COLLEGE MANAGEMENT SYSTEM

A MINI PROJECT REPORT

SUBMITTED BY

BALAJI S	221701009
UDAY KIRAN K	221701062
SHARVESH S	221701052
RAVEEN P	221701046

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND DESIGN

RAJALAKSHMI ENGINEERING COLLEGE

THANDALAM CHENNAI – 602105

ANNA UNIVERSITY: CHENNAI 600625

BONAFIDE CERTIFICATE

Certified that this project report “COLLEGE MANAGEMENT SYSTEM” is the Bonafide work of “BALAJI S (221701009), UDAY KIRAN K(221701062), RAVEEN P (221701046), SHARVESH S (221701052)” who carried out the project work undermy supervision.

SIGNATURE	**SIGNATURE**
Mr.S.Uma Maheswara Roa M.E.,	Mr.Vijaykumar M.Tech.,
Professor and Head,	Asst. Professor (SS),
Computer Science and Design,	Computer Science and design,
Rajalakshmi Engineering College,	Rajalakshmi Engineering College,
Thndalam, Chennai – 602105.	Thandalam, Chennai – 602105.
EXTERNAL EXAMINER	INTERNAL EXAMINAR

ACKNOWLEGEMENT

We are highly obliged in taking the opportunity to thank our Chairman Mr. S. Meganathan, Chairperson Dr.Thangam Meganathan and our Principal Dr.S.N.Murugesan for providing all the facilities which are required to carry out this project work.

We are ineffably indebted to our H.O.D MR.S.UMA MAHESWARA RAO M.E., for his conscientious guidance and encouragement to make this project a recognizable one.

We are extremely thankful to our faculty Mr.Vijaykumar M.Tech., for his valuable guidance and indefatigable support and extend our heartfelt thanks to all the teaching and non-teaching staff of Computer Science department who helped us directly or indirectly in the completion of this project successfully.

At last but not least gratitude goes to our friends who helped us compiling the project and finally to god who made all things possible.

Any omission in this brief acknowledgement doesn’t mean lack of gratitude.

BALAJI S	221701009
UDAY KIRAN K	221701062
RAVEEN P	221701046
SHARVESH S	221701052

ABSTRACT

The College Management System is an advanced web-based application designed to streamline and automate the management of diverse college activities, catering to administrators, faculty, and students. Built using modern technologies such as Node.js, Express.js, MongoDB, HTML, CSS, and JavaScript, this system offers a robust and intuitive interface for effective handling of academic, administrative, and operational tasks. The platform empowers administrators to efficiently manage student records, faculty data, course registrations, attendance tracking, exam schedules, and fee collections. Faculty members can update grades, track attendance, and access course materials seamlessly. Students benefit from features such as accessing their academic history, class schedules, real-time notifications, and the ability to submit assignments and make payments online. With centralized data storage powered by MongoDB, the system ensures secure, scalable, and efficient handling of information. The integration of real-time updates and analytics aids in decision-making while minimizing administrative burden. This project aims to revolutionize traditional college management processes by improving data accuracy, reducing manual errors, and enhancing user convenience, thus fostering a more productive and connected academic environment.

College Management System is a web-based application designed to streamline and automate the processes of managing college-related activities for both administrators and students. Developed using a combination of technologies—Node.js, Express.js, MongoDB, HTML, CSS, and JavaScript—this system offers a user-friendly interface for effective management of academic and administrative functions. The platform allows administrators to manage student records, monitor faculty performance, handle course registrations, and update schedules, while students can view their academic history, class schedules, and submit assignments. The system centralizes data storage, ensuring secure and efficient handling of academic and administrative information. This project aims to reduce administrative workload, improve data accuracy, and provide students and faculty with easy access to essential college information in real-time. College Management System is a web-based application designed to streamline and automate the processes of managing college-related activities for both administrators and students. Developed using a combination of technologies—Node.js, Express.js, MongoDB, HTML, CSS, and JavaScript—this system offers a user-friendly interface for effective

TABLE OF CONTENTS

Page No.

CHAPTER – 1

1.1. INTRODUCTION

The College Management System is designed to meet the growing need for an automated solution that simplifies and organizes the academic and administrative functions of a college. This system offers an efficient interface for students, faculty, and administrators, enabling seamless access to academic records, schedules, and administrative operations. The system enhances transparency for stakeholders and improves the efficiency of college management processes.

1.2. SCOPE OF THE WORK

This project involves the development of a comprehensive web-based application to manage various aspects of college operations efficiently:

- **Student Module**
  - Allows students to register, view their academic records, check schedules, and access examination details.
- **Faculty Module**
  - Provides faculty members with tools to manage courses, upload materials, track attendance, and evaluate student performance.
- **Admin Module**
  - Enables administrators to oversee student enrollment, faculty assignments, fee management, timetable creation, and reporting.
- **Data Management**
  - Ensures secure and centralized storage of academic and administrative data.
- **User Interface**
  - Features an intuitive and responsive design for ease of access by all users.

1.3. PROBLEM STATEMENT

Managing college operations manually or using fragmented systems is time-consuming, error-prone, and inefficient. This can lead to delays in academic processes, lack of real-time information for stakeholders, and overall dissatisfaction. Colleges require a centralized and automated system that can handle multiple processes simultaneously, such as:

- **Student and Faculty Records Management**
- **Course Scheduling**
- **Examination Handling**
- **Fee and Attendance Tracking**

An effective system should improve data accessibility, streamline processes, and ensure secure data management while minimizing human errors.

1.4 AIM AND OBJECTIVES OF THE PROJECT

Aim: To develop an automated College Management System that simplifies administrative tasks, ensures real-time access to academic and administrative data, and supports efficient college operations.

Objectives:

1. Design and develop a centralized, user-friendly web application integrating student, faculty, and administrative modules.
2. Implement secure data storage and management for handling academic and administrative records.
3. Provide tools for administrators to manage enrollment, fee collection, schedules, and reporting.
4. Enable faculty to manage course materials, attendance, and assessments effectively.
5. Allow students to access academic records, schedules, and examination details in real-time.
6. Improve the accuracy, efficiency, and transparency of college operations through automation and reliable data handling.

CHAPTER – 2

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS

Processor	:	Intel i5
Memory Size	:	8GB (Minimum)
SDD	:	512 GB (Minimum)

2.2 SOFTWARE SPECIFICATIONS

Operating System	:	WINDOWS 11
Front – End	:	JavaScript
Back - End	:	Express.js, MongoDB
Language	:	JavaScript, Mongo query Language

CHAPTER – 3

Module Description

The **College Management System** is a web-based application designed to streamline administrative and academic operations for students, faculty, and administrators. The system leverages modern web technologies such as HTML, CSS, JavaScript (frontend), Express.js (backend), and MongoDB (database management). It provides various modules to manage tasks such as enrollment, course assignments, fee management, and academic records.

Modules

1. Database Module (MongoDB)

This module is responsible for creating and managing the database that stores user, course, and administrative data.

- **Tables**:
  - **users**: Stores user credentials (username, password, role—admin, faculty, or student).
  - **courses**: Stores course details (course ID, course name, faculty ID, schedule).
  - **students**: Contains student information (student ID, name, contact details, enrolled courses).
  - **fees**: Stores fee details (fee ID, student ID, amount, payment status).
- **Key Operations**:
  - Create, read, update, and delete records for users, courses, students, and fee details.

---

2. User Interface Module (HTML, CSS, JavaScript)

This module provides a responsive and user-friendly interface for students, faculty, and administrators.

- **Features**:
  - **Admin Dashboard**: Administrators can manage courses, assign faculty, track fee payments, and generate reports.
  - **Faculty Dashboard**: Faculty members can manage their assigned courses, track attendance, and evaluate student performance.
  - **Student Dashboard**: Students can view academic records, schedules, and fee status.
  - **Login/Signup**: Users can create accounts or log in to their respective dashboards.

---

3. Authentication and User Management Module

This module handles user authentication and ensures role-based access to features.

- **Features**:
    - **Signup**: Allows users to register with unique credentials.
    - **Login**: Provides secure login functionality for users.
    - **Role-based Access Control**:
        - Admins: Full access to all modules.
        - Faculty: Limited to managing courses and students.
        - Students: Restricted to viewing personal records.

---

4. Academic Management Module

This module facilitates the management of academic data, including courses, schedules, and performance.

- **Admin Functions**:
    - Add and assign courses to faculty members.
    - Manage student enrollments.
- **Faculty Functions**:
    - Update course schedules and content.
    - Record and manage attendance.
    - Evaluate student performance and upload grades.
- **Student Functions**:
    - View assigned courses, attendance, and grades.

---

5. Fee Management Module

This module simplifies the handling of fee-related processes.

- **Admin Functions**:
    - Add fee records and update payment status.
    - Generate payment reports.
- **Student Functions**:
    - View fee details, including due dates and payment status.
    - Pay fees online via integrated payment gateways.

---

6. Error Handling and Validation Module

This module ensures the accuracy and security of user inputs.

- **Features**:
    - Validates credentials during login/signup.
    - Ensures accurate data entry for course schedules, fee records, and academic data.
    - Displays appropriate error messages for invalid inputs or unauthorized access.

---

7. Notification and Messaging Module

This module improves communication among students, faculty, and administrators.

- **Features**:
  - Email or SMS notifications for important updates (e.g., fee deadlines, course changes, grades).
  - Message system for internal communication.

---

8. UI Customization and Features

This module enhances the user experience by implementing custom designs and accessibility features.

- **Components**:
  - **Dynamic Themes**: Customizable themes for a personalized experience.
  - **Responsive Design**: Ensures usability on mobile and desktop devices.
  - **Error Handling Messages**: Informative prompts for guiding users.

---

9. Reporting and Analytics Module

This module provides detailed reports and insights for administrators to make data-driven decisions.

- **Reports**:
  - Enrollment statistics.
  - Academic performance analytics.
  - Fee collection summaries.
- **Visualizations**: Graphical representations of key data points (e.g., attendance trends, grade distributions).

The modular structure of the **College Management System** ensures scalability, security, and usability while addressing the varied needs of students, faculty, and administrators.

CHAPTER – 4

SAMPLE CODING

```
import React, { useEffect, useState } from "react";
import Navbar from "../../components/Navbar";
import Profile from "./Profile";
import Timetable from "./Timetable";
import Marks from "./Marks";
import Notice from "../../components/Notice";
import Material from "./Material";
import { Toaster } from "react-hot-toast";
import { useLocation, useNavigate } from "react-router-dom";
const Home = () => {
  const [selectedMenu, setSelectedMenu] = useState("My Profile");
  const router = useLocation();
  const navigate = useNavigate();
  const [load, setLoad] = useState(false);
  useEffect(() => {
    if (router.state === null) {
      navigate("/");
    }
    setLoad(true);
  }, [navigate, router.state]);
  return (
    <section>
      {load && (
        <>
          <Navbar />
          <div className="max-w-6xl mx-auto">
            <ul className="flex justify-evenly items-center gap-10 w-full mx-auto my-8">
              <li
                className={`text-center rounded-sm px-4 py-2 w-1/5 cursor-pointer ease-linear duration-300 hover:ease-linear hover:duration-300 hover:transition-all transition-all ${
```

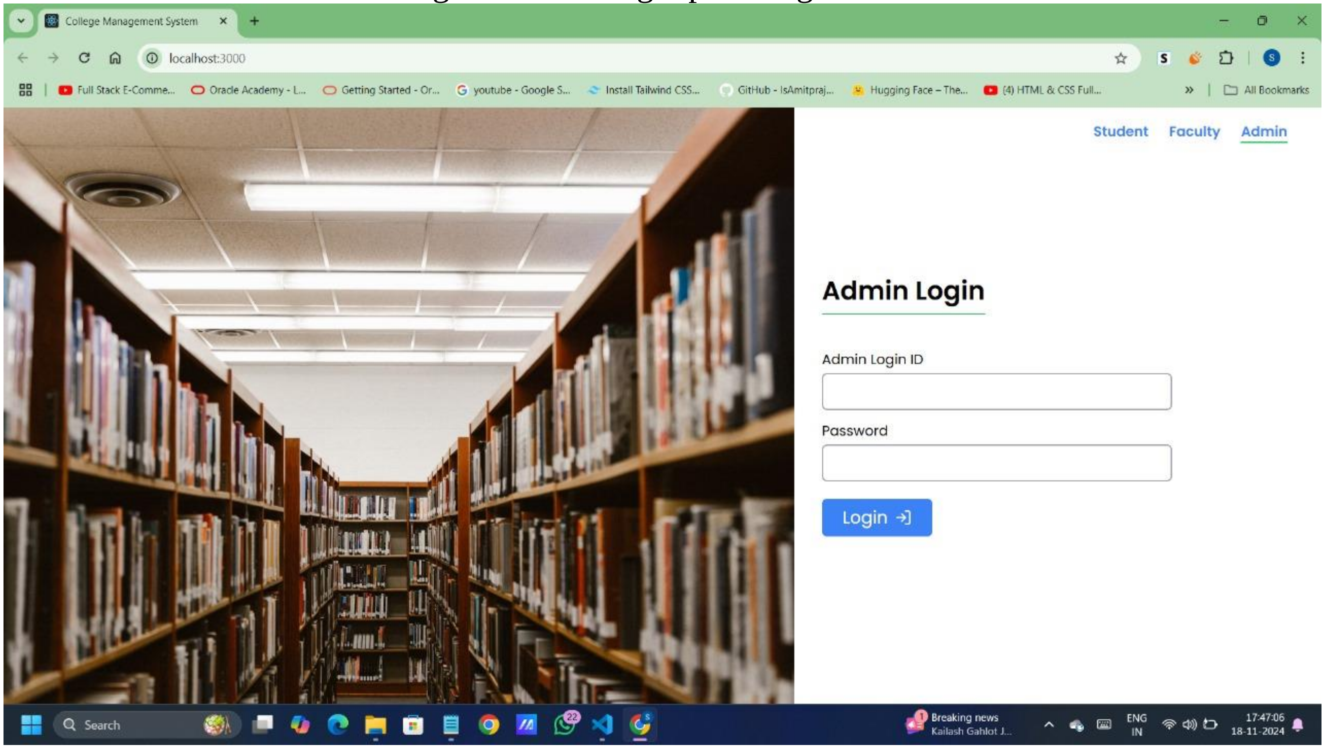
```
        selectedMenu === "My Profile"
          ? "border-b-2 pb-2 border-blue-500 bg-blue-100 rounded-sm"
          : "bg-blue-500 text-white hover:bg-blue-600 border-b-2 border-blue-500"
      }`}
      onClick={() => setSelectedMenu("My Profile")}
    >
      My Profile
    </li>
    <li
      className={`text-center rounded-sm px-4 py-2 w-1/5 cursor-pointer ease-linear duration-300 hover:ease-linear hover:duration-300 hover:transition-all transition-all ${
        selectedMenu === "Timetable"
          ? "border-b-2 pb-2 border-blue-500 bg-blue-100 rounded-sm"
          : "bg-blue-500 text-white hover:bg-blue-600 border-b-2 border-blue-500"
      }`}
      onClick={() => setSelectedMenu("Timetable")}
    >
      Timetable
    </li>
    <li
      className={`text-center rounded-sm px-4 py-2 w-1/5 cursor-pointer ease-linear duration-300 hover:ease-linear hover:duration-300 hover:transition-all transition-all ${
        selectedMenu === "Marks"
          ? "border-b-2 pb-2 border-blue-500 bg-blue-100 rounded-sm"
          : "bg-blue-500 text-white hover:bg-blue-600 border-b-2 border-blue-500"
      }`}
      onClick={() => setSelectedMenu("Marks")}
    >
      Marks
    </li>
    <li
      className={`text-center rounded-sm px-4 py-2 w-1/5 cursor-pointer ease-linear duration-300 hover:ease-linear hover:duration-300 hover:transition-all transition-all ${
         selectedMenu === "Material"
          ? "border-b-2 pb-2 border-blue-500 bg-blue-100 rounded-sm"
          : "bg-blue-500 text-white hover:bg-blue-600 border-b-2 border-
blue-500"
        }`}
        onClick={() => setSelectedMenu("Material")}
       >
        Material
       </li>
       <li
        className={`text-center rounded-sm px-4 py-2 w-1/5 cursor-pointer
ease-linear duration-300 hover:ease-linear hover:duration-300 hover:transition-
all transition-all ${
         selectedMenu === "Notice"
          ? "border-b-2 pb-2 border-blue-500 bg-blue-100 rounded-sm"
          : "bg-blue-500 text-white hover:bg-blue-600 border-b-2 border-
blue-500"
        }`}
        onClick={() => setSelectedMenu("Notice")}
       >
        Notice
       </li>
      </ul>
      {selectedMenu === "Timetable" && <Timetable />}
      {selectedMenu === "Marks" && <Marks />}
      {selectedMenu === "Material" && <Material />}
      {selectedMenu === "Notice" && <Notice />}
      {selectedMenu === "My Profile" && <Profile />}
     </div>
    </>
   )}
   <Toaster position="bottom-center" />
  </section>
 );
};

export default Home;
```

CHAPTER - 5

SCREEN SHOTS

Fig 5.1 Admin signup and signin

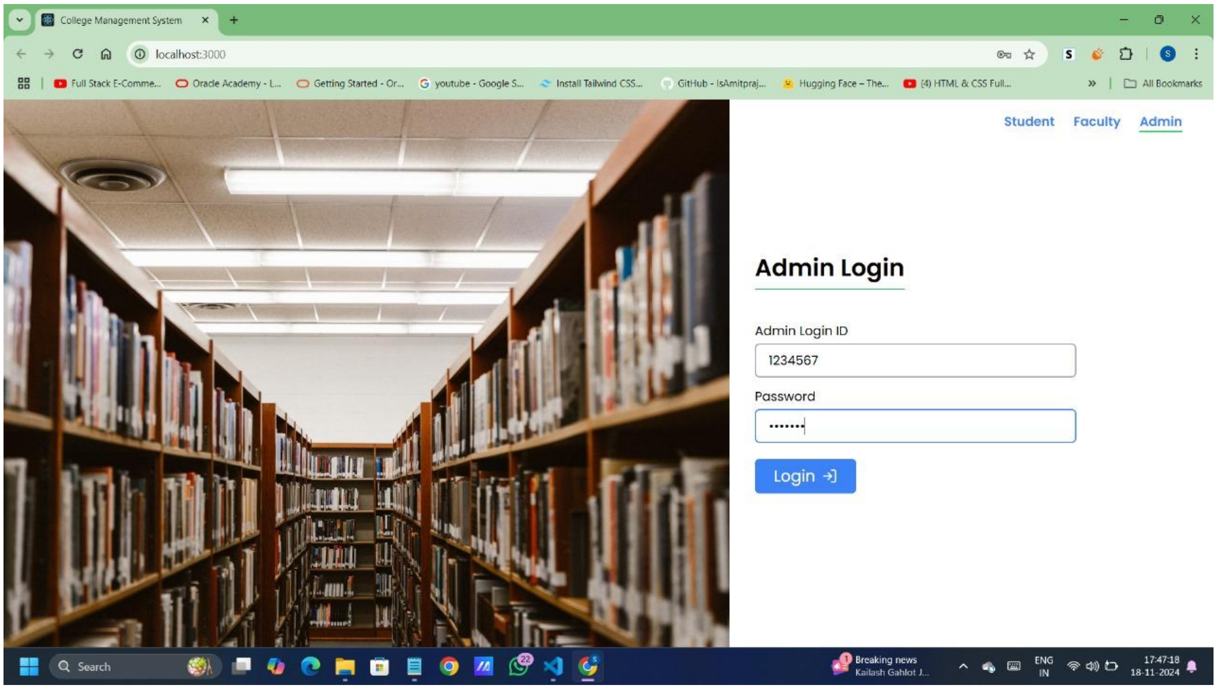


Admin Login

Admin Login ID

Password

Fig 5.2 Admin login

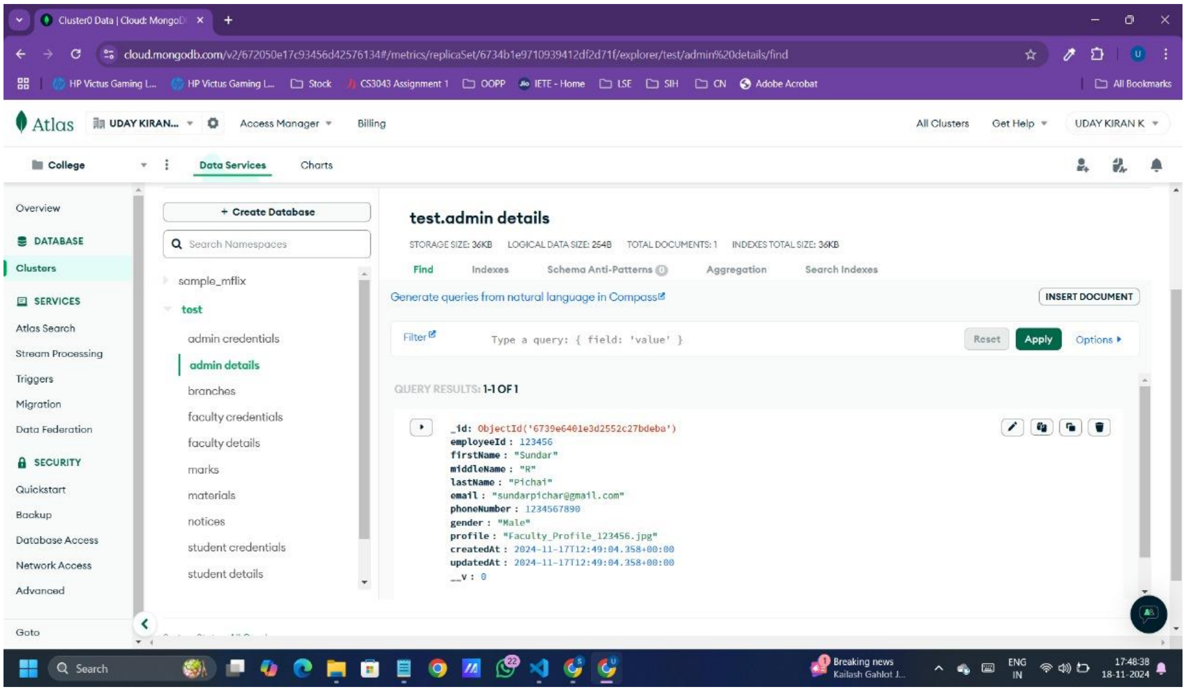


Admin Login

Admin Login ID: 1234567

Password: •••••••

Fig 5.3 Login details in Database



test.admin details

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 254B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 36KB

QUERY RESULTS: 1-1 OF 1

```
_id: ObjectId('6739e6401e3d2552c27bdeba')
employeeId : 123456
firstName : "Sundar"
middleName : "R"
lastName : "Pichai"
email : "sundarpichar@gmail.com"
phoneNumber : 1234567890
gender : "Male"
profile : "Faculty_Profile_123456.jpg"
createdAt : 2024-11-17T12:49:04.358+00:00
updatedAt : 2024-11-17T12:49:04.358+00:00
__v : 0
```

CHAPTER 6

Conclusion and Future Enhancement

The **College Management System** developed using modern web technologies provides a reliable and efficient solution for automating and managing academic and administrative processes in a college environment. Through its modular and user-friendly design, the system addresses common challenges in manual management, such as inefficiencies, human errors, and difficulty in accessing real-time data.

By streamlining key functions like student enrollment, fee tracking, course management, and academic records handling, the system enhances operational efficiency and improves user experiences for students, faculty, and administrators. Administrators can effectively oversee college operations, manage resources, and monitor performance, while students and faculty can easily access schedules, records, and relevant resources. Secure data storage ensures the integrity and confidentiality of academic and administrative information, boosting transparency and trust.

Overall, this system achieves its objective of creating a centralized, automated platform for college management. It simplifies workflows, reduces errors, and improves the quality of service provided to stakeholders.

FUTURE ENHANCEMENTS:

To make the College Management System even more robust and accessible, the following enhancements could be considered:

1. **Advanced Analytics**: Integrating predictive analytics to track student performance and identify areas of improvement.
2. **Mobile Compatibility**: Developing a mobile application to provide on-the-go access for students, faculty, and administrators.
3. **Online Payment Gateway**: Allowing online payments for fees and other transactions directly through the system.
4. **Integration with Learning Management Systems (LMS)**: Enabling seamless access to course materials, assignments, and virtual classrooms.
5. **Biometric or RFID Integration**: Enhancing attendance tracking and security through automated identification systems.
6. **Multilingual Support**: Offering support for multiple languages to cater to diverse user bases.

By incorporating these enhancements, the College Management System can evolve into a comprehensive and cutting-edge platform for managing educational institutions efficiently.

CHAPTER – 7

REFERENCES

1. https://www.w3schools.com/js/
2. https://www.tutorialspoint.com/javascript/index.html
3. https://www.mongodb.com/docs/manual/tutorial/
4. https://www.youtube.com/
5. https://en.wikipedia.org/wiki/Web_development

THANK YOU