AIM: Flutter program using layout, widget and state.

management.

❖ THEORY:

- ○ **Flutter:**

  1. Flutter is a free and open-source mobile UI framework created by Google and released in May 2017.

  2. In a few words, it allows you to create a native mobile application with only one codebase.

  3. This means that you can use one programming language and one codebase to create two different apps (for iOS and Android).

  4. Flutter consists of two important parts:

     a. An SDK (Software Development Kit): A collection of tools that are going to help you develop your applications. This includes tools to compile your code into native machine code (code for iOS and Android).

     b. A Framework (UI Library based on widgets): A collection of reusable UI elements (buttons, text inputs, sliders, and so on) that you can personalize for your own needs.

  5. To develop with Flutter, you will use a programming language called Dart.

  6. Dart focuses on front-end development, and you can use it to create mobile and web applications.

o **Layouts in Flutter:**

1. The main concept of the layout mechanism is the widget.

2. We know that flutter assume everything as a widget.

3. So, the image, icon, text, and even the layout of your app are all widgets.

4. Here, some of the things you do not see on your app UI, such as rows, columns, and grids that arrange, constrain, and align the visible widgets are also the widgets.

5. Flutter allows us to create a layout by composing multiple widgets to build more complex widgets.

6. We can categories the layout widget into two types:

   a. Single Child Widget: The single child layout widget is a type of widget, which can have only one child widget inside the parent layout widget. These widgets can also contain special layout functionality. Flutter provides us many single child widgets to make the app UI attractive. If we use these widgets appropriately, it can save our time and makes the app code more readable.

   b. Multiple Child Widget: The multiple child widgets are a type of widget, which contains more than one child widget, and the layout of these widgets are unique. For example, Row widget laying out of its child widget in a horizontal direction, and Column widget laying out of its child widget in a vertical direction. If we combine the

Row and Column widget, then it can build any level of the complex widget.

o **Flutter State Management:**

1. The widget can be classified into two categories, one is a Stateless widget, and another is a Stateful widget.

2. The Stateless widget does not have any internal state.

3. A Stateful widget is dynamic and has a state. It means we can modify it easily throughout its lifecycle without reinitialized it again.

4. A state is information that can be read when the widget is built and might change or modified over a lifetime of the app.

5. If you want to change your widget, you need to update the state object, which can be done by using the setState() function available for Stateful widgets.

6. In Flutter, the state management categorizes into two conceptual types, which are given below:

   a. Ephemeral State: It is a type of state which is related to the specific widget, or you can say that it is a state that contains in a single widget. In this kind of state, you do not need to use state management techniques. The common example of this state is Text Field.

   b. App State: It is different from the ephemeral state. It is a type of state that we want to share across various

parts of our app and want to keep between user
sessions. Thus, this type of state can be used globally.

A) Create a widget ProductBox that contains the details of the product, such as image, name, price, and description. In the ProductBox widget, we use the following child widgets: Container, Row, Column, Expanded, Card, Text, Image, etc. This widget contains the following layout:

- **CODE:**
  - **main.dart:**

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

void main() {
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo', theme: ThemeData(
      primarySwatch: Colors.blue,
    ),
      home: MyHomePage(title: 'Product layout demo home page'),
    );
  }
}
class MyHomePage extends StatelessWidget {
  MyHomePage({Key? key, required this.title}) : super(key: key);
  final String title;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Product Listing")),
      body: ListView(
        shrinkWrap: true,
        padding: const EdgeInsets.fromLTRB(2.0, 10.0, 2.0, 10.0),
        children: <Widget>[
          ProductBox(
```

```dart
          name: "iPhone",
          description: "If you have a kidney you have a iPhone",
          price: 99999,
          image: "iPhone.png"
        ),
        ProductBox(
          name: "Pixel",
          description: "Pixel, garbage thing!!",
          price: 50000,
          image: "Pixel.png"
        ),
        ProductBox(
          name: "Laptop",
          description: "Alienware feel the POWER OF ALIEN!!!",
          price: 200000,
          image: "Laptop.png"
        ),
        ProductBox(
          name: "Tablet",
          description: "Ipad the one to eat your money!!!!",
          price: 60000,
          image: "Tablet.png"
        ),
        ProductBox(
          name: "Pendrive",
          description: "Pendrive FAST SMOOTH AND IT SAVES YOUR DATA!!!",
          price: 200,
          image: "Pendrive.png"
        ),
        ProductBox(
          name: "Floppy Drive",
          description: "Floppy drive???!!! Go visit a DOCTOR!!!",
          price: 0,
          image: "Floppy.png"
        ),
      ],
    )
  );
 }
}
class ProductBox extends StatelessWidget {
  ProductBox({Key? key, required this.name, required this.description, required this.price, required
this.image}) :
      super(key: key);
```

```dart
final String name;
final String description;
final int price;
final String image;

Widget build(BuildContext context) {
  return Container(
    padding: EdgeInsets.all(2),
    height: 120,
    child: Card(
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: <Widget>[
          Image.asset("assets/" + image),
          Expanded(
            child: Container(
              padding: EdgeInsets.all(5),
              child: Column(
                mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                children: <Widget>[
                  Text(
                    this.name, style: TextStyle(
                    fontWeight: FontWeight.bold
                  )
                  ),
                  Text(this.description), Text(
                    "Price: " + this.price.toString()
                  ),
                ],
              )
            )
          )
        ])  )
  );  }}
```
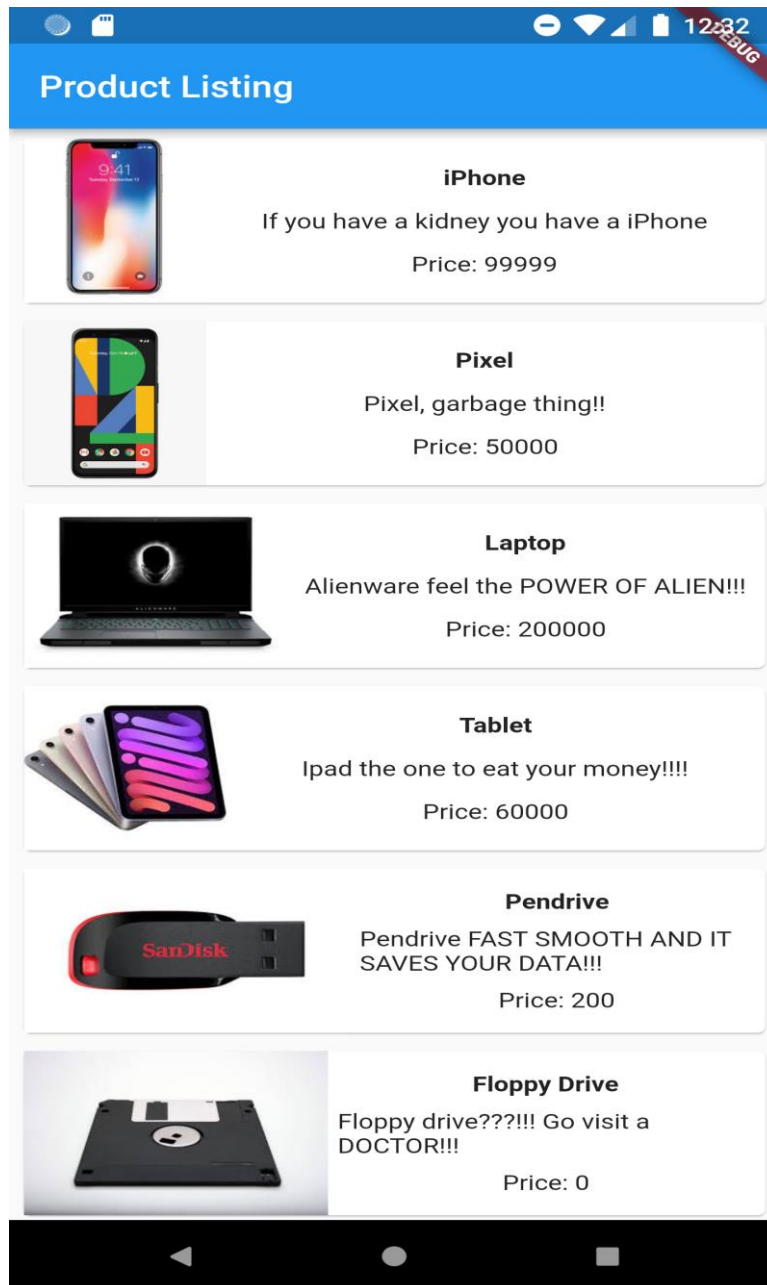
o **pubspec.yaml:**

```yaml
assets:
 - assets/iPhone.png
 - assets/Laptop.png
 - assets/Pendrive.png
 - assets/Pixel.png
```

- assets/Tablet.png
- assets/Floppy.png

- **OUTPUT:**

❖ **CONCLUSION:**

Hence we successfully flutter program using layout, widget and state.