***Documentation Packet [ 15 06 12 24 ] Dec 6th 2024***

|  |  |
| --- | --- |
| Student Name: |  |
| Goals:  1. Assess completion percent of course 2. Learn about EJS Partials 3. Use EJS to make a static page | Events:  1. No school Monday, Dec 2nd |
| Included Documentation  1. Task List Audit 2. Task List 3. CIW Lesson 7 4. EJS Partials and Rendering 5. Reflections 6. Job Openings | Required Documentation:  1. Forum Board 2. CIW Lesson 7 3. EJS Partials 4. Reflections |
| Changes/Notes:  * None | |

# Task List Audit

Included in this DocPac is a Task List spread sheet. This task list contains all of the skills you need to prove you have mastered before graduating.

* For each line in the task list:
  1. Find an example of you doing that task list item in your work (past or present)
  2. Screenshot that example. Use arrows to point out specific parts of the work
  3. Name the screenshot the number of the task list item
  4. Save that screenshot to a folder called ‘tasklist/<firstname lastname>’ in this docpac
     + Obviously, the parts in < brackets > are placeholder text for your name
  5. Commit and Pull Request the screenshots of all the items you can find when finished.

# CIW Lesson 7

1. Using the CIW Submission Rules, complete CIW Javascript Specialist Lesson 7

# EJS Partials and Rendering

EJS partials are segments of EJS templates that can be **included** in other EJS template. The most common way this is used is to have common page information, such as the <head> tag in one partial, then **include** that partial in other templates. This will allow you to make a single change to your <head> and have it affect your entire website.

EJS can also be used to create an HTML page as a string. This is very useful for preparing HTML for applications that are not being rendered in an HTTP request. For example, you can build a fancy page with specific details, then send it in an email. In this assignment, we will create a webpage from EJS and save it as a static HTML page.

1. Create a NodeJS application that has an express HTTP Listen server as usual
2. Configure the ExpressJS application to use EJS as its view engine
3. Create the appropriate ‘views’ and ‘partials’ folder
4. Create a ‘head.ejs’ partial containing the first half of a standard web page
   1. This includes doctype, opening html tag, head, style, script, etc.
   2. Stop before the opening body tag
   3. Use CSS to test
5. Create a ‘foot.ejs’ partial containing everything after the body element.
   1. Typically, this is just the closing html tag. This is fine.
6. Create an ‘index.ejs’ template that **includes** the ‘head.ejs’ partial, and contains the body element. Then include the ‘foot.ejs’ partial.
   1. If the EJS variable “viewport” equals “online”, create a “print” button that redirects the user to “/print”
7. Create two endpoints: ‘/’ and ‘/print’
8. If the user requests ‘/’, render the index.ejs, passing along the parameter “viewport” with the value “online”
9. If the user requests ‘/print’, use EJS to create a page from index.ejs, with the viewport value as “offline”, and save it to a file called “index.html”. Use res.send() to tell the user if the file was successfully written or not.
   1. You will need to learn how to write files with the ‘fs’ module if you have not already
10. Test your work.
    1. Do the partials work? Does the CSS show up?
    2. Does the “Print” button show up on your computer?
    3. Does pressing “Print” write a file and tell the user if it worked?
    4. Does the new index.html page NOT have a Print button in it?

# Reflection

**What caused the biggest delays in your projects the last two weeks? What will you change to prevent those delays from happening again?**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**What was one mistake you made in school or otherwise that you can recognize? What can you do in the future to prevent it from happening again?**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**How much of this year’s lessons were you able to recall to make this week’s project? What can you do to reinforce what you have forgotten?**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

**How difficult was the assignment this week? What made it difficult/easy for you?**

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

# A picture containing text, monitor, screen, clipart Description automatically generatedGrading

|  |  |  |
| --- | --- | --- |
| 10 | You went above and beyond expectations. You applied knowledge that was not taught in this class in addition to what was taught. Additional rewards are given | * ***If the assignment does not have its own rubric, it will default to the rubric on the left.*** * All assignments start at 10/10 possible points * 1 point is deducted per infraction   + Lateness   + Mistakes   + Unprofessionalism   + Not following instructions * Outstanding submissions, or submissions on assignments not marked in “Required Documentation” can reward pogs |
| 10 | You performed as well as can be expected for this class. You show a complete understanding and made no mistakes. You have mastered the subject. |
| 8 | Assignment is complete. You show a good understanding of the subject, but there are mistakes or minor incorrect details. You are ready to move to new subjects. |
| 7 | You show and understanding of the subject, but there are serious errors, or there are pieces you can practically use without understanding them. Remediation needed. |
| 6 | Assignment is incomplete but/or you showed that you understand at least the fundamentals of the subject. Assignment is low effort. Serious need of remediation. |
| 5 | You show minimum effort, assignment is incomplete, or have serious mistakes. You did not demonstrate that you understand the content or purpose of the submission. |
| 0 | The work was not submitted, damaged, seriously incorrect, or unprofessional. The submission is rejected. |

# CIW Lesson 7 *CIW Submission Rules, DP09*

|  |  |  |  |
| --- | --- | --- | --- |
| **1-2, 4** | Correct directory structure, with correct naming convention | |  |
| **3** | Read all pages | Used all Flash Cards |  |
| All Quizzes over 80% | All Exercises over 80% |
| **4** | All labs completed as assigned | |  |

# Task List

|  |  |  |
| --- | --- | --- |
|  | Accurate and clear screenshots |  |
|  | Percent of Task List Items Complete |  |
|  | Followed Instructions |  |

# EJS Partials and Rendering

|  |  |  |
| --- | --- | --- |
| **10a** | Do the partials work? Does the CSS show up? |  |
| **10b** | Does the “Print” button show up on your computer? |  |
| **10c** | Does pressing “Print” write a file and tell the user if it worked? |  |
| **10d** | Does the new index.html page NOT have a Print button in it? |  |

# DocPac and Reflection *DocPac Submission Rules, DP09*

|  |  |  |
| --- | --- | --- |
| **1** | DocPac is turned in on time |  |
| **3** | a. DocPac is neatly folded |  |
| **3** | b. DocPac is not stained or damaged |  |
| **3** | c. No doodles, scribbles, or unnecessary writing |  |
| **4** | a. Answered each question in each prompt |  |
| **4** | b. Spelling and handwriting |  |
| **4** | c. No repeated answers from other DocPacs |  |
| **4** | d. Did not paraphrase assigned work |  |
| **6** | You are prepared to justify the use of any AI (you know what it does and why) |  |