

AUTOMATED CGPA CALCULATOR

A MINI-PROJECT REPORT

Submitted by

G MADHU MITHA (221801030)

In partial fulfillment for the award of the degree of

**BACHELOR OF TECHNOLOGY IN
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**



RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2025

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled “**AUTOMATED CGPA CALCULATOR**” is the Bonafide work of **MADHU MITHA (221801030)** who carried out the work under my supervision.

SIGNATURE

Dr. J.M. Gnanasekar M.E., Ph.D.,
Professor and Head
Department of Artificial Intelligence
and Data Science
Rajalakshmi Engineering College
Chennai – 602 105

SIGNATURE

Mrs S. Renuka Devi M.E,
Associate Professor,
Department of Artificial Intelligence
and Data Science
Rajalakshmi Engineering College
Chennai – 602 105

Submitted for the project viva-voce examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Mr. S. MEGANATHAN, B.E, F.I.E., our Vice Chairman Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S., and our respected Chairperson Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D., for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to Dr. S.N. MURUGESAN, M.E., Ph.D., our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to Dr.. J.M. GNANASEKAR., M.E., Ph.D., Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. We are glad to express our sincere thanks and regards to our supervisor Mrs. D SORNA SHANTHI, M.Tech, Associate Professor, Department of Artificial Intelligence and Data Science and coordinator, S. RENUKA DEVI, M.E., Associate Professor, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for her valuable guidance throughout the course of the project.

Finally we express our thanks for all teaching, non-teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

ABSTRACT

The CGPA (Cumulative Grade Point Average) Calculator is a user-friendly web-based application designed to simplify and automate the process of calculating semester-wise GPA and overall CGPA for university students. This project aims to eliminate manual errors, reduce calculation time, and provide an efficient academic performance tracking solution. The application allows users to input subject names, assign corresponding credit hours, and select obtained grades through an intuitive interface. It then computes the GPA for each semester based on predefined grade-point mappings and accumulates the results to calculate the CGPA over multiple semesters. Developed using modern web technologies such as HTML, CSS, JavaScript, and optionally PHP for backend operations, the tool supports data storage and retrieval functions, ensuring users can add, edit, or download their academic records. The system is especially helpful for students in understanding their academic standing in real-time and for institutions that require reliable grade management tools. Additionally, the calculator can be enhanced with features such as authentication modules for personalized access, downloadable reports, mobile responsiveness, and integration with academic databases. The project not only addresses the challenges faced by students in managing academic grades but also promotes digital transformation in educational systems by offering a scalable and accessible GPA management platform.

TABLE OF CONTENTS

S.NO	TITLE	PAGE NO.
	ABSTRACT	IV
	LIST OF FIGURES	VII
1	INTRODUCTION	1
	1.1 GENERAL	1
	1.2 NEED FOR THE STUDY	2
	1.3 OBJECTIVES OF THE STUDY	3
	1.4 OVERVIEW OF THE PROJECT	4
2	REVIEW OF LITERATURE	5
	2.1 INTRODUCTION	5
	2.2 FRAMEWORK OF LCA	6
3	SYSTEM OVERVIEW	9
	3.1 EXISTING SYSTEM	9
	3.2 PROPOSED SYSTEM	12
	3.3 FEASIBILITY STUDY	14
4	SYSTEM REQUIREMENTS	18
	4.1 HARDWARE REQUIREMENTS	18
	4.2 SOFTWARE REQUIREMENTS	19

5	SYSTEM DESIGN	20
	5.1 SYSTEM ARCHITECTURE	20
	5.2 MODULE DESCRIPTION	21
	5.2.1 MODULE 1	21
	5.2.2 MODULE 2	21
	5.2.2 MODULE 3	22
	5.2.2 MODULE 4	23
6	RESULTS AND DISCUSSION	24
7	CONCLUSION AND FUTURE ENHANCEMENT	27
	7.1 CONCLUSION	27
	7.2 FUTURE ENHANCEMENT	28
	APPENDIX	23
	A1.1 SCREENSHOTS	29
	REFERENCES	31

LIST OF FIGURES

Fig No	Fig Name	Page No
2.2.1	Life Cycle Approach	6
5.1.1	System Architecture	20
5.2.1	Admin & Signup Module	21
5.2.2	Login Module	22
5.2.3	Student Login Module	23
5.2.4	GPA & CGPA Calculating Module	23
A1.1	Admin & Signup Page	29
A1.2	User Login Module	29
A1.3	Student Login Module	30
A1.4	GPA & CGPA Calculating Module	30

CHAPTER 1

INTRODUCTION

1.1 GENERAL

In today's academic environment, accurate and timely calculation of a student's academic performance is crucial. One of the most widely used methods to evaluate student performance in higher education is the **Cumulative Grade Point Average (CGPA)** system. CGPA is calculated based on the grades earned in various subjects, weighted by their respective credit values. Traditionally, students compute their CGPA manually, which involves complex calculations that are often prone to human errors, misinterpretation of grade-to-point conversions, and inconsistency in applying institutional grading standards. Such manual efforts not only lead to inaccuracies but also consume significant time and effort.

To address these challenges, this project proposes the development of an **automated, web-based CGPA Calculator**. The system is designed to simplify the process of calculating CGPA by allowing students to input subject names, corresponding grades, and credit points. Using a predefined grade-to-point conversion scale and a weighted average algorithm, the system computes the CGPA instantly and accurately. It eliminates the need for manual computation and ensures that the results adhere to academic standards.

Furthermore, the system incorporates a **secure authentication mechanism** to ensure that only authorized users can access and manage CGPA data. It features a **user-friendly interface** for smooth navigation and an **interactive grade entry module** that dynamically adjusts based on the number of subjects. To enhance the utility of the system, students are also provided with the option to **export their CGPA reports in PDF and Excel formats**, which can be used for academic or professional purposes.

By automating the CGPA calculation process, this project aims to **increase accuracy**, **improve efficiency**, and provide a **convenient platform** for both students and academic institutions to manage academic performance data.

1.2 NEED FOR THE STUDY

The need for this study arises from the growing reliance on accurate academic performance evaluation in modern education systems. The **Cumulative Grade Point Average (CGPA)** is a critical metric used to assess a student's overall academic achievement. However, **manual CGPA calculation** is time-consuming, prone to errors, and often confusing due to varying grading systems and credit structures across institutions. Mistakes in such calculations can negatively impact students' academic records, scholarship eligibility, and career opportunities.

Moreover, most existing CGPA calculators lack **security features, report export options, or user-friendly interfaces**. Students are forced to rely on inconsistent tools that do not retain data, cannot be customized, and do not provide downloadable reports for formal submission. This makes it difficult for both students and academic staff to maintain clear and accessible academic records.

There is a clear need for a system that not only **automates the CGPA computation process** but also provides **secure, reliable, and accessible tools** for academic tracking. By developing a **web-based CGPA Calculator**, this project addresses these issues by offering a platform that simplifies data input, ensures accurate calculations using standard formulas, and allows secure access with data export functionality.

This study is essential to support **digital transformation in education**, reduce the dependency on manual processes, and provide students with a dependable solution for tracking and managing their academic performance efficiently.

1.3 OBJECTIVES OF THE STUDY

The primary objective of this study is to design and develop a **web-based CGPA Calculator** that automates the calculation of the Cumulative Grade Point Average, ensuring accuracy, security, and ease of use. The system aims to replace manual CGPA computation methods with a reliable and efficient digital solution.

- **1.3.1 To Automate CGPA Calculation**

- Implement an algorithm that calculates CGPA using standardized grade-to-point conversion and weighted average formulas.
- Reduce human errors and save time compared to manual calculations.

- **1.3.2 To Ensure Secure Access and Data Privacy**

- Integrate a secure admin login system to restrict access to authorized users only.
- Protect student academic data from unauthorized access or modification.

- **1.3.3 To Provide a User-Friendly Interface**

- Design an intuitive and responsive interface for easy navigation and data input.
- Allow dynamic addition of subjects, grades, and credit values.

- **1.3.4 To Enable Data Export Options**

- Include features to export CGPA reports in both **Excel** and **PDF** formats.
- Help students generate formal records for academic or professional use.

- **1.3.5 To Support Flexibility and Compatibility**

- Develop a cross-platform compatible system accessible from desktops, tablets, and smartphones.
- Allow adaptation to different university grading scales for broader usability.

These objectives collectively aim to enhance the overall academic experience for students by providing a reliable and efficient tool for CGPA computation and record management.

1.4 OVERVIEW OF THE PROJECT

- **1.4 Overview of the Project**

This project focuses on the development of a **web-based CGPA Calculator** that simplifies and automates the process of calculating the **Cumulative Grade Point Average (CGPA)** for students. The system is designed to address the limitations of manual CGPA calculation methods, which are often error-prone, time-consuming, and lack data security. By providing a structured and interactive platform, the CGPA Calculator enables users to enter subject-wise grades and corresponding credit values, which are then processed using a standardized **grade-to-point conversion** and **weighted average formula** to generate an accurate CGPA result.

The application includes a **secure login system** to restrict unauthorized access and protect user data. A major feature of the system is its **dynamic input form**, which allows users to add any number of subjects and associated information without constraints. Once the data is entered, the system instantly calculates the CGPA and presents the result in a clear and concise manner.

Additionally, the system provides functionality to **export CGPA reports in Excel and PDF formats**, allowing students to save or submit their academic performance records with ease. The user interface is built to be **simple, responsive, and accessible** across different devices, ensuring a smooth user experience for all types of users. Overall, this project delivers a **smart, accurate, and secure solution** for CGPA calculation and academic record management, addressing both functional and usability needs of students and academic institutions alike.

CHAPTER 2

REVIEW OF LITERATURE

2.1 INTRODUCTION

The review of literature is a critical part of any research or development project, as it provides a foundation for understanding existing work and identifying the scope for innovation. In the context of this project, which focuses on the design and development of a web-based CGPA (Cumulative Grade Point Average) Calculator, it is essential to study various methods, tools, and systems that have been previously implemented for academic performance evaluation.

Over the years, numerous academic institutions and independent developers have created CGPA calculators using different technologies and approaches. These include desktop-based applications, mobile apps, and web-based tools, each offering varying levels of functionality. While some tools focus primarily on automating CGPA calculations, others incorporate features such as GPA prediction, visual analytics, and institutional grading variations. However, many of these systems lack certain critical features such as data privacy, secure login mechanisms, customization for different grading scales, and the ability to export reports in useful formats like Excel or PDF.

In addition to technical systems, several scholarly papers and articles have discussed the theoretical background of CGPA computation, its role in academic planning, and comparisons between CGPA and other assessment methods. These studies highlight the importance of accurate performance measurement and the challenges students face in managing academic results manually.

By analysing the literature, this chapter aims to:

- Understand the evolution of CGPA computation tools.
- Identify common features and limitations in existing systems.
- Highlight the research gap and justify the need for the proposed web-based CGPA Calculator.

2.2 FRAMEWORK OF LCA

LIFE CYCLE APPROACH

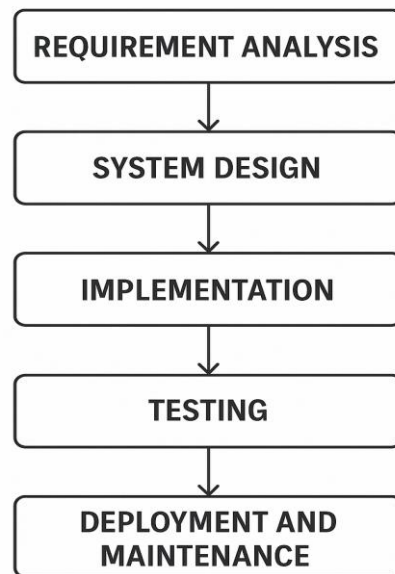


Fig 1: LCA

For a **Life Cycle Assessment (LCA) Framework** of a **Student CGPA Calculator Project**, we can break it down into stages from concept and planning to deployment and maintenance. LCA in software development typically refers to the evaluation of the project's full life cycle, ensuring efficient development, operation, and sustainability of the system.

- **1. Planning and Requirement Analysis**
- **Objective:** Understand the core requirements of the system: CGPA calculation, GPA viewing, marks downloading, and automation.
- **Target Users:** Students, Administrators, and Faculty.
- **Features:**
 - Students can input their marks for various subjects.
 - Automated CGPA calculator based on predefined grading criteria.
 - Students can download their marks and CGPA.

- Faculty/Admin can update grading criteria or manage students' data.
- **2. System Design**
- **Architecture:**
 - **Frontend:** HTML, CSS, JavaScript (for UI/UX), and dynamic features with jQuery.
 - **Backend:** Flask/Django (Python) or Node.js for handling user requests, calculating CGPA, and data processing.
 - **Database:** MySQL/PostgreSQL for storing student details and marks.
 - **Downloadable Report:** PDF or CSV format for marks.
- **Modules:**
 - **Login and Registration:** For student and admin access.
 - **Marks Entry:** Students can input marks per subject or upload CSV files.
 - **CGPA Calculation:** Formula-based GPA and CGPA calculation.
 - **Downloadable Marks:** Generate downloadable CSV or PDF report.
 - **Admin Panel:** For managing students and updating criteria.
- **3. Development**
- **Frontend:**
 - **Signup/Login Pages:** Use HTML forms and CSS for styling.
 - **Marks Input Page:** Form to input or upload marks.
 - **Result Display Page:** Display GPA, CGPA, and the download button.
- **Backend:**
 - Implement routes and logic to calculate GPA and CGPA.
 - Design the API for generating downloadable marks reports.
- **Database:**
 - Set up tables for storing student details, marks, and grades.
 - Use a relational database to link student records and marks.
- **4. Testing**
- **Unit Testing:** Test individual modules, especially the CGPA calculation logic and data entry forms.

- **Integration Testing:** Ensure that all system components (frontend, backend, database) work together seamlessly.
- **User Acceptance Testing (UAT):** Have a small group of students test the system to ensure it meets requirements (CGPA calculation, marks download, etc.).
- **5. Deployment**
- **Cloud/Server:** Deploy the application on cloud services like Heroku, AWS, or a local server.
- **Database Setup:** Ensure the database is hosted and integrated properly with the web application.
- **Security:** Implement basic security practices such as user authentication, data encryption, etc.
- **6. Maintenance**
- **Monitoring:** Track usage and system performance (e.g., GPA calculations, server performance).
- **Bug Fixing:** Continuously address issues reported by users.
- **Updates:** Periodically update the grading system or any other parameters.
- **7. Evaluation (Post-deployment)**
- **Performance Metrics:** Analyze system response time, user satisfaction, and CGPA calculation accuracy.
- **User Feedback:** Collect feedback from students and faculty to improve the system.
- **Downloadable Report:**
 - Python (Flask) with ReportLab or FPDF for PDF generation
 - CSV handling with Python (csv module) or JavaScript (FileSaver.js)
- **Security:**
 - JWT or session-based authentication for login
 - HTTPS for secure communication

This LCA framework provides an end-to-end solution for a Student CGPA Calculator system, from conceptualization and development to deployment and maintenance. Let me know if you need any further details or assistance!

CHAPTER 3

SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

Various CGPA calculator systems are already available online, which assist students in automatically calculating their Grade Point Average (GPA) and Cumulative Grade Point Average (CGPA) based on their individual subject grades. These systems have been developed with the aim of providing a user-friendly interface, reducing manual errors in grade calculation, and offering an efficient method for students to track their academic performance.

Most of these systems offer basic functionality such as grade input, GPA/CGPA calculation, and report generation. However, they often lack features such as data persistence, customized grading systems, and scalability for large institutions. Below are a few popular existing systems that serve as the foundation and inspiration for the CGPA Calculator project:

Examples of Existing CGPA Calculator Systems

- **1. Tutorialspoint CGPA Calculator**

Tutorialspoint provides a widely used online **CGPA Calculator** tool. It allows students to enter their grades for multiple subjects and instantly calculates both GPA and CGPA. The system supports the **10-point grading scale**, which is common in many institutions, and outputs the GPA for each subject along with the overall CGPA.

- **Features:**

- Input marks for multiple subjects.
- Automatically calculates GPA for each subject.
- Provides an overall CGPA based on the grades entered.
- Allows students to download the calculated report in **CSV format**.

- Simple user interface with no login required.

This tool is ideal for students who want a quick and efficient way to calculate their CGPA, and the **CSV report generation** feature is highly useful for keeping track of academic progress.

- **Website:** [TutorialsPoint CGPA Calculator](#)

• 2. Pega CGPA Calculator

Pega offers another online tool for GPA/CGPA calculation, providing more flexibility in terms of the grading system. It supports multiple grading scales, including both the **4-point scale** and the **10-point scale**, allowing students from different educational backgrounds to use the tool. Pega's calculator also generates GPA and CGPA, but it focuses more on institutional-level customization.

• Features:

- Supports multiple grading systems (e.g., **4-point scale**, **10-point scale**).
- Real-time GPA/CGPA calculation.
- Option to **download reports** in both PDF and CSV formats.
- Simple and responsive user interface.

Pega is widely used in educational institutions due to its flexibility and ability to accommodate different grading systems. It is particularly useful for students who are following specialized or custom grading systems used by their institution.

- **Website:** [Pega CGPA Calculator](#)

• 3. College/University-specific CGPA Calculators

Several universities and colleges have developed their own **CGPA calculator tools** as part of their online student portals. These calculators are tailored to specific institutional grading systems and provide a seamless user experience for students enrolled in the respective institutions. They typically allow students to:

- Enter marks for various courses.
- Choose the **semester** or **term** for which they want to calculate GPA.
- Receive immediate feedback on their grades and overall CGPA.

- Download and print detailed **academic reports** for personal records or academic advising purposes.

For instance, universities like **MIT**, **Harvard**, and **IITs** often have internal tools that integrate with their grading systems and offer advanced features like **academic progress tracking**, **class ranking**, and **automatic updates** based on course completion.

- **4. Mobile Applications**

In addition to web-based tools, there are **mobile applications** that offer GPA/CGPA calculation features. These apps are designed for students who prefer to manage their academic progress on-the-go. Popular apps like **MyCGPA** (for Android) allow students to:

- Enter their grades for each subject and automatically calculate their GPA and CGPA.
- View their grade history and track their academic progress over time.
- Get push notifications about upcoming grades or semester results.
- Download or share their GPA reports via email or social media.

While these mobile apps provide portability and convenience, they often lack the deeper customization and institutional-level integration found in web-based systems used by universities.

3.2 PROPOSED SYSTEM

The **Proposed CGPA Calculator System** aims to automate the process of calculating the **Cumulative Grade Point Average (CGPA)** for students based on their marks and grades. This system will be an enhancement over existing tools by addressing limitations such as customization, data persistence, and scalability. It will be developed as a web-based application with the following features:

- **Customizable Grading Systems:** The system will support multiple grading scales (e.g., 10-point, 4-point, and others) and can be easily customized to match the grading scale of different educational institutions.
- **Student Authentication:** Users will log in with secure credentials to access their personal academic data, ensuring data privacy and security.
- **Data Persistence:** The system will allow students to store their grades over multiple semesters, enabling them to track their academic performance over time.
- **Automated Report Generation:** The system will automatically generate detailed reports, including GPA for each subject and overall CGPA, which can be downloaded in **CSV** or **PDF** formats for personal records.
- **Admin Panel:** An admin interface will allow faculty members or academic administrators to update grading criteria, view student records, and manage user access.

• 3.2.1 Features of the Proposed System

1. User Interface:

- Simple, clean, and intuitive interface for students to input marks for each subject.
- Dropdowns for selecting subjects and grades.
- Real-time GPA/CGPA calculation with immediate feedback.
- Downloadable report generation in multiple formats (CSV, PDF).

2. Backend and Database:

- A **MySQL** or **SQLite** database to store student marks, grades, and calculated GPA/CGPA for data persistence.
- **Flask** or **Django** as the web framework to handle the server-side logic, including authentication, data processing, and report generation.

3. Security:

- Secure login and authentication for students using modern practices (e.g., JWT tokens or session-based authentication).
- Admin access control to restrict access to sensitive data, ensuring only authorized personnel can manage student records.

4. Customization:

- The ability to add or modify grading scales, which can be tailored to different educational institutions' requirements.
- Support for additional features like tracking academic history over multiple semesters or providing insights into GPA trends.

• 3.2.3 Advantages of the Proposed System

- **Efficiency:** By automating the GPA/CGPA calculation and report generation process, the system saves time and eliminates human errors.
- **Scalability:** The system can be scaled to accommodate a large number of users, making it suitable for use in both small colleges and large universities.
- **Flexibility:** It offers the flexibility to support different grading systems, making it adaptable to the needs of various institutions.
- **Data Management:** Students can access historical data about their academic performance, enabling them to track their progress throughout their academic journey.

3.3 FEASIBILITY STUDY

3.3.1 Introduction

The **Feasibility Study** evaluates the practicality and viability of the **Proposed CGPA Calculator System** from three key perspectives: **Technical Feasibility**, **Economic Feasibility**, and **Operational Feasibility**. This study aims to ensure that the proposed system can be successfully developed and implemented without significant obstacles.

3.3.2 Technical Feasibility

The technical feasibility of the CGPA Calculator System is assessed based on the available technology, the skills required for implementation, and the development tools.

1. Technology Stack:

- **Frontend:** The frontend of the system will be built using **HTML5**, **CSS3**, and **JavaScript** for a responsive and user-friendly interface. The use of **Bootstrap** will ensure a mobile-friendly layout.
- **Backend:** The backend will be developed using **Python** with the **Flask** or **Django** framework. These frameworks are well-suited for building web applications and are supported by large communities for troubleshooting and support.
- **Database:** **MySQL** or **SQLite** will be used to store student marks, grades, and CGPA records. These are reliable and widely used databases with extensive support.
- **Authentication:** The system will use **JWT (JSON Web Tokens)** or **session-based authentication** to securely log in students and manage sessions.
- **Report Generation:** Python libraries like **pandas** and **FPDF** will be used to generate downloadable CSV and PDF reports.

2. Development Environment:

- The system will be developed using **Visual Studio Code** as the Integrated Development Environment (IDE), which provides excellent support for Python, HTML, CSS, and JavaScript.

- The system will be hosted on a local or cloud server for testing and eventual deployment, such as **Heroku** or **AWS** for production environments.

3. Required Skills:

- **Frontend Development:** Knowledge of HTML, CSS, and JavaScript for building the user interface.
- **Backend Development:** Proficiency in Python with Flask or Django for server-side logic and database integration.
- **Database Management:** Familiarity with **MySQL** or **SQLite** for setting up and managing the database.
- **Report Generation:** Experience in working with libraries like **pandas** and **FPDF** for generating reports in various formats.

3.3.3 Economic Feasibility

Economic feasibility examines the cost of developing, maintaining, and implementing the system to determine if it is affordable within the available budget.

1. Development Costs:

- The development cost primarily involves software and hardware requirements. Since the system is built using **open-source technologies** (Flask, MySQL, etc.), the cost of development is significantly reduced.
- The project can be completed using existing infrastructure (e.g., laptops, desktops) and open-source tools, eliminating the need for expensive proprietary software.

2. Operational Costs:

- The operational cost includes the cost of **hosting** the web application (e.g., using cloud services like **Heroku** or **AWS**).

- Database storage and server maintenance can incur minimal costs, depending on the number of users and frequency of access. This can be scaled as needed.

3. **Return on Investment (ROI):**

- For educational institutions, the return on investment comes in the form of **time-saving**, improved **accuracy** in grade calculations, and **streamlined report generation**.
- The ability to track and manage student performance more efficiently can result in better academic advising and data-driven decision-making.

3.3.4 **Operational Feasibility**

Operational feasibility assesses whether the system can be successfully implemented and used in real-world environments, considering user acceptance and institutional support.

1. **User Acceptance:**

- Students and faculty will find the system easy to use, thanks to the intuitive interface, automatic GPA/CGPA calculations, and the ability to generate downloadable reports. Training for users will be minimal, as the system is designed to be user-friendly.
- Administrative staff will appreciate the ability to manage student records efficiently and securely through the admin panel.

2. **Institutional Support:**

- The system requires minimal hardware resources and can be easily integrated into the existing IT infrastructure of educational institutions.
- The backend system can be hosted on an internal server or a cloud platform, ensuring accessibility for students and faculty alike.

- The proposed system supports multiple user roles (students, faculty, administrators), ensuring that the institution can manage data securely.

3. Scalability:

- The system is scalable and can accommodate a growing number of students without performance degradation. As the number of students increases, the system's database can be expanded, and additional features (e.g., advanced analytics, automated notifications) can be added.

3.3.5 Conclusion of Feasibility Study

Based on the technical, economic, and operational assessments, the **CGPA Calculator System** is deemed **feasible** for development and implementation. The system's design is supported by accessible and affordable technologies, and it provides a high return on investment through increased efficiency and improved student experience. The project is expected to be both **technically and economically viable** and can be smoothly integrated into educational institutions of various sizes.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

1. 4.1 Hardware Requirements

The hardware requirements for the proposed CGPA Calculator system are minimal and can be easily met by most modern computing devices. Below are the recommended hardware specifications:

2. Processor:

- Minimum: **Intel Core i3** or equivalent.
- Recommended: **Intel Core i5** or equivalent for better performance.

3. RAM:

- Minimum: **4 GB RAM**.
- Recommended: **8 GB RAM** for smooth performance, especially when handling large datasets.

4. Storage:

- Minimum: **500 MB of free disk space** for installation and data storage.
- Recommended: **1 GB or more** to accommodate larger databases and system logs.

5. Display:

- Minimum: **1024x768 resolution** for a clear user interface.
- Recommended: **1280x1024 resolution** for better user experience.

6. Network:

- **Internet connection** for web hosting, updates, and report generation if hosted on a cloud platform.

4.2 Software Requirements

1. Operating System:

- **Windows 7** or higher.
- **macOS 10.12** or higher.
- **Linux** distributions (Ubuntu, CentOS, etc.).

2. **Web Browser:**

- **Google Chrome, Mozilla Firefox, Safari**, or any modern web browser with **JavaScript** enabled.

3. **Development Software:**

- **Visual Studio Code** (or any code editor like Sublime Text, Atom).
- **Git** for version control.

4. **Programming Languages:**

- **Python 3.x** for backend development.
- **HTML5, CSS3, and JavaScript** for frontend development.

5. **Frameworks & Libraries:**

- **Flask** or **Django** (Python web frameworks) for backend development.
- **Bootstrap** for frontend design to make the application responsive.
- **pandas** for handling data and report generation.
- **FPDF** for generating downloadable PDF reports.
- **MySQL** or **SQLite** for database management.

6. **Database Management System (DBMS):**

- **MySQL** or **SQLite** (lightweight and suitable for smaller applications).

7. **Authentication Libraries:**

- **Flask-Login** or **Django Allauth** for handling user authentication and sessions.

CHAPTER 5

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

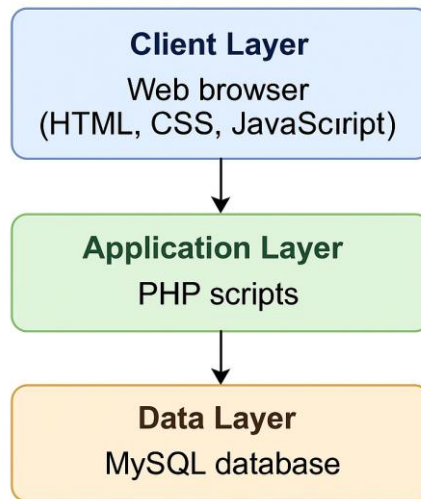


Fig 5.1.1 System Architecture

The System Architecture of the Doctor Appointment and Hospital Management System follows a three-tier design comprising the Client Layer, Application Layer, and Data Layer. The Client Layer consists of the user interface, built using HTML, CSS, and JavaScript, which allows users to interact with the system via a web browser. The Application Layer is responsible for processing business logic and handling user requests, managing sessions, and interacting with the database. Finally, the Data Layer uses MySQL to store and manage all system data, such as student records, teacher profiles, and CGPA calculator, ensuring fast and reliable data retrieval. This three-tier architecture ensures a well-organized, scalable, and maintainable system where each layer has a distinct responsibility, providing a seamless user experience and robust backend operations.

5.2 MODULE DESCRIPTION

Module 1: Admin & Signup Module

The Admin Module serves as the control center of the entire system. Administrators have comprehensive access to user management, including the ability to add, update, or remove both Faculty and students profiles. This module also allows the admin to review and monitor system activity, generate GPA and CGPA calculating, manage viewing and Downloading. The administrator ensures that the platform functions smoothly and that data remains consistent and secure across all modules.

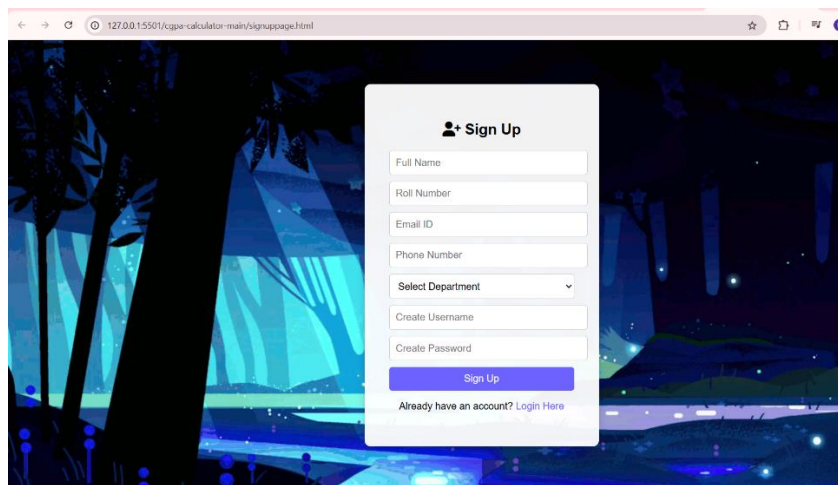


Fig 5.2.1 Admin & Signup Module

Module 2: Login Module

The **Login Module** of the CGPA Calculator system serves as the secure entry point for students to access their academic dashboard, where they can view marks, calculate GPA/CGPA, and download reports. It features a clean and user-friendly interface with input fields for username and password, ensuring only registered users can log in. If the user does not have an account, a “Register Here” link is provided for easy onboarding. The module performs basic input validation and integrates with a backend server (like Flask or PHP) to authenticate credentials against stored user data. On successful login, users are granted access to their personalized dashboard, making the system both secure and accessible.

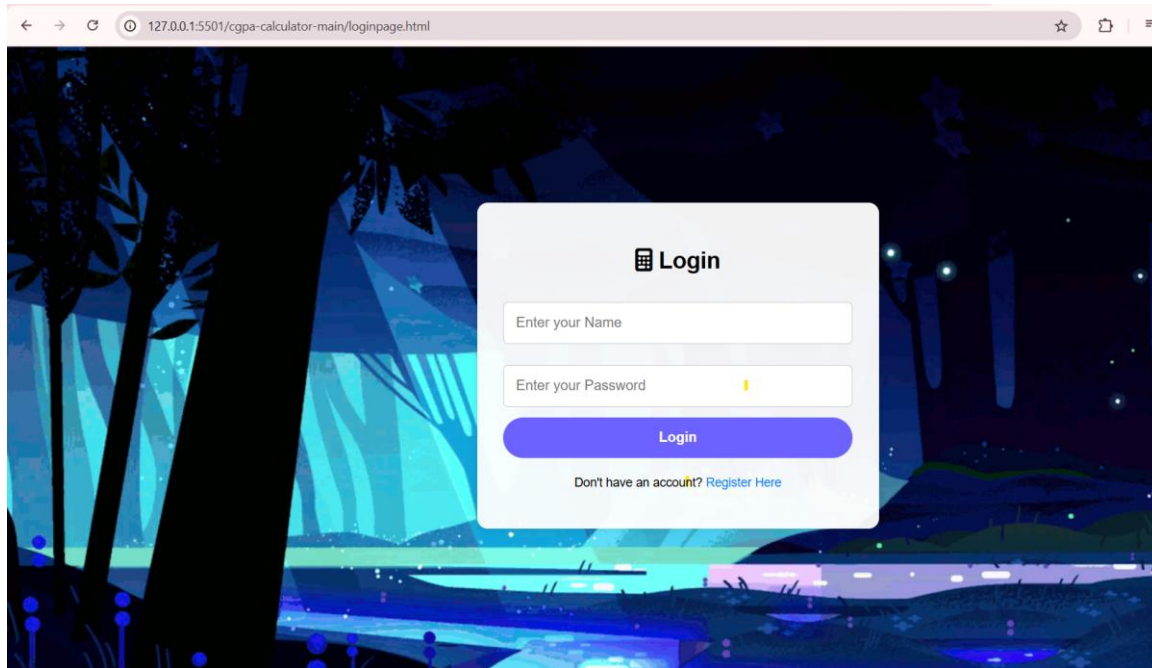


Fig 5.2.2 Login Module

Module 3: Student login Module

The Student Login Module of the CGPA Calculator system acts as a personalized access point for students to securely enter the platform using their credentials and view academic information tailored to them. Upon successful login, students are greeted with a customized welcome message and presented with a clean, interactive dashboard that provides essential tools such as CGPA calculation, subject-wise grade insights, and downloadable reports. This module ensures a smooth user experience while protecting student data, enabling learners to monitor their academic progress and understand their strengths and weaknesses effectively.

welcome madhumi!

The grading system will help the students understand their points of strength and weaknesses and address them accordingly. The uniformity of grading allows the universities to judge the results of the applicants in a more efficient way. It ensures the overall evaluation of the students on the basis of their year round academic performance.

[Calculate CGPA](#)

CGPA CALCULATOR

The CGPA (Cumulative Grade Point Average) calculator helps students calculate their overall academic performance.

SECTION

[What is CGPA?](#)
[Marks and Letter Grades](#)
[How to Calculate CGPA?](#)

© 2025 CGPA CALC. All rights reserved.

Fig 5.2.3 Student login Module

Module 4: GPA & CGPA Calculating Module

The **GPA & CGPA Calculating Module** allows students to input their grades and corresponding credit hours for each subject in a semester, enabling the system to compute both the GPA for the current semester and the overall CGPA across all semesters. The module offers a clear and organized table format where students can select grades from dropdowns and view the credit allocation per subject. Once the data is entered, the system automatically calculates the results and displays them at the bottom of the page. Additionally, features like "Add Semester" and "Download" allow users to expand their academic records and save their results for future reference, making academic tracking seamless and efficient.

CGPA Calculator

Semester 1

Subject	Grade	Credit Hours
Communicative English	<input type="text" value="O"/>	2
Matrices, Differential and Integral Calculus	<input type="text" value="O"/>	4
Engineering Chemistry	<input type="text" value="O"/>	3
Programming for Problem Solving in C	<input type="text" value="O"/>	3
Engineering Graphics	<input type="text" value="O"/>	4
Chemistry Laboratory	<input type="text" value="O"/>	1
C Programming Laboratory	<input type="text" value="O"/>	2
Communication English Laboratory	<input type="text" value="O"/>	1

GPA for Current Semester: N/A
Overall CGPA: N/A

[Add Semester](#)
[Download](#)

Fig 5.2.4 GPA & CGPA calculating Module

CHAPTER 6

SAMPLE CODE SNIPPETS

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="cgpa.css">
  <title>CGPA Calculator</title>

</head>

<body>
  <script
src="https://cdnjs.cloudflare.com/ajax/libs/html2canvas/1.4.1/html2canvas.min.js"></
script>
  <div class="header">
    <h1>CGPA Calculator</h1>
  </div>
  <div id='cgpa'>
    <div id="semesters"></div>
    <div id="overall-gpa-display">
      GPA for Current Semester: N/A
    </div>

    <!-- Display overall CGPA -->
    <div id="overall-cgpa-display">
      Overall CGPA: N/A
    </div>
```

```

</div>
<button onclick="addSemester()">Add Semester</button>
<button id="screenshot-button">Download</button>

<footer>

</div>
<div class="footer-bottom ">
    <p>&copy; 2025 CGPA CALZ. All rights reserved.</p>
</div>
</footer>

<script src="mainpage.js">
</script>
</body>

</html>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Login - CGPA Calculator</title>

    <!-- Font Awesome for icons -->
    <link    rel="stylesheet"    href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.5.2/css/all.min.css" />

    <!-- Link to custom CSS -->

```



```

<link rel="stylesheet" href="loginpage.css" />
</head>

<body>
  <div class="login-container">
    <form id="loginform" action="landing.html" method="GET" class="login-form">
      <h2><i class="fa-solid fa-calculator"></i> Login</h2>

      <input type="text" placeholder="Enter your Name" name="name" required />
      <input type="password" placeholder="Enter your Password" name="password"
required />

      <button type="submit">Login</button>

      <p class="signup-link">
        Don't have an account? <a href="signup.html">Register Here</a>
      </p>
    </form>
  </div>

  <script src="login.js"></script>
</body>

</html>

```

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1)CONCLUSION

The CGPA Calculator web application is a valuable academic tool designed to assist students in accurately calculating their Grade Point Average (GPA) for individual semesters and their overall Cumulative Grade Point Average (CGPA) throughout their academic journey. By providing a user-friendly interface with intuitive navigation, dropdown grade selectors, and predefined credit hours, the system simplifies the once complex process of grade evaluation. It empowers students to gain clear insights into their academic performance, helping them identify strengths and areas for improvement. The inclusion of features like dynamic semester addition, downloadable reports, and categorized guidance (such as "What is CGPA?" and "How to Calculate CGPA?") further adds educational value and self-sufficiency for users. Moreover, the module ensures uniformity in grade interpretation, aligning with standard academic evaluation systems across institutions. Overall, this system promotes academic transparency, reduces manual errors, and enhances student awareness and motivation towards academic excellence.

7.2)FUTURE ENHANCEMENT

- **1. User Authentication and Role Management**

Future versions of the CGPA Calculator can include secure user login and authentication mechanisms, enabling individual students to log in with credentials and access personalized academic data. Implementing role-based access control will allow different user types—such as students, faculty, and administrators—to have customized views and access levels, ensuring both privacy and data integrity.

- **2. Database Integration for Data Persistence**

Currently, most CGPA calculators operate on temporary sessions. Integrating a backend database like MySQL or MongoDB can enable persistent storage of student records, course details, grades, and calculated GPAs. This feature would allow students to revisit past academic performance without re-entering data and enable administrators to manage student records more effectively.

- **3. Graphical Performance Analysis**

Adding visual representations like bar charts or line graphs to track GPA and CGPA progress across semesters will provide users with an insightful overview of their academic trends. These analytics can help students identify performance patterns, academic strengths, and areas needing improvement in a more intuitive manner..

- **4. Support for Multiple Grading Systems**

To cater to a broader range of users from different universities and institutions, the system can be enhanced to support multiple grading schemes and credit systems. Users should be able to choose or customize their grading scale (such as A-F, 10-point, or percentage-based) and set corresponding credit weights to reflect their specific academic system accurately.

- **5. Admin Upload and Grade Automation**

A future enhancement could allow faculty or admin users to upload official grade sheets for students, automatically updating each student's GPA and CGPA without requiring manual input. This feature would reduce the burden on students and ensure data accuracy across the system.

APPENDIX

A1.1 SCREENSHOTS

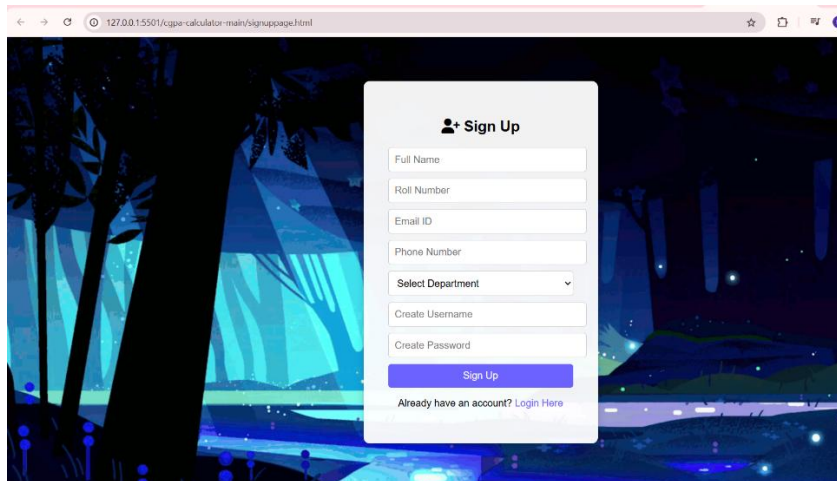


Fig A1.1 Admin & Signup page

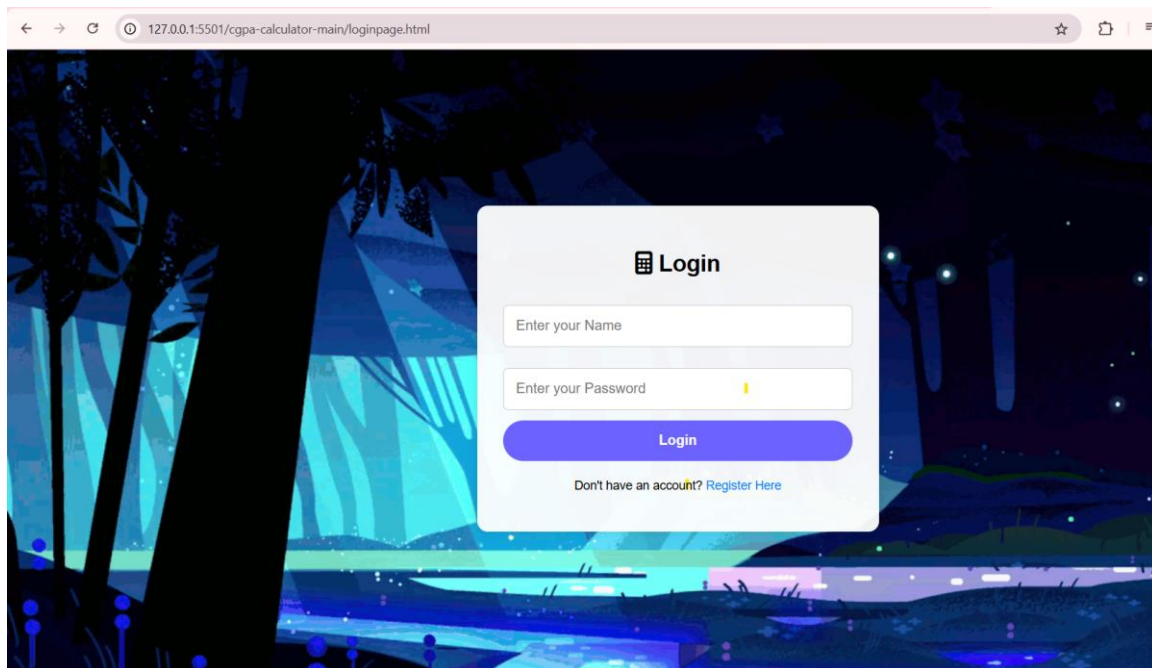


Fig A1.2 User Login Module

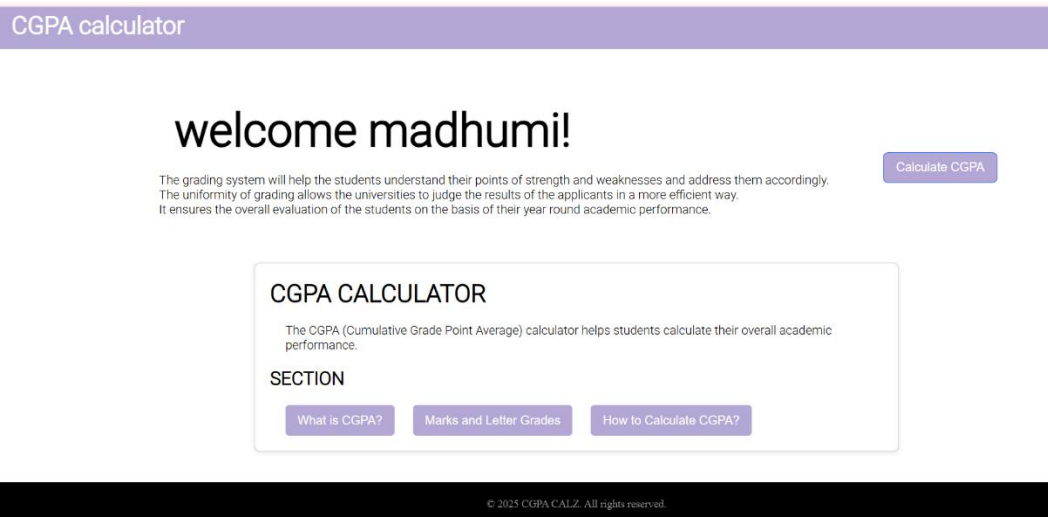


Fig A1.3 Student Login Module

CGPA Calculator		
Semester 1		
Subject	Grade	Credit Hours
Communicative English	<input type="text" value="0"/> ▾	2
Matrices, Differential and Integral Calculus	<input type="text" value="0"/> ▾	4
Engineering Chemistry	<input type="text" value="0"/> ▾	3
Programming for Problem Solving in C	<input type="text" value="0"/> ▾	3
Engineering Graphics	<input type="text" value="0"/> ▾	4
Chemistry Laboratory	<input type="text" value="0"/> ▾	1
C Programming Laboratory	<input type="text" value="0"/> ▾	2
Communication English Laboratory	<input type="text" value="0"/> ▾	1

GPA for Current Semester: N/A
Overall CGPA: N/A

Fig A1.4 GPA & CGPA Calculating module

REFERENCES

1. M. A. Khan, S. A. Khan, and M. S. Khan, "Design and Development of an Efficient CGPA Calculator for Accurate Academic Assessment," *International Journal of Computer Applications*, vol. 182, no. 23, pp. 1–5, Dec. 2018. [Online]. Available: <https://www.ijcaonline.org/archives/volume182/number23/30278-2018917497>(ResearchGate)
2. S. Sharma and R. K. Sharma, "Student Management System and SGPA, CGPA Calculator," *International Journal of New Innovations in Engineering and Technology*, vol. 22, no. 1, pp. 763–770, Apr. 2023. [Online]. Available: <https://www.ijniet.org/wp-content/uploads/2023/04/120.pdf>(IJNIET)
3. B. Griffin, "Grade Conversions: How IEE Accurately Calculates Your GPA from Transcripts," *IEE Blog*, Jun. 24, 2024. [Online]. Available: <https://iee.com/blog/iee-grade-conversions/>(iee.com)
4. A. M. El-Sayed and M. A. El-Moursy, "Converting Cumulative Grade Point Average to an Equivalent Percentage Value Based on a Fuzzy Logic," *International Journal of Computer Applications*, vol. 181, no. 3, pp. 1–7, Jul. 2018. [Online]. Available: <https://www.researchgate.net/publication/356903940>(ResearchGate)
5. Q. Hu and H. Rangwala, "Academic Performance Estimation with Attention-based Graph Convolutional Networks," *arXiv preprint arXiv:2001.00632*, Dec. 2019. [Online]. Available: <https://arxiv.org/abs/2001.00632>(arXiv)
6. C. Cui et al., "Tri-Branch Convolutional Neural Networks for Top-k Focused Academic Performance Prediction," *arXiv preprint arXiv:2107.10424*, Jul. 2021. [Online]. Available: <https://arxiv.org/abs/2107.10424>(arXiv)

7. H. Yao et al., "Predicting Academic Performance for College Students: A Campus Behavior Perspective," *arXiv preprint arXiv:1903.06726*, Mar. 2019.
[Online]. Available: <https://arxiv.org/abs/1903.06726>(arXiv)
8. R. S. Baker and K. Yacef, "The State of Educational Data Mining in 2009: A Review and Future Visions," *Journal of Educational Data Mining*, vol. 1, no. 1, pp.3–17,2009.[Online].Available:
<https://jedm.educationaldatamining.org/index.php/JEDM/article/view/8>(Wikipedia)
9. C. Romero and S. Ventura, "Educational Data Mining: A Review of the State of the Art," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 6, pp. 601–618, Nov. 2010, doi: 10.1109/TSMCC.2010.2053532.(Wikipedia)
- 10.A. Mitrovic, "Fifteen Years of Constraint-Based Tutors: What We Have Achieved and Where We Are Going," *User Modeling and User-Adapted Interaction*, vol. 22, no. 1–2, pp. 39–72, Apr. 2012, doi: 10.1007/s11257-011-9105-9.(Wikipedia)
- 11.S. Nwana, "Intelligent Tutoring Systems: An Overview," *Artificial Intelligence Review*, vol. 4, no. 4, pp. 251–277, Dec. 1990, doi: 10.1007/BF00168958.(Wikipedia)
- 12.R. Freedman, "What is an Intelligent Tutoring System?," *Intelligence*, vol. 11, no. 1, pp. 15–24, 2000.(Wikipedia)
- 13.R. Nkambou, R. Mizoguchi, and J. Bourdeau, *Advances in Intelligent Tutoring Systems*, Springer, 2010.(Wikipedia)

14. S. Ohlsson, "Constraint-Based Student Modeling," *Artificial Intelligence in Education*, vol. 3, no. 4, pp. 429–447, 1992. ([Wikipedia](#))
15. A. Mitrovic and S. Ohlsson, "Evaluation of a Constraint-Based Tutor for a Database Language," *International Journal of Artificial Intelligence in Education*, vol. 10, no. 3–4, pp. 238–256, 1999. ([Wikipedia](#))
16. A. Mitrovic, B. Martin, and P. Suraweera, "Intelligent Tutors for All: Constraint-Based Modeling Methodology, Systems and Authoring," *IEEE Intelligent Systems*, vol. 22, no. 4, pp. 38–45, Jul.–Aug. 2007, doi: 10.1109/MIS.2007.74. ([Wikipedia](#))
17. K. Zakharov, A. Mitrovic, and S. Ohlsson, "Feedback Micro-engineering in EER-Tutor," in *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, 2005, pp. 718–725. ([Wikipedia](#))
18. A. Mitrovic and S. Ohlsson, "Fidelity and Efficiency of Knowledge Representations for Intelligent Tutoring Systems," *Technology, Instruction, Cognition and Learning*, vol. 5, no. 2, pp. 101–132, 2007. ([Wikipedia](#))
19. A. Mitrovic, S. Ohlsson, and D. Barrow, "The Effect of Positive Feedback in a Constraint-Based Intelligent Tutoring System," *Computers & Education*, vol. 60, no. 1, pp. 264–272, Jan. 2013, doi: 10.1016/j.compedu.2012.07.006. ([Wikipedia](#))
20. H. Anderson and M. Koedinger, "Intelligent Tutoring Goes to School in the Big City," *International Journal of Artificial Intelligence in Education*, vol. 8, pp. 30–43, 1997. ([Wikipedia](#))