**Problem 1: Why do we say Node uses an Event-Driven-Architecture?**

Node uses an event driven architecture. When an event like database operation, file operation, other I/O operations occur, Node uses callbacks/promises to handle these events. When an event occurs, it stores that event callback into event loop(queue) and it gets executed by V8 when Node reaches it.

Callback functions can help to make Node code non-blocking.

**Problem 2: Can you write Node code that is not event driven? Why?**

Yes, we can write code in Node that is not event driven. We can write some code to be executed by Node when no events occur. Like, we can write code to connect to the database at the very beginning of the app.js(entry js file) which gets executed when no event occurs. There can be some business logic or some initial setup code for the project that has to be executed at the very beginning during setup, then it can be executed without any events.

So, Node supports both event driven and non event driven architecture.

**Problem 3: Why does Chrome use a layout tree?**

Chrome uses layout tree to store information related to what's visible on the page. If display: none is applied, that element is not part of the layout tree, however, an element with visibility: hidden is in the layout tree. Similarly, if a pseudo class with content like p::before{content:"Helloo!"} is applied, it is included in the layout tree even though that is not in the DOM tree. It is used to display visible html elements at different places in the browser.