Tracking of Preventive Maintenance System

Divyansh Singh 12308167

Umesh Desaboina 12312875

Palak sachan 12314117

Sahil Yadav 12309967

Ritika Saha **12015741**

Abstract

This document details the design and implementation of a web-based Preventive Maintenance Tracking System developed using PHP and MySQL. The system aims to automate the process of scheduling, tracking, and resolving preventive maintenance tasks. It includes modules for user registration, complaint lodging, task assignment, and performance tracking. The report provides an in-depth explanation of the purpose, architecture, file structure, and database schema of the system.

I. Introduction

Preventive maintenance is a crucial strategy adopted by organizations to ensure the uninterrupted functioning of equipment and infrastructure. It involves regularly scheduled inspections and servicing of equipment to prevent unexpected breakdowns and reduce long-term repair costs. This project presents a web-based system that streamlines the management of preventive maintenance tasks. The proposed system enables users to report complaints, allows administrators to assign tasks to technicians, and permits technicians to update the progress of tasks. The objective is to enhance operational efficiency and accountability while offering a transparent and traceable maintenance process.

II. System Overview

The system is developed using PHP and MySQL and operates as a web application with a structured role-based access control mechanism. It identifies three user roles: Admin, Technician, and General User. Each role has designated permissions and access to specific features:

- **Admin:** Manages users and technicians, assigns complaints, and monitors overall maintenance progress.
- **Technician:** Views assigned tasks and updates the status of complaints.
- **User:** Files complaints and tracks their resolution status.

The core system files include:

- **db.php:** Handles connection to the MySQL database using PHP's MySQLi extension.
- **index.php:** Provides a login interface where users authenticate themselves using email and password.
- **signup.php:** Facilitates new user registration and stores details in the users table.
- **dashboard.php:** Displays all system statistics and complaint information for admins.
- **user_dashboard.php:** Shows complaints filed by the logged-in user.
- **technician_dashboard.php:** Lists tasks assigned to the logged-in technician.

- **assign_task.php and update_status.php:** Allow task assignments and complaint status updates respectively.
- **manage_users.php, manage_technicians.php, delete_user.php:** Offer admin functionalities for user and technician management.

III. Database Design

The system uses a relational database to maintain structured and normalized data. Two key tables are explained below:

A. users Table

This table is responsible for storing authentication and role-related information of all users within the system. It enables access control and user-specific dashboard redirection. CREATE TABLE users (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100), email VARCHAR(100) UNIQUE, pass VARCHAR(255), role VARCHAR(50)

);

Each field plays a specific role:

- id is a unique auto-incremented identifier for each user.
- name contains the full name of the user.
- email ensures unique login credentials.
- pass stores the encrypted password (note: should be hashed using algorithms like bcrypt).
- role distinguishes between Admin, Technician, and General User.

B. complaints Table

This table stores information about user complaints or maintenance requests. It supports tracking, assignment, and status updates of preventive maintenance tasks. CREATE TABLE complaints (id INT AUTO_INCREMENT PRIMARY KEY,
user_id INT, description

TEXT, status VARCHAR(50),

assigned_to VARCHAR(100),

created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP

);

Field descriptions are as follows:

- id is a unique identifier for each complaint.
- user_id references the id of the user who filed the complaint.
- description contains detailed information about the maintenance issue.
- status tracks the state of the complaint such as "Pending", "In Progress", or "Resolved".
- assigned_to refers to the technician responsible for resolving the complaint.
- created_at records the timestamp of complaint submission.

IV. File-wise Code Explanation

This section explains the role of each PHP file and its contribution to the system’s functionality.

A. db.php

This file initiates the connection between the application and the MySQL database. It ensures the application can perform queries and data transactions.

```
$conn = new mysqli("localhost", "root", "",
"maintenance_db");
```

```
if ($conn->connect_error) {
    die("Connection failed: " . $conn>connect_error);
}
```

B. index.php

Acts as the login interface for users. It validates email and password input, checks user credentials against the users table, and redirects them to their respective dashboards based on their role.

C. signup.php

Presents a user registration form and processes the inputs to insert new users into the users table. It ensures no duplicate email is used and prepares the system for user login.

D. dashboard.php

The central hub for admin users. It verifies session credentials, retrieves all complaints using SQL JOIN operations, and allows admins to view user data and task assignments.

E. raise_complaint.php

This file allows authenticated users to report maintenance complaints. The form collects a description and stores it in the complaints table with status set to "Pending" by default. **F. assign_task.php**

Admins use this module to assign specific complaints to technicians. It updates the assigned_to field of the relevant row in the complaints table.

G. update_status.php

Technicians access this interface to update the progress of the tasks assigned to them. It modifies the status field in the complaints table.

H. manage_users.php / manage_technicians.php

These are admin modules that list all users or technicians. Admins can perform Create, Read, Update, and Delete (CRUD) operations through forms and buttons.

I. download_reports.php

Generates downloadable reports in CSV format for all complaints. This is particularly useful for analytics and audit purposes.

V. Key Features

This system provides several technical and operational features that enhance its reliability and usability:

Session Management: Session variables track authenticated users, and unauthorized access to restricted pages is blocked. This is achieved using session_start() and conditional redirects.

SQL Joins: JOIN queries are used to fetch data that spans multiple tables, especially to show complaints along with user and technician information.

Role-based Access: Each user role is given a different interface and permissions, ensuring secure and organized data access.

CSV Reporting: Admins can export complaint data to CSV files to track key performance indicators (KPIs) and for further analysis.

VI. Security Considerations

While the system performs basic role validation and user management, additional layers of security are recommended:

Password Hashing: Currently, passwords are stored in plain text. They should be hashed using algorithms like bcrypt before insertion into the database.

SQL Injection Prevention: All SQL queries should use prepared statements or parameterized queries to avoid injection attacks.

Input Validation: Every form input should be validated and sanitized to prevent Cross-Site

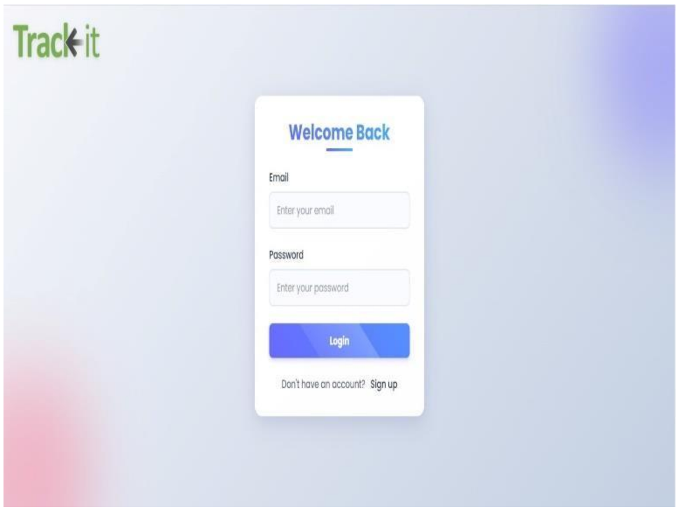
Scripting (XSS) and other common web vulnerabilities.

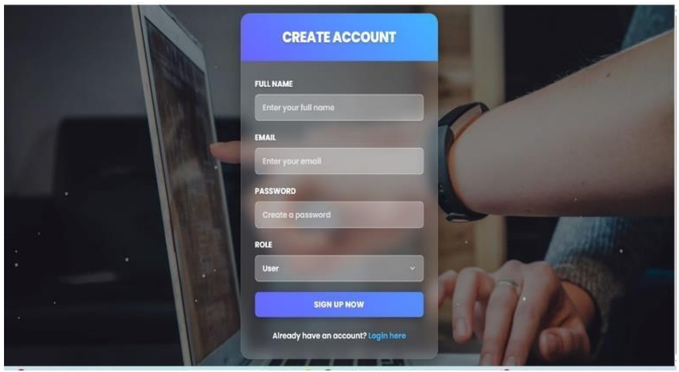
VII. Improvements and Future Work

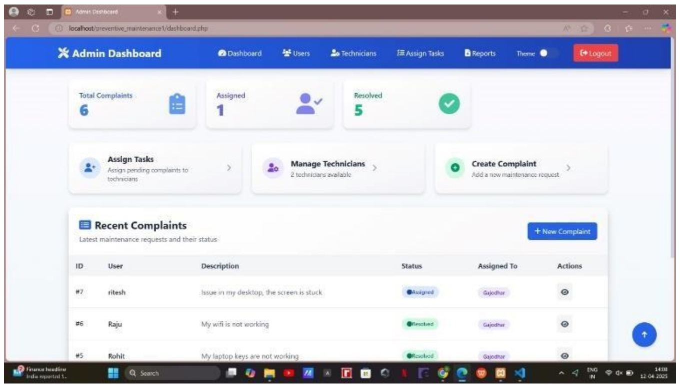
Several enhancements can be made to increase the system's scalability and security:

- **Email Notifications:** Send updates to users and technicians when a complaint is raised or resolved.
- **Dashboard Charts:** Include graphical reports such as complaint status pie charts and technician performance graphs.
- **Calendar Integration:** Schedule preventive maintenance tasks and view them using a calendar interface.
- **Two-Factor Authentication (2FA):** Improve security by implementing 2FA for all users.
- **Cloud Deployment:** Host the system on a cloud platform with regular backups for better availability.

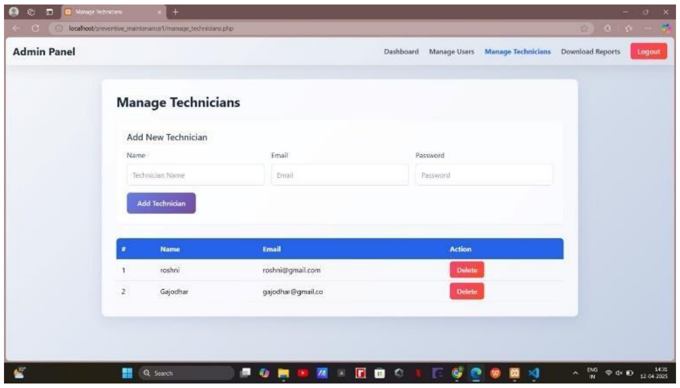
VIII. Final Output:



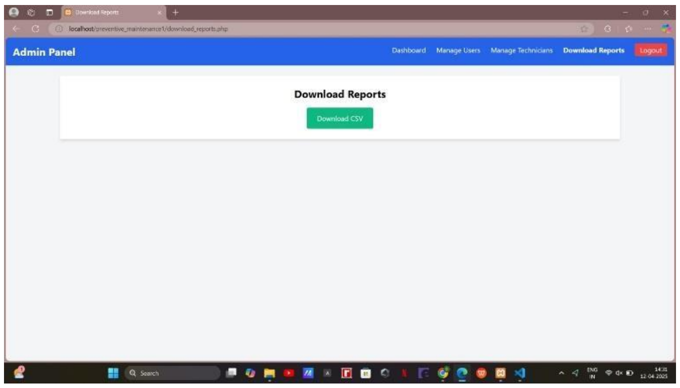


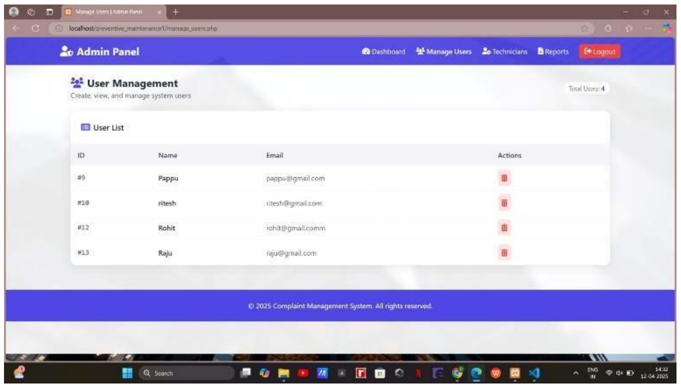


ID	User	Description	Status	Assigned To	Actions
#7	ritesh	Issue in my desktop, the screen is stuck	Assigned	Gajodhar	👁
#6	Raju	My wifi is not working	Resolved	Gajodhar	👁
#5	Rohit	My laptop keys are not working	Resolved	Gajodhar	👁



#	Name	Email	Action
1	roshni	roshni@gmail.com	Delete
2	Gajodhar	gajodhar@gmail.co	Delete




Manage Users | Admin Panel
localhost/preventive_maintenance1/manage_users.php
Admin Panel
Dashboard
Manage Users
Technicians
Reports
Logout

User Management
Create, view, and manage system users

Total Users: 4

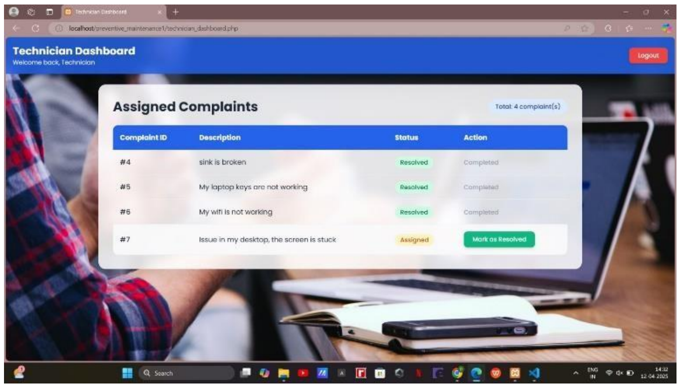
User List

ID	Name	Email	Actions
#9	Pappu	pappu@gmail.com	
#10	ritesh	ritesh@gmail.com	
#12	Rohit	rohit@gmail.comm	
#13	Raju	raju@gmail.com	

© 2025 Complaint Management System. All rights reserved.

IX. Conclusion

The Preventive Maintenance Tracking System is a functional and scalable web application that automates maintenance workflows. It simplifies user interactions, technician assignments, and administrative oversight, thereby minimizing delays and human errors. With further development and deployment in institutional or industrial environments, this system can provide significant operational value.

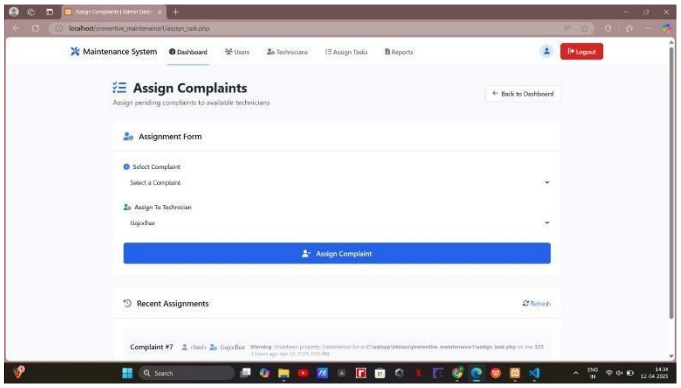


Technician Dashboard
Welcome back, Technician

Assigned Complaints

Total: 4 complaint(s)

Complaint ID	Description	Status	Action
#4	sink is broken	Resolved	Completed
#5	My laptop keys are not working	Resolved	Completed
#6	My wifi is not working	Resolved	Completed
#7	Issue in my desktop, the screen is stuck	Assigned	Mark as Resolved



Assign Complaints
Assign pending complaints to available technicians

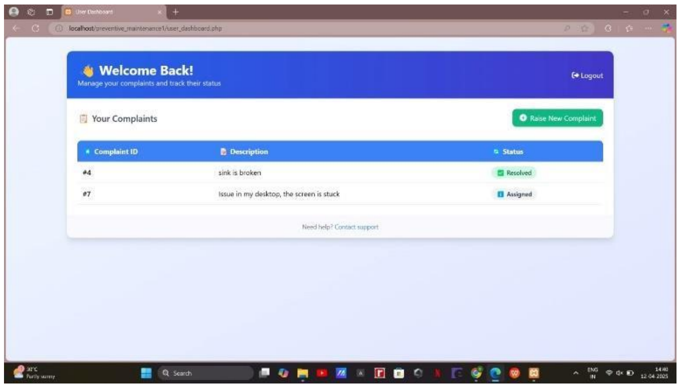
Assignment Form

Select Complaint: Select a Complaint

Assign To Technician: Gajodhar

Recent Assignments

Complaint #7 ritesh Gajodhar Warning: Undefined property: DateInterval::$w in C:\xampp\htdocs\preventive_maintenance1\assign_task.php on line 323



Welcome Back!
Manage your complaints and track their status

Your Complaints

Complaint ID	Description	Status
#4	sink is broken	Resolved
#7	Issue in my desktop, the screen is stuck	Assigned

Need help? Contact support

Profiles (Github):

https://github.com/Palaksachan5099/Preventive-Maintenance

https://github.com/Sahilyadav28/Preventive-Maintenance-

https://github.com/UmeshDesaboina/PREVENTIVE_MAINTANANCETRACKER

https://github.com/RitikaSaha14/Preventive_maintainence_system

Profiles(Linkedin):

https://www.linkedin.com/feed/update/urn:li:activity:7316879555380551680/

https://www.linkedin.com/posts/umesh-desaboina_webdevelopment-phpproject-javascript-activity-7316869826906333185-p1p-?utm_source=share&utm_medium=member_desktop&rcm=ACoAAEfla58BmcRYcS7GxhiLH8cn0-6Xa_-bnrU

https://www.linkedin.com/posts/palak-sachan-577905294_preventive-maintenance-tracking-web-app-streamlining-activity-7318343280188932097-HApd?utm_source=share&utm_medium=member_android&rcm=ACoAAEdTUOsBIWiVtDWFYuQBk25QUJ2-br7T6gc

https://www.linkedin.com/posts/ritika-saha-b87975202_chatgpt-you-said-i-want-a-linkeind-post-activity-7318538667726110721-8V02?utm_source=share&utm_medium=member_android&rcm=ACoAADPBq1cBIXURKeqJVbn8qjbqPeDwBAvh0OU

https://www.linkedin.com/posts/sahil-yadav28_project-update-tracking-preventive-maintenance-activity-7318484958073798656-zj7c?utm_source=share&utm_medium=member_desktop&rcm=ACoAAEgE3sgBd3CRiZcVmT12K1X8GVD27V0F2TA

References

[1] IEEE Citation Guidelines: https://ieeeauthorcenter.ieee.org [2] PHP Documentation: https://www.php.net/manual/en/ [3] MySQL Documentation: https://dev.mysql.com/doc/

[4] G. Mobley, *An Introduction to Predictive Maintenance*, 2nd ed., Elsevier, 2002.
→ A foundational book explaining preventive and predictive maintenance in industrial contexts.

[5] M. Rausand, A. Hoyland, *System Reliability Theory: Models, Statistical Methods, and*

Applications, 2nd ed., Wiley-Interscience, 2004.

→ Useful for understanding the reliability aspects that preventive maintenance aims to support.

[6] OWASP Foundation, *OWASP Top Ten Security Risks*, 2021. [Online]. Available: https://owasp.org/www-project-top-ten/ → Reference for implementing better security measures in web applications.

[7] J. Ullman and J. Widom, *A First Course in Database Systems*, 3rd ed., Pearson, 2007.

→ Explains relational database design principles used in systems like yours.

[8] W3Schools. *PHP MySQL Database*. [Online]. Available: https://www.w3schools.com/php/php_mysql_intro.asp

→ Beginner-friendly and relevant to your PHPMySQL-based implementation.

[9] T. Limoncelli, C. Hogan, and S. Chalup, *The Practice of System and Network Administration*, 3rd ed., Addison-Wesley, 2016.

→ Includes insights into managing large-scale infrastructure and maintenance.

[10] Stack Overflow Developer Survey 2023. [Online]. Available: https://survey.stackoverflow.co/2023/ → Can be used to support technology choices such as PHP and MySQL with recent developer trends.