Gate Pass Management System

RFID based Efficient Gate pass management system with IoT technology

Supervised by :

Prof. Avishek Barman
Assistant professor,
Dept. of Computer Science
Ramakrishna Mission Vidyamandira

Submitted By :

Anirban Santra
Gourab Das
Arkaprava Chakraborty

Gate Pass Management System
(GMS)

By
Anirban Santra, Arkaprava Chakraborty, Gourab Das

Under guidance of
Prof. Avishek Barman,
Assistant Professor,
Dept. of Computer Science

A project submitted in partial fulfilment of the requirements for the coursework of CC-10.

1st Year, 2nd Semester
Session – 2025
Batch of 2024-26

Ramakrishna Mission Vidyamandira
Belur Math,Howrah
2025

CERTIFICATE

We hereby certify that the work presented in this M.Sc. 2nd Semester Project Report, titled *"**Gate Pass Management System (GMS)**"*, is an original and authentic record of our research and development. This project has been undertaken as part of the coursework for **CC-10** in partial fulfillment of the requirements for the M.Sc. program.

The report has been submitted to the Department of Computer Science & Electronics, Ramakrishna Mission Vidyamandira, Belur Math, and documents work carried out from January 2024 to May 2024 (2nd Semester) under the esteemed supervision of **Avishek Barman**, Assistant Professor, Department of Computer Science & Electronics, Ramakrishna Mission Vidyamandira, Belur Math.

We affirm that the content of this report has not been submitted, either in part or in full, for the award of any other degree or certification at any institution.

________________	________________________	________________
Anirban Santra	Arkaprava Chakraborty	Gourab Das
A04-1112-0556-24	A04-1112-0554-24	A04-1112-0555-24

This is to certify that the above statement made by students is correct to the best of my knowledge.

_________________	__________________	________________________
Sign. of Supervisor	Sign. of HOD	Sign. of External Examiner
Avishek Barman	Dr. Arindam Sarkar	

Place: Belur Math, Howrah-711202

Date : _____/_____/ _________

ACKNOWLEDGEMENT

We extend our deepest gratitude to **Professor Avishek Barman**, our project guide, for his invaluable mentorship, unwavering support, and expert guidance throughout the challenging stages of this project. His insightful suggestions and constant encouragement played a crucial role in bringing this work to completion.

We are also profoundly grateful to **Dr. Arindam Sarkar**, Head of the Department of Computer Science & Electronics, for fostering an environment of academic excellence and for his continuous support. Our sincere appreciation extends to all the faculty members of the department, whose teachings have shaped our knowledge and provided the foundation for this project.

Additionally, we express our heartfelt thanks to **Sri Sanjib Kumar Basu & Ajaharul Islam Molla**, our dedicated laboratory attendants, for his constant support and encouragement.

Furthermore, we are deeply thankful to **Br. Arijit Maharaj**, our Bhavan Superintendent, for his continuous support and guidance throughout our academic journey.

Finally, we convey our sincere gratitude to **Swami Mahaprajnananda** Ji Maharaj, our esteemed Principal, for providing us with the necessary infrastructural facilities to undertake this project, and to **Br. Tattvachaitanya** Ji Maharaj, our Vice Principal, for his constant encouragement. We also extend our appreciation to **Ramakrishna Mission Vidyamandira**, whose academic environment and values have instilled in us the discipline and dedication required to accomplish this work.

Anirban Santra	Arkaprava Chakraborty	Gourab Das
A04-1112-0556-24	A04-1112-0554-24	A04-1112-0555-24

Place: Belur Math, Howrah-711202

Date : _____/_____/ _________

Table of Contents

List of Figures

Abstract

The RFID-Based Gate Pass Management System was designed and implemented specifically for **Ramakrishna Mission Vidyamandira** to enhance security, automate student movement tracking, and streamline hostel management. This system leverages **RFID technology, ESP32 microcontrollers, Firebase real-time databases, and a web-based dashboard** to replace traditional manual registers with an efficient, automated solution.

Students use RFID cards to check in and out of the hostel, with their movements being recorded in real-time. The system enforces **predefined hostel exit rules** while allowing students to request **gate passes for late exits**, which wardens can approve through an intuitive web interface. Database synchronization between **Firebase and MySQL** ensures data consistency, while **real-time monitoring** enables authorities to oversee student activities effectively.

By addressing the specific security and administrative requirements of **Ramakrishna Mission Vidyamandira**, this project provides a **scalable and paperless** solution that improves hostel management. Future enhancements include **mobile app development, biometric authentication, AI-based movement analytics, and multi-hostel integration**, ensuring greater security and ease of use.

Chapter 1 Introduction

Objectives

- Discuss key features of the project.
- Highlight Objectives.

1. Introduction

With the growing need for automation and security in **educational institutions, workplaces, and residential areas**, traditional manual access control systems are no longer efficient. These conventional methods often lead to **delays, errors, and unauthorized access**, making them unreliable. To address these challenges, this project introduces an **RFID-Based Smart Gate Pass System**, integrating **IoT, cloud computing, and real-time authentication** to ensure seamless and secure entry management.

The system utilizes **RFID technology**, where each individual is assigned a unique RFID card. Upon scanning at the gate, the RFID reader extracts the **unique ID** and sends it to an **ESP32 microcontroller**, which communicates with a **Firebase Realtime Database**. The system then checks **entry permissions, predefined time constraints, and access logs** before granting or denying entry. The process is further **automated using LEDs and a buzzer**, providing instant feedback to the user.

This **smart access control system** is particularly beneficial for **hostels, offices, and gated communities**, where controlled access is essential. By leveraging cloud technology, the system ensures **remote monitoring, real-time data synchronization, and efficient entry tracking**.

Ramakrishna Mission Vidyamandira
(A Residential Autonomous College under University of Calcutta with CPE status)
Belur Math, Howrah, 711202
Gatepass Management System
Developed by -
Dept. of Computer Science & Electronics
Anirban, Arkaprava & Gourab

Figure 1 Proposed Gate pass Management System

1.1 Key Features

- **Automated Entry & Exit** – Eliminates manual checking, ensuring faster access.
- **Real-Time Authentication** – Verifies RFID tags against the **Firebase database** instantly.
- **Time-Based Access Control** – Restricts entry based on **predefined schedules** (e.g., hostel curfew hours).
- **ESP32 & WiFi Connectivity** – Enables seamless **wireless communication** with cloud storage.
- **Visual & Audio Indicators** – **LEDs & buzzer** signal whether access is granted or denied.
- **Database Logging & Tracking** – Keeps track of **entry/exit timestamps** for security monitoring.
- **Remote Database Management** – Admins can update access permissions from anywhere.

1.2 Objectives

- **Enhance Security:** Implement a robust **RFID-based authentication system** to prevent unauthorized access.
- **Reduce Manual Intervention:** Automate the gate pass system to **minimize human errors and delays**.
- **Ensure Real-Time Data Updates:** Synchronize entry/exit logs with **Firebase for instant updates**.
- **Implement Time-Based Access Rules:** Restrict movements based on **institutional regulations** (e.g., hostel timings).
- **Improve System Scalability:** Develop a **cloud-connected** solution that can be **expanded** to multiple locations.
- **Enable Remote Monitoring & Management:** Allow administrators to **modify access settings from anywhere**.

Chapter 2 System Review

Objectives

- Identify inefficiencies and security risks.
- Gather insights from users.
- Assess technical and economic viability.
- Highlight the need for a digital solution.

2. System Review

2.1 Existing System

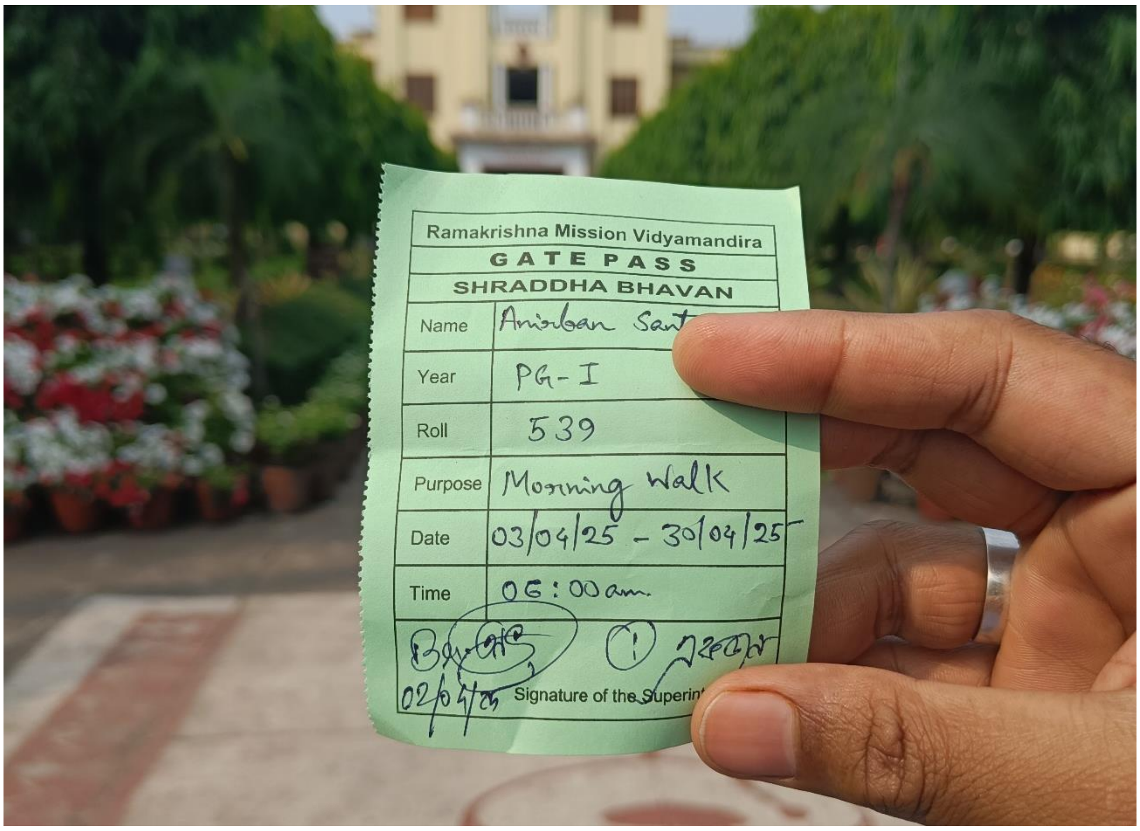
Traditional hostel entry and exit management systems primarily rely on manual processes, leading to inefficiencies and security vulnerabilities. The key components of these systems include:

2.1.1 Manual Register Maintenance

- Students are required to record their details, including name, roll number, time, and purpose, in a physical register before leaving or entering the hostel.
- Security personnel manually verify and approve these entries.

2.1.2 Handwritten Gate Passes

- Students requiring permission for late entry/exit must obtain a handwritten gate pass from the warden.
- These passes are verified at the gate, adding an extra layer of manual checking.



Ramakrishna Mission Vidyamandira

GATE PASS

SHRADDHA BHAVAN

Name	Anirban Sant
Year	PG-I
Roll	539
Purpose	Morning Walk
Date	03/04/25 - 30/04/25
Time	06:00 am.

[illegible] 02/04/25 [illegible]

Signature of the Superin

Figure 2 Manual Gate pass - Existing System

2.1.3 Manual Verification at the Gate

- Security personnel verify student details against the register or gate pass before allowing entry or exit.
- The verification process is inconsistent and susceptible to human errors.

2.1.4 Lack of Real-Time Monitoring

- There is no automated tracking system to notify authorities about students returning late.
- Retrieving past records is cumbersome and time-consuming.

2.2. Challenges in the Existing System

The manual nature of the current system presents several challenges:

2.2.1 Time-Consuming Processes

- Logging information and verifying details manually leads to delays and long queues at entry and exit points.

2.2.2 Prone to Errors

- Handwritten records can be misread, altered, or misplaced.
- Human oversight increases the risk of unauthorized access.

2.2.3 Limited Accessibility

- Wardens and administrators lack real-time access to student movement data.
- Searching for past records is inefficient and labour-intensive.

2.2.4 Security Risks

- Unauthorized entries or exits may go unnoticed.
- No automated alerts exist for students violating hostel timings.

2.3. Primary Investigation & Feedback

To assess the limitations of the current system, discussions were conducted with students, wardens, and security personnel. The findings are as follows:

2.3.1 Student Feedback

- Students expressed frustration over long waiting times for manual verification.
- Many reported issues with losing or forgetting gate passes.
- A digital system was preferred for improved efficiency and ease of use.

2.3.2 Warden Feedback

- Wardens found maintaining registers cumbersome and prone to errors.
- They faced challenges in tracking students who failed to return on time.
- A real-time tracking system was suggested for better hostel management.

2.3.3 Security Personnel Feedback

- Checking registers and verifying gate passes manually was slow and inefficient.
- The possibility of fake or altered gate passes was a concern.

Figure 3 Gate pass presented and verified at Gate

2.4. Motivation

The inefficiencies and security risks associated with the manual system necessitated a **reliable and automated solution**. The traditional approach of maintaining physical registers and handwritten gate passes was not only labour-intensive but also vulnerable to errors and security breaches. Security personnel faced challenges in verifying gate passes, while unauthorized entries were difficult to track. Moreover, the absence of real-time monitoring hindered accurate student movement records, raising safety concerns.

Stakeholder feedback underscored the urgent need for digital transformation. Wardens struggled with register maintenance, students found the process inconvenient, and security personnel found verification inefficient, especially during peak hours. The integration of **RFID technology, ESP32-based automation, and Firebase real-time databases** presented an optimal solution.

The motivation behind this project was to eliminate manual processes, **enhance security, ensure real-time tracking, and streamline hostel entry/exit management**. By leveraging modern technology, this system aims to reduce human errors and improve operational efficiency for both students and administrators.

2.5. Feasibility Study

Before implementation, a feasibility study was conducted to assess the project's viability across multiple dimensions:

2.5.1 Technical Feasibility

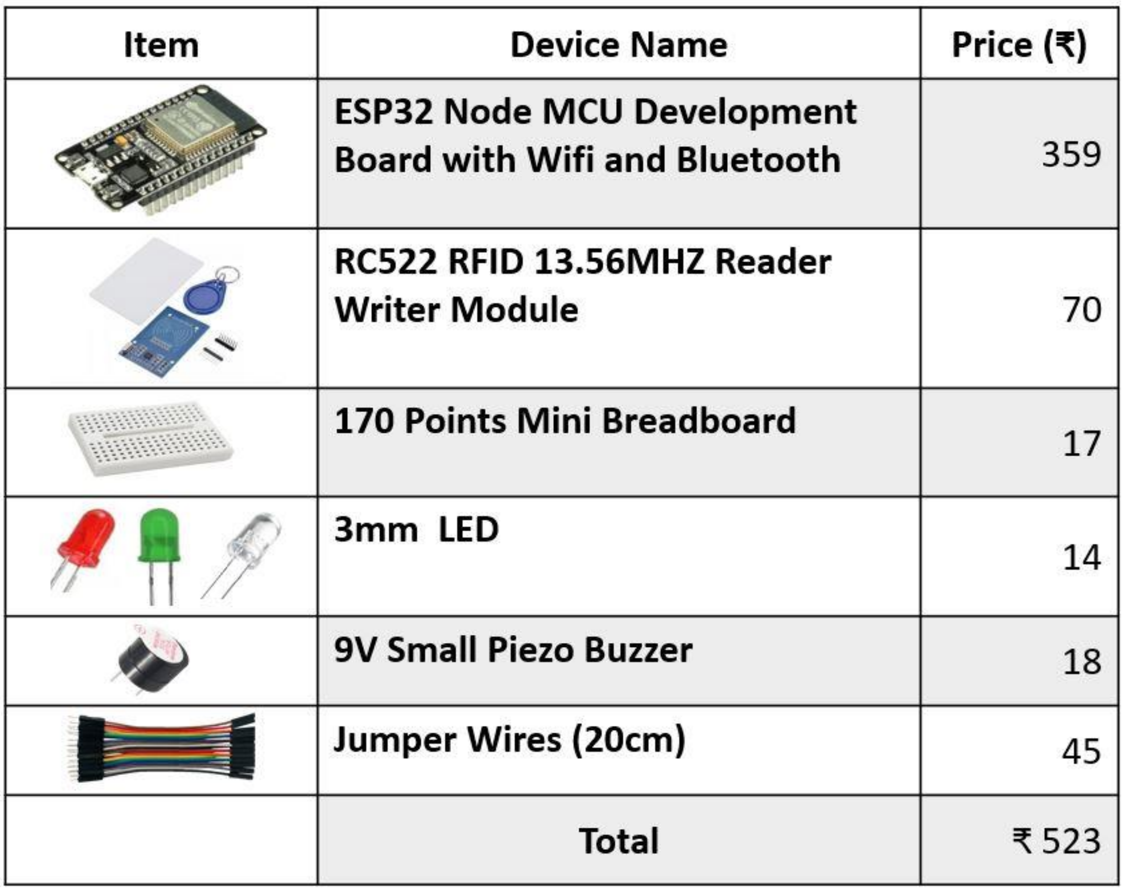
- The system is technically feasible as it utilizes widely available hardware components such as **ESP32, RFID sensors, LED indicators, and buzzers** for automation.
- The software implementation is based on **PHP, MySQL, and Firebase**, which are well-documented and supported.
- The integration of **ESP32 with Firebase** ensures **real-time updates**, making the system efficient and easy to manage.

2.5.2 Schedule Feasibility

- The project was divided into **manageable modules**, including:
    - Database setup
    - RFID authentication
    - ESP32 integration
    - Web interface development
- The prototype implementation was estimated to be feasible within a few weeks, with full deployment achievable within a few months after testing and refinements.

2.5.3 Economic Feasibility

- The system is economically viable, significantly reducing operational costs in the long run.
- The cost of **RFID cards, ESP32 modules, and LED indicators** is minimal compared to the expenses of maintaining manual registers, printing gate passes, and employing additional personnel for record-keeping.
- The use of **open-source technologies** minimizes software development costs, making the system a cost-effective solution.



Item	Device Name	Price (₹)
	ESP32 Node MCU Development Board with Wifi and Bluetooth	359
	RC522 RFID 13.56MHZ Reader Writer Module	70
	170 Points Mini Breadboard	17
	3mm LED	14
	9V Small Piezo Buzzer	18
	Jumper Wires (20cm)	45
	Total	₹ 523

Figure 4 Cost of all equipment

2.5.4 Cultural Feasibility

- Transitioning from a manual to a digital system required acceptance from students, wardens, and security staff.
- Initial resistance was observed, particularly among staff accustomed to traditional methods.

- Training sessions and demonstrations were conducted to familiarize users with the system, resulting in positive acceptance, especially among students who preferred digital solutions.

2.5.5 Legal Feasibility

- The system complies with legal considerations, including **data privacy and security regulations**.
- Personal data such as student IDs and entry/exit logs are securely stored with restricted access.
- The system adheres to **institutional and governmental policies** governing hostel management.

2.5.6 Resource Feasibility

- All necessary **hardware and software components** are readily available.
- The expertise required for **software development, database management, and ESP32 programming** is within the capabilities of the development team.
- Maintenance and future updates can be managed with minimal resource allocation, ensuring long-term sustainability.

This feasibility study confirms that the **RFID-Based Smart Gate Pass System** is **technically, economically, and operationally viable**, ensuring enhanced **security, efficiency, and real-time monitoring** for hostel management.

3

Chapter System Design

Objectives

- Choose Agile for flexibility and iterative development.
- Ensure stakeholder collaboration and rapid improvements.
- Develop system in structured phases with testing.
- Define system interactions using diagrams and requirements.

3. System Design

3.1 Design Methodology

There are various SDLC models available to implement a project, each having its own advantages and disadvantages. Choosing the right model depends on the project scope, team size, and complexity of the system being developed.

For this project, we selected the **Agile Model** due to its flexibility, iterative nature, and ability to accommodate continuous improvements. Below, we discuss the Agile Model and its implementation in our project.

3.1.1 The Agile Model

The **Agile SDLC model** is a combination of **iterative** and **incremental** process models. It focuses on **process adaptability** and **customer satisfaction** by ensuring the rapid delivery of a working software product. Agile breaks the product into **small incremental builds**, which are developed in iterations.

Each iteration typically lasts between **one to three weeks**, and at the end of each cycle, a **working product** is demonstrated to stakeholders. In our case, the important stakeholder was our **supervisor**, to whom we presented the **working system** after each iteration.

Contrary to popular belief, Agile does not mean an **unplanned** or **unstructured** approach. Instead, it follows a structured **iterative development cycle**, ensuring that the final product is built systematically while incorporating necessary changes along the way.

3.1.2 Advantages of the Agile Model

- A **realistic approach** to software development.
- More **flexible** and adaptable to changes.
- Functionality can be developed **rapidly** and demonstrated in early stages.
- Delivers **early partial working solutions** to stakeholders.
- Enables **concurrent development and delivery** within an overall planned framework.
- Requires **minimal initial planning**.
- Easier to manage due to **short iterations**.
- Gives **greater flexibility** to developers, allowing them to experiment and improve.

3.1.3 Disadvantages of the Agile Model

- Not suitable for **handling complex dependencies**.
- Higher risk of **sustainability, maintainability, and extensibility** challenges.

- Team members need to fully **embrace the Agile methodology** for it to be effective.
- Lack of discipline can lead to **inconsistencies** in development.
- **Peer pressure** in a small team can be intense.

Limited documentation makes it harder to transfer knowledge to new members

---

3.1.4 Why We Chose the Agile Model

- **Iterative Development & Quick Adaptation** – Agile allowed incremental development, starting with basic RFID authentication and gradually adding Firebase integration, real-time monitoring, and access rules. Frequent testing ensured quick adaptation to changes.
- **Enhanced Collaboration & Stakeholder Involvement** – With multiple components like ESP32 firmware, Firebase, and a web dashboard, Agile ensured continuous communication among hardware, software teams, and stakeholders for smooth integration.
- **Faster Deployment & Time-to-Market** – Functional versions were deployed in stages, enabling real-world testing and improving stability without waiting for full system completion.
- **Managing Changing Requirements** – Since access rules and security policies evolve, Agile allowed easy modification of system rules and user permissions without disrupting overall functionality.
- **Continuous Integration & Testing** – Regular unit testing for ESP32, Firebase APIs, and web dashboard ensured bugs were identified early, maintaining system reliability.
- **Risk Management & Scalability** – Agile's modular approach prevented failures in one component (e.g., Firebase communication) from halting the entire system and enabled future scalability, like biometric integration or mobile access.

3.1.5 How this Agile Model is used

The Agile methodology ensures that the final product evolves through multiple iterations, with each iteration delivering a functional build. Below, we describe how our project progressed through different Agile iterations.

First Iteration: Basic Interfacing

- Developed a basic program to take user input via the command line & RFID.
- Program relayed input to hardware, initially consisting of simple LED circuits acting as indicators.
- Focused on interfacing ESP32 GPIO Pins with hardware circuits.
- Implemented Validation of RFIDs after successful Read.

This helped establish a basic communication framework between the software and hardware.

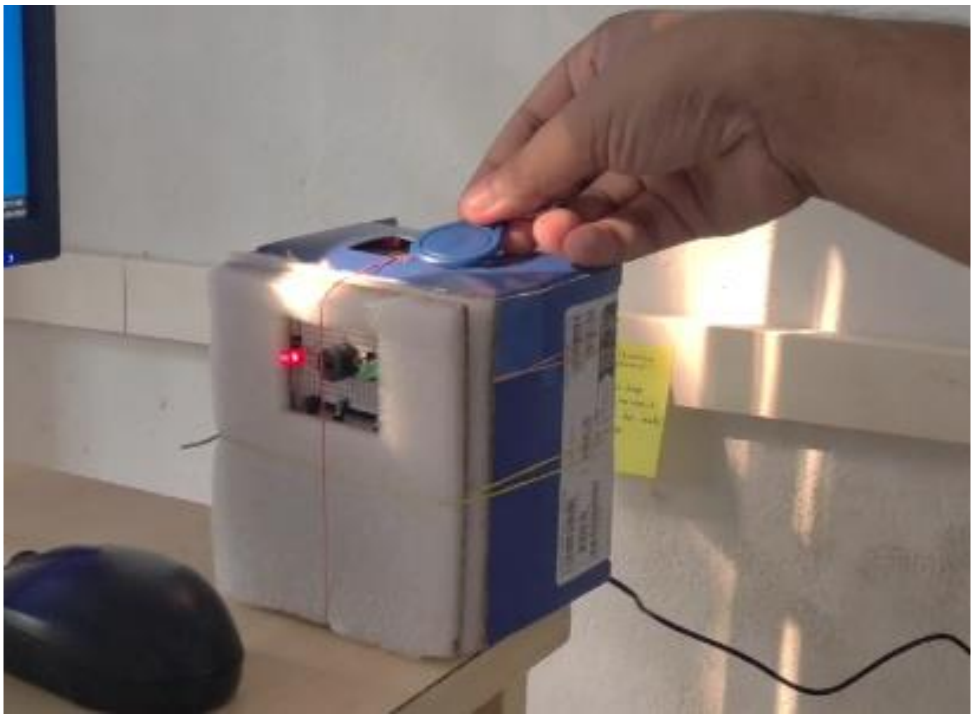


Figure 5 First Phase implementation

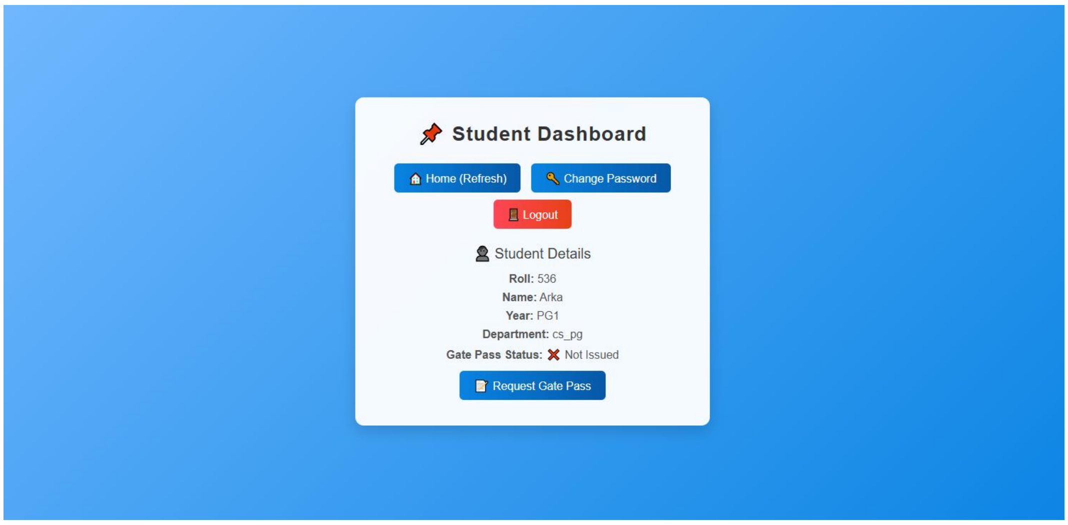
Second Iteration: Server and Web Interface

- Implemented a web server using PHP.
- Developed a basic web interface with buttons to control connected devices (RFID-based system).

At this stage, the system was operational but lacked authentication, meaning anyone could access the controls.

Third Iteration: Authentication & Student Login

- Introduced a login system for warden authentication.
- Implemented session-based access control to ensure only authorized personnel could access the system.
- Introduced student login feature to request Gate pass.



📌 Student Dashboard

👤 Student Details

Roll: 536
Name: Arka
Year: PG1
Department: cs_pg
Gate Pass Status: ❌ Not Issued

Figure 3 Student Dashboard implemented

Final Iteration: Responsiveness, Improved UI & Internet Exposure.

- Added more functions as requested by Wardens.
- Refined the User Interface (UI) to enhance usability.
- Added exception handling statements and made the interface responsive.
- A landing Index page was created and the project was exposed to internet using ngrok.

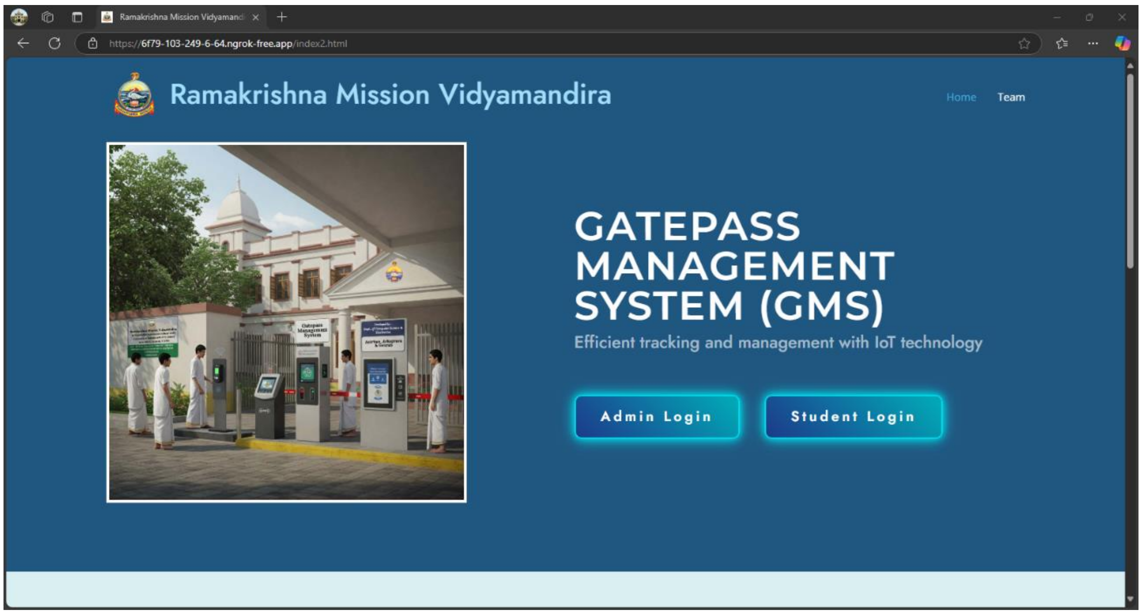

Ramakrishna Mission Vidyamandi
https://6f79-103-249-6-64.ngrok-free.app/index2.html
Ramakrishna Mission Vidyamandira
Home
Team
GATEPASS MANAGEMENT SYSTEM (GMS)
Efficient tracking and management with IoT technology
Admin Login
Student Login

Figure 6 Homepage of The Project

3.2 Design Documents

The design phase requires the creation of several documents to serve as the documentation of the project. Two of those are the Use Case Diagram and the Data Flow Diagram.

3.2.1 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. It gives a very high-level view of the system and is usually prepared before the development starts.

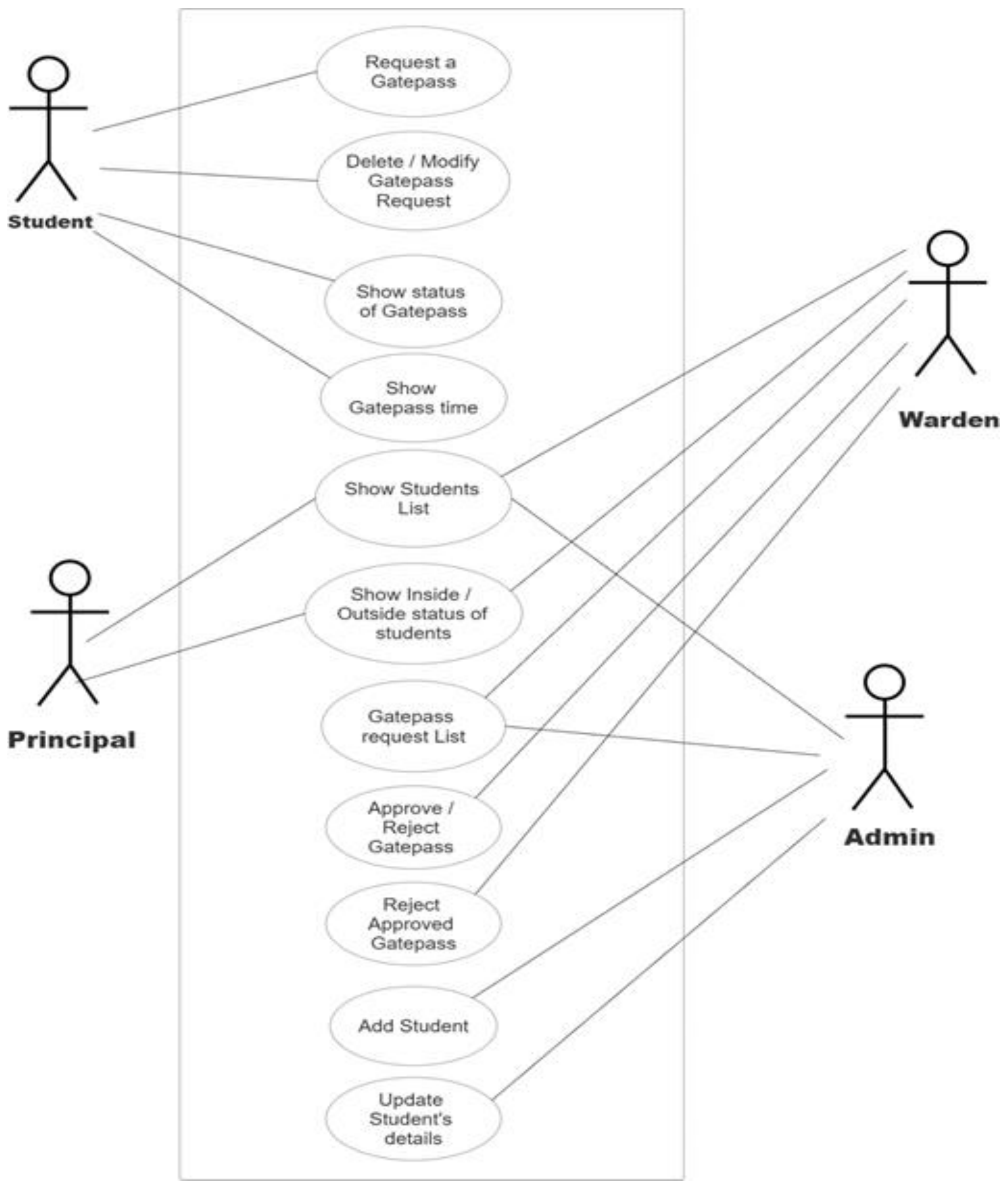


Figure 7 Use Case Diagram

The **Use Case Diagram** in *Figure 7* represents a **Student Gate pass Management System**, outlining how different users (students, wardens, principals, and admins) interact with the system. Each role has specific permissions, ensuring a well-structured and secure process for managing student movements.

Students are the primary users who can request a gate pass, check their request status, and modify or delete requests before approval. They can also view their assigned gate pass time and check a list of other students who have requested passes. However, they cannot approve their own requests and must wait for **warden or admin approval** before they can exit the premises. This ensures a **controlled movement process** within the hostel system.

Wardens are responsible for managing gate pass requests and tracking student movement. They can view the **inside/outside status** of students, approve or reject requests, and even revoke an approved gate pass if necessary. Their role ensures that students follow designated **exit and return timings**, maintaining hostel security. Additionally, **principals** can oversee student movements but do not have the authority to approve or reject gate passes, making them **observers rather than decision-makers**.

The **admin** has the highest authority in the system, with the ability to **approve/reject gate passes**, revoke approved requests, and manage student records. They can **add new students** to the database and **update student details** as needed. The admin also plays a key role in ensuring that all **RFID and access records are properly maintained** in both MySQL and Firebase, ensuring real-time updates across the system.

This structured **hierarchical approval system** prevents unauthorized exits and ensures that student movements are regulated. The **real-time synchronization** with Firebase allows for instant updates, making the system **efficient and transparent**

3.2.2 Data Flow Diagram (Level 0)

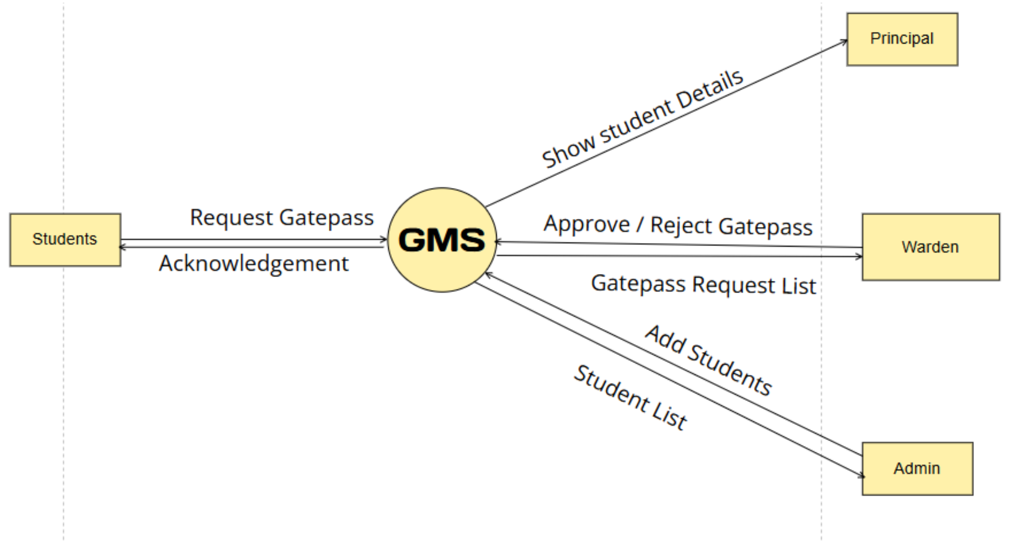


Figure 8 DFD Level 0

3.2.3 Data Flow Diagram (Level 1)

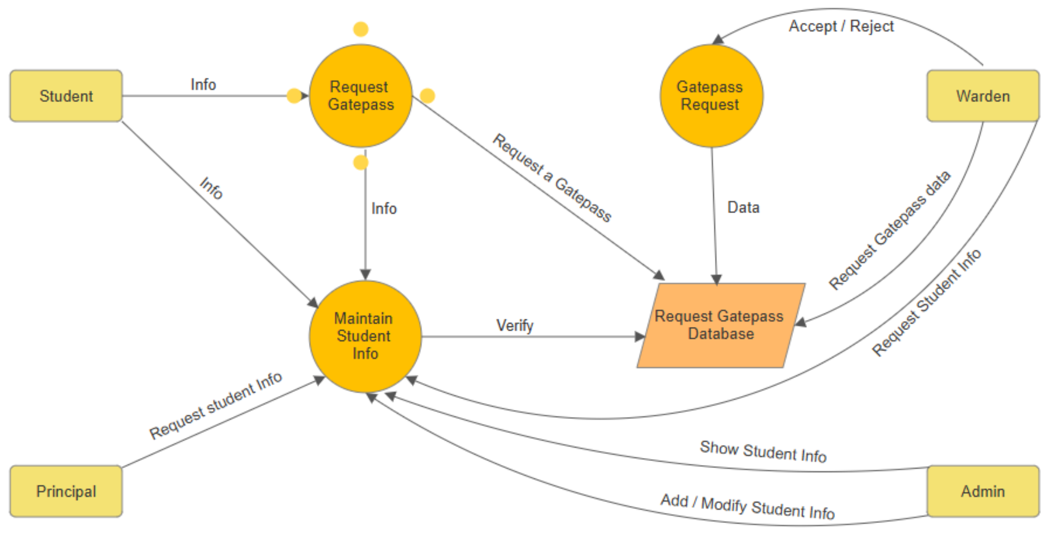


Figure 9 DFD Level 1

3.2.3 Data Flow Diagram (Level 2)

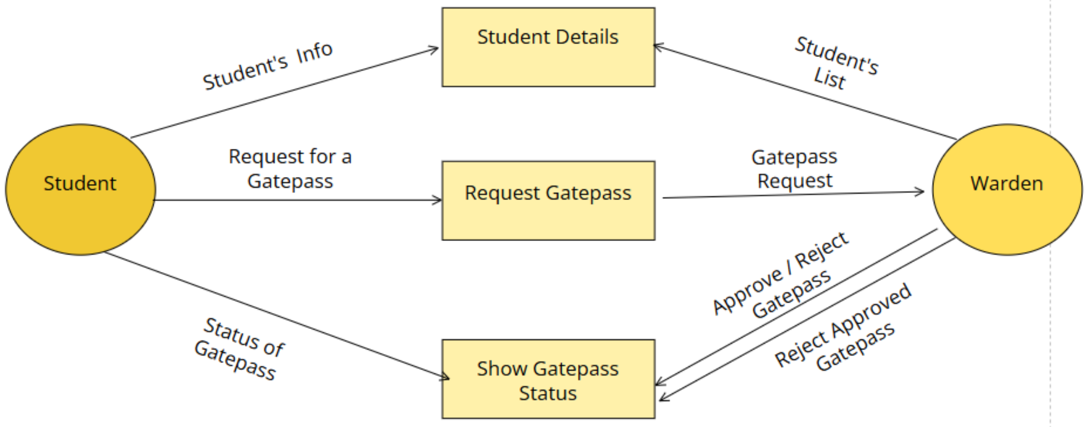


Figure 10 DFD Level 2

3.3 System Requirements

3.3.1 Hardware Components

Component	Description
ESP32 Microcontroller	Controls the entire system, processes RFID data, and connects to Firebase.
RFID Reader (MFRC522)	Reads RFID cards and extracts unique IDs for authentication.
RFID Tags (Cards/Keychains)	Unique identification cards assigned to students for access control.
LED Indicators (Red, Green, White)	Provide visual feedback for access granted/denied.
Buzzer	Emits sound signals to indicate successful or denied authentication.
Power Supply (5V/3.3V)	Provides necessary power to the ESP32 and other electronic components.
WiFi Router	Ensures internet connectivity for real-time database synchronization.
Jumper Wires & PCB Board	Used for circuit connections and assembling the hardware setup.
Enclosure (Casing for Components)	Protects hardware components from external damage.

3.3.2 Software Components

Software	Purpose
Arduino IDE	Used to program and upload code to the ESP32 microcontroller.
Firebase Realtime Database	Stores student access records and authentication logs.
MySQL Database	Maintains local records of student information and access logs.
PHP & Web Technologies (HTML, CSS, JavaScript)	Develops the web-based admin panel for access control and monitoring.
Ngrok	Exposes local servers securely to the internet.
ESP32 Libraries (WiFi, HTTP, MFRC522, Time Libraries)	Required for ESP32 communication, RFID processing, and time synchronization.

4

Chapter Implementation

Objectives

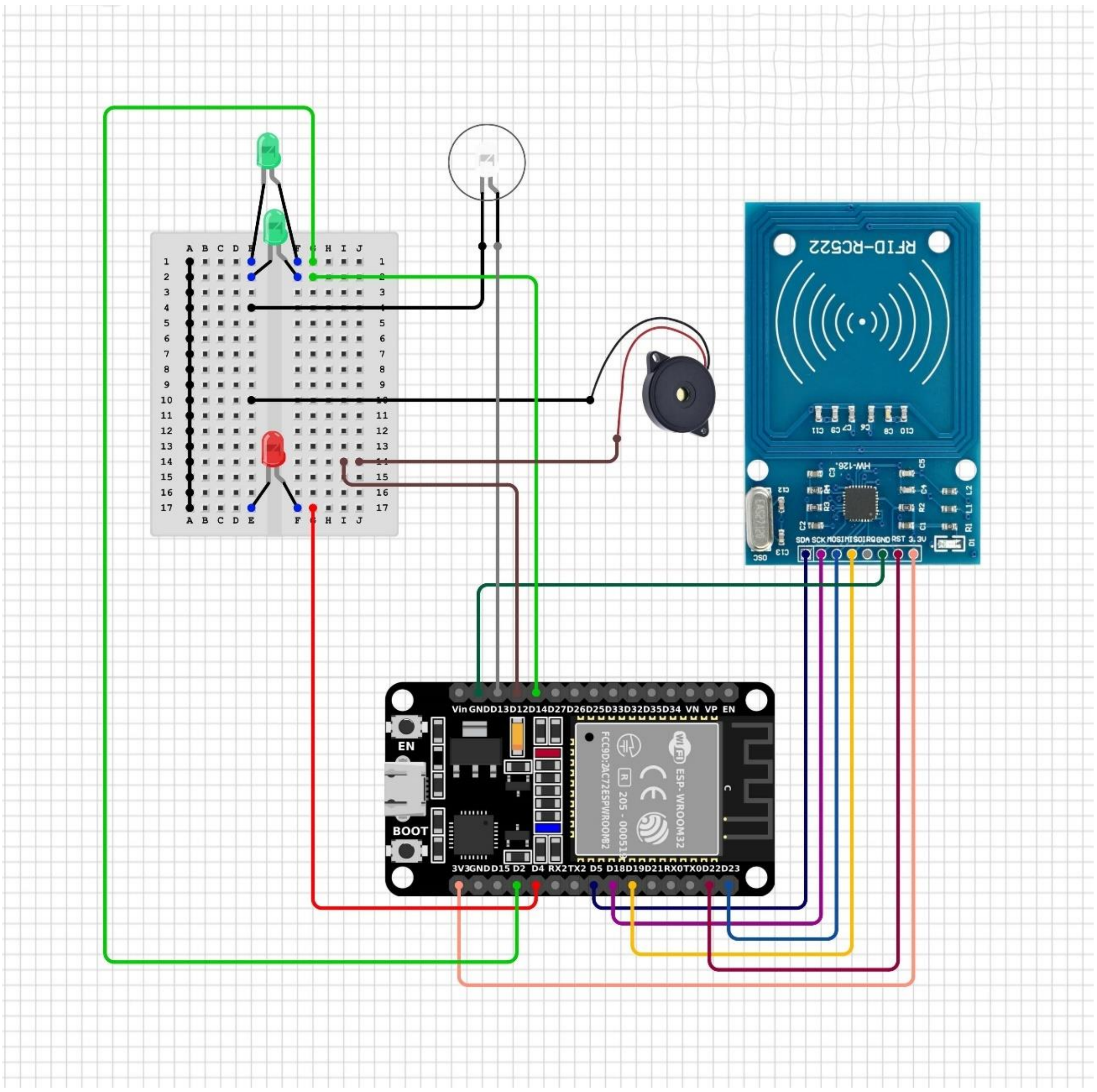
- Ensure smooth RFID, ESP32, Firebase, and MySQL communication.
- Grant or deny entry based on time and database rules.
- Instantly update Firebase and MySQL for tracking.
- Provide intuitive dashboards for management.

4. Implementation

The implementation of the **RFID-based Gate pass Management System** involves integrating **hardware and software components** to automate student movement tracking and access control. The system is designed to ensure **secure, real-time monitoring** using **RFID, Firebase, MySQL, ESP32, and a web-based dashboard**.

4.1 Hardware Implementation

4.1.1 Circuit Design



RFID-RC522
HW-126
SDA SCK MOSI MISO IRQ GND RST 3.3V
Vin GND D13 D12 D14 D27 D26 D25 D33 D32 D35 D34 VN VP EN
EN
BOOT
ESP-WROOM32
3V3 GND D15 D2 D4 RX2 TX2 D5 D18 D19 D21 RX0 TX0 D22 D23

Figure 11 Circuit Diagram of The Device

The **Gate pass Management System** hardware circuit is designed to facilitate secure and automated access control. The key hardware components and their connections ensure smooth operation, real-time data processing, and seamless communication between the RFID reader, microcontroller, and database.

---

1. ESP32 Microcontroller (Brain of the System)

- The **ESP32** acts as the central processing unit, managing RFID scanning, database communication, and access decisions.
- It connects to **WiFi**, allowing real-time Firebase/MySQL updates.
- It controls **LEDs, Buzzer, and other peripherals** based on authentication results.

Connections:

- **SPI Communication:** Connected to the **RFID Reader (RC522)** for reading RFID tags.
- **GPIO Pins:** Used for **LEDs, buzzer, and RFID reader control.**

2. RFID Reader (MFRC522) – User Identification

- The **RFID reader** detects RFID tags when scanned and sends the **unique ID (UID)** to the ESP32.
- The ESP32 then verifies the UID in the database to determine whether access is granted or denied.

Connections (SPI Interface with ESP32):

- **SDA (SS)** → GPIO **5**
- **SCK** → GPIO **18**
- **MOSI** → GPIO **23**
- **MISO** → GPIO **19**
- **RST** → GPIO **22**
- **GND** → GND
- **VCC** → 3.3V

3. LED Indicators – Visual Feedback for Users

- **Green LED:** Lights up if access is granted.
- **Red LED:** Lights up if access is denied.
- **Additional Green2 LED:** Used for double confirmation during entry/exit.
- **White LED:** Used for illumination during scanning.

Connections:

- **GREEN_LED** → GPIO **2**
- **RED_LED** → GPIO **4**
- **GREEN2** → GPIO **14**
- **WHITE LED** → GPIO **13**

4. Buzzer – Audio Feedback for Users

- The **buzzer provides an audio alert** when an RFID scan occurs and when access is either granted or denied.
- It **beeps differently for approved and rejected entries**, enhancing user experience.

Connection:

- **BUZZER** → GPIO **12**

5. NTP (Network Time Protocol) server – Accurate Timekeeping

- The ESP32 fetches the time from an **NTP (Network Time Protocol) server**.
- It allows access **only within the permitted time slots** as per the database.

6. Power Supply – Stable Operation

- The system operates on **5V and 3.3V power**.
- **ESP32 uses 3.3V**, while **some components (like the buzzer and LED) require 5V.**
- A **regulated power supply** ensures stable operation.

4.1.2 How the Circuit Works

The **circuit design** enables the ESP32 to communicate with the **RFID reader** and verify student access in real-time using Firebase. Below is the **working mechanism**:

Step-by-Step Working Process:

1. **System Power-Up**
    - The **ESP32 microcontroller** is powered via a **5V power supply** or USB connection.
    - It initializes the **RFID module, WiFi, and time synchronization settings**.
2. **RFID Card Scanning**
    - When a student places their **RFID card** near the **MFRC522 reader**, the reader captures the **unique ID** of the card.
    - The ID is sent to the **ESP32 for processing**.
3. **Database Verification**
    - The **ESP32 connects to Firebase Realtime Database** via WiFi.
    - It checks the student's RFID tag against stored **student records**.
    - If a matching ID is found, it retrieves the student's **entry permissions and time constraints**.
4. **Access Decision (LED & Buzzer Indication)**
    - If access is **granted**:
        - **Green LED** turns ON.

        - **Buzzer beeps twice** to indicate successful authentication.
        - The **Firebase database is updated** to log the entry.
    - If access is **denied**:
        - **Red LED** turns ON.
        - **Buzzer sounds continuously** for half a second.
        - The system rejects entry, and no database update occurs.
5. **Real-Time Status Update**
    - If the student **successfully exits**, the system updates their **"Out" status** in Firebase.
    - If they **enter late** or have **restricted access**, an alert can be triggered for **administrators or wardens**.
6. **Continuous Monitoring**
    - The system **loops continuously**, waiting for the next RFID scan.
    - Administrators can **remotely monitor** the logs using a **web interface or mobile application**.

4.1.3 Logic Behind Access Control

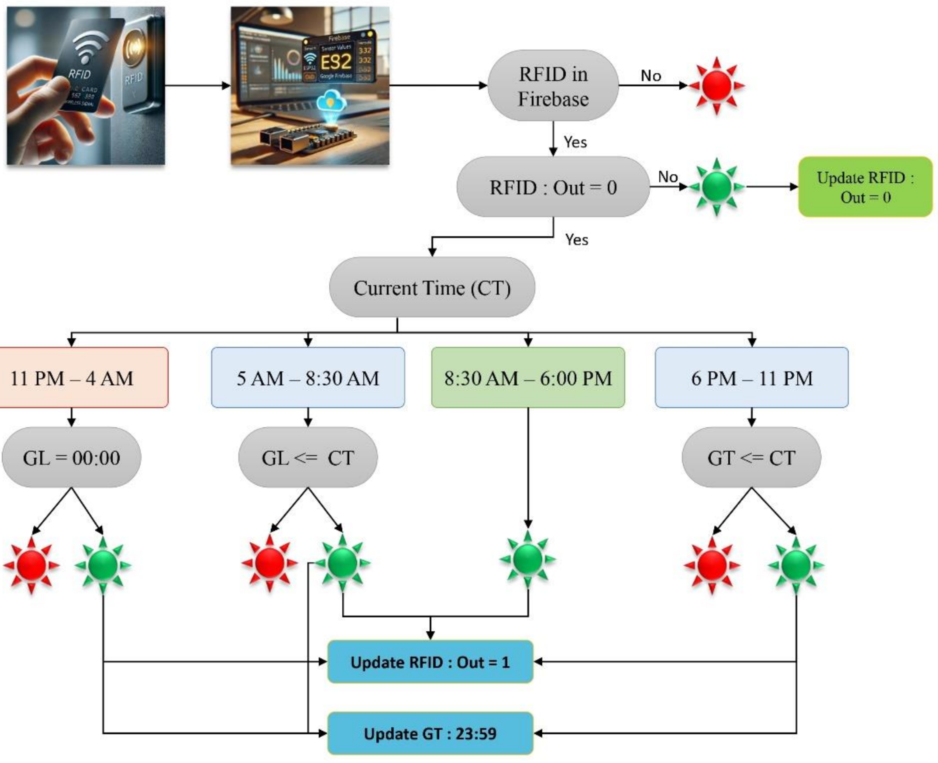


Figure 12 Device Control Logic

The system grants or denies access based on three key factors:

1. **Current Time (CT)** – The time at which the student scans their RFID card.
2. **Green LED Time (GT)** – The pre-approved time until which the student is allowed to remain outside.
3. **OUT Status** – Indicates whether the student is currently inside (OUT = 0) or outside (OUT = 1).

The decision-making process is divided into two main cases:

1. **Student wants to go out (OUT = 0)**
2. **Student wants to come in (OUT = 1)**

---

1. Student Wants to Go Out (OUT = 0)

This means the student is **currently inside** and is **requesting permission to leave the hostel**. The system evaluates the conditions based on **CT and GT** before granting access.

Conditions for Granting Exit (OUT = 0 → OUT = 1)

Time Slot (CT)	Condition for Granting Exit	Access Granted?	New GT Value	OUT Value After Exit
11:00 PM – 4:00 AM	GT must be **"00:00"**	✅ Yes	**23:59**	**1** (Student has exited)
5:00 AM – 8:30 AM	GT must be ≤ **CT**	✅ Yes	**23:59**	**1**
8:30 AM – 6:00 PM	**No GT check required**	✅ Yes	**No change**	**1**
6:00 PM – 11:00 PM	GT must be ≤ **CT**	✅ Yes	**23:59**	**1**
Other Cases	**Condition NOT met**	✕ No	**No change**	**0** (Student remains inside)

✅ **If conditions are met, access is granted:**

- **OUT is updated to 1** (student has exited the hostel).
- **GT is set to 23:59**, meaning further exit is restricted until re-approved.

✕ **If conditions are NOT met, access is denied** and the student cannot exit.

---

2. Student Wants to Come In (OUT = 1)

If OUT = 1, this means the student is **outside the hostel** and is now **requesting to enter**.

✅ **Access is always granted when a student wants to come in**

- **OUT is updated to 0**, meaning the student is now inside the hostel.

Example Scenarios

Scenario 1: Student Wants to Exit at 2:00 AM (CT = 2:00 AM)

- GT must be "00:00" for access to be granted.
- If GT = "00:00", ✅ **Access Granted**, OUT = 1, GT updated to 23:59.
- If GT is not "00:00", ❌ **Access Denied**.

Scenario 2: Student Wants to Exit at 7:00 AM (CT = 7:00 AM)

- GT must be **≤ CT (7:00 AM or earlier)**.
- If GT = "06:30 AM", ❌ **Access Denied** (GT > CT).
- If GT = "06:00 AM", ✅ **Access Granted**, OUT = 1, GT updated to 23:59.

Scenario 3: Student Wants to Exit at 3:00 PM (CT = 3:00 PM)

- **No GT check is required** for this time range (8:30 AM – 6:00 PM).
- ✅ **Access Granted**, OUT = 1.

Scenario 4: Student Wants to Enter at Any Time

- ✅ **Access is always granted**, and OUT is updated to 0.

4.2 Software Implementation

The software implementation integrates cloud-based real-time databases, local servers, and a web-based management system. Below is a detailed breakdown of the key components.

4.2.1 Google Firebase Database

Google Firebase provides a cloud-based **NoSQL database** that enables real-time updates and synchronization across devices. It is used to track student entries/exits efficiently and ensure security.

Key Features in This Project

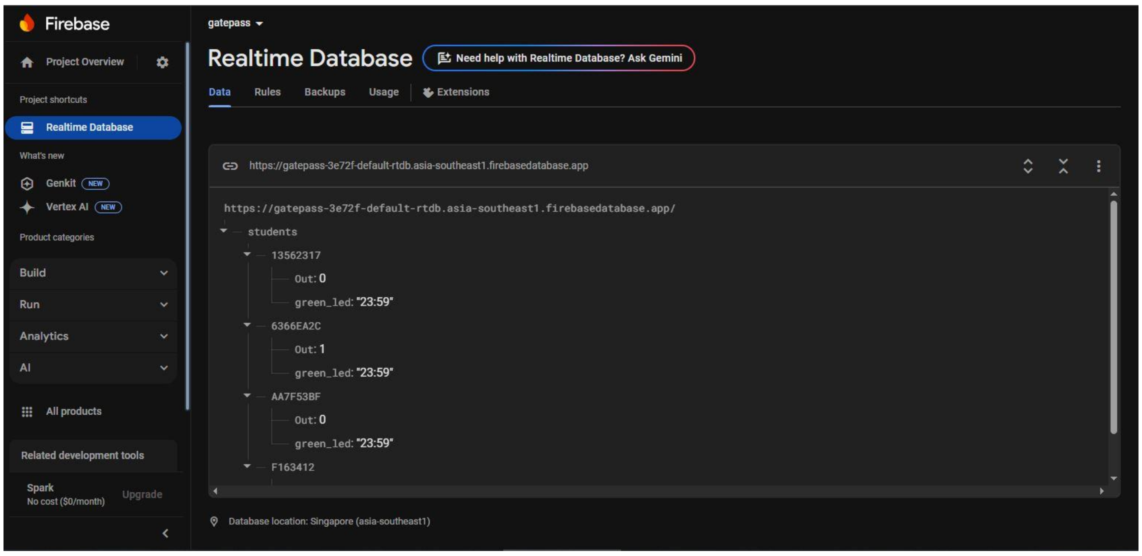
- **Real-time Synchronization:** Any change in access permissions (like green_led values) instantly updates all connected devices.
- **Data Structure (JSON-based):** Firebase follows a document-based format for easy retrieval and updates.
- **ESP32 Communication:** The RFID-based ESP32 microcontroller fetches and updates data in Firebase.
- **Security & Authentication:** Firebase security rules ensure that only authorized users can modify student records.

Firebase Database Structure

Firebase organizes student records under the students node:

```
{
  "students": {
    "13562317": {
      "Out": 0,
      "green_led": "23:59"
    },
    "6366EA2C": {
      "Out": 0,
      "green_led": "15:49"
    }
  }
}
```

- **RFID as Key:** Each student is uniquely identified by their **RFID tag**.
- **Out Field:** Represents whether the student is outside (1) or inside (0).
- **green_led Field:** Specifies the last permitted exit time.



```
https://gatepass-3e72f-default-rtdb.asia-southeast1.firebasedatabase.app/
students
    13562317
        Out: 0
        green_led: "23:59"
    6366EA2C
        Out: 1
        green_led: "23:59"
    AA7F53BF
        Out: 0
        green_led: "23:59"
    F163412
```

Figure 13 Firebase Dashboard

4.2.2 XAMPP Server & MySQL Database

XAMPP is used to **host the local web server** and manage the MySQL database, ensuring structured and efficient data storage.

Key Features in This Project

- **Structured Data Management:** SQL-based storage ensures reliability and efficiency.
- **Synchronization with Firebase:** Updates in MySQL also reflect in Firebase for real-time tracking.
- **Role-Based Access:** Different database tables handle student data, gate pass requests, and authentication.
-

Database Tables in MySQL

vidyamandira (Student Details Table)		
Column Name	**Data Type**	**Description**
rfid	VARCHAR(20)	Unique RFID tag of the student (Primary Key)
bhavan	VARCHAR(50)	Hostel/Bhavan name
roll	VARCHAR(4)	Student's Roll Number (Unique)

name	VARCHAR(50)	Student's Name
year_	VARCHAR(3)	Year of Study (PG1, UG3, etc.)
dept	VARCHAR(20)	Department Name
green_led	VARCHAR(5)	Last permitted exit time (Format: HH:MM)
Out	TINYINT(1)	0 = Inside, 1 = Outside

This table stores student information, including RFID, name, department, and access permissions.

gate pass_requests (Gate Pass Management)		
Column Name	**Data Type**	**Description**
id	INT (Auto Increment)	Unique request ID (Primary Key)
roll	VARCHAR(50)	Student's Roll Number (Foreign Key from vidyamandira)
request_time	VARCHAR(10)	Time of exit request (HH:MM)
reason	TEXT	Student's reason for requesting a gate pass
status	ENUM	‘Pending’, ‘Approved’, ‘Rejected’
created_at	TIMESTAMP	Timestamp of request creation

This table tracks students requesting permission to exit after the **permitted hours**.

student_users (Login Credentials for Students)		
Column Name	**Data Type**	**Description**
roll	VARCHAR(4)	Student's Roll Number (Primary Key)
password	VARCHAR(255)	Hashed password for secure authentication

This table stores **hashed passwords** for student logins.

wardens (Warden Login Table)		
Column Name	**Data Type**	**Description**
bhavan	VARCHAR(50)	Hostel/Bhavan name (Primary Key)
password	VARCHAR(255)	Hashed password for security

This table allows wardens to **log into the dashboard** and manage student records.

4.2.3 PHP & HTML for Website Management

The website allows **administrators, wardens, and students** to interact with the system, request gate passes, and monitor hostel movements.

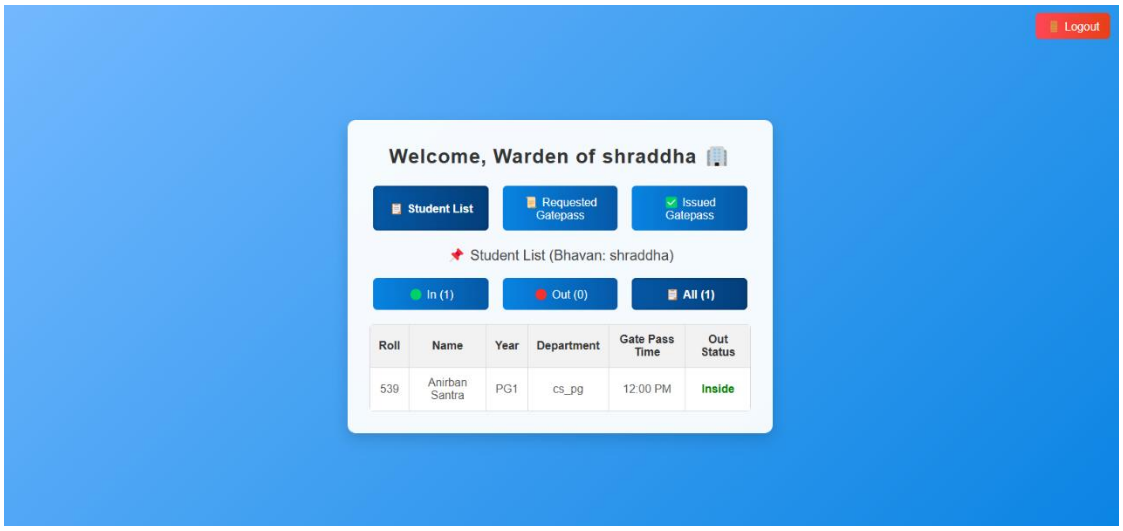
Key Features in This Project

- **Admin Dashboard:** Provides **real-time student movement tracking** and manual data entry capabilities.
- **Student Login:** Enables students to request **gate passes** and check their status.
- **Automated Updates:** When a student exits, their status (Out) is updated in both MySQL and Firebase.
- **Bootstrap UI Design:** Ensures a **responsive and user-friendly** interface.

Major Web Modules

1. Warden Panel

- **Student Management** – Add, update, or remove student details from the database.
- **Gate Pass Requests** – Approve or reject student requests for exit based on system rules.
- **Live Monitoring** – View real-time entry/exit logs of students.



Roll	Name	Year	Department	Gate Pass Time	Out Status
539	Anirban Santra	PG1	cs_pg	12:00 PM	Inside

Figure 14 Warden Dashboard

2. Admin Panel (Master Panel)

- **Warden & User Management** – Manage warden accounts and access permissions for different hostels.
- **Student Management** – Modify and Add new student to the system.
- **System Configuration** – Modify access control rules and security policies dynamically.
- **Emergency Control** – Override student access manually in case of emergencies.

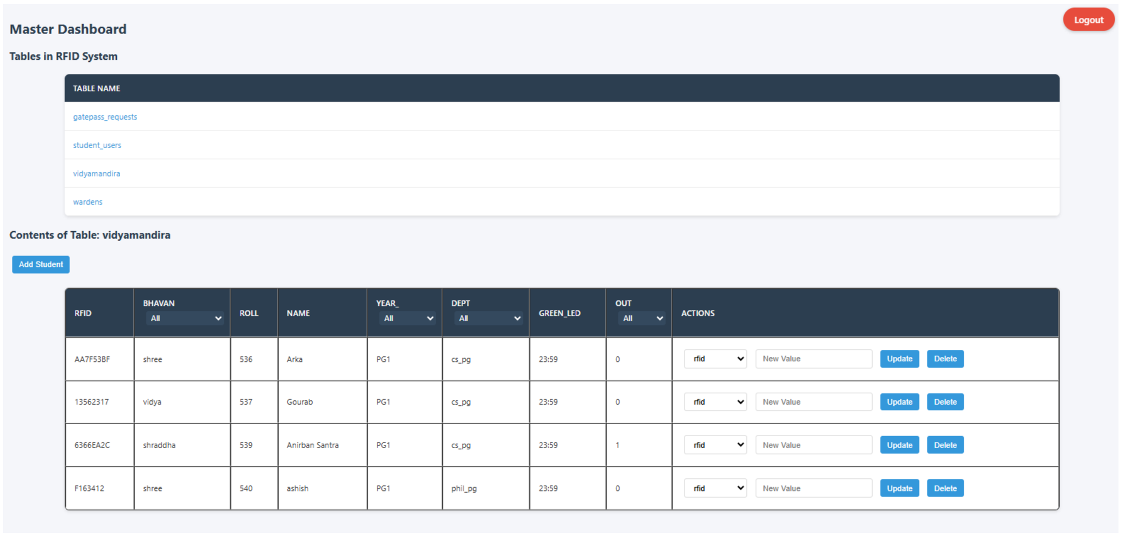


TABLE NAME
gatepass_requests
student_users
vidyamandira
wardens

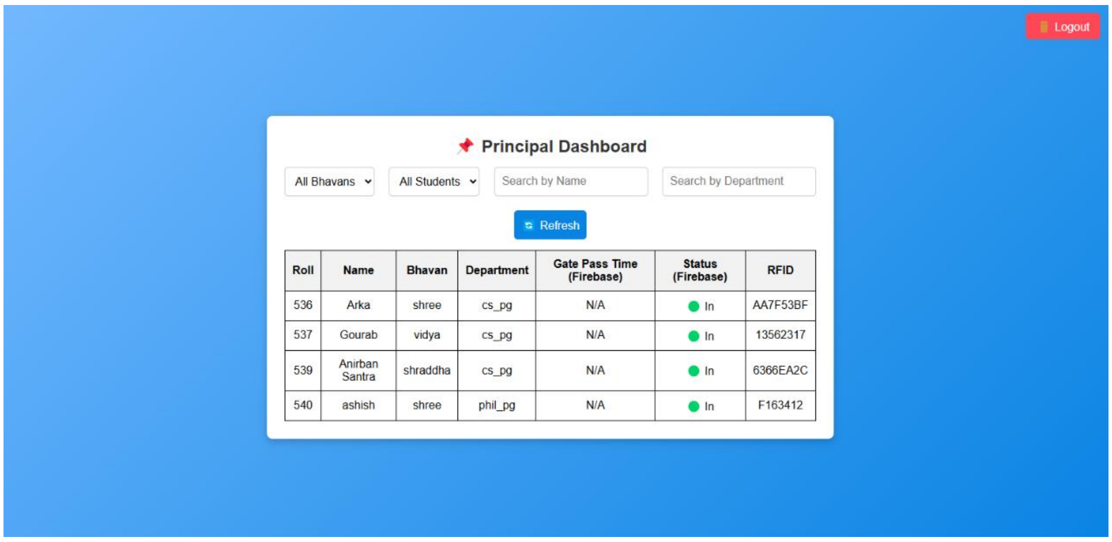
Contents of Table: vidyamandira

RFID	BHAVAN All	ROLL	NAME	YEAR_ All	DEPT All	GREEN_LED	OUT All	ACTIONS
AA7F53BF	shree	536	Arka	PG1	cs_pg	23:59	0	rfid New Value Update Delete
13562317	vidya	537	Gourab	PG1	cs_pg	23:59	0	rfid New Value Update Delete
6366EA2C	shraddha	539	Anirban Santra	PG1	cs_pg	23:59	1	rfid New Value Update Delete
F163412	shree	540	ashish	PG1	phil_pg	23:59	0	rfid New Value Update Delete

Figure 15 Master or Admin Dashboard

3. Principal Panel

- **Overview Dashboard** – Get a complete view of student attendance, movements, and gate pass requests.

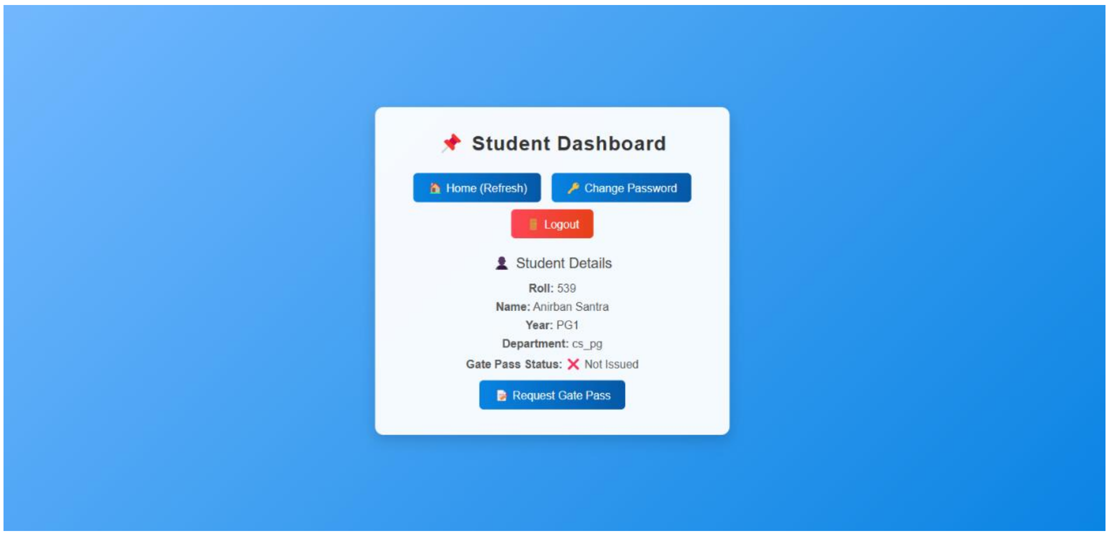


Roll	Name	Bhavan	Department	Gate Pass Time (Firebase)	Status (Firebase)	RFID
536	Arka	shree	cs_pg	N/A	● In	AA7F53BF
537	Gourab	vidya	cs_pg	N/A	● In	13562317
539	Anirban Santra	shraddha	cs_pg	N/A	● In	6366EA2C
540	ashish	shree	phil_pg	N/A	● In	F163412

Figure 16 Principal Dashboard

4. Student Panel

- **Secure Login** – Students log in using their credentials.
- **Gate Pass Requests** – Submit requests for leaving beyond standard permitted hours.
- **Status Updates** – Track whether their gate pass is approved or rejected.



Roll: 539
Name: Anirban Santra
Year: PG1
Department: cs_pg
Gate Pass Status: ✗ Not Issued

Figure 17 Student Dashboard

5. RFID Integration with ESP32

- **RFID Scanning** – Reads the student's RFID card when they scan at the gate.
- **Database Check** – Compares scanned data with Firebase and MySQL to determine access eligibility.
- **Access Decision** – Grants or denies entry based on green_led timing rules.
- **Automatic Updates** – If access is granted, the student's **Out** status is updated in Firebase and MySQL.

5 Chapter Demonstration

Objectives

- Demonstrate RFID-based access control in real time.
- Validate Dashboard data with esp32 in real time.
- Ensure smooth functionality & check for exceptions.

5. Demonstration

In this section, we will demonstrate a how our system works in details step by step.

5.1 Homepage

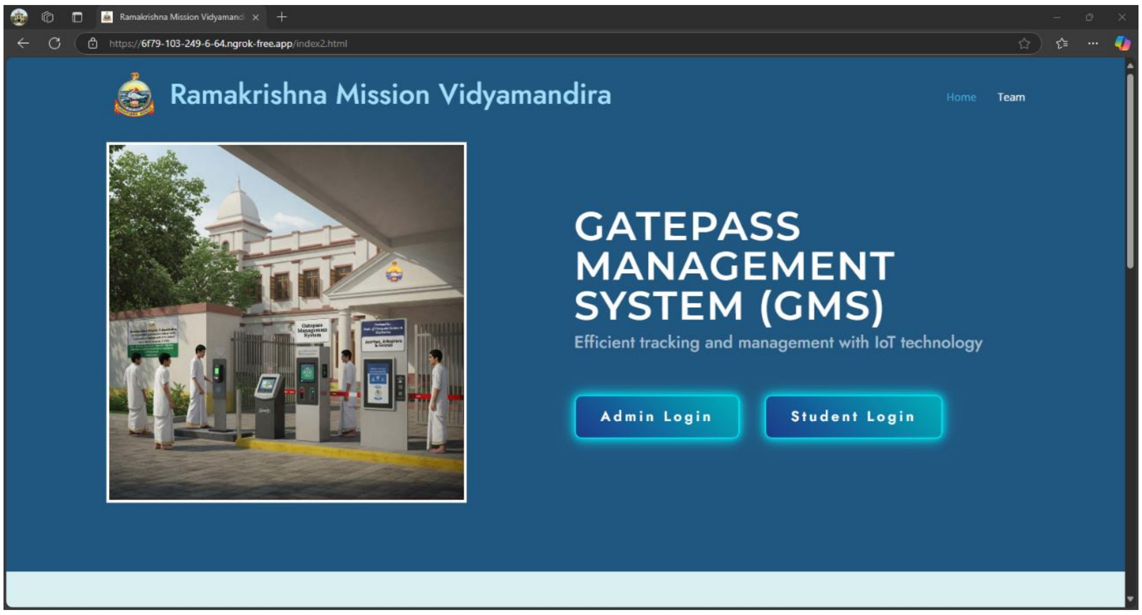

Ramakrishna Mission Vidyamandira
Home
Team
GATEPASS MANAGEMENT SYSTEM (GMS)
Efficient tracking and management with IoT technology
Admin Login
Student Login

Figure 18 Homepage

5.2 Admin Login

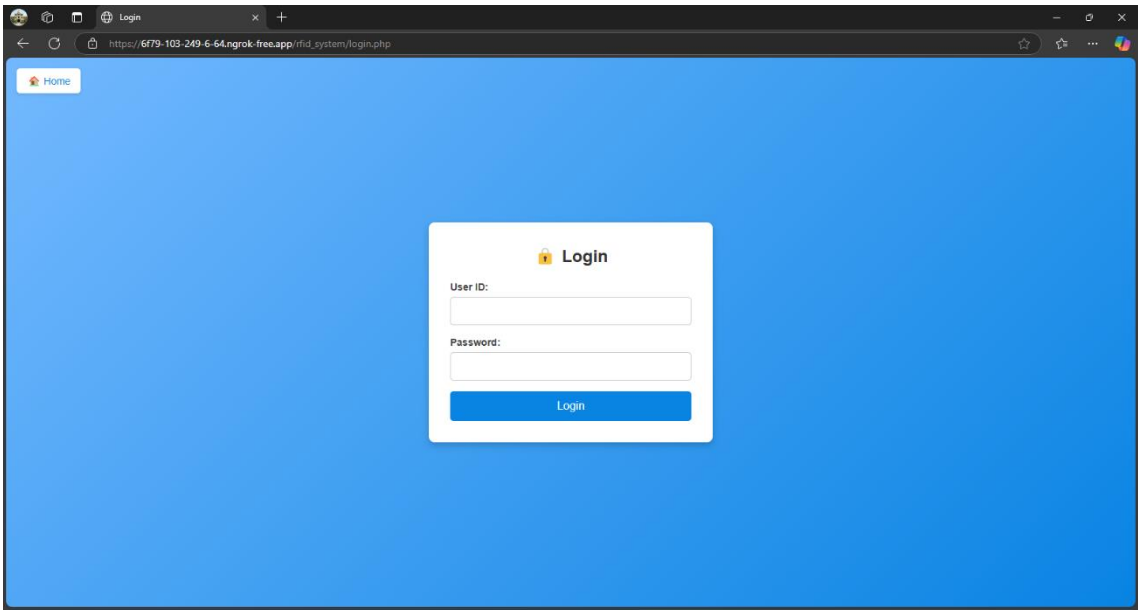


Figure 19 Admin Login Page for Wardens, Master and Principal

5.3 Student Add in Master Dashboard

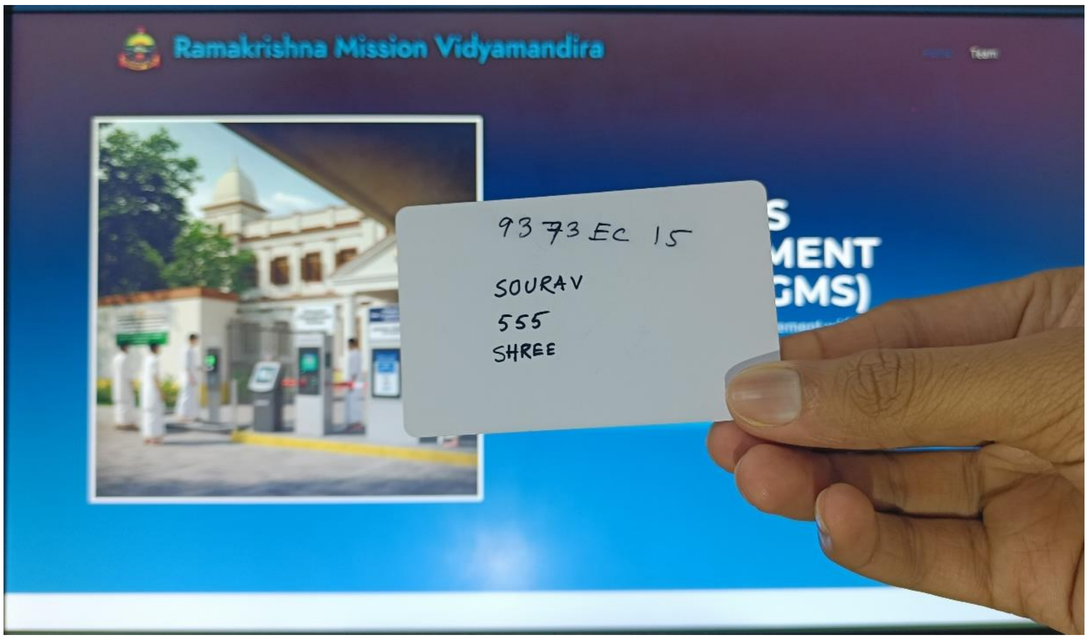
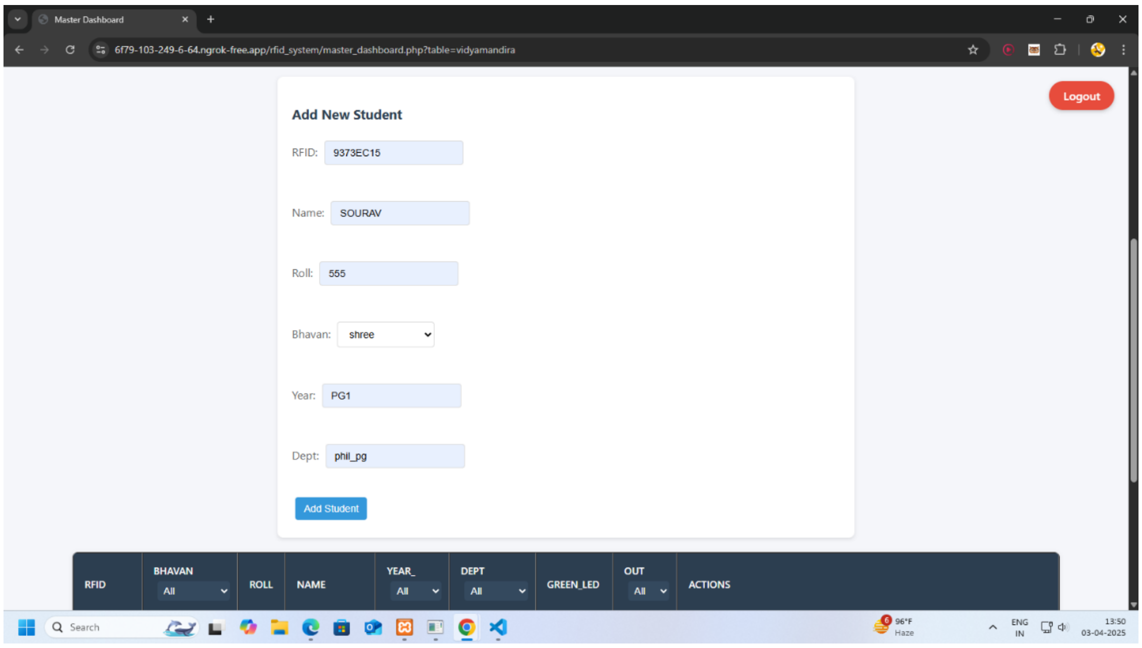


Figure 20 Student ID Card



Master Dashboard

6f79-103-249-6-64.ngrok-free.app/rfid_system/master_dashboard.php?table=vidyamandira

Logout

Add New Student

RFID: 9373EC15

Name: SOURAV

Roll: 555

Bhavan: shree

Year: PG1

Dept: phil_pg

Add Student

RFID	BHAVAN All	ROLL	NAME	YEAR_ All	DEPT All	GREEN_LED	OUT All	ACTIONS

Figure 21 Adding New Student

<u>5.4 Student Login & Requests Gatepass</u>

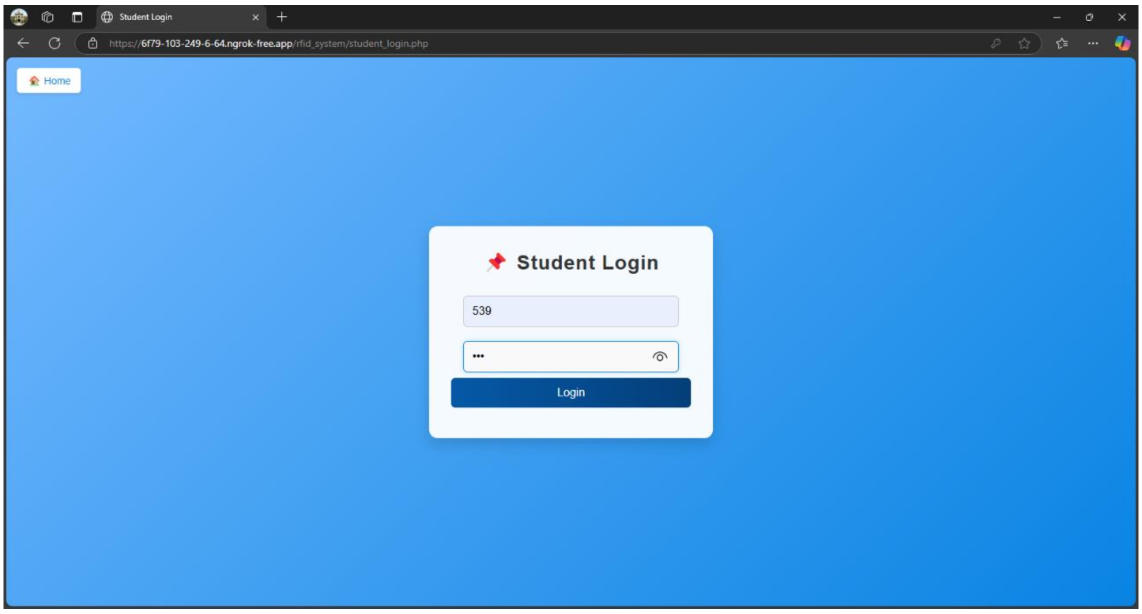
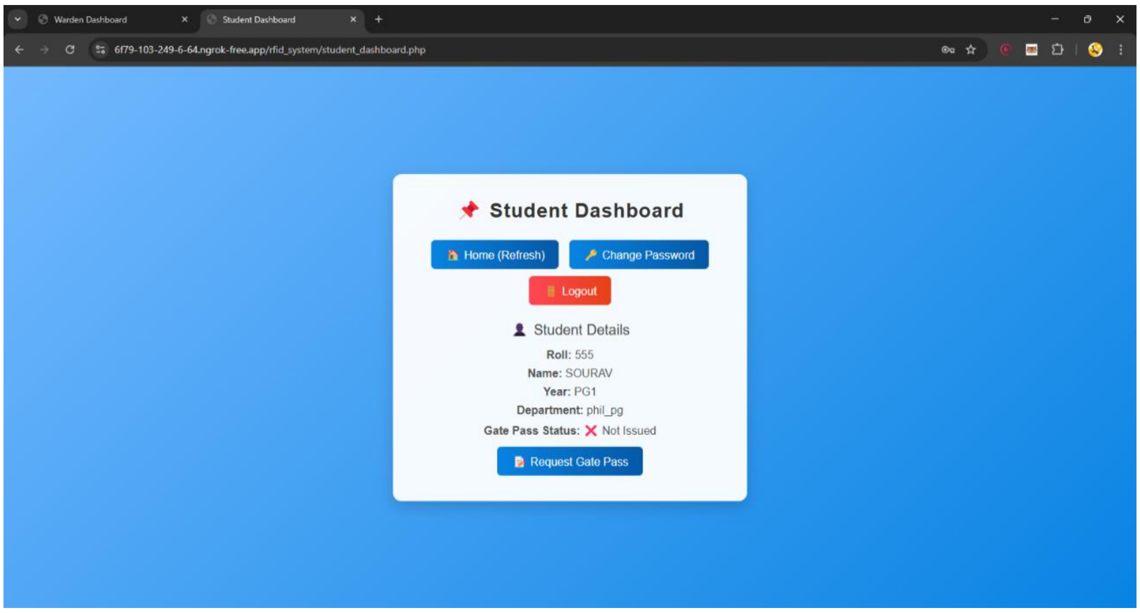
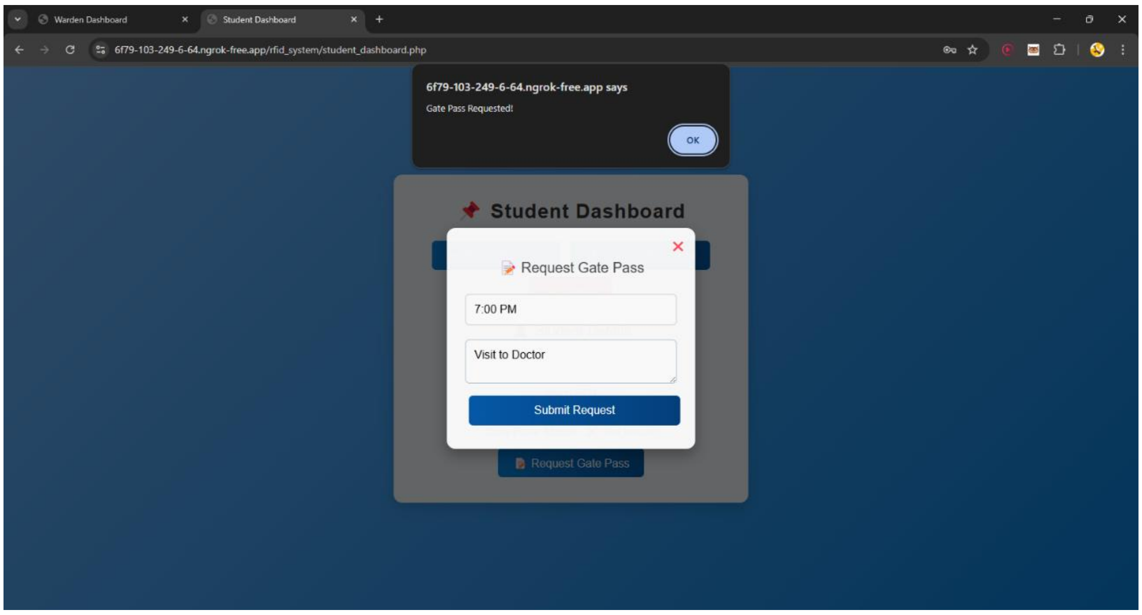

Student Login
https://6f79-103-249-6-64.ngrok-free.app/rfid_system/student_login.php
Home
Student Login
539
•••
Login

Figure 22 Student Login Page



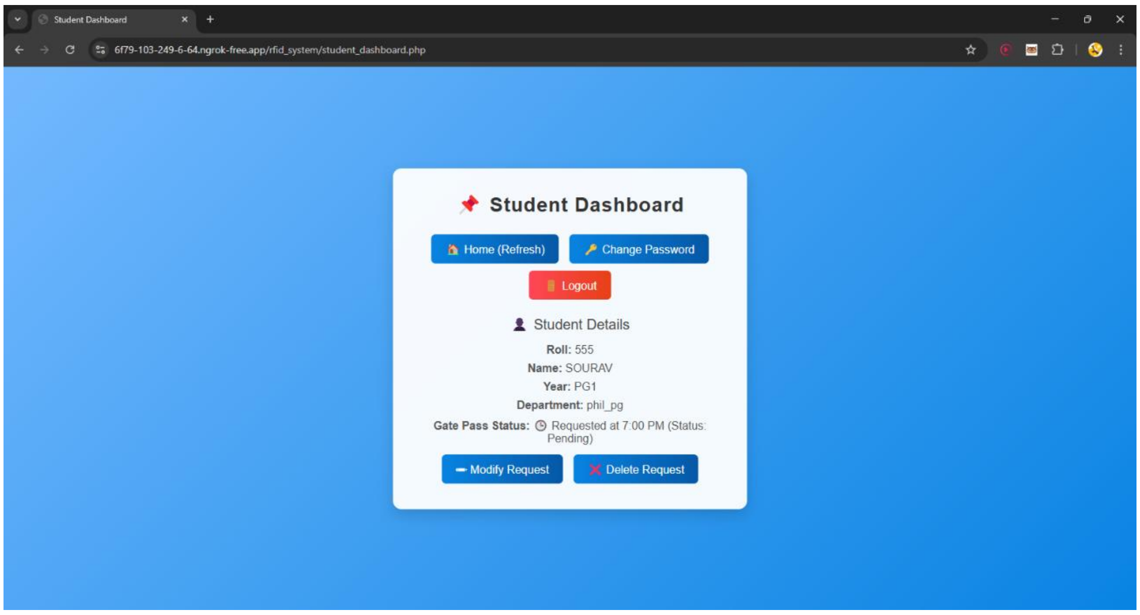
Roll: 555
Name: SOURAV
Year: PG1
Department: phil_pg
Gate Pass Status: ❌ Not Issued

Figure 23 Student Dashboard


6f79-103-249-6-64.ngrok-free.app says

Gate Pass Requested!

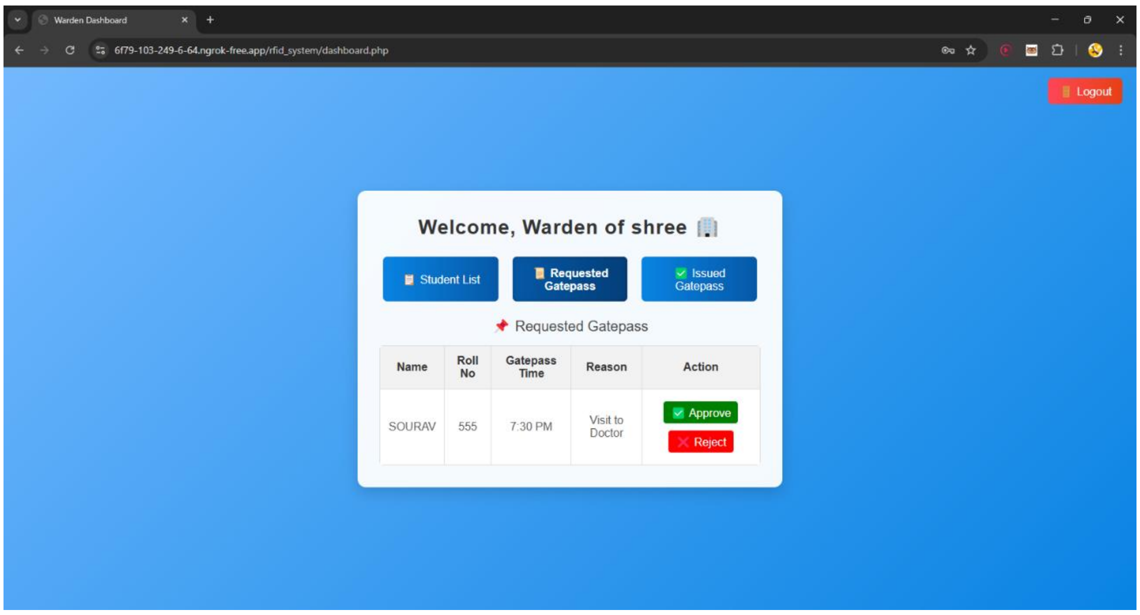
Figure 24 Filling Details for Gate pass



Roll: 555
Name: SOURAV
Year: PG1
Department: phil_pg
Gate Pass Status: Requested at 7:00 PM (Status: Pending)

Figure 25 Dashboard after Gate pass request

5.5 Warden Dashboard & Accept Gatepass

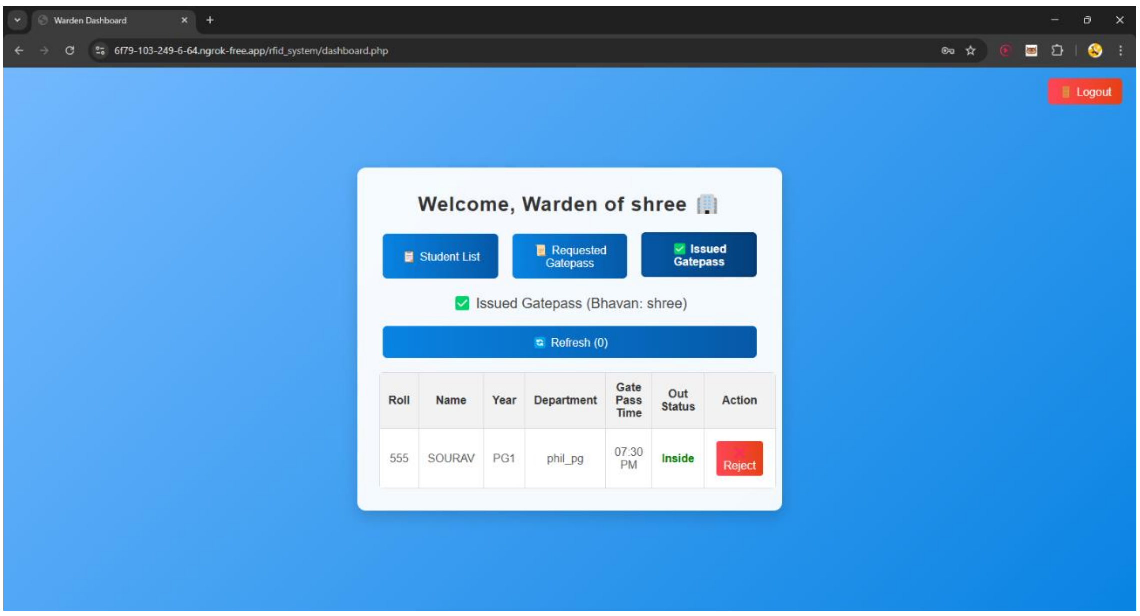


Welcome, Warden of shree

Requested Gatepass

Name	Roll No	Gatepass Time	Reason	Action
SOURAV	555	7:30 PM	Visit to Doctor	Approve Reject

Figure 26 Warden Dashboard - Requested Gate pass

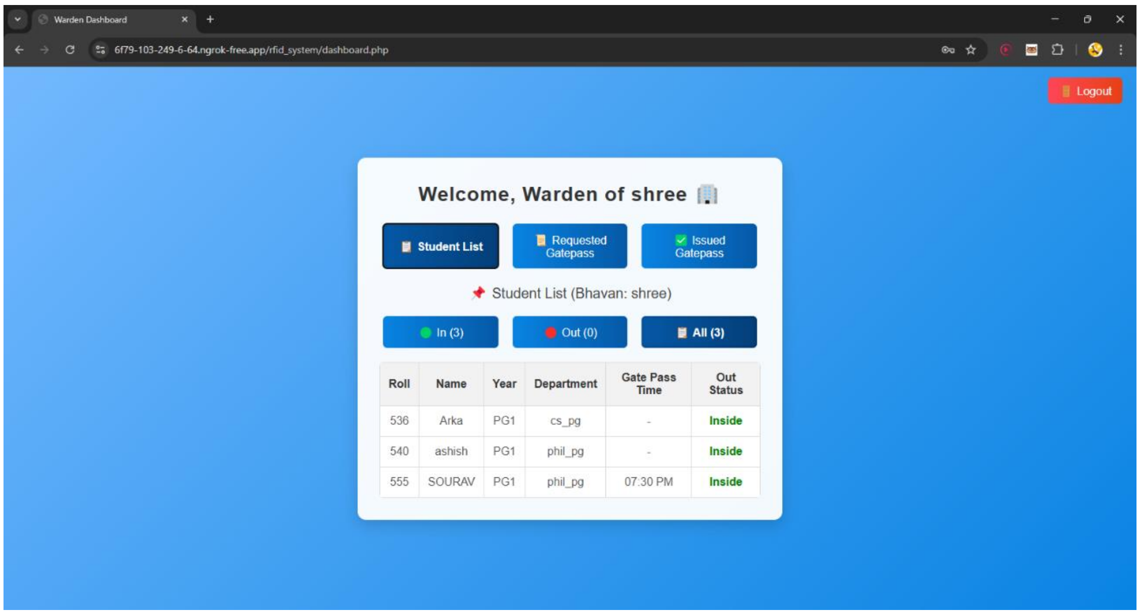


Welcome, Warden of shree

Issued Gatepass (Bhavan: shree)

Roll	Name	Year	Department	Gate Pass Time	Out Status	Action
555	SOURAV	PG1	phil_pg	07:30 PM	Inside	Reject

Figure 27 Warden Dashboard - Issued Gate pass



Welcome, Warden of shree 🏢

📌 Student List (Bhavan: shree)

Roll	Name	Year	Department	Gate Pass Time	Out Status
536	Arka	PG1	cs_pg	-	Inside
540	ashish	PG1	phil_pg	-	Inside
555	SOURAV	PG1	phil_pg	07:30 PM	Inside

Figure 28 Warden Dashboard - All Students List

5.6 Student Goes Out with RFID Card

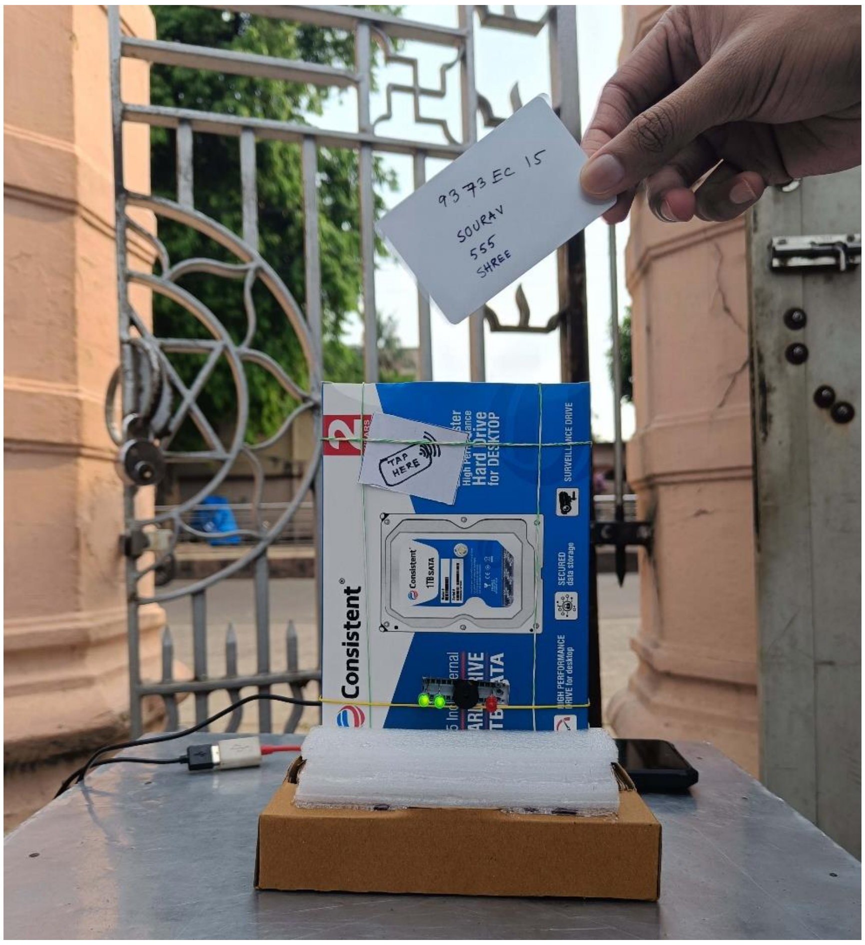
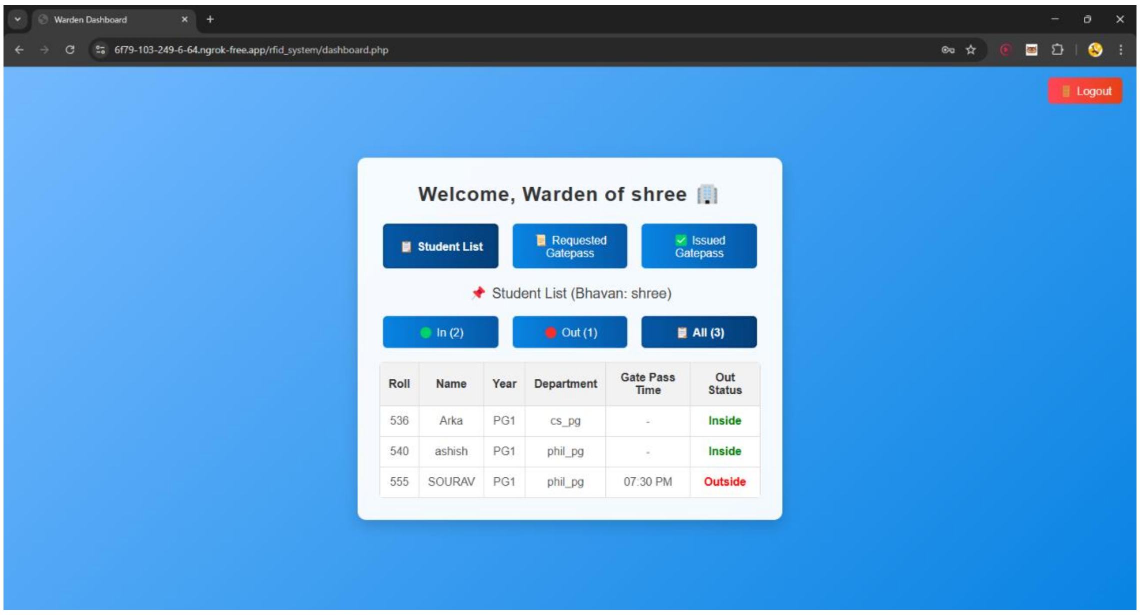


Figure 29 Student Goes Out with his ID Card as his Gate Pass

5.7 Warden & Principal Dashboard shows updated Status



Warden Dashboard

6f79-103-249-6-64.ngrok-free.app/rfid_system/dashboard.php

Logout

Welcome, Warden of shree

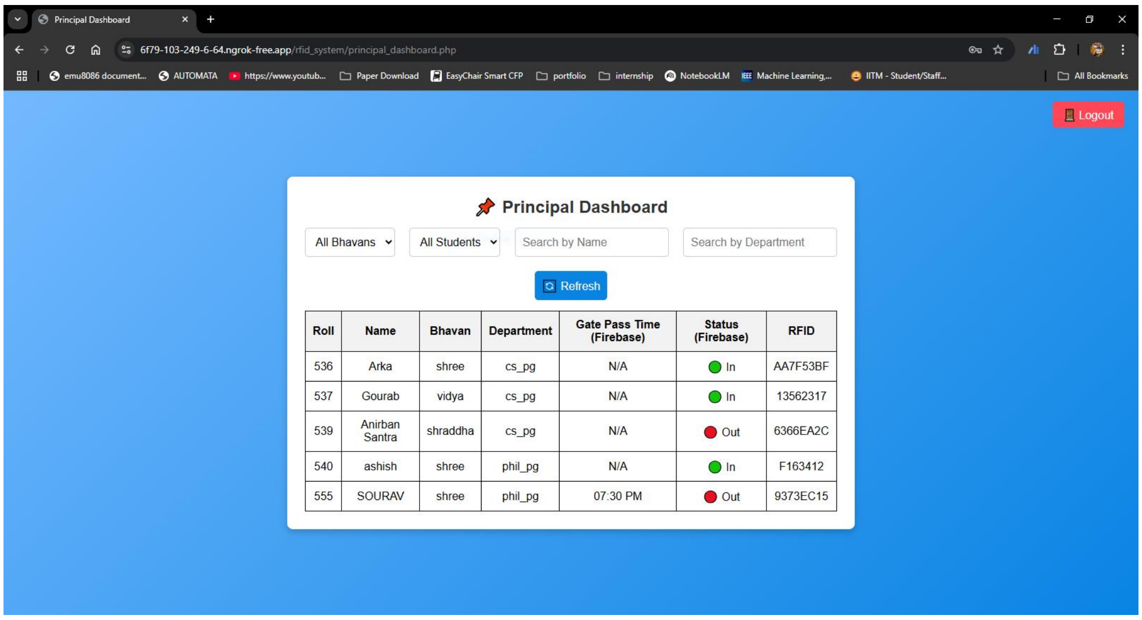
Student List | Requested Gatepass | Issued Gatepass

Student List (Bhavan: shree)

In (2) | Out (1) | All (3)

Roll	Name	Year	Department	Gate Pass Time	Out Status
536	Arka	PG1	cs_pg	-	Inside
540	ashish	PG1	phil_pg	-	Inside
555	SOURAV	PG1	phil_pg	07:30 PM	Outside

Figure 30 Warden Dashboard shows his status as Outside



Principal Dashboard

6f79-103-249-6-64.ngrok-free.app/rfid_system/principal_dashboard.php

emu8086 document... AUTOMATA https://www.youtub... Paper Download EasyChair Smart CFP portfolio internship NotebookLM Machine Learning,... IITM - Student/Staff... All Bookmarks

Logout

Principal Dashboard

All Bhavans | All Students | Search by Name | Search by Department

Refresh

Roll	Name	Bhavan	Department	Gate Pass Time (Firebase)	Status (Firebase)	RFID
536	Arka	shree	cs_pg	N/A	🟢 In	AA7F53BF
537	Gourab	vidya	cs_pg	N/A	🟢 In	13562317
539	Anirban Santra	shraddha	cs_pg	N/A	🔴 Out	6366EA2C
540	ashish	shree	phil_pg	N/A	🟢 In	F163412
555	SOURAV	shree	phil_pg	07:30 PM	🔴 Out	9373EC15

Figure 31 Principal Dashboard shows that he is outside

6 Chapter Conclusion

Objectives

- Discuss the Achievements of the project
- Highlight the Shortcomings.
- Bring out the room for improvements.

6. Conclusion

The implementation of an RFID-based student tracking system for hostel management has significantly enhanced the efficiency, security, and accuracy of monitoring student movements. This system successfully replaces traditional manual register-based tracking with an automated, digital approach that ensures real-time updates, centralized control, and reduced administrative workload.

By leveraging ESP32 microcontrollers, RFID technology, Firebase real-time databases, and a web-based dashboard, the system facilitates seamless entry and exit management. Predefined rules govern hostel timings, enabling students to exit the hostel without requiring manual approvals, while late exits are managed via a structured gate pass system. Additionally, the integration of MySQL with Firebase ensures a robust backend, allowing wardens and administrators to monitor student activities effortlessly.

Despite these achievements, certain limitations exist, such as dependency on internet connectivity, the need for additional security enhancements, and the absence of a dedicated mobile application. Nevertheless, this project establishes a strong foundation for future improvements in hostel security and student monitoring systems.

6.1 Achievements

- **Automated Entry/Exit Management:** The system eliminates manual logbooks and ensures accurate, real-time tracking of student movements using RFID technology.
- **Seamless Database Synchronization:** Student entry and exit records are automatically updated across both Firebase and MySQL, ensuring consistency in data storage.
- **Efficient Gate Pass System:** Students can request permission for late exits through an online system, reducing paperwork and manual verification efforts.
- **Real-Time Monitoring and Alerts:** The system enables hostel authorities to remotely monitor student activity, facilitating quick responses to any rule violations.
- **Secure and Role-Based Access Control:** Separate access levels for students, wardens, and administrators enhance data security and usability.
- **User-Friendly Web Interface:** Wardens can approve gate passes, check student statuses, and manage hostel activities through an intuitive dashboard.
- **Paperless and Environmentally Friendly:** The elimination of physical registers and gate passes contributes to an eco-friendly digital solution.

6.2 Shortcomings

- **Dependency on Internet Connectivity:** The system requires an active internet connection for real-time updates, making synchronization vulnerable to network failures.

- **Limited Offline Support:** While the RFID reader functions independently, entry and exit logs cannot be stored locally and must sync with Firebase.
- **Hardware Limitations:** The efficiency of RFID detection may be affected by interference, incorrect tag placement, or low power supply.
- **Manual Rule Modification:** Any changes to hostel exit timing rules must be manually updated in the database rather than dynamically configured via the dashboard.
- **Lack of Biometric Security:** Currently, access is granted solely based on RFID tags, which can be misused if borrowed or lost.
- **Absence of a Mobile Application:** The system is web-based, requiring students and wardens to access the dashboard via a browser rather than a dedicated mobile application.

6.3 Future Scope

- **Development of a Mobile Application:** A dedicated mobile application for Android and iOS would allow students to request gate passes and check their status with greater convenience.
- **Implementation of Offline Data Logging:** Local storage in ESP32 devices will enable the system to temporarily store entry and exit logs during internet downtimes and synchronize them once connectivity is restored.
- **Integration of Biometric Authentication:** Fingerprint scanners or facial recognition technology can be incorporated to ensure that only the authorized student uses the assigned RFID card.
- **Automated Alert System:** Real-time notifications via SMS, email, or push notifications can be sent to students and wardens in cases of rule violations or delayed returns.
- **AI-Based Student Movement Analytics:** Machine learning algorithms can be employed to analyze movement patterns and detect anomalies, such as students returning too late or engaging in unauthorized exits.
- **Integration with the Hostel Attendance System:** The RFID system can be connected to the college’s main attendance database to track student participation in academic and extracurricular activities.
- **Implementation of Dynamic Rule Configuration:** Wardens can be allowed to modify hostel exit rules dynamically via the dashboard, eliminating the need for manual database updates.
- **Introduction of Voice-Based Assistance:** An AI-powered chatbot can be implemented to assist students and wardens with queries related to gate passes, hostel timings, and rule violations.

References

Books:

1. O. Hersent, D. Boswarthick, & O. Elloumi, *The Internet of Things: Key Applications and Protocols*, Wiley, 2012.
2. K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication*, Wiley, 2010.
3. S. Monk, *Programming the ESP32: Getting Started with the ESP32*, McGraw-Hill, 2020.

Research Papers:

4. A. Juels, “RFID Security and Privacy: A Research Survey,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 381–394, 2006.
5. D. M. Dobkin, *The RF in RFID: Passive UHF RFID in Practice*, Elsevier Science, 2012.
6. L. Atzori, A. Iera, & G. Morabito, “The Internet of Things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

Online Documentation:

7. Espressif Systems, “ESP32 Series Datasheet,” Available at: https://www.espressif.com/en/products/socs/esp32
8. Google Firebase Documentation, Available at: https://firebase.google.com/docs
9. XAMPP Documentation, Available at: https://www.apachefriends.org/index.html
10. W3Schools, “HTML, CSS, JavaScript, PHP, SQL Tutorials,” Available at: https://www.w3schools.com

YouTube Video Tutorials:

11. **ESP32 and RFID RC522 Tutorial with LCD Screen** – Available at: https://www.youtube.com/watch?v=tDQD2e2tLls
12. **Easy ESP32 + Firebase Realtime: Newest Connection Guide! 2024** – Available at: https://www.youtube.com/watch?v=qukrE8RZyZw
13. **XAMPP Installation on Windows - Easy Setup Tutorial** – Available at: https://www.youtube.com/watch?v=CDl2Dl9Cwd0
14. **Build Secure IoT Projects: ESP32 Firebase Authentication with ESP-IDF** – Available at: https://www.youtube.com/watch?v=3HVMTXCsTAY
15. **PHP MySQL Tutorial: Create a Webserver on Your Computer with XAMPP** – Available at: https://www.youtube.com/watch?v=lB9aKoFjwkg

Appendix

Device Picture

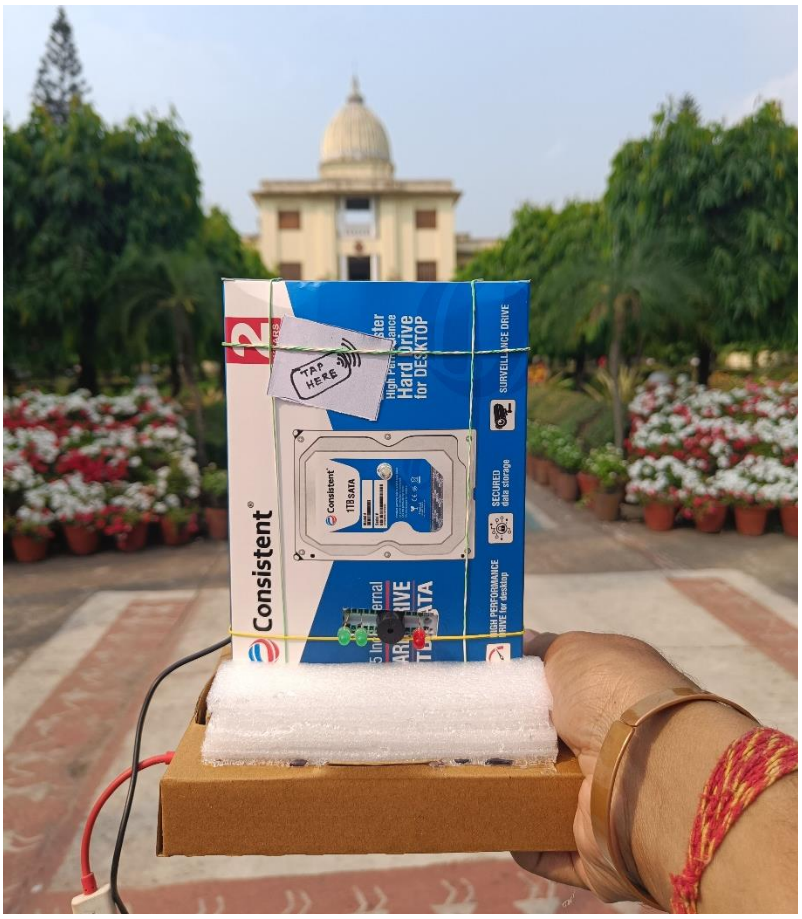
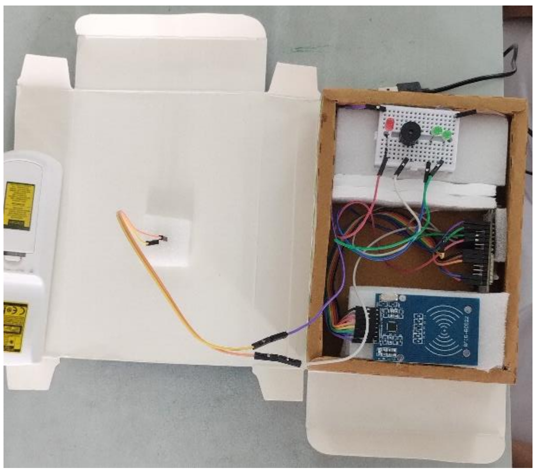


Figure 32 Device

Device Breakdown



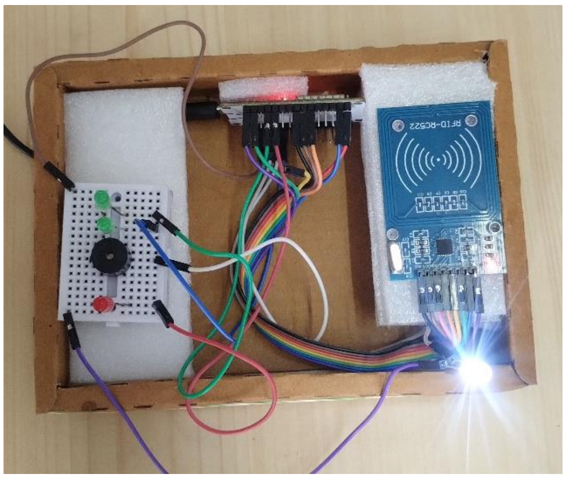


Figure 33 Internal Structure of The Device

Figure 34 Circuit Display

Project Codes

The project has over 2000 lines of codes and has been uploaded for open access in GitHub and hence can be accessed via the links below.

https://github.com/gourab21/GMS-Gate-Pass-Management-System

Scan this QR Code to access the Files.