Software Requirement Specification for TAC Portal

Name	Sabarish R
Roll no	7376222IT237
Seat no	331
Project ID	11
Problem Statement	Bulk Bitsathy Mail Blocking and Unblocking

1. Introduction

1.1. Purpose:

The purpose of the project is to develop a comprehensive **portal system for managing student email IDs** within the context of a college environment. This system enables efficient administration of email accounts, allowing administrators **to perform bulk actions such as blocking or unblocking student email IDs.**

Through seamless integration with th**e Google Workspace Admin API,** the system facilitates secure and streamlined management of email accounts, enhancing administrative efficiency and providing students with a centralized platform to **monitor the status of their email IDs**. Additionally, the system ensures accountability by maintaining detailed logs of email ID actions, fostering transparency and accountability within the college community.

1.2. Scope of Project:

The scope of the project encompasses the development of a robust portal system tailored for managing student email IDs within a college environment. Key functionalities include user authentication, differentiated dashboards for admins and students, bulk action capabilities for admins **(such as blocking or unblocking email IDs),** processing of bulk actions via CSV file uploads, comprehensive logging of email ID actions, and providing students with access to their email ID action history. The project aims to enhance administrative efficiency, accountability, and transparency by providing a centralized platform for email ID management while ensuring a seamless **user experience for both administrators and students.**

2. System Overview:

2.1.Student Dashboard:

- The student dashboard provides a personalized and user-friendly interface for students to manage their email IDs within the college system.
- It offers features such as viewing the current status of their email IDs, accessing logs to track any actions performed on their email IDs, and receiving notifications about important updates or changes related to their email accounts.
- Through the student dashboard, students can conveniently monitor the status and activity of their email IDs, enhancing their overall experience and enabling them to stay informed about any changes made to their accounts.

2.2.Admin Dashboard:

- The admin dashboard serves as a powerful administrative tool for managing student email IDs efficiently and effectively. Admins have access to bulk action capabilities, allowing them to perform actions such as blocking or unblocking multiple email IDs simultaneously.
- Additionally, the admin dashboard provides comprehensive logging functionality, enabling admins to track and audit all actions performed on email IDs within the system.
- With features for CSV file uploads for bulk processing, the admin dashboard streamlines administrative tasks, enhances accountability, and ensures seamless management of student email IDs across the college environment.

2.2. Features:

1.User Authentication: Secure login functionality for both students and administrators to access their respective dashboards.

2.Differentiated Dashboards: Separate dashboards tailored for students and administrators, providing customized views and functionalities based on user roles.

3.Bulk Action Capabilities: Admins can perform bulk actions such as blocking or unblocking multiple student email IDs at once for efficient management.

4.CSV File Upload: Support for uploading CSV files containing lists of email IDs to perform bulk actions, simplifying administrative tasks.

5.Comprehensive Logging: Detailed logs of email ID actions, including blocking, unblocking, and any other relevant activities, to track changes and ensure accountability.

6.Student Email ID History: Students can access logs to view the history of actions performed on their email IDs, promoting transparency and awareness.

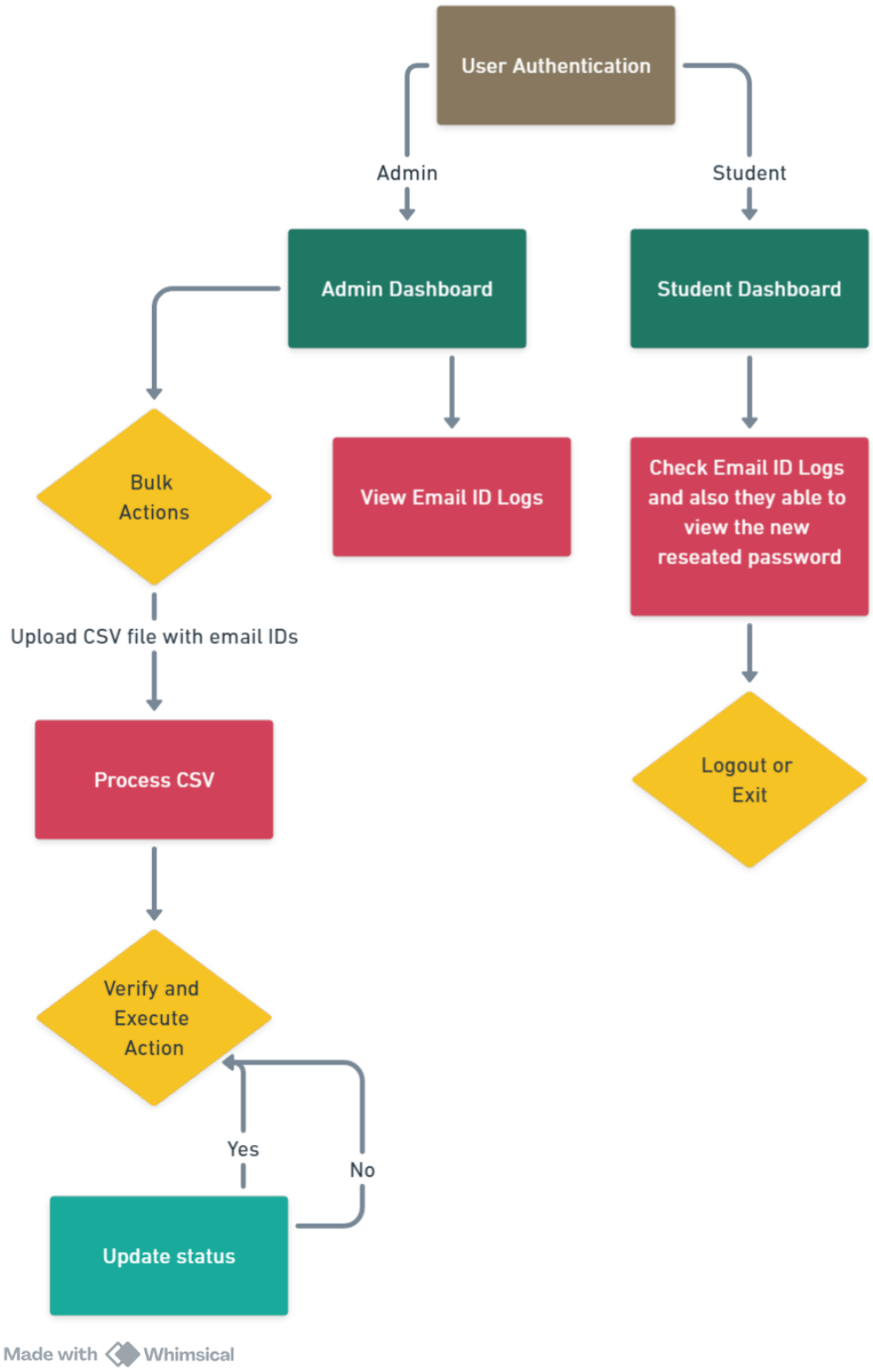
7.Notifications: Automated notifications to inform students and administrators about important updates, changes, or events related to email ID management.

8.Secure Access: Implementation of secure authentication protocols and data encryption to protect user information and maintain system integrity.

9.User-Friendly Interface: Intuitive and easy-to-use interface for seamless navigation and enhanced user experience for both students and administrators.

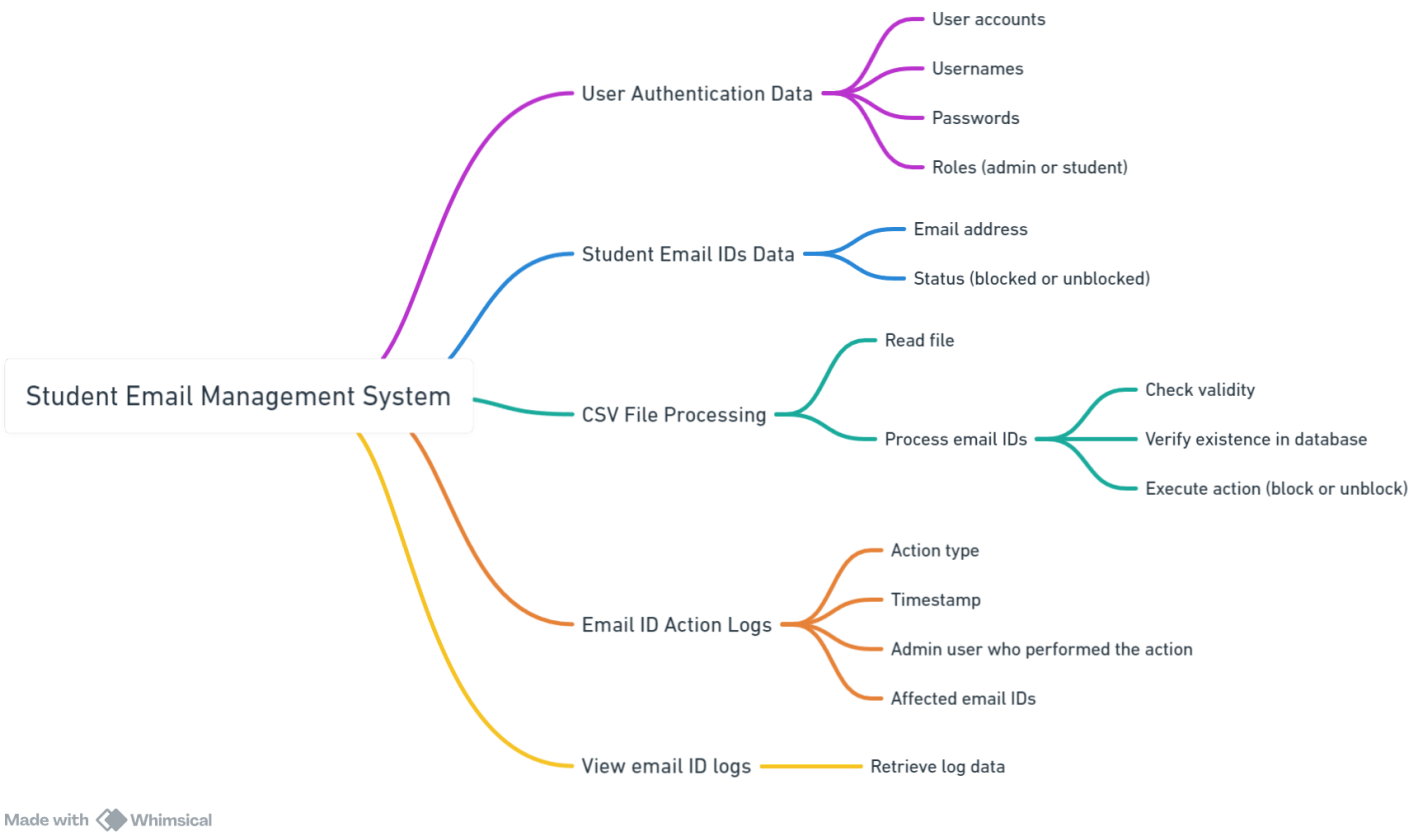
10.Scalability: Ability to scale the system to accommodate the growing needs of the college environment and handle a large volume of email IDs effectively.

11.Customization: Flexible configuration options to customize the system according to the specific requirements and preferences of the college administration.


User Authentication
Admin
Student
Admin Dashboard
Student Dashboard
Bulk Actions
View Email ID Logs
Check Email ID Logs and also they able to view the new reseated password
Upload CSV file with email IDs
Process CSV
Logout or Exit
Verify and Execute Action
Yes
No
Update status
Made with Whimsical

Stack:

Front End	React,Tailwind css
Backend	Node Js, Express
Data Base	MongoDB



1.Google Workspace Integration: Integration with the Google Workspace Admin API to perform actions such as blocking or unblocking email IDs, retrieving user information, and managing user permissions. This integration ensures synchronization between the email ID management system and Google Workspace.

2.LDAP Integration: Integration with Lightweight Directory Access Protocol (LDAP) services for user authentication and directory services. LDAP integration allows the system to authenticate users against the college's directory server and retrieve user attributes for access control.

3.Single Sign-On (SSO): Implementation of Single Sign-On functionality to enable users to log in once and access multiple systems or applications without re-authenticating. This enhances user convenience and security by centralizing authentication and access control.

4.Data Migration: Migration of existing email ID data from legacy systems or databases to the new email ID management system. This involves extracting data, transforming it into the required format, and loading it into the new system while ensuring data integrity and consistency.

5.Third-Party Service Integration: Integration with third-party services or tools for additional functionality such as analytics, reporting, or communication. This may include integrating with email delivery services for notifications or analytics platforms for monitoring system performance.

Users interact with the frontend UI and are presented with a login form to enter their credentials

Frontend application sends a request to the backend server for user authentication

Frontend may perform basic validation on user input before sending the request

Backend server verifies provided credentials against user data stored in the database (e.g., MongoDB

Upon valid credentials, backend generates a JSON Web Token (JWT) using a secret key from environment variables

JWT typically contains information about the user and is signed by the server, then sent back to frontend

Frontend securely stores JWT, usually in browser storage (e.g., localStorage or sessionStorage

JWT is included in subsequent requests to backend, typically in Authorization header of HTTP requests

Backend verifies JWT in each request for user authentication and access control purposes

Backend extracts user information from JWT payload to determine role and permissions for access control decisions