Tribhuvan University

Faculty of Humanities and Social Science

STUDENT PERFORMANCE EVALUATION SYSTEM

A PROJECT REPORT

Submitted to

Tribhuvan University

Vedas College

In partial fulfillment of the requirements for the Bachelor's in Computer Application

Submitted by

Rachana Khatri (14902071)

Sujal Chhetri Karki (14902086)

April, 2026

Under the Supervision of:

Harendra Raj Bist

Tribhuvan University

Faculty of Humanities and Social Sciences

Vedas College

SUPERVISOR'S RECOMMENDATION

I hereby recommend that this project prepared under my supervision by RACHANA KHATRI, SUJAL CHHETRI KARKI entitled "**STUDENT PERFORMANCE EVALUATION SYSTEM"** in partial fulfillment of the requirements for the degree of Bachelors of Computer Application is recommended for the final evaluation.

_____

Mr. Harendra Raj Bist

Project Supervisor

Department Of Computer Application

Vedas College

Tribhuvan University

Faculty of Humanities and Social Sciences

Vedas College

LETTER OF APPROVAL

This is to certify that this project prepared by RACHANA KHATRI, SUJAL CHHETRI KARKI entitled **"STUDENT PERFORMANCE EVALUATION SYSTEM"** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application has been evaluated. In our opinion, it is satisfactory in terms of scope and quality as a project for the required degree.

_____ Harendra Raj Bist Supervisor Vedas College, Lalitpur	_____ Harendra Raj Bist Program Coordinator-BCA Vedas College, Lalitpur
_____ Internal Examiner	_____ External Examiner

ACKNOWLEDGEMENTS

We would like to take this opportunity to express our heartfelt gratitude to all our respected teachers who have supported this project throughout journey. A very special thanks goes to our supervisor, Mr. Harendra Raj Bist, whose consistent guidance, valuable feedback, and encouragement were crucial in shaping and completing this project.

We are also thankful to Vedas College for providing us with the necessary resources and a supportive learning environment that made this project possible.

Furthermore, we would like to acknowledge the support and motivation we received from our friends, seniors, and family members. Their direct and indirect contributions have played an important role in helping us bring Student Performance Evaluation System to life. We sincerely welcome any feedback or suggestions for further improvement.

Sincerely,
Rachana Khatri (14902071)
Sujal Chhetri Karki (14902086)

ABSTRACT

The Student Performance Evaluation System is a web-based application developed to manage and evaluate student academic performance in an efficient and organized manner. The system is built using PHP, MySQL, HTML, CSS, and runs on an Apache server environment. It follows role-based architecture consisting of three main users: Admin, Teacher, and Student. The admin has full control over the system, including creating and managing student and teacher accounts, assigning subjects, managing exams, approving results, and verifying administrative codes. Teachers can enter and update student marks and manage attendance records. Students can log in using credentials provided by the admin to view their results, attendance, and profile information. The platform integrates e-sewa payment gateway for pre-admission fee processing, enabling secure and convenient online transactions.

Keywords: Student performance evaluation, web-based system, PHP, MySQL, role-based access control, Counting Sort algorithm, eSewa integration, result management.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

CSS- Cascading Style Sheets

DFD- Data Flow Diagram

HTML- Hyper Text Markup Language

IDE- Integrated Development Environment

MVT- Model View Template

UI- User Interface

UX- User Experience

VS Code- Visual Studio Code

XSS- Cross Site Scripting

CHAPTER 1: INTRODUCTION

1.1 Introduction

In modern educational institutions, managing and evaluating student academic performance efficiently is a critical task. Traditional methods of recording grades, calculating results, and ranking students are often time-consuming, error-prone, and lack transparency. With the growing adoption of digital technologies, educational institutions are increasingly seeking reliable, automated solutions to streamline academic management and ensure accurate evaluation.

The Student Performance Evaluation System is a web-based application designed to address these challenges by providing a secure, organized, and user-friendly platform for managing student performance. Built using PHP, MySQL, HTML, and CSS, and hosted on an Apache server, the system follows a role-based architecture with three main types of users: Admin, Teacher, and Student.

The admin has full control over the system, including managing student and teacher accounts, assigning subjects, organizing exams, approving results, and verifying administrative codes. Teachers can enter and update student marks, manage attendance records, and monitor academic progress. Students can log in to view their results, attendance, and personal academic profiles.

The system uses a Counting Sort algorithm to organize student marks and a Ranking Algorithm to assign academic positions, ensuring accurate merit lists and fair evaluation. It also integrates the eSewa payment gateway for secure online fee processing, enhancing convenience for students and parents.

By automating result calculation, ranking, and record management, the system reduces manual errors, saves time, and improves overall efficiency in educational institutions. The Student Performance Evaluation System offers a reliable, secure, and modern digital solution for managing student performance in a systematic and transparent way. Additionally, the system can be further expanded with features like report generation, performance analytics, and notifications to enhance usability and decision-making.

Overall, the Student Performance Evaluation System is a modern solution that enhances the efficiency of academic management. It reduces manual workload, minimizes errors, and improves transparency in the evaluation process.

1.2 Problem Statement

Many schools and colleges struggle to keep track of their student's performance efficiently. Doing everything manually like recording marks, calculating results, ranking students, and managing attendance takes a lot of time and can easily lead to mistakes. Teachers and admins often spend more time on paperwork than helping students improve. Some of the main problems are:

i. Manually calculating results and ranks are slow and prone to errors.
ii. Keeping records of marks, attendance, and student profiles is confusing and unorganized.
iii. Students and teachers don't have a simple way to access results or updates online.

1.3 Objectives

To tackle the challenges in managing student performance, the Student Performance Evaluation System aims to provide a smart, user-friendly, and secure web-based platform with the following solutions:

- To create a role-based system where admins, teachers, and students can access the system easily, with admins managing accounts, subjects, exams, and results.
- To allow teachers to enter and update marks, track attendance, and manage academic records efficiently, reducing manual work and errors.

1.4 Scope and Limitations

Scope of the Project:

• Efficient Academic Management:
The system provides schools and colleges with a centralized platform to manage student marks, attendance, and exam records, making academic operations smoother and more organized.

• Easy Access for Users:
Students, teachers, and admins can access the platform from anywhere. Students can check their results and attendance, while teachers can update marks and attendance easily.

• **Time and Error Reduction:**

Automatic calculation of results, ranking, and report generation reduces manual effort and minimizes errors, improving accuracy and reliability.

• **Streamlined Administration:**

Admins can manage student and teacher accounts, assign subjects, organize exams, and handle fee transactions securely, making overall administration more efficient.

• **Scalability and Future Expansion:**

The system can be expanded to include additional modules such as a performance analytics, report cards, parent access, or notifications, making it adaptable to future needs of educational institutions.

Limitations:

• **Server Dependency:**

The system relies on a running server environment (Apache). If the server has issues or crashes, users won't be able to access the system until it's fixed.

• **Database Capacity:**

The system uses MySQL, which is suitable for small to medium-scale applications. For larger systems, more advanced database solutions may be required.

1.5 Development Methodology

Agile Methodology

Student Performance Evaluation System follows the Agile methodology, an iterative development approach that allows flexibility and continuous improvement throughout the project.

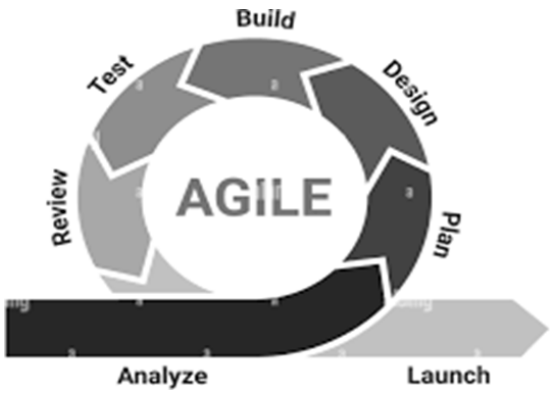


Figure 1.1 Agile Methodology for Student Performance Evaluation System

For this project, the Agile methodology is adopted to ensure a flexible, adaptive, and user-centered development process. Agile is especially suitable for a Student Performance Evaluation System, where features such as student enrollment, grade entry, performance analytics, and automated report generation must be frequently updated based on evolving institutional requirements and academic standards. By breaking down the development into iterative sprints, the team can focus on delivering key modules—such as the teacher portal or student dashboard—in smaller, manageable phases while maintaining continuous testing and improvement. This ensures that the platform remains responsive to real-time feedback from educators, whether it's refining grading rubrics, enhancing data visualization charts, or optimizing the user interface for better accessibility. Overall, using Agile supports the goal of building a reliable, efficient and user-friendly system that adapts to changing requirements.

1.6 Report Organization

This report provides a detailed overview of the development of the Student Performance Evaluation System. The report is as follows:

Chapter 1: Introduces the project, including the background, objectives, problem statement, scope, and limitations.

Chapter 2: Analyzes the existing system used in educational institutions, highlighting its limitations.

Chapter 3: Focuses on system analysis and design, including feasibility studies, data modeling, and process modeling.

Chapter 4: Describes the technologies and tools used to develop the system, including front-end and back-end development, database management, and testing procedures

Chapter 5: Concludes the project by summarizing outcomes, benefits, and key findings.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1. Background of Study

Managing student academic performance is a critical aspect of modern educational institutions. Traditionally, schools and colleges rely on manual methods for recording marks, calculating results, tracking attendance, and generating merit lists. These processes are often time-consuming, prone to errors, and lack transparency, making it difficult for teachers, administrators, and students to access accurate academic information efficiently [1].

With the growing number of students and subjects, there is a rising need for digital solutions that can automate these tasks while providing secure, organized, and easily accessible records [2]. Many institutions still use separate spreadsheets, paper records, or offline tools. This can lead to mismanagement, delayed results, and limited communication between teachers, students, and administrators [2], [3].

The Student Performance Evaluation System is designed to address these challenges by providing a web-based platform that centralizes academic management [4]. It allows admins to manage student and teacher accounts, subjects, and exams; teachers to record and update marks and attendance; and students to view their results and academic profiles anytime [4].

By integrating features like automated result calculation, ranking algorithms, and secure online systems, the platform ensures accuracy, fairness, and efficiency in evaluating student performance [5]. The system also improves transparency and accessibility, enabling better communication among all stakeholders [5].

This background study highlights the technological, functional, and user-focused requirements needed to build a reliable, scalable, and effective academic management system aligned with modern educational needs [6].

The Student Performance Evaluation System is designed to modernize institutional management by transitioning from fragmented, manual record-keeping to a centralized digital framework. By replacing error-prone spreadsheets and paper ledgers with a robust, structured web-based architecture, the platform ensures seamless data flow between administrative, teaching, and student layers. Key features such as real-time attendance tracking, automated mark entry, and the integration of high-speed sorting algorithms for merit list generation ensure both operational efficiency and accuracy [7].

2.2. Study of Existing System (Literature Review)

a. School Management Systems (SMS)

School Management Systems are used to manage administrative and academic activities in schools. They help track attendance, grades, and performance reports. Systems like Fedena and OpenSIS provide features such as automated report cards, exam scheduling, and parent-teacher communication. These systems highlight the importance of user-friendly dashboards and real-time performance tracking for effective evaluation [1], [2].

b. Online Learning Platforms (Moodle, Google Classroom)

Online platforms like Moodle and Google Classroom allow teachers to assign tasks, conduct assessments, and monitor student progress digitally. They provide analytics on student performance and activity. Such features help identify weak areas, generate insights, and support personalized learning in a Student Performance Evaluation System [3], [4].

c. Existing Local Systems in Nepal

Many schools in Nepal use simple locally developed software or Excel-based systems. Although easy to use, they often lack real-time updates, automated grading, and proper security controls. These limitations show the need for a more secure and efficient system with different access rights for teachers and administrators [5].

d. Data-Driven Performance Analysis

Educational systems use data analysis methods such as Counting Sort algorithms, score aggregation, and trend analysis to evaluate student performance. These methods help rank students objectively and identify performance trends, ensuring fair and accurate evaluation [6].

e. Ethical and Security Considerations

Student data must be handled carefully to maintain privacy and fairness. The system should include secure authentication, protected data storage, and controlled access to ensure information safety and reliability [7].

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

The Student Performance Evaluation System is a web-based platform designed to manage and monitor student academic performance efficiently. The system allows teachers to record grades, track attendance, and evaluate overall student performance with ease. Students can create accounts to view their results, track progress, and receive feedback on their academic activities. Administrators manage the platform by creating and updating student and teacher accounts, processing exam results, handling queries, and maintaining records securely. The system is designed for accuracy, reliability, and user convenience, supporting features such as automated ranking using Counting Sort algorithm, real-time performance updates, and secure pre-admission payments through eSewa. Overall, the Student Performance Evaluation System ensures smooth and transparent experience for students, teachers, and administrators, helping educational institutions efficiently manage academic records and make informed decisions.

3.1.1 Requirement Analysis

The Student Performance Evaluation System requires a clear definition of both functional and non-functional requirements to ensure efficient and user-friendly experience for students, teachers, and administrators. Functional requirements focus on student registration, grade entry, attendance tracking, ranking, and pre-admission payments via eSewa. Administrators must be able to create and manage student and teacher accounts, update results, monitor performance, and generate reports. Non-functional requirements emphasize performance, security, accessibility, and system reliability under various loads, ensuring smooth operation and scalability for future expansion.

i. Functional Requirements

User Registration & Authentication:

- Students and teachers must be able to sign up, log in, and manage their profiles.
- Include features such as password reset, email verification, and role-based access control.

Student Records Management:

- Admin should be able to create, update, and delete student and teacher records.

- Track student information such as personal details, class, subjects, and academic history.

Performance & Grades:

- Teachers can enter grades, attendance, and exam results for each student.
- The system should automatically calculate total marks, percentage, and ranks using Counting Sort algorithm.

Pre-admission & Payment Integration:

- Students can complete pre-admission forms and payments securely through eSewa.
- Admin can verify payment status and update student enrollment records accordingly.

Report Generation:

- Generate detailed performance reports for students, classes, and subjects.
- Provide downloadable or printable report cards for student reference.

Search & Filtering:

- Admins and teachers can search for students by name, ID, class, or performance metrics.
- Filtering options to view top-performing or low-performing students.

Notifications & Feedback:

- System can notify students about results, attendance alerts, and admission updates.
- Admins can provide feedback or announcements to students and teachers.

ii. Non-Functional Requirements

Performance:

- The system should load pages and process results quickly, even with multiple simultaneous users.
- Must handle large datasets of students and grades efficiently.

Scalability:

- Should support increasing numbers of students, classes, and subjects as the school grows.

Usability:

- Interface must be simple, responsive, and accessible to all users, including students, teachers, and administrators.
- Must provide smooth navigation across desktops, tablets, and mobile devices.

Security:

- All student data, grades, and payment information must be encrypted.
- Proper authentication and authorization mechanisms must protect against unauthorized access.

Availability & Reliability:

- The platform must be accessible 24/7 with minimal downtime.
- Regular backups should be performed to prevent data loss.

Compatibility:

- The system should function across all major browsers (Chrome, Firefox, Edge, Safari) and standard devices.

System Requirements

i. **Software Requirements:**

- Code editor (e.g., VS Code, Sublime Text)
- Operating System: Windows 7/10/11 or higher
- Web Browser: Chrome, Firefox, Edge
- Database: MySQL or SQLite
- Front-end: HTML, CSS, Javascript
- Backend: PHP, Xampp

ii. **Hardware Requirements:**

- Laptop/PC with at least 4GB RAM and dual-core processor
- Stable internet connection for testing and deployment
- Printer (optional – for generating report cards and documentation)

Use Case Diagram

• A Use Case Diagram visually represents how different users interact with the Student Performance Evaluation System to perform various tasks. It helps identify the key functionalities of the system and the roles of its users. In this project, the primary users are Student, Teacher, and Admin.

• Students can register, log in, view grades, track attendance, check performance reports, and complete pre-admission payments via eSewa.

• Teachers can record grades, update attendance, generate performance reports, and provide feedback to students.

• Admin manages student and teacher accounts, oversees exam results, monitors performance analytics, and handles pre-admission processes.

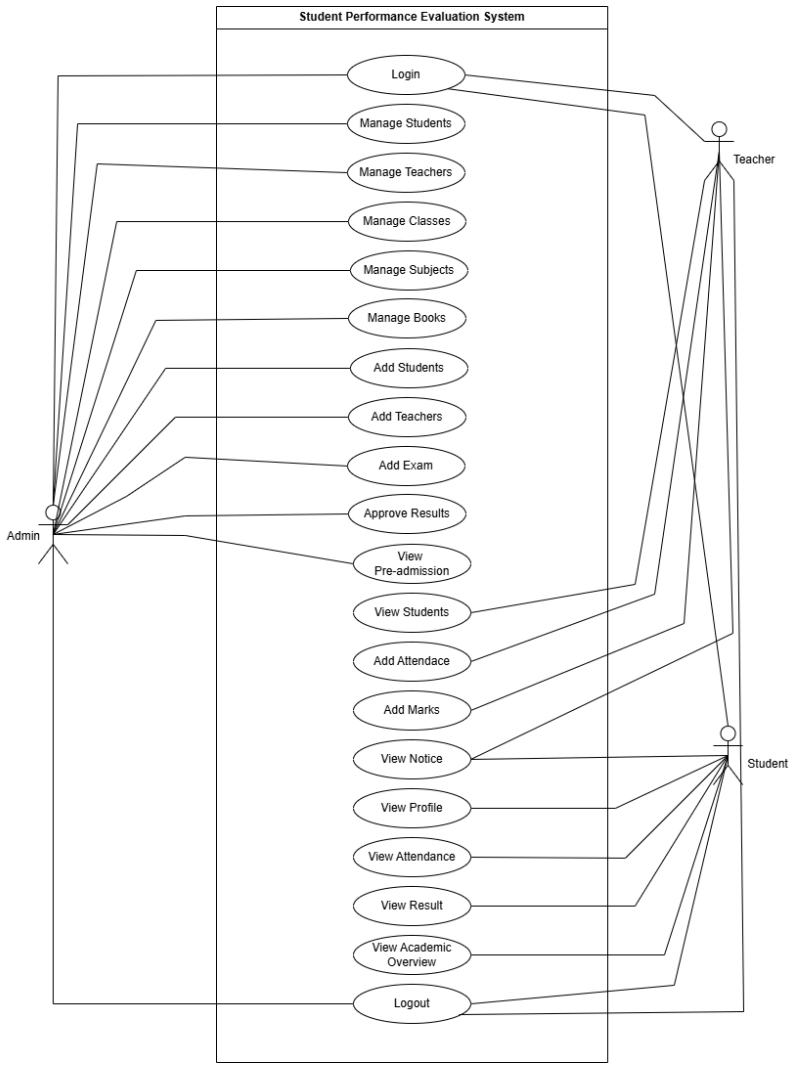


Figure 3.1: Use Case Diagram of Student Performance Evaluation System

Actors in the System

Admin: The administrator oversees the entire platform and manages student and teacher accounts, exam results, performance reports, and pre-admission processes.

Teacher: The teacher records grades, updates attendance, generates reports, and provides feedback to students.

Student/User: A student who can view grades, track performance, monitor attendance, and complete pre-admission payments via eSewa.

Admin Functionalities

The admin has access to the following operations:

- Login: Securely logs into the dashboard using valid credentials.
- Student Management: Adds, updates, or removes student and teacher records.
- Performance Management: Monitors grades, calculates rankings using Counting Sort algorithm, and ensures accurate academic records.
- Admission Management: Oversees pre-admission forms, verifies payments, and confirms enrollment.
- Logout: Safely logs out after completing administrative tasks.

Teacher Functionalities

Teachers can interact with the system through the following features:

- Login: Access their account using secure credentials.
- Grade Entry: Record student grades for exams and assignments.
- Attendance Management: Mark and updating student attendance.
- Logout: Securely exit the system after completing tasks.

Student Functionalities

Students can interact with the system through the following features:

- Login: Access their account securely.
- View Grades: Check exam scores, total marks, and ranking.
- Attendance Tracking: Monitor their attendance records.
- Pre-Admission & Payment: Complete admission forms and payments via eSewa.
- Logout: Securely exit the platform.

Non-Functional Requirements

a. **Availability:**

The system will be deployed on a live server and accessible 24/7 with minimal downtime.

b. **Security**:

All user data, grades, and payment transactions will be encrypted to ensure safety and prevent unauthorized access.

c. **Performance**:

Developed using HTML, CSS, Javascript and Bootstrap.

d. **Reliability**:

A robust backend and well-tested features ensure smooth system operation with minimal errors or crashes.

e. **User Friendliness:**

The interface is clean, intuitive, and mobile-responsive, making it suitable for students, teachers, and administrators of all technical backgrounds.

3.1.2 Feasibility Study

a. Operational Feasibility:

The system uses a combination of HTML, CSS, and PHP to create a responsive platform.

b. Economic Feasibility:

By using open-source technologies and minimal server requirements, the cost of development and maintenance is low.

c. Schedule Feasibility:

The project can be developed and deployed within a reasonable timeline. The chosen technologies allow for rapid development, easy testing, and smooth deployment.

Gantt Chart

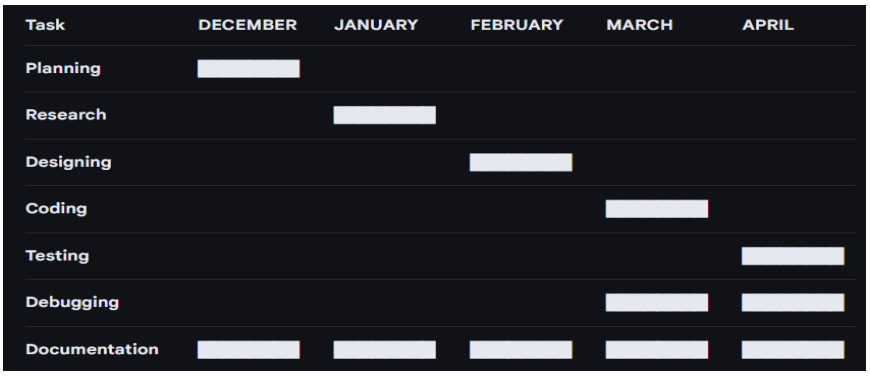


Figure 3.2 Gantt chart

3.1.3 Data Modeling

1. Entity Relationship Diagram

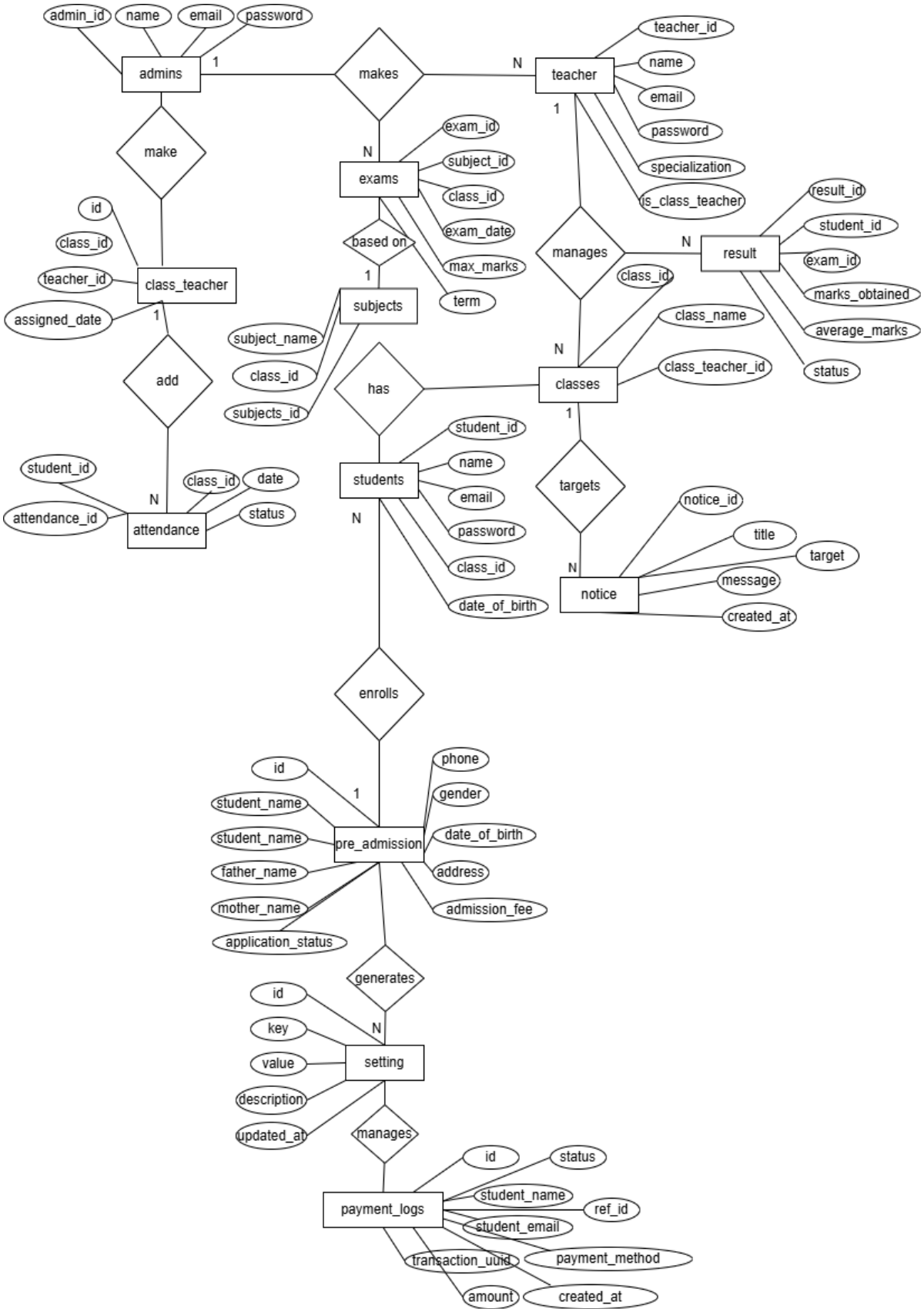


Figure 3.3: Entity Relationship Diagram of Student Performance Evaluation System

In the above E-R Diagram (Entity-Relationship Diagram), we have 8 entities, each connected through relationships and described by their respective attributes. This ER diagram helps us visualize the overall data flow and structural organization of the Student Performance Evaluation System. Below is a breakdown of each entity along with their relationships and cardinality:

i. Admin: The admin entity has four attributes: Id which is the primary key, first name, last name, and email. The admin manages multiple entities including teacher, student, and subject, with one-to-many cardinality, meaning a single admin can manage many teachers, students, and subjects in the system.

ii. Teacher: The teacher entity consists of five attributes: Id which is the primary key, name, email, phone.no, and department. A teacher can evaluate multiple students, assign marks, and manage multiple subjects, forming a one-to-many relationship with student performance and subject entities.

iii. Student: The student entity contains six attributes: Id as the primary key, name, email, phone.no, address, and class. A student can have multiple performance records, receive multiple evaluations, and be enrolled in multiple subjects, resulting in one-to-many relationships with performance and enrollment entities.

iv. Subject: The subject entity includes three attributes: Id which is the primary key, name, and code. A subject can be assigned to multiple students and taught by multiple teachers, forming many-to-many relationships with student and teacher entities.

v. Performance: The performance entity consists of five attributes: Id which is the primary key, student, subject marks, and grade. Each performance record is associated with one student and one subject, indicating many-to-one relationships from performance to both student and subject entities.

vi. Class: The class entity has three attributes: Id as the primary key, name, and section. A class can include many students, forming a one-to-many relationship from class to student.

vii. Evaluation: The evaluation entity includes five attributes: Id as the primary key, teacher, student, date, and remarks. A teacher can evaluate multiple students, and a student can receive multiple evaluations, forming many-to-many relationships between teacher and student through evaluation.

viii. Attendance: The attendance entity has five attributes: Id which is the primary key, student, date, status, and subject. A student can have multiple attendance records for different subjects, forming a many-to-one relationship from attendance to student and subject entities.

3.1.4. Process Modelling

1. Level 0 Diagram

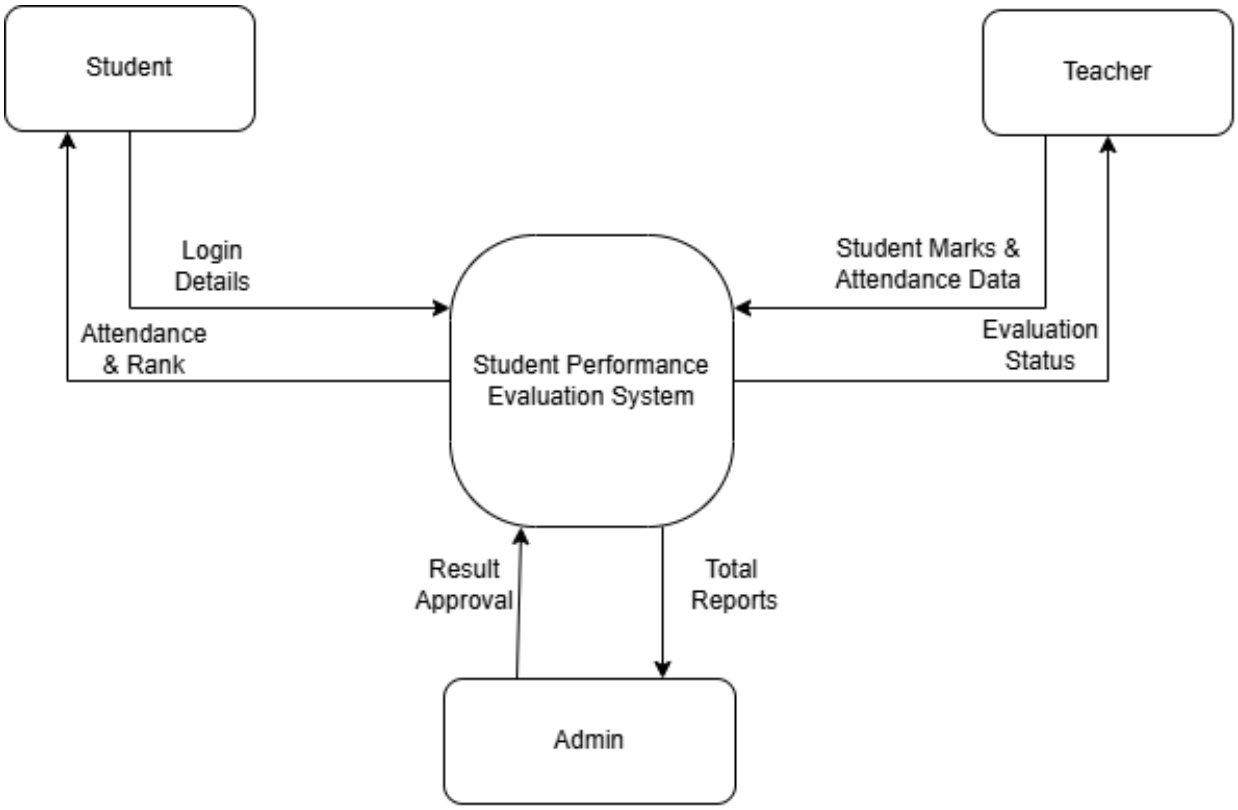


Figure 3.4: Level 0 Diagram of Student Performance Evaluation System

The diagram represents the Context Level Diagram (Level 0 DFD) of the Student Performance Evaluation System, illustrating the system as a single high-level process interacting with its external entities. This diagram defines the system boundaries and the primary data flows between the platform and its key actors: Admins, Teachers, and Students.

This Level 0 DFD does not detail internal logic or database tables but provides a broad overview of how information moves between the actors and the central system. It serves as the foundation for the Level 1 DFD, where these processes are broken down into specific sub-processes like User Authentication, Result Processing, and Report Generation.

- **DFD Diagram**

• DFD Level-1

DFD Level-1 of Admin

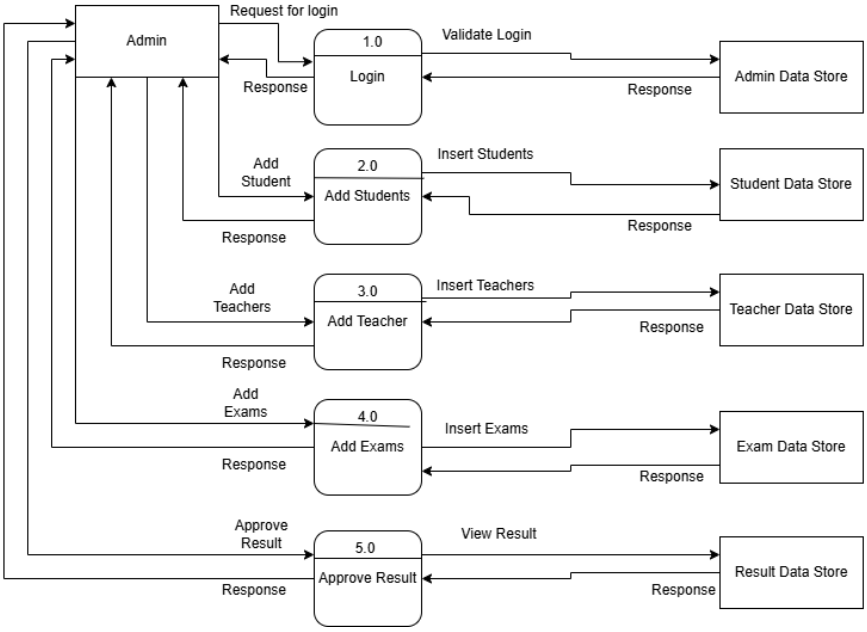


Figure 3.5: DFD Level-1 of Admin

The Student Module illustrates a user-centric information flow focused on data retrieval and personal record management. The process begins with the Login (1.0) process, which validates student credentials against the Student Data Store to grant secure access. Once authenticated, the system facilitates three primary inquiry-based flows: the See Attendance (2.0) process, which pulls real-time records from the Attendance Data Store; the See Result (3.0) process, which retrieves finalized grades from the Result Data Store; and the See Profile (4.0) process, which fetches personal details from the Profile Data Store.

DFD Level-1 of Teacher

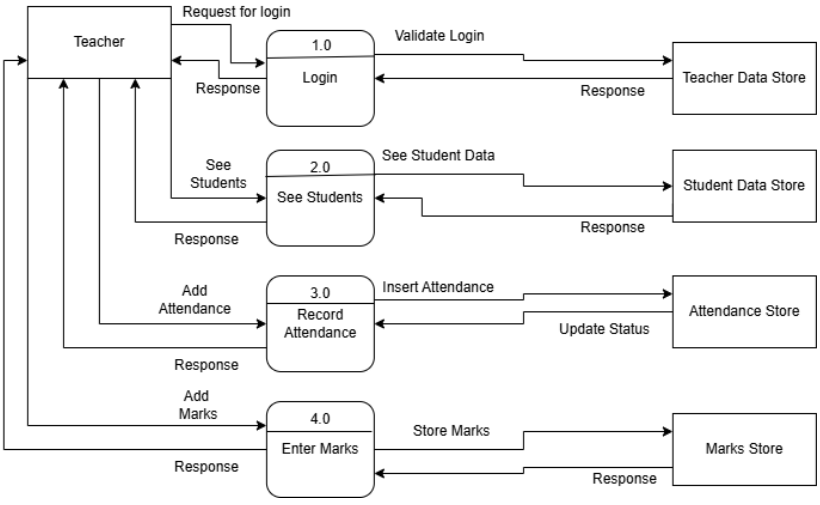


Figure 3.6: DFD Level-1 of Teacher

The Teacher Module represents the primary data-entry pipeline where raw classroom information is converted into formal system records. Following a successful Login (1.0) via the Teacher Data Store, the instructor interacts with three core management processes. The See Students (2.0) process allows the teacher to query the Student Data Store for class rosters and individual student profiles. The operational core involves the Record Attendance (3.0) and Enter Marks (4.0) processes; these act as data gateways that take direct input from the teacher and perform "Insert" or "Store" operations into the Attendance and Marks Data Stores, respectively.

DFD Level-1 of Student

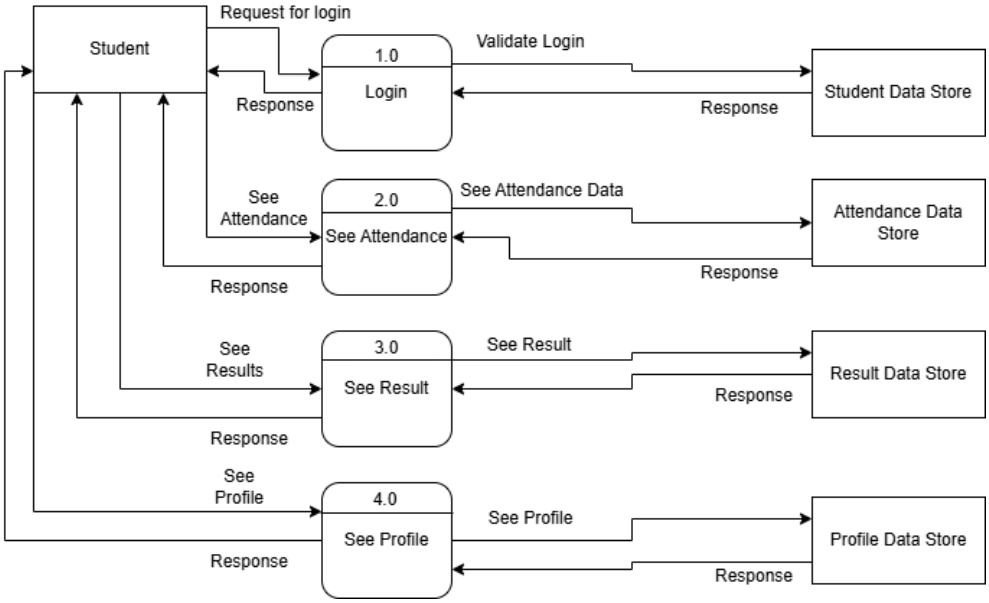


Figure 3.7: DFD Level-1 of Student

The Admin Module serves as the system's high-level governance and configuration layer, managing the entities and final outputs of the evaluation cycle. After a secure Login (1.0) into the Admin Data Store, the administrator oversees the system’s structural integrity through the Add Students (2.0), Add Teacher (3.0), and Add Exams (4.0) processes. These processes function as administrative entry points that populate the respective Student, Teacher, and Exam Data Stores with foundational data. The workflow culminates in the Approve Result (5.0) process, which acts as a final quality-control gate. This process pulls data from the Result Data Store for review, allowing the Admin to finalize and authorize the release of performance reports, ensuring the system's outputs are official and verified.

3.2 System Design

3.2.1. Architecture Design

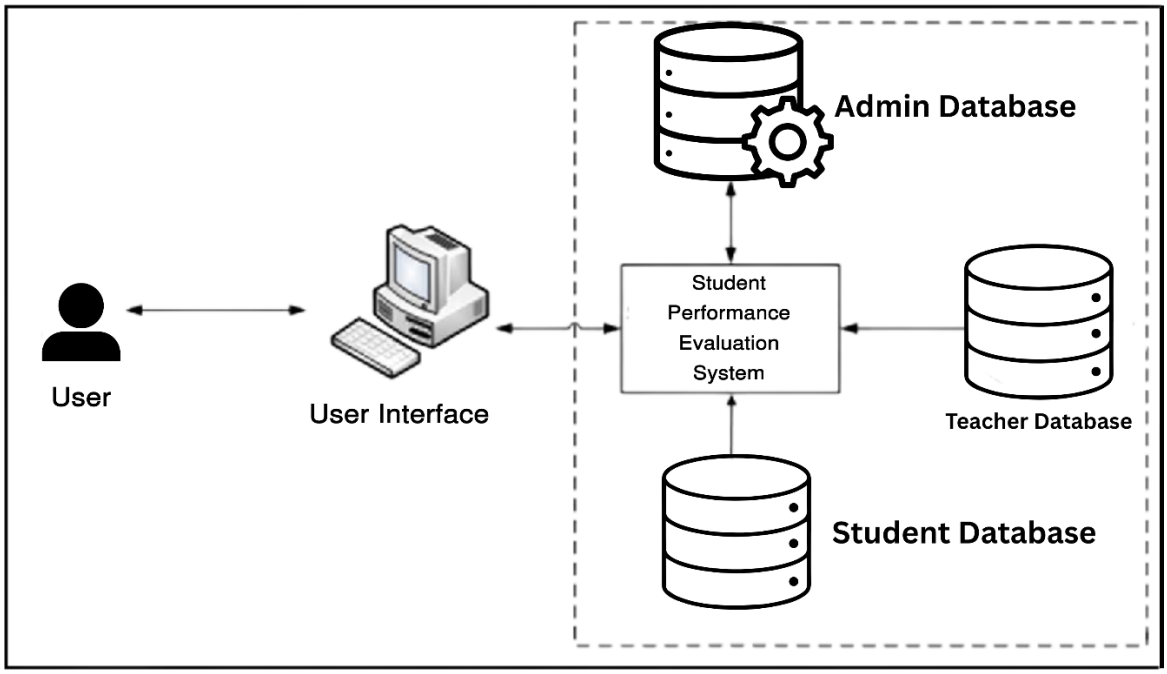


Figure 3.8: Architecture Design of Student Performance Evaluation System

The Student Performance Evaluation System is developed using a simple web-based architecture to ensure smooth interaction between user inputs and data processing. In this system, the user interface acts as the front-end where Students, Teachers, and Admins can input data and view performance reports. When a user performs an action, the system processes the request using backend logic such as validating login credentials, calculating ranks, and managing academic records, and then communicates with the database to store or retrieve information.

- User Interface = Front-end (HTML, CSS, JavaScript)
- System Processing = Backend Logic (PHP)
- Database (Admin, Teacher, Student) = MySQL

This structure allows a seamless flow of information, where data is efficiently processed and accurately displayed to users for proper academic management and decision-making.

3.2.2. Database Schema Design

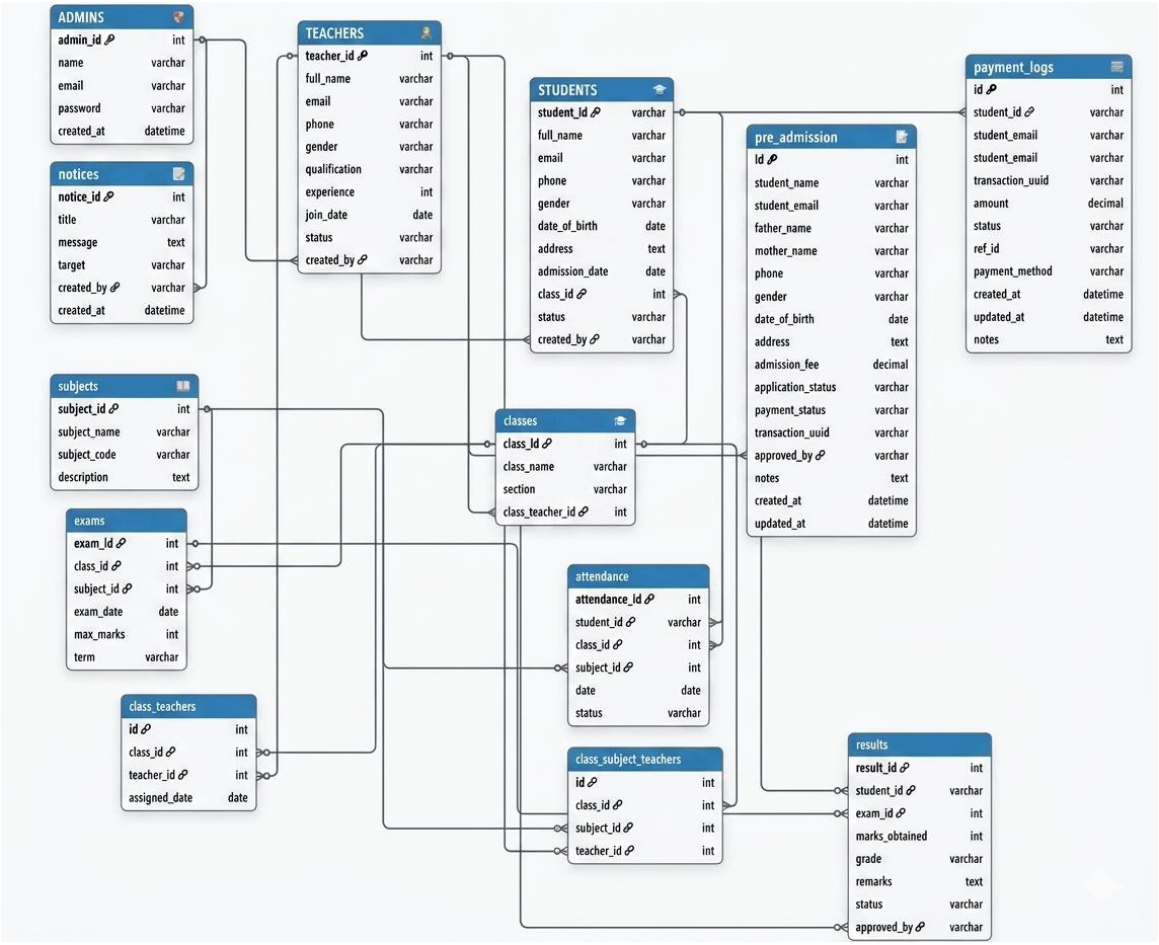


Figure 3.9: Database Schema Design

The Student Performance Evaluation System database schema includes key entities like students, teachers, subjects, classes, performance records, attendance, evaluations, and admins. Tables are connected using primary keys for unique records and foreign keys to establish relationships. For example, students are linked to classes and performance records, and teachers are linked to subjects and evaluations. This structure ensures smooth data management, accurate performance tracking, and efficient handling of student and teacher information within the system.

3.2.3. Interface Design

i. Login Page:

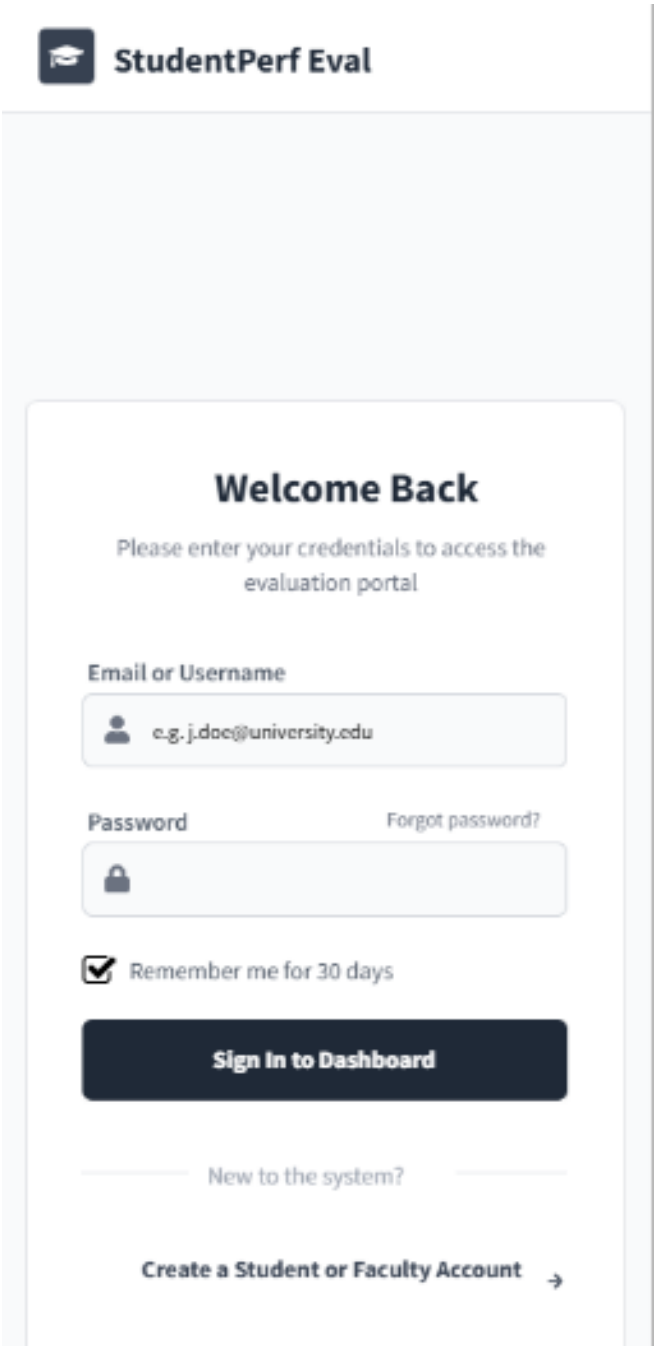


Figure 3.10: Wireframe of login Page

The wireframe presents a simple and user-friendly login page designed for the Student Performance Evaluation System. At the top of the page, a navigation bar is included, displaying the system logo, dashboard link, features section, search option, and login/register buttons.

ii. Main Page:

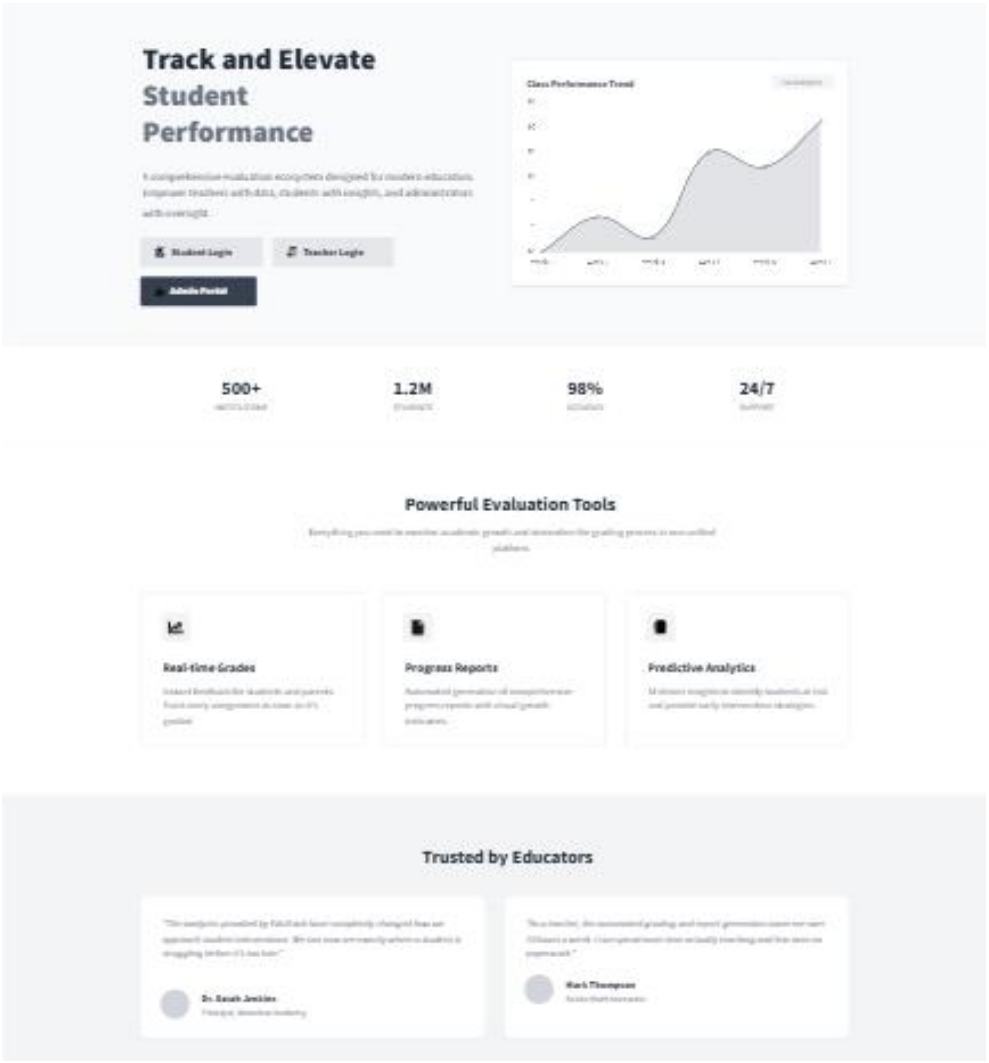


Figure 3.11: Wireframe of Main menu

The wireframe of the homepage for the Student Performance Evaluation System presents two main sections: “Student Performance Overview” and “Top Performing Students.”

The performance overview section displays summarized information about students’ academic progress, including grades, attendance, and overall performance indicators. This allows users, such as teachers or administrators, to quickly assess student performance at a glance.

Below this, the top-performing students section highlights students with outstanding academic results based on evaluation criteria such as marks, participation, or consistency. This helps in recognizing high achievers and encouraging motivation among students.

Together, these sections provide a balanced and informative interface that combines detailed performance insights with recognition of excellence, while maintaining a clean and organized layout for easy navigation and efficient data access.

iii. Admin Page:

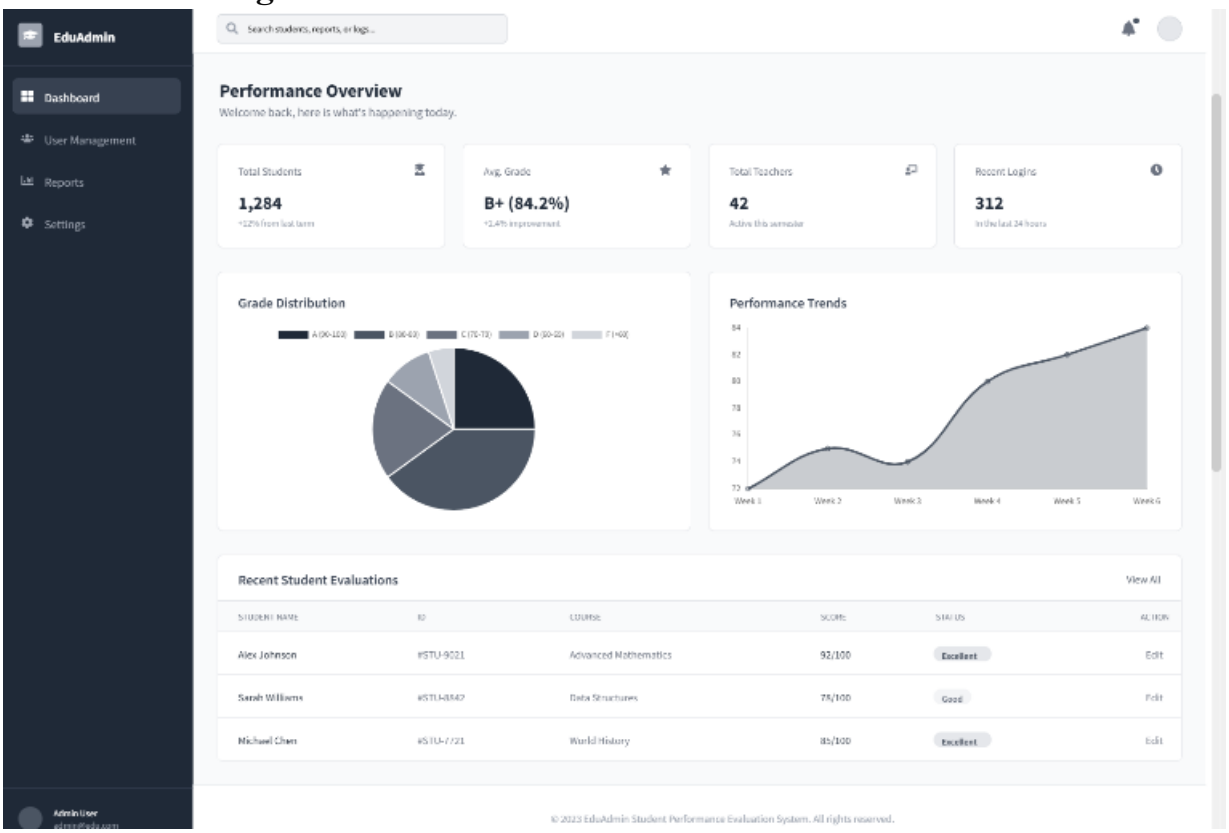


Figure 3.12: Wireframe of Admin Page

The wireframe shows the admin dashboard of the Student Performance Evaluation System with a simple and organized layout. The top section includes navigation options like dashboard, student management, reports, and logout, allowing easy access to different parts of the system.

The page provides filters and a search option to view student records easily. These features help the admin quickly find specific students and sort data based on different criteria.

The main section displays student details such as name, ID, grades, and performance, along with action buttons like “View,” “Edit,” and “Delete.” This makes it easy to manage student information directly from the dashboard.

Pagination controls are included at the bottom for smooth navigation through multiple records. This keeps the interface clean and avoids overcrowding of data.

Icons and simple visual elements can also be used to improve readability and make actions more understandable. The layout is responsive, meaning it can work properly on different screen sizes like laptops and tablets.

Overall, the design is clean, user-friendly, and focused on efficient management of student data, helping administrators perform tasks quickly and effectively.

iv. Teacher Dashboard:

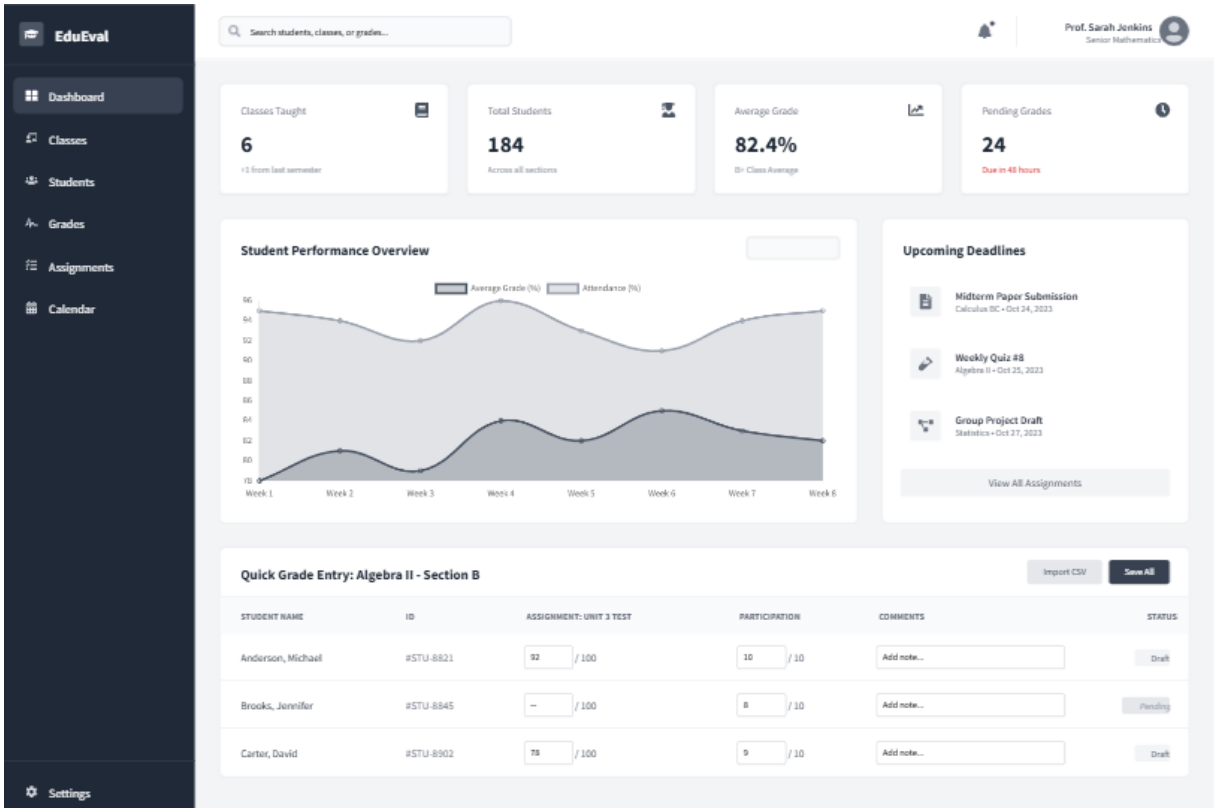


Figure 3.13: Wireframe of Teacher Dashboard

The wireframe shows the teacher dashboard of the Student Performance Evaluation System with a clean and simple layout. It displays student performance details such as grades, attendance, and overall progress, helping teachers monitor student development effectively.

Teachers can select options like class or subject to view specific data. These filtering options make it easier to focus on particular groups of students and analyze their performance in a structured way.

The page also includes features to add or update marks and view detailed student reports. This allows teachers to keep records up to date and maintain accuracy in evaluation.

A section for feedback or remarks is included to support evaluation. Teachers can provide comments on student performance, strengths, and areas for improvement, which helps in better academic guidance.

The interface is designed to be user-friendly with clear labels and organized sections, reducing complexity and saving time during daily use. It may also support secure login access to ensure that only authorized teachers can manage student data.

Overall, the design focuses on easy access to student information, accurate performance tracking, and efficient management, enabling teachers to carry out their responsibilities effectively.

v. Students Portal:

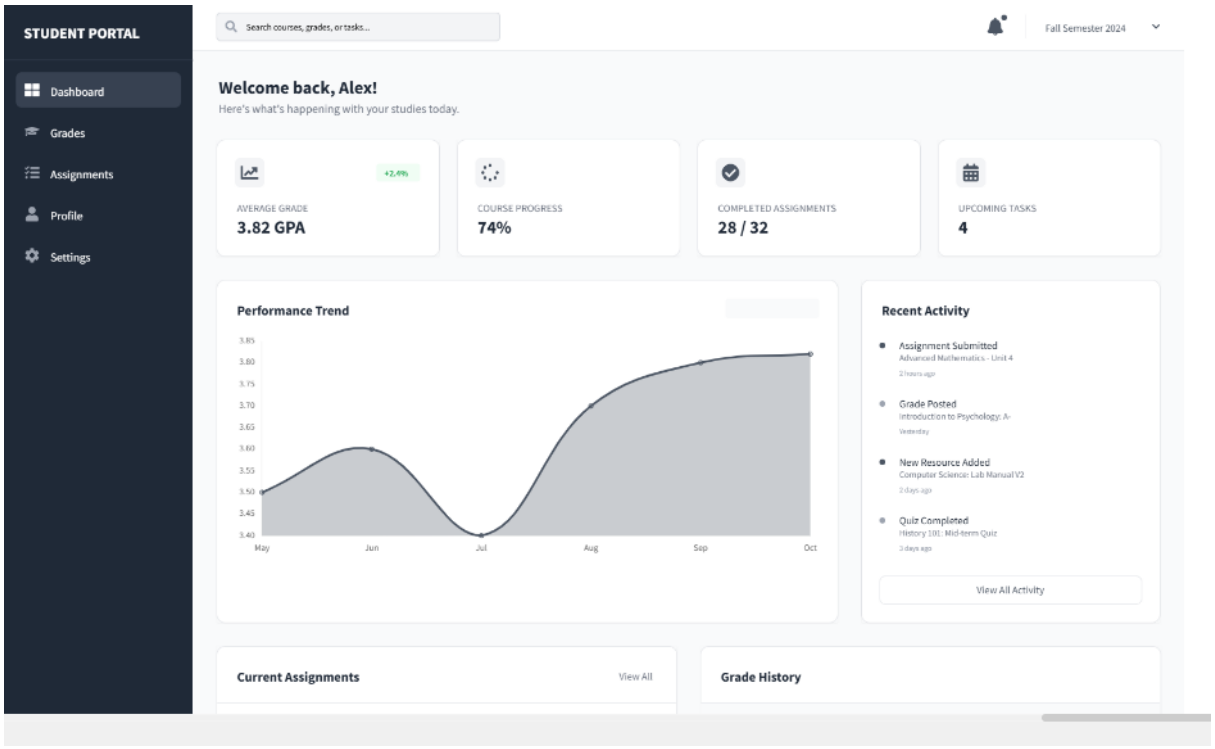


Figure 3.14: Wireframe of Student Portal

The wireframe shows the student portal of the Student Performance Evaluation System with a simple and user-friendly layout. It allows students to view their grades, performance, and feedback in a clear and organized manner. Students can check subject-wise results and overall progress, helping them understand their strengths and areas that need improvement. The information is presented in an easy-to-read format for better understanding.

Additionally, the portal may include features like progress charts or performance indicators, allowing students to visually track their improvement over time. This makes learning more interactive and engaging. The system can also provide options to download reports or view past performance records, helping students keep track of their academic history.

The interface is designed with simplicity and clarity, ensuring that students can easily navigate through different sections without confusion. It may also support secure login to protect student data and privacy.

Overall, the design is clean, efficient, and focused on providing easy access to academic information

3.2.4. Physical DFD

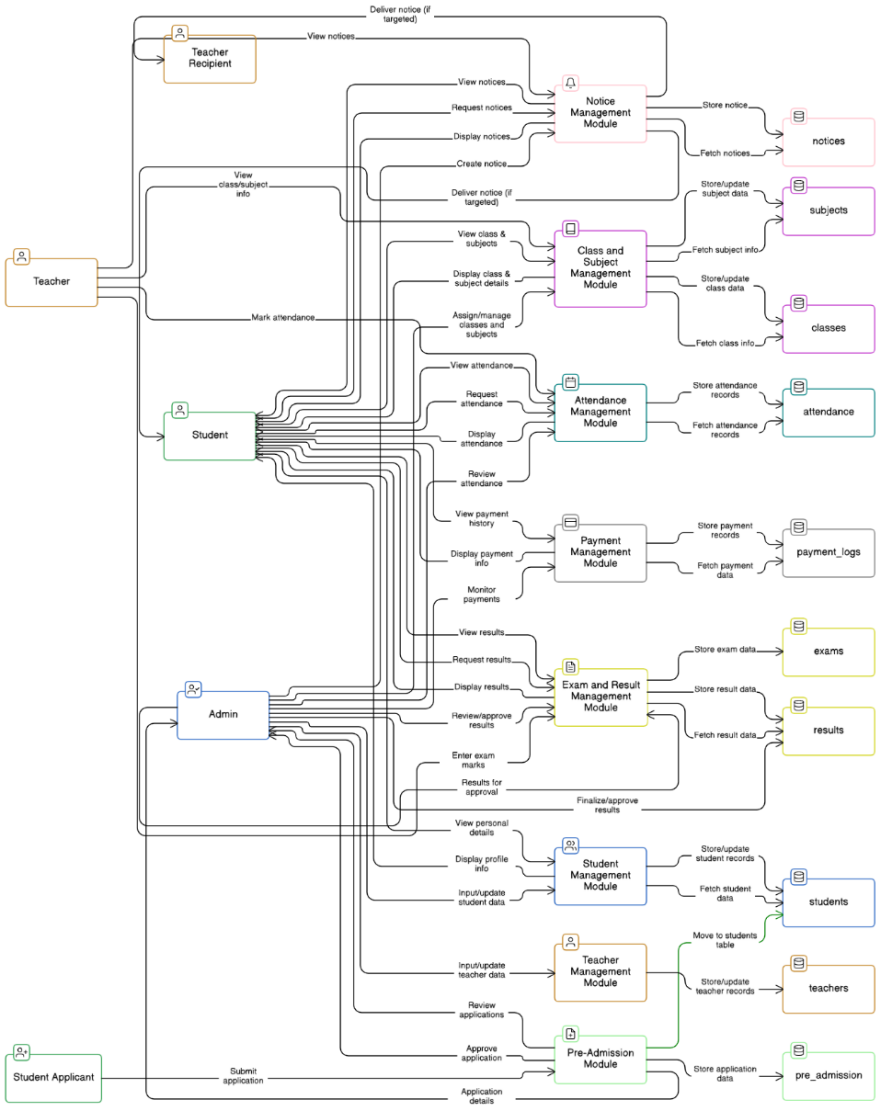


Figure 3.15: Physical DFD of Student Performance Evaluation System

The provided diagram illustrates the Detailed Data Flow for the Student Performance Evaluation System, showcasing how diverse academic modules integrate with backend databases. The architecture is organized into eight specialized functional blocks: Pre-Admission, Student Management, Teacher Management, Attendance, Payment, Class/Subject, Exam/Result, and Notice Management. Data flows are initiated by three primary actors—Admins, Teachers and Students—who interact with the system to perform tasks ranging from initial enrollment and attendance tracking and exam grading. Each module serves as a bridge, fetching and storing data in dedicated tables (such as results, attendance, and payment_logs) to ensure data integrity. This comprehensive flow ensures that once a Teacher inputs marks, the Admin can review, approve, and finalize performance reports, creating a seamless transition from raw administrative data to finalized academic evaluations.

3.3 Algorithm

3.3.1. Ranking Algorithm

Algorithm: Student Ranking System Using Counting Sort algorithm

Description:

The **Student Performance Evaluation System** requires an efficient method to process and rank large volumes of student academic data. To optimize performance, the system implements a **Counting Sort** algorithm. This is a non-comparison-based sorting technique that operates in **Linear Time**, making it significantly faster than standard sorting methods for the specific use case of academic grading.

Algorithmic Logic:

The algorithm avoids the overhead of comparing student marks against each other. Instead, it uses the frequency of marks to determine placement.

1. **Initialization:** An auxiliary array (Count Array) is created with a size of 101, representing the possible integer range of marks (0 to 100).
2. **Frequency Mapping:** The system iterates through the student records. For every student, their average mark (rounded to an integer) serves as an index in the Count Array, where the student object is stored.
3. **Linear Reconstruction:** The system traverses the Count Array in descending order (from index 100 down to 0). This automatically assembles a sorted list of students based on academic performance.
4. **Rank Generation:** The final rank is assigned based on the student's position in the newly reconstructed array, where $Rank = Index + 1$.

Mathematical Model:

The efficiency can be modeled by the following equation:

$T(n) = \boldsymbol{\theta}\ (n + k)$

In the context project, where k = 101, the growth rate of the execution time is strictly proportional to the number of students (n), ensuring the system remains responsive as the database scales.

Academic Depth: It uses terms like "θ notation" and "non-comparison-based," which examiners look for.

Practicality: It links the code directly to the performance of your system.

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1. Implementation

4.1.1 Tools used

CASE Tools

a. Draw.io Software

Draw.io was used to design diagrams such as ERD, DFD, and Use Case diagrams. It offers both automated and manual layout options for flexible design. For the Student Performance Evaluation System project, we used Draw.io to design diagrams such as the context diagram, use case diagram, and ER diagram.

b. Microsoft Word

Microsoft Word was used for documentation and report preparation. Throughout the development of the Student Performance Evaluation System project, we used Microsoft Word to document the entire process from the initial planning stages to the final report ensuring clear and organized presentation of all project-related information.

c. Programming Languages

HTML

HTML (HyperText Markup Language) is the foundation of web development used to create and structure content on the web. It defines elements like headings, paragraphs, images, and links, allowing browsers to render webpages that users can interact with.

CSS

CSS (Cascading Style Sheets) is used to style and visually design HTML content. It controls layout, fonts, colors, spacing, and responsiveness, enabling consistent and appealing presentation across different devices.

JavaScript

JavaScript is a client-side scripting language used to add interactivity to web pages. It allows dynamic content updates, user input validation, animations, and communication with servers without reloading the page, enhancing the user experience.

PHP (Hypertext Preprocessor)

A server-side scripting language used to handle backend logic, process form data, manage sessions, and interact with the database.

XAMPP

A local development environment that provides tools like Apache server and MySQL database, allowing you to run and test your PHP-based web application locally.

d. Database Platform

MySQL

MySQL is a widely used relational database management system used to store and manage application data. It works efficiently with PHP and is commonly included in local development environments like XAMPP. It allows structured storage of data in tables and supports operations such as inserting, updating, retrieving, and deleting records using SQL (Structured Query Language). MySQL is suitable for medium to large-scale web applications due to its reliability, performance, and scalability.

4.1.2. Implementation of the System

Admin Login

Admins can log in to the system using their registered username and password. The system verifies credentials using PHP and database validation. Only authorized admins can access the dashboard, while unauthorized users are restricted. Admin data is securely stored in the MySQL database.

Add Student Records

The admin can add new student details through a form that includes information such as student name, roll number, class, and marks. When the form is submitted, PHP processes the data and stores it in the MySQL database. If successful, the system redirects to the student list page; otherwise, an error message is displayed.

```
$sql = "INSERT INTO students (name, email, password, class_id, date_of_birth, gender)
    VALUES (?, ?, ?, ?, ?, ?)";
```

Edit Student Records

Admins can update student information such as marks or personal details. The student record is retrieved using a unique ID, and updated data is saved back to the database using SQL queries.

```
$stmt = $conn->prepare("SELECT * FROM students WHERE student_id=?");

$stmt->bind_param("i", $student_id);

$stmt->execute();
```

Delete Student Records

To delete a student, the admin selects the students, and its ID is passed to the backend. The student object is retrieved using the ID and removed from the database.

```
$stmt_delete = $conn->prepare("DELETE FROM students WHERE student_id = ?");
   $stmt_delete->bind_param("s", $student_id);
   $stmt_delete->execute();
```

User Module (Student)

View Performance

Students can log in and view their academic performance, including marks, grades, and subject-wise results. The system retrieves data from the database and displays it in a structured format.

```
SELECT r.marks_obtained, r.average_marks, e.exam_date, e.term, sub.subject_name, e.max_marks
FROM results r
JOIN exams e ON r.exam_id=e.exam_id
JOIN subjects sub ON e.subject_id=sub.subject_id
WHERE r.student_id=? AND r.status='Approved' ";
```

Result Analysis

The system calculates total marks, percentage, and grades using PHP logic. It helps students understand their performance level.

```
$total_percentage = 0;
$total_subjects = 0;
$rows = [];
while($r = $results->fetch_assoc()) {
   $percentage = ($r['max_marks'] > 0) ? (($r['marks_obtained'] / $r['max_marks']) * 100) : 0;
   $r['percentage'] = $percentage;
   $rows[] = $r;
   $total_percentage += $percentage;
```

```
    $total_subjects++;
}
$overall_avg = ($total_subjects > 0) ? ($total_percentage / $total_subjects) : 0;
```

Search / Filter

admins can search for records using student name, roll number, or class.

```
$search = "";
if (isset($_GET['search'])) {
    $search = mysqli_real_escape_string($conn, $_GET['search']);
}
```

Ranking Algorithm

On the result page, student performance is displayed based on their ranking. The system calculates the total marks and percentage of each student from the algorithm. Using PHP and SQL queries, students are sorted in descending order according to their percentage. The system then assigns ranks to students (e.g., Rank 1, Rank 2, etc.). The rank is displayed along with the total number of students, such as **"Rank 3 out of 20"**, helping students understand their position in the class. This feature allows students and administrators to quickly identify top-performing students and compare overall academic performance.

```
private static function countingSort($students) {
    $maxMarks = 100;
    $count = array_fill(0, $maxMarks + 1, []);
    // Linear Mapping
    foreach ($students as $student) {
        $marks = intval($student['avg_marks']);
        $count[$marks][] = $student;
    }
    // Rebuild in Descending Order
    $result = [];
```

```
    for ($i = $maxMarks; $i >= 0; $i--) {
        foreach ($count[$i] as $student) {
            $result[] = $student;
        }
    }
    return $result;
}
```

4.2. Testing

4.2.1 Test cases for Unit

Table 4.1: Admin Login Test Cases

S.N.	Payload	Expected Result	Actual Result	Result	Appendix
1	Username or Email: Sujal Password:	Please Fill out this Field. Or Wrong Email or password	Please Fill out this Field. Or Wrong Email or password	Pass	Appendix 4
2	Email: SujalChhetriKarki @gmail. com Password: admin12	Logged in. Then, redirected to dashboard.	logged in and redirected to dashboard.	Pass	Appendix 3
3	Username or Email:RachanaKC @gmail.com Password: Rachana@123	Logged in and redirected to dashboard	Logged in and redirected to dashboard	Pass	Appendix 3

Table 4.2: Student Login Test Cases

S.N.	Payload	Expected Result	Actual Result	Result	Appendix
1	Username or Email: Rachana Password: 1234	Please Fill out this Field. Or Wrong Email or password	Please Fill out this Field. Or Wrong Email or password	Pass	Appendix 2
2	Username or Email: RachanaKC@gmail.com Password: Rk@123	Logged in and redirected to dashboard	Logged in and redirected to dashboard.	Pass	Appendix 2

Table 4.3: Teacher Login Test Cases

S.N.	Payload	Expected Result	Actual Result	Result	Appendix
1	Username or Email: Rachana Password: 1234	Please Fill out this Field. Or Wrong Email or password	Please Fill out this Field. Or Wrong Email or password	Pass	Appendix 2
2	Username or Email: RachanaKC@gmail.com Password: KC@123	Logged in and redirected to dashboard	Logged in and redirected to dashboard.	Pass	Appendix 2

Table 4.4: Counting Sort Algorithm Test Cases

S.N.	Payload	Expected Result	Actual Result	Result	Appendix
1	Rank of the Student	Shows Rank	Rank 1 of 10	Pass	Appendix 11
2	Rank of the student	Rank displayed successfully	Rank 4 of 10	Pass	Appendix 11

Table 4.5: Adding attendance Test Case

S.N.	Payload	Expected Result	Actual Result	Result	Appendix
1	Default Absent	Absent.	Successfully updated	Pass	Appendix 10
2	Present	Present	Updated	Pass	Appendix 10

Table 4.6: Pre-Admission Test Case

S.N.	Payload	Expected Result	Actual Result	Result	Appendix
1	Name, Email, DOB, Gender, Father Mother name, Address	Name, DOB, email, and Father name are required.	Name, DOB, email, and Father name are required.	Pass	Appendix 5
2	Waiting for approval of Pre-admission	Admin Access	Admin Access	Pass	Appendix 5

CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS

5.1. Conclusion

By completing the Student Performance Evaluation System project, we successfully developed a fully functional academic management platform. The system allows users to create accounts, log in, and manage academic records. Teachers and Admins can browse different categories of student data, view detailed performance reports, rank students using sorting algorithms, and proceed to generate evaluations. The payment process for pre-admission or fees is integrated with eSewa for secure and instant transactions. Once data is entered, the admin can view student details, update attendance and marks, and manage records. Admins can also add, edit, or delete student profiles and academic data using the system's administrative dashboard.

5.2. Lesson Learnt

The lessons that we learnt during the course of this project are given below.

- Time management:

The value of time management was one of the most significant things this project taught us. We came to understand how crucial it was to rank activities according to how complex the system's components were. We are able to easily meet project deadlines and optimize their process because of this competence.

- Communication skills:

The We improved our communication skills by coordinating with team members, discussing project ideas, and presenting our progress effectively.

- Problem-solving skills:

We were able to hone their problem-solving abilities thanks to the project's many hurdles. We gained knowledge on how to recognize various problems with the system and put fixes in place to fix them.

- Report-writing:

Along with learning how to utilize various case tools, such as use case diagrams, charts, data flow diagrams, and ER-diagrams, among others, we have also learned how to construct proposals and project-related documentation.

5.3 Future Enhancements

In the future, the Student Performance Evaluation System will feature several key enhancements. AI-driven predictive analytics will be integrated to forecast student performance trends and provide personalized learning suggestions for each student. A real-time notification system for attendance and grade updates will be introduced, allowing parents and students to stay informed instantly. Additionally, a mobile application will be developed to provide easier access to academic reports, notice boards, and internal messaging. A digital certificate generation module will also be added to automate the process of issuing transcripts and character certificates. These upgrades will make the Student Performance Evaluation System a more comprehensive and efficient platform for academic management.

REFERENCES

[1] “Student Performance Evaluation System - Overview,” ScienceDirect, [Online]. Available: https://www.sciencedirect.com/topics/computer-science/student-performance. [Accessed: Apr. 7, 2026].

[2] “Student Information System (SIS),” Oracle Corporation, [Online]. Available: https://www.oracle.com/education/student-information-system. [Accessed: Apr. 7, 2026].

[3] “Google Classroom - Education Tools,” Google, [Online]. Available: https://classroom.google.com. [Accessed: Apr. 7, 2026].

[4] “Moodle - Open-source Learning Platform,” Moodle, [Online]. Available: https://moodle.org. [Accessed: Apr. 7, 2026].

[5] “Blackboard Learn - LMS Platform,” Blackboard Inc., [Online]. Available: https://www.blackboard.com. [Accessed: Apr. 7, 2026].

[6] “Khan Academy - Student Performance Tracking,” Khan Academy, [Online]. Available: https://www.khanacademy.org. [Accessed: Apr. 7, 2026].

[7] “Education Management Information System (EMIS),” UNESCO, [Online]. Available: https://emis.uis.unesco.org. [Accessed: Apr. 7, 2026].

APPENDIX A

- **Home Page**

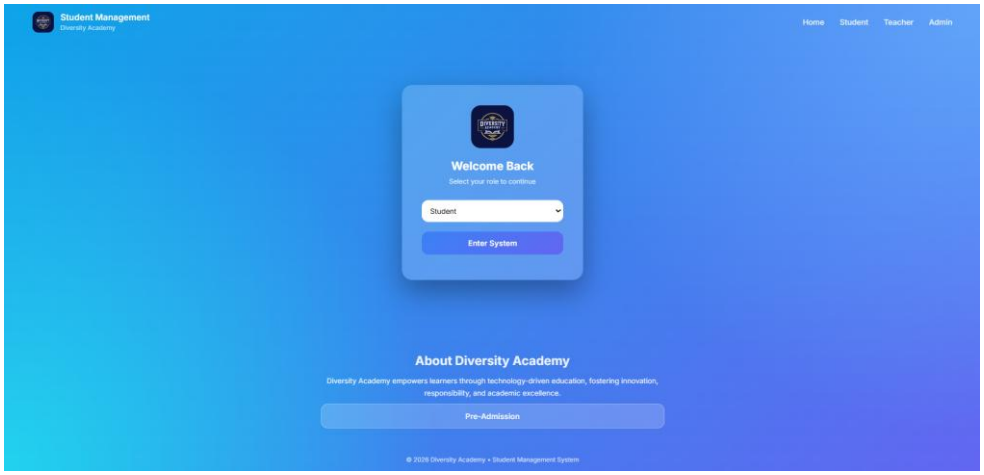


Figure A1: Home Page

- **User Student/Teacher Login**

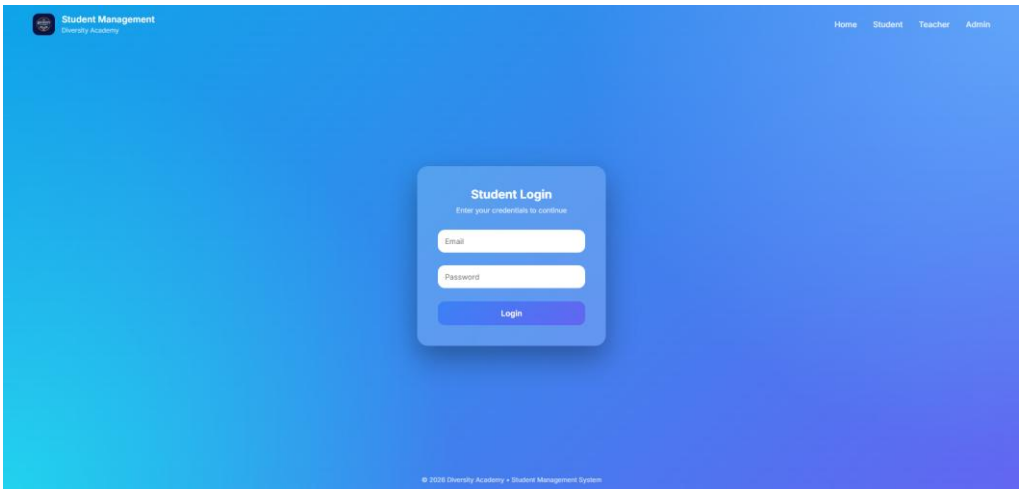


Figure A2: Student Login Page

- **Admin Login Page**

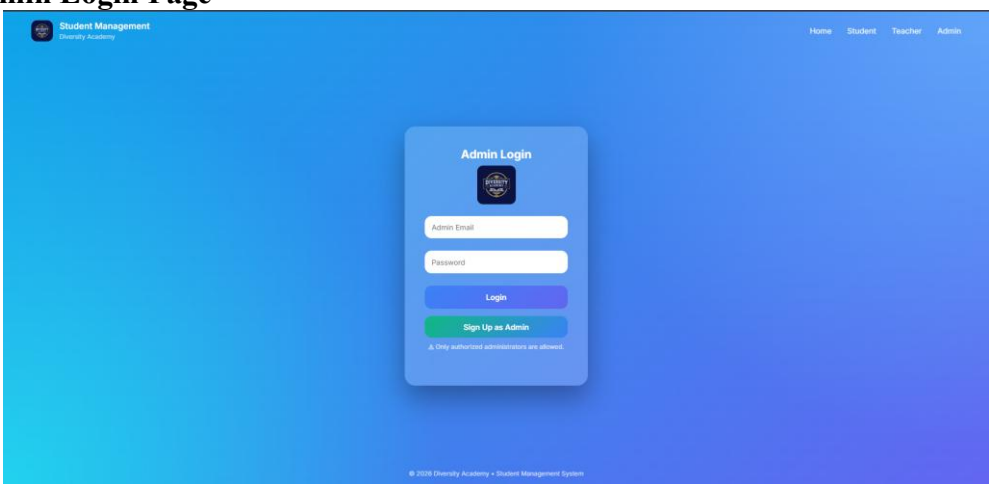


Figure A3: Admin Login Page

• **Admin Login Unsuccess**

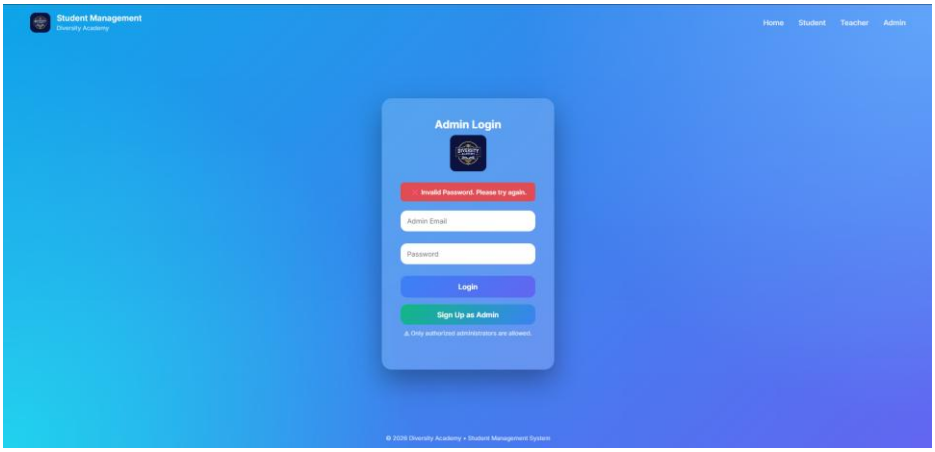


Figure A4: Admin Login Unsuccess

• **Pre-admission Page**

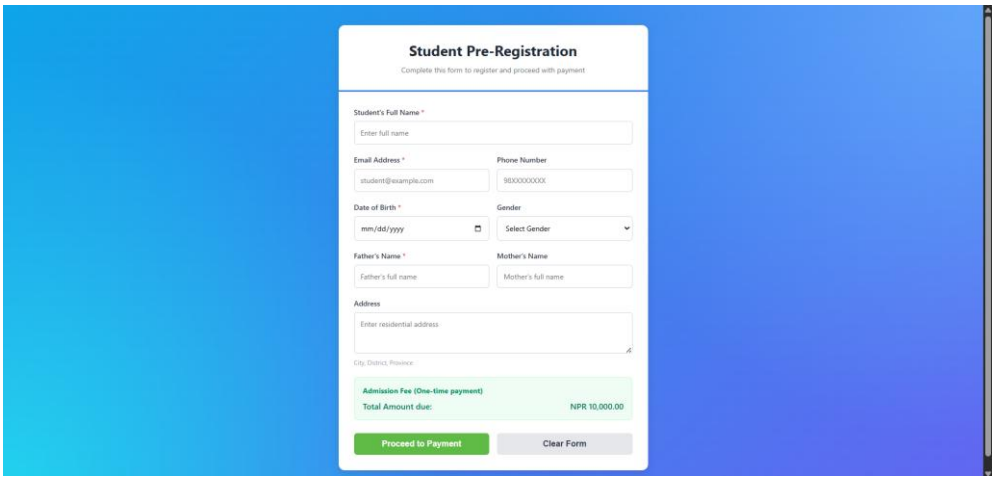


Figure A5: Payment Page

• **Admin Dashboard Page**

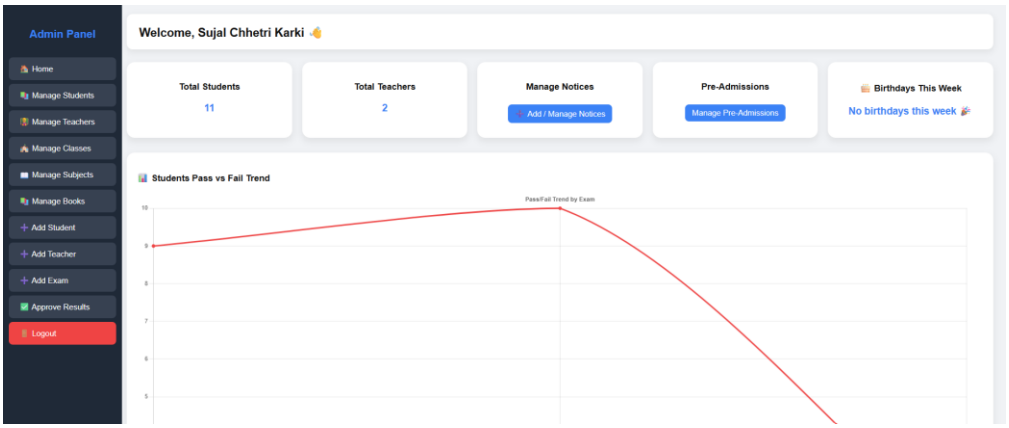


Figure A6: Admin Dashboard Page

• **Students Register Page**

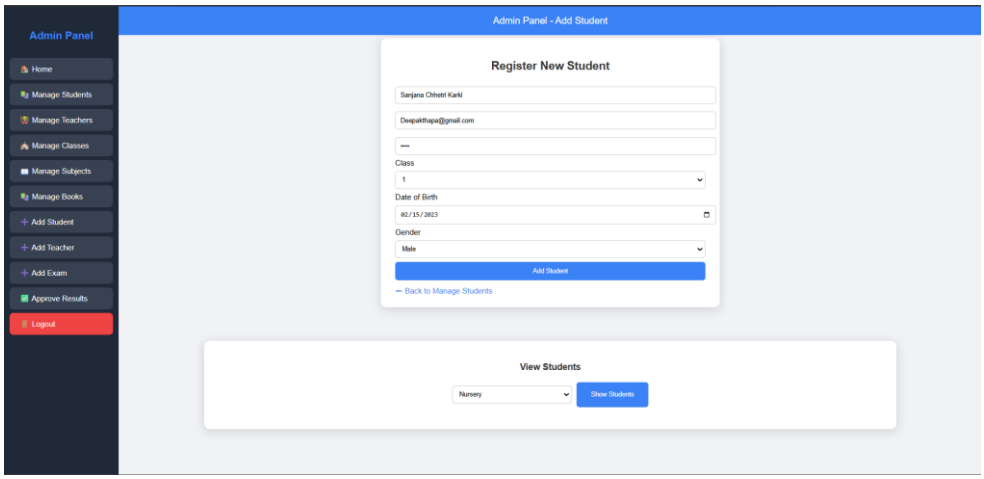


Figure A7: Student Registration Page

• **Manage Students Dashboard Page**

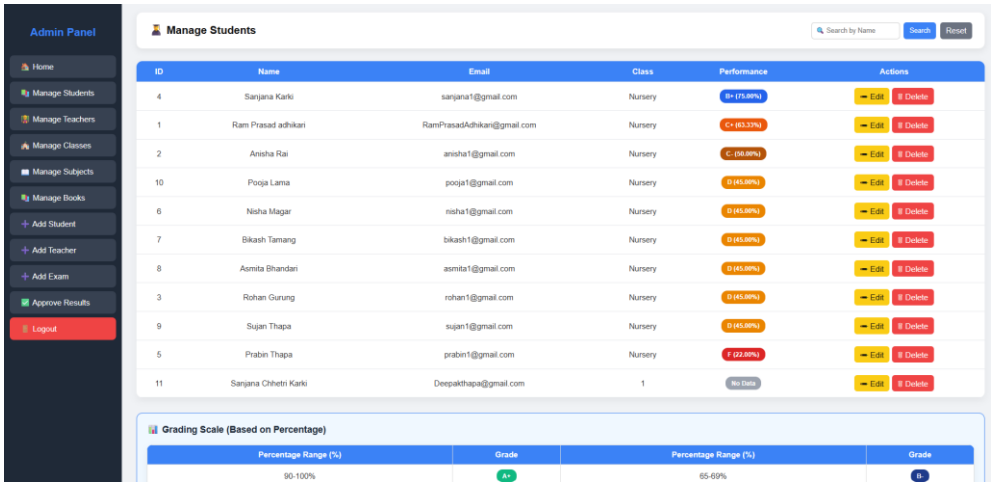


Figure A8: Student Manage Page

• **Edit Students Info**

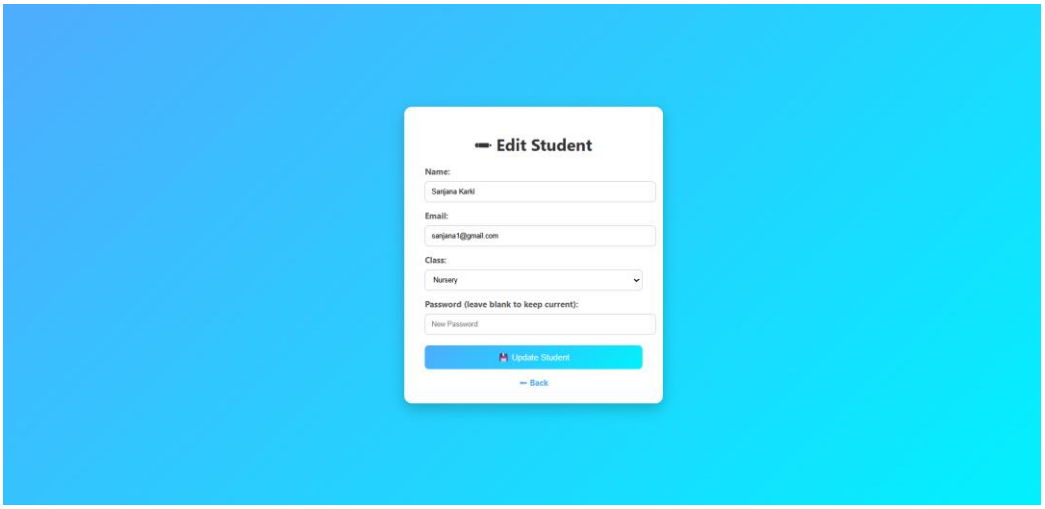


Figure A9: Edit Students

• **Students Dashboard Page**

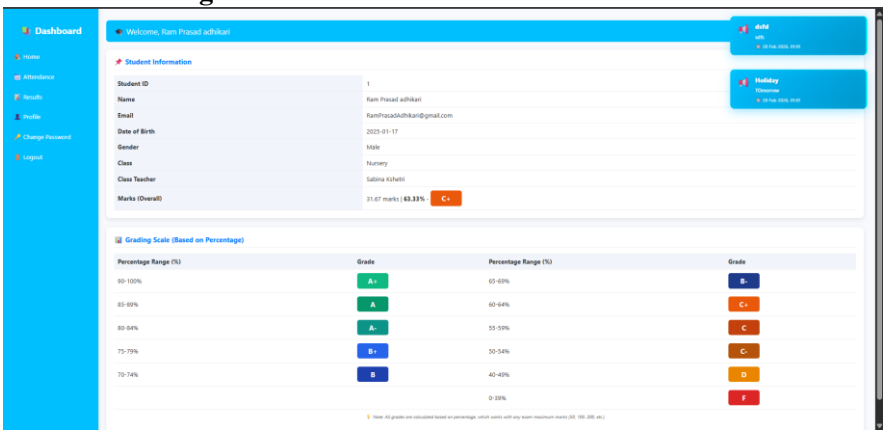


Figure A10: Student Dashboard Page

• **Teachers Dashboard Page**

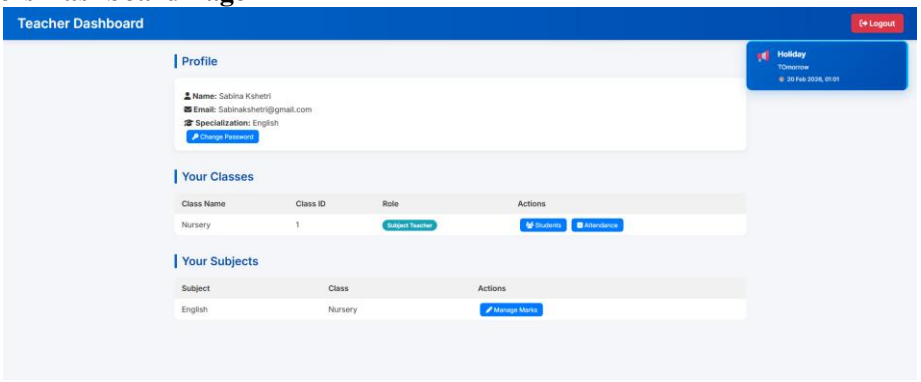


Figure A11: Teacher Dashboard Page

• **Attendance Dashboard Page**

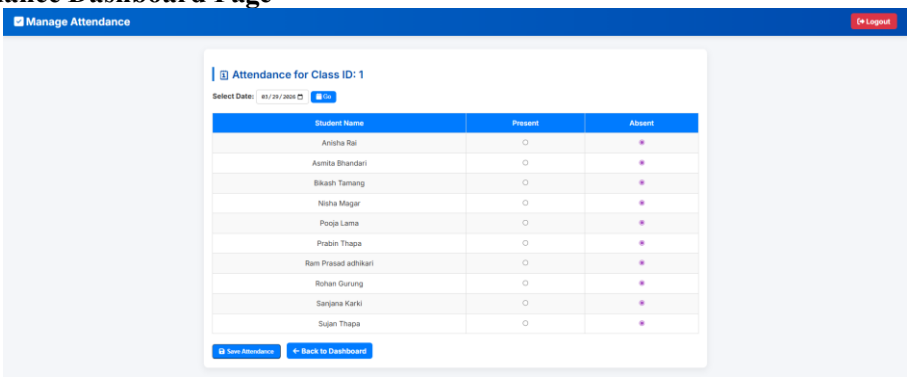


Figure A12: Attendance Dashboard Page

• **Ranking Algorithm Page**

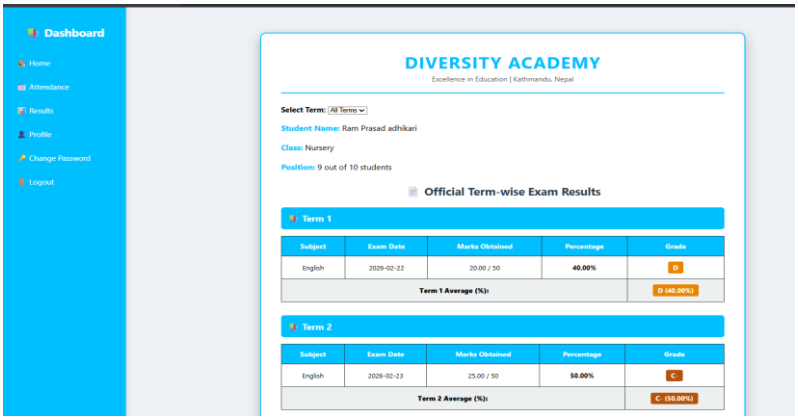
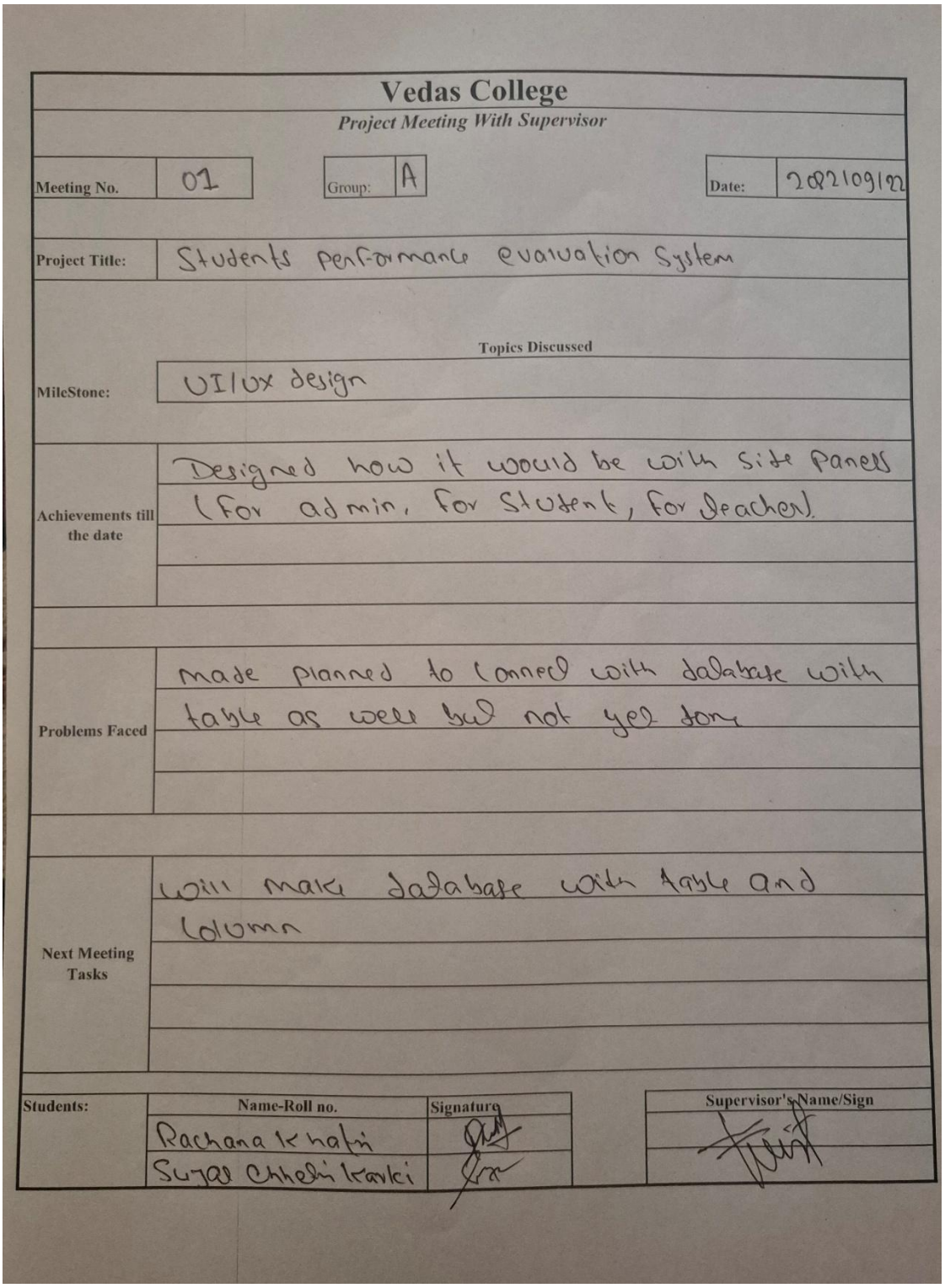


Figure A13: Counting Algorithm Page

APPENDIX B

- **Meeting B1**



Vedas College

Project Meeting With Supervisor

Meeting No.	01	Group:	A	Date:	2021/09/22

Project Title: Students performance evaluation System

Topics Discussed

MileStone: UI/UX design

Achievements till the date: Designed how it would be with site panels (For admin, for Student, for Teacher).

Problems Faced: made planned to connect with database with table as well but not yet done

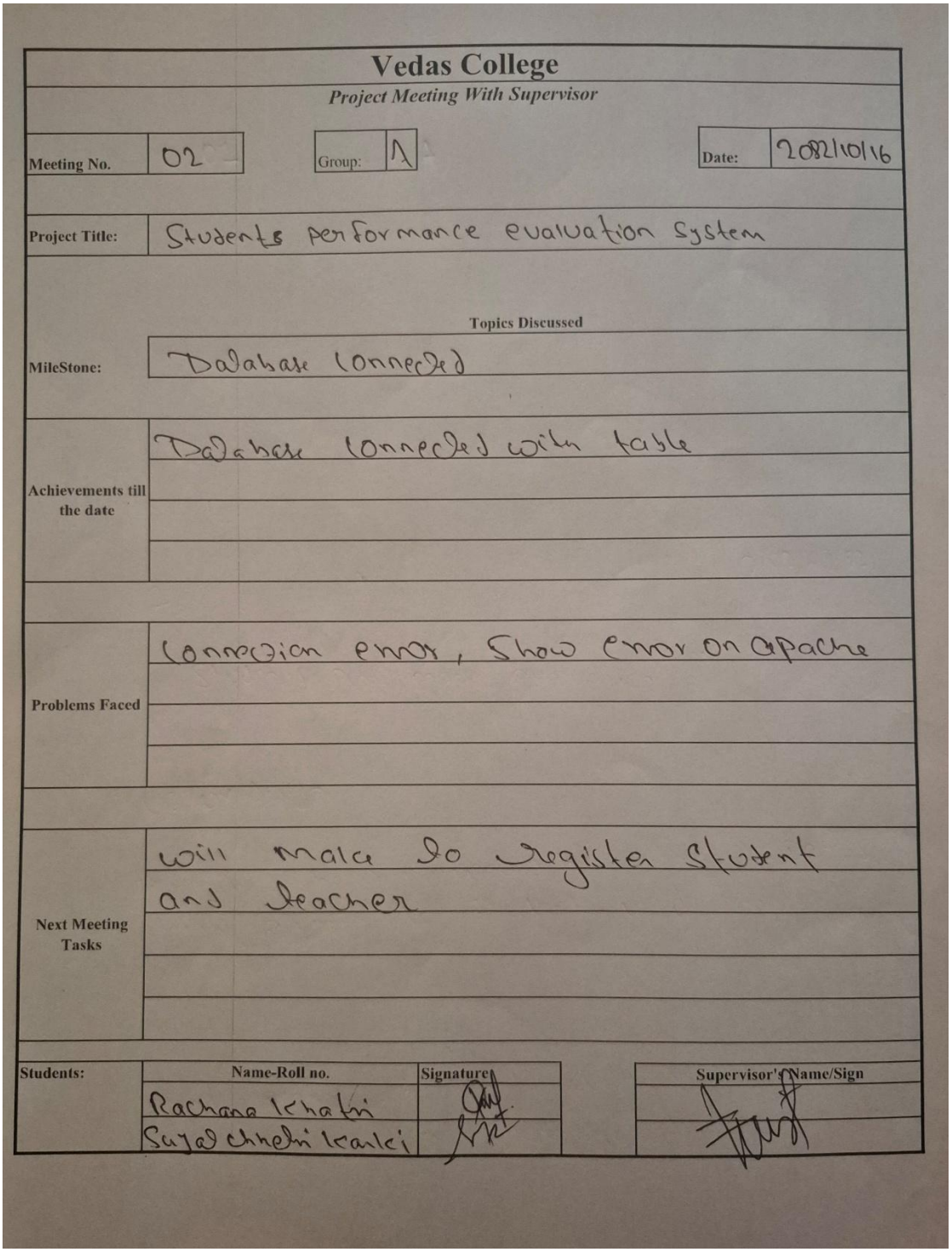
Next Meeting Tasks: will make database with table and column

Students:	Name-Roll no.	Signature
	Rachana Khatri	
	Surya Chhetri Kavki	

Supervisor's Name/Sign

Figure B1: Meeting 1

- **Meeting B2**



Vedas College

Project Meeting With Supervisor

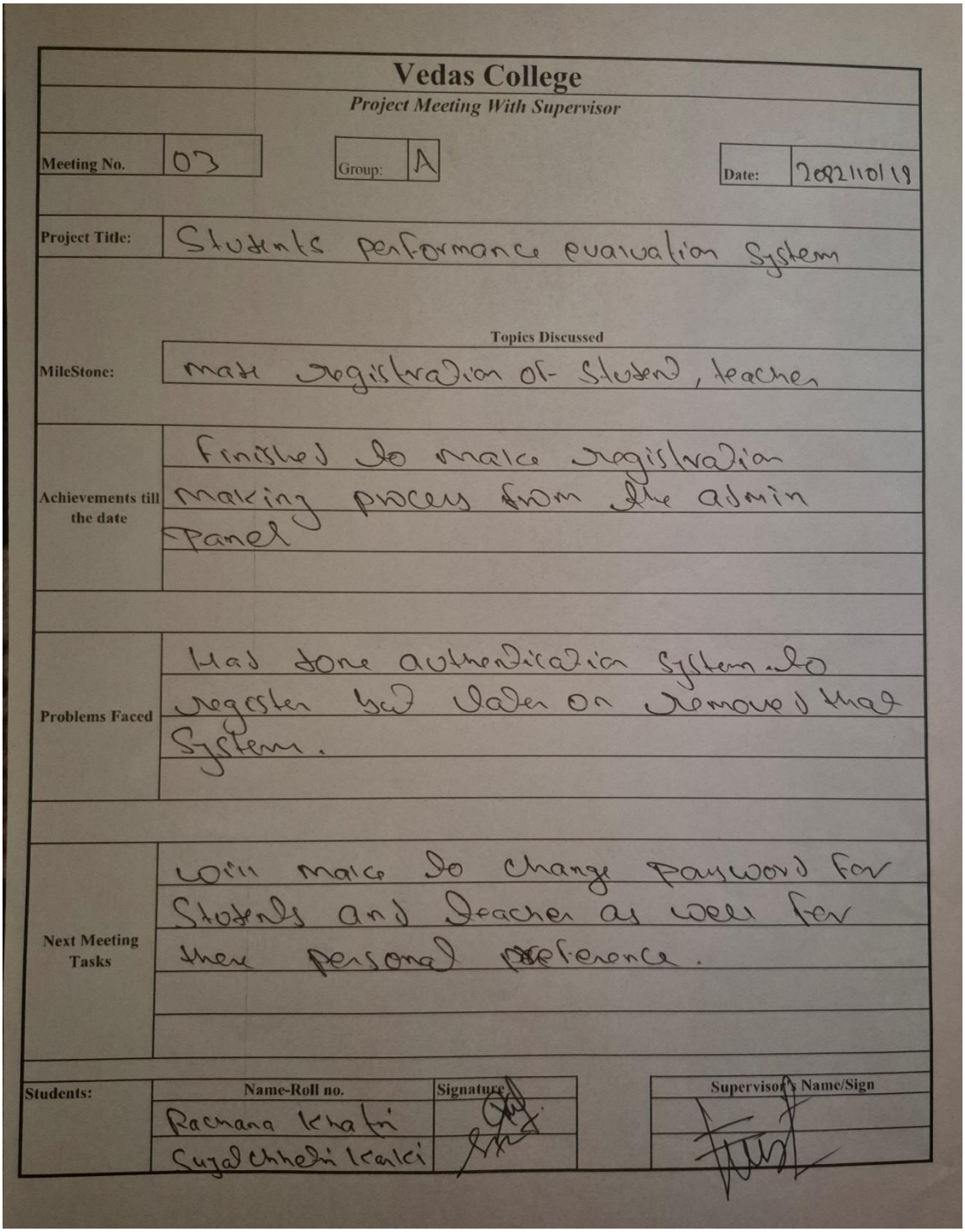
Meeting No.	02	Group:	A	Date:	2021/10/16

Project Title:	Students performance evaluation system
	Topics Discussed
MileStone:	Database connected
Achievements till the date	Database connected with table
Problems Faced	Connection error, Show error on Apache
Next Meeting Tasks	will make to register student and teacher

Students:	Name-Roll no.	Signature	Supervisor's Name/Sign
	Rachana Ichati		
	Sujal Chhetri Ichangi		

Figure B2: Meeting 2

- **Meeting B3**



Vedas College

Project Meeting With Supervisor

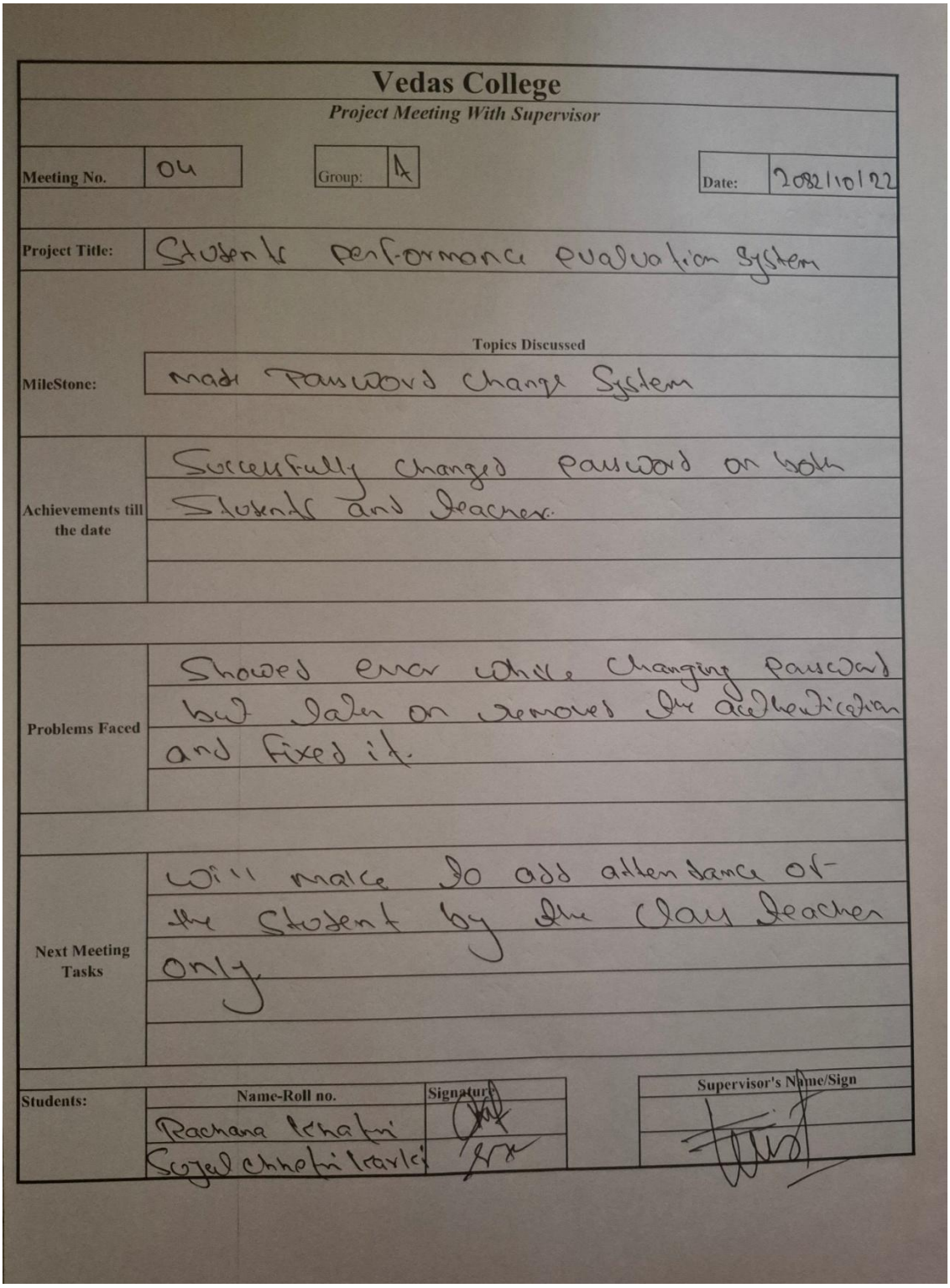
Meeting No.	03	Group:	A	Date:	2021/10/19

Project Title:	Students performance evaluation System
	Topics Discussed
MileStone:	main registration of- student, teacher
Achievements till the date	Finished to make registration making process from the admin Panel
Problems Faced	Had done authentication system. to register by user on remove that System.
Next Meeting Tasks	will make to change password for Students and Teacher as well for their personal preference.

Students:	Name-Roll no.	Signature	Supervisor's Name/Sign
	Rachana Khatri		
	Sujal Chhetri Kalci		

Figure B3: Meeting 3

- **Meeting B4**



Vedas College

Project Meeting With Supervisor

Meeting No.	04	Group:	A	Date:	2082/10/22

Project Title: Student performance evaluation System

Topics Discussed

MileStone: made Password change System

Achievements till the date: Succesfully changed password on both Student and Teacher.

Problems Faced: Showed error while changing password but later on removed the authentication and fixed it.

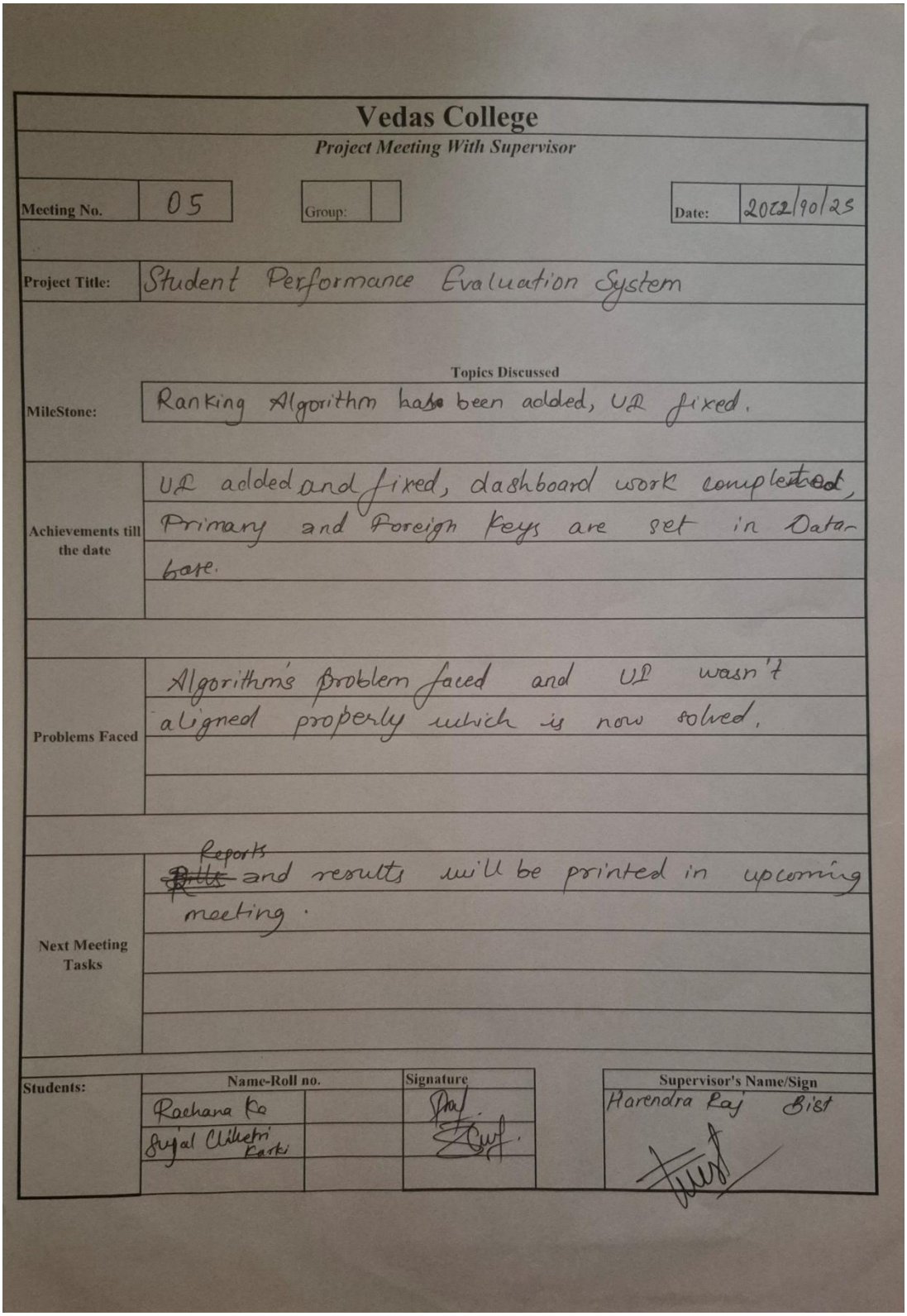
Next Meeting Tasks: Will make to add attendance of the Student by the class Teacher only.

Students:	Name-Roll no.	Signature
	Rachana Nepali	
	Sujal Chhetri (Leader)	

Supervisor's Name/Sign

Figure B4: Meeting 4

- **Meeting B5**



Vedas College

Project Meeting With Supervisor

Meeting No.	05	Group:		Date:	2022/10/25

Project Title: Student Performance Evaluation System

Topics Discussed

MileStone: Ranking Algorithm has been added, UI fixed.

Achievements till the date: UI added and fixed, dashboard work completed, Primary and Foreign keys are set in Database.

Problems Faced: Algorithms problem faced and UI wasn't aligned properly which is now solved.

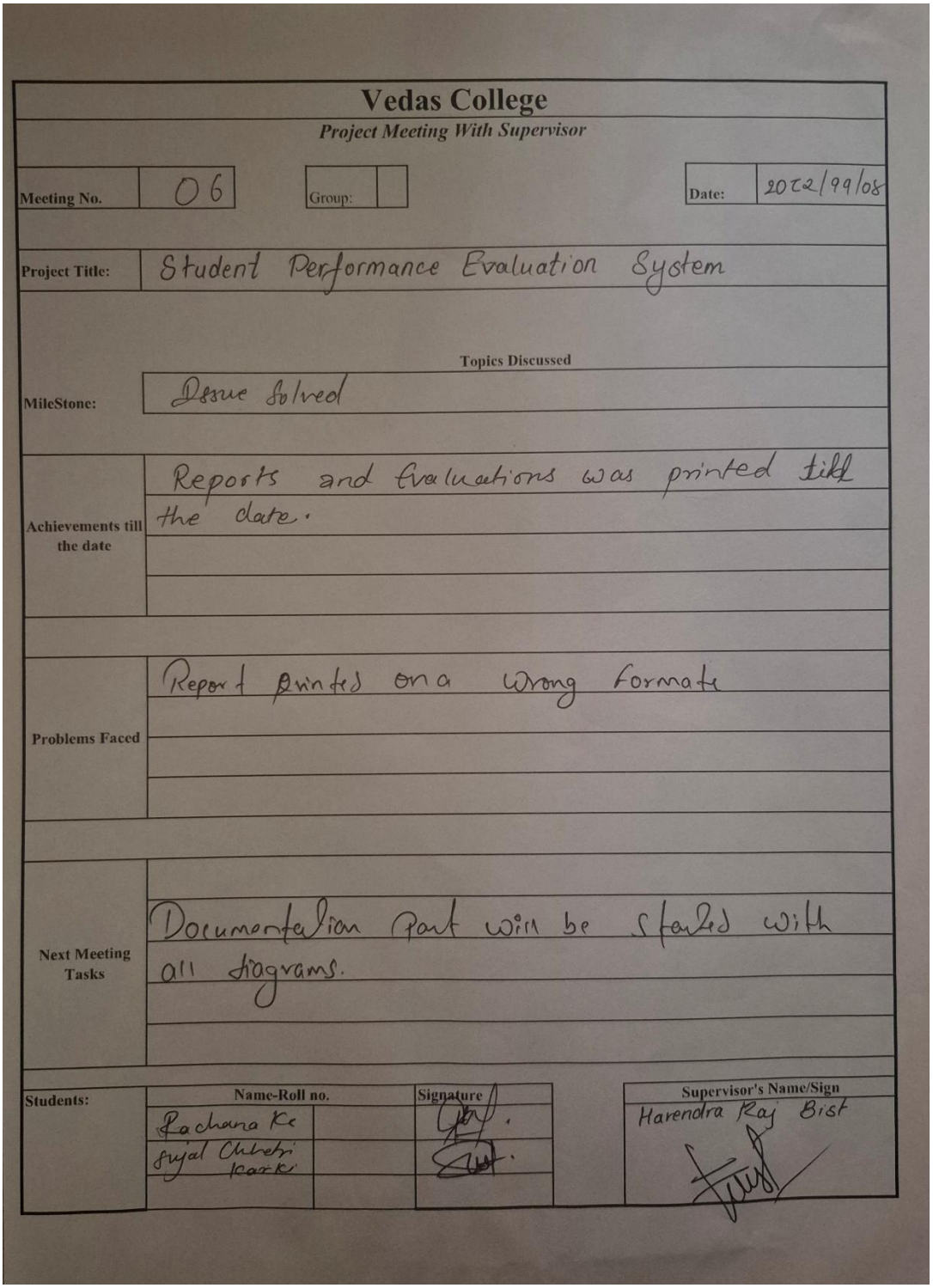
Next Meeting Tasks: ~~Bills~~ Reports and results will be printed in upcoming meeting.

Students:	Name-Roll no.	Signature
	Rachana Ka	
	Sujal Chhetri Karki	

Supervisor's Name/Sign: Harendra Raj Bist

Figure B5: Meeting 5

- **Meeting B6**



Vedas College

Project Meeting With Supervisor

Meeting No.	06	Group:		Date:	2022/09/08

Project Title: Student Performance Evaluation System

Topics Discussed

MileStone: Issue Solved

Achievements till the date: Reports and Evaluations was printed till the date.

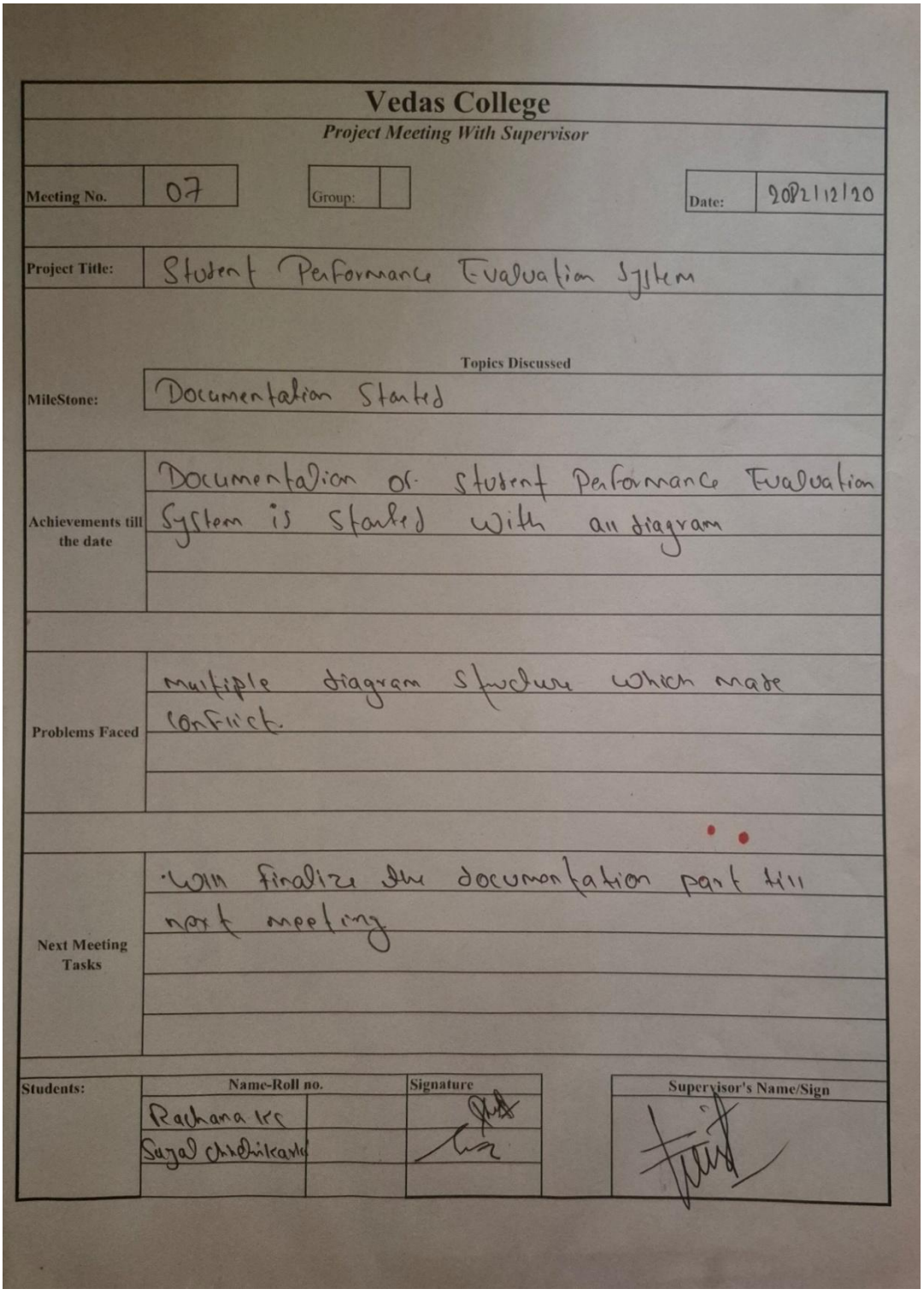
Problems Faced: Report printed on a wrong formate

Next Meeting Tasks: Documentation part will be started with all diagrams.

Students:	Name-Roll no.		Signature	Supervisor's Name/Sign
	Rachana K.C			Harendra Raj Bist
	Sujal Chhetri			

Figure B6: Meeting 6

- **Meeting B7**



Vedas College

Project Meeting With Supervisor

Meeting No.	07	Group:		Date:	2021/12/20

Project Title: Student Performance Evaluation System

Topics Discussed

MileStone: Documentation Started

Achievements till the date: Documentation of student performance Evaluation System is started with all diagram

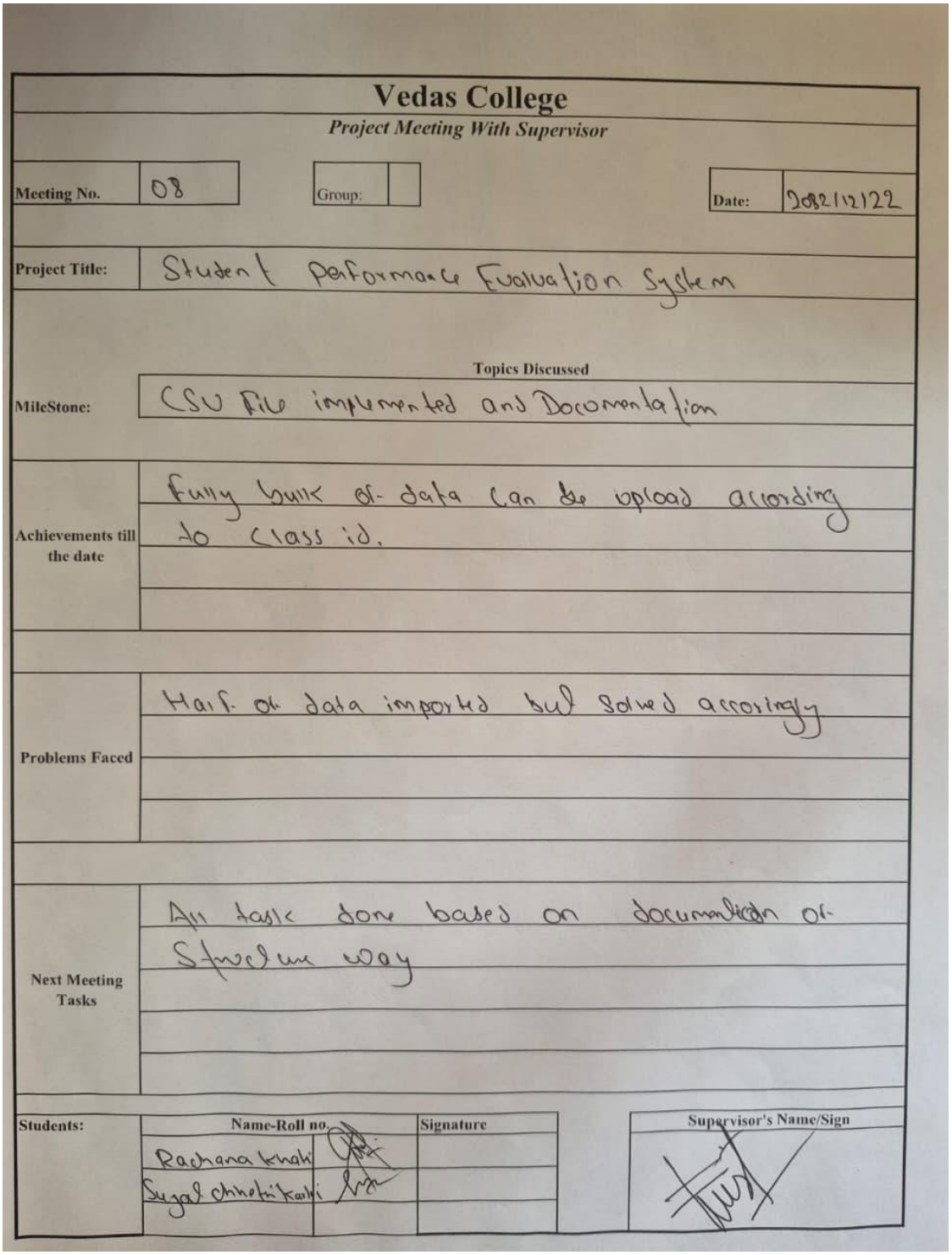
Problems Faced: multiple diagram structure which make conflict.

Next Meeting Tasks: will finalize the documentation part till next meeting

Students:	Name-Roll no.		Signature	Supervisor's Name/Sign
	Rachana KC			
	Sajal Chhetri			

Figure B7: Meeting 7

- **Meeting B8**



Vedas College

Project Meeting With Supervisor

Meeting No. 08 | Group: | Date: 2082/12/22

Project Title: Student Performance Evaluation System

Topics Discussed

MileStone: CSV file implemented and Documentation

Achievements till the date: Fully built of data can be upload according to class id.

Problems Faced: Half of data imported but solved accordingly

Next Meeting Tasks: All task done based on documentation of Structure way

Students:	Name-Roll no.	Signature	Supervisor's Name/Sign
	Rachana Khati		
	Sugal chhetri Karki		

Figure B8: Meeting 8