Write a Node.js endpoint using Express that interfaces with a MongoDB database to perform a CRUD operation. Include a brief explanation of how you would handle the asynchronous nature of Node.js to ensure that responses are sent only after database operations are completed.

```
const express = require('express'); const mongoose = require('mongoose'); const bodyParser
= require('body-parser'); const Item = require('./models/Item'); // Assuming you have a
Mongoose model for items

const app = express(); const PORT =
process.env.PORT || 3000;
app.use(bodyParser.json());

app.post('/items', async (req, res) => {
try {
      // Extract item data from the request body
const { name, description } = req.body;

      const newItem = new Item({ name, description });
await newItem.save();        res.status(201).json(newItem);
   } catch (error) {        console.error('Error creating
item:', error);        res.status(500).json({ error: 'Error
creating item' });
   }
});
mongoose.connect('mongodb://localhost:27017/your_database', {
useNewUrlParser: true,     useUnifiedTopology: true
```

```
}).then(() => {     console.log('Connected to MongoDB');
app.listen(PORT, () => {         console.log(`Server is running on
http://localhost:${PORT}`);
   });
}).catch((error) => {     console.error('Error connecting to
MongoDB:', error);
});
```