

سینا کریمی 97105509

توضیحات مربوط به سوال سوم:

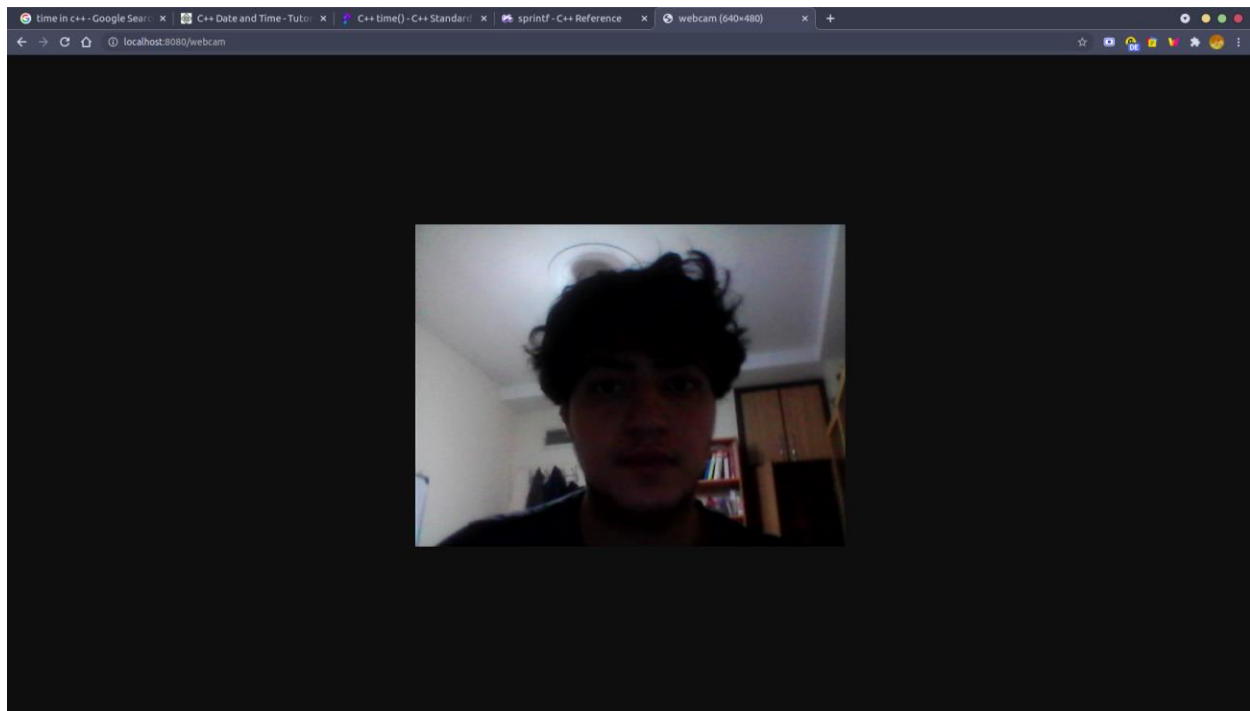
در این سوال برای درست کردن وب سرور از فایل آماده httpServerFast استفاده شده است.

ابتدا نتایج را بررسی کنیم:

ابتدا سرور را به شکل زیر ران میکنیم:

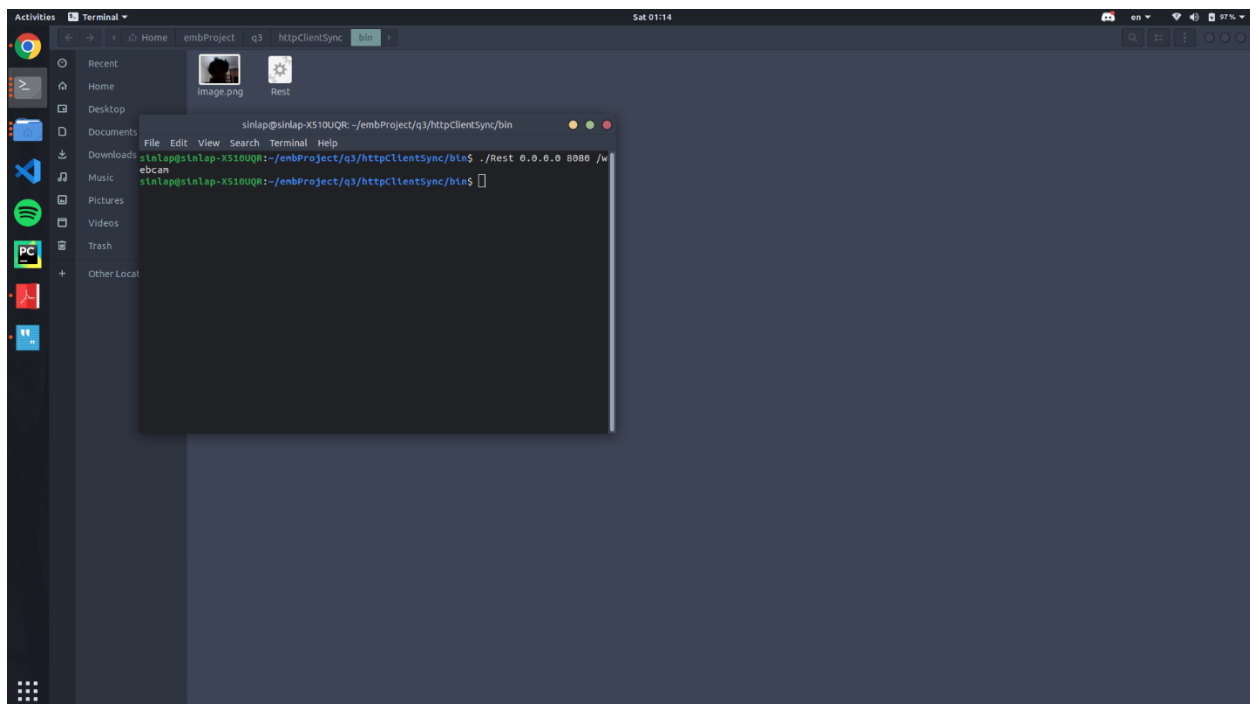
```
sinlap@sinlap-X510UQR: ~/embProject/q3/httpServerFast/bin
File Edit View Search Terminal Help
sinlap@sinlap-X510UQR:~/embProject/q3/httpServerFast/bin$ ./restServer
Usage: http_server_fast <address> <port> <doc_root> <num_workers> {spin|block}
For IPV4, try:
    http_server_fast 0.0.0.0 80 . 100 block
For IPV6, try:
    http_server_fast 0:::0 80 . 100 block
sinlap@sinlap-X510UQR:~/embProject/q3/httpServerFast/bin$ ./restServer 0.0.0.0 8
080 . 100 block
█
```

و سپس زمانی که از بروزر میخواهیم وارد سایت شویم و درخواست webcam / را بدهیم، با صفحه مقابل مواجه میشویم:



که دقیقا چیزی است که نیاز داریم.

به همراه این برنامه، کد مربوط به client نیز درست شده است که تصویر را گرفته و به صورت png در محل خود برنامه ذخیره میکند. نمونه ران کردن برنامه client :



توضیحات مربوط به کد:

در کد از کتابخانه boost 1.76 و کتابخانه OpenCV استفاده شده است. قسمتی که در برنامه httpServerFast تغییر داده شده این قسمت میباشد:

```
void process_request(http::request<request_body_t, http::basic_fields<alloc_t>> const& req)
{
    switch (req.method())
    {
    case http::verb::get:
        std::cout << req.target() << "\n";
        if (req.target() == "/webcam" ){

            cv::VideoCapture camera(0);
            cv::Mat frame;
            camera >> frame;
            cv::imwrite("image.png", frame);
            send_file("/image.png");

        }
        else {
            send_file(req.target());
        }
        break;
    }
```

در این قسمت گفته شده است که اگر درخواست /webcam بود، تصویر وبکم گرفته میشود و به صورت یک فایل image.png در می آید و از تابع send\_file خود برنامه استفاده میشود تا دستور فرستاده شود. برای برنامه client نیز فقط این قسمت اضافه شده است:

```
ofstream myfile ("image.png");
if (myfile.is_open())
{
    myfile << res.body();
    myfile.close();
}
else cout << "Unable to open file";
// res.get().body().open(local_filename, boost::
```

پیام را به جای اینکه به صورت dynamic\_body بگیریم به صورت string\_body میگیریم تا بتوانیم آن را به هر صورت که میخواهیم ذخیره کنیم.