Name: Gonzales, Ethan James G. Section: WD-302

1. Discuss on what is REST

REST or Representational State Transfer is an architectural style that is commonly used to build web computer systems, web-based application programming interfaces (API's), or other web services that are capable of easy communication with each other. It relies on HTTP methods or verbs such as GET, PUT, POST, and DELETE to interact with different resources that are identified by URL's. In this architectural style, there is a separation of clients and servers. Clients can send requests to a server, and the server can send a response back. However, servers cannot send a request to clients and clients cannot respond, which means that all of the interactions are only initiated by a client. Furthermore, systems adhering to REST are stateless, meaning the server does not need to be aware of the client's state, and the client does not need to know the server's state. The server does not retain any information about previous interactions. These features are only a few of what makes RESTful API's popular as they are simple, scalable, and can be easily used with web technologies.

2. Explain on what is stateless (cite a scenario)

In REST, statelessness means that every client request to the server must contain all the necessary details for the server to process it, since the server does not keep track of any prior interactions. The server will treat each request made by the user or client as new and will not need previous history or sessions. An example scenario that applies statelessness is when booking for a flight. When looking for available flights, the server will process the request based on the details that you have provided without putting into consideration any past bookings, searches, or requests. Each request is handled independently, and the server does not rely on a user's past data.

3. Enumerate the 3 forms of authentication that can be used in an app and be able to differentiate them with one another.

- Basic Authentication
    - This form of authenticates a client or a request using a username and a password. Once the client enters their credentials, the server will decode it and will verify if it is valid or not based on the stored information of the client. This is easy to implement but is less secure as login credentials can be easily intercepted by others, especially if the site is not secure.

- Session Authentication
    - In this form of authentication, a session is created in a server once a user has logged in. Then, the server will send back a session ID which will be used by the user for each request that they make. The server will use the

session ID to authenticate the request and to verify the user. Once the user logs out, the session will be stopped or killed from the server.

- Token-based Authentication
    - This involves issuing a token when a user logs in successfully with a username and a password. The token will be included in the header of the requests which will be verified by the server for each request that the user makes to authenticate the user.

Sources:

https://www.codecademy.com/article/what-is-rest

https://medium.com/@alyragab70/authentication-types-and-techniques-eb8232eebcfb