Meena Mall

Quiz 2

Same instructions as quiz 1: read everything twice before answering questions.

When you are finished, you are free to leave. Quiz is open notes, open Internet. The only things you can't do are Generative AI of any kind, run Live Share, talk to each other (or any other students), and post the questions on StackExchange and the like.

You must sign the sign-in sheet at the front of the room. No name on sign-in sheet = 0 on quiz.

Good luck!

Part 1

1.1. Explain three possible features of a web application that require (or, at least, made easier by) a server-side component written in a language such as PHP. Don't just mention the feature, explain in detail what it involves.

In this class, we have had both labs and lectures that have involved phpMyAdmin and MySQL. We use this to create databases with various tables and organize information. It is very important to have a database so that we can store data on the back-end portion of our application. One possible feature is in the case of using a backend server. It is easier to perform functions such as data storage retrieval in which data that is imputed on the front end can be organized on the back end, similar to using PHPMyAdmin. Also, you can easily get the data back due to it being saved on the back end. Another feature is when on a backend server, you can use PHP to streamline code across multiple pages. You can use the same code to create various pages and make them all look organized. Also using a backend server is great to utilize when implementing a user login because it will allow us to for authorization. Also, to ensure that the user has permission and is granted access to view and editing on a page. This is also very important to use as a server-side component because you can store the user's login information as well.

1.2. Explain two actions that can be taken to secure a web application. These may be relateduser authentiactiontion & authorization, server configuration, codebase, and/or network infrastructure. Don't just mention the feature, explain in detail what it involves.

One action is Instead of storing plain text passwords another approach a user can perform is using other methods such as hashing or salting(what we learned in class wink wink lol). In this case, a user can create an account, and the system will implement hash functions to the password inputted and convert it into a unique code so that later on, when a user logs on, the system will apply the same hash function to its password and will compare it to the already stored code. This only works if both match because if they don't, then the user access is NOT granted. By using this method, will not only make it difficult for hackers to hack into your systems but also prevent security breaches and protect your password from all kinds of harmful actions and will make it overall a more safer system. Another action is in the case of a user logging into a well-known or not very well known website where they have to input their username and password. This scenario everyone experiences because every time you input these credentials, the system will check if this provided information matches what's stored for that specific user. Again, if there's a match between both, the user will gain access to continue. If it is not a match, then the system will provide feedback to the user on the specific part of the login information that is wrong. Doing this will make sure that only authorized users are granted access to access the system which makes the overall system more safe and secure and will prevent harmful actions from the outside such as unauthorized entry.

Part 2

Explain this code segment in two different ways: first, explain the overall picture without using any technical jargon, as if you were explaining the code to someone who doesn't understand any programming, and; second, explain in as exacting detail as possible, line by line, what the code is doing. If there are any mistakes or errors in the code, fix them inline using a different color. If you come up to me to tell me there are mistakes, -5 points.

```
if (isset($_GET['lname'])) {
      if ($_GET['lname'] != '') {
            $pstmt = $conn->prepare('SELECT * from customers WHERE
lname = :ln');
            $pstmt->bindParam('ln', $_GET['lname'], PDO::PARAM_STR);
      } else {
            echo "lname not given, outputting entire file";
            $pstmt = $conn->prepare('SELECT * from customers');
      }
      $pstmt->execute();
      while ($row = $pstmt->fetch()) {
            printf("%s %s",$row['fname'],$row['lname']);
      }
}
```

In the provided code above, is a very simple process of checking to see if a user mentioned a last name in the web address. In this case, if there is a last name already provided then the code creates a special SQL query request to its database which will be the customers table. This code looks for the people in the entire file whose last name matches to the last name mentioned. If there is no last name shown then a statement will shown basically saying that the lastname (lname) is not given and it is outputting the entire file. After that, it looks into the file and tells you the first and last names of everyone. So in this case, it is very simple, it is either finding specific people with a given last name or showing everyone if no last name is given. Also, this code uses a 'prepared' statement as well to prevent SQL injection.

Row 1: Retrieves the last name ('lname')

Row 2: This verifies that 'lname' is not empty

Row 3: This line of code establishes a connection to the database using a prepared statement and the SQL query selects rows where 'ln' matches the provided 'lname' which is in the 'customers' table

Row 4: The prepared statement associates 'ln' with 'lname' using the get method. Also, the PDO assigns a variable but as a placeholder

Row 5: If 'lname' is not provided

Row 6: It uses the echo command and it prints a statement - "lname not given, outputting entire file."

Row 7: A prepared statement is ready to connect to the database and to fetch information from the 'customers' table

Row 8:The prepared statement is ready for execution

Row 9:This displays information in both the 'fname' and 'lname' rows

Part 3

Here is a Google Drive folder containing all the lectures for my Modern Binary Exploitation course:
https://drive.google.com/drive/folders/1rjc___npAFJn2oyz-1QpNpyH2qqHankwQ?usp=sharing

Extend your (or create a) mini-LMS application to include MBE. You should create a JSON object that represents the lectures. Each lecture should be represented with a Title, Description, and Link. Since I know (most of) you have not (yet) taken MBE, you may use what you see on the opening slide for each lecture as its Description.

When the user logs into the mini-LMS, they should be able to select which course to display. The left-hand side should be dynamically generated from the appropriate JSON object. Clicking on an item on the left-hand side should populate the preview window containing the Title, Description, and Link for that particular item.

Add a button that, when clicked, switches from one course to the other. Clicking this button should destroy and dynamically regenerate the left-hand side with the other JSON object.

Make sure your archive button is able to archive both the MBE and Web Systems courses.

Creativity counts for this! Don't just stop once this works. Showcase all your talents in HTML, CSS, Javascript, PHP, and MySQL.

README.md Don't forget a readme! Briefly explain your solution and any issues you faced. Tell us what you'd like to have considered for creativity. Don't forget to put your citations! **No citations = 0 on quiz (that's plagiarism!)**

Submission

- Put everything into a quiz2 folder on your personal GitHub repo
- Part 2 must be hosted on your VM at https://[FQDN]/[your-repo]/quiz2

Rubric

Part 1	15 Points
Part 2	15 Points
Part 3:	
JSON object	10 Points
HTML/CSS/JS/PHP/MySQL	40 Points
Creativity	10 Points
README.md	10 Points
Total	**100 Points**

Extra Credit (+5 points)

What are the lyrics to the Alma Mater of RPI. No typos. All or nothing.

Here's to old RPI, her fame may never die.
Here's to old Rensselaer, she stands today without a peer.
Here's to those olden days,
Here's to those golden days,
Here's to the friends we made at dear old RPI.

(im gonna sing this on nov 28th haha)

Extra Credit 2 (+1 point to your final grade at the end of the semester)
Get **everyone** in the class to sing the Alma Mater at the same time on November 28. All or nothing. If even one person doesn't sing, no extra credit.

OKAY !