EXPERIMENT NO: - 06

Name:- Prajyot Shinde **Class:-** D15A **Roll:No: -** 55

AIM: - To connect Flutter UI with Firebase database.

Theory: -

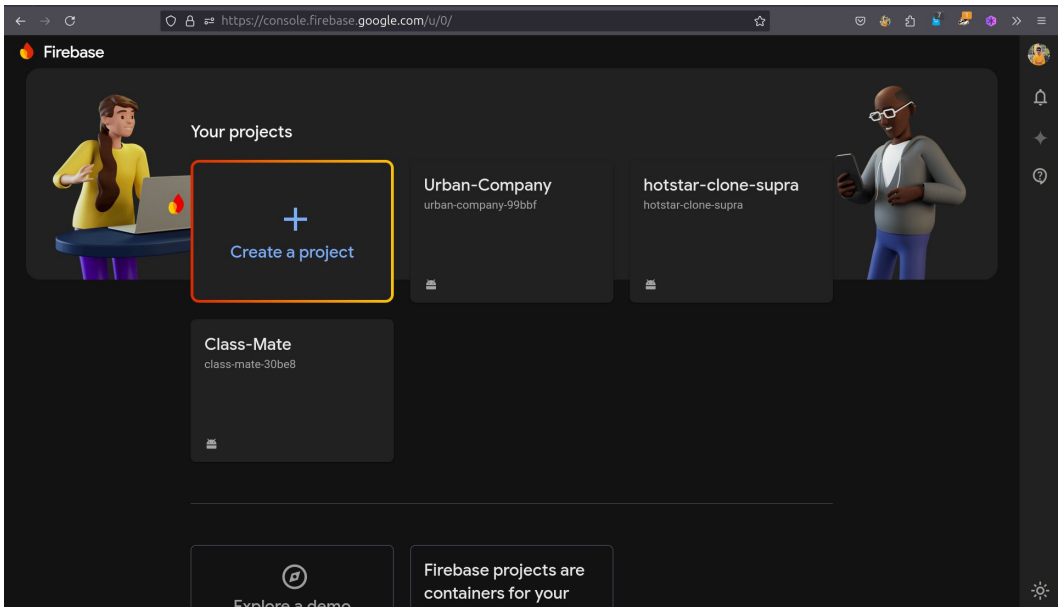
Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. Firebase, a Backend-as-a-Service (BaaS) platform, provides real-time database, authentication, and cloud storage services, making it a powerful backend solution for Flutter applications.

By integrating Firebase with Flutter, developers can store and retrieve data in real time, authenticate users, and manage cloud-based data efficiently. This is particularly useful for applications requiring dynamic content updates and user interactions.

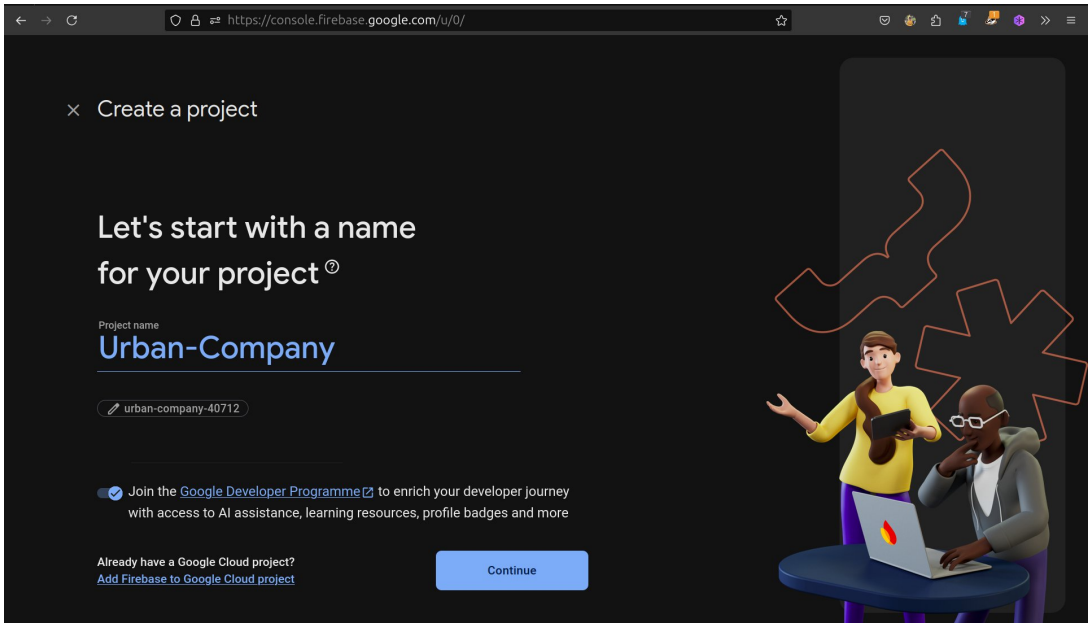
➢ Steps to Connect Flutter UI with Firebase Database

Step 1:

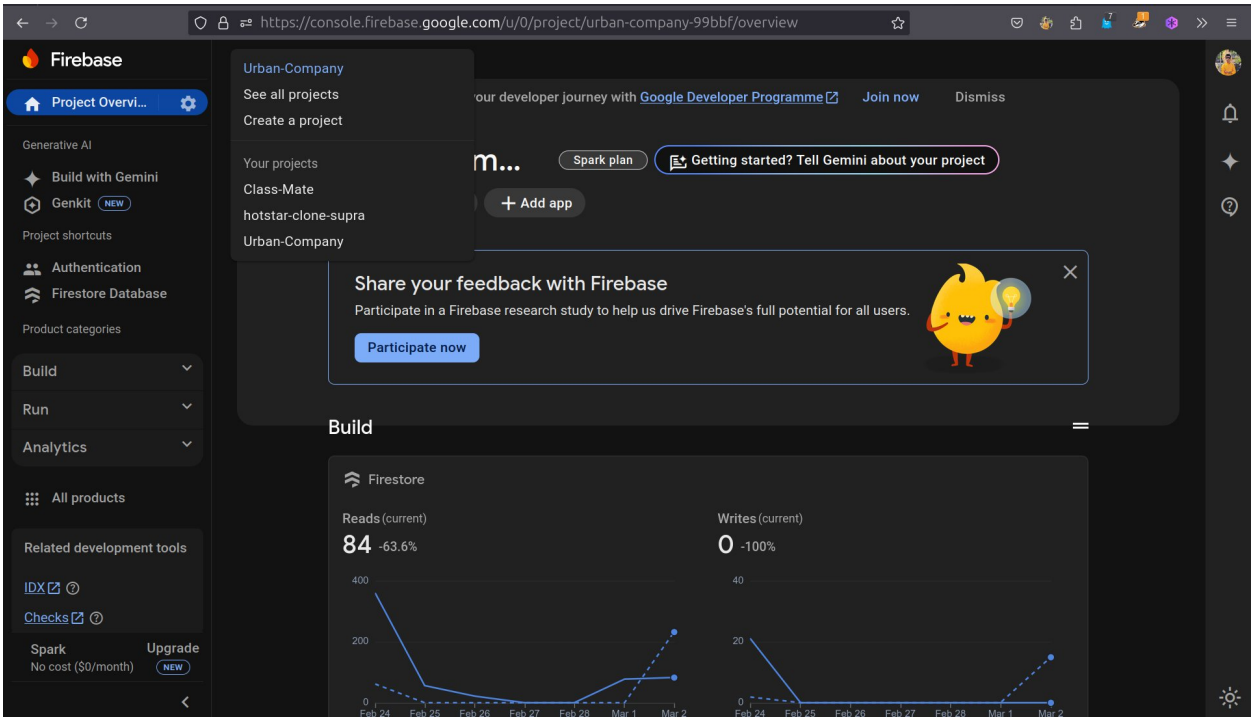
1.1) Go to Firebase Console and Create a Firebase Project



1.2) Click on Create a Project and give it a suitable name.

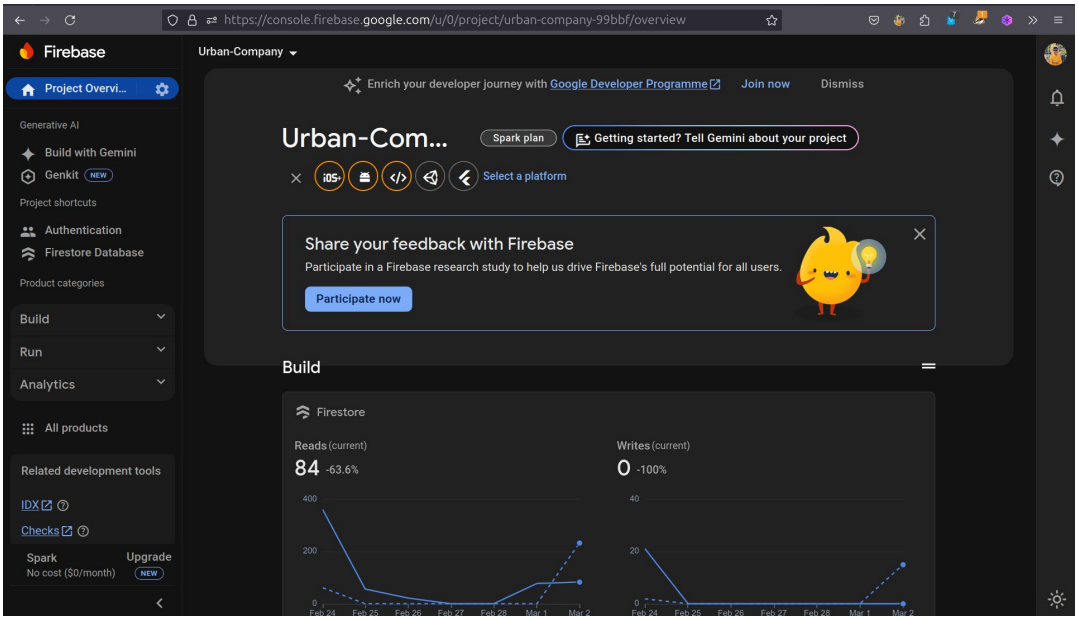


1.3) Enable Google Analytics (optional) & Click continue and complete the setup

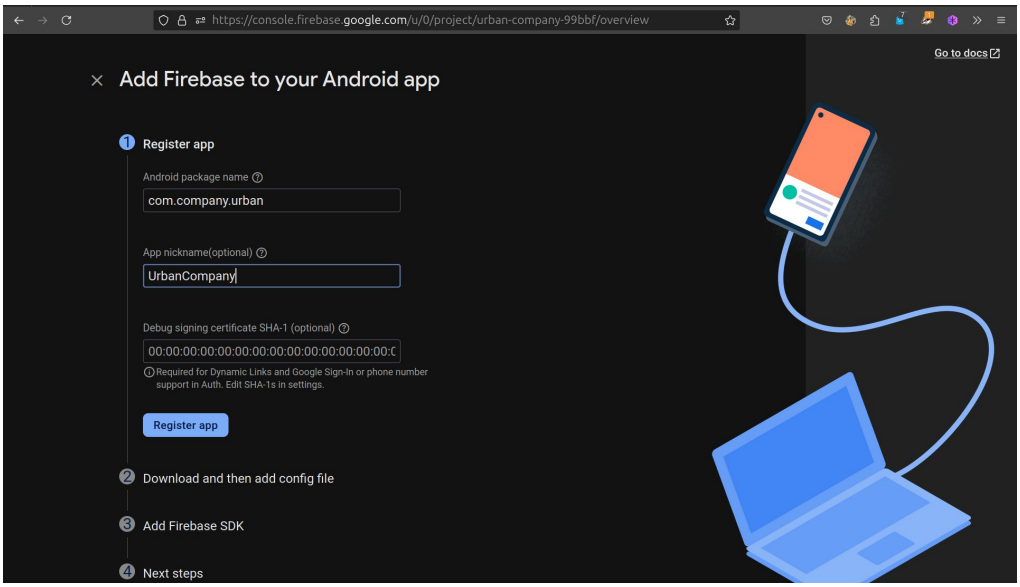


Step 2:- Add Firebase to Your Flutter App

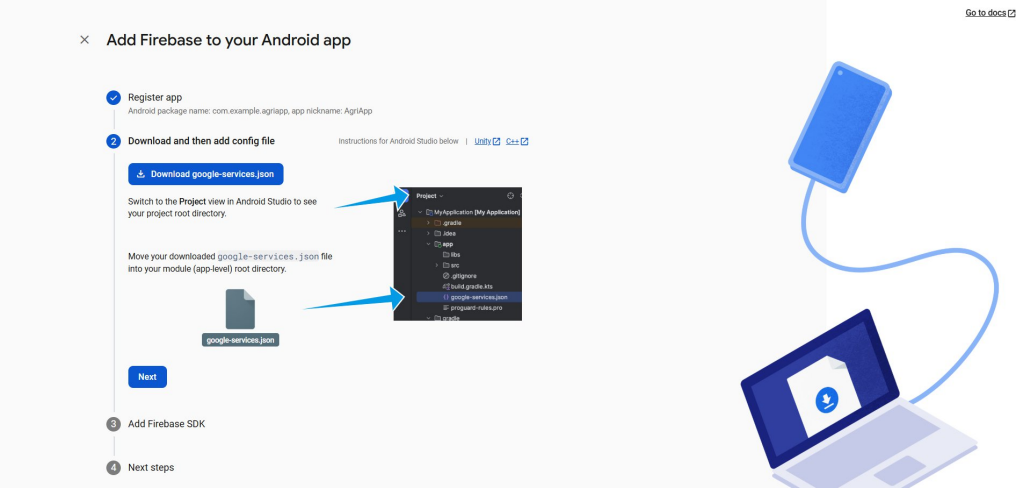
2.1) Click on Android/iOS/Web based on your Flutter application

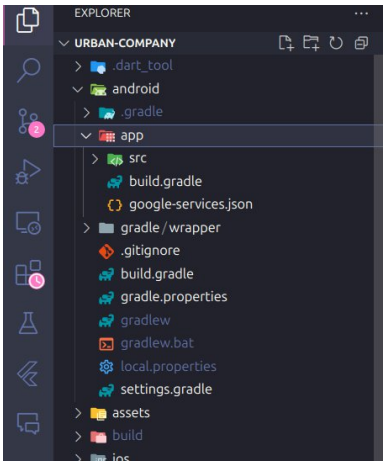


2.2) Register your app with a unique package name (found in android/app/build.gradle for Android).

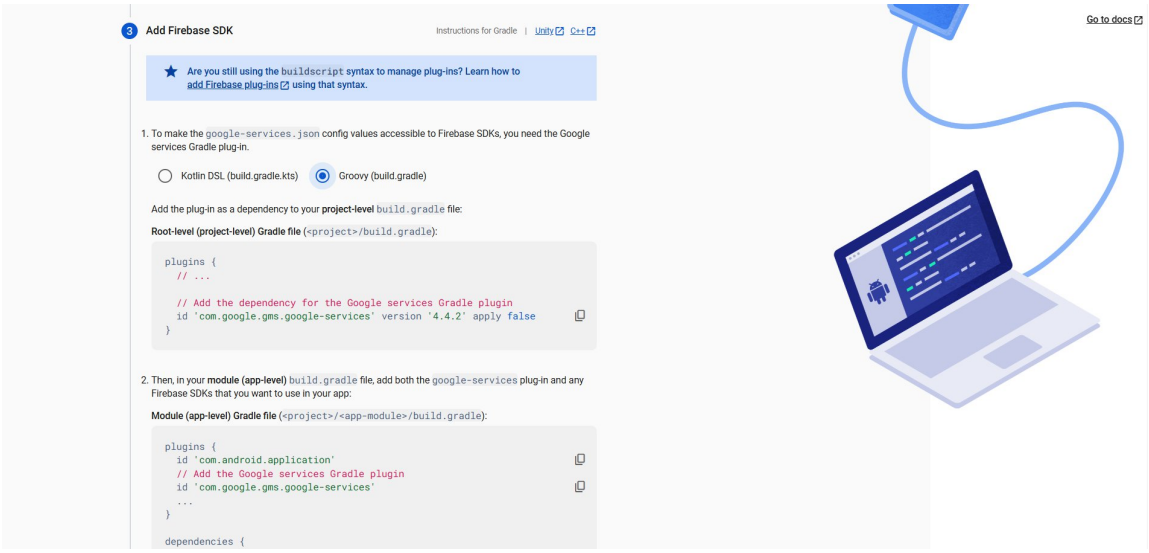


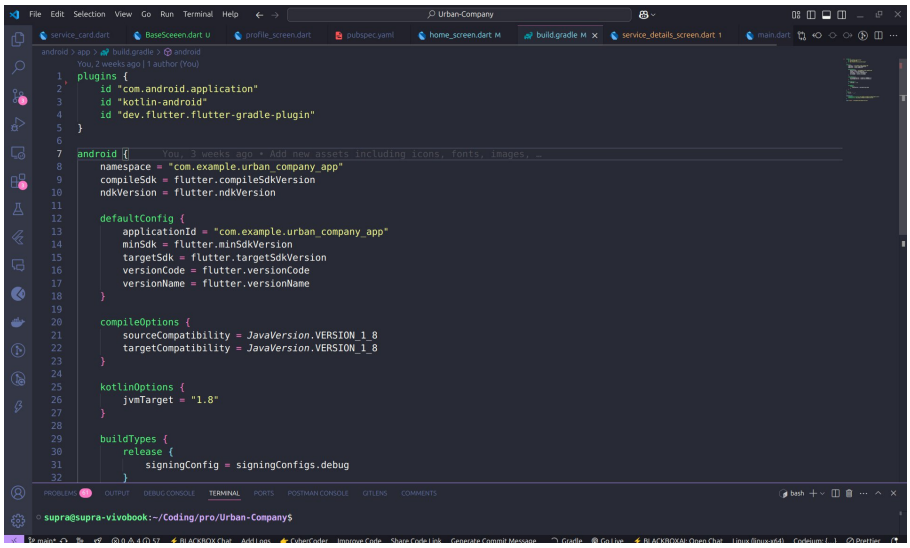
2.3) Download the google-services.json (for Android) & place the JSON file inside android/app/ directory.





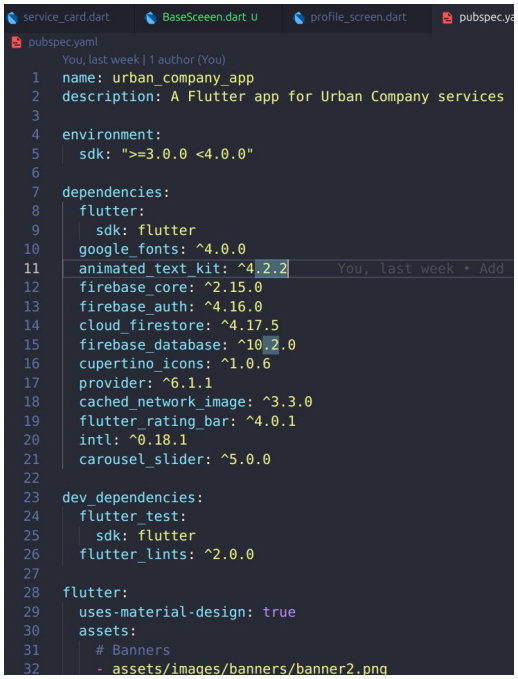
2.4) Add Firebase SDK dependencies to android/build.gradle



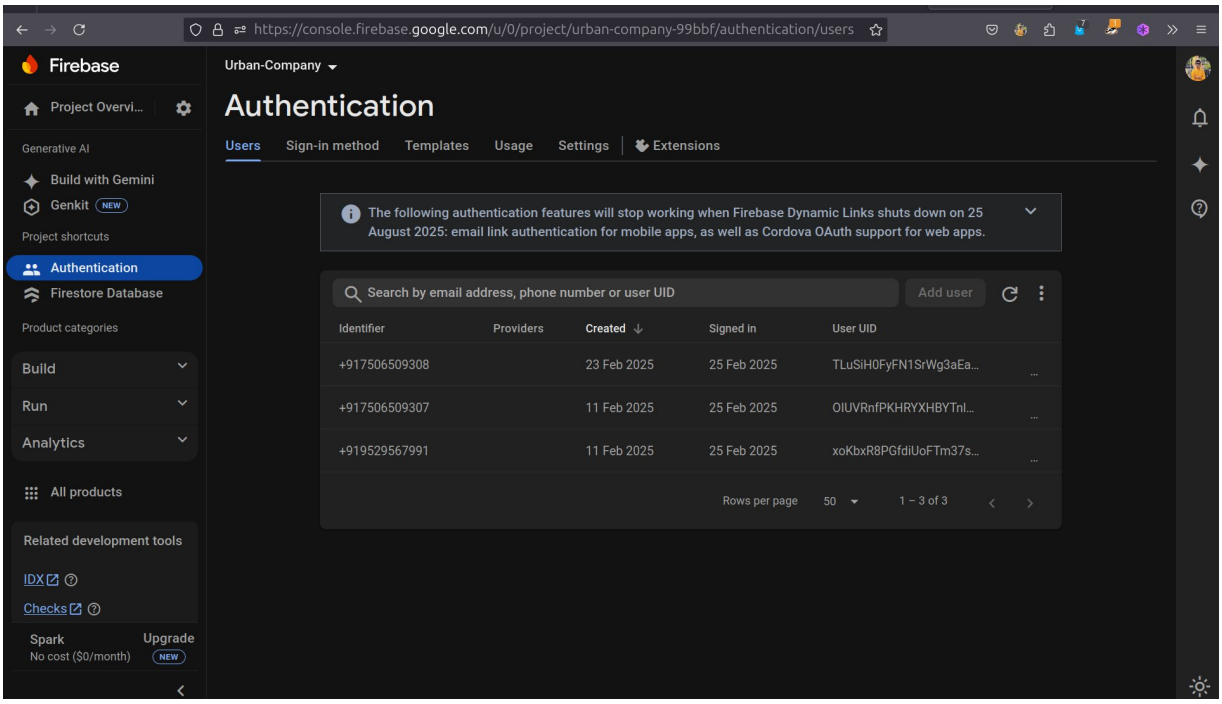


Step 3: - Add Firebase Authentication to Your App

3.1) Add Firebase Authentication Dependencies



3.2) Enable Authentication in Firebase Console
Go to **Firebase Console → Authentication**.
Click on **Sign-in method** and enable **Email/Password** (or any other method like Google).
Click Save



3.3) Implement Authentication in Flutter
Modify main.dart

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

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}
```

Step 4: -Configure Firebase Realtime Database

4.1) Go to Firebase Console → Realtime Database.
4.2) Click **Create Database** → Choose location → Set rules (for development, set read/write to true).
4.3) Click **Publish**.

Code:-

Signup_page.dart

```
import 'package:flutter/material.dart';
import 'auth_service.dart';
import 'package:firebase_auth/firebase_auth.dart';

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

  @override
  State<SignupScreen> createState() =>
_SignupScreenState();
}

class _SignupScreenState extends
State<SignupScreen> {
  final _formKey = GlobalKey<FormState>();
  final _nameController = TextEditingController();
  final _emailController = TextEditingController();
  final _addressController = TextEditingController();
  final _authService = AuthService();
  bool _isLoading = false;

  Future<void> _signup() async {
  if (_formKey.currentState!.validate()) {
    setState(() => _isLoading = true);
    try {
      await _authService.signupUser(
        name: _nameController.text,
        email: _emailController.text,
        address: _addressController.text,
      );

      Navigator.pushReplacementNamed(context,
'/home');
    } catch (e) {
      print("Error in signup: $e");
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text('Error signing up: $e')),
      );
    } finally {
      setState(() => _isLoading = false);
    }
  }
}

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Create Your Account'),
        centerTitle: true,
        elevation: 0,
      ),
```

```
      body: SafeArea(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: SingleChildScrollView(
            child: Form(
              key: _formKey,
              child: Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
                children: [
                  const SizedBox(height: 24),
                  Text(
                    'Please provide the following details:',
                    style: TextStyle(
                      fontSize: 16,
                      color: Colors.grey[700],
                    ),
                  ),
                  const SizedBox(height: 24),
                  _buildTextField(
                    controller: _nameController,
                    labelText: 'Full Name',
                    validator: (value) {
                      if (value == null || value.isEmpty) {
                        return 'Please enter your name';
                      }
                      return null;
                    },
                  ),
                  const SizedBox(height: 16),
                  _buildTextField(
                    controller: _emailController,
                    labelText: 'Email Address',
                    validator: (value) {
                      if (value == null || value.isEmpty) {
                        return 'Please enter your email';
                      }
                      if (!value.contains('@')) {
                        return 'Please enter a valid email';
                      }
                      return null;
                    },
                  ),
                  const SizedBox(height: 16),
                  _buildTextField(
                    controller: _addressController,
                    labelText: 'Address',
                    validator: (value) {
                      if (value == null || value.isEmpty) {
                        return 'Please enter your address';
                      }
                      return null;
                    },
                  ),
                  const SizedBox(height: 24),
                  SizedBox(
                    width: double.infinity,
                    child: ElevatedButton(
                      onPressed: _isLoading ? null : _signup,
                      style: ElevatedButton.styleFrom(
                        backgroundColor: Colors.deepPurple,
                        padding: const
EdgeInsets.symmetric(vertical: 16),
                        shape: RoundedRectangleBorder(
```

```
                    borderRadius:
BorderRadius.circular(8),
                  ),
                ),
                child: _isLoading
                    ? const CircularProgressIndicator(
                        color: Colors.white,
                      )
                    : const Text(
                        'Sign Up',
                        style: TextStyle(
                          fontSize: 18,
                          fontWeight: FontWeight.bold,
                        ),
                      ),
              ),
            ),
          ],
        ),
      ),
    ),
  );
}

// Custom TextField widget for cleaner code
Widget _buildTextField({
  required TextEditingController controller,
  required String labelText,
  required String? Function(String?) validator,
}) {
  return TextFormField(
    controller: controller,
    decoration: InputDecoration(
      labelText: labelText,
      border: OutlineInputBorder(
        borderRadius: BorderRadius.circular(8),
      ),
      contentPadding: const
EdgeInsets.symmetric(vertical: 16, horizontal: 12),
      filled: true,
      fillColor: Colors.grey[100],
    ),
    validator: validator,
  );
}
}
```

login_page.dart

```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import
'package:animated_text_kit/animated_text_kit.dart';
import 'auth_service.dart';

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

  @override
  State<LoginScreen> createState() =>
_LoginScreenState();
}
```

```
class _LoginScreenState extends State<LoginScreen>
{
 final _formKey = GlobalKey<FormState>();
 final _phoneController = TextEditingController();
 final _authService = AuthService();
 bool _isLoading = false;

 @override
 void dispose() {
  _phoneController.dispose();
  super.dispose();
 }

 Future<void> _sendOtp() async {
  if (_formKey.currentState!.validate()) {
   setState(() => _isLoading = true);
   final phoneNumber = _phoneController.text;

   await _authService.verifyPhone(
    phoneNumber: phoneNumber,
    onCodeSent: (verificationId) {
     setState(() => _isLoading = false);
     Navigator.pushNamed(
      context,
      '/otp',
      arguments: {
       'phoneNumber': phoneNumber,
       'verificationId': verificationId,
      },
     );
    },
    onError: (error) {
     setState(() => _isLoading = false);
     ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text(error)),
     );
    },
   );
  }
 }

 @override
 Widget build(BuildContext context) {
  return Scaffold(
   body: SafeArea(
    child: Padding(
     padding: const EdgeInsets.all(16.0),
     child: Form(
      key: _formKey,
      child: Column(
       crossAxisAlignment:
CrossAxisAlignment.stretch,
       children: [
        const SizedBox(height: 48),
        Center(
         child: AnimatedTextKit(
          animatedTexts: [
           TyperAnimatedText(
            'Urban Company',
            textStyle: GoogleFonts.poppins(
             fontSize: 36.0,
             fontWeight: FontWeight.bold,
             color: Colors.deepPurple,
```

```
            ),
            speed: const Duration(milliseconds:
100),
          ),
        ],
        repeatForever: true,
        pause: const Duration(seconds: 2),
      ),
    ),
    const SizedBox(height: 24),
    const Text(
      'Welcome Back',
      style: TextStyle(
        fontSize: 32,
        fontWeight: FontWeight.bold,
      ),
      textAlign: TextAlign.center,
    ),
    const SizedBox(height: 8),
    const Text(
      'Enter your mobile number to continue',
      style: TextStyle(
        fontSize: 16,
        color: Colors.black26,
      ),
      textAlign: TextAlign.center,
    ),
    const SizedBox(height: 32),
    TextFormField(
      controller: _phoneController,
      keyboardType: TextInputType.phone,
      decoration: InputDecoration(
        prefixText: '+91 ',
        hintText: 'Phone number',
        border: OutlineInputBorder(
          borderRadius:
BorderRadius.circular(8),
        ),
      ),
      validator: (value) {
        final trimmedValue = value?.trim() ?? '';
        if (trimmedValue.isEmpty) {
          return 'Please enter your phone
number';
        }
        if (trimmedValue.length != 10) {
          return 'Phone number must be 10
digits';
        }
        if
(!RegExp(r'^\d{10}$').hasMatch(trimmedValue)) {
          return 'Please enter a valid phone
number';
        }
        return null;
      },
    ),
    const SizedBox(height: 24),
    Container(
      decoration: BoxDecoration(
        gradient: const LinearGradient(
          colors: [Colors.deepPurple,
Colors.purpleAccent],
        ),
        borderRadius: BorderRadius.circular(8),
      ),
```

```
              child: ElevatedButton(
                onPressed: _isLoading ? null : _sendOtp,
                style: ElevatedButton.styleFrom(
                  backgroundColor: Colors.transparent,
                  foregroundColor: Colors.white,
                  minimumSize: const
Size(double.infinity, 50),
                  elevation: 0,
                ),
                child: _isLoading
                    ? const SizedBox(
                        height: 20,
                        width: 20,
                        child: CircularProgressIndicator(
                          strokeWidth: 2,
                          color: Colors.white,
                        ),
                      )
                    : const Text('Continue',
                        style: TextStyle(fontSize: 18)),
              ),
            ),
          ],
        ),
      ),
    ),
  );
 }
}
```

Profile_screen.dart

```
import 'package:flutter/material.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import '../auth/auth_service.dart';
import 'package:intl/intl.dart';


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


 @override
 State<ProfileScreen> createState() =>
_ProfileScreenState();
}


class _ProfileScreenState extends
State<ProfileScreen> {
 final _formKey = GlobalKey<FormState>();
 final _nameController = TextEditingController();
 final _emailController = TextEditingController();
```

```
  final _addressController = TextEditingController();
  final _phoneController = TextEditingController();
  bool _isLoading = true;

  final _authService = AuthService();

  @override
  void initState() {
   super.initState();
   _loadProfile();
  }

  String _formatDate(String timestamp) {
   try {
    final date = DateTime.parse(timestamp);
    return DateFormat('yyyy-MM-dd')
      .format(date); // Formats date as YYYY-MM-
DD
   } catch (e) {
    return 'Invalid date';
   }
  }

  Future<void> _loadProfile() async {
   final user = FirebaseAuth.instance.currentUser;
   if (user == null) {
    setState(() {
     _isLoading = false;
    });
    // Redirect to login screen if the user is not
authenticated
    Navigator.pushReplacementNamed(context,
'/login');
    return;
   }

   try {
    // Normalize phone number
    String userPhone = user.phoneNumber ?? '';
    String normalizedPhone =
_normalizePhoneNumber(userPhone);

    // Fetch profile data
    final profileData =
      await
_authService.getUserProfileByPhoneNumber(normal
izedPhone);

    if (profileData != null) {
     setState(() {
      _nameController.text = profileData['name'] ??
'';
      _emailController.text = profileData['email'] ??
'';
      _addressController.text =
profileData['address'] ?? '';
      _phoneController.text = profileData['phone'] ??
'';
      _isLoading = false;
     });
    } else {
     ScaffoldMessenger.of(context).showSnackBar(
      const SnackBar(content: Text('No profile
found')),
     );
     setState(() {
```

```
        _isLoading = false;
      });
    }
  } catch (e) {
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text('Error loading profile: $e')),
    );
    setState(() {
      _isLoading = false;
    });
  }
}

void _cancelSubscription(BuildContext context, String subscriptionId) async {
  await FirebaseFirestore.instance
      .collection('user_subscriptions')
      .doc(subscriptionId)
      .update({'status': 'cancelled'});

  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content: Text('Subscription cancelled.')),
  );
  setState(() {}); // Refresh the UI
}

String _normalizePhoneNumber(String phoneNumber) {
  String normalizedPhone = phoneNumber.replaceAll(RegExp(r'[^0-9]'), '');
  if (normalizedPhone.length > 10) {
    normalizedPhone = normalizedPhone.substring(normalizedPhone.length - 10);
  }
  return normalizedPhone;
}

@override
Widget build(BuildContext context) {
  if (_isLoading) {
    return const Scaffold(
      body: Center(child: CircularProgressIndicator()),
    );
  }

  return Scaffold(
    appBar: AppBar(
      title: const Text('My Profile'),
      actions: [
        IconButton(
          icon: const Icon(Icons.logout),
          onPressed: _logout,
        ),
      ],
    ),
    body: SingleChildScrollView(
      padding: const EdgeInsets.all(16),
      child: Column(
        children: [
          Card(
            elevation: 4,
            shape: RoundedRectangleBorder(
```

```
            borderRadius: BorderRadius.circular(12),
          ),
          child: Padding(
            padding: const EdgeInsets.all(16),
            child: Form(
              key: _formKey,
              child: Column(
                crossAxisAlignment:
CrossAxisAlignment.start,
                children: [
                  const Center(
                    child: CircleAvatar(
                      radius: 50,
                      backgroundColor:
Colors.deepPurple,
                      child: Icon(
                        Icons.person,
                        size: 50,
                        color: Colors.white,
                      ),
                    ),
                  ),
                  const SizedBox(height: 24),
                  Text(
                    'Phone:
${FirebaseAuth.instance.currentUser?.phoneNumber
?? _phoneController.text}',
                    style: const TextStyle(
                      fontSize: 16,
                      fontWeight: FontWeight.bold,
                    ),
                    textAlign: TextAlign.center,
                  ),
                  const SizedBox(height: 24),
                  TextFormField(
                    controller: _nameController,
                    decoration: const InputDecoration(
                      labelText: 'Full Name',
                      border: OutlineInputBorder(),
                      prefixIcon: Icon(Icons.person),
                    ),
                    validator: (value) {
                      if (value == null || value.isEmpty) {
                        return 'Please enter your name';
                      }
                      return null;
                    },
                  ),
                  const SizedBox(height: 16),
                  TextFormField(
                    controller: _emailController,
                    decoration: const InputDecoration(
                      labelText: 'Email',
                      border: OutlineInputBorder(),
                      prefixIcon: Icon(Icons.email),
                    ),
                    validator: (value) {
                      if (value == null || value.isEmpty) {
                        return 'Please enter your email';
                      }
                      if (!value.contains('@')) {
                        return 'Please enter a valid email';
                      }
                      return null;
                    },
```

```
                ),
                const SizedBox(height: 16),
                TextFormField(
                  controller: _addressController,
                  decoration: const InputDecoration(
                    labelText: 'Address',
                    border: OutlineInputBorder(),
                    prefixIcon: Icon(Icons.home),
                  ),
                  maxLines: 3,
                  validator: (value) {
                    if (value == null || value.isEmpty) {
                      return 'Please enter your address';
                    }
                    return null;
                  },
                ),
                const SizedBox(height: 24),
                SizedBox(
                  width: double.infinity,
                  child: ElevatedButton(
                    onPressed: _isLoading ? null :
_saveProfile,
                    style: ElevatedButton.styleFrom(
                      backgroundColor:
Colors.deepPurple,
                      padding: const
EdgeInsets.symmetric(vertical: 16),
                      shape: RoundedRectangleBorder(
                        borderRadius:
BorderRadius.circular(8),
                      ),
                    ),
                    child: _isLoading
                        ? const CircularProgressIndicator(
                            color: Colors.white,
                          )
                        : const Text(
                            'Save Profile',
                            style: TextStyle(
                              fontSize: 18,
                              fontWeight: FontWeight.bold,
                            ),
                          ),
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
      const SizedBox(height: 24),
      ExpansionTile(
        title: const Text(
          'My Subscriptions',
          style: TextStyle(
            fontSize: 18,
            fontWeight: FontWeight.bold,
          ),
        ),
        initiallyExpanded: true,
        children: [
          FutureBuilder<QuerySnapshot>(
            future: FirebaseFirestore.instance
                .collection('user_subscriptions')
```

```
                .where('userId',
                    isEqualTo:
FirebaseAuth.instance.currentUser!.uid)
                .get(),
              builder: (context, snapshot) {
               if (snapshot.connectionState ==
ConnectionState.waiting) {
                return const Center(child:
CircularProgressIndicator());
               } else if (snapshot.hasError) {
                return Center(child: Text('Error:
${snapshot.error}'));
               } else if (!snapshot.hasData ||
                  snapshot.data!.docs.isEmpty) {
                return const Center(
                  child: Text('No active
subscriptions.'));
               } else {
                final subscriptions =
snapshot.data!.docs;
                return ListView.builder(
                  shrinkWrap: true,
                  physics: const
NeverScrollableScrollPhysics(),
                  itemCount: subscriptions.length,
                  itemBuilder: (context, index) {
                   final subscription =
subscriptions[index].data()
                      as Map<String, dynamic>;
                   return Card(
                    elevation: 4,
                    margin: const
EdgeInsets.symmetric(vertical: 8),
                    shape: RoundedRectangleBorder(
                     borderRadius:
BorderRadius.circular(12),
                    ),
                    child: Padding(
                     padding: const EdgeInsets.all(16),
                     child: Column(
                      crossAxisAlignment:
CrossAxisAlignment.start,
                      children: [
                       Row(
                        children: [
                         const Icon(Icons.assignment,
                           color: Colors.deepPurple),
                         const SizedBox(width: 8),
                         Expanded(
                          child: Text(

subscription['serviceDetails']
                              ['title'],
                           style: const TextStyle(
                            fontSize: 18,
                            fontWeight:
FontWeight.bold,
                           ),
                           overflow:
TextOverflow.ellipsis,
                          ),
                         ),
                        ],
                       ),
                       const SizedBox(height: 8),
                       Row(
                        children: [
                         const
Icon(Icons.calendar_today,
```

```
                    size: 16, color:
Colors.blue),
                  const SizedBox(width: 8),
                  Text(
                   'Next Delivery:
${_formatDate(subscription['nextDelivery'])}',
                    style: const TextStyle(
                     fontSize: 14,
                     color: Colors.blue,
                    ),
                   ),
                  ],
                 ),
                 const SizedBox(height: 8),
                 Row(
                  children: [
                   const Icon(Icons.star,
                     size: 16, color:
Colors.orange),
                   const SizedBox(width: 8),
                   Text(
                    'Status:
${subscription['status']}',
                    style: TextStyle(
                     fontSize: 14,
                     color:
                       subscription['status'] ==
'active'
                         ? Colors.green
                         : Colors.red,
                    ),
                   ),
                  ],
                 ),
                 const SizedBox(height: 16),
                 Align(
                  alignment:
Alignment.centerRight,
                  child: ElevatedButton(
                   onPressed: () {
                    _cancelSubscription(
                      context,
subscriptions[index].id);
                   },
                   style:
ElevatedButton.styleFrom(
                     backgroundColor:
Colors.red,
                     padding: const
EdgeInsets.symmetric(
                       horizontal: 16, vertical:
8),
                     shape:
RoundedRectangleBorder(
                      borderRadius:

BorderRadius.circular(8),
                     ),
                    ),
                   child: const Text(
                    'Cancel',
                    style: TextStyle(
                     fontSize: 14, color:
Colors.white),
                    ),
                   ),
                  ),
                 ],
                ),
```

```
                    ),
                  );
                },
              );
            }
          },
        ),
      ],
    ),
  ],
 ),
),
);
}

Future<void> _saveProfile() async {
 if (_formKey.currentState!.validate()) {
  setState(() {
   _isLoading = true;
  });

  try {
   final user = FirebaseAuth.instance.currentUser;
   if (user == null) {
    throw Exception('User not logged in');
   }

   final phoneNumber = user.phoneNumber ?? '';
   await _authService.updateUserProfile(
    phone: phoneNumber,
    name: _nameController.text,
    email: _emailController.text,
    address: _addressController.text,
   );

   if (mounted) {
    ScaffoldMessenger.of(context).showSnackBar(
     const SnackBar(content: Text('Profile updated successfully')),
    );
   }
  } catch (e) {
   if (mounted) {
    ScaffoldMessenger.of(context).showSnackBar(
     SnackBar(content: Text('Error updating profile: $e')),
    );
   }
  } finally {
   if (mounted) {
    setState(() {
     _isLoading = false;
    });
   }
  }
 }

 Future<void> _logout() async {
  try {
   await FirebaseAuth.instance.signOut();
   if (mounted) {
    Navigator.pushReplacementNamed(context, '/login');
```

```
    }
   } catch (e) {
    if (mounted) {
     ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(content: Text('Error signing out: $e')),
     );
    }
   }
  }
}
```

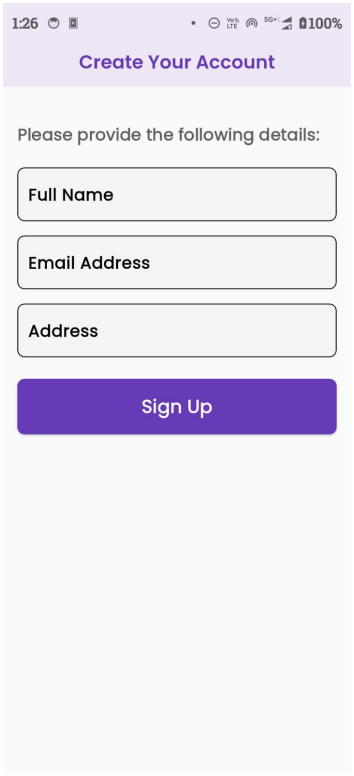
Main.dart

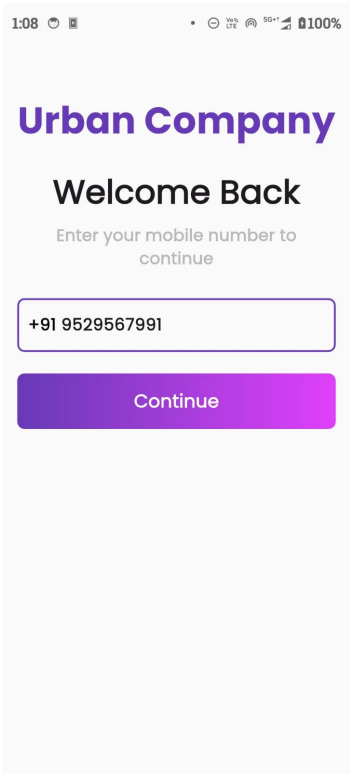
```
import 'package:flutter/material.dart';
import 'package:google_fonts/google_fonts.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import '/screens/home_screen.dart';
import '/screens/auth/login_screen.dart';
import '/screens/bookings/bookings_screen.dart';
import '/screens/profile/profile_screen.dart';
import '/screens/service_details_screen.dart';
import '/screens/help/help_screen.dart';
import '/screens/auth/otp_verify.dart';
import '/screens/auth/signup_screen.dart';
import './screens/handyman_service_screen.dart';



void main() async {
 WidgetsFlutterBinding.ensureInitialized();
 await Firebase.initializeApp();
 runApp(const MyApp());
}


class MyApp extends StatelessWidget {
 const MyApp({super.key});


 @override
 Widget build(BuildContext context) {
  return MaterialApp(
   debugShowCheckedModeBanner: false,
   title: 'Urban Company',
   theme: ThemeData(
    colorScheme: ColorScheme.light(
     primary: Colors.deepPurple,
     secondary: Colors.orangeAccent,
    ),
    scaffoldBackgroundColor: Colors.grey[50],
    fontFamily: 'Poppins',
    textTheme: GoogleFonts.poppinsTextTheme(),
    appBarTheme: AppBarTheme(
     backgroundColor: Colors.deepPurple[50],
     elevation: 2,
     iconTheme: const IconThemeData(color: Colors.deepPurple),
     titleTextStyle: GoogleFonts.poppins(
      color: Colors.deepPurple,
      fontSize: 18,
      fontWeight: FontWeight.w600,
     ),
    ),
```

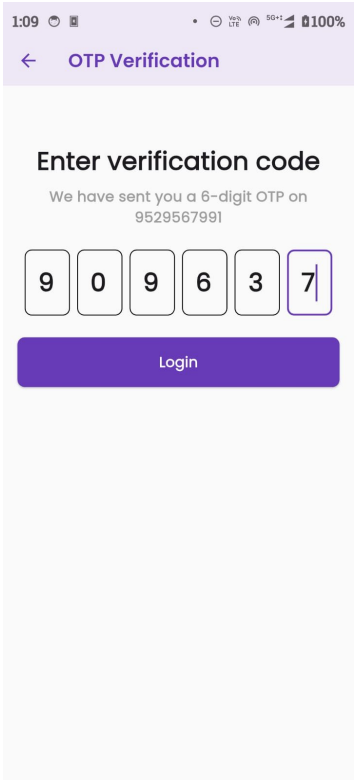
```
      elevatedButtonTheme:
ElevatedButtonThemeData(
        style: ElevatedButton.styleFrom(
          backgroundColor: Colors.deepPurple,
          foregroundColor: Colors.white,
          padding: const EdgeInsets.symmetric(vertical:
16, horizontal: 24),
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(8),
          ),
        ),
      ),
    ),
    home: AuthWrapper(),
    routes: {
      '/signup': (context) => const SignupScreen(),
      '/otp': (context) {
        final args =
ModalRoute.of(context)?.settings.arguments;

        if (args is Map<String, String>) {
          return OtpVerificationScreen(
            phoneNumber: args['phoneNumber'] ?? '',
            verificationId: args['verificationId'] ?? '',
          );
        } else {
          return const Scaffold(
            body: Center(child: Text('Invalid arguments
for OTP screen')),
          );
        }
      },
      '/login': (context) => const LoginScreen(),
      '/home': (context) => const HomeScreen(),
      '/bookings': (context) => const
BookingsScreen(),
      '/help': (context) => const HelpScreen(),
      '/profile': (context) => const ProfileScreen(),
      '/service-details': (context) =>
ServiceDetailsScreen(),
      '/handyman-services': (context) =>
HandymanServiceScreen(),
    },
   );
  }
}

class AuthWrapper extends StatelessWidget {
  const AuthWrapper({super.key});

  @override
  Widget build(BuildContext context) {
    return StreamBuilder<User?>(
      stream:
FirebaseAuth.instance.authStateChanges(),
      builder: (context, authSnapshot) {
        if (authSnapshot.connectionState ==
ConnectionState.waiting) {
          return const Scaffold(
            body: Center(child:
CircularProgressIndicator()),
          );
        } else if (authSnapshot.hasData) {
          return FutureBuilder<Map<String,
dynamic>?>(
            future:
_checkUserProfile(authSnapshot.data!),
            builder: (context, profileSnapshot) {
```

```
          if (profileSnapshot.connectionState ==
ConnectionState.waiting) {
            return const Scaffold(
              body: Center(child:
CircularProgressIndicator()),
            );
          } else if (profileSnapshot.hasError
|| !profileSnapshot.hasData) {

WidgetsBinding.instance.addPostFrameCallback((_)
{

Navigator.pushReplacementNamed(context, '/login');
            });
            return const Scaffold(
              body: Center(child: Text('Redirecting to
login...')),
            );
          } else {
            return const HomeScreen();
          }
        },
      );
    } else {
      return const LoginScreen();
    }
  },
 );
}
```

```
Future<Map<String, dynamic>?>
_checkUserProfile(User user) async {
  try {
   final doc = await FirebaseFirestore.instance
     .collection('users')
     .doc(user.uid)
     .get();
   if (doc.exists) {
    return doc.data();
   } else {
    return null;
   }
  } catch (e) {
   print("Error checking user profile: $e");
   return null;
  }
 }
}
```

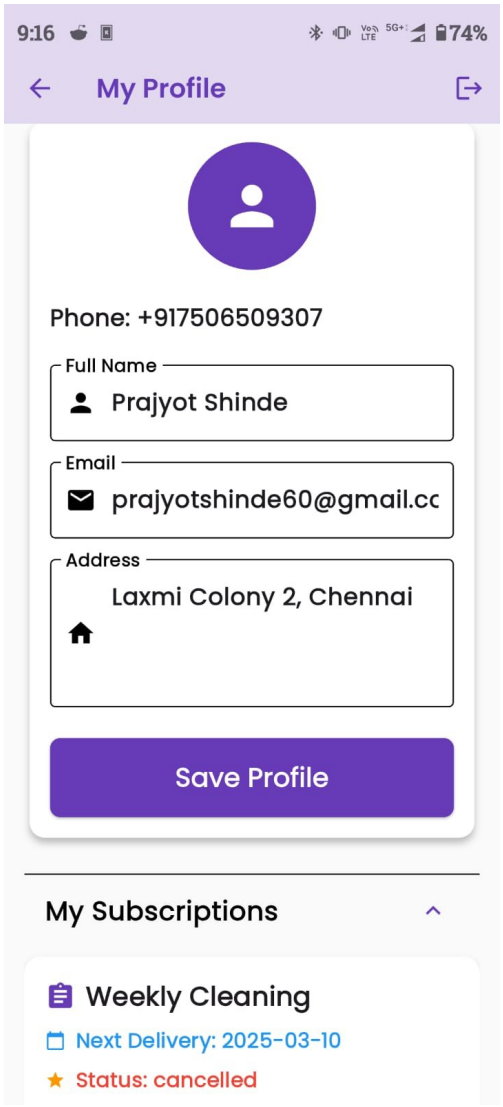

1:26
100%
Create Your Account
Please provide the following details:
Full Name
Email Address
Address
Sign Up

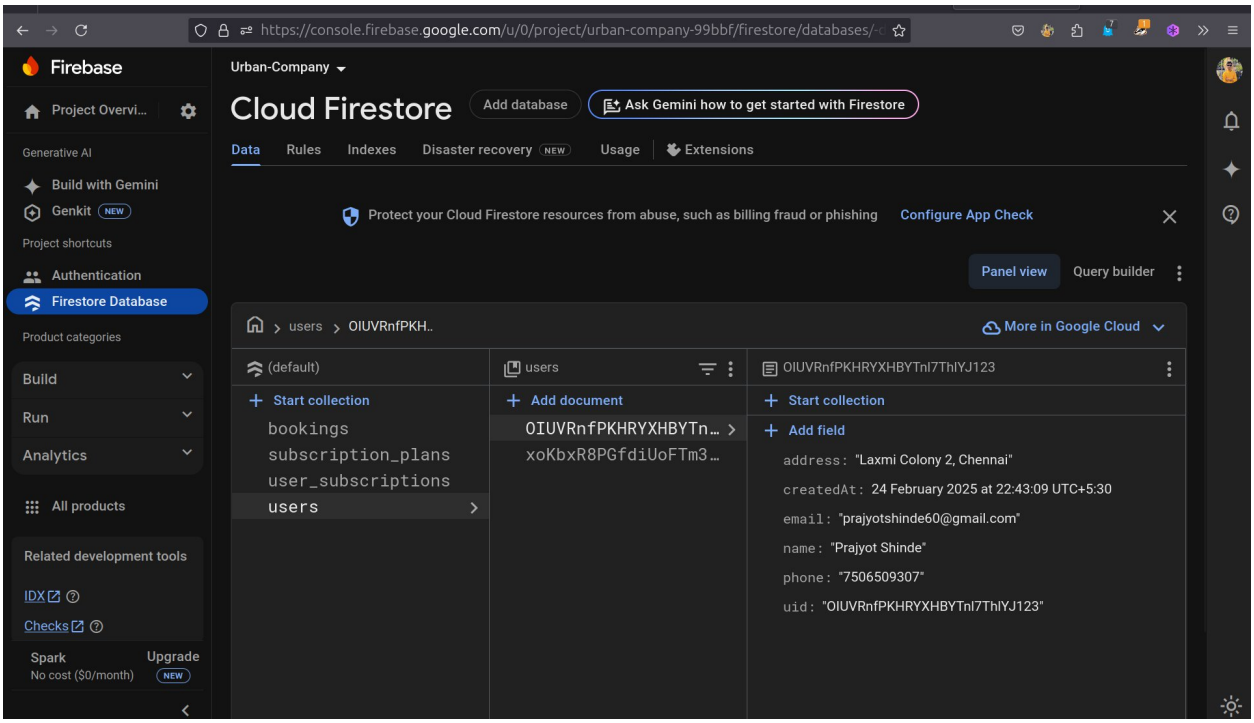

1:08
100%
Urban Company
Welcome Back
Enter your mobile number to continue
+91 9529567991
Continue


1:09
100%
OTP Verification
Enter verification code
We have sent you a 6-digit OTP on 9529567991
9 0 9 6 3 7
Login

After Registering, the user details is saved in the database.

User details get fetched from the database in the profile page.




https://console.firebase.google.com/u/0/project/urban-company-99bbf/firestore/databases/-
Firebase
Urban-Company
Project Overvi...
Generative AI
Build with Gemini
Genkit NEW
Project shortcuts
Authentication
Firestore Database
Product categories
Build
Run
Analytics
All products
Related development tools
IDX
Checks
Spark
No cost ($0/month)
Upgrade
NEW
Cloud Firestore
Add database
Ask Gemini how to get started with Firestore
Data
Rules
Indexes
Disaster recovery NEW
Usage
Extensions
Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing
Configure App Check
Panel view
Query builder
users > OIUVRnfPKH..
More in Google Cloud
(default)
Start collection
bookings
subscription_plans
user_subscriptions
users
users
Add document
OIUVRnfPKHRYXHBYTn...
xoKbxR8PGfdiUoFTm3...
OIUVRnfPKHRYXHBYTnI7ThIYJ123
Start collection
Add field
address: "Laxmi Colony 2, Chennai"
createdAt: 24 February 2025 at 22:43:09 UTC+5:30
email: "prajyotshinde60@gmail.com"
name: "Prajyot Shinde"
phone: "7506509307"
uid: "OIUVRnfPKHRYXHBYTnI7ThIYJ123"