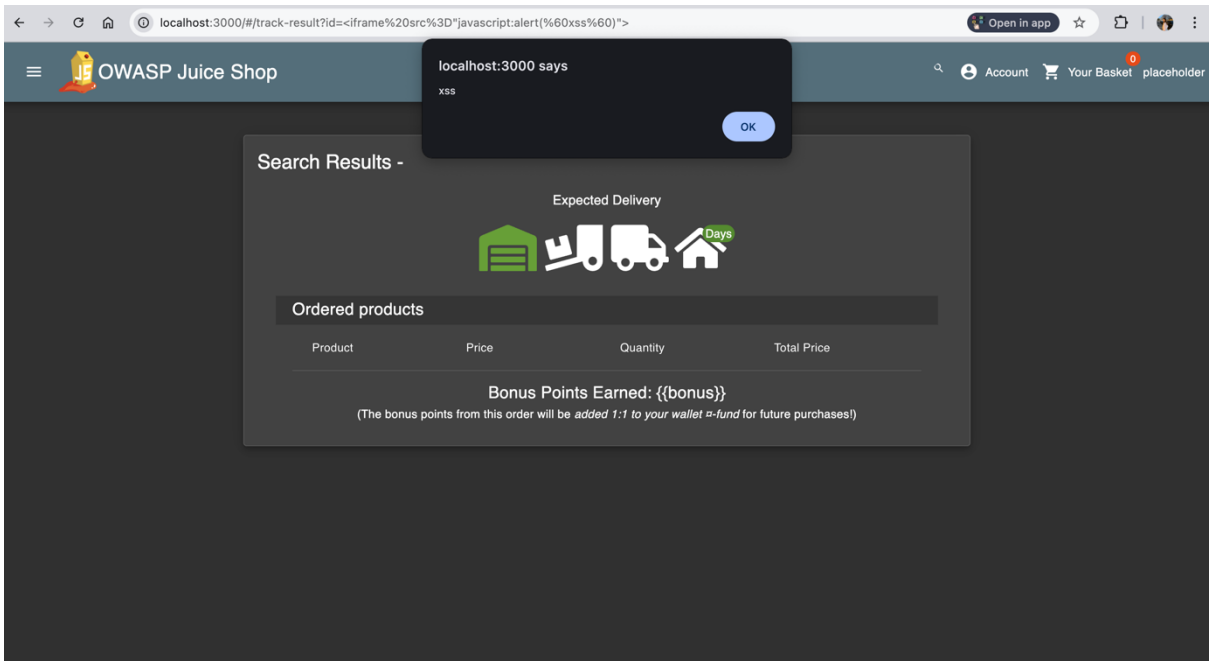
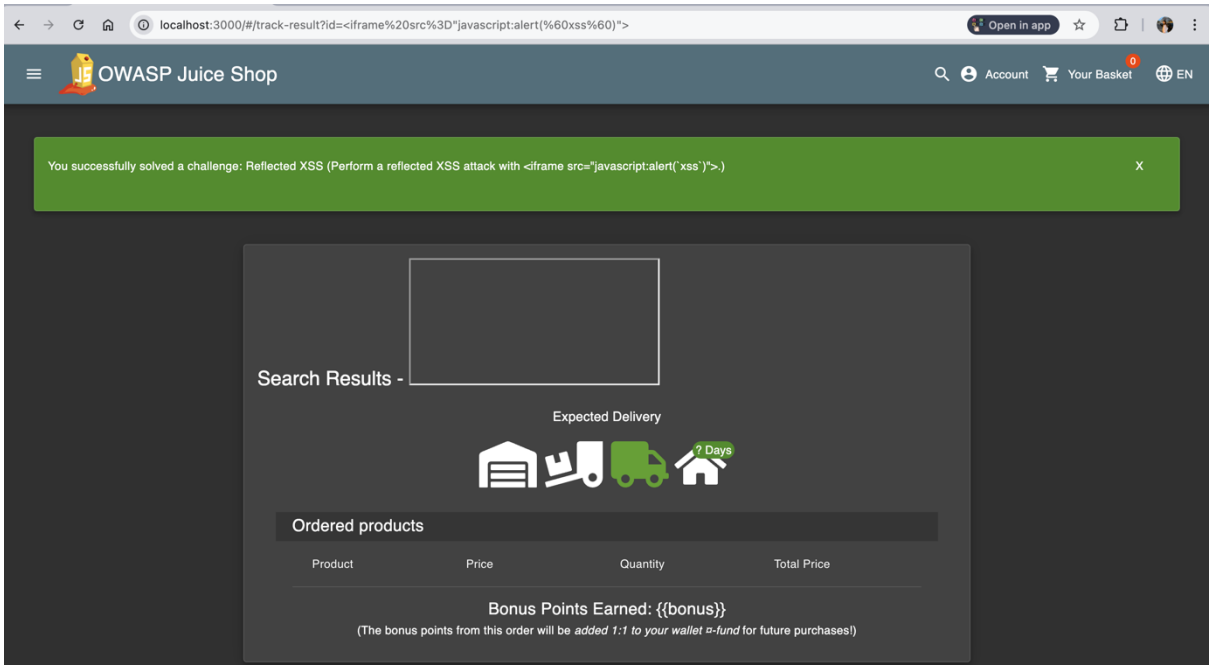
XSS attack

http://localhost:3000/#/track-result?id=%3Ciframe%20src%3D%22javascript:alert(%60xss%60)%22%3E

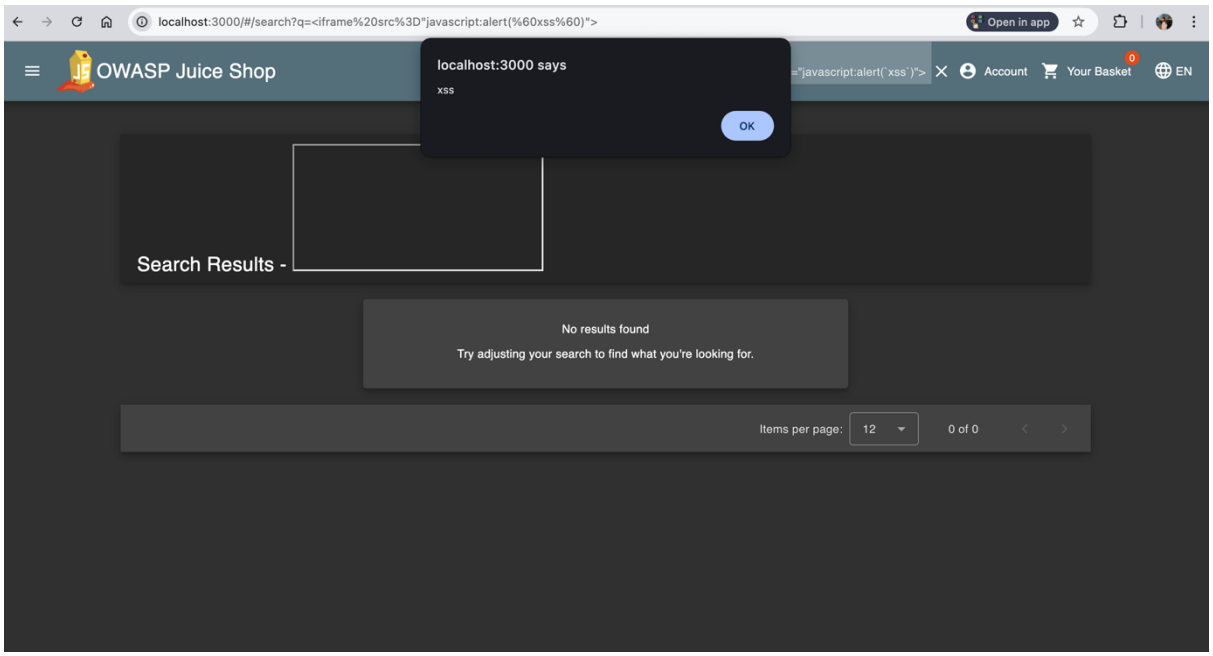


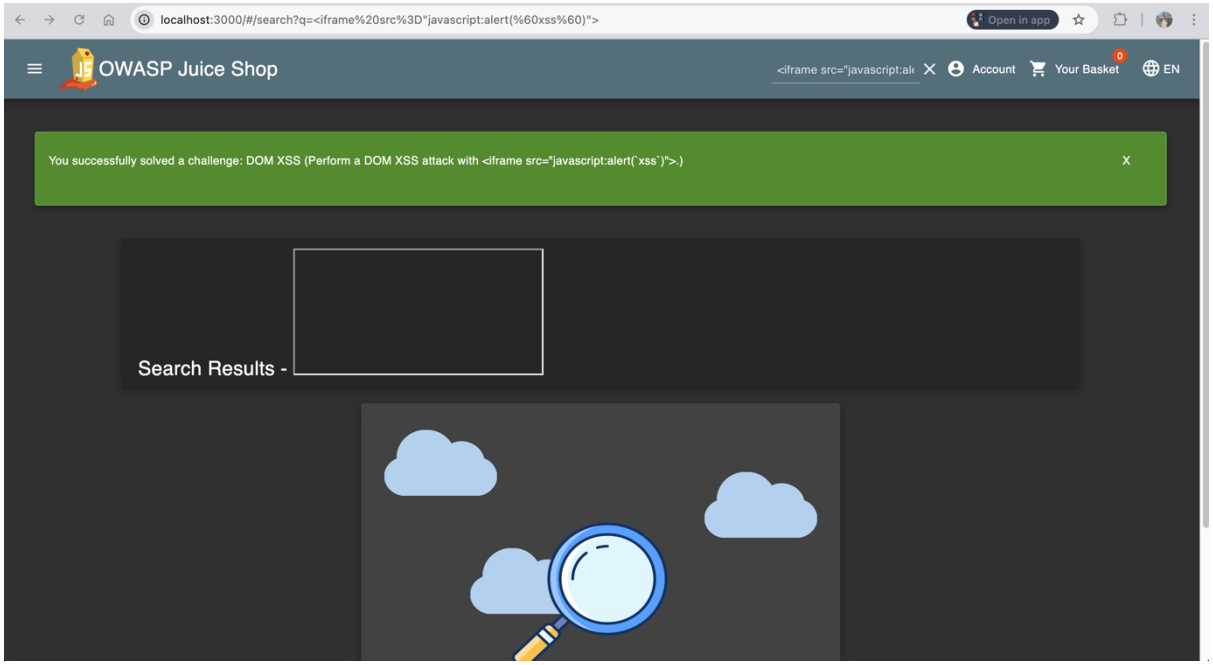


Először beléptem a Juice Shop felületére, majd a Order History oldalon rákattintottam a teherautó ikonra, ami átirányított egy URL-re, amely tartalmazott egy id paramétert. Ezután kipróbáltam egy iframe-s XSS payloadot, amit közvetlenül az URL-ben adtam meg. Miután betöltöttem ezt az URL-t, megjelent az alert("XSS") felugró ablak, ami azt jelezte, hogy a bemeneti adatot a JavaScript kód szűrés nélkül építi be az oldalba.

DOM XSS attack

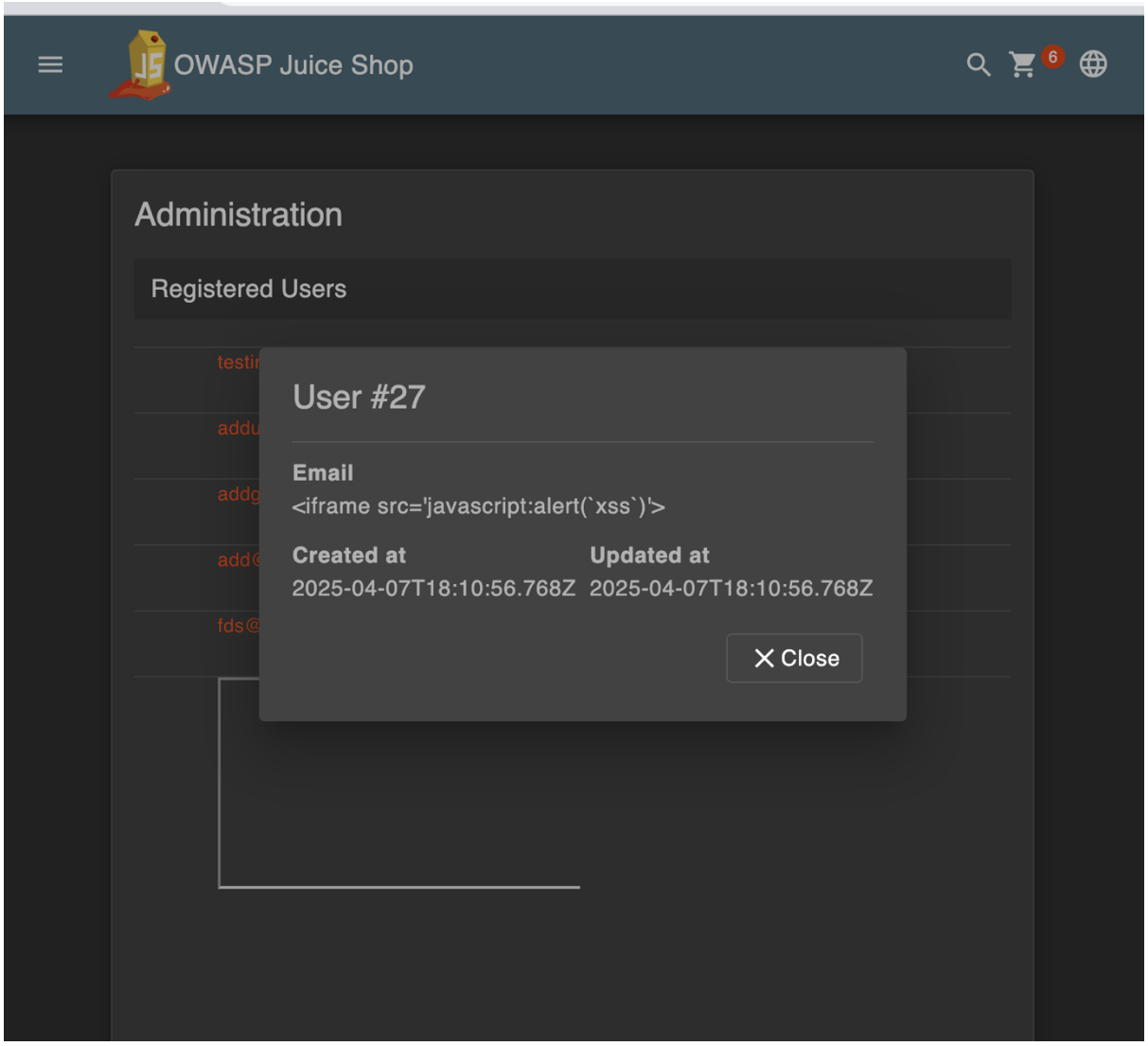
A keresőben - <iframe src="javascript:alert(`xss`)">





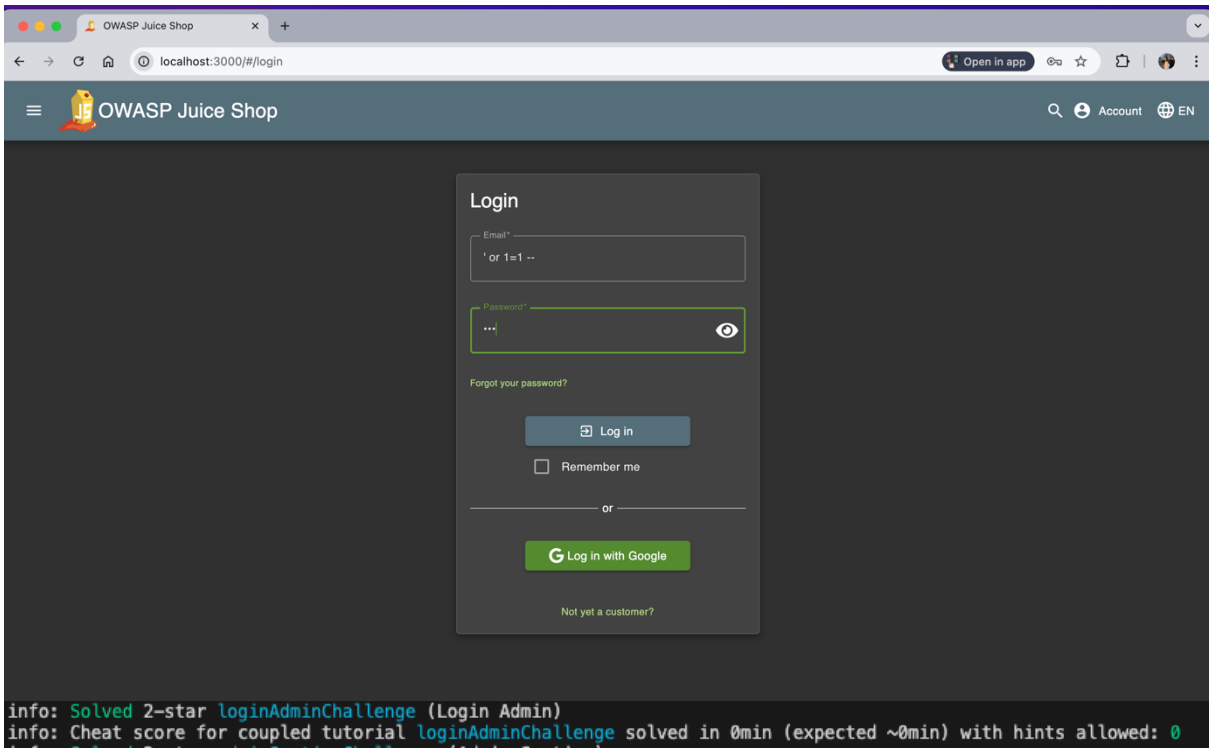
Beírtam a keresőbe egy XSS payloadot, majd megfigyeltem a Juice Shop githubján a search-result.component.html fájlt ahol találtam egy olyan részt, ahol a keresési kulcsszó közvetlenül jelenik meg az oldalon az innerHtml használatával. A JavaScript a keresett szöveget HTML-ként helyezte be az oldalba => DOM XSS sebezhetőség.

Perform a persisted XSS attack bypassing a client side security mechanism!



Csináltam egy új felhasználót, majd a requestet postmanból újraküldtem ú.h. az email címet kicseréltem a következőre: <iframe src=’javascript:alert(`xss`)’> .

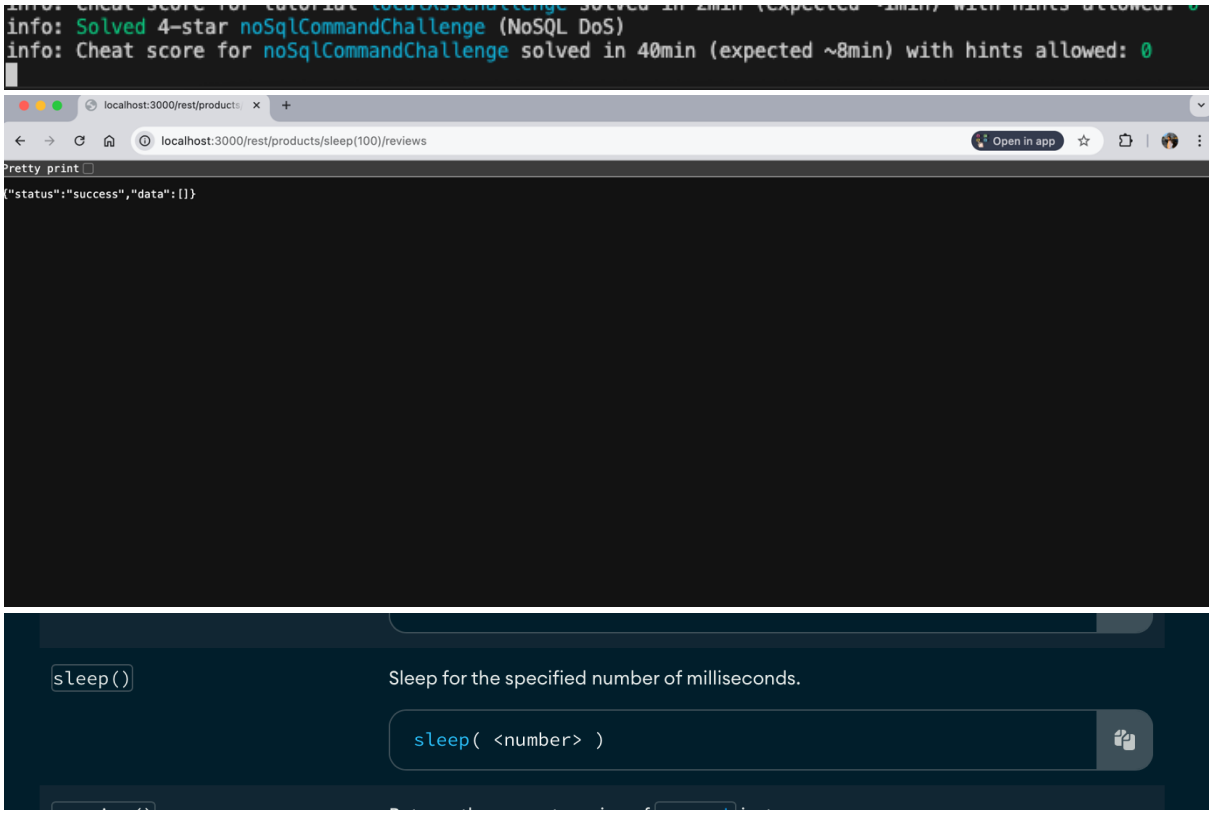
Log in with the administrator's user account!



SELECT * FROM Users WHERE email = '' or 1=1 -- AND password = '${security.hash(req.body.password || '')}' AND deletedAt IS NULL

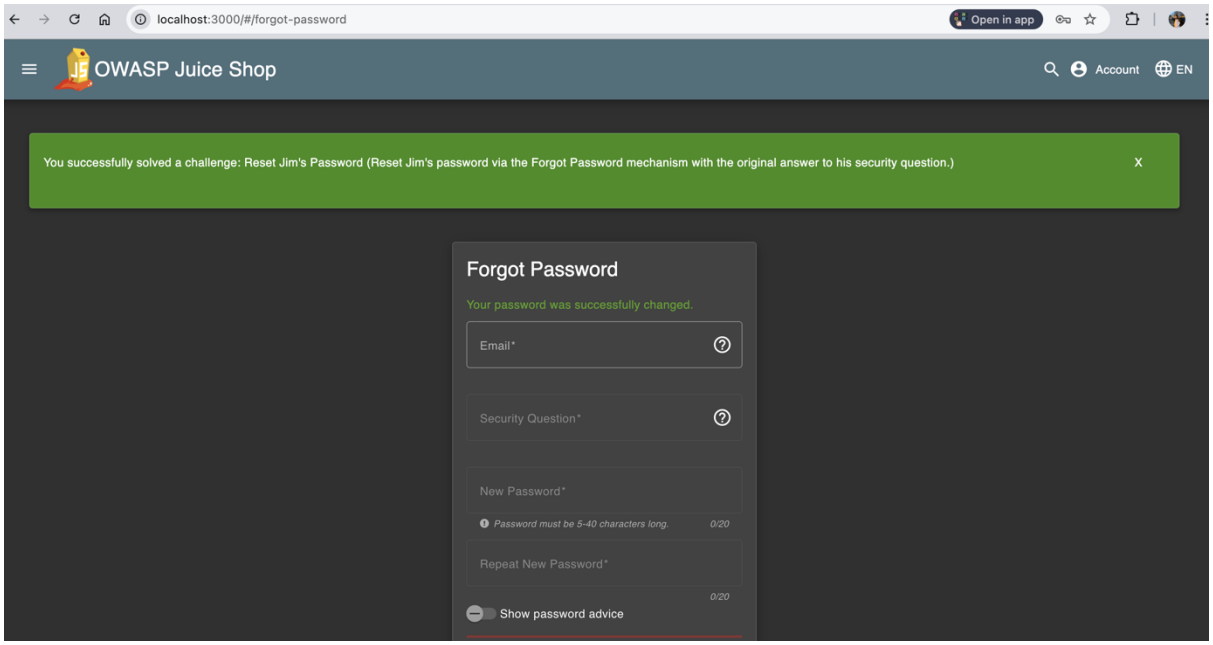
Így az sql command többi része ignorálva lesz

Let the server sleep for some time!



Megnéztem a mongo db oldalát illetve a server side codeot és az urlb-be beírva a sleep(100)-at.

Reset Jim's password via the Forgot Password mechanism.



Jim testvérének a neve Samuel volt az interneten szerzett információk alapján.