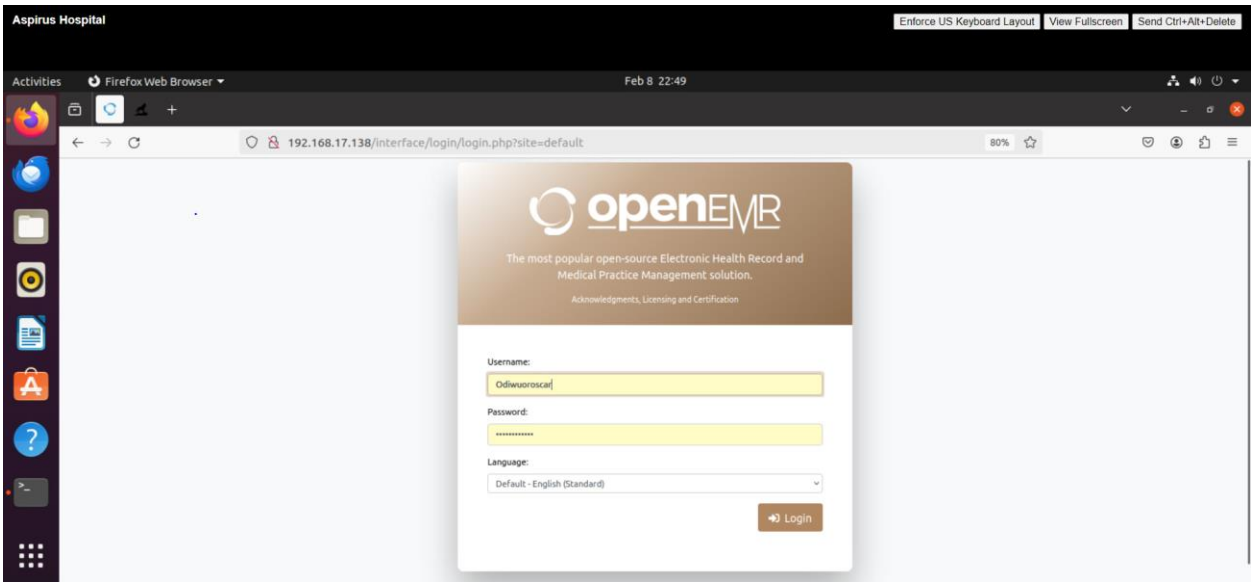
1. Creating virtual machines and assigning IP Addresses

Hospital | HIE	**OS Compatible with HAPI-FHIR?**	**OS Compatible with OpenEHR?**	**IP Address**	**Successfully Pinged the Other 4 VMs?** **Yes or No**
Aspirus	Yes	Yes	192.168.17.138	Yes
Portage Health	Yes	Yes	192.168.17.139	Yes
BCMH	Yes	Yes	192.168.17.140	Yes
MGH	Yes	Yes	192.168.17.141	Yes
UPHIE	Yes	Yes	192.168.17.142	Yes

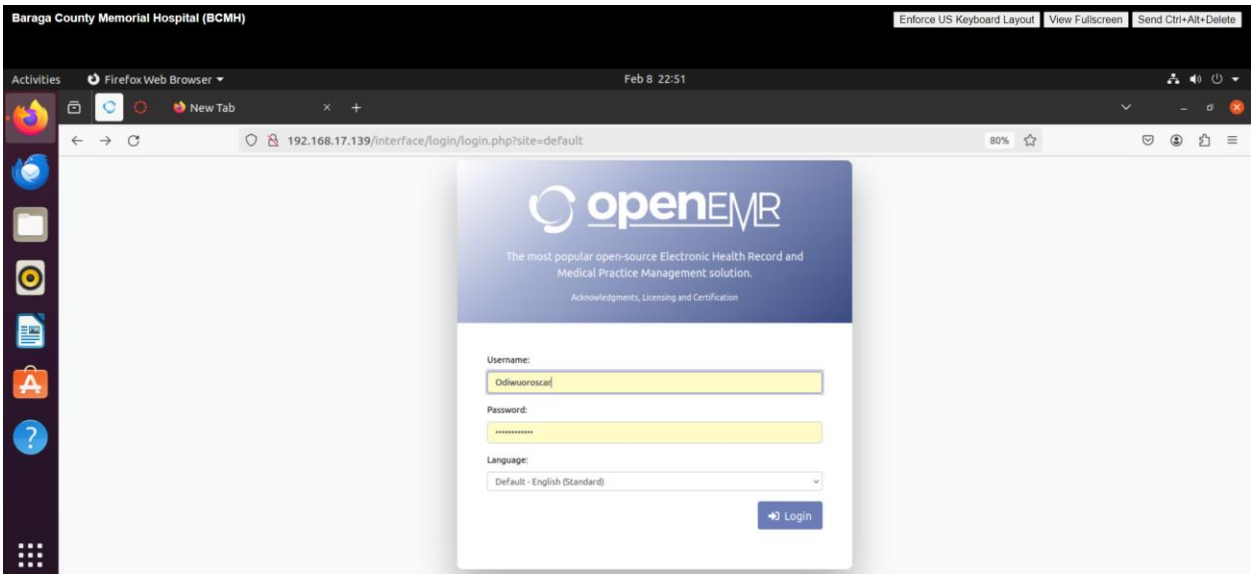
2. Installation, Configuration, and Security of OpenEMR.

A. Web page screenshot of the each hospital's successful installation of OpenEMR (8 points)

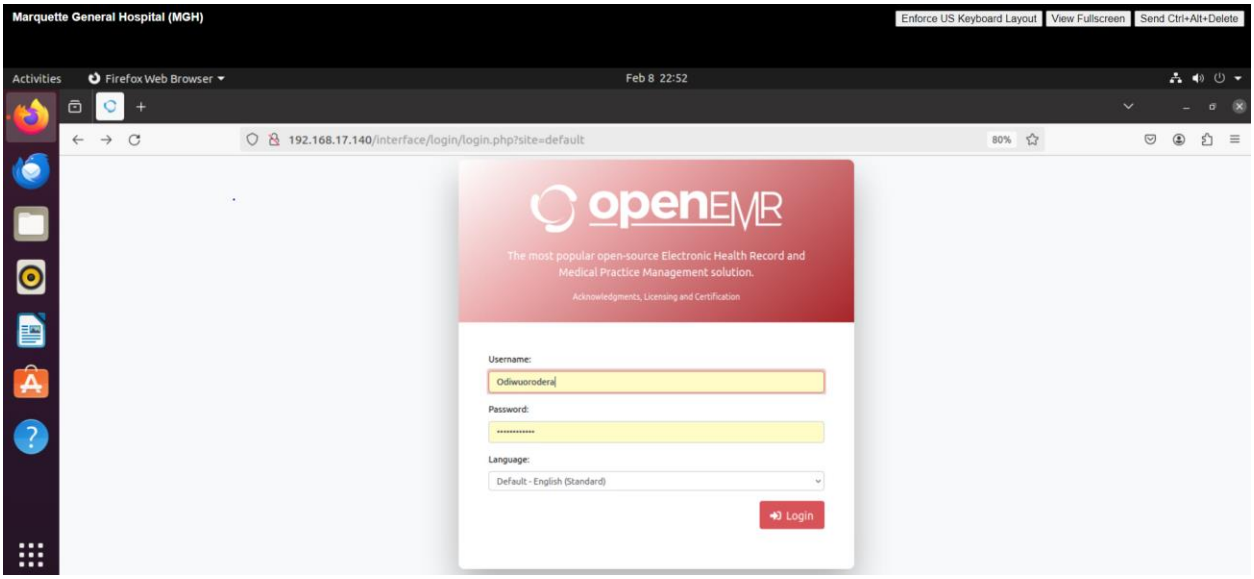
Aspirus Hospital



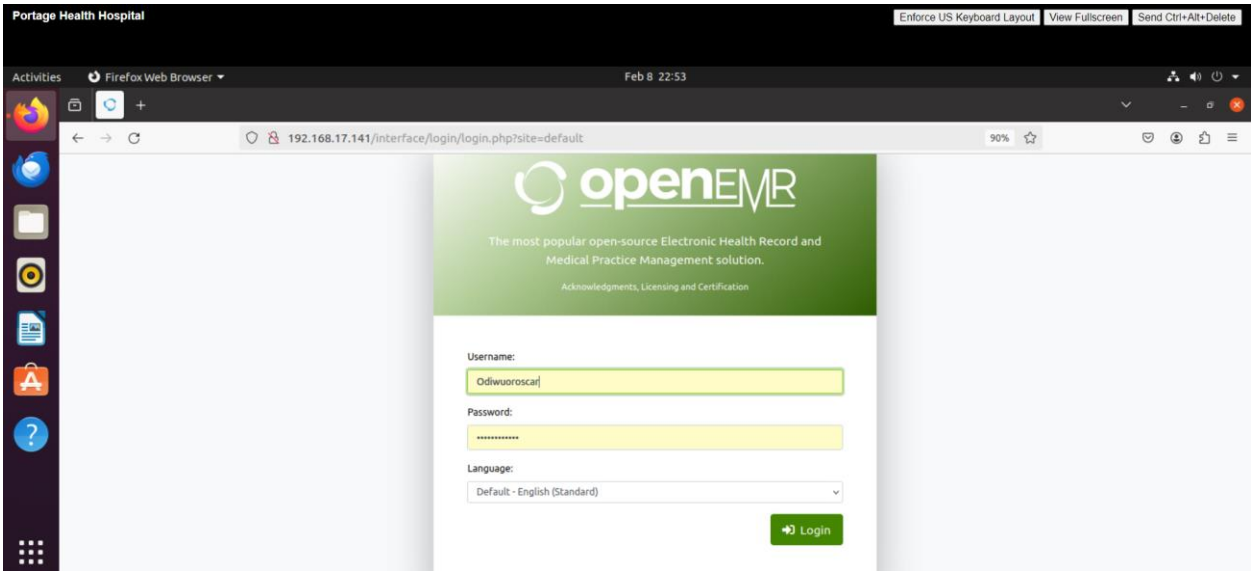
Baraga County Memorial Hospital



Marquette General Hospital



Portage Health Hospital



B. Show the steps and commands you used to secure OpenEMR (10 points)

After getting updates, upgrades and installation of unattended upgrades, I did the following to secure OpenEMR.

1. ***Enabling automatic security updates***

sudo dpkg-reconfigure --priority=low unattended-upgrades

2. ***Firewall configuration***

a) **Installation of ufw**; sudo apt-get install ufw
b) **Allowing HTTP, HTTPS & SSH;** sudo ufw allow http
sudo ufw allow https
sudo ufw allow ssh
c) ***Enabling firewall;*** *sudo ufw enable*

3. ***Securing apache***

a) **Editing apache security configuration file**; sudo nano /etc/apache2/conf-available/security.conf
b) **Modification of lines;**

ServerTokens Prod
ServerSignature Off
TraceEnable Off
Header set X-Content-Type-Options: "nosniff"
Header set X-Frame-Options: "sameorigin"
Header set X-XSS-Protection: "1; mode=block"
Header set X-Robots-Tag: "none"
Header set X-Download-Options: "noopen"
Header set X-Permitted-Cross-Domain-Policies: "none"

c) Enabling new security headers

sudo a2enconf security

4. Securing PHP

Editing php configuration: sudo nano /etc/php/8.1/apache2/php.ini
Line modification;expose_php = Off
display_errors = Off

5. File system permissions

sudo chown -R www-data:www-data openemr
sudo chmod -R 755 openemr

6. Use of strong passwords and two-factor authentication.

C. What other types of attacks would still be susceptible to the OpenEMR platform? (2 points)

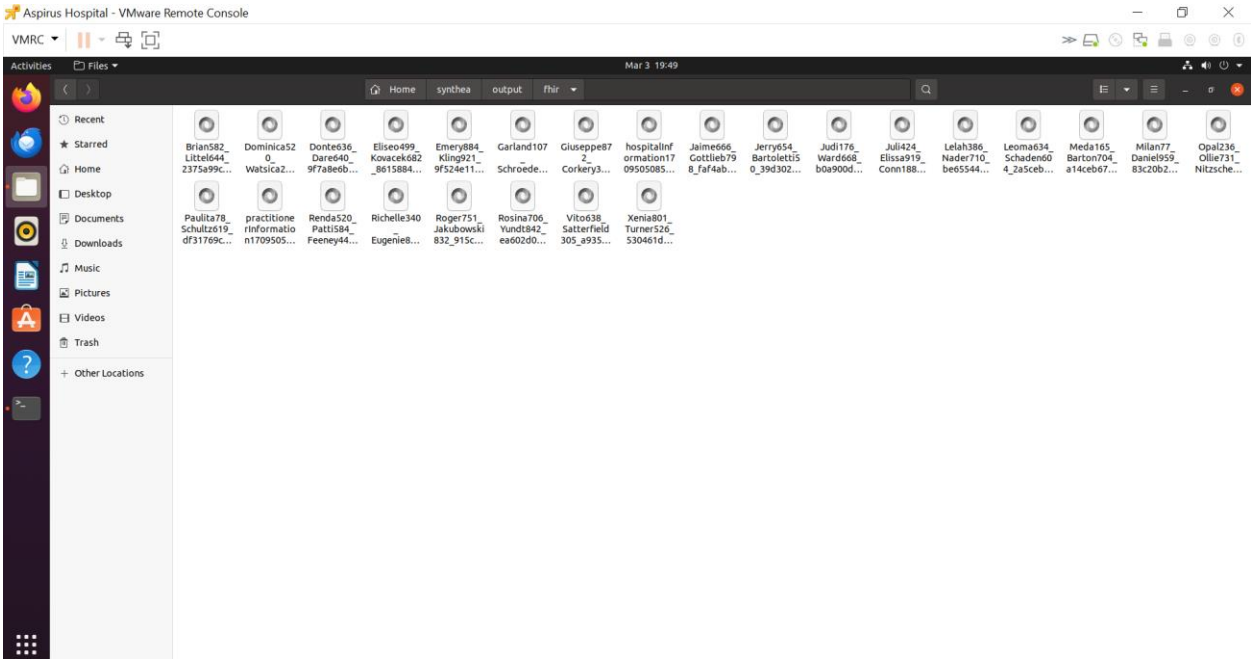
Physical security breaches
This refers to unauthorized access to physical devices hosting the virtual machine/Openemr server, such as my laptop or storage devices. This can result in data theft or device destruction. I can use cctv cameras and grill reinforcement at entry points to help secure the devices. I can also back up data in several other devices in separate locations.

Insider threats
Disgruntled employees within my 4 hospitals may abuse their privileged access to the OpenEMR platform to steal sensitive patient data, manipulate records, or disrupt operations. I can assign role-based access control (RBAC) and regular auditing to help detect and prevent insider threats.

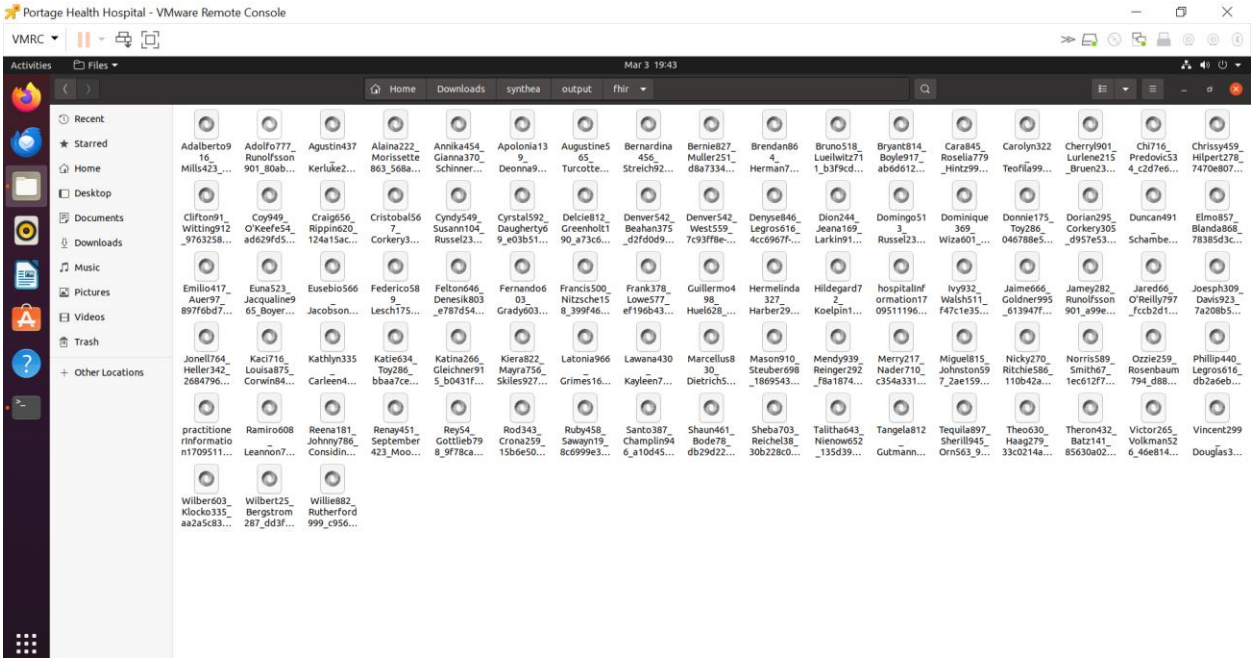
3. Creating Synthetic Data in Virtual Machines

A) .JSON Screenshots

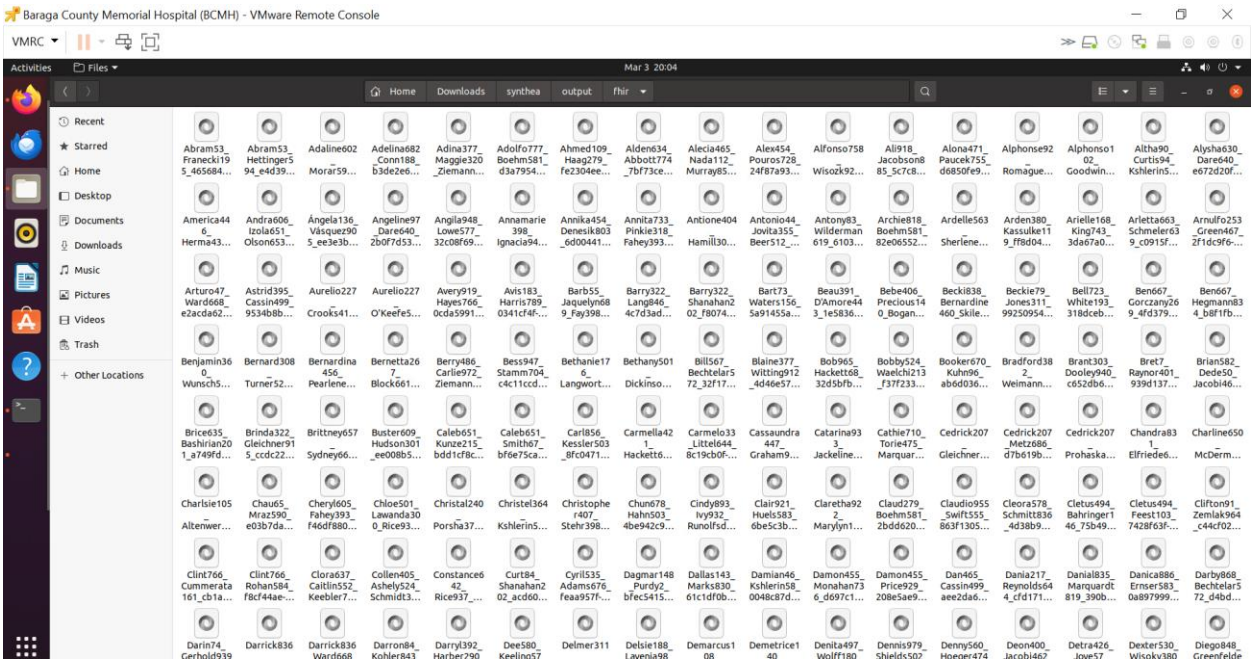
1. Aspirus



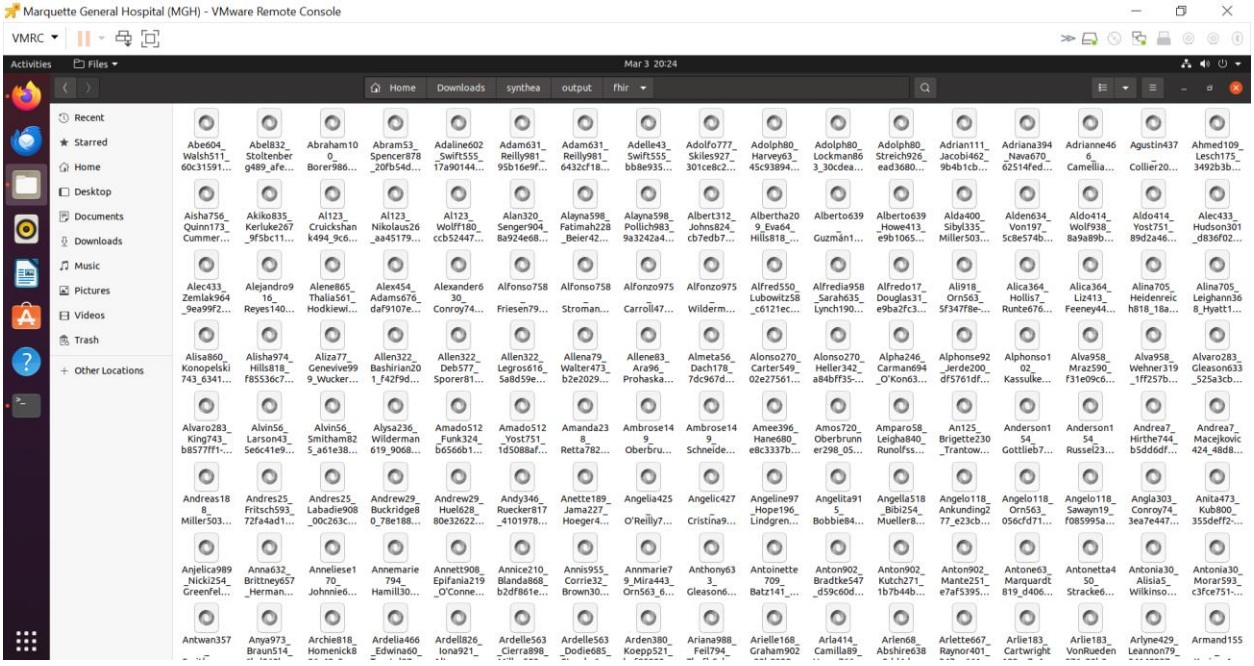
2. Portage



3. Baraga



Marquette



B) Complete the table below (4 points)

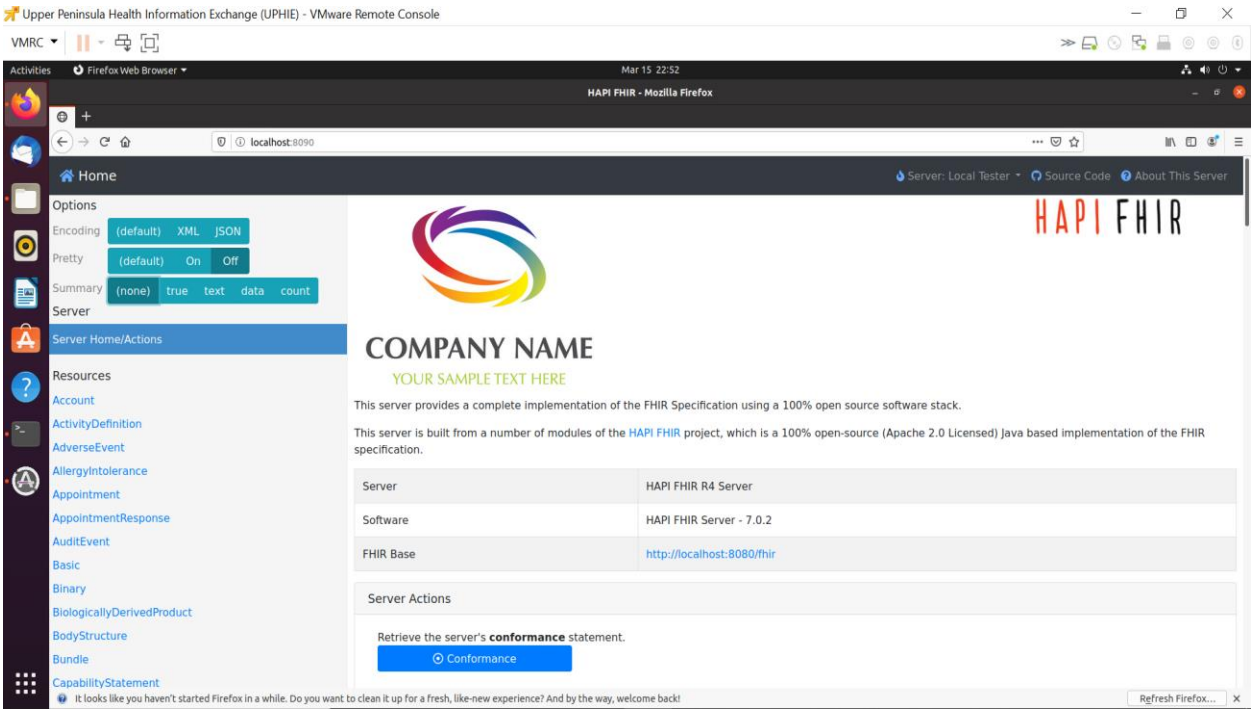
Hospital	**% Used of Total Served Hospital Population**	**Amount of COVID Patients Created**
Aspirus	0.4	20
Portage Health	0.7	63
BCMH	3	210
MGH	9	1800

C. What was the most challenging portion about this assignment? Why? (4 points)

Running the commands to generate patient data with Covid-19 simulation messages. This is because the provided naming convention of the hospitals resulted into an error “No demographics found”.

4. Hapi-FHIR Configuration

1. Take a screenshot of the HAPI-FHIR server’s default web UI and submit it with your other lab materials. (5 points)



2. What command would you use to check the total, used, and free memory in the system if you thought the HAPI-FHIR server was not running properly due to lack of memory? (2 points) *free -m*

```
test@sat3210-virtual-machine:~$ free -m
               total        used        free      shared  buff/cache   available
Mem:            3911        2124         444          67        1342        1483
Swap:           2047         967        1080
```

3. You've updated the firewall rules on your Ubuntu server and want to check if the port (say 8080) used by the HAPI-FHIR server is open. What command would you use? (5 points) *nmap -p 8090 localhost*

```
test@sat3210-virtual-machine:~$ nmap -p 8090 localhost
Starting Nmap 7.80 ( https://nmap.org ) at 2024-03-19 17:12 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00022s latency).

PORT     STATE SERVICE
8090/tcp open  opsmessaging
```

4. If the HAPI-FHIR server fails to start, you want to ensure that there are no other processes occupying the default port (8080). What command should you use to check if the port 8080 is currently in use? (4 points)

sudo apt-get update && sudo apt-get install net-tools

netstat -tuln | grep 8090

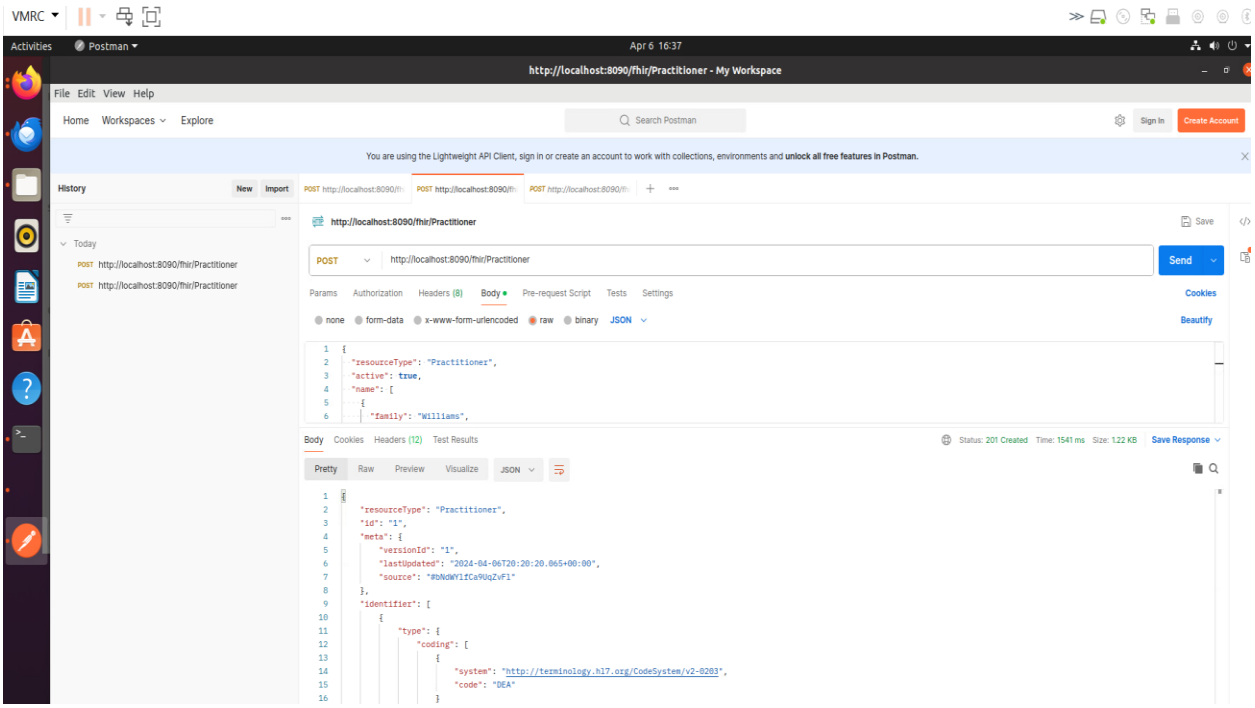
```
test@sat3210-virtual-machine:~$ netstat -tuln | grep 8090
tcp        0      0 0.0.0.0:8090            0.0.0.0:*               LISTEN
tcp6       0      0 :::8090                 :::*                    LISTEN
```

Bonus Section (2 points)

1. Customize the default HAPI-FHIR UI (web page) so that it says, “Upper Peninsula Health Care Network” instead of “Company Name” and “The Health Information Exchange for the Upper Peninsula“ instead of “Your Sample Text Here”. Submit a screen shot to verify your work. (2 points)

5. Interoperability with Hapi-FHIR and Postman

Take a screenshot, save as response.code.png (or other picture format extension), and submit with your assignment on canvas demonstrating successful response code. **(5 points)**



1. What is the process to create a new FHIR resource using Postman and HAPI FHIR server? (5 points)

1. **Setting up HAPI FHIR server on the computer or virtual machine**: I will ensure that I have the HAPI FHIR server running and accessible by testing the ports.
2. **Installation and launching of Postman**: I will use this code on *Terminal* *sudo snap install postman* then open it then select the POST method to signify the creation of a new resource on the server.
3. **Set request URL**: insert the URL of the HAPI FHIR server endpoint e.g. if I want to create a Practitioner resource, the URL will be **http://localhost:8090/fhir/Practitioner**

4. **Construction the request body** by using the from the example from https://fhir.cerner.com/millennium/r4/base/individuals/practitioner/

5. **Setting the request body type to JSON**: This is for the server to precisely interpret and understand the data transmitted in the request body.

6. **Send the request**: Upon setting up the request URL and body, click the "Send" button in Postman to send the request to the server, HAPI FHIR.

7. **Verification of response and error handling**: Confirm the response from the server to certify that the resource was generated successfully. The response should contain the newly created resource, its ID and any other relevant information like the response code 201.

2. What is the process to manage authentication when using Postman with a HAPI FHIR server? (5 points)

1. **Identification of the authentication methods**; HAPI FHIR server uses authentication methods , e.g. Basic Authentication, OAuth 2.0, API keys, and client certificates. Identify this from the server before configuration on postman.
2. **Configuration of Authorization in Postman**: Open Postman and launch a new request intended for communication with the HAPI FHIR server. Access the request settings and locate the Authorization tab.
3. **Selection of the appropriate authentication method** supported by the HAPI FHIR server (e.g., Basic Auth, OAuth 2.0).
4. **Enter the requisite credentials**; Depending on the authentication method, provide the required credentials like username/password, client ID/client secret, API key, token, etc. in the fields provided.
5. **Testing Authentication and validation**: Send the request from Postman to the HAPI FHIR server then confirm the success of the authentication process by probing the response from the server. In case of authentication failure, recheck the credentials or tokens provided in Postman.

3. How can you handle error responses from the HAPI FHIR server when using Postman? (5 points)

1. **Identification of the error response**: After sending a request from Postman to the HAPI FHIR server, cautiously examine the response received.
2. **Understanding the error Code**: Reference is made to the HAPI FHIR server documentation or FHIR specification design to understand the meaning of the specific error code received.
3. **Review the error message**: Check the error message provided in the response body for additional elements on the issue encountered.

4. **Troubleshoot the error**: Depending on the error code and message, resolve the issue by examining the request parameters, headers, and payload sent from Postman.
6. **Redo the request**: If the error seems to persist may be due to a temporary issue, consider redoin