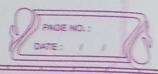
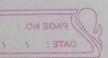
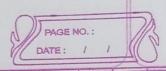
Module-3



Introduction to Flutter Widget and UI Components. Explain the difference between Stateless and Stateful widgets with examples. Stateless Widget: (11) A statelesswidget does not change once it's built (2.) It's like a photo - came every time we look at it. (3) When to use: when the screen or UT doesn't need to change. (4) 7+'s used for static content like labell, icons, ox titles (5) Example: import package: flutter/material.dart: Hun App (Materia lApp (home: My widget()); class Mywidget extends Statelesswidget & Doverdide. Widget build (Build Context context) 5 Hetwin Scaffold (body: Center (child: Text(" I never change!").





(2) Stateful widget: (1.) A Stateful Widget can change its content while the app is running.
(2) It's like a video - it moves and updates. (3.) When to use: when the UI changes based on user actions. (4.) It's used for counting button tops, switching themes, showing or hiding items, etc. (5.) Example: void main() & HUNApp (Material App (home: Mywidget()); managed in diaterial . class thywidget extends Statefulwidget & MywidgetState createstate() => . MywidgetState(); class Mywidgetstate extends State & int value = 0; poveruide widget build (Build Context context) & retwin Scaffold body: Center (Continue child: Column (SizedBox (height: 15), Text ("Counter Value: \$value"), SizedBox (height: 15), Elevated Button (onPressed: () E setstate(1) E

15 1800 3); Chillian Color

1 903, 1014 1111111 child: Text("Count")

period of periode with paids

2. Describe the widget lifecycle and how state is managed in stateful widgets. When we use a Stateful Widget, Flutter needs to:

· Create it

· Update it when things change

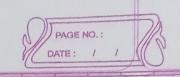
· Destroy it when no longer needed This jowiney from birth - changes - death is called the widget lifetycle.

Lifewcle of a Statful Widget:

A statefulwidget has two parts:
The widget (the bureprint / UI description

. The state (the actual data that can change). wit sould restain " tool

Here are the important lifecycle methods (steps):



(1) createstates: called only once when the widget is created. It makes the ctate object that holds

(2) initstate(): Rune once at the start, when the state object is first created. Best place to set up things like:

· Fetching initial data

· Starting animation

· setting défault values.

13) build(): Called whenever the UI needs to be drawn.

It shows the current state on the

Runc again when cetstate() is called. ingle child widget us can de

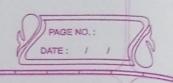
(4.) setState(): Used to update the data (state).

when called, it rebuilds the widget so the UI shows the new data. Example: incrementing a counter.

(5) didUpdateWidget(): Rune when the widget is suplaced by another widget of the same type. Example: When pavent widget sends

new data. 3) Column: Assignation without

(6) dispose (): Called when the widget is removed permanently.



Good place to:

- · Close stelams
- · Stop animations
- · Free up resources.

Managing State: The state (data that can change) lives incide the State object, not in the widget itself.

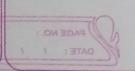
To update vJ, we can never build rebuild the widget manually, instead we call setstated, and flutter rebuilds it for y ws.

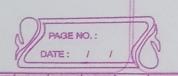
3. List and descuibe for common Flutter layout widgets le.g., Container, Column, Row).

(1) Container: A box that can hold a cingle child widget we can ctyle it with padding, margin, color, borders, and size.

- ouse when we need to decorate or position a widget
- (2) Row: Aswanger widget in hostizont--al line (left + right).
- · Use when we want item cide by side.
- · Example: buttons next to each other.
- (3) Column: Axuanges widgets in a ventical linux(top > botton).

· Use when we want items, stacked.





- · Example: four fields placed one below another.
- (4) Stack: Places widgets on top of each other (like layers).
- each other (like layers).

 · Use when we want to overlap items.

 · Example: putting text over an image
- (5.) Expanded: Used inside a Row or Column to make a widget take available space.
- · Use when we want one widget to stretch and fill space while others

take their needed size.