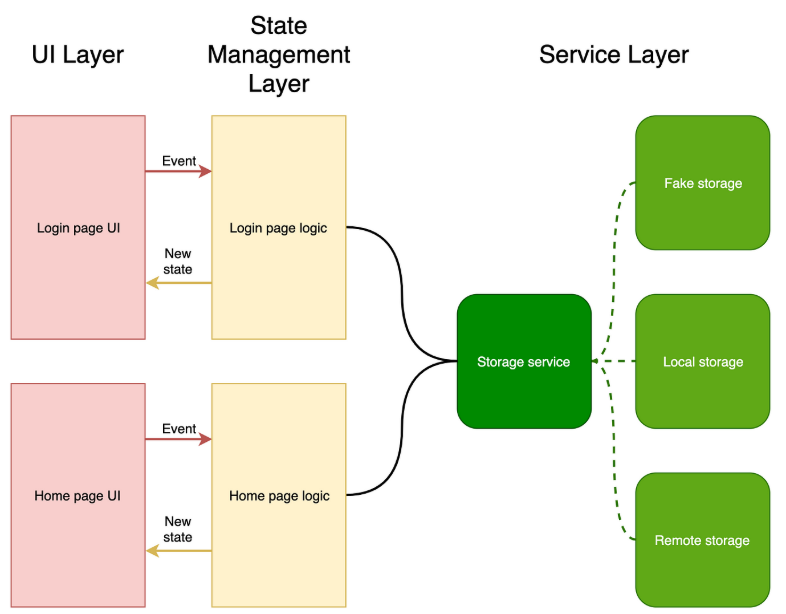
**State management**

State management in Flutter refers to how you handle and update the data that affects your app's UI. It's crucial for building complex and responsive applications, ensuring data consistency and efficient UI updates. Flutter offers various approaches, including built-in methods like StatefulWidget and external packages like Provider, Riverpod, and BLoC.

When the login button is pressed, it triggers an event (a function call like loginUser()), which executes the login logic—such as validating inputs, sending API/Firebase requests, updating the UI based on success or error, and notifying listeners (in Provider) to rebuild UI widgets.

What is Provider?

Provider is a **state management solution** in Flutter that allows you to **share and update data (state)** across your widget tree in a clean, reactive, and efficient way. It works by using the **ChangeNotifier class**, which notifies the widgets when there’s a change in data, allowing them to rebuild only when necessary.

A screenshot of a computer program

AI-generated content may be incorrect.A screen shot of a computer program

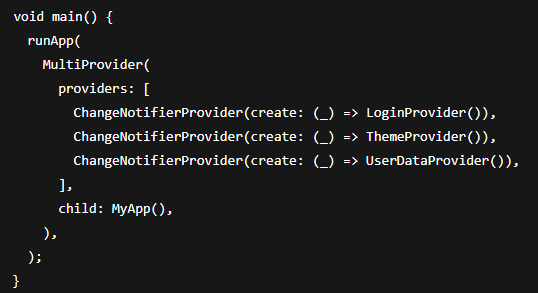
AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.A computer screen shot of a black screen

AI-generated content may be incorrect.

Use Case of MultiProvider

* Your app uses **more than one provider** (e.g., authentication, products, cart, etc.).
* You want to **reduce nested code** and improve readability.
* You want to manage **separate app logic layers** like state for UI, auth, and data.

✅ With MultiProvider (Clean & Scalable):

A screen shot of a computer program

AI-generated content may be incorrect.❌ Without MultiProvider (Nested Providers – messy):