**1.1**

* PHP can create dynamic web pages instead of just static ones. For instance, PHP can do server side processing, such as connecting to a database to get information, then generate a webpage based on it, whereas HTML could only generate static pages server side.
* PHP can accept data from forms on the server. This allows the server to receive information from the client which can then be used to affect a database or the like. The form would need to have an action to a php file, then the form fields could be accessed with $\_POST or $\_GET arrays.
* PHP can allow a user to have the same state across multiple devices. Implementing authentication can allow the server to store personal information which can then be displayed across any device, rather than storing the personal information in the client browser.
* PHP can be used to provide user permissions to certain resources. The server can require that the user complete certain security measures in order to receive access to resources. Then when a user requests a protected resource, the server can make a decision based on the authentication about what to send to the user.

**1.2**

* To protect against SQL injection, prepared statements in PHP should be used to ensure that nothing inputted by the user can be run as SQL code.
* It is strongly recommended that any production database cannot be accessed remotely from outside the server that is using it. The PHP server should be able to access it locally, but the database port blocked to the public, and remote connections not allowed to prevent vulnerabilities to people trying to access it.
* Never store environment variables, database passwords, or secret keys in a repository, or send them in an email or anything that would compromise their integrity. It is a common mistake to add them to a GIT repo, but this is a very bad idea.

**2.1**

This code allows a user to get a list of customers from the database, optionally filtered by last name.

First it checks that a last name GET variable was set. Then, if it is, it checks if the last name is blank or not. If it is not blank, it will prepare a SQL statement which will select only customers from the database that have the last name provided. If the last name is blank, it will prepare a SQL statement that will select all customers from the database. Then the statement is executed, and all results fetched, and the results are printed out in the format ‘First Last’.

**2.2**

This code will display a popup box of all people’s professions stored in a data file on the server.

This adds an event listener on all elements with the id of ‘trigger’, so that when it is clicked, it runs an AJAX request to fetch a file called ‘people.json’, which is stored on the server. Once the file is received on the client successfully, the code iterates through the ‘people’ object of the returned JSON, calling alert to print the properties of name and profession for each person. There is an error in the code, there should be an extra closing curly brace after the alert statement.