

## EXP-2

Name- Anuj Chitari

Div: D15A R011- 11

**AIM-** To include icons, images, fonts in Flutter app

**THEORY: 2** This widget holds the image which can fetch it from multiple sources like from the asset folder or directly from the URL. It provides many constructors for loading image, which are given below:

- o Image: It is a generic image loader, which is used by ImageProvider.

- o asset: It load image from your project asset folder.

- o file: It loads images from the system folder.

- o memory: It load image from memory.

- o network: It loads images from the network.

To add an image in the project, you need first to create an assets folder where you keep your images and then add the below line in pubspec.yaml file.

Assets:

- assets/comp.jpeg

CODE:

```
import 'package:flutter/material.dart';
import 'package:food_panda/admin/home_admin.dart';
import 'package:food_panda/pages/bottomnav.dart';
import 'package:food_panda/pages/onboard.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';
import 'package:food_panda/admin/admin_login.dart';
```

```

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(
    options: FirebaseOptions(
      apiKey: 'AlzaSyCT2s9X9jKNBctZbW5dCRGpyqr1Z_-phM8',
      appId: '1:431523531027:android:562d4418509346182b32e6',
      messagingSenderId: '431523531027',
      projectId: 'foodeliveryapp-5a9ad',
      storageBucket: 'foodeliveryapp-5a9ad.appspot.com',)
  );
  runApp(MyApp()); // Replace MyApp() with your main application widget
}

```

```

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // TRY THIS: Try running your application with "flutter run". You'll see
        // the application has a purple toolbar. Then, without quitting the app,
        // try changing the seedColor in the colorScheme below to Colors.green
        // and then invoke "hot reload" (save your changes or press the "hot
        // reload" button in a Flutter-supported IDE, or press "r" if you used
        // the command line to start the app).
        //
        // Notice that the counter didn't reset back to zero; the application
        // state is not lost during the reload. To reset the state, use hot
        // restart instead.
        //
        // This works for code too, not just values: Most code changes can be
        // tested with just a hot reload.

```

```

        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
    ),
    home: Onboard(),
    // home: HomeAdmin(),
    // home: BottomNav(),
    // home: AdminLogin(),

);
}
}

```

```

class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});

```

```

  // This widget is the home page of your application. It is stateful, meaning
  // that it has a State object (defined below) that contains fields that affect
  // how it looks.

```

```

  // This class is the configuration for the state. It holds the values (in this
  // case the title) provided by the parent (in this case the App widget) and
  // used by the build method of the State. Fields in a Widget subclass are
  // always marked "final".

```

```

  final String title;

```

```

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

```

```

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

```

```

  void _incrementCounter() {
    setState(() {
      // This call to setState tells the Flutter framework that something has
      // changed in this State, which causes it to rerun the build method below
      // so that the display can reflect the updated values. If we changed
      // _counter without calling setState(), then the build method would not be
      // called again, and so nothing would appear to happen.
    });
  }
}

```

```
    _counter++;  
  });  
}
```

@override

```
Widget build(BuildContext context) {  
  // This method is rerun every time setState is called, for instance as done  
  // by the _incrementCounter method above.  
  //  
  // The Flutter framework has been optimized to make rerunning build methods  
  // fast, so that you can just rebuild anything that needs updating rather  
  // than having to individually change instances of widgets.  
  return Scaffold(  
    appBar: AppBar(  
      // TRY THIS: Try changing the color here to a specific color (to  
      // Colors.amber, perhaps?) and trigger a hot reload to see the AppBar  
      // change color while the other colors stay the same.  
      backgroundColor: Theme.of(context).colorScheme.inversePrimary,  
      // Here we take the value from the MyHomePage object that was created by  
      // the App.build method, and use it to set our appbar title.  
      title: Text(widget.title),  
    ),  
    body: Center(  
      // Center is a layout widget. It takes a single child and positions it  
      // in the middle of the parent.  
      child: Column(  
        // Column is also a layout widget. It takes a list of children and  
        // arranges them vertically. By default, it sizes itself to fit its  
        // children horizontally, and tries to be as tall as its parent.  
        //  
        // Column has various properties to control how it sizes itself and  
        // how it positions its children. Here we use mainAxisAlignment to  
        // center the children vertically; the main axis here is the vertical  
        // axis because Columns are vertical (the cross axis would be  
        // horizontal).  
        //  
        // TRY THIS: Invoke "debug painting" (choose the "Toggle Debug Paint"  
        // action in the IDE, or press "p" in the console), to see the  
        // wireframe for each widget.  
        mainAxisAlignment: MainAxisAlignment.center,
```

```

children: <Widget>[
  const Text(
    'You have pushed the button this many times:',
  ),
  Text(
    '$_counter',
    style: Theme.of(context).textTheme.headlineMedium,
  ),
],
),
),
floatingActionButton: FloatingActionButton(
  onPressed: _incrementCounter,
  tooltip: 'Increment',
  child: const Icon(Icons.add),
), // This trailing comma makes auto-formatting nicer for build methods.
);
}

```

## OUTPUT

