Test Cases

Assumptions
- The web server runs and is accessible locally via localhost on port 9000.

Test Cases - Server Start

Test Case 1.1.1 - Start Server Successful
Run the web server to make it accessible to external agents, such as Firefox or Safari.

Input
- Run src/se.lnu.http/HTTPServerConsole.java with arguments:
    1. "9000"
    2. "bin/se/lnu/http/resources/inner/"

Output
- The server is running successfully.

Test Case 1.1.2 - Start Server Writes To Log Succesfully
A note is added regarding the success of the web server starting.

Input
- Perform test case: 1.1.1.

Output
- A message is written to the access log.

Test Case 1.1.3 - Start Server Failure: Socket in Use
A socket in use prevents the server to be started.

Input
- Perform test case: 1.1.1.

Output
- The server does not run.
- A message is written to the access log: "Socket XX was taken", where XX stands for the socket's number.

Test Case 1.1.4 - Start Server Failure: Restriction on Shared Resource
A restriction on the shared resource container prevents access.

Input
- Perform test case: 1.1.1.
- Try to access a restricted resource, such as "\var\www" (Linux only!)

Output
- The server does not run.
- A message is displayed: "No access to folder XX", where XX represents a directory.

Test Case 1.1.5 - Start Server
No log entry could be added.

Input
- Perform test case: 1.1.1.

Output
- No log entry is added.
- A message is displayed in the console: "Cannot write to server log file log.txt".

Test Cases - Server Stop

Test Case 2.1.1 - Stop Server
Close the server to prevent future HTTP requests from clients.

Precondition
- Server is running.

Input
- Terminate running process: src/se.lnu.http/HTTPServerConsole.java

Output
- The web server is closed.
- A message about the server being stopped is added to the access log.
- Requested resources will be unavailable, leading to error message(s) on the users' web browsers.

Test Cases - Request Shared Resource

Test Case 3.1.1 - Request Shared Resource
Access a specific resource.

Assumptions
- The protocol HTTP 1.1 is used to access the resources.

Input
- Perform test case: 1.1.1.
- Navigate to address: localhost:9000

Output
- An entry in the access log is added with information about the incoming HTML request.

Test Case 3.1.2 - Successful HTTP Request
Receive a successful HTTP response.

Input
- Perform Test Case 1.1.1 (Start Server)
- Using Insomnia REST, send a HTTP Get Request to address: localhost:9000

Ouput
- The HTTP Response status code should be: 200 OK.

Test Case 3.1.3 - Bad HTTP Request
Make a bad request.

Input
- Perform Test Case 1.1.1 (Start Server)
- Using Insomnia REST, send a HTTP Get Request to address: localhost:9000 with an incomplete header. For example, send the header "Accept-Language", but with no associated value.

Ouput
- The HTTP Response status code should be: 400 Bad request.

Test Case 3.1.4 - Forbidden HTTP Request
Make a forbidden request.

Input
- Perform Test Case 1.1.1 (Start Server)
- Using Insomnia REST, send a HTTP Get Request to address to request the resource: ../secret.html.

Ouput
- The HTTP Response status code should be: 403 Forbidden.

Test Case 3.1.5 - Not Found HTTP Request
Make a request for an unknown resource (misspelled resource, or non-existing).

Input
- Perform Test Case 1.1.1 (Start Server)
- Using Insomnia REST, send a HTTP Get Request to address: localhost:9000/nonexisting.

Ouput
- The HTTP Response status code should be: 404 Not Found.

Test Case 3.1.6 - Reject POST Requests
Making a post request to the home page: localhost:9000 should be rejected.

Input
- Perform Test Case 1.1.1 (Start Server)
- Using Insomnia REST, send a HTTP POST Request to address: localhost:9000

Ouput
- The HTTP Response status code should be: 405 Method POST not supported.

Test Case 3.1.7 - Successful loading of resource: "works2.png".
See the image upon entering the home page: localhost.

Input
- Perform Test Case 1.1.1 (Start Server)
- Using a web browser, enter the web address: localhost:9000

Ouput
- The text "It works" is displayed, along with the image file: works2.png.

Test Cases - Ports

Test Case 5.1.1 - Busy Port

The requested port to be used is already in use, preventing the server from starting.

Input

- Occupy the same port that will be used by the web server, by for example hosting yet another application on localhost, or by first running the server, and then the code coverage tool so that they compete for the same port.
- Perform Test Case 1.1.1 (Start Server)

Ouput

- An error message will be displayed in the console, stating that the port is already in use.

Test Case 5.1.2 - Invalid Port

The server does not start due to a faulty port value - port number not inside range of 0 - 65535.

Input

- Change the application starting argument "9000", to " 65536".
- Perform Test Case 1.1.1 (Start Server)

Ouput

- An error message will be displayed regarding the chosen port number.