Topic: Milestone 1 – Create user registration page

Date: June 14, 2019

Revision: 1

Team: Matt Sievers

|  |  |  |  |
| --- | --- | --- | --- |
| Tasks | Team Member | Hours Worked | Hours Remaining |
| Create login.html form should include the following inputs: username  Password  Login btn | Matt Sievers | 1 | 0 |
| Create loginHandler.php  Basic form construction to handle the input from the login.html page. The following items are needed:  validate fields on the page are not blank or empty. create a way to log visits or attempts on password entry. offer “try again” button that redirects back to login | Matt Sievers | 2 | 0 |
| Password encryption needed. Use the password\_hash to create a hashed representation of the password and use password\_veryify as needed. Ensure that password and confirm password match on the userRegistrationPage.html | Matt Sievers | 2 | 0 |
| Update userRegistrationPage:  Password input needed confirmPassword input needed | Matt Sievers | 1 | 0 |
| Update createUser.php  Add password encryption, ensure that password and confirmPassword match.  Update the insert statement to include the new hashPassword. | Matt Sievers | 20min | 0 |
| ***BUG*** – update the logic checks to see if the fields are blank to include an exit after the echo. Users were being created even if fields were blank | Matt Sievers | 20min | 0 |
| Create index.html add buttons for login or register login should redirect to login.html register should redirect to userRegistrationPage.html. | Matt Sievers | 20min | 0 |
| Create homepage.html  Need a place to land on successful login. | Matt Sievers | 5min | 0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

GIT/Bitbucket URL: <https://bitbucket.org/mSievers82/cst126/commits/all>

Hosting URL: NA

Peer Reviews: NA

Install Instructions: execute the script to create the user\_info table and ip table. Ensure that MAMP is installed and both apache and mysql servers are up and running. Navigate to <http://localhost/CST-126/Milestone/index.html> to interact with the form.

General Technical Approach: Ensure that required fields were tested to not be blank or not set from the POST request, if so throw an error. Realized there was a bug that was introduced with this approach and an exit was needed in all these logic attempts to ensure that bad users weren’t created. Password was required for the user creation page to allow future logins. Needed to ensure the password was \*\*\* out as well as make sure it wasn’t stored in the DB as a literal string. I used the password\_hash to create a hashed representation of the password that could be stored in the DB. I also wanted a way to track the attempts on login and found a great resource that created a new DB table to track IP traffic and limit it based on the amount of attempts within 10 mins (stack overflow 2019). Also needed to add some logic to check a password restriction of cap, lowercase and number. Found another great reference at codereview.stackexchange.com (2019).

Key Technical Design Decisions: track IP for login attempts and how to validate a password using regex.

ER Diagram: NA

DDL Scripts: NA

Sitemap Diagram: NA

User Interface Diagrams: NA

Other Documentation: NA

References:

https://codereview.stackexchange.com/questions/173831/user-id-password-constraints-checker

Stack overflow https://stackoverflow.com/questions/37120328/how-to-limit-the-number-of-login-attempts-in-a-login-script