Nicole Spaulding
Quiz 2
Websys
Section 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.

One feature possible is user authentication and the managing of a session for a user. This is made much easier (and safer) when using a server to keep track of user id's, roles, etc. as they can be stored in secured databases that use hash functions/salting without exposing the hash to malicious actors on the internet like client side javaScript (JS) might do. Another feature is making API calls, which is again much easier with a server to keep secure for similar reasons of being able to make API calls without potentially exposing the API keys, something that is blocked by CORS usually for JS. This could additionally be useful to be done on a server if there are costly computations that need to be done on the API data, which can be done server side to prioritize a quick client experience. A last feature might be form validation, which can be done with JS yet can be disabled, where on a server could be done in a way that is securely hidden from malicious users that may be trying to sneak detrimental data into a database.

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.

Something we learned about in the club RPISEC is how SQL injections can be made. One action can be taken to prevent these is by using a prepared statement (ex: PDO) in order to avoid having direct insertion of the input field into the DB queries. EX: $pdo->prepare("SELECT Country, ReportingDate, Cases FROM rift_valley_data WHERE Country = ? AND ReportingDate = ?");
This statement from our project demonstrates that this statement is prepared beforehand and only later does execute() supply the fields, preventing malicious inserts.

A second action is what was mentioned in class, password salting and hash functions. These work together so that even with something like a rainbow table attack, having a unique salt allows for more secure passwords. With a complex enough hash function in addition, this makes it extremely difficult for malicious actors to gain access to user accounts, securing a web application.