The issue at hand is Cross-Site Scripting (XSS) on the product detail page. Through the section where users can write comments, the page becomes vulnerable to DOM-based XSS.

**Demonstration of the corrected vulnerability:**

Here is the product page of the insecure application with a comment about to be submitted by an attacker.

Uma imagem com texto, captura de ecrã, Tipo de letra, design

Descrição gerada automaticamente

Uma imagem com texto, captura de ecrã, software, Software de multimédia

Descrição gerada automaticamente

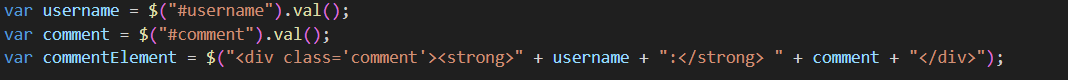
And here we see the result of the attack because the input for comments was not properly sanitized.

After modifying the JavaScript code, we can now see that the same no longer occurs in the secure application.Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente

**Comparison of the corrected code:**

Insecure code:



Despite the difference being small, we can now observe that the input for comments is being properly sanitized:

Uma imagem com captura de ecrã, texto, file

Descrição gerada automaticamente