**Aim:** To design Flutter UI by including common widgets.

**Theory:**

Flutter follows a widget-based approach where everything in the UI is a widget. Widgets can be classified into two main types:

- Stateless Widgets: Do not change their state once built (e.g., Text, Container).
- Stateful Widgets: Can update dynamically based on user interaction (e.g., TextField, Checkbox).

Commonly Used Widgets in Flutter-

(a) Scaffold Widget

The Scaffold widget provides the basic structure for a Flutter app, including an AppBar, Drawer, FloatingActionButton, and BottomNavigationBar. It is a fundamental widget used to create a standard screen layout in Flutter.

(b) Container Widget

A Container is a box model widget that can hold other widgets. It is commonly used for adding padding, margins, borders, and background decorations.

(c) Row and Column Widgets

- Row: Arranges widgets horizontally.
- Column: Arranges widgets vertically.
  These two widgets are fundamental for designing layouts in Flutter.

(d) ListView Widget

The ListView widget is used for displaying a scrollable list of items. It is useful for showing large amounts of data dynamically.

(e) Stack Widget

The Stack widget is used to place widgets on top of each other. This is useful for creating overlapping UI elements such as banners, profile images, or layered designs.

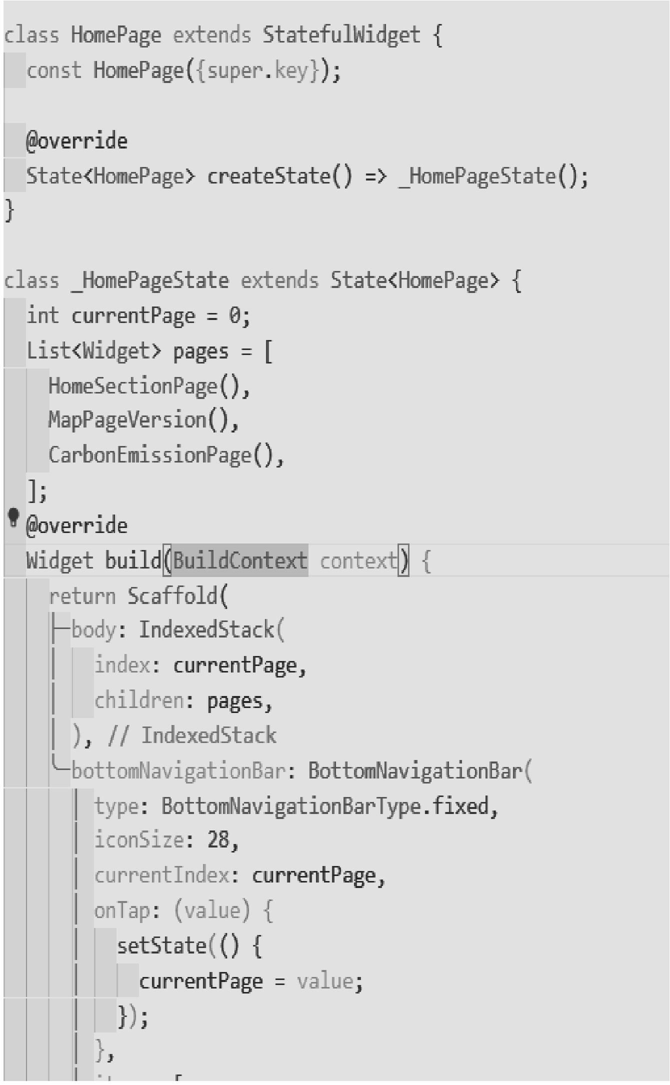
(f) ElevatedButton Widget

The ElevatedButton widget is used for clickable buttons with a raised effect. It is a commonly used button in Flutter applications.

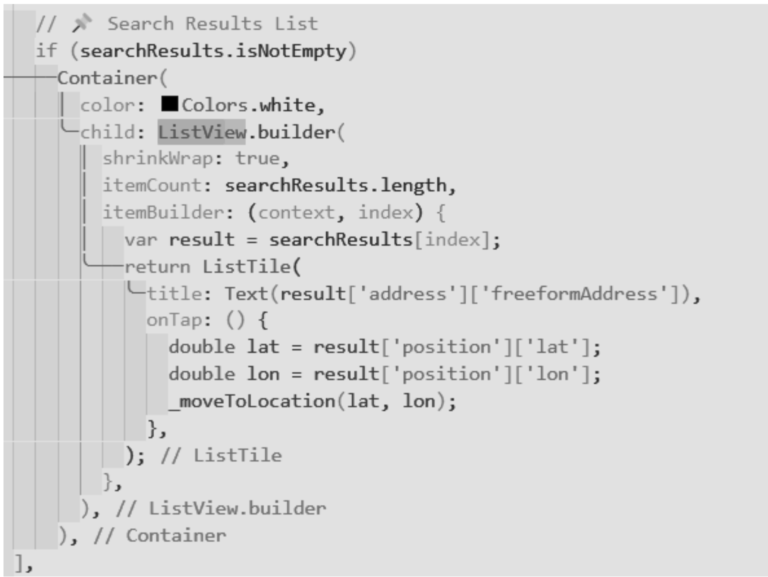
(g) TextField Widget

The TextField widget is used to take user input, such as entering a name, email, or password. It is commonly used in forms and authentication screens.

Code :

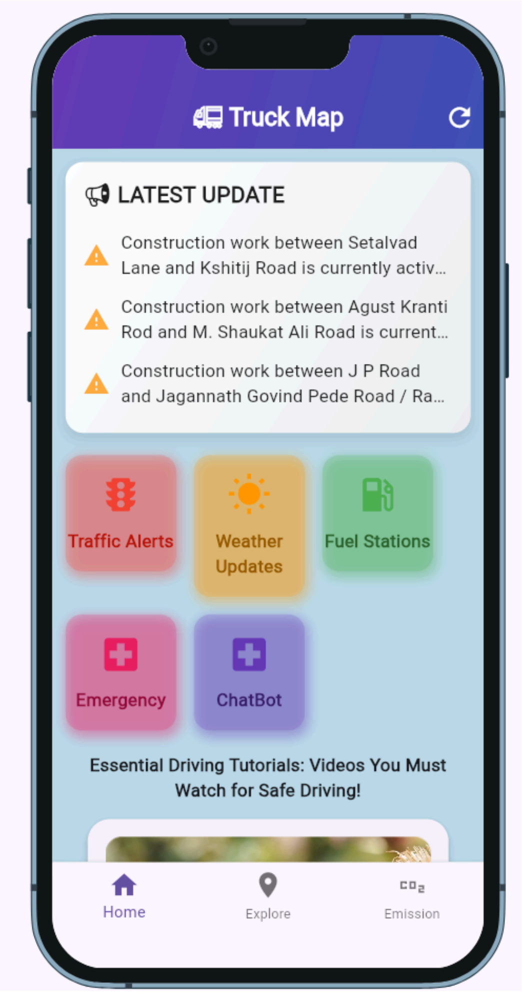


```
class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  int currentPage = 0;
  List<Widget> pages = [
    HomeSectionPage(),
    MapPageVersion(),
    CarbonEmissionPage(),
  ];
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: IndexedStack(
        index: currentPage,
        children: pages,
      ), // IndexedStack
      bottomNavigationBar: BottomNavigationBar(
        type: BottomNavigationBarType.fixed,
        iconSize: 28,
        currentIndex: currentPage,
        onTap: (value) {
          setState(() {
            currentPage = value;
          });
        },
```



```
  // 📌 Search Results List
  if (searchResults.isNotEmpty)
    Container(
      color: ■Colors.white,
      child: ListView.builder(
        shrinkWrap: true,
        itemCount: searchResults.length,
        itemBuilder: (context, index) {
          var result = searchResults[index];
          return ListTile(
            title: Text(result['address']['freeformAddress']),
            onTap: () {
              double lat = result['position']['lat'];
              double lon = result['position']['lon'];
              _moveToLocation(lat, lon);
            },
          ); // ListTile
        },
      ), // ListView.builder
    ), // Container
],
```

Output : -



Truck Map

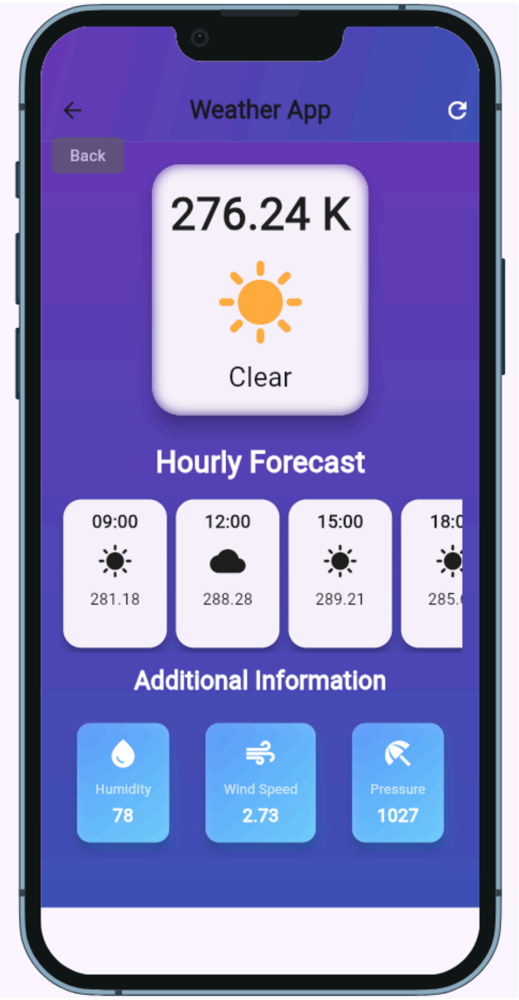
LATEST UPDATE

Construction work between Setalvad Lane and Kshitij Road is currently activ…

Construction work between Agust Kranti Rod and M. Shaukat Ali Road is current…

Construction work between J P Road and Jagannath Govind Pede Road / Ra…

Essential Driving Tutorials: Videos You Must Watch for Safe Driving!



Weather App

Back

276.24 K

Clear

Hourly Forecast

| 09:00 | 12:00 | 15:00 | 18:0 |
|---|---|---|---|
| 281.18 | 288.28 | 289.21 | 285. |

Additional Information

| Humidity | Wind Speed | Pressure |
|---|---|---|
| 78 | 2.73 | 1027 |

**Conclusion:**

Flutter's widget-based architecture allows for flexible and efficient UI design. By using common widgets like `Scaffold`, `Container`, Row, `Column`, `ListView`, `Stack`, `ElevatedButton`, and `TextField`, developers can build responsive and interactive user interfaces with ease.