

```

import 'dart:convert';
import 'package:flutter/material.dart';
import 'package:http/http.dart' as http;

void main() {
  runApp(WeatherApp());
}

class WeatherApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData.dark(),
      home: WeatherScreen(),
      debugShowCheckedModeBanner: false,
    );
  }
}

class WeatherScreen extends StatefulWidget {
  @override
  WeatherScreenState createState() => WeatherScreenState();
}

class WeatherScreenState extends State<WeatherScreen> {
  final TextEditingController cityController = TextEditingController();
  String weather = '';
  String temperature = '';
  String description = '';
  String imageUrl = '';
  String cityname = '';

  Future<void> fetchWeather(String city) async {
    final apiKey = 'fe82f67fa387c1969d71b9295fa43d37'; // Replace with your key
    final url = Uri.parse(
      'https://api.openweathermap.org/data/2.5/weather?q=$city&appid=$apiKey&units=metric',
    );

    try {
      final response = await http.get(url);

      if (response.statusCode == 200) {
        final data = json.decode(response.body);
        setState(() {
          cityname = data['name'];
          weather = data['weather'][0]['main'];
          description = data['weather'][0]['description'];
          temperature = '${data['main']['temp']} °C';
          imageUrl = 'https://openweathermap.org/img/wn/${data['weather'][0]['icon']}@2x.png';
        });
      } else {
        setState(() {
          weather = 'City not found';
          description = '';
          temperature = '';
          imageUrl = '';
        });
      }
    } catch (e) {
      setState(() {
        weather = 'Error fetching weather';
        description = '';
        temperature = '';
        imageUrl = '';
      });
    }
  }
}

```

```

}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(title: Text('Weather App')),
    body: Padding(
      padding: EdgeInsets.all(16),
      child: Column(
        children: [
          TextField(
            controller: cityController,
            decoration: InputDecoration(
              labelText: 'Enter City',
              border: OutlineInputBorder(),
            ),
          ),
          SizedBox(height: 16),
          ElevatedButton(
            onPressed: () => fetchWeather(cityController.text),
            child: Text('Get Weather'),
          ),
          SizedBox(height: 32),
          if (weather.isNotEmpty) ...[
            Text(cityname, style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),
            if (iconUrl.isNotEmpty) Image.network(iconUrl),
            Text(weather, style: TextStyle(fontSize: 32, fontWeight: FontWeight.bold)),
            Text(description, style: TextStyle(fontSize: 20)),
            Text(temperature, style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold)),
          ],
        ],
      ),
    ),
  );
}

```