Task 1:

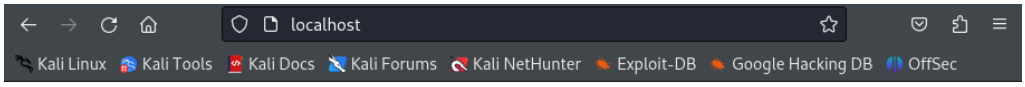
Installing Apache

```
┌──(aleyna㉿kali)-[~]
└─$ sudo apt update
[sudo] password for aleyna:
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [19.9 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [47.4 MB]
Get:4 http://kali.download/kali kali-rolling/contrib amd64 Packages [113 kB]
Get:5 http://kali.download/kali kali-rolling/contrib amd64 Contents (deb) [271 kB]
Get:6 http://kali.download/kali kali-rolling/non-free amd64 Packages [192 kB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Contents (deb) [862 kB]
Get:8 http://kali.download/kali kali-rolling/non-free-firmware amd64 Packages [33.1 kB]
Get:9 http://kali.download/kali kali-rolling/non-free-firmware amd64 Contents (deb) [16.9 kB]
Fetched 68.8 MB in 22s (3182 kB/s)
514 packages can be upgraded. Run 'apt list --upgradable' to see them.

┌──(aleyna㉿kali)-[~]
└─$ sudo apt install apache2
apache2 is already the newest version (2.4.59-2).
apache2 set to manually installed.
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 514
```

Creating Webpage

```
┌──(aleyna㉿kali)-[~]
└─$ sudo nano /var/www/html/index.html

┌──(aleyna㉿kali)-[~]
└─$ sudo systemctl start apache2
[sudo] password for aleyna:
```

```
<!DOCTYPE html>
<html lang="en">
<head>
        <meta charset="UTF-8">
        <meta name="viewport" content = "width=device-width, initial-scale=1.0">
        <title>Welcome</title>
</head>
<body>
        <h1>Welcome to my webpage!</h1>
        <p>Course: Network Protocols Security - CS 646 850</p>
        <p>Name: Aleyna Aydin 31963811</p>
</body>
</html>
```



Welcome to my webpage!

Course: Network Protocols Security - CS 646 850

Name: Aleyna Aydin 31963811

Task 2:

Generating SSL/TLS Certificate

```
┌──(aleyna㉿kali)-[~]
└─$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/selfsigned.key
-out /etc/ssl/certs/selfsigned.crt

[sudo] password for aleyna:
....+.+.....+++++++++++++++++++++++++++++++++++++++*.+ ... +++++++++++++++++++++++++++++++++++++++*..
.............+....+........+.+......+........+.......+...........+ ... +.+..............+.+ ... + ... + ...
..+.......+ ... + ... +........+..+....+..+.........+ ... +..........+.................+..........
.+ ... +....+.....+....+..+ ... +......+ ... +.........+..+....+......+............+.+..............
... + ... +.++++++
....+..............+.........+.....+.+..........++++++++++++++++++++++++++++++++++++++++*.....+ ...
+.+.....+...........+......+.........+ ... + .. +.......+........+++++++++++++++++++++++++++++++++++++++
++* ... +......+ ... + .. + ... +.........+.......+ ... + .. +....+..+.........+..+......+.......+........+....
.....+.+.....+.+ ... + .. +.+........+.+......+.....+.........+.+..+.......+ ... + .. + ... +.+.....+.......+.
.+ ... +.......+.....+.+..............+ ... +............+...........+............+ ... +.....+.........
....+ .. + ... +.......+........+.+........+.+...........+.+ .. +.+......+......+.....+ ... +......+....+ ...
...........+.+.........+.....+......+....+.....+......+.+ ... + ... + .. +.............+ ... + .. +....+ ... + ..
.+.....+.......+.........+.........+ ... +.+.........+...........+....+...........+.....+....+.........
....+ ... + .. +.......+ .. +.....+ ... +.+.........+.........+ .. +.+ .. +..............+ ... +..........+ .. +.+
..............+............+ .. +.+ .. +.........+........+.+.....+.+ ... +.....+...........+....+.....
... +.+.......................+ ... + .. +..............++++++
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:New Jersey
Locality Name (eg, city) []:Newark
Organization Name (eg, company) [Internet Widgits Pty Ltd]:NJIT
Organizational Unit Name (eg, section) []:CS 646
Common Name (e.g. server FQDN or YOUR name) []:10.0.2.15
Email Address []:aga32@njit.edu
```

Config File for SSL

```
┌──(aleyna㉿kali)-[~]
└─$ sudo nano /etc/apache2/sites-available/default-ssl.conf
```

```
 GNU nano 8.0                 /etc/apache2/sites-available/default-ssl.conf
        #   issuer chain before deciding the certificate is not valid.
        #SSLVerifyClient require
        #SSLVerifyDepth  10

        #   SSL Engine Options:
        #   Set various options for the SSL engine.
        #   o FakeBasicAuth:
        #    Translate the client X.509 into a Basic Authorisation.  This means that
        #    the standard Auth/DBMAuth methods can be used for access control.  The
        #    user name is the `one line' version of the client's X.509 certificate.
        #    Note that no password is obtained from the user. Every entry in the user
        #    file needs this password: `xxj31ZMTZzkVA'.
        #   o ExportCertData:
        #    This exports two additional environment variables: SSL_CLIENT_CERT and
        #    SSL_SERVER_CERT. These contain the PEM-encoded certificates of the
        #    server (always existing) and the client (only existing when client
        #    authentication is used). This can be used to import the certificates
        #    into CGI scripts.
        #   o StdEnvVars:
        #    This exports the standard SSL/TLS related `SSL_*' environment variables.
        #    Per default this exportation is switched off for performance reasons,
        #    because the extraction step is an expensive operation and is usually
        #    useless for serving static content. So one usually enables the
        #    exportation for CGI and SSI requests only.
        #   o OptRenegotiate:
        #    This enables optimized SSL connection renegotiation handling when SSL
        #    directives are used in per-directory context.
        #SSLOptions +FakeBasicAuth +ExportCertData +StrictRequire
        <FilesMatch "\.(cgi|shtml|phtml|php)$">
                SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
                SSLOptions +StdEnvVars
        </Directory>
        BrowserMatch "MSIE [2-6]" \
                nokeepalive ssl-unclean-shotdown \
                downgrade-1.0 force-response-1.0
</VirtualHost>

                                [ Wrote 104 lines ]
```

Task 3:

Configuring Web Server

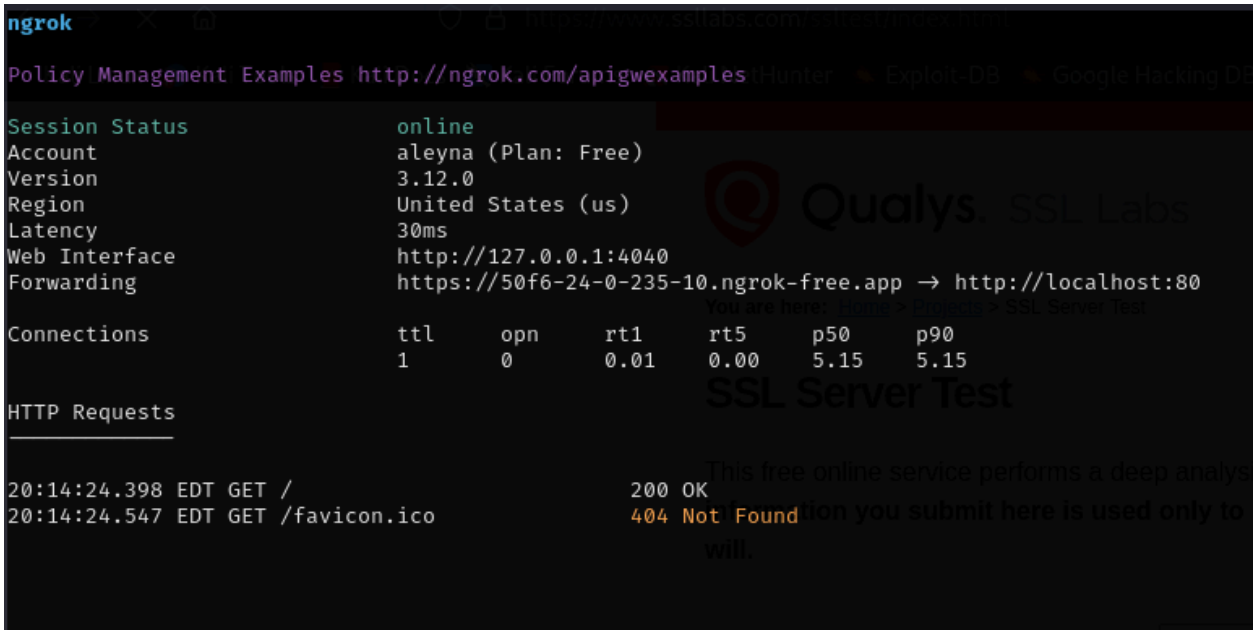
```
┌──(aleyna㉿kali)-[~]
└─$ sudo a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2

┌──(aleyna㉿kali)-[~]
└─$ sudo a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  systemctl reload apache2

┌──(aleyna㉿kali)-[~]
└─$ sudo systemctl restart apache2
```

Task 4:

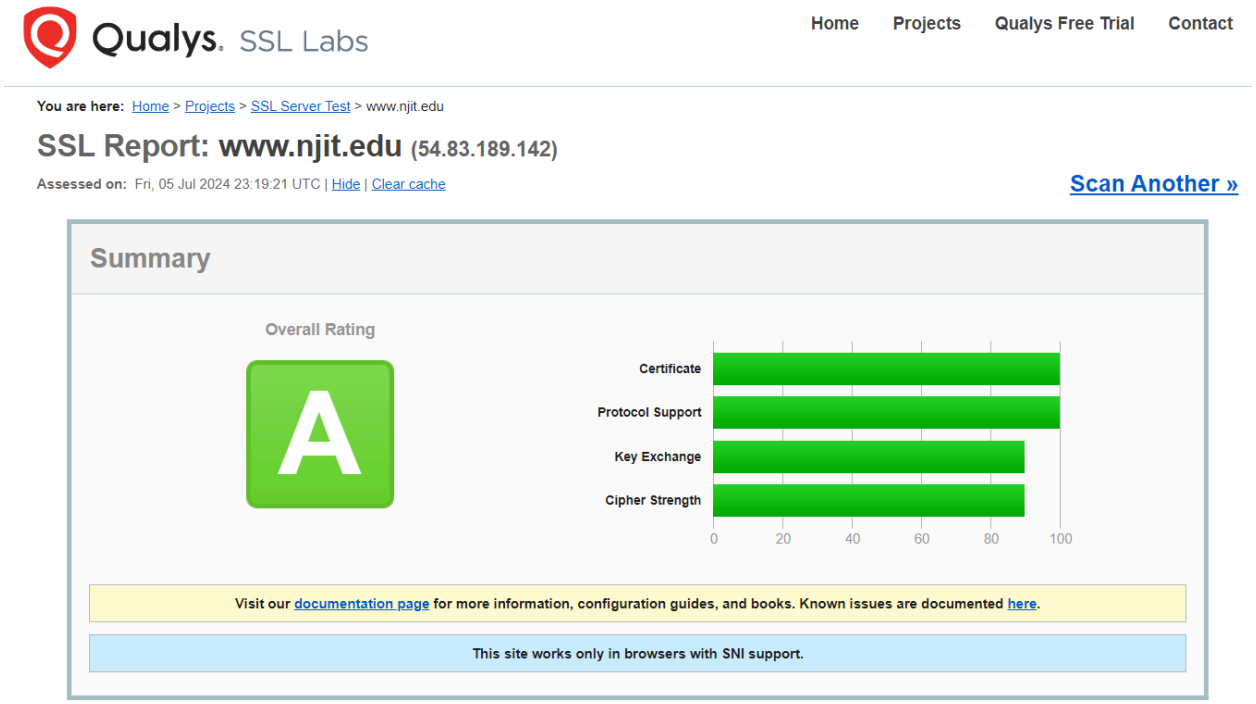
SSL/TLS Configuration

Installing and using ngrok to get web page domain name to search on SSL Labs

```
┌──(aleyna㉿kali)-[~]
└─$ sudo tar xvzf ~/Downloads/ngrok-v3-stable-linux-amd64.tgz -C /usr/local/bin
ngrok

┌──(aleyna㉿kali)-[~]
└─$ curl -s https://ngrok-agent.s3.amazonaws.com/ngrok.asc | sudo tee /etc/apt/trusted.gpg.d/ngrok.asc >/dev/n
ull && echo "deb https://ngrok-agent.s3.amazonaws.com buster main" | sudo tee /etc/apt/sources.list.d/ngrok.li
st && sudo apt update && sudo apt install ngrok
deb https://ngrok-agent.s3.amazonaws.com buster main
Get:1 https://ngrok-agent.s3.amazonaws.com buster InRelease [20.3 kB]
Hit:2 http://http.kali.org/kali kali-rolling InRelease
Get:3 https://ngrok-agent.s3.amazonaws.com buster/main amd64 Packages [4856 B]
Get:4 https://ngrok-agent.s3.amazonaws.com buster/main amd64 Contents (deb) [78 B]
Fetched 25.2 kB in 1s (30.1 kB/s)
514 packages can be upgraded. Run 'apt list --upgradable' to see them.
Installing:
  ngrok

Summary:
  Upgrading: 0, Installing: 1, Removing: 0, Not Upgrading: 514
  Download size: 6507 kB
  Space needed: 0 B / 9651 MB available

Get:1 https://ngrok-agent.s3.amazonaws.com buster/main amd64 ngrok amd64 3.12.0 [6507 kB]
Fetched 6507 kB in 2s (4004 kB/s)
Selecting previously unselected package ngrok.
(Reading database ... 390905 files and directories currently installed.)
Preparing to unpack .../ngrok_3.12.0_amd64.deb ...
Unpacking ngrok (3.12.0) ...
Setting up ngrok (3.12.0) ...

┌──(aleyna㉿kali)-[~]
└─$ ngrok config add-authtoken 2iqi66ReGv4NI4sh95HuKuOyqEk_r8DmBRMx9GBqEQ1uWWC
Authtoken saved to configuration file: /home/aleyna/.ngrok2/ngrok.yml
Default version saved to configuration file: /home/aleyna/.ngrok2/ngrok.yml

┌──(aleyna㉿kali)-[~]
└─$ ngrok http 80
```



NJIT:



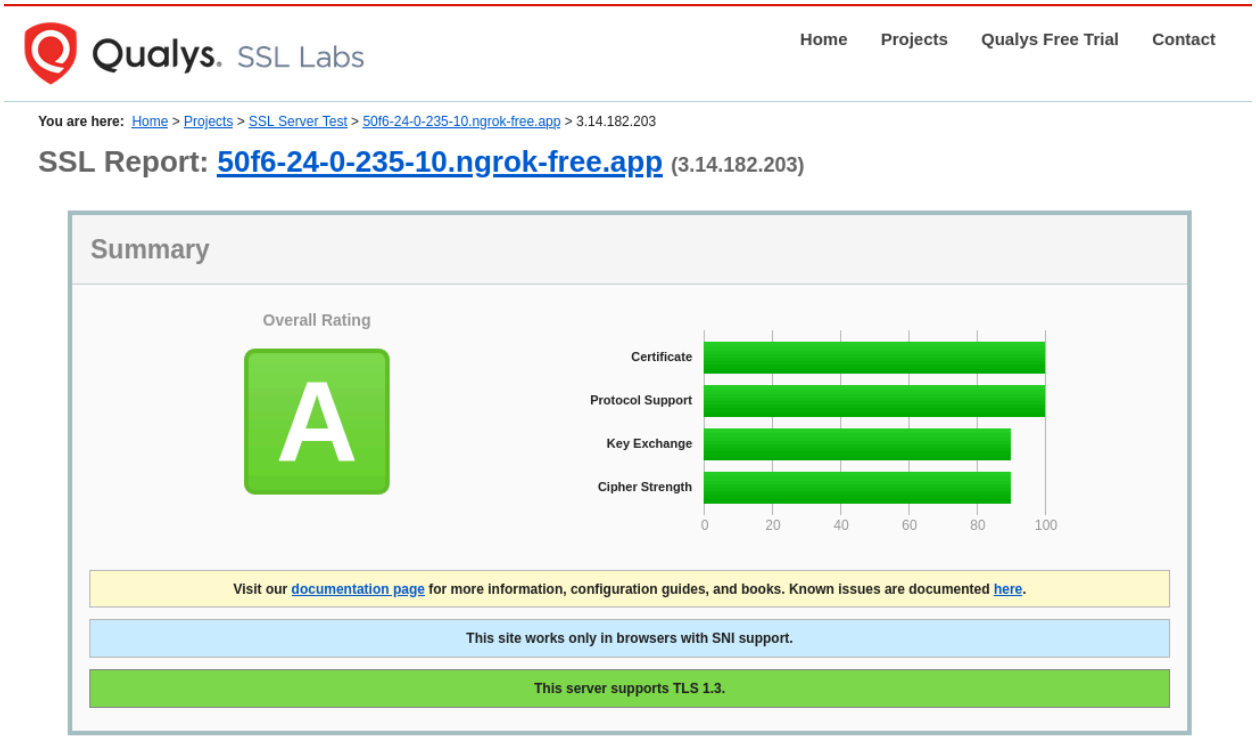
Protocol support: TLS 1.3, 1.2, 1.1, 1.0; SSL 3, 2
Cipher Suites and Strength:
TLS 1.2

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH secp256r1 (eq. 3072 bits RSA) FS 128

TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) ECDH secp256r1 (eq. 3072 bits RSA) FS **WEAK**	128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH secp256r1 (eq. 3072 bits RSA) FS	256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) ECDH secp256r1 (eq. 3072 bits RSA) FS **WEAK**	256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) **WEAK**	128
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c) **WEAK**	128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d) **WEAK**	256
TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d) **WEAK**	256

Key Exchange: RSA 2048 bits (certificates 1/2)
No vulnerabilities and both have security feature HSTS

Web Page:



Protocol Support: TLS 1.3, 1.2, 1.1, 1.0; SSL 3, 2
Cipher Strength:
TLS 1.3

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH secp256r1 (eq. 3072 bits RSA) FS		128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) ECDH secp256r1 (eq. 3072 bits RSA) FS **WEAK**		128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH secp256r1 (eq. 3072 bits RSA) FS		256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) ECDH secp256r1 (eq. 3072 bits RSA) FS **WEAK**		256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) **WEAK**		128
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c) **WEAK**		128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d) **WEAK**		256
TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d) **WEAK**		256

TLS 1.2

TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) ECDH secp256r1 (eq. 3072 bits RSA) FS		128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027) ECDH secp256r1 (eq. 3072 bits RSA) FS **WEAK**		128
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030) ECDH secp256r1 (eq. 3072 bits RSA) FS		256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028) ECDH secp256r1 (eq. 3072 bits RSA) FS **WEAK**		256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0x9c) **WEAK**		128
TLS_RSA_WITH_AES_128_CBC_SHA256 (0x3c) **WEAK**		128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0x9d) **WEAK**		256
TLS_RSA_WITH_AES_256_CBC_SHA256 (0x3d) **WEAK**		256

Key Exchange: EC 256 bits (certificate 3), RSA 2048 bits (certificates 1/2)
No vulnerabilities and both have security feature HSTS