

1

CSC 361: Computer Communications and 2 Networks (Fall 2025)

3

Programming Assignment 1: Web Tester

4

5 Deadline: 23:59, Sep. 30, 2025

6

1 Goal

7 The project is to build a tool that collects information regarding a web server. The purpose of this
8 project is twofold:

- 9
- to provide students with hands-on experience with socket programming **in Python**,
 - 10 • to help students understand the application-layer protocols HTTP/HTTPPs. Note that HTTPPs
11 is not a standalone protocol. Instead, it is HTTP over Transport Layer Security (TLS). In
12 this assignment, your main focus is on HTTP, not TLS.

13

2 Background

14

2.1 HTTP

15 HTTP stands for Hyper Text Transfer Protocol and is used for communication among web servers.

16 The web client initiates a conversation by opening a connection to a web server. Once the
17 connection is set up, the client sends an HTTP request. The server sends an HTTP response back
18 to the client and closes the connection. An HTTP request consists of two parts: a header and a
19 body. The header specifies whether a body follows a header or not.

20 Using *single-line header of HTTP request* as an example, the first line of any request header
21 should be:

- 22
- the method field: The method field can take on several different values, including GET, POST,
23 HEAD, etc.
 - 24 • the URL field: It is the field to identify a network resource, e.g., “<http://www.csc.uvic.ca/index.html>”.
 - 25 • the HTTP version field: This field is “HTTP/1.1”.

26 The response from a server also has two parts: a header and a body. The first line of a header
27 should be:

- 28
- the HTTP version field,
 - 29 • the status code field,

- 30 • the phrase field.

31 Two main status codes include 200 and 404. The status code 200 means that the request
32 succeeded and the information is returned in the response. The status code 404 means that the
33 requested document does not exist on this server. Two example response messages are: “*HTTP/1.1*
34 *404 Not Found\r\n\r\n*” and “*HTTP/1.1 200 OK\r\n\r\n data data data ...*” Another two status
35 codes 505: “*HTTP Version Not Supported*”, and 302: “*302 found*” for URL redirection are also
36 useful for this assignment.

37 **2.2 URI**

38 URI stands for Uniform Resource Identifier and is also known as the combination of Uniform
39 Resource Locators (URL) and Uniform Resource Names (URN). It is a formatted string which
40 identifies a network resource. It generally has the format: *protocol://host[:port]/filepath*. When a
41 port is not specified, the default HTTP port number is 80, and the default HTTPS port number is
42 443.

43 **2.3 Cookies**

44 An HTTP cookie is a small piece of data that a server sends to the user’s web browser. The browser
45 may store it and send it back with the next request to the same server. Typically, it’s used to tell
46 if two requests came from the same browser keeping a user logged-in, for example. It remembers
47 stateful information for the stateless HTTP protocol. Cookies have many applications in web, such
48 as tracking, authentication, and web analytics. Due to this reason, cookies also cause many concerns
49 on security and privacy breach.

50 The textbook includes simple introduction on cookies. More detailed information could be
51 found at: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies>. Python includes dedicated
52 modules to handle Cookies: <https://docs.python.org/3/library/http.cookies.html>. Nevertheless,
53 you are no allowed to use this package because it defeats the purpose of a network-course
54 assignment.

55 **3 Project Description**

56 You are required to build a web client tool, called *WebTester*, in Python. **Note that for consistency, program in other language will not be accepted!**

58 Given the URL of a web server, your *WebTester* needs to find out the following information
59 regarding the web server:

- 60 • 1. whether or not the web server supports http2,
- 61 • 2. the cookie name, the expire time (if any), and the domain name (in any) of cookies that
62 the web server will use,
- 63 • 3. whether or not the requested web page is password-protected.

64 Your program first accepts URI from stdin and parses it. Then it connects to a server, sends an
65 HTTP request, and receives an HTTP response. You should also implement a routine that prints
66 out the response from the server, marking the header and the body. When you finish the client, you

67 can try to connect to any HTTP server. For instance, type “`https://www.uvic.ca/`” as the input
68 to the client program and see what response you get.

69 As an example output, after you run your code with

70 `% python WebTester.py www.uvic.ca`

71 Your *WebTester* may output the received response from the server (**optional**), e.g.,

```
72 ---Request begin---
73 GET http://www.uvic.ca/index.html HTTP/1.1
74 Host: www.uvic.ca
75 Connection: Keep-Alive
76
77 ---Request end---
78 HTTP request sent, awaiting response...
79
80 ---Response header ---
81 HTTP/1.1 200 OK
82 Date: Tue, 02 Jan 2018 22:42:27 GMT
83 Expires: Thu, 19 Nov 1981 08:52:00 GMT
84 Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
85 Pragma: no-cache
86 Set-Cookie: SESSID_UV_128004=VD3v0JhqL3YUbmaZSTJre1; path=/; domain=www.uvic.ca
87 Set-Cookie: uvic_bar=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; Max-Age=0; path=/; dom
88 Keep-Alive: timeout=5, max=100
89 Connection: close
90 Content-Type: text/html; charset=UTF-8
91 Set-Cookie: www_def=2548525198.20480.0000; path=/
92 Set-Cookie: TS01a564a5=0183e07534a2511a2dc274bee873845d67a2c07b7074587c948f80a42c427b1f7ea
93 Set-Cookie: TS01c8da3c=0183e075346a73ab4544c7b9ba9d7fa022c07af441fc6214c4960d6a9d0db2896a8c
94 Set-Cookie: TS014bf86f=0183e075347c174a4754aeb42d669781e0fafb1f43d3eb2783b1354159a9ad8d81f7
95
96 --- Response body ---
97 Body Body .... (the actual content)
98
```

99 Note that some lines in the above output were truncated. Your code may need to send multiple
100 requests in order to find out the required information. In particular, if you get an HTTP response
101 with code 302 or 301, you need to send further HTTP requests to the new URI provided by the
102 Location header.

103 Your code should output the final results (**mandatory**), for example:

```
104
105 website: www.uvic.ca
106 1. Supports http2: no
107 2. List of Cookies:
108 cookie name: SESSID_UV_128004, domain name: www.uvic.ca
109 cookie name: uvic_bar, expires time: Thu, 04-Jan-2018 00:00:01 GMT; domain name: .uvic.ca
```

```
110 cookie name: www_def,  
111 cookie name: TS01a564a5  
112 cookie name: TS01c8da3c, domain name: www.uvic.ca  
113 cookie name: TS014bf86f, domain name: .uvic.ca  
114 3. Password-protected: no
```

115 Note that the above output may be outdated and does not necessarily reflect the
116 ground truth of the current configuration of www.uvic.ca.

117 3.1 Other Notes

- 118 1. Regarding other printout: Anything not specified in Assignment 1 is optional. For example,
119 you can decide whether or not to print out the IP address, port number, and so on. When
120 TAs test your code, if your code works fine without any problem, you are fine even if you
121 do not print out anything not required in Assignment 1. Nevertheless, if your code does not
122 work, TAs will not spend time to figure out what is wrong and you get a zero mark on the
123 required function (Refer to the table in Section 5 of Assignment 1). In this case, if your code
124 includes some printout to show intermediate results, TAs will have an idea on how far you
125 have achieved and give you some partial mark based on their own judgement.
- 126 2. Regarding readme file. Readme file is important. Without it TAs will not know how to
127 compile your code and how to run your code. It would waste our time to deal with your
128 complaint if TAs cannot run your code and give you a zero.
- 129 3. For more information on HTTP, HTML, URI, etc., please refer to <http://www.w3.org>. It is
130 the home page of W3 Consortium and you will find many useful links to subjects related to
131 the World Wide Web.

132 4 Schedule

133 In order to help you finish this programming assignment successfully, the schedule of this assignment
134 has been synchronized with both the lectures and the tutorials. Before the final deadline, there are
135 three tutorial sessions arranged during the course of this assignment. A schedule is listed as follows:

Session	Tutorial	Milestones
Tutorial 1	P1 spec go-through, design hints, python	design and code skeleton
Tutorial 2	socket programming and testing	alpha code done
Tutorial 3	socket programming and last-minute help	beta code done and demo

136 5 Deliveries and Marking Scheme

137 For your final submission of each assignment you are required to submit your source code to
138 brightSpace in a single zip file (double-check your zip file to make sure all required files have
139 been included before submission!). You should include a readme file to tell TA how to compile and
140 run your code.

141 **Note:** For consistency and ease of test, you should test/run your code on the server linux.csc.uvic.ca
142 by running python3 and the python packages supported by the server linux.csc.uvic.ca. In other
143 words, TAs will test your code on linux.csc.uvic.ca and give marks based on the test results over
144 linux.csc.uvic.ca rather than the results from your local computer.

145 The marking scheme is as follows:

Components	Weight
Error handling	10
Correct output for “support of http2”	20
handling http redirect 302/301	20
List of Cookies	30
Correct output for password-protected	15
Readme.txt	5
Total Weight	100

146
147 **Important Note:** listing cookies is a very tricky business, and it is possible that you will not get
148 a unique, static answer due to the dynamic changes in some cookies created dynamically based on
149 users interactive input. Some online tool, such as <http://www.cookie-checker.com/>, can find cookies
150 that are triggered by javascript or php code. Nevertheless, finding those cookies is optional for this
151 Assignment.

152 6 Plagiarism

153 This assignment is to be done individually. You are encouraged to discuss the design of your solution
154 with your classmates, but each person must implement their own assignment.

155 The End
