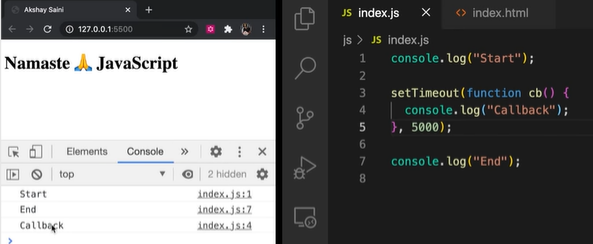
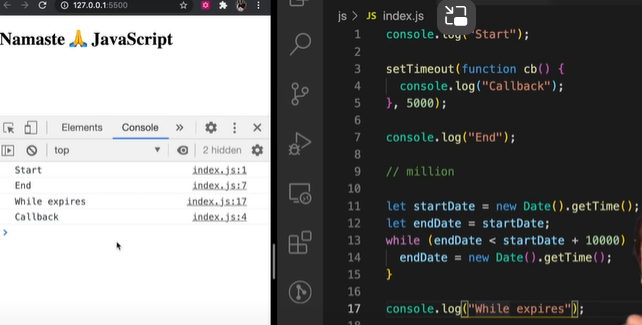


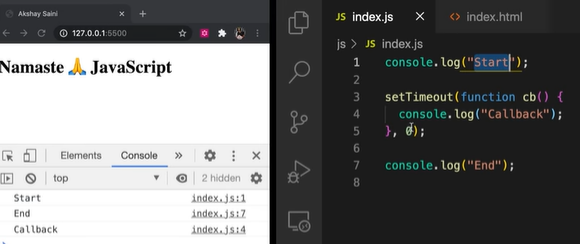
If call stack is not empty, if program is so big or time taking, then after timer expire then it continuously waiting inside callback queue.

Practical :-



As while loop expires after 10000 ms,

setTimeout executed.

/////////////////////////

As setTimeout is of 0 ms, It has to pass throw the whole process and wait

In the callback queue for completion of rest of the tasks.

* **setTimeout(fn, 0)**: Schedules the function **fn** to be executed after the current code has finished, allowing other tasks in the event loop to be processed before the function runs.
* Directly calling a function: Executes the function immediately in the current execution context without waiting for any asynchronous operations or event loop processing.





Things learned:

1. The setTimeout function stores it in the callback queue which is executed only after call stack is empty, even if setTimeout is set to 0ms.

2. setTimeout ensures that minimum it will take the time mentioned because it may be paused due to call stack not empty.