Table of Contents

Introduction:

This project is a web-based weather application that provides current weather data, forecasts, and historical search functionality. It uses an external weather API for data fetching and presents this information through a responsive UI, allowing users to search for weather data in various locations and view past searches. The goal is to create an intuitive and responsive weather dashboard that gives users a seamless experience across devices.

UI Structure:

HTML Structure and Nested div Elements:

The **index.html** file is the main file that organizes the structure and layout of the application. It contains various div elements to separate sections and to create a modular design. Here's an overview of how the HTML structure is nested:

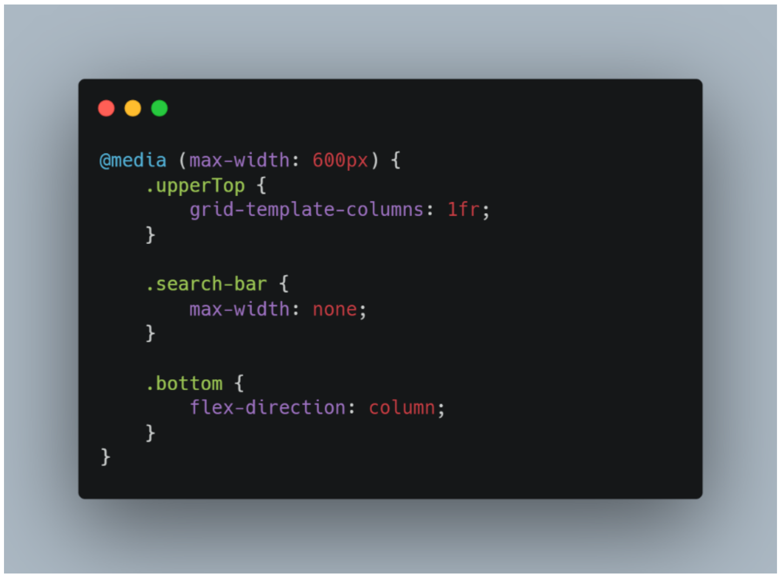
- **Main Container (<div class="container">)**
  - **Header Section (<div class="header">)**
    - Displays the application title and a brief description.
  - **Search Section (<div class="search-section">)**
    - Contains an input box for city names, allowing users to initiate searches.
  - **Current Weather Section (<div class="current-weather">)**
    - Displays the current weather details of the searched city.
  - **Forecast Section (<div class="forecast">)**
    - Displays a multi-day weather forecast for the searched city.
  - **Search History Section (<div class="search-history">)**
    - Lists previously searched cities, enabling users to revisit past searches quickly.

These sections are designed to be both modular and easy to modify, as each primary feature is wrapped in its div, keeping the code clean and organized.

Responsiveness:

The CSS file (style.css) handles responsiveness by adjusting the layout based on screen width. Key techniques used include:

- **Media Queries**: Media queries in the CSS file adjust the layout, font sizes, and margins for smaller devices (e.g., @media (max-width: 768px)).
- **Flexbox**: Flexbox is applied to several container elements, allowing the layout to adjust gracefully between desktop and mobile views.
- **Dynamic Widths**: Instead of fixed widths, percentages and vh/vw units are used, making the design adapt to various screen sizes.



```css
@media (max-width: 600px) {
    .upperTop {
        grid-template-columns: 1fr;
    }

    .search-bar {
        max-width: none;
    }

    .bottom {
        flex-direction: column;
    }
}
```

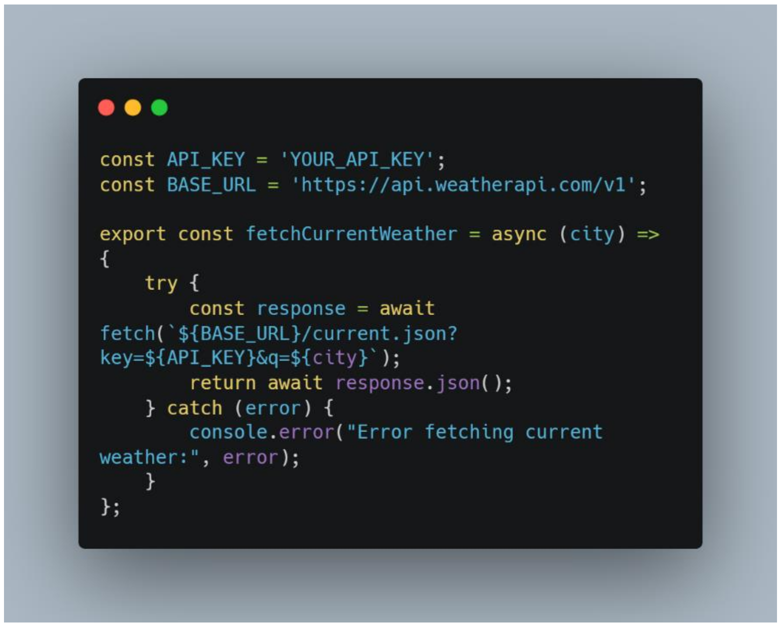
Functionality:

This project's functionality is organized into several JavaScript modules that interact to deliver core features.

Main Modules

Weather API Integration (api/weatherApi.js):

This module interacts with an external weather API to fetch data. It contains functions to retrieve both current weather and forecast data based on the user's input. The code snippet below demonstrates how the API is called:



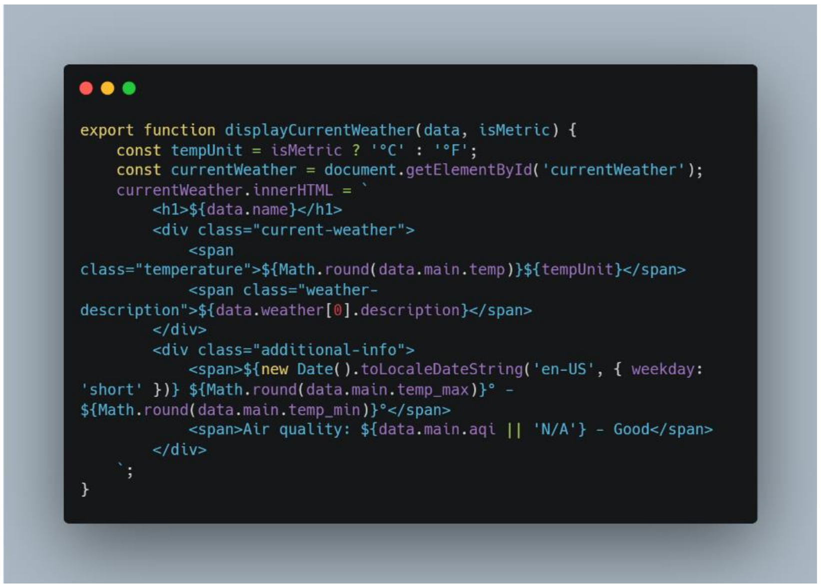
```
const API_KEY = 'YOUR_API_KEY';
const BASE_URL = 'https://api.weatherapi.com/v1';

export const fetchCurrentWeather = async (city) =>
{
    try {
        const response = await
fetch(`${BASE_URL}/current.json?
key=${API_KEY}&q=${city}`);
        return await response.json();
    } catch (error) {
        console.error("Error fetching current
weather:", error);
    }
};
```

The **fetchCurrentWeather** function takes a city name as input and returns the JSON response containing weather data.

Current Weather Display (components/currentWeather.js):

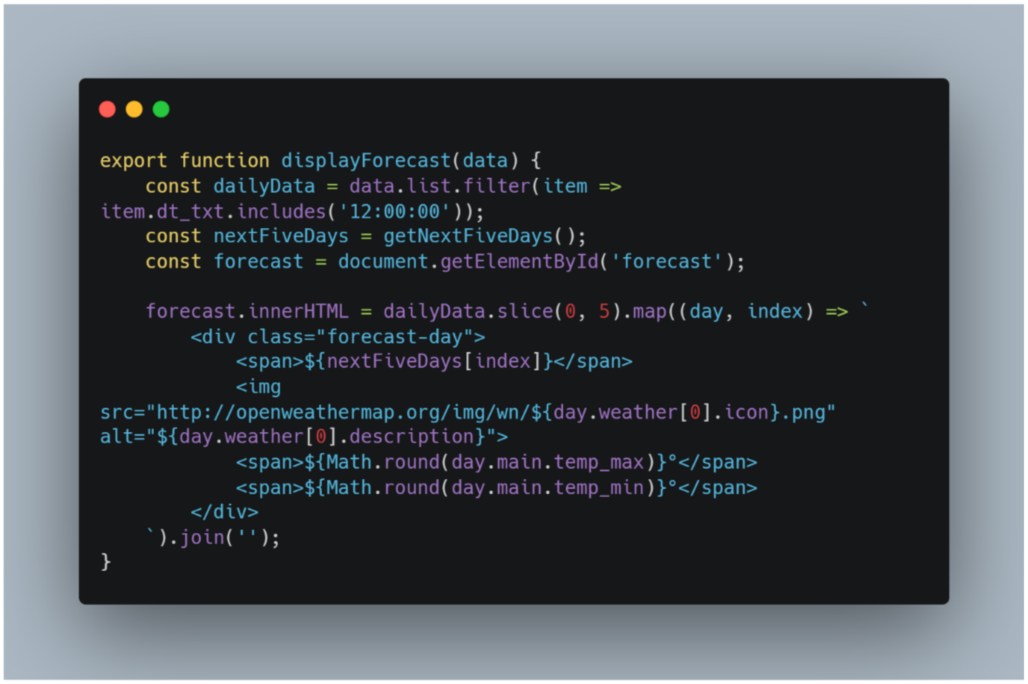
This module is responsible for displaying the current weather details. It fetches the data from the API and updates the UI with temperature, humidity, and wind speed.



```
export function displayCurrentWeather(data, isMetric) {
    const tempUnit = isMetric ? '°C' : '°F';
    const currentWeather = document.getElementById('currentWeather');
    currentWeather.innerHTML = `
        <h1>${data.name}</h1>
        <div class="current-weather">
            <span
class="temperature">${Math.round(data.main.temp)}${tempUnit}</span>
            <span class="weather-
description">${data.weather[0].description}</span>
        </div>
        <div class="additional-info">
            <span>${new Date().toLocaleDateString('en-US', { weekday:
'short' })} ${Math.round(data.main.temp_max)}° -
${Math.round(data.main.temp_min)}°</span>
            <span>Air quality: ${data.main.aqi || 'N/A'} - Good</span>
        </div>
    `;
}
```

Weather Forecast (components/forecast.js):

The **forecast.js** module handles the multi-day weather forecast display. It utilizes a separate API endpoint for forecast data and formats it for each day, displaying it within the forecast section.

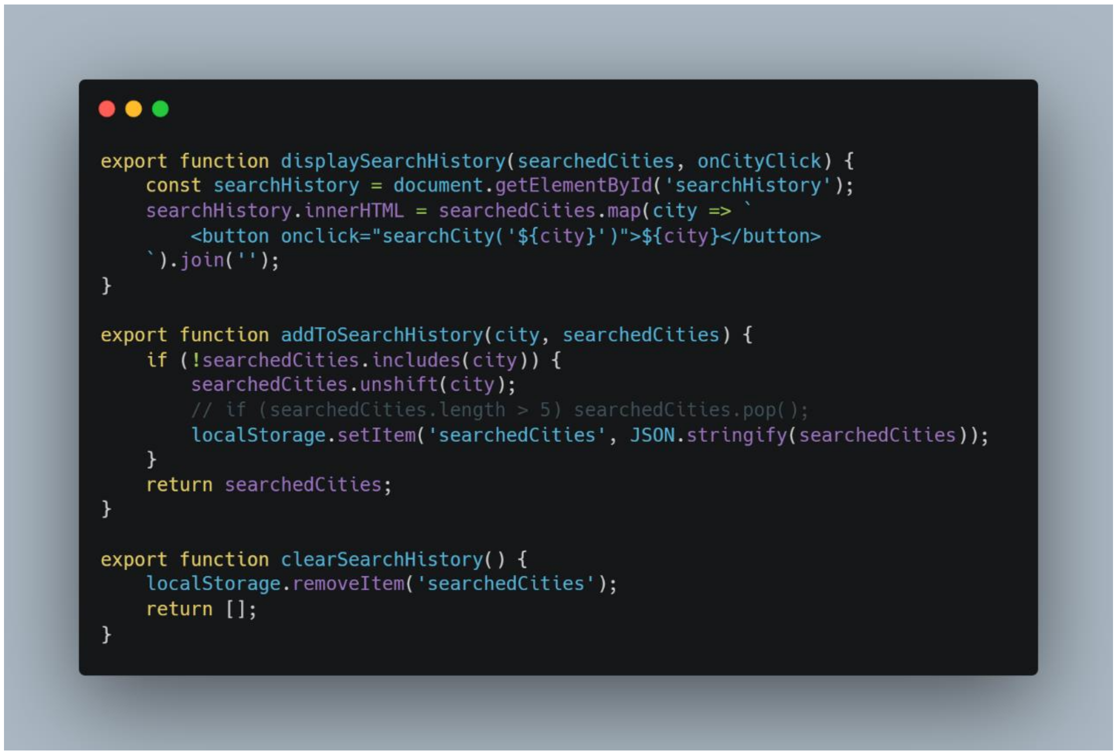


```
export function displayForecast(data) {
    const dailyData = data.list.filter(item =>
item.dt_txt.includes('12:00:00'));
    const nextFiveDays = getNextFiveDays();
    const forecast = document.getElementById('forecast');

    forecast.innerHTML = dailyData.slice(0, 5).map((day, index) => `
        <div class="forecast-day">
            <span>${nextFiveDays[index]}</span>
            <img
src="http://openweathermap.org/img/wn/${day.weather[0].icon}.png"
alt="${day.weather[0].description}">
            <span>${Math.round(day.main.temp_max)}°</span>
            <span>${Math.round(day.main.temp_min)}°</span>
        </div>
    `).join('');
}
```

Search History Management (components/searchHistory.js):

This module manages the search history, allowing users to re-select cities they've previously searched. It uses **local storage** to persist data across sessions.



```
export function displaySearchHistory(searchedCities, onCityClick) {
    const searchHistory = document.getElementById('searchHistory');
    searchHistory.innerHTML = searchedCities.map(city => `
        <button onclick="searchCity('${city}')">${city}</button>
    `).join('');
}

export function addToSearchHistory(city, searchedCities) {
    if (!searchedCities.includes(city)) {
        searchedCities.unshift(city);
        // if (searchedCities.length > 5) searchedCities.pop();
        localStorage.setItem('searchedCities', JSON.stringify(searchedCities));
    }
    return searchedCities;
}

export function clearSearchHistory() {
    localStorage.removeItem('searchedCities');
    return [];
}
```

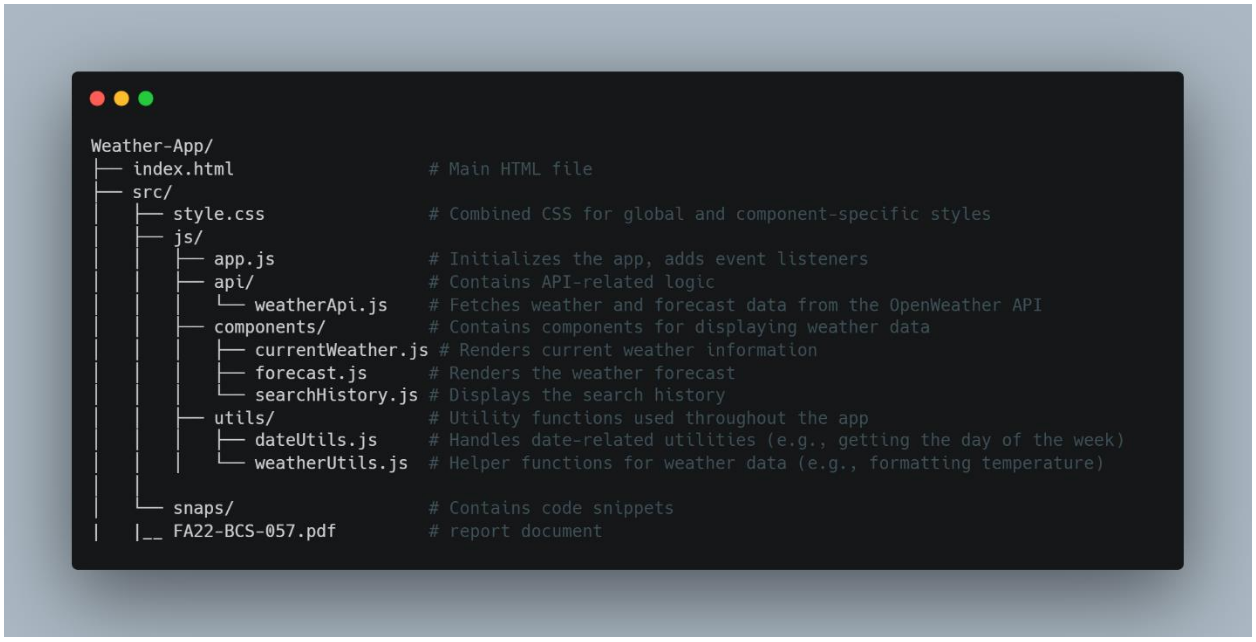
Utility Functions (utils/dateUtils.js & utils/weatherUtils.js):

The utility functions simplify date formatting and weather data processing, making the code more maintainable and organized.

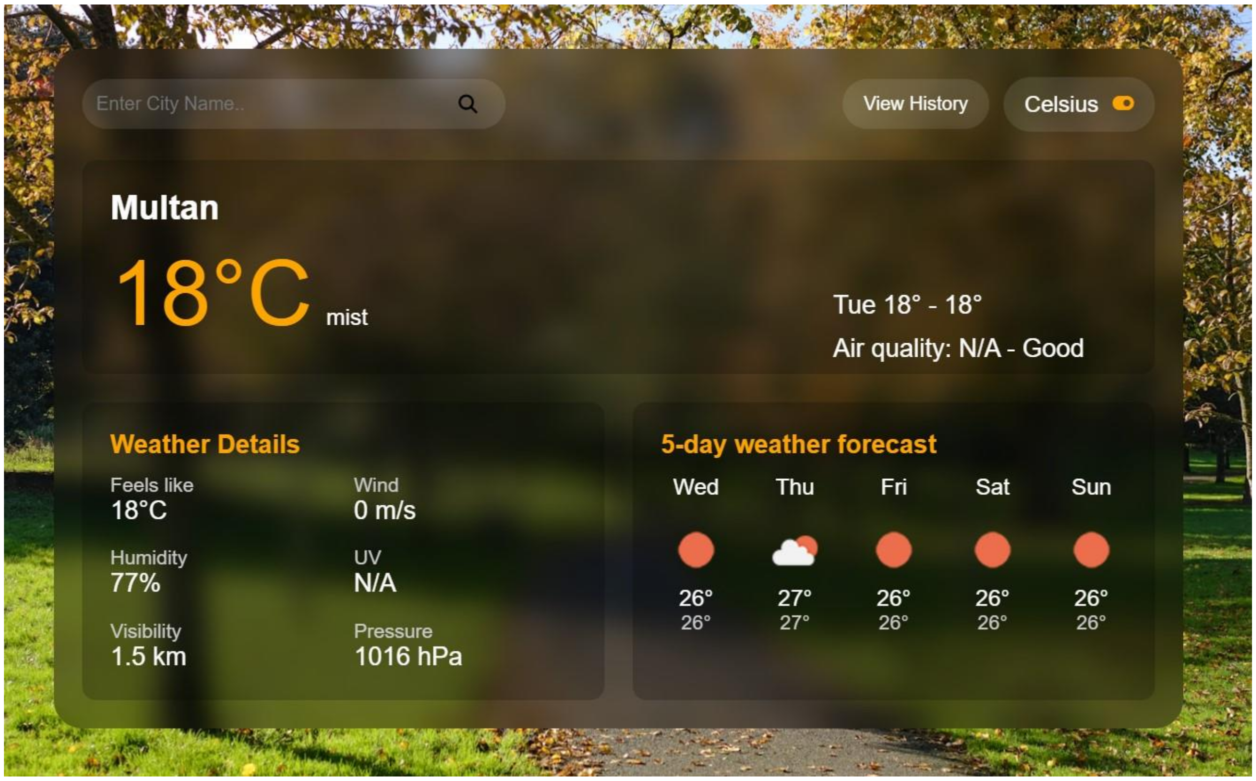
- **dateUtils.js**: Contains functions for formatting dates in a human-readable way.
- **weatherUtils.js**: Provides helper functions specific to weather data processing, such as converting temperature units.

References:

- ChatGPT
- UI Design
- Vercel



```
Weather-App/
├── index.html                # Main HTML file
├── src/
│   ├── style.css             # Combined CSS for global and component-specific styles
│   ├── js/
│   │   ├── app.js            # Initializes the app, adds event listeners
│   │   ├── api/              # Contains API-related logic
│   │   │   └── weatherApi.js # Fetches weather and forecast data from the OpenWeather API
│   │   ├── components/       # Contains components for displaying weather data
│   │   │   ├── currentWeather.js # Renders current weather information
│   │   │   ├── forecast.js   # Renders the weather forecast
│   │   │   └── searchHistory.js # Displays the search history
│   │   ├── utils/            # Utility functions used throughout the app
│   │   │   ├── dateUtils.js  # Handles date-related utilities (e.g., getting the day of the week)
│   │   │   └── weatherUtils.js # Helper functions for weather data (e.g., formatting temperature)
│   │
│   └── snaps/                # Contains code snippets
|   |__ FA22-BCS-057.pdf      # report document
```



Multan

18°C mist

Tue 18° - 18°

Air quality: N/A - Good

Weather Details

Feels like 18°C

Wind 0 m/s

Humidity 77%

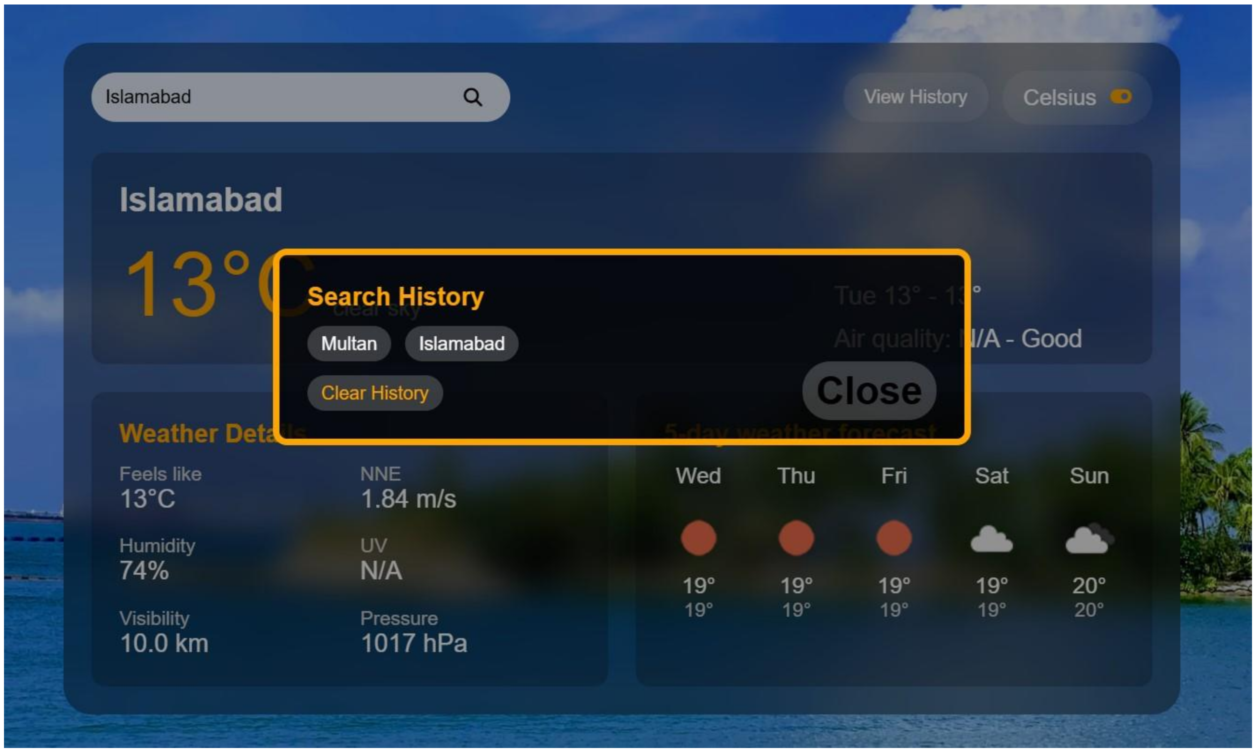
UV N/A

Visibility 1.5 km

Pressure 1016 hPa

5-day weather forecast

Wed	Thu	Fri	Sat	Sun
26°	27°	26°	26°	26°
26°	27°	26°	26°	26°



Islamabad

13°C clear sky

Tue 13° - 13°

Air quality: N/A - Good

Search History

Weather Details

Feels like 13°C

NNE 1.84 m/s

Humidity 74%

UV N/A

Visibility 10.0 km

Pressure 1017 hPa

5-day weather forecast

Wed	Thu	Fri	Sat	Sun
19°	19°	19°	19°	20°
19°	19°	19°	19°	20°