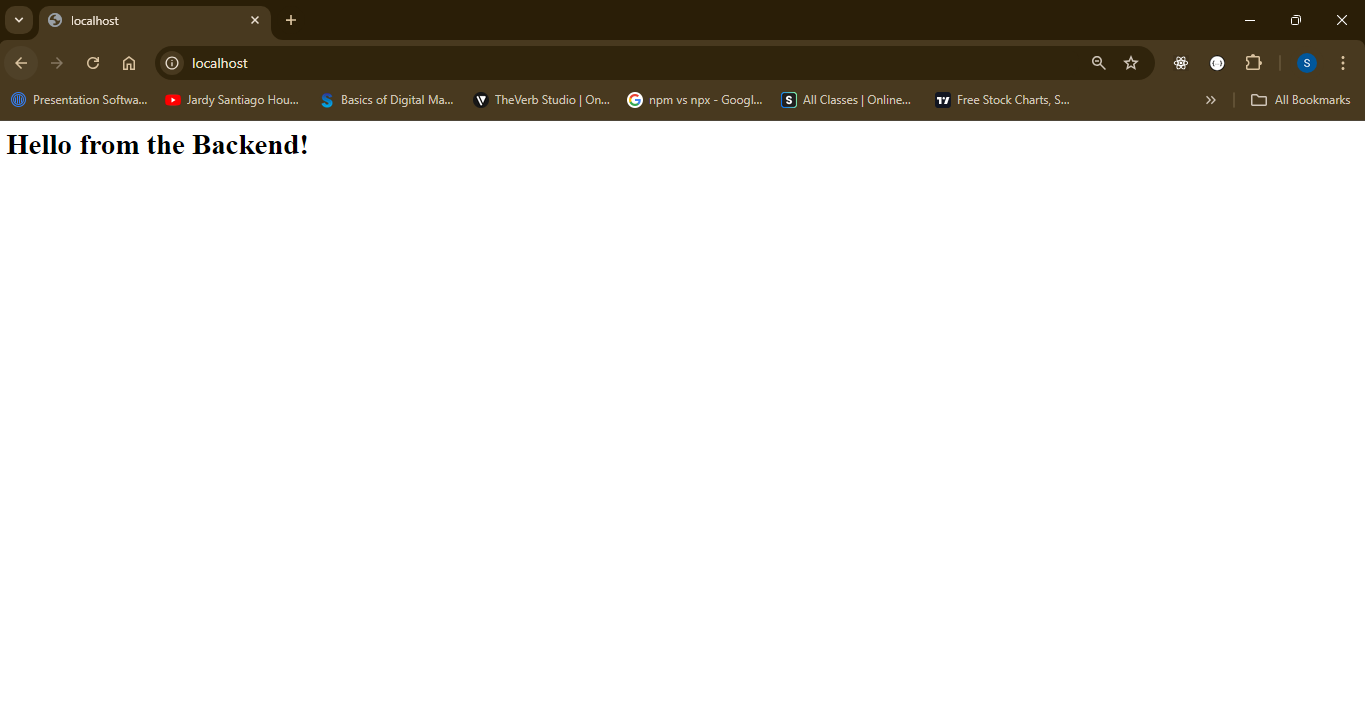
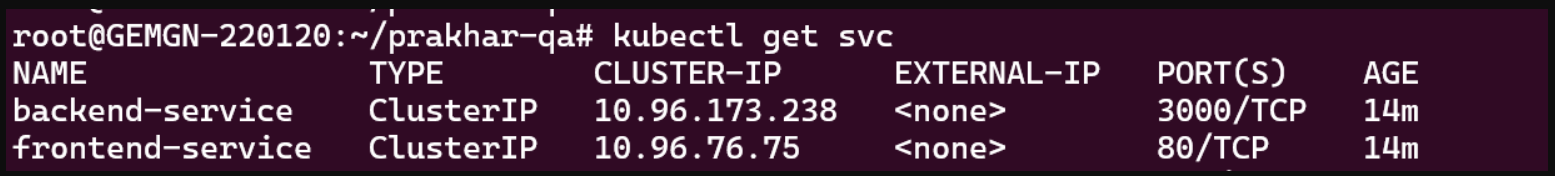
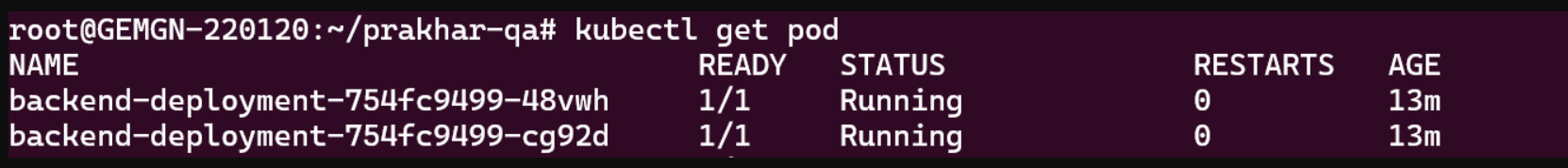
**Problem Statement 1:**

****

****

****

****

**Problem Statement 2:**

1. **System Health Monitoring Script:**

Develop a script that monitors the health of a Linux system. It should check

CPU usage, memory usage, disk space, and running processes. If any of

these metrics exceed predefined thresholds (e.g., CPU usage > 80%), the

script should send an alert to the console or a log file.

**Bash Script –**

#!/bin/bash

# Thresholds

CPU\_THRESHOLD=80

MEM\_THRESHOLD=80

DISK\_THRESHOLD=90

# Log file location

LOG\_FILE="/var/log/system\_health.log"

# Function to log alerts

log\_alert() {

local message=$1

echo "$(date) : $message" | tee -a $LOG\_FILE

}

# Check CPU usage

cpu\_usage=$(top -bn1 | grep "Cpu(s)" | sed "s/.\*, \*\([0-9.]\*\)%\* id.\*/\1/" | awk '{print 100 - $1}')

if (( $(echo "$cpu\_usage > $CPU\_THRESHOLD" | bc -l) )); then

log\_alert "ALERT: CPU usage is above threshold! Usage: $cpu\_usage%"

fi

# Check memory usage

mem\_usage=$(free | grep Mem | awk '{print $3/$2 \* 100.0}')

if (( $(echo "$mem\_usage > $MEM\_THRESHOLD" | bc -l) )); then

log\_alert "ALERT: Memory usage is above threshold! Usage: $mem\_usage%"

fi

# Check disk space usage

disk\_usage=$(df / | grep / | awk '{ print $5 }' | sed 's/%//g')

if [ $disk\_usage -gt $DISK\_THRESHOLD ]; then

log\_alert "ALERT: Disk space usage is above threshold! Usage: $disk\_usage%"

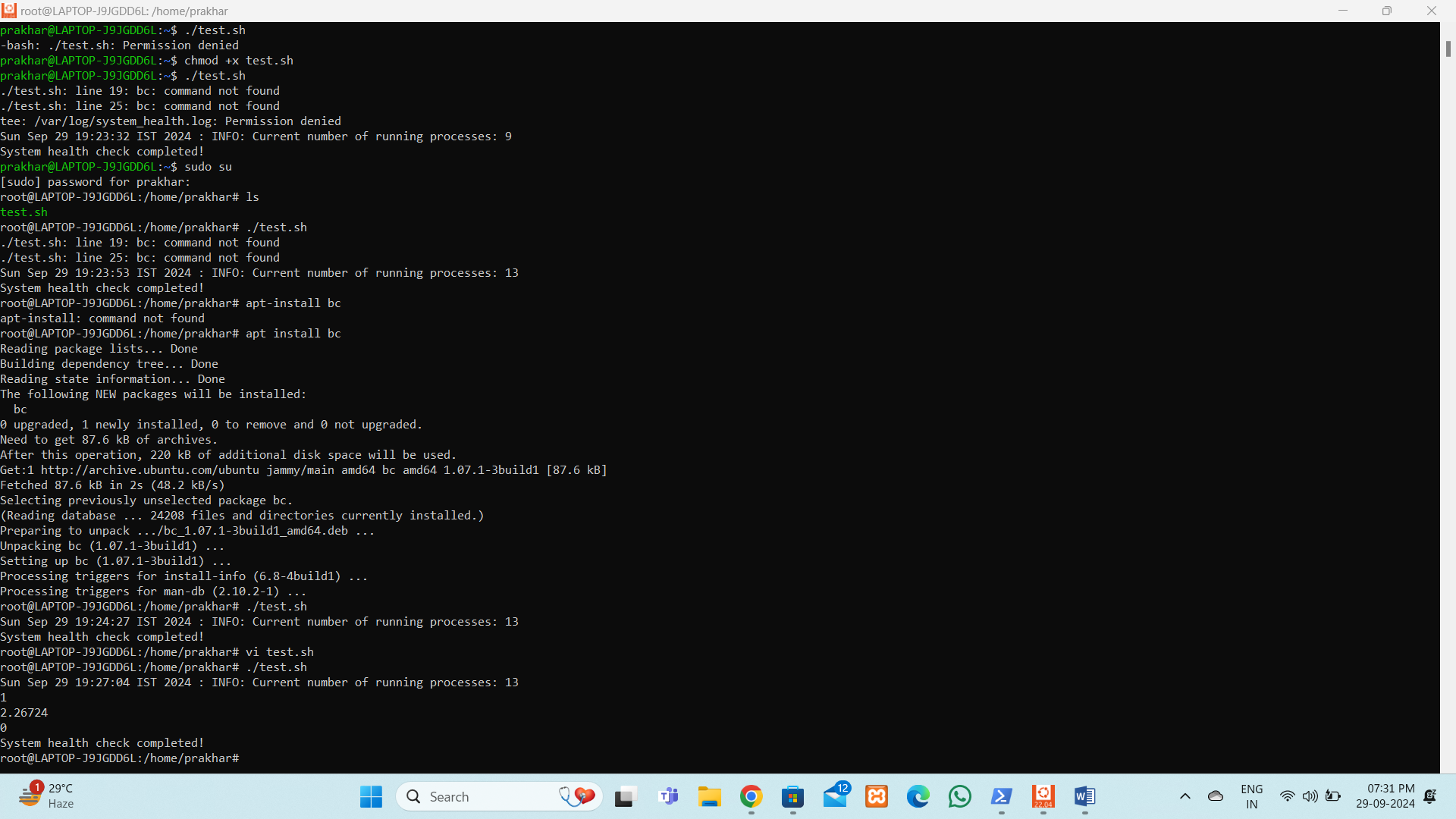
fi

# Check the number of running processes

process\_count=$(ps aux | wc -l)

log\_alert "INFO: Current number of running processes: $process\_count"

echo "System health check completed!"



4. Application Health Checker:

Please write a script that can check the uptime of an application and

determine if it is functioning correctly or not. The script must accurately

assess the application's status by checking HTTP status codes. It should be

able to detect if the application is 'up', meaning it is functioning correctly, or

'down', indicating that it is unavailable or not responding.

**Bash Script –**

#!/bin/bash

# Application URL to check

URL="http://your-application-url.com"

# Log file location

LOG\_FILE="/var/log/app\_health.log"

# Function to log the application status

log\_status() {

local message=$1

echo "$(date) : $message" | tee -a $LOG\_FILE

}

# Make an HTTP request and get the status code

status\_code=$(curl -o /dev/null -s -w "%{http\_code}" $URL)

# Check the HTTP status code and determine the application's health

if [ "$status\_code" -eq 200 ]; then

log\_status "Application is UP. Status code: $status\_code"

else

log\_status "Application is DOWN or not responding. Status code: $status\_code"

fi

