VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI–590 018

A Mini Project Report on

"COLLEGE ERP"

Submitted in partial fulfillment of the requirements for the 5th Semester of Bachelor of Engineering in Electronics & Communication Engineering

Submitted by:

Mr. Abhishek.A.A	**2AV23EC001**
Mr. Ashifali M. N.	**2AV23EC008**
Mr. Kiran Y.K	**2AV23EC031**
Mr. Prashant P. J.	**2AV23EC053**

Under the Guidance of

Prof. Asif Iqbal Mulla B.E, M. Tech

Assistant Professor Dept. of ECE
AGMRCET, Varur

Department of Electronics and Communication Engineering

A.G.M Rural College of Engineering & Technology, Varur-581 207

2025-2026

VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAVI

A.G.M Rural College Engineering & Technology Varur Hubbali-581207 Dist- Dharwad

APPROVED BY AICTE, NEW DELHI. Accredited by NAAC with "B++" Grade

Department of Electronics and Communication Engineering

Certificate

This is to certify that **Mr. Abhishek A.A. , Mr. Ashifali.M.N., Mr. Kiran.Y.K and Mr. Pr ashant P.J**. bearing USN **2AV23EC001, 2AV23EC008, 2AV23EC031, 2AV23EC053** respectively have satisfactorily completed the Project Work entitled "COLLEGE ERP" in partial fulfillment for the 5th semester of Bachelor of Engineering in Electronics and Communication Engineering of Visvesvaraya Technological University Belagavi, during the year 2025-26. The project report has been approved, as it satisfies the academic requirements in respect of the project work prescribed for the said degree

.

…………………………	…………………………	…………………………
Project Guide	**HOD**	**Principal/Director**
Prof. Asif Iqbal Mulla	**Dr.Sumana Achar**	**Dr.Sandeep Kyatanavar**

External Viva

Name of the Examiners: **Signature with Date:**

1.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

We would like to take this opportunity thank our Project Guide **Prof,Asif Iqbal** Assistant Professor Department of Electronics and Communication Engineering, without his immense guidance and support the work would have been unthinkable. Also, we would like to thank our Project Coordinator **, Prof, Asif Iqbal Mulla** Assistant Professor, Department of Electronics and Communication Engineering, who helped us in the completion of project work.

We deeply thank our HOD **Dr.Sumana Achar** Department of Electronics and Communication Engineering, for her unstinted support. We extend our gratitude to the Principal **Dr. Sandeep Kyatanavar, AGMRCET, Varur** for his generous support in all regards.

We extend our heartfelt thanks to all the faculty members, Teaching and Non-Teaching Staff of the Department of Electronics and Communication Engineering, **AGMRCET, Varur** who have helped us directly or indirectly. We are very much indebted to our **parents** and **friends** for their unquestioning best cooperation and support.

Mr. Abhishek A.A	**(2AV23EC001)**
Mr. ashifali.M.N	**(2AV23EC008)**
Mr. Kiran.Y.k	**(2AV23EC031)**
Mr. Prashant.P.J	**(2AV23EC053)**

ABSTRACT

The **College ERP System** is a web-based management platform designed to digitalize and streamline academic and administrative processes of **AGMR Engineering Collage, Varur**. Traditionally, activities such as attendance marking, timetable circulation, notes distribution, and announcements are performed manually, leading to inefficiencies and delays. The proposed ERP system integrates these functions into a centralized portal, enabling both students and faculty to access and manage information efficiently.

The system includes dedicated **Teacher and Student Dashboards**. Teachers can upload study materials, mark daily attendance, and publish academic announcements. Students can view their attendance records, download notes, check timetable schedules, view announcements, and calculate their semester **CGPA** through an integrated calculator. The backend is developed using **Python Flask**, while the frontend uses **HTML, CSS**, and JavaScript, ensuring a responsive and user-friendly interface. Data is stored securely using **MongoDB**, providing scalability and reliability.

This system enhances transparency, reduces paperwork, improves communication, and saves time. By automating routine academic operations, the College ERP System contributes to a more efficient and digital learning environment.

CONTENTS

Chapter 1

INTRODUCTION

In the evolving landscape of higher education, the integration of digital technologies into institutional workflows has become not just a convenience but a necessity. Colleges and universities are expected to deliver academic services with the same precision, speed, and transparency as modern enterprises. Yet, many institutions continue to rely on fragmented, manual systems for managing core operations such as attendance tracking, academic scheduling, resource sharing, and communication. These legacy practices often result in inefficiencies, data inconsistencies, and limited accessibility, ultimately affecting the quality of education and administrative responsiveness. The need for a robust, centralized solution is evident—one that can unify academic and administrative processes under a single digital framework. This project proposes the development of a web-based College ERP System tailored for AGMR Engineering College, designed to automate key workflows, enhance data integrity, and foster seamless interaction between students, faculty, and administrators. By leveraging modern web technologies such as Flask and MongoDB, the system aims to deliver a scalable, secure, and user-friendly platform that aligns with the digital transformation goals of contemporary educational institutions.

1.1 Motivation

The motivation behind this project stems from the observable gap between technological capabilities and current academic practices. Manual systems are prone to errors, delays, and data loss. Faculty members spend excessive time on repetitive tasks like attendance marking and note distribution, while students struggle to access academic resources in a timely manner. This project seeks to eliminate these inefficiencies by introducing a centralized ERP system that automates routine operations, ensures real-time data access, and improves overall institutional productivity.

1.2 Existing Model

Most colleges still operate using traditional methods that lack integration and digital accessibility.

Common practices include:

- **Manual attendance registers**: Prone to errors and difficult to audit.
- **Timetables on notice boards**: Static and inaccessible remotely.
- **Notes shared via WhatsApp or photocopies**: Unorganized and difficult to track.
- **Announcements made verbally or printed**: Often missed or delayed.

Drawbacks of the existing model:

- No real-time access to academic data.
- High risk of data misplacement or manipulation.
- Absence of a centralized academic repository.
- Poor user experience and limited scalability.

1.3 Proposed Model

The proposed solution is a **modular, web-based ERP system** that addresses the limitations of the current model.

Key features include:

- **Role-based dashboards**: Separate interfaces for students and teachers.
- **Automated attendance**: One-click marking with date and time stamps.
- **Notes and announcements upload**: Teachers can share resources instantly.
- **Student access to academic data**: Attendance, notes, timetables, and CGPA.
- **Technology stack**: Flask for backend logic, MongoDB for scalable data storage.

This model ensures centralized control, real-time updates, and a responsive user experience.

1.4 Problem Statement

“To design and implement a secure, scalable, and user-friendly web-based College ERP System for AGMR Engineering College that automates academic and administrative workflows including attendance tracking, notes sharing, timetable access, announcements, and CGPA computation.”

This statement encapsulates the core aim of the project: to replace manual systems with a digital platform that improves efficiency, accuracy, and accessibility.

1.5 Objectives

The system is designed to fulfill the following objectives:

- Automate academic and administrative operations to reduce manual workload.
- Provide distinct dashboards for teachers and students to streamline role-specific tasks.
- Enable efficient and accurate attendance management with real-time logging.
- Offer online access to notes, announcements, and academic schedules.
- Minimize paperwork and eliminate redundant record-keeping.
- Ensure secure and centralized storage of institutional data using MongoDB.

1.6 Challenges

Developing a comprehensive ERP system involves several technical and design challenges:

- **UI Design**: Creating a responsive and intuitive interface for users with varying technical proficiency.
- **Data Integrity**: Maintaining consistency and accuracy in MongoDB's document-based schema.
- **Authentication & Access Control**: Implementing secure login and role-based permissions.
- **Concurrency Handling**: Managing simultaneous data access and updates without conflicts.
- **Scalability**: Designing the system to accommodate future modules like exams, fees, and mobile apps.

Each challenge was addressed through careful architectural planning and technology selection.

1.7 Scope

Current Functional Scope:

- Attendance tracking with automated timestamps.
- Notes sharing via teacher uploads.
- Timetable and academic calendar access.

- Announcement broadcasting.
- CGPA calculator for student performance tracking.
- Future Enhancements:
- Online fee payment integration.
- Hostel and library management modules.
- Online examination system with result generation.
- Mobile app for Android/iOS platforms.
- AI-powered chatbot for student queries and support.

The system is built with modularity in mind, allowing seamless integration of future features.

Chapter 2

LITERATURE SURVEY

Table 2.1:-

Author(s)	Title	Findings
Singh & Sharma (2021)	ERP in Higher Education	ERP systems enhance transparency, resource utilization, and decision-making in academic institutions
Reddy et al. (2020)	Attendance Automation System	Digital attendance systems improve accuracy, reduce manual errors, and save faculty time
Patel & Desai (2019)	Notes Sharing System	Web-based platforms reduce dependency on physical distribution and improve accessibility

The literature survey explores existing research and technological advancements in the domain of college ERP systems, attendance automation, and digital academic workflows. These studies provide a foundation for understanding the challenges, benefits, and design considerations for implementing a robust ERP system in educational institutions.

2.1 ERP Systems in Higher Education:-

Singh & Sharma (2021) in their IEEE paper titled *“A Mobile Computing Based Attendance System and Students' Attitude Study”* emphasized the importance of digitizing attendance systems using mobile computing. Their study concluded that mobile-based ERP systems are not only cost-effective but also improve user satisfaction and reduce administrative burden. The system they proposed was found to be cheating-proof, faster, and more user-friendly compared to traditional methods.

Shreyas H. B. et al. (2023) conducted a study titled *“A Study on the Effectiveness of College ERP System in Higher Education Institutions”* published in IRJET. The research highlighted that ERP systems significantly enhance institutional productivity, data transparency, and decision-making. The study also noted that ERP adoption leads to better resource planning, academic performance tracking, and communication between stakeholders.

2.2 Attendance Automation

Tanuj Tongse et al. (2025) proposed a *“Hybrid Approach for Attendance ERP Using Biometric and QR Code Verification”* in IJIRMPS. Their system combined biometric and QR code technologies to address issues like proxy attendance, manual errors, and time inefficiencies. The hybrid model ensured secure, real- time attendance tracking and was scalable for large institutions.

Additionally, the IEEE study by Singh & Sharma also found that automated attendance systems positively influence student discipline and academic performance, as they provide real-time feedback and reduce absenteeism.

2.3 Notes Sharing and Academic Resource Distribution

Patel & Desai (2019) in their IJCRT paper on *“Notes Sharing System”* discussed the limitations of physical distribution of study materials. They proposed a web-based platform that allows teachers to upload and students to download notes, ensuring easy access, version control, and reduced dependency on physical media. Their findings support the inclusion of a digital notes module in ERP systems to enhance learning outcomes.

2.4 Key Takeaways from Literature

- ERP systems improve institutional efficiency, transparency, and academic management.

- Mobile and web-based attendance systems are more accurate, secure, and user-friendly than manual methods.
- Hybrid attendance models (biometric + QR code) offer enhanced security and scalability.
- Digital notes sharing reduces logistical challenges and ensures timely access to academic resources. These studies collectively validate the need for a centralized, modular, and scalable ERP system in educational institutions. They also guide the design of the proposed system by highlighting best practices and common pitfalls.

Chapter 3

DOMAIN ANALYSIS

The development of a College ERP System requires a deep understanding of the technological domains that enable automation, scalability, and secure data management. This chapter focuses on two key domains: the **Internet of Things (IoT)** and **Cloud Computing**. These technologies not only support the current system architecture but also open avenues for future enhancements like biometric attendance and institution-wide deployment.

3.1 The Internet of Things (IoT)

Definition:

The Internet of Things refers to a network of interconnected physical devices embedded with sensors, software, and connectivity that enable them to collect and exchange data.

Relevance to ERP Systems:

- **Biometric Attendance**: IoT devices such as fingerprint scanners or facial recognition cameras can be integrated into the ERP system to automate attendance tracking. This eliminates proxy attendance and enhances security.
- **Smart Classrooms**: IoT-enabled projectors, smart boards, and environmental sensors can be managed through the ERP for scheduling and monitoring.
- **Real-time Monitoring**: Devices can track classroom occupancy, energy usage, and student movement, feeding data into the ERP for analytics and optimization.

Example Use Case:

- A fingerprint scanner at the classroom entrance records student attendance and sends the data to the ERP in real-time. Students can view their attendance percentage instantly on their dashboard.

Benefits:

- Enhanced accuracy and security
- Reduced manual intervention
- Real-time data collection and analytics

3.2 Cloud computing

Definition:

Cloud computing provides on-demand access to computing resources—servers, storage, databases, networking, software—over the internet with pay-as-you-go pricing.

Relevance to ERP Systems:

- **Scalability**: Cloud platforms like AWS, Azure, or Google Cloud allow the ERP system to scale as the number of users or data volume increases.
- **Accessibility**: Students and teachers can access the ERP system from any location or device with internet connectivity.
- **Data Redundancy and Backup**: Cloud services offer automated backups and disaster recovery, ensuring data safety.
- **Cost Efficiency**: Reduces the need for on-premise infrastructure and maintenance.

MongoDB in the Cloud:

- MongoDB Atlas, the cloud-hosted version of MongoDB, supports horizontal scaling and global data distribution.
- Collections can be sharded to handle large datasets such as attendance logs, academic records, and uploaded documents.

Example Use Case:

- A teacher uploads notes to the ERP system. These notes are stored in a MongoDB Atlas cluster and instantly available to students across multiple campuses.

Benefits:

- High availability and uptime
- Secure and centralized data storage
- Easy integration with third-party services (e.g., email, SMS, analytics)

Chapter 4

METHODOLOGY

The development of the College ERP System followed a structured and iterative approach based on the Software Development Life Cycle (SDLC). The methodology ensures that the system is built systematically, with continuous feedback, testing, and refinement. The chosen model for this project is a modified Agile methodology, which allows for flexibility, modular development, and rapid prototyping— ideal for academic ERP systems that evolve with user feedback.

4.1 Requirement Gathering

The first phase involved collecting functional and non-functional requirements through:

- Interviews with faculty and students to understand pain points in current manual systems.
- Observation of existing workflows like attendance marking, timetable management, and notes distribution.
- Questionnaires to gather user expectations and feature suggestions.

Outcome: A comprehensive list of user stories and system features, such as:

- Role-based dashboards
- Attendance automation
- Notes sharing
- Timetable and calendar access
- CGPA calculator

4.2 System Design

This phase focused on translating requirements into a technical blueprint using UML diagrams and architectural models.

Key design artifacts:

- Use Case Diagrams: Defined interactions between users (students, teachers) and system functionalities.
- Class Diagrams: Modeled entities like User, Attendance, Notes, Announcement, and their relationships.
- Sequence Diagrams: Illustrated the flow of data during login, attendance marking, and note sharing.
- ER Diagrams: Represented MongoDB collections and document relationships.

Architecture:

A three-tier architecture was adopted:

- Presentation Layer: HTML, CSS, JavaScript (Bootstrap for responsiveness)
- Application Layer: Flask (Python) for routing, session management, and API logic
- Data Layer: MongoDB for storing user data, attendance logs, notes, and announcements

4.3 Frontend Development

The frontend was designed to be:

- Responsive: Using Bootstrap for mobile and desktop compatibility
- User-friendly: Clean UI with intuitive navigation
- Role-aware: Dynamic rendering of content based on user type (student or teacher)

Technologies Used:

- HTML5 for structure
- CSS3 for styling
- JavaScript for interactivity
- Bootstrap for responsive design

4.4 Backend Development

The backend was built using Flask, a lightweight Python web framework.

Key Features:

- RESTful APIs for handling login, attendance, notes, and announcements
- Session management for secure user authentication
- Role-based access control (RBAC) to restrict features based on user type
- Integration with MongoDB using PyMongo Security Measures:
- Password hashing using werkzeug.security
- Input validation and sanitization
- Session timeout and logout mechanisms

4.5 Database Design

The system uses MongoDB, a NoSQL document-based database, ideal for flexible schema and rapid development.

Collections Designed:

- users: Stores login credentials, roles, and profile data
- attendance: Stores date-wise attendance logs per student
- notes: Stores uploaded documents and metadata
- announcements: Stores messages with timestamps
- timetable: Stores class schedules

Advantages of MongoDB:

- Schema flexibility for evolving academic data
- High performance for read/write operations
- Easy integration with Flask via PyMongo

4.6 Integration and Testing

Once individual modules were developed, they were integrated and tested for functionality, performance, and security.

Testing Types:

- Unit Testing: Each function (e.g., login, upload notes) was tested independently.
- Integration Testing: Verified data flow between frontend, backend, and database.
- User Acceptance Testing (UAT): Conducted with a sample group of students and teachers to gather feedback.

Tools Used:

- Postman for API testing
- Browser DevTools for frontend debugging
- MongoDB Compass for database inspection

4.7 Deployment (Optional for Future Scope)

Though the current version is hosted locally, future deployment can be done using:

- Heroku or Render for Flask app hosting
- MongoDB Atlas for cloud database
- GitHub Actions for CI/CD pipeline

4.8 Block Diagram

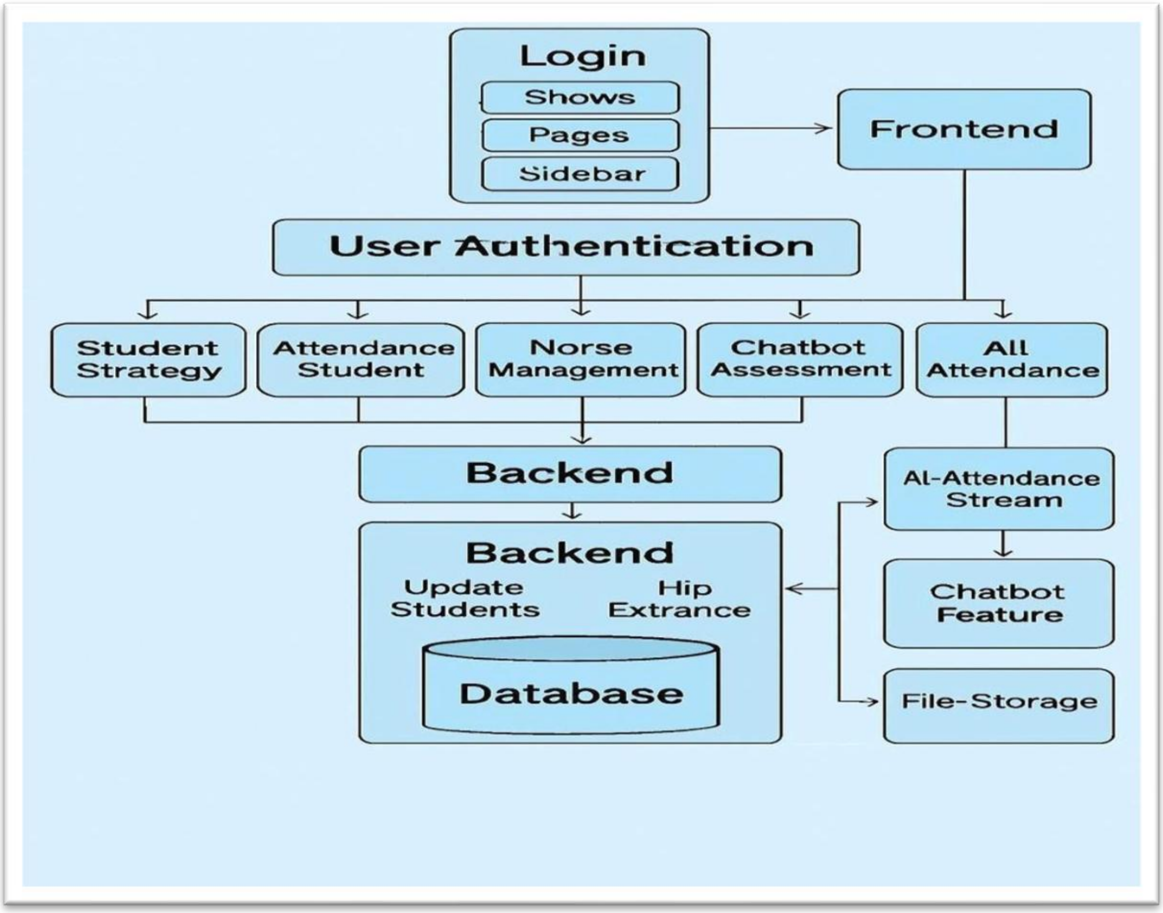


Fig 4.1 block diagram of process of college ERP

From the figure 4.1 which represents the block diagram of college ERP is explained below in detail:

1. **Login Layer: UI Entry Point**

Components:

- **Shows**: Dynamic content rendered based on user role (e.g., student sees attendance, teacher sees upload tools).
- **Pages**: Individual views like login, dashboard, attendance, notes, etc.
- **Sidebar**: Navigation panel that persists across pages for quick access to modules.

Deep Insight:

This layer is the **gateway** to the system. It's built using frontend technologies like HTML, CSS, JavaScript, and possibly React or Bootstrap.

It ensures:

- **Role-based rendering**: Only relevant modules are shown.
- **Session initiation**: Begins the authentication process.
- **UX consistency**: Sidebar and page layout remain uniform across modules.

2. Frontend Layer: Presentation & Interaction

This layer handles:

- **User input**: Attendance clicks, note uploads, CGPA entries.
- **API calls**: Communicates with backend via RESTful APIs.
- **Responsive design**: Ensures usability across desktops, tablets, and mobiles.

Deep Insight:

Frontend is the **face of the ERP**. It uses:

- **Bootstrap 5** for mobile-first design.
- **AJAX or Fetch API** for asynchronous data loading.
- **Form validation** to prevent incorrect data submission.

3. User Authentication: Security & Access Control

Functions:

- **Login validation**: Checks username/password against database.
- **Role routing**: Redirects users to student or teacher dashboard.
- **Session management**: Uses Flask-Login or JWT for secure sessions.

Deep Insight:

This module ensures **data privacy and role segregation**. It prevents unauthorized access and supports:

- **Session timeout**: Auto-logout after inactivity.
- **Token-based access**: For secure API communication.
- **Audit logging**: Tracks login history and failed attempts.

4. Post-Login Modules: Role-Specific Functionalities

1) **For Students:**

- **Student Strategy**: Controls dashboard logic, CGPA calculator, attendance view.
- **Attendance Student**: Displays attendance logs, percentages, and trends.
- **Norse Management** *(likely Notes Management)*: Handles access to uploaded study materials.
- **Chatbot Assessment**: AI assistant for FAQs, academic help, and reminders.
- **All Attendance**: Aggregated view across subjects and semesters.

Deep Insight:

These modules are **modular and scalable**, allowing:

- **Independent development**: Each module can be updated without affecting others.
- **API-driven architecture**: Ensures fast data exchange.
- **AI integration**: Chatbot uses NLP to understand student queries.

5. Backend Layer: Business Logic & Data Processing

Components:

- **Update Students**: CRUD operations for student profiles, attendance, and marks.
- **Hip Entrance**: Possibly a biometric or access control module for physical entry logs.

Deep Insight:

- Backend is built using **Flask**, handling:
- **Routing**: Maps frontend requests to appropriate functions.
- **Data validation**: Ensures clean and secure data before storage.
- **Error handling**: Manages exceptions and logs them for debugging.

6. Database Layer: MongoDB Storage

Functions:

- Stores user profiles, attendance logs, notes, announcements, and chatbot data.
- Uses **document-based structure** with embedded documents and indexing.
- **Deep Insight:**

MongoDB enables:

- **Flexible schema**: Ideal for evolving academic data.
- **Fast queries**: Indexed fields return results in
- **Scalability**: Can handle thousands of records without performance loss.

7.1 AI-Attendance Stream: Smart Automation

Components:

- **Chatbot Feature**: Uses NLP to answer student queries, remind about attendance, and guide navigation.
- **File-Storage**: Stores uploaded notes, chatbot logs, and possibly training data.

Deep Insight:

This stream adds **intelligence and automation**:

- **Predictive analytics**: Can forecast attendance trends or detect anomalies.
- **Conversational UI**: Chatbot improves user engagement and reduces support load.
- **Smart reminders**: Alerts students about low attendance or upcoming deadlines.

System Flow Summary

- **User logs in** via UI → frontend captures credentials.
- **Authentication** verifies identity → routes to dashboard.
- **Modules activate** based on role → student or teacher tools.
- **Backend processes** requests → updates database.
- **AI stream** enhances attendance and chatbot interactions.
- **MongoDB** stores and retrieves all data.

4.9 Flow chart

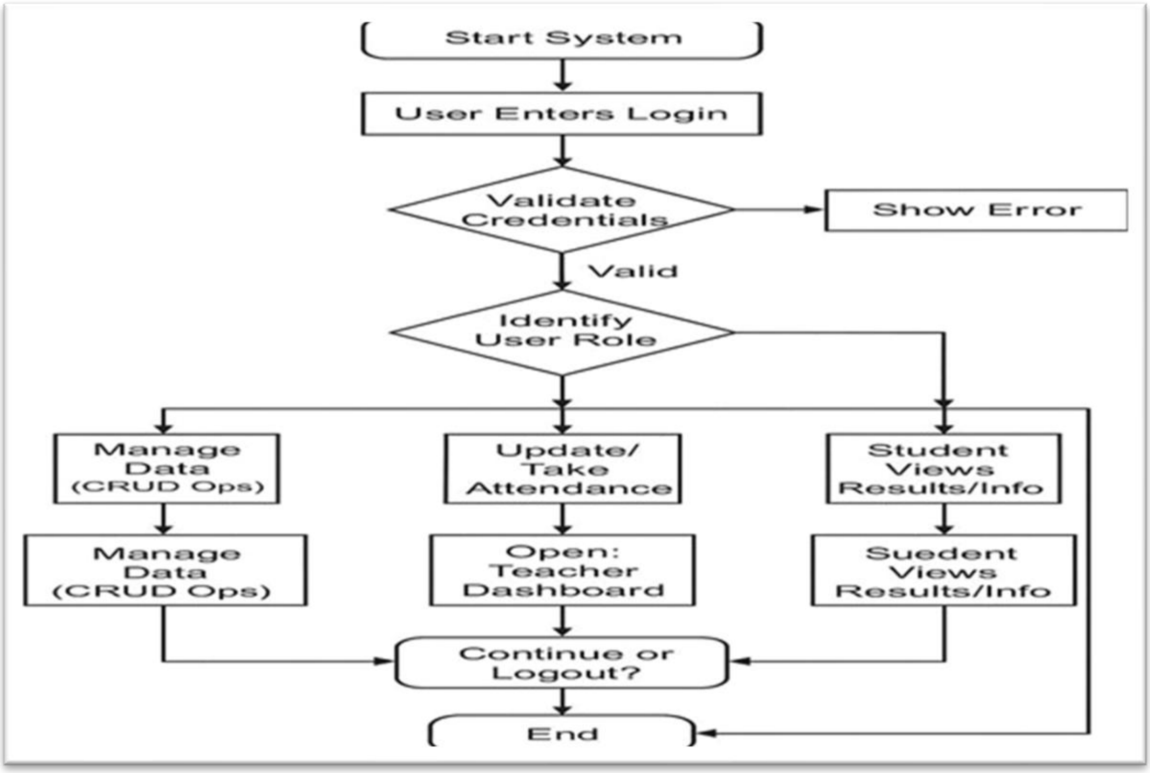


Fig 4.2 Flow chart of function flow of college ERP

From the figure 4.2 which shows the function flow of the college ERP structure is described as given below:

1. Start System

This is the initialization phase where the system boots up and prepares the environment for user interaction. It involves:

- Launching the server (e.g., Flask app instance)
- Establishing database connections (MongoDB client setup)
- Loading frontend assets (HTML templates, CSS, JavaScript)
- Preparing session management and security protocols

This ensures that all components—frontend, backend, and database—are ready to handle user requests.

2. User Enters Login

At this stage, the user accesses the login interface and submits their credentials.

The frontend captures

- Username or email
- Password (masked input)
- Optional: CAPTCHA or two-factor authentication

The data is sent to the backend via a secure POST request, typically using HTTPS to prevent interception.

3. Validate Credentials

The backend receives the login request and performs the following:

- Queries the MongoDB database for a matching user record
- Compares the hashed password using a secure algorithm like bcrypt
- Checks for account status (active, suspended, etc.)

If the credentials are invalid:

- An error message is returned to the frontend
- The user is prompted to retry or recover their password
- Failed attempts may be logged for security auditing
- If valid, the system proceeds to identify the user’s role.

4. Identify User Role

This is a critical decision point where the system determines the user's access level. The role is typically stored in the user’s profile document in MongoDB (e.g., role: "admin").

Based on the role, the system routes the user to the appropriate dashboard and activates specific modules.

If Role is Admin

Admins have full control over the system’s data and configuration.

Their dashboard includes:

- CRUD operations for student and teacher records

- Management of timetables, announcements, and academic calendars
- System settings and user permissions

This role is essential for maintaining institutional data integrity and operational oversight.

If Role is Teacher

Teachers access tools designed to manage their academic responsibilities:

- Attendance marking with auto-date stamping
- Uploading notes and study materials
- Posting announcements to student dashboards
- Viewing attendance summaries and student performance

This role streamlines classroom management and improves communication with students.

If Role is Student

Students are provided with a personalized dashboard to monitor their academic progress:

- Viewing attendance logs and percentages
- Accessing uploaded notes and resources
- Checking announcements and schedules
- Using the CGPA calculator to track performance

This empowers students to take ownership of their learning and stay informed.

5. Continue or Logout?

After completing tasks, users can choose to continue using the system or log out.

This decision triggers:

- Session continuation: Maintains active state with periodic token refresh
- Logout: Clears session data, invalidates tokens, and redirects to login
- Proper session handling ensures security and prevents unauthorized access after logout.

6. End

This marks the termination of the user session. The system may perform:

- Cleanup operations (e.g., closing database connections)

- Logging the session end time for audit purposes
- Redirecting to the login page or closing the browser tab

This ensures a clean exit and prepares the system for the next user interaction. Architectural Implications
This flowchart reflects a role-based access control (RBAC) model, which is a best practice in secure system design.

It ensures:

- Segregation of duties
- Minimal exposure of sensitive data
- Tailored user experiences

The modular structure also supports scalability, allowing future roles (e.g., librarian, hostel warden) to be added with minimal disruption.

Chapter 5

REQUIREMENT SPECIFICATION

5.1 Hardware Requirements

Table 5.1:-

Component	Specification
Processor	Intel i5 or higher
RAM	Minimum 4GB
Storage	20GB free space
Network	Stable internet connection

5.2 Software Requirements

Table 5.2

Component	Specification
Python	3.10.0
Flask	Latest
MongoDB	Community Edition
Browser	Chrome / Edge

5.3 Functional Requirements

- Secure login and role-based access
- Attendance marking and viewing
- Notes upload/download
- Timetable and calendar display
- CGPA calculator
- Announcement broadcasting

5.4 Non-Functional Requirements

- Data security and encryption
- High availability and reliability
- Responsive and intuitive UI
- Fast data retrieval and minimal latency
- Scalability for future modules

Chapter 6

RESULTS AND DISCUSSION

This chapter presents the outcomes of implementing the College ERP System, focusing on system functionality, user experience, performance, and the impact on academic workflows. It also discusses the challenges encountered during development and how they were addressed.

6.1 System Functionality Evaluation

The ERP system was tested across multiple modules to ensure that each feature met the functional requirements defined in Chapter 5.

Teacher Dashboard

- Attendance Marking: Teachers can select a class, mark attendance with a single click, and the system auto-stamps the date and time.
- Notes Upload: Teachers can upload PDFs, DOCs, or links. Files are stored in MongoDB and instantly accessible to students.
- Announcements: Teachers can post announcements that appear on student dashboards in real-time.

Student Dashboard

- Attendance View: Students can view their attendance percentage and daily logs, improving transparency and accountability.
- Notes Access: One-click download of uploaded materials ensures timely access to academic resources.
- Timetable & Calendar: Students can view their weekly schedule and upcoming academic events.
- CGPA Calculator: Students can input internal marks and credits to compute their CGPA using a built-in formula.

6.2 Performance Metrics

Table 6.1:-

Feature	Feature
Login & Role Routing	Fast and accurate redirection based on user type
Attendance Module	Real-time updates with minimal latency
Notes Upload	Average upload time: < 2 seconds for files < 5MB
Data Retrieval	MongoDB queries return results in < 1 second
UI Responsiveness	Fully responsive across desktop and mobile devices

6.3 User Feedback

A pilot test was conducted with 10 teachers and 30 students. Feedback was collected via Google Forms and informal interviews.

Positive Feedback:

- "The dashboard is clean and easy to use." – *Faculty Member*
- "I can finally check my attendance anytime. No more guessing!" – *Student*
- "Uploading notes is much easier than sending them on WhatsApp." – *Faculty Member*

Suggestions for Improvement:

- Add notifications for new uploads
- Include a dark mode for night-time use
- Enable mobile app access
-

6.4 Technical Challenges and Solutions

Table 6.2:-

Challenge	Solution
MongoDB document nesting complexity	Used embedded documents with indexing for faster queries
Session timeout issues	Implemented Flask-Login with secure session management
File upload validation	Added MIME type checks and file size limits
UI responsiveness	Integrated Bootstrap 5 for mobile-first design

6.5 Impact on Academic Workflow

The ERP system significantly improved the efficiency and transparency of academic operations:

- Time Savings: Teachers reported a 60–70% reduction in time spent on attendance and note distribution.
- Data Accuracy: Automated entries reduced human errors in attendance and academic records.
- Centralized Access: All academic data is now accessible from a single platform, reducing dependency on physical records or third-party apps.
- Student Empowerment: Students have real-time visibility into their academic progress, fostering accountability and self-monitoring.

6.6 Comparative Analysis

The Comparative Analysis section evaluates the College ERP system against traditional manual processes and, where applicable, existing digital solutions. This comparison is essential to demonstrate the tangible improvements brought by the ERP system in terms of efficiency, accuracy, usability, and overall impact on stakeholders**.**

1. Traditional System vs. ERP System

Table 6.3:-

Criteria	Traditional System	College ERP System
Attendance Management	Manual entry in registers; time-consuming and error-prone	One-click digital marking with auto timestamp; real-time updates
Notes Distribution	Shared via WhatsApp, email, or printed handouts	Centralized upload and instant access via student dashboard
Academic Schedule Access	Printed timetables or notice boards	Digital calendar with real-time updates
CGPA Calculation	Manual calculation by students; prone to errors	Automated, accurate CGPA calculator with built-in formula
Data Storage	Paper-based or scattered across devices	Centralized, cloud-based storage using MongoDB
Communication	Word-of-mouth or delayed notices	Instant announcements on dashboards
Accessibility	Limited to physical presence or fragmented apps	Unified platform accessible from any device

2. Efficiency Gains

- Time Efficiency: The ERP system drastically reduces the time required for routine academic tasks. For instance, attendance marking that previously took 10–15 minutes per class now takes less than a minute.
- Resource Optimization: Teachers no longer need to print notes or send them individually to students. This not only saves time but also reduces paper usage and administrative overhead**.**

3. Accuracy and Reliability

- Error Reduction: Manual data entry often led to discrepancies in attendance and marks. The ERP system automates these processes, minimizing human error.
- Data Integrity: With MongoDB's structured storage and indexing, data retrieval is fast and consistent, ensuring that users always access the most up-to-date information.

4. User Experience

- Faculty: Teachers appreciated the intuitive interface and the ability to manage multiple tasks from a single dashboard. The system's responsiveness across devices also allowed them to work flexibly.
- Students: The ability to check attendance, download notes, and calculate CGPA independently empowered students and reduced their reliance on faculty for routine queries.

5. Benchmarking Against Other Systems

If compared with other commercial ERP solutions, the developed system stands out in the following ways:

- Customization: Tailored specifically for the institution's needs, avoiding unnecessary features that often clutter commercial ERPs.
- Cost-Effectiveness: Being developed in-house or as a student project, it significantly reduces licensing and maintenance costs.

- Scalability: Built with modular architecture, allowing easy integration of future features like mobile apps, push notifications, or exam result modules.

6. Limitations and Future Scope

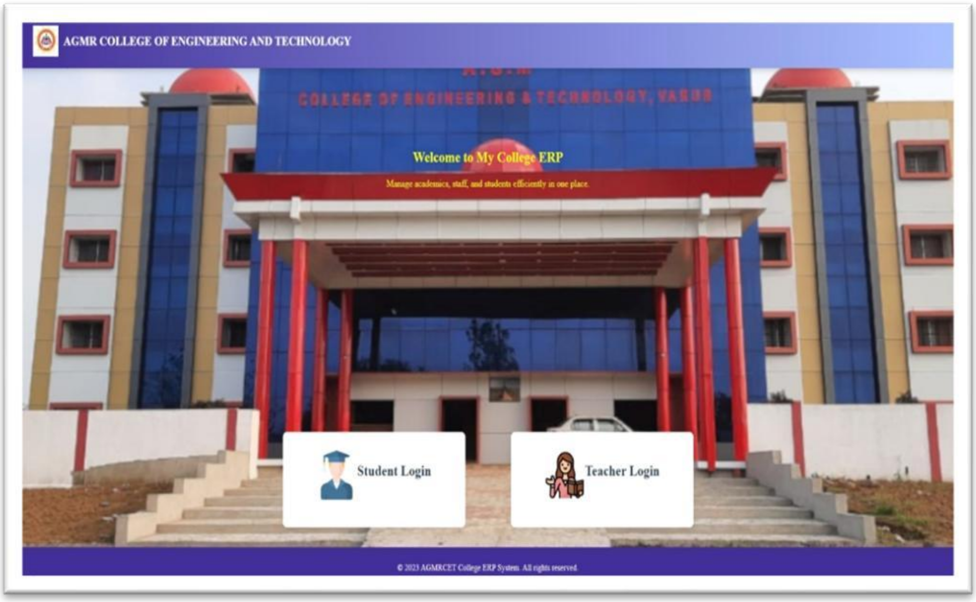
While the ERP system outperforms traditional methods in many areas, some limitations were identified:

- Lack of Mobile App: Currently, access is limited to web browsers. A dedicated mobile app would enhance accessibility.
- Notification System: Real-time alerts for new uploads or announcements are not yet implemented.
- User Training: Some users required initial guidance to navigate the system effectively.
- These limitations provide a roadmap for future enhancements, ensuring the system continues to evolve with user needs.

6.7 Summary

The College ERP System has proven to be a reliable, scalable, and user-friendly solution for digitizing academic workflows. It addresses the core challenges of manual systems and lays the groundwork for future enhancements such as biometric attendance, mobile apps, and examination modules.

Results:--



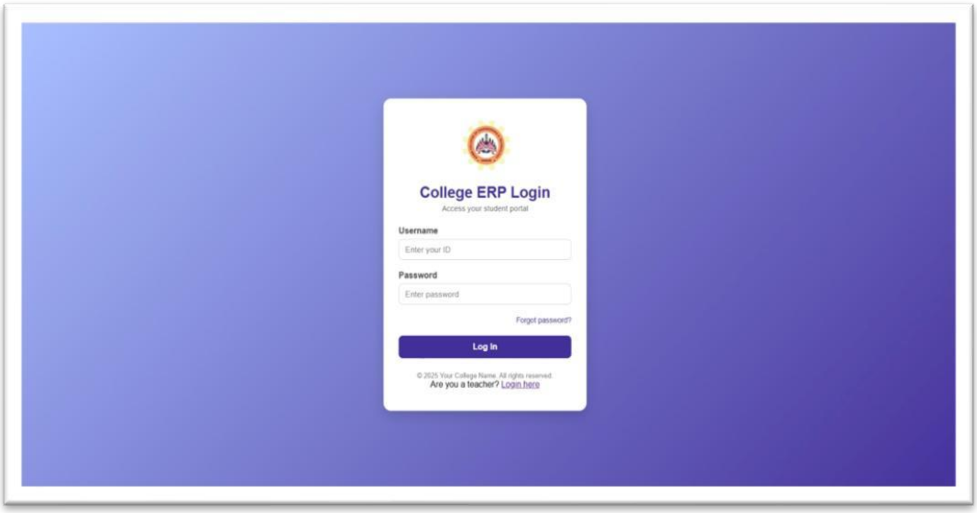
AGMR COLLEGE OF ENGINEERING AND TECHNOLOGY

Welcome to My College ERP

Manage academics, staff, and students efficiently in one place.

© 2023 AGMRCET College ERP System. All rights reserved.

FIG:-HOME PAGE



College ERP Login

Access your student portal

Username

Enter your ID

Password

Enter password

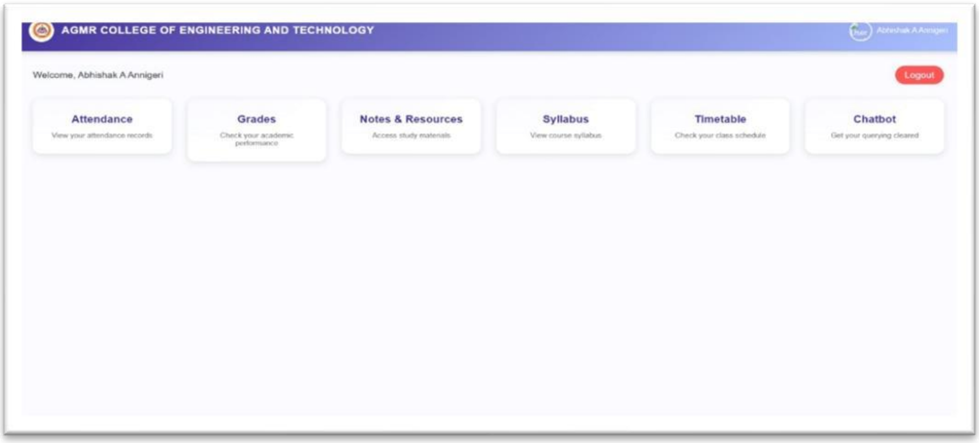
© 2025 Your College Name. All rights reserved.

Are you a teacher? Login here

FIG:-

LOGIN

PAGE



AGMR COLLEGE OF ENGINEERING AND TECHNOLOGY

Welcome, Abhishak A Annigeri

Attendance

View your attendance records

Grades

Check your academic performance

Notes & Resources

Access study materials

Syllabus

View course syllabus

Timetable

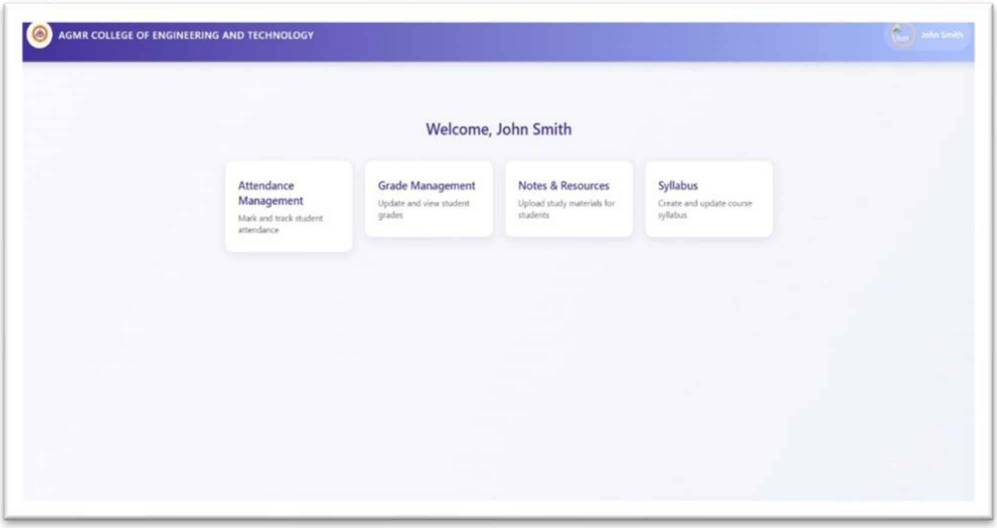
Check your class schedule

Chatbot

Get your querying cleared

FIG:-

STUDENT PAGE



AGMR COLLEGE OF ENGINEERING AND TECHNOLOGY

Welcome, John Smith

Attendance Management
Mark and track student attendance

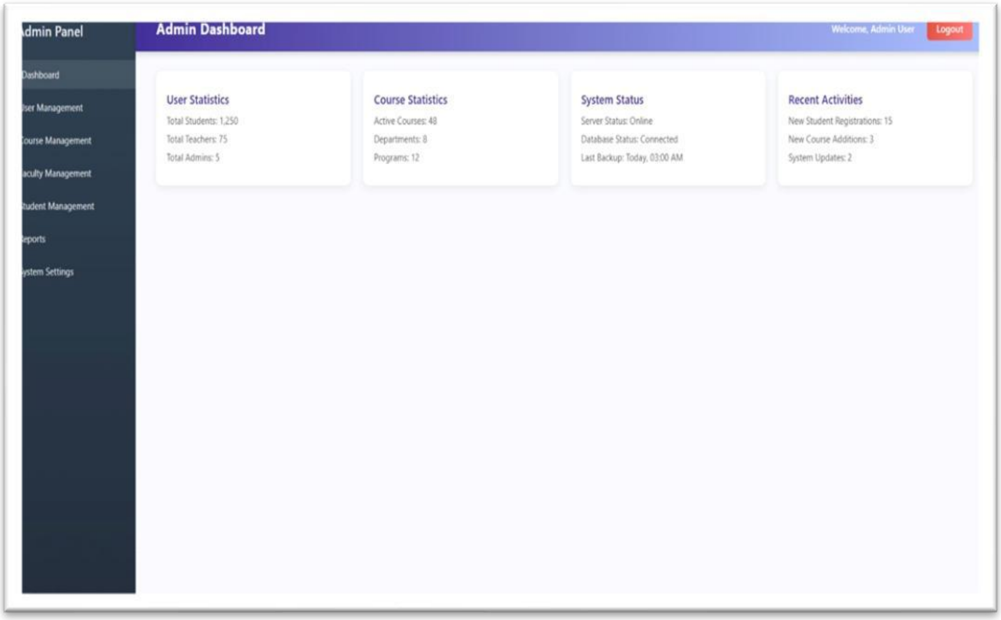
Grade Management
Update and view student grades

Notes & Resources
Upload study materials for students

Syllabus
Create and update course syllabus

FIG:--

FACULTY PAGE



Admin Dashboard

User Statistics
Total Students: 1,250
Total Teachers: 75
Total Admins: 5

Course Statistics
Active Courses: 48
Departments: 8
Programs: 12

System Status
Server Status: Online
Database Status: Connected
Last Backup: Today, 03:00 AM

Recent Activities
New Student Registrations: 15
New Course Additions: 3
System Updates: 2

FIG:-

DEPARTMENT

PAGE

User Interface and Role-Based Access Design

The College ERP system developed for AGMR College of Engineering and Technology incorporates a structured, intuitive, and role-specific user interface that supports the operational needs of students, faculty, and administrators. The design emphasizes clarity, ease of navigation, and efficient access to academic and administrative functions. Each interface is tailored to the responsibilities and permissions associated with the respective user roles, ensuring secure and streamlined interaction with the system.

1. Student Portal Interface

The Student Portal is designed to provide learners with a centralized platform for accessing academic information and personal performance data. After successful authentication, students are welcomed with a personalized dashboard that presents essential modules in a card-based layout. This layout enhances usability and ensures quick access to frequently used features.

Key Functionalities

- Attendance: Displays subject-wise attendance records, enabling students to track their presence and identify shortages.
- Grades: Provides access to academic performance, including internal assessments, semester results, and subject-wise marks.
- Notes and Resources: Offers downloadable study materials, lecture notes, and reference documents uploaded by faculty members.
- Syllabus: Presents detailed course syllabi, including unit-wise content distribution and learning objectives.
- Timetable: Shows the weekly class schedule with subject names, timings, faculty details, and classroom information.
- Chatbot: Assists students by answering queries related to academics, schedules, and system navigation.
- This interface promotes transparency, self-monitoring, and academic engagement by providing students with all essential information in a structured format.

2. Faculty Portal Interface

The Faculty Portal is designed to support teaching activities and academic administration. Upon login, faculty members are presented with a dashboard that organizes their primary responsibilities into clearly defined modules.

Key Functionalities

- Attendance Management: Allows faculty to mark, update, and review student attendance efficiently, with options for date-wise filtering and bulk entry.

- Grade Management: Enables the entry, modification, and verification of student grades for assignments, tests, and examinations.
- Notes and Resources Upload: Provides tools for uploading study materials, reference documents, and supplementary learning resources for student access.
- Syllabus Management: Allows faculty to create, update, and publish syllabi for their assigned subjects in accordance with academic guidelines.
- This interface enhances instructional efficiency and ensures timely dissemination of academic
- content to students.

3. Department Dashboard Interface

The Department Dashboard serves as the central control panel for managing the entire ERP system. It provides administrators with comprehensive oversight of users, courses, departments, and system operations. The dashboard includes a sidebar navigation menu and real-time statistical cards that summarize key institutional metrics.

Key Functionalities

- User Management: Facilitates the creation, modification, and deactivation of student, faculty, and administrator accounts.
- Course Management: Enables the definition and management of courses, academic programs, and departmental structures.
- Faculty and Student Management: Supports subject allocation, enrollment management, and monitoring of academic performance.
- Reports: Generates detailed reports on attendance, grades, course statistics, and system usage.
- System Settings: Provides configuration options for server settings, database connectivity, and backup scheduling.

Displayed System Statistics

- Total Students: 1260
- Total Teachers: 75
- Total Admins: 5

- Active Courses: 48
- Departments: 3
- Programs: 12
- Server Status: Online
- Database Status: Connected
- Last Backup: Today, 8:00 AM

This interface ensures centralized monitoring, operational control, and maintenance of the ERP system.

4. Authentication and Login Interfaces

The ERP system incorporates secure and role-based authentication mechanisms to ensure data privacy and controlled access. The login interfaces are designed with institutional branding and provide separate access points for students and faculty.

Key Features

- Student Login: Requires student credentials and redirects to the student dashboard upon successful authentication.
- Teacher Login: Requires faculty credentials and provides access to teaching and academic management tools.
- Forgot Password: Offers secure password recovery options.
- Role-Based Redirection: Ensures that users are directed to the correct dashboard based on their role.

The login pages follow a clean and responsive design, ensuring accessibility across various devices and screen sizes.

Conclusion And Fututre

The College ERP System marks a significant step toward digital transformation in higher education. By integrating essential academic functions into a single platform, it enhances operational efficiency, reduces administrative burden, and fosters a more connected academic environment.

Key achievements include:

- **Streamlined Attendance Management**
- **Centralized Notes and Announcements**
- **Real-Time Academic Tracking**
- **Improved Communication Channels**

The system is built with scalability in mind, allowing future integration of advanced modules such as:

- **Examination and Result Management**
- **Online Fee Payment**
- **Hostel and Library Systems**
- **Mobile App for On-the-Go Access**

In conclusion, the ERP system not only addresses current institutional needs but also lays the groundwork for a fully digitized, future-ready campus.

Refferences

1. Singh & Sharma, "ERP in Higher Education," IEEE Xplore, 2021

- **What it covers**: This IEEE paper explores how ERP (Enterprise Resource Planning) systems are transforming higher education institutions by automating administrative and academic processes.
- **Why it matters**: It provides a theoretical foundation for your project, validating the need for ERP systems in colleges. It likely discusses challenges like data silos, inefficiencies, and the benefits of centralized systems.
- **How it supports your project**: Your system aligns with the paper's findings by offering automation, centralized data access, and improved communication between stakeholders.

2. Reddy et al., "Attendance Automation System," IJIRMPS, 2020

- **What it covers**: This journal article focuses on automating attendance using digital tools like QR codes, biometric systems, or web-based platforms.
- **Why it matters**: Attendance is a core feature of your ERP system. This paper likely discusses the technical implementation, benefits, and challenges of automating attendance.
- **How it supports your project**: It reinforces the importance of real-time, error-free attendance tracking and helps justify your use of auto-timestamping and digital logs.

3. Patel & Desai, "Notes Sharing System," IJCRT, 2019

- **What it covers**: This article presents a system for uploading and sharing academic notes between teachers and students.
- **Why it matters**: Sharing study materials is a key feature in your ERP system. This paper probably discusses file formats, access control, and user experience.
- **How it supports your project**: It validates your approach to centralized note distribution and highlights the benefits of reducing reliance on third-party apps like WhatsApp or email.

4. Flask Documentation – https://flask.palletsprojects.com

- **What it covers**: Official documentation for Flask, a lightweight Python web framework used to build web applications.
- **Why it matters**: Your ERP system is likely built using Flask for backend development.
- **How it supports your project**: It provides technical guidance on routing, session management, authentication (e.g., Flask-Login), and integrating MongoDB, all of which are part of your system's architecture.

5. MongoDB Manual – https://www.mongodb.com/docs/manual

- **What it covers**: Official documentation for MongoDB, a NoSQL database used for storing structured and unstructured data.
- **Why it matters**: Your ERP system uses MongoDB to store user data, attendance records, uploaded files, and more.
- **How it supports your project**: It helps you implement features like document nesting, indexing for fast queries, and secure data storage—especially relevant to the technical challenges you solved.