IIT Goa CS 378 (Computer Networks LAB)
Instructor: Neha Karanjkar

LAB 3: The Web and HTTP

Instructions:

- For this lab you can work individually OR in teams of 2 persons. Choose your teammate. If working in a team, both team members need to understand the approach and the solution and should work on each problem together and in sequence. Do not divide up the questions among team members.
- You **are allowed** to use Google search and the www for information.
- You **can** help other teams. However, remember that there is a time limit. Complete your own work first before helping others. Help others by sharing the method and the knowledge to arrive at a solution.
- **Do not share your code with another team.** There will be a grade penalty for directly sharing code with another team.
- There is a strict time limit of 2 hours. You will be graded for the exercises you can complete within this time budget.
- When you **complete all the exercises** (or as many as you can within the time budget) please notify the instructor for a demo of the solutions.

ROLL NUMBERS:

______________________________(Team member 1)

______________________________(Team member 2, if any)

1. Choose a name for your team. Create a simple, personalized website dedicated to this team using only raw HTML and a text editor.
    - The website should consist of atleast 2 webpages, where one of the pages should be named “index.html”
    - Each webpage should contain a hyper-link to the other.
    - The webpages should contain:
        - alteast one link to some external website
        - atleast one image
        - some formatted text such as headings and bold text.

    (The website should **not** include any NSFW content)

    Store the objects (.html files, images etc) in a single folder on the local machine and test your webpages by opening the html files in a browser.

2. Create a web-server for your website using Python (preferably Python3). The webserver process should accept a TCP connection from a Client process on port 9000 and generate responses to GET requests for objects contained in your website's folder.

   Hint: You can directly make use of libraries such as http.server instead of parsing each request read from the TCP socket. (See https://www.afternerd.com/blog/python-http-server/ for a short tutorial).

   a) Test your webserver by entering the following url in a browser: "http://localhost:9000" on the same machine.
   b) Check if your website can be accessed from another machine in the lab. Use the private IP address of the webserver (10.xx.xx.xx) instead of 'localhost'.
   c) View the HTTP Request and Response traffic generated by the browser using the browser's developer/Network tool.

3. [BONUS QUESTION]
   Modify your webserver so that it is able to serve multiple client processes simultaneously. Check this by accessing your website from multiple browsers at the same time.

4. [BONUS QUESTION]
   Modify your webserver to keep track of past visitors using Cookies.

   Everytime a new user connects to the webserver, the server should set a cookie called "FOO123" on the client (your browser). Check that the client indeed received this cookie and has stored it. Thus for every subsequent visit to the website, the Client should automatically send a "Cookie: FOO123" header along with the request.

   If and only if the cookie is set, the displayed webpage should contain the text "Hello user, we've met before".