

Week 6 Deliverables

Overview: In this week, you have studied additional Python language syntax for rendering HTML templates using the Python Flask module. The Lab for this week demonstrates your knowledge of this additional Python functionality. Be sure to use the examples in the textbook and readings along with the associated libraries, functions and processes when completing the assignments for this week.

Submission requirements for this project includes multiple 2 files. (Zipping them into one file is acceptable and encouraged):

- Python Web Page Code (Python code, Templates, CSS and other associated files)
- Word or PDF file containing your test and pylint results

Python Applications for this lab: (total 100 points):

This lab consists of two parts.

1. **(80 points)** This exercise **(80 points)** uses your programming environment to generate a simple Web site using Python flask. The site should be unique, include at least 3 routes (e.g. 3 pages one can navigate), each route should render the HTML pages by using the `render_template()` functionality. A style sheet should be included that is used by all Web pages. Proper organization should take place of the web site including the location of templates and static pages. Keep in the basic HTML form for a function web page includes the following components:

```
<!DOCTYPE html>

<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    ...your page content...
  </body>
</html>
```

In addition to the requirements list above the following functionality should be found within your **web site** on one or more web pages.

- Use at least 3 different heading styles (e.g. `<h1>`, `<h2>`, `<h3>`)
- Paragraph (`<p>`)
- Comments `<!-- -->`
- Ordered list
- Unordered list
- At least 3 Links to other External Web Sites
- Display the Date and Time on a Web page (Hint: Just use the Python datetime functions)

The content and topic of the Web site is up to you. Consider an information web site about a topic you are interested. It should be unique and something you want to create.

Hints:

1. Be sure to end tags that are started (e.g. `<h1>` `</h1>`)
2. Start early. This will take you longer than you think.
3. Use comments to document your code
4. Test with many combinations.
5. Use pylint to verify the code style – the goal is a 10!

2. **(20 points)** Document your testing results using your programming environment. You should also include and discuss your pylint results for the application. The test document should include a test table that includes the input values, the expected results and the actual results. A screen capture should be included that shows the actual test results of running each test case found in the test table. Be sure to include multiple test cases to provide full coverage for all code and for each function you develop and test.

Any submissions that do not represent work originating from the student will be submitted to the Dean's office and evaluated for possible academic integrity violations and sanctions.