Scenario

You're part of the growing security team at a company for sneaker enthusiasts and collectors. The business is preparing to launch a mobile app that makes it easy for their customers to buy and sell shoes.

You are performing a threat model of the application using the PASTA framework. You will go through each of the seven stages of the framework to identify security requirements for the new sneaker company app.

PASTA worksheet

Stages	Sneaker company
I. Define business and security objectives	The application should seamlessly connect sellers and shoppers. User sign-up and log in should be easy, and managing account preferences should be effortless. The app should be able to process financial transactions between shoppers and sellers and should abide by the Payment Card Industry Data Security Standard.
II. Define the technical scope	List of technologies used by the application: • *Application programming interface (API)* • *Public key infrastructure (PKI)* • *SHA-256* • *SQL* The API allows data exchange between sellers, shoppers, and employees running the app, connecting users and systems together for the app to function. There are several API's available, so that should be a primary consideration before addressing the fact that API's are prone to security vulnerabilities of having a large attack surface.

III. Decompose application	Sample data flow diagram
IV. Threat analysis	List **2 types of threats** in the PASTA worksheet that are risks to the information being handled by the application. SQL Injection, where the attacker might interfere with queries that the user or shopper makes when trying to access a database. Session hijacking, due to the fact that cookies might not be processed properly between multiple layers on input and output.
V. Vulnerability analysis	List **2 vulnerabilities** in the PASTA worksheet that could be exploited. Broken API token/session Hijacking Lack of prepared statements
VI. Attack modeling	Sample attack tree diagram
VII. Risk analysis and impact	List **4 security controls** that you've learned about that can reduce risk. Hashing using SHA256 Principle of least privilege Set procedures for incident response Password policy (MFA, OTP)