The checkStockAndDeduct function in the provided code serves two purposes:

1. **Checking real-time stock for ordered items:**
   - It iterates through each item in the items array passed as an argument.
   - For each item, it uses the Product.findById method to find the corresponding product document in the database.
   - It checks if the product has enough stock available by comparing the quantity in the order item to the stockQuantity in the product document.
2. **Deducting purchased quantities from product stock:**
   - If the product has sufficient stock, the function updates the product document using the findByIdAndUpdate method.
   - It uses the $inc operator to decrease the stockQuantity by the ordered quantity.
   - This ensures that the product stock reflects the recent purchase.

Here is a breakdown of the code within the checkStockAndDeduct function:

JavaScript

```
async function checkStockAndDeduct(items, session) {
  try {
    for (const item of items) {
      const product = await
Product.findById(item.product).session(session);
      if (!product || product.stockQuantity < item.quantity) {
        throw new Error(`Insufficient stock for product ${product ?
product.name : 'Unknown'}`);
      }
      await Product.findByIdAndUpdate(item.id, { $inc: { qantity:
-item.quantity } }).session(session);
    }
  } finally {
    session.endSession();
  }
}
```

Key points to remember:

- The function uses a Mongoose session for transaction management. This ensures that all operations within the loop are treated as a single unit and either succeed or fail together.
- The try-catch block ensures that any errors during stock checks or updates are handled appropriately.
- The finally block ensures that the Mongoose session is always closed, regardless of

whether there were errors or not.

This function plays a crucial role in maintaining data consistency and preventing overselling of products in your e-commerce application. By ensuring sufficient stock availability before processing orders, you can provide a better customer experience and avoid order cancellations or delays.