**Web ?**

1. Techgatha => 3-tier architecture
2. Use HTTP protocol to communicate over the web
3. Browsers understand HTML/ CSS/ Js only => static websites [ banners on the streets ]
4. Javascript has limitations as it cannot communicate with the database directly **if** used for client side scripting.
5. To create dynamic websites [ showing dynamic data based on users request or who is using that website ]
6. Dynamic data => always involves database based on who is logged in / based on what user is searching
7. Browser[ Client] sends request to the server and if the request goes to a java application then need a java compliant server
8. There are few terminologies with respect to HTTP
   1. HTTP status codes : 1xx, 2xx, 3xx, 4xx, 5xx
   2. HTTP methods : GET, POST, PUT , DELETE , PATCH, HEAD, TRACE
   3. HTTP headers: Accept/ Content-Type/ Cookies

**Servlet**

1. Is a technology to communicate over HTTP protocol and process the request and send appropriate response
2. Uses java as a programming language
3. To create a servlet, the class should extend HttpServlet class
4. Servlets provide methods for respective HTTP methods doGet(), doPost()
5. doGet(), doPost() methods take 2 parameters:
   1. HttpServletRequest => responsible to process the request
   2. HttpServletResponse -> responsible to send content as a response that can be HTML / TEXT/ PDF/ JSON/ XML/ Images / Video/ Audio…
6. PrintWriter => to send HTML / TEXT content
7. GET request can be made via URL refresh, anchor links, form submissions or javascript AJAX
8. POST request can be made via form submissions or javascript AJAX
9. GET => used for fetching data. Less secured and has limitation on the amount of data that can be appended as part of the url
10. POST => used for form submission. Preferred as it secures data by not exposing over the URL. No limitations on the amount of data that can be sent
11. To process the HTTP request and delegate to respective Servlet, a server is needed
12. Tomcat / JBOSS/ Glassfish/ WebLogic etc server to process the HTTP request