PASTA worksheet

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.
Description: Our application should seamlessly connect sellers and shoppers. It should be easy for users to sign-up, log in, and manage their accounts. Data privacy is a big concern for us. We want users to feel confident that we're being responsible with their information.
Buyers should be able to directly message sellers with questions. They should also have the ability to rate sellers to encourage good service. Sales should be clear and quick to process. Users should have several payment options for a smooth checkout process. Proper payment handling is really important because we want to avoid legal issues.

Stages	Sneaker company
I. Define business and security objectives	• The app will allow for user registration, log in and account management. The users will be able to contact the seller and rate their service. The app will be used to process transactions. • It will be an app back-end intensive since it will be using the database to search for products, sellers, private messages, storage of user information and processing transaction. • With this features the app will need to comply with PCI DSS, GDPR regulations and apply NIST Special Publication 800-53 to protect it's users' information.
II. Define the technical scope	List of technologies used by the application: • *API* • *PKI* • *AES* • *SHA-256* • *SQL* The API will be the work horse of this application, it will serve as a messenger between the user and the server to request and respond. The server will use SQL to communicate with the database and send and retrieve data. To encrypt user information it's recommended to use AES jounce with a PKI since it's an algorithm used to encrypt and decrypt data. Lastly SHA-256 to hash passwords to prevent password storage in plain text.
III. Decompose application	View DataFlowDiagram.png

IV. Threat analysis	Internal: • Unauthorized Access • Misconfiguration External: • SQL Injection • Cross-Site Scripting
V. Vulnerability analysis	The app could be vulnerable to command injection if not properly sanitized or suffer from weak authentication processes. Also if the db isn't encrypted attackers could read plain text PII. If the server isn't properly configured it could lead to network hopping, compromising other machines and leakage of more sensitive information.
VI. Attack modeling	View AttackTree.png
VII. Risk analysis and impact	To reduce risk in an application environment, implementing several security controls is crucial. Input validation and sanitization are essential to prevent attacks like SQL Injection and Cross-Site Scripting by ensuring malicious content is not processed or displayed. Multi-Factor Authentication (MFA) adds an extra layer of security, significantly decreasing unauthorized access by requiring additional verification beyond passwords. Regular security patching and updates are vital to close known vulnerabilities, reducing the window for exploitation. Lastly, encrypting data both at rest and in transit protects sensitive information from unauthorized access, ensuring confidentiality even if data is intercepted or databases are compromised. These controls, when combined, create a robust defense mechanism against various threats, enhancing the overall security posture of the application.