

Distributed Systems Programming Assignment 1

Submitted by: Shreya Bisen (sbisen@scu.edu)

Description:

The goal of this assignment is to build a simple web server in Java that listens for client connections and serves files using the HTTP/1.0 and HTTP/1.1 protocol. The server will translate relative file requests to absolute paths based on a specified document root, check file existence and permissions, and respond with appropriate HTTP status codes. Implemented this in **Java** using **multi-threaded** architecture to handle client connections. The server supports various static file types: HTML, TXT, JPG (JPEG), PNG, GIF, JSON and SVG.

The program **implements a basic HTTP server** that:

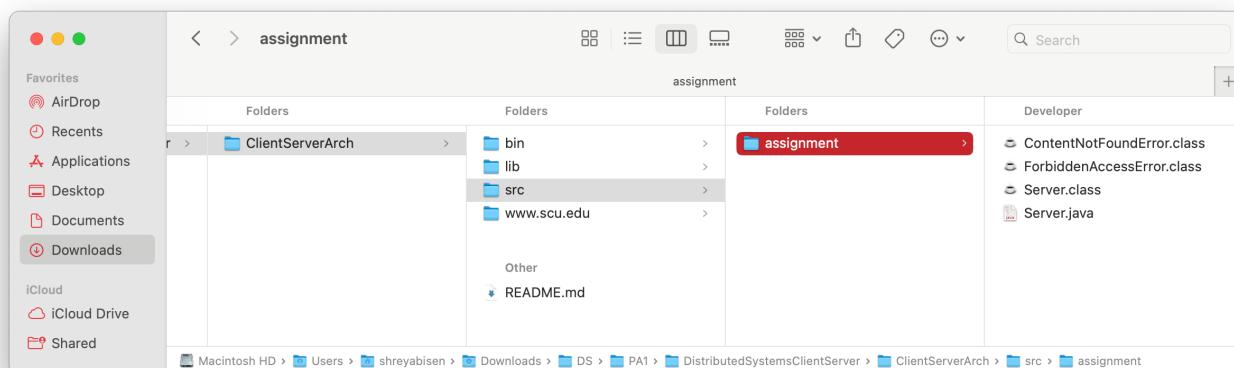
- Listens for incoming connections.
- Handles GET requests for files from a specified directory.
- Sends responses with file data or error messages based on the request's validity and the file's accessibility.
- Supports concurrent processing of multiple clients using a thread pool for efficient resource management.

Rubrics fulfilled:

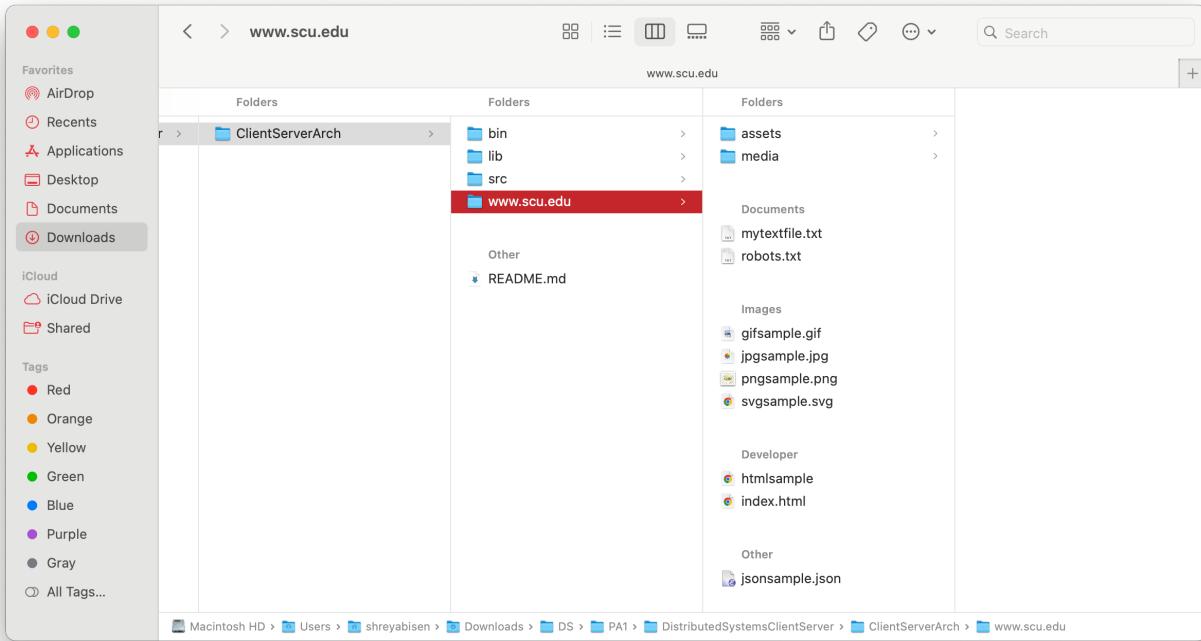
- Handling Default page: Used default redirect to index.html
- Handling - All types of errors: 200 (HTTP OK), 400 (BAD REQUEST), 403 (Forbidden), 404 (File Not Found)
- Page displayed correctly : All supported file types are displayed correctly, along with accurate error displays.
- Images displayed correctly : All the supported image files are displayed properly
- Approach Used: Multithreading
- Close the connection after every request: Closed connection after every request made
- Single threaded -4, multi threaded -10 (should use active connections for heuristic) : Used fixed thread pool (multi-threaded) -10 (should use active connections for heuristic)

A list of submitted files:

1. Java Project files



2. www.scu.edu folder inside the same Java project folder



Submitting a tarball file to unzip and use.

(Excluded the file txtsample.txt whose permission was changed to write only -> to display the Forbidden Access Error. Since it prevents from creating a tarball archive file, I had to exclude the file. Instructions are presented below on how to change permission of file.)

3. README.pdf (This file which has all the compile instructions & snapshots)

Instructions for running program:

Unzip the tarball archive and do the following-

Use any IDE of your choice with JDK installed on your machine, type the following commands on the terminal to compile and start the server:

- javac Server.java
- java Server -document_root <document_root_path> -port <port>

Example usage:

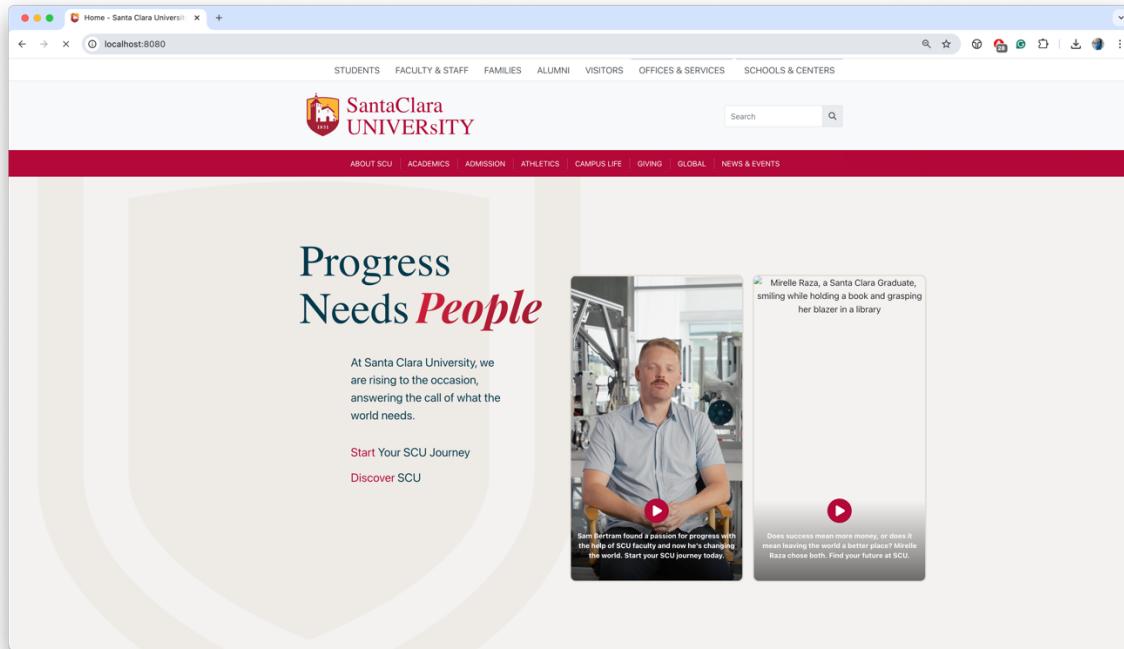
- javac Server.java
- java Server -document_root /Users/shreyabisen/Downloads/DS/PA1/DistributedSystemsClientServer/ClientServerArch/www.scu.ed u -port 8080

("assignment" was my package name, that's why I had to access using Java file using "assignment.Server")

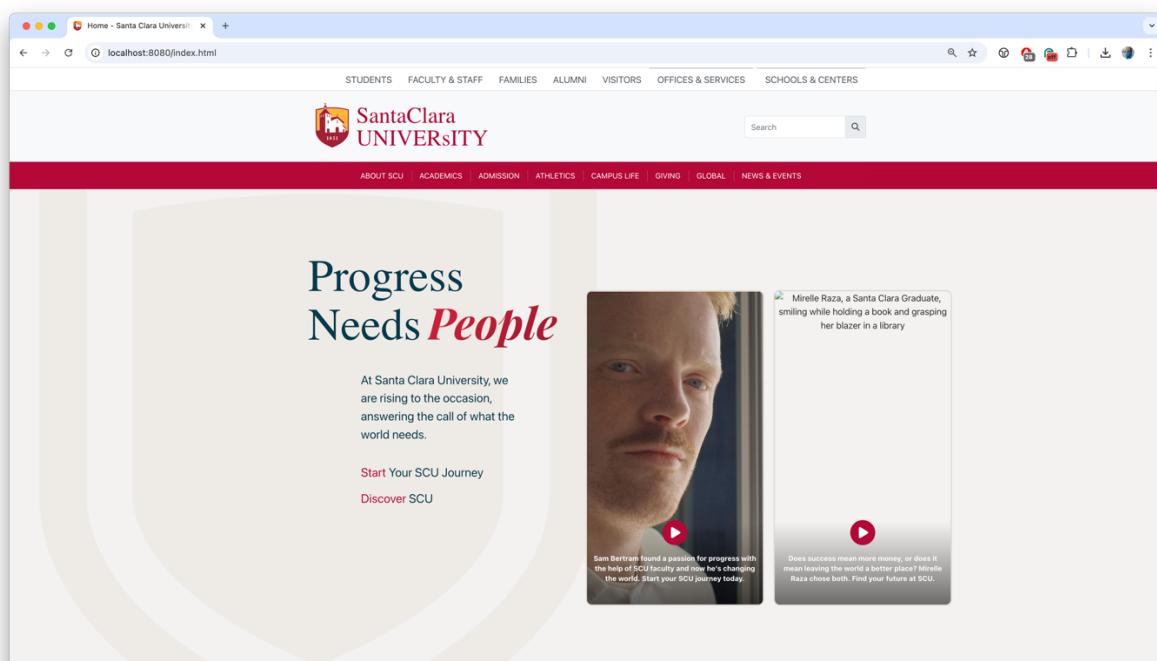
```
(base) shreyabisen@Shreyas-MacBook-Air-3 src % cd assignment
(base) shreyabisen@Shreyas-MacBook-Air-3 assignment % javac Server.java
(base) shreyabisen@Shreyas-MacBook-Air-3 assignment % cd ..
(base) shreyabisen@Shreyas-MacBook-Air-3 src % java assignment.Server -document_root /Users/shreyabisen/Downloads/DS/PA1/DistributedSystemsClientServer/ClientServerArch/www.scu.ed
u -port 8080
Starting server on port 8080 with document root /Users/shreyabisen/Downloads/DS/PA1/DistributedSystemsClientServer/ClientServerArch/www.scu.edu
Server is running on port 8080
REQUEST: GET / HTTP/1.1
```

Snapshots of a web browser accessing web server:

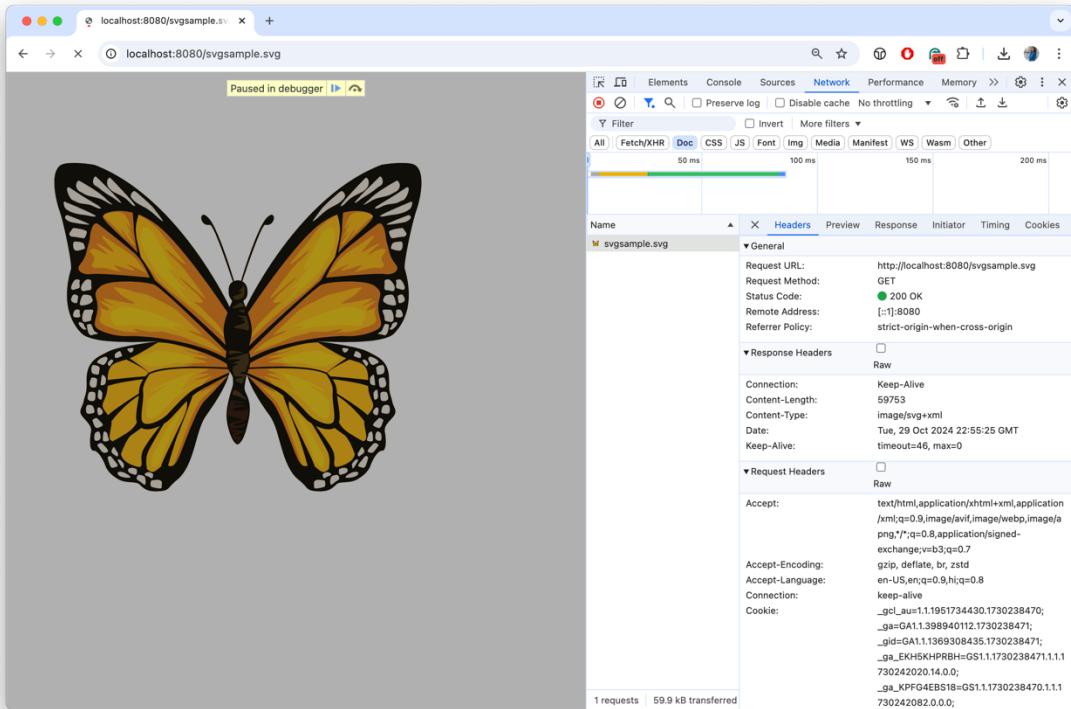
Accessing localhost:8080, displays the Home page of www.scu.edu



Accessing localhost:8080/index.html, also displays the Home page of www.scu.edu



Accessing localhost:8080/svgsample.svg, displays the image correctly (along with “200 OK” Status Code)



The screenshot shows a browser window with the URL `localhost:8080/svgsample.svg`. The main content area displays a detailed illustration of a monarch butterfly with its wings spread. To the right of the browser is the Chrome DevTools Network tab, which is paused in a debugger. The Network tab shows a single request for the SVG file. The "General" section of the request details shows:

- Request URL: `http://localhost:8080/svgsample.svg`
- Request Method: GET
- Status Code: 200 OK
- Remote Address: [::]:8080
- Referrer Policy: strict-origin-when-cross-origin

The "Response Headers" section shows:

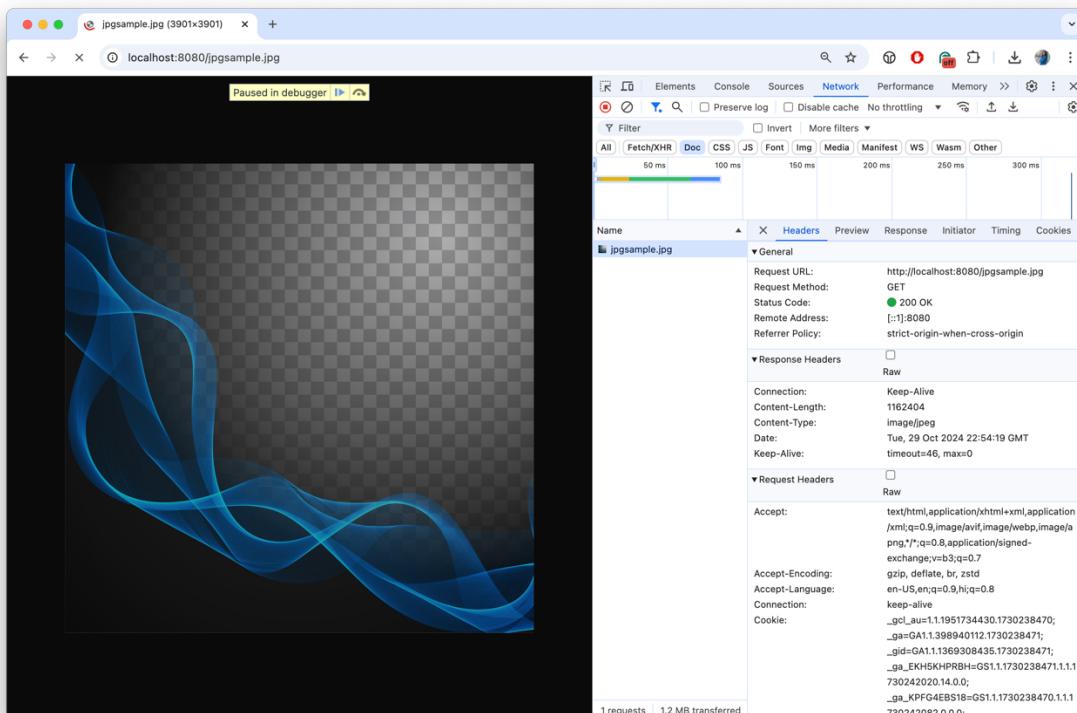
- Connection: Keep-Alive
- Content-Length: 59753
- Content-Type: image/svg+xml
- Date: Tue, 29 Oct 2024 22:55:25 GMT
- Keep-Alive: timeout=46, max=0

The "Request Headers" section shows:

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
- Accept-Encoding: gzip, deflate, br, zstd
- Accept-Language: en-US,en;q=0.9,hi;q=0.8
- Connection: keep-alive
- Cookie: _gcl_au=1.1951734430.1730238470; _ga=GA1.1.398940112.1730238471; _gid=GAI.1.1369308435.1730238471; _ga_EKH5KHPRBH=GS1.1.1730238471.1.1.1730242020.14.0.; _ga_KPFGE4EB51b=GS1.1.1730238470.1.1.1730242082.0.0.0;

At the bottom of the Network tab, it says "1 requests | 59.9 kB transferred".

Accessing localhost:8080/jpgsample.jpg, displays the image correctly



The screenshot shows a browser window with the URL `localhost:8080/jpgsample.jpg`. The main content area displays a blue wavy abstract graphic against a black background. To the right of the browser is the Chrome DevTools Network tab, which is paused in a debugger. The Network tab shows a single request for the JPEG file. The "General" section of the request details shows:

- Request URL: `http://localhost:8080/jpgsample.jpg`
- Request Method: GET
- Status Code: 200 OK
- Remote Address: [::]:8080
- Referrer Policy: strict-origin-when-cross-origin

The "Response Headers" section shows:

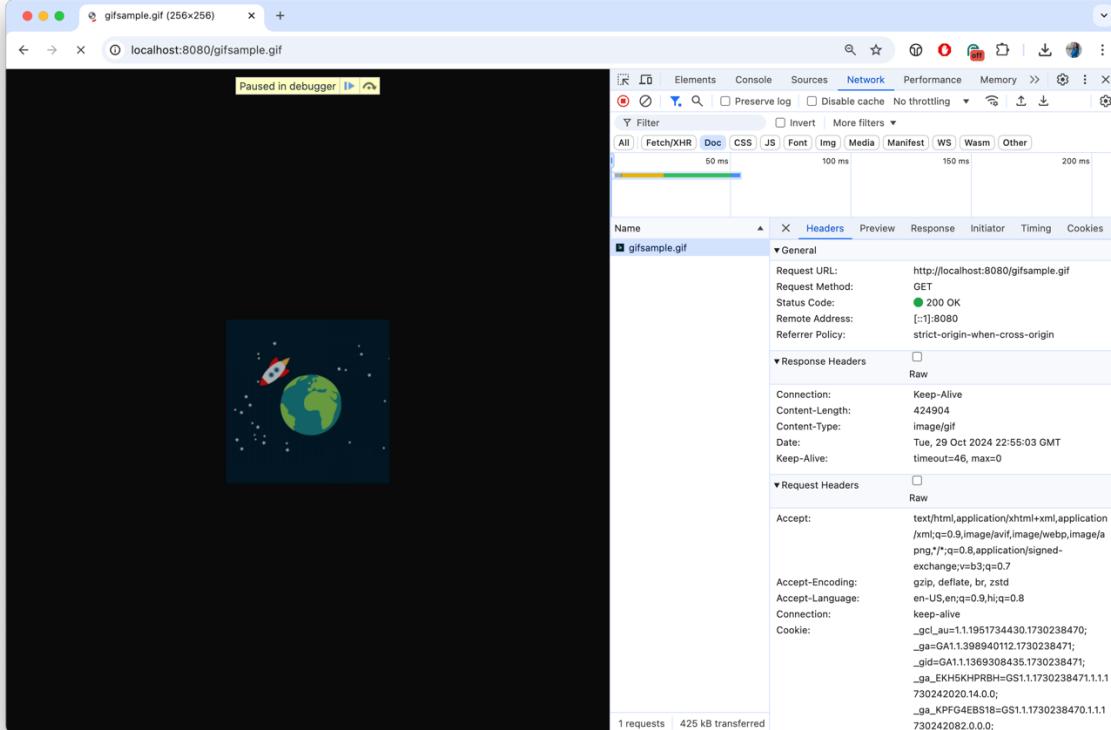
- Connection: Keep-Alive
- Content-Length: 1162404
- Content-Type: image/jpeg
- Date: Tue, 29 Oct 2024 22:54:19 GMT
- Keep-Alive: timeout=46, max=0

The "Request Headers" section shows:

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
- Accept-Encoding: gzip, deflate, br, zstd
- Accept-Language: en-US,en;q=0.9,hi;q=0.8
- Connection: keep-alive
- Cookie: _gcl_au=1.1951734430.1730238470; _ga=GA1.1.398940112.1730238471; _gid=GAI.1.1369308435.1730238471; _ga_EKH5KHPRBH=GS1.1.1730238471.1.1.1730242020.14.0.; _ga_KPFGE4EB51b=GS1.1.1730238470.1.1.1730242082.0.0.0;

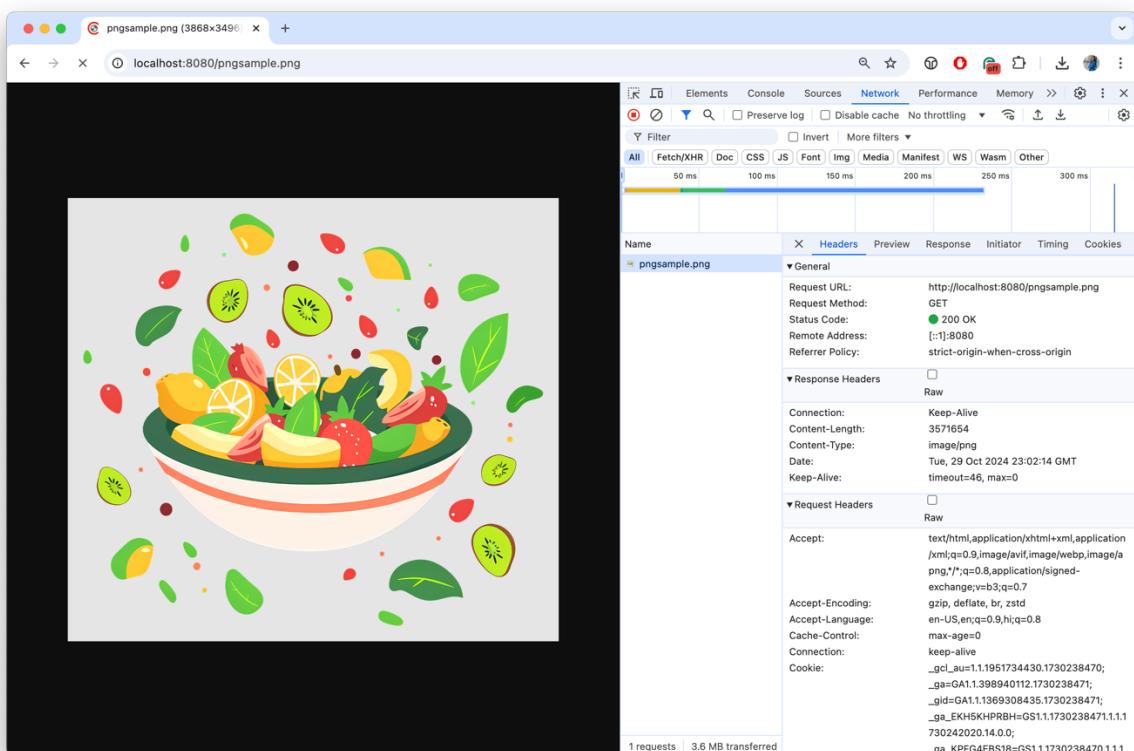
At the bottom of the Network tab, it says "1 requests | 1.2 MB transferred".

Accessing localhost:8080/gifsample.gif, displays the gif correctly



A screenshot of a web browser window. The address bar shows "localhost:8080/gifsample.gif". The main content area displays a small animated GIF of a rocket launching from Earth. To the right of the image is the browser's developer tools Network tab. The "Headers" section is selected. A request for "gifsample.gif" is listed. The "General" row shows the Request URL as "http://localhost:8080/gifsample.gif", Request Method as "GET", Status Code as "200 OK", and Remote Address as "[::1]:8080". The "Response Headers" section lists Connection: "Keep-Alive", Content-Length: "424904", Content-Type: "image/gif", Date: "Tue, 29 Oct 2024 22:55:03 GMT", and Keep-Alive: "timeout=46, max=0". The "Request Headers" section shows various browser and Google Analytics cookies.

Accessing localhost:8080/pngsample.png, displays the image correctly



A screenshot of a web browser window. The address bar shows "localhost:8080/pngsample.png". The main content area displays a vibrant illustration of a fruit salad bowl with various fruits like kiwi, orange, and strawberry. To the right of the image is the browser's developer tools Network tab. The "Headers" section is selected. A request for "pngsample.png" is listed. The "General" row shows the Request URL as "http://localhost:8080/pngsample.png", Request Method as "GET", Status Code as "200 OK", and Remote Address as "[::1]:8080". The "Response Headers" section lists Connection: "Keep-Alive", Content-Length: "3571654", Content-Type: "image/png", Date: "Tue, 29 Oct 2024 23:02:14 GMT", and Keep-Alive: "timeout=46, max=0". The "Request Headers" section shows various browser and Google Analytics cookies.

Accessing localhost:8080/htmlsample.html, displays the html file correctly

The screenshot shows the Network tab of a browser developer tools interface. A single request for "HtmlSample.html" is listed. The request URL is `http://localhost:8080/HtmlSample.html`, the method is GET, and the status code is 200 OK. The response headers include:

Name	Value
Request-URL	<code>http://localhost:8080/HtmlSample.html</code>
Status-Code	200 OK
Remote-Address	[::1]:8080
Referrer-Policy	strict-origin-when-cross-origin
Content-Type	text/html
Date	Tue, 29 Oct 2024 22:55:54 GMT
Keep-Alive	timeout=46, max=0
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding	gzip, deflate, br, zstd
Accept-Language	en-US,en;q=0.9,hi;q=0.8
Connection	keep-alive
Cookie	_gcl_au=1.19951734430.1730238470; _ga=GA1.1.398940112.1730238471; _gid=GA1.1.369308435.1730238471; _ga_EKH5KHPRBH=GS1.1.1730238471.1.1.1730242020.14.0; _ga_KPFG4EBS18=GS1.1.1730238470.1.1.1730242082.0.0; _ga_1.398940112.1730238471; _gid=GA1.1.369308435.1730238471; _ga_EKH5KHPRBH=GS1.1.1730238471.1.1.1730242020.14.0; _ga_KPFG4EBS18=GS1.1.1730238470.1.1.1730242082.0.0;

Accessing localhost:8080/txtsample.txt, displays the text file correctly

The screenshot shows the Network tab of a browser developer tools interface. A single request for "txtsample.txt" is listed. The request URL is `http://localhost:8080/txtsample.txt`, the method is GET, and the status code is 200 OK. The response headers include:

Name	Value
Request-URL	<code>http://localhost:8080/txtsample.txt</code>
Status-Code	200 OK
Remote-Address	[::1]:8080
Referrer-Policy	strict-origin-when-cross-origin
Content-Type	text/plain
Date	Tue, 29 Oct 2024 22:54:43 GMT
Keep-Alive	timeout=8, max=0
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding	gzip, deflate, br, zstd
Accept-Language	en-US,en;q=0.9,hi;q=0.8
Connection	keep-alive
Cookie	_gcl_au=1.19951734430.1730238470; _ga=GA1.1.398940112.1730238471; _gid=GA1.1.369308435.1730238471; _ga_EKH5KHPRBH=GS1.1.1730238471.1.1.1730242020.14.0; _ga_KPFG4EBS18=GS1.1.1730238470.1.1.1730242082.0.0; _ga_1.398940112.1730238471; _gid=GA1.1.369308435.1730238471; _ga_EKH5KHPRBH=GS1.1.1730238471.1.1.1730242020.14.0; _ga_KPFG4EBS18=GS1.1.1730238470.1.1.1730242082.0.0;

Accessing localhost:8080/jsonsample.json, displays the json file correctly

The screenshot shows a browser window with the URL `localhost:8080/jsonsample.json`. The developer tools Network tab is open, displaying a single request for `jsonsample.json`. The request details show a successful `GET` request with a status code of `200 OK`. The response headers include `Content-Type: application/json`, `Date: Tue, 29 Oct 2024 22:58:40 GMT`, and `Keep-Alive: timeout=48, max=0`. The response body is a JSON object:

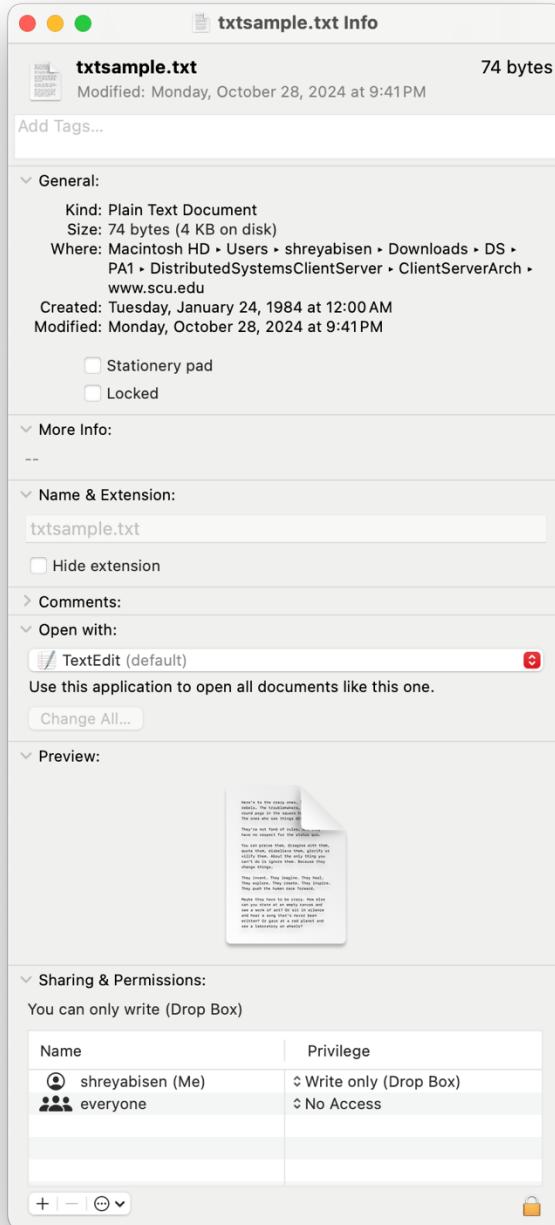
```
{  
  "user_id": "shreyabisen",  
  "email": "sbisen@scu.edu",  
  "name": "Shreya"  
}
```

Error Codes

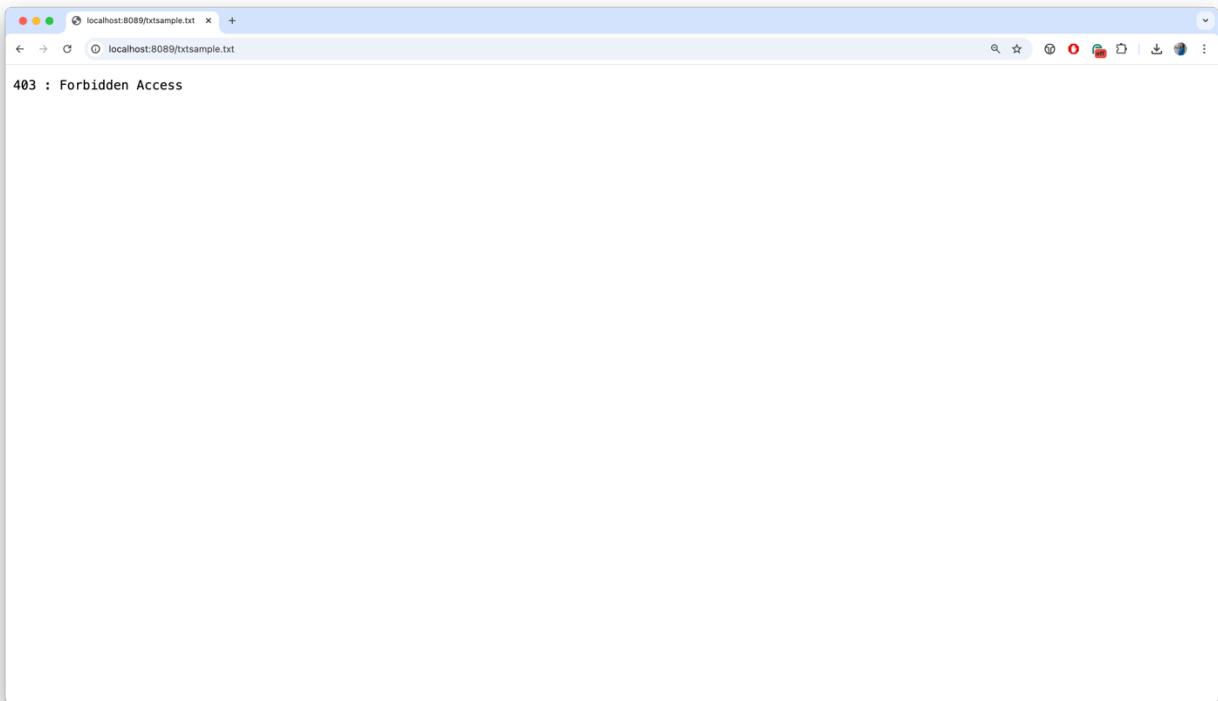
The command `chmod a-r filename.txt` on a **removes the read permissions** for all users on the file `filename.txt`. Here, I am trying to change the permissions of file `txtsample.txt`, removing its read permission. Accordingly my code restricts access of this file and shows a 403 Forbidden Access Error (displayed on the web browser as well as network console).

```
www.scu.edu -- zsh - 143x7  
(base) shreyabisen@Shreyas-MacBook-Air-3 ~ % cd /Users/shreyabisen/Downloads/DS/PA1/DistributedSystemsClientServer/ClientServerArch/www.scu.edu  
(base) shreyabisen@Shreyas-MacBook-Air-3 www.scu.edu % chmod a-r txtsample.txt  
  
(base) shreyabisen@Shreyas-MacBook-Air-3 www.scu.edu % ls -l txtsample.txt  
--w----- 1 shreyabisen staff 74 Oct 28 21:41 txtsample.txt  
(base) shreyabisen@Shreyas-MacBook-Air-3 www.scu.edu %
```

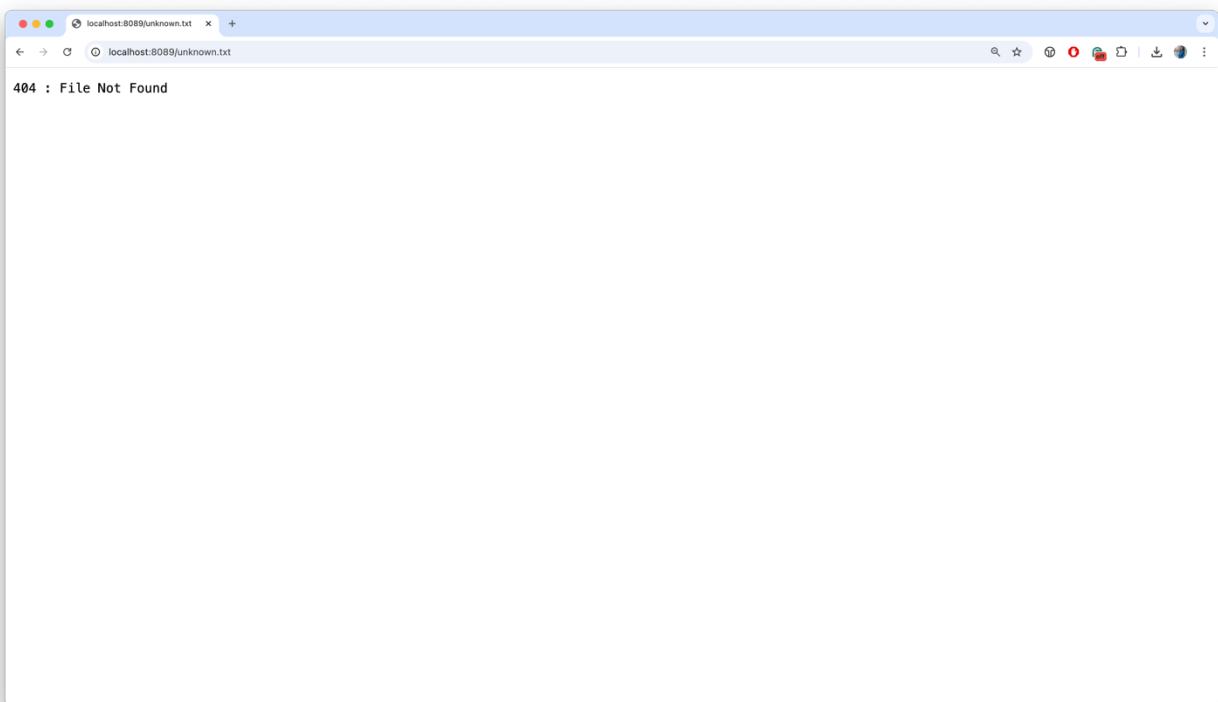
The file info shows, only write access to the file.



Accessing localhost:8080/txtsample.txt, displays the appropriate error (403 : Forbidden Access)



Accessing a non-existent file on localhost:8080/unknown.txt, displays the appropriate error (404 : File Not Found)



Below commands are tested on WSL (Linux environment on Windows)

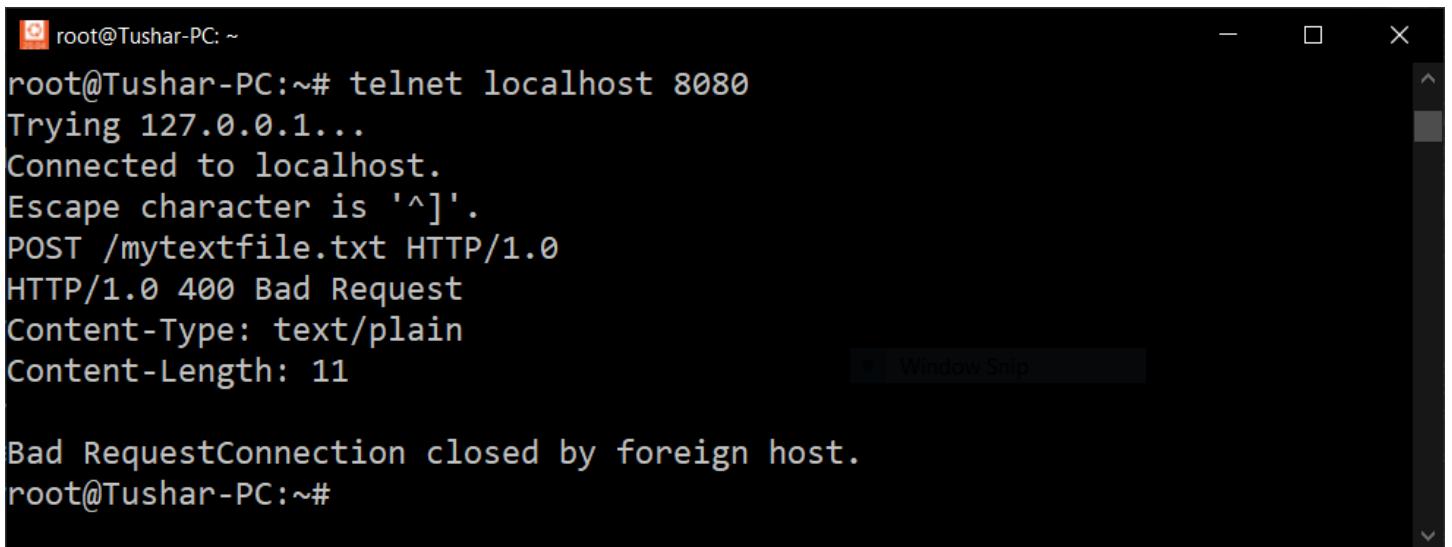
Trying to access files using POST method, throws “400 Bad Request” error, as the program only handles “GET” requests.

Appropriate headers are also displayed:

- Content-Type
- Content-Length

Command:

```
telnet localhost 8080  
POST /mytextfield.txt HTTP/1.0
```



```
root@Tushar-PC:~# telnet localhost 8080  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
POST /mytextfield.txt HTTP/1.0  
HTTP/1.0 400 Bad Request  
Content-Type: text/plain  
Content-Length: 11  
  
Bad RequestConnection closed by foreign host.  
root@Tushar-PC:~#
```

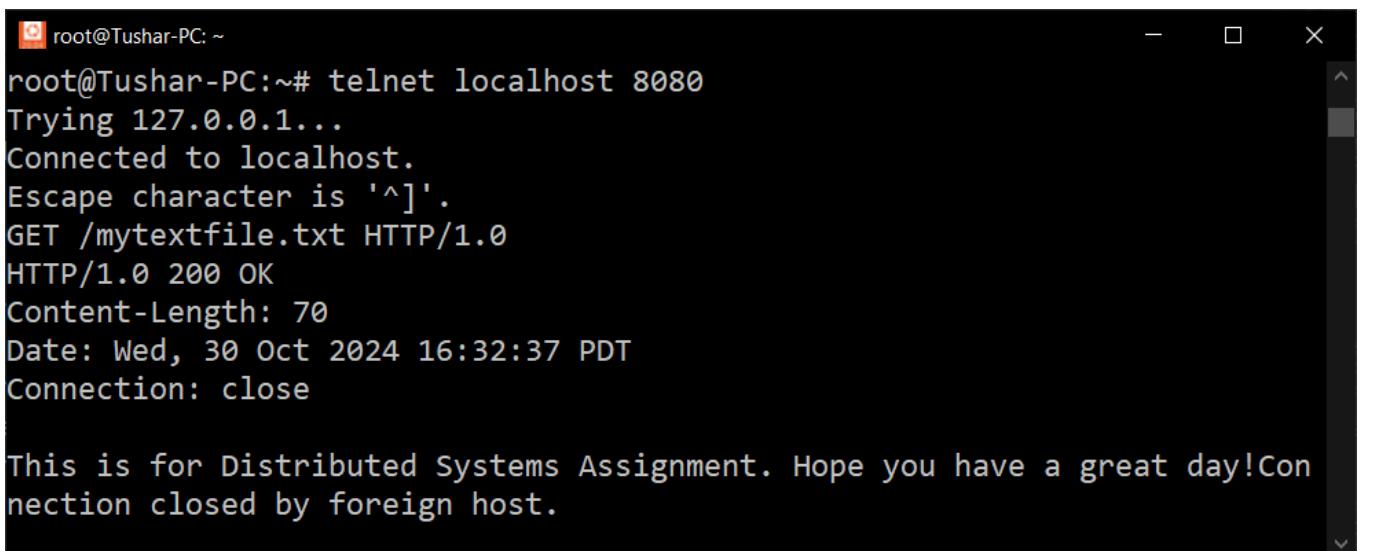
Requests with HTTP/1.0 protocol, gets closed instantly after serving a request.

Appropriate headers are also displayed:

- Content-Length
- Date

Command:

```
telnet localhost 8080  
GET /mytextfield.txt HTTP/1.0
```



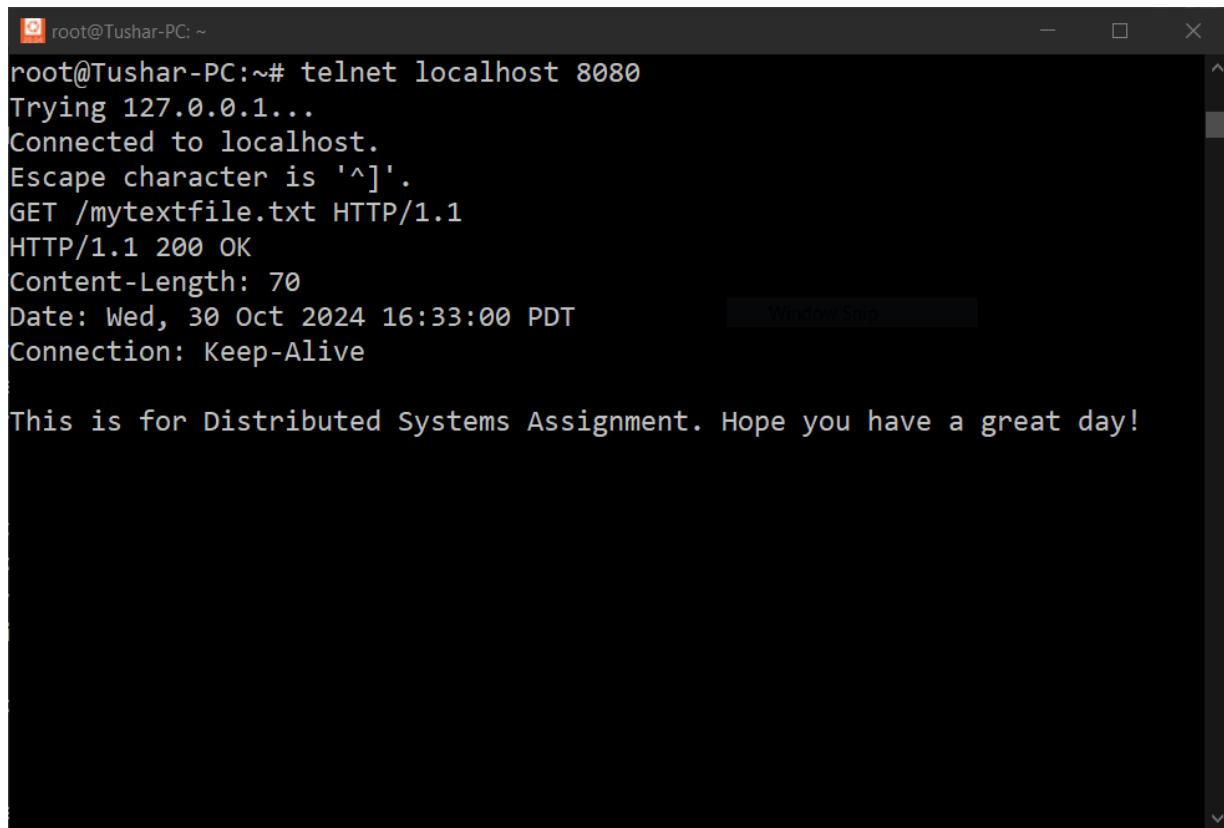
```
root@Tushar-PC:~# telnet localhost 8080  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
GET /mytextfield.txt HTTP/1.0  
HTTP/1.0 200 OK  
Content-Length: 70  
Date: Wed, 30 Oct 2024 16:32:37 PDT  
Connection: close  
  
This is for Distributed Systems Assignment. Hope you have a great day!Connection closed by foreign host.
```

Requests with HTTP/1.1 protocol, remains Alive for a certain period based on the timeout set according to `numberOfConnections`

Command:

telnet localhost 8080

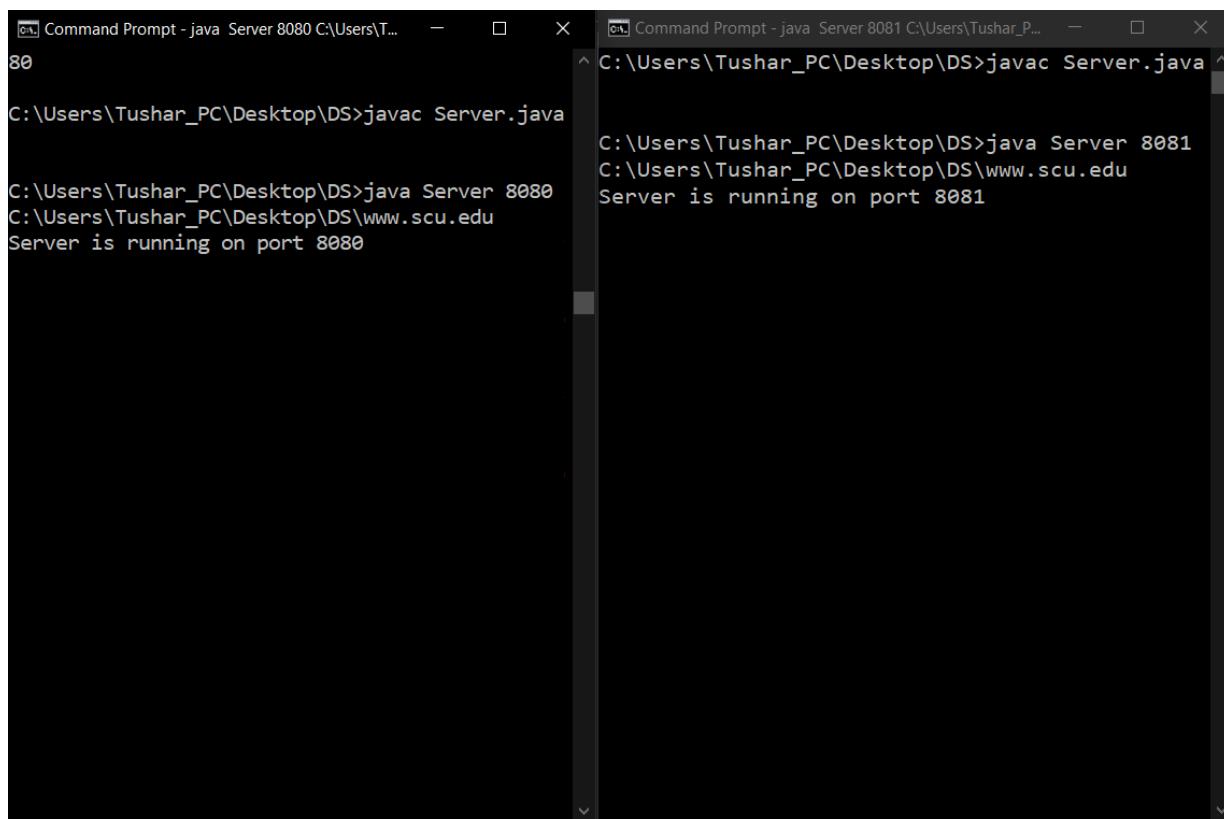
GET /mytextfield.txt HTTP/1.1



```
root@Tushar-PC:~# telnet localhost 8080
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /mytextfield.txt HTTP/1.1
HTTP/1.1 200 OK
Content-Length: 70
Date: Wed, 30 Oct 2024 16:33:00 PDT
Connection: Keep-Alive

This is for Distributed Systems Assignment. Hope you have a great day!
```

Web server, serving 2 ports (8080 & 8081) simultaneously



```
Command Prompt - java Server 8080 C:\Users\Tushar_P... 80
C:\Users\Tushar_PC\Desktop\DS>javac Server.java
C:\Users\Tushar_PC\Desktop\DS>java Server 8080
C:\Users\Tushar_PC\Desktop\DS\www.scu.edu
Server is running on port 8080

Command Prompt - java Server 8081 C:\Users\Tushar_P... 80
C:\Users\Tushar_PC\Desktop\DS>javac Server.java
C:\Users\Tushar_PC\Desktop\DS>java Server 8081
C:\Users\Tushar_PC\Desktop\DS\www.scu.edu
Server is running on port 8081
```

Logs of requests client made to the server

TERMINAL

```
Server is running on port 8089
REQUEST: GET / HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/stylesheets/f.css HTTP/1.1
REQUEST: GET /assets/public/scu.js HTTP/1.1
REQUEST: GET /media/external/brand-rollout-announcement/sun-red.svg HTTP/1.1
REQUEST: GET /index.html HTTP/1.1
REQUEST: GET /index.html HTTP/1.1
REQUEST: GET /assets/public/scu.js HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/stylesheets/f.css HTTP/1.1
REQUEST: GET /index.html HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/stylesheets/f.css HTTP/1.1
REQUEST: GET /index.html HTTP/1.1
REQUEST: GET /assets/public/scu.js HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/stylesheets/f.css HTTP/1.1
REQUEST: GET /index.html HTTP/1.1
REQUEST: GET /assets/public/scu.js HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/stylesheets/f.css HTTP/1.1
REQUEST: GET /index.html HTTP/1.1
REQUEST: GET /assets/public/scu.js HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/stylesheets/f.css HTTP/1.1
REQUEST: GET /index.html HTTP/1.1
REQUEST: GET /assets/public/scu.js HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/stylesheets/f.css HTTP/1.1
REQUEST: GET /index.html HTTP/1.1
REQUEST: GET /assets/public/scu.js HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/stylesheets/f.css HTTP/1.1
REQUEST: GET /index.html HTTP/1.1
REQUEST: GET /assets/public/scu.js HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/stylesheets/f.css HTTP/1.1
REQUEST: GET /index.html HTTP/1.1
REQUEST: GET /assets/public/scu.js HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/stylesheets/f.css HTTP/1.1
REQUEST: GET /index.html HTTP/1.1
REQUEST: GET /assets/public/scu.js HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/javascript/lazyloading.js HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/stylesheets/preloading.css HTTP/1.1
REQUEST: GET /media/external/brand-rollout-announcement/scu_aquaponics-20190521-jag-1127_img-web.jpg HTTP/1.1
REQUEST: GET /media/external/brand-rollout-announcement/scu-vs-gonzaga-50_img-web-1160x831.jpg HTTP/1.1
REQUEST: GET /appdata/site-alerts.json?c=0.3008319693880146&callback=jsonCallback&_=1730350855141 HTTP/1.1
REQUEST: GET /media/external/brand-rollout-announcement/amelia_fuller_lab_20231115_jg_1219_img-web-759x545.jpg HTTP/1.1
REQUEST: GET /media/external/brand-rollout-announcement/business-commencement-3534_img-web-1160x831.jpg HTTP/1.1
REQUEST: GET /media/external/brand-rollout-announcement/scu-mission-church_img-web-759x545.jpg HTTP/1.1
REQUEST: GET /media/external/brand-rollout-announcement/global-village-2024-60_img-web-759x545.jpg HTTP/1.1
REQUEST: GET /media/santa-clara-rising/SCU_Rising_logo_Alt_360_WEB.png HTTP/1.1
REQUEST: GET /media/external/brand-rollout-announcement/sun-red.svg HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/javascript/lazyloading.js HTTP/1.1
REQUEST: GET /media/scuedu/style-assets/stylesheets/preloading.css HTTP/1.1
REQUEST: GET /media/external/brand-rollout-announcement/beige-badge-background-icon.png HTTP/1.1
REQUEST: GET /media/external/brand-rollout-announcement/scu_aquaponics-20190521-jag-1127_img-web.jpg HTTP/1.1
REQUEST: GET /appdata/site-alerts.json?c=0.7411687014666852&callback=jsonCallback&_=1730350855240 HTTP/1.1
```