1. The differences in functionality and purposes Client-Side and Server-Side code serves in a full-stack web application.

|  |  |
| --- | --- |
| Client-side JavaScript | Server-side JavaScript |
| * Runs in a browser process. * Renders the page. * Handles user interactions. * Handles user interactions. * Has window and document objects. * Interfaces with HTML and the DOM. * Uses AJAX to make HTTP requests. * Uses script tags to split code into files. | * Runs in a Node.js process. * Does not render anything. * Handles HTTP requests. * Sends responses. * Has no window or document objects. * Does not interface with HTML or DOM; no selectors, etc. * Can make direct HTTP requests. * Uses modules to split code into files. |

1. What explains Nodes' rise in popularity and use? What does "Isomorphic JavaScript programming" mean? Provide some real-world examples not listed in this checkpoint of companies using Node.js.

* Node.js is a powerful server-side JavaScript platform for building "fast, scalable network applications." Node.js is a JavaScript runtime built on Chrome's V8 JavaSCript engine. Node.js is widely used to develop a variety of programs including web applications, IoT applications, cloud-native apps, databases focused apps, message systems, and more.
* "Isomorphic JavaScript programming" means JavaScript that can run both on the client-side and the server-side.
* Company using Node.js: Walmart, Trello, Groupon, Ebay, etc.

1. Draw a diagram of a full-stack web application and its key components.

Client Machine

Browser

React

Application

Back-End Server

Node.js

Express

MongoDB Driver

A

P

P

A

P

I

MongoDB