**Task#1**

**Step 2.a**

**HTTP:** Is a built-in module in Node.js that is a fundamental component for building web applications. It allows developers to create HTTP servers to handle incoming web requests, making it possible to build custom web applications, REST APIs, and more. You can use the **createServer()** method to set response headers and send data back to clients. By specifying a port and hostname, the server can listen for incoming connections. The module also offers support for handling events like **request** and **listening**. Additionally, it enables making HTTP requests to external resources. While **http** deals with regular HTTP, Node.js provides an **https** module for secure communication.

**Globals:** Is anobject in Node.js that is a vital component that offers access to global variables, essential built-in objects, and file-related information without the need for explicit imports. It includes \_\_dirname and \_\_filename for retrieving directory and file paths. Developers can also create custom global variables by assigning values to the global object, allowing data sharing across modules. It is advisable to use custom global variables judiciously to avoid naming conflicts and maintain code clarity.

**Process:** Is anobject in Node.js that is a versatile and crucial component that serves as a bridge between your Node.js application and the underlying operating system. It offers many functionalities, including providing information about the current process, accessing environment variables, managing standard I/O streams, controlling the event loop, handling signals and events, and much more. It allows developers to interact with the runtime environment, access system information, and control program execution. Whether it's managing process exit codes, responding to user input, or monitoring memory usage, the process object plays a central role in Node.js applications.