Vaultify

IoT-Enabled Smart Vault with RFID, Motion Detection, AI, and Cloud Alerts

Github : https://github.com/SH-Nihil-Mukkesh-25/Vaultify.git

Abstract

Vaultify is a comprehensive smart security system that combines ESP32/Arduino hardware, motion detection sensors, RFID-based authentication, servo-controlled locking, and cloud-connected SMS alerts via Twilio. It integrates AI-powered analytics for real-time monitoring and pattern recognition. The system demonstrates a hybrid model where IoT hardware seamlessly interacts with backend and frontend software, providing automated alerts, logging, and intelligent insights. Vaultify offers a scalable, low-cost, and customizable approach to personal, academic, and small-business security.

Introduction

In the modern world, security threats extend beyond digital systems into physical environments. Traditional safes and vaults often lack intelligent monitoring and notification capabilities. Vaultify addresses this gap by combining IoT-enabled hardware (ESP32, RFID, MPU6050) with cloud and AI services, allowing users to monitor, control, and analyze security events in real-time. This project bridges physical security with intelligent digital insights.

Problem Statement

Traditional vaults and security systems face the following limitations:

- Weak or easily bypassed authentication mechanisms
- Lack of integration between physical and digital layers
- No real-time monitoring or instant notifications
- Limited logging, auditing, and analysis capabilities

Vaultify overcomes these challenges by integrating RFID access control, motion-based theft detection, servo-controlled locking, SMS alerts, and AI-driven insights into a single unified system.

Objectives

- Design a hybrid smart vault system integrating hardware and software security.
- Implement secure RFID-based authentication.
- Detect unauthorized access attempts using the MPU6050 accelerometer.
- Send real-time SMS alerts via Twilio over Wi-Fi.
- Provide automated servo-based door locking/unlocking.
- Enable live system monitoring via LCD and a web dashboard.
- Integrate AI for pattern recognition, trend analysis, and natural language query capabilities.

System Architecture

Vaultify follows a multi-layered security workflow:

User Input → RFID Authentication → MPU6050 Motion Detection → ESP32 Processing → Servo Motor Lock/Unlock → LCD Display & Buzzer Feedback → Wi-Fi SMS Alerts → AI Analysis & Dashboard

Circuit Diagram:
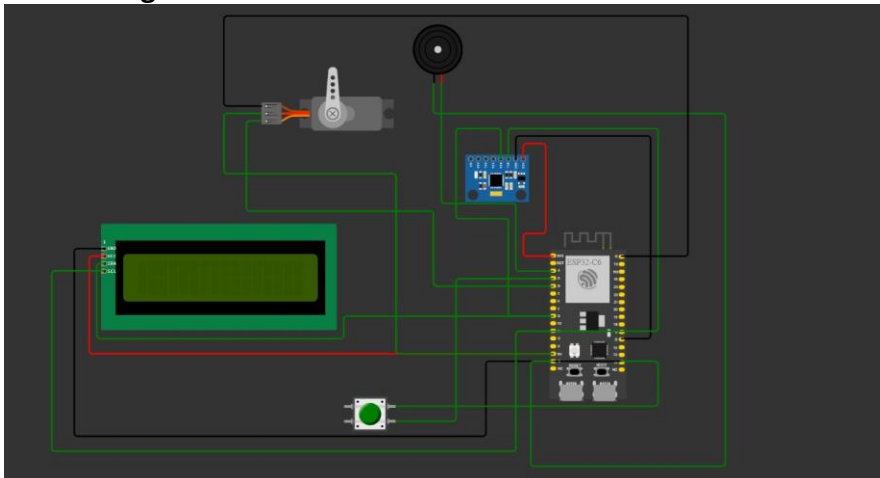
Methodology

1. Authentication (RFID)

Authorized RFID cards are stored in the system. On scanning, the ESP32 verifies the UID:

- **Valid Card:** Door unlocks, LCD and dashboard update, event logged.
- **Invalid Card:** Alarm triggers, SMS alert sent, AI logs suspicious activity.

---

2. Theft Detection (MPU6050)

The MPU6050 accelerometer monitors vault motion. Any abnormal tilt or vibration beyond a defined threshold triggers:

- Audible alarm via buzzer
- Visual alert on LCD
- Instant SMS notification via Twilio
- Event logged for AI analysis

---

3. Locking Mechanism (Servo Motor)

The servo motor locks/unlocks the vault:

- **Auto-Lock:** Configurable timeout ensures the vault never remains open.
- **Manual Unlock:** Triggered via authorized RFID or backend command.
- **Adaptive Locking:** Servo positions fine-tuned to vault mechanism.

---

4. Alerts & Logging (Twilio SMS)

Twilio's API enables:

- Instant alerts for valid/invalid access, motion detection, and auto-lock events
- Retry-safe message delivery in case of network instability
- Logging of all security events for AI-powered analysis

---

5. User Interface (LCD & Buzzer)

The system provides immediate feedback:

- **LCD Display:** Shows system status, lock state, and active alarms
- **Buzzer Feedback:**
    - Single beep → Successful action
    - Triple beep → Failed or suspicious activity

---

6. AI-Powered Analysis

Vaultify leverages Google Gemini AI for:

- **Security Insights:** "What events occurred today?"
- **Trend Analysis:** "How many invalid attempts this week?"
- **Threat Assessment:** Detect suspicious activity patterns
- **Natural Language Queries:** Ask plain-English questions about the system
- **Pattern Recognition:** Identify repeated access attempts, motion anomalies, and vulnerabilities

---

Hardware Requirements

Component	Quantity	Purpose
ESP32 DevKit	1x	Main Controller
MPU6050	1x	Motion Detection
MFRC522 RFID Module	1x	Access Control
16x2 I2C LCD	1x	Status Display
SG90 Servo	1x	Lock Mechanism
Buzzer	1x	Audio Alerts
RFID Cards	2-5x	Access Keys

Software Requirements

Arduino IDE Libraries

- LiquidCrystal_I2C
- MPU6050_light
- MFRC522
- ESP32Servo

Python Backend Dependencies

- fastapi, uvicorn
- langchain, langchain-google-genai, langchain-community
- python-dotenv, requests

AI Integration

- Gemini AI key setup via .env file
- Real-time question answering and trend analysis via API endpoints

---

Features and Functionalities

- RFID-based secure access
- Motion detection with alarm and SMS notification
- Auto-lock after configurable timeout
- Real-time LCD and dashboard display
- AI-powered analysis, pattern recognition, and natural language querying
- Full logging for monitoring and historical analysis
- Retry-safe communication for critical events

Use Cases

- Personal vaults and lockers
- Academic projects and IoT demonstrations
- Small business secure entry points
- Smart home security solutions

Limitations

- Requires Wi-Fi for SMS alerts
- Power consumption from servo + Wi-Fi modules
- LCD interface is basic; no mobile app yet
- RFID cards can potentially be cloned without additional security
- Logging is file-based, not centralized in a database

Future Scope

- Mobile app integration for remote monitoring
- Cloud-based centralized logging
- Multi-factor authentication (RFID + PIN/Fingerprint)
- Camera module for visual evidence
- Battery backup with tamper detection
- Encrypted storage for sensitive data
- AI-assisted predictive alerts based on access patterns

Conclusion

Vaultify demonstrates a unique integration of IoT hardware and cloud services to create a smart vault system. It successfully combines RFID, motion detection, servo locking, and SMS alerts to deliver a reliable, low-cost, and scalable solution for physical security. This prototype provides a strong foundation for both academic learning and future commercial deployments.

Group Members

Dakshin Raj P   - CB.SC.U4CSE24**534**

Nihil Mukkesh SH   - CB.SC.U4CSE24**531**

Sharvesh C   - CB.SC.U4CSE24**547**

Prawin Aadhithya AM   - CB.SC.U4CSE24**538**