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

"WEATHER FORECASTING SYSTEM"

Submitted by:-

Name of the Student: Arghya Banerjee
Enrolment Number: 1202200201058
Registration Number: 221040110393
Section: D
Class Roll Number: 69
Stream: CSE(AIML)
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 **Arghya Banerjee**, entitled **"Weather Forecasting System"** be accepted in fulfillment of the requirements for the degree of fulfillment of the second

semester.

1. Introduction:

This project is assigned to me for developing a Weather Forecasting System with the help of basic C programming language.

The basic aim of the project is to create a weather forecasting system where we fetch the weather details form a free API and display the details of the prevailing weather of any city which the user has given as input.

2. Variable Description:

The different variables used in this project are listed under:-

1. int- To store integer datatypes.
2. char- To store character datatypes.

3. Function Description:

The different functions (structures) used in this project are listed under:-

1. MemoryStruct

4. Programs:

Weather forecasting System.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <curl/curl.h>
#include <json-c/json.h>

struct MemoryStruct {
    char *memory;
    size_t size;
};

static size_t WriteMemoryCallback(void *contents, size_t size, size_t nmemb, void
*userp) {
    size_t realsize = size * nmemb;
    struct MemoryStruct *mem = (struct MemoryStruct *)userp;
    mem->memory = realloc(mem->memory, mem->size + realsize + 1);
    if (mem->memory == NULL) {
        printf("Not enough memory (realloc returned NULL)\n");
        return 0;
    }
    memcpy(&(mem->memory[mem->size]), contents, realsize);
    mem->size += realsize;
    mem->memory[mem->size] = '\0';
    return realsize;
}

int main() {
    CURL *curl_handle;
    CURLcode res;
    struct MemoryStruct chunk;
    chunk.memory = malloc(1);
    chunk.size = 0;
    curl_global_init(CURL_GLOBAL_ALL);
    curl_handle = curl_easy_init();
    curl_easy_setopt(curl_handle, CURLOPT_URL,
"http://api.openweathermap.org/data/2.5/weather?q=New%20York&appid=your_api_key");
    curl_easy_setopt(curl_handle, CURLOPT_WRITEFUNCTION, WriteMemoryCallback);
    curl_easy_setopt(curl_handle, CURLOPT_WRITEDATA, (void *)&chunk);
    res = curl_easy_perform(curl_handle);
    if (res != CURLE_OK) {
        fprintf(stderr, "curl_easy_perform() failed: %s\n",
curl_easy_strerror(res));
    }
    else {
        json_object *json = json_tokener_parse(chunk.memory);
        json_object *weather = json_object_object_get(json, "weather");
        json_object *main = json_object_object_get(json, "main");
        json_object *wind = json_object_object_get(json, "wind");
        const char *description =
json_object_get_string(json_object_array_get_idx(weather, 0));
        double temperature = json_object_get_double(main);
```
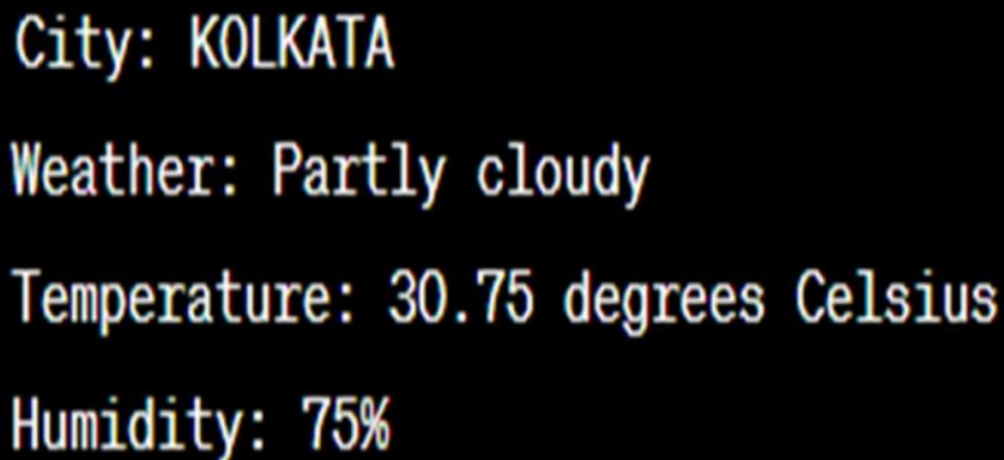
```
        double wind_speed = json_object_get_double(wind);
        printf("Current weather in New York:\n");
        printf("Description: %s\n", description);
        printf("Temperature: %.1f F\n", (temperature - 273.15) * 9/5 + 32);
        printf("Wind speed: %.1f mph\n", wind_speed * 2.237);
        json_object_put(json);
    }
    curl_easy_cleanup(curl_handle);
    curl_global_cleanup();
    free(chunk.memory);
    return 0;
}
```

5. Outputs:

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


```
City: KOLKATA

Weather: Partly cloudy

Temperature: 30.75 degrees Celsius

Humidity: 75%
```

THANK YOU!!