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.

Working with a server-side language like php has many benefits stemming from you being able to store data long term. The first benefit is being able actively process data. If you can get sanatized user data, then store it in a database it can be very beneficial. Additionally, being able to generate content based of that data is a huge benefit. You can generate and change a page based on the contents of a database, and having php gives you access to those databases. Finally, having user authentication for multiple users becomes a lot easier as you can add user info to a database and then check login info through the database. Then using that info you can find out the permsissionsthey have.

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.

The first action that can be taken to secure a web application is to sanatize the user inputs to make sure they are not trying to inject any attacks. This involves reading the user input and removing/not allowing the input to pass through if a command is attempted to be passed through. Additionally, another very useful action that can be taken is user authorization. If you do not take the correct steps to authorize your users, permissions can fall into the wrong hands and lead to malicious actions in the application. One of the most basic ways to remedy is to enforce users to use a strong password.

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'])) { #Checking to see if the user provided a
last name using GET

     if ($_GET['lname'] != '') { # Checking to see if last name
isn't empty

          $pstmt = $conn->prepare('SELECT * from customers WHERE
lname = :ln'); #This is the line to retrieve the customers with the
same last time.
          $pstmt->bindParam('ln', $_GET['lname'],
PDO::PARAM_STR);#This bindsthe value of lname to :ln

     } else {
          echo "lname not given, outputting entire file"; #echo
prints this to the user
          $pstmt = $conn->prepare('SELECT * from customers'); #Gets
all customers since last name not provided

     }
     $pstmt->execute(); #Executres sql statement
     while ($row = $pstmt->fetch()) { #iterates over records
          printf("%s %s",$row['fname'],$row['lname']); #Prints the
data
     }
}
```

This code checks if a user has provided a last name, then retrieves the records from the database from that last name. If a last name is not provided, then it retrieves every record from the database. Finally it prints the first and lasat name of every retrieved record.

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.
Oke