

SMART OBSTACLE DETECTION STICK WITH DATA ANALYTICS SYSTEM

PROJECT REPORT 2025-2026

Submitted By

24591646 SUGI
24591636 AKASH
24507221 SURYA
24507200 JEEVA A

Under the guidance of

Mrs.N.MOIDHEEN BEE M.Sc.,B.Ed

LECT/COMPUTER

Diploma in Computer Engineering

of the Directorate of Technical Education, Government of Tamil Nadu



DEPARTMENT OF COMPUTER ENGINEERING

AKT MEMORIAL POLYTECHNIC COLLEGE

KALLAKURICHI-606202.

AKT MEMORIAL POLYTECHNIC COLLEGE

KALLAKURICHI-606202

Department of Computer Engineering

BONAFIDE CERTIFICATE

This is certified that this project work entitled **Smart Obstacle Detection Stick with Data Analytics System** has been submitted **SUGI, AKASH, SURYA, JEEVA A** in the partial fulfilment of the requirements for the award of Diploma in Computer Engineering during the academic year 2025-2026, who carried out the project work under our supervision.

Project Guide

Mrs.N.MOIDHEEN BEE M.Sc.,B.Ed
LECT/COMPUTER

Head of the Department

Mrs. C.SRIDEVI B.E.,(CSE)
HOD-COMPUTER & IT

This is to certify that _____
was examined for the project work viva-voce held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

Acknowledgement

We express our sincere and profound thanks to our chairman **Mr.A.K.T.Mahendiran B.Com.**, for creating this magnificent edifice of learning by providing all necessary facilities to complete diploma engineering course successfully.

We express our gratitude to our principal **Mr.P.K.Kabilar M.E.,MBA.**, for constant encouragement and facilities provided towards the successful completion of our project work.

At the same time we would like to express our heartfelt thanks to our head of the department **Mrs.C.Sridevi.,B.E.,(CSE).**, and also our project guide **Mrs. N.Moidheen Bee M.Sc., B.Ed.**, for guiding and needful advices and time to complete this project.

We would like to show our gratitude to our department staffs for all instructions and guidance given throughout.

Our sincere thanks and affection to our parents and friends who gave hand in all our steps.

CONTENTS

CHAPTER	TITLE	PAGE NO
	ABSTRACT	5
01	INTRODUCTION	6
02	LITERATURE SURVEY	8
03	PROPOSED SYSTEM	11
	3.1 ARCHITECTURE	13
	3.2 MODULES DESCRIPTION	17
04	IMPLEMENTATION DETAILS	23
	4.1 SOFTWARE ENVIRONMENT	26
	4.2 SYSTEM REQUIREMENTS	28
	4.3 SAMPLE CODING	29
	4.4 SCREENSHOT	41
05	CONCLUSION	47
	REFERENCES	49

ABSTRACT

The Smart Obstacle Detection Stick with Data Analytics System is an innovative assistive solution designed to enhance the mobility and safety of visually impaired individuals. Traditional walking sticks provide limited functionality, mainly offering physical support without real-time awareness of surrounding obstacles. This project addresses these limitations by integrating smart sensing technology with a web-based data management and analytics platform developed using Core PHP and MySQL.

The system is capable of detecting obstacles using sensor inputs such as ultrasonic or infrared sensors, which identify objects within a specific distance range. Upon detection, the system generates alerts through vibration or sound mechanisms to notify the user. In addition to real-time assistance, the system collects and stores data related to obstacle detection events, including distance, time, location (if integrated), and alert type.

A key feature of this project is its ability to perform multi-level data analysis. The collected data is stored in a structured MySQL database and processed through a web-based dashboard. The system provides analytical insights such as obstacle frequency, high-risk zones, sensor accuracy, user movement patterns, and system performance metrics. These insights are visualized using charts and reports, enabling administrators to monitor usage trends and improve system efficiency.

The application includes multiple modules such as user management, sensor data logging, obstacle records, alert history, analytics dashboard, report generation, and system settings. Each module supports full CRUD operations, ensuring flexibility and scalability. The use of a responsive Bootstrap-based interface ensures accessibility and ease of use.

Overall, this project demonstrates how embedded sensing technology combined with web-based analytics can significantly improve assistive devices. It not only enhances real-time navigation support but also contributes to long-term safety improvements through data-driven insights, making it a comprehensive and impactful final-year project..

1. INTRODUCTION

In today's rapidly advancing technological world, assistive technologies play a crucial role in improving the quality of life for individuals with disabilities. Among these, mobility assistance for visually impaired people remains a significant challenge. According to global studies, millions of people suffer from visual impairments, making independent navigation difficult and often unsafe. Traditional walking sticks, although widely used, provide only basic support and fail to offer real-time awareness of obstacles in the user's path. This limitation increases the risk of accidents, injuries, and dependency on others.

To overcome these challenges, the Smart Obstacle Detection Stick with Data Analytics System is proposed as an innovative solution that combines hardware-based sensing with software-based monitoring and analysis. The smart stick is designed to detect obstacles using sensors such as ultrasonic or infrared sensors, which can measure the distance between the user and nearby objects. When an obstacle is detected within a predefined range, the system provides immediate feedback through vibration or audio alerts, allowing the user to react promptly and avoid potential hazards.

Beyond real-time detection, this project introduces an advanced concept of data collection and analytical processing. Each obstacle detection event is recorded along with relevant parameters such as distance, timestamp, type of alert, and environmental conditions. This data is transmitted or stored in a backend system developed using Core PHP and MySQL, forming the foundation for a web-based monitoring dashboard. The integration of data analytics transforms the device from a simple assistive tool into an intelligent system capable of learning and improving over time.

The web application component of the project provides a comprehensive interface for administrators to manage users, monitor system activity, and analyze collected data. It includes features such as graphical reports, usage statistics, sensor performance evaluation, and identification of high-risk areas. By analyzing patterns in obstacle

detection, the system can help improve navigation strategies and enhance the overall safety of users. This analytical capability adds significant value to the project, making it suitable for advanced-level academic requirements.

Additionally, the project is designed with scalability and modularity in mind. The system consists of multiple interconnected modules, including user management, sensor data logging, obstacle tracking, alert management, analytics dashboard, and report generation. Each module is developed as a separate component using Core PHP, ensuring maintainability and ease of future enhancements. The use of a responsive Bootstrap-based user interface ensures accessibility across devices and provides a professional look and feel.

Furthermore, this project can be extended into an Internet of Things (IoT) solution by enabling real-time data transmission using wireless communication technologies such as Wi-Fi or GSM. It also has the potential to integrate artificial intelligence for predictive analysis and smarter decision-making. These possibilities make the project not only relevant for academic purposes but also valuable for real-world applications.

2. LITERATURE SURVEY

The development of assistive technologies for visually impaired individuals has been an active area of research for many years. Traditional mobility aids such as the white cane and guide dogs have been widely used to help visually impaired people navigate their surroundings. While these methods are effective to some extent, they have limitations in detecting obstacles beyond a short range and do not provide real-time analytical insights. With the advancement of embedded systems, sensors, and web technologies, researchers have proposed various smart solutions to overcome these challenges.

Early research focused on enhancing the traditional walking stick by integrating basic electronic components such as ultrasonic sensors and buzzers. These systems were designed to detect obstacles within a certain distance and alert the user through sound or vibration. Studies showed that ultrasonic sensors are effective in detecting objects in front of the user, especially in indoor environments. However, these systems were limited in functionality as they could only detect obstacles in a single direction and did not provide information about the type or severity of the obstacle.

Further developments introduced multi-sensor systems that combined ultrasonic sensors, infrared sensors, and water detection sensors. These systems improved the accuracy of obstacle detection and enabled detection of different types of hazards such as walls, stairs, and water puddles. Researchers also experimented with GPS modules to provide location tracking and navigation assistance. While these enhancements increased the usability of the smart stick, most systems still lacked a centralized data storage mechanism and analytical capabilities.

In recent years, the integration of Internet of Things (IoT) technology has significantly improved assistive devices. IoT-based smart sticks can transmit real-time data to cloud platforms, enabling remote monitoring and analysis. These systems allow caregivers or administrators to track user movements, monitor device performance, and receive alerts in case of emergencies. Research studies have demonstrated that IoT-

enabled devices can enhance user safety and provide valuable insights through continuous data collection. However, many of these solutions rely on complex hardware setups and expensive cloud services, making them less accessible for academic or low-cost implementations.

Another important area of research is the use of mobile applications and web-based systems to manage and visualize data collected from assistive devices. Several studies have proposed mobile apps that connect with smart sticks via Bluetooth to provide additional features such as voice navigation, route guidance, and emergency communication. Similarly, web-based dashboards have been used to display analytical reports, including obstacle frequency, user activity patterns, and sensor performance. These systems highlight the importance of combining hardware with software to create a complete and intelligent solution.

Artificial Intelligence (AI) and Machine Learning (ML) techniques have also been explored in assistive technologies. Researchers have developed systems that use computer vision and image processing to identify objects and provide detailed audio descriptions to users. While these systems offer advanced capabilities, they often require high computational power and are not suitable for low-cost implementations using Core PHP and MySQL. Therefore, simpler analytical approaches based on stored sensor data remain more practical for academic projects.

Despite these advancements, there are still gaps in existing systems. Many solutions focus primarily on real-time obstacle detection but do not emphasize data storage, historical analysis, and performance evaluation. Additionally, some systems are overly dependent on hardware complexity or require continuous internet connectivity, which may not be feasible in all environments. There is a need for a balanced solution that combines effective obstacle detection with a robust, low-cost, and user-friendly data management system.

The proposed project, **Smart Obstacle Detection Stick with Data Analytics System**, addresses these gaps by integrating obstacle detection with a web-based analytics platform developed using Core PHP and MySQL. Unlike previous systems, this project not only provides real-time alerts but also stores detailed records of obstacle detection events. These records are analyzed to generate meaningful insights such as usage patterns, high-risk areas, and system performance metrics. By focusing on both functionality and data analysis, the project offers a comprehensive solution that aligns with modern technological trends while remaining cost-effective and scalable.

3. PROPOSED SYSTEM

The proposed system, Smart Obstacle Detection Stick with Data Analytics System, is designed to provide an intelligent, reliable, and cost-effective solution to assist visually impaired individuals in navigating their surroundings safely. Unlike traditional walking sticks, this system integrates sensing technology with a web-based software platform to deliver both real-time assistance and long-term analytical insights.

The system consists of two major components: a hardware-based obstacle detection unit and a software-based data management and analytics system developed using Core PHP and MySQL. The hardware component includes sensors such as ultrasonic sensors and infrared sensors, which continuously monitor the environment for obstacles. These sensors measure the distance between the user and nearby objects and detect potential hazards such as walls, furniture, or pits. When an obstacle is detected within a predefined threshold distance, the system immediately alerts the user through vibration motors or buzzer sounds, enabling quick reaction and avoidance.

In addition to real-time detection, the system is designed to collect and store data for further analysis. Each detection event is recorded with attributes such as obstacle distance, detection time, alert type, and sensor status. This data is either transmitted through a communication module (such as Bluetooth or Wi-Fi) or manually uploaded to the server. The backend system stores this data in a structured MySQL database, ensuring efficient retrieval and management.

The software component plays a crucial role in transforming raw data into meaningful information. A web-based dashboard is developed using Core PHP and Bootstrap, providing an intuitive interface for administrators. The dashboard includes multiple modules such as user management, sensor data logging, obstacle tracking, alert history, and system monitoring. Each module supports full CRUD operations, allowing administrators to manage and update system records efficiently.

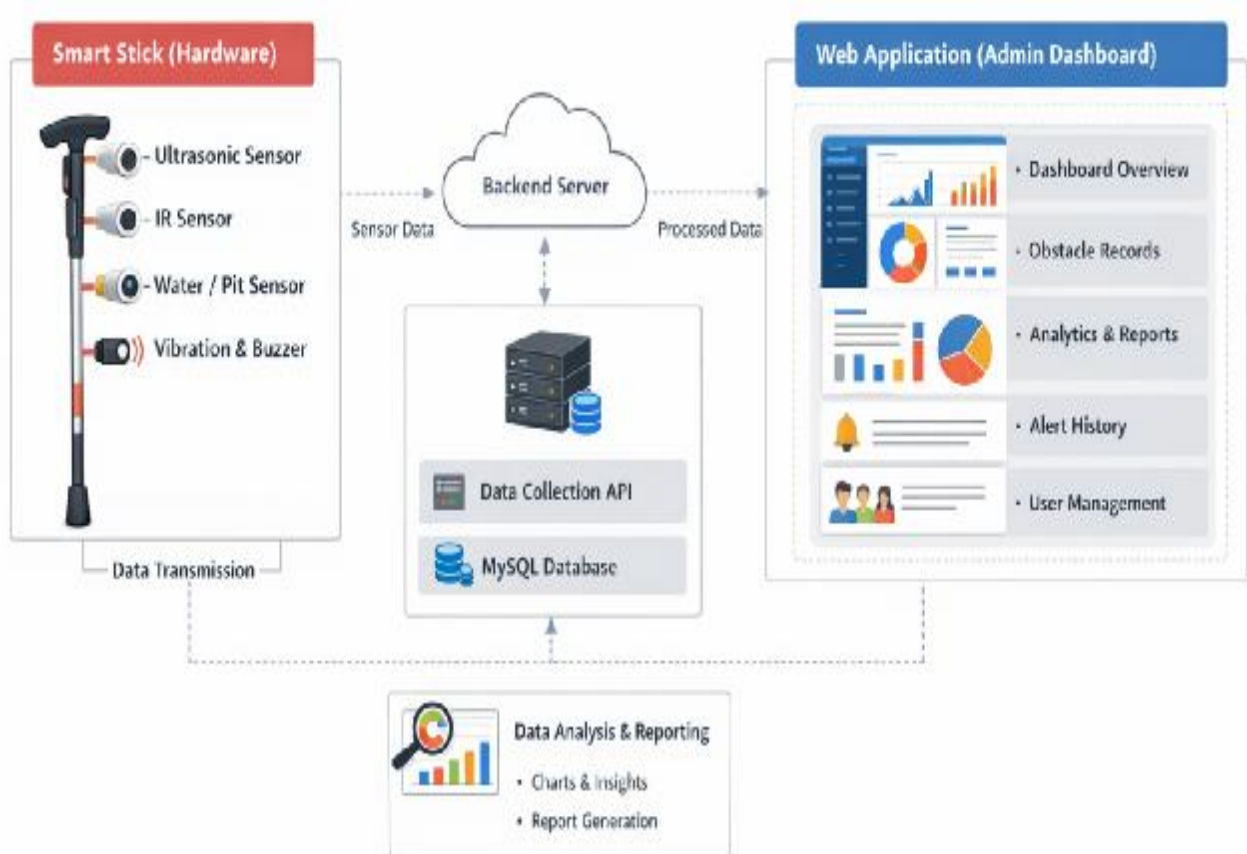
One of the key features of the proposed system is its data analytics capability. The system processes stored data to generate insights such as obstacle frequency, high-risk areas, sensor performance, and user movement patterns. These insights are presented using graphical charts, tables, and reports, making it easier to understand trends and identify areas for improvement. For example, the system can determine which locations have the highest number of obstacle detections or analyze the accuracy of sensors under different conditions.

The proposed system also includes a report generation module, which allows users to generate daily, weekly, or monthly reports. These reports can be used for academic evaluation, system improvement, and user safety assessment. Additionally, an admin panel is provided to manage system settings, monitor usage statistics, and control access to different modules.

To ensure flexibility and scalability, the system is designed using a modular architecture. Each functional component is implemented as a separate module, making it easier to maintain and extend the system in the future. The use of Core PHP ensures simplicity and compatibility, while MySQL provides a robust and efficient database management system.

3.1 ARCHITECTURE

Smart Obstacle Detection Stick with Data Analytics System Architecture



The architecture of the Smart Obstacle Detection Stick with Data Analytics System is designed as a layered and modular model that connects the physical assistive device with a web-based monitoring and analytics platform. The purpose of this architecture is to ensure that obstacle detection, alert generation, data storage, and analytical reporting work together as one integrated system. The overall architecture can be divided into four main layers: the smart stick hardware layer, the data transmission layer, the backend server and database layer, and the web application and analytics layer.

At the first level, the smart stick hardware layer forms the core sensing unit of the system. This layer includes components such as the ultrasonic sensor, infrared sensor, and water or pit detection sensor. The ultrasonic sensor is used to measure the distance between the stick and nearby obstacles, helping detect objects in front of the user. The infrared sensor supports short-range object sensing and improves detection accuracy in specific situations. The water or pit sensor is used to identify wet surfaces, holes, or sudden drops on the ground, which are especially dangerous for visually impaired users. These sensors continuously scan the environment while the stick is in use. Whenever a potential obstacle or hazard is detected, the system activates the alert mechanism using a buzzer or vibration motor, thereby informing the user instantly.

The second layer is the data transmission layer, which handles communication between the smart stick and the backend system. In a practical implementation, sensor readings can be sent through communication technologies such as Bluetooth, Wi-Fi, GSM, or serial USB transfer. For the academic Core PHP and MySQL version of this project, the system can simulate or receive obstacle records through an API or manual data entry interface. This layer is important because it acts as the bridge between hardware events and software processing. Each detection record may contain values such as sensor type, obstacle distance, alert mode, date, time, and status. The transmission layer ensures that the sensed data is forwarded in a structured format to the backend server for storage and analysis.

The third layer is the backend server and database layer, which is responsible for data processing, storage, and business logic execution. This layer is developed using Core PHP for server-side logic and MySQL for database management. The backend receives incoming sensor data through a data collection API or form submission modules. It validates the input, stores it in the respective database tables, and prepares it for further usage in reports and dashboard screens. In this layer, multiple tables are maintained for users, devices, obstacle logs, alert history, analytics summaries, reports, and other related entities. The backend also performs operations such as data filtering, record

classification, summary generation, and administrative control. Since the project is modular, each backend function is separated into individual PHP files, which improves code readability, maintenance, and future expansion.

The fourth layer is the web application and analytics layer, which provides a professional interface for system administrators and project evaluators. This layer is created using Bootstrap, HTML, CSS, JavaScript, Core PHP, and MySQL integration. It contains the dashboard, user management module, sensor records module, obstacle records module, alerts module, reports module, analytics charts, and settings pages. The dashboard presents a high-level overview of the system, including total users, total detections, total alerts, recent activity, and monthly summaries. The analytics section transforms stored obstacle data into visual insights such as bar charts, pie charts, and trend reports. These analytics help identify high-risk zones, frequent obstacle occurrences, device activity patterns, and system performance over time. This layer is essential because it converts raw detection logs into understandable information for decision-making and academic presentation.

The architecture also follows a centralized data flow model. First, the hardware sensors detect environmental obstacles. Second, the data is transmitted to the server. Third, the backend validates and stores the data in the MySQL database. Fourth, the web application retrieves the stored data and displays it as tables, reports, and graphs. This step-by-step data movement ensures that every obstacle detection event is not only used for immediate user safety but also preserved for historical analysis. In this way, the project supports both real-time assistance and post-event analytical evaluation.

Another important architectural feature is modularity. Instead of building the entire project as a single large file, the system is separated into modules such as authentication, dashboard, sensor management, obstacle management, alert management, analytics, reports, and settings. Each module interacts with the database independently but remains connected through the common application layout and navigation menu. This modular design makes the project more professional and easy to manage. It also

supports CRUD operations for major entities, which is important in final-year project development.

From a system design perspective, the architecture supports scalability and enhancement. In future versions, the same architecture can be extended by adding GPS tracking, voice navigation, mobile app support, cloud synchronization, caregiver monitoring, and machine learning–based risk prediction. Since the backend already stores structured data, these advanced features can be integrated without completely redesigning the system. This makes the architecture flexible and future-ready.

3.2 MODULES DESCRIPTION

1. User Management Module

The User Management Module is responsible for maintaining the details of all users who interact with the Smart Obstacle Detection Stick with Data Analytics System. This module mainly handles administrator accounts, system operators, and, if needed, registered stick users. It allows the admin to add new users, edit user details, update contact information, assign roles, and remove inactive accounts. The purpose of this module is to ensure that only authorized persons can access and manage the system.

This module also supports login credential management and account status control. User records may include name, email, mobile number, username, password, role, and status. In the project, this module is connected to the authentication system, ensuring secure access to the dashboard and data management pages. It improves accountability because every action performed in the system can be traced to a logged-in user or administrator.

From the final-year project point of view, this module is important because it demonstrates CRUD operations clearly. It also helps structure the entire application around role-based usage and organized data access. A professional interface with tables, forms, edit options, and delete confirmation makes this module complete and practical.

2. Authentication and Login Module

The Authentication and Login Module provides the security foundation of the project. It controls how users enter the system and protects sensitive project data from unauthorized access. This module includes login forms, session handling, password verification, and logout features. Only valid users with correct credentials are allowed to access the admin dashboard and project modules.

When a user enters a username and password, the system verifies the credentials from the MySQL database. If the details are correct, a session is created and the user is redirected to the dashboard. If invalid details are entered, the system shows an error message. This prevents direct access to internal pages without login permission.

This module is essential in a Core PHP project because it demonstrates session management, secure page control, and access restrictions. It also improves the professionalism of the final-year project by giving it a real software application feel. Since every advanced web application requires secure access, this module becomes one of the most important components of the overall system architecture.

3. Device Management Module

The Device Management Module is used to manage the smart sticks or related devices registered in the system. Each smart stick can be treated as a separate device with its own identification number, model name, sensor setup, registration date, and current status. This helps administrators keep track of available devices, active devices, and maintenance requirements.

Through this module, the admin can add a new device, assign it to a user, edit device specifications, and update its condition. For example, the device record may contain details such as device code, owner name, sensor type, connectivity mode, battery status, and deployment status. This information is useful for maintaining multiple smart sticks under a single system.

In a final-year project, this module gives more depth because it shifts the project from a single-device demo to a manageable system. It shows that the project is scalable and capable of handling multiple units. It also supports future enhancements such as maintenance tracking, repair logs, and usage-based monitoring.

4. Sensor Data Logging Module

The Sensor Data Logging Module is one of the most important technical modules in the project. Its role is to store all incoming sensor readings from the smart obstacle detection stick. These readings may come from ultrasonic sensors, IR sensors, or water/pit sensors. The module records important values such as sensor type, detected distance, detection time, device ID, and reading status.

This module acts as the raw data storage section of the system. Every obstacle sensing event is captured and entered into the database, either through live API submission, simulated device data, or manual entry. The stored sensor data becomes the

foundation for later analytics, reports, and history tracking. Without this module, the system would only provide real-time alerts and would not support data analysis.

Academically, this module adds strong technical value to the project because it demonstrates how real-world device data can be collected and stored in a structured format. It also supports reporting and future machine learning possibilities. The module can be presented with table views, filter options, date search, and detailed sensor logs.

5. Obstacle Detection Records Module

The Obstacle Detection Records Module manages all confirmed obstacle events detected by the smart stick. While sensor logs may include raw readings, this module focuses specifically on meaningful obstacle records that require alerts or attention. It stores information such as obstacle type, obstacle distance, alert level, device used, date, time, and remarks.

The main aim of this module is to provide a clean and organized record of hazards identified during stick usage. It helps administrators understand how often obstacles occur, what kinds of hazards are common, and at what distances they are usually detected. This module can also categorize obstacles into near, medium, and far danger levels.

This module is useful during project demonstration because it clearly shows the practical output of the smart stick. Instead of only presenting sensor numbers, it presents safety-related detection events in a meaningful form. It also supports data filtering, editing, deletion, and report generation, making it a complete and useful functional module.

6. Alert Management Module

The Alert Management Module controls how alerts are generated, stored, and monitored in the system. When an obstacle is detected, the smart stick may respond through buzzer sound, vibration, or both. This module keeps a record of every alert generated, including alert type, obstacle reference, user/device, alert time, and response status.

This module is important because alerts are the direct assistance mechanism given to visually impaired users. The project becomes more realistic when it not only detects an obstacle but also records exactly what alert was triggered. This allows the admin to analyze alert effectiveness and track whether the warning mechanism is working properly.

From a software engineering point of view, this module enhances the completeness of the project. It connects hardware response with software history tracking. The admin can view past alerts, search by device or date, and examine how frequently each alert type is used. This information can later support improvements in warning design and user safety methods.

7. Analytics Dashboard Module

The Analytics Dashboard Module is the central visual component of the system. It displays summarized information collected from all modules in the form of cards, charts, graphs, and key performance indicators. This module gives a quick overview of total devices, total obstacle detections, total alerts, recent activity, and user statistics.

The dashboard transforms raw database records into meaningful visual insights. For example, it can show monthly obstacle counts, obstacle type distribution, device-wise activity comparison, and daily alert frequency. These visualizations help both the admin and project evaluators understand the performance of the system instantly without reading large tables of data.

This module greatly improves the quality of the final-year project because it demonstrates analytical thinking, reporting design, and dashboard-based monitoring. It also gives the software a modern and professional look. Using Bootstrap cards and chart libraries, this module can become one of the most attractive and presentation-friendly parts of the project.

8. Reports Generation Module

The Reports Generation Module is responsible for creating structured summaries from the stored project data. It allows the administrator to generate daily, weekly, monthly, or custom-date reports based on obstacle detections, alert history, device

activity, and user usage. These reports help in project review, performance study, and final documentation.

This module can provide downloadable or printable reports showing total detections, most frequent obstacle types, average sensing range, and alert statistics. It can also support filter-based reporting by device, user, or date range. This gives flexibility in analyzing the project at different levels.

For final-year evaluation, this module is very valuable because it shows how stored system data can be converted into official records and analytical outputs. It also improves the usefulness of the application beyond real-time operations. Reports make the system suitable for academic documentation, admin review, and future system assessment.

9. Feedback and Issue Tracking Module

The Feedback and Issue Tracking Module allows users or administrators to record comments, problems, suggestions, or maintenance-related issues associated with the smart stick. This module can store entries such as feedback title, message, issue type, submitted by, date, and resolution status. It improves project quality by adding a support and improvement mechanism.

This module is helpful when multiple users are involved or when device problems need to be monitored. For example, users can report false obstacle detections, delayed alerts, sensor failures, or battery-related issues. The admin can review these entries and mark them as pending, resolved, or under process.

In academic presentation, this module shows that the project is not only functional but also maintainable. It adds a real-world management perspective by providing a way to improve the system based on user observations and technical issues. This makes the project more complete and closer to an actual deployed application.

10. Settings and System Configuration Module

The Settings and System Configuration Module is used to manage the overall configuration of the application. It allows the admin to define project-level settings such as obstacle threshold distance, default alert type, system name, contact details, sensor preferences, and interface settings. This module ensures that the system can be customized without changing the source code directly.

The admin can use this module to update operational parameters according to project requirements. For example, the minimum detection distance can be modified, a preferred notification method can be selected, or the dashboard display name can be updated. This makes the software flexible and easier to manage.

This module adds professional value to the project because modern software applications usually include configurable settings. It also shows that the system is designed with usability and administration in mind. In a final-year Core PHP and MySQL project, the inclusion of a settings module reflects a more advanced and well-organized development approach.

4. IMPLEMENTATION

The implementation of the Smart Obstacle Detection Stick with Data Analytics System is carried out using Core PHP, MySQL, HTML, CSS, Bootstrap, and JavaScript. The project is designed as a web-based application that manages and analyzes obstacle detection data collected from the smart stick. The implementation focuses on creating a structured, modular, and user-friendly system that supports real-time-style logging, historical record management, and analytical reporting. The overall development process includes database design, frontend interface creation, backend logic development, module integration, testing, and report generation.

At the beginning of implementation, the project database is designed in MySQL with around ten interconnected tables. These tables store data related to administrators, users, registered devices, sensor logs, obstacle records, alerts, reports, feedback, settings, and activity history. Proper primary keys and foreign key relationships are used to maintain consistency and avoid duplicate or invalid data. Sample records are inserted into the tables so that the system can be demonstrated effectively during project review and presentation. The SQL file is included with the project to allow easy setup on any local server environment such as XAMPP, WAMP, or a VPS-based Apache server.

The frontend of the application is implemented using HTML, CSS, Bootstrap, and JavaScript to provide a professional dashboard interface. A left-side navigation menu is used to access all major modules such as Dashboard, User Management, Device Management, Sensor Logs, Obstacle Records, Alerts, Analytics, Reports, Feedback, and Settings. Bootstrap components such as cards, tables, buttons, forms, and modals are used to create a responsive and attractive user interface. This ensures that the application looks modern and remains easy to use across different screen sizes.

The backend logic is developed in Core PHP, where each module is handled using separate PHP files. This modular file structure makes the system easy to maintain and understand. Common files such as database connection, header, footer, sidebar, session check, and configuration are reused across all pages. CRUD operations are

implemented for major entities so that the administrator can add, edit, update, delete, and view records efficiently. For example, in the User Management module, the admin can create a new user account, update user details, deactivate accounts, or remove invalid entries. Similarly, in the Device Management module, each smart stick can be registered and linked with usage records.

The Authentication Module is implemented first to secure the system. Login and logout features are created using PHP sessions. When an administrator enters valid credentials, the system verifies them from the database and grants access to internal modules. Unauthorized users are redirected to the login page. This ensures that system data remains protected and only authorized users can manage the project records.

The Sensor Data Logging Module is implemented to store sensor readings either from simulated API input, manual form entry, or sample records. Every record includes fields such as device ID, sensor type, measured distance, detection date, time, and status. These readings are processed and, where necessary, converted into obstacle events. The Obstacle Detection Module then displays meaningful entries such as nearby wall detection, water hazard detection, or close object warning. The Alert Module stores the response generated for each event, such as vibration alert, buzzer alert, or combined warning.

The Analytics Dashboard Module is implemented by fetching summary values from the MySQL database using PHP queries and displaying them in cards and chart sections. JavaScript chart libraries or Bootstrap-based visual sections can be used to represent detection trends, alert frequency, device-wise activity, and monthly summaries. This module is particularly important because it visually demonstrates the analytical capability of the system and makes the project more attractive during final-year evaluation.

The Reports Module is implemented to generate date-wise and category-wise summaries. The administrator can view or print reports related to obstacle logs, alert

history, or device performance. Search and filter functionality is included to retrieve records based on date range, user, or device. This helps in detailed analysis and academic documentation. The Feedback Module is also implemented to capture user observations, issues, and maintenance remarks, which adds a real-world management dimension to the software.

Testing is an important part of implementation. Each module is individually tested for input validation, CRUD accuracy, session control, navigation flow, and database interaction. Integration testing is then carried out to verify that all modules work together correctly. For example, when a sensor log is added, it should reflect in obstacle records, alert history, and dashboard statistics where applicable. Sample data is used extensively to ensure that the project behaves like a live working system even in the absence of actual hardware integration.

4.1 SOFTWARE REQUIREMENTS

1. Operating System

- Windows 10 / 11 (for development)
- Linux (Ubuntu/CentOS) recommended for production server

2. Web Server

- Apache Server (with mod_rewrite enabled)

OR

- Nginx Server

3. Backend Technology

- PHP \geq 8.2
- Laravel Framework (Latest Version)

4. Database

- MySQL Server (5.7 or higher)

OR

- MariaDB

5. Frontend Technology

- HTML5
- CSS3
- JavaScript
- Bootstrap (for responsive UI design)
- Blade Template Engine (Laravel)

6. Required PHP Extensions

- PDO PHP Extension
- OpenSSL PHP Extension
- Mbstring PHP Extension
- Exif PHP Extension
- Fileinfo Extension
- XML PHP Extension
- Ctype PHP Extension
- JSON PHP Extension
- Tokenizer PHP Extension

- cURL PHP Extension

7. Additional Tools & Software

- Composer (Dependency Manager for PHP)
- Git (Version Control System)
- Node.js & NPM (for frontend asset compilation, optional)

8. Server Configuration

- Enable **mod_rewrite** (Apache)
- Enable **HTTPS (SSL Certificate)** for secure communication
- Configure **.env** file for database and app settings
- Set proper file permissions for storage and cache

9. Browser Compatibility

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Safari

10. Hosting Environment

- VPS Server / Cloud Hosting (AWS, DigitalOcean, etc.)
- Minimum recommended configuration:
 - 2+ CPU Cores
 - 4 GB RAM
 - 50 GB Storage

4.2 SYSTEM REQUIREMENTS

1. Hardware Requirements

Development System Requirements

These are the minimum hardware requirements for developing and testing the project:

- **Processor:** Intel Core i3 / i5 or higher
- **RAM:** 8 GB minimum
- **Hard Disk:** 256 GB SSD or higher
- **Monitor:** 14-inch or above
- **Keyboard and Mouse:** Standard input devices
- **Internet Connection:** Stable broadband connection

These specifications are sufficient for coding, database handling, local server testing, and UI development.

Server Requirements (VPS Hosting)

For live deployment, the application requires a **Virtual Private Server (VPS)** with the following configuration:

Component	Specification
Server Type	Virtual Private Server (VPS)
CPU	8 Core Processor
RAM	32 GB
Storage	300 GB NVMe SSD
Bandwidth	High-speed / Unlimited preferred
Operating System	Ubuntu / CentOS / AlmaLinux
Web Server	Apache or Nginx
Database Server	MariaDB
Control Panel	WHM / cPanel
Backup Support	Daily / Weekly Backup Recommended
SSL Certificate	Required for secure access

4.3 SAMPLE CODING

```
<?php
session_start();
require_once __DIR__ . '/config/database.php';
$error = "";
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $username = trim($_POST['username'] ?? "");
    $password = trim($_POST['password'] ?? "");

    $stmt = $pdo->prepare('SELECT * FROM admins WHERE username = ? LIMIT 1');
    $stmt->execute([$username]);
    $admin = $stmt->fetch();

    if ($admin && $password === $admin['password']) {
        $_SESSION['admin_id'] = $admin['id'];
        $_SESSION['admin_name'] = $admin['full_name'];
        header('Location: dashboard.php');
        exit;
    }
    $error = 'Invalid username or password';
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login - Smart Obstacle Detection Stick</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```

    <link href="assets/css/style.css" rel="stylesheet">
</head>
<body>
<div class="login-wrapper">
    <div class="card login-card shadow-lg p-4">
        <div class="card-body">
            <h2 class="text-center mb-2">Smart Obstacle Detection Stick</h2>
            <p class="text-center text-muted mb-4">Core PHP & MySQL Admin
Login</p>
            <?php if ($error): ?>
                <div class="alert alert-danger"><?php echo htmlspecialchars($error);
?></div>
            <?php endif; ?>
            <form method="post">
                <div class="mb-3">
                    <label class="form-label">Username</label>
                    <input type="text" name="username" class="form-control" required>
                </div>
                <div class="mb-3">
                    <label class="form-label">Password</label>
                    <input type="password" name="password" class="form-control"
required>
                </div>
                <button class="btn btn-primary w-100">Login</button>
            </form>
            <div class="mt-3 small text-muted text-center">Default: <strong>admin /
admin123</strong></div>
        </div>
    </div>
</div>
</body>

```

```
</html>
```

Dashboard

```
<?php include 'includes/header.php'; include 'includes/sidebar.php';

$tables = [
    'users', 'devices', 'sensors', 'locations', 'obstacle_logs', 'alerts', 'routes', 'maintenance_logs', 'feedback'];

$count = [];
foreach ($tables as $table) {
    $count[$table] = (int)$pdo->query("SELECT COUNT(*) FROM $table")->fetchColumn();
}

$chartData = $pdo->query("SELECT obstacle_type, COUNT(*) total FROM obstacle_logs GROUP BY obstacle_type ORDER BY total DESC LIMIT 5")->fetchAll();

$recentAlerts = $pdo->query("SELECT a.*, u.full_name, d.device_code FROM alerts a LEFT JOIN users u ON a.user_id=u.id LEFT JOIN devices d ON a.device_id=d.id ORDER BY a.alert_time DESC LIMIT 5")->fetchAll();

?>

<div class="row g-4 mb-4">
    <?php foreach ([
        'Users' => $count['users'],
        'Devices' => $count['devices'],
        'Sensors' => $count['sensors'],
        'Locations' => $count['locations'],
        'Obstacle Logs' => $count['obstacle_logs'],
        'Alerts' => $count['alerts']
    ] as $label => $value): ?>
        <div class="col-md-4 col-xl-2">
            <div class="card stat-card p-3">
```

```

        <div class="text-muted small"><?php echo $label; ?></div>
        <div class="display-6 fw-bold"><?php echo $value; ?></div>
    </div>
</div>
<?php endforeach; ?>
</div>
<div class="row g-4">
    <div class="col-lg-7">
        <div class="card shadow-sm">
            <div class="card-header bg-white fw-bold">Obstacle Analysis</div>
            <div class="card-body"><canvas id="obstacleChart"></canvas></div>
        </div>
    </div>
    <div class="col-lg-5">
        <div class="card shadow-sm">
            <div class="card-header bg-white fw-bold">Recent Alerts</div>
            <div class="card-body">
                <div class="table-responsive">
                    <table class="table table-bordered table-sm">
<thead><tr><th>User</th><th>Device</th><th>Severity</th><th>Time</th></tr></thead>
<tbody>
<?php foreach ($recentAlerts as $row): ?>
    <tr>
        <td><?php echo htmlspecialchars($row['full_name'] ?? 'N/A');
?></td>
        <td><?php echo htmlspecialchars($row['device_code'] ?? 'N/A');
?></td>
        <td><span class="badge bg-warning text-dark"><?php echo
htmlspecialchars($row['severity']); ?></span></td>

```



```

<?php include 'includes/header.php'; include 'includes/sidebar.php';
$id = $_GET['edit'] ?? null;
$editData = ['full_name'=>'', 'email'=>'', 'phone'=>'', 'gender'=>'', 'status'=>'Active'];
if ($id) {
    $stmt = $pdo->prepare('SELECT * FROM users WHERE id=?'); $stmt-
    >execute([$id]); $editData = $stmt->fetch() ?: $editData;
}
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $data = [trim($_POST['full_name']), trim($_POST['email']), trim($_POST['phone']),
    trim($_POST['gender']), trim($_POST['status'])];
    if (!empty($_POST['id'])) {
        $data[] = $_POST['id'];
        $stmt = $pdo->prepare('UPDATE users SET full_name=?, email=?, phone=?,
        gender=?, status=? WHERE id=?');
        $stmt->execute($data);
    } else {
        $stmt = $pdo->prepare('INSERT INTO users
        (full_name,email,phone,gender,status) VALUES (?,?,,?,?)');
        $stmt->execute($data);
    }
    header('Location: users.php'); exit;
}
if (isset($_GET['delete'])) { $stmt = $pdo->prepare('DELETE FROM users WHERE
id=?'); $stmt->execute($_GET['delete']); header('Location: users.php'); exit; }
$rows = $pdo->query('SELECT * FROM users ORDER BY id DESC')->fetchAll();
?>
<div class="row g-4">
    <div class="col-lg-4"><div class="card shadow-sm"><div class="card-header bg-
white fw-bold">User Form</div><div class="card-body">
        <form method="post">

```



```

        </tbody></table>
    </div></div></div>
</div>
<?php include 'includes/footer.php'; ?>

```

Routes

```

<?php include 'includes/header.php'; include 'includes/sidebar.php';
$locations=$pdo->query('SELECT id, location_name FROM locations ORDER BY
location_name')->fetchAll();
$Id=$_GET['edit']                ??                null;
$editData=['route_name'=>,"','start_location_id'=>,"','end_location_id'=>,"','distance_m'=>
>,"','safety_score'=>80,'status'=>'Active'];
if($Id){$stmt=$pdo->prepare('SELECT * FROM routes WHERE id=?');$stmt-
>execute([$Id]);$editData=$stmt->fetch()?:$editData;}
if($_SERVER['REQUEST_METHOD']=='POST'){

$data=[trim($_POST['route_name']),$_POST['start_location_id'],$_POST['end_locati
on_id'],trim($_POST['distance_m']),trim($_POST['safety_score']),trim($_POST['statu
s'])];

    if(!empty($_POST['id'])){$data[]=$_POST['id'];$stmt=$pdo->prepare('UPDATE
routes SET route_name=?, start_location_id=?, end_location_id=?, distance_m=?,
safety_score=?, status=? WHERE id=?');}

    else{$stmt=$pdo->prepare('INSERT                INTO                routes
(route_name,start_location_id,end_location_id,distance_m,safety_score,status)
VALUES (?,?,?,?,?,?)');}

    $stmt->execute($data); header('Location: routes.php'); exit;
}
if(isset($_GET['delete'])){$pdo->prepare('DELETE FROM routes WHERE id=?')-
>execute($_GET['delete']); header('Location: routes.php'); exit;}

```

```
$rows=$pdo->query('SELECT r.*, s.location_name AS start_loc, e.location_name AS
end_loc FROM routes r LEFT JOIN locations s ON r.start_location_id=s.id LEFT JOIN
locations e ON r.end_location_id=e.id ORDER BY r.id DESC')->fetchAll();
```

```
?>
```

```
<div class="row g-4"><div class="col-lg-4"><div class="card"><div class="card-
header bg-white fw-bold">Route Form</div><div class="card-body"><form
method="post"><input type="hidden" name="id" value="<?php echo
htmlspecialchars($editData['id'] ?? ''); ?>"><input class="form-control mb-3"
name="route_name" placeholder="Route Name" value="<?php echo
htmlspecialchars($editData['route_name']); ?>"><select class="form-select mb-3"
name="start_location_id"><?php foreach($locations as $l): ?><option value="<?php
echo $l['id']; ?>" <?php if($l['id']==($editData['start_location_id']??'')) echo 'selected';
?><?php echo htmlspecialchars($l['location_name']); ?></option><?php endforeach;
?></select><select class="form-select mb-3" name="end_location_id"><?php
foreach($locations as $l): ?><option value="<?php echo $l['id']; ?>" <?php
if($l['id']==($editData['end_location_id']??'')) echo 'selected'; ?><?php echo
htmlspecialchars($l['location_name']); ?></option><?php
endforeach;
?></select><input class="form-control mb-3" name="distance_m"
placeholder="Distance (m)" value="<?php echo
htmlspecialchars((string)$editData['distance_m']); ?>"><input class="form-control
mb-3" name="safety_score" type="number" min="0" max="100" placeholder="Safety
Score" value="<?php echo htmlspecialchars((string)$editData['safety_score']);
?>"><select class="form-select mb-3"
name="status"><option>Active</option><option
<?php
if($editData['status']=== 'Blocked') echo 'selected'; ?>>Blocked</option><option
<?php
if($editData['status']=== 'Under Review') echo 'selected'; ?>>Under
Review</option></select><button class="btn btn-primary w-100">Save
Route</button></form></div></div></div><div class="col-lg-8"><div
class="card"><div class="card-header bg-white fw-bold">Safe Routes</div><div
class="card-body table-responsive"><table class="table table-
bordered"><thead><tr><th>Route</th><th>Start</th><th>End</th><th>Distance</t
```

```

h><th>Score</th><th>Action</th></tr></thead><tbody><?php foreach($rows as $r):
?><tr><td><?php echo htmlspecialchars($r['route_name']); ?></td><td><?php echo
htmlspecialchars($r['start_loc']);           ?></td><td><?php           echo
htmlspecialchars($r['end_loc']); ?></td><td><?php echo (int)$r['distance_m']; ?>
m</td><td><?php echo (int)$r['safety_score']; ?></td><td><a class="btn btn-sm btn-
success" href="?edit=<?php echo $r['id']; ?>">Edit</a> <a class="btn btn-sm btn-
danger" onclick="return confirm('Delete route?')" href="?delete=<?php echo $r['id'];
?>">Delete</a></td></tr><?php           endforeach;
?></tbody></table></div></div></div></div>
<?php include 'includes/footer.php'; ?>

```

```

<?php include 'includes/header.php'; include 'includes/sidebar.php';
$users = $pdo->query('SELECT id, full_name FROM users ORDER BY full_name')-
>fetchAll();
$id = $_GET['edit'] ?? null;
$editData = [
    'user_id'=>,$_GET['user_id'],
    'device_code'=>,$_POST['device_code'],
    'model_name'=>,$_POST['model_name'],
    'firmware_version'=>,$_POST['firmware_version'],
    'battery_level'=>100,
    'status'=>'Active'];
if ($id) { $stmt = $pdo->prepare('SELECT * FROM devices WHERE id=?'); $stmt-
>execute([$id]); $editData = $stmt->fetch() ?: $editData; }
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $data = [$_POST['user_id'], trim($_POST['device_code']),
    trim($_POST['model_name']), trim($_POST['firmware_version']),
    $_POST['battery_level'], trim($_POST['status'])];
    if (!empty($_POST['id'])) { $data[]=$_POST['id']; $stmt=$pdo->prepare('UPDATE
devices SET user_id=?, device_code=?, model_name=?, firmware_version=?,
battery_level=?, status=? WHERE id=?'); }
    else { $stmt=$pdo->prepare('INSERT INTO devices
(user_id,device_code,model_name,firmware_version,battery_level,status) VALUES
(?,?,?,?,?,?)'); }
}

```

```

$stmt->execute($data); header('Location: devices.php'); exit;
}
if (isset($_GET['delete'])) { $pdo->prepare('DELETE FROM devices WHERE id=?')-
>execute($_GET['delete']); header('Location: devices.php'); exit; }
$rows = $pdo->query('SELECT d.*, u.full_name FROM devices d LEFT JOIN users u
ON d.user_id=u.id ORDER BY d.id DESC')->fetchAll();
?>
<div class="row g-4"><div class="col-lg-4"><div class="card"><div class="card-
header bg-white fw-bold">Device Form</div><div class="card-body"><form
method="post">
<input type="hidden" name="id" value="<?php echo htmlspecialchars($editData['id']
?? "); ?>">
<select class="form-select mb-3" name="user_id" required><?php foreach($users as
$u): ?><option value="<?php echo $u['id']; ?>" <?php
if($u['id']==($editData['user_id']??)) echo 'selected'; ?><?php echo
htmlspecialchars($u['full_name']); ?></option><?php endforeach; ?></select>
<input class="form-control mb-3" name="device_code" placeholder="Device Code"
value="<?php echo htmlspecialchars($editData['device_code']); ?>" required>
<input class="form-control mb-3" name="model_name" placeholder="Model Name"
value="<?php echo htmlspecialchars($editData['model_name']); ?>" required>
<input class="form-control mb-3" name="firmware_version" placeholder="Firmware
Version" value="<?php echo htmlspecialchars($editData['firmware_version']); ?>">
<input class="form-control mb-3" type="number" min="0" max="100"
name="battery_level" placeholder="Battery" value="<?php echo
htmlspecialchars((string)$editData['battery_level']); ?>">
<select class="form-select mb-3" name="status"><option>Active</option><option
<?php if($editData['status']=== 'Inactive') echo 'selected'; ?>>Inactive</option><option
<?php if($editData['status']=== 'Maintenance') echo 'selected';
?>>Maintenance</option></select>
<button class="btn btn-primary w-100">Save
Device</button></form></div></div></div>

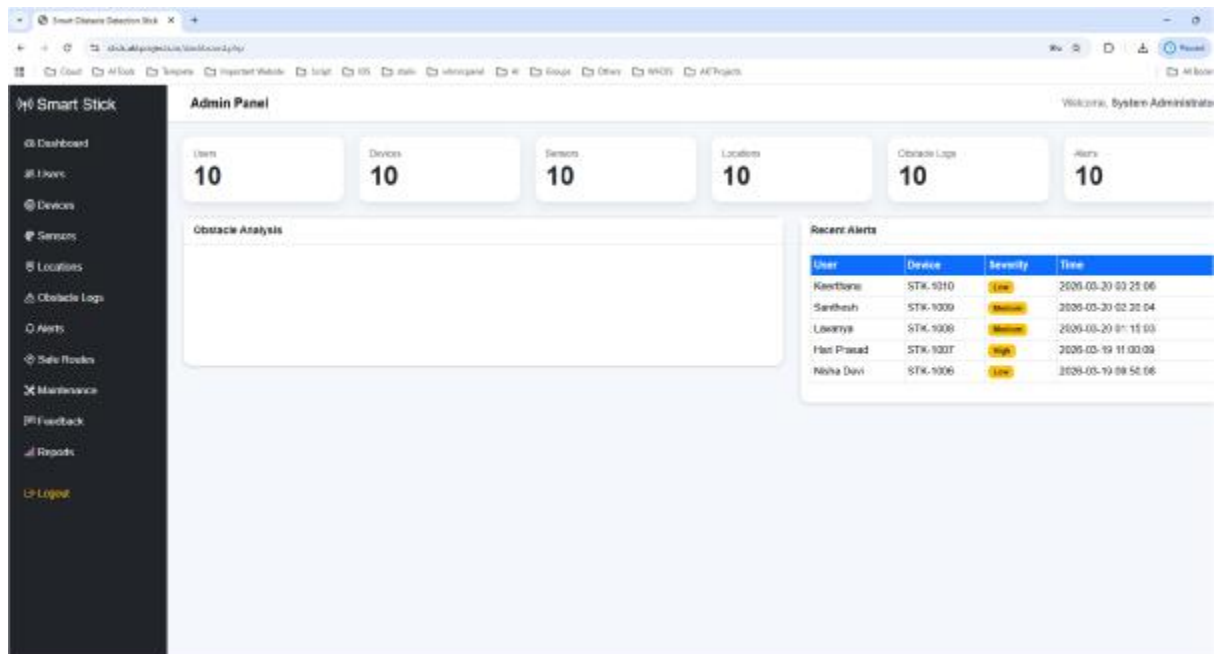
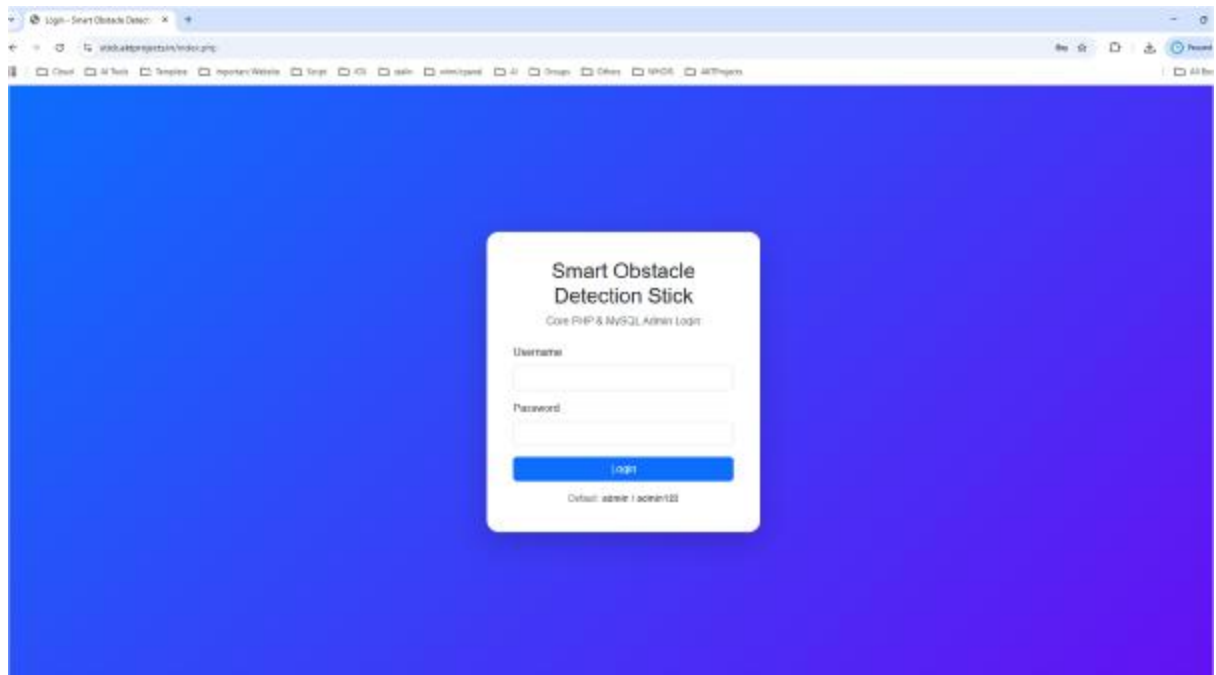
```

```

<div class="col-lg-8"><div class="card"><div class="card-header bg-white fw-
bold">Devices</div><div class="card-body table-responsive"><table class="table
table-
bordered"><thead><tr><th>Code</th><th>User</th><th>Model</th><th>Battery</t
h><th>Status</th><th>Action</th></tr></thead><tbody><?php foreach($rows as $r):
?><tr><td><?php echo htmlspecialchars($r['device_code']); ?></td><td><?php echo
htmlspecialchars($r['full_name']);          ?></td><td><?php          echo
htmlspecialchars($r['model_name']); ?></td><td><?php echo (int)$r['battery_level'];
?>%</td><td><?php echo htmlspecialchars($r['status']); ?></td><td><a class="btn
btn-sm btn-success" href="?edit=<?php echo $r['id']; ?>">Edit</a> <a class="btn btn-
sm btn-danger" href="?delete=<?php echo $r['id']; ?>" onclick="return confirm('Delete
device?')">Delete</a></td></tr><?php          endforeach;
?></tbody></table></div></div></div></div>
<?php include 'includes/footer.php'; ?>

```

4.4. SCREEN SHOT



Smart Stick Admin Panel - User Management

Dashboard | Users | Devices | Sensors | Locations | Outside Logs | Alerts | Safe Routes | Maintenance | Feedback | Reports | Logout

Welcome, System Administrator

User Form

Full Name:

Email:

Phone:

Male Female

Active

[Add User](#)

Users List

ID	Name	Email	Phone	Status	Action
10	Keerthana	keerthana@example.com	9979543218	Active	Edit Delete
9	Santhosh	santhosh@example.com	9979543218	Inactive	Edit Delete
8	Levanya	levanya@example.com	9979543217	Active	Edit Delete
7	Hari Prasad	hari@example.com	9979543216	Active	Edit Delete
6	Nisha Devi	nisha@example.com	9979543215	Active	Edit Delete
5	Pavani Raj	pavani@example.com	9979543214	Active	Edit Delete
4	Meena Lakshmi	meena@example.com	9979543213	Inactive	Edit Delete
3	Karthik Raja	karthik@example.com	9979543212	Active	Edit Delete
2	Divya Sri	divya@example.com	9979543211	Active	Edit Delete
1	Anun Kumar	anun@example.com	9979543210	Active	Edit Delete

Smart Stick Admin Panel - Device Management

Dashboard | Users | Devices | Sensors | Locations | Outside Logs | Alerts | Safe Routes | Maintenance | Feedback | Reports | Logout

Welcome, System Administrator

Device Form

User: Anun Kumar

Device Code:

Model Name:

Firmware Version:

ID:

Active

[Save Device](#)

Devices

Code	User	Model	Battery	Status	Action
STK-1010	Keerthana	Mobility Asset	95%	Active	Edit Delete
STK-1009	Santhosh	Mobility Asset	98%	Active	Edit Delete
STK-1008	Levanya	GridSense Mini	34%	Maintenance	Edit Delete
STK-1007	Hari Prasad	GridSense Mini	70%	Active	Edit Delete
STK-1006	Nisha Devi	PathFinder Lite	50%	Inactive	Edit Delete
STK-1005	Pavani Raj	PathFinder Lite	99%	Active	Edit Delete
STK-1004	Meena Lakshmi	StreetNav Pro	40%	Maintenance	Edit Delete
STK-1003	Karthik Raja	StreetNav Pro	91%	Active	Edit Delete
STK-1002	Divya Sri	VisionSolo X1	72%	Active	Edit Delete
STK-1001	Anun Kumar	VisionSolo X1	85%	Active	Edit Delete

Smart Obstacle Detection Stick

id:skidproject@vishv.edu

Dashboard | Users | Devices | Sensors | Locations | Obstacle Logs | Alerts | Safe Routes | Maintenance | Feedback | Reports | Logout

Welcome, System Administrator

Admin Panel

Obstacle Log Form

STK-1001

Admin Block Lobby

Wall

Distance (cm)

Medium

2020-03-21 11:05:15

Save Log

Obstacle Logs

Device	Location	Type	Distance	Severity	Action
STK-1010	Admin Block Lobby	Person	30 cm	Low	Edit Delete
STK-1009	Library Corridor	Table	47 cm	Medium	Edit Delete
STK-1008	Lab Passage	Wall	52 cm	Medium	Edit Delete
STK-1007	Caravan Side Path	Barrier	25 cm	High	Edit Delete
STK-1006	Auditorium Hall	Chair	60 cm	Low	Edit Delete
STK-1005	Hostel Walkway	Pole	38 cm	Medium	Edit Delete
STK-1004	Bus Stop Entrance	Pit	15 cm	High	Edit Delete
STK-1003	Parking Entrance	Vehicle	70 cm	Low	Edit Delete
STK-1002	Science Block Staircase	Star Edge	20 cm	High	Edit Delete
STK-1001	College Main Gate	Wall	45 cm	Medium	Edit Delete

Smart Obstacle Detection Stick

id:skidproject@vishv.edu

Dashboard | Users | Devices | Sensors | Locations | Obstacle Logs | Alerts | Safe Routes | Maintenance | Feedback | Reports | Logout

Welcome, System Administrator

Admin Panel

Alert Form

Arun Kumar

STK-1001

Vibration

Medium

2020-03-21 11:06:25

Sent

Save Alert

Alerts

User	Device	Type	Severity	Status	Action
Keerthana	STK-1010	Buzzer	Low	Sent	Edit Delete
Ranjith	STK-1009	Voice	Medium	Sent	Edit Delete
Lanara	STK-1008	Vibration	Medium	Sent	Edit Delete
Hari Prasad	STK-1007	Buzzer	High	Sent	Edit Delete
Nishu Devi	STK-1006	Voice	Low	Failed	Edit Delete
Prasanna Raj	STK-1005	Vibration	Medium	Sent	Edit Delete
Meena Lakshmi	STK-1004	SOS	High	Pending	Edit Delete
Karthik Raja	STK-1003	Buzzer	Low	Sent	Edit Delete
Dhivy Sri	STK-1002	Voice	High	Sent	Edit Delete
Arun Kumar	STK-1001	Vibration	Medium	Sent	Edit Delete

Smart Stick Admin Panel - Route Form

Route Name: Admin Block Lobby

Admin Block Lobby

Distance (m): 60

Action: [Save Route](#)

Safe Routes

Route	Start	End	Distance	Score	Action
Hotel to Auditorium	Hotel Walkway	Auditorium Hall	270 m	78	Edit Delete
Admin Block to Canteen	Admin Block Lobby	Canteen Side Path	250 m	84	Edit Delete
Bus Stop to Gate	Bus Stop Entrance	College Main Gate	200 m	75	Edit Delete
Lab to Staircase	Lab Passage	Science Block Staircase	95 m	58	Edit Delete
Auditorium to Library	Auditorium Hall	Library Corridor	140 m	87	Edit Delete
Parking to Bus Stop	Parking Entrance	Bus Stop Entrance	210 m	81	Edit Delete
Gate to Admin Block	College Main Gate	Admin Block Lobby	160 m	88	Edit Delete
Hotel to Canteen	Hotel Walkway	Canteen Side Path	250 m	80	Edit Delete
Library to Lab	Library Corridor	Lab Passage	120 m	86	Edit Delete
Gate to Library	College Main Gate	Library Corridor	180 m	82	Edit Delete

Smart Stick Admin Panel - Maintenance Form

STK-1001

21-03-2028

Issue Found

Action Taken

Technician Name

Completed

[Save Maintenance](#)

Maintenance Logs

Device	Date	Issue	Technician	Status	Action
STK-1010	2028-03-10	Speaker low volume	Muthu	Completed	Edit Delete
STK-1009	2028-03-08	Rx issue found	Ramesh	Completed	Edit Delete
STK-1008	2028-03-08	Firmware lag	Kavin	Completed	Edit Delete
STK-1007	2028-03-07	Onp vibration weak	Aru	Completed	Edit Delete
STK-1006	2028-03-06	IR false alerts	Suresh	Pending	Edit Delete
STK-1005	2028-03-05	Water sensor noise	Muthu	Completed	Edit Delete
STK-1004	2028-03-04	Battery draining fast	Ramesh	In Progress	Edit Delete
STK-1003	2028-03-03	GPS delay	Kavin	Completed	Edit Delete
STK-1002	2028-03-02	Loose buzzer wiring	Suresh	Completed	Edit Delete
STK-1001	2028-03-01	Incor calibration drift	Ramesh	Completed	Edit Delete

Smart Obstacle Detection Stick

100x100x100px/feedback.php

Cloud AI Tool Image Importer Webcam Sops IoT pin vhd/pin AI Group Other AWS IoTThings All Systems

Smart Stick

- Dashboard
- Users
- Devices
- Sensors
- Locations
- Obstacle Logs
- Alerts
- Safe Routes
- Maintenance
- Feedback
- Reports
- Logout

Admin Panel

Welcome, System Administrator

Feedback Form

Alan Kumar

5

Comments

2025-03-21 11:07:27

Reviewed

View Feedback

User Feedback

User	Rating	Comments	Status	Action
Koethana	4.5	System is easy to use and maintain	Reviewed	View Delete
Sarthosh	5.0	Accurate in crowded places	Reviewed	View Delete
Lavanya	4.0	Dashboard reports are useful	Reviewed	View Delete
Hari Prasad	5.0	Vibration feedback works really well	Reviewed	View Delete
Nisha Devi	2.0	Sometimes gives false warning	Closed	View Delete
Praveen Raj	4.5	Good support in outdoor paths	Reviewed	View Delete
Meena Lakshmi	3.0	Needs better pit detection accuracy	Pending	View Delete
Karthik Raja	5.0	Battery life is excellent	Reviewed	View Delete
Divya Sri	4.0	Voice alert is clear and useful	Reviewed	View Delete
Arun Kumar	5.0	Very helpful near walls and corridors	Reviewed	View Delete

Smart Obstacle Detection Stick

100x100x100px/reports.php

Cloud AI Tool Image Importer Webcam Sops IoT pin vhd/pin AI Group Other AWS IoTThings All Systems

Smart Stick

- Dashboard
- Users
- Devices
- Sensors
- Locations
- Obstacle Logs
- Alerts
- Safe Routes
- Maintenance
- Feedback
- Reports
- Logout

Admin Panel

Welcome, System Administrator

Average Feedback Rating

4.10

Average Device Battery

71.60%

Top Risk Locations

Users with Most Alerts

5. CONCLUSION

The Smart Obstacle Detection Stick with Data Analytics System presents an effective and innovative solution aimed at improving the mobility, safety, and independence of visually impaired individuals. By integrating sensing technology with a web-based data management and analytics platform, the project successfully transforms a traditional walking aid into an intelligent assistive system. The implementation demonstrates how real-time obstacle detection combined with structured data storage and analysis can significantly enhance user awareness and decision-making.

Throughout the project, various modules such as user management, device tracking, sensor data logging, obstacle detection, alert management, analytics dashboard, and report generation have been developed and integrated into a unified system. These modules work together to provide both immediate assistance through alerts and long-term insights through data analysis. The use of Core PHP and MySQL ensures that the system remains simple, cost-effective, and suitable for academic environments, while still maintaining a professional structure and interface.

One of the key strengths of this project is its ability to collect and analyze data at multiple levels. Instead of limiting the system to real-time obstacle detection, it extends functionality by storing historical records and generating meaningful reports. This analytical capability helps identify patterns such as frequent obstacles, high-risk conditions, and system performance trends. As a result, the project not only improves navigation for users but also provides valuable information for system evaluation and future improvements.

The modular architecture of the system makes it scalable and flexible for future enhancements. Features such as GPS tracking, IoT-based real-time communication, voice assistance, and AI-based predictive analysis can be easily integrated into the existing framework. This adaptability ensures that the project remains relevant in the context of modern technological advancements and real-world applications.

From an academic perspective, the project successfully demonstrates important concepts such as database design, CRUD operations, session management, modular programming, data visualization, and system integration. It also highlights the practical application of web technologies in solving real-world problems. The professional dashboard interface, structured modules, and analy

REFERENCES

1. World Health Organization – *World Report on Vision*, 2019. Provides global statistics and insights on visual impairment and the need for assistive technologies.
2. National Institute of Standards and Technology – Research on sensor technologies and embedded systems. Useful for understanding ultrasonic and infrared sensing mechanisms.
3. IEEE – IEEE Xplore Digital Library. Various research papers on smart walking sticks and assistive technologies for visually impaired individuals.
4. International Journal of Advanced Research in Computer Science – Articles on IoT-based smart stick systems and obstacle detection techniques.
5. Elsevier – ScienceDirect Database. Research papers related to embedded systems, sensor networks, and data analytics.
6. PHP Official Documentation – <https://www.php.net>
Used for backend development and server-side scripting.
7. MySQL Official Documentation – <https://www.mysql.com>
Used for database design, storage, and data retrieval.
8. Bootstrap Documentation – <https://getbootstrap.com>
Used for designing responsive and professional user interfaces.
9. XAMPP by Apache Friends – <https://www.apachefriends.org>
Used as a local development environment for running PHP and MySQL.
10. Arduino Official Website – <https://www.arduino.cc>
Reference for hardware implementation and sensor integration.
11. Google Scholar – <https://scholar.google.com>
Used to explore research papers related to smart assistive devices and obstacle detection systems.
12. Research Paper: “Smart Walking Stick for Visually Impaired People Using Ultrasonic Sensors,” IEEE Conference Publications.
Discusses obstacle detection using ultrasonic technology.

13. Research Paper: “IoT-Based Smart Assistive System for Blind People,”
International Journal Publications.

Explains integration of IoT with assistive devices.

14. Research Paper: “Design and Implementation of Smart Blind Stick,”
International Conference on Engineering and Technology.

Covers system design, sensor usage, and alert mechanisms.

15. Books:

- “Embedded Systems: Architecture, Programming and Design” by Raj Kamal
- “Web Technologies: HTML, CSS, JavaScript, PHP, MySQL” by Black Book Publications