**Department of Collegiate & Technical Education**

**Capstone Project**

**Format-5**

**RISK ANALYSIS**

**Capstone Project Name:** INVENTORY MANAGEMENT SYSTEM

**Capstone Project Members:** SHESHAGIRI S [149CS21046]

DIVJA J [149CS21021]

DILEEP G P [149CS21020]

DHANYASHREE A P [149CS21019

**Risk Analysis Report For Inventory Management Report**

This report identifies and analyzes potential risks associated with developing and using an inventory management system built with PHP. It outlines the impact and likelihood of each risk, and proposes mitigation strategies to minimize their occurrence or severity. By understanding and addressing these risks, you can enhance the security, reliability, and effectiveness of your inventory management system.

**I. System Description:**

Please provide a brief description of your inventory management system, including its functionalities, target audience, and deployment environment (e.g., web-based, cloud-hosted). This information will help tailor the risk analysis to your specific context.

**II. Risk Identification:**

1. Development Risks:

* Security vulnerabilities in PHP code: Improper coding practices or known vulnerabilities in PHP libraries can create entry points for attackers.
* SQL injection attacks: Malicious code injected into user input can manipulate database queries and compromise data.
* Cross-site scripting (XSS) attacks: Injection of malicious scripts into web pages can steal user data or redirect them to phishing sites.
* Insufficient input validation: Unvalidated user input can lead to data corruption, denial-of-service attacks, or unexpected behavior.
* Lack of access control: Unauthorized users might gain access to sensitive inventory data or functionalities.

1. **Operational Risks:**

* System downtime: Hardware failures, software bugs, or cyberattacks can disrupt inventory management operations.
* Data loss or corruption: Accidental deletion, hardware failure, or cyberattacks can lead to data loss, impacting inventory accuracy and decision-making.
* Integration issues: Integration with other systems like accounting or e-commerce platforms may introduce compatibility problems or data inconsistencies.
* Human error: Manual data entry or process errors can lead to inaccurate inventory levels and order fulfillment issues.

1. **Business Risks:**

* Stockouts: Inaccurate inventory data or inefficient ordering processes can result in stockouts, leading to lost sales and customer dissatisfaction.
* Overstocking: Excess inventory ties up capital, increases storage costs, and risks product obsolescence.
* Fraudulent activity: Weak access controls or security measures can facilitate theft or manipulation of inventory data.
* Compliance issues: Failure to comply with industry regulations or data privacy laws can lead to fines and reputational damage.

**III. Risk Assessment:**

For each identified risk, consider the following factors to assess its overall impact and likelihood:

* Impact: What are the potential consequences of the risk occurring (e.g., financial loss, reputational damage, operational disruption)?
* Likelihood: How probable is it that the risk will occur, considering current security measures and development practices?

Based on this assessment, prioritize risks based on their overall severity (impact multiplied by likelihood).

**IV. Risk Mitigation Strategies:**

For each prioritized risk, outline specific actions to mitigate its impact or likelihood. Consider strategies like:

* Secure coding practices: Employ secure coding guidelines and regularly update libraries to address known vulnerabilities.
* Input validation and sanitization: Validate and sanitize all user input to prevent malicious code injection.
* Access control mechanisms: Implement role-based access control and strong authentication methods.
* Data backups and disaster recovery: Regularly back up data and have a disaster recovery plan in place.
* System monitoring and logging: Monitor system activity for suspicious behaviour and log access attempts.
* Integration testing: Thoroughly test integrations with other systems to ensure data consistency.
* User training: Train users on secure password practices and how to identify phishing attempts.
* Regular security audits: Conduct periodic security audits to identify and address vulnerabilities.
* Compliance review: Ensure your system adheres to relevant industry regulations and data privacy laws.

V. Conclusion:

By proactively identifying, assessing, and mitigating risks, you can significantly improve the security, reliability, and effectiveness of your PHP-based inventory management system. Continuously monitor and update your risk management strategy as your system evolves and new threats emerge. Remember, security is an ongoing process, not a one-time event.

Date

Signature of the student Signature of the cohort owner