Self-Intensive Training on Full Stack Development

**Student Name:** Sharmilaa Devi P A **Register Number:** 7376221EC310 **Seat No:** 371

**Project ID:** 11

**Project Title:** Bulk Mail Blocking/Unblocking

**Domain:** Students Affairs

## Problem Statement:

Currently, the process of blocking or unblocking student email IDs for mass communication is inefficient and time-consuming. This is because it likely involves manual manipulation of email lists or contacting IT support for each individual student.

This inefficiency leads to several problems:

* **Delays in communication:** Important information or announcements may be delayed in reaching students due to the time required for manual blocking/unblocking.
* **Increased workload for staff:** Staff may be burdened with repetitive tasks of managing student email access for bulk communication.
* **Potential for errors:** Manual processes are prone to human error, which could result in students being accidentally blocked or unblocked.

To address these issues, we propose the development of a secure and user- friendly portal system that allows authorized persons to efficiently block and unblock student email IDs for bulk communication purposes.

## Project Description:

Build a portal system to block/unblock bulk BIT email ID of the students.

## Project Requirements:

Here are some initial project requirements for developing a student email blocking/unblocking portal system:

# Functional Requirements:

## User Management:

* + The portal should allow for the management of user accounts with different access levels (e.g., administrator, department head).

## Student Management:

* + The system shall enable uploading of student lists in a standard format (e.g., CSV) for bulk blocking/unblocking.

## Blocking/Unblocking Functionality:

* + The portal shall provide a user-friendly interface for selecting students and applying blocking/unblocking actions.
  + The portal shall clearly indicate the current status (blocked/unblocked) of each student's email address.
  + The portal should offer the ability to schedule blocking/unblocking actions for future dates.

# Non-Functional Requirements:

## Security:

* + The system shall be secure and compliant with relevant data privacy regulations.
  + User access and student data must be protected with appropriate security measures.

## Performance:

* + The system shall be responsive and handle bulk actions efficiently without significant delays.

## Scalability:

* + The system should be scalable to accommodate a growing number of students and users.

## Usability:

* + The system interface should be user-friendly and intuitive for authorized persons with varying levels of technical expertise.

## Additional Considerations:

* + Integration with existing email infrastructure.
  + User training materials on how to utilize the portal system effectively.
  + Development of a disaster recovery plan to ensure system availability in case of outages.

This is a starting point, and the specific requirements may evolve as the project progresses and stakeholders are further consulted.

# Technical Components:

|  |  |
| --- | --- |
| **Component** | **Tech Stack** |
| Frontend | * HTML * CSS * JavaScript |
| Backend | * Linux * Apache Web Server * PHP with Laravel framework |
| Database | * MySQL |
| API | * REST Ful services |

## LAMP Stack:

The **LAMP stack (Linux, Apache, MySQL, PHP)** is a popular and powerful combination for developing web applications. Here's a breakdown of how you can build your **bulk mail blocking/unblocking portal** using LAMP:

## Setting Up the LAMP Stack:

* + **Install LAMP:** I am preferring XAMPP Control Panel, Visual Studio code.
  + **Design:** Use MySQL to create a database to store candidate information. Design tables with relevant fields like Mail ID, Password, ID etc.

## Web Development with PHP:

* + **Develop the Website:** Use PHP to build the website's functionalities, which can provide afoundation for user authentication, routing, and database interaction.

## User Interface (UI) Design:

* + **HTML & CSS:** Use HTML to structure the website's content and CSS to style the visual elements.
  + **User-Friendly Interface:** Design an intuitive interface for admin to easily navigate and manage candidate data. Consider using forms, tables, and clear labelling for data fields.

## Security Considerations:

* + **User Authentication:** Implement a secure login system using PHP.
  + **Regular Backups:** Implement a backup system for your database and website files to ensure data recovery in case of emergencies.

## Deployment:

* + **Web Server Configuration:** Configure your web server (Apache) to serve the website files and interact with the MySQL database.
  + **Testing & Deployment:** Thoroughly test the website functionalities before deploying it to a live environment. Consider user acceptance testing to ensure it meets your needs.

## Limitations of LAMP Stack:

While LAMP is a powerful choice, you might consider alternatives for specific needs:

* + **Complexity:** Building a complex application from scratch requires significant development expertise.
  + **Security:** Maintaining security requires ongoing vigilance and updates.

## Workflow:

This workflow outlines the process for authorized personnel to utilize the student email blocking/unblocking portal system:

## User Login:

* + Authorized personnel access the portal using their email and password.

## Student Selection:

* + Users can choose individual students or select groups based onpre-defined criteria (e.g., department, year).
  + Users can upload a CSV file containing a list ofstudents email addresses for bulk actions.

## Action Selection:

* + The user specifies whether to block or unblock the selected student email addresses.
  + An option to blocking/unblocking for a future date and time may be enabled.

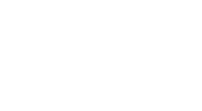
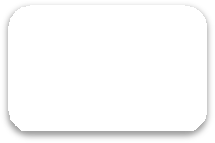
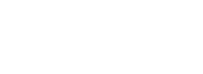
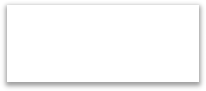
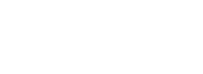
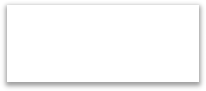
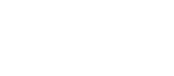
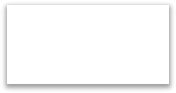
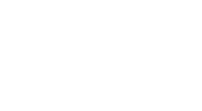
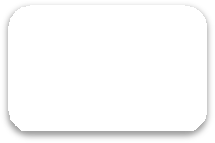
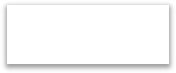
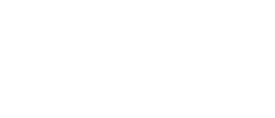
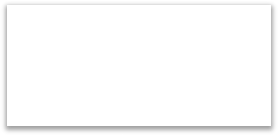
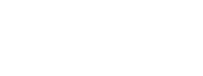
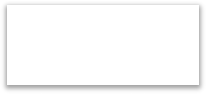
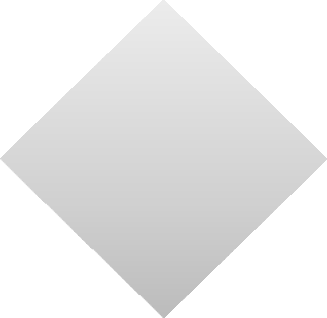
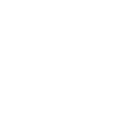
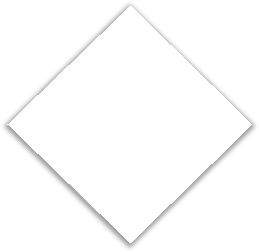
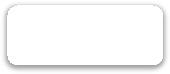
## Confirmation and Processing:

* + The system displays a confirmation screen summarizing theselected students and the chosen action (block/unblock).
  + Upon confirmation, the system executes the blocking/unblockingaction for the designated students.

## Additional Considerations:

* + The system may provide functionalities for searching and filtering students to assist with specific selections.
  + Users may be able to save frequently used student groupings forquicker selection in future tasks.
  + The system should display clear notifications and informative messages throughout the workflow to guide users and confirm actions.

# Flowchart:



**Database**

**Monitor and**

**access**

* **Block**
* **Unblock**

**Add**

**Update or**

**edit settings**

**Unblock the**

**student**

If

id==admin

**Database**

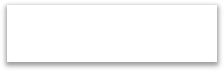
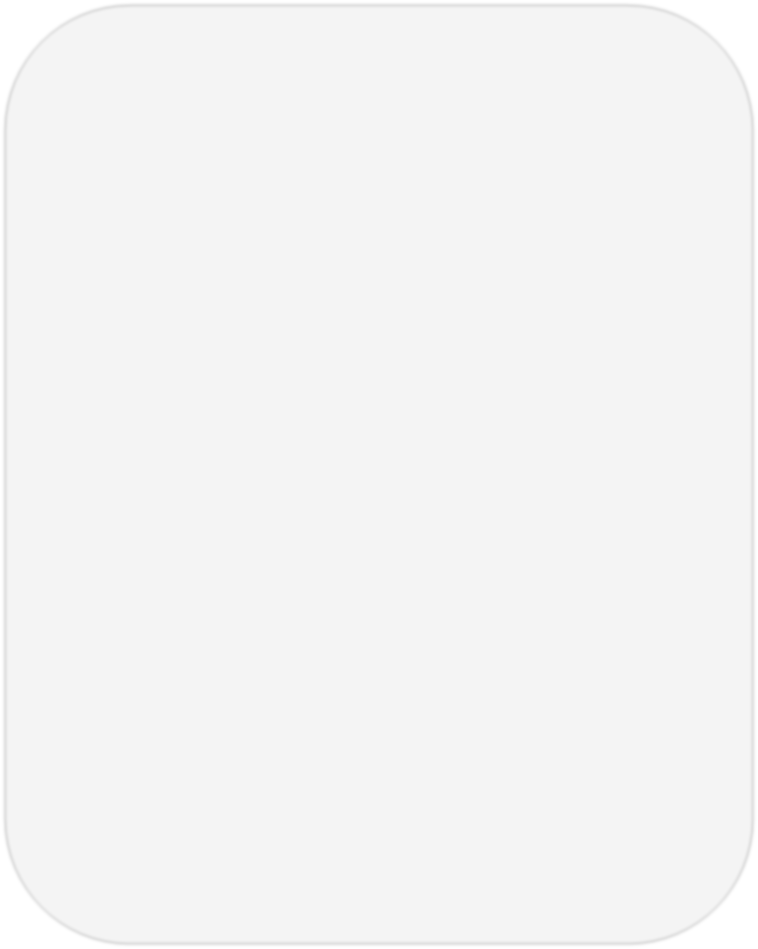
**Student status**

**Open Admin**

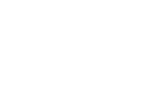
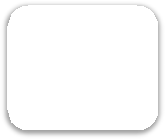
**Page**

**Start**

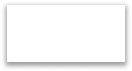
**Login**



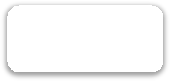
**Invalid user id**



**List of students to be blocked**

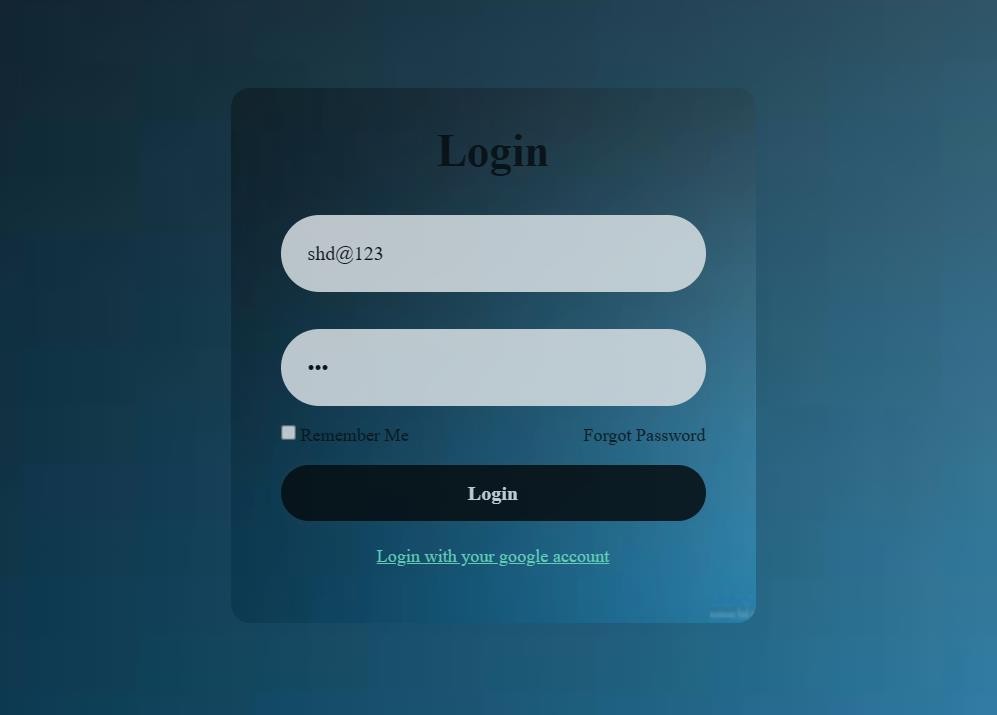


**Logout**



**Exit**

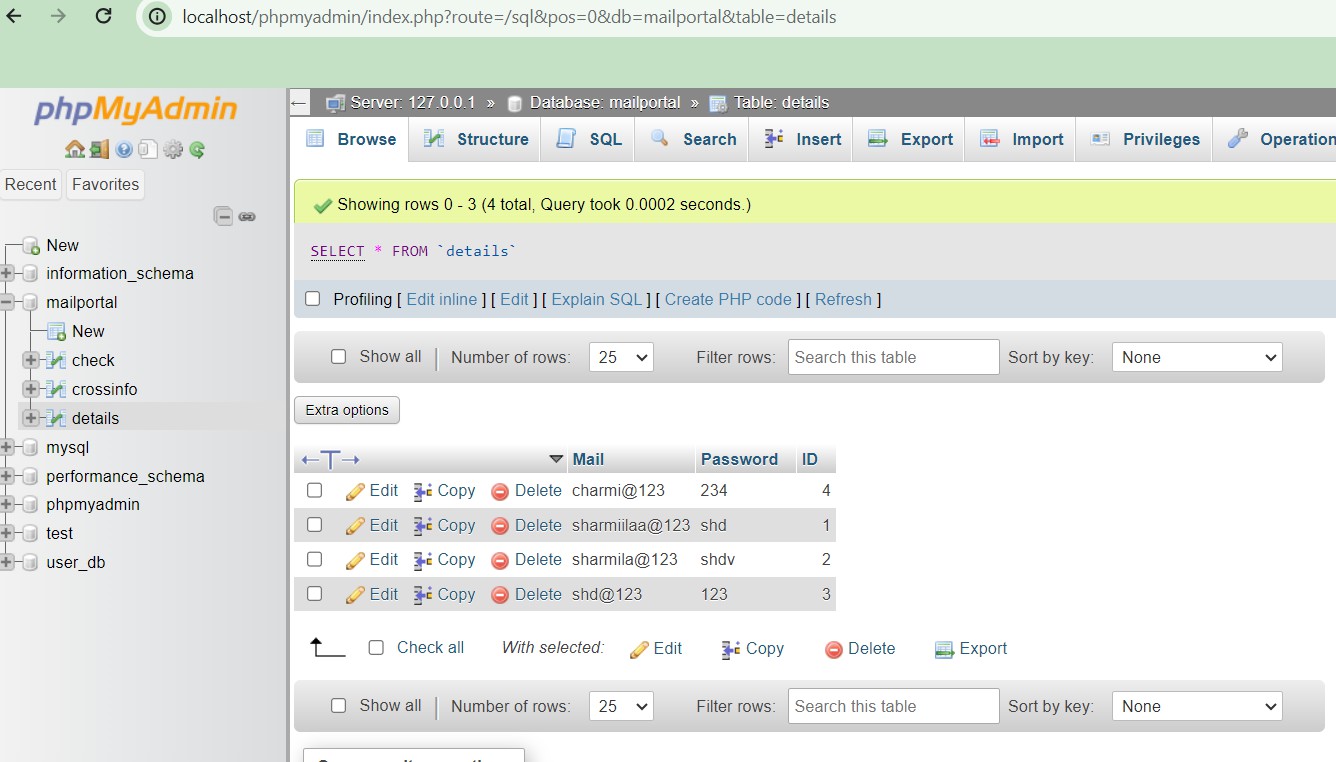
Login page:



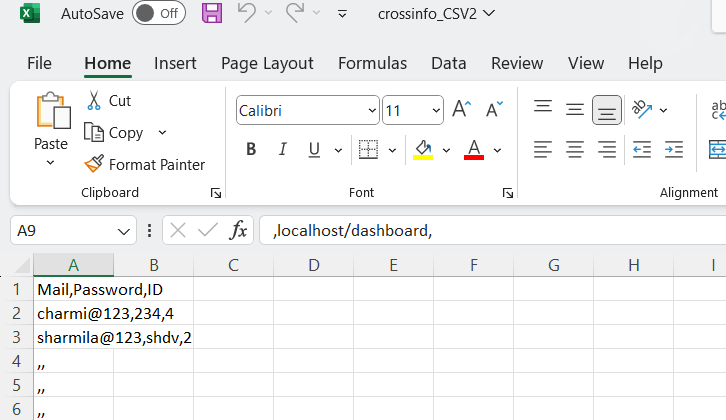
Admin’s Dashboard:



Sample list of all student ID’s:



Sample CSV file:



Sample list of ID’s to be blocked or unblocked:

