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. 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.

Transfer data, PHP can transfer data from front to end, and this can make us use a database to save them and use it to do something.

Security, One way to ensure the security of user data is to implement authentication and authorization mechanisms. The backend can securely handle login information, authenticate users, and control user access to specific resources.

Integrate third-party services: Many applications need to integrate with third-party services such as payment gateways, email services, and social media APIs. The backend can serve as a middle layer to handle complex authentication processes and data exchanges, while also providing a simpler interface for the frontend.

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

Use salt, it is a hash table to store passwords and create different hashes to make sure everyone’s password is unique.

Authenticate, like RPI’s duo, It can remind you that you are in the login process to confirm whether it is really you.

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']);
      }
}
```

If we get 'lname' and 'lname' is not empty,the application will bind 'ln' to 'lname' else it will report an error and .Than execute pstmt.Final, print first name and last name.

Make sure we get 'lname'

Make sure 'lname' is not empty

If is not:

Set a variable called pstmt that is table conn's all customers who lname is ln. And blind 'ln' to 'lname'

If empty:

report error

Set a variable called pstmt that is table conn's all customers.

Then execute pstmt

Final output each fname and name in pstmt

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.

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.