## Applying Threat Modeling to "ShopEasy"

In this step, you'll apply the STRIDE framework to identify potential threats in "ShopEasy."

### Instructions

1. **Identify Assets**:
   * User data (e.g., usernames, passwords)
   * Payment information (e.g., credit card details)
   * Product database
2. **Brainstorm Threats Using STRIDE**:
   * **Spoofing**: Could an attacker impersonate a user to make unauthorized purchases?
   * **Tampering**: Could an attacker modify product prices or order details?
   * **Repudiation**: Could a user deny making a purchase?
   * **Information Disclosure**: Could sensitive data (e.g., credit card info) be exposed?
   * **Denial of Service**: Could the application be overwhelmed to make it unavailable?
   * **Elevation of Privilege**: Could a user gain admin access?
3. **Document Threats**:
   * Create a list of potential threats for each STRIDE category, noting where they might occur in the application.

**Example**:

* **Threat**: SQL injection in the login form (Tampering).
* **Vulnerability**: Unsanitized user inputs in database queries.

**Why It Matters**: This exercise helps you systematically uncover security risks, thinking critically about the application's weaknesses.

### Security Analysis:

#### **Spoofing:**

#### **Tampering:**

#### **Repudiation:**

#### **Information Disclosure:**

#### **Denial of Service:**

#### **Elevation of Privilege:**

### Recommendations: