***Documentation Packet [ 87 10 10 24 ] Oct 10th 2024***

|  |  |
| --- | --- |
| Student Name: |  |
| Goals:  1. Store and retrieve encrypted entries from a database 2. Identify CRUD and HTTP Methods | Events:  1. DocPac Due *Next Tuesday* 2. No School Friday 3. No School Monday |
| Included Documentation  1. Login Basic System 2. CRUD Research | Required Documentation:  1. Login System 2. CRUD Research 3. Reflections |
| Changes/Notes:  * None | |

# CRUD Research

Research online and answer the following questions (do not ask AI)

**What does CRUD stand for?**

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

**Why do we use this model?**

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

**What is the appropriate HTTP Request Method for each of the four principles?**

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

# Login Basic System

1. Create a new NodeJS project with ExpressJS and EJS
   1. Name the main file ‘app.js’
2. Import the `sqlite3` npm module
3. Import the native `crypto` module
4. Set up the `crypto` module to use a key
5. Create a database file using DB Browser (SQLite)
   1. Create a table called “users” with four columns:
      1. “uid” is unique, not null, primary key, and auto incrementing
      2. “username” is not null and unique
      3. “email” is not null and unique
      4. “password” is not null
6. Create a ‘/’ GET endpoint that renders an ‘index.ejs’
   1. This links to ‘/login’ and ‘/signup’
7. Create a ‘/login’ GET endpoint that renders an `login.ejs`
   1. This has a form asking for the username and password.
      1. Make sure the password field is starred out so users cannot read the password
   2. The form has a submit button, with the method `POST` and the action `/login`
8. Create a `/login` POST endpoint
   1. Use try/catch or callbacks to catch errors. Redirect to `/error` if there are problems
   2. Check for a username and password and validate the information
      1. Manually throw and error if something is not valid
   3. Use `crypto` to encrypt the password with your key
   4. Use sqlite3’s `get()` to find a user WHERE the username and encrypted password match what the user provided
      1. If there is an error, manually throw an error
   5. If there are no errors, redirect to ‘/home?user=<username>&email=<email>`, where <username> and <email> is the user found in the database.
9. Create a `/signup` GET endpoint that renders a `signup.ejs`
   1. This has a form that asks for their username, email address, and password
   2. The form has a submit button, with the method `POST` and the action `/signup`
10. Create a `/signup` POST endpoint
    1. Use try/catch or callbacks to catch errors. Redirect to `/error` if there are problems
    2. Check the username, email, and password and validate the information
       1. Manually throw and error if the information is not correct
    3. INSERT the new information into the database
       1. If the username and email fields are set to unique, sqlite3 will have an error if you try to use the username or email again. Manually throw an error if this is the case
    4. If there are any errors, redirect to `/error`.
    5. Otherwise, redirect to `/login`
11. Create an `/error` GET endpoint that renders `error.js` that simply states there is an error
12. Create a `/home` GET endpoint that renders `home.ejs`
    1. This reads the `user` and `email` query parameters and displays them in the page
13. Test your work thouroughly
    1. Make sure you can sign up and log in, but also try everything you can to break it
14. When complete, delete your node\_modules folder and commit, push, and Pull Request

# 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?**

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

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

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

**What is fundamentally wrong with the Login Basic System we did this week?**

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

# 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 |  |  |

# Login Basic System

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