State Management in Flutter

State Management:

While building an application that involves Dynamic-user interactions and Real-time Updates we use State Management

State Management is a core concept in an interactive application that:

1- Provides seamless user experience
2- Keeps data consistent.
3- Provides optimal performance.

What is State?

State is a collection data representing information that is used to create UI.

It requires ***refreshing*** the ***UI*** to ***align with the State***.

Encompasses data that may change while a ***widget*** is ***active.***

Involves user interactions such as ***button clicks*** and ***data retrieved with the help of APIs.***

Types of States—

1- **Ephemeral State**:
The ***state*** that is handled in a ***single widget***. Example the ***checked state of a CheckBox.***

2- App State:
A ***State*** that needs to be shared across **the *multiple parts of an application***.
Example ***theme settings*** **,** ***user authentication status.***

Reasons for State Management:

State Management ensures that UI displays the most recent accurate data.

1- To separate business login from UI design.
Significance: Important to manage the Codebase with large data Volume.

2- *To optimize application to its fullest potential*.
Significance: Avoid using the setState() for every small widget.

Basic State Management Techniques:

1- **Use of setSate():**

    Example code snippet : Click Here

2- **Use of Inherited Widget:**

For more complex state Management the Inherited Widget should be used. This allows you to pass the data down the Widget Tree and react to changes.

Example Code Snipped: Click Here