Project Iteration 2: 3 Items Fixed with Report

Video of website: https://www.youtube.com/watch?v=iNxV4df0NGU

Documenting / Panning Issues:

Some planning issues that were not completed during the initial planning for the project were understanding how hosting and databases work online. I had originally started working on the database with Postgres, but then found out that GoDaddy doesn't fully support Postgres, it only really supports MySQL. So, then I had to download Xampp and create a new database in MySQL while learning PHP at the same time. My lack of skills, knowledge, and the Dunning Kruger effect led me to believe that I would be able to do it all in a short time easily. Which I was completely wrong with. When I sent my project over to my uncle in Michigan that works with PHP in the backend, he found lots of flaws with my code and told me I needed to fix them asap. Since he was on vacation, he couldn't really help me all that much without me being too much of a bother. While I still have a good number of bugs and issues that need to be 100% sorted out, a lot of the main issues are fixed, and I'll be showing it in the next section. Next time I plan a project, I will not mark things that I do not know how to do as easy or low. Nevertheless, this experience has helped me tremendously in understanding how the backend works and how databases work. Also, next time I'm going to make sure to read on what services I will be using, so that I can use other products that are compatible with what I will be hosting the website on.

Fixed Code:

```
        echo '<form action="includes/logout.inc.php" method="post">
    <button type="submit" name="logout-submit">Logout</button>
</form>';
    }
    else{
        echo '<form action="includes/login.inc.php" method="post"> <!-- use method=post for security -->
    <input type="text" name="mailuid" placeholder="Username...">
    <input type="password" name="pwd" placeholder="Password...">
    <button type="submit" name="login-submit">Login</button>
</form>
```

Description of image above: using post method to help encrypt form with HTTP and then using type = "password" so when the password is being entered, it is not show. Good for when people or cameras are around!

```
post method to check if they got here legit
(isset($_POST['signup-submit'])){
```

Description of image above: used that so this inc.php file is only able to be accessed when it is from the form and not from the URL. This is to fight against cross site scripting.

```php
// prepared statement is safer so no php injection
$sql = "SELECT uidUsers FROM users WHERE uidUsers =?";
$stmt = mysqli_stmt_init($conn);
if(!mysqli_stmt_prepare($stmt, $sql)){
    header( header: "Location: ../signup.php?error=sqlError");
    exit();
}
```

Description of image above: using prepared statements makes sure that whatever the user enters is treated as a string literal in SQL

```php
// mysqli_real_escape_string() used so that escape special characters in a string
//help prevent php injection
$username = mysqli_real_escape_string($conn,$_POST['uid']);
$email = mysqli_real_escape_string($conn,$_POST['mail']);
$password = mysqli_real_escape_string($conn,$_POST['pwd']);
$passwordRepeat = mysqli_real_escape_string($conn,$_POST['pwd-repeat']);
```

Description of image above: mysqli_real_escape_string escapes special characters in a string for use in an SQL query.

pwdUsers
$2y$10$5Tp2G.fUV0pjb5zyN4p6GeocGi95JJX1RnrQaT7j41W...
$2y$10$VfRnyDiZmtKwCUGDtPvty.y/9tE7.HjvHXRB33aqCC0...
$2y$10$qBeJWuUaxEU.k51jmso54OlMZeSG24wYsAlr.ECHCzF...
$2y$10$99xUq7/x1s0kX1/P47J80uuKeZZ39PrK8qcwhVOOplb...
$2y$10$gQlSa6liSgkwrGjOQzkoYOdAlsyssaUi5hbye7wyDSB...
$2y$10$94SnimacgAFlgJR9n.ssrOMKzuF9o6h58VqXC1ViKoo...
$2y$10$6fcC7rYJnQc.xp/nIU5Al.II9HJTwppV09fHMH09b8Q...

Description of image above: the nice, hashed password in my db safer from when I had them stored as regular text.

```php
else{
    // hash password for security
    $hashedPwd = password_hash($password, algo: PASSWORD_DEFAULT);
    mysqli_stmt_bind_param($stmt, types: "sss", &vars: $username, &vars: $email, $hashedPwd);
    mysqli_stmt_execute($stmt);
    header( header: "Location: ../signup.php?signup=success");
    exit();
}
```

Description of image above: function found on youtube that hashes passwords

How an Online Repository saved me:

Thanks to GitHub my uncle that was on vacation in Guatemala was able to view my code no problems. He helped me find simple yet important security issues that can cause major flaws on the website. Nevertheless, other people that I ask to do security checks on my website can easily download it and check it out for themselves. It increases security because people can find problems and report them. Its good for projects in groups because I can see changes made and who made them (accounting).

Security testing:

A form of security testing I can do is allow some of my family that are in programming and security to look at my website and try and break it. I can also use programs like Acunetix that conduct security testing by trying to break into the website with its knowledge of over 6k vulnerabilities. Only problem is that it cost money