

Student Name: Gökay Gülsoy

Student Number: 270201072

-CENG421 Project-

In my Project I have implemented an small web server called tiny that supports GET requests made from any web browser of your choice via http protocol (beware https does not work!! because I haven't implemented openssl functionality) following are the commands to build and run the web server:

Folder Structure of the Project

- bin
 - tiny
- obj
 - helpers.o
 - tiny.o
- public
 - form.html
 - index.html
 -
 - page2.html
 - smile.png
 - test.txt
- src
 - helpers.h
 - helpers.c
 - tiny.c
- Makefile

Building the Server

Run the *make* command from the top-level directory (PROJECT directory) as follows:

make

Running the *make* command will create object files under the *obj* directory. As an output of running *make* command you should see output on your terminal similar to following if program was built successfully:

mkdir -p obj

gcc -Wall -Werr -c src/helpers.c -o obj/helpers.o

gcc -Wall -Werr -c src/tiny.c -o obj/tiny.o

gcc -Wall -Werr obj/helpers.o obj/tiny.o -o bin/tiny

Final step is running our tiny web server via following command and command must be in the form **<path_to_tiny_executable> <port_number>**:

./bin/tiny 1095

If all the above steps are successfully done, you can go to your favorite web browser and type the URL in the following format to access our local web server:

http://localhost:<port_number>/public/<any_file_to_be_served>

When making http request to our web server tiny, you should provide a path to any file that you want to get located under the public directory, available files are as follows:

- Public
 - ◆ Form.html
 - ◆ index.html
 - ◆ page2.html
 - ◆ test.txt

Cleaning up the Project Directory

Run the following command from the top-level directory (which is PROJECT) to clean up the project directory, so that you can later rebuild the project:

make clean

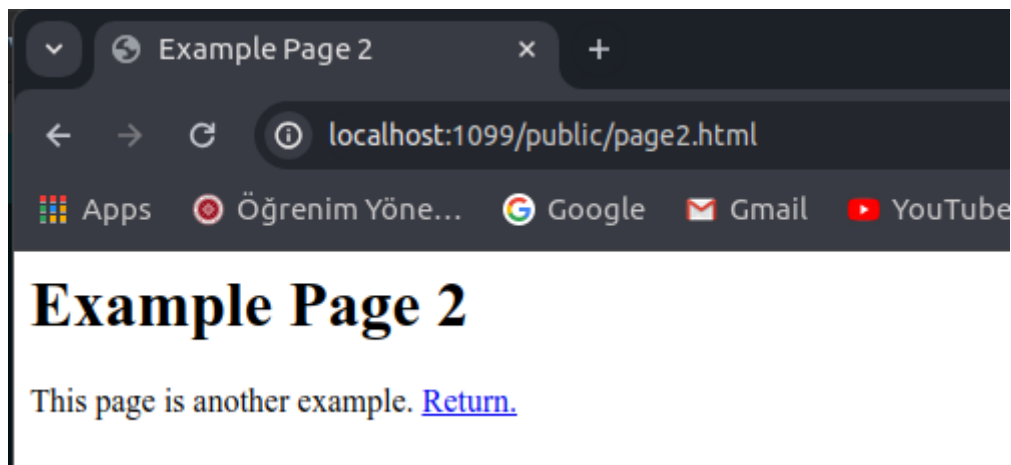
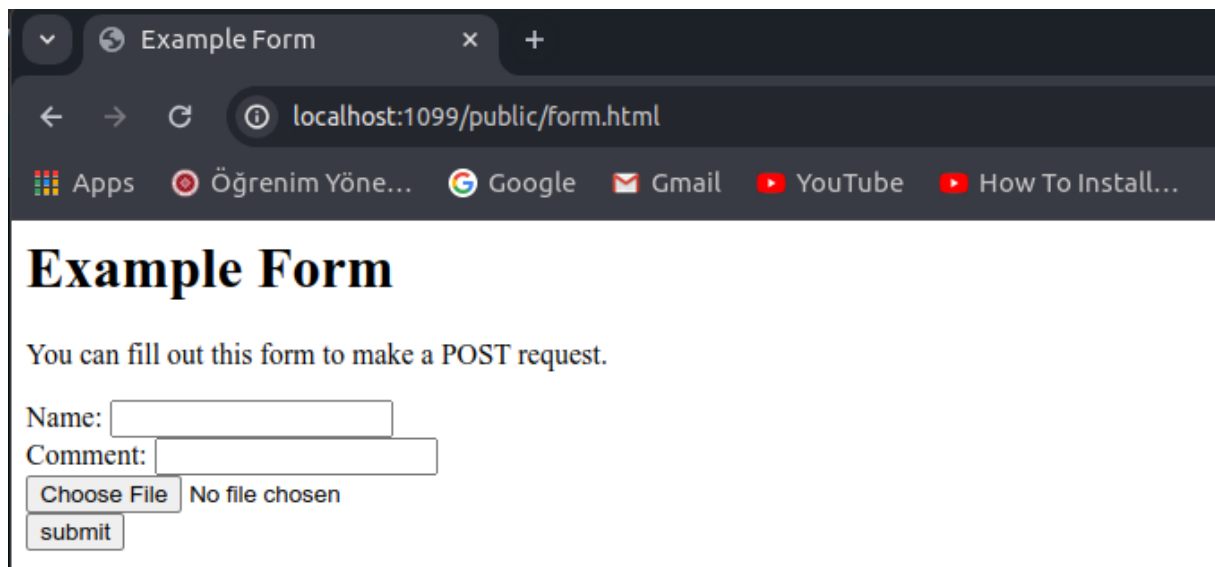
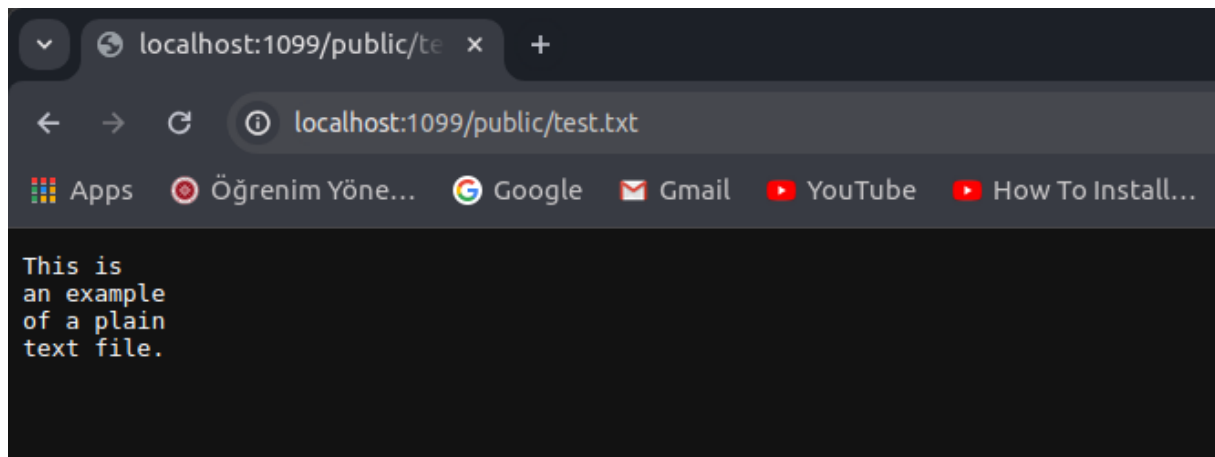
You can find steps to be followed in image format as follows:

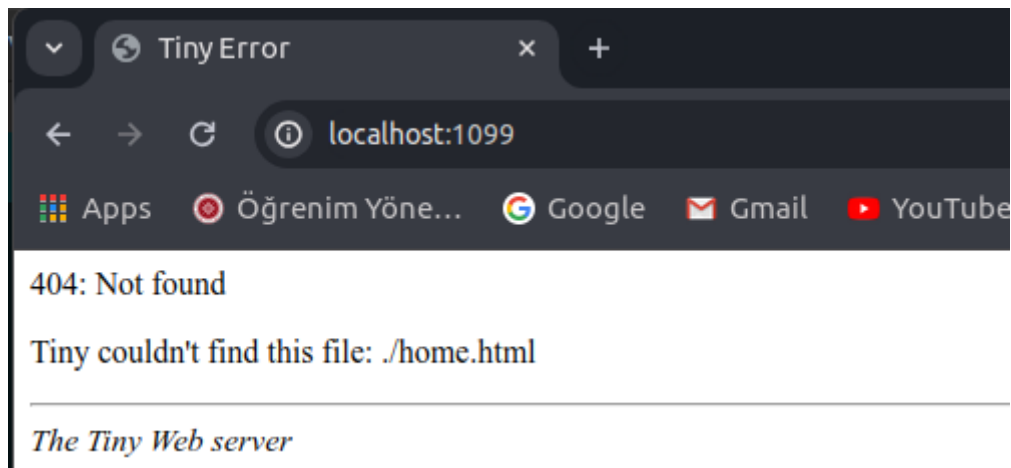


The image contains two screenshots of a terminal window. The top screenshot shows the compilation process: a terminal prompt with a status bar (bash, MEM: 23.36%, 4/15GB, 24ms) and a directory path (Project). The user enters 'make', which runs 'mkdir -p obj', 'gcc -Wall -Werror -c src/helpers.c -o obj/helpers.o', 'gcc -Wall -Werror -c src/tiny.c -o obj/tiny.o', and 'gcc -Wall -Werror obj/helpers.o obj/tiny.o -o bin/tiny'. The bottom screenshot shows the execution of the program: the same terminal prompt and directory path, with the user entering './bin/tiny 1099'.

```
bash MEM: 23.36% | 4/15GB | 24ms
16:24 | Project
└─ make
mkdir -p obj
gcc -Wall -Werror -c src/helpers.c -o obj/helpers.o
gcc -Wall -Werror -c src/tiny.c -o obj/tiny.o
gcc -Wall -Werror obj/helpers.o obj/tiny.o -o bin/tiny

bash MEM: 23.49% | 4/15GB | 133ms
16:25 | Project
└─ ./bin/tiny 1099
```





We can also see log messages of our web server tiny on terminal when we make a GET request to valid file, for example for test.txt file log is as follows:

```
Accepted connection from (localhost, 35300)
GET /public/test.txt HTTP/1.1
Host: localhost:1099
Connection: keep-alive
sec-ch-ua: "Not_A Brand";v="8", "Chromium";v="120", "Google Chrome";v="120"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36
Sec-Purpose: prefetch;prerender
Purpose: prefetch
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en;q=0.9,tr-TR;q=0.8,tr;q=0.7,de-DE;q=0.6,de;q=0.5,en-US;q=0.4
```