Web Programming Homework

Table of contents

- Include your name in each file you write.
- Place a copy of `hwweb_tester.php` in your web homework folder.
- After you complete each assignment, visit the `hwweb_tester.php` webpage on your browser to make sure it works there. When you complete the assignment, revisit the `hwweb_tester.php` and print it to a pdf file. Save the pdf file in your web homework folder.
- Your `hwweb_tester.php` webpage should look like: hwweb_tester.pdf.

Grading: 25% for each section.

0.1 Basic HTML: Periodic Table

Create a webpage `periodictable.html` that has the following:

- Yellow background and red text.
- A header that says "Periodic Table"
- A table that gives information about the first five atoms in the periodic table.
  - First row should be a header row of fields "Atomic Number", "Symbol", and "Name".
  - Each of the following rows should have information about an atom.
- The table must have thead and tbody sections. Header fields must use th tags.
- There must be borders separating rows and columns of the table.
- Atomic Number and Symbol information should be centered within the column; Name information should be left-aligned.

0.2 File Upload

Create a php script `fileupload.php` that:

- Has a form and a file element to upload a file. The name of the file input element should be "myfile".
- When a file is uploaded, show its name, size, and the first line of text. When printing the first line of text, you must escape any html characters.
- The form should not be shown when a file is uploaded.
- If the form is submitted without a file, display an error message and show the form again.

0.3 Using Sessions

Write a php script `session.php` that

- Has a form to ask the user for their name. The name of the input field should be "myname".
- Once the user submits their name, show their name on the page and don't show the form.
- Closing the page and revisiting it should still show the name.
- Have a link to forget the name. The link should use a url parameter "forgetme=1".
- If the user follows the link to forget the name, the form should be shown again.
- Your php code should accept both GET and POST parameters.

0.4 Running Python from php

Write a php script `prostaterisk.php` that

- Asks the user for:
  - presence of family history of prostate cancer (checkbox),
  - amount of european ancestry (textbox),
  - number of AR_GGC repeat (textbox), and
  - type of CYP3A4/CYP3A5 haplotype (selection menu with AA, GA, AG, GG options).
- When the user submits their information, print their Prostate Risk value using `prostaterisk.py` file you created in a previous assignment.
  - Create a copy of your `prostaterisk.py` file for this assignment and make any changes to make it work with your `prostaterisk.php`.

- Your php script should pass the user's information to the python script `prostaterisk.py` that calculates and prints the prostate risk calculated as numprostate/(numprostate + numcontrol), where the decision tree and numprostate/numcontrol are available from the Figure 3 of "Decision Tree-Based Modeling of Androgen Pathway Genes and Prostate Cancer Risk"
- Your php script should then show the calculated prostate risk on the webpage.
- Your php code should work with either GET or POST parameters.
- When the form is submitted, do not show the form.
  - Use an input field named "submit" determine whether the form has been submitted.