Tribhuvan University

Faculty of Humanities and Social Sciences

Student Performance Analyzer

Using Decision Tree Algorithm

A PROJECT Report

Submitted to

Department of Computer Application

Asian College of Higher Studies

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Bipul Tamang (190619)

Under the Supervision of

Smita Ghimire

Abstract

The “Student Performance Analyzer” is a web-based application designed to provide a platform for teachers to maintain their courses and students where they are able to entry the attendance, test conducts and marks allocation and view the summarized data with visual graphs. The web application also trains a model using decision tree algorithm based on the data fed by the teachers where the performance of student are analyzed based on their demographic values. After a completion of the trained model the users will be able to input combination of various demographic values on attributes to monitor the performance of student.

Acknowledgement

We would like to express our gratitude to supervisor, Mrs. Smita Ghimire for assisting in learning about student analyzing systems for this project "**Student Performance Analyzer**". They provided us with the opportunity to work on this fantastic project, which has allowed us to learn a lot of new things, and we are grateful to our professors for that.

Apart from our supervisor we would like to thank our professors and friends for allowing us to use the available equipment for this project.

Project Member

Bipul Tamang

Table of Contents

List of abbreviations

Cases

CASE : Computer aided software engineering 15
CRUD : Create Read Update Delete 3
SPA : Student Performance Analyzer 1

List of figures

List of Tables

Chapter 1 Introduction

1.1 Introduction

The Student Performance Analyzer (SPA) is a web application designed to provide teachers with valuable insights into student performance based on demographic values where the data of the teachers will be used to train a model and the web provides a platform for teachers to record the attendance and test result of a student. As there are fewer valuable data available to analyze the student performance based on the demographic values this application offers a tool for analyzing the student's performance based on where they graduated from and the available data can be used to improve the education sector of the place focusing with other factors such as age, gender, family status as well.

In today's education landscape it is crucial to identify factors that may influence student outcomes. By using the power of Student Performance Analyzer, it aims to shed light on patterns and correlations between student demographics and academic performance. Through the utilization of decision tree analysis teacher can help to train a model that provides the analysis of student performance based on their demographic values which can be specifically used to improve the education sectors of the key area focusing on the key problems that might be impacting the performance of the student.

In the following sections of this mid-defense report we will go through the system architecture, features, functionality, implementation details, objectives… of the Student Performance Analyzer web application.

1.2 Problem Statement

In the field of education, it is essential to have a deep understanding of student performance and factors that affect them in order to provide effective support and guidance. However, traditional methods fall short in capturing the complex relationship between student demographics and academic outcomes. As a result, educators face challenges in identifying and addressing the factors that contribute to disparities in student achievement.

The current education system also faces several challenges in monitoring the student attendance and academic scores in a modern effective way without any platform that effective data to both teachers and student. Which makes a problem for both teachers and student to evaluate their attendance and academic scores in modern way.

The lack of timely feedbacks to the student also plays an important role between the interaction of teachers and student. Due to the lack of feedback to student the student won't know the key area he/she have to improve. Most of the available Student Performance Analyzer web application are ineffective in providing a user-friendly interface to the user and provides a complex data and graphs which are of no use if the user doesn't have extensive knowledge on the working of the complex graph and data behind it.

1.3 Objectives

The main objectives of Student Performance Analyzer (SPA) are:

- To provide an online platform for teachers to record the student attendance and test marks.
- To provide a summary and graph of the student attendance and various test conducted by teacher.
- To incorporate decision tree algorithm to train the model on the data of teachers and provide the analysis of student performance based on demographic values.
- To provide a user-friendly platform by implementing CRUD operation for various tasks in a same page.
- To provide a platform for student to view their academic data including academic records and attendance.

1.4 Scope and Limitation

The scope of Student Performance Analyzer (SPA) includes:

- The system will allow students, teachers to access academic data including grades, attendance records.
- The system will be designed to be user-friendly and clean.
- The system will be able to train a model based on the data of the student and analyze the performance of the students based on demographic values.
- The system will be able to provide a summary and graph for attendance, tests as a part of data visualization.

The limitation of Student Performance Analyzer (SPA) includes:

- The system does not consider non-academic factors that impact the performance.
- The system success relies on user’s willingness to use, since accurate data plays a key role.
- The system is used for online basis only.
- The potential for algorithmic bias in algorithm analysis raises ethical considerations.

Chapter 2 Background Study and Literature Review

2.1 Background Study

The field of education has witnessed a growing interest in utilizing data analysis techniques to gain insights into student performance and enhance educational outcomes. Analyzing student performance data can provide valuable information for teachers as well as the general population and administrators to identify the areas of improvement and find out the causes of the factors that affects the education of certain demographic people and make data-driven decisions.

Existing solutions in the educational technology offer tools for student management and analysis but few focus specifically on comprehensive performance analysis at a national or country level. Therefore, the development of the "Student Performance Analyzer" aims to bridge this gap by providing a web-based application that enables teachers to record attendance and test data, and help to train a model that performs data analysis using decision tree algorithm to provide analyzed data on student performance based on their demographic factors.

By conducting a comprehensive background study, we aim to understand the challenges and opportunities in the domain of student performance analysis and develop a robust and effective web application that supports teachers in keeping track of their student and as well as making sound decisions and improving student outcomes.

2.2 Literature Review

Student performance analysis plays a critical role in education, enabling educators to gain insights into student learning and make informed decisions to enhance educational outcomes. Numerous studies emphasize the importance of student performance analysis in educational settings. By analyzing demographic information, educators can identify the performance of various student of different demographic attributes and make a sound decision to improve the education sector targeting specific key areas.

Several software systems and tools exist for student performance analysis. These systems offer features for data entry, analysis, and reporting. However, there is a need for a comprehensive and user-friendly solution that integrates decision tree algorithms with data analysis functionalities to provide teachers and administrators with actionable insights for educational decision-making.

However, the site **"Spaplatform"** was studied which is a student performance analyzer site that was designed to be used by school leaders. It provides an environment to house all your student analytics software modules in one place. Staff login once and have access to all the modules your school choose to use. (Holmes-Smith, 2013)

Chapter 3 System Analysis and Design

3.1 System Analysis

3.1.1 Requirement Analysis

i. Functional requirements

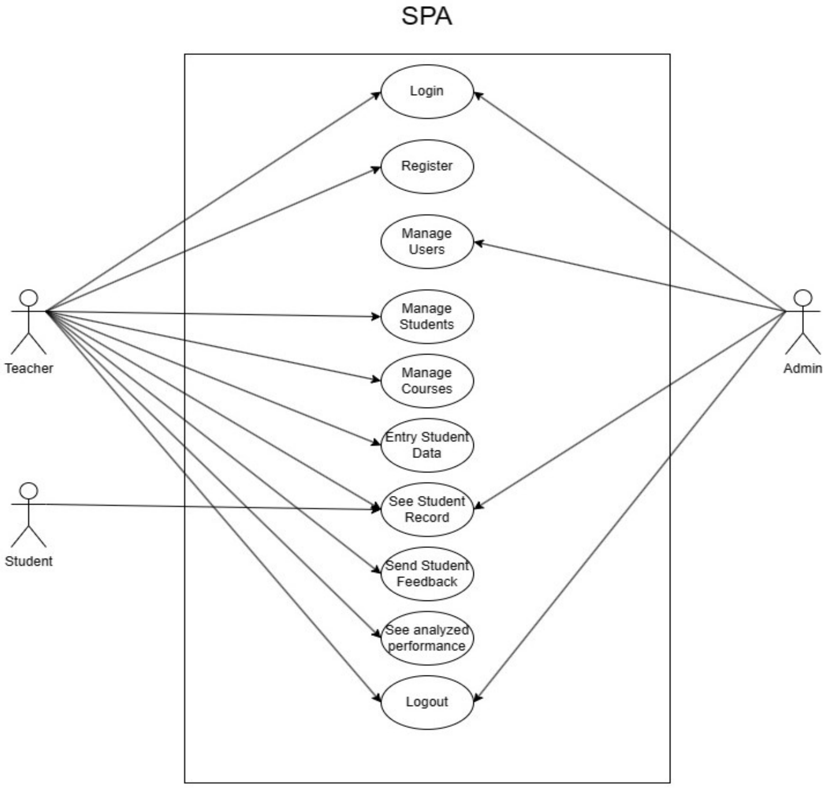


Figure 3.1.1 Use Case Diagram of SPA

Here with the above Use Case Diagram of SPA web application we can understand that the user teacher will be able to login, register along with manage students and courses he/she teach at college and will be able to entry the student data and the summary record of the student. The user will also be able to send feedback to the student and see the analyzed data of the student performance based on the demographic values. The admin will be able to monitor the uses and will be able to manage them. Whereas the student are able to see their summary record on the course.

ii. Non-functional requirements

- The system will be available online.
- The interface will be intuitive and require minimal training to perform tasks.
- The system will be able to handle increasing number of records.
- The application is compatible with latest version of commonly used browsers.
- The system will be secure and user info won’t be shared.

3.1.2 Feasibility Analysis

Following feasibilities were studied before building the system to see if the system could be built with exact requirements in the expected time.

i. Technical feasibility

In order to design the system, it uses existing technologies, software and hardware hence there’s no technical problem that might arise to build this system.

ii. Economic feasibility

The system does not require extra software and hardware i.e it uses open-source technologies. So there is no recurring cost.

iii. Operational feasibility

The system uses simple technologies to design. So it us user friendly.

iv. Legal feasibility

The system is completely legal and should not face any problem regarding illegal issues.

3.1.3 Data Modeling ER Diagram

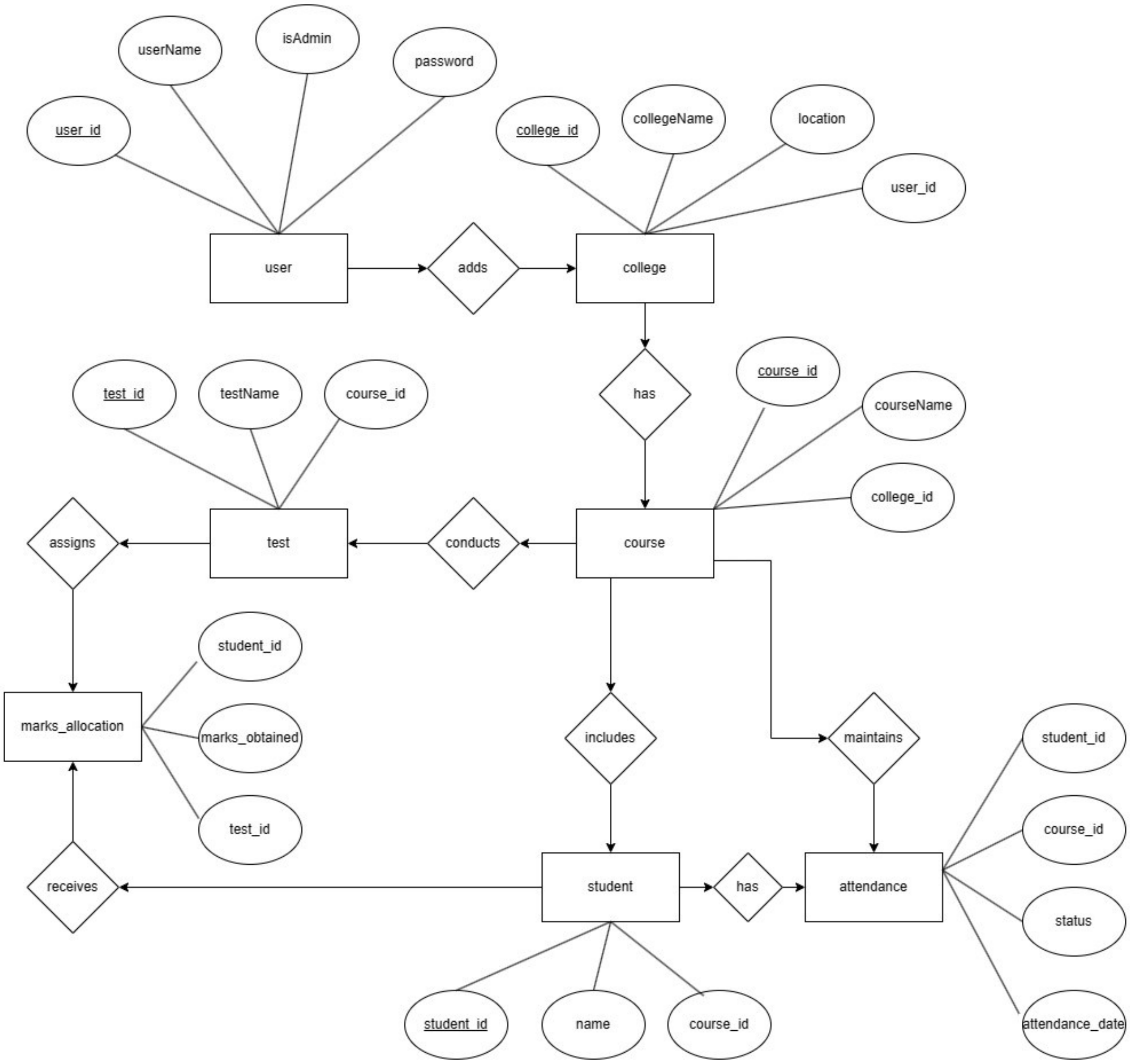


Figure 3.1.2 ER Diagram of SPA

The above entity relationship diagram shows the different entities involved in the system and relationship among them. The user is able to add the college he/she teach and along with it add the course based on the college he/she teach where each course consists of student studying on the course where test are conducted. The marks allocation and attendance records are stored on behalf of the students.

3.1.4 Process Modeling DFD

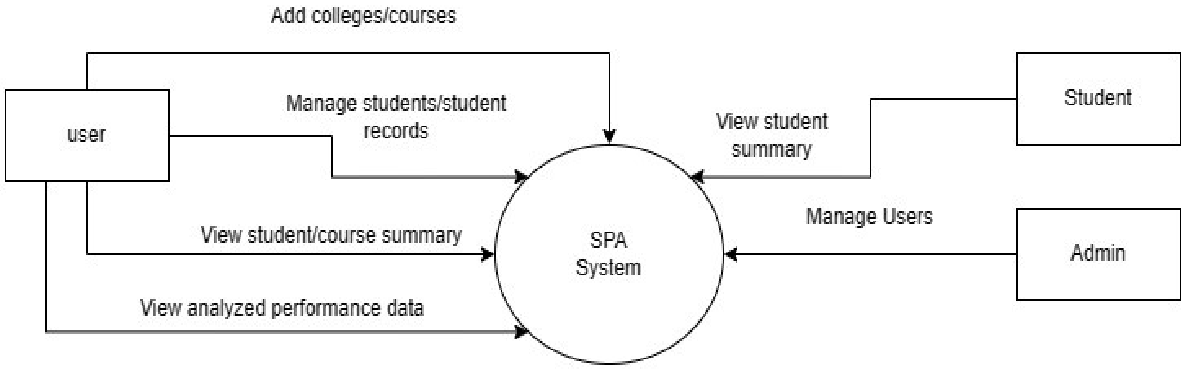

Add colleges/courses
user
Manage students/student records
View student/course summary
View analyzed performance data
SPA System
View student summary
Student
Manage Users
Admin

Figure 3.1.4 DFD Diagram

Here in the above figure we can see the process modeling DFD of our system where the user is able to perform the operations provided by SPA system to the user. The user will be able to manage his/her courses and students i.e. adding, updating, deleting the records. The user will also be able to view the course summary and student summary based on the data recorded in the system by the user. The user will also be able to view analyzed performance data of the students where the model is trained on the data feuded by the users. The admin in this system only has a role of managing users and Student only view their summary on course.

3.2 System Design

3.2.1 Architectural Design

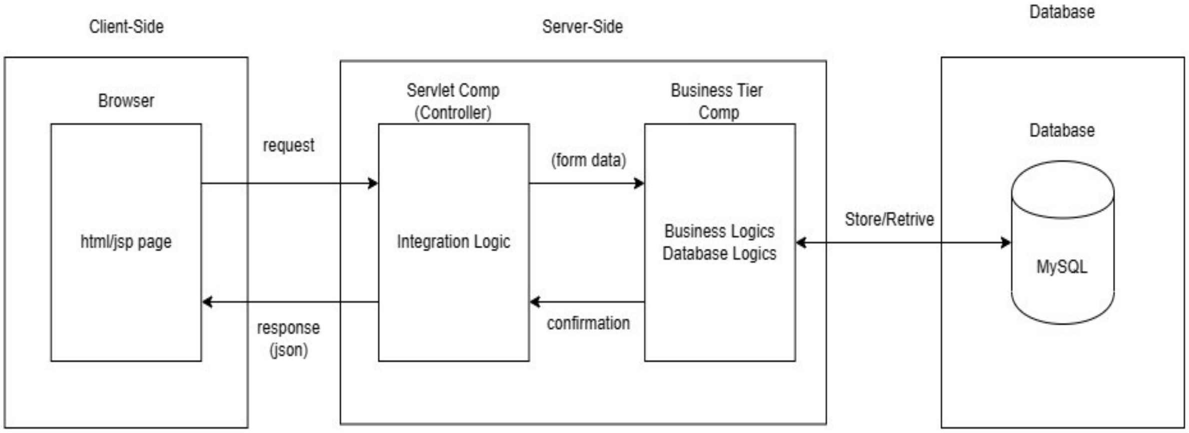


Figure 3.2.1Architectural Design of SPA

Here the architecture design of the SPA system is divided into three layers i.e., client-side, server-side and database. In the client side it consists of the user side where the jsp/html page requests the server side for the http methods to perform on the server side. The Server side is broken up into two more sub layers i.e., Servlet component and Business tier component where server component is given a role for integration logics and business layer and database logics are carried out by the business tier component. The database layer consists of the database to store the data and the requested data are retrieved via business tier component.

3.2.2 Database Schema design

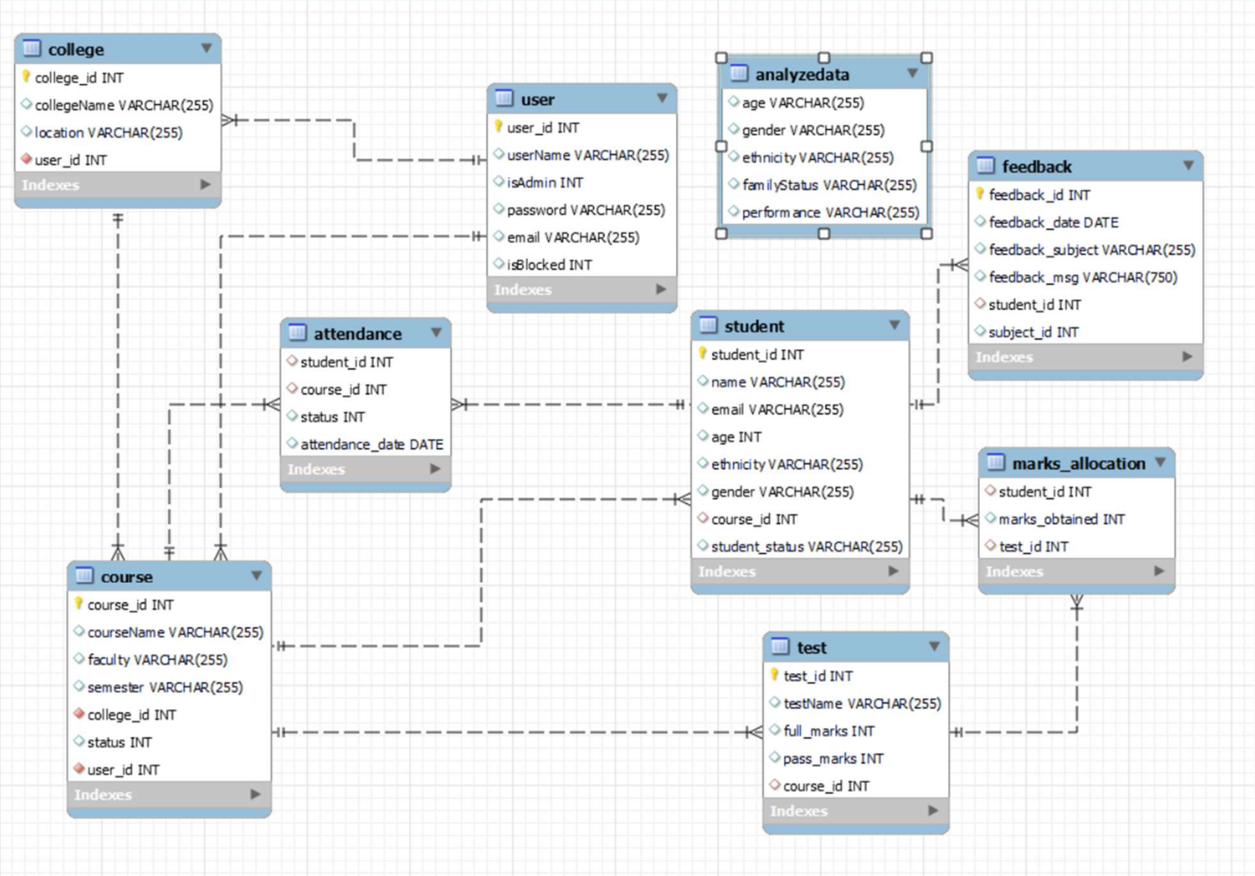
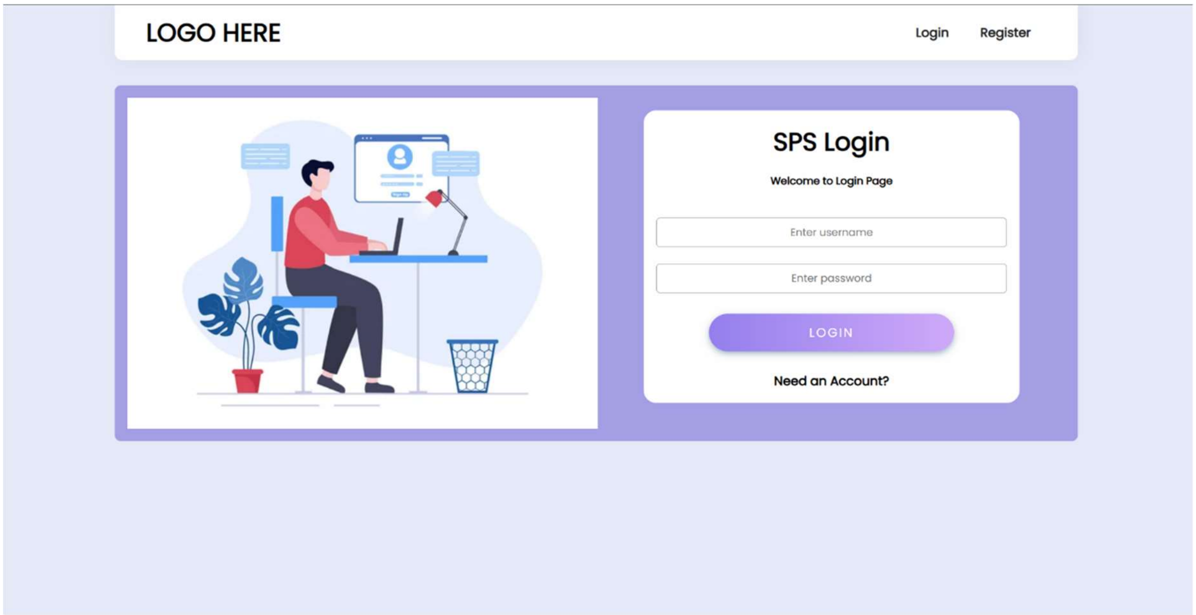


Figure 3.2.2 Database Schema design of SPA

Here the above figure presents the database schema design of the SPA system where the relation between tables are shown clearly with their respective primary keys and foreign key relating to the other table and creating a relationship between the tables.

3.2.3 Interface design (UI/UX)

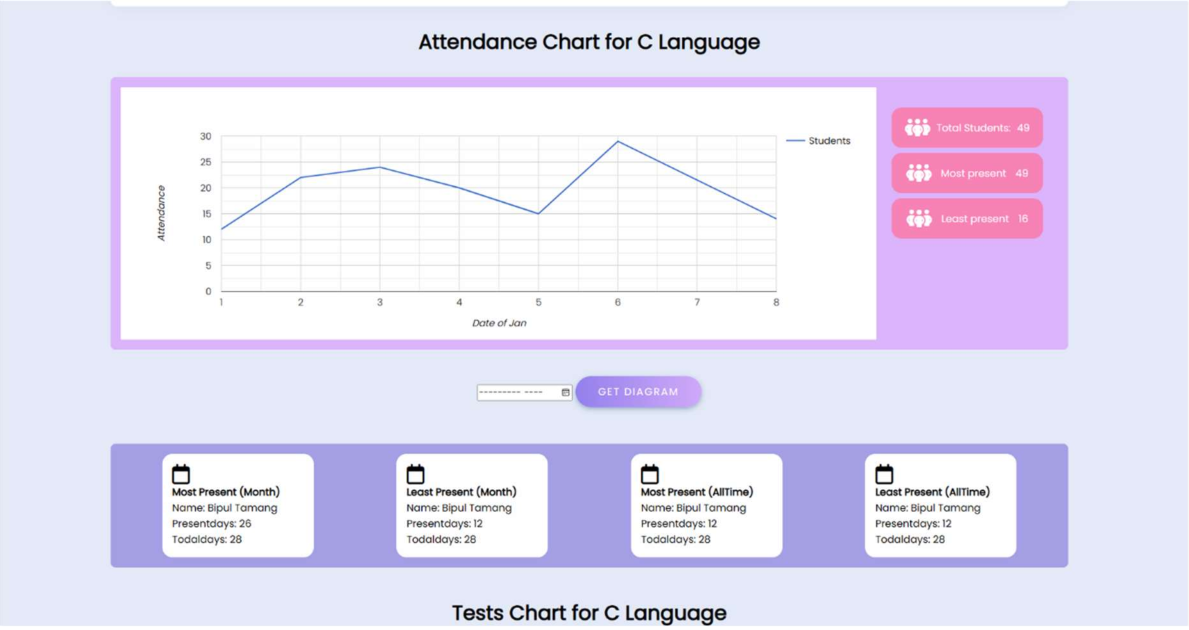


SPS Login

Welcome to Login Page

Need an Account?

Figure 3.2.3 UI design for login page


Attendance Chart for C Language

Most Present (Month)
Name: Bipul Tamang
Presentdays: 26
Todaldays: 28

Least Present (Month)
Name: Bipul Tamang
Presentdays: 12
Todaldays: 28

Most Present (AllTime)
Name: Bipul Tamang
Presentdays: 12
Todaldays: 28

Least Present (AllTime)
Name: Bipul Tamang
Presentdays: 12
Todaldays: 28

Tests Chart for C Language

Figure 3.2.4 UI design for course Summary

3.3 Algorithm

Decision Tree using Gini index

In the "Student Performance Analyzer", the decision tree algorithm is utilized to analyze student performance based on demographic factors such as age, gender, family status, ethnicity they graduated from. The decision tree algorithm employed in this report uses the Gini index as the impurity measure to construct the decision tree.

The decision tree algorithm with the Gini index is a popular and widely used approach for classification tasks in machine learning and data analysis. It aims to create a tree-like model of decisions based on the given dataset. The Gini index measures the impurity or disorder within a set of data, indicating how well a particular attribute separates the classes or categories.

The algorithm starts by selecting the best attribute to split the dataset at the root of the decision tree. This attribute is chosen based on its ability to minimize the Gini index, thereby maximizing the purity of the resulting subsets. The Gini index calculates the probability of misclassifying a randomly chosen element from the set if it were randomly labeled according to the class distribution.

Once the root attribute is selected, the dataset is split into subsets based on the attribute's values. The process is then recursively applied to each subset, selecting the best attribute at each level to split the data further. This recursive splitting continues until a stopping criterion is met, such as reaching a maximum tree depth or having pure leaf nodes where all instances belong to the same class. The decision tree constructed using the Gini index provides a hierarchical structure that represents the decision-making process based on the input features. Each internal node of the tree represents a decision based on a specific attribute, while the leaf nodes represent the final class labels or predictions. (Thakar, 2022)

The Gini index formula for decision tree splitting is calculated as follows:

Gini Index = 1 - Σ (p_i)^2

where:

- Gini Index: The Gini index of a particular attribute used for splitting.
- Σ: The summation symbol.
- p_i: The probability of an instance belonging to class i.

Chapter 4 Implementation and Testing

4.1 Implementation

4.1.1 Tools used

i. CASE TOOLS

For the designing and implementation of the system code the following CASE tools were used:

a) Eclipse IDE

Eclipse IDE was used to handle the java code Servlets and logics of the system application. The system was built with java, html, JavaScript, CSS which was all created and managed on Eclipse IDE. It provided a robust and feature rich environment for development of this web application

b) Tomcat Server

Tomcat server was chosen as the web server for hosting and deploying the Student Performance Analyzer web application. It provided a reliable and scalable platform for running Java-based web applications and handle web Servlets and JSP well.

c) Draw.io

Draw.io was utilized as a case tool for designing and creating visual representations, such as system flow diagrams, ER diagrams etc. during system analysis and design phase.

ii. Programming languages

For coding purpose, we used the following programming languages for our project:

a) Java

Java programming language was used for the backend development of Student Performance Analyzer web application. It provided the foundation for implementing core functionalities such as data processing, decision tree algorithm, web Servlets and JSP.

b) JavaScript

JavaScript was used for client-side scripting and interactivity in the Student Performance Analyzer web application. It was used to create dynamic content generation, user interface enhancements.

c) CSS

CSS was utilized for styling and visual presentation of the Student Performance Analyzer web application. It allowed for customization of fonts color, layouts and overall visual aesthetics.

iii. Database Platforms

a) MySQL

MySQL database platform was chosen as the backend database management system for the Student Performance Analyzer web application. It provided a reliable and scalable solution for storing, retrieving and managing the data.

4.1.2 Implementation details of modules

i. **User Authentication Module**

Procedures:

a) loginCheck (): Verifies the user's credentials against the stored user accounts in database and grant access to application upon successful authentication.

b) RegisterUser (): Handles the user registration process by creating a new user account and storing it in the database.

Functions/Classes/Methods:

a) LoginServlet/RegisterServlet: Contains logic for handling user input and calling the necessary function for login and register.

b) LoginLogic/RegisterLogic: Implements the data access layer for interacting with user account in database.

ii. **Data Entry Module**

Procedure

a) insertStudent (): Receives the student data from the user interface, performs validation checks and store the data in database.

Functions/Classes/Methods

a) MyStudentServlet: Contains logic for handling the input sent by the client and call necessary function to insert into database.

iii. **Reporting Module**

Procedure

a) getAttendanceChartData (): Retrieves the analyzed data from the database generates attendance reports on specified dates and presents to the user in a chart.

Functions/Classes/Methods

a) CourseAttendanceChartServlet: Handles the input of date and pass it to the function to get the attendance chart values and sends it as response to the client side.

iv. Database Management Module

Procedure

a) getConnection (): Establishes a connection to the MySQL database using JDBC for performing database operations.

Functions/Classes/Methods

a) EstablishConnection: Provides methods for establishing a database connection.

4.2 Testing

4.2.1 Test cases for Unit Testing

Table 4.2-1 Unit Testing

Test Case	Test data	Expected Outcome	Test result
Valid Data	Form data with all field containing valid values	Indicates data is valid	True
Invalid Data	Form data with one or more fields containing invalid values	Indicates data is invalid	True
Missing required fields	Form data with one or more required fields missing	Doesn't perform the submit action	True
Data Format Validation	Form data with fields requiring specific data formats (email)	True if data formats are correct	True

4.2.2 Test cases for System Testing

i. User Registration and Login

- Register a new user with valid credentials.
- Verify that user is successfully registered and stored in the database.
- Attempt to login using the registered credentials
- Verify that user is able to login and access the application
- Attempt to login with invalid credentials
- Verify that appropriate error messages are displayed and user is not granted access.

ii. Data Entry

- Enter valid student data in the data entry form.
- Save the data and verify that it is successfully stored in the database.
- Enter invalid or incomplete student data in the form.
- Verify that appropriate validation errors are displayed.
- Edit and update existing data and verify the changes in the database.

iii. Report Chart Generation

- Set attendance and test marks of the class students.
- Select specific month parameters to generate attendance chart report and select specific course details for test summary.
- Verify the graph data.

iv. System performance and Scalability

- Simulate concurrent user activity by accessing and interacting with application simultaneously.
- Measure the response time of the application under different load conditions.
- Verify that system handles the load and maintains acceptable performance levels without errors or crashes.

Chapter 5 Conclusion and Future Recommendation

5.1 Conclusion

The Student Performance Analyzer represents a significant advancement in student management and performance evaluation. Its integration of decision trees for performance analysis based on demographic values enhances its capability to provide actionable insights for educators and administrators. This system has the potential to improve student outcomes and contribute to the overall success of educational institutions.

Moreover, the benefits of the Student Performance Analyzer cascade across the educational ecosystem. Administrators are equipped with a potent tool for strategic planning, enabling resource optimization and curriculum refinement. Teachers, armed with real-time attendance and performance data, can tailor their teaching methodologies to address specific challenges faced by individual students or groups. Students, on the other hand, are presented with an equitable chance to excel, as interventions are tailored to their unique needs

5.2 Lesson learnt/Outcome

The development and implementation of the Student Performance Analyzer yielded several valuable lessons and outcomes, contributing to a deeper understanding of technology-driven educational solutions and their impact on student management and performance evaluation:

- Data Quality and Preprocessing:
  The quality of input data directly impacts the accuracy of decision tree analysis. Ensuring accurate and well-preprocessed data is crucial for generating meaningful insights.
- User-Centric Design:
  The user interface design significantly influences user adoption and satisfaction. A user-centric approach, involving educators and administrators in the design process, led to an intuitive and easy-to-navigate interface that streamlined tasks and reduced the learning curve for users.

- Continuous Improvement:
  Launching the Student Performance Analyzer was just the beginning. Regular feedback and ongoing assessments from educators and administrators proved invaluable in identifying areas for improvement and driving iterative enhancements.
- Long-Term Vision:
  The project highlighted the importance of aligning technological solutions with a long-term vision for education. While immediate benefits are essential, ensuring that the system evolves to meet future educational demands and embraces emerging technologies is crucial for sustained success.

The following are the outcome of the completed project

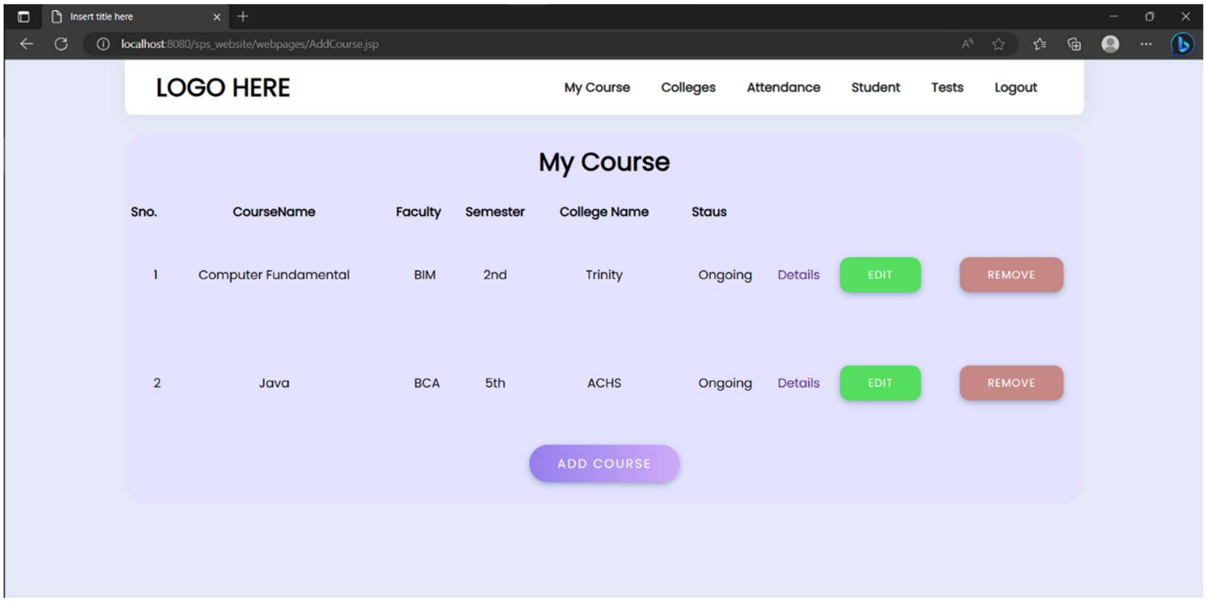
- The user is able to login and register.
- The user is able to add colleges and course and manage them.
- The user is able to add student on course and manage them.
- The user is able to add tests and assign marks and review the summary of test.
- The user is able to assign attendance for various days in a month and review the summary of attendance.
- The user is able to view the visual representation of tests and attendance of the student inside course details.
- The user is able to analyze the performance of certain demographic value student based on demographic values.
- The user is able to provide feedback to particular student and mail goes to the student.
- Student is able to view their summary via student portal.
- Admin is able to block users.

5.3 Future Recommendations

For the further future recommendations following things could be added or improved:

- Include subscription for adding more than particular number of courses.
- Select particular city than ethnicity for more improved data analysis.
- Extend compatibility to mobile platforms.

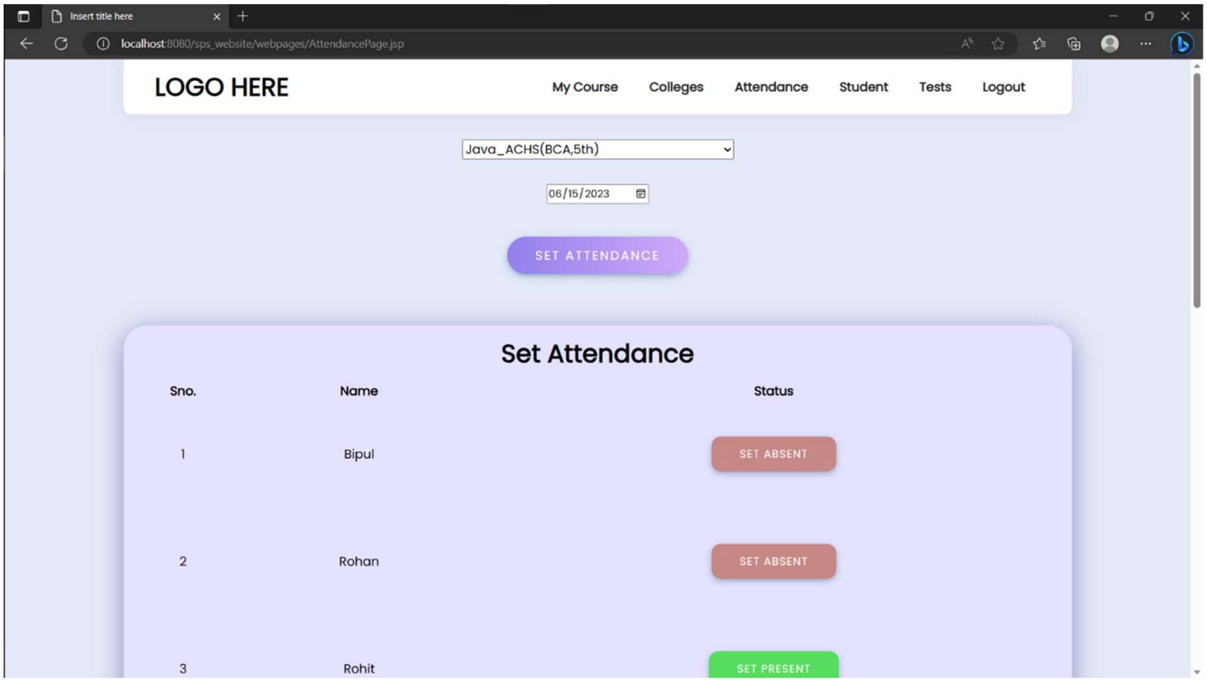
Appendices



My Course

Sno.	CourseName	Faculty	Semester	College Name	Staus			
1	Computer Fundamental	BIM	2nd	Trinity	Ongoing	Details	EDIT	REMOVE
2	Java	BCA	5th	ACHS	Ongoing	Details	EDIT	REMOVE

Figure 5.3.1Course management page of SPA



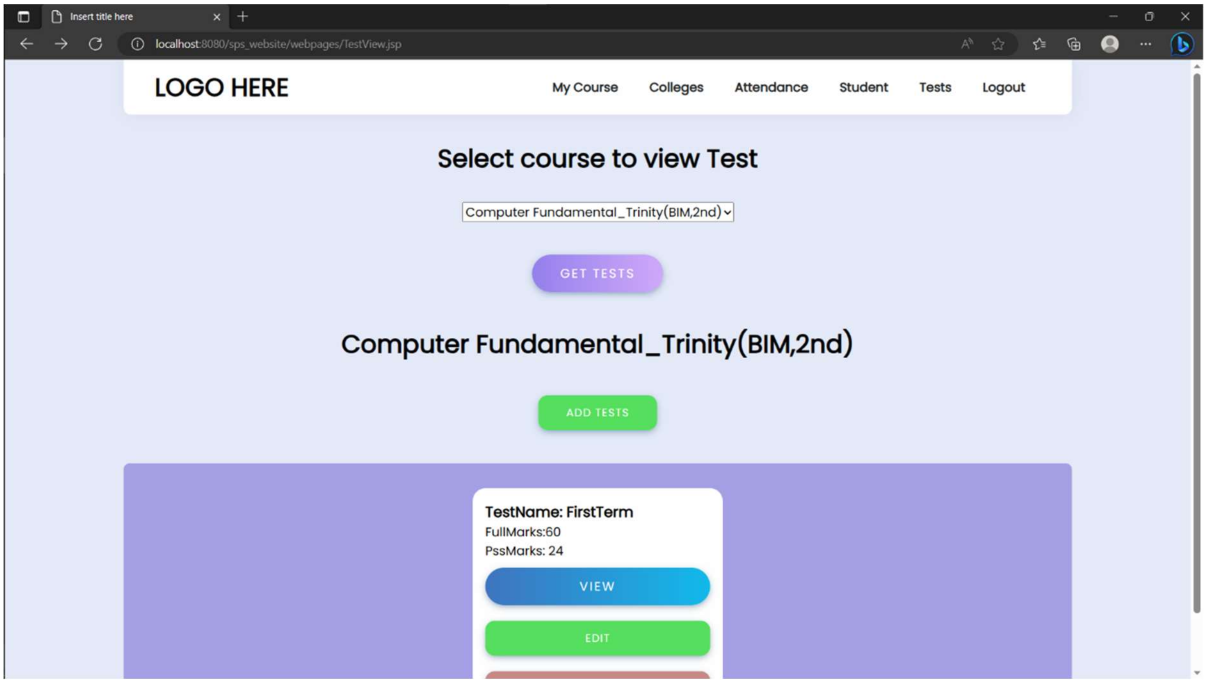
Java_ACHS(BCA,5th)

06/15/2023

Set Attendance

Sno.	Name	Status
1	Bipul	SET ABSENT
2	Rohan	SET ABSENT
3	Rohit	SET PRESENT

Figure 5.3.2 Attendance page of SPA



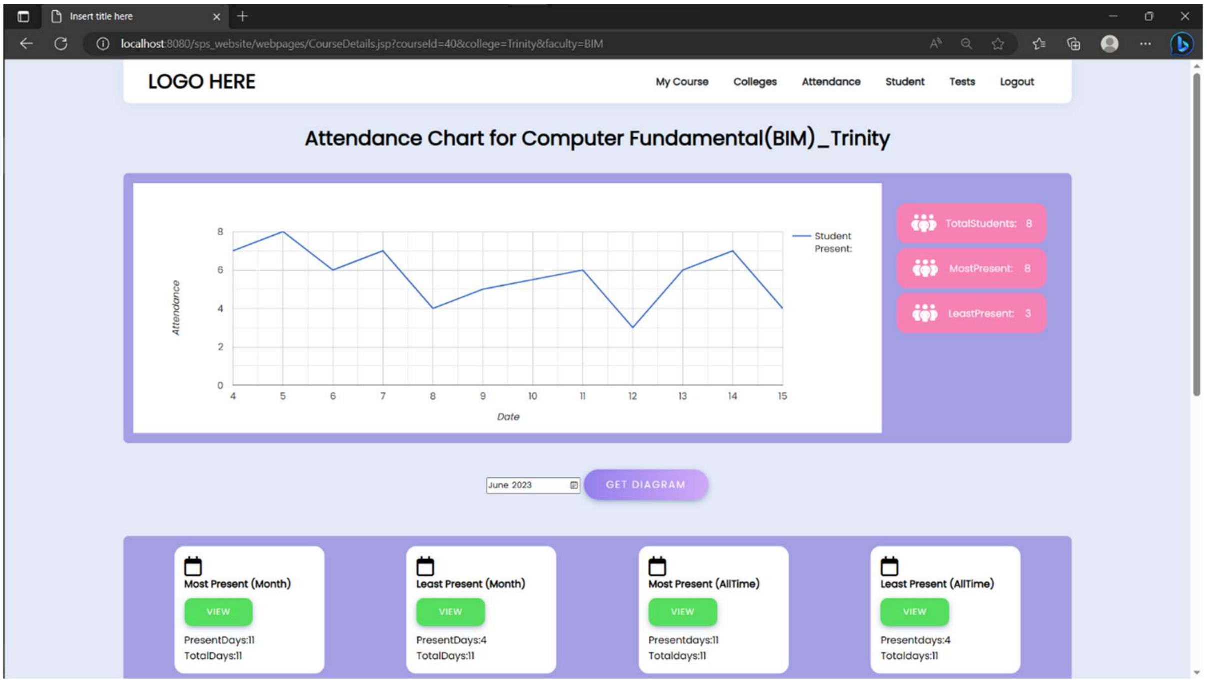
Select course to view Test

Computer Fundamental_Trinity(BIM,2nd)

Computer Fundamental_Trinity(BIM,2nd)

TestName: FirstTerm
FullMarks:60
PssMarks: 24

Figure 5.3.3 Test page of SPA



Attendance Chart for Computer Fundamental(BIM)_Trinity

TotalStudents: 8
MostPresent: 8
LeastPresent: 3

June 2023

Most Present (Month)
PresentDays:11
TotalDays:11

Least Present (Month)
PresentDays:4
TotalDays:11

Most Present (AllTime)
Presentdays:11
Totaldays:11

Least Present (AllTime)
Presentdays:4
Totaldays:11

Figure 5.3.4 Course detail page of SPA

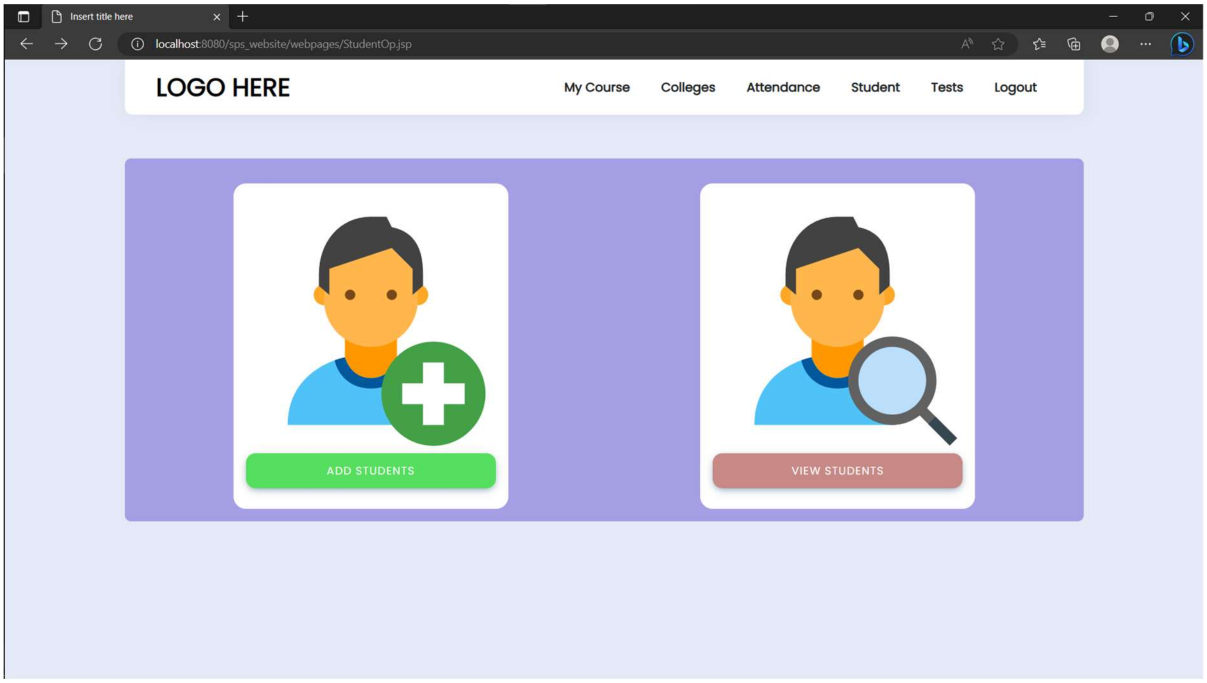


Figure 5.3.5 Student page of SPA

Bibliography

Holmes-Smith, P. (2013). Retrieved from sapplatform: https://www.spaplatform.com.au

Thakar, C. (2022, November 24). Retrieved from blog.quantinsti: https://blog.quantinsti.com/gini-index/