



Tribhuvan University

Faculty of Humanities and Social Sciences

A Project Report on

College Management System

Submitted to

Department of Computer Applications

Nepal Kasthamandap College

In the partial fulfillment of the requirement for bachelor's in computer applications

Submitted by

Suyog Luitel (6-2-144-27-2019)

Urmila Thakuri (6-2-144-28-2019)

Under the supervision of

Binaya Subedi



Tribhuvan University

Faculty of Humanities and Social Sciences

Nepal Kasthamandap College

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision of Suyog Luitel and Urmila Thakuri entitled “**College Management System**” in partial fulfillment of the requirement for bachelor's in computer applications is recommended for final evaluation.

Supervisor

Binaya Subedi

Nepal Kasthamandap College



Tribhuvan University

Faculty of Humanities and Social Sciences

Nepal Kasthamandap College

Letter of Approval

This is to certify that this project prepared by Suyog Luitel and Urmila Thakuri entitled “College Management System” in the partial fulfillment of the requirement for the bachelor’s in computer application has been evaluated. In our opinion it is satisfactory in scope and quality for the required degree.

<hr/> Supervisor Binaya Subedi Nepal Kasthamandap College	<hr/> Coordinator Sujan Poudel Nepal Kasthamandap College
<hr/> Internal Examiner	<hr/> External Examiner Prof. Dr. Subarna Shakya

Abstract

The College Management System is a comprehensive application solution designed to streamline and enhance the administrative and academic operations of educational institutions. This system leverages cutting-edge technology to facilitate efficient management of student information. It offers a user-friendly interface for students, faculty, and administrators, enabling easy access to academic records, attendance tracking, and academic records. Additionally, it automates tasks such as enrollment, assigning teachers to the subjects, and report generations, reducing manual effort and minimizing effort.

Acknowledgement

The project “College Management Application” would not have been possible without the joint efforts of many individuals. We would like to express our sincere gratitude to all those who have contributed to the successful completion of this project.

We would like to extend our sincere gratitude and thanks to project supervisor Er. Binaya Subedi, and Coordinator Er. Sujan Poudel for the support, valuable suggestions and guidance who made this project to reach this level and the management of Nepal Kasthamandap College for providing all the guidelines, reference books for the development of this project.

Additionally, we would like to acknowledge the inspiration and encouragement we received from family and friends.

Table of Contents

Abstract	iii
Acknowledgement	iv
Chapter 1 Introduction	1
1.1 Introduction	1
1.2 Problem Statement	1
1.3 Objectives	2
1.4 Scope and Limitations	2
1.5 Development Methodology	3
1.6 Report Organization	5
Chapter 2 Background Studies and Literature Review	6
2.1 Background Studies	6
2.2 Literature Review	6
Chapter 3 System Analysis and Design	8
3.1 System Analysis	8
3.1.1 Requirement Analysis	8
3.1.2 Feasibility Study	12
3.1.3 Progress Diagram	15
3.1.4 State Diagram	16
3.1.5 Sequence Diagram	17
3.2 System Design	18
3.2.1 Refinement of Classes and Objects	18
3.2.2 Deployment Diagram	19
Chapter 4 Implementation and Testing	20
4.1 Implementation	20
4.1.1 Tools used:	20
4.2 Testing	20

4.2.1 Test case for unit testing.....	20
Chapter 5 Conclusion and Future Recommendations.....	23
5.1 Conclusion	23
5.2 Outcome.....	23
5.3 Future Recommendation.....	23
References.....	24
Appendices.....	25
Source code.....	30

List of Figures

Figure 1.1: Agile Methodology.....	3
Figure 3.1: Use Case Diagram	9
Figure 3.2: Progress Diagram	15
Figure 3.3: State Diagram	16
Figure 3.4: Sequence Diagram.....	17
Figure 3.5: Descriptive Dom model of CMS	18
Figure 3.6: Deployment Diagram	19

List of Tables

Table 1.1: Outline of the Report	5
Table 3.1: Use Case for Login	10
Table 3.2: Use Case for Adding Courses	10
Table 3.3: Use Case for Adding Attendance	11
Table 3.4: Use Case for Adding Marks	11
Table 3.5: Time Schedule	14
Table 4.1: Test case for Login	21
Table 4.2: Test case for Adding Course.....	21
Table 4.3: Test case for Adding Marks.....	22

List of abbreviations used:

CMS = College Management System

UML = Unified Modeling Language

DOM = Document Object Model

IDE = Integrated Development Environment

XML = Extensible Markup Language

UI = User Interface

Chapter 1 Introduction

1.1 Introduction

A college management system is a desktop application designed to help colleges and higher education institutes to manage the different activities efficiently. It has the perspective of keeping the information sure and management easier. The college management system provides an environment where one can manipulate data and information about students and faculty members. This software provides a user-friendly interface for the user, and it reduces redundancy. The amazing thing while using the software is that centralized information is available which can be accessed by numbers of users. In manual and file system of college management it is difficult and very time-consuming process which involves record keeping of student, teachers, staffs, salary, attendance, transaction and so on. The institute admin is fully dependent on staff to manage the records that too with a big chance of error. So, to make those daily activities fast, relevant, and easy this software replaces the manual task.

Basically, a college management system is an attempt to create a computer-based management system that will replace the manual work being done with a lot of time daily. This software is easy to use and fast. This system is developed by keeping general requirements of education institutions.

1.2 Problem Statement

The College Management System seeks to overcome manual administrative tasks, disconnected communication channels, and inefficient course/exam management by providing a centralized, user-friendly platform. It aims to streamline student records, attendance, and faculty information while enabling efficient course allocation, and result processing. The system ensures easy accessibility to academic and administrative information, enhancing overall college operations and performance. This project mainly deals with Student and Faculty Profile, Marksheet, Attendance Management which can be managed by the Admin.

1.3 Objectives

The main objectives of this project are as follows:

- To eradicate the problem of traditional management systems with the software-based college management system.
- To record the information of students based on their course and subjects.
- To record the information of faculty members and assign them their subjects.
- To record the attendance of students based on subjects.
- To record the marks obtained by the students and display the result.

1.4 Scope and Limitations

The scope of the project deals with streamline, automation, and impasse a wide range of functionalities and modules that assist in managing student information, academic records, admission, attendance, admissions, fees, and other related tasks. Also, it provides a centralized platform for storing and accessing data to generate reports, attendance, etc.

Some of the limitations of this project are:

- It is only available on a desktop application.
- Students and faculty members cannot sign up themselves.
- Only students' attendance is recorded.
- Admin must record all the students and faculty members.

1.5 Development Methodology

These selected projects we are going to do build have frequent changes and an uncertain and turbulent environment. So, to build the project, we will be using agile methodology.

Agile methodology is chosen for projects due to its flexibility, customer-centric approach, and ability to deliver faster and adapt to changing requirements. By breaking the project into smaller iterations, Agile ensures quicker delivery of usable features, reducing time-to-market. Regular feedback from stakeholders ensures that the final product aligns with customer expectations. Agile emphasis on open communication, transparency, and quality focus fosters collaboration, risk mitigation, and continuous improvement. Empowered teams and reduced waste lead to higher productivity and better outcomes. Overall, Agile adaptability to change and its emphasis on delivering value make it a popular choice for various software development and business projects. [1]

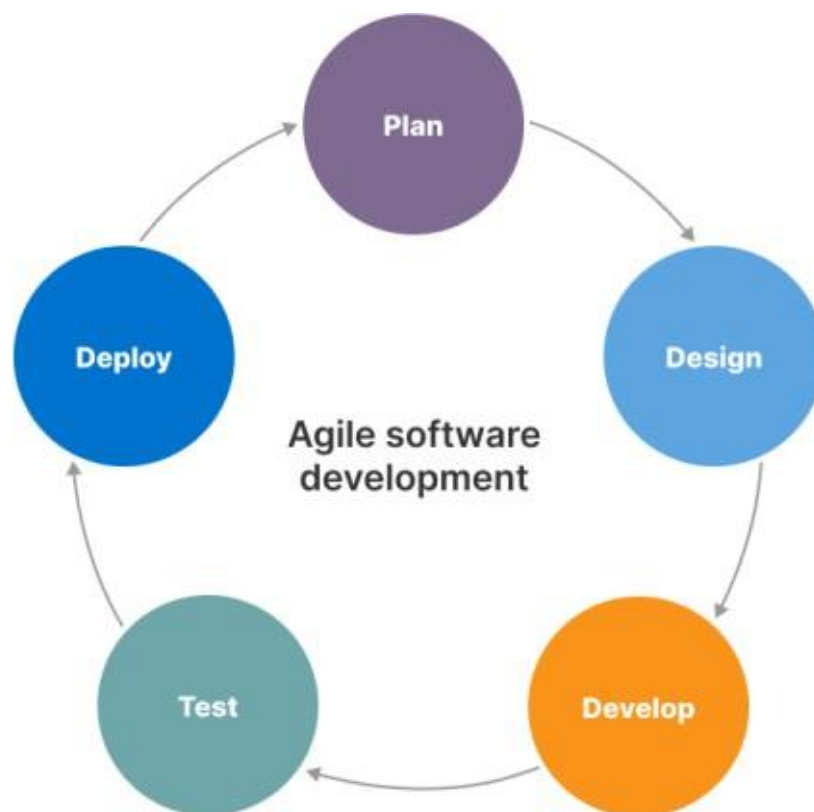


Figure 1.1: Agile Methodology

Agile methodology is a project management approach that emphasizes flexibility, collaboration, and customer satisfaction. It is an iterative and incremental approach that focuses on delivering a working product in short periods of time, typically in two to four weeks. The following represents our agility implementation for the selected project:

Planning: We determine the work of our project that we needed to accomplish during a specific time as in the Gantt chart we made. Then, we define the overall objectives and scope of our project. We also breakdown the project into smaller iterations with specific goals and deliveries.

Design: We conducted user research and analysis to understand the needs and preferences of the systems users. Then collaborate to gather feedback and refine the designs. Documentation of architectural and technical design of the system, including databases, APIs and integrations points were also done in this stage.

Development: Implementation of the functionalities was done on this page. We adopt an iterative and incremental development approach focusing on delivering value. We divided the development tasks among the members and established a coding and documentation style. Afterward, we integrate code changes into a version control system to ensure code quality and collaboration.

Testing: In this phase, we develop and execute test cases and scenarios to validate the functionality and performance of the system. We ensure proper documentation of test cases, test results, and any bug fixes or enhancement made during testing.

Deployment: In this last stage, we prepare the system for deployment by configuring servers, databases, and necessary infrastructure. Then we monitor the system's performance and address any issues that arise during the deployment phase

1.6 Report Organization

Table 1.1: Outline of the Report

Introduction	It describes an overview of the College Management System along with its objectives, problem statements, scope and limitations and development methodology being used.
Background Study	It studies about the limitations of the manual system of present CMS working procedures and need of management system.
System Analysis	It describes the functional and non-functional requirements along with feasibility analysis of the CMS.
Implementation and Testing	It describes about the tools used in the system and the testing that are done.
Conclusion	It describes about the outcome of this system as well as the future recommendations for the CMS.

Chapter 2 Background Studies and Literature Review

2.1 Background Studies

A college management system is a comprehensive software platform designed to streamline and automate various administrative and academic processes within an educational institution. It serves as the central hub for managing student information, enrollment, attendance, grading, scheduling, and faculty records, among other essential functions. This system not only enhances the efficiency of administrative tasks but also provides valuable insights through data analysis to support informed decision-making. Additionally, it often includes features for communication between students, faculty, and administrators, facilitating a seamless exchange of information. The implementation of a college management system is crucial in modern educational institutions to ensure smooth operations, improve transparency, and ultimately enhance the overall learning experience for students. [2]

2.2 Literature Review

College Management Systems (CMS) have gained significant attention in recent years due to the increasing complexity of administrative and academic processes in higher education institutions. A comprehensive literature review reveals that CMS plays a vital role in optimizing the management of colleges and universities. These systems are designed to automate various tasks, including student information management, attendance tracking, fee collection, and academic record-keeping. The primary objective of CMS is to enhance efficiency, accuracy, and communication within educational institutions, ultimately improving the overall educational experience for students and the administrative processes for staff. [3]

In examining the literature on CMS, it becomes evident that these systems comprise several key features and components. Student information management is at the core, maintaining records of students' personal details, academic progress, and enrollment history. Attendance management tracks student presence in classes, ensuring compliance with academic requirements. Fee management automates the collection and tracking of student fees, streamlining financial operations. Timetable management simplifies the

scheduling of classes and other activities. Additionally, CMS often includes modules for grading, transcript generation, library management, and communication tools, providing a comprehensive solution for educational institutions. [4]

Numerous studies highlight the benefits and challenges associated with CMS implementation. Benefits include enhanced efficiency, data accuracy, and security, which contribute to streamlined administrative processes. Improved communication tools strengthen collaboration among stakeholders, such as students, faculty, parents, and administrators. CMS also empowers data-driven decision-making through robust reporting and analytics. However, challenges persist, including concerns about data security and privacy, the complexity of integrating CMS with other systems like Learning Management Systems (LMS), the need for ongoing maintenance and updates, and the necessity of user training and adoption. [5]

The literature suggests that the evolution of CMS is ongoing, with emerging trends and future directions. Integration with artificial intelligence (AI) and machine learning for predictive analytics is gaining traction, allowing institutions to anticipate student needs and optimize resource allocation. Enhanced mobile accessibility ensures that users can access CMS from any device, increasing convenience and usability. A growing emphasis on data analytics is expected to drive improvements in student outcomes and institutional effectiveness. Customization and scalability are becoming essential as CMS providers cater to institutions of various sizes and needs, ensuring that these systems remain adaptable and relevant in the dynamic landscape of higher education management. As institutions continue to prioritize efficiency and data-driven decision-making, CMS will likely play an increasingly central role in their operations. [6]

Chapter 3 System Analysis and Design

3.1 System Analysis

It is the process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is the process of examining and evaluating an existing system or proposed system to determine its components, inputs, outputs, and the interactions between them. The goal of system analysis is to understand the problem domain, identify areas for improvement, and propose solutions to optimize the system's performance.

System analysis is a critical component of software development, as it allows developers to identify and address potential problems before the system is built. It is often conducted in collaboration with stakeholders, such as end-users and managers, to ensure that the system meets their needs and requirements. [7]

3.1.1 Requirement Analysis

Requirement analysis is a process used to determine the needs and expectations of the CMS. It involves frequent communication to define expectations, resolve conflicts and document all the key requirements.

Functional Requirements:

A functional requirement is a description of the service that the CMS must offer. It is a statement of services the system should provide, how the system should react to inputs and how the system should behave in situations.

Use Case Diagram:

Use case diagrams are considered for high level requirements analysis. The purpose of a use case diagram is to capture the dynamic aspect of a system. It is used to gather the requirements of a system including internal and external influences. It is outside view of a system. [8]

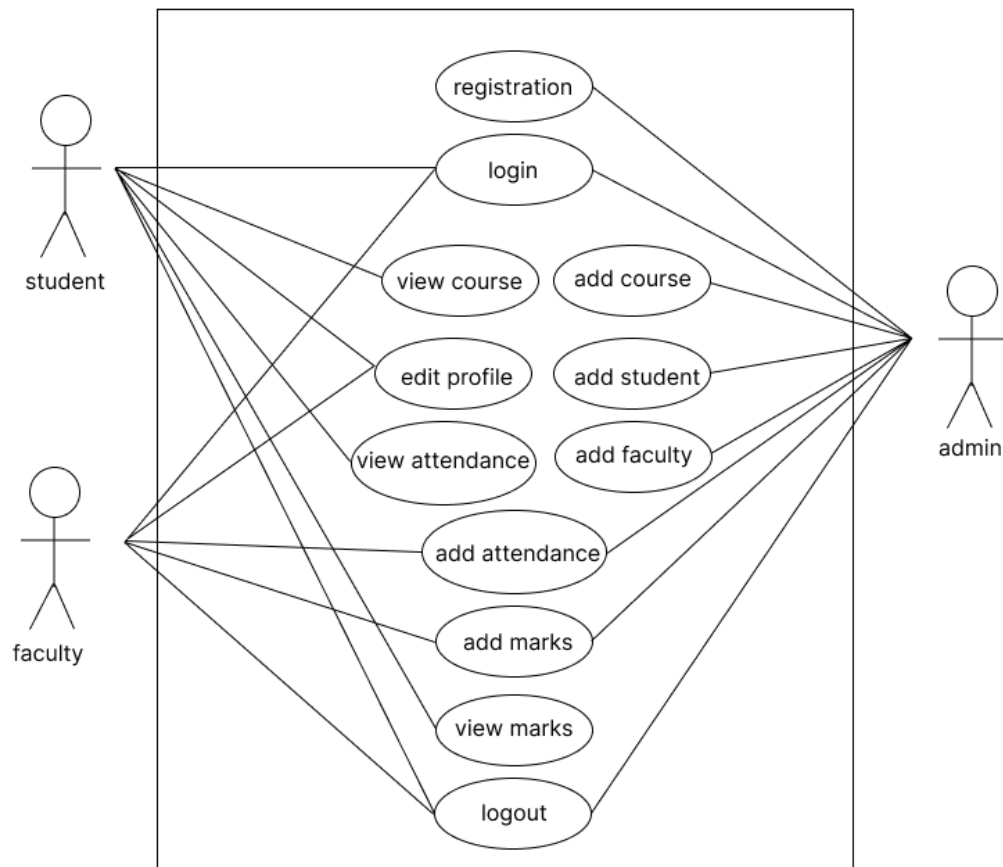


Figure 3.1: Use Case Diagram

In the above use case diagram, there are three actors: admin, student, and faculty where all of them will be able to login and logout from the system. Admin can add courses, students, and faculty. Both admin and faculty can add attendance and marks of students. Students can view all the details entered by admin. Similarly, both students and faculty can update their profile.

Table 3.1: Use Case for Login

Use Case Identifier	UC1: Login
Primary Actor	Admin
Secondary Actor	The Student, Faculty should be able to login to the system.
Success Scenario	<ol style="list-style-type: none">1. The user is logged in.2. The user is acknowledged.3. Redirected to respective pages.
Failure Scenario	<ol style="list-style-type: none">1. The user is not logged in.2. Users are not acknowledged.3. Flash error message.

Table 3.2: Use Case for Adding Courses

Use Case Identifier	UC1: Add Courses
Primary Actor	admin
Secondary Actor	The admin should be able to add a new course.
Success Scenario	<ol style="list-style-type: none">1. The course is added to the database.2. The admin is acknowledged.3. Redirected to all course lists.
Failure Scenario	<ol style="list-style-type: none">1. The database is not updated.2. The admin is not acknowledged.3. Flash error message.

Table 3.3: Use Case for Adding Attendance

Use Case Identifier	UC1: Add Attendance
Primary Actor	admin
Secondary Actor	The logged in faculty should be able to update the Attendance.
Success Scenario	4. Attendance is added to the database. 5. The admin is acknowledged. 6. Redirected to all Attendance lists.
Failure Scenario	4. The database is not updated. 5. The admin is not acknowledged. 6. Flash error message.

Table 3.4: Use Case for Adding Marks

Use Case Identifier	UC1: Add marks
Primary Actor	admin
Secondary Actor	The logged in faculty should be able to update the marks.
Success Scenario	7. Marks are added to the database. 8. The admin is acknowledged. 9. Redirected to all mark's lists.
Failure Scenario	7. The database is not updated. 8. The admin is not acknowledged. 9. Flash error message.

Non-Functional Requirements:

The performance of the CMS will highly depend on the performance of the hardware and software components of the installed devices.

Performance:

This system is designed for smooth performance with optimization and good response even for the low-end devices.

Security:

In this system only authorized personnel can gain access to the admin panel and only valid users with valid username and password can access the user dashboard.

Availability:

The system is designed to be available 24*7. In case of major malfunctions, the system will be repaired quickly so that the business is not severely affected.

Ease of Use:

Users with a simple level of understanding of how sites work can easily use this system as it is built with that in focus.

3.1.2 Feasibility Study

After making an initial investigation, a feasibility study is carried out to check the workability of the system. Feasibility study is the study of a system proposed according to specification.

Technical Feasibility

Technical Feasibility determines whether the work for the project can be done with existing equipment, software technology and available personnel. The proposed system may run on any machine supporting window devices and can work on the best software and hardware that had been used while designing application so it may be feasible in all technical terms. Thus, the project is technically feasible.

Operational Feasibility

Project are beneficial only if they can be turned out into information system and they will meet the organization's operational requirements. It will check the application can be used and works properly if it is being developed and implemented. User interface is designed to make it user friendly. Any user with simple knowledge can operate this system. The well-planned design will ensure the optimal utilization of mobile resources and will help in the improvement of performance status. Therefore, this project is operationally feasible.

Economic Feasibility

Economic Feasibility decides whether the necessary application can generate financial profits for an organization. The cost for the proposed system does not exceed the cost and efforts involved in maintaining any communication. The development cost for the application satisfies the organization. The application also reduces the administrative and technical effort to do various jobs that a single application can do. So, this project is economically feasible.

Schedule Feasibility

This criteria model is parallel goes with the schedule time. The ability to finish the project on time is measured under this model. The CMS system has been completed on the scheduled time where more priority is given to customers' satisfaction.

Table 3.5: Time Schedule

	Gantt Chart								
	20-Mar	1-Apr	16-Apr	1-May	16-May	1-Jun	16-Jun	1-Jul	20-Jul
Planning									
Research									
Design									
Implementation									
Usability Testing									
Documentaiton									

3.1.3 Progress Diagram

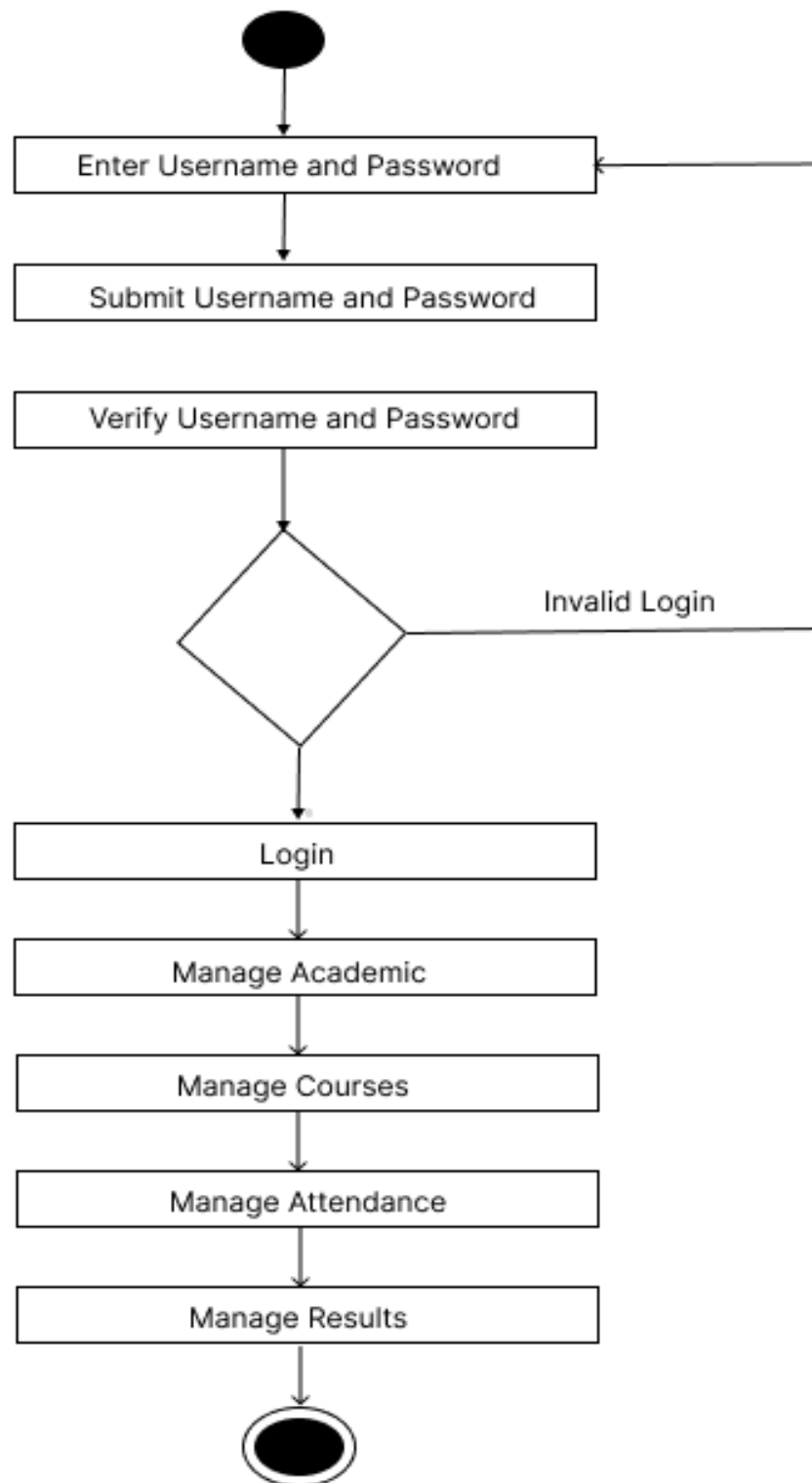


Figure 3.2: Progress Diagram

3.1.4 State Diagram

A state diagram of CMS is used to describe the different stages of a system or a section of a system at a specific point in time. It states the behavior of the state of the application. All steps have their own behavior as function. It is used to denote the behavior of a class but not every class is true.

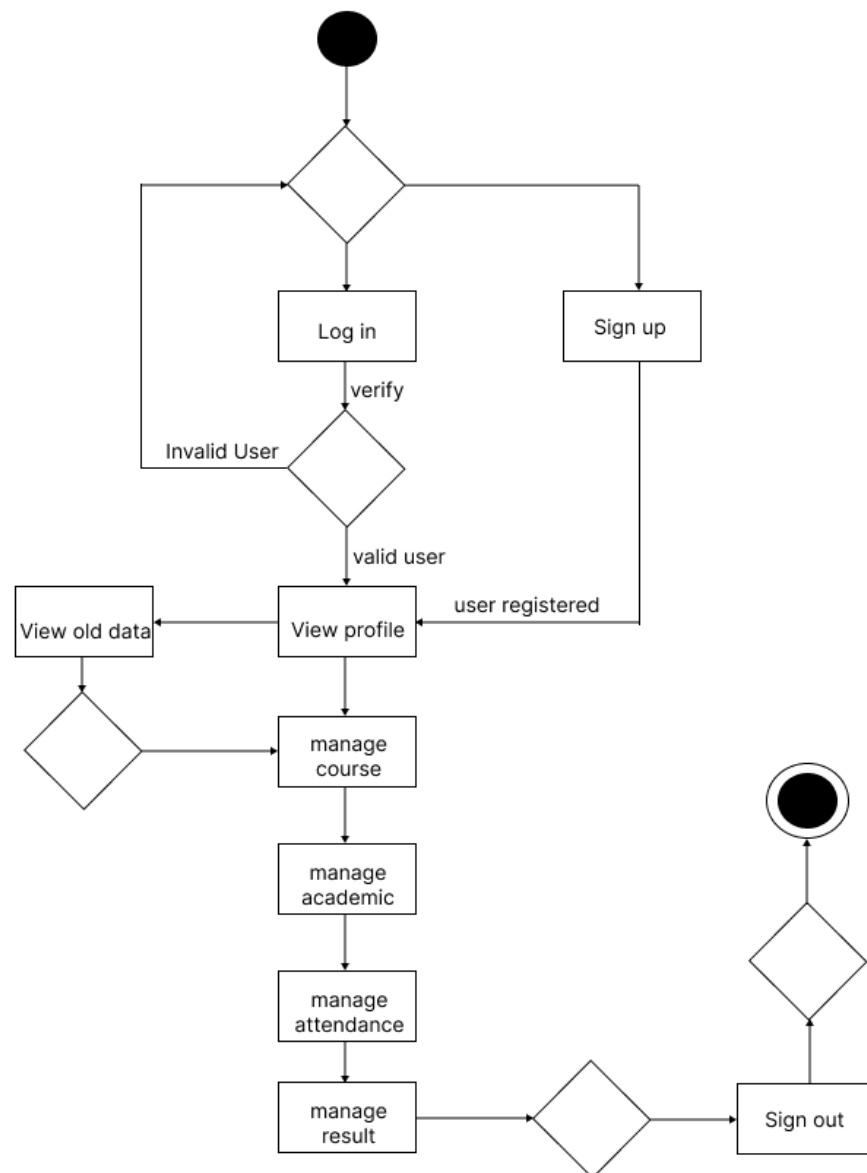


Figure 3.3: State Diagram

3.1.5 Sequence Diagram

The connection between admin, academic which includes both student and faculty, CMS system and database is shown below on the one the sequence diagram. It describes the flow of the application and how the operations are carried out inside the application. The events that occur inside the application are illustrated and presented in the sequence diagram.

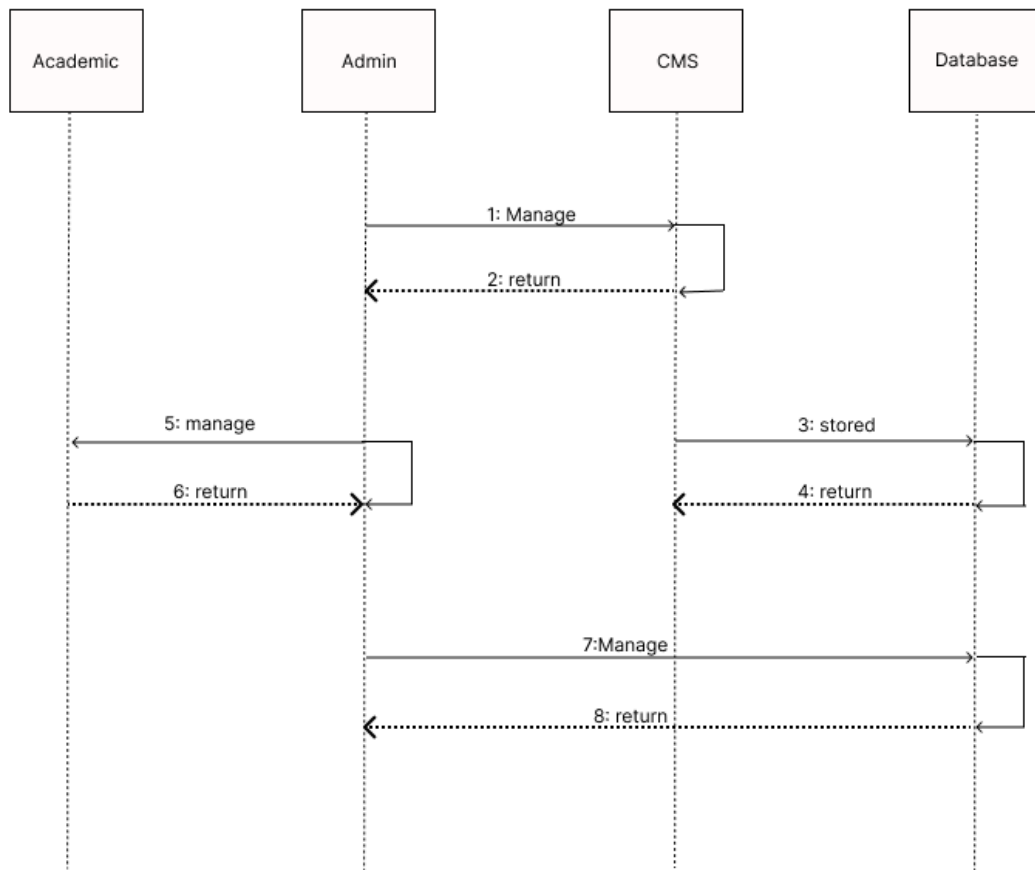


Figure 3.4: Sequence Diagram

3.2 System Design

3.2.1 Refinement of Classes and Objects

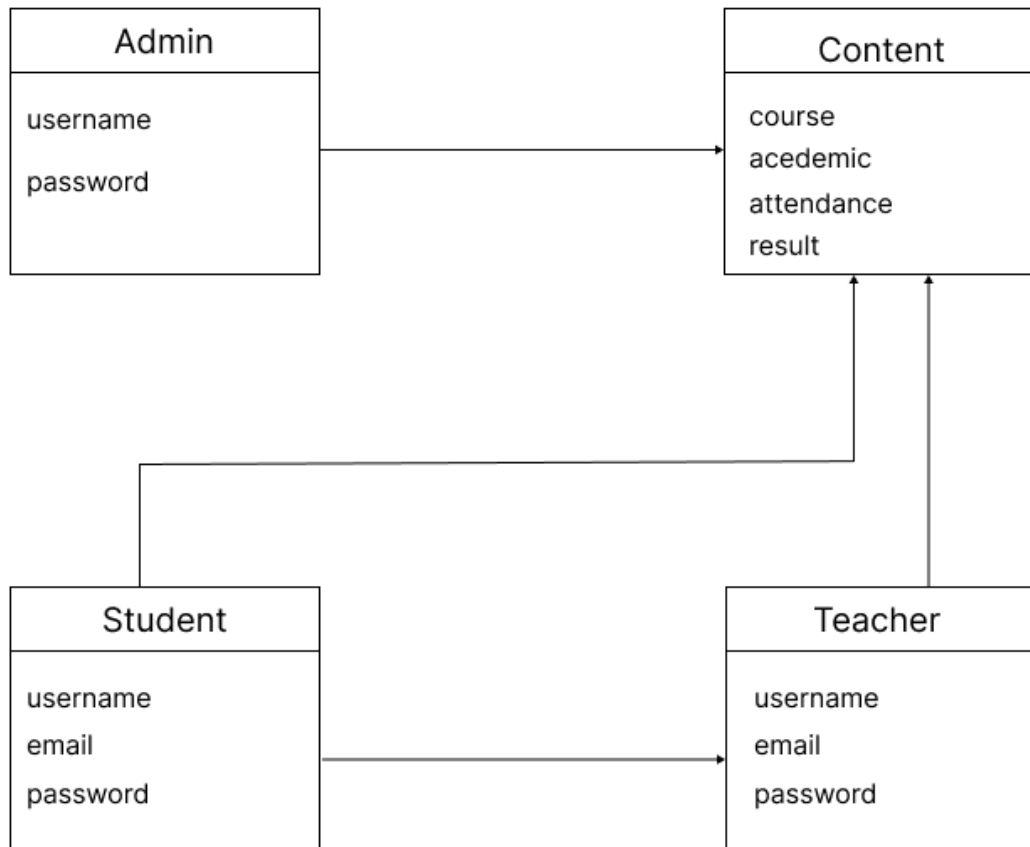


Figure 3.5: Descriptive Dom model of CMS

3.2.2 Deployment Diagram

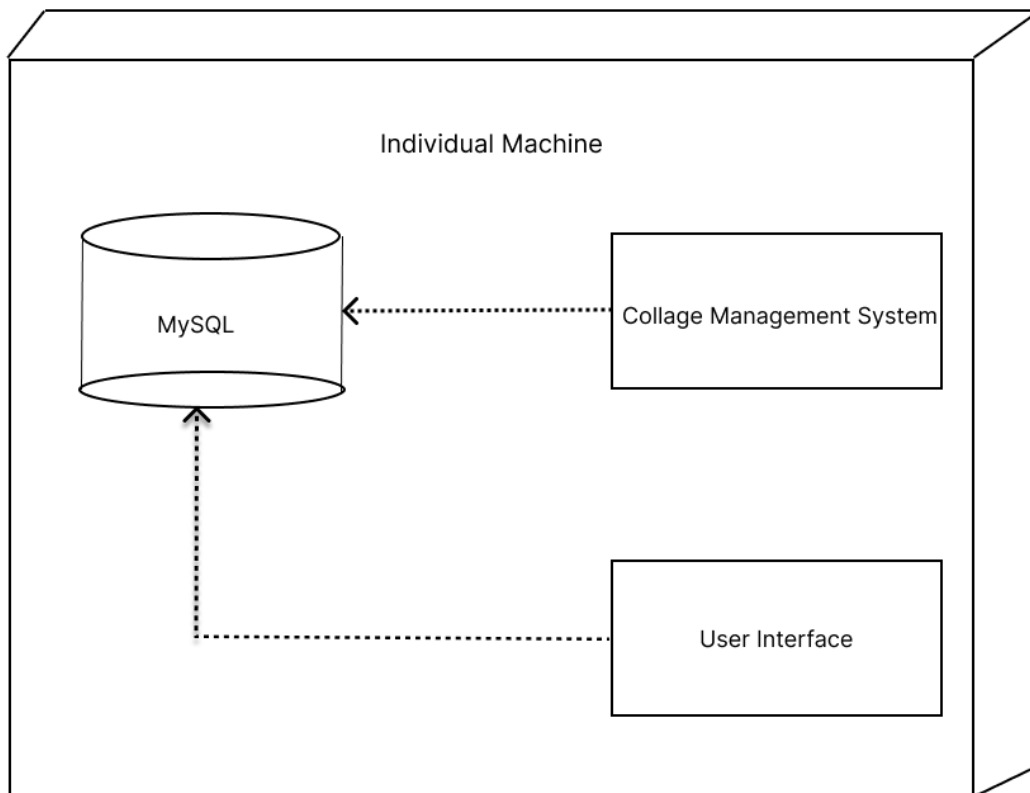


Figure 3.6: Deployment Diagram

Chapter 4 Implementation and Testing

4.1 Implementation

In this stage, physical system specifications are converted into a working and reliable solution. There is where the system is developed. On receiving the system design document, the work is divided into modules/units and actual coding is started. It is followed by testing. Several tools are used in this phase of software development.

4.1.1 Tools used:

Front end: Java is a widely used, versatile, and object-oriented programming language that is known for its portability and platform independence. It is commonly used for developing a wide range of software applications, from web and mobile applications to desktop software, scientific applications, and more.

Back end: MySQL is an open-source relational database management system (RDBMS) that is widely used for managing and storing data in various applications, from web-based systems to enterprise-level databases.

IDE: NetBeans is an integrated development environment (IDE) primarily used for Java development, although it supports multiple programming languages. It provides a range of features and tools for software development, making it a popular choice among developers.

4.2 Testing

4.2.1 Test case for unit testing

Unit testing refers to the testing of every small modular component of the system, keeping them isolated from other modules. In this test phase, the errors were corrected, and modifications made as and where required. Steps followed in performing unit tests:

- Conduct the code execution tests.
- Identify and resolve any error.
- Determine that the test is complete.

Table 4.1: Test case for Login

Pre-conditions: User is already registered in the application.					
Step	Test Step	Test Data	Expected Result	Actual Result	Status
1	Visit the Login Panel		Login Panel opened	as expected,	Pass
2	Enter username and password	Username = admin Password = admin	Verified credentials detail are entered	as expected,	Pass
4	Click Login Button		User should be redirected to the application	as expected,	Pass
Post-condition: User is validated with database and logged in to the system.					

Table 4.2: Test case for Adding Course

Pre-conditions: User is authorized and logged in.					
Step	Test Step	Test Data	Expected Result	Actual Result	Status
1	Navigate to add course		Add course page should open	as expected,	Pass
2	Enter the credential required	Course code = CACS 101 Course name = BCA Sem/year = Sem No. of Sem = 8	Course added successfully	as expected,	Pass

Post-condition: Course is added to the course list, and database is updated.

Table 4.3: Test case for Adding Marks

Pre-conditions: User is authorized and logged in.					
Step	Test Step	Test Data	Expected Result	Actual Result	Status
1	Navigate to mark attendance		Add attendance page should open	as expected,	Pass
2	Enter the credential required	Select course = BCA Select Sem = Sem 6 Select subject = Mobile Programming	Student should be fetched	as expected,	Pass
3	Enter the marks	Theory Marks = 40 Practical Marks = 35	Various credentials are entered	as expected,	Pass
4	Click submit		Marks are added to the database	as expected,	pass
Post-condition: Marks are added, and database is updated.					

Chapter 5 Conclusion and Future Recommendations

5.1 Conclusion

The project entitled “College Management System” is the system that deals with the issues related to a particular institution. This project was successfully implemented with all the features mentioned in system requirements specification. The application provides appropriate information to users regarding the chosen service. The project was designed to keep in view the day-to-day problems faced by a college.

Deployment of our application certainly helps the college to reduce unnecessary wastage of time by personally going to each department for some information. Awareness and the right information about any college is essential for both the development of student as well as faculty. So, this serves the right purpose in achieving the desired requirements of both the communities.

5.2 Outcome

After the completion of our project, all the requirements and operations are implemented properly. The system is ready to run and provide some outcome as:

- The admin can login or signup into the application.
- The admin can add and update students and faculty members.
- The admin and faculty can add marks of student and declare result.
- The admin can record the attendance of students.

5.3 Future Recommendation

The project can satisfy only a certain number of users. There are still some improvements that need to be made. Some of the improvement for future application can be:

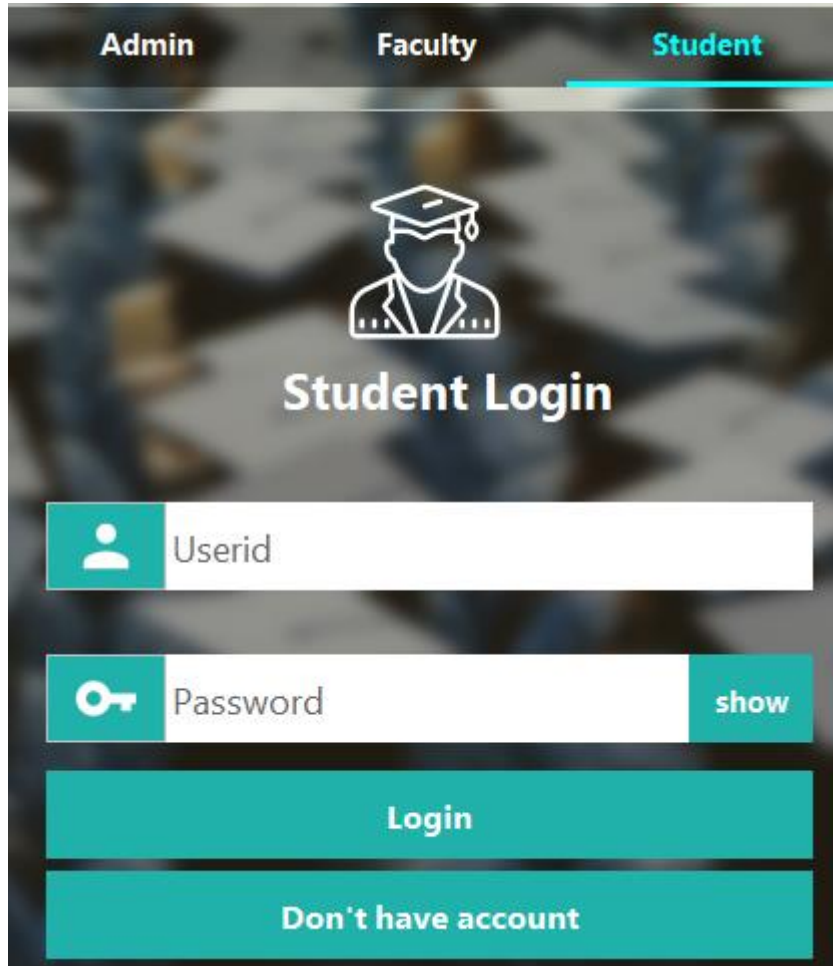
- Available on other devices such as android, iPhone and webpage.
- Teacher’s attendance.
- Payment gateway for fees.
- Generating ID cards.

References

- [1] J. A. & J. L. M. Smith, "Agile Methodology in Software Development: A Comprehensive Review," *Journal of Software Engineering*, pp. 123-145, 2021.
- [2] J. A. Smith, *Modernizing Higher Education: College Management Systems in the Digital Age*, Academic Press, 2020.
- [3] A. B. & S. C. D. Johnson, "Introduction to College Management Systems (CMS): A Comprehensive Overview," *Journal of Educational Administration*, pp. 78-92, 2021.
- [4] E. F. & W. H. G. Brown, "Key Features and Components of College Management Systems (CMS): A Comparative Analysis," *Educational Technology Research Journal*, pp. 215-230, 2022.
- [5] L. M. & J. P. A. Smith, "Benefits and Challenges of Implementing College Management Systems (CMS): An Institutional Perspective," *Journal of Higher Education Management*, pp. 45-62, 2023.
- [6] R. J. & M. S. L. Adams, "Future Trends and Directions in College Management Systems (CMS): A Prospective Analysis," *Journal of Educational Technology Advancements*, pp. 321-336, 2022.
- [7] A. W. B. H. & T. D. Dennis, *Systems Analysis and Design: An Object-Oriented Approach with UML*. Wiley, 2015.
- [8] I. C. M. J. P. & Ö. G. Jacobson, *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1992.

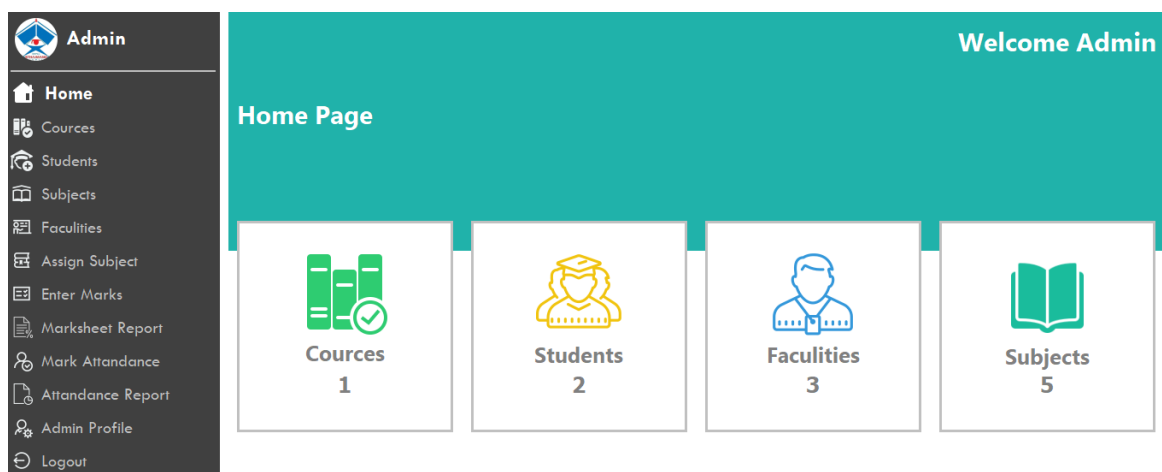
Appendices

Login Page



The login page features a header with three tabs: 'Admin', 'Faculty', and 'Student'. The 'Student' tab is selected and highlighted with a teal underline. Below the header is a large background image of a graduation cap. In the center, there is a white outline icon of a person wearing a graduation cap. Below the icon, the text 'Student Login' is displayed in a large, bold, white font. Underneath the title, there are two input fields. The first field is labeled 'Userid' and has a teal icon of a person. The second field is labeled 'Password' and has a teal icon of a key. To the right of the password field is a teal button labeled 'show'. Below the input fields is a large teal button labeled 'Login'. At the bottom, there is a teal button labeled 'Don't have account'.

Dashboard



The dashboard is divided into two main sections. On the left is a dark grey sidebar with a white header containing a blue and white icon and the text 'Admin'. Below the header is a list of menu items with icons: Home, Courses, Students, Subjects, Faculties, Assign Subject, Enter Marks, Marksheet Report, Mark Attendance, Attendance Report, Admin Profile, and Logout. The main content area has a teal header with the text 'Welcome Admin' on the right. Below the header is a large teal box with the text 'Home Page' on the left. At the bottom of the main content area are four white boxes, each containing an icon and text: 'Courses 1' (green icon of three books), 'Students 2' (yellow icon of a person with a graduation cap), 'Faculties 3' (blue icon of a person), and 'Subjects 5' (green icon of an open book).

Add Courses:

×

Add New Course

Course Code

Course Name

Sem/Year

---Select Sem/Year---

▼

Total Sem/Year

Add Course

All Courses					
					Add Course
Index no.	Course Code	Course Name	Subjects	Students	Total Sem/Year
1	101	BCA	5	2	8 sem

Add Student

×

Add New Student

---Select Course---

Roll Number

First Name

Last Name

Email ID

Contact Number

Date of Birth

03-10-2023

Date Of Birth

---Select Gender---

District


City

Father Name

Father Occupation

Mother Name

Mother Occupation



Photo

Choose File

No file choosen

Image size < 1024 KB


Add Student

All Students

Add Student

Class	Roll Number	Student Name	Course Name	Sem/Year
101	1	Abish Thapa	BCA	sem-6
101	2	Roj Gopali	BCA	sem-6

Add Faculty

×

Add New Faculty

Faculty ID

104

Faculty Name

District

City

Email ID

Contact Number

Qualification


Experience

Date of Birth

03-10-2023

Gender

---Select Gender---



Photo

Choose File

No File Chosen

Imagesize < 1024 KB

Add Faculty

All Faculties

Add Faculty

Faculty ID	Faculty Name	Email ID	Qualification	Experience
101	Binaya Subedi	binayasubedi@gmail.com	Engineer	5 years
102	Bikash Acharya	bikashacharya@gmail.com	Engineer	4 years
103	Shyam Maharjan	shyammaharjan@gmail.com	Master	5 years

Add Marks

Enter Student Marks

Theory Marks

Practical Marks

Select Course : BCA

Select Sem/Year : Semester 6

Select Subject : Mobile Programming

Roll Number	Student Name	Subject Name	Max Theory Marks	Theory Marks
1	Abish	Mobile Programming	60	40
2	Roj	Mobile Programming	60	40

Submit Marks

Add Attendance

Mark Attendance

Select Date : 03-10-2023

Course Name : BCA

Semester or Years : Semester 6

Select Subject : Mobile Programming

Fetch Students

Roll Number	Student Name	Subject Code	Course	Sem/Year	Attendance Date	<input type="radio"/>
1	Abish Thapa	CACS351	101	6	03-10-2023	<input type="radio"/>
2	Roj Gopali	CACS351	101	6	03-10-2023	<input type="radio"/>

Submit

Source code

```
public class LoginPanel extends JPanel implements ActionListener {

    public HintTextField useridfield;

    public JPasswordField passwordfield;

    public JButton loginbutton, signupbutton;

    String loginprofile;

    private LoginPageFrame loginpageframe;

    private SignupPanel signupFrame;

    public LoginPanel(String loginprofile, ImageIcon imageicon, LoginPageFrame lpf) {

        this.loginprofile = loginprofile;

        this.loginpageframe = lpf;

        setBorder(new LineBorder(new Color(192, 192, 192)));

        setBackground(new Color(0, 0, 0, 80));

        setBounds(490, 206, 420, 434);

        setLayout(null);

        JLabel lblPassword = new JLabel("");

        lblPassword.setOpaque(true);

        lblPassword.setBackground(new Color(32, 178, 170));

        lblPassword.setIcon(new ImageIcon(".\\assets\\password1.png"));

        lblPassword.setBounds(20, 272, 60, 44);

        add(lblPassword);

        lblPassword.setHorizontalAlignment(SwingConstants.CENTER);

        lblPassword.setFont(new Font("Segoe UI", Font.PLAIN, 18));

        lblPassword.setBorder(new LineBorder(new Color(192, 192, 192)));

        useridfield = new HintTextField("Userid");

        useridfield.setBorder(new EmptyBorder(0, 3, 0, 0));

        useridfield.setToolTipText("User Id");

        useridfield.setFont(new Font("Segoe UI", Font.PLAIN, 18));
```

```

useridfield.setBounds(80, 196, 323, 44);

useridfield.setForeground(Color.DARK_GRAY);

add(useridfield);

useridfield.setColumns(10);

JLabel lblEmailId = new JLabel("");

lblEmailId.setOpaque(true);

lblEmailId.setFocusable(true);

lblEmailId.setBackground(new Color(32, 178, 170));

lblEmailId.setIcon(new ImageIcon(".\\assets\\userid.png"));

lblEmailId.setBounds(20, 196, 60, 44);

add(lblEmailId);

lblEmailId.setHorizontalAlignment(SwingConstants.CENTER);

lblEmailId.setBorder(new LineBorder(new Color(192, 192, 192)));

lblEmailId.setFont(new Font("Segoe UI", Font.PLAIN, 18));

loginbutton = new JButton("Login");

loginbutton.setFont(new Font("Segoe UI", Font.BOLD, 16));

loginbutton.setForeground(new Color(255, 255, 255));

loginbutton.addActionListener(this);

loginbutton.setBackground(new Color(32, 178, 170));

loginbutton.setBounds(20, 330, 383, 44);

loginbutton.setFocusable(false);

loginbutton.setBorderPainted(false);

add(loginbutton);

signupbutton = new JButton("Don't have account");

signupbutton.setFont(new Font("Segoe UI", Font.BOLD, 16));

signupbutton.setForeground(new Color(255, 255, 255));

signupbutton.setBackground(new Color(32, 178, 170));

signupbutton.addActionListener(this);

signupbutton.setBounds(20, 380, 383, 44);

signupbutton.setFocusable(false);

```

```

signupbutton.setBorderPainted(false);

add(signupbutton);

JLabel lblStudentLogin = new JLabel(loginprofile + " Login");

lblStudentLogin.setForeground(new Color(255, 255, 255));

lblStudentLogin.setFont(new Font("Segoe UI", Font.BOLD, 25));

lblStudentLogin.setHorizontalAlignment(SwingConstants.CENTER);

lblStudentLogin.setBounds(10, 121, 420, 38);

add(lblStudentLogin);

JLabel userprofilelabel = new JLabel();

userprofilelabel.setIcon(imageicon);

userprofilelabel.setBounds(169, 28, 100, 98);

add(userprofilelabel);

passwordfield = new HintPasswordField("Password");

passwordfield.setBorder(useridfield.getBorder());

passwordfield.setToolTipText("Password");

passwordfield.setFont(new Font("Segoe UI", Font.PLAIN, 18));

passwordfield.setBounds(80, 272, 261, 44);

add(passwordfield);

JButton showandhidebutton = new JButton("show");

showandhidebutton.setForeground(new Color(255, 255, 255));

showandhidebutton.setBounds(341, 272, 62, 44);

showandhidebutton.setBorder(new EmptyBorder(0, 0, 0, 0));

showandhidebutton.setFocusable(false);

showandhidebutton.setFocusPainted(false);

showandhidebutton.setFont(new Font("Segoe UI", Font.BOLD, 15));

showandhidebutton.setBackground(new Color(32, 178, 170));

showandhidebutton.setBorderPainted(false);

showandhidebutton.addActionListener(e

    -> {

        if (showandhidebutton.getText().equals("show")) {

```

```

        passwordfield.setEchoChar("\u0000");

        showandhidebutton.setText("hide");

    } else {

        passwordfield.setEchoChar((Character) UIManager.get("PasswordField.echoChar"));

        showandhidebutton.setText("show");

    }

});

add(showandhidebutton);

}

@SuppressWarnings("deprecation")

public void actionPerformed(ActionEvent e) {

    if (e.getSource() == loginbutton) {

        if (loginprofile.equals("Admin")) {

            boolean result = new AdminData().checkPassword(useridfield.getText(), passwordfield.getText());

            if (result == true) {

                AdminMain am = new AdminMain();

                am.setVisible(true);

                am.setLocationRelativeTo(null);

                loginpageframe.timer.stop();

                loginpageframe.imagetimer.stop();

                System.out.println("Timer running " + loginpageframe.timer.isRunning());

                loginpageframe.dispose();

            }

        } else if (loginprofile.equals("Faculty")) {

            boolean result = new FacultyData().checkPassword(useridfield.getText(), passwordfield.getText());

            if (result == true) {

                Faculty f = new FacultyData().getFacultyInfoById(useridfield.getText());

                if (!f.getSourceCode().equals("Not Assigned")) {

                    new UserData().addFacultyLoginTime(f);

```

```

        FacultyMain fm = new FacultyMain(f);

        fm.setVisible(true);

        fm.setLocationRelativeTo(null);

        loginpageframe.timer.stop();

        loginpageframe.imagetimer.stop();

        loginpageframe.dispose();

    } else {

        JOptionPane.showMessageDialog(null, "Your account is not activated. contact principal", "Login Error",
JOptionPane.ERROR_MESSAGE);

    }

}

} else if (loginprofile.equals("Student")) {

    boolean result = new StudentData().checkPassword(useridfield.getText(), passwordfield.getText());

    if (result == true) {

        Student s = new StudentData().getStudentDetailsByUserId(useridfield.getText());

        new UserData().addStudentLoginTime(s);

        StudentMain sm = new StudentMain(s);

        sm.setVisible(true);

        sm.setLocationRelativeTo(null);

        loginpageframe.timer.stop();

        loginpageframe.imagetimer.stop();

        loginpageframe.dispose();

    }

}

} else if (e.getSource() == signupbutton) {

    loginpageframe.setVisible(false);

    signupFrame = new SignupPanel();

    signupFrame.setVisible(true);

    signupFrame.setLocation(-7, 0);

} } }

```