Callbacks in JavaScript

DIY - 25

edureka!

Problem Statement:

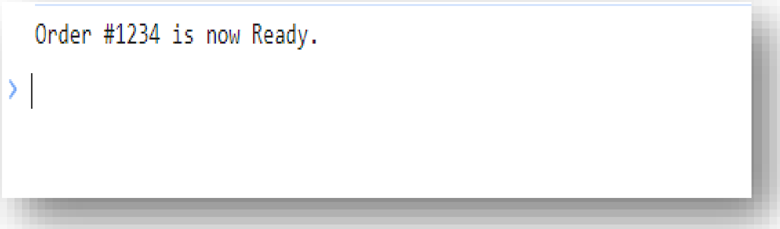
You are developing a food ordering application. Your task is to implement a function **placeOrder** that takes the order details and a callback function as arguments. The **placeOrder** function should simulate placing an order to a restaurant and then, after a delay (to simulate waiting time for order preparation), it should execute the callback function to notify the user that their order is ready for pickup or delivery.

Objective:

1. The **placeOrder** function should accept two parameters:
   - **orderDetails** (an object containing at least **orderId** and **items**).
   - **callback** (a function that will be called after the order is "prepared").
2. Simulate the order preparation time using **setTimeout** with a delay of 3 seconds.
3. The callback function should accept two arguments:
   - An error object (null if no error occurred).
   - An order object that includes the original order details and a **status** property set to **"Ready"**.

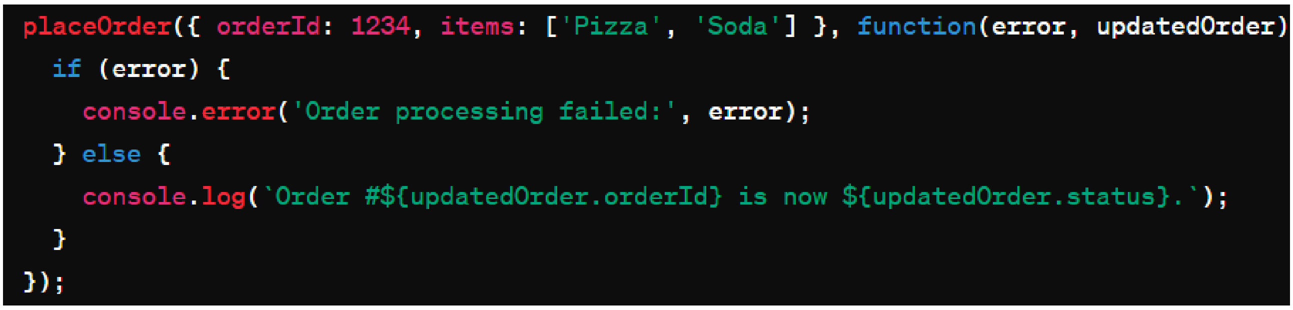
Write the placeOrder function according to the requirements above.

Expected output: After 3 Seconds



```
Order #1234 is now Ready.
> |
```

Hint:



```
placeOrder({ orderId: 1234, items: ['Pizza', 'Soda'] }, function(error, updatedOrder)
  if (error) {
    console.error('Order processing failed:', error);
  } else {
    console.log(`Order #${updatedOrder.orderId} is now ${updatedOrder.status}.`);
  }
});
```

Submission:

- Organize your code neatly and use meaningful variable names.
- Provide comments where necessary to explain your code.
- Include any necessary user prompts for input.
- Ensure your code runs without errors and produces the expected output.