



PRODUCTION/SHIFT MANAGEMENT

Database-driven website to present dynamic content



28 JULY 2026

OYEBAMIJI OLUWAFEMI OYEDOKUN
2409692

TABLE OF CONTENTS

1.0	ABOUT THE PRODUCTION AND SHIFT MANAGEMENT PHP WEBSITE	2
2.0	SQL INJECTION REPORT FOR PRODUCTION & SHIFT MANAGEMENT SYSTEM....	3
3.0	ANTI-SQL INJECTION DEFENCE IMPLEMENTED.....	5
4.0	DISCUSSION OF WEAKNESSES OF OTHER PAGES	7
5.0	CONCLUSION	8
	REFERENCES	9

1.0 ABOUT THE PRODUCTION AND SHIFT MANAGEMENT PHP WEBSITE

Introduction

Production and shift management is the system that the factory employs to coordinate who works where, daily shift Roaster and what product is running on each production line during each shift. In a busy production environment with many lines running different goods and workers, proper planning is vital to avoid understaffing, inappropriate product set-ups, missed targets and uncertainty about who is responsible for what [1].

MySQL Database

The Production & Shift Management website is database driven and shows dynamic content. All data related to employees, production lines, products, schedules, shift Roaster and shifts are stored in a MySQL database and displayed through PHP pages. The file name of MYSQL database is production_management_system.

System interactive

As an administrator, user can add, edit, view, and delete records for employees, production lines, products, roles, and shift rosters, making the system interactive and always up to date.

The Production Line Staffing Report

The Production Line Staffing Report says if each line is staffed with the right amount of people to avoid delays and safety issues.

The Product Running Report

The Product Running on the line Report verifies the product assigned to each line to reduce mix-ups and downtime.

The Daily Shift Report

The Daily Shift Report gives a summary of who worked, on what line and what was produced, boosting traceability and accountability.

Login Authentication

The basic security features like login authentication, session management and anti-SQL-injection protection guarantee that only authorised users may access and edit data, keeping the operating information of the factory secure, as well as enabling efficient and dependable production management.

2.0 SQL INJECTION REPORT FOR PRODUCTION & SHIFT MANAGEMENT SYSTEM

Injection of SQL queries is one of the biggest security holes in websites that use databases. It happens when an attacker changes user input, like

in login fields, search boxes, or URL parameters, to add bad SQL orders to a query. If the website puts user input into SQL statements without checking them or binding parameters, an attacker can make the database run commands that were not meant to be run. This could allow someone else to get in without permission, steal data, delete records, or even take over the whole production system.

SQL Injection is a big problem in a Production & Shift Management System because it saves private work information like employee information, shift schedules, production line assignments, and daily reports. If someone gets into these tables, they could change records about staffing, delete shift history, or get private information about employees. This would stop the factory from working, which would put safety and responsibility at risk.

3.0 ANTI-SQL INJECTION DEFENCE IMPLEMENTED

Anti-SQL Injection defences are strategies employed to prevent adversaries from embedding malicious SQL instructions into your database queries [2].

The image below shows the preparation statements with parameter binding are used by the login system. This stops SQL Injection because the SQL query is already written and any user input is only seen as data and not executable SQL. Even if an attacker enters malicious input such as “Mobolanle123” the database will not execute it as part of the query. Instead, the input is safely linked to the empty space (?). So, the effort to inject fails, and the login is turned down.

```
    } else {  
        $stmt = $conn->prepare("SELECT Username, Password FROM users WHERE Username = ?");  
        $stmt->bind_param("s", $username);  
        $stmt->execute();  
        $result = $stmt->get_result();  
  
        if ($result->num_rows === 1) {  
            $user = $result->fetch_assoc();  
        }  
    }  
}
```

Image 1 shows the coding of secured statement to protect the website

In the image below, the wrong password was inserted, and the system came up with incorrect password because the password is not match with what is in database as OYEBAMIJI for Username and OYEBAMIJI for password.

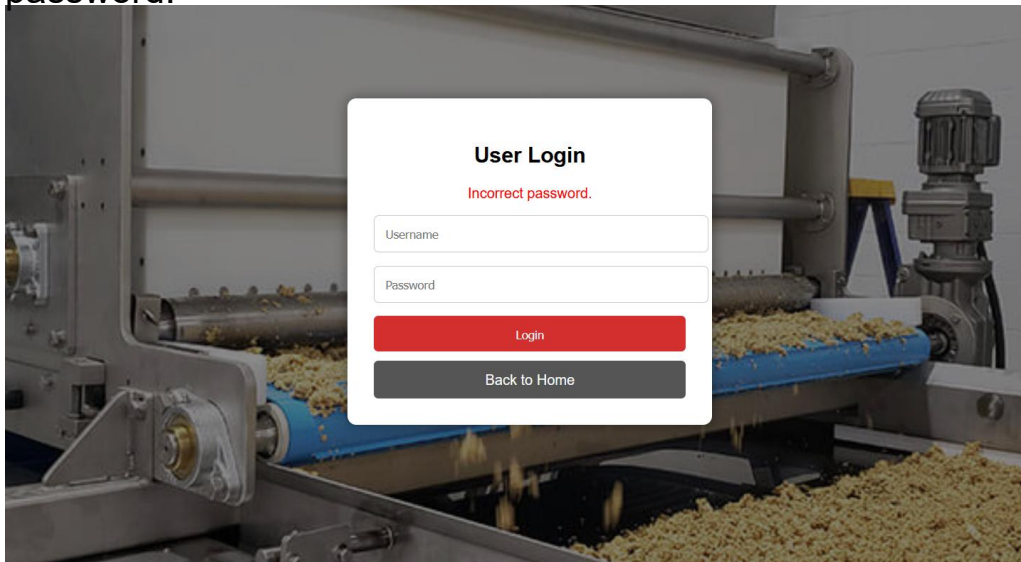


Image 2 shows the output of the wrong password after secured

The image below shows the login is successful after inserted OYEBAMIJI for Username and OYEBAMIJI for password. This allows user to access home page of Production & Shift Management

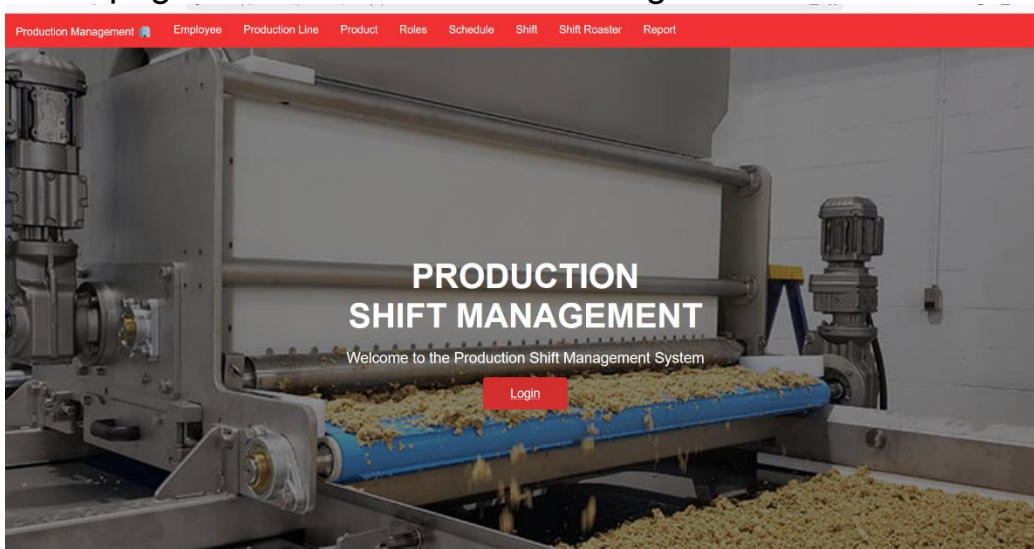


Image 3 shows the login is successful after inserted right login details

4.0 DISCUSSION OF WEAKNESSES OF OTHER PAGES

Some parts of the Production and Shift Management website still use unsafe dynamic SQL queries, even though the login page uses prepared statements to access the Homepages. Because these queries add user input straight to the SQL command, they can be hacked using SQL Injection. Pages like `shift_roaster.php`, `createEmployee.php` and others use raw SQL that doesn't bind parameters.

For Example. As shown in the previous screenshot, an attacker can bypass the homepage and access other page. These other pages need full protection; these parts need to be refactored to use prepared statements.

The weakness is inserting queries without prepared statement, update and view queries without validation and delete queries without prepared statements.

These sections must be rewritten using prepared statements with parameter binding to ensure full protection across the website.

5.0 CONCLUSION

The Production and Shift Management System make things run more smoothly by centralising daily Roaster, scheduling employees, and assigning shifts.

The coding techniques was used such like prepared statements; this protects the data better and stop SQL injection. The method improves overall productivity, accuracy, and dependability in all daily business tasks.

REFERENCES

- [1] S. K. Deb and P. K. Das, "A study on production line optimisation using real-time monitoring systems," *Automation Science and Engineering*, vol. 20, no. 2, pp. 780-792, 2023.
- [2] J. M. Santos and R. F. Silva, "Evaluating the effectiveness of prepared statements in preventing SQL injection attacks," *Journal of Cybersecurity and Digital Forensics*, vol. 5, no. 2, pp. 90-102, 2024.