Creating a Tailored Resume in HTML/PHP/CSS

Acknowledgements

Thank you to Jason Blanchard and his wife Jackie for encouraging me to document this utility.

When I was losing my job in June 2020, someone in the Black Hills Info Sec or TrustedSec discord told me about the "IRL Job Hunt" Jason (and sometimes Jackie) conducts on Twitch twice per week. I mentioned in the stream comments that I had recently switched from using a spreadsheet to make resumes targeted to a company I was applying to in favor of a web product that makes re-ordering items and en/dis -abling line items much easier. They pushed me to document the tool.

Preamble

First and foremost, I'm not a web developer by any means. My web dev experience is primarily related to utility sites for company internal use (*e.g.* employee directory, IT support ticket reporting) or personal tools (*e.g.* personal finance ledger). This is another such tool. A proper web developer could/should tweak the CSS to wow the viewer, especially if that's the job being applied to.

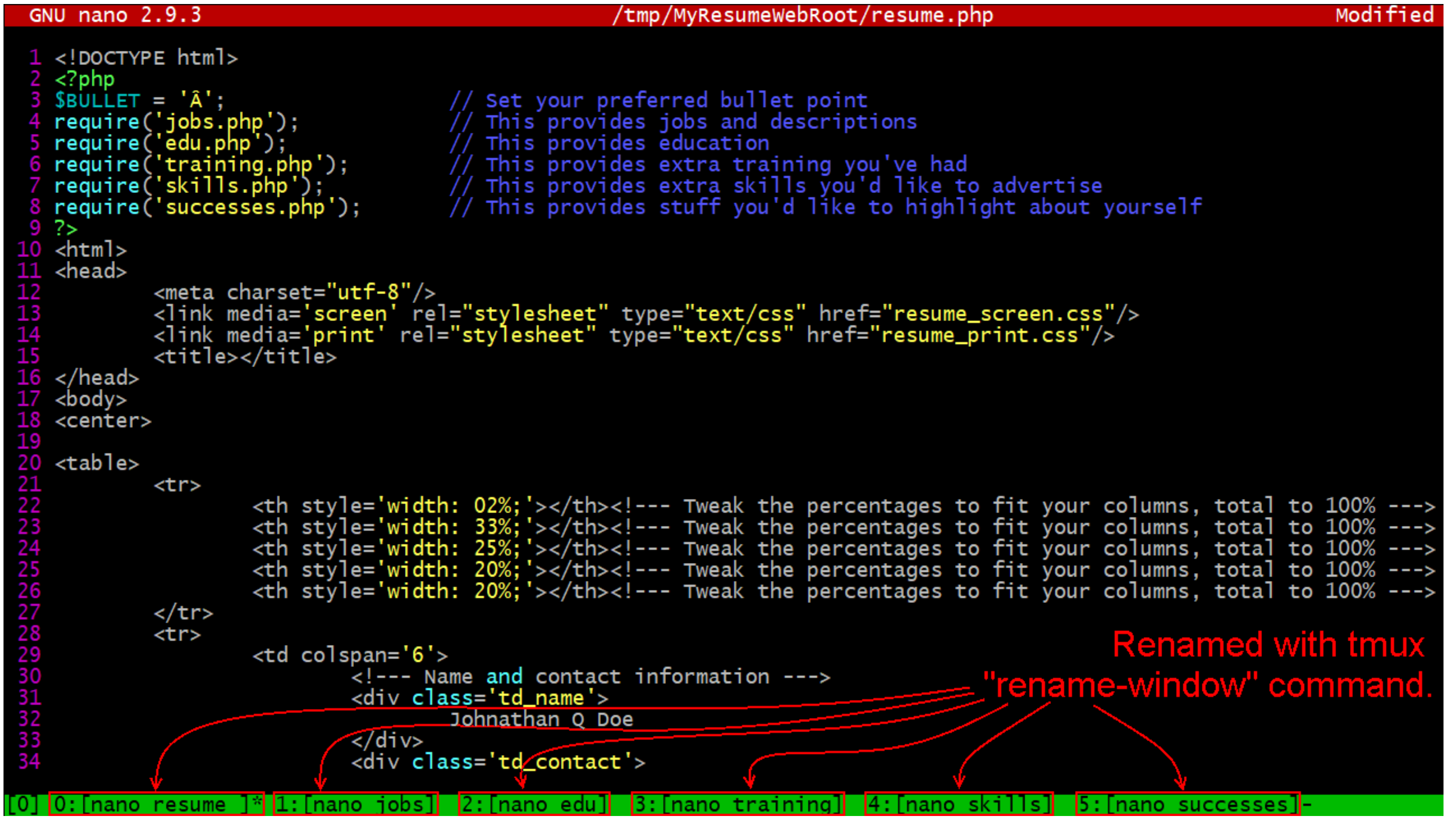
In the past, I used to use a spreadsheet to make a tailored resume because the alignment stayed crisp, I could hide/show rows, duplicate a sheet to make a company specific revision, etc. However, it wasn't quick to re-order items. For example, I could hide a row to not include it, but to reprioritize an item lower or higher in a list entailed inserting row(s), moving the item(s), and deleting the null row(s). Further, I had to ensure I wasn't deleting any hidden rows I might need in the future. Exporting the PDF could be cumbersome too depending on the word processor and the export options that aren't remembered each time.

This tool is intended to make it easier to re-order and/or in/ex -clude items that may pertain to one company and not to another. Chrome web browser retains "Save as PDF" settings as well.

Working Environment

My host platform is Ubuntu Server. The web server is Apache, with PHP enabled. I configured a new VirtualHost running on HTTPS with an SSL certificate from LetsEncrypt. I have personally identifiable information in my resume, so I password protect it with the `htpasswd` utility. I also limit it to connections from specific IP addresses. When I'm not exporting resumes, I disable the VirtualHost with `a2dissite`. One could leave a resume visible and redact the PII.

My workstation is Windows 10. So that I can use a GUI text editor (Notepad++ or Sublime Text) when making many edits, I use MountainDuck SFTP with the root folder being the VirtualHost DocumentRoot folder. MountainDuck is not free. There is a trial version. To remain completely free, one could use `nano` via SSH, and re-order with `Ctrl+K`/`Ctrl+U` to cut and paste one or more items. Pro tip, if editing exclusively via SSH, use `tmux` to open multiple windows in the SSH session and bounce between the windows with `BindKey`+[`n`,`p`,`#`].



```
GNU nano 2.9.3                    /tmp/MyResumeWebRoot/resume.php                    Modified

<!DOCTYPE html>
<?php
$BULLET = 'Â';                  // Set your preferred bullet point
require('jobs.php');            // This provides jobs and descriptions
require('edu.php');             // This provides education
require('training.php');        // This provides extra training you've had
require('skills.php');          // This provides extra skills you'd like to advertise
require('successes.php');       // This provides stuff you'd like to highlight about yourself
?>
<html>
<head>
        <meta charset="utf-8"/>
        <link media='screen' rel="stylesheet" type="text/css" href="resume_screen.css"/>
        <link media='print' rel="stylesheet" type="text/css" href="resume_print.css"/>
        <title></title>
</head>
<body>
<center>

<table>
        <tr>
                <th style='width: 02%;'></th><!--- Tweak the percentages to fit your columns, total to 100% --->
                <th style='width: 33%;'></th><!--- Tweak the percentages to fit your columns, total to 100% --->
                <th style='width: 25%;'></th><!--- Tweak the percentages to fit your columns, total to 100% --->
                <th style='width: 20%;'></th><!--- Tweak the percentages to fit your columns, total to 100% --->
                <th style='width: 20%;'></th><!--- Tweak the percentages to fit your columns, total to 100% --->
        </tr>
        <tr>
                <td colspan='6'>
                        <!--- Name and contact information --->
                        <div class='td_name'>
                                Johnathan Q Doe
                        </div>
                        <div class='td_contact'>
[0] 0:[nano resume ]* 1:[nano jobs]  2:[nano edu]  3:[nano training]  4:[nano skills]  5:[nano successes]-
```

Renamed with tmux "rename-window" command.

CSS styles

The CSS files contain properties for classes and elements. There're also inline `style` attributes. Change these at your preference.

"index.php" file

There're comments in the PHP files where I felt necessary. The "resume.php" file ("index.php" is symlinked to it in my environment) is the primary document. The first few lines

- define the bullet point symbol (not rendered properly in the `nano` window in the picture above)
- `require` the extra PHP files that generate the job history, education history, training, skills, *etc.* variables

The `head` tag includes separate screen and print CSS documents. I prefer to use Chrome so I can tweak the margins when exporting the PDF. The screen CSS shows table row and data borders to see where fields end. The print CSS doesn't include borders.

The `table` header section of the table (`th`) defines the width of each table column. The print and screen CSS render the columns slightly differently such that a long company name may break onto a new line in the print preview or exported PDF. Proofread the document before export and sending to a potential employer.

The first row of the table has your name and extra info. The name is in a `div` that is centered in the `td`. The optional and extra contact info is in another `div` that is pinned to the right and stacked higher (`z-index` property). If you have this publicly viewable, you may not want your PII included. As I mentioned above, I don't have this publicly viewable all the time, so I leave the PII in place, export the PDFs as needed, then stop the VirtualHost.

With respect to the PII, in my resume, I have telephone embedded as

```
<a target='_blank' href='tel:+15555552160'>&#9742; 555-555-2160</a>
```

so that a person viewing on a phone can press the phone number to call me directly. Similarly, the address is embedded as a link that opens Google maps on an Android device.

The extra (SFW) social media pieces are embedded as hyperlinks where the clickable item is the site's "favicon.ico" image limited to 16x16 pixels.

After the first row is the objective field. Put what you want here.

After this section is where the `require` PHP statements above echo the remaining table output.

"jobs.php" and "edu.php" files

The Jobs and Education PHP files have similar formats. The `$arr_JOB` and `$arr_EDU` variables are each an array and contain a nested array of job descriptions or educational highlights, respectively.

```
[
  'Company Name',
  'Job Title',
  'Date Range',
  'Location',
  [
    'Job Description #3 re-ordered',
    'Job Description #1',
    'Job Description #2',
    //'Commented Job Description #4',
    '...',
    'Job Description #n',
  ]
]
```

```
[
  'School Name',
  'Campus',
  'Date Range',
  'Location',
  [
    'Highlight #3 re-ordered',
    'Highlight #1',
    'Highlight #2',
    //'Commented Highlight #4',
    '...',
    'Highlight #n',
  ]
]
```

The array items are ordered by date, oldest to newest. To add a job or school, duplicate the last array item and update accordingly. After the array is created, it's reversed such that the resume is ordered newest to oldest. Change this at your preference.

In any case, job duties and highlights should

- start with action verbs (e.g. "Manage", "Create", "Decrease")
- contain quantifiable figures whenever possible (*e.g.* "10% year over year")
- have correct tense (*e.g.* "Managed", "Created", "Reduced" for a previous position)

After the array reversal, the section header is initialized with its table layout.

In the subsequent `foreach` loop, the job title, employer, location, and dates are printed. The nested `foreach` loop iterates over the job duties/education highlights and prints those with a leading bullet. Note that this structure manually creates a bulleted list. Alternatively, one could change this to be an HTML unordered list (`<ul><li>…</li></ul>`) and change the bullet symbol to match the one identified at the start of the index document.

Regardless of how the bulleted list is written to the media, the power of this tool is illustrated by the quick ability to change the order of the duty/highlight array with

- `Ctrl+Shift+↑/↓` in Notepad++ and Sublime Text with one or more lines selected
- `Ctrl+K` on one or more lines to cut, then Ctrl+U to paste in `nano`

- Yank (`yy`), delete (`dd`), and paste/put (`p`) operations in `vi(m)`.

and un/commenting items with

- `Ctrl+k` to comment `Ctrl+Shift+k` to uncomment in Notepad++
- `Ctrl+/` to toggle comment in Sublime Text
- `Meta+3` to toggle comment in `nano`
- I wish you the best in `vi(m)`

Save the PHP file when the items have been rearranged and/or un/commented.

When the `foreach` loops have finished, the `$sec_JOB` and `$sec_EDU` variables contain all the `tr` items for this part of the table.

“training.php”, “skills.php”, and “successes.php” files

These three files have the same operation as the Jobs and Education PHP files. The primary difference is that the `$arr_TRN`, `$arr_SKL`, and `$arr_SUC` variables are single dimension arrays, *i.e.* they don’t contain nested arrays.

The sole `foreach` loop iterates over the array and builds the `$sec_TRN`, `$sec_SKL`, and `$sec_SUC` section variable, which contains the `tr` data for the respective section.

The last PHP statement in each of the required files is a commented “`echo $sec_`*`ABC`*`;`” statement. The purpose of this is to allow one to make changes and check the output independent of the index file. One could view the source of the page https://myres.example.com/jobs.php to see the output of that individual PHP file.

Returning to the index file

Each of the `$sec_`*`ABC`* variables is echoed to the screen, then the table and body elements are closed.

Conclusion

As mentioned, this tool is utilitarian and creates a structured, informative, no frills resume. It should have some CSS enhancements applied to wow the viewer. Use Chrome’s developer tools to test some CSS enhancements and then migrate those to the separate CSS files or the inline `style` attributes.