***Documentation Packet [ 9 25 10 24 ] Oct 25th 2024***

|  |  |
| --- | --- |
| Student Name: |  |
| Goals:  1. Identify risks of hosting user data 2. Use OAuth2 | Events:  1. DocPac Due Monday 2. Two-hour Delay Wednesday |
| Included Documentation  1. Formbar OAuth2 2. User Login Handling 3. CIW Chapter 4 4. Nodejs Submission Rules 5. CIW Submission Rules | Required Documentation:  1. Formbar OAuth2 2. User Login Handling 3. CIW Chapter 4 4. Reflections |
| Changes/Notes:  * None | |

# Formbar Oauth2

1. Create a new Nodejs project using the Nodejs Submission Rules (DP09)
   1. Use ExpressJS, EJS, SQLite3, and jsonwebtoken
2. Go to the WIKI of Formbar.js: <https://github.com/csmith1188/Formbar.js/wiki>
3. Create a new database file with a ‘users’ table containing the following rows:
   1. fb\_name
   2. fb\_id
   3. profile\_checked
4. Use the documentation on the OAuth2 page to create a web application in which users can log in via Formbar.js
   1. If you are working at home, consider forking Formbar.js and running it locally on your machine to test.
   2. If you test on a Formbar other than the one in class, be sure to configure the program to work on the one in class before submitting.
5. This web application should have four endpoints:
   1. ‘/’ – GET – The homepage with links to ‘/login’ and ‘/profile’
   2. ‘/login’ – GET – If the user is already logged in, redirect them to ‘/profile’, otherwise, use the Formbar.js login
   3. ‘/profile’ – GET – If the user is logged in, display a form with the user’s Formbar name and one check box the user can change and submit. The state of the checkbox is determined by its value in the database. Otherwise, use the Formbar.js login
   4. ‘/profile’ – POST – If the user is logged in, handle changing the checkbox from ‘/profile’. Save the changes to the database.

If at any point the user logs in with Formbar and their name and id are not already in the database, INSERT the information into the database.

# CIW Chapter 4

1. Using the CIW Submission Rules, complete CIW Javascript Specialist Lesson 4

# User Login Handling

**What are common security risks when storing user login data, and how can developers protect against them?**

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

**What are the consequences of a data breach for both users and developers?**

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

**What legal responsibilities do developers have when storing sensitive user information?**

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

**What are the costs and challenges of maintaining database security, and how can poor management increase risks?**

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

# Reflection

**What personal projects have you worked on recently? What do you think the impact is of doing/not doing personal projects has been?**

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

**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?**

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

**If your name is Hayden, what is a second mistake you made this week? What can you do in the future to prevent it from happening again?** (If your name is not Hayden, skip this question)

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

**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. |
|  |  |  |

# CRUD Research

|  |  |  |  |
| --- | --- | --- | --- |
| Answered Each Question | Questions were accurate and sincere |  |  |

# Formbar Oauth2

|  |  |  |  |
| --- | --- | --- | --- |
| Pull Request correct | Works as described | Did not commit  ‘node\_modules’ folder |  |

# Reflection

|  |  |  |  |
| --- | --- | --- | --- |
| Selected an answer for each question that is unique to you and this week | Answered every question in each prompt | Answers were not repeats of previous weeks | Answers were not copies of assigned work this week |