Example4

Essential Packages on Ubuntu

1. apt-get update

```
apt-get update
```

2. apt-get install -y sudo vim wget unzip g++ cmake curl pkg-config libssl-dev libsasl2-dev git python3 nano

```
apt-get install -y sudo vim wget unzip g++ cmake curl pkg-config libssl-dev
libsasl2-dev git python3 nano
```

3. mkdir clientcpp

```
mkdir clientcpp
```

4. cd clientcpp

```
cd clientcpp
```

Extract Crow Framework

5. wget https://github.com/CrowCpp/Crow/releases/download/v1.0%2B5/crow-v1.0+5.tar.gz

```
wget https://github.com/CrowCpp/Crow/releases/download/v1.0%2B5/crow-v1.0+5.tar.gz
```

6. mkdir crow

```
mkdir crow
```

7. tar xvfz crow-v1.0+5.tar.gz -C crow --strip-components=1

```
tar xvfz crow-v1.0+5.tar.gz -C crow --strip-components=1
```

Extract Boost Libraries

8. wget https://boostorg.jfrog.io/artifactory/main/release/1.83.0/source/boost_1_83_0.tar.gz

```
wget https://boostorg.jfrog.io/artifactory/main/release/1.83.0/source/boost_1_83_0.tar.gz
```

9. tar -xzvf boost_1_83_0.tar.gz

```
tar -xzvf boost_1_83_0.tar.gz
```

Extract cpr library and build

10. git clone https://github.com/libcpr/cpr.git

```
git clone https://github.com/libcpr/cpr.git
```

11. cd CPR && mkdir build && cd build

```
cd cpr && mkdir build && cd build
```

12. cmake .. -DCPR_USE_SYSTEM_CURL=ON

```
cmake .. -DCPR_USE_SYSTEM_CURL=ON
```

13. cmake --build . --parallel

```
cmake --build . --parallel
```

14. sudo cmake --install .

```
sudo cmake --install .
```

15. nano main.cpp

```
nano main.cpp
```

```cpp
#include "crow.h"
#include <cpr/cpr.h>

int main() {
    crow::SimpleApp app;

    CROW_ROUTE(app, "/send_request")
    ([]{
        // Using CPR to send an HTTP GET request
        cpr::Response r = cpr::Get(cpr::Url{"http://localhost:8080"});

        if (r.status_code == 200) {  // HTTP status code: OK
            return crow::response(200, r.text);
        } else {
            return crow::response(500, "Failed to get data");
        }
    });

    app.port(8081).multithreaded().run();
}
```

16. nano CMakeLists.txt

```
nano CMakeLists.txt
```

```cmake
cmake_minimum_required(VERSION 3.15)
project(clientcpp)

# Define the include directories
set(INCLUDE_PATHS ./boost_1_83_0 ./crow/include)

# Add the executable target
add_executable(clientcpp main.cpp)

# Include the defined paths
target_include_directories(clientcpp PUBLIC ${INCLUDE_PATHS})

# Specify the C++ standard
set_target_properties(clientcpp PROPERTIES
    CXX_STANDARD 17
    CXX_STANDARD_REQUIRED TRUE
)
```

```
find_package(cpr REQUIRED)
target_link_libraries(clientcpp PRIVATE cpr::cpr)
```

Build the Application

17. mkdir build

```
mkdir build
```

18. cd build

```
cd build
```

19. cmake ..

```
cmake ..
```

20. make

```
make
```

Run the Application

21. ./clientcpp

```
./clientcpp &
```

Once executed, it will start a web server on a specified port 8081 & run simplecpp application for testing purpose

17. curl http://localhost:8081/send_request

```
curl http://localhost:8080/send_request
```

Excpected Output

```
HelloWorld
```