Exp-3

NAME: Mohit Tarachandani
Batch : C
Class : D15A
Roll No. : 62

Aim: To include icons, images,fonts in Flutter app

Theory:

Icons: Flutter provides built-in support for incorporating icons into your app's UI. The Icons class offers a wide range of material design icons that can be used out of the box. Icons can be added to your app using widgets like Icon or IconButton. Additionally, Flutter allows you to use custom icons by importing them as assets and referencing them using the Image.asset widget.

Images: Flutter supports various image formats such as JPEG, PNG, GIF, WebP, and SVG (using the flutter_svg package). You can include images in your app from different sources like local assets, network URLs, memory, or file paths. To use images from local assets, add them to your project's pubspec.yaml file under the assets section and reference them using the Image.asset widget. Network images can be loaded using the Image.network widget by providing the URL of the image. Memory images can be displayed using the Image.memory widget, where you pass the image data as a byte array.

Custom Fonts: Flutter allows you to use custom fonts to enhance the typography of your app. To include custom fonts, add the font files (typically .ttf or .otf files) to your project's pubspec.yaml file under the fonts section. After adding the font files, run flutter pub get to fetch the fonts. You can then specify the custom font in the TextStyle of a Text widget by referring to the font family defined in the pubspec.yaml file.

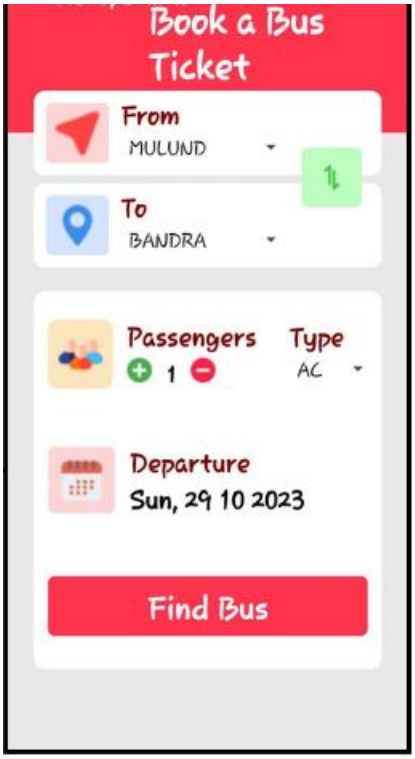
Code:

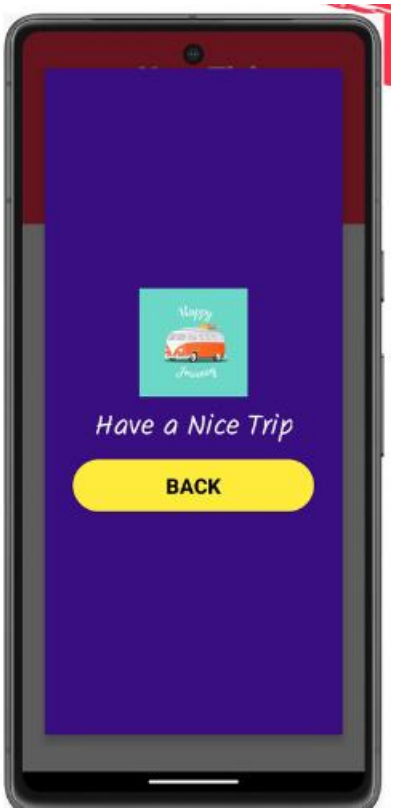
```
import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

class BookingPage extends StatefulWidget {
  final String from;
  final String to;
  final String date;
  final String tPass;
  final String email;
  final String name;

  BookingPage({
    required this.from,
    required this.to,
    required this.date,
    required this.tPass,
```

```
    required this.email,
    required this.name,
  });

  @override
  _BookingPageState createState() => _BookingPageState();
}

class _BookingPageState extends State<BookingPage> {
  late FirebaseFirestore db;
  late List<Trip> trips;

  @override
  void initState() {
    super.initState();
    db = FirebaseFirestore.instance;
    trips = [];
    if (widget.from == "DHAKA" && widget.to == "SYLHET") {
      fetchDhkSylTrips();
    } else if (widget.from == "KHULNA" && widget.to == "RAJSHAHI") {
      fetchKhlRajTrips();
    }
  }

  void fetchDhkSylTrips() {
    db.collection("DhkSyl").orderBy("rTime", descending:
false).snapshots().listen((QuerySnapshot snapshot) {
      setState(() {
        trips = snapshot.docs.map((doc) => Trip.fromMap(doc.data())).toList();
      });
    });
  }

  void fetchKhlRajTrips() {
    db.collection("KhlRaj").orderBy("rTime", descending:
false).snapshots().listen((QuerySnapshot snapshot) {
      setState(() {
        trips = snapshot.docs.map((doc) => Trip.fromMap(doc.data())).toList();
      });
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Booking'),
      ),
      body: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
```

```
        Text('From: ${widget.from}'),
        Text('To: ${widget.to}'),
        Text('Date: ${widget.date}'),
        Text('Pass: ${widget.tPass}'),
        Expanded(
          child: ListView.builder(
            itemCount: trips.length,
            itemBuilder: (BuildContext context, int index) {
              return ListTile(
                title: Text('Trip ${index + 1}'),
                subtitle: Text('Route: ${trips[index].pick} - ${trips[index].drop}'),
                onTap: () {
                  // Handle trip selection
                },
              );
            },
          ),
        ),
      ],
    ),
  );
 }
}

class Trip {
 final String rTime;
 final String tTime;
 final String pick;
 final String drop;
 final String price;

 Trip({
  required this.rTime,
  required this.tTime,
  required this.pick,
  required this.drop,
  required this.price,
 });

 factory Trip.fromMap(Map<String, dynamic> map) {
  return Trip(
   rTime: map['rTime'],
   tTime: map['tTime'],
   pick: map['pick'],
   drop: map['drop'],
   price: map['price'],
  );
 }
}}
```

Output:





Conclusion:

To design simple flutter UI by including common widgets.