My Weather Dashboard breakdown by file.

Core:

**Api\_client.py**: This code fetches the current weather and 5-day forecast for a given city using the OpenWeatherMap API. It sends two HTTP requests—one to get current conditions and another for the forecast—using a specified API key and city name. The responses are returned in JSON format and converted to Python dictionaries. The function fetch\_weather\_data(city) handles all of this and returns both the current weather and forecast data so they can be used elsewhere in the program.

**Utils.py:** This code downloads an image (like a weather icon) from the internet using a URL, resizes it to 80x80 pixels (or a size you choose), and converts it into a format that can be used in a Tkinter GUI. If something goes wrong (like a bad URL or download issue), it returns None instead of crashing.

**Weather\_data.py:** This code takes raw weather and forecast data and organizes it into a simple dictionary. It extracts details like temperature, humidity, weather description, icon code, and converts the sunrise and sunset times from timestamps into readable times (like "06:45 AM"). Then it bundles everything together—including the forecast—and returns it for use in other parts of the program.

Gui:

**Components.py:** This code creates and updates different parts of a weather app's user interface using **Tkinter**. It fills a weather panel with a city's name, weather icon, temperature, conditions, humidity, sunrise/sunset times, and displays icons for allergens like pollen and dust. It also includes two forecast views: one showing a **7-day forecast** using panels with icons and descriptions, and another **horizontal layout** for showing each day's weather in a row. It uses different background colors for each day depending on the weather description (like blue for rain or yellow for sunny). All icons are fetched from the internet and displayed using PIL and ImageTk.

**Main\_window.py:** This code builds the main part of a weather dashboard app using **Tkinter**. It lets users choose between viewing weather for **one city or two cities side by side**. When the user enters city names and clicks "View Weather," it fetches data from the OpenWeatherMap API, processes it, and displays weather info like temperature, conditions, sunrise/sunset, a forecast, and allergen icons. It also changes the background theme based on the weather. The interface adjusts automatically depending on the selected view mode (single or dual city). If there's an error (like a misspelled city), it shows a message to the user.

Themes.py: This code sets an animated background (GIF) in a Tkinter window based on the current weather description. It picks the right weather animation (like rain, snow, clouds, or clear) and plays it in a loop as the background. If it can't load the animation for any reason, it simply clears the background image.

**App.py:** This code starts the weather app. It imports the WeatherApp class from the gui.main\_window file, creates an instance of the app, and runs it. The if \_\_name\_\_ == "\_\_main\_\_": part makes sure the app only runs when the file is executed directly, not when it's imported into another file.