Assignment 3: As an architect design error handling with examples for ecommerce system.

# Solution:

Error handling is an essential aspect of any system, including an ecommerce system. As an architect, you need to design a comprehensive error handling mechanism to ensure that errors are captured, logged, and handled appropriately. Below are some steps to design an error handling mechanism with examples for an ecommerce system.

Define the Error Handling Strategy: The first step in designing an error handling mechanism is to define a strategy that outlines how the system should handle errors. The strategy should cover how to capture, log, and report errors.

Define Error Messages: The next step is to define error messages that the system should use to communicate errors to users. Error messages should be clear, concise, and provide users with enough information to understand the error and how to resolve it. For example, a common error message for an ecommerce system could be "Payment Failed: Insufficient Funds."

Use HTTP Status Codes: HTTP status codes are a standardized way of communicating the status of a request to a client. In an ecommerce system, HTTP status codes can be used to communicate the success or failure of a request. For example, a 200 OK status code can be used to indicate that a request was successful, while a 404 Not Found status code can be used to indicate that a requested resource was not found.

Use Error Codes: Error codes can be used to provide more detailed information about errors. For example, an error code could be used to indicate that a payment failed due to insufficient funds. Error codes should be unique and standardized across the system.

Log Errors: Logging errors is essential for debugging and troubleshooting issues. Errors should be logged with as much detail as possible, including the error message, error code, timestamp, and any relevant request or response data.

Provide Feedback to Users: Users should be provided with feedback when errors occur, indicating what went wrong and how to resolve the issue. Feedback can be provided via error messages or a dedicated error page.

Examples:

Invalid Input: If a user enters invalid input, such as an invalid email address, the system should return a 400 Bad Request status code with an appropriate error message, such as "Invalid Email Address."

Resource Not Found: If a user requests a resource that does not exist, such as an invalid product ID, the system should return a 404 Not Found status code with an appropriate error message, such as "Product Not Found."

Authentication Failed: If a user fails to authenticate, the system should return a 401 Unauthorized status code with an appropriate error message, such as "Invalid Credentials."

Payment Failed: If a payment fails, the system should return a 402 Payment Required status code with an appropriate error message, such as "Insufficient Funds."

Server Error: If a server error occurs, the system should return a 500 Internal Server Error status code with an appropriate error message, such as "Server Error Occurred." The system should also log the error with as much detail as possible to facilitate debugging and troubleshooting.