CS425: Computer Networks
IIT Kanpur
Project1: Designing a Server with Persistent Connection
Date: Fri, Aug 19

Name: Neeraj Kumar
Roll: 13427
Email: neerajkr@iitk.ac.in

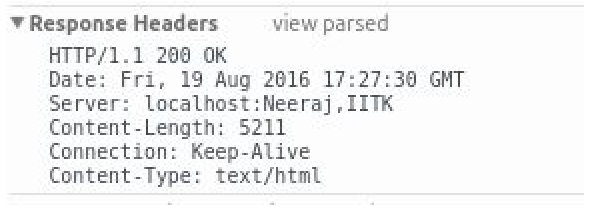
List of implemented options:

1. Allow the server port to be initialized at start up, via commandline
2. Include the Date and Server fields in the Response message header.

List and brief discussion of optional features:

1. **Allow the server port to be initialized at start up, via commandline**
    a. Port number has to be provided from command line i.e. ./http-server port_no e.g. http-server 5432
    b. I am reading this port number in main function using argv[]
2. **Include the Date and Server fields in the Response message header.**
    a. I am using <time.h> header file to send current time and date in GMT to response header
    b. I am also sending the response in server field

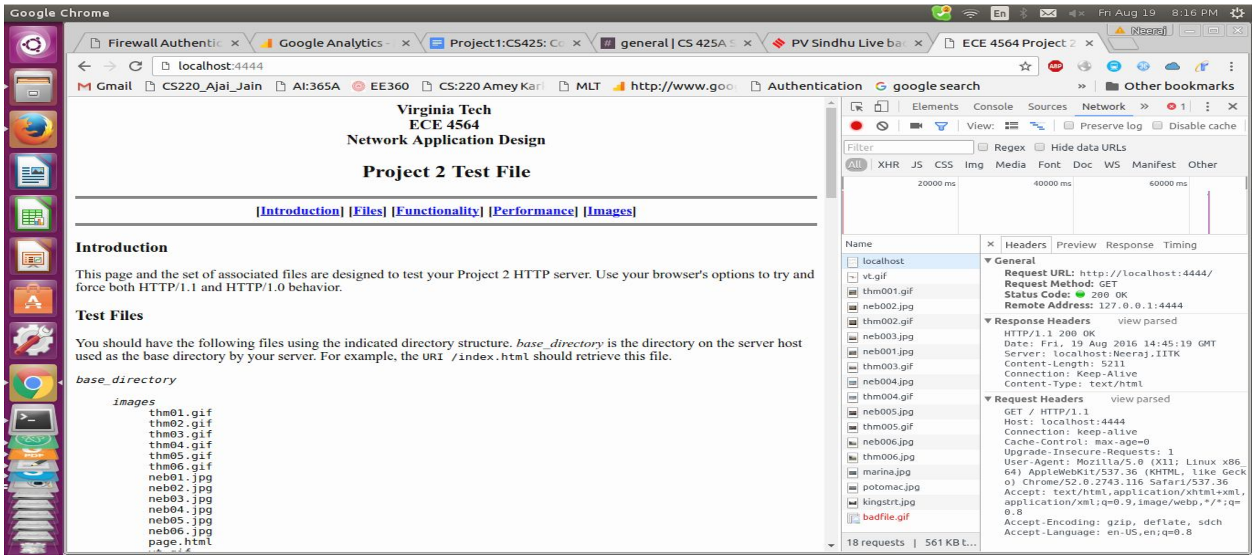
Fig. Below figure has been taken from the browser chrome



```
HTTP/1.1 200 OK
Date: Fri, 19 Aug 2016 17:27:30 GMT
Server: localhost:Neeraj,IITK
Content-Length: 5211
Connection: Keep-Alive
Content-Type: text/html
```

<u>Testing Results:</u>

- After compiling and running the server on terminal, i tested with Chrome, Firefox browsers and also on Mobile phone
- I opened the server link with it's port number i.e. localhost:port/ to open index.html
- I opened localhost:4444/ and index.html file came as shown in below image
- I also checked the content of both request and response header and it was same as expected, also shown in below image



Virginia Tech
ECE 4564
Network Application Design

Project 2 Test File

[Introduction] [Files] [Functionality] [Performance] [Images]

Introduction

This page and the set of associated files are designed to test your Project 2 HTTP server. Use your browser's options to try and force both HTTP/1.1 and HTTP/1.0 behavior.

Test Files

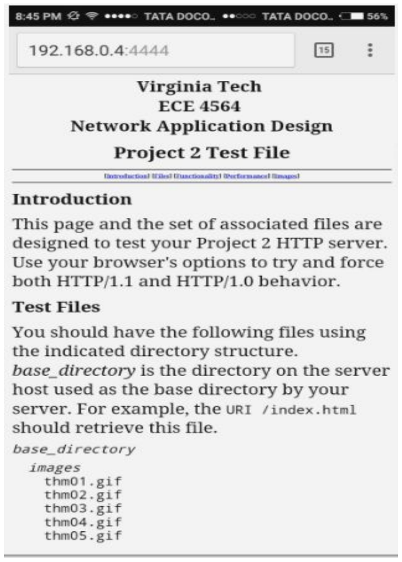
You should have the following files using the indicated directory structure. *base_directory* is the directory on the server host used as the base directory by your server. For example, the URI `/index.html` should retrieve this file.

```
base_directory

    images
        thm01.gif
        thm02.gif
        thm03.gif
        thm04.gif
        thm05.gif
        thm06.gif
        neb01.jpg
        neb02.jpg
        neb03.jpg
        neb04.jpg
        neb05.jpg
        neb06.jpg
        page.html
```

```
localhost
vt.gif
thm001.gif
neb002.jpg
thm002.gif
neb003.jpg
neb001.jpg
thm003.gif
neb004.jpg
thm004.gif
neb005.jpg
thm005.gif
neb006.jpg
thm006.jpg
marina.jpg
potomac.jpg
kingstrt.jpg
badfile.gif
18 requests | 561 KB t...
```

```
▼ General
    Request URL: http://localhost:4444/
    Request Method: GET
    Status Code: ● 200 OK
    Remote Address: 127.0.0.1:4444
▼ Response Headers      view parsed
    HTTP/1.1 200 OK
    Date: Fri, 19 Aug 2016 14:45:19 GMT
    Server: localhost:Neeraj,IITK
    Content-Length: 5211
    Connection: Keep-Alive
    Content-Type: text/html
▼ Request Headers      view parsed
    GET / HTTP/1.1
    Host: localhost:4444
    Connection: keep-alive
    Cache-Control: max-age=0
    Upgrade-Insecure-Requests: 1
    User-Agent: Mozilla/5.0 (X11; Linux x86_
    64) AppleWebKit/537.36 (KHTML, like Geck
    o) Chrome/52.0.2743.116 Safari/537.36
    Accept: text/html,application/xhtml+xml,
    application/xml;q=0.9,image/webp,*/*;q=
    0.8
    Accept-Encoding: gzip, deflate, sdch
    Accept-Language: en-US,en;q=0.8
```

- On mobile the output was as shown below:



Virginia Tech
ECE 4564
Network Application Design

Project 2 Test File

[Introduction] [Files] [Functionality] [Performance] [Images]

Introduction

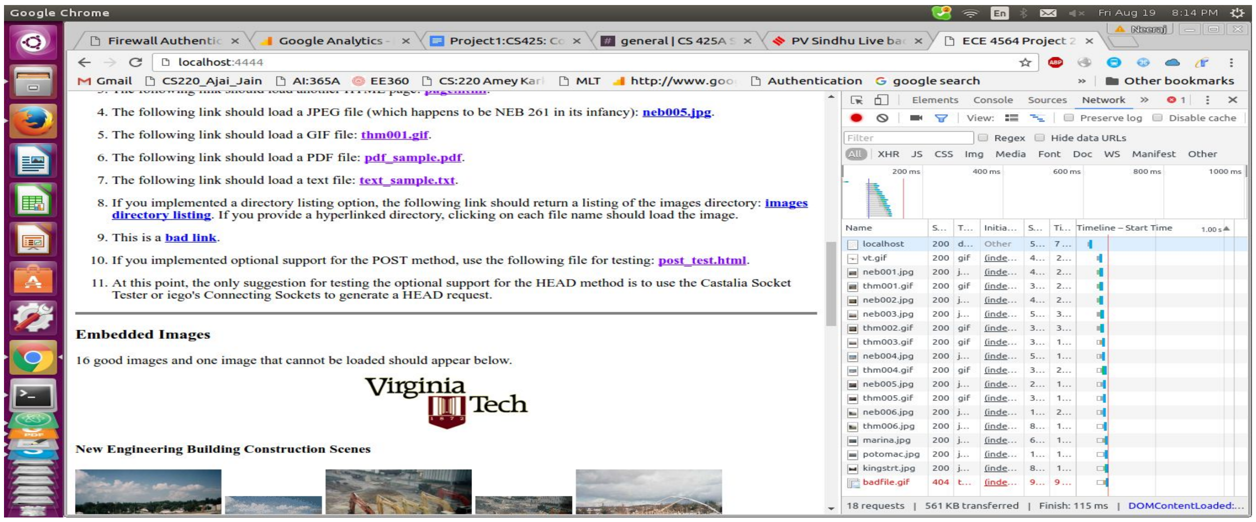
This page and the set of associated files are designed to test your Project 2 HTTP server. Use your browser's options to try and force both HTTP/1.1 and HTTP/1.0 behavior.

Test Files

You should have the following files using the indicated directory structure. *base_directory* is the directory on the server host used as the base directory by your server. For example, the URI `/index.html` should retrieve this file.

```
base_directory
  images
    thm01.gif
    thm02.gif
    thm03.gif
    thm04.gif
    thm05.gif
```

- I checked all the links hyperlinked on index.html and except directory listing(which is extra)



4. The following link should load a JPEG file (which happens to be NEB 261 in its infancy): neb005.jpg.
5. The following link should load a GIF file: thm001.gif.
6. The following link should load a PDF file: pdf_sample.pdf.
7. The following link should load a text file: text_sample.txt.
8. If you implemented a directory listing option, the following link should return a listing of the images directory: images directory listing. If you provide a hyperlinked directory, clicking on each file name should load the image.
9. This is a bad link.
10. If you implemented optional support for the POST method, use the following file for testing: post_test.html.
11. At this point, the only suggestion for testing the optional support for the HEAD method is to use the Castalia Socket Tester or iego's Connecting Sockets to generate a HEAD request.

Embedded Images

16 good images and one image that cannot be loaded should appear below.

New Engineering Building Construction Scenes

Name	S...	T...	Initia...	S...	Ti...
localhost	200	d...	Other	5...	7 ...
vt.gif	200	gif	(inde...	4...	2...
neb001.jpg	200	j...	(inde...	4...	2...
thm001.gif	200	gif	(inde...	3...	2...
neb002.jpg	200	j...	(inde...	4...	2...
neb003.jpg	200	j...	(inde...	5...	3...
thm002.gif	200	gif	(inde...	3...	3...
thm003.gif	200	gif	(inde...	3...	1...
neb004.jpg	200	j...	(inde...	5...	1...
thm004.gif	200	gif	(inde...	3...	2...
neb005.jpg	200	j...	(inde...	2...	1...
thm005.gif	200	gif	(inde...	3...	1...
neb006.jpg	200	j...	(inde...	1...	2...
thm006.jpg	200	j...	(inde...	8...	1...
marina.jpg	200	j...	(inde...	6...	1...
potomac.jpg	200	j...	(inde...	1...	1...
kingstrt.jpg	200	j...	(inde...	8...	1...
badfile.gif	404	t...	(inde...	9...	9 ...

18 requests | 561 KB transferred | Finish: 115 ms | DOMContentLoaded:...

Note: More screenshots are attached in Appendix

Appendix:

Sourcefile Code:

```
/* Server Program in which port number is provided at the terminal */

#include <stdio.h>                    /*libraries*/
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>                //contains definitions of data types used in system call
#include <sys/socket.h>               //definition needed for socket structures
#include <netinet/in.h>               //contains values of constants used in defition of Internet Domain Addresses
#include <iostream>
#include <cstdio>
#include<strings.h>
#include <sys/stat.h>
#include <time.h>
#include <string>

#define BUF_SIZE 8192             //Buffer size
#define conf_file "Initial.conf"
using namespace std;

char root[200], extra[200]  ;

int status_code,content_length;              //Initialization of variables
string content_type;
socklen_t clilen;
int sockfd, newsockfd, portno;          //variables for socket and port no
struct sockaddr_in serv_addr, cli_addr;          //data structure to store internet addresses


string date_time_header(void){                    // function for getting current time

  char buf[1000];
  time_t now = time(0);
  struct tm tm = *gmtime(&now);
  strftime(buf, sizeof buf, "%a, %d %b %Y %H:%M:%S %Z", &tm);
  string str = string(buf);
  str="Date: "+str+"\r\n";
  std::cout << str << std::endl;
  return str;
}
```

```
size_t getFilesize(const char* filename) {                    //function for getting file size of local file
    struct stat st;
    if(stat(filename, &st) != 0) {
        return 0;
    }
    return st.st_size;
}

void file_length(FILE *file){                       //function for getting file length
  fseek (file, 0, SEEK_END);                            //take the file pointer to last of text file
  content_length=ftell(file);                //find the file length
  std::cout << "content_length:" <<content_length << std::endl;
  fseek (file, 0, SEEK_SET);
}

void type_of_content(string fn){                            //function for geeting cotent type
  if(fn.substr(fn.find_last_of(".") + 1) == "html")
    content_type="Content-Type: text/html\r\n";
  else if(fn.substr(fn.find_last_of(".") + 1) == "htm")
    content_type="Content-Type: text/html\r\n";
  else if(fn.substr(fn.find_last_of(".") + 1) == "txt")
    content_type="Content-Type: text/plain\r\n";
  else if(fn.substr(fn.find_last_of(".") + 1) == "jpeg")
    content_type="Content-Type: image/jpeg\r\n";
  else if(fn.substr(fn.find_last_of(".") + 1) == "jpg")
    content_type="Content-Type: image/jpeg\r\n";
  else if(fn.substr(fn.find_last_of(".") + 1) == "gif")
    content_type="Content-Type: image/gif\r\n";
  else if(fn.substr(fn.find_last_of(".") + 1) == "pdf")
    content_type="Content-Type: Application/pdf\r\n";
  else content_type="Content-Type: text/html\r\n";
  std::cout << content_type << std::endl;
}

string status_msg(int status_code){                          //function for status msg
  switch (status_code) {
    case 200: return  "200 OK\r\n";
    case 400: return  "400 Bad Request\r\n";
    case 404: return  "404 Not Found\r\n";
    case 500: return  "500 Internal Server Error\r\n";
    case 501: return  "501 Not Implemeted\r\n";
    default: return   "200 OK\r\n";
  }
}

void send_response_header_404(void){                  //header response
  status_code=404;
  string http_version="HTTP/1.1 ";
  string status=status_msg(status_code);
  string headers;
  string dateTime=date_time_header();
  string server="Server: localhost:Neeraj,IITK\r\n";
  headers=http_version+status+dateTime+server+"\r\n";

  char Headers[4024] ;
  memset(Headers, '\0', sizeof(Headers));
  std::string::size_type i;
  for( i=0;i < headers.size(); ++i) {
    Headers[i]=headers[i];
  }
  //printf("%s\n", Headers);
  write(newsockfd,Headers,strlen(Headers));
}

void send_response_header(void){                    //header response
  status_code=200;
  string http_version="HTTP/1.1 ";
  string status=status_msg(status_code);
  char buf[50];
  sprintf(buf, "%d", content_length);
  string ContentLength="Content-Length: "+string(buf)+"\r\n";
  string connection_typ="Connection: Keep-Alive\r\n";
  string headers;
  string server="Server: localhost:Neeraj,IITK\r\n";
  string dateTime=date_time_header();
  headers=http_version+status+dateTime+server+ContentLength+connection_typ+content_type+"\r\n";
  std::cout << content_type << std::endl;
  std::cout << headers << std::endl;
  char Headers[4024] ;
  memset(buf, '\0', sizeof(buf))  ;
  memset(Headers, '\0', sizeof(Headers));
  std::string::size_type i;
  for( i=0;i < headers.size(); ++i) {
    Headers[i]=headers[i];
  }
  //printf("%s\n", Headers);
  cout <<strlen(Headers) <<endl;
  write(newsockfd,Headers,strlen(Headers));
}

FILE * file_open(string filename){               //function to open file
  FILE *f;
  if (filename=="/") {
      f = fopen ("index.html", "rb");                  //open the file and give handle to file pointer f
      return f;
  }
  else if(filename.back()=='/'){
    filename=filename+"index.html";
    filename.erase(0,1);
    f = fopen (filename.c_str(), "rb");               //open the file and give handle to file pointer f
    if(f==NULL) std::cout << "file not opened" << std::endl;
    return f;

  }
  else{
     filename.erase(0,1);
     //std::cout << filename << std::endl;
     f = fopen (filename.c_str(), "rb");                    //open the file and give handle to file pointer f
     if(f==NULL) std::cout << "file not opened" << std::endl;
     return f;
  }
}

void error(const char *err)      //error msg printing
{
    perror(err);
    exit(1);
}

int main(int argc, char *argv[])                    //main function and port number is taken in its argument
{
        char buffer[BUF_SIZE];                              //acts as a buffer
        int n;
        sockfd = socket(AF_INET, SOCK_STREAM, 0);           //sys call to create a new socket for TCP connection
        if (sockfd < 0)    error("socket creation error\n"); //error for unsuccesful creation of socket
        bzero((char *) &serv_addr, sizeof(serv_addr));  //initialises serv_addr to zero on calling bzero() function
        portno = atoi(argv[1]);                             //extracting port number in nuerical form
        serv_addr.sin_family = AF_INET;                     //assigning values to serv_addr which is a structure of type
sockaddr_in;AF_INET is a symbolic contant
        serv_addr.sin_addr.s_addr = INADDR_ANY;             //symbolic constant INADDR_ANY, gets IP address of
server machine
        serv_addr.sin_port = htons(portno);                 //htons convert host byte order to network byte order
        if (bind(sockfd, (struct sockaddr *) &serv_addr,sizeof(serv_addr)) < 0)        error("ERROR on binding");
//binding socket to a address and error msg if failed to bind
        listen(sockfd,10);                  //it allows the system to listen for connections
        clilen = sizeof(cli_addr);

        static int counter=0;
        int pid;
   while(1){                         //once connected go in infinite loop for connecting with client
          newsockfd = accept(sockfd, (struct sockaddr *) &cli_addr, &clilen);  //blocking the process unless a client
connects to server
          if ((pid = fork()) == -1){
            close(newsockfd);
            continue;
          }
          else if(pid>0){
            close(newsockfd);
            counter++;
            printf("here2:%d\n",pid);
```

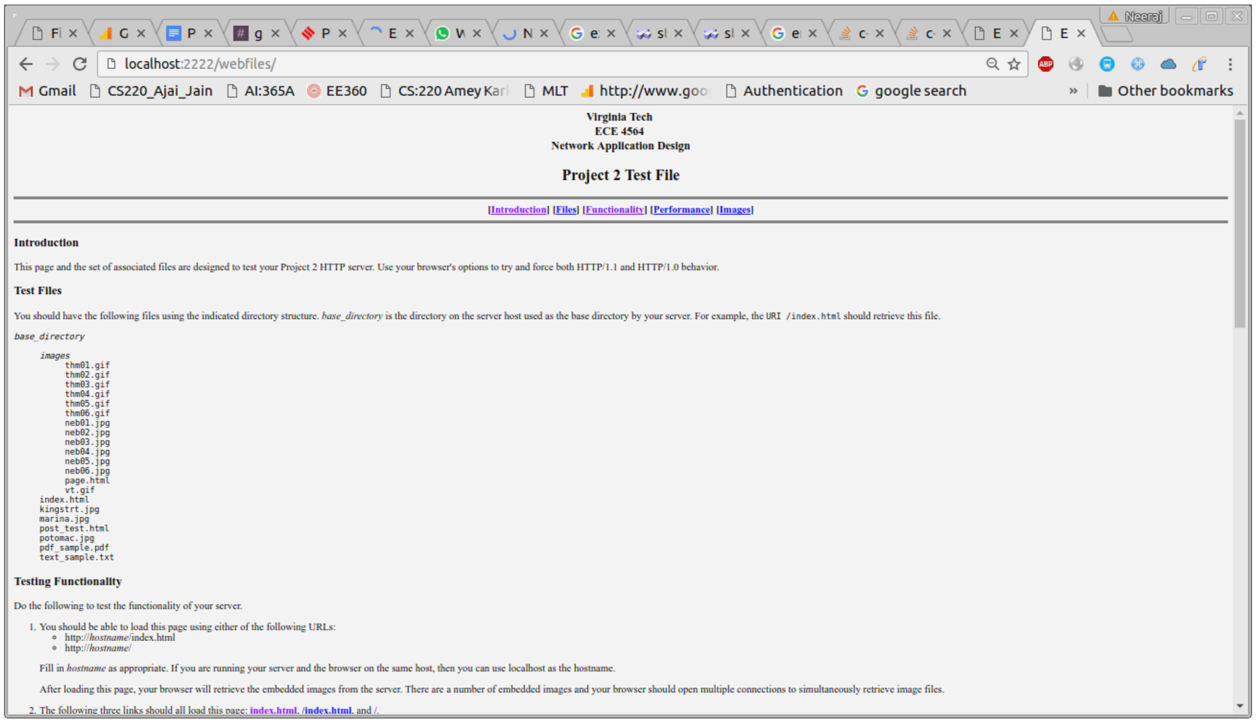
```
      continue;

    }
    else if(pid==0){
      printf("hereChild:%d\n",pid);
      bzero(buffer,BUF_SIZE);              //clearing the buffer to read file name from client side
      n = read(newsockfd,buffer,BUF_SIZE);
      if (n <= 0) error("ERROR reading from socket");

      //printf("File Name You are searching For:\n%s",buffer );                //printing the file name

      char * pch;
      pch = strtok (buffer," ");
      string request[100];
      int req_word_no=0;
      while (pch != NULL)
      {
        request[req_word_no]=pch;
        req_word_no++;
        pch = strtok (NULL, " ");
      }
      bzero(buffer,BUF_SIZE);
      FILE * f;
      string filename;
      filename=request[1];
      type_of_content(filename);
      f=file_open(filename);
      if(f==NULL){               //if file name is inavlid or file doesn't open send error acknowledgement as "0" to
client
        //n = write(newsockfd,"-1",sizeof("-1"));
        std::cout << "In main: File is NULL\n" << std::endl;
        send_response_header_404();
        close(newsockfd);
        break;
      }
      file_length(f);
      std::cout << "In main:" << content_type << std::endl;
      std::cout << "In main:\nsending Headers" << std::endl;
      send_response_header();
      fseek (f, 0, SEEK_SET);   //take file pointer to intial point
      while (!feof(f)) {              //send file one packet at a time
              bzero(buffer,BUF_SIZE);
              int bytes_read=fread (buffer, 1, BUF_SIZE, f);
              int bytes_sent=send(newsockfd, buffer, bytes_read, 0)  ;
        }
      std::cout << "In main:file has been sent\n" << std::endl;
      fclose (f);
      close(newsockfd);
      break;
    }
```

```
}
close(sockfd);
return 0;
}
```

Some of more screenshots after testing result:

Test Results and screenshots:

- All the images from index.html and other images were loading correctly loading
- Even when typing or going in any directory and putting '/' at the end of directory name, then if index.html file exists in that directory, it will open that file or else give an 404 error, as shown in below screenshot for webfiles/



Virginia Tech
ECE 4564
Network Application Design

Project 2 Test File

[Introduction] [Files] [Functionality] [Performance] [Images]

Introduction

This page and the set of associated files are designed to test your Project 2 HTTP server. Use your browser's options to try and force both HTTP/1.1 and HTTP/1.0 behavior.

Test Files

You should have the following files using the indicated directory structure. *base_directory* is the directory on the server host used as the base directory by your server. For example, the `URI /index.html` should retrieve this file.

```
base_directory

     images
          thm01.gif
          thm02.gif
          thm03.gif
          thm04.gif
          thm05.gif
          thm06.gif
          neb01.jpg
          neb02.jpg
          neb03.jpg
          neb04.jpg
          neb05.jpg
          neb06.jpg
          page.html
          vt.gif
     index.html
     kingstrt.jpg
     marina.jpg
     post_test.html
     potomac.jpg
     pdf_sample.pdf
     text_sample.txt
```

Testing Functionality

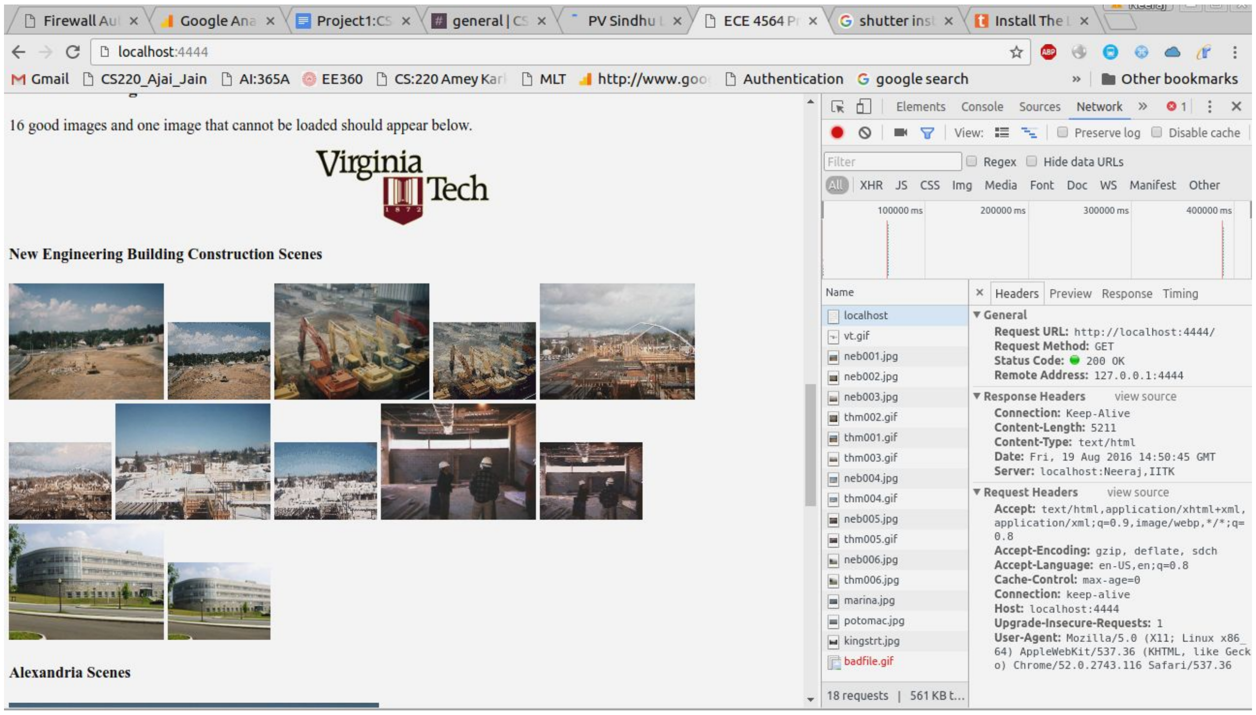
Do the following to test the functionality of your server.

1. You should be able to load this page using either of the following URLs:
   - http://*hostname*/index.html
   - http://*hostname*/

   Fill in *hostname* as appropriate. If you are running your server and the browser on the same host, then you can use localhost as the hostname.

   After loading this page, your browser will retrieve the embedded images from the server. There are a number of embedded images and your browser should open multiple connections to simultaneously retrieve image files.
2. The following three links should all load this page: **index.html**, **/index.html**, and /.

- This screenshot is showing that all the images are correctly loading and we can headers in this screenshot as well



localhost:4444

16 good images and one image that cannot be loaded should appear below.

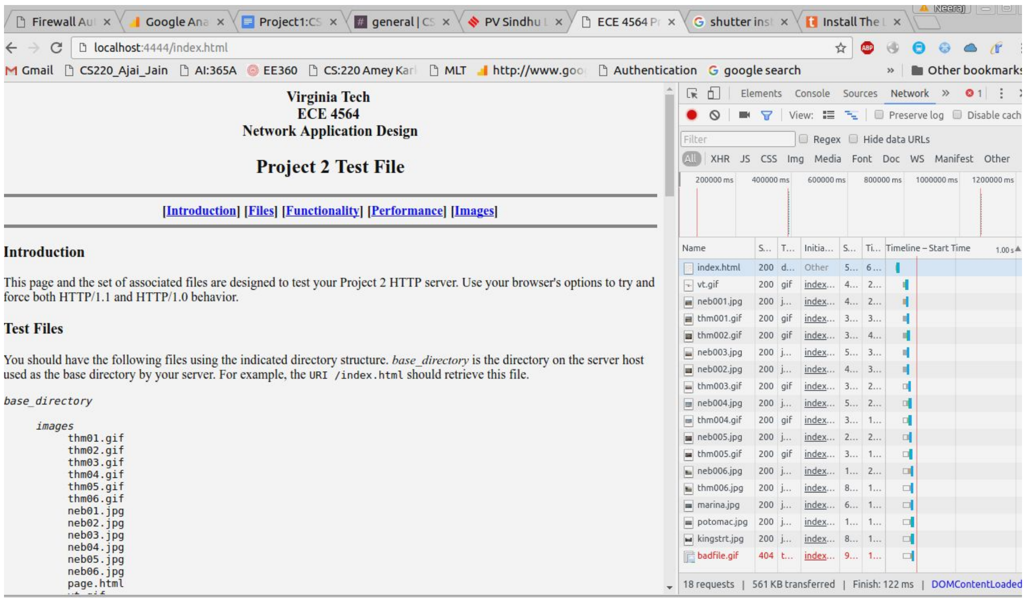
New Engineering Building Construction Scenes

Alexandria Scenes

```
Name
localhost
vt.gif
neb001.jpg
neb002.jpg
neb003.jpg
thm002.gif
thm001.gif
thm003.gif
neb004.jpg
thm004.gif
neb005.jpg
thm005.gif
neb006.jpg
thm006.jpg
marina.jpg
potomac.jpg
kingstrt.jpg
badfile.gif

18 requests | 561 KB t...
```

```
▼ General
    Request URL: http://localhost:4444/
    Request Method: GET
    Status Code: ● 200 OK
    Remote Address: 127.0.0.1:4444
▼ Response Headers     view source
    Connection: Keep-Alive
    Content-Length: 5211
    Content-Type: text/html
    Date: Fri, 19 Aug 2016 14:50:45 GMT
    Server: localhost:Neeraj,IITK
▼ Request Headers     view source
    Accept: text/html,application/xhtml+xml,
    application/xml;q=0.9,image/webp,*/*;q=
    0.8
    Accept-Encoding: gzip, deflate, sdch
    Accept-Language: en-US,en;q=0.8
    Cache-Control: max-age=0
    Connection: keep-alive
    Host: localhost:4444
    Upgrade-Insecure-Requests: 1
    User-Agent: Mozilla/5.0 (X11; Linux x86_
    64) AppleWebKit/537.36 (KHTML, like Geck
    o) Chrome/52.0.2743.116 Safari/537.36
```

- localhost:4444/index.html is also giving the same result as localhost:4444/ and localhost:4444 as in below screenshot



localhost:4444/index.html

Virginia Tech
ECE 4564
Network Application Design

Project 2 Test File

[Introduction] [Files] [Functionality] [Performance] [Images]

Introduction

This page and the set of associated files are designed to test your Project 2 HTTP server. Use your browser's options to try and force both HTTP/1.1 and HTTP/1.0 behavior.

Test Files

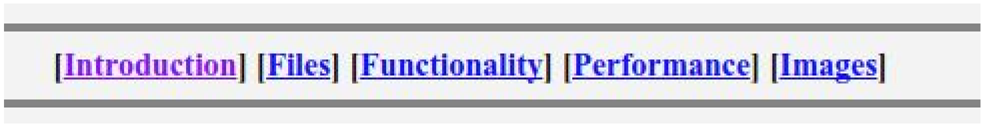
You should have the following files using the indicated directory structure. *base_directory* is the directory on the server host used as the base directory by your server. For example, the URI /index.html should retrieve this file.

```
base_directory

    images
        thm01.gif
        thm02.gif
        thm03.gif
        thm04.gif
        thm05.gif
        thm06.gif
        neb01.jpg
        neb02.jpg
        neb03.jpg
        neb04.jpg
        neb05.jpg
        neb06.jpg
        page.html
```

Name	S...	T...	Initia...	S...	Ti...
index.html	200	d...	Other	5...	6...
vt.gif	200	gif	index...	4...	2...
neb001.jpg	200	j...	index...	4...	2...
thm001.gif	200	gif	index...	3...	3...
thm002.gif	200	gif	index...	3...	4...
neb003.jpg	200	j...	index...	5...	3...
neb002.jpg	200	j...	index...	4...	3...
thm003.gif	200	gif	index...	3...	2...
neb004.jpg	200	j...	index...	5...	2...
thm004.gif	200	gif	index...	3...	1...
neb005.jpg	200	j...	index...	2...	2...
thm005.gif	200	gif	index...	3...	1...
neb006.jpg	200	j...	index...	1...	2...
thm006.jpg	200	j...	index...	8...	1...
marina.jpg	200	j...	index...	6...	1...
potomac.jpg	200	j...	index...	1...	1...
kingstrt.jpg	200	j...	index...	8...	1...
badfile.gif	404	t...	index...	9...	1...

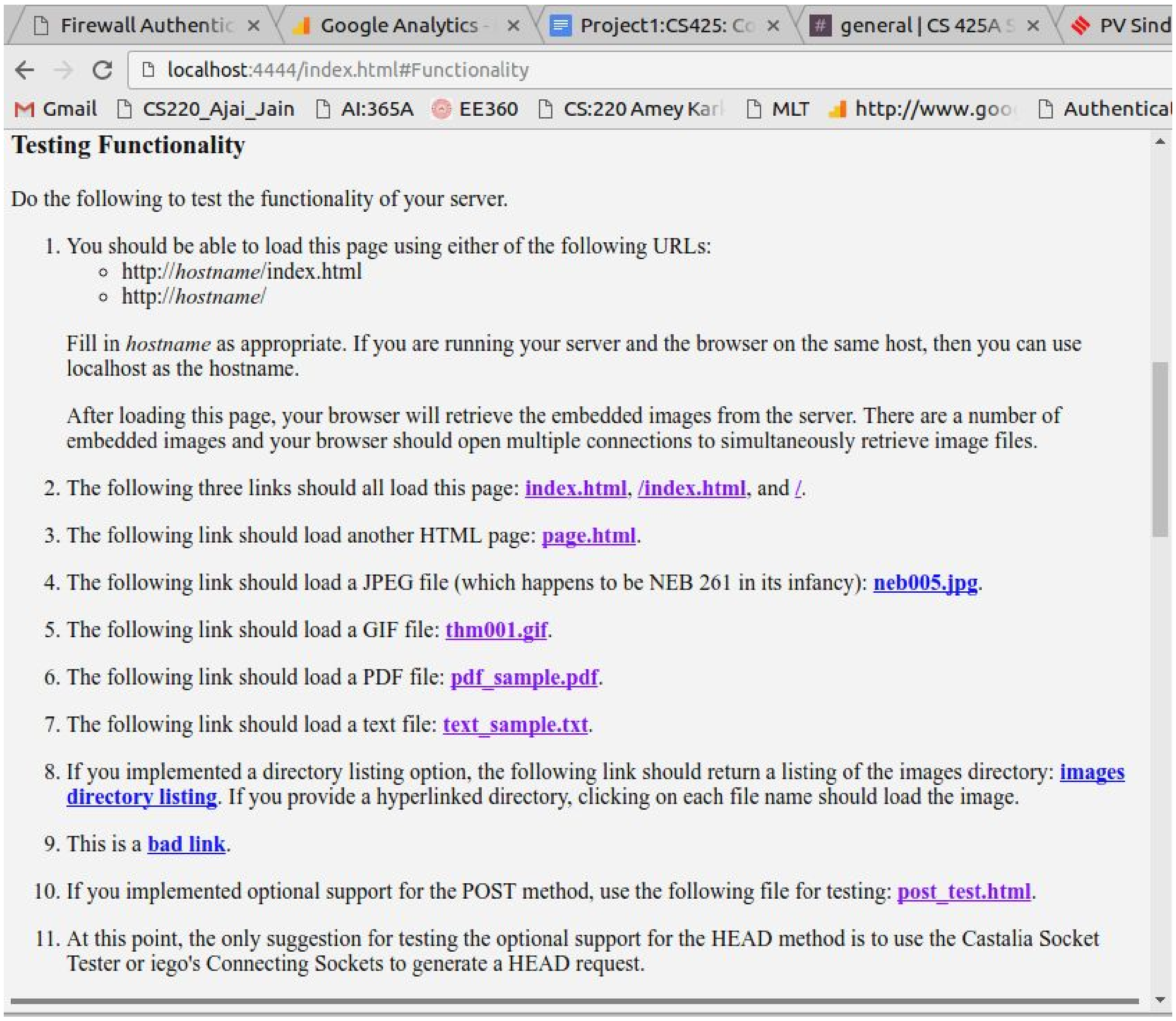
18 requests | 561 KB transferred | Finish: 122 ms | DOMContentLoaded

- Clicking on all the links of menu bar is working fine:



[Introduction] [Files] [Functionality] [Performance] [Images]

- For e.g. clicking on Functionality is taking me to below screenshot



Testing Functionality

Do the following to test the functionality of your server.

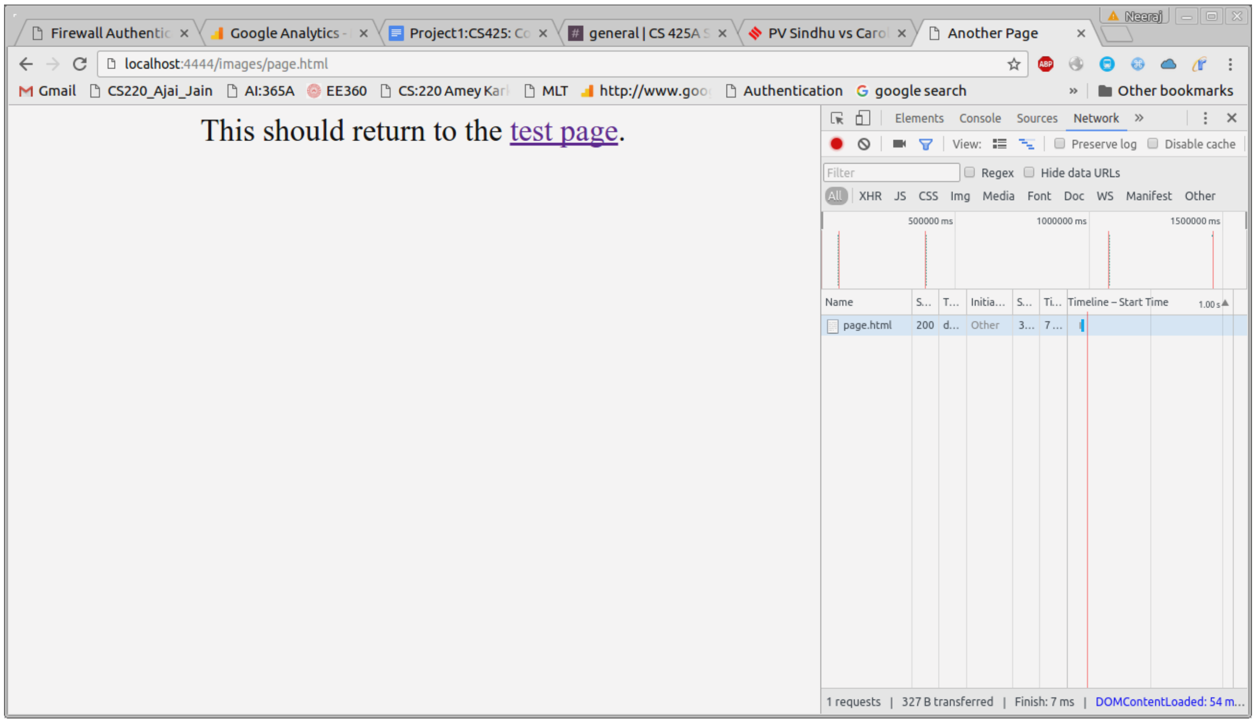
1. You should be able to load this page using either of the following URLs:
   - http://*hostname*/index.html
   - http://*hostname*/

   Fill in *hostname* as appropriate. If you are running your server and the browser on the same host, then you can use localhost as the hostname.

   After loading this page, your browser will retrieve the embedded images from the server. There are a number of embedded images and your browser should open multiple connections to simultaneously retrieve image files.
2. The following three links should all load this page: **index.html**, **/index.html**, and **/**.
3. The following link should load another HTML page: **page.html**.
4. The following link should load a JPEG file (which happens to be NEB 261 in its infancy): **neb005.jpg**.
5. The following link should load a GIF file: **thm001.gif**.
6. The following link should load a PDF file: **pdf_sample.pdf**.
7. The following link should load a text file: **text_sample.txt**.
8. If you implemented a directory listing option, the following link should return a listing of the images directory: **images directory listing**. If you provide a hyperlinked directory, clicking on each file name should load the image.
9. This is a **bad link**.
10. If you implemented optional support for the POST method, use the following file for testing: **post_test.html**.
11. At this point, the only suggestion for testing the optional support for the HEAD method is to use the Castalia Socket Tester or iego's Connecting Sockets to generate a HEAD request.

Testing functionality:

- All the three links are loading the same index.html page: **index.html**, **/index.html**, and **/**.
- The following screenshot is taken after clicking on page.html: **page.html**.



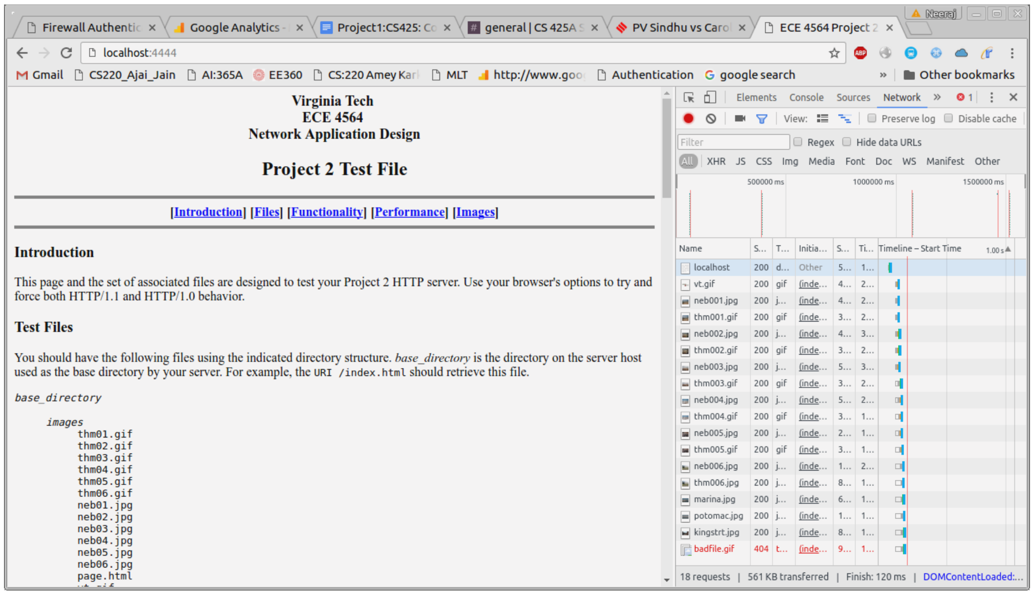
localhost:4444/images/page.html

This should return to the test page.

Name	S...	T...	Initia...	S...	Ti...
page.html	200	d...	Other	3...	7...

1 requests | 327 B transferred | Finish: 7 ms | DOMContentLoaded: 54 m...

- Below screenshot came after clicking on "test page" in above link



localhost:4444

Virginia Tech
ECE 4564
Network Application Design

Project 2 Test File

[Introduction] [Files] [Functionality] [Performance] [Images]

Introduction

This page and the set of associated files are designed to test your Project 2 HTTP server. Use your browser's options to try and force both HTTP/1.1 and HTTP/1.0 behavior.

Test Files

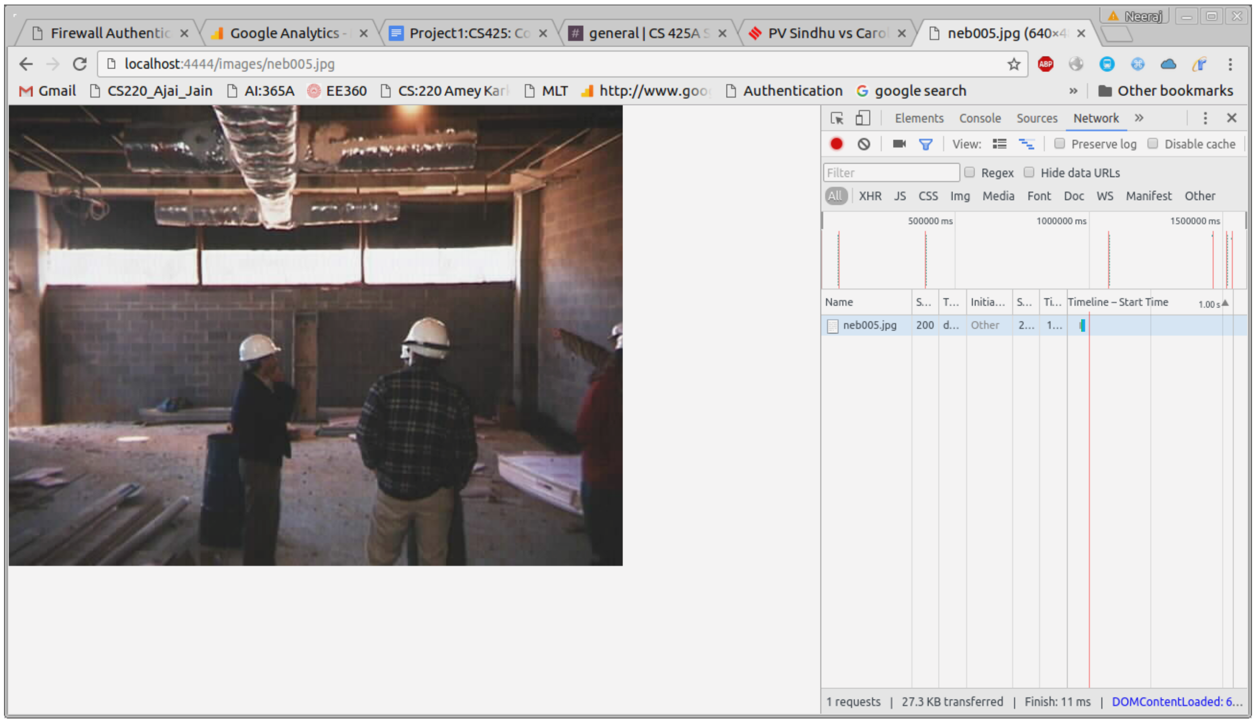
You should have the following files using the indicated directory structure. *base_directory* is the directory on the server host used as the base directory by your server. For example, the `URI /index.html` should retrieve this file.

```
base_directory

     images
          thm01.gif
          thm02.gif
          thm03.gif
          thm04.gif
          thm05.gif
          thm06.gif
          neb01.jpg
          neb02.jpg
          neb03.jpg
          neb04.jpg
          neb05.jpg
          neb06.jpg
          page.html
```

Name	S...	T...	Initia...	S...	Ti...
localhost	200	d...	Other	5...	1...
vt.gif	200	gif	(inde...	4...	2...
neb001.jpg	200	j...	(inde...	4...	2...
thm001.gif	200	gif	(inde...	3...	2...
neb002.jpg	200	j...	(inde...	4...	3...
thm002.gif	200	gif	(inde...	3...	2...
neb003.jpg	200	j...	(inde...	5...	3...
thm003.gif	200	gif	(inde...	3...	2...
neb004.jpg	200	j...	(inde...	5...	2...
thm004.gif	200	gif	(inde...	3...	1...
neb005.jpg	200	j...	(inde...	2...	1...
thm005.gif	200	gif	(inde...	3...	1...
neb006.jpg	200	j...	(inde...	1...	2...
thm006.jpg	200	j...	(inde...	8...	1...
marina.jpg	200	j...	(inde...	6...	1...
potomac.jpg	200	j...	(inde...	1...	1...
kingstrt.jpg	200	j...	(inde...	8...	1...
badfile.gif	404	t...	(inde...	9...	1...

18 requests | 561 KB transferred | Finish: 120 ms | DOMContentLoaded:...

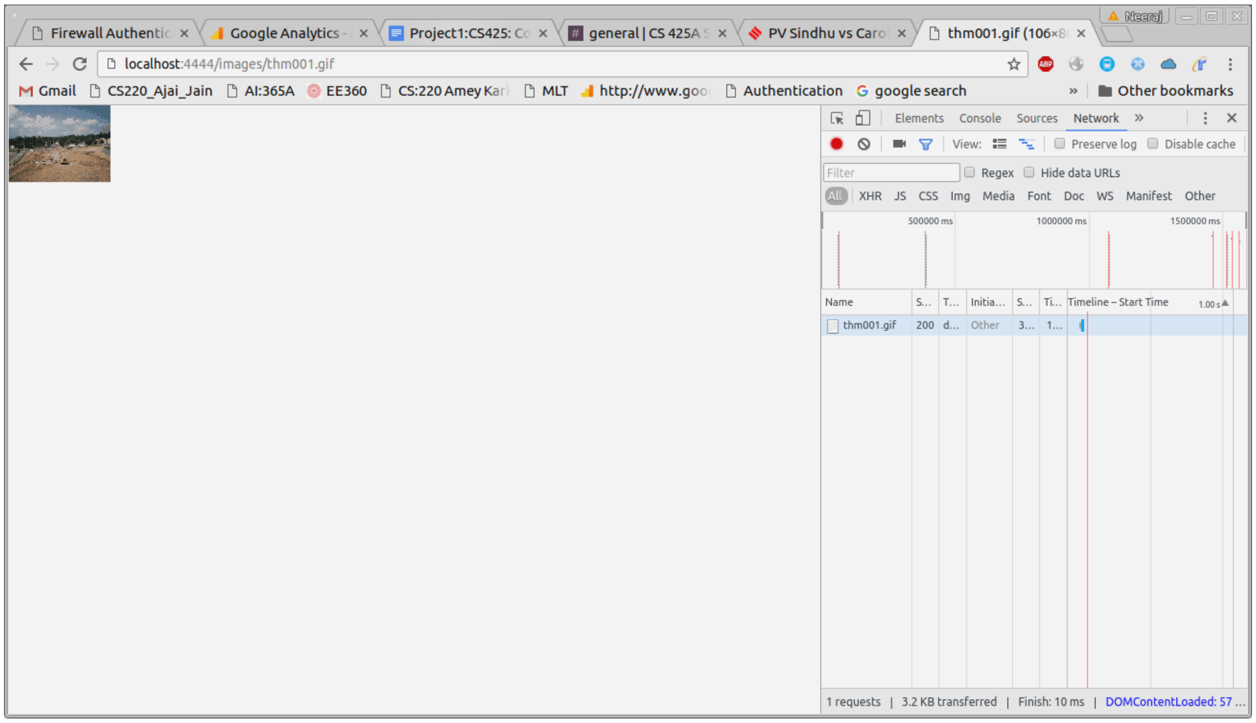
- Below screenshot came after “The following link should load a JPEG file (which happens to be NEB 261 in its infancy): **neb005.jpg**.”



Name	S...	T...	Initia...	S...	Ti...	Timeline – Start Time
neb005.jpg	200	d...	Other	2...	1...	

1 requests | 27.3 KB transferred | Finish: 11 ms | DOMContentLoaded: 6...

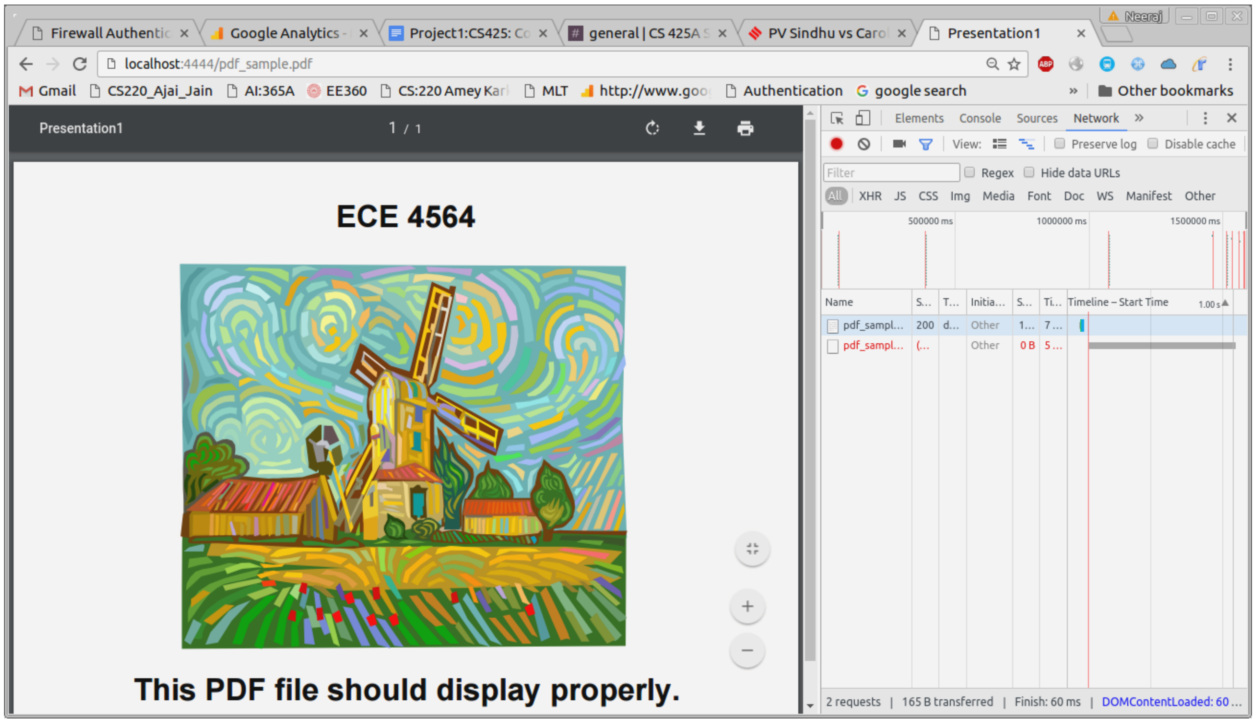
- Below screenshot came after “Below screenshot came after “The following link should load a GIF file: **thm001.gif**..”



Name	S...	T...	Initia...	S...	Ti...	Timeline – Start Time
thm001.gif	200	d...	Other	3...	1...	

1 requests | 3.2 KB transferred | Finish: 10 ms | DOMContentLoaded: 57 ...

- Below screen came after clicking on “The following link should load a PDF file: pdf_sample.pdf.”


localhost:4444/pdf_sample.pdf
Presentation1
1 / 1
Elements Console Sources Network
Preserve log Disable cache

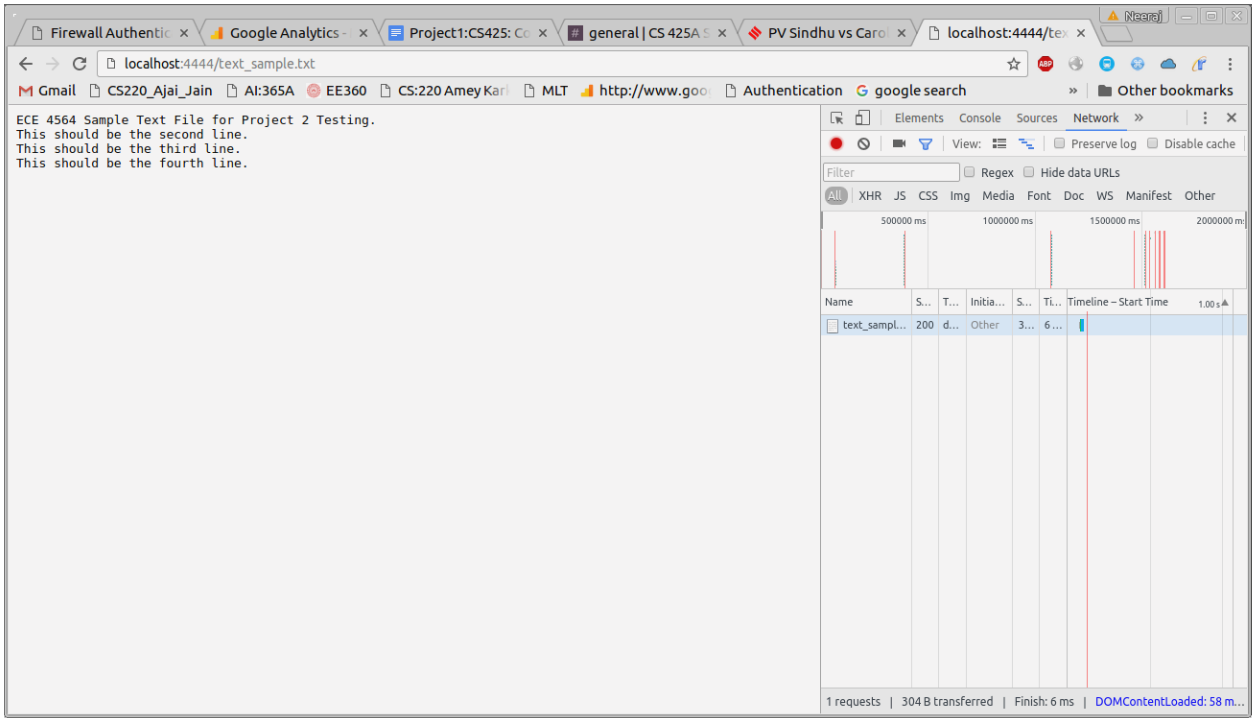
ECE 4564

This PDF file should display properly.

Name	S...	T...	Initia...	S...	Ti...
pdf_sampl...	200	d...	Other	1...	7 ...
pdf_sampl...	(...		Other	0 B	5 ...

2 requests | 165 B transferred | Finish: 60 ms | DOMContentLoaded: 60 ...

- Below screen came after clicking on “The following link should load a text file: text_sample.txt.”

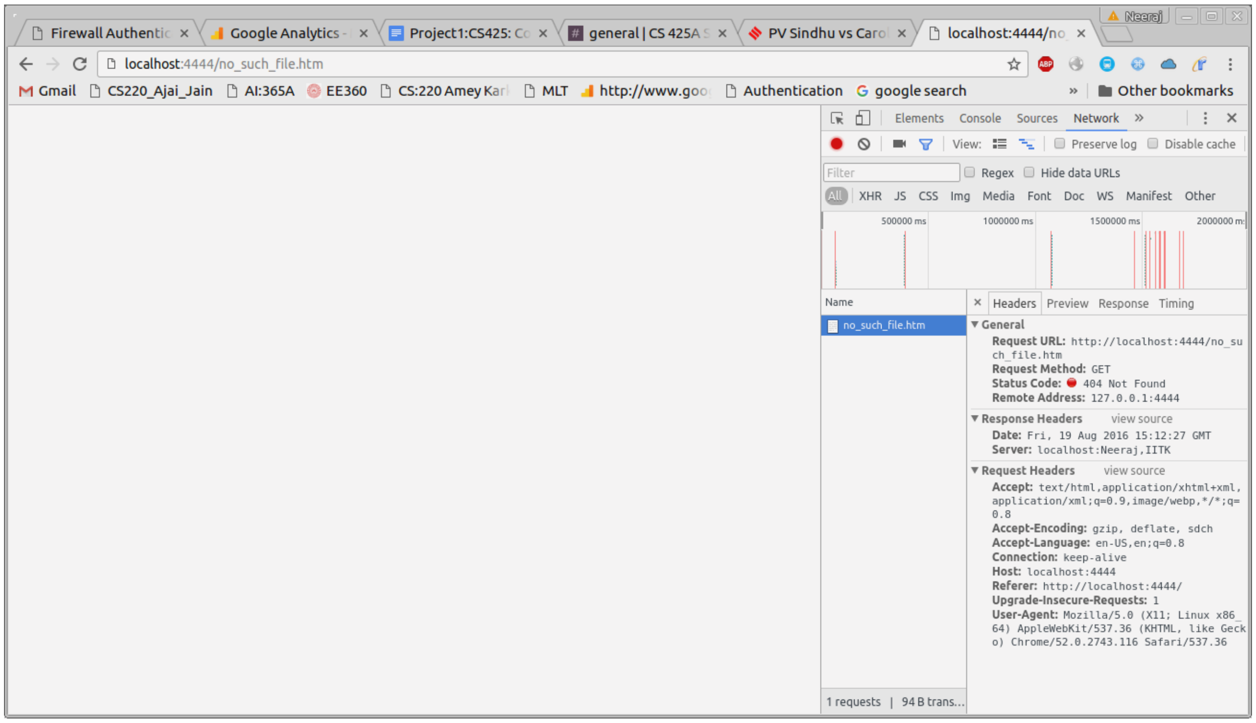

localhost:4444/text_sample.txt
Elements Console Sources Network
Preserve log Disable cache

```
ECE 4564 Sample Text File for Project 2 Testing.
This should be the second line.
This should be the third line.
This should be the fourth line.
```

Name	S...	T...	Initia...	S...	Ti...
text_sampl...	200	d...	Other	3...	6 ...

1 requests | 304 B transferred | Finish: 6 ms | DOMContentLoaded: 58 m...

- Below screen came after clicking on “This is a bad link.”
- We can go back after clicking on backspace arrow

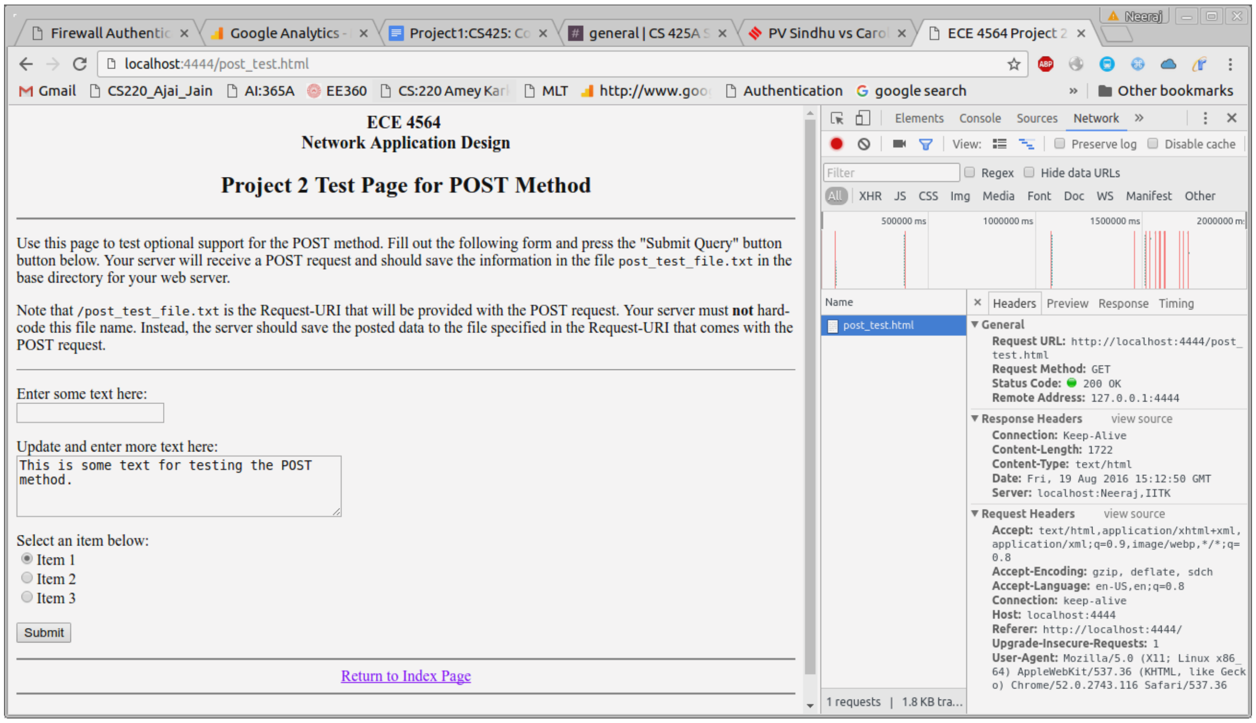


localhost:4444/no_such_file.htm

no_such_file.htm

```
▼ General
    Request URL: http://localhost:4444/no_such_file.htm
    Request Method: GET
    Status Code: ● 404 Not Found
    Remote Address: 127.0.0.1:4444
▼ Response Headers
    Date: Fri, 19 Aug 2016 15:12:27 GMT
    Server: localhost:Neeraj,IITK
▼ Request Headers
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
    Accept-Encoding: gzip, deflate, sdch
    Accept-Language: en-US,en;q=0.8
    Connection: keep-alive
    Host: localhost:4444
    Referer: http://localhost:4444/
    Upgrade-Insecure-Requests: 1
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36
```

- Below screen came after clicking on “If you implemented optional support for the POST method, use the following file for testing: post_test.html.”



localhost:4444/post_test.html

ECE 4564
Network Application Design

Project 2 Test Page for POST Method

Use this page to test optional support for the POST method. Fill out the following form and press the "Submit Query" button button below. Your server will receive a POST request and should save the information in the file `post_test_file.txt` in the base directory for your web server.

Note that `/post_test_file.txt` is the Request-URI that will be provided with the POST request. Your server must **not** hard-code this file name. Instead, the server should save the posted data to the file specified in the Request-URI that comes with the POST request.

Enter some text here:

Update and enter more text here:
This is some text for testing the POST method.

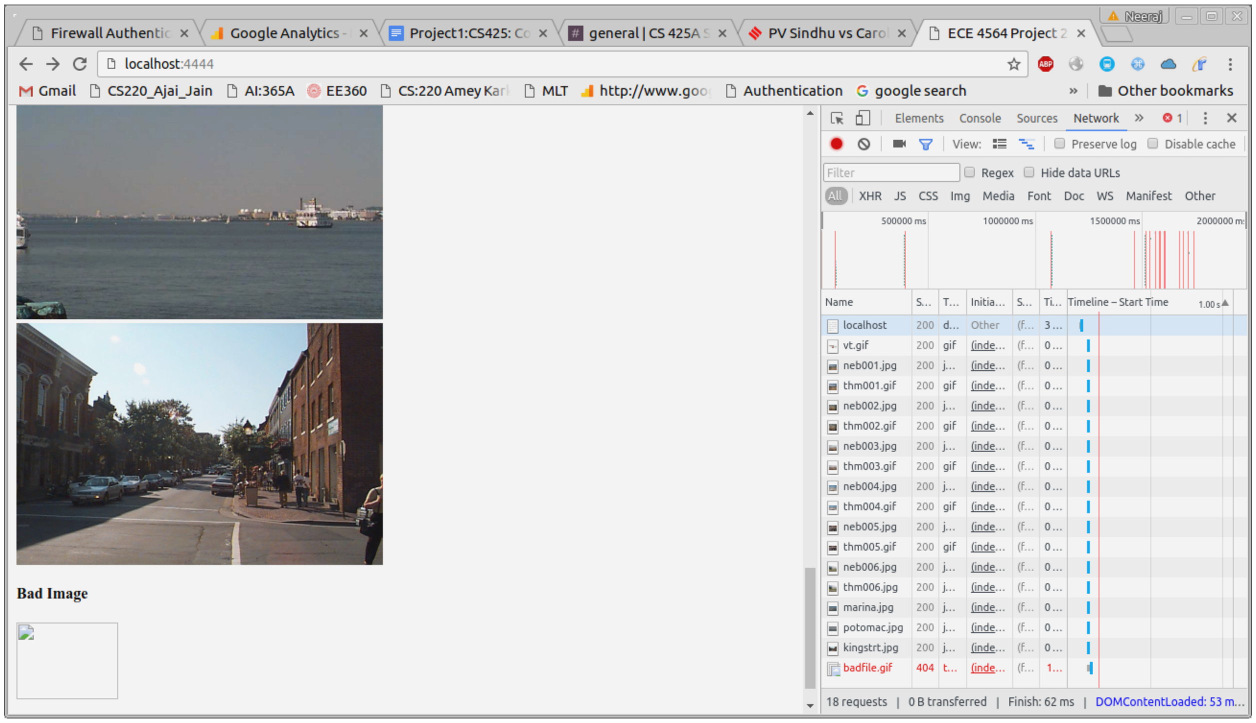
Select an item below:
[✓ Item 1]
Item 2
Item 3

Submit

Return to Index Page

post_test.html

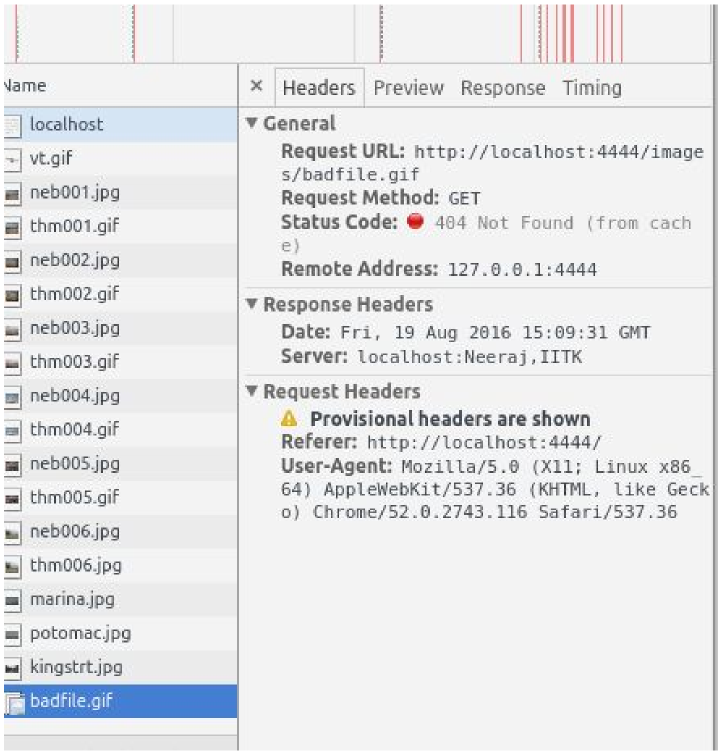
```
▼ General
    Request URL: http://localhost:4444/post_test.html
    Request Method: GET
    Status Code: ● 200 OK
    Remote Address: 127.0.0.1:4444
▼ Response Headers
    Connection: Keep-Alive
    Content-Length: 1722
    Content-Type: text/html
    Date: Fri, 19 Aug 2016 15:12:50 GMT
    Server: localhost:Neeraj,IITK
▼ Request Headers
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
    Accept-Encoding: gzip, deflate, sdch
    Accept-Language: en-US,en;q=0.8
    Connection: keep-alive
    Host: localhost:4444
    Referer: http://localhost:4444/
    Upgrade-Insecure-Requests: 1
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36
```



Name	S...	T...	Initia...	S...	Ti...
localhost	200	d...	Other	(f...	3 ...
vt.gif	200	gif	(inde...	(f...	0 ...
neb001.jpg	200	j...	(inde...	(f...	0 ...
thm001.gif	200	gif	(inde...	(f...	0 ...
neb002.jpg	200	j...	(inde...	(f...	0 ...
thm002.gif	200	gif	(inde...	(f...	0 ...
neb003.jpg	200	j...	(inde...	(f...	0 ...
thm003.gif	200	gif	(inde...	(f...	0 ...
neb004.jpg	200	j...	(inde...	(f...	0 ...
thm004.gif	200	gif	(inde...	(f...	0 ...
neb005.jpg	200	j...	(inde...	(f...	0 ...
thm005.gif	200	gif	(inde...	(f...	0 ...
neb006.jpg	200	j...	(inde...	(f...	0 ...
thm006.jpg	200	j...	(inde...	(f...	0 ...
marina.jpg	200	j...	(inde...	(f...	0 ...
potomac.jpg	200	j...	(inde...	(f...	0 ...
kingstrt.jpg	200	j...	(inde...	(f...	0 ...
badfile.gif	404	t...	(inde...	(f...	1...

18 requests | 0 B transferred | Finish: 62 ms | DOMContentLoaded: 53 m...

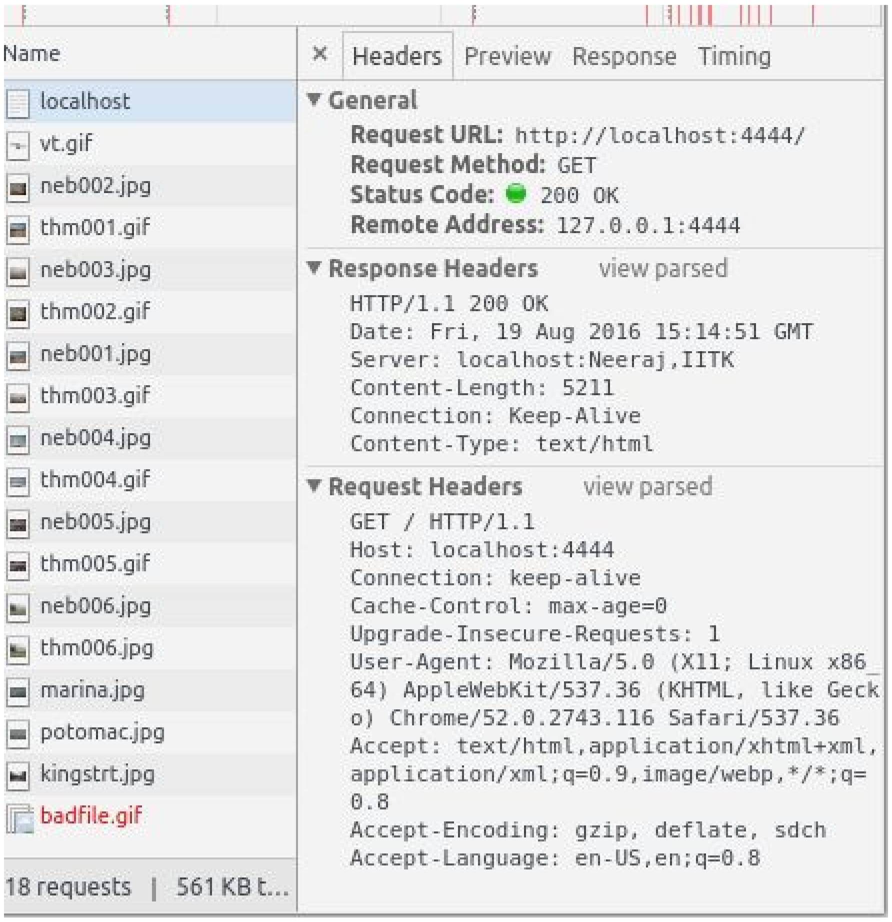
- Header of bad file clearing showing the 404 status



Name: localhost, vt.gif, neb001.jpg, thm001.gif, neb002.jpg, thm002.gif, neb003.jpg, thm003.gif, neb004.jpg, thm004.gif, neb005.jpg, thm005.gif, neb006.jpg, thm006.jpg, marina.jpg, potomac.jpg, kingstrt.jpg, badfile.gif

```
▼ General
    Request URL: http://localhost:4444/images/badfile.gif
    Request Method: GET
    Status Code: ● 404 Not Found (from cache)
    Remote Address: 127.0.0.1:4444
▼ Response Headers
    Date: Fri, 19 Aug 2016 15:09:31 GMT
    Server: localhost:Neeraj,IITK
▼ Request Headers
    ⚠ Provisional headers are shown
    Referer: http://localhost:4444/
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36
```

- Header showing OK status when opening the client for first time



Name

localhost
vt.gif
neb002.jpg
thm001.gif
neb003.jpg
thm002.gif
neb001.jpg
thm003.gif
neb004.jpg
thm004.gif
neb005.jpg
thm005.gif
neb006.jpg
thm006.jpg
marina.jpg
potomac.jpg
kingstrt.jpg
badfile.gif

18 requests | 561 KB t...

▼ General

```
Request URL: http://localhost:4444/
Request Method: GET
Status Code: ● 200 OK
Remote Address: 127.0.0.1:4444
```

▼ Response Headers   view parsed

```
HTTP/1.1 200 OK
Date: Fri, 19 Aug 2016 15:14:51 GMT
Server: localhost:Neeraj,IITK
Content-Length: 5211
Connection: Keep-Alive
Content-Type: text/html
```

▼ Request Headers   view parsed

```
GET / HTTP/1.1
Host: localhost:4444
Connection: keep-alive
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_
64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/52.0.2743.116 Safari/537.36
Accept: text/html,application/xhtml+xml,
application/xml;q=0.9,image/webp,*/*;q=
0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
```