**ITWS Quiz 2**

**Tyler Haynes**

**=====================================================================**

**Part 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 what it involves.**
  + One feature is back-end user authentication. PHP is needed in order to take the inputs of a form (that in this case would be asking for a username and password) and compare against a database to validate whether or not the input user exists within it. If so then they are granted access to whatever they’re being locked out of.
  + A database is also a server-side feature that is very useful for a web application. Databases can hold an extremely large amount of information that can then be queried for by a user on your web application.
  + A web server is also nice if you want to build a web application that involves some kind of file transmission. Say if you make a site where a user can download books or movies-you need a server (such as an FTP server) that hosts this content that a user can then download and have delivered to their computer from your website.
* **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.**
  + One simple way to add security to your website is to ensure that you always sanitize user inputs. This will prevent you from falling down to simple tricks such as SQL Injections which would allow users to manipulate, and potentially outright delete a database of yours. Also prevents them from easily bypassing a password field and being granted access to accounts that aren’t theirs.
  + Another simple way to secure your web application is to use HTTPS over HTTP for your web app. HTTPS is encrypted and will prevent those using programs like wireshark sniffing for packets from being able to examine all web traffic being sent over such as plaintext passwords. With regular HTTP an attacker sees packet info displayed as:
    - GET /hello.txt HTTP/1.1
    - User-Agent: curl/7.63.0 libcurl/7.63.0 OpenSSL/1.1.l zlib/1.2.11
    - Host: www.example.com
    - Accept-Language: en
  + However with HTTPS it is just a random configuration of letters and numbers that they cannot decipher:
    - t8Fw6T8UV81pQfyhDkhebbz7+oiwldr1j2gHBB3L3RFTRsQCpaSnSBZ78Vme+DpDVJPvZdZUZHpzbbcqmSW1+3xXGsERHg9YDmpYk0VVDiRvw1H5miNieJeJ/FNUjgH0BmVRWII6+T4MnDwmCMZUI/orxP3HGwYCSIvyzS3MpmmSe4iaWKCOHQ==

**=====================================================================**

**Part 2**

**=====================================================================**

* **Code for 2.1**
  + Explanation: What this code is essentially doing is that it is asking a database if it has a parameter called “lname.” If it does then it will select from that database and print all last names inserted into it. If it discovers that a last name was not supplied, then it will simply output the entire contents of the database table that it is reading from.
* **Code for 2.2**
  + Explanation: This is a jQuery function that runs when a user clicks on an element (this could be a button for example) of a site. Once the click happens the script then opens up a file named “people.json” which contains the names of people and their professions. After it does this, it loops through the file, going through all the pairs (the pairs, again, being made up of a name and then a profession) and sends out an alert for each pair. The alert pops up from the top of the screen (at least on Chrome) and will have a dark grey text box and might, for example, say “Brian, Professor.”

**=====================================================================**

**Part 3**

**=====================================================================**

* **Add to ReadMe**
  + Fixes for Insert statement provided is listed on ReadMe
  + Along with general explanation