```
nandy@DESKTOP-1FTMM1B:~$ vi Check_Status.sh
nandy@DESKTOP-1FTMM1B:~$ chmod +X Check_Status.sh
nandy@DESKTOP-1FTMM1B:~$ ./Check_Status.sh
-bash: ./Check_Status.sh: Permission denied
nandy@DESKTOP-1FTMM1B:~$ chmod +X Check_Status.sh
nandy@DESKTOP-1FTMM1B:~$ vi Check_Status.sh
nandy@DESKTOP-1FTMM1B:~$ ./Check_Status.sh
-bash: ./Check_Status.sh: Permission denied
nandy@DESKTOP-1FTMM1B:~$ chmod +x Check_Status.sh
nandy@DESKTOP-1FTMM1B:~$ ./Check_Status.sh
HTTP Status Code for https://guvi.in: 301
Unknown status: Could not determine success or failure based on the status code.
nandy@DESKTOP-1FTMM1B:~$ chmod +X Check_Status.sh
```

```bash
#!/bin/bash

URL="https://guvi.in"

# Use curl to get only the HTTP status code
# -s: Silent mode, hides progress meter and error messages
# -o /dev/null: Discard the response body
# -w "%{http_code}": Print only the HTTP status code to stdout
HTTP_CODE=$(curl -s -o /dev/null -w "%{http_code}" "$URL")

echo "HTTP Status Code for $URL: $HTTP_CODE"

# Check the HTTP status code and print a message
if [[ "$HTTP_CODE" -ge 200 && "$HTTP_CODE" -lt 300 ]]; then
  echo "Success: The website is accessible."
elif [[ "$HTTP_CODE" -ge 400 && "$HTTP_CODE" -lt 500 ]]; then
  echo "Failure: Client-side error (e.g., Page Not Found)."
elif [[ "$HTTP_CODE" -ge 500 && "$HTTP_CODE" -lt 600 ]]; then
  echo "Failure: Server-side error."
else
  echo "Unknown status: Could not determine success or failure based on the status code."
fi
~
~
~
~
~
~
~
"Check_Status.sh" 22L, 823B
```