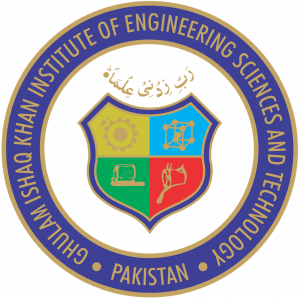
**Secure Software Design & Engineering(CY-321)**

**Project Proposal**

**Warehouse Management System**



**Group Members**

1. Ahmad Rashid(2022677)
2. Aizaz ur Rehman(2022078)
3. Muhammad Aurangzaib Bhatti(2022355)

**Introduction**

In an increasingly digital world, securing sensitive business data is crucial, particularly in warehouse management systems that handle inventory, transactions, and user information. This project aims to develop a **Warehouse Management System (WMS)** with integrated security features to safeguard both operational data and user privacy. The system will incorporate best practices in secure software design, such as role-based access control, multi-factor authentication (MFA), data encryption, and input validation, to prevent unauthorized access and mitigate cybersecurity risks. By prioritizing security, our goal is to provide a robust and reliable solution that enhances operational efficiency while ensuring the protection of critical business information.

# Problem Statement

# Warehouse management systems often deal with sensitive data, including inventory records, employee information, and transactional details. Without robust security measures, these systems are vulnerable to various cyber threats, such as unauthorized access, data breaches, and manipulation of inventory records. Existing solutions may lack comprehensive security features, leaving critical business data at risk. There is a clear need for a secure warehouse management system that incorporates real-time protection against cyber threats, ensuring the integrity and confidentiality of operational data.Solution

We will develop a website that:

* Uses multi-factor authentication (MFA) to enhance the security of user logins, particularly for administrative access.
* Encrypts sensitive data, such as passwords and inventory records, to prevent unauthorized access.
* Incorporates real-time input validation and sanitization to protect against injection attacks like SQL injection and XSS.

**Security Requirements**

To ensure the security and integrity of the Warehouse Management System, we will implement the following security measures:

* **Data Encryption**: All sensitive data, including user credentials, inventory details, and transactional records, will be encrypted both at rest and in transit using strong encryption algorithms (e.g., AES-256) to protect against unauthorized access.
* **Multi-Factor Authentication (MFA)**: To enhance login security, multi-factor authentication will be required, especially for administrative users. This provides an additional layer of protection beyond just passwords.
* **Secure Communication**: All communication between users and the warehouse system, as well as between system components, will be secured using HTTPS (SSL/TLS encryption) to prevent eavesdropping and man-in-the-middle (MITM) attacks.
* **Input Validation and Sanitization**: The system will ensure that all user inputs are validated and sanitized to prevent common vulnerabilities such as SQL injection, cross-site scripting (XSS), and command injection attacks.
* **Audit Logs**: The system will maintain comprehensive logs of all user activities for auditing purposes. These logs will help detect and track any suspicious activity and ensure accountability.
* **Data Backup and Recovery**: Regular backups of critical system data, including inventory and user information, will be performed to ensure data recovery in the event of system failures or attacks (e.g., ransomware).

### **Security Planning**

1. **Threat Modeling & Risk Assessment:**
   * **Potential Threats:**
     + Unauthorized access to sensitive data (e.g., inventory records, user credentials).
     + SQL injection or cross-site scripting (XSS) attacks on the application.
     + Insider threats, such as employees with elevated privileges performing malicious activities.
     + Data breaches during communication between the frontend and backend systems.
   * **Mitigation Strategies:**
     + Utilize **multi-factor authentication (MFA)** for high-privilege users and administrators.
     + Ensure all communication is encrypted using **HTTPS** (SSL/TLS) to prevent man-in-the-middle attacks.
2. **System Architecture & Secure Design:**
   * **High-Level Architecture Diagram:**
     + Show user interactions and detailing how users access inventory, update records, and generate reports.
     + Illustrate the communication flow between the frontend (user interface), backend (server-side logic), and the database (data storage).
     + Highlight secure areas where sensitive data is stored and accessed (e.g., user authentication data, inventory information).
   * **Component-Level Diagram:**
     + Diagram of key system features like **user authentication**, **inventory management**, **report generation**, and **audit logging**.
     + Security layers within each component (e.g., encryption for data storage, session management for login).
3. **Secure Coding & Initial Implementation:**
   * **Data Storage Security:**
     + Ensure sensitive data such as passwords are **hashed** and not stored in plain text.
     + Use **encryption** for sensitive data in the database (e.g., inventory details, transaction logs).
   * **Phishing & Malware Detection Mechanism:**
     + Develop a system for detecting and preventing malicious activities (e.g. unauthorized access attempts).
     + Implement security features like **input validation** to prevent malicious data from being entered into the system.
4. **Security Testing & Vulnerability Analysis:**
   * **Penetration Testing:**
     + Simulate attacks (e.g., SQL injection, XSS) to identify weaknesses in the system.
   * **Vulnerability Scanning:**
     + Conduct a **security audit** of the entire system to check for common vulnerabilities such as insecure endpoints or improper session handling.
5. **Final Implementation & Secure Code Review:**
   * **Security Fixes:**
     + Address any vulnerabilities discovered during the testing phase (e.g., fixing insecure database queries, correcting XSS vulnerabilities).
   * **Optimization:**
     + Ensure the system’s performance is optimized while maintaining security best practices (e.g., efficient encryption algorithms and session management techniques).
   * **Code Review:**
     + Conduct a **thorough code review** to ensure that all input data is sanitized and validated, all security best practices are followed, and the system is free from common vulnerabilities.
     + Ensure **security logging** is enabled to track user actions and flag suspicious activity.

**Conclusion**

The Warehouse Management System (WMS) with integrated security will provide a comprehensive and secure solution to manage inventory and sensitive business data. By incorporating robust security measures such as role-based access control, multi-factor authentication, and encryption, this system aims to protect critical data from unauthorized access and cyber threats. With a focus on secure coding practices and ongoing vulnerability testing, this project seeks to ensure the integrity and confidentiality of warehouse operations while enhancing the overall efficiency of inventory management.