ATTENDANCE MANAGEMENT SYSTEM

A Project Report submitted in partial fulfillment of the requirement for the award of

Bachelor of

Computer Applications

of Bharathiar University, Coimbatore-46.

By

MUHAMMED SWALIH K H

Reg. No: 2122J0590

Under the guidance of

Mr. B. RAMESH KUMAR M.Sc Cs.,M.Phil.,B,Ed.,

Assistant professor,

Department of Computer Science

SREE NARAYANA GURU COLLEGE

K.G.CHAVADI, COIMBATORE –641105.

(Affiliated to Bharathiar University)

(Re-Accredited By NAAC & An ISO 9001-2015 Certified Institution)

March-2024

ATTENDANCE MANAGEMENT SYSTEM

Bonafide Work Done by

MUHAMMED SWALIH K H

Reg. No: 2122J0590

Under the guidance of

Mr. B. RAMESH KUMAR M.Sc Cs.,M.Phil.,B,Ed.,

Assistant professor,

Department of Computer Science

A project report submitted in partial fulfillment of the requirements for the award of

Bachelor of

Computer Applications

of Bharathiar University

SREE NARAYANA GURU COLLEGE

(RE-ACCREDITED BY NAAC& AN ISO 9001-2015 CERTIFIED INSTITUTION)

(Affiliated to Bharathiar University)

K.G.Chavadi, Coimbatore-641105

March -2024

ATTENDANCE MANAGEMENT SYSTEM

Bonafide Work Done by

MUHAMMED SWALIH K H

Reg. No: 2122J0590

Under the guidance of

Mr. B. RAMESH KUMAR M.Sc Cs.,M.Phil.,B,Ed.,

Assistant professor,

Department of Computer Science

A project report submitted in partial fulfillment of the requirements for the award of

Bachelor of

Computer Applications

of Bharathiar University, Coimbatore-105.

Guide **HOD**

Submitted for the viva voice examination held on …….

Internal Examiner **External Examiner**

DECLARATION

I, **MUHAMMED SWALIH K H** here by declaring that the project report entitles as **"ATTENDANCE MANAGEMENT SYSTEM"** done as the partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science is an independent project report done by me during the project duration of one period of study in **Sree Narayana Guru College**, K.G. Chavadi, Coimbatore under the guidance of **Mr. B. Ramesh Kumar M.Sc Cs.,M.Phil.,B,Ed.,**Assistant professor of Department of Computer Science.

Signature of the Guide **Signature of the Student**

MUHAMMED SWALIH K H

AKNOWLEDGEMENT

ACKNOWLEDGEMENT

First and foremost, I express my sincere thanks to our beloved and respected Principal **Dr. Kalpana D Ph.D.,** Sree Narayana Guru College , K.G Chavadi ,Coimbatore.

I would like to take this opportunity to express my gratitude to **Dr. R. PRIYA MCA.,M.Phill., Ph.D.,**Associate Professor & Head, Department of Computer science, Sree Narayana Guru College, Permitted me todo the project work.

I would like to this opportunity to express my sincere thanks to my project guide **Mr. B. Ramesh Kumar M.Sc Cs.,M.Phil.,B,Ed.,** Assistant Professor, Department of Computer science, Sree Narayana Guru College, for valuable guidance, encouragement and helping me to do the project work .

I would like to express my heartfelt gratitude to all the staff members of our department, Sree NarayanaGuru College, K.G Chavadi, Coimbatore.

I would like to express my special thanks to my family and my friends who gave their assistance at each and every stage of this project.

CONTENTS

CONTENTS

SYNOPSIS

SYNOPSIS

The Attendance Management System presents a comprehensive solution for organizations seeking to streamline attendance tracking processes efficiently. By integrating biometric technology, web, and mobile access, alongside automated reporting and leave management functionalities, the system ensures accurate and real-time attendance monitoring. Its implementation strategy, including requirements analysis, system design, development, deployment, and training, guarantees seamless adoption and utilization. With benefits ranging from time and cost savings to improved employee satisfaction and data-driven insights, the system empowers organizations to optimize workforce management, enhance productivity, and achieve strategic goals effectively.

1. INTRODUCTION

Attendance Management System is aimed at developing an online attendance management for a college. In most educational institutions the attendance is taken manually. It is not only time consuming, but it is also unsecure and unreliable and it can be lost. To overcome these problems, I have developed a better system which is Web based; it is fully responsive where a user can use in mobile, tablets and different computer systems. In this system records are kept safe and secure and the attendance information of particular or all students of particular class can be accessed easily and without time consuming, the report is generated automatically. This system has two modules. They are

- Student
- Teacher
- Administrator

Students can view their attendance by logging in with the details provided to them. Teachers can mark attendance, edit them. Administrator is the one who add users to the applications ie, students and teachers and can add departments, classes, course to the system and assign teachers to each class.

1.1 ORGANIZATION PROFILE

Sree Naryana Guru College

With a view of imparting quality education for the aspiring youth of Coimbatore, Sree Narayana Guru Educational Trust members came forward with a plan to start a college for Applied Sciences and Management Studies. Sree Narayana Guru College (SNGC) came into existence in 1994, Affiliated to Bharathiar University, Accredited by NAAC and Approved by Govt.ofTamilnadu. 1 SreeNarayana Guru College strives to emerge as a premiere institute of international standards promoting excellence and equity in higher education. To mould the students of the institution through a socially committed, intellectually inclined, value based and culture driven paradigms of learning using the state-of-the-art educational technologies. • To enhance the learning

capacity and knowledge of the students by imparting quality education of national and international standards. • To make the students think critically, objectively, creatively and to be life-long learners, engaged leaders and productive citizens. • To motivate the students to pursue research, to advance knowledge and to address the state, national and global challenges. • To mould their character and develop a value system so as to manifest oneness among students of diverse socio-cultural and economic backgrounds. • To identify the inherent talents of the students and provide a platform to exhibit them

1.2 SYSTEM SPECIFICATION

1.2.1 HARDWARE SPECIFICATION

- Processor : i3
- Ram : 4GB
- Storage : 40GB
- Speed : 3.5GHz
- Monitor : 20” Colour monitor
- Keyboard : Standard 104 keyboard
- Mouse : 3 button

1.2.2 SOFTWARE SPECIFICATION

- Operating system: Windows 7 and Above
- Front End : HTML,CSS
- Back End : Python
- Database : SQLite
- Browser : Internet Explorer 6.0, Google chrome
- Other Requirement: Microsoft Word 2010, Visual Studio Code

SOFTWARE FEATURES

A general set of engineering characteristics for a programming language can be established. They include ease of translation from design to code, computer efficiency, source code portability, availability of development tools and maintainability. The art of choosing a language is to start with the problem, decide what it requires and their relative importance since it will probably be impossible to satisfy all of them equally. Available languages should be measured against a lot of requirements.

The criteria that are applied during an evaluation of available languages are:

- General Application Area
- Environment in which software will execute
- Performance consideration
- Data structure complexity

All the above requirements are considered while choosing the programming language for the project inventory control system.

PYTHON

Python is a high-level, interpreted programming language that is widely used for developing a variety of applications, from web development to data analysis, machine learning, and scientific computing. It was created by Guido van Rossum in the late 1980s and named after the British comedy group Monty Python.

One of the key features of Python is its readability and simplicity, which makes it an ideal language for beginners to learn. It uses a syntax that is easy to understand and requires fewer lines of code than other languages, such as Java or C++. Additionally, Python has a large standard library and a wide range of third-party modules that can be easily installed using package managers like pip.

Python is an interpreted language, which means it does not require compilation before execution. It is instead compiled on the fly during runtime, allowing for speedier development and debugging. It also supports a variety of programming paradigms, such as procedural, functional, and object-oriented programming. Python is widely used in web

development frameworks such as Django and Flask, data analysis and visualization libraries such as pandas and Matplotlib, scientific computing libraries such as NumPy and SciPy, and machine learning libraries such as TensorFlow and PyTorch.

Features of python

- Python has a straightforward syntax that makes it simple to read and write, making it simple to learn. Its code is easy for beginners to grasp since it resembles the English language.
- Cross-platform: Python code is compatible with Windows, Mac, and Linux operating systems. Because of this, it's a fantastic option for creating cross-platform applications.
- Large standard library: Python has a sizable standard library that contains modules for many different activities, including string processing, regular expressions, and other things.
- Python is dynamically typed, so you don't have to specify the types of your variables before using them. This makes writing and reading code simple.
- Python executes code line-by-line without the requirement for compilation because it is an interpreted language. As a result, writing and testing code is expedited.
- Python supports object-oriented programming, which enables programmers to divide their programmes into modular, reusable parts.
- High-level: Python is a high-level language, which abstracts away many low-level aspects such as memory management and enables developers to concentrate on resolving issues at a higher degree of abstraction.
- Strong community: Python has a sizable and engaged developer community that contributes to open-source projects, libraries, and frameworks, making it simple to locate support and materials when needed.
- Scalable: Python has the ability to manage applications and workloads of all sizes. Some of the biggest businesses in the world, like Google, Dropbox, and Instagram, use it.

Django

A high-level Python web framework called Django was created to make it easier and faster for programmers to create scalable, secure, and maintainable web applications. It adheres to the Model-View-Controller (MVC) architectural pattern and places a strong emphasis on the DRY principle.

For typical web development activities like handling user authentication, maintaining databases, handling forms, and creating HTML templates, Django offers a potent collection of tools and libraries. Additionally, it contains an integrated Object-Relational Mapping (ORM) framework that enables programmers to interact with databases using Python code rather than SQL.

Features

- Django has an integrated admin interface that enables developers to easily manage site content, users, and permissions.
- The Django URL routing system makes it simple to create sophisticated web applications with numerous pages and routes by allowing developers to map URLs to views and functions.
- Security: Django contains a number of built-in security measures, including defence against XSS, SQL injection, and Cross-Site Request Forgery (CSRF) threats.
- Template engine: Using Python code, developers can construct dynamic HTML templates with Django's template engine.
- Scalability: Django was created with scalability in mind and is capable of handling high volumes of traffic and data.

SQLite

SQLite is a relational database management system (RDBMS) that is lightweight, open-source, and commonly used in embedded systems and small to medium-scale applications. It is a serverless database engine, which means it does not require a separate server process to execute and stores the entire database in a single file on disc.

SQLite is intended to be simple, fast, and dependable. Most typical SQL operations, such as SELECT, INSERT, UPDATE, and DELETE, are supported, as well as transactions, views,

triggers, and indexes. It's also ACID-compliant, which protects data integrity and consistency even when the system fails or crashes.

SQLite's tiny size and low memory footprint are essential properties, making it excellent for usage in embedded systems, mobile devices, and other resource-constrained contexts. It's also quite portable and works on a variety of platforms, including Windows, Linux, macOS, and mobile operating systems like Android and iOS.

SQLite is widely used in a wide range of applications, including web browsers (for example, Firefox and Chrome), operating systems (for example, Android, iOS, and macOS), mobile apps, desktop programs, and embedded systems. It is also a popular choice for developing and testing applications due to its ease of use and lack of additional software or configuration.

Features of SQLite

- Zero Configuration: Unlike other conventional database engines, SQLite does not need any configuration files or a separate server process.
- Cross-Platform Support: SQLite is compatible with practically all operating systems, including Windows, macOS, Linux, iOS, and Android.
- Full support for SQL, including the SELECT, INSERT, UPDATE, and DELETE operations, is offered by SQLite.
- Atomicity, consistency, isolation, and durability of transactions are all ensured by SQLite's compliance with ACID.
- Small Code Footprint: With a code footprint of less than 500KB, SQLite is exceptionally lightweight, making it the perfect choice for embedded systems, mobile devices, and other resource-constrained environments.
- SQLite includes transactions and rollback mechanisms to guarantee the consistency and integrity of data.
- Full-Text Search: You can search for text-based content contained in the database using SQLite's full-text search features.
- Several Connections: SQLite is perfect for concurrent applications since it allows several connections to access the same database at once.
- Support for Triggers: SQLite has support for triggers, a potent tool for triggering the execution of custom code in response to specific database events.

- Indexing: To improve the speed of data retrieval and query performance, SQLite offers a variety of indexing algorithms, including B-tree and R-tree indexes.
- Backup and recovery functions are supported by SQLite, making it simple to generate and restore database backups.
- SQLite is offered with a public domain licence, which allows you to use it in any project, whether for profit or not, without paying any licencing costs or royalties.

Visual Studio Code

Microsoft's Visual Studio Code (VS Code) is a free, open-source code editor. It is a lightweight, cross-platform editor that supports a variety of programming languages and technologies, including JavaScript, Python, C++, and many others. VS Code is intended to be extremely flexible and extendable, with support for hundreds of extensions and plugins from the Visual Studio Code Marketplace. It has an integrated debugger, intelligent code completion, code refactoring tools, Git integration, and a robust command-line interface.

The integrated terminal in VS Code is a critical feature that allows developers to run commands and scripts without leaving the editor. It also has a task runner and support for numerous terminals, making complicated operations simple to manage and automate.

Another noteworthy feature of VS Code is its live sharing function, which allows developers to interact in real-time with other developers working on the same project regardless of their location.

Because of its ease of use, versatility, and strong features, VS Code has become one of the most popular code editors among developers. It is available for Windows, Linux, and macOS.

2. SYSTEM STUDY

2.1 EXISTING SYSTEM

The Existing system is a manual entry for the students. Here the attendance will be carried out in the hand written registers. It will be a tedious job to maintain the record for the user. The human effort is more here. The retrieval of the information is not as easy as the records are maintained in the hand written registers. This application requires correct feed on input

into the respective field. Suppose the wrong inputs are entered, the application resist to work. so the user find it difficult to use.

DRAWBACKS

The Existing system offer many benefits, but they also come with certain drawbacks that organizations should consider:

- Initial Setup Costs: Implementing an AMS can be expensive, involving costs for software licensing, hardware, installation, and customization. Small businesses or organizations with limited budgets may find these costs prohibitive.
- Technical Issues: Like any software system, AMS can encounter technical problems such as software bugs, compatibility issues, or server downtime. These issues can disrupt attendance tracking and affect overall system reliability.
- Dependency on Technology: AMS relies on technology such as biometric scanners, RFID tags, or mobile applications. Any failure or malfunction in these technologies can lead to inaccurate attendance records or system downtime.
- Data Security Concerns: Attendance data is sensitive information that needs to be protected from unauthorized access, manipulation, or theft. Security breaches can compromise employee privacy and expose the organization to legal and reputational risks.
- User Resistance: Employees may resist using the new system due to concerns about privacy, surveillance, or changes in established routines. Resistance can affect the accuracy and reliability of attendance data and hinder system adoption.
- Training Requirements: Introducing a new AMS requires training employees on how to use the system effectively. Training programs can be time-consuming and costly, especially if employees are not tech-savvy or resistant to change.
- Integration Challenges: Integrating the AMS with existing HR, payroll, or other systems can be complex, especially if they use different technologies or data formats. Integration challenges can delay implementation and increase costs.
- Scalability Issues: As organizations grow or change, the AMS needs to accommodate an increasing number of users, locations, or attendance tracking requirements. Scalability issues can arise if the system cannot handle the growing workload efficiently.

2.2 PROPOSED SYSTEM

To overcome the drawbacks of the existing system, the proposed system has been evolved. This project aims to reduce the paper work and saving time to generate accurate results from the student's attendance. The system provides with the best user interface. The efficient reports can be generated by using this proposed system.

FEATURES OF NEW SYSTEM

The aim of the proposed system is to address the limitations of the current system. The requirements for the system have been gathered from the defects recorded in the past and also based on the feedback from users of previous metrics tools. Following are the advantages of proposed system :

1. It is trouble-free to use.
2. It is a relatively fast approach to enter attendance
3. Is highly reliable, approximate result from user
4. Best user Interface
5. Efficient reports

FEASIBILITY STUDY

Feasibility analysis begins once the goals are defined. It starts by generating broad possible solutions, which are possible to give an indication of what the new system should look like. This is where creativity and imagination are used. Analysts must think up new ways of doing things- generate new ideas. There is no need to go into the detailed system operation yet. The solution should provide enough information to make reasonable estimates about project cost and give users an indication of how the new system will fit into the organization. It is important not to exert considerable effort at this stage only to find out that the project is not worthwhile or that there is a need significantly change the original goal. Feasibility of a new system means ensuring that the new system, which we are going to implement, is efficient and affordable. There are various types of feasibility to be determined. They are,

Technical Feasibility

Technical feasibility centres on the existing manual system of the test management process and to what extent it can support the system. According to feasibility analysis procedure the technical feasibility of the system is analysed and the technical requirements such as software facilities, procedure, inputs are identified. It is also one of the important phases of the system development activities. It is technically feasible, since the whole system is designed into the latest technologies like Python and SQLite which are the most recent technologies to develop web based systems and design databases.

The system offers greater levels of user friendliness combined with greater processing speed. Therefore, the cost of maintenance can be reduced. Since, processing speed is very high and the work is reduced in the maintenance point of view management convince that the project is operationally feasible.

Economic Feasibility

Economic analysis is most frequently used for evaluation of the effectiveness of the system. More commonly known as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions is made to design and implement the system.

This part of feasibility study gives the top management the economic justification for the new system. This is an important input to the management the management, because very often the top management does not like to get confounded by the various technicalities that bound to be associated with a project of this kind. A simple economic analysis that gives the actual comparison of costs and benefits is much more meaningful in such cases.

It is economically feasible, it will only require a single operator to operate the system, who is responsible for entering the data into the database via a user interface provided to him, who can also able to show all the data in html tabular form so to provide information

regarding the students who are either taken admission or to take admission, since it requires only a single person to operate the whole system thus reduces the cost to operate the system. In the system, the organization is most satisfied by economic feasibility. Because, if the organization implements this system, it need not require any additional hardware resources as well as it will be saving lot of time.

Operational Feasibility

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. Technical performance include issues such as determining whether the system can provide the right information for the Department personnel student details, and whether the system can be organized so that it always delivers this information at the right place and on time using internet services. Acceptance revolves around the current system and its personnel.

3. SYSTEM DEVELOPMENT

The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The first step in system designing is to determine how the output is to be produced and in what format. Samples of the output and input are also presented. In the second step, input data and master files are to be designed to meet requirement of the proposed output. The processing phases are handled through program construction and testing, including a list of the programs needed to meet the system's objectives and complete documentation.

3.1 INPUT DESIGN

Input design is a part of overall system design, which requires very careful attention.

The designer has a number of clear objectives in input design

- To produce a cost effective method of input.
- To achieve the highest possible level of accuracy.
- To ensure that the input is acceptable and is understood by the user. Input design is the link that ties information system into the world of its users. Input design is the process of converting user-oriented input to a computer-based format. It also includes determining the record media, method of input speed of capture and entry into the system. Input data are collected and organized into groups of similar data.

The design for handling input specifies how data are accepted for computer processing. Input design is a part of overall system design that needs careful attention. The collection of input data is considered to be the most expensive part of the system design. Since the inputs have to be planned in such a manner so as to get the relevant information extreme care is taken to obtain the information. The goal of designing input data is to make data entry as easy, logical and face from errors as possible

3.2 OUTPUT DESIGN

Outputs from computer systems are required primarily to communicate the result of processing to users. They are also used to provide a permanent copy of these results for later consultation. The output design is the process where the inputs are processed so as to get the information such that the concerned authorities can take the various decisions regarding the

various outputs generated. The approach to output design is very dependent on the type of output data. Once the output medium has been chosen, detailed specification of the output document can be carried

Computer output is the most important and direct source of information to the user. Efficient, intelligible output design should improve the systems relationships with the user and help in decision making. A major form of output is hard copy from the printer. Printouts should be designed around the output requirements of the user.

3.3DATABASE DESIGN

The database design for the attendance management system includes three tables: Students, Courses, and Attendance. The Students table contains fields such as student_id, student_name, and student_email to store information about enrolled students. The Courses table consists of fields like course_id, course_name, and course_code to manage course details. The Attendance table tracks attendance records with fields like attendance_id, student_id (linked to Students table), course_id (linked to Courses table), date, and status to record student attendance for each course session. This design establishes relationships between students and attendance, as well as between courses and attendance, enabling efficient management of attendance records within the system.

3.4SYSTEM DEVELEPMENT

3.4.1 DESCRIPTION OF MODULES

The modules involved in this project are:.

1. User Management Module:

This module is responsible for managing user accounts within the system. It includes functionalities such as user registration, login authentication, and profile management. Administrators can add, modify, and deactivate user accounts, while users can update their personal information and change passwords.

2. Attendance Tracking Module:

The core functionality of the AMS revolves around tracking attendance. This module allows users to record their attendance status, such as present, absent, or late. It includes features for marking attendance manually or automatically based on predefined rules, such as biometric scans or RFID tags. Additionally, this module provides real-time monitoring of attendance status and alerts for anomalies.

3. Reporting Module:

The reporting module enables users to generate various reports related to attendance data. It includes predefined report templates for common use cases, such as daily attendance reports, monthly summaries, and attendance trends analysis. Users can customize report parameters, such as date range and employee filters, to generate insights and track performance metrics.

4. Organization Structure Module:

This module manages the hierarchical structure of the organization, including departments, teams, and employee roles. Administrators can define organizational units and assign users to specific roles within the structure. It provides a framework for organizing attendance data and facilitates role-based access control to system functionalities.

5. Leave Management Module:

In addition to attendance tracking, the AMS includes a leave management module for managing employee leave requests. Users can submit leave applications, specify the type of leave (e.g., sick leave, vacation), and track the status of their

requests. Administrators can review and approve leave requests, ensuring smooth coordination of workforce scheduling.

6. Notifications Module:

The notifications module facilitates communication between users and administrators regarding attendance-related activities. It sends automated notifications for upcoming events, such as scheduled shifts or leave approvals. Users can also receive notifications for late attendance or unauthorized absences, promoting accountability and compliance with organizational policies.

7. Settings and Configuration Module:

This module provides system administrators with tools for configuring settings and parameters related to attendance management. It includes options for defining working hours, holiday calendars, and overtime rules. Administrators can customize the system's behavior to align with organizational policies and regulatory requirements.

8. Integration Module:

The integration module allows the AMS to integrate with external systems and devices for enhanced functionality. It supports integration with biometric devices, RFID scanners, payroll systems, and HR management software. This enables seamless data exchange and interoperability with existing infrastructure.

9. Security Module:

Security is paramount in the AMS to protect sensitive attendance data and ensure compliance with privacy regulations. The security module includes features such

as user authentication, access control, data encryption, and audit logging. It safeguards against unauthorized access, data breaches, and malicious activities.

10. Dashboard Module:

The dashboard module provides users with a centralized view of attendance-related information through interactive dashboards and visualizations. It displays key metrics, such as attendance rates, trends, and compliance levels, allowing users to monitor performance and make informed decisions.

4. TESTING AND IMPLEMENTATION

SYSTEM IMPLEMENTATION

Implementation is an activity that is contained throughout the development phase. It is a process of bringing a developed system into operational use and turning it over to the user. The new system and its components are to be tested in a structured and planned manner. A successful system should be delivered and users should have confidence that the system would work efficiently and effectively. The more complex the system being implemented the more involved will be the system analysis and design effort required for implementation. The major activities in implementation plan are cost estimation, schedule and milestone determination, project staffing, quality control plans, and controlling and monitoring plans The implementation plan involves the following:

- Testing to confirm effectiveness.
- Detection and correction of errors.

SYSTEM TESTING

Software testing is a critical element of the software development cycle. The testing is essential for ensuring the Quality of the software developed and represents the ultimate view of specification, design and code generation. Software testing is defined as the process by which one detects the defects in the software. Testing is a set of activities that work towards the integration of entire computer based system.

A good test case is one that has a high probability of finding an as-yet undiscovered error. A successful test is one such uncovers or finds such errors. If testing is conducted successfully, it will uncover errors in the software. It also demonstrates that software functions are being performed according to specifications and also behavioral and performance requirements are satisfied. For this, test plans have to be prepared. The implementation of a computer system requires that test data has to be prepared and that all the elements in the system are tested in a planned and efficient manner. Nothing is complete without testing, as it is vital success of the system.

Unit Testing

Unit testing is carried out screen-wise, each screen being identified as an object. Attention is diverted to individual modules, independently to one another to locate errors. This has enabled the detection of errors in coding and logic. This is the first level of testing. In this, codes are written such that from one module, we can move on to the next module according to the choice we enter.

Integration Testing

This testing strategies combines all the modules involved in the system. After the independent modules are tested, dependent modules that use the independent modules are tested. This sequence of testing layers of dependent modules continues until the entire system is constructed. Though each module individually, they should work after linking them together. Data may be lost across interface and one module can have adverse effect on another. Subroutines, after linking, may not do the desired function expected by the main routine. Integration testing is a systematic technique for constructing program structure while at the same time, conducting test to uncover errors associated with the interface. In the testing the programs are contructed and tested in the small segments.

Regression Testing

Each time a new module is added, the software changes. New IO may occur, new control logic is invoked. These changes may causes problems. Regression testing is the re -execution of some subset of tests that have already been conducted to ensure that changes have not propagated unintended side effects. Testing a system requires more effort while developing it because it is one of the final steps. Early planning for this stage can ensure smooth and easy testing. Adequate preparation needs to be made before testing begins, so that it can be

performed effectively. The aim of testing is to prove that the developed system addresses the predefined processing requirements and will perform reliably and efficiently when running live. In the Automation System, providing test data to check the working of the system as specified performs testing. The complete system is tested to the satisfaction of users.

Security Testing

Depending on the nature of the project, developers may need to be particularly vigilant in addressing security concerns. An Internet application that processes credit cards and stores customer information needs to be secure. It can't be mostly secure or reasonably secure. It needs to be 100 percent secure. Of course, an application that maintains a library of no confidential company information for all employees to view may not need a security test at all.

Requirements Testing

The purpose of the requirements test is to validate that the system meets all business requirements. This test is relatively straightforward, but it requires that you have done a good job documenting the requirements during the analysis phase. If you don't have written requirements, you have nothing to validate against. A good technique is to list the business requirements in the first column of a table. In the second column, describe how you will test that the requirement is satisfied. This could include the specific test case(s) used. The third column should include an indication that the test was completed and what the results were. Remember, you are validating that all features and functions work as they should. You are not trying to break the system. Just include one or more cases that test the business requirement and make sure that the results are as planned.

EDUCATION AND TRAINING

To achieve the objective and benefits expected from computer based, it is essential for the people who will be involved to be confident of their role in the new system. This involves them in understanding the overall system and its effect on the organization and in being able to carry out effectively their specific task. As systems become more complex, the need for education and training is more and more important.

Training requirements are easy to determine. They arise from the changes that the systems analyst is bringing about. User managers must inform of how the whole system works its objective, new documentation, files and procedures. These requirements will be

set out in principle in the user system specification. Thus the new system can be explained to the user management and new tasks specified in job specification, with the analysis on hand to answer and queries which may arise.

POST IMPLEMENTATION REVIEW

After the system is implemented and conversion is complete, a review should be conducted to determine whether the system is meaning expectation and where improvements are needed. A post implementation review measures the system performance against pre-defined requirements. It determines how well the system continues to meet performance specification. It also provides information to determine whether major redesign or modification is required.

A post- implementation review is an evaluation of a system in terms of the extent to which the system accomplishes stated objectives and actual project costs exceed initial estimates. It is usually a review of major problems that need converting and those that surfaced during the implementation phase.

The post implementation study begins with the review team, which gathers and reviewers requests for evaluation. Unexpected change in the system that affects the user or system performance is a primary factor that prompts system review. Once request is filed, the user is asked how well the system is functioning to specifications or how well the measured benefits have been realized. Suggestions regarding changes and improvements are also asked.

Review Plan

- The review team prepares a formal review plan around the objectives of the review, the type of evaluation to be carried out and the time schedule required. An overall plan covers the following areas:
- Administrative plan: Review area objectives, operating costs actual operating performance and benefits.
- Personal requirements plan: Review performance objectives and training performance to data.
- Hardware plan: Review performance specification.
- Documentation review plan: Review the system development effort. Approaches to software evaluation

For software evaluation following approaches have been discussed:

- Bench marking: Benchmark is nothing but a sample program specially required to evaluate the comparative performance of hardware or software.
- Experience of other users: Vendors generally give a list of users who are satisfied with their work. But it is advisable to seek the opinion independently from the existing users whose configuration and operational environment is closely identical.
- Report of independent research organization:

Now-a-days many research organizations undertake project of evaluating the proprietary software offered by various software agencies. These evaluations are objective and comprehensive in nature. They publish the report at regular interval. The prospective buyer of a software package can have faith in their evaluation.

5.CONCLUSION

In conclusion, the attendance management system presents a robust solution for efficiently tracking student attendance across various courses. By implementing a well-structured database design, the system can accurately record and manage attendance records, providing educators with valuable insights into student participation and engagement. The system's ability to establish relationships between students, courses, and attendance facilitates seamless data management and retrieval, enhancing overall administrative processes. Furthermore, leveraging Python as the frontend and SQLite as the backend ensures flexibility, scalability, and ease of integration with other systems. Overall, the attendance management system offers an effective tool for educational institutions to optimize attendance monitoring and improve overall academic performance.

BIBLOGRAPHY

WEB RESOURCES

- http://www.wikipedia.org
- https://www.tutorialspoint.com
- http://www.google.com
- https://www.geeksforgeeks.org

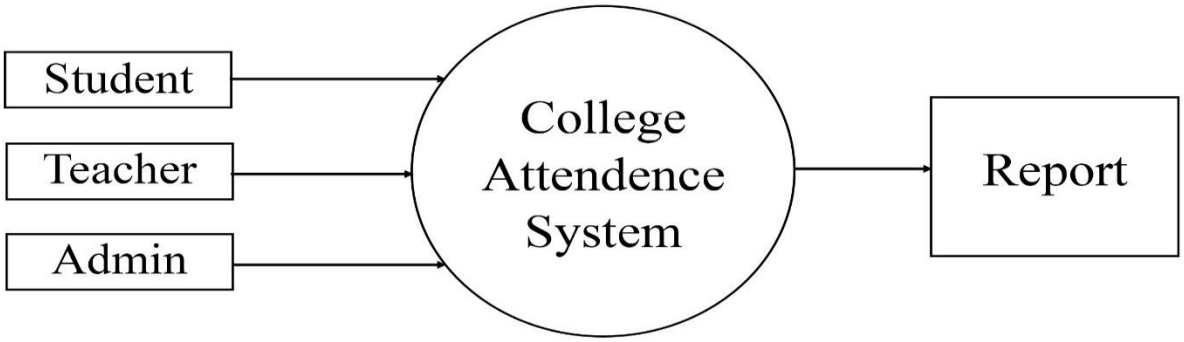
BOOK REFERENCES

- Microsoft SQL Server Black Book by Patrick Dalton, Prentic Hall Publications
- Software Engineering and Application by Rogger S. Pressman, TMH Publications

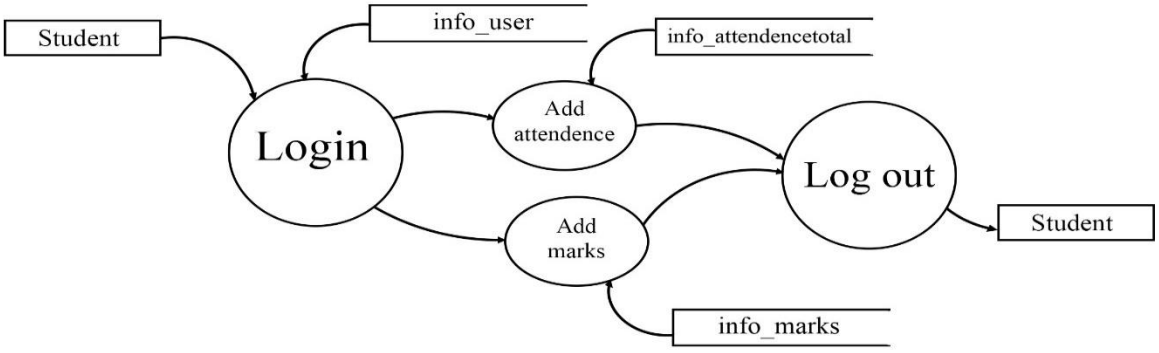
APPENDICS

A. DATA FLOW DIAGRAM

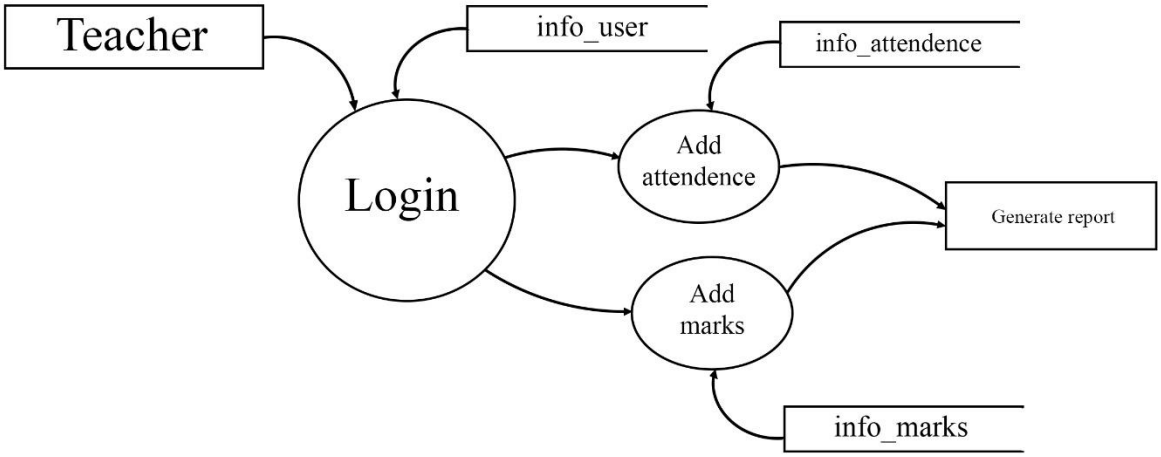
Level 0



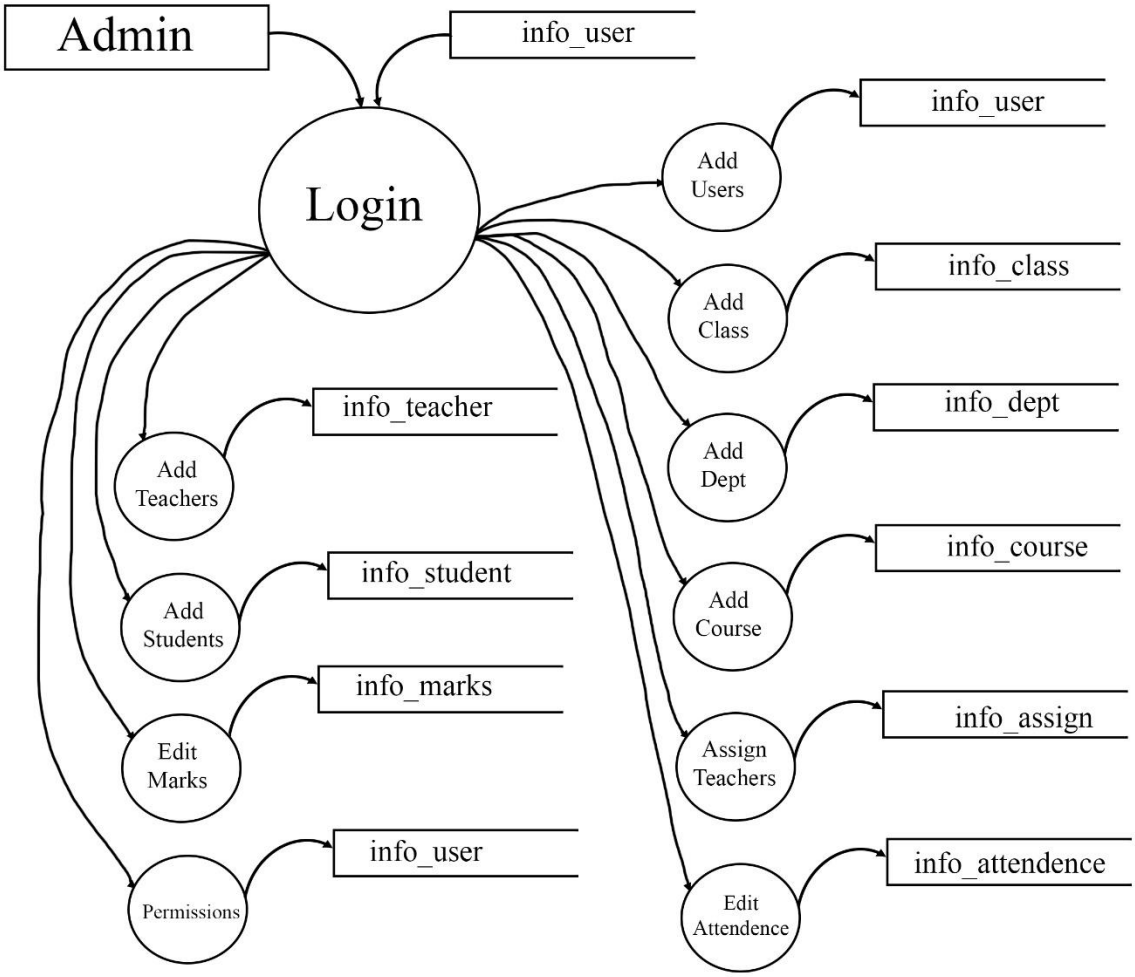
Level 1 - Student



Level 1 - Teacher



Level 1 - Admin



B. TABLE STRUCTURE

Table name : info_user

Primary key : id

Field name	Field type	Field description
id	integer	User id
password	varchar(128)	password
last_login	datetime	Last login date and time
is_superuser	bool	Is the user a superuser
username	varchar(150)	User name
last_name	varchar(150)	Last name of user
email	varchar(254)	email
is_staff	bool	Is a staff or not
is_active	bool	Is the user active or not
date_joined	datetime	Dateandtime when the user was added
first_name	varchar(150)	First name of user

Table name : info_teacher

Primary key : id

Field name	Field type	Field description
id	varchar(100)	User id
name	varchar(100)	Class id
sex	varchar(50)	Course id
DOB	datetime	Teacher id
user_id	integer	User id
dept_id	varchar(100)	Dept id

Table name : info_studentcourse

Primary key : id

Field name	Field type	Field description
id	integer	User id
course_id	varchar(50)	Course id
student_id	varchar(100)	Student id

Table name : info_student

Primary key : USN

Field name	Field type	Field description
USN	varchar(100)	User id
name	varchar(200)	name
sex	varchar(50)	gender
class_id_id	varchar(100)	Class id
user_id	integer	User id
DOB	datetime	Date of birth

Table name : info_attendence

Primary key : id

Field name	Field type	Field description
id	integer	User id
date	datetime	date
status	bool	status
attendenceclass_id	integer	Attendance class
course_id	varchar(50)	Course id
student_id	varchar(100)	Student id

Table name : info_assign

Primary key : id

Field name	Field type	Field description
id	integer	User id
class_id_id	varchar(100)	Course id
course_id	varchar(50)	Student id
teacher_id	varchar(100)	Teacher id

C. SAMPLE CODE

Login

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>Login</title>
     {% load static %}
    <!-- Bootstrap core CSS-->
    <link href="{% static '/info/bootstrap/vendor/bootstrap/css/bootstrap.min.css' %}" rel="stylesheet">

    <!-- Custom fonts for this template-->
```

```
    <link href="{% static '/info/bootstrap/vendor/fontawesome-free/css/all.min.css' %}" rel="stylesheet" type="text/css">

    <!-- Custom styles for this template-->
    <link href="{% static '/info/bootstrap/css/sb-admin.css' %}" rel="stylesheet">

  </head>

  <body class="bg-dark">

    <div class="container">
      <div class="card card-login mx-auto mt-5">
        <div class="card-header">Login</div>
        <div class="card-body">
          <form method="post">
            {% csrf_token %}
            {{ form.as_p }}
            <button  class="btn btn-success" type="submit">Login</button>
          </form>

        </div>
      </div>
    </div>

    <!-- Bootstrap core JavaScript-->
```

```
    <script src="{% static '/info/bootstrap/vendor/jquery/jquery.min.js' %}"></script>
    <script src="{% static '/info/bootstrap/vendor/bootstrap/js/bootstrap.bundle.min.js' %}"></script>

    <!-- Core plugin JavaScript-->
    <script src="{% static '/info/bootstrap/vendor/jquery-easing/jquery.easing.min.js' %}"></script>

  </body>
</html>
```

Attendence

```
{% extends 'info/base.html' %}
{% block content %}
{% if c.student_set.all %}
<form action="{% url 'confirm' assc.id %}" method="post">
          {% csrf_token %}
    <div class="card mb-3">
       <div class="card-header">
         <i class="fas fa-table"></i>
        <b>{{ dept1.name }}</b></div>
       <div class="card-body">
         <div class="table-responsive">
           <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
             <thead>
```

```
            <tr>
              <th>Student name</th>
              <th></th>
            </tr>
          </thead>
            <tbody>
            {% for s in c.student_set.all %}
              <tr>
              <td>{{s.name}}</td>
              <td>
                <div class="btn-group btn-group-toggle" data-toggle="buttons">
<label class="btn btn-outline-success active">
                    <input type="radio" name="{{ s.USN }}" id="option1"
autocomplete="off" value="present" checked> Present
                  </label>
<label class="btn btn-outline-danger">
                    <input type="radio" name="{{ s.USN }}" id="option2"
autocomplete="off" value="absent"> Absent
                  </label>
                </div>
              </td>
              </tr>
            {% endfor %}
            </tbody>
          </table>
```

```
        </div>
      </div>
    </div>
<input class="btn btn-success" type="submit" value="Submit">
</form>
{% else %}
  <p>No students in Class</p>
{% endif %}
{% endblock %}
```

Student hompage

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <meta name="description" content="">
  <meta name="author" content="">

  <title>homepage</title>
   {% load static %}

  <!-- Bootstrap core CSS -->
   <link href="{% static '/info/homepage/vendor/bootstrap/css/bootstrap.min.css' %}" rel="stylesheet">
```

```
    <!-- Custom styles for this template -->
     <link href="{% static '/info/homepage/css/heroic-features.css' %}" rel="stylesheet">
        <link href="{% static '/info/bootstrap/vendor/fontawesome-free/css/all.min.css' %}" rel="stylesheet" type="text/css">
  </head>
  <body>
    <!-- Navigation -->
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
     <div class="container">
        <a class="navbar-brand" href="{% url 'index' %}">College Management System</a>
       <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarResponsive" aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
       </button>
       <div class="collapse navbar-collapse" id="navbarResponsive">
        <ul class="navbar-nav ml-auto">
          <li class="nav-item">
            <a class="nav-link text-capitalize">{{ request.user.student.name }}</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#" data-toggle="modal" data-target="#logoutModal">Logout</a>
          </li>
        </ul>
       </div>
```

```
    </div>
  </nav>

  <!-- Page Content -->
  <div class="container">

    <!-- Jumbotron Header -->
    <header class="jumbotron my-4">
      <h1 class="display-3 text-capitalize">Welcome {{ request.user.student.name }}</h1>
{#      <p class="lead">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ipsa, ipsam, eligendi, in quo sunt possimus non incidunt odit vero aliquid similique quaerat nam nobis illo aspernatur vitae fugiat numquam repellat.</p>#}
{#      <a href="#" class="btn btn-primary btn-lg">Call to action!</a>#}
    </header>

    <!-- Page Features -->
    <div class="row text-center">

      <div class="col-lg-3 col-md-6 mb-4">
        <div class="card">
          <a href="{% url 'attendance' request.user.student.USN %}">
            <img class="card-img-top" src="{% static 'info/images/attendance2.jpg' %}" alt="">
          </a>
          <div class="card-body">
          <h4 class="card-title">Attendance</h4>
```

```
<p class="card-text">View the attendance status for each of your courses. The attendance of each course is
    also displayed as list of classes that were conducted.</p>
  </div>
  <div class="card-footer">
    <a class="btn btn-primary" role="button" href="{% url 'attendance' request.user.student.USN %}"> View Attendance</a>
  </div>
 </div>
</div>

<div class="col-lg-3 col-md-6 mb-4">
 <div class="card">
   <a href="{% url 'marks_list' request.user.student.USN %}">
    <img class="card-img-top" src="{% static 'info/images/marks.jpg' %}" alt="">
   </a>
   <div class="card-body">
   <h4 class="card-title">Marks</h4>
   <p class="card-text">View the marks obtained for each of your courses. These include the marks of 3
    internal assessment, 2 events and the Semester End Exam  </p>
  </div>
  <div class="card-footer">
    <a class="btn btn-primary" role="button" href="{% url 'marks_list' request.user.student.USN %}">View Marks</a>
  </div>
```

```
        </div>
      </div>
    </div>
    <!-- /.row -->


  </div>
  <!-- /.container -->
  <!-- Logout Modal-->
  <div class="modal fade" id="logoutModal" tabindex="-1" role="dialog" aria-labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
      <div class="modal-content">
        <div class="modal-header">
          <h5 class="modal-title" id="exampleModalLabel">Ready to Leave?</h5>
          <button class="close" type="button" data-dismiss="modal" aria-label="Close">
            <span aria-hidden="true">×</span>
          </button>
        </div>
        <div class="modal-body">Select "Logout" below if you are ready to end your current session.</div>
        <div class="modal-footer">
          <button class="btn btn-secondary" type="button" data-dismiss="modal">Cancel</button>
          <a class="btn btn-primary" href="/accounts/logout">Logout</a>
        </div>
      </div>
```

```
  </div>
 </div>

 <!-- Bootstrap core JavaScript -->


  <script src="{% static '/info/homepage/vendor/jquery/jquery.min.js' %}"></script>
 <script src="{% static '/info/homepage/vendor/bootstrap/js/bootstrap.bundle.min.js' %}"></script>


</body>

</html>
```

Student attendance

```
{% extends 'info/base.html' %}
 {% load static %}

 {% block content %}
        <div class="card mb-3">
      <div class="card-header">
        <i class="fas fa-table"></i>
      <b>Attendance</b></div>
      <div class="card-body">
        <div class="table-responsive">
```

```
<table class="table table-bordered text-center" id="dataTable" width="100%" cellspacing="0">
 <thead class="thead-light ">
  <tr>
     <th>Course ID</th>
     <th>Course name</th>
     <th>Attended classes</th>
     <th>Total classes</th>
     <th>Attendance %</th>
     <th>Classes to attend</th>
  </tr>
 </thead>
 <tbody>
  {% for a in att_list %}
  <tr>
     <td>{{ a.course_id }}</td>
     <td><a href="{% url 'attendance_detail' a.student.USN a.course.id %}">{{a.course.name}}</a></td>
     <td>{{ a.att_class }}</td>
     <td>{{ a.total_class }}</td>
     {% if a.attendance < 75 %}
        <td class="p-3 mb-2 bg-danger text-white">{{ a.attendance }}</td>
     {% else %}
        <td class="p-3 mb-2 bg-success text-white">{{ a.attendance }}</td>
     {% endif %}
```

```
                <td>{{ a.classes_to_attend }}</td>
              </tr>
              {% empty %}
                  <p>student has no courses</p>
              {% endfor %}

            </tbody>
          </table>
        </div>
      </div>
    </div>


  {% endblock %}
```

Logout

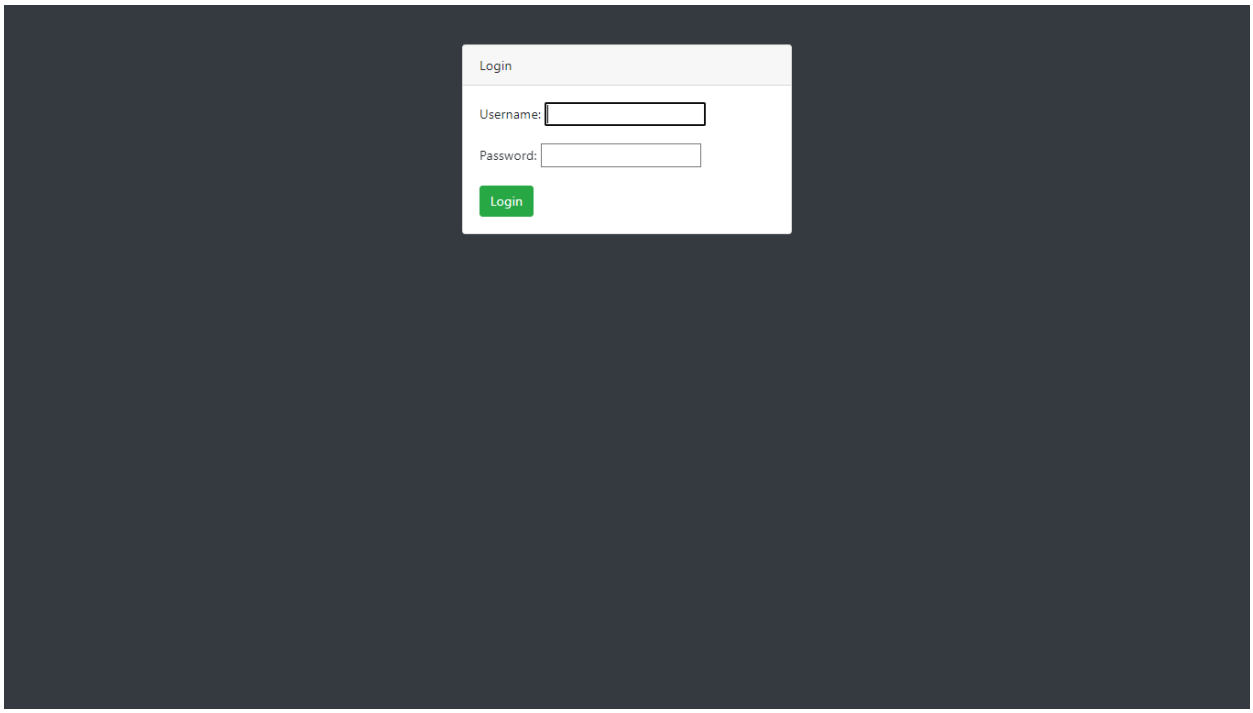
```
<!DOCTYPE html>
<html lang="en">

  <head>

    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

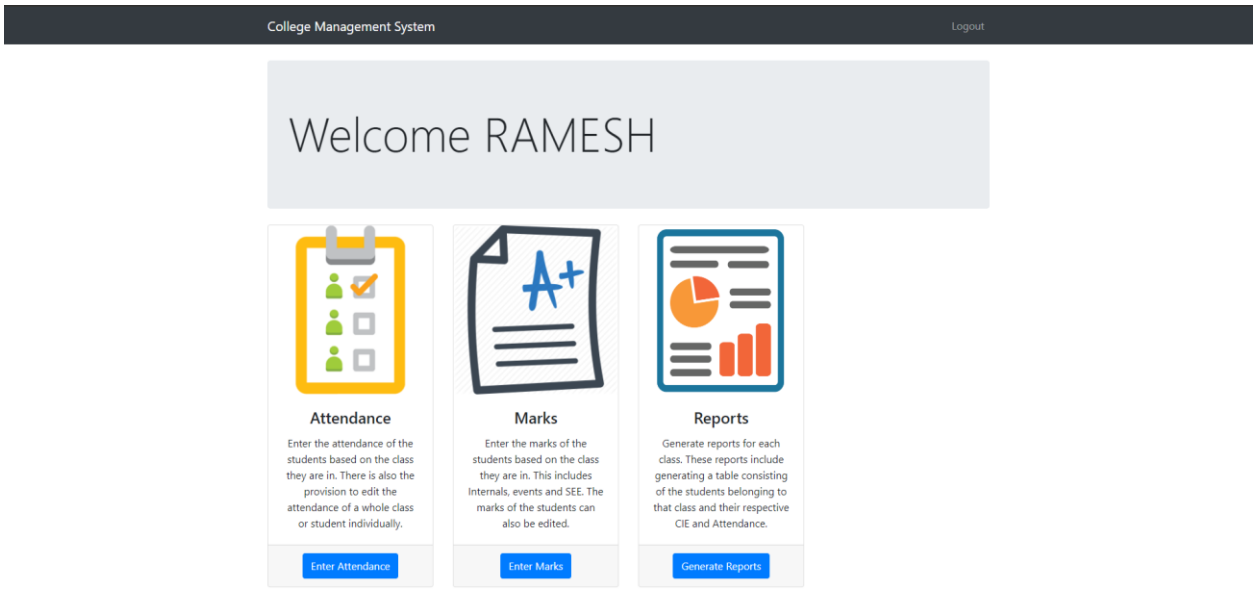
```
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>{% block title %}{% endblock %}</title>
     {% load static %}

    <!-- Bootstrap core CSS-->
    <link href="{% static '/info/bootstrap/vendor/bootstrap/css/bootstrap.min.css' %}" rel="stylesheet">
<!-- Custom fonts for this template-->
    <link href="{% static '/info/bootstrap/vendor/fontawesome-free/css/all.min.css' %}" rel="stylesheet" type="text/css">

    <!-- Page level plugin CSS-->
    <link href="{% static '/info/bootstrap/vendor/datatables/dataTables.bootstrap4.css' %}" rel="stylesheet">

    <!-- Custom styles for this template-->
    <link href="{% static '/info/bootstrap/css/sb-admin.css' %}" rel="stylesheet">

    <!-- Latest compiled and minified CSS -->


     {% block css %}
```

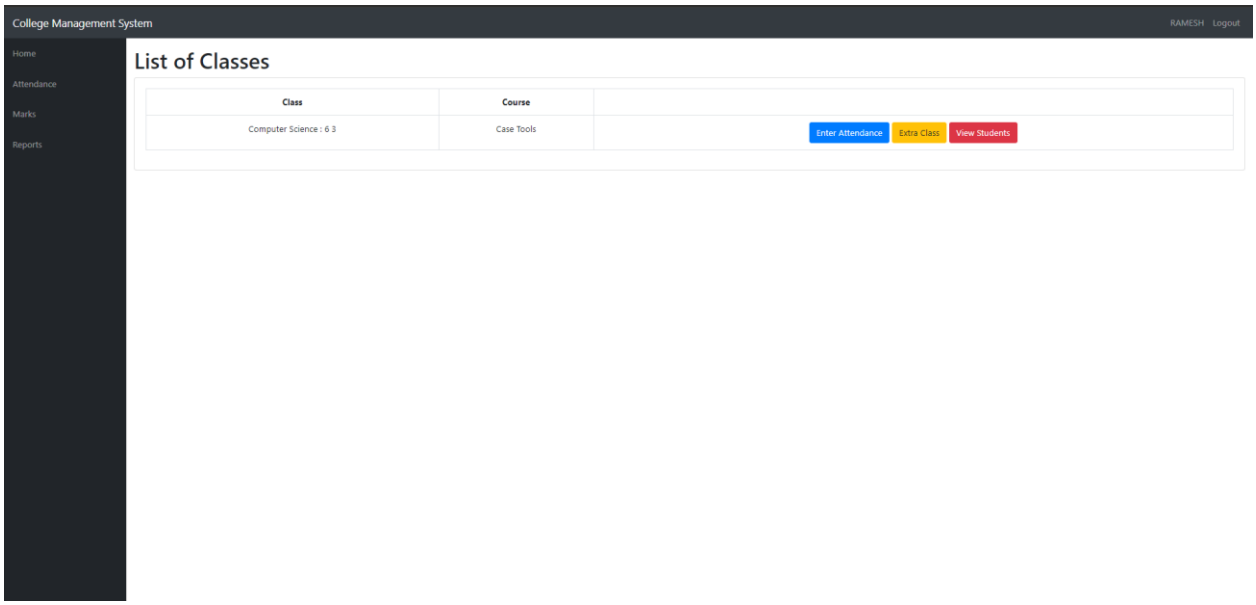
```
    {% endblock %}

  </head>

  <body id="page-top">

    <nav class="navbar navbar-expand navbar-dark bg-dark static-top">

      <a class="navbar-brand mr-1" href="{% url 'index' %}">College Management System</a>

      <button class="btn btn-link btn-sm text-white order-1 order-sm-0" id="sidebarToggle" href="#">
        <i class="fas fa-bars"></i>
      </button>


    </nav>
<div id="wrapper">
<div id="content-wrapper">
<div class="container-fluid">
<div class="text-center h3">
            you have been logged out
            <br><br>
            <a class="btn btn-success" href="/accounts/login">login?</a>
```

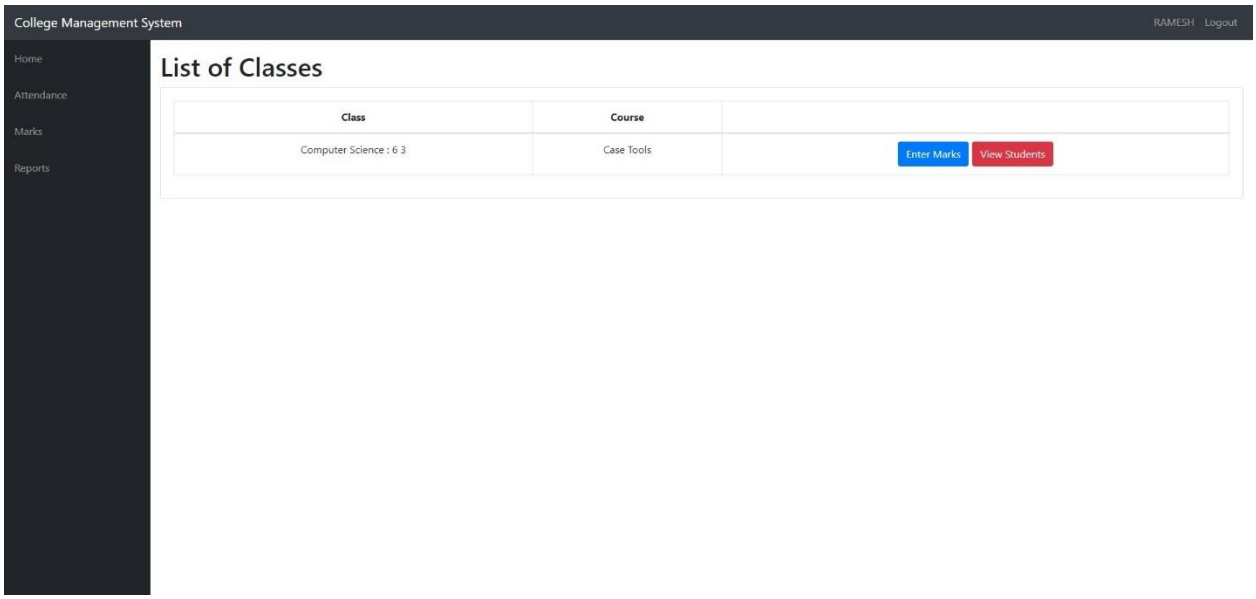
```
          </div>
</div>
       <!-- /.container-fluid -->
</div>
     <!-- /.content-wrapper -->
  </div>
   <!-- /#wrapper -->


   <!-- Scroll to Top Button-->
   <a class="scroll-to-top rounded" href="#page-top">
     <i class="fas fa-angle-up"></i>
   </a>
 <!-- Bootstrap core JavaScript-->
   <script src="{% static '/info/bootstrap/vendor/jquery/jquery.min.js' %}"></script>
   <script src="{% static '/info/bootstrap/vendor/bootstrap/js/bootstrap.bundle.min.js'
%}"></script>
<!-- Core plugin JavaScript-->
   <script src="{% static '/info/bootstrap/vendor/jquery-easing/jquery.easing.min.js'
%}"></script>
<!-- Custom scripts for all pages-->
   <script src="{% static '/info/bootstrap/js/sb-admin.min.js' %}"></script>
{% block scripts %}
   {% endblock %}
  </body>
</html>
```

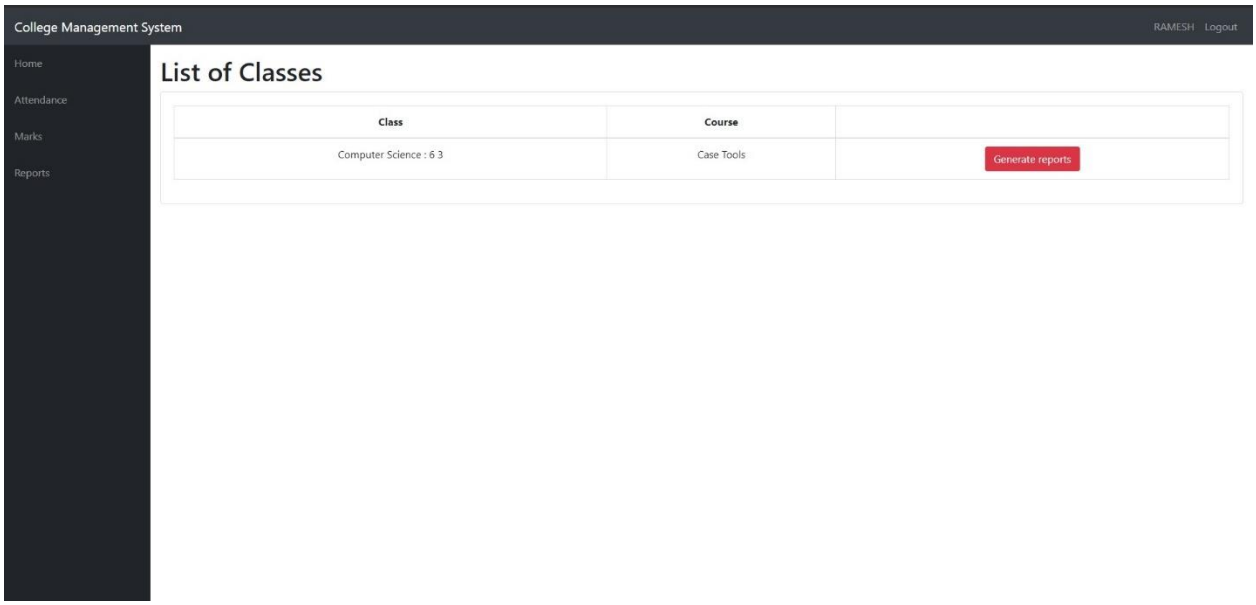
D. SAMPLE INPUT



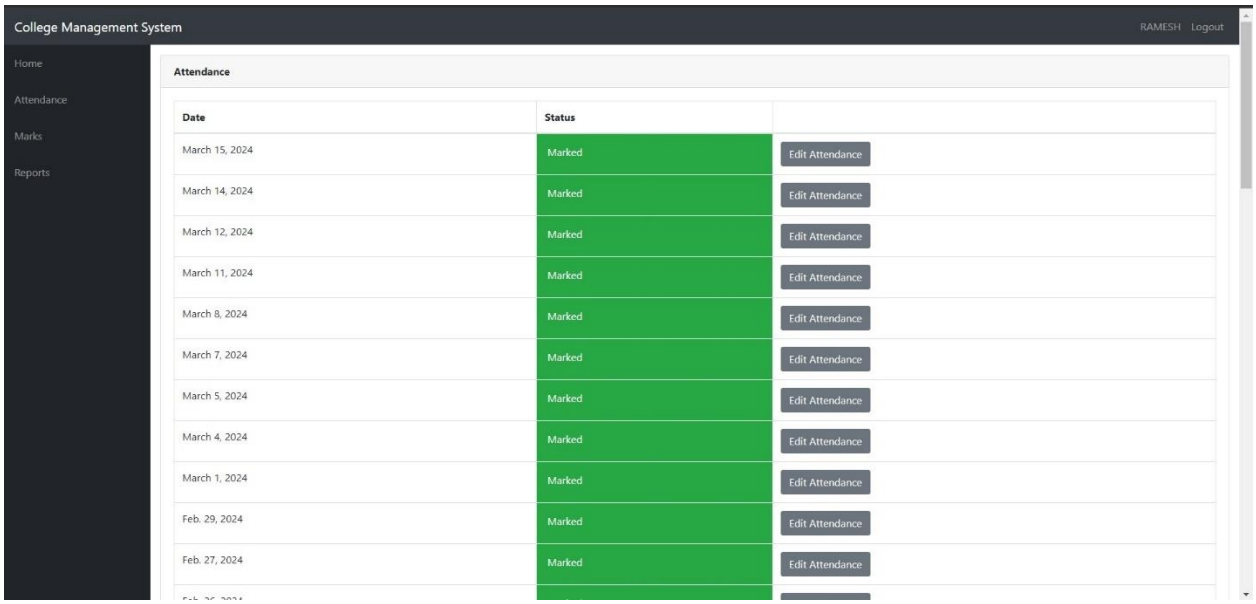


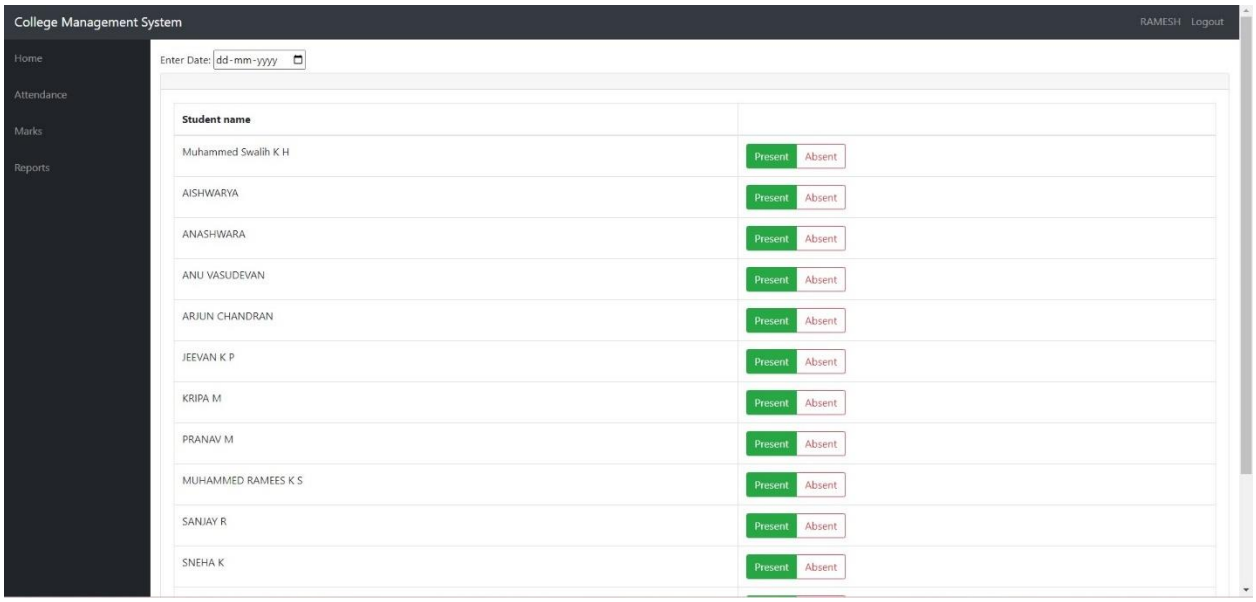

College Management System
RAMESH Logout
Home
Attendance
Marks
Reports
List of Classes
Class
Course
Computer Science : 6 3
Case Tools
Enter Attendance
Extra Class
View Students

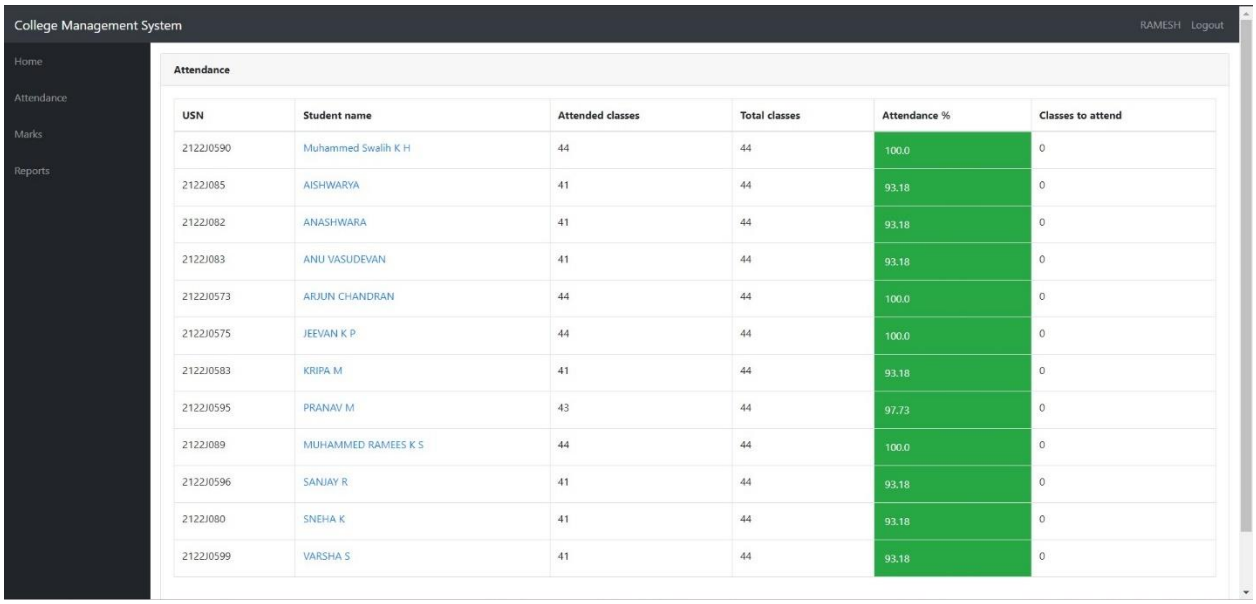

College Management System
RAMESH Logout
Home
Attendance
Marks
Reports
List of Classes
Class
Course
Computer Science : 6 3
Case Tools
Enter Marks
View Students

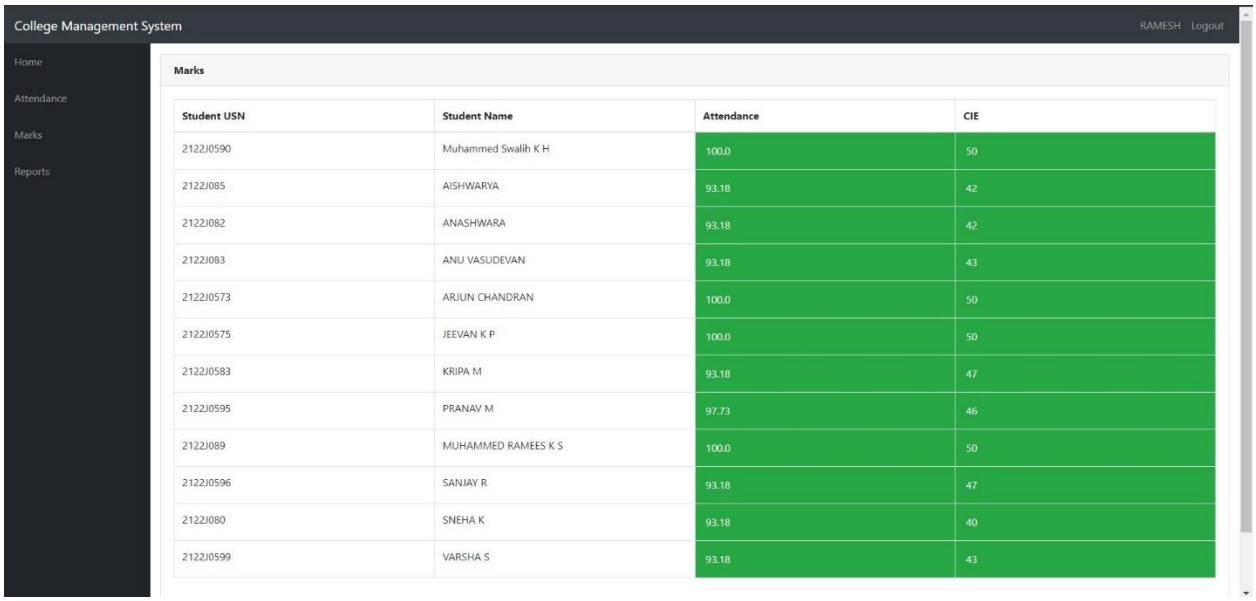


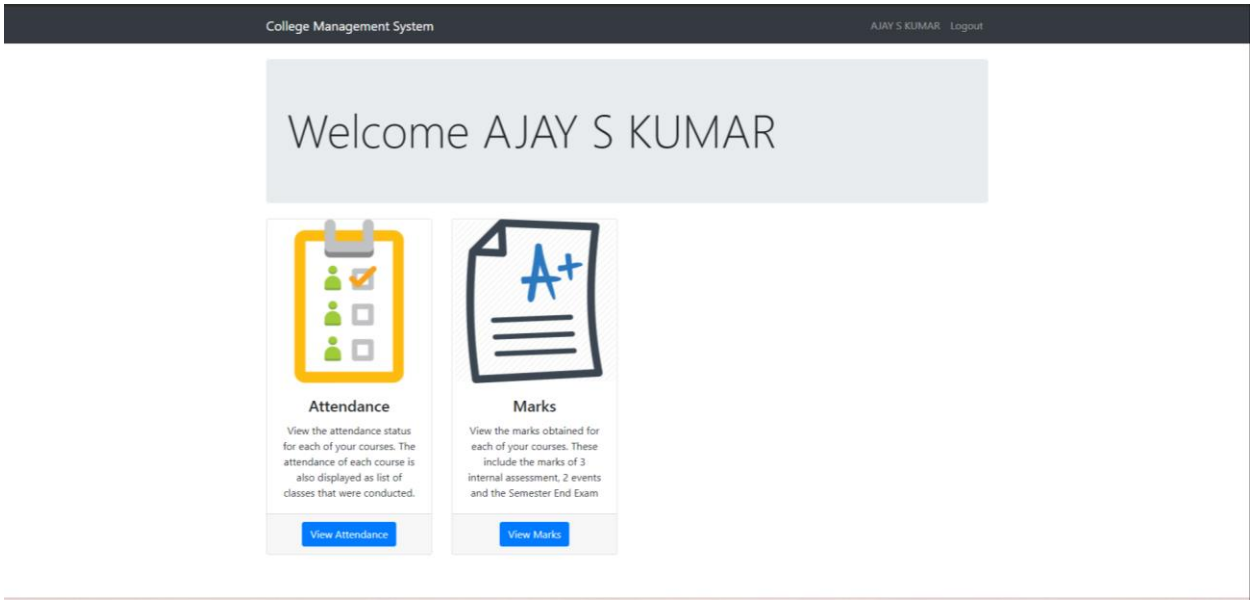
E.SAMPLE OUTPUT

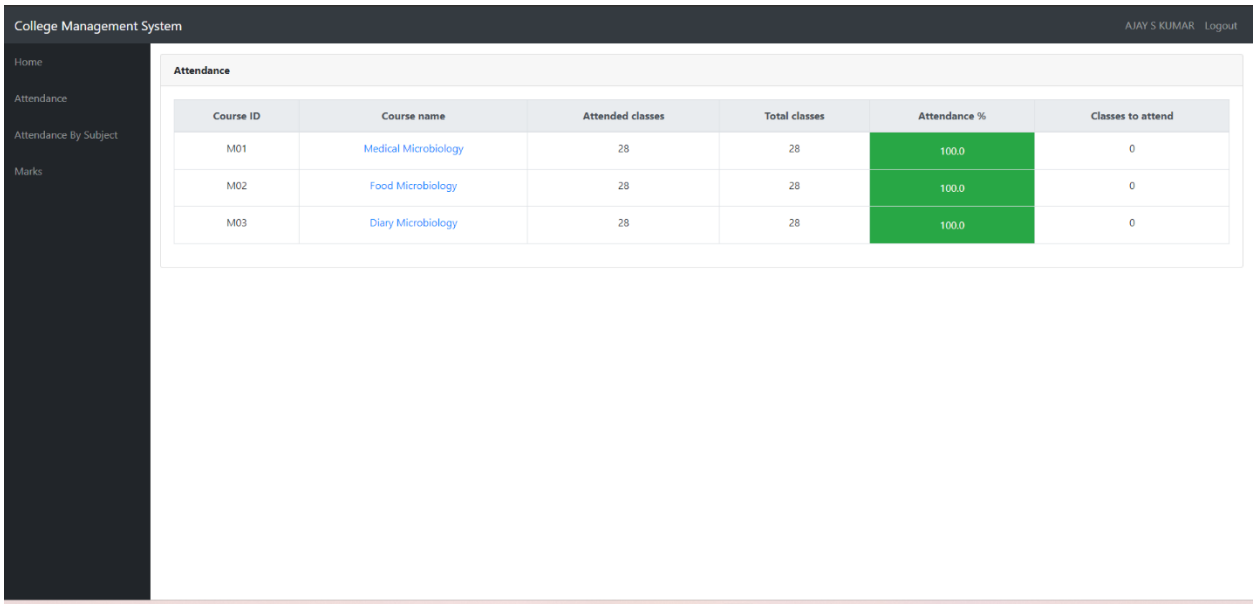


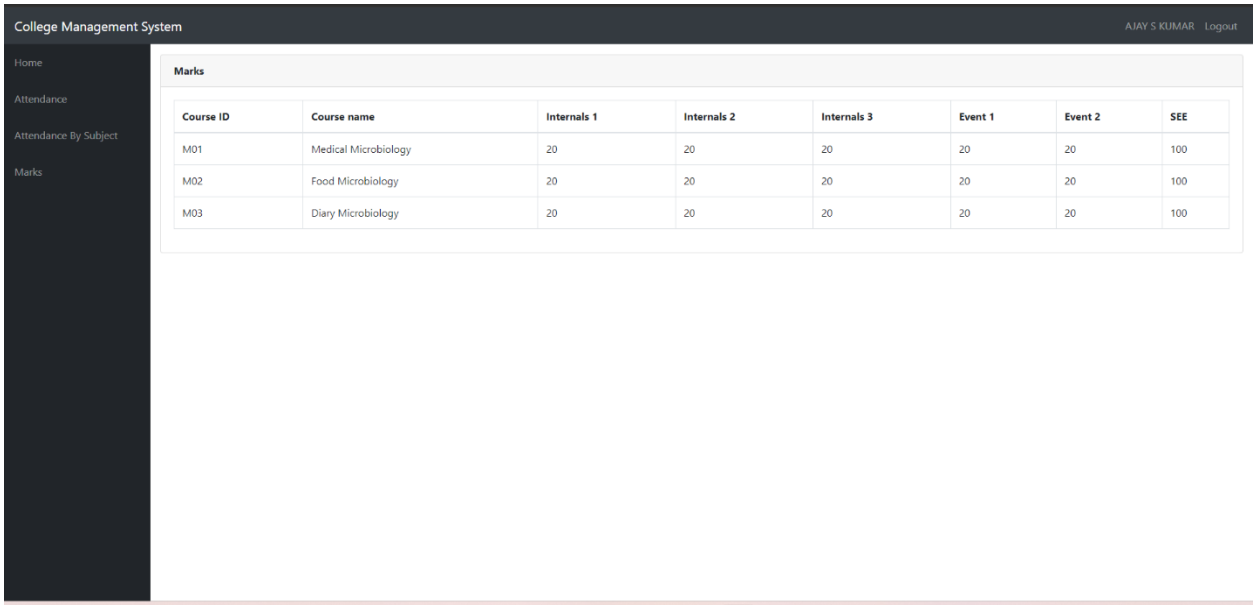

College Management System
RAMESH Logout
Home
Attendance
Marks
Reports
Enter Date: dd-mm-yyyy
Student name
Muhammed Swalih K H
AISHWARYA
ANASHWARA
ANU VASUDEVAN
ARJUN CHANDRAN
JEEVAN K P
KRIPA M
PRANAV M
MUHAMMED RAMEES K S
SANJAY R
SNEHA K
Present
Absent

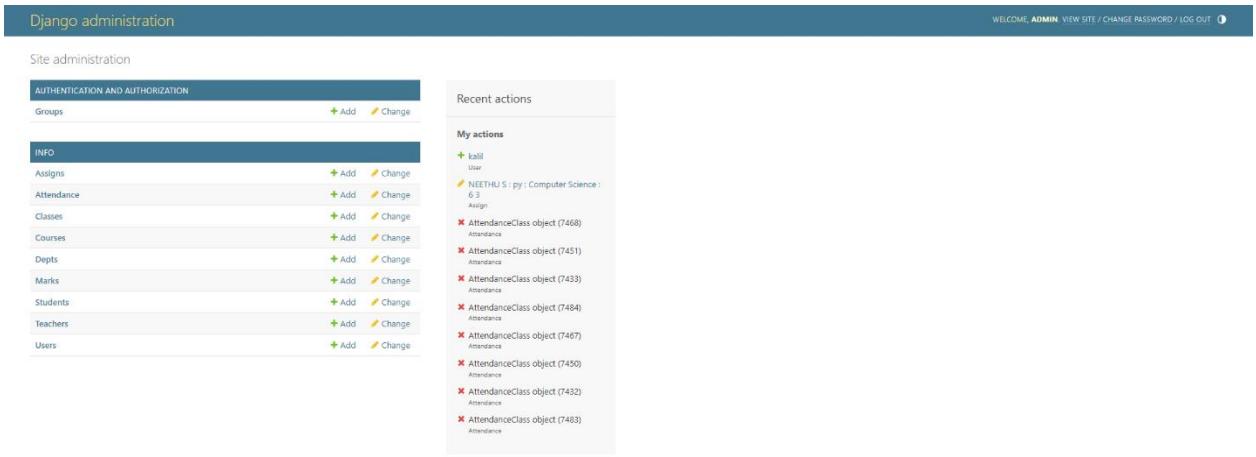

College Management System
RAMESH Logout
Home
Attendance
Marks
Reports
Attendance
USN
Student name
Attended classes
Total classes
Attendance %
Classes to attend
2122J0590 Muhammed Swalih K H 44 44 100.0 0
2122J085 AISHWARYA 41 44 93.18 0
2122J082 ANASHWARA 41 44 93.18 0
2122J083 ANU VASUDEVAN 41 44 93.18 0
2122J0573 ARJUN CHANDRAN 44 44 100.0 0
2122J0575 JEEVAN K P 44 44 100.0 0
2122J0583 KRIPA M 41 44 93.18 0
2122J0595 PRANAV M 43 44 97.73 0
2122J089 MUHAMMED RAMEES K S 44 44 100.0 0
2122J0596 SANJAY R 41 44 93.18 0
2122J080 SNEHA K 41 44 93.18 0
2122J0599 VARSHA S 41 44 93.18 0


College Management System
RAMESH Logout
Home
Attendance
Marks
Reports
Marks
Student USN
Student Name
Attendance
CIE
2122J0590
Muhammed Swalih K H
100.0
50
2122J085
AISHWARYA
93.18
42
2122J082
ANASHWARA
93.18
42
2122J083
ANU VASUDEVAN
93.18
43
2122J0573
ARJUN CHANDRAN
100.0
50
2122J0575
JEEVAN K P
100.0
50
2122J0583
KRIPA M
93.18
47
2122J0595
PRANAV M
97.73
46
2122J089
MUHAMMED RAMEES K S
100.0
50
2122J0596
SANJAY R
93.18
47
2122J080
SNEHA K
93.18
40
2122J0599
VARSHA S
93.18
43


College Management System
AJAY S KUMAR Logout
Welcome AJAY S KUMAR
Attendance
View the attendance status for each of your courses. The attendance of each course is also displayed as list of classes that were conducted.
View Attendance
Marks
View the marks obtained for each of your courses. These include the marks of 3 internal assessment, 2 events and the Semester End Exam
View Marks


College Management System
AJAY S KUMAR Logout
Home
Attendance
Attendance By Subject
Marks
Attendance
Course ID
Course name
Attended classes
Total classes
Attendance %
Classes to attend
M01
Medical Microbiology
28
28
100.0
0
M02
Food Microbiology
28
28
100.0
0
M03
Diary Microbiology
28
28
100.0
0


College Management System
AJAY S KUMAR Logout
Home
Attendance
Attendance By Subject
Marks
Marks
Course ID
Course name
Internals 1
Internals 2
Internals 3
Event 1
Event 2
SEE
M01
Medical Microbiology
20
20
20
20
20
100
M02
Food Microbiology
20
20
20
20
20
100
M03
Diary Microbiology
20
20
20
20
20
100


Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Site administration
AUTHENTICATION AND AUTHORIZATION
Groups + Add Change
INFO
Assigns + Add Change
Attendance + Add Change
Classes + Add Change
Courses + Add Change
Depts + Add Change
Marks + Add Change
Students + Add Change
Teachers + Add Change
Users + Add Change
Recent actions
My actions
+ kalil
User
NEETHU S : py : Computer Science : 6 3
Assign
AttendanceClass object (7468)
Attendance
AttendanceClass object (7451)
Attendance
AttendanceClass object (7433)
Attendance
AttendanceClass object (7484)
Attendance
AttendanceClass object (7467)
Attendance
AttendanceClass object (7450)
Attendance
AttendanceClass object (7432)
Attendance
AttendanceClass object (7483)
Attendance