
Smart Locker System with ESP32

An IoT-enabled smart locker system using ESP32, offering secure, remote-controlled access and real-time status monitoring via a mobile application

Department of Computer Science and Engineering
Presented by : Yash Gupta

Background

- Traditional lockers rely on **mechanical keys or combinations** and are widely used in various settings like schools and offices.
- These lockers provide basic storage but lack smart features such as **remote access** and **real-time monitoring**.



Problem Statement

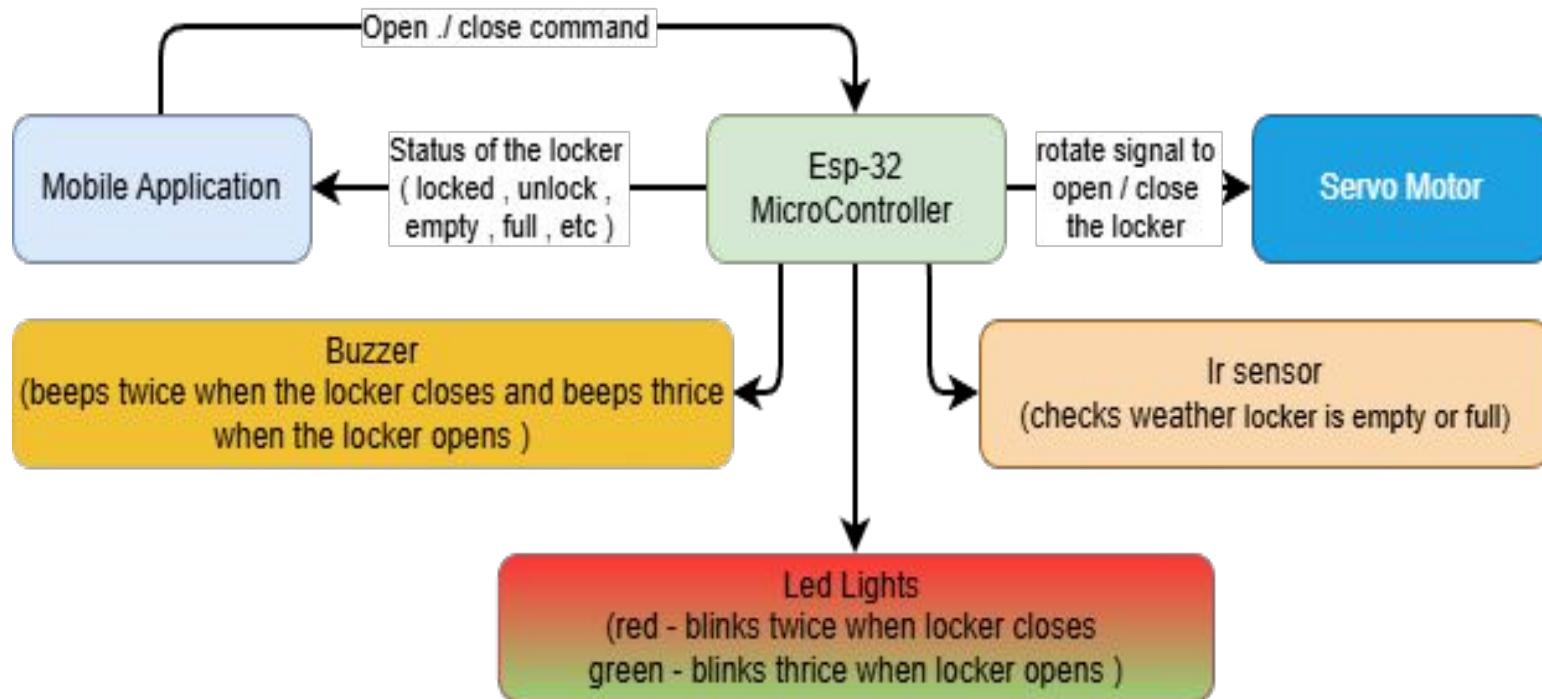
- Traditional lockers suffer from **key management** issues, such as **lost or stolen keys** causing access delays and security risks.
- **Lack of access control** and audit trails makes it difficult to track locker usage.
- Manual management is **time-consuming and prone** to errors.
- Existing lockers do not provide **remote operation** or **status updates**.
- **Maintenance costs** and **user inconvenience** are considerable for traditional systems.

Objectives

- Enable **remote locker control** via a dedicated mobile application.
- Provide **real-time locker status** monitoring including open/closed and content presence.
- Implement **secure user authentication** to restrict locker access.
- **Store Wi-Fi** credentials persistently in the ESP32 for seamless connectivity.
- Design a **user-friendly interface** for locker operation and status checking.
- Ensure reliable **feedback via LEDs and buzzer** for locker actions.



System Architecture



System Architecture

- **ESP32 Microcontroller:** Central control unit hosting the web server, handling Wi-Fi connectivity, and controlling peripherals.
- **Servo Motor:** Acts as the physical locking/unlocking mechanism for the locker door.
- **IR Sensor:** Detects presence or absence of contents inside the locker (content detection).
- **LEDs and Buzzer:** Provide visual and audio feedback for locker status and actions (e.g., locked/unlocked notifications).
- **Mobile App (React Native):** User interface for remote locker control, status checks, and authentication.
- **Wi-Fi Network:** Enables communication between the ESP32 and the mobile app for remote access.

Hardware implementation

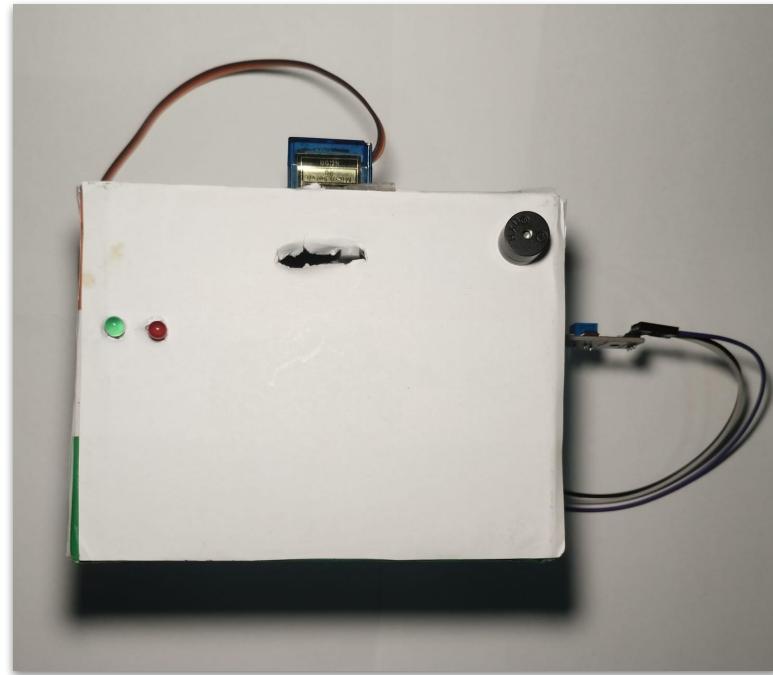
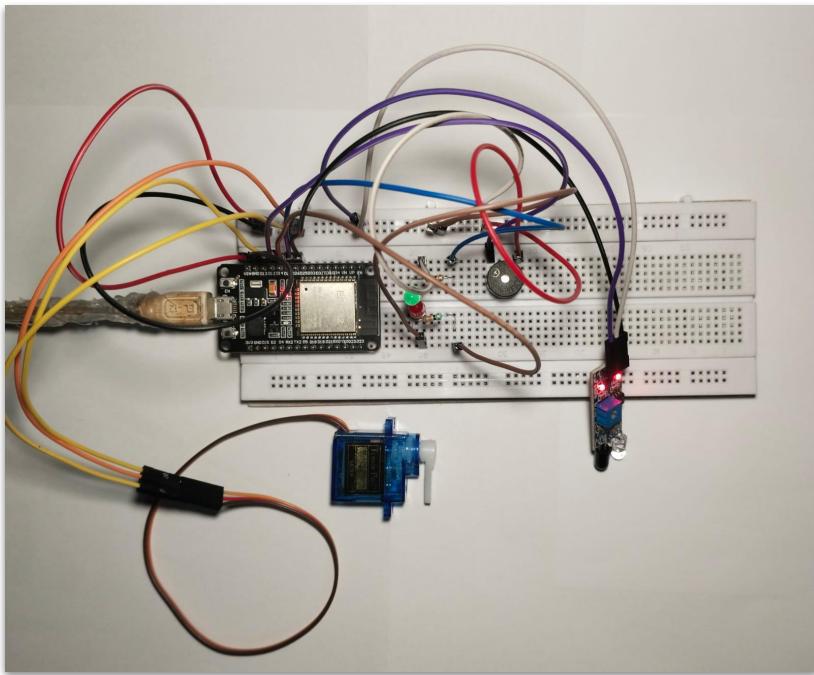
ESP32 Board :

- NodeMCU ESP32 Development Board
- Handles processing, Wi-Fi, and peripheral control

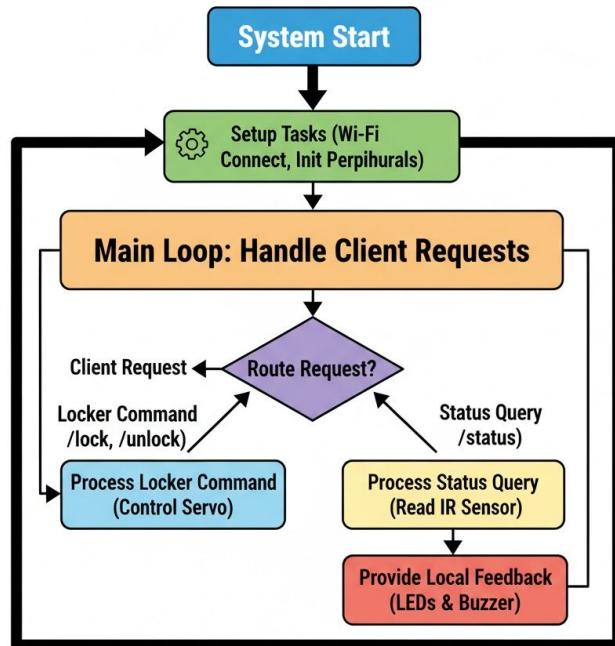
Pin Connections :

- Servo Motor connected to pin 13 (controls locker lock mechanism)
- Buzzer connected to pin 12 (provides audio feedback)
- Red LED connected to pin 14 (indicates locker closed)
- Green LED connected to pin 27 (indicates locker open)
- IR Sensor connected to pin 26 (detects content presence inside locker)

Hardware Setup

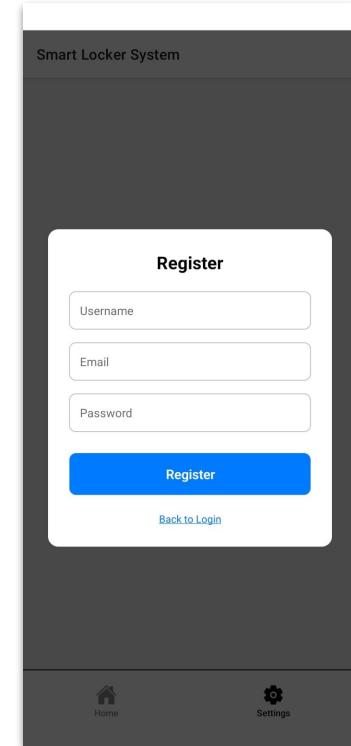
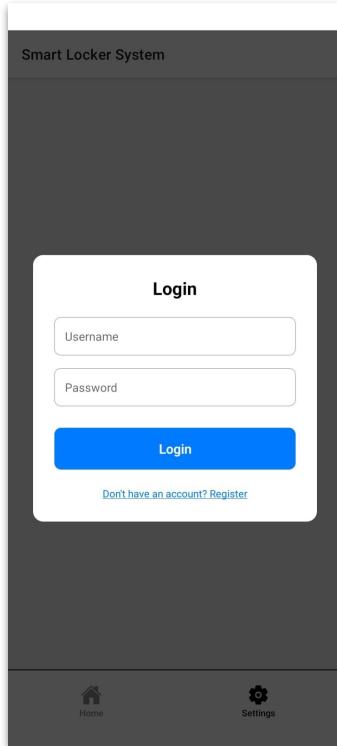


Software implementation



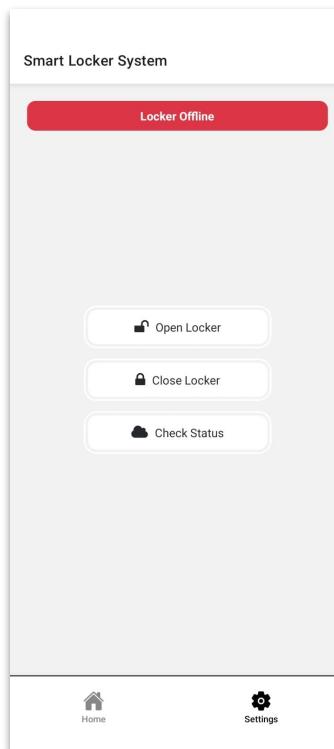
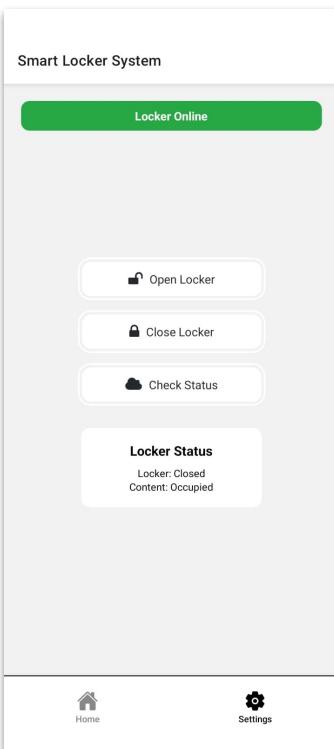
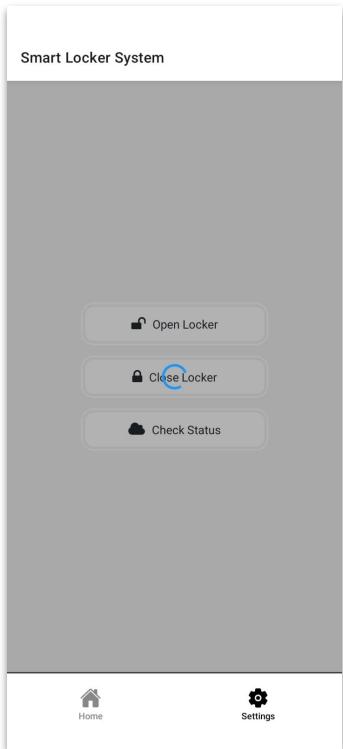
- **Setup:** Initialize serial, pins, Wi-Fi connection, servo attachment, web server routes, and start server
- **Loop:** Handle client requests continuously via web server
- **Web Server Routes:** Listen to endpoints `/open`, `/close`, `/status`, and respond accordingly

Apps Screens



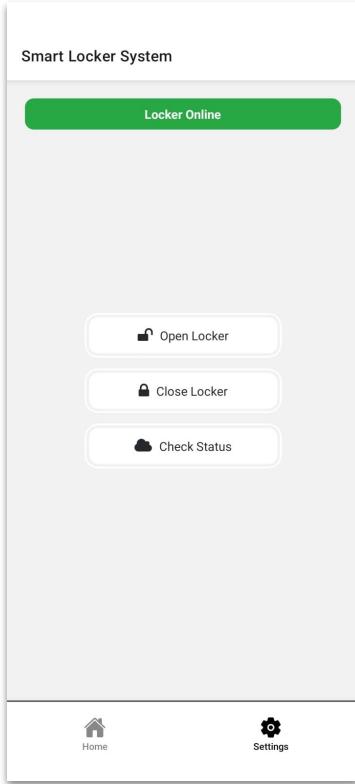
User is Able to Sign up and login to the app using their Credential .

App Screens



The Home page loads and tries to connect with the Locker System if successful a green Status bar showing that the locker is Online , if it fails a red status bar showing that the locker is offline appears .

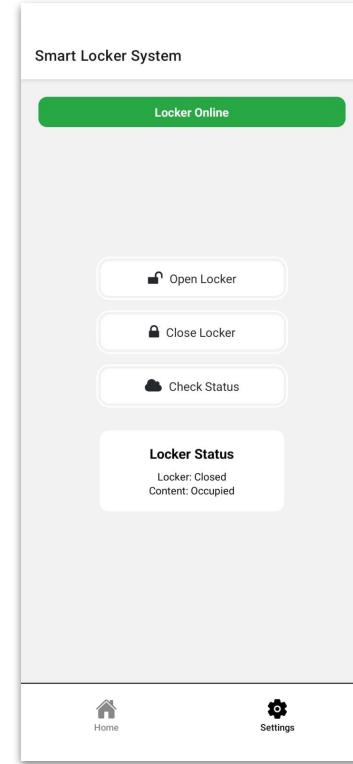
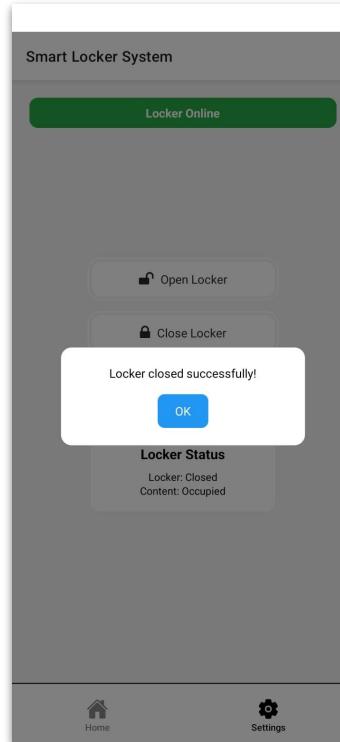
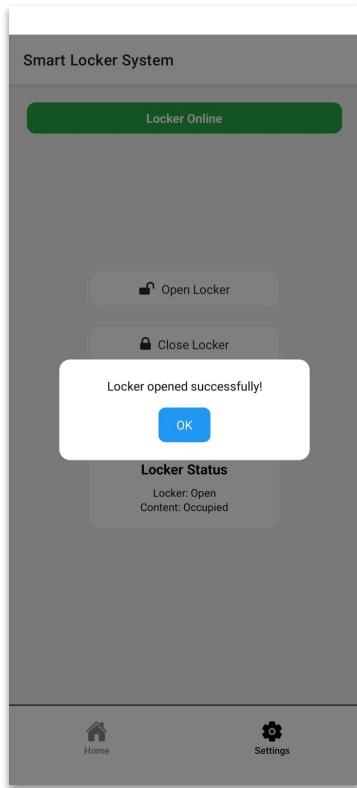
Apps Screens



The Home Screen allows the user to remotely lock and unlock the locker and it also allows the user to check the status of the locker such as whether the locker is occupied or not.

The Settings screen allows the user to change the wifi to which the locker is connected.

App Screens



Features and functionality

- **Remote Open/Close:** Locker can be controlled remotely via web interface and mobile app, enabling convenient access from anywhere.
- **Status Check:** Real-time monitoring of locker state (open/closed) and content presence using IR sensor, visible in the app.
- **Authentication and Security:** Secure login mechanism on the mobile app ensures only authorized users can operate locker.
- **Feedback Mechanisms:** Visual and audio feedback with LEDs (green for open, red for closed) and buzzer alerts ensure user awareness of actions.
- **Wi-Fi Credential Update:** Users can update locker Wi-Fi credentials remotely through the mobile app for network configuration flexibility.

Testing and results

Error Handling and Retry Logic

- Handles network connectivity issues by showing offline alerts.
- Implements **retry mechanisms** when fetching status or sending control commands fails.

Performance Metrics

- **Quick response times** under typical Wi-Fi conditions (~1-2 seconds from app command to lock action).
- Reliable operation with minimal communication errors.
- User feedback via **LEDs and buzzer** ensures confirmation of locker commands.

Challenges and solutions

- **Wi-Fi Issues:** Unstable connection caused communication drops.
Solution: Added **retry logic** and offline alerts.
- **Authentication:** Ensuring only authorized users control locker.
Solution: **Login modal** with credential check.
- **Status Accuracy:** Reliable updates on locker and content status.
Solution: Used **IR sensor** and **servo position** for precise status.
- **User Feedback:** Clear, immediate action confirmation needed.
Solution: Integrated **LEDs** and **buzzer** for visual and audio feedback.

Future Scope

- **Integration with cloud storage** to maintain detailed access logs and usage analytics.
- Support for **multiple users** with **individual profiles** and **access permissions**.
- Incorporation of advanced **authentication methods** such as **biometric sensors** (fingerprint, face recognition) and **One-Time Passwords (OTP)** for higher security.
- **Integration with other IoT devices** and **smart home systems** for expanded automation and monitoring.
- Implementation of **Over-The-Air (OTA) firmware updates** for easier maintenance and feature upgrades.
- Adding **notifications (SMS, email)** for locker activities and alerts.

Conclusion

Summary of Achievements

- Developed an ESP32 smart locker with remote app control.
- Real-time monitoring of locker and content status.
- Secure authentication and persistent Wi-Fi storage.
- Feedback via LEDs and buzzer for user interaction.
- User-friendly React Native app for locker operation.

Key Takeaways and Learning Outcomes

- Gained IoT design and ESP32 programming experience.
- Integrated hardware (servo, sensors, LEDs) with software.
- Tackled wireless connectivity and security challenges.
- Learned React Native for IoT app development.
- Identified future enhancements like cloud and advanced auth

Q&A

- Thank you for your attention.
- Please feel free to ask any questions or clarifications about the ESP32-based Smart Locker System project.
- Looking forward to your feedback and discussion.

Thank You !