Documentation

Code: System Monitoring

API: The rest API is developed by using the Django framework.
Testing: monitor/testing.py is written to manually test the endpoints
Authentication: monitor/authentication.py is written to authenticate the api. SIngle Key authentication is used to authenticate. CS218 is the key. I will try to extend it further by creating a database and storing various keys for different users.
Daemon: After deployment on AWS EC2 instance the api is daemonized with the help of systemd
A service file is written. Below is the code for the service file.

Create a service file in /etc/systemd/system and name it as rest_api.service

```
Sudo chmod +x rest_api.service

[Unit]
Description= System Monitoring Service
After= network.target

[Service]
ExecStart=/home/ec2-user/rest_api.sh
Restart=Always
StandardOutput=file:/var/log/rest_api.log
StandardError=file:/var/log/rest_api.err

[Install]
WantedBy=multi-user.target
```

Deployment: To deploy the API on AWS EC2 instance, a docker image of the api is created and pushed onto the docker hub. Below is the code of the dockerfile(this is present in github as well)

Dockerfile

```
FROM python:3.9-slim

WORKDIR /monitor

COPY . .

RUN pip install --no-cache-dir -r requirements.txt

EXPOSE 8000
#CMD ["gunicorn", "--bind", "0.0.0.0:8000", "system_monitor.wsgi:application"]
CMD ["python", "manage.py", "runserver","0.0.0.0:8000"]
```

With the help of the docker file the api is dockerized and available at DockerHub

Steps for deploying on EC2 instance before daemonizing the API:

1) Sudo yum install docker
2) Sudo systemctl start docker
3) sudo docker pull m4h1m4/rest_api:latest
4) sudo docker run -d -p 8000:8000 m4h1m4/rest_api
5) curl -L -H "Authorization:CS218" http://localhost:8000/api/cpu

   The same command can be extended to other endpoints as well.
6) Sudo docker ps - check the container ID to stop
7) Sudo docker stop <cont_id>
8) Sudo docker rm <cont_id>

After testing the API on EC2, systemd is used for daemonizing. The steps below gives the flow.

9) Create a bash file
10) Vim rest_api.sh

Rest_api.sh

```
#!/bin/bash
sudo docker run -d -p 8000:8000 m4h1m4/rest_api
```

```
Sudo chmod +x rest_api.sh
```

Create a service file in /etc/systemd/system and name it as rest_api.service

```
Sudo chmod +x rest_api.service
```

```
[Unit]
Description= System Monitoring Service
After= network.target

[Service]
ExecStart=/home/ec2-user/rest_api.sh
Restart=Always
StandardOutput=file:/var/log/rest_api.log
StandardError=file:/var/log/rest_api.err

[Install]
WantedBy=multi-user.target
```

```
Sudo systemctl enable rest_api.service
Sudo systemctl start rest_api
Sudo systemctl status
Sudo docker ps #to check whether the docker image is running on re-starting the EC2 instance.
```