****

**MANAV RACHNA UNIVERSITY**

**COURSE: BTECH CSE**

**SEMESTER – 3**

**User Interface II**

**Khushi Arora**

**CSE 3C**

**2K20CSUN01058**

**CASE STUDY**

**SUBMITTED TO: Mrs. Deepanshi**

**Q1 What is NodeJS?**

Node. js is a JavaScript runtime environment that achieves low latency and high throughput by taking a “non-blocking” approach to serving requests. In other words, Node. js wastes no time or resources on waiting for I/O requests to return.

**Q2 Why do we use NodeJS?**

Node. js brings event-driven programming to web servers, enabling development of fast web servers in JavaScript. Developers can create scalable servers without using threading, by using a simplified model of event-driven programming that uses callbacks to signal the completion of a task

**Q3 What are the properties of NodeJS?**

|  |  |
| --- | --- |
| **Property** | **Description** |
| attributes | A NamedNodeMap containing the attributes of this node (if it is an Element) |
| [baseURI](https://www.w3schools.com/xml/prop_node_baseuri.asp) | Returns the absolute base URI of a node |
| [childNodes](https://www.w3schools.com/xml/prop_node_childnodes.asp) | Returns a NodeList of child nodes for a node |
| [firstChild](https://www.w3schools.com/xml/prop_node_firstchild.asp) | Returns the first child of a node |
| [lastChild](https://www.w3schools.com/xml/prop_node_lastchild.asp) | Returns the last child of a node |
| [nextSibling](https://www.w3schools.com/xml/prop_node_nextsibling.asp) | Returns the node immediately following a node |
| [nodeName](https://www.w3schools.com/xml/prop_node_nodename.asp) | Returns the name of a node, depending on its type |
| [nodeType](https://www.w3schools.com/xml/prop_node_nodetype.asp) | Returns the type of a node |
| [nodeValue](https://www.w3schools.com/xml/prop_node_nodevalue.asp) | Sets or returns the value of a node, depending on its type |
| [ownerDocument](https://www.w3schools.com/xml/prop_node_ownerdocument.asp) | Returns the root element (document object) for a node |
| [parentNode](https://www.w3schools.com/xml/prop_node_parentnode.asp) | Returns the parent node of a node |
| [prefix](https://www.w3schools.com/xml/prop_node_prefix.asp) | Sets or returns the namespace prefix of a node |
| [previousSibling](https://www.w3schools.com/xml/prop_node_previoussibling.asp) | Returns the node immediately before a node |
| [textContent](https://www.w3schools.com/xml/prop_node_textcontent.asp) | Sets or returns the textual content of a node and its descendants |

**Q4 What companies uses Node.js ?**

Some companies that use Node.js include the following:

1. LinkedIn
2. Netflix
3. Uber
4. Trello
5. PayPal
6. NASA
7. eBay
8. Medium
9. Groupon
10. Walmart
11. Mozilla
12. GoDaddy

**Q5 What services of Node.js are these companies using?**

1. **LinkedIn -** server service
2. **Netflix** - user interface (UI) of Netflix was built using Node.js
3. **Uber -** error-checking; fast processing speeds; and open-source community
4. **Trello -** multiple open connections
5. **PayPal -** user-facing content
6. **NASA -** application development
7. **eBay -** live connections to servers
8. **Medium -** app servers
9. **Groupon -** framework services
10. **Walmart** - framework for its UIs and APIs creation
11. **Mozilla -** web apps creation
12. **GoDaddy** - back-end services

**Q6 What are the requirements of these companies from Node.js ?**

1. **LinkedIn**- LinkedIn switched from Ruby on Rails to Node.js in late 2011. This was mainly because Linkedin developers wanted to optimize their mobile app where now everything is written entirely in Node.js. They were able to drastically cut down on server resources and the app is said to run 20 times faster than its predecessor.
2. **Netflix-** The user interface (UI) of Netflix was built using Node.js. According to the Netflix team, the modularity of the framework encouraged them to use Node.js.
3. **Uber-** There are three main reasons why Uber chose Node.js: ease of error-checking; fast processing speeds; and continuous development thanks to the open-source community.
4. **Trello-** It is important for Trello and its users to have immediate updates and Node.js has the capacity to hold many open connections at once.
5. **PayPal**- PayPal uses Node.js to build user-facing content.
6. **NASA-** Microservices is a feature of Node.js where developers approach application development by presenting the software as a collection of small services rather than as an autonomous unit. Using this feature NASA was able to move its enterprise to the cloud.
7. **eBay-** Node.js helps eBay by being able to maintain live connections to servers for nearly 200 million eBay users every day

1. **Medium-** Medium developers build their app servers with Node.js accompanied by the webserver NGINX.
2. **Groupon-** Groupon software developers migrated both their mobile and web traffic to the framework. Higher scalability was their main justification but this switch resulted in resource reduction for high traffic and faster web pages.
3. **Walmart-** Walmart relies on the framework for its UIs and also as an orchestration layer to create application programming interfaces (APIs) for the company’s various apps.
4. **Mozilla-** Many of its web apps use Node.js because of its memory capacity and for the ease of use involved in being able to keep everything in a single JavaScript repository.
5. **GoDaddy-** Their back-end infrastructure is dependent on Node.js. Node.js is practical for testing, NPM resources, and REST.

**Q7 what are call back functions?**

A callback is a function called at the completion of a given task; this prevents any blocking, and allows other code to be run in the meantime.

**Q8** **What is synchronous and asynchronous IOS?**

**Synchronous:** waits until the task has completed

**Asynchronous:** completes a task in background and can notify you when complete.

**Q9 What are blocking and non-blocking IOS?**

**Blocking** is when the execution of additional JavaScript in the Node.js process must wait until a non-JavaScript operation completes. This happens because the event loop is unable to continue running JavaScript while a blocking operation is occurring.

All of the I/O methods in the Node.js standard library provide asynchronous versions, which are **non-blocking**, and accept callback functions. Some methods also have blocking counterparts, which have names that end with Sync.