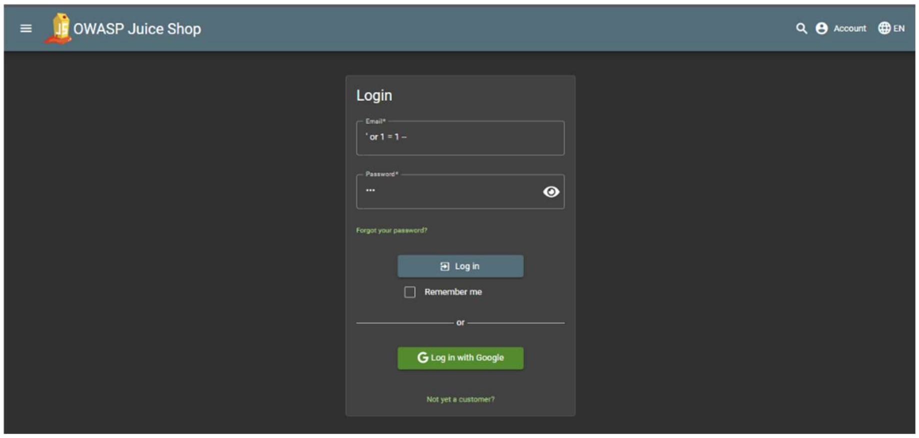
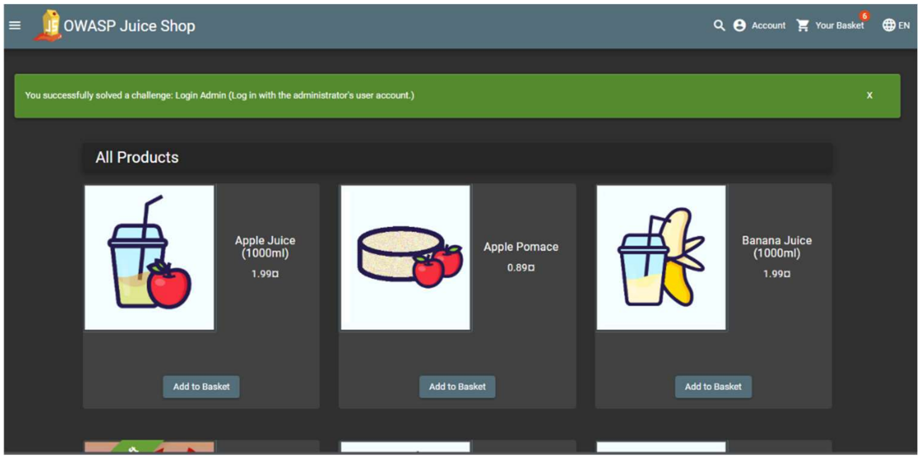
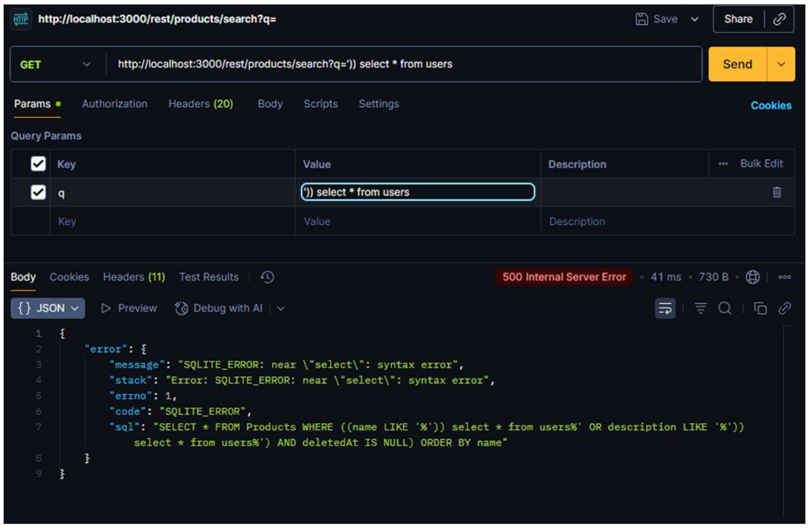
	Haryo Arief Wicaksono
1.	Ditemukan celah keamanan saat log in dengan memasukkan script SQL
	Bukti Screenshot:  OWASP Juice Shop Account EN Login Forgot your password? Log in Remember me or Log in with Google Not yet a customer? Email* ' or 1 = 1 – Password* ••• □ Remember me  OWASP Juice Shop Account Your Basket 6 EN X All Products Apple Juice (1000ml) 1.99¤ Apple Pomace 0.89¤ Banana Juice (1000ml) 1.99¤ Add to Basket Add to Basket Add to Basket You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)
	Root Cause Analysis: Penyebabnya diduga karena terdapat sql injection di dalam kodingan. Recommendation: Gunakan bind parameter atau replacement saat melakukan query ke dalam database. Verification scenario: Tidak akan lagi bisa melakukan SQL injection. Dan login jadi lebih terjaga.

2.	Ditemukan SQL Injection di API Kolom Search. Struktur database jadi terlihat kalau error. Dan dapat dieksploitasi untuk mendapatkan data yang seharusnya tidak dipublikasikan.

	Bukti Screenshot:
	
	http://localhost:3000/rest/products/search?q=
	GET http://localhost:3000/rest/products/search?q=')) select * from users
	Query Params
	☑ Key: q — Value: ')) select * from users
	500 Internal Server Error · 41 ms · 730 B
	{
	    "error": {
	        "message": "SQLITE_ERROR: near \"select\": syntax error",
	        "stack": "Error: SQLITE_ERROR: near \"select\": syntax error",
	        "errno": 1,
	        "code": "SQLITE_ERROR",
	        "sql": "SELECT * FROM Products WHERE ((name LIKE '%')) select * from users%' OR description LIKE '%')) select * from users%') AND deletedAt IS NULL) ORDER BY name"
	    }
	}

	Root Cause Analysis: Penyebabnya diduga karena terdapat sql injection di dalam kodingan. Recommendation: Gunakan bind parameter atau replacement saat melakukan query ke dalam database. Verification Scenario: Tidak akan bisa melakukan SQL Injection lagi. Dan keamanan database lebih terjaga.