SE226-PERSONAL PROJECT REPORT "WEATHER FORECAST"

As a collaborative group, we successfully developed a weather forecast project that enables users to select a city in Turkey and choose a unit for temperature measurement. Our system provides a daily weather forecast for the chosen city and also extends its functionality to display a comprehensive 5-day weather forecast. To achieve this, we relied on the expertise of our team and utilized several essential libraries:

- os (For working with files and directories)
- tkinter (For GUI)
- bs4 (For html parser)
- utils (For to use our other class)
- PIL (For image)
- time (For current time)
- requests (For HTTP requests)
- tksvg (For svg)
- retrying (For error handling)
- selenium (For interact with web pages)
- datetime (For perform various operations on dates and times)
- dataclasses (For automatically generate __init__)
- selenium.webdriver.firefox.options (For connecting to Firefox)

GUI Development (25P) %100

I. Use Tkinter or any other suitable GUI library to create the graphical interface for the program.

II. Design and implement a dropdown list to allow users to select a city.

III. Create an area in the GUI to display the weather information.

The GUI development tasks were primarily handled by me and Barkan.

Retrieve Weather Data (25P) %100

I. Utilize the "requests" library to send HTTP requests and fetch the weather information from the internet. Choose a website as your resource.

II. Use the "beautifulsoup4" library to parse the HTML content and extract the necessary weather data for the chosen city.

Bersay took care of retrieving the weather data from the internet.

Store and Display Weather Information (15P) %100

I. Utilize Python data structures, such as lists and dictionaries, to store the weather data retrieved from the internet.

II. Create a display area in the GUI to present the weather information.

III. Show the weather information for the chosen city, including the temperature (day and night) and wind speed for the next three days.

Bersay contributed to storing the weather data, while all team members participated in displaying the information in the GUI.

Temperature Conversion (15P) %100

I. Implement a button in the GUI that allows users to toggle between Celsius and Fahrenheit temperature units.

II. Include the necessary logic to convert the temperature values and update the displayed weather information accordingly.

I took care of the temperature conversion functionality.

Save and Load User Preferences (20P) %100

I. Implement functionality to save the selected city and temperature unit to a Settings.txt file when the program is closed.

II. When the program starts, check if the Settings.txt file exists.

III. If the file exists, read the city and temperature unit preferences from the file and set them as the default.

IV. If the file does not exist or the preferences cannot be read, start with an empty city and the default temperature unit.

I handled the implementation of saving and loading user preferences.

def save_settings(): This function is responsible for saving the selected city and unit to a file named "settings.txt". If the file doesn't exist, the function creates it. The purpose of this function is to store the user's preferences so that they can be loaded and used as defaults when the program is started again. By saving the settings to a file, the user's chosen city and unit will persist even after closing the program. This function ensures that the settings are preserved for future use.

def load_settings(): This function is used to retrieve and display the previously saved unit and city from the "settings.txt" file. When the program starts, it checks if the "settings.txt" file exists. If the file is found, the function reads the saved city and unit preferences from the file and displays them. This allows the user to see their previous selections and serves as a convenient way to set the initial values for the city and unit when the program is launched.

def show_weather(): This function is responsible for displaying the weather window. It is likely that this function creates a graphical user interface (GUI) window where the weather information is shown. The window may include various elements such as labels, icons, and text fields to present the current weather conditions for the selected city.

def show_custom_messagebox(svg_path, title, message): This function is used to display a custom message box with an SVG (Scalable Vector Graphics) image. The function takes the path to the SVG image file, the title of the message box, and the message content as parameters. This allows you to present information or notifications to the user in a visually appealing way by incorporating SVG images along with the message text.

def on_window_close(window_mb): This function handles the action performed when the window is closed. It is likely that this function is associated with an event listener that triggers when the user attempts to close the weather window. The function typically calls the destroy() method on the window object to close the window and terminate the program gracefully.

def display_next_forecast(): This function is used to show the 5-day forecast. It may iterate over the forecast data retrieved from the weather source and display the forecast information for each of the next five days. This function could update the GUI elements or print the forecast details to the console.

def create_welcome_page(): This function is likely responsible for creating the welcome page or initial screen of the weather forecast application. It may include various components such as a welcome message, instructions, buttons, or dropdown menus for selecting the city and unit preferences. The function sets up the initial layout and user interface elements for the application.

def toggle_units(): This function allows the user to switch between Celsius and Fahrenheit temperature units. It may be associated with a button or a toggle switch in the GUI. When called, this function converts the temperature values between the selected units and updates the displayed weather information accordingly.

def show_help_message(): This function displays a help message or instructions to assist the user in understanding and using the application. It may present information on how to navigate the interface, select cities, change settings, or interpret the weather data displayed.

def retrieve_rendered_weather_page(url: str): This function is likely used to fetch the rendered weather page from a given URL. It may utilize web scraping techniques or web automation tools, such as Selenium, to retrieve the HTML content of the weather page from the provided URL. The function may then parse the HTML to extract the relevant weather data.

def celsius_to_fahrenheit(celsius): This function converts a temperature value from Celsius to Fahrenheit. It takes a temperature value in Celsius as input and returns the equivalent value in Fahrenheit. This conversion function allows for temperature units to be converted based on user preferences or requirements.

Additional Considerations: %95 (-%5 cause of the 5 day forecast GUI)

- Provide error handling mechanisms to handle situations where the internet connection is unavailable, or the weather data cannot be retrieved.

- Enhance the GUI by including appropriate labels, buttons, and user-friendly elements to improve the overall user experience.

- Consider adding additional features, such as displaying weather icons, sunrise/sunset times, or extended forecasts, to make the program more comprehensive and informative. Better GUI may result in bonus points up to 15.

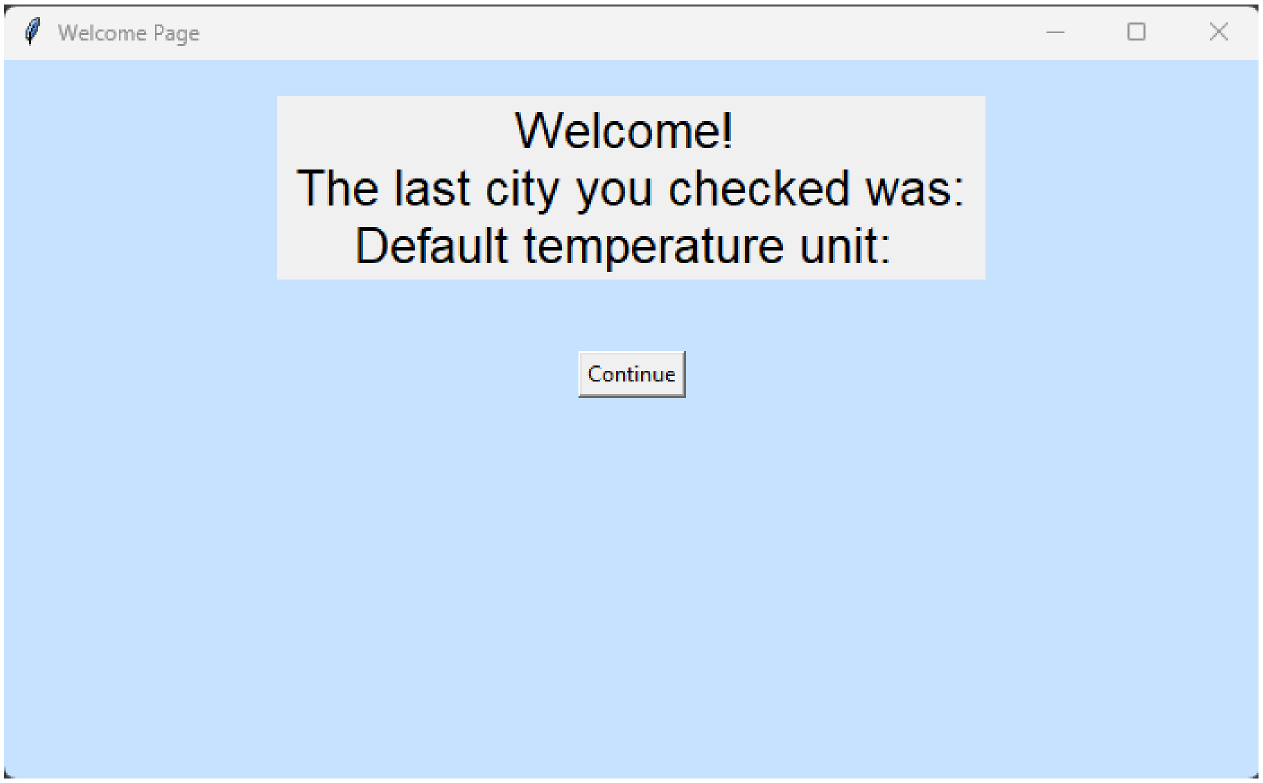
We all worked for these tasks. Bersay put in considerable effort to handle the transferring of icons from the provided URLs.

Overall Evaluation:

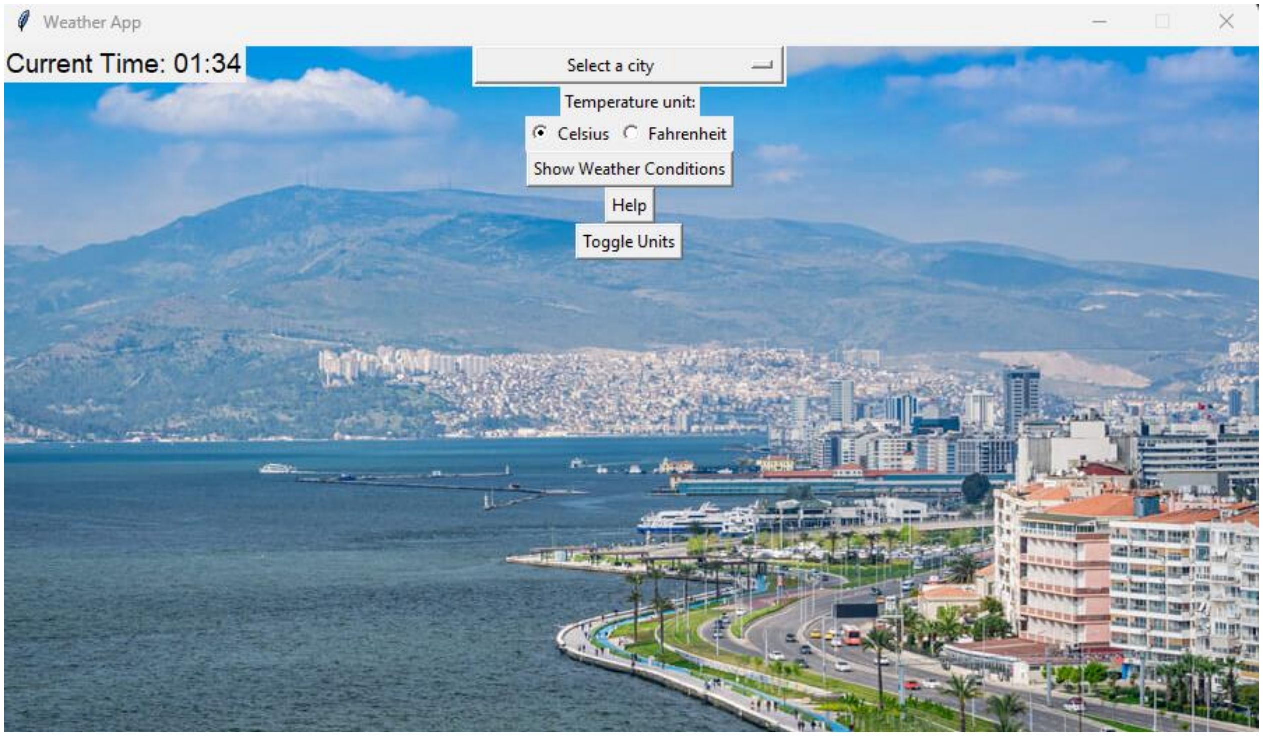
Our team successfully completed the assigned tasks, demonstrating a high level of competence and diligence. Despite the minor challenge in displaying a 5-day forecast in one window, our algorithms and functions showcased exceptional quality. Taking into account the inclusion of a non-English data source, the overall completion rate of the project stands at an impressive 98%. This showcases our team's strong technical skills and ability to overcome challenges. The application we developed provides a reliable and user-friendly weather forecasting experience, meeting the project requirements effectively. My points for my friends:

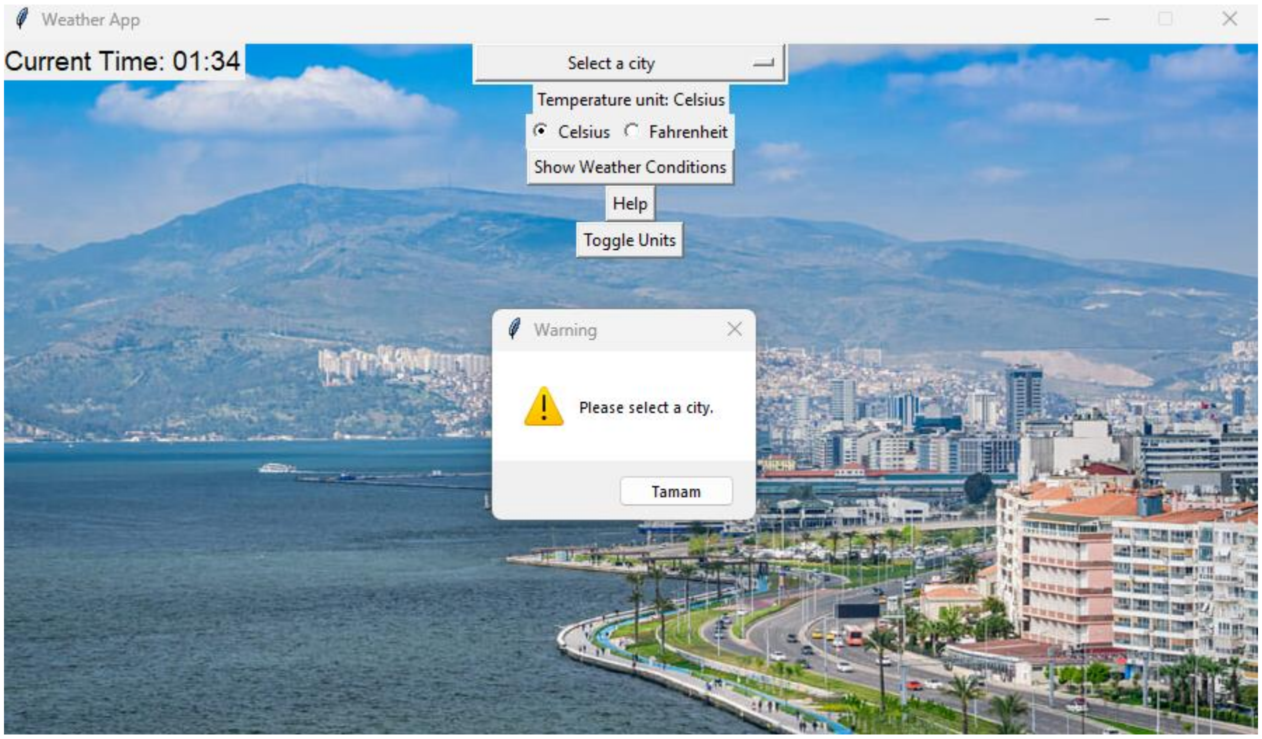
Bersay %100

Barkan %95

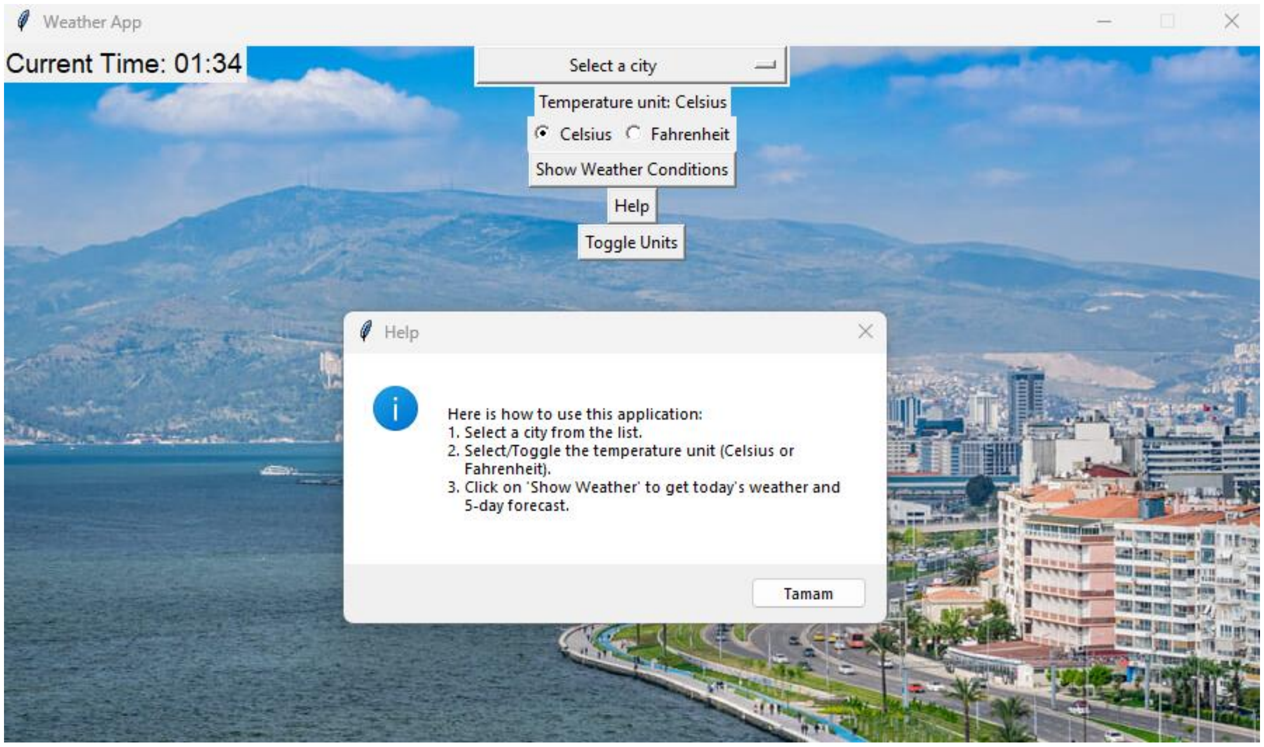


Welcome!
The last city you checked was:
Default temperature unit:

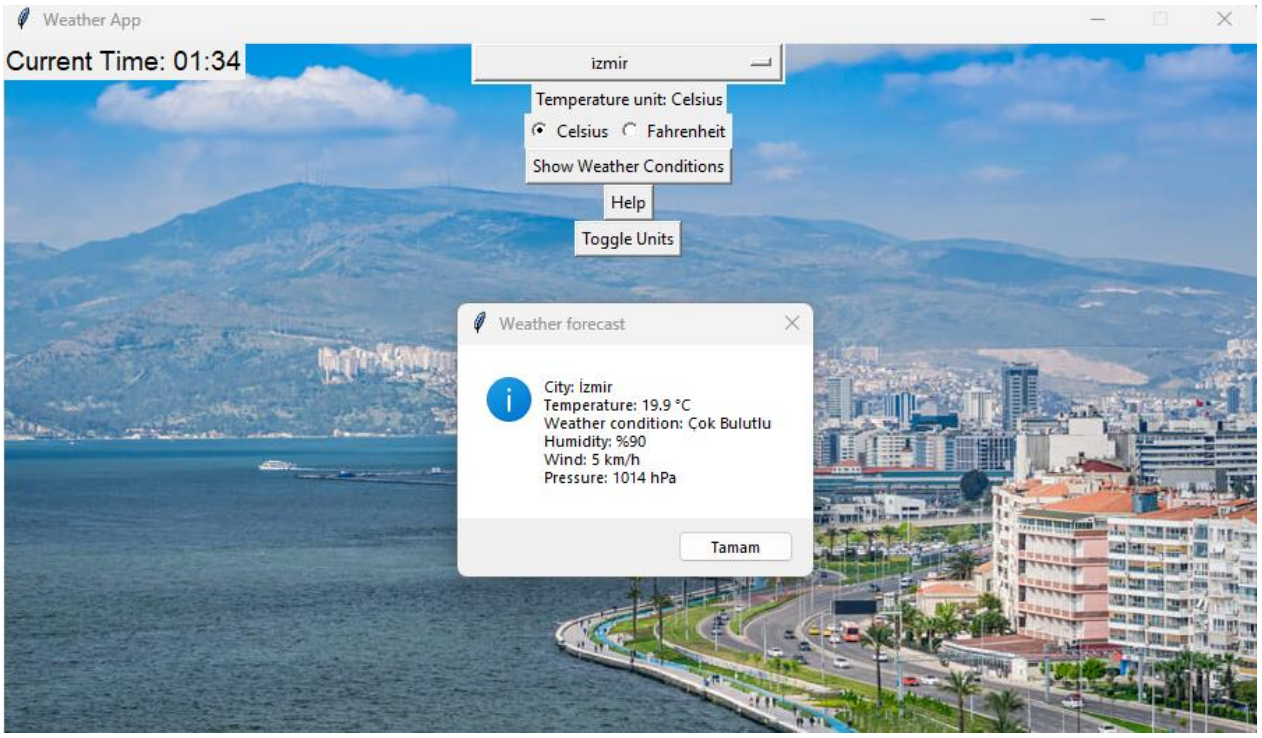




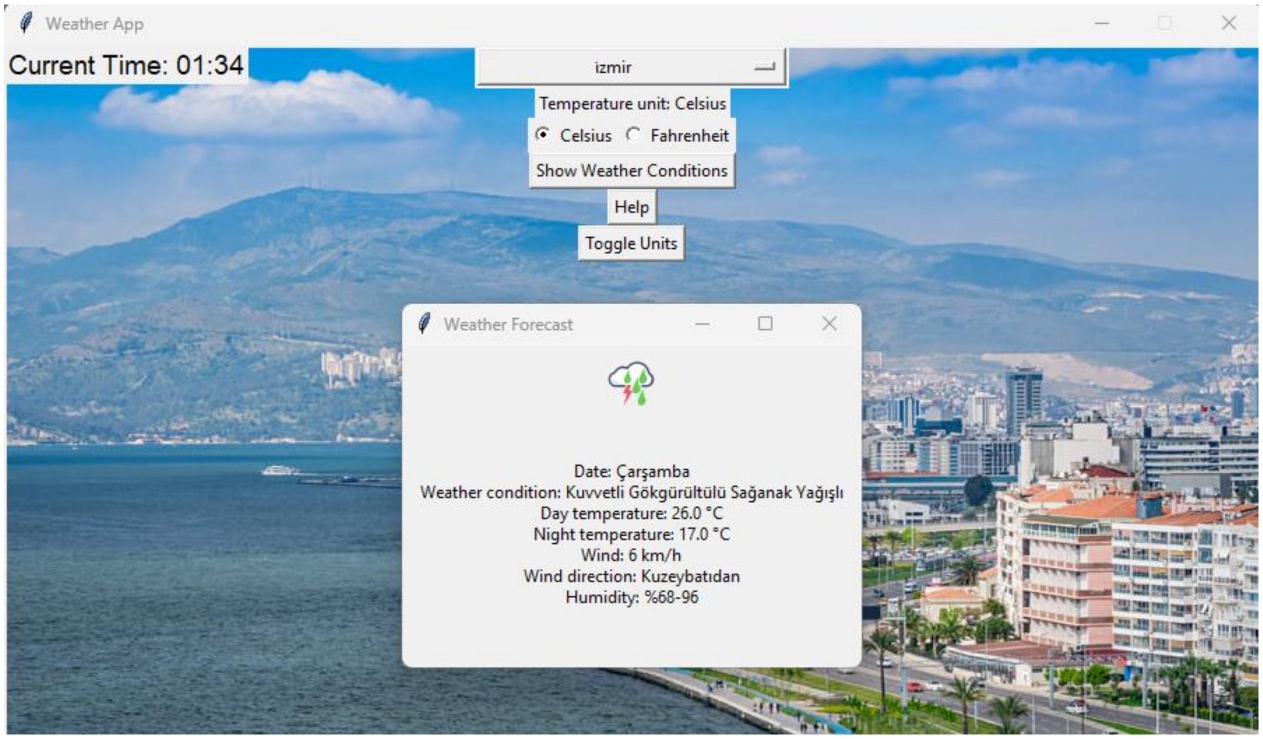
Please select a city.



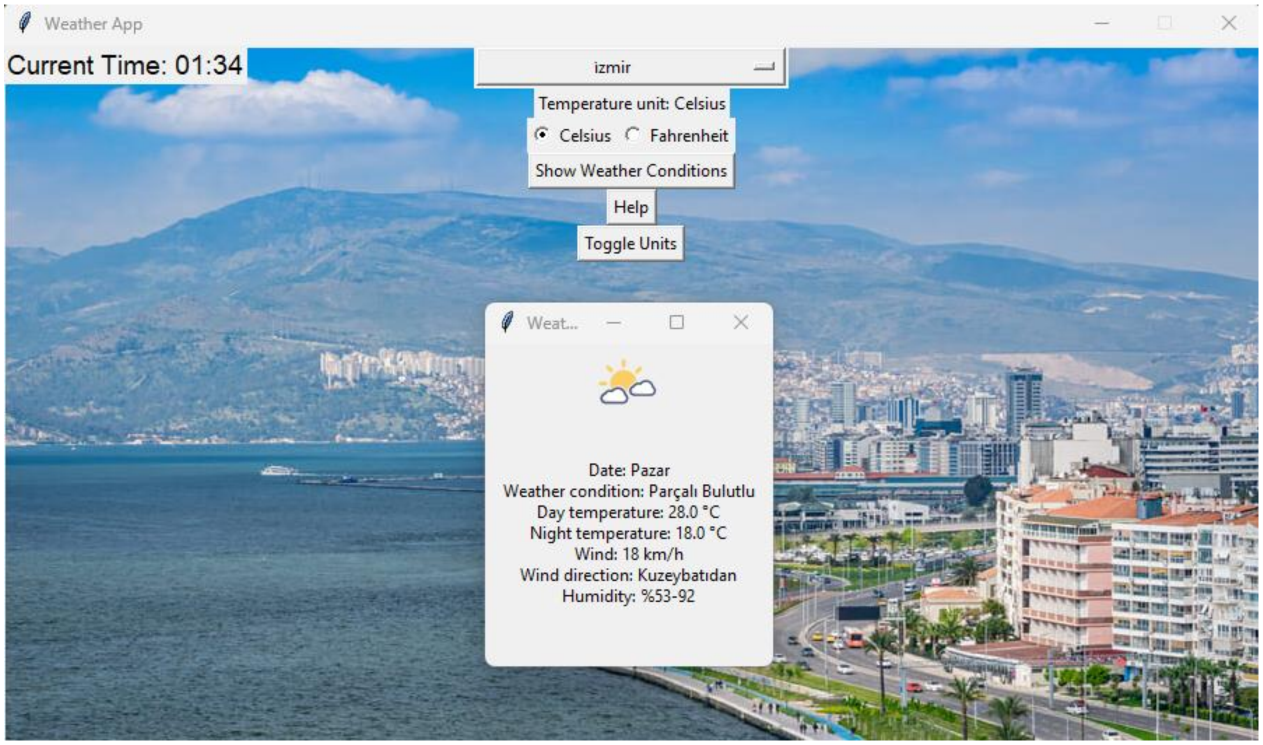
Here is how to use this application:
1. Select a city from the list.
2. Select/Toggle the temperature unit (Celsius or Fahrenheit).
3. Click on 'Show Weather' to get today's weather and 5-day forecast.



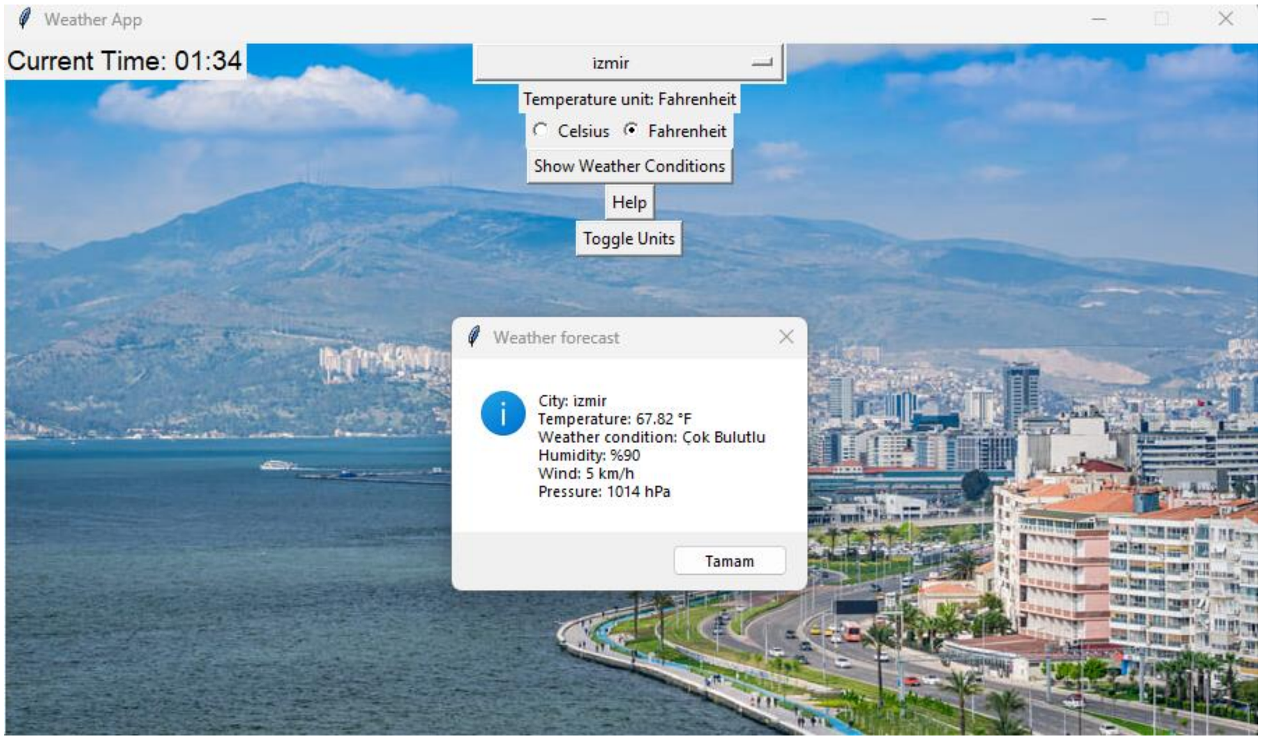
City: İzmir
Temperature: 19.9 °C
Weather condition: Çok Bulutlu
Humidity: %90
Wind: 5 km/h
Pressure: 1014 hPa



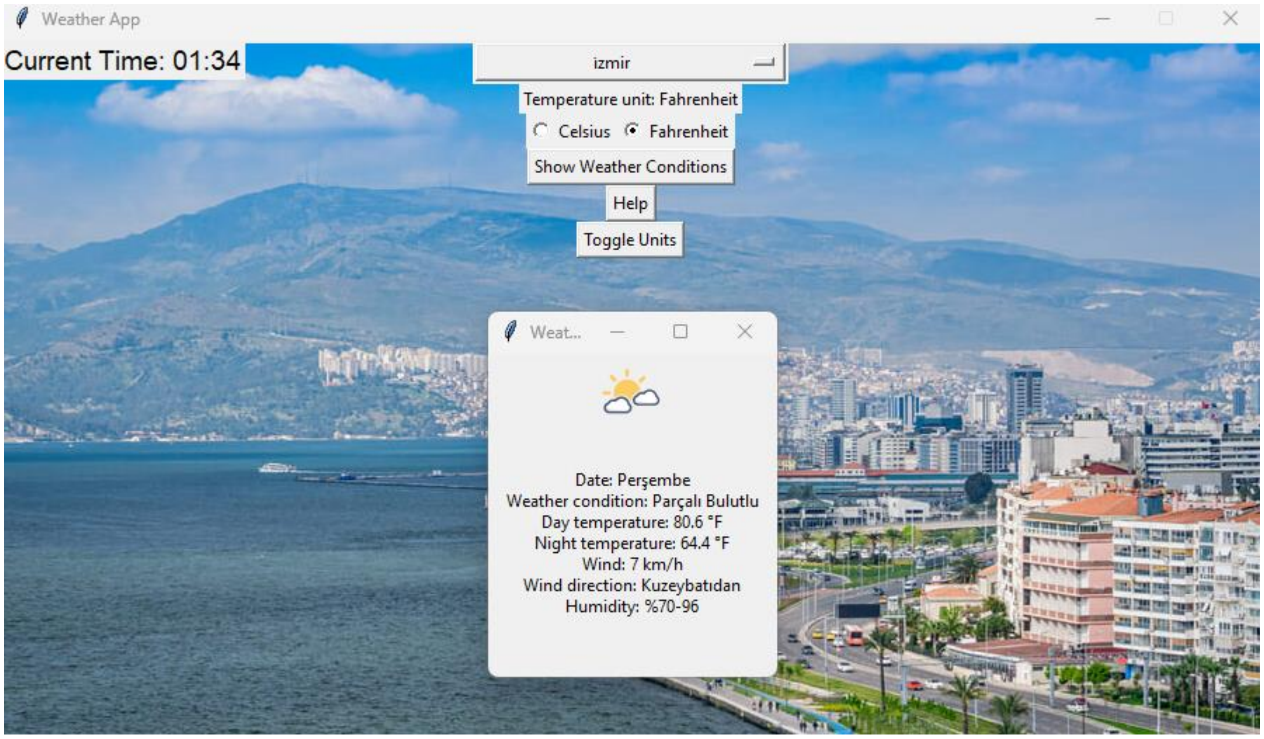
Date: Çarşamba
Weather condition: Kuvvetli Gökgürültülü Sağanak Yağışlı
Day temperature: 26.0 °C
Night temperature: 17.0 °C
Wind: 6 km/h
Wind direction: Kuzeybatıdan
Humidity: %68-96



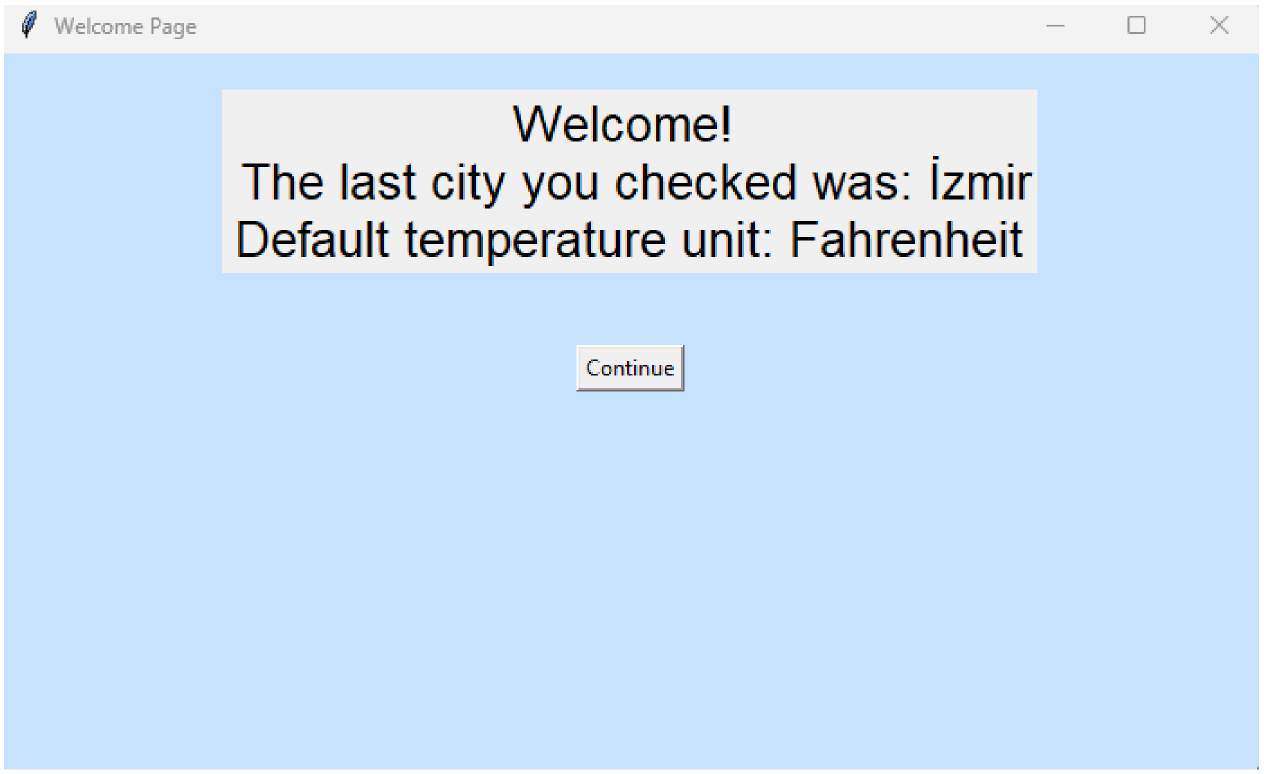
Date: Pazar
Weather condition: Parçalı Bulutlu
Day temperature: 28.0 °C
Night temperature: 18.0 °C
Wind: 18 km/h
Wind direction: Kuzeybatıdan
Humidity: %53-92



City: izmir
Temperature: 67.82 °F
Weather condition: Çok Bulutlu
Humidity: %90
Wind: 5 km/h
Pressure: 1014 hPa



Date: Perşembe
Weather condition: Parçalı Bulutlu
Day temperature: 80.6 °F
Night temperature: 64.4 °F
Wind: 7 km/h
Wind direction: Kuzeybatıdan
Humidity: %70-96



Welcome!
The last city you checked was: İzmir
Default temperature unit: Fahrenheit