"Student Progress Tracker"

A Project Report Submitted to
Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of
Bachelor of Technology
(*Computer Science and Engineering)*

Submitted By:
Atharva Puranik (0827CS201049)
Atharva Pagare(0827CS201048)

Guided By:
Prof. Ronak Jain
Department of Computer Science And Engineering,
AITR, Indore

Acropolis Institute of Technology & Research, Indore

July - December 2022

EXAMINER APPROVAL

The Project entitled ***"Student Progress Tracker"*** submitted by **Atharva Puranik (0827CS201049)** and **Atharva Pagare (0827CS201048)** has been examined and is hereby approved towards partial fulfillment for the award of ***Bachelor of Technology degree in Computer Science*** discipline, for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve anystatement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner) **(External Examiner)**

Date: **Date:**

GUIDE RECOMMENDATION

This is to certify that the work embodied in this project entitled **"*Student Progress Tracker*"** submitted by **Atharva Puranik (0827CS201049)** and **Atharva Pagare (0827CS201048)** is a satisfactory account of the bonafide work done under the supervision of ***Dr. Kamal Kumar Sethi***, is recommended towards partial fulfillment for the award of the Bachelor of Engineering (Computer Science) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

(Project Guide) **(Project Coordinator)**

STUDENTS UNDERTAKING

This is to certify that project entitled ***"Student Progress Tracker"*** has developed by us under the supervision of ***Dr. Kamal Kumar Sethi***. The whole responsibilityof work done in this project is ours. The sole intension of this work is only for practical learning and research.

We further declare that to the best of our knowledge, this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work found then we are liable for explanation to this.

Atharva Puranik(0827CS201049)

Atharva Pagare(0827CS20148)

Acknowledgement

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

There are number of people without whom this projects work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide and mentor **Prof. Ronak Jain,** Professor, AITR, Indore for his motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Dr Kamal Kumar Sethi**, HOD CSE, AITR Indore for his support, suggestion and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of CSE Dept, AITR Indore for providing me all support, help and advice during the project. We would be failing in our duty if do not acknowledge the support and guidance received from **Dr S C Sharma**, Director, AITR, Indore whenever needed. We take opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me all necessary facilities for project to achieve our objectives.

We are grateful to **our parent** and **family members** who have always loved and supported us unconditionally. To all of them, we want to say "Thank you", for being the best family that one could ever have and without whom none of this would have been possible.

Atharva Puranik (0827CS201049)

Atharva Pagare (0827CS201048)

Executive Summary

Student Progress Tracker

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal(MP), India for partial fulfillment of Bachelor of Engineering in Information Technology branch under the sagacious guidance and vigilant supervision of Dr. Kamal Kumar Sethi.

In this project, we are making a web-based student management system that will lessen the amount of manual work required and allow teachers to manage student data, including their marks and attendance, on the same platform. It will also improve communication between the administration and the students and teachers through notice.

We used Django framework for the development of this project. As Django is Backward compatible and it offers the provision of working with its older versions and makes use of its older formats and features.

.

Keywords: Python, Django

“A little progress each day adds up to big results.”

-Satya

List of Figures

List of Tables

Table of Contents

Chapter 1. Introduction

Introduction

Schools and Universities are the foundation of knowledge and an educational body on which students rely upon. Therefore, they need to maintain a proper database of its students to keep all the updated records and easily share information with students.

Over the recent years, the performance and efficiency of the education industry have been enhanced by using the Student Management System. This tool has productively taken over the workload of the admin department with its well-organized, easy, and reliable online management Web Application.

1.1 Overview

The project is based on the students' growth. The student data is stored on a website that is also used to analyze and track it. The development of a system enables the storing and analysis of student grades and marks. It can monitor student performance, analyze their grades, monitor their attendance, and provide notifications.

1.2 Problem Statement and Objectives

Teachers currently prepare and track students' learning outcomes through report cards using manual labor. This is the national standard practice, which is carried out manually. In the present, it is impossible for teachers to pay close attention to every student and bring out the best in each individual student. In order to follow a student's performance in real time, including their academic development, solutions must be developed. To manage students' performance in academics throughout their academic education. To manage student information. And to reduce unnecessary paper work to manage record of the student

1.3 Scope of the Project

This system is entirely user-friendly and effective in its design. The tasks of this system, which could include registering new students, monitoring student performance in academic and extracurricular activities, and doing any other duties required to keep the school's administrative division operating smoothly, will be helpful to instructors. Because it aids in learning, assessment is an essential part of education. Students can judge their understanding of the course material and keep track of how they are doing in their areas of interest when they can monitor how they are performing in a class. Additionally, assessment can inspire kids.

1.4 Requirement Analysis

Functional Requirements:

1. **Log in Module (LM)**

The Login Module must be able to load in the web browser for users (admin, students, and teachers). The LM must assist the user with system login. User name and password boxes must both be present on the login screen. When a user types, the password field should be covered by symbols.

Additionally, a button marked "Login" must be present.

Only the user will be able to utilise the system features when the database administrator has confirmed the user's username and password after the user clicks the Login button.

2. **Registered Users Module (RUM)**

The user will be able to continue exploring the website and access detailed information about the school or college after successfully logging in.

Users (admin, students, and teachers) will be able to edit and manage their profiles after successfully logging in, including updating their password and personal information.

3. **Server Module (SM)**

Between the various modules and the DB, SM should be. All requests must be sent to SM, who will then format the pages for display properly. All requests from the other modules must be validated and carried out by SM.

4. **Administrator Module (AM)**

The system will reveal administrative features after a successful login. The administrative tasks displayed must be add and update.

Administrators can add new student details, delete any that are no longer needed, and do many other things when they click the "add" button.

The system will display a section where the administrator can update the student information and lecture schedule that are currently kept in the database when the administrator hits the update button.

The AM module will submit the request to the Server Module, which will make the necessary modifications to the DB, when an administrator adds, modifies, or deletes a record.

1.5 Team Organization

- **Atharva Puranik**: In addition to using Django, I worked on the front end of the project, created the HTML templates, and styled the website. On part of the backend stuff, I also worked. I conducted an initial inquiry, comprehended the limitations of the current system, investigated the subject and its application, and looked through numerous research articles pertaining to the student management system and the technology that would be used. As part of my work on this project, I also completed some documentation. On top of that, I researched it and worked on the backend.

- **Atharva Pagare:** I focused primarily on the backend tasks. Along with conducting preliminary research and being aware of the current system's shortcomings, I also read up on the subject and its application and perused a number of research papers and websites pertaining to the student management system and the technology that would be used. I also contributed to the documentation for this project. Additionally, I work on making the graphs on the website using apex charts and connected it to the data.

1.6 Report Structure

The project Student Progress Tracker is primarily concerned with managing the student data and keep the record of student's attendance, result etc. whole project report is categorized into five chapters.

Chapter 1: Introduction- introduces the background of the problem followed by rationale for the project undertaken. The chapter describes the objectives, scope and applications of the project. Further, the chapter gives the details of team members and their contribution in development of project which is then subsequently ended with report outline.

Chapter 2: Review of Literature- explores the work done in the area of Project undertaken and discusses the limitations of existing system and highlights the issues and challenges of project area. The chapter finally ends up with the requirement identification for present project work based on findings drawn from reviewed literature and end user interactions.

Chapter 3: Proposed System - starts with the project proposal based on requirement identified, followed by benefits of the project. The chapter also illustrate software engineering paradigm used along with different design representation. The chapter also includes block diagram and details of major modules of the project. Chapter also gives insights of different type of feasibility study carried out for the project undertaken. Later it gives details of the different deployment requirements for the developed project.

Chapter 4: Implementation - includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designed in project along with their functionality. Further it discuss the experiment results along with testing of the project. The chapter ends with evaluation of project on different parameters like accuracy and efficiency.

Chapter 5: Conclusion - Concludes with objective wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

Chapter 2. Review of Literature

Review of Literature

Providing proper academic grades to each student. The control process includes a forward information flow resulting from academics, painting, sports, and other activities, and a feedback information flow for monitoring purposes to meet the overall grade. Progress tracking is one of the most important feedback types where decision making and, consequently, project success, undeniably depend on accurate and efficient progress tracking in the education field.

2.1 Existing Technologies:

1. Kahoot:

Kahoot is an online game-based learning platform. It allows teachers, organizations, and parents to set up fun web-based learning for others. This could include your coaches, athletes, or parents. Kahoot can be used as a fun trivia activity to do with members of your organization or coaches to use with their players about the sport or just a series of fun questions.

Merits: It develops interest in the student since they are keeping track using course-related quizzes and puzzles

Demerits: Sometimes students may cheat and give the answer to the question, in which case they get extra marks in the progress report.

Reference link: https://kahoot.com/what-is-kahoot/

2. Google Sheets

Google Sheets is a web-based application that enables users to create, update and modify spreadsheets and share the data online in real time.

Google's product offers typical spreadsheet features, such as the ability to add, delete and sort rows and columns. But unlike other spreadsheet programs, Google Sheets also enables multiple geographically dispersed users to collaborate on a spreadsheet at the same time and chat through a built-in instant messaging program.

Merits: It is the easiest way of storing and manage the data. We get everything on spreadsheet.

Demerits: Sometimes teachers need to give access to students to fill in the details in spreadsheets.

Reference link: https://www.techtarget.com/whatis/definition/Google-Spreadsheets

3. ClassDojo

ClassDojo is digital sharing platform that allows teachers to document the day in class and share that with families via a web browser so that nearly any device can access the content – from a simple smartphone to a laptop computer. As long as it has a browser, photos and videos can be viewed.

ClassDojo's messaging service is another big draw since it allows parents and teachers to communicate through commenting on photos and videos and messaging directly. The translation service that offers more than 35 languages is a great tool since it allows teachers to enter text in their native language and have all parents and guardians read it in theirs.

Merits: It is a good website since the teacher can give feedback on the performance of the student and can also communicate with their parents.

Demerits: They can charge commission on every sale. This can range between 10-20 per cent of the gross cost. There may be restrictive terms and conditions imposed by OTAs such as guest cancellation and automatic room reselling policies.

Reference link: https://www.techlearning.com/how-to/what-is-classdojo/

4. Google Classroom:

Google Classroom is a suite of online tools that allows teachers to set assignments, have work submitted by students, to mark, and to return graded papers. It was created to get eliminate paper in classes and to make digital learning possible. It was initially planned for use with laptops in schools, such as Chromebooks, in order to allow the teacher and students to share information and assignments more efficiently.

Merits: Paperless Teachers and students won't have excessive amounts of paper to shuffle since Classroom is completely paperless. When teachers upload assignments and assessments to Classroom, they are simultaneously saved to Drive. Students can complete assignments and assessments directly through Classroom, and their work is also saved to Drive.

Demerits: Because of the need for an individual Google account, it's possible (although not likely) that parents may opt-out of their child creating a student account. Unfortunately, refusal of the Google Terms of Service will keep a student out of Classroom. Teachers may need to work with families so that they understand the value of their child joining the platform.

Reference link: https://en.m.wikipedia.org/wiki/Google_Classroom

5. Google Forms:

Google Forms allows you to create online tests, surveys, questionnaires, and other forms of information crowdsourcing. This program is part of the Google Workspace and serves solutions, particularly for your survey administration and form needs. Google Forms is equipped with drag-and-drop features, ready-made templates, question customization, and other tools needed to create surveys and forms.

Merits: It is a free online tool, that allows you to collect information easily and efficiently.

Demerits: There are certain limitations regarding the capabilities of this tool. It accepts texts up to 500 Kb; images up to 2 Mb; and for spreadsheets the limit is 256 cells or 40 sheets.

2.1.1 Conclusion

This chapter reviews the literature surveys that have been done during the research work. The related work that has been proposed by many researchers has been discussed. After surveying the existing systems, finding out the advantages and disadvantages, we have decided to make the student progress Tracker which overcomes disadvantages of the existing systems to some extent.

Chapter3. Proposed System

Proposed System

3.1 The Proposal

The proposal is to deploy a system which is designed to be more efficient than the manual system. It invokes all base tasks that are now carried out manually, such as the grading student and maintaining data. The proposed System is completely computer-based application. Thousands of records can search and displayed without taking any significant time.

3.2 Benefits of the Proposed System

Careful handling of student data Numerous apps or websites are not required to store a variety of data.

- Student managing in real-time.
- Online and anytime attendance and record with date.
- Student can View Result, attendance, and notice information.
- Feedback of Students Performance Can be given by Teachers.

3.3 Feasibility study

A feasibility study is an analysis of how successfully a system can be implemented, accounting for factors that affect it such as economic, technical, and operational factors to determine its potential positive and negative outcomes before investing a considerable amount of time and money into it.

3.3.1 Technical

There is a need to create a dynamic and effective website. Students and staff can easily operate the website. For this, Django is used, which is an advanced web framework written in Python that makes use of the model-view-controller (MVC) architectural pattern. It supports all the leading databases, such as SQLite

3.3.2 Operational

Our system's key credo is that users can easily conduct all functions. The technology can accomplish it precisely and effectively, making it operationally practical.

3.4 Design Representation

3.4.1 Data Flow Diagrams

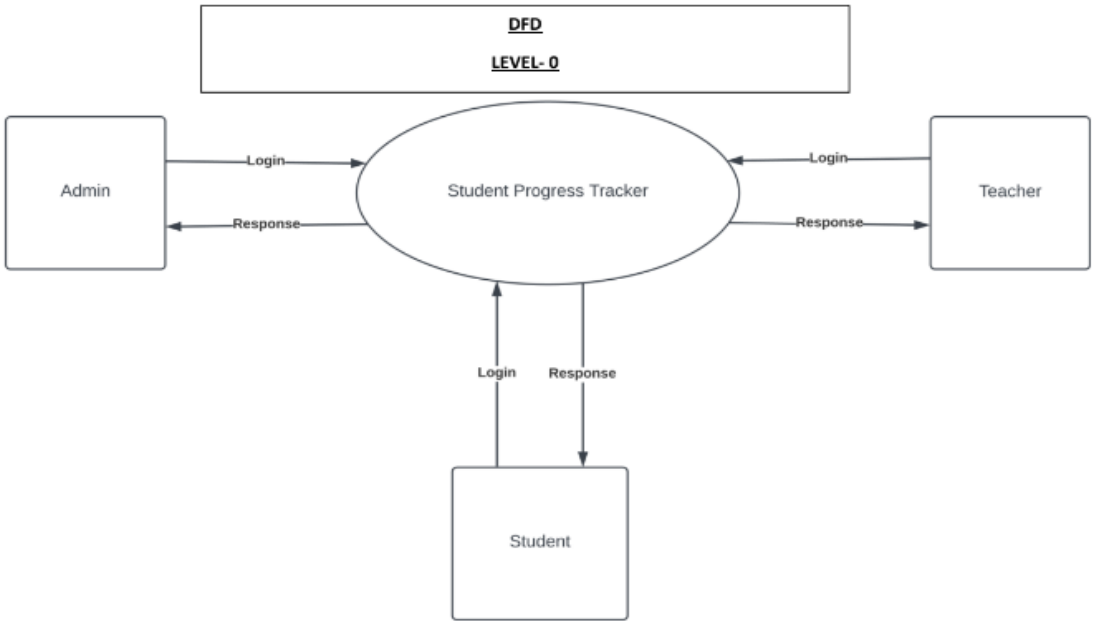

Fig 3-1: DFD DIAGRAM (LEVEL 0)

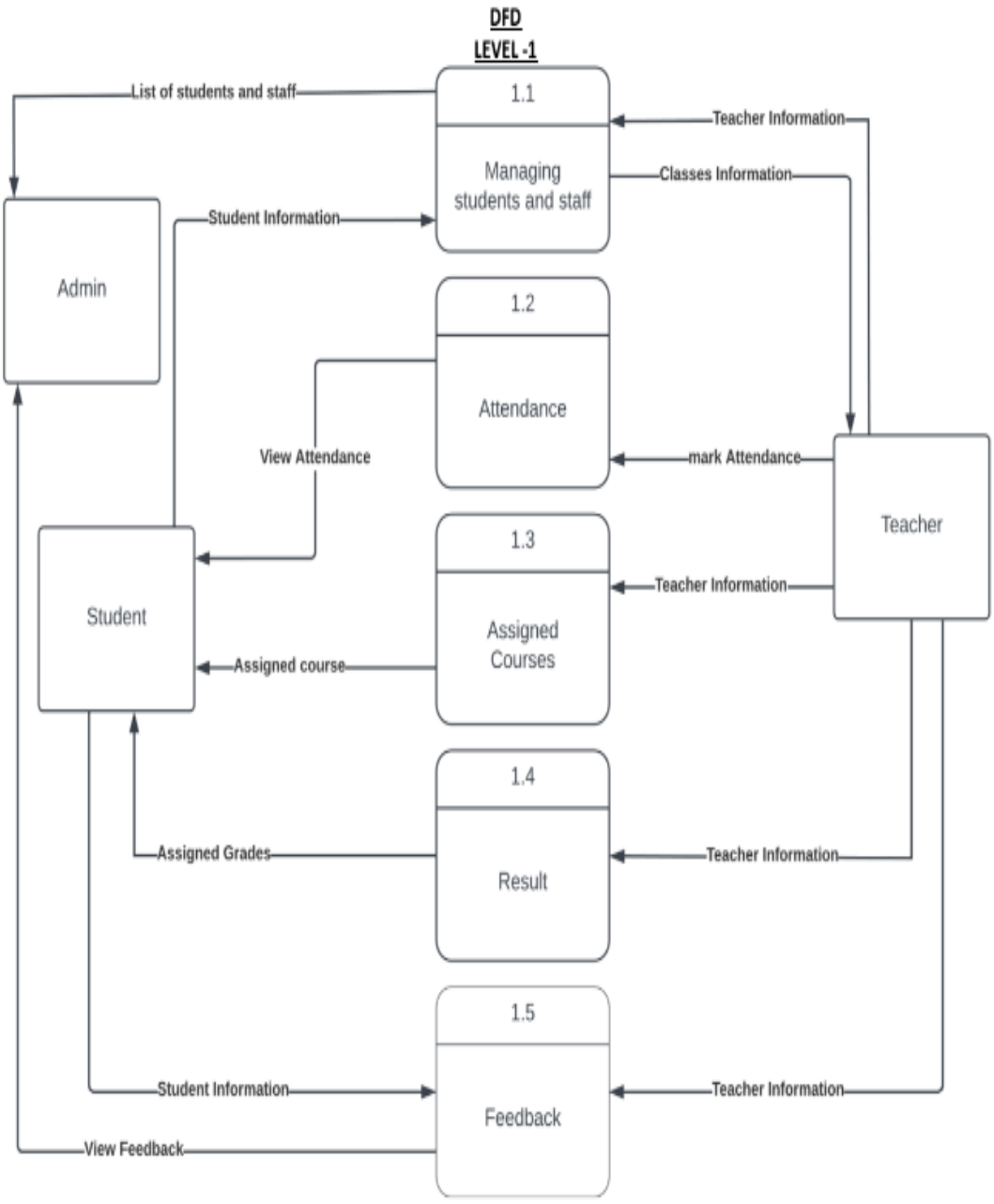

Fig 3-2: DFD DIAGRAM (LEVEL 1)

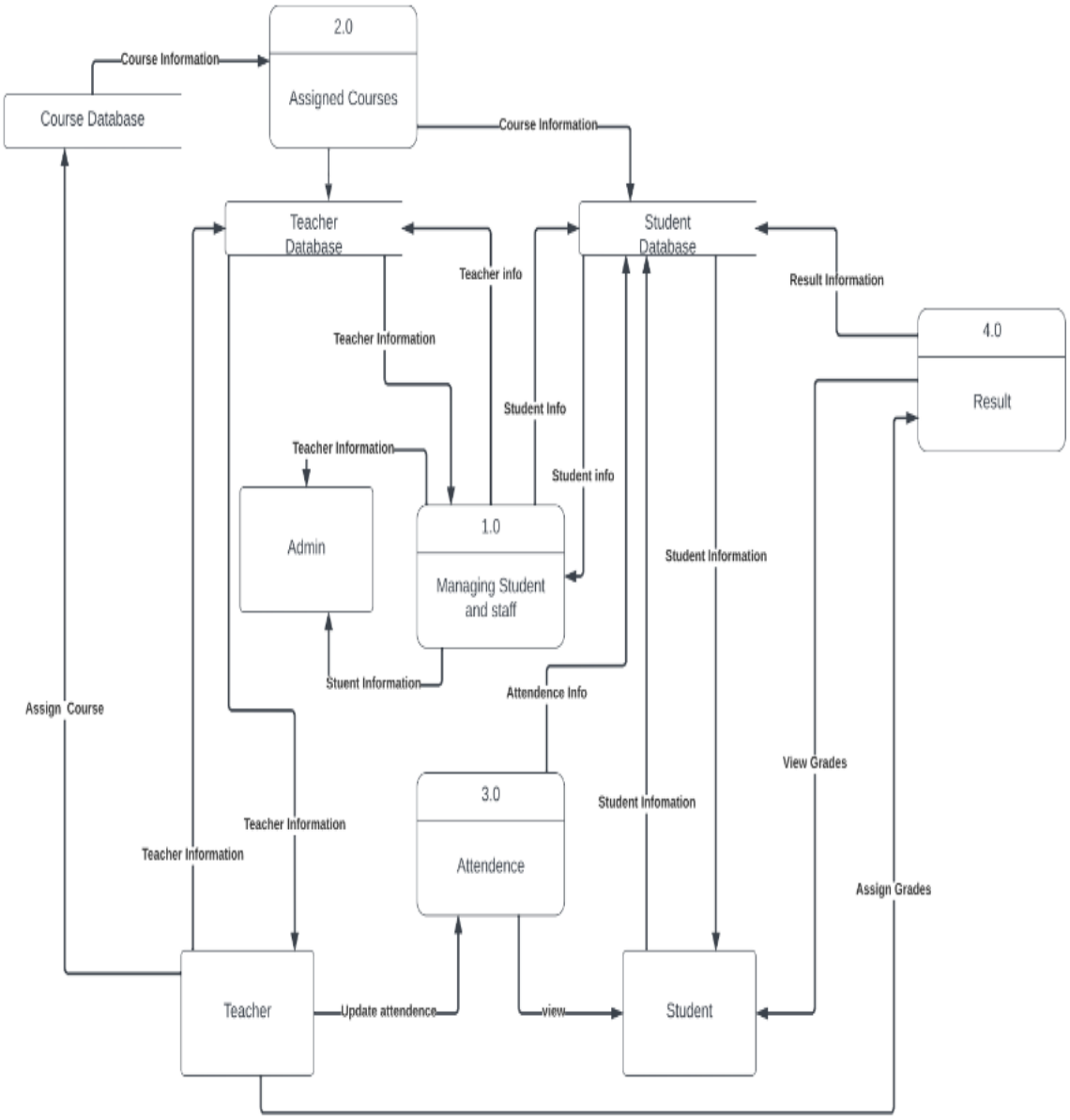

Fig 3-3: DFD DIAGRAM (LEVEL 2)

3.4.2 E R Diagram

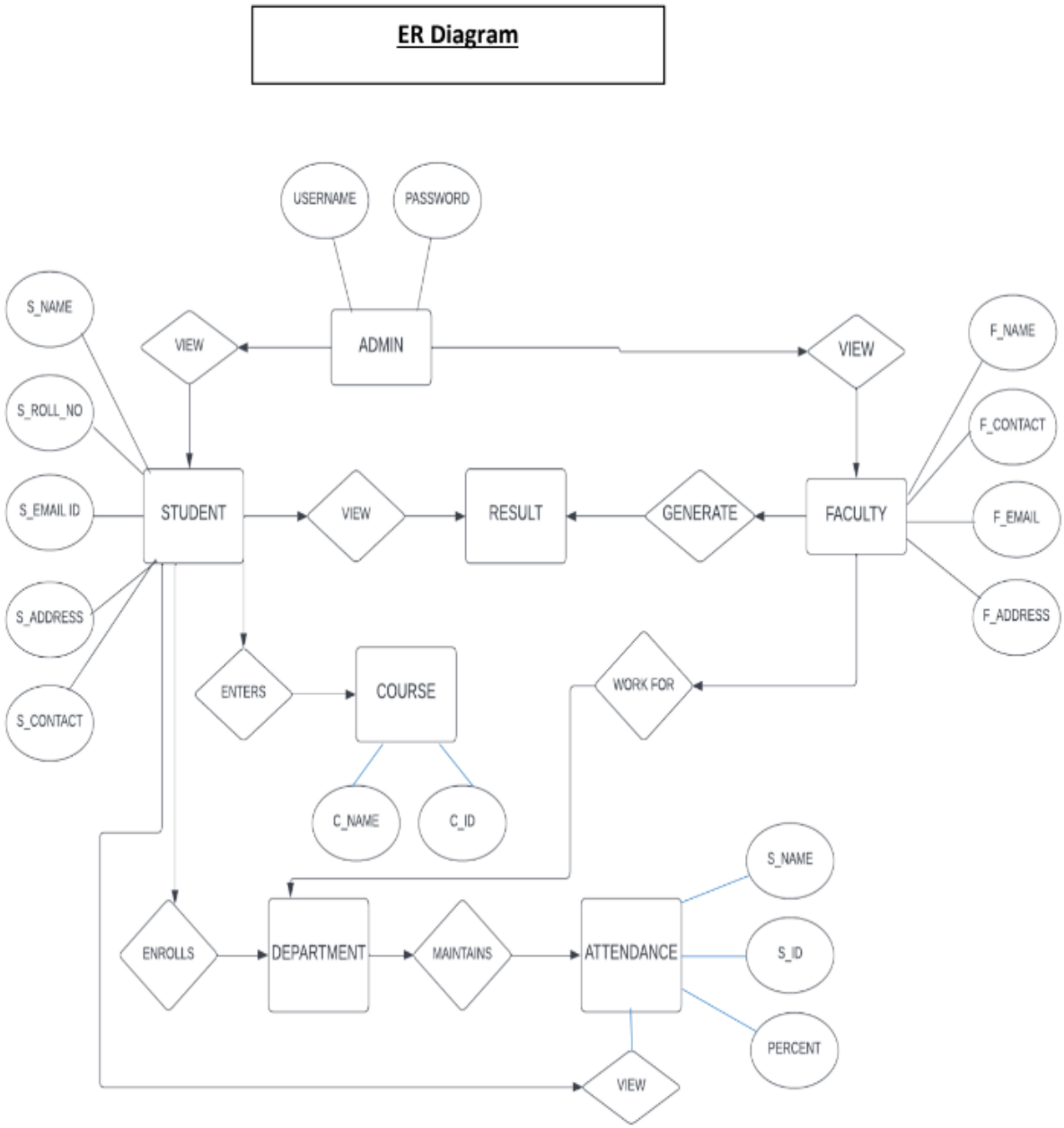

Fig 3-4: ER Diagram

3.4.3 Use Case Diagram

Fig 3-5: Use Case Diagram

3.5 Deployment Requirements

There are various requirements (hardware, software and services) to successfully deploy the system. These are mentioned below :

3.5.1 Hardware

- Processor: Minimum 1 GHz; Recommended 2GHz or more.
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi).
- Hard Drive: Minimum 32 GB; Recommended 64 GB or more.
- Memory (RAM): Minimum 1 GB; Recommended 4 GB or above.
- Backup Drive- Have backup drive for your computer to ensure that you don't lose the investment you've made in your computer.

3.5.2 Software

- Basic Text or HTML Editor
- Web Browsers
- Graphics Editor
- FTP Client-Need an FTP (file transfer protocol) client to transfer your HTML files and supporting images and graphics to your web server.

Chapter 4. Implementation

Implementation

4.1 Technology And Languages Used

4.1.1 Front end-

- **Bootstrap**

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains HTML, CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. Bootstrap is an HTML, CSS & JS Library that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to a project, Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

- **Apex Charts**

Apex chart is a modern charting library that helps developers to create beautiful and interactive visualizations for web pages.

It is an open-source project licensed under MIT and is free to use in commercial applications.

- **HTML -**It stands for 'HYPERTEXT MARKUP LANGUAGE'. HTML is a standardized system for tagging text files that creates the structure for just about every page that we find and use on the web. It's HTML that adds in page breaks, paragraphs, bold lettering, italics, and more. HTML works to build this structure by using tags that tell browsers what to do with text.

- **CSS –** It stands for 'CASCADING STYLE SHEET.' CSS is used for defining the styles for web pages. It describes the look and formatting of a document which is written in a markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It is easier to make the web pages presentable using CSS. It is easy to learn and understand and used to control the presentation of an HTML document. CSS helps us to control the text color, font style, the spacing between paragraphs, sizing of columns, layout designs, and many more. It is independent of HTML, and we can use it with any XML-based markup language.

  It is recommended to use CSS because the HTML attributes are being deprecated. So, for making HTML pages compatible with future browsers, it is good to start using CSS in HTML pages.

- **JavaScript -** JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

4.1.2 BACK-END –

- **Python –**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

- **Django –**

Django (/ˈdʒæŋgoʊ/ jang-goh) is a free and open-source web application framework, written in Python. A web framework is a set of components that helps you to develop websites faster

and easier. When you're building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc. Luckily for you, other people long ago noticed that web developers face similar problems when building a new site, so they teamed up and created frameworks (Django being one of them) that give you ready-made components to use. Frameworks exist to save you from having to reinvent the wheel and to help alleviate some of the overhead when you are building a new site.

4.2 Tools Used

4.2.1 SQLite -

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is a database, which is zero-configured, which means like other databases you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

4.3 Screenshots

The Following are the screenshots of the result of the project:

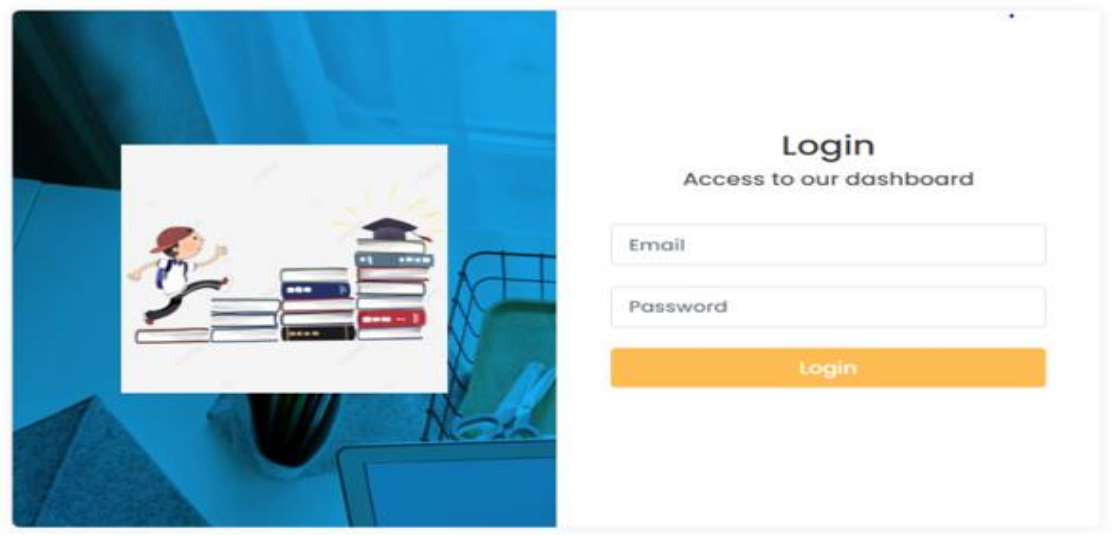

Fig 4-1: Login Page

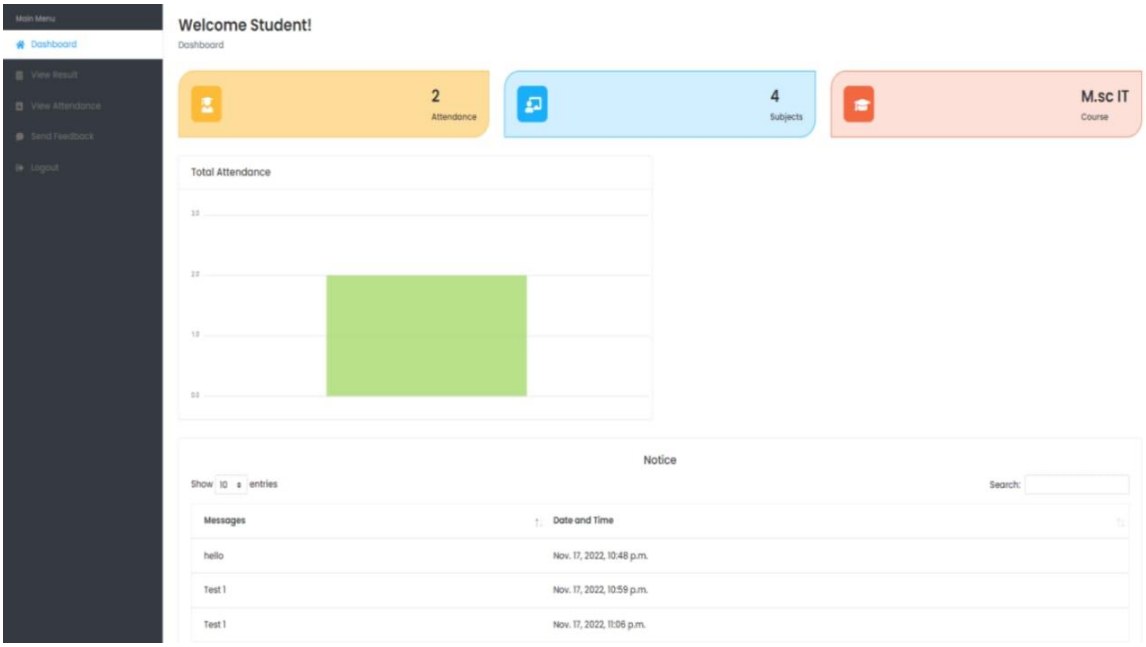

Fig 4-2: Student Dashboard

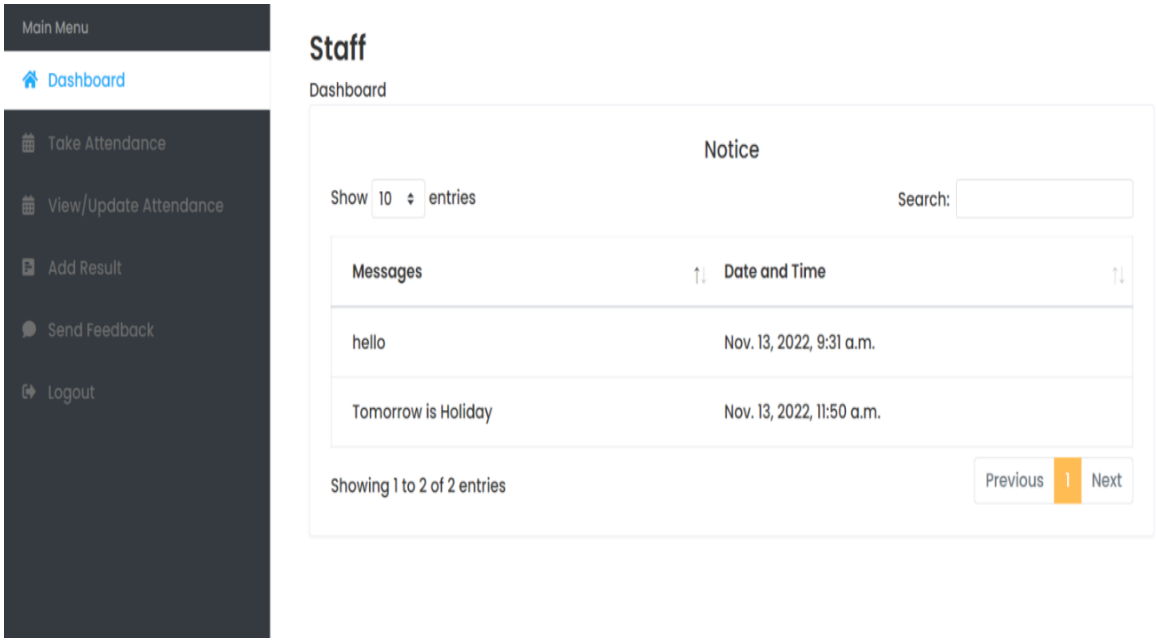

Fig 4-3: Staff Dashboard

Fig 4-4: Admin Dashboard

4.4 Testing

Testing is the process of evaluation of a system to detect differences between given input and expected output and to assess the feature of the system. Testing assesses the quality of the product. It is a process that is done during the development process.

4.4.1 Strategy Used

Tests can be conducted based on two approaches –

- Functionality testing
- Implementation testing

The texting method used here is Black Box Testing. It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise.

4.4.2 Test Case and Analysis

- **TEST CASE: 1**

Test Case ID	TC001
Test Case Summary	The information about the student can be edited and added by the administrator.
Expected Result	The administrator can easily and without any issues alter the student's details.
Actual Result	The admin can do so.
Status	PASS

Table 4-1:Test Case 1

TEST CASE 1 INPUT

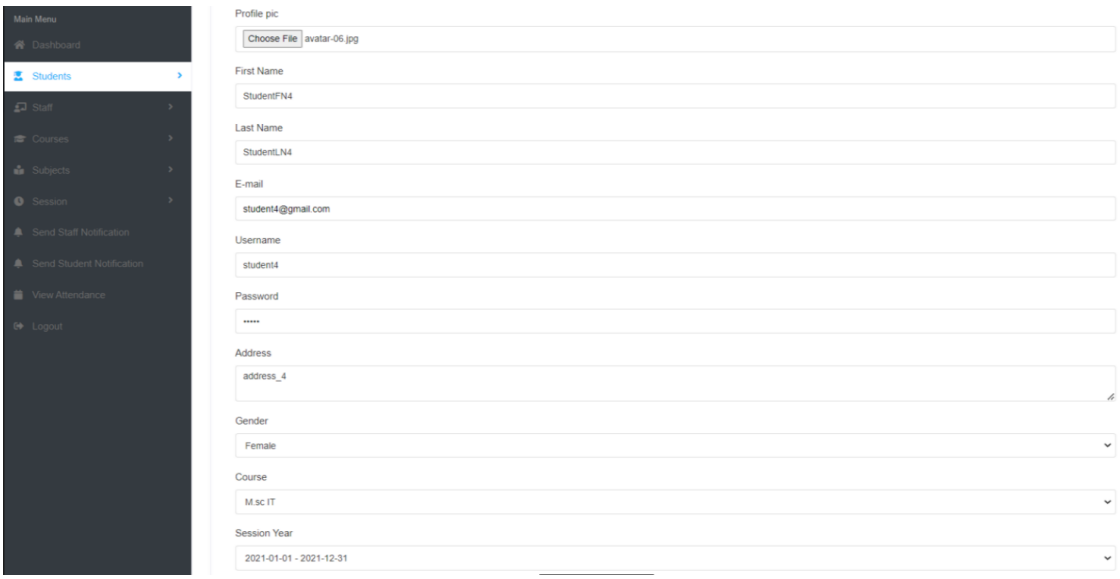

Fig 4-5: Admin adding student

TEST CASE 1 OUTPUT

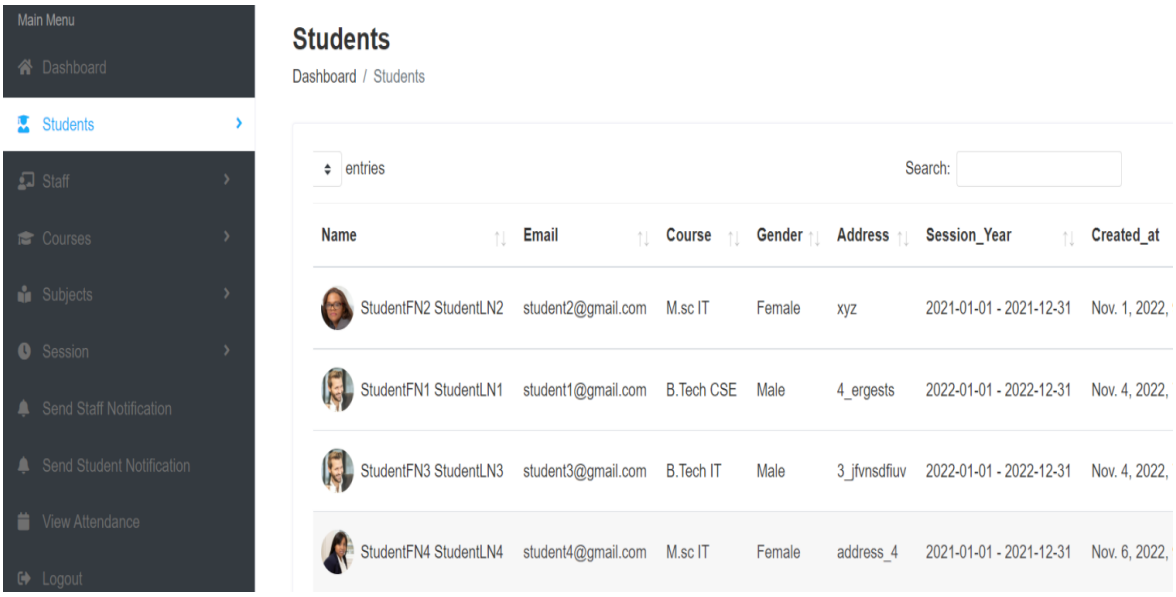

Fig 4-6: view Students

- **TEST CASE: 2**

Test Case ID	TC002
Test CaseSummary	Staff notice was sent by administration.
Expected Result	The administrator can easily send notices to staff and Staff is getting the Notice.
Actual Result	The admin can send the Notice and Staff gets it.
Status	PASS

Table 4-2: Test Case 2

TEST CASE 2 INPUT

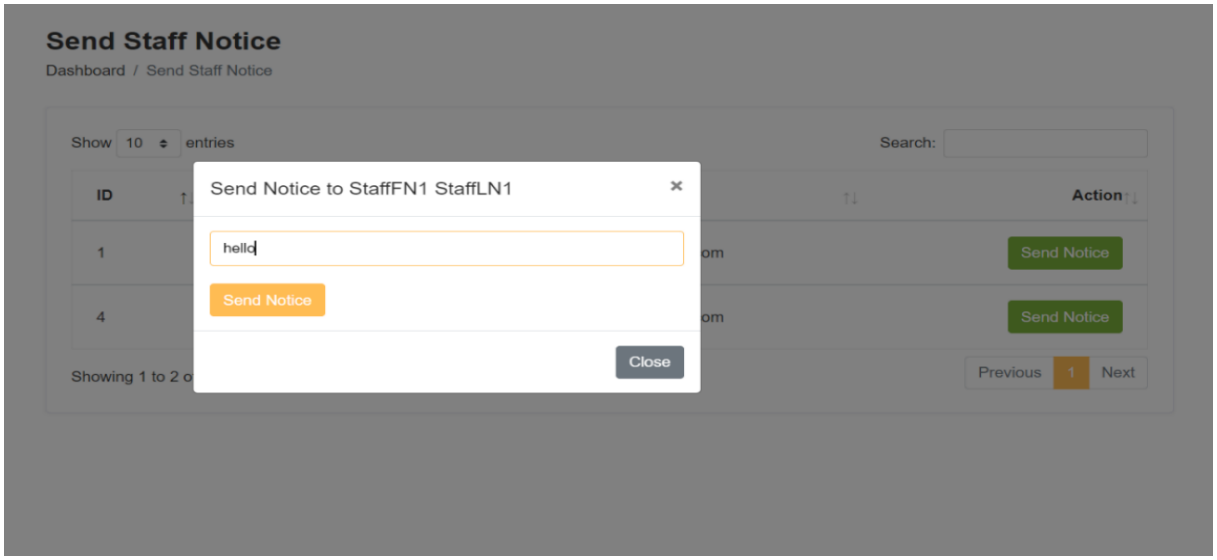

Fig 4-7: Admin send Notice

TEST CASE 2 OUTPUT

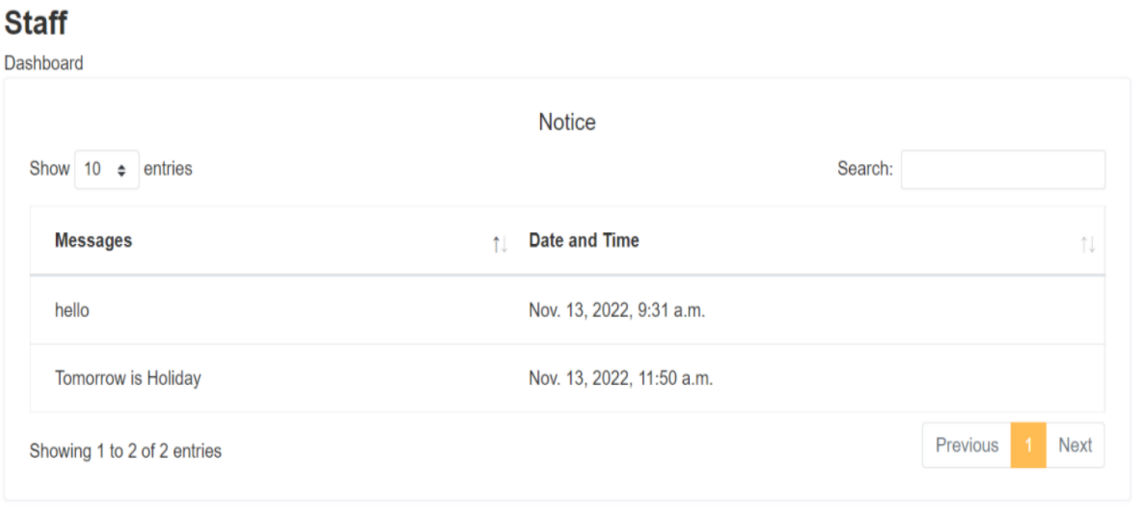

Fig 4-8: Staff receives Notice

- **TEST CASE:** 3

Test Case ID	TC003
Test CaseSummary	Staff Add Result
Expected Result	Staff can add result and student is able to view the result
Actual Result	The staff can do and student is able to view the result.
Status	PASS

Table 4-3: Test Case 3

TEST CASE 3 INPUT

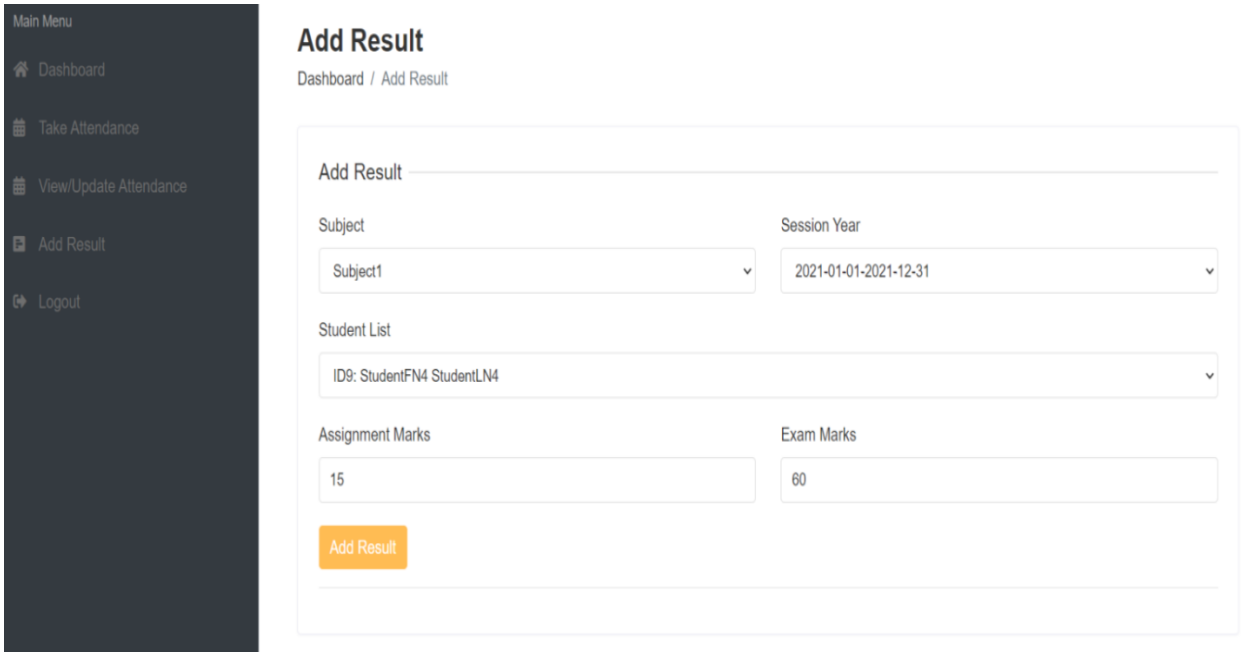

Fig 4-9: Staff Add Result

TEST CASE 3 OUTPUT

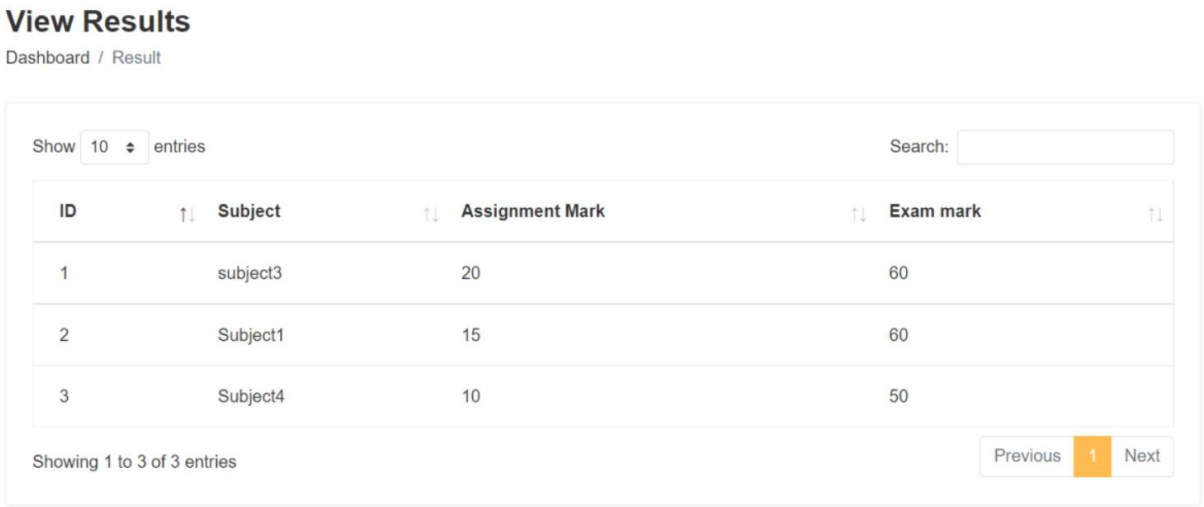

Fig 4-10: Student view Result

Chapter 5 Conclusion

Conclusion

5.1 Conclusion

Monitoring student progress educational establishments can easily retain their student records by using the system. This system aids in preserving the organization's students' academic success. It keeps track of students' progress and is simple for teachers and students to access. The manual system makes it difficult to accomplish this goal because the information is dispersed, often redundant, and gathering pertinent data might take a lot of time. This project resolves each of these issues.

5.2 Limitations of the Work

One major limitation is there are already existing websites providing such facilities that's why our focus will be on making it more user friendly.

5.3 Suggestion and Recommendations for Future Work

The website interface would be more variant.

- More progress tracking features.
- There should be more graphical representation.

Bibliography

[1] “Web Based Project Collaboration, Monitoring and Management System” (ICTer)-109-155/ 2014 IEEE.

[2] Software project management / Bob Hughes and Mike Cotterell, London [etc.]: McGraw-Hill, c2002, 3rd Ed.

[3] Information systems project management: methods, tools and techniques / John McManus and Trevor Wood- Harper, Harlow [etc.] : Prentice Hall, c2003

[4] Subversion version control: using the Subversion version control system in development projects / William Nagel, Upper Saddle River (N.J.): Prentice Hall/PTR, c2005

[5] Systems Analysis and Design Shelly Cashman Adamski Boston 1991

[6] Software Engineering Roger S.Pressman UK, c2000, 5th Ed.

[7] Samakova J, Koltnerova K, & Rybansky R, (2012). Project Communication in Functions, Process and Project-Oriented Industrial Companies,20(Special Number):120-125.

[8] Aladwani, A.M. (2002). IT project uncertainty planning and success: An empirical investigation from Kuwait. Information Technology & People, 15(3), 210-226.

[9] S.R.Bharmagoudar, Geeta R.B, Totad S.G, (2013). Web based student information system, International Journal of Advanced Research in Computer and Communication Engineering 2278-1021.

[10] Brock S, Hendricks D, Linnell S, & Smith D. (2003). A balanced approach to IT project management. Proceedings of the 2003 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on Enablement through Technology (pp. 2-10).

[11] Du, S. M. Johnson, & Kiel M. (2004). Project management courses in IS graduate programs: What is being taught, Journal of Information Systems Education, 15(2), 181-187.

[12] Stuart Palmer., (2005). An Evaluation of On-Line Assignment Submission, marking and Return, University of Melbourne 34(2), 57-67.

[13] Ellen, N., & West J., (2003). Classroom management of project management: A review of approaches to managing a student’s information system project development. Journal of

American Academy of Business, 3(1/2), 93 -97.

[14] William ibbs C., Young hoon kwak., (2000). Assessing project management maturity, The George Washington University, School of Business and Public Management, Department of Management Science volume 31(1), 32-43.

Source Code

Models.py:

```
from email.headerregistry import Address
from email.policy import default
from unittest.util import _MAX_LENGTH
from django.db import models
from django.contrib.auth.models import AbstractUser



class CustomUser(AbstractUser):
   USER =(
      (1,'HOD'),
      (2,'STAFF'),
      (3,'STUDENT'),

   )

   user_type= models.CharField(choices=USER,max_length=40,default=1)
   profile_pic = models.ImageField(upload_to='media/profile_pic')

class Course(models.Model):
    name = models.CharField(max_length=100)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.name

class Session_Year(models.Model):
    session_start = models.CharField(max_length=100)
    session_end = models.CharField(max_length=100)
```

```
    def __str__(self):
        return self.session_start+ "-" + self.session_end

class Student(models.Model):
    admin = models.OneToOneField(CustomUser,on_delete=models.CASCADE)
    address = models.TextField()
    gender = models.CharField(max_length=100)
    course_id = models.ForeignKey(Course,on_delete=models.DO_NOTHING)
    session_year_id = models.ForeignKey(Session_Year,on_delete=models.DO_NOTHING)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.admin.first_name + " " + self.admin.last_name

class Staff(models.Model):
    admin = models.OneToOneField(CustomUser,on_delete=models.CASCADE)
    address = models.TextField()
    gender = models.CharField(max_length=100)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.admin.username

class Subject(models.Model):
    name = models.CharField(max_length=100)
    course = models.ForeignKey(Course,on_delete=models.CASCADE)
    staff = models.ForeignKey(Staff,on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True,null=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.name
```

```
class Staff_Notification(models.Model):
    staff_id = models.ForeignKey(Staff,on_delete=models.CASCADE)
    message = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return self.staff_id.admin.first_name

class Student_Notification(models.Model):
    student_id = models.ForeignKey(Student,on_delete=models.CASCADE)
    message = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    def __str__(self):
        return self.student_id.admin.first_name

class Attendance(models.Model):
    subject_id = models.ForeignKey(Subject,on_delete=models.DO_NOTHING)
    attendance_date = models.DateField()
    session_year_id = models.ForeignKey(Session_Year,on_delete=models.DO_NOTHING)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.subject_id.name

class Attendance_Report(models.Model):
    student_id = models.ForeignKey(Student,on_delete=models.DO_NOTHING)
    attendance_id = models.ForeignKey(Attendance,on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.student_id.admin.first_name + " " + self.student_id.admin.last_name

class StudentResult(models.Model):
    student_id = models.ForeignKey(Student,on_delete=models.CASCADE)
```

```
    subject_id = models.ForeignKey(Subject,on_delete=models.CASCADE)
    assignment_mark = models.IntegerField()
    exam_mark = models.IntegerField()
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.student_id.admin.first_name
```

Admin DashBoard[backend]:

```
def HOME(request):
    student_count = Student.objects.all().count()
    staff_count = Staff.objects.all().count()
    course_count = Course.objects.all().count()
    subject_count = Subject.objects.all().count()

    student_gender_male = Student.objects.filter(gender = 'Male').count()
    student_gender_female = Student.objects.filter(gender = 'Female').count()


    context = {
        'student_count' : student_count,
        'staff_count' : staff_count,
        'course_count' : course_count,
        'subject_count' : subject_count,
        'student_gender_male' : student_gender_male,
        'student_gender_female' : student_gender_female,
    }
    return render(request,'Hod/home.html',context)
```

Admin DashBoard[frontend]:

```
{% extends 'base.html' %}
{% block content %}

<div class="page-header">
  <div class="row">
    <div class="col-sm-12">
      <h3 class="page-title">Welcome Admin!</h3>
      <ul class="breadcrumb">
        <li class="breadcrumb-item active">Dashboard</li>
      </ul>
    </div>
  </div>
</div>
<div class="row">
  <div class="col-xl-3 col-sm-6 col-12 d-flex">
    <div class="card bg-one w-100">
      <div class="card-body">
        <div class="db-widgets d-flex justify-content-between align-items-center">
          <div class="db-icon">
            <i class="fas fa-user-graduate"></i>
          </div>
          <div class="db-info">
            <h3>{{student_count}}</h3>
            <h6>Students</h6>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="col-xl-3 col-sm-6 col-12 d-flex">
    <div class="card bg-two w-100">
      <div class="card-body">
        <div class="db-widgets d-flex justify-content-between align-items-center">
          <div class="db-icon">
```

```
            <i class="fas fa-chalkboard-teacher"></i>
          </div>
          <div class="db-info">
            <h3>{{staff_count}}</h3>
            <h6>Staff</h6>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="col-xl-3 col-sm-6 col-12 d-flex">
    <div class="card bg-three w-100">
      <div class="card-body">
        <div class="db-widgets d-flex justify-content-between align-items-center">
          <div class="db-icon">
            <i class="fas fa-graduation-cap"></i>
          </div>
          <div class="db-info">
            <h3>{{course_count}}</h3>
            <h6>Course</h6>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div class="col-xl-3 col-sm-6 col-12 d-flex">
    <div class="card bg-four w-100">
      <div class="card-body">
        <div class="db-widgets d-flex justify-content-between align-items-center">
          <div class="db-icon">
            <i class="fas fa-book"></i>
          </div>
          <div class="db-info">
            <h3>{{subject_count}}</h3>
            <h6>Subject</h6>
```

```
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="row">
  <div class="col-md-12 col-lg-6">
    <div class="card card-chart">
      <div class="card-header">
        <div class="row align-items-center">
          <div class="col-6">
            <h5 class="card-title">All Data</h5>
          </div>
        </div>
      </div>
      <div class="card-body">
        <div id="all_details"></div>
      </div>
    </div>
  </div>
  <div class="col-md-12 col-lg-6">
    <div class="card card-chart">
      <div class="card-header">
        <div class="row align-items-center">
          <div class="col-6">
            <h5 class="card-title">Number of Students</h5>
          </div>
        </div>
      </div>
      <div class="card-body">
        <div id="stud_gender"></div>
      </div>
    </div>
  </div>
```

```
 </div>
</div>
{% include 'includes/chart.html' %}
{% endblock %}
```

Chart.html

```
<script>
   $(document).ready(function() {

      // Area chart

      var options = {
         series: [{{student_count}}, {{staff_count}}, {{subject_count}}, {{course_count}}],
         chart: {
            height: 350,
            width: '100%',
         type: 'pie',
       },
       labels: ['Student', 'Staff', 'Subject', 'Course',],
       responsive: [{
        breakpoint: 480,
        options: {
          chart: {
            width: 200
          },
          legend: {
            position: 'bottom'
          }
        }
       }]
       };

       var chart = new ApexCharts(document.querySelector("#all_details"), options);
       chart.render();

      // Bar chart
```

```
if ($('#stud_gender').length > 0) {
var optionsBar = {
   chart: {
      type: 'bar',
      height: 350,
      width: '100%',
      stacked: true,
      toolbar: {
         show: false
      },
   },
   dataLabels: {
      enabled: false
   },
   plotOptions: {
      bar: {
         columnWidth: '45%',
      }
   },
   series: [{
      name: "Boys",
      color: '#fdbb38',
      data: [{{student_gender_male}}],
   }, {
      name: "Girls",
      color: '#19affb',
      data: [{{student_gender_female}}],
   }],
   labels: [2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020],
   xaxis: {
      labels: {
         show: false
      },
      axisBorder: {
```

```
            show: false
          },
          axisTicks: {
            show: false
          },
        },
        yaxis: {
          axisBorder: {
            show: false
          },
          axisTicks: {
            show: false
          },
          labels: {
            style: {
              colors: '#777'
            }
          }
        },
        title: {
          text: '',
          align: 'left',
          style: {
            fontSize: '18px'
          }
        }

      }

      var chartBar = new ApexCharts(document.querySelector('#stud_gender'), optionsBar);
      chartBar.render();
      }

    });
</script>
```

Staff DashBoard[Backend]:

```
def HOME(request):
   staff = Staff.objects.filter(admin = request.user.id)
   for i in staff:
      staff_id = i.id
      notification = Staff_Notification.objects.filter(staff_id = staff_id)

      context = {
         'notification':notification,
      }
   return render(request,'Staff/home.html',context)
```

Staff DashBoard[frontend]:

```
{% extends 'base.html' %}
{% block content %}

<div class="content container-fluid">
   <div class="page-header">
    <div class="row align-items-center">
      <div class="col">
        <h3 class="page-title">Staff</h3>
        <ul class="breadcrumb">
          <li class="breadcrumb-item"><a href="index.html">Dashboard</a></li>
        </ul>
      </div>
      {% include 'includes/messages.html'%}

  </div>
  <div class="row">
    <div class="col-sm-12">
      <div class="card card-table">
        <div class="card-body">
          <div class="table-responsive">
            <table id="table_id" class="table table-hover table-center mb-0 ">
              <h5 class="card-title" align="center">Notice</h5>
```

```
            <thead>
              <tr>
                <th>Messages</th>
                <th>Date and Time</th>
              </tr>
            </thead>
            <tbody>
            {% for i in notification %}
              <tr>
                <td>{{i.message}}</td>
                <td>{{i.created_at}}</td>

              </tr>
              {% endfor %}
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </div>

  </div>
{% endblock %}
```

Student DashBoard[Backend]:

```
def HOME(request):
    student_id = Student.objects.get(admin = request.user.id)
    Attendance_total = Attendance_Report.objects.filter(student_id = student_id).count()
    course = Course.objects.get(id=student_id.course_id.id )
    Subjects = Subject.objects.filter(course = course).count()
    session_Year = Session_Year.objects.get(id=student_id.session_year_id.id)
    student = Student.objects.filter(admin = request.user.id)
    for i in student:
        student_id = i.id
```

```
    notification = Student_Notification.objects.filter(student_id = student_id)

  context = {
    'Attendance_total' : Attendance_total,
    'Subjects' : Subjects,
    'course' : course,
    'session_Year' : session_Year,
    'session_Year' : session_Year,
    'notification':notification,
  }


  return render(request,'Student/home.html',context)
```

Student DashBoard[frontend]:

```
{% extends 'base.html' %}
{%block content %}

<div class="page-header">
  <div class="row">
    <div class="col-sm-12">
      <h3 class="page-title">Welcome Student!</h3>
      <ul class="breadcrumb">
        <li class="breadcrumb-item active">Dashboard</li>
      </ul>
    </div>
  </div>
  {% include 'includes/messages.html'%}
</div>
<div class="row">
  <div class="col-xl-4 col-sm-6 col-12 d-flex">
    <div class="card bg-one w-100">
      <div class="card-body">
        <div class="db-widgets d-flex justify-content-between align-items-center">
          <div class="db-icon">
```

```
            <i class="fas fa-user-graduate"></i>
          </div>
          <div class="db-info">
            <h3>{{Attendance_total}}</h3>
            <h6>Attendance</h6>
          </div>
        </div>
      </div>
    </div>
  <div class="col-xl-4 col-sm-6 col-12 d-flex">
    <div class="card bg-two w-100">
      <div class="card-body">
        <div class="db-widgets d-flex justify-content-between align-items-center">
          <div class="db-icon">
            <i class="fas fa-chalkboard-teacher"></i>
          </div>
          <div class="db-info">
            <h3>{{Subjects}}</h3>
            <h6>Subjects</h6>
          </div>
        </div>
      </div>
    </div>
  <div class="col-xl-4 col-sm-6 col-12 d-flex">
    <div class="card bg-three w-100">
      <div class="card-body">
        <div class="db-widgets d-flex justify-content-between align-items-center">
          <div class="db-icon">
            <i class="fas fa-graduation-cap"></i>
          </div>
          <div class="db-info">
            <h3>{{course}}</h3>
            <h6>Course</h6>
```

```
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
<div class="row">
  <div class="col-md-12 col-lg-6 align-center">
    <div class="card card-chart">
      <div class="card-header">
        <div class="row align-items-center">
          <div class="col-6">
            <h5 class="card-title">Total Attendance</h5>
          </div>
        </div>
      </div>
      <div class="card-body">
        <div id="stud_attendance"></div>
      </div>
    </div>
  </div>

  <div class="col-sm-12">
    <div class="card card-table">
      <div class="card-body">
        <div class="table-responsive">
          <table id="table_id" class="table table-hover table-center mb-0 ">
            <h5 class="card-title" align="center">Notice</h5>
            <thead>
              <tr>
                <th>Messages</th>
                <th>Date and Time</th>
              </tr>
            </thead>
            <tbody>
```

```
            {% for i in notification %}
              <tr>
                <td>{{i.message}}</td>
                <td>{{i.created_at}}</td>

              </tr>
              {% endfor %}
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </div>
</div>

</div>
 {% include 'includes/studentChart.html' %}
{% endblock %}
```

StudentChart.html:

```
<script>
   $(document).ready(function() {

      // Bar chart

      if ($('#stud_attendance').length > 0) {
      var optionsBar = {
         chart: {
            type: 'bar',
            height: 350,
            width: '100%',
            stacked: true,
            toolbar: {
               show: false
            },
```

```
},
dataLabels: {
   enabled: false
},
plotOptions: {
   bar: {
      columnWidth: '45%',
   }
},
series: [{
   name: "count",
   color: '#abdd75',
   data: [{{Attendance_total}}],
}],
labels: ['Attendance'],
xaxis: {
   labels: {
      show: false
   },
   axisBorder: {
      show: false
   },
   axisTicks: {
      show: false
   },
},
yaxis: {
   axisBorder: {
      show: false
   },
   axisTicks: {
      show: false
   },
   labels: {
      style: {
```

```
                colors: '#777'
              }
            }
          },
          title: {
            text: '',
            align: 'left',
            style: {
              fontSize: '18px'
            }
          }

        }

        var chartBar = new ApexCharts(document.querySelector('#stud_attendance'), optionsBar);
        chartBar.render();
        }
      });
</script>
```