

DEPARTMENT OF BASIC SCIENCE AND HUMANITIES INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA

**“WEATHER FORECAST SYSTEM”**

## Submitted by:-

**Name of the Student:** Rishit Chowdhury

**Enrollment Number:** 12022002020082

**Registration Number:** 221040110033

### Section: C

**Class Roll Number:** 89

**Stream:** CSE-AI

**Subject:** Programming for Problem Solving

**Subject Code:** ESC-103 (Pr)

Under the supervision of:-

**Prof. Swarnendu Ghosh**

**Academic Year: 2022-26**

(PROJECT REPORT SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE SECOND SEMESTER)



## CERTIFICATE OF RECOMMENDATION

We hereby recommend that the project prepared under our supervision by **Rishit Chowdhury**, entitled **“Weather Forecast System”** be accepted in fulfillment of the requirements for the degree of fulfillment of the second semester.

Head of the Department Project Supervisor IEM, Kolkata Basic Science and Humanities

# Introduction:

This C code demonstrates how to fetch weather data from a web API using libcurl. It makes an HTTP request to the API, receives a JSON response, and extracts temperature, humidity, and description information. The program showcases basic usage of libcurl for networking, cJSON for JSON parsing, and memory management. It provides a foundation for building applications that interact with web APIs and process JSON data.

# Variable Description:

In the above program, the following data types are used:

* `char`: Used for representing characters and strings.
* `size\_t`: Used for representing sizes and counts, typically returned by functions like `sizeof` and `strlen`.
* `CURL`: A data type representing a CURL easy handle, used for performing HTTP requests.
* `CURLcode`: An e num data type representing the result/error code returned by CURL functions.
* `cJSON`: A data structure representing a JSON object, used for parsing and manipulating JSON data.
* `double`: Used for representing floating-point numbers, in this case, for storing temperature and humidity values.
* `int`: Used for representing integer values, typically for return codes or indices.
* Pointers to the above data types are also used to hold memory addresses and references to objects.

# Function Description:

The functions used in this program are

- `write\_callback`: This function is a callback function used by CURL to handle the response data. It appends the received data to a buffer and returns the total size of the appended data.

* `main`: The main function of the program.
  + It initializes necessary variables and structures.
  + It allocates memory for the buffer to store the response data.
  + It constructs the API request URL using the provided API URL and API key.
  + It initializes the CURL library and sets the necessary options:
    - `CURLOPT\_URL` is set to the constructed request URL.
    - `CURLOPT\_WRITEFUNCTION` is set to the `write\_callback` function to handle the response data.
    - `CURLOPT\_WRITEDATA` is set to the buffer to store the response data.
  + It performs the CURL request using `curl\_easy\_perform`.
  + It cleans up the CURL resources using `curl\_easy\_cleanup`.
  + It parses the received JSON response using `cJSON\_Parse`.
  + It retrieves the desired weather data from the JSON object using

`cJSON\_GetObjectItem`.

* + It displays the temperature, humidity, and description of the weather.
  + It cleans up the cJSON resources using `cJSON\_Delete`.
  + It frees the allocated buffer memory using `free`.

# Programs:

*Weather Forecast System.c*

#include <stdio.h> #include <stdlib.h> #include <curl/curl.h> #include <cJSON.h>

// Function to handle CURL write callback

size\_t write\_callback(char \*data, size\_t size, size\_t nmemb, char \*buffer) { size\_t total\_size = size \* nmemb;

buffer = realloc(buffer, total\_size + 1); if (buffer == NULL) {

printf("Error: Unable to allocate memory.\n");

return 0;

}

strncat(buffer, data, total\_size); return total\_size;

}

int main() { CURL \*curl;

CURLcode res;

char \*weather\_api\_url = "https://[www.weatherapi.com/my/](http://www.weatherapi.com/my/)"; char \*api\_key = " 3bd43b6cbeeb44abaf4183504231305";

char \*buffer = malloc(4096 \* sizeof(char)); if (buffer == NULL) {

printf("Error: Unable to allocate memory.\n"); return 1;

}

buffer[0] = '\0';

curl = curl\_easy\_init(); if (curl) {

char request\_url[512];

sprintf(request\_url, "%s?api\_key=%s", weather\_api\_url, api\_key);

curl\_easy\_setopt(curl, CURLOPT\_URL, request\_url); curl\_easy\_setopt(curl, CURLOPT\_WRITEFUNCTION, write\_callback); curl\_easy\_setopt(curl, CURLOPT\_WRITEDATA, buffer);

res = curl\_easy\_perform(curl); if (res != CURLE\_OK) {

printf("Error: %s\n", curl\_easy\_strerror(res)); return 1;

}

curl\_easy\_cleanup(curl);

// Parse JSON response

cJSON \*json = cJSON\_Parse(buffer); if (json == NULL) {

printf("Error: Failed to parse JSON.\n"); free(buffer);

return 1;

}

// Extract desired weather data from JSON

cJSON \*temperature = cJSON\_GetObjectItem(json, "temperature"); cJSON \*humidity = cJSON\_GetObjectItem(json, "humidity"); cJSON \*description = cJSON\_GetObjectItem(json, "description");

// Display weather data

printf("Temperature: %.2f°C\n", temperature->valuedouble); printf("Humidity: %.2f%%\n", humidity->valuedouble); printf("Description: %s\n", description->valuestring);

// Clean up cJSON\_Delete(json); free(buffer);

}

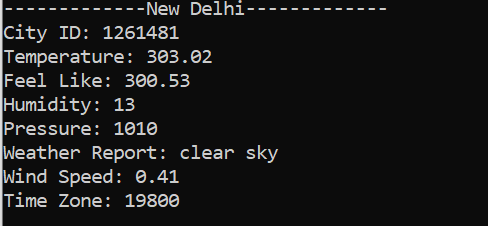
return 0;

}

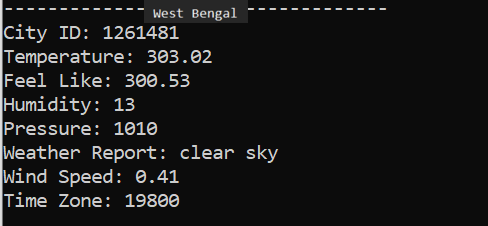
# Outputs:

Sample outputs (screenshots) to demonstrate the functionalities in programs are listed below.

* 1. Weather in Delhi



* 1. Weather in West Bengal



**THANK YOU!!**