Project Development Phase

Exception Handling

TEAM LEAD	S.KARTHIKRAJ
NM ID	B2DF7E4FB2160FDA23848332D658AAE7
PROJECT NAME	Creating a social media ad campaign in facebook

Error Page: Create custom error pages for common HTTP error status codes to provide users with a user-friendly message and possibly a way to navigate back to the main site.

Logging: Implement robust logging mechanisms to record errors and exceptions. Logs can help developers diagnose issues and track problems.

Validation: Use input validation to prevent common errors, such as invalid data submission in forms or SQL injection attempts. Client-side and server-side validation can help catch and prevent errors.

Exception Types: Distinguish between different types of exceptions, such as application-specific errors and system-level errors. Handle each type appropriately. For example, you might want to show a user-friendly message for application-specific errors but log system-level errors for investigation.

Security: Exception handling should not leak sensitive information about your website’s structure or code in error messages. Ensure that error messages are general and don’t reveal specifics about your server or database.

Monitoring and Alerts: Set up monitoring tools that can detect anomalies and issues in real-time. Implement alerting systems to notify the development team when critical errors occur.

Testing: Thoroughly test your website to uncover and fix potential exceptions and errors. This includes both automated testing (e.g., unit tests) and manual testing.

User Feedback: If a user encounters an error, provide a way for them to report the issue, such as a contact form or support email, to gather more information for debugging.

```
Try {
    // Your JavaScript code that may throw exceptions
    // For example, handling a click event
    Document.getElementById(“buy-button”).addEventListener(“click”, function() {
        // Simulate an error for demonstration
        Throw new Error(“Product out of stock.”);
    });
} catch (error) {
    // Handle the client-side exception
    Console.error(“Client-Side Error:”, error.message);
    // You can display an error message to the user or perform other actions
}
```