**Task Description:**

1. Create a shell script to print the HTTP error code of guvi.in & print, the success/failure message based on the error code response.

**Solution:**

#!/bin/bash

# Function to get HTTP status code

get\_http\_status() {

local url="$1"

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

echo "$status\_code"

}

# Function to interpret common HTTP status codes

interpret\_status() {

local status\_code="$1"

case "$status\_code" in

200) echo "200 OK - Request succeeded.";;

301) echo "301 Moved Permanently - The resource has moved.";;

302) echo "302 Found - The resource has moved temporarily.";;

400) echo "400 Bad Request - The server cannot process the request.";;

401) echo "401 Unauthorized - Authentication is required.";;

403) echo "403 Forbidden - Access is denied.";;

404) echo "404 Not Found - The resource doesn't exist.";;

500) echo "500 Internal Server Error - Server encountered an error.";;

502) echo "502 Bad Gateway - Invalid response from upstream server.";;

503) echo "503 Service Unavailable - Server is temporarily unavailable.";;

\*) echo "Unknown or less common status code: $status\_code";;

esac

}

# Main script

url="http://guvi.in"

status=$(get\_http\_status "$url")

echo "HTTP Status Code for $url: $status"

interpret\_status "$status"

if [[ "$status" -ge 200 && "$status" -lt 300 ]]; then

echo "Success!"

else

echo "Failure!"

fi

**Output:**



1. Given a file, replace all occurrence of the word "give" with "learning" from 5th line till the end in only those lines that contain the word "welcome"

**Solution:**

#!/bin/bash

file="$1" # File name from command-line argument

if [ ! -f "$file" ]; then

echo "File '$file' not found."

exit 1

fi

line\_count=$(wc -l < "$file")

start\_line=5

if [ "$line\_count" -lt "$start\_line" ]; then

echo "File has less than 5 lines. No replacement needed."

exit 0

fi

temp\_file=$(mktemp)

{

head -n "$((start\_line - 1))" "$file" # Copy first 4 lines unchanged

tail -n "+$start\_line" "$file" | while IFS= read -r line; do

if [[ "$line" == welcome ]]; then

echo "${line//give/learning}" #replace if line contains welcome

else

echo "$line" #print line if it does not contain welcome.

fi

done

} > "$temp\_file"

mv "$temp\_file" "$file"

echo "Replacement completed."

**Output:**

