

Content Management System

An Idea Implementation Of Database Project

Proposed by:

Kashish (023-22-0098)

Suman (023-22-0138)

Kirti Kour (023-22-0121)

Proposed to:

Dr. Muhammad Hussain Mughal

Introduction

Welcome to our Content Management System (CMS)!

CMS is a Content Management System desktop application that is used by almost all the educational institutions (in today's era of technology) to manage their records. With distinct user roles, Admin, Instructor and Student, our system ensures secure login processes and enhanced functionalities. Admins are responsible for addition/registration of teachers, students, and the courses. In addition, they will also be responsible for managing the timetables of the classes. Lastly, they will be able to change their passwords (if they want to). Teachers have the rights to view their personal information and their class schedules, mark student's attendances, assign grades to students, and change their profile's password (if needed). However, students can view their personal information and class schedules, check their attendances, and the grades assigned to them, along with changing their profile's passwords. The application is quite user-friendly and easy to use.

The primary objective of this application is to develop an enhanced version of the Content Management System (CMS) for the ease of educational institutions. The software will be developed using the Graphical User Interface and some other methods of Java Programming language on the front end, while on the backend, it will be developed using MySQL.

Problem Statement

In today's rapidly evolving educational landscape, the efficient management of academic and administrative tasks within educational institutions has become increasingly challenging. The manual, paper-based methods employed for student management, teacher coordination, and administrative tasks have resulted in significant administrative burdens, reduced transparency, and hindered decision-making processes. The challenges most frequently faced are:

1. Manual Administrative Processes: The reliance on manual processes for student registration, course enrolment, attendance tracking, and grading has led to inefficiencies, errors, and delays in administrative tasks along with increased administrative workload.

- 2. Disparate Data Systems: Data is currently fragmented across multiple systems and spreadsheets, leading to data silos and difficulties in accessing and consolidating relevant information for decision-making. This Reduced efficiency in academic operations, results in delays and errors.
- **3. Limited Accessibility**: Students, teachers, and administrators face challenges in accessing timely and relevant information, such as class schedules, grades, and attendance records, leading to frustration and inefficiencies in academic operations.

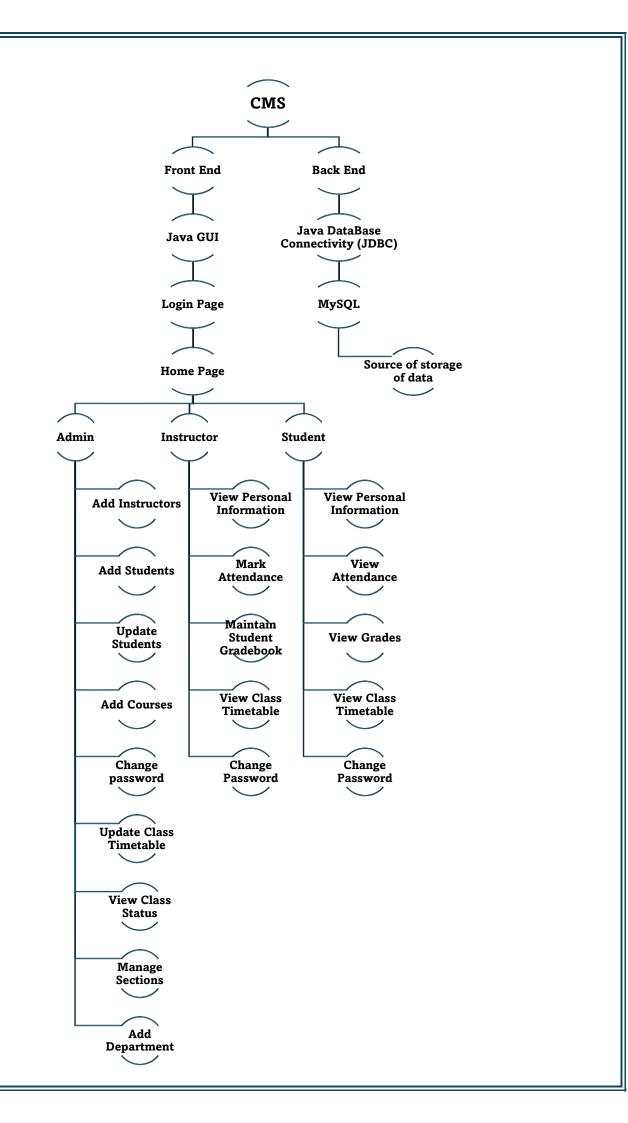
Project Scope

The CMS desktop application aims to streamline administrative tasks within educational institutions. It will provide user-friendly interfaces for administrators, instructors, and students to manage user registrations, course information, timetables, attendance records, grades, and passwords. The system will enhance communication, improve data management efficiency, and ensure a seamless experience for all users involved.

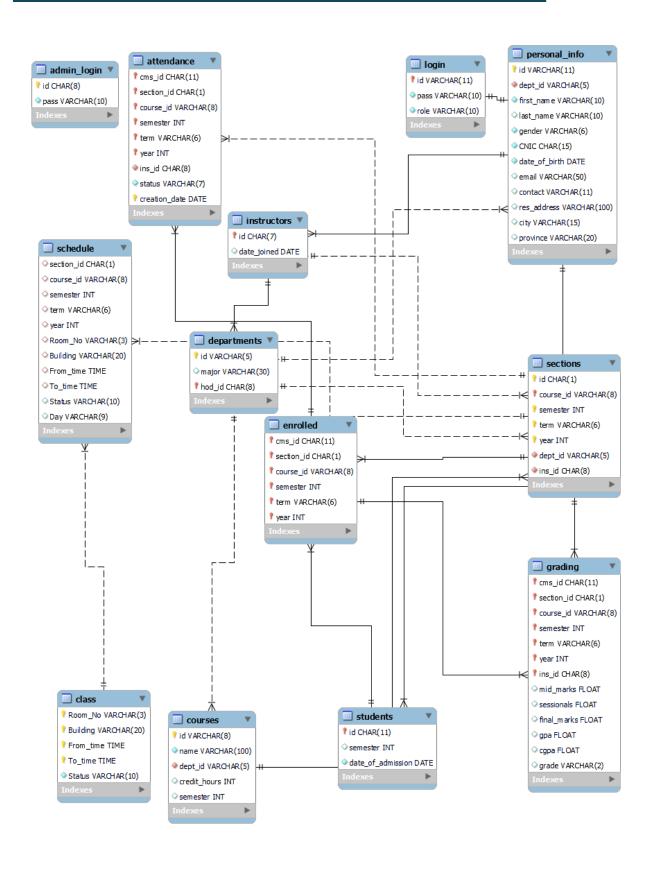
Proposed Methodology

The project is developed using the Java programming language using Netbeans IDE 18 as the Integrated Development Environment, and the JDK-20 as the version of Java Development Kit.

Overall, the proposed methodology resulted in a CMS desktop application that is engaging and pleasurable to use for users with varying degrees of expertise. With the help of Java's many frameworks and tools, we were able to create a feature-rich application with an intuitive user interface and good user experience. The system provides a smooth platform for managing educational duties and data for administrators, instructors, and students, resulting in a productive and interesting interaction for all parties involved.



Entity-Relationship Diagram



Relational Schema

```
CREATE TABLE `Admin_Login`(
  'id' CHAR(8) PRIMARY KEY,
  'pass' VARCHAR(10) NOT NULL
CREATE TABLE `Departments`(
  'id' VARCHAR(5) PRIMARY KEY,
  'major' VARCHAR(30),
  'hod id' CHAR(8),
  FOREIGN KEY('hod_id') REFERENCES 'Instructors'('id')
  ON UPDATE CASCADE
);
CREATE TABLE 'Personal Info'(
  'id' VARCHAR(11) PRIMARY KEY,
  `dept_id` VARCHAR(5) NOT NULL,
  `first_name` VARCHAR(10) NOT NULL,
  `last_name` VARCHAR(10),
  'gender' VARCHAR(6) NOT NULL,
  'CNIC' CHAR(15) NOT NULL UNIQUE,
  'date of birth' DATE NOT NULL,
  `email` VARCHAR(50) UNIQUE,
  'contact' VARCHAR(11) UNIQUE,
  `res_address` VARCHAR(100),
  `city` VARCHAR(15),
  `province` VARCHAR(20),
  FOREIGN KEY ('dept_id') REFERENCES 'Departments'('id')
  ON UPDATE CASCADE
CREATE TABLE `Instructors`(
 'id' CHAR(7) PRIMARY KEY,
  `date_joined` DATE CONSTRAINT CHECK (YEAR(`date_joined`) >=
2023),
  FOREIGN KEY ('id') REFERENCES 'Personal Info'('id')
  ON UPDATE CASCADE
CREATE TABLE `Login`(
  'id' VARCHAR(11) PRIMARY KEY,
  'pass' VARCHAR(10) NOT NULL,
  'role' VARCHAR(10) NOT NULL,
  FOREIGN KEY ('id') REFERENCES 'Personal Info'('id')
  ON UPDATE CASCADE
```

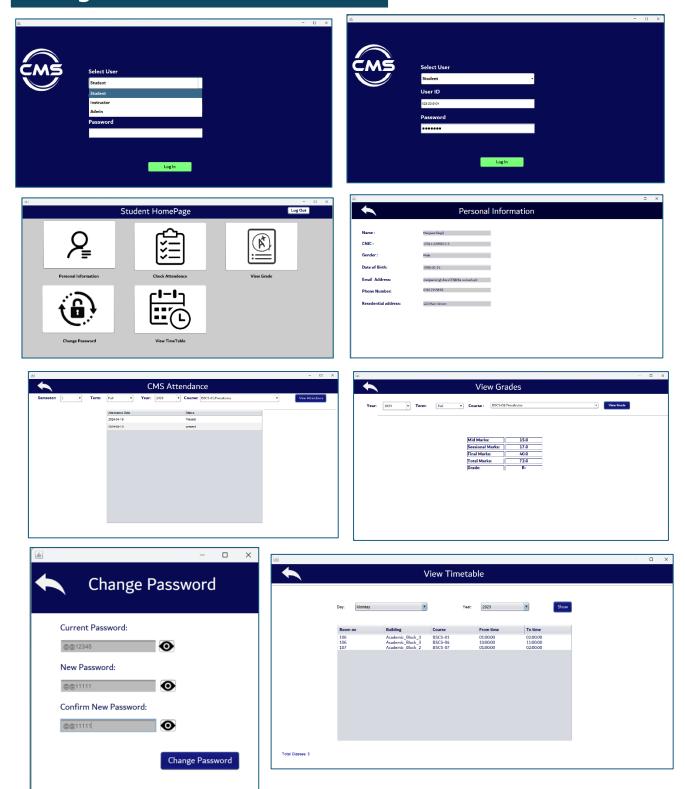
```
);
CREATE TABLE `Courses`(
  'id' VARCHAR(8) PRIMARY KEY,
  'name' VARCHAR(100) NOT NULL,
  `dept_id` VARCHAR(5) NOT NULL,
  `credit_hours` INT UNSIGNED,
     'semester' INT UNSIGNED CHECK ('semester' <= 8),
  FOREIGN KEY ('dept_id') REFERENCES 'Departments'('id')
  ON UPDATE CASCADE
CREATE TABLE `Sections` (
  'id' CHAR(1),
  `course_id` VARCHAR(8),
  `semester` INT UNSIGNED CHECK (`semester` <= 8),
  `term` VARCHAR(6) NOT NULL,
  'year' INT NOT NULL CHECK (year>=2023),
  `dept_id` VARCHAR(5) NOT NULL,
  `ins_id` CHAR(8) NOT NULL,
  PRIMARY KEY('id', 'course_id', 'semester', 'term', 'year'),
  FOREIGN KEY ('dept_id')
  REFERENCES 'Departments'('id')
  ON UPDATE CASCADE,
  FOREIGN KEY ('ins_id')
  REFERENCES 'Instructors'('id')
  ON UPDATE CASCADE,
  FOREIGN KEY ('course_id')
  REFERENCES 'Courses'('id')
  ON UPDATE CASCADE
CREATE TABLE `Students`(
  'id' CHAR(11) PRIMARY KEY,
  `semester` INT UNSIGNED CHECK (`semester` <= 8),
  'date of admission'
                                     NOT
                                               NULL
                                                           CHECK
(YEAR(`date_of_admission`) >= 2023),
  FOREIGN KEY ('id') REFERENCES 'Personal_Info'('id')
  ON UPDATE CASCADE
);
CREATE TABLE `Class` (
 `Room_No` VARCHAR(3),
 `Building` VARCHAR(20),
```

```
`From_time` TIME,
 'To time' TIME,
 `Status` VARCHAR(10) NOT NULL,
  PRIMARY KEY('Room_No', 'Building', 'From_time', 'To_time')
CREATE TABLE `Enrolled`(
  `cms_id` CHAR(11),
  `section_id` CHAR(1),
  `course id` VARCHAR(8),
  `semester` INT UNSIGNED,
  `term` VARCHAR(6),
  'year' INT NOT NULL,
  PRIMARY KEY ('cms_id', 'section_id', 'course_id', 'semester', 'term',
`year`),
  FOREIGN KEY ('cms_id') REFERENCES 'Students'('id')
  ON UPDATE CASCADE.
  FOREIGN KEY ('section_id', 'course_id', 'semester', 'term', 'year')
  REFERENCES 'Sections'('id', 'course_id', 'semester', 'term', 'year')
  ON UPDATE CASCADE
CREATE TABLE `Schedule`(
     `section_id` CHAR(1),
     `course_id` VARCHAR(8),
     `semester` INT UNSIGNED.
     `term` VARCHAR(6),
     'year' INT,
     'Room No' VARCHAR(3),
     `Building` VARCHAR(20),
     `From time` TIME,
     'To time' TIME,
     `Status` VARCHAR(10).
     `Day` VARCHAR(9),
     FOREIGN KEY ('section id', 'course id', 'semester', 'term', 'year')
     REFERENCES 'Sections'('id', 'course_id', 'semester', 'term', 'year')
     ON UPDATE CASCADE.
     FOREIGN KEY ('Room_no', 'Building', 'from_time', 'To_time')
REFERENCES 'class' ('Room_no', 'Building', 'from_time', 'To_time')
     ON UPDATE CASCADE
CREATE TABLE 'Grading'(
  cms id CHAR(11),
  `section_id` CHAR(1),
```

```
`course_id` VARCHAR(8),
  `semester` INT UNSIGNED,
  `term` VARCHAR(6),
  'year' INT NOT NULL,
  'ins_id' CHAR(8) NOT NULL,
  'mid marks' FLOAT DEFAULT 0.00.
  'sessionals' FLOAT DEFAULT 0.00.
  `final_marks` FLOAT DEFAULT 0.00,
  `gpa` FLOAT DEFAULT 0.00,
  `cgpa` FLOAT DEFAULT 0.00,
  `grade` VARCHAR(2),
  PRIMARY KEY ('cms_id', 'ins_id', 'section_id', 'course_id', 'semester',
`term`, `year`),
  FOREIGN KEY ('ins_id') REFERENCES 'Sections'('ins_id')
  ON UPDATE CASCADE,
  FOREIGN KEY ('cms_id', 'section_id', 'course_id', 'semester', 'term',
'year') REFERENCES 'Enrolled'('cms_id', 'section_id', 'course_id',
`semester`, `term`, `year`) ON UPDATE CASCADE
CREATE TABLE `Attendance`(
     `cms_id` CHAR(11) NOT NULL,
  `section_id` CHAR(1) NOT NULL,
  `course_id` VARCHAR(8) NOT NULL,
  `semester` INT UNSIGNED.
  `term` VARCHAR(6),
  'year' INT UNSIGNED,
  `ins_id` CHAR(8),
  `status` VARCHAR(7),
  `creation date` DATE,
  PRIMARY KEY('cms_id', 'section_id', 'course_id', 'semester', 'term',
`year`, `creation_date`),
  FOREIGN KEY ('cms_id', 'section_id', 'course_id', 'semester', 'term',
`year`)
                      `Enrolled`(`cms_id`, `section_id`, `course_id`,
     REFERENCES
`semester`, `term`, `year`)
     ON UPDATE CASCADE,
  FOREIGN KEY ('ins_id') REFERENCES 'Sections'('ins_id') ON UPDATE
CASCADE
```

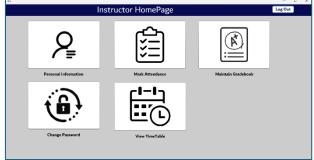
Below are the screenshots in the order as student login, instructor login and admin login.

Project Screenshots



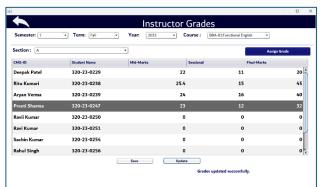
The change password, personal information, and view timetable pages layout is same for the instructor as for the students; thus, these screenshots are not attached again for instructor.







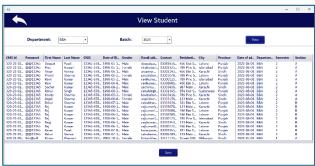




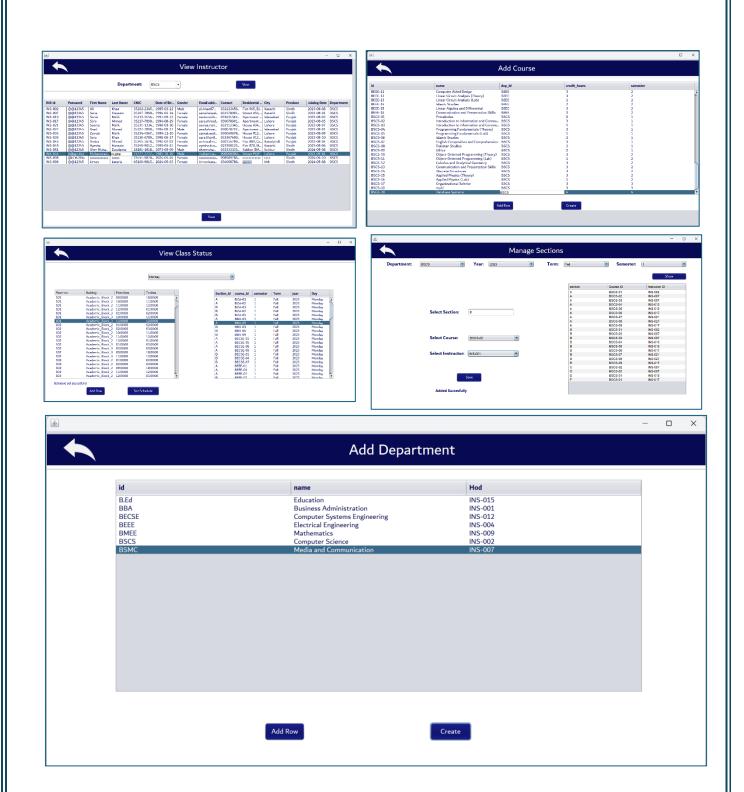












The change password page is same for the admin as for the student and for the instructor, thus the screenshot is not attached.

Major Outcomes

The major outcomes of implementing a content management system can have far-reaching benefits for educational institutions, students, teachers, administrators, and other stakeholders. Here are some of the key outcomes:

- 1. Reduced Administrative Workload
- 2. Always up-to-date information (leading to better decision making)
- 3. Increased Accessibility for Students, teachers, and administrators
- 4. Enhanced Student Experience
- 5. Enhanced Teacher Experience
- 6. Enhanced Administrator Experience
- **7.** Better Resource Utilization of campus facilities (resulting in cost-saving and improved efficiency)
- 8. Increased Institutional Reputation

Conclusion

In conclusion, this desktop application is going to be very helpful for the users; Admins, Teachers, and Students, as it provides a simple user experience and in based on the principles of Object-Oriented Programming. The project will assist users in making calculations, managing time, and handling data.