Create a webservice using flask and javascript, where the latency between the user and the server is measured. Use bootstrap library for styling.

Upon clicking a button, measure the RTT from the browser and display the RTT for the user in a line, also showing the time. Measure RTT every second, and keep old entries on the screen. Offer the user a button to stop the measurement. Show the RTT and timestamp on the left 30% of the screen.

The user should send back the response time and append it to a file, using the user’s IP address and date as filename. On the server, store every generated data in a subfolder.

Offer a button to download the latency times with the date and time for the user as attachment. send\_from\_directory() requires 2 mandatory arguments: *directory and path.*

Create a plot of all the previous latency times of the user, with the date and time on the x axis and the RTT on the y axis. Show only the date and time on the x axis on the plotted chart, and do not show the full timestamp, do not show time zone and year. Use a specific version of chart.js as the frontend solution for the plot. Whenever there is a new latency time, replace the old plot with the new plot, where the new plot contains all previous latency times. Make the chart animation-free: when loading a chart, display it without any transition. Offer a button to download the plot. Show the plot on the right 70% of the screen.

Keep some space between buttons and show all of them next to each other close to the top. On the main page of the webservice, tell a short summary what this webservice does, and where the server is located on the internet. Find an automated, cost free and registration free way to tell the server location and name. Don't ask for manual input.

Show all the dependencies for python and javascript too, and how to hook them up into the service. Import all the necessary dependencies in the backend script, and host from the root, i.e. provide a root from “/”. Show the complete backend python script and frontend html in one go. Separate the frontend javascript into an app.js file in the root folder. Start the answer with the front end.

User

latencyData is not defined in the backend.

I get the error: "Uncaught (in promise) Error: This method is not implemented: either no adapter can be found or an incomplete integration was provided." I also get no chart.

Connect the dots in the graph with straight lines.

User

Set y origina to 0.

For the server location, find the geographical location, using an automated, cost free and registration free way to tell the server location and name. Don't ask for manual input.

User

use gevent, and on the backend, create folders if necessary