**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.
  + Authentication and Authorization
    - PHP can authenticate users and give them authorization based on their permissions
    - Form data is received from html in the form of a user name and password
    - The password is hashed and salted then the username and password are sent to the database
    - The username and encrypted password are compared to what is in the database
    - If it is match a session token is sent back allowing the user to access all data there account has authorization to use
  + Display dynamic content with data from database
    - Data from a database can be loaded onto a webpage and displayed which can be done with other methods but PHP enables use of your own server and better control of the data.
  + Modify Database
    - PHP can be used to modify a database via pdo and msqli which mean you can store and display from forms.
* 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.
  + User-authentication and authorization
    - This is effective at securing sites because it allows control over what users can access and modify. This helps ensure that potential exploiters have limited ability to take down or damage your site.
  + Data encryption
    - HTTPS encrypts data but using secure APIs is also crucial as unsecure APIs leave you open to being hacked through hackers intercepting the data and information received by them.

**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.

This code takes a user’s input which is a last name then it asks the data base for all the customers with this last name and prints out everyone with the last name. If no last name is given, it prints the all the customers.

Checks to see if the variable $\_GET lname exists i.e. is set or declared to enter   
if (isset($\_GET['lname'])) {

If the variable is not an empty string enter and perform the computation  
 if ($\_GET['lname'] != '') {

First prepare the statement where it is selecting all data from the customers table where the ln column value is equal to the lname string  
 $pstmt = $conn->prepare('SELECT \* from customers WHERE lname = :ln');

Bind the $\_GET lname variable to the prepared statement as a parameter  
 $pstmt->bindParam('ln', $\_GET['lname'], PDO::PARAM\_STR);

If the if statement above is not satisfied perform the following computations  
 } else {

Let the user know the $GET\_ lname variable was an empty string  
 echo "lname not given, outputting entire file";

Prepare the statement to select all data from the customers table  
 $pstmt = $conn->prepare('SELECT \* from customers');  
 }

The DB executes and performs the prepared SQL statement  
 $pstmt->execute();

While there is a row from the result set of the SQL query print the first and last name in the row  
 while ($row = $pstmt->fetch()) {  
 printf("%s %s",$row['fname'],$row['lname']);  
 }  
}

**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.*