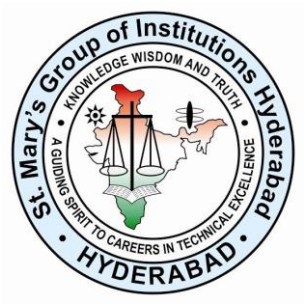
A project report on

"SCHOOL MANAGEMENT SYSTEM"

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING



Submitted by

K.INDHU 22D01A05C6

K.VARDINI 22D01A05B3

K.UMA PARMESHWARI 22D01A05C0

J.NIVAS REDDY 22D01A0591

K.VINAY KUMAR 22D01A05B7

Under the guidance of

Mr.B.MANOHAR M.Tech (PH.D)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

St. Mary's Group of Institutions Hyderabad

(Affiliated to JNTU. Hyderabad)

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

(2024-2025)

St. Mary's Group of Institutions Hyderabad
(Approved by AICTE, Permitted by Govt. of T.S., Affiliated to JNTUH, Accredited by NBA, ISO 9001 : 2000 Certified)
Near Ramoji Film City, Behind Mount Opera, Deshmukhi (V), Pochampally (M) Nalgonda (Dt.), & Batasingaram (V), Hayatnagar (M), Ranga Reddy (Dt.), Greater Hyderabad - 508 284. Under HMDA.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project work "**Co-Extracting Opinion Targets And Opinion Words From Online Reviews Based on the Word Alignment Model**" is Submitted in partial fulfilment of the requirements for the award of the Degree **Bachelor of Technology** in **COMPUTER SCIENCE & ENGINEERING** from **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD** is a bonafied work carried out under the guidance and supervision of Mr B.MANOHAR M.TECH(PH.D)

S. No	Name of the Student	Hall Ticket No.
1	**K.INDHU**	22D01AO5C6
2	**K. VARDINI**	22D01AO5B3
3	**K.UMA PARMESHWARI**	22D01AO5C0
4	**J.NIVAS REDDY**	22D01AO591
5	**K.VINAY KUMAR**	22D01AO5B7

INTERNAL GUIDE **Head of the Department**

Mr. B.MANOHAR M.TECH (PH.D) Mr. S..JOY KUMAR M.TECH (PH.D)

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

From our efforts, the success of this project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have helped us in this project. We offer praises to the almighty, without whose grace our dream would not have come true.

We thank my beloved parents for their blessings persistent help and support to do this project. We wish to express our earned gratefulness to our honorable founder and **CHAIRMAN, "Mr.Rev.K.V.K.RAO"** for his encouragement extended to us to.

We would like to thank and express our gratitude to our

PRINCIPAL, "**Dr.KAMAL HASAN**"SIR

For providing all necessary facilities for the completion of the project

We would like to express profusely our deep sense of gratitude to our **HEAD OF THE DEPARTMENT "Mr. S.JOY KUMAR M.TECH(PH.D)"** Assistant professor who extended his heartiest encouragement, Advice, and guidance throughout the project.

We are immensely grateful to our guide and project coordinator "**Mr. B.MANOHAR M.TECH(PH.D)" , Asst.prof** for his valuable inputs , able guidance, and encouragement throughout our project.

We render our sincere thanks for this great support in completing this project successfully. We thank our parents, faculty members, supporting staff and friends for their help extended during this time

BY

K.INDHU

K.VARDINI

K.UMA PARMESHWARI

J.NIVAS REDDY

K.VINAY KUMAR

DECLARATION

We are hereby declare that the results embodied in this dissertation entitled **"Coextracting Opinion Targets and Opinion Words from Online Reviews based on the Word Alignment Model"** is carried out by me during the year 2024-2025 in partial fulfilment of the award of Degree of **Bachelor of Technology** in Computer Science & Engineering from **St. Mary's Group of Institution Hyderabad.** I have not submitted the same to any other university or organisation for the award of any other degree.

Place: Hyderabad

Students Name with Signature

Date:

K.INDHU 22D01A05C6

K.VARDINI 22D01A05B3

K.UMA PARMESHWARI 22D01A05C0

J.NIVAS REDDY 22D01A0591

K.VINAY KUMAR 22D01A05B7

ABSTRACT

School Management System

The School Management System is a comprehensive console-based application designed to manage students,teachers, and courses within a school. This system facilitatesthe addition of students and teachers, the creation of courses, and the viewing of all records. Developed in Python,this application provides an efficient way to handle school management tasks through a user-friendly interface.

Key features include adding student and teacher records,creating courses with assigned teachers, and viewing detailed information about students, teachers, and courses.This project demonstrates how a simple application can significantly improve the organization and management of school-related data.

The implementation of the School Management System offers significant benefits, such as streamlined data management, enhanced organization, and improvedefficiency in handling school operations.

In today's digital era, efficient management of educational institutions is crucial for enhancing productivity and ensuring smooth operations. A School Management System (SMS) serves as a comprehensive platform designed to streamline the administrative tasks and daily operations of schools. This abstract explores the key functionalities and benefits of an SMS in modern educational settings.

An SMS integrates various modules that encompass essential aspects of school administration, including student information management, attendance tracking, academic performance monitoring, timetable scheduling, and communication with parents and staff. By centralizing these functions into a single digital interface, the system minimizes redundancy, reduces paperwork, and enhances data accuracy.

INDEX

CHAPTER-1

Introduction of the Project

The School Management System project, developed as a console-based application in Python, aims to provide an efficient and straightforward way to manage school-related data. The application allows for the addition of students and teachers, the creation of courses, and viewing of all records in a structured format.By leveraging Python's capabilities, the system handles data storage, retrieval, and manipulation seamlessly. The console-based interface ensures ease of use and accessibility, making it suitable for schools seeking areliable tool for managing their records

In the realm of modern education, the effective management of educational institutions plays a pivotal role in ensuring smooth operations and fostering an environment conducive to learning and growth. A School Management System (SMS) project aims to leverage technology to streamline administrative tasks, enhance communication channels, and provide valuable insights for informed decision-making.

The primary objective of implementing an SMS is to integrate various administrative functions into a unified platform that simplifies processes and improves efficiency. Traditionally, schools have managed student records, attendance tracking, examination schedules, and communication through disparate systems or manual methods, often leading to inefficiencies and inaccuracies. By adopting an SMS, educational institutions can automate these tasks, thereby reducing administrative burdens, minimizing errors, and freeing up valuable time for educators to focus on teaching and student development.

Moreover, an SMS enhances transparency and communication among stakeholders, including students, parents, teachers, and administrators. Through dedicated portals and mobile applications, parents can easily access their child's academic progress, attendance records, and important announcements. Teachers can efficiently manage classroom activities, submit grades, and communicate with parents, fostering a collaborative learning environment.

CHAPTER-2

2.PROBLEM DISCRIPTION

Managing student, teacher, and course records manually can be time-consuming and prone to errors. Traditional methods such as paper records or basic spreadsheets lack the efficiency and organization needed to handle large amounts of data effectively. These inefficiencies can lead to misplaced records, difficulty in accessing information, and overall poor management of school operations.

In today's educational landscape, traditional methods of managing school administration are often labor-intensive, prone to errors, and inefficient. Educational institutions face challenges such as fragmented data management, manual processes for attendance tracking and grading, limited communication channels between stakeholders, and difficulty in accessing real-time insights for informed decision-making. These inefficiencies not only hinder operational effectiveness but also impact the overall educational experience for students, teachers, and parents.

There is a pressing need for a comprehensive School Management System (SMS) that can integrate all administrative tasks into a unified platform. This system should automate routine processes such as student enrollment, attendance management, timetable scheduling, and examination management. Additionally, it should facilitate seamless communication between students, parents, teachers, and administrators through intuitive interfaces and mobile accessibility.

This problem statement identifies the current challenges faced by educational institutions in managing administrative tasks and highlights the need for an integrated School Management System to address these issues effectively. It sets the stage for designing a solution that enhances efficiency, communication, and decision-making within schools.

CHAPTER-3

Objective

To address these challenges, we propose the development of a Python-based console application for school management. This system aims to:

1.Streamline Data Management: Provide an efficient way to add and track student, teacher, and course records.

2**.Enhance Organization**: Help users maintain a well-organized database of school-related information.

3.Improve Efficiency: Facilitate easy access to and management of records, reducing the time and effort required for school management tasks.

4.**Automate Administrative Tasks**: Develop a system to automate routine administrative processes such as student enrollment, attendance management, and grading to reduce manual effort and improve accuracy.

5. **Integrate Information**: Create a unified platform that integrates various school functions including student records, academic schedules, and financial data to ensure data consistency and accessibility.

6. **Enhance Communication**: Implement features to facilitate seamless communication between students, parents, teachers, and administrators through portals, notifications, and messaging systems.

7. **Provide Real-time Insights**: Enable administrators to access real-time data and analytics on student performance, attendance trends, and resource utilization to support informed decision-making and strategic planning.

By creating a console-based application, this project aims to deliver a simple, reliable, and effective solution for schools looking to improve their management processes

CHAPTER-4

Software Requirements

Python 3.8+: The core programming language for developing the application.

The software requirements for a School Management System (SMS) project typically include a range of components to support its functionalities. Here are the essential software requirements.

1. **Operating System (OS)**:

- Server Side: Linux (e.g., Ubuntu Server, CentOS) or Windows Server

- Client Side: Windows, macOS, Linux (depending on compatibility with web browsers)

2. **Web Server**:

- Apache HTTP Server, Nginx, or Microsoft IIS for hosting the application

3. **Database Management System (DBMS)**:

- MySQL, PostgreSQL, Oracle Database, or Microsoft SQL Server for storing and managing data

4. **Programming Languages and Frameworks**:

- Backend:

  - Programming Languages: Java, Python, PHP, C#

- Frameworks: Spring Framework (Java), Django (Python), Laravel (PHP), ASP.NET (C#)

- Frontend:

- HTML5, CSS3, JavaScript (ES6+)

- Frameworks/Libraries: React.js, Angular, Vue.js

5. **Middleware**:

- Message brokers (e.g., RabbitMQ) for handling asynchronous tasks

- Caching mechanisms (e.g., Redis) for improving performance

6. **Authentication and Authorization**:

- Integration with OAuth 2.0, LDAP, or Active Directory for user authentication

- Role-based access control (RBAC) to manage permissions

7. **Data Storage and Management**:

- Implement data encryption mechanisms (e.g., AES) for sensitive data

- Implement data backup and recovery procedures

These software requirements are essential for building a robust and scalable School Management System that meets the needs of administrators, teachers, parents, and students while ensuring security, usability, and compliance with regulatory standards.

CHAPTER-5

Software Environment

Development Environment:

1. **Integrated Development Environment (IDE)**:
    - IntelliJ IDEA, Eclipse, NetBeans (for Java-based development)
    - PyCharm, VS Code, PyDev (for Python-based development)
    - Visual Studio, Visual Studio Code (for C#/.NET-based development)
2. **Version Control System**:
    - Git for version control management (GitHub, GitLab, Bitbucket for hosting repositories)
3. **Database Management**:
    - MySQL Workbench, pgAdmin, SQL Developer (for managing MySQL, PostgreSQL, Oracle databases respectively)

Server-Side Environment:

1. **Operating System**:
    - Linux (Ubuntu Server, CentOS) or Windows Server depending on deployment preferences
2. **Web Server**:
    - Apache HTTP Server, Nginx, Microsoft IIS for hosting the application
3. **Application Server**:
    - Django's built-in development server, or Gunicorn with Nginx for Python-based applications
4. **Database Management System (DBMS)**:
    - MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server (based on project requirements and scalability needs)

Python Package Manager (pip): For installing and managing Python packages and dependencies

.**Version Control System (Git)**: For tracking changes, collaborating with others, and maintaining the project's source code.

CHAPTER-6

PYTHON CODE

```
import datetime

class SchoolManagementSystem:
    def __init__(self):
        self.students = {}  # Stores student info with student ID as key
        self.teachers = {}  # Stores teacher info with teacher ID as key
        self.courses = {}   # Stores course info with course ID as key
        self.current_student_id = 0
        self.current_teacher_id = 0
        self.current_course_id = 0


    def add_student(self, name, age, grade):
        self.current_student_id += 1
        student_info = {
            "id": self.current_student_id,
            "name": name,
            "age": age,
            "grade": grade,
            "enrollment_date": datetime.datetime.now()
        }
        self.students[self.current_student_id] = student_info
        return f"Student {name} added successfully with ID {self.current_student_id}."
```

```
def add_teacher(self, name, subject):
    self.current_teacher_id += 1
    teacher_info = {
        "id": self.current_teacher_id,
        "name": name,
        "subject": subject,
        "hire_date": datetime.datetime.now()
    }
    self.teachers[self.current_teacher_id] = teacher_info
    return f"Teacher {name} added successfully with ID {self.current_teacher_id}."

def add_course(self, course_name, teacher_id):
    if teacher_id in self.teachers:
        self.current_course_id += 1
        course_info = {
            "id": self.current_course_id,
            "course_name": course_name,
            "teacher_id": teacher_id,
            "creation_date": datetime.datetime.now()
        }
        self.courses[self.current_course_id] = course_info
        return f"Course {course_name} added successfully with ID {self.current_course_id}."
    else:
        return "Teacher ID not found."
```

```
    def view_students(self):
        for student_id, student_info in self.students.items():
            print(f"ID: {student_id}, Name: {student_info['name']}, Age: {student_info['age']}, Grade: {student_info['grade']}, Enrollment Date: {student_info['enrollment_date']}")

    def view_teachers(self):
        for teacher_id, teacher_info in self.teachers.items():
            print(f"ID: {teacher_id}, Name: {teacher_info['name']}, Subject: {teacher_info['subject']}, Hire Date: {teacher_info['hire_date']}")

    def view_courses(self):
        for course_id, course_info in self.courses.items():
            if course_info['teacher_id'] in self.teachers:
                teacher_name = self.teachers[course_info['teacher_id']]['name']
            else:
                teacher_name = "Not Found"
            print(f"ID: {course_id}, Course Name: {course_info['course_name']}, Teacher: {teacher_name}, Creation Date: {course_info['creation_date']}")

def main():
    sms = SchoolManagementSystem()

    while True:
        print("\nSchool Management System")
```

```
print("1. Add Student")
print("2. Add Teacher")
print("3. Add Course")
print("4. View Students")
print("5. View Teachers")
print("6. View Courses")
print("7. Exit")

choice = input("Enter your choice: ")

if choice == '1':
    name = input("Enter student name: ")
    age = int(input("Enter student age: "))
    grade = input("Enter student grade: ")
    print(sms.add_student(name, age, grade))
elif choice == '2':
    name = input("Enter teacher name: ")
    subject = input("Enter teacher subject: ")
    print(sms.add_teacher(name, subject))
elif choice == '3':
    course_name = input("Enter course name: ")
    teacher_id = int(input("Enter teacher ID: "))
    print(sms.add_course(course_name, teacher_id))
elif choice == '4':
```

```
            sms.view_students()
        elif choice == '5':
            sms.view_teachers()
        elif choice == '6':
            sms.view_courses()
        elif choice == '7':
            print("Exiting...")
            break
        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    main()
```

CHAPTER-7

Sample Output

```
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit
Enter your choice: 1
Enter student name: Alice
Enter student age: 14
```

Enter student grade: 9

Student Alice added successfully with ID 1.

School Management System

1. Add Student

2. Add Teacher

3. Add Course

4. View Students

5. View Teachers

6. View Courses

7. Exit

Enter your choice: 2

Enter teacher name: Mr. Smith

Enter teacher subject: Mathematics

Teacher Mr. Smith added successfully with ID 1.

School Management System

1. Add Student

2. Add Teacher

3. Add Course

4. View Students

5. View Teachers

6. View Courses

7. Exit

Enter your choice: 3

Enter course name: Algebra

Enter teacher ID: 1

Course Algebra added successfully with ID 1.

School Management System

1. Add Student

2. Add Teacher

3. Add Course

4. View Students

5. View Teachers

6. View Courses

7. Exit

Enter your choice: 4

ID: 1, Name: Alice, Age: 14, Grade: 9, Enrollment Date: 2024-06-18 12:00:00.000000

School Management System

1. Add Student

2. Add Teacher

3. Add Course

4. View Students

5. View Teachers

6. View Courses

7. Exit

Enter your choice: 5

ID: 1, Name: Mr. Smith, Subject: Mathematics, Hire Date: 2024-06-18 12:00:00.000000

School Management System

1. Add Student

2. Add Teacher

3. Add Course

4. View Students

5. View Teachers

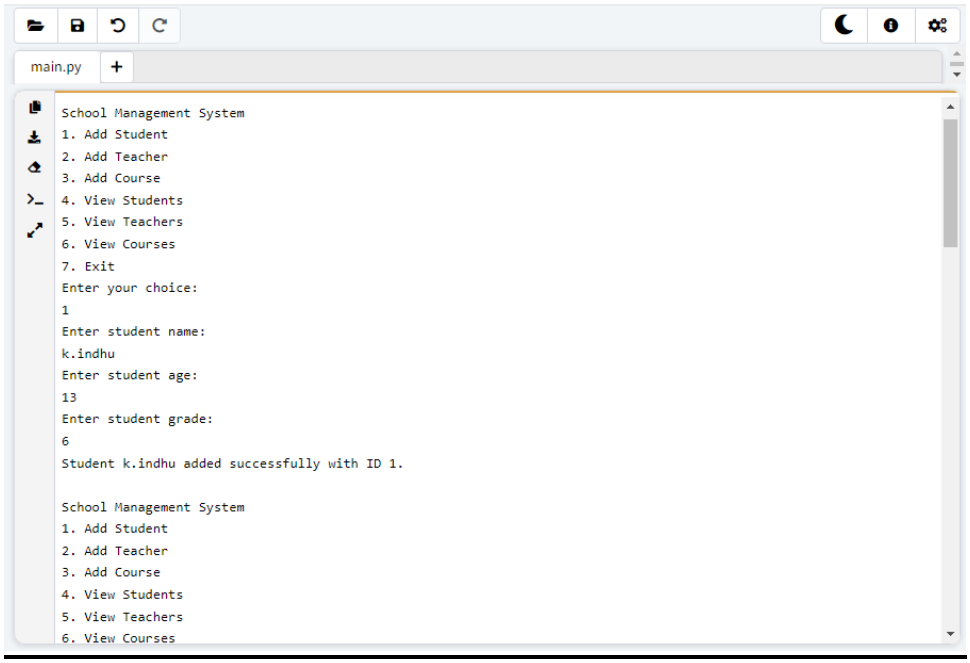
6. View Courses

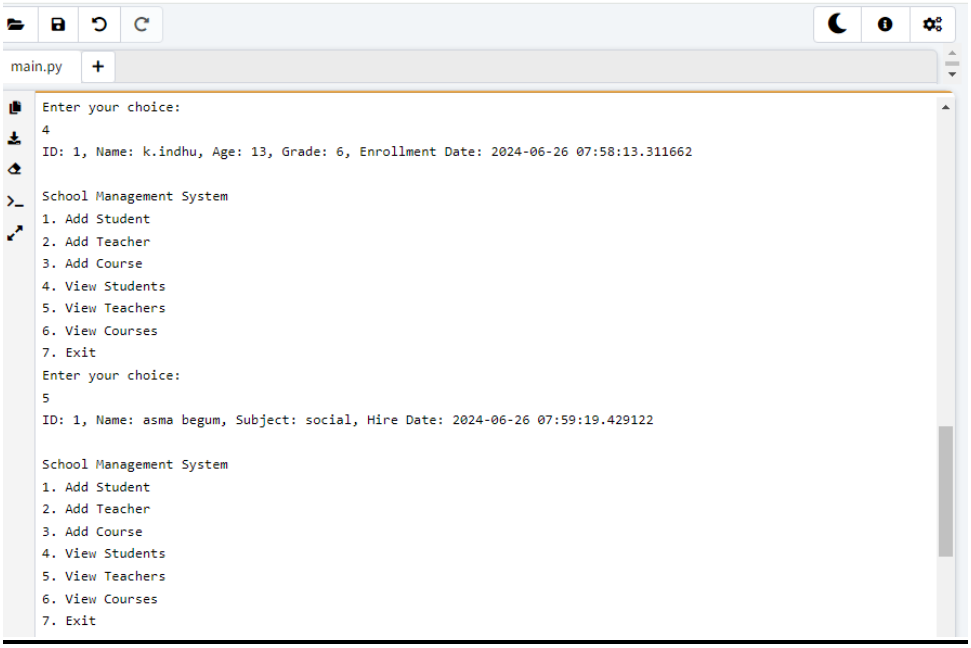
7. Exit

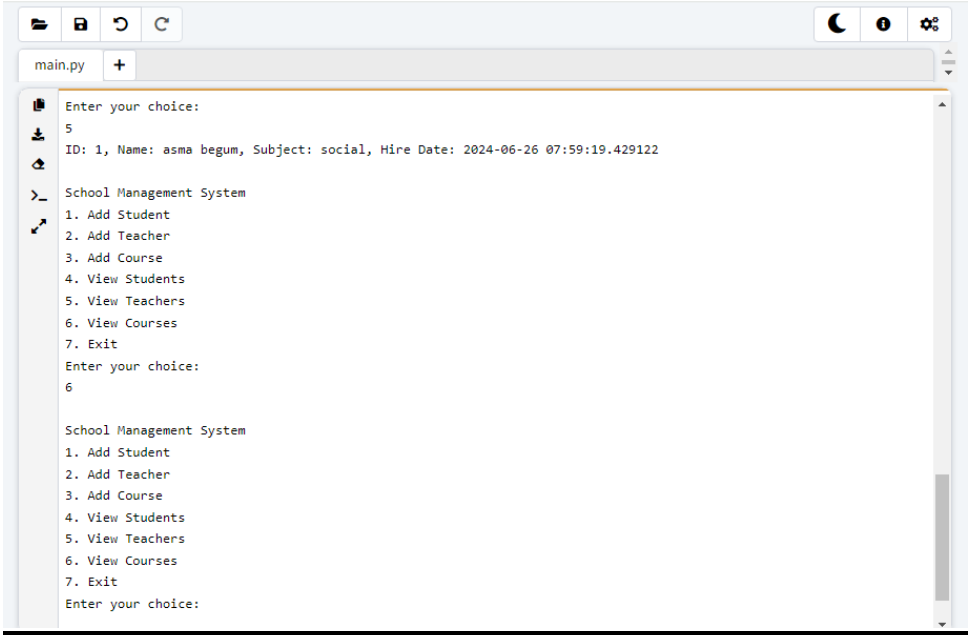
Enter your choice: 6

ID: 1, Course Name: Algebra, Teacher: Mr. Smith, Creation Date: 2024-06-18 12:00:00.000000

Output of the project




main.py
Enter your choice:
4
ID: 1, Name: k.indhu, Age: 13, Grade: 6, Enrollment Date: 2024-06-26 07:58:13.311662
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit
Enter your choice:
5
ID: 1, Name: asma begum, Subject: social, Hire Date: 2024-06-26 07:59:19.429122
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit


main.py
Enter your choice:
5
ID: 1, Name: asma begum, Subject: social, Hire Date: 2024-06-26 07:59:19.429122
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit
Enter your choice:
6
School Management System
1. Add Student
2. Add Teacher
3. Add Course
4. View Students
5. View Teachers
6. View Courses
7. Exit
Enter your choice:

CHAPTER-8

Conclusion

The School Management System project provides an effective solution for managing students, teachers, and courses within a school. This Python-based console application allows users to add students and teachers, createcourses, and view detailed records. By offering a simple and efficient interface, the system enhances the organization and management of school-related data, making it an invaluable tool for educational institutions.

The School Management System (SMS) project represents a significant step towards modernizing educational administration and enhancing the overall efficiency of educational institutions. By leveraging advanced technology and integrating comprehensive functionalities into a unified platform, the SMS addresses critical challenges faced by schools in managing administrative tasks, communication, and data management.

In conclusion, the SMS not only streamlines administrative workflows but also empowers educational institutions to focus more on educational excellence and student development. By embracing technology and innovation, the SMS project contributes to creating a more efficient, transparent, and collaborative educational ecosystem, ultimately benefiting students, educators, parents, and administrators alike.

As the SMS continues to evolve and adapt to the changing needs of educational institutions, ongoing support, maintenance, and enhancement will be essential to sustain its impact and effectiveness in enhancing educational outcomes and institutional success.

Throughout the development and implementation phases, the project has focused on several key objectives:

Automation of Administrative Processes: The SMS automates routine tasks such as student enrollment, attendance management, and grade tracking, reducing manual effort and improving accuracy.

Integration and Centralization of Data: By centralizing student records, academic schedules, and financial data, the SMS ensures data consistency and accessibility across different departments and stakeholders

CHAPTER-9

References

1. python.org

2. Code Academy

3. tutorialsPoint.com

4. PythonChallenge com

5. Google's Python Class

6. LearnPython.org

7. layak.in