

Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre



A Project Report
on
Student Management System

[Code No: COMP 207]

(For partial fulfillment of 2nd/2ndYear/Semester in Computer Engineering)

Submitted By:

Nirmal Dahal (11)

Ram Koirala (29)

Sirish Mainali (33)

Lokesh Sapkota (47)

Submitted To:

Dr. Rajani Chulyadyo

Department of Computer Science and Engineering

Submission Date:

2022/05/06

Bonafide Certificate

**This project work on
“Student Management System” is the bonafide work of
Nirmal Dahal, Ram Koirala, Sirish Mainali , Lokesh Sapkota who carried out the project
work under my supervision.**

Project Supervisor

**Amrit Dahal
Lecturer, DOCSE**

Abstract

Student Management System is a web app to create a digital platform that manages online resources for students and teachers or any educational institute for effective communication, learning, and management. Recent disruptions in normal life due to covid have kept people in their homes and looking for reliable services that can be accessed with ease. Several niche applications aim to serve the services and switching between several platforms for resources and user interactivity can be hectic, unreliable, and inefficient. The Student Management System aims to address this issue in the education sector by utilizing the required tools and services. It has utilized well-known frameworks for web development like react, bootstrap/CSS, and HTML for the frontend, Django for the backend, and PostgreSQL for the database.

Keywords:

Management, E-Learning, Student Management System, Student Information System

Abbreviations

- **DBMS Database Management System**
- **GUI Graphical User Interface**
- **HTML Hypertext Markup Language**
- **CSS Cascading Style Sheet**
- **LMS Learning management system**
- **ERP Enterprise Resource Planning**
- **REST Representational State Transfer**
- **MOODLE Modular Object-Oriented Dynamic Learning Environment**
- **JSON JavaScript Object Notation**

Table of contents

Contents	Page no.
Abstract	iii
Keywords:	iii
Abbreviations	iv
Table of contents	v
Chapter 1: Introduction	1
1.1 Background	1
1.2 Objectives	2
1.3 Motivation and Significance	2
1.3.1 Significance	3
Chapter 2: Related works and projects	4
2.1 Moodle	4
2.2 ELF KU	5
2.3 Blackbaud Education Management Solutions.....	5
2.4 Delta Tech school management software.....	6
Chapter 3. Design and Implementation	8
3.1 System Design.....	8
3.1.1 ER Diagram	8
3.1.2 Use-Case Diagram	9
3.1.3 APIs	10
3.2 Process of development.....	11
3.2.1 Research and learning	11
3.2.2 Coding	11
3.2.3 Testing and Debugging.....	12
3.3 System Requirement Specification	12
3.3.1. Software Specification.....	12
Chapter 4: Discussion on the Achievements	14
4.1 Elements in the Student Management Systems	14
4.1.1 HomePage.....	14

4.1.2 Departments Page	15
4.1.3 Login Page	15
4.1.4 Sign Up Page	16
4.1.5 Profile Page.....	16
4.1.6 Courses Page.....	17
4.1.7 Course Detail Page	17
4.1.8 Assignments Page.....	18
4.1.9 MCQ Page	18
Chapter 5: Conclusion and Recommendation.....	20
5.1 Limitation.....	20
5.2 Future Enhancement.....	20
References	21
Bibliography	22
Appendices.....	22

List of figures

Figures	Page no.
Fig 2. 1 Homepage of Moodle showing a short description	4
Fig 2. 2 A screenshot from ELF showing the course overview	5
Fig 2. 3 A screenshot from Blackbaud showing its system features	6
Fig 2. 4 A screenshot from Delta Tech showing a short description	7
Fig 3.1. 1 ER Diagram of Student Management System	8
Fig 3.1. 2 Use case diagram of Student Management System showing 3 roles	9
Fig 3.1. 3 API docs as seen in endpoint/docs/	10
Fig 3.1. 4 API docs as seen in endpoint/docs/	11
Fig 4.1. 1 HomePage in SMS	15
Fig 4.1. 2 Login Page in SMS.....	15
Fig 4.1. 3 Sign UP Page in SMS.....	16
Fig 4.1. 4 Profile page in SMS.....	16
Fig 4.1. 5 Courses Page in SMS	17
Fig 4.1. 6 Course Detail Page in SMS	17
Fig 4.1. 7 Assignment Pages in SMS.....	18
Fig 6. 1 HomePage in SMS	23
Fig 6. 2 Login Page in SMS.....	23
Fig 6. 3 Sign Up Page in SMS	24
Fig 6. 4 Profile page in SMS.....	24
Fig 6. 5 Courses Page in SMS	25
Fig 6. 6 Course Detail Page in SMS	25
Fig 6. 7 MCQ Page in SMS	26

Chapter 1: Introduction

1.1 Background

A school system generates and uses a large amount of data. This data must be communicated appropriately to students, faculty, and parents for the functioning of the school system. To address this necessity, student management systems are developed. Conventionally, this system is done using papers, files, and binders. But the ever-changing trend and developments in technology have brought the necessity for the automation of everything from paper and pen-based to absolute computer domination (Student Information Systems, 2013). So, the main aim of this project is to develop a computerized student management system to automate all the functions performed daily in the educational institution. This project aims to create a system that is capable of storing the details of students, faculty, and teachers and maintaining their details dynamically and systematically.

A traditional management system is done using papers and files which requires the human workforce to maintain them. So, as the number of students in the institute increases, this process of manually managing all the data becomes a hectic job for the administrator. This system stores all the data in the database which makes it easy to fetch and update whenever needed.

Student Management System (SMS) is a web application to replace conventional managerial procedures as well as learning methods. This system aims to provide educational institutions with a user-friendly environment for centralized and computerized data management systems. This system is designed to manage as well as record all the data utilized in the school such as a student's grades, interpersonal activity records, administration, etc., as well as other school-related data.

This system uses different segments(modules) to perform the specific task which helps the system to develop easily and makes it more user-friendly (Student management system, n.d.).

1.2 Objectives

- To provide educational institutions with a computerized student management system to centralize the data administration process.
- To develop a well manageable system that can collect as well as organize user data.
- To erase traditional ways of using paperwork for large-scale organizations where data mishandling is imminent.
- To provide data from physical harm and other unauthorized systems

1.3 Motivation and Significance

Organizing large data can be challenging for a fairly big institution, establishment, or other. There are several ways of organizing data but not all of them are efficient while retrieving data when needed. Paper works or otherwise, there is always a high chance of disorientation of data. Making a well-structured database management system helps, store, retrieve, evaluate and study a given set of data with maximum efficiency. While learning database management systems, one can verify the use of various management techniques on different kinds of data depending on the way one can store, manage, retrieve, and analyze data. Using such management systems makes handling data much faster and more efficient.

The COVID-19 pandemic led to the closure of universities and colleges throughout the world, with the hope that public health officials' suggestion of social distancing would help flatten the sickness curve and reduce overall mortality from the outbreak. This led to a gap in having reliable services that fully acknowledge the needs of the institutions.

The findings have shown that the desire of students to use SMS had beneficial effects during the COVID-19 pandemic on learning as sustainability engagement. (Alturki and Aldraiweesh,2021). This project's foundation is built upon this same finding and making an application that can fully cater to the needs of the educational institutions.

1.3.1 Significance

This project was started with the belief that a management system was required to organize data and activities of students during the covid-19 to make online teaching/ learning easier for schools, colleges, and students. The project was initiated to help students, and teachers share studying materials and conduct school activities online quickly, effectively, and securely. An organization might not find the proper means and resources during such pandemics to conduct educational processes smoothly, so to solve this problem the making of “Student Management System” (SMS) was initiated. SMS helps apprehend data within an organization, analyze it and organize accordingly, thus making educational processes easier and eliminating factors that might otherwise hamper natural educational processes. The database design is straightforward, thus removing difficulties in handling this system.

Chapter 2: Related works and projects

COVID-19 had resulted in schools shut all across the world. Globally, over 1.2 billion children were out of the classroom. As a result, education has changed dramatically, with the distinctive rise of e-learning, whereby teaching is undertaken remotely and on digital platforms. (Li and Lalani, 2022) Several student management systems exist to simplify and streamline the process of student data management:

2.1 Moodle

Moodle LMS is an open-source learning management system with inherent security and privacy features used by hundreds of millions of learners worldwide (Dougiamas, 2002). Moodle is probably the most popular learning management system. Moodle LMS allows educators in any sector to create flexible, safe, accessible, and highly engaging online spaces for their learners.

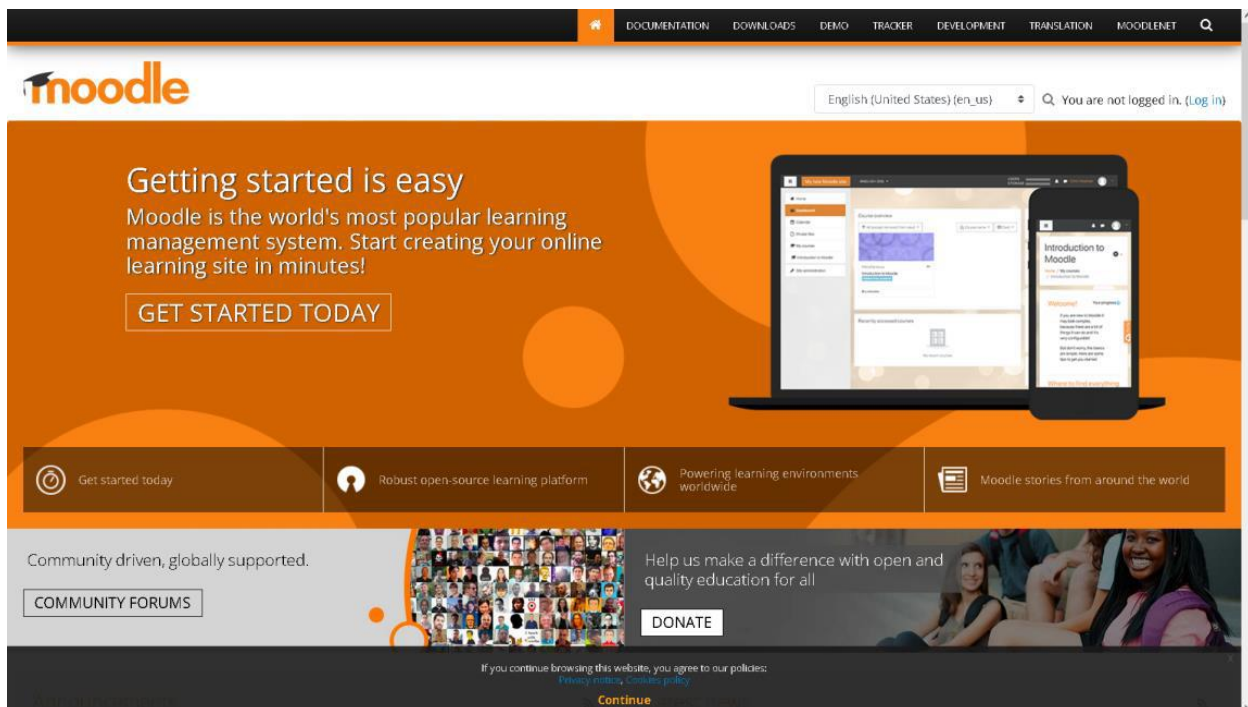


Fig 2. 1 Homepage of Moodle showing a short description

2.2 ELF KU

Elf is a student management system used at Kathmandu University (Kathmandu University E-Learning, n.d.). Elf is built with the concept of a Modular object-oriented dynamic learning environment and uses the same themes as Moodle does.

It has features like Communication tools, Student Collaboration, 24/7 Access, Student/teacher interaction, Online learning, Digital tools, a Discussion board, Assessment Tools, etc.

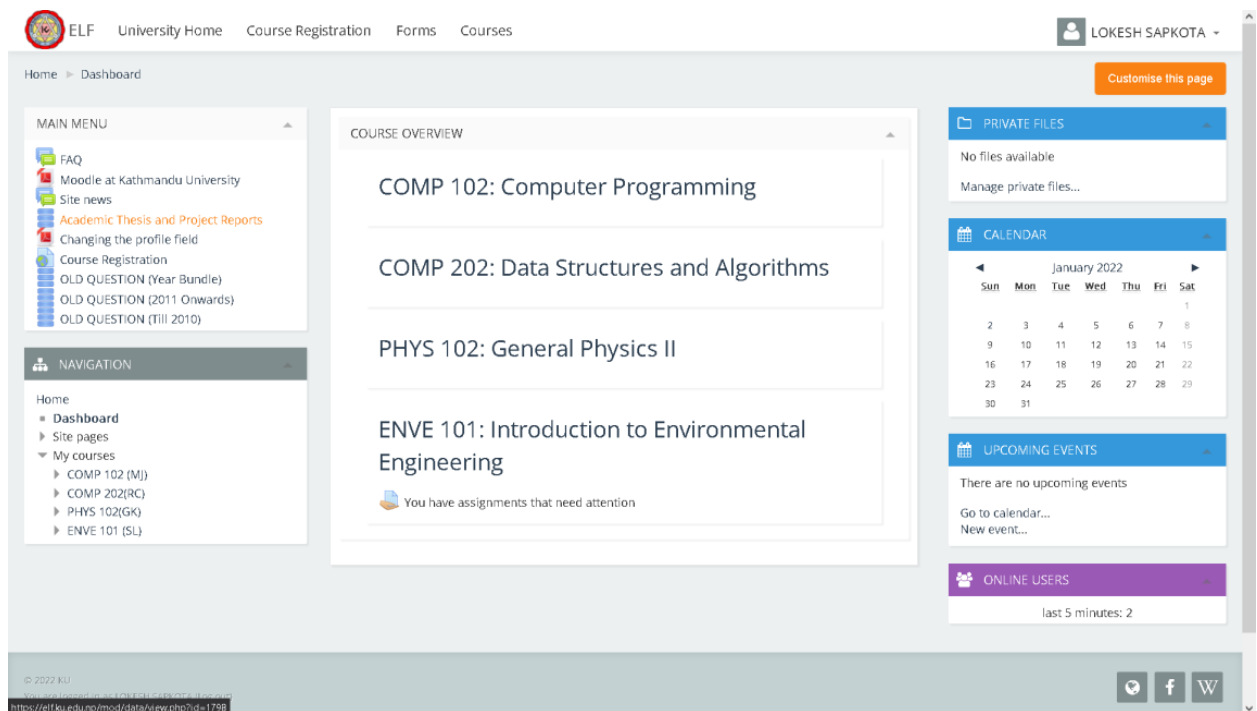


Fig 2. 2 A screenshot from ELF showing the course overview

2.3 Blackbaud Education Management Solutions

Blackbaud Education Management Solutions is a unified student management system for K–12 private schools in the USA (Education Management for K-12 Schools, n.d.). Blackbaud Enrollment Management System gives schools and families the ability to track admissions progress from first impressions through the application process.

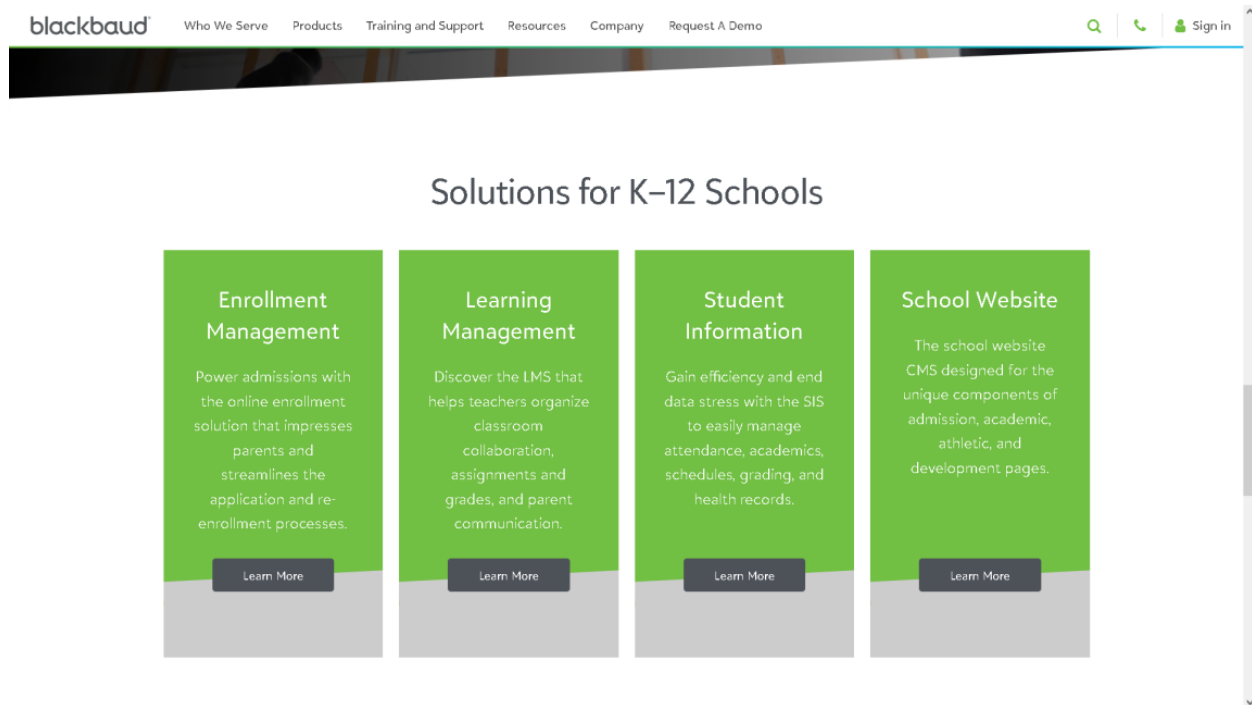


Fig 2. 3 A screenshot from Blackbaud showing its system features

2.4 Delta Tech school management software

Delta Tech's school management software is a cloud-based school ERP system that manages all elements of schools, including students, courses, exams, teachers, and employees in a single platform (School ERP Software- School management system in Nepal, n.d.). They claim that their software automates and streamlines almost all school-related activities, making it easier to keep records clean, complete, organized, and easy to retrieve.

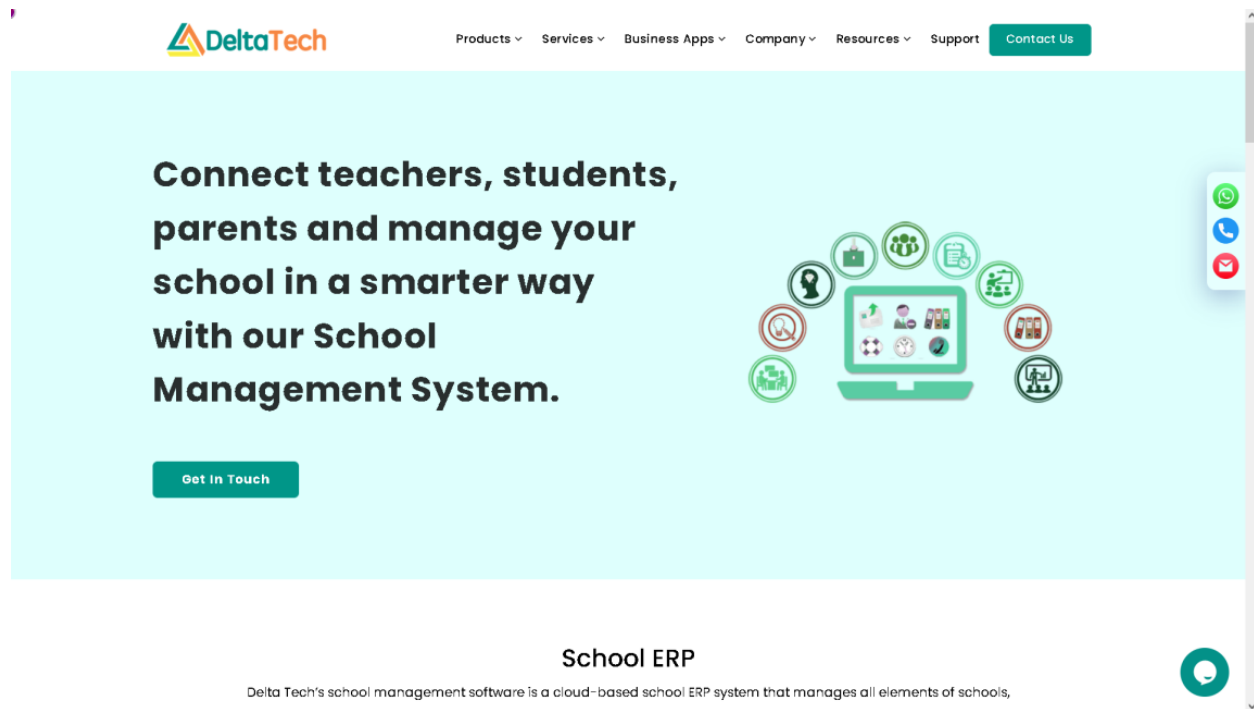


Fig 2. 4 A screenshot from Delta Tech showing a short description

Chapter 3. Design and Implementation

SMS is a Student Management System. SMS has features like creating users with different roles and providing functionalities based on that role. Initially, a quick brainstorming session was held and the group came up with some conceptual diagrams and an ER diagram. Basic features that the program was supposed to support were finalized and piece by piece construction was started. Each member of the project got a specific set of tasks and a goal that defined what their implementation part was going to be when whole features were to be stitched together.

3.1 System Design

3.1.1 ER Diagram

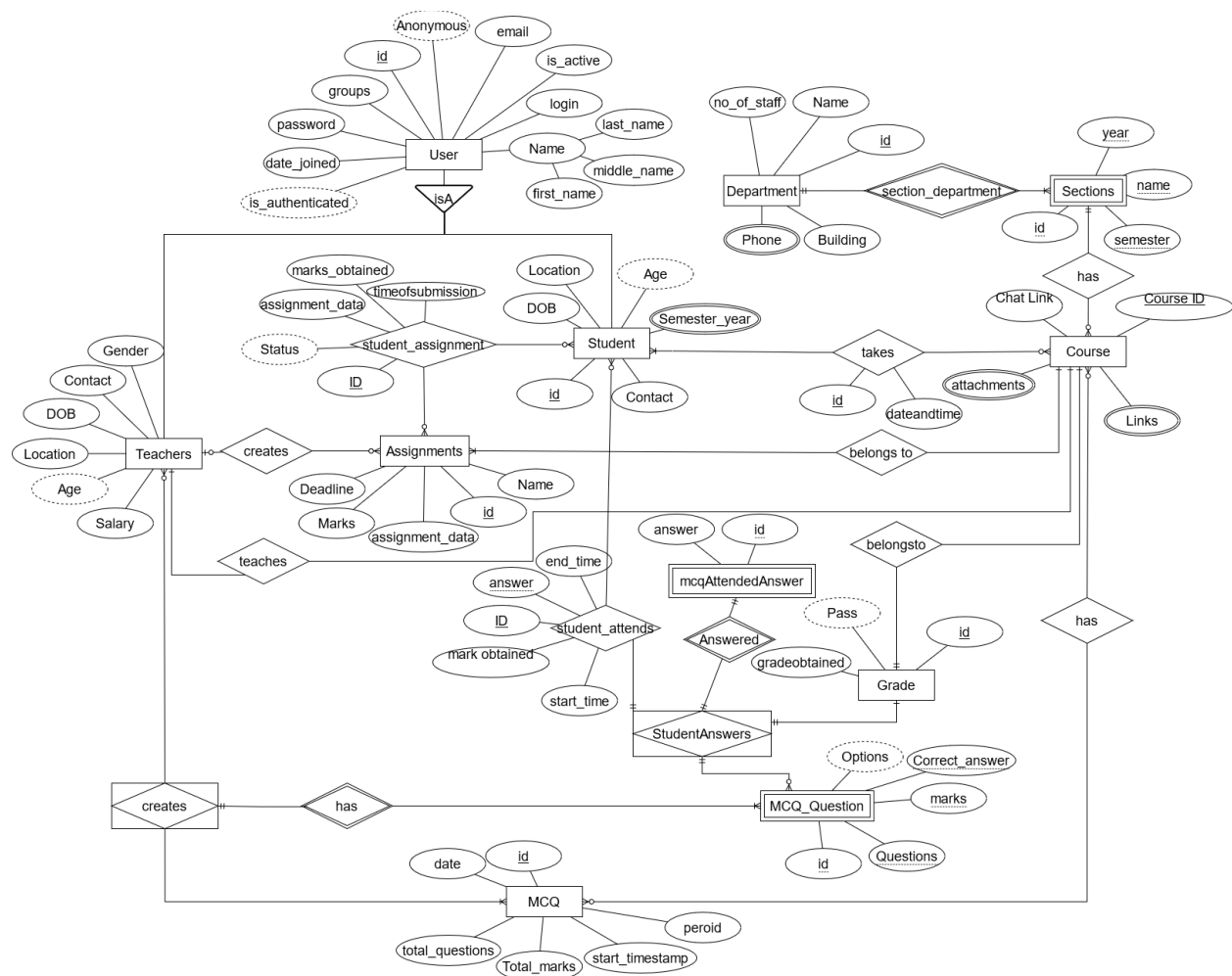
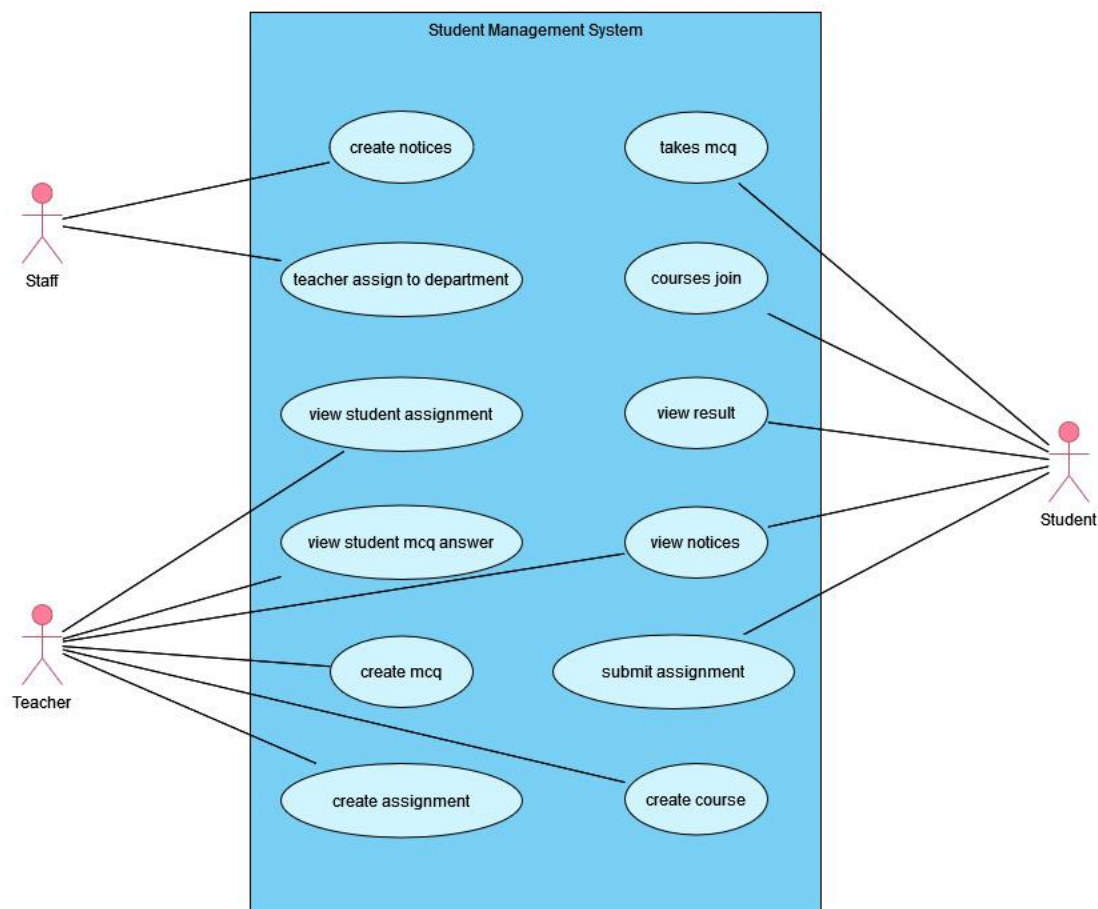


Fig 3.1. 1 ER Diagram of Student Management System

ER Diagram stands for Entity Relationship Diagram. Here, the diagram displays the relationship of entity sets stored in a database and explains the logical structure of databases. The ER diagram was created based on three basic concepts: entities, attributes, and relationships. There are 11 entities in the diagram. They are User, Department, Sections, Course, Student, Teachers, Assignments, mcqAttendedAnswer, Grade, MCQ_Question, and MCQ. The ER diagram was made at the very beginning of the project to assist in designing the schema in which the entities were related to each other.

3.1.2 Use-Case Diagram

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Fig 3.1. 2 Use case diagram of Student Management System showing 3 roles

There are three types of users in the Student Management System. ‘Staff’ regulates and assigns ‘teacher’ to the department and publishes notices for all users. The role ‘staff’ has fewer abilities in SMS than ‘teacher’ and ‘student’. The role ‘teacher’ can create courses, assignments, and MCQs or set additional information like meet links to the courses panel. The ‘teacher’ monitors students’ progress by the assignments and MCQs. The role ‘student’ has to follow routines and complete tasks assigned by the ‘teacher’ on the courses page.

3.1.3 APIs

The documentation of the API endpoints can be viewed from the endpoint/docs/. There are a total of **52** endpoints in this project. Primary HTTP methods were used in the API. They are POST, GET, PUT and DELETE. The GET method was used to retrieve data from the server. This is a read-only method, so it has no risk of mutating or corrupting the data. The POST method sends data to the server and creates a new resource. The resource it creates is subordinate to some other parent resource. The PUT method was used to update an existing resource. The DELETE method was used to delete a resource specified by its URI.

The screenshot displays the 'SMS API' documentation interface. On the left is a dark sidebar with a list of API endpoints: add_notice, course, create_user, department, login, notice, student, teacher, token, upload_profile, and user. The main content area is titled 'SMS API' and shows details for three endpoints: 'add_notice', 'course', and 'create_0'. Each endpoint section includes its name, the HTTP method (e.g., POST for add_notice, PUT for course update), and the endpoint path (e.g., /add_notice/, /course/, /course/{course id}/). To the right of each endpoint is a green 'Interact' button and a box containing instructions for using the 'coreapi' command-line client, such as 'Load the schema document' and 'Interact with the API endpoint'.

Fig 3.1. 3 API docs as seen in endpoint/docs/

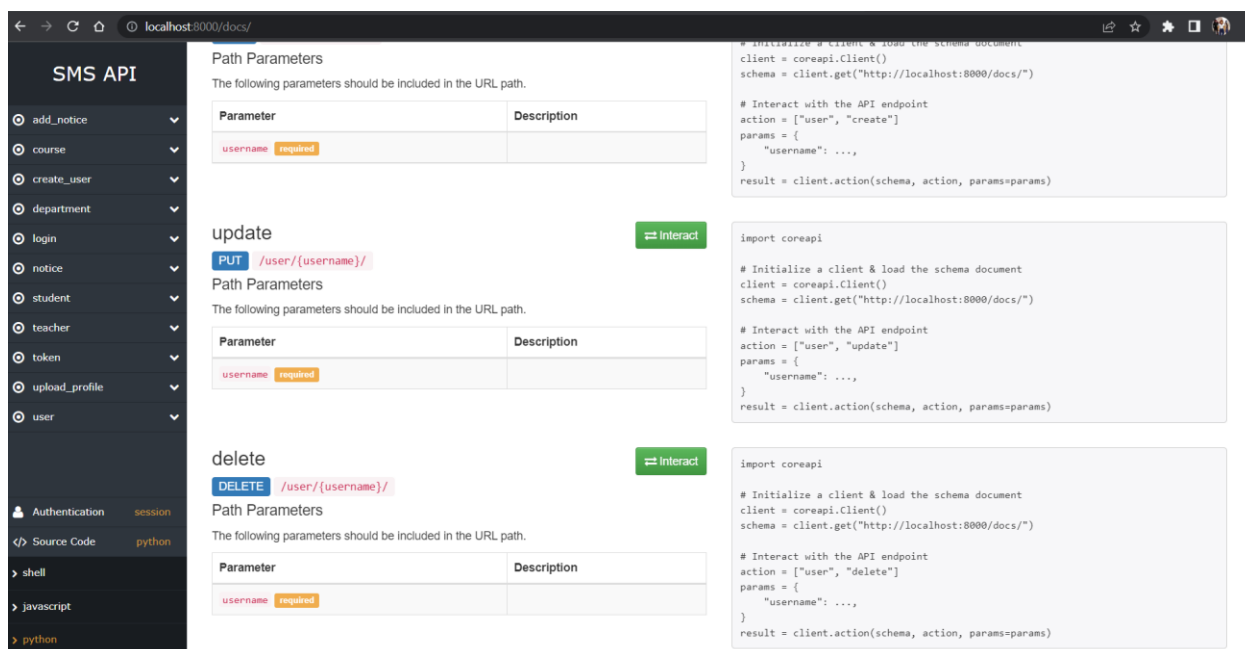


Fig 3.1. 4 API docs as seen in endpoint/docs/

3.2 Process of development

3.2.1 Research and learning

At the very beginning, it was discussed what the program would be and it was decided to build a web application named 'Student Management System'. The requirement for this particular website was felt as all the institutional learning was happening through online media. The project is aimed to integrate or aggregate many facilities that other existing management systems don't provide, like providing a dedicated field for video streaming links where teachers and students can join. Before starting the project, at first, identifying the objectives, users, and requirements for the project was done. Based on the requirements and knowledge, the Django rest framework was selected as the backend system, and react was selected for frontend design.

3.2.2 Coding

Various tools and frameworks were used in this project. Django framework was used in creating endpoints that serve data to the frontend application by React JS. Firstly, a rough sketch of the design was done and the prototype of the web app was made. The prototype

was then modified as per need/ requirement to make a user-friendly UI and proper functioning backend.

3.2.3 Testing and Debugging

To make the development process smoother and faster, the testing of the application was done from the very beginning to recognize the existing bug. The project continued by identifying the cause of those errors/bugs and removing the existing and potential errors of the program and as the final product, a fully functional user-friendly website is available.

3.2.4 Documentation

To make the process uniform and organized, Github was used. This helped us to maintain the consistency of the program as well as monitor the progress. Also, the documentation of the API was done every time a new API was used.

3.3 System Requirement Specification

3.3.1. Software Specification

This program doesn't need any other external tools because all tools chosen are inclusive in a browser.

3.3.1.1 WebClient or Browser

The program runs on a web browser like Chrome, or Firefox, and the UI is responsive (runs on small screens to large screens).

3.3.1.2 React JS

The React JS is a framework that manages the components system. The normal HTML codes are grouped in reusable blocks called components. The React JS translates JSX(JavaScript Based React Syntax) or TSX(Typescript Based React Syntax) into browser runnable JavaScript code by the Babel compiler. This project uses JS.

3.3.1.3 Javascript

Javascript is a dynamically typed language popular for scripts that run on browsers.

3.3.1.4 HTML

It is a standard markup language for creating web pages. HTML is used for structuring static pages. The dynamicity can be achieved by assisting with programming languages.

3.3.1.5 CSS

A style sheet that describes the presentation of a document written in a markup language. It assists HTML to create beautiful pages.

3.3.1.6 Python

Python is a dynamically typed interpreted language. The project backend is handled by a python-based framework for website development called Django.

3.3.1.7 Django Rest Framework

Django Rest Framework is a module that utilizes Django (python based) sub-modules to provide restful services through various endpoints. Django is a very powerful framework because of encapsulations and inheritances supported by the python language. The underlying complexities are hidden and the same code can be reused multiple times without code duplication. Data through APIs are fetched or posted in React applications.

Chapter 4: Discussion on the Achievements

After discussion at the very beginning of the project, using ReactJS framework with Javascript as frontend and using Django Rest Framework as backend was finalized. The API development process was slower than how normally websites are developed. The APIs required extra time and work for making the database serializable to JSON which provides a better view on the client side. Here, serializable means converting the relational data to an easily comprehensible and fragmented format so that each data can be isolated and modified or inserted through a set of APIs.

While working with APIs, the API calls didn't work until Cross-Origin Resource Sharing configurations were handled. In the beginning, when the authentication system was designed to be session-based, the javascript module fetched that calls on an endpoint didn't have access to session-based cookies. It was impossible to make authorized calls to the Django server. Thus, the authentication system was designed using JSON Web Tokens (JWT). JWT is stored in local storage in the browser and passed whenever authorized calls have to be made. Finally, the goal was achieved despite these challenges.

4.1 Elements in the Student Management Systems

4.1.1 HomePage

The homepage is simple and includes a greeting. There is a section for viewing departmental notices. Globally, the navbar section can be viewed on the left portion that contains nav links which leads the user to specific links.

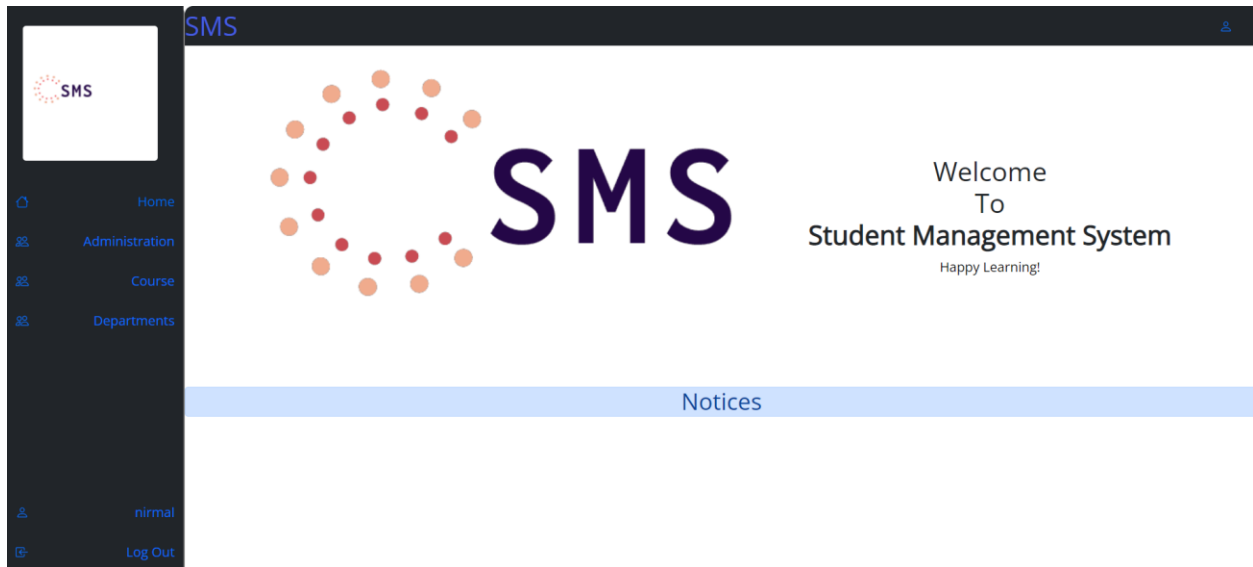


Fig 4.1. 1 HomePage in SMS

4.1.2 Departments Page

The department page consists of static content pre-designed by the administrator. The departments consist of a list of different departments under which different sections lie.

4.1.3 Login Page

From this page, teachers, students, and staff can authenticate themselves and unlock protective routes that are only available to them.

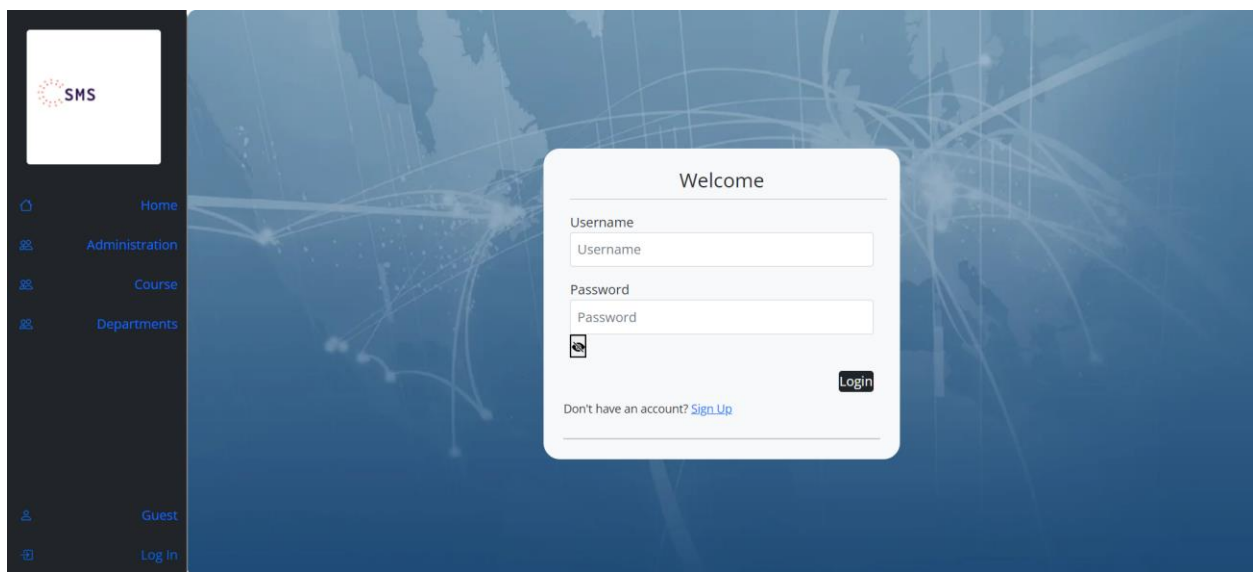
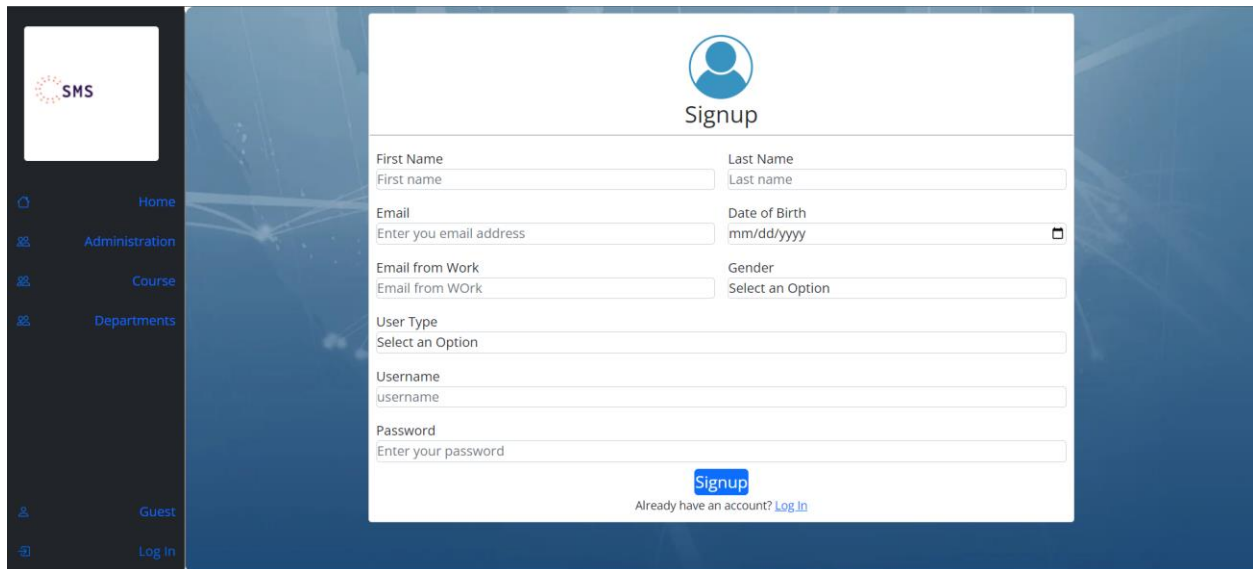


Fig 4.1. 2 Login Page in SMS

4.1.4 Sign Up Page

On this page, student, teachers, or staff can send their credentials to the system for identification and authorization.

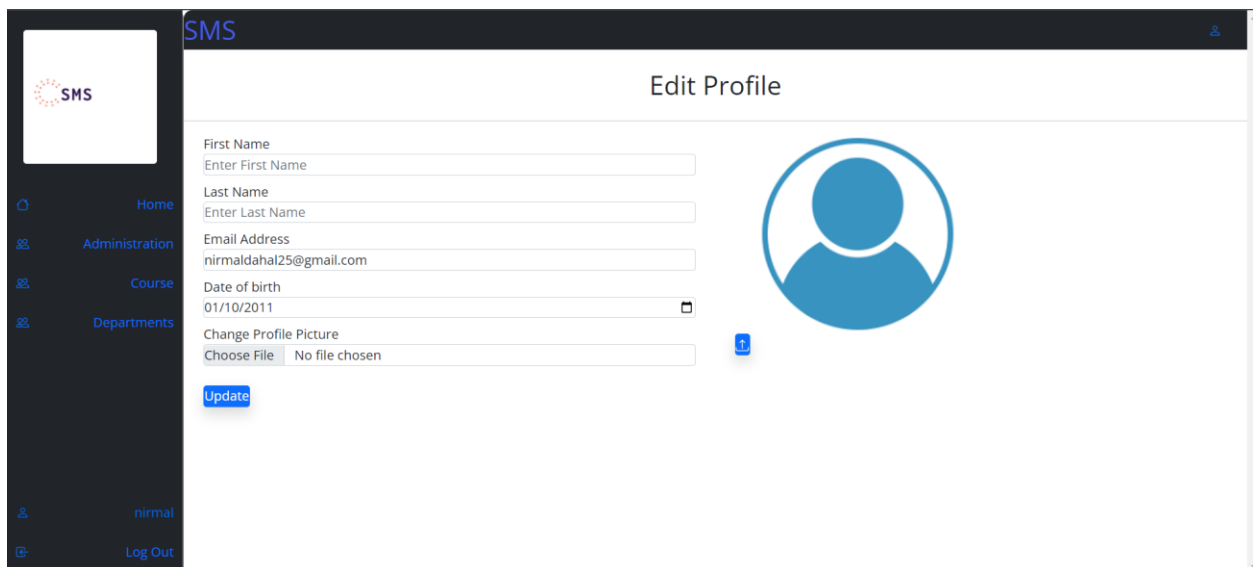


The screenshot shows the 'Sign Up' page of the SMS system. On the left is a dark sidebar with the SMS logo and navigation links: Home, Administration, Course, Departments, Guest, and Log In. The main content area has a blue header with a user icon and the title 'Signup'. Below the header is a form with the following fields: First Name (placeholder: First name), Last Name (placeholder: Last name), Email (placeholder: Enter you email address), Date of Birth (placeholder: mm/dd/yyyy), Email from Work (placeholder: Email from Work), Gender (placeholder: Select an Option), User Type (placeholder: Select an Option), Username (placeholder: username), and Password (placeholder: Enter your password). At the bottom of the form is a blue 'Signup' button and a link that says 'Already have an account? Log In'.

Fig 4.1. 3 Sign UP Page in SMS

4.1.5 Profile Page

On this page, the user can modify, submit and view their profile. But the user cannot change their role once the role is assigned.



The screenshot shows the 'Edit Profile' page of the SMS system. On the left is a dark sidebar with the SMS logo and navigation links: Home, Administration, Course, Departments, nirmal, and Log Out. The main content area has a blue header with the title 'Edit Profile'. Below the header is a form with the following fields: First Name (placeholder: Enter First Name), Last Name (placeholder: Enter Last Name), Email Address (placeholder: nirmaldahal25@gmail.com), Date of birth (placeholder: 01/10/2011), and Change Profile Picture (placeholder: Choose File, No file chosen). At the bottom of the form is a blue 'Update' button. To the right of the form is a large blue circular profile picture placeholder.

Fig 4.1. 4 Profile page in SMS

4.1.6 Courses Page

This page can be accessed by teachers and students. The courses are given a card view. The teacher can view all the courses created by themselves and the student can view all the courses they have joined.

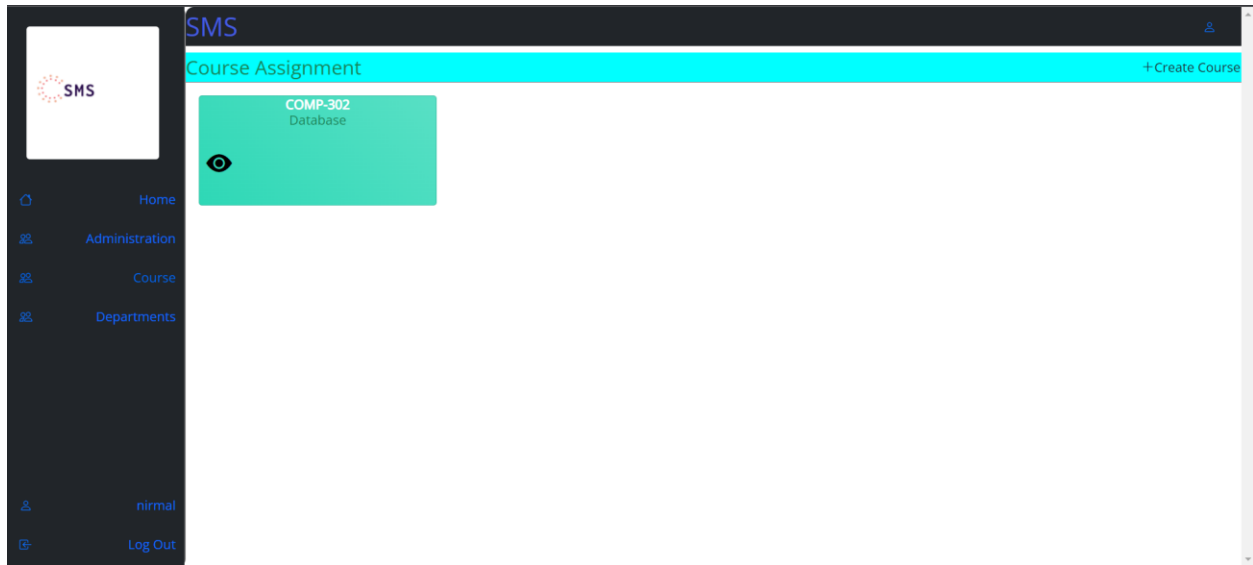


Fig 4.1. 5 Courses Page in SMS

4.1.7 Course Detail Page

The course detail page contains links to assignments, MCQs, and grades. The teacher can monitor all the student's progress while the student can only view their report.



Fig 4.1. 6 Course Detail Page in SMS

4.1.8 Assignments Page

The assignments page can be found inside the course page. Here, the teacher can create assignments that are to be submitted by students. The assignment also has marks/points and deadlines for the student to see.

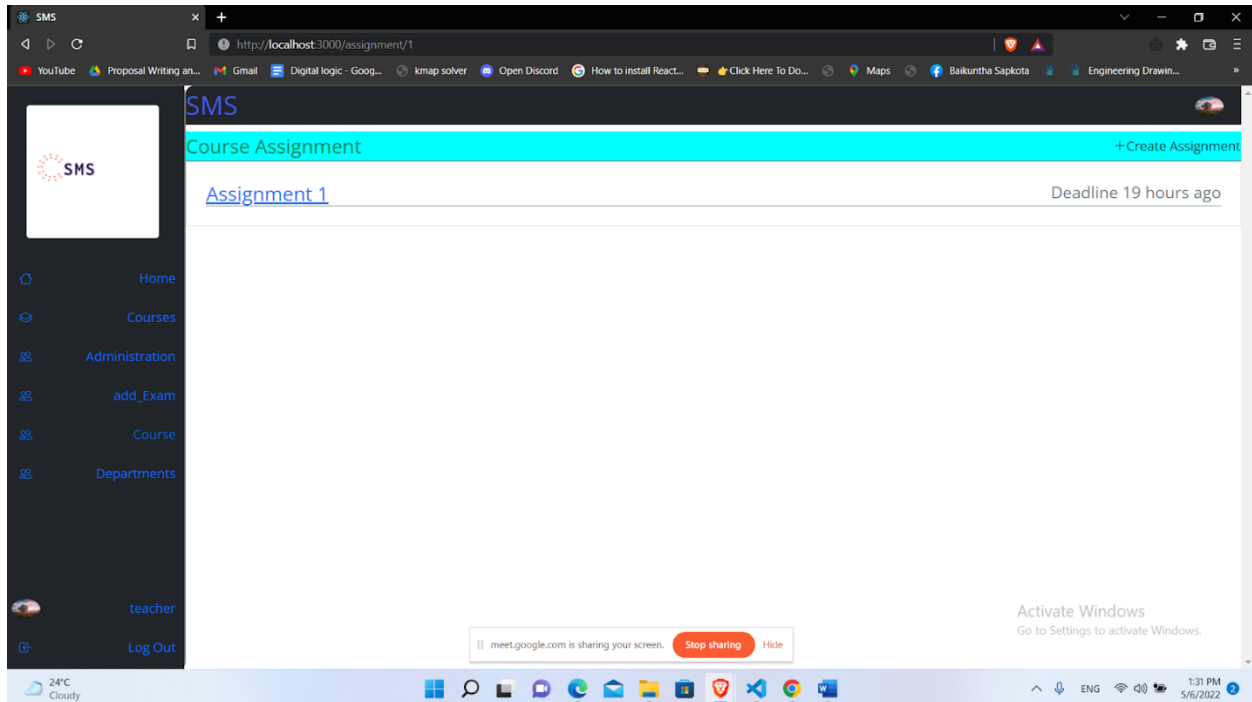



Fig 4.1. 7 Assignment Pages in SMS

4.1.9 MCQ Page

The MCQ is one of the methods for student evaluation. In this, the respective course teacher creates MCQs with starting and end times where each MCQ question holds points set by the teacher. The student has to attend the MCQ exam in the specified time interval.



Home

Administration

Course

Departments

nirmal

Log Out

SMS

MCQ Name

1) Convert 10 decibel to bel. Marks: 1

☐ 1 bel

☐ 10 bel

☐ 100 bel

☐ None of Above

Submit

Start Time

End Time

Elapsed Time

2021/01/21 12:00:00

2021/01/21 12:00:00

00:01:00

Fig. 4.1.8 MCQ Page in SMS

Chapter 5: Conclusion and Recommendation

This project provides a platform for the educational institutions and the respective users with a computerized student management system to centralize the data administration process. Upon completing this project, a fully functional management system was made where an online learning environment was created that provides better interactivity, communication, and management of resources. This project is fully compatible with any platform and the responsive UI design runs on various window sizes. A simple UI design has been implemented while developing the website.

5.1 Limitation

An utmost effort was put into developing this project and there are a few limitations of this project.

- Mathematical symbols or images cannot be used in MCQs.
- The web app does not have Email services.
- WebRTC-based services are not available.

5.2 Future Enhancement

In the future, the project can be enhanced in the following ways.

- Markdown support for supporting mathematical symbols as well as the support of images in MCQs can be added.
- The project was also supposed to use WebRTC services, but due to time limitations, it was not possible. In the future, the project can integrate these services.

References

- Alturki, U., & Aldraiweesh, A. (2021, October 3). *Application of learning management system (LMS) during the COVID-19 pandemic: A sustainable acceptance model of the expansion technology approach*. MDPI. Retrieved May 6, 2022, from <https://doi.org/10.3390/su131910991>
- Li, C., & Lalani, F. (2020, April). *The COVID-19 pandemic has changed education forever. this is how*. World Economic Forum. Retrieved May 6, 2022, from <https://www.weforum.org/agenda/2020/04/coronavirus-education-global-covid19-online-digital-learning/>
- Dougiamas, M. (2022, April 7). *Moodle LMS*. Moodle. Retrieved May 6, 2022, from <https://moodle.com/lms/>
- Education Management Solutions for K-12 schools*. Blackbaud. (n.d.). Retrieved May 6, 2022, from <https://www.blackbaud.com/solutions/organizational-and-program-management/education-management/k-12>
- Kathmandu University e-learning*. Kathmandu University E-Learning: Log in to the site. (n.d.). Retrieved May 6, 2022, from <https://elf.ku.edu.np/>
- Delta Tech - software company based in Nepal. (n.d.). *School ERP software: School Management System in Nepal*. Delta Tech. Retrieved May 6, 2022, from <https://deltatechnepal.com/business-apps/school-erp>
- Student Mangement System. (n.d.). Retrieved May 6, 2022, from <https://www.lovelycoding.org/student-management-system/> 12
- Final thesis of Student Information Systems*. Studymode. (n.d.). Retrieved May 6, 2022, from <https://www.studymode.com/essays/Final-Thesis-Of-Student-Information-Systems-1815766.html>

Bibliography

Vincent, W. S. (2018). *Django for apis: Build web apis with python and Django*. Willaim S. Vincent.

Christie, T. (n.d.). *Django rest framework*. Home - Django REST framework. Retrieved May 6, 2022, from <https://www.django-rest-framework.org/>

Mark Otto, J. T. (n.d.). *Bootstrap*. Bootstrap · The most popular HTML, CSS, and JS library in the world. Retrieved May 6, 2022, from <https://getbootstrap.com/>

React – a JavaScript library for building user interfaces. – A JavaScript library for building user interfaces. (n.d.). Retrieved May 6, 2022, from <https://reactjs.org/>

official, 9cv9. (2019, July 19). *What are get, post, put, Patch, delete? A walkthrough with JavaScript's fetch API*. Medium. Retrieved May 6, 2022, from <https://medium.com/@9cv9official/what-are-get-post-put-patch-delete-a-walkthrough-with-javascrpts-fetch-api-17be31755d28>

Appendices

- **Appendix I (Home Page)**

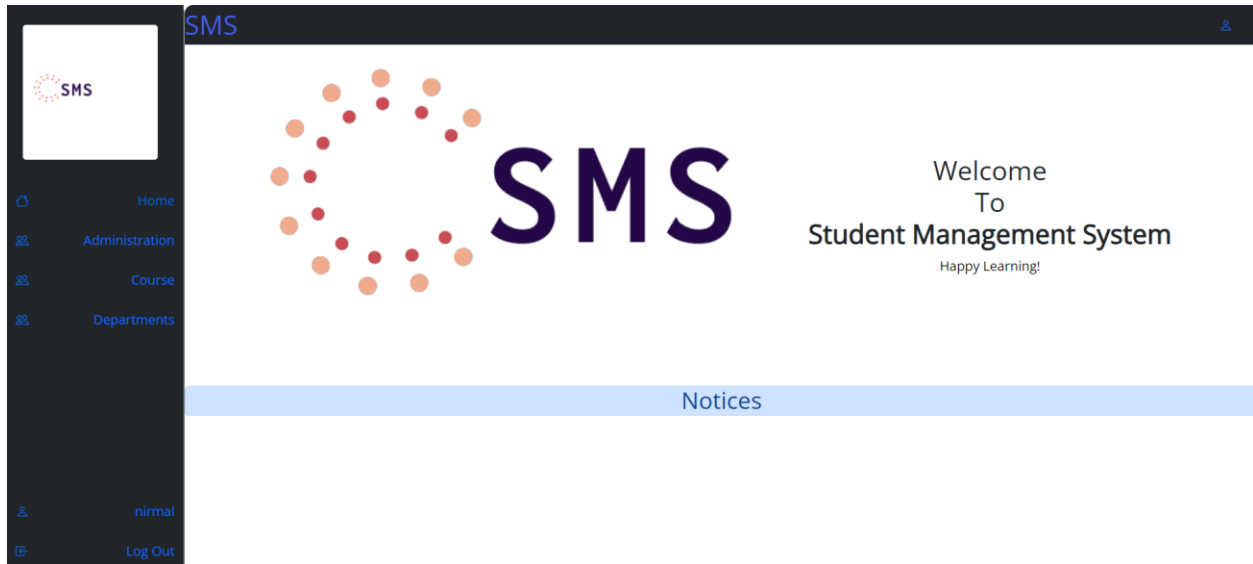


Fig 6. 1 HomePage in SMS

- **Appendix II (Login Page)**

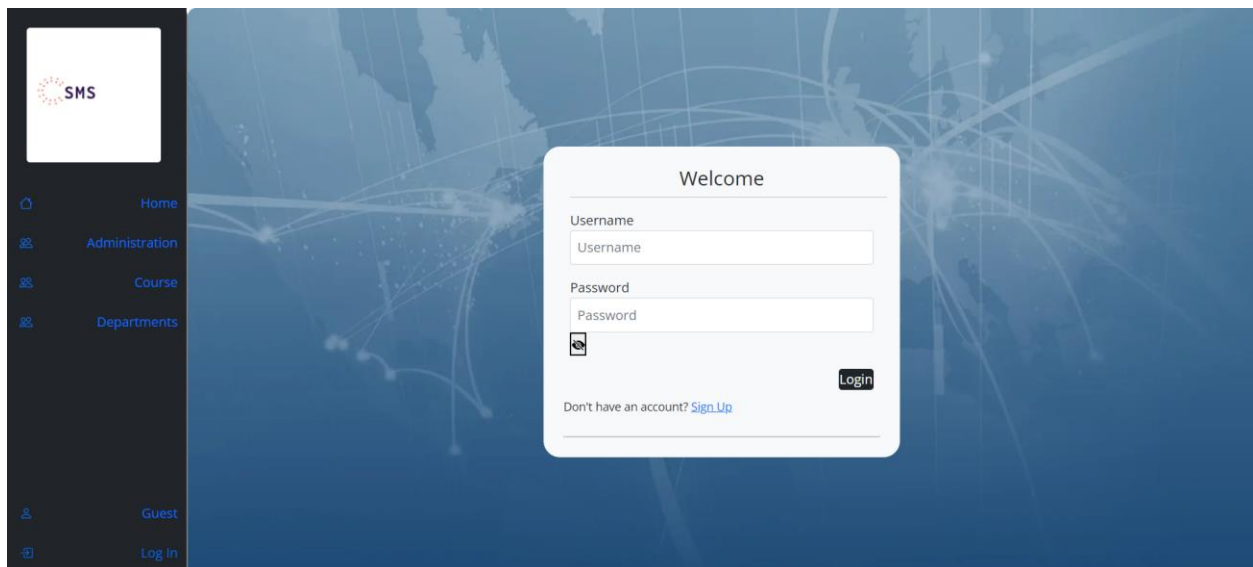


Fig 6. 2 Login Page in SMS

- **Appendix III (Sign Up Page)**

Fig 6. 3 Sign Up Page in SMS

- **Appendix IV (Profile Page)**

Fig 6. 4 Profile page in SMS

- **Appendix V (Courses Page)**

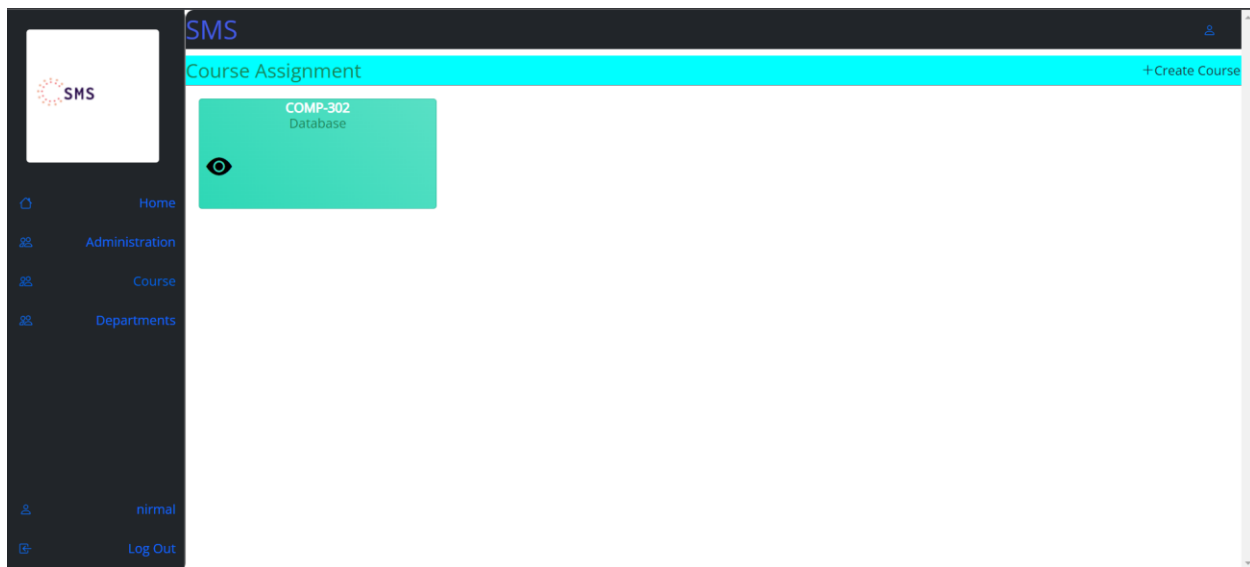



Fig 6. 5 Courses Page in SMS

- **Appendix VI (Course Detail Page)**



Fig 6. 6 Course Detail Page in SMS

- **Appendix VII (MCQ Page)**



Home

Administration

Course

Departments

nirmal

Log Out

SMS

MCQ Name

1) Convert 10 decibel to bel.

Marks: 1

☐ 1 bel

☐ 10 bel

☐ 100 bel

☐ None of Above

Submit

Start Time

End Time

Elapsed Time

2021/01/21 12:00:00

2021/01/21 12:00:00

00:01:00

Fig 6. 7 MCQ Page in SMS