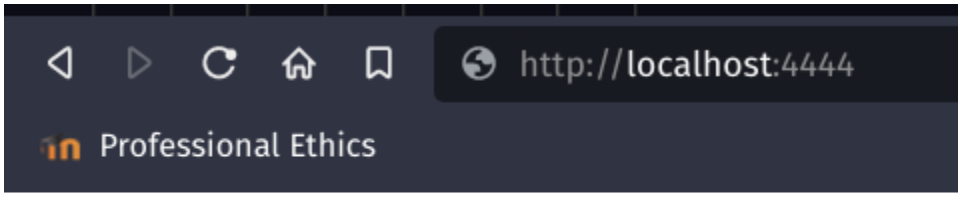
Q3



Hello World

```c
#include<sys/socket.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <string.h>
#include<stdio.h>

#define PORT 4444

int main(int argc, char const *argv[]){
   int server_fd, new_sock;
   struct sockaddr_in address;
   // char *hello = "Hello World";
   char *hello = "'HTTP/1.1 200 OK\r\nContent-Type:
text/plain\r\nConnection: close\r\n\r\nHello World";
   // Had to change the format of the response, because just "Hello World"
was resulting in an invalid response

   // write your code here

   // Socket file descriptor for server
   // We use AF_INET for IPv6
   // SOCK_STREAM = relaible and connection oriented
   server_fd = socket(AF_INET, SOCK_STREAM, 0);

   // Error Checking
   if(server_fd<0){
       printf("Error in opening socket\n");
       return 0;
   }
```

```
   // Need to specifiy address and port number so that it can bind to it
   address.sin_family = AF_INET;
   address.sin_addr.s_addr = INADDR_ANY;
   address.sin_port = htons(PORT);


   // need to bind the server_fd to an address and a port as mentioned in
struct (sockaddr_in) (4444 in this case)
   if(bind(server_fd, (struct sockaddr *) &address, sizeof(address)) ==
-1){
       printf("Cannot bind\n");
       return 0;
   }


   // this is needed so that the socket waits for client to approach. The
number 7 is the maximum queue size.
   listen(server_fd, 7);

   int addrlen = sizeof(address); // had to initialize this variable in
order to pass it as a pointer in a later function
   // it accepts the first connection request from the listening socket
queue and creates a new connected socket and returns a file descriptor
   new_sock = accept(server_fd, (struct sockaddr *) &address,
(socklen_t*)&addrlen);

   // Error Checking
   if(new_sock == -1){
       printf("Cannot accept\n");
       return 0;
   }


   send(new_sock, hello, strlen(hello), 0);
   printf("Hello message sent to browser\n");
   return 0;
}
```