PASTA Threat Modeling Worksheet: Sneaker Shopping App

# Stage I – Business Objectives

* Secure user experience: Ensure data privacy and protect sensitive user information (e.g., personal details, payment methods).
* Smooth and fast transactions: Enable fast checkout with multiple payment options while maintaining security.
* Community trust: Build a reliable platform through features like user ratings and direct messaging, ensuring communication remains secure.

# Stage II – Technology Requirements

Prioritized Technology: Application Programming Interface (API)  
APIs are a top priority because they are the primary gateway between the mobile app and the backend services. If not properly secured, they could expose sensitive endpoints, enable unauthorized data access, or allow injection attacks. Many modern breaches exploit API vulnerabilities, making them a key focus in threat modeling.  
Other technologies involved:  
- Public Key Infrastructure (PKI) using AES for data encryption and RSA for secure key exchange.  
- SHA-256 for hashing sensitive data like passwords.  
- SQL for interacting with the product and user database.

# Stage III – Application Decomposition

The data flow includes users submitting login credentials, performing sneaker searches, initiating direct messages, and completing transactions. Each of these processes involves transferring data through the app’s API and storing/retrieving from the SQL database, using PKI and SHA-256 to secure those flows.

# Stage IV – Threat Identification

* SQL Injection Attacks targeting the app’s login or product search inputs.
* API Endpoint Exploitation where threat actors attempt unauthorized access to private or administrative data via unprotected or poorly documented API routes.

# Stage V – Vulnerability Identification

* Improper input validation in user forms (login, messaging, search) could allow injection attacks.
* Weak API authentication could expose data if API keys or tokens are not securely implemented or rotated regularly.

# Stage VI – Attack Tree Consideration

The attacker could:  
- Target the login form with SQL injection to bypass authentication.  
- Exploit API endpoints to extract customer data.  
- Intercept unencrypted messages if TLS isn’t enforced.  
- Abuse password reset functions through social engineering or broken logic.

# Stage VII – Risk Mitigation Controls

* Prepared Statements for SQL Queries – Prevent SQL injection by using parameterized inputs.
* API Rate Limiting and Authentication – Use OAuth2 or token-based auth, and limit requests per IP.
* Input Validation and Sanitization – Reject or sanitize unexpected user input at all entry points.
* Transport Layer Security (TLS) – Encrypt all communication between client and server using HTTPS.