Group 3

Lazaro Loureiro, Catalina Perez, Deividas Ilgunas, Allison Brown

WEB Server Configuration

(CRN 86201) COP3350 – Systems Administration & Programming

09 December 2024

Team Member Contributions

Deividas Ilgunas – Did the DNS server conf and set up zones, also troubleshooted when DNS wasn't working.

Catalina Perez – Did WEB server and HTML code

Allison Brown – Helped with the DNS file configuration and troubleshooting, a lot of work with VM network settings to get working IP.

Lazaro Loureiro – HTML code and some troubleshooting

Introduction

For our group project, we have chosen to work with WEB server configuration, which will involve setting up a secure and optimized environment for hosting websites. As part of this task, we will also have to configure a DNS server to ensure seamless domain name resolution and functionality. This project provides an excellent opportunity to apply and enhance the skills in Linux server management that we have learned throughout the semester. By collaborating closely, we aim to demonstrate our understating of key concepts learned in class and showcase our ability to implement them in a real-world scenario.

Task 1: DNS Server Configuration

Install

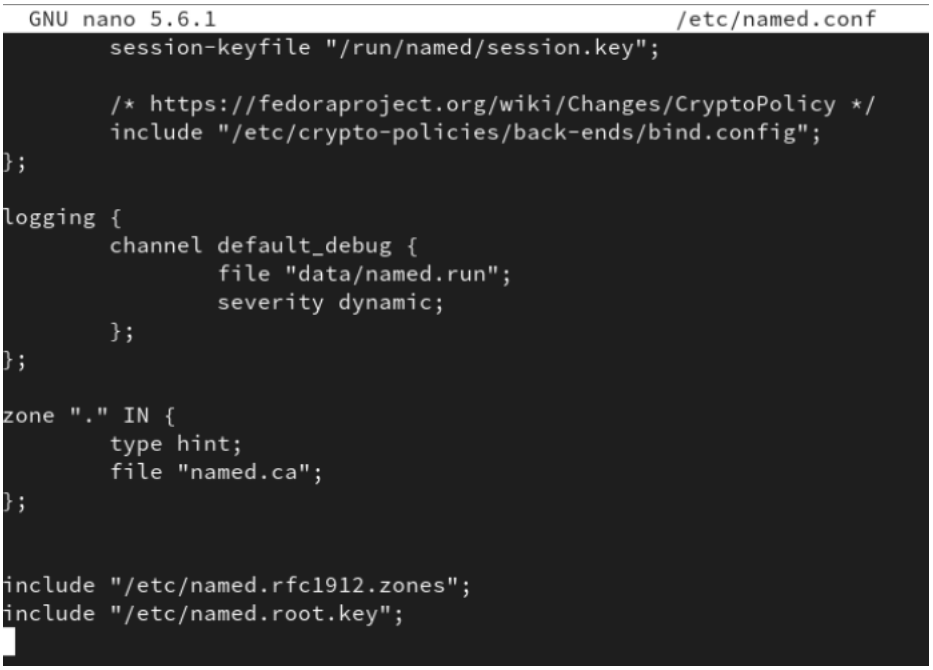
To begin, we must install the necessary packages to create and run our DNS service, this will include BIND packages.

Commands used: sudo yum install bind* -y

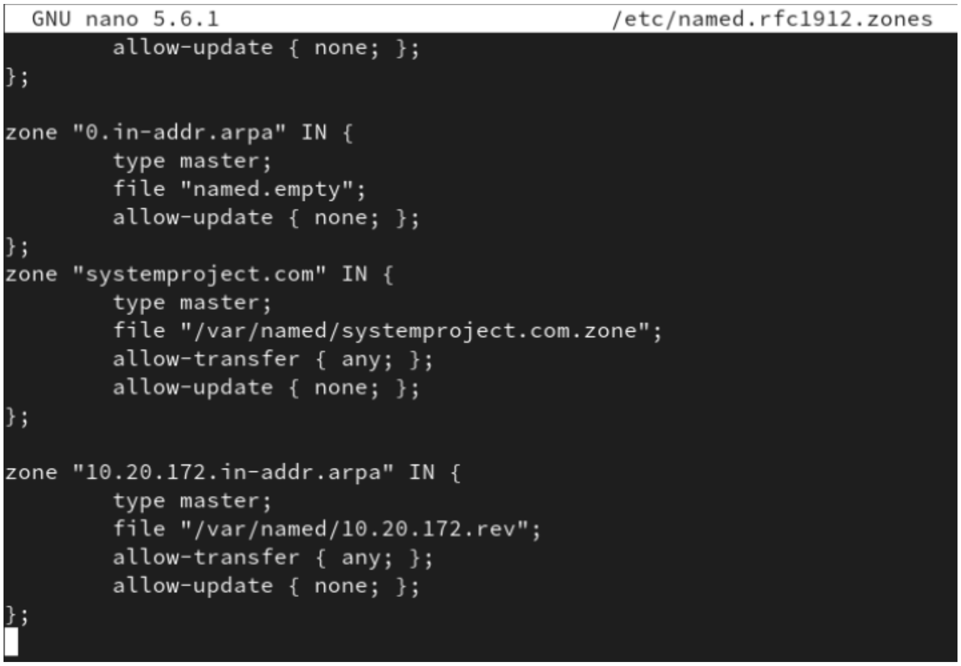
Config

To configure the DNS server, we first configure the /etc/named.conf file, mostly checking if the default file is correct, I changed some things such as instead of listening to a certain IP address it finds any. Once configured we move onto the /etc/named.rfc1912.zones to configure our forward and reverse zones. We wrote two new blocks of code with systemproject.com as our ip forward and 10.20.172.in-addr.arpa as our reverse zone. We allowed any queries and no auto update. Now we just program both the forward and reverse zones. These were pretty straightforward codes. Once each zone's had their code and everything was configured, I made sure each file had the right permissions for each file. I also made sure the name owned the files as well. After we were ready to enable and start the server. I also did some minor firewall configuring making sure that it will work.

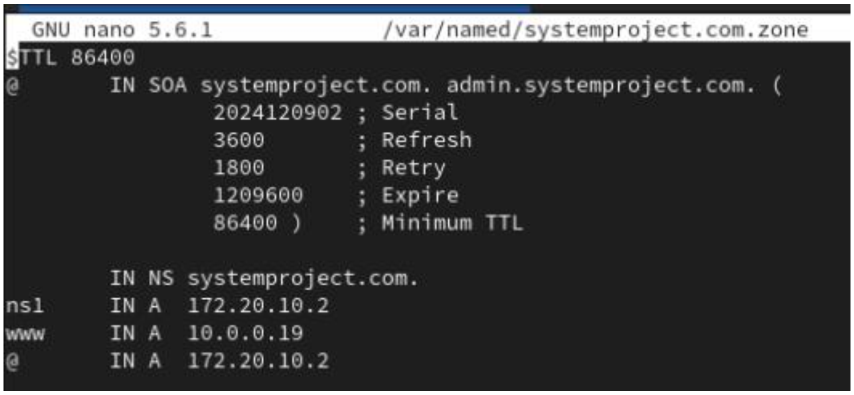
Commands used: sudo nano /etc/named.conf | /etc/named.rfc1912.zones | /var/named/systemproject.com.zone | /var/named/10.20.172.rev, sudo chmod 755 /etc/named.conf | /etc/named.rfc1912.zones | /var/named/systemproject.com.zone, chown named:named /etc/named.conf | /etc/named.rfc1912.zones | /var/named/systemproject.com.zone, sudo systemctl enable named, sudo systemctl start named, sudo firewall-cmd --add-port=53/tcp --permanent | --add-port=53/udp --permanent | --reload



```
  GNU nano 5.6.1                                 /etc/named.conf
        session-keyfile "/run/named/session.key";

        /* https://fedoraproject.org/wiki/Changes/CryptoPolicy */
        include "/etc/crypto-policies/back-ends/bind.config";
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {
        type hint;
        file "named.ca";
};


include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

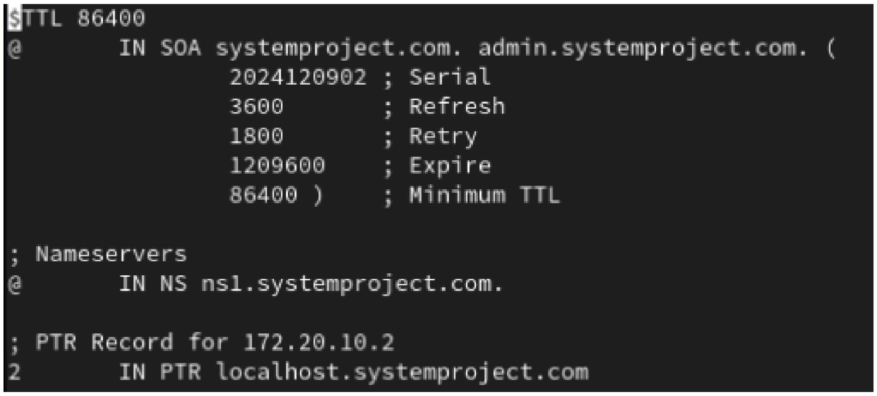


```
  GNU nano 5.6.1                           /etc/named.rfc1912.zones
        allow-update { none; };
};

zone "0.in-addr.arpa" IN {
        type master;
        file "named.empty";
        allow-update { none; };
};
zone "systemproject.com" IN {
        type master;
        file "/var/named/systemproject.com.zone";
        allow-transfer { any; };
        allow-update { none; };
};

zone "10.20.172.in-addr.arpa" IN {
        type master;
        file "/var/named/10.20.172.rev";
        allow-transfer { any; };
        allow-update { none; };
};
```

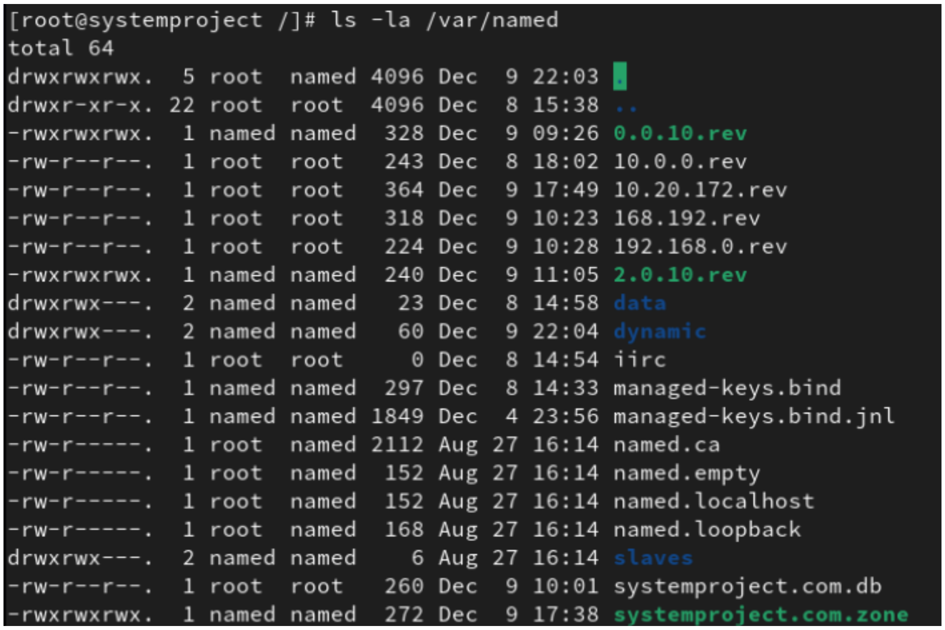


```
  GNU nano 5.6.1              /var/named/systemproject.com.zone
$TTL 86400
@       IN SOA systemproject.com. admin.systemproject.com. (
                2024120902 ; Serial
                3600       ; Refresh
                1800       ; Retry
                1209600    ; Expire
                86400 )    ; Minimum TTL

        IN NS systemproject.com.
ns1     IN A  172.20.10.2
www     IN A  10.0.0.19
@       IN A  172.20.10.2
```

Below: /var/named/10.20.172.rev



```
$TTL 86400
@       IN SOA systemproject.com. admin.systemproject.com. (
                2024120902 ; Serial
                3600       ; Refresh
                1800       ; Retry
                1209600    ; Expire
                86400 )    ; Minimum TTL

; Nameservers
@       IN NS ns1.systemproject.com.

; PTR Record for 172.20.10.2
2       IN PTR localhost.systemproject.com
```

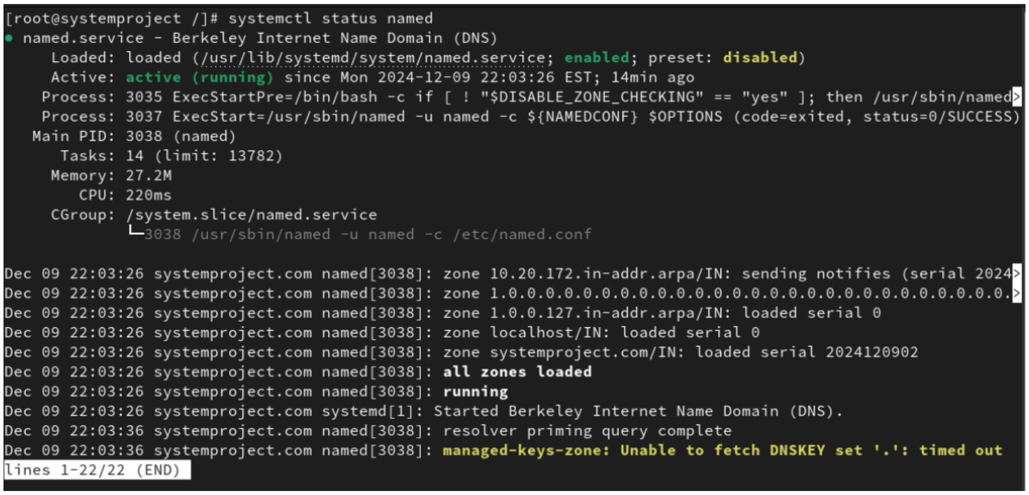


```
[root@systemproject /]# ls -la /var/named
total 64
drwxrwxrwx.  5 root  named 4096 Dec  9 22:03 .
drwxr-xr-x. 22 root  root  4096 Dec  8 15:38 ..
-rwxrwxrwx.  1 named named  328 Dec  9 09:26 0.0.10.rev
-rw-r--r--.  1 root  root   243 Dec  8 18:02 10.0.0.rev
-rw-r--r--.  1 root  root   364 Dec  9 17:49 10.20.172.rev
-rw-r--r--.  1 root  root   318 Dec  9 10:23 168.192.rev
-rw-r--r--.  1 root  root   224 Dec  9 10:28 192.168.0.rev
-rwxrwxrwx.  1 named named  240 Dec  9 11:05 2.0.10.rev
drwxrwx---.  2 named named   23 Dec  8 14:58 data
drwxrwx---.  2 named named   60 Dec  9 22:04 dynamic
-rw-r--r--.  1 root  root     0 Dec  8 14:54 iirc
-rw-r--r--.  1 named named  297 Dec  8 14:33 managed-keys.bind
-rw-r--r--.  1 named named 1849 Dec  4 23:56 managed-keys.bind.jnl
-rw-r-----.  1 root  named 2112 Aug 27 16:14 named.ca
-rw-r-----.  1 root  named  152 Aug 27 16:14 named.empty
-rw-r-----.  1 root  named  152 Aug 27 16:14 named.localhost
-rw-r-----.  1 root  named  168 Aug 27 16:14 named.loopback
drwxrwx---.  2 named named    6 Aug 27 16:14 slaves
-rw-r--r--.  1 root  root   260 Dec  9 10:01 systemproject.com.db
-rwxrwxrwx.  1 named named  272 Dec  9 17:38 systemproject.com.zone
```

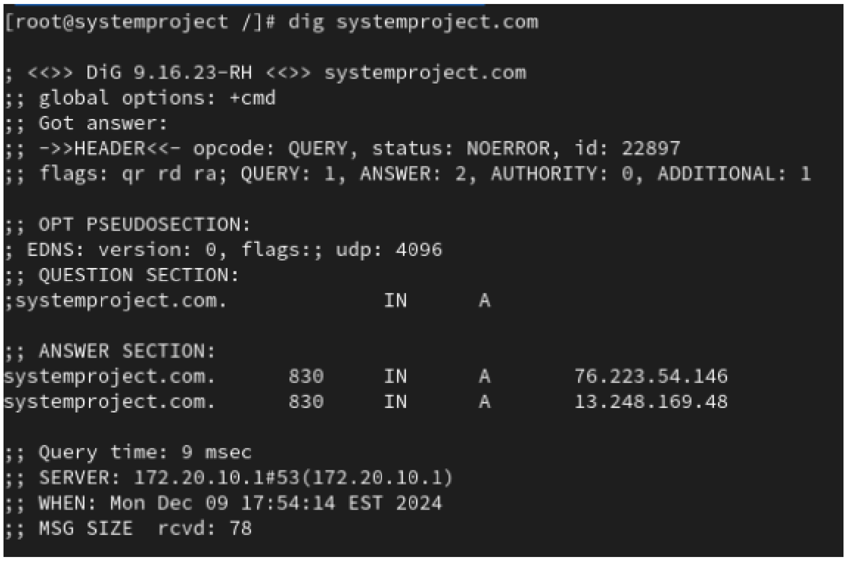
Testing

Now we check if the DNS server is up by checking the status. If all zones have loaded and it says its active that means we are good to go. We further test each zone by digging their ip addresses and domain addresses. They all came back with a proper response and gave us a correct answer section with the url and ip next to them. That confirms we are good to continue with our WEB server.

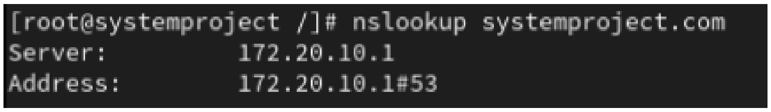
Commands used: sudo systemctl status named, dig @127.0.0.1 systemproject.com | @127.0.0.1 -x 172.20.10.2 | systemproject.com | www.systemproject.com, nslookup systemproject.com



```
[root@systemproject /]# systemctl status named
● named.service - Berkeley Internet Name Domain (DNS)
     Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: disabled)
     Active: active (running) since Mon 2024-12-09 22:03:26 EST; 14min ago
    Process: 3035 ExecStartPre=/bin/bash -c if [ ! "$DISABLE_ZONE_CHECKING" == "yes" ]; then /usr/sbin/named>
    Process: 3037 ExecStart=/usr/sbin/named -u named -c ${NAMEDCONF} $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 3038 (named)
      Tasks: 14 (limit: 13782)
     Memory: 27.2M
        CPU: 220ms
     CGroup: /system.slice/named.service
             └─3038 /usr/sbin/named -u named -c /etc/named.conf

Dec 09 22:03:26 systemproject.com named[3038]: zone 10.20.172.in-addr.arpa/IN: sending notifies (serial 2024>
Dec 09 22:03:26 systemproject.com named[3038]: zone 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.>
Dec 09 22:03:26 systemproject.com named[3038]: zone 1.0.0.127.in-addr.arpa/IN: loaded serial 0
Dec 09 22:03:26 systemproject.com named[3038]: zone localhost/IN: loaded serial 0
Dec 09 22:03:26 systemproject.com named[3038]: zone systemproject.com/IN: loaded serial 2024120902
Dec 09 22:03:26 systemproject.com named[3038]: all zones loaded
Dec 09 22:03:26 systemproject.com named[3038]: running
Dec 09 22:03:26 systemproject.com systemd[1]: Started Berkeley Internet Name Domain (DNS).
Dec 09 22:03:36 systemproject.com named[3038]: resolver priming query complete
Dec 09 22:03:36 systemproject.com named[3038]: managed-keys-zone: Unable to fetch DNSKEY set '.': timed out
lines 1-22/22 (END)
```



```
[root@systemproject /]# dig systemproject.com

; <<>> DiG 9.16.23-RH <<>> systemproject.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22897
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;systemproject.com.             IN      A

;; ANSWER SECTION:
systemproject.com.      830     IN      A       76.223.54.146
systemproject.com.      830     IN      A       13.248.169.48

;; Query time: 9 msec
;; SERVER: 172.20.10.1#53(172.20.10.1)
;; WHEN: Mon Dec 09 17:54:14 EST 2024
;; MSG SIZE  rcvd: 78
```

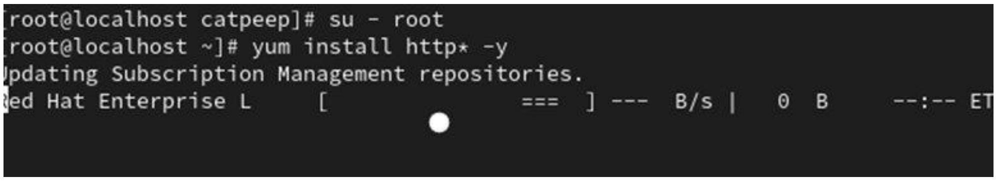


```
[root@systemproject /]# nslookup systemproject.com
Server:         172.20.10.1
Address:        172.20.10.1#53
```

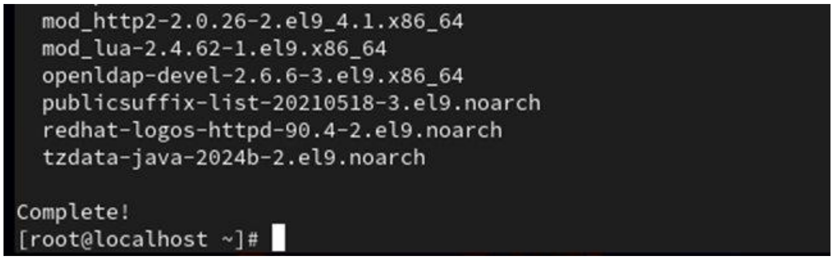
Task 2: WEB Server Configuration

Install

Here we installed the necessary software packages to get the web server started

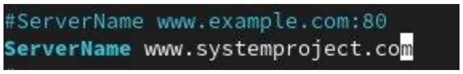


```
[root@localhost catpeep]# su - root
[root@localhost ~]# yum install http* -y
Updating Subscription Management repositories.
Red Hat Enterprise L     [              ===  ] ---  B/s |   0  B     --:-- ET
```



```
  mod_http2-2.0.26-2.el9_4.1.x86_64
  mod_lua-2.4.62-1.el9.x86_64
  openldap-devel-2.6.6-3.el9.x86_64
  publicsuffix-list-20210518-3.el9.noarch
  redhat-logos-httpd-90.4-2.el9.noarch
  tzdata-java-2024b-2.el9.noarch

Complete!
[root@localhost ~]#
```

Here we created our server's name after editing the httpd.conf



```
#ServerName www.example.com:80
ServerName www.systemproject.com
```

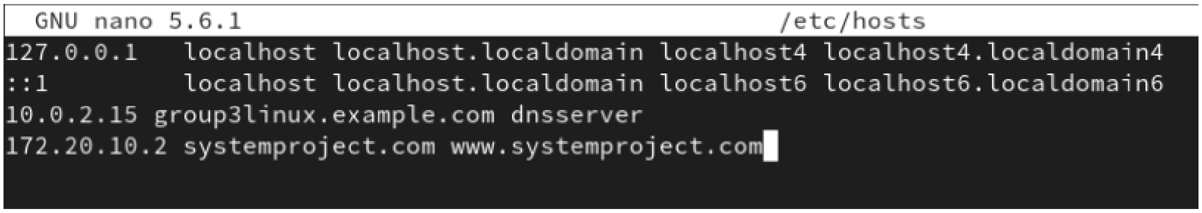
Commands used: sudo yum install httpd –y

Config

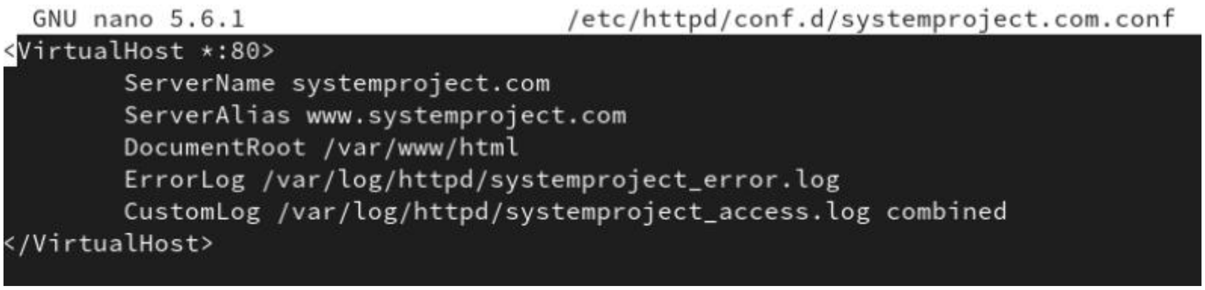
To configure the WEB server, we first had to cd into the conf folder using “cd /etc/httpd/conf”.

Then, we had to edit, httpd.conf with nano. In this file, we needed to add our server’s name, www.systemproject.com, and the IP address, 172.20.10.2. Now we can finally enable and start the httpd. After, we moved on to our HTML code.

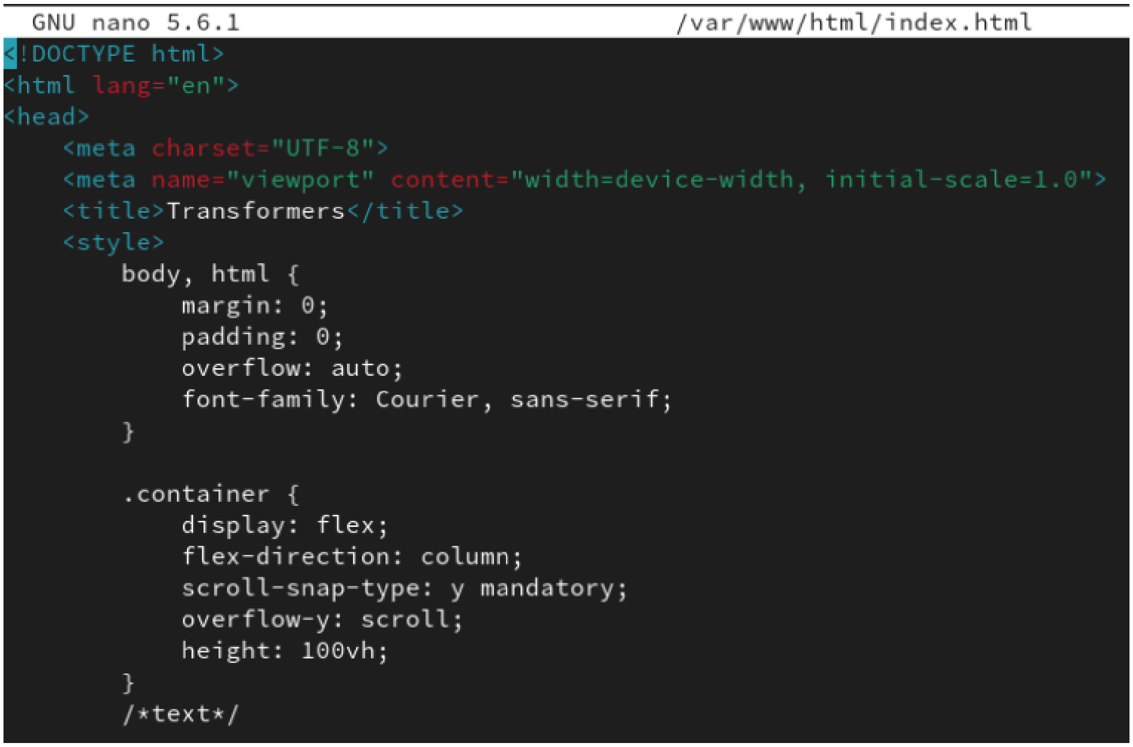
Commands used: sudo nano /etc/hosts | /etc/httpd/conf.d/systemproject.com.conf | /var/www/html/index.html | /etc/httpd/conf/httpd.conf



```
  GNU nano 5.6.1                                    /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.2.15 group3linux.example.com dnsserver
172.20.10.2 systemproject.com www.systemproject.com
```

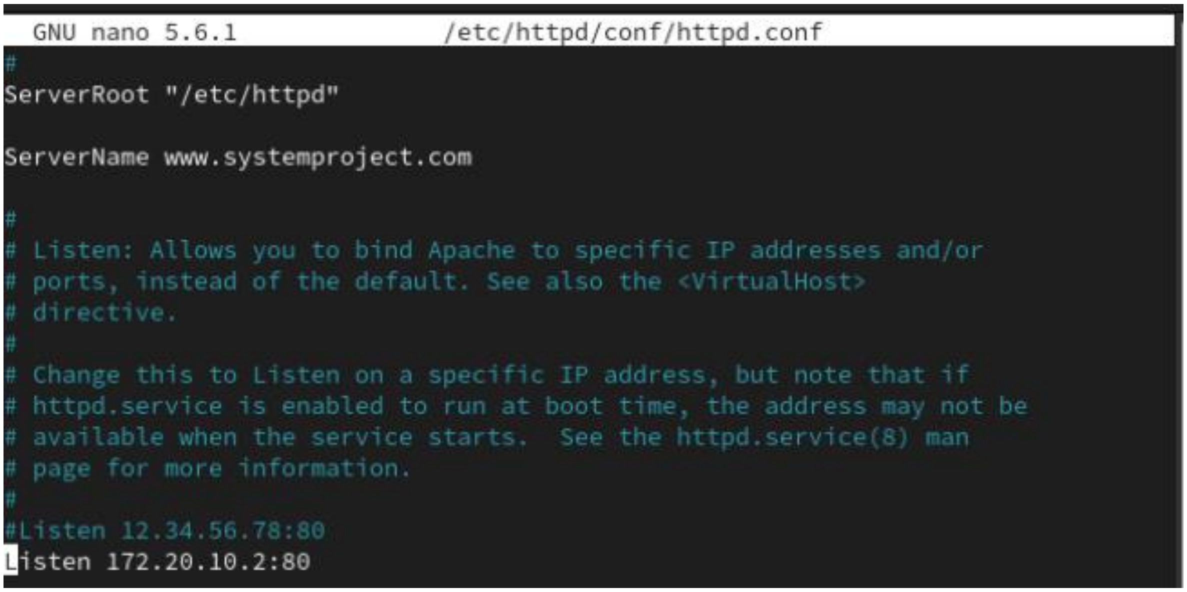


```
  GNU nano 5.6.1                     /etc/httpd/conf.d/systemproject.com.conf
<VirtualHost *:80>
        ServerName systemproject.com
        ServerAlias www.systemproject.com
        DocumentRoot /var/www/html
        ErrorLog /var/log/httpd/systemproject_error.log
        CustomLog /var/log/httpd/systemproject_access.log combined
</VirtualHost>
```



```
  GNU nano 5.6.1                                  /var/www/html/index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Transformers</title>
    <style>
        body, html {
            margin: 0;
            padding: 0;
            overflow: auto;
            font-family: Courier, sans-serif;
        }

        .container {
            display: flex;
            flex-direction: column;
            scroll-snap-type: y mandatory;
            overflow-y: scroll;
            height: 100vh;
        }
        /*text*/
```

(HTML continues) ^^

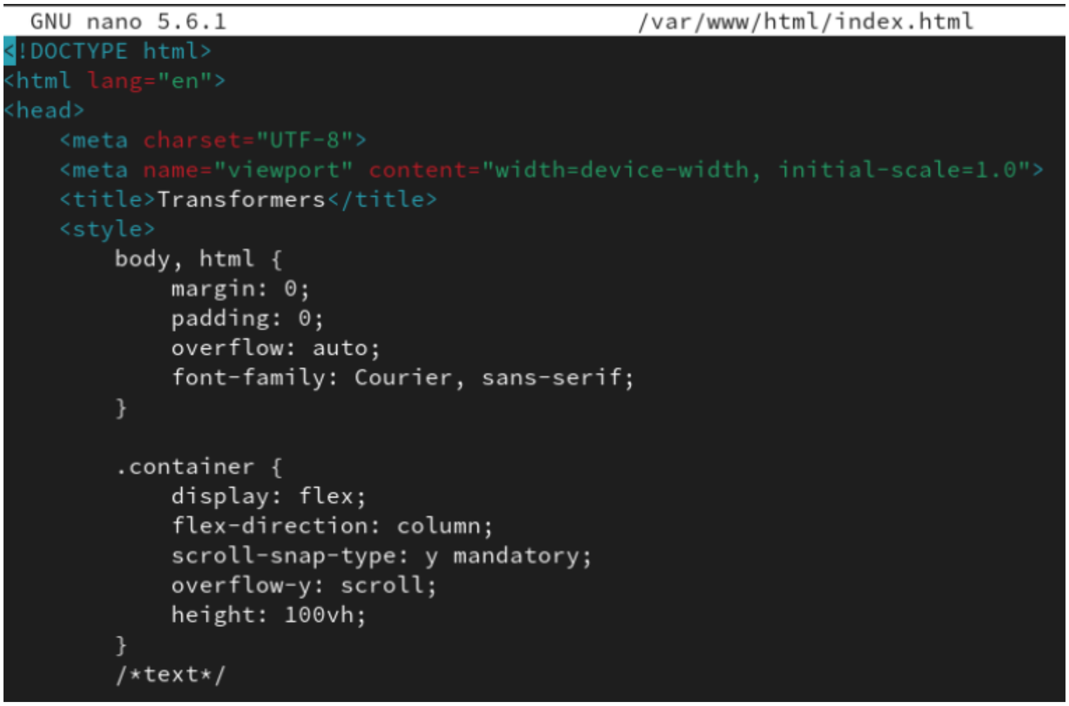


```
  GNU nano 5.6.1              /etc/httpd/conf/httpd.conf
#
ServerRoot "/etc/httpd"

ServerName www.systemproject.com

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts.  See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 172.20.10.2:80
```

Website

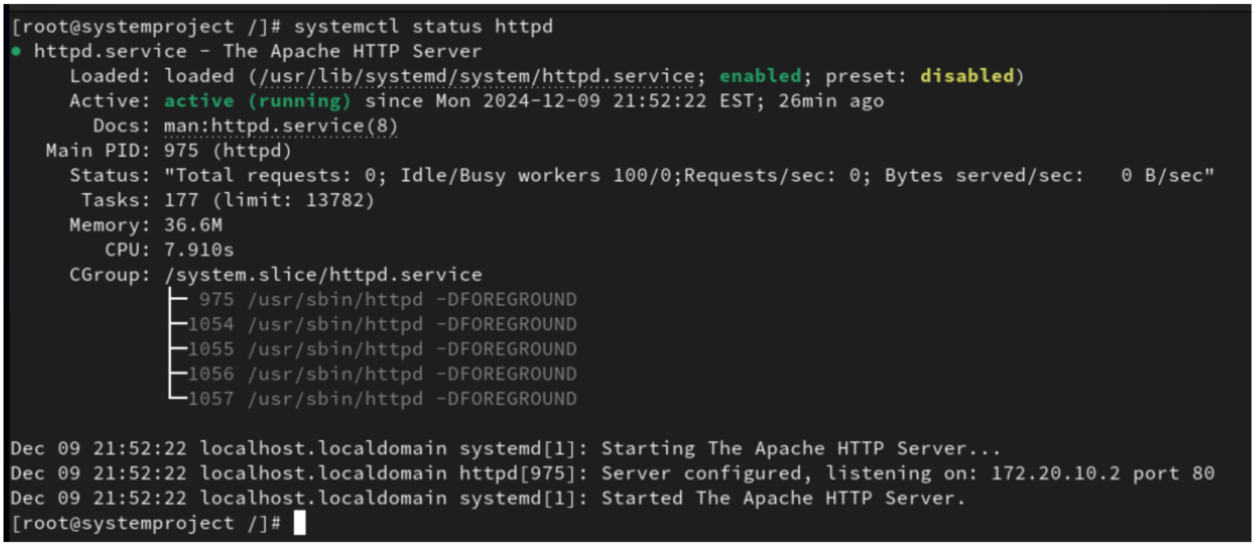
The HTML website we created is based on Transformers. Once on the site, the user will be able to scroll through different transformers. Each transformer has a picture of themselves next to a description of their character. The HTML code for this site was used for the WEB server by using “cd /var/www/html” and then putting the code into “index.html” using vim editor. Afterwards, we needed to restart the server because of the changes. We used “systemctl restart httpd” to restart it. Then, we can use the website by entering the server's name, www.systemproject.com, or by entering the server IP, 172.20.10.2.

Commands used: sudo nano /var/www/html/index.html, systemctl restart httpd, systemctl status httpd



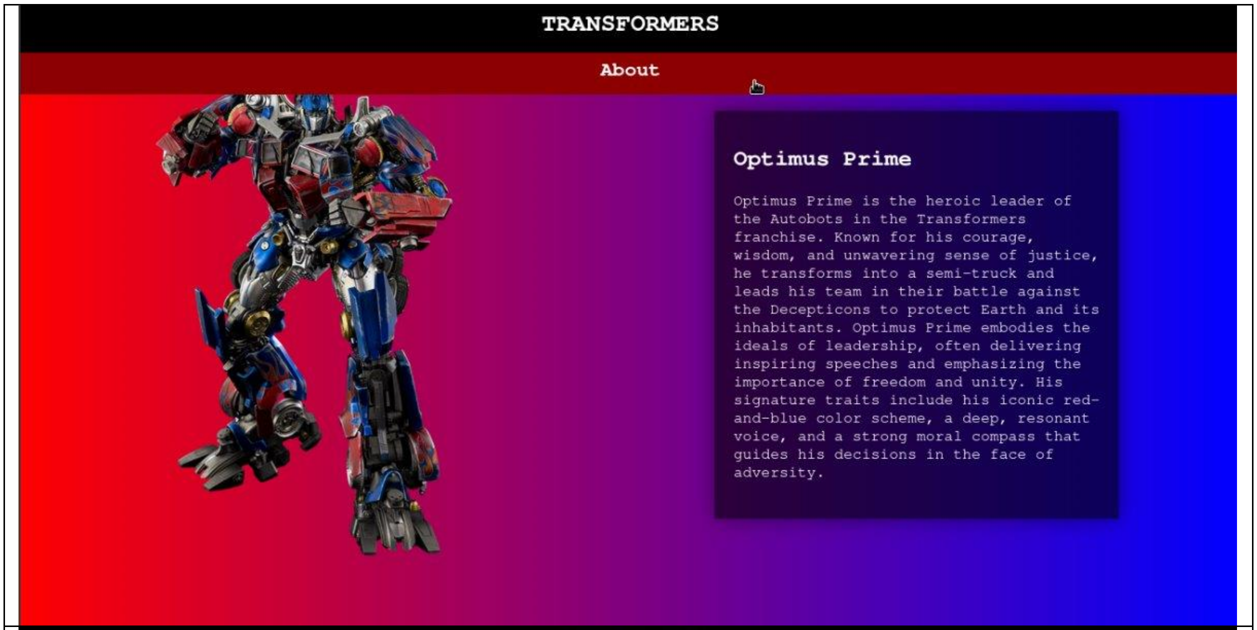
```
  GNU nano 5.6.1                                    /var/www/html/index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Transformers</title>
    <style>
        body, html {
            margin: 0;
            padding: 0;
            overflow: auto;
            font-family: Courier, sans-serif;
        }

        .container {
            display: flex;
            flex-direction: column;
            scroll-snap-type: y mandatory;
            overflow-y: scroll;
            height: 100vh;
        }
        /*text*/
```

(HTML continues) ^^



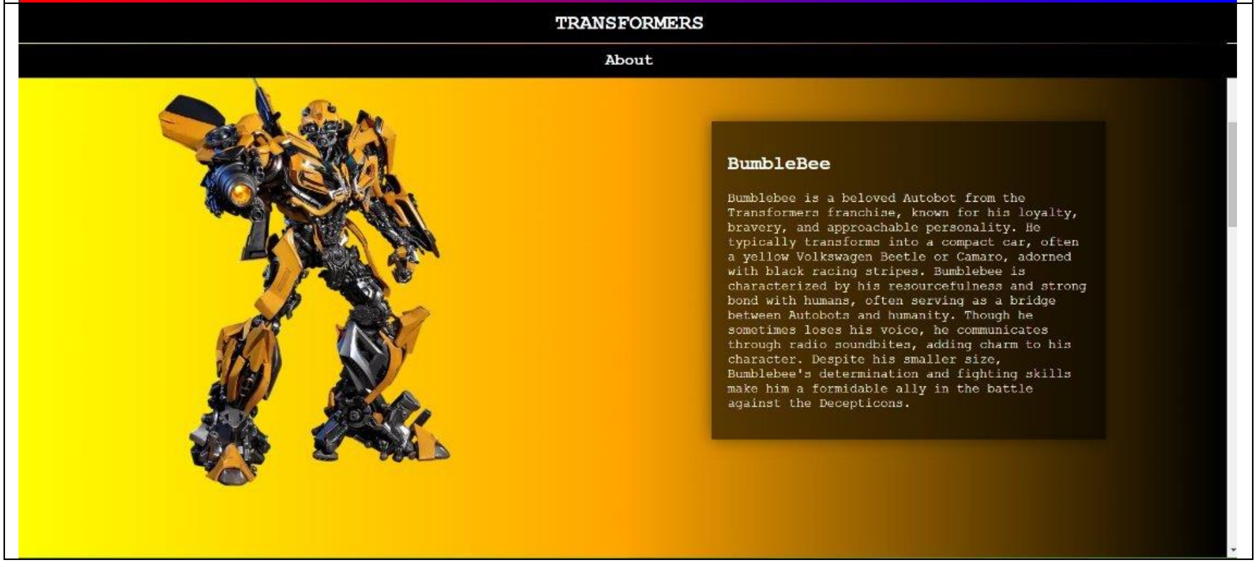
```
[root@systemproject /]# systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
     Active: active (running) since Mon 2024-12-09 21:52:22 EST; 26min ago
       Docs: man:httpd.service(8)
   Main PID: 975 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 13782)
     Memory: 36.6M
        CPU: 7.910s
     CGroup: /system.slice/httpd.service
             ├─ 975 /usr/sbin/httpd -DFOREGROUND
             ├─1054 /usr/sbin/httpd -DFOREGROUND
             ├─1055 /usr/sbin/httpd -DFOREGROUND
             ├─1056 /usr/sbin/httpd -DFOREGROUND
             └─1057 /usr/sbin/httpd -DFOREGROUND

Dec 09 21:52:22 localhost.localdomain systemd[1]: Starting The Apache HTTP Server...
Dec 09 21:52:22 localhost.localdomain httpd[975]: Server configured, listening on: 172.20.10.2 port 80
Dec 09 21:52:22 localhost.localdomain systemd[1]: Started The Apache HTTP Server.
[root@systemproject /]# █
```

Website Screenshots



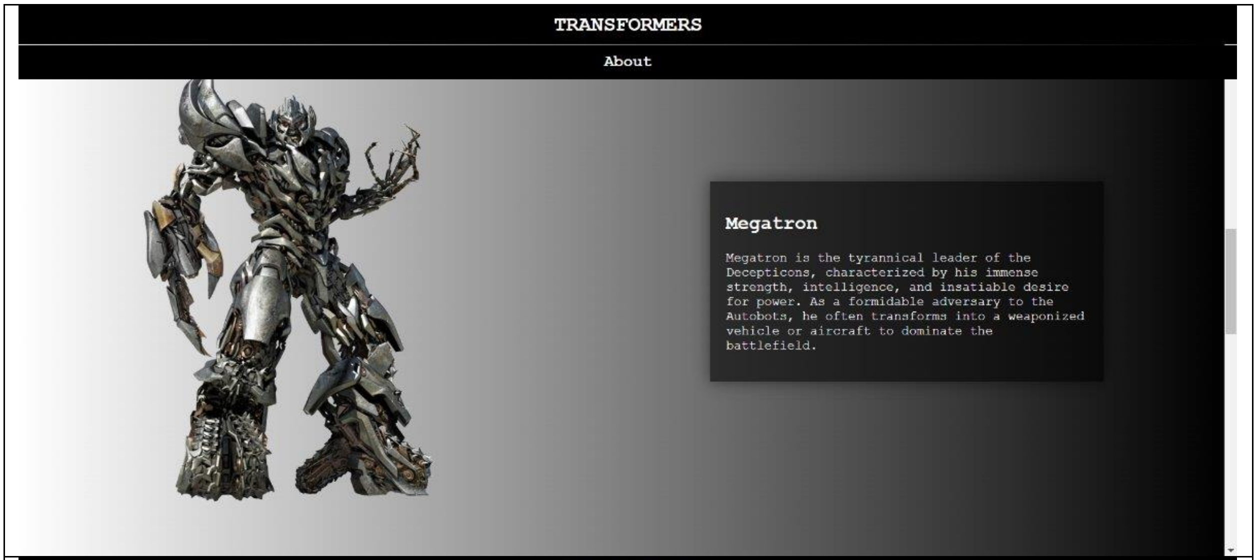
Optimus Prime

Optimus Prime is the heroic leader of the Autobots in the Transformers franchise. Known for his courage, wisdom, and unwavering sense of justice, he transforms into a semi-truck and leads his team in their battle against the Decepticons to protect Earth and its inhabitants. Optimus Prime embodies the ideals of leadership, often delivering inspiring speeches and emphasizing the importance of freedom and unity. His signature traits include his iconic red-and-blue color scheme, a deep, resonant voice, and a strong moral compass that guides his decisions in the face of adversity.



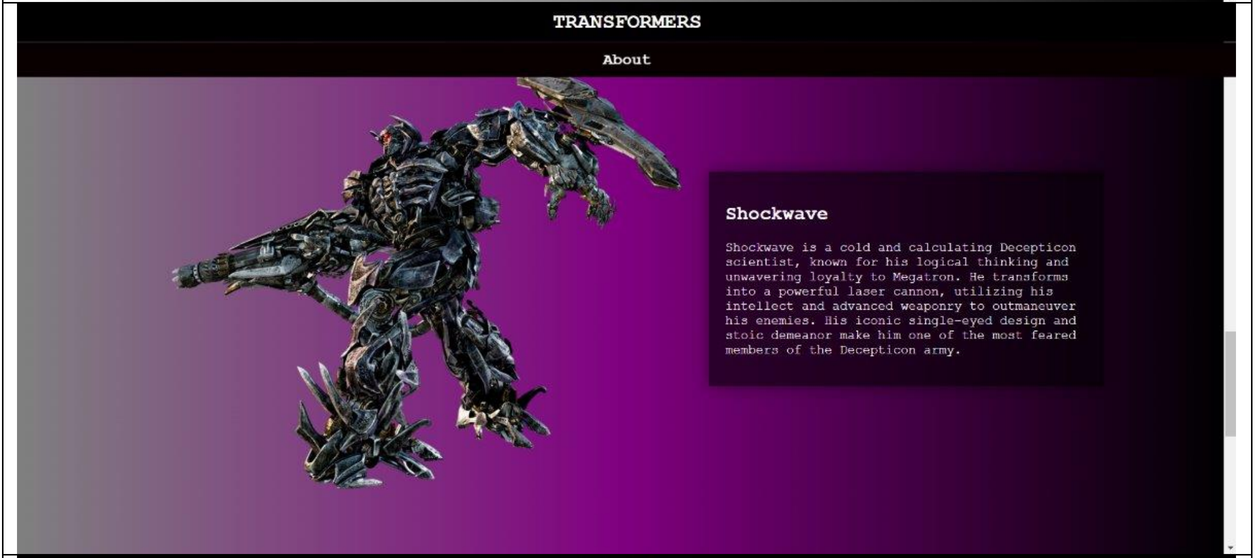
BumbleBee

Bumblebee is a beloved Autobot from the Transformers franchise, known for his loyalty, bravery, and approachable personality. He typically transforms into a compact car, often a yellow Volkswagen Beetle or Camaro, adorned with black racing stripes. Bumblebee is characterized by his resourcefulness and strong bond with humans, often serving as a bridge between Autobots and humanity. Though he sometimes loses his voice, he communicates through radio soundbites, adding charm to his character. Despite his smaller size, Bumblebee's determination and fighting skills make him a formidable ally in the battle against the Decepticons.



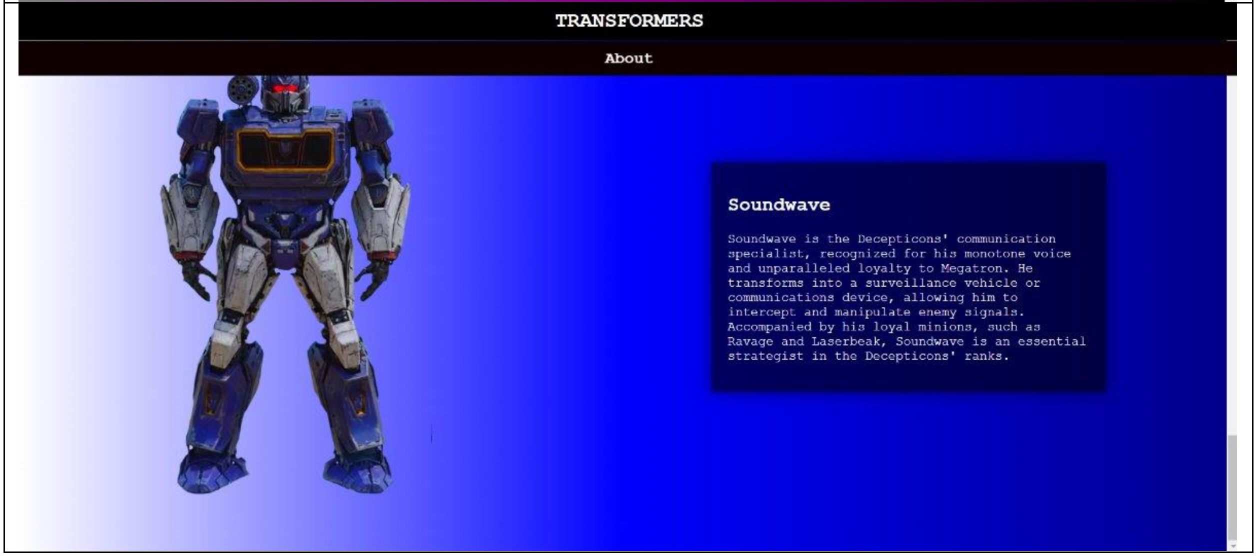
Megatron

Megatron is the tyrannical leader of the Decepticons, characterized by his immense strength, intelligence, and insatiable desire for power. As a formidable adversary to the Autobots, he often transforms into a weaponized vehicle or aircraft to dominate the battlefield.



Shockwave

Shockwave is a cold and calculating Decepticon scientist, known for his logical thinking and unwavering loyalty to Megatron. He transforms into a powerful laser cannon, utilizing his intellect and advanced weaponry to outmaneuver his enemies. His iconic single-eyed design and stoic demeanor make him one of the most feared members of the Decepticon army.



Soundwave

Soundwave is the Decepticons' communication specialist, recognized for his monotone voice and unparalleled loyalty to Megatron. He transforms into a surveillance vehicle or communications device, allowing him to intercept and manipulate enemy signals. Accompanied by his loyal minions, such as Ravage and Laserbeak, Soundwave is an essential strategist in the Decepticons' ranks.

Challenges

- **Setting up the DNS server**

  The DNS server was the main challenge since alot can go wrong while we configure. Just understanding which ip to use and figuring out in the end that the school wifi doesn't work really made it challenging. Also, what to program inside the zones and how to properly have them set up was a big learning curve, but we managed to figure it out some were difficult to troubleshoot others were simple. Just making sure we had an online DNS server was the stress, first time I thought I had it running but some of my zone configurations were off.

- **Time Management**

  With it being close to finals, time management was very tricky. Most of us have an average of 4-5 classes to manage and also part-time jobs so finding time was getting difficult. We managed to work on the project at least once a day for 1-2 hours.

Reflection/Summary

Thanks to this we were able to learn how to configure and run a web server using a Linux based operating system through RedHat. We faced a few difficulties when it came time to set the IPs through our DNS server as our original connection did not allow for a stable connection, but we moved to using a hotspot instead. Also, when we attempted to use two separate devices to run the web server it did not work so we opted to get everything onto one device. We were able to install the web server configurations without many complications and our HTML code as well. The only issue we faced was our images did not properly load onto our web page.

References

References made in regard to this project can be found in the Canvas course files, including materials given on DNS and Web Server configuration. Any additional references were made in regard to syntax on Red Hat Linux, which were found on the Red Hat Documentation Website.