Exam Part 1

1. Describe, in detail, how a server side web application using sessions determines which session (and associated data) belongs to which client. Remember that the server will have many connected clients, so there needs to be a way to determine this.
2. First we need to create a session. When the client (the web browser) connects to the server, it sends HTTP request. The serve will receive the request and initialize the session for the client if it does not exist. The sessions are manages using unique identifiers known as session IDs.
3. Then we generate a Session ID. When the client receives the first request the server will generate a unique session ID for this specific client. The ID is usually long and is alphanumeric. The session ID is crucial because it allows the serve to distinguish between different client sessions.
4. Sens the Session ID to the Client. Server will send back the session ID to the client by either cookies or URL rewriting. Cookies is the most common. With each subsequence request the client will automatically send the cookie back to the server. The URL rewriting is when cookies are disabled, the session ID will create these extra parameters in the URL such as: <http://example.com/page?sessionid=h2kc93f4lg85fdl2>

However, this is less common because of security issues.

1. Session data storage is found where the server maintains the data structure in the form of hash table or associative arrays. The Session ID will act as the key in the data structure.
2. Anytime

Pph interpretor will interpreate what files to execute then process the request and when it is finished generating the html, json, css etc to send to client the php process till forward it back to we server and back to initial client who made the request.

The web server will be listening on port 80 for example. Lsof -i:80 you can see which ports are listening on port 80 or if you go to port 443 for https you can see what is listening on that port.

1. Describe the cause of SQL injection vulnerabilities. Include a small pseudocode example of what a vulnerability in code might look like.
2. Describe some of the benefits of building server side applications as web APIs. Think about the flexibility that we gain by not sending back HTML to clients.
3. You’ve just started as a SWE at New Co. Your team lead is starting you on a new project for a blog feature. The blog feature will allow users toc create, read (a single blog, and get all of the blog posts), update and delete blog individual posts.

You’re responsible for designing the blog API endpoints for these actions. Design these endpoints. Think about the endpoints in your api.php file from lab 6. Describe the endpoints in the following format

**<HTTP request method> /path/ ← description**

For example, one of the endpoints in lab 6 was:

GET /articles ← return all articles

Pay special attention to the HTTP request methods chosen.

1. Describe what Cross-Site Scripting (XSS) is, and how to prevent it.
2. Describe some issues with the following code for a simple registration page. You may assume that the **saveUser** function parameters being passed in are correct, and that the function is not vulnerable to SQL injection.



Describe the issues below.

1. What data does HTTPS encrypt in transit? What data is not encrypted?
2. Describe what a digital certificate is and why it is used in TLS.