

Experiment 6

Aim: To connect flutter UI with firebase database

Theory: **Firebase** provides a cloud-based backend service with features like:

- Realtime Database
- Authentication
- Firestore
- Cloud Storage

To connect Flutter to Firebase:

1. Create a Firebase project at console.firebase.google.com.
2. Add your app's package name and download google-services.json.
3. Add Firebase dependencies in pubspec.yaml.
4. Initialize Firebase in your app using `Firebase.initializeApp()`.

Use `firebase_database` package to interact with Realtime Database.

Code:

Edit Dependencies in pubspec.yaml:

```
firebase_auth: ^5.5.1
firebase_core: ^3.12.1
image_picker: ^1.1.2
intl: ^0.18.0
cloud_firestore: ^5.6.5
firebase_storage: ^12.4.4
file_picker: ^9.2.2
cross_file: ^0.3.4+2
path: ^1.9.0
provider: ^6.1.4
```

Firestore Setup Steps:

1. Go to Firebase Console
2. Create a project → Add Android app
3. Register with your package name (e.g., com.example.finderbase)
4. Download google-services.json and place it in android/app/
5. In android/build.gradle and android/app/build.gradle, enable Google services
6. Initialize Firebase in main.dart **Initialize firebase in main.dart** import

```
'package:flutter/material.dart'; import 'package:firebase_core/firebase_core.dart'; import  
'report_form.dart';
```

```
void main() async {
```

```
  WidgetsFlutterBinding.ensureInitialized();
```

```
  await Firebase.initializeApp();
```

```
  runApp(MealMonkeyApp());
```

```
}
```

```
class MealMonkeyApp extends StatelessWidget {
```

```
  @override
```

```
  Widget build(BuildContext context) { return
```

```
    MaterialApp( title: 'FinderBase', theme:
```

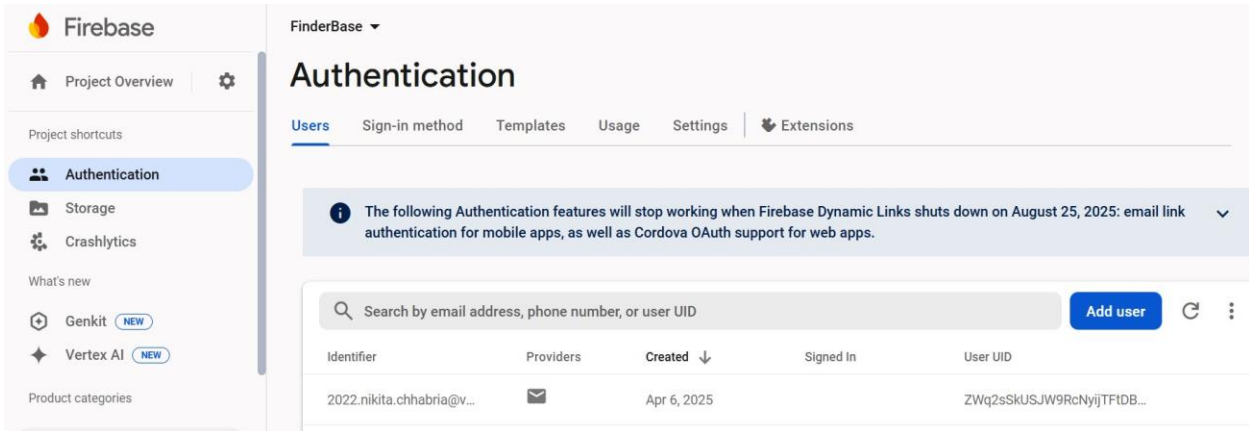
```
    ThemeData(primarySwatch: Colors.deepPurple), home:
```

```
    ReportForm(),
```

```
  );
```

```
}  
  
}
```

Screenshots:



Conclusion:

In this lab, we successfully connected a Flutter app to **Firebase Realtime Database**, submitted data from a UI form, and stored it in the cloud. This allows real-time data syncing, making **FinderBase** more scalable and interactive.