Human Machine Interface Monitoring System

User Manual

Document Control

- **Document Version:** 1.0
- **Date of Issue:** 12-03-2024
- **Authors:** Francesco Carbone, Khosro Pourkavoos, Rajani Bhat, Omar Abouelgheit
- **Document Classification:** Public

Table of Contents

1. Introduction

1.1 Purpose of the Document

To facilitate and guide the deployment, installation, and operation of this software to monitor industrial Human Machine Interfaces, HMIs.

1.2 Intended Audience

Flexematic Inc: currently provides intelligent HMIs to industrial customers, future developers working on this project, and product owners/clients such as Dr. Wang this semester.

1.3 Document Scope

Topics covered in this document include system requirements and compatibility, hardware requirements, software installation and configuration, data acquisition, and user interface navigation.

2. System Overview

2.1 Product Description

This software allows users to connect to a vnc server installed on the human machine interface of a piece of industrial equipment, to periodically take a screenshot, to process the screenshot , and to extract, log, and display data.

2.2 Key Features

Small memory and CPU usage footprint, intuitive and easy to navigate user interface. Extensive use of well-tested and industry-standard python libraries, including python AI modules for OCR and Resnet18 image recognition.

2.3 System Architecture

64-bit architecture

2.4 Compatibility and Requirements

Compatible with Linux and Windows 11 OS. Not tested on Raspberry Pi, but the Raspbian OS otherwise meets requirements.

3. Installation and Deployment

3.1 Pre-Installation Requirements

100 MB for disk space if image-archiving is enabled; otherwise, 50 MB should suffice.

3.1.1 Hardware Specifications

Processing images via Resnet18 neural network would optimally function on an AMD Ryzen7 CPU, an Intel Core i7 CPU, or better.

3.1.2 Software Prerequisites

A Python3 interpreter is essential for all functionalities. The processing of images via the Resnet18 neural network would optimally function on an AMD Ryzen 7 CPU or an Intel Core i7 CPU or better. The speed has not been tested on a Raspberry Pi yet, but the Raspbian meets all requirements.

Required Python3 libraries as specified in requirements.txt.

Browser: Firefox or similar HTML5, CSS, and JavaScript-enabled browser

Gimp: open-source image manipulation software or similar

3.1.3 Network Configuration

The device requires a wired or wireless network interface to connect to the VNC server via LAN or Wi-Fi.

3.2 Installation Procedure

3.2.1 Step-by-Step Installation Guide

Ping the VNC server on the industrial equipment to verify connection.

The software directory structure is as follows:

```
└── hmi
    ├── HMI_Webpage
    ├── vnc
    │   ├── m000
    │   │   ├── alerts
    │   │   ├── archive
    │   │   │   ├── aai
    │   │   │   ├── images
    │   │   │   └── sqlite3
    │   │   │       └── spc
    │   │   ├── config
    │   │   │   └── semiauto_segmentation
    │   │   ├── display
    │   │   ├── docs
    │   │   ├── logs
    │   │   ├── pys
    │   │   ├── ref_images
    │   │   ├── ss
    │   │   └── tmp
    └── webcam
```

DIRECTORIES:

HMI_Webpage: location of hmi.html, that contains the most recent data extracted

vnc: for all images and scripts that process screenshots from the vnc server

m000: industrial machine 000

m001: industrial machine 001

alerts: directory containing alerts_log.txt

archive: contains archived data, including the ofkrse.db database, and spc, scripts for statistical process control

config: configuration files such as the segmentation table, and min-max table.

docs: documentation

logs: log files (in addition to sqlite3 database entries

pys: python scripts

ss: screenshot, that serves as the starting point for data processing

tmp: temporary files, including image segments, and processed images

webcam: for processing images obtained via a webcam

The format for the alerts_log.txt is as follows:

segment_number date time CURRENT_VALUE min max

The min and max values are user-defined and should be set by the user in the parameters_min_max.txt table.

To test the screenshotting functionality run:

python3 pys/take-screenshot.py

To configure: see 3.2.2

After configuration, test the image processing functionality by running:

python3 pys/050___READ_ALL_v01.py

Troubleshoot, by reviewing any error messages printed to the console.

Starting the web server:

```
        python3 HMI_Webpage/start_web_server_v2.py
```

Launch any graphical web browser connected the network and aim it at port 8080:

```
Firefox 127.0.0.1:8080
```

3.2.2 Configuration Settings

To clone a GitHub repository, navigate to the repository on GitHub, copy the URL under "Code" (usually an HTTPS link), then open your terminal and run the command "git clone [paste the copied URL]" in the desired directory where you want to clone the repository; this will download a complete copy of the repository to your local machine.

Navigate to the config/ directory.

Install required python modules:

```
        pip3 install –r requirements.txt
```

Segmentation:

```
cd config/semiauto_segmentation/
```

Copy the screenshot here

```
cp  ../ss/ss.jpg  Demo_Screenshot_BEFORE-CLEANING.png
```

Use gimp rectangle tool to cut all image elements that do not contain interpretable data. Save that image to:

Demo_Screenshot_cleaned.png

Then run:

```
python3   010___semi_auto_segmentation_plus_rectangles.py
```

Demo_Screenshot_with_rectangles_BEFORE_EDIT.png is automatically generated.

Now view the Demo_Screenshot_with_rectangles_BEFORE_EDIT.png using gimp:

```
gimp Demo_Screenshot_with_rectangles_BEFORE_EDIT.png
```

To add possible missing segments follow:

020___READ-ME___must_HAND-EDIT_segmentation_table.txt

Then edit the segmentation table if necessary:

```
emacs segmentation_table_BEFORE_EDIT.txt
```

The format for the above table:

The segmentation_table.txt charts the rectangular coordinates of the bounding box for each segment. For example:

```
s01 525 279 227 183
```

MEANS:

s01, the first segment

top_left corner = (525, 279) # (x, y)

rectangle_size = (227, 183) # (width, height)

It is important that there is no white space after the last entry!

Save the edited version as:

segmentation_table.txt

To visualize the corrected version run:

```
python3   030___draw_rectangles_AFTER_EDIT_script_v2.py
```

Lock this important segmentation table:

```
chmod 444 segmentation_table.txt
```

Next specify and or edit Upper and Lower Control Limits for all monitored data in the following self-explanatory configuration file:

config/parameters_min_max.txt

3.2.3 Initial System Verification

Just for testing now run:

```
python3 pys/050___READ_ALL_v01.py
```

4. System Configuration

No further system configuration apart from the above is necessary

4.1 Initial Setup

Clear browser cache every fresh instance of running code

4.2 User Roles and Permissions

Floor manager and machine operator are the two currently intended types of users for the system.

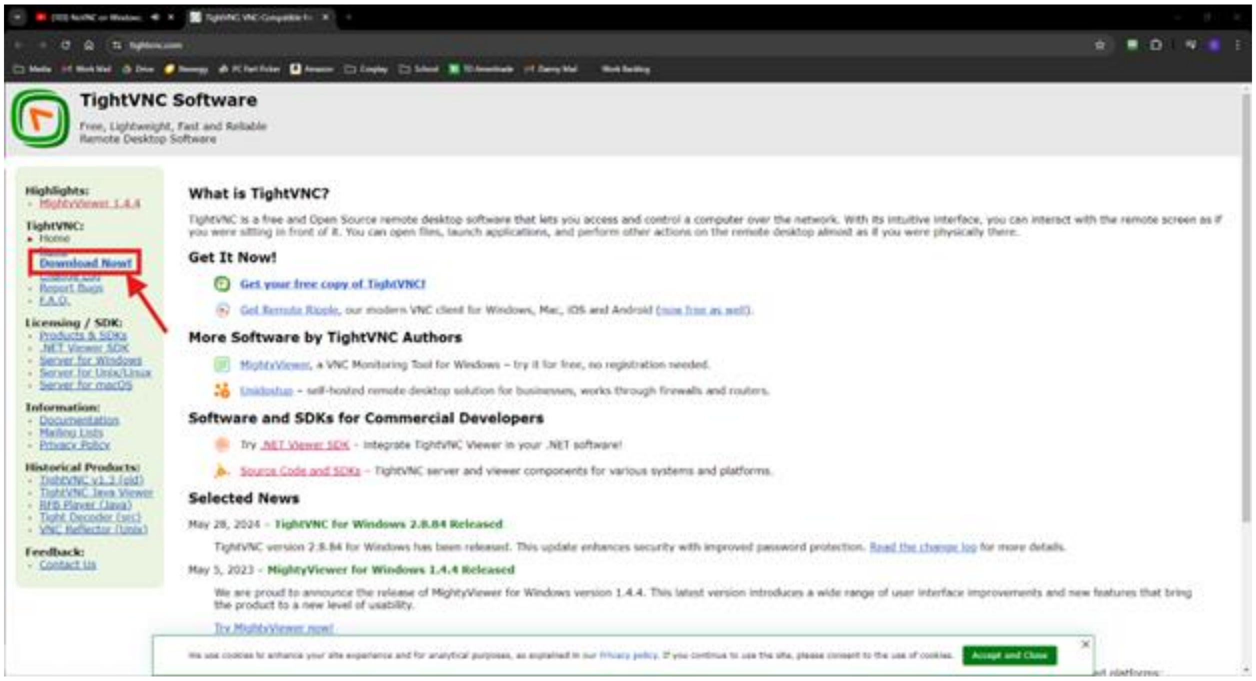
4.3 VNC Connection Parameters

Use YouTube Video:

https://www.youtube.com/watch?v=ExBDUpEkTfk&ab_channel=Jub%27sTechTutorials

For VNC installation and setup:

Go to https://www.tightvnc.com/ and click “Download Now!” on the left


TightVNC Software

Free, Lightweight, Fast and Reliable Remote Desktop Software

Highlights:
- MightyViewer 1.4.4

TightVNC:
- Home
- Download Now!
- Report Bugs
- F.A.Q.

Licensing / SDK:
- Products & SDKs
- .NET Viewer SDK
- Server for Windows
- Server for Unix/Linux
- Server for macOS

Information:
- Documentation
- Mailing Lists
- Privacy Policy

Historical Products:
- TightVNC v1.3 (old)
- TightVNC Java Viewer
- RFB Player (Java)
- Tight Decoder (src)
- VNC Reflector (Unix)

Feedback:
- Contact Us

What is TightVNC?

TightVNC is a free and Open Source remote desktop software that lets you access and control a computer over the network. With its intuitive interface, you can interact with the remote screen as if you were sitting in front of it. You can open files, launch applications, and perform other actions on the remote desktop almost as if you were physically there.

Get It Now!

Get your free copy of TightVNC!

Get Remote Ripple, our modern VNC client for Windows, Mac, iOS and Android (now free as well).

More Software by TightVNC Authors

MightyViewer, a VNC Monitoring Tool for Windows – try it for free, no registration needed.

Unidoshup – self-hosted remote desktop solution for businesses, works through firewalls and routers.

Software and SDKs for Commercial Developers

Try .NET Viewer SDK - integrate TightVNC Viewer in your .NET software!

Source Code and SDKs – TightVNC server and viewer components for various systems and platforms.

Selected News

May 28, 2024 - TightVNC for Windows 2.8.84 Released

TightVNC version 2.8.84 for Windows has been released. This update enhances security with improved password protection. Read the change log for more details.

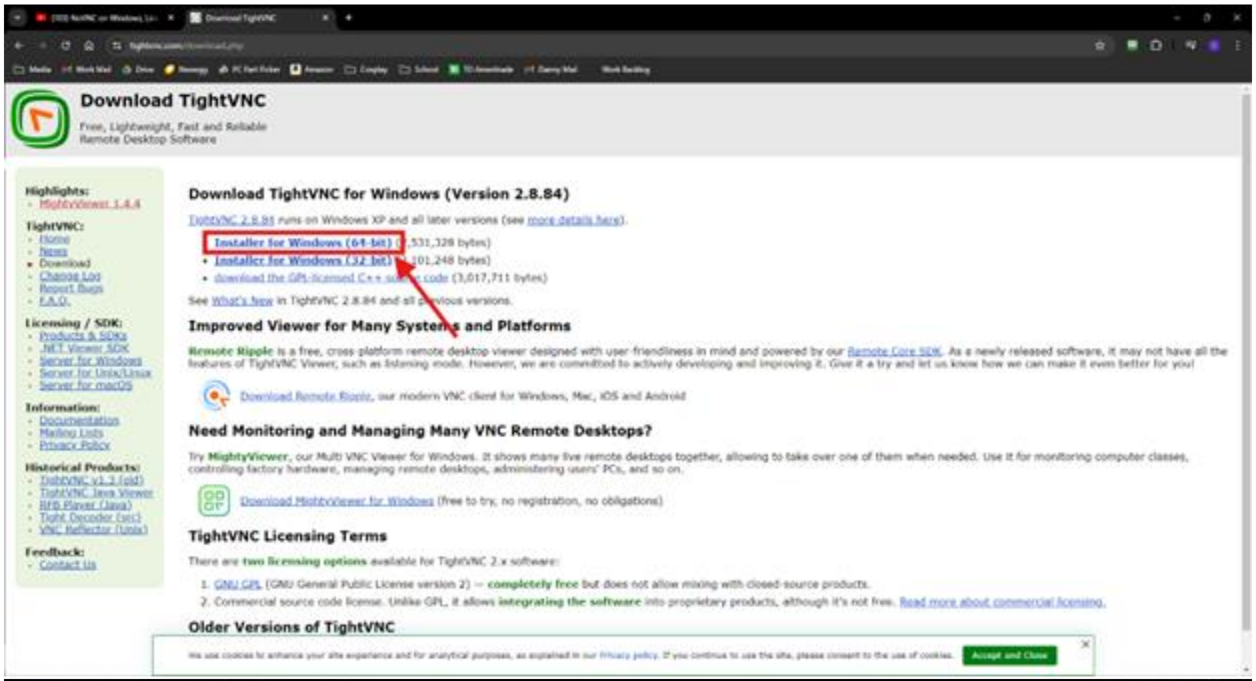
May 5, 2023 - MightyViewer for Windows 1.4.4 Released

We are proud to announce the release of MightyViewer for Windows version 1.4.4. This latest version introduces a wide range of user interface improvements and new features that bring the product to a new level of usability.

Try MightyViewer now!

We use cookies to enhance your site experience and for analytical purposes, as explained in our Privacy policy. If you continue to use the site, please consent to the use of cookies.

Click “Installer for Windows (64-bit)” to download the installer. Once it finishes downloading, run the installer.



Download TightVNC

Free, Lightweight, Fast and Reliable
Remote Desktop Software

Highlights:
- MightyViewer 1.4.4

TightVNC:
- Home
- News
- Download
- Change Log
- Report Bugs
- F.A.Q.

Licensing / SDK:
- Products & SDKs
- .NET Viewer SDK
- Server for Windows
- Server for Unix/Linux
- Server for macOS

Information:
- Documentation
- Mailing Lists
- Privacy Policy

Historical Products:
- TightVNC v1.3 (old)
- TightVNC Java Viewer
- RFB Player (Java)
- Tight Decoder (src)
- VNC Reflector (Unix)

Feedback:
- Contact Us

Download TightVNC for Windows (Version 2.8.84)

TightVNC 2.8.84 runs on Windows XP and all later versions (see more details here).

- **Installer for Windows (64-bit)** (2,531,328 bytes)
- **Installer for Windows (32-bit)** (2,101,248 bytes)
- download the GPL-licensed C++ source code (3,017,711 bytes)

See What's New in TightVNC 2.8.84 and all previous versions.

Improved Viewer for Many Systems and Platforms

Remote Ripple is a free, cross-platform remote desktop viewer designed with user-friendliness in mind and powered by our Remote Core SDK. As a newly released software, it may not have all the features of TightVNC Viewer, such as listening mode. However, we are committed to actively developing and improving it. Give it a try and let us know how we can make it even better for you!

Download Remote Ripple, our modern VNC client for Windows, Mac, iOS and Android

Need Monitoring and Managing Many VNC Remote Desktops?

Try **MightyViewer**, our Multi VNC Viewer for Windows. It shows many live remote desktops together, allowing to take over one of them when needed. Use it for monitoring computer classes, controlling factory hardware, managing remote desktops, administering users' PCs, and so on.

Download MightyViewer for Windows (free to try, no registration, no obligations)

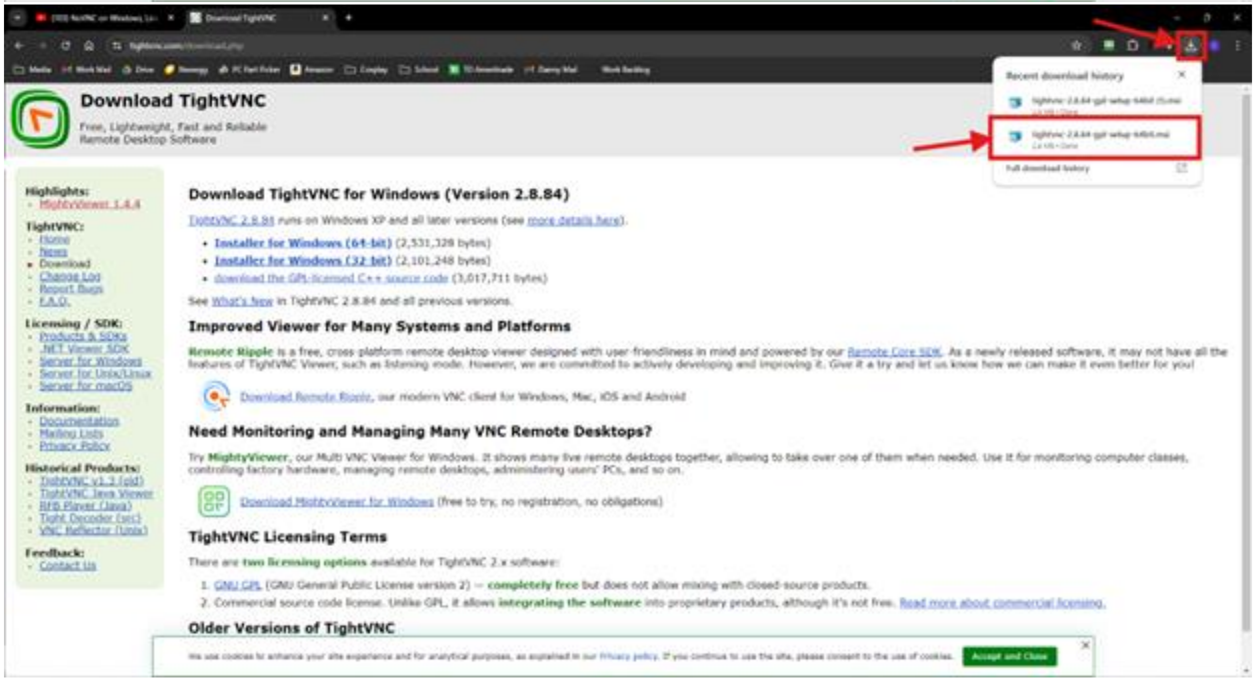
TightVNC Licensing Terms

There are **two licensing options** available for TightVNC 2.x software:

1. GNU GPL (GNU General Public License version 2) — **completely free** but does not allow mixing with closed-source products.
2. Commercial source code license. Unlike GPL, it allows **integrating the software** into proprietary products, although it's not free. Read more about commercial licensing.

Older Versions of TightVNC

We use cookies to enhance your site experience and for analytical purposes, as explained in our Privacy policy. If you continue to use the site, please consent to the use of cookies.



Recent download history

tightvnc-2.8.84-gpl-setup-64bit (1).msi

tightvnc-2.8.84-gpl-setup-64bit.msi
2.4 MB • Done

Full download history

Download TightVNC

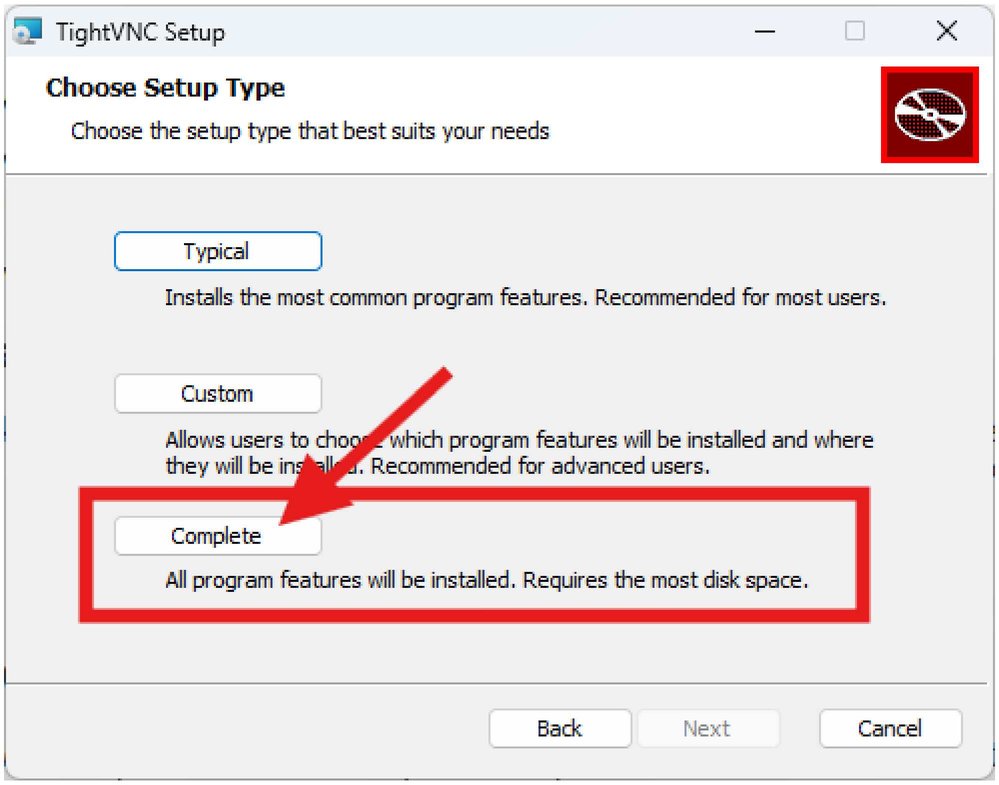
Free, Lightweight, Fast and Reliable
Remote Desktop Software

Download TightVNC for Windows (Version 2.8.84)

TightVNC 2.8.84 runs on Windows XP and all later versions (see more details here).

- **Installer for Windows (64-bit)** (2,531,328 bytes)
- **Installer for Windows (32-bit)** (2,101,248 bytes)
- download the GPL-licensed C++ source code (3,017,711 bytes)

Go through the steps of the TightVNC installer. When you get to the “Choose Setup Type” screen, click the “Complete” button.


TightVNC Setup

Choose Setup Type

Choose the setup type that best suits your needs

Typical

Installs the most common program features. Recommended for most users.

Custom

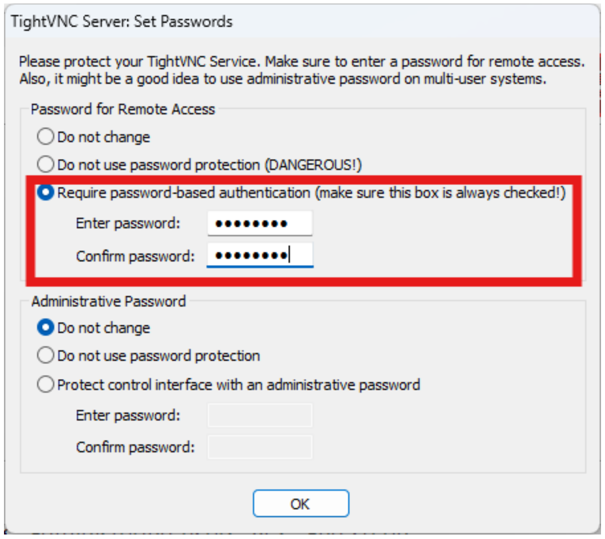
Allows users to choose which program features will be installed and where they will be installed. Recommended for advanced users.

Complete

All program features will be installed. Requires the most disk space.

Do not change any other settings in the installer. Finish installation (requires administrative privileges) and select “Yes” when asked if you want the program to be able to make changes to your device.

Enter a secure password in the “Password for Remote Access” section during installation. Make sure the password is saved somewhere, as it will be needed to access the server PC from any client.


TightVNC Server: Set Passwords

Please protect your TightVNC Service. Make sure to enter a password for remote access. Also, it might be a good idea to use administrative password on multi-user systems.

Password for Remote Access

- Do not change
- Do not use password protection (DANGEROUS!)
- [✓ Require password-based authentication (make sure this box is always checked!)]

Enter password: ••••••••

Confirm password: ••••••••

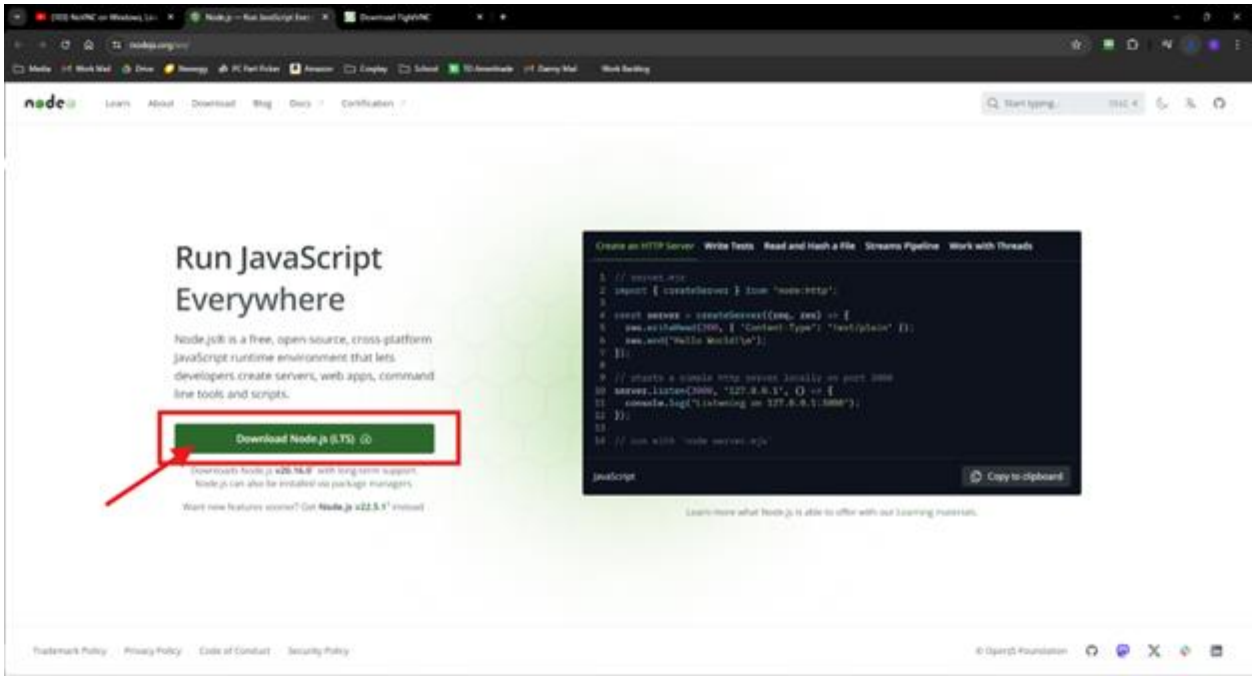
Administrative Password

- [✓ Do not change]
- Do not use password protection
- Protect control interface with an administrative password

Enter password:

Confirm password:

Go to nodejs.org and click the “Download Node.js (LTS)” button to download the NodeJS installer.



Run JavaScript Everywhere

Node.js® is a free, open-source, cross-platform JavaScript runtime environment that lets developers create servers, web apps, command line tools and scripts.

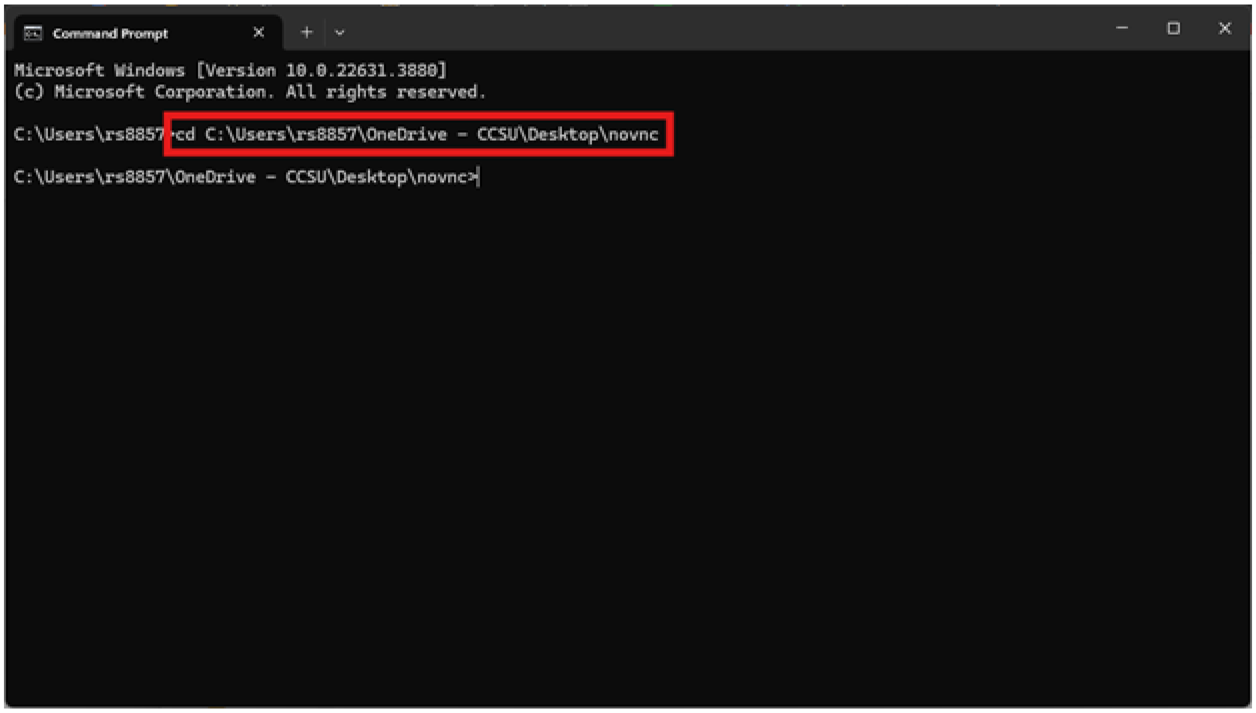
Downloads Node.js v20.16.0 with long-term support. Node.js can also be installed via package managers.

Want new features sooner? Get Node.js v22.5.1 instead.

```
// server.mjs
import { createServer } from 'node:http';

const server = createServer((req, res) => {
  res.writeHead(200, { 'Content-Type': 'text/plain' });
  res.end('Hello World!\n');
});

// starts a simple http server locally on port 3000
server.listen(3000, '127.0.0.1', () => {
  console.log('Listening on 127.0.0.1:3000');
});

// run with `node server.mjs`
```

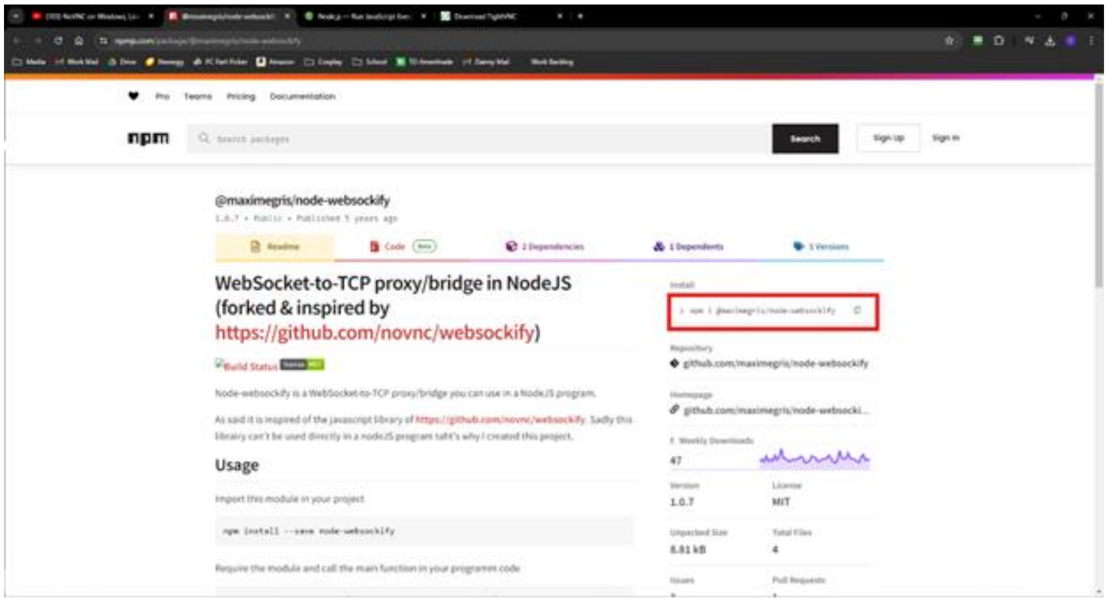
Learn more what Node.js is able to offer with our Learning materials.

Create a new folder on your computer for the vnc files. Open command prompt and use the cd command to change directories to the folder you just made.



```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rs8857>cd C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc

C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc>
```

Go to https://www.npmjs.com/package/@maximegris/node-websockify and copy the installation command seen in the top right of the webpage.



@maximegris/node-websockify

1.0.7 • Public • Published 5 years ago

WebSocket-to-TCP proxy/bridge in NodeJS (forked & inspired by https://github.com/novnc/websockify)

Build Status

Node-websockify is a WebSocket-to-TCP proxy/bridge you can use in a NodeJS program.

As said it is inspired of the javascript library of https://github.com/novnc/websockify. Sadly this librairy can't be used directly in a nodeJS program taht's why I created this project.

Usage

Import this module in your project

```
npm install --save node-websockify
```

Require the module and call the main function in your programm code

Install

```
> npm i @maximegris/node-websockify
```

Repository: github.com/maximegris/node-websockify

Homepage: github.com/maximegris/node-websocki…

Weekly Downloads: 47

Version: 1.0.7

License: MIT

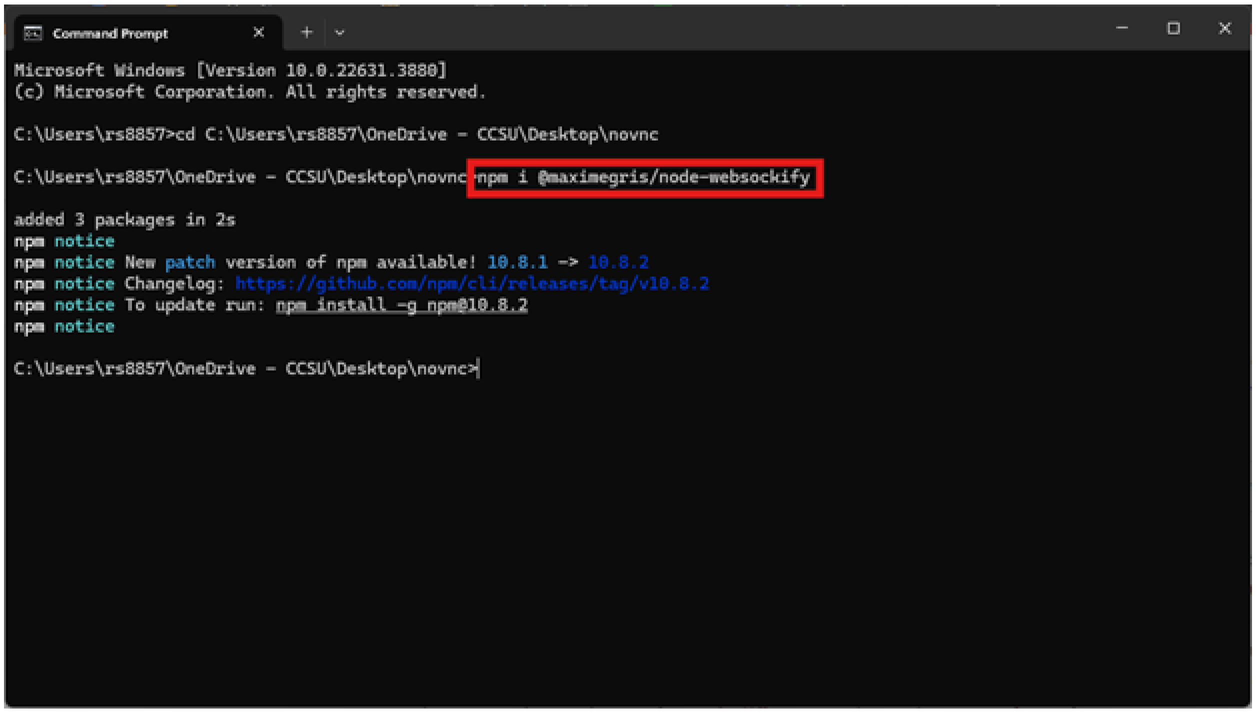
Unpacked Size: 8.81 kB

Total Files: 4

Issues

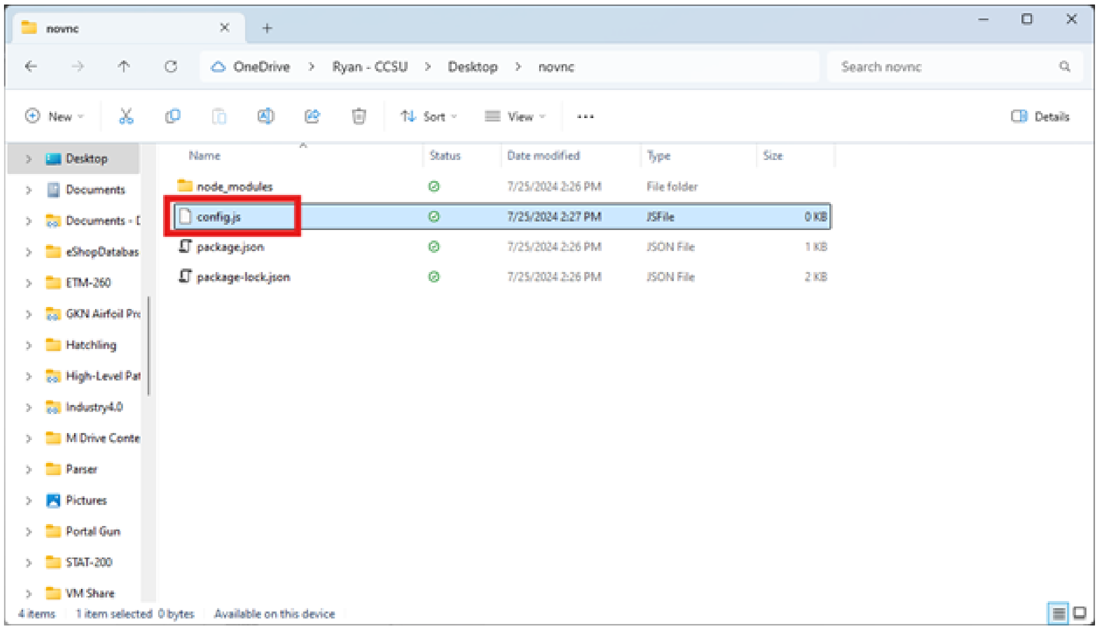
Pull Requests

Paste the command into command prompt and run it to install the node-websockify package.



```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rs8857>cd C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc

C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc>npm i @maximegris/node-websockify

added 3 packages in 2s
npm notice
npm notice New patch version of npm available! 10.8.1 -> 10.8.2
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.8.2
npm notice To update run: npm install -g npm@10.8.2
npm notice

C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc>
```

In the folder you created, there should now be two new files named package.json and package-lock.json as well as a new folder named node_modules. Create a new file called config.js and make sure the file type is JSFile (extension ending in .js).



Name	Status	Date modified	Type	Size
node_modules	✓	7/25/2024 2:26 PM	File folder	
config.js	✓	7/25/2024 2:27 PM	JSFile	0 KB
package.json	✓	7/25/2024 2:26 PM	JSON File	1 KB
package-lock.json	✓	7/25/2024 2:26 PM	JSON File	2 KB

Open config.js with notepad or any other text editor or IDE and paste the following inside:

var websockify = require('@maximegris/node-websockify');

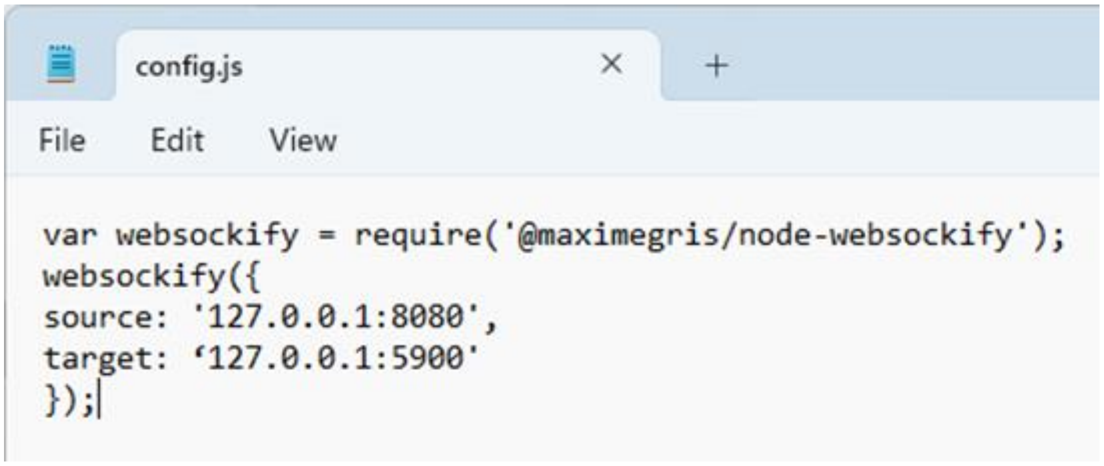
websockify({

source: '127.0.0.1:8080',

target: '127.0.0.1:5900'

});

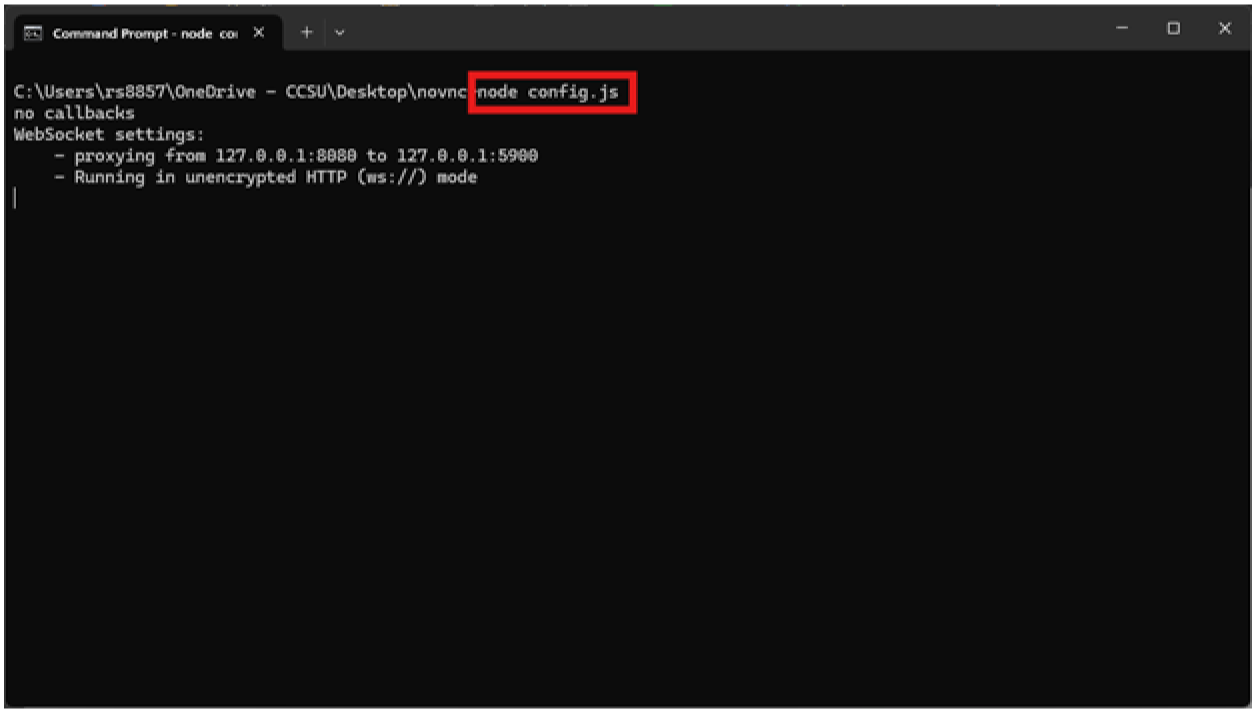
The template for this code can also be found on the node-websockify website linked above, but a few changes are needed.



```
var websockify = require('@maximegris/node-websockify');
websockify({
source: '127.0.0.1:8080',
target: '127.0.0.1:5900'
});
```

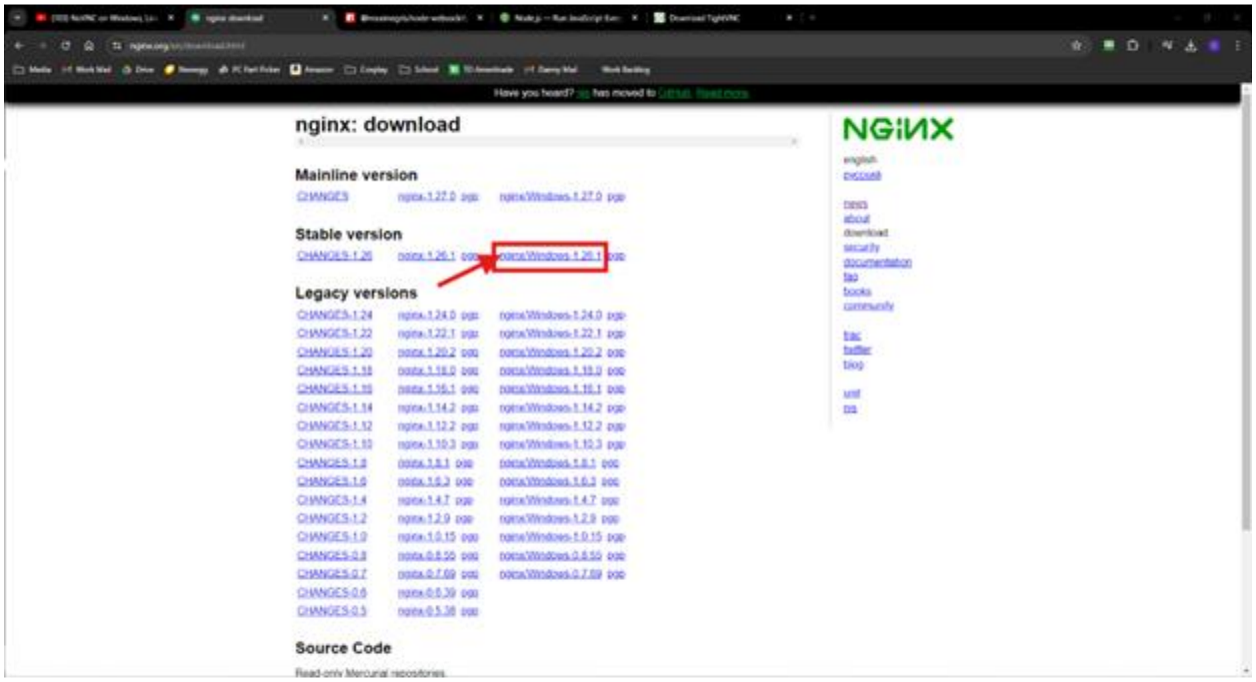
Save the changes and close notepad.

In command prompt, run the command node config.js to make your changes take effect.



```
C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc>node config.js
no callbacks
WebSocket settings:
    - proxying from 127.0.0.1:8080 to 127.0.0.1:5900
    - Running in unencrypted HTTP (ws://) mode
```

Next go to https://nginx.org/en/download.html and download the latest stable version of NGinx for Windows.



Have you heard? njs has moved to GitHub. Read more

nginx: download

Mainline version

CHANGES	nginx-1.27.0 pgp	nginx/Windows-1.27.0 pgp

Stable version

CHANGES-1.26	nginx-1.26.1 pgp	nginx/Windows-1.26.1 pgp

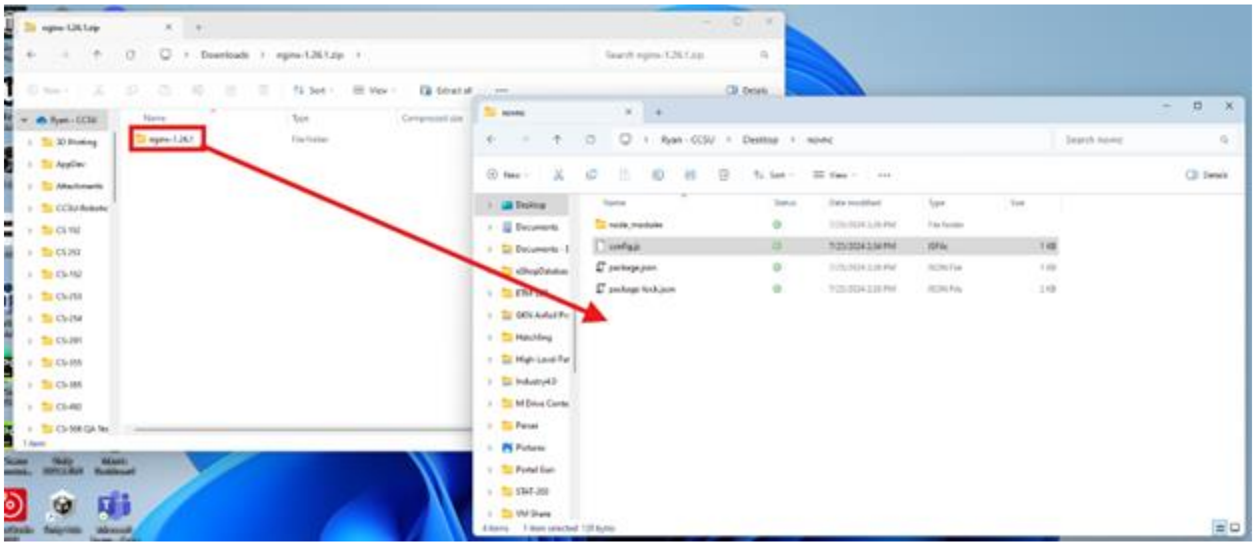
Legacy versions

CHANGES-1.24	nginx-1.24.0 pgp	nginx/Windows-1.24.0 pgp
CHANGES-1.22	nginx-1.22.1 pgp	nginx/Windows-1.22.1 pgp
CHANGES-1.20	nginx-1.20.2 pgp	nginx/Windows-1.20.2 pgp
CHANGES-1.18	nginx-1.18.0 pgp	nginx/Windows-1.18.0 pgp
CHANGES-1.16	nginx-1.16.1 pgp	nginx/Windows-1.16.1 pgp
CHANGES-1.14	nginx-1.14.2 pgp	nginx/Windows-1.14.2 pgp
CHANGES-1.12	nginx-1.12.2 pgp	nginx/Windows-1.12.2 pgp
CHANGES-1.10	nginx-1.10.3 pgp	nginx/Windows-1.10.3 pgp
CHANGES-1.8	nginx-1.8.1 pgp	nginx/Windows-1.8.1 pgp
CHANGES-1.6	nginx-1.6.3 pgp	nginx/Windows-1.6.3 pgp
CHANGES-1.4	nginx-1.4.7 pgp	nginx/Windows-1.4.7 pgp
CHANGES-1.2	nginx-1.2.9 pgp	nginx/Windows-1.2.9 pgp
CHANGES-1.0	nginx-1.0.15 pgp	nginx/Windows-1.0.15 pgp
CHANGES-0.8	nginx-0.8.55 pgp	nginx/Windows-0.8.55 pgp
CHANGES-0.7	nginx-0.7.69 pgp	nginx/Windows-0.7.69 pgp
CHANGES-0.6	nginx-0.6.39 pgp	
CHANGES-0.5	nginx-0.5.38 pgp	

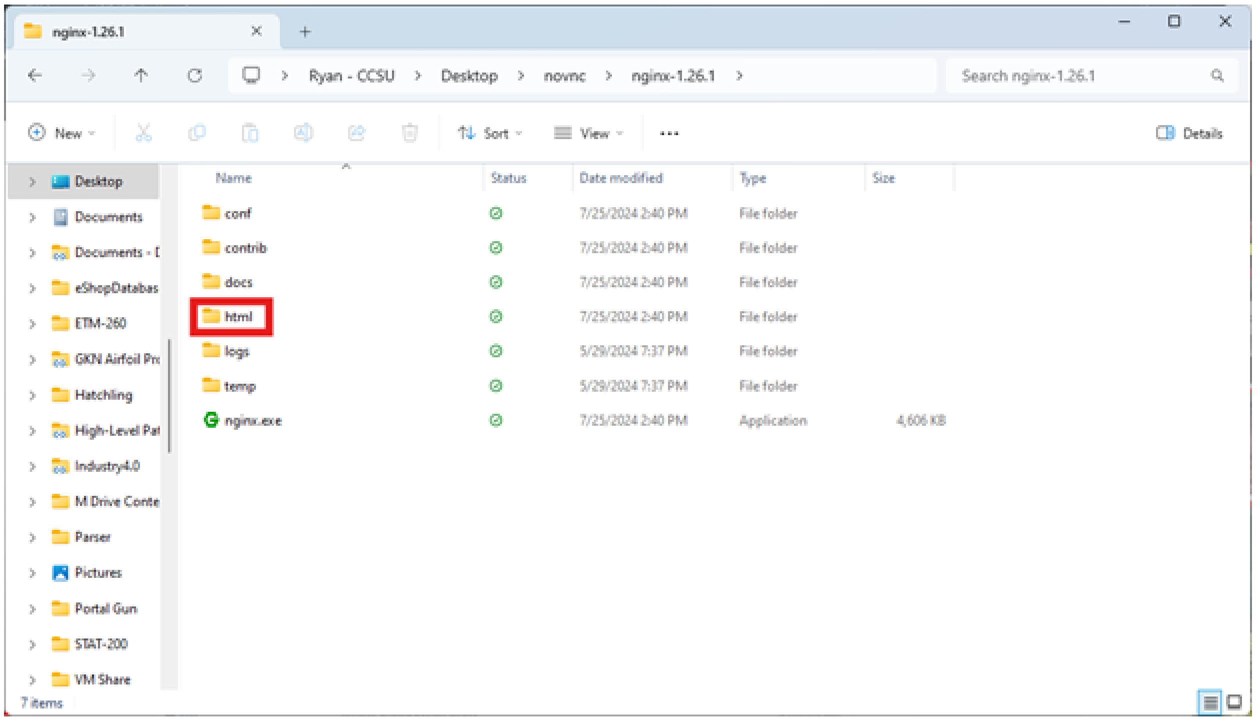
Source Code

Read-only Mercurial repositories:

Open the .zip folder that downloads, and drag the “nginx-[version]” folder into the folder you created earlier.

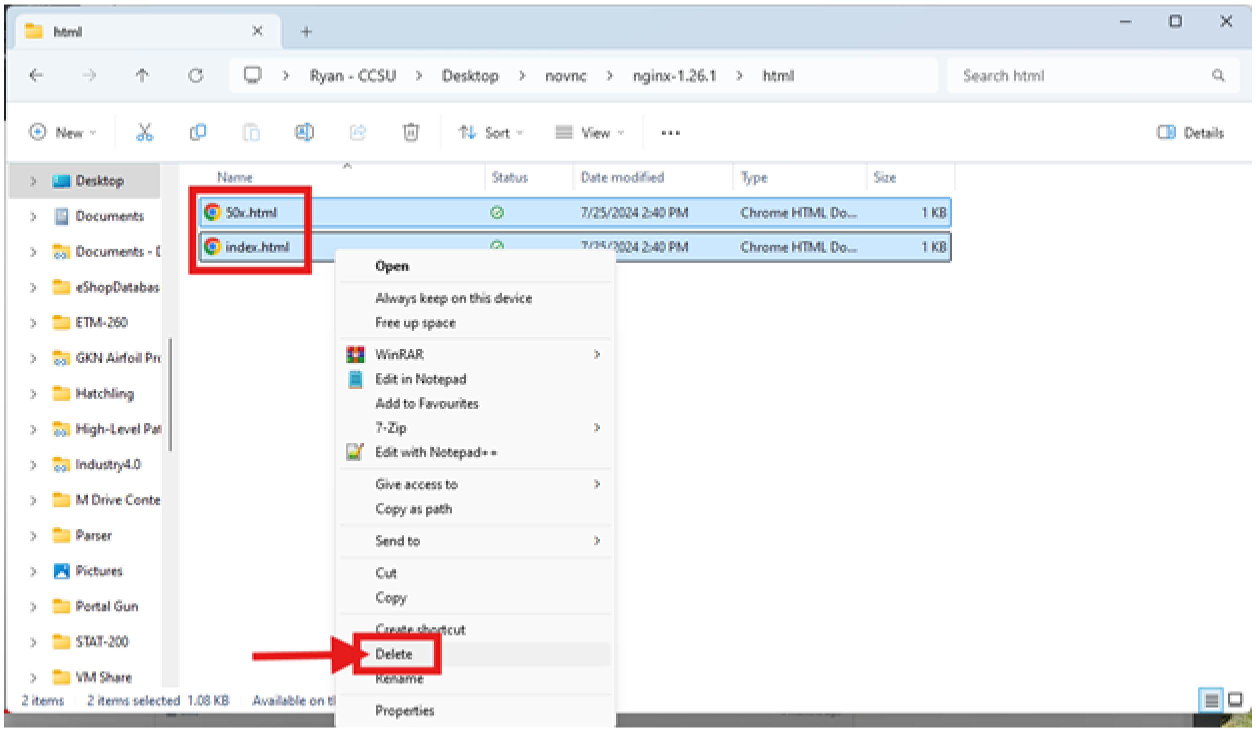


Inside the nginx-[version] folder, navigate to the “html” folder.



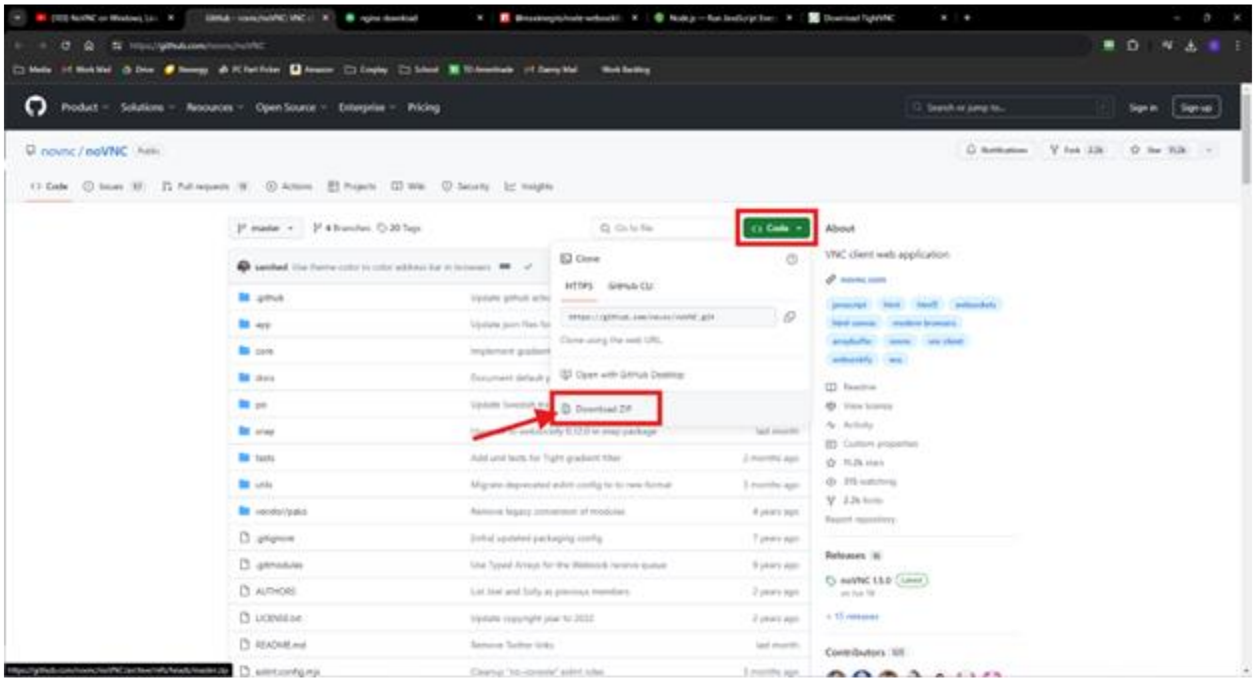
Name	Status	Date modified	Type	Size
conf		7/25/2024 2:40 PM	File folder	
contrib		7/25/2024 2:40 PM	File folder	
docs		7/25/2024 2:40 PM	File folder	
html		7/25/2024 2:40 PM	File folder	
logs		5/29/2024 7:37 PM	File folder	
temp		5/29/2024 7:37 PM	File folder	
nginx.exe		7/25/2024 2:40 PM	Application	4,606 KB

Delete the contents of the folder, which should be two .html files.

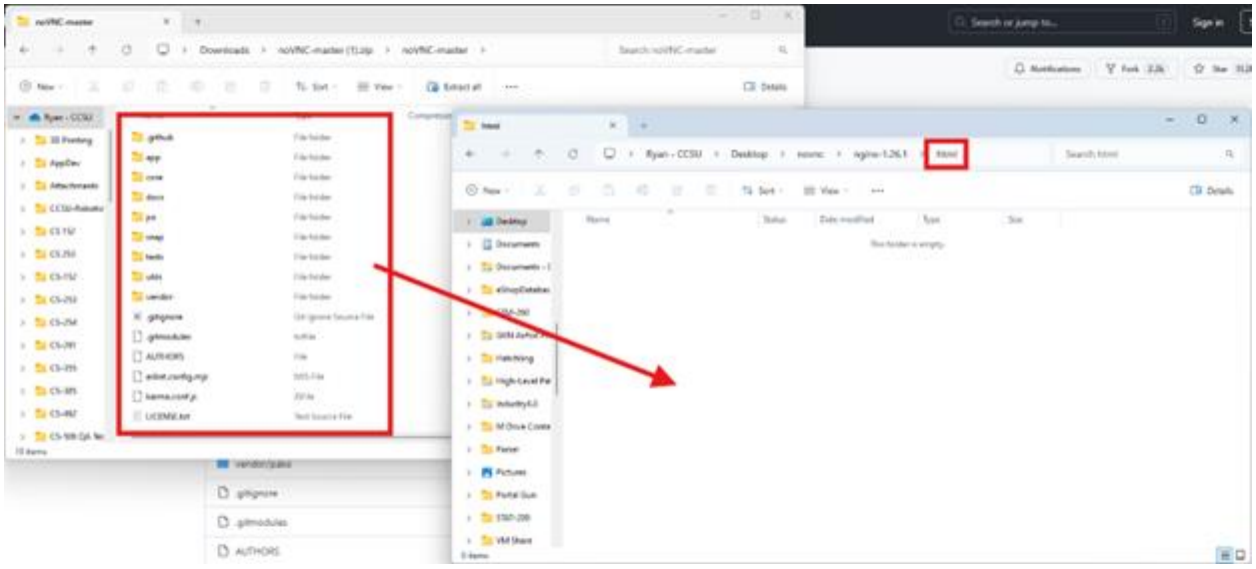


Name	Status	Date modified	Type	Size
50x.html		7/25/2024 2:40 PM	Chrome HTML Do...	1 KB
index.html		7/25/2024 2:40 PM	Chrome HTML Do...	1 KB

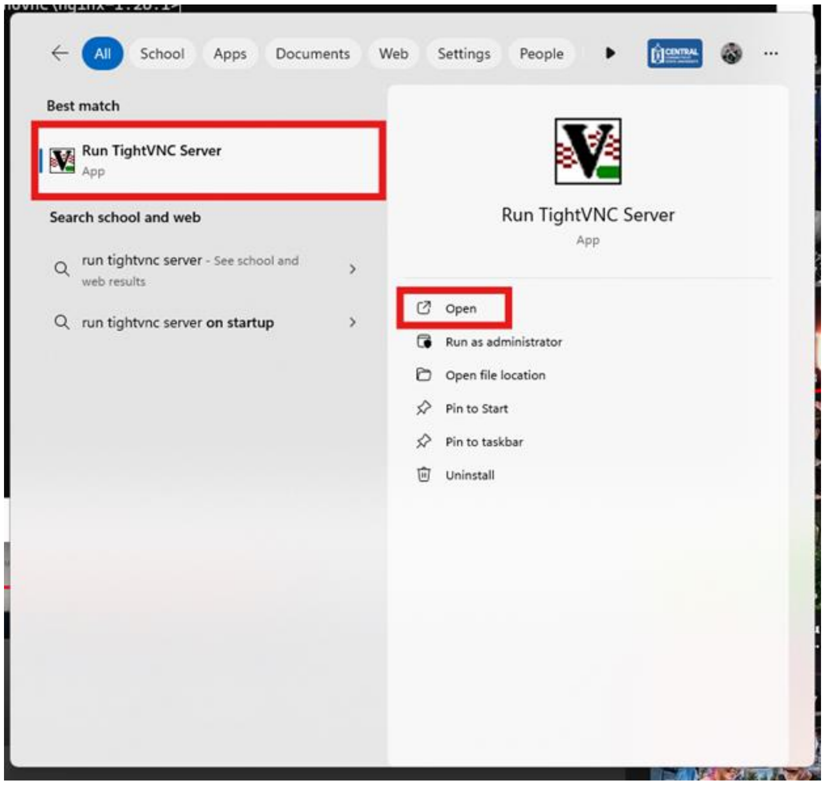
Go to https://github.com/novnc/noVNC and download all files as a .zip.



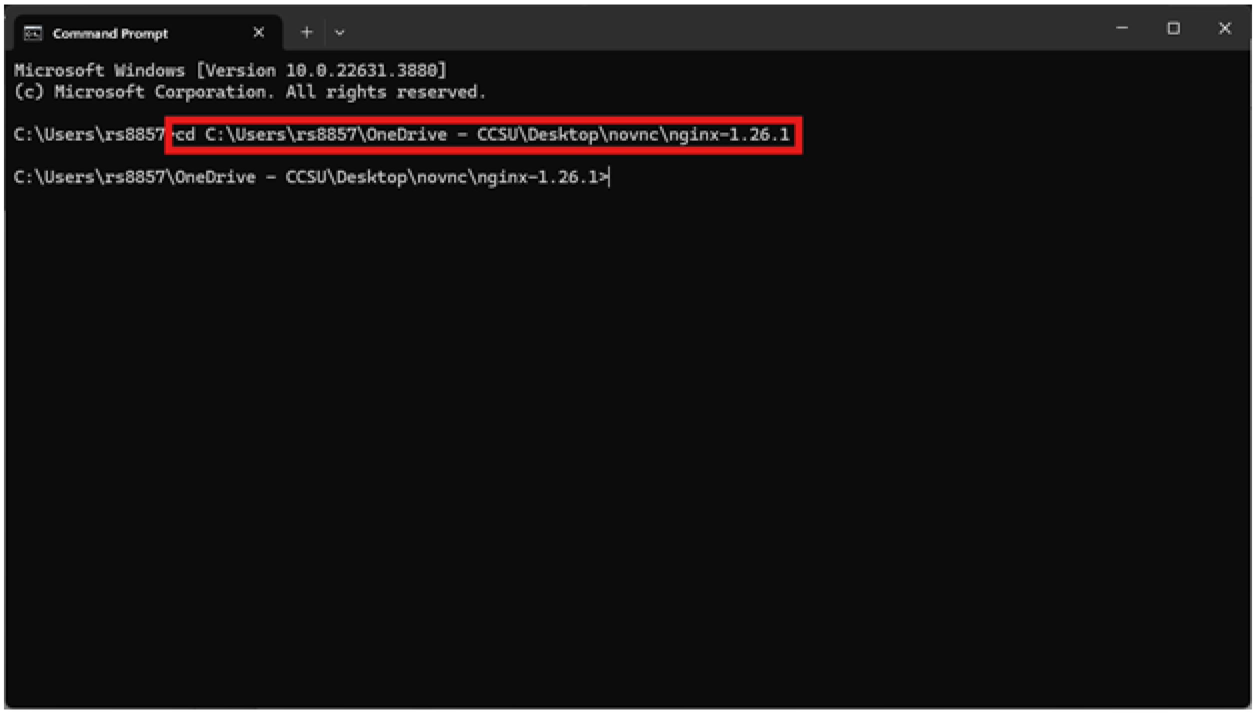
Extract the files into the html folder that you just emptied.



Press Windows+S to open the Windows search bar and open the app "Run TightVNC Server"

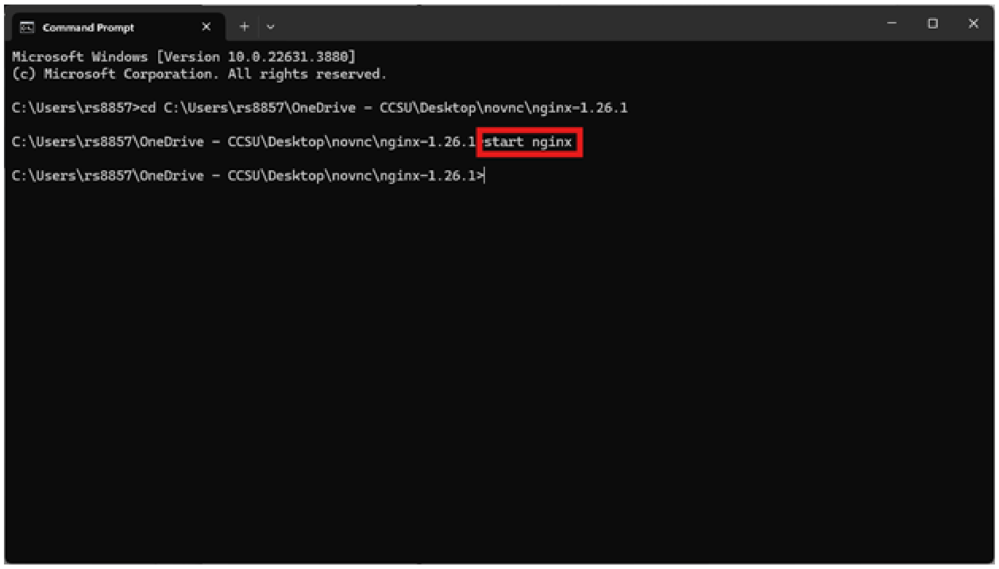


Open a new command prompt window and navigate to the nginx-[version] folder inside the folder you created (one level up from the html folder).



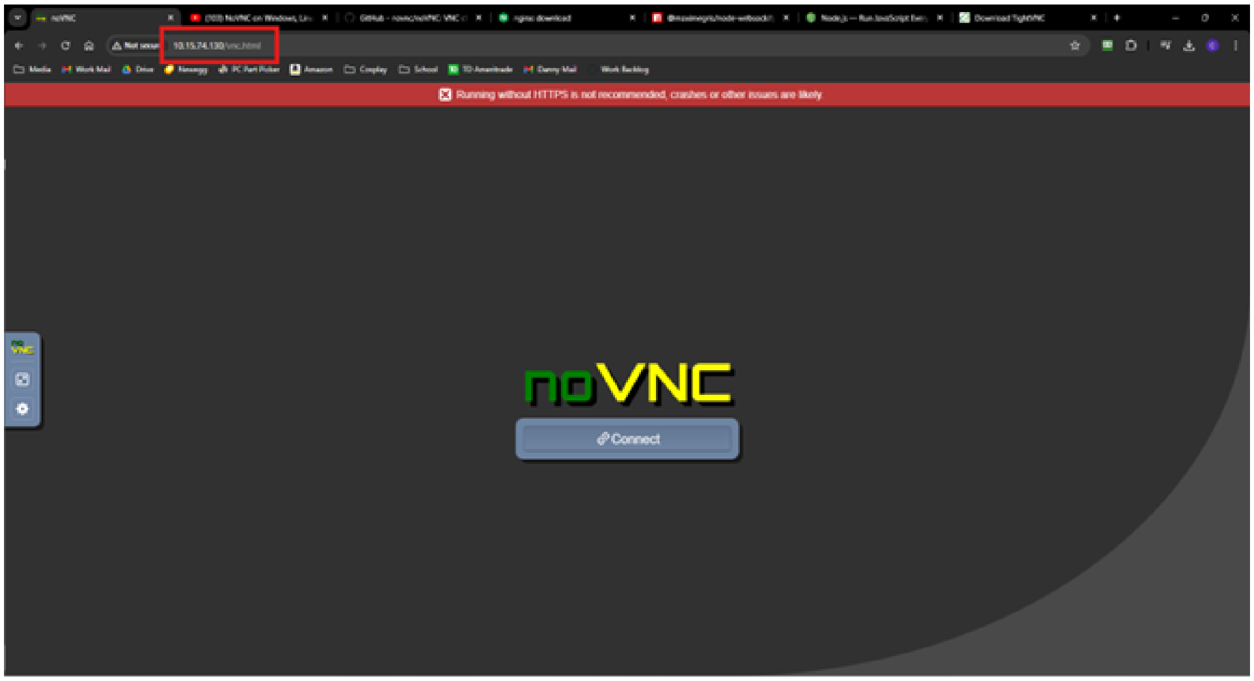
```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rs8857>cd C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc\nginx-1.26.1

C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc\nginx-1.26.1>
```

Run the command start nginx to spin up the VNC server

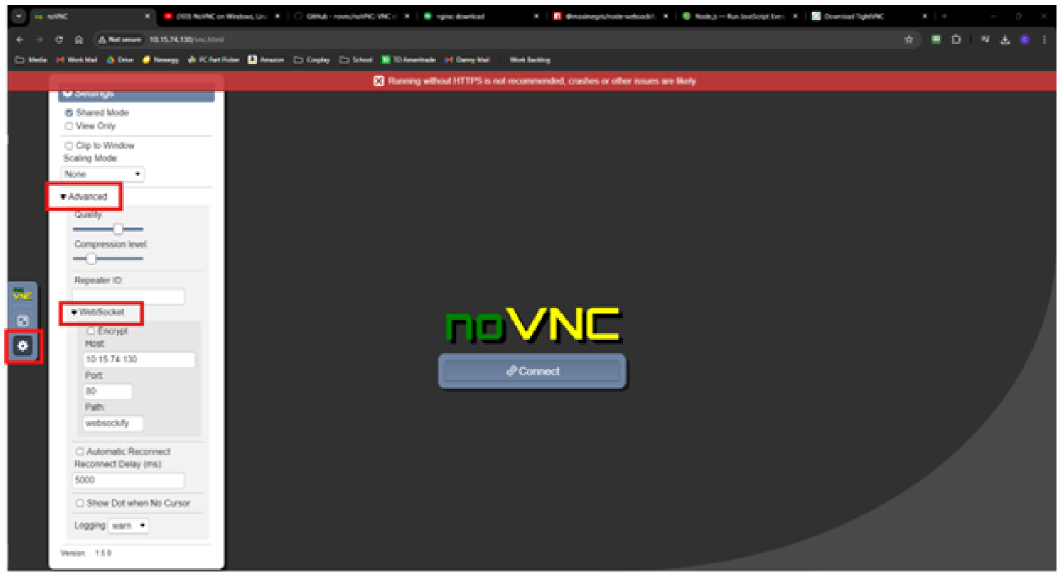


```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rs8857>cd C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc\nginx-1.26.1

C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc\nginx-1.26.1>start nginx

C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc\nginx-1.26.1>
```

In your web browser type [computer IP]/vnc.html, where [computer IP] is the IP of the computer that has the VNC server set up on it. If you are testing this from the same computer, you can use the loopback IP (127.0.0.1)

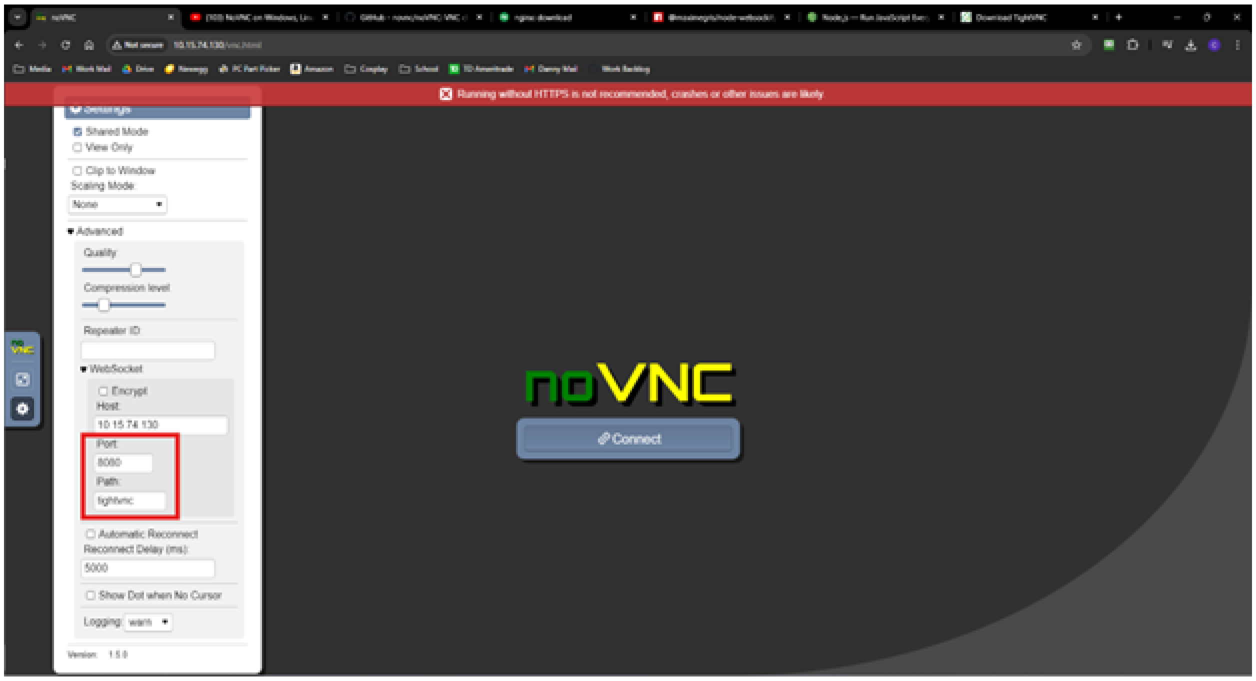

10.15.74.130/vnc.html
Running without HTTPS is not recommended, crashes or other issues are likely
noVNC
Connect

On the left of the screen, click the cog icon to open the settings and expand the “Advanced” and “WebSocket” dropdowns.


10.15.74.130/vnc.html
Running without HTTPS is not recommended, crashes or other issues are likely
Settings
noVNC
Connect

☑ Shared Mode
☐ View Only
☐ Clip to Window
Scaling Mode: None
▾ Advanced
Quality
Compression level
Repeater ID
▾ WebSocket
☐ Encrypt
Host: 10.15.74.130
Port: 80
Path: websockify
☐ Automatic Reconnect
Reconnect Delay (ms): 5000
☐ Show Dot when No Cursor
Logging: warn
Version: 1.5.0

In the “Port” text box use the port for the source IP you typed in config.js earlier. If you pasted the provided text, this should be 8080. In the “Path” text box, type “tightvnc”. Be sure to check that all fields are filled, including Host, port, and path. VNC sometimes erases a field if you refresh so it will not work.


Running without HTTPS is not recommended, crashes or other issues are likely

Settings
- [✓] Shared Mode
- [ ] View Only
- [ ] Clip to Window

Scaling Mode: None

Advanced
Quality
Compression level
Repeater ID:

WebSocket
- [ ] Encrypt

Host: 10.15.74.130
Port: 8080
Path: tightvnc

- [ ] Automatic Reconnect

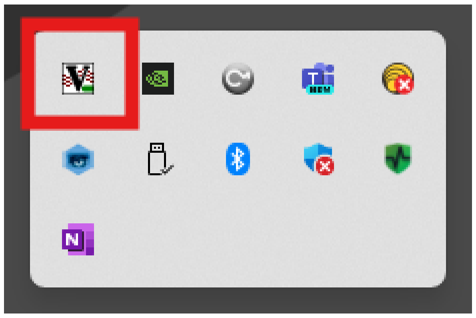
Reconnect Delay (ms): 5000

- [ ] Show Dot when No Cursor

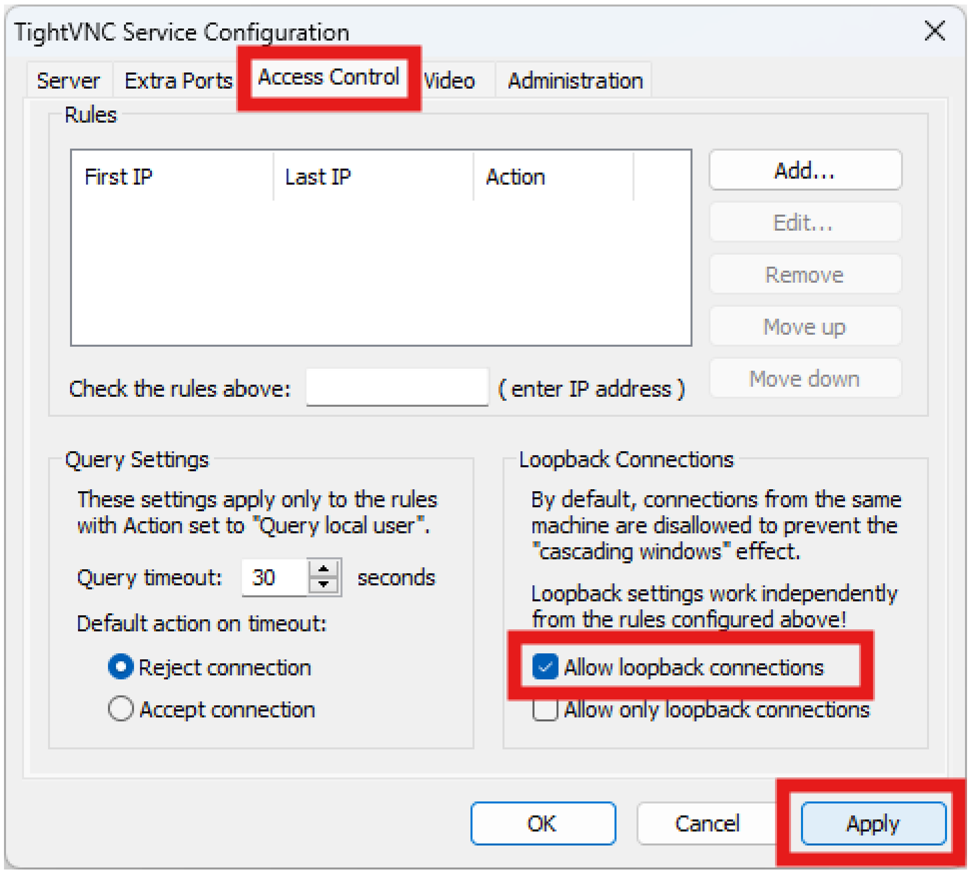
Logging: warn

Version: 1.5.0

Click the “Connect” button in the center of the screen to connect to the VNC server. If you are testing this on the same computer, you may get an error at the top of the screen saying “New connection has been rejected with reason: Sorry, loopback connections are not enabled”. This is normal. To fix this, click the ^ symbol on the bottom right side of the taskbar on your computer’s main monitor to open the system tray. Find the TightVNC Service app and double click it to open it.



Inside TightVNC Service, go to the "Access Control" tab, under the Loopback Connections section click the checkbox next to "Allow loopback connections" and click "Apply". Then click "OK" to close the window.



Rules

First IP	Last IP	Action

Check the rules above: (enter IP address)

Query Settings

These settings apply only to the rules with Action set to "Query local user".

Query timeout: 30 seconds

Default action on timeout:

[✓ Reject connection]
Accept connection

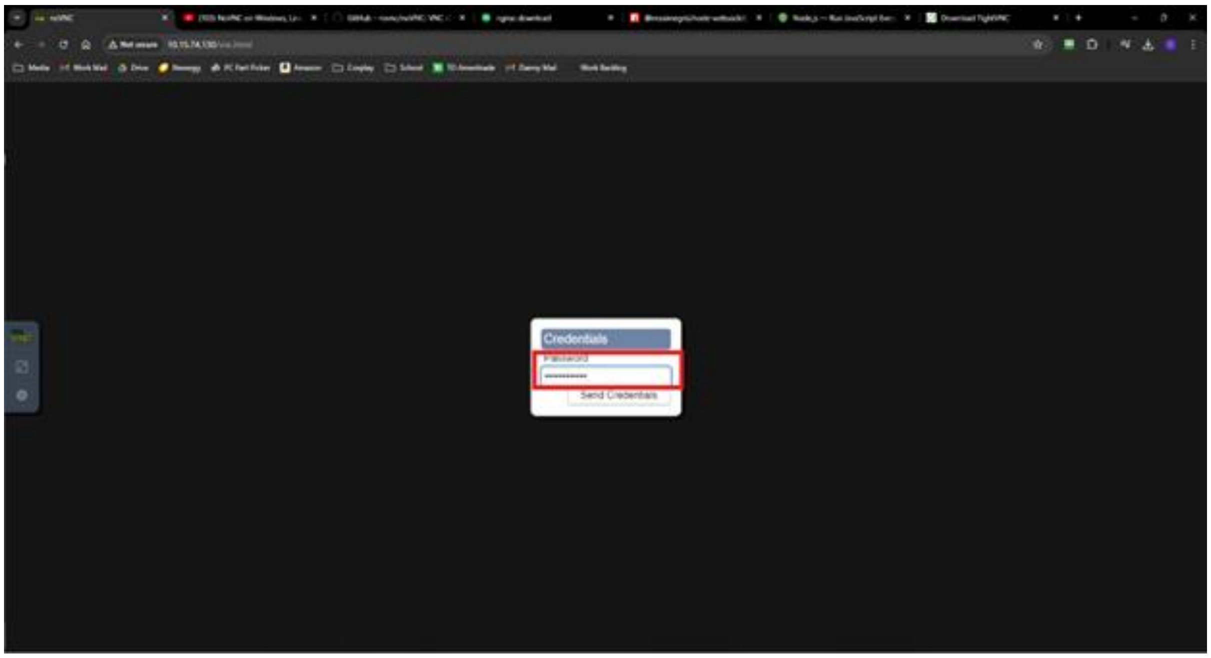
Loopback Connections

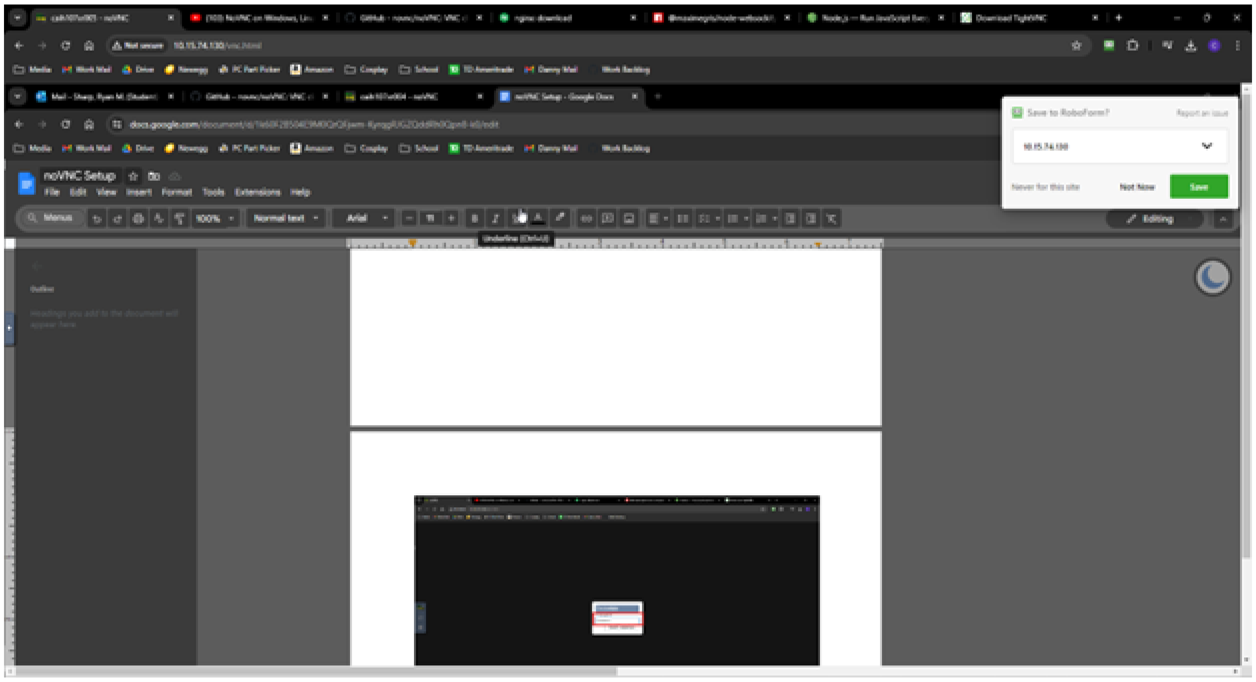
By default, connections from the same machine are disallowed to prevent the "cascading windows" effect.

Loopback settings work independently from the rules configured above!

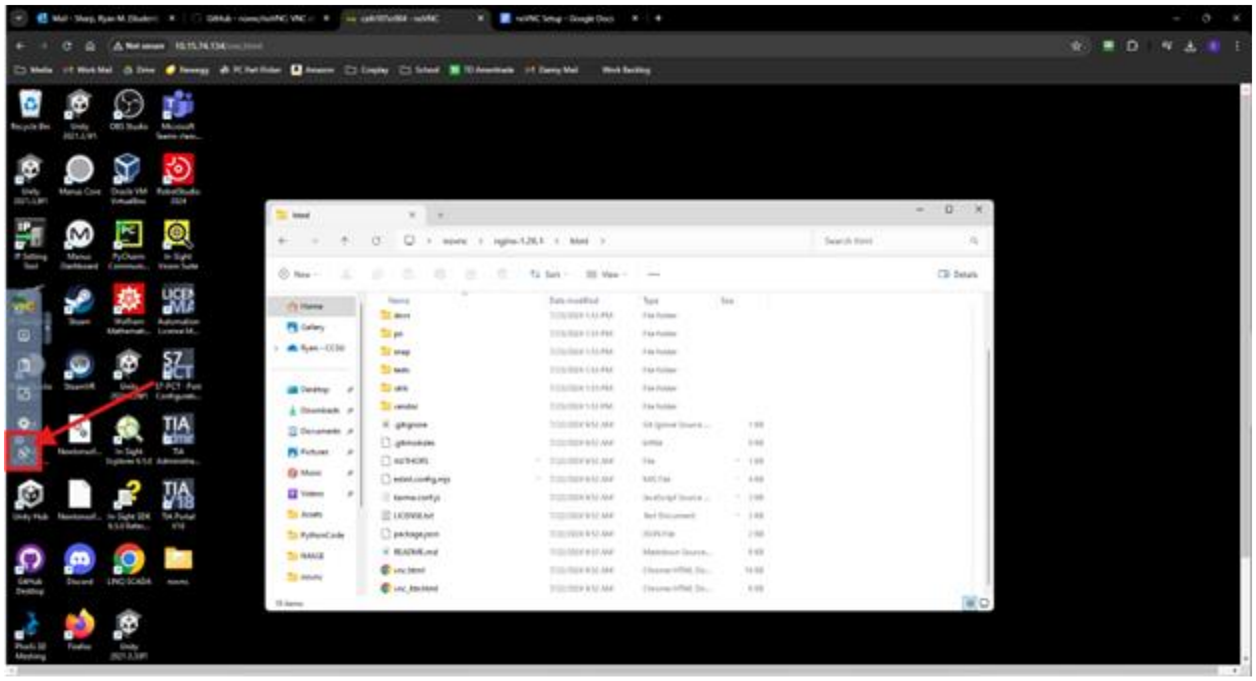
[✓ Allow loopback connections]
Allow only loopback connections

Now if you try clicking the Connect button in noVNC on your browser again, you should be greeted with the credentials screen instead of an error. Enter the password you made earlier while installing TightVNC and you should see the screen of the server PC. If you are testing this on the same computer as the server and you only have one monitor, you may see the header of your browser duplicated over and over and you will find that you can't move your mouse to the rest of the screen. If this happens, you can end the connection by refreshing the tab, or regain control of your whole screen by changing browser tabs.

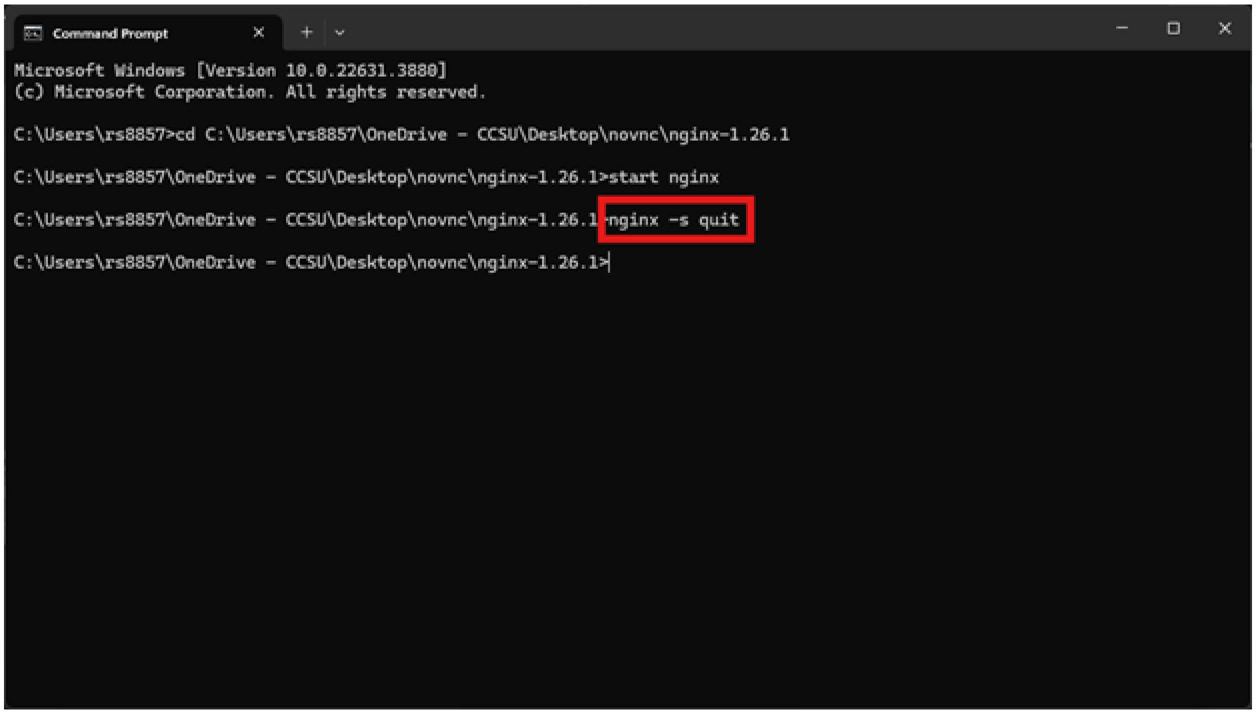




When you are finished, use the menu on the left of the screen to disconnect from the server PC.



To shut down the server on the host, open command prompt to the nginx-[version] folder or use the same window as before if it is still open and type the command nginx -s quit



```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\rs8857>cd C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc\nginx-1.26.1

C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc\nginx-1.26.1>start nginx

C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc\nginx-1.26.1>nginx -s quit

C:\Users\rs8857\OneDrive - CCSU\Desktop\novnc\nginx-1.26.1>
```

4.4 Screenshot Interpretation Settings

See above.

5. Main Usage Scenario

5.1 Primary Workflow

After the initial setup as above, the hmi device is booted, main.py is launched to obtain the screenshot, to interpret, and to generate the hmi.html.

Then direct any browser at the appropriate port to monitor current data: 127.0.0.1:8080

5.1.1 Establishing VNC Connection

See steps above

5.1.2 Capturing HMI Screenshot

Utilizing a script provided by Ryan Sharp, originally called client.py it uses a python library to take a screenshot of your computer screen when you run in it. We updated the script to but a buffer in the time from run and screenshotting because we were running the HMI emulator and the scripts on the same device, so we had to switch in between screens to make sure the screenshot was taken of the correct things.

5.1.3 Data Interpretation and Display

Data processing and interpretation occurs via main.py.

The file hmi.html is periodically created using data stored in the ofkrse.db database and in the log files.

Multiple machines can be monitored simultaneously, and the switching of the display occurs through the drop-down menu at the top left corner of the web page.

5.1.4 Data Interaction and Analysis

Download Linq Builder and get Demo.fsuprj file from CCSU-robotics github, open that demo in linq builder and that is how you open the virtual HMI where you can edit the values for testing purposes.

Further analysis of data collected in the “log” table in the “ofkrse.db” database is possible. Sample scripts for Statistical Process Control and sample output graphs can be found are under:

hmi/vnc/m000/archive/sqlite3/spc/

Hmi/vnc/m000/archive/sqlite3/

5.1.5 Alert functionality

Python function, send_email, which sends an email alert when certain conditions are met. Here's a brief explanation of its components

Functionality

1. Extracting Values from Alert:
    a. The function parses alert_from_call to extract the current reading value (value), minimum threshold (min_val), and maximum threshold (max_val).
    b. It checks if value is less than min_val or greater than max_val to determine the alert message.
2. Composing the Email:
    a. Sets up the sender's email (sender_email), the Gmail app password (sender_password), and the receiver's email (receiver_email).
    b. Constructs the email subject and body, which include the segment name and a description of the threshold breach.
3. Creating the Email Message:
    a. Uses email. Mime modules to construct the email with a text body in plain format.
4. Sending the Email:
    a. Connects to Gmail's SMTP server (smtp.gmail.com) on port 587.
    b. Starts a secure connection using starttls().
    c. Logs in to the sender's email using the provided credentials.
    d. Sends the composed email using server.send_message(msg).
5. Error Handling:
    a. If there's an exception during the email-sending process, it catches the error and prints the error message.

Potential Improvements or Notes

- **Security**:
    - The sender's Gmail app password is hardcoded, which is insecure. Using environment variables or a secure credentials management system is recommended.
- **Receiver Email**:
    - The receiver's email (receiver_email) is set for testing. In a production environment, this should be dynamically set.
- **String Parsing**:
    - The code assumes alert_from_call follows a strict format (with specific words and numeric indices). Validation should be added to handle malformed inputs.

6. Alternative Usage Scenarios

None

7. Troubleshooting

Error messages printed to console can guide this process.

7.1 Common Issues

One issue to be addressed in the future is the functionality that determines the color of an indicator light. What we may perceive as “red” is sometimes perceived and labelled as “orange” and sometimes as “brown.” The python scripts may need to be modified to convert these three value to “red,” before recording this to the log file.

To further address this issue, it may be beneficial to refers to the official jpg documentation regarding colors:

https://www.w3.org/Graphics/JPEG/jfif3.pdf

7.2 Error Messages and Resolutions

Depending on the error message, the remedy is often not complicated.

7.3 Performance Optimization

None.

8. Maintenance

Periodically backup data collected in the ofkrse.db sqlite3 database for future data mining.

8.1 System Updates

Depending on further development.

8.2 Backup Procedures

One easy option for backing up the database table using the Linux Bash commandline is as follows:

```
$ sqlite3 ofkrse.db

.output log_backup.sql

.dump log

.quit
```

8.3 Performance Monitoring

Periodic monitoring of the memory usage of this and any other software using the command line tool “top” is recommended:

```
$ top
```

9. Technical Specifications

9.1 Hardware Requirements

AMD Ryzen, Intel Core i7, or better. Although the system is likely to perform well on less powerful legacy hardware as well.

9.2 Software Requirements

Debian Linux or other distribution, or Windows 11

Python3

Gimp

9.3 Network Requirements

LAN or Wi-Fi

9.4 Performance Specifications

On an Intel Core i7, the running time of the running time of the core python process:

```
$ time pys/050___READ_ALL_v01.py

real	0m14.074s

user	0m18.061s

sys	0m7.112s
```

9.5 Data Handling Capabilities

The system has been tested on screenshots with the following resolution: 1920x1040

Processing higher resolution screenshots will likely be more resource-intensive.

10. Safety and Compliance

It is recommended that in the alpha-phase of testing only non-critical industrial equipment are monitored using this software, so that any software failure and resultant industrial equipment malfunction would not pose ANY human safety risks.

10.1 Safety Warnings

None other precautions provided by the manufacturer of the industrial equipment being monitored.

10.2 Regulatory Compliance

Before deploying this software to European customers, the compliance of this software with the European Union General Data Protection Regulation, the GDPR, should be further explored.

10.3 Data Privacy Considerations

None at this stage, unless the monitored industrial equipment is used to collect data that is covered under the "Privacy Act of 1974" or "HIPPA" and other applicable state and federal statutes.

11. Support and Contact Information

General queries: Dr. Haoyu Wang

11.1 Technical Support

Ryan Sharp, Robotics Lab Graduate Intern, until May 2025; then Dr. Haoyu Wang.

11.2 Reporting Issues

Dr. Haoyu Wang

12. Appendices

12.1 Glossary of Terms

HMI: Human Machine Interface

12.2 Abbreviations

VNC: Virtual Network Computing

12.3 Reference Documents

None.

12.4 Version History

12/3/2024: Version 1.0