COP 3813 – Introduction to Internet Computing – Dr. Marques

Project 5 – Currency conversion using PHP

Goal

Design a currency conversion app using PHP and a publicly available API.

Recommended Procedure:

1. Make sure you have your **lamp.cse.fau.edu** server space set up. *Please note that you **must** have a server setup for your PHP code to work.*
2. Check the tutorial at http://www.phpzag.com/convert-currency-using-google-api/ and play with the associated demo: http://phpzag.com/demo/convert-currency-using-google-api/
3. Download the **code** from the tutorial (available on Canvas), upload it to your area on the **lamp.cse.fau.edu** server, run it from a browser, and ensure that you understand it.
    - It is a relatively simple, but nevertheless functional, script for currency conversion.
    - It uses the **Google Finance Converter**.
    - It includes 6 ‘from’ and ‘to’ currencies in its pre-populated drop-down lists.
4. **Modify** the starter code in meaningful ways (or start from scratch, if you prefer). You will probably **not** need to change the ***currencyConverter*** function! For every significant change, upload the file to your area on the **lamp.cse.fau.edu** server and ensure that it works.
5. **Implement modifications to make the converter’s look-and-feel different from the starter code**, i.e., clean up and/or (re-)design a good-looking **interface** using **CSS and Bootstrap**.
6. **(OPTIONAL) Implement modifications to the starter code’s *functionality***, i.e., add a novel feature.
7. **(OPTIONAL) Clean up redundant/useless portions (potentially entire files) of the starter code.** Hint: is login.js necessary?
8. **Test your app** after every significant change / addition and **document all meaningful changes in your report**.
9. Once you’ve reached a point where your app is complete and fully functional in the browser of your choice (Chrome, Firefox, Opera, or Safari), **prepare the final package** (single zip, all that is needed, and nothing else).
10. **Submit the final package** via Canvas.

Minimum requirements:

- Your page should demonstrate a basic understanding of using forms and PHP to produce interactive, “data-driven” web pages.
- The conversion must take place on the server (PHP code), **not** on the client (JavaScript example code).
- Your final product should not resemble in any obvious way the code provided to you as starter code.

Deliverables

- **A single zip file** containing **all** files (.php, .css, .jpg, .js, etc.) associated with your project and your report (see below).
- A brief **report** (2-3 pages) describing any relevant aspect that I cannot tell just by looking at the page files.
    - For example: the editor(s) you used, the browser(s) you tested your page on, how was the PHP learning curve, which tasks were more time consuming, etc.

Please name your file using your FAU username as a filename, e.g., jsmith85.zip.

- **Grading rubric:**

PHP code – correctness:	**20%**
PHP code – functionality:	**20%**
PHP code – quality of code (and inline comments)	**10%**
Proper use of CSS and Bootstrap	**20%**
Overall presentation (alignment, colors, fonts, images, etc.):	**20%**
Report	**10%**