PASTA worksheet

Stages	Sneaker company
I. Define business and security objectives	• *The app should store, accept, process, and transmit credit card payments securely over the network.* • *PII data must be kept safe from unauthorized users* • *The app must meet compliance regulations, including PCI DSS & GDPR* • *Users can create profiles internally or by connecting third-party accounts.*
II. Define the technical scope	*Login, sign-up, and payment processing forms are stored in SQL databases, making them prone to SQL injection attacks. The code should require sanitized inputs to prevent data leaks. Prepared statements ensure user queries are executed before being passed to the database server. Additionally, PII and SPII should be salted and hashed using secure hash algorithms such as SHA-256.*
III. Decompose application	Sample data flow diagram A user sends a request for a product from a list of product catalogs stored in a database. The database server responds with the intended output. This is a bidirectional process.
IV. Threat analysis	List **2 types of threats** in the PASTA worksheet that are risks to the information being handled by the application. • *The app login form could be injected with malicious code since SQL is a popular programming language for storing data.* • *Business competitors could hire unauthorized hackers and propose offers to those who can access our systems and obtain sensitive private information. Such as business plans and goals.*
V. Vulnerability	List **2 vulnerabilities** in the PASTA worksheet that could be

analysis	exploited. ● *Outdated dependencies could be exploited by attackers.* ● *Unsanitized user inputs would lead to SQL injection attacks* ● *Basic HTTP auth transmits sensitive data over the network without securing them. HTTPS is recommended.*
VI. Attack modeling	[Sample attack tree diagram] Insecure collection of user data can result in SQL injection, where an attacker inserts malicious code into queries to gain access to sensitive information and administrative access. This attack can be due to a lack of prepared statements. Session cookies are vulnerable to theft by attackers if they are mishandled. This gives attackers complete access to user accounts. This vulnerability can arise from weak login credentials. Attackers can use brute force methods to guess usernames and passwords, ultimately gaining access to valid accounts.
VII. Risk analysis and impact	1. Encryption to ensure data at rest and in transit is unreadable to unintended users. 2. Prepared statements to execute SQL statements before passing them to the database server 3. An effective control measure would be to avoid embedding sensitive information, such as API keys, directly into the app's source code. 4. Security must be implemented in all stages of the Software Development Lifecycle (SDLC)