Assignment 5

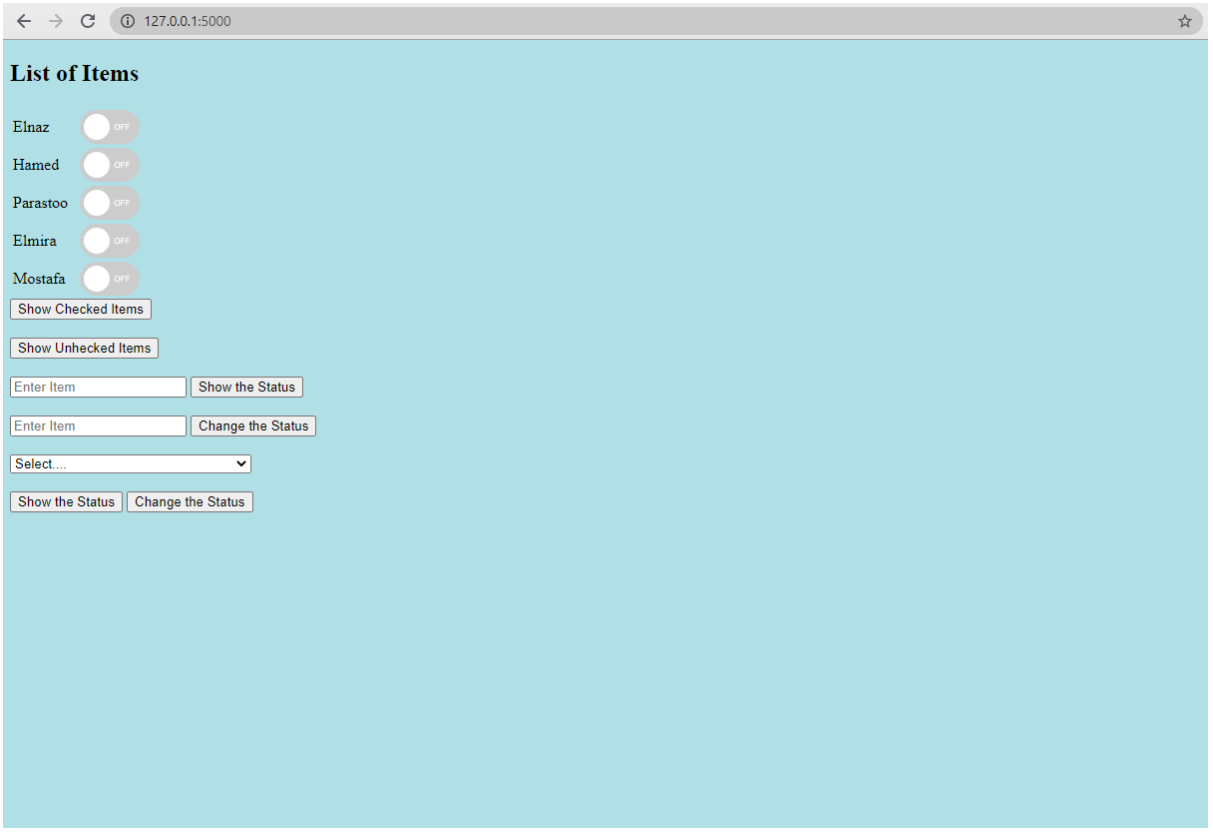
1. Simple Flask Web App

Create a simple flask-based web application that does the following:

- Shows a list of predefined items (choose whatever you like)
- For each item, there is a status of "on" or "off"
- The app shows the predefined list with their status
- Define an input box where you can type one of the items and press a "status" button
- When the button is pressed, the status of the specified item will be shown
- Create another input box that asks whether you want to change the status of the specified items and has a "yes" button that you can click
- If you click that "yes" button it flips the status of that item

Feel free to use Flask or Dash, whichever you prefer.

Here’s an example of what the app may look like. You are free to customize it however you like as long as the basic functional blocks are present.



2. *[Optional]* Directory Details Script

Create a Bash script that goes through a given directory (can be hard-coded) and for each file in the directory prints out the file's: name, number of lines in the file, and size in KB.

- You should use the provided *data-shell* folder (see the Data Files section) as your target directory to go through
- The script should be runnable from the same level as the folder (i.e. the path in the script should look something like *./data-shell/creatures*)
- If you need to add any clarifying comments to your script you can do so using the # character

You can tackle this question by complete these steps, one at a time:

1. How can we reference a directory? (Use a variable)
2. How can we get the files and directories in a folder? (Use a for loop to do this)
3. How can we get check something is a file or directory? (Use an if statement)
4. How can we print the size and number of lines? (Use the *wc* and *echo* commands)

And then combine all of the steps together for a fully functional script.

Optional challenges:

1. Modify your script so that the directory isn't hard-coded but instead given as a parameter when running the script. (e.g. *details.sh ./data-shell/molecules*)
2. Modify your script to go through each subdirectory and print the details of those files as well.