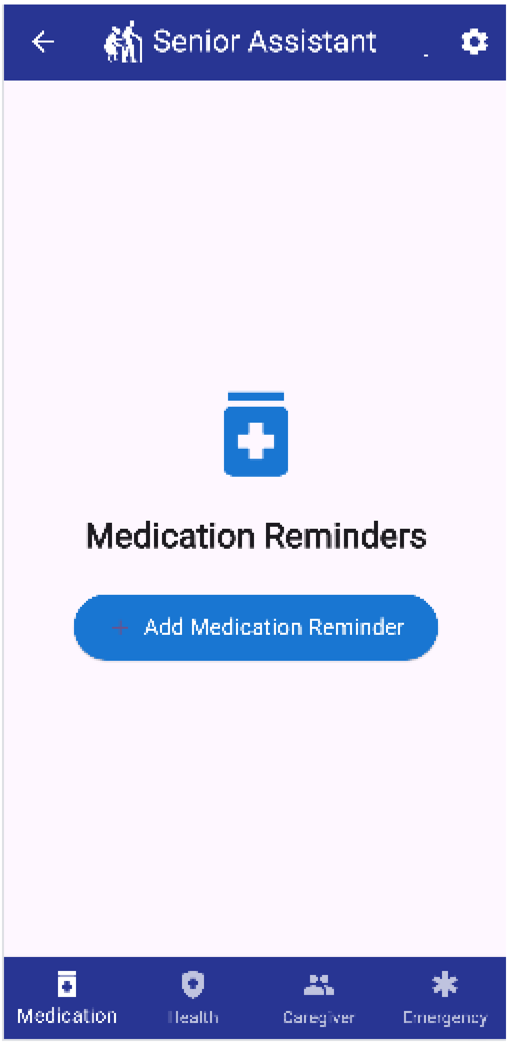
```
 void _resetPassword() {
    final email = _emailTextController.text.trim();

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

    FirebaseAuth.instance
        .sendPasswordResetEmail(email: email)
        .then((value) {
      setState(() {
        _isLoading = false;
      });
      _showSnackbar("Password reset email sent successfully!");
      Navigator.of(context).pop();
```

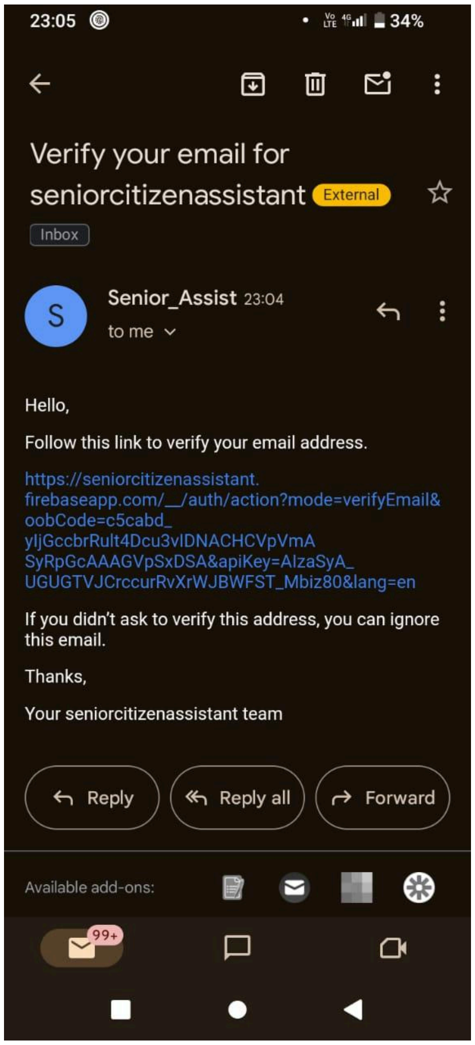
```
  }).catchError((error) {
    setState(() {
      _isLoading = false;
    });
    _showSnackbar("Error: ${error.message}");
  });
 }
```

VALIDATION code:

```
(value) {
                  if (value == null || value.isEmpty) {
                    return "Please enter your email";
                  }
                  if (!RegExp(r'^[^@]+@[^@]+\.[^@]+').hasMatch(value)) {
                    return "Enter a valid email address";
                  }
                  return null;
                },

(value) {
                  if (value == null || value.isEmpty) {
                    return 'Password is required';
                  }
                  if (value.length < 8) {
                    return 'Password must be at least 8 characters long';
                  }
                  return null;
                },
```


YashRahate
2022.yash.rahate@ves.ac.in
•••••••••
SIGN UP
Already have an account? Log In


Senior Assistant
Medication Reminders
Add Medication Reminder
Medication
Health
Caregiver
Emergency



23:05 34%

Verify your email for seniorcitizenassistant External

Inbox

Senior_Assist 23:04
to me

Hello,

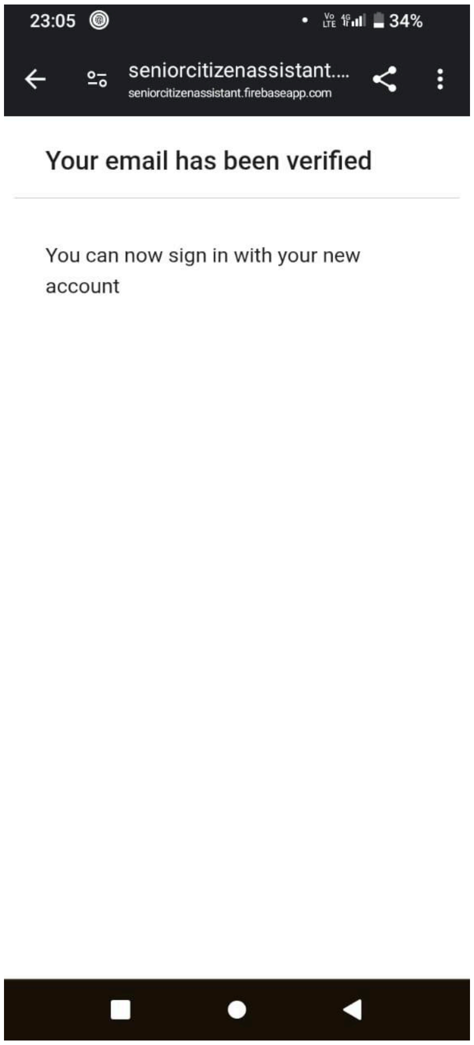
Follow this link to verify your email address.

https://seniorcitizenassistant.firebaseapp.com/__/auth/action?mode=verifyEmail&oobCode=c5cabd_yljGccbrRult4Dcu3vlDNACHCVpVmASyRpGcAAAGVpSxDSA&apiKey=AIzaSyA_UGUGTVJCrccurRvXrWJBWFST_Mbiz80&lang=en

If you didn't ask to verify this address, you can ignore this email.

Thanks,

Your seniorcitizenassistant team

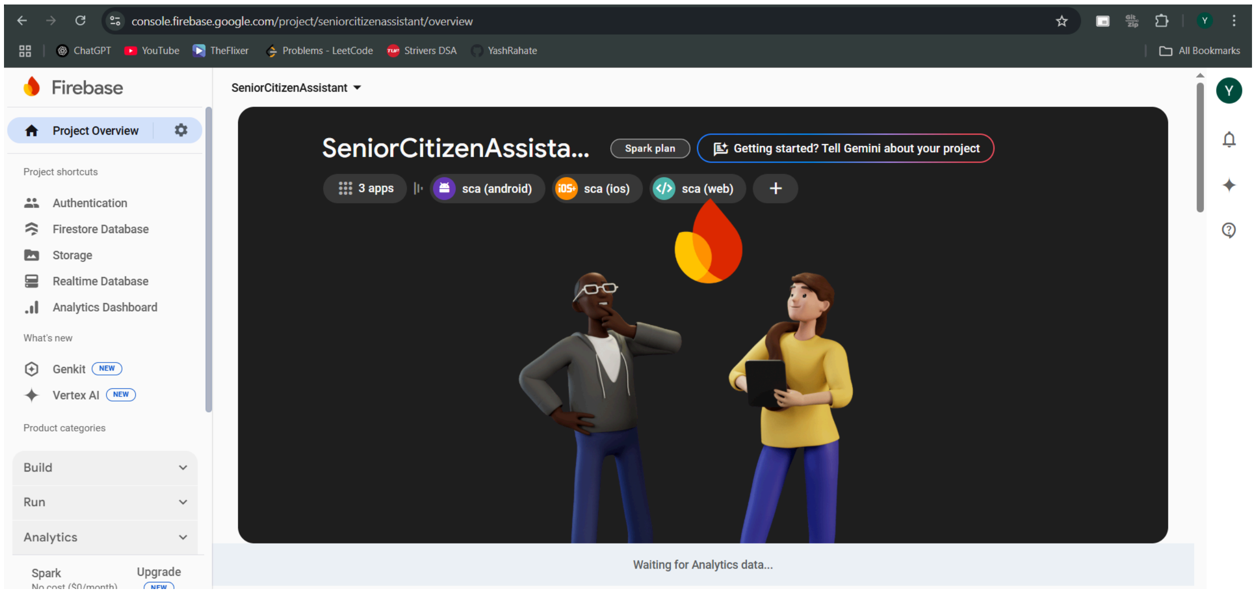


23:05 34%

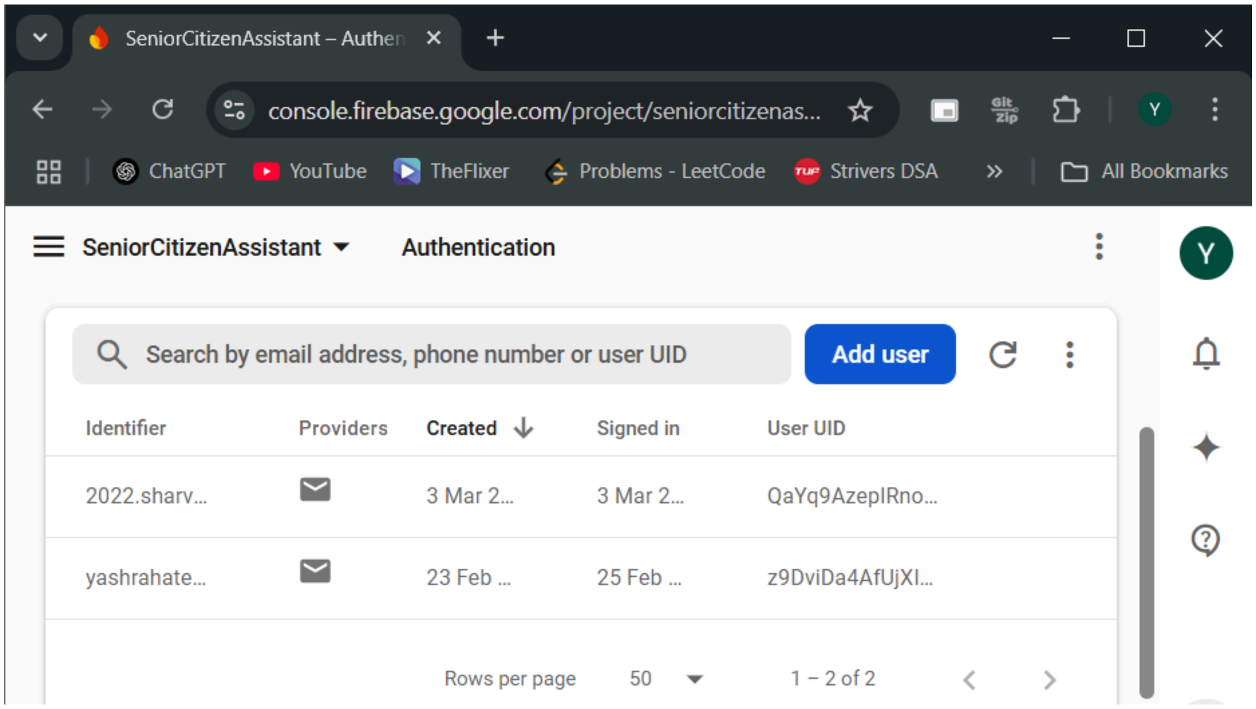
seniorcitizenassistant....
seniorcitizenassistant.firebaseapp.com

Your email has been verified

You can now sign in with your new account

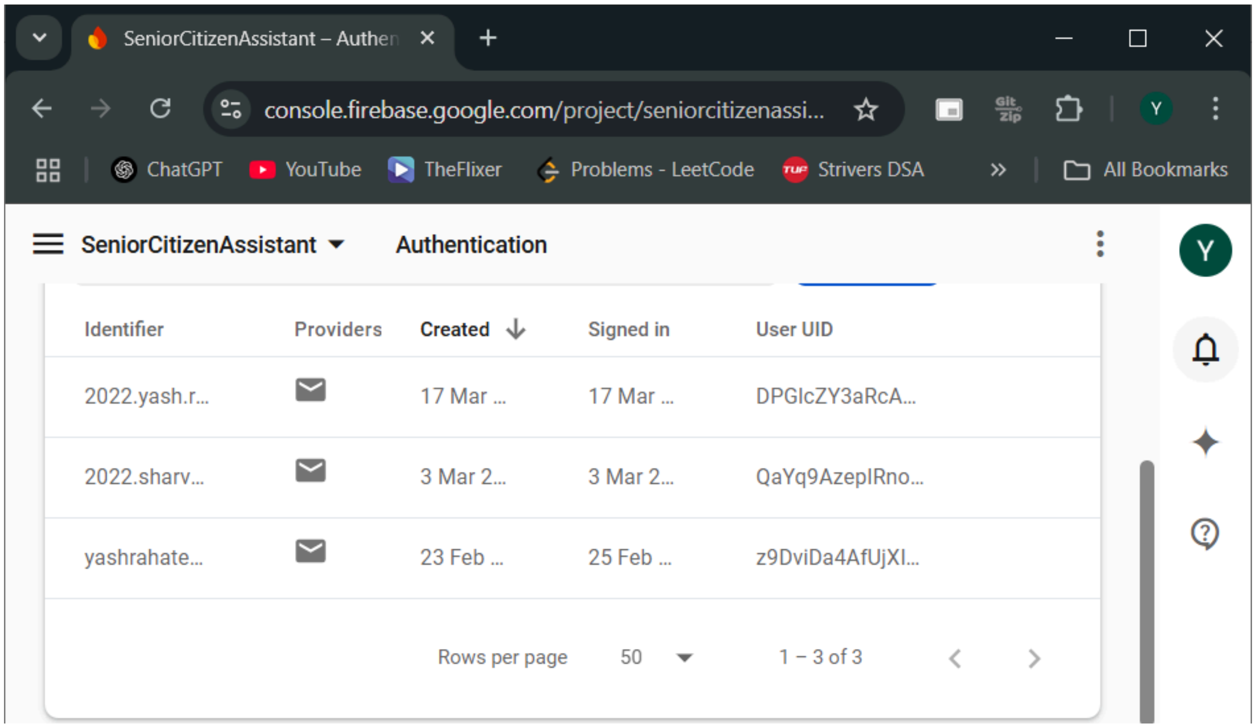


console.firebase.google.com/project/seniorcitizenassistant/overview


SeniorCitizenAssistant – Authen
console.firebase.google.com/project/seniorcitizenas...
ChatGPT
YouTube
TheFlixer
Problems - LeetCode
Strivers DSA
All Bookmarks
SeniorCitizenAssistant
Authentication
Search by email address, phone number or user UID
Add user

Identifier	Providers	Created	Signed in	User UID
2022.sharv...	✉	3 Mar 2...	3 Mar 2...	QaYq9AzepIRno...
yashrahate...	✉	23 Feb ...	25 Feb ...	z9DviDa4AfUjXI...

Rows per page 50 1 – 2 of 2



Identifier	Providers	Created	Signed in	User UID
2022.yash.r...	✉	17 Mar ...	17 Mar ...	DPGIcZY3aRcA...
2022.sharv...	✉	3 Mar 2...	3 Mar 2...	QaYq9AzepIRno...
yashrahate...	✉	23 Feb ...	25 Feb ...	z9DviDa4AfUjXI...

Rows per page 50 1 – 3 of 3

Conclusion: We successfully integrated Firebase Authentication with Flutter, implementing signup, login, email verification, password reset, and Firestore data storage. During development, we faced issues such as username not fetching from Firestore and Firestore offline errors, which we resolved by forcing online data retrieval, handling cache properly, and ensuring Firestore writes before navigation.