ATTENDANCE MANAGEMENT SYSTEM

A MINI PROJECT REPORT

Submitted by

MITHILESH T   **220701165**

KIRUTHIKROKAN B   **220701133**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE (AUTONOMOUS)

THANDALAM
CHENNAI-
602105

2024-2025

BONAFIDE CERTIFICATE

Certified that this project report "**ATTENDANCE MANAGEMENT SYSTEM**" is the bonafide work of "**MITHILESH T (220701165)**, **KIRUTHIKROKAN B (220701133)"**who carried out the project under my supervision.

Submitted for the Practical Examination held on _____

SIGNATURE   SIGNATURE
Dr.R.SABITHA   **Ms.D.KALPANA**
Professor and II Year Academic Head   **Assistant Professor (SG),**
Computer Science and Engineering,   **Computer Science and Engineering,**
Rajalakshmi Engineering College   **Rajalakshmi Engineering College**
(Autonomous),   **(Autonomous),**
Thandalam, Chennai - 602 105   **Thandalam, Chennai - 602 105**

INTERNAL EXAMINER   **EXTERNAL EXAMINER**

ABSTRACT

The Attendance Management System (AMS) represents a paradigm shift in the way educational institutions and workplaces handle attendance tracking. Through a seamless integration of secure user authentication, intuitive interfaces, and robust data management capabilities, the AMS promises to revolutionize the entire attendance recording and analysis process. Users, including administrators, instructors, and participants, are empowered with the tools they need to efficiently record attendance for classes, meetings, or events, complete with accurate timestamps for enhanced precision and reliability.Central to the AMS is its centralized data storage architecture, which facilitates easy retrieval and analysis of attendance records. This centralized repository ensures that administrators have quick access to comprehensive attendance data, empowering them with valuable insights into attendance trends, patterns, and exceptions.

By automating attendance management processes, the AMS not only reduces administrative overhead but also enhances accountability and fosters a culture of data-driven decision-making. With the AMS in place, educational institutions and organizations can optimize their attendance tracking efforts, thereby improving overall efficiency and productivity.

TABLE OF CONTENTS

CHAPTER 1

1.1 INTRODUCTION

The Attendance Management System is critical in educational institutions and workplaces for accountability and resource optimization. Manual tracking methods are prone to errors and inefficiencies. The introduction of an Automated Attendance Management System (AMS) offers a solution. Leveraging modern technologies, the AMS automates attendance recording, storage, and analysis. Through intuitive interfaces and robust data management, it promises streamlined tracking, improved accuracy, and valuable insights into attendance patterns. This introduction sets the stage for exploring the AMS's design and implementation, emphasizing its significance in enhancing efficiency and accountability in diverse organizational settings.

1.2 OBJECTIVES

The Attendance Management System (AMS) aims to modernize attendance tracking by automating recording processes, enhancing accuracy, and saving time. It prioritizes secure user authentication for data integrity and confidentiality. Centralized data storage enables easy retrieval and analysis, empowering decision-makers with valuable insights. Intuitive interfaces cater to administrators, instructors, and participants, ensuring a seamless user experience. Comprehensive reporting tools identify attendance trends, aiding in informed decision-making. The system's scalability and flexibility allow adaptation to diverse organizational needs. Leveraging modern technologies, the AMS promises reliability, performance, and compatibility across platforms. Ultimately, it strives to optimize efficiency and accountability in educational and organizational settings.

1.3 MODULES

- Registration Module.
- Record Management Module.
- Percentage Management Module.

CHAPTER-2

2.1 SOFTWARE DESCRIPTION

Visual studio Code:

Visual Studio Code combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

2.2 LANGUAGES

1. Python:

- It is used for scripting the application's logic, managing database operations, and integrating different modules.

2.Streamlit:

- A powerful Python library for creating interactive web applications with simple Python scripts, enabling rapid prototyping and deployment of data-driven applications with ease..

3.MongoDB:

- A scalable NoSQL database solution, offering high performance, flexibility, and seamless integration with modern applications..

CHAPTER-3

REQUIREMENT AND ANALYSIS

3.1 REQUIREMENT SPECIFICATION:

The Attendance Management System (AMS) must encompass robust functional and non-functional requirements. It should allow secure user authentication, facilitate efficient attendance recording with timestamps, manage data securely in a centralized database, generate comprehensive reports and analytics, send automated notifications, and provide intuitive user interfaces. Non-functionally, it should prioritize security with encryption and access controls, ensure high performance and scalability, maintain reliability with backup mechanisms, prioritize usability with a user-centric design, and ensure compatibility across various platforms and systems. These requirements collectively aim to deliver a seamless, secure, and efficient attendance tracking solution for educational institutions and workplaces.

## 3.2 HARDWARE AND SOFTWARE REQUIREMENTS:

Hardware Requirements

- Processor: 1 GHz or faster processor

- RAM: 2 GB or more

- Storage: At least 500 MB of available disk space

- Display: Minimum resolution of 1024x768

- Input Devices: Keyboard and mouse

Software Requirements

- Operating System: Windows 7 or later, macOS, or Linux

- Python: Version 3.6 or higher

- SQLite: Version 3 or higher

- Python Libraries:

  - 'pymongo' connecting to and interacting with MongoDB (use command pip install pymongo)

  - 'Streamlit' building the web application interface (use command pip install streamlit)

3.3 ARCHITECTURE DIAGRAM:
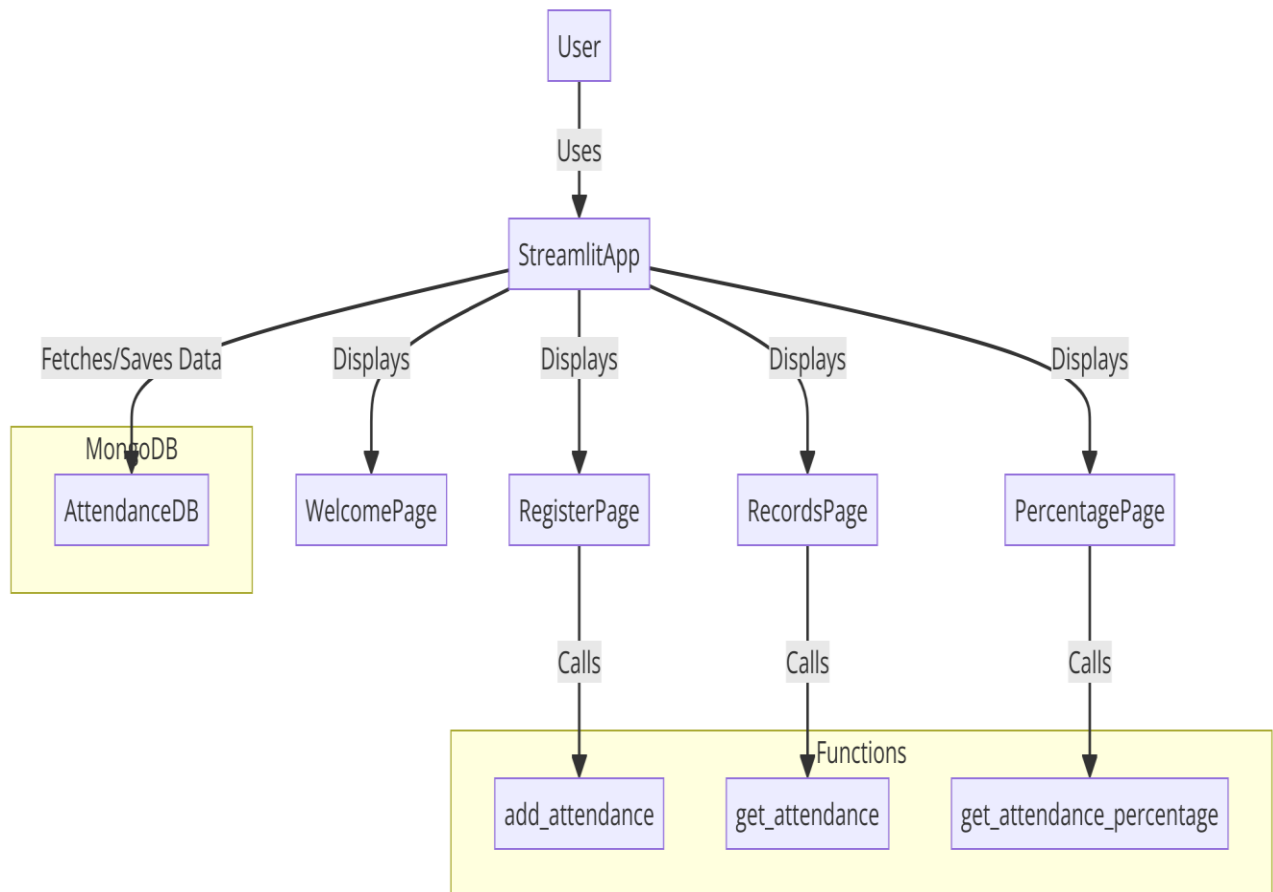
3.4 ER DIAGRAM:

CHAPTER-4

PROGRAM CODE

```python
import streamlit as st
from pymongo import MongoClient
from datetime import datetime

client = MongoClient("mongodb://localhost:27017/")
db = client['attendance_db']
collection = db['attendance']

def add_attendance(name, date, status):
    record = {
        "name": name,
        "date": date.strftime('%Y-%m-%d'),
        "status": status
    }
    collection.insert_one(record)


def get_attendance():
    records = list(collection.find({}, {"_id": 0}))
    if not records:
        st.write("No records found in the database.")
```

```python
    else:
        return records


def get_attendance_percentage(name):
    total_records = collection.count_documents({"name": name})
    present_records = collection.count_documents({"name": name,
"status": "Present"})
    if total_records == 0:
        return 0
    else:
        return (present_records / total_records) * 100


def welcome():
    st.title('Welcome to Attendance Management System')
    st.write('Please register your attendance or view records.')


def register():
    st.title('Register Attendance')
    with st.form(key='attendance_form'):
        name = st.text_input('Name')
        date = st.date_input('Date', datetime.now())
        status = st.selectbox('Status', ['Present', 'Absent'])
        submit_button = st.form_submit_button(label='Submit')
```

```python
    if submit_button:
        add_attendance(name, date, status)
        st.success(f'Attendance for {name} on {date.strftime("%Y-%m-%d")} marked as {status}')


def view_records():
    st.title('Attendance Records')
    records = get_attendance()
    if not records:
        st.write('No records found.')
    else:
        for record in records:
            st.write(f"Name: {record['name']}, Date: {record['date']}, Status: {record['status']}")


def view_attendance_percentage():
    st.title('Attendance Percentage')
    names = list(set(record['name'] for record in get_attendance()))
    for name in names:
        percentage = get_attendance_percentage(name)
        st.write(f"Name: {name}, Attendance Percentage: {percentage:.2f}%")
```

```python
pages = ['Welcome', 'Register', 'View Records', 'View Attendance
Percentage']
selection = st.sidebar.radio('Go to', pages)



if selection == 'Welcome':
    welcome()
elif selection == 'Register':
    register()
elif selection == 'View Records':
    view_records()
elif selection == 'View Attendance Percentage':
    view_attendance_percentage()


client.close()
```

CHAPTER-5

RESULTS AND DISCUSSION

5.1 USER DOCUMENTATION:

REGISTERATION MODULE:

RECORD MANAGEMENT MODULE:

Attendance Records

Name: kiruthikrokan, Date: 2024-05-27, Status: Present

SALES MANAGEMENT MODULE:

Attendance Percentage

Name: kiruthikrokan, Attendance Percentage: 100.00%

CHAPTER-6

6.1 CONCLUSION:

After completing the Attendance Management System project, we are confident that it effectively addresses the challenges present. This computerized system is designed to reduce human errors and significantly enhance operational efficiency. The primary goal of this project was to minimize human effort by automating routine tasks and streamlining the management processes.For instance, users can simply type a search string to quickly find specific records, and editing records is simplified through easy-to-use update functions.

Overall, the Attendance Management System achieves its primary objective of providing accurate and efficient management of attendance records. By automating data management and simplifying navigation and editing, the system ensures that records can focus on more strategic tasks. This project demonstrates the practical application of database management systems (DBMS) in addressing real-world challenges, particularly in attendance management.

CHAPTER-7

7.1 REFERENCES:

1. Python Documentation: Python Software Foundation. (n.d.). Python Documentation. Retrieved from https://docs.python.org/3/

2. Tkinter Documentation: TkDocs. (n.d.). Tkinter Tutorial. Retrieved from https://tkdocs.com/tutorial/

3. SQLite Documentation: SQLite. (n.d.). SQLite Documentation. Retrieved from https://www.sqlite.org/docs.html

4. Automate the Boring Stuff with Python: Sweigart, A. (2015). Automate the Boring Stuff with Python: Practical Programming for Total Beginners. No Starch Press.

5. Python GUI Programming with Tkinter: Grayson, J. E. (2000). Python and Tkinter Programming. Manning Publications.

6. SQLite Database Programming: Owens, M. (2006). The Definitive Guide to SQLite. Apress.

7. Database System Concepts: Silberschatz, A., Korth, H. F., & Sudarshan, S. (2010). Database System Concepts (6th ed.). McGraw-Hill.

8. GeeksforGeeks: Various authors. (n.d.). GeeksforGeeks. Retrieved from https://www.geeksforgeeks.org/

9. W3Schools: W3Schools. (n.d.). SQL Tutorial. Retrieved from https://www.w3schools.com/sql/