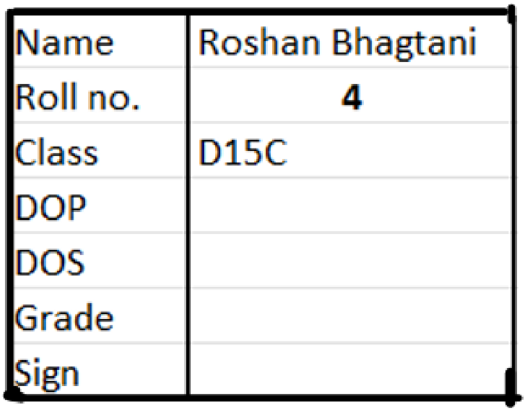
Name: Roshan Bhagtani Class: D15C/ Batch A Roll No:4

Experiment 06



Name	Roshan Bhagtani
Roll no.	**4**
Class	D15C
DOP	
DOS	
Grade	
Sign	

Aim: To Connect Flutter UI with fireBase database.

Theory:

Connecting a Flutter UI with a Firebase database allows for seamless interaction between the app and cloud-based services, enabling features like real-time data storage, retrieval, and user management. To begin, Firebase needs to be integrated into a Flutter project, which involves setting up a Firebase project and adding the necessary Firebase configuration files to the Flutter app. Firebase provides several services that can be integrated with the app, including Firestore for cloud database management, Firebase Authentication for user management, and Firebase Storage for handling media files. Once Firebase is configured, developers can use the `firebase_core` package to initialize Firebase in the app. With Firestore, data can be fetched and displayed using a `StreamBuilder`, which listens for real-time updates. This allows the app

UI to automatically reflect changes in the database without needing to refresh. Data can also be sent to Firestore using methods like `add()` or `set()`, enabling the app to store user data or other relevant information. Furthermore, Firebase Authentication can be used for handling user sign-ins and sign-ups, ensuring secure access to the app. The integration also allows for data updates, ensuring that any changes to the database, such as updating an order status or user information, are immediately reflected in the app. By connecting Flutter UI with Firebase, developers can leverage Firebase's cloud infrastructure to create scalable, dynamic applications that are capable of providing real-time experiences and robust backend management.

Implementation:

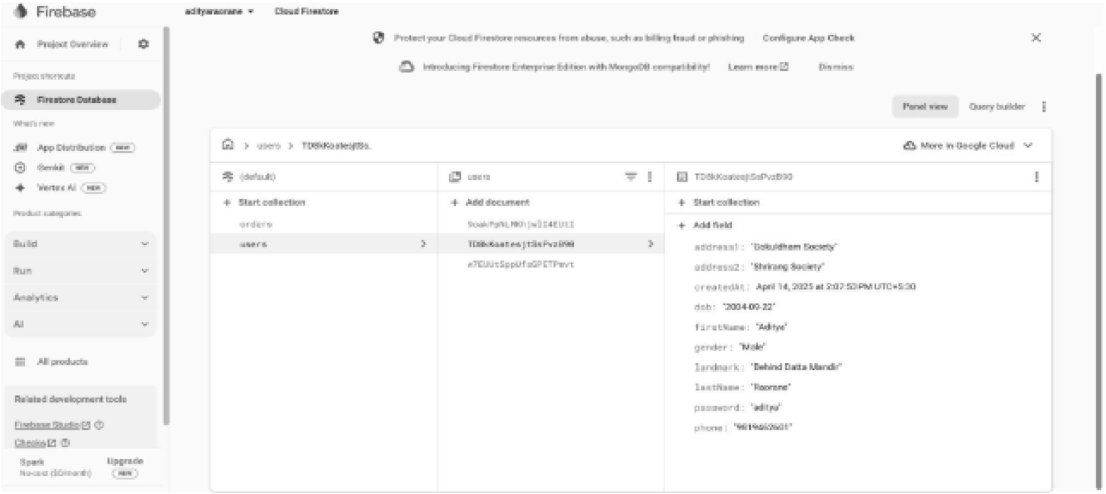
To create a web app using Firebase and connect it to your Firebase project, follow these steps:

1. **Set up Firebase Project**: Start by creating a Firebase project on the Firebase Console.

Go to the Firebase Console, sign in with your Google account, and create a new project. Enable the necessary Firebase services like Firestore, Firebase Authentication, and others based on your app's needs.

2. **Add Firebase to Web App**: Once your project is set up, you'll need to configure your Firebase project for the web. In the Firebase Console, navigate to your project settings and select the option to add a web app. This will generate a unique configuration snippet that includes details like the Firebase API key, project ID, and other identifiers.

Output:



Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing

Introducing Firestore Enterprise Edition with MongoDB compatibility!

orders
users

SoakFpNLMKhjwlI4EUtI
TD8kKoatesjtSsPvzB98
e7EUUtSppUfoQPETPmvt

TD8kKoatesjtSsPvzB90

```
address1: "Gokuldham Society"
address2: "Shrirang Society"
createdAt: April 14, 2025 at 2:07:53 PM UTC+5:30
dob: "2004-09-22"
firstName: "Aditya"
gender: "Male"
landmark: "Behind Datta Mandir"
lastName: "Raorane"
password: "aditya"
phone: "9619692601"
```

Conclusion:

Integrating Firebase with a web application provides an efficient and scalable solution for backend services such as real-time databases, authentication, and hosting. By following the steps to set up Firebase, including configuring your Firebase project, adding the necessary SDKs, and utilizing Firebase services like Firestore and Authentication, you can easily manage user data and interactions in your web app. Firebase simplifies backend development, enabling developers to focus on building and deploying feature-rich applications without managing complex infrastructure. Additionally,