BANNARI AMMAN INSTITUTE OF TECHNOLOGY
An Autonomous Institution, Affiliated to Anna University,
Approved by AICTE, Accredited by NAAC with 'A+' Grade
Stay Ahead

Student Name: MAGATHI M

Project ID: 36

Project title: Student Faculty ratio mapping for NBA, Anna university

Technical Components

COMPONENT	TECH STACK
FRONT END	Vue. js
BACK END	Node.js , Express.js
DATABASE	MongoDB
API	Open API

MongoDB: As the database management system, MongoDB provides a flexible and scalable solution for storing and retrieving user data, such as profiles, posts, connections, and job listings. Its document-oriented structure allows for efficient storage and querying of complex data structures.

Express.js: This web application framework for Node.js serves as the back end server, handling HTTP requests and responses. Express.js is used to build RESTful APIs for various functionalities like user authentication, CRUD operations on data, and integrating with third-party services (e.g., Google Auth).

Vue.js: As the front-end library,Vue.js is responsible for building the user interface components and handling the interactive elements of the portal. It enables the creation of reusable UI components for features like profile editing, feed display, messaging, and job application forms.

Node.js: This JavaScript runtime environment powers the back end server and provides an efficient event-driven architecture for handling concurrent connections. Node.js is used to run the Express.js server and handle real-time operations like websocket communication for instant messaging.

Implementation Timeline

Phase	Deadline	Status	Notes
Stage 1	01-08-2024	Approved	Planning and Requirement Gathering
Stage 2	-	In progress	Design and Prototyping
Stage 3	-		DataBase Designing
Stage 4	-		Back end implementation
Stage 5	-		Testing & implementation

Phase Notes

- Stage 1 - Planning and Requirement gathering
- Stage 2 - Design and Prototyping
- Stage 3 - DB Designing
- Stage 4 - Back end Implementation
- Stage 5 - Testing & Implementation

PROBLEM STATEMENT:

The ongoing system of managing and monitoring student-faculty ratios at our institution is inefficient and prone to errors due to its manual nature. Faculty and administrative staff face significant challenges, including inaccurate data entry, lack of real-time visibility into current ratios, and cumbersome administrative tasks. This results in imprecise ratio calculations, delays in reporting, and overall inefficiencies in meeting Anna University visit and NBA accreditation standards.

PROJECT-FLOW:

Purpose:

The purpose of this project is to develop an automated system to efficiently manage and monitor the student-faculty ratio to Anna University visit and NBA accreditation purposes. The system aims to streamline data collection, reduce manual errors, and provide accurate and real-time insights into the student-faculty ratio.

Scope:

The scope of this project includes the development of a web application using the MEVN stack (MongoDB, Express.js, Vue.js, Node.js). Key features will include user authentication, real-time data visualization, ratio calculation, automated notifications, and administrative reporting tools.

Business Context:

Maintaining an optimal student-faculty ratio is crucial for academic excellence and compliance with NBA accreditation standards. The current manual process is error-prone and time-consuming, affecting efficiency and accuracy. An automated system will enhance organization and productivity

Consideration:

- All faculty possess active Google accounts for authentication.
- Users have regular access to internet-enabled devices.

Dependencies:

Success depends on stakeholder input, technical resources, and front-end back-end integration. Coordination with the IT department for deployment and maintenance is essential.

User personas:

- Faculty: Requires access to update and manage their student lists and track the student-faculty ratio.
- Admin Staff: Manages faculty data, resolves issues related to faculty assignments, and approves updates to the system.

User Stories:

- As a faculty member, I can view the current student-faculty ratio for my department.
- As an admin staff, I can log in using my Google account to update my student list.
- As an admin staff, I can approve or reject faculty updates to ensure data accuracy.
- As an admin staff, I can generate reports on student-faculty ratios for accreditation purposes.

Functional Requirements:

1. **User Authentication:**
   - Faculty members and administrative staff must be able to log in using their Google accounts.
   - Role-based access control to differentiate between faculty and admin functionalities.

2. **Data Management:**
   - Provide a real-time view of the current student-faculty ratio.
   - Display detailed information about each faculty member, including their department and assigned students.
   - Allow faculty members to update their student lists online.
   - Enable users to modify or correct their data.
   - Prevent data conflicts by ensuring that updates are approved by admin staff.

3. **Ratio Calculation:**
   - Automatically calculate the student-faculty ratio based on updated data.
   - Provide a dashboard view to display the ratio and highlight any discrepancies.

4. **Administrative Features:**
   - Admin users must be able to add, update, or remove faculty and student data.
   - Generate reports on student-faculty ratios, trends, and other relevant metrics.
   - Manage user accounts and permissions.

5. **User Dashboard:**
   - Provide a personalized dashboard for users to view their student lists and ratio status.

6. **Data Security:**
   - Ensure secure storage and transmission of user data.
   - Implement measures to protect against unauthorized access and data breaches.

7. **Mobile Responsiveness:**
   - Ensure the system is fully responsive and usable on mobile devices.

8. Algorithm

- Dynamic Faculty Reassignment Algorithm.

FLOW CHART: