NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

Nitte (DU) established under Section 3 of UGC Act 1956 | Accredited with 'A+' Grade by NAAC

Department of Computer Science & Engineering

Report on Mini Project

Student Attendance Management System

Course Code: CS2102-1

Course Name**: Database Management System**

Semester: IV SEM Section: A

Submitted To:

Ms. Jayapadmini Kanchan
Assistant Professor [Gd-II]
Department of Computer Science and Engineering

Submitted By:

Prajwal K Sagar (NNM22CS127)
Yashas Hegde (NNM22CS209)

Date of submission:
19/04/2024

Signature of Course Instructor

ABSTRACT

The Student Attendance Management System (SAMS) is a web-based application designed to streamline the process of recording and managing student attendance in educational institutions. Leveraging XAMPP server, PHP, HTML, and CSS, SAMS offers a user-friendly and efficient solution for faculty and administrators. The system boasts functionalities such as automated attendance recording, allowing faculty to mark student presence through various methods (depending on implementation) within the application. This eliminates the need for manual attendance sheets and reduces the risk of errors. SAMS securely stores attendance data in a database, enabling the generation of comprehensive reports. These reports provide valuable insights into individual and class attendance patterns, facilitating data-driven decision making. Additionally, the system allows for the creation and management of user accounts for faculty and administrators. By automating attendance recording, SAMS offers significant improvements in efficiency, saving faculty valuable time and effort. Furthermore, the system enhances accuracy by eliminating human error associated with manual attendance tracking. The comprehensive reports generated by SAMS provide administrators with valuable data to identify and address potential issues related to student absenteeism. In some cases, the system can even be configured to provide authorized users, such as parents, with access to attendance information, fostering transparency.

ACKNOWLEDGMENT

It is with great satisfaction and enthusiasm that we are submitting the Mini Project Report on "**Student Attendance Management System**". We have completed it as a part of the IV **semester DBMS Laboratory with Mini Project (CS2102-1)** of Bachelor of Engineering in **Computer Science & Engineering** of Nitte University, Deralakatte.

I take this opportunity to express my gratitude and regards to my project guide, **Ms. Jayapadmini Kanchan**, Assistant Professor Gd-II, Department of Computer Science Engineering for her guidance, monitoring, and encouragement throughout this Mini project.

I would like to express my deepest gratitude **to Dr. Jyothi Shetty**, Head of the Department of Computer Science and Engineering, for her invaluable guidance, unwavering support, and encouragement throughout my academic journey.

My sincere thanks to our beloved principal**, Dr. Niranjan N Chiplunkar**, and vice principal **Dr. I.R.Mithanthaya** for permitting us to carry out this Mini project at our college and providing us with all needed facilities.

I thank my parents and friends for the immense support, honest opinion, and suggestions throughout the mini project.

Table of Contents

List of Figures

Chapter 1

INTRODUCTION

In the domain of education, the meticulous upkeep of precise and effective attendance logs holds paramount importance. Yet, conventional approaches often prove laborious and susceptible to inaccuracies. Addressing this issue head-on, the Student Attendance Management System (SAMS) emerges as a transformative solution. Leveraging the XAMPP server alongside PHP, HTML, JavaScript and CSS, SAMS introduces a seamless web-based interface, catering to the needs of faculty and administrators alike. This introduction serves as a gateway to explore the comprehensive functionalities, advantages, and transformative potential of SAMS in reshaping attendance management across educational establishments.

1.1 Purpose

At its core, the Student Attendance Management System (SAMS) endeavors to transform the landscape of student attendance tracking and administration within educational institutions. Our project endeavors to transcend the limitations of conventional, labor-intensive practices by harnessing the power of technology to deliver a user-friendly and highly effective web-based solution. Developed utilizing the XAMPP server, PHP, HTML, and CSS, this system aims to simplify workflows for both faculty members and administrators, leading to heightened data precision and furnishing invaluable insights to bolster educational achievements.

1.2 Scope

The scope of the Student Attendance Management System (SAMS) revolves around streamlining attendance recording and management within educational institutions. Its core functionalities target faculty, allowing them to efficiently record attendance through various methods. SAMS securely stores this data in a database, enabling administrators to generate comprehensive reports for analysis. User management allows for creating and managing faculty and administrator accounts. Looking ahead, the scope of SAMS can be expanded to include a mobile application for on-the-go access, biometric authentication for secure recording, and functionalities for managing student leave requests. Ultimately, the scope of SAMS prioritizes user-friendliness, scalability, and security while offering a roadmap for future development that caters to the evolving needs of educational institutions.

.

1.3 Overview

The Student Attendance Management System (SAMS) tackles the challenge of inefficient and error-prone attendance tracking methods in educational institutions. Built using XAMPP server, PHP, HTML, and CSS, SAMS offers a user-friendly web-based platform to streamline attendance recording and management for faculty and administrators. The system automates attendance recording through various methods (depending on implementation), eliminating the need for manual processes and reducing errors. SAMS securely stores attendance data in a database, allowing for the generation of comprehensive reports that provide valuable insights into student attendance patterns. This information empowers administrators to make data-driven decisions to improve educational outcomes. Furthermore, the system offers user management for faculty and administrators, ensuring appropriate access levels. SAMS presents a comprehensive solution for educational institutions seeking to modernize attendance management and gain valuable data for informed decision-making.

Chapter 2

REQUIREMENTS SPECIFICATION

2.1 Hardware Specification

- Processor : Intel(R) Core(TM) i3-1005G1 CPU @ 1.20GHz 1.19 GHz
- RAM : 8GB
- Hard Disk : 1TB
- Input Device : Standard keyboard and Mouse
- Output Device : Monitor

2.2 Software Specification

- Database: MySQL 5.5
- Markup Language: HTML5
- Scripting Language: PHP 7.0.1
- IDE: Visual Studio Code
- Server: Apache
- Browser: Google Chrome, Microsoft Edge , Firefox

Chapter 3

SYSTEM DESIGN

3.1 ER Diagram

For the project we are taking 4 strong entities Student, Course, Faculty, Session. Student has 3 attributes id, name, rollno where id is primary key. Course has cid, code, title, credits and cid is primary key. For Faculty there are fid, username, password, fullname where fid is primary key. Session has sid, year, term where sid is primary key.

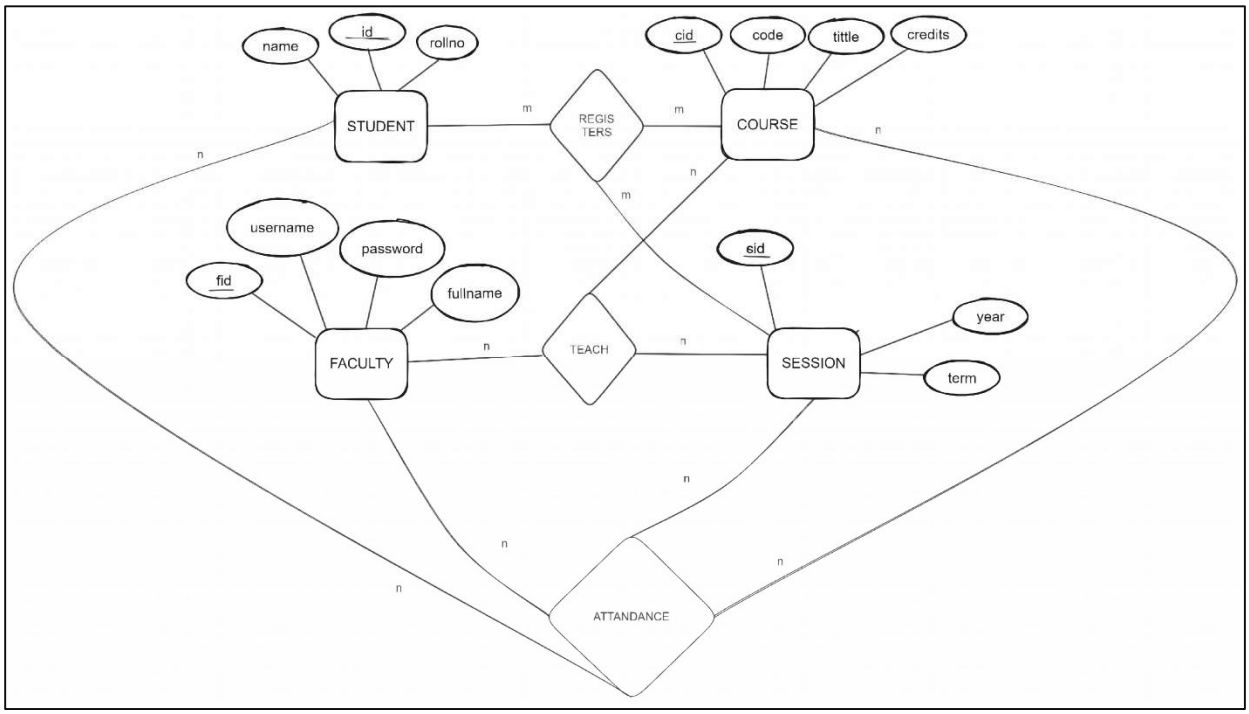


Fig 3.1: ER Diagram of SAMS

3.2 Mapping From ER diagram to Schema

To convert ER to schema we follow 7 steps which are as follows:

1. **Mapping of Regular Entities**: This step involves mapping all the regular strong entities types to tabular format by identifying their primary keys.
2. **Mapping of 1:1 Relation:** In this step foreign keys are assigned using foreign key approach. The primary key of the participating relations are added as primary key to second entity types by looking at the participating constraints.
3. **Mapping of 1:N Relation:** Foreign key approach is used to add one sided primary key to the n sided entity at foreign key.
4. **Mapping of M:N Relation:** Here we use the cross-reference approach where the relationship is converted to a new relation within attributes on primary keys of both participating relations.
5. **Mapping of Weak Entity:** When mapping weak entity types along with other attributes the partial key and primary key of parent entity together will form their primary key of the new relation.
6. **Mapping of N-ary Relation:** For mapping N array relationship we create a new relation with a relationship name in its attribute and primary keys of all participating entity types.
7. **Mapping of Multivalued Relation:** For multivalued attributes a separate relation has to be created along with primary key of parent relation.

To get schema for database we will follow these steps:

1. **Mapping of Regular Entities:** Initially, we will identify all the strong entities (the entities which have primary key in them). In our database these are the entities with the attributes
2. **Mapping of 1:1 Relation:** None of the entities are participating in the 1:1 relation type. In it each record in 1 table corresponds uniquely to a record in another table.
3. **Mapping of 1:N Relation:** In our database all the entities are participating in 1:n. In a one-to-many relationship, the "n" side entity includes a foreign key referencing the

4. **Mapping of M:N Relation:** All of the entities are participating in m:n relation. In a many-to-many relationship, a separate associative entity is created to link the participating entities.

   attendance_details (fid, cid, sid, id, date)

   course_allotment (fid, cid, sid)

   course_details (cid, code, title, credits)

   course_registrations (id, sid, cid,)

   faculty_details (cid, user_name, name, password)

   session_details (cid, year, term)

   student_details (sid, roll_no, name)

5. **Mapping of Weak Entities:** We will identify all the weak entities(the entities which don't have primary key in them). In our database these are no entities with such attributes.
6. **Mapping of N-ary Relation:** None of the entities are participating in this relation. In it the relation is linked to and linked from same entity.
7. **Mapping of Multivalued relation:** A multivalued attribute allows an entity to have multiple values for a single attribute. This is typically represented as a separate table with a foreign key referencing the primary key of the original entity.

3.3 Assumptions

1. Assumption of Database Usage: The assumption that the MySQL database is used for storing data related to the project.

2. Assumption of Table Structure: The assumption that the database consists of several tables representing different entities such as Faculty, Courses, Students, Sessions, and Attendance.

3. Assumption of Primary Keys: The assumption that each table has a primary key

column (e.g., id, cid, fid, sid) to uniquely identify each record within that table.

4. Assumption of Indexes: The assumption that indexes are created on certain columns (e.g., userame, id) to improve query performance.

5. Assumption of Auto-increment: The assumption that certain primary key columns (e.g., id, cid, fid, sid) are set to auto-increment to automatically generate unique values for new records.

3.4 Schema Diagram

A Schema is a pictorial representation of the relationship between the tables in the database that is created. The term ”schema” refers to the representation of data as a blueprint of how the database is constructed (divided into database tables in the case of relational databases). The formal definition of a schema is a set of formulas (sentences) called integrity constraints imposed on a database. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language .The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real-world entities are modelled in the database.

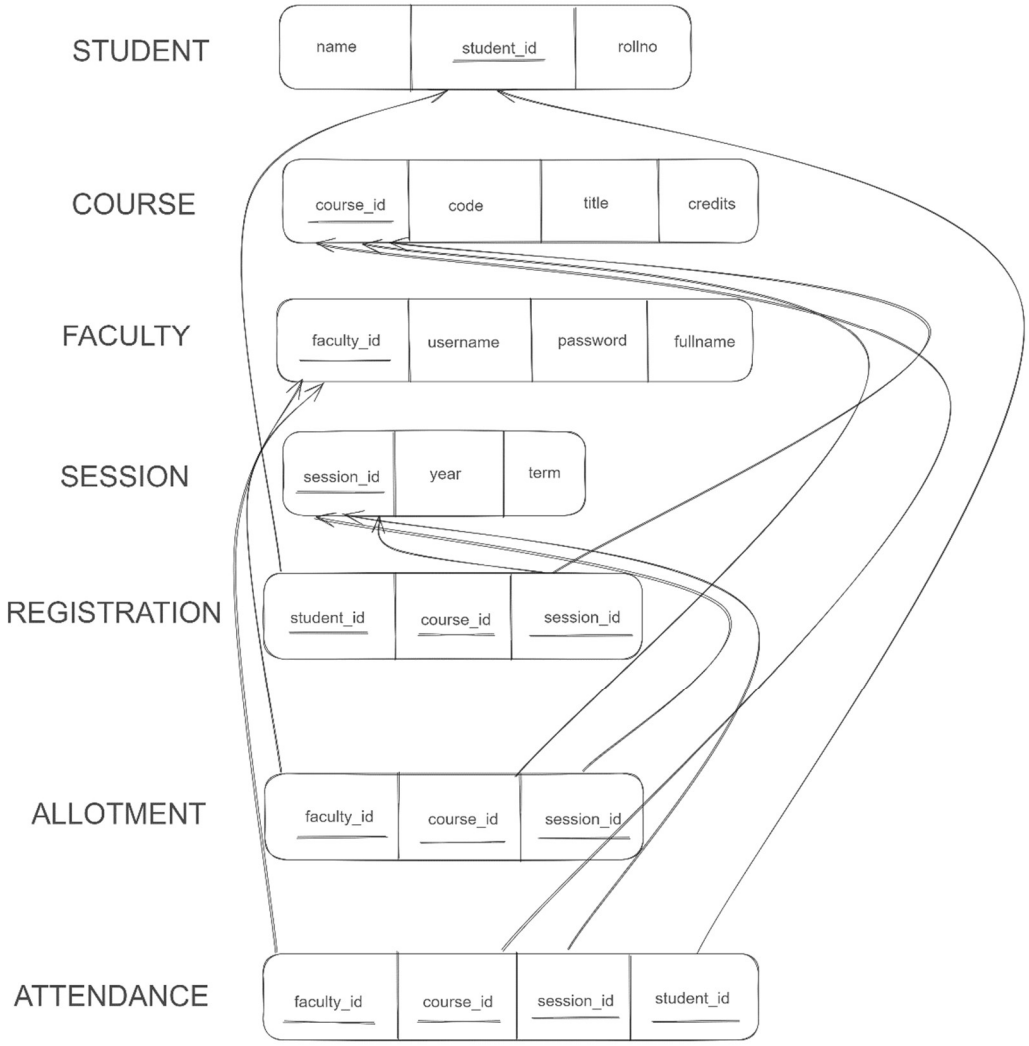


Fig 3.2: Schema Diagram

Chapter 4

IMPLEMENTATION

4.1 Pseudocodes used

Pseudocode to connect SQL and PHP:

In order to store or access the data inside a MySQL database, we first need to connect to the MySQL database server. In PHP we can do this using the MySQL connect () function. All communication between PHP and the MySQL database server takes place through this connection. The hostname parameter in the above syntax specifies the host name, whereas the username and password parameters specify the credentials to access MySQL server, and the database parameter, if provided will specify the default MySQL database to be used when performing queries. The default username for MySQL database server is root and there is no password and hostname is localhost.

```php
<?php
class Database
{
    private $servername = "localhost";
    private $username = "root";
    private $password = "";
    private $dbname = "attendance_db";
    public  $conn=null;

    public  function __construct() {
        try {
            $this->conn = new PDO("mysql:host=$this->servername;dbname=$this->dbname", $this->username, $this->password);
            // set the PDO error mode to exception
            $this->conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
            //echo "Connected successfully";
          } catch(PDOException $e) {
            echo "Connection failed: " . $e->getMessage();
          }
    }
}

?>
```

Fig 4.1 Pseudocode to connect SQL

4.2 Tables used

Total 7 tables and they are as following :

attendance_details (fid, cid, sid, id, date)

course_allotment (fid, cid, sid)

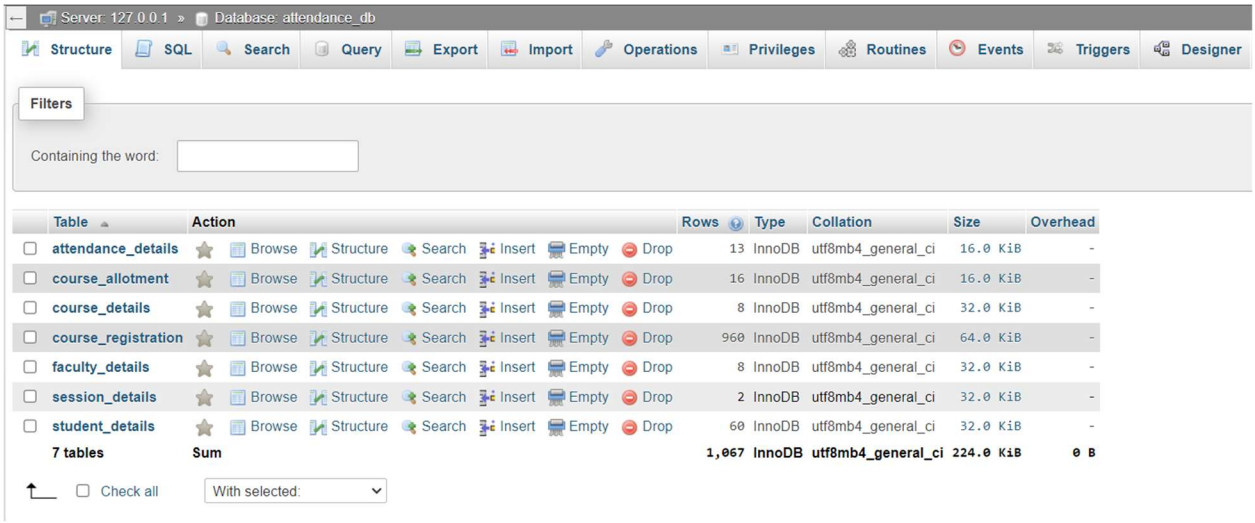
course_details (cid, code, title, credits)

course_registrations (id, sid, cid,)

faculty_details (cid, user_name, name, password)

session_details (cid, year, term)

student_details (sid, roll_no, name)



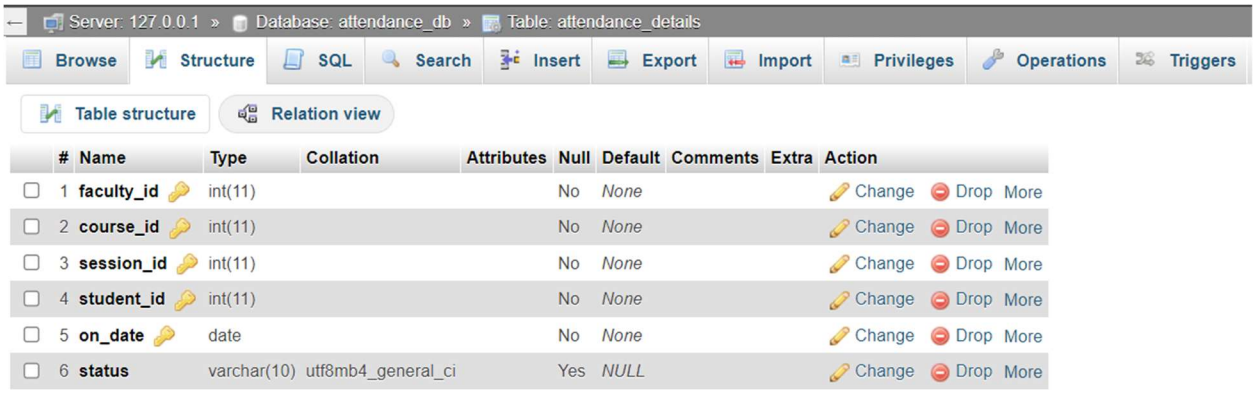


Fig 4.2 Attendance details Table

Fig 4.3 Course allotment Table

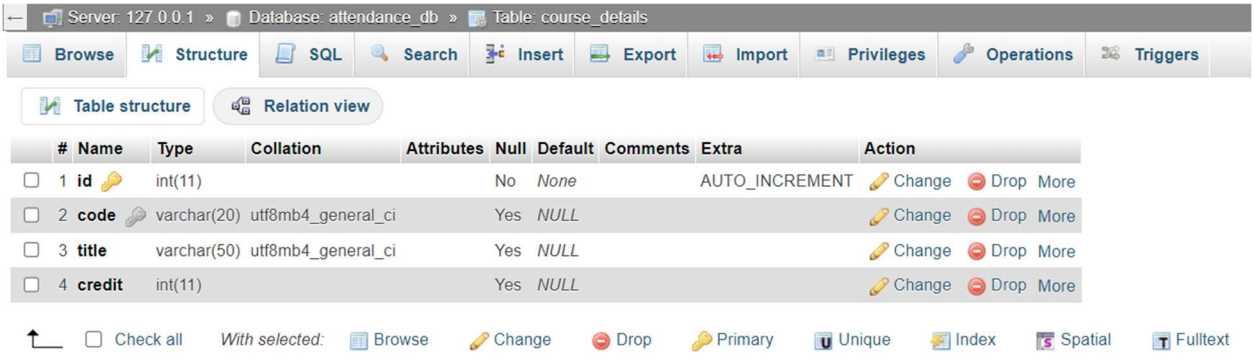


Fig 4.4 Course details Table

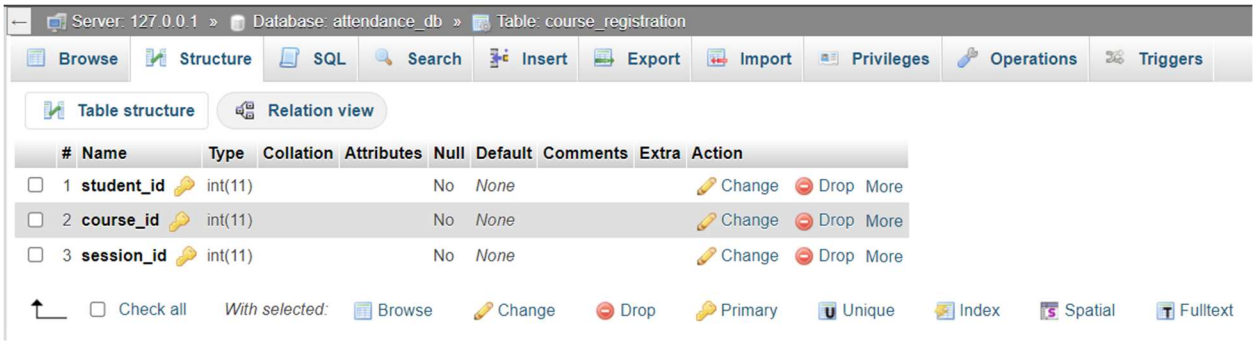


Fig 4.5 Course registration Table

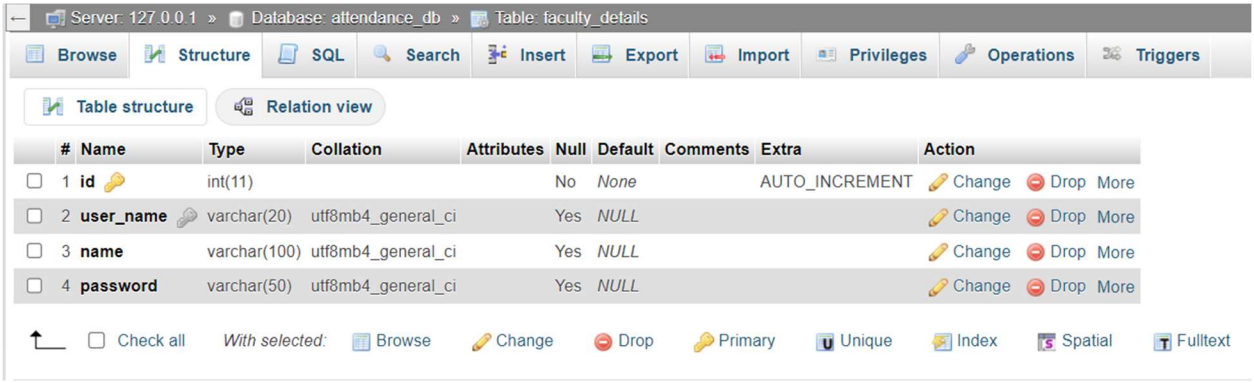


Fig 4.6 Faculty details Table

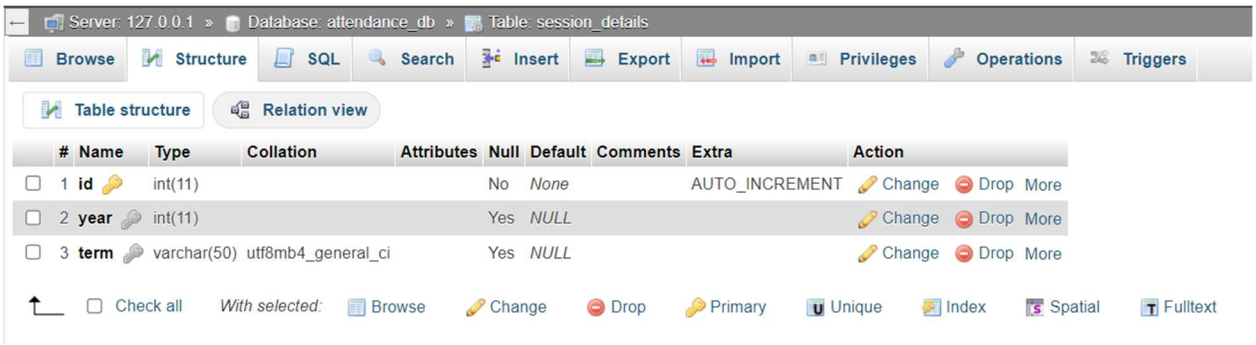


Fig 4.7 Session details Table

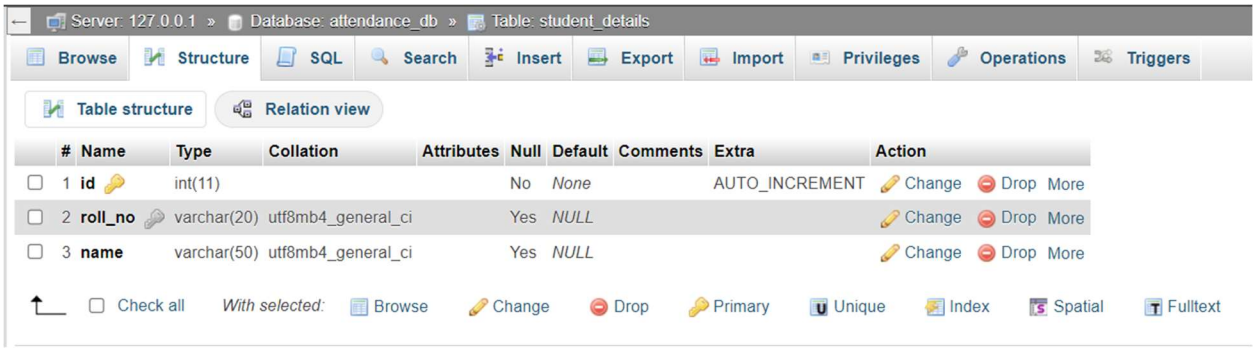


Fig 4.8 Student details Table

Chapter 5

RESULTS AND DISCUSSION

Login page:

Through this page Faculty can login to his/her account

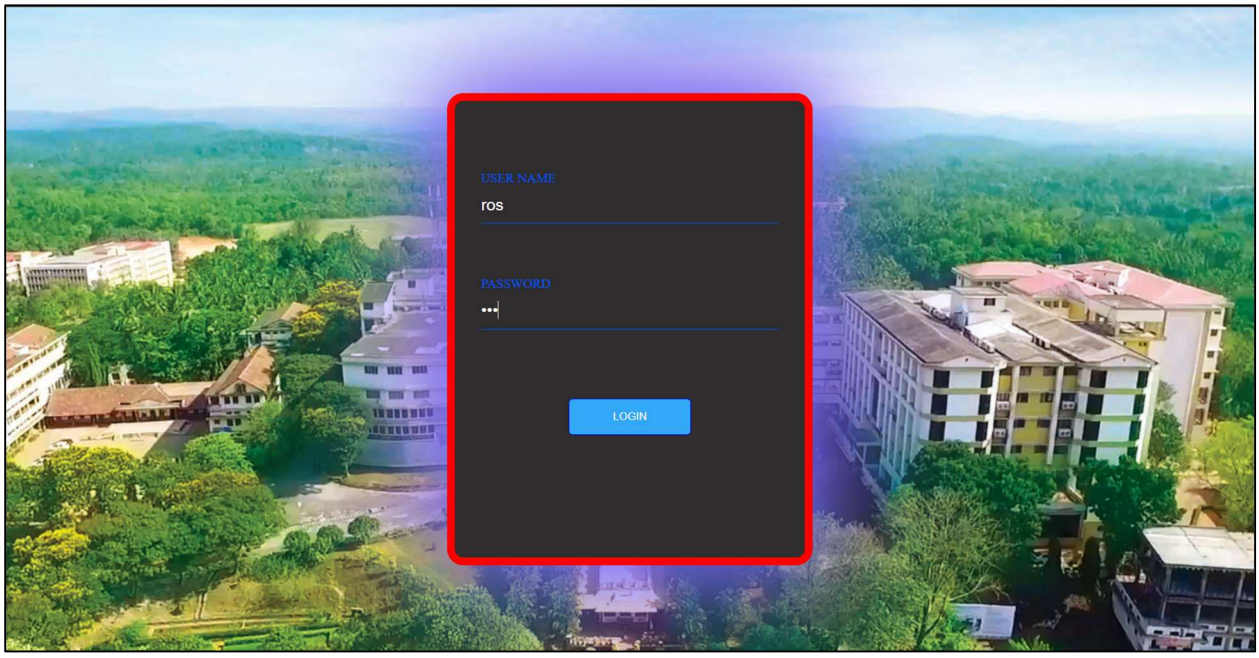


Fig 4.9: Login Page

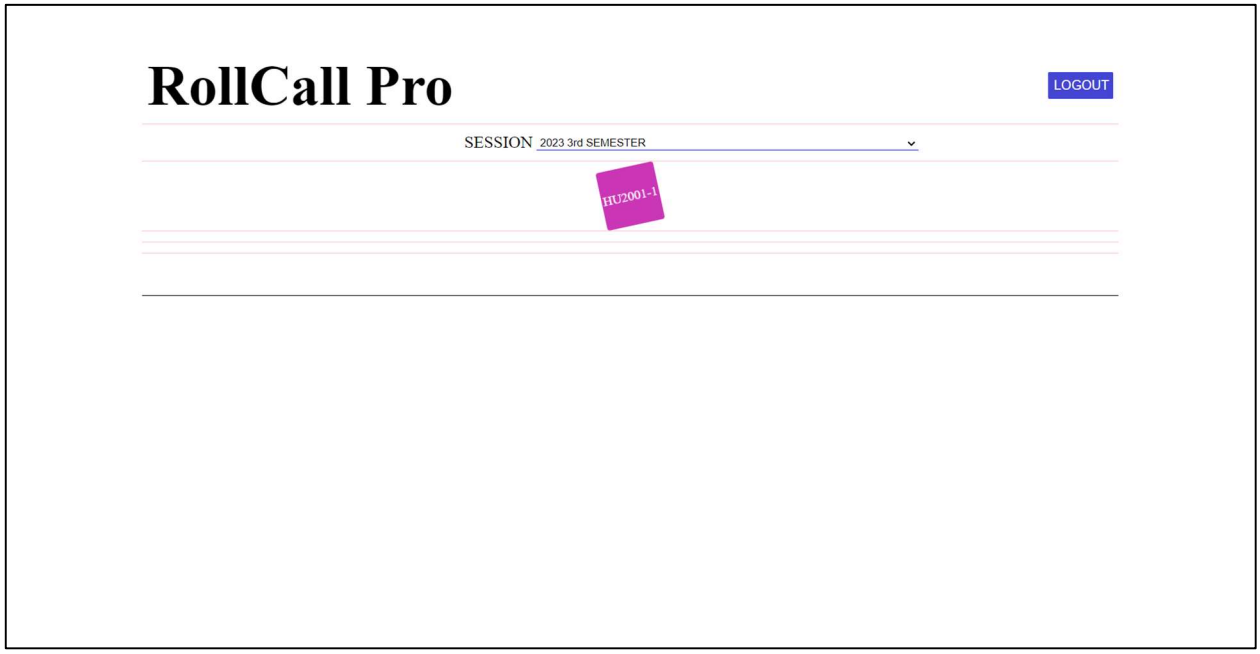


Fig 5.1 Course selection Page

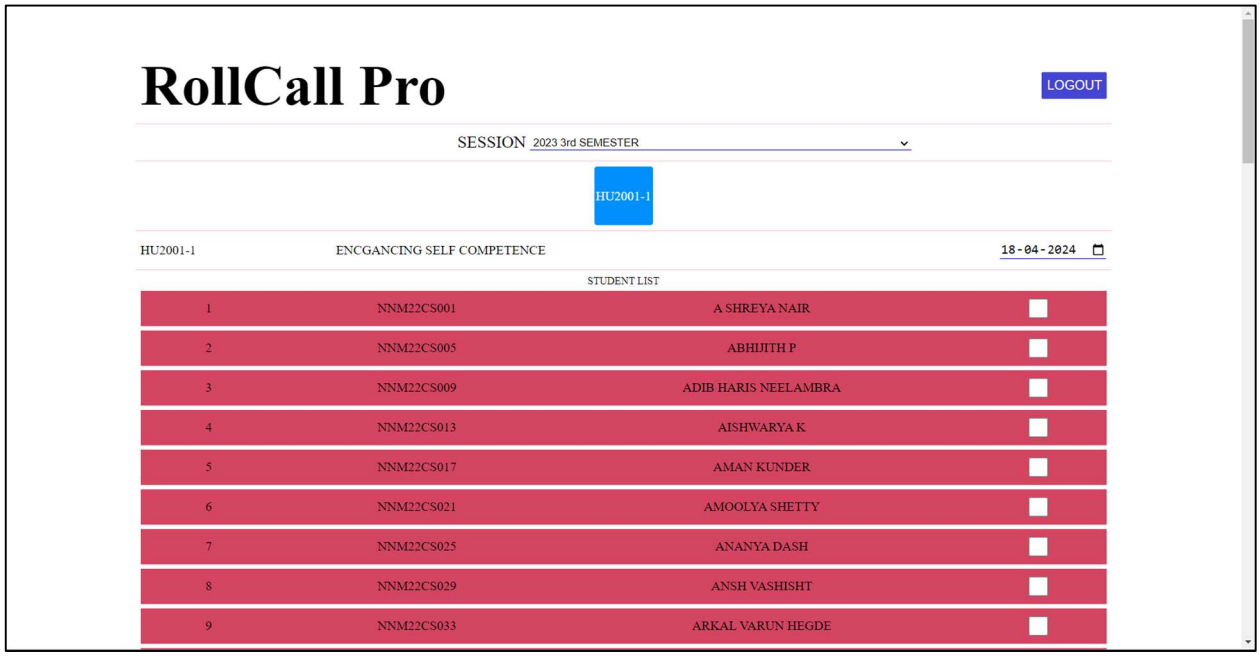


Fig 5.2: Student list Page

Attendance Report in csv format:

Through this faculty can get a csv file of the attendance report of that class

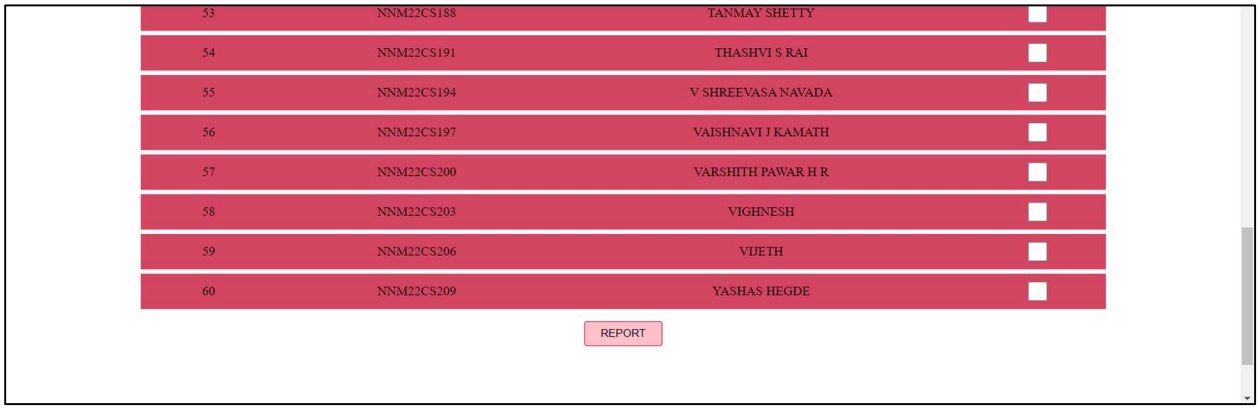


Fig: 5.3 Report

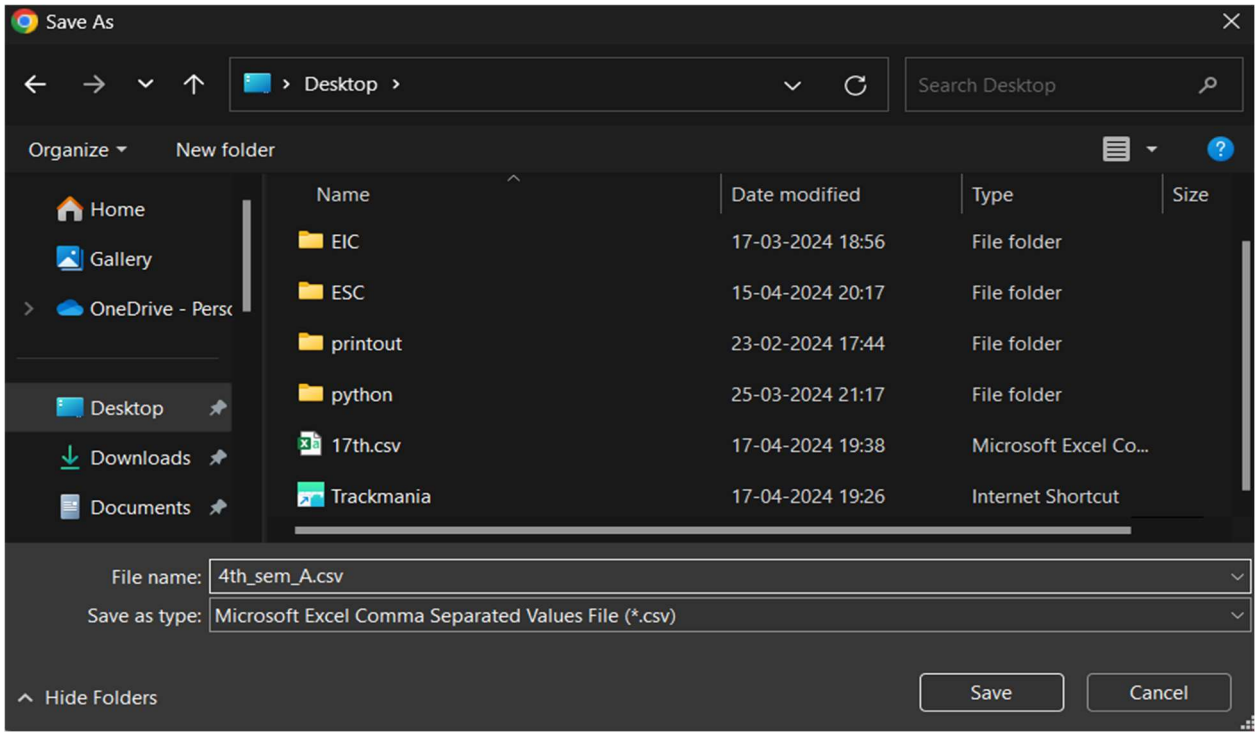


Fig: 5.4 Saving the report

Chapter 6

CONCLUSION AND FUTURE WORK

The Student Attendance Management System (SAMS) proves its effectiveness as a user-friendly and efficient solution for educational institutions. Built with XAMPP server, PHP, HTML, and CSS, SAMS automates attendance recording, saving faculty time and boosting accuracy through secure data storage and eliminating manual errors. Comprehensive reports generated by the system offer valuable insights into student attendance patterns, empowering administrators to make data-driven decisions. Furthermore, SAMS promotes transparency by providing authorized users with access to attendance information.

Looking ahead, SAMS holds exciting potential for further development. A mobile application would allow for convenient access to attendance information on the go, while integrating biometric authentication could offer a more secure and efficient recording method. Expanding functionalities to include managing student leave requests would provide a more comprehensive solution. Additionally, features for advanced analytics could be implemented to gain deeper insights into attendance patterns, and customization options could allow schools to tailor SAMS to their specific needs. By continuing to develop and expand, SAMS can become an even more valuable tool for educational institutions.

REFERENCES

[1] Database systems Models, Languages, Design and Application Programming, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, Pearson.

[2] Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill.

[3] Silberschatz Korth and Sudharshan: Database System Concepts, 6th Edition, McGraw Hill, 2013.

[4] Coronel, Morris, and Rob, Database Principles Fundamentals of Design, Implementation and Management, Cengage Learning 2012.