

# **Project Report**

## **Course: Database Systems**

## **Topic: Network Access Registration system**

Team Members:

N. Priyadarshini (CS22B1009)

G. Satya Priya (CS22B1012)

K. Sri Harsha Priya (CS22B1017)

P. Neelima (CS22B1032)

CH. Gayathri (CS22B2018)

NARS, the Network Access Registration System, emerges as a transformative solution aimed at simplifying and enhancing the process of granting campus network access to student-owned devices a sophisticated web-based platform designed to revolutionize LAN registry. Gone are the days of physical presence – with our system, students can seamlessly log in using their email credentials, providing essential details including MAC and IP addresses." To achieve this, we embarked on front-end development using HTML and CSS. Backend is done using phpMyAdmin to create database and php to connect the database to frontend.

## **SRS (Software requirement specification)**

### **1. Introduction:**

NARS aims to simplify and enhance the process of granting campus network access to devices owned by students. This document provides detailed outlines of the development the requirements for NARS to ensure it meets the need of users Efficiently and Effectively.

## **2. Purpose:**

The purpose of NARS is to provide a user-friendly and efficient solution for managing network access and device registration within the campus environment. By automating processes related to device registration, access control, and issue tracking, NARS aims to streamline administrative tasks, improve connectivity management, and ensure a secure and reliable network infrastructure for students. Ultimately, the goal of NARS is to enhance the overall campus experience by simplifying network access procedures and promoting a more connected and digitally integrated campus community.

## **3. Scope:**

NARS will incorporate modules for device registration, user management, network access control, reporting, and system administration. It will support both manual and automated registration processes. NARS will support integration with existing campus systems and ensure compatibility with various devices and network configurations, aiming to provide a comprehensive solution for efficient campus network management.

## **4. Functional Requirements:**

### **4.1 User Management**

- i. Allow students to log in using their unique identifiers such as college roll numbers or student IDs, names, etc.
- ii. Provide administrative staff with the ability to manage user accounts, including creating, editing, and deactivating accounts.
- iii. Define user roles and permissions based on the type of user (e.g., student, faculty, administrative staff) to control access to features and data within NARS.

## **4.2 Device Registration**

- i. Enable students to register their devices by providing necessary details such as device type, MAC address, and owner information such as id.
- ii. Allocate devices to specific categories or groups for better organization and management.

## **4.3 Network Access Management**

- i. Manage network access permissions for registered devices, specifying access status (e.g., granted, denied) and access levels (e.g., full access, restricted access, guest access, No access etc).
- ii. Track and log network access activities, including timestamps and access status changes.

## **4.4 Issue**

- i. Allow students to report network connectivity issues or device registration problems.
- ii. Capture detailed information about reported issues, including severity, description, and status.
- iii. Assign reported issues to designated staff members for resolution and track their progress.

## **4.5 Solution**

- i. Generate reports on device registration status, network access permissions, and issue resolution progress.
- ii. Provide customizable reports for administrators to monitor network activity, identify trends, and assess system performance.
- iii. Support exporting reports in various formats (e.g., PDF, TXT) for further analysis and sharing.

## **5. Non-Functional Requirements:**

## **5.1 Performance**

- i. Ensure NARS can handle concurrent user access and large volumes of device registration data without experiencing performance degradation.
- ii. Maintain response times for critical functions, such as user authentication and device registration, within acceptable limits to ensure a smooth user experience.

## **5.2 Scalability**

- i. Design NARS to scale horizontally and vertically to accommodate the growing number of students and registered devices within the campus environment.
- ii. Implement mechanisms to handle increased user demand and data volumes without compromising system performance or reliability.

## **5.3 Security**

- i. Implement robust authentication and authorization mechanisms to ensure that only authorized users can access NARS and perform designated actions.
- ii. Encrypt data transmission between client devices and the NARS server, as well as data storage within the system, to maintain confidentiality and integrity of sensitive information.

## **5.4 Usability**

- i. Design NARS with an intuitive user interface that offers clear navigation and easy access to key features, such as device registration and issue reporting.
- ii. Provide informative feedback messages throughout the system to guide users and help them understand their actions and the system's responses.

- iii. Offer comprehensive documentation and user training resources to assist users in understanding NARS functionality and best practices for using the system effectively.

## **6. Constraints:**

The NARS project must adhere to the following constraints:

### **6.1 Technology Compatibility**

Develop NARS using technologies and platforms that are compatible with the organization's existing infrastructure and IT policies. Ensure compatibility with the campus network architecture, server configurations, and security protocols.

### **6.2 Regulatory Compliance**

Adhere to relevant industry standards and regulatory requirements governing network access management and data protection. Ensure compliance with data privacy regulations, such as GDPR to safeguard user information collected within NARS.

## **7. Glossary:**

NARS: Network Access Registration System

MAC: Media Access Control address

IP: internet protocol address

PDF: Portable Document Format

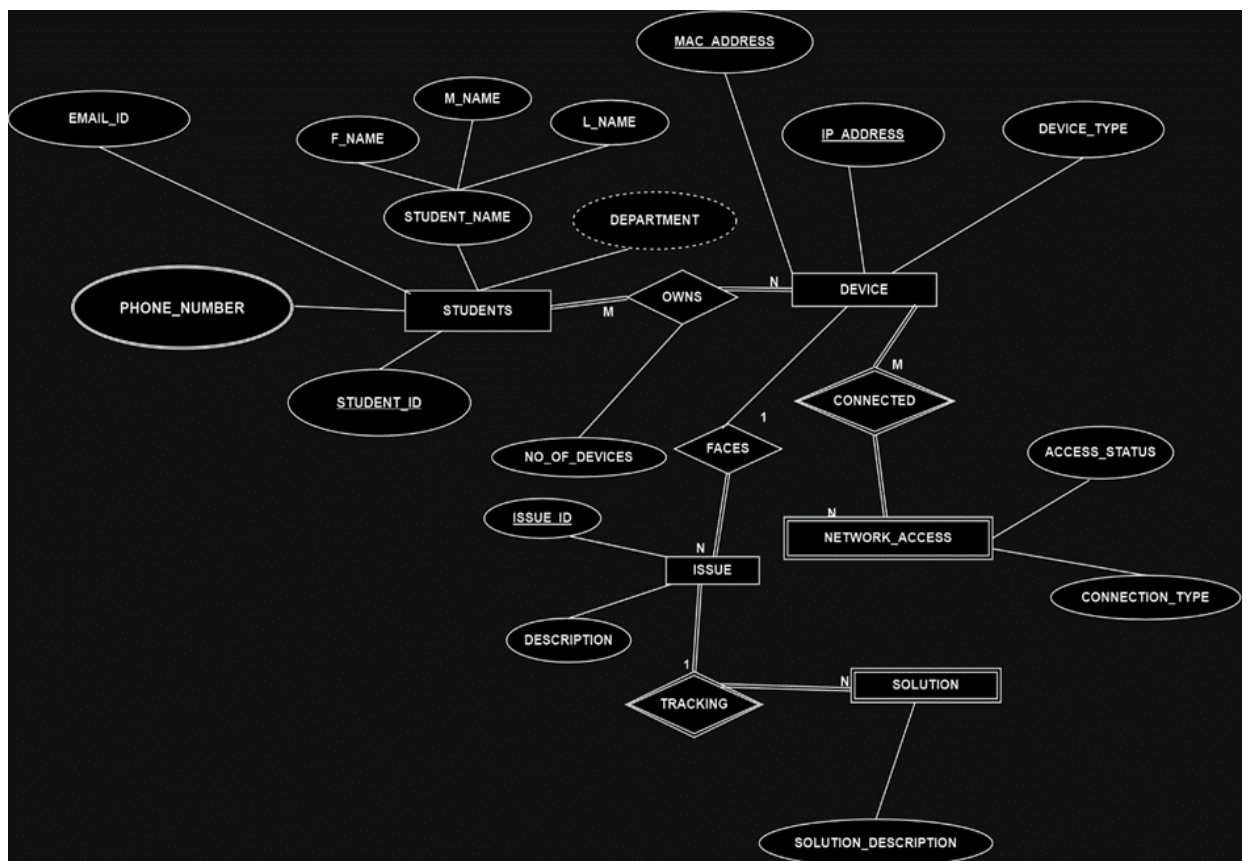
TXT: Text file format

## **8. Conclusion:**

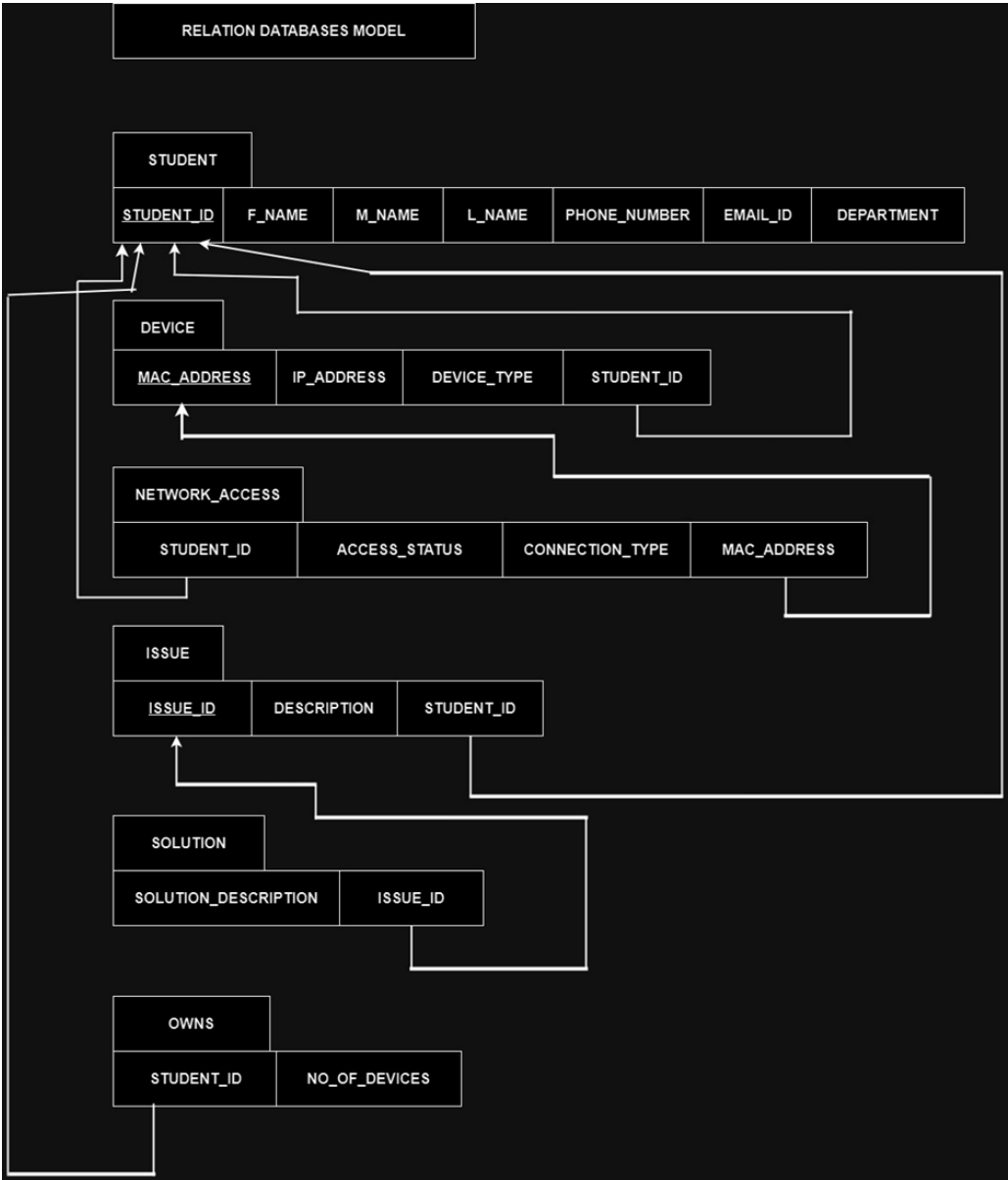
The Network Access Registration System (NARS) described in this document will be instrumental in streamlining network access management within the campus environment. By fulfilling the

outlined requirements, NARS will facilitate seamless registration of devices and efficient monitoring of network access, leading to improved operational efficiency and enhanced security. Through its implementation, NARS will contribute to maintaining the integrity and reliability of the campus network infrastructure.

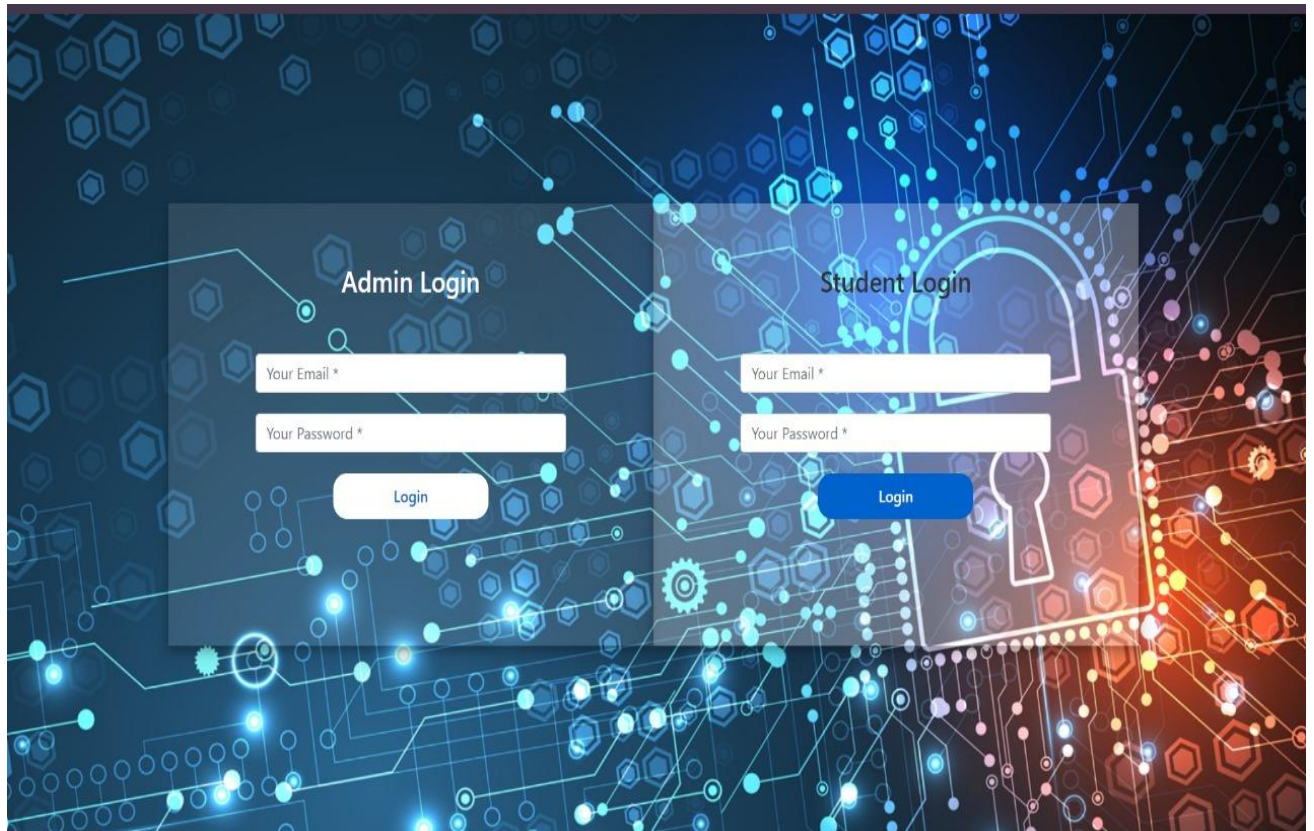
## ER diagram:



# Relational Database:

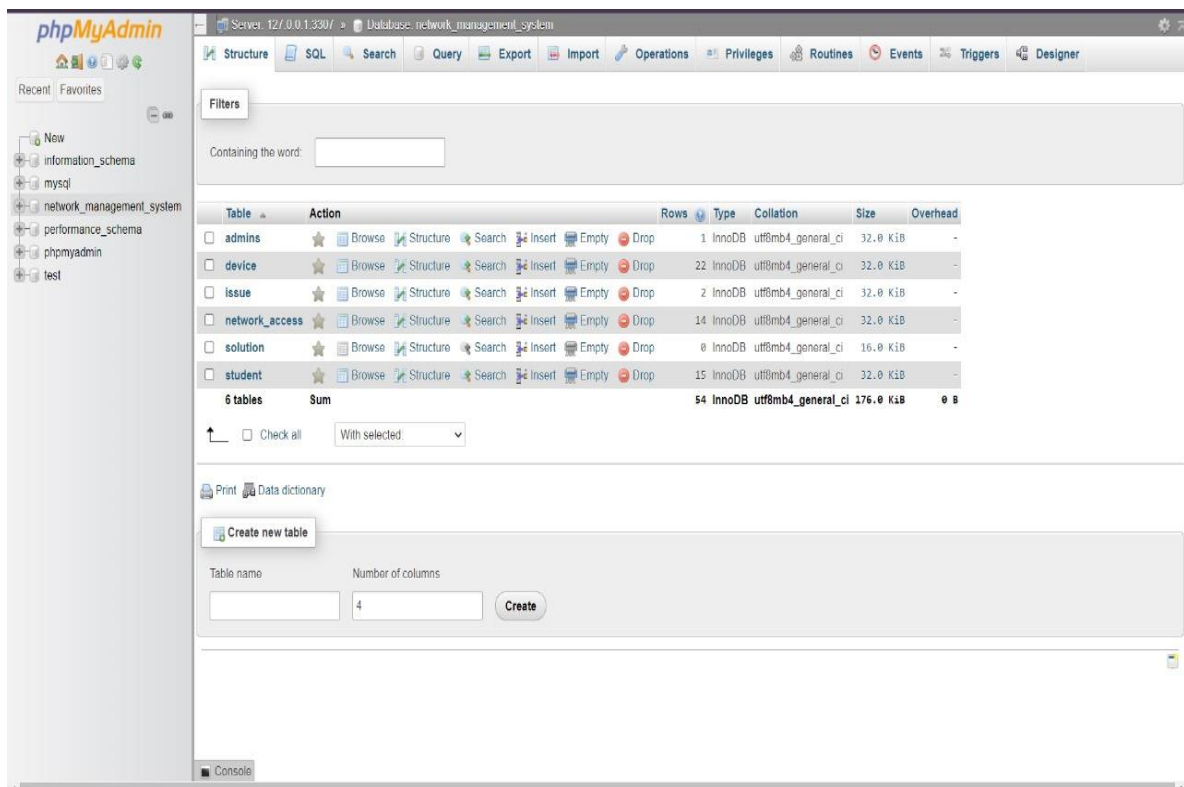


## Web page description:



The image above depicts the frontend interface of our project, a webpage utilized by both students to request LAN access and administrators to grant LAN permissions.





The image above illustrates the backend infrastructure of our project, comprising entities such as student, device, network\_access, issue, and solution, where pertinent information provided by students during webpage login is seamlessly integrated. This data is efficiently stored within this database, ensuring a structured and organized repository for future reference and management by administrators.

## METHODOLOGY:

### Frontend:

- **HTML (Hypertext Markup Language):** Used for structuring our web page (NARS) by defining the content and layout. We used HTML tags like <header>, <nav>, <div>, and <form> to structure the different parts of our webpage.
- **CSS (Cascading Style Sheets):** It is Responsible for styling the HTML elements, including layout, fonts, Colors, and overall visual presentation.

for our webpage(frontend) by writing CSS rules like color, font-size, margin, and padding to make everything look just right.

- JavaScript (JS): Adds interactivity and dynamic Behaviour to web pages, enabling features such as form validation, animations, and event handling.

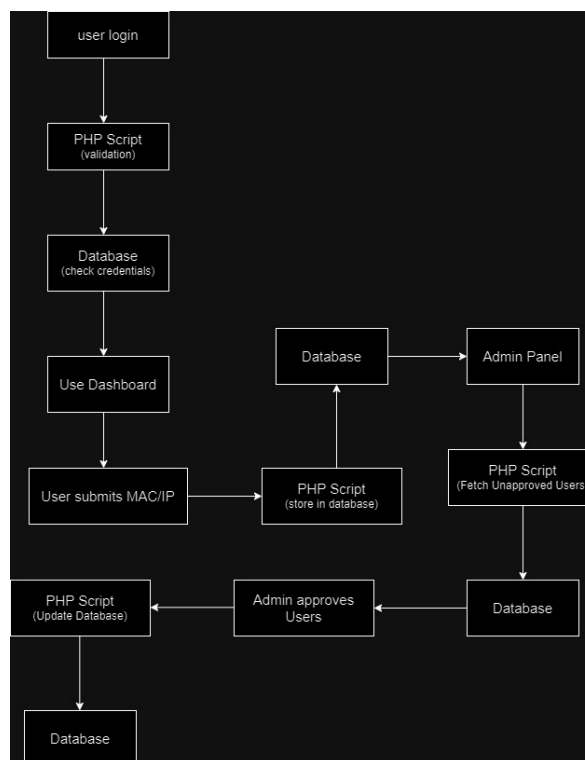
## Backend:

- PHP (Hypertext Preprocessor): A server-side scripting language which we used for web development. In this context, PHP processes requests from the frontend, interacts with the database, and generates dynamic content to be sent back to the client's browser.

## Database:

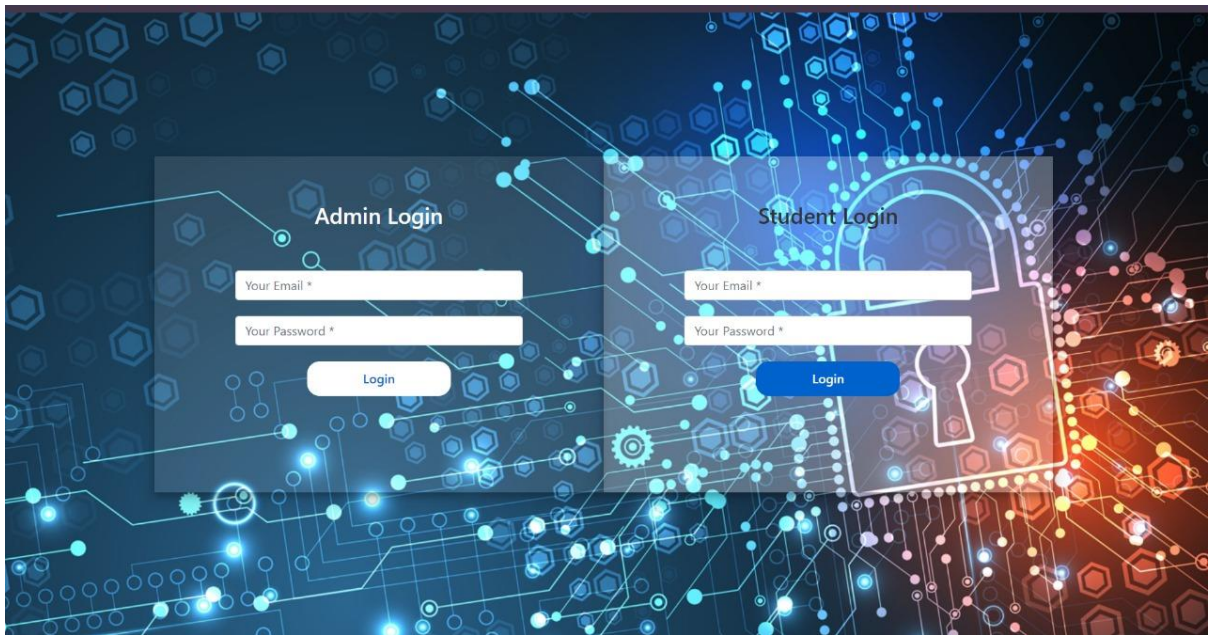
- MySQL: MySQL serves as our database management system. It houses structured data concerning students, devices, network access, issues, and solutions. SQL (Structured Query Language) facilitates seamless querying and management of our data.

## Flow Chart of the System process:

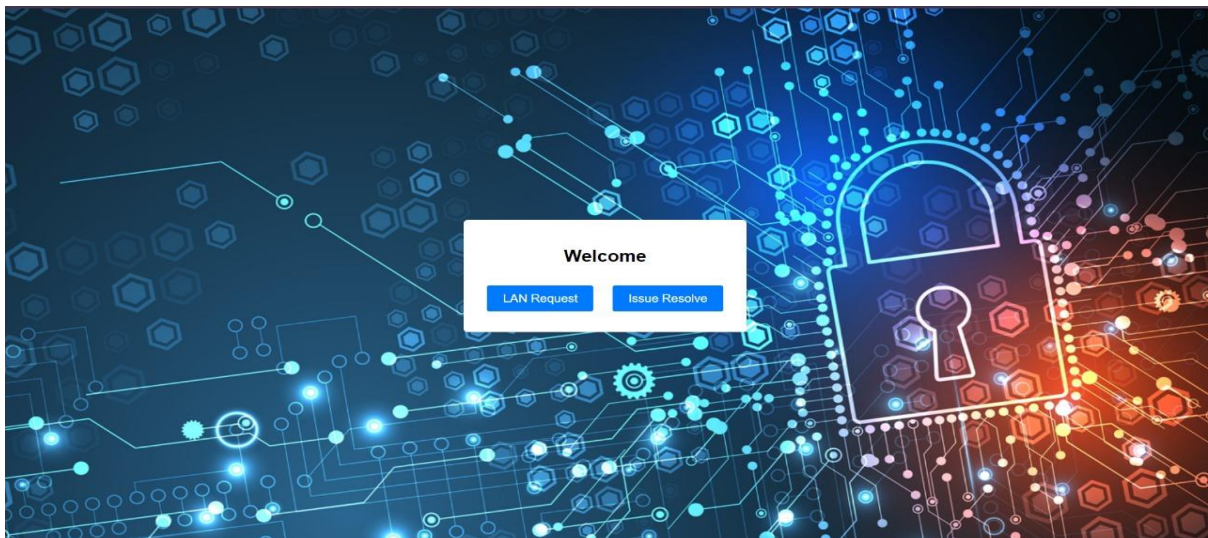


Screenshots depicting the various steps outlined in the flowchart:

Step1:

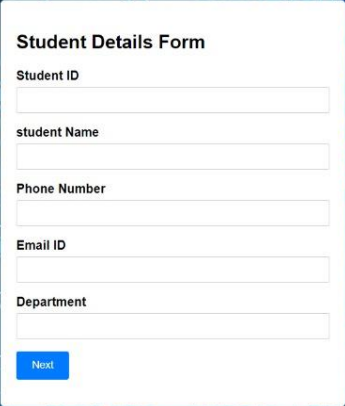


Step 2: after the login verification they will be directed to



Step 3: If they want to request lan access it will ask for their credentials





The image shows a 'Student Details Form' overlaid on a digital background. The background features a dark blue field with glowing blue and orange hexagonal patterns, circuit-like lines, and a large, stylized padlock icon on the right side. The form is a white rectangle with a blue 'Next' button at the bottom.

**Student Details Form**

Student ID

student Name

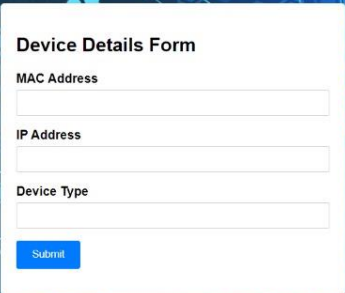
Phone Number

Email ID

Department

[Next](#)

And if entered next it will further ask their device details like mac address,ip address.



The image shows a 'Device Details Form' overlaid on the same digital background as the first form. The form is a white rectangle with a blue 'Submit' button at the bottom.

**Device Details Form**

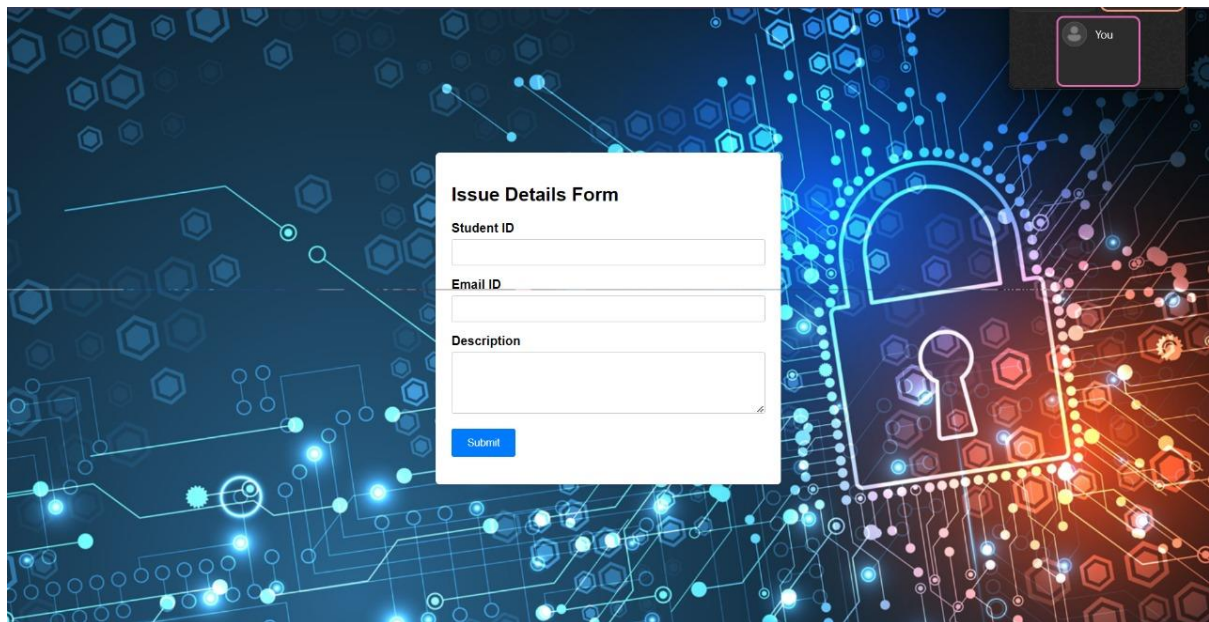
MAC Address

IP Address

Device Type

[Submit](#)

If student need to report an issue, then he will go to issue report page this will contain:



The image shows a web application interface with a dark blue background featuring a glowing circuit pattern and a large, stylized padlock icon on the right. A white modal form titled "Issue Details Form" is centered on the screen. The form contains three input fields: "Student ID", "Email ID", and "Description". Below these fields is a blue "Submit" button. In the top right corner of the application, there is a small dark box with a user profile icon and the text "You".

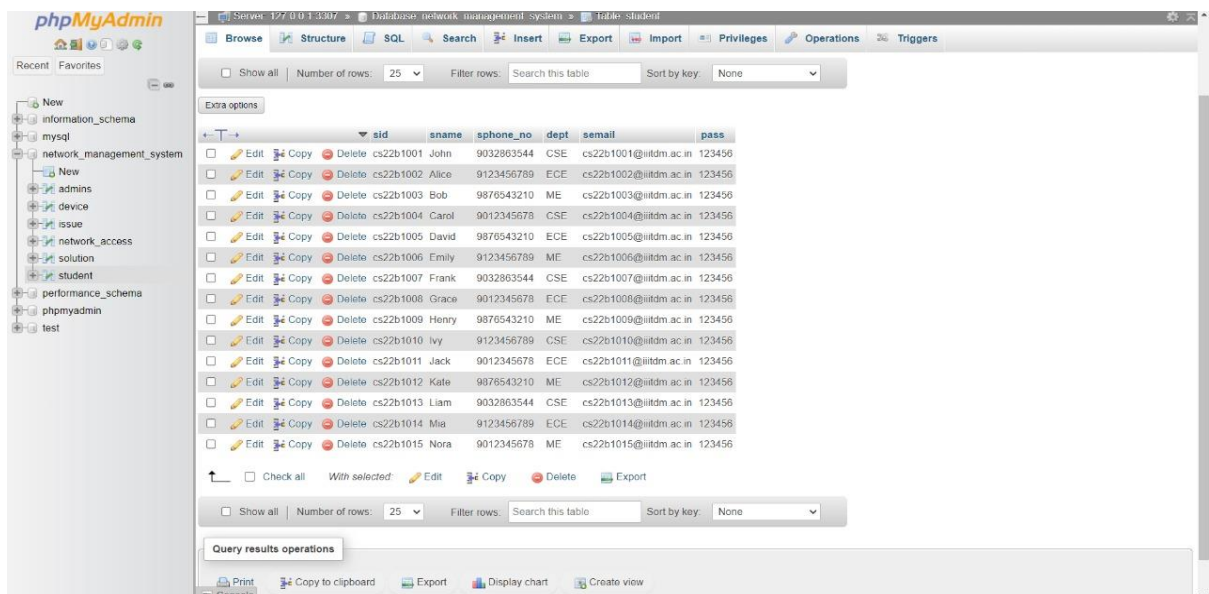
### Issue Details Form

Student ID

Email ID

Description

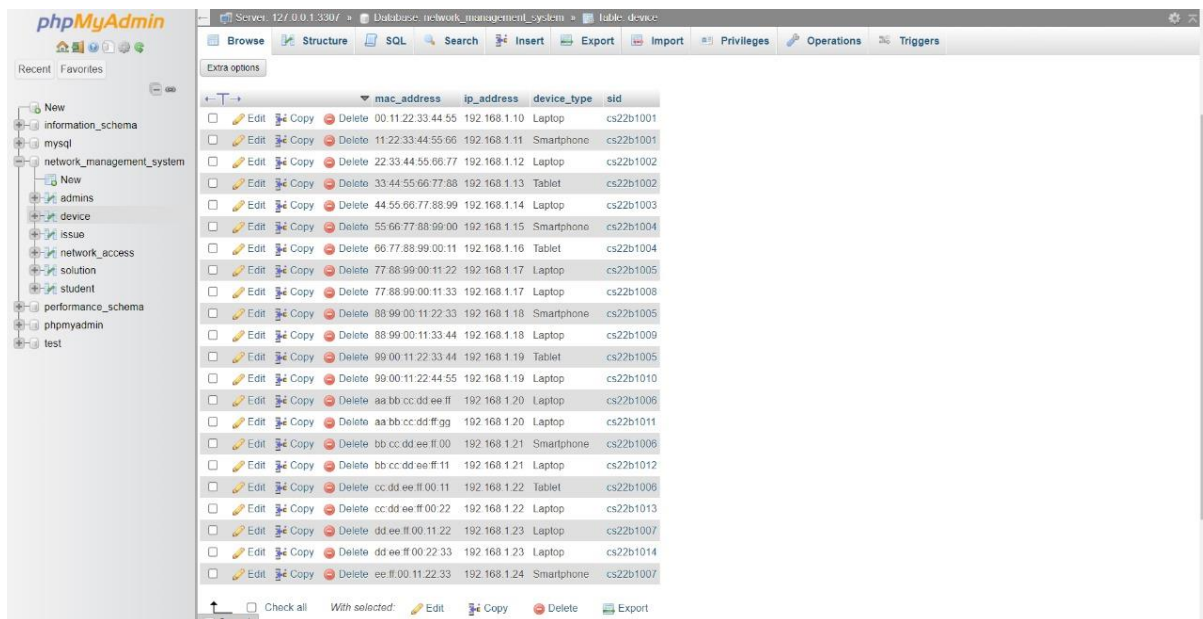
Step 4: after submitting the details are directly saved to the database (backend):



The screenshot displays the phpMyAdmin interface. On the left, the database structure is shown, with 'network\_management\_system' selected. The main area shows the 'student' table with 25 rows. The table has columns: sid, sname, sphone\_no, dept, semail, and pass. Each row includes action icons for Edit, Copy, and Delete. Below the table, there are options to 'Check all', 'With selected', 'Edit', 'Copy', 'Delete', and 'Export'. The bottom of the interface shows 'Query results operations' with buttons for 'Print', 'Copy to clipboard', 'Export', 'Display chart', and 'Create view'.

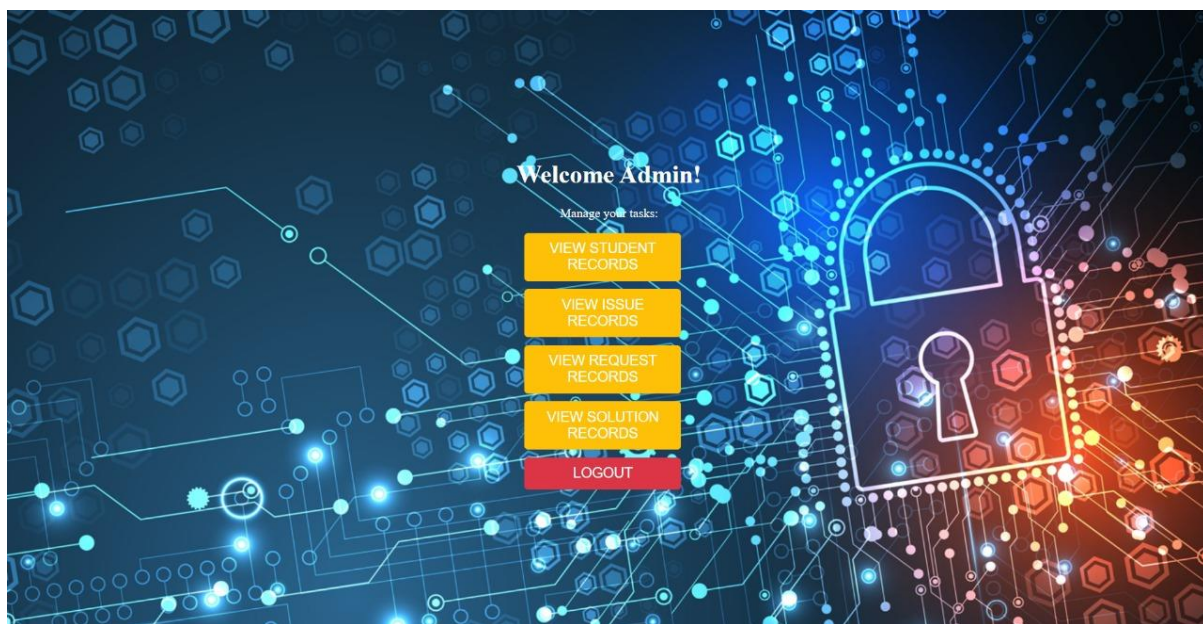
	sid	sname	sphone_no	dept	semail	pass
<input type="checkbox"/>	cs22b1001	John	9032863544	CSE	cs22b1001@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1002	Alice	9123456789	ECE	cs22b1002@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1003	Bob	9876543210	ME	cs22b1003@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1004	Carol	9012345678	CSE	cs22b1004@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1005	David	9876543210	ECE	cs22b1005@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1006	Emily	9123456789	ME	cs22b1006@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1007	Frank	9032863544	CSE	cs22b1007@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1008	Grace	9012345678	ECE	cs22b1008@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1009	Henry	9876543210	ME	cs22b1009@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1010	Ivy	9123456789	CSE	cs22b1010@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1011	Jack	9012345678	ECE	cs22b1011@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1012	Kate	9876543210	ME	cs22b1012@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1013	Liam	9032863544	CSE	cs22b1013@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1014	Mia	9123456789	ECE	cs22b1014@iitdm.ac.in	123456
<input type="checkbox"/>	cs22b1015	Nora	9012345678	ME	cs22b1015@iitdm.ac.in	123456





	mac_address	ip_address	device_type	sid
<input type="checkbox"/>	00:11:22:33:44:55	192.168.1.10	Laptop	cs22b1001
<input type="checkbox"/>	11:22:33:44:55:66	192.168.1.11	Smartphone	cs22b1001
<input type="checkbox"/>	22:33:44:55:66:77	192.168.1.12	Laptop	cs22b1002
<input type="checkbox"/>	33:44:55:66:77:88	192.168.1.13	Tablet	cs22b1002
<input type="checkbox"/>	44:55:66:77:88:99	192.168.1.14	Laptop	cs22b1003
<input type="checkbox"/>	55:66:77:88:99:00	192.168.1.15	Smartphone	cs22b1004
<input type="checkbox"/>	66:77:88:99:00:11	192.168.1.16	Tablet	cs22b1004
<input type="checkbox"/>	77:88:99:00:11:22	192.168.1.17	Laptop	cs22b1005
<input type="checkbox"/>	77:88:99:00:11:33	192.168.1.17	Laptop	cs22b1008
<input type="checkbox"/>	88:99:00:11:22:33	192.168.1.18	Smartphone	cs22b1005
<input type="checkbox"/>	88:99:00:11:33:44	192.168.1.18	Laptop	cs22b1009
<input type="checkbox"/>	99:00:11:22:33:44	192.168.1.19	Tablet	cs22b1005
<input type="checkbox"/>	99:00:11:22:44:55	192.168.1.19	Laptop	cs22b1010
<input type="checkbox"/>	aa:bb:cc:dd:ee:ff	192.168.1.20	Laptop	cs22b1008
<input type="checkbox"/>	aa:bb:cc:dd:ff:gg	192.168.1.20	Laptop	cs22b1011
<input type="checkbox"/>	bb:cc:dd:ee:ff:00	192.168.1.21	Smartphone	cs22b1008
<input type="checkbox"/>	bb:cc:dd:ee:ff:11	192.168.1.21	Laptop	cs22b1012
<input type="checkbox"/>	cc:dd:ee:ff:00:11	192.168.1.22	Tablet	cs22b1008
<input type="checkbox"/>	cc:dd:ee:ff:00:22	192.168.1.22	Laptop	cs22b1013
<input type="checkbox"/>	dd:ee:ff:00:11:22	192.168.1.23	Laptop	cs22b1007
<input type="checkbox"/>	dd:ee:ff:00:22:33	192.168.1.23	Laptop	cs22b1014
<input type="checkbox"/>	ee:ff:00:11:22:33	192.168.1.24	Smartphone	cs22b1007

Step 5: as soon as students requested lan admin can view their request and can access if they want to or can decline



As we can see from the above picture admins can also view or the issues facing by students, and they can send solutions if the solution can be solved by admins (network issues only)

Step 6: After the giving LAN access to the student it will update database for that student from not accessed to accessed.

## SOME SQL QUERIES:

Query 1: selecting student with particular student id.

The screenshot shows the phpMyAdmin interface with the 'student' table selected. The SQL query entered is `select * from student where sid='cs22b1012';`. The result shows one row for the student with sid 'cs22b1012'.

sid	sname	sphone_no	dept	semail	pass
cs22b1012	Kate	9876543210	ME	cs22b1012@iitdm.ac.in	123456

Query 2: selecting students from particular department:

The screenshot shows the phpMyAdmin interface with the 'student' table selected. The SQL query entered is `select * from student where dept='CSE';`. The result shows five rows for students in the 'CSE' department.

sid	sname	sphone_no	dept	semail	pass
cs22b1001	John	9032863544	CSE	cs22b1001@iitdm.ac.in	123456
cs22b1004	Carol	9012345678	CSE	cs22b1004@iitdm.ac.in	123456
cs22b1007	Frank	9032863544	CSE	cs22b1007@iitdm.ac.in	123456
cs22b1010	Ivy	9123456789	CSE	cs22b1010@iitdm.ac.in	123456
cs22b1013	Liam	9032863544	CSE	cs22b1013@iitdm.ac.in	123456

Query 3: selecting students where access status got denied.

phpMyAdmin interface showing the results of a query on the `network_access` table. The query is:

```
select * from network_access where access_status='denied';
```

The results show 5 rows of data, all with `access_status` set to `Denied`.

sid	mac_id	access_status	connection_type
cs22b1003	22 33 44 55 66 77	Denied	wireless
cs22b1005	44 55 66 77 88 99	Denied	wired
cs22b1007	66 77 88 99 00 11	Denied	wired
cs22b1010	99 00 11 22 33 44	Denied	wired
cs22b1013	cc dd ee ff 00 11	Denied	wired

Query 4: selecting from issues.

phpMyAdmin interface showing the results of a query on the `issue` table. The query is:

```
SELECT * FROM "issue";
```

The results show 2 rows of data.

issue_id	mac_id	issue_description
1	00 11 22 33 44 55	Network Connectivity issue
2	22 33 44 55 66 77	Slow internet speed

Query 5: selecting from students.



The screenshot shows the phpMyAdmin web interface. On the left is a sidebar with a tree view of the database structure. The main area displays the 'student' table from the 'network\_management\_system' database. The table has 15 rows of data, including columns for student ID, name, phone number, department, email, and password. The interface includes a top navigation bar with links like 'Browse', 'Structure', 'SQL', 'Search', 'Insert', 'Export', 'Import', 'Privileges', 'Operations', and 'Triggers'. A 'Show query box' button is visible above the SQL editor. The SQL editor contains a simple query: 'select \* from student;'. Below the editor, there are options for 'Profiling', 'Edit inline', 'Edit', 'Explain SQL', 'Create PHP code', and 'Refresh'. A 'Show all' button and a 'Number of rows' dropdown are also present. The table view shows the first 15 rows of the 'student' table.

			sid	sname	sphone_no	dept	semail	pass	
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1001	John	9032803544	CSE	cs22b1001@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1002	Alice	9123456789	ECE	cs22b1002@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1003	Bob	9876543210	ME	cs22b1003@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1004	Carol	9012345678	CSE	cs22b1004@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1005	David	9876543210	ECE	cs22b1005@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1006	Emily	9123456789	ME	cs22b1006@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1007	Frank	9032803544	CSE	cs22b1007@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1008	Grace	9012345678	ECE	cs22b1008@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1009	Henry	9876543210	ME	cs22b1009@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1010	Ivy	9123456789	CSE	cs22b1010@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1011	Jack	9012345678	ECE	cs22b1011@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1012	Kate	9876543210	ME	cs22b1012@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1013	Liam	9032803544	CSE	cs22b1013@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1014	Mia	9123456789	ECE	cs22b1014@iitdm.ac.in	123456
<input type="checkbox"/>	Edit	Copy	Delete	cs22b1015	Nora	9012345678	ME	cs22b1015@iitdm.ac.in	123456