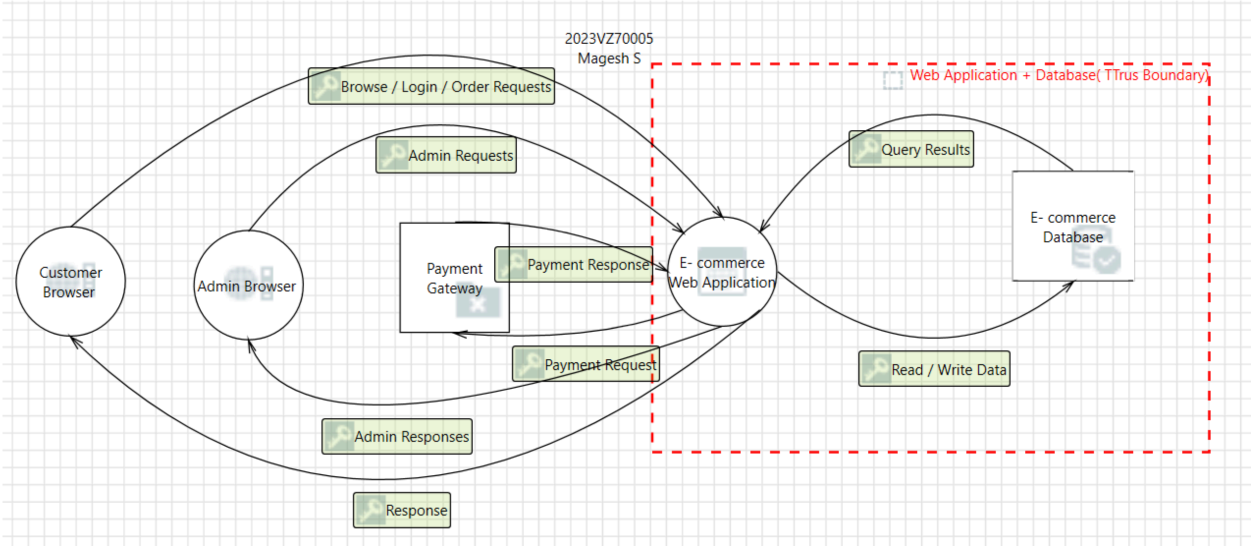
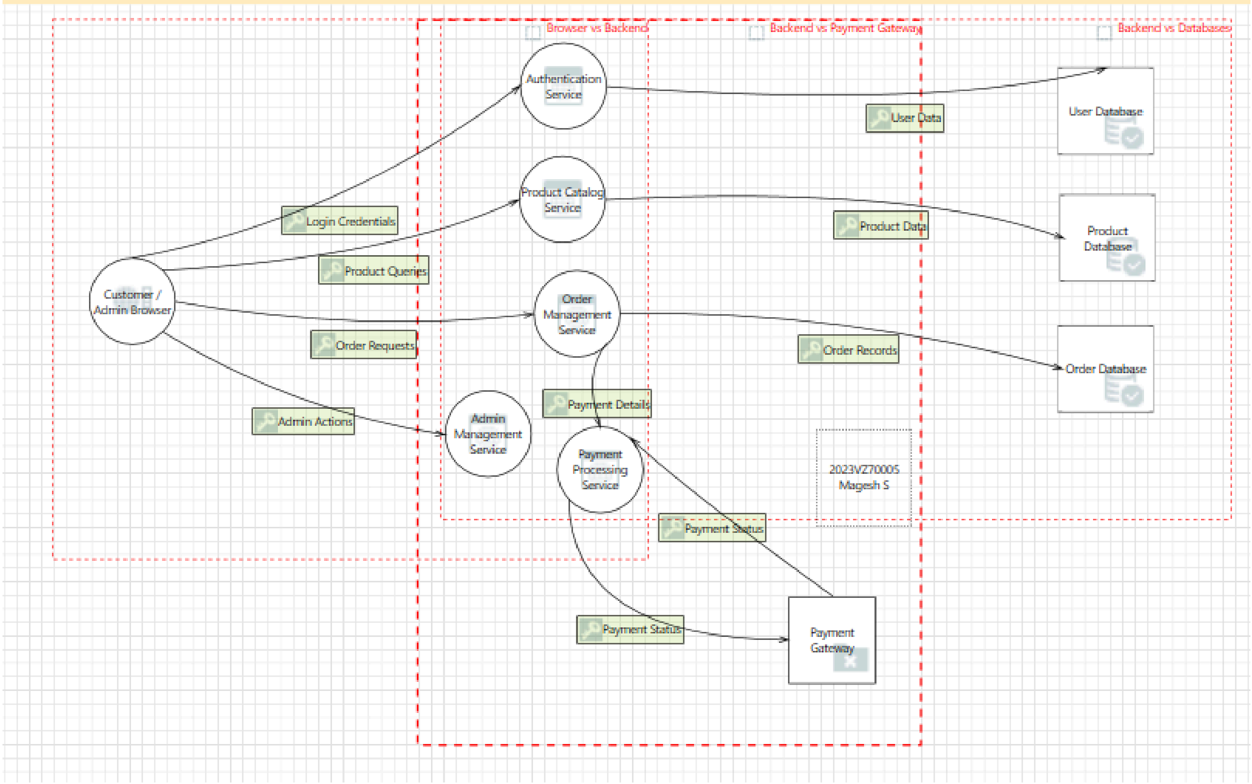
1. Create Level 0 and Level 1 DFDs showing processes, data stores, data flows, trust boundaries, and actors.

Level 0 DFD



Level 1 DFD



2. For each significant DFD element, list applicable STRIDE threats (S/T/R/I/D/E) with a short attack scenario

A. External User / Admin

STRIDE	Threat Scenario	Likelihood	Impact
S	Attacker spoofs user/admin credentials to access system	Med	High
T	Malicious client tampers with request parameters (price, role)	Med	High
R	User denies submitting an order or admin action	Med	Med
I	Sensitive data exposed via verbose errors or responses	Low	Med
D	Automated requests flood login or admin endpoints	Med	Med
E	User escalates to admin via broken access control	Low–Med	High

B. Authentication Service

STRIDE	Threat Scenario	Likelihood	Impact
S	Forged tokens or replayed credentials accepted	Med	High
T	Auth requests altered in transit across trust boundary	Low–Med	High
R	Auth service denies receiving or issuing auth decision	Med	High
I	Credential/token leakage via logs or memory	Low	High
D	Login brute-force or auth service exhaustion	Med	High

E	Weak role claims allow privilege escalation	Low	High

C. Order Management Service

STRIDE	Threat Scenario	Likelihood	Impact
S	Fake service or user submits forged order	Med	High
T	Order contents modified (quantity, price)	Med	High
R	Service denies receiving order from external source	Med	High
I	Order/customer data exposed to unauthorized parties	Low–Med	High
D	Order endpoint flooded, blocking legitimate purchases	Med	High
E	Order APIs used to access admin-only functions	Low	High

D. Product / Order Data Store

STRIDE	Threat Scenario	Likelihood	Impact
S	Unauthorized service impersonates DB client	Low	High
T	Data at rest altered (price, inventory, orders)	Low–Med	High
R	No audit trail for data changes	Med	Med
I	Data leakage via misconfigured access controls	Med	High
D	DB locking or resource exhaustion	Low–Med	High
E	Over-privileged DB roles abused	Low	High

E. Inter-Service Data Flows (Trust Boundaries)

STRIDE	Threat Scenario	Likelihood	Impact
S	One service impersonates another	Low–Med	High
T	Messages modified in transit	Low	High
R	Sender/receiver disputes transaction	Med	Med
I	Cleartext or weakly protected data flows	Low	High
D	Network-level DoS on service links	Med	Med
E	Service gains broader access than intended	Low	High

3. Rate likelihood and impact (Low/Med/High). Prioritize the top 8 threats and justify briefly.
4. Propose mitigations per prioritized threat (short-term and recommended long-term control).

Justification for both questions , below

1. Elevation Using Impersonation [State: Mitigation Implemented]
[Priority: High]
Category: Elevation Of Privilege
Description: E- commerce Web Application may be able to impersonate the context of Admin Browser in order to gain additional privilege.
Justification:
Short-term: Enforce strict session management and audit admin action logs.
Long-term: Implement attribute-based access control (ABAC) and multi-factor authentication (MFA) for all admin actions.
Short Description: A user subject gains increased capability or privilege by taking advantage of an implementation bug.

2. Spoofing the E- commerce Web Application Process [State: Mitigation Implemented]
[Priority: High]
Category: Spoofing

Description: E- commerce Web Application may be spoofed by an attacker and this may lead to unauthorized access to Admin Browser . Consider using a standard authentication mechanism to identify the source process.
Justification:
Short-term: Enforce strong TLS/HTTPS with valid certificates for the web application.
Long-term: Implement WAF with anti-spoofing rules and integrate with a centralized identity provider (e.g., OAuth/OpenID Connect).
Short Description: Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.

3. Elevation Using Impersonation [State: Mitigation Implemented]
[Priority: High]
Category: Elevation Of Privilege
Description: E- commerce Web Application may be able to impersonate the context of Customer Browser in order to gain additional privilege.
Justification:
Short-term: Enforce secure session tokens and server-side authorization checks.
Long-term: Implement a centralized identity provider with continuous authentication monitoring.
Short Description: A user subject gains increased capability or privilege by taking advantage of an implementation bug.

4. Elevation by Changing the Execution Flow in E- commerce Web Application [State: Mitigation Implemented]
[Priority: High]
Category: Elevation Of Privilege
Description: An attacker may pass data into E- commerce Web Application in order to change the flow of program execution within E- commerce Web Application to the attacker's choosing.
Justification:
Short-term: Implement strict input validation and server-side authorization checks.
Long-term: Deploy runtime application self-protection (RASP) and conduct regular threat modeling on the application flow.
Short Description: A user subject gains increased capability or privilege by taking advantage of an implementation bug.

5. Data Flow Payment Request Is Potentially Interrupted [State: Mitigation Implemented]
[Priority: High]
Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification:
Short-term: Secure payment flows with HTTPS, retry logic, and circuit breakers.
Long-term: Implement multi-region deployment and DDoS mitigation for payment service availability.
Short Description: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

6. Potential Process Crash or Stop for E- commerce Web Application [State: Mitigation Implemented]
[Priority: High]
Category: Denial Of Service

Description: E- commerce Web Application crashes, halts, stops or runs slowly; in all cases violating an availability metric.
Justification:
Short-term: Implement rate limiting and request throttling on critical endpoints.
Long-term: Deploy auto-scaling, health monitoring, and DDoS protection services.
Short Description: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

7. E- commerce Web Application May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented]
[Priority: High]
Category: Elevation Of Privilege
Description: Payment Gateway may be able to remotely execute code for E- commerce Web Application.
Justification:
Short-term: Sanitize and validate all incoming data from the Payment Gateway.
Long-term: Implement a secure API gateway with strict schema validation and deploy sandboxing for external integrations.
Short Description: A user subject gains increased capability or privilege by taking advantage of an implementation bug.

8. Potential Excessive Resource Consumption for E- commerce Web Application or E- commerce Database [State: Mitigation Implemented]
[Priority: High]
Category: Denial Of Service
Description: Does E- commerce Web Application or E- commerce Database take explicit steps to control resource consumption? Resource consumption attacks can be hard to deal with, and there are times that it makes sense to let the OS do the job. Be careful that your resource requests don't deadlock, and that they do timeout.
Justification:
Short-term: Enforce rate limiting, request quotas, and query timeouts.
Long-term: Implement auto-scaling, resource isolation, and real-time anomaly detection.
Short Description: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

9. Potential SQL Injection Vulnerability for E- commerce Database [State: Mitigation Implemented]
[Priority: High]
Category: Tampering
Description: SQL injection is an attack in which malicious code is inserted into strings that are later passed to an instance of SQL Server for parsing and execution. Any procedure that constructs SQL statements should be reviewed for injection vulnerabilities because SQL Server will execute all syntactically valid queries that it receives. Even parameterized data can be manipulated by a skilled and determined attacker.
Justification:
Short-term: Enforce parameterized queries and strict input validation for all database operations.
Long-term: Deploy a database firewall and integrate static/dynamic SQL injection testing into the CI/CD pipeline.

Short Description: Tampering is the act of altering the bits. Tampering with a process involves changing bits in the running process. Similarly, Tampering with a data flow involves changing bits on the wire or between two running processes.

10. Data Flow Response Is Potentially Interrupted [State: Mitigation Implemented] [Priority: High]
Category: Denial Of Service
Description: An external agent interrupts data flowing across a trust boundary in either direction.
Justification:
Short-term: Implement HTTPS with retry logic and timeouts for resilient communication.
Long-term: Deploy a global load balancer with DDoS protection and auto-scaling to maintain availability.
Short Description: Denial of Service happens when the process or a datastore is not able to service incoming requests or perform up to spec.

11. Customer Browser May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: High]
Category: Elevation Of Privilege
Description: E- commerce Web Application may be able to remotely execute code for Customer Browser .
Justification:
Short-term: Implement Content Security Policy (CSP) headers and enforce strict output encoding.
Long-term: Deploy client-side protection tools and conduct regular dynamic application security testing (DAST).
Short Description: A user subject gains increased capability or privilege by taking advantage of an implementation bug.

12. Admin Management Service May be Subject to Elevation of Privilege Using Remote Code Execution [State: Mitigation Implemented] [Priority: High]
Category: Elevation Of Privilege
Description: Customer / Admin Browser may be able to remotely execute code for Admin Management Service.
Justification:
Short-term: Implement strict input validation and parameterized queries.
Long-term: Deploy a Web Application Firewall (WAF) with RCE rules and enforce MFA and network segmentation.
Short Description: A user subject gains increased capability or privilege by taking advantage of an implementation bug.

13. Potential Data Repudiation by Authentication Service [State: Mitigation Implemented] [Priority: High]
Category: Repudiation
Description: Authentication Service claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification:
Short-term: Log all authentication events with timestamps and source IPs.
Long-term: Integrate with a centralized SIEM for immutable logs and enforce digital signatures on authentication requests.
Short Description: Repudiation threats involve an adversary denying that something happened.

14. Potential Data Repudiation by Order Management Service [State: Mitigation Implemented] [Priority: High]
Category: Repudiation
Description: Order Management Service claims that it did not receive data from a source outside the trust boundary. Consider using logging or auditing to record the source, time, and summary of the received data.
Justification:
Short-term: Implement comprehensive logging with timestamps and source identifiers for all order requests.
Long-term: Enforce digital signatures or HMAC-based authentication to create tamper-evident, non-repudiable audit trails.
Short Description: Repudiation threats involve an adversary denying that something happened.

15. Spoofing the Product Catalog Service Process [State: Mitigation Implemented] [Priority: High]
Category: Spoofing
Description: Product Catalog Service may be spoofed by an attacker and this may lead to unauthorized access to Product Database. Consider using a standard authentication mechanism to identify the source process.
Justification:
Short-term: Enforce TLS client certificate authentication for service-to-service communication.
Long-term: Implement a service mesh (e.g., Istio) for automated mTLS and identity-based authorization across all microservices.
Short Description: Spoofing is when a process or entity is something other than its claimed identity. Examples include substituting a process, a file, website or a network address.