

Department of Electrical and Computer Engineering

ENCS3320-Computer Networks

**Project#1 due 31/8/2023**

**Rules & Guidelines:**

1. This is a group project, so you are allowed to work in groups of max 3 students.
2. Do not use libraries to implement the project. Use socket programming.
3. **Important: each screenshot should include the date and time of your computer.**

**You have to submit:**

1. A report in pdf format (only pdf format) on moodle (itc.birzeit.edu) that contains screenshots with detailed explanation, codes, runs, etc.
2. The code with comments (include the code in the pdf file and as text file .py or .java or .c as well).
3. You are allowed to send compressed file (e.g., .zip). But you have to send the report as pdf file separately.

**Requirements:**

**Part1:**

1. In your own words, what are **ping**, **tracert**, **nslookup**, and **telnet** (**write** one sentence for each one).
2. Make sure that your computer is connected to the internet and then run the following commands:
3. **ping** [www.amazon.de](http://www.amazon.de)
4. **tracert** [www.amaon.de](http://www.amaon.de)
5. **nslookup** [www.amazon.de](http://www.amazon.de)

**Provide** a **screenshot** of the runs and brief explanation of the output.

**Part2:**

Using socket programming, implement a simple but a complete web server in go, python, java or C that is listening on port 12345. The user types in the browser something like [http://localhost:12345/](http://localhost:9977/)ar or [http://localhost:12345/en](http://localhost:9977/en)

**The program should check**

1. If the request is **/ or /index.html or /main\_en.html or /en (for example localhost:12345/ or localhost:12345/en)** then the server should send main\_en.html file with

Content-Type: text/html; charset=UTF-8

1. The main\_en.html file should contain HTML webpage, which contains the following:
2. “ENCS3320-My Tiny Webserver” in the title
3. “Welcome to our course **Computer Networks, This is a tiny webserver” (**part of the phrase is in **Blue)**
4. Group membersnames and IDs
5. Some information about the group members. For instance, projects you have done during different course (programming, electrical, math, etc), skills, hobbies, etc.
6. Use CSS to make the page looks nice
7. Divide the page in different boxes and put student’s information in the different boxes
8. Include CSS as a separate file
9. The page should contain at least an image with extention.jpg and an image with extension .png
10. A link to a local html file (an html file)
11. A link to <https://www.w3schools.com/python/python_tuples.asp>
12. If the request is /ar then the server response with main\_ar.html which is an Arabic version of main\_en.html.
13. Include a text file (or you can use csv file) that contains names and prices of at least 10 laptops.
14. If the request is **/SortByName** then the output on the browser should be the names in capital letters (the entire name in capital) and prices of the laptops sorted by the name. The server should send text page with Content-Type: text/plain; charset=UTF-8. If you wish, you can use text/html to display the output in a more convenient way.
15. If the request is **/SortByPrice** then the output on the browser should be name and price of the laptops sorted by its price. It provides all the sum of all prices of all laptops. The server should send text page with Content-Type: text/plain; charset=UTF-8. If you wish, you can use text/html to display the output in a more convenient way.
16. If the request is a .**html** file, then the server should send the requested html file with Content-Type: text/html; charset=UTF-8. You can use any html file.
17. If the request is a **.css** file, then the server should send the requested css file with Content-Type: text/css; charset=UTF-8. You can use any CSS file.
18. If the request is a .**png** then the server should send the png image with Content-Type: image/png. You can use any image.
19. If the request is a **.jpg** then the server should send the jpg image with Content-Type: image/jpeg. You can use any image.
20. Use the status code 307 Temporary Redirect to redirect the following
    1. If the request is /azn then redirect to amazon website
    2. If the request is /so then redirect to stackoverflow.com website
    3. If the request is /bzu then redirect to birzeit website
21. If the request is wrong or the file doesn’t exist, the server should return a simple HTML webpage that contains (Content-Type: text/html; charset=UTF-8 )
22. “HTTP/1.1 404 Not Found” in the response status
23. “Error 404” in the title
24. “The file is not found” in the body in **red**
25. Yournames and IDs in **Bold**
26. The IP and port number of the client
27. The program should print the HTTP requests on the terminal window (command line window).

**Provide** screenshots of the browser with brief descriptions to show that your project works as expected. **(/main\_en.html /imagename.png, /azn, etc.)** . Test the project from a browser on the **same computer.**

**Provide** also a screenshot of the HTTP request printed on the command line.

**Hint**: Have a look on HTTP response in **Listing 1** and the HTML code in **Listing 2**. You may use the minimal header and HTML code. Have a look also on **rfc2616** (https://tools.ietf.org/html/rfc2616)

*HTTP/1.1 200 OK*

*Connection: close*

*Date: Fri, 03 Mar 2017 06:19:37 GMT*

*Server: Apache/2.4.6 (CentOS) OpenSSL/1.0.1e-fips PHP/5.4.16*

*Last-Modified: Fri, 03 Mar 2017 05:28:07 GMT*

*Content-Length: 6821*

*Content-Type: text/html; charset=UTF-8*

*data data data data data ...*

Listing 1: HTTP Response

*<!DOCTYPE html>*

*<html>*

*<head><title >XYZ Company INC.</ title ></head>*

*<body><h1>Welcome <b>XYZ</b> Company</h1>*

*<img src="http:www. xyz.com/ images / logo.gif " ALT="XYZ Logo"><br>*

*We are so happy that you have chosen to visit our website.*

*</body>*

*</html>*

Listing 2: Simple HTML Code