Experiment 03

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

Theory:

In a Flutter app, icons, images, and fonts are essential elements for creating visually appealing and interactive user interfaces. Let's explore the theory behind each of these components:

1. Icons:

- Icons in Flutter are graphical symbols that represent actions, objects, or features in your application.
- Flutter provides a set of built-in icons, which can be easily used with the \`Icon\` widget. Examples include icons for navigation, action buttons, and more.
- Custom icons or icons from external sources can be incorporated using the \`Icon\` widget with a specified font package or image.

2. Images:

- Images are essential for displaying graphics, logos, and other visual elements in a Flutter app.
- Flutter supports various image formats, including JPEG, PNG, GIF, and WebP.
- The \`Image\` widget is used to display images in Flutter, and you can load images from the network, assets, or local files.

3. Fonts:

- Fonts play a crucial role in defining the text style and visual appeal of your app.
- Flutter allows you to use custom fonts in your app, and you can load them from assets or network resources.
- The \`TextStyle\` class is used to define the appearance of text, including font family, size, color, etc.

4. Asset Management:

- To use images, icons, or custom fonts, you need to manage assets in your Flutter project.
- Add the assets to your \`pubspec.yaml\` file, and use the \`pubspec.yaml\` file to specify the location of these assets.

5. Network Resources:

- In addition to local assets, Flutter also allows you to load images and fonts from network resources using widgets like `Image.network` or by specifying the font URL.

Code:

Main.dart:

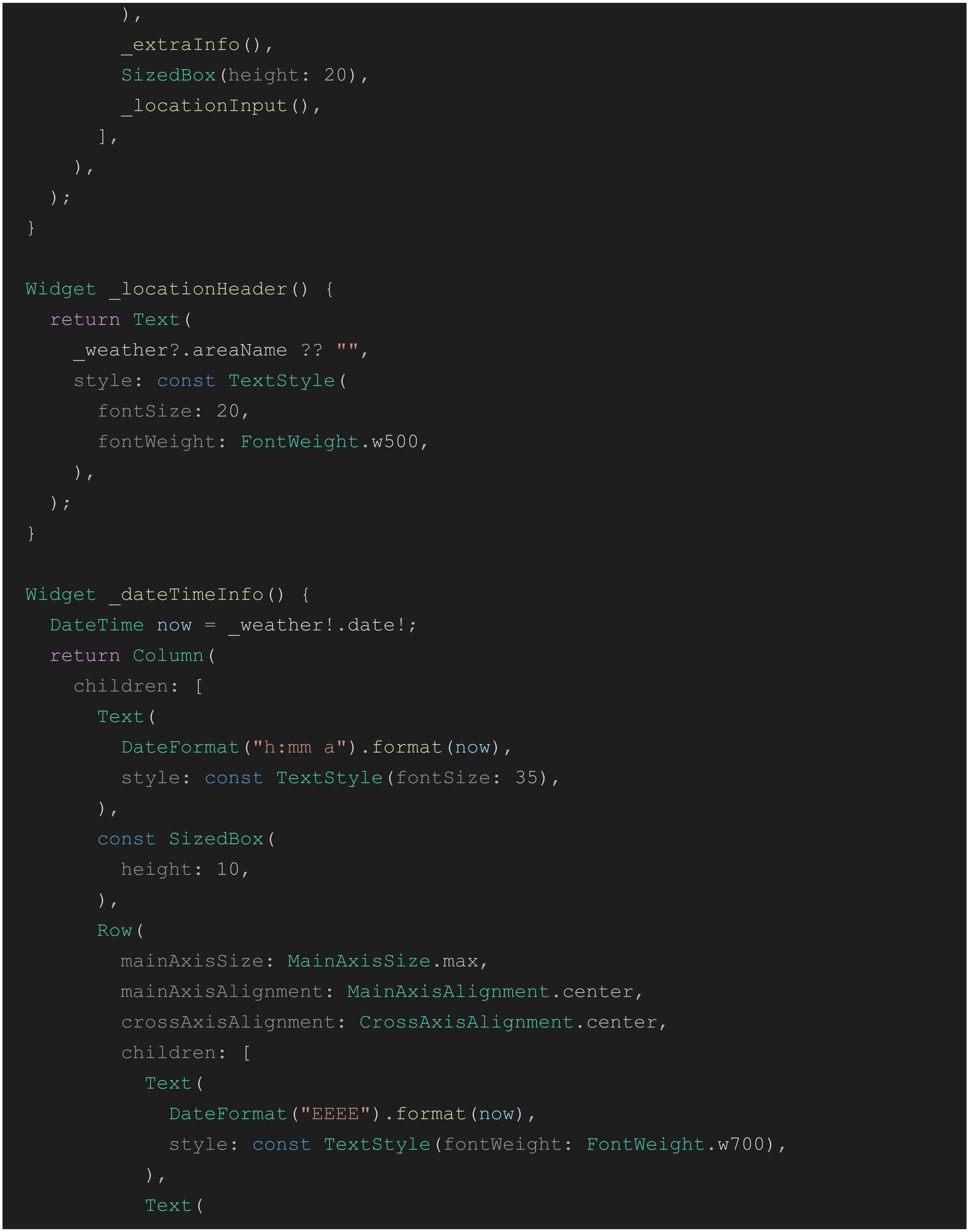
```
import 'package:flutter/material.dart';
import 'package:weather_app/pages/home_page.dart';

void main() {
  runApp(const MyApp());
}

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}

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

  final String title;

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

```
class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return HomePage();
  }
}
```

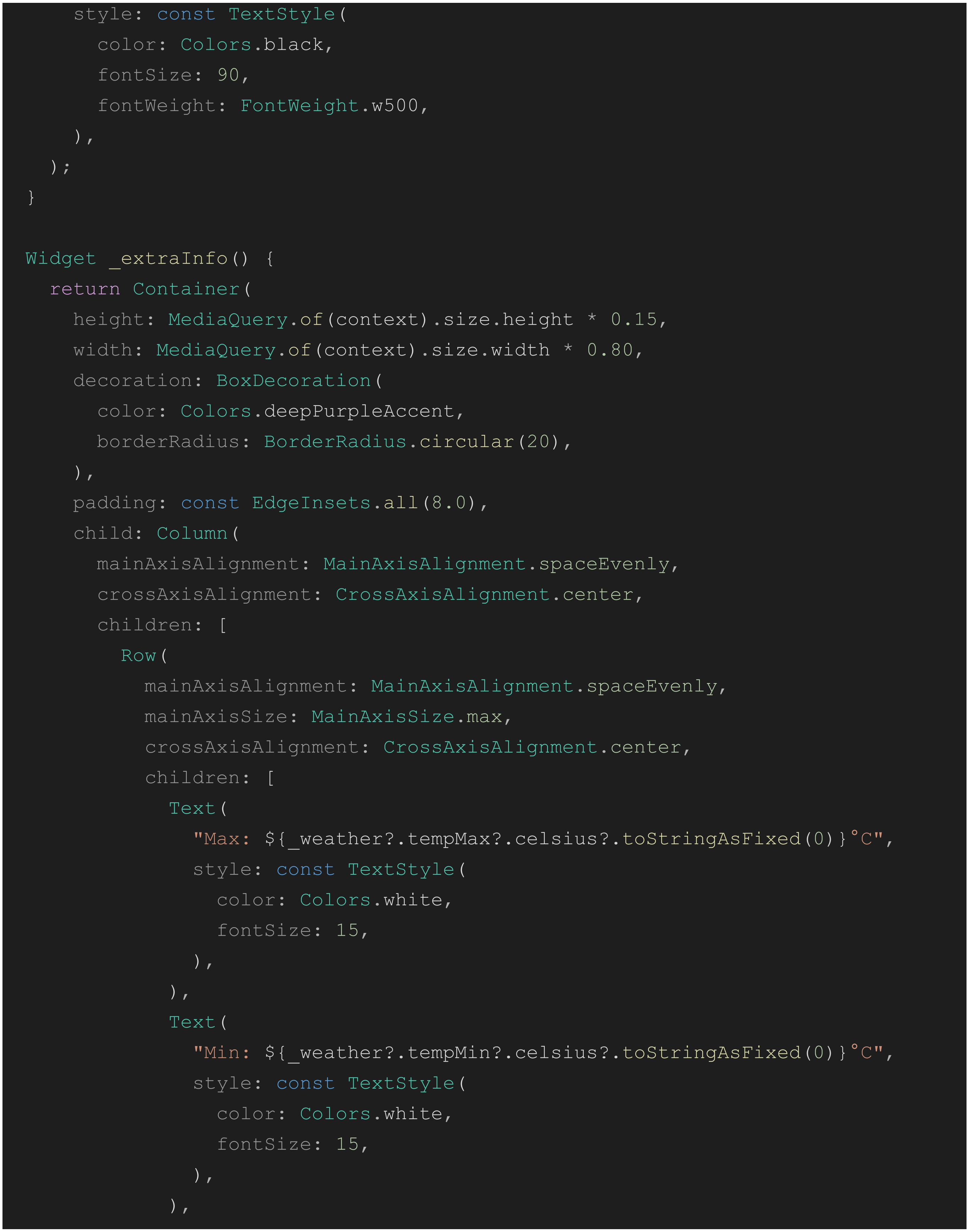
home_page.dart:

```
import 'package:flutter/material.dart';
import 'package:intl/intl.dart';
import 'package:weather/weather.dart';
import 'package:weather_app/consts.dart';

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

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

class _HomePageState extends State<HomePage> {
  final WeatherFactory _wf = WeatherFactory(OPENWEATHER_API_KEY);

  Weather? _weather;
  TextEditingController _locationController = TextEditingController();

  @override
  void initState() {
    super.initState();
    _getLocationWeather("Mumbai"); // Default location, you can change this
  }

  void _getLocationWeather(String location) {
    _wf.currentWeatherByCityName(location).then((w) {
      setState(() {
        _weather = w;
      });
    });
```

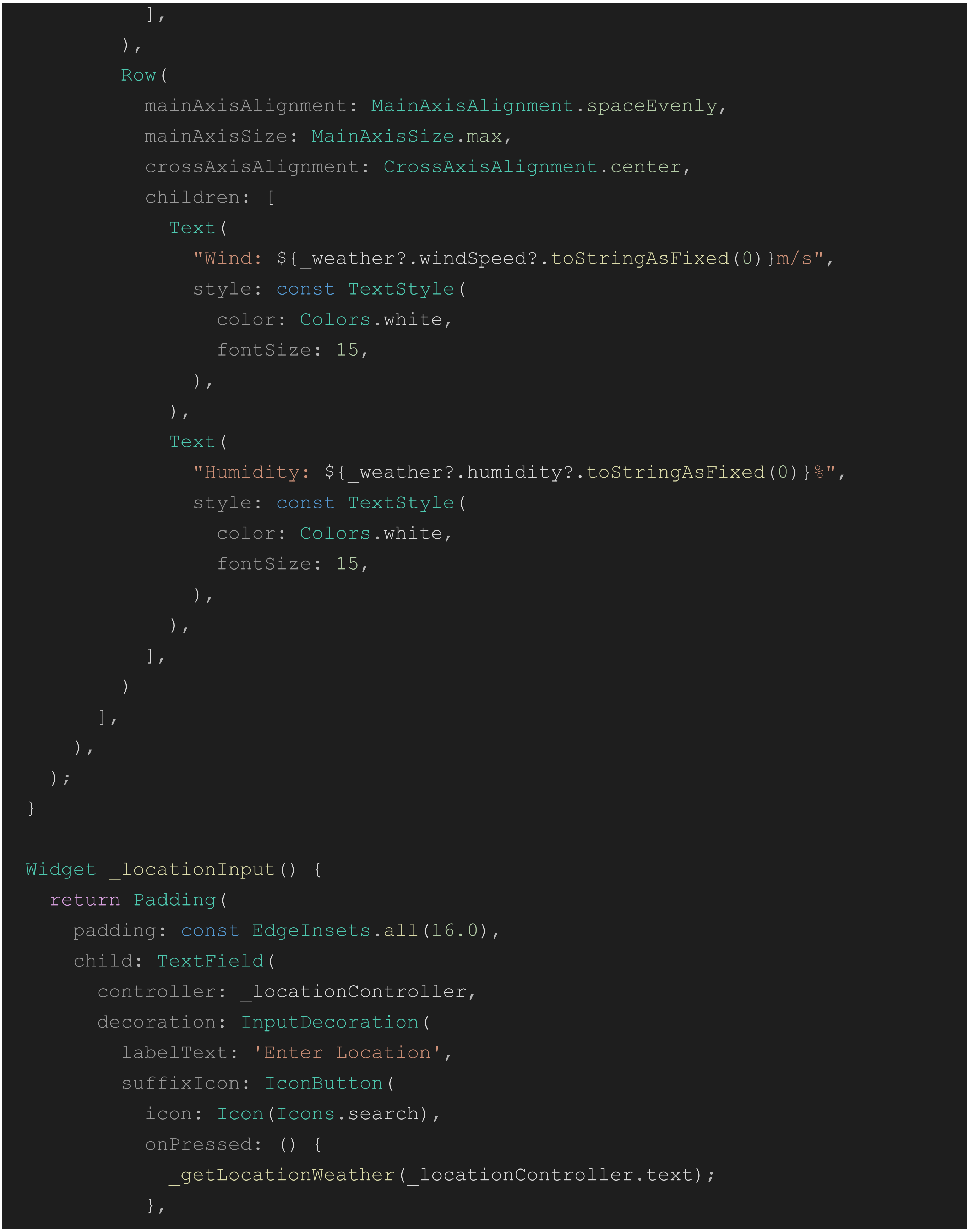
```
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: _buildUI(),
  );
}

Widget _buildUI() {
  if (_weather == null) {
    return const Center(
      child: CircularProgressIndicator(),
    );
  }

  return SingleChildScrollView(
    child: Column(
      mainAxisSize: MainAxisSize.max,
      mainAxisAlignment: MainAxisAlignment.center,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: [
        SizedBox(
          height: MediaQuery.of(context).size.height * 0.05,
        ),
        _locationHeader(),
        SizedBox(
          height: MediaQuery.of(context).size.height * 0.05,
        ),
        _dateTimeInfo(),
        SizedBox(
          height: MediaQuery.of(context).size.height * 0.05,
        ),
        _weatherIcon(),
        SizedBox(
          height: MediaQuery.of(context).size.height * 0.02,
        ),
        _currentTemp(),
        SizedBox(
          height: MediaQuery.of(context).size.height * 0.02,
```



```
        ),
        _extraInfo(),
        SizedBox(height: 20),
        _locationInput(),
      ],
    ),
  );
}

Widget _locationHeader() {
  return Text(
    _weather?.areaName ?? "",
    style: const TextStyle(
      fontSize: 20,
      fontWeight: FontWeight.w500,
    ),
  );
}

Widget _dateTimeInfo() {
  DateTime now = _weather!.date!;
  return Column(
    children: [
      Text(
        DateFormat("h:mm a").format(now),
        style: const TextStyle(fontSize: 35),
      ),
      const SizedBox(
        height: 10,
      ),
      Row(
        mainAxisSize: MainAxisSize.max,
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.center,
        children: [
          Text(
            DateFormat("EEEE").format(now),
            style: const TextStyle(fontWeight: FontWeight.w700),
          ),
          Text(
              " ${DateFormat("d.M.y").format(now)}",
              style: const TextStyle(fontWeight: FontWeight.w700),
            )
          ],
        )
      ],
    );
  }

  Widget _weatherIcon() {
    return Column(
      mainAxisSize: MainAxisSize.min,
      mainAxisAlignment: MainAxisAlignment.center,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: [
        Container(
          height: MediaQuery.of(context).size.height * 0.20,
          decoration: BoxDecoration(
            image: DecorationImage(
              image: NetworkImage(

"https://openweathermap.org/img/wn/${_weather?.weatherIcon}@4x.png",
              ),
            ),
          ),
        ),
        Text(
          _weather?.weatherDescription ?? "",
          style: const TextStyle(
            color: Colors.black,
            fontSize: 20,
          ),
        )
      ],
    );
  }

  Widget _currentTemp() {
    return Text(
      "${_weather?.temperature?.celsius?.toStringAsFixed(0)}°C",
```

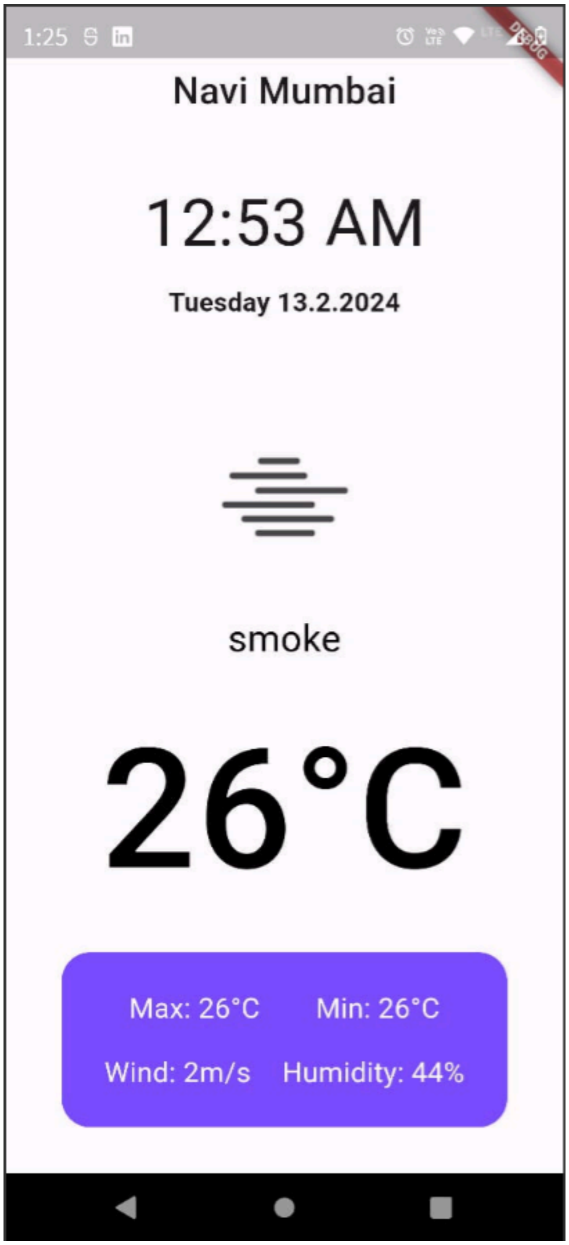


```
    style: const TextStyle(
      color: Colors.black,
      fontSize: 90,
      fontWeight: FontWeight.w500,
    ),
  );
}

Widget _extraInfo() {
  return Container(
    height: MediaQuery.of(context).size.height * 0.15,
    width: MediaQuery.of(context).size.width * 0.80,
    decoration: BoxDecoration(
      color: Colors.deepPurpleAccent,
      borderRadius: BorderRadius.circular(20),
    ),
    padding: const EdgeInsets.all(8.0),
    child: Column(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      crossAxisAlignment: CrossAxisAlignment.center,
      children: [
        Row(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          mainAxisSize: MainAxisSize.max,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            Text(
              "Max: ${_weather?.tempMax?.celsius?.toStringAsFixed(0)}°C",
              style: const TextStyle(
                color: Colors.white,
                fontSize: 15,
              ),
            ),
            Text(
              "Min: ${_weather?.tempMin?.celsius?.toStringAsFixed(0)}°C",
              style: const TextStyle(
                color: Colors.white,
                fontSize: 15,
              ),
            ),
```



```
          ],
        ),
        Row(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          mainAxisSize: MainAxisSize.max,
          crossAxisAlignment: CrossAxisAlignment.center,
          children: [
            Text(
              "Wind: ${_weather?.windSpeed?.toStringAsFixed(0)}m/s",
              style: const TextStyle(
                color: Colors.white,
                fontSize: 15,
              ),
            ),
            Text(
              "Humidity: ${_weather?.humidity?.toStringAsFixed(0)}%",
              style: const TextStyle(
                color: Colors.white,
                fontSize: 15,
              ),
            ),
          ],
        )
      ],
    ),
  );
}

Widget _locationInput() {
  return Padding(
    padding: const EdgeInsets.all(16.0),
    child: TextField(
      controller: _locationController,
      decoration: InputDecoration(
        labelText: 'Enter Location',
        suffixIcon: IconButton(
          icon: Icon(Icons.search),
          onPressed: () {
            _getLocationWeather(_locationController.text);
          },
          ),
        ),
      ),
    );
  }
}
```

Output:



Navi Mumbai

12:53 AM

Tuesday 13.2.2024

smoke

26°C

Max: 26°C   Min: 26°C

Wind: 2m/s   Humidity: 44%

Conclusion:

By using icons, images, and fonts effectively, you can enhance the visual appeal and user experience of your Flutter applications.