Software Requirement Specification for Student Database Management

NAME	BARANIKAA SRI R
ROLL NO	7376222CB107
SEAT NO	25
PROJECT ID	25
PROJECT STATEMENT	Student Database Management System

STACK :

Front End	React js
Backend	Node Js , Express Js
Database	MangoDB

PROJECT FLOW:

Purpose :

The purpose of this project is to create an user-friendly Student Database Management System. The respective admin (staff) member can access the website for retrieving and analyzing the details of students . When there is a need to specifically search students based on gender , department , community , seat category etc, their details are displayed by filtering . Count of each attribute has to be displayed in the website .

Scope of Project :

This software system can serve as the real time dashboard for viewing the details of students and visualizing it graphically . Sending email effectively to the specific set of students . Implementing authentication mechanisms to ensure safe access to the system. Design and implementation of a relational database schema to store student details such as gender, department, community, seat category, and many other attributes .

Objectives :

Centralize Student Information : Consolidate all student data into a single, centralized database for easy access and management.
Enhance Search and Retrieval : Enable administrative staff to quickly search and retrieve student records based on various criteria such as gender, department, community, seat category, etc.
Facilitate Data Analysis: Generate reports and analytics on student demographics, enrollment trends, and other relevant metrics to aid decision-making and planning.
Increase Security and Compliance: Implement security measures to protect sensitive student data and ensure compliance with data privacy regulations.

Dependencies:

Availability of existing student data, and any necessary processes or tools for migrating data to the new database system.
Implementation of appropriate security measures, such as encryption, access controls, and auditing, to protect sensitive student data from unauthorized access .

Use case:

- As a faculty member , I have to know the details of students in an efficient manner .
- Admin updates an existing student record with revised information, such as changes in contact details, and academic status.

Accessibility Considerations :

Accessible Forms: Design forms with clear labels and instructions, and ensure that form fields are programmatically associated with their labels. Use appropriate input types and attributes to enhance accessibility.

Error Handling: Provide clear and descriptive error messages when users encounter input errors or validation issues. Ensure that error messages are presented in a way that is accessible to all users, including those using screen readers.

Functional Requirements :

Reporting and Analytics: Generation of reports summarizing student demographics, enrollment statistics, or other relevant metrics.Visualization tools (e.g., charts, graphs) to present data analytics in a visually appealing manner.

Data Export: Ability to export student data from the system in various formats (e.g., CSV, Excel) for further analysis or reporting purposes

User Management: Administration of user accounts, including creation, modification, and deletion of user accounts.

Resetting passwords and managing user permissions.

User Authentication: The system should require users to log in with valid credentials (username/password) before accessing any functionality.

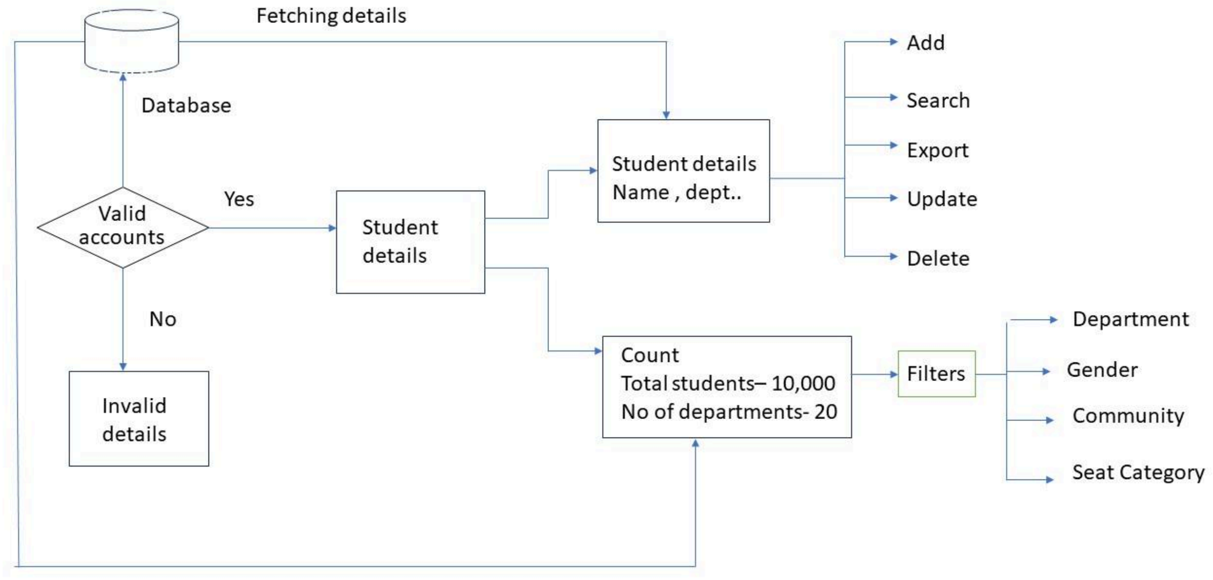
Non functional Requirements:

Performance : Response times for common operations (e.g., search, data retrieval) should be within acceptable limits, even under peak loads.

Scalability: The system should be able to scale horizontally or vertically to accommodate increases in data volume and user load over time.

Reliability: The system should be highly reliable, with minimal downtime and errors.Data integrity should be maintained, and data loss should be prevented through appropriate backup and recovery mechanisms.

Maintainability: The system should be designed with clean, modular code and well-documented architecture to facilitate ease of maintenance and future enhancements.Code changes and updates should be manageable without disrupting system functionality.



BACKEND :

Student entity

Name	String
Email id	string - Primary Entity
Gender	string
Age	int
Community	string
Address	string
Phone Number	int
Aadhar Number	int
Student status	string

Register Number	Int - Primary Entity
Father Name	string
Mother Name	string
Occupation	string
Blood Group	string

Academic details :

Year	int
Department	string
Student Category	string
Seat Category	string
Student Status	string
Semester	int
Year of completion	int
University	string
School marks	int
Regulation	int
Degree Level	string
Quota	string