**INFORMATION SYSTEM**

**Table of Contents**

[Introduction 3](#_Toc164705444)

[Implementation 3](#_Toc164705445)

[Result 4](#_Toc164705446)

[Conclusion 6](#_Toc164705447)

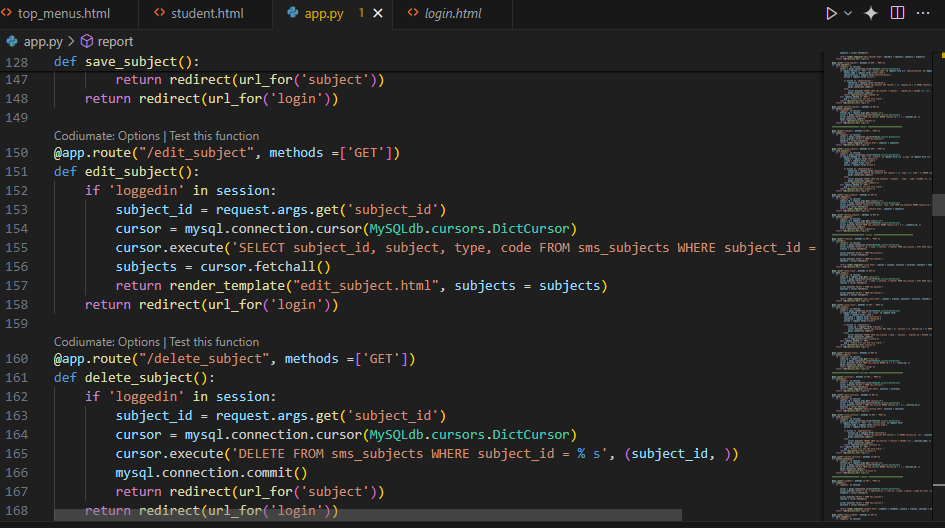
[References 7](#_Toc164705448)

# Introduction

The implementation of reporting mechanism on the student management system provides direction to utilizing statistical modeling in the educational frameworks. This function allows the admin stuff to see different parts of the pupils' accomplishments, participant, and academic performance. The system facilitates customizable and specific reports that can be adjusted to the specific needs of each user. It is because of the system, the administrative workload is expected to decrease, and preventive measures are created to boost the result of the students' achievement. Thereby, the flask functions on the backend and frontend (HTML, CSS, JavaScript) will be cleanly integrated. Users will be able to generate a user interface, which is easily available and meaningful for all.

# Implementation

Deal reporting feature will be available in the student command system which the developers will follow a systematic route. The first step would be to create a new endpoint in the Flask backend that will serve as the point where the request for report generation is processed and the endpoint is /report. This goes along with the backend that does querying the database in order to obtain real data utilized for making a report. In the report we can include factors like student attendance records, academic performance or any other the information which is relevant for the report.

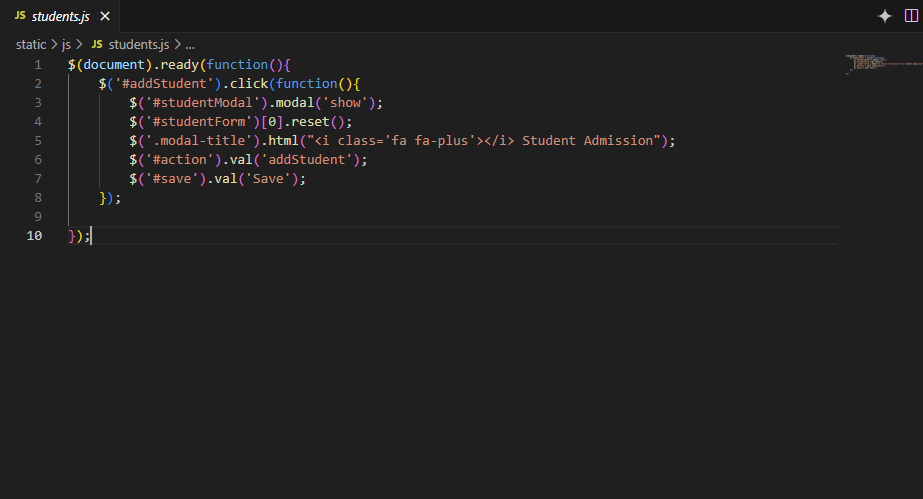


**Figure 1: Implement Subject arrangement**

(Source: self-created)

In the place of them, the data will be forwarded to the front end for observation. In this case an HTML, CSS and JavaScript codings will be required for the structure of the template to be accomplished (Zanevych, 2024). It would be the JavaScript libraries like React and Vue.Js handling the job of showing information in clean and stunning way in the front section.

The page should let users choose definite parameters for creating a report such as class, section, date range or some others. (Idris *et al.* 2020). This can be done by using a front interface with dropdown menus, input fields, or date pickers.



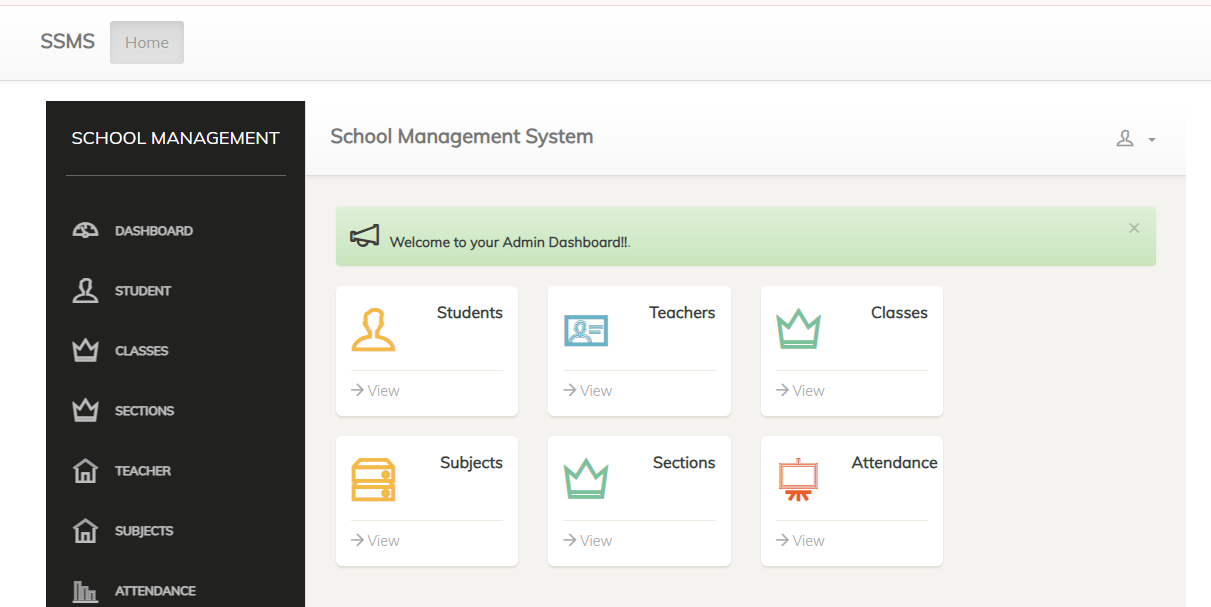
**Figure 2: Student data Event Handlar**

(Source: self-created)

Then the user feeds in their custom choices and the back-end will process this information, and it will query the database for all the data that is supposed to be rendered back to the front-end. After this, the front end of the system will display the data that was retrieved dynamically on report page using different types of format such as table, chart or graph depending on the type of report being generated.

# Result

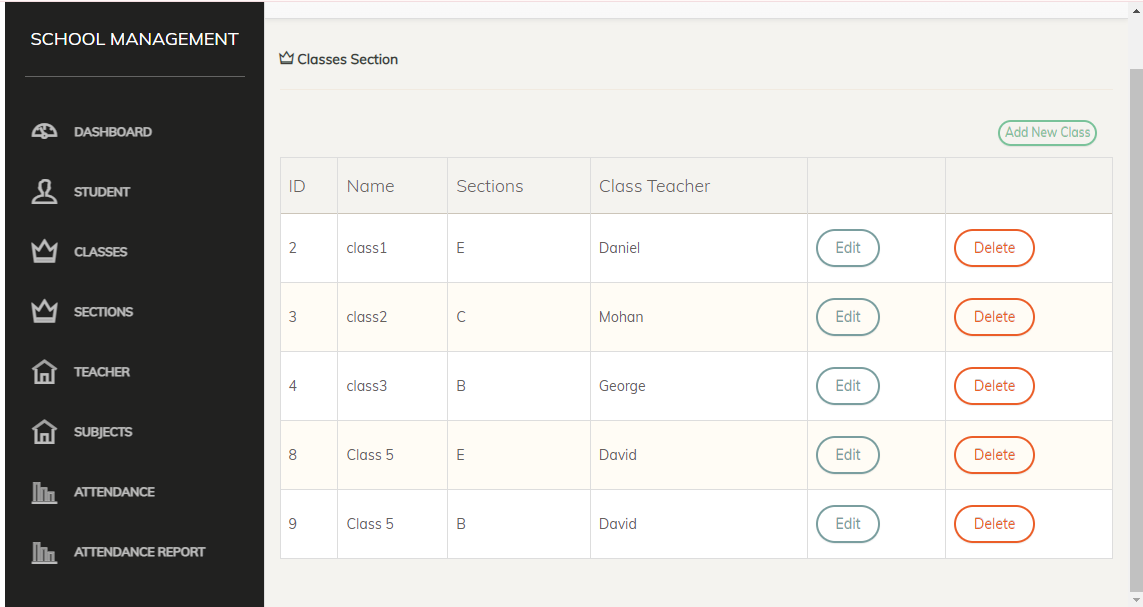
The student management system has a report feature that enables data to be easily viewed and analyzed amongst many aspects of students' data. The users can set parameters in a way so that they can capture the classes, sections, and date ranges of their interest as per their needs. When the system receives student submission, it gets this information from the database, such as student’s attendance records, academic performance and other vital data (Anggoro and Aziz, 2021).



**Figure 3: Admin Dashboard**

(Source: self-created)

Finally, this data is then dynamically displayed on the report page which is a structured format displaying the data in table or graph.



**Figure 4: Admin Class Dashboard**

(Source: self-created)

Stakeholders e.g. the administrators, teachers, and other interested parties can use these reports data to inquire on the student’s progress, attendance trends, and overall performance allowing them to make sound judgments and effective utilization of educational resources.

# Conclusion

In the end, the release of the report thinking nature student management system would be an important development in the tracking and analysis of student information. Such a system permits managers and teachers to generate customized reports with the particular parameters used for the report, including class, section and date range, which increases transparency and accountability in the learning process of the students. The unified Flask backends with frontend HTML, CSS, and JavaScript help roll out a user friendly interface whereby fetching and presenting of data is efficient.

# References

Anggoro, D.A. and Aziz, N.C., 2021. Implementation of K-nearest neighbors algorithm for predicting heart disease using python flask. Iraqi Journal of Science, pp.3196-3219.

Idris, N., Foozy, C.F.M. and Shamala, P., 2020. A generic review of web technology: Django and flask. International Journal of Advanced Science Computing and Engineering, 2(1), pp.34-40.

Zanevych, Y., 2024. FLASK VS. DJANGO VS. SPRING BOOT: NAVIGATING FRAMEWORK CHOICES FOR MACHINE LEARNING OBJECT DETECTION PROJECTS. Collection of scientific papers «ΛΌГOΣ», (March 29, 2024; Cambridge, UK), pp.311-318.