PASTA worksheet

Stages	Sneaker company
I. Define business and security objectives	• Establish a trustworthy platform that optimizes interactions between buyers and sellers through secure and efficient channels. • Implement robust authentication mechanisms and identity management to ensure secure and seamless user access. • Ensure the confidentiality and integrity of personal data through strict privacy policies and security controls. • Foster a trusted digital environment by mitigating risks and threats that could compromise user security.
II. Define the technical scope	Public Key Infrastructure (PKI) is prioritized due to its critical role in securing online data exchange. The mobile app relies on a hybrid encryption model using AES for sensitive data and RSA for key exchange. Misconfigurations in PKI could expose users to data breaches, undermining trust in the platform's security controls.
III. Decompose application	Sample data flow diagram
IV. Threat analysis	**Internal Threats:** • **Misconfiguration by employees:** Improper configuration of encryption or access controls could expose sensitive user or inventory data. • **Social engineering attacks:** Employees tricked via phishing or impersonation could inadvertently disclose credentials or sensitive system access.

	External Threats: • **SQL Injection:** Exploiting SQL queries to gain unauthorized access to database contents, such as product listings or user data. • **Malware/Viruses:** Compromised user or admin devices could lead to system intrusion or data exfiltration. • **API Abuse / Broken Object Level Authorization:** Attackers manipulating API endpoints to access unauthorized resources or sensitive data. • **Man-in-the-Middle (MitM):** Interception of data in transit due to weak PKI or misconfigured TLS, compromising confidentiality and integrity. • **Hash Cracking (SHA-256):** If hashes are unsalted, attackers could brute-force credentials or sensitive data. • **Denial of Service (DoS):** Flooding the product search functionality to disrupt availability for legitimate users.
V. Vulnerability analysis	**Insecure API Endpoints (Codebase Vulnerability):** The application may lack proper authentication or input validation on API endpoints, making it susceptible to exploitation via broken access controls or injection attacks. **Unpatched Database Configuration (Database Vulnerability):** The database may be running with default settings or outdated software, exposing it to known exploits, such as privilege escalation or SQL injection.
VI. Attack modeling	Sample attack tree diagram
VII. Risk analysis and impact	**Multi-Factor Authentication (MFA)** Implement MFA for all user and employee accounts, especially for administrative access, to minimize the risk of unauthorized access due to compromised credentials (phishing or social engineering). **Strict API Validation and Access Control** Apply rigorous validation on API endpoints, including strong authentication, role-based access control (RBAC), and protection against code injection to prevent abuse and unauthorized access

	to sensitive resources. **Management and Monitoring of PKI and TLS Configuration** Conduct regular automated audits to ensure that the Public Key Infrastructure (PKI) and TLS configurations are correctly implemented and up to date, preventing Man-in-the-Middle (MitM) attacks and ensuring confidentiality and integrity of data in transit. **Continuous Patching and Updating of Software and Databases** Keep operating systems, databases, and applications updated with the latest patches to mitigate known vulnerabilities, including those that could allow SQL injection or privilege escalation.