Steps for Hosting VIFF's PDF Maker

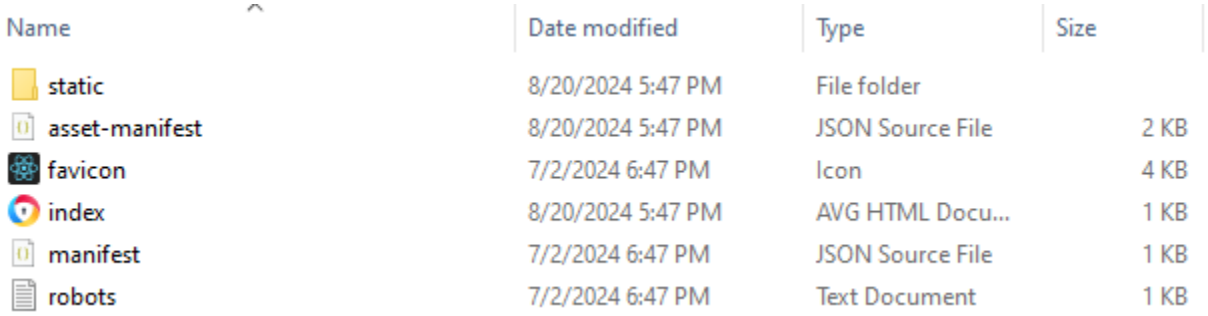
Version 1.0 Created September 9, 2024

1. You must have the codebase saved on your computer, and Node.js installed. As of September 2024, the latest version of the program can be found at: https://github.com/leanneholmes/VIFF-Schedule-Generator

2. From a terminal within the main directory of the codebase, run this command:

```
npm run build
```

   This will output the static web files needed to host in a folder called **/build**. In some cases, it may be named **/dist.** You should see the following files inside this folder:



Name	Date modified	Type	Size
static	8/20/2024 5:47 PM	File folder	
asset-manifest	8/20/2024 5:47 PM	JSON Source File	2 KB
favicon	7/2/2024 6:47 PM	Icon	4 KB
index	8/20/2024 5:47 PM	AVG HTML Docu...	1 KB
manifest	7/2/2024 6:47 PM	JSON Source File	1 KB
robots	7/2/2024 6:47 PM	Text Document	1 KB

3. The VM has been created by ITS Consulting. You will need the Server IP address from them, as well as an OpenSSH server for you to login and upload the web files. For the 2024 deployment, the following information was provided:

```
Ubuntu 24.04 LTS
Server name: VSVIFFPDF
Server IP: 192.168.175.30

OpenSSH server was installed and tested, can be accessed external
via 216.19.177.132:17324
```

   Be sure to update the IP addresses and ports in the commands accordingly if you are deploying, as they will likely change. From the info above, 17324 is the port number.

   ITS also created user credentials named **bcit** with a password that will be shared securely via a vault link. These credentials will be on a timed expiry, deployment must be completed before this.

4. Open a Windows Powershell terminal and login via SSH:

```
ssh bcit@216.19.177.132 -p 17324
```

   Type in the password when prompted and hit Enter. Remember that nothing will appear in the terminal for any letter that you type in a password field; be careful of typos.

5. Install and enable Nginx using the apt package manager:

```
sudo apt update
sudo apt install nginx
sudo systemctl enable nginx
```

   Nginx should start automatically, but you can start it manually to make sure it's running:

```
sudo systemctl start nginx
```

   This command can be used to verify that Nginx is active by checking its status:

```
sudo systemctl status nginx
```

6. Create a directory named viffpdf for the web files, and change its ownership:

```
sudo mkdir /var/www/html/viffpdf
sudo chmod -R 777 /var/www/html/viffpdf
```

7. Check the path on your local machine to where the **/build** files were created. For me, it was in C:/Users/Leanne/Projects/VIFF-Schedule-Generator/build. If there are any spaces in the directory names, rename or move them.

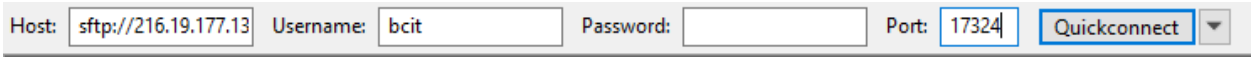
   Type 'logout' to end the ssh session. Then use this command to securely copy your build files to the new directory on the virtual machine:

```
scp -P 17324 -r C:/Users/Leanne/Projects/VIFF-Schedule-
Generator/build bcit@216.19.177.132:/var/www/html/viffpdf/
```

   It will prompt you for the bcit password. After completing, log back in over SSH and navigate to the directory. This will list the contents for you to double check they transferred successfully:

```
ssh bcit@216.19.177.132 -p 17324
cd /var/www/html/viffpdf/
ls
```

If you prefer a graphical interface, you can install FileZilla and login via sftp. See below for what to enter in the Host field.



You'll be able to view the VM files like a normal directory, and simply drag and drop your build files into /var/www/html/viffpdf.

8. Create an Nginx configuration file for the webpage.

```
sudo nano /etc/nginx/conf.d/viffpdf.conf
```

This will open the nano text editor. Paste in the following text:

```
server {
    listen 80;
    listen [::]:80;
    root /var/www/html/viffpdf/;
    index index.html;
    server_name 192.168.175.30;
    location / {
        try_files $uri $uri/ =404;
    }
}
```

Save your changes and close the file. (Ctrl+O, Enter, Ctrl+X). Verify that there are no Nginx configuration errors:

```
sudo nginx -t
```

9. Restart the Nginx service to apply your changes.

```
sudo systemctl restart nginx
```

The site should now be running at http://192.168.175.30.

Troubleshooting

If the site goes down, login via SSH and check the Nginx status for the error message.

```
sudo systemctl status nginx
```